Farnsworth Knitted Thesis Code

2025-05-03

```r
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(quadprog)
library(ggplot2)
library(knitr)
library(kableExtra)
```

```
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
```

```
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
library(showtext)
```

```
## Loading required package: sysfonts

## Loading required package: showtextdb
```

```r
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(sysfonts)
library(future)
library(future.apply)
library(patchwork)
library(lmtest)
library(sandwich)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
```

```
##     combine
```

```r
library(grid)
showtext_auto()
font_add_google("Merriweather", "latexfont")
df <- read_xlsx("~/Downloads/FarnsworthCrashData (1).xlsx")
```

```r
# CALCULATE ATT

# Format columns
df <- df %>%
  rename_all(tolower) %>%
  mutate(
    month_num = match(tolower(month), tolower(month.abb)),
    year = suppressWarnings(as.numeric(year))) %>%
  filter(!is.na(year), !is.na(month_num)) %>%
  mutate(
    date = make_date(year, month_num, 1),
    crash_rate = crashes / vmt)

# Set pre- and post- periods
policy_start <- as.Date("2021-02-01")
df_pre <- df %>% filter(date < policy_start)
df_post <- df %>% filter(date >= policy_start)

# Pivot wide in the pre-period to prepare for the matrix minimization problem
df_pre_wide <- df_pre %>%
  select(date, state, crash_rate) %>%
  pivot_wider(names_from = state, values_from = crash_rate, values_fn = sum)

# Set treated versus untreated states
treated_state <- "Oregon"
all_states <- setdiff(colnames(df_pre_wide), "date")
donor_states <- setdiff(all_states, treated_state)

# Set y_o as the outcome vector for the treated unit in the pre-period
y_o <- df_pre_wide[[treated_state]]

# Set Y_d as the matrix of outcome vectors for donor states in the pre-period
Y_d <- df_pre_wide[, donor_states] %>% as.matrix()

# Define function for synthetic control
scm_fit <- function(y, Y) {
  Dmat <- 2 * t(Y) %*% Y
  dvec <- 2 * t(Y) %*% y
  Amat <- cbind(rep(1, ncol(Y)), diag(ncol(Y)))
  bvec <- c(1, rep(0, ncol(Y)))
  meq <- 1
  sol <- solve.QP(Dmat, dvec, Amat, bvec, meq = meq)
  sol$solution}

# Minimize the squared error between Oregon and the donor states in the pre-period
w_hat <- scm_fit(y_o, Y_d)

# Display donor weights
```

```r
weights_df <- data.frame(donor = donor_states,
  weight = round(w_hat, 4)) %>%
  arrange(desc(weight))
  kable(weights_df, caption = "Donor Weights") %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

```
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
```

```
## See help("Deprecated")
```

```
## Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

Table 1: Donor Weights

donor	weight
Washington	0.4805
California	0.2593
Wyoming	0.1263
Idaho	0.0660
Colorado	0.0471
Utah	0.0208
Arizona	0.0000
Wisconsin	0.0000
North Carolina	0.0000
South Dakota	0.0000

```
# Create synthetic Oregon from donor states and flag post-policy months
df_wide_all <- df %>%
  select(date, state, crash_rate) %>%
  pivot_wider(names_from = state, values_from = crash_rate, values_fn = sum) %>%
  mutate(synthetic = rowSums(across(all_of(donor_states), ~ .x * w_hat[which(donor_states == cur_column
    post = date >= ymd("2020-11-01"))

# Apply scalar correction to match Oregon's level
c <- mean(df_wide_all$Oregon[!df_wide_all$post], na.rm = TRUE) /
     mean(df_wide_all$synthetic[!df_wide_all$post], na.rm = TRUE)

# Caclulate the estimated difference between Oregon and synthetic control at each time point
df_wide_all <- df_wide_all %>%
  mutate(
    synthetic_scaled = synthetic * c,
    gap = Oregon - synthetic_scaled)

# Compute average treatment gap before and after policy
att_summary <- df_wide_all %>%
  group_by(post) %>%
  summarise(avg_gap = mean(gap, na.rm = TRUE), .groups = "drop")

# Calculate ATT as the post-period average gap minus pre-period average gap
att <- diff(att_summary$avg_gap)
print(att)
```

```
## [1] -0.07996006
```

```
# Calculate pre-treatment MSPE
pre_mspe <- df_wide_all %>%
  filter(!post) %>%
  summarise(
    mspe = mean((Oregon - synthetic_scaled)^2, na.rm = TRUE),
    rmse = sqrt(mspe),
```

```r
    mae = mean(abs(Oregon - synthetic_scaled), na.rm = TRUE))

# Create a table to display the pre-treatment fit metrics
fit_metrics_table <- tibble(
  Metric = c("Mean Squared Prediction Error (MSPE)",
             "Root Mean Squared Error (RMSE)",
             "Mean Absolute Error (MAE)"),
  Value = c(round(pre_mspe$mspe, 6), round(pre_mspe$rmse, 6), round(pre_mspe$mae, 6)))

kable(fit_metrics_table, caption = "Pre-Treatment Fit Quality Metrics") %>%
kable_styling(latex_options = c("striped", "hold_position"))
```

```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

```
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
```

```
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

Table 2: Pre-Treatment Fit Quality Metrics

Metric	Value
Mean Squared Prediction Error (MSPE)	0.010199
Root Mean Squared Error (RMSE)	0.100990
Mean Absolute Error (MAE)	0.081102

```r
# PLACEBO TESTS

# Create storage for placebo ATT estimates
placebo_att <- c()

# Loop through each donor state and treat it as if it were "treated"
for (state in donor_states) {

# Construct synthetic control for placebo state
y_placebo <- df_pre_wide[[state]]
Y_d_placebo <- df_pre_wide[, setdiff(donor_states, state)] %>% as.matrix()

# Estimate weights using earlier function
w_placebo <- tryCatch({
    scm_fit(y_placebo, Y_d_placebo)
  }, error = function(e) return(NULL))

# Create synthetic control for placebo-treated state
if (!is.null(w_placebo)) {
  synth_placebo <- as.matrix(df_wide_all[, setdiff(donor_states, state)]) %*% w_placebo
  gap_placebo <- df_wide_all[[state]] - synth_placebo
  post_period <- df_wide_all$date >= as.Date("2021-02-01")

# Compute placebo ATT and store it
  placebo_ATT <- mean(gap_placebo[post_period], na.rm = TRUE)
  placebo_att <- c(placebo_att, placebo_ATT)}}

# Compute p-value by calculating share of placebo ATTs as extreme as Oregon's
placebo_p_value <- mean(abs(placebo_att) >= abs(att))

# Compute standard error using standard deviation of placebo ATTs
se_placebo <- sd(placebo_att, na.rm = TRUE)

print(placebo_p_value)
```

```
## [1] 0.4
```

```r
print(se_placebo)
```

```
## [1] 0.3554559
```

```r
# Compute pre- and post-treatment MSPEs for Oregon
oregon_pre_mspe <- mean((df_wide_all$Oregon[!df_wide_all$post] -
                        df_wide_all$synthetic_scaled[!df_wide_all$post])^2, na.rm = TRUE)
oregon_post_mspe <- mean((df_wide_all$Oregon[df_wide_all$post] -
                         df_wide_all$synthetic_scaled[df_wide_all$post])^2, na.rm = TRUE)
oregon_mspe_ratio <- oregon_post_mspe / oregon_pre_mspe

# Calculate MSPE ratios for all placebo states
placebo_mspe_ratios <- c()
placebo_pre_mspes <- c()
state_names <- c()

# Repeat synthetic control estimation for each placebo
for (state in donor_states) {
  y_placebo <- df_pre_wide[[state]]
  Y_d_placebo <- df_pre_wide[, setdiff(donor_states, state)] %>% as.matrix()

  w_placebo <- tryCatch({
    scm_fit(y_placebo, Y_d_placebo)
  }, error = function(e) return(NULL))

  if (!is.null(w_placebo)) {
    synth_placebo <- as.matrix(df_wide_all[, setdiff(donor_states, state)]) %*% w_placebo

# Compute pre- and post-MSPEs
    pre_mspe <- mean((df_wide_all[[state]][!df_wide_all$post] -
                     synth_placebo[!df_wide_all$post])^2, na.rm = TRUE)
    post_mspe <- mean((df_wide_all[[state]][df_wide_all$post] -
                      synth_placebo[df_wide_all$post])^2, na.rm = TRUE)

# Store MSPE ratio if valid
    if (pre_mspe > 0) {
      mspe_ratio <- post_mspe / pre_mspe
      placebo_mspe_ratios <- c(placebo_mspe_ratios, mspe_ratio)
      placebo_pre_mspes <- c(placebo_pre_mspes, pre_mspe)
      state_names <- c(state_names, state)}}}

# Create MSPE ratio comparison table
mspe_ratio_df <- data.frame(
  State = c("Oregon", state_names),
  Pre_MSPE = c(oregon_pre_mspe, placebo_pre_mspes),
  MSPE_Ratio = c(oregon_mspe_ratio, placebo_mspe_ratios)) %>%
  arrange(desc(MSPE_Ratio))

# Display MSPE ratio table and Oregon's implied p-value
kable(mspe_ratio_df, caption = "MSPE Ratio Comparison (Post/Pre)") %>%
kable_styling(latex_options = c("striped", "hold_position"))
```

```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
```

```
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

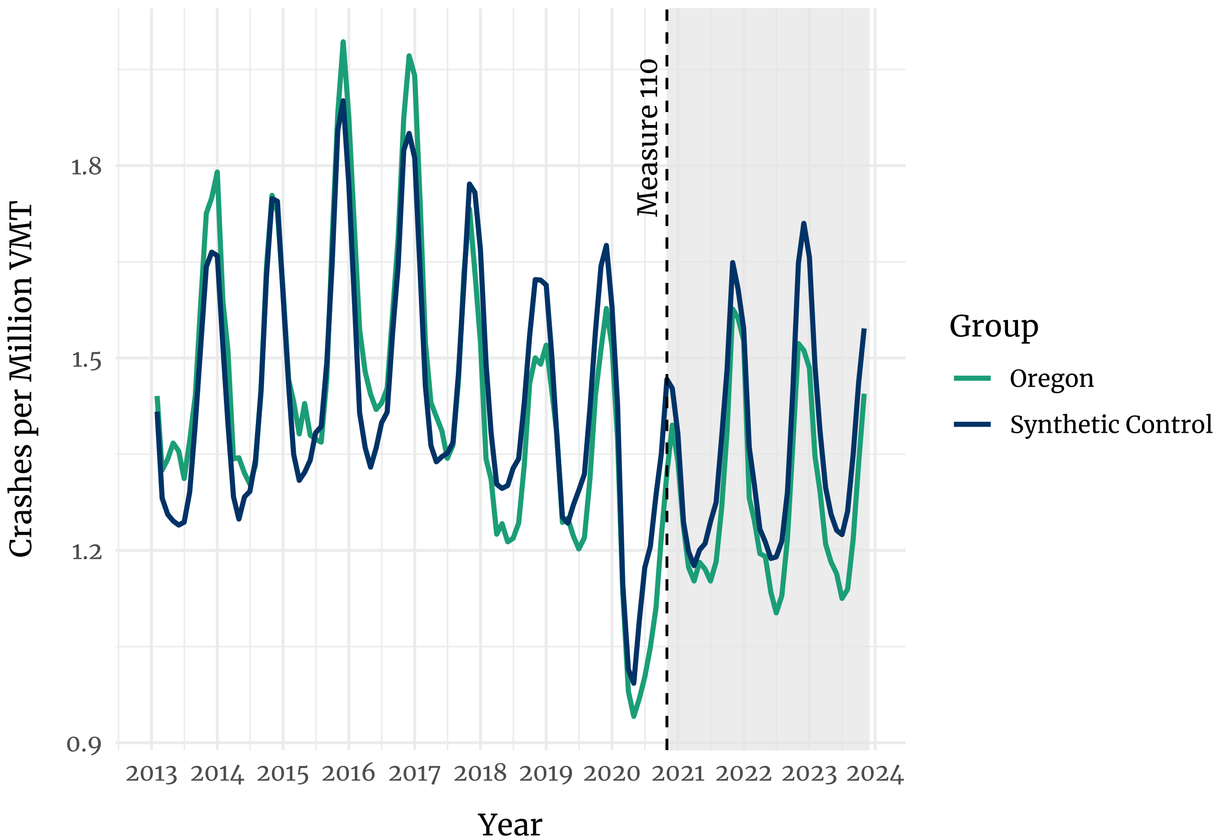
Table 3: MSPE Ratio Comparison (Post/Pre)

State	Pre_MSPE	MSPE_Ratio
North Carolina	0.0526758	4.7819119
Colorado	0.0633336	1.9632051
Wyoming	0.0223791	1.5847923
Oregon	0.0101989	1.0263402
California	0.9557949	1.0042035
Wisconsin	0.0431936	0.9798528
Washington	0.0348025	0.8948216
Idaho	0.0397932	0.8627429
South Dakota	0.0528337	0.8326771
Utah	0.0266501	0.7714646
Arizona	0.0196941	0.6356507

```r
# CREATE PLOT

# Prepare plot data by selecting Oregon and synthetic control crash rates, reshaping to long format
df_plot <- df_wide_all %>%
  select(date, Oregon, synthetic_scaled) %>%
  pivot_longer(cols = c("Oregon", "synthetic_scaled"), names_to = "series", values_to = "crash_rate") %
  filter(!is.na(crash_rate)) %>%
  group_by(series) %>%
  mutate(smoothed_rate = zoo::rollmean(crash_rate, k = 3, fill = NA, align = "center")) %>%
  ungroup() %>%
  mutate(series = recode(series, synthetic_scaled = "Synthetic Control"))

# Plot crash rates over time with treatment date and shaded post-treatment region
ggplot(df_plot, aes(x = date, y = smoothed_rate, color = series)) +
  annotate("rect", xmin = as.Date("2020-11-01"), xmax = max(df_plot$date),
           ymin = -Inf, ymax = Inf, fill = "gray90", alpha = 0.7) +
  geom_line(size = 0.9) +
  geom_vline(xintercept = as.Date("2020-11-01"), linetype = "dashed") +
  annotate("text", x = as.Date("2020-11-01"),
           y = max(df_plot$smoothed_rate, na.rm = TRUE),
           label = "Measure 110", angle = 90, vjust = -0.5, hjust = 1.1,
           size = 3.5, family = "latexfont") +
  scale_y_continuous(name = "Crashes per Million VMT") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_color_manual(values = c("Oregon" = "#1b9e77", "Synthetic Control" = "#003366")) +
  labs(x = "Year", color = "Group") +
  theme_minimal(base_family = "latexfont") +
  theme(
    axis.title.x = element_text(margin = margin(t = 10)),
    axis.title.y = element_text(margin = margin(r = 10)),
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold")
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
```

```
## generated.

