Final: County

Ricky Truong

August 2025

Background

Delete everything in environment

{r, message = FALSE, warning = FALSE} # rm(list = ls())

Load libraries

```
library(arrow)
library(CausalArima)
library(colorspace)
library(glue)
library(scales)
library(sf)
library(tidyverse)
library(tigris)
```

Load data for U.S. on-road CO2 emissions at census block group level[1]

```
# Create path to the .gdb directory
co2_path <- "CMS_DARTE_V2_1735/data/DARTE_v2.gdb"

# List the available layers (feature classes) inside the .gdb
st_layers(dsn = co2_path)
```

```
## Driver: OpenFileGDB
## Available layers:
##                            layer_name geometry_type features fields   crs_name
## 1 DARTE_v2_blockgroup_kgco2_1980_2017 Multi Polygon   220333     43 Vulcan_LCC
```

[1] https://daac.ornl.gov/CMS/guides/CMS_DARTE_V2.html

```
# Load specific layer
co2 <- st_read(dsn = co2_path, layer = "DARTE_v2_blockgroup_kgco2_1980_2017")
```

```
## Reading layer `DARTE_v2_blockgroup_kgco2_1980_2017' from data source
##   `/n/home10/rtruong/CMS_DARTE_V2_1735/data/DARTE_v2.gdb' using driver `OpenFileGDB'
## Simple feature collection with 220333 features and 43 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -7034829 ymin: -1957575 xmax: 3488418 ymax: 4595604
## Projected CRS: Vulcan_LCC
```

Load data for county shapefile (from `library(tigris)`)

```
# Load county shapefile (includes FIPS codes and geometry)
counties_sf <- counties(cb = TRUE, resolution = "20m", year = 2020) %>%
  mutate(fips = paste0(STATEFP, COUNTYFP))
```

```
##   |                                                                              |
```

```
# Filter for contiguous U.S.
contiguous_fips <- setdiff(sprintf("%02d", 1:56),
                           c("02", "15", "60", "66", "69", "72", "78"))
counties_sf <- counties_sf %>%
  filter(STATEFP %in% contiguous_fips)
```

Data wrangling for co2

Convert co2 to tidy format and aggregate (to county level)

```
# Drop geometry to speed up transformation
co2 <- st_drop_geometry(co2)

# Convert to tidy format
co2 <- co2 %>%
  pivot_longer(cols = starts_with("kgco2_"),
               names_to = "year",
               names_prefix = "kgco2_",
               values_to = "value") %>%
  mutate(year = as.integer(year))

# Add variables for state and county
co2 <- co2 %>%
  mutate(state = substring(GEOID, 1, 2),
         county = substring(GEOID, 3, 5))

# Identify counties where at least one block group has NA
na_counties <- co2 %>%
  filter(is.na(value)) %>%
  distinct(state, county)

# Remove all rows from counties that include any NA block group
co2 <- co2 %>%
  anti_join(na_counties, by = c("state", "county"))

# Aggregate by county
co2 <- co2 %>%
  group_by(state, county, year) %>%
  summarize(value = sum(value), .groups = "drop")

# Filter for years less than or equal to 2010 (to avoid confounding)
co2 <- co2 %>%
  filter(year <= 2010)

# Convert from kilograms to million metric tons (1 mmt = 1e9 kg)
co2 <- co2 %>%
  mutate(value = value / 1e9)

# Create variable for FIPS code
co2 <- co2 %>%
  mutate(fips = paste0(state, county)) %>%
  select(fips, state, county, year, value)
```

