

Empirical Quantitative Analyses of Research Software Engineering Projects in
Scientific Computing.

by

Samuel David Schwartz

A dissertation accepted and approved in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in Computer Science

Dissertation Committee:

Stephen F. Fickas, Co-Chair

Boyana Norris, Co-Chair

Daniel Lowd, Core Member

Michal Young, Core Member

Joanna Goode, Institutional Representative

University of Oregon

Summer 2024

© 2024 Samuel David Schwartz
All rights reserved.

DISSERTATION ABSTRACT

Samuel David Schwartz

Doctor of Philosophy in Computer Science

Title: Empirical Quantitative Analyses of Research Software Engineering Projects in Scientific Computing.

This dissertation is about empirically-driven quantitative and qualitative analyses of software projects in two large ecosystems of research production in the United States: national laboratories and universities.

It is grounded in the fields of software engineering and software repository mining. In the 2002 paper, “What makes good research in software engineering” the authors identified five categories of software engineering research questions and gave examples. Three of these categories include:

- *Method or Means of Development* (e.g., How can we do/create X?)
- *Generalization or Characterization* (e.g., What, exactly, do we mean by X? What are the important characteristics of X, What are the varieties of X, and how are they related?)
- *Design, Evaluation, or Analysis of a Particular Instance* (e.g., How does X compare to Y? What is the current state of S/practice of P?)

This work interrogates these three categories as applied to software engineering projects categorized under the umbrella of the emerging field of research software engineering. Focused on the domains of the United State’s national laboratories and universities due to their high levels of publicly available research output, we are asking the following research questions (RQs):

RQ1 How can we find open source software repositories connected to universities and national laboratories? (Method of Development)

RQ2 Given our methodology, what is the current state of affairs? Just how many open source software repositories and projects affiliated with universities and national laboratories are out there? (Analysis of a Particular Instance/Domain)

RQ3 What are the properties, characteristics, and varieties of software projects with a nexus to these research institutions? (Generalization or Characterization, Analysis of a Particular Instance/Domain)

RQ4 How do the characteristics of repositories in the university ecosystem compare with the characteristics of repositories in the national laboratory ecosystem? (Analysis of a Particular Instance/Domain)

RQ5 How does the code in these research projects relate with and depend on other projects in the ecosystem? (Generalization or Characterization)

In this work we contextualize these questions with background information and answer each in turn.

This dissertation includes previously published and unpublished coauthored material.

ACKNOWLEDGEMENTS

I could not have succeeded in this program without the kindness, support, and wise advice from my two co-advisors, Steve and Boyana. Taking a risk on me, and supporting me through these last several years has been a wonderful journey of growth and learning. Thank you!

It takes a village to graduate a PhD student, and I am incredibly indebted to the rest of my committee. Daniel has always been willing to lend a listening ear, and I thank him for the 2-3 years of work we did together when I was the student representative to the departmental graduate education committee. Learning from Michal's talent in effectively running large lecture undergraduate classes is worth a dissertation on its own, and I'm grateful for his mentorship whenever I was fortunate enough to be his GE. I am also so grateful to Joanna for her insights on navigating the world of CS education research.

I am also indebted to many other administrators and faculty members from across the university who I've met through my various activities over the years. I think it's safe to say that I've been more involved in some of the "back of house" operations of making a university run than most graduate students, and I appreciate the mentors which have helped me learn just how to navigate and change university systems in order to make the University of Oregon a little better for everyone.

The support I've received from other graduate students has been so helpful. Graduate programs are hard. My peers have made the road easier. Thank You!

Lastly, I can't forget the unconditional encouragement and support I've received from my friends, family, and loved ones. I have a large family – dozens of first cousins. I'm the first in this large extended family, to my knowledge, to earn a PhD.

I'm lucky to have had their cheerleading through the good times and the discouraging times. Particularly from my faithful dog, Chico, who has been with me since my first semester of graduate school seven years ago when I adopted him from the local animal shelter as a puppy.

.....

I started my program academic year 2019-2020, the year the global pandemic of COVID struck and protests engulfed the country. This included over 100 days of widespread civil unrest over racial injustice in Oregon. We also battled catastrophic wildfires in 2020, with ashes that rained from the sky and smoke so thick it was hard to breath. This was followed by inflation at the highest rates in decades and no meaningful increases in pay for several years. This year, a fever pitch of tension related to the controversial Israel-Hamas war caused high profile protests/riots on campuses nationwide, including at UO.

At a department-wide meeting of graduate students at the beginning of this academic year, the chair of the department's graduate education committee asked all the CS graduate students gathered to raise their hands by cohort. Only a handful from my year or above remained, and no students remained from the 2020-2021 cohort. Graduate programs are tough, and I respectfully submit my cohort had some heavy duty issues to contend with, above and beyond the traditional rigors of a PhD program. I want to acknowledge and express my gratitude for these challenges, because I've grown from them, too.

To my many mentors. Including my first computing mentor, Andrew Dennis Lloyd;
may you rest in peace.

And to my students. No matter what my job title, may I always be as good a
teacher to others as the mentors I have been blessed to learn from.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	21
II. RELATED WORK: A FIVE-YEAR SURVEY OF LITERATURE IN SOFTWARE ENGINEERING AND REPOSITORY MINING RESEARCH	28
2.1. Summary of the Literature Review	28
2.2. Methodological Approach	30
2.2.1. First Step: Amass a large collection of works	30
2.2.2. Second Step: Filter the collection of works	30
2.2.3. Third Step: Sort works into topics	31
2.2.4. Fourth Step: Sort papers in each topic into four distinct categories	32
2.2.5. Fifth Step: Identify cross-cutting themes	33
2.2.6. Results	34
2.3. Topic: Code writing and refactoring	35
2.3.1. Tools	35
2.3.2. Psychology / Social Science	39
2.3.3. Analysis / Broader Studies	43
2.3.4. Curated Datasets	43
2.4. Topic: Code comprehension	45
2.4.1. Tools	45
2.4.2. Psychology / Social Science	46
2.4.3. Analysis / Broader Studies	49
2.4.4. Curated Datasets	51
2.5. Topic: Smells and quality	52

Chapter	Page
2.5.1. Tools	52
2.5.2. Psychology / Social Science	53
2.5.3. Analysis / Broader Studies	53
2.5.4. Curated Datasets	54
2.6. Topic: Whole project aspects	56
2.6.1. Tools	56
2.6.2. Psychology / Social Science	57
2.6.3. Analysis / Broader Studies	61
2.6.4. Curated Datasets	64
2.7. Topic: Human and team dynamics	66
2.7.1. Tools	66
2.7.2. Psychology / Social Science	68
2.7.3. Analysis / Broader Studies	73
2.7.4. Curated Datasets	73
2.8. Topic and cross cutting theme: Machine Learning	75
2.8.1. Raw Inputs and Final Outputs	79
2.8.2. Input Transformation and Output Transformation	81
2.8.3. Papers relating to an Algorithm, Model Architecture, or Tuning	84
2.8.3.1. Curated Datasets	86
2.9. Cross cutting theme: Bots	87
2.10. Cross cutting theme: Venues	89
2.11. Conclusion	91
III. INVENTORY OF SOFTWARE REPOSITORIES IN NATIONAL LABORATORIES	92
3.1. Brief Summary of Chapter	92

Chapter	Page
3.2. Introduction and Motivation	93
3.3. What are all the open-source software repositories with a nexus to a national laboratory?	95
3.3.1. Web Scraping	96
3.3.1.1. Results	96
3.3.2. Manually Searching on GitHub	98
3.3.2.1. Results	99
3.3.3. Spack Mining	101
3.3.3.1. Results	101
3.3.4. Consolidation	103
3.4. Analyzing Project Popularity	103
3.4.0.1. Results	107
3.5. Identifying Repositories in Need of Sustainability Supports	107
3.6. Sidebar: An analysis mining GitHub via BigQuery	110
3.7. Conclusion	114
IV. INVENTORY OF SOFTWARE REPOSITORIES IN U.S. UNIVERSITIES	116
4.1. Brief Summary of Chapter	116
4.2. Introduction	116
4.3. RQ1: Projects with a nexus to a major US research universities	118
4.3.1. Definitions, Initial Scope, Overarching Approach:	118
4.3.2. Approach 1: Scraping university websites	118
4.3.2.1. Results	119
4.3.3. Approach 2: Searching on GitHub	119
4.3.3.1. Results	121
4.4. RQ2: Is a given repository an RSE repository?	122

Chapter	Page
4.4.1. Results	125
4.5. RQ3: Popularity of RSE Projects	137
4.6. RQ4: Which RSE projects are active? Which are on life support? How do they differ?	139
4.7. Future Work	140
4.8. Conclusion	142
V. DISCUSSION: COMPARISON BETWEEN NATIONAL LABORATORIES AND UNIVERSITIES AND CHARACTERISTICS OF RSE PROJECTS	144
5.1. Chapter Summary	144
5.2. Properties, characteristics, and varieties of research institution linked GitHub repositories at national laboratories and research universities	144
5.2.1. Language	146
5.2.2. License	147
5.2.3. Size in KB	147
5.2.4. Forks	150
5.2.5. Stars	150
5.2.6. Time Since Repository Creation	152
5.2.7. Time Since Last Push	153
5.2.8. Has {feature} or Is {property}	153
5.3. Classifying RSE projects as lab-related or university-related	153
5.4. Correlations	154
5.5. Taxonomy of RSE projects	157
5.5.1. Classification	157
5.5.2. Nearby galaxies	158
5.6. Discussion and Conclusion	159

Chapter	Page
VI. DEPENDENCY RELATIONSHIPS	161
6.1. Chapter Summary	161
6.2. Methodology	161
6.3. Results	162
6.3.1. Step 1 and 2: Find a representative set of RSE repositories and obtain their dependencies	162
6.3.1.1. What is the distribution of “first order” dependencies per repository?	164
6.3.1.2. What is the distribution of “first order” dependencies by package manager?	166
6.3.1.3. What is the distribution of package managers per repository?	166
6.3.1.4. Of the set of immediate “first order” dependencies, how often is each used by an RSE repository?	168
6.3.2. Step 3 and 4: Construct a network and find key nodes	174
6.3.2.1. Node Importance Flow Algorithm	175
6.4. Conclusion and Future Work	183
VII. CONCLUSION	184
7.1. Research Questions and Answers	184
7.1.1. RQ1: How can we find open source software repositories?	184
7.1.2. RQ2: What is the current state of affairs?	185
7.1.3. RQ3 and RQ4: What are the properties, characteristics, and varieties of software projects with a nexus to these research institutions? How do the characteristics of repositories in the university ecosystem compare with the characteristics of repositories in the national laboratory ecosystem?	186
7.1.4. RQ5: How does the code in these research projects relate with and depend on each other project in the ecosystem?	186

Chapter	Page
7.2. Limitations and Future Work	186
7.3. Contributions	187
 APPENDICES	
A. TAXONOMY MATRIX OF TOPICS, CATEGORIES, AND CROSS CUTTING THEMES	189
A.1. Learning / School	189
A.2. Onboarding	191
A.3. New features and requirements	192
A.4. Code writing and refactoring	193
A.5. Help, Q&A, code comprehension, and documentation consumption	198
A.6. Documentation production	204
A.7. Testing	206
A.8. Commits, merges, and conflicts	207
A.9. Pull requests and code reviews	210
A.10.Smells and quality	213
A.11.Maintainability, technical debt, production performance	215
A.12.Bugs, faults, and vulnerabilities	219
A.13.Traces, links, and context	224
A.14.Deprecation	227
A.15.Whole project / repository aspects and status	228
A.16.Human and team dynamics	233
A.17.Machine learning foundations	239
A.18.Software Engineering / Repository Mining Research Meta Analysis	242
A.19.Input-Output Types of Machine Learning Tools	243
B. RSE DEPENDENCIES	259

Chapter	Page
REFERENCES CITED	683

LIST OF FIGURES

Figure	Page
1. Tools in the code comprehension space	46
2. Components of a typical pipeline in an ML-based tool.	79
3. Histograms of repositories per stars and per forks, on a \log_{10} scale. We use these histograms to construct criteria for whether a repository is sufficiently popular and has a community for further analysis. We assert a repository is more likely to have a community if it has at least six stars or six forks (red lines).	105
4. Combined 2D bin plot, showing the number of repositories per each star-fork intersection on a \log_{10} - \log_{10} - \log_2 scale. Note that the vast majority of repositories are those which we assert are not popular enough for further analysis and are highlighted by the red bounding box of six forks and six stars.	106
5. Histogram of the last time someone pushed to a repository, going back five years. We consider the 2,005 repositories identified in RQ2 with a link to a national laboratory and a likely community. The red line is the six month mark.	108
6. Graph T_{14} . Universities where the number of RSE repositories they have in common is greater than or equal to 14 repositories. Nodes are labeled by university and (total number of RSE repositories) in parentheses, edges are labeled by the number of RSE repositories in common.	127
7. Histograms of the number of stars and forks per RSE repo. The red line, six in both cases, is a threshold used in Schwartz, Fickas, Norris, and Dubey (2024) to indicate an RSE repository has a community. These histograms seem to follow that same trend.	138
8. Matrix view of the number of repositories per star-fork pair. The Pearson correlation coefficient of Stars and Forks is 0.865, which is in line with previous research. Yamamoto, Kondo, Nishiura, and Mizuno (2020)	138

Figure	Page
9. Histogram of the last time an RSE project with community received a push. The red line indicates the six month mark. The blue line indicates the two year mark. These lines form the arbitrary threshold boundaries we selected for “healthy,” “dying,” and, “dead” repositories, which also match prior work Schwartz et al. (2024).	139
10. Violin plots of the size of RSE repositories in kilobytes at research universities and national laboratories.	149
11. Violin plots of forks of RSE repositories at research universities and national laboratories. The red line is 6, our cutoff in previous chapters for community.	149
12. Violin plots of stars of RSE repositories at research universities and national laboratories. The red line is 6, our cutoff in previous chapters for community.	151
13. Violin plots of the time since a repository was created of RSE repositories at research universities and national laboratories.	151
14. Violin plots of the last time a repository had a push of RSE repositories at research universities and national laboratories.	152
15. Histogram of the number of repositories associated with each dependency.	164
16. Percentage of all RSE repositories with a given number of dependencies. Fitted trendline in blue and red.	166
17. Cullen and Frey graphs for the logged data shown in Figure 16, with distribution observations color-coded as matching red and blue.	167
18. Distribution of dependency importance by our flow algorithm score.	177

LIST OF TABLES

Table	Page
1. Table of United States Department of Energy National Laboratories and their websites.	97
2. Repositories found on each domain crawled with our web scraping spider.	98
3. Number of unique repositories with a legitimate nexus to a national laboratory, by grouping of initial search terms. Note that the results sum to a number much greater than the 6,864 unique repositories found. This is due to substantial overlap in search findings among several of the different search groupings. Search terms denoted with an * returned over 250 search results of individual repositories and organizational pages with multiple repositories listed – up to 55,070 repositories and organizations in the case of “Energy” – and these results were not examined further.	100
4. Search queries which were found in a Spack configuration file, by number of GitHub repositories found.	102
5. Number of GitHub commits in Google’s BigQuery archive service made with an email address containing gov.	112
6. Number of distinct GitHub repositories which a user with an email from the associated domain has ever committed to.	113
7. Domains where contributors from the two labs committed to 10 or more repositories. Domain refers to the domain of a contributors email address, repos are the number of distinct repos found for each domain, IS is the intersection size (i.e., the number of repositories in common), OC is the overlap coefficient, and SD is the Sorensen-Dice metric.	114
8. Comparison of repositories found in the US Department of Energy National Laboratory from the data obtained in writing Chapter III and US R1 Universities that permitted website scraping.	120
9. Summary of human and ChatGPT agreement when labeling repositories in two different sets as RSE or non-RSE	124

Table	Page
10. Summary statistics (minimum, 25%, median/50%, mean, 75%, and maximum) of the number of repositories with a nexus to a university, and the number of RSE repositories, found across all universities.	126
11. All GitHub repositories found by university, through both scraping .edu websites for links to GitHub and by searching for keywords related to the university on GitHub itself. The number of RSE repositories, as determined by ChatGPT provided probabilities with a threshold of at least a 75% likelihood, is also shown.	127
12. Top languages of healthy RSE repos:	141
13. Top languages of dying RSE repos:	141
14. Top languages of dead RSE repos:	142
15. Percentage differences in languages of RSE repositories at national laboratories vs research universities. Only languages with more than 1% prevalence in either labs or universities are shown. . .	147
16. Percentage differences in software licenses of RSE repositories at national laboratories vs research universities.	148
17. Summary statistics about the size of RSE repositories linked to National Laboratories and Research Universities.	150
18. Summary statistics about Forks in RSE repositories linked to National Laboratories and Research Universities.	150
19. Summary statistics about Stars in RSE repositories linked to National Laboratories and Research Universities.	150
20. Summary statistics about the time passed since repository creation, in months, in RSE repositories linked to National Laboratories and Research Universities.	152
21. Summary statistics about the time since the last push, in months, in RSE repositories linked to National Laboratories and Research Universities.	153
22. Summary statistics about various features and properties of RSE repositories linked to National Laboratories and Research Universities.	153
23. Correlations of metadata attributes of RSE repositories.	155

Table	Page
24. Number of Dependencies by Package Manager	168
25. Top 10 RSE Dependencies by Build System. The % is in reference to the number of repositories which utilize the corresponding build system manager.	169
26. Spearman correlation between various metrics of RSE dependency importance.	175
27. Primary input and output types of all machine learning model driven tools examined in this report.	244
27. Primary input and output types of all machine learning model driven tools examined in this report.	245
27. Primary input and output types of all machine learning model driven tools examined in this report.	246
27. Primary input and output types of all machine learning model driven tools examined in this report.	247
27. Primary input and output types of all machine learning model driven tools examined in this report.	248
27. Primary input and output types of all machine learning model driven tools examined in this report.	249
27. Primary input and output types of all machine learning model driven tools examined in this report.	250
27. Primary input and output types of all machine learning model driven tools examined in this report.	251
27. Primary input and output types of all machine learning model driven tools examined in this report.	252
27. Primary input and output types of all machine learning model driven tools examined in this report.	253
27. Primary input and output types of all machine learning model driven tools examined in this report.	254
27. Primary input and output types of all machine learning model driven tools examined in this report.	255
27. Primary input and output types of all machine learning model driven tools examined in this report.	256

Table	Page
27. Primary input and output types of all machine learning model driven tools examined in this report.	257
27. Primary input and output types of all machine learning model driven tools examined in this report.	258
28. Dependencies used by more than 10 RSE repositories. The % is in reference to the number of repositories which utilize the corresponding build system manager.	259

CHAPTER I

INTRODUCTION

Academic scholarship in the field of research software engineering (RSE) is in its infancy. As we write these words in October 2023, the very first academic conference on this sub-field of software engineering, named simply “RSE,” is currently underway in Chicago. But just because the academic study of RSE is just beginning one shouldn’t mistake the target of this scholarly inquiry as new. On the contrary, research software engineering itself is as old as computing in science and is today a multi-billion dollar industry in the public domain alone. Namely, through direct or indirect government expenditures in academia and national laboratories, which collectively fund hundreds of thousands of scientists, professors, software developers, system administrators, graduate and undergraduate students, and technicians – all of whom are writing code with various levels of professionalism, technical difficulty, and organizational complexity in the advancement of a taxpayer funded research agenda.

So just who is a research software engineer, exactly? This is a somewhat open question. The Society of Research Software Engineering in the United Kingdom (“UK-RSE,” “RSE Society,” or simply the “Society”) is the oldest group dedicated to forming a distinct community around research software engineering, with its legal predecessors first formed in 2013. In the Society’s definition, “A Research Software Engineer (RSE) combines professional software engineering expertise with an intimate understanding of research.” In our reading, this is to say an RSE is first and foremost a professional software engineer, typically with credentialing indicta which includes a university degree in computer science, that happens to work on research projects.

The United States Research Software Engineer Association (“US-RSE”), which has its roots dating to the winter of 2017-2018, takes a more inclusive approach. US-

RSE states, “[R]esearch Software Engineers [(RSEs)] encompass those who regularly use expertise in programming to advance research. This includes researchers who spend a significant amount of time programming, full-time software engineers writing code to solve research problems, and those somewhere in-between. [RSEs] aspire to apply the skills and practices of software development to create more robust, manageable, and sustainable research software.”

These definitions are about engineers, the *people*. This work is tied to these related-yet-competing definitions, yet also different, in that this dissertation primarily focuses on software engineering *projects* with some nexus to research. We are especially interested in research projects which directly or indirectly produce artifacts that are intended to be shared with the broader world. Such projects might contain code that, while not a noteworthy contribution in-and-of itself, helps lead to peer-reviewed articles. Perhaps the contribution is the code itself, via open source software used by a wider community. The point being, just as the concept of research software engineers can have tighter, looser, or differently delineating definitions, so can research software engineering projects. This dissertation will explore the spectrum of research projects in Chapter V and contribute a theoretical framing for taxonifying these different flavors of research projects.

This dissertation, therefore, primarily focuses on software engineering projects with some nexus to research. We focus ourselves by identifying and mining software projects housed in open source GitHub repositories with a nexus to the national laboratories and universities of the United States. We choose this scoping due to the missions of public research that these groups promulgate, and the feasibility of working with GitHub at an empirical scale. With this scope in mind, we overview the remainder of this dissertation.

Chapter II

In Chapter II we first motivate and situate this work by conducting a literature review focused on the main software engineering and software repository mining conference venues. In this review, we taxonify existing tools for, and articles about, research oriented computing and software engineering. In undertaking this survey, we place a special focus on the advanced scientific computing (ASC) space. That is, we focus on the scholarship or the tools most likely to be applicable to or used by RSEs that would “count” under the UK Society’s narrower, more professionalized definition of an RSE, as ASC software tends to run on complex supercomputing clusters.

Bottom line: there is a lot of interesting research about – and tools for – general software engineering. But there is very little scholarship or tools specifically tailored to the RSE experience. It also quickly becomes clear that no single entity even knows how many software repositories exist in the university or national laboratory ecosystems. To say nothing about what flavors of tools or scholarship might benefit the RSE community from the perspective of an empirical inventory of existing repositories or from the existing literature in the most mainstream software engineering and repository mining research venues.

Co-author Acknowledgment: This chapter contains both unpublished and published material with and without co-authorship. Specific contributions are detailed in the beginning of the chapter, but co-authors include Dr. Steve Fickas, Dr. Boyana Norris, and Dr. Michal Young.

Chapter III

This lack of tools, data, or scholarship around research software engineers and engineering uncovered by the literature review (and further evidenced by the creation of the new RSE conference) motivates us to Chapter III, which is the first inventory

of open source GitHub repositories with a nexus to the US Department of Energy's National Laboratories. To once again control for scope, we initially assert that each repository forms its own project (an assertion which we will interrogate and couch with nuance in later chapters.) In this chapter, we ask the following three research questions, which are sub-questions of those RQs identified in the abstract:

1. What are all of the public software projects / repositories with a nexus to the US Department of Energy or its national laboratories? How do we find them? (Sub-Questions of RQ1 and RQ2)
2. Of the projects / repositories found in (1.), which ones are actively used by people outside of the project's core developers? That is, which projects are used by a community? (Subquestion of RQ3)
3. Of the projects / repositories used by a community, which are still actively developed or maintained? Are there unmaintained projects with community use? (Subquestion of RQ3)

We are motivated by these questions, in part, due to their political and budgetary implications. Understanding which labs are creating and maintaining many projects in a metric of productivity. Given the traditionally cyclical nature of Department of Energy funding processes, understanding which projects are (un)maintained yet have community use provides justification for allocating sustainability support funding to these projects. Understanding just how many projects there are, and the types of projects involved in a lab's ecosystem, helps illustrate the potential budget implications for policymakers.

Beyond business decisions, the inventory also allows for subsequent mining of these repositories. Among other analyses that can be done, this mining can include

monitoring repositories for security vulnerabilities (e.g., did an intern commit code with passwords in the clear?), which is a particularly important concern given that many of these laboratories develop the science undergirding the US’s nuclear program. The inventory also makes the creation of RSE-specific machine learning tools significantly easier, since there’s now an actual corpora of RSE-specific software data to train against.

Co-author Acknowledgment: This chapter contains both unpublished and published material with and without co-authorship. Specific contributions are detailed in the beginning of the chapter, but co-authors include Dr. Steve Fickas, Dr. Boyana Norris, and Dr. Anshu Dubey.

Chapter IV

In Chapter IV we ask the same three research sub questions as in Chapter III and apply a similar methodology in answering these questions, but target a different ecosystem; a closely adjacent ecosystem which we also assert, almost by definition, to contain RSE projects: institutions of higher education in the United States (e.g., R1 universities).

Co-author Acknowledgment: This chapter contains both unpublished and published material with and without co-authorship. Specific contributions are detailed in the beginning of the chapter, but co-authors include Dr. Steve Fickas and Dr. Boyana Norris.

Chapter V

We then compare and contrast the findings from Chapters III and IV in Chapter V, where we answer RQ4 in full. As part of this discussion, we create and empirically motivate a theoretical taxonomy for RSE projects and construct a framework for understanding some of their common characteristics, to fully answer RQ3. It is in

this chapter which we will tease out the nuances of different RSE projects and, just as the term “research software engineer” can have somewhat differing definitions (as seen by the UK-RSE Society and US-RSE Association’s similar-yet-subtly-different definitions), explore how different labels of “research software engineering project” might or might not be applied to various categories of projects found in our inventory of repositories linked to the national laboratory and higher education ecosystems.

Co-author Acknowledgment: This chapter contains both unpublished and published material with and without co-authorship. Specific contributions are detailed in the beginning of the chapter, but co-authors include Dr. Steve Fickas and Dr. Boyana Norris.

Chapter VI

Critical to understanding RSE repositories and projects is their interdependence on other repositories and projects, both RSE and non-RSE. To that end, in Chapter VI we answer RQ5 as we investigate RSE project and repository dependency through an exploratory methodology using tools from graph theory and machine learning, which is grounded in physical-world analogs of supply chain management and bills of materials.

Co-author Acknowledgment: This chapter contains both unpublished and published material with and without co-authorship. Specific contributions are detailed in the beginning of the chapter, but co-authors include Dr. Steve Fickas and Dr. Boyana Norris.

Chapter VII

Finally, we recap the work we’ve done in our conclusion in Chapter VII, where we also outline the many ample areas for future work in this new field of inquiry, and how this dissertation will help inform those groundbreaking areas.

Co-author Acknowledgment: This chapter contains no unpublished nor published co-authored material.

To summarize, in addition to conducting an extensive literature review (Chapter II), the contributions of this dissertation are threefold:

1. We contribute groundbreaking quantitative insights about RSE projects by instantiating a broad theoretical framework for mining software repositories with domain-specific treatments in the United States' national laboratories and universities. These insights have public policy and budgetary implications for stakeholders. (Chapters III, IV; RQ1, RQ2)
2. From these quantitative insights, we contribute a theoretical framework for understanding common characteristics of projects in the RSE community through a positivist-constructionist approach based on empirical data. We also hold a discussion of the inherent limitations in our framework's construction. (Chapter V; RQ3, RQ4)
3. We contribute a methodological approach for analyzing software dependencies in the RSE ecosystems using graph theoretic techniques. (Chapter VI; RQ5)

This work is exploratory. There is lots still to uncover. But we are excited to outline new findings that, in addition to shedding light on an unexplored corner of science, also have policy implications around software in science.

Note on Co-Author Acknowledgments: To clearly separate from technical content, details of publication status and co-authorship are denoted in an italic font at the beginning of each chapter or appendix. Note that unpublished technical content may be revised and submitted for publication with yet unknown co-authors going forward.

CHAPTER II

RELATED WORK: A FIVE-YEAR SURVEY OF LITERATURE IN SOFTWARE ENGINEERING AND REPOSITORY MINING RESEARCH

The organization of related work described herein was developed in collaboration with Dr. Steve Fickas and Dr. Boyana Norris as part of my comprehensive written qualifying examination, known as the area exam, for which Dr. Michal Young was an evaluating committee member. I did the primary work of reading all papers, categorization of papers, and composition of text. All other authors assisted with editing and discussions in an advisory capacity.

2.1 Summary of the Literature Review

To orient our work, it is helpful to understand the landscape of the existing software engineering literature. In the long run, this dissertation is the first of many steps to better understand how to create tools to further research software engineering projects.

More specifically, this survey of literature is based on my interest in software engineering tools to support the advancement of advanced scientific computing (ASC). Building on my prior experience in the ASC domain and master's thesis work, I also have particular interest in how machine learning can be leveraged to support the development of these tools.

I surveyed over 1,500 papers, primarily from three venues: the MSR, ICSE, and ICSME conferences for the five years from 2018 to October 2022. These papers were filtered and taxonified into 18 different topics in an approach detailed in Section 2.2.

I further synthesized papers from five topics out of the 18: code writing and refactoring (Section 2.3), code comprehension (Section 2.4), smells and quality (Section 2.5), aspects of development applicable to an entire software

project (Section 2.6), and human dynamics within a project or development team (Section 2.7).

In addition to the 18 topics identified above, six cross-cutting themes emerged. I focus and elaborate on three of these cross-cutting themes: machine learning (Section 2.8), bots (Section 2.9), and specific venues; e.g., open source projects, start ups, national labs, etc. (Section 2.10).

I summarize key takeaways from this work in the conclusion (Section 2.11). Namely, there is very little published research on software engineering tools specifically tailored to the work of developers of advanced scientific computing projects, such as those commonly encountered in national laboratory environments. Consequently, this is a fruitful area for future research, which is explored in subsequent chapters of this dissertation.

2.2 Methodological Approach

The survey of the literature was done from a *tabula rasa* angle, and done in several steps. This section describes each of these steps which were taken to collect and taxonify the extant literature.

2.2.1 First Step: Amass a large collection of works. In pursuit of my aim to survey the literature broadly, I collected the titles and abstracts of all works published in the three conferences and associated journals of Mining Software Repositories (MSR), International Conference on Software Engineering (ICSE), and International Conference on Software Maintenance and Evolution (ICSME) from 2018 to October 2022. Over 1,500 works resulted from this initial query.

2.2.2 Second Step: Filter the collection of works. I then filtered these 1,500+ works against several porous analytical sieves. One was the necessity of the work being a research paper or well documented tutorial of a tool. Panel discussions or other presentations without an accompanying manuscript were excluded. A second porous sieve used to filter the corpora of works was a manuscript's relevance to my interests. Specifically, the broad criterion for relevance used as I read these hundreds of abstracts was the following:

Related somehow (even loosely) to tools, dynamics, or other analytical framework utilizing graph theory or machine learning techniques to analyze software developer communication, behavior, and git activity juxtaposed with development productivity or code and documentation quality metrics at any point in the software engineering/development process.

This filtering resulted in a selection of just over 400 papers.

2.2.3 Third Step: Sort works into topics. I manually sorted these 400+ filtered papers into 18 topics, some of which I elaborate on more in this work. These topics are:

- Learning or school/education related papers
- New developer onboarding
- New features and requirements
- Code writing and refactoring (Section [2.3](#))
- Code comprehension, and documentation consumption (Section [2.4](#))
- Documentation production
- Testing
- Commits, merges, and conflicts
- Pull requests and code reviews
- Smells and quality (Section [2.5](#))
- Maintainability, technical debt, production performance
- Bugs, faults, and vulnerabilities
- Traces, links, and context
- Deprecation
- Whole project / entire repository aspects and status (Section [2.6](#))
- Human and team dynamics (Section [2.7](#))

- Machine learning foundations (Section 2.8)
- Papers on research in software engineering and repository mining, such as literature reviews

These broad 18 topics are more formally defined in Appendix A.

2.2.4 Fourth Step: Sort papers in each topic into four distinct categories. Within each of the 18 topics, I further partitioned each paper into the general categories of “Tools,” “Psychology or Social Science,” “Analysis or Broader Studies,” and “Curated dataset.” I define these categories as:

- A “*Tools*” paper is any work which introduces a software artifact or other device which their target audience can use as part of their software development toolkit.
- A “*Psychology or Social Science*” paper is one where human beings were heavily used in the research methodology. (Excluding those tool papers where humans were simply used to validate the utility of the tool.)
- A “*Curated dataset*” paper is a work where the authors provide data to facilitate future research by others. In rare instances this may also include a tool, but a tool intended to be primarily used by others doing research in software engineering or repository mining – not typical developers.
- An “*Analysis or Broader Studies*” paper is essentially any other work not already categorized by the above slots. However, these papers are often characterized by phrases like “empirical study,” “investigation,” “evaluation of ...” “relating X with Y,” “does Z implicate W?,” and so forth.

Papers were filtered into one or more of these 18 topics \times 4 categories. This formed a matrix of papers, which are all listed in Appendix A.

2.2.5 Fifth Step: Identify cross-cutting themes. In sorting papers into this initial placement of topics and categories, there were additional themes that emerged which cut across the 18 topics \times 4 categories grid described above. These cross-cutting themes were identified and relevant papers annotated in a second round of assessment. The cross-cutting themes identified are:

- Bots
- Diversity, Equity, and Inclusion (DEI); Specifically identifying the following sub-themes:
 - DEI relating to Race/Ethnicity/National Origin,
 - DEI relating to Sex/Gender,
 - General DEI / Other DEI
- Graph Theory
- Machine Learning
- Privacy or Security
- Venue specific works; Specifically identifying the following sub-themes:
 - Academia, Industry, Open Source, Start Up, National Laboratory / ASC

2.2.6 Results. The final filtering for all papers can be found in Appendix [A](#).

Five topics were selected for further synthesis: code writing and refactoring (Section [2.3](#)), code comprehension (Section [2.4](#)), smells and quality (Section [2.5](#)), aspects of development applicable to an entire software project (Section [2.6](#)), and human dynamics within a project or development team (Section [2.7](#)).

Additionally, three cross-cutting themes receive additional treatment: machine learning (Section [2.8](#)), bots (Section [2.9](#)), and specific venues; e.g., open source projects, start ups, national labs, etc. (Section [2.10](#)).

The next few sections of this review focus on selected papers from these five topics and three cross-cutting themes.

2.3 Topic: Code writing and refactoring

The topic of “code writing and refactoring,” for our purposes, is the act of a developer actually constructing, copying, small-level editing, and wholesale revising of source code in some language(s) – which typically happens in a text editor or integrated development environment (IDE).

2.3.1 Tools. New tools created in the last five years designed to support developers in the process of writing and rewriting code generally fall into one of three categories:

1. *Automatic code writing.*

This category includes tools designed to predict what the developer will do next in the (new) code writing process, with a machine learning engine usually running the tool.

2. *Existing code analysis.*

This category includes tools designed to analyze and help a developer refactor and revise existing code, such as ensuring naming consistency in APIs.

3. *Specialized IDEs or IDE enhancements.*

This category includes tools – usually GUI dashboards or plug-ins – in an IDE, or entire IDEs themselves, to enhance or enable some aspect of code writing.

These three categories are not mutually exclusive. Rather, the tools available in the code writing and refactoring space often fall into two or more of these categories. The rest of this subsection provides illuminating examples of tools in each of these three categories.

1. *Automatic code writing* tools in the modern era almost uniformly utilize some sort of machine learning algorithm under the hood. While the exact and

idiosyncratic methodologies vary by tool for training and validation, the trend is that machine learning algorithms have gotten more and more accurate at predicting what a developer is going to type next, with [Bulmer, Montgomery, and Damian \(2018\)](#) reporting 64% accuracy in their 2018 code prediction tool, [T. Nguyen, Vu, and Nguyen \(2019a\)](#) reporting 70% accuracy in their 2019 tool, and a plateau thereafter, with [Wen, Aghajani, Nagy, Lanza, and Bavota \(2021\)](#) reporting a mere increase of 1% and an overall accuracy of 71% in their tool in 2021.

What has increased in machine learning tools since 2019 is a wider breadth of supported languages, or specializations within those languages. [Wen, Ferrari, et al. \(2021\)](#) supports Android projects, for example. And, as an example of machine learning code prediction tools specializing for niches within a language, [Mir, Latoškinas, Proksch, and Gousios \(2022\)](#) leverages a new standard in Python which allows for static typing. The tool predicts what type a variable is (string, int, float, etc.) allowing a developer to quickly add static types to existing code, which makes the performance of this modified code much more efficient when executed with newer Python interpreters. Such work might be relevant to ASC codes, which are often decades old. Tools to “auto upgrade” the code itself to new language standards might find a lot of utility in older and densely technical codes which contain many routine “no brainer” changes to be made.

Other developments in the machine-learning-for-code-writing space focus on improving the machine learning-based tools themselves. In [Svyatkovskiy et al. \(2021\)](#), the authors note that deploying code completion machine learning tools are often extremely resource intensive – to the point their use can bog down a developer’s machine. Svyatkovskiy et al. present a “suggested options tool” which takes up just 6MB of RAM, and the authors claim 90% accuracy in their top-5 suggestions

(which is roughly on par with other works claiming about 70% accuracy for their top-1 suggestion). This feat of equivalent accuracy to existing systems but with less resource consumption by a code prediction oracle was done by relying less on machine learning training and more on static analysis, and then cleverly blending the results of the two techniques.

Blending static analysis with machine learning prediction, however, presents trade offs. The larger the code base of a project (or set of reference projects), the more difficult it is to integrate static analysis techniques (e.g., clone identification and consequent suggestion) as part of a code prediction tool. On the other hand, limiting the amount of static analysis done by a tool requires more leaning on a pre-trained neural network, which can be large and bulky to include in an IDE. This is the challenge that [Silavong, Moran, Georgiadis, Saphal, and Otter \(2022\)](#) grappled with, and presented a code hashing solution as a possible compromise between the machine learning and static analysis tensions, with computation times for code search suggestions on the order of hundreds of times faster than Facebook’s 2019 code recommendation tool, Aroma [Luan, Yang, Barnaby, Sen, and Chandra \(2019\)](#).

2. Existing code analysis tools can be analogized to spelling and grammatical checking tools of the word processing world – they help make existing code better, rather than directly suggest its initial creation. In particular, these tools are largely focused on (i) improving code readability by analyzing class and method names and suggesting changes; (ii) analyzing overarching patterns and anti-patterns at a project-wide level and providing suggestions or warnings; and (iii) analyzing how code has changed over time to predict upcoming needed changes.

Examples of improving code readability by analyzing class and method names and suggesting changes include [Liu et al. \(2019\)](#), which presents a tool with a binary

output: does a method named X actually do X? That is, if I have method called `func square(x){return x*x;}`, the tool would return “True.” But if my method was `func square(x){return x;}`, the tool would return, “False.” Perhaps it’s unsurprising, given the difficulty of the problem, that the tool only had a 25% accuracy – which was much better than the 1% accuracy of its comparator. Still, it was able to identify 66 real-world method-name inconsistencies in the wild, indicating the utility of the tool as a possible “let’s double check this” warning to a developer. Comparing the results of this tool with nominal documentation may prevent code smells in musty software.

A different flavor of tool, but in the same vein as [Liu et al. \(2019\)](#), was presented in [S. Nguyen, Phan, Le, and Nguyen \(2020\)](#). This tool suggested method names based on a method’s implementation. For example, if the code was `{return x*x;}` the tool might suggest the name `square`. The authors subsequently analyzed a large number of existing method names in open source projects with their tool and identified methods which the authors thought their tool’s suggested name was better. The new name was accepted in pull requests 74% of the time.

Examples of analyzing overarching patterns and anti-patterns at a project-wide level and providing suggestions or warnings include work like FOCUSP. [T. Nguyen et al. \(2019\)](#), which uses context aware mining to identify suggested API calls at the point of code writing, or at refactoring. It also includes tools like [Barbez, Khomh, and Guéhéneuc \(2019\)](#), which identified anti-patterns (anti-patterns defined as, “poor solutions to recurring design problems”) from a mix of structural and historical data from a project or code file’s git history to identify well known anti-patterns, such as the “God Class” anti-pattern – where everything and the kitchen sink is thrown into one massive, convoluted class.

Lastly, examples of analyzing how code has changed over time to predict upcoming needed changes include [Tsantalis, Mansouri, Eshkevari, Mazinianian, and Dig \(2018\)](#). This paper utilized techniques that analyzed abstract syntax tree change over time in mining to determine candidate code ready for refactoring without relying on excessive user-defined thresholds or non-generalizable default settings. This work was followed up by [Tufano, Pantiuchina, Watson, Bavota, and Poshyvanyk \(2019\)](#), which was the first machine-learning based paper which used time-series data to predict refactoring activities, particularly bug fixes in legacy code.

3. *Specialized IDEs or IDE enhancement* tools which don't have a machine learning or refactoring component are the least common type of tool encountered in the surveyed literature, but include works like [Hempel, Lubin, Lu, and Chugh \(2018\)](#), which presented an IDE with clickable widgets embedded in the GUI that allow developers to graphically drag-and-drop syntax. Essentially, an "adult" version of Scratch. Other exemplary tools include CodeRibbon [Klein and Henley \(2021\)](#), which provides an alternative visual layout for workspace management and is available as a plug-in for popular IDEs like Visual Studio Code or Atom. None of these tools are immediately applicable to the ASC paradigm in any differentiating way, but it's important to be aware of them.

2.3.2 Psychology / Social Science. There is a rich array of work which studies how humans act when writing code.

Two studies are noteworthy for their use of fMRI imaging of the brain. One study, [Huang et al. \(2019\)](#), compared data structure manipulations of lists, arrays and trees with other spatial rotation exercises on 76 participants. They found that they were all distinct but related neural tasks, and that the more difficult computer science problems required a higher cognitive load (i.e., measurable brain activity) –

eventually surpassing the cognitive load of the spatial reasoning problems participants grappled with. Another study, [Krueger et al. \(2020\)](#), compared code writing with prose writing, and found that the two activities are extremely dissimilar at the neural level. Code writing primarily activated the right hemisphere of the brain, while prose writing activated the left. Still, neurological studies of the software engineering process are in their infancy, and many questions remain. How does the writing of technical documentation of complex algorithms compare to pure code writing or pure prose writing, for example?

One limitation of fMRI scans is the amount of time one has to measure activity. Work such as [Bellman, Seet, and Baysal \(2018\)](#) recorded clicks and keyboard activity from IDEs long programming sessions, specifically focusing on contextualizing developer activity during build failures and debugger usage. They found that developers spend much of their time debugging code, and using breakpoints do help in that process. This high-level finding is unsurprising, but the details are helpful in creating models to measure developer productivity. As a crude example, one expects less lines of code to be produced when a developer is working on “harder” debugging issues than more straightforward logic. Recording developer behavior in an IDE environment also forms the basis for [Damevski, Chen, Shepherd, Kraft, and Pollock \(2018\)](#), which found certain coding behavior had probabilistic distributions analogous to natural language production and processing. This indicates that, while prose writing and code writing are distinct events at a biological neurological level in humans, machine learning techniques designed for natural language processing and prediction can be applied to code processing and prediction in analogous ways.

Pairing developer behavior with eye-tracking software has also been done. One study, [Abid, Sharif, Dragan, Alrasheed, and Maletic \(2019\)](#), found significant

differences in eye-tracking behavior between novice and expert developers, with experts spending less time on a function call and more time on the implementation for complex functions. They also found that both novices and experts revisit control flow terms (e.g., if-statements) in repeated but short bursts. With that said, other work, such as [Zyrianov, Guarnera, Peterson, Sharif, and Maletic \(2020\)](#), notes that there are technological shortcomings to most eye tracking + IDE software combinations. Namely, most eye tracking software before 2020 isn't that accurate when focused on typical-font sized lines of code which is being scrolled or moved – especially when the developer switches among different open tabs in a typical editor. They presented a tool called *Deja Vu* to fix these problems. Consequently, results from earlier work on code development and eye tracking should be taken with a grain of salt.

Of additional interest is the study of developer emotion during the code writing process. Some works in 2018 continue that year's trend of heightened community interest in sentiment analysis, finding that off the shelf tools for sentiment analysis usually didn't work in the code writing and comprehension arena (e.g., developers expressing happiness when a library worked or frustration when encountering a bug) [Calefato, Lanubile, Maiorano, and Novielli \(2018\)](#); [Lin, Zampetti, Bavota, et al. \(2018\)](#). Other work, such as [Girardi, Novielli, Fucci, and Lanubile \(2020\)](#), found that wearing a smartwatch-like wristband that measured electrodermal and heart activity reliably measured developer's emotional activity. Moreover, the authors found that, perhaps unsurprisingly, positive emotions were correlated with being “in flow” or in a productive coding state of mind, while negative emotions were correlated with frustration and roadblocks.

While the above studies have focused on developers as individuals, other studies have focused on code writing from a team or group identity perspective. For example,

in [Amlekar, Gamboa, Gallaba, and McIntosh \(2018\)](#) the authors examine whether software engineers, and other code writers like academic researchers, students, hobby programmers, etc. utilize code writing tools like auto-complete in different ways. They found inconclusive results, and suggested more work needs to be done to understand how different practitioners code. This inconclusive finding has particular relevance to the ASC situation, given the unique blends of professional software developer and professional scientists which tend to contribute to ASC code bases, and suggests more research needs to be done in this area.

Another example of work analyzing team-based code writing contributions include [Stefano, Pecorelli, Tamburri, Palomba, and Lucia \(2020\)](#), which purports that socio-technical incongruence leads to worse code. That is, poor coordination amongst developers in a team leads to increased technical debt in the project, thus requiring more refactoring down the road. The authors in [Stefano et al. \(2020\)](#) provide a possible framework to coordinate refactoring ideas. This work was timely, given the investigation of developer perceptions and decision making processes explored in [Alomar \(2019\)](#), which examined when a developer labeled a code change as a “refactor” in a commit message vs when a change wasn’t given that label.

Finally, we come full circle back to the utility of tools used to automatically analyze and suggest names discussed in the previous subsection in the study by [Alsuhaibani, Newman, Decker, Collard, and Maletic \(2021\)](#) which surveyed over 1,100 developers on standards for source code method names and analyzed the responses based on things like years of experience and programming language knowledge. The authors found high degrees of conformity on the importance of adhering to the established standards, and provided a foundation for automated method name

assessment which could then be used either by automated tools or during human-led code review.

2.3.3 Analysis / Broader Studies. Relative to the number of papers about tools and social science in the code writing space, there are fewer papers in the broader analysis category.

One theme which does show up are empirical studies on some aspect of machine learning or data mining. For example, [Ciniselli et al. \(2021\)](#) examined the use of BERT models for code completion tools in an empirical study, finding and affirming that BERT models alone, without additional algorithmic cleverness, are good at predicting code at about the 60% accuracy level. This finding is congruent with an empirical study done on GitHub Copilot, branded as an “AI pair programmer,” which found great variety in code suggestion accuracy based on language, but no more than 60% accuracy [N. Nguyen and Nadi \(2022\)](#). The authors in [Y. Huang and Zimmermann \(2021\)](#) present work similar to that done by [Svyatkovskiy et al. \(2021\)](#) in proposing a ranking system idea which merges both machine learning results and other prediction algorithms, such as abstract syntax tree mining based tools, to improve accuracy. And, with respect to abstract syntax tree mining itself, [P. T. Nguyen et al. \(2019\)](#) presents a novel graph-theoretic paradigm to extract program dependencies and change patterns at scale.

Finally, an analysis was done in [A. Rahman \(2018\)](#) which explored how code writing activities correlated with code comprehension activities, finding that the relationship is nuanced and complex, but that when code is difficult to comprehend edits are made at a much slower rate.

2.3.4 Curated Datasets. There were no papers which released a curated dataset for code writing activities as the primary contribution of their work. That

said, MSR has code mining challenges which are typically recorded and released to the community. This data can form the basis for some exploratory studies before authors collect their own custom data.

2.4 Topic: Code comprehension

Developers don't just spend time writing code and documentation. Rather, undertaking the task of code comprehension – and seeking out the resources to gain understanding – is a major component of a developer's activity. In [D. L. Z. X. A. E. H. X. Xia L. Bao and Li \(2018\)](#), the authors found that approximately 58% of a developer's time was spent on code comprehension activities. Code comprehension includes reading existing documentation, performing web searches, reading Q&A websites (e.g., Stack Overflow), and communicating with other developers.

2.4.1 Tools. In many ways, tools in the code comprehension space are very similar to the kinds of tools one will find in the code writing space. This is understandable – once a developer understands a code fragment, a developer will often choose to implement it if the fragment has utility. Consequently, many tools in this space are oriented around porting code from places where code comprehension happens (such as Stack Overflow, API/Library documentation sites, similar project's repositories, etc.) and into the code artifact the developer is working on.

Tools in the code comprehension space can be best thought of as lying in a plane with two axes. The first axis is usefulness (from very useful to almost useless), the second is breadth (from niche to all-encompassing). Most tools seem to lie within a zone around a line which starts at (niche, very useful) and terminates at (all encompassing, almost useless).

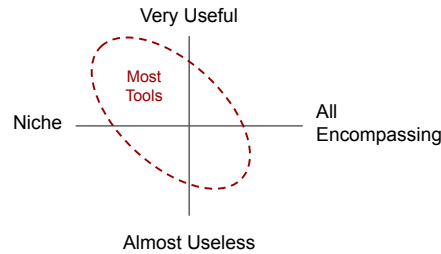


Figure 1. Tools in the code comprehension space

Perhaps the best example of this pattern comes from tools related to APIs and API comprehension. At one end of the spectrum, in the (niche, very useful) zone are tools like those presented in [H. Phan and Nguyen \(2018\)](#), which simply finds the fully qualified name of various API calls from snippets posted on sites like StackOverflow. Slightly further down the line is [H. Li et al. \(2018\)](#), which is a tool that uses natural language processing to identify various caveats and exceptions to common API usages. Yet further down the line is FOCUS [P. T. Nguyen et al. \(2019\)](#), which provides API recommendations based on mining and analysis of API usage in other open source systems deemed similar to the current project. Next in line is FaCoY [Kim et al. \(2018\)](#), a code to code search engine. And, at the far end of the spectrum, is [Eberhart and McMillan \(2021\)](#), which essentially seeks to eventually replace a developer colleague as a one-stop-shop for code understanding question-and-answer dialog. This work, while quite comprehensive, and trailblazing in the right direction, is not particularly functional yet.

2.4.2 Psychology / Social Science. Stack Overflow and similar Q&A sites dominate the code comprehension space where humans interact with each other. Consequently, several studies have been conducted to critically evaluate the quality of the developers providing the answers compared to Stack Overflow’s metrics,

evaluating the speed at which other users provide answers to questions, and analyzing fact vs opinion-based responses.

With respect to developer quality, [T. H. C. Y. T. S. Wang D. M. German and Hassan \(2021\)](#) succinctly answers that area in their self-explanatory paper, *“Is reputation on Stack Overflow always a good indicator for users’ expertise? No!”*. Nevertheless, developers are still asking and answering questions on these platforms. There has been particular attention paid to the speed at which answers are posted, with [T. H. C. S. Wang and Hassan \(2018\)](#) first doing a systemic analysis on some 46 factors related to the question itself, the accepted answer, the user posing the question, and the user answering with an accepted answer. Assessed on four Stack Overflow sites (Stack Overflow, Mathematics, Ask Ubuntu, and Superuser), the authors found that factors related to the user answering the question most strongly impacted the statistical likelihood of an answer being accepted. The key takeaway: a quickly provided answer of mediocre quality submitted by a frequently contributing user is much more likely to be accepted than a high quality answer provided by an expert user that infrequently contributes.

This dynamic is reaffirmed and explained in [Y. Lu and Li \(2020\)](#), *“Haste Makes Waste: An Empirical Study of Fast Answers in Stack Overflow”* which concluded that quickly provided answers aren’t always the best ones – measured in part by the amount of follow up required in the comments – despite being the popular answers. The authors attribute this to the gamification style of Stack Overflow to encourage participation and interaction. Consequently, aspects of the gamification may need to be tweaked to maximize answer quality. On the other hand, Stack Overflow is also a business which has an income stream from advertisement revenue. Therefore,

a business decision to maximize human participation on the site at the expense of good-but-not-optimal Q&A-gamification practices may be in play.

Other works in this space analyze the reliability of code fragments provided in Stack Overflow answers. In [A. R. H. R. T. Zhang G. Upadhyaya and Kim \(2018\)](#) the authors find that more than 30% of all Stack Overflow posts contain API usage violations which can compromise the integrity of software which incorporates them. And in [S. Mondal and Roy \(2021\)](#), the authors survey “rollback edits” – when a user posts an answer, then edits the answer, then edits or rolls back the edit. They find that these kinds of back-and-forth editing dynamics lead to inconsistencies and more than 80% of professional developers assess that they lead to detrimental post quality. Moreover, a developer’s understanding of whether answer content on Stack Overflow is applicable to them often requires additional context than what was initially provided, with [A. Galappaththi and Treude \(2022\)](#) finding that almost half of the Stack Overflow threads in their empirical study eventually included clarifying context that was not in the initial question.

With the relative unreliability of code snippets posted online in mind, particularly those posted in new threads anchored by novel questions, perhaps it’s not surprising that some developers try to find work-arounds from API calls posted on sites like Stack Overflow altogether, which is what was explored in [Lamothe and Shang \(2020\)](#). This work, complemented by similar findings in [M. A. Al Alamin and Iqbal \(2021\)](#) which studied developer discussions in software development challenges, found that the most common types of questions which related to low post quality revolved around “customization” and “dynamic event handling.” This suggests that APIs which allow developers to do “customization” and “dynamic event handling” easily are more likely to be adopted.

Stack Overflow, while a major component of human interaction with code comprehensibility, isn't the only option for Q&A / code comprehension sites. Other sources of help with comprehension exist and, like with any other aspect of human interaction on the web, search engines loom large. In [M. M. Rahman et al. \(2018\)](#), the authors created a machine learning classifier to automatically identify which real-world queries from several hundred developers were code-based Google searches vs non-code related queries. The authors found that code related searches required more effort (search term modification, multiple result clicks, etc.) than non-code search. Further work in [Hora \(2021\)](#) found that most search queries by developers that aren't copy-pasting code or error messages are typically short (three words or less), start with a limited set of key words – often the name of the framework, language, or platform (e.g., Python, Android, etc.), and omit functional words. They found that Stack Overflow results dominate, but YouTube is also a relevant source nowadays. This may indicate that different developers prefer comprehension via different mediums, or that different problem types lend themselves to be addressed via different mediums. More research in this space is needed.

2.4.3 Analysis / Broader Studies. In an effort to create better tools to support code understanding, several foundational machine learning papers have been presented. These are precursors to more advanced machine learning models which today can predict code. Rather, in an attempt to solve an easier problem of simple classification, there were two noteworthy papers in 2020 which focused on sentiment analysis in the software engineering domain [E. Biswas and Vijay-Shanker \(2020\)](#); [F. T. S. A. H. D. L. T. Zhang B. Xu and Jiang \(2020\)](#), particularly using BERT. More modern work has expanded into using these BERT-based models as pre-trained bases for general source code understanding models which can then understand and

write code. This work is bleeding edge and not quite ready for day-to-day usage as an engine for any sort of broadly useful tool. These prior works do, however, form the building blocks for tools which, if successful, will likely have broad utility. For example, if one can train a neural network model to do sentiment analysis well on text that includes code snippets or technical jargon, that base model can become a core of a new model via transfer learning to do novel tasks.

Meanwhile, humans still rule the day, and so evaluating techniques to increase readability – like in [J. Johnson and Sharif \(2019\)](#), which evaluated different code writing rules to increase comprehensibility with some 275 developers – as well as a system literature review on comprehensibility in [D. Oliveira and Castor \(2020\)](#) are key. Particularly in the [D. Oliveira and Castor \(2020\)](#) paper, which found that assessing code readability is a highly subjective exercise.

Two other studies in the Stack Overflow arena were focused on specific types of application development. For example, [G. L. Scoccia and Autili \(2021\)](#) did topic mining of Stack Overflow posts related to desktop web applications, and found that (1) build and deployment processes were some of the most common issues developers faced; (2) reuse of existing libraries in the desktop app development space is cumbersome; and (3) debugging of native API problems is tough; all of which tracks with [M. A. Al Alamin and Iqbal \(2021\)](#)'s finding that API customization and dynamic event handling are common roadblocks in development. Similarly, [Abdellatif, Costa, Badran, Abdalkareem, and Shihab \(2020\)](#) did a topic analysis mining of Stack Overflow posts related to chatbot development, finding that most posts revolve around chatbot model training and integration. Given that the recurring themes of API customization, dynamic event handling, and application deployment + integration categorize the most vexing issues of most development, further work

in this area may include an analysis of projects where these aren't the most common issues, and seeing if some categorization of these projects can be made. Moreover, understanding how ASC projects relate to these kinds of common issues in other open source or industry projects remains unstudied.

2.4.4 Curated Datasets. Three datasets in this area have been published since 2018. One dataset, reflective of the papers published in 2018 relative to ML and sentiment analysis in this space, is a collection of 4,800 Stack Overflow questions, answers, and comments which were hand labeled with emotions [Novielli, Calefato, and Lanubile \(2018\)](#). Another two datasets, which provided similar manual labeling of sentiments, was provided in 2018 by [Lin, Zampetti, Oliveto, et al. \(2018\)](#). Additionally, [B. Kou and Zhang \(2022\)](#) provides a dataset of over 2,200 popular Stack Overflow posts with manually provided summaries, which can be used as training data in a ML context to summarize discussion.

Lastly, [Baltes, Dumani, Treude, and Diehl \(2018\)](#) presents SOTorrent which is a tool that facilitates the ease of mining of Stack Overflow data, as well as incorporates various similarity metrics and data which can facilitate time series analysis useful to researchers doing mining on the website.

2.5 Topic: Smells and quality

Code smells are symptoms of poor implementation choices applied during software evolution [Pecorelli, Palomba, Khomh, and De Lucia \(2020\)](#). While smells were once intuitively identified by experienced developers, various definitions have since been developed and standardized. We also include in this topic the closely related notion of measuring formal smells by also including code quality, and the various metrics to assess code quality, as part of our discussion.

2.5.1 Tools. There are two flavors of papers in the intersection of “tools” and “smells + code quality metrics”: (1) Where code quality (and code quality metrics) are used to further a goal within a tool; or (2) tools used to evaluate and visualize the quality of the code itself, or to identify particular smells.

As an example of a tool used to further a goal, in [Nayrolles and Hamou-Lhadj \(2018\)](#) the authors use code quality metrics (with clone detection) in their tool, CLEVER, which does just-in-time fault prevention in large industrial projects. In this same vein, [Trockman et al. \(2018\)](#) reevaluates a study ([Scalabrino et al. \(2017\)](#)) which assesses the understandability of written code from a human perspective – in much the same way an algorithm might classify a book as grade school level understandability or college level understandability – using differing kinds of code metrics in their tool. Or, [A. Utture and Palsberg \(2022\)](#) which uses static analysis, metric calculation, and graph theory to provide inputs to a machine learning model which is subsequently used to identify and remove null pointer bugs.

An example of a tool which evaluates code is [Sharma and Kessentini \(2021a\)](#), which presented a platform called QScored that identifies and visualizes various quality metrics of an overall repository or project. The motivation being to have better access to metrics for project comparison than simply the number of stars or

issues that a GitHub based repository might have. Another tool in this vein of code evaluation is the machine learning model developed by [Pecorelli et al. \(2020\)](#), which used both code metrics and developer perception to rank the smellyness of design issues.

2.5.2 Psychology / Social Science. “The mind is a powerful place.” So claims [M. Wyrich and Wagner \(2021\)](#) in the title of their seminal paper on how code comprehensibility metrics intersect with code understanding. In their work, the authors undertook a double-blind study which evaluated the extent of which a displayed code comprehensibility metric impacted a developer’s subjective belief that the code was, in fact, comprehensible. The authors found that, regardless of the actual code comprehensibility, developer belief of code comprehensibility was incredibly influenced by a displayed score.

This finding is in congruence with earlier work done in [J. Pantiuchina and Bavota \(2018\)](#). In this work, the authors empirically evaluate code quality metrics’ claim to identify smells as valid. They do so by evaluating whether developer submitted commits which specifically claim to improve one of four attributes in the commit message (cohesion, coupling, readability, complexity) are truly improved by various metrics. It turns out that, despite code being improved from the developers perspective, most code quality metrics fail to capture improvement.

2.5.3 Analysis / Broader Studies. While code quality metrics may not be as reliable as a human software developer’s expert assessment in a general context (despite popular belief in the authority of formally defined metrics), they do seem to have utility in the narrow area of testing and test smells. In [A. Z. M. B. D. Spadini F. Palomba and Bacchelli \(2018\)](#) the authors investigate the relationship between test smells – that is, when the testing code itself is smelly – and code quality of the software

being tested. Specifically, the authors analyzed 221 releases of 10 software systems and compared six types of smells with various kinds of software quality metrics. They found that smelly tests were correlated with lower quality, more defect-prone software.

Further work in [G. Grano and Gall \(2020\)](#) investigated a similar question as in [M. Wyrich and Wagner \(2021\)](#) (namely, are code quality metrics congruent with developer perceptions?) and found that, in the narrow area of unit test code quality, metrics are a necessary but not sufficient condition for identifying problematic code. Consequently, the current state of the literature suggests that code quality metrics have only limited utility compared with expert developer assessment.

Further muddying the water is an empirical study by [D. Kavalier and Filkov \(2019\)](#) which found that not only do the code quality metrics themselves matter, but so does the idiosyncratic quality assurance tools which are used to automatically assess code as part of CI, as well as the order in the pipeline in which they are introduced.

Still, code smells and quality metrics shouldn't be written off entirely. In [P. Gnoyke and Krüger \(2021\)](#), the authors identified how architecture smells can evolve over time. In particular, the authors note that if quality assurance is postponed or abandoned for a system then smells and their correlated issue-proneness can dramatically increase.

Consequentially, the state of the literature appears to point to a paradox: smells and code quality metrics being close to meaningless in terms of production quality – that is, will the code break or not – at an individual code fragment level (with some caveats for test code), yet vitally important for the overall health of the project.

2.5.4 Curated Datasets. More work is needed to tease out the relationship between developer perception, metric utility at the code fragment level

(that is, code about the length of a single method implementation), and overall impact on the issue-proneness of the project as a whole. To assist in this future work [Sharma and Kessentini \(2021a\)](#) submits, via the QScored platform discussed in their parallel work [Sharma and Kessentini \(2021b\)](#), a large dataset of quality metrics and code scores which can subsequently be mined for future insights.

2.6 Topic: Whole project aspects

While other topics in this survey narrow down on specific aspects of the software development process, this topic covers impacts to an entire software repository or project as a whole, or empirical analysis across multiple repositories.

2.6.1 Tools. There are relatively few tools in this space which don't have a primary home in another topic. Tools which are applicable to this topic typically involve mining an entire project's repository (or multiple repositories) in order to address a legal, security, or privacy issue. For example, [R. Feng and Zhang \(2022\)](#) presents a tool for automated detection of passwords in public repositories, which they deployed on GitHub and found that over 60 thousand public repositories had passwords embedded in publicly available code. Another paper, [X. Xu and Liu \(2021\)](#), presented a tool used to find software reuse in public repositories which violates licensing agreements, with a 97% accuracy rate.

These types of tools are dependent on widespread mining ability. As the number of repositories grows, and as the size of each individual repository also contains more artifacts, the ability to mine and crawl becomes a bigger challenge. The authors of [F. Heseding and Döllner \(2022\)](#) confront this challenge with a command line tool and library called *pyrepositoryminer* built for multi-threaded repository mining, achieving 15x speedup against other existing off-the-shelf, generic web-based mining tools typically used for repository mining. A different tool, LAGOON [S. Dey and Woods \(2022\)](#), does similar things as *pyrepositoryminer*, but focuses more on the exploration and visualization of sociotechnical data of open source software projects from a variety of sources (code repositories, mailing lists, project websites, etc.) What seems to be missing in this space is a clear standard for the dissemination and sharing

of technosocio repository data, as each tool uses its own idiosyncratic data formats for saving and sharing insights gleaned from mining.

2.6.2 Psychology / Social Science. Four themes are emergent in this area:

1. Developer mindset or attitude to project-wide characteristics or issues.
2. Analysis on the motivations, behaviors, and characteristics of contributors to projects, particularly open source projects.
3. Analysis on the motivations, behaviors, and characteristics of financial backers of projects, particularly open source projects.
4. Legal policies and licensing issues, and their effects on users, developers, and projects.

1. *Developer mindset* papers include works like [Hadar et al. \(2018\)](#), which explored how engineers approached privacy. The authors found that most developers used the vocabulary of data security to inform privacy concerns, and as such limits the perspective of top privacy issues to mostly external threats. Other work analyzing developer mindset includes [T. Sedano and Péraire \(2019\)](#), which explored how developers and other stakeholders (like project managers) conceptualized, created, and dealt with backlog of technical to-dos – things like feature requests, bug fixes, and known technical debt servicing. The authors found that existing theoretical frameworks for backlog in other business domains didn't really apply to software backlog, but by having team members simply thoughtfully reflect on the items in a backlog as a group, collective sense-making allowed for more effective prioritization of tasks leading to greater productivity. Still other work like [S. Biswas and Rajan \(2022\)](#)

presented an empirical evaluation of data science pipelines and how data scientists conceptualize the work to be done in a data science project. They present two related conceptual/theoretical models – data science in the large, and data science in the small – for how data science projects are conceptualized in practice by practitioners.

2. *Analysis on the motivations, behaviors, and characteristics of contributors to projects* includes work like [H. Fang and Vasilescu \(2022\)](#), which studied the efficacy of social media – particularly Twitter – in advertising open source projects on GitHub. Among other things, the authors found that tweets did impact repos by increasing the number of people starring a GitHub project, and a modest link of new contributors to these projects. Other work focuses on analyzing the geographic history and diversity of contributors to public code bases, such as [Rossi and Zacchiroli \(2022\)](#). They found that, over the last 50 years, public code contributions have been dominated by North American and European developers, with contributions from other geographic areas like South America, Central Asia, and Africa slowly picking up starting around 1995 and increasing roughly linearly since that time. Today, non-North American and non-European contributors provide roughly 30% of all open source project commits. Notably, China provides very few contributions to the open source project ecosystem relative to its population.

Beyond where contributors are from and how they are incentivized to join projects, work has been done on collaboration and co-commit patterns of developers. One large scale study on some 200 thousand GitHub repositories was [Cohen and Consens \(2018\)](#), which found that the most active developers have tighter, more insular, and less collaborative networks than developers as a whole. This work was highly technical, and metrics were based on various graph-theoretic metrics like node connectivity (where a node is a developer, and an edge is placed between two

developers if they contributed to the same repository). Results are, frankly, difficult to intuitively interpret.

This led to work like [Lyulina and Jahanshahi \(2021\)](#), which presented a tool to visualize projects, developers, and their contributions via interactive graph. A challenge they faced is the sheer size of such networks, which required some sort of pruning for the interactive visualization to be digestible for a human being. Future work in this space includes thoughtful analysis for such filtering and how to motivate the utility of undertaking exploratory analysis via visualized interaction graphs.

3. Analysis on the motivations, behaviors, and characteristics of financial backers of projects, particularly open source projects.

Money makes the world go round, and that is also increasingly the case for open source projects. While software project financing in industry, government, and academia are usually self-explanatory as to where the money is coming from and why (self-explanatory at a high level, exact financing details can be quite complex), the same is not true of open source projects. The authors of [C. Overney and Vasilescu \(2020\)](#) explored 25,885 GitHub projects that asked for donations, out of a total of 77,934,441 repositories (0.04%). They found that popular projects (as measured by number of stars), mature projects, and projects with recent activity all were more likely to ask for donations. The authors also concluded that most donated funds are advertised to go to engineering efforts, but there is no systemic evidence that funding makes much of an impact on project activity levels.

Other work by [N. Shimada and Matsumoto \(2022\)](#) explores GitHub Sponsors, a program launched in 2019 which allows donors to fund specific developers that contribute to open source software projects. The authors found that developers who are sponsored are more active than non sponsored developers seeking sponsorship,

sponsored developers are typically top contributors to projects before getting a sponsorship, that roughly two-thirds of sponsors are developers themselves, and that sponsors and sponsorees are usually part of the same tight-knit networking clusters.

4. Legal policies and licensing issues, and their effects on users, developers, and projects.

Underlying software development are issues related to licensing and intellectual property. Developers are generally not attorneys, yet modifications to software licenses can have significant legal impacts. This phenomena was studied in [C. Vendome and Poshyvanyk \(2018\)](#), which empirically examined “licensing bugs,” finding that everything from laws and their interpretation, to the legal re-usability of seemingly open source code with conflicting licenses (or even no provided license), to jurisdictional issues all present complex and novel problems which both developers and attorneys have conflicting views on. Understanding just how prevalent these kinds of legal issues are was examined in [Golubev, Eliseeva, Povarov, and Bryksin \(2020\)](#), which empirically studied Java projects on GitHub, searched the repositories for code clones, and analyzed the original licenses of the source of the copied code and the embedding project. They found that up to 30% of projects involved code borrowing and about 9.4% contained copied code which could violate the licensing usage agreement of its source.

Beyond the intellectual property implications of code use and reuse in software projects, other empirical work on the non-discrimination policies of software artifacts is gaining traction. For example, work like [F. E. M. Tushev and Mahmoud \(2021\)](#) found that most non-discrimination policies are buried deep within “Terms of Service” documents (as opposed to a separate document, like many privacy polices), if there is a written policy at all. The policies that do exist are usually very brief and

boilerplate, and almost always have no written enforcement mechanism. Given real-world allegations and court findings of discriminatory behavior of apps or app users related to well known companies like Uber and AirBnB, not to mention concern about algorithmic fairness – particularly in machine learning / artificial intelligence algorithms – closer attention to written non-discrimination policies (or lack thereof) appears poised to be a dynamic and evolving field in the next few years.

2.6.3 Analysis / Broader Studies. There are four themes about software projects and their repositories that reoccur in the literature:

1. Taxonification of projects and their artifacts.
2. Qualification of projects and their artifacts.
3. Quantification of projects and their artifacts.
4. Analysis of how projects change over time.

1. *“What is software?”* This is the question asked by students in introductory programming classes everywhere which [Pfeiffer \(2020\)](#) sought to empirically answer by mining 23,715 GitHub repositories. They organized their findings into 19 different categories and assert that, far from software simply consisting of source code and perhaps an executable, software also contains scripts, configuration files, images, databases, documentation, licenses, and so forth. Some of their research questions include “Does a characteristic distribution of frequencies of artifact categories exist?” The author’s answer is no, but it is unlikely (less than 1%) that a repository contains more documentation artifacts than data artifacts (like images), and more data artifacts than source code artifacts. This work was assessed on 23,715 repositories covering a wide range of projects, and therefore future work in this area includes a

tightened focus on examining a narrow set of related projects. For example, asking the same questions about software artifacts, but restricting the evaluation set to ASC projects.

2. Quality of projects in research

Moving beyond individual artifacts are papers assessing attributes around the quality of repositories as a whole. Of particular note is [Hasabnis \(2022\)](#), which was a hackathon that resulted in GitRank, a tool to measure the quality of repositories. The motivating issue was an assertion that poor-quality repositories should be excluded from use in machine learning training data, and as such a tool needed to be developed to rank and compare such repositories when working at a large scale. This assertion that poor-quality repositories are problematic is a well founded one, as the paper “Is ‘Better Data’ Better than ‘Better Data Miners’?” [Agrawal and Menzies \(2018\)](#) found the answer to be “Yes.” The authors found that ML-based tools which do things like defect prediction performed better with much higher quality training data than simply improving or modifying the underlying ML+data mining model.

3. Quantification of projects in research

The quality of the data isn’t the only factor in research. Quantity matters, too, particularly when investigating human productivity in relation to GitHub data. The authors of “*Big Data = Big Insights? Operationalizing Brooks’ Law in a Massive GitHub Data Set*” [C. Gote and Scholtes \(2022\)](#) concluded that conflicting results in prior empirical work about Brooks’ law in open source software projects was primarily driven by poor data collection and aggregation pitfalls that occur when doing massive data analysis. (As a reminder, Brooks’ law asserts that adding developers to a project counterintuitively causes overall progress to slow down, not accelerate.)

Gote et al. found that, “Studies of collaborative software projects found evidence for a strong [...] effect for different team sizes, programming languages, and development phases. Other studies, however, found a positive linear or even super-linear relationship between the size of a team and the productivity of its members.” and produced a long list of citations of conflicting work. They found that differing methodologies when doing statistical analysis accounted for many of the perceived differences. Methodologies which, in their view, were sloppy. For example, neglecting to do proper stratified sampling.

The takeaway is that big data requires big caution when analyzing human interaction data in a repository mining context, and thus smaller and more curated datasets can often give clearer insights than larger, noisier ones. This echoes earlier work on the threats of aggregating software repository data in [M. P. Robillard and McIntosh \(2018\)](#), which identified and described common threats to big data analysis in a software mining context.

4. *Time series analysis of projects*

High quality, low quality; big data or small; there's an interest in evaluating how projects change over time. Themes in this area include papers which present new metrics, like [Benkoczi, Gaur, Hossain, and Khan \(2018\)](#), which presented a framework for identifying how changes in commit patterns impact dependencies and metrics like cohesion and coupling in various code modules over time. Other work, like [S. K. M. Tushev and Mahmoud \(2019\)](#) finds that linguistic features (for example, variable names and patterns) in a multi-developer code base changes quite a bit in the first era of a project's existence before mostly standardizing, usually around the creation of documentation, with slow but notable changes thereafter, with patterns which match human language changes in written speech in long-lived projects. And yet other work like [A. Ait and Cabot \(2022\)](#) analyzes the survival rate of GitHub projects, finding that open source projects are prone to death, with less than 50% surviving more than five years, and most projects are characterized by high bursts of activity followed by long periods of low/no active development or maintenance.

2.6.4 Curated Datasets. While there are several papers discussing data collection in repository mining, there are relatively few works which present curated datasets as the primary contribution. Two are relatively niche: [Zacchiroli \(2022\)](#), which presents a large dataset of open source licenses and variants, and [V. Tawosi and Sarro \(2022\)](#) which provides a dataset of open source projects that specifically and actively use a particular agile methodology in development.

One dataset of note is provided by [N. Riquet and Vanderose \(2022\)](#), which contains git data from an anonymous private company with over 100 repositories and some 120+ developers over the course of more than a decade. Given that software engineering and data mining research on data from private companies is

such a rarity, this dataset could help form a useful benchmark against open source assumptions. Unfortunately, the company required the data to be heavily anonymized (including the name of the company itself) as a condition of the data's release. Consequently, there are limitations to the usefulness of this data when used as a comparator benchmark.

2.7 Topic: Human and team dynamics

This section focuses on the human element of software development. Namely, how do humans relate to the engineering process and each other while engaging in development work.

2.7.1 Tools. Tools relating to human dynamics fall into one of the following three buckets:

1. Characterizing developer behavior or attributes from human-sourced information pools.
2. Characterizing software from human-sourced information pools.
3. Assisting the developer in the software engineering process.

As is a recurring theme throughout this literature survey, there are no tools developed specifically for ASC, but we do highlight open source-based and other noteworthy tools which may have some utility in the ASC field or spark ideas.

1. *Developer behavior* tools focus on helping a team lead or manager understand what's going on with individual contributors. For example, in [T. Dey and Mockus \(2021\)](#), the authors try to solve the problem of matching open problems / tasks which need finishing in an open source software setting with relevant developer expertise and skill set. The authors accomplish this by using machine learning embeddings of code. Namely, the authors take machine learning embeddings of open source project code which needs an additional maintainer, and compares this vector embedding with code written by various software developers. The idea is that developers who have worked on similar projects as the ones needing maintainers will have code embedding vectors which are likewise similar.

Another tool which may help managers and other team leads is EmoD [K. P. Neupane and Wang \(2019\)](#) which captures communication records and identifies the emotions therein. Paired with a visualization, the tool can alert team leads to possibly unhealthy emotions in online communication as developers collaborate with each other.

2. *Software behavior* tools pull information from human dynamics to inform the development process. For example, [L. Shi and Wang \(2020\)](#) is a machine learning tool which mined an extremely large set of developer chats to predict new feature requests from the text, creating an automated repository of ideas to possibly develop and implement. Other tools are more helpful in audit and compliance. For example, the natural language processing machine learning tool developed by [Hu et al. \(2021\)](#) identifies 26 predefined types of undesirable software behavior from user reviews of mobile applications that violate GooglePlay’s marketplace rules. This tool can help human auditors monitor for compliance for things like making sure games or apps which contain nudity are properly rated for adults. While advanced scientific computing tools are unlikely to contain nudity, the broader framework to automatically detect undesirable app behavior may be helpful.

3. *Assistants* to a developer are also in this space. While there is much more to be said in §2.9 on bots, tools include assistants like Robin [L. D. Silva and Moissakis \(2020\)](#) which is a “voice controlled virtual teammate.” Specifically, Robin is an Amazon Alexa or OKGoogle-like assistant for git and GitHub. Robin is able to, via oral command, do trivial git-related tasks, such as closing issues or pull requests. This tool was trained based on extensive data collection from real-world software teams asking each other favors in offices. As for its practical utility, well, it seems to still be in the early stages.

2.7.2 Psychology / Social Science. There is a lot of fun work to read relating to the psychology and social science of software development. There's even work done on cannabis usage among programmers! [M. Endres and Weimer \(2022\)](#) found that 35% of their sample of programmers had engaged in software development while high.

More generally, work can be sub-categorized into one of the following bins:

1. Leadership in software development.
2. Motivation for staying on a project.
3. How programmers organize and manage themselves to get work done.
4. How humans perceive and measure productivity, proficiency, effort, and efficiency.
5. How information and results are communicated with actors outside the team.

1. Leadership

Leadership research in software development is somewhat conflicted. While there is a lot of research done in business and management on skills and attributes of leaders, mapping those traits aren't always clear-cut in the software development world and vary depending on the organizational context. For example, in [E. Kalliamvakou and German \(2018\)](#) the authors conducted a mixed methods study to find what made a great manager of software engineers, deploying a survey to nearly 3,700 developers at Microsoft. They found that deep technical expertise wasn't important (although a minimum level of competence was needed). Rather, the people skills of providing support, coaching, direction, and motivation is what was most valued.

This contrasts somewhat with the work done by [E. Dias and Pinto \(2021\)](#) about the leadership attributes of lead maintainers of major open source projects (e.g., the Linux Kernel). Dias et al. found that deep technical expertise, along with quality communication, were the key attributes of excellent leaders in long-term open source projects.

Yet later work in [Gren and Ralph \(2022\)](#) explores leadership in agile software development, which often rejects a hierarchical organization structure with a clear leader in favor of a group of small teams working closely together (e.g., Spotify’s internal software development organizational structure). Gren et al. finds leadership is not a property of individual contributors but rather a property of individual teams, within which responsibility and leadership is dynamically shared by developers. The most effective and leadership-like teams demonstrated a strong sense of belonging for their members, an ability to navigate competing interests with other teams in pursuit of the larger organizational mission, and strong communication skills by all team members.

The key theme of software development leadership, it seems, is the ability to motivate and communicate, with technical expertise taking a very important but secondary role.

2. Motivation

Motivation is another key ingredient to successful software development. While we did not find motivation research specifically focused on paid software development roles, there are several studies done on motivation in the open source community, including [A. Barcomb and Fitzgerald \(2019\)](#); [Gerosa et al. \(2021\)](#); [H. S. Qiu and Vasilescu \(2019\)](#); [Y. Huang and Zimmermann \(2021\)](#). While the methodologies and foci of these studies vary, all point to an underlying importance of two factors in

developer motivation for contributing to these communities: (1) feeling like they belong to the community to which they contribute, which often includes recognition; (2) the belief that their contribution impacts the world for social good in some way. While these motivations are for the open source communities, applying them to a national laboratory context by having lab leaders reimagine their work as contributing to the social good may help with motivation.

3. Team organization

Team organization and human-centric studies on the development process is a well studied area. It covers things like pair-programming work in [Zieris and Prechelt \(2020\)](#) which found pairs with experiences with different libraries are more effective than pairs with homogeneous knowledge. Other interesting work looked at whether programmers work at night or on the weekend [M. Claes and Adams \(2018\)](#), finding paid programmers tend to work standard office hours but there is wide variability among developers generally, with open source software contributors tending to work at heterogeneous times. Yet more work is area specific, with [Kim et al. \(2018\)](#); [N. Nahar and Kästner \(2022\)](#) focusing on the specific issues of data science and machine learning collaboration challenge. They find that communication challenges among stakeholders and mis-matched expectations are the most common roadblocks in machine learning team success.

Other research identified characteristics of work productivity, such as [F. Sarker and Filkov \(2019\)](#). It found that as the frequency of developer commits increased, so did the negative sentimentality of their comments, which the authors suggest is due to increased stress as developer work-rate increases. More generally, [S. Datta and Sarkar \(2021\)](#) theorized that projects tend to start out with developers relying on each other more, but as projects mature, developers become more independent. This

leads to a situation where developers on a team in successful projects usually don't need much formal structure in the beginning of a project – they are all in frequent contact with each other and the problems are nebulous enough that formal process is useless – but as the project matures, formal contribution guidelines are helpful.

This brings us to [O. Elazhary and Zaidman \(2019\)](#), which studied whether nominal contribution guidelines matched the real-world process for over 50 GitHub projects. They found that in nearly 70% of these projects with explicit contribution guidelines the actual contribution processes materially differed.

That was in 2019. In 2020, COVID struck. [de Souza Santos and Ralph \(2022\)](#) found that the transition from in-person to remote teams was disruptive, suggesting that managers needed to do more to encourage team engagement and provide greater support for those with child care responsibilities. Other work by [C. Miller and Zimmermann \(2021\)](#) suggests that 74% of developers who moved to a work from home environment missed their colleagues, with over 50% saying that communication had become more difficult. How these statistics look now, in 2023, remains to be seen. In any case, leaders sought to motivate their teams. A common team motivator is gamification of a goal (e.g., whoever has the longest daily commit streak “wins”), but work by [L. Moldon and Wachs \(2021\)](#) suggests that doing so often leads to lower levels of productivity after the game ends. Understanding the best practices of how to lead and motivate remote or hybrid teams remains open work.

4. Perception of productivity

A variety of studies have examined what makes a developer appear to be productive, usually through surveys. This includes work such as [Rodriguez, Tanaka, and Kamei \(2018\)](#); [P. S. K. X. Xia Z. Wan and Lo \(2019\)](#), which found a diverse constellation of factors – time commitment, quality of code produced, etc. – impacted

the perception of a developer’s productivity. But more work is increasingly focused on what makes software development teams appear to be productive. Work such as [Ruvimova et al. \(2022\)](#) suggests that highly productive teams in a paid setting are typically made up of highly productive individuals with fluid roles. That is, an ability for the individual members of the team to quickly and interchangeably self select tasks which need to be done. However, more research needs to be done to determine how this finding applies across multiple domains, and whether it replicates in the advanced scientific computing setting.

5. Communication outside the team

Development teams typically communicate with two types of outside actors: other developers and users. In the latter case, work by [S. Hassan and Hassan \(2018\)](#) suggests that developers are more likely to respond to longer or more negative user reviews of their software than positive reviews: squeaky wheel gets the grease. But in the former case, of developers on a team talking about their work with other developers, papers like [Fang, Klug, Lamba, Herbsleb, and Vasilescu \(2020\)](#) have found that the nominal role of a developer on a team impacts how people view the messages. For example, developers perceived to be the team lead or repository owner have different types of engagement on Twitter when broadcasting their work than other members of a team. This broad trend of “the lead” being the key communicator in evangelizing work is reflected in other research like [Aniche et al. \(2018\)](#); [M. Papoutsoglou and Kapitsaki \(2021\)](#). This has relevance in the national laboratory context if one goal is to advertise the open source work done in the labs to other user groups: if you want to share your work, it’s important to get it out there on multiple channels, and have the person sharing (1) perceived to be a technical person, not a communications/HR person and (2) be perceived as the leader of the

team. There is a tension here, because we just saw in perceptions of team productivity and effective organization that many of the most productive teams are those with members that contribute and lead in fluid ways, as opposed to a strict hierarchy.

2.7.3 Analysis / Broader Studies. Broader studies in the human dynamics space have focused largely on bias [Chattopadhyay et al. \(2020\)](#); [J. M. Zhang and Harman \(2021\)](#), typically along gender lines [Hilderbrand et al. \(2020\)](#); [Lee and Carver \(2019\)](#); [N. Imtiaz and Murphy-Hill \(2019\)](#) or age [W. Kopec and Casati \(2018\)](#). The recurring takeaways seem to be that women are not as included in software development teams as their male counterparts, and that special attention should be paid to individuals belonging to a minority demographic among any sufficiently large group setting (e.g., older adults in mostly young groups).

Other work is a bit of a grab bag of topics. One study analyzed toxicity in comments made by developers in places like GitHub and compared it with toxic commentary online generally, finding that toxic communication between developers is often characterized as more demanding and arrogant but with fewer open insults than toxic conversation found elsewhere online. This presents challenges in any natural language processing work done to characterize emotions in developer discourse.

But perhaps most amusing of all is a study done by Google called, “Do Developers Discover New Tools on The Toilet?” in which internal teams advertised new tools to their colleagues by placing physical advertisements/newspapers in workplace restrooms. The technique was effective, and perhaps an option for in-person devops teams seeking to evangelize their tools in large workplaces like national laboratories.

2.7.4 Curated Datasets. There was only one paper which presented a curated dataset designed for use by other researchers, DISCO [K. M. Subash and Baysal \(2022\)](#), which is a dataset of discord chat conversations about software

engineering from four programming language communities (e.g., the official python discord server). While the main contribution of their work is the cleaned dataset, the tools they used to create the dataset are interesting and helpful in their own right for mining other discord servers.

2.8 Topic and cross cutting theme: Machine Learning

.....
A brief sidebar on ChatGPT and opinion on its societal implications.

As of this writing, ChatGPT has taken the world by a storm since its release in November 2022. It is the subject of countless news stories and has entered the public conscious. It is also increasingly being used to generate code.

ChatGPT – having not been published as an academic work before our cutoff date of October 2022 when undertaking the area exam which formed the basis for most of this review in this dissertation, or in the three venues of ICSE, ICSME, or MSR – is outside the context of this initial related work. Nevertheless, a short and high level overview is appropriate for this survey, given the nature of the model and its use later in this work.

The architecture of the ChatGPT model isn't particularly complex once one is able to understand the fundamental building blocks of modern deep learning. Namely, transformers, attention, word tokenization, embeddings, transfer learning, and sequence-to-sequence generative models. The underlying generative pre-trained transformer (GPT) model also utilizes reward modeling and reinforcement learning paradigms as part of its training. All of these concepts are well documented and are readily accessible to a machine learning practitioner in formats outside of academic literature, like blog posts and (admittedly advanced) online tutorials.

What separates ChatGPT is not any major conceptual novelty or clever algorithm not already in the literature. Rather, the major driver underlying ChatGPT is the sheer scale of the number of parameters in the model – in the billions – and the terrabytes of training data used. In many ways, the primary limitation of another entity replicating ChatGPT isn't in the software, but rather in the hardware infrastructure needed to train the massive model. OpenAI, the organization behind ChatGPT, converted from a non-profit to a “capped profit” business in 2019 for the purpose of gaining enough investment capital to actually buy the hardware needed to train models.

In some ways, this is a concerning development. Most modern technological innovations related to information, to date, have resulted in long-term benefit for the common person and cheapening ownership costs over time. For example, printers are now ubiquitous, relatively cheap, and there are several companies which manufacture printers in a competitive marketplace environment. Printers are democratized. But large language models, like ChatGPT, will require hardware investment in the billions of dollars and highly specialized software development expertise for meaningful competitors to emerge. In practice, only a handful of players – tech titans and governments – will have the resources to duplicate the feat.

It's also important to note that the key underlying model driving ChatGPT, a model called GPT3, has not been publicly released. While GPT2 has been released as an open source product for anyone to download, GPT3 – built on top of GPT2 and described in various releases

by OpenAI – has not been made available for download. OpenAI has made GPT3 available for public consumption by developers only through API calls (which can easily be made monetizable as a subscription service). At a high level, ChatGPT is essentially a light wrapper for GPT3, which OpenAI has made more user friendly to non-software developers by adding a website interface. It is this website interface which the general public thinks of as ChatGPT.

By treating the language model as a software-as-a-service, OpenAI (and other tech players) have set the precedent that consumers of machine learning models are not owners of the model. Depending on your politics, this may be concerning. One can stream music from a service like Spotify or Apple Music, and many find that to be very convenient. But, at the end of the day, one can also purchase a song and own it. Many software products are the same way – there is usually an ownership alternative to software-as-a-service product. At least at the enterprise/business-to-business level.

But, given the massive capital required to create a machine learning model of this scale, it is likely that only tech titans and governments will have the resources to duplicate ChatGPT. This is concerning, as the barrier to entry for potential competitors will grow increasingly higher in the near to mid term. This will result in a situation where only a few select entities will have control over some of the most powerful and generalized machine learning models, or the realistic ability to improve on them, thus exacerbating the problem by increasing the barrier to entry for new actors.

A potential remedy to this problem, assuming there is societal agreement that artificial intelligence products should be democratized, is to treat these artificial intelligence products in much the same way we do prescription drugs. After a period of time of exclusive use for monetization, require companies with monetized machine learning algorithms to release the source code and data for the underlying trained model as a “generic” machine learning algorithm for anyone else to use.

In the meanwhile, I take heart that ChatGPT hasn’t fully taken over the world of machine learning as it relates to software development. It means that the remainder of this area exam chapter remains relevant to any entity creating machine learning based tools for software development without a billion dollars of capital to spend.

.....
Machine learning (ML), in the context of software engineering and repository mining, is used primarily as the underlying engine for tools that aids practitioners in some aspect of software development. Almost all ML work in this space is grounded in a supervised learning context.

Within this larger context there are two flavors of machine learning papers which are published in ICSE, ICSME, and MSR. One flavor of paper reports on machine learning models as a component of a tool. These machine learning-based tools are a cross cutting theme, featuring in topics ranging from bug detection to code suggestions.

The other flavor of paper focuses on some aspect of the machine learning development process itself. That is, the authors of ML development papers focus on an aspect of the training or model design as their core contribution to the community,

often in the pursuit of a larger goal to make a useful machine learning-based tool. For example, the task of automatically disambiguating natural language from technical jargon from code before ingestion by a language-based ML model, which tends to improve results.

This section will primarily focus on papers in this input → input transformation → model / algorithm → output transformation → final output pipeline, highlighting examples of various tools along the way. This pipeline, and the subcategories of papers in each pipeline, is visualized in Figure 2.

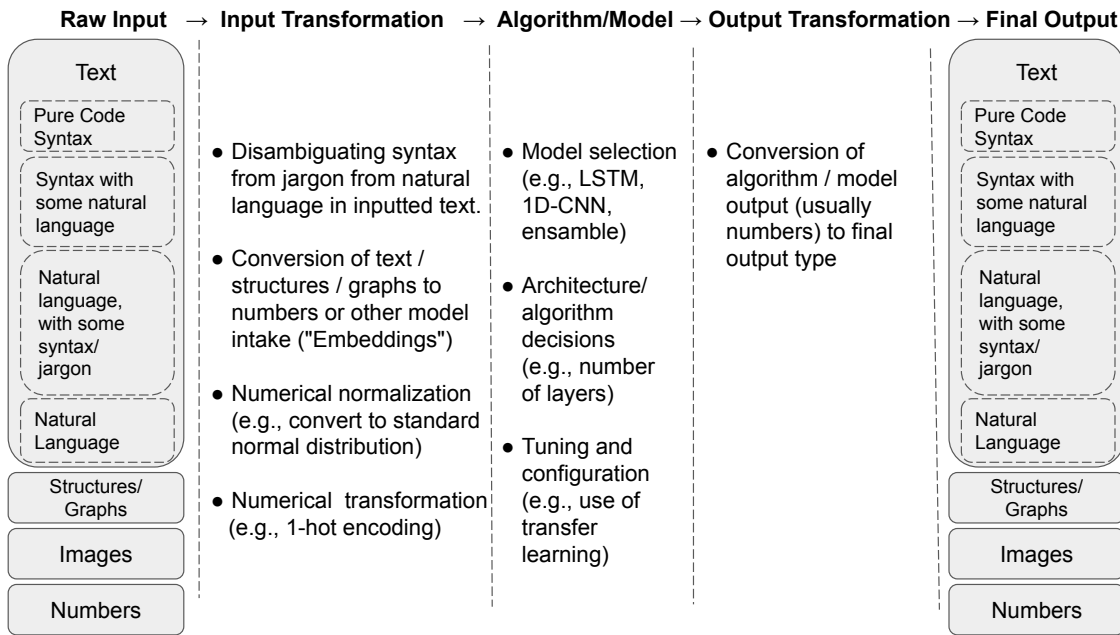


Figure 2. Components of a typical pipeline in an ML-based tool.

2.8.1 Raw Inputs and Final Outputs. It is important to note the different types of inputs and outputs typical in software engineering machine learning models. Namely, models can ingest and output raw numbers, images, structured data – such as graphs, and text. Within the category of text is a spectrum of possible

input, ranging from pure code, to a mix of natural language and code intertwined with technical jargon (e.g., documentation), to non-code natural language.

Examples of software development tools which use images include work like [N. Cooper and Poshyvanyk \(2021\)](#), which presents a tool that uses computer vision and image-based machine learning techniques to identify and consolidate duplicate video-based bug reports in a gaming development context from user submissions.

It's important to note that many output types are actually the transformed result of a classifier. For example, there are many tools for bug identification and most provide a text summarization. But often that summarization is linked to a large existing bank of pre-identified bug types. That is, code is transformed and ingested to a machine learning model, the model produces a classification (e.g., "bug type number 20"), which is then linked to a particular output message.

We provide a full table of the raw input type and the model output type before final transformation of nearly all machine learning model driven tools discussed in this report in [Table 27](#), which is located in [Appendix A.19](#).

Finally, it's important to note that software development tools which use numbers as input are generally uninteresting to the community, from a machine learning development improvement perspective, as numerical manipulation as input is a well studied topic in machine learning generally. Graph-based machine learning as applied to software engineering is very new and papers are still being produced as of this writing. Yet results are promising, particularly when applied to the repository organizational context. That is, including the mapping of artifact relationships within a repository as input to a machine learning model. Which brings us to the next section of how inputs are actually transformed.

2.8.2 Input Transformation and Output Transformation. Machine learning models which ingest software engineering artifacts as input have been around for nearly as long as there has been machine learning. The question is how to do so, as deep learning models take in numbers, not source code, and are optimized over a loss function. The conversion from a software engineering artifact to something any given deep learning model can consume is known as an embedding. This embedding is almost always a vector, although it could be a tensor or matrix.

It's important to note that a variety of software artifacts can be embedded. [Theeten, Vandeputte, and Van Cutsem \(2019\)](#) presented an embedding called Import2Vec, which transformed entire software libraries across different programming languages to vectors. Other work, like CC2Vec by [T. Hoang and Lawall \(2020\)](#) embedded code change distributions as a vector, finding that their code change embedding had utility in log generation and just-in-time bug patch prediction – that is, improving predicting of bugs and possible fixes. Yet other types of embeddings include code itself, like Code2Vec and improvements on it [Compton, Frank, Patros, and Koay \(2020\)](#), entire repositories like Repo2Vec [M. O. F. Rokon and Faloutsos \(2021\)](#), and graphs which map relations among artifacts within repositories in GraphCode2Vec in [Ma et al. \(2022\)](#).

Almost all of these *XtoVec* papers rely on text as the initial input to be vectorized. In general, text which is code does not play well in traditional natural language processing paradigms. Yet most software engineering artifacts are a mix of code, technical jargon, and natural language. It is a non-trivial problem to disambiguate among the three, and then to process the code and jargon once they've been separated.

Consequently, there has been several years of work done to answer these questions. [Mantyla, Calefato, and Claes \(2018\)](#) was one of the first packages which automatically pre-processed text to simply disambiguate whether textual information was natural language or not. Other work like [Efstathiou, Chatzilenas, and Spinellis \(2018\)](#) release a modified Word2Vec embedding trained on StackOverflow data so that ML practitioners had a natural language library which could actually handle technical jargon. Follow up work by [Biswas, Vijay-Shanker, and Pollock \(2019\)](#) analyzed different approaches to improve these types of libraries, finding that improving “off the shelf” natural language embeddings can easily be improved for the software engineering context simply by oversampling of jargon-infused data sources. This approach, however, requires that there is a sufficient diversity of data to oversample from. If the jargon one wants to include is related to a highly popular topic – say, general android application development, there’s probably enough on StackOverflow to mine from. But if the topic is even more niche or even confidential, as is the case for highly technical projects at many national laboratories, this approach may suffer.

Subsequent work in [Efstathiou and Spinellis \(2019\)](#) focused on how to integrate natural language which is found in code itself, such as inline comments, with the surrounding code. The primary contribution was six language-specific pretrained distributed code representation models for Python, Java, PHP, C, C++, and C#. This work was followed up by [P. P. Pârachi and Barr \(2020\)](#) which submitted their techniques to automatically and simultaneously tag natural vs programming languages. This was an improvement of over 20%. Similarly, [M. Osama and Ibrahim \(2020\)](#) submitted a disambiguation tool utilizing a scoring based technique trained on highly intertwined technical-code-language documents, namely software requirements documents written in natural language made for developers. Use of these two

2020 contributions is the current best approach for disambiguating code vs natural language in machine learning models in which the output is classification. That said, for situations where the output is a sequence, the sequence-to-sequence pretrained source code ML model presented by [Niu et al. \(2022\)](#) is a novel approach, but the contribution is “research-ware” and still needs significant programming support for general purpose use.

All of the above papers present software contributions to a machine learning pipeline. But other work in this space examines existing tools to determine the best fit for code processing. For example, [M. Jimenez and Papadakis \(2018a\)](#) found that among the most common off-the-shelf word tokenizers, an approach known as Modified Kneser-Ney works best for code analysis, when balancing against reasonable computation time.

Other work wanted to validate what certain ML paradigms asserted was important vs human perception, with [A. Marques and Murphy \(2020\)](#) having 20 human software developers manually highlight the important parts of key artifacts. The authors found that an approach called frame semantics most closely matched the consensus human highlights. More recent work, which incorporates attention, includes [Wan et al. \(2022\)](#), which found pretrained language embedding models could induce certain code syntax trees. This suggests that future embedding models should not only include raw text but a separate preprocessing step that extracts the code’s syntax tree to help provide more accurate results.

Just as important as input encoding, however, is output encoding. Examples of output encodings in a classification context include one-hot encoding, scaled label encoding for ordinal values, and so forth. Work by [Gong and Chen \(2022\)](#), using software performance learning systems as the underlying models, studied 105 output

type encoding configurations. That is, the authors varied output encoding types across a number of different classification models. They found that there are tradeoffs between accuracy and training time based on output encoding, with one hot encoding usually being the most accurate but taking the longest for the model to converge. Scaled label training, on the other hand, caused neural network model training to converge more quickly but had lower overall accuracy. The takeaway is that for classification tasks with code or jargon as input, one should use scaled labels for debugging the model and ballpark parameter tuning before switching to one hot encoding for the final training of the model.

2.8.3 Papers relating to an Algorithm, Model Architecture, or Tuning. Hyperparameter tuning is key to any neural network architecture, and most papers on it are applicable to models focused on software engineering applications. However, there is one paper, [M. Jimenez and Papadakis \(2018b\)](#), which reports on an automated tuning approach for word tokenization based on how natural source code is. That is, if a source code file contains lots of natural language comments, the tokenizer will treat the source code as more of a regular natural language file. And vice versa: if the source code file has strictly code syntax with little natural language, the tool will apply tokenization based on code rules. This 2018 tool, however, has mostly been made obsolete with general advances in embedding and network design (e.g., attention).

With the quick obsolescence of some deep learning techniques, it's important to verify that a common base model for code ingestion, CodeBERT, is accurate for a wide variety of tasks. CodeBERT is a transformer based machine learning model which can ingest a mix of code and natural language for a variety of classification and generative tasks. CodeBERT used natural language code search and code documentation

generation as example use cases when it was initially introduced. Work by [X. Zhou and Lo \(2021\)](#) independently verified that CodeBERT was useful beyond these two narrow use cases, empirically finding that CodeBERT is generalizable far beyond its initial out-of-the-box training data.

While CodeBERT is certainly a specific machine learning tool targeted to the software engineering domain, the majority of machine learning model architecture for software engineering tools is not specialized for software engineering per-se. Most models, given different training data and input embeddings, would work just as well on other domain problems if properly tuned. That said, work like [A. B. A. LeClair and McMillan \(2021\)](#) suggests that ensemble models work the best for models with source code as input, with a performance boost of about 15% above singular model approaches. Further work, like [Majumder, Balaji, Brey, Fu, and Menzies \(2018\)](#), suggests that non neural network models, like support vector machines, can preform just as well as deep learning but with significantly quicker training times.

Flipping the script a bit, software repository mining can also help machine learning development. Work like [G. Nguyen and Rajan \(2022\)](#) notes that finding the exact neural architecture for a problem is time consuming, requiring lots of tuning and tweaking with tools like Auto-Keras. Their paper introduced a tool, Manas, which takes a shortcut. Namely, the authors first searched and cataloged thousands of machine learning models in GitHub, and Manas uses the catalog to suggest machine learning architectures based on models that are already in use. Then, given a base architecture which is more likely to be closer to a working solution (for example, in image classification), significant time is saved.

Other work attempts to not just select a model, but to automate the entire ML process, from preprocessing steps to final output. [Saha et al. \(2022\)](#) is a tool

which takes in a high level human description of a machine learning pipeline and then predicts a plausible set of machine learning components in code which fit the description. Automatic architecture generation, for both ML and non-ML pipelines, while outside the scope of this review, is likely to gain more traction in the coming years, and is something to consider incorporating as a tool in the advance scientific computing developer’s toolkit.

2.8.3.1 Curated Datasets. All of these pipelines and tools for machine learning are useless without quality training data, however. While there is only one curated dataset for machine learning in the software engineering space – a time series dataset for open source software evolution [Sousa, Bigonha, Ferreira, and Franco \(2022\)](#) – there is plenty of work on effective mining practices to accrue training data in the first place. For example, [Sun, Li, Liu, Du, and Li \(2022\)](#) suggests a particular framework which includes a variety of rule based filters to clean Stack Overflow data will generally result in more accurate models which are trained on that data.