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```r
# Save file to folder
ggsave("images/synthdid_plot_scaled.pdf", width = 8, height = 4, dpi = 300)
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```r
# SENSITIVITY ANALYSIS

# Define donor subsets
subset_list <- list(
  Neighbors          = c("Washington", "Idaho", "California"),
  Exclude_Washington = setdiff(donor_states, "Washington"))

# Prepare storage
subset_results <- list()
weights_by_subset <- list()
se_by_subset <- list()
pval_by_subset <- list()

# Loop over each donor subset
for (label in names(subset_list)) {
  donors_subset <- subset_list[[label]]
  y_oregon <- df_pre_wide$Oregon
  Y_subset <- df_pre_wide[, donors_subset] %>% as.matrix()
```

```r
  w_subset <- tryCatch(scm_fit(y_oregon, Y_subset), error = function(e) NULL)
  if (is.null(w_subset)) next

# Construct synthetic series and gap
  synth_all   <- as.matrix(df_wide_all[, donors_subset]) %*% w_subset
  gap         <- df_wide_all$Oregon - synth_all
  post_period <- df_wide_all$date >= as.Date("2020-11-01")

# Compute raw ATT
  ATT_raw <- mean(gap[post_period], na.rm = TRUE)

# compute placebo ATTs for the subset
  placebo_ATT_subset <- c()
  for (pseudo_treated in donors_subset) {
    Y_pseudo <- df_pre_wide[, setdiff(donors_subset, pseudo_treated)] %>% as.matrix()
    w_pseudo <- tryCatch(scm_fit(df_pre_wide[[pseudo_treated]], Y_pseudo),
                         error = function(e) NULL)
    if (is.null(w_pseudo)) next
    synth_pseudo <- as.matrix(df_wide_all[, setdiff(donors_subset, pseudo_treated)]) %*% w_pseudo
    gap_pseudo   <- df_wide_all[[pseudo_treated]] - synth_pseudo
    placebo_ATT_subset <- c(placebo_ATT_subset,
                            mean(gap_pseudo[post_period], na.rm = TRUE))}

# Correct ATT bias
  bias_subset <- mean(placebo_ATT_subset, na.rm = TRUE)
  ATT_corrected_subset <- ATT_raw - bias_subset
  se_corrected_subset <- sd(placebo_ATT_subset - bias_subset, na.rm = TRUE)
  pval_corrected_subset <- mean(
    abs(placebo_ATT_subset - bias_subset) >= abs(ATT_corrected_subset),
    na.rm = TRUE)

# Store results
  subset_results[[label]] <- ATT_corrected_subset
  weights_by_subset[[label]] <- setNames(round(w_subset, 4), donors_subset)
  se_by_subset[[label]] <- se_corrected_subset
  pval_by_subset[[label]] <- pval_corrected_subset}

# Combine into tables
subset_df <- tibble(
  Subset           = names(subset_results),
  ATT_Estimate     = unlist(subset_results),
  Std_Error        = unlist(se_by_subset),
  Empirical_p_value= unlist(pval_by_subset))

weight_long <- bind_rows(
  lapply(names(weights_by_subset), function(label) {
    tibble(
      Subset = label,
      Donor  = names(weights_by_subset[[label]]),
      Weight = weights_by_subset[[label]])}))

# Display
kable(subset_df, caption = "Bias-Corrected Sensitivity Analysis by Donor Subset") %>%
```

```r
  kable_styling("striped", "hold_position")
```

```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
```

```
## See help("Deprecated")
```

Table 4: Bias-Corrected Sensitivity Analysis by Donor Subset

Subset	ATT_Estimate	Std_Error	Empirical_p_value
Neighbors	0.0978369	0.6871422	1
Exclude_Washington	0.0120047	0.3828464	1

```
kable(weight_long, caption = "Donor Weights by Subset") %>%
  kable_styling("striped", "hold_position")
```

```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

```
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

```
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

Table 5: Donor Weights by Subset

Subset	Donor	Weight
Neighbors	Washington	0.6149
Neighbors	Idaho	0.1131
Neighbors	California	0.2721
Exclude_Washington	Idaho	0.2619
Exclude_Washington	Utah	0.0000
Exclude_Washington	Colorado	0.0867
Exclude_Washington	California	0.1499
Exclude_Washington	Arizona	0.2182
Exclude_Washington	Wisconsin	0.0000
Exclude_Washington	North Carolina	0.0000
Exclude_Washington	Wyoming	0.2833
Exclude_Washington	South Dakota	0.0000

```
# PSEUDO-TREATMENT TEST

# Set cutoff for pseudo-treatment dates after 2015 so we have at least two years to create a synthetic
pseudo_periods <- df_pre$date[df_pre$date >= as.Date("2015-01-01")]
n_placebo_draws <- 50

# Loop over each pseudo-treatment date and estimate synthetic control gaps
pseudo_results_nested <- lapply(pseudo_periods, function(pseudo_date) {

# Filter training data before the pseudo-treatment date
  df_train <- df_pre %>% filter(date < pseudo_date)

# Pivot training data to wide format
  df_train_wide <- df_train %>%
    select(date, state, crash_rate) %>%
    pivot_wider(names_from = state, values_from = crash_rate, values_fn = sum)
  if (nrow(df_train_wide) <= 2) {
    return(list(ATT = NA, SE = NA, p_value = NA))}

# Construct synthetic Oregon from pre-pseudo data
  y_oregon <- df_train_wide$Oregon
  Y_d_oregon <- df_train_wide[, donor_states] %>% as.matrix()

# Estimate weights using earlier function
  w_hat_oregon <- tryCatch({
```

```r
    scm_fit(y_oregon, Y_d_oregon)
  }, error = function(e) NULL)

# Transform dataset to apply synthetic control across full period
  df_all_wide <- df_pre %>%
    select(date, state, crash_rate) %>%
    pivot_wider(names_from = state, values_from = crash_rate, values_fn = sum)

# Create synthetic control before pseudo-treatment date
  synthetic_series_oregon <- rowSums(as.matrix(df_all_wide[, donor_states]) *
                                  matrix(rep(w_hat_oregon, each = nrow(df_all_wide)), ncol = length(do

# Scale so fake synthetic control matches fake treated unit
scalar_oregon <- mean(df_all_wide$Oregon[df_all_wide$date < pseudo_date], na.rm = TRUE) /
                 mean(synthetic_series_oregon[df_all_wide$date < pseudo_date], na.rm = TRUE)
synthetic_series_oregon_scaled <- synthetic_series_oregon * scalar_oregon

# Compute the gap between actual and synthetic Oregon
df_all_wide <- df_all_wide %>%
  mutate(
    synthetic_pseudo = synthetic_series_oregon_scaled,
    gap_pseudo = Oregon - synthetic_pseudo,
    post_pseudo = date >= pseudo_date)

# Compute pseudo ATT
  att_summary_pseudo <- df_all_wide %>%
    group_by(post_pseudo) %>%
    summarise(
      avg_gap = mean(gap_pseudo, na.rm = TRUE),
      .groups = "drop")
  if (nrow(att_summary_pseudo) != 2) {
    return(list(ATT = NA, SE = NA, p_value = NA))}

  att_oregon <- diff(att_summary_pseudo$avg_gap)

# Calculate SEs for pseudo treatment estimates using placebo-based inference

# Create empty vector to store placebo ATTs
  placebo_att_list <- c()

# Loop over each donor state
  for (draw in 1:n_placebo_draws) {
    fake_treated <- sample(donor_states, 1)

# Construct synthetic control for placebo state
    Y_d_fake <- df_train_wide[, setdiff(donor_states, fake_treated)] %>% as.matrix()
    y_fake <- df_train_wide[[fake_treated]]

# Estimate weights using earlier function
  w_hat_fake <- tryCatch({
    scm_fit(y_fake, Y_d_fake)
  }, error = function(e) NULL)
```

```r
# Create synthetic control for placebo-treated state
  synthetic_series_fake <- rowSums(as.matrix(df_all_wide[, setdiff(donor_states, fake_treated)]) *
    matrix(rep(w_hat_fake, each = nrow(df_all_wide)), ncol = length(setdiff(donor_states, fake_treated)

# Scale each placebo's synthetic control to match its own pre-period mean
  post_fake <- df_all_wide$date >= pseudo_date
  scalar_fake <- mean(df_all_wide[[fake_treated]][!post_fake], na.rm = TRUE) /
                 mean(synthetic_series_fake[!post_fake], na.rm = TRUE)
  synthetic_series_fake_scaled <- synthetic_series_fake * scalar_fake

# Compute and store placebo ATT for this draw
  gap_fake <- df_all_wide[[fake_treated]] - synthetic_series_fake_scaled
  placebo_ATT <- mean(gap_fake[post_fake], na.rm = TRUE) - mean(gap_fake[!post_fake], na.rm = TRUE)
  placebo_att_list <- c(placebo_att_list, placebo_ATT)}

# Compute SE using distribution of placebo estimates
  se_pseudo <- sd(placebo_att_list, na.rm = TRUE)
  if (!is.na(att_oregon) & !is.na(se_pseudo) & se_pseudo > 0) {
    z_val <- att_oregon / se_pseudo
    p_val <- 2 * pnorm(-abs(z_val))
  } else {
    p_val <- NA}

# Return ATT, SE, and p-value as list for this pseudo-treatment period
  list(ATT = att_oregon, SE = se_pseudo, p_value = p_val)})

# Merge results across all pseudo-treatment dates into a dataframe
pseudo_results_nested_df <- data.frame(
  Pseudo_Treatment_Date = pseudo_periods,
  ATT = sapply(pseudo_results_nested, `[[`, "ATT"),
  SE = sapply(pseudo_results_nested, `[[`, "SE"),
  p_value = sapply(pseudo_results_nested, `[[`, "p_value"))

# Correct ATT's bias
bias_estimate <- mean(pseudo_results_nested_df$ATT, na.rm = TRUE)
att_bias_corrected <- att - bias_estimate
print(bias_estimate)
```

```
## [1] -0.1232997
```

```r
print(att_bias_corrected)
```

```
## [1] 0.04333967
```

```r
# Display results
kable(pseudo_results_nested_df, caption = "Nested Parallel Pseudo-Treatment Test Results") %>%
  kable_styling(bootstrap_options = c("striped", "hold_position"))
```

```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")

## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
```

```
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
## Warning in attr(.knitEnv$meta, "knit_meta_id"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

```
## Warning in attr(x, "align"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

```
## Warning in attr(x, "format"): 'xfun::attr()' is deprecated.
## Use 'xfun::attr2()' instead.
## See help("Deprecated")
```

Table 6: Nested Parallel Pseudo-Treatment Test Results

Pseudo_Treatment_Date	ATT	SE	p_value
2015-01-01	-0.0827550	0.1627413	0.6110983
2015-02-01	-0.0706487	0.1637458	0.6661385
2015-03-01	-0.0757093	0.1884103	0.6878075
2015-04-01	-0.0732283	0.1770219	0.6791171
2015-05-01	-0.0859242	0.1585736	0.5879172

2015-06-01	-0.0858965	0.1345224	0.5231294
2015-07-01	-0.0891618	0.1404061	0.5254103
2015-08-01	-0.0954612	0.1670356	0.5676594
2015-09-01	-0.0942475	0.1704264	0.5802568
2015-10-01	-0.0976868	0.2092531	0.6406175
2015-11-01	-0.0949838	0.1637562	0.5618931
2015-12-01	-0.0936220	0.1847434	0.6123176
2016-01-01	-0.0981282	0.1604705	0.5408667
2016-02-01	-0.1026357	0.1884116	0.5859312
2016-03-01	-0.1098459	0.1401633	0.4332164
2016-04-01	-0.1125975	0.1245931	0.3661433
2016-05-01	-0.1271328	0.1558954	0.4147860
2016-06-01	-0.1298673	0.1785972	0.4671327
2016-07-01	-0.1328737	0.1670893	0.4264823
2016-08-01	-0.1350954	0.1728816	0.4345480
2016-09-01	-0.1353912	0.1610854	0.4006318
2016-10-01	-0.1388422	0.1822714	0.4462192
2016-11-01	-0.1406127	0.1899659	0.4591789
2016-12-01	-0.1432327	0.1994012	0.4725639
2017-01-01	-0.1454488	0.1955731	0.4570547
2017-02-01	-0.1550669	0.1605314	0.3340642
2017-03-01	-0.1578954	0.1752808	0.3676873
2017-04-01	-0.1591483	0.2180138	0.4653953
2017-05-01	-0.1648333	0.2005028	0.4110200
2017-06-01	-0.1647402	0.1989160	0.4075631
2017-07-01	-0.1662160	0.1888377	0.3787479
2017-08-01	-0.1682317	0.2109204	0.4250982
2017-09-01	-0.1707842	0.1738409	0.3258947
2017-10-01	-0.1744075	0.1864221	0.3495040
2017-11-01	-0.1704315	0.2017590	0.3982625
2017-12-01	-0.1687330	0.1883158	0.3702470
2018-01-01	-0.1511033	0.1747993	0.3873468
2018-02-01	-0.1407953	0.1868888	0.4512313
2018-03-01	-0.1373423	0.1734883	0.4285641
2018-04-01	-0.1315467	0.2075634	0.5262332
2018-05-01	-0.1332178	0.2022406	0.5100823
2018-06-01	-0.1301488	0.1506565	0.3876551
2018-07-01	-0.1311230	0.1886679	0.4870591
2018-08-01	-0.1278244	0.1952637	0.5127097
2018-09-01	-0.1219431	0.2182920	0.5764185
2018-10-01	-0.1233561	0.1865990	0.5085636
2018-11-01	-0.1216885	0.2248845	0.5884278
2018-12-01	-0.1231813	0.2098372	0.5571818
2019-01-01	-0.1239120	0.1818236	0.4955580
2019-02-01	-0.1150870	0.1859132	0.5358924
2019-03-01	-0.1207534	0.1790284	0.4999981
2019-04-01	-0.1210245	0.1910577	0.5264434
2019-05-01	-0.1279102	0.2304936	0.5789355
2019-06-01	-0.1320526	0.1739349	0.4477287
2019-07-01	-0.1377227	0.2006557	0.4924840