NOTE: BY THIS POINT, THERE ARE NO MISSING VALUES FOR co2, BUT THERE ARE A LOT OF 0S… TAKE A LOOK AT THIS LATER

Exporting

Export co2 (at county level)

```
write_csv(co2, "~/co2_county.csv")
```

Individual county-level analysis

Run CausalArima on specified county to estimate statistics

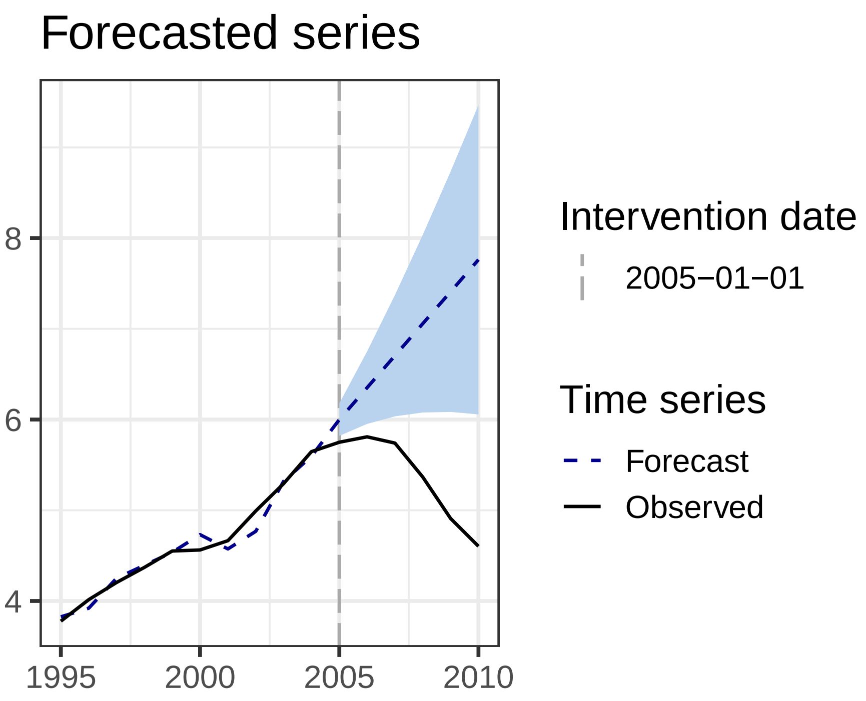
```r
# FIPS "34003" is Bergen County, NJ (very negative)

# Specify FIPS code for individual county
my_fips <- "34003"

# Get county_df for specified FIPS
county_df <- co2 %>%
  filter(fips == my_fips)

# Define intervention time point
intervention_date <- as.Date("2005-01-01")

# Create vector for dates
all_dates <- as.Date(paste0(county_df$year, "-01-01"))

# Create time series for outcome with yearly seasonality
y_ts <- ts(county_df$value, frequency = 1)

# Run CausalArima()
ce <- CausalArima(y = y_ts,
                  dates = all_dates,
                  int.date = intervention_date,
                  nboot = 1000)

# Get impact as list
imp <- impact(ce)

# Extract via impact_norm cumulative effect and other statistics
norm_effect <- imp$impact_norm$sum
cumulative_effect <- norm_effect$estimate
sd_norm <- norm_effect$sd
ci_lower_norm <- cumulative_effect - 1.96 * sd_norm
ci_upper_norm <- cumulative_effect + 1.96 * sd_norm

# Calculate relative cumulative effect as a decimal
post_years <- 2005:2010
post_indices <- which(county_df$year %in% post_years)
observed_post <- county_df$value[post_indices]
predicted_post <- ce$forecast
sum_obs <- sum(observed_post, na.rm = TRUE)
sum_pred <- sum(predicted_post, na.rm = TRUE)
rel_cumulative_effect <- (sum_obs - sum_pred) / sum_pred
```

```
# Extract via impact_boot cumulative effect and other statistics
boot_list <- imp$impact_boot
boot_effect <- boot_list$effect_cum[3, ]

# Store data as tibble
object <- tibble(cumulative_effect = cumulative_effect,
                 rel_cumulative_effect = rel_cumulative_effect,
                 sd_norm = sd_norm,
                 ci_lower_norm = ci_lower_norm,
                 ci_upper_norm = ci_upper_norm,
                 p_value_left_norm = norm_effect$p_value_left,
                 sd_boot = boot_effect$sd,
                 ci_lower_boot = boot_effect$inf,
                 ci_upper_boot = boot_effect$sup,
                 p_value_left_boot = as.numeric(boot_list$p_values["p"]))
object
```

```
## # A tibble: 1 x 10
##   cumulative_effect rel_cumulative_effect sd_norm ci_lower_norm ci_upper_norm
##               <dbl>                 <dbl>   <dbl>         <dbl>         <dbl>
## 1             -9.10                -0.221   0.223         -9.54         -8.67
## # i 5 more variables: p_value_left_norm <dbl>, sd_boot <dbl>,
## #   ci_lower_boot <dbl>, ci_upper_boot <dbl>, p_value_left_boot <dbl>
```