Other work like [Ahmed and Devanbu \(2022\)](#) suggests that, perhaps counter-intuitively, machine learning models which support multiple programming languages as text input are more robust than monolingual models, so long as the training data from the diverse languages all do roughly the same thing. For example, a model trained with “Hello World” in Java, Python, and C/C++ is more robust than just a single instance of “Hello World” in one language alone.

2.9 Cross cutting theme: Bots

One of the cross cutting themes of the software engineering experience is the growing reliance on various bots. In §2.7 we discussed Robin, a voice controlled “virtual teammate” for software development teams [B. da Silva and Sereesathien \(2020\)](#). As of this writing, ChatGPT is making headlines around the world for its human-like text in response to queries.

The development of these bots, whether geared for general purpose language interaction like ChatGPT or specialized bots/automations for software development like Robin or GitHub’s Copilot, is non trivial and often clunky. Challenges in chatbot development, in particular, were analyzed in [Abdellatif et al. \(2020\)](#) which found that machine learning training of chatbots in the software development space was particularly difficult due to the limited data sets available and mix of code, jargon, and non-jargon natural language in many training sets.

To that end, [E. Koshchenko and Kovalenko \(2022\)](#) suggests bots designed for software development incorporate a multimodal approach. That is, if the bot’s primary feature is to provide a recommendation to the developer to do X, the training should be sourced from a variety of collaboration platforms such as GitHub and Slack, as just focusing on one channel tends to lead to overfitting.

With these challenges in mind, many bots along the full spectrum of software engineering tasks have been developed. Notable are bots that write sample code [Wen, Aghajani, et al. \(2021\)](#), or that provide refactoring suggestions and other edits at pull request / code review time. These include well known bots like GitHub’s Copilot, which tends to do better with Java projects than many other languages [N. Nguyen and Nadi \(2022\)](#). But there’s a wider question about how useful bots actually are. By one measure, the answer is, “somewhat,” with [M. Wessel and Gerosa \(2020\)](#) finding

that the existence of a code review bot on open source software projects increases the number of merged pull requests but decreases communication among developers. That work was followed up two years later in [Wessel et al. \(2022\)](#) where 22 different design strategies and recommendations were presented to make bots better mediators in the human-pull request dynamic. To oversimplify, Wessel et al. suggests that bots should be partitioned into a separate space discrete from human-to-human activity, and also suggest that an overarching bot coordinator – that is, a bot which manages all the other bots – might be helpful.

Of especially great interest to the community is the question of determining if an action has been done by a human or a bot, with tools like BotHunter [A. Abdellatif and Shihab \(2022\)](#), BoDeGha [Chidambaram and Mazrae \(2022\)](#), and earlier tools like BIMAN [T. Dey et al. \(2020\)](#) available for use.

The potential utility of bots for the advanced scientific computing space is large. National laboratories are on budgets, but bots are only used minimally. If tools like bots can add meaningful value to a developer’s daily work, there can be impressive gains made. Effectively designing, evangelizing, and incorporating useful bots into the daily habits of developers is an area where ASC can really improve.

2.10 Cross cutting theme: Venues

As we examined these papers, we identified if there was a specific venue in which the research paper was based in. Namely, papers with data from/geared towards/analysis of open source projects, projects in start ups, projects in established industry players, and projects developed in academia.

To the best of our knowledge, there has not been a single paper published on software engineering specifically focused on how the development process is unique in the advanced scientific computing community of US National Laboratories. As we surveyed papers for this literature review, we found hundreds of papers focused on open source projects – papers which may have incidentally included a few open source ASC repositories in their corpus of evaluated projects, but ASC was not the focus of the study. We only found 27 papers focused on projects or repositories in industry, two related to start ups in some fashion, and only one from academia from the major software engineering research conferences.

The one research paper from academia about software engineering was not immediately applicable to the advanced scientific community: it was about the teaching of software tools to students in an academic context.

In the case of research related to start ups, much is not totally relevant to the ASC space: start ups are usually under enormous financial and time pressure to bring their product to market. The two papers in this space, [Gralha, Damian, Wasserman, Goulão, and Araújo \(2018\)](#); [T. Besker and Bosch \(2018\)](#), while focusing on different issues like technical debt and requirements gathering, both conclude that typical pressures in a start up environment require corner cutting in one or more areas which eventually must be dealt with as the company stabilizes and the product matures. It's not a question of if corners will be cut, but how. This differs from the national

laboratory context markedly, as most projects are very mature and new projects typically have longer time lines and larger budgets.

More interesting were the papers on projects in industry, but these works were typically applicable to any software engineering project or too niche to be of use to ASC. There are some interesting highlights: [Ju, Sajnani, Kelly, and Herzig \(2021\)](#) investigates how on-boarding processes impact development, finding that onboarding which has a new employee first learn the big picture of the tasks they're assigned to do, finishing their first task quickly, and receiving positive feedback for their first successfully completed tasks all strongly corresponded to positive long term employee outcomes. Thinking about how to do onboarding better in a national lab setting might help with long term attrition issues. Understanding how early interactions among developers can be modeled and subsequently used to predict future success is an open question for future work.

One common theme among industry focused papers was a reliance on using just one large tech company with enough resources to have its own research branch (e.g., Microsoft, Google, etc.) as the data source. That is to say, a common theme of some team at Microsoft sending out a survey to all of its developers, having them complete it, and then writing up the results – perhaps with a collaborator or two from another company. But Microsoft, Google, and other large tech firms are an abnormality in the vast ecosystem of computing. It is important to verify that findings from studies centered on these companies are replicable. Especially to other venues which aren't computing centric. Replicating works done at large tech companies at other large companies that aren't tech firms yet nonetheless heavily rely on programming and employ lots of software developers – like banks or state governments – is an important area of research to pay attention to going forward.

2.11 Conclusion

This report surveyed hundreds of papers, classified them, and evaluated several topics in depth. Although there are many papers relevant to software development generally, which are helpful to the advanced scientific computing community, there are no papers focused specifically on software engineering in the advanced scientific computing context. Formally and rigorously examining how aspects of the development process are similar or different in the ASC context is a fruitful area for further investigation.

CHAPTER III

INVENTORY OF SOFTWARE REPOSITORIES IN NATIONAL LABORATORIES

This chapter contains previously published and unpublished material with co-authorship. It is based off of work submitted to CiSE titled, “A Survey of Open Source Software Repositories in the US Department of Energy’s National Laboratories.” Elements of the “Brief Summary” and “Introduction and Motivation” sections were drawn from text written by Dr. Anshu Dubey. All other figures, text, experimental design and code which may appear in the CiSE article were developed by me. Dr. Steve Fickas and Dr. Boyana Norris assisted with editing and discussions in an advisory capacity.

3.1 Brief Summary of Chapter

There are 17 national laboratory systems in the United States operating under the auspices of the US Department of Energy. These government labs employ tens of thousands of people engaging in research software engineering activities across a variety of missions. To support this work, a large number of open source projects are maintained. Many of these projects have broad utility to the computing community at large and domain scientists in a variety of fields. However, the complexity and decentralized nature of the laboratory system has resulted in a situation where no one entity even knows about all the open source software projects in this ecosystem, let alone crude metrics of their health, as we observed in Chapter II. In this chapter, we do the first external inventory of all open source software repositories with a nexus to DOE labs. We then identify repositories with a community, and of those, which could use sustainability supports.

3.2 Introduction and Motivation

Engineering and sustainability of research software has emerged as an important element of scientific discovery. As more software projects move to the open source model at research institutions, an inventory of such projects can lead to many insights about the work being done, where gaps exist, and how we can develop tools, deploy resources, or create policies to bridge gaps and accelerate existing research software engineering work.

Given the hundreds of research institutions around the globe writing software – universities, think-tanks, research divisions of large tech giants, etc. – we begin by scoping our work by focusing solely on the US Department of Energy’s (DOE’s) system of national laboratories (labs).

The DOE’s system of 17 national laboratories is large, complex, and heavily reliant on a variety of software projects to deliver on its multi-pronged research missions related to the nation’s energy development and independence, nuclear stewardship, and general scientific advancement.

Readers less familiar with the national laboratory system might appreciate an analogy to a state university system. While all labs engage in research activities, each lab is an independent institution with a distinct culture, history, and traditions, with friendly rivalries and shared research collaborations with their fellow labs. They primarily employ scientists with advanced degrees and most labs have robust student programs, too.

The many open sourced projects developed in the DOE lab system – often developed in close collaboration with academics and industry professionals – occupy a unique niche in computing. Due to the unique positions of the labs in US society, the research projects developed by the labs are not quite traditional closed-sourced

industry projects, but nor are they wholly volunteer-developed open source software. An empirical focus on these research software engineering projects in the DOE lab ecosystem has received very little attention in the empirical software engineering and repository mining literature [Mundt et al. \(2022\)](#); [Schwartz \(2023\)](#), and so we begin here.

This chapter doesn't even look to do a preliminary comparison of national laboratory software repositories with other kinds of projects in industry or open source. That is future work. Rather, the exploration of this important niche of software repositories is at such an early stage that we first need to examine the landscape and take a rough inventory of the existing set of projects in the ecosystem before further analysis can be done.

Creating an inventory of lab-affiliated software repositories is a non-trivial task. Indeed, while some individual national laboratories are in the process of aggregating and highlighting some projects done on their campuses, such as the Radius initiative at Lawrence Livermore National Laboratory, there is not a single listing of all the software projects and repositories created at the national laboratories.

It is with this in mind that we ask and explore the following research questions (RQs):

RQ1 What are all of the public software projects / repositories with a nexus to the US Department of Energy's national laboratories?

RQ2 Of the projects / repositories in RQ1, which ones are likely to be popular outside of the project's core developers? That is, which projects are used by the community – either a lab-internal community, or external/world-at-large community?

RQ3 Of the projects / repositories used by the community, which are still actively developed or maintained? Are there unmaintained projects with community use?

We answer each of these questions through three distinct analyses, each with their own exploratory methodology and results, and described in the following three sections.

It is also important for readers to also understand the wider context in which this study takes place. With the sunseting of the multi-laboratory, multi-year, multi-billion dollar Exascale Computing Project (ECP), a renewed focus on software sustainability has taken root in the national laboratory community, as evidenced by the PESO project and its related spokes. In essence, the PESO project is concerned with the following question: now that a variety of science-related software projects in which the government has invested millions of dollars over many years will no longer have dedicated funding, what will happen to them? How will they be maintained over time? Will they be included in the next major funding cycle, which is expected to be artificial intelligence focused? While these are largely business and political funding questions outside the scope of this chapter, this work is timely in that we believe this contribution will also help decision-makers to better address these policy decisions in future funding cycles.

3.3 What are all the open-source software repositories with a nexus to a national laboratory?

Our goal is to cast a wide net in capturing software repositories with some sort of nexus to one of the national laboratories. To do this, we take a multi-pronged approach to building our inventory. To summarize at a high level, we found lab software repositories by:

1. Scraping the lab websites for links to GitHub repositories.
2. Searching on GitHub for repositories with lab-related keywords.
3. Mining data within Spack, a publicly available build system for scientific software and supercomputers which is widely used within many labs, for links to GitHub repositories.

We chose these approaches because they are (1) replicable from year to year as new repositories are made publicly available, (2) do not depend on personnel or information from inside the national laboratory ecosystem, and (3) do not depend on third-party curations of GitHub data, which may not always be up-to-date.

We describe each of these approaches, and their results, in more depth below. Unless otherwise specified, we deployed these approaches the weeks of June 12-16, 2023. For reference, each of the approaches takes advantage of the information contained within Table 1.

3.3.1 Web Scraping. We employed Scrapy, a Python based web scrapping framework, to mine for GitHub projects. Specifically, we pointed our web-scraping “spider” to the home pages listed in Table 1, and told it to crawl all the webpages under that domain while adhering to the restrictions outlined in each website’s `robots.txt` file. The spider crawled the website of each of the national labs and the Department of Energy website itself using a breadth-first-search, looking for GitHub links on each page. We gave a timeout window of 200 hours for the spider to run and then used regex matching to identify GitHub repository landing pages.

3.3.1.1 Results. Over 200 hours, our spider traversed 5,205,676 different URLs, and found 1,664 unique repositories directly linked from one or more of the national lab websites. The breakdown of repositories by source domain is in Table 2.

Table 1. Table of United States Department of Energy National Laboratories and their websites.

Name (NL=National Laboratory)	Acronym	Website
Ames NL	Ames Lab	ameslab.gov
Argonne NL	ANL	anl.gov
Brookhaven NL	BNL	bnl.gov
Fermi National Accelerator Laboratory	FNAL	fnal.gov
Idaho NL	INL	inl.gov
Lawrence Berkley NL	LBL	lbl.gov
Lawrence Livermore NL	LLNL	llnl.gov
Los Alamos NL	LANL	lanl.gov
National Energy Technology Laboratory	NETL	netl.doe.gov
National Renewable Energy Laboratory	NREL	nrel.gov
Oak Ridge NL	ORNL	ornl.gov
Pacific Northwest NL	PNNL	pnnl.gov
Princeton Plasma Physics Laboratory	PPPL	pppl.gov
SLAC National Accelerator Laboratory	SLAC	slac.stanford.edu
Sandia NL	SNL	sandia.gov
Savannah River NL	SRNL	srnl.doe.gov
Thomas Jefferson National Accelerator Facility	JLAB	jlab.org
US Department of Energy	DOE	energy.gov

Table 2. Repositories found on each domain crawled with our web scraping spider.

Domain	Repositories Found
ameslab.gov	10
anl.gov	482
bnl.gov	82
energy.gov	60
fnal.gov	9
inl.gov	34
jlab.org	35
lanl.gov	84
lbl.gov	184
llnl.gov	400
netl.doe.gov	11
nrel.gov	4
ornl.gov	212
pnnl.gov	52
pppl.gov	2
sandia.gov	29
slac.stanford.edu	8
srnl.doe.gov	0

Some repositories were linked from multiple domains. Of the $\binom{18}{2} = 153$ pairs of domains, only 31 pairs (20%) had any repository links in common, and even then there was very little overlap – typically just one or two repositories at most.

3.3.2 Manually Searching on GitHub. We used GitHub’s search feature to attempt to find repositories and organizational accounts affiliated or linked with each of the 17 national labs. To do this, we did at least three separate searches for each lab. We searched using the name of the lab, including with the phrase “national laboratory” as well as without (e.g., both “Sandia” and “Sandia National Laboratory”), the acronym of the lab, and the lab’s website. We then identified any lab-affiliated organization on GitHub from repositories in the search results and recorded all repositories housed within the organizational account. We also identified repositories found with a nexus to a national laboratory but not housed under an

organizational umbrella, such as those repositories owned by a user with a lab email address or other clear affiliation (e.g., “I work at Argonne National Lab” written on the profile).

This labor-intensive identification and selection work was primarily done by the first author, with other authors verifying a subset of the search results and identified repositories to confirm consistency in the selection process.

Once again, we chose to err on the side of inclusion when identifying repositories possibly linked to a national laboratory in some capacity. However, preliminary analysis demonstrated that some search terms resulted in too many unrelated repositories in the first few pages of search results. For example, the “ANL” keyword resulted in over 3000 repositories, many of which were related to *automated/artificial natural language* or *a native linux* and thus not (directly) related to national laboratory work. Given the limited resources of our human team to manually comb through these results in assessing each repository’s salience regarding a nexus to a national laboratory, search queries which led to over 250 repositories being listed were not examined.

3.3.2.1 Results. Using GitHub’s search feature to find lab-related projects resulted in 6,864 unique repositories. We break out the results by search query in Table 3.

Unlike in the web-crawling approach, where there was minimal overlap between lab groups, some of these search queries resulted in substantial overlap in relevant found repositories. For example, the search phrase “Department of Energy” listed the organizational page <https://github.com/NREL> in the results. The search phrase “National Renewable Energy Laboratory” also linked to this page. Therefore, the 520+ NREL repositories listed on the GitHub NREL page could be found just one

Table 3. Number of unique repositories with a legitimate nexus to a national laboratory, by grouping of initial search terms. Note that the results sum to a number much greater than the 6,864 unique repositories found. This is due to substantial overlap in search findings among several of the different search groupings. Search terms denoted with an * returned over 250 search results of individual repositories and organizational pages with multiple repositories listed – up to 55,070 repositories and organizations in the case of “Energy” – and these results were not examined further.

Search Queries	Repositories Found
Ames National Laboratory, Ames*, Ames Lab, ameslab.org	3
Argonne National Laboratory, Argonne, ANL*, anl.gov	582
Brookhaven, Brookhaven National Laboratory, BNL*, bnl.gov	286
Fermi National Accelerator Laboratory, Fermi*, FNAL*, fnal.gov	302
Idaho*, INL*, Idaho National Laboratory, inl.gov	6
Lawrence Berkeley National Laboratory, LBNL, Lawrence Berkeley, lbl.gov	1789
Lawrence Livermore National Laboratory, Lawrence Livermore, LLNL, llnl.gov	904
Los Alamos National Laboratory, Los Alamos, LANL, lanl.gov	441
National Energy Technology Laboratory, NETL*, netl.doe.gov	13
National Renewable Energy Laboratory, NREL*, nrel.gov	560
Oak Ridge National Laboratory, Oak Ridge, ORNL, ornl.gov	282
Pacific Northwest, Pacific Northwest National Laboratory, PNNL, pnnl.gov	863
Princeton Plasma Physics Laboratory, PPPL, pppl.gov	92
SLAC National Accelerator Laboratory, SLAC*, SLAC NAL, slac.stanford.edu	301
Sandia*, SNL, Sandia National Laboratory, sandia.gov	871
Savannah River National Laboratory, Savannah River, SRNL, srnl.doe.gov	480
Thomas Jefferson National Accelerator Laboratory, TJNAL, Jefferson Laboratory, jlab.org	522
Energy*, DOE*, Department of Energy, doe.gov, energy.gov	691

or two clicks away after searching for either “Department of Energy” or “National Renewable Energy Laboratory.”

3.3.3 Spack Mining. Spack is a build system for high performance computing software, and widely used within the national laboratory ecosystem [Gamblin et al. \(2015\)](#). As a build system, Spack maintains an internal listing of software repositories, their source locations, and their dependencies via plaintext package configuration files. These files are located at <https://github.com/spack/spack/tree/develop/var/spack/repos/builtin/packages/>.

We mined these files. Specifically, for each software package maintained, we parsed the `package.py` configuration file for the same keywords used in our general GitHub mining via keyword search. If text in a package configuration matched one or more of the search keywords and had a linked GitHub repository, that repository was recorded. We excluded the copyright line from consideration, which always lists Lawrence Livermore since Spack itself is a product from Lawrence Livermore National Lab.

3.3.3.1 Results. We mined Spack configuration files by keyword search using the same queries used in the manual GitHub mining and found 159 distinct repositories. Many queries resulted in no results at all. The queries which did result in a repository are in [Table 4](#).

Finally, there were only six pairs of search terms from non-concordant laboratories (e.g., `lbl.gov` and `ornl.gov`) with repositories in common. In each case, there was only one repository in common between two distinct labs. This is expected, as most projects have one primary institutional home, which is what would be formally recorded in a build system configuration file, even if the contributor or sponsorship base is distributed across many different laboratories and affiliated institutions.

Table 4. Search queries which were found in a Spack configuration file, by number of GitHub repositories found.

Search Query	Repositories Found
ANL	5
anl.gov	3
BNL	1
bnl.gov	1
Fermi	3
FNAL	3
lbl.gov	4
Lawrence Livermore	1
LLNL	89
llnl.gov	20
Los Alamos National Laboratory	1
Los Alamos	3
LANL	26
lanl.gov	3
NREL	5
ORNL	6
ornl.gov	4
Sandia	7
sandia.gov	5
Energy	12
DOE	2
Department of Energy	1

3.3.4 Consolidation. The GitHub repositories found through the above approaches were pooled into one set of consolidated projects which could then be analyzed. We found 8,174 distinct national laboratory-linked repositories through our three approaches.

We provide this list of repositories as supplemental material here: (link to sup1.txt)

There was surprisingly little overlap across the three approaches. Of the 6,864 repositories found through manual searching and the 1,669 repositories found through web scraping, only 73 repositories were found through both approaches. The approach through mining spack, with its total of 159 repositories, fared a bit better; with 72 and 22 repositories overlapping with the manual searching and web scraping approaches, respectively. Understanding differences and similarities among repositories which were found through one approach or several is an area of future work.

A cursory glance at the list of 8,174 curated repositories with a nexus to a national laboratory that we found in the previous section shows that our wide-cast net pulled in multiple flavors of software repositories by keywords in the names alone; from projects related to student summer internships, to software carpentry, to conference workshops, as well as repositories housing complex scientific codes. All of these projects make up the software landscape of national laboratories, and understanding these projects better is an ample area of future work. We focus our sights first on those projects which aren't just developed by-and-for one or two people, but are popular and have a community.

3.4 Analyzing Project Popularity

In this section we answer RQ2: Of the projects/repositories in RQ1, which ones are actively used by people outside of the project's core developers? That is,

which projects are used by the community – either a lab-internal community, or external/world-at-large community?

To do further analysis on the repositories identified in the previous section, we must first gain additional information about these projects. We obtained meta-data on each of them by using the command-line GitHub API tool with the command `gh api -H "Accept: application/vnd.github+json" -H "X-GitHub-API-Version: 2022-11-28" /repos/OWNER/REPO` where OWNER/REPO is replaced with each name of the repository listed in the supplemental material.

The returned meta-data included the number of individuals who had “starred” the repository on GitHub, as well as the number of forks of the repository. Previous research has indicated these two metrics are highly correlated with repository popularity [Borges, Hora, and Valente \(2016\)](#). We also considered the number of GitHub issues opened and closed on a project, but unfortunately the meta-data available through the API only contained the number of currently open issues. Obtaining the full count of all issues ever opened required significantly more API calls, which GitHub rate-limits. Given forks, stars, and issues are all correlated [Yamamoto et al. \(2020\)](#), we decided to stick with stars and forks in light of the large set of repositories we were already collecting data from.

We were able to obtain meta-data on all but 2.9% of repositories. We continued the analysis on the remaining 97% of repositories for which meta-data was returned. We provide this meta-data as a cleaned CSV file located here: ([link to sup2.csv](#))

To better understand how our dataset compared with others, we did a histogram analysis of repositories by number of stars and forks to find a threshold where repositories have a community within the national laboratory context.

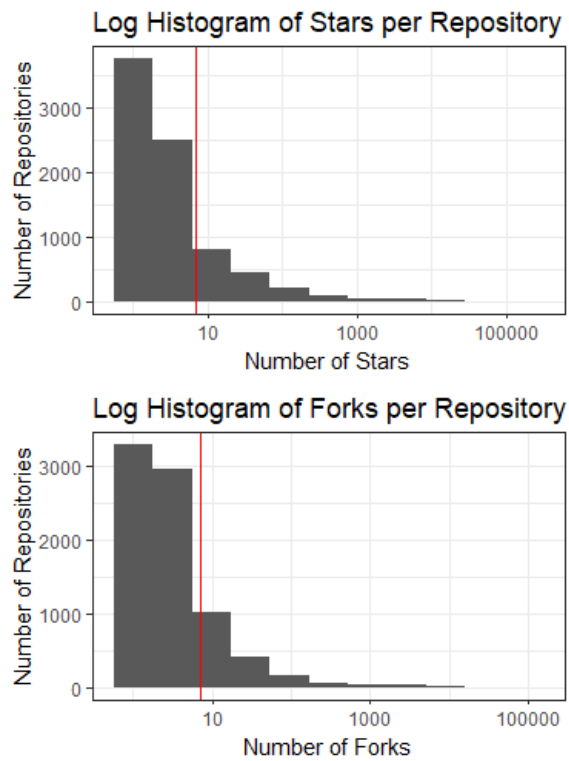


Figure 3. Histograms of repositories per stars and per forks, on a \log_{10} scale. We use these histograms to construct criteria for whether a repository is sufficiently popular and has a community for further analysis. We assert a repository is more likely to have a community if it has at least six stars or six forks (red lines).

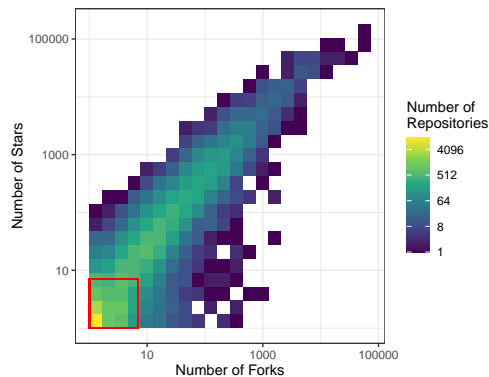


Figure 4. Combined 2D bin plot, showing the number of repositories per each star-fork intersection on a \log_{10} - \log_{10} - \log_2 scale. Note that the vast majority of repositories are those which we assert are not popular enough for further analysis and are highlighted by the red bounding box of six forks and six stars.

Based on the plots in 3, we assert that six forks or stars is an appropriate threshold for identifying repositories with at least some popularity or community. We also see an exceptionally strong correlation between stars and forks, with Pearson’s r correlation coefficient at 0.9638, which we see clearly visualized in figure 4 and which aligns with previous work.

3.4.0.1 Results. To answer the question of how many national laboratory projects have a community, we thresholded our set of repositories identified in RQ1 by repositories with six stars or six forks, following our histogram analysis. The number of repositories meeting these thresholds, and therefore the number of repositories we reasonably conclude have a community, is 2,005. This is 25% of all identified national laboratory linked projects for which meta-data was available.

Further, within the context of national laboratory-linked repositories, we find that stars and forks are closely correlated, which echoes prior work. These findings suggest that the popularity of open source repositories linked to national laboratories follow trends around open source repository popularity generally. Future work in this area includes questions about whether the trajectory of a repository becoming popular over time mirrors that of open source projects generally.

We should also caution and remind readers that repositories with a nexus to a national laboratory, which includes repositories that were simply linked from a national laboratory webpage, are not necessarily national laboratory owned repositories. For example, the repository in our sample with the most stars and forks is the repository for tensorflow, the popular neural network library, which is owned by Google. However, there are clearly highly popular lab-owned projects too. Spack itself is an open source project housed on GitHub, and it has well over a thousand stars and forks.

3.5 Identifying Repositories in Need of Sustainability Supports

In this section we answer RQ3: Of the projects/repositories used by the community, which are still actively developed or maintained? Are there unmaintained projects with community use?

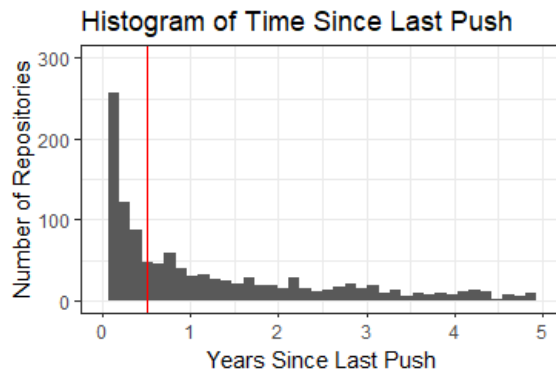


Figure 5. Histogram of the last time someone pushed to a repository, going back five years. We consider the 2,005 repositories identified in RQ2 with a link to a national laboratory and a likely community. The red line is the six month mark.

Also included in the meta-data, downloaded as part of our efforts to answer RQ2 in the previous section, is the last time a repository was pushed to. We use this data, in consultation with histogram analysis shown in Figure 5, to begin to identify active and inactive projects with communities.

We find that, of our set of 2,005 popular repositories with a nexus to a laboratory identified in RQ2, just over half were updated within the previous 12 weeks, and 1,200 (60%) were updated within the previous six months. We also assume that projects last updated more than two years ago are more likely than not to have communities which have fully moved on. While not pictured in a figure, we also found that around two years the cumulative distribution function of this data begins to flatline.

Consequently, we focus our attention on those repositories updated within the last two years, but longer than six months ago. This corresponds to 371 repositories, which we believe reflects a fuzzy upper bound for the number of open source repositories which could use sustainability supports. We call our upper bound “fuzzy” as there may be repositories which are not well-advertised online, and thus not

included in our initial set of 8,174 lab-linked repositories, yet are still used by a community. It also does not include repositories which are closed-source.

We provide the full list of 371 repositories as supplemental information here: (link to sup3.txt)

An examination of these repositories suggests that a fuzzy lower bound for the number of repositories which could use sustainability supports is significantly lower than our upper bound. For example, a cursory examination of these 371 repositories indicate that several have the name “workshop” in the repository name, suggesting that these repositories were made for a particular workshop in the past then presented and stared and forked by the attendees of the workshop. Repositories developed for one-time workshops are unlikely to need ongoing sustainability supports. Other projects may still be active but their code has been reorganized into other repositories. For example, Alpine-DAV/vtk-h, the code for which is still being actively developed but under the auspices of the Alpine-DAV/ascent repository.

To better understand these 371 projects, we selected a random sample of 30 repositories and, by reviewing their home pages on GitHub, we identified whether they would be worthy of consideration for sustainability supports or if there were clear reasons why ongoing sustainability supports from national laboratories would be inappropriate. A selection of 30 repositories against a population of 371 has a margin of error of under $\pm 15\%$ on a confidence interval of 90%.

We found that 21/30 (70%) of repositories in our sample were clearly research software engineering projects owned or primarily developed by an employee at a national laboratory which, at a cursory glance, might welcome sustainability supports. Three were from ANL, one from Jefferson, five from LANL, four from LLNL, one from NREL, five from PNNL, and two from Sandia.

Four projects (13%) were clearly not suited for sustainability supports. One was code for an “about page” of a popular consortium’s website, two were conference tutorials, and one was a “quick start” guide for running code on a supercomputer at ORNL.

The remaining five projects (17%) were “maybes” in terms of sustainability supports, assuming the support comes from national laboratory budgets. Three of these projects were research software engineering projects which could use sustainability supports, but were owned or developed by non-lab sources (two were US university projects, one was owned by Toyota Research). One repository hosted a non-research software engineering project dealing with authentication protocols. Yet another repository hosted a multi-year internship project expanded on by new LLNL interns every year.

Assuming 70% of our 371 repositories are reasonably eligible for sustainability supports as research software engineering projects owned by a national laboratory, then we’re looking at around 260 projects as a tighter fuzzy upper bound.

Understanding which of each of these plausibly eligible repositories truly do need sustainability supports ultimately comes down to talking with the human beings behind the projects, which is beyond the scope of this paper. However, this list of 371 projects is a good starting point for future work.

3.6 Sidebar: An analysis mining GitHub via BigQuery

As a point of interest, we also explored BigQuery data. Google’s BigQuery system has a public database with multiple terabytes of GitHub activity. We ultimately decided against using the findings in our consolidated results due to an excess of non-RSE repositories in the results. However, as a point of interest, we include this “sidebar” of information for completeness.

We mined this public GitHub database in BigQuery with the following command, which searched for all committers with a .gov email address:

```
[breaklines,      rulecolor=,      frame=single,language=sql]      SELECT
repo_name, author.name AS author_name, author.email AS author_email, committer.name as committer_name
public - data.github_repos.commits WHERE CONTAINS SUBSTR('bigquery -
public - data.github_repos.commits'. 'author'. 'email', '.gov') OR CONTAINS SUBSTR('bigquery -
public - data.github_repos.commits'. 'committer'. 'email', '.gov') OR CONTAINS SUBSTR('bigquery -
public - data.github_repos.commits'. 'trailer', '.gov')
```

We chose to search by a broad .gov filter initially so that we could compare national lab GitHub activity with general government GitHub activity, and we speak to this in our results section.

We also repeated the query above, but with the domains `slac.stanford.edu` and `jlab.org` replacing .gov so that all lab website domains were included in our final data set for analysis. We then further filtered these combined results by committing users with email addresses from the 17 national labs¹ to arrive at our final pool of repositories from this approach.

We first queried the Google BigQuery GitHub database for all commits with a .gov email address. Nearly a million commits were found. The oldest commit was timestamped at Friday, April 1, 1994 and the newest at Friday, November 25, 2022.² As a point of interest, we aggregate the email addresses by both commits and by unique committer, annotating for national lab addresses, obvious foreign email addresses, and other government addresses.

¹In addition to the @website.gov addresses listed in Table 1, we also included @pnl.gov to account for grandfathered Pacific Northwest National Laboratory addresses.

²GitHub released a new version of its API on the following Monday, which may be why the database hasn't been updated since November 2022.

Table 5. Number of GitHub commits in Google’s BigQuery archive service made with an email address containing gov.

Domain	Number of Commits	% of all Commits	% of Category
<i>Non-US</i>			
gov.uk	223128	22.4%	76.5%
gov.au	31033	3.1%	10.6%
gov.br	24196	2.4%	8.3%
gov.pl	4039	0.4%	1.4%
gov.ar	2290	0.2%	0.8%
govt.nz	2105	0.2%	0.7%
<i>All others</i>	4720	0.5%	0.2%
All Non-US Domains	291511	29.3%	100%
<i>US, Non-Lab</i>			
nasa.gov	61187	6.2%	21.9%
nih.gov	57859	5.8%	20.7%
usgs.gov	28786	2.9%	10.3%
cfpb.gov	27579	2.8%	9.9%
noaa.gov		1.7%	6.0%
<i>All others</i>	87327	8.8%	31.2%
All Non-Lab US Gov Domains	279520	28.1%	100%
<i>US Lab</i>			
ornl.gov	80166	8.1%	18.9%
lbl.gov	64521	6.5%	15.3%
llnl.gov	59338	6.0%	14.0%
inl.gov	41001	4.1%	9.7%
anl.gov	38608	3.9%	9.1%
bnl.gov	33224	3.3%	7.9%
lanl.gov	29243	2.9%	6.9%
fnal.gov	23080	2.3%	5.5%
sandia.gov	18023	1.8%	4.3%
nrel.gov	17521	1.8%	4.1%
pnnl.gov	15860	1.6%	3.7%
pnl.gov	1440	0.1%	0.3%
pppl.gov	826	0.1%	0.2%
ameslab.gov	169	0.0%	0.0%
doe.gov	44	0.0%	0.0%
All US Gov Lab Domains	423064	42.6%	100%
<i>All domains</i>	994095	100%	100%

We also included committers with emails from the domains `jlab.org` and `slac.stanford.edu` in a secondary query to obtain the full complement of national lab email addresses, which contributed 3833 and 7663 commits, respectively.

A total of 15472 distinct repositories were found from this approach, with the results broken down by domain.

Table 6. Number of distinct GitHub repositories which a user with an email from the associated domain has ever committed to.

Domain	Users with email addresses from domain have committed to this many distinct repos
ameslab.gov	3
anl.gov	489
bnl.gov	284
fnal.gov	227
inl.gov	95
jlab.org	20
lanl.gov	260
lbl.gov	596
llnl.gov	285
nrel.gov	168
ornl.gov	418
pnl.gov	31
pnnl.gov	145
pppl.gov	13
sandia.gov	215
slac.stanford.edu	178
srnl.doe.gov	0
netl.doe.gov	5

Note that the sum of the repositories does not add up to 15472, due to the overlap of repositories.

We note there is more interaction amongst the repositories by collection through this approach. Of the $\binom{18}{2} = 153$ pairs of domains, 80 pairs (52%) had any repository

Table 7. Domains where contributors from the two labs committed to 10 or more repositories. Domain refers to the domain of a contributors email address, repos are the number of distinct repos found for each domain, IS is the intersection size (i.e., the number of repositories in common), OC is the overlap coefficient, and SD is the Sorensen-Dice metric.

Domain	Repos	Domain	Repos	IS	OC	SD
anl.gov	489	inl.gov	95	10	0.11	0.03
llnl.gov	285	sandia.gov	215	10	0.05	0.04
anl.gov	489	lanl.gov	260	11	0.04	0.03
anl.gov	489	sandia.gov	215	11	0.05	0.03
bnl.gov	284	slac.stanford.edu	178	11	0.06	0.05
lbl.gov	596	slac.stanford.edu	178	12	0.07	0.03
lanl.gov	260	llnl.gov	285	13	0.05	0.05
llnl.gov	285	ornl.gov	418	16	0.06	0.05
lanl.gov	260	sandia.gov	215	16	0.07	0.07
anl.gov	489	llnl.gov	285	17	0.06	0.04
ornl.gov	418	sandia.gov	215	17	0.08	0.05
lanl.gov	260	ornl.gov	418	18	0.07	0.05
pnl.gov	31	pnnl.gov	145	18	0.58	0.20
bnl.gov	284	ornl.gov	418	30	0.11	0.09
anl.gov	489	bnl.gov	284	37	0.13	0.10
bnl.gov	284	lbl.gov	596	42	0.15	0.10
lbl.gov	596	ornl.gov	418	62	0.15	0.12
anl.gov	489	ornl.gov	418	67	0.16	0.15
anl.gov	489	lbl.gov	596	114	0.23	0.21

links in common, with 18 pairs of domains having more than 10 repositories in common. Of the top three, we once again see ANL, BNL, and ORNL having the most interaction.

3.7 Conclusion

In this chapter we asked three questions: What are the public repositories with a nexus to a US DOE national laboratory? Of these projects, which are popular with users outside of the projects' core developers? Of the popular projects, how many are actively maintained vs unmaintained?

Key Takeaway: Of the more than 420 million repositories on GitHub, we found 8,174 open-source repositories with a nexus to one of the United States Department of Energy’s national laboratories. Of these 8,174 repositories, we conjecture that 2,005 likely have some community interest. Among these 2,005 repositories, we submit that up to approximately 260 could use sustainability supports to help with ongoing maintenance.

Discussion and Implications: Hiring research software engineers to provide sustainability support to projects will cost money. National laboratories are complex organizations from a business and budgeting perspective, and the financial resource allocations of these organizations go well beyond the scope of this paper. Still, we submit new funding is needed to sustain all of the open-source software repositories currently at risk of being abandoned by developers across all national laboratories.

Future work: This space has ample future work. One area includes examining other venues where open-source research software engineering projects are developed – such as in research universities and comparing them with the national laboratory context, which we do in the next chapter. Other areas for future work are more methodological, such as incorporating additional data from GitHub, like issues and developer discussion, as part of this empirical analysis. Moreover, conducting this kind of systemic inventory on a regular basis to monitor changes and understand developments over time would also be an exciting comparison going forward. Finally, doing bus factor analysis – quantifying and identifying developers whose departure from a project could have a significantly negative impact – may also help in further identifying and prioritizing projects with respect to ongoing sustainability supports and funding.

CHAPTER IV

INVENTORY OF SOFTWARE REPOSITORIES IN U.S. UNIVERSITIES

This chapter contains previously unpublished material with co-authorship. All figures, text, experimental design and code which appear were developed by me. Dr. Steve Fickas and Dr. Boyana Norris assisted with editing and discussions in an advisory capacity.

4.1 Brief Summary of Chapter

In this chapter we aim to partially answer the question, “Just how many research software projects are out there?” by searching for open source GitHub projects affiliated with research universities in the United States. We explore this through keyword searches on GitHub itself and by scraping university websites for links to GitHub repositories. We then filter these results by using a large language model to classify GitHub repositories as research software engineering projects or not, finding over 35,000 RSE repositories. We report our results by each university. We then analyze these repositories against metrics of popularity, such as stars and repository forks, and find just under 14,000 RSE repositories meet our minimum criteria for projects which have a community. Based on the time since a developer last pushed a change to a RSE repository with a community, we further posit that 3,300 RSE repositories with communities and a link to a research university are at risk of dying, and thus may benefit from sustainability support. Finally, across all RSE projects linked to a research university, we empirically find the top repository languages are Python, C++, and Jupyter Notebook.

4.2 Introduction

Research software engineering (RSE), as a formal academic field of study, is relatively new. Broad questions like, “*Just how many research software projects are*

out there? Empirically speaking, what do they look like?” continues to be ripe for answering. In this chapter we expand on our earlier work in Chapter III, which explored and inventoried open source software projects hosted on GitHub with some nexus to a US Department of Energy national laboratory. In this work we apply a similar methodology to uncover RSE repositories, but point to a comparatively larger ecosystem: all major research universities in the United States.

Specifically, the overarching research questions which guide this work are:

RQ1 What are all of the public software projects / repositories on GitHub with some nexus, even a weak one, to the major research universities in the United States?

RQ2 Of these software projects with some sort of nexus to a research university, which ones are RSE projects? More generally, if given a large set of repositories how can we determine which ones are RSE and which are non-RSE?

RQ3 Of the RSE projects / repositories found through answering RQ2, which ones are likely to be popular outside of the project’s core developers? That is, which projects are used by the community – either a university-internal community, or external/world-at-large community?

RQ4 Of the RSE projects / repositories used by the community, which are still actively developed or maintained? Are there unmaintained projects with community use? If so, do we have any sense of what skills are needed to maintain them?

In answering these questions through previously established exploratory approaches, we also contribute new findings around the use and efficacy of a modern large language model as an aid to automatically identify Research Software Engineering projects at an empirical scale.

4.3 RQ1: Projects with a nexus to a major US research universities

4.3.1 Definitions, Initial Scope, Overarching Approach: Our goal is to cast a wide net when answering our formal research question, “RQ1: What are all of the public software projects / repositories with some nexus, even a weak one, to the major research universities in the United States?” That said, we do define some scoping parameters. First, we will limit our search to projects ultimately hosted on GitHub. We also define “major research university” as “R1” universities under the Carnegie Classification of Institutions of Higher Education.

We identify universities meeting the R1 classification by relying on the data reported by the universities themselves to the federal government’s Integrated Postsecondary Education Data System (IPEDS) run by the National Center for Education Statistics. All institutions of higher education are required to submit information to this system on an annual basis. We used the IPEDS data from the 2022 reporting cycle, the most recent available, to identify 146 R1 universities to evaluate. This data set also contained information about a university’s institutional characteristics, such as the website of each university, and self-reported aliases and nicknames (e.g., “MIT” for the Massachusetts Institute of Technology).

To identify software projects associated with these research universities we deployed two approaches. (1) We first scraped the websites of each of the universities, looking for links to GitHub repositories. (2) We then searched GitHub itself for repositories by using keywords associated with each of the universities.

4.3.2 Approach 1: Scraping university websites. On December 27, 2023 we deployed a webcrawling spider on all of the 147 R1 universities, using the root domain extracted by the self-reported URL from the 2022 IPEDS data as a starting point. This spider, which obeyed the robots.txt file from each university,

scoured the institutional websites using a breadth-first search traversal of links on HTML pages. The spider ran until all linkable HTML pages had been traversed or 200 hours had passed. On each page, the spider scraped any link to a GitHub repository. These links were subsequently cleaned and filtered for repository links of the form `github.com/owner/repository`. Duplicates were consolidated into a set of unique repositories.

4.3.2.1 Results. Of the 146 R1 universities, 116 (70%) allowed scraping of their website. The rest of this analysis focuses on these 116 universities.

Overall, 44,566,662 distinct URLs were traversed, and average of 384,195 per site. A total of 10,056 unique GitHub repositories were pointed to by one or more links across all of the 116 universities that permitted website scraping.

We compare these raw findings with the national laboratory study [Schwartz et al. \(2024\)](#) in Table 8.

4.3.3 Approach 2: Searching on GitHub. We also used GitHub itself to search for keywords associated with each university to find affiliated repositories, similar to the approach used in Chapter III.

The keywords used were lowercase versions of (1) the official name of the university (e.g., “University of Oregon”); (2) aliases drawn from self-reported IPEDS data (e.g., “UO,”) or derived from the domain name (e.g., “UOregon”) – which was feasible since GitHub searches are not case sensitive; and (3) the domain name itself (e.g., “uoregon.edu”). To provide consistency, we examined only the 116 universities which allowed web scraping.

GitHub helpfully provides a command line interface which provides the same search results (up to 1000 results) as its website’s search function, which sorts the results by GitHub’s self-described “best match.” We applied the command

Table 8. Comparison of repositories found in the US Department of Energy National Laboratory from the data obtained in writing Chapter III and US R1 Universities that permitted website scraping.

Comparison Group:	National Labs	R1 Universities
Number of institutions in group	17	116
Mean number of URLs scraped per institutional website	306k	384k
Minimum number of unique repositories found on a institutional website	0	0
Mean number of unique repositories found on institutional websites	94	58
Median number of unique repositories found on institutional websites	35	16
Maximum number of unique repositories found on a institutional website	482	951

`$ gh search repos keyword --limit 1000 --json url` to each of the keywords identified for each institution, which resulted in a list of up to 1,000 repositories for each keyword search.

4.3.3.1 Results. Across all 116 universities, 187,728 unique repositories were identified through keyword searches. However, unlike in Chapter III, we were unable to manually examine each repository and make a human determination if the project was a university-affiliated RSE repository or not. There were just too many results given time constraints.

However, we did carry out a cursory examination of several keyword search results. We found that some keywords resulted in many repositories which had little connection to either research or universities. This was particularly true for keywords associated with state or flagship public universities, whose keywords often were identical to the name of the state in which they were located. For example, when we examined the keyword “Arizona,” we found the query produced several search results which were clearly linked to research software projects at either the Arizona State University and the University of Arizona. These results also included repositories related to tourism in Arizona, a gaming engine framework that happened to be called Arizona, and other non-RSE projects. That said, we observed that the overwhelming majority of RSE projects found by searching with broad keywords like “Arizona” were, indeed, associated with one of the universities in Arizona.

Our next question then, became, “How do we identify which of these 187,728 repositories are RSE projects? For that matter, which of the 10,056 links to GitHub repositories found in Approach 1 are RSE projects? How can we do this in an automated fashion?”

4.4 RQ2: Is a given repository an RSE repository?

To determine whether a repository is an RSE repository at scale, we turned to OpenAI’s GPT 3.5 Large Language Model (ChatGPT). After exploring several different prompt templates, we settled on a format similar to the approach used by Masud and Faloutsos (2024). Namely, we asked ChatGPT to determine whether a repository was an RSE repository by providing a definition for RSE, providing a description of the repository, and, in breaking with Masud and Faloutsos (2024)’s requested binary yes-no output, we asked here for a probability from 0 to 1 that the repository was RSE. We did this because exploratory work with a handful of repositories suggested that relaxing to a continuous case led to more consistent and easily parsable responses than prompts which asked for a binary yes-no answer.

The definition for “RSE” came from <https://us-rse.org/about/what-is-an-rse/>. We created a description of each repository by piping the text obtained from the GitHub CLI command `gh repo view owner/repository`, which includes the repository’s name, brief description, and the contents of the README document if available. Putting it all together, the prompt followed this template:

Consider the following definition of a Research Software Engineer:

“We like an inclusive definition of Research Software Engineers to encompass those who regularly use expertise in programming to advance research. This includes researchers who spend a significant amount of time programming, full-time software engineers writing code to solve research problems, and those somewhere in-between. We aspire to apply the skills and practices of software development to research to create

more robust, manageable, and sustainable research software.”

Consider the following information about a Github repository:

====

"View Repo" Data Inserted Here

====

Given the above definition and GitHub repository information, is this GitHub repository a research software engineering project? Report your answer as a probability between 0 and 1. Place this number at the end of the message.

To validate this approach, we randomly selected two sets of 385 repositories, which gives a margin of error of $\pm 5\%$ on a 95% confidence interval. One set of 385 randomly selected repositories came from the pool of repositories found by scraping university websites (“scrape set”) and one set of 385 from the pool of repositories found by searching GitHub for keywords (“search set”). The first author manually identified each repository in both of these sets as an RSE or non-RSE repository in light of the description above.

In the case of the scrape set, 227 of 385 (59%) were manually identified as RSE repositories. In the case of the search set, only 50 of 385 (13%) were manually identified as RSE repositories.

In both of the sample sets, the last number of each output message given the above prompt to ChatGPT was, indeed, a figure between 0 and 1 which corresponded

to the likelihood of the repository being an RSE project or not. This number was extracted and paired with each repository.

Consequently, we had rows data, where each row was of the form [repo-name, human-assigned-boolean-value, chat-gpt-assigned-probability].

To convert the ChatGPT assigned probability to a Boolean value, we needed to apply a threshold. E.g., [repo-name, human-assigned-boolean-value, chat-gpt-assigned-probability > threshold].

To determine this threshold, we looked for values which would maximize a sense of agreement across the distribution between the human-labeled data and the ChatGPT labeled data. Specifically, we looked at Cohen’s Kappa, a common measure of inter-rater reliability. We also looked at F1 score, as a different measure to balance precision and recall. Across both data sets, Cohen’s Kappa and F1 score was maximized (or nearly maximized) when the threshold was set at 0.75. Results summarizing this agreement are in Table 9.

Table 9. Summary of human and ChatGPT agreement when labeling repositories in two different sets as RSE or non-RSE

Data Set:	Scrape	Search
Simple percentage agreement between human and ChatGPT RSE determination.	77%	90%
Cohen’s Kappa (Agreement levels of None, Minimal, Weak, Moderate, Strong, Almost Perfect McHugh (2012))	0.5 (Weak)	0.6 (Moderate)
F1 Score	0.82	0.66

These results are not perfect, but they seem to align with the results in the very little existing work around the classification of niche and technical topics using ChatGPT, such as in [Masud and Faloutsos \(2024\)](#). In this work the authors explored the empirical identification of malicious repositories that are nominally educational. They found that in “gray cases” ChatGPT couldn’t even agree with itself more than

90% of the time. In [Koopman and Zucco \(2023\)](#) the authors examined whether ChatGPT gave accurate yes-no answers to niche medical questions and found simple percentage accuracy of around 80%. Given this context, the agreement results we report here seem within the realm of state-of-the-art when using ChatGPT for classifying concepts which largely revolve around niche technical information and fuzzy definitions, although it is also clear there is ample room for improvement.

4.4.1 Results. We applied the above prompt to the combined 193,921 repositories from the union of the “scrape” and “search” data sets.

There were a few errors. In 615 cases (0.317%), GitHub could not find the repository. This was due to the repository being deleted or an error in the repository’s name from a link that was scraped. In 19 cases (0.010%), ChatGPT could not assign a probability. This was due to a lack of description or README about the repository. In 579 cases (0.299%), ChatGPT errored out, saying the prompt had an issue. This was always due to the project’s README having a non-standard character encoding, a binary file uploaded as the README, or an excessively long length.

In total, there were errors in 1,213 cases (0.626%). All of these repositories which encountered errors during processing were set aside. In the case of 192,708 repositories (99.374%) we were able to assign an RSE probability between 0 and 1. Applying our threshold of 0.75 to be considered RSE, we and ChatGPT ultimately identified 35,361 repositories (18.2%) as RSE repositories under the US-RSE definition. These results are disaggregated by university and shown in [Table 11](#). We also provide summary statistics about the number of repositories and RSE repositories found in [Table 10](#)

We also did overlap analysis. That is, which universities had repositories in common? Given our set of 116 universities, this resulted in $\binom{116}{2} = 6,670$ pairs of universities to examine. Of these 6,670 pairs, 5,142 (77%) had no repositories in

Table 10. Summary statistics (minimum, 25%, median/50%, mean, 75%, and maximum) of the number of repositories with a nexus to a university, and the number of RSE repositories, found across all universities.

	All Repositories	RSE Repositories
Min.	131	14
1st Qu.	1093	127
Median	1495	261
Mean	1796	334
3rd Qu.	2353	417
Max.	11856	2117

common at all, RSE or otherwise. 6,101 of the 6,670 (91%) had no RSE repositories in common. In the vast majority of these 9% of pairs with some RSE repositories in common, most institutions had just one or two. Understanding why overlap is so sparse, and the relationships among those universities which do have overlap, remain as future work.

Indeed, there were only 36 pairs which had at least 14 RSE repositories in common – 14 being the minimum number of RSE repositories found at any R1 university when employing both the scraping and keyword searching approaches. Within this set of 36 pairs, we could have up to 72 distinct universities listed. Instead, there were only 34 unique universities on this list, of which 13 appeared more than once.

This led us to create a threshold graph $T_{14} = (V, E)$ where V is the set of universities and E is a weighted edge, where weight is given by the number of RSE repositories in common between two between two vertices in V , and which exists if and only if the weight is greater than or equal to 14. This graph is visualized in Figure 6.

Figure 6 leads to more questions than we have answers. For example, why is there so much overlap between certain universities (e.g., University of Miami and

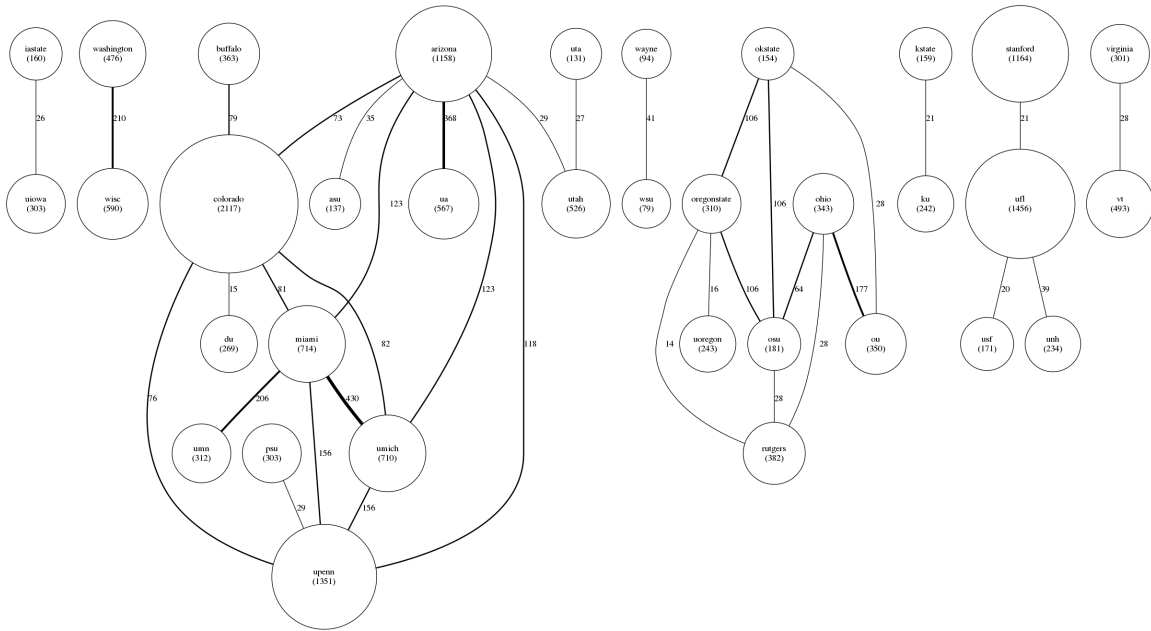


Figure 6. Graph T_{14} . Universities where the number of RSE repositories they have in common is greater than or equal to 14 repositories. Nodes are labeled by university and (total number of RSE repositories) in parentheses, edges are labeled by the number of RSE repositories in common.

University of Michigan)? Why these particular universities? Most of the Ivy League is missing, for example. Why are these universities clustered this way? Do human research networks also align somewhat with this graph? This is an area ripe for future work.

Table 11. All GitHub repositories found by university, through both scraping .edu websites for links to GitHub and by searching for keywords related to the university on GitHub itself. The number of RSE repositories, as determined by ChatGPT provided probabilities with a threshold of at least a 75% likelihood, is also shown.

Uni	Total Scrape	RSE (%)	Total Search	RSE (%)	Total Scrape ∪ Search	RSE(%)
-----	--------------	---------	--------------	---------	-----------------------------	--------

All	6346	4453 (70.2%)	187728	30971 (16.5%)	193921	35361 (18.2%)
arizona	81	42 (51.9%)	4146	1116 (26.9%)	4222	1158 (27.4%)
asu	10	7 (70%)	1371	130 (9.5%)	1381	137 (9.9%)
auburn	25	1 (4%)	1036	101 (9.7%)	1061	102 (9.6%)
bc	0	0 (0%)	1268	313 (24.7%)	1268	313 (24.7%)
binghamton	29	12 (41.4%)	434	45 (10.4%)	463	57 (12.3%)
brandeis	10	2 (20%)	490	103 (21%)	499	105 (21%)
brown	57	48 (84.2%)	1511	373 (24.7%)	1568	421 (26.8%)
bu	63	33 (52.4%)	3078	518 (16.8%)	3141	551 (17.5%)
buffalo	45	34 (75.6%)	2363	329 (13.9%)	2408	363 (15.1%)
case	17	15 (88.2%)	2391	488 (20.4%)	2408	503 (20.9%)
clemson	30	21 (70%)	782	147 (18.8%)	812	168 (20.7%)

cmu	132	113 (85.6%)	2361	746 (31.6%)	2492	858 (34.4%)
colorado	6	3 (50%)	11850	2114 (17.8%)	11856	2117 (17.9%)
columbia	49	27 (55.1%)	1871	404 (21.6%)	1919	431 (22.5%)
cornell	773	489 (63.3%)	1595	389 (24.4%)	2357	876 (37.2%)
dartmouth	75	45 (60%)	1165	213 (18.3%)	1239	258 (20.8%)
drexel	37	22 (59.5%)	1019	151 (14.8%)	1056	173 (16.4%)
du	0	0 (0%)	1616	269 (16.6%)	1616	269 (16.6%)
duke	23	18 (78.3%)	1909	246 (12.9%)	1932	264 (13.7%)
emory	28	20 (71.4%)	847	145 (17.1%)	874	165 (18.9%)
fiu	1	1 (100%)	1056	66 (6.3%)	1057	67 (6.3%)
fsu	25	19 (76%)	2498	213 (8.5%)	2523	232 (9.2%)
gatech	67	59 (88.1%)	2025	347 (17.1%)	2092	406 (19.4%)

georgetown	75	44 (58.7%)	1643	243 (14.8%)	1717	286 (16.7%)
harvard	110	69 (62.7%)	2470	397 (16.1%)	2578	466 (18.1%)
iastate	71	40 (56.3%)	1558	120 (7.7%)	1627	160 (9.8%)
illinois	16	12 (75%)	4333	581 (13.4%)	4348	592 (13.6%)
indiana	2	1 (50%)	2335	204 (8.7%)	2337	205 (8.8%)
jhu	6	1 (16.7%)	2747	311 (11.3%)	2752	312 (11.3%)
k-state	5	4 (80%)	1138	155 (13.6%)	1143	159 (13.9%)
ku	23	13 (56.5%)	1593	229 (14.4%)	1616	242 (15%)
louisiana	0	0 (0%)	1889	331 (17.5%)	1889	331 (17.5%)
louisville	0	0 (0%)	1259	60 (4.8%)	1259	60 (4.8%)
lsu	15	5 (33.3%)	1031	111 (10.8%)	1046	116 (11.1%)
memphis	4	1 (25%)	744	61 (8.2%)	748	62 (8.3%)
miami	3	1 (33.3%)	3312	713 (21.5%)	3315	714 (21.5%)

missouri	0	0 (0%)	2542	447 (17.6%)	2542	447 (17.6%)
mit	547	436 (79.7%)	2088	500 (23.9%)	2628	933 (35.5%)
msu	27	10 (37%)	3547	584 (16.5%)	3573	593 (16.6%)
ncsu	67	31 (46.3%)	2504	405 (16.2%)	2571	436 (17%)
nd	46	31 (67.4%)	1741	459 (26.4%)	1785	490 (27.5%)
ndsu	0	0 (0%)	243	42 (17.3%)	243	42 (17.3%)
njit	1	1 (100%)	1042	74 (7.1%)	1043	75 (7.2%)
northeastern	0	0 (0%)	1505	220 (14.6%)	1505	220 (14.6%)
northwestern	200	94 (47%)	1657	272 (16.4%)	1856	365 (19.7%)
odu	0	0 (0%)	1095	70 (6.4%)	1095	70 (6.4%)
ohio	5	3 (60%)	2382	340 (14.3%)	2387	343 (14.4%)
okstate	3	1 (33.3%)	1350	153 (11.3%)	1353	154 (11.4%)
olemiss	2	0 (0%)	129	14 (10.9%)	131	14 (10.7%)

oregonstate	158	108 (68.4%)	2084	202 (9.7%)	2242	310 (13.8%)
osu	13	7 (53.8%)	1299	174 (13.4%)	1312	181 (13.8%)
ou	2	1 (50%)	2101	349 (16.6%)	2103	350 (16.6%)
pitt	67	39 (58.2%)	1481	188 (12.7%)	1547	227 (14.7%)
psu	19	13 (68.4%)	1982	291 (14.7%)	2000	303 (15.2%)
purdue	14	5 (35.7%)	2554	439 (17.2%)	2568	444 (17.3%)
rice	90	76 (84.4%)	1914	274 (14.3%)	2003	349 (17.4%)
rochester	5	3 (60%)	832	132 (15.9%)	837	135 (16.1%)
rpi	74	56 (75.7%)	1209	294 (24.3%)	1283	350 (27.3%)
rutgers	61	44 (72.1%)	2597	339 (13.1%)	2657	382 (14.4%)
sc	3	0 (0%)	2159	355 (16.4%)	2162	355 (16.4%)
stanford	234	184 (78.6%)	3819	982 (25.7%)	4050	1164 (28.7%)

syracuse	1	1 (100%)	2461	394 (16%)	2462	395 (16%)
tamu	11	8 (72.7%)	1077	97 (9%)	1088	105 (9.7%)
tufts	33	18 (54.5%)	1208	198 (16.4%)	1241	216 (17.4%)
tulane	28	17 (60.7%)	231	26 (11.3%)	259	43 (16.6%)
ua	19	16 (84.2%)	1776	553 (31.1%)	1793	567 (31.6%)
uab	1	0 (0%)	1015	59 (5.8%)	1016	59 (5.8%)
uah	0	0 (0%)	1015	90 (8.9%)	1015	90 (8.9%)
uark	1	1 (100%)	715	88 (12.3%)	716	89 (12.4%)
ucdenver	9	5 (55.6%)	1099	150 (13.6%)	1108	155 (14%)
uci	16	13 (81.3%)	2743	1150 (41.9%)	2758	1163 (42.2%)
ucla	80	65 (81.3%)	1132	217 (19.2%)	1210	281 (23.2%)
uconn	4	3 (75%)	949	103 (10.9%)	952	106 (11.1%)
ucr	10	4 (40%)	1282	174 (13.6%)	1291	178 (13.8%)

ucsb	91	57 (62.6%)	1143	230 (20.1%)	1230	286 (23.3%)
ucsd	64	49 (76.6%)	2278	383 (16.8%)	2338	430 (18.4%)
udel	2	1 (50%)	1483	279 (18.8%)	1485	280 (18.9%)
ufl	951	870 (91.5%)	2946	586 (19.9%)	3897	1456 (37.4%)
uh	59	45 (76.3%)	1275	209 (16.4%)	1334	254 (19%)
uic	8	1 (12.5%)	1290	137 (10.6%)	1297	138 (10.6%)
uiowa	16	8 (50%)	1611	295 (18.3%)	1626	303 (18.6%)
umaine	14	8 (57.1%)	187	32 (17.1%)	200	40 (20%)
umass	49	29 (59.2%)	1306	225 (17.2%)	1354	254 (18.8%)
umbc	20	1 (5%)	1030	96 (9.3%)	1043	97 (9.3%)
umd	582	336 (57.7%)	1256	229 (18.2%)	1837	565 (30.8%)
umich	35	16 (45.7%)	2567	694 (27%)	2602	710 (27.3%)

umn	9	1 (11.1%)	2120	311 (14.7%)	2129	312 (14.7%)
umt	1	0 (0%)	1114	108 (9.7%)	1115	108 (9.7%)
unc	3	1 (33.3%)	1085	415 (38.2%)	1088	416 (38.2%)
unh	138	120 (87%)	1050	114 (10.9%)	1188	234 (19.7%)
unlv	10	7 (70%)	511	50 (9.8%)	521	57 (10.9%)
unm	11	2 (18.2%)	1107	207 (18.7%)	1118	209 (18.7%)
uoregon	1	0 (0%)	1763	243 (13.8%)	1764	243 (13.8%)
upenn	68	45 (66.2%)	5730	1308 (22.8%)	5796	1351 (23.3%)
usc	10	7 (70%)	1299	260 (20%)	1309	267 (20.4%)
usf	9	8 (88.9%)	1140	163 (14.3%)	1149	171 (14.9%)
usm	0	0 (0%)	1016	55 (5.4%)	1016	55 (5.4%)
usu	11	8 (72.7%)	1107	66 (6%)	1118	74 (6.6%)
uta	2	0 (0%)	1141	131 (11.5%)	1143	131 (11.5%)

utah	1	0 (0%)	2695	526 (19.5%)	2695	526 (19.5%)
utdallas	76	60 (78.9%)	515	101 (19.6%)	590	161 (27.3%)
utep	0	0 (0%)	305	36 (11.8%)	305	36 (11.8%)
utexas	36	23 (63.9%)	1570	310 (19.7%)	1605	333 (20.7%)
utsa	2	0 (0%)	1027	30 (2.9%)	1029	30 (2.9%)
uwm	286	161 (56.3%)	876	90 (10.3%)	1161	251 (21.6%)
vanderbilt	24	18 (75%)	2717	320 (11.8%)	2740	338 (12.3%)
vcu	16	5 (31.3%)	1029	101 (9.8%)	1045	106 (10.1%)
virginia	4	1 (25%)	2350	300 (12.8%)	2354	301 (12.8%)
vt	8	7 (87.5%)	1626	486 (29.9%)	1634	493 (30.2%)
washington	91	63 (69.2%)	2262	413 (18.3%)	2353	476 (20.2%)
wayne	3	2 (66.7%)	2088	92 (4.4%)	2091	94 (4.5%)
wisc	25	12 (48%)	2392	578 (24.2%)	2416	590 (24.4%)

wsu	4	0 (0%)	1334	79 (5.9%)	1338	79 (5.9%)
wustl	6	1 (16.7%)	1197	223 (18.6%)	1203	224 (18.6%)
wvu	0	0 (0%)	684	70 (10.2%)	684	70 (10.2%)
yale	161	101 (62.7%)	1225	163 (13.3%)	1384	264 (19.1%)

4.5 RQ3: Popularity of RSE Projects

Of the RSE projects / repositories found through answering RQ2, which ones are likely to be popular outside of the project’s core developers? That is, which projects are used by the community – either a university-internal community, or external/world-at-large community? We tackle this question by first engaging in a histogram analysis of the stars and forks of the 35,361 RSE projects identified in § 4.4. These visualizations can be found in Figures 8 and 7.

To do this analysis we obtained additional meta information about these projects. We obtained meta-data on each of them by using the command-line GitHub API tool with the command `gh api -H "Accept: application/vnd.github+json" -H "X-GitHub-API-Version: 2022-11-28" /repos/OWNER/REPO` where OWNER/REPO is replaced with each name of the repository listed in the supplemental material. This metadata includes the number of stars and forks for a repository, as well as the timestamp of the last push to the repository.

In light of the threshold values of six stars or six forks or more as reasonable choices for defining an RSE project as one with community, and in line with previous work [Schwartz et al. \(2024\)](#), we filter the 35,361 RSE projects accordingly. This

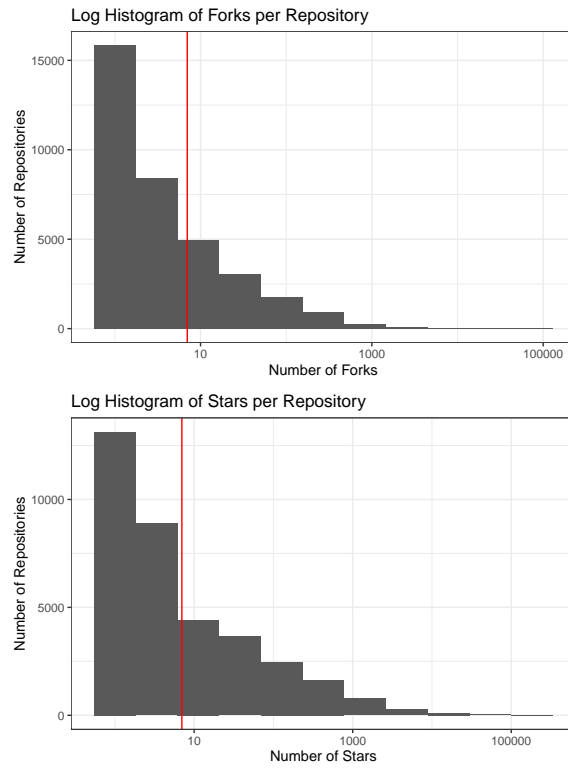


Figure 7. Histograms of the number of stars and forks per RSE repo. The red line, six in both cases, is a threshold used in [Schwartz et al. \(2024\)](#) to indicate an RSE repository has a community. These histograms seem to follow that same trend.

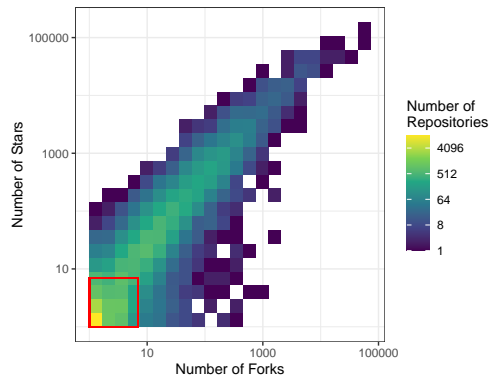


Figure 8. Matrix view of the number of repositories per star-fork pair. The Pearson correlation coefficient of Stars and Forks is 0.865, which is in line with previous research. [Yamamoto et al. \(2020\)](#)

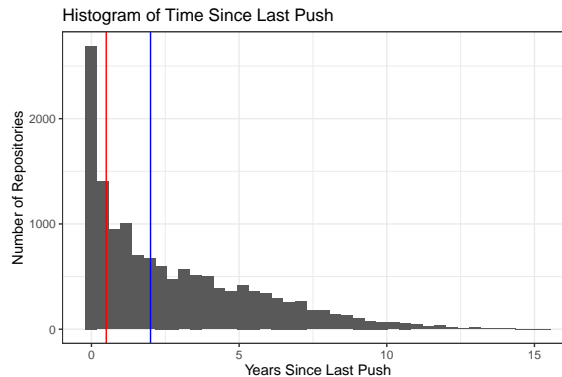


Figure 9. Histogram of the last time an RSE project with community received a push. The red line indicates the six month mark. The blue line indicates the two year mark. These lines form the arbitrary threshold boundaries we selected for “healthy,” “dying,” and, “dead” repositories, which also match prior work [Schwartz et al. \(2024\)](#).

resulted in 13,940 RSE repositories, or 39.4%, as being considered as having a community – a notably higher percentage than the national laboratory case, which had only 25% of it’s RSE repositories found to have had a community. Understanding the “why” is important future work, and will likely involve individual case studies or analysis of the number of contributors, as projects in universities may have more student volunteers over time than projects in national laboratories, thus artificially increasing the number of repositories with community sizes in the 6-16 people range. This, however, is speculation.

4.6 RQ4: Which RSE projects are active? Which are on life support?

How do they differ?

The metadata pulled from the GitHub command line API in the previous section also contains data on when the last push to a repository occurred. We examined this “last push” timestamp on the 13,940 RSE projects we classify as having a community. We found that 3,805 (27%) repositories had a push in the last six months, 3,346 (24%) repositories had their last push more than six months ago but less than two years ago, and 6,784 (49%) of repositories had their last push more than two years ago.

Using the broad buckets from Chapter III, we classify these repositories as healthy (push in last six months), dying (push more than six months ago, less than two years ago), and dead (push more than two years ago). This is visualized in Figure 9.

Of increasing interest to the community is the importance of sustainability support for RSE projects Carver, Weber, Ram, Gesing, and Katz (2022). To determine the crudest sense of what technical skills are needed for sustainability support for these projects, particularly “dying” projects, we examined the most common language used in each project according to the GitHub metadata. We examined all languages which had at least 1% prevalence in a partition. The three most common top languages in RSE repositories, across all categories of “healthy,” “dying,” and “dead” included Python, Jupyter Notebook, and C++. We show these full rankings in Tables 12, 13, and 14.

These findings suggest that the skills needed for sustainability support for the vast majority of vulnerable RSE projects is reflective of the skills used in the development of RSE projects generally. Better understanding the attributes of these projects from a code/repository analysis perspective, and how it compares with reports from RSE developers such as in Carver et al. (2022) is important future work.

4.7 Future Work

There remains ample future work in this area. In §4.3, for example, different approaches to identifying university affiliated RSE projects can certainly be explored. For example, first identifying GitHub users working at research universities via self-defined indicta of institutional affiliation (e.g., email address), and subsequently attempting to identify if any of these users’ repositories are RSE projects. Even in this work, we note that of the 147 R1 universities identified in the IPEDS data

Table 12. Top languages of healthy RSE repos:

	Language	RSE Repositories	Percentage
1	Python	1211	33%
2	C++	472	13%
3	Jupyter Notebook	261	7%
4	C	229	6%
5	JavaScript	158	4%
6	Java	142	4%
7	Go	136	4%
8	Rust	121	3%
9	R	99	3%
10	TypeScript	95	3%
11	HTML	84	2%
12	C#	76	2%
13	Shell	72	2%
14	MATLAB	45	1%
15	Julia	43	1%
16	Ruby	43	1%

Table 13. Top languages of dying RSE repos:

	Language	RSE Repositories	Percentage
1	Python	1285	40%
2	Jupyter Notebook	351	11%
3	C++	318	10%
4	JavaScript	165	5%
5	C	163	5%
6	Java	120	4%
7	HTML	100	3%
8	MATLAB	73	2%
9	R	70	2%
10	C#	65	2%
11	Shell	53	2%
12	Rust	50	2%
13	TypeScript	48	1%
14	Go	39	1%

Table 14. Top languages of dead RSE repos:

	Language	RSE Repositories	Percentage
1	Python	2084	32%
2	C++	780	12%
3	Jupyter Notebook	741	11%
4	C	479	7%
5	Java	316	5%
6	JavaScript	305	5%
7	MATLAB	181	3%
8	Matlab	163	3%
9	R	160	2%
10	HTML	156	2%
11	C#	125	2%
12	Shell	118	2%
13	Go	71	1%

only 116 were examined due to website scraping policies. These remaining unexplored 31 universities certainly warrant further investigation. Better understanding of the interactions between repositories with homes at multiple different universities is also an area of interesting work. Better identification of RSE projects using large language models is certainly needed and a major threat to the validity of our work. Lastly, there are many further analyses of the projects identified as RSE that can be done to help identify characteristics of these repositories compared to other types of open source software projects.

4.8 Conclusion

In this chapter we applied two different approaches to obtain a set of open source GitHub repositories at all R1 US research universities which allowed website scraping. We then used a GitHub repository’s description and README file in a prompt to ChatGPT to identify 35,361 research software engineering repositories of the 193,921 repositories identified as having some possible nexus to a research

university. ChatGPT agreed 77%-90% of the time with a human's classification of a RSE.

Of these 35,361 RSE projects, it's plausible that at least 13,940 (39.4%) have some community use outside the core development team. Of these repositories with a sufficient level of community engagement, only 3,805 (27%) had received a push in the last six months, suggesting many RSE projects could use sustainability support. Across all RSE projects linked to a research university, the top languages were Python, C++, and Jupyter Notebook.

CHAPTER V

DISCUSSION: COMPARISON BETWEEN NATIONAL LABORATORIES AND UNIVERSITIES AND CHARACTERISTICS OF RSE PROJECTS

This chapter contains previously unpublished material with co-authorship. All figures, text, experimental design and code which appear were developed by me. Dr. Steve Fickas and Dr. Boyana Norris assisted with editing and discussions in an advisory capacity.

5.1 Chapter Summary

In this chapter we compare and contrast the characteristics of RSE repositories at research universities vs national laboratories, finding that a random forest classifier could not accurately distinguish the two. We then compared attributes, finding that underlying correlations around time and popularity formed latent factors in our proposed taxonification of RSE projects. However, we also found that there are no sharp cutoffs in our data, with multiple unsupervised clustering algorithms and dimensionality reduction techniques all pointing to our RSE data as a singular high-dimensional blob.

5.2 Properties, characteristics, and varieties of research institution linked GitHub repositories at national laboratories and research universities

To better understand RSE projects we first need to understand some of their characteristics. In Chapters [III](#) and [IV](#) we examined attributes like the number of stars, forks, and time since the repository was last pushed to. We wanted to expand this analysis to additional characteristics and directly compare and contrast RSE repositories in national laboratories with RSE repositories linked to R1 universities.

To do this analysis we obtained additional meta information about these RSE projects. We obtained meta-data on each of the RSE projects we

found associated with national laboratories and R1 universities by using the command-line GitHub API tool with the command `gh api -H "Accept: application/vnd.github+json" -H "X-GitHub-API-Version: 2022-11-28" /repos/OWNER/REPO` where OWNER/REPO is replaced with each name of the repository listed in the supplemental material.

We then mined and cleaned information about repositories from these large meta-data files, which are named and described below. Data was successfully obtained from 8,066 RSE repositories affiliated with national laboratories discussed in Chapter III and 35,252 RSE repositories affiliated with R1 universities discussed in Chapter IV. Given our findings of 8,174 national laboratory linked RSE repositories in Chapter V and 35,361 research university linked RSE repositories in Chapter IV, this corresponds to 98.68% and 99.69% success rates of obtaining cleaned data. Instances when the data was not cleaned for some repositories had to do with malformed or unresponsive API results for reasons that remain unclear in all cases. However, we suspect it's overwhelmingly due to repositories which were either made private or deleted from GitHub entirely.

Attributes obtained for each cleaned repository are below. The key is “`attribute_name` (default value on repository creation): description.”

- `language` (none): The primary programming language GitHub assigns to the project based off the source code in the repository.
- `license` (none): The legal license outlining acceptable for use by other developers.
- `size_in_kb` (0): The size of the repository in Kilobytes.
- `forks` (0): The number of forks of the repository.

- `stars` (0): The number of stars of the repository.
- `time_since_creation_months` (0): The number of months since the repository was created, rounded to two decimal places.
- `time_since_last_push_months` (0): The number of months since the repository was last pushed to, rounded to two decimal places.
- `has_projects` (true): Whether the GitHub projects feature has been enabled or not.
- `has_downloads` (true): Whether the GitHub code downloads feature has been enabled or not.
- `has_issues` (true): Whether the GitHub issues feature has been enabled or not.
- `has_wiki` (false): Whether the GitHub wiki feature has been enabled or not.
- `has_pages` (false): Whether the GitHub pages feature has been enabled or not.
- `has_discussions` (false): Whether the GitHub discussions feature has been enabled or not.
- `is_archived` (false): Whether the repository has been archived.

In the next few subsections we do a side-by-side comparison of these attributes with RSE repositories in a university context and a national laboratory context:

5.2.1 Language. GitHub assigns languages to repositories using the code at <https://github.com/github-linguist/linguist>. A cursory view of RSE repositories with “None” listed as the primary language revealed that many of these repositories tended to have heavily documented code. Due to the vigorous documentation, linguist

Table 15. Percentage differences in languages of RSE repositories at national laboratories vs research universities. Only languages with more than 1% prevalence in either labs or universities are shown.

Language	National Laboratory	Research University	Difference	Magnitude Difference
C/C++	22.12	12.33	9.78	1.793X
Jupyter Notebook	7.54	15.26	-7.73	-2.025X
None	11.93	8.47	3.46	1.408X
R	1.57	4.62	-3.05	-2.936X
MATLAB	1.18	4.18	-3.00	-3.547X
Shell	3.73	1.47	2.26	2.535X
Java	2.52	4.55	-2.03	-1.808X
Fortran	2.33	0.40	1.93	5.827X
Python	25.17	26.99	-1.82	-1.072X
JavaScript	2.69	4.29	-1.60	-1.593X
HTML	3.29	3.95	-0.67	-1.203X
Ruby	1.28	0.66	0.62	1.941X
Julia	1.04	0.44	0.60	2.382X
CSS	1.04	0.46	0.58	2.263X
Go	0.59	1.02	-0.42	-1.711X
TeX	1.05	0.73	0.33	1.446X
C#	1.56	1.53	0.03	1.02X

was unable to determine which programming language was primary – particularly if the repository was multilingual. This suggests that National Laboratories have better documented RSE projects, and this is an interesting area of future work to more vigorously analyze.

5.2.2 License. On examining a random selection of national laboratory licenses, “Other” typically means a custom license reserving rights to the laboratory which wrote the code. Universities, by contrast, rarely have licenses reserving intellectual property rights to the institution.

5.2.3 Size in KB. All in all, the shape of the distributions and overall sizes are not all that different between the two types of institutions for repository size.

Table 16. Percentage differences in software licenses of RSE repositories at national laboratories vs research universities.

License	National Laboratory	Research University	Difference	Magnitude Difference
other	27.13	6.30	20.82	4.304X
no license	42.44	59.73	-17.30	-1.408X
mit	9.79	17.17	-7.37	-1.753X
bsd-3-clause	9.03	1.83	7.19	4.919X
gpl-3.0	2.43	5.64	-3.21	-2.321X
bsd-2-clause	1.81	0.54	1.27	3.358X
apache-2.0	3.84	5.10	-1.26	-1.328X
lgpl-2.1	0.76	0.14	0.61	5.214X
agpl-3.0	0.16	0.56	-0.40	-3.472X
lgpl-3.0	0.68	0.40	0.28	1.705X
cc0-1.0	0.17	0.38	-0.21	-2.201X
unlicense	0.09	0.26	-0.17	-2.931X
cc-by-sa-4.0	0.01	0.09	-0.08	-7.333X
isc	0.02	0.08	-0.06	-3.4X
epl-1.0	0.07	0.03	0.04	2.176X
cc-by-4.0	0.14	0.16	-0.03	-1.213X
bsd-3-clause-clear	0.01	0.04	-0.02	-3.083X
gpl-2.0	1.18	1.16	0.02	1.018X
mpl-2.0	0.21	0.22	-0.01	-1.062X
bsl-1.0	0.01	0.02	-0.01	-1.917X
afl-3.0	0.01	0.00	0.01	4X

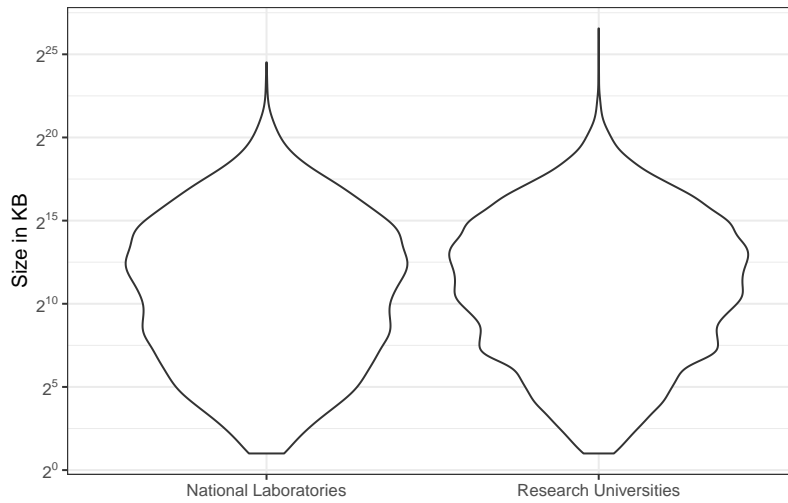


Figure 10. Violin plots of the size of RSE repositories in kilobytes at research universities and national laboratories.

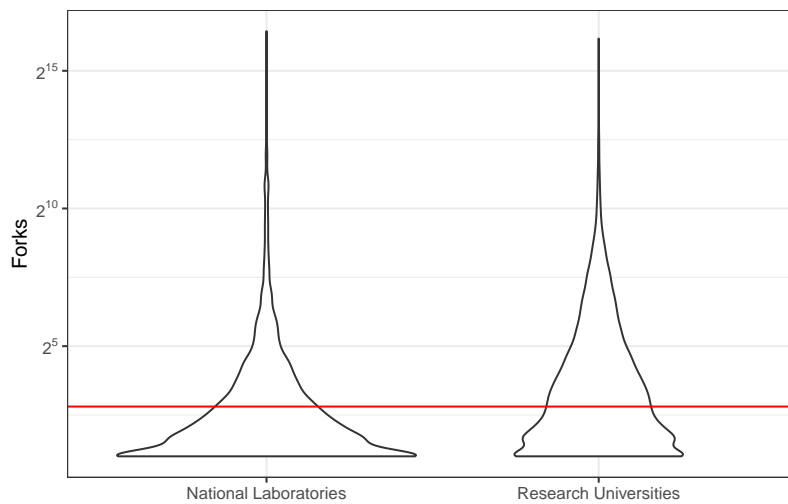


Figure 11. Violin plots of forks of RSE repositories at research universities and national laboratories. The red line is 6, our cutoff in previous chapters for community.

Table 17. Summary statistics about the size of RSE repositories linked to National Laboratories and Research Universities.

	National Laboratories	Research Universities
Min.	0.00	0.00
1st Qu.	106.25	169.00
Median	1530.50	2000.00
Mean	52369.55	60765.71
3rd Qu.	16227.50	17543.25
Max.	23961428.00	98826468.00

Table 18. Summary statistics about Forks in RSE repositories linked to National Laboratories and Research Universities.

	National Laboratories	Research Universities
Min.	0	0
1st Qu.	0	0
Median	1	1
Mean	58.93	39.90
3rd Qu.	4	8
Max.	88654	73877

5.2.4 Forks. National laboratories seem to have a higher ratio of highly forked repositories versus non-forked repositories when compared to universities, but nothing terribly different.

Table 19. Summary statistics about Stars in RSE repositories linked to National Laboratories and Research Universities.

	National Laboratories	Research Universities
Min.	0	0
1st Qu.	0	0
Median	1	2
Mean	168.15	180.80
3rd Qu.	4	20
Max.	175949	182317

5.2.5 Stars. Similarly with stars, national laboratories seem to have a higher ratio of highly starred repositories versus non-starred repositories when compared to universities, but nothing terribly different.

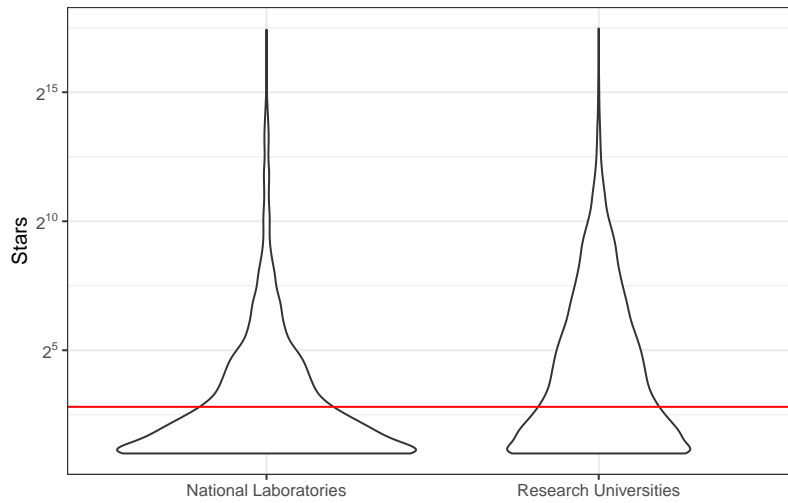


Figure 12. Violin plots of stars of RSE repositories at research universities and national laboratories. The red line is 6, our cutoff in previous chapters for community.

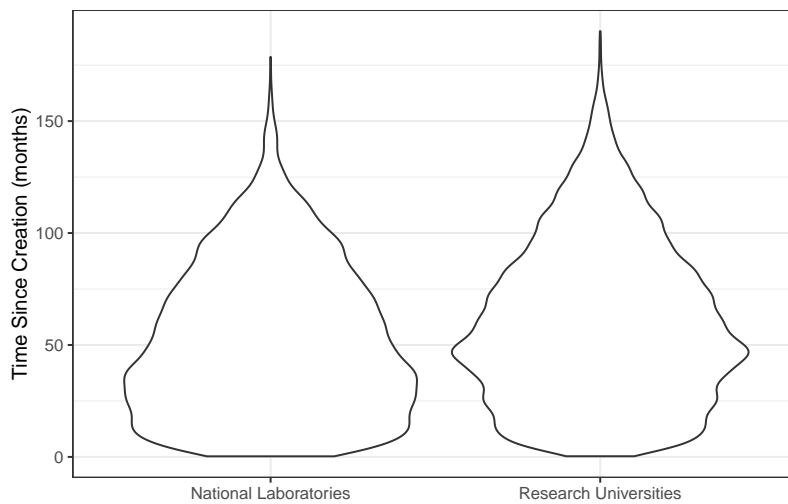


Figure 13. Violin plots of the time since a repository was created of RSE repositories at research universities and national laboratories.

Table 20. Summary statistics about the time passed since repository creation, in months, in RSE repositories linked to National Laboratories and Research Universities.

	National Laboratories	Research Universities
Min.	0.27	0.30
1st Qu.	24.18	31.49
Median	46.23	54.84
Mean	50.95	59.48
3rd Qu.	74.36	83.14
Max.	178.62	190.13

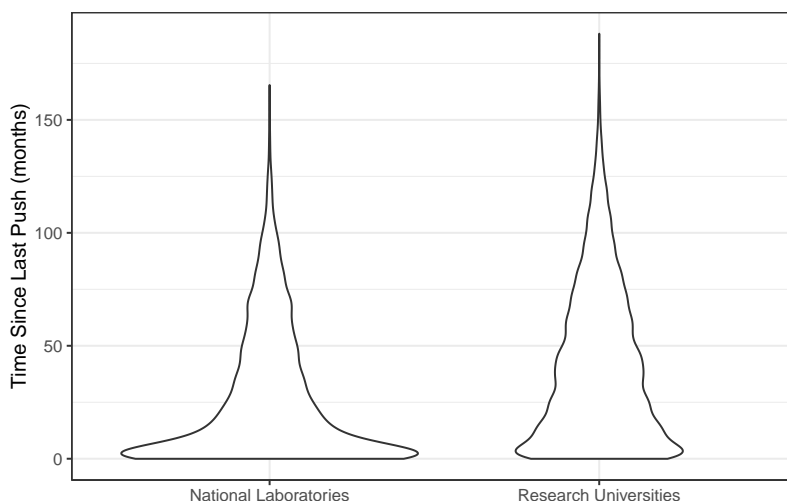


Figure 14. Violin plots of the last time a repository had a push of RSE repositories at research universities and national laboratories.

5.2.6 Time Since Repository Creation. Now time since repository creation is an area where there are interesting differences. 50 months ago was right around when COVID started. In the research university case, the rate of RSE repository creation fell after COVID started and it still has not recovered. By contrast, at national laboratories the rate held steady or even slightly increased for a bit before starting to taper back down just a tad under pre-COVID rates.

Table 21. Summary statistics about the time since the last push, in months, in RSE repositories linked to National Laboratories and Research Universities.

	National Laboratories	Research Universities
Min.	-0.02	-0.02
1st Qu.	2.52	11.75
Median	15.58	33.97
Mean	27.09	40.97
3rd Qu.	44.78	63.14
Max.	165.41	188.14

5.2.7 Time Since Last Push. This is another point of slight distinction between universities and national laboratories: national laboratories have people pushing to their repositories much more often than in the university case.

Table 22. Summary statistics about various features and properties of RSE repositories linked to National Laboratories and Research Universities.

	National Laboratories	Research Universities	Difference
has_pages (false)	13.85	8.21	5.64
has_issues (true)	91.33	96.72	5.39
has_discussions (false)	3.68	2.78	0.90
has_wiki (false)	93.23	94.10	0.87
has_projects (true)	96.96	97.27	0.31
is_archived (false)	2.00	2.14	0.14
has_downloads (true)	99.73	99.71	0.02

5.2.8 Has {feature} or Is {property}.

5.3 Classifying RSE projects as lab-related or university-related

We were curious as to whether we could classify RSE projects as lab-related or university related, specifically by using decision trees for its straightforward interpretability.

To do this, we wanted to clean the meta-data described above in the following ways:

- The \log_2 of a repository’s size, stars, forks, and open issues were taken.

- All categorical values were one-hot dummy-coded.
- Every value was then scaled to have a mean of 0 and unit variance.

We then used 10-fold cross-validation on a selected stratified random sample of 385 instances, which we did for purposes of training time and resources. 385 samples has a margin of error of $\pm 5\%$ on a 95% confidence interval, which we considered good enough for this experiment.

The model was unable to differentiate between the two classes of national laboratory RSE projects and university RSE projects, which we found encouraging, to the point that the model’s best tree was not a tree at all: just predict one class uniformly.

This is encouraging, since it suggests that there is no meaningful predictive difference between university and national laboratory RSE projects when considering all factors combined, even if there are slight differences in overall distribution by individual factor.

5.4 Correlations

We now switch to considering the combined universe of RSE projects, both lab-related and university-related. We combined our list of repositories into one unified set of RSE repositories. We then cleaned this data in the same manner as we did in §5.3. Using a Pearson magnitude of 0.2 as our cutoff (moderate correlation), we found the following relationships were correlated in Table 23.

Table 23 implies several notable observations:

- Stars and forks are highly correlated, which follows the trend for open source projects generally.

Table 23. Correlations of metadata attributes of RSE repositories.

Var1	Var2	Value	Value
forks	stars	0.91	0.91
forks	open_issues	0.71	0.71
stars	open_issues	0.70	0.70
time_since_creation	time_since_last_push	0.64	0.64
has_projects_True	has_wiki_True	0.54	0.54
stars	license_None	-0.39	0.39
forks	license_None	-0.36	0.36
open_issues	license_None	-0.35	0.35
open_issues	has_discussions_True	0.33	0.33
forks	time_since_creation_months	0.31	0.31
open_issues	time_since_last_push	-0.31	0.31
stars	has_discussions_True	0.28	0.28
forks	has_discussions_True	0.27	0.27
forks	size_in_kb	0.27	0.27
stars	size_in_kb	0.26	0.26
stars	time_since_creation	0.26	0.26
open_issues	size_in_kb	0.26	0.26
size_in_kb	language_None	-0.24	0.24
stars	time_since_last_push	-0.22	0.22
open_issues	time_since_creation	0.22	0.22
time_since_last_push	license_None	0.22	0.22

- Similarly, open_issues are correlated with stars and forks, although to a lesser degree, which also corresponds with the existing literature’s findings about open source projects.
- Of the 21 correlated relationships in Table 23, 17 contain stars, forks or open_issues as a variable. Taking stars, forks, and open_issues as proxies for a project popularity/community, this implies that one of the biggest underlying latent variables is a project’s popularity.
- The four correlations were found which did not contain stars/forks/open_issues were:

* `time_since_creation` and `time_since_last_push`

This is perhaps unsurprising; the time since the last push is always going to be shorter than the duration of the project. More to the point of the correlation, we know that many projects (roughly two-thirds) are dead or dying. As time progresses, the lifespan of many projects will follow some distribution, and the bulk of projects will have a last-push date similar to $\text{time_since_creation} + \{\text{median project lifespan}\} \simeq \text{time_since_last_push}$.

* `has_projects=true` and `has_wiki=true`

This correlation is surprising, mostly due to the observation that this is a positive correlation but wikis and projects have discordant default values of false and true, respectively. Upon further investigation, it appears that wikis were, at one point, enabled by default. As new repos come online, the correlation with active projects and wikis becomes more notable, which is why this correlation may pop more than other binary features.

* `size_in_kb` and `language=None`

While forks and stars did not correlate with any specific language on their own (including “None”) with our threshold, these attributes (along with `license=None`) were the top correlating features with `language=None`. In other words, as a project grows larger in digital size and is more popular (e.g., stars and forks), it’s more likely that a clear language identifiable by the GitHub linguist software is selected.

* `time_since_last_push` and `license=None` While the number of commits was not included in these meta-statistics, it’s plausible that more actively worked on repositories (i.e., those with more recent push dates) are more

likely to have licenses. This tracks with stars and forks positively correlated with `license≠None`.

In short, there are very few surprises or detours from the established literature on attribute correlation in open source software repositories generally, with the one question mark of wiki-projects correlation.

5.5 Taxonomy of RSE projects

With this background in mind, we submit a possible taxonomy for the classification of RSE projects:

5.5.1 Classification. The key differentiating latent factors of RSE projects are “popularity,” and “time,” through which we are able to categorize RSEs:

- Personal projects

These projects are typically characterized by 0 stars and forks by users other than the repository owner. Their active lifespan is usually less than 12 months.

- Community projects

These projects are typically characterized by more than 6 stars and forks by users other than the repository owner. Their active lifespans range drastically, with a majority of the projects in the universe of identified RSE repositories dead or dying and in need of sustainability support.

- Small team projects

These projects are characterized by stars and forks by users other than the repository owner between 1 and 5. Their active lifespans are bifurcated between one year or less, or continuous contributions.

We also attempted to use clustering to separate RSE projects programmatically. Perhaps foreshadowed by our inability to split RSE projects by national laboratories vs universities, we found that, across multiple clustering algorithms, from hierarchical clustering via Dendrograms to DBSCAN/OPTICS, all over multiple hyper-parameters, our data is one single cluster blob. PCA and FAMD analysis similarly suggested that there was little in the way of dimensionality reduction to be done.

With GitHub having recently released new API documentation around time-series data, incorporating change over time of RSE repositories is an encouraging avenue for more features that can help provide internal points of differentiation.

5.5.2 Nearby galaxies. In the entire universe of open source software, RSE repositories form one galaxy. To help us better understand nearby galaxies, we turn to a manual factor analysis of repositories labeled during the manual identification of RSE repositories for large language model verification in Chapter IV. During that process, the following categories of repositories linked to universities were identified:

- Academic repositories: These repositories primarily contain LaTeX templates, such as dissertation/thesis formats for a particular department or university, journal/conference starterpacks, etc.
- Auxiliary repositories: These repositories don't have research-specific content per-se, but may contain information about how to do thing like log into a HPC computer system, connect through a firewall, or install and load software on a specific node.

- Class repositories: These repositories are created by students (or sometimes instructors or workshop facilitators) for the purpose of learning. There can sometimes be fuzzy overlap between class repositories and RSE repositories for final projects in graduate-level classes.
- Hackathon repositories: Created for developing code in the context of a hackathon.

Noting these repositories were all associated with research universities helps contextualize repositories which may not meet an RSE definition, but still may be encountered when engaging with repositories linked to research institutions. Some of these galaxies have had their own in-depth research done, such as on Hackathons in the dissertation done by Dr. Caroline Hardin while at UW-Madison. Understanding how these other galaxies are similar to, or different from, each other also remains future work.

5.6 Discussion and Conclusion

In this chapter we analyzed the most obtainable attributes from GitHub on our collection of RSE projects. From this we found that RSE projects are nearly identical in terms of being able to correctly classify one as a national lab RSE project or research university RSE project from the features we investigated.

However, we did find a few points of differentiation in the distributions between the two: slight differences in project creation rates around COVID, ongoing greater productivity in terms of repository pushes at national laboratories, different licensing rates – namely, national laboratories are somewhat more fickle about copyrighting and customized licensing of their code, and slightly different usage rates of Jupyter Notebooks and C++.

We also correlated the features of a unified set of RSE repositories. In correlating these features, two undergirding themes arose as key features for discriminating among repositories: the importance of popularity and time.

We proposed three categories for binning RSE projects based on their popularity and characterization of time-related events. However, we also note that there are no hard-and-fast delineators of clusters based on our treatment of the data.

Our findings around unsupervised clustering and dimensionality reduction suggests that there are latent factors we are not capturing. This categorization is limited. Due to the influence of popularity and time, future work in this area certainly involves mining GitHub's recently documented API endpoints for commit activity and other time-series data to better understand how these RSE projects change over time.

CHAPTER VI

DEPENDENCY RELATIONSHIPS

This chapter contains previously unpublished material with co-authorship. All figures, text, experimental design and code which appear were developed by me. Dr. Steve Fickas and Dr. Boyana Norris assisted with editing and discussions in an advisory capacity.

6.1 Chapter Summary

We posit a policy implication of our work: maybe the government should pay attention to heavily used dependencies and provide sustainability or cybersecurity support of RSE projects. This then leads to the questions of, “What are the key dependencies (third-party packages) of RSE projects, empirically speaking?” and “What dependencies, if compromised, could impact large numbers of RSE projects?”

In this chapter we outline an approach to understanding dependency reliance in RSE repositories. We then analyze the RSE repositories identified in Chapters [III](#) and [IV](#).

6.2 Methodology

Our methodology consists of the following steps:

1. Find a sufficiently large collection of RSE repositories to analyze.

In many ways, the toughest part of our methodology has already been answered in the previous chapters: how to select an appropriate set of repositories?

In our case, we turn to the RSE repositories we gathered from national laboratories and research universities as a single, consolidated set.

2. Obtain dependencies of each RSE repository.

There are multiple ways to do this. One way is to use the build system for language-specific components as a resource. In Python, this would be using PIP to identify dependencies.

We want to be more language agnostic. Instead, we turn to a relatively new feature of GitHub, software bills of materials (SBOMs).

These SBOMs, analogous to bills of materials of physical goods going through customs, list dependencies and components of the software product. Partly prompted by federal Executive Order 14028 (2022), which mandated SBOMs for some types of exported software, GitHub allows for the generation of these SBOMs. While not every project will have one, we expect there will be enough SBOMs to create a network.

3. Create a dependency network (i.e., a digraph) through mining the SBOMs mined in step 2 for all RSE repositories identified in step 1.
4. From this network created in step 3, use existing network techniques to identify key nodes. This can be as simple as counting inbound edges of each node to network flow techniques to bleeding edge work on key node detection like ?. We will begin by simply counting inbound edges on nodes in our graph.

6.3 Results

In this section we report the findings of our methodology.

6.3.1 Step 1 and 2: Find a representative set of RSE repositories and obtain their dependencies. We used the combined set of national laboratory and research university RSE repositories discussed in Chapters III and IV, a total of 43,399 repositories.

We then used the GitHub command line tool to make the following API call on each repository: `gh api -H "Accept: application/vnd.github+json" -H "X-GitHub-API-Version: 2022-11-28" /repos/OWNER/REPO/dependency-graph/sbom` with OWNER/REPO replaced with the appropriate full repository name.

From this, we found a total of 10,832 repositories with both a SBOM and at least one listed dependency in the SBOM, or 24.96% of all RSE repositories in our set of national laboratory and research university RSE repositories.

We quickly observed that the dependency relationships in the GitHub repositories' SBOMs relied on build system links, and no dependency which wasn't incorporated as part of a fixed set of build systems was included in any SBOM. This is disappointing, as it means projects which heavily rely on makefiles (e.g., C and C++ files) are not explicitly included. Still, in this exploratory analysis we were able to ask and answer the following questions about "first order" dependencies¹ of RSE repositories.

- What is the distribution of "first order" dependencies per repository?
- What is the distribution of "first order" dependencies by package manager?
- What is the distribution of package managers per repository?
- What is the distribution of "first order" dependencies per repository, stratified by package manager?
- Of the set of immediate "first order" dependencies, how often is each used by an RSE repository?