2019-08-01	-0.1354053	0.2262569	0.5495342
2019-09-01	-0.1310543	0.2398062	0.5847217
2019-10-01	-0.1360527	0.2761552	0.6222476
2019-11-01	-0.1247594	0.2934737	0.6707545
2019-12-01	-0.1190971	0.2362134	0.6141260
2020-01-01	-0.1145537	0.2409321	0.6344588
2020-02-01	-0.1166256	0.3052027	0.7023685
2020-03-01	-0.1269430	0.2446045	0.6037799
2020-04-01	-0.1322853	0.2686923	0.6224860
2020-05-01	-0.1517854	0.2636377	0.5647943
2020-06-01	-0.1579029	0.3101533	0.6106735
2020-07-01	-0.1493356	0.1954311	0.4447873
2020-08-01	-0.1369775	0.2376456	0.5643487
2020-09-01	-0.1147620	0.2546726	0.6522593
2020-10-01	-0.1100578	0.3001318	0.7138440
2020-11-01	-0.0643787	0.2715289	0.8125816
2020-12-01	-0.0592107	0.2137676	0.7817906
2021-01-01	0.0435573	0.1936818	0.8220640
2015-01-01	-0.0827550	0.1616487	0.6086906
2015-02-01	-0.0706487	0.1583792	0.6555442
2015-03-01	-0.0757093	0.1548251	0.6248424
2015-04-01	-0.0732283	0.1750776	0.6757556
2015-05-01	-0.0859242	0.1529873	0.5743597
2015-06-01	-0.0858965	0.1880101	0.6477634
2015-07-01	-0.0891618	0.1732959	0.6068982
2015-08-01	-0.0954612	0.1835729	0.6030511
2015-09-01	-0.0942475	0.1910798	0.6218457
2015-10-01	-0.0976868	0.1477640	0.5085481
2015-11-01	-0.0949838	0.1859832	0.6095528
2015-12-01	-0.0936220	0.1822659	0.6074923
2016-01-01	-0.0981282	0.1868488	0.5994620
2016-02-01	-0.1026357	0.1799241	0.5683802
2016-03-01	-0.1098459	0.1625379	0.4991567
2016-04-01	-0.1125975	0.1816332	0.5353126
2016-05-01	-0.1271328	0.2114145	0.5476111
2016-06-01	-0.1298673	0.1768074	0.4626362
2016-07-01	-0.1328737	0.1881390	0.4800309
2016-08-01	-0.1350954	0.1919016	0.4814437
2016-09-01	-0.1353912	0.1777588	0.4462648
2016-10-01	-0.1388422	0.1864003	0.4563563
2016-11-01	-0.1406127	0.2067760	0.4964890
2016-12-01	-0.1432327	0.1926425	0.4571697
2017-01-01	-0.1454488	0.2080361	0.4844574
2017-02-01	-0.1550669	0.1919185	0.4191003
2017-03-01	-0.1578954	0.1861300	0.3962671
2017-04-01	-0.1591483	0.1785535	0.3727576
2017-05-01	-0.1648333	0.2239808	0.4617756
2017-06-01	-0.1647402	0.2172288	0.4482285
2017-07-01	-0.1662160	0.1832713	0.3644387
2017-08-01	-0.1682317	0.1834155	0.3590293

2017-09-01	-0.1707842	0.1843860	0.3543255
2017-10-01	-0.1744075	0.1898426	0.3582550
2017-11-01	-0.1704315	0.1984586	0.3904641
2017-12-01	-0.1687330	0.1889034	0.3717372
2018-01-01	-0.1511033	0.1962196	0.4412571
2018-02-01	-0.1407953	0.1812407	0.4372525
2018-03-01	-0.1373423	0.1920186	0.4744514
2018-04-01	-0.1315467	0.1967339	0.5037168
2018-05-01	-0.1332178	0.1841615	0.4694496
2018-06-01	-0.1301488	0.2047240	0.5249536
2018-07-01	-0.1311230	0.1755861	0.4552003
2018-08-01	-0.1278244	0.2002370	0.5232359
2018-09-01	-0.1219431	0.2046272	0.5512232
2018-10-01	-0.1233561	0.1959448	0.5289922
2018-11-01	-0.1216885	0.1791512	0.4969792
2018-12-01	-0.1231813	0.1976447	0.5331227
2019-01-01	-0.1239120	0.1963689	0.5280298
2019-02-01	-0.1150870	0.1565901	0.4623656
2019-03-01	-0.1207534	0.1881884	0.5210926
2019-04-01	-0.1210245	0.2483991	0.6261040
2019-05-01	-0.1279102	0.2261594	0.5716824
2019-06-01	-0.1320526	0.2017963	0.5128634
2019-07-01	-0.1377227	0.2571269	0.5922200
2019-08-01	-0.1354053	0.2153334	0.5294689
2019-09-01	-0.1310543	0.2541988	0.6061629
2019-10-01	-0.1360527	0.2049846	0.5068685
2019-11-01	-0.1247594	0.2709721	0.6452189
2019-12-01	-0.1190971	0.1893776	0.5294231
2020-01-01	-0.1145537	0.3048821	0.7071168
2020-02-01	-0.1166256	0.2970276	0.6945836
2020-03-01	-0.1269430	0.2934574	0.6653218
2020-04-01	-0.1322853	0.2720625	0.6268032
2020-05-01	-0.1517854	0.3136027	0.6283820
2020-06-01	-0.1579029	0.2622347	0.5470788
2020-07-01	-0.1493356	0.2465032	0.5446368
2020-08-01	-0.1369775	0.2855036	0.6313873
2020-09-01	-0.1147620	0.2633149	0.6629561
2020-10-01	-0.1100578	0.2658202	0.6788512
2020-11-01	-0.0643787	0.2612240	0.8053338
2020-12-01	-0.0592107	0.2139575	0.7819794
2021-01-01	0.0435573	0.1644157	0.7910696
2015-01-01	-0.0827550	0.1506287	0.5827328
2015-02-01	-0.0706487	0.1468817	0.6305231
2015-03-01	-0.0757093	0.1713525	0.6586095
2015-04-01	-0.0732283	0.1469130	0.6181692
2015-05-01	-0.0859242	0.1897309	0.6506397
2015-06-01	-0.0858965	0.1389690	0.5365107
2015-07-01	-0.0891618	0.1507651	0.5542555
2015-08-01	-0.0954612	0.1734634	0.5820965
2015-09-01	-0.0942475	0.1608789	0.5579906

2015-10-01	-0.0976868	0.1805895	0.5885538
2015-11-01	-0.0949838	0.1340661	0.4786443
2015-12-01	-0.0936220	0.1698407	0.5814729
2016-01-01	-0.0981282	0.1594616	0.5383092
2016-02-01	-0.1026357	0.1814781	0.5716971
2016-03-01	-0.1098459	0.1518074	0.4693190
2016-04-01	-0.1125975	0.1857678	0.5444355
2016-05-01	-0.1271328	0.1642419	0.4388965
2016-06-01	-0.1298673	0.1677832	0.4389198
2016-07-01	-0.1328737	0.1421511	0.3499245
2016-08-01	-0.1350954	0.1541572	0.3808410
2016-09-01	-0.1353912	0.1685915	0.4219329
2016-10-01	-0.1388422	0.2234837	0.5344266
2016-11-01	-0.1406127	0.2051928	0.4931728
2016-12-01	-0.1432327	0.1453413	0.3243825
2017-01-01	-0.1454488	0.1707856	0.3944109
2017-02-01	-0.1550669	0.1692011	0.3594231
2017-03-01	-0.1578954	0.2019269	0.4342480
2017-04-01	-0.1591483	0.1949146	0.4142128
2017-05-01	-0.1648333	0.1996685	0.4090679
2017-06-01	-0.1647402	0.2120019	0.4371179
2017-07-01	-0.1662160	0.1929317	0.3889474
2017-08-01	-0.1682317	0.2132055	0.4300777
2017-09-01	-0.1707842	0.2014797	0.3966332
2017-10-01	-0.1744075	0.2158512	0.4190912
2017-11-01	-0.1704315	0.1886186	0.3662194
2017-12-01	-0.1687330	0.1927758	0.3814210
2018-01-01	-0.1511033	0.2007642	0.4516660
2018-02-01	-0.1407953	0.1705665	0.4091123
2018-03-01	-0.1373423	0.1900628	0.4699163
2018-04-01	-0.1315467	0.2059694	0.5230368
2018-05-01	-0.1332178	0.1971682	0.4992593
2018-06-01	-0.1301488	0.1792447	0.4677801
2018-07-01	-0.1311230	0.1999322	0.5119286
2018-08-01	-0.1278244	0.1946057	0.5112852
2018-09-01	-0.1219431	0.2064546	0.5547528
2018-10-01	-0.1233561	0.2249094	0.5833691
2018-11-01	-0.1216885	0.2114356	0.5649297
2018-12-01	-0.1231813	0.1586057	0.4373647
2019-01-01	-0.1239120	0.1971311	0.5296262
2019-02-01	-0.1150870	0.2069700	0.5781723
2019-03-01	-0.1207534	0.2116437	0.5683044
2019-04-01	-0.1210245	0.2109057	0.5660813
2019-05-01	-0.1279102	0.2260717	0.5715332
2019-06-01	-0.1320526	0.2220365	0.5520214
2019-07-01	-0.1377227	0.1976803	0.4859941
2019-08-01	-0.1354053	0.1714627	0.4296987
2019-09-01	-0.1310543	0.1760524	0.4566314
2019-10-01	-0.1360527	0.2315985	0.5569010
2019-11-01	-0.1247594	0.2563186	0.6264456

2019-12-01	-0.1190971	0.2743498	0.6642110
2020-01-01	-0.1145537	0.2423874	0.6364945
2020-02-01	-0.1166256	0.2219551	0.5992725
2020-03-01	-0.1269430	0.2260925	0.5744805
2020-04-01	-0.1322853	0.2179219	0.5438305
2020-05-01	-0.1517854	0.2437392	0.5334575
2020-06-01	-0.1579029	0.2682824	0.5561500
2020-07-01	-0.1493356	0.1997420	0.4546758
2020-08-01	-0.1369775	0.2684489	0.6098724
2020-09-01	-0.1147620	0.2364769	0.6274643
2020-10-01	-0.1100578	0.2811655	0.6954762
2020-11-01	-0.0643787	0.2564200	0.8017620
2020-12-01	-0.0592107	0.1977818	0.7646549
2021-01-01	0.0435573	0.1635319	0.7899668
2015-01-01	-0.0827550	0.1309149	0.5273031
2015-02-01	-0.0706487	0.1744127	0.6854287
2015-03-01	-0.0757093	0.1641823	0.6447055
2015-04-01	-0.0732283	0.1856989	0.6933307
2015-05-01	-0.0859242	0.1182175	0.4673294
2015-06-01	-0.0858965	0.1246955	0.4909178
2015-07-01	-0.0891618	0.1825783	0.6253032
2015-08-01	-0.0954612	0.1772139	0.5901091
2015-09-01	-0.0942475	0.1870401	0.6143392
2015-10-01	-0.0976868	0.1705270	0.5667447
2015-11-01	-0.0949838	0.1451579	0.5128875
2015-12-01	-0.0936220	0.1849063	0.6126309
2016-01-01	-0.0981282	0.1781498	0.5817583
2016-02-01	-0.1026357	0.1748974	0.5573155
2016-03-01	-0.1098459	0.1563940	0.4824504
2016-04-01	-0.1125975	0.1968379	0.5673007
2016-05-01	-0.1271328	0.1941489	0.5125830
2016-06-01	-0.1298673	0.1358481	0.3390851
2016-07-01	-0.1328737	0.1579087	0.4000910
2016-08-01	-0.1350954	0.1759473	0.4425958
2016-09-01	-0.1353912	0.1687698	0.4224232
2016-10-01	-0.1388422	0.2242460	0.5358169
2016-11-01	-0.1406127	0.1952308	0.4713784
2016-12-01	-0.1432327	0.1891494	0.4489024
2017-01-01	-0.1454488	0.2124890	0.4936593
2017-02-01	-0.1550669	0.1838548	0.3989933
2017-03-01	-0.1578954	0.1883600	0.4018825
2017-04-01	-0.1591483	0.2246626	0.4787044
2017-05-01	-0.1648333	0.1999881	0.4098169
2017-06-01	-0.1647402	0.1875303	0.3796876
2017-07-01	-0.1662160	0.1837263	0.3656277
2017-08-01	-0.1682317	0.1768362	0.3414308
2017-09-01	-0.1707842	0.2132801	0.4232760
2017-10-01	-0.1744075	0.2175474	0.4227272
2017-11-01	-0.1704315	0.1911837	0.3726854
2017-12-01	-0.1687330	0.1699995	0.3209293

