Preliminary Assignment

Ricky Truong

March 2025

Libraries and data frames

```r
# Clear environment
# rm(list = ls())

# Load CausalArima
# install.packages("devtools")
# install.packages("tidybayes")
# devtools::install_github("FMenchetti/CausalArima")
library(CausalArima)
```

```
## Loading required package: forecast
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Loading required package: ggplot2
```

```
## Loading required package: gridExtra
```

```
## Loading required package: kableExtra
```

```r
# Load libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.4     v tibble    3.2.1
## v purrr     1.0.4     v tidyr     1.3.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::combine()    masks gridExtra::combine()
## x dplyr::filter()     masks stats::filter()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()        masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to bec
```

```r
library(readxl)
library(readr)
```

```r
# Load data frames
GDP <- read_delim("GDP.csv", delim = ";",
                  escape_double = FALSE, trim_ws = TRUE)
```

```
## Rows: 60 Columns: 28
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr  (1): GeoName
## dbl (27): 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, ...
##
## 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.
```

```r
CO2 <- read_excel("CO2.xlsx")
```

```
## New names:
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
## * `` -> `...7`
## * `` -> `...8`
## * `` -> `...9`
## * `` -> `...10`
## * `` -> `...11`
## * `` -> `...12`
## * `` -> `...13`
## * `` -> `...14`
## * `` -> `...15`
## * `` -> `...16`
## * `` -> `...17`
## * `` -> `...18`
## * `` -> `...19`
## * `` -> `...20`
## * `` -> `...21`
## * `` -> `...22`
## * `` -> `...23`
## * `` -> `...24`
## * `` -> `...25`
## * `` -> `...26`
## * `` -> `...27`
## * `` -> `...28`
## * `` -> `...29`
## * `` -> `...30`
## * `` -> `...31`
## * `` -> `...32`
## * `` -> `...33`
## * `` -> `...34`
## * `` -> `...35`
## * `` -> `...36`
## * `` -> `...37`
## * `` -> `...38`
## * `` -> `...39`
## * `` -> `...40`
## * `` -> `...41`
## * `` -> `...42`
## * `` -> `...43`
## * `` -> `...44`
## * `` -> `...45`
## * `` -> `...46`
## * `` -> `...47`
## * `` -> `...48`
## * `` -> `...49`
## * `` -> `...50`
## * `` -> `...51`
## * `` -> `...52`
## * `` -> `...53`
## * `` -> `...54`
## * `` -> `...55`
## * `` -> `...56`
## * `` -> `...57`
## * `` -> `...58`
```

Part 0: Wrangle data

```r
# Delete unnecessary rows in CO2 data frame
CO2 <- CO2[-c(1, 2, 3, 4, 57),]

# Define values for CO2
years <- seq(1970, 2022)
newnames <- paste0(years, "_CO2")
newnames <- append(newnames, "state", 0)
newnames <- append(newnames, "1970_2022_CO2_percent_change", 54)
newnames <- append(newnames, "1970_2022_CO2_absolute_change", 55)
newnames <- append(newnames, "2021_2022_CO2_percent_change", 56)
newnames <- append(newnames, "2021_2022_CO2_absolute_change", 57)

# Rename columns of CO2 using names()
names(CO2) <- newnames

# Define values for GDP
```

```r
years <- seq(1997, 2023)
newnames <- paste0(years, "_GDP")
newnames <- append(newnames, "geo_name", 0)

# Rename columns of GDP using names()
names(GDP) <- newnames
```

Part 1: Combine data frames

```r
# Combine data frames via inner_join()
states_inner <- inner_join(CO2, GDP, join_by("state" == "geo_name"))

# Combine data frames via full_join()
states_full <- full_join(CO2, GDP, join_by("state" == "geo_name"))

# Combine data frames via left_join(), with CO2 as the left-hand data frame
states_left <- left_join(CO2, GDP, join_by("state" == "geo_name"))
```