¹I.e., the immediate neighbor nodes of an RSE repository in a dependency graph.

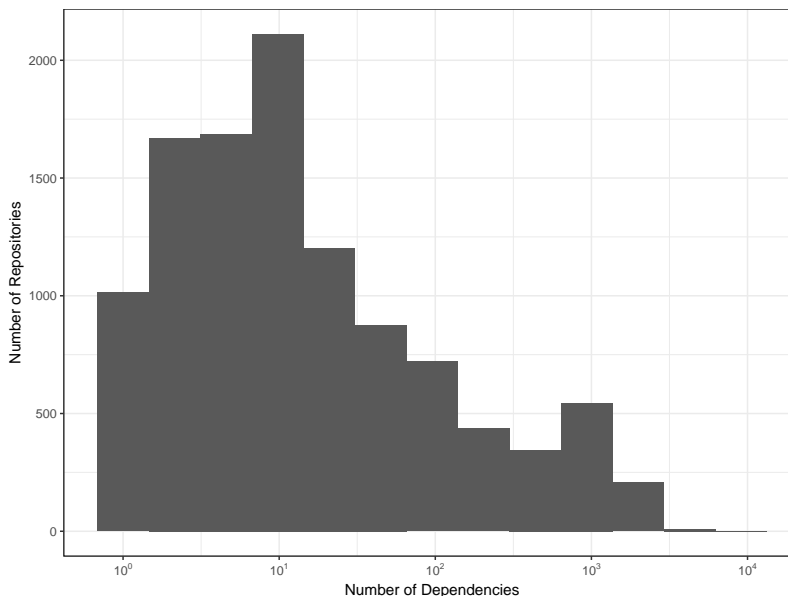


Figure 15. Histogram of the number of repositories associated with each dependency.

Each of these questions are answered in the below subsections. Note that we consider each dependency by general name only, independent of version. For example, the python package numpy is considered as one dependency of the pip package manager, even though there are over 115 distinct versions of numpy listed across all RSE repository dependencies.

6.3.1.1 What is the distribution of “first order” dependencies per repository? We wanted to investigate the distribution of dependencies per repository. This was first explored by plotting the histogram displayed in Figure 15. Noting the need for a log transformation to better display the data, we explored whether taking a log would result in a linear relationship. We found that, for repositories with less than or equal to 125 dependencies, which composed 85% of all RSE repositories, we could model this analytically by fitting a logged quadratic piecewise regression to the data:

percentage_of_all_RSE_repositories_with_exactly_x_number_of_dependancies(x) =

$$\left\{ \begin{array}{l} \exp(-2.164392128 - 0.169745390x + 0.001622215x^2) \\ x \leq 20; R^2 = 0.995; \text{All coefficients significant at the } p < 0.01 \text{ level} \\ \exp(-3.294857880 - 0.105544700x + 0.001176225x^2 - 0.000004916x^3) \\ 20 < x \leq 200; R^2 = 0.819; \text{All coefficients significant at the } p < 0.01 \text{ level} \end{array} \right.$$

This is plotted in Figure 16. Tests of linearity after the log is taken all pass; residuals are evenly distributed, etc. However, we were concerned about over-fitting, and whether taking the log was an appropriate choice. To gain further assurance, we also examined the Cullen and Frey graphs for both components of our pieces in the function. These can be found in Figure 17. This provides some reassurance that, for repositories with 20 or fewer dependencies (65-66% of all RSE repositories), the distribution is, indeed, exponential. For repositories with more than 20 dependencies but less than 125 (20% of all repositories), there is still a baseline exponential pattern, but influenced by an underlying normal distribution. Repositories with more than 125 dependencies are relatively infrequent (15% of all repositories) and their distributions are difficult to model given the small integer nature of their prevalence in this sample (no more than 6 repositories had the same number of dependencies above 125, with the majority being sui generis or duo generis).

We also stratified by the dependency manager. While the degree of curvature was distinct, with slightly different distributions suggested when stratified by manager – for example, for pip an underlying lognormal distribution close to the normal distribution was suggested by the Collen and Frey plot– the overall pattern of

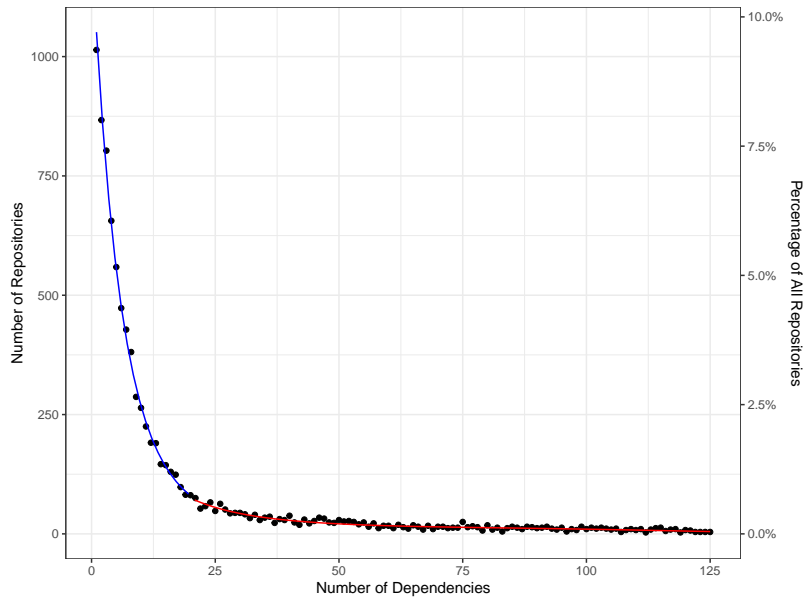


Figure 16. Percentage of all RSE repositories with a given number of dependencies. Fitted trendline in blue and red.

relatively few dependencies per repository held, with a rapid decay in repository count as frequency of dependencies increased.

Bottom line: over half of all repositories had fewer than 10 dependencies.

6.3.1.2 What is the distribution of “first order” dependencies by package manager? There are 11 package managers found in the RSE SBOM files. These package managers, primary affiliated language, and number of dependencies are found in Table 24. Surprisingly, the language with the most dependencies is not Python, as we might expect from the overview of RSE languages discussed in Chapter V.

6.3.1.3 What is the distribution of package managers per repository? In Chapter V we discussed the prevalence of language by repository, of which GitHub assigned only one. However, we know in reality that projects may involve multi-lingual code paradigms. When we exclude GitHub actions from our analysis, we find that 91.41% of all RSE repositories contain dependencies from a

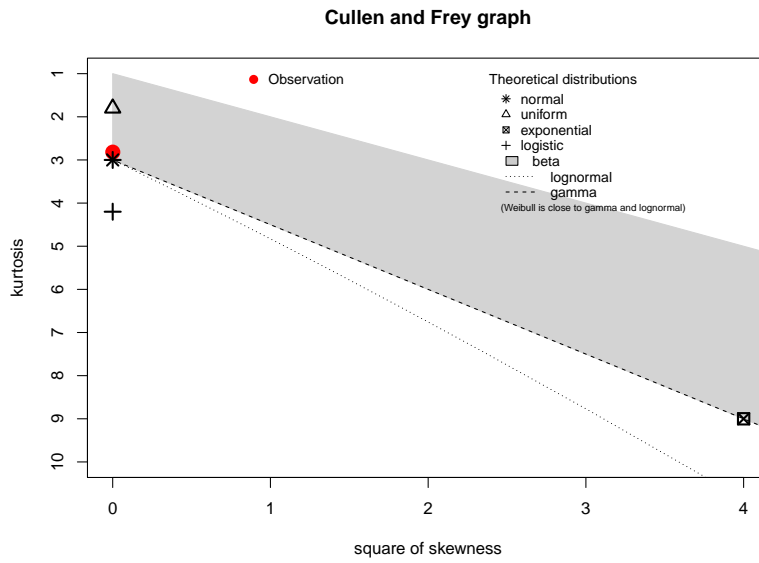
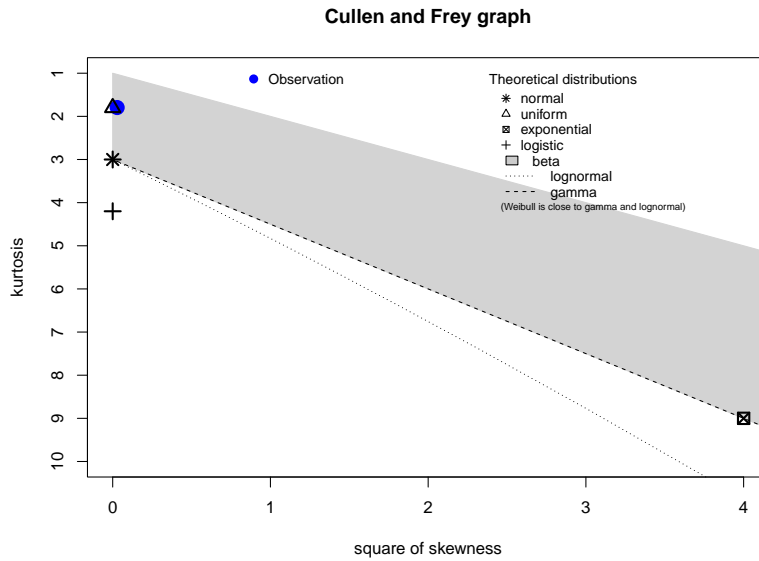


Figure 17. Cullen and Frey graphs for the logged data shown in Figure 16, with distribution observations color-coded as matching red and blue.

Table 24. Number of Dependencies by Package Manager

Primary Language	Manager	Sum(Dependency X * Number of RSE Repositories with Dependency X)	(%)	Unique Dependencies	(%)
Node.js	npm	1,124,110	81.35	25,320	46.71
Python	pip	12,3661	8.95	8,442	15.57
Rust	rust	39,907	2.89	4,683	8.64
Maven	maven	13,877	1.00	4,603	8.49
Go	go	17254	1.25	3039	5.61
Ruby	rubygems	29,164	2.11	2,767	5.10
.Net	nuget	6,244	0.45	1,839	3.39
GitHub	actions	19,014	1.38	1,672	3.08
PHP	composer	6,305	0.46	1,241	2.29
Dart	pub	2,075	0.15	476	0.88
Swift	swift	198	0.01	127	0.23

single build system, 7.71% have dependencies from two build systems, and only 0.88% use three or more build systems. Of those very few which used three or more build systems, a full 25% included maven, npm, and pip. These projects often processed or obtained data in Java, used machine learning in Python, then displayed the results in a website which used JavaScript libraries.

Of the 7.71% repositories with dependencies from two build systems, 76.8% included npm, 61.4% included pip, and a full 41.7% were npm and pip. Once again, these projects are usually characterized by python data processing paired with web application presentation involving JavaScript.

6.3.1.4 Of the set of immediate “first order” dependencies, how often is each used by an RSE repository?. We found that, across all RSE repositories with SBOMs, there were 54,209 unique dependencies (version agnostic). The vast majority of these dependencies were used only by one repository. Specifically, 26,448 dependencies, 48.8% of the total, were needed by only one repository.

This distribution was exponential-like. 48.8% of dependencies were needed by only one repository, 83.0% were needed by 10 or fewer repositories, and 95.4% of all dependencies were needed by 100 or fewer RSE repositories.

Relatedly, there were a small number of dependencies used by many RSE repositories. While we report all dependencies used by more than 10 RSE repositories (of which there are 9,190, or 17%) in Table 28 in the appendix, we highlight here in Table 25 the top 10 dependencies by build system.

Table 25. Top 10 RSE Dependencies by Build System. The % is in reference to the number of repositories which utilize the corresponding build system manager.

Dependency	Used by # of Repositories	Used by % of Repositories which use {actions, pip, npm, etc.}
actions:actions/checkout	3721	93.42
actions:actions/setup-python	1407	35.33
actions:actions/upload-artifact	873	21.92
actions:actions/cache	579	14.54
actions:codecov/codecov-action	422	10.60
actions:docker/login-action	346	8.69
actions:actions/setup-node	332	8.34
actions:actions/download-artifact	331	8.31
actions:pypa/gh-action-pypi-publish	321	8.06
actions:docker/build-push-action	305	7.66
composer:phpunit/phpunit	83	52.87

composer:php	61	38.85
composer:psr/log	57	36.31
composer:symfony/polyfill-mbstring	54	34.39
composer:symfony/console	51	32.48
composer:phpunit/php-code-coverage	47	29.94
composer:phpunit/php-file-iterator	47	29.94
composer:phpunit/php-text-template	47	29.94
composer:phpunit/php-timer	47	29.94
composer:sebastian/diff	47	29.94
composer:sebastian/exporter	47	29.94
composer:symfony/yaml	47	29.94
go:golang.org/x/sys	195	65.22
go:golang.org/x/net	158	52.84
go:golang.org/x/text	149	49.83
go:gopkg.in/yaml.v3	133	44.48
go:golang.org/x/crypto	130	43.48
go:github.com/pkg/errors	126	42.14
go:github.com/stretchr/testify	124	41.47
go:github.com/golang/protobuf	122	40.80
go:google.golang.org/protobuf	119	39.80
go:github.com/davecgh/go-spew	115	38.46
maven:junit:junit	404	48.21
maven:org.apache.maven.plugins:maven-compiler-plugin	342	40.81

maven:org.apache.maven.plugins:maven-surefire-plugin	190	22.67
maven:org.apache.maven.plugins:maven-javadoc-plugin	149	17.78
maven:org.slf4j:slf4j-api	134	15.99
maven:org.apache.maven.plugins:maven-jar-plugin	131	15.63
maven:org.apache.maven.plugins:maven-source-plugin	125	14.92
maven:org.apache.maven.plugins:maven-shade-plugin	115	13.72
maven:com.google.guava:guava	108	12.89
maven:org.apache.commons:commons-lang3	108	12.89
npm:debug	1319	59.87
npm:inherits	1306	59.28
npm:ms	1291	58.60
npm:minimatch	1240	56.29
npm:brace-expansion	1235	56.06
npm:balanced-match	1234	56.01
npm:semver	1230	55.83
npm:concat-map	1228	55.74
npm:glob	1221	55.42
npm:once	1214	55.11
nuget:Newtonsoft.Json	131	25.84
nuget:Microsoft.NET.Test.Sdk	115	22.68

nuget:PRISM-Library	88	17.36
nuget:NUnit	81	15.98
nuget:xunit	68	13.41
nuget:xunit.runner.visualstudio	65	12.82
nuget:MSTest.TestAdapter	52	10.26
nuget:MSTest.TestFramework	52	10.26
nuget:coverlet.collector	51	10.06
nuget:System.Data.SQLite.Core	38	7.50
pip:numpy	2385	48.85
pip:scipy	1489	30.50
pip:pandas	1472	30.15
pip:matplotlib	1431	29.31
pip:six	1332	27.28
pip:requests	1280	26.22
pip:python-dateutil	1125	23.04
pip:pytz	1024	20.98
pip:pillow	960	19.66
pip:certifi	953	19.52
pub:collection	31	79.49
pub:meta	31	79.49
pub:matcher	30	76.92
pub:path	30	76.92
pub:string_scanner	30	76.92
pub:async	29	74.36
pub:boolean_selector	29	74.36

pub:source_span	29	74.36
pub:stack_trace	29	74.36
pub:stream_channel	29	74.36
pub:term_glyph	29	74.36
pub:test_api	29	74.36
rubygems:rake	251	48.08
rubygems:i18n	213	40.80
rubygems:nokogiri	205	39.27
rubygems:tzinfo	204	39.08
rubygems:activesupport	196	37.55
rubygems:ffi	195	37.36
rubygems:minitest	194	37.16
rubygems:addressable	191	36.59
rubygems:concurrent-ruby	175	33.52
rubygems:public_suffix	164	31.42
rust:serde	206	58.86
rust:libc	199	56.86
rust:quote	187	53.43
rust:rand	182	52.00
rust:syn	182	52.00
rust:proc-macro2	180	51.43
rust:lazy_static	174	49.71
rust:cfg-if	172	49.14
rust:bitflags	170	48.57
rust:log	165	47.14

6.3.2 Step 3 and 4: Construct a network and find key nodes. We focus on Python dependencies in this section, due to Python being the most heavily used RSE language.

We constructed a digraph representing all RSE repositories and their dependency trees. Digraph G is defined as $G = (V, E)$ where V is the set of $x \cup \{d\}_x \cup \{d'\}_d$, \forall RSE repositories x , where $\{d\}_x$ is the set of x 's pip dependencies according to the GitHub SBOM files, and $\{d'\}_d$ is the set of each dependency of d in d 's dependency tree, according to the tool `pipdeptree`. An edge $\exists a \rightarrow b \in E$ if a directly requires b as a dependency. We disregarded version numbers for the purpose of our graph construction. The dependencies of the most recent version of a package were used to form our graph.

This resulted in a digraph with 13,673 nodes and 133,012 edges. In this graph, 4,807 nodes were RSE repositories, meaning that 35.15% of repositories in this dependency graph were RSE repositories.

We consider several metrics to analyze node importance in this graph. Namely:

- In Degree: The number of inbound edges to a node.
- RSE “Direct” In Degree: The number of inbound edges to a node, where the originator node is an RSE repository.
- Eigenvector Centrality: A measure of influence in a graph based off the Eigenvector equation of a graph’s adjacency matrix.
- PageRank: Google PageRank algorithm.
- Node Importance Flow: A customized flow algorithm we developed, which we describe in §6.3.2.1.

Table 26. Spearman correlation between various metrics of RSE dependency importance.

	eigenvector centrality	in degree	node importance flow	pagerank	rse direct in degree
eigenvector centrality	1.00	0.94	0.99	0.91	0.81
in degree	0.94	1.00	0.97	0.91	0.94
node importance flow	0.99	0.97	1.00	0.92	0.86
pagerank	0.91	0.91	0.92	1.00	0.80
rse direct in degree	0.81	0.94	0.86	0.80	1.00

In Table 26 we note that all of these metrics were highly correlated with one another.

6.3.2.1 Node Importance Flow Algorithm. Our motivation for developing this algorithm is simple: what dependencies are most heavily used by RSE repositories when examining the entire chain of dependencies? For example, perhaps there is a utility repository which few pay attention to, but which is included as a dependency for a project like numpy or other widely used library. An overlooked library which, if meddled with, could cause a security vulnerability which would echo across the entire RSE ecosystem?

To do this, we imagined water “flowing” through the dependency graph. Each node would be a source node, with one unit of flow on its own. Each dependency node would also have inbound flow from its dependencies. But, unlike real-world water, the flow would not be diluted when split by different “pipes”/edges. Nor would we put capacity restraints on the edges, as done in a normal flow problem. The most important nodes, therefore, would be the nodes with the most flow after the graph had been saturated.

While dependency graphs can often be thought of as simple trees for many situations, in reality dependency graphs are messy, and are not simple directed acyclic graphs let alone trees. Rather, they are composed of complex interrelated cycles

even setting aside (as we have) the variations of the graph depending on different dependency versions. Consequently, we had to make variations to our flow algorithm in a two-step approach:

Step 1: Put dependencies into layers.

To ensure we did not receive feedback loops through a simple traversal, we first organized nodes into different layers. We did this by doing a depth-first traversal of each RSE repository's dependency tree. If a node had a layer assigned from a previous RSE repository's traversal, the layer with the greatest depth took precedence.

Step 2: Sum cascading dependencies by layer.

Once the graph has been organized into layers, we run a simple nested for-loop: for each node n in layer i , $\text{value}(n) = 1 + \sum_{\text{parent } p \text{ of } n} \text{value}(p)$.

The computational complexity of a depth first search is $O(|V| + |E|)$. Since we perform a depth first search on an arbitrary number of RSE nodes, the complexity of Step 1 is $O(|V|^2 + |VE|)$. Since we are merely iterating over the vertices in a clever way in Step 2, the complexity for this step is $O(|V|)$. Therefore, the overall time complexity of this algorithm is $O(|V|^2 + |VE|)$. A quadratic time complexity is not ideal. However, we found that our graph was comparatively shallow; only a maximum depth of 25 layers was reached. Unsurprisingly, there was a strong linear correlation between the log of a node's flow score and the node's layer in the graph.

While we have not proved it, we believe this algorithm is a variation of the Eigenvector centrality metric, which explains why there is such a high degree of non-parametric correlation between the two for our data. However, the mechanism for calculating the score ultimately derives from a flow approach, rather than an algebraic decompositional approach.

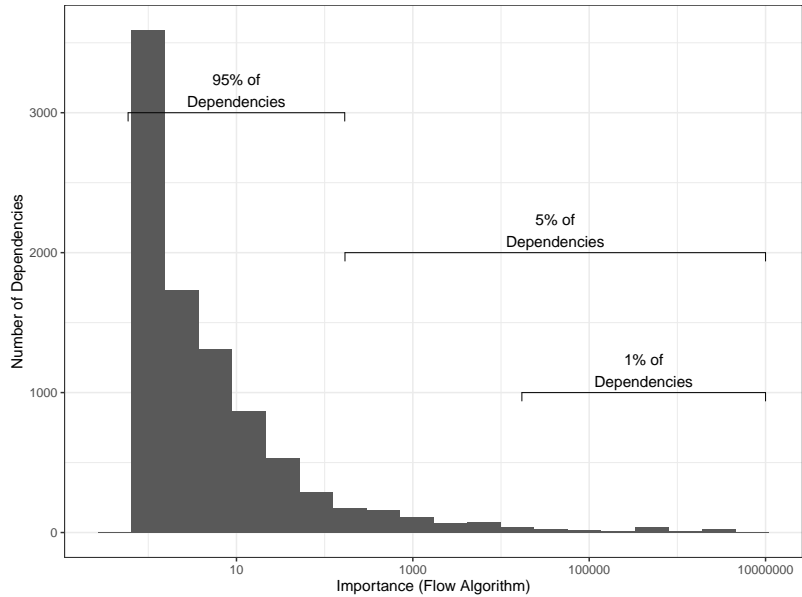


Figure 18. Distribution of dependency importance by our flow algorithm score.

Another algorithm which gives the substantially identical result (which dependency is most utilized) is to reverse the direction all edges in the graph, iterate a depth-first-search traversal over each node, and the node affiliated with the largest reachable subgraph is the most utilized dependency; an algorithm which also has an identical computational time complexity as ours.

More importantly, however, the distribution of node importance scores in Figure 18 illustrates that there are a few dependencies which have enormous influence in the dependency chain for RSE projects.

We list the top 150 dependencies. Dependencies marked with an * indicate that this dependency is not in the top 150 “first order” dependencies of an RSE repository. 72 dependencies (48%) found by our flow algorithm in the top 150 are not first order / direct dependencies.

1 packaging	19 alabaster*
2 pygments	20 sphinxcontrib-serializinghtml*
3 typing-extensions	21 sphinxcontrib-applehelp*
4 docutils	22 sphinxcontrib-devhelp*
5 attrs	23 sphinxcontrib-htmlhelp*
6 tomli	24 sphinxcontrib-qthelp*
7 babel	25 sphinxcontrib-jsmath*
8 certifi	26 sphinx
9 markupsafe	27 exceptiongroup*
10 urllib3	28 six
11 idna	29 python-dateutil
12 charset-normalizer	30 cattrs*
13 requests	31 importlib-resources*
14 typeguard*	32 jsii*
15 publication*	33 pluggy
16 jinja2	34 iniconfig
17 snowballstemmer	35 pytest
18 image-size*	36 traitlets

37 click	55 decorator
38 soupsieve	56 breathe*
39 beautifulsoup4	57 hypothesis*
40 setuptools	58 pickleshare
41 cython	59 pexpect
42 zipp	60 simplegeneric*
43 importlib-metadata	61 ipython
44 pytz	62 wheel
45 ptyprocess	63 tabulate*
46 accessible-pygments*	64 mypy-extensions
47 pydata-sphinx-theme*	65 coverage
48 wcwidth	66 pytest-cov
49 prompt-toolkit	67 numpydoc*
50 pycodestyle	68 mypy*
51 sortedcontainers*	69 smmap*
52 meson*	70 gitdb*
53 pyproject-metadata*	71 spin*
54 meson-python*	72 gitpython*

73	scipy-openblas32*	91	jsonschema-specifications*
74	scipy-openblas64*	92	jsonschema
75	numpy	93	entrypoints
76	aws-cdk-cloud-assembly- schema*	94	fastjsonschema*
77	rpds-py*	95	nbformat
78	constructs*	96	jupyter-client
79	platformdirs	97	cycler
80	aws-cdk-cx-api*	98	kiwisolver
81	jupyter-core	99	contourpy*
82	aws-cdk-region-info*	100	fonttools
83	referencing*	101	matplotlib
84	tornado	102	tzdata*
85	nest-asyncio	103	pandas
86	aws-cdk-core*	104	aws-cdk-aws-iam*
87	pyzmq	105	pyproject-hooks*
88	pillow	106	build*
89	pyparsing	107	pyparser
90	scipy	108	ffi

109 cloudpickle	126 webencodings
110 ipython-genutils	127 nbclient*
111 sphinx-design*	128 pyasn1
112 sphinx-copybutton*	129 ipykernel
113 pyyaml	130 click-plugins*
114 sphinx-basic-ng*	131 joblib
115 furo*	132 cligj*
116 sphinxcontrib- moderncmakedomain*	133 mdurl*
117 sphinxcontrib- svg2pdfconverter*	134 markdown-it-py*
118 pybind11*	135 tqdm
119 ruff*	136 werkzeug
120 tokenize-rt*	137 pyproj*
121 doit*	138 fiona*
122 cython-lint*	139 geopandas*
123 jupyterlite-sphinx*	140 networkx
124 pythran*	141 aws-cdk-aws-kms*
125 shapely*	142 filelock
	143 bleach

144 greenlet*

145 defusedxml

146 mistune

147 itsdangerous

148 pandocfilters

149 jupyterlab-pygments*

150 tinycss2*

6.4 Conclusion and Future Work

In this chapter we obtained the first order dependencies of all RSE repositories in our collection. We captured their frequency of use and presented it in this work as a full table in the appendix. We then examined Python RSE repositories and used the pip build system to generate the dependency trees for each. From these trees we created an RSE dependency graph/network, on which we could do analysis.

We used several “out of the box” algorithms to identify node importance in our graph, and also put forth our own algorithm to answer the question, “which dependencies are heavily relied upon, not only by RSE projects directly, but relied on by their dependencies as well.”

We found several dependencies which are important but not always heavily relied upon directly. This included projects with security implications, such as *publication*, a project used to publish API data in a clean way. If this dependency was found to have a security flaw, thousands of RSE projects would, eventually, be vulnerable. This has public policy implications.

As in all of these chapters, there is ample room for future work. Understanding how the dependency graph changes shape over time, expanding the graph to other languages, and doing comparative analysis with all dependency characteristics like stars and forks – particularly, which dependency/repositories may have low numbers of stars or forks yet have comparatively large footprints – all remain open avenues for exploration.

CHAPTER VII

CONCLUSION

In this dissertation we aimed to answer the following set of research questions (RQs):

RQ1 How can we find open source software repositories connected to universities and national laboratories? (Method of Development)

RQ2 Given our methodology, what is the current state of affairs? Just how many open source software repositories and projects affiliated with universities and national laboratories are out there? (Analysis of a Particular Instance/Domain)

RQ3 What are the properties, characteristics, and varieties of software projects with a nexus to these research institutions? (Generalization or Characterization, Analysis of a Particular Instance/Domain)

RQ4 How do the characteristics of repositories in the university ecosystem compare with the characteristics of repositories in the national laboratory ecosystem? (Analysis of a Particular Instance/Domain)

RQ5 How does the code in these research projects relate with and depend on each other project in the ecosystem? (Generalization or Characterization)

We briefly recap the answers in the subsections below.

7.1 Research Questions and Answers

7.1.1 RQ1: How can we find open source software repositories?.

We can find open source projects connected to universities and national laboratories through a combination of parsing affiliated human readable media for links to source repositories (e.g., crawling websites), examining build systems (e.g., mining spack),

and searching keywords on repository servers themselves (e.g., searching GitHub itself). Further, we can also use large language models to filter repositories based on definitions such as “research software engineering” (RSE).

7.1.2 RQ2: What is the current state of affairs?. We found 35,361 RSE repositories associated with 116 R1 research universities in the United States (of 147 R1 universities.) Given that overlap was minimal amongst institutions, by projecting linearly we can ballpark that there are roughly 44,800 RSE repositories in our nation’s top research university ecosystem.

We also found 8,174 repositories with an affiliation to a national laboratory. While we did not evaluate all government sources of RSE software, such as in non-DOE agencies like NASA or NIH, we suggest that DOE repositories represent roughly 60% of all government repositories based on BigQuery data of GitHub activity. While not all government activity is based in RSE work, some agencies do have a strong RSE focus. If we estimate institutions like NASA, NIH and similar others (representing an additional 27% of all non-laboratory government-affiliated repositories) have RSE activity at roughly the same rates as the national laboratory ecosystem, then this suggests there are an additional 3,680 RSE repositories unobserved directly by our work, for a total of 11,850 RSE repositories directly affiliated with a US government agency or subsidiary.

Putting it all together, we assert that there are roughly 57,000 RSE repositories affiliated with research universities or the US government in the United States today.

7.1.3 RQ3 and RQ4: What are the properties, characteristics, and varieties of software projects with a nexus to these research institutions? How do the characteristics of repositories in the university ecosystem compare with the characteristics of repositories in the national laboratory

ecosystem?. In general, there is a higher reliance on “lower level” and compiled languages like C/C++ than non-RSE projects. Most projects are around 2 MB in size. The vast majority have small to no community, with very few forked or stared. RSE projects at a national laboratory are more likely to have recent updates, and are also likely to have GitHub pages enabled and issues disabled than at university-affiliated repositories. We speculate this is due to national laboratories more likely to have robust (and sometimes separate) infrastructure for handling project administration, such as communication around bugs/issues and documentation.

We also found that research universities saw a sharp drop-off of RSE repository creation around the start of COVID, which has still not fully recovered. Meanwhile, RSE creation at national laboratories increased or stayed constant before slightly dropping off in recent years.

7.1.4 RQ5: How does the code in these research projects relate with and depend on each other project in the ecosystem?. The vast majority of RSE projects we were able to analyze are independent of other RSE projects connected to national laboratories, although we were limited in the languages we were able to evaluate. That said, in the languages we did evaluate, a few dependencies were used heavily in many RSE projects. For example, the python packages numpy, scipy, and pandas were dependencies in 49%, 31%, and 30% of all python RSE repositories. We also found that there are several key “dependencies of dependencies” that are not able to be initially seen without empirical evaluation.

7.2 Limitations and Future Work

There are many limitations to our work, in all research questions explored. These limitations primarily point to a great deal of future work to more rigorously answer the questions posed in this dissertation. Namely, for RQ1, exploring additional

sources of RSE project homes, such as links in peer-reviewed papers, GitLab searches, and additional build system mining. For RQ2, analyzing non-DOE government laboratories and non-US government RSE projects remains a new high-level area of future work. We can also further explore the use of large language models as vehicles for the correct classification and categorization of RSE repositories. As GitHub releases more API endpoints (and, crucially, better documentation of those endpoints), we can more deeply interrogate properties of RSE repositories, expanding on RQ3 and RQ4. Of special interest is the analysis of commit patterns and issue resolution over time. Finally, we look forward to doing more comprehensive and complex dependency analysis, as well as intra-project analyses – like bus factor calculations; i.e., identifying and quantifying developers whose departure from a project could have a significantly negative impact – on an empirical scale going forward.

7.3 Contributions

As we said in the introduction, in addition to the answering of the above research questions and contributing a substantial literature review (Chapter II), the contributions of this dissertation are threefold:

1. We contribute groundbreaking quantitative insights about RSE projects by instantiating a broad theoretical framework for mining software repositories with domain-specific treatments in the United States’ national laboratories and universities. These insights, which include the first ballpark numbers of government-funded RSE projects, have public policy and budgetary implications for stakeholders. (Chapters III, IV; RQ1, RQ2)
2. From these quantitative insights, we contribute a theoretical framework for understanding common characteristics of projects in the RSE community

through a positivist-constructionist approach based on empirical data. Namely, RSE projects are chiefly characterized by the size of their community. We also hold a discussion of the inherent limitations in our framework's construction. (Chapter [V](#); RQ3, RQ4)

3. We contribute a methodological approach for analyzing software dependencies in the RSE ecosystems using graph theoretic techniques. (Chapter [VI](#); RQ5)

□

APPENDIX A

TAXONOMY MATRIX OF TOPICS, CATEGORIES, AND CROSS CUTTING THEMES

This appendix chapter is a user’s reference for all of the papers read and classified for this literature review. It holds the “flattened” 2D matrix of a taxonomy of 18 software engineering topics partitioned by the categories Tools, Psychology/Social Science, Analysis / Broader Studies, and Curated Datasets.

Many of the papers also contain cross cutting themes. We annotate each paper with a cross cutting theme with the following key at the end of the citation:

^A = Bots

^B = DEI - General

^C = DEI - Race/Ethnicity/National Origin

^D = DEI - Sex/Gender

^E = Graph Theory

^F = Machine Learning

^G = Privacy or Security

^H = Venue - Academia

^I = Venue - Industry

^J = Venue - Open Source

^K = Venue - Start ups

A.1 Learning / School

Several papers revolve around the learning of software engineering along some dimension, with a number of papers focused on classroom learning. This topic

incorporates any paper which touches on the learning of software or software engineering, or which has nexus with a school or university environment.

Tools.

- Toward Automatic Summarization of Arbitrary Java Statements for Novice Programmers, *M. Hassan and E. Hill*, [Hassan and Hill \(2018\)](#);
- Teaching Software Maintenance, *K. Gallagher, M. Fioravanti and S. Kozaitis*, [Gallagher, Fioravanti, and Kozaitis \(2019\)](#)^J;
- Recommending Exception Handling Code, *T. Nguyen, P. Vu and T. Nguyen*, [T. Nguyen, Vu, and Nguyen \(2019b\)](#)^F;
- A software maintenance-focused process and supporting toolset for academic environments, *R. Hardt*, [Hardt \(2020a\)](#);
- A toolset to support a software maintenance process in academic environments, *R. Hardt*, [Hardt \(2020b\)](#)^H;

Psychology / Social Science.

- How Modern News Aggregators Help Development Communities Shape and Share Knowledge, *M. Aniche et al.*, [Aniche et al. \(2018\)](#)^J;
- How Practitioners Perceive Coding Proficiency, *X. Xia, Z. Wan, P. S. Kochhar and D. Lo*, [P. S. K. X. Xia Z. Wan and Lo \(2019\)](#);
- Do Developers Discover New Tools On The Toilet?, *E. Murphy-Hill et al.*, [Murphy-Hill et al. \(2019\)](#)^I;

- Relating Reading, Visualization, and Coding for New Programmers: A Neuroimaging Study, *M. Endres, Z. Karas, X. Hu, I. Kovelman and W. Weimer*, [Endres, Karas, Hu, Kovelman, and Weimer \(2021\)](#);

Analysis / Broader Studies.

- Here We Go Again: Why Is It Difficult for Developers to Learn Another Programming Language?, *N. Shrestha, C. Botta, T. Barik and C. Parnin*, [Shrestha, Botta, Barik, and Parnin \(2020\)](#);
- Towards Bidirectional Live Programming for Incomplete Programs, *X. Zhang and Z. Hu, ?*;

A.2 Onboarding

Onboarding research relates to the process of inducing a developer to a project or team, or the steps which occur after evangelizing an existing project to other developers to adopt and these other developers first try it out.

Psychology / Social Science.

- How Modern News Aggregators Help Development Communities Shape and Share Knowledge, *M. Aniche et al.*, [Aniche et al. \(2018\)](#)^J;
- Almost There: A Study on Quasi-Contributors in Open-Source Software Projects, *I. Steinmacher, G. Pinto, I. S. Wiese and M. A. Gerosa*, [I. Steinmacher and Gerosa \(2018\)](#)^J;
- Do Developers Discover New Tools On The Toilet?, *E. Murphy-Hill et al.*, [Murphy-Hill et al. \(2019\)](#)^I;
- A Case Study of Onboarding in Software Teams: Tasks and Strategies, *A. Ju, H. Sajjani, S. Kelly and K. Herzig*, [Ju et al. \(2021\)](#)^I;

- This Is Damn Slick!” Estimating the Impact of Tweets on Open Source Project Popularity and New Contributors, *H. Fang, H. Lamba, J. Herbsleb and B. Vasilescu*, [H. Fang and Vasilescu \(2022\)](#)^J;

Analysis / Broader Studies.

- Which Contributions Predict Whether Developers Are Accepted into Github Teams *Middleton, Justin and Murphy-Hill, Emerson and Green, Demetrius and Meade, Adam and Mayer, Roger and White, David and McDonald, Steve Middleton et al.* [\(2018\)](#)^J;
- Open Source Barriers to Entry, Revisited: A Sociotechnical Perspective, *C. Mendez et al.*, [Mendez et al. \(2018\)](#)^D;
- Assessing the Characteristics of FOSS Contributions in Network Automation Projects, *J. Anderson, I. Steinmacher and P. Rodeghero*, [Anderson, Steinmacher, and Rodeghero \(2020\)](#)^J;
- Onboarding vs. Diversity, Productivity and Quality - Empirical Study of the OpenStack Ecosystem, *A. Foundjem, E. Eghan and B. Adams*, [Foundjem, Eghan, and Adams \(2021\)](#)^{BDJ};
- The Shifting Sands of Motivation: Revisiting What Drives Contributors in Open Source, *M. Gerosa et al.*, [Gerosa et al. \(2021\)](#)^J;

A.3 New features and requirements

The topic of inducing new features to a project and the requirements engineering which accompanies it consists of an entire subfield on its own, with entire conferences dedicated to the topic. Consequently, most requirement engineering tools end up being presented in conferences like RE. We focus here on just a few papers which

lie in the intersection between requirements engineering and the broader software development process.

Psychology / Social Science.

- Goal-Conflict Likelihood Assessment Based on Model Counting, *R. Degiovanni, P. Castro, M. Arroyo, M. Ruiz, N. Aguirre and M. Frias*, [Degiovanni et al. \(2018\)^F](#);
- Do this! Do that!, and Nothing will Happen” Do Specifications Lead to Securely Stored Passwords?, *J. Hallett, N. Patnaik, B. Shreeve and A. Rashid*, [Hallett, Patnaik, Shreeve, and Rashid \(2021\)^G](#);

Analysis / Broader Studies.

- The Evolution of Requirements Practices in Software Startups, *C. Galha, D. Damian, A. Wasserman, M. Goulao and J. Araujo*, [Galha et al. \(2018\)^K](#);

A.4 Code writing and refactoring

The code writing and refactoring category involves the actual production and manipulation of symbols in the creation or modification of software – usually in some sort of editor. This is the stereotype of programming that non developers usually think as the whole of software development.

Tools.

- Predicting Developers’ IDE Commands with Machine Learning *Bulmer, Tyson and Montgomery, Lloyd and Damian, Daniela* [Bulmer et al. \(2018\)^F](#);
- Deuce: A Lightweight User Interface for Structured Editing, *B. Hempel, J. Lubin, G. Lu and R. Chugh*, [Hempel et al. \(2018\)](#);

- Generating Accurate and Compact Edit Scripts Using Tree Differencing, *V. Frick, T. Grassauer, F. Beck and M. Pinzger*, [Frick, Grassauer, Beck, and Pinzger \(2018\)^E](#);
- Statistical Learning of API Fully Qualified Names in Code Snippets of Online Forums, *H. Phan, H. A. Nguyen, N. M. Tran, L. H. Truong, A. T. Nguyen and T. N. Nguyen*, [H. Phan and Nguyen \(2018\)^F](#);
- CLEVER: Combining Code Metrics with Clone Detection for Just-in-Time Fault Prevention and Resolution in Large Industrial Projects *Nayrolles, Mathieu and Hamou-Lhadj, Abdelwahab* [Nayrolles and Hamou-Lhadj \(2018\)^{FG}](#);
- On Learning Meaningful Code Changes Via Neural Machine Translation, *M. Tufano, J. Pantiuchina, C. Watson, G. Bavota and D. Poshyvanyk*, [Tufano et al. \(2019\)^F](#);
- FOCUS: A Recommender System for Mining API Function Calls and Usage Patterns, *P. T. Nguyen, J. Di Rocco, D. Di Ruscio, L. Ochoa, T. Degueule and M. Di Penta*, [P. T. Nguyen et al. \(2019\)^F](#);
- Learning to Spot and Refactor Inconsistent Method Names, *K. Liu et al.*, [Liu et al. \(2019\)^F](#);
- Personalized Code Recommendation, *T. Nguyen, P. Vu and T. Nguyen*, [T. Nguyen et al. \(2019a\)^F](#);
- Deep Learning Anti-Patterns from Code Metrics History, *A. Barbez, F. Khomh and Y. -G. Gueheneuc*, [Barbez et al. \(2019\)^F](#);
- Analyzing and Supporting Adaptation of Online Code Examples, *T. Zhang, D. Yang, C. Lopes and M. Kim*, [C. L. T. Zhang D. Yang and Kim \(2019\)](#);

- Suggesting Natural Method Names to Check Name Consistencies, *S. Nguyen, H. Phan, T. Le and T. N. Nguyen*, [S. Nguyen et al. \(2020\)^F](#);
- Improving Data Scientist Efficiency with Provenance, *J. Hu, J. Joung, M. Jacobs, K. Z. Gajos and M. I. Seltzer*, [J. Hu and Seltzer \(2020\)](#);
- Fast and Memory-Efficient Neural Code Completion, *A. Svyatkovskiy, S. Lee, A. Hadjitofi, M. Riechert, J. V. Franco and M. Allamanis*, [Svyatkovskiy et al. \(2021\)^F](#);
- Siri, Write the Next Method, *F. Wen, E. Aghajani, C. Nagy, M. Lanza and G. Bavota*, [Wen, Aghajani, et al. \(2021\)^{AF}](#);
- CodeRibbon: More Efficient Workspace Management and Navigation for Mainstream Development Environments, *B. P. Klein and A. Z. Henley*, [Klein and Henley \(2021\)](#);
- FeaRS: Recommending Complete Android Method Implementations, *F. Wen, V. Ferrari, E. Aghajani, C. Nagy, M. Lanza and G. Bavota*, [Wen, Ferrari, et al. \(2021\)^F](#);
- IDEAL: An Open-Source Identifier Name Appraisal Tool, *A. Peruma, V. Arnaoudova and C. D. Newman*, [Peruma, Arnaoudova, and Newman \(2021\)](#);
- Sirius: Static Program Repair with Dependence Graph-Based Systematic Edit Patterns, *K. Noda, H. Yokoyama and S. Kikuchi*, [K. Noda and Kikuchi \(2021\)^{EJ}](#);
- Senatus - A Fast and Accurate Code-to-Code Recommendation Engine, *F. Silavong, S. Moran, A. Georgiadis, R. Saphal and R. Otter*, [Silavong et al. \(2022\)^F](#);

- Type4Py: Practical Deep Similarity Learning-Based Type Inference for Python, *A. M. Mir, E. Latoskinas, S. Proksch and G. Gousios*, [Mir et al. \(2022\)^F](#);
- Psychology / Social Science.**
- Do Software Engineers Use Autocompletion Features Differently than Other Developers? *Amlekar, Rahul and Gamboa, Andrés Felipe Rincón and Gallaba, Keheliya and McIntosh, Shane* [Amlekar et al. \(2018\)](#);
- The Road to Live Programming: Insights from the Practice, *J. Kubelka, R. Robbes and A. Bergel*, [Kubelka, Robbes, and Bergel \(2018\)](#);
- Sentiment Polarity Detection for Software Development, *F. Calefato, F. Lanubile, F. Maiorano and N. Novielli*, [Calefato et al. \(2018\)^F](#);
- Towards Better Understanding Developer Perception of Refactoring, *E. A. Alomar*, [Alomar \(2019\)](#);
- Developer Reading Behavior While Summarizing Java Methods: Size and Context Matters, *N. J. Abid, B. Sharif, N. Dragan, H. Alrasheed and J. I. Maletic*, [Abid et al. \(2019\)](#);
- How Practitioners Perceive Coding Proficiency, *X. Xia, Z. Wan, P. S. Kochhar and D. Lo*, [P. S. K. X. Xia Z. Wan and Lo \(2019\)](#);
- Socio-Technical Work-Rate Increase Associates With Changes in Work Patterns in Online Projects, *F. Sarker, B. Vasilescu, K. Blincoe and V. Filkov*, [F. Sarker and Filkov \(2019\)^J](#);
- Do as I Do, Not as I Say: Do Contribution Guidelines Match the GitHub Contribution Process?, *O. Elazhary, M. -A. Storey, N. Ernst and A. Zaidman*, [O. Elazhary and Zaidman \(2019\)^J](#);

- Recognizing Developers’ Emotions while Programming, *D. Girardi, N. Novielli, D. Fucci and F. Lanubile*, [Girardi et al. \(2020\)](#);
- Neurological Divide: An fMRI Study of Prose and Code Writing, *R. Krueger, Y. Huang, X. Liu, T. Santander, W. Weimer and K. Leach*, [Krueger et al. \(2020\)](#);
- Refactoring Recommendations Based on the Optimization of Socio-Technical Congruence, *M. D. Stefano, F. Pecorelli, D. A. Tamburri, F. Palomba and A. D. Lucia*, [Stefano et al. \(2020\)](#);
- On the Naming of Methods: A Survey of Professional Developers, *R. Alsuhaibani, C. Newman, M. Decker, M. Collard and J. Maletic*, [Alsuhaibani et al. \(2021\)](#)^I;

Analysis / Broader Studies.

- Studying Developer Build Issues and Debugger Usage via Timeline Analysis in Visual Studio IDE *Bellman, Christopher and Seet, Ahmad and Baysal, Olga* [Bellman et al. \(2018\)](#);
- Predicting Future Developer Behavior in the IDE Using Topic Models, *K. Damevski, H. Chen, D. C. Shepherd, N. A. Kraft and L. Pollock*, [Damevski et al. \(2018\)](#);
- Does the Propagation of Artifact Changes Across Tasks Reflect Work Dependencies?, *C. Mayr-Dorn and A. Egyed*, [Mayr-Dorn and Egyed \(2018\)](#);
- Towards a Model to Appraise and Suggest Identifier Names, *A. Peruma*, [Peruma \(2019\)](#);

- Self-Admitted Technical Debt Removal and Refactoring Actions: Co-Occurrence or More?, *M. Iammarino, F. Zampetti, L. Aversano and M. Di Penta*, [M. Iammarino and Penta \(2019\)^J](#);
- Investigating Context Adaptation Bugs in Code Clones, *M. Mondal, B. Roy, C. K. Roy and K. A. Schneider*, [M. Mondal and Schneider \(2019\)^J](#);
- An Empirical Study on the Usage of BERT Models for Code Completion, *M. Ciniselli, N. Cooper, L. Pascarella, D. Poshyanyk, M. Di Penta and G. Bavota*, [Ciniselli et al. \(2021\)^F](#);
- An Empirical Evaluation of GitHub Copilot’s Code Suggestions, *N. Nguyen and S. Nadi*, [N. Nguyen and Nadi \(2022\)^{AF}](#);
- To What Extent do Deep Learning-based Code Recommenders Generate Predictions by Cloning Code from the Training Set?, *M. Ciniselli, L. Pascarella and G. Bavota*, [M. Ciniselli and Bavota \(2022\)^{FJ}](#);

A.5 Help, Q&A, code comprehension, and documentation consumption

A key part of software development is an interaction between the writing of the code itself and learning about facets of the implementation details. This topic includes papers related to those parts of learning – whether interacting and reading from Stack Overflow, chatting with another developer, or digesting documentation.

Tools.

- FaCoY – A Code-to-Code Search Engine, *K. Kim et al.*, [Kim et al. \(2018\)](#);
- Programming Not Only by Example, *H. Peleg, S. Shoham and E. Yahav*, [H. Peleg and Yahav \(2018\)](#);

- Statistical Learning of API Fully Qualified Names in Code Snippets of Online Forums, *H. Phan, H. A. Nguyen, N. M. Tran, L. H. Truong, A. T. Nguyen and T. N. Nguyen*, [H. Phan and Nguyen \(2018\)](#)^F;
- Improving API Caveats Accessibility by Mining API Caveats Knowledge Graph, *H. Li et al.*, [H. Li et al. \(2018\)](#)^{EF};
- Enlightened Debugging, *X. Li, S. Zhu, M. d’Amorim and A. Orso*, [X. Li and Orso \(2018\)](#)^F;
- Context-Aware Software Documentation, *E. Aghajani*, [Aghajani \(2018\)](#);
- FOCUS: A Recommender System for Mining API Function Calls and Usage Patterns, *P. T. Nguyen, J. Di Rocco, D. Di Ruscio, L. Ochoa, T. Degueule and M. Di Penta*, [P. T. Nguyen et al. \(2019\)](#)^F;
- Know-How in Programming Tasks: From Textual Tutorials to Task-Oriented Knowledge Graph, *J. Sun, Z. Xing, R. Chu, H. Bai, J. Wang and X. Peng*, [J. Sun and Peng \(2019\)](#)^E;
- Automatic Identification of Rollback Edit with Reasons in Stack Overflow Q&A Site, *S. Mondal, G. Uddin and C. K. Roy*, [S. Mondal and Roy \(2020\)](#)^F;
- Attention-based model for predicting question relatedness on Stack Overflow, *J. Pei, Y. Wu, Z. Qin, Y. Cong and J. Guan*, [Pei, Wu, Qin, Cong, and Guan \(2021\)](#)^F;
- Dialogue Management for Interactive API Search, *Z. Eberhart and C. McMillan*, [Eberhart and McMillan \(2021\)](#)^F;

- Automated Recovery of Issue-Commit Links Leveraging Both Textual and Non-textual Data, *P. R. Mazrae, M. Izadi and A. Heydarnoori*, [P. R. Mazrae and Heydarnoori \(2021\)^F](#);
- SoCCMiner: A Source Code-Comments and Comment-Context Miner, *M. Sridharan, M. Mantyla, M. Claes and L. Rantala*, [M. Sridharan and Rantala \(2022\)](#);

Psychology / Social Science.

- Evaluating How Developers Use General-Purpose Web-Search for Code Retrieval *Rahman, Md Masudur and Barson, Jed and Paul, Sydney and Kayani, Joshua and Lois, Federico Andrés and Quezada, Sebastián Fernandez and Parnin, Christopher and Stolee, Kathryn T. and Ray, Baishakhi* [M. M. Rahman et al. \(2018\)^F](#);
- Are Code Examples on an Online Q&A Forum Reliable?: A Study of API Misuse on Stack Overflow, *T. Zhang, G. Upadhyaya, A. Reinhardt, H. Rajan and M. Kim*, [A. R. H. R. T. Zhang G. Upadhyaya and Kim \(2018\)](#);
- Understanding the Factors for Fast Answers in Technical Q&A Websites: An Empirical Study of Four Stack Exchange Websites, *S. Wang, T. -H. Chen and A. E. Hassan*, [T. H. C. S. Wang and Hassan \(2018\)^J](#);
- Measuring Program Comprehension: A Large-Scale Field Study with Professionals, *X. Xia, L. Bao, D. Lo, Z. Xing, A. E. Hassan and S. Li*, [D. L. Z. X. A. E. H. X. Xia L. Bao and Li \(2018\)](#);
- How Modern News Aggregators Help Development Communities Shape and Share Knowledge, *M. Aniche et al.*, [Aniche et al. \(2018\)^J](#);

- Towards Better Understanding Developer Perception of Refactoring, *E. A. Alomar*, [Alomar \(2019\)](#);
- Developer Reading Behavior While Summarizing Java Methods: Size and Context Matters, *N. J. Abid, B. Sharif, N. Dragan, H. Alrasheed and J. I. Maletic*, [Abid et al. \(2019\)](#);
- Pattern-Based Mining of Opinions in Q&A Websites, *B. Lin, F. Zampetti, G. Bavota, M. Di Penta and M. Lanza*, [B. Lin and Lanza \(2019\)^{FJ}](#);
- An Empirical Study Assessing Source Code Readability in Comprehension, *J. Johnson, S. Lubo, N. Yedla, J. Aponte and B. Sharif*, [J. Johnson and Sharif \(2019\)^B](#);
- Do as I Do, Not as I Say: Do Contribution Guidelines Match the GitHub Contribution Process?, *O. Elazhary, M. -A. Storey, N. Ernst and A. Zaidman*, [O. Elazhary and Zaidman \(2019\)^J](#);
- Haste Makes Waste: An Empirical Study of Fast Answers in Stack Overflow, *Y. Lu, X. Mao, M. Zhou, Y. Zhang, T. Wang and Z. Li*, [Y. Lu and Li \(2020\)](#);
- Rollback Edit Inconsistencies in Developer Forum, *S. Mondal, G. Uddin and C. K. Roy*, [S. Mondal and Roy \(2021\)](#);
- The Mind Is a Powerful Place: How Showing Code Comprehensibility Metrics Influences Code Understanding, *M. Wyrich, A. Preikschat, D. Graziotin and S. Wagner*, [M. Wyrich and Wagner \(2021\)](#);

Analysis / Broader Studies.

- Studying Developer Build Issues and Debugger Usage via Timeline Analysis in Visual Studio IDE *Bellman, Christopher and Seet, Ahmad and Baysal, Olga* [Bellman et al. \(2018\)](#);
- Comprehension Effort and Programming Activities: Related? Or Not Related? *Rahman, Akond* [A. Rahman \(2018\)](#);
- Does the Propagation of Artifact Changes Across Tasks Reflect Work Dependencies?, *C. Mayr-Dorn and A. Egyed*, [Mayr-Dorn and Egyed \(2018\)](#);
- Analyzing and Supporting Adaptation of Online Code Examples, *T. Zhang, D. Yang, C. Lopes and M. Kim*, [C. L. T. Zhang D. Yang and Kim \(2019\)](#);
- 9.6 Million Links in Source Code Comments: Purpose, Evolution, and Decay, *H. Hata, C. Treude, R. G. Kula and T. Ishio*, [H. Hata and Ishio \(2019\)](#)^J;
- Challenges in Chatbot Development: A Study of Stack Overflow Posts *Abdellatif, Ahmad and Costa, Diego and Badran, Khaled and Abdalkareem, Rabe and Shihab, Emad* [Abdellatif et al. \(2020\)](#)^{AFJ};
- Sentiment Analysis for Software Engineering: How Far Can Pre-trained Transformer Models Go?, *T. Zhang, B. Xu, F. Thung, S. A. Haryono, D. Lo and L. Jiang*, [F. T. S. A. H. D. L. T. Zhang B. Xu and Jiang \(2020\)](#)^F;
- Evaluating Code Readability and Legibility: An Examination of Human-centric Studies, *D. Oliveira, R. Bruno, F. Madeiral and F. Castor*, [D. Oliveira and Castor \(2020\)](#);
- Characterizing Task-Relevant Information in Natural Language Software Artifacts, *A. Marques, N. C. Bradley and G. C. Murphy*, [A. Marques and Murphy \(2020\)](#);

- Studying the Change Histories of Stack Overflow and GitHub Snippets, *S. S. Manes and O. Baysal*, [Manes and Baysal \(2021\)](#)^J;
- Challenges in Developing Desktop Web Apps: a Study of Stack Overflow and GitHub, *G. L. Scoccia, P. Migliarini and M. Autili*, [G. L. Scoccia and Autili \(2021\)](#)^J;
- Googling for Software Development: What Developers Search For and What They Find, *A. Hora*, [Hora \(2021\)](#);
- Does This Apply to Me? An Empirical Study of Technical Context in Stack Overflow, *A. Galappaththi, S. Nadi and C. Treude*, [A. Galappaththi and Treude \(2022\)](#)^J;
- Multimodal Recommendation of Messenger Channels, *E. Koshchenko, E. Klimov and V. Kovalenko*, [E. Koshchenko and Kovalenko \(2022\)](#)^A;
- Bridging Pre-trained Models and Downstream Tasks for Source Code Understanding, *D. Wang et al.*, [D. Wang et al. \(2022\)](#)^F;

Curated Datasets.

- Two Datasets for Sentiment Analysis in Software Engineering, *B. Lin, F. Zampetti, R. Oliveto, M. Di Penta, M. Lanza and G. Bavota*, [Lin, Zampetti, Oliveto, et al. \(2018\)](#)^F;
- A Gold Standard for Emotion Annotation in Stack Overflow *Novielli, Nicole and Calefato, Fabio and Lanubile, Filippo* [Novielli et al. \(2018\)](#)^F;
- SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts, *Baltes, Sebastian and Dumani, Lorik and Treude, Christoph and Diehl, Stephan* [Baltes et al. \(2018\)](#)

- SOSum: A Dataset of Stack Overflow Post Summaries, *B. Kou, Y. Di, M. Chen and T. Zhang*, [B. Kou and Zhang \(2022\)](#);

A.6 Documentation production

In order for other developers to read documentation and have it be useful, a programmer (or bot) must also create documentation. This topic is all about the creation of code documentation and code summarization in natural language so that humans can readily understand what’s going on without having to do manual code tracing. It also can include papers with tangential elements, such as the creation of documentation while refactoring, for example

Tools.

- Context-Aware Software Documentation, *E. Aghajani*, [Aghajani \(2018\)](#);
- Toward Automatic Summarization of Arbitrary Java Statements for Novice Programmers, *M. Hassan and E. Hill*, [Hassan and Hill \(2018\)](#);
- Retrieval-based Neural Source Code Summarization, *J. Zhang, X. Wang, H. Zhang, H. Sun and X. Liu*, [J. Zhang and Liu \(2020\)](#)^F;
- Automated Recovery of Issue-Commit Links Leveraging Both Textual and Non-textual Data, *P. R. Mazrae, M. Izadi and A. Heydarnoori*, [P. R. Mazrae and Heydarnoori \(2021\)](#)^F;
- Towards Reliable Agile Iterative Planning via Predicting Documentation Changes of Work Items, *J. Pasuksmit, P. Thongtanunam and S. Karunasekera*, [J. Pasuksmit and Karunasekera \(2022\)](#)^F;

- RoPGen: Towards Robust Code Authorship Attribution via Automatic Coding Style Transformation, *Z. Li, G. Q. Chen, C. Chen, Y. Zou and S. Xu*, [Z. Li and Xu \(2022\)^F](#);

Psychology / Social Science.

- Towards Better Understanding Developer Perception of Refactoring, *E. A. Alomar*, [Alomar \(2019\)](#);
- Towards Just-Enough Documentation for Agile Effort Estimation: What Information Should Be Documented?, *J. Pasuksmit, P. Thongtanunam and S. Karunasekera*, [J. Pasuksmit and Karunasekera \(2021\)](#);
- Comments on Comments: Where Code Review and Documentation Meet, *N. Rao, J. Tsay, M. Hirzel and V. J. Hellendoorn*, [N. Rao and Hellendoorn \(2022\)](#);
- Practitioners’ Expectations on Automated Code Comment Generation, *X. Hu, X. Xia, D. Lo, Z. Wan, Q. Chen and T. Zimmermann*, [X. Hu and Zimmermann \(2022\)^F](#);

Analysis / Broader Studies.

- When Not to Comment: Questions and Tradeoffs with API Documentation for C++ Projects, *A. Head, C. Sadowski, E. Murphy-Hill and A. Knight*, [A. Head and Knight \(2018\)](#);
- Inappropriate Usage Examples in Web API Documentations, *M. Hosono et al.*, [Hosono et al. \(2019\)](#);
- Software Documentation Issues Unveiled, *E. Aghajani et al.*, [Aghajani et al. \(2019\)^J](#);

- Software Documentation: The Practitioners’ Perspective, *E. Aghajani et al.*, [Aghajani et al. \(2020\)](#);
- On the need for automatic knowledge management in modern collaboration tools to improve software maintenance, *V. Balachandran*, [Balachandran \(2020\)](#);
- An Empirical Study on Code Comment Completion, *A. Mastropaolo, E. Aghajani, L. Pascarella and G. Bavota*, [A. Mastropaolo and Bavota \(2021\)^F](#);

A.7 Testing

Well tested code leads to fewer bugs. But what is testing? Do most developers find testing to be a chore? How does it factor into the larger software process? Research papers in this topic address these flavors of questions.

Tools.

- Automatic Test Smell Detection Using Information Retrieval Techniques, *F. Palomba, A. Zaidman and A. De Lucia*, [F. Palomba and Lucia \(2018\)](#);
- Automatically Generating Precise Oracles from Structured Natural Language Specifications, *M. Motwani and Y. Brun*, [Motwani and Brun \(2019\)^F](#);
- Automatic Web Testing Using Curiosity-Driven Reinforcement Learning, *Y. Zheng et al.*, [Zheng et al. \(2021\)^{AFGJ}](#);
- Graph-Based Fuzz Testing for Deep Learning Inference Engines, *W. Luo, D. Chai, X. Ruan, J. Wang, C. Fang and Z. Chen*, [W. Luo and Chen \(2021\)^{EF}](#);
- Automated Assertion Generation via Information Retrieval and Its Integration with Deep learning, *H. Yu et al.*, [Yu et al. \(2022\)^F](#);

Psychology / Social Science.

- How the Experience of Development Teams Relates to Assertion Density of Test Classes, *G. Catolino, F. Palomba, A. Zaidman and F. Ferrucci*, [G. Catolino and Ferrucci \(2019\)](#);

Analysis / Broader Studies.

- When Testing Meets Code Review: Why and How Developers Review Tests, *D. Spadini, M. Aniche, M. -A. Storey, M. Bruntink and A. Bacchelli*, [M. A. S. M. B. D. Spadini M. Aniche and Bacchelli \(2018\)](#);
- On the Relation of Test Smells to Software Code Quality, *D. Spadini, F. Palomba, A. Zaidman, M. Bruntink and A. Bacchelli*, [A. Z. M. B. D. Spadini F. Palomba and Bacchelli \(2018\)](#);
- How Do Code Changes Evolve in Different Platforms? A Mining-Based Investigation, *M. Vigiato, J. Oliveira, E. Figueiredo, P. Jamshidi and C. Kastner*, [M. Vigiato and Kästner \(2019\)](#);
- Studying Test Annotation Maintenance in the Wild, *D. J. Kim, N. Tsantalis, T. -H. Chen and J. Yang*, [D. J. Kim and Yang \(2021\)](#);

A.8 Commits, merges, and conflicts

Once code is written, documentation developed, and tests executed, it's time to commit. In modern software projects, that includes committing the code and then (potentially) dealing with merges and merge conflicts. This is a tedious challenge, and several works have focused on what happens when there's complex code overlaps. This topic does blur somewhat with the topic of pull requests, which often happen immediately after (or alongside) merge conflicts are sorted out.

Tools.

- Linking Source Code to Untangled Change Intents, *X. Liu, L. Huang, C. Li and V. Ng*, [X. Liu and Ng \(2018\)](#)^F;
- CLEVER: Combining Code Metrics with Clone Detection for Just-in-Time Fault Prevention and Resolution in Large Industrial Projects *Nayrolles, Mathieu and Hamou-Lhadj, Abdelwahab* [Nayrolles and Hamou-Lhadj \(2018\)](#)^{FG};
- Detecting and Characterizing Bots That Commit Code *Dey, Tapajit and Mousavi, Sara and Ponce, Eduardo and Fry, Tanner and Vasilescu, Bogdan and Filippova, Anna and Mockus, Audris* [T. Dey et al. \(2020\)](#)^{AFJ};
- Planning for Untangling: Predicting the Difficulty of Merge Conflicts, *C. Brindescu, I. Ahmed, R. Leano and A. Sarma*, [R. L. C. Brindescu I. Ahmed and Sarma \(2020\)](#)^F;
- FIRA: Fine-Grained Graph-Based Code Change Representation for Automated Commit Message Generation, *J. Dong et al.*, [Dong et al. \(2022\)](#)^{EF};

Psychology / Social Science.

- Communicative Intention in Code Review Questions, *F. Ebert, F. Castor, N. Novielli and A. Serebrenik*, [F. Ebert and Serebrenik \(2018\)](#);
- Socio-Technical Work-Rate Increase Associates With Changes in Work Patterns in Online Projects, *F. Sarker, B. Vasilescu, K. Blincoe and V. Filkov*, [F. Sarker and Filkov \(2019\)](#)^J;
- Lifting the Curtain on Merge Conflict Resolution: A Sensemaking Perspective, *C. Brindescu, Y. Ramirez, A. Sarma and C. Jensen*, [A. S. C. Brindescu Y. Ramirez and Jensen \(2020\)](#);

Analysis / Broader Studies.

- Analyzing Conflict Predictors in Open-Source Java Projects *Accioly, Paola and Borba, Paulo and Silva, Léuson and Cavalcanti, Guilherme* [Accioly, Borba, Silva, and Cavalcanti \(2018\)](#)^J;
- How do Multiple Pull Requests Change the Same Code: A Study of Competing Pull Requests in GitHub, *X. Zhang et al.*, [X. Zhang et al. \(2018\)](#)^J;
- The List is the Process: Reliable Pre-Integration Tracking of Commits on Mailing Lists, *R. Ramsauer, D. Lohmann and W. Mauerer*, [R. Ramsauer and Mauerer \(2019\)](#)^J;
- Can Program Synthesis be Used to Learn Merge Conflict Resolutions? An Empirical Analysis, *R. Pan, V. Le, N. Nagappan, S. Gulwani, S. Lahiri and M. Kaufman*, [R. Pan and Kaufman \(2021\)](#)^J;
- On the Evaluation of Commit Message Generation Models: An Experimental Study, *W. Tao et al.*, [Tao et al. \(2021\)](#)^F;
- Which contributions count? Analysis of attribution in open source, *J. -G. Young, A. Casari, K. McLaughlin, M. Z. Trujillo, L. Hebert-Dufresne and J. P. Bagrow*, [J. G. Young and Bagrow \(2021\)](#)^J;
- Why Security Defects Go Unnoticed During Code Reviews? A Case-Control Study of the Chromium OS Project, *R. Paul, A. K. Turzo and A. Bosu*, [R. Paul and Bosu \(2021\)](#)^{GJ};
- What Makes a Good Commit Message?, *Y. Tian, Y. Zhang, K. -J. Stol, L. Jiang and H. Liu*, [Y. Tian and Liu \(2022\)](#)^J;

Curated Datasets.

- A Graph-Based Dataset of Commit History of Real-World Android Apps *Geiger, Franz-Xaver and Malavolta, Ivano and Pascarella, Luca and Palomba, Fabio and Di Nucci, Dario and Bacchelli, Alberto* [Geiger et al. \(2018\)^J](#);

A.9 Pull requests and code reviews

Code is complete and merged / compatible with an existing project. Now comes the pull request and code reviews: a human element that determines whether your code is up to snuff and good enough to be part of the project. These papers

Tools.

- BLIMP Tracer: Integrating Build Impact Analysis with Code Review, *R. Wen, D. Gilbert, M. G. Roche and S. McIntosh*, [R. Wen and McIntosh \(2018\)](#);
- CLEVER: Combining Code Metrics with Clone Detection for Just-in-Time Fault Prevention and Resolution in Large Industrial Projects *Nayrolles, Mathieu and Hamou-Lhadj, Abdelwahab* [Nayrolles and Hamou-Lhadj \(2018\)^{FG}](#);
- Developers’ Game: A Preliminary Study Concerning a Tool for Automated Developers Assessment, *W. Fracz and J. Dajda*, [Fracz and Dajda \(2018\)](#);
- On Learning Meaningful Code Changes Via Neural Machine Translation, *M. Tufano, J. Pantiuchina, C. Watson, G. Bavota and D. Poshyvanyk*, [Tufano et al. \(2019\)^F](#);
- Expanding the Number of Reviewers in Open-Source Projects by Recommending Appropriate Developers, *A. Chueshev, J. Lawall, R. Bendraou and T. Ziadi*, [A. Chueshev and Ziadi \(2020\)^J](#);

- Robin: A Voice Controlled Virtual Teammate for Software Developers and Teams, *B. da Silva, C. Hebert, A. Rawka and S. Sereesathien*, [B. da Silva and Sereesathien \(2020\)](#)^{AF};
- Towards Automating Code Review Activities, *R. Tufano, L. Pascarella, M. Tufano, D. Poshyvanyk and G. Bavota*, [M. T. D. P. R. Tufano L. Pascarella and Bavota \(2021\)](#)^F;
- Using Pre-Trained Models to Boost Code Review Automation, *R. Tufano, S. Masiero, A. Mastropaolo, L. Pascarella, D. Poshyvanyk and G. Bavota*, [A. M. L. P. D. P. R. Tufano S. Masiero and Bavota \(2022\)](#)^F;
- Modeling Review History for Reviewer Recommendation: A Hypergraph Approach, *G. Rong, Y. Zhang, L. Yang, F. Zhang, H. Kuang and H. Zhang*, [G. Rong and Zhang \(2022\)](#)^E;

Psychology / Social Science.

- “Was My Contribution Fairly Reviewed?” A Framework to Study the Perception of Fairness in Modern Code Reviews, *D. M. German, G. Robles, G. Poo-Caamano, X. Yang, H. Iida and K. Inoue*, [D. M. German and Inoue \(2018\)](#)^J;
- Almost There: A Study on Quasi-Contributors in Open-Source Software Projects, *I. Steinmacher, G. Pinto, I. S. Wiese and M. A. Gerosa*, [I. Steinmacher and Gerosa \(2018\)](#)^J;
- Socio-Technical Work-Rate Increase Associates With Changes in Work Patterns in Online Projects, *F. Sarker, B. Vasilescu, K. Blincoe and V. Filkov*, [F. Sarker and Filkov \(2019\)](#)^J;

- Less is More: Supporting Developers in Vulnerability Detection during Code Review, *L. Braz, C. Aeberhard, G. Calikli and A. Bacchelli*, [L. Braz and Bacchelli \(2022\)](#);

Analysis / Broader Studies.

- When Testing Meets Code Review: Why and How Developers Review Tests, *D. Spadini, M. Aniche, M. -A. Storey, M. Bruntink and A. Bacchelli*, [M. A. S. M. B. D. Spadini M. Aniche and Bacchelli \(2018\)](#);
- Studying the Impact of Adopting Continuous Integration on the Delivery Time of Pull Requests *Bernardo, João Helis and da Costa, Daniel Alencar and Kulesza, Uirá* [Bernardo, da Costa, and Kulesza \(2018\)](#)^J;
- How do Multiple Pull Requests Change the Same Code: A Study of Competing Pull Requests in GitHub, *X. Zhang et al.*, [X. Zhang et al. \(2018\)](#)^J;
- Effects of Adopting Code Review Bots on Pull Requests to OSS Projects, *M. Wessel, A. Serebrenik, I. Wiese, I. Steinmacher and M. A. Gerosa*, [M. Wessel and Gerosa \(2020\)](#)^{AJ};
- How Does Modern Code Review Impact Software Design Degradation? An In-depth Empirical Study, *A. Uchoa et al.*, [Uchoa et al. \(2020\)](#);
- Predicting Design Impactful Changes in Modern Code Review: A Large-Scale Empirical Study, *A. Uchoa et al.*, [Uchoa et al. \(2021\)](#)^{FJ};
- Does Code Review Promote Conformance? A Study of OpenStack Patches, *P. Sri-iesaranusorn, R. G. Kula and T. Ishio*, [P. Sri-iesaranusorn and Ishio \(2021\)](#)^J;

- Mining Code Review Data to Understand Waiting Times Between Acceptance and Merging: An Empirical Analysis, *G. Kudrjavets, A. Kumar, N. Nagappan and A. Rastogi*, [N. N. G. Kudrjavets A. Kumar and Rastogi \(2022\)^J](#);
- The Unexplored Treasure Trove of Phabricator Code Reviews, *G. Kudrjavets, N. Nagappan and A. Rastogi*, [N. N. G. Kudrjavets and Rastogi \(2022\)^J](#);
- Bots for Pull Requests: The Good, the Bad, and the Promising, *M. Wessel et al.*, [Wessel et al. \(2022\)^{AJ}](#);
- Which bugs are missed in code reviews: An empirical study on SmartSHARK dataset, *F. Khoshnoud, A. R. Nasab, Z. Toudeji and A. Sami*, [F. Khoshnoud and Sami \(2022\)^J](#);

Curated Datasets.

- A Graph-Based Dataset of Commit History of Real-World Android Apps *Geiger, Franz-Xaver and Malavolta, Ivano and Pascarella, Luca and Palomba, Fabio and Di Nucci, Dario and Bacchelli, Alberto* [Geiger et al. \(2018\)^J](#);

A.10 Smells and quality

This topic has to do with the metrics of code – namely, papers which revolve around code quality, smells, or other similar types of measurements.

Tools.

- Automatic Test Smell Detection Using Information Retrieval Techniques, *F. Palomba, A. Zaidman and A. De Lucia*, [F. Palomba and Lucia \(2018\)](#);
- CLEVER: Combining Code Metrics with Clone Detection for Just-in-Time Fault Prevention and Resolution in Large Industrial Projects *Nayrolles, Mathieu and Hamou-Lhadj, Abdelwahab* [Nayrolles and Hamou-Lhadj \(2018\)^{FG}](#);

- Beyond Metadata: Code-Centric and Usage-Based Analysis of Known Vulnerabilities in Open-Source Software, *S. E. Ponta, H. Plate and A. Sabetta*, [S. E. Ponta and Sabetta \(2018\)^J](#);
- Developers’ Game: A Preliminary Study Concerning a Tool for Automated Developers Assessment, *W. Fracz and J. Dajda*, [Fracz and Dajda \(2018\)](#);
- Deep Learning Anti-Patterns from Code Metrics History, *A. Barbez, F. Khomh and Y. -G. Gueheneuc*, [Barbez et al. \(2019\)^F](#);
- Static source code metrics and static analysis warnings for fine-grained just-in-time defect prediction, *A. Trautsch, S. Herbold and J. Grabowski*, [A. Trautsch and Grabowski \(2020\)^F](#);
- Automated Recovery of Issue-Commit Links Leveraging Both Textual and Non-textual Data, *P. R. Mazrae, M. Izadi and A. Heydarnoori*, [P. R. Mazrae and Heydarnoori \(2021\)^F](#);
- RoPGen: Towards Robust Code Authorship Attribution via Automatic Coding Style Transformation, *Z. Li, G. Q. Chen, C. Chen, Y. Zou and S. Xu*, [Z. Li and Xu \(2022\)^F](#);

Psychology / Social Science.

- The Mind Is a Powerful Place: How Showing Code Comprehensibility Metrics Influences Code Understanding, *M. Wyrich, A. Preikschat, D. Graziotin and S. Wagner*, [M. Wyrich and Wagner \(2021\)](#);

Analysis / Broader Studies.

- Improving Code: The (Mis) Perception of Quality Metrics, *J. Pantiuchina, M. Lanza and G. Bavota*, [J. Pantiuchina and Bavota \(2018\)](#);

- On the Relation of Test Smells to Software Code Quality, *D. Spadini, F. Palomba, A. Zaidman, M. Bruntink and A. Bacchelli*, [A. Z. M. B. D. Spadini F. Palomba and Bacchelli \(2018\)](#);
- Design Smell Detection and Analysis for Open Source Java Software, *A. Imran*, [Imran \(2019\)](#)^J;
- Which contributions count? Analysis of attribution in open source, *J. -G. Young, A. Casari, K. McLaughlin, M. Z. Trujillo, L. Hebert-Dufresne and J. P. Bagrow*, [J. G. Young and Bagrow \(2021\)](#)^J;
- An Evolutionary Analysis of Software-Architecture Smells, *P. Gnoyke, S. Schulze and J. Kruger*, [P. Gnoyke and Krüger \(2021\)](#);

Curated Datasets.

- QScored: A Large Dataset of Code Smells and Quality Metrics, *T. Sharma and M. Kessentini*, [Sharma and Kessentini \(2021a\)](#)^J;

A.11 Maintainability, technical debt, production performance

Once code is written and its quality analyzed, there are other metrics recorded about its behavior under real-world conditions. This topic is all about the behavior of code once its been released and how changes are measured and occur.

Tools.

- Enlightened Debugging, *X. Li, S. Zhu, M. d’Amorim and A. Orso*, [X. Li and Orso \(2018\)](#)^F;
- CLEVER: Combining Code Metrics with Clone Detection for Just-in-Time Fault Prevention and Resolution in Large Industrial Projects *Nayrolles, Mathieu and Hamou-Lhadj, Abdelwahab* [Nayrolles and Hamou-Lhadj \(2018\)](#)^{FG};

- Supporting Software Architecture Maintenance by Providing Task-Specific Recommendations, *M. Galster, C. Treude and K. Blincoe*, [M. Galster and Blincoe \(2019\)^F](#);
- How Android Developers Handle Evolution-induced API Compatibility Issues: A Large-scale Study, *H. Xia et al.*, [Xia et al. \(2020\)^F](#);
- FeaRS: Recommending Complete Android Method Implementations, *F. Wen, V. Ferrari, E. Aghajani, C. Nagy, M. Lanza and G. Bavota*, [Wen, Ferrari, et al. \(2021\)^F](#);
- Sirius: Static Program Repair with Dependence Graph-Based Systematic Edit Patterns, *K. Noda, H. Yokoyama and S. Kikuchi*, [K. Noda and Kikuchi \(2021\)^{EJ}](#);
- You Look so Different: Finding Structural Clones and Subclones in Java Source Code, *W. Amme, T. S. Heinze and A. Schafer*, [W. Amme and Schäfer \(2021\)^F](#);
- DEAR: A Novel Deep Learning-based Approach for Automated Program Repair, *Y. Li, S. Wang and T. N. Nguyen*, [Y. Li and Nguyen \(2022\)^F](#);
- Tooling for Time- and Space-efficient git Repository Mining, *F. Heseding, W. Scheibel and J. Dollner*, [F. Heseding and Döllner \(2022\)](#);

Psychology / Social Science.

- Embracing Technical Debt, from a Startup Company Perspective, *T. Besker, A. Martini, R. Edirisooriya Lokuge, K. Blincoe and J. Bosch*, [T. Besker and Bosch \(2018\)^{IK}](#);

- Do You Remember This Source Code?, *J. Kruger, J. Wiemann, W. Fenske, G. Saake and T. Leich*, [J. Krüger and Leich \(2018\)](#);
- How the Experience of Development Teams Relates to Assertion Density of Test Classes, *G. Catolino, F. Palomba, A. Zaidman and F. Ferrucci*, [G. Catolino and Ferrucci \(2019\)](#);
- Haste Makes Waste: An Empirical Study of Fast Answers in Stack Overflow, *Y. Lu, X. Mao, M. Zhou, Y. Zhang, T. Wang and Z. Li*, [Y. Lu and Li \(2020\)](#);
- Towards Just-Enough Documentation for Agile Effort Estimation: What Information Should Be Documented?, *J. Pasuksmit, P. Thongtanunam and S. Karunasekera*, [J. Pasuksmit and Karunasekera \(2021\)](#);

Analysis / Broader Studies.

- Was Self-Admitted Technical Debt Removal a Real Removal? An in-Depth Perspective *Zampetti, Fiorella and Serebrenik, Alexander and Di Penta, Massimiliano* [Zampetti, Serebrenik, and Di Penta \(2018\)^J](#);
- How Maintainability Issues of Android Apps Evolve, *I. Malavolta, R. Verdecchia, B. Filipovic, M. Bruntink and P. Lago*, [I. Malavolta and Lago \(2018\)^J](#);
- How Do Code Changes Evolve in Different Platforms? A Mining-Based Investigation, *M. Vigiato, J. Oliveira, E. Figueiredo, P. Jamshidi and C. Kastner*, [M. Vigiato and Kästner \(2019\)](#);
- 9.6 Million Links in Source Code Comments: Purpose, Evolution, and Decay, *H. Hata, C. Treude, R. G. Kula and T. Ishio*, [H. Hata and Ishio \(2019\)^J](#);

- Self-Admitted Technical Debt Removal and Refactoring Actions: Co-Occurrence or More?, *M. Iammarino, F. Zampetti, L. Aversano and M. Di Penta*, [M. Iammarino and Penta \(2019\)^J](#);
- Investigating Context Adaptation Bugs in Code Clones, *M. Mondal, B. Roy, C. K. Roy and K. A. Schneider*, [M. Mondal and Schneider \(2019\)^J](#);
- How Does Modern Code Review Impact Software Design Degradation? An In-depth Empirical Study, *A. Uchoa et al.*, [Uchoa et al. \(2020\)](#);
- On the Recall of Static Call Graph Construction in Practice, *L. Sui, J. Dietrich, A. Tahir and G. Fourtounis*, [L. Sui and Fourtounis \(2020\)^E](#);
- Challenges in Developing Desktop Web Apps: a Study of Stack Overflow and GitHub, *G. L. Scoccia, P. Migliarini and M. Autili*, [G. L. Scoccia and Autili \(2021\)^J](#);
- An Empirical Study of Refactorings and Technical Debt in Machine Learning Systems, *Y. Tang, R. Khatchadourian, M. Bagherzadeh, R. Singh, A. Stewart and A. Raja*, [Y. Tang and Raja \(2021\)^F](#);
- Cross-language Code Coupling Detection: A Preliminary Study on Android Applications, *B. Shen et al.*, [Shen et al. \(2021\)^J](#);
- To What Extent do Deep Learning-based Code Recommenders Generate Predictions by Cloning Code from the Training Set?, *M. Ciniselli, L. Pascarella and G. Bavota*, [M. Ciniselli and Bavota \(2022\)^{FJ}](#);
- Log-based Anomaly Detection with Deep Learning: How Far Are We?, *V. -H. Le and H. Zhang*, [Le and Zhang \(2022\)^{FJ}](#);

Curated Datasets.

- QScored: A Large Dataset of Code Smells and Quality Metrics, *T. Sharma and M. Kessentini*, [Sharma and Kessentini \(2021a\)](#)^J;

A.12 Bugs, faults, and vulnerabilities

This topic deals with problems in the code. This topic is very large, and not all papers related to bugs and other problems which were published in the three conferences surveyed are included here. But a significant and representative number are listed, particularly ones which may have some utility in the advanced scientific computing space.

Tools.

- Cloned Buggy Code Detection in Practice Using Normalized Compression Distance, *T. Ishio, N. Maeda, K. Shibuya and K. Inoue*, [Ishio, Maeda, Shibuya, and Inoue \(2018\)](#);
- Enlightened Debugging, *X. Li, S. Zhu, M. d’Amorim and A. Orso*, [X. Li and Orso \(2018\)](#)^F;
- CLEVER: Combining Code Metrics with Clone Detection for Just-in-Time Fault Prevention and Resolution in Large Industrial Projects *Nayrolles, Mathieu and Hamou-Lhadj, Abdelwahab* [Nayrolles and Hamou-Lhadj \(2018\)](#)^{FG};
- Developers’ Game: A Preliminary Study Concerning a Tool for Automated Developers Assessment, *W. Fracz and J. Dajda*, [Fracz and Dajda \(2018\)](#);
- Recommending Exception Handling Code, *T. Nguyen, P. Vu and T. Nguyen*, [T. Nguyen et al. \(2019b\)](#)^F;

- Learning to Spot and Refactor Inconsistent Method Names, *K. Liu et al.*, [Liu et al. \(2019\)^F](#);
- Ticket Tagger: Machine Learning Driven Issue Classification, *R. Kallis, A. Di Sorbo, G. Canfora and S. Panichella*, [R. Kallis and Panichella \(2019\)^{FJ}](#);
- Learning to Identify Security-Related Issues Using Convolutional Neural Networks, *D. N. Palacio, D. McCrystal, K. Moran, C. Bernal-Cardenas, D. Poshyvanyk and C. Shenefiel*, [D. N. Palacio and Shenefiel \(2019\)^F](#);
- Improving Data Scientist Efficiency with Provenance, *J. Hu, J. Joung, M. Jacobs, K. Z. Gajos and M. I. Seltzer*, [J. Hu and Seltzer \(2020\)](#);
- Graph Neural Network-based Vulnerability Predication, *Q. Feng, C. Feng and W. Hong*, [Q. Feng and Hong \(2020\)^F](#);
- Static source code metrics and static analysis warnings for fine-grained just-in-time defect prediction, *A. Trautsch, S. Herbold and J. Grabowski*, [A. Trautsch and Grabowski \(2020\)^F](#);
- Can I Solve It? Identifying APIs Required to Complete OSS Tasks, *F. Santos, I. Wiese, B. Trinkenreich, I. Steinmacher, A. Sarma and M. A. Gerosa*, [F. Santos and Gerosa \(2021\)^F](#);
- Applying CodeBERT for Automated Program Repair of Java Simple Bugs, *E. Mashhadi and H. Hemmati*, [Mashhadi and Hemmati \(2021\)^F](#);
- It Takes Two to Tango: Combining Visual and Textual Information for Detecting Duplicate Video-Based Bug Reports, *N. Cooper, C. Bernal-Cardenas, O. Chaparro, K. Moran and D. Poshyvanyk*, [N. Cooper and Poshyvanyk \(2021\)](#);

- Sirius: Static Program Repair with Dependence Graph-Based Systematic Edit Patterns, *K. Noda, H. Yokoyama and S. Kikuchi*, [K. Noda and Kikuchi \(2021\)](#)^{EJ};
- Towards Bidirectional Live Programming for Incomplete Programs, *X. Zhang and Z. Hu*, ?;
- LineVD: Statement-level Vulnerability Detection using Graph Neural Networks, *D. Hin, A. Kan, H. Chen and M. A. Babar*, [D. Hin and Babar \(2022\)](#)^{EF};
- Fast Changeset-based Bug Localization with BERT, *A. Ciborowska and K. Damevski*, [Ciborowska and Damevski \(2022\)](#)^F;
- Online Summarizing Alerts through Semantic and Behavior Information, *J. Chen, P. Wang and W. Wang*, [J. Chen and Wang \(2022\)](#)^F;
- DEAR: A Novel Deep Learning-based Approach for Automated Program Repair, *Y. Li, S. Wang and T. N. Nguyen*, [Y. Li and Nguyen \(2022\)](#)^F;

Psychology / Social Science.

- Do Programmers Work at Night or During the Weekend?, *M. Claes, M. V. Mantyla, M. Kuutila and B. Adams*, [M. Claes and Adams \(2018\)](#)^J;
- An Empirical Study Assessing Source Code Readability in Comprehension, *J. Johnson, S. Lubo, N. Yedla, J. Aponte and B. Sharif*, [J. Johnson and Sharif \(2019\)](#)^B;
- Studying the Impact of Policy Changes on Bug Handling Performance, *Z. Abou Khalil*, [Khalil \(2019\)](#);

- The Relationship Between Cognitive Complexity and the Probability of Defects, *B. S. Alqadi*, [Alqadi \(2019\)](#);
- On the Relationship between User Churn and Software Issues *El Zarif, Omar and Da Costa, Daniel Alencar and Hassan, Safwat and Zou, Ying El Zarif, Da Costa, Hassan, and Zou (2020)^{FIJ}*;
- Failures and Fixes: A Study of Software System Incident Response, *J. Sillito and E. Kutomi*, [Sillito and Kutomi \(2020\)](#);
- Mea culpa: How developers fix their own simple bugs differently from other developers, *W. Zhu and M. W. Godfrey*, [Zhu and Godfrey \(2021\)](#);
- How heated is it? Understanding GitHub locked issues, *I. Ferreira, B. Adams and J. Cheng*, [I. Ferreira and Cheng \(2022\)^J](#);

Analysis / Broader Studies.

- Studying Developer Build Issues and Debugger Usage via Timeline Analysis in Visual Studio IDE *Bellman, Christopher and Seet, Ahmad and Baysal, Olga* [Bellman et al. \(2018\)](#);
- When Testing Meets Code Review: Why and How Developers Review Tests, *D. Spadini, M. Aniche, M. -A. Storey, M. Bruntink and A. Bacchelli*, [M. A. S. M. B. D. Spadini M. Aniche and Bacchelli \(2018\)](#);
- How Maintainability Issues of Android Apps Evolve, *I. Malavolta, R. Verdecchia, B. Filipovic, M. Bruntink and P. Lago*, [I. Malavolta and Lago \(2018\)^J](#);

- Do Automated Program Repair Techniques Repair Hard and Important Bugs?, *M. Motwani, S. Sankaranarayanan, R. Just and Y. Brun*, [M. Motwani and Brun \(2018\)](#);
- A Conceptual Replication Study on Bugs that Get Fixed in Open Source Software, *H. Wang and H. Kagdi*, [H. Wang and Kagdi \(2018\)](#)^{FJ};
- How Do Code Changes Evolve in Different Platforms? A Mining-Based Investigation, *M. Vigiato, J. Oliveira, E. Figueiredo, P. Jamshidi and C. Kastner*, [M. Vigiato and Kästner \(2019\)](#);
- Mining Software Defects: Should We Consider Affected Releases?, *S. Yatish, J. Jiarpakdee, P. Thongtanunam and C. Tantithamthavorn*, [S. Yatish and Tantithamthavorn \(2019\)](#)^F;
- Investigating Context Adaptation Bugs in Code Clones, *M. Mondal, B. Roy, C. K. Roy and K. A. Schneider*, [M. Mondal and Schneider \(2019\)](#)^J;
- An Industrial Study on the Differences between Pre-Release and Post-Release Bugs, *R. Rwemalika, M. Kintis, M. Papadakis, Y. Le Traon and P. Lorrach*, [R. Rwemalika and Lorrach \(2019\)](#)^I;
- Challenges in Developing Desktop Web Apps: a Study of Stack Overflow and GitHub, *G. L. Scoccia, P. Migliarini and M. Autili*, [G. L. Scoccia and Autili \(2021\)](#)^J;
- Early Life Cycle Software Defect Prediction. Why? How?, *S. N.C., S. Majumder and T. Menzies*, [S. N.C. and Menzies \(2021\)](#)^{FJ};
- Incorporating Multiple Features to Predict Bug Fixing Time with Neural Networks, *W. Yuan, Y. Xiong, H. Sun and X. Liu*, [W. Yuan and Liu \(2021\)](#)^F;

- Duplicate Bug Report Detection by Using Sentence Embedding and Fine-tuning, *H. Isotani, H. Washizaki, Y. Fukazawa, T. Nomoto, S. Ouji and S. Saito*, [H. Isotani and Saito \(2021\)^F](#);
- Sine-Cosine Algorithm for Software Fault Prediction, *T. Sharma and O. P. Sangwan*, [Sharma and Sangwan \(2021\)^F](#);
- Which bugs are missed in code reviews: An empirical study on SmartSHARK dataset, *F. Khoshnoud, A. R. Nasab, Z. Toudeji and A. Sami*, [F. Khoshnoud and Sami \(2022\)^J](#);
- Beyond Duplicates: Towards Understanding and Predicting Link Types in Issue Tracking Systems, *C. M. Luders, A. Bouraffa and W. Maalej*, [C. M. Lüders and Maalej \(2022\)^J](#);

A.13 Traces, links, and context

Traces, links, and context all refer to the connections between code and its broader environment. For example, error and log messages which are printed out, inline popups in an IDE which provide the documentation to a third party API call, and so on. Essentially, this category has to do with the aspects involved with bridging between the code and everything else.

Tools.

- Context-Aware Software Documentation, *E. Aghajani*, [Aghajani \(2018\)](#);
- Automatic Traceability Maintenance via Machine Learning Classification, *C. Mills, J. Escobar-Avila and S. Haiduc*, [J. E.-A. C. Mills and Haiduc \(2018\)^F](#);
- Ticket Tagger: Machine Learning Driven Issue Classification, *R. Kallis, A. Di Sorbo, G. Canfora and S. Panichella*, [R. Kallis and Panichella \(2019\)^{FJ}](#);

- Improving the Effectiveness of Traceability Link Recovery using Hierarchical Bayesian Networks, *K. Moran et al.*, [Moran et al. \(2020\)^F](#);
- Traceability Transformed: Generating More Accurate Links with Pre-Trained BERT Models, *J. Lin, Y. Liu, Q. Zeng, M. Jiang and J. Cleland-Huang*, [J. Lin and Cleland-Huang \(2021\)^F](#);
- Automated Recovery of Issue-Commit Links Leveraging Both Textual and Non-textual Data, *P. R. Mazrae, M. Izadi and A. Heydarnoori*, [P. R. Mazrae and Heydarnoori \(2021\)^F](#);
- Online Summarizing Alerts through Semantic and Behavior Information, *J. Chen, P. Wang and W. Wang*, [J. Chen and Wang \(2022\)^F](#);

Psychology / Social Science.

- Latent Patterns in Activities: A Field Study of How Developers Manage Context, *S. Chattopadhyay, N. Nelson, Y. Ramirez Gonzalez, A. Amelia Leon, R. Pandita and A. Sarma*, [S. Chattopadhyay and Sarma \(2019\)](#);
- Need for Tweet: How Open Source Developers Talk About Their GitHub Work on Twitter *Fang, Hongbo and Klug, Daniel and Lamba, Hemank and Herbsleb, James and Vasilescu, Bogdan* [Fang et al. \(2020\)^J](#);

Analysis / Broader Studies.

- Predicting Future Developer Behavior in the IDE Using Topic Models, *K. Damevski, H. Chen, D. C. Shepherd, N. A. Kraft and L. Pollock*, [Damevski et al. \(2018\)](#);

- Revisiting "Programmers' Build Errors" in the Visual Studio Context: A Replication Study Using IDE Interaction Traces *Rabbani, Noam and Harvey, Michael S. and Saquif, Sadnan and Gallaba, Keheliya and McIntosh, Shane* [Rabbani, Harvey, Saquif, Gallaba, and McIntosh \(2018\)^{IJ}](#);
- Does the Propagation of Artifact Changes Across Tasks Reflect Work Dependencies?, *C. Mayr-Dorn and A. Egyed*, [Mayr-Dorn and Egyed \(2018\)](#);
- Why are Features Deprecated? An Investigation Into the Motivation Behind Deprecation, *A. A. Sawant, G. Huang, G. Vilen, S. Stojkovski and A. Bacchelli*, [G. V. S. S. A. A. Sawant G. Huang and Bacchelli \(2018\)^F](#);
- Studying the Dialogue Between Users and Developers of Free Apps in the Google Play Store, *S. Hassan, C. Tantithamthavorn, C. -P. Bezemer and A. E. Hassan*, [S. Hassan and Hassan \(2018\)^J](#);
- 9.6 Million Links in Source Code Comments: Purpose, Evolution, and Decay, *H. Hata, C. Treude, R. G. Kula and T. Ishio*, [H. Hata and Ishio \(2019\)^J](#);
- Tracing with Less Data: Active Learning for Classification-Based Traceability Link Recovery, *C. Mills, J. Escobar-Avila, A. Bhattacharya, G. Kondyukov, S. Chakraborty and S. Haiduc*, [A. B. G. K. S. C. C. Mills J. Escobar-Avila and Haiduc \(2019\)^F](#);
- Leveraging Intermediate Artifacts to Improve Automated Trace Link Retrieval, *A. D. Rodriguez, J. Cleland-Huang and D. Falessi*, [A. D. Rodriguez and Falessi \(2021\)^{IJ}](#);
- Log-based Anomaly Detection with Deep Learning: How Far Are We?, *V. -H. Le and H. Zhang*, [Le and Zhang \(2022\)^{FJ}](#);

- Beyond Duplicates: Towards Understanding and Predicting Link Types in Issue Tracking Systems, *C. M. Luders, A. Bouraffa and W. Maalej*, [C. M. Lüders and Maalej \(2022\)](#)^J;

Curated Datasets.

- A Graph-Based Dataset of Commit History of Real-World Android Apps *Geiger, Franz-Xaver and Malavolta, Ivano and Pascarella, Luca and Palomba, Fabio and Di Nucci, Dario and Bacchelli, Alberto* [Geiger et al. \(2018\)](#)^J;
- A Traceability Dataset for Open Source Systems, *M. Hammoudi, C. Mayr-Dorn, A. Mashkooor and A. Egyed*, [M. Hammoudi and Egyed \(2021\)](#)^J;
- Apache Software Foundation Incubator Project Sustainability Dataset, *L. Yin, Z. Zhang, Q. Xuan and V. Filkov*, [L. Yin and Filkov \(2021\)](#)^J;

A.14 Deprecation

Over time code becomes obsolete. This section is all about the (usually deliberate) sunsetting of code or projects which aren't actively used anymore.

Tools.

- Can Automated Impact Analysis Techniques Help Predict Decaying Modules?, *N. Sae-Lim, S. Hayashi and M. Saeki*, [N. Sae-Lim and Saeki \(2019\)](#);

Psychology / Social Science.

- Understanding Developers' Needs on Deprecation as a Language Feature, *A. A. Sawant, M. Aniche, A. van Deursen and A. Bacchelli*, [A. v. D. A. A. Sawant M. Aniche and Bacchelli \(2018\)](#);

Analysis / Broader Studies.

- Why are Features Deprecated? An Investigation Into the Motivation Behind Deprecation, *A. A. Sawant, G. Huang, G. Vilen, S. Stojkovski and A. Bacchelli*, [G. V. S. S. A. A. Sawant G. Huang and Bacchelli \(2018\)^F](#);

A.15 Whole project / repository aspects and status

This topic incorporates aspects of the software development process from a 30,000 foot view. It often incorporates metrics or aspects which project managers or other “bosses” are interested in and responsible for.

Tools.

- Towards Automatically Identifying Paid Open Source Developers *Claes, Maëlick and Mäntylä, Mika and Kuutilla, Miikka and Farooq, Umar* [Claes, Mantyla, Kuutilla, and Farooq \(2018\)^{FJ}](#);
- Interpretation-Enabled Software Reuse Detection Based on a Multi-level Birthmark Model, *X. Xu, Q. Zheng, Z. Yan, M. Fan, A. Jia and T. Liu, X. Xu and Liu* (2021)^G;
- CHAMP: Characterizing Undesired App Behaviors from User Comments Based on Market Policies, *Y. Hu et al.*, [Hu et al. \(2021\)^{FJ}](#);
- Online Summarizing Alerts through Semantic and Behavior Information, *J. Chen, P. Wang and W. Wang*, [J. Chen and Wang \(2022\)^F](#);
- BotHunter: An Approach to Detect Software Bots in GitHub, *A. Abdellatif, M. Wessel, I. Steinmacher, M. A. Gerosa and E. Shihab*, [A. Abdellatif and Shihab \(2022\)^{AFJ}](#);

- Bot Detection in GitHub Repositories, *N. Chidambaram and P. R. Mazrae*, [Chidambaram and Mazrae \(2022\)](#)^{AFJ};
- Tooling for Time- and Space-efficient git Repository Mining, *F. Heseding, W. Scheibel and J. Dollner*, [F. Heseding and Döllner \(2022\)](#);
- Automated Detection of Password Leakage from Public GitHub Repositories, *R. Feng, Z. Yan, S. Peng and Y. Zhang*, [R. Feng and Zhang \(2022\)](#)^{FGJ};
- Starting the InnerSource Journey: Key Goals and Metrics to Measure Collaboration, *D. Izquierdo-Cortazar, J. Alonso-Gutierrez, A. Perez Garcia-Plaza, G. Robles and J. M. Gonzalez-Barahona*, [D. Izquierdo-Cortázar and González-Barahona \(2022\)](#)^I;
- SoCCMiner: A Source Code-Comments and Comment-Context Miner, *M. Sridharan, M. Mantyla, M. Claes and L. Rantala*, [M. Sridharan and Rantala \(2022\)](#);

Psychology / Social Science.

- Embracing Technical Debt, from a Startup Company Perspective, *T. Besker, A. Martini, R. Edirisooriya Lokuge, K. Blincoe and J. Bosch*, [T. Besker and Bosch \(2018\)](#)^{IK};
- Privacy by Designers: Software Developers’ Privacy Mindset, *I. Hadar et al.*, [Hadar et al. \(2018\)](#)^G;
- Latent Patterns in Activities: A Field Study of How Developers Manage Context, *S. Chattopadhyay, N. Nelson, Y. Ramirez Gonzalez, A. Amelia Leon, R. Pandita and A. Sarma*, [S. Chattopadhyay and Sarma \(2019\)](#);

- The Product Backlog, *T. Sedano, P. Ralph and C. Peraire*, [T. Sedano and Péraire \(2019\)^I](#);
- EmoD: An End-to-End Approach for Investigating Emotion Dynamics in Software Development, *K. P. Neupane, K. Cheung and Y. Wang*, [K. P. Neupane and Wang \(2019\)^F](#);
- How Practitioners Perceive Coding Proficiency, *X. Xia, Z. Wan, P. S. Kochhar and D. Lo*, [P. S. K. X. Xia Z. Wan and Lo \(2019\)](#);
- Do as I Do, Not as I Say: Do Contribution Guidelines Match the GitHub Contribution Process?, *O. Elazhary, M. -A. Storey, N. Ernst and A. Zaidman*, [O. Elazhary and Zaidman \(2019\)^J](#);
- A Study of Potential Code Borrowing and License Violations in Java Projects on GitHub *Golubev, Yaroslav and Eliseeva, Maria and Povarov, Nikita and Bryksin, Timofey* [Golubev et al. \(2020\)^{GIJ}](#);
- How Software Practitioners Use Informal Local Meetups to Share Software Engineering Knowledge, *C. Ingram and A. Drachen*, [Ingram and Drachen \(2020\)](#);
- Analysis of Non-Discrimination Policies in the Sharing Economy, *M. Tushev, F. Ebrahimi and A. Mahmoud*, [F. E. M. Tushev and Mahmoud \(2021\)^B](#);
- Playing Planning Poker in Crowds: Human Computation of Software Effort Estimates, *M. Alhamed and T. Storer*, [Alhamed and Storer \(2021\)^{IJ}](#);
- What Makes a Great Maintainer of Open Source Projects?, *E. Dias, P. Meirelles, F. Castor, I. Steinmacher, I. Wiese and G. Pinto*, [E. Dias and Pinto \(2021\)^J](#);

- Geographic Diversity in Public Code Contributions: An Exploratory Large-Scale Study Over 50 Years, *D. Rossi and S. Zacchiroli*, [Rossi and Zacchiroli \(2022\)](#)^{BCJ};
- “This Is Damn Slick!” Estimating the Impact of Tweets on Open Source Project Popularity and New Contributors, *H. Fang, H. Lamba, J. Herbsleb and B. Vasilescu*, [H. Fang and Vasilescu \(2022\)](#)^J;

Analysis / Broader Studies.

