WriteUp

#loading in the packages  
  
library(rio)  
library(fixest)  
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

library(vtable)

Loading required package: kableExtra

Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':  
  
 group\_rows

library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ forcats 1.0.0 ✔ readr 2.1.4  
✔ ggplot2 3.4.4 ✔ stringr 1.5.1  
✔ lubridate 1.9.3 ✔ tibble 3.2.1  
✔ purrr 1.0.2 ✔ tidyr 1.3.0

── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ kableExtra::group\_rows() masks dplyr::group\_rows()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(lubridate)  
library(stringr)  
library(ggplot2)

#finding wokring directory  
getwd()

[1] "/Users/oliverhall/Library/CloudStorage/OneDrive-SeattleUniversity/Documents/School/3 Junior/WQ/ECON/Data Exploration Project"

#setting working directory  
setwd("/Users/oliverhall/Library/CloudStorage/OneDrive-SeattleUniversity/Documents/School/3 Junior/WQ/ECON/Data Exploration Project")  
  
#importing data  
Data1 <- import('final\_data.xlsx')

#renaming column to an easier convention  
Data1 <- rename(Data1, earnings = "md\_earn\_wne\_p10-REPORTED-EARNINGS")  
  
#column contains either PrivacySuppressed or NULL need to be removed  
Data1 <- Data1 %>%  
 filter(!str\_detect(earnings, "PrivacySuppressed|NULL"))

#converting earnings to an integer  
Data2 <- as.integer(Data1$earnings)  
  
Data2 <- data.frame(Data2)  
  
#determining what constitutes high and low earning colleges  
mean(Data2$Data2, na.rm = TRUE)

[1] 42482.8

sd(Data2$Data2, na.rm = TRUE)

[1] 11812.96

#defining what a high, low, and medium earning college is   
  
Data1 <- Data1 %>%  
 mutate(earning\_level = case\_when(  
 earnings <= 30646.05 ~ 'low',  
 earnings >= 42466.35 ~ 'high',  
 TRUE ~ 'Medium'  
 ))

#generating groups based off when the scorecard tool was introduced  
Data1 <- Data1 %>%  
 mutate(scorecard\_effect = case\_when(  
 date <= '2015-09-01' ~ 'before',  
 earnings > '2015-09-01' ~ 'after'  
 ))

#removing the 'Medium' field from earnings\_level in order to just compare high to low earning colleges as well as NA's  
low\_or\_high <- filter(Data1, earning\_level != 'Medium')  
#removing NA's  
low\_or\_high <- na.omit(low\_or\_high)

#  
low\_or\_high2 <- low\_or\_high %>% group\_by(scorecard\_effect, earning\_level, date) %>%  
 summarize(mean\_index = mean(standardized\_index))

`summarise()` has grouped output by 'scorecard\_effect', 'earning\_level'. You  
can override using the `.groups` argument.

#graph  
ggplot(data = low\_or\_high2, aes(x=date, y = mean\_index, color= earning\_level)) + geom\_line() + labs(x = "Year", y = "Average Index", color = "Earning Level") + theme()



#creating binary variables  
regData <- low\_or\_high2 %>%   
 mutate(high\_earners = ifelse(earning\_level == "high", 1, 0),  
 low\_earners = ifelse(earning\_level == "low", 1, 0))  
  
regData <- regData %>%   
 mutate(screlease = ifelse(date < '2015-09-01', 0, 1))

reggg <- feols(mean\_index ~ high\_earners \* screlease, data = regData)  
etable(reggg)

reggg  
Dependent Var.: mean\_index  
   
Constant 0.0581 (0.0592)  
high\_earners 0.0126 (0.0837)  
screlease -0.3214\* (0.1470)  
high\_earners x screlease -0.0141 (0.2079)  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
S.E. type IID  
Observations 74  
R2 0.12512  
Adj. R2 0.08763  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

**Write Up:**

I decided to aggregate my data by month so that I had a month-to-month scale. This was important to do because there was a huge amount of data and I personally preferred to have a data set with less noise and be a bit cleaner even if that meant losing some power.

In order to create my groups for high and low-earning colleges I calculated the mean (42465.76) and the standard deviation (11818.2) for earnings. I defined my high-earnings college as the mean plus one standard deviation and my low-earnings college as the mean minus one standard deviation.

I decided that I wanted to graph my data to see if I was heading in the right direction. I grouped by earning level and date to get averages for the index scores based on the month and earning level (high and low). It was interesting to see that the index scores go up every year around the June period for both high and low-earning colleges. This is likely because this is around when people are making college admissions.

The introduction of the College Scorecard increased search activity on Google Trends for colleges with high-earning graduates by -0.0141 units relative to what it did for colleges with low-earning graduates, with a standard error of 0.2079. This result comes from the high\_earnings x screlease coefficient in my regression.