Part 1 Alternative: Make our own data frame in Tidy format

I want a Tidy data set where the rows are years, there's a column for year,

a column for CO2_{state}, and a column for GDP_{state}

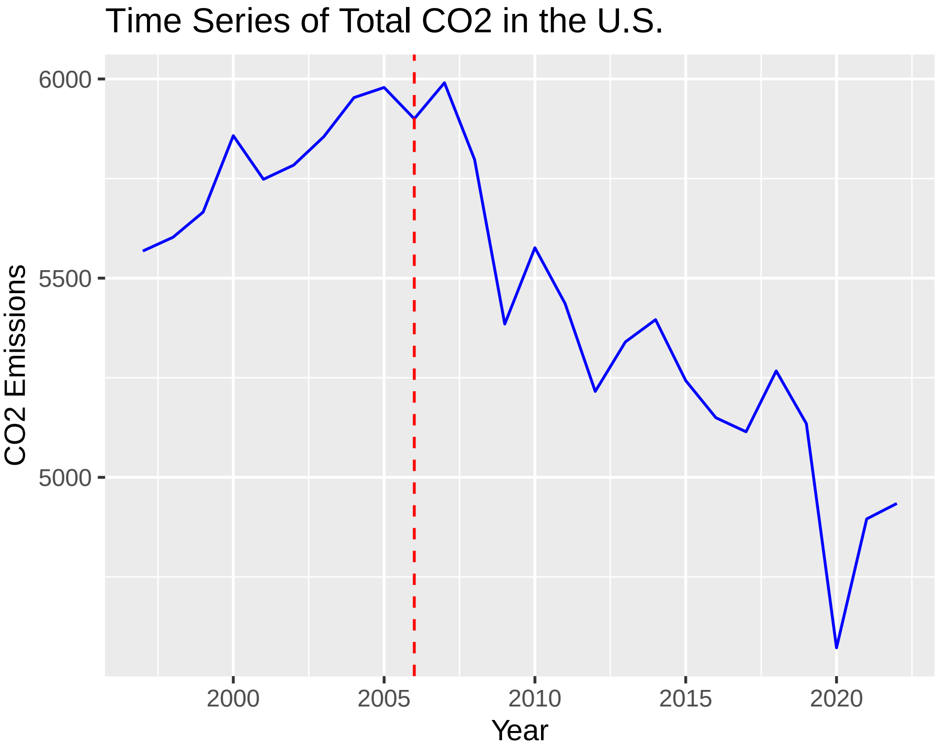
For now, I will create a data set, new, with only totals and years 1997-2022

```r
# Tranpose CO2
CO2_alt <- CO2 %>%
  t()

# Delete unnecessary rows in CO2 (so that only years 1997-2022 are included)
CO2_alt <- CO2_alt[-c(seq(1, 28), seq(55, 58)),]

# Store total CO2 of states as vector of doubles
total_CO2 <- as.double(CO2_alt[,52])

# Tranpose GDP
GDP_alt <- GDP %>%
  t()

# Delete unnecessary rows in GDP (so that only years 1997-2022 are included)
GDP_alt <- GDP_alt[-c(1, 28),]
```

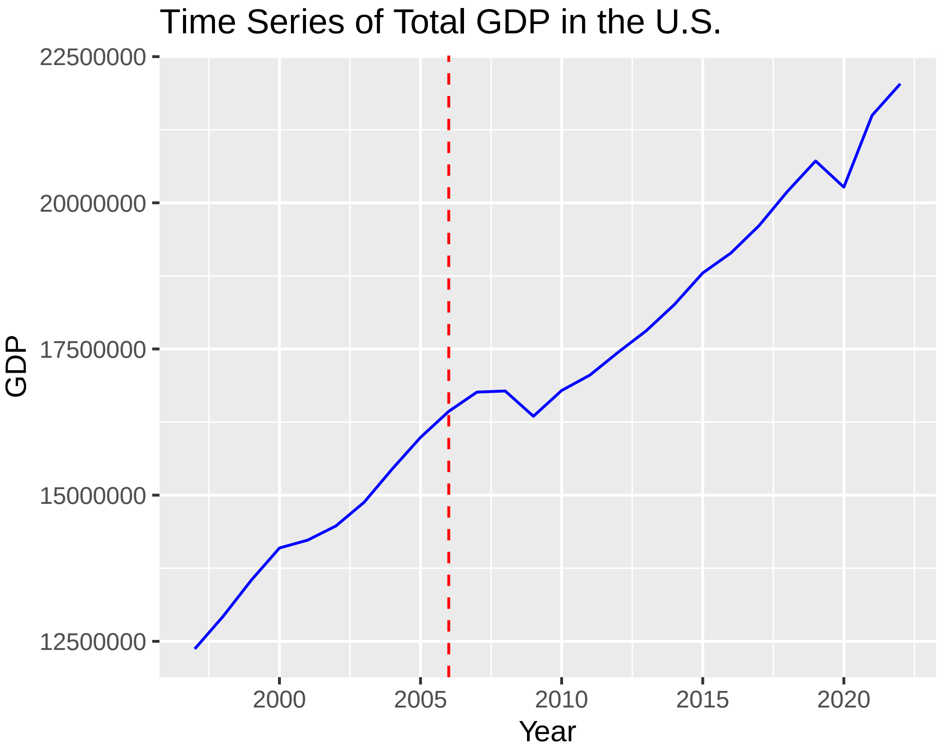
```r
# Store total GDP of states as vector of doubles
total_GDP <- as.double(GDP_alt[,1])

# Create a new data frame manually
new <- data.frame("year" = seq(1997, 2022),
                  "total_CO2" = total_CO2,
                  "total_GDP" = total_GDP)

# Rename indices
rownames(new) <- NULL
```