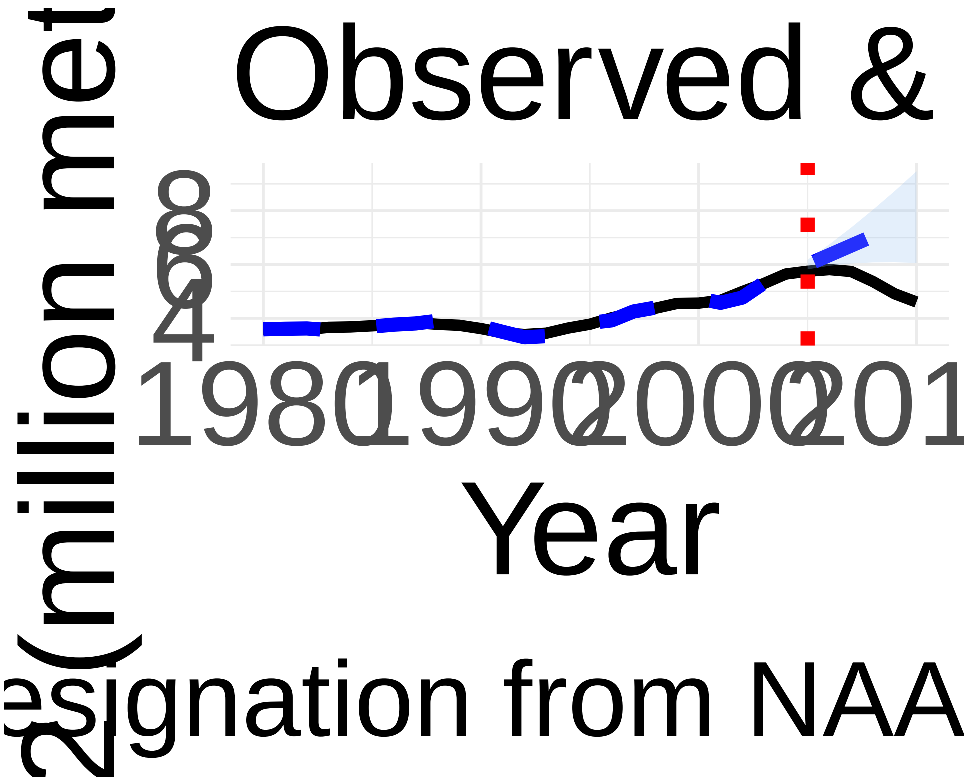
Visualize observed and counterfactual

```
# Plot counterfactual graph (default)
co2_plot_county_counterfactual_default <- plot(ce, type="forecast")
co2_plot_county_counterfactual_default
```



```
# Extract data used in ribbon layer of default plot (i.e., CI)
ribbon_data <- layer_data(co2_plot_county_counterfactual_default, 2)

# Convert CI to data frame
ci_df <- ribbon_data %>%
  transmute(date = as.Date(x, origin = "1970-01-01"),
            lower = ymin,
            upper = ymax)

# Extract forecasted values from ARIMA model and combine with CI
forecast_df <- tibble(date = as.Date(paste0(2005:2010, "-01-01")),
                      value = ce$forecast) %>%
  left_join(ci_df, by = "date")

# Extract fitted values from ARIMA model (before treatment)
fitted_df <- tibble(date = as.Date(paste0(county_df$year[county_df$year < 2005],
                                          "-01-01")),
                    value = as.numeric(fitted(ce$model)),
                    lower = NA,
                    upper = NA)

# Combine everything into one counterfactual data frame
counterfactual_df <- bind_rows(fitted_df, forecast_df)

# Extract observed values into data frame
obs_df <- county_df %>%
  mutate(date = as.Date(paste0(year, "-01-01")))
```