- The Evolution of Requirements Practices in Software Startups, *C. Gralha, D. Damian, A. Wasserman, M. Goulao and J. Araujo*, [Gralha et al. \(2018\)](#)^K;
- A Qualitative Study of Variability Management of Control Software for Industrial Automation Systems, *J. Fischer, S. Bougouffa, A. Schlie, I. Schaefer and B. Vogel-Heuser*, [J. Fischer and Vogel-Heuser \(2018\)](#);
- A Design Structure Matrix Approach for Measuring Co-Change-Modularity of Software Products *Benkoczi, Robert and Gaur, Daya and Hossain, Shahadat and Khan, Muhammad A.* [Benkoczi et al. \(2018\)](#);
- Adapting Neural Text Classification for Improved Software Categorization, *A. LeClair, Z. Eberhart and C. McMillan*, [Z. E. A. LeClair and McMillan \(2018\)](#)^F;
- How to Not Get Rich: An Empirical Study of Donations in Open Source, *C. Overney, J. Meinicke, C. Kastner and B. Vasilescu*, [C. Overney and Vasilescu \(2020\)](#)^J;
- What Constitutes Software? An Empirical, Descriptive Study of Artifacts *Pfeiffer, Rolf-Helge* [Pfeiffer \(2020\)](#);

- Challenges in Developing Desktop Web Apps: a Study of Stack Overflow and GitHub, *G. L. Scoccia, P. Migliarini and M. Autili*, [G. L. Scoccia and Autili \(2021\)^J](#);
- Which contributions count? Analysis of attribution in open source, *J. -G. Young, A. Casari, K. McLaughlin, M. Z. Trujillo, L. Hebert-Dufresne and J. P. Bagrow*, [J. G. Young and Bagrow \(2021\)^J](#);
- An Empirical Study on the Survival Rate of GitHub Projects, *A. Ait, J. L. C. Izquierdo and J. Cabot*, [A. Ait and Cabot \(2022\)^J](#);
- GitRank: A Framework to Rank GitHub Repositories, *N. Hasabnis*, [Hasabnis \(2022\)^F](#);
- Big Data = Big Insights? Operationalising Brooks' Law in a Massive GitHub Data Set, *C. Gote, P. Mavrodiev, F. Schweitzer and I. Scholtes*, [C. Gote and Scholtes \(2022\)^J](#);
- The Art and Practice of Data Science Pipelines: A Comprehensive Study of Data Science Pipelines In Theory, In-The-Small, and In-The-Large, *S. Biswas, M. Wardat and H. Rajan*, [S. Biswas and Rajan \(2022\)^F](#);

Curated Datasets.

- Threats of Aggregating Software Repository Data *M. P. Robillard, M. Nassif and S. McIntosh*, [M. P. Robillard and McIntosh \(2018\)^J](#);
- A Graph-Based Dataset of Commit History of Real-World Android Apps *Geiger, Franz-Xaver and Malavolta, Ivano and Pascarella, Luca and Palomba, Fabio and Di Nucci, Dario and Bacchelli, Alberto* [Geiger et al. \(2018\)^J](#);

- Apache Software Foundation Incubator Project Sustainability Dataset, *L. Yin, Z. Zhang, Q. Xuan and V. Filkov*, [L. Yin and Filkov \(2021\)](#)^J;
- A Large-scale Dataset of (Open Source) License Text Variants, *S. Zacchiroli, Zacchiroli (2022)*^J;
- A Versatile Dataset of Agile Open Source Software Projects, *V. Tawosi, A. Al-Subaihini, R. Moussa and F. Sarro*, [V. Tawosi and Sarro \(2022\)](#)^J;

A.16 Human and team dynamics

Software engineering is technical, but also very human. This topic explores the human element of the engineering process.

Tools.

- Who’s This? Developer Identification Using IDE Event Data *Wilkie, John and Halabi, Ziad Al and Karaoglu, Alperen and Liao, Jiafeng and Ndungu, George and Ragkhitwetsagul, Chaiyong and Paixao, Matheus and Krinke, Jens* [Wilkie et al. \(2018\)](#)^F;
- Developers’ Game: A Preliminary Study Concerning a Tool for Automated Developers Assessment, *W. Fracz and J. Dajda*, [Fracz and Dajda \(2018\)](#);
- Towards Automatically Identifying Paid Open Source Developers *Claes, Maëlick and Mäntylä, Mika and Kuutila, Miikka and Farooq, Umar* [Claes et al. \(2018\)](#)^{FJ};
- Detection of Hidden Feature Requests from Massive Chat Messages via Deep Siamese Network, *L. Shi, M. Xing, M. Li, Y. Wang, S. Li and Q. Wang*, [L. Shi and Wang \(2020\)](#)^F;

- Robin: A Voice Controlled Virtual Teammate for Software Developers and Teams, *B. da Silva, C. Hebert, A. Rawka and S. Sereesathien*, [B. da Silva and Sereesathien \(2020\)](#)^{AF};
- CHAMP: Characterizing Undesired App Behaviors from User Comments Based on Market Policies, *Y. Hu et al.*, [Hu et al. \(2021\)](#)^{FJ};
- Representation of Developer Expertise in Open Source Software, *T. Dey, A. Karnauch and A. Mockus*, [T. Dey and Mockus \(2021\)](#)^{FJ};

Psychology / Social Science.

- Embracing Technical Debt, from a Startup Company Perspective, *T. Besker, A. Martini, R. Edirisooriya Lokuge, K. Blincoe and J. Bosch*, [T. Besker and Bosch \(2018\)](#)^{IK};
- Communicative Intention in Code Review Questions, *F. Ebert, F. Castor, N. Novielli and A. Serebrenik*, [F. Ebert and Serebrenik \(2018\)](#);
- Empirical Study on the Relationship between Developer’s Working Habits and Efficiency *Rodriguez, Ariel and Tanaka, Fumiya and Kamei, Yasutaka* [Rodriguez et al. \(2018\)](#);
- Do Programmers Work at Night or During the Weekend?, *M. Claes, M. V. Mantyla, M. Kuuttila and B. Adams*, [M. Claes and Adams \(2018\)](#)^J;
- How Modern News Aggregators Help Development Communities Shape and Share Knowledge, *M. Aniche et al.*, [Aniche et al. \(2018\)](#)^J;
- What Makes a Great Manager of Software Engineers?, *E. Kalliamvakou, C. Bird, T. Zimmermann, A. Begel, R. DeLine and D. M. German*, [E. Kalliamvakou and German \(2018\)](#)^I;

- Older Adults and Hackathons: A Qualitative Study, *W. Kopec, B. Balcerzak, R. Nielek, G. Kowalik, A. Wierzbicki and F. Casati*, [W. Kopec and Casati \(2018\)^J](#);
- Data Scientists in Software Teams: State of the Art and Challenges, *M. Kim, T. Zimmermann, R. DeLine and A. Begel*, [M. Kim and Begel \(2018\)^I](#);
- Team Maturity in Agile Software Development: The Impact on Productivity, *S. L. Ramirez-Mora and H. Oktaba*, [Ramirez-Mora and Oktaba \(2018\)](#);
- How the Experience of Development Teams Relates to Assertion Density of Test Classes, *G. Catolino, F. Palomba, A. Zaidman and F. Ferrucci*, [G. Catolino and Ferrucci \(2019\)](#);
- Latent Patterns in Activities: A Field Study of How Developers Manage Context, *S. Chattopadhyay, N. Nelson, Y. Ramirez Gonzalez, A. Amelia Leon, R. Pandita and A. Sarma*, [S. Chattopadhyay and Sarma \(2019\)](#);
- The Product Backlog, *T. Sedano, P. Ralph and C. Péraire*, [T. Sedano and Péraire \(2019\)^I](#);
- How Practitioners Perceive Coding Proficiency, *X. Xia, Z. Wan, P. S. Kochhar and D. Lo*, [P. S. K. X. Xia Z. Wan and Lo \(2019\)](#);
- Investigating the Effects of Gender Bias on GitHub, *N. Imtiaz, J. Middleton, J. Chakraborty, N. Robson, G. Bai and E. Murphy-Hill*, [N. Imtiaz and Murphy-Hill \(2019\)^{DJ}](#);
- Socio-Technical Work-Rate Increase Associates With Changes in Work Patterns in Online Projects, *F. Sarker, B. Vasilescu, K. Blincoe and V. Filkov*, [F. Sarker and Filkov \(2019\)^J](#);

- Do as I Do, Not as I Say: Do Contribution Guidelines Match the GitHub Contribution Process?, *O. Elazhary, M. -A. Storey, N. Ernst and A. Zaidman, O. Elazhary and Zaidman (2019)^J*;
- FLOSS Participants’ Perceptions About Gender and Inclusiveness: A Survey, *A. Lee and J. C. Carver, Lee and Carver (2019)^{DJ}*;
- Going Farther Together: The Impact of Social Capital on Sustained Participation in Open Source, *H. S. Qiu, A. Nolte, A. Brown, A. Serebrenik and B. Vasilescu, H. S. Qiu and Vasilescu (2019)^{DJ}*;
- Why Do Episodic Volunteers Stay in FLOSS Communities?, *A. Barcomb, K. -J. Stol, D. Riehle and B. Fitzgerald, A. Barcomb and Fitzgerald (2019)^J*;
- A Study of Potential Code Borrowing and License Violations in Java Projects on GitHub *Golubev, Yaroslav and Eliseeva, Maria and Povarov, Nikita and Bryksin, Timofey Golubev et al. (2020)^{GIJ}*;
- Need for Tweet: How Open Source Developers Talk About Their GitHub Work on Twitter *Fang, Hongbo and Klug, Daniel and Lamba, Hemank and Herbsleb, James and Vasilescu, Bogdan Fang et al. (2020)^J*;
- How Software Practitioners Use Informal Local Meetups to Share Software Engineering Knowledge, *C. Ingram and A. Drachen, Ingram and Drachen (2020)*;
- A Tale from the Trenches: Cognitive Biases and Software Development, *S. Chattopadhyay et al., Chattopadhyay et al. (2020)^B*;
- Engineering Gender-Inclusivity into Software: Ten Teams’ Tales from the Trenches, *C. Hilderbrand et al., Hilderbrand et al. (2020)^D*;

- Remote Pair Programming in Virtual Reality, *J. Dominic, B. Tubre, C. Ritter, J. Houser, C. Smith and P. Rodeghero*, [J. Dominic and Rodeghero \(2020\)](#);
- Towards Just-Enough Documentation for Agile Effort Estimation: What Information Should Be Documented?, *J. Pasuksmit, P. Thongtanunam and S. Karunasekera*, [J. Pasuksmit and Karunasekera \(2021\)](#);
- Mining DEV for social and technical insights about software development, *M. Papoutsoglou, J. Wachs and G. M. Kapitsaki*, [M. Papoutsoglou and Kapitsaki \(2021\)](#)^J;
- Ignorance and Prejudice” in Software Fairness, *J. M. Zhang and M. Harman*, [J. M. Zhang and Harman \(2021\)](#)^F;
- Playing Planning Poker in Crowds: Human Computation of Software Effort Estimates, *M. Alhamed and T. Storer*, [Alhamed and Storer \(2021\)](#)^{IJ};
- How Was Your Weekend?” Software Development Teams Working From Home During COVID-19, *C. Miller, P. Rodeghero, M. -A. Storey, D. Ford and T. Zimmermann*, [C. Miller and Zimmermann \(2021\)](#)^I;
- How Gamification Affects Software Developers: Cautionary Evidence from a Natural Experiment on GitHub, *L. Moldon, M. Strohmaier and J. Wachs*, [L. Moldon and Wachs \(2021\)](#)^J;
- Clustering, Separation, and Connection: A Tale of Three Characteristics, *S. Datta, A. Mysore, H. Wira and S. Sarkar*, [S. Datta and Sarkar \(2021\)](#)^E;
- What Makes a Great Maintainer of Open Source Projects?, *E. Dias, P. Meirelles, F. Castor, I. Steinmacher, I. Wiese and G. Pinto*, [E. Dias and Pinto \(2021\)](#)^J;

- This Is Damn Slick!” Estimating the Impact of Tweets on Open Source Project Popularity and New Contributors, *H. Fang, H. Lamba, J. Herbsleb and B. Vasilescu*, [H. Fang and Vasilescu \(2022\)^J](#);
- Hashing It Out: A Survey of Programmers’ Cannabis Usage, Perception, and Motivation, *M. Endres, K. Boehnke and W. Weimer*, [M. Endres and Weimer \(2022\)](#);
- What Makes Effective Leadership in Agile Software Development Teams?, *L. Gren and P. Ralph*, [Gren and Ralph \(2022\)](#);
- Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process, *N. Nahar, S. Zhou, G. Lewis and C. Kastner*, [N. Nahar and Kästner \(2022\)^F](#);
- An Exploratory Study of Productivity Perceptions in Software Teams, *A. Ruvimova et al.*, [Ruvimova et al. \(2022\)](#);
- A Grounded Theory of Coordination in Remote-First and Hybrid Software Teams, *R. E. de Souza Santos and P. Ralph*, [de Souza Santos and Ralph \(2022\)](#);

Analysis / Broader Studies.

- Challenges in Developing Desktop Web Apps: a Study of Stack Overflow and GitHub, *G. L. Scoccia, P. Migliarini and M. Autili*, [G. L. Scoccia and Autili \(2021\)^J](#);
- Which contributions count? Analysis of attribution in open source, *J. -G. Young, A. Casari, K. McLaughlin, M. Z. Trujillo, L. Hebert-Dufresne and J. P. Bagrow*, [J. G. Young and Bagrow \(2021\)^J](#);

- The Shifting Sands of Motivation: Revisiting What Drives Contributors in Open Source, *M. Gerosa et al.*, [Gerosa et al. \(2021\)](#)^J;
- Report From The Trenches A Case Study In Modernizing Software Development Practices, *M. H. Houekpetodji, N. Anquetil, S. Ducasse, F. Djareddir and J. Sudich*, [M. H. Houekpetodji and Sudich \(2021\)](#)^I;
- DISCO: A Dataset of Discord Chat Conversations for Software Engineering Research, *K. M. Subash, L. P. Kumar, S. L. Vadlamani, P. Chatterjee and O. Baysal*, [K. M. Subash and Baysal \(2022\)](#)^A;

A.17 Machine learning foundations

In addition to the cross cutting theme of applied machine learning, which interweaves itself with many tools, there are several works related to research in the domain-specific foundational algorithms needed for machine learning to later be successful in the software engineering tool space. This topic investigates these works.

Tools.

- TUNA: TUning Naturalness-Based Analysis, *M. Jimenez, C. Maxime, Y. Le Traon and M. Papadakis*, [M. Jimenez and Papadakis \(2018b\)](#)^F;
- Natural Language or Not (NLON): A Package for Software Engineering Text Analysis Pipeline *Mäntylä, Mika V. and Calefato, Fabio and Claes, Maelick* [Mantyla et al. \(2018\)](#)^F;
- Word Embeddings for the Software Engineering Domain *Efstathiou, Vasiliki and Chatzilenas, Christos and Spinellis, Diomidis* [Efstathiou et al. \(2018\)](#)^F;

- Learning to Identify Security-Related Issues Using Convolutional Neural Networks, *D. N. Palacio, D. McCrystal, K. Moran, C. Bernal-Cardenas, D. Poshyanyk and C. Shenefiel*, [D. N. Palacio and Shenefiel \(2019\)^F](#);
- Import2vec Learning Embeddings for Software Libraries *Theeten, Bart and Vandeputte, Frederik and Van Cutsem, Tom* [Theeten et al. \(2019\)^F](#);
- Retrieval-based Neural Source Code Summarization, *J. Zhang, X. Wang, H. Zhang, H. Sun and X. Liu*, [J. Zhang and Liu \(2020\)^F](#);
- Score-Based Automatic Detection and Resolution of Syntactic Ambiguity in Natural Language Requirements, *M. Osama, A. Zaki-Ismail, M. Abdelrazek, J. Grundy and A. Ibrahim*, [M. Osama and Ibrahim \(2020\)^F](#);
- Embedding Java Classes with Code2vec: Improvements from Variable Obfuscation *Compton, Rhys and Frank, Eibe and Patros, Panos and Koay, Abigail* [Compton et al. \(2020\)^F](#);
- CC2Vec: Distributed Representations of Code Changes, *T. Hoang, H. J. Kang, D. Lo and J. Lawall*, [T. Hoang and Lawall \(2020\)^F](#);
- POSIT: Simultaneously Tagging Natural and Programming Languages, *P. - P. Partachi, S. K. Dash, C. Treude and E. T. Barr*, [P. P. Pârachi and Barr \(2020\)^F](#);
- Repo2Vec: A Comprehensive Embedding Approach for Determining Repository Similarity, *M. O. F. Rokon, P. Yan, R. Islam and M. Faloutsos*, [M. O. F. Rokon and Faloutsos \(2021\)^{FJ}](#);
- Fast Changeset-based Bug Localization with BERT, *A. Ciborowska and K. Damevski*, [Ciborowska and Damevski \(2022\)^F](#);

- GraphCode2Vec: Generic Code Embedding via Lexical and Program Dependence Analyses, *W. Ma et al.*, [Ma et al. \(2022\)](#)^F;
- Manas: Mining Software Repositories to Assist AutoML, *G. Nguyen, M. J. Islam, R. Pan and H. Rajan*, [G. Nguyen and Rajan \(2022\)](#)^F;

Psychology / Social Science.

- Ignorance and Prejudice” in Software Fairness, *J. M. Zhang and M. Harman*, [J. M. Zhang and Harman \(2021\)](#)^F;

Analysis / Broader Studies.

- On the Impact of Tokenizer and Parameters on N-Gram Based Code Analysis, *M. Jimenez, C. Maxime, Y. Le Traon and M. Papadakis*, [M. Jimenez and Papadakis \(2018a\)](#)^F;
- Semantic Source Code Models Using Identifier Embeddings *Efstathiou, Vasiliki and Spinellis, Diomidis* [Efstathiou and Spinellis \(2019\)](#)^{FJ};
- Assessing Generalizability of CodeBERT, *X. Zhou, D. Han and D. Lo*, [X. Zhou and Lo \(2021\)](#)^F;
- Log-based Anomaly Detection with Deep Learning: How Far Are We?, *V. -H. Le and H. Zhang*, [Le and Zhang \(2022\)](#)^{FJ};
- Does Configuration Encoding Matter in Learning Software Performance? An Empirical Study on Encoding Schemes, *J. Gong and T. Chen*, [Gong and Chen \(2022\)](#)^F;
- What Do They Capture? - A Structural Analysis of Pre-Trained Language Models for Source Code, *Y. Wan, W. Zhao, H. Zhang, Y. Sui, G. Xu and H. Jin*, [Wan et al. \(2022\)](#)^{EF};

- Multilingual training for Software Engineering, *T. Ahmed and P. Devanbu*, [Ahmed and Devanbu \(2022\)](#)^F;
- The Art and Practice of Data Science Pipelines: A Comprehensive Study of Data Science Pipelines In Theory, In-The-Small, and In-The-Large, *S. Biswas, M. Wardat and H. Rajan*, [S. Biswas and Rajan \(2022\)](#)^F;

Curated Datasets.

- A Time Series-Based Dataset of Open-Source Software Evolution, *B. L. Sousa, M. A. S. Bigonha, K. A. M. Ferreira and G. C. Franco*, [Sousa et al. \(2022\)](#)^J;

A.18 Software Engineering / Repository Mining Research Meta Analysis

This topic covers research on research, which includes things like literature reviews, papers on methodology, and so forth.

Tools.

- Sampling Projects in GitHub for MSR Studies, *O. Dabic, E. Aghajani and G. Bavota*, [O. Dabic and Bavota \(2021\)](#)^J;
- SoCCMiner: A Source Code-Comments and Comment-Context Miner, *M. Sridharan, M. Mantyla, M. Claes and L. Rantala*, [M. Sridharan and Rantala \(2022\)](#);

Psychology / Social Science.

- A Study on the Prevalence of Human Values in Software Engineering Publications, 2015 - 2018, *H. Perera et al.*, [Perera et al. \(2020\)](#)^B;
- Social Science Theories in Software Engineering Research, *T. Lorey, P. Ralph and M. Felderer*, [T. Lorey and Felderer \(2022\)](#);

Analysis / Broader Studies.

- Synthesizing Qualitative Research in Software Engineering: A Critical Review, *X. Huang, H. Zhang, X. Zhou, M. Ali Babar and S. Yang*, [X. Huang and Yang \(2018\)](#);
- What Constitutes Software? An Empirical, Descriptive Study of Artifacts *Pfeiffer, Rolf-Helge* [Pfeiffer \(2020\)](#);
- Ethical Mining: A Case Study on MSR Mining Challenges *Gold, Nicolas E. and Krinke, Jens* [Gold and Krinke \(2020\)^B](#);
- Empirical Standards for Repository Mining, *P. Chatterjee, T. Sharma and P. Ralph*, [T. S. P. Chatterjee and Ralph \(2022\)](#);

Curated Datasets.

- The Unsolvable Problem or the Unheard Answer? A Dataset of 24,669 Open-Source Software Conference Talks, *K. Truong, C. Miller, B. Vasilescu and C. Kastner*, [K. Truong and Kästner \(2022\)^J](#);

A.19 Input-Output Types of Machine Learning Tools

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
500+ Times Faster than Deep Learning: A Case Study Exploring Faster Methods for Text Mining Stackoverflow	Majumder et al. (2018)	Text (Stack Overflow)	Classification
A Gold Standard for Emotion Annotation in Stack Overflow	Novielli et al. (2018)	Text (Stack Overflow)	Classification
A Practical Approach to the Automatic Classification of Security-Relevant Commits	Sabetta and Bezzi (2018)	Text (Commits)	Classification
A Simple NLP-Based Approach to Support Onboarding and Retention in Open Source Communities	Stanik, Montgomery, Martens, Fucci, and Maalej (2018)	Text (Issues)	Classification
Achieving Reliable Sentiment Analysis in the Software Engineering Domain using BERT	E. Biswas and Vijay-Shanker (2020)	Text (Stack Overflow)	Classification

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
Applying CodeBERT for Automated Program Repair of Java Simple Bugs	Mashhadi and Hemmati (2021)	Text (Bug-ridden code)	Classification
Attention-based model for predicting question relatedness on Stack Overflow	Pei et al. (2021)	Text (Stack Overflow)	Classification
Automated Assertion Generation via Information Retrieval and Its Integration with Deep learning	Yu et al. (2022)	Text (Code)	Text (Code)
Automated Characterization of Software Vulnerabilities	D. Gonzalez and Mirakhorli (2019)	Text (Jargon)	Classification
Automated Detection of Password Leakage from Public GitHub Repositories	R. Feng and Zhang (2022)	Text (GitHub/Repo artifacts)	Classification
Automated Recovery of Issue-Commit Links Leveraging Both Textual and Non-textual Data	P. R. Mazrae and Heydarnoori (2021)	Text (Issues and Commits); Numbers	Classification

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
Automatic Extraction of Opinion-Based Q&A from Online Developer Chats	K. D. P. Chatterjee and Pollock (2021)	Text (Jargon)	Classification
Automatic Identification of Rollback Edit with Reasons in Stack Overflow Q&A Site	S. Mondal and Roy (2020)	Text (Jargon)	Classification
Automatic Traceability Maintenance via Machine Learning Classification	J. E.-A. C. Mills and Haiduc (2018)	Text (GitHub/Repo artifacts)	Classification
Automatic Web Testing Using Curiosity-Driven Reinforcement Learning	Zheng et al. (2021)	Text (Code)	Complex – See Paper; Classification
Automatically Generating Precise Oracles from Structured Natural Language Specifications	Motwani and Brun (2019)	Text (Jargon)	Text (Code)
Bot Detection in GitHub Repositories	Chidambaram and Mazrae (2022)	Text (GitHub/Repo artifacts)	Classification
BotHunter: An Approach to Detect Software Bots in GitHub	A. Abdellatif and Shihab (2022)	Text (GitHub/Repo artifacts)	Classification

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
BugListener: Identifying and Synthesizing Bug Reports from Collaborative Live Chats	Shi et al. (2022)	Text (Jargon)	Text (Jargon); Classification
Can I Solve It? Identifying APIs Required to Complete OSS Tasks	F. Santos and Gerosa (2021)	Text (GitHub/Repo artifacts)	Text (Code); Classification
Caspar: Extracting and Synthesizing User Stories of Problems from App Reviews	Guo and Singh (2020)	Text (Natural Language)	Classification
CHAMP: Characterizing Undesired App Behaviors from User Comments Based on Market Policies	Hu et al. (2021)	Text (Natural Language)	Classification
CLEVER: Combining Code Metrics with Clone Detection for Just-in-Time Fault Prevention and Resolution in Large Industrial Projects	Nayrolles and Hamou-Lhadj (2018)	Text (Code)	Classification

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
DEAR: A Novel Deep Learning-based Approach for Automated Program Repair	Y. Li and Nguyen (2022)	Text (Code)	Text (Code)
Deep Learning Anti-Patterns from Code Metrics History	Barbez et al. (2019)	Text (Code)	Classification
Detecting and Characterizing Bots That Commit Code	T. Dey et al. (2020)	Text (GitHub/Repo artifacts)	Classification
Detection of Hidden Feature Requests from Massive Chat Messages via Deep Siamese Network	L. Shi and Wang (2020)	Text (Natural Language)	Text (Jargon)
Developer-Driven Code Smell Prioritization	Pecorelli et al. (2020)	Text (Code)	Classification
Dialogue Management for Interactive API Search	Eberhart and McMillan (2021)	Text (Natural Language)	Text (Code)
Efficient Bug Triage For Industrial Environments	W. Zhang (2020)	Text (Jargon)	Classification

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
EmoD: An End-to-End Approach for Investigating Emotion Dynamics in Software Development	K. P. Neupane and Wang (2019)	Text (Jargon)	Classification
Ensemble Models for Neural Source Code Summarization of Subroutines	A. B. A. LeClair and McMillan (2021)	Text (Code)	Text (Natural Language)
Fast and Memory-Efficient Neural Code Completion	Svyatkovskiy et al. (2021)	Text (Code)	Text (Code)
Fast Changeset-based Bug Localization with BERT	Ciborowska and Damevski (2022)	Text (Code)	Complex – See Paper; Classification
FeaRS: Recommending Complete Android Method Implementations	Wen, Ferrari, et al. (2021)	Text (Code)	Text (Code)
FIRA: Fine-Grained Graph-Based Code Change Representation for Automated Commit Message Generation	Dong et al. (2022)	Text (Code)	Text (Commits)

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
Graph Neural Network-based Vulnerability Predication	Q. Feng and Hong (2020)	Text (Code)	Classification
Graph-Based Fuzz Testing for Deep Learning Inference Engines	W. Luo and Chen (2021)	Text (Code); Complex – see paper	Complex – See Paper; Classification
HARP: Holistic Analysis for Refactoring Python-Based Analytics Programs	W. Zhou and Shen (2020)	Text (Code)	Complex – See Paper; Classification
How (Not) to Find Bugs: The Interplay Between Merge Conflicts, Co-Changes, and Bugs	Amaral et al. (2020)	Text (Commits with Bugs)	Classification
How Android Developers Handle Evolution-induced API Compatibility Issues: A Large-scale Study	Xia et al. (2020)	Text (Code)	Classification
Human, bot or both? A study on the capabilities of classification models on mixed accounts	N. Cassee and Serebrenik (2021)	Text (GitHub/Repo artifacts)	Classification

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
Improving Bug Triaging with High Confidence Predictions at Ericsson	A. Sarkar and Bartalos (2019)	Text (Jargon)	Classification
Improving the Effectiveness of Traceability Link Recovery using Hierarchical Bayesian Networks	Moran et al. (2020)	Text (GitHub/Repo artifacts)	Classification
Incorporating Multiple Features to Predict Bug Fixing Time with Neural Networks	W. Yuan and Liu (2021)	Complex – see paper	Classification
Learning Off-By-One Mistakes: An Empirical Study	H. Sellik and Aniche (2021)	Text (Code)	Classification
Learning to Identify Security-Related Issues Using Convolutional Neural Networks	D. N. Palacio and Shenefiel (2019)	Text (Jargon)	Classification
Learning to Spot and Refactor Inconsistent Method Names	Liu et al. (2019)	Text (Code)	Text (Code)

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
LineVD: Statement-level Vulnerability Detection using Graph Neural Networks	D. Hin and Babar (2022)	Text (Code)	Classification
Linking Source Code to Untangled Change Intents	X. Liu and Ng (2018)	Text (Code)	Classification
Manas: Mining Software Repositories to Assist AutoML	G. Nguyen and Rajan (2022)	Text (GitHub/Repo artifacts)	Complex – See Paper; Text (Code)
Natural Language or Not (NLON): A Package for Software Engineering Text Analysis Pipeline	Mantyla et al. (2018)	Text	Classification
On Learning Meaningful Code Changes Via Neural Machine Translation	Tufano et al. (2019)	Text (Pull Requests)	Text (Code)
On the Importance of Building High-quality Training Datasets for Neural Code Search	Sun et al. (2022)	Text (GitHub/Repo artifacts)	Text (Code)

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
Online Summarizing Alerts through Semantic and Behavior Information	J. Chen and Wang (2022)	Text (Jargon)	Text (Natural Language)
Personalized Code Recommendation	T. Nguyen et al. (2019a)	Text (Code)	Text (Code)
Planning for Untangling: Predicting the Difficulty of Merge Conflicts	R. L. C. Brindescu I. Ahmed and Sarma (2020)	Text (GitHub/Repo artifacts)	Classification
POSIT: Simultaneously Tagging Natural and Programming Languages	P. P. Pârachi and Barr (2020)	Text	Classification
Predicting Developers' IDE Commands with Machine Learning	Bulmer et al. (2018)	Complex – see paper	Classification
Predicting Software Maintainability in Object-Oriented Systems Using Ensemble Techniques	H. Alsolai and Nassar (2018)	Numbers (Metrics from Code)	Regression
Quickly Generating Diverse Valid Test Inputs with Reinforcement Learning	S. Reddy and Sen (2020)	Text (Code)	Complex – See Paper

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
Recommending Exception Handling Code	T. Nguyen et al. (2019b)	Text (Code)	Classification
Recommending Good First Issues in GitHub OSS Projects	W. Xiao and Zhou (2022)	Text (GitHub/Repo artifacts)	Classification
Representation of Developer Expertise in Open Source Software	T. Dey and Mockus (2021)	Text (GitHub/Repo artifacts)	Complex – See Paper
Retrieval-based Neural Source Code Summarization	J. Zhang and Liu (2020)	Text (Code)	Text (Natural Language)
Robin: A Voice Controlled Virtual Teammate for Software Developers and Teams	B. da Silva and Sereesathien (2020)	Text (Natural Language)	Complex – See Paper
RoPGen: Towards Robust Code Authorship Attribution via Automatic Coding Style Transformation	Z. Li and Xu (2022)	Text (Code)	Classification

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
SAPIENTML: Synthesizing Machine Learning Pipelines by Learning from Human-Written Solutions	Saha et al. (2022)	Text (Natural Language)	Complex – See Paper
Score-Based Automatic Detection and Resolution of Syntactic Ambiguity in Natural Language Requirements	M. Osama and Ibrahim (2020)	Text (Natural Language)	Complex – See Paper
Senatus - A Fast and Accurate Code-to-Code Recommendation Engine	Silavong et al. (2022)	Text (Code)	Text (Code)
Siri, Write the Next Method	Wen, Aghajani, et al. (2021)	Text (Natural Language)	Text (Code)
SPT-Code: Sequence-to-Sequence Pre-Training for Learning Source Code Representations	Niu et al. (2022)	Text (Code)	Text (Code)
Static source code metrics and static analysis warnings for fine-grained just-in-time defect prediction	A. Trautsch and Grabowski (2020)	Metrics from Code	Classification

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
Statistical Learning of API Fully Qualified Names in Code Snippets of Online Forums	H. Phan and Nguyen (2018)	Text (Jargon)	Text (Code)
Striking a Balance: Pruning False-Positives from Static Call Graphs	A. Utture and Palsberg (2022)	Text (Code)	Classification
Suggesting Natural Method Names to Check Name Consistencies	S. Nguyen et al. (2020)	Text (Code)	Text (Natural Language), Text (Code)
Supporting Software Architecture Maintenance by Providing Task-Specific Recommendations	M. Galster and Blincoe (2019)	Text (GitHub/Repo artifacts)	Classification
Ticket Tagger: Machine Learning Driven Issue Classification	R. Kallis and Panichella (2019)	Text (Issues)	Classification
Toward Less Hidden Cost of Code Completion with Acceptance and Ranking Models	J. Li, Huang, Li, Yao, and Tan (2021)	Text (Code)	Text (Code)

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
Towards Automatically Identifying Paid Open Source Developers	Claes et al. (2018)	Text (GitHub/Repo artifacts)	Classification
Towards Automating Code Review Activities	M. T. D. P. R. Tufano and L. Pascarella and Bavota (2021)	Text (GitHub/Repo artifacts)	Complex – See paper
Towards Reliable Agile Iterative Planning via Predicting Documentation Changes of Work Items	J. Pasuksmit and Karunasekera (2022)	Text (GitHub/Repo artifacts)	Complex – See paper
Traceability Transformed: Generating More Accurate Links with Pre-Trained BERT Models	J. Lin and Cleland-Huang (2021)	Text (GitHub/Repo artifacts)	Classification
Type4Py: Practical Deep Similarity Learning-Based Type Inference for Python	Mir et al. (2022)	Text (Code)	Text (Code)
Using Pre-Trained Models to Boost Code Review Automation	A. M. L. P. D. P. R. Tufano, S. Masiero and Bavota (2022)	Text (GitHub/Repo artifacts)	Classification
Where is Your App Frustrating Users?	Y. Wang and Wang (2022)	Text (Natural Language)	Complex – See Paper

Table 27. Primary input and output types of all machine learning model driven tools examined in this report.

Title	Citation	Input	Output
Who's This? Developer Identification Using IDE Event Data	Wilkie et al. (2018)	Complex – see paper	Classification
You Look so Different: Finding Structural Clones and Subclones in Java Source Code	W. Amme and Schäfer (2021)	Text (Code)	Classification

APPENDIX B
RSE DEPENDENCIES

Table 28. Dependencies used by more than 10 RSE repositories. The % is in reference to the number of repositories which utilize the corresponding build system manager.

Dependency	Used by # of Repositories	Used by % of Repositories which use {actions,pip,npm,etc.}
actions:actions/checkout	3721	93.42
actions:actions/setup-python	1407	35.33
actions:actions/upload-artifact	873	21.92
actions:actions/cache	579	14.54
actions:codecov/codecov-action	422	10.60
actions:docker/login-action	346	8.69
actions:actions/setup-node	332	8.34
actions:actions/download-artifact	331	8.31
actions:pypa/gh-action-pypi-publish	321	8.06
actions:docker/build-push-action	305	7.66
actions:github/codeql-action/analyze	240	6.03
actions:github/codeql-action/init	240	6.03
actions:docker/setup-buildx-action	237	5.95
actions:actions/setup-java	214	5.37
actions:peaceiris/actions-gh-pages	201	5.05

actions:conda-incubator/setup-miniconda	198	4.97
actions:github/codeql-action/autobuild	195	4.90
actions:r-lib/actions/setup-r	163	4.09
actions:docker/setup-qemu-action	154	3.87
actions:actions/setup-go	152	3.82
actions:JamesIves/github-pages-deploy-action	139	3.49
actions:docker/metadata-action	137	3.44
actions:r-lib/actions/setup-pandoc	128	3.21
actions:julia-actions/setup-julia	109	2.74
actions:softprops/action-gh-release	101	2.54
actions:r-lib/actions/setup-r-dependencies	93	2.33
actions:actions/deploy-pages	91	2.28
actions:actions/upload-pages-artifact	89	2.23
actions:actions/stale	85	2.13
actions:julia-actions/julia-runtest	80	2.01
actions:pcdshub/pcds-ci-helpers/.github/workflows/twincat-standard.yml	79	1.98
actions:julia-actions/julia-buildpkg	78	1.96
actions:JuliaRegistries/TagBot	78	1.96

actions:actions/github-script	77	1.93
actions:actions/create-release	75	1.88
actions:peter-evans/create-pull-request	74	1.86
actions:ruby/setup-ruby	74	1.86
actions:actions-rs/toolchain	70	1.76
actions:r-lib/actions/check-r-package	70	1.76
actions:julia-actions/julia-processcoverage	65	1.63
actions:actions/configure-pages	61	1.53
actions:pre-commit/action	57	1.43
actions:mदाविसावर/ci-core-dumper	54	1.36
actions:ad-m/github-push-action	53	1.33
actions:actions/setup-dotnet	49	1.23
actions:actions/upload-release-asset	49	1.23
actions:actions-rs/cargo	44	1.10
actions:pypa/cibuildwheel	44	1.10
actions:github/codeql-action/upload-sarif	40	1.00
actions:actions/labeler	38	0.95
actions:golangci/golangci-lint-action	38	0.95

actions:styfle/cancel-workflow-action	38	0.95
actions:svenstaro/upload-release-action	38	0.95
actions:dtolnay/rust-toolchain	37	0.93
actions:ncipollo/release-action	37	0.93
actions:release-drafter/release-drafter	36	0.90
actions:coverallsapp/github-action	35	0.88
actions:aws-actions/configure-aws-credentials	34	0.85
actions:msys2/setup-msys2	34	0.85
actions:codespell-project/actions-codespell	30	0.75
actions:goreleaser/goreleaser-action	30	0.75
actions:ilammy/msvc-dev-cmd	30	0.75
actions:psf/black	29	0.73
actions:Swatinem/rust-cache	29	0.73
actions:actions/add-to-project	28	0.70
actions:gradle/gradle-build-action	28	0.70
actions:mamba-org/setup-micromamba	28	0.70
actions:stefanzweifel/git-auto-commit-action	28	0.70

actions:pcdshub/pcds-ci- helpers/.github/workflows/python- standard.yml	27	0.68
actions:microsoft/setup-msbuild	24	0.60
actions:tj-actions/changed-files	24	0.60
actions:peter-evans/dockerhub- description	23	0.58
actions:snok/install-poetry	23	0.58
actions:peter-evans/create-or- update-comment	22	0.55
actions:atlassian/gajira-create	21	0.53
actions:atlassian/gajira-login	21	0.53
actions:crate-ci/typos	21	0.53
actions:EndBug/add-and-commit	20	0.50
actions:dawidd6/action-download- artifact	19	0.48
actions:marvinpinto/action- automatic-releases	19	0.48
actions:r-lib/actions/pr-fetch	19	0.48
actions:r-lib/actions/pr-push	19	0.48
actions:shivammathur/setup-php	18	0.45
actions:sigstore/cosign-installer	18	0.45
actions:uraimo/run-on-arch-action	17	0.43
actions:actions/cache/restore	16	0.40
actions:cachix/install-nix-action	16	0.40

actions:cla-assistant/github-action	16	0.40
actions:gradle/wrapper-validation-action	16	0.40
actions:openjournals/openjournals-draft-action	16	0.40
actions:actions/cache/save	15	0.38
actions:actions/dependency-review-action	15	0.38
actions:actions/first-interaction	15	0.38
actions:cvmfs-contrib/github-action-cvmfs	15	0.38
actions:dorny/paths-filter	15	0.38
actions:gaurav-nelson/github-action-markdown-link-check	15	0.38
actions:github/super-linter	15	0.38
actions:jidicula/clang-format-action	15	0.38
actions:julia-actions/cache	15	0.38
actions:DoozyX/clang-format-lint-action	14	0.35
actions:lukka/get-cmake	14	0.35
actions:ossf/scorecard-action	14	0.35
actions:abatilo/actions-poetry	13	0.33
actions:amannn/action-semantic-pull-request	13	0.33

actions:ammaraskar/sphinx-action	13	0.33
actions:aws-actions/amazon-ecr-login	13	0.33
actions:google-github-actions/setup-gcloud	13	0.33
actions:hendrikmuhs/ccache-action	13	0.33
actions:jurplel/install-qt-action	13	0.33
actions:larsoner/circleci-artifacts-redirector-action	13	0.33
actions:matlab-actions/setup-matlab	13	0.33
actions:s-weigand/setup-conda	13	0.33
actions:taiki-e/install-action	13	0.33
actions:actions/jekyll-build-pages	12	0.30
actions:eic/run-cvmfs-osg-eic-shell	12	0.30
actions:google-github-actions/auth	12	0.30
actions:jwlawson/actions-setup-cmake	12	0.30
actions:mamba-org/provision-with-micromamba	12	0.30
actions:mattnotmitt/doxygen-action	12	0.30
actions:seanmiddleditch/gha-setup-ninja	12	0.30
actions:actions-rs/clippy-check	11	0.28

actions:aquasecurity/trivy-action	11	0.28
actions:azure/setup-helm	11	0.28
actions:fkirc/skip-duplicate-actions	11	0.28
actions:flux-framework/pr-validator	11	0.28
actions:google-github-actions/release-please-action	11	0.28
actions:helm/kind-action	11	0.28
actions:r-lib/actions/setup-tinytex	11	0.28
actions:8398a7/action-slack	10	0.25
actions:akhileshns/heroku-deploy	10	0.25
actions:appleboy/ssh-action	10	0.25
actions:crazy-max/ghaction-github-pages	10	0.25
actions:cypress-io/github-action	10	0.25
actions:EnricoMi/publish-unit-test-result-action	10	0.25
actions:google/oss-fuzz/infra/cifuzz/actions/build_fuzzers	10	0.25
actions:Jimver/cuda-toolkit	10	0.25
actions:marocchino/sticky-pull-request-comment	10	0.25
actions:matlab-actions/run-command	10	0.25

actions:mikepenz/action-junit-report	10	0.25
actions:PennyLaneAI/automation/version_bump_action	10	0.25
actions:pnpm/action-setup	10	0.25
composer:phpunit/phpunit	83	52.87
composer:php	61	38.85
composer:psr/log	57	36.31
composer:symfony/polyfill-mbstring	54	34.39
composer:symfony/console	51	32.48
composer:phpunit/php-code-coverage	47	29.94
composer:phpunit/php-file-iterator	47	29.94
composer:phpunit/php-text-template	47	29.94
composer:phpunit/php-timer	47	29.94
composer:sebastian/diff	47	29.94
composer:sebastian/exporter	47	29.94
composer:symfony/yaml	47	29.94
composer:sebastian/environment	46	29.30
composer:sebastian/version	46	29.30
composer:psr/http-message	45	28.66
composer:sebastian/comparator	45	28.66
composer:sebastian/global-state	44	28.03

composer:sebastian/recursion- context	44	28.03
composer:symfony/event- dispatcher	44	28.03
composer:symfony/process	44	28.03
composer:webmozart/assert	44	28.03
composer:symfony/polyfill-ctype	43	27.39
composer:psr/container	42	26.75
composer:myclabs/deep-copy	41	26.11
composer:symfony/finder	41	26.11
composer:nikic/php-parser	40	25.48
composer:sebastian/code-unit- reverse-lookup	40	25.48
composer:sebastian/object- enumerator	40	25.48
composer:monolog/monolog	39	24.84
composer:doctrine/instantiator	38	24.20
composer:doctrine/lexer	38	24.20
composer:symfony/css-selector	38	24.20
composer:guzzlehttp/psr7	37	23.57
composer:symfony/routing	37	23.57
composer:guzzlehttp/guzzle	36	22.93
composer:phar-io/manifest	36	22.93
composer:phar-io/version	36	22.93

composer:sebastian/object-reflector	36	22.93
composer:symfony/http-foundation	36	22.93
composer:theseer/tokenizer	36	22.93
composer:symfony/translation	35	22.29
composer:egulias/email-validator	34	21.66
composer:phpdocumentor/reflection-docblock	34	21.66
composer:sebastian/resource-operations	34	21.66
composer:symfony/deprecation-contracts	34	21.66
composer:symfony/http-kernel	34	21.66
composer:symfony/service-contracts	34	21.66
composer:symfony/var-dumper	34	21.66
composer:symfony/polyfill-php72	33	21.02
composer:guzzlehttp/promises	32	20.38
composer:phpdocumentor/reflection-common	32	20.38
composer:phpdocumentor/type-resolver	32	20.38
composer:symfony/polyfill-php80	32	20.38
composer:phpspec/prophecy	31	19.75

composer:symfony/polyfill-intl-normalizer	31	19.75
composer:doctrine/inflector	30	19.11
composer:ralouphie/getallheaders	30	19.11
composer:symfony/event-dispatcher-contracts	30	19.11
composer:symfony/polyfill-intl-idn	30	19.11
composer:league/flysystem	28	17.83
composer:symfony/error-handler	28	17.83
composer:symfony/translation-contracts	28	17.83
composer:phpunit/php-token-stream	27	17.20
composer:psr/http-client	27	17.20
composer:psr/simple-cache	27	17.20
composer:mockery/mockery	26	16.56
composer:nesbot/carbon	26	16.56
composer:psr/http-factory	26	16.56
composer:ramsey/uuid	26	16.56
composer:symfony/polyfill-intl-grapheme	26	16.56
composer:symfony/string	26	16.56
composer:symfony/mime	25	15.92
composer:vlucas/phpdotenv	25	15.92
composer:laravel/framework	24	15.29

composer:psy/psys	24	15.29
composer:sebastian/type	24	15.29
composer:symfony/filesystem	24	15.29
composer:twig/twig	24	15.29
composer:psr/event-dispatcher	23	14.65
composer:hamcrest/hamcrest-php	22	14.01
composer:psr/cache	22	14.01
composer:squizlabs/php_codesniffer	22	14.01
composer:phpunit/php-invoker	21	13.38
composer:sebastian/cli-parser	21	13.38
composer:sebastian/code-unit	21	13.38
composer:sebastian/complexity	21	13.38
composer:sebastian/lines-of-code	21	13.38
composer:doctrine/annotations	20	12.74
composer:dragonmantank/cron-expression	20	12.74
composer:brick/math	19	12.10
composer:filp/whoops	19	12.10
composer:laravel/tinker	19	12.10
composer:phpoption/phpoption	19	12.10
composer:ramsey/collection	19	12.10
composer:symfony/polyfill-php73	19	12.10
composer:league/mime-type-detection	18	11.46
composer:paragonie/random_compat	18	11.46

composer:tijssverkoyen/css-to-inline-styles	18	11.46
composer:composer/semver	17	10.83
composer:dflydev/dot-access-data	17	10.83
composer:nunomaduro/collision	17	10.83
composer:phpunit/phpunit-mock-objects	17	10.83
composer:symfony/debug	17	10.83
composer:doctrine/cache	16	10.19
composer:symfony/dependency-injection	16	10.19
composer:symfony/stopwatch	16	10.19
composer:symfony/validator	16	10.19
composer:doctrine/dbal	15	9.55
composer:league/commonmark	15	9.55
composer:symfony/config	15	9.55
composer:symfony/http-client-contracts	15	9.55
composer:symfony/polyfill-iconv	15	9.55
composer:voku/portable-ascii	15	9.55
composer:doctrine/collections	14	8.92
composer:ezyang/htmlpurifier	14	8.92
composer:fakerphp/faker	14	8.92
composer:graham-campbell/result-type	14	8.92

composer:swiftmailer/swiftmailer	14	8.92
composer:masterminds/html5	13	8.28
composer:symfony/mailer	13	8.28
composer:symfony/polyfill-php81	13	8.28
composer:doctrine/deprecations	12	7.64
composer:doctrine/event-manager	12	7.64
composer:maennchen/zipstream- php	12	7.64
composer:nette/utils	12	7.64
composer:symfony/dom-crawler	12	7.64
composer:symfony/var-exporter	12	7.64
composer:doctrine/common	11	7.01
composer:fzaninotto/faker	11	7.01
composer:laravel/serializable- closure	11	7.01
composer:mikey179/vfsStream	11	7.01
composer:spatie/backtrace	11	7.01
composer:symfony/cache	11	7.01
composer:symfony/serializer	11	7.01
composer:dnoegel/php-xdg-base- dir	10	6.37
composer:drush/drush	10	6.37
composer:justinrainbow/json- schema	10	6.37
composer:lcobucci/jwt	10	6.37

composer:league/config	10	6.37
composer:nette/schema	10	6.37
composer:opis/closure	10	6.37
composer:phpstan/phpdoc-parser	10	6.37
composer:symfony/cache-contracts	10	6.37
composer:symfony/options-resolver	10	6.37
composer:symfony/psr-http- message-bridge	10	6.37
go:golang.org/x/sys	195	65.22
go:golang.org/x/net	158	52.84
go:golang.org/x/text	149	49.83
go:gopkg.in/yaml.v3	133	44.48
go:golang.org/x/crypto	130	43.48
go:github.com/pkg/errors	126	42.14
go:github.com/stretchr/testify	124	41.47
go:github.com/golang/protobuf	122	40.80
go:google.golang.org/protobuf	119	39.80
go:github.com/davecgh/go-spew	115	38.46
go:github.com/spf13/pflag	114	38.13
go:github.com/google/uuid	112	37.46
go:gopkg.in/yaml.v2	107	35.79
go:github.com/google/go-cmp	95	31.77
go:github.com/matttn/go-isatty	91	30.43
go:golang.org/x/sync	91	30.43
go:golang.org/x/term	89	29.77

go:golang.org/x/time	88	29.43
go:github.com/pmezard/go-difflib	87	29.10
go:github.com/gogo/protobuf	86	28.76
go:google.golang.org/grpc	86	28.76
go:github.com/json-iterator/go	85	28.43
go:golang.org/x/oauth2	84	28.09
go:github.com/modern-go/reflect2	83	27.76
go:github.com/modern-go/concurrent	82	27.42
go:github.com/spf13/cobra	82	27.42
go:github.com/sirupsen/logrus	81	27.09
go:golang.org/x/tools	81	27.09
go:k8s.io/apimachinery	81	27.09
go:github.com/go-logr/logr	79	26.42
go:k8s.io/client-go	79	26.42
go:github.com/prometheus/client_golang	77	25.75
go:google.golang.org/appengine	77	25.75
go:github.com/fsnotify/fsnotify	75	25.08
go:github.com/cespare/xxhash/v2	74	24.75
go:k8s.io/api	74	24.75
go:github.com/inconshreveable/mousetrap	72	24.08
go:github.com/matttn/go-colorable	72	24.08
go:github.com/prometheus/client_model	71	23.75
go:github.com/prometheus/common	71	23.75
go:github.com/golang/groupcache	69	23.08

go:github.com/imdario/mergo	68	22.74
go:github.com/prometheus/procfs	68	22.74
go:github.com/beorn7/perks	67	22.41
go:k8s.io/klog/v2	67	22.41
go:golang.org/x/exp	66	22.07
go:google.golang.org/genproto	66	22.07
go:github.com/google/gofuzz	65	21.74
go:k8s.io/utils	65	21.74
go:github.com/mailru/easyjson	64	21.40
go:golang.org/x/mod	63	21.07
go:sigs.k8s.io/yaml	63	21.07
go:gopkg.in/inf.v0	62	20.74
go:github.com/josharian/intern	61	20.40
go:github.com/go-openapi/swag	60	20.07
go:sigs.k8s.io/structured-merge- diff/v4	59	19.73
go:github.com/go- openapi/jsonpointer	58	19.40
go:k8s.io/kube-openapi	58	19.40
go:github.com/go- openapi/jsonreference	57	19.06
go:github.com/mitchellh/mapstructure	56	18.73
go:github.com/klauspost/compress	54	18.06
go:github.com/fatih/color	53	17.73
go:github.com/matttproud/golang-protobuf_extensions	53	17.73

go:sigs.k8s.io/json	53	17.73
go:go.uber.org/multierr	52	17.39
go:github.com/munnerz/goautoneg	50	16.72
go:go.uber.org/zap	48	16.05
go:github.com/emicklei/go-restful/v3	46	15.38
go:github.com/matttn/go-runewidth	46	15.38
go:github.com/opencontainers/go-digest	46	15.38
go:github.com/spf13/viper	46	15.38
go:github.com/evanphx/json-patch	45	15.05
go:github.com/gorilla/websocket	44	14.72
go:github.com/spf13/afero	43	14.38
go:github.com/spf13/cast	43	14.38
go:google.golang.org/genproto/googleapis/rpc	43	14.38
go:github.com/gorilla/mux	42	14.05
go:github.com/Microsoft/go-winio	42	14.05
go:github.com/pelletier/go-toml/v2	41	13.71
go:gopkg.in/ini.v1	41	13.71
go:github.com/docker/docker	40	13.38
go:github.com/opencontainers/image-spec	40	13.38
go:github.com/rivo/uniseg	40	13.38

go:go.opentelemetry.io/otel	39	13.04
go:go.opentelemetry.io/otel/trace	39	13.04
go:github.com/hashicorp/hcl	38	12.71
go:github.com/magiconair/properties	38	12.71
go:go.uber.org/atomic	38	12.71
go:github.com/go-logr/stdr	37	12.37
go:github.com/onsi/gomega	37	12.37
go:github.com/russross/blackfriday/v2	37	12.37
go:github.com/docker/go-units	36	12.04
go:github.com/felixge/httpsnoop	36	12.04
go:go.opencensus.io	36	12.04
go:k8s.io/component-base	36	12.04
go:github.com/docker/distribution	35	11.71
go:gopkg.in/check.v1	35	11.71
go:github.com/hashicorp/errwrap	34	11.37
go:github.com/hashicorp/go-multierror	34	11.37
go:github.com/subosito/gotenv	34	11.37
go:go.opentelemetry.io/otel/metric	34	11.37
go:k8s.io/apiextensions-apiserver	34	11.37
go:github.com/golang/snappy	33	11.04
go:github.com/mitchellh/go-homedir	33	11.04
go:github.com/cpuguy83/go-md2man/v2	32	10.70

go:github.com/cenkalti/backoff/v4	31	10.37
go:github.com/docker/go-connections	31	10.37
go:google.golang.org/genproto/googleapis/api	31	10.37
go:github.com/aws/aws-sdk-go	30	10.03
go:github.com/Azure/go-ansiterm	30	10.03
go:sigs.k8s.io/controller-runtime	30	10.03
go:github.com/asaskevich/govalidator	29	9.70
go:go.opentelemetry.io/contrib/instrumentation/net/http/otelhttp	29	9.70
go:golang.org/x/xerrors	29	9.70
go:cloud.google.com/go	28	9.36
go:github.com/BurntSushi/toml	28	9.36
go:github.com/google/gnostic-models	28	9.36
go:github.com/moby/term	28	9.36
go:github.com/pelletier/go-toml	28	9.36
go:google.golang.org/api	28	9.36
go:github.com/coreos/go-systemd/v22	27	9.03
go:github.com/google/gnostic	27	9.03
go:github.com/jmespath/go-jmespath	27	9.03
go:github.com/kr/pretty	26	8.70
go:github.com/kr/text	26	8.70
go:github.com/olekukonko/tablewriter	26	8.70

go:github.com/go-sql-driver/mysql	25	8.36
go:github.com/hashicorp/golang-lru	25	8.36
go:github.com/klauspost/cpuid/v2	25	8.36
go:github.com/moby/spdystream	25	8.36
go:github.com/rogppe/go-internal	25	8.36
go:github.com/ulikunitz/xz	25	8.36
go:github.com/cloudflare/circl	24	8.03
go:github.com/cyphar/filepath-securejoin	24	8.03
go:github.com/dustin/go-humanize	24	8.03
go:github.com/go-playground/locales	24	8.03
go:github.com/go-playground/universal-translator	24	8.03
go:github.com/google/pprof	24	8.03
go:github.com/onsi/ginkgo	24	8.03
go:github.com/spf13/jwalterweatherman	24	8.03
go:github.com/containerd/containerd	23	7.69
go:github.com/grpc-ecosystem/grpc-gateway/v2	23	7.69
go:go.opentelemetry.io/otel/sdk	23	7.69
go:gopkg.in/natefinch/lumberjack.v2	23	7.69

go:github.com/go-playground/validator/v10	22	7.36
go:github.com/golang-jwt/jwt/v4	22	7.36
go:github.com/googleapis/gax-go/v2	22	7.36
go:github.com/leodido/go-urn	22	7.36
go:github.com/mitchellh/wordwrap	22	7.36
go:go.opentelemetry.io/contrib/instrumentation/google.golang.org/grpc/otelgrpc	22	7.36
go:go.opentelemetry.io/proto/otlp	22	7.36
go:cloud.google.com/go/compute	21	7.02
go:cloud.google.com/go/compute/metadata	21	7.02
go:github.com/docker/docker-credential-helpers	21	7.02
go:github.com/gin-gonic/gin	21	7.02
go:github.com/lib/pq	21	7.02
go:github.com/Masterminds/semver/v2	21	7.02
go:github.com/onsi/ginkgo/v2	21	7.02
go:github.com/xeipuuv/gojsonpointer	21	7.02
go:go.opentelemetry.io/otel/exporters/otlp/otlptrace	21	7.02
go:k8s.io/apiserver	21	7.02
go:github.com/gabriel-vasile/mimetype	20	6.69
go:github.com/go-errors/errors	20	6.69
go:github.com/google/btree	20	6.69

go:github.com/miekg/dns	20	6.69
go:github.com/ProtonMail/go-crypto	20	6.69
go:go.etcd.io/bbolt	20	6.69
go:go.opentelemetry.io/otel/exporters/otlp/otlptrace/otlptracegrpc	20	6.69
go:github.com/aws/aws-sdk-go-v2	19	6.35
go:github.com/aws/aws-sdk-go-v2/credentials	19	6.35
go:github.com/blang/semver/v4	19	6.35
go:github.com/coreos/go-semver	19	6.35
go:github.com/docker/cli	19	6.35
go:github.com/evanphx/json-patch/v5	19	6.35
go:github.com/go-task/slim-sprig	19	6.35
go:github.com/google/shlex	19	6.35
go:github.com/grpc-ecosystem/go-grpc-prometheus	19	6.35
go:github.com/hashicorp/go-cleanhttp	19	6.35
go:github.com/sergi/go-diff	19	6.35
go:github.com/ugorji/go/codec	19	6.35
go:gopkg.in/tomb.v1	19	6.35
go:k8s.io/kubectl	19	6.35
go:github.com/aws/aws-sdk-go-v2/config	18	6.02

go:github.com/aws/aws-sdk-go-v2/internal/configsources	18	6.02
go:github.com/aws/aws-sdk-go-v2/internal/endpoints/v2	18	6.02
go:github.com/aws/smithy-go	18	6.02
go:github.com/goccy/go-json	18	6.02
go:github.com/googleapis/enterprise-certificate-proxy	18	6.02
go:github.com/shopspring/decimal	18	6.02
go:github.com/stretchr/objx	18	6.02
go:github.com/xeipuuv/gojsonreference	18	6.02
go:github.com/xlab/treeprint	18	6.02
go:gomodules.xyz/jsonpatch/v2	18	6.02
go:gopkg.in/square/go-jose.v2	18	6.02
go:gopkg.in/warnings.v0	18	6.02
go:k8s.io/cli-runtime	18	6.02
go:sigs.k8s.io/kustomize/kyaml	18	6.02
go:github.com/aws/aws-sdk-go-v2/feature/ec2/imds	17	5.69
go:github.com/aws/aws-sdk-go-v2/internal/ini	17	5.69
go:github.com/aws/aws-sdk-go-v2/service/internal/presigned-url	17	5.69
go:github.com/aws/aws-sdk-go-v2/service/sso	17	5.69

go:github.com/aws/aws-sdk-go-v2/service/sts	17	5.69
go:github.com/Azure/go-autorest	17	5.69
go:github.com/Azure/go-autorest/autorest	17	5.69
go:github.com/googleapis/gnostic	17	5.69
go:github.com/kevinburke/ssh_config	17	5.69
go:github.com/mitchellh/copystructure	17	5.69
go:github.com/mitchellh/reflectwalk	17	5.69
go:github.com/monochromegane/go-gitignore	17	5.69
go:github.com/opencontainers/runc	17	5.69
go:github.com/xeipuuv/gojsonschema	17	5.69
go:go.etcd.io/etcd/api/v3	17	5.69
go:go.starlark.net	17	5.69
go:sigs.k8s.io/kustomize/api	17	5.69
go:dario.cat/mergo	16	5.35
go:github.com/Azure/go-autorest/autorest/adal	16	5.35
go:github.com/gin-contrib/sse	16	5.35
go:github.com/go-git/go-billy/v5	16	5.35
go:github.com/go-git/go-git/v5	16	5.35
go:github.com/golang/mock	16	5.35
go:github.com/gregjones/httpcache	16	5.35

go:github.com/hashicorp/go-version	16	5.35
go:github.com/lucasb-eyer/go-colorful	16	5.35
go:github.com/moby/sys/mountinfo	16	5.35
go:github.com/morikuni/aec	16	5.35
go:github.com/opencontainers/runtime-spec	16	5.35
go:github.com/peterbourgon/diskv	16	5.35
go:github.com/tidwall/pretty	16	5.35
go:github.com/xanzy/ssh-agent	16	5.35
go:go.etcd.io/etcd/client/v3	16	5.35
go:github.com/alecthomas/units	15	5.02
go:github.com/aws/aws-sdk-go-v2/service/ssooidc	15	5.02
go:github.com/emirpasic/gods	15	5.02
go:github.com/ghodss/yaml	15	5.02
go:github.com/go-git/gcfg	15	5.02
go:github.com/go-logfmt/logfmt	15	5.02
go:github.com/go-ole/go-ole	15	5.02
go:github.com/golang-jwt/jwt	15	5.02
go:github.com/golang/glog	15	5.02
go:github.com/google/go-containerregistry	15	5.02
go:github.com/huandu/xstrings	15	5.02

go:github.com/jbenet/go-context	15	5.02
go:github.com/liggitt/tabwriter	15	5.02
go:github.com/Microsoft/hcsshim	15	5.02
go:github.com/mxk/go-flowrate	15	5.02
go:github.com/urfave/cli/v2	15	5.02
go:github.com/vbatts/tar-split	15	5.02
go:go.etcd.io/etcd/client/pkg/v3	15	5.02
go:gonum.org/v1/gonum	15	5.02
go:github.com/Azure/go- autoREST/autoREST/date	14	4.68
go:github.com/Azure/go- autoREST/logger	14	4.68
go:github.com/Azure/go- autoREST/tracing	14	4.68
go:github.com/containerd/cgroups	14	4.68
go:github.com/dgryski/go- rendezvous	14	4.68
go:github.com/go-openapi/spec	14	4.68
go:github.com/gofrs/uuid	14	4.68
go:github.com/google/go- querystring	14	4.68
go:github.com/hashicorp/go- retryablehttp	14	4.68
go:github.com/klauspost/pgzip	14	4.68
go:github.com/MakeNowJust/heredoc	14	4.68

go:github.com/Masterminds/goutils	14	4.68
go:github.com/pierrec/lz4	14	4.68
go:github.com/syndtr/goleveldb	14	4.68
go:golang.org/x/arch	14	4.68
go:k8s.io/klog	14	4.68
go:cloud.google.com/go/iam	13	4.35
go:github.com/distribution/reference	13	4.35
go:github.com/exponent- io/jsonpath	13	4.35
go:github.com/go-logr/zapr	13	4.35
go:github.com/gobwas/glob	13	4.35
go:github.com/google/cel-go	13	4.35
go:github.com/kylelemons/godebug	13	4.35
go:github.com/Masterminds/sprig/v3	13	4.35
go:github.com/mattn/go-sqlite3	13	4.35
go:github.com/nxadm/tail	13	4.35
go:github.com/opencontainers/selinux	13	4.35
go:github.com/pjbgf/sha1cd	13	4.35
go:github.com/sagikazarmark/locafero	13	4.35
go:github.com/sagikazarmark/slog- shim	13	4.35
go:github.com/skeema/knownhosts	13	4.35
go:github.com/sourcegraph/conc	13	4.35
go:github.com/stoewer/go-strcase	13	4.35
go:github.com/tidwall/gjson	13	4.35

go:github.com/tidwall/match	13	4.35
go:github.com/VividCortex/ewma	13	4.35
go:golang.org/x/image	13	4.35
go:gorm.io/gorm	13	4.35
go:gotest.tools/v3	13	4.35
go:github.com/alecthomas/template	12	4.01
go:github.com/andybalholm/brotli	12	4.01
go:github.com/bytedance/sonic	12	4.01
go:github.com/cenkalti/backoff	12	4.01
go:github.com/chai2010/gettext-go	12	4.01
go:github.com/dlclark/regexp2	12	4.01
go:github.com/docker/go-metrics	12	4.01
go:github.com/dsnet/compress	12	4.01
go:github.com/go-redis/redis/v8	12	4.01
go:github.com/godbus/dbus/v5	12	4.01
go:github.com/gofrs/flock	12	4.01
go:github.com/google/s2a-go	12	4.01
go:github.com/moby/locker	12	4.01
go:github.com/muesli/termenv	12	4.01
go:github.com/niemeyer/pretty	12	4.01
go:github.com/NYTimes/gziphandler	12	4.01
go:github.com/pkg/browser	12	4.01
go:github.com/robfig/cron/v3	12	4.01
go:github.com/satori/go.uuid	12	4.01
go:github.com/urfave/cli	12	4.01

go:github.com/vishvananda/netlink	12	4.01
go:github.com/vishvananda/netns	12	4.01
go:github.com/xrash/smetrics	12	4.01
go:cloud.google.com/go/storage	11	3.68
go:github.com/aws/aws-sdk- go-v2/service/internal/accept- encoding	11	3.68
go:github.com/chenzhuoyu/base64x	11	3.68
go:github.com/containerd/continuity	11	3.68
go:github.com/containerd/stargz- snapshotter/estargz	11	3.68
go:github.com/go-kit/log	11	3.68
go:github.com/gorilla/handlers	11	3.68
go:github.com/jmoiron/sqlx	11	3.68
go:github.com/Masterminds/semver	11	3.68
go:github.com/matttn/go- shellwords	11	3.68
go:github.com/mgutz/ansi	11	3.68
go:github.com/miekg/pkcs11	11	3.68
go:github.com/muesli/reflow	11	3.68
go:github.com/opentracing/opentracing- go	11	3.68
go:github.com/pierrec/lz4/v4	11	3.68
go:github.com/PuerkitoBio/purell	11	3.68
go:github.com/PuerkitoBio/urlesc	11	3.68

go:github.com/samber/lo	11	3.68
go:github.com/secure-systems-lab/go-securesystemslib	11	3.68
go:github.com/twitchyliquid64/golang-asm	11	3.68
go:go.mongodb.org/mongo-driver	11	3.68
go:sigs.k8s.io/apiserver-network-proxy/konnectivity-client	11	3.68
go:github.com/antlr/antlr4/runtime/Go/antlr/v4	10	3.34
go:github.com/aws/aws-sdk-go-v2/internal/v4a	10	3.34
go:github.com/aws/aws-sdk-go-v2/service/s3	10	3.34
go:github.com/aymanbagabas/go-osc52/v2	10	3.34
go:github.com/golang/freetype	10	3.34
go:github.com/grpc-ecosystem/go-grpc-middleware	10	3.34
go:github.com/hashicorp/go-uuid	10	3.34
go:github.com/julienschmidt/httprouter	10	3.34
go:github.com/rcrowley/go-metrics	10	3.34
go:github.com/titanous/rocacheck	10	3.34
go:github.com/valyala/bytebufferpool	10	3.34
go:github.com/zclconf/go-cty	10	3.34
go:gotest.tools	10	3.34

go:k8s.io/kms	10	3.34
maven:junit:junit	404	48.21
maven:org.apache.maven.plugins:maven-compiler-plugin	342	40.81
maven:org.apache.maven.plugins:maven-surefire-plugin	190	22.67
maven:org.apache.maven.plugins:maven-javadoc-plugin	149	17.78
maven:org.slf4j:slf4j-api	134	15.99
maven:org.apache.maven.plugins:maven-jar-plugin	131	15.63
maven:org.apache.maven.plugins:maven-source-plugin	125	14.92
maven:org.apache.maven.plugins:maven-shade-plugin	115	13.72
maven:com.google.guava:guava	108	12.89
maven:org.apache.commons:commons-lang3	108	12.89
maven:commons-io:commons-io	97	11.58
maven:org.apache.maven.plugins:maven-gpg-plugin	97	11.58
maven:com.google.code.gson:gson	88	10.50
maven:org.codehaus.mojo:exec-maven-plugin	77	9.19

maven:com.fasterxml.jackson.core:jackson-databind	75	8.95
maven:org.apache.maven.plugins:maven-dependency-plugin	71	8.47
maven:org.jacoco:jacoco-maven-plugin	70	8.35
maven:org.apache.httpcomponents:httpclient	69	8.23
maven:ch.qos.logback:logback-classic	64	7.64
maven:org.apache.maven.plugins:maven-release-plugin	64	7.64
maven:log4j:log4j	63	7.52
maven:mysql:mysql-connector-java	63	7.52
maven:org.slf4j:slf4j-log4j12	61	7.28
maven:org.sonatype.plugins:nexus-staging-maven-plugin	61	7.28
maven:org.apache.maven.plugins:maven-checkstyle-plugin	59	7.04
maven:org.apache.maven.plugins:maven-assembly-plugin	58	6.92
maven:org.apache.maven.plugins:maven-resources-plugin	58	6.92
maven:org.codehaus.mojo:build-helper-maven-plugin	56	6.68
maven:org.mockito:mockito-core	54	6.44

maven:org.apache.commons:commons-math3	53	6.32
maven:org.json:json	52	6.21
maven:commons-cli:commons-cli	51	6.09
maven:commons-lang:commons-lang	51	6.09
maven:commons-codec:commons-codec	48	5.73
maven:org.apache.logging.log4j:log4j-core	48	5.73
maven:com.fasterxml.jackson.core:jackson-core	42	5.01
maven:org.apache.maven.plugins:maven-war-plugin	41	4.89
maven:org.junit.jupiter:junit-jupiter-engine	41	4.89
maven:org.slf4j:slf4j-simple	41	4.89
maven:javax.servlet:javax.servlet-api	40	4.77
maven:org.apache.maven.plugins:maven-site-plugin	40	4.77
maven:edu.stanford.nlp:stanford-corenlp	39	4.65
maven:org.apache.logging.log4j:log4j-api	39	4.65

maven:javax.xml.bind:jaxb-api	36	4.30
maven:org.eclipse.m2e:lifecycle- mapping	36	4.30
maven:org.projectlombok:lombok	36	4.30
maven:commons-logging:commons- logging	35	4.18
maven:org.apache.maven.plugins:maven- deploy-plugin	35	4.18
maven:com.google.protobuf:protobuf- java	33	3.94
maven:org.apache.maven.plugins:maven- enforcer-plugin	33	3.94
maven:org.jsoup:jsoup	33	3.94
maven:joda-time:joda-time	32	3.82
maven:org.apache.felix:maven- bundle-plugin	32	3.82
maven:org.scala-lang:scala-library	32	3.82
maven:org.apache.lucene:lucene- core	31	3.70
maven:org.junit.jupiter:junit- jupiter-api	31	3.70
maven:javax.servlet:servlet-api	30	3.58
maven:org.apache.maven.plugins:maven- failsafe-plugin	29	3.46

maven:com.fasterxml.jackson.core:jackson- annotations	28	3.34
maven:commons- collections:commons-collections	28	3.34
maven:org.apache.maven.plugins:maven- install-plugin	28	3.34
maven:org.apache.maven.plugins:maven- project-info-reports-plugin	28	3.34
maven:org.apache.commons:commons- compress	27	3.22
maven:org.apache.hadoop:hadoop- common	27	3.22
maven:com.googlecode.json- simple:json-simple	26	3.10
maven:org.apache.commons:commons- csv	26	3.10
maven:org.assertj:assertj-core	26	3.10
maven:org.springframework:spring- context	26	3.10
maven:org.apache.maven.plugins:maven- jxr-plugin	25	2.98
maven:com.alibaba:fastjson	24	2.86
maven:net.alchim31.maven:scala- maven-plugin	24	2.86
maven:org.mockito:mockito-all	24	2.86

maven:com.puppcrawl.tools:checkstyle	22	2.63
maven:org.apache.httpcomponents:httpcore	22	2.63
maven:org.apache.lucene:lucene-analyzers-common	22	2.63
maven:org.apache.maven.plugins:maven-surefire-report-plugin	22	2.63
maven:org.slf4j:jcl-over-slf4j	22	2.63
maven:org.springframework:spring-core	22	2.63
maven:org.testng:testng	22	2.63
maven:org.postgresql:postgresql	21	2.51
maven:com.opencsv:opencsv	20	2.39
maven:commons-configuration:commons-configuration	20	2.39
maven:org.apache.commons:commons-collections4	20	2.39
maven:org.apache.commons:commons-text	20	2.39
maven:org.apache.hadoop:hadoop-core	20	2.39
maven:org.apache.maven.plugins:maven-antrun-plugin	20	2.39
maven:org.apache.maven.plugins:maven-clean-plugin	20	2.39

maven:org.codehaus.mojo:findbugs-maven-plugin	20	2.39
maven:org.junit.jupiter:junit-jupiter	20	2.39
maven:org.mongodb:mongo-java-driver	20	2.39
maven:org.springframework:spring-jdbc	20	2.39
maven:org.springframework:spring-test	20	2.39
maven:org.springframework:spring-webmvc	20	2.39
maven:org.yaml:snakeyaml	20	2.39
maven:com.h2database:h2	19	2.27
maven:javax.annotation:javax.annotation-api	19	2.27
maven:net.java.dev.jna:jna	19	2.27
maven:org.codehaus.mojo:cobertura-maven-plugin	19	2.27
maven:org.apache.lucene:lucene-queryparser	18	2.15
maven:org.hamcrest:hamcrest-core	18	2.15
maven:org.junit.jupiter:junit-jupiter-params	18	2.15

maven:org.springframework.boot:spring- boot-maven-plugin	18	2.15
maven:com.google.code.findbugs:jsr305	17	2.03
maven:commons- fileupload:commons-fileupload	17	2.03
maven:org.apache.maven.plugins:maven- pmd-plugin	17	2.03
maven:org.apache.poi:poi-ooxml	17	2.03
maven:org.apache.zookeeper:zookeeper	17	2.03
maven:org.bouncycastle:bcprov- jdk15on	17	2.03
maven:org.glassfish.jaxb:jaxb- runtime	17	2.03
maven:org.ow2.asm:asm	17	2.03
maven:org.seleniumhq.selenium:selenium- java	17	2.03
maven:org.springframework:spring- beans	17	2.03
maven:org.springframework:spring- web	17	2.03
maven:ch.qos.logback:logback-core	16	1.91
maven:com.sun.xml.bind:jaxb-impl	16	1.91
maven:javax.validation:validation- api	16	1.91
maven:org.antlr:antlr4-runtime	16	1.91

maven:org.apache.maven.plugins:maven- eclipse-plugin	16	1.91
maven:org.apache.poi:poi	16	1.91
maven:org.codehaus.jackson:jackson- mapper-asl	16	1.91
maven:org.eluder.coveralls:coveralls- maven-plugin	16	1.91
maven:org.javassist:javassist	16	1.91
maven:commons- beanutils:commons-beanutils	15	1.79
maven:jakarta.xml.bind:jakarta.xml.bind- api	15	1.79
maven:org.apache.avro:avro	15	1.79
maven:org.apache.logging.log4j:log4j- slf4j-impl	15	1.79
maven:org.apache.maven.plugins:maven- plugin-plugin	15	1.79
maven:org.apache.maven:maven- plugin-api	15	1.79
maven:org.apache.spark:spark- core_2.10	15	1.79
maven:org.hibernate:hibernate- core	15	1.79
maven:org.apache.hadoop:hadoop- client	14	1.67

maven:org.apache.hadoop:hadoop- mapreduce-client-core	14	1.67
maven:org.eclipse.jetty:jetty-server	14	1.67
maven:org.hamcrest:hamcrest-all	14	1.67
maven:org.mybatis:mybatis	14	1.67
maven:org.slf4j:jul-to-slf4j	14	1.67
maven:redis.clients:jedis	14	1.67
maven:com.beust:jcommander	13	1.55
maven:com.sparkjava:spark-core	13	1.55
maven:commons- httpclient:commons-httpclient	13	1.55
maven:javax.servlet:jstl	13	1.55
maven:net.sf.trove4j:trove4j	13	1.55
maven:org.apache.maven.plugin- tools:maven-plugin-annotations	13	1.55
maven:org.easymock:easymock	13	1.55
maven:org.eclipse.tycho:target- platform-configuration	13	1.55
maven:org.eclipse.tycho:tycho- maven-plugin	13	1.55
maven:org.hamcrest:hamcrest- library	13	1.55
maven:org.jdom:jdom	13	1.55
maven:org.jlab.coat:coat-libs	13	1.55
maven:org.pitest:pitest-maven	13	1.55

maven:org.reflections:reflections	13	1.55
maven:org.springframework.boot:spring- boot-dependencies	13	1.55
maven:org.xolstice.maven.plugins:protobuf- maven-plugin	13	1.55
maven:com.github.eirslett:frontend- maven-plugin	12	1.43
maven:com.github.spotbugs:spotbugs- maven-plugin	12	1.43
maven:com.sun.jersey:jersey-server	12	1.43
maven:commons-dbcp:commons- dbcp	12	1.43
maven:io.netty:netty-all	12	1.43
maven:javax.ws.rs:javax.ws.rs-api	12	1.43
maven:net.sf.jopt-simple:jopt- simple	12	1.43
maven:org.codehaus.mojo:versions- maven-plugin	12	1.43
maven:org.eclipse.jgit:org.eclipse.jgit	12	1.43
maven:org.elasticsearch:elasticsearch	12	1.43
maven:org.hamcrest:hamcrest	12	1.43
maven:org.pitest:pitest-junit5- plugin	12	1.43
maven:org.springframework:spring- tx	12	1.43

maven:org.xerial:sqlite-jdbc	12	1.43
maven:args4j:args4j	11	1.31
maven:cglib:cglib	11	1.31
maven:com.squareup.okhttp3:okhttp	11	1.31
maven:com.sun.xml.bind:jaxb-core	11	1.31
maven:io.grpc:grpc-protobuf	11	1.31
maven:io.grpc:grpc-stub	11	1.31
maven:org.apache.commons:commons41	11	1.31
math		
maven:org.apache.opennlp:opennlp-	11	1.31
tools		
maven:org.apache.tomcat.maven:tomcat7-	11	1.31
maven-plugin		
maven:org.codehaus.mojo:license-	11	1.31
maven-plugin		
maven:org.glassfish:javax.json	11	1.31
maven:org.hibernate:hibernate-	11	1.31
validator		
maven:org.mockito:mockito-junit-	11	1.31
jupiter		
maven:org.mybatis:mybatis-spring	11	1.31
maven:org.slf4j:slf4j-jdk14	11	1.31
maven:org.springframework.cloud:spring-	11	1.31
cloud-dependencies		

maven:org.springframework:spring-aop	11	1.31
maven:org.springframework:spring-context-support	11	1.31
maven:com.mycila:license-maven-plugin	10	1.19
maven:dom4j:dom4j	10	1.19
maven:javax.mail:mail	10	1.19
maven:net.bytebuddy:byte-buddy	10	1.19
maven:org.antlr:antlr4-maven-plugin	10	1.19
maven:org.apache.commons:commons-dbcp2	10	1.19
maven:org.apache.hadoop:hadoop-hdfs	10	1.19
maven:org.awaitility:awaitility	10	1.19
maven:org.bouncycastle:bcpkix-jdk15on	10	1.19
maven:org.codehaus.jackson:jackson-core-asl	10	1.19
maven:org.eclipse.jetty:jetty-maven-plugin	10	1.19
maven:org.glassfish.jersey.core:jersey-client	10	1.19
maven:org.hsqldb:hsqldb	10	1.19

maven:org.jetbrains.kotlin:kotlin-maven-plugin	10	1.19
maven:org.openjdk.jmh:jmh-core	10	1.19
maven:org.openjdk.jmh:jmh-generator-annprocess	10	1.19
maven:org.slf4j:log4j-over-slf4j	10	1.19
maven:org.springframework.boot:spring-boot-starter-test	10	1.19
maven:org.springframework.boot:spring-boot-starter-web	10	1.19
npm:debug	1319	59.87
npm:inherits	1306	59.28
npm:ms	1291	58.60
npm:minimatch	1240	56.29
npm:brace-expansion	1235	56.06
npm:balanced-match	1234	56.01
npm:semver	1230	55.83
npm:concat-map	1228	55.74
npm:glob	1221	55.42
npm:once	1214	55.11
npm:safe-buffer	1213	55.06
npm:wrappy	1213	55.06
npm:path-is-absolute	1201	54.52
npm:supports-color	1196	54.29
npm:ansi-styles	1190	54.02

npm:chalk	1190	54.02
npm:inflight	1189	53.97
npm:fs.realpath	1185	53.79
npm:lodash	1183	53.70
npm:ansi-regex	1176	53.38
npm:strip-ansi	1171	53.15
npm:has-flag	1165	52.88
npm:color-name	1157	52.52
npm:escape-string-regexp	1157	52.52
npm:color-convert	1155	52.43
npm:rimraf	1154	52.38
npm:minimist	1153	52.34
npm:graceful-fs	1125	51.07
npm:punycode	1115	50.61
npm:util-deprecate	1113	50.52
npm:which	1110	50.39
npm:isexe	1108	50.30
npm:mime-db	1103	50.07
npm:is-extglob	1101	49.98
npm:is-glob	1101	49.98
npm:mime-types	1101	49.98
npm:fill-range	1097	49.80
npm:is-number	1097	49.80
npm:braces	1096	49.75
npm:isarray	1094	49.66

npm:source-map	1094	49.66
npm:is-fullwidth-code-point	1091	49.52
npm:string-width	1091	49.52
npm:function-bind	1086	49.30
npm:glob-parent	1085	49.25
npm:resolve	1080	49.02
npm:lru-cache	1079	48.98
npm:iconv-lite	1078	48.93
npm:yallist	1077	48.89
npm:string_decoder	1073	48.71
npm:readable-stream	1070	48.57
npm:path-exists	1067	48.43
npm:find-up	1065	48.34
npm:path-parse	1063	48.25
npm:ajv	1057	47.98
npm:object-assign	1054	47.84
npm:cross-spawn	1052	47.75
npm:normalize-path	1051	47.71
npm:fsevents	1048	47.57
npm:signal-exit	1046	47.48
npm:acorn	1037	47.07
npm:core-util-is	1036	47.03
npm:mkdirp	1035	46.98
npm:commander	1028	46.66
npm:anymatch	1026	46.57

npm:qs	1026	46.57
npm:fast-deep-equal	1023	46.44
npm:shebang-regex	1021	46.35
npm:to-regex-range	1021	46.35
npm:safer-buffer	1020	46.30
npm:shebang-command	1019	46.26
npm:json-schema-traverse	1018	46.21
npm:p-limit	1017	46.16
npm:argparse	1012	45.94
npm:js-yaml	1011	45.89
npm:path-key	1006	45.67
npm:camelcase	1005	45.62
npm:esutils	1004	45.57
npm:fast-json-stable-stringify	1004	45.57
npm:micromatch	1004	45.57
npm:locate-path	1003	45.53
npm:p-locate	1003	45.53
npm:js-tokens	1002	45.48
npm:yargs	1001	45.44
npm:resolve-from	999	45.35
npm:@types/node	998	45.30
npm:wrap-ansi	986	44.76
npm:cliui	982	44.58
npm:process-nextick-args	970	44.03
npm:yargs-parser	966	43.85

npm:chokidar	963	43.71
npm:estraverse	963	43.71
npm:express	961	43.62
npm:binary-extensions	960	43.58
npm:uuid	959	43.53
npm:readdirp	958	43.49
npm:globals	957	43.44
npm:has	956	43.40
npm:is-binary-path	956	43.40
npm:imurmurhash	954	43.30
npm:mime	953	43.26
npm:y18n	953	43.26
npm:get-caller-file	945	42.90
npm:require-directory	945	42.90
npm:strip-json-comments	940	42.67
npm:is-arrayish	936	42.49
npm:esprima	933	42.35
npm:path-type	929	42.17
npm:json5	927	42.08
npm:is-stream	922	41.85
npm:uri-js	919	41.72
npm:error-ex	911	41.35
npm:parse-json	910	41.31
npm:sprintf-js	905	41.08
npm:get-stream	901	40.90

npm:pify	900	40.85
npm:emoji-regex	894	40.58
npm:convert-source-map	892	40.49
npm:source-map-support	891	40.44
npm:kind-of	890	40.40
npm:buffer-from	886	40.22
npm:depd	884	40.13
npm:callsites	882	40.04
npm:esrecurse	880	39.95
npm:has-symbols	877	39.81
npm:to-fast-properties	877	39.81
npm:p-try	876	39.76
npm:negotiator	873	39.63
npm:statuses	871	39.54
npm:slash	869	39.45
npm:form-data	866	39.31
npm:http-errors	866	39.31
npm:combined-stream	865	39.26
npm:delayed-stream	865	39.26
npm:asynckit	864	39.22
npm:ignore	861	39.08
npm:setprototypeof	861	39.08
npm:bytes	860	39.04
npm:eslint	860	39.04
npm:body-parser	859	38.99

npm:escape-html	858	38.95
npm:unpipe	855	38.81
npm:ee-first	854	38.77
npm:on-finished	854	38.77
npm:regenerator-runtime	853	38.72
npm:encodeurl	852	38.67
npm:object-keys	852	38.67
npm:parseurl	852	38.67
npm:jsesc	849	38.54
npm:cookie	848	38.49
npm:path-to-regexp	847	38.45
npm:browserslist	846	38.40
npm:mimic-fn	846	38.40
npm:accepts	845	38.36
npm:range-parser	844	38.31
npm:make-dir	842	38.22
npm:destroy	841	38.18
npm:electron-to-chromium	841	38.18
npm:tslib	838	38.04
npm:async	837	37.99
npm:fresh	836	37.95
npm:define-properties	834	37.86
npm:etag	834	37.86
npm:finalhandler	833	37.81
npm:utils-merge	832	37.77

npm:execa	831	37.72
npm:npm-run-path	831	37.72
npm:serve-static	831	37.72
npm:ws	830	37.68
npm:send	827	37.54
npm:strip-bom	827	37.54
npm:@babel/code-frame	826	37.49
npm:fast-levenshtein	826	37.49
npm:raw-body	826	37.49
npm:caniuse-lite	822	37.31
npm:pkg-dir	822	37.31
npm:@babel/highlight	821	37.27
npm:picomatch	820	37.22
npm:vary	820	37.22
npm:deep-is	819	37.18
npm:onetime	815	37.00
npm:optionator	814	36.95
npm:prelude-ls	814	36.95
npm:type-check	814	36.95
npm:levn	813	36.90
npm:content-type	812	36.86
npm:media-typer	812	36.86
npm:type-is	811	36.81
npm:xtend	808	36.68
npm:eslint-scope	806	36.59

npm:ini	805	36.54
npm:globby	803	36.45
npm:postcss	797	36.18
npm:ipaddr.js	795	36.09
npm:content-disposition	794	36.04
npm:methods	793	36.00
npm:array-union	784	35.59
npm:proxy-addr	780	35.41
npm:forwarded	779	35.36
npm:merge-descriptors	779	35.36
npm:array-flatten	777	35.27
npm:cookie-signature	775	35.18
npm:is-typedarray	773	35.09
npm:is-regex	771	35.00
npm:decamelize	770	34.95
npm:set-blocking	768	34.86
npm:base64-js	765	34.73
npm:natural-compare	761	34.54
npm:text-table	761	34.54
npm:tough-cookie	760	34.50
npm:fs-extra	754	34.23
npm:is-callable	750	34.04
npm:extend	749	34.00
npm:is-date-object	746	33.86
npm:ieee754	745	33.82

npm:universalify	745	33.82
npm:object-inspect	743	33.73
npm:isobject	742	33.68
npm:ansi-escapes	741	33.64
npm:import-fresh	741	33.64
npm:is-path-inside	741	33.64
npm:events	740	33.59
npm:@babel/parser	738	33.50
npm:webpack	738	33.50
npm:@babel/types	736	33.41
npm:jsonfile	736	33.41
npm:randombytes	735	33.36
npm:buffer	732	33.23
npm:type-fest	729	33.09
npm:is-symbol	728	33.05
npm:node-releases	726	32.96
npm:@babel/core	724	32.86
npm:acorn-jsx	722	32.77
npm:core-js	722	32.77
npm:is-buffer	722	32.77
npm:object.assign	721	32.73
npm:loose-envify	718	32.59
npm:doctrine	717	32.55
npm:espre	716	32.50
npm:flat-cache	714	32.41

npm:is-plain-object	714	32.41
npm:end-of-stream	713	32.36
npm:@babel/generator	712	32.32
npm:@babel/template	711	32.27
npm:@babel/traverse	711	32.27
npm:es-abstract	711	32.27
npm:es-to-primitive	711	32.27
npm:file-entry-cache	711	32.27
npm:@babel/helper-function-name	710	32.23
npm:@babel/helper-split-export- declaration	708	32.14
npm:hosted-git-info	707	32.09
npm:performance-now	707	32.09
npm:entities	706	32.05
npm:escalade	701	31.82
npm:merge-stream	701	31.82
npm:esquery	697	31.64
npm:@babel/helper-validator- identifier	695	31.55
npm:indent-string	692	31.41
npm:follow-redirects	691	31.37
npm:sax	691	31.37
npm:bluebird	689	31.28
npm:spdx-expression-parse	689	31.28
npm:spdx-license-ids	689	31.28

npm:commondir	685	31.09
npm:json-schema	685	31.09
npm:normalize-package-data	680	30.87
npm:is-core-module	679	30.82
npm:parent-module	679	30.82
npm:postcss-value-parser	679	30.82
npm:spdx-correct	678	30.78
npm:validate-npm-package-license	678	30.78
npm:regenerate	676	30.69
npm:ajv-keywords	675	30.64
npm:regjsparser	675	30.64
npm:@nodelib/fs.stat	673	30.55
npm:@babel/helper-module- imports	672	30.50
npm:@babel/helpers	671	30.46
npm:toidentifier	671	30.46
npm:regexpu-core	670	30.41
npm:json-stable-stringify-without- jsonify	669	30.37
npm:is-plain-obj	668	30.32
npm:requires-port	668	30.32
npm:is-wsl	666	30.23
npm:@babel/runtime	664	30.14
npm:find-cache-dir	663	30.10
npm:react	662	30.05

npm:typescript	662	30.05
npm:@babel/helper-module-transforms	661	30.00
npm:@babel/helper-simple-access	661	30.00
npm:tapable	657	29.82
npm:ci-info	656	29.78
npm:enhanced-resolve	653	29.64
npm:eventemitter3	653	29.64
npm:pump	653	29.64
npm:eslint-visitor-keys	651	29.55
npm:os-tmpdir	649	29.46
npm:through	649	29.46
npm:cssesc	648	29.41
npm:neo-async	648	29.41
npm:dir-glob	645	29.28
npm:get-intrinsic	645	29.28
npm:watchpack	645	29.28
npm:repeat-string	644	29.23
npm:call-bind	643	29.19
npm:regenerator-transform	643	29.19
npm:@babel/helper-hoist-variables	642	29.14
npm:@babel/helper-plugin-utils	642	29.14
npm:dom-serializer	642	29.14
npm:domelementtype	642	29.14
npm:domutils	642	29.14

npm:fast-glob	641	29.10
npm:flatted	641	29.10
npm:react-dom	641	29.10
npm:merge2	640	29.05
npm:json-stringify-safe	639	29.01
npm:read-pkg	635	28.82
npm:serialize-javascript	635	28.82
npm:write-file-atomic	635	28.82
npm:psl	634	28.78
npm:is-extendable	633	28.73
npm:on-headers	632	28.69
npm:cosmiconfig	631	28.64
npm:webidl-conversions	630	28.60
npm:postcss-selector-parser	628	28.51
npm:tr46	628	28.51
npm:require-main-filename	626	28.42
npm:minimalistic-assert	625	28.37
npm:which-module	625	28.37
npm:big.js	624	28.33
npm:resolve-cwd	624	28.33
npm:webpack-sources	624	28.33
npm:whatwg-url	624	28.33
npm:loader-utils	620	28.14
npm:domhandler	617	28.01
npm:emojis-list	617	28.01

npm:import-local	614	27.87
npm:tunnel-agent	614	27.87
npm:jsbn	612	27.78
npm:babel-loader	610	27.69
npm:co	610	27.69
npm:request	609	27.64
npm:he	608	27.60
npm:is-windows	606	27.51
npm:autoprefixer	604	27.42
npm:chownr	603	27.37
npm:loader-runner	603	27.37
npm:spdx-exceptions	603	27.37
npm:for-in	602	27.33
npm:tweetnacl	602	27.33
npm:component-emitter	601	27.28
npm:@babel/plugin-syntax-object-rest-spread	600	27.24
npm:concat-stream	599	27.19
npm:p-map	598	27.14
npm:typedarray	598	27.14
npm:arr-flatten	597	27.10
npm:schema-utils	597	27.10
npm:side-channel	597	27.10
npm:aproba	596	27.05
npm:asn1	595	27.01

npm:css-loader	595	27.01
npm:nth-check	595	27.01
npm:assert-plus	594	26.96
npm:boolbase	593	26.92
npm:extsprintf	592	26.87
npm:isstream	592	26.87
npm:verror	591	26.83
npm:arr-diff	590	26.78
npm:bcrypt-pbkdf	590	26.78
npm:abbrev	589	26.74
npm:array-unique	589	26.74
npm:repeat-element	589	26.74
npm:dashdash	588	26.69
npm:nopt	588	26.69
npm:caseless	587	26.65
npm:ecc-jsbn	587	26.65
npm:expand-brackets	587	26.65
npm:extglob	587	26.65
npm:getpass	587	26.65
npm:nan	587	26.65
npm:sshpk	587	26.65
npm:decode-uri-component	586	26.60
npm:css-what	585	26.55
npm:forever-agent	585	26.55
npm:terser	585	26.55

npm:@babel/plugin-syntax-optional-catch-binding	584	26.51
npm:aws-sign2	584	26.51
npm:aws4	584	26.51
npm:css-select	584	26.51
npm:http-signature	584	26.51
npm:jsprim	584	26.51
npm:read-pkg-up	584	26.51
npm:htmlparser2	582	26.42
npm:parse5	582	26.42
npm:@babel/plugin-syntax-async-generators	581	26.37
npm:@babel/plugin-syntax-json-strings	580	26.33
npm:atob	580	26.33
npm:jest-worker	580	26.33
npm:http-proxy	579	26.28
npm:run-parallel	577	26.19
npm:gensync	576	26.15
npm:normalize-url	576	26.15
npm:compression	572	25.96
npm:normalize-range	572	25.96
npm:@babel/helper-optimise-call-expression	571	25.92
npm:@babel/helper-replace-supers	571	25.92

npm:lines-and-columns	571	25.92
npm:react-is	571	25.92
npm:source-map-resolve	571	25.92
npm:@babel/helper-member-expression-to-functions	570	25.87
npm:cli-cursor	569	25.83
npm:restore-cursor	569	25.83
npm:tmp	569	25.83
npm:compressible	567	25.74
npm:oauth-sign	567	25.74
npm:@babel/compat-data	564	25.60
npm:@babel/helper-compilation-targets	564	25.60
npm:har-validator	564	25.60
npm:reusify	564	25.60
npm:acorn-walk	563	25.56
npm:extend-shallow	563	25.56
npm:fastq	563	25.56
npm:remove-trailing-separator	563	25.56
npm:@nodelib/fs.scandir	561	25.47
npm:@nodelib/fs.walk	561	25.47
npm:pico-colors	560	25.42
npm:pinkie	558	25.33
npm:pinkie-promise	557	25.28
npm:is-obj	556	25.24

npm:@babel/preset-env	554	25.15
npm:@types/json-schema	550	24.97
npm:har-schema	549	24.92
npm:p-finally	549	24.92
npm:nanoid	545	24.74
npm:dotenv	543	24.65
npm:istanbul-lib-coverage	543	24.65
npm:number-is-nan	542	24.60
npm:slice-ansi	542	24.60
npm:through2	542	24.60
npm:@babel/helper-annotate-as-pure	541	24.56
npm:faye-websocket	540	24.51
npm:strip-indent	540	24.51
npm:lodash.memoize	539	24.47
npm:postcss-modules-extract-imports	539	24.47
npm:postcss-modules-local-by-default	539	24.47
npm:postcss-modules-scope	539	24.47
npm:postcss-modules-values	539	24.47
npm:prop-types	538	24.42
npm:webpack-dev-server	538	24.42
npm:websocket-extensions	538	24.42
npm:websocket-driver	536	24.33

npm:regjsgen	535	24.29
npm:figures	534	24.24
npm:source-map-url	534	24.24
npm:@babel/plugin-transform-modules-commonjs	533	24.19
npm:escodegen	533	24.19
npm:map-cache	533	24.19
npm:pako	533	24.19
npm:agent-base	532	24.15
npm:minipass	532	24.15
npm:https-proxy-agent	531	24.10
npm:istanbul-reports	531	24.10
npm:progress	531	24.10
npm:resolve-url	531	24.10
npm:tar	530	24.06
npm:istanbul-lib-report	529	24.01
npm:regexp.prototype.flags	529	24.01
npm:istanbul-lib-instrument	528	23.97
npm:urix	528	23.97
npm:node-forge	527	23.92
npm:queue-microtask	526	23.88
npm:arr-union	525	23.83
npm:object.pick	525	23.83
npm:chrome-trace-event	523	23.74

npm:@babel/plugin-transform-parameters	522	23.69
npm:load-json-file	522	23.69
npm:set-value	522	23.69
npm:upath	522	23.69
npm:word-wrap	522	23.69
npm:@babel/plugin-transform-arrow-functions	521	23.65
npm:@babel/plugin-transform-block-scoping	521	23.65
npm:@babel/plugin-transform-classes	521	23.65
npm:@babel/plugin-transform-computed-properties	521	23.65
npm:@babel/plugin-transform-destructuring	521	23.65
npm:@babel/plugin-transform-literals	521	23.65
npm:@babel/plugin-transform-shorthand-properties	521	23.65
npm:@babel/plugin-transform-spread	521	23.65
npm:@babel/plugin-transform-template-literals	521	23.65
npm:batch	521	23.65

npm:@babel/plugin-transform-async-to-generator	520	23.60
npm:ansi-colors	520	23.60
npm:process	520	23.60
npm:regenerate-unicode-properties	520	23.60
npm:unicode-canonical-property-names-ecmascript	520	23.60
npm:unicode-match-property-ecmascript	520	23.60
npm:unicode-match-property-value-ecmascript	520	23.60
npm:unicode-property-aliases-ecmascript	520	23.60
npm:@babel/plugin-transform-for-of	519	23.56
npm:@babel/plugin-transform-function-name	519	23.56
npm:clean-css	519	23.56
npm:get-value	519	23.56
npm:split-string	519	23.56
npm:webpack-dev-middleware	519	23.56
npm:@babel/plugin-transform-block-scoped-functions	518	23.51
npm:@babel/plugin-transform-exponentiation-operator	518	23.51

npm:istanbul-lib-source-maps	518	23.51
npm:@babel/plugin-syntax-top-level-await	517	23.47
npm:@babel/plugin-transform-regenerator	517	23.47
npm:assign-symbols	517	23.47
npm:connect-history-api-fallback	517	23.47
npm:ret	517	23.47
npm:serve-index	517	23.47
npm:union-value	517	23.47
npm:@babel/helper-builder-binary-assignment-operator-visitor	516	23.42
npm:@babel/helper-remap-async-to-generator	516	23.42
npm:@babel/helper-wrap-function	516	23.42
npm:@babel/plugin-transform-object-super	516	23.42
npm:@babel/plugin-transform-sticky-regex	516	23.42
npm:@babel/plugin-transform-unicode-regex	516	23.42
npm:to-object-path	516	23.42
npm:@babel/plugin-syntax-optional-chaining	515	23.38
npm:mixin-deep	515	23.38

npm:q	515	23.38
npm:@babel/plugin-syntax-nullish-coalescing-operator	514	23.33
npm:define-property	514	23.33
npm:is-accessor-descriptor	514	23.33
npm:is-data-descriptor	514	23.33
npm:is-descriptor	514	23.33
npm:json-parse-even-better-errors	514	23.33
npm:safe-regex	514	23.33
npm:terser-webpack-plugin	514	23.33
npm:@babel/plugin-syntax-dynamic-import	513	23.29
npm:@webassemblyjs/ast	513	23.29
npm:@webassemblyjs/floating-point-hex-parser	513	23.29
npm:@webassemblyjs/helper-wasm-bytecode	513	23.29
npm:@webassemblyjs/wast-printer	513	23.29
npm:collection-visit	513	23.29
npm:eslint-plugin-import	513	23.29
npm:has-ansi	513	23.29
npm:has-values	513	23.29
npm:map-visit	513	23.29
npm:object-visit	513	23.29
npm:pascalcase	513	23.29

npm:@babel/plugin-transform-modules-amd	512	23.24
npm:@webassemblyjs/helper-buffer	512	23.24
npm:@webassemblyjs/helper-wasm-section	512	23.24
npm:@webassemblyjs/leb128	512	23.24
npm:@webassemblyjs/wasm-edit	512	23.24
npm:@webassemblyjs/wasm-gen	512	23.24
npm:@webassemblyjs/wasm-opt	512	23.24
npm:@webassemblyjs/wasm-parser	512	23.24
npm:base	512	23.24
npm:cache-base	512	23.24
npm:class-utils	512	23.24
npm:copy-descriptor	512	23.24
npm:has-value	512	23.24
npm:invariant	512	23.24
npm:object-copy	512	23.24
npm:regex-not	512	23.24
npm:snapdragon	512	23.24
npm:snapdragon-node	512	23.24
npm:snapdragon-util	512	23.24
npm:static-extend	512	23.24
npm:to-regex	512	23.24
npm:unset-value	512	23.24
npm:use	512	23.24

npm:@babel/plugin-transform-dotall-regex	511	23.20
npm:@babel/plugin-transform-duplicate-keys	511	23.20
npm:@babel/plugin-transform-modules-systemjs	511	23.20
npm:@babel/plugin-transform-modules-umd	511	23.20
npm:@babel/plugin-transform-new-target	511	23.20
npm:@babel/plugin-transform-typeof-symbol	511	23.20
npm:fragment-cache	511	23.20
npm:nanomatch	511	23.20
npm:posix-character-classes	511	23.20
npm:source-list-map	511	23.20
npm:supports-preserve-symlinks-flag	510	23.15
npm:@webassemblyjs/helper-api-error	509	23.10
npm:http-proxy-middleware	509	23.10
npm:@webassemblyjs/ieee754	508	23.06
npm:@webassemblyjs/utf8	508	23.06
npm:icss-utils	508	23.06
npm:test-exclude	508	23.06