Part 1.5 Alternative: Visualize data

```r
# Graph for CO2
ggplot(data = new, mapping = aes(x = year,
                                 y = total_CO2)) +
  geom_line(color = "blue") +
  geom_vline(xintercept = 2006, linetype = "dashed", color = "red") +
  labs(title = "Time Series of Total CO2 in the U.S.",
       x = "Year",
       y = "CO2 Emissions")
```



```r
# Graph for GDP
ggplot(data = new, mapping = aes(x = year,
                                 y = total_GDP)) +
  geom_line(color = "blue") +
  geom_vline(xintercept = 2006, linetype = "dashed", color = "red") +
  labs(title = "Time Series of Total GDP in the U.S.",
       x = "Year",
       y = "GDP")
```



Part 1.75 Alternative: Wrangle data

```
# Create a vector of years in class Date
# For consistency, we'll use the first day of each year (e.g., 2006-01-01)
years_dates <- paste0(seq(1997, 2022), "-01-01")
```

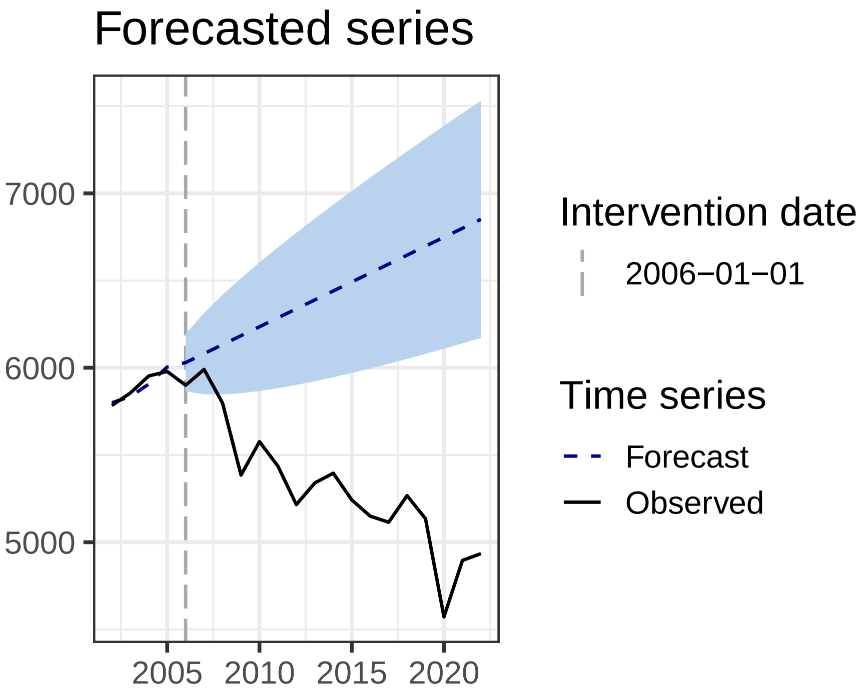
Part 2: Use CausalArima (WITHOUT regressor)

Template code from https://github.com/RobsonTigre/CausalTimeSeries/blob/main/1%20-%20CausalImpact%20and%20CausalArima.R