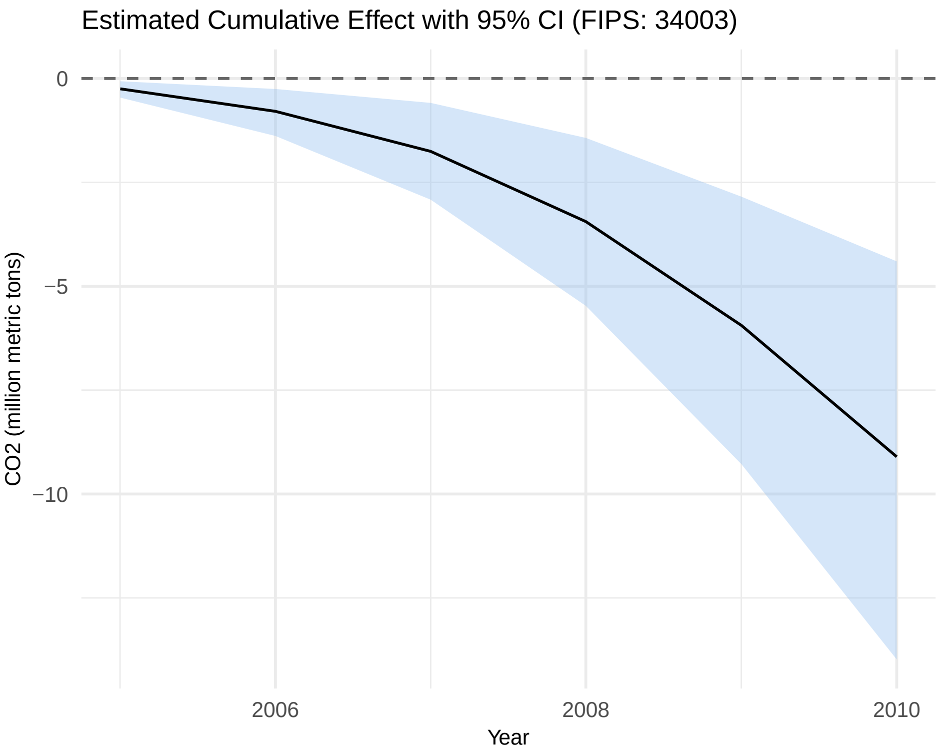
```r
# Create counterfactual plot (with extended domain and range)
co2_plot_county_counterfactual_extended <- ggplot() +
  geom_line(data = obs_df,
            aes(x = date, y = value),
            color = "black", linewidth = 2) +
  geom_line(data = counterfactual_df,
            aes(x = date, y = value, linetype = "Counterfactual"),
            color = "blue", linewidth = 2.5) +
  geom_ribbon(data = counterfactual_df,
              aes(x = date, ymin = lower, ymax = upper, fill = "95% CI"),
              alpha = 0.25) +
  geom_vline(aes(xintercept = as.Date("2005-01-01"),
                 linetype = "2005: Nonattainment Designation from NAAQS"),
             color = "red", linewidth = 2.5) +
  coord_cartesian(xlim = as.Date(c("1980-01-01", "2010-01-01"))) +
  labs(title = glue("Observed & Counterfactual CO2 Emissions (FIPS: {my_fips})"),
       x = "Year",
       y = "CO2 (million metric tons)",
       fill = "",
       linetype = "") +
  scale_linetype_manual(values = c("2005: Nonattainment Designation from NAAQS" = "dotted",
                                   "Counterfactual" = "dashed")) +
  scale_fill_manual(values = c("95% CI" = "#97C2F0")) +
  guides(linetype = guide_legend(order = 1),
         fill = guide_legend(order = 2)) +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 50),
        axis.text = element_text(size = 45),
        axis.title = element_text(size = 50),
        legend.text = element_text(size = 40))

# Plot and save counterfactual graph (with extended domain and range)
co2_plot_county_counterfactual_extended
```



```r
ggsave("co2_plot_county_counterfactual_extended.png",
       plot = co2_plot_county_counterfactual_extended,
       width = 20, height = 18, dpi = 300,
       bg = "white")
```