npm:leven	507	23.01
npm:url	506	22.97
npm:util	506	22.97
npm:yocto-queue	506	22.97
npm:@xtuc/long	505	22.92
npm:@xtuc/ieee754	504	22.88
npm:html-entities	504	22.88
npm:http-parser-js	504	22.88
npm:require-from-string	504	22.88
npm:strip-final-newline	504	22.88
npm:@babel/plugin-transform-named-capturing-groups-regex	503	22.83
npm:@babel/plugin-syntax-numeric-separator	501	22.74
npm:code-point-at	501	22.74
npm:node-fetch	500	22.70
npm:setimmediate	500	22.70
npm:strip-eof	499	22.65
npm:@babel/helper-validator-option	498	22.61
npm:async-each	498	22.61
npm:human-signals	498	22.61
npm:lodash.merge	498	22.61
npm:querystring	498	22.61
npm:file-loader	496	22.51

npm:is-string	495	22.47
npm:json-parse-better-errors	495	22.47
npm:lodash.uniq	494	22.42
npm:mute-stream	494	22.42
npm:clone	493	22.38
npm:core-js-compat	493	22.38
npm:object.values	493	22.38
npm:@babel/plugin-transform-member-expression-literals	491	22.29
npm:@babel/plugin-transform-property-literals	491	22.29
npm:detect-node	491	22.29
npm:string.prototype.trimend	489	22.20
npm:string.prototype.trimstart	489	22.20
npm:deep-equal	488	22.15
npm:ip	488	22.15
npm:errno	485	22.02
npm:prr	485	22.02
npm:regexp	485	22.02
npm:retry	485	22.02
npm:@babel/plugin-transform-reserved-words	484	21.97
npm:array-includes	484	21.97
npm:deepmerge	484	21.97
npm:sockjs	484	21.97

npm:wide-align	484	21.97
npm:redent	483	21.92
npm:@babel/helper-create-class-features-plugin	482	21.88
npm:uglify-js	482	21.88
npm:diff	481	21.83
npm:asap	480	21.79
npm:querystringify	480	21.79
npm:url-parse	480	21.79
npm:@jridgewell/sourcemap-codec	479	21.74
npm:array-uniq	479	21.74
npm:yaml	478	21.70
npm:exit	477	21.65
npm:nice-try	477	21.65
npm:style-loader	476	21.61
npm:thunky	476	21.61
npm:dns-packet	475	21.56
npm:lower-case	475	21.56
npm:multicast-dns	475	21.56
npm:obuf	475	21.56
npm:rxjs	475	21.56
npm:handle-thing	474	21.52
npm:hpack.js	474	21.52
npm:http-deceiver	474	21.52
npm:select-hose	474	21.52

npm:spdy	474	21.52
npm:spdy-transport	474	21.52
npm:wbuf	474	21.52
npm:@babel/plugin-syntax-class-properties	473	21.47
npm:sha.js	473	21.47
npm:builtin-modules	472	21.43
npm:eslint-utils	471	21.38
npm:no-case	471	21.38
npm:del	470	21.33
npm:@jridgewell/resolve-uri	469	21.29
npm:csso	468	21.24
npm:path-browserify	468	21.24
npm:pretty-format	468	21.24
npm:functional-red-black-tree	467	21.20
npm:selfsigned	467	21.20
npm:svgo	467	21.20
npm:delegates	466	21.15
npm:inquirer	466	21.15
npm:@jridgewell/trace-mapping	465	21.11
npm:babel-plugin-istanbul	465	21.11
npm:shell-quote	465	21.11
npm:arrify	464	21.06
npm:camel-case	464	21.06
npm:source-map-js	464	21.06

npm:fs-minipass	463	21.02
npm:abab	461	20.93
npm:kleur	461	20.93
npm:@babel/helper-create-regexp-features-plugin	460	20.88
npm:param-case	460	20.88
npm:has-unicode	459	20.84
npm:interpret	459	20.84
npm:gauge	458	20.79
npm:postcss-loader	458	20.79
npm:are-we-there-yet	457	20.74
npm:cli-width	457	20.74
npm:npmlog	457	20.74
npm:open	457	20.74
npm:path-is-inside	457	20.74
npm:run-async	457	20.74
npm:type-detect	457	20.74
npm:@babel/plugin-syntax-logical-assignment-operators	456	20.70
npm:console-control-strings	456	20.70
npm:eslint-plugin-react	456	20.70
npm:axios	455	20.65
npm:@babel/plugin-syntax-jsx	454	20.61
npm:jest	454	20.61

npm:@babel/plugin-proposal-object-rest-spread	453	20.56
npm:dns-equal	453	20.56
npm:duplexer	452	20.52
npm:glob-to-regexp	451	20.47
npm:is-path-cwd	451	20.47
npm:node-int64	451	20.47
npm:ripemd160	451	20.47
npm:@babel/plugin-proposal-class-properties	450	20.43
npm:deep-extend	450	20.43
npm:jsdom	450	20.43
npm:acorn-globals	449	20.38
npm:is-ci	449	20.38
npm:bn.js	447	20.29
npm:browserify-aes	447	20.29
npm:is-negative-zero	447	20.29
npm:postcss-load-config	447	20.29
npm:assert	446	20.25
npm:bser	446	20.25
npm:cssnano	446	20.25
npm:fb-watchman	446	20.25
npm:global-prefix	446	20.25
npm:internal-slot	446	20.25
npm:lodash.debounce	446	20.25

npm:prettier	445	20.20
npm:wordwrap	445	20.20
npm:babel-jest	443	20.11
npm:astral-regex	442	20.06
npm:caniuse-api	442	20.06
npm:eslint-import-resolver-node	442	20.06
npm:html-webpack-plugin	442	20.06
npm:promise	442	20.06
npm:@eslint/eslintrc	441	20.02
npm:makeerror	441	20.02
npm:postcss-calc	441	20.02
npm:postcss-discard-duplicates	441	20.02
npm:relateurl	441	20.02
npm:tmpl	441	20.02
npm:tsutils	441	20.02
npm:walker	441	20.02
npm:@babel/plugin-proposal- unicode-property-regex	440	19.97
npm:eslint-module-utils	440	19.97
npm:global-modules	440	19.97
npm:is-boolean-object	440	19.97
npm:is-number-object	440	19.97
npm:jest-util	440	19.97
npm:postcss-svg	440	19.97
npm:stream-browserify	440	19.97

npm:string-length	440	19.97
npm:console-browserify	439	19.93
npm:path-dirname	439	19.93
npm:postcss-colormin	439	19.93
npm:postcss-convert-values	439	19.93
npm:postcss-discard-comments	439	19.93
npm:postcss-discard-empty	439	19.93
npm:postcss-discard-overrides	439	19.93
npm:postcss-merge-longhand	439	19.93
npm:postcss-merge-rules	439	19.93
npm:postcss-minify-font-values	439	19.93
npm:postcss-minify-gradients	439	19.93
npm:postcss-minify-params	439	19.93
npm:postcss-minify-selectors	439	19.93
npm:postcss-normalize-charset	439	19.93
npm:postcss-normalize-url	439	19.93
npm:postcss-ordered-values	439	19.93
npm:postcss-reduce-initial	439	19.93
npm:postcss-reduce-transforms	439	19.93
npm:postcss-unique-selectors	439	19.93
npm:table	438	19.88
npm:elliptic	437	19.84
npm:hash.js	437	19.84
npm:ssri	437	19.84
npm:brorand	436	19.79

npm:crypto-browserify	436	19.79
npm:csstype	436	19.79
npm:jest-diff	435	19.75
npm:xml-name-validator	435	19.75
npm:colorette	434	19.70
npm:hmac-drbg	434	19.70
npm:minimalistic-crypto-utils	434	19.70
npm:os-homedir	434	19.70
npm:browserify-zlib	433	19.66
npm:cipher-base	433	19.66
npm:create-hash	433	19.66
npm:create-hmac	433	19.66
npm:detect-newline	433	19.66
npm:@babel/plugin-proposal-optional-catch-binding	432	19.61
npm:@types/estree	432	19.61
npm:pirates	432	19.61
npm:whatwg-encoding	432	19.61
npm:@typescript-eslint/parser	431	19.56
npm:html-encoding-sniffer	431	19.56
npm:scheduler	431	19.56
npm:cssstyle	430	19.52
npm:has-tostringtag	430	19.52
npm:prepend-http	430	19.52

npm:@babel/plugin-proposal- async-generator-functions	429	19.47
npm:buffer-xor	429	19.47
npm:evp_bytestokey	429	19.47
npm:hash-base	429	19.47
npm:jest-mock	429	19.47
npm:symbol-tree	429	19.47
npm:jest-message-util	428	19.43
npm:jest-regex-util	428	19.43
npm:pseudomap	428	19.43
npm:sass-loader	428	19.43
npm:jest-haste-map	427	19.38
npm:jest-matcher-utils	427	19.38
npm:jest-docblock	426	19.34
npm:jest-environment-node	426	19.34
npm:is-bigint	425	19.29
npm:jest-validate	425	19.29
npm:pbkdf2	425	19.29
npm:@babel/preset-modules	424	19.25
npm:external-editor	424	19.25
npm:which-boxed-primitive	424	19.25
npm:asn1.js	423	19.20
npm:constants-browserify	423	19.20
npm:cssom	423	19.20
npm:https-browserify	423	19.20

npm:@jridgewell/gen-mapping	422	19.16
npm:@jridgewell/set-array	422	19.16
npm:builtin-status-codes	422	19.16
npm:stream-http	422	19.16
npm:@babel/plugin-proposal-json-strings	421	19.11
npm:is-docker	421	19.11
npm:jest-snapshot	421	19.11
npm:minizlib	421	19.11
npm:os-browserify	421	19.11
npm:vm-browserify	421	19.11
npm:@typescript-eslint/typescript-estree	420	19.06
npm:domain-browser	420	19.06
npm:dot-prop	420	19.06
npm:jest-resolve	420	19.06
npm:timers-browserify	420	19.06
npm:tty-browserify	420	19.06
npm:babel-plugin-jest-hoist	419	19.02
npm:babel-preset-jest	419	19.02
npm:dom-converter	419	19.02
npm:has-bigints	419	19.02
npm:html-escaper	419	19.02
npm:http-proxy-agent	419	19.02
npm:jest-changed-files	419	19.02

npm:jest-resolve-dependencies	419	19.02
npm:jest-runtime	419	19.02
npm:pretty-error	419	19.02
npm:renderkid	419	19.02
npm:utila	419	19.02
npm:cacache	418	18.97
npm:jest-config	418	18.97
npm:mini-css-extract-plugin	418	18.97
npm:querystring-es3	418	18.97
npm:unique-filename	418	18.97
npm:unique-slug	418	18.97
npm:v8-compile-cache	418	18.97
npm:babel-runtime	417	18.93
npm:object.getownpropertydescriptors	417	18.93
npm:des.js	416	18.88
npm:jest-get-type	416	18.88
npm:md5.js	416	18.88
npm:stack-utils	416	18.88
npm:default-gateway	415	18.84
npm:log-symbols	415	18.84
npm:sisteransi	415	18.84
npm:browserify-cipher	414	18.79
npm:browserify-des	414	18.79
npm:miller-rabin	414	18.79

npm:@babel/helper-environment-visitor	413	18.75
npm:browserify-rsa	413	18.75
npm:browserify-sign	413	18.75
npm:create-ecdh	413	18.75
npm:diffie-hellman	413	18.75
npm:parse-asn1	413	18.75
npm:public-encrypt	413	18.75
npm:rc	413	18.75
npm:promise-inflight	412	18.70
npm:prompts	412	18.70
npm:@babel/helper-explode-assignable-expression	410	18.61
npm:is-arguments	410	18.61
npm:@types/react	408	18.52
npm:@typescript-eslint/eslint-plugin	408	18.52
npm:duplexify	408	18.52
npm:stream-shift	408	18.52
npm:chardet	407	18.47
npm:@types/yargs	404	18.34
npm:unbox-primitive	403	18.29
npm:expect	402	18.25
npm:clean-stack	401	18.20
npm:jsx-ast-utils	398	18.07

npm:@babel/helper-skip-transparent-expression-wrappers	397	18.02
npm:@babel/plugin-proposal-optional-chaining	396	17.98
npm:coa	396	17.98
npm:@types/istanbul-lib-coverage	395	17.93
npm:@types/yargs-parser	395	17.93
npm:css-tree	395	17.93
npm:to-arraybuffer	395	17.93
npm:whatwg-mimetype	395	17.93
npm:@babel/plugin-transform-runtime	394	17.88
npm:aggregate-error	394	17.88
npm:mdn-data	394	17.88
npm:memory-fs	394	17.88
npm:@humanwhocodes/config-array	393	17.84
npm:@humanwhocodes/object-schema	393	17.84
npm:diff-sequences	393	17.84
npm:jest-runner	392	17.79
npm:object-is	392	17.79
npm:@babel/plugin-proposal-nullish-coalescing-operator	391	17.75
npm:is-generator-fn	390	17.70

npm:magic-string	390	17.70
npm:update-browserslist-db	390	17.70
npm:jest-leak-detector	389	17.66
npm:@babel/plugin-transform-unicode-escapes	388	17.61
npm:domexception	388	17.61
npm:@ampproject/remapping	387	17.57
npm:@types/babel__traverse	386	17.52
npm:data-urls	386	17.52
npm:gzip-size	386	17.52
npm:@types/babel__core	385	17.48
npm:@types/babel__generator	385	17.48
npm:@types/babel__template	385	17.48
npm:@jest/types	384	17.43
npm:@babel/plugin-syntax-export-namespace-from	382	17.34
npm:@types/istanbul-lib-report	382	17.34
npm:@types/istanbul-reports	382	17.34
npm:aria-query	382	17.34
npm:nwsapi	382	17.34
npm:randomfill	382	17.34
npm:async-limiter	381	17.29
npm:color	381	17.29
npm:jest-each	380	17.25
npm:jest-environment-jsdom	380	17.25

npm:jest-watcher	380	17.25
npm:node-libs-browser	379	17.20
npm:babel-plugin-dynamic-import- node	378	17.16
npm:colors	378	17.16
npm:util.promisify	378	17.16
npm:invert-kv	377	17.11
npm:lcid	377	17.11
npm:os-locale	377	17.11
npm:rollup	377	17.11
npm:stable	377	17.11
npm:@babel/plugin-proposal- dynamic-import	376	17.07
npm:color-string	376	17.07
npm:clone-deep	375	17.02
npm:has-property-descriptors	375	17.02
npm:shallow-clone	375	17.02
npm:@types/parse-json	374	16.98
npm:@babel/plugin-syntax- typescript	372	16.89
npm:@types/stack-utils	372	16.89
npm:is-shared-array-buffer	372	16.89
npm:object.entries	372	16.89
npm:@istanbuljs/schema	371	16.84
npm:jest-cli	371	16.84

npm:throat	371	16.84
npm:@babel/plugin-proposal-numeric-separator	370	16.80
npm:jest-pnp-resolver	370	16.80
npm:object.fromentries	369	16.75
npm:@jest/fake-timers	367	16.66
npm:@types/prop-types	367	16.66
npm:array.prototype.flat	367	16.66
npm:is-path-in-cwd	367	16.66
npm:whatwg-fetch	367	16.66
npm:@jest/environment	366	16.61
npm:p-retry	366	16.61
npm:@babel/helper-get-function-arity	365	16.57
npm:portfinder	365	16.57
npm:@jest/transform	364	16.52
npm:@jest/console	363	16.48
npm:@jest/source-map	363	16.48
npm:@jest/test-result	363	16.48
npm:get-symbol-description	363	16.48
npm:object-hash	361	16.39
npm:@jest/reporters	360	16.34
npm:ejs	360	16.34
npm:@jest/core	359	16.30
npm:@jest/test-sequencer	359	16.30

npm:caller-path	359	16.30
npm:crypto-random-string	359	16.30
npm:filesize	359	16.30
npm:pretty-bytes	359	16.30
npm:unique-string	359	16.30
npm:eslint-plugin-jsx-a11y	356	16.16
npm:is-weakref	356	16.16
npm:browser-process-hrtime	354	16.07
npm:pumpify	354	16.07
npm:query-string	354	16.07
npm:get-stdin	353	16.02
npm:w3c-hr-time	353	16.02
npm:stylehacks	350	15.89
npm:typedarray-to-buffer	350	15.89
npm:xmlchars	350	15.89
npm:saxes	349	15.84
npm:css-declaration-sorter	348	15.80
npm:cssnano-preset-default	348	15.80
npm:postcss-normalize-display-values	348	15.80
npm:postcss-normalize-positions	348	15.80
npm:postcss-normalize-repeat-style	348	15.80
npm:postcss-normalize-string	348	15.80
npm:postcss-normalize-timing-functions	348	15.80

npm:postcss-normalize-unicode	348	15.80
npm:postcss-normalize-whitespace	348	15.80
npm:@tootallnate/once	347	15.75
npm:axobject-query	347	15.75
npm:babel-code-frame	347	15.75
npm:bl	346	15.71
npm:@istanbuljs/load-nyc-config	345	15.66
npm:@typescript-eslint/types	345	15.66
npm:@typescript-eslint/visitor- keys	345	15.66
npm:babel-plugin-polyfill-corejs2	345	15.66
npm:babel-plugin-polyfill-corejs3	345	15.66
npm:babel-plugin-polyfill- regenerator	345	15.66
npm:babylon	345	15.66
npm:@babel/helper-define-polyfill- provider	344	15.62
npm:get-package-type	344	15.62
npm:uniq	344	15.62
npm:w3c-xmlserializer	344	15.62
npm:@typescript-eslint/scope- manager	343	15.57
npm:jest-jasmine2	343	15.57
npm:@babel/plugin-syntax- import-meta	342	15.52

npm:is-promise	342	15.52
npm:opn	341	15.48
npm:es5-ext	340	15.43
npm:moment	340	15.43
npm:strict-uri-encode	340	15.43
npm:@babel/helper-string-parser	339	15.39
npm:@babel/plugin-bugfix-v8-spread-parameters-in-optional-chaining	339	15.39
npm:@babel/plugin-transform-react-jsx	338	15.34
npm:@types/q	338	15.34
npm:for-own	338	15.34
npm:damerau-levenshtein	337	15.30
npm:ast-types-flow	335	15.21
npm:d	335	15.21
npm:es6-iterator	335	15.21
npm:es6-symbol	335	15.21
npm:is-utf8	335	15.21
npm:@sinonjs/commons	334	15.16
npm:dotenv-expand	334	15.16
npm:from2	334	15.16
npm:pkg-up	334	15.16
npm:num2fraction	333	15.12
npm:functions-have-names	332	15.07

npm:flush-write-stream	331	15.02
npm:@babel/plugin-proposal-private-property-in-object	329	14.93
npm:@babel/plugin-syntax-class-static-block	329	14.93
npm:@babel/plugin-syntax-private-property-in-object	329	14.93
npm:is-finite	329	14.93
npm:@babel/plugin-transform-react-display-name	328	14.89
npm:@babel/preset-react	327	14.84
npm:indexes-of	327	14.84
npm:estree-walker	326	14.80
npm:ip-regex	326	14.80
npm:osenv	326	14.80
npm:@types/express	325	14.75
npm:camelcase-keys	325	14.75
npm:lodash.sortby	325	14.75
npm:map-obj	325	14.75
npm:meow	325	14.75
npm:url-loader	325	14.75
npm:address	324	14.71
npm:is-directory	324	14.71
npm:rechoir	324	14.71
npm:arg	323	14.66

npm:worker-farm	322	14.62
npm:trim-newlines	321	14.57
npm:@babel/plugin-proposal-private-methods	320	14.53
npm:@types/eslint	320	14.53
npm:@types/express-serve-static-core	320	14.53
npm:@types/mime	320	14.53
npm:@types/qs	320	14.53
npm:jest-serializer	320	14.53
npm:@types/serve-static	319	14.48
npm:eslint-plugin-react-hooks	319	14.48
npm:@types/minimatch	318	14.43
npm:infer-owner	318	14.43
npm:is-absolute-url	318	14.43
npm:tsconfig-paths	318	14.43
npm:@types/connect	316	14.34
npm:@types/body-parser	314	14.25
npm:react-scripts	314	14.25
npm:repeating	314	14.25
npm:string.prototype.matchall	314	14.25
npm:file-uri-to-path	313	14.21
npm:unquote	311	14.12
npm:bindings	310	14.07
npm:@types/range-parser	309	14.03

npm:private	308	13.98
npm:sort-keys	308	13.98
npm:got	307	13.94
npm:json3	307	13.94
npm:lowercase-keys	307	13.94
npm:@bcoe/v8-coverage	305	13.84
npm:babel-eslint	305	13.84
npm:dedent	304	13.80
npm:detect-indent	304	13.80
npm:is-resolvable	304	13.80
npm:function.prototype.name	303	13.75
npm:array.prototype.flatmap	302	13.71
npm:at-least-node	302	13.71
npm:case-sensitive-paths-webpack- plugin	302	13.71
npm:jquery	302	13.71
npm:mem	302	13.71
npm:v8-to-istanbul	302	13.71
npm:@sinonjs/fake-timers	301	13.66
npm:css-select-base-adapter	301	13.66
npm:min-indent	301	13.66
npm:sourcemap-codec	301	13.66
npm:write	300	13.62
npm:@types/json5	299	13.57
npm:fs-write-stream-atomic	299	13.57

npm:hoist-non-react-statics	299	13.57
npm:iferr	298	13.53
npm:@babel/plugin-transform- typescript	297	13.48
npm:copy-concurrently	297	13.48
npm:move-concurrently	297	13.48
npm:run-queue	297	13.48
npm:@types/glob	295	13.39
npm:cyclist	295	13.39
npm:for-each	295	13.39
npm:parallel-transform	295	13.39
npm:buffer-indexof	294	13.35
npm:http-cache-semantics	294	13.35
npm:mississippi	294	13.35
npm:multicast-dns-service-types	294	13.35
npm:stream-each	294	13.35
npm:@babel/plugin-bugfix-safari- id-destructuring-collision-in- function-expression	293	13.30
npm:@babel/plugin-syntax-bigint	293	13.30
npm:bonjour	293	13.30
npm:dns-txt	293	13.30
npm:@babel/plugin-proposal- export-namespace-from	292	13.25
npm:react-router-dom	292	13.25

npm:@babel/plugin-proposal-decorators	291	13.21
npm:dot-case	291	13.21
npm:pascal-case	291	13.21
npm:alphanum-sort	290	13.16
npm:babel-core	290	13.16
npm:collect-v8-coverage	290	13.16
npm:eventsourcing	290	13.16
npm:sockjs-client	288	13.07
npm:@babel/plugin-proposal-logical-assignment-operators	287	13.03
npm:@types/graceful-fs	287	13.03
npm:babel-preset-current-node-syntax	287	13.03
npm:es-module-lexer	287	13.03
npm:loglevel	287	13.03
npm:raf	287	13.03
npm:defaults	286	12.98
npm:emittery	286	12.98
npm:jsonify	286	12.98
npm:@babel/plugin-syntax-decorators	285	12.94
npm:@babel/preset-typescript	285	12.94
npm:available-typed-arrays	285	12.94
npm:css-color-names	285	12.94

npm:which-typed-array	285	12.94
npm:@jest/globals	284	12.89
npm:defined	284	12.89
npm:error-stack-parser	284	12.89
npm:stackframe	284	12.89
npm:@types/react-dom	283	12.85
npm:ansi-html	283	12.85
npm:is-typed-array	283	12.85
npm:mocha	283	12.85
npm:core-js-pure	282	12.80
npm:encoding	282	12.80
npm:simple-swizzle	281	12.76
npm:uniqs	281	12.76
npm:vendors	281	12.76
npm:ajv-errors	280	12.71
npm:babel-plugin-macros	280	12.71
npm:char-regex	280	12.71
npm:internal-ip	280	12.71
npm:json-buffer	280	12.71
npm:keyv	279	12.66
npm:next-tick	279	12.66
npm:@types/eslint-scope	278	12.62
npm:@types/scheduler	277	12.57
npm:gopd	277	12.57
npm:detect-libc	276	12.53

npm:handlebars	276	12.53
npm:has-proto	276	12.53
npm:lazy-cache	276	12.53
npm:@webassemblyjs/helper- numbers	275	12.48
npm:cjs-module-lexer	275	12.48
npm:webpack-merge	275	12.48
npm:fraction.js	274	12.44
npm:@typescript- eslint/experimental-utils	273	12.39
npm:quick-lru	273	12.39
npm:@jridgewell/source-map	272	12.35
npm:recursive-readdir	272	12.35
npm:decimal.js	271	12.30
npm:@babel/plugin-transform- react-jsx-development	270	12.26
npm:is-regexp	270	12.26
npm:acorn-import-assertions	269	12.21
npm:common-tags	269	12.21
npm:node-gyp	269	12.21
npm:supports-hyperlinks	269	12.21
npm:amdefine	268	12.17
npm:json-stable-stringify	267	12.12
npm:react-dev-utils	267	12.12
npm:original	266	12.07

npm:react-error-overlay	266	12.07
npm:define-lazy-prop	265	12.03
npm:confusing-browser-globals	263	11.94
npm:ora	263	11.94
npm:postcss-custom-properties	263	11.94
npm:figgy-pudding	262	11.89
npm:postcss-custom-media	262	11.89
npm:postcss-nesting	262	11.89
npm:postcss-preset-env	262	11.89
npm:wcwidth	262	11.89
npm:@webassemblyjs/helper-code-frame	261	11.85
npm:@webassemblyjs/helper-fsm	261	11.85
npm:@webassemblyjs/wast-parser	261	11.85
npm:array-find-index	260	11.80
npm:cli-spinners	260	11.80
npm:is-root	260	11.80
npm:lilconfig	260	11.80
npm:postcss-attribute-case-insensitive	260	11.80
npm:postcss-color-hex-alpha	260	11.80
npm:postcss-color-rebeccapurple	260	11.80
npm:postcss-custom-selectors	260	11.80
npm:postcss-font-variant	260	11.80
npm:postcss-pseudo-class-any-link	260	11.80

npm:postcss-replace-overflow-wrap	260	11.80
npm:postcss-selector-not	260	11.80
npm:fork-ts-checker-webpack-plugin	259	11.76
npm:html-minifier-terser	259	11.76
npm:minipass-collect	259	11.76
npm:minipass-flush	259	11.76
npm:minipass-pipeline	259	11.76
npm:stringify-object	259	11.76
npm:@types/jest	258	11.71
npm:is-potential-custom-element-name	258	11.71
npm:postcss-flexbugs-fixes	258	11.71
npm:@webassemblyjs/helper-module-context	257	11.67
npm:get-own-enumerable-property-symbols	257	11.67
npm:postcss-initial	257	11.67
npm:cssdb	256	11.62
npm:postcss-dir-pseudo-class	256	11.62
npm:postcss-focus-visible	256	11.62
npm:postcss-focus-within	256	11.62
npm:postcss-gap-properties	256	11.62
npm:postcss-image-set-function	256	11.62
npm:postcss-logical	256	11.62

npm:postcss-overflow-shorthand	256	11.62
npm:postcss-page-break	256	11.62
npm:postcss-place	256	11.62
npm:postcss-color-functional-notation	255	11.58
npm:postcss-lab-function	255	11.58
npm:postcss-media-minmax	255	11.58
npm:@babel/plugin-syntax-flow	254	11.53
npm:@babel/plugin-transform-flow-strip-types	254	11.53
npm:@types/prettier	254	11.53
npm:classnames	254	11.53
npm:@humanwhocodes/module-importer	253	11.48
npm:detect-port-alt	253	11.48
npm:eslint-plugin-flowtype	253	11.48
npm:babel-types	252	11.44
npm:growly	252	11.44
npm:postcss-double-position-gradients	252	11.44
npm:react-router	252	11.44
npm:shellwords	252	11.44
npm:css-blank-pseudo	251	11.39
npm:css-has-pseudo	251	11.39
npm:css-prefers-color-scheme	251	11.39

npm:immer	251	11.39
npm:killable	251	11.39
npm:make-error	251	11.39
npm:caller-callsite	250	11.35
npm:es-shim-unsupported	250	11.35
npm:ignore-walk	250	11.35
npm:jest-circus	250	11.35
npm:node-notifier	250	11.35
npm:ajv-formats	249	11.30
npm:postcss-env-function	249	11.30
npm:eslint-config-react-app	248	11.26
npm:resolve-url-loader	248	11.26
npm:@types/http-proxy	247	11.21
npm:cors	247	11.21
npm:trim-right	247	11.21
npm:webpack-manifest-plugin	247	11.21
npm:buffer-crc32	246	11.17
npm:url-parse-lax	246	11.17
npm:@babel/plugin-transform-react-pure-annotations	245	11.12
npm:currently-unhandled	245	11.12
npm:bootstrap	244	11.08
npm:decompress-response	243	11.03
npm:import-from	243	11.03
npm:klona	243	11.03

npm:mimic-response	243	11.03
npm:@types/ws	242	10.99
npm:@typescript-eslint/utils	242	10.99
npm:@types/normalize-package-data	241	10.94
npm:@typescript-eslint/type-utils	241	10.94
npm:adjust-sourcemap-loader	241	10.94
npm:babel-preset-react-app	241	10.94
npm:regex-parser	241	10.94
npm:@rollup/pluginutils	240	10.89
npm:es6-promise	240	10.89
npm:globalthis	240	10.89
npm:npm-bundled	240	10.89
npm:npm-packlist	240	10.89
npm:@testing-library/react	239	10.85
npm:@testing-library/jest-dom	238	10.80
npm:indexof	238	10.80
npm:xmlbuilder	238	10.80
npm:chai	237	10.76
npm:loud-rejection	237	10.76
npm:smart-buffer	237	10.76
npm:xdg-basedir	237	10.76
npm:socket.io	236	10.71
npm:widest-line	236	10.71
npm:ansi-align	235	10.67

npm:global-dirs	235	10.67
npm:is-builtin-module	235	10.67
npm:@babel/plugin-syntax- import-assertions	234	10.62
npm:boxen	234	10.62
npm:cli-boxes	234	10.62
npm:@babel/plugin-proposal-class- static-block	233	10.58
npm:is-svg	233	10.58
npm:memfs	233	10.58
npm:object.hasown	233	10.58
npm:registry-url	233	10.58
npm:lodash._reinterpolate	232	10.53
npm:react-refresh	232	10.53
npm:registry-auth-token	232	10.53
npm:safe-regex-test	232	10.53
npm:ua-parser-js	232	10.53
npm:underscore	232	10.53
npm:webpack-log	232	10.53
npm:dom-helpers	231	10.49
npm:exec-sh	231	10.49
npm:fs-monkey	231	10.49
npm:lodash.template	231	10.49
npm:lodash.templatesettings	231	10.49
npm:terminal-link	231	10.49

npm>window-size	231	10.49
npm:@babel/plugin-transform-react-constant-elements	230	10.44
npm:babel-plugin-transform-react-remove-prop-types	230	10.44
npm:configstore	230	10.44
npm:flatten	230	10.44
npm:identity-obj-proxy	230	10.44
npm:optimist	230	10.44
npm:@svgr/core	229	10.39
npm:ansi-html-community	229	10.39
npm:babel-messages	229	10.39
npm:@types/retry	228	10.35
npm:babel-traverse	228	10.35
npm:css	228	10.35
npm:lodash.camelcase	228	10.35
npm:pend	228	10.35
npm:@types/connect-history-api-fallback	227	10.30
npm:react-transition-group	227	10.30
npm:socks	227	10.30
npm:sane	226	10.26
npm:@svgr/babel-plugin-add-jsx-attribute	225	10.21

npm:@svgr/babel-plugin-remove-jsx-attribute	225	10.21
npm:@svgr/babel-plugin-remove-jsx-empty-expression	225	10.21
npm:@svgr/babel-plugin-replace-jsx-attribute-value	225	10.21
npm:@svgr/babel-plugin-svg-dynamic-title	225	10.21
npm:@svgr/babel-plugin-svg-em-dimensions	225	10.21
npm:@svgr/babel-plugin-transform-react-native-svg	225	10.21
npm:@svgr/babel-plugin-transform-svg-component	225	10.21
npm:@svgr/babel-preset	225	10.21
npm:@svgr/hast-util-to-babel-ast	225	10.21
npm:@svgr/plugin-jsx	225	10.21
npm:read-cache	225	10.21
npm:@svgr/webpack	224	10.17
npm:duplexer3	224	10.17
npm:html-comment-regex	224	10.17
npm:is-unicode-supported	224	10.17
npm:p-defer	224	10.17
npm:yauzl	224	10.17
npm:babel-template	223	10.12

npm:fd-slicer	223	10.12
npm:package-json	223	10.12
npm:upper-case	223	10.12
npm:@svgr/plugin-svgo	222	10.08
npm:harmony-reflect	222	10.08
npm:homedir-polyfill	222	10.08
npm:jsonpointer	222	10.08
npm:parse-passwd	222	10.08
npm:proxy-from-env	222	10.08
npm:import-lazy	221	10.03
npm:postcss-import	221	10.03
npm:semver-diff	221	10.03
npm:symbol-observable	221	10.03
npm:import-cwd	220	9.99
npm:lodash.clonedeep	220	9.99
npm:strip-comments	220	9.99
npm:timsort	220	9.99
npm:@sinclair/typebox	219	9.94
npm:type	219	9.94
npm:@jest/schemas	218	9.90
npm:@testing-library/user-event	218	9.90
npm:check-types	218	9.90
npm:needle	218	9.90
npm:request-promise-core	218	9.90
npm:stealthy-require	218	9.90

npm:@types/resolve	217	9.85
npm:is-installed-globally	217	9.85
npm:shelljs	217	9.85
npm:@types/html-minifier-terser	216	9.80
npm:axe-core	216	9.80
npm:copy-webpack-plugin	216	9.80
npm:svg-parser	216	9.80
npm:tryer	216	9.80
npm:p-is-promise	215	9.76
npm:resolve.exports	215	9.76
npm:tar-stream	215	9.76
npm:color-support	214	9.71
npm:language-subtag-registry	214	9.71
npm:language-tags	214	9.71
npm:workbox-cacheable-response	214	9.71
npm:workbox-core	214	9.71
npm:workbox-routing	214	9.71
npm:workbox-strategies	214	9.71
npm:workbox-sw	214	9.71
npm:@types/semver	213	9.67
npm:is-array-buffer	213	9.67
npm:temp-dir	213	9.67
npm:webpack-cli	213	9.67
npm:workbox-background-sync	213	9.67
npm:workbox-build	213	9.67

npm:workbox-google-analytics	213	9.67
npm:workbox-navigation-preload	213	9.67
npm:workbox-precaching	213	9.67
npm:workbox-range-requests	213	9.67
npm:workbox-streams	213	9.67
npm:browser-resolve	212	9.62
npm:immutable	212	9.62
npm:is-module	212	9.62
npm:is-npm	212	9.62
npm:jest-watch-typeahead	212	9.62
npm:latest-version	212	9.62
npm:react-app-polyfill	212	9.62
npm:babel-plugin-named-asset- import	211	9.58
npm:env-paths	211	9.58
npm:workbox-webpack-plugin	211	9.58
npm:d3-array	210	9.53
npm:date-now	210	9.53
npm:enquirer	210	9.53
npm:js-base64	210	9.53
npm:@babel/runtime-corejs3	209	9.49
npm:esbuild	209	9.49
npm:map-age-cleaner	209	9.49
npm:node-sass	209	9.49
npm:update-notifier	209	9.49

npm:babel-generator	208	9.44
npm:expand-tilde	208	9.44
npm:@hapi/hoek	207	9.40
npm:bfj	207	9.40
npm:d3-color	207	9.40
npm:eslint-config-prettier	207	9.40
npm:fstream	207	9.40
npm:es6-weak-map	206	9.35
npm:longest	206	9.35
npm:resolve-dir	206	9.35
npm:hoopy	205	9.31
npm:is-primitive	205	9.31
npm:request-promise-native	205	9.31
npm:string.prototype.trim	205	9.31
npm:any-promise	204	9.26
npm:babel-plugin-syntax-object-rest-spread	204	9.26
npm:d3-interpolate	204	9.26
npm:ts-node	204	9.26
npm:workbox-expiration	204	9.26
npm:@hapi/topo	203	9.21
npm:workbox-broadcast-update	203	9.21
npm:workbox-window	203	9.21
npm:babel-plugin-transform-object-rest-spread	202	9.17

npm:babel-register	202	9.17
npm:glob-base	202	9.17
npm:nodemon	202	9.17
npm:object.omit	202	9.17
npm:@types/sockjs	201	9.12
npm:event-emitter	201	9.12
npm:expand-range	201	9.12
npm:ext	201	9.12
npm:rsvp	201	9.12
npm:sass	201	9.12
npm:source-map-loader	201	9.12
npm:@types/bonjour	200	9.08
npm:@types/serve-index	200	9.08
npm:css.escape	200	9.08
npm:engine.io-parser	200	9.08
npm:randomatic	200	9.08
npm:socket.io-parser	200	9.08
npm:contains-path	199	9.03
npm:is-dotfile	199	9.03
npm:parse-glob	199	9.03
npm:socks-proxy-agent	199	9.03
npm:postcss-normalize	198	8.99
npm:is-equal-shallow	197	8.94
npm:is-posix-bracket	197	8.94
npm:regex-cache	197	8.94

npm:@rollup/plugin-node-resolve	196	8.90
npm:@testing-library/dom	196	8.90
npm:filename-regex	196	8.90
npm:jake	196	8.90
npm:lodash.get	196	8.90
npm:preserve	196	8.90
npm:@csstools/normalize.css	195	8.85
npm:jsonparse	195	8.85
npm:p-cancelable	195	8.85
npm:postcss-browser-comments	195	8.85
npm:eslint-plugin-jest	194	8.81
npm:lodash.isplainobject	194	8.81
npm:node-pre-gyp	194	8.81
npm:set-immediate-shim	194	8.81
npm:es-set-tostringtag	193	8.76
npm:filelist	193	8.76
npm:typed-array-length	193	8.76
npm:d3-format	192	8.72
npm:d3-time-format	192	8.72
npm:@sindresorhus/is	191	8.67
npm:d3-time	191	8.67
npm:dom-accessibility-api	191	8.67
npm:hex-color-regex	191	8.67
npm:hsl-regex	191	8.67
npm:hsla-regex	191	8.67

npm:html-minifier	191	8.67
npm:is-color-stop	191	8.67
npm:pluralize	191	8.67
npm:rgb-regex	191	8.67
npm:rgba-regex	191	8.67
npm:warning	191	8.67
npm:@eslint-community/eslint- utils	190	8.62
npm:@eslint/js	190	8.62
npm:align-text	189	8.58
npm:camelcase-css	189	8.58
npm:center-align	189	8.58
npm:d3-scale	189	8.58
npm:right-align	189	8.58
npm:capture-exit	188	8.53
npm:colord	188	8.53
npm:uglify-to-browserify	188	8.53
npm:@babel/plugin-transform- react-jsx-source	187	8.49
npm:@eslint-community/regexpp	187	8.49
npm:array-equal	187	8.49
npm:base64id	187	8.49
npm:socket.io-adapter	187	8.49
npm:@leichtgewicht/ip-codec	186	8.44
npm:babel-helpers	186	8.44

npm:cacheable-request	186	8.44
npm:engine.io	186	8.44
npm:responselike	186	8.44
npm:findup-sync	185	8.40
npm:home-or-tmp	185	8.40
npm:cssnano-util-get-arguments	184	8.35
npm:cssnano-util-get-match	184	8.35
npm:cssnano-util-raw-cache	184	8.35
npm:cssnano-util-same-parent	184	8.35
npm:icss-replace-symbols	184	8.35
npm:is-map	184	8.35
npm:is-set	184	8.35
npm:clsx	183	8.31
npm:connect	183	8.31
npm:dlv	183	8.31
npm:xml2js	183	8.31
npm:@babel/plugin-transform-react-jsx-self	182	8.26
npm:@popperjs/core	182	8.26
npm:@trysound/sax	182	8.26
npm:d3-shape	182	8.26
npm:fastparse	182	8.26
npm:history	182	8.26
npm:postcss-nested	182	8.26
npm:rollup-plugin-terser	182	8.26

npm:tailwindcss	182	8.26
npm:browser-stdout	181	8.22
npm:yn	181	8.22
npm:bonjour-service	180	8.17
npm:extract-zip	180	8.17
npm:sanitize.css	180	8.17
npm:call-me-maybe	178	8.08
npm:didyoumean	178	8.08
npm:fs-constants	178	8.08
npm:postcss-js	178	8.08
npm:tempy	178	8.08
npm:clone-response	177	8.03
npm:redux	177	8.03
npm:block-stream	176	7.99
npm:d3-path	176	7.99
npm:acorn-dynamic-import	175	7.94
npm:css-selector-tokenizer	175	7.94
npm:deep-eql	175	7.94
npm:assertion-error	174	7.90
npm:hoek	174	7.90
npm:@npmcli/move-file	173	7.85
npm:@babel/helper-define-map	172	7.81
npm:array-buffer-byte-length	172	7.81
npm:is-weakmap	172	7.81
npm:is-weakset	172	7.81

npm:node-addon-api	172	7.81
npm:touch	172	7.81
npm:which-collection	172	7.81
npm:@szmarczak/http-timer	171	7.76
npm:defer-to-connect	171	7.76
npm:eslint-loader	171	7.76
npm:image-size	171	7.76
npm:vue	171	7.76
npm:@types/aria-query	170	7.72
npm:babel-plugin-syntax-jsx	170	7.72
npm:cssnano-utils	170	7.72
npm:find-root	170	7.72
npm:lz-string	170	7.72
npm:p-each-series	170	7.72
npm:timed-out	170	7.72
npm:babel-plugin-transform-es2015-modules-commonjs	169	7.67
npm:d3	169	7.67
npm:eslint-plugin-prettier	169	7.67
npm:d3-timer	168	7.63
npm:gaze	168	7.63
npm:opener	168	7.63
npm:rw	168	7.63
npm:thenify	168	7.63
npm:void-elements	168	7.63

npm:react-lifecycles-compat	167	7.58
npm:babel-plugin-transform-strict-mode	166	7.54
npm:d3-dispatch	166	7.54
npm:globule	166	7.54
npm:ignore-by-default	166	7.54
npm:toposort	166	7.54
npm:detect-file	165	7.49
npm:term-size	165	7.49
npm:@aashutoshrathi/word-wrap	164	7.44
npm:date-fns	164	7.44
npm:err-code	164	7.44
npm:flat	164	7.44
npm:grunt	164	7.44
npm:mz	164	7.44
npm:@emotion/memoize	163	7.40
npm:@types/lodash	163	7.40
npm:eslint-webpack-plugin	163	7.40
npm:is-interactive	163	7.40
npm:thenify-all	163	7.40
npm:babel-helper-function-name	162	7.35
npm:is-retry-allowed	162	7.35
npm:@babel/helper-regex	161	7.31
npm:@cnakazawa/watch	161	7.31

npm:@pmmmwh/react-refresh- webpack-plugin	161	7.31
npm:babel-helper-get-function- arity	161	7.31
npm:detective	161	7.31
npm:graphemer	161	7.31
npm:is-generator-function	161	7.31
npm:socket.io-client	161	7.31
npm:@rollup/plugin-replace	160	7.26
npm:agentkeepalive	160	7.26
npm:babel-plugin-syntax-trailing- function-commas	160	7.26
npm:duplexer2	160	7.26
npm:@emotion/hash	159	7.22
npm:@types/react-transition-group	159	7.22
npm:ast-types	159	7.22
npm:@emotion/unitless	158	7.17
npm:fast-diff	158	7.17
npm:hasown	158	7.17
npm:immediate	158	7.17
npm:npm-normalize-package-bin	158	7.17
npm:undefsafe	158	7.17
npm:@babel/eslint-parser	157	7.13
npm:babel-plugin-transform- es2015-block-scoping	157	7.13

npm:circular-json	157	7.13
npm:common-path-prefix	157	7.13
npm:css-minimizer-webpack-plugin	157	7.13
npm:envinfo	157	7.13
npm:node-modules-regexp	157	7.13
npm:optimize-css-assets-webpack-plugin	157	7.13
npm:@esbuild/linux-loong64	156	7.08
npm:@mrmlnc/readdir-enhanced	156	7.08
npm:babel-helper-call-delegate	156	7.08
npm:babel-helper-define-map	156	7.08
npm:babel-helper-hoist-variables	156	7.08
npm:babel-helper-optimise-call-expression	156	7.08
npm:babel-helper-replace-supers	156	7.08
npm:babel-plugin-check-es2015-constants	156	7.08
npm:babel-plugin-transform-es2015-arrow-functions	156	7.08
npm:babel-plugin-transform-es2015-classes	156	7.08
npm:babel-plugin-transform-es2015-destructuring	156	7.08
npm:babel-plugin-transform-es2015-for-of	156	7.08

npm:babel-plugin-transform-es2015-function-name	156	7.08
npm:babel-plugin-transform-es2015-parameters	156	7.08
npm:babel-plugin-transform-es2015-spread	156	7.08
npm:babel-plugin-transform-es2015-block-scoped-functions	155	7.04
npm:babel-plugin-transform-es2015-computed-properties	155	7.04
npm:babel-plugin-transform-es2015-literals	155	7.04
npm:babel-plugin-transform-es2015-object-super	155	7.04
npm:babel-plugin-transform-es2015-shorthand-properties	155	7.04
npm:babel-plugin-transform-es2015-template-literals	155	7.04
npm:pathval	155	7.04
npm:pn	155	7.04
npm:rfdc	155	7.04
npm:babel-helper-regex	154	6.99
npm:babel-plugin-transform-es2015-sticky-regex	154	6.99

npm:babel-plugin-transform-es2015-unicode-regex	154	6.99
npm:babel-plugin-transform-regenerator	154	6.99
npm:eslint-plugin-testing-library	154	6.99
npm:humanize-ms	154	6.99
npm:morgan	154	6.99
npm:promise-retry	154	6.99
npm:acorn-node	153	6.95
npm:babel-plugin-transform-es2015-duplicate-keys	153	6.95
npm:babel-plugin-transform-es2015-modules-amd	153	6.95
npm:babel-plugin-transform-es2015-modules-systemjs	153	6.95
npm:babel-plugin-transform-es2015-modules-umd	153	6.95
npm:babel-plugin-transform-es2015-typeof-symbol	153	6.95
npm:builtins	153	6.95
npm:d3-geo	153	6.95
npm:react-redux	153	6.95
npm:@npmcli/fs	152	6.90
npm:@rushstack/eslint-patch	152	6.90
npm:klaw	152	6.90

npm:lodash.defaults	152	6.90
npm:popper.js	152	6.90
npm:tree-kill	152	6.90
npm:cross-env	151	6.85
npm:d3-ease	151	6.85
npm:make-fetch-happen	151	6.85
npm:marked	151	6.85
npm:wildcard	151	6.85
npm:@babel/regjsgen	150	6.81
npm:d3-hierarchy	150	6.81
npm:karma	150	6.81
npm:long	150	6.81
npm:web-vitals	150	6.81
npm:boom	149	6.76
npm:check-error	149	6.76
npm:engine.io-client	149	6.76
npm:pnpm-webpack-plugin	149	6.76
npm:string-natural-compare	149	6.76
npm:watchpack-chokidar2	149	6.76
npm:xmlhttprequest-ssl	149	6.76
npm:@surma/rollup-plugin-off-main-thread	148	6.72
npm:dequal	148	6.72
npm:get-func-name	148	6.72
npm:isbinaryfile	148	6.72

npm:postcss-safe-parser	148	6.72
npm:base64-arraybuffer	147	6.67
npm:cryptiles	147	6.67
npm:d3-selection	147	6.67
npm:es-get-iterator	147	6.67
npm:foreach	147	6.67
npm:hawk	147	6.67
npm:lodash.assign	147	6.67
npm:sntp	147	6.67
npm:fbjs	146	6.63
npm:idb	146	6.63
npm:loader-fs-cache	146	6.63
npm:stack-trace	146	6.63
npm:time-stamp	146	6.63
npm:define-data-property	145	6.58
npm:stringstream	145	6.58
npm:stylis	145	6.58
npm:@esbuild/android-arm	144	6.54
npm:basic-auth	144	6.54
npm:d3-force	144	6.54
npm:es6-set	144	6.54
npm:grapheme-splitter	144	6.54
npm:last-call-webpack-plugin	144	6.54
npm:rx-lite	144	6.54
npm:split	144	6.54

npm:tiny-warning	144	6.54
npm:ts-pnp	144	6.54
npm:d3-quadtree	143	6.49
npm:mixin-object	143	6.49
npm:ultron	143	6.49
npm:vite	143	6.49
npm:@rollup/plugin-babel	142	6.45
npm:cli-truncate	142	6.45
npm:es6-map	142	6.45
npm:growl	142	6.45
npm:pstree.remy	142	6.45
npm:tiny-invariant	142	6.45
npm:es-array-method-boxes- properly	141	6.40
npm:foreground-child	141	6.40
npm:cheerio	140	6.35
npm:gulp	140	6.35
npm:@discoveryjs/json-ext	139	6.31
npm:d3-dsv	139	6.31
npm:d3-transition	139	6.31
npm:JSONStream	139	6.31
npm:streamsearch	139	6.31
npm:backo2	138	6.26
npm:@emotion/is-prop-valid	137	6.22
npm:append-transform	137	6.22

npm:istanbul-lib-hook	137	6.22
npm:xregexp	137	6.22
npm:archy	136	6.17
npm:default-require-extensions	136	6.17
npm:@esbuild/win32-x64	135	6.13
npm:@jest/expect-utils	135	6.13
npm:@types/testing-library__jest-dom	135	6.13
npm:d3-drag	135	6.13
npm:less	135	6.13
npm:require-uncached	135	6.13
npm:@esbuild/linux-x64	134	6.08
npm:async-foreach	134	6.08
npm:log-update	134	6.08
npm:sass-graph	134	6.08
npm:@esbuild/darwin-arm64	133	6.04
npm:array.prototype.tosorted	133	6.04
npm:dayjs	133	6.04
npm:eastasianwidth	133	6.04
npm:lodash.isstring	133	6.04
npm:validate-npm-package-name	133	6.04
npm:@esbuild/android-arm64	132	5.99
npm:@esbuild/android-x64	132	5.99
npm:@esbuild/darwin-x64	132	5.99
npm:@esbuild/freebsd-arm64	132	5.99

npm:@esbuild/freebsd-x64	132	5.99
npm:@esbuild/linux-arm	132	5.99
npm:@esbuild/linux-arm64	132	5.99
npm:@esbuild/linux-ia32	132	5.99
npm:@esbuild/linux-mips64el	132	5.99
npm:@esbuild/linux-ppc64	132	5.99
npm:@esbuild/linux-riscv64	132	5.99
npm:@esbuild/linux-s390x	132	5.99
npm:@esbuild/netbsd-x64	132	5.99
npm:@esbuild/openbsd-x64	132	5.99
npm:@esbuild/sunos-x64	132	5.99
npm:@esbuild/win32-arm64	132	5.99
npm:@esbuild/win32-ia32	132	5.99
npm:@hapi/address	132	5.99
npm:@hapi/joi	132	5.99
npm:@types/minimist	132	5.99
npm:babel-preset-es2015	132	5.99
npm:dateformat	132	5.99
npm:lodash.truncate	132	5.99
npm:prettier-linter-helpers	132	5.99
npm:reduce-css-calc	132	5.99
npm:resolve-pathname	132	5.99
npm:buffer-equal-constant-time	131	5.95
npm:ecdsa-sig-formatter	131	5.95
npm:jwa	131	5.95

npm:jws	131	5.95
npm:postcss-values-parser	131	5.95
npm:stdout-stream	131	5.95
npm:v8flags	131	5.95
npm:value-equal	131	5.95
npm:@csstools/postcss- progressive-custom-properties	130	5.90
npm:@gar/promisify	130	5.90
npm:fastest-levenshtein	130	5.90
npm:isomorphic-fetch	130	5.90
npm:mongodb	130	5.90
npm:scss-tokenizer	130	5.90
npm:split2	130	5.90
npm:karma-chrome-launcher	129	5.86
npm:lie	129	5.86
npm:postcss-selector-matches	129	5.86
npm:replace-ext	129	5.86
npm:webpack-bundle-analyzer	129	5.86
npm:@hapi/bourne	128	5.81
npm:babel-preset-env	128	5.81
npm:set-function-length	128	5.81
npm:undici-types	128	5.81
npm:@csstools/postcss-font- format-keywords	127	5.76

npm:@csstools/postcss-hwb-function	127	5.76
npm:@csstools/postcss-is-pseudo-class	127	5.76
npm:@csstools/postcss-normalize-display-values	127	5.76
npm:@csstools/selector-specificity	127	5.76
npm:babel-plugin-syntax-async-functions	127	5.76
npm:d3-zoom	127	5.76
npm:html-tags	127	5.76
npm:json-loader	127	5.76
npm:postcss-opacity-percentage	127	5.76
npm:realpath-native	127	5.76
npm:@types/trusted-types	126	5.72
npm:babel-plugin-transform-exponentiation-operator	126	5.72
npm:callsite	126	5.72
npm:config-chain	126	5.72
npm:is-redirect	126	5.72
npm:npm-package-arg	126	5.72
npm:parse5-htmlparser2-tree-adapter	126	5.72
npm:@emotion/serialize	125	5.67
npm:@emotion/utils	125	5.67

npm:babel-helper-builder-binary-assignment-operator-visitor	125	5.67
npm:babel-helper-explode-assignable-expression	125	5.67
npm:babel-helper-remap-async-to-generator	125	5.67
npm:babel-plugin-syntax-exponentiation-operator	125	5.67
npm:babel-plugin-transform-async-to-generator	125	5.67
npm:buffer-alloc	125	5.67
npm:buffer-alloc-unsafe	125	5.67
npm:buffer-fill	125	5.67
npm:husky	125	5.67
npm:is-property	125	5.67
npm:proto-list	125	5.67
npm:@csstools/convert-colors	124	5.63
npm:@csstools/postcss-color-function	124	5.63
npm:@csstools/postcss-ic-unit	124	5.63
npm:@csstools/postcss-oklab-function	124	5.63
npm:has-cors	124	5.63
npm:launch-editor	124	5.63
npm:lodash-es	124	5.63

npm:postcss-color-gray	124	5.63
npm:postcss-color-mod-function	124	5.63
npm:tar-fs	124	5.63
npm:true-case-path	124	5.63
npm:unzip-response	124	5.63
npm:yeast	124	5.63
npm:@emotion/cache	123	5.58
npm:@emotion/sheet	123	5.58
npm:@emotion/weak-memoize	123	5.58
npm:busboy	123	5.58
npm:create-require	123	5.58
npm:d3-axis	123	5.58
npm:d3-brush	123	5.58
npm:escope	123	5.58
npm:highlight.js	123	5.58
npm:is-relative	123	5.58
npm:mdurl	123	5.58
npm:natural-compare-lite	123	5.58
npm:parseqs	123	5.58
npm:parseuri	123	5.58
npm:safe-array-concat	123	5.58
npm:is-lambda	122	5.54
npm:joi	122	5.54
npm:lodash.isequal	122	5.54
npm:minipass-fetch	122	5.54

npm:minipass-sized	122	5.54
npm:postcss-clamp	122	5.54
npm:is-absolute	121	5.49
npm:left-pad	121	5.49
npm:lodash.isarguments	121	5.49
npm:lodash.once	121	5.49
npm:vue-template-compiler	121	5.49
npm:blob	120	5.45
npm:capture-stack-trace	120	5.45
npm:create-error-class	120	5.45
npm:event-target-shim	120	5.45
npm:graceful-readlink	120	5.45
npm:to-readable-stream	120	5.45
npm:@types/color-name	119	5.40
npm:@types/cookie	119	5.40
npm:after	119	5.40
npm:arraybuffer.slice	119	5.40
npm:component-bind	119	5.40
npm:component-inherit	119	5.40
npm:generate-function	119	5.40
npm:grunt-contrib-uglify	119	5.40
npm:p-reduce	119	5.40
npm:to-array	119	5.40
npm:array-slice	118	5.36
npm:decamelize-keys	118	5.36

npm:ent	118	5.36
npm:lodash.flatten	118	5.36
npm:minimist-options	118	5.36
npm:workbox-recipes	118	5.36
npm:@apideck/better-ajv-errors	117	5.31
npm:@colors/colors	117	5.31
npm:cookie-parser	117	5.31
npm:jsonc-parser	117	5.31
npm:jszip	117	5.31
npm:lodash.keys	117	5.31
npm:merge-source-map	117	5.31
npm:p-timeout	117	5.31
npm:string.prototype.trimleft	117	5.31
npm:string.prototype.trimright	117	5.31
npm:@types/source-list-map	116	5.27
npm:@types/webpack-sources	116	5.27
npm:abort-controller	116	5.27
npm:cross-fetch	116	5.27
npm:d3-chord	116	5.27
npm:d3-collection	116	5.27
npm:d3-polygon	116	5.27
npm:d3-random	116	5.27
npm:d3-scale-chromatic	116	5.27
npm:mamacro	116	5.27
npm:shallowequal	116	5.27

npm:arch	115	5.22
npm:array-filter	115	5.22
npm:is-unc-path	115	5.22
npm:lazystream	115	5.22
npm:unc-path-regex	115	5.22
npm:bson	114	5.17
npm:cli-table3	114	5.17
npm:yargs-unparser	114	5.17
npm:@types/debug	113	5.13
npm:array.prototype.reduce	113	5.13
npm:arraybuffer.prototype.slice	113	5.13
npm:microevent.ts	113	5.13
npm:typed-array-buffer	113	5.13
npm:typed-array-byte-length	113	5.13
npm:typed-array-byte-offset	113	5.13
npm:worker-rpc	113	5.13
npm:eslint-config-airbnb-base	112	5.08
npm:jsonwebtoken	112	5.08
npm:lodash._getnative	112	5.08
npm:user-home	112	5.08
npm:consolidate	111	5.04
npm:hash-sum	111	5.04
npm:postcss-discard-unused	111	5.04
npm:postcss-merge-idents	111	5.04
npm:postcss-reduce-idents	111	5.04

npm:postcss-zindex	111	5.04
npm:@babel/plugin-transform-optional-chaining	110	4.99
npm:@types/tapable	110	4.99
npm:es6-promisify	110	4.99
npm:in-publish	110	4.99
npm:jasmine-core	110	4.99
npm:lodash.isarray	110	4.99
npm:@emotion/react	109	4.95
npm:@types/unist	109	4.95
npm:grunt-contrib-jshint	109	4.95
npm:lodash.mergeWith	109	4.95
npm:math-expression-evaluator	109	4.95
npm:reduce-function-call	109	4.95
npm:uglifyjs-webpack-plugin	109	4.95
npm:@babel/plugin-syntax-import-attributes	108	4.90
npm:@ungap/structured-clone	108	4.90
npm:karma-jasmine	108	4.90
npm:set-function-name	108	4.90
npm:simple-concat	108	4.90
npm:vue-loader	108	4.90
npm:vue-router	108	4.90
npm:vue-style-loader	108	4.90

npm:@babel/plugin-syntax- unicode-sets-regex	107	4.86
npm:@babel/plugin-transform- async-generator-functions	107	4.86
npm:@babel/plugin-transform- class-properties	107	4.86
npm:@babel/plugin-transform- class-static-block	107	4.86
npm:@babel/plugin-transform- dynamic-import	107	4.86
npm:@babel/plugin-transform- export-namespace-from	107	4.86
npm:@babel/plugin-transform- json-strings	107	4.86
npm:@babel/plugin-transform- logical-assignment-operators	107	4.86
npm:@babel/plugin-transform- nullish-coalescing-operator	107	4.86
npm:@babel/plugin-transform- numeric-separator	107	4.86
npm:@babel/plugin-transform- object-rest-spread	107	4.86
npm:@babel/plugin-transform- optional-catch-binding	107	4.86

npm:@babel/plugin-transform-private-methods	107	4.86
npm:@babel/plugin-transform-private-property-in-object	107	4.86
npm:@babel/plugin-transform-unicode-property-regex	107	4.86
npm:@babel/plugin-transform-unicode-sets-regex	107	4.86
npm:@csstools/postcss-stepped-value-functions	107	4.86
npm:@csstools/postcss-unset-value	107	4.86
npm:better-assert	107	4.86
npm:node-gyp-build	107	4.86
npm:unist-util-stringify-position	107	4.86
npm:@babel/helper-call-delegate	106	4.81
npm:caniuse-db	106	4.81
npm:extract-text-webpack-plugin	106	4.81
npm:generate-object-property	106	4.81
npm:log4js	106	4.81
npm:array-differ	105	4.77
npm:babel-preset-react	105	4.77
npm:bignumber.js	105	4.77
npm:hard-rejection	105	4.77
npm:has-yarn	105	4.77
npm:js-sdsl	105	4.77

npm:object-component	105	4.77
npm:big-integer	104	4.72
npm:character-entities	104	4.72
npm:path-scurry	104	4.72
npm:rework	104	4.72
npm:rework-visit	104	4.72
npm:@types/cors	103	4.68
npm:@types/webpack	103	4.68
npm:clap	103	4.68
npm:global	103	4.68
npm:internmap	103	4.68
npm:mkdirp-classic	103	4.68
npm:mongoose	103	4.68
npm:npm-pick-manifest	103	4.68
npm:react-bootstrap	103	4.68
npm:sinon	103	4.68
npm:traverse	103	4.68
npm:whet.extend	103	4.68
npm:@nicolo-ribaudo/eslint-scope-5-internals	102	4.63
npm:data-uri-to-buffer	102	4.63
npm:is-yarn-global	102	4.63
npm:min-document	102	4.63
npm:postcss-message-helpers	102	4.63
npm:pretty-hrtime	102	4.63

npm:vfile	102	4.63
npm:@csstools/postcss-cascade-layers	101	4.58
npm:babel-extract-comments	101	4.58
npm:dom-walk	101	4.58
npm:escape-goat	101	4.58
npm:js-levenshtein	101	4.58
npm:less-loader	101	4.58
npm:parse-node-version	101	4.58
npm:react-fast-compare	101	4.58
npm:redux-thunk	101	4.58
npm:ts-jest	101	4.58
npm:vue-template-es2015-compiler	101	4.58
npm:@babel/helper-builder-react-jsx	100	4.54
npm:@types/uglify-js	100	4.54
npm:fileset	100	4.54
npm:jiti	100	4.54
npm:semver-compare	100	4.54
npm:stop-iteration-iterator	100	4.54
npm:vfile-message	100	4.54
npm:vue-hot-reload-api	100	4.54
npm:babel-polyfill	99	4.49
npm:bail	99	4.49
npm:d3-contour	99	4.49

npm:istanbul-api	99	4.49
npm:mri	99	4.49
npm:postcss-filter-plugins	99	4.49
npm:trough	99	4.49
npm:tslint	99	4.49
npm:@csstools/postcss-trigonometric-functions	98	4.45
npm:@protobufjs/utf8	98	4.45
npm:@types/events	98	4.45
npm:babel-plugin-transform-class-properties	98	4.45
npm:buffer-equal	98	4.45
npm:colormin	98	4.45
npm:eslint-plugin-promise	98	4.45
npm:hyphenate-style-name	98	4.45
npm:merge	98	4.45
npm:pupa	98	4.45
npm:rx-lite-aggregates	98	4.45
npm:unified	98	4.45
npm:@cspotcode/source-map-support	97	4.40
npm:@protobufjs/aspromise	97	4.40
npm:@protobufjs/base64	97	4.40
npm:@protobufjs/codegen	97	4.40
npm:@protobufjs/eventemitter	97	4.40

npm:@protobufjs/fetch	97	4.40
npm:@protobufjs/float	97	4.40
npm:@protobufjs/inquire	97	4.40
npm:@protobufjs/path	97	4.40
npm:@protobufjs/pool	97	4.40
npm:@tsconfig/node10	97	4.40
npm:@tsconfig/node12	97	4.40
npm:@tsconfig/node14	97	4.40
npm:@tsconfig/node16	97	4.40
npm:custom-event	97	4.40
npm:d3-fetch	97	4.40
npm:is-my-json-valid	97	4.40
npm:path-root	97	4.40
npm:path-root-regex	97	4.40
npm:memoize-one	96	4.36
npm:pacote	96	4.36
npm:resize-observer-polyfill	96	4.36
npm:@emotion/babel-plugin	95	4.31
npm:@types/yauzl	95	4.31
npm:babel-plugin-syntax-class-properties	95	4.31
npm:cookiejar	95	4.31
npm:eslint-plugin-vue	95	4.31
npm:formidable	95	4.31
npm:lint-staged	95	4.31

npm:npm-registry-fetch	95	4.31
npm:vinyl	95	4.31
npm:clone-stats	94	4.27
npm:di	94	4.27
npm:dom-serialize	94	4.27
npm:grunt-contrib-watch	94	4.27
npm:reflect-metadata	94	4.27
npm:@emotion/styled	93	4.22
npm:@types/send	93	4.22
npm:markdown-it	93	4.22
npm:math-random	93	4.22
npm:unist-util-is	93	4.22
npm:arity-n	92	4.18
npm:array-map	92	4.18
npm:array-reduce	92	4.18
npm:bs-logger	92	4.18
npm:compose-function	92	4.18
npm:de-indent	92	4.18
npm:fast-safe-stringify	92	4.18
npm:lodash._basecopy	92	4.18
npm:parse-filepath	92	4.18
npm:sigmund	92	4.18
npm:workerpool	92	4.18
npm:@types/ms	91	4.13
npm:express-session	91	4.13

npm:is-alphabetical	91	4.13
npm:is-alphanumeric	91	4.13
npm:is-decimal	91	4.13
npm:litr2	91	4.13
npm:lodash.isboolean	91	4.13
npm:simple-get	91	4.13
npm:@adobe/css-tools	90	4.09
npm:d3-voronoi	90	4.09
npm:read-package-json	90	4.09
npm:unist-util-visit	90	4.09
npm:unist-util-visit-parents	90	4.09
npm:@jest/expect	89	4.04
npm:@pkgjs/parseargs	89	4.04
npm:character-entities-legacy	89	4.04
npm:filename-reserved-regex	89	4.04
npm:filenamify	89	4.04
npm:jackspeak	89	4.04
npm:lodash.isiterateecall	89	4.04
npm:qjobs	89	4.04
npm:remark-parse	89	4.04
npm:reselect	89	4.04
npm:strip-outer	89	4.04
npm:trim-repeated	89	4.04
npm:v8-compile-cache-lib	89	4.04
npm:winston	89	4.04

npm:@isaacs/cliui	88	3.99
npm:@types/long	88	3.99
npm:es6-error	88	3.99
npm:is-reference	88	3.99
npm:linkify-it	88	3.99
npm:protobufjs	88	3.99
npm:superagent	88	3.99
npm:svg-tags	88	3.99
npm:uc.micro	88	3.99
npm:untildify	88	3.99
npm:@csstools/postcss-nested-calc	87	3.95
npm:@csstools/postcss-text-decoration-shorthand	87	3.95
npm:@types/hoist-non-react-statics	87	3.95
npm:babel-plugin-transform-flow-strip-types	87	3.95
npm:chart.js	87	3.95
npm:get-port	87	3.95
npm:has-binary2	87	3.95
npm:levenary	87	3.95
npm:object-path	87	3.95
npm:read	87	3.95
npm:serialize-error	87	3.95
npm:@webpack-cli/serve	86	3.90

npm:babel-plugin-syntax-dynamic-import	86	3.90
npm:character-reference-invalid	86	3.90
npm:eslint-config-airbnb	86	3.90
npm:is-hexadecimal	86	3.90
npm:npm-install-checks	86	3.90
npm:npm-run-all	86	3.90
npm:parse-entities	86	3.90
npm:raw-loader	86	3.90
npm:@webpack-cli/info	85	3.86
npm:date-format	85	3.86
npm:lodash.isnumber	85	3.86
npm:pidtree	85	3.86
npm:string.prototype.padend	85	3.86
npm:@material-ui/core	84	3.81
npm:babel-plugin-syntax-flow	84	3.81
npm:eslint-config-standard	84	3.81
npm:fs-readdir-recursive	84	3.81
npm:lodash.includes	84	3.81
npm:map-stream	84	3.81
npm:streamroller	84	3.81
npm:sucrase	84	3.81
npm:ts-interface-checker	84	3.81
npm:@types/mdast	83	3.77
npm:copy-to-clipboard	83	3.77

npm:css-unit-converter	83	3.77
npm:lodash.kebabcase	83	3.77
npm:memory-pager	83	3.77
npm:recast	83	3.77
npm:rollup-pluginutils	83	3.77
npm:sparse-bitfield	83	3.77
npm:toggle-selection	83	3.77
npm:ts-loader	83	3.77
npm:zone.js	83	3.77
npm:@angular/common	82	3.72
npm:@angular/compiler	82	3.72
npm:@babel/register	82	3.72
npm:@types/http-errors	82	3.72
npm:@webpack-cli/configtest	82	3.72
npm:adm-zip	82	3.72
npm:babel-plugin-transform-react- jsx	82	3.72
npm:babel-plugin-transform- runtime	82	3.72
npm:charenc	82	3.72
npm:crypt	82	3.72
npm:flagged-respawn	82	3.72
npm:glob-stream	82	3.72
npm:lodash.isinteger	82	3.72
npm:mdast-util-to-string	82	3.72

npm:ordered-read-streams	82	3.72
npm:space-separated-tokens	82	3.72
npm:unique-stream	82	3.72
npm:url-join	82	3.72
npm:@angular/core	81	3.68
npm:babel-helper-builder-react-jsx	81	3.68
npm:exit-hook	81	3.68
npm:fancy-log	81	3.68
npm:@angular/forms	80	3.63
npm:@angular/platform-browser	80	3.63
npm:@types/jasmine	80	3.63
npm:@vue/component-compiler- utils	80	3.63
npm:concurrently	80	3.63
npm:denque	80	3.63
npm:fined	80	3.63
npm:grunt-contrib-concat	80	3.63
npm:react-test-renderer	80	3.63
npm:@angular/platform-browser- dynamic	79	3.59
npm:array-each	79	3.59
npm:babel-plugin-transform-react- display-name	79	3.59
npm:comma-separated-tokens	79	3.59
npm:lodash.escape	79	3.59

npm:lodash.tail	79	3.59
npm:object.defaults	79	3.59
npm:property-information	79	3.59
npm:read-package-json-fast	79	3.59
npm:uid-number	79	3.59
npm:watch	79	3.59
npm:@sideway/address	78	3.54
npm:@sideway/formula	78	3.54
npm:@sideway/pinpoint	78	3.54
npm:@yarnpkg/lockfile	78	3.54
npm:browserify	78	3.54
npm:clipboardy	78	3.54
npm:compare-versions	78	3.54
npm:eslint-plugin-node	78	3.54
npm:lodash.restparam	78	3.54
npm:lodash.throttle	78	3.54
npm:merge-deep	78	3.54
npm:temp	78	3.54
npm:validator	78	3.54
npm:@angular/compiler-cli	77	3.50
npm:dependency-graph	77	3.50
npm:slide	77	3.50
npm:throttleit	77	3.50
npm:@types/gejson	76	3.45
npm:camelize	76	3.45

npm:is-object	76	3.45
npm:moment-timezone	76	3.45
npm:mpath	76	3.45
npm:stream-combiner	76	3.45
npm:@angular/router	75	3.40
npm:ansi-wrap	75	3.40
npm:babel-plugin-transform-react- jsx-self	75	3.40
npm:babel-plugin-transform-react- jsx-source	75	3.40
npm:ccount	75	3.40
npm:gulp-rename	75	3.40
npm:mini-create-react-context	75	3.40
npm:string-argv	75	3.40
npm:vue-eslint-parser	75	3.40
npm:@npmcli/git	74	3.36
npm:@npmcli/promise-spawn	74	3.36
npm:@types/http-cache-semantics	74	3.36
npm:cacheable-lookup	74	3.36
npm:css-parse	74	3.36
npm:event-stream	74	3.36
npm:file-type	74	3.36
npm:gh-pages	74	3.36
npm:graphql	74	3.36
npm:kareem	74	3.36

npm:loupe	74	3.36
npm:make-iterator	74	3.36
npm:mquery	74	3.36
npm:@angular/cli	73	3.31
npm:@swc/helpers	73	3.31
npm:dezalgo	73	3.31
npm:eventemitter2	73	3.31
npm:lodash.flattendeep	73	3.31
npm:memorystream	73	3.31
npm:object.map	73	3.31
npm:prismjs	73	3.31
npm:vinyl-fs	73	3.31
npm:@npmcli/installed-package-contents	72	3.27
npm:@npmcli/run-script	72	3.27
npm:@types/eslint-visitor-keys	72	3.27
npm:babel-preset-flow	72	3.27
npm:clone-buffer	72	3.27
npm:consola	72	3.27
npm:es-errors	72	3.27
npm:font-awesome	72	3.27
npm:glogg	72	3.27
npm:js-cookie	72	3.27
npm:liftoff	72	3.27
npm:md5	72	3.27

npm:minipass-json-stream	72	3.27
npm:node-emoji	72	3.27
npm:random-bytes	72	3.27
npm:sade	72	3.27
npm:sparkles	72	3.27
npm:@emotion/stylis	71	3.22
npm:@emotion/use-insertion- effect-with-fallbacks	71	3.22
npm:@npmcli/node-gyp	71	3.22
npm:@types/tough-cookie	71	3.22
npm:array.prototype.findlastindex	71	3.22
npm:crc-32	71	3.22
npm:esbuild-darwin-64	71	3.22
npm:esbuild-linux-64	71	3.22
npm:esbuild-windows-64	71	3.22
npm:grunt-cli	71	3.22
npm:gulplog	71	3.22
npm:http2-wrapper	71	3.22
npm:object.groupby	71	3.22
npm:proc-log	71	3.22
npm:resolve-alpn	71	3.22
npm:simple-update-notifier	71	3.22
npm:uncontrollable	71	3.22
npm:@angular/animations	70	3.18
npm:@polka/url	70	3.18

npm:arc	70	3.18
npm:electron	70	3.18
npm:esbuild-android-arm64	70	3.18
npm:esbuild-darwin-arm64	70	3.18
npm:esbuild-freebsd-64	70	3.18
npm:esbuild-freebsd-arm64	70	3.18
npm:esbuild-linux-32	70	3.18
npm:esbuild-linux-arm	70	3.18
npm:esbuild-linux-arm64	70	3.18
npm:esbuild-linux-mips64le	70	3.18
npm:esbuild-linux-ppc64le	70	3.18
npm:esbuild-netbsd-64	70	3.18
npm:esbuild-openbsd-64	70	3.18
npm:esbuild-sunos-64	70	3.18
npm:esbuild-windows-32	70	3.18
npm:esbuild-windows-arm64	70	3.18
npm:is-what	70	3.18
npm:karma-jasmine-html-reporter	70	3.18
npm:macaddress	70	3.18
npm:node-abi	70	3.18
npm:sirv	70	3.18
npm:styled-components	70	3.18
npm:through2-filter	70	3.18
npm:uid-safe	70	3.18
npm:@types/hast	69	3.13

npm:@types/keyv	69	3.13
npm:@types/responselike	69	3.13
npm:@types/sizzle	69	3.13
npm:hasha	69	3.13
npm:jasmine	69	3.13
npm:mrmime	69	3.13
npm:pause-stream	69	3.13
npm:pure-rand	69	3.13
npm:saslprep	69	3.13
npm:selenium-webdriver	69	3.13
npm:unbzip2-stream	69	3.13
npm:uniqid	69	3.13
npm:when	69	3.13
npm:@remix-run/router	68	3.09
npm:ansi-gray	68	3.09
npm:cloneable-readable	68	3.09
npm:from	68	3.09
npm:gulp-util	68	3.09
npm:javascript-stringify	68	3.09
npm:prop-types-extra	68	3.09
npm:stylus	68	3.09
npm:@socket.io/component-emitter	67	3.04
npm:@vue/cli-service	67	3.04
npm:cli-table	67	3.04

npm:is-async-function	67	3.04
npm:is-finalizationregistry	67	3.04
npm:is-in-browser	67	3.04
npm:license-webpack-plugin	67	3.04
npm:lodash.mapvalues	67	3.04
npm:multimatch	67	3.04
npm:postcss-load-options	67	3.04
npm:postcss-load-plugins	67	3.04
npm:react-popover	67	3.04
npm:reflect.getprototypeof	67	3.04
npm:split-on-first	67	3.04
npm:unist-util-position	67	3.04
npm:which-builtin-type	67	3.04
npm:@angular-devkit/core	66	3.00
npm:@vue/compiler-sfc	66	3.00
npm:es-iterator-helpers	66	3.00
npm:iterator.prototype	66	3.00
npm:karma-coverage	66	3.00
npm:mdast-util-from-markdown	66	3.00
npm:micromark	66	3.00
npm:mitt	66	3.00
npm:multer	66	3.00
npm:passport	66	3.00
npm:plist	66	3.00
npm:to-absolute-glob	66	3.00

npm:totalist	66	3.00
npm:zwitch	66	3.00
npm:@angular-devkit/schematics	65	2.95
npm:@types/chai	65	2.95
npm:@types/node-fetch	65	2.95
npm:archiver	65	2.95
npm:lodash._baseassign	65	2.95
npm:lodash._bindcallback	65	2.95
npm:lodash.pick	65	2.95
npm:readline2	65	2.95
npm:stream-combiner2	65	2.95
npm:vlq	65	2.95
npm:xmlhttprequest	65	2.95
npm:babel-cli	64	2.91
npm:command-exists	64	2.91
npm:debounce	64	2.91
npm:eslint-plugin-standard	64	2.91
npm:llex	64	2.91
npm:node-dir	64	2.91
npm:protocol-buffers-schema	64	2.91
npm:tinycolor2	64	2.91
npm:vfile-location	64	2.91
npm:vuex	64	2.91
npm:@alloc/quick-lru	63	2.86
npm:@material-ui/icons	63	2.86

npm:@types/cacheable-request	63	2.86
npm:@types/fs-extra	63	2.86
npm:ansicolors	63	2.86
npm:babel-preset-stage-2	63	2.86
npm:compress-commons	63	2.86
npm:copy-anything	63	2.86
npm:crc32-stream	63	2.86
npm:mdast-util-to-hast	63	2.86
npm:require_optional	63	2.86
npm:style-to-object	63	2.86
npm:uglify-es	63	2.86
npm:use-sync-external-store	63	2.86
npm:zip-stream	63	2.86
npm:css-color-keywords	62	2.81
npm:css-to-react-native	62	2.81
npm:es-define-property	62	2.81
npm:expand-template	62	2.81
npm:fast-xml-parser	62	2.81
npm:fstream-ignore	62	2.81
npm:github-from-package	62	2.81
npm:inline-style-parser	62	2.81
npm:karma-source-map-support	62	2.81
npm:mustache	62	2.81
npm:netmask	62	2.81
npm:npm	62	2.81

npm:puppeteer	62	2.81
npm:sift	62	2.81
npm:std-env	62	2.81
npm:unist-util-remove-position	62	2.81
npm:@ngtools/webpack	61	2.77
npm:@schematics/angular	61	2.77
npm:browser	61	2.77
npm:cache-loader	61	2.77
npm:delaunator	61	2.77
npm:gl-matrix	61	2.77
npm:grunt-contrib-cssmin	61	2.77
npm:gud	61	2.77
npm:jest-environment-jsdom- fourteen	61	2.77
npm:jss	61	2.77
npm:lodash.defaultsdeep	61	2.77
npm:please-upgrade-node	61	2.77
npm:prebuild-install	61	2.77
npm:regexp-tree	61	2.77
npm:rx	61	2.77
npm:spawn-command	61	2.77
npm:styled-jsx	61	2.77
npm:web-streams-polyfill	61	2.77
npm:webpack-chain	61	2.77

npm:@angular-devkit/build-angular	60	2.72
npm:@mui/material	60	2.72
npm:@types/mocha	60	2.72
npm:character-parser	60	2.72
npm:constantinople	60	2.72
npm:css-vendor	60	2.72
npm:is-function	60	2.72
npm:lodash._root	60	2.72
npm:lodash.map	60	2.72
npm:next	60	2.72
npm:stylus-loader	60	2.72
npm:thread-loader	60	2.72
npm:with	60	2.72
npm:@floating-ui/core	59	2.68
npm:@floating-ui/dom	59	2.68
npm:@vue/shared	59	2.68
npm:before-after-hook	59	2.68
npm:create-react-class	59	2.68
npm:degenerator	59	2.68
npm:get-uri	59	2.68
npm:glob-watcher	59	2.68
npm:grunt-contrib-qunit	59	2.68
npm:hast-util-parse-selector	59	2.68
npm:is-valid-glob	59	2.68

npm:istanbul	59	2.68
npm:lodash.foreach	59	2.68
npm:pac-proxy-agent	59	2.68
npm:pac-resolver	59	2.68
npm:protractor	59	2.68
npm:static-eval	59	2.68
npm:strip-bom-string	59	2.68
npm:three	59	2.68
npm:@types/react-router-dom	58	2.63
npm:append-field	58	2.63
npm:archiver-utils	58	2.63
npm:aws-sdk	58	2.63
npm:babel-plugin-syntax-decorators	58	2.63
npm:compute-scroll-into-view	58	2.63
npm:d3-delaunay	58	2.63
npm:dargs	58	2.63
npm:errorhandler	58	2.63
npm:esbuild-linux-s390x	58	2.63
npm:filter-obj	58	2.63
npm:grunt-contrib-copy	58	2.63
npm:gulp-uglify	58	2.63
npm:hastscript	58	2.63
npm:js-beautify	58	2.63
npm:jstransformer	58	2.63

npm:longest-streak	58	2.63
npm:lru-queue	58	2.63
npm:memoizee	58	2.63
npm:options	58	2.63
npm:proxy-agent	58	2.63
npm:request-progress	58	2.63
npm:supertest	58	2.63
npm:timers-ext	58	2.63
npm:to-buffer	58	2.63
npm:webpack-subresource-integrity	58	2.63
npm:@babel/cli	57	2.59
npm:@octokit/endpoint	57	2.59
npm:@octokit/request	57	2.59
npm:@octokit/request-error	57	2.59
npm:@restart/hooks	57	2.59
npm:@vue/cli-plugin-babel	57	2.59
npm:@vue/cli-plugin-eslint	57	2.59
npm:@vue/compiler-core	57	2.59
npm:@vue/compiler-dom	57	2.59
npm:asynciterator.prototype	57	2.59
npm:columnify	57	2.59
npm:earcut	57	2.59
npm:editorconfig	57	2.59
npm:eslint-plugin-es	57	2.59
npm:nwmatcher	57	2.59

npm:pause	57	2.59
npm:pbf	57	2.59
npm:promise-polyfill	57	2.59
npm:quickselect	57	2.59
npm:ramda	57	2.59
npm:react-select	57	2.59
npm:serve-favicon	57	2.59
npm:tar-pack	57	2.59
npm:universal-user-agent	57	2.59
npm:useragent	57	2.59
npm:xmlDOM	57	2.59
npm:@grpc/proto-loader	56	2.54
npm:@types/warning	56	2.54
npm:@vue/compiler-ssr	56	2.54
npm:array-ify	56	2.54
npm:babel-plugin-styled-components	56	2.54
npm:compare-func	56	2.54
npm:dicer	56	2.54
npm:grunt-contrib-clean	56	2.54
npm:mdast-util-definitions	56	2.54
npm:nodemailer	56	2.54
npm:pngjs	56	2.54
npm:resolve-protobuf-schema	56	2.54
npm:ts-api-utils	56	2.54

npm:@mapbox/node-pre-gyp	55	2.50
npm:@mui/system	55	2.50
npm:@next/env	55	2.50
npm:@octokit/types	55	2.50
npm:conventional-changelog- angular	55	2.50
npm:conventional-commits-parser	55	2.50
npm:deprecation	55	2.50
npm:easy-stack	55	2.50
npm:enzyme	55	2.50
npm:event-pubsub	55	2.50
npm:is-negated-glob	55	2.50
npm:is-text-path	55	2.50
npm:js-message	55	2.50
npm:seedrandom	55	2.50
npm:text-extensions	55	2.50
npm:unist-util-generated	55	2.50
npm:@fortawesome/fontawesome- free	54	2.45
npm:@mui/private-theming	54	2.45
npm:@mui/styled-engine	54	2.45
npm:@mui/types	54	2.45
npm:@mui/utils	54	2.45
npm:@types/d3-scale	54	2.45
npm:@types/d3-time	54	2.45

npm:@types/history	54	2.45
npm:@types/react-redux	54	2.45
npm:@vue/babel-helper-vue-jsx-merge-props	54	2.45
npm:@vue/babel-plugin-transform-vue-jsx	54	2.45
npm:@vue/babel-preset-jsx	54	2.45
npm:@vue/babel-sugar-functional-vue	54	2.45
npm:@vue/babel-sugar-inject-h	54	2.45
npm:@vue/babel-sugar-v-model	54	2.45
npm:@vue/babel-sugar-v-on	54	2.45
npm:babel-plugin-syntax-async-generators	54	2.45
npm:bplist-parser	54	2.45
npm:exenv	54	2.45
npm:fecha	54	2.45
npm:gulp-sourcemaps	54	2.45
npm:jasmine-spec-reporter	54	2.45
npm:jshint	54	2.45
npm:launch-editor-middleware	54	2.45
npm:matcher	54	2.45
npm:nyc	54	2.45
npm:passport-strategy	54	2.45
npm:react-icons	54	2.45

npm:robust-predicates	54	2.45
npm:trim	54	2.45
npm:@angular-devkit/architect	53	2.41
npm:@babel/plugin-bugfix-v8- static-class-fields-readonly	53	2.41
npm:@fortawesome/fontawesome- svg-core	53	2.41
npm:@grpc/grpc-js	53	2.41
npm:@next/swc-darwin-arm64	53	2.41
npm:@next/swc-darwin-x64	53	2.41
npm:@next/swc-linux-arm64-gnu	53	2.41
npm:@next/swc-linux-arm64-musl	53	2.41
npm:@next/swc-linux-x64-gnu	53	2.41
npm:@next/swc-linux-x64-musl	53	2.41
npm:@next/swc-win32-arm64- msvc	53	2.41
npm:@next/swc-win32-ia32-msvc	53	2.41
npm:@next/swc-win32-x64-msvc	53	2.41
npm:babel-preset-stage-3	53	2.41
npm:concat-with-sourcemaps	53	2.41
npm:is-url	53	2.41
npm:just-extend	53	2.41
npm:leaflet	53	2.41
npm:lodash.cond	53	2.41
npm:ncp	53	2.41

npm:nise	53	2.41
npm:rollup-plugin-babel	53	2.41
npm:sliced	53	2.41
npm:@mui/base	52	2.36
npm:@octokit/auth-token	52	2.36
npm:@rollup/plugin-commonjs	52	2.36
npm:@types/jquery	52	2.36
npm:@types/react-router	52	2.36
npm:@vitejs/plugin-react	52	2.36
npm:@vue/reactivity	52	2.36
npm:cardinal	52	2.36
npm:cheerio-select	52	2.36
npm:cli-highlight	52	2.36
npm:colorspace	52	2.36
npm:debuglog	52	2.36
npm:enabled	52	2.36
npm:esbuild-linux-riscv64	52	2.36
npm:executable	52	2.36
npm:file-saver	52	2.36
npm:kuler	52	2.36
npm:load-grunt-tasks	52	2.36
npm:markdown-table	52	2.36
npm:ncname	52	2.36
npm:nullthrows	52	2.36
npm:p-queue	52	2.36

npm:path	52	2.36
npm:pkg-conf	52	2.36
npm:plugin-error	52	2.36
npm:redyed	52	2.36
npm:strnum	52	2.36
npm:text-hex	52	2.36
npm:tunnel	52	2.36
npm:undici	52	2.36
npm:web-namespaces	52	2.36
npm:xml-char-classes	52	2.36
npm:@angular-devkit/build- webpack	51	2.32
npm:@babel/helper-builder-react- jsx-experimental	51	2.32
npm:@fortawesome/free-solid-svg- icons	51	2.32
npm:@sinonjs/samsam	51	2.32
npm:@soda/friendly-errors- webpack-plugin	51	2.32
npm:@vue/cli-overlay	51	2.32
npm:@vue/cli-shared-utils	51	2.32
npm:@vue/web-component- wrapper	51	2.32
npm:babel-helper-bindify- decorators	51	2.32

npm:babel-helper-explode-class	51	2.32
npm:babel-plugin-transform-async-generator-functions	51	2.32
npm:babel-plugin-transform-decorators	51	2.32
npm:cachedir	51	2.32
npm:has-gulplog	51	2.32
npm:hast-util-from-parse5	51	2.32
npm:is-nan	51	2.32
npm:jss-plugin-camel-case	51	2.32
npm:jss-plugin-default-unit	51	2.32
npm:jss-plugin-global	51	2.32
npm:jss-plugin-nested	51	2.32
npm:karma-coverage-istanbul-reporter	51	2.32
npm:micromark-util-character	51	2.32
npm:micromark-util-symbol	51	2.32
npm:micromark-util-types	51	2.32
npm:napi-build-utils	51	2.32
npm:p-event	51	2.32
npm:readdir-scoped-modules	51	2.32
npm:string-width-cjs	51	2.32
npm:strip-ansi-cjs	51	2.32
npm:sumchecker	51	2.32
npm:wrap-ansi-cjs	51	2.32

npm:@material-ui/system	50	2.27
npm:@material-ui/utils	50	2.27
npm:@vue/runtime-core	50	2.27
npm:@vue/runtime-dom	50	2.27
npm:buffer-shims	50	2.27
npm:detect-port	50	2.27
npm:fbjs-css-vars	50	2.27
npm:get-tsconfig	50	2.27
npm:graphql-tag	50	2.27
npm:gulp-sass	50	2.27
npm:init-package-json	50	2.27
npm:is-odd	50	2.27
npm:jss-plugin-props-sort	50	2.27
npm:jss-plugin-rule-value-function	50	2.27
npm:jss-plugin-vendor-prefixer	50	2.27
npm:lodash.union	50	2.27
npm:micromark-factory-space	50	2.27
npm:micromark-util-decode-numeric-character-reference	50	2.27
npm:micromark-util-decode-string	50	2.27
npm:micromark-util-encode	50	2.27
npm:micromark-util-sanitize-uri	50	2.27
npm:promzard	50	2.27
npm:stacktrace-parser	50	2.27
npm:subarg	50	2.27

npm:vinyl-sourcemaps-apply	50	2.27
npm:@material-ui/styles	49	2.22
npm:@octokit/openapi-types	49	2.22
npm:@octokit/plugin-paginate-rest	49	2.22
npm:@types/anymatch	49	2.22
npm:base64url	49	2.22
npm:decode-named-character-reference	49	2.22
npm:first-chunk-stream	49	2.22
npm:fs	49	2.22
npm:jmespath	49	2.22
npm:mdast-util-to-markdown	49	2.22
npm:micromark-core-commonmark	49	2.22
npm:micromark-factory-destination	49	2.22
npm:micromark-factory-label	49	2.22
npm:micromark-factory-title	49	2.22
npm:micromark-factory-whitespace	49	2.22
npm:micromark-util-chunked	49	2.22
npm:micromark-util-classify-character	49	2.22
npm:micromark-util-combine-extensions	49	2.22
npm:micromark-util-html-tag-name	49	2.22

npm:micromark-util-normalize-identifier	49	2.22
npm:micromark-util-resolve-all	49	2.22
npm:micromark-util-subtokenize	49	2.22
npm:read-cmd-shim	49	2.22
npm:safe-stable-stringify	49	2.22
npm:snake-case	49	2.22
npm:tape	49	2.22
npm:throttle-debounce	49	2.22
npm:@babel/plugin-proposal-export-default-from	48	2.18
npm:@mapbox/point-geometry	48	2.18
npm:@mui/icons-material	48	2.18
npm:@octokit/rest	48	2.18
npm:@types/selenium-webdriver	48	2.18
npm:@vue/babel-preset-app	48	2.18
npm:cmd-shim	48	2.18
npm:collapse-white-space	48	2.18
npm:content-type-parser	48	2.18
npm:coveralls	48	2.18
npm:esbuild-android-64	48	2.18
npm:gulp-concat	48	2.18
npm:is-my-ip-valid	48	2.18
npm:js2xmlparser	48	2.18
npm:kdbush	48	2.18

npm:logform	48	2.18
npm:one-time	48	2.18
npm:opencollective-postinstall	48	2.18
npm:pg	48	2.18
npm:ps-tree	48	2.18
npm:pug	48	2.18
npm:triple-beam	48	2.18
npm:underscore.string	48	2.18
npm:winston-transport	48	2.18
npm:@octokit/plugin-rest- endpoint-methods	47	2.13
npm:@types/d3-color	47	2.13
npm:@vue/server-renderer	47	2.13
npm:app-root-path	47	2.13
npm:async-validator	47	2.13
npm:base-x	47	2.13
npm:csv-parse	47	2.13
npm:eslint-import-resolver- typescript	47	2.13
npm:fast-fifo	47	2.13
npm:firebase	47	2.13
npm:github-slugg	47	2.13
npm:gray-matter	47	2.13
npm:hast-util-whitespace	47	2.13
npm:isemail	47	2.13

npm:json-bigint	47	2.13
npm:nested-error-stacks	47	2.13
npm:node-domexception	47	2.13
npm:tinyqueue	47	2.13
npm:tsscmp	47	2.13
npm:which-pm-runs	47	2.13
npm:@rollup/rollup-linux-x64-gnu	46	2.09
npm:@rollup/rollup-linux-x64-musl	46	2.09
npm:@types/d3-interpolate	46	2.09
npm:@types/sinonjs_fake-timers	46	2.09
npm:anser	46	2.09
npm:ansi-red	46	2.09
npm:beeper	46	2.09
npm:codelyzer	46	2.09
npm:email-addresses	46	2.09
npm:find-versions	46	2.09
npm:grunt-contrib-connect	46	2.09
npm:jsdoc	46	2.09
npm:lodash._basetostring	46	2.09
npm:lunr	46	2.09
npm:semver-regex	46	2.09
npm:syntax-error	46	2.09
npm:use-isomorphic-layout-effect	46	2.09
npm:yorkie	46	2.09

npm:@fortawesome/fontawesome-common-types	45	2.04
npm:@mapbox/unitbezier	45	2.04
npm:@material-ui/types	45	2.04
npm:@rollup/rollup-win32-x64-msvc	45	2.04
npm:@types/d3-path	45	2.04
npm:@types/d3-shape	45	2.04
npm:@types/uuid	45	2.04
npm:buffers	45	2.04
npm:chainsaw	45	2.04
npm:client-only	45	2.04
npm:codemirror	45	2.04
npm:create-jest	45	2.04
npm:fast-url-parser	45	2.04
npm:git-raw-commits	45	2.04
npm:google-auth-library	45	2.04
npm:parents	45	2.04
npm:path-platform	45	2.04
npm:pg-connection-string	45	2.04
npm:qrcode-terminal	45	2.04
npm:regexp-clone	45	2.04
npm:saucelabs	45	2.04
npm:stream-buffers	45	2.04
npm:stylelint	45	2.04

npm:unherit	45	2.04
npm:webpack-hot-middlewre	45	2.04
npm:@mapbox/jsonlint-lines- primitives	44	2.00
npm:@rollup/rollup-android-arm- eabi	44	2.00
npm:@rollup/rollup-android-arm64	44	2.00
npm:@rollup/rollup-darwin-arm64	44	2.00
npm:@rollup/rollup-darwin-x64	44	2.00
npm:@rollup/rollup-linux-arm- gnueabihf	44	2.00
npm:@rollup/rollup-linux-arm64- gnu	44	2.00
npm:@rollup/rollup-linux-arm64- musl	44	2.00
npm:@rollup/rollup-win32-arm64- msvc	44	2.00
npm:@rollup/rollup-win32-ia32- msvc	44	2.00
npm:@types/whatwg-url	44	2.00
npm:binary	44	2.00
npm:crypto-js	44	2.00
npm:csscolorparser	44	2.00
npm:devtools-protocol	44	2.00
npm:dompurify	44	2.00

npm:find-babel-config	44	2.00
npm:has-symbol-support-x	44	2.00
npm:has-to-string-tag-x	44	2.00
npm:into-stream	44	2.00
npm:jasmine2	44	2.00
npm:js-sha3	44	2.00
npm:karma-phantomjs-launcher	44	2.00
npm:lodash.difference	44	2.00
npm:matchdep	44	2.00
npm:module-deps	44	2.00
npm:multipipe	44	2.00
npm:remark-rehype	44	2.00
npm:topo	44	2.00
npm:uvu	44	2.00
npm:webpack-core	44	2.00
npm:xmlcreate	44	2.00
npm:zod	44	2.00
npm:@babel/plugin-syntax-export- default-from	43	1.95
npm:@babel/polyfill	43	1.95
npm:@babel/preset-flow	43	1.95
npm:@react-aria/ssr	43	1.95
npm:@sinonjs/text-encoding	43	1.95
npm:@types/mdurl	43	1.95
npm:@types/react-is	43	1.95

npm:@types/webidl-conversions	43	1.95
npm:@types/webpack-env	43	1.95
npm:@vue/reactivity-transform	43	1.95
npm:append-buffer	43	1.95
npm:babel-preset-fbjs	43	1.95
npm:browser-pack	43	1.95
npm:browser-sync	43	1.95
npm:combine-source-map	43	1.95
npm:conventional-changelog-writer	43	1.95
npm:conventional-commits-filter	43	1.95
npm:deps-sort	43	1.95
npm:doctypes	43	1.95
npm:eyes	43	1.95
npm:fs-mkdirp-stream	43	1.95
npm:gcp-metadata	43	1.95
npm:gtoken	43	1.95
npm:inline-source-map	43	1.95
npm:insert-module-globals	43	1.95
npm:is-expression	43	1.95
npm:isurl	43	1.95
npm:js-stringify	43	1.95
npm:lead	43	1.95
npm:lodash._createassigner	43	1.95
npm:lodash.some	43	1.95

npm:mongodb-connection-string-url	43	1.95
npm:now-and-later	43	1.95
npm:pathe	43	1.95
npm:pug-attrs	43	1.95
npm:pug-error	43	1.95
npm:pug-runtime	43	1.95
npm:pug-strip-comments	43	1.95
npm:remove-bom-buffer	43	1.95
npm:remove-bom-stream	43	1.95
npm:resolve-options	43	1.95
npm:sanitize-filename	43	1.95
npm:stat-mode	43	1.95
npm:to-through	43	1.95
npm:token-stream	43	1.95
npm:truncate-utf8-bytes	43	1.95
npm:umd	43	1.95
npm:url-to-options	43	1.95
npm:utf8-byte-length	43	1.95
npm:value-or-function	43	1.95
npm:vinyl-sourcemap	43	1.95
npm:wait-on	43	1.95
npm:webpack-virtual-modules	43	1.95
npm:@esbuild/aix-ppc64	42	1.91
npm:@mui/core-downloads-tracker	42	1.91

npm:@octokit/core	42	1.91
npm:@octokit/graphql	42	1.91
npm:@types/jsdom	42	1.91
npm:ansi-cyan	42	1.91
npm:boolean	42	1.91
npm:cached-path-relative	42	1.91
npm:catharsis	42	1.91
npm:change-case	42	1.91
npm:character-entities-html4	42	1.91
npm:fault	42	1.91
npm:format	42	1.91
npm:html-void-elements	42	1.91
npm:inflection	42	1.91
npm:lodash._basevalues	42	1.91
npm:lodash._reescape	42	1.91
npm:lodash._reevaluate	42	1.91
npm:lodash.clone	42	1.91
npm:lodash.filter	42	1.91
npm:lodash.isfunction	42	1.91
npm:lodash.reduce	42	1.91
npm:lodash.set	42	1.91
npm:lodash.transform	42	1.91
npm:moo	42	1.91
npm:murmurhash-js	42	1.91
npm:node-environment-flags	42	1.91

npm:node-uuid	42	1.91
npm:output-file-sync	42	1.91
npm:parse-headers	42	1.91
npm:passport-local	42	1.91
npm:pug-code-gen	42	1.91
npm:pug-filters	42	1.91
npm:pug-lexer	42	1.91
npm:pug-linker	42	1.91
npm:pug-load	42	1.91
npm:pug-parser	42	1.91
npm:pug-walk	42	1.91
npm:read-all-stream	42	1.91
npm:requizzle	42	1.91
npm:stringify-entities	42	1.91
npm:supercluster	42	1.91
npm:topojson-client	42	1.91
npm:upper-case-first	42	1.91
npm:@firebase/database	41	1.86
npm:@firebase/util	41	1.86
npm:@floating-ui/react-dom	41	1.86
npm:@mapbox/vector-tile	41	1.86
npm:@octokit/plugin-request-log	41	1.86
npm:@types/jasminewd2	41	1.86
npm:@vue/cli-plugin-router	41	1.86
npm:@xmldom/xmldom	41	1.86

npm:array-last	41	1.86
npm:blocking-proxy	41	1.86
npm:caching-transform	41	1.86
npm:charcodes	41	1.86
npm:chromium-pickle-js	41	1.86
npm:coffee-script	41	1.86
npm:cypress	41	1.86
npm:gaxios	41	1.86
npm:git-up	41	1.86
npm:htmlescape	41	1.86
npm:is-ssh	41	1.86
npm:jade	41	1.86
npm:labeled-stream-splicer	41	1.86
npm:libnpmaccess	41	1.86
npm:libnpmublish	41	1.86
npm:lodash.escaperegexp	41	1.86
npm:markdown-it-anchor	41	1.86
npm:papaparse	41	1.86
npm:parse-url	41	1.86
npm:protocols	41	1.86
npm:read-only-stream	41	1.86
npm:redis-parser	41	1.86
npm:section-matter	41	1.86
npm:tiny-emitter	41	1.86
npm:trim-trailing-lines	41	1.86

npm:webdriver-js-extender	41	1.86
npm:@angular/language-service	40	1.82
npm:@firebase/app-types	40	1.82
npm:@firebase/database-types	40	1.82
npm:@rollup/rollup-linux-riscv64-gnu	40	1.82
npm:@vue/cli-plugin-vuex	40	1.82
npm:aria-hidden	40	1.82
npm:async-done	40	1.82
npm:babelify	40	1.82
npm:buffer-json	40	1.82
npm:cuint	40	1.82
npm:denodeify	40	1.82
npm:fetch-blob	40	1.82
npm:fs-access	40	1.82
npm:geojson-vt	40	1.82
npm:globrex	40	1.82
npm:inline-style-prefixer	40	1.82
npm:is-subset	40	1.82
npm:is-whitespace-character	40	1.82
npm:is-word-character	40	1.82
npm:js-queue	40	1.82
npm:karma-firefox-launcher	40	1.82
npm:log-driver	40	1.82
npm:markdown-escapes	40	1.82