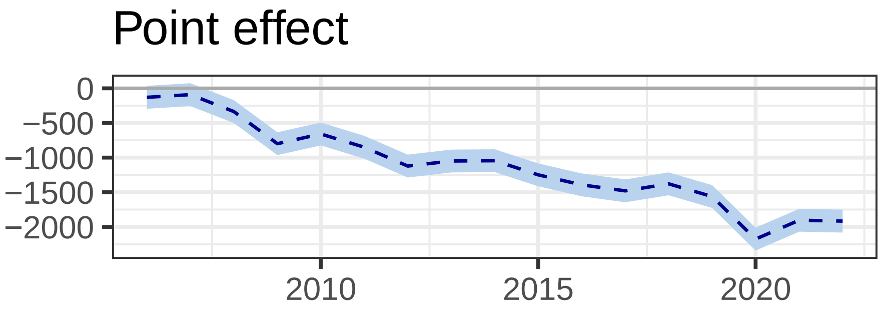
and https://github.com/FMenchetti/CausalArima?tab=readme-ov-file

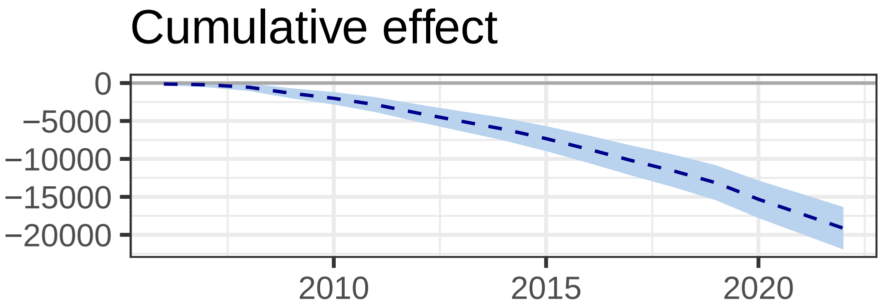
```
# EN-US: Define the intervention timepoint (day when the intervention occurs)
# Intervention is 2006, which is 9 years (~3285 days) from start of 1997
```

```
intervention_time <- 3285

# EN-US: Create a time series object for y with YEARLY seasonality
CO2_ts <- ts(new$total_CO2, frequency = 1)

# EN-US: Define pre-intervention and post-intervention indices
# Pre-Period: Days 1 to 3285 (before the intervention).
pre_period <- 1:(intervention_time - 1)
# Post-Period: Days 3285 to 6205 (after the intervention).
post_period <- intervention_time:length(CO2_ts)

# EN-US: Set up the parameters for CausalArima's int.date (the intervention date)
# and dates (the full date sequence) arguments
intervention_date <- as.Date("2006-01-01")
all_dates <- as.Date(years_dates)

# EN-US: Fit the CausalArima model for causal effect estimation
ce <- CausalArima(y = ts(new$total_CO2, frequency = 1),
                  dates = all_dates, int.date = intervention_date,
                   nboot = 1000)

# EN-US: Plot the impact
forecasted <- plot(ce, type = "forecast")
print(forecasted)
```



```
impact_p <-plot(ce, type = "impact")
grid.arrange(impact_p$plot, impact_p$cumulative_plot)
```





```r
# EN-US: Display the normalized impact
summary_model <- impact(ce)
summary_model$arima
```

```
## $arima_order
##             p d q
## arima_order 0 1 0
##
## $param
##           coef       se  t value
## drift 51.31535 27.82266 1.844372
##
## $accuracy
##                     ME     RMSE     MAE         MPE      MAPE      MASE
## Training set 0.6129757 74.21602 49.3432 0.003713157 0.8489406 0.6269587
##                    ACF1
## Training set -0.3889488
##
## $log_stats
##            loglik      aic      bic     aicc
## metrics -46.27602 96.55205 96.71093 98.95205
```

```r
summary_model$impact_norm
```