2018-01-01	-0.1511033	0.1887145	0.4233066
2018-02-01	-0.1407953	0.1866508	0.4506545
2018-03-01	-0.1373423	0.1700288	0.4192295
2018-04-01	-0.1315467	0.1748153	0.4517567
2018-05-01	-0.1332178	0.2048293	0.5154439
2018-06-01	-0.1301488	0.1845170	0.4805933
2018-07-01	-0.1311230	0.1762938	0.4570123
2018-08-01	-0.1278244	0.1955965	0.5134273
2018-09-01	-0.1219431	0.1986967	0.5394039
2018-10-01	-0.1233561	0.1991044	0.5355508
2018-11-01	-0.1216885	0.2154234	0.5721549
2018-12-01	-0.1231813	0.1975908	0.5330112
2019-01-01	-0.1239120	0.2040536	0.5436836
2019-02-01	-0.1150870	0.2097961	0.5833033
2019-03-01	-0.1207534	0.1993863	0.5447636
2019-04-01	-0.1210245	0.1693193	0.4747508
2019-05-01	-0.1279102	0.2010753	0.5246911
2019-06-01	-0.1320526	0.1931749	0.4942335
2019-07-01	-0.1377227	0.1953931	0.4809040
2019-08-01	-0.1354053	0.2309430	0.5576639
2019-09-01	-0.1310543	0.2575288	0.6108281
2019-10-01	-0.1360527	0.2763789	0.6225294
2019-11-01	-0.1247594	0.2342789	0.5943623
2019-12-01	-0.1190971	0.2269156	0.5996862
2020-01-01	-0.1145537	0.1972737	0.5614535
2020-02-01	-0.1166256	0.2550273	0.6474507
2020-03-01	-0.1269430	0.2356661	0.5901239
2020-04-01	-0.1322853	0.2754991	0.6311093
2020-05-01	-0.1517854	0.2835550	0.5924463
2020-06-01	-0.1579029	0.2819462	0.5754480
2020-07-01	-0.1493356	0.2484602	0.5478105
2020-08-01	-0.1369775	0.2695529	0.6113372
2020-09-01	-0.1147620	0.2689459	0.6695903
2020-10-01	-0.1100578	0.2840084	0.6983742
2020-11-01	-0.0643787	0.2665598	0.8091545
2020-12-01	-0.0592107	0.2241251	0.7916367
2021-01-01	0.0435573	0.1720771	0.8001702
2015-01-01	-0.0827550	0.1615620	0.6084983
2015-02-01	-0.0706487	0.1434393	0.6223418
2015-03-01	-0.0757093	0.1310165	0.5633578
2015-04-01	-0.0732283	0.1754893	0.6764730
2015-05-01	-0.0859242	0.1663351	0.6054546
2015-06-01	-0.0858965	0.1885672	0.6487339
2015-07-01	-0.0891618	0.2048203	0.6633321
2015-08-01	-0.0954612	0.1983476	0.6303159
2015-09-01	-0.0942475	0.1414520	0.5052283
2015-10-01	-0.0976868	0.2003149	0.6257863
2015-11-01	-0.0949838	0.1207392	0.4314659
2015-12-01	-0.0936220	0.1901124	0.6223968
2016-01-01	-0.0981282	0.1859609	0.5977202

2016-02-01	-0.1026357	0.1830978	0.5751036
2016-03-01	-0.1098459	0.1585169	0.4883345
2016-04-01	-0.1125975	0.1544268	0.4659213
2016-05-01	-0.1271328	0.1751508	0.4679324
2016-06-01	-0.1298673	0.1427896	0.3630857
2016-07-01	-0.1328737	0.1785067	0.4566572
2016-08-01	-0.1350954	0.1921042	0.4819062
2016-09-01	-0.1353912	0.1774502	0.4454745
2016-10-01	-0.1388422	0.1917239	0.4689567
2016-11-01	-0.1406127	0.1827684	0.4416860
2016-12-01	-0.1432327	0.1961448	0.4652438
2017-01-01	-0.1454488	0.1787929	0.4159290
2017-02-01	-0.1550669	0.2062714	0.4521944
2017-03-01	-0.1578954	0.1921847	0.4113151
2017-04-01	-0.1591483	0.2034135	0.4339866
2017-05-01	-0.1648333	0.1765973	0.3506209
2017-06-01	-0.1647402	0.1891307	0.3837329
2017-07-01	-0.1662160	0.2017697	0.4100584
2017-08-01	-0.1682317	0.1961832	0.3911556
2017-09-01	-0.1707842	0.2084687	0.4126541
2017-10-01	-0.1744075	0.2175073	0.4226416
2017-11-01	-0.1704315	0.2132159	0.4240944
2017-12-01	-0.1687330	0.1741420	0.3325755
2018-01-01	-0.1511033	0.1434076	0.2920374
2018-02-01	-0.1407953	0.2017345	0.4852250
2018-03-01	-0.1373423	0.1910395	0.4721898
2018-04-01	-0.1315467	0.1634617	0.4209606
2018-05-01	-0.1332178	0.2054550	0.5167238
2018-06-01	-0.1301488	0.1890439	0.4911644
2018-07-01	-0.1311230	0.1659531	0.4294570
2018-08-01	-0.1278244	0.1684760	0.4480263
2018-09-01	-0.1219431	0.1705257	0.4745466
2018-10-01	-0.1233561	0.2179948	0.5714839
2018-11-01	-0.1216885	0.1938428	0.5301551
2018-12-01	-0.1231813	0.2112515	0.5598242
2019-01-01	-0.1239120	0.1643880	0.4509827
2019-02-01	-0.1150870	0.1630978	0.4804170
2019-03-01	-0.1207534	0.1737156	0.4869795
2019-04-01	-0.1210245	0.1928364	0.5302648
2019-05-01	-0.1279102	0.2385525	0.5918252
2019-06-01	-0.1320526	0.2065537	0.5226189
2019-07-01	-0.1377227	0.2015997	0.4945123
2019-08-01	-0.1354053	0.2208047	0.5397205
2019-09-01	-0.1310543	0.2340101	0.5754543
2019-10-01	-0.1360527	0.2419727	0.5739357
2019-11-01	-0.1247594	0.2955185	0.6729000
2019-12-01	-0.1190971	0.2678156	0.6565380
2020-01-01	-0.1145537	0.2679773	0.6690332
2020-02-01	-0.1166256	0.2730103	0.6692453
2020-03-01	-0.1269430	0.2316601	0.5837117

2020-04-01	-0.1322853	0.2414195	0.5837276
2020-05-01	-0.1517854	0.2287510	0.5069849
2020-06-01	-0.1579029	0.2818359	0.5752985
2020-07-01	-0.1493356	0.2757018	0.5880553
2020-08-01	-0.1369775	0.2553144	0.5916093
2020-09-01	-0.1147620	0.2610173	0.6601747
2020-10-01	-0.1100578	0.2474203	0.6564488
2020-11-01	-0.0643787	0.2625521	0.8062989
2020-12-01	-0.0592107	0.2466349	0.8102727
2021-01-01	0.0435573	0.2013709	0.8287507
2015-01-01	-0.0827550	0.1350977	0.5401697
2015-02-01	-0.0706487	0.1454590	0.6271833
2015-03-01	-0.0757093	0.1496913	0.6130184
2015-04-01	-0.0732283	0.1827502	0.6886397
2015-05-01	-0.0859242	0.1499369	0.5665981
2015-06-01	-0.0858965	0.1470755	0.5592003
2015-07-01	-0.0891618	0.1521177	0.5577836
2015-08-01	-0.0954612	0.1670243	0.5676330
2015-09-01	-0.0942475	0.1587125	0.5526292
2015-10-01	-0.0976868	0.0903052	0.2793676
2015-11-01	-0.0949838	0.1737090	0.5845172
2015-12-01	-0.0936220	0.2157514	0.6643357
2016-01-01	-0.0981282	0.1553620	0.5276418
2016-02-01	-0.1026357	0.1592455	0.5192433
2016-03-01	-0.1098459	0.1540443	0.4757961
2016-04-01	-0.1125975	0.1916301	0.5568162
2016-05-01	-0.1271328	0.1972793	0.5192964
2016-06-01	-0.1298673	0.1698451	0.4444966
2016-07-01	-0.1328737	0.1938953	0.4931635
2016-08-01	-0.1350954	0.1790586	0.4505638
2016-09-01	-0.1353912	0.1683952	0.4213919
2016-10-01	-0.1388422	0.1928889	0.4716459
2016-11-01	-0.1406127	0.1690726	0.4055949
2016-12-01	-0.1432327	0.1711567	0.4026764
2017-01-01	-0.1454488	0.1816207	0.4232255
2017-02-01	-0.1550669	0.1922741	0.4199610
2017-03-01	-0.1578954	0.1755157	0.3683289
2017-04-01	-0.1591483	0.1845246	0.3884250
2017-05-01	-0.1648333	0.1795491	0.3585977
2017-06-01	-0.1647402	0.1939447	0.3956486
2017-07-01	-0.1662160	0.1948526	0.3936402
2017-08-01	-0.1682317	0.1746863	0.3355222
2017-09-01	-0.1707842	0.2101729	0.4164538
2017-10-01	-0.1744075	0.2030433	0.3903587
2017-11-01	-0.1704315	0.1997218	0.3934682
2017-12-01	-0.1687330	0.1831773	0.3569742
2018-01-01	-0.1511033	0.1902329	0.4270170
2018-02-01	-0.1407953	0.1752531	0.4217537
2018-03-01	-0.1373423	0.1929510	0.4765895
2018-04-01	-0.1315467	0.1996836	0.5100399

2018-05-01	-0.1332178	0.2074980	0.5208602
2018-06-01	-0.1301488	0.1901116	0.4936018
2018-07-01	-0.1311230	0.1867801	0.4826678
2018-08-01	-0.1278244	0.1851947	0.4900582
2018-09-01	-0.1219431	0.2105205	0.5624234
2018-10-01	-0.1233561	0.2350147	0.5996618
2018-11-01	-0.1216885	0.1950105	0.5326209
2018-12-01	-0.1231813	0.2056852	0.5492512
2019-01-01	-0.1239120	0.1551573	0.4245099
2019-02-01	-0.1150870	0.1802591	0.5231788
2019-03-01	-0.1207534	0.2016986	0.5493848
2019-04-01	-0.1210245	0.1797274	0.5007069
2019-05-01	-0.1279102	0.2258684	0.5711871
2019-06-01	-0.1320526	0.2373408	0.5779485
2019-07-01	-0.1377227	0.2188608	0.5291717
2019-08-01	-0.1354053	0.2024075	0.5035121
2019-09-01	-0.1310543	0.2247663	0.5598468
2019-10-01	-0.1360527	0.2363578	0.5648709
2019-11-01	-0.1247594	0.2561831	0.6262631
2019-12-01	-0.1190971	0.1928445	0.5368516
2020-01-01	-0.1145537	0.2567215	0.6554400
2020-02-01	-0.1166256	0.2434570	0.6319101
2020-03-01	-0.1269430	0.2316826	0.5837482
2020-04-01	-0.1322853	0.2647118	0.6172628
2020-05-01	-0.1517854	0.2642822	0.5657439
2020-06-01	-0.1579029	0.2792609	0.5717804
2020-07-01	-0.1493356	0.2134317	0.4841221
2020-08-01	-0.1369775	0.2571610	0.5942739
2020-09-01	-0.1147620	0.2543702	0.6518733
2020-10-01	-0.1100578	0.2235080	0.6224286
2020-11-01	-0.0643787	0.2910552	0.8249440
2020-12-01	-0.0592107	0.2068899	0.7747293
2021-01-01	0.0435573	0.1763916	0.8049582
2015-01-01	-0.0827550	0.1256186	0.5100372
2015-02-01	-0.0706487	0.1325264	0.5939701
2015-03-01	-0.0757093	0.1734060	0.6624004
2015-04-01	-0.0732283	0.1726938	0.6715399
2015-05-01	-0.0859242	0.1605477	0.5925157
2015-06-01	-0.0858965	0.1463622	0.5572871
2015-07-01	-0.0891618	0.1466681	0.5432435
2015-08-01	-0.0954612	0.1632354	0.5586775
2015-09-01	-0.0942475	0.1677526	0.5742358
2015-10-01	-0.0976868	0.1823926	0.5922452
2015-11-01	-0.0949838	0.1778031	0.5931977
2015-12-01	-0.0936220	0.1609757	0.5608421
2016-01-01	-0.0981282	0.1839189	0.5936596
2016-02-01	-0.1026357	0.1950678	0.5987814
2016-03-01	-0.1098459	0.1894431	0.5620252
2016-04-01	-0.1125975	0.1568106	0.4727280
2016-05-01	-0.1271328	0.1536443	0.4079828

