econproject, Rachel Nguyen

library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.1 ✔ readr 2.1.4  
✔ forcats 1.0.0 ✔ stringr 1.5.0  
✔ ggplot2 3.4.1 ✔ tibble 3.2.1  
✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
✔ purrr 1.0.1   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(dplyr)  
library(lubridate)  
library(tidyr)  
library(purrr)  
library(stringr)  
library(fixest)  
library(rio)  
library(haven)

# Cleaning the data

* When cleaning data, think about what you want your final data to look like
* For this project, we want Google trends index data at some level (perhaps its original keyword-by-week level, or aggregated to keyword-by-month, or college-by-week, or college-by-month etc.) combined with the Scorecard data about schools
* So our tasks are:

1. Read in the Google Trends data
2. Aggregate the Google Trends data how we want it
3. Read in the Scorecard data
4. Merge in the Scorecard data

googletrends <- list.files(pattern = "trends\_up\_to\_", full.names = TRUE)  
#read in that vector of filenames, using rbind = TRUE to bind all the results together into a single dataset  
df1 <- import\_list(googletrends, rbind = TRUE, fill = TRUE )

# Aggregating the Google Trends data   
df1 <- df1 %>%   
 mutate(week = str\_sub(monthorweek, start = 1, end = 10)) %>%   
 mutate(week = ymd(week)) %>%   
 mutate(month = floor\_date(week, unit = "month"))

# Aggregating   
df1 <- df1 %>%  
 group\_by(schname, keyword) %>%  
 mutate(std\_index = (index - mean(index))/sd(index)) #to aggregate your standardized index to the keyword-month level, or school-week level, or school-month level, or whatever you want

scorecard <- import("Most+Recent+Cohorts+(Scorecard+Elements).csv")  
id\_name\_link <- import("id\_name\_link.csv")

id\_name\_link <- id\_name\_link %>%  
 group\_by(schname) %>%  
 mutate(n = n()) %>%  
 filter(n == 1) # or drop can be fine.  
  
   
  
colnames(scorecard)[colnames(scorecard) == "UNITID"] = "unitid"  
  
  
id\_link <- inner\_join(id\_name\_link, scorecard, by ="unitid")  
  
gg\_link <-inner\_join(df1, id\_link, by ="schname")

#  
# This is the clean data after we clean and we will work with it.   
export(gg\_link, "finaldata.csv")

#  
export

function (x, file, format, ...)   
{  
 if (missing(file) & missing(format)) {  
 stop("Must specify 'file' and/or 'format'")  
 }  
 else if (!missing(file) & !missing(format)) {  
 fmt <- tolower(format)  
 cfile <- file  
 f <- find\_compress(file)  
 file <- f$file  
 compress <- f$compress  
 }  
 else if (!missing(file) & missing(format)) {  
 cfile <- file  
 f <- find\_compress(file)  
 file <- f$file  
 compress <- f$compress  
 fmt <- get\_ext(file)  
 }  
 else if (!missing(format)) {  
 fmt <- get\_type(format)  
 file <- paste0(as.character(substitute(x)), ".", fmt)  
 compress <- NA\_character\_  
 }  
 fmt <- get\_type(fmt)  
 outfile <- file  
 if (fmt %in% c("gz", "gzip")) {  
 fmt <- tools::file\_ext(tools::file\_path\_sans\_ext(file,   
 compression = FALSE))  
 file <- gzfile(file, "w")  
 on.exit(close(file))  
 }  
 data\_name <- as.character(substitute(x))  
 if (!is.data.frame(x) & !is.matrix(x)) {  
 if (!fmt %in% c("xlsx", "html", "rdata", "rds", "json")) {  
 stop("'x' is not a data.frame or matrix")  
 }  
 }  
 else if (is.matrix(x)) {  
 x <- as.data.frame(x)  
 }  
 class(file) <- c(paste0("rio\_", fmt), class(file))  
 .export(file = file, x = x, ...)  
 if (!is.na(compress)) {  
 cfile <- compress\_out(cfile = cfile, filename = file,   
 type = compress)  
 unlink(file)  
 return(invisible(cfile))  
 }  
 invisible(unclass(outfile))  
}  
<bytecode: 0x7f90e6b4fba8>  
<environment: namespace:rio>

merged\_df <- import("finaldata.csv")

merged\_df <- merged\_df %>%  
 filter(PREDDEG == 3)  
  
weekly <- merged\_df %>%  
 group\_by(schname, monthorweek) %>%  
 mutate(weekly\_meanindex = mean(index)) %>%  
 na.omit()