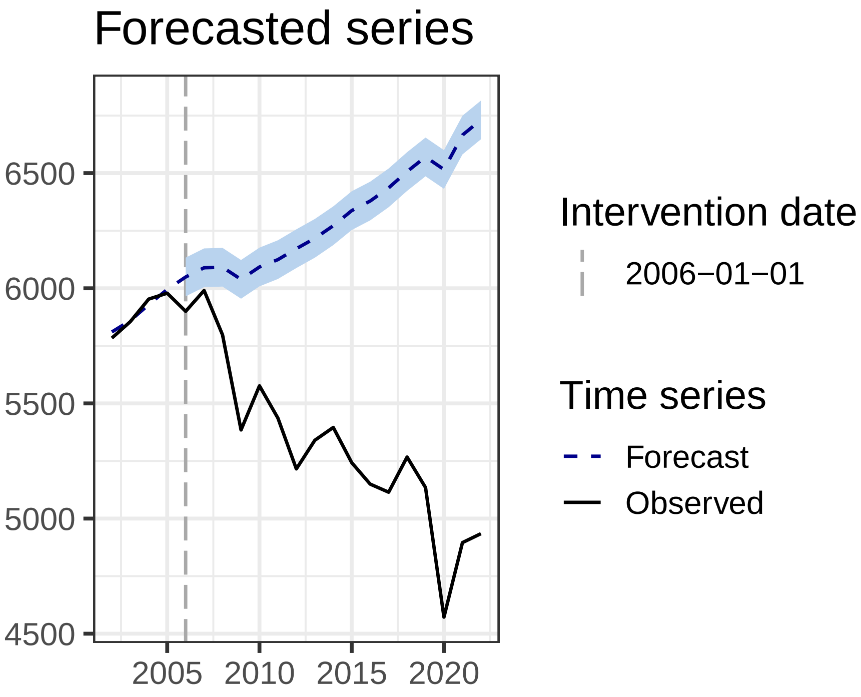
```
## $average
##    estimate       sd p_value_left p_value_bidirectional p_value_right
## 1 -1125.884 20.41011            0                     0             1
##
## $sum
##    estimate      sd p_value_left p_value_bidirectional p_value_right
## 1 -19140.02 346.972            0                     0             1
##
## $point_effect
##    estimate       sd  p_value_left p_value_bidirectional p_value_right
## 1 -1916.575 84.15306 4.068642e-115                     0             1
```

Part 3: Use CausalArima (WITH regressor)

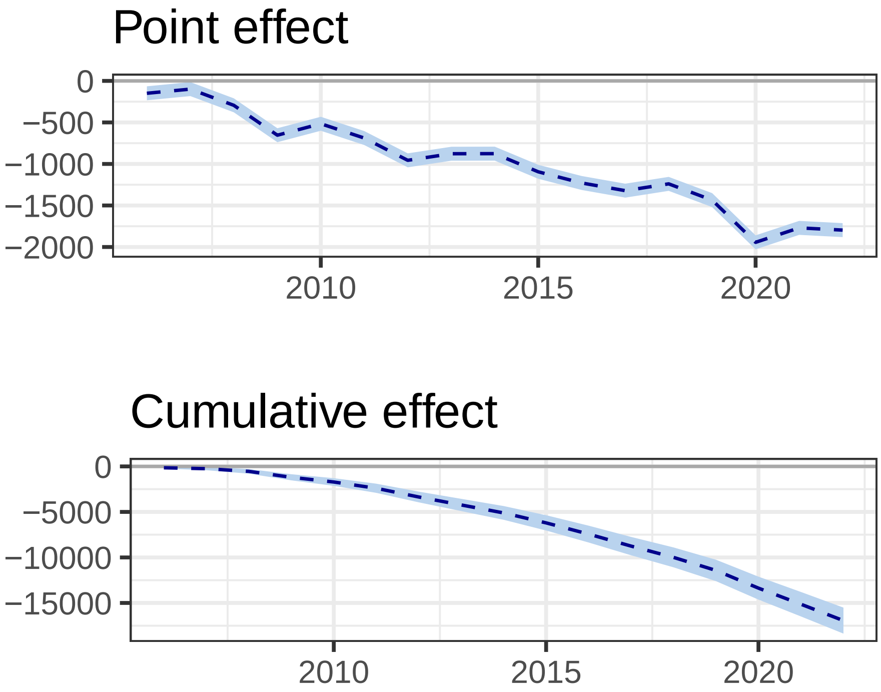
Template code from https://github.com/RobsonTigre/CausalTimeSeries/blob/main/1%20-%20CausalImpact%20and%20CausalArima.R

