```
# Aidan Potts
# Thesis Research

# Load packages ----
library(pacman)
p_load(tidyverse, janitor, readxl, lubridate, estimatr, Synth, scales,
stats, lmtest)


# Load data ----
OR_EV_df = read_excel("~/Desktop/UO/Honors College/Thesis/data/10-25-2021-
ODOT-EV-Data-All-Attributes-Public.xlsx")
CA_EV_df = read_csv("~/Desktop/UO/Honors College/Thesis/data/
ca_ev_registrations_public.csv")
CO_EV_df = read_csv("~/Desktop/UO/Honors College/Thesis/data/
co_ev_registrations_public.csv")
NY_EV_df = read_csv("~/Desktop/UO/Honors College/Thesis/data/
ny_ev_registrations_public.csv")
TX_EV_df = read_csv("~/Desktop/UO/Honors College/Thesis/data/
tx_ev_registrations_public.csv")
VT_EV_df = read_csv("~/Desktop/UO/Honors College/Thesis/data/
vt_ev_registrations_public.csv")
charging_df = read_csv("~/Desktop/UO/Honors College/Thesis/data/
alt_fuel_stations (Feb 8 2022).csv")


# Clean variable names ----
OR_EV_df = OR_EV_df %>% clean_names()
CA_EV_df = CA_EV_df %>% clean_names()
CO_EV_df = CO_EV_df %>% clean_names()
NY_EV_df = NY_EV_df %>% clean_names()
TX_EV_df = TX_EV_df %>% clean_names()
VT_EV_df = VT_EV_df %>% clean_names()
charging_df = charging_df %>% clean_names()

# Data cleaning: State-specific dates ----
# Add 'date' column for OR
OR_EV_df = OR_EV_df %>% mutate(date = first_reg %>% mdy())

# Convert 'registration_valid_date' to date from string
VT_EV_df = VT_EV_df %>% mutate(date = registration_valid_date %>% mdy())

# Add 'year'
OR_EV_df = OR_EV_df %>% mutate(yr = year(date))
CA_EV_df = CA_EV_df %>% transmute(yr = year(registration_valid_date))
NY_EV_df = NY_EV_df %>% transmute(yr = year(registration_valid_date))
CO_EV_df = CO_EV_df %>% transmute(yr = year(registration_valid_date))
VT_EV_df = VT_EV_df %>% mutate(yr = year(date))

# Collapse to the year
or_df = OR_EV_df %>% group_by(yr) %>% summarize(n_ev = n())
ca_df = CA_EV_df %>% group_by(yr) %>% summarize(n_ev = n())
ny_df = NY_EV_df %>% group_by(yr) %>% summarize(n_ev = n())
co_df = CO_EV_df %>% group_by(yr) %>% summarize(n_ev = n())
vt_df = VT_EV_df %>% group_by(yr) %>% summarize(n_ev = n())

# Add the state's name
or_df = or_df %>% mutate(st = 'OR')
```

```
ca_df = ca_df %>% mutate(st = 'CA')
ny_df = ny_df %>% mutate(st = 'NY')
co_df = co_df %>% mutate(st = 'CO')
vt_df = vt_df %>% mutate(st = 'VT')

# Re-order the columns
or_df = or_df %>% select(yr, st, n_ev)
ca_df = ca_df %>% select(yr, st, n_ev)
ny_df = ny_df %>% select(yr, st, n_ev)
co_df = co_df %>% select(yr, st, n_ev)
vt_df = vt_df %>% select(yr, st, n_ev)

# Bind the dataframes together and add missing years
ev_df = bind_rows(or_df, ca_df, ny_df, co_df, vt_df)
ev_df = ev_df %>% filter(yr %in% 2010:2019) # note: some states don't have
complete data in 2020, so my analysis
ev_df = complete(ev_df, st, yr, fill = list(n_ev = 0)) # note continued:
will go through 2019.