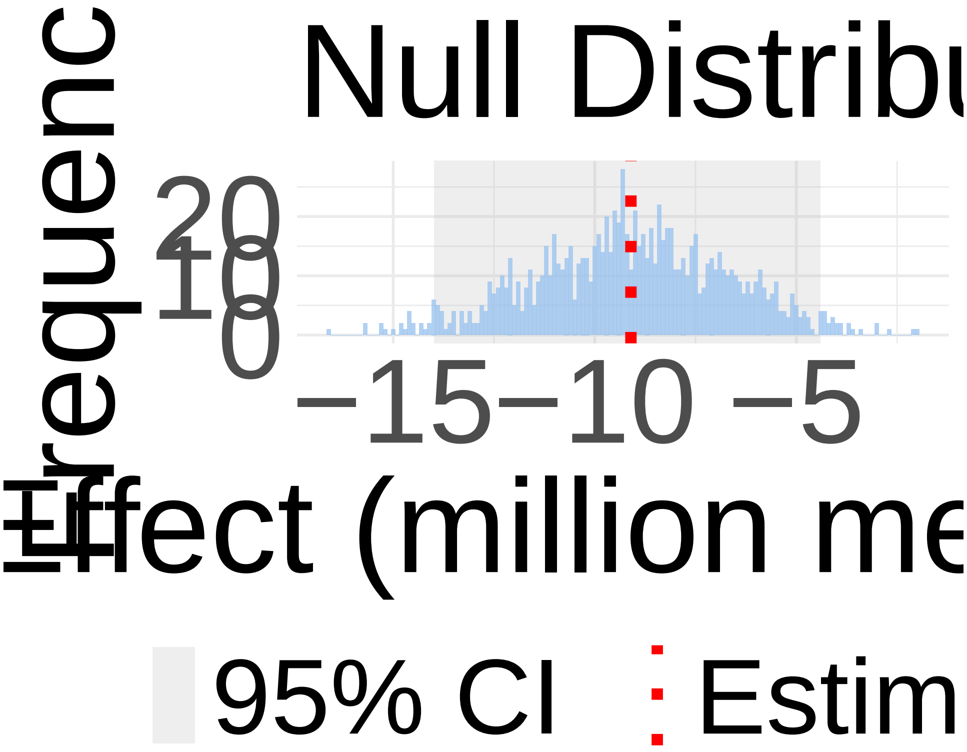
Visualize cumulative effect and null distribution

```r
# Define post-treatment period
post_years <- 2005:2010
post_dates <- as.Date(paste0(post_years, "-01-01"))

# Extract observed values for post-treatment years into data frame
observed_post <- county_df %>%
  filter(year %in% post_years) %>%
  mutate(date = as.Date(paste0(year, "-01-01")),
         obs = value)

# Extract forecasted values for post-treatment years into data frame
forecast_post <- tibble(date = post_dates,
                        pred = ce$forecast)

# Combine everything into one data frame
effect_df <- left_join(observed_post, forecast_post, by = "date") %>%
  mutate(diff = obs - pred,
         cum_effect = cumsum(diff))

# Get matrix of bootstrap forecasts
boot_matrix <- ce$boot$boot.distrib

# Repeat observed values to match dimensions
obs_vector <- effect_df$obs
obs_mat <- matrix(rep(obs_vector, times = ncol(boot_matrix)),
                  nrow = length(obs_vector), ncol = ncol(boot_matrix))

# Compute cumulative effect for each bootstrap sample
diff_mat <- obs_mat - boot_matrix
cum_mat <- apply(diff_mat, 2, cumsum)

# Extract confidence intervals into data frame
ci_df <- tibble(date = effect_df$date,
                lower = apply(cum_mat, 1, quantile, probs = 0.025),
                upper = apply(cum_mat, 1, quantile, probs = 0.975))

# Create plot (with confidence intervals)
co2_plot_county_cumulative <- ggplot(effect_df, aes(x = date, y = cum_effect)) +
  geom_ribbon(data = ci_df,
              aes(x = date, ymin = lower, ymax = upper),
              fill = "#97C2F0", alpha = 0.4,
              inherit.aes = FALSE) +
  geom_line() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "gray40") +
  labs(title = glue("Estimated Cumulative Effect with 95% CI (FIPS: {my_fips})"),
       x = "Year",
       y = "CO2 (million metric tons)") +
  theme_minimal() +
  theme(plot.title = element_text(size = 10),
        axis.title = element_text(size = 8),
        axis.text = element_text(size = 8))

# Plot and save graph
co2_plot_county_cumulative
```