2016-06-01	-0.1298673	0.1786625	0.4672955
2016-07-01	-0.1328737	0.1764469	0.4514187
2016-08-01	-0.1350954	0.1781404	0.4482329
2016-09-01	-0.1353912	0.1717660	0.4305619
2016-10-01	-0.1388422	0.1643547	0.3982384
2016-11-01	-0.1406127	0.1790590	0.4322853
2016-12-01	-0.1432327	0.2014514	0.4770822
2017-01-01	-0.1454488	0.1752385	0.4065361
2017-02-01	-0.1550669	0.1897868	0.4138951
2017-03-01	-0.1578954	0.1988390	0.4271450
2017-04-01	-0.1591483	0.1956748	0.4160286
2017-05-01	-0.1648333	0.2045327	0.4202993
2017-06-01	-0.1647402	0.2058602	0.4235645
2017-07-01	-0.1662160	0.2103164	0.4293444
2017-08-01	-0.1682317	0.2246449	0.4539306
2017-09-01	-0.1707842	0.1859369	0.3583541
2017-10-01	-0.1744075	0.1883123	0.3543624
2017-11-01	-0.1704315	0.1768432	0.3351743
2017-12-01	-0.1687330	0.1582969	0.2864564
2018-01-01	-0.1511033	0.2023982	0.4553266
2018-02-01	-0.1407953	0.1794024	0.4325699
2018-03-01	-0.1373423	0.2055907	0.5041097
2018-04-01	-0.1315467	0.1907132	0.4903439
2018-05-01	-0.1332178	0.1777974	0.4536960
2018-06-01	-0.1301488	0.1570468	0.4072595
2018-07-01	-0.1311230	0.1908756	0.4921108
2018-08-01	-0.1278244	0.1852597	0.4902106
2018-09-01	-0.1219431	0.1859624	0.5119911
2018-10-01	-0.1233561	0.1859844	0.5071635
2018-11-01	-0.1216885	0.1753212	0.4876264
2018-12-01	-0.1231813	0.1888636	0.5142569
2019-01-01	-0.1239120	0.1754864	0.4801222
2019-02-01	-0.1150870	0.2174858	0.5966873
2019-03-01	-0.1207534	0.1956776	0.5371663
2019-04-01	-0.1210245	0.2019846	0.5490550
2019-05-01	-0.1279102	0.2119879	0.5462530
2019-06-01	-0.1320526	0.2283924	0.5631406
2019-07-01	-0.1377227	0.1786148	0.4406714
2019-08-01	-0.1354053	0.1938393	0.4848369
2019-09-01	-0.1310543	0.2592765	0.6132347
2019-10-01	-0.1360527	0.2424463	0.5746843
2019-11-01	-0.1247594	0.2208792	0.5721887
2019-12-01	-0.1190971	0.2106632	0.5718401
2020-01-01	-0.1145537	0.2440619	0.6388100
2020-02-01	-0.1166256	0.2159044	0.5890780
2020-03-01	-0.1269430	0.2106371	0.5467334
2020-04-01	-0.1322853	0.2421530	0.5848679
2020-05-01	-0.1517854	0.3228664	0.6382705
2020-06-01	-0.1579029	0.2690313	0.5572499
2020-07-01	-0.1493356	0.2236233	0.5042614

2020-08-01	-0.1369775	0.2887722	0.6352542
2020-09-01	-0.1147620	0.2591213	0.6578460
2020-10-01	-0.1100578	0.2687541	0.6821645
2020-11-01	-0.0643787	0.2999251	0.8300408
2020-12-01	-0.0592107	0.2577525	0.8183100
2021-01-01	0.0435573	0.1892515	0.8179707
2015-01-01	-0.0827550	0.1874539	0.6588738
2015-02-01	-0.0706487	0.1845057	0.7017879
2015-03-01	-0.0757093	0.1808313	0.6754545
2015-04-01	-0.0732283	0.1457688	0.6154147
2015-05-01	-0.0859242	0.1716624	0.6166938
2015-06-01	-0.0858965	0.1658528	0.6045227
2015-07-01	-0.0891618	0.1861179	0.6318953
2015-08-01	-0.0954612	0.1608697	0.5529088
2015-09-01	-0.0942475	0.1451866	0.5162433
2015-10-01	-0.0976868	0.1547170	0.5277855
2015-11-01	-0.0949838	0.1765397	0.5905556
2015-12-01	-0.0936220	0.1895274	0.6213230
2016-01-01	-0.0981282	0.1984461	0.6209652
2016-02-01	-0.1026357	0.1738851	0.5550231
2016-03-01	-0.1098459	0.1857007	0.5541710
2016-04-01	-0.1125975	0.1976996	0.5689910
2016-05-01	-0.1271328	0.1792378	0.4781404
2016-06-01	-0.1298673	0.1611884	0.4204235
2016-07-01	-0.1328737	0.1580447	0.4004965
2016-08-01	-0.1350954	0.1498457	0.3672889
2016-09-01	-0.1353912	0.1797903	0.4514195
2016-10-01	-0.1388422	0.1644602	0.3985413
2016-11-01	-0.1406127	0.1891501	0.4572444
2016-12-01	-0.1432327	0.1909088	0.4530937
2017-01-01	-0.1454488	0.1862890	0.4349381
2017-02-01	-0.1550669	0.1722339	0.3679459
2017-03-01	-0.1578954	0.2011782	0.4325395
2017-04-01	-0.1591483	0.2155478	0.4603060
2017-05-01	-0.1648333	0.2233140	0.4604394
2017-06-01	-0.1647402	0.1675148	0.3253927
2017-07-01	-0.1662160	0.2163187	0.4422585
2017-08-01	-0.1682317	0.1904035	0.3769373
2017-09-01	-0.1707842	0.1812979	0.3461882
2017-10-01	-0.1744075	0.2164551	0.4203901
2017-11-01	-0.1704315	0.1676593	0.3093747
2017-12-01	-0.1687330	0.2028785	0.4055813
2018-01-01	-0.1511033	0.1799070	0.4009663
2018-02-01	-0.1407953	0.1639550	0.3904823
2018-03-01	-0.1373423	0.2167078	0.5262329
2018-04-01	-0.1315467	0.1834708	0.4733803
2018-05-01	-0.1332178	0.1915250	0.4867020
2018-06-01	-0.1301488	0.2105269	0.5364403
2018-07-01	-0.1311230	0.1754230	0.4547813
2018-08-01	-0.1278244	0.2071226	0.5371401

2018-09-01	-0.1219431	0.1930504	0.5276060
2018-10-01	-0.1233561	0.2285413	0.5893665
2018-11-01	-0.1216885	0.1827869	0.5055773
2018-12-01	-0.1231813	0.2110648	0.5594772
2019-01-01	-0.1239120	0.2098512	0.5548719
2019-02-01	-0.1150870	0.2019520	0.5687638
2019-03-01	-0.1207534	0.2007283	0.5474562
2019-04-01	-0.1210245	0.2277300	0.5951149
2019-05-01	-0.1279102	0.2268668	0.5728821
2019-06-01	-0.1320526	0.1939314	0.4959195
2019-07-01	-0.1377227	0.2337229	0.5556892
2019-08-01	-0.1354053	0.2297496	0.5556195
2019-09-01	-0.1310543	0.1895306	0.4892716
2019-10-01	-0.1360527	0.2571383	0.5967338
2019-11-01	-0.1247594	0.2606908	0.6322428
2019-12-01	-0.1190971	0.2099807	0.5705912
2020-01-01	-0.1145537	0.2381233	0.6304677
2020-02-01	-0.1166256	0.2439721	0.6326298
2020-03-01	-0.1269430	0.2696664	0.6378256
2020-04-01	-0.1322853	0.2769471	0.6328954
2020-05-01	-0.1517854	0.2933275	0.6048346
2020-06-01	-0.1579029	0.2277819	0.4881718
2020-07-01	-0.1493356	0.2481516	0.5473127
2020-08-01	-0.1369775	0.2006774	0.4948750
2020-09-01	-0.1147620	0.2337470	0.6234500
2020-10-01	-0.1100578	0.2520891	0.6624137
2020-11-01	-0.0643787	0.2572720	0.8024049
2020-12-01	-0.0592107	0.1971146	0.7638818
2021-01-01	0.0435573	0.1992946	0.8269948
2015-01-01	-0.0827550	0.1735393	0.6334574
2015-02-01	-0.0706487	0.1524531	0.6430683
2015-03-01	-0.0757093	0.1572895	0.6302767
2015-04-01	-0.0732283	0.1794979	0.6833016
2015-05-01	-0.0859242	0.1720529	0.6174938
2015-06-01	-0.0858965	0.1717458	0.6169786
2015-07-01	-0.0891618	0.1852919	0.6303768
2015-08-01	-0.0954612	0.1756359	0.5867734
2015-09-01	-0.0942475	0.1543401	0.5414326
2015-10-01	-0.0976868	0.1618659	0.5461733
2015-11-01	-0.0949838	0.2064254	0.6454184
2015-12-01	-0.0936220	0.1700296	0.5818928
2016-01-01	-0.0981282	0.1913375	0.6080536
2016-02-01	-0.1026357	0.1488934	0.4906199
2016-03-01	-0.1098459	0.1565846	0.4829838
2016-04-01	-0.1125975	0.2006084	0.5746067
2016-05-01	-0.1271328	0.1669632	0.4463933
2016-06-01	-0.1298673	0.1839563	0.4802077
2016-07-01	-0.1328737	0.1881766	0.4801186
2016-08-01	-0.1350954	0.1800138	0.4529703
2016-09-01	-0.1353912	0.1745881	0.4380506

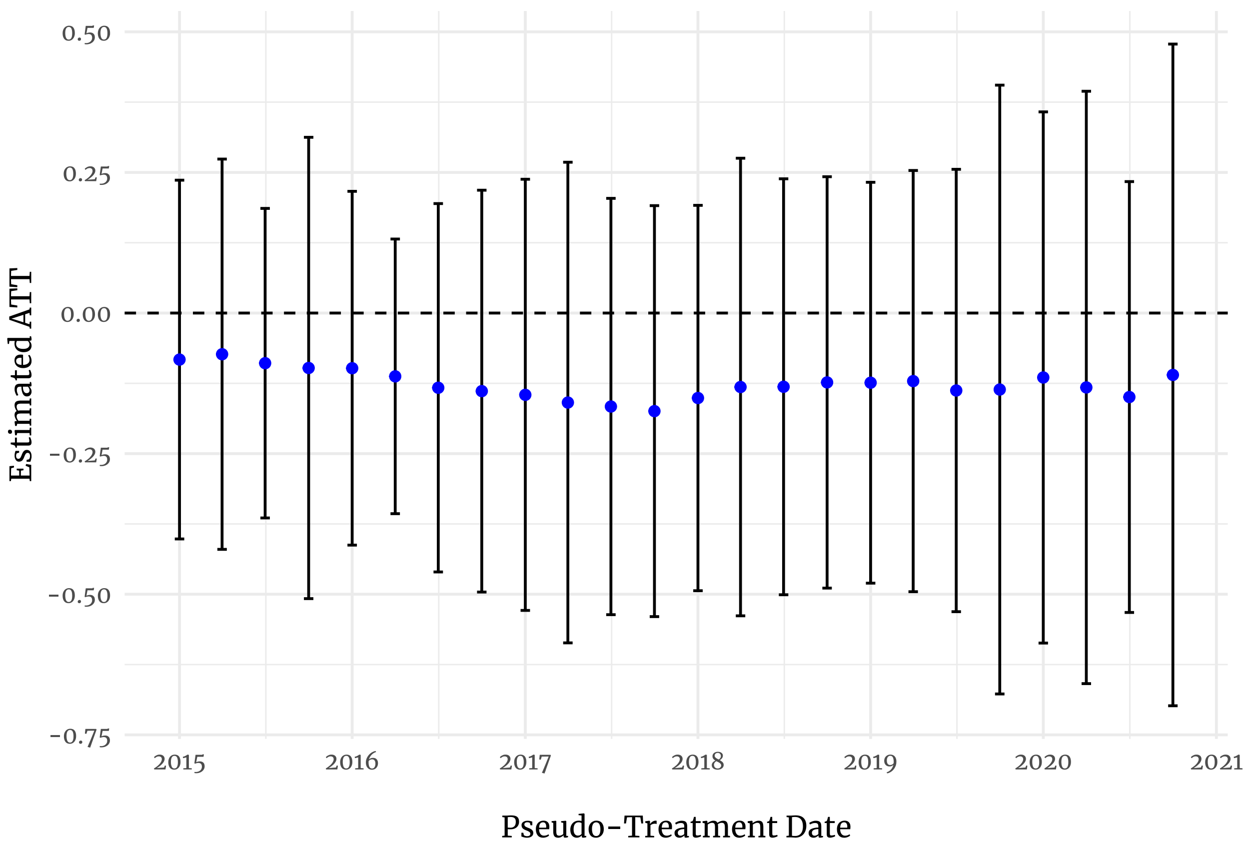
2016-10-01	-0.1388422	0.2081283	0.5047089
2016-11-01	-0.1406127	0.1885522	0.4558186
2016-12-01	-0.1432327	0.1718967	0.4047046
2017-01-01	-0.1454488	0.1636298	0.3740627
2017-02-01	-0.1550669	0.1931364	0.4220395
2017-03-01	-0.1578954	0.1809789	0.3829620
2017-04-01	-0.1591483	0.2078199	0.4437959
2017-05-01	-0.1648333	0.1832058	0.3682711
2017-06-01	-0.1647402	0.1965290	0.4018912
2017-07-01	-0.1662160	0.1576287	0.2916640
2017-08-01	-0.1682317	0.1665038	0.3123144
2017-09-01	-0.1707842	0.2219374	0.4415875
2017-10-01	-0.1744075	0.2122726	0.4112929
2017-11-01	-0.1704315	0.2079964	0.4125602
2017-12-01	-0.1687330	0.1846994	0.3609509
2018-01-01	-0.1511033	0.1597635	0.3442540
2018-02-01	-0.1407953	0.1830473	0.4417898
2018-03-01	-0.1373423	0.1818519	0.4501036
2018-04-01	-0.1315467	0.2029973	0.5169707
2018-05-01	-0.1332178	0.2000691	0.5055011
2018-06-01	-0.1301488	0.1811460	0.4724647
2018-07-01	-0.1311230	0.2172195	0.5460810
2018-08-01	-0.1278244	0.2101056	0.5429344
2018-09-01	-0.1219431	0.1999978	0.5420458
2018-10-01	-0.1233561	0.2348723	0.5994407
2018-11-01	-0.1216885	0.1886129	0.5188132
2018-12-01	-0.1231813	0.1951998	0.5280064
2019-01-01	-0.1239120	0.1776489	0.4854834
2019-02-01	-0.1150870	0.1809201	0.5246979
2019-03-01	-0.1207534	0.2059432	0.5576450
2019-04-01	-0.1210245	0.2156236	0.5746089
2019-05-01	-0.1279102	0.1849933	0.4892944
2019-06-01	-0.1320526	0.2045915	0.5186387
2019-07-01	-0.1377227	0.2027232	0.4969077
2019-08-01	-0.1354053	0.2558971	0.5967085
2019-09-01	-0.1310543	0.2688712	0.6259577
2019-10-01	-0.1360527	0.2342267	0.5613354
2019-11-01	-0.1247594	0.2337417	0.5935152
2019-12-01	-0.1190971	0.2649576	0.6530747
2020-01-01	-0.1145537	0.2250667	0.6107686
2020-02-01	-0.1166256	0.2631972	0.6576852
2020-03-01	-0.1269430	0.2562097	0.6202716
2020-04-01	-0.1322853	0.3009561	0.6602629
2020-05-01	-0.1517854	0.2468968	0.5387048
2020-06-01	-0.1579029	0.2322732	0.4966214
2020-07-01	-0.1493356	0.2584647	0.5634130
2020-08-01	-0.1369775	0.2614420	0.6003267
2020-09-01	-0.1147620	0.2877484	0.6900203
2020-10-01	-0.1100578	0.2308930	0.6336030
2020-11-01	-0.0643787	0.2257928	0.7755500