and https://github.com/FMenchetti/CausalArima?tab=readme-ov-file

```r
# EN-US: Define the intervention timepoint (day when the intervention occurs)
# Intervention is 2006, which is 9 years (~3285 days) from start of 1997
intervention_time <- 3285

# EN-US: Create a time series object for y with YEARLY seasonality
CO2_ts <- ts(new$total_CO2, frequency = 1)

# EN-US: Define pre-intervention and post-intervention indices
# Pre-Period: Days 1 to 3285 (before the intervention).
pre_period <- 1:(intervention_time - 1)
# Post-Period: Days 3285 to 6205 (after the intervention).
post_period <- intervention_time:length(CO2_ts)

# EN-US: Set up the parameters for CausalArima's int.date (the intervention date)
# and dates (the full date sequence) arguments
intervention_date <- as.Date("2006-01-01")
all_dates <- as.Date(years_dates)

# EN-US: Fit the CausalArima model for causal effect estimation
ce <- CausalArima(y = ts(new$total_CO2, frequency = 1),
                  dates = all_dates, int.date = intervention_date,
                  xreg = total_GDP, nboot = 1000)

# EN-US: Plot the impact
forecasted <- plot(ce, type = "forecast")
print(forecasted)
```



```r
impact_p <-plot(ce, type = "impact")
grid.arrange(impact_p$plot, impact_p$cumulative_plot)
```



```r
# EN-US: Display the normalized impact
summary_model <- impact(ce)
summary_model$arima
```

```
## $arima_order
##              p d q
## arima_order 0 0 0
##
## $param
##                   coef           se    t value
## intercept 4.048319e+03 1.647628e+02 24.5705808
## xreg      1.217476e-04 3.335336e-04  0.3650237
##
## $accuracy
##                          ME     RMSE      MAE          MPE      MAPE      MASE
## Training set -9.094947e-13 37.80702 29.05976 -0.004268842 0.5016234 0.3692357
##                    ACF1
## Training set -0.3993017
##
## $log_stats
##           loglik     aic      bic     aicc
## metrics -45.4629 96.9258 97.51747 101.7258
```

```
summary_model$impact_norm
```

```
## $average
##    estimate       sd p_value_left p_value_bidirectional p_value_right
## 1 -996.4233 10.39729            0                     0             1
##
## $sum
##   estimate      sd p_value_left p_value_bidirectional p_value_right
## 1 -16939.2 176.754            0                     0             1
##
## $point_effect
##    estimate       sd p_value_left p_value_bidirectional p_value_right
## 1 -1796.599 42.86913            0                     0             1
```

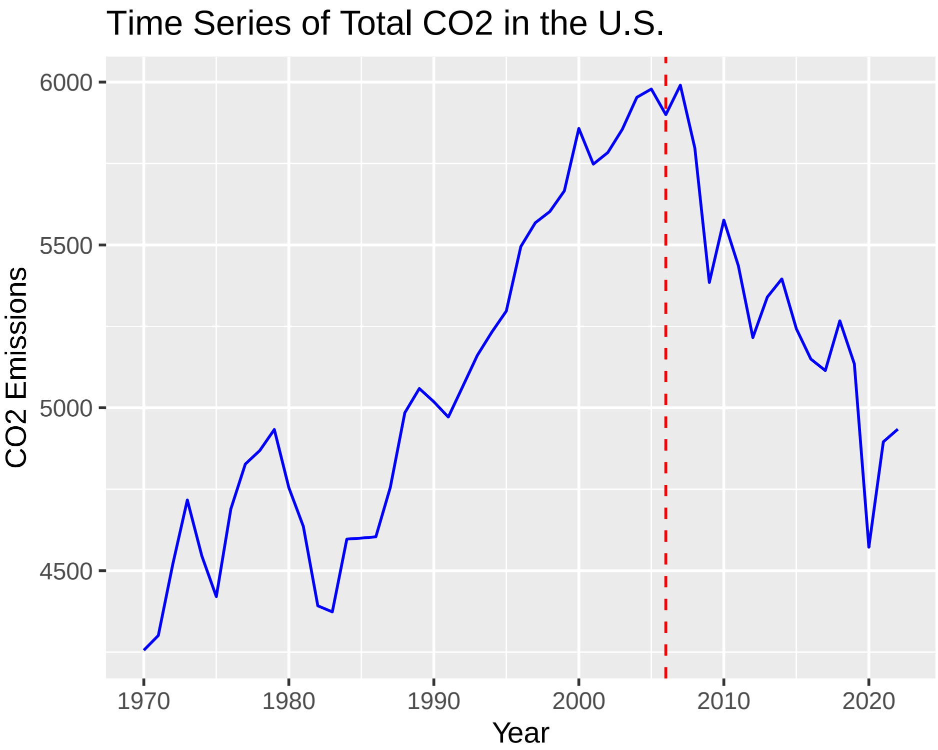
Part 4: Comment on results

I'm very happy with the result of this project. Though I wasn't very familiar with times series or causal inference at the start of this (conceptually and code-wise), I feel I better understand it now; from what I understand, we try to extrapolate the time series data (before the intervention) into the future to see what "would've been" if there wasn't any intervention, and we compare this "counterfactual" to our observed data from after the intervention. Just from that intuition and the graph from Part 2, I'm sure anyone can see there appears to be evidence toward a causal impact from the 2006 revision, both with and without GDP as a regressor (to be honest, I feel I don't fully understand this part and its significance).