```r
ggsave("co2_plot_county_cumulative.png",
       plot = co2_plot_county_cumulative,
       width = 8, height = 6, dpi = 300,
       bg = "white")

# Compute bootstrapped cumulative sums
boot_sums <- colSums((obs_mat - boot_matrix))

# Compute bounds for 95% CI
ci_bounds <- quantile(boot_sums, probs = c(0.025, 0.975))

# Calculate observed cumulative effect
obs_cum <- sum(effect_df$diff)

# Create data frame to shade CI (with label for legend)
ci_shade <- tibble(xmin = ci_bounds[1],
                   xmax = ci_bounds[2],
                   ymin = -Inf,
                   ymax = Inf,
                   fill_label = "95% CI")

# Create data frame for observed estimate/red line (with label for legend)
obs_line <- tibble(xintercept = obs_cum,
                   linetype_label = "Estimate")

# Create null distribution
```

```
co2_plot_county_null_dist <- ggplot(tibble(boot_sum = boot_sums), aes(x = boot_sum)) +
  geom_rect(data = ci_shade,
            aes(xmin = xmin, xmax = xmax, ymin = -Inf, ymax = Inf, fill = fill_label),
            alpha = 0.25, inherit.aes = FALSE) +
  geom_histogram(binwidth = 0.1, fill = "#97C2F0", alpha = 0.75) +
  geom_vline(data = obs_line,
             aes(xintercept = xintercept, linetype = linetype_label),
             color = "red", linewidth = 2,
             show.legend = c(linetype = TRUE, fill = FALSE)) +
  scale_fill_manual(values = c("95% CI" = "grey")) +
  scale_linetype_manual(values = c("Estimate" = "dotted")) +
  labs(title = glue("Null Distribution of Cumulative Effects (FIPS: {my_fips})"),
       x = "Cumulative Effect (million metric tons of CO2)",
       y = "Frequency",
       fill = "",
       linetype = "") +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 50),
        axis.text = element_text(size = 45),
        axis.title = element_text(size = 50),
        legend.text = element_text(size = 40))

# Plot and save graph
co2_plot_county_null_dist
```



```
ggsave("co2_plot_county_null_dist.png",
       plot = co2_plot_county_null_dist,
       width = 20, height = 18, dpi = 300,
       bg = "white")
```