2020-12-01	-0.0592107	0.1930434	0.7590548
2021-01-01	0.0435573	0.1997223	0.8273594
2015-01-01	-0.0827550	0.1521893	0.5866041
2015-02-01	-0.0706487	0.1900376	0.7100701
2015-03-01	-0.0757093	0.1848230	0.6820762
2015-04-01	-0.0732283	0.1657548	0.6586434
2015-05-01	-0.0859242	0.1643558	0.6011179
2015-06-01	-0.0858965	0.1682976	0.6097824
2015-07-01	-0.0891618	0.1452567	0.5393327
2015-08-01	-0.0954612	0.2000869	0.6332918
2015-09-01	-0.0942475	0.1759854	0.5922755
2015-10-01	-0.0976868	0.1838549	0.5951933
2015-11-01	-0.0949838	0.1676003	0.5708989
2015-12-01	-0.0936220	0.1884118	0.6192588
2016-01-01	-0.0981282	0.1678509	0.5588066
2016-02-01	-0.1026357	0.1733811	0.5538736
2016-03-01	-0.1098459	0.1582703	0.4876569
2016-04-01	-0.1125975	0.1668411	0.4997531
2016-05-01	-0.1271328	0.1233354	0.3026397
2016-06-01	-0.1298673	0.1494037	0.3847172
2016-07-01	-0.1328737	0.1379752	0.3355345
2016-08-01	-0.1350954	0.1684275	0.4224963
2016-09-01	-0.1353912	0.1740780	0.4367095
2016-10-01	-0.1388422	0.1905252	0.4661646
2016-11-01	-0.1406127	0.1824898	0.4409892
2016-12-01	-0.1432327	0.1815933	0.4302551
2017-01-01	-0.1454488	0.1670062	0.3837988
2017-02-01	-0.1550669	0.1900436	0.4145261
2017-03-01	-0.1578954	0.1968615	0.4225164
2017-04-01	-0.1591483	0.2095799	0.4476325
2017-05-01	-0.1648333	0.1800208	0.3598584
2017-06-01	-0.1647402	0.1770933	0.3522443
2017-07-01	-0.1662160	0.2097889	0.4281850
2017-08-01	-0.1682317	0.1924380	0.3820026
2017-09-01	-0.1707842	0.1940797	0.3788760
2017-10-01	-0.1744075	0.1930776	0.3663652
2017-11-01	-0.1704315	0.2081755	0.4129623
2017-12-01	-0.1687330	0.1733983	0.3305060
2018-01-01	-0.1511033	0.1680464	0.3685589
2018-02-01	-0.1407953	0.2064123	0.4951713
2018-03-01	-0.1373423	0.1533969	0.3706058
2018-04-01	-0.1315467	0.1944081	0.4986265
2018-05-01	-0.1332178	0.1892424	0.4814618
2018-06-01	-0.1301488	0.1883618	0.4895962
2018-07-01	-0.1311230	0.1874381	0.4842061
2018-08-01	-0.1278244	0.1768081	0.4697074
2018-09-01	-0.1219431	0.1921036	0.5255734
2018-10-01	-0.1233561	0.2086865	0.5544485
2018-11-01	-0.1216885	0.1690696	0.4716765
2018-12-01	-0.1231813	0.1687485	0.4654086

2019-01-01	-0.1239120	0.1905475	0.5155019
2019-02-01	-0.1150870	0.1718613	0.5030806
2019-03-01	-0.1207534	0.2016243	0.5492375
2019-04-01	-0.1210245	0.1968911	0.5387680
2019-05-01	-0.1279102	0.2093707	0.5412479
2019-06-01	-0.1320526	0.1955264	0.4994411
2019-07-01	-0.1377227	0.2074623	0.5067899
2019-08-01	-0.1354053	0.2415349	0.5750678
2019-09-01	-0.1310543	0.3029081	0.6652663
2019-10-01	-0.1360527	0.2655855	0.6084589
2019-11-01	-0.1247594	0.2811455	0.6572205
2019-12-01	-0.1190971	0.1668682	0.4754006
2020-01-01	-0.1145537	0.2220454	0.6059227
2020-02-01	-0.1166256	0.2555009	0.6480600
2020-03-01	-0.1269430	0.2290421	0.5794183
2020-04-01	-0.1322853	0.2282352	0.5621837
2020-05-01	-0.1517854	0.2896005	0.6001950
2020-06-01	-0.1579029	0.2598309	0.5433771
2020-07-01	-0.1493356	0.2528648	0.5548053
2020-08-01	-0.1369775	0.2263482	0.5450707
2020-09-01	-0.1147620	0.2508799	0.6473563
2020-10-01	-0.1100578	0.2644277	0.6772552
2020-11-01	-0.0643787	0.2771725	0.8163288
2020-12-01	-0.0592107	0.2016001	0.7689843
2021-01-01	0.0435573	0.1790100	0.8077549
2015-01-01	-0.0827550	0.1141649	0.4685302
2015-02-01	-0.0706487	0.1741502	0.6849799
2015-03-01	-0.0757093	0.1501313	0.6140593
2015-04-01	-0.0732283	0.1948468	0.7070469
2015-05-01	-0.0859242	0.1645316	0.6015067
2015-06-01	-0.0858965	0.1875746	0.6470011
2015-07-01	-0.0891618	0.1132648	0.4311662
2015-08-01	-0.0954612	0.2062818	0.6435287
2015-09-01	-0.0942475	0.1647839	0.5673586
2015-10-01	-0.0976868	0.1853557	0.5981774
2015-11-01	-0.0949838	0.1918589	0.6205499
2015-12-01	-0.0936220	0.1920250	0.6258677
2016-01-01	-0.0981282	0.1729861	0.5705371
2016-02-01	-0.1026357	0.1460931	0.4823449
2016-03-01	-0.1098459	0.1789255	0.5392683
2016-04-01	-0.1125975	0.1577903	0.4754812
2016-05-01	-0.1271328	0.1569702	0.4179881
2016-06-01	-0.1298673	0.1190913	0.2754992
2016-07-01	-0.1328737	0.2086740	0.5242861
2016-08-01	-0.1350954	0.1649214	0.4127009
2016-09-01	-0.1353912	0.1831143	0.4596757
2016-10-01	-0.1388422	0.1498115	0.3540412
2016-11-01	-0.1406127	0.2128900	0.5089363
2016-12-01	-0.1432327	0.1861997	0.4417496
2017-01-01	-0.1454488	0.1752439	0.4065506

2017-02-01	-0.1550669	0.1976587	0.4327359
2017-03-01	-0.1578954	0.1814452	0.3841858
2017-04-01	-0.1591483	0.1896684	0.4014207
2017-05-01	-0.1648333	0.1985610	0.4064599
2017-06-01	-0.1647402	0.1770944	0.3522473
2017-07-01	-0.1662160	0.1974719	0.3999448
2017-08-01	-0.1682317	0.2058261	0.4137292
2017-09-01	-0.1707842	0.2068356	0.4089742
2017-10-01	-0.1744075	0.2158502	0.4190890
2017-11-01	-0.1704315	0.1970593	0.3871077
2017-12-01	-0.1687330	0.2015816	0.4025668
2018-01-01	-0.1511033	0.1553555	0.3307378
2018-02-01	-0.1407953	0.1826644	0.4408335
2018-03-01	-0.1373423	0.1711815	0.4223683
2018-04-01	-0.1315467	0.1912854	0.4916425
2018-05-01	-0.1332178	0.1998807	0.5051001
2018-06-01	-0.1301488	0.1652766	0.4310124
2018-07-01	-0.1311230	0.1587849	0.4089231
2018-08-01	-0.1278244	0.1764356	0.4687700
2018-09-01	-0.1219431	0.2108573	0.5630478
2018-10-01	-0.1233561	0.2247100	0.5830349
2018-11-01	-0.1216885	0.2181362	0.5769429
2018-12-01	-0.1231813	0.1987695	0.5354426
2019-01-01	-0.1239120	0.2084160	0.5521500
2019-02-01	-0.1150870	0.1906592	0.5460919
2019-03-01	-0.1207534	0.2149396	0.5742513
2019-04-01	-0.1210245	0.2263284	0.5928370
2019-05-01	-0.1279102	0.2276119	0.5741390
2019-06-01	-0.1320526	0.1943997	0.4969580
2019-07-01	-0.1377227	0.2459657	0.5755295
2019-08-01	-0.1354053	0.2253752	0.5479736
2019-09-01	-0.1310543	0.2298392	0.5685420
2019-10-01	-0.1360527	0.2301902	0.5544906
2019-11-01	-0.1247594	0.2465137	0.6127897
2019-12-01	-0.1190971	0.2720599	0.6615599
2020-01-01	-0.1145537	0.2717610	0.6733727
2020-02-01	-0.1166256	0.2014077	0.5625538
2020-03-01	-0.1269430	0.2554056	0.6191713
2020-04-01	-0.1322853	0.2603156	0.6113321
2020-05-01	-0.1517854	0.2810443	0.5891443
2020-06-01	-0.1579029	0.2732184	0.5633070
2020-07-01	-0.1493356	0.2412659	0.5359381
2020-08-01	-0.1369775	0.2521498	0.5869654
2020-09-01	-0.1147620	0.2621207	0.6615158
2020-10-01	-0.1100578	0.2807695	0.6950682
2020-11-01	-0.0643787	0.2624791	0.8062461
2020-12-01	-0.0592107	0.2320325	0.7985819
2021-01-01	0.0435573	0.1742916	0.8026565

```r
# Keep only one pseudo-treatment per month to prevent overplotting
pseudo_results_filtered <- pseudo_results_nested_df %>%
  mutate(month = lubridate::floor_date(Pseudo_Treatment_Date, "3 months")) %>%
  group_by(month) %>%
  slice(1) %>%
  ungroup()

pseudo_results_filtered <- pseudo_results_filtered %>%
  slice_head(n = nrow(.) - 1)

# Plot pseudo-treatment ATT over time
ggplot(pseudo_results_filtered, aes(x = month, y = ATT)) +
  geom_errorbar(aes(ymin = ATT - 1.96 * SE, ymax = ATT + 1.96 * SE), width = 20, color = "black") +
  geom_point(color = "blue") +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_vline(xintercept = att_bias_corrected, color = "red", linetype = "solid") +
  scale_x_date(
    date_breaks = "1 year",
    date_labels = "%Y") +
  labs(
    x = "Pseudo-Treatment Date",
    y = "Estimated ATT") +
  theme_minimal(base_family = "latexfont") +
  theme(axis.title.x = element_text(margin = margin(t = 15)))
```



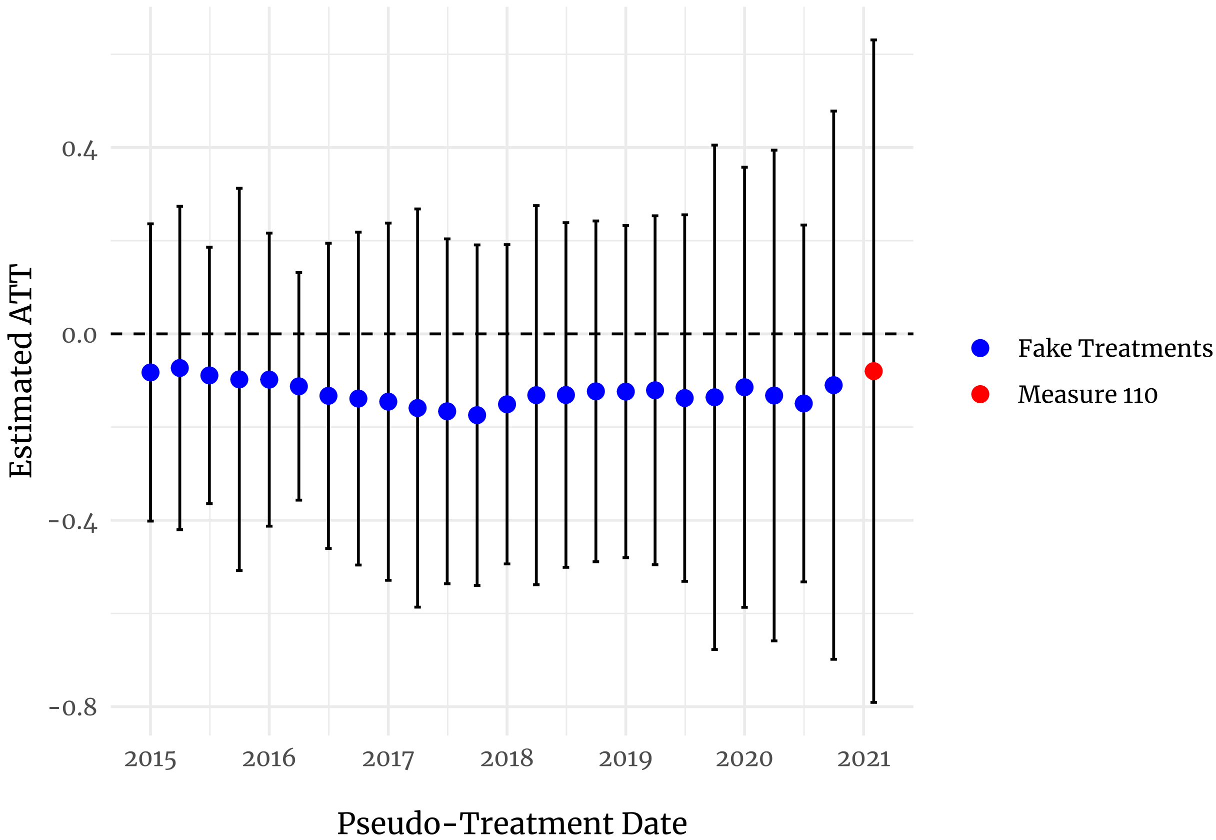

```r
# Report the proportion of pseudo-treatments with p < 0.05
n_significant_nested <- sum(pseudo_results_nested_df$p_value < 0.05, na.rm = TRUE)
```

```r
n_total_nested <- nrow(pseudo_results_nested_df)
cat("\nOut of", n_total_nested, "pseudo-treatments,", n_significant_nested,
    "had p < 0.05 (", round(100 * n_significant_nested/n_total_nested, 2), "% ).\n")
```

```
##
## Out of 803 pseudo-treatments, 0 had p < 0.05 ( 0 % ).
```

```r
# Save file to folder
ggsave("images/pseudo_treatment.pdf", width = 8, height = 4, dpi = 300)
```

```r
# PLOT PSEUDO TREATMENT RESULTS WITH OREGON

# Define the real treatment result
true_treatment_date <- as.Date("2021-02-01")
att_or <- -0.07996006
se_or <- 0.3626841

# Append Oregon result to filtered pseudo results
pseudo_results_filtered <- pseudo_results_filtered %>%
  mutate(is_real = FALSE) %>%
  bind_rows(
    data.frame(
      month = true_treatment_date,
      ATT = att_or,
      SE = se_or,
      is_real = TRUE))

# Plot pseudo-treatment ATT over time
ggplot(pseudo_results_filtered, aes(x = month, y = ATT, color = is_real)) +
  geom_errorbar(aes(ymin = ATT - 1.96 * SE, ymax = ATT + 1.96 * SE), width = 20, color = "black") +
  geom_point(size = 2.5) +
  scale_color_manual(
    values = c("FALSE" = "blue", "TRUE" = "red"),
    labels = c("FALSE" = "Fake Treatments", "TRUE" = "Measure 110"),
    name = NULL
  ) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  labs(
    x = "Pseudo-Treatment Date",
    y = "Estimated ATT",
  ) +
  theme_minimal(base_family = "latexfont") +
  theme(
    axis.title.x = element_text(margin = margin(t = 15)),
    legend.position = "right"  # you can also try "top", "bottom", or "left"
    )
```