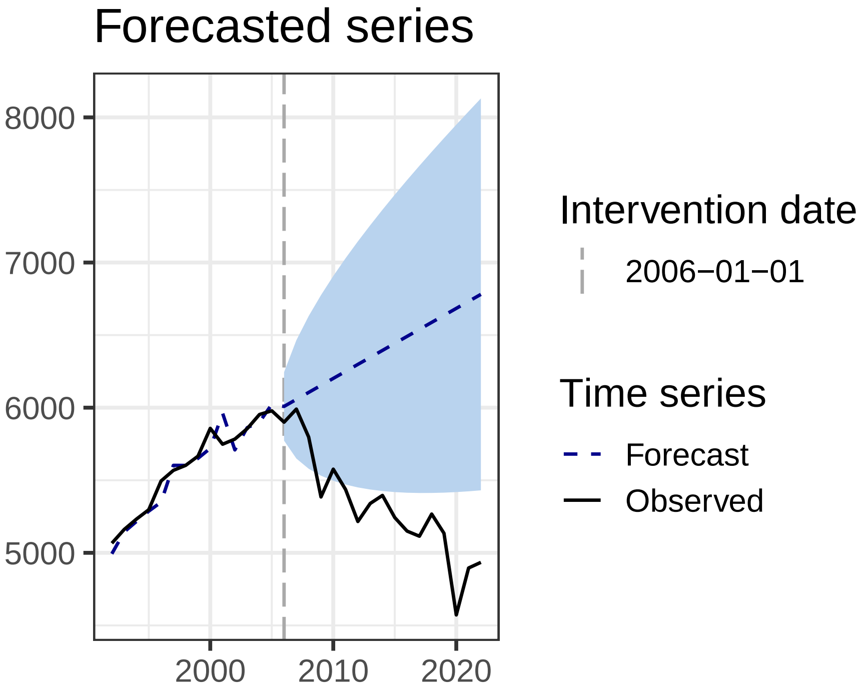
I'm also happy because I challenged myself to not use any GenAI, which I feel helped me with my coding/problem-solving skills; I instead relied on a lot of helpful resources I found on Google and YouTube. The only GenAI I used was ChatGPT near the beginning to install the CausalArima library (this is elaborated further on the email). Not using GenAI was definitely a challenge given that the datasets were not in the Tidy format I'm used to, so I had to get a bit creative with some of the data wrangling. Related, for possible improvements in the future, I'd love to consider individual states/regions rather than the entire U.S. for analysis, which would require the "new" data set to contain more information (as outlined in Part 1 Alternative). Also, I'd love to use GenAI to polish the code as I'm sure I wasn't being the most efficient in some of my code.

Part 5: Use CausalArima (WITHOUT regressor and WITH more years)

```r
### DATA WRANGLING:
# Tranpose CO2
CO2_updated <- CO2 %>%
  t()

# Delete unnecessary rows in CO2
CO2_updated <- CO2_updated[-c(1, seq(55, 58)),]

# Store total CO2 of states as vector of doubles
total_CO2 <- as.double(CO2_updated[,52])

# Create a new data frame manually
new_updated <- data.frame("year" = seq(1970, 2022),
                          "total_CO2" = total_CO2)

# Rename indices
rownames(new) <- NULL

### DATA VISUALIZATION:
# Graph for CO2
ggplot(data = new_updated, mapping = aes(x = year,
                                         y = total_CO2)) +
  geom_line(color = "blue") +
  geom_vline(xintercept = 2006, linetype = "dashed", color = "red") +
  labs(title = "Time Series of Total CO2 in the U.S.",
       x = "Year",
       y = "CO2 Emissions")
```