Iteration (through each county)

Run run_causal_arima.R with source("run_causal_arima.R") in Console

After iteration

Load data from run_causal_arima.R

```
co2_county_causal_arima <- read_csv("co2_county_causal_arima.csv")
```

```
## Rows: 2820 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): fips
## dbl (10): cumulative_effect, rel_cumulative_effect, sd_norm, ci_lower_norm, ...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

Create map_data by adding polygons to co2_county_causal_arima

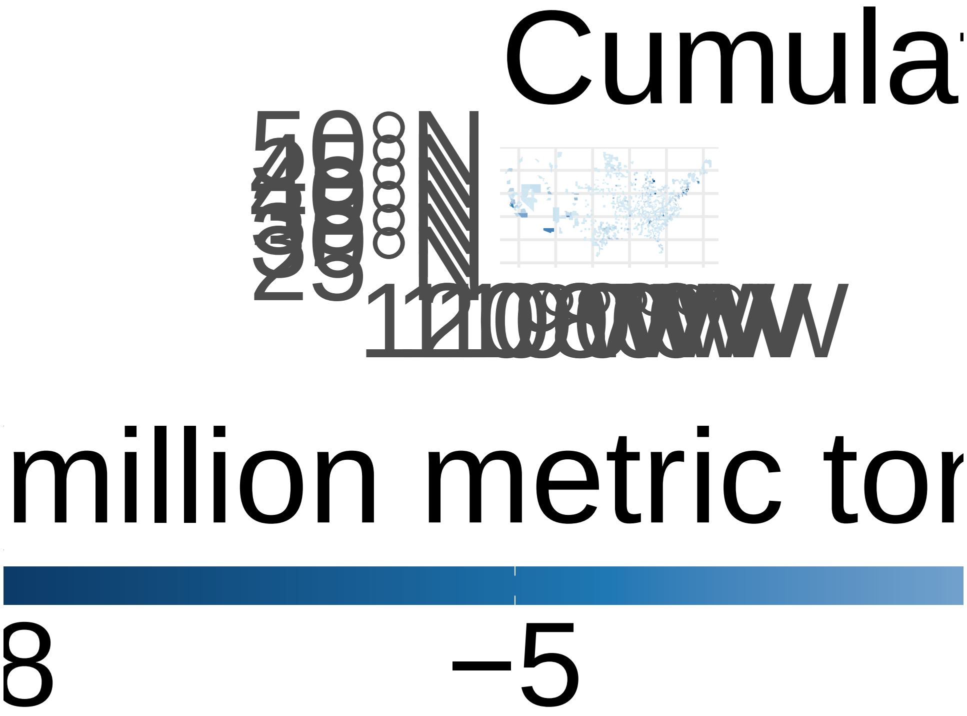
```
# Join shapefile with co2_county_causal_arima
map_data <- counties_sf %>%
 inner_join(co2_county_causal_arima, by = "fips")
```

Filter for significant counties with p_value_left < 0.05

```
sig_map_data <- map_data %>%
  filter(p_value_left_norm < 0.05, p_value_left_boot < 0.05)
```

Plot choropleth map of U.S. with cumulative effect at each county

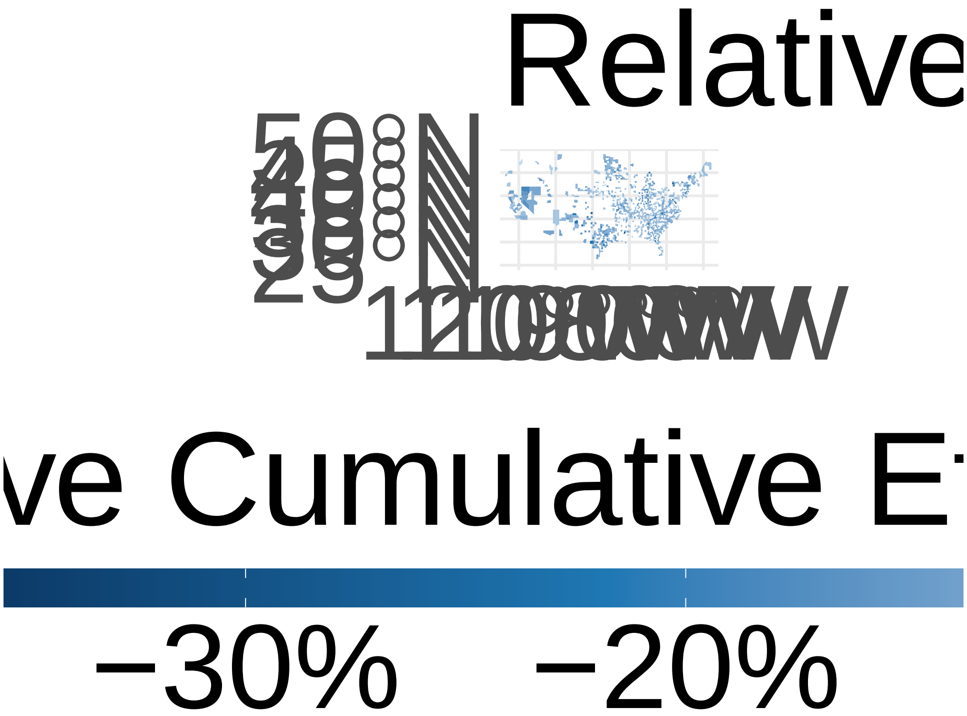
```
# Define sequential color palette
seq_colors <- c("#021B40", "#1F78B4", "#D6ECF5")