npm:modify-values	40	1.82
npm:mongodb-core	40	1.82
npm:node-ipc	40	1.82
npm:null-check	40	1.82
npm:os-name	40	1.82
npm:package-hash	40	1.82
npm:possible-typed-array-names	40	1.82
npm:potpack	40	1.82
npm:release-zalgo	40	1.82
npm:resolve-pkg-maps	40	1.82
npm:roarr	40	1.82
npm:sentence-case	40	1.82
npm:shallow-copy	40	1.82
npm:spawn-wrap	40	1.82
npm:sqlstring	40	1.82
npm:state-toggle	40	1.82
npm:stream-exhaust	40	1.82
npm:stream-splicer	40	1.82
npm:streamx	40	1.82
npm:ufo	40	1.82
npm:unist-builder	40	1.82
npm:@firebase/logger	39	1.77
npm:@types/d3-array	39	1.77
npm:@types/d3-timer	39	1.77
npm:arr-map	39	1.77

npm:array-sort	39	1.77
npm:b4a	39	1.77
npm:btoa	39	1.77
npm:cac	39	1.77
npm:constant-case	39	1.77
npm:default-compare	39	1.77
npm:esm	39	1.77
npm:exponential-backoff	39	1.77
npm:fast-text-encoding	39	1.77
npm:global-agent	39	1.77
npm:limiter	39	1.77
npm:lodash.assignin	39	1.77
npm:object.reduce	39	1.77
npm:p-filter	39	1.77
npm:path-case	39	1.77
npm:plur	39	1.77
npm:randexp	39	1.77
npm:string-template	39	1.77
npm:typical	39	1.77
npm:@angular/cdk	38	1.72
npm:@intervolga/optimize- cssnano-plugin	38	1.72
npm:@sheerun/mutationobserver- shim	38	1.72
npm:@types/parse5	38	1.72

npm:@vue/preload-webpack-plugin	38	1.72
npm:arr-filter	38	1.72
npm:array-initial	38	1.72
npm:async-settle	38	1.72
npm:babel-plugin-module-resolver	38	1.72
npm:babel-plugin-transform-react-constant-elements	38	1.72
npm:bach	38	1.72
npm:clean-webpack-plugin	38	1.72
npm:collection-map	38	1.72
npm:copy-props	38	1.72
npm:default-resolution	38	1.72
npm:each-props	38	1.72
npm:fbemitter	38	1.72
npm:google-p12-pem	38	1.72
npm:hammerjs	38	1.72
npm:hast-util-raw	38	1.72
npm:hast-util-to-parse5	38	1.72
npm:just-debounce	38	1.72
npm:last-run	38	1.72
npm:mute-stdout	38	1.72
npm:native-url	38	1.72
npm:natives	38	1.72
npm:normalize.css	38	1.72
npm:parse-path	38	1.72

npm:postcss-media-query-parser	38	1.72
npm:pretty-ms	38	1.72
npm:queue-tick	38	1.72
npm:react-chartjs-2	38	1.72
npm:react-shallow-renderer	38	1.72
npm:remark-mdx	38	1.72
npm:replace-homedir	38	1.72
npm:semver-greatest-satisfied-range	38	1.72
npm:strip-bom-stream	38	1.72
npm:sver-compat	38	1.72
npm:tildify	38	1.72
npm:tryit	38	1.72
npm:tsconfig	38	1.72
npm:undertaker	38	1.72
npm:undertaker-registry	38	1.72
npm:vt-pbf	38	1.72
npm:@emotion/css	37	1.68
npm:@firebase/app	37	1.68
npm:@firebase/auth-interop-types	37	1.68
npm:@firebase/component	37	1.68
npm:@firebase/firestore	37	1.68
npm:@floating-ui/utils	37	1.68
npm:@mdx-js/react	37	1.68
npm:@restart/ui	37	1.68

npm:@types/d3-scale-chromatic	37	1.68
npm:@types/testing-library__dom	37	1.68
npm:@types/testing-library__react	37	1.68
npm:array-back	37	1.68
npm:async-retry	37	1.68
npm:babel-plugin-emotion	37	1.68
npm:bcryptjs	37	1.68
npm:bs58	37	1.68
npm:cli-color	37	1.68
npm:css-in-js-utils	37	1.68
npm:error	37	1.68
npm:formdata-polyfill	37	1.68
npm:ftp	37	1.68
npm:git-url-parse	37	1.68
npm:junk	37	1.68
npm:keycode	37	1.68
npm:livereload-js	37	1.68
npm:lodash.bind	37	1.68
npm:nearley	37	1.68
npm:parse-ms	37	1.68
npm:parsejson	37	1.68
npm:postcss-url	37	1.68
npm:rc-slider	37	1.68
npm:redis	37	1.68
npm:shiki	37	1.68

npm:should	37	1.68
npm:string-hash	37	1.68
npm:synckit	37	1.68
npm:trim-lines	37	1.68
npm:typedoc	37	1.68
npm:wait-for-expect	37	1.68
npm:@firebase/auth	36	1.63
npm:@firebase/messaging	36	1.63
npm:@firebase/storage	36	1.63
npm:@firebase/webchannel- wrapper	36	1.63
npm:@fortawesome/react- fontawesome	36	1.63
npm:@mapbox/geojson-rewind	36	1.63
npm:@mapbox/tiny-sdf	36	1.63
npm:@mapbox/whoots-js	36	1.63
npm:@mdx-js/mdx	36	1.63
npm:@npmcli/map-workspaces	36	1.63
npm:@npmcli/name-from-folder	36	1.63
npm:@parcel/watcher	36	1.63
npm:@turf/helpers	36	1.63
npm:@types/d3	36	1.63
npm:@types/d3-ease	36	1.63
npm:@types/d3-selection	36	1.63
npm:@types/linkify-it	36	1.63

npm:@types/markdown-it	36	1.63
npm:@vitejs/plugin-vue	36	1.63
npm:array.prototype.find	36	1.63
npm:babel-preset-stage-0	36	1.63
npm:bin-links	36	1.63
npm:blueimp-md5	36	1.63
npm:discontinuous-range	36	1.63
npm:eslint-config-next	36	1.63
npm:expand-braces	36	1.63
npm:flow-parser	36	1.63
npm:fromentries	36	1.63
npm:get-assigned-identifiers	36	1.63
npm:grunt-jscs	36	1.63
npm:js-string-escape	36	1.63
npm:lodash._basecreate	36	1.63
npm:lodash.isobject	36	1.63
npm:lodash.snakecase	36	1.63
npm:lodash.uniqby	36	1.63
npm:mapbox-gl	36	1.63
npm:nocache	36	1.63
npm:p-pipe	36	1.63
npm:pg-types	36	1.63
npm:railroad-diagrams	36	1.63
npm:react-hook-form	36	1.63
npm:seek-bzip	36	1.63

npm:stream-events	36	1.63
npm:strip-dirs	36	1.63
npm:stubs	36	1.63
npm:sudo-prompt	36	1.63
npm:svelte	36	1.63
npm:tiny-glob	36	1.63
npm:tmp-promise	36	1.63
npm:urijs	36	1.63
npm:worker-loader	36	1.63
npm:xhr	36	1.63
npm:@electron/get	35	1.59
npm:@firebase/auth-types	35	1.59
npm:@firebase/firestore-types	35	1.59
npm:@firebase/storage-types	35	1.59
npm:@ungap/promise-all-settled	35	1.59
npm:bcrypt	35	1.59
npm:chartjs-color	35	1.59
npm:check-more-types	35	1.59
npm:circular-dependency-plugin	35	1.59
npm:critters	35	1.59
npm:cycle	35	1.59
npm:decompress	35	1.59
npm:decompress-tar	35	1.59
npm:decompress-tarbz2	35	1.59
npm:decompress-targz	35	1.59

npm:decompress-unzip	35	1.59
npm:detect-node-es	35	1.59
npm:electron-builder	35	1.59
npm:elegant-spinner	35	1.59
npm:eslint-restricted-globals	35	1.59
npm:fflate	35	1.59
npm:friendly-errors-webpack- plugin	35	1.59
npm:globalyzer	35	1.59
npm:has-binary	35	1.59
npm:heap	35	1.59
npm:helmet	35	1.59
npm:html-loader	35	1.59
npm:http-server	35	1.59
npm:ip-address	35	1.59
npm:is-natural-number	35	1.59
npm:jscodeshift	35	1.59
npm:known-css-properties	35	1.59
npm:lazy-ass	35	1.59
npm:lodash.omit	35	1.59
npm:milly	35	1.59
npm:parse5-html-rewriting-stream	35	1.59
npm:parse5-sax-parser	35	1.59
npm:pg-pool	35	1.59
npm:pgpass	35	1.59

npm:pkg-types	35	1.59
npm:postgres-array	35	1.59
npm:postgres-bytea	35	1.59
npm:postgres-date	35	1.59
npm:postgres-interval	35	1.59
npm:pretty-time	35	1.59
npm:puppeteer-core	35	1.59
npm:rc-util	35	1.59
npm:readdir-glob	35	1.59
npm:remark-stringify	35	1.59
npm:secp256k1	35	1.59
npm:server-destroy	35	1.59
npm:unfetch	35	1.59
npm:use-callback-ref	35	1.59
npm:use-sidecar	35	1.59
npm:zen-observable	35	1.59
npm:@cypress/xvfb	34	1.54
npm:@emotion/core	34	1.54
npm:@next/eslint-plugin-next	34	1.54
npm:@types/node-forge	34	1.54
npm:@vue/babel-sugar- composition-api-inject-h	34	1.54
npm:@vue/babel-sugar- composition-api-render-instance	34	1.54
npm:async-each-series	34	1.54

npm:blob-util	34	1.54
npm:browserstack	34	1.54
npm:buffer-writer	34	1.54
npm:compare-version	34	1.54
npm:create-react-context	34	1.54
npm:cssaaron	34	1.54
npm:dom-align	34	1.54
npm:fn.name	34	1.54
npm:getos	34	1.54
npm:grunt-autoprefixer	34	1.54
npm:gulp-header	34	1.54
npm:header-case	34	1.54
npm:is	34	1.54
npm:lodash.ismatch	34	1.54
npm:mdast-util-phrasing	34	1.54
npm:npm-conf	34	1.54
npm:packet-reader	34	1.54
npm:phantomjs-prebuilt	34	1.54
npm:react-modal	34	1.54
npm:react-overlays	34	1.54
npm:react-textarea-autosize	34	1.54
npm:semver-dsl	34	1.54
npm:teeny-request	34	1.54
npm:tiny-lr	34	1.54
npm:utf-8-validate	34	1.54

npm:webpackbar	34	1.54
npm:wtf-8	34	1.54
npm:@assemblyscript/loader	33	1.50
npm:@dabh/diagnostics	33	1.50
npm:@firebase/analytics	33	1.50
npm:@firebase/analytics-types	33	1.50
npm:@firebase/functions	33	1.50
npm:@firebase/functions-types	33	1.50
npm:@firebase/installations	33	1.50
npm:@firebase/performance	33	1.50
npm:@firebase/performance-types	33	1.50
npm:@firebase/remote-config	33	1.50
npm:@firebase/remote-config-types	33	1.50
npm:@mapbox/mapbox-gl-supported	33	1.50
npm:@npmcli/package-json	33	1.50
npm:@tufjs/models	33	1.50
npm:@types/d3-geo	33	1.50
npm:@vue/babel-helper-vue-transform-on	33	1.50
npm:@vue/babel-plugin-jsx	33	1.50
npm:@vue/devtools-api	33	1.50
npm:array-from	33	1.50
npm:async-exit-hook	33	1.50
npm:byline	33	1.50

npm:byte-size	33	1.50
npm:chartjs-color-string	33	1.50
npm:command-line-args	33	1.50
npm:d3-queue	33	1.50
npm:dash-ast	33	1.50
npm:enzyme-adapter-react-16	33	1.50
npm:esbuild-wasm	33	1.50
npm:get-nonce	33	1.50
npm:grid-index	33	1.50
npm:grunt-contrib-csslint	33	1.50
npm:irregular-plurals	33	1.50
npm:isomorphic-ws	33	1.50
npm:jest-matchers	33	1.50
npm:lcov-parse	33	1.50
npm:lodash.has	33	1.50
npm:lodash.unescape	33	1.50
npm:luxon	33	1.50
npm:macos-release	33	1.50
npm:mysql	33	1.50
npm:ospath	33	1.50
npm:piscina	33	1.50
npm:react-dropzone	33	1.50
npm:react-native	33	1.50
npm:react-remove-scroll	33	1.50
npm:react-remove-scroll-bar	33	1.50

npm:react-style-singleton	33	1.50
npm:react-toastify	33	1.50
npm:requireindex	33	1.50
npm:samsam	33	1.50
npm:shasum	33	1.50
npm:sigstore	33	1.50
npm:sqlite3	33	1.50
npm:tuf-js	33	1.50
npm:walk-up-path	33	1.50
npm:xml-js	33	1.50
npm:@cypress/request	32	1.45
npm:@sigstore/protobuf-specs	32	1.45
npm:@soda/get-current-script	32	1.45
npm:@types/jsonwebtoken	32	1.45
npm:babel-plugin-add-module-exports	32	1.45
npm:bower	32	1.45
npm:cli	32	1.45
npm:command-line-usage	32	1.45
npm:css-box-model	32	1.45
npm:falafel	32	1.45
npm:find-index	32	1.45
npm:find-replace	32	1.45
npm:genfun	32	1.45
npm:litr	32	1.45

npm:lodash.create	32	1.45
npm:lodash.reject	32	1.45
npm:marked-terminal	32	1.45
npm:node-fetch-npm	32	1.45
npm:nprogress	32	1.45
npm:platform	32	1.45
npm:protoduck	32	1.45
npm:queue	32	1.45
npm:rc-tooltip	32	1.45
npm:react-markdown	32	1.45
npm:redux-logger	32	1.45
npm:requirejs	32	1.45
npm:sanitize-html	32	1.45
npm:string.prototype.padstart	32	1.45
npm:stylelint-config-recommended	32	1.45
npm:text-encoding	32	1.45
npm:utility-types	32	1.45
npm:xlsx	32	1.45
npm:xxhashjs	32	1.45
npm:zen-observable-ts	32	1.45
npm:@angular-devkit/build-optimizer	31	1.41
npm:@ctrl/tinycolor	31	1.41
npm:@graphql-typed-document-node/core	31	1.41

npm:@nicolo-ribaudo/chokidar-2	31	1.41
npm:@sentry/core	31	1.41
npm:@sentry/types	31	1.41
npm:@sentry/utils	31	1.41
npm:@types/request	31	1.41
npm:any-observable	31	1.41
npm:chai-as-promised	31	1.41
npm:comment-parser	31	1.41
npm:common-ancestor-path	31	1.41
npm:csv-stringify	31	1.41
npm:default-browser-id	31	1.41
npm:dev-ip	31	1.41
npm:es6-object-assign	31	1.41
npm:form-data-encoder	31	1.41
npm:framer-motion	31	1.41
npm:is-observable	31	1.41
npm:listr-silent-renderer	31	1.41
npm:listr-update-renderer	31	1.41
npm:listr-verbose-renderer	31	1.41
npm:local-pkg	31	1.41
npm:lodash.curry	31	1.41
npm:mdast-util-find-and-replace	31	1.41
npm:mdast-util-gfm	31	1.41
npm:mdast-util-gfm-autolink- literal	31	1.41

npm:mdast-util-gfm-strikethrough	31	1.41
npm:mdast-util-gfm-table	31	1.41
npm:mdast-util-gfm-task-list-item	31	1.41
npm:method-override	31	1.41
npm:mongoose-legacy-pluralize	31	1.41
npm:nice-napi	31	1.41
npm:node-status-codes	31	1.41
npm:noop-logger	31	1.41
npm:npm-user-validate	31	1.41
npm:rc-align	31	1.41
npm:rc-trigger	31	1.41
npm:read-package-tree	31	1.41
npm:recompose	31	1.41
npm:resumer	31	1.41
npm:safe-json-parse	31	1.41
npm:scroll-into-view-if-needed	31	1.41
npm:sw-toolbox	31	1.41
npm:taffydb	31	1.41
npm:typed-assert	31	1.41
npm:vitest	31	1.41
npm:windows-release	31	1.41
npm:wordwrapjs	31	1.41
npm:@angular/material	30	1.36
npm:@reduxjs/toolkit	30	1.36
npm:@sinonjs/formatio	30	1.36

npm:@types/caseless	30	1.36
npm:@types/d3-dsv	30	1.36
npm:@types/d3-format	30	1.36
npm:@types/d3-time-format	30	1.36
npm:@types/d3-transition	30	1.36
npm:@types/d3-zoom	30	1.36
npm:@types/invariant	30	1.36
npm:babel-preset-stage-1	30	1.36
npm:blakejs	30	1.36
npm:bufferutil	30	1.36
npm:combine-lists	30	1.36
npm:cookies	30	1.36
npm:d3-request	30	1.36
npm:globjoin	30	1.36
npm:hdr-histogram-js	30	1.36
npm:istanbul-lib-processinfo	30	1.36
npm:iterall	30	1.36
npm:karma-mocha	30	1.36
npm:karma-webpack	30	1.36
npm:keygrip	30	1.36
npm:localtunnel	30	1.36
npm:mathml-tag-names	30	1.36
npm:micromark-extension-gfm	30	1.36
npm:micromark-extension-gfm-autolink-literal	30	1.36

npm:micromark-extension-gfm-strikethrough	30	1.36
npm:micromark-extension-gfm-table	30	1.36
npm:micromark-extension-gfm-tagfilter	30	1.36
npm:micromark-extension-gfm-task-list-item	30	1.36
npm:node-preload	30	1.36
npm:openurl	30	1.36
npm:portscanner	30	1.36
npm:postcss-resolve-nested-selector	30	1.36
npm:process-on-spawn	30	1.36
npm:proper-lockfile	30	1.36
npm:redis-commands	30	1.36
npm:remark-gfm	30	1.36
npm:retry-request	30	1.36
npm:rollup-plugin-node-resolve	30	1.36
npm:slugify	30	1.36
npm:stylelint-config-standard	30	1.36
npm:table-layout	30	1.36
npm:typed-styles	30	1.36
npm:utf8	30	1.36
npm:vscode-textmate	30	1.36
npm:websocket	30	1.36

npm:@fastify/busboy	29	1.32
npm:@material-ui/lab	29	1.32
npm:@sentry/hub	29	1.32
npm:@tufjs/canonical-json	29	1.32
npm:@turf/meta	29	1.32
npm:@types/d3-drag	29	1.32
npm:@types/d3-force	29	1.32
npm:@types/d3-hierarchy	29	1.32
npm:@types/d3-random	29	1.32
npm:7zip-bin	29	1.32
npm:antd	29	1.32
npm:array-tree-filter	29	1.32
npm:atob-lite	29	1.32
npm:babel-helper-vue-jsx-merge-props	29	1.32
npm:babel-plugin-syntax-export-extensions	29	1.32
npm:babel-plugin-transform-export-extensions	29	1.32
npm:bluebird-lst	29	1.32
npm:builder-util	29	1.32
npm:builder-util-runtime	29	1.32
npm:change-emitter	29	1.32
npm:cidr-regex	29	1.32
npm:cli-columns	29	1.32

npm:dmg-builder	29	1.32
npm:electron-publish	29	1.32
npm:eslint-plugin-react-refresh	29	1.32
npm:gitconfiglocal	29	1.32
npm:gulp-clean-css	29	1.32
npm:has-color	29	1.32
npm:is-cidr	29	1.32
npm:lazy-val	29	1.32
npm:lockfile	29	1.32
npm:metro	29	1.32
npm:metro-cache	29	1.32
npm:metro-core	29	1.32
npm:metro-react-native-babel- preset	29	1.32
npm:metro-resolver	29	1.32
npm:metro-source-map	29	1.32
npm:npm-audit-report	29	1.32
npm:npm-profile	29	1.32
npm:pg-int8	29	1.32
npm:polished	29	1.32
npm:react-devtools-core	29	1.32
npm:react-input-autosize	29	1.32
npm:react-json-view	29	1.32
npm:react-table	29	1.32
npm:read-config-file	29	1.32

npm:resp-modifier	29	1.32
npm:rst-selector-parser	29	1.32
npm:shortid	29	1.32
npm:sinon-chai	29	1.32
npm:style-search	29	1.32
npm:temp-file	29	1.32
npm:tiny-relative-date	29	1.32
npm:tinypool	29	1.32
npm:tinyspy	29	1.32
npm:ts-dedent	29	1.32
npm:use-latest	29	1.32
npm:varint	29	1.32
npm:weak-map	29	1.32
npm:xml	29	1.32
npm:yaeti	29	1.32
npm:@commitlint/execute-rule	28	1.27
npm:@commitlint/load	28	1.27
npm:@commitlint/resolve-extends	28	1.27
npm:@iarna/toml	28	1.27
npm:@mapbox/geojson-types	28	1.27
npm:@samverschueren/stream-to-observable	28	1.27
npm:@types/d3-axis	28	1.27
npm:@types/d3-brush	28	1.27
npm:@types/d3-chord	28	1.27

npm:@types/d3-dispatch	28	1.27
npm:@types/d3-polygon	28	1.27
npm:@types/d3-quadtree	28	1.27
npm:@types/estree-jsx	28	1.27
npm:@types/js-yaml	28	1.27
npm:@types/webpack-dev-server	28	1.27
npm:@vue/test-utils	28	1.27
npm:ansi-to-html	28	1.27
npm:app-builder-bin	28	1.27
npm:asar	28	1.27
npm:babel-plugin-syntax-class- constructor-call	28	1.27
npm:babel-plugin-transform-class- constructor-call	28	1.27
npm:backbone	28	1.27
npm:binaryextensions	28	1.27
npm:body	28	1.27
npm:clipboard	28	1.27
npm:continuable-cache	28	1.27
npm:conventional-changelog-core	28	1.27
npm:conventional-changelog- preset-loader	28	1.27
npm:cosmiconfig-typescript-loader	28	1.27
npm:eventemitter-asynresource	28	1.27
npm:getobject	28	1.27

npm:git-remote-origin-url	28	1.27
npm:git-semver-tags	28	1.27
npm:grunt-html	28	1.27
npm:grunt-legacy-log	28	1.27
npm:grunt-legacy-log-utils	28	1.27
npm:grunt-legacy-util	28	1.27
npm:hdr-histogram-percentiles-obj	28	1.27
npm:hooker	28	1.27
npm:is-ip	28	1.27
npm:is-number-like	28	1.27
npm:jonschema	28	1.27
npm:jwt-decode	28	1.27
npm:libnpmhook	28	1.27
npm:libnpmorg	28	1.27
npm:libnpmsearch	28	1.27
npm:libnpmteam	28	1.27
npm:lodash.isfinite	28	1.27
npm:mdast-util-mdx-expression	28	1.27
npm:metro-config	28	1.27
npm:metro-minify-uglify	28	1.27
npm:node-abort-controller	28	1.27
npm:node-stream-zip	28	1.27
npm:nomnom	28	1.27
npm:parse-srcset	28	1.27
npm:pkginfo	28	1.27

npm:preact	28	1.27
npm:reduce-flatten	28	1.27
npm:rollup-plugin-commonjs	28	1.27
npm:serviceworker-cache-polyfill	28	1.27
npm:sharp	28	1.27
npm:strip-literal	28	1.27
npm:textextensions	28	1.27
npm:uid2	28	1.27
npm:use-composed-ref	28	1.27
npm:vscode-oniguruma	28	1.27
npm:vuetify	28	1.27
npm:xss	28	1.27
npm:yup	28	1.27
npm:@aws-sdk/types	27	1.23
npm:@commitlint/cli	27	1.23
npm:@google-cloud/projectify	27	1.23
npm:@google-cloud/promisify	27	1.23
npm:@npmcli/arborist	27	1.23
npm:@npmcli/config	27	1.23
npm:@npmcli/metavuln-calculator	27	1.23
npm:@sentry/minimal	27	1.23
npm:@types/d3-fetch	27	1.23
npm:@types/strip-bom	27	1.23
npm:@types/strip-json-comments	27	1.23
npm:@wry/equality	27	1.23

npm:abortcontroller-polyfill	27	1.23
npm:algiasearch	27	1.23
npm:attr-accept	27	1.23
npm:better-opn	27	1.23
npm:css-url-regex	27	1.23
npm:dom-urls	27	1.23
npm:easy-extender	27	1.23
npm:eazy-logger	27	1.23
npm:exit-on-epipe	27	1.23
npm:find-yarn-workspace-root	27	1.23
npm:flux	27	1.23
npm:fs-exists-sync	27	1.23
npm:get-pkg-repo	27	1.23
npm:glob2base	27	1.23
npm:google-gax	27	1.23
npm:gulp-autoprefixer	27	1.23
npm:gulp-cli	27	1.23
npm:hast-to-hyperscript	27	1.23
npm:jjj	27	1.23
npm:json-stringify-nice	27	1.23
npm:jsonpath	27	1.23
npm:just-diff	27	1.23
npm:just-diff-apply	27	1.23
npm:karma-sourcemap-loader	27	1.23
npm:keccak	27	1.23

npm:kew	27	1.23
npm:lodash.startcase	27	1.23
npm:native-promise-only	27	1.23
npm:parse-conflict-json	27	1.23
npm:periscope	27	1.23
npm:postcss-scss	27	1.23
npm:printj	27	1.23
npm:promise-all-reject-late	27	1.23
npm:promise-call-limit	27	1.23
npm:property-expr	27	1.23
npm:react-helmet	27	1.23
npm:remark	27	1.23
npm:set-cookie-parser	27	1.23
npm:sort-object-keys	27	1.23
npm:sw-precache	27	1.23
npm:title-case	27	1.23
npm:treeverse	27	1.23
npm:ts-invariant	27	1.23
npm:undeclared-identifiers	27	1.23
npm:unist-util-remove	27	1.23
npm:unzipper	27	1.23
npm:vite-node	27	1.23
npm:webdriver-manager	27	1.23
npm:write-json-file	27	1.23
npm:x-is-string	27	1.23

npm:@angular/http	26	1.18
npm:@ant-design/colors	26	1.18
npm:@aws-crypto/ie11-detection	26	1.18
npm:@aws-crypto/sha256-browser	26	1.18
npm:@aws-crypto/sha256-js	26	1.18
npm:@aws-crypto/supports-web-crypto	26	1.18
npm:@aws-crypto/util	26	1.18
npm:@aws-sdk/credential-provider-env	26	1.18
npm:@aws-sdk/credential-provider-ini	26	1.18
npm:@aws-sdk/credential-provider-node	26	1.18
npm:@aws-sdk/credential-provider-process	26	1.18
npm:@aws-sdk/middleware-host-header	26	1.18
npm:@aws-sdk/middleware-logger	26	1.18
npm:@aws-sdk/middleware-user-agent	26	1.18
npm:@aws-sdk/util-locate-window	26	1.18
npm:@aws-sdk/util-user-agent-browser	26	1.18

npm:@aws-sdk/util-user-agent-node	26	1.18
npm:@aws-sdk/util-utf8-browser	26	1.18
npm:@commitlint/config-conventional	26	1.18
npm:@firebase/app-check-interop-types	26	1.18
npm:@react-native-community/cli-platform-android	26	1.18
npm:@react-native-community/cli-platform-ios	26	1.18
npm:@react-native-community/cli-tools	26	1.18
npm:@react-native-community/cli-types	26	1.18
npm:@types/bn.js	26	1.18
npm:@types/d3-contour	26	1.18
npm:@types/mkdirp	26	1.18
npm:@vitest/expect	26	1.18
npm:@vitest/snapshot	26	1.18
npm:@vitest/spy	26	1.18
npm:@vitest/utils	26	1.18
npm:aes-js	26	1.18
npm:ansi-fragments	26	1.18
npm:app-builder-lib	26	1.18

npm:archive-type	26	1.18
npm:assert-never	26	1.18
npm:base16	26	1.18
npm:bplist-creator	26	1.18
npm:browser-sync-ui	26	1.18
npm:bs-recipes	26	1.18
npm:bs58check	26	1.18
npm:buffer-indexof-polyfill	26	1.18
npm:capital-case	26	1.18
npm:cssfilter	26	1.18
npm:d3-geo-projection	26	1.18
npm:deep-diff	26	1.18
npm:delegate	26	1.18
npm:double-ended-queue	26	1.18
npm:download	26	1.18
npm:electron-osx-sign	26	1.18
npm:estree-util-is-identifier-name	26	1.18
npm:ethers	26	1.18
npm:gulp-jshint	26	1.18
npm:is-inside-container	26	1.18
npm:is-lower-case	26	1.18
npm:is-upper-case	26	1.18
npm:istextorbinary	26	1.18
npm:jsc-android	26	1.18
npm:json-stringify-pretty-compact	26	1.18

npm:json2mq	26	1.18
npm:listenercount	26	1.18
npm:lodash._objecttypes	26	1.18
npm:lodash.flow	26	1.18
npm:lodash.toArray	26	1.18
npm:logkitty	26	1.18
npm:lower-case-first	26	1.18
npm:md5-file	26	1.18
npm:mdast-util-mdxjs-esm	26	1.18
npm:metro-babel-transformer	26	1.18
npm:metro-symbolicate	26	1.18
npm:oauth	26	1.18
npm:ob1	26	1.18
npm:prop-types-exact	26	1.18
npm:rc-cascader	26	1.18
npm:rc-drawer	26	1.18
npm:rc-tree	26	1.18
npm:react-base16-styling	26	1.18
npm:react-error-boundary	26	1.18
npm:recharts	26	1.18
npm:rlp	26	1.18
npm:sort-package-json	26	1.18
npm:static-module	26	1.18
npm:stream-throttle	26	1.18
npm:string-convert	26	1.18

npm:sw-precache-webpack-plugin	26	1.18
npm:swap-case	26	1.18
npm:tiny-inflate	26	1.18
npm:util-promisify	26	1.18
npm:vscode-languageserver-types	26	1.18
npm:@aws-sdk/client-sso	25	1.13
npm:@aws-sdk/client-sts	25	1.13
npm:@aws-sdk/credential- provider-sso	25	1.13
npm:@aws-sdk/credential- provider-web-identity	25	1.13
npm:@aws-sdk/middleware-signing	25	1.13
npm:@babel/runtime-corejs2	25	1.13
npm:@commitlint/types	25	1.13
npm:@firebase/app-check	25	1.13
npm:@firebase/database-compat	25	1.13
npm:@firebase/installations-types	25	1.13
npm:@react-native-community/cli- debugger-ui	25	1.13
npm:@rollup/plugin-json	25	1.13
npm:@turf/bbox	25	1.13
npm:@types/acorn	25	1.13
npm:@types/rimraf	25	1.13
npm:@vitest/runner	25	1.13
npm:brfs	25	1.13

npm:c8	25	1.13
npm:conventional-changelog- conventionalcommits	25	1.13
npm:cp-file	25	1.13
npm:eslint-plugin-html	25	1.13
npm:ethereumjs-util	25	1.13
npm:fast-json-patch	25	1.13
npm:formatio	25	1.13
npm:frac	25	1.13
npm:generic-names	25	1.13
npm:good-listener	25	1.13
npm:grunt-known-options	25	1.13
npm:html-element-map	25	1.13
npm:i18next	25	1.13
npm:is-hex-prefixed	25	1.13
npm:is-window	25	1.13
npm:istanbul-instrumenter-loader	25	1.13
npm:lazy-req	25	1.13
npm:lerna	25	1.13
npm:make-error-cause	25	1.13
npm:mdast-util-mdx-jsx	25	1.13
npm:metro-inspector-proxy	25	1.13
npm:metro-react-native-babel- transformer	25	1.13
npm:plotly.js	25	1.13

npm:postcss-reporter	25	1.13
npm:quote-stream	25	1.13
npm:rc-motion	25	1.13
npm:rc-select	25	1.13
npm:react-draggable	25	1.13
npm:react-helmet-async	25	1.13
npm:readline	25	1.13
npm:reflect.ownkeys	25	1.13
npm:request-promise	25	1.13
npm:run-applescript	25	1.13
npm:script-js	25	1.13
npm:select	25	1.13
npm:semver-intersect	25	1.13
npm:serve-handler	25	1.13
npm:set-getter	25	1.13
npm:signinfo	25	1.13
npm:spark-md5	25	1.13
npm:ssf	25	1.13
npm:stackback	25	1.13
npm:strip-hex-prefix	25	1.13
npm:synchronous-promise	25	1.13
npm:tabbable	25	1.13
npm:tinybench	25	1.13
npm:titleize	25	1.13
npm:valid-url	25	1.13

npm:why-is-node-running	25	1.13
npm:zustand	25	1.13
npm:@ant-design/icons	24	1.09
npm:@aws-sdk/middleware- recursion-detection	24	1.09
npm:@commitlint/parse	24	1.09
npm:@develar/schema-utils	24	1.09
npm:@isaacs/string-locale-compare	24	1.09
npm:@jest/create-cache-key- function	24	1.09
npm:@pkgr/utils	24	1.09
npm:@react-native-community/cli- hermes	24	1.09
npm:@react-native-community/cli- server-api	24	1.09
npm:@sigstore/tuf	24	1.09
npm:@types/component-emitter	24	1.09
npm:@types/form-data	24	1.09
npm:@types/pug	24	1.09
npm:adler-32	24	1.09
npm:airbnb-prop-types	24	1.09
npm:angular	24	1.09
npm:basic-ftp	24	1.09
npm:bundle-name	24	1.09
npm:cfb	24	1.09

npm:cluster-key-slot	24	1.09
npm:codepage	24	1.09
npm:corser	24	1.09
npm:default-browser	24	1.09
npm:detab	24	1.09
npm:eslint-plugin-jsdoc	24	1.09
npm:fuse.js	24	1.09
npm:grunt-compare-size	24	1.09
npm:hermes-profile-transformer	24	1.09
npm:import-meta-resolve	24	1.09
npm:is-stream-ended	24	1.09
npm:lodash.padend	24	1.09
npm:lodash.upperfirst	24	1.09
npm:mdast-util-gfm-footnote	24	1.09
npm:mkdirp-infer-owner	24	1.09
npm:ndjson	24	1.09
npm:obliterator	24	1.09
npm:optimism	24	1.09
npm:p-map-series	24	1.09
npm:pbkdf2-compat	24	1.09
npm:pixelmatch	24	1.09
npm:pure-color	24	1.09
npm:rc-pagination	24	1.09
npm:react-side-effect	24	1.09
npm:remark-footnotes	24	1.09

npm:resolve-global	24	1.09
npm:sitemap	24	1.09
npm:standard	24	1.09
npm:time-grunt	24	1.09
npm:union	24	1.09
npm:watchify	24	1.09
npm:@aws-sdk/middleware-sdk-sts	23	1.04
npm:@braintree/sanitize-url	23	1.04
npm:@commitlint/ensure	23	1.04
npm:@commitlint/format	23	1.04
npm:@commitlint/is-ignored	23	1.04
npm:@commitlint/lint	23	1.04
npm:@commitlint/message	23	1.04
npm:@commitlint/read	23	1.04
npm:@commitlint/rules	23	1.04
npm:@commitlint/to-lines	23	1.04
npm:@commitlint/top-level	23	1.04
npm:@firebase/analytics-compat	23	1.04
npm:@firebase/app-check-compat	23	1.04
npm:@firebase/app-compat	23	1.04
npm:@firebase/auth-compat	23	1.04
npm:@firebase/firestore-compat	23	1.04
npm:@firebase/functions-compat	23	1.04
npm:@firebase/messaging-compat	23	1.04

npm:@firebase/messaging-interop- types	23	1.04
npm:@firebase/performance- compat	23	1.04
npm:@firebase/polyfill	23	1.04
npm:@firebase/remote-config- compat	23	1.04
npm:@firebase/storage-compat	23	1.04
npm:@google-cloud/paginator	23	1.04
npm:@lezer/common	23	1.04
npm:@lezer/lr	23	1.04
npm:@malept/cross-spawn- promise	23	1.04
npm:@radix-ui/react-use-callback- ref	23	1.04
npm:@react-native-community/cli	23	1.04
npm:@restart/context	23	1.04
npm:@swc/core	23	1.04
npm:@swc/core-darwin-arm64	23	1.04
npm:@swc/core-darwin-x64	23	1.04
npm:@swc/core-linux-arm- gnueabihf	23	1.04
npm:@swc/core-linux-arm64-gnu	23	1.04
npm:@swc/core-linux-arm64-musl	23	1.04
npm:@swc/core-linux-x64-gnu	23	1.04

npm:@swc/core-linux-x64-musl	23	1.04
npm:@swc/core-win32-arm64-msvc	23	1.04
npm:@swc/core-win32-ia32-msvc	23	1.04
npm:@swc/core-win32-x64-msvc	23	1.04
npm:@tootallnate/quickjs- emscripten	23	1.04
npm:@types/cookiejar	23	1.04
npm:@types/js-cookie	23	1.04
npm:@types/superagent	23	1.04
npm:@vue/eslint-config-prettier	23	1.04
npm:abstract-leveldown	23	1.04
npm:add-stream	23	1.04
npm:caw	23	1.04
npm:chromium-bidi	23	1.04
npm:css-stringify	23	1.04
npm:csv	23	1.04
npm:diff-match-patch	23	1.04
npm:enzyme-adapter-utils	23	1.04
npm:enzyme-shallow-equal	23	1.04
npm:eslint-config-standard-jsx	23	1.04
npm:eslint-import-resolver- webpack	23	1.04
npm:file-selector	23	1.04
npm:generic-pool	23	1.04
npm:get-proxy	23	1.04

npm:git-hooks-list	23	1.04
npm:gonzales-pe	23	1.04
npm:graphlib	23	1.04
npm:grunt-git-authors	23	1.04
npm:gulp-replace	23	1.04
npm:hast-util-is-element	23	1.04
npm:jodid25519	23	1.04
npm:jpeg-js	23	1.04
npm:micromark-extension-gfm-footnote	23	1.04
npm:mkdirp-promise	23	1.04
npm:mv	23	1.04
npm:mysql2	23	1.04
npm:pino	23	1.04
npm:posthtml	23	1.04
npm:posthtml-parser	23	1.04
npm:posthtml-render	23	1.04
npm:rc-menu	23	1.04
npm:rc-resize-observer	23	1.04
npm:react-inspector	23	1.04
npm:rehype-parse	23	1.04
npm:screenfull	23	1.04
npm:shallow-equal	23	1.04
npm:shasum-object	23	1.04
npm:srcset	23	1.04

npm:strong-log-transformer	23	1.04
npm:tempfile	23	1.04
npm:testswarm	23	1.04
npm:uws	23	1.04
npm:vue-demi	23	1.04
npm:xcode	23	1.04
npm:xml-doc	23	1.04
npm:yamljs	23	1.04
npm:@aws-sdk/token-providers	22	1.00
npm:@aws-sdk/util-endpoints	22	1.00
npm:@babel/plugin-transform-object-assign	22	1.00
npm:@graphql-tools/merge	22	1.00
npm:@graphql-tools/schema	22	1.00
npm:@graphql-tools/utils	22	1.00
npm:@humanwhocodes/gitignore-to-minimatch	22	1.00
npm:@jupyterlab/coreutils	22	1.00
npm:@jupyterlab/services	22	1.00
npm:@mdx-js/util	22	1.00
npm:@npmcli/agent	22	1.00
npm:@puppeteer/browsers	22	1.00
npm:@radix-ui/primitive	22	1.00
npm:@radix-ui/react-compose-refs	22	1.00
npm:@radix-ui/react-context	22	1.00

npm:@radix-ui/react-primitive	22	1.00
npm:@radix-ui/react-slot	22	1.00
npm:@radix-ui/react-use-layout-effect	22	1.00
npm:@storybook/addon-actions	22	1.00
npm:@storybook/addon-links	22	1.00
npm:@types/mdx	22	1.00
npm:@types/offscreencanvas	22	1.00
npm:@vueuse/core	22	1.00
npm:@wry/context	22	1.00
npm:appdirsjs	22	1.00
npm:array.prototype.filter	22	1.00
npm:aststring	22	1.00
npm:bech32	22	1.00
npm:browser-sync-client	22	1.00
npm:canonical-path	22	1.00
npm:chain-function	22	1.00
npm:compression-webpack-plugin	22	1.00
npm:conventional-recommended-bump	22	1.00
npm:date-time	22	1.00
npm:estree-util-visit	22	1.00
npm:ethjs-util	22	1.00
npm:framesync	22	1.00
npm:grunt-contrib-requirejs	22	1.00

npm:html2canvas	22	1.00
npm:lodash.find	22	1.00
npm:lowlight	22	1.00
npm:mdast-squeeze-paragraphs	22	1.00
npm:metro-cache-key	22	1.00
npm:metro-runtime	22	1.00
npm:metro-transform-plugins	22	1.00
npm:metro-transform-worker	22	1.00
npm:null-loader	22	1.00
npm:pg-protocol	22	1.00
npm:phantomjs	22	1.00
npm:postcss-modules	22	1.00
npm:quick-format-unescaped	22	1.00
npm:rbush	22	1.00
npm:rc-checkbox	22	1.00
npm:rc-dropdown	22	1.00
npm:rc-progress	22	1.00
npm:rc-tabs	22	1.00
npm:rc-virtual-list	22	1.00
npm:react-color	22	1.00
npm:react-docgen	22	1.00
npm:react-event-listener	22	1.00
npm:react-hot-loader	22	1.00
npm:react-smooth	22	1.00
npm:react-syntax-highlighter	22	1.00

npm:recharts-scale	22	1.00
npm:redis-errors	22	1.00
npm:remark-squeeze-paragraphs	22	1.00
npm:require-like	22	1.00
npm:sander	22	1.00
npm:sequelize	22	1.00
npm:simple-plist	22	1.00
npm:speedometer	22	1.00
npm:standard-engine	22	1.00
npm:tfunk	22	1.00
npm:toml	22	1.00
npm:urlpattern-polyfill	22	1.00
npm:value-or-promise	22	1.00
npm:vinyl-file	22	1.00
npm:web3-utils	22	1.00
npm:@ant-design/icons-svg	21	0.95
npm:@commitlint/config-validator	21	0.95
npm:@cspotcode/source-map-consumer	21	0.95
npm:@electron/universal	21	0.95
npm:@ethersproject/abi	21	0.95
npm:@ethersproject/abstract-provider	21	0.95
npm:@ethersproject/abstract-signer	21	0.95

npm:@ethersproject/address	21	0.95
npm:@ethersproject/base64	21	0.95
npm:@ethersproject/bignumber	21	0.95
npm:@ethersproject/bytes	21	0.95
npm:@ethersproject/constants	21	0.95
npm:@ethersproject/hash	21	0.95
npm:@ethersproject/keccak256	21	0.95
npm:@ethersproject/logger	21	0.95
npm:@ethersproject/networks	21	0.95
npm:@ethersproject/properties	21	0.95
npm:@ethersproject/rlp	21	0.95
npm:@ethersproject/signing-key	21	0.95
npm:@ethersproject/strings	21	0.95
npm:@ethersproject/transactions	21	0.95
npm:@ethersproject/web	21	0.95
npm:@next/swc-android-arm64	21	0.95
npm:@next/swc-linux-arm- gnueabihf	21	0.95
npm:@react-native-community/cli- plugin-metro	21	0.95
npm:@react-spring/animated	21	0.95
npm:@react-spring/core	21	0.95
npm:@react-spring/shared	21	0.95
npm:@react-spring/types	21	0.95
npm:@sigstore/bundle	21	0.95

npm:@sigstore/sign	21	0.95
npm:@types/babel-types	21	0.95
npm:@types/babylon	21	0.95
npm:@types/classnames	21	0.95
npm:@types/sinon	21	0.95
npm:@vue/eslint-config-typescript	21	0.95
npm:@yarnpkg/parsers	21	0.95
npm:babel-plugin-apply-mdx-type-prop	21	0.95
npm:babel-plugin-extract-import-names	21	0.95
npm:babel-plugin-transform-function-bind	21	0.95
npm:babel-walk	21	0.95
npm:bin-version	21	0.95
npm:bootstrap-vue	21	0.95
npm:bottleneck	21	0.95
npm:btoa-lite	21	0.95
npm:charm	21	0.95
npm:colornames	21	0.95
npm:connect-flash	21	0.95
npm:crypto	21	0.95
npm:deferred-leveldown	21	0.95
npm:deprecated	21	0.95
npm:dir-compare	21	0.95

npm:eslint-friendly-formatter	21	0.95
npm:ethereum-cryptography	21	0.95
npm:ethjs-unit	21	0.95
npm:ext-list	21	0.95
npm:ext-name	21	0.95
npm:faker	21	0.95
npm:fast-equals	21	0.95
npm:filenamify-url	21	0.95
npm:focus-lock	21	0.95
npm:global-tunnel-ng	21	0.95
npm:grunt-karma	21	0.95
npm:gulp-eslint	21	0.95
npm:gulp-less	21	0.95
npm:hermes-parser	21	0.95
npm:humanize-url	21	0.95
npm:install	21	0.95
npm:is-gzip	21	0.95
npm:jose	21	0.95
npm:katex	21	0.95
npm:levelup	21	0.95
npm:localforage	21	0.95
npm:ltgt	21	0.95
npm:maxmin	21	0.95
npm:md5-hex	21	0.95
npm:mdast-util-mdx	21	0.95

npm:merge-options	21	0.95
npm:micromark-extension-mdx-expression	21	0.95
npm:micromark-extension-mdx-jsx	21	0.95
npm:micromark-extension-mdx-md	21	0.95
npm:micromark-extension-mdxjs	21	0.95
npm:micromark-extension-mdxjs-esm	21	0.95
npm:micromark-factory-mdx-expression	21	0.95
npm:micromark-util-events-to-acorn	21	0.95
npm:mnemonist	21	0.95
npm:mobx	21	0.95
npm:node.extend	21	0.95
npm:number-to-bn	21	0.95
npm:orchestrator	21	0.95
npm:passport-oauth2	21	0.95
npm:pino-std-serializers	21	0.95
npm:progress-stream	21	0.95
npm:rc-collapse	21	0.95
npm:rc-dialog	21	0.95
npm:rc-input-number	21	0.95
npm:rc-notification	21	0.95
npm:rc-rate	21	0.95

npm:rc-steps	21	0.95
npm:rc-switch	21	0.95
npm:rc-table	21	0.95
npm:rc-tree-select	21	0.95
npm:rc-upload	21	0.95
npm:react-copy-to-clipboard	21	0.95
npm:react-dnd	21	0.95
npm:react-resize-detector	21	0.95
npm:refractor	21	0.95
npm:require-relative	21	0.95
npm:sequencify	21	0.95
npm:shimmer	21	0.95
npm:should-equal	21	0.95
npm:single-line-log	21	0.95
npm:sorcery	21	0.95
npm:sort-keys-length	21	0.95
npm:stream-consume	21	0.95
npm:strip-url-auth	21	0.95
npm:stylis-rule-sheet	21	0.95
npm:svelte-hmr	21	0.95
npm:thunkify	21	0.95
npm:time-zone	21	0.95
npm:ts-essentials	21	0.95
npm:umask	21	0.95
npm:unist-util-position-from-estree	21	0.95

npm:universal-cookie	21	0.95
npm:url-template	21	0.95
npm:@algolia/cache-browser-local-storage	20	0.91
npm:@algolia/cache-common	20	0.91
npm:@algolia/cache-in-memory	20	0.91
npm:@algolia/client-account	20	0.91
npm:@algolia/client-analytics	20	0.91
npm:@algolia/client-common	20	0.91
npm:@algolia/client-search	20	0.91
npm:@algolia/logger-common	20	0.91
npm:@algolia/logger-console	20	0.91
npm:@algolia/requester-browser-xhr	20	0.91
npm:@algolia/requester-common	20	0.91
npm:@algolia/requester-node-http	20	0.91
npm:@algolia/transporter	20	0.91
npm:@docsearch/react	20	0.91
npm:@es-joy/jsdoccomment	20	0.91
npm:@fortawesome/free-regular-svg-icons	20	0.91
npm:@gulp-sourcemaps/map-sources	20	0.91
npm:@jupyterlab/application	20	0.91
npm:@noble/hashes	20	0.91

npm:@npmcli/ci-detect	20	0.91
npm:@npmcli/disparity-colors	20	0.91
npm:@radix-ui/react-presence	20	0.91
npm:@schematics/update	20	0.91
npm:@storybook/react	20	0.91
npm:@turf/invariant	20	0.91
npm:@types/hammerjs	20	0.91
npm:@types/sax	20	0.91
npm:@types/triple-beam	20	0.91
npm:absolute-path	20	0.91
npm:atomic-sleep	20	0.91
npm:babel-plugin-syntax-do- expressions	20	0.91
npm:babel-plugin-syntax-function- bind	20	0.91
npm:babel-plugin-transform-do- expressions	20	0.91
npm:bin-version-check	20	0.91
npm:codecov	20	0.91
npm:coffeescript	20	0.91
npm:commitplease	20	0.91
npm:debug-fabulous	20	0.91
npm:decimal.js-light	20	0.91
npm:diagnostics	20	0.91
npm:dotignore	20	0.91

npm:element-ui	20	0.91
npm:env-variable	20	0.91
npm:estree-util-attach-comments	20	0.91
npm:fast-redact	20	0.91
npm:feed	20	0.91
npm:fp-ts	20	0.91
npm:grunt-zip	20	0.91
npm:gulp-babel	20	0.91
npm:hast-util-to-estree	20	0.91
npm:hexoid	20	0.91
npm:i	20	0.91
npm:imagemin	20	0.91
npm:indefinite-observable	20	0.91
npm:is-dom	20	0.91
npm:is-invalid-path	20	0.91
npm:is2	20	0.91
npm:javascript-natural-sort	20	0.91
npm:jsdoc-type-pratt-parser	20	0.91
npm:jshint-stylish	20	0.91
npm:klaw-sync	20	0.91
npm:libnpmdiff	20	0.91
npm:libnpmfund	20	0.91
npm:libnpm-pack	20	0.91
npm:libnpm-version	20	0.91
npm:load-script	20	0.91

npm:lodash._arraycopy	20	0.91
npm:lodash._arrayeach	20	0.91
npm:lodash._basefor	20	0.91
npm:lodash.padstart	20	0.91
npm:lru_map	20	0.91
npm:markdown-extensions	20	0.91
npm:memoizerific	20	0.91
npm:module-alias	20	0.91
npm:mout	20	0.91
npm:nock	20	0.91
npm:nugget	20	0.91
npm:password-prompt	20	0.91
npm:purgecss	20	0.91
npm:rc-overflow	20	0.91
npm:react-colorful	20	0.91
npm:react-dnd-html5-backend	20	0.91
npm:react-il8next	20	0.91
npm:react-leaflet	20	0.91
npm:redux-devtools-extension	20	0.91
npm:run-sequence	20	0.91
npm:safe-json-stringify	20	0.91
npm:semver-truncate	20	0.91
npm:should-format	20	0.91
npm:should-type	20	0.91
npm:sonic-boom	20	0.91

npm:sortablejs	20	0.91
npm:svelte-preprocess	20	0.91
npm:tcp-port-used	20	0.91
npm:unicode-trie	20	0.91
npm:universal-analytics	20	0.91
npm:use-memo-one	20	0.91
npm:vscode-uri	20	0.91
npm:web3	20	0.91
npm:wmf	20	0.91
npm:write-pkg	20	0.91
npm:@aws-crypto/crc32	19	0.86
npm:@docsearch/css	19	0.86
npm:@firebase/app-check-types	19	0.86
npm:@google-cloud/common	19	0.86
npm:@google-cloud/storage	19	0.86
npm:@hutson/parse-repository-url	19	0.86
npm:@jsdoc/salty	19	0.86
npm:@pnpm/network.ca-file	19	0.86
npm:@pnpm/npm-conf	19	0.86
npm:@radix-ui/react-arrow	19	0.86
npm:@radix-ui/react-direction	19	0.86
npm:@radix-ui/react-id	19	0.86
npm:@radix-ui/react-popover	19	0.86
npm:@radix-ui/react-portal	19	0.86

npm:@radix-ui/react-use-controllable-state	19	0.86
npm:@radix-ui/react-use-escape-keydown	19	0.86
npm:@radix-ui/react-use-rect	19	0.86
npm:@radix-ui/react-use-size	19	0.86
npm:@radix-ui/rect	19	0.86
npm:@react-spring/web	19	0.86
npm:@rollup/plugin-terser	19	0.86
npm:@rollup/plugin-typescript	19	0.86
npm:@storybook/node-logger	19	0.86
npm:@sveltejs/vite-plugin-svelte	19	0.86
npm:@turf/area	19	0.86
npm:@turf/centroid	19	0.86
npm:@types/concat-stream	19	0.86
npm:@types/pbkdf2	19	0.86
npm:@types/react-test-renderer	19	0.86
npm:@types/secp256k1	19	0.86
npm:@types/seedrandom	19	0.86
npm:@types/vfile-message	19	0.86
npm:abs-svg-path	19	0.86
npm:array.prototype.map	19	0.86
npm:bare-events	19	0.86
npm:base-64	19	0.86
npm:bip66	19	0.86

npm:bulma	19	0.86
npm:cbor	19	0.86
npm:child_process	19	0.86
npm:debug-log	19	0.86
npm:editions	19	0.86
npm:electron-download	19	0.86
npm:eslint-plugin-cypress	19	0.86
npm:eslint-plugin-n	19	0.86
npm:estree-util-build-jsx	19	0.86
npm:eval	19	0.86
npm:file-system-cache	19	0.86
npm:find-node-modules	19	0.86
npm:grunt-bowercopy	19	0.86
npm:gulp-plumber	19	0.86
npm:hermes-estree	19	0.86
npm:is-valid-path	19	0.86
npm:isomorphic-unfetch	19	0.86
npm:iterate-iterator	19	0.86
npm:iterate-value	19	0.86
npm:jschardet	19	0.86
npm:koa	19	0.86
npm:koa-compose	19	0.86
npm:libnpmexec	19	0.86
npm:lodash.isnative	19	0.86
npm:lodash.shimkeys	19	0.86

npm:lodash.debounce	19	0.86
npm:lodash.values	19	0.86
npm:map-limit	19	0.86
npm:map-or-similar	19	0.86
npm:markdown-to-jsx	19	0.86
npm:monaco-editor	19	0.86
npm:named-placeholders	19	0.86
npm:npm-lifecycle	19	0.86
npm:os-filter-obj	19	0.86
npm:parse-svg-path	19	0.86
npm:path-posix	19	0.86
npm:portal-vue	19	0.86
npm:postcss-less	19	0.86
npm:raf-schd	19	0.86
npm:rc-mentions	19	0.86
npm:react-beautiful-dnd	19	0.86
npm:reserved-words	19	0.86
npm:rtl-css-js	19	0.86
npm:semantic-release	19	0.86
npm:seq-queue	19	0.86
npm:simple-git	19	0.86
npm:specificity	19	0.86
npm:stack-generator	19	0.86
npm:stacktrace-gps	19	0.86
npm:stacktrace-js	19	0.86

npm:sugarss	19	0.86
npm:to-no-case	19	0.86
npm:to-space-case	19	0.86
npm:trim-off-newlines	19	0.86
npm:url-set-query	19	0.86
npm:vscode-jsonrpc	19	0.86
npm:vscode-languageserver- protocol	19	0.86
npm:yarn	19	0.86
npm:z-schema	19	0.86
npm:@algolia/autocomplete-core	18	0.82
npm:@algolia/autocomplete- preset-algolia	18	0.82
npm:@algolia/autocomplete-shared	18	0.82
npm:@algolia/client- personalization	18	0.82
npm:@aws-sdk/is-array-buffer	18	0.82
npm:@aws-sdk/util-buffer-from	18	0.82
npm:@babel/plugin-external- helpers	18	0.82
npm:@emotion/styled-base	18	0.82
npm:@headlessui/react	18	0.82
npm:@jupyterlab/observables	18	0.82
npm:@lerna/create	18	0.82
npm:@mongodb-js/saslprep	18	0.82

npm:@nrwl/tao	18	0.82
npm:@nuxt/opencollective	18	0.82
npm:@octokit/plugin-enterprise-rest	18	0.82
npm:@parcel/fs	18	0.82
npm:@parcel/logger	18	0.82
npm:@parcel/Utils	18	0.82
npm:@parcel/workers	18	0.82
npm:@sentry/browser	18	0.82
npm:@sentry/node	18	0.82
npm:@storybook/addons	18	0.82
npm:@storybook/channels	18	0.82
npm:@storybook/client-logger	18	0.82
npm:@storybook/components	18	0.82
npm:@storybook/core-events	18	0.82
npm:@storybook/csf	18	0.82
npm:@tokenizer/token	18	0.82
npm:@types/underscore	18	0.82
npm:@types/use-sync-external-store	18	0.82
npm:@types/vfile	18	0.82
npm:apollo-link	18	0.82
npm:apollo-utilities	18	0.82
npm:app-root-dir	18	0.82
npm:array-find	18	0.82

npm:autolinker	18	0.82
npm:babel	18	0.82
npm:bin-check	18	0.82
npm:bit-twiddle	18	0.82
npm:buffer-to-arraybuffer	18	0.82
npm:chai-http	18	0.82
npm:connect-mongo	18	0.82
npm:css-functions-list	18	0.82
npm:css-line-break	18	0.82
npm:csv-generate	18	0.82
npm:csvtojson	18	0.82
npm:cytoscape	18	0.82
npm:cz-conventional-changelog	18	0.82
npm:dagre	18	0.82
npm:data-view-buffer	18	0.82
npm:data-view-byte-length	18	0.82
npm:data-view-byte-offset	18	0.82
npm:devlop	18	0.82
npm:eol	18	0.82
npm:eslint-config-google	18	0.82
npm:estree-util-to-js	18	0.82
npm:ethereum-bloom-filters	18	0.82
npm:express-validator	18	0.82
npm:grunt-contrib-less	18	0.82
npm:grunt-sass	18	0.82

npm:http-assert	18	0.82
npm:ioredis	18	0.82
npm:is-data-view	18	0.82
npm:karma-cli	18	0.82
npm:lazy-universal-dotenv	18	0.82
npm:level-errors	18	0.82
npm:level-iterator-stream	18	0.82
npm:lodash._baseclone	18	0.82
npm:lodash.capitalize	18	0.82
npm:lodash.isempty	18	0.82
npm:mock-fs	18	0.82
npm:multibase	18	0.82
npm:multihashes	18	0.82
npm:normalize-selector	18	0.82
npm:nx	18	0.82
npm:only	18	0.82
npm:p-waterfall	18	0.82
npm:parse-bmfont-ascii	18	0.82
npm:parse-numeric-range	18	0.82
npm:peek-readable	18	0.82
npm:postcss-cli	18	0.82
npm:promise.allsettled	18	0.82
npm:react-addons-test-utils	18	0.82
npm:react-clientside-effect	18	0.82
npm:react-focus-lock	18	0.82

npm:react-native-screens	18	0.82
npm:reactstrap	18	0.82
npm:remark-slug	18	0.82
npm:rtlcss	18	0.82
npm:secure-compare	18	0.82
npm:semantic-ui-react	18	0.82
npm:solc	18	0.82
npm:speed-measure-webpack- plugin	18	0.82
npm:stackblur-canvas	18	0.82
npm:stream-parser	18	0.82
npm:stream-transform	18	0.82
npm:telejson	18	0.82
npm:token-types	18	0.82
npm:use-subscription	18	0.82
npm:vali-date	18	0.82
npm:vscode-languageserver- textdocument	18	0.82
npm:w3c-keyname	18	0.82
npm:web3-eth-abi	18	0.82
npm:webpack-node-externals	18	0.82
npm:word	18	0.82
npm:xhr-request	18	0.82
npm:@achrinza/node-ipc	17	0.77
npm:@ant-design/react-slick	17	0.77

npm:@aws-sdk/abort-controller	17	0.77
npm:@aws-sdk/config-resolver	17	0.77
npm:@aws-sdk/credential- provider-imds	17	0.77
npm:@aws-sdk/fetch-http-handler	17	0.77
npm:@aws-sdk/hash-node	17	0.77
npm:@aws-sdk/invalid-dependency	17	0.77
npm:@aws-sdk/middleware- content-length	17	0.77
npm:@aws-sdk/middleware-retry	17	0.77
npm:@aws-sdk/middleware-serde	17	0.77
npm:@aws-sdk/middleware-stack	17	0.77
npm:@aws-sdk/node-config- provider	17	0.77
npm:@aws-sdk/node-http-handler	17	0.77
npm:@aws-sdk/property-provider	17	0.77
npm:@aws-sdk/protocol-http	17	0.77
npm:@aws-sdk/querystring-builder	17	0.77
npm:@aws-sdk/querystring-parser	17	0.77
npm:@aws-sdk/service-error- classification	17	0.77
npm:@aws-sdk/shared-ini-file- loader	17	0.77
npm:@aws-sdk/signature-v4	17	0.77
npm:@aws-sdk/smithy-client	17	0.77

npm:@aws-sdk/url-parser	17	0.77
npm:@aws-sdk/util-body-length-browser	17	0.77
npm:@aws-sdk/util-body-length-node	17	0.77
npm:@aws-sdk/util-hex-encoding	17	0.77
npm:@aws-sdk/util-uri-escape	17	0.77
npm:@ethersproject/basex	17	0.77
npm:@ethersproject/contracts	17	0.77
npm:@ethersproject/hdnode	17	0.77
npm:@ethersproject/json-wallets	17	0.77
npm:@ethersproject/pbkdf2	17	0.77
npm:@ethersproject/providers	17	0.77
npm:@ethersproject/random	17	0.77
npm:@ethersproject/sha2	17	0.77
npm:@ethersproject/units	17	0.77
npm:@ethersproject/wallet	17	0.77
npm:@ethersproject/wordlists	17	0.77
npm:@jupyter-widgets/base	17	0.77
npm:@malept/flatpak-bundler	17	0.77
npm:@mdi/font	17	0.77
npm:@next/swc-android-arm-eabi	17	0.77
npm:@node-ipc/js-queue	17	0.77
npm:@opentelemetry/api	17	0.77
npm:@radix-ui/number	17	0.77

npm:@radix-ui/react-collection	17	0.77
npm:@radix-ui/react-dismissable-layer	17	0.77
npm:@radix-ui/react-focus-guards	17	0.77
npm:@radix-ui/react-focus-scope	17	0.77
npm:@radix-ui/react-use-previous	17	0.77
npm:@radix-ui/react-visually-hidden	17	0.77
npm:@react-native-community/cli-clean	17	0.77
npm:@react-native-community/cli-config	17	0.77
npm:@react-native-community/cli-doctor	17	0.77
npm:@react-native/normalize-color	17	0.77
npm:@scure/base	17	0.77
npm:@semantic-release/github	17	0.77
npm:@storybook/router	17	0.77
npm:@storybook/theming	17	0.77
npm:@types/bluebird	17	0.77
npm:@types/d3-delaunay	17	0.77
npm:@types/inquirer	17	0.77
npm:@types/plist	17	0.77
npm:@types/react-native	17	0.77
npm:@types/sass	17	0.77

npm:@types/through	17	0.77
npm:@types/verror	17	0.77
npm:@vueuse/shared	17	0.77
npm:@xobotyi/scrollbar-width	17	0.77
npm:addressparser	17	0.77
npm:ansi-sequence-parser	17	0.77
npm:asn1js	17	0.77
npm:astw	17	0.77
npm:bmp-js	17	0.77
npm:brotli	17	0.77
npm:bundle-require	17	0.77
npm:bytebuffer	17	0.77
npm:canvas	17	0.77
npm:cids	17	0.77
npm:code-block-writer	17	0.77
npm:condense-newlines	17	0.77
npm:d3-sankey	17	0.77
npm:deasync	17	0.77
npm:defu	17	0.77
npm:devalue	17	0.77
npm:dmg-license	17	0.77
npm:dnd-core	17	0.77
npm:dottie	17	0.77
npm:drbg.js	17	0.77
npm:ecstatic	17	0.77

npm:electron-packager	17	0.77
npm:emoticon	17	0.77
npm:envify	17	0.77
npm:es-object-atoms	17	0.77
npm:es6-shim	17	0.77
npm:expo	17	0.77
npm:exports-loader	17	0.77
npm:fast-json-parse	17	0.77
npm:fast-shallow-equal	17	0.77
npm:fastest-stable-stringify	17	0.77
npm:fetch-retry	17	0.77
npm:focus-trap	17	0.77
npm:formik	17	0.77
npm:get-east-asian-width	17	0.77
npm:gl-mat4	17	0.77
npm:glob-promise	17	0.77
npm:gsl-tokenizer	17	0.77
npm:gslify	17	0.77
npm:grunt-contrib-htmlmin	17	0.77
npm:hey-listen	17	0.77
npm:http-response-object	17	0.77
npm:iconv-corefoundation	17	0.77
npm:is-whitespace	17	0.77
npm:joycon	17	0.77
npm:jquery-ui	17	0.77

npm:knex	17	0.77
npm:level-codec	17	0.77
npm:level-supports	17	0.77
npm:lexical-scope	17	0.77
npm:load-yaml-file	17	0.77
npm:lodash.groupby	17	0.77
npm:material-colors	17	0.77
npm:mathjs	17	0.77
npm:mobx-react	17	0.77
npm:multicodec	17	0.77
npm:nano-css	17	0.77
npm:napi-macros	17	0.77
npm:normalize-scroll-left	17	0.77
npm:octokit-pagination-methods	17	0.77
npm:os-shim	17	0.77
npm:preferred-pm	17	0.77
npm:pretty	17	0.77
npm:prism-react-renderer	17	0.77
npm:propagate	17	0.77
npm:rc-field-form	17	0.77
npm:rc-picker	17	0.77
npm:react-tooltip	17	0.77
npm:react-use	17	0.77
npm:react-window	17	0.77
npm:readable-web-to-node-stream	17	0.77

npm:reading-time	17	0.77
npm:regl	17	0.77
npm:remark-emoji	17	0.77
npm:remark-frontmatter	17	0.77
npm:retry-as-promised	17	0.77
npm:revalidator	17	0.77
npm:rgbcolor	17	0.77
npm:scmp	17	0.77
npm:scryptsy	17	0.77
npm:set-harmonic-interval	17	0.77
npm:should-type-adaptors	17	0.77
npm:should-util	17	0.77
npm:smob	17	0.77
npm:spawn-sync	17	0.77
npm:store2	17	0.77
npm:string-similarity	17	0.77
npm:strip-bom-buf	17	0.77
npm:strtok3	17	0.77
npm:svg-arc-to-cubic-bezier	17	0.77
npm:sync-request	17	0.77
npm:toposort-class	17	0.77
npm:transformers	17	0.77
npm:ts-easing	17	0.77
npm:unicorn-magic	17	0.77
npm:url-regex	17	0.77

npm:urlgrey	17	0.77
npm:vinyl-source-stream	17	0.77
npm:vm2	17	0.77
npm:vue-i18n	17	0.77
npm:web-worker	17	0.77
npm:web3-core	17	0.77
npm:web3-eth	17	0.77
npm:web3-eth-accounts	17	0.77
npm:web3-eth-contract	17	0.77
npm:web3-eth-iban	17	0.77
npm:web3-eth-personal	17	0.77
npm:web3-net	17	0.77
npm:which-pm	17	0.77
npm:wkx	17	0.77
npm:@aws-sdk/client-sso-oidc	16	0.73
npm:@base2/pretty-print-object	16	0.73
npm:@csstools/css-parser- algorithms	16	0.73
npm:@csstools/css-tokenizer	16	0.73
npm:@csstools/media-query-list- parser	16	0.73
npm:@ethersproject/solidity	16	0.73
npm:@gulp-sourcemaps/identity- map	16	0.73
npm:@jupyterlab/builder	16	0.73

npm:@lerna/child-process	16	0.73
npm:@lumino/widgets	16	0.73
npm:@noble/secp256k1	16	0.73
npm:@nrwl/devkit	16	0.73
npm:@peculiar/asn1-schema	16	0.73
npm:@peculiar/json-schema	16	0.73
npm:@peculiar/webcrypto	16	0.73
npm:@semantic-release/error	16	0.73
npm:@semantic-release/npm	16	0.73
npm:@sentry/tracing	16	0.73
npm:@smithy/types	16	0.73
npm:@storybook/addon-essentials	16	0.73
npm:@storybook/core-client	16	0.73
npm:@storybook/core-common	16	0.73
npm:@swc/types	16	0.73
npm:@types/accepts	16	0.73
npm:@types/marked	16	0.73
npm:@types/npmlog	16	0.73
npm:@types/pretty-hrtime	16	0.73
npm:@types/web-bluetooth	16	0.73
npm:@vueuse/metadata	16	0.73
npm:almost-equal	16	0.73
npm:amqplib	16	0.73
npm:ansistyles	16	0.73
npm:argv	16	0.73

npm:array.prototype.findlast	16	0.73
npm:array.prototype.toreversed	16	0.73
npm:babel-plugin-transform-vue- jsx	16	0.73
npm:backoff	16	0.73
npm:bin-build	16	0.73
npm:bin-wrapper	16	0.73
npm:canvg	16	0.73
npm:class-is	16	0.73
npm:clone-regexp	16	0.73
npm:cpy	16	0.73
npm:current-script-polyfill	16	0.73
npm:dag-map	16	0.73
npm:dataloader	16	0.73
npm:decache	16	0.73
npm:deglob	16	0.73
npm:deprecated-react-native-prop- types	16	0.73
npm:downshift	16	0.73
npm:es5-shim	16	0.73
npm:es6-templates	16	0.73
npm:eslint-plugin-unicorn	16	0.73
npm:eth-lib	16	0.73
npm:ethereumjs-abi	16	0.73
npm:execall	16	0.73

npm:filled-array	16	0.73
npm:find-yarn-workspace-root2	16	0.73
npm:firebase-admin	16	0.73
npm:fontfaceobserver	16	0.73
npm:get-user-locale	16	0.73
npm:gsl-token-string	16	0.73
npm:gslify-bundle	16	0.73
npm:gslify-deps	16	0.73
npm:google-protobuf	16	0.73
npm:grunt-contrib-sass	16	0.73
npm:gulp-if	16	0.73
npm:hogan.js	16	0.73
npm:http-basic	16	0.73
npm:http-https	16	0.73
npm:imports-loader	16	0.73
npm:io-ts	16	0.73
npm:jest-junit	16	0.73
npm:json2csv	16	0.73
npm:jspdf	16	0.73
npm:karma-spec-reporter	16	0.73
npm:koa-convert	16	0.73
npm:load-bmfont	16	0.73
npm:lodash.castarray	16	0.73
npm:lodash.istypedarray	16	0.73
npm:metro-file-map	16	0.73

npm:mini-svg-data-uri	16	0.73
npm:node-html-parser	16	0.73
npm:node-schedule	16	0.73
npm:nofilter	16	0.73
npm:oboe	16	0.73
npm:parse-bmfont-binary	16	0.73
npm:parse-bmfont-xml	16	0.73
npm:phin	16	0.73
npm:pkg-config	16	0.73
npm:prettier-plugin-svelte	16	0.73
npm:process-warning	16	0.73
npm:promise.prototype.finally	16	0.73
npm:prompt	16	0.73
npm:proto3-json-serializer	16	0.73
npm:pvtsutils	16	0.73
npm:pvutils	16	0.73
npm:rc-image	16	0.73
npm:rc-textarea	16	0.73
npm:react-app-rewired	16	0.73
npm:react-datepicker	16	0.73
npm:react-element-to-jsx-string	16	0.73
npm:react-motion	16	0.73
npm:react-native-gesture-handler	16	0.73
npm:react-reconciler	16	0.73
npm:react-router-config	16	0.73

npm:react-universal-interface	16	0.73
npm:react-virtualized	16	0.73
npm:redux-mock-store	16	0.73
npm:redux-saga	16	0.73
npm:require-package-name	16	0.73
npm:rollup-plugin-typescript2	16	0.73
npm:run-node	16	0.73
npm:signale	16	0.73
npm:silent-error	16	0.73
npm:simple-is	16	0.73
npm:standard-as-callback	16	0.73
npm:stringify-package	16	0.73
npm:svelte-check	16	0.73
npm:temp-write	16	0.73
npm:test-value	16	0.73
npm:then-request	16	0.73
npm:util-extend	16	0.73
npm:vuetify-loader	16	0.73
npm:web3-core-helpers	16	0.73
npm:web3-core-method	16	0.73
npm:web3-core-subscriptions	16	0.73
npm:web3-providers-http	16	0.73
npm:web3-providers-ipc	16	0.73
npm:web3-providers-ws	16	0.73
npm:web3-shh	16	0.73

npm:webcrypto-core	16	0.73
npm:xhr-request-promise	16	0.73
npm:xml-parse-from-string	16	0.73
npm:yazl	16	0.73
npm:@aws-sdk/client-cognito-identity	15	0.68
npm:@aws-sdk/credential-provider-cognito-identity	15	0.68
npm:@choojs/findup	15	0.68
npm:@firebase/installations-compat	15	0.68
npm:@fortawesome/free-brands-svg-icons	15	0.68
npm:@ljharb/through	15	0.68
npm:@lumino/disposable	15	0.68
npm:@lumino/messaging	15	0.68
npm:@npmcli/query	15	0.68
npm:@pkgr/core	15	0.68
npm:@pnpm/config.env-replace	15	0.68
npm:@radix-ui/react-roving-focus	15	0.68
npm:@react-navigation/native	15	0.68
npm:@react-spring/rafz	15	0.68
npm:@scure/bip39	15	0.68
npm:@semantic-release/commit-analyzer	15	0.68

npm:@semantic-release/release- notes-generator	15	0.68
npm:@smithy/protocol-http	15	0.68
npm:@storybook/channel- postmessage	15	0.68
npm:@storybook/client-api	15	0.68
npm:@types/bson	15	0.68
npm:@types/d3-voronoi	15	0.68
npm:@types/lodash-es	15	0.68
npm:@types/lru-cache	15	0.68
npm:@types/mongodb	15	0.68
npm:@types/passport	15	0.68
npm:@types/react-syntax- highlighter	15	0.68
npm:@types/three	15	0.68
npm:@zkochan/js-yaml	15	0.68
npm:argv-formatter	15	0.68
npm:author-regex	15	0.68
npm:ava	15	0.68
npm:babel-plugin-transform- decorators-legacy	15	0.68
npm:babel-preset-expo	15	0.68
npm:brcast	15	0.68
npm:bs-snippet-injector	15	0.68
npm:buffer-more-ints	15	0.68

npm:chromedriver	15	0.68
npm:clamp	15	0.68
npm:color-space	15	0.68
npm:console-stream	15	0.68
npm:conventional-commit-types	15	0.68
npm:copy-text-to-clipboard	15	0.68
npm:cron-parser	15	0.68
npm:css-modules-loader-core	15	0.68
npm:csv-parser	15	0.68
npm:datatables.net	15	0.68
npm:deep-freeze	15	0.68
npm:delay	15	0.68
npm:dom-event-types	15	0.68
npm:dtype	15	0.68
npm:dup	15	0.68
npm:env-ci	15	0.68
npm:enzyme-to-json	15	0.68
npm:eslint-plugin-eslint-comments	15	0.68
npm:eta	15	0.68
npm:eventsource-polyfill	15	0.68
npm:exif-parser	15	0.68
npm:extract-files	15	0.68
npm:flatten-vertex-data	15	0.68
npm:flow-bin	15	0.68
npm:git-log-parser	15	0.68

npm:gsls-inject-defines	15	0.68
npm:gsls-resolve	15	0.68
npm:gsls-token-assignments	15	0.68
npm:gsls-token-defines	15	0.68
npm:gsls-token-depth	15	0.68
npm:gsls-token-descope	15	0.68
npm:gsls-token-inject-block	15	0.68
npm:gsls-token-properties	15	0.68
npm:gsls-token-scope	15	0.68
npm:gsls-token-whitespace-trim	15	0.68
npm:gulp-connect	15	0.68
npm:hast-util-to-html	15	0.68
npm:home-path	15	0.68
npm:htmlnano	15	0.68
npm:http	15	0.68
npm:infinity-agent	15	0.68
npm:intl-messageformat	15	0.68
npm:issue-parser	15	0.68
npm:java-properties	15	0.68
npm:karma-ng-html2js- preprocessor	15	0.68
npm:level	15	0.68
npm:libbase64	15	0.68
npm:libmime	15	0.68
npm:libqp	15	0.68

npm:lodash.isundefined	15	0.68
npm:logalot	15	0.68
npm:long-timeout	15	0.68
npm:lpad-align	15	0.68
npm:material-ui	15	0.68
npm:memory-cache	15	0.68
npm:mixme	15	0.68
npm:nano-json-stream-parser	15	0.68
npm:node-cache	15	0.68
npm:npm-path	15	0.68
npm:npm-which	15	0.68
npm:objectorarray	15	0.68
npm:omggif	15	0.68
npm:openai	15	0.68
npm:optipng-bin	15	0.68
npm:pad-left	15	0.68
npm:parse-author	15	0.68
npm:parse-cache-control	15	0.68
npm:patch-package	15	0.68
npm:pdfjs-dist	15	0.68
npm:pegjs	15	0.68
npm:playwright-core	15	0.68
npm:popmotion	15	0.68
npm:precond	15	0.68
npm:react-native-reanimated	15	0.68

npm:react-popover-tooltip	15	0.68
npm:react-spring	15	0.68
npm:reactcss	15	0.68
npm:remove-accent	15	0.68
npm:resolve-path	15	0.68
npm:rollup-plugin-postcss	15	0.68
npm:rollup-plugin-visualizer	15	0.68
npm:rtl-detect	15	0.68
npm:rustbn.js	15	0.68
npm:serve	15	0.68
npm:servify	15	0.68
npm:sha1	15	0.68
npm:shx	15	0.68
npm:sort-asc	15	0.68
npm:sort-desc	15	0.68
npm:sort-object	15	0.68
npm:spawn-error-forwarder	15	0.68
npm:squeak	15	0.68
npm:style-value-types	15	0.68
npm:sum-up	15	0.68
npm:swarm-js	15	0.68
npm:ts-toolbelt	15	0.68
npm:unixify	15	0.68
npm:validate.io-array	15	0.68
npm:validate.io-integer	15	0.68

npm:validate.io-number	15	0.68
npm:vue-cli-plugin-vuetify	15	0.68
npm:vue-functional-data-merge	15	0.68
npm:vue-tsc	15	0.68
npm:vuedraggable	15	0.68
npm:web3-bzz	15	0.68
npm:web3-core-promievent	15	0.68
npm:web3-core-requestmanager	15	0.68
npm:web3-eth-ens	15	0.68
npm:webpack-assets-manifest	15	0.68
npm:yaml-ast-parser	15	0.68
npm:@apidevtools/json-schema-ref-parser	14	0.64
npm:@apollo/client	14	0.64
npm:@apollo/protobufjs	14	0.64
npm:@aws-sdk/util-config-provider	14	0.64
npm:@aws-sdk/util-defaults-mode-browser	14	0.64
npm:@aws-sdk/util-defaults-mode-node	14	0.64
npm:@aws-sdk/util-middleware	14	0.64
npm:@chakra-ui/accordion	14	0.64
npm:@chakra-ui/alert	14	0.64
npm:@chakra-ui/anatomy	14	0.64
npm:@chakra-ui/avatar	14	0.64

npm:@chakra-ui/breadcrumb	14	0.64
npm:@chakra-ui/button	14	0.64
npm:@chakra-ui/checkbox	14	0.64
npm:@chakra-ui/clickable	14	0.64
npm:@chakra-ui/close-button	14	0.64
npm:@chakra-ui/color-mode	14	0.64
npm:@chakra-ui/control-box	14	0.64
npm:@chakra-ui/counter	14	0.64
npm:@chakra-ui/css-reset	14	0.64
npm:@chakra-ui/descendant	14	0.64
npm:@chakra-ui/editable	14	0.64
npm:@chakra-ui/focus-lock	14	0.64
npm:@chakra-ui/form-control	14	0.64
npm:@chakra-ui/hooks	14	0.64
npm:@chakra-ui/icon	14	0.64
npm:@chakra-ui/image	14	0.64
npm:@chakra-ui/input	14	0.64
npm:@chakra-ui/layout	14	0.64
npm:@chakra-ui/live-region	14	0.64
npm:@chakra-ui/media-query	14	0.64
npm:@chakra-ui/menu	14	0.64
npm:@chakra-ui/modal	14	0.64
npm:@chakra-ui/number-input	14	0.64
npm:@chakra-ui/pin-input	14	0.64
npm:@chakra-ui/popover	14	0.64

npm:@chakra-ui/popper	14	0.64
npm:@chakra-ui/portal	14	0.64
npm:@chakra-ui/progress	14	0.64
npm:@chakra-ui/provider	14	0.64
npm:@chakra-ui/radio	14	0.64
npm:@chakra-ui/react	14	0.64
npm:@chakra-ui/react-env	14	0.64
npm:@chakra-ui/react-utils	14	0.64
npm:@chakra-ui/select	14	0.64
npm:@chakra-ui/skeleton	14	0.64
npm:@chakra-ui/slider	14	0.64
npm:@chakra-ui/spinner	14	0.64
npm:@chakra-ui/stat	14	0.64
npm:@chakra-ui/styled-system	14	0.64
npm:@chakra-ui/switch	14	0.64
npm:@chakra-ui/system	14	0.64
npm:@chakra-ui/table	14	0.64
npm:@chakra-ui/tabs	14	0.64
npm:@chakra-ui/tag	14	0.64
npm:@chakra-ui/textarea	14	0.64
npm:@chakra-ui/theme	14	0.64
npm:@chakra-ui/theme-tools	14	0.64
npm:@chakra-ui/toast	14	0.64
npm:@chakra-ui/tooltip	14	0.64
npm:@chakra-ui/transition	14	0.64

npm:@chakra-ui/utils	14	0.64
npm:@chakra-ui/visually-hidden	14	0.64
npm:@codemirror/state	14	0.64
npm:@codemirror/view	14	0.64
npm:@date-io/date-fns	14	0.64
npm:@docusaurus/types	14	0.64
npm:@egjs/hammerjs	14	0.64
npm:@expo/vector-icons	14	0.64
npm:@graphql-tools/load	14	0.64
npm:@jsdevtools/ono	14	0.64
npm:@juggle/resize-observer	14	0.64
npm:@jupyterlab/apputils	14	0.64
npm:@jupyterlab/settingregistry	14	0.64
npm:@lumino/algorithm	14	0.64
npm:@lumino/commands	14	0.64
npm:@lumino/coreutils	14	0.64
npm:@lumino/signaling	14	0.64
npm:@material/animation	14	0.64
npm:@material/base	14	0.64
npm:@material/ripple	14	0.64
npm:@material/theme	14	0.64
npm:@react-navigation/core	14	0.64
npm:@scure/bip32	14	0.64
npm:@sindresorhus/merge-streams	14	0.64
npm:@smithy/abort-controller	14	0.64

npm:@smithy/config-resolver	14	0.64
npm:@smithy/credential-provider- imds	14	0.64
npm:@smithy/eventstream-codec	14	0.64
npm:@smithy/fetch-http-handler	14	0.64
npm:@smithy/hash-node	14	0.64
npm:@smithy/invalid-dependency	14	0.64
npm:@smithy/is-array-buffer	14	0.64
npm:@smithy/middleware-content- length	14	0.64
npm:@smithy/middleware- endpoint	14	0.64
npm:@smithy/middleware-retry	14	0.64
npm:@smithy/middleware-serde	14	0.64
npm:@smithy/middleware-stack	14	0.64
npm:@smithy/node-config-provider	14	0.64
npm:@smithy/node-http-handler	14	0.64
npm:@smithy/property-provider	14	0.64
npm:@smithy/querystring-builder	14	0.64
npm:@smithy/querystring-parser	14	0.64
npm:@smithy/service-error- classification	14	0.64
npm:@smithy/shared-ini-file-loader	14	0.64
npm:@smithy/signature-v4	14	0.64
npm:@smithy/smithy-client	14	0.64

npm:@smithy/url-parser	14	0.64
npm:@smithy/util-base64	14	0.64
npm:@smithy/util-body-length-browser	14	0.64
npm:@smithy/util-body-length-node	14	0.64
npm:@smithy/util-buffer-from	14	0.64
npm:@smithy/util-config-provider	14	0.64
npm:@smithy/util-defaults-mode-browser	14	0.64
npm:@smithy/util-defaults-mode-node	14	0.64
npm:@smithy/util-hex-encoding	14	0.64
npm:@smithy/util-middleware	14	0.64
npm:@smithy/util-retry	14	0.64
npm:@smithy/util-stream	14	0.64
npm:@smithy/util-uri-escape	14	0.64
npm:@smithy/util-utf8	14	0.64
npm:@storybook/addon-controls	14	0.64
npm:@storybook/api	14	0.64
npm:@storybook/core-server	14	0.64
npm:@sveltejs/kit	14	0.64
npm:@tailwindcss/typography	14	0.64
npm:@ts-morph/common	14	0.64
npm:@tweenjs/tween.js	14	0.64

npm:@types/chai-subset	14	0.64
npm:@types/cheerio	14	0.64
npm:@types/clean-css	14	0.64
npm:@types/content-disposition	14	0.64
npm:@types/emscripten	14	0.64
npm:@types/express-session	14	0.64
npm:@types/lodash.mergewith	14	0.64
npm:@types/mapbox-gl	14	0.64
npm:@types/styled-components	14	0.64
npm:@types/tmp	14	0.64
npm:@types/validator	14	0.64
npm:@vue/cli-plugin-typescript	14	0.64
npm:@wojtekmaj/date-utils	14	0.64
npm:@wry/trie	14	0.64
npm:add-dom-event-listener	14	0.64
npm:airbnb-js-shims	14	0.64
npm:algoliasearch-helper	14	0.64
npm:ansi	14	0.64
npm:any-base	14	0.64
npm:auto-bind	14	0.64
npm:babel-helper-evaluate-path	14	0.64
npm:babel-helper-mark-eval-scopes	14	0.64
npm:babel-helper-remove-or-void	14	0.64
npm:babel-plugin-lodash	14	0.64

npm:babel-plugin-transform-remove-console	14	0.64
npm:bare-fs	14	0.64
npm:bare-os	14	0.64
npm:bare-path	14	0.64
npm:batch-processor	14	0.64
npm:benchmark	14	0.64
npm:binary-search-bounds	14	0.64
npm:bitmap-sdf	14	0.64
npm:brace	14	0.64
npm:cache-content-type	14	0.64
npm:chroma-js	14	0.64
npm:cli-progress	14	0.64
npm:color-parse	14	0.64
npm:color-rgba	14	0.64
npm:commitizen	14	0.64
npm:complex.js	14	0.64
npm:component-classes	14	0.64
npm:component-indexof	14	0.64
npm:component-type	14	0.64
npm:connect-livereload	14	0.64
npm:copyfiles	14	0.64
npm:cron	14	0.64
npm:css-animation	14	0.64
npm:d3-hexbin	14	0.64

npm:deep-object-diff	14	0.64
npm:detect-browser	14	0.64
npm:difflib	14	0.64
npm:dom-storage	14	0.64
npm:echarts	14	0.64
npm:editor	14	0.64
npm:element-resize-detector	14	0.64
npm:ember-cli-string-utils	14	0.64
npm:emotion-theming	14	0.64
npm:endent	14	0.64
npm:eth-ens-namehash	14	0.64
npm:ethereumjs-tx	14	0.64
npm:exec-buffer	14	0.64
npm:expo-asset	14	0.64
npm:expo-constants	14	0.64
npm:expo-file-system	14	0.64
npm:expo-font	14	0.64
npm:fast-decode-uri-component	14	0.64
npm:find-parent-dir	14	0.64
npm:get-package-info	14	0.64
npm:graphql-ws	14	0.64
npm:grpc	14	0.64
npm:grunt-concurrent	14	0.64
npm:grunt-contrib-compress	14	0.64
npm:grunt-contrib-imagemin	14	0.64

npm:grunt-mocha-test	14	0.64
npm:gulp-clean	14	0.64
npm:gulp-load-plugins	14	0.64
npm:gulp-minify-css	14	0.64
npm:gulp-watch	14	0.64
npm:hardhat	14	0.64
npm:hsluv	14	0.64
npm:html-parse-stringify	14	0.64
npm:httpntlm	14	0.64
npm:httppreq	14	0.64
npm:idna-uts46-hx	14	0.64
npm:imagemin-optipng	14	0.64
npm:infima	14	0.64
npm:intersection-observer	14	0.64
npm:is-mobile	14	0.64
npm:is-png	14	0.64
npm:jimp	14	0.64
npm:js-file-download	14	0.64
npm:js-sha256	14	0.64
npm:jsc-safe-url	14	0.64
npm:json-schema-ref-parser	14	0.64
npm:karma-mocha-reporter	14	0.64
npm:keyboard-key	14	0.64
npm:lightningcss	14	0.64
npm:lightningcss-darwin-arm64	14	0.64

npm:lightningcss-darwin-x64	14	0.64
npm:lightningcss-linux-arm-gnueabihf	14	0.64
npm:lightningcss-linux-arm64-gnu	14	0.64
npm:lightningcss-linux-arm64-musl	14	0.64
npm:lightningcss-linux-x64-gnu	14	0.64
npm:lightningcss-linux-x64-musl	14	0.64
npm:lightningcss-win32-x64-msvc	14	0.64
npm:lodash.pad	14	0.64
npm:lodash.pickby	14	0.64
npm:mark.js	14	0.64
npm:match-sorter	14	0.64
npm:math-log2	14	0.64
npm:meant	14	0.64
npm:metro-minify-terser	14	0.64
npm:mkpath	14	0.64
npm:msgpackr	14	0.64
npm:msgpackr-extract	14	0.64
npm:mumath	14	0.64
npm:nanoclone	14	0.64
npm:nerf-dart	14	0.64
npm:node-fetch-native	14	0.64
npm:node-gyp-build-optional-packages	14	0.64
npm:normalize-svg-path	14	0.64

npm:normalize-wheel	14	0.64
npm:on-exit-leak-free	14	0.64
npm:outdent	14	0.64
npm:outpipe	14	0.64
npm:parse-png	14	0.64
npm:passport-google-oauth20	14	0.64
npm:pidusage	14	0.64
npm:pino-abstract-transport	14	0.64
npm:point-in-polygon	14	0.64
npm:postcss-html	14	0.64
npm:postcss-sass	14	0.64
npm:rc-animate	14	0.64
npm:rcedit	14	0.64
npm:react-deep-force-update	14	0.64
npm:react-ga	14	0.64
npm:react-loadable	14	0.64
npm:react-map-gl	14	0.64
npm:react-onclickoutside	14	0.64
npm:react-proxy	14	0.64
npm:read-chunk	14	0.64
npm:real-require	14	0.64
npm:rehype-raw	14	0.64
npm:remark-external-links	14	0.64
npm:right-now	14	0.64
npm:rrweb-cssom	14	0.64

npm:serialport	14	0.64
npm:sha	14	0.64
npm:sorted-array-functions	14	0.64
npm:stream-to-array	14	0.64
npm:style-mod	14	0.64
npm:stylelint-order	14	0.64
npm:subscriptions-transport-ws	14	0.64
npm:svg-pathdata	14	0.64
npm:swagger-ui-dist	14	0.64
npm:symbol.prototype.description	14	0.64
npm:sync-rpc	14	0.64
npm:thread-stream	14	0.64
npm:ts-morph	14	0.64
npm:tsickle	14	0.64
npm:twilio	14	0.64
npm:type-name	14	0.64
npm:typedarray-pool	14	0.64
npm:unist-util-find-after	14	0.64
npm:unplugin	14	0.64
npm:utile	14	0.64
npm:vitefu	14	0.64
npm:vue-jest	14	0.64
npm:vue-server-renderer	14	0.64
npm:walkdir	14	0.64
npm:xhr2	14	0.64

npm:ylru	14	0.64
npm:@blueprintjs/core	13	0.59
npm:@date-io/core	13	0.59
npm:@date-io/moment	13	0.59
npm:@docusaurus/core	13	0.59
npm:@docusaurus/mdx-loader	13	0.59
npm:@docusaurus/plugin-content- blog	13	0.59
npm:@docusaurus/plugin-content- docs	13	0.59
npm:@docusaurus/plugin-content- pages	13	0.59
npm:@docusaurus/plugin-debug	13	0.59
npm:@docusaurus/plugin-google- analytics	13	0.59
npm:@docusaurus/plugin-google- gtag	13	0.59
npm:@docusaurus/plugin-sitemap	13	0.59
npm:@docusaurus/preset-classic	13	0.59
npm:@docusaurus/theme-classic	13	0.59
npm:@docusaurus/theme-search- algolia	13	0.59
npm:@docusaurus/utils	13	0.59
npm:@docusaurus/utils-validation	13	0.59
npm:@google-cloud/firestore	13	0.59

npm:@hypnosphi/create-react-context	13	0.59
npm:@josephg/resolvable	13	0.59
npm:@jupyterlab/docregistry	13	0.59
npm:@jupyterlab/nbformat	13	0.59
npm:@lumino/properties	13	0.59
npm:@material/dom	13	0.59
npm:@material/feature-targeting	13	0.59
npm:@next/swc-freebsd-x64	13	0.59
npm:@radix-ui/react-popover	13	0.59
npm:@radix-ui/react-tabs	13	0.59
npm:@react-native/virtualized-lists	13	0.59
npm:@semantic-ui-react/event-stack	13	0.59
npm:@sindresorhus/slugify	13	0.59
npm:@sindresorhus/transliterate	13	0.59
npm:@storybook/addon-backgrounds	13	0.59
npm:@storybook/addon-docs	13	0.59
npm:@storybook/addon-toolbars	13	0.59
npm:@storybook/addon-viewport	13	0.59
npm:@storybook/csf-tools	13	0.59
npm:@storybook/postinstall	13	0.59
npm:@storybook/ui	13	0.59
npm:@testing-library/react-hooks	13	0.59

npm:@types/chai-as-promised	13	0.59
npm:@types/cookies	13	0.59
npm:@types/d3-collection	13	0.59
npm:@types/enzyme	13	0.59
npm:@types/expect	13	0.59
npm:@types/highlight.js	13	0.59
npm:@types/is-function	13	0.59
npm:@types/keygrip	13	0.59
npm:@types/koa	13	0.59
npm:@types/raf	13	0.59
npm:@types/react-router-config	13	0.59
npm:@types/supertest	13	0.59
npm:@types/vinyl	13	0.59
npm:@types/webxr	13	0.59
npm:apollo-link-http	13	0.59
npm:application-config-path	13	0.59
npm:ascli	13	0.59
npm:babel-helper-flip-expressions	13	0.59
npm:babel-helper-is-nodes-equiv	13	0.59
npm:babel-helper-is-void-0	13	0.59
npm:babel-helper-to-multiple- sequence-expressions	13	0.59
npm:babel-plugin-add-react- displayname	13	0.59
npm:babel-plugin-minify-builtins	13	0.59

npm:babel-plugin-minify-constant-folding	13	0.59
npm:babel-plugin-minify-dead-code-elimination	13	0.59
npm:babel-plugin-minify-flip-comparisons	13	0.59
npm:babel-plugin-minify-guarded-expressions	13	0.59
npm:babel-plugin-minify-infinity	13	0.59
npm:babel-plugin-minify-mangle-names	13	0.59
npm:babel-plugin-minify-numeric-literals	13	0.59
npm:babel-plugin-minify-replace	13	0.59
npm:babel-plugin-minify-simplify	13	0.59
npm:babel-plugin-minify-type-constructors	13	0.59
npm:babel-plugin-transform-inline-consecutive-adds	13	0.59
npm:babel-plugin-transform-member-expression-literals	13	0.59
npm:babel-plugin-transform-merge-sibling-variables	13	0.59
npm:babel-plugin-transform-minify-booleans	13	0.59

npm:babel-plugin-transform-property-literals	13	0.59
npm:babel-plugin-transform-regexp-constructors	13	0.59
npm:babel-plugin-transform-remove-debugger	13	0.59
npm:babel-plugin-transform-remove-undefined	13	0.59
npm:babel-plugin-transform-simplify-comparison-operators	13	0.59
npm:babel-plugin-transform-undefined-to-void	13	0.59
npm:basic-auth-connect	13	0.59
npm:colour	13	0.59
npm:concordance	13	0.59
npm:conf	13	0.59
npm:conventional-changelog	13	0.59
npm:core-object	13	0.59
npm:cssfontparser	13	0.59
npm:cssnano-preset-advanced	13	0.59
npm:direction	13	0.59
npm:dotenv-webpack	13	0.59
npm:dset	13	0.59
npm:each-async	13	0.59
npm:electron-log	13	0.59

npm:element-size	13	0.59
npm:encode-utf8	13	0.59
npm:express-rate-limit	13	0.59
npm:fast-csv	13	0.59
npm:gifsicle	13	0.59
npm:global-modules-path	13	0.59
npm:googleapis	13	0.59
npm:grunt-eslint	13	0.59
npm:gulp-size	13	0.59
npm:hast-util-to-text	13	0.59
npm:hook-std	13	0.59
npm:is-blob	13	0.59
npm:is-electron	13	0.59
npm:is-file-esm	13	0.59
npm:is-jpg	13	0.59
npm:jss-camel-case	13	0.59
npm:jss-default-unit	13	0.59
npm:jss-global	13	0.59
npm:jss-nested	13	0.59
npm:jss-props-sort	13	0.59
npm:jss-vendor-prefixer	13	0.59
npm:JSV	13	0.59
npm:karma-requirejs	13	0.59
npm:khroma	13	0.59
npm:laravel-mix	13	0.59

npm:lazy	13	0.59
npm:lodash._baseuniq	13	0.59
npm:lodash._createset	13	0.59
npm:lodash.flatmap	13	0.59
npm:lodash.without	13	0.59
npm:loglevelnext	13	0.59
npm:mailcomposer	13	0.59
npm:mdast-util-frontmatter	13	0.59
npm:memdown	13	0.59
npm:mermaid	13	0.59
npm:micromark-extension- frontmatter	13	0.59
npm:mqtt	13	0.59
npm:node-modules-path	13	0.59
npm:node-static	13	0.59
npm:npm-cache-filename	13	0.59
npm:nunjucks	13	0.59
npm:optjs	13	0.59
npm:parcel-bundler	13	0.59
npm:parse-color	13	0.59
npm:passport-jwt	13	0.59
npm:path-complete-extname	13	0.59
npm:physical-cpu-count	13	0.59
npm:pick-by-alias	13	0.59
npm:polybooljs	13	0.59

npm:pop-iterate	13	0.59
npm:postcss-sort-media-queries	13	0.59
npm:postcss-sorting	13	0.59
npm:postcss-syntax	13	0.59
npm:probe-image-size	13	0.59
npm:progress-webpack-plugin	13	0.59
npm:proj4	13	0.59
npm:promise.series	13	0.59
npm:react-ace	13	0.59
npm:react-calendar	13	0.59
npm:react-docgen-typescript	13	0.59
npm:react-immutable-proptypes	13	0.59
npm:react-native-safe-area-context	13	0.59
npm:react-native-vector-icons	13	0.59
npm:react-router-bootstrap	13	0.59
npm:read-installed	13	0.59
npm:relay-runtime	13	0.59
npm:remarkable	13	0.59
npm:replacestream	13	0.59
npm:rootpath	13	0.59
npm:select2	13	0.59
npm:settle-promise	13	0.59
npm:shader-loader	13	0.59
npm:signedsource	13	0.59
npm:sort-css-media-queries	13	0.59

npm:sorted-object	13	0.59
npm:style-inject	13	0.59
npm:stylelint-scss	13	0.59
npm:systemjs	13	0.59
npm:tarn	13	0.59
npm:update-check	13	0.59
npm:vscode-languageserver	13	0.59
npm:vue-class-component	13	0.59
npm:webpack-notifier	13	0.59
npm:well-known-symbols	13	0.59
npm:x-xss-protection	13	0.59
npm:@angular-eslint/eslint-plugin	12	0.54
npm:@apollographql/apollo-tools	12	0.54
npm:@aws-sdk/client-s3	12	0.54
npm:@aws-sdk/credential- providers	12	0.54
npm:@aws-sdk/middleware- bucket-endpoint	12	0.54
npm:@aws-sdk/middleware-expect- continue	12	0.54
npm:@aws-sdk/middleware- location-constraint	12	0.54
npm:@aws-sdk/middleware-sdk-s3	12	0.54
npm:@aws-sdk/middleware-ssec	12	0.54
npm:@aws-sdk/util-arn-parser	12	0.54

npm:@aws-sdk/xml-builder	12	0.54
npm:@codemirror/language	12	0.54
npm:@docusaurus/cssnano-preset	12	0.54
npm:@docusaurus/react-loadable	12	0.54
npm:@docusaurus/theme-common	12	0.54
npm:@expo/config-types	12	0.54
npm:@expo/json-file	12	0.54
npm:@expo/plist	12	0.54
npm:@expo/spawn-async	12	0.54
npm:@firebase/messaging-types	12	0.54
npm:@formatjs/ecma402-abstract	12	0.54
npm:@formatjs/intl-localematcher	12	0.54
npm:@graphql-tools/batch-execute	12	0.54
npm:@graphql-tools/code-file-loader	12	0.54
npm:@graphql-tools/delegate	12	0.54
npm:@graphql-tools/graphql-tag-pluck	12	0.54
npm:@graphql-tools/relay-operation-optimizer	12	0.54
npm:@graphql-tools/wrap	12	0.54
npm:@heroicons/react	12	0.54
npm:@jimp/bmp	12	0.54
npm:@jimp/core	12	0.54
npm:@jimp/custom	12	0.54

npm:@jimp/gif	12	0.54
npm:@jimp/jpeg	12	0.54
npm:@jimp/plugin-blit	12	0.54
npm:@jimp/plugin-blur	12	0.54
npm:@jimp/plugin-color	12	0.54
npm:@jimp/plugin-contain	12	0.54
npm:@jimp/plugin-cover	12	0.54
npm:@jimp/plugin-crop	12	0.54
npm:@jimp/plugin-displace	12	0.54
npm:@jimp/plugin-dither	12	0.54
npm:@jimp/plugin-flip	12	0.54
npm:@jimp/plugin-gaussian	12	0.54
npm:@jimp/plugin-invert	12	0.54
npm:@jimp/plugin-mask	12	0.54
npm:@jimp/plugin-normalize	12	0.54
npm:@jimp/plugin-print	12	0.54
npm:@jimp/plugin-resize	12	0.54
npm:@jimp/plugin-rotate	12	0.54
npm:@jimp/plugin-scale	12	0.54
npm:@jimp/plugins	12	0.54
npm:@jimp/png	12	0.54
npm:@jimp/tiff	12	0.54
npm:@jimp/types	12	0.54
npm:@jimp/utls	12	0.54