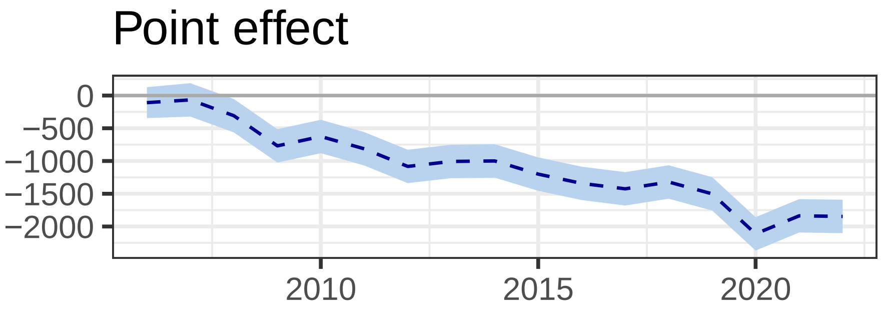


```
### DATA WRANGLING:
# Create a vector of years in class Date
# For consistency, we'll use the first day of each year (e.g., 2006-01-01)
years_dates <- paste0(seq(1970, 2022), "-01-01")

### CAUSALARIMA:
# EN-US: Define the intervention timepoint (day when the intervention occurs)
# Intervention is 2006, which is 36 years (~13140 days) from start of 1970
intervention_time <- 13140

# EN-US: Create a time series object for y with YEARLY seasonality
CO2_ts <- ts(new_updated$total_CO2, frequency = 1)

# EN-US: Define pre-intervention and post-intervention indices
# Pre-Period: Days 1 to 13140 (before the intervention).
pre_period <- 1:(intervention_time - 1)
# Post-Period: Days 13140 to 18980 (after the intervention).
post_period <- intervention_time:length(CO2_ts)

# EN-US: Set up the parameters for CausalArima's int.date (the intervention date)
# and dates (the full date sequence) arguments
intervention_date <- as.Date("2006-01-01")
all_dates <- as.Date(years_dates)

# EN-US: Fit the CausalArima model for causal effect estimation
ce <- CausalArima(y = ts(new_updated$total_CO2, frequency = 1),
                  dates = all_dates, int.date = intervention_date,
                   nboot = 1000)

# EN-US: Plot the impact
forecasted <- plot(ce, type = "forecast")
print(forecasted)
```

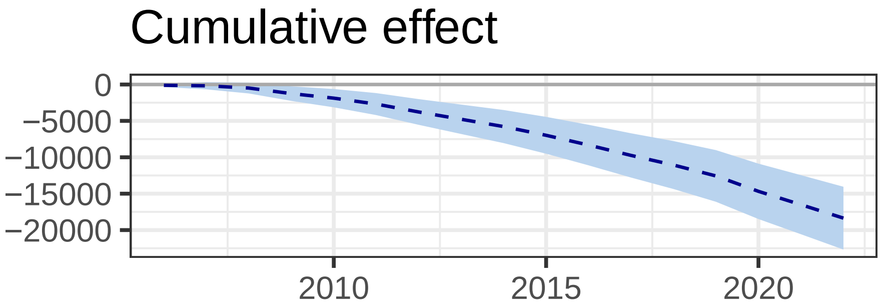


```
impact_p <-plot(ce, type = "impact")
grid.arrange(impact_p$plot, impact_p$cumulative_plot)
```

Point effect



Cumulative effect



```
# EN-US: Display the normalized impact
summary_model <- impact(ce)
summary_model$arima
```

```
## $arima_order
##             p d q
## arima_order 0 1 1
##
## $param
##              coef          se  t value
## ma1    0.4083077  0.1781916 2.291397
## drift 48.2419835 27.6076088 1.747416
##
## $accuracy
##                      ME     RMSE      MAE         MPE     MAPE      MASE
## Training set 0.4164129 115.2695 83.09138 -0.01919972 1.699822 0.7592858
##                      ACF1
## Training set -0.07426968
##
## $log_stats
##            loglik      aic      bic     aicc
## metrics -216.4009 438.8019 443.4679 439.5761
```

```
summary_model$impact_norm
```

```
## $average
##   estimate       sd  p_value_left p_value_bidirectional p_value_right
## 1 -1080.13 40.51863 7.301054e-157                     0             1
##
```

```
## $sum
##    estimate       sd  p_value_left p_value_bidirectional p_value_right
## 1 -18362.22 688.8168 7.301054e-157                     0             1
##
## $point_effect
##    estimate       sd p_value_left p_value_bidirectional p_value_right
## 1 -1846.235 130.0443  4.78291e-46                     0             1
```