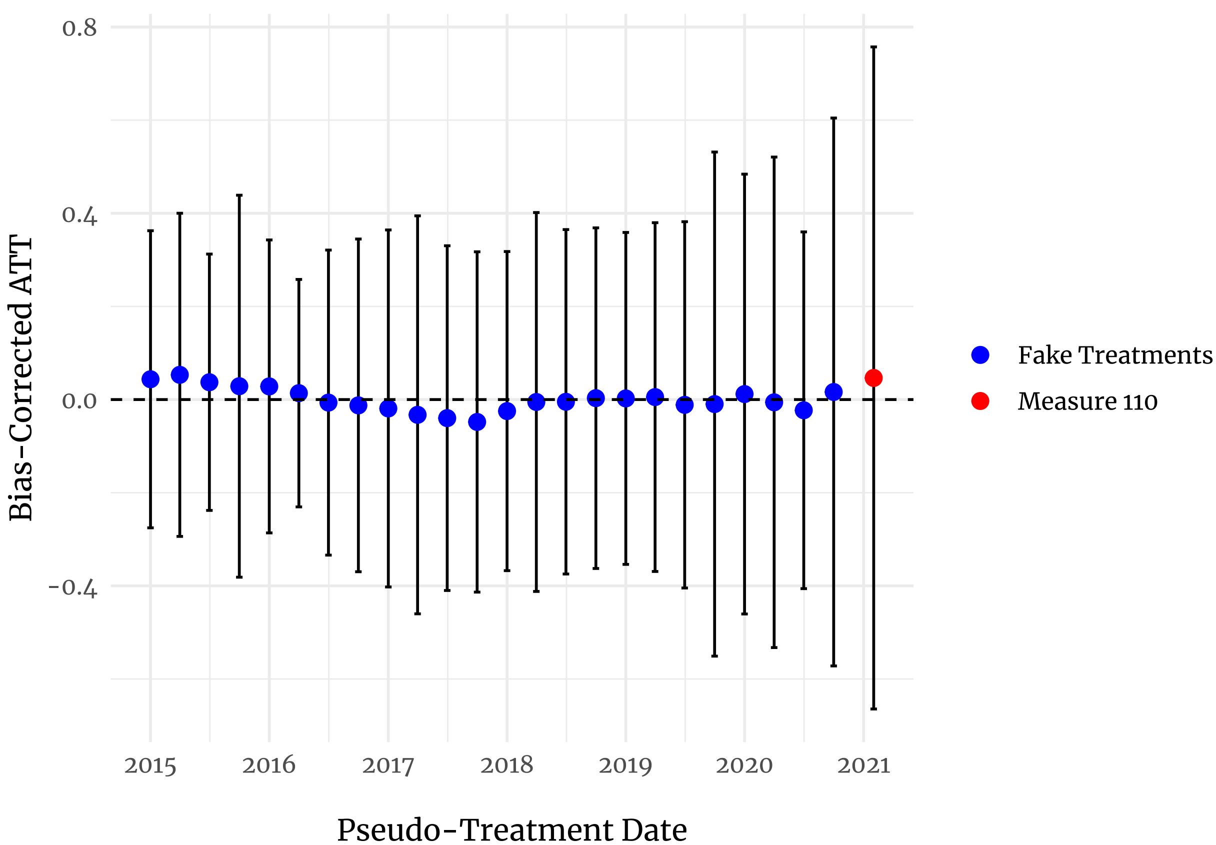

```
# Report the proportion of pseudo-treatments with p < 0.05
n_significant_nested <- sum(pseudo_results_nested_df$p_value < 0.05, na.rm = TRUE)
n_total_nested <- nrow(pseudo_results_nested_df)
cat("\nOut of", n_total_nested, "pseudo-treatments,", n_significant_nested,
    "had p < 0.05 (", round(100 * n_significant_nested/n_total_nested, 2), "% ).\n")
```

```
##
## Out of 803 pseudo-treatments, 0 had p < 0.05 ( 0 % ).
```

```
# Save file to folder
ggsave("images/pseudo_treatment.pdf", width = 8, height = 4, dpi = 300)
```

```
# PLOT PSEUDO TREATMENT RESULTS WITH OREGON, CORRECT FOR BIAS

# Compute average placebo ATT (drift)
bias_drift <- mean(pseudo_results_filtered$ATT[!pseudo_results_filtered$is_real], na.rm = TRUE)

# Apply correction to ATT and compute new y-axis values
pseudo_results_filtered <- pseudo_results_filtered %>%
  mutate(
    ATT_corrected = ATT - bias_drift,
    ymin_corrected = ATT_corrected - 1.96 * SE,
    ymax_corrected = ATT_corrected + 1.96 * SE)

# Plot pseudo-treatment ATT over time
ggplot(pseudo_results_filtered, aes(x = month, y = ATT_corrected, color = is_real)) +
  geom_errorbar(aes(ymin = ymin_corrected, ymax = ymax_corrected), width = 20, color = "black") +
  geom_point(size = 2.5) +
```

```
  scale_color_manual(
    values = c("FALSE" = "blue", "TRUE" = "red"),
    labels = c("FALSE" = "Fake Treatments", "TRUE" = "Measure 110"),
    name = NULL
  ) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  labs(
    x = "Pseudo-Treatment Date",
    y = "Bias-Corrected ATT",
  ) +
  theme_minimal(base_family = "latexfont") +
  theme(
    axis.title.x = element_text(margin = margin(t = 15)),
    legend.position = "right"
  )
```




```
# Report the proportion of pseudo-treatments with p < 0.05
n_significant_nested <- sum(pseudo_results_nested_df$p_value < 0.05, na.rm = TRUE)
n_total_nested <- nrow(pseudo_results_nested_df)
cat("\nOut of", n_total_nested, "pseudo-treatments,", n_significant_nested,
    "had p < 0.05 (", round(100 * n_significant_nested/n_total_nested, 2), "% ).\n")
```

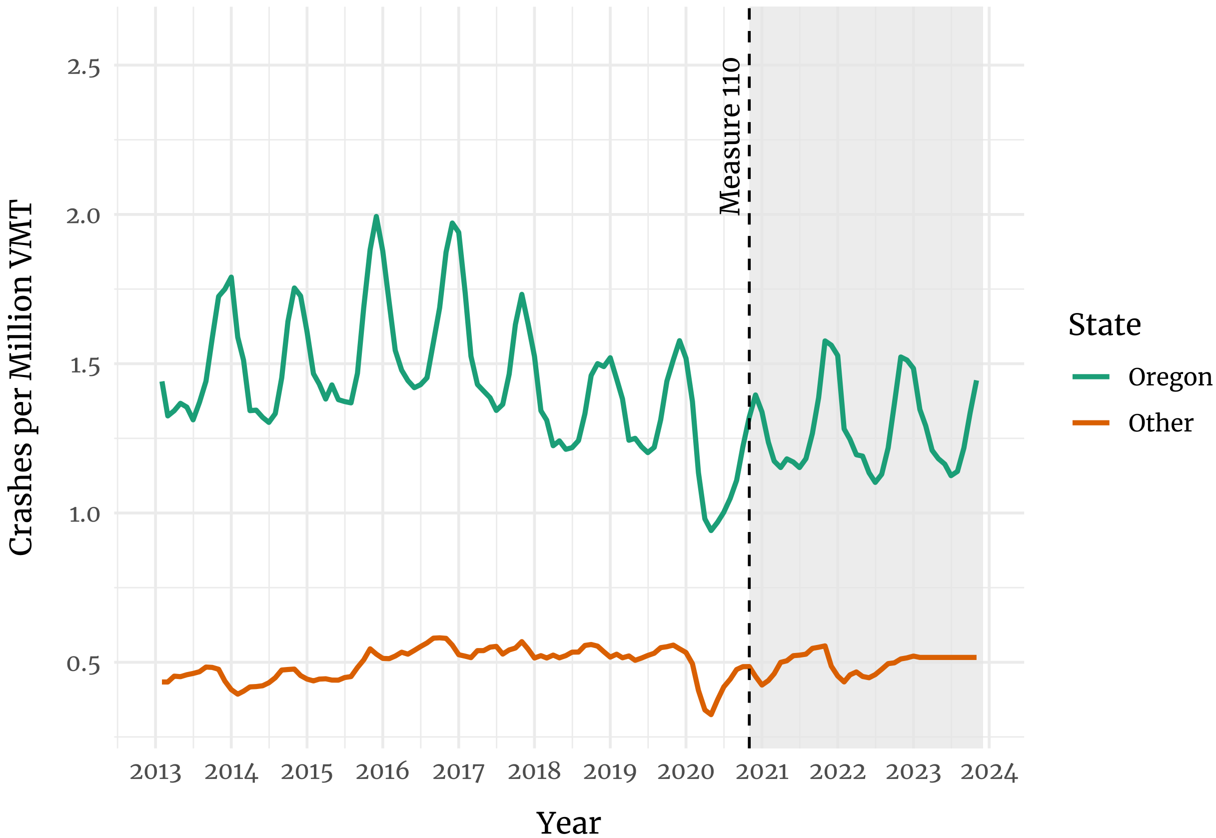
```
##
## Out of 803 pseudo-treatments, 0 had p < 0.05 ( 0 % ).
```

```
# Save file to folder
ggsave("images/pseudo_treatment_bias_corrected.pdf", width = 8, height = 4, dpi = 300)
```

```r
# PLOT DONOR STATES AGAINST OREGON

# Compute shared y-axis limits
df_ca <- df_wide_all %>% select(date, Oregon, California)
df_wy <- df_wide_all %>% select(date, Oregon, Wyoming)
df_wa <- df_wide_all %>% select(date, Oregon, Washington)

smooth_df <- function(df, other) {
  df %>%
    rename(Other = !!sym(other)) %>%
    pivot_longer(c("Oregon","Other"), names_to="series", values_to="crash_rate") %>%
    group_by(series) %>%
    mutate(smoothed_rate = zoo::rollmean(crash_rate, 3, fill=NA, align="center")) %>%
    ungroup()}

sm_ca <- smooth_df(df_ca, "California")
sm_wy <- smooth_df(df_wy, "Wyoming")
sm_wa <- smooth_df(df_wa, "Washington")

global_ylim <- range(
  c(sm_ca$smoothed_rate, sm_wy$smoothed_rate, sm_wa$smoothed_rate),
  na.rm = TRUE)
```

```r
# Oregon vs. California
ggplot(sm_ca, aes(date, smoothed_rate, color = series)) +
  annotate("rect",
    xmin = as.Date("2020-11-01"), xmax = max(sm_ca$date),
    ymin = -Inf, ymax = Inf, fill = "gray90", alpha = 0.7
  ) +
  geom_line(size = 0.9) +
  geom_vline(xintercept = as.Date("2020-11-01"), linetype = "dashed") +
  annotate("text",
    x = as.Date("2020-11-01"), y = global_ylim[2],
    label = "Measure 110", angle = 90, vjust = -0.5, hjust = 1.1,
    size = 3.5, family = "latexfont"
  ) +
  scale_y_continuous("Crashes per Million VMT", limits = global_ylim) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_color_manual(
    values = c("Oregon" = "#1b9e77", "Other" = "#d95f02"),
    labels = c("Oregon", "California", color = "State")
  ) +
  labs(x = "Year", color = "State") +
  theme_minimal(base_family = "latexfont") +
  theme(legend.position = "right",
        axis.title.x = element_text(margin = margin(t = 10)),
        axis.title.y = element_text(margin = margin(r = 10)))
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



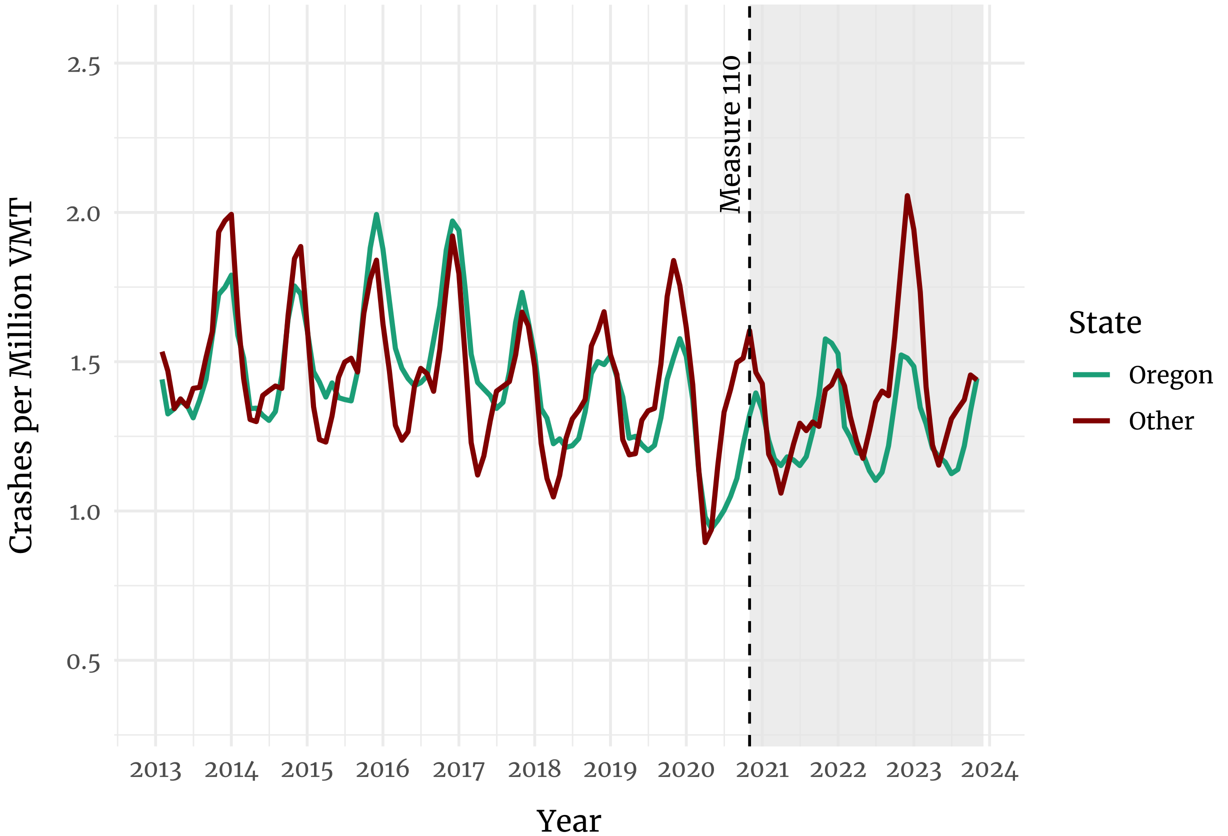