# Data cleaning: adding EV chargers ----

# Cleaning and filtering charging_df
charging_df = charging_df %>% mutate(yr = year(open_date)) %>%
group_by(state, yr) %>% summarize(n_chargers = n())
chargers = filter(charging_df, state %in% c('OR', 'CA', 'NY', 'CO', 'VT'))

# Change the name of the variable 'state' and 'open_date'
chargers = chargers %>% rename(st = state)

# Fill in 0s
chargers = complete(chargers, st, yr, fill = list(n_chargers = 0))

# Merge
full_df = left_join(x = ev_df, y = chargers, by = c('st', 'yr'))

# Fill NAs with 0s
full_df = replace_na(full_df, replace = list(n_chargers = 0))

# Adding in population ----
year_pop = c(32270000, 37640000, 37950000, 38260000, 38600000, 38920000,
39170000, 39360000, 39460000, 39510000, # California 2010-2019
             5095000, 5121000, 5193000, 5269000, 5350000, 5451000,
5539000, 5612000, 5691000, 5759000, # Colorado 2010-2019
             19580000, 19500000, 19570000, 19620000, 19650000, 19650000,
19630000, 19590000, 19530000, 19450000, # New York 2010-2019
             3856000, 3872000, 3899000, 3922000, 3963000, 4016000,
4090000, 4144000, 4182000, 4218000,  # Oregon 2010-2019
             622433, 627049, 626090, 626210, 625214, 625216, 6236557,
624344, 624358, 623989) # Vermont 2010-2019
full_df$year_pop <- year_pop
full_df = mutate(full_df, n_ev_pc = n_ev/year_pop, n_chargers_pc =
n_chargers/year_pop)




# Graphing and Regressions ----
# Graph data
```

```
# Number of chargers per capita
ggplot(data = full_df, aes(x = yr, y = n_chargers_pc, color = st)) +
  geom_line() +
  geom_point() +
  ggtitle("Number of New Chargers Installed By Year",
          subtitle = "Per Capita") +
  scale_x_continuous('Year', breaks = seq(2005, 2020, 2)) +
  scale_y_continuous('Number of chargers per capita',
                     labels = scales::comma) +
  scale_color_viridis_d('State') +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5), plot.subtitle =
element_text(hjust = 0.5))

# Number of EVs per capita
ggplot(data = full_df, aes(x = yr, y = n_ev_pc, color = st)) +
  geom_line() +
  geom_point() +
  ggtitle("Number of New EVs Purchased and Registered By Year",
          subtitle = "Per Capita") +
  scale_x_continuous('Year', breaks = seq(2005, 2020, 2)) +
  scale_y_continuous('Number of EVs per capita',
                     labels = scales::comma) +
  scale_color_viridis_d('State') +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5), plot.subtitle =
element_text(hjust = 0.5))

# Number of new EV registrations (by year)
ggplot(data = full_df, aes(x = yr, y = n_ev, color = st)) +
  geom_line() +
  geom_point() +
  ggtitle("Number of New EVs Purchased and Registered By Year") +
  scale_x_continuous('Year', breaks = seq(2005, 2020, 2)) +
  scale_y_continuous('Number of EVs',
                     labels = scales::comma) +
  scale_color_viridis_d('State') +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

# Number of new chargers (by year)
ggplot(data = full_df, aes(x = yr, y = n_chargers, color = st)) +
  geom_line() +
  geom_point() +
  ggtitle("Number of New Chargers Installed By Year") +
  scale_x_continuous('Year', breaks = seq(2005, 2020, 2)) +
  scale_y_continuous('Number of chargers',
                     labels = scales::comma) +
  scale_color_viridis_d('State') +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))



standard_reg = lm(n_ev ~ n_chargers + lag(n_chargers), data = full_df, st
== "OR")
summary(standard_reg)
```

```
lmtest::dwtest(standard_reg)

lagged_standard_fixed = lm(n_ev ~ n_chargers + lag(n_chargers) +
factor(yr), data = full_df, st == "OR")
summary(lagged_standard_fixed)
lmtest::dwtest(lagged_standard_fixed)

lagged_pc = lm(n_ev_pc ~ lag(n_chargers_pc) + n_chargers_pc, data =
full_df, factor(yr))
summary(lagged_pc)

lagged_st_yr = lm(n_ev ~ n_chargers + lag(n_chargers) + st + yr, data =
full_df)
summary(lagged_st_yr)


ggplot(data = ev_df, aes(x = yr, n_ev/year_pop, color = st)) +
  geom_point() + geom_line() + theme_minimal()
```

```
# Aidan Potts
# Thesis Research (Synthetic Control analysis)

# Data conversion to be synth-compatible
tmp = full_df %>% mutate(
  n_ev_pc = n_ev/year_pop,
  n_chargers_pc = n_chargers/year_pop,
  st_num = st %>% factor() %>% as.numeric()
) %>% as.data.frame()

synth_prep = dataprep(
  foo = tmp,
  predictors = 'n_chargers_pc',
  dependent = 'n_ev_pc',
  unit.variable = 'st_num',
  time.variable = 'yr',
  treatment.identifier = 'OR',
  unit.names.variable = 'st',
  controls.identifier = c('CA', 'NY', 'CO'),
  time.predictors.prior = c(2010:2016),
  time.optimize.ssr = c(2010:2016),
  time.plot = 2010:2018
)

synth_out = synth(synth_prep)

path.plot(synth.res = synth_out,
          dataprep.res = synth_prep,
          Ylab = c("EVs Per Capita"),
          Xlab = c("Year"),
          Legend = c("Oregon", "Synthetic Oregon"))
abline(v = 2016, lty = 2)

gaps.plot(synth.res = synth_out,
          dataprep.res = synth_prep,
          Ylab = c("Gap Between Treatment and Synthetic Control"),
          Xlab = c("Year"))
abline(v = 2016, lty = 2)
```