npm:@jupyterlab/rendermime- interfaces	12	0.54
npm:@jupyterlab/statedb	12	0.54
npm:@kwsites/file-exists	12	0.54
npm:@kwsites/promise-deferred	12	0.54
npm:@lezer/highlight	12	0.54
npm:@lumino/collections	12	0.54
npm:@lumino/domutils	12	0.54
npm:@lumino/dragdrop	12	0.54
npm:@lumino/keyboard	12	0.54
npm:@lumino/polling	12	0.54
npm:@lumino/virtualdom	12	0.54
npm:@material/elevation	12	0.54
npm:@material/rtl	12	0.54
npm:@material/typography	12	0.54
npm:@mischnic/json-sourcemap	12	0.54
npm:@msgpackr- extract/msgpackr-extract-darwin- arm64	12	0.54
npm:@msgpackr- extract/msgpackr-extract-darwin- x64	12	0.54
npm:@msgpackr- extract/msgpackr-extract-linux- arm	12	0.54

npm:@msgpackr- extract/msgpackr-extract-linux- arm64	12	0.54
npm:@msgpackr- extract/msgpackr-extract-linux- x64	12	0.54
npm:@msgpackr- extract/msgpackr-extract-win32- x64	12	0.54
npm:@opentelemetry/semantic- conventions	12	0.54
npm:@parcel/cache	12	0.54
npm:@parcel/codeframe	12	0.54
npm:@parcel/diagnostic	12	0.54
npm:@parcel/events	12	0.54
npm:@parcel/markdown-ansi	12	0.54
npm:@parcel/package-manager	12	0.54
npm:@parcel/plugin	12	0.54
npm:@parcel/source-map	12	0.54
npm:@parcel/types	12	0.54
npm:@parcel/watcher-android- arm64	12	0.54
npm:@parcel/watcher-darwin- arm64	12	0.54
npm:@parcel/watcher-darwin-x64	12	0.54

npm:@parcel/watcher-linux-arm-glibc	12	0.54
npm:@parcel/watcher-linux-arm64-glibc	12	0.54
npm:@parcel/watcher-linux-arm64-musl	12	0.54
npm:@parcel/watcher-linux-x64-glibc	12	0.54
npm:@parcel/watcher-linux-x64-musl	12	0.54
npm:@parcel/watcher-win32-arm64	12	0.54
npm:@parcel/watcher-win32-x64	12	0.54
npm:@playwright/test	12	0.54
npm:@plotly/d3-sankey	12	0.54
npm:@radix-ui/react-collapsible	12	0.54
npm:@radix-ui/react-select	12	0.54
npm:@react-native/assets	12	0.54
npm:@react-native/assets-registry	12	0.54
npm:@react-native/codegen	12	0.54
npm:@react-native/gradle-plugin	12	0.54
npm:@react-native/js-polyfills	12	0.54
npm:@react-native/normalize-colors	12	0.54
npm:@react-native/polyfills	12	0.54

npm:@react-navigation/routers	12	0.54
npm:@rollup/plugin-alias	12	0.54
npm:@rollup/plugin-inject	12	0.54
npm:@rollup/rollup-linux-s390x-gnu	12	0.54
npm:@semantic-release/git	12	0.54
npm:@slorber/static-site-generator-webpack-plugin	12	0.54
npm:@solidity-parser/parser	12	0.54
npm:@storybook/addon-measure	12	0.54
npm:@storybook/core	12	0.54
npm:@swc/counter	12	0.54
npm:@types/backbone	12	0.54
npm:@types/color-convert	12	0.54
npm:@types/duplexify	12	0.54
npm:@types/http-assert	12	0.54
npm:@types/jss	12	0.54
npm:@types/koa-compose	12	0.54
npm:@types/leaflet	12	0.54
npm:@types/mime-types	12	0.54
npm:@types/nodemailer	12	0.54
npm:@types/xml2js	12	0.54
npm:@types/zen-observable	12	0.54
npm:@zxing/text-encoding	12	0.54
npm:antlr4ts	12	0.54

npm:apollo-cache-inmemory	12	0.54
npm:apollo-client	12	0.54
npm:apollo-datasource	12	0.54
npm:apollo-link-http-common	12	0.54
npm:apollo-server-env	12	0.54
npm:apollo-server-errors	12	0.54
npm:array-bounds	12	0.54
npm:array-normalize	12	0.54
npm:array-range	12	0.54
npm:array-rearrange	12	0.54
npm:ast-transform	12	0.54
npm:async-eventemitter	12	0.54
npm:autocomplete.js	12	0.54
npm:awesome-typescript-loader	12	0.54
npm:babel-plugin-react-docgen	12	0.54
npm:babel-plugin-react-native-web	12	0.54
npm:babel-plugin-transform-flow- enums	12	0.54
npm:browserify-optional	12	0.54
npm:buildmail	12	0.54
npm:bunyan	12	0.54
npm:canvas-fit	12	0.54
npm:case	12	0.54
npm:clean-regexp	12	0.54
npm:color-id	12	0.54

npm:color-normalize	12	0.54
npm:combine-promises	12	0.54
npm:content-security-policy-builder	12	0.54
npm:continuation-local-storage	12	0.54
npm:conventional-changelog-eslint	12	0.54
npm:convert-hrtime	12	0.54
npm:country-regex	12	0.54
npm:css-font-size-keywords	12	0.54
npm:css-font-stretch-keywords	12	0.54
npm:css-font-style-keywords	12	0.54
npm:css-font-weight-keywords	12	0.54
npm:css-global-keywords	12	0.54
npm:css-system-font-keywords	12	0.54
npm:csurf	12	0.54
npm:ctype	12	0.54
npm:dont-sniff-mimetype	12	0.54
npm:dotenv-defaults	12	0.54
npm:draw-svg-path	12	0.54
npm:emitter-component	12	0.54
npm:eslint-plugin-compat	12	0.54
npm:esniff	12	0.54
npm:espurify	12	0.54
npm:estree-to-babel	12	0.54
npm:estree-util-value-to-estree	12	0.54

npm:ethereumjs-common	12	0.54
npm:expo-keep-awake	12	0.54
npm:expose-loader	12	0.54
npm:express-handlebars	12	0.54
npm:fast-isnumeric	12	0.54
npm:flow-enums-runtime	12	0.54
npm:fn-name	12	0.54
npm:format-util	12	0.54
npm:fs-vacuum	12	0.54
npm:get-canvas-context	12	0.54
npm:getenv	12	0.54
npm:graphql-request	12	0.54
npm:grunt-exec	12	0.54
npm:grunt-newer	12	0.54
npm:grunt-usemin	12	0.54
npm:gulp-match	12	0.54
npm:has-glob	12	0.54
npm:has-hover	12	0.54
npm:has-passive-events	12	0.54
npm:helmet-csp	12	0.54
npm:hide-powered-by	12	0.54
npm:highcharts	12	0.54
npm:hpkp	12	0.54
npm:hsts	12	0.54
npm:imagemin-gifsicle	12	0.54

npm:imagemin-svgo	12	0.54
npm:iota-array	12	0.54
npm:is-browser	12	0.54
npm:is-empty	12	0.54
npm:is-error	12	0.54
npm:is-gif	12	0.54
npm:is-iexplorer	12	0.54
npm:is-json	12	0.54
npm:is-svg-path	12	0.54
npm:iso-url	12	0.54
npm:jetifier	12	0.54
npm:jstom-global	12	0.54
npm:json-parse-helpfulerror	12	0.54
npm:json-schema-deref-sync	12	0.54
npm:libphonenumber-js	12	0.54
npm:lmdb	12	0.54
npm:locate-character	12	0.54
npm:lodash.isnil	12	0.54
npm:lodash.keysin	12	0.54
npm:lru-memoizer	12	0.54
npm:meros	12	0.54
npm:metro-hermes-compiler	12	0.54
npm:mouse-change	12	0.54
npm:mouse-event	12	0.54
npm:mouse-event-offset	12	0.54

npm:mouse-wheel	12	0.54
npm:multiformats	12	0.54
npm:ndarray	12	0.54
npm:nodemailer-fetch	12	0.54
npm:nodemailer-shared	12	0.54
npm:ordered-binary	12	0.54
npm:p-all	12	0.54
npm:parse-rect	12	0.54
npm:parse-unit	12	0.54
npm:passport-oauth1	12	0.54
npm:playwright	12	0.54
npm:promptly	12	0.54
npm:qunit	12	0.54
npm:react-highlight-words	12	0.54
npm:react-loadable-ssr-addon-v5- slorber	12	0.54
npm:react-merge-refs	12	0.54
npm:react-native-codegen	12	0.54
npm:react-sizeme	12	0.54
npm:react-spinners	12	0.54
npm:react-tabs	12	0.54
npm:react-use-measure	12	0.54
npm:read-yaml-file	12	0.54
npm:regl-error2d	12	0.54
npm:regl-line2d	12	0.54

npm:regl-scatter2d	12	0.54
npm:rehype-stringify	12	0.54
npm:rollup-plugin-uglify	12	0.54
npm:run-parallel-limit	12	0.54
npm:semaphore	12	0.54
npm:signum	12	0.54
npm:strongly-connected- components	12	0.54
npm:superscript-text	12	0.54
npm:svg-path-bounds	12	0.54
npm:svg-path-sdf	12	0.54
npm:swr	12	0.54
npm:text-encoding-utf-8	12	0.54
npm:timm	12	0.54
npm:to-camel-case	12	0.54
npm:to-float32	12	0.54
npm:to-px	12	0.54
npm:to-vfile	12	0.54
npm:treeify	12	0.54
npm:tsconfig-paths-webpack-plugin	12	0.54
npm:tsx	12	0.54
npm:tweetnacl-util	12	0.54
npm:type-of	12	0.54
npm:typed-function	12	0.54
npm:typedoc-default-themes	12	0.54

npm:uint8arrays	12	0.54
npm:umzug	12	0.54
npm:unist-util-find-all-after	12	0.54
npm:update-diff	12	0.54
npm:uri-path	12	0.54
npm:vega	12	0.54
npm:vega-lite	12	0.54
npm:vega-statistics	12	0.54
npm:vinyl-buffer	12	0.54
npm:weak-lru-cache	12	0.54
npm:web-encoding	12	0.54
npm:webgl-context	12	0.54
npm:webworker-promise	12	0.54
npm:world-calendars	12	0.54
npm:xhr2-cookies	12	0.54
npm:@actions/core	11	0.50
npm:@algolia/events	11	0.50
npm:@angular-eslint/eslint-plugin- template	11	0.50
npm:@apollographql/graphql- playground-html	11	0.50
npm:@aws-crypto/crc32c	11	0.50
npm:@aws-crypto/sha1-browser	11	0.50
npm:@aws-sdk/middleware- flexible-checksums	11	0.50

npm:@aws-sdk/signature-v4-multi-region	11	0.50
npm:@aws-sdk/util-base64	11	0.50
npm:@aws-sdk/util-utf8-node	11	0.50
npm:@chakra-ui/object-utils	11	0.50
npm:@chakra-ui/react-use-safe-layout-effect	11	0.50
npm:@chakra-ui/shared-utils	11	0.50
npm:@chakra-ui/theme-utils	11	0.50
npm:@docusaurus/utils-common	11	0.50
npm:@expo/config	11	0.50
npm:@expo/config-plugins	11	0.50
npm:@expo/image-utils	11	0.50
npm:@expo/metro-config	11	0.50
npm:@expo/sdk-runtime-versions	11	0.50
npm:@formatjs/fast-memoize	11	0.50
npm:@formatjs/icu-messageformat-parser	11	0.50
npm:@formatjs/icu-skeleton-parser	11	0.50
npm:@graphql-codegen/core	11	0.50
npm:@graphql-codegen/plugin-helpers	11	0.50
npm:@graphql-codegen/schema-ast	11	0.50
npm:@graphql-codegen/typescript	11	0.50

npm:@graphql-codegen/typescript-operations	11	0.50
npm:@graphql-codegen/visitor-plugin-common	11	0.50
npm:@graphql-tools/optimize	11	0.50
npm:@hapi/boom	11	0.50
npm:@icons/material	11	0.50
npm:@ioredis/commands	11	0.50
npm:@kurkle/color	11	0.50
npm:@lerna/add	11	0.50
npm:@lerna/bootstrap	11	0.50
npm:@lerna/changed	11	0.50
npm:@lerna/check-working-tree	11	0.50
npm:@lerna/clean	11	0.50
npm:@lerna/cli	11	0.50
npm:@lerna/collect-uncommitted	11	0.50
npm:@lerna/collect-updates	11	0.50
npm:@lerna/command	11	0.50
npm:@lerna/conventional-commits	11	0.50
npm:@lerna/create-symlink	11	0.50
npm:@lerna/describe-ref	11	0.50
npm:@lerna/diff	11	0.50
npm:@lerna/exec	11	0.50
npm:@lerna/filter-options	11	0.50
npm:@lerna/filter-packages	11	0.50

npm:@lerna/get-npm-exec-opts	11	0.50
npm:@lerna/get-packed	11	0.50
npm:@lerna/github-client	11	0.50
npm:@lerna/gitlab-client	11	0.50
npm:@lerna/global-options	11	0.50
npm:@lerna/has-npm-version	11	0.50
npm:@lerna/import	11	0.50
npm:@lerna/info	11	0.50
npm:@lerna/init	11	0.50
npm:@lerna/link	11	0.50
npm:@lerna/list	11	0.50
npm:@lerna/listable	11	0.50
npm:@lerna/log-packed	11	0.50
npm:@lerna/npm-conf	11	0.50
npm:@lerna/npm-dist-tag	11	0.50
npm:@lerna/npm-install	11	0.50
npm:@lerna/npm-publish	11	0.50
npm:@lerna/npm-run-script	11	0.50
npm:@lerna/otplease	11	0.50
npm:@lerna/output	11	0.50
npm:@lerna/pack-directory	11	0.50
npm:@lerna/package	11	0.50
npm:@lerna/package-graph	11	0.50
npm:@lerna/prerelease-id-from- version	11	0.50

npm:@lerna/profiler	11	0.50
npm:@lerna/project	11	0.50
npm:@lerna/prompt	11	0.50
npm:@lerna/publish	11	0.50
npm:@lerna/pulse-till-done	11	0.50
npm:@lerna/query-graph	11	0.50
npm:@lerna/resolve-symlink	11	0.50
npm:@lerna/rimraf-dir	11	0.50
npm:@lerna/run	11	0.50
npm:@lerna/run-lifecycle	11	0.50
npm:@lerna/run-topologically	11	0.50
npm:@lerna/symlink-binary	11	0.50
npm:@lerna/symlink-dependencies	11	0.50
npm:@lerna/timer	11	0.50
npm:@lerna/validation-error	11	0.50
npm:@lerna/version	11	0.50
npm:@lerna/write-log-file	11	0.50
npm:@material/button	11	0.50
npm:@material/density	11	0.50
npm:@material/touch-target	11	0.50
npm:@metamask/eth-sig-util	11	0.50
npm:@mui/lab	11	0.50
npm:@mui/x-date-pickers	11	0.50
npm:@noble/curves	11	0.50
npm:@open-draft/until	11	0.50

npm:@parcel/bundler-default	11	0.50
npm:@parcel/compressor-raw	11	0.50
npm:@parcel/core	11	0.50
npm:@parcel/graph	11	0.50
npm:@parcel/namer-default	11	0.50
npm:@parcel/node-resolver-core	11	0.50
npm:@parcel/packager-js	11	0.50
npm:@parcel/packager-raw	11	0.50
npm:@parcel/reporter-dev-server	11	0.50
npm:@parcel/resolver-default	11	0.50
npm:@parcel/runtime-js	11	0.50
npm:@parcel/transformer-js	11	0.50
npm:@parcel/transformer-json	11	0.50
npm:@plotly/d3-sankey-circular	11	0.50
npm:@radix-ui/react-dialog	11	0.50
npm:@radix-ui/react-label	11	0.50
npm:@rails/actioncable	11	0.50
npm:@reach/router	11	0.50
npm:@react-dnd/asap	11	0.50
npm:@react-dnd/invariant	11	0.50
npm:@react-dnd/shallowequal	11	0.50
npm:@rollup/rollup-linux- powerpc64le-gnu	11	0.50
npm:@segment/loosely-validate- event	11	0.50

npm:@semantic-release/changelog	11	0.50
npm:@sigstore/core	11	0.50
npm:@sigstore/verify	11	0.50
npm:@socket.io/base64-arraybuffer	11	0.50
npm:@storybook/addon-outline	11	0.50
npm:@storybook/semver	11	0.50
npm:@tensorflow/tfjs	11	0.50
npm:@types/argparse	11	0.50
npm:@types/aws-lambda	11	0.50
npm:@types/chart.js	11	0.50
npm:@types/compression	11	0.50
npm:@types/cookie-parser	11	0.50
npm:@types/cross-spawn	11	0.50
npm:@volar/source-map	11	0.50
npm:angular-mocks	11	0.50
npm:apollo-cache	11	0.50
npm:apollo-server-core	11	0.50
npm:apollo-server-express	11	0.50
npm:apollo-server-plugin-base	11	0.50
npm:apollo-server-types	11	0.50
npm:async-lock	11	0.50
npm:async-sema	11	0.50
npm:better-path-resolve	11	0.50
npm:bfj-node4	11	0.50
npm:bip39	11	0.50

npm:bitsyntax	11	0.50
npm:buble	11	0.50
npm:change-case-all	11	0.50
npm:cjson	11	0.50
npm:coalescy	11	0.50
npm:code-excerpt	11	0.50
npm:color-alpha	11	0.50
npm:commoner	11	0.50
npm:content-hash	11	0.50
npm:conventional-changelog-atom	11	0.50
npm:conventional-changelog-codemirror	11	0.50
npm:conventional-changelog-ember	11	0.50
npm:conventional-changelog-express	11	0.50
npm:conventional-changelog-jquery	11	0.50
npm:conventional-changelog-jshint	11	0.50
npm:convert-to-spaces	11	0.50
npm:copy-to	11	0.50
npm:crelt	11	0.50
npm:csrf	11	0.50
npm:css-font	11	0.50
npm:dasherize	11	0.50
npm:death	11	0.50
npm:decompress-zip	11	0.50

npm:destr	11	0.50
npm:detect-kerning	11	0.50
npm:dijkstra.js	11	0.50
npm:electron-devtools-installer	11	0.50
npm:electron-is-dev	11	0.50
npm:elementary-circuits-directed-graph	11	0.50
npm:elk.js	11	0.50
npm:encoding-down	11	0.50
npm:ensure-posix-path	11	0.50
npm:esbuild-register	11	0.50
npm:eslint-config-airbnb-typescript	11	0.50
npm:eslint-plugin-babel	11	0.50
npm:eslint-plugin-mocha	11	0.50
npm:eslint-plugin-svelte3	11	0.50
npm:eslint-rule-composer	11	0.50
npm:esprima-fb	11	0.50
npm:estree-is-function	11	0.50
npm:expo-modules-autolinking	11	0.50
npm:expo-modules-core	11	0.50
npm:expo-status-bar	11	0.50
npm:express-fileupload	11	0.50
npm:express-ws	11	0.50
npm:extendable-error	11	0.50
npm:extract-from-css	11	0.50

npm:fast-querystring	11	0.50
npm:fetch-ponyfill	11	0.50
npm:file-sync-cmp	11	0.50
npm:file-url	11	0.50
npm:firebase-functions	11	0.50
npm:font-atlas	11	0.50
npm:font-measure	11	0.50
npm:fork-stream	11	0.50
npm:formdata-node	11	0.50
npm:getopts	11	0.50
npm:gl-text	11	0.50
npm:gl-util	11	0.50
npm:goober	11	0.50
npm:graphql-subscriptions	11	0.50
npm:graphql-type-json	11	0.50
npm:grunt-browserify	11	0.50
npm:gulp-mocha	11	0.50
npm:gunzip-maybe	11	0.50
npm:hast-util-has-property	11	0.50
npm:hast-util-to-jsx-runtime	11	0.50
npm:hast-util-to-string	11	0.50
npm:hermes-engine	11	0.50
npm:http-status-codes	11	0.50
npm:https	11	0.50
npm:idb-keyval	11	0.50

npm:imagemin-jpegtran	11	0.50
npm:immutability-helper	11	0.50
npm:indx	11	0.50
npm:insert-css	11	0.50
npm:is-base64	11	0.50
npm:is-deflate	11	0.50
npm:is-firefox	11	0.50
npm:is-string-blank	11	0.50
npm:is-subdir	11	0.50
npm:jit-grunt	11	0.50
npm:join-component	11	0.50
npm:jsdoctypeparser	11	0.50
npm:jsonlint	11	0.50
npm:jwks-rsa	11	0.50
npm:koa-is-json	11	0.50
npm:koa-send	11	0.50
npm:koa-static	11	0.50
npm:konva	11	0.50
npm:level-js	11	0.50
npm:level-packager	11	0.50
npm:level-ws	11	0.50
npm:libnpmconfig	11	0.50
npm:lit-html	11	0.50
npm:load-tsconfig	11	0.50
npm:lodash._baseindexof	11	0.50

npm:lodash._baseisequal	11	0.50
npm:lodash._cacheindexof	11	0.50
npm:lodash._createcache	11	0.50
npm:lodash._escapehtmlchar	11	0.50
npm:lodash._escapestringchar	11	0.50
npm:lodash._htmlescapes	11	0.50
npm:lodash._reunesapedhtml	11	0.50
npm:lodash.chunk	11	0.50
npm:lodash.partialright	11	0.50
npm:lusca	11	0.50
npm:matcher-collection	11	0.50
npm:merkle-patricia-tree	11	0.50
npm:metro-babel-register	11	0.50
npm:mgrs	11	0.50
npm:mobx-react-lite	11	0.50
npm:mpromise	11	0.50
npm:msw	11	0.50
npm:multiparty	11	0.50
npm:node-object-hash	11	0.50
npm:nodemailer-direct-transport	11	0.50
npm:nodemailer-smtp-pool	11	0.50
npm:nodemailer-smtp-transport	11	0.50
npm:nodemailer-wellknown	11	0.50
npm:noms	11	0.50
npm:npm-registry-client	11	0.50

npm:number-is-integer	11	0.50
npm:omit.js	11	0.50
npm:ono	11	0.50
npm:overlayscrollbars	11	0.50
npm:parenthesis	11	0.50
npm:parse-github-repo-url	11	0.50
npm:passport-facebook	11	0.50
npm:peek-stream	11	0.50
npm:pre-commit	11	0.50
npm:proxyquire	11	0.50
npm:qrcode	11	0.50
npm:react-addons-shallow-compare	11	0.50
npm:react-cookie	11	0.50
npm:react-csv	11	0.50
npm:react-prop-types	11	0.50
npm:react-router-redux	11	0.50
npm:referrer-policy	11	0.50
npm:regextras	11	0.50
npm:regl-splom	11	0.50
npm:remove-trailing-slash	11	0.50
npm:requireg	11	0.50
npm:rifm	11	0.50
npm:rndm	11	0.50
npm:rollup-plugin-json	11	0.50
npm:scope-analyzer	11	0.50

npm:semantic-ui-css	11	0.50
npm:sequelize-cli	11	0.50
npm:sequelize-pool	11	0.50
npm:simple-html-tokenizer	11	0.50
npm:smtp-connection	11	0.50
npm:sponge-case	11	0.50
npm:ssh2	11	0.50
npm:stats-webpack-plugin	11	0.50
npm:stdin-discarder	11	0.50
npm:stream-cache	11	0.50
npm:strict-event-emitter	11	0.50
npm:string-split-by	11	0.50
npm:string-to-arraybuffer	11	0.50
npm:swagger-ui-express	11	0.50
npm:ternary-stream	11	0.50
npm:text-segmentation	11	0.50
npm:timespan	11	0.50
npm:tsort	11	0.50
npm:tsup	11	0.50
npm:unorm	11	0.50
npm:utif	11	0.50
npm:utrie	11	0.50
npm:vega-canvas	11	0.50
npm:vega-crossfilter	11	0.50
npm:vega-dataflow	11	0.50

npm:vega-encode	11	0.50
npm:vega-event-selector	11	0.50
npm:vega-expression	11	0.50
npm:vega-force	11	0.50
npm:vega-geo	11	0.50
npm:vega-hierarchy	11	0.50
npm:vega-loader	11	0.50
npm:vega-parser	11	0.50
npm:vega-projection	11	0.50
npm:vega-runtime	11	0.50
npm:vega-scale	11	0.50
npm:vega-scenegraph	11	0.50
npm:vega-transforms	11	0.50
npm:vega-typings	11	0.50
npm:vega-util	11	0.50
npm:vega-view	11	0.50
npm:vega-view-transforms	11	0.50
npm:vega-voronoi	11	0.50
npm:vega-wordcloud	11	0.50
npm:vfile-reporter	11	0.50
npm:vtk.js	11	0.50
npm:vue-property-decorator	11	0.50
npm:walk-sync	11	0.50
npm:ware	11	0.50
npm:wkt-parser	11	0.50

npm:workbox-broadcast-cache-update	11	0.50
npm:workbox-cache-expiration	11	0.50
npm:xstate	11	0.50
npm:@algolia/autocomplete-plugin-algolia-insights	10	0.45
npm:@angular-eslint/builder	10	0.45
npm:@angular-eslint/bundled-angular-compiler	10	0.45
npm:@angular-eslint/schematics	10	0.45
npm:@angular-eslint/template-parser	10	0.45
npm:@angular-eslint/utils	10	0.45
npm:@antfu/utils	10	0.45
npm:@ardatan/relay-compiler	10	0.45
npm:@aws-sdk/middleware-endpoint	10	0.45
npm:@babel/node	10	0.45
npm:@chakra-ui/breakpoint-utils	10	0.45
npm:@chakra-ui/dom-utils	10	0.45
npm:@chakra-ui/event-utils	10	0.45
npm:@chakra-ui/lazy-utils	10	0.45
npm:@chakra-ui/number-utils	10	0.45
npm:@chakra-ui/react-children-utils	10	0.45

npm:@chakra-ui/react-context	10	0.45
npm:@chakra-ui/react-types	10	0.45
npm:@chakra-ui/react-use- animation-state	10	0.45
npm:@chakra-ui/react-use- callback-ref	10	0.45
npm:@chakra-ui/react-use- controllable-state	10	0.45
npm:@chakra-ui/react-use- disclosure	10	0.45
npm:@chakra-ui/react-use-event- listener	10	0.45
npm:@chakra-ui/react-use-focus- effect	10	0.45
npm:@chakra-ui/react-use-focus- on-pointer-down	10	0.45
npm:@chakra-ui/react-use-interval	10	0.45
npm:@chakra-ui/react-use-latest- ref	10	0.45
npm:@chakra-ui/react-use-merge- refs	10	0.45
npm:@chakra-ui/react-use-outside- click	10	0.45
npm:@chakra-ui/react-use-pan- event	10	0.45

npm:@chakra-ui/react-use-previous	10	0.45
npm:@chakra-ui/react-use-size	10	0.45
npm:@chakra-ui/react-use-timeout	10	0.45
npm:@chakra-ui/react-use-update-effect	10	0.45
npm:@changesets/apply-release-plan	10	0.45
npm:@changesets/assemble-release-plan	10	0.45
npm:@changesets/changelog-git	10	0.45
npm:@changesets/cli	10	0.45
npm:@changesets/config	10	0.45
npm:@changesets/errors	10	0.45
npm:@changesets/get-dependents-graph	10	0.45
npm:@changesets/get-release-plan	10	0.45
npm:@changesets/get-version-range-type	10	0.45
npm:@changesets/git	10	0.45
npm:@changesets/logger	10	0.45
npm:@changesets/parse	10	0.45
npm:@changesets/pre	10	0.45
npm:@changesets/read	10	0.45
npm:@changesets/types	10	0.45

npm:@changesets/write	10	0.45
npm:@codemirror/autocomplete	10	0.45
npm:@docusaurus/logger	10	0.45
npm:@docusaurus/module-type-aliases	10	0.45
npm:@docusaurus/theme-translations	10	0.45
npm:@expo/bunyan	10	0.45
npm:@expo/osascript	10	0.45
npm:@expo/prebuild-config	10	0.45
npm:@fontsource/roboto	10	0.45
npm:@fullhuman/postcss-purgecss	10	0.45
npm:@graphql-tools/graphql-file-loader	10	0.45
npm:@graphql-tools/import	10	0.45
npm:@graphql-tools/json-file-loader	10	0.45
npm:@graphql-tools/url-loader	10	0.45
npm:@jimp/plugin-circle	10	0.45
npm:@jimp/plugin-fisheye	10	0.45
npm:@jimp/plugin-shadow	10	0.45
npm:@jimp/plugin-threshold	10	0.45
npm:@jupyterlab/mainmenu	10	0.45
npm:@jupyterlab/rendermime	10	0.45
npm:@lmdb/lmdb-darwin-arm64	10	0.45

npm:@lmdb/lmdb-darwin-x64	10	0.45
npm:@lmdb/lmdb-linux-arm	10	0.45
npm:@lmdb/lmdb-linux-arm64	10	0.45
npm:@lmdb/lmdb-linux-x64	10	0.45
npm:@lmdb/lmdb-win32-x64	10	0.45
npm:@lumino/application	10	0.45
npm:@manypkg/find-root	10	0.45
npm:@manypkg/get-packages	10	0.45
npm:@material-ui/pickers	10	0.45
npm:@material/checkbox	10	0.45
npm:@material/icon-button	10	0.45
npm:@material/linear-progress	10	0.45
npm:@material/shape	10	0.45
npm:@mdx-js/loader	10	0.45
npm:@microsoft/tsdoc	10	0.45
npm:@nestjs/common	10	0.45
npm:@nestjs/core	10	0.45
npm:@ng-bootstrap/ng-bootstrap	10	0.45
npm:@parcel/fs-search	10	0.45
npm:@parcel/hash	10	0.45
npm:@phosphor/widgets	10	0.45
npm:@plotly/point-cluster	10	0.45
npm:@radix-ui/react-accordion	10	0.45
npm:@rails/webpacker	10	0.45
npm:@react-navigation/elements	10	0.45

npm:@react-spring/three	10	0.45
npm:@storybook/builder- webpack4	10	0.45
npm:@storybook/channel- websocket	10	0.45
npm:@storybook/react-docgen- typescript-plugin	10	0.45
npm:@storybook/store	10	0.45
npm:@sveltejs/adapter-auto	10	0.45
npm:@tanstack/query-core	10	0.45
npm:@tensorflow/tfjs-converter	10	0.45
npm:@tensorflow/tfjs-core	10	0.45
npm:@tensorflow/tfjs-data	10	0.45
npm:@tensorflow/tfjs-layers	10	0.45
npm:@turf/boolean-clockwise	10	0.45
npm:@turf/clone	10	0.45
npm:@turf/distance	10	0.45
npm:@types/chrome	10	0.45
npm:@types/codemirror	10	0.45
npm:@types/file-saver	10	0.45
npm:@types/html-minifier	10	0.45
npm:@types/katex	10	0.45
npm:@types/multer	10	0.45
npm:@types/passport-strategy	10	0.45
npm:@types/relateurl	10	0.45

npm:@types/tern	10	0.45
npm:@urql/core	10	0.45
npm:@vercel/nft	10	0.45
npm:@vue/cli-plugin-unit-jest	10	0.45
npm:@yarnpkg/fslib	10	0.45
npm:@yarnpkg/libzip	10	0.45
npm:@zag-js/element-size	10	0.45
npm:@zag-js/focus-visible	10	0.45
npm:@zeit/schemas	10	0.45
npm:ace-builds	10	0.45
npm:angular-animate	10	0.45
npm:ansi-escape-sequences	10	0.45
npm:apollo-link-error	10	0.45
npm:apollo-reporting-protobuf	10	0.45
npm:apollo-server-caching	10	0.45
npm:async-listener	10	0.45
npm:babel-plugin-import	10	0.45
npm:babel-plugin-react-transform	10	0.45
npm:bcrypt-nodejs	10	0.45
npm:blessed	10	0.45
npm:breakword	10	0.45
npm:buffer-to-vinyl	10	0.45
npm:call-limit	10	0.45
npm:chrome-launcher	10	0.45
npm:chunkd	10	0.45

npm:ci-parallel-vars	10	0.45
npm:clean-yaml-object	10	0.45
npm:color2k	10	0.45
npm:conventional-changelog-cli	10	0.45
npm:cose-base	10	0.45
npm:css-selector-parser	10	0.45
npm:cytoscape-cose-bilkent	10	0.45
npm:dedent-js	10	0.45
npm:deep-assign	10	0.45
npm:dev-null	10	0.45
npm:dfa	10	0.45
npm:dns-prefetch-control	10	0.45
npm:duplicate-package-checker- webpack-plugin	10	0.45
npm:electron-rebuild	10	0.45
npm:emitter-listener	10	0.45
npm:emojilib	10	0.45
npm:emotion	10	0.45
npm:errlop	10	0.45
npm:escape-latex	10	0.45
npm:eth-block-tracker	10	0.45
npm:eth-query	10	0.45
npm:ethereum-common	10	0.45
npm:ethereumjs-account	10	0.45
npm:ethereumjs-block	10	0.45

npm:exec-async	10	0.45
npm:express-flash	10	0.45
npm:express-jwt	10	0.45
npm:fast-base64-decode	10	0.45
npm:fbjs-scripts	10	0.45
npm:figlet	10	0.45
npm:fill-keys	10	0.45
npm:find-npm-prefix	10	0.45
npm:flatbuffers	10	0.45
npm:flora-colossus	10	0.45
npm:frameguard	10	0.45
npm:freeport-async	10	0.45
npm:galactus	10	0.45
npm:gatsby	10	0.45
npm:gentle-fs	10	0.45
npm:get-port-please	10	0.45
npm:git-config-path	10	0.45
npm:gl-vec3	10	0.45
npm:glob-all	10	0.45
npm:global-directory	10	0.45
npm:googleapis-common	10	0.45
npm:grunt-contrib-coffee	10	0.45
npm:grunt-postcss	10	0.45
npm:grunt-svgmin	10	0.45
npm:gulp-decompress	10	0.45

npm:gulp-flatten	10	0.45
npm:gulp-postcss	10	0.45
npm:gulp-rev	10	0.45
npm:hooks-fixed	10	0.45
npm:human-id	10	0.45
npm:i18next-browser- languagedetector	10	0.45
npm:ienoopen	10	0.45
npm:interface-datastore	10	0.45
npm:ionicons	10	0.45
npm:is-alphanumeric	10	0.45
npm:is-bzip2	10	0.45
npm:is-fn	10	0.45
npm:is-tar	10	0.45
npm:is-zip	10	0.45
npm:items	10	0.45
npm:jest-canvas-mock	10	0.45
npm:jest-serializer-vue	10	0.45
npm:jimp-compact	10	0.45
npm:jpegtran-bin	10	0.45
npm:jscs	10	0.45
npm:json-rpc-random-id	10	0.45
npm:json-schema-typed	10	0.45
npm:ktx-parse	10	0.45
npm:ky	10	0.45

npm:layout-base	10	0.45
npm:lazy-property	10	0.45
npm:leveldown	10	0.45
npm:libcipm	10	0.45
npm:libnpx	10	0.45
npm:liftup	10	0.45
npm:lit-element	10	0.45
npm:lock-verify	10	0.45
npm:lodash.assignwith	10	0.45
npm:mailgun-js	10	0.45
npm:mcl-wasm	10	0.45
npm:md5hex	10	0.45
npm:mdast-util-compact	10	0.45
npm:mem-fs	10	0.45
npm:merge-class-names	10	0.45
npm:module-not-found-error	10	0.45
npm:moo-color	10	0.45
npm:mousetrap	10	0.45
npm:muri	10	0.45
npm:native-request	10	0.45
npm:node-machine-id	10	0.45
npm:nodeify	10	0.45
npm:non-layered-tidy-tree-layout	10	0.45
npm:normalize-html-whitespace	10	0.45
npm:npm-logical-tree	10	0.45

npm:ol	10	0.45
npm:optional-require	10	0.45
npm:parcel	10	0.45
npm:parse-git-config	10	0.45
npm:parse-github-url	10	0.45
npm:passport-twitter	10	0.45
npm:path-proxy	10	0.45
npm:pm2	10	0.45
npm:png-js	10	0.45
npm:prettier-plugin-organize- imports	10	0.45
npm:promise-to-callback	10	0.45
npm:protobufjs-cli	10	0.45
npm:python-shell	10	0.45
npm:qw	10	0.45
npm:react-freeze	10	0.45
npm:react-google-charts	10	0.45
npm:react-hot-toast	10	0.45
npm:react-konva	10	0.45
npm:react-player	10	0.45
npm:redbox-react	10	0.45
npm:remark-math	10	0.45
npm:response-iterator	10	0.45
npm:robust-orientation	10	0.45
npm:robust-scale	10	0.45

npm:robust-subtract	10	0.45
npm:robust-sum	10	0.45
npm:rollup-plugin-replace	10	0.45
npm:safe-identifier	10	0.45
npm:secure-json-parse	10	0.45
npm:seed-random	10	0.45
npm:signed-varint	10	0.45
npm:smartwrap	10	0.45
npm:sorted-union-stream	10	0.45
npm:spawndamnit	10	0.45
npm:ssr-window	10	0.45
npm:staged-git-files	10	0.45
npm:stream-iterate	10	0.45
npm:stylelint-config-prettier	10	0.45
npm:superstruct	10	0.45
npm:supertap	10	0.45
npm:swiper	10	0.45
npm:to-array-buffer	10	0.45
npm:tslint-config-prettier	10	0.45
npm:tty-table	10	0.45
npm:tv4	10	0.45
npm:two-product	10	0.45
npm:two-sum	10	0.45
npm:typechain	10	0.45
npm:unicode-properties	10	0.45

npm:uuid-browser	10	0.45
npm:validate.io-nonnegative-integer	10	0.45
npm:vega-embed	10	0.45
npm:vega-format	10	0.45
npm:vega-functions	10	0.45
npm:vega-regression	10	0.45
npm:vega-selections	10	0.45
npm:vega-time	10	0.45
npm:vfile-sort	10	0.45
npm:vfile-statistics	10	0.45
npm:victory	10	0.45
npm:vinyl-assign	10	0.45
npm:vow	10	0.45
npm:vow-fs	10	0.45
npm:vow-queue	10	0.45
npm:vscode-nls	10	0.45
npm:warn-once	10	0.45
npm:webdriverio	10	0.45
npm:webpack-filter-warnings-plugin	10	0.45
npm:wonka	10	0.45
npm:worker-plugin	10	0.45
npm:wrap-fn	10	0.45
npm:xml-escape	10	0.45

npm:xterm	10	0.45
npm:xterm-addon-fit	10	0.45
npm:xxhash-wasm	10	0.45
npm:zrender	10	0.45
nuget:Newtonsoft.Json	131	25.84
nuget:Microsoft.NET.Test.Sdk	115	22.68
nuget:PRISM-Library	88	17.36
nuget:NUnit	81	15.98
nuget:xunit	68	13.41
nuget:xunit.runner.visualstudio	65	12.82
nuget:MSTest.TestAdapter	52	10.26
nuget:MSTest.TestFramework	52	10.26
nuget:coverlet.collector	51	10.06
nuget:System.Data.SQLite.Core	38	7.50
nuget:Moq	35	6.90
nuget:Swashbuckle.AspNetCore	33	6.51
nuget:NUnit3TestAdapter	32	6.31
nuget:MathNet.Numerics	30	5.92
nuget:Microsoft.EntityFrameworkCore	29	5.72
nuget:Microsoft.EntityFrameworkCore.Tools	29	5.72
nuget:PRISM-DatabaseUtils	29	5.72
nuget:jQuery	28	5.52
nuget:Microsoft.EntityFrameworkCore.SqlServer	27	5.33
nuget:Microsoft.Web.Infrastructure	27	5.33
nuget:OxyPlot.Wpf	27	5.33

nuget:EntityFramework	26	5.13
nuget:System.Runtime.CompilerServices.Unsafe	26	5.13
nuget:Microsoft.AspNet.WebApi.Client	25	4.93
nuget:Microsoft.Extensions.Logging.Abstractions	25	4.93
nuget:System.ValueTuple	25	4.93
nuget:CsvHelper	24	4.73
nuget:Microsoft.AspNet.Mvc	23	4.54
nuget:Microsoft.AspNet.Razor	22	4.34
nuget:Microsoft.AspNet.WebPages	22	4.34
nuget:System.Numerics.Vectors	22	4.34
nuget:System.Memory	21	4.14
nuget:System.Threading.Tasks.Extensions	21	4.14
nuget:Microsoft.AspNet.WebApi.Core	20	3.94
nuget:Microsoft.AspNet.WebApi.WebHost	20	3.94
nuget:Microsoft.AspNet.WebApi	19	3.75
nuget:System Buffers	19	3.75
nuget:Microsoft.CodeDom.Providers.DotNetCompilerPlatform	18	3.55
nuget:SharpZipLib	18	3.55
nuget:FluentAssertions	17	3.35
nuget:Microsoft.CSharp	17	3.35
nuget:Microsoft.EntityFrameworkCore.Design	17	3.35
nuget:Microsoft.NETCore.UniversalWindowsPlatform	17	3.35
nuget:Microsoft.SourceLink.GitHub	17	3.35
nuget:Nerdbank.GitVersioning	17	3.35
nuget:CommandLineParser	16	3.16

nuget:NETStandard.Library	16	3.16
nuget:HtmlAgilityPack	15	2.96
nuget:log4net	15	2.96
nuget:Microsoft.AspNet.Web.Optimization	15	2.96
nuget:Microsoft.Extensions.Configuration.Json	15	2.96
nuget:Microsoft.Extensions.DependencyInjection.Abstractions	15	2.96
nuget:Microsoft.Extensions.Options	15	2.96
nuget:Microsoft.VisualStudio.Web.CodeGeneration.Design	15	2.96
nuget:Modernizr	15	2.96
nuget:OxyPlot.Core	15	2.96
nuget:System.Diagnostics.DiagnosticSource	15	2.96
nuget:System.Text.Json	15	2.96
nuget:WebGrease	15	2.96
nuget:Antlr	14	2.76
nuget:AutoMapper	14	2.76
nuget:Microsoft.Extensions.DependencyInjection	14	2.76
nuget:Microsoft.Net.Compilers	14	2.76
nuget:Microsoft.Owin	14	2.76
nuget:Owin	14	2.76
nuget:PSI_Interface	14	2.76
nuget:System.Net.Http	14	2.76
nuget:System.Reflection	14	2.76
nuget:System.Text.Encoding	14	2.76
nuget:BenchmarkDotNet	13	2.56
nuget:bootstrap	13	2.56

nuget:Microsoft.Extensions.Logging	13	2.56
nuget:Microsoft.Owin.Security	13	2.56
nuget:System.Collections.Concurrent	13	2.56
nuget:System.Data.SQLite	13	2.56
nuget:System.IO	13	2.56
nuget:System.IO.Compression	13	2.56
nuget:System.Resources.ResourceManager	13	2.56
nuget:System.Runtime	13	2.56
nuget:System.Runtime.Extensions	13	2.56
nuget:System.Text.Encoding.Extensions	13	2.56
nuget:System.Xml.ReaderWriter	13	2.56
nuget:System.Xml.XDocument	13	2.56
nuget:jQuery.Validation	12	2.37
nuget:Microsoft.CodeAnalysis.CSharp	12	2.37
nuget:Microsoft.Extensions.Hosting	12	2.37
nuget:Microsoft.Owin.Host.SystemWeb	12	2.37
nuget:Microsoft.VisualStudio.Azure.Containers.Tools.Targets	12	2.37
nuget:Serilog	12	2.37
nuget:StyleCop.Analyzers	12	2.37
nuget:System.Collections	12	2.37
nuget:System.Collections.Immutable	12	2.37
nuget:System.Diagnostics.Debug	12	2.37
nuget:System.Globalization	12	2.37
nuget:System.Linq	12	2.37
nuget:System.Linq.Expressions	12	2.37

nuget:System.Runtime.InteropServices	1.2	RuntimeInformation	2.37
nuget:System.Security.Cryptography	1.2	Algorithms	2.37
nuget:System.Text.RegularExpressions	1.2		2.37
nuget:System.Threading	1.2		2.37
nuget:Microsoft.AspNetCore.Authentication	1.1	JwtBearer	2.17
nuget:Microsoft.AspNetCore.Identity	1.1	EntityFrameworkCore	2.17
nuget:Microsoft.CodeAnalysis.Analyzers	1.1		2.17
nuget:Microsoft.Extensions.Configuration	1.1		2.17
nuget:PRISMWin-Library	1.1		2.17
nuget:ProteinFileReader	1.1		2.17
nuget:System.Diagnostics.Tools	1.1		2.17
nuget:System.Reflection.Extensions	1.1		2.17
nuget:System.Runtime.InteropServices	1.1		2.17
nuget:System.Runtime.Numerics	1.1		2.17
nuget:System.Security.Cryptography	1.1	X509Certificates	2.17
nuget:System.Threading.Tasks	1.1		2.17
nuget:Autofac	1.0		1.97
nuget:FSharp.Core	1.0		1.97
nuget:Microsoft.AspNetCore.Identity.Core	1.0		1.97
nuget:Microsoft.AspNetCore.Identity.Owin	1.0		1.97
nuget:Microsoft.AspNetCore.Mvc.Newtonsoft.Json	1.0		1.97
nuget:Microsoft.Bcl.AsyncInterfaces	1.0		1.97
nuget:Microsoft.EntityFrameworkCore.Sqlite	1.0		1.97
nuget:Microsoft.Net.Http	1.0		1.97
nuget:Microsoft.NETCore.Platforms	1.0		1.97

nuget:Microsoft.Owin.Security.Cookies	10	1.97
nuget:Microsoft.Owin.Security.OAuth	10	1.97
nuget:System.ObjectModel	10	1.97
nuget:System.Security.Cryptography.Encoding	10	1.97
nuget:System.Security.Cryptography.Primitives	10	1.97
nuget:UIMFLibrary	10	1.97
pip:numpy	2385	48.85
pip:scipy	1489	30.50
pip:pandas	1472	30.15
pip:matplotlib	1431	29.31
pip:six	1332	27.28
pip:requests	1280	26.22
pip:python-dateutil	1125	23.04
pip:pytz	1024	20.98
pip:pillow	960	19.66
pip:certifi	953	19.52
pip:scikit-learn	932	19.09
pip:urllib3	921	18.87
pip:pyarsing	900	18.44
pip:jinja2	895	18.33
pip:idna	876	17.94
pip:pyyaml	839	17.19
pip:tqdm	838	17.17
pip:torch	833	17.06
pip:markupsafe	820	16.80

pip:click	737	15.10
pip:cycler	676	13.85
pip:packaging	676	13.85
pip:pytest	643	13.17
pip:pygments	627	12.84
pip:kiwisolver	580	11.88
pip:decorator	579	11.86
pip:werkzeug	569	11.66
pip:ipython	548	11.22
pip:attrs	545	11.16
pip:protobuf	522	10.69
pip:sphinx	515	10.55
pip:typing-extensions	515	10.55
pip:chardet	503	10.30
pip:flask	502	10.28
pip:tornado	486	9.95
pip:h5py	466	9.55
pip:wcwidth	464	9.50
pip:joblib	456	9.34
pip:ffi	448	9.18
pip:pyzmq	448	9.18
pip:importlib-metadata	447	9.16
pip:traitlets	446	9.14
pip:charset-normalizer	445	9.12
pip:networkx	444	9.09

pip:torchvision	443	9.07
pip:zipp	442	9.05
pip:jjsonschema	441	9.03
pip:tensorflow	440	9.01
pip:pyparser	428	8.77
pip:prompt-toolkit	421	8.62
pip:colorama	415	8.50
pip:pickleshare	414	8.48
pip:seaborn	414	8.48
pip:pexpect	411	8.42
pip:ptyprocess	403	8.25
pip:beautifulsoup4	396	8.11
pip:jedi	390	7.99
pip:jupyter-core	390	7.99
pip:ipykernel	380	7.78
pip:jupyter-client	376	7.70
pip:ipython-genutils	375	7.68
pip:parso	373	7.64
pip:psutil	368	7.54
pip:docutils	363	7.44
pip:opencv-python	354	7.25
pip:wrapt	343	7.03
pip:nbformat	342	7.01
pip:itsdangerous	338	6.92
pip:markdown	338	6.92

pip:bleach	337	6.90
pip:mistune	333	6.82
pip:webencodings	333	6.82
pip:setuptools	332	6.80
pip:future	331	6.78
pip:nbconvert	323	6.62
pip:backcall	320	6.55
pip:notebook	320	6.55
pip:ipywidgets	318	6.51
pip:tensorboard	315	6.45
pip:terminado	306	6.27
pip:scikit-image	305	6.25
pip: lxml	300	6.15
pip:entrypoints	297	6.08
pip:pyasn1	297	6.08
pip:absl-py	288	5.90
pip:jupyter	287	5.88
pip:pandocfilters	287	5.88
pip:grpcio	285	5.84
pip:cryptography	280	5.74
pip:pluggy	280	5.74
pip:nlTK	277	5.67
pip:defusedxml	272	5.57
pip:pyrsistent	271	5.55
pip:cython	270	5.53

pip:termcolor	266	5.45
pip:sqlalchemy	263	5.39
pip:cachetools	256	5.24
pip:pyasn1-modules	255	5.22
pip:gunicorn	254	5.20
pip:black	248	5.08
pip:rsa	248	5.08
pip:send2trash	247	5.06
pip:widgetsnbextension	246	5.04
pip:prometheus-client	245	5.02
pip:soupsieve	240	4.92
pip:fonttools	239	4.90
pip:filelock	238	4.88
pip:oauthlib	238	4.88
pip:threadpoolctl	236	4.83
pip:transformers	234	4.79
pip:coverage	231	4.73
pip:toml	230	4.71
pip:google-auth	227	4.65
pip:regex	227	4.65
pip:jupyter-console	226	4.63
pip:platformdirs	226	4.63
pip:qtconsole	226	4.63
pip:isort	224	4.59
pip:mccabe	224	4.59

pip:requests-oauthlib	224	4.59
pip:babel	223	4.57
pip:imageio	222	4.55
pip:keras	222	4.55
pip:tomli	220	4.51
pip:flake8	217	4.44
pip:wheel	216	4.42
pip:testpath	214	4.38
pip:py	205	4.20
pip:django	201	4.12
pip:gast	200	4.10
pip:sphinx-rtd-theme	196	4.01
pip:pylint	195	3.99
pip:appnope	191	3.91
pip:boto3	190	3.89
pip:pytest-cov	189	3.87
pip:more-itertools	187	3.83
pip:matplotlib-inline	186	3.81
pip:nest-asyncio	186	3.81
pip:pycodestyle	185	3.79
pip:numba	183	3.75
pip:google-auth-oauthlib	182	3.73
pip:iniconfig	182	3.73
pip:plotly	182	3.73
pip:dill	179	3.67

pip:keras-preprocessing	178	3.65
pip:mypy-extensions	176	3.61
pip:snowballstemmer	172	3.52
pip:sklearn	166	3.40
pip:pydantic	165	3.38
pip:cloudpickle	163	3.34
pip:pathspec	160	3.28
pip:statsmodels	160	3.28
pip:alabaster	159	3.26
pip:mock	157	3.22
pip:appdirs	156	3.20
pip:imagesize	156	3.20
pip:google-pasta	155	3.17
pip:botocore	154	3.15
pip:pyflakes	154	3.15
pip:html5lib	152	3.11
pip:aiohttp	151	3.09
pip:python	151	3.09
pip:pywavelets	151	3.09
pip:greenlet	150	3.07
pip:tabulate	149	3.05
pip:pip	147	3.01
pip:python-dotenv	147	3.01
pip:tensorflow-estimator	147	3.01
pip:gitpython	146	2.99

pip:nbclient	145	2.97
pip:websocket-client	144	2.95
pip:contourpy	143	2.93
pip:jupyterlab-pygments	143	2.93
pip:pyopenssl	143	2.93
pip:jmespath	141	2.89
pip:asttokens	140	2.87
pip:debugpy	140	2.87
pip:psycogp2	139	2.85
pip:sniffio	138	2.83
pip:toolz	138	2.83
pip:argon2-cffi	137	2.81
pip:nose	137	2.81
pip:simplegeneric	137	2.81
pip:async-timeout	136	2.79
pip:astroid	135	2.77
pip:executing	135	2.77
pip:jupyterlab	134	2.74
pip:pure-eval	134	2.74
pip:stack-data	134	2.74
pip:virtualenv	134	2.74
pip:importlib-resources	133	2.72
pip:openpyxl	133	2.72
pip:sympy	133	2.72
pip:astor	132	2.70

pip:lazy-object-proxy	132	2.70
pip:opt-einsum	132	2.70
pip:s3transfer	131	2.68
pip:tzdata	131	2.68
pip:shapely	130	2.66
pip:sphinxcontrib-serializinghtml	130	2.66
pip:sphinxcontrib-applehelp	129	2.64
pip:sphinxcontrib-devhelp	129	2.64
pip:sphinxcontrib-htmlhelp	129	2.64
pip:sphinxcontrib-qthelp	129	2.64
pip:qtpy	128	2.62
pip:yaml	126	2.58
pip:llvmlite	125	2.56
pip:sphinxcontrib-jsmath	125	2.56
pip:msgpack	124	2.54
pip:mpmath	123	2.52
pip:multidict	123	2.52
pip:mypy	122	2.50
pip:tensorboard-plugin-wit	122	2.50
pip:pyjwt	121	2.48
pip:anyio	120	2.46
pip:xlrd	118	2.42
pip:olefile	116	2.38
pip:tensorboardx	116	2.38
pip:exceptiongroup	113	2.31

pip:pysocks	113	2.31
pip:tensorflow-gpu	113	2.31
pip:spacy	111	2.27
pip:fsspec	108	2.21
pip:pre-commit	108	2.21
pip:dask	107	2.19
pip:flask-cors	107	2.19
pip:pyarrow	107	2.19
pip:gitdb	105	2.15
pip:patsy	105	2.15
pip:pymongo	105	2.15
pip:sortedcontainers	105	2.15
pip:tokenizers	105	2.15
pip:jupyterlab-widgets	104	2.13
pip:sqlparse	104	2.13
pip:blinker	103	2.11
pip:keras-applications	103	2.11
pip:smmmap	103	2.11
pip:atomicwrites	102	2.09
pip:ruamel-yaml	101	2.07
pip:gensim	99	2.03
pip:astunparse	98	2.01
pip:distlib	98	2.01
pip:fastjsonschema	98	2.01
pip:et-xmlfile	96	1.97

pip:gym	96	1.97
pip:jupyterlab-server	96	1.97
pip:mkl-fft	96	1.97
pip:pbr	96	1.97
pip:json5	94	1.93
pip:psycog2-binary	93	1.90
pip:pywin32	93	1.90
pip:simplejson	93	1.90
pip:singledispatch	91	1.86
pip:watchdog	91	1.86
pip:rich	90	1.84
pip:tiffle	90	1.84
pip:ply	89	1.82
pip:huggingface-hub	88	1.80
pip:redis	88	1.80
pip:mkl-random	86	1.76
pip:numexpr	86	1.76
pip:sentencepiece	86	1.76
pip:argon2-ffi-bindings	84	1.72
pip:h11	84	1.72
pip:tensorboard-data-server	84	1.72
pip:typed-ast	84	1.72
pip:autopep8	83	1.70
pip:boto	83	1.70
pip:mkl-service	83	1.70

pip:torchaudio	83	1.70
pip:docopt	82	1.68
pip:flatbuffers	82	1.68
pip:selenium	82	1.68
pip:aiohttp	81	1.66
pip:configparser	81	1.66
pip:googleapis-common-protos	81	1.66
pip:jupyter-server	81	1.66
pip:tenacity	81	1.66
pip:tinycss2	81	1.66
pip:asn1crypto	80	1.64
pip:frozenset	80	1.64
pip:httplib2	80	1.64
pip:keyring	80	1.64
pip:markdown-it-py	80	1.64
pip:bs4	78	1.60
pip:mako	77	1.58
pip:alembic	76	1.56
pip:arrow	76	1.56
pip:argparse	75	1.54
pip:asgiref	75	1.54
pip:datasets	75	1.54
pip:mdurl	75	1.54
pip:pytorch-lightning	74	1.52
pip:gevent	73	1.50

pip:librosa	73	1.50
pip:pkginfo	73	1.50
pip:zope-interface	73	1.50
pip:biopython	72	1.47
pip:comm	72	1.47
pip:fastapi	72	1.47
pip:jsonpointer	72	1.47
pip:bokeh	71	1.45
pip:enum34	71	1.45
pip:requests-toolbelt	71	1.45
pip:dataclasses	70	1.43
pip:uvicorn	70	1.43
pip:flask-sqlalchemy	69	1.41
pip:pyqt5	69	1.41
pip:google-api-core	68	1.39
pip:tables	68	1.39
pip:graphviz	67	1.37
pip:djangorestframework	66	1.35
pip:tzlocal	66	1.35
pip:websockets	66	1.35
pip:mkdocs	65	1.33
pip:nodeenv	65	1.33
pip:cfgv	64	1.31
pip:contextlib2	64	1.31
pip:multiprocess	64	1.31

pip:pywinpty	64	1.31
pip:smart-open	64	1.31
pip:wandb	64	1.31
pip:identify	63	1.29
pip:pyproj	63	1.29
pip:retrying	63	1.29
pip:twine	63	1.29
pip:wsgiref	63	1.29
pip:xgboost	63	1.29
pip:async-generator	62	1.27
pip:bcrypt	62	1.27
pip:einops	62	1.27
pip:geopandas	62	1.27
pip:pathlib2	62	1.27
pip:streamlit	62	1.27
pip:unidecode	62	1.27
pip:dnspython	61	1.25
pip:wincertstore	61	1.25
pip:breathe	60	1.23
pip:notebook-shim	60	1.23
pip:openai	60	1.23
pip:secretstorage	60	1.23
pip:timm	60	1.23
pip:cytoolz	59	1.21
pip:google-api-python-client	58	1.19

pip:nbclassic	58	1.19
pip:pysam	58	1.19
pip:cymem	57	1.17
pip:murmurhash	57	1.17
pip:preshed	57	1.17
pip:sacremoses	56	1.15
pip:sentry-sdk	56	1.15
pip:thinc	56	1.15
pip:xlsxwriter	56	1.15
pip:docker	55	1.13
pip:loket	55	1.13
pip:partd	55	1.13
pip:path-py	55	1.13
pip:prettytable	55	1.13
pip:backports-abc	54	1.11
pip:backports-functools-lru-cache	54	1.11
pip:deprecated	54	1.11
pip:numpydoc	54	1.11
pip:dash	53	1.09
pip:typer	53	1.09
pip:ujson	53	1.09
pip:webcolors	53	1.09
pip:xarray	53	1.09
pip:xmltodict	53	1.09
pip:backports-ssl-match-hostname	52	1.07

pip:brotlipy	52	1.07
pip:myst-parser	52	1.07
pip:pynacl	52	1.07
pip:wasabi	52	1.07
pip:zict	52	1.07
pip:blis	51	1.04
pip:click-plugins	51	1.04
pip:jdcal	51	1.04
pip:pathtools	51	1.04
pip:pycurl	51	1.04
pip:pydot	51	1.04
pip:pyserial	51	1.04
pip:backports-shutil-get-terminal-size	50	1.02
pip:celery	50	1.02
pip:distributed	50	1.02
pip:heapdict	50	1.02
pip:isodate	50	1.02
pip:pep8	50	1.02
pip:pycrypto	50	1.02
pip:rfc3339-validator	50	1.02
pip:srsly	50	1.02
pip:tox	50	1.02
pip:mkdocs-material	49	1.00
pip:munch	49	1.00

pip:wtfoms	49	1.00
pip:bottleneck	48	0.98
pip:paramiko	48	0.98
pip:ruamel-yaml-club	48	0.98
pip:altair	47	0.96
pip:flask-wtf	47	0.96
pip:httplib	47	0.96
pip:pyglet	47	0.96
pip:python-json-logger	47	0.96
pip:starlette	47	0.96
pip:subprocess32	47	0.96
pip:xlwt	47	0.96
pip:astropy	46	0.94
pip:gdown	46	0.94
pip:jax	46	0.94
pip:jeepney	46	0.94
pip:pyqt5-sip	46	0.94
pip:torchmetrics	46	0.94
pip:coveralls	45	0.92
pip:cssselect	45	0.92
pip:fqdn	45	0.92
pip:isoduration	45	0.92
pip:rfc3986	45	0.92
pip:sphinx-copybutton	45	0.92
pip:tomlkit	45	0.92

pip:uri-template	45	0.92
pip:futures	44	0.90
pip:jupyter-events	44	0.90
pip:libclang	44	0.90
pip:pycosat	44	0.90
pip:recommonmark	44	0.90
pip:rfc3986-validator	44	0.90
pip:tblib	44	0.90
pip:fiona	43	0.88
pip:google-cloud-core	43	0.88
pip:jupyter-server-terminals	43	0.88
pip:omegaconf	43	0.88
pip:promise	43	0.88
pip:twisted	43	0.88
pip:uritemplate	43	0.88
pip:docker-pycreds	42	0.86
pip:ftfy	42	0.86
pip:google-cloud-storage	42	0.86
pip:multipledispatch	42	0.86
pip:pycocotools	42	0.86
pip:pygame	42	0.86
pip:text-unidecode	42	0.86
pip:cached-property	41	0.84
pip:catalogue	41	0.84
pip:ipdb	41	0.84

pip:overrides	41	0.84
pip:stevedore	41	0.84
pip:tensorflow-probability	41	0.84
pip:unicodcsv	41	0.84
pip:yacs	41	0.84
pip:accelerate	40	0.82
pip:elasticsearch	40	0.82
pip:funcsig	40	0.82
pip:gignoreadline	40	0.82
pip:hydra-core	40	0.82
pip:plac	40	0.82
pip:pymysql	40	0.82
pip:rope	40	0.82
pip:shortuuid	40	0.82
pip:soundfile	40	0.82
pip:sphinxcontrib-bibtex	40	0.82
pip:tk	40	0.82
pip:validators	40	0.82
pip:build	39	0.80
pip:cligj	39	0.80
pip:mdit-py-plugins	39	0.80
pip:openssl	39	0.80
pip:pytest-xdist	39	0.80
pip:sqlite	39	0.80
pip:zlib	39	0.80

pip:commonmark	38	0.78
pip:dash-core-components	38	0.78
pip:dash-html-components	38	0.78
pip:flask-login	38	0.78
pip:jsonschema-specifications	38	0.78
pip:referencing	38	0.78
pip:rpds-py	38	0.78
pip:setuputils-scm	38	0.78
pip:sip	38	0.78
pip:yapf	38	0.78
pip:alumentations	37	0.76
pip:brotli	37	0.76
pip:distro	37	0.76
pip:fire	37	0.76
pip:google-auth-httpplib2	37	0.76
pip:intel-openmp	37	0.76
pip:mkl	37	0.76
pip:natsort	37	0.76
pip:nibabel	37	0.76
pip:pydata-sphinx-theme	37	0.76
pip:pydocstyle	37	0.76
pip:pyodbc	37	0.76
pip:sphinx-autodoc-typehints	37	0.76
pip:conda	36	0.74
pip:flask-compress	36	0.74

pip:functools32	36	0.74
pip:ninja	36	0.74
pip:pkgutil-resolve-name	36	0.74
pip:responses	36	0.74
pip:rtree	36	0.74
pip:tensorflow-io-gcs-filesystem	36	0.74
pip:whitenoise	36	0.74
pip:ca-certificates	35	0.72
pip:django-cors-headers	35	0.72
pip:dominate	35	0.72
pip:google-resumable-media	35	0.72
pip:readme-renderer	35	0.72
pip:sphinxcontrib-websupport	35	0.72
pip:torch-geometric	35	0.72
pip:typing	35	0.72
pip:xxhash	35	0.72
pip:xz	35	0.72
pip:flask-restful	34	0.70
pip:geopy	34	0.70
pip:glob2	34	0.70
pip:httpcore	34	0.70
pip:oauth2client	34	0.70
pip:opencv-python-headless	34	0.70
pip:portalocker	34	0.70
pip:pytest-mock	34	0.70

pip:rdflib	34	0.70
pip:aniso8601	33	0.68
pip:backports-weakref	33	0.68
pip:bitarray	33	0.68
pip:clyent	33	0.68
pip:imageio-ffmpeg	33	0.68
pip:jaraco-classes	33	0.68
pip:jsonpickle	33	0.68
pip:kombu	33	0.68
pip:lockfile	33	0.68
pip:mpi4py	33	0.68
pip:opencv-contrib-python	33	0.68
pip:pybind11	33	0.68
pip:ruff	33	0.68
pip:anaconda-client	32	0.66
pip:dash-table	32	0.66
pip:dj-database-url	32	0.66
pip:easydict	32	0.66
pip:libpng	32	0.66
pip:netcdf4	32	0.66
pip:py-cpuinfo	32	0.66
pip:pydub	32	0.66
pip:pygobject	32	0.66
pip:pymdown-extensions	32	0.66
pip:pyopengl	32	0.66

pip:readline	32	0.66
pip:aiofiles	31	0.63
pip:freetype	31	0.63
pip:hypothesis	31	0.63
pip:jpeg	31	0.63
pip:libffi	31	0.63
pip:libtiff	31	0.63
pip:mysqlclient	31	0.63
pip:onnx	31	0.63
pip:pybtex	31	0.63
pip:setproctitle	31	0.63
pip:wget	31	0.63
pip:blas	30	0.61
pip:colorlog	30	0.61
pip:marshmallow	30	0.61
pip:ncurses	30	0.61
pip:pycryptodome	30	0.61
pip:python-slugify	30	0.61
pip:rasterio	30	0.61
pip:resampy	30	0.61
pip:amqp	29	0.59
pip:annotated-types	29	0.59
pip:antlr4-python3-runtime	29	0.59
pip:audioread	29	0.59
pip:fastcache	29	0.59

pip:ghp-import	29	0.59
pip:icu	29	0.59
pip:monotonic	29	0.59
pip:msgpack-python	29	0.59
pip:pydantic-core	29	0.59
pip:pyqtgraph	29	0.59
pip:pytorch	29	0.59
pip:visdom	29	0.59
pip:awscli	28	0.57
pip:dash-renderer	28	0.57
pip:django-crispy-forms	28	0.57
pip:fasteners	28	0.57
pip:loguru	28	0.57
pip:numpy-base	28	0.57
pip:pygithub	28	0.57
pip:ray	28	0.57
pip:safetensors	28	0.57
pip:sentence-transformers	28	0.57
pip:simpleitk	28	0.57
pip:theano	28	0.57
pip:torchtext	28	0.57
pip:bkcharts	27	0.55
pip:coloredlogs	27	0.55
pip:constantly	27	0.55
pip:django-extensions	27	0.55

pip:execnet	27	0.55
pip:faker	27	0.55
pip:folium	27	0.55
pip:fvcore	27	0.55
pip:imgaug	27	0.55
pip:ipaddress	27	0.55
pip:jaxlib	27	0.55
pip:lmdb	27	0.55
pip:nbsphinx	27	0.55
pip:netifaces	27	0.55
pip:pymatgen	27	0.55
pip:reportlab	27	0.55
pip:sortedcollections	27	0.55
pip:spyder	27	0.55
pip:triton	27	0.55
pip:typeguard	27	0.55
pip:umap-learn	27	0.55
pip:zope-event	27	0.55
pip:django-filter	26	0.53
pip:feedparser	26	0.53
pip:flask-migrate	26	0.53
pip:fuzzywuzzy	26	0.53
pip:gradio	26	0.53
pip:mkdocs-material-extensions	26	0.53
pip:python-editor	26	0.53

pip:python-gflags	26	0.53
pip:python-levenshtein	26	0.53
pip:pywin32-ctypes	26	0.53
pip:pyxdg	26	0.53
pip:types-python-dateutil	26	0.53
pip:wsproto	26	0.53
pip:zstd	26	0.53
pip:anaconda-project	25	0.51
pip:billiard	25	0.51
pip:cmake	25	0.51
pip:django-debug-toolbar	25	0.51
pip:gmpy2	25	0.51
pip:humanize	25	0.51
pip:incremental	25	0.51
pip:latexcodec	25	0.51
pip:omxruntime	25	0.51
pip:pennylane	25	0.51
pip:pkg-resources	25	0.51
pip:pyinstaller	25	0.51
pip:sacrebleu	25	0.51
pip:semantic-version	25	0.51
pip:conda-build	24	0.49
pip:furo	24	0.49
pip:geojson	24	0.49
pip:google-cloud-bigquery	24	0.49

pip:google-crc32c	24	0.49
pip:imbalanced-learn	24	0.49
pip:jsonpatch	24	0.49
pip:libgcc-ng	24	0.49
pip:libstdcxx-ng	24	0.49
pip:libxml2	24	0.49
pip:lightgbm	24	0.49
pip:moviepy	24	0.49
pip:mujoco-py	24	0.49
pip:python-apt	24	0.49
pip:python-multipart	24	0.49
pip:qt	24	0.49
pip:scandir	24	0.49
pip:service-identity	24	0.49
pip:sphinx-design	24	0.49
pip:torch-scatter	24	0.49
pip:trimesh	24	0.49
pip:vine	24	0.49
pip:anaconda-navigator	23	0.47
pip:automat	23	0.47
pip:conda-verify	23	0.47
pip:dash-bootstrap-components	23	0.47
pip:dm-tree	23	0.47
pip:hdf5	23	0.47
pip:hyperlink	23	0.47

pip:kubernetes	23	0.47
pip:lazy-loader	23	0.47
pip:mergedeep	23	0.47
pip:mkdocstrings	23	0.47
pip:navigator-updater	23	0.47
pip:pep517	23	0.47
pip:progressbar2	23	0.47
pip:pyqt	23	0.47
pip:python-socketio	23	0.47
pip:pyyaml-env-tag	23	0.47
pip:qtawesome	23	0.47
pip:sqlalchemy-utils	23	0.47
pip:tensorflow-addons	23	0.47
pip:-libgcc-mutex	22	0.45
pip:argh	22	0.45
pip:ase	22	0.45
pip:bzip2	22	0.45
pip:emoji	22	0.45
pip:flask-testing	22	0.45
pip:gitdb2	22	0.45
pip:glfw	22	0.45
pip:imutils	22	0.45
pip:inflect	22	0.45
pip:inflection	22	0.45
pip:invoke	22	0.45

pip:munkres	22	0.45
pip:nbdime	22	0.45
pip:orjson	22	0.45
pip:pydicom	22	0.45
pip:pyproject-hooks	22	0.45
pip:spacy-legacy	22	0.45
pip:tensorflow-datasets	22	0.45
pip:tiktoken	22	0.45
pip:allennlp	21	0.43
pip:codecov	21	0.43
pip:email-validator	21	0.43
pip:expat	21	0.43
pip:glib	21	0.43
pip:hyperopt	21	0.43
pip:iopath	21	0.43
pip:libedit	21	0.43
pip:nvidia-cublas-cu11	21	0.43
pip:nvidia-cudnn-cu11	21	0.43
pip:pandoc	21	0.43
pip:passlib	21	0.43
pip:pathlib	21	0.43
pip:pathos	21	0.43
pip:pcre	21	0.43
pip:pypdf2	21	0.43
pip:qiskit	21	0.43

pip:shap	21	0.43
pip:tensorflow-tensorboard	21	0.43
pip:w3lib	21	0.43
pip:altgraph	20	0.41
pip:backports-zoneinfo	20	0.41
pip:base58	20	0.41
pip:configobj	20	0.41
pip:ffmpeg	20	0.41
pip:flask-socketio	20	0.41
pip:fontconfig	20	0.41
pip:httplib2	20	0.41
pip:iso8601	20	0.41
pip:jieba	20	0.41
pip:leveldb	20	0.41
pip:libarchive-c	20	0.41
pip:monty	20	0.41
pip:nvidia-cuda-nvrtc-cu11	20	0.41
pip:nvidia-cuda-runtime-cu11	20	0.41
pip:pint	20	0.41
pip:pyaml	20	0.41
pip:pyperclip	20	0.41
pip:python-engineio	20	0.41
pip:smmmap2	20	0.41
pip:textblob	20	0.41
pip:torch-sparse	20	0.41

pip:zeromq	20	0.41
pip:conda-package-handling	19	0.39
pip:cookiecutter	19	0.39
pip:cudatoolkit	19	0.39
pip:cvxpy	19	0.39
pip:ecdsa	19	0.39
pip:flask-caching	19	0.39
pip:flask-script	19	0.39
pip:gpustat	19	0.39
pip:humanfriendly	19	0.39
pip:idna-ssl	19	0.39
pip:intervaltree	19	0.39
pip:langchain	19	0.39
pip:libfortran-ng	19	0.39
pip:libsodium	19	0.39
pip:libuuid	19	0.39
pip:msgpack-numpy	19	0.39
pip:nvidia-ml-py3	19	0.39
pip:outcome	19	0.39
pip:pathy	19	0.39
pip:pooch	19	0.39
pip:pyreadline	19	0.39
pip:pytest-asyncio	19	0.39
pip:python-magic	19	0.39
pip:python-utils	19	0.39

pip:rouge-score	19	0.39
pip:tensorflow-hub	19	0.39
pip:trio	19	0.39
pip:trio-websocket	19	0.39
pip:types-requests	19	0.39
pip:virtualenv-clone	19	0.39
pip:wordcloud	19	0.39
pip:xlwings	19	0.39
pip:apscheduler	18	0.37
pip:backports-tempfile	18	0.37
pip:bandit	18	0.37
pip:binaryornot	18	0.37
pip:branca	18	0.37
pip:cachecontrol	18	0.37
pip:descartes	18	0.37
pip:diffusers	18	0.37
pip:editdistance	18	0.37
pip:ffmpeg-python	18	0.37
pip:flax	18	0.37
pip:googledrivedownloader	18	0.37
pip:jupyter-lsp	18	0.37
pip:keyrings-alt	18	0.37
pip:kornia	18	0.37
pip:libxcb	18	0.37
pip:lz4-c	18	0.37

pip:mlflow	18	0.37
pip:mysql-connector-python	18	0.37
pip:pip-tools	18	0.37
pip:progress	18	0.37
pip:proto-plus	18	0.37
pip:pydeck	18	0.37
pip:pytz-deprecation-shim	18	0.37
pip:semver	18	0.37
pip:shellingham	18	0.37
pip:typing-inspect	18	0.37
pip:uvloop	18	0.37
pip:win-inet-pton	18	0.37
pip:yaml	18	0.37
pip:async-lru	17	0.35
pip:atari-py	17	0.35
pip:autograd	17	0.35
pip:bumpversion	17	0.35
pip:configargparse	17	0.35
pip:dbus	17	0.35
pip:factory-boy	17	0.35
pip:flask-mail	17	0.35
pip:flask-pymongo	17	0.35
pip:gdal	17	0.35
pip:geographiclib	17	0.35
pip:jellyfish	17	0.35

pip:libiconv	17	0.35
pip:netaddr	17	0.35
pip:palettable	17	0.35
pip:pydotplus	17	0.35
pip:pyspark	17	0.35
pip:pytest-forked	17	0.35
pip:pytorch-pretrained-bert	17	0.35
pip:scrapy	17	0.35
pip:sphinxcontrib-jquery	17	0.35
pip:spyder-kernels	17	0.35
pip:texttable	17	0.35
pip:tf-estimator-nightly	17	0.35
pip:xdoctest	17	0.35
pip:accessible-pygments	16	0.33
pip:argcomplete	16	0.33
pip: blessings	16	0.33
pip:conllu	16	0.33
pip:diskcache	16	0.33
pip:django-storages	16	0.33
pip:faiss-cpu	16	0.33
pip:gst-plugins-base	16	0.33
pip:gstreamer	16	0.33
pip:mpld3	16	0.33
pip:path	16	0.33
pip:pox	16	0.33

pip:ppft	16	0.33
pip:proglog	16	0.33
pip:py4j	16	0.33
pip:pygraphviz	16	0.33
pip:pyomo	16	0.33
pip:pytest-doctestplus	16	0.33
pip:qdarkstyle	16	0.33
pip:rapidfuzz	16	0.33
pip:sphinx-book-theme	16	0.33
pip:uncertainties	16	0.33
pip:av	15	0.31
pip:bidict	15	0.31
pip:chex	15	0.31
pip:datashape	15	0.31
pip:gmp	15	0.31
pip:google	15	0.31
pip:grpcio-status	15	0.31
pip:httptools	15	0.31
pip:igraph	15	0.31
pip:jinja2-time	15	0.31
pip:langdetect	15	0.31
pip:ld-impl-linux-64	15	0.31
pip:livereload	15	0.31
pip:ml-collections	15	0.31
pip:optuna	15	0.31

pip:pybtex-docutils	15	0.31
pip:pycountry	15	0.31
pip:pypng	15	0.31
pip:pytest-remotedata	15	0.31
pip:queuelib	15	0.31
pip:torch-cluster	15	0.31
pip:torchfile	15	0.31
pip:transforms3d	15	0.31
pip:vtk	15	0.31
pip:waitress	15	0.31
pip:backoff	14	0.29
pip:cftime	14	0.29
pip:colorcet	14	0.29
pip:command-not-found	14	0.29
pip:crcmod	14	0.29
pip:dateparser	14	0.29
pip:django-enviro	14	0.29
pip:frozendict	14	0.29
pip:graphql-core	14	0.29
pip:h2	14	0.29
pip:hpack	14	0.29
pip:hyperframe	14	0.29
pip:langcodes	14	0.29
pip:matplotlib-base	14	0.29
pip:optax	14	0.29

pip:parsel	14	0.29
pip:pika	14	0.29
pip:pybedtools	14	0.29
pip:pyee	14	0.29
pip:pyqt5-qt5	14	0.29
pip:pytest-arraydiff	14	0.29
pip:pytest-astropy	14	0.29
pip:pytest-openfiles	14	0.29
pip:pytest-runner	14	0.29
pip:pytest-timeout	14	0.29
pip:python-debian	14	0.29
pip:python-louvain	14	0.29
pip:qiskit-aer	14	0.29
pip:qiskit-terra	14	0.29
pip:requests-unixsocket	14	0.29
pip:sequeval	14	0.29
pip:sphinx-external-toc	14	0.29
pip:sphinx-gallery	14	0.29
pip:stable-baselines3	14	0.29
pip:terminaltables	14	0.29
pip:thop	14	0.29
pip:webdriver-manager	14	0.29
pip:asyncctest	13	0.27
pip:autoflake	13	0.27
pip:bz2file	13	0.27

pip:cairo	13	0.27
pip:cairoffi	13	0.27
pip:catboost	13	0.27
pip:cerberus	13	0.27
pip:deepspeed	13	0.27
pip:deprecation	13	0.27
pip:diff-match-patch	13	0.27
pip:django-allauth	13	0.27
pip:elasticsearch-dsl	13	0.27
pip:eventlet	13	0.27
pip:fabric	13	0.27
pip:flaky	13	0.27
pip:geocoder	13	0.27
pip:grpcio-tools	13	0.27
pip:jsonlines	13	0.27
pip:jsonpath-ng	13	0.27
pip:kaggle	13	0.27
pip:line-profiler	13	0.27
pip:linkify-it-py	13	0.27
pip:ml-dtypes	13	0.27
pip:nh3	13	0.27
pip:parse	13	0.27
pip:parsimonious	13	0.27
pip:peewee	13	0.27
pip:pixman	13	0.27

pip:poyo	13	0.27
pip:pydispatcher	13	0.27
pip:pymndescent	13	0.27
pip:pytest-django	13	0.27
pip:python-ldap	13	0.27
pip:python3-openid	13	0.27
pip:s3fs	13	0.27
pip:sacred	13	0.27
pip:site	13	0.27
pip:sphinx-autobuild	13	0.27
pip:ssh-import-id	13	0.27
pip:stomp-py	13	0.27
pip:uc-micro-py	13	0.27
pip:watchfiles	13	0.27
pip:affine	12	0.25
pip:bump2version	12	0.25
pip:cachelib	12	0.25
pip:cherrypy	12	0.25
pip:datetime	12	0.25
pip:dendropy	12	0.25
pip:fairseq	12	0.25
pip:flake8-bugbear	12	0.25
pip:flask-bcrypt	12	0.25
pip:flask-session	12	0.25
pip:gputil	12	0.25

pip:imblearn	12	0.25
pip:mkddocs-autorefs	12	0.25
pip:mmev	12	0.25
pip:mne	12	0.25
pip:odfpy	12	0.25
pip:ogb	12	0.25
pip:osmnx	12	0.25
pip:param	12	0.25
pip:pefile	12	0.25
pip:pyaudio	12	0.25
pip:pycairo	12	0.25
pip:pycups	12	0.25
pip:pynvml	12	0.25
pip:pyquaternion	12	0.25
pip:pyquery	12	0.25
pip:pytorch-transformers	12	0.25
pip:retworkx	12	0.25
pip:scs	12	0.25
pip:spacy-loggers	12	0.25
pip:sphinx-autoapi	12	0.25
pip:sphinx-automodapi	12	0.25
pip:sphinxcontrib-napoleon	12	0.25
pip:sqlitedict	12	0.25
pip:stanza	12	0.25
pip:tensorflow-cpu	12	0.25

pip:tensorflow-metadata	12	0.25
pip:tweepy	12	0.25
pip:virtualenvwrapper	12	0.25
pip:wurlitzer	12	0.25
pip:zstandard	12	0.25
pip:anndata	11	0.23
pip:apex	11	0.23
pip:arviz	11	0.23
pip:bitsandbytes	11	0.23
pip:blaze	11	0.23
pip:cvxopt	11	0.23
pip:dataclasses-json	11	0.23
pip:django-compressor	11	0.23
pip:ecos	11	0.23
pip:en-core-web-sm	11	0.23
pip:fastai	11	0.23
pip:fastavro	11	0.23
pip:flair	11	0.23
pip:flake8-docstrings	11	0.23
pip:globus-sdk	11	0.23
pip:google-cloud-vision	11	0.23
pip:gspread	11	0.23
pip:holidays	11	0.23
pip:jaraco-functools	11	0.23
pip:jupyter-text	11	0.23

pip:kaleido	11	0.23
pip:krb5	11	0.23
pip:libssh2	11	0.23
pip:lz4	11	0.23
pip:macholib	11	0.23
pip:mysql-python	11	0.23
pip:myst-nb	11	0.23
pip:nameparser	11	0.23
pip:odo	11	0.23
pip:open3d	11	0.23
pip:opencv	11	0.23
pip:ordered-set	11	0.23
pip:osqp	11	0.23
pip:pdfminer-six	11	0.23
pip:pipenv	11	0.23
pip:pycryptodomex	11	0.23
pip:pyct	11	0.23
pip:pyfaidx	11	0.23
pip:pyfiglet	11	0.23
pip:pyhamcrest	11	0.23
pip:pylint-django	11	0.23
pip:pyshp	11	0.23
pip:pytorch-ignite	11	0.23
pip:requests-file	11	0.23
pip:requests-futures	11	0.23

pip:requests-ntlm	11	0.23
pip:scanpy	11	0.23
pip:sh	11	0.23
pip:sites	11	0.23
pip:skimage	11	0.23
pip:snuggs	11	0.23
pip:sphinx-notfound-page	11	0.23
pip:sphinx-tabs	11	0.23
pip:tf-slim	11	0.23
pip:tlxextract	11	0.23
pip:torchsummary	11	0.23
pip:tsfresh	11	0.23
pip:ufw	11	0.23
pip:yarg	11	0.23
pip:zmq	11	0.23
pip:aiocoap	10	0.20
pip:apache-beam	10	0.20
pip:asyncio	10	0.20
pip:autobahn	10	0.20
pip:boltons	10	0.20
pip:click-didyoumean	10	0.20
pip:contextily	10	0.20
pip:curl	10	0.20
pip:dask-core	10	0.20
pip:defer	10	0.20

pip:dgl	10	0.20
pip:django-bootstrap4	10	0.20
pip:dlib	10	0.20
pip:flask-admin	10	0.20
pip:flask-babel	10	0.20
pip:flask-bootstrap	10	0.20
pip:freezegun	10	0.20
pip:gin-config	10	0.20
pip:google-cloud-firestore	10	0.20
pip:graphene	10	0.20
pip:harfbuzz	10	0.20
pip:hiredis	10	0.20
pip:ipaddr	10	0.20
pip:ipympl	10	0.20
pip:jupyter-book	10	0.20
pip:language-selector	10	0.20
pip:lark	10	0.20
pip:libcurl	10	0.20
pip:lmfit	10	0.20
pip:marshmallow-sqlalchemy	10	0.20
pip:mkdodstrings-python	10	0.20
pip:mlxtend	10	0.20
pip:mongomock	10	0.20
pip:moto	10	0.20
pip:mp-api	10	0.20

pip:ndg-httpsclient	10	0.20
pip:nilearn	10	0.20
pip:nvidia-cuda-cupti-cu12	10	0.20
pip:nvidia-cuda-nvrtc-cu12	10	0.20
pip:nvidia-cuda-runtime-cu12	10	0.20
pip:nvidia-cudnn-cu12	10	0.20
pip:orderdict	10	0.20
pip:orderedmultidict	10	0.20
pip:paho-mqtt	10	0.20
pip:pandas-datareader	10	0.20
pip:parameterized	10	0.20
pip:peft	10	0.20
pip:pep8-naming	10	0.20
pip:pipreqs	10	0.20
pip:py-spy	10	0.20
pip:pympler	10	0.20
pip:pystache	10	0.20
pip:python-abi	10	0.20
pip:python-daemon	10	0.20
pip:python-docx	10	0.20
pip:python-mimeparse	10	0.20
pip:rdkit	10	0.20
pip:requests-mock	10	0.20
pip:restructuredtext-lint	10	0.20
pip:rfc3987	10	0.20

pip:rpy2	10	0.20
pip:scikit-optimize	10	0.20
pip:slackclient	10	0.20
pip:slicer	10	0.20
pip:sphinx-click	10	0.20
pip:sphinxcontrib- moderncmakedomain	10	0.20
pip:sphinxcontrib-svg2pdfconverter	10	0.20
pip:system-service	10	0.20
pip:tb-nightly	10	0.20
pip:tbb	10	0.20
pip:testing-postgresql	10	0.20
pip:toposort	10	0.20
pip:torch-spline-conv	10	0.20
pip:twilio	10	0.20
pip:types-setuptools	10	0.20
pip:unattended-upgrades	10	0.20
pip:uwsgi	10	0.20
pip:visitor	10	0.20
pip:win-unicode-console	10	0.20
pip:xformers	10	0.20
pip:xkit	10	0.20
pip:yaspin	10	0.20
pip:yfinance	10	0.20
pip:zc-lockfile	10	0.20

pub:collection	31	79.49
pub:meta	31	79.49
pub:matcher	30	76.92
pub:path	30	76.92
pub:string_scanner	30	76.92
pub:async	29	74.36
pub:boolean_selector	29	74.36
pub:source_span	29	74.36
pub:stack_trace	29	74.36
pub:stream_channel	29	74.36
pub:term_glyph	29	74.36
pub:test_api	29	74.36
pub:typed_data	28	71.79
pub:vector_math	26	66.67
pub:cupertino_icons	25	64.10
pub:crypto	22	56.41
pub:http_parser	22	56.41
pub:fake_async	21	53.85
pub:args	20	51.28
pub:clock	20	51.28
pub:characters	19	48.72
pub:lints	19	48.72
pub:charcode	18	46.15
pub:http	18	46.15
pub:js	18	46.15

pub:provider	18	46.15
pub:convert	17	43.59
pub:flutter_lints	17	43.59
pub:intl	17	43.59
pub:file	16	41.03
pub:logging	14	35.90
pub:material_color_utilities	14	35.90
pub:nested	14	35.90
pub:pedantic	14	35.90
pub:plugin_platform_interface	14	35.90
pub:test	14	35.90
pub:json_annotation	13	33.33
pub:quiver	13	33.33
pub:web_socket_channel	13	33.33
pub:yaml	13	33.33
pub:glob	12	30.77
pub:petitparser	12	30.77
pub:platform	12	30.77
pub:stream_transform	12	30.77
pub:xml	12	30.77
pub:build_runner	11	28.21
pub:io	11	28.21
pub:package_config	11	28.21
pub:pool	11	28.21
pub:pub_semver	11	28.21

pub:_fe_analyzer_shared	10	25.64
pub:analyzer	10	25.64
pub:ffi	10	25.64
pub:graphs	10	25.64
pub:http_multi_server	10	25.64
pub:mime	10	25.64
pub:path_provider	10	25.64
pub:shelf	10	25.64
pub:shelf_web_socket	10	25.64
pub:watcher	10	25.64
rubygems:rake	251	48.08
rubygems:i18n	213	40.80
rubygems:nokogiri	205	39.27
rubygems:tzinfo	204	39.08
rubygems:activesupport	196	37.55
rubygems:ffi	195	37.36
rubygems:minitest	194	37.16
rubygems:addressable	191	36.59
rubygems:concurrent-ruby	175	33.52
rubygems:public_suffix	164	31.42
rubygems:sass	162	31.03
rubygems:rb-fsevent	159	30.46
rubygems:rb-inotify	159	30.46
rubygems:execjs	157	30.08
rubygems:listen	148	28.35

rubygems:faraday	145	27.78
rubygems:jekyll	141	27.01
rubygems:rubyzip	136	26.05
rubygems:thread_safe	135	25.86
rubygems:json	134	25.67
rubygems:builder	133	25.48
rubygems:rack	133	25.48
rubygems:unicode-display_width	129	24.71
rubygems:coffee-script	128	24.52
rubygems:coffee-script-source	128	24.52
rubygems:multi_json	126	24.14
rubygems:thor	126	24.14
rubygems:mini_portile2	124	23.75
rubygems:rack-test	122	23.37
rubygems:rails	122	23.37
rubygems:activemodel	120	22.99
rubygems:activerecord	119	22.80
rubygems:github-pages	119	22.80
rubygems:tilt	119	22.80
rubygems:actionpack	116	22.22
rubygems:mail	116	22.22
rubygems:railties	114	21.84
rubygems:rspec	114	21.84
rubygems:actionmailer	112	21.46
rubygems:multipart-post	109	20.88

rubygems:sprockets	108	20.69
rubygems:rexml	105	20.11
rubygems:diff-lcs	104	19.92
rubygems:rspec-mocks	104	19.92
rubygems:safe_yaml	104	19.92
rubygems:rspec-core	103	19.73
rubygems:rspec-expectations	102	19.54
rubygems:jekyll-feed	99	18.97
rubygems:kramdown	97	18.58
rubygems:actionview	95	18.20
rubygems:rouge	95	18.20
rubygems:sass-listen	95	18.20
rubygems:sass-rails	94	18.01
rubygems:sprockets-rails	94	18.01
rubygems:jquery-rails	93	17.82
rubygems:webrick	93	17.82
rubygems:loofah	92	17.62
rubygems:mime-types	92	17.62
rubygems:rails-dom-testing	91	17.43
rubygems:rails-html-sanitizer	91	17.43
rubygems:uglifier	90	17.24
rubygems:liquid	89	17.05
rubygems:method_source	89	17.05
rubygems:activejob	88	16.86
rubygems:byebug	88	16.86

rubygems:colorator	88	16.86
rubygems:globalid	88	16.86
rubygems:rspec-support	88	16.86
rubygems:jekyll-sass-converter	87	16.67
rubygems:jekyll-watch	87	16.67
rubygems:mercenary	87	16.67
rubygems:arel	86	16.48
rubygems:eventmachine	86	16.48
rubygems:jekyll-sitemap	84	16.09
rubygems:terminal-table	84	16.09
rubygems:http_parser.rb	83	15.90
rubygems:rspec-rails	83	15.90
rubygems:forwardable-extended	82	15.71
rubygems:pathutil	82	15.71
rubygems:rubocop	82	15.71
rubygems:jbuilder	81	15.52
rubygems:racc	81	15.52
rubygems:capybara	80	15.33
rubygems:parallel	80	15.33
rubygems:simplecov	79	15.13
rubygems:jekyll-paginate	77	14.75
rubygems:coffee-rails	76	14.56
rubygems:mini_mime	76	14.56
rubygems:unf	76	14.56
rubygems:unf_ext	76	14.56

rubygems:xpath	75	14.37
rubygems:ruby-progressbar	74	14.18
rubygems:typhoeus	74	14.18
rubygems:em-websocket	73	13.98
rubygems:ethon	72	13.79
rubygems:pg	72	13.79
rubygems:websocket-driver	72	13.79
rubygems:websocket-extensions	72	13.79
rubygems:zeitwerk	72	13.79
rubygems:bcrypt	71	13.60
rubygems:jekyll-seo-tag	71	13.60
rubygems:coderay	69	13.22
rubygems:jekyll-redirect-from	69	13.22
rubygems:minima	69	13.22
rubygems:crass	68	13.03
rubygems:nio4r	68	13.03
rubygems:rainbow	68	13.03
rubygems:sqlite3	68	13.03
rubygems:erubis	67	12.84
rubygems:ruby2_keywords	67	12.84
rubygems:octokit	66	12.64
rubygems:ast	65	12.45
rubygems:devise	65	12.45
rubygems:jemoji	65	12.45
rubygems:puma	65	12.45

rubygems:sawyer	65	12.45
rubygems:warden	65	12.45
rubygems:actioncable	64	12.26
rubygems:orm_adapter	64	12.26
rubygems:spring	64	12.26
rubygems:web-console	64	12.26
rubygems:kramdown-parser-gfm	63	12.07
rubygems:parser	61	11.69
rubygems:pry	61	11.69
rubygems:faraday-net_http	60	11.49
rubygems:jeekyll-gist	59	11.30
rubygems:jeekyll-mentions	59	11.30
rubygems:responders	59	11.30
rubygems:autoprefixer-rails	58	11.11
rubygems:erubi	58	11.11
rubygems:gemoji	58	11.11
rubygems:jeekyll-remote-theme	58	11.11
rubygems:simplecov-html	58	11.11
rubygems:turbolinks	58	11.11
rubygems:jeekyll-coffeescript	57	10.92
rubygems:jeekyll-github-metadata	57	10.92
rubygems:mime-types-data	57	10.92
rubygems:docile	56	10.73
rubygems:html-pipeline	56	10.73
rubygems:bundler	55	10.54

rubygems:github-pages-health-check	54	10.34
rubygems:hashie	53	10.15
rubygems:jekyll-relative-links	53	10.15
rubygems:kaminari	53	10.15
rubygems:rdoc	53	10.15
rubygems:sassc	53	10.15
rubygems:selenium-webdriver	53	10.15
rubygems:bootstrap-sass	52	9.96
rubygems:jekyll-theme-cayman	52	9.96
rubygems:jekyll-avatar	51	9.77
rubygems:jekyll-default-layout	51	9.77
rubygems:jekyll-optional-front-matter	51	9.77
rubygems:jekyll-readme-index	51	9.77
rubygems:jekyll-swiss	51	9.77
rubygems:jekyll-theme-minimal	51	9.77
rubygems:jekyll-titles-from-headings	51	9.77
rubygems:tzinfo-data	51	9.77
rubygems:jekyll-theme-architect	50	9.58
rubygems:jekyll-theme-dinky	50	9.58
rubygems:jekyll-theme-hacker	50	9.58
rubygems:jekyll-theme-leap-day	50	9.58
rubygems:jekyll-theme-merlot	50	9.58

rubygems:jekyll-theme-midnight	50	9.58
rubygems:jekyll-theme-modernist	50	9.58
rubygems:jekyll-theme-primer	50	9.58
rubygems:jekyll-theme-slate	50	9.58
rubygems:jekyll-theme-tactile	50	9.58
rubygems:jekyll-theme-time-machine	50	9.58
rubygems:rack-protection	50	9.58
rubygems:net-ssh	49	9.39
rubygems:regexp_parser	49	9.39
rubygems:webmock	48	9.20
rubygems:commonmarker	47	9.00
rubygems:rsolr	47	9.00
rubygems:dnsruby	46	8.81
rubygems:jekyll-commonmark	46	8.81
rubygems:jekyll-commonmark-ghpages	46	8.81
rubygems:net-scp	46	8.81
rubygems:bindindex	45	8.62
rubygems:domain_name	45	8.62
rubygems:jwt	45	8.62
rubygems:redis	45	8.62
rubygems:childprocess	44	8.43
rubygems:database_cleaner	44	8.43
rubygems:msgpack	44	8.43

rubygems:multi_xml	44	8.43
rubygems:turbolinks-source	44	8.43
rubygems:connection_pool	43	8.24
rubygems:http-cookie	43	8.24
rubygems:activestorage	42	8.05
rubygems:crack	42	8.05
rubygems:marcel	42	8.05
rubygems:slop	41	7.85
rubygems:blacklight	40	7.66
rubygems:deprecation	40	7.66
rubygems:factory_bot	40	7.66
rubygems:hashdiff	40	7.66
rubygems:capistrano	39	7.47
rubygems:haml	39	7.47
rubygems:httpclient	39	7.47
rubygems:rubocop-rspec	39	7.47
rubygems:factory_bot_rails	38	7.28
rubygems:rest-client	38	7.28
rubygems:daemons	37	7.09
rubygems:simpleidn	37	7.09
rubygems:cancancan	36	6.90
rubygems:dotenv	36	6.90
rubygems:jeekyll-include-cache	36	6.90
rubygems:matrix	35	6.70
rubygems:sinatra	35	6.70

rubygems:kaminari-actionview	34	6.51
rubygems:kaminari-activerecord	34	6.51
rubygems:kaminari-core	34	6.51
rubygems:ruby_dep	34	6.51
rubygems:sshkit	34	6.51
rubygems:aws-sdk-core	33	6.32
rubygems:httparty	33	6.32
rubygems:rubocop-ast	33	6.32
rubygems:spring-watcher-listen	33	6.32
rubygems:therubyracer	33	6.32
rubygems:jmespath	32	6.13
rubygems:libv8	32	6.13
rubygems:bootsnap	31	5.94
rubygems:omniauth	31	5.94
rubygems:debug_inspector	30	5.75
rubygems:htmlentities	30	5.75
rubygems:mini_magick	30	5.75
rubygems:mysql2	30	5.75
rubygems:net-http-persistent	30	5.75
rubygems:net-smtp	30	5.75
rubygems:oauth2	30	5.75
rubygems:rdf	30	5.75
rubygems:rubocop-performance	30	5.75
rubygems:sdoc	30	5.75
rubygems:websocket	30	5.75

rubygems:aws-sigv4	29	5.56
rubygems:capistrano-bundler	29	5.56
rubygems:net-imap	29	5.56
rubygems:net-pop	29	5.56
rubygems:net-protocol	29	5.56
rubygems:netrc	29	5.56
rubygems:timeout	29	5.56
rubygems:actionmailbox	28	5.36
rubygems:actiontext	28	5.36
rubygems:aws-partitions	28	5.36
rubygems:aws-sdk-s3	28	5.36
rubygems:capistrano-rails	28	5.36
rubygems:launchy	28	5.36
rubygems:rails-controller-testing	28	5.36
rubygems:redis-namespace	28	5.36
rubygems:ref	28	5.36
rubygems:activerecord-import	27	5.17
rubygems:aws-sdk-kms	27	5.17
rubygems:date	27	5.17
rubygems:devise-guests	27	5.17
rubygems:retriable	27	5.17
rubygems:sassc-rails	27	5.17
rubygems:simplecov_json_formatter	27	5.17
rubygems:uber	27	5.17
rubygems:active-fedora	26	4.98

rubygems:airbrussh	26	4.98
rubygems:binding_of_caller	26	4.98
rubygems:equivalent-xml	26	4.98
rubygems:factory_girl	26	4.98
rubygems:faker	26	4.98
rubygems:faraday_middleware	26	4.98
rubygems:font-awesome-rails	26	4.98
rubygems:hike	26	4.98
rubygems:rubocop-rails	26	4.98
rubygems:ruby-enum	26	4.98
rubygems:wdm	26	4.98
rubygems:highline	25	4.79
rubygems:psych	25	4.79
rubygems:rails-deprecated_sanitizer	25	4.79
rubygems:aws-eventstream	24	4.60
rubygems:dotenv-rails	24	4.60
rubygems:jquery-ui-rails	24	4.60
rubygems:openstudio-standards	24	4.60
rubygems:sidekiq	24	4.60
rubygems:temple	24	4.60
rubygems:twitter-typeahead-rails	24	4.60
rubygems:webdrivers	24	4.60
rubygems:xml-simple	24	4.60
rubygems:coveralls	23	4.41
rubygems:ebnf	23	4.41

rubygems:faraday-excon	23	4.41
rubygems:faraday- net_http_persistent	23	4.41
rubygems:json-ld	23	4.41
rubygems:link_header	23	4.41
rubygems:rdف-turtle	23	4.41
rubygems:simple_form	23	4.41
rubygems:sxp	23	4.41
rubygems:declarative	22	4.21
rubygems:hydra-core	22	4.21
rubygems:ice_nine	22	4.21
rubygems:rdف-isomorphic	22	4.21
rubygems:rdف-vocab	22	4.21
rubygems:representable	22	4.21
rubygems:vcr	22	4.21
rubygems:yard	22	4.21
rubygems:active-triples	21	4.02
rubygems:chronic	21	4.02
rubygems:factory_girl_rails	21	4.02
rubygems:faraday-em_http	21	4.02
rubygems:faraday-em_synchrony	21	4.02
rubygems:faraday-httpclient	21	4.02
rubygems:faraday-patron	21	4.02
rubygems:faraday-retry	21	4.02
rubygems:json-schema	21	4.02

rubygems:openstudio-extension	21	4.02
rubygems:qa	21	4.02
rubygems:rdf-rdfxml	21	4.02
rubygems:rdf-xsd	21	4.02
rubygems:shoulda-matchers	21	4.02
rubygems:solrizer	21	4.02
rubygems:term-ansicolor	21	4.02
rubygems:awesome_print	20	3.83
rubygems:babel-source	20	3.83
rubygems:babel-transpiler	20	3.83
rubygems:carrierwave	20	3.83
rubygems:dry-core	20	3.83
rubygems:faraday-rack	20	3.83
rubygems:googleauth	20	3.83
rubygems:html-proofer	20	3.83
rubygems:http	20	3.83
rubygems:http_logger	20	3.83
rubygems:hydra-access-controls	20	3.83
rubygems:hydra-head	20	3.83
rubygems:mimemagic	20	3.83
rubygems:pry-byebug	20	3.83
rubygems:rails_autolink	20	3.83
rubygems:request_store	20	3.83
rubygems:signet	20	3.83
rubygems:solr_wrapper	20	3.83

