Data Exploration Project

2024-02-20

## R-Markdown

extracted\_folder\_path <- "Lab3\_Rawdata"  
file\_names <- list.files(path = extracted\_folder\_path, pattern = "trends\_up\_to\_", full.names = TRUE)  
combined\_data <- import\_list(file\_names, rbind = TRUE)  
combined\_data<- combined\_data%>%  
 mutate(short\_date = str\_sub(monthorweek, 1, 10))  
combined\_data<- combined\_data%>%  
 mutate(actual\_date= ymd(short\_date))  
combined\_data <-combined\_data%>%  
 mutate(month = floor\_date(actual\_date, unit = "month"))

scorecard\_data <-import('Lab3\_Rawdata/Most+Recent+Cohorts+(Scorecard+Elements).csv')  
id\_name\_link <- import('Lab3\_Rawdata/id\_name\_link.csv')

id\_name\_link <- id\_name\_link %>%  
 group\_by(schname) %>%  
 mutate(n = n()) %>%  
 filter(n == 1) %>%  
 ungroup() %>%  
 select(-n)

cutoff\_date <- as.Date('2015-09-15')  
month\_cutoff <- as.Date('2015-10-15')

combined\_data\_joined<- combined\_data %>%  
 inner\_join(id\_name\_link, by = c("schname" = "schname"))%>%  
 mutate(post\_treatment = if\_else(actual\_date < cutoff\_date, FALSE, TRUE))%>%  
 mutate(pre\_