# Compute global limits for cumulative_effect
global_min <- min(sig_map_data$cumulative_effect, na.rm = TRUE)
global_max <- max(sig_map_data$cumulative_effect, na.rm = TRUE)

# Create graph of cumulative effects
sig_cumulative_effect_map <- ggplot(data = sig_map_data) +
  geom_sf(aes(fill = cumulative_effect), color = NA) +
  scale_fill_gradientn(colors = seq_colors,
                       na.value = "white",
                       limits = c(global_min, 0),
                       labels = label_number(accuracy = 1),
                       guide = guide_colorbar(barwidth = 50,
                                              barheight = 1,
                                              title.position = "top",
                                              title.vjust = 1)) +
  labs(title = glue("Cumulative Effect by County, 2006-2010: p < 0.05"),
       fill = "CO2 (million metric tons)") +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 50),
        axis.text = element_text(size = 40),
        legend.title = element_text(size = 50),
        legend.text = element_text(size = 45)) +
  coord_sf(xlim = c(-125, -66), ylim = c(24, 50), expand = FALSE)

# Plot and save graph
sig_cumulative_effect_map
```



```r
ggsave("sig_cumulative_effect_map.png",
       plot = sig_cumulative_effect_map,
       width = 20, height = 18, dpi = 300,
       bg = "white")
```

Plot choropleth map of U.S. with relative cumulative effect at each county

```r
# Define sequential color palette
seq_colors <- c("#021B40", "#1F78B4", "#D6ECF5")

# Compute global limits for rel_cumulative_effect
global_min <- min(sig_map_data$rel_cumulative_effect, na.rm = TRUE)
global_max <- max(sig_map_data$rel_cumulative_effect, na.rm = TRUE)

# Create graph of relative cumulative effects
sig_rel_cumulative_effect_map <- ggplot(data = sig_map_data) +
  geom_sf(aes(fill = rel_cumulative_effect), color = NA) +
  scale_fill_gradientn(colors = seq_colors,
                       na.value = "white",
                       limits = c(global_min, 0),
                       labels = scales::label_percent(scale = 100),
                       guide = guide_colorbar(barwidth = 50,
                                              barheight = 1,
                                              title.position = "top",
                                              title.vjust = 1)) +
  labs(title = glue("Relative Cumulative Effect by County, 2006-2010: p < 0.05"),
       fill = "Relative Cumulative Effect (%)") +
  theme_minimal() +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 50),
        axis.text = element_text(size = 40),
        legend.title = element_text(size = 50),
        legend.text = element_text(size = 45)) +
  coord_sf(xlim = c(-125, -66), ylim = c(24, 50), expand = FALSE)

# Plot and save graph
sig_rel_cumulative_effect_map
```



```
ggsave("sig_rel_cumulative_effect_map.png",
       plot = sig_rel_cumulative_effect_map,
       width = 20, height = 18, dpi = 300,
       bg = "white")
```