rubygems:sparql-client	20	3.83
rubygems:will_paginate	20	3.83
rubygems:base64	19	3.64
rubygems:bootstrap	19	3.64
rubygems:bootstrap_form	19	3.64
rubygems:dry-configurable	19	3.64
rubygems:excon	19	3.64
rubygems:guard	19	3.64
rubygems:linkeddata	19	3.64
rubygems:memoist	19	3.64
rubygems:mustermann	19	3.64
rubygems:os	19	3.64
rubygems:parslet	19	3.64
rubygems:polyglot	19	3.64
rubygems:popper_js	19	3.64
rubygems:powerpack	19	3.64
rubygems:tins	19	3.64
rubygems:treetop	19	3.64
rubygems:better_errors	18	3.45
rubygems:colorize	18	3.45
rubygems:dry-logic	18	3.45
rubygems:dry-types	18	3.45
rubygems:faraday-multipart	18	3.45
rubygems:http-form_data	18	3.45
rubygems:hydra-editor	18	3.45

rubygems:json_pure	18	3.45
rubygems:ldap	18	3.45
rubygems:net-ldap	18	3.45
rubygems:rdf-aggregate-repo	18	3.45
rubygems:rdf-json	18	3.45
rubygems:rdf-microdata	18	3.45
rubygems:rdf-n3	18	3.45
rubygems:rdf-rdfa	18	3.45
rubygems:rdf-trig	18	3.45
rubygems:rdf-trix	18	3.45
rubygems:sparql	18	3.45
rubygems:yell	18	3.45
rubygems:activeresource	17	3.26
rubygems:ansi	17	3.26
rubygems:bcp47	17	3.26
rubygems:ffi-compiler	17	3.26
rubygems:logger	17	3.26
rubygems:minitest-reporters	17	3.26
rubygems:omniauth-oauth2	17	3.26
rubygems:sprockets-es6	17	3.26
rubygems:stomp	17	3.26
rubygems:bigdecimal	16	3.07
rubygems:blacklight- access_controls	16	3.07
rubygems:breadcrumbs_on_rails	16	3.07

rubygems:capistrano-passenger	16	3.07
rubygems:dry-container	16	3.07
rubygems:hamster	16	3.07
rubygems:noid	16	3.07
rubygems:om	16	3.07
rubygems:pry-rails	16	3.07
rubygems:rdf-reasoner	16	3.07
rubygems:rdf-tabular	16	3.07
rubygems:resque	16	3.07
rubygems:blacklight-gallery	15	2.87
rubygems:chunky_png	15	2.87
rubygems:dry-equalizer	15	2.87
rubygems:formatador	15	2.87
rubygems:hydra-	15	2.87
file_characterization		
rubygems:hydra-role-management	15	2.87
rubygems:llhttp-ffi	15	2.87
rubygems:mono_logger	15	2.87
rubygems:openseadragon	15	2.87
rubygems:rack-cache	15	2.87
rubygems:redcarpet	15	2.87
rubygems:backports	14	2.68
rubygems:clipboard-rails	14	2.68
rubygems:colored	14	2.68
rubygems:cucumber	14	2.68

rubygems:database_cleaner- active_record	14	2.68
rubygems:database_cleaner-core	14	2.68
rubygems:dry-validation	14	2.68
rubygems:hydra-derivatives	14	2.68
rubygems:hydra-pcdm	14	2.68
rubygems:hydra-works	14	2.68
rubygems:jaro_winkler	14	2.68
rubygems:marc	14	2.68
rubygems:mini_portile	14	2.68
rubygems:minitar	14	2.68
rubygems:net-sftp	14	2.68
rubygems:posix-spawn	14	2.68
rubygems:rack-proxy	14	2.68
rubygems:rack-ssl	14	2.68
rubygems:rubocop- checkstyle_formatter	14	2.68
rubygems:vegas	14	2.68
rubygems:view_component	14	2.68
rubygems:blacklight_range_limit	13	2.49
rubygems:columnize	13	2.49
rubygems:declarative-option	13	2.49
rubygems:dry-struct	13	2.49
rubygems:faraday-encoding	13	2.49
rubygems:fcrepo_wrapper	13	2.49

rubygems:git	13	2.49
rubygems:language_server-protocol	13	2.49
rubygems:lumberjack	13	2.49
rubygems:mocha	13	2.49
rubygems:mutex_m	13	2.49
rubygems:poltergeist	13	2.49
rubygems:redlock	13	2.49
rubygems:rspec-its	13	2.49
rubygems:rubocop-capbara	13	2.49
rubygems:spring-commands-rspec	13	2.49
rubygems:timecop	13	2.49
rubygems:almond-rails	12	2.30
rubygems:awesome_nested_set	12	2.30
rubygems:brakeman	12	2.30
rubygems:bson	12	2.30
rubygems:cliver	12	2.30
rubygems:compass	12	2.30
rubygems:flot-rails	12	2.30
rubygems:geocoder	12	2.30
rubygems:guard-compat	12	2.30
rubygems:guard-rspec	12	2.30
rubygems:haml-rails	12	2.30
rubygems:journey	12	2.30
rubygems:nenv	12	2.30
rubygems:nom-xml	12	2.30

rubygems:notiffany	12	2.30
rubygems:oauth	12	2.30
rubygems:omniauth- rails_csrf_protection	12	2.30
rubygems:rack-mini-profiler	12	2.30
rubygems:rspec_junit_formatter	12	2.30
rubygems:select2-rails	12	2.30
rubygems:shellany	12	2.30
rubygems:simplecov-lcov	12	2.30
rubygems:active_encode	11	2.11
rubygems:bixby	11	2.11
rubygems:browse-everything	11	2.11
rubygems:config	11	2.11
rubygems:deep_merge	11	2.11
rubygems:dry-inflector	11	2.11
rubygems:font-awesome-sass	11	2.11
rubygems:google-apis-core	11	2.11
rubygems:google_drive	11	2.11
rubygems:hiredis	11	2.11
rubygems:jekyll-scholar	11	2.11
rubygems:jettywrapper	11	2.11
rubygems:jquery-datatables-rails	11	2.11
rubygems:ld-patch	11	2.11
rubygems:mini_racer	11	2.11
rubygems:openstudio_measure_tester	11	2.11

rubygems:ostrucht	11	2.11
rubygems:rack-cors	11	2.11
rubygems:roo	11	2.11
rubygems:rspec-activemodel-mocks	11	2.11
rubygems:ruby-box	11	2.11
rubygems:scrub_rb	11	2.11
rubygems:thin	11	2.11
rubygems:uri	11	2.11
rubygems:webpacker	11	2.11
rubygems:whenever	11	2.11
rubygems:active_model_serializers	10	1.92
rubygems:aws-sdk	10	1.92
rubygems:capistrano-rvm	10	1.92
rubygems:colored2	10	1.92
rubygems:delayed_job	10	1.92
rubygems:descendants_tracker	10	1.92
rubygems:drb	10	1.92
rubygems:fugit	10	1.92
rubygems:google-api-client	10	1.92
rubygems:honeybadger	10	1.92
rubygems:http-accept	10	1.92
rubygems:json-ld-preloaded	10	1.92
rubygems:kaminari_route_prefix	10	1.92
rubygems:legato	10	1.92
rubygems:mongo	10	1.92

rubygems:net-http	10	1.92
rubygems:oj	10	1.92
rubygems:okcomputer	10	1.92
rubygems:omniauth-shibboleth	10	1.92
rubygems:rdf-normalize	10	1.92
rubygems:rubocop-factory_bot	10	1.92
rubygems:semantic_range	10	1.92
rubygems:shex	10	1.92
rubygems:stringio	10	1.92
rubygems:virtus	10	1.92
rust:serde	206	58.86
rust:libc	199	56.86
rust:quote	187	53.43
rust:rand	182	52.00
rust:syn	182	52.00
rust:proc-macro2	180	51.43
rust:lazy_static	174	49.71
rust:cfg-if	172	49.14
rust:bitflags	170	48.57
rust:log	165	47.14
rust:serde_json	156	44.57
rust:regex	155	44.29
rust:winapi	152	43.43
rust:winapi-i686-pc-windows-gnu	146	41.71

rust:winapi-x86_64-pc-windows-gnu	146	41.71
rust:autocfg	145	41.43
rust:memchr	145	41.43
rust:serde_derive	142	40.57
rust:clap	140	40.00
rust:wasi	133	38.00
rust:getrandom	132	37.71
rust:num-traits	132	37.71
rust:once_cell	131	37.43
rust:byteorder	129	36.86
rust:regex-syntax	127	36.29
rust:cc	126	36.00
rust:hermit-abi	123	35.14
rust:rand_core	123	35.14
rust:thiserror	122	34.86
rust:itoa	120	34.29
rust:aho-corasick	116	33.14
rust:ryu	116	33.14
rust:unicode-ident	115	32.86
rust:rand_chacha	113	32.29
rust:either	112	32.00
rust:version_check	112	32.00
rust:num_cpus	110	31.43
rust:ppv-lite86	108	30.86

rust:strsim	108	30.86
rust:redox_syscall	107	30.57
rust:scopeguard	107	30.57
rust:wasm-bindgen	106	30.29
rust:crossbeam-utils	104	29.71
rust:itertools	104	29.71
rust:smallvec	103	29.43
rust:hashbrown	102	29.14
rust:windows_aarch64_msvc	101	28.86
rust:windows_i686_gnu	101	28.86
rust:windows_i686_msvc	101	28.86
rust:windows_x86_64_gnu	101	28.86
rust:windows_x86_64_msvc	101	28.86
rust:winapi-util	100	28.57
rust:tokio	99	28.29
rust:windows-sys	98	28.00
rust:js-sys	97	27.71
rust:anyhow	96	27.43
rust:chrono	96	27.43
rust:heck	96	27.43
rust:indexmap	95	27.14
rust:num-integer	95	27.14
rust:bumpalo	94	26.86
rust:wasm-bindgen-backend	93	26.57
rust:wasm-bindgen-macro	93	26.57

rust:wasm-bindgen-macro-support	93	26.57
rust:wasm-bindgen-shared	93	26.57
rust:bytes	90	25.71
rust:miniz_oxide	90	25.71
rust:rayon	90	25.71
rust:thiserror-impl	90	25.71
rust:windows_aarch64_gnullvm	89	25.43
rust:windows_x86_64_gnullvm	89	25.43
rust:parking_lot	88	25.14
rust:tempfile	88	25.14
rust:unicode-width	88	25.14
rust:memoffset	87	24.86
rust:lock_api	86	24.57
rust:typenum	86	24.57
rust:windows-targets	86	24.57
rust:crossbeam-deque	85	24.29
rust:crossbeam-epoch	85	24.29
rust:adler	84	24.00
rust:semver	84	24.00
rust:unicode-xid	84	24.00
rust:base64	83	23.71
rust:fnv	82	23.43
rust:pkg-config	81	23.14
rust:futures	80	22.86
rust:web-sys	80	22.86

rust:futures-core	79	22.57
rust:time	79	22.57
rust:tracing	79	22.57
rust:futures-util	78	22.29
rust:parking_lot_core	78	22.29
rust:pin-project-lite	78	22.29
rust:slab	78	22.29
rust:flate2	76	21.71
rust:futures-sink	76	21.71
rust:generic-array	76	21.71
rust:atty	75	21.43
rust:digest	75	21.43
rust:env_logger	75	21.43
rust:regex-automata	75	21.43
rust:crc32fast	74	21.14
rust:errno	73	20.86
rust:mio	73	20.86
rust:termcolor	72	20.57
rust:crossbeam-channel	71	20.29
rust:pin-utils	71	20.29
rust:rayon-core	71	20.29
rust:fastrand	70	20.00
rust:futures-task	70	20.00
rust:percent-encoding	70	20.00
rust:rustc_version	70	20.00

rust:core-foundation-sys	69	19.71
rust:linux-raw-sys	69	19.71
rust:rustix	69	19.71
rust:backtrace	68	19.43
rust:clap_lex	68	19.43
rust:futures-channel	68	19.43
rust:spin	68	19.43
rust:block-buffer	67	19.14
rust:socket2	67	19.14
rust:tracing-core	67	19.14
rust:rustc-demangle	66	18.86
rust:futures-io	65	18.57
rust:url	65	18.57
rust:walkdir	65	18.57
rust:nom	64	18.29
rust:paste	64	18.29
rust:arrayvec	63	18.00
rust:thread_local	63	18.00
rust:http	62	17.71
rust:textwrap	62	17.71
rust:tokio-util	62	17.71
rust:glob	61	17.43
rust:bincode	60	17.14
rust:instant	60	17.14
rust:tinyvec	60	17.14

rust:tinyvec_macros	60	17.14
rust:unicode-bidi	60	17.14
rust:unicode-normalization	60	17.14
rust:futures-macro	59	16.86
rust:hyper	59	16.86
rust:idna	59	16.86
rust:same-file	59	16.86
rust:hex	58	16.57
rust:httparse	58	16.57
rust:sha2	58	16.57
rust:addr2line	57	16.29
rust:ahash	57	16.29
rust:object	57	16.29
rust:gimli	56	16.00
rust:uuid	56	16.00
rust:form_urlencoded	55	15.71
rust:num-complex	55	15.71
rust:anstyle	54	15.43
rust:clap_builder	54	15.43
rust:clap_derive	54	15.43
rust:crypto-common	54	15.43
rust:futures-executor	54	15.43
rust:humantime	54	15.43
rust:unicode-segmentation	54	15.43
rust:async-trait	53	15.14

rust:cpufeatures	53	15.14
rust:http-body	53	15.14
rust:num-rational	53	15.14
rust:signal-hook-registry	53	15.14
rust:vcpkg	53	15.14
rust:toml	52	14.86
rust:tower-service	52	14.86
rust:ansi_term	51	14.57
rust:proc-macro-error	51	14.57
rust:try-lock	51	14.57
rust:utf8parse	51	14.57
rust:want	51	14.57
rust:core-foundation	50	14.29
rust:h2	50	14.29
rust:httpdate	50	14.29
rust:vec_map	50	14.29
rust:proc-macro-error-attr	49	14.00
rust:rustversion	49	14.00
rust:tracing-attributes	49	14.00
rust:tracing-subscriber	49	14.00
rust:wasm-bindgen-futures	49	14.00
rust:anstream	48	13.71
rust:anstyle-parse	48	13.71
rust:anstyle-wincon	48	13.71
rust:num-bigint	48	13.71

rust:static_assertions	48	13.71
rust:anstyle-query	47	13.43
rust:colorchoice	47	13.43
rust:reqwest	47	13.43
rust:subtle	47	13.43
rust:mime	46	13.14
rust:nix	46	13.14
rust:pin-project	46	13.14
rust:redox_users	46	13.14
rust:zeroize	46	13.14
rust:android_system_properties	45	12.86
rust:criterion	45	12.86
rust:equivalent	45	12.86
rust:iana-time-zone	45	12.86
rust:tokio-macros	45	12.86
rust:csv	44	12.57
rust:foreign-types	44	12.57
rust:foreign-types-shared	44	12.57
rust:iana-time-zone-haiku	44	12.57
rust:libm	44	12.57
rust:pin-project-internal	44	12.57
rust:serde_urlencoded	43	12.29
rust:bytemuck	42	12.00
rust:libloading	42	12.00
rust:openssl-probe	42	12.00

rust:rand_hc	42	12.00
rust:schannel	42	12.00
rust:is-terminal	41	11.71
rust:security-framework	41	11.71
rust:security-framework-sys	41	11.71
rust:bindgen	40	11.43
rust:num-iter	40	11.43
rust:ring	40	11.43
rust:rustc-hash	40	11.43
rust:semver-parser	40	11.43
rust:untrusted	40	11.43
rust:crunchy	39	11.14
rust:encoding_rs	39	11.14
rust:ipnet	39	11.14
rust:openssl-sys	39	11.14
rust:android-tzdata	38	10.86
rust:bstr	38	10.86
rust:half	38	10.86
rust:minimal-lexical	38	10.86
rust:num	38	10.86
rust:strum	38	10.86
rust:strum_macros	38	10.86
rust:windows	38	10.86
rust:openssl	37	10.57
rust:petgraph	37	10.57

rust:sharded-slab	37	10.57
rust:time-macros	37	10.57
rust:winreg	37	10.57
rust:approx	36	10.29
rust:fixedbitset	36	10.29
rust:opaque-debug	36	10.29
rust:time-core	36	10.29
rust:derive_more	35	10.00
rust:hmac	35	10.00
rust:quick-error	35	10.00
rust:valuable	35	10.00
rust:csv-core	34	9.71
rust:jobserver	34	9.71
rust:rustls	34	9.71
rust:zerocopy	34	9.71
rust:native-tls	33	9.43
rust:stable_deref_trait	33	9.43
rust:which	33	9.43
rust:zerocopy-derive	33	9.43
rust:darling	32	9.14
rust:siphasher	32	9.14
rust:synstructure	32	9.14
rust:tracing-log	32	9.14
rust:crossbeam-queue	31	8.86
rust:io-lifetimes	31	8.86

rust:ndarray	31	8.86
rust:openssl-macros	31	8.86
rust:proc-macro-crate	31	8.86
rust:toml_edit	31	8.86
rust:convert_case	30	8.57
rust:errno-dragonfly	30	8.57
rust:image	30	8.57
rust:matrixmultiply	30	8.57
rust:memmap2	30	8.57
rust:rustls-pemfile	30	8.57
rust:tokio-stream	30	8.57
rust:toml_datetime	30	8.57
rust:bit-vec	29	8.29
rust:darling_core	29	8.29
rust:darling_macro	29	8.29
rust:ident_case	29	8.29
rust:pest	29	8.29
rust:phf_shared	29	8.29
rust:rawpointer	29	8.29
rust:sct	29	8.29
rust:winnow	29	8.29
rust:dirs-sys-next	28	8.00
rust:lazycell	28	8.00
rust:matches	28	8.00
rust:plotters	28	8.00

rust:remove_dir_all	28	8.00
rust:sha3	28	8.00
rust:tokio-rustls	28	8.00
rust:windows-core	28	8.00
rust:bit-set	27	7.71
rust:clang-sys	27	7.71
rust:fxhash	27	7.71
rust:nu-ansi-term	27	7.71
rust:zstd	27	7.71
rust:zstd-sys	27	7.71
rust:arrayref	26	7.43
rust:cexpr	26	7.43
rust:dirs	26	7.43
rust:hyper-tls	26	7.43
rust:overload	26	7.43
rust:png	26	7.43
rust:zstd-safe	26	7.43
rust:cmake	25	7.14
rust:colored	25	7.14
rust:deranged	25	7.14
rust:dirs-sys	25	7.14
rust:futures-lite	25	7.14
rust:peeking_take_while	25	7.14
rust:proc-macro-hack	25	7.14
rust:pyo3	25	7.14

rust:rand_pcg	25	7.14
rust:sha1	25	7.14
rust:ucd-trie	25	7.14
rust:unicase	25	7.14
rust:webpki-roots	25	7.14
rust:bitvec	24	6.86
rust:cast	24	6.86
rust:crossbeam	24	6.86
rust:encode_unicode	24	6.86
rust:home	24	6.86
rust:keccak	24	6.86
rust:plotters-backend	24	6.86
rust:plotters-svg	24	6.86
rust:shlex	24	6.86
rust:signature	24	6.86
rust:tap	24	6.86
rust:tokio-native-tls	24	6.86
rust:void	24	6.86
rust:cipher	23	6.57
rust:constant_time_eq	23	6.57
rust:rand_distr	23	6.57
rust:rand_xorshift	23	6.57
rust:scoped-tls	23	6.57
rust:base64ct	22	6.29
rust:block-padding	22	6.29

rust:criterion-plot	22	6.29
rust:failure	22	6.29
rust:funty	22	6.29
rust:hyper-rustls	22	6.29
rust:kernel32-sys	22	6.29
rust:matchers	22	6.29
rust:oorandom	22	6.29
rust:os_str_bytes	22	6.29
rust:phf	22	6.29
rust:powerfmt	22	6.29
rust:radium	22	6.29
rust:rustls-webpki	22	6.29
rust:structopt	22	6.29
rust:target-lexicon	22	6.29
rust:tinytemplate	22	6.29
rust:tower	22	6.29
rust:winapi-build	22	6.29
rust:wyz	22	6.29
rust:zeroize_derive	22	6.29
rust:aes	21	6.00
rust:allocator-api2	21	6.00
rust:ciborium	21	6.00
rust:cloudabi	21	6.00
rust:color_quant	21	6.00
rust:concurrent-queue	21	6.00

rust:console	21	6.00
rust:const-oid	21	6.00
rust:curve25519-dalek	21	6.00
rust:data-encoding	21	6.00
rust:der	21	6.00
rust:event-listener	21	6.00
rust:flume	21	6.00
rust:libz-sys	21	6.00
rust:num_enum	21	6.00
rust:parking	21	6.00
rust:phf_generator	21	6.00
rust:sync_wrapper	21	6.00
rust:tower-layer	21	6.00
rust:async-channel	20	5.71
rust:indicatif	20	5.71
rust:linked-hash-map	20	5.71
rust:num_enum_derive	20	5.71
rust:serde_spanned	20	5.71
rust:sha-1	20	5.71
rust:spki	20	5.71
rust:arc-swap	19	5.43
rust:async-task	19	5.43
rust:console_error_panic_hook	19	5.43
rust:dirs-next	19	5.43
rust:new_debug_unreachable	19	5.43

rust:nodrop	19	5.43
rust:pkcs8	19	5.43
rust:prost	19	5.43
rust:tiny-keccak	19	5.43
rust:waker-fn	19	5.43
rust:webpki	19	5.43
rust:atomic-waker	18	5.14
rust:blake2	18	5.14
rust:bzip2-sys	18	5.14
rust:ciborium-io	18	5.14
rust:ciborium-ll	18	5.14
rust:dashmap	18	5.14
rust:derivative	18	5.14
rust:doc-comment	18	5.14
rust:futures-timer	18	5.14
rust:jpeg-decoder	18	5.14
rust:nalgebra	18	5.14
rust:num_threads	18	5.14
rust:ordered-float	18	5.14
rust:pest_derive	18	5.14
rust:polling	18	5.14
rust:signal-hook	18	5.14
rust:threadpool	18	5.14
rust:anes	17	4.86
rust:async-io	17	4.86

rust:async-stream	17	4.86
rust:bit_field	17	4.86
rust:byte-tools	17	4.86
rust:fallible-iterator	17	4.86
rust:gif	17	4.86
rust:globset	17	4.86
rust:indoc	17	4.86
rust:inout	17	4.86
rust:miow	17	4.86
rust:net2	17	4.86
rust:num-conv	17	4.86
rust:pbkdf2	17	4.86
rust:portable-atomic	17	4.86
rust:precomputed-hash	17	4.86
rust:rustls-native-certs	17	4.86
rust:string_cache	17	4.86
rust:tar	17	4.86
rust:terminal_size	17	4.86
rust:utf-8	17	4.86
rust:async-lock	16	4.57
rust:codespan-reporting	16	4.57
rust:ctr	16	4.57
rust:failure_derive	16	4.57
rust:filetime	16	4.57
rust:fuchsia-zircon	16	4.57

rust:fuchsia-zircon-sys	16	4.57
rust:hex-literal	16	4.57
rust:num-derive	16	4.57
rust:prettyplease	16	4.57
rust:structopt-derive	16	4.57
rust:system-configuration	16	4.57
rust:system-configuration-sys	16	4.57
rust:tracing-futures	16	4.57
rust:adler32	15	4.29
rust:aead	15	4.29
rust:async-executor	15	4.29
rust:async-std	15	4.29
rust:async-stream-impl	15	4.29
rust:axum	15	4.29
rust:blake2b_simd	15	4.29
rust:blocking	15	4.29
rust:byte-slice-cast	15	4.29
rust:bytecount	15	4.29
rust:combine	15	4.29
rust:fuchsia-cprng	15	4.29
rust:libredox	15	4.29
rust:md-5	15	4.29
rust:merlin	15	4.29
rust:multimap	15	4.29
rust:pest_generator	15	4.29

rust:pest_meta	15	4.29
rust:prost-types	15	4.29
rust:rdrand	15	4.29
rust:simd-adler32	15	4.29
rust:wait-timeout	15	4.29
rust:weeql	15	4.29
rust:widestring	15	4.29
rust:xml-rs	15	4.29
rust:zip	15	4.29
rust:axum-core	14	4.00
rust:core-graphics	14	4.00
rust:core-graphics-types	14	4.00
rust:impl-trait-for-tuples	14	4.00
rust:jni	14	4.00
rust:jni-sys	14	4.00
rust:lru	14	4.00
rust:mach	14	4.00
rust:matchit	14	4.00
rust:mime_guess	14	4.00
rust:nanorand	14	4.00
rust:nohash-hasher	14	4.00
rust:number_prefix	14	4.00
rust:proptest	14	4.00
rust:prost-build	14	4.00
rust:prost-derive	14	4.00

rust:psm	14	4.00
rust:rand_isaac	14	4.00
rust:rocksdb	14	4.00
rust:serde_bytes	14	4.00
rust:serde_yaml	14	4.00
rust:statrs	14	4.00
rust:term	14	4.00
rust:universal-hash	14	4.00
rust:bzip2	13	3.71
rust:cfg_aliases	13	3.71
rust:crossterm	13	3.71
rust:ctor	13	3.71
rust:ctrlc	13	3.71
rust:diff	13	3.71
rust:downcast-rs	13	3.71
rust:dyn-clone	13	3.71
rust:ed25519-dalek	13	3.71
rust:fdeflate	13	3.71
rust:ff	13	3.71
rust:gloo-timers	13	3.71
rust:group	13	3.71
rust:hostname	13	3.71
rust:iovec	13	3.71
rust:ntapi	13	3.71
rust:parity-scale-codec	13	3.71

rust:parity-scale-codec-derive	13	3.71
rust:phf_codegen	13	3.71
rust:predicates	13	3.71
rust:raw-window-handle	13	3.71
rust:serde_with	13	3.71
rust:serde_with_macros	13	3.71
rust:simba	13	3.71
rust:tiff	13	3.71
rust:uint	13	3.71
rust:unindent	13	3.71
rust:xattr	13	3.71
rust:alloc-no-stdlib	12	3.43
rust:alloc-stdlib	12	3.43
rust:brotli	12	3.43
rust:brotli-decompressor	12	3.43
rust:bs58	12	3.43
rust:cesu8	12	3.43
rust:cpp_demangle	12	3.43
rust:crypto-bigint	12	3.43
rust:crypto-mac	12	3.43
rust:cxx	12	3.43
rust:cxx-build	12	3.43
rust:cxxbridge-flags	12	3.43
rust:cxxbridge-macro	12	3.43
rust:dtoa	12	3.43

rust:ecdsa	12	3.43
rust:ed25519	12	3.43
rust:elliptic-curve	12	3.43
rust:ena	12	3.43
rust:enum-as-inner	12	3.43
rust:fake-simd	12	3.43
rust:fixed-hash	12	3.43
rust:gethostname	12	3.43
rust:hkdf	12	3.43
rust:ignore	12	3.43
rust:impl-codec	12	3.43
rust:Inflector	12	3.43
rust:link-cplusplus	12	3.43
rust:maplit	12	3.43
rust:match_cfg	12	3.43
rust:objc	12	3.43
rust:phf_macros	12	3.43
rust:pretty_assertions	12	3.43
rust:primitive-types	12	3.43
rust:pyo3-build-config	12	3.43
rust:pyo3-macros	12	3.43
rust:pyo3-macros-backend	12	3.43
rust:rand_jitter	12	3.43
rust:rand_os	12	3.43
rust:rustc-hex	12	3.43

rust:safe_arch	12	3.43
rust:scratch	12	3.43
rust:tempdir	12	3.43
rust:tungstenite	12	3.43
rust:twox-hash	12	3.43
rust:urlencoding	12	3.43
rust:utf8-ranges	12	3.43
rust:ws2_32-sys	12	3.43
rust:x25519-dalek	12	3.43
rust:yaml-rust	12	3.43
rust:aes-gcm	11	3.14
rust:base16ct	11	3.14
rust:blake3	11	3.14
rust:bytemuck_derive	11	3.14
rust:critical-section	11	3.14
rust:derive_builder	11	3.14
rust:float-cmp	11	3.14
rust:impl-serde	11	3.14
rust:lz4-sys	11	3.14
rust:malloc_buf	11	3.14
rust:md5	11	3.14
rust:nalgebra-macros	11	3.14
rust:ndk-sys	11	3.14
rust:polyval	11	3.14
rust:pyo3-ffi	11	3.14

rust:quickcheck	11	3.14
rust:rfc6979	11	3.14
rust:rusty-fork	11	3.14
rust:sec1	11	3.14
rust:secp256k1	11	3.14
rust:tokio-tungstenite	11	3.14
rust:tonic	11	3.14
rust:wasm-bindgen-test	11	3.14
rust:wasm-parser	11	3.14
rust:wide	11	3.14
rust:arbitrary	10	2.86
rust:ark-ff	10	2.86
rust:ark-std	10	2.86
rust:async-global-executor	10	2.86
rust:async-process	10	2.86
rust:backtrace-sys	10	2.86
rust:block	10	2.86
rust:clipboard-win	10	2.86
rust:config	10	2.86
rust:crc	10	2.86
rust:crossterm-winapi	10	2.86
rust:derive-new	10	2.86
rust:directories-next	10	2.86
rust:error-code	10	2.86
rust:event-listener-strategy	10	2.86

rust:exr	10	2.86
rust:fiat-crypto	10	2.86
rust:fs2	10	2.86
rust:ghash	10	2.86
rust:hash32	10	2.86
rust:heapless	10	2.86
rust:k256	10	2.86
rust:lebe	10	2.86
rust:lru-cache	10	2.86
rust:ndk	10	2.86
rust:ndk-context	10	2.86
rust:platforms	10	2.86
rust:predicates-core	10	2.86
rust:predicates-tree	10	2.86
rust:prometheus	10	2.86
rust:serde_repr	10	2.86
rust:snap	10	2.86
rust:unreachable	10	2.86

REFERENCES CITED

- A. Abdellatif, I. S. M. A. G., M. Wessel, & Shihab, E. (2022). Bothunter: An approach to detect software bots in github. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796262&isnumber=9796155> doi: 10.1145/3524842.3527959
- A. Ait, J. L. C. I., & Cabot, J. (2022). An empirical study on the survival rate of github projects. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796216&isnumber=9796155> doi: 10.1145/3524842.3527941
- A. A. Sawant, A. v. D., M. Aniche, & Bacchelli, A. (2018). Understanding developers' needs on deprecation as a language feature. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453124&isnumber=8453044> doi: 10.1145/3180155.3180170
- A. A. Sawant, G. V. S. S., G. Huang, & Bacchelli, A. (2018). Why are features deprecated? an investigation into the motivation behind deprecation. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8529833&isnumber=8529823> doi: 10.1109/ICSME.2018.00011
- A. Barcomb, D. R., K. J. Stol, & Fitzgerald, B. (2019). Why do episodic volunteers stay in floss communities? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8811892&isnumber=8811891> doi: 10.1109/ICSE.2019.00100
- Abdellatif, A., Costa, D., Badran, K., Abdalkareem, R., & Shihab, E. (2020). Challenges in chatbot development: A study of stack overflow posts. In *Msr*. Retrieved from <https://doi.org/10.1145/3379597.3387472> doi: 10.1145/3379597.3387472
- Abid, N. J., Sharif, B., Dragan, N., Alrasheed, H., & Maletic, J. I. (2019). Developer reading behavior while summarizing java methods: Size and context matters. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812039&isnumber=8811891> doi: 10.1109/ICSE.2019.00052

- Accioly, P., Borba, P., Silva, L., & Cavalcanti, G. (2018). Analyzing conflict predictors in open-source java projects. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196437> doi: 10.1145/3196398.3196437
- A. Chueshev, R. B., J. Lawall, & Ziadi, T. (2020). Expanding the number of reviewers in open-source projects by recommending appropriate developers. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240650&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00054
- A. D. Rodriguez, J. C.-H., & Falessi, D. (2021). Leveraging intermediate artifacts to improve automated trace link retrieval. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609143&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00014
- A. Galappaththi, S. N., & Treude, C. (2022). Does this apply to me? an empirical study of technical context in stack overflow. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796260&isnumber=9796155> doi: 10.1145/3524842.3528435
- Aghajani, E. (2018). Context-aware software documentation. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530093&isnumber=8529823> doi: 10.1109/ICSME.2018.00090
- Aghajani, E., Nagy, C., Linares-Vasquez, M., Moreno, L., Bavota, G., Lanza, M., & Shepherd, D. C. (2020). Software documentation: The practitioners' perspective. In *Proceedings of the acm/ieee 42nd international conference on software engineering* (p. 590–601). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3377811.3380405> doi: 10.1145/3377811.3380405
- Aghajani, E., Nagy, C., Vega-Marquez, O. L., Linares-Vasquez, M., Moreno, L., Bavota, G., & Lanza, M. (2019). Software documentation issues unveiled. In *2019 ieee/acm 41st international conference on software engineering (icse)* (p. 1199-1210). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8811931&isnumber=8811891> doi: 10.1109/ICSE.2019.00122

- Agrawal, A., & Menzies, T. (2018). Is "better data" better than "better data miners"? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453183&isnumber=8453044> doi: 10.1145/3180155.3180197
- A. Head, E. M.-H., C. Sadowski, & Knight, A. (2018). When not to comment: Questions and tradeoffs with api documentation for c++ projects. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453133&isnumber=8453044> doi: 10.1145/3180155.3180176
- Ahmed, T., & Devanbu, P. (2022). Multilingual training for software engineering. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794126&isnumber=9793541> doi: 10.1145/3510003.3510049
- A. LeClair, A. B., & McMillan, C. (2021). Ensemble models for neural source code summarization of subroutines. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609119&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00032
- A. LeClair, Z. E., & McMillan, C. (2018). Adapting neural text classification for improved software categorization. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530052&isnumber=8529823> doi: 10.1109/ICSME.2018.00056
- Alhamed, M., & Storer, T. (2021). Playing planning poker in crowds: Human computation of software effort estimates. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402047&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00014
- Alomar, E. A. (2019). Towards better understanding developer perception of refactoring. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919112&isnumber=8918933> doi: 10.1109/ICSME.2019.00100
- Alqadi, B. S. (2019). The relationship between cognitive complexity and the probability of defects. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8918939&isnumber=8918933> doi: 10.1109/ICSME.2019.00095

- Alsuhaibani, R., Newman, C., Decker, M., Collard, M., & Maletic, J. (2021). On the naming of methods: A survey of professional developers. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9401967&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00061
- Amaral, L., Oliveira, M. C., Luz, W., Fortes, J., Bonifacio, R., Alencar, D., . . . Lo, D. (2020). How (not) to find bugs: The interplay between merge conflicts, co-changes, and bugs. In *2020 ieee international conference on software maintenance and evolution (icsme)* (p. 441-452). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240688&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00049
- A. Marques, N. C. B., & Murphy, G. C. (2020). Characterizing task-relevant information in natural language software artifacts. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240640&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00052
- A. Mastropaolo, L. P., E. Aghajani, & Bavota, G. (2021). An empirical study on code comment completion. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609149&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00021
- Amlekar, R., Gamboa, A. F. R., Gallaba, K., & McIntosh, S. (2018). Do software engineers use autocompletion features differently than other developers? In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196471> doi: 10.1145/3196398.3196471
- Anderson, J., Steinmacher, I., & Rodeghero, P. (2020). Assessing the characteristics of foss contributions in network automation projects. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240613&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00039
- Aniche, M., Treude, C., Steinmacher, I., Wiese, I., Pinto, G., Storey, M.-A., & Gerosa, M. A. (2018). How modern news aggregators help development communities shape and share knowledge. In *2018 ieee/acm 40th international conference on software engineering (icse)* (p. 499-510). doi: 10.1145/3180155.3180180

- A. Sarkar, P. C. R., & Bartalos, B. (2019). Improving bug triaging with high confidence predictions at ericsson. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919115&isnumber=8918933> doi: 10.1109/ICSME.2019.00018
- A. Trautsch, S. H., & Grabowski, J. (2020). Static source code metrics and static analysis warnings for fine-grained just-in-time defect prediction. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240701&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00022
- A. Utture, C. G. K., S. Liu, & Palsberg, J. (2022). Striking a balance: Pruning false-positives from static call graphs. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794059&isnumber=9793541> doi: 10.1145/3510003.3510166
- Balachandran, V. (2020). On the need for automatic knowledge management in modern collaboration tools to improve software maintenance. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240629&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00080
- Baltes, S., Dumani, L., Treude, C., & Diehl, S. (2018). Sotorrent: Reconstructing and analyzing the evolution of stack overflow posts. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196430> doi: 10.1145/3196398.3196430
- Barbez, A., Khomh, F., & Guéhéneuc, Y. G. (2019). Deep learning anti-patterns from code metrics history. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919037&isnumber=8918933> doi: 10.1109/ICSME.2019.00021
- B. da Silva, A. R., C. Hebert, & Sereesathien, S. (2020). Robin: A voice controlled virtual teammate for software developers and teams. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240682&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00092
- Bellman, C., Seet, A., & Baysal, O. (2018). Studying developer build issues and debugger usage via timeline analysis in visual studio ide. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196463> doi: 10.1145/3196398.3196463

- Benkoczi, R., Gaur, D., Hossain, S., & Khan, M. A. (2018). A design structure matrix approach for measuring co-change-modularity of software products. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196409> doi: 10.1145/3196398.3196409
- Bernardo, J. a. H., da Costa, D. A., & Kulesza, U. (2018). Studying the impact of adopting continuous integration on the delivery time of pull requests. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196421> doi: 10.1145/3196398.3196421
- Biswas, E., Vijay-Shanker, K., & Pollock, L. (2019). Exploring word embedding techniques to improve sentiment analysis of software engineering texts. In *Msr*. Retrieved from <https://doi.org/10.1109/MSR.2019.00020> doi: 10.1109/MSR.2019.00020
- B. Kou, M. C., Y. Di, & Zhang, T. (2022). Sosum: A dataset of stack overflow post summaries. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796176&isnumber=9796155> doi: 10.1145/3524842.3528487
- B. Lin, G. B. M. D. P., F. Zampetti, & Lanza, M. (2019). Pattern-based mining of opinions in q&a websites. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8811960&isnumber=8811891> doi: 10.1109/ICSE.2019.00066
- Borges, H., Hora, A., & Valente, M. T. (2016). Understanding the factors that impact the popularity of github repositories. In *2016 ieee international conference on software maintenance and evolution (icsme)* (p. 334-344). doi: 10.1109/ICSME.2016.31
- Bulmer, T., Montgomery, L., & Damian, D. (2018). Predicting developers' ide commands with machine learning. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196459> doi: 10.1145/3196398.3196459
- Calefato, F., Lanubile, F., Maiorano, F., & Novielli, N. (2018). Sentiment polarity detection for software development. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453070&isnumber=8453044> doi: 10.1145/3180155.3182519
- Carver, J. C., Weber, N., Ram, K., Gesing, S., & Katz, D. S. (2022). A survey of the state of the practice for research software in the united states. In *Peerj computer science 8:e963*. doi: 10.7717/peerj-cs.963

- C. Brindescu, A. S., Y. Ramirez, & Jensen, C. (2020). Lifting the curtain on merge conflict resolution: A sensemaking perspective. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240655&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00057
- C. Brindescu, R. L., I. Ahmed, & Sarma, A. (2020). Planning for untangling: Predicting the difficulty of merge conflicts. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284005&isnumber=9283910> doi: nan
- C. Gote, F. S., P. Mavrodiev, & Scholtes, I. (2022). Big data = big insights? operationalising brooks' law in a massive github data set. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794099&isnumber=9793541> doi: 10.1145/3510003.3510619
- Chattopadhyay, S., Nelson, N., Au, A., Morales, N., Sanchez, C., Pandita, R., & Sarma, A. (2020). A tale from the trenches: Cognitive biases and software development. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)* (p. 654-665). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9283965&isnumber=9283910>
- Chidambaram, N., & Mazrae, P. R. (2022). Bot detection in github repositories. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796354&isnumber=9796155> doi: 10.1145/3524842.3528520
- Ciborowska, A., & Damevski, K. (2022). Fast changeset-based bug localization with bert. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794003&isnumber=9793541> doi: 10.1145/3510003.3510042
- Ciniselli, M., Cooper, N., Pascarella, L., Poshyvanyk, D., Penta, M. D., & Bavota, G. (2021). An empirical study on the usage of bert models for code completion. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463129&isnumber=9463063> doi: 10.1109/MSR52588.2021.00024
- Claes, M., Mantyla, M., Kuuttila, M., & Farooq, U. (2018). Towards automatically identifying paid open source developers. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196447> doi: 10.1145/3196398.3196447

- C. Miller, M. A. S. D. F., P. Rodeghero, & Zimmermann, T. (2021). How was your weekend?" software development teams working from home during covid-19. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9401956&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00064
- C. Mills, A. B. G. K. S. C., J. Escobar-Avila, & Haiduc, S. (2019). Tracing with less data: Active learning for classification-based traceability link recovery. In *Icsmc*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919048&isnumber=8918933> doi: 10.1109/ICSME.2019.00020
- C. Mills, J. E.-A., & Haiduc, S. (2018). Automatic traceability maintenance via machine learning classification. In *Icsmc*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530044&isnumber=8529823> doi: 10.1109/ICSME.2018.00045
- C. M. Lüders, A. B., & Maalej, W. (2022). Beyond duplicates: Towards understanding and predicting link types in issue tracking systems. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796252&isnumber=9796155> doi: 10.1145/3524842.3528457
- Cohen, E., & Consens, M. P. (2018). Large-scale analysis of the co-commit patterns of the active developers in github's top repositories. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196436> doi: 10.1145/3196398.3196436
- Compton, R., Frank, E., Patros, P., & Koay, A. (2020). Embedding java classes with code2vec: Improvements from variable obfuscation. In *Msr*. Retrieved from <https://doi.org/10.1145/3379597.3387445> doi: 10.1145/3379597.3387445
- C. Overney, C. K., J. Meinicke, & Vasilescu, B. (2020). How to not get rich: An empirical study of donations in open source. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9283983&isnumber=9283910> doi: nan
- C. Vendome, M. D. P. G. B. M. L.-V., D. German, & Poshyvanyk, D. (2018). To distribute or not to distribute? why licensing bugs matter. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453086&isnumber=8453044> doi: 10.1145/3180155.3180221

- Damevski, K., Chen, H., Shepherd, D. C., Kraft, N. A., & Pollock, L. (2018). Predicting future developer behavior in the ide using topic models. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453171&isnumber=8453044> doi: 10.1145/3180155.3182541
- Degiovanni, R., Castro, P., Arroyo, M., Ruiz, M., Aguirre, N., & Frias, M. (2018). Goal-conflict likelihood assessment based on model counting. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453193&isnumber=8453044> doi: 10.1145/3180155.3180261
- de Souza Santos, R. E., & Ralph, P. (2022). A grounded theory of coordination in remote-first and hybrid software teams. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794129&isnumber=9793541> doi: 10.1145/3510003.3510105
- Dey, S., & Woods, W. (2022). Lagoon: An analysis tool for open source communities. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796249&isnumber=9796155> doi: 10.1145/3524842.3528504
- Dey, T., Mousavi, S., Ponce, E., Fry, T., Vasilescu, B., Filippova, A., & Mockus, A. (2020). Detecting and characterizing bots that commit code. In *Msr*. Retrieved from <https://doi.org/10.1145/3379597.3387478> doi: 10.1145/3379597.3387478
- D. Gonzalez, H. H., & Mirakhorli, M. (2019). Automated characterization of software vulnerabilities. In *Icsmc*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8918946&isnumber=8918933> doi: 10.1109/ICSME.2019.00023
- D. Hin, H. C., A. Kan, & Babar, M. A. (2022). Linevd: Statement-level vulnerability detection using graph neural networks. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796269&isnumber=9796155> doi: 10.1145/3524842.3527949
- D. Izquierdo-Cortázar, A. P. G.-P. G. R., J. Alonso-Gutiérrez, & González-Barahona, J. M. (2022). Starting the innersource journey: Key goals and metrics to measure collaboration. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796353&isnumber=9796155> doi: 10.1145/3524842.3528044

- D. J. Kim, T. H. C., N. Tsantalis, & Yang, J. (2021). Studying test annotation maintenance in the wild. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402083&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00019
- D. Kavaler, B. V., A. Trockman, & Filkov, V. (2019). Tool choice matters: Javascript quality assurance tools and usage outcomes in github projects. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812106&isnumber=8811891> doi: 10.1109/ICSE.2019.00060
- D. M. German, G. P.-C. X. Y. H. I., G. Robles, & Inoue, K. (2018). Was my contribution fairly reviewed?" a framework to study the perception of fairness in modern code reviews. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453118&isnumber=8453044> doi: 10.1145/3180155.3180217
- D. N. Palacio, K. M.-C. B.-C. D. P., D. McCrystal, & Shenefiel, C. (2019). Learning to identify security-related issues using convolutional neural networks. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919098&isnumber=8918933> doi: 10.1109/ICSME.2019.00024
- D. Oliveira, F. M., R. Bruno, & Castor, F. (2020). Evaluating code readability and legibility: An examination of human-centric studies. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240710&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00041
- Dong, J., Lou, Y., Zhu, Q., Sun, Z., Li, Z., Zhang, W., & Hao, D. (2022). Fira: Fine-grained graph-based code change representation for automated commit message generation. In *2022 ieee/acm 44th international conference on software engineering (icse)* (p. 970-981). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793882&isnumber=9793541> doi: 10.1145/3510003.3510069
- D. Spadini, A. Z. M. B., F. Palomba, & Bacchelli, A. (2018). On the relation of test smells to software code quality. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8529832&isnumber=8529823> doi: 10.1109/ICSME.2018.00010

- D. Spadini, M. A. S. M. B., M. Aniche, & Bacchelli, A. (2018). When testing meets code review: Why and how developers review tests. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453136&isnumber=8453044> doi: 10.1145/3180155.3180192
- Eberhart, Z., & McMillan, C. (2021). Dialogue management for interactive api search. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609185&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00031
- E. Biswas, L. P., M. E. Karabulut, & Vijay-Shanker, K. (2020). Achieving reliable sentiment analysis in the software engineering domain using bert. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240599&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00025
- E. Dias, F. C. I. S. I. W., P. Meirelles, & Pinto, G. (2021). What makes a great maintainer of open source projects? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402023&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00093
- Efstathiou, V., Chatzilenas, C., & Spinellis, D. (2018). Word embeddings for the software engineering domain. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196448> doi: 10.1145/3196398.3196448
- Efstathiou, V., & Spinellis, D. (2019). Semantic source code models using identifier embeddings. In *Msr*. Retrieved from <https://doi.org/10.1109/MSR.2019.00015> doi: 10.1109/MSR.2019.00015
- E. Kalliamvakou, T. Z. A. B. R. D., C. Bird, & German, D. M. (2018). What makes a great manager of software engineers? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453140&isnumber=8453044> doi: 10.1145/3180155.3182525
- E. Koshchenko, E. K., & Kovalenko, V. (2022). Multimodal recommendation of messenger channels. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796325&isnumber=9796155> doi: 10.1145/3524842.3528441

- El Zarif, O., Da Costa, D. A., Hassan, S., & Zou, Y. (2020). On the relationship between user churn and software issues. In *Msr*. Retrieved from <https://doi.org/10.1145/3379597.3387456> doi: 10.1145/3379597.3387456
- Endres, M., Karas, Z., Hu, X., Kovelman, I., & Weimer, W. (2021). Relating reading, visualization, and coding for new programmers: A neuroimaging study. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402035&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00062
- Fang, H., Klug, D., Lamba, H., Herbsleb, J., & Vasilescu, B. (2020). Need for tweet: How open source developers talk about their github work on twitter. In *Msr*. Retrieved from <https://doi.org/10.1145/3379597.3387466> doi: 10.1145/3379597.3387466
- F. Ebert, N. N., F. Castor, & Serebrenik, A. (2018). Communicative intention in code review questions. In *Icsmc*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530057&isnumber=8529823> doi: 10.1109/ICSME.2018.00061
- F. Heseding, W. S., & Döllner, J. (2022). Tooling for time- and space-efficient git repository mining. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796175&isnumber=9796155> doi: 10.1145/3524842.3528503
- F. Khoshnoud, Z. T., A. R. Nasab, & Sami, A. (2022). Which bugs are missed in code reviews: An empirical study on smartshark dataset. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796172&isnumber=9796155> doi: 10.1145/3524842.3527997
- Foundjem, A., Eghan, E., & Adams, B. (2021). Onboarding vs. diversity, productivity and quality — empirical study of the openstack ecosystem. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402037&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00097
- F. Palomba, A. Z., & Lucia, A. D. (2018). Automatic test smell detection using information retrieval techniques. In *Icsmc*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530039&isnumber=8529823> doi: 10.1109/ICSME.2018.00040

- Frick, V., Grassauer, T., Beck, F., & Pinzger, M. (2018). Generating accurate and compact edit scripts using tree differencing. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530035&isnumber=8529823> doi: 10.1109/ICSME.2018.00036
- Fracz, W., & Dajda, J. (2018). Developers' game: A preliminary study concerning a tool for automated developers assessment. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530082&isnumber=8529823> doi: 10.1109/ICSME.2018.00079
- F. Santos, B. T. I. S. A. S., I. Wiese, & Gerosa, M. A. (2021). Can i solve it? identifying apis required to complete oss tasks. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463078&isnumber=9463063> doi: 10.1109/MSR52588.2021.00047
- F. Sarker, K. B., B. Vasilescu, & Filkov, V. (2019). Socio-technical work-rate increase associates with changes in work patterns in online projects. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8811968&isnumber=8811891> doi: 10.1109/ICSE.2019.00099
- Gallagher, K., Fioravanti, M., & Kozaitis, S. (2019). Teaching software maintenance. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919211&isnumber=8918933> doi: 10.1109/ICSME.2019.00054
- Gamblin, T., LeGendre, M., Collette, M. R., Lee, G. L., Moody, A., de Supinski, B. R., & Futral, S. (2015, November 15-20). The Spack Package Manager: Bringing Order to HPC Software Chaos. Austin, Texas, USA. Retrieved from <https://github.com/spack/spack> (LLNL-CONF-669890) doi: 10.1145/2807591.2807623
- G. Catolino, A. Z., F. Palomba, & Ferrucci, F. (2019). How the experience of development teams relates to assertion density of test classes. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919174&isnumber=8918933> doi: 10.1109/ICSME.2019.00034
- Geiger, F.-X., Malavolta, I., Pascarella, L., Palomba, F., Di Nucci, D., & Bacchelli, A. (2018). A graph-based dataset of commit history of real-world android apps. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196460> doi: 10.1145/3196398.3196460

- Gerosa, M., Wiese, I., Trinkenreich, B., Link, G., Robles, G., Treude, C., ... Sarma, A. (2021). The shifting sands of motivation: Revisiting what drives contributors in open source. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (p. 1046-1058). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402044&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00098
- G. Grano, F. P., C. De Iaco, & Gall, H. C. (2020). Pizza versus pinsa: On the perception and measurability of unit test code quality. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240623&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00040
- Girardi, D., Novielli, N., Fucci, D., & Lanubile, F. (2020). Recognizing developers' emotions while programming. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284015&isnumber=9283910> doi: nan
- G. Kudrjavets, N. N., & Rastogi, A. (2022). The unexplored treasure trove of phabricator code reviews. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796228&isnumber=9796155> doi: 10.1145/3524842.3528005
- G. Kudrjavets, N. N., A. Kumar, & Rastogi, A. (2022). Mining code review data to understand waiting times between acceptance and merging: An empirical analysis. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796315&isnumber=9796155> doi: 10.1145/3524842.3528432
- G. L. Scoccia, P. M., & Autili, M. (2021). Challenges in developing desktop web apps: a study of stack overflow and github. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463138&isnumber=9463063> doi: 10.1109/MSR52588.2021.00039
- G. Nguyen, R. P., M. J. Islam, & Rajan, H. (2022). Manas: Mining software repositories to assist automl. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794122&isnumber=9793541> doi: 10.1145/3510003.3510052

- Gold, N. E., & Krinke, J. (2020). Ethical mining: A case study on msr mining challenges. In *Msr*. Retrieved from <https://doi.org/10.1145/3379597.3387462> doi: 10.1145/3379597.3387462
- Golubev, Y., Eliseeva, M., Povarov, N., & Bryksin, T. (2020). A study of potential code borrowing and license violations in java projects on github. In *Msr*. Retrieved from <https://doi.org/10.1145/3379597.3387455> doi: 10.1145/3379597.3387455
- Gong, J., & Chen, T. (2022). Does configuration encoding matter in learning software performance? an empirical study on encoding schemes. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796199&isnumber=9796155> doi: 10.1145/3524842.3528431
- Gralha, C., Damian, D., Wasserman, A., Goulão, M., & Araújo, J. (2018). The evolution of requirements practices in software startups. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453156&isnumber=8453044> doi: 10.1145/3180155.3180158
- Gren, L., & Ralph, P. (2022). What makes effective leadership in agile software development teams? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794040&isnumber=9793541> doi: 10.1145/3510003.3510100
- G. Rong, L. Y. F. Z. H. K., Y. Zhang, & Zhang, H. (2022). Modeling review history for reviewer recommendation: A hypergraph approach. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793876&isnumber=9793541> doi: 10.1145/3510003.3510213
- Guo, H., & Singh, M. P. (2020). Caspar: Extracting and synthesizing user stories of problems from app reviews. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9283933&isnumber=9283910> doi: nan
- Hadar, I., Hasson, T., Ayalon, O., Toch, E., Birnhack, M., Sherman, S., & Balissa, A. (2018). [journal first] privacy by designers: Software developers' privacy mindset. In *2018 ieee/acm 40th international conference on software engineering (icse)* (p. 396-396). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453098&isnumber=8453044> doi: 10.1145/3180155.3182531

- Hallett, J., Patnaik, N., Shreeve, B., & Rashid, A. (2021). Do this! do that!, and nothing will happen” do specifications lead to securely stored passwords? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402024&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00053
- H. Alsolai, M. R., & Nassar, D. (2018). Predicting software maintainability in object-oriented systems using ensemble techniques. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530091&isnumber=8529823> doi: 10.1109/ICSME.2018.00088
- Hardt, R. (2020a). A software maintenance-focused process and supporting toolset for academic environments. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240620&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00042
- Hardt, R. (2020b). A toolset to support a software maintenance process in academic environments. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240681&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00100
- Hasabnis, N. (2022). Gitrank: A framework to rank github repositories. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796321&isnumber=9796155> doi: 10.1145/3524842.3528519
- Hassan, M., & Hill, E. (2018). Toward automatic summarization of arbitrary java statements for novice programmers. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530061&isnumber=8529823> doi: 10.1109/ICSME.2018.00063
- Hempel, B., Lubin, J., Lu, G., & Chugh, R. (2018). Deuce: A lightweight user interface for structured editing. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453134&isnumber=8453044> doi: 10.1145/3180155.3180165
- H. Fang, J. H., H. Lamba, & Vasilescu, B. (2022). This is damn slick!” estimating the impact of tweets on open source project popularity and new contributors. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794070&isnumber=9793541> doi: 10.1145/3510003.3510121

- H. Hata, R. G. K., C. Treude, & Ishio, T. (2019). 9.6 million links in source code comments: Purpose, evolution, and decay. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8811933&isnumber=8811891> doi: 10.1109/ICSE.2019.00123
- Hilderbrand, C., Perdriau, C., Letaw, L., Emard, J., Steine-Hanson, Z., Burnett, M., & Sarma, A. (2020). Engineering gender-inclusivity into software: Ten teams' tales from the trenches. In *Proceedings of the acm/ieee 42nd international conference on software engineering* (p. 433–444). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3377811.3380371> doi: 10.1145/3377811.3380371
- H. Isotani, Y. F. T. N. S. O., H. Washizaki, & Saito, S. (2021). Duplicate bug report detection by using sentence embedding and fine-tuning. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609125&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00054
- Hora, A. (2021). Googling for software development: What developers search for and what they find. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463121&isnumber=9463063> doi: 10.1109/MSR52588.2021.00044
- Hosono, M., Washizaki, H., Honda, K., Nagumo, H., Sonoda, H., Fukazawa, Y., . . . Monpratarnchai, S. (2019). Inappropriate usage examples in web api documentations. In *2019 ieee international conference on software maintenance and evolution (icsme)* (p. 343–347). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919166&isnumber=8918933> doi: 10.1109/ICSME.2019.00052
- H. Peleg, S. S., & Yahav, E. (2018). Programming not only by example. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453192&isnumber=8453044> doi: 10.1145/3180155.3180189
- H. Phan, N. M. T. L. H. T. A. T. N., H. A. Nguyen, & Nguyen, T. N. (2018). Statistical learning of api fully qualified names in code snippets of online forums. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453132&isnumber=8453044> doi: 10.1145/3180155.3180230

- H. Sellik, G. G., O. van Paridon, & Aniche, M. (2021). Learning off-by-one mistakes: An empirical study. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463090&isnumber=9463063> doi: 10.1109/MSR52588.2021.00019
- H. S. Qiu, A. B. A. S., A. Nolte, & Vasilescu, B. (2019). Going farther together: The impact of social capital on sustained participation in open source. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812044&isnumber=8811891> doi: 10.1109/ICSE.2019.00078
- Hu, Y., Wang, H., Ji, T., Xiao, X., Luo, X., Gao, P., & Guo, Y. (2021). Champ: Characterizing undesired app behaviors from user comments based on market policies. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (p. 933-945). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402070&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00089
- Huang, Y., Liu, X., Krueger, R., Santander, T., Hu, X., Leach, K., & Weimer, W. (2019). Distilling neural representations of data structure manipulation using fmri and fnirs. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)* (p. 396-407).
- I. Ferreira, B. A., & Cheng, J. (2022). How heated is it? understanding github locked issues. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796356&isnumber=9796155> doi: 10.1145/3524842.3527957
- I. Malavolta, B. F. M. B., R. Verdecchia, & Lago, P. (2018). How maintainability issues of android apps evolve. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530041&isnumber=8529823> doi: 10.1109/ICSME.2018.00042
- Imran, A. (2019). Design smell detection and analysis for open source java software. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919054&isnumber=8918933> doi: 10.1109/ICSME.2019.00104
- Ingram, C., & Drachen, A. (2020). How software practitioners use informal local meetups to share software engineering knowledge. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9283934&isnumber=9283910> doi: nan

- Ishio, T., Maeda, N., Shibuya, K., & Inoue, K. (2018). Cloned buggy code detection in practice using normalized compression distance. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530071&isnumber=8529823> doi: 10.1109/ICSME.2018.00022
- I. Steinmacher, I. S. W., G. Pinto, & Gerosa, M. A. (2018). Almost there: A study on quasi-contributors in open-source software projects. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453084&isnumber=8453044> doi: 10.1145/3180155.3180208
- J. Chen, P. W., & Wang, W. (2022). Online summarizing alerts through semantic and behavior information. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793860&isnumber=9793541> doi: 10.1145/3510003.3510055
- J. Dominic, C. R. J. H. C. S., B. Tubre, & Rodeghero, P. (2020). Remote pair programming in virtual reality. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240715&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00046
- J. Fischer, A. S. I. S., S. Bougouffa, & Vogel-Heuser, B. (2018). A qualitative study of variability management of control software for industrial automation systems. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530074&isnumber=8529823> doi: 10.1109/ICSME.2018.00071
- J. G. Young, K. M. M. Z. T. L. H.-D., A. Casari, & Bagrow, J. P. (2021). Which contributions count? analysis of attribution in open source. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463079&isnumber=9463063> doi: 10.1109/MSR52588.2021.00036
- J. Hu, M. J. K. Z. G., J. Joung, & Seltzer, M. I. (2020). Improving data scientist efficiency with provenance. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284103&isnumber=9283910> doi: nan
- J. Johnson, N. Y. J. A., S. Lubo, & Sharif, B. (2019). An empirical study assessing source code readability in comprehension. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8918951&isnumber=8918933> doi: 10.1109/ICSME.2019.00085

- J. Krüger, W. F. G. S., J. Wiemann, & Leich, T. (2018). Do you remember this source code? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453149&isnumber=8453044> doi: 10.1145/3180155.3180215
- J. Lin, Q. Z. M. J., Y. Liu, & Cleland-Huang, J. (2021). Traceability transformed: Generating more accurate links with pre-trained bert models. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402118&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00040
- J. Pantiuchina, M. L., & Bavota, G. (2018). Improving code: The (mis) perception of quality metrics. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530019&isnumber=8529823> doi: 10.1109/ICSME.2018.00017
- J. Pasuksmit, P. T., & Karunasekera, S. (2021). Towards just-enough documentation for agile effort estimation: What information should be documented? In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609208&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00017
- J. Pasuksmit, P. T., & Karunasekera, S. (2022). Towards reliable agile iterative planning via predicting documentation changes of work items. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796335&isnumber=9796155> doi: 10.1145/3524842.3528445
- J. Sun, R. C. H. B. J. W., Z. Xing, & Peng, X. (2019). Know-how in programming tasks: From textual tutorials to task-oriented knowledge graph. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919028&isnumber=8918933> doi: 10.1109/ICSME.2019.00039
- Ju, A., Sajnani, H., Kelly, S., & Herzig, K. (2021). A case study of onboarding in software teams: Tasks and strategies. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9401978&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00063
- J. Zhang, H. Z. H. S., X. Wang, & Liu, X. (2020). Retrieval-based neural source code summarization. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284039&isnumber=9283910> doi: nan

- Khalil, Z. A. (2019). Studying the impact of policy changes on bug handling performance. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8918977&isnumber=8918933> doi: 10.1109/ICSME.2019.00093
- Kim, K., Kim, D., Bissyandé, T. F., Choi, E., Li, L., Klein, J., & Le Traon, Y. (2018). Facoy – a code-to-code search engine. In *2018 IEEE/ACM 40th International Conference on Software Engineering (ICSE)* (p. 946-957). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453174&isnumber=8453044> doi: 10.1145/3180155.3180187
- Klein, B. P., & Henley, A. Z. (2021). Coderibbon: More efficient workspace management and navigation for mainstream development environments. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609170&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00065
- K. M. Subash, S. L. V. P. C., L. P. Kumar, & Baysal, O. (2022). Disco: A dataset of discord chat conversations for software engineering research. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796319&isnumber=9796155> doi: 10.1145/3524842.3528018
- K. Noda, H. Y., & Kikuchi, S. (2021). Sirius: Static program repair with dependence graph-based systematic edit patterns. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609220&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00045
- Koopman, B., & Zuccon, G. (2023, December). Dr ChatGPT tell me what I want to hear: How different prompts impact health answer correctness. In H. Bouamor, J. Pino, & K. Bali (Eds.), *Proceedings of the 2023 conference on empirical methods in natural language processing* (pp. 15012–15022). Singapore: Association for Computational Linguistics. Retrieved from <https://aclanthology.org/2023.emnlp-main.928> doi: 10.18653/v1/2023.emnlp-main.928
- K. P. Neupane, K. C., & Wang, Y. (2019). Emod: An end-to-end approach for investigating emotion dynamics in software development. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919242&isnumber=8918933> doi: 10.1109/ICSME.2019.00038

- Krueger, R., Huang, Y., Liu, X., Santander, T., Weimer, W., & Leach, K. (2020). Neurological divide: An fmri study of prose and code writing. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284060&isnumber=9283910> doi: nan
- K. Truong, B. V., C. Miller, & Kästner, C. (2022). The unsolvable problem or the unheard answer? a dataset of 24,669 open-source software conference talks. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796182&isnumber=9796155> doi: 10.1145/3524842.3528488
- Kubelka, J., Robbes, R., & Bergel, A. (2018). The road to live programming: Insights from the practice. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453190&isnumber=8453044> doi: 10.1145/3180155.3180200
- Lamothe, M., & Shang, W. (2020). When apis are intentionally bypassed: An exploratory study of api workarounds. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9283953&isnumber=9283910> doi: nan
- L. Braz, G. , C. Aeberhard, & Bacchelli, A. (2022). Less is more: Supporting developers in vulnerability detection during code review. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794090&isnumber=9793541> doi: 10.1145/3510003.3511560
- L. D. Silva, W. M. T. B., P. Borba, & Moisakis, J. (2020). Detecting semantic conflicts via automated behavior change detection. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240661&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00026
- Le, V. H., & Zhang, H. (2022). Log-based anomaly detection with deep learning: How far are we? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794117&isnumber=9793541> doi: 10.1145/3510003.3510155
- Lee, A., & Carver, J. C. (2019). Floss participants' perceptions about gender and inclusiveness: A survey. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812068&isnumber=8811891> doi: 10.1109/ICSE.2019.00077

- Li, H., Li, S., Sun, J., Xing, Z., Peng, X., Liu, M., & Zhao, X. (2018). Improving api caveats accessibility by mining api caveats knowledge graph. In *2018 ieee international conference on software maintenance and evolution (icsme)* (p. 183-193). doi: 10.1109/ICSME.2018.00028
- Li, J., Huang, R., Li, W., Yao, K., & Tan, W. (2021). Toward less hidden cost of code completion with acceptance and ranking models. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609202&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00024
- Lin, B., Zampetti, F., Bavota, G., Penta, M. D., Lanza, M., & Oliveto, R. (2018). Sentiment analysis for software engineering: How far can we go? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453067&isnumber=8453044> doi: 10.1145/3180155.3180195
- Lin, B., Zampetti, F., Oliveto, R., Penta, M. D., Lanza, M., & Bavota, G. (2018). Two datasets for sentiment analysis in software engineering. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530087&isnumber=8529823> doi: 10.1109/ICSME.2018.00084
- Liu, K., Kim, D., Bissyandé, T. F., Kim, T., Kim, K., Koyuncu, A., ... Le Traon, Y. (2019). Learning to spot and refactor inconsistent method names. In *2019 ieee/acm 41st international conference on software engineering (icse)* (p. 1-12). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812134&isnumber=8811891> doi: 10.1109/ICSE.2019.00019
- L. Moldon, M. S., & Wachs, J. (2021). How gamification affects software developers: Cautionary evidence from a natural experiment on github. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402097&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00058
- L. Shi, M. L. Y. W. S. L., M. Xing, & Wang, Q. (2020). Detection of hidden feature requests from massive chat messages via deep siamese network. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9283914&isnumber=9283910> doi: nan
- L. Sui, A. T., J. Dietrich, & Fourtounis, G. (2020). On the recall of static call graph construction in practice. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9283958&isnumber=9283910> doi: nan

- Luan, S., Yang, D., Barnaby, C., Sen, K., & Chandra, S. (2019, oct). Aroma: Code recommendation via structural code search. *Proc. ACM Program. Lang.*, 3(OOPSLA). Retrieved from <https://doi.org/10.1145/3360578> doi: 10.1145/3360578
- L. Yin, Q. X., Z. Zhang, & Filkov, V. (2021). Apache software foundation incubator project sustainability dataset. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463075&isnumber=9463063> doi: 10.1109/MSR52588.2021.00081
- Lyulina, E., & Jahanshahi, M. (2021). Building the collaboration graph of open-source software ecosystem. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463070&isnumber=9463063> doi: 10.1109/MSR52588.2021.00086
- Ma, W., Zhao, M., Soremekun, E., Hu, Q., Zhang, J. M., Papadakis, M., . . . Le Traon, Y. (2022). Graphcode2vec: Generic code embedding via lexical and program dependence analyses. In *2022 ieee/acm 19th international conference on mining software repositories (msr)* (p. 524-536). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796339&isnumber=9796155> doi: 10.1145/3524842.3528456
- M. A. Al Alamin, G. U. S. A. T. B. H., S. Malakar, & Iqbal, A. (2021). An empirical study of developer discussions on low-code software development challenges. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463132&isnumber=9463063> doi: 10.1109/MSR52588.2021.00018
- Majumder, S., Balaji, N., Brey, K., Fu, W., & Menzies, T. (2018). 500+ times faster than deep learning: A case study exploring faster methods for text mining stackoverflow. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196424> doi: 10.1145/3196398.3196424
- Manes, S. S., & Baysal, O. (2021). Studying the change histories of stack overflow and github snippets. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463143&isnumber=9463063> doi: 10.1109/MSR52588.2021.00040
- Mantyla, M. V., Calefato, F., & Claes, M. (2018). Natural language or not (nlon): A package for software engineering text analysis pipeline. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196444> doi: 10.1145/3196398.3196444

- Mashhadi, E., & Hemmati, H. (2021). Applying codebert for automated program repair of java simple bugs. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463106&isnumber=9463063> doi: 10.1109/MSR52588.2021.00063
- Masud, M. R., & Faloutsos, M. (2024). *Unveiling a hidden risk: Exposing educational but malicious repositories in github*.
- Mayr-Dorn, C., & Egyed, A. (2018). Does the propagation of artifact changes across tasks reflect work dependencies? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453099&isnumber=8453044> doi: 10.1145/3180155.3180185
- McHugh, M. L. (2012). *Interrater reliability: the kappa statistic*.
- M. Ciniselli, L. P., & Bavota, G. (2022). To what extent do deep learning-based code recommenders generate predictions by cloning code from the training set? In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796377&isnumber=9796155> doi: 10.1145/3524842.3528440
- M. Claes, M. K., M. V. Mäntylä, & Adams, B. (2018). Do programmers work at night or during the weekend? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453143&isnumber=8453044> doi: 10.1145/3180155.3180193
- Mendez, C., Padala, H. S., Steine-Hanson, Z., Hildebrand, C., Horvath, A., Hill, C., ... Burnett, M. (2018). Open source barriers to entry, revisited: A sociotechnical perspective. In *2018 ieee/acm 40th international conference on software engineering (icse)* (p. 1004-1015). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453179&isnumber=8453044> doi: 10.1145/3180155.3180241
- M. Endres, K. B., & Weimer, W. (2022). Hashing it out: A survey of programmers' cannabis usage, perception, and motivation. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793986&isnumber=9793541> doi: 10.1145/3510003.3510156

- M. Galster, C. T., & Blincoe, K. (2019). Supporting software architecture maintenance by providing task-specific recommendations. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8918960&isnumber=8918933> doi: 10.1109/ICSME.2019.00060
- M. Hammoudi, A. M., C. Mayr-Dorn, & Egyed, A. (2021). A traceability dataset for open source systems. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463088&isnumber=9463063> doi: 10.1109/MSR52588.2021.00073
- M. H. Houekpetodji, S. D. F. D., N. Anquetil, & Sudich, J. (2021). Report from the trenches a case study in modernizing software development practices. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609169&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00052
- M. Iammarino, L. A., F. Zampetti, & Penta, M. D. (2019). Self-admitted technical debt removal and refactoring actions: Co-occurrence or more? In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919002&isnumber=8918933> doi: 10.1109/ICSME.2019.00029
- Middleton, J., Murphy-Hill, E., Green, D., Meade, A., Mayer, R., White, D., & McDonald, S. (2018). Which contributions predict whether developers are accepted into github teams. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196429> doi: 10.1145/3196398.3196429
- Mir, A. M., Latoškinas, E., Proksch, S., & Gousios, G. (2022). Type4py: Practical deep similarity learning-based type inference for python. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793925&isnumber=9793541> doi: 10.1145/3510003.3510124
- M. Jimenez, Y. L. T., C. Maxime, & Papadakis, M. (2018a). On the impact of tokenizer and parameters on n-gram based code analysis. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530050&isnumber=8529823> doi: 10.1109/ICSME.2018.00053
- M. Jimenez, Y. L. T., C. Maxime, & Papadakis, M. (2018b). Tuna: Tuning naturalness-based analysis. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530090&isnumber=8529823> doi: 10.1109/ICSME.2018.00087

- M. Kim, R. D., T. Zimmermann, & Begel, A. (2018). Data scientists in software teams: State of the art and challenges. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453127&isnumber=8453044> doi: 10.1145/3180155.3182515
- M. Mondal, C. K. R., B. Roy, & Schneider, K. A. (2019). Investigating context adaptation bugs in code clones. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919065&isnumber=8918933> doi: 10.1109/ICSME.2019.00026
- M. Motwani, R. J., S. Sankaranarayanan, & Brun, Y. (2018). Do automated program repair techniques repair hard and important bugs? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453058&isnumber=8453044> doi: 10.1145/3180155.3182533
- M. O. F. Rokon, R. I., P. Yan, & Faloutsos, M. (2021). Repo2vec: A comprehensive embedding approach for determining repository similarity. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609112&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00038
- Moran, K., Palacio, D. N., Bernal-Cárdenas, C., McCrystal, D., Poshyvanyk, D., Shenefiel, C., & Johnson, J. (2020). Improving the effectiveness of traceability link recovery using hierarchical bayesian networks. In *2020 ieee/acm 42nd international conference on software engineering (icse)* (p. 873-885). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284034&isnumber=9283910>
- M. Osama, M. A. J. G., A. Zaki-Ismail, & Ibrahim, A. (2020). Score-based automatic detection and resolution of syntactic ambiguity in natural language requirements. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240680&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00067
- Motwani, M., & Brun, Y. (2019). Automatically generating precise oracles from structured natural language specifications. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812070&isnumber=8811891> doi: 10.1109/ICSE.2019.00035

- M. Papoutsoglou, J. W., & Kapitsaki, G. M. (2021). Mining dev for social and technical insights about software development. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463115&isnumber=9463063> doi: 10.1109/MSR52588.2021.00053
- M. P. Robillard, M. N., & McIntosh, S. (2018). Threats of aggregating software repository data. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530085&isnumber=8529823> doi: 10.1109/ICSME.2018.00082
- M. Sridharan, M. C., M. Mäntylä, & Rantala, L. (2022). Soccminer: A source code-comments and comment-context miner. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796266&isnumber=9796155> doi: 10.1145/3524842.3527998
- M. Tushev, F. E., & Mahmoud, A. (2021). Analysis of non-discrimination policies in the sharing economy. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609168&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00016
- M. Tushev, S. K., & Mahmoud, A. (2019). Linguistic change in open source software. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8918934&isnumber=8918933> doi: 10.1109/ICSME.2019.00045
- Mundt, M. R., Beattie, K., Bisila, J., Ferenbaugh, C. R., Godoy, W. F., Gupta, R., ... Teves, J. B. (2022). For the public good: Connecting, retaining, and recognizing current and future rses at u.s. national research laboratories and agencies. *Computing in Science Engineering*, 24(6), 6-13. doi: 10.1109/MCSE.2023.3256759
- Murphy-Hill, E., Smith, E. K., Sadowski, C., Jaspán, C., Winter, C., Jorde, M., ... Gross, S. (2019). Do developers discover new tools on the toilet? In *2019 IEEE/ACM 41st international conference on software engineering (icse)* (p. 465-475). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812046&isnumber=8811891> doi: 10.1109/ICSE.2019.00059
- M. Vigiato, E. F. P. J., J. Oliveira, & Kästner, C. (2019). How do code changes evolve in different platforms? a mining-based investigation. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919007&isnumber=8918933> doi: 10.1109/ICSME.2019.00033

- M. Wessel, I. W. I. S., A. Serebrenik, & Gerosa, M. A. (2020). Effects of adopting code review bots on pull requests to oss projects. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240622&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00011
- M. Wyrich, D. G., A. Preikschat, & Wagner, S. (2021). The mind is a powerful place: How showing code comprehensibility metrics influences code understanding. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402040&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00055
- Nayrolles, M., & Hamou-Lhadj, A. (2018). Clever: Combining code metrics with clone detection for just-in-time fault prevention and resolution in large industrial projects. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196438> doi: 10.1145/3196398.3196438
- N. Cassee, E. C., C. Kitsanelis, & Serebrenik, A. (2021). Human, bot or both? a study on the capabilities of classification models on mixed accounts. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609135&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00075
- N. Cooper, O. C. K. M., C. Bernal-Cárdenas, & Poshyvanyk, D. (2021). It takes two to tango: Combining visual and textual information for detecting duplicate video-based bug reports. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9401992&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00091
- Nguyen, N., & Nadi, S. (2022). An empirical evaluation of github copilot's code suggestions. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796235&isnumber=9796155> doi: 10.1145/3524842.3528470
- Nguyen, P. T., Rocco, J. D., Ruscio, D. D., Ochoa, L., Degueule, T., & Penta, M. D. (2019). Focus: A recommender system for mining api function calls and usage patterns. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812051&isnumber=8811891> doi: 10.1109/ICSE.2019.00109
- Nguyen, S., Phan, H., Le, T., & Nguyen, T. N. (2020). Suggesting natural method names to check name consistencies. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284075&isnumber=9283910> doi: nan

- Nguyen, T., Vu, P., & Nguyen, T. (2019a). Personalized code recommendation. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919008&isnumber=8918933> doi: 10.1109/ICSME.2019.00047
- Nguyen, T., Vu, P., & Nguyen, T. (2019b). Recommending exception handling code. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919036&isnumber=8918933> doi: 10.1109/ICSME.2019.00066
- N. Imtiaz, J. C. N. R. G. B., J. Middleton, & Murphy-Hill, E. (2019). Investigating the effects of gender bias on github. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812110&isnumber=8811891> doi: 10.1109/ICSE.2019.00079
- Niu, C., Li, C., Ng, V., Ge, J., Huang, L., & Luo, B. (2022). Spt-code: Sequence-to-sequence pre-training for learning source code representations. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793930&isnumber=9793541> doi: 10.1145/3510003.3510096
- N. Nahar, G. L., S. Zhou, & Kästner, C. (2022). Collaboration challenges in building ml-enabled systems: Communication, documentation, engineering, and process. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793960&isnumber=9793541> doi: 10.1145/3510003.3510209
- Novielli, N., Calefato, F., & Lanubile, F. (2018). A gold standard for emotion annotation in stack overflow. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196453> doi: 10.1145/3196398.3196453
- N. Rao, M. H., J. Tsay, & Hellendoorn, V. J. (2022). Comments on comments: Where code review and documentation meet. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796192&isnumber=9796155> doi: 10.1145/3524842.3528475
- N. Riquet, X. D., & Vanderose, B. (2022). Gtdelver enterprise dataset (gded): An industrial closed-source dataset for socio-technical research. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796217&isnumber=9796155> doi: 10.1145/3524842.3528003

- N. Sae-Lim, S. H., & Saeki, M. (2019). Can automated impact analysis techniques help predict decaying modules? In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919044&isnumber=8918933> doi: 10.1109/ICSME.2019.00088
- N. Shimada, H. H. C. T., T. Xiao, & Matsumoto, K. (2022). Github sponsors: Exploring a new way to contribute to open source. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794031&isnumber=9793541> doi: 10.1145/3510003.3510116
- O. Dabic, E. A., & Bavota, G. (2021). Sampling projects in github for msr studies. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463094&isnumber=9463063> doi: 10.1109/MSR52588.2021.00074
- O. Elazhary, N. E., M. A. Storey, & Zaidman, A. (2019). Do as i do, not as i say: Do contribution guidelines match the github contribution process? In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919187&isnumber=8918933> doi: 10.1109/ICSME.2019.00043
- P. Chatterjee, K. D., & Pollock, L. (2021). Automatic extraction of opinion-based q&a from online developer chats. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402078&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00115
- P. Chatterjee, T. S., & Ralph, P. (2022). Empirical standards for repository mining. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796211&isnumber=9796155> doi: 10.1145/3524842.3528032
- Pecorelli, F., Palomba, F., Khomh, F., & De Lucia, A. (2020). Developer-driven code smell prioritization. In *Msr*. Retrieved from <https://doi.org/10.1145/3379597.3387457> doi: 10.1145/3379597.3387457
- Pei, J., Wu, Y., Qin, Z., Cong, Y., & Guan, J. (2021). Attention-based model for predicting question relatedness on stack overflow. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463123&isnumber=9463063> doi: 10.1109/MSR52588.2021.00023

- Perera, H., Hussain, W., Whittle, J., Nurwidyantoro, A., Mougouei, D., Shams, R. A., & Oliver, G. (2020). A study on the prevalence of human values in software engineering publications, 2015 – 2018. In *Proceedings of the acm/ieee 42nd international conference on software engineering* (p. 409–420). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3377811.3380393> doi: 10.1145/3377811.3380393
- Peruma, A. (2019). Towards a model to appraise and suggest identifier names. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8918988&isnumber=8918933> doi: 10.1109/ICSME.2019.00103
- Peruma, A., Arnaoudova, V., & Newman, C. D. (2021). Ideal: An open-source identifier name appraisal tool. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609217&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00064
- Pfeiffer, R.-H. (2020). What constitutes software? an empirical, descriptive study of artifacts. In *Msr*. Retrieved from <https://doi.org/10.1145/3379597.3387442> doi: 10.1145/3379597.3387442
- P. Gnoyke, S. S., & Krüger, J. (2021). An evolutionary analysis of software-architecture smells. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609226&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00043
- P. P. Pârachi, C. T., S. K. Dash, & Barr, E. T. (2020). Posit: Simultaneously tagging natural and programming languages. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284084&isnumber=9283910> doi: nan
- P. R. Mazrae, M. I., & Heydarnoori, A. (2021). Automated recovery of issue-commit links leveraging both textual and non-textual data. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609165&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00030
- P. Sri-iesaranusorn, R. G. K., & Ishio, T. (2021). Does code review promote conformance? a study of openstack patches. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463066&isnumber=9463063> doi: 10.1109/MSR52588.2021.00056

- Q. Feng, C. F., & Hong, W. (2020). Graph neural network-based vulnerability predication. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240607&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00096
- Rabbani, N., Harvey, M. S., Saquif, S., Gallaba, K., & McIntosh, S. (2018). Revisiting "programmers' build errors" in the visual studio context: A replication study using ide interaction traces. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196469> doi: 10.1145/3196398.3196469
- Rahman, A. (2018). Comprehension effort and programming activities: Related? or not related? In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196470> doi: 10.1145/3196398.3196470
- Rahman, M. M., Barson, J., Paul, S., Kayani, J., Lois, F. A., Quezada, S. F., . . . Ray, B. (2018). Evaluating how developers use general-purpose web-search for code retrieval. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196425> doi: 10.1145/3196398.3196425
- Ramirez-Mora, S. L., & Oktaba, H. (2018). Team maturity in agile software development: The impact on productivity. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530094&isnumber=8529823> doi: 10.1109/ICSME.2018.00091
- R. Feng, S. P., Z. Yan, & Zhang, Y. (2022). Automated detection of password leakage from public github repositories. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794113&isnumber=9793541> doi: 10.1145/3510003.3510150
- R. Kallis, G. C., A. Di Sorbo, & Panichella, S. (2019). Ticket tagger: Machine learning driven issue classification. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8918993&isnumber=8918933> doi: 10.1109/ICSME.2019.00070
- Rodriguez, A., Tanaka, F., & Kamei, Y. (2018). Empirical study on the relationship between developer's working habits and efficiency. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196458> doi: 10.1145/3196398.3196458

- Rossi, D., & Zacchiroli, S. (2022). Geographic diversity in public code contributions: An exploratory large-scale study over 50 years. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796333&isnumber=9796155> doi: 10.1145/3524842.3528471
- R. Pan, N. N. S. G. S. L., V. Le, & Kaufman, M. (2021). Can program synthesis be used to learn merge conflict resolutions? an empirical analysis. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402036&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00077
- R. Paul, A. K. T., & Bosu, A. (2021). Why security defects go unnoticed during code reviews? a case-control study of the chromium os project. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402130&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00124
- R. Ramsauer, D. L., & Mauerer, W. (2019). The list is the process: Reliable pre-integration tracking of commits on mailing lists. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812060&isnumber=8811891> doi: 10.1109/ICSE.2019.00088
- R. Rwemalika, M. P. Y. L. T., M. Kintis, & Lorrach, P. (2019). An industrial study on the differences between pre-release and post-release bugs. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8919245&isnumber=8918933> doi: 10.1109/ICSME.2019.00019
- R. Tufano, A. M. L. P. D. P., S. Masiero, & Bavota, G. (2022). Using pre-trained models to boost code review automation. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794027&isnumber=9793541> doi: 10.1145/3510003.3510621
- R. Tufano, M. T. D. P., L. Pascarella, & Bavota, G. (2021). Towards automating code review activities. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402025&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00027

- Ruvimova, A., Lill, A., Gugler, J., Howe, L., Huang, E., Murphy, G., & Fritz, T. (2022). An exploratory study of productivity perceptions in software teams. In *2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE)* (p. 99-111). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793555&isnumber=9793541> doi: 10.1145/3510003.3510081
- R. Wen, M. G. R., D. Gilbert, & McIntosh, S. (2018). Blimp tracer: Integrating build impact analysis with code review. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530081&isnumber=8529823> doi: 10.1109/ICSME.2018.00078
- Sabetta, A., & Bezzi, M. (2018). A practical approach to the automatic classification of security-relevant commits. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530068&isnumber=8529823> doi: 10.1109/ICSME.2018.00058
- Saha, R. K., Ura, A., Mahajan, S., Zhu, C., Li, L., Hu, Y., ... Prasad, M. R. (2022). Sapienml: Synthesizing machine learning pipelines by learning from human-written solutions. In *2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE)* (p. 1932-1944). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794084&isnumber=9793541> doi: 10.1145/3510003.3510226
- S. Biswas, M. W., & Rajan, H. (2022). The art and practice of data science pipelines: A comprehensive study of data science pipelines in theory, in-the-small, and in-the-large. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793883&isnumber=9793541> doi: 10.1145/3510003.3510057
- Scalabrino, S., Bavota, G., Vendome, C., Linares-Vasquez, M., Poshyvanyk, D., & Oliveto, R. (2017). Automatically assessing code understandability: How far are we? In *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)* (pp. 417-427).
- S. Chattopadhyay, Y. R. G. A. A. L. R. P., N. Nelson, & Sarma, A. (2019). Latent patterns in activities: A field study of how developers manage context. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8811986&isnumber=8811891> doi: 10.1109/ICSE.2019.00051

- Schwartz, S. D. (2023). A five-year survey of literature in software engineering and repository mining research. In *Research reports, department of computer science, university of oregon*.
- Schwartz, S. D., Fickas, S. F., Norris, B., & Dubey, A. (2024). *A survey of open source software repositories in the us department of energy's national laboratories (under review)*.
- S. Datta, H. W., A. Mysore, & Sarkar, S. (2021). Clustering, separation, and connection: A tale of three characteristics. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609186&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00078
- S. E. Ponta, H. P., & Sabetta, A. (2018). Beyond metadata: Code-centric and usage-based analysis of known vulnerabilities in open-source software. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530051&isnumber=8529823> doi: 10.1109/ICSME.2018.00054
- Sharma, T., & Kessentini, M. (2021a). Qscored: A large dataset of code smells and quality metrics. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463095&isnumber=9463063> doi: 10.1109/MSR52588.2021.00080
- Sharma, T., & Kessentini, M. (2021b). Qscored: A large dataset of code smells and quality metrics. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463095&isnumber=9463063> doi: 10.1109/MSR52588.2021.00080
- Sharma, T., & Sangwan, O. P. (2021). Sine-cosine algorithm for software fault prediction. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609148&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00084
- S. Hassan, C. P. B., C. Tantithamthavorn, & Hassan, A. E. (2018). Studying the dialogue between users and developers of free apps in the google play store. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453075&isnumber=8453044> doi: 10.1145/3180155.3182523

- Shen, B., Zhang, W., Yu, A., Wei, Z., Liang, G., Zhao, H., & Jin, Z. (2021). Cross-language code coupling detection: A preliminary study on android applications. In *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (p. 378-388). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609127&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00040
- Shi, L., Mu, F., Zhang, Y., Yang, Y., Chen, J., Chen, X., ... Wang, Q. (2022). Buglistener: Identifying and synthesizing bug reports from collaborative live chats. In *Proceedings of the 44th international conference on software engineering* (p. 299-311). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3510003.3510108> doi: 10.1145/3510003.3510108
- Shrestha, N., Botta, C., Barik, T., & Parnin, C. (2020). Here we go again: Why is it difficult for developers to learn another programming language? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284077&isnumber=9283910> doi: nan
- Silavong, F., Moran, S., Georgiadis, A., Saphal, R., & Otter, R. (2022). Senatus - a fast and accurate code-to-code recommendation engine. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796316&isnumber=9796155> doi: 10.1145/3524842.3527947
- Sillito, J., & Kutomi, E. (2020). Failures and fixes: A study of software system incident response. In *Icsmc*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240602&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00027
- S. Mondal, G. U., & Roy, C. K. (2020). Automatic identification of rollback edit with reasons in stack overflow q&a site. In *Icsmc*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240665&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00111
- S. Mondal, G. U., & Roy, C. K. (2021). Rollback edit inconsistencies in developer forum. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463153&isnumber=9463063> doi: 10.1109/MSR52588.2021.00050
- S. N.C., S. M., & Menzies, T. (2021). Early life cycle software defect prediction. why? how? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9401968&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00050

- Sousa, B. L., Bigonha, M. A. S., Ferreira, K. A. M., & Franco, G. C. (2022). A time series-based dataset of open-source software evolution. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796226&isnumber=9796155> doi: 10.1145/3524842.3528492
- S. Reddy, R. P., C. Lemieux, & Sen, K. (2020). Quickly generating diverse valid test inputs with reinforcement learning. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284117&isnumber=9283910> doi: nan
- Stanik, C., Montgomery, L., Martens, D., Fucci, D., & Maalej, W. (2018). A simple nlp-based approach to support onboarding and retention in open source communities. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530027&isnumber=8529823> doi: 10.1109/ICSME.2018.00027
- Stefano, M. D., Pecorelli, F., Tamburri, D. A., Palomba, F., & Lucia, A. D. (2020). Refactoring recommendations based on the optimization of socio-technical congruence. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240606&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00094
- Sun, Z., Li, L., Liu, Y., Du, X., & Li, L. (2022). On the importance of building high-quality training datasets for neural code search. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793971&isnumber=9793541> doi: 10.1145/3510003.3510160
- Svyatkovskiy, A., Lee, S., Hadjitofi, A., Riechert, M., Franco, J. V., & Allamanis, M. (2021). Fast and memory-efficient neural code completion. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463109&isnumber=9463063> doi: 10.1109/MSR52588.2021.00045
- S. Wang, T. H. C., & Hassan, A. E. (2018). Understanding the factors for fast answers in technical q&a websites: An empirical study of four stack exchange websites. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453164&isnumber=8453044> doi: 10.1145/3180155.3182521
- S. Wang, T. H. C. Y. T., D. M. German, & Hassan, A. E. (2021). Is reputation on stack overflow always a good indicator for users' expertise? no! In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609200&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00067

- S. Yatish, P. T., J. Jiarpakdee, & Tantithamthavorn, C. (2019). Mining software defects: Should we consider affected releases? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8811982&isnumber=8811891> doi: 10.1109/ICSE.2019.00075
- Tao, W., Wang, Y., Shi, E., Du, L., Han, S., Zhang, H., ... Zhang, W. (2021). On the evaluation of commit message generation models: An experimental study. In *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (p. 126-136). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609189&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00018
- T. Besker, R. E. L. K. B., A. Martini, & Bosch, J. (2018). Embracing technical debt, from a startup company perspective. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530048&isnumber=8529823> doi: 10.1109/ICSME.2018.00051
- T. Dey, A. K., & Mockus, A. (2021). Representation of developer expertise in open source software. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9401957&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00094
- Theeten, B., Vandeputte, F., & Van Cutsem, T. (2019). Import2vec learning embeddings for software libraries. In *Msr*. Retrieved from <https://doi.org/10.1109/MSR.2019.00014> doi: 10.1109/MSR.2019.00014
- T. Hoang, D. L., H. J. Kang, & Lawall, J. (2020). Cc2vec: Distributed representations of code changes. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284081&isnumber=9283910> doi: nan
- T. Lorey, P. R., & Felderer, M. (2022). Social science theories in software engineering research. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793978&isnumber=9793541> doi: 10.1145/3510003.3510076
- Trockman, A., Cates, K., Mozina, M., Nguyen, T., Kastner, C., & Vasilescu, B. (2018). Automatically assessing code understandability” reanalyzed: Combined metrics matter. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196441> doi: 10.1145/3196398.3196441

- Tsantalis, N., Mansouri, M., Eshkevari, L., Mazinanian, D., & Dig, D. (2018). Accurate and efficient refactoring detection in commit history. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453111&isnumber=8453044> doi: 10.1145/3180155.3180206
- T. Sedano, P. R., & Péraire, C. (2019). The product backlog. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812076&isnumber=8811891> doi: 10.1109/ICSE.2019.00036
- Tufano, M., Pantiuchina, J., Watson, C., Bavota, G., & Poshyvanyk, D. (2019). On learning meaningful code changes via neural machine translation. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8811910&isnumber=8811891> doi: 10.1109/ICSE.2019.00021
- T. Zhang, A. R. H. R., G. Upadhyaya, & Kim, M. (2018). Are code examples on an online q&a forum reliable?: A study of api misuse on stack overflow. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453166&isnumber=8453044> doi: 10.1145/3180155.3180260
- T. Zhang, C. L., D. Yang, & Kim, M. (2019). Analyzing and supporting adaptation of online code examples. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812028&isnumber=8811891> doi: 10.1109/ICSE.2019.00046
- T. Zhang, F. T. S. A. H. D. L., B. Xu, & Jiang, L. (2020). Sentiment analysis for software engineering: How far can pre-trained transformer models go? In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240704&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00017
- Uchoa, A., Barbosa, C., Coutinho, D., Oizumi, W., Assuncao, W. K. G., Vergilio, S. R., ... Garcia, A. (2021). Predicting design impactful changes in modern code review: A large-scale empirical study. In *2021 ieee/acm 18th international conference on mining software repositories (msr)* (p. 471-482). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463137&isnumber=9463063> doi: 10.1109/MSR52588.2021.00059

- Uchoa, A., Barbosa, C., Oizumi, W., Blenilio, P., Lima, R., Garcia, A., & Bezerra, C. (2020). How does modern code review impact software design degradation? an in-depth empirical study. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (p. 511-522). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240657&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00055
- V. Tawosi, R. M., A. Al-Subaihin, & Sarro, F. (2022). A versatile dataset of agile open source software projects. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796320&isnumber=9796155> doi: 10.1145/3524842.3528029
- W. Amme, T. S. H., & Schäfer, A. (2021). You look so different: Finding structural clones and subclones in java source code. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609181&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00013
- Wan, Y., Zhao, W., Zhang, H., Sui, Y., Xu, G., & Jin, H. (2022). What do they capture? - a structural analysis of pre-trained language models for source code. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794032&isnumber=9793541> doi: 10.1145/3510003.3510050
- Wang, D., Jia, Z., Li, S., Yu, Y., Xiong, Y., Dong, W., & Liao, X. (2022). Bridging pre-trained models and downstream tasks for source code understanding. In *2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE)* (p. 287-298). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793959&isnumber=9793541> doi: 10.1145/3510003.3510062
- Wang, H., & Kagdi, H. (2018). A conceptual replication study on bugs that get fixed in open source software. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530038&isnumber=8529823> doi: 10.1109/ICSME.2018.00039
- Wen, F., Aghajani, E., Nagy, C., Lanza, M., & Bavota, G. (2021). Siri, write the next method. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402018&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00025

- Wen, F., Ferrari, V., Aghajani, E., Nagy, C., Lanza, M., & Bavota, G. (2021). Fears: Recommending complete android method implementations. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609117&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00062
- Wessel, M., Abdellatif, A., Wiese, I., Conte, T., Shihab, E., Gerosa, M. A., & Steinmacher, I. (2022). Bots for pull requests: The good, the bad, and the promising. In *2022 ieee/acm 44th international conference on software engineering (icse)* (p. 274-286). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793907&isnumber=9793541> doi: 10.1145/3510003.3512765
- Wilkie, J., Halabi, Z. A., Karaoglu, A., Liao, J., Ndungu, G., Ragkhitwetsagul, C., ... Krinke, J. (2018). Who's this? developer identification using ide event data. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196461> doi: 10.1145/3196398.3196461
- W. Kopec, R. N. G. K. A. W., B. Balcerzak, & Casati, F. (2018). Older adults and hackathons: A qualitative study. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453141&isnumber=8453044> doi: 10.1145/3180155.3182547
- W. Luo, X. R. J. W. C. F., D. Chai, & Chen, Z. (2021). Graph-based fuzz testing for deep learning inference engines. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9401995&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00037
- W. Xiao, W. X. X. T. J. D., H. He, & Zhou, M. (2022). Recommending good first issues in github oss projects. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793998&isnumber=9793541> doi: 10.1145/3510003.3510196
- W. Yuan, H. S., Y. Xiong, & Liu, X. (2021). Incorporating multiple features to predict bug fixing time with neural networks. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609133&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00015

- W. Zhou, G. Z., Y. Zhao, & Shen, X. (2020). Harp: Holistic analysis for refactoring python-based analytics programs. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284016&isnumber=9283910> doi: nan
- X. Hu, D. L. Z. W. Q. C., X. Xia, & Zimmermann, T. (2022). Practitioners' expectations on automated code comment generation. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793950&isnumber=9793541> doi: 10.1145/3510003.3510152
- X. Huang, X. Z. M. A. B., H. Zhang, & Yang, S. (2018). Synthesizing qualitative research in software engineering: A critical review. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453202&isnumber=8453044> doi: 10.1145/3180155.3180235
- Xia, H., Zhang, Y., Zhou, Y., Chen, X., Wang, Y., Zhang, X., . . . Yang, Z. (2020). How android developers handle evolution-induced api compatibility issues: A large-scale study. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)* (p. 886-898). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284041&isnumber=9283910>
- X. Li, M. d., S. Zhu, & Orso, A. (2018). Enlightened debugging. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453065&isnumber=8453044> doi: 10.1145/3180155.3180242
- X. Liu, C. L., L. Huang, & Ng, V. (2018). Linking source code to untangled change intents. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530046&isnumber=8529823> doi: 10.1109/ICSME.2018.00047
- X. Xia, D. L. Z. X. A. E. H., L. Bao, & Li, S. (2018). Measuring program comprehension: A large-scale field study with professionals. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8453126&isnumber=8453044> doi: 10.1145/3180155.3182538
- X. Xia, P. S. K., Z. Wan, & Lo, D. (2019). How practitioners perceive coding proficiency. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8812050&isnumber=8811891> doi: 10.1109/ICSE.2019.00098

- X. Xu, Z. Y. M. F. A. J., Q. Zheng, & Liu, T. (2021). Interpretation-enabled software reuse detection based on a multi-level birthmark model. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402081&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00084
- X. Zhou, D. H., & Lo, D. (2021). Assessing generalizability of codebert. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9609166&isnumber=9609104> doi: 10.1109/ICSME52107.2021.00044
- Yamamoto, K., Kondo, M., Nishiura, K., & Mizuno, O. (2020). Which metrics should researchers use to collect repositories: An empirical study. In *2020 IEEE 20th international conference on software quality, reliability and security (QRS)* (p. 458-466). doi: 10.1109/QRS51102.2020.00065
- Y. Huang, D. F., & Zimmermann, T. (2021). Leaving my fingerprints: Motivations and challenges of contributing to oss for social good. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402112&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00096
- Y. Li, S. W., & Nguyen, T. N. (2022). Dear: A novel deep learning-based approach for automated program repair. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793900&isnumber=9793541> doi: 10.1145/3510003.3510177
- Y. Lu, M. Z. Y. Z. T. W., X. Mao, & Li, Z. (2020). Haste makes waste: An empirical study of fast answers in stack overflow. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240678&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00013
- Y. Tang, M. B. R. S. A. S., R. Khatchadourian, & Raja, A. (2021). An empirical study of refactorings and technical debt in machine learning systems. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9401990&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00033
- Y. Tian, K. J. S. L. J., Y. Zhang, & Liu, H. (2022). What makes a good commit message? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9794134&isnumber=9793541> doi: 10.1145/3510003.3510205

- Yu, H., Lou, Y., Sun, K., Ran, D., Xie, T., Hao, D., ... Wang, Q. (2022). Automated assertion generation via information retrieval and its integration with deep learning. In *2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE)* (p. 163-174). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793891&isnumber=9793541> doi: 10.1145/3510003.3510149
- Y. Wang, H. Z. X. M. L. S., J. Wang, & Wang, Q. (2022). Where is your app frustrating users? In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793551&isnumber=9793541> doi: nan
- Zacchiroli, S. (2022). A large-scale dataset of (open source) license text variants. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9796220&isnumber=9796155> doi: 10.1145/3524842.3528491
- Zampetti, F., Serebrenik, A., & Di Penta, M. (2018). Was self-admitted technical debt removal a real removal? an in-depth perspective. In *Msr*. Retrieved from <https://doi.org/10.1145/3196398.3196423> doi: 10.1145/3196398.3196423
- Zhang, J. M., & Harman, M. (2021). Ignorance and prejudice” in software fairness. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402057&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00129
- Zhang, W. (2020). Efficient bug triage for industrial environments. In *Icsme*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240673&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00082
- Zhang, X., Chen, Y., Gu, Y., Zou, W., Xie, X., Jia, X., & Xuan, J. (2018). How do multiple pull requests change the same code: A study of competing pull requests in github. In *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (p. 228-239). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8530032&isnumber=8529823> doi: 10.1109/ICSME.2018.00032
- Zheng, Y., Liu, Y., Xie, X., Liu, Y., Ma, L., Hao, J., & Liu, Y. (2021). Automatic web testing using curiosity-driven reinforcement learning. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (p. 423-435). Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9402046&isnumber=9401950> doi: 10.1109/ICSE43902.2021.00048

- Zhu, W., & Godfrey, M. W. (2021). Mea culpa: How developers fix their own simple bugs differently from other developers. In *Msr*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9463125&isnumber=9463063> doi: 10.1109/MSR52588.2021.00065
- Zieris, F., & Prechelt, L. (2020). Explaining pair programming session dynamics from knowledge gaps. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9284068&isnumber=9283910> doi: nan
- Z. Li, C. C. Y. Z., G. Q. Chen, & Xu, S. (2022). Ropgen: Towards robust code authorship attribution via automatic coding style transformation. In *Icse*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9793924&isnumber=9793541> doi: 10.1145/3510003.3510181
- Zyrianov, V., Guarnera, D. T., Peterson, C. S., Sharif, B., & Maletic, J. I. (2020). Automated recording and semantics-aware replaying of high-speed eye tracking and interaction data to support cognitive studies of software engineering tasks. In *Icsmc*. Retrieved from <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9240624&isnumber=9240598> doi: 10.1109/ICSME46990.2020.00051