```r
ggsave("images/Oregon_vs_California.pdf", width = 6, height = 4, dpi = 300)
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```r
# Oregon vs. Wyoming
ggplot(sm_wy, aes(date, smoothed_rate, color = series)) +
  annotate("rect",
    xmin = as.Date("2020-11-01"), xmax = max(sm_wy$date),
    ymin = -Inf, ymax = Inf, fill = "gray90", alpha = 0.7
  ) +
  geom_line(size = 0.9) +
  geom_vline(xintercept = as.Date("2020-11-01"), linetype = "dashed") +
  annotate("text",
    x = as.Date("2020-11-01"), y = global_ylim[2],
    label = "Measure 110", angle = 90, vjust = -0.5, hjust = 1.1,
    size = 3.5, family = "latexfont"
  ) +
  scale_y_continuous("Crashes per Million VMT", limits = global_ylim) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_color_manual(
    values = c("Oregon" = "#1b9e77", "Other" = "#800000"),
    labels = c("Oregon", "Wyoming", color = "State")
  ) +
  labs(x = "Year", color = "State") +
  theme_minimal(base_family = "latexfont") +
  theme(legend.position = "right",
        axis.title.x = element_text(margin = margin(t = 10)),
```

```
      axis.title.y = element_text(margin = margin(r = 10)),
  plot.title = element_text(hjust = 0.5, size = 14, face = "bold"))
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



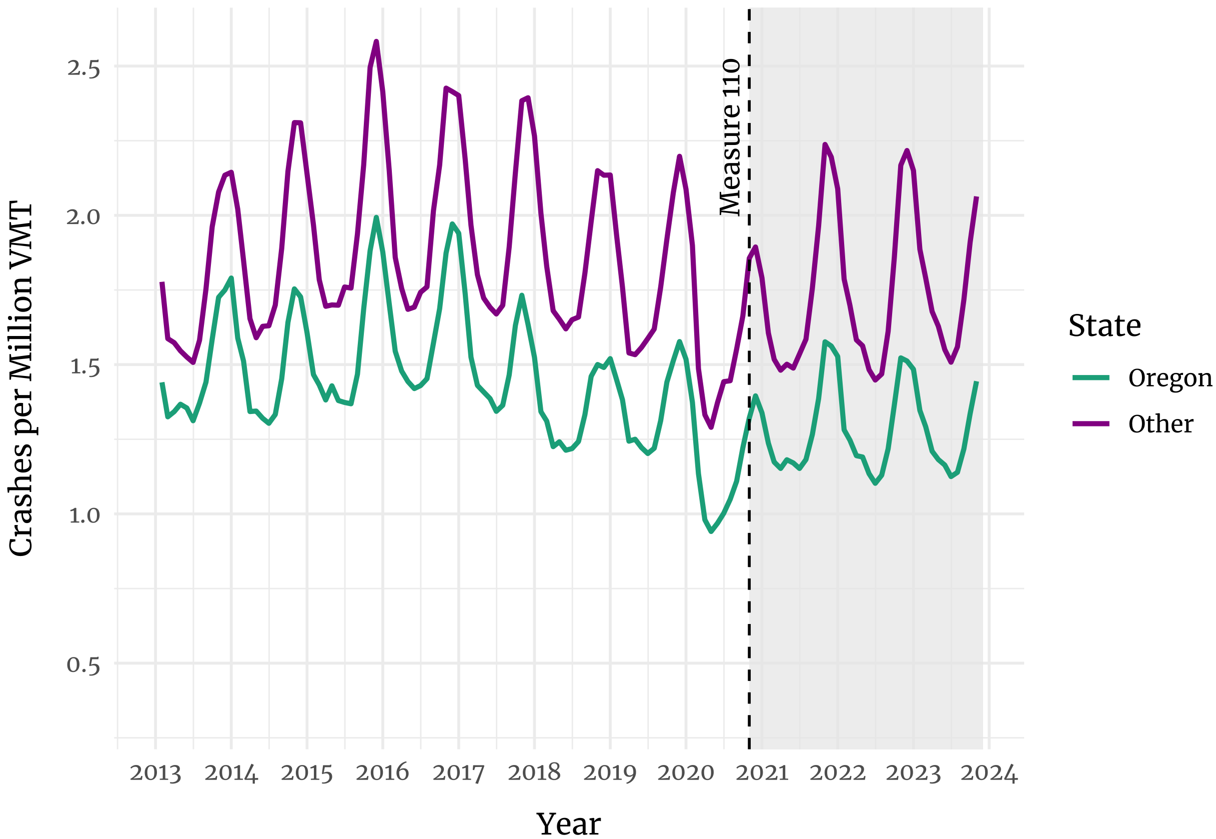

```
ggsave("images/Oregon_vs_Wyoming.pdf", width = 6, height = 4, dpi = 300)
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
# Oregon vs. Washington
ggplot(sm_wa, aes(date, smoothed_rate, color = series)) +
  annotate("rect",
    xmin = as.Date("2020-11-01"), xmax = max(sm_wa$date),
    ymin = -Inf, ymax = Inf, fill = "gray90", alpha = 0.7
  ) +
  geom_line(size = 0.9) +
  geom_vline(xintercept = as.Date("2020-11-01"), linetype = "dashed") +
  annotate("text",
    x = as.Date("2020-11-01"), y = global_ylim[2],
    label = "Measure 110", angle = 90, vjust = -0.5, hjust = 1.1,
    size = 3.5, family = "latexfont"
  ) +
  scale_y_continuous("Crashes per Million VMT", limits = global_ylim) +
  scale_x_date(date_breaks = "1 year", date_labels = "%Y") +
  scale_color_manual(
    values = c("Oregon" = "#1b9e77", "Other" = "#800080"),
```

```r
    labels = c("Oregon", "Washington", color = "State")
  ) +
  labs(x = "Year", color = "State") +
  theme_minimal(base_family = "latexfont") +
  theme(legend.position = "right",
        axis.title.x = element_text(margin = margin(t = 10)),
        axis.title.y = element_text(margin = margin(r = 10)))
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```r
ggsave("images/Oregon_vs_Washington.pdf", width = 6, height = 4, dpi = 300)
```

```
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```r
# REGRESSION ESTIMATE

# Run DID regression
lm.DID  <- lm(crash_rate ~ treated*post + vmt + month_num + pop_density + factor(state), data = df)
clustered_se <- vcovCL(lm.DID, cluster = ~state)
coeftest(lm.DID, vcov = clustered_se)
```

```
##
## t test of coefficients:
##
##                               Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)                 1.9522e+00  6.6338e-01  2.9429  0.003304 **
```

```
## treated                      -6.3107e-01  5.8426e-01 -1.0801   0.280271
## postTRUE                     -1.1608e-01  3.9157e-02 -2.9644   0.003083 **
## vmt                          -6.4267e-05  5.4155e-05 -1.1867   0.235538
## month_num                     2.9107e-02  5.3226e-03  5.4686 5.344e-08 ***
## pop_density                   1.7527e-04  8.3580e-03  0.0210   0.983272
## factor(state)California       1.0383e-01  2.0681e+00  0.0502   0.959968
## factor(state)Colorado         2.8119e-01  1.0754e-01  2.6147   0.009024 **
## factor(state)Idaho           -5.7660e-01  4.4719e-01 -1.2894   0.197470
## factor(state)North Carolina   7.6267e-01  4.5912e-01  1.6611   0.096902 .
## factor(state)Oregon           9.9692e-02  3.4569e-01  0.2884   0.773096
## factor(state)South Dakota     4.5305e-01  4.5397e-02  9.9797 < 2.2e-16 ***
## factor(state)Utah            -5.8953e-02  2.9091e-01 -0.2027   0.839436
## factor(state)Washington       2.8243e-02  3.5052e-01  0.0806   0.935791
## factor(state)Wisconsin        2.3468e-01  3.7564e-01  0.6248   0.532229
## treated:postTRUE              3.2211e-02  6.0684e-02  0.5308   0.595635
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# PLOT PSEUDO TREATMENT DISTRIBUTION

# Extract placebo ATTs and estimate bias
placebo_atts <- na.omit(pseudo_results_nested_df$ATT)
bin_width <- 0.01
n_placebos <- length(placebo_atts)

bias_estimate <- mean(placebo_atts)
att_bias_corrected <- att_or - bias_estimate

# Bias-correct placebo ATTs
placebo_atts_corrected <- placebo_atts - bias_estimate

# Calculate test p-value
p_value <- mean(abs(placebo_atts_corrected) >= abs(att_bias_corrected))

# Build histogram
hist_counts <- hist(placebo_atts_corrected, plot = FALSE, breaks = seq(
  min(placebo_atts_corrected) - bin_width,
  max(placebo_atts_corrected) + bin_width,
  by = bin_width))$counts

# Placebo ATT distribution dataframe
placebo_df <- data.frame(value = placebo_atts_corrected)

# Theme
publication_theme <- theme_minimal() +
  theme(
    text = element_text(family = "latexfont", color = "black"),
    plot.title = element_text(size = 11, face = "bold", hjust = 0),
    plot.subtitle = element_text(size = 10, hjust = 0, margin = margin(b = 10)),
    axis.title.x = element_text(size = 10, margin = margin(t = 10)),
    axis.title.y = element_text(size = 10, margin = margin(r = 10)),
    axis.text = element_text(size = 9, color = "black"),
    legend.position = "none",
    panel.grid.major = element_line(color = "gray90"),
```

```r
    panel.grid.minor = element_line(color = "gray95"),
    plot.margin = margin(20, 20, 20, 20),
    plot.caption = element_text(size = 8, hjust = 0, margin = margin(t = 15)),
    panel.border = element_blank(),
    axis.line = element_line(color = "black", size = 0.5))
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
# Main histogram + density + Oregon ATT
main_plot <- ggplot(placebo_df, aes(x = value)) +
  geom_histogram(
    binwidth = bin_width,
    fill = "gray80",
    color = "gray20",
    alpha = 0.7) +
  geom_density(
    aes(y = ..density.. * n_placebos * bin_width),
    color = "darkblue",
    size = 1.2,
    adjust = 1.7) +
  geom_vline(
    xintercept = att_bias_corrected,
    color = "red3",
    linetype = "solid",
    size = 0.9) +
  annotate("text",
    x = att_bias_corrected+0.037,
    y = max(hist_counts) * 0.85,
    label = "Oregon ATT",
    color = "red3",
    size = 3.2,
    hjust = 1,
    family = "latexfont") +
  labs(
    x = "Bias-Corrected ATT from Pseudo-Treatments",
    y = "Frequency") +
  publication_theme +
  scale_x_continuous(
    expand = c(0.02, 0.02),
    breaks = scales::pretty_breaks(n = 8)) +
  scale_y_continuous(
    expand = c(0, 0.1),
    breaks = scales::pretty_breaks(n = 6)) +
  coord_cartesian(ylim = c(0, max(hist_counts) * 1.2))

# Save + print
ggsave("images/synthetic_did_results.pdf", plot = main_plot, width = 8, height = 5, dpi = 300)
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
```

```
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```r
print(main_plot)
```