# Ensure the column is numeric  
merged\_df$`md\_earn\_wne\_p10-REPORTED-EARNINGS` <- as.numeric(as.character(merged\_df$`md\_earn\_wne\_p10-REPORTED-EARNINGS`))

Warning: NAs introduced by coercion

# Calculate median, lower quartile (25th percentile), and upper quartile (75th percentile)  
  
income\_median <- median(merged\_df$`md\_earn\_wne\_p10-REPORTED-EARNINGS`, na.rm = TRUE)  
income\_low <- quantile(merged\_df$`md\_earn\_wne\_p10-REPORTED-EARNINGS`, 0.25, na.rm = TRUE) # low income   
income\_high <- quantile(merged\_df$`md\_earn\_wne\_p10-REPORTED-EARNINGS`, 0.75, na.rm = TRUE) # high income   
  
  
# Categorize incomes into High, Middle, and Low  
merged\_df <- merged\_df %>%  
 mutate(treated = case\_when(  
 `md\_earn\_wne\_p10-REPORTED-EARNINGS` >= income\_high ~ "High",  
 `md\_earn\_wne\_p10-REPORTED-EARNINGS` <= income\_low ~ "Low",  
 TRUE ~ "Middle" # this covers all other cases  
 ))

# Create binary variable for High/Low income  
merged\_df <- merged\_df %>%  
 mutate(Earnings = ifelse(`md\_earn\_wne\_p10-REPORTED-EARNINGS` >= income\_median, "High", "Low"))  
  
# Categorize incomes into High, Middle, and Low  
merged\_df <- merged\_df %>%  
 mutate(after = case\_when(  
 `md\_earn\_wne\_p10-REPORTED-EARNINGS` >= income\_high ~ "High",  
 `md\_earn\_wne\_p10-REPORTED-EARNINGS` <= income\_low ~ "Low",  
 TRUE ~ "Middle" # this covers all other cases  
 ))

# Create binary variable for High/Low income  
merged\_df <- merged\_df %>%  
 mutate(Earnings\_binary = ifelse(`md\_earn\_wne\_p10-REPORTED-EARNINGS` >= income\_median, "High", "Low"))

With having median as 41700, we can consider that the low income is below this number and high income is considered if it is higher that 41700

# keep only variable that we will use   
merged\_2<- merged\_df %>%  
 select(unitid, schname, keyword, week, `md\_earn\_wne\_p10-REPORTED-EARNINGS`, Earnings, std\_index)

merged\_2 <- drop\_na(merged\_2)

merged\_2 <- merged\_2 %>%   
 mutate(aftervariable = `md\_earn\_wne\_p10-REPORTED-EARNINGS` >= income\_high, before\_variable = week >= as.Date("2015-09-12"))

regression <- feols(std\_index ~aftervariable\* before\_variable, data = merged\_2)  
etable(regression)

regression  
Dependent Var.: std\_index  
   
Constant 0.0267\*\*\* (0.0015)  
aftervariableTRUE 0.0347\*\*\* (0.0029)  
before\_variableTRUE -0.1457\*\*\* (0.0035)  
aftervariableTRUE x before\_variableTRUE -0.1916\*\*\* (0.0069)  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
S.E. type IID  
Observations 741,598  
R2 0.00671  
Adj. R2 0.00671  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

ggplot(merged\_2, aes(week, std\_index, color = Earnings)) +  
 stat\_summary(geom = 'line')

No summary function supplied, defaulting to `mean\_se()`



labs(  
 title = 'Search index between high and low earning universities before and after treatment',  
 x = 'Year',   
 y = 'Standardized Index'  
 ) +  
 geom\_vline(xintercept = as.Date("2015-09-12"), color = "black", size = 0.5) +  
 annotate(  
 "text",   
 x = as.Date("2015-09-12"),   
 y = min(merged\_2$std\_index),   
 label = "Treatment date",   
 hjust = 1.2,  
 vjust = -1.0,  
 color = "black"  
 ) +  
 theme\_classic()

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.

NULL

# Write Up

The goal of this analysis is to answer the research question: Among colleges that predoinantly grant bachelor’s degrees, did the release of the Scoredcard shift student interest to high-earnings colleges relative to low-earnings ones.

There is a variable in the Scorecard with information about the median earnings of graduates ten years after graduation for each colleges. The average income was calculated as $41700. Any value below is will be considered as low-income and high-income when the value is larger than $41700.

After cleaning data, I choose those variables for this analysis:

1. Data ( 2015-Sep-12)
2. Low - High income ( 25th pecentile , 75th percentile)
3. UNITED ( ID)
4. Standardized trending index
5. Median earnings of students

I run a regression to find out what the link is between the release Scorecard and student interest. Based on the p-value, we reject the null hypothesis that after the Scorecard came out, there was no change between the treated group and the normal group.

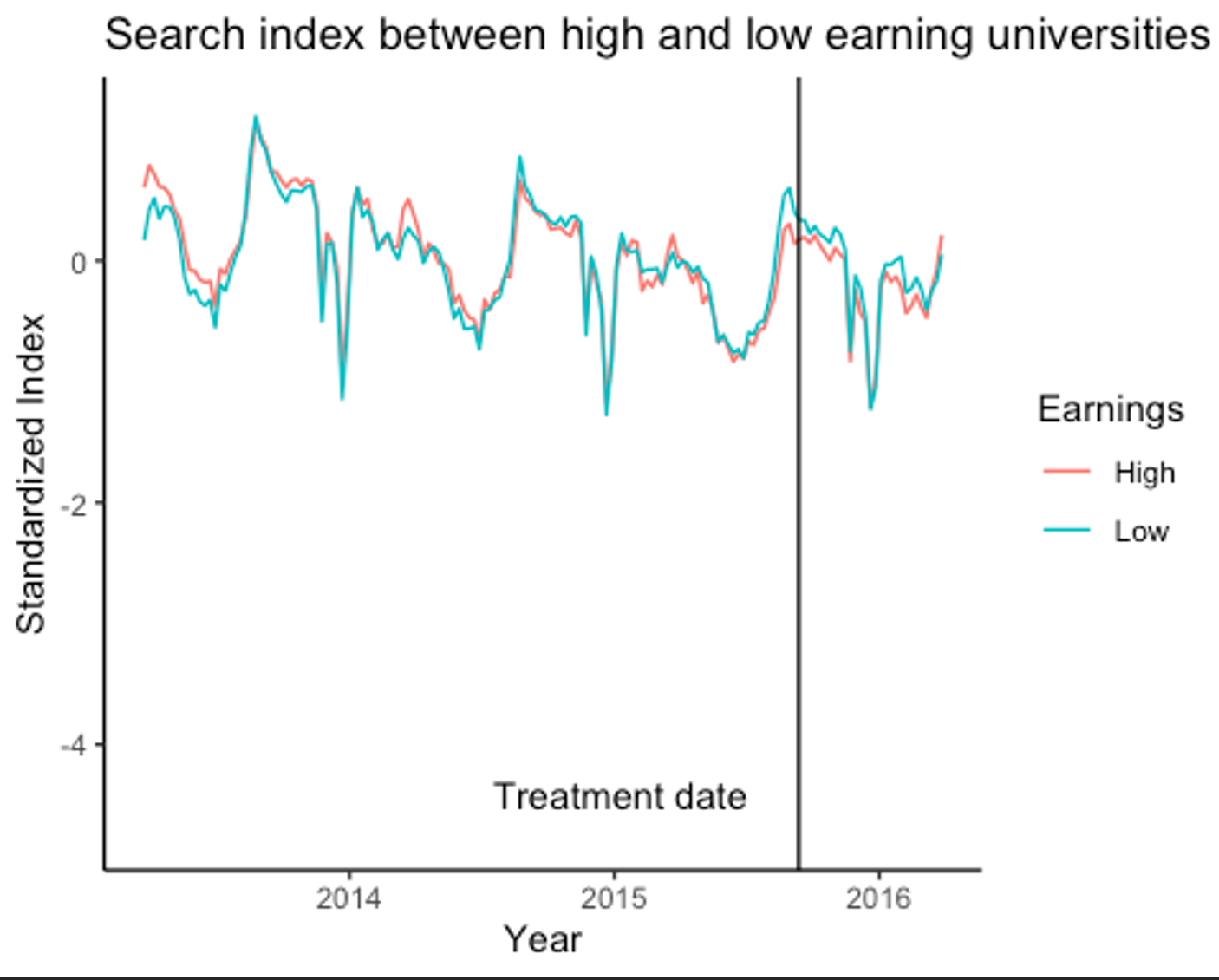
std\_index = β0 + β1 \* income1 + β2 \* beforevariable  + β3 \* (income1 \* beforevariable + ε

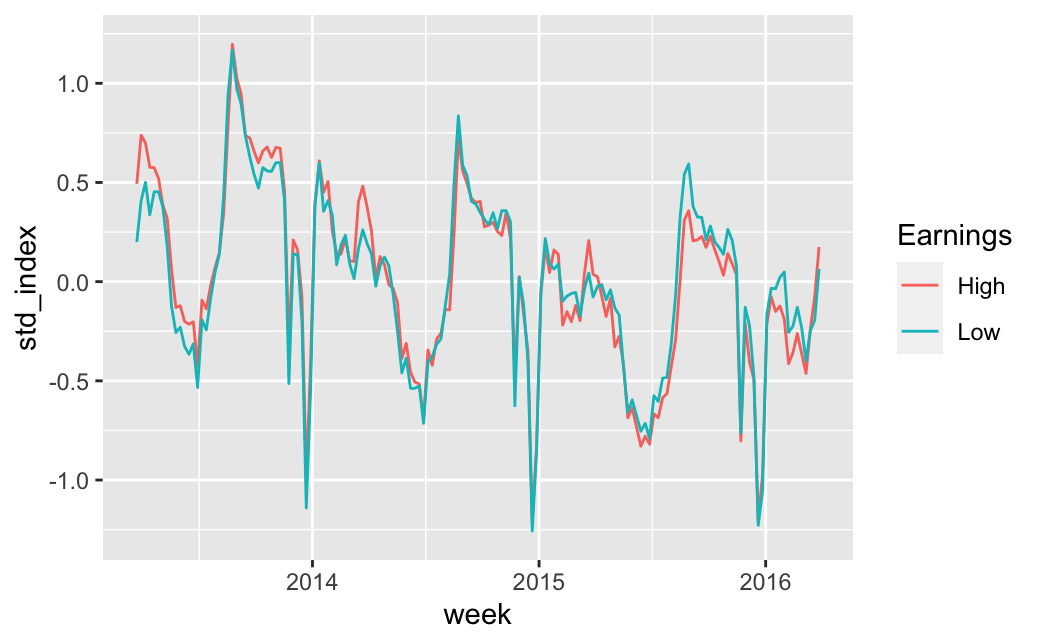
aftervariableTRUE (0.0347): This coefficient measures the difference between the std\_index result when the aftervariableTRUE is true (presumably showing the post-treatment period) and the baseline period (presumably the pre-treatment period). If everything else stays the same, the positive number shows that std\_index is higher in the post-treatment period than in the pre-treatment

period.

before\_variableTRUE (-0.1457): This coefficient measures the difference between the result of the std\_index when the before\_variableTRUE is true (presumably showing a different state in the pre-treatment period) and when the before\_variableTRUE is false (presumably showing the same state in the baseline period). When this condition is met during the pre-treatment time, the negative number shows that std\_index is lower, all other things being equal.

To answer the study question more fully, I made a line chart outside that shows the blackline (when treatment began) and how it affects things.





The Scorecard caused a big change in how students chose colleges after it was made available. There was a clear rise in interest in colleges that led to higher salaries, which suggests an increased focus on schools that lead to higher salaries after graduation. On the other hand, search activity did not change much for colleges with lower earnings, which could mean that people are less interested in schools with lower earning possibilities.

These results show that the Scorecard has a big effect on how students decide which colleges to apply to when they are looking for colleges. The Scorecard’s clear and easy-to-find information about income seems to have been a big factor in getting students to choose schools that will help them make more money after they graduate.

This change in students’ tastes shows that they care more about how education will affect their finances. This could mean that they are becoming more aware of how their educational choices will affect their finances. It also shows how important it is to have access to reliable facts when making these kinds of decisions. By showing potential income results, the Scorecard has given students a more well-rounded view and helped them match their educational goals with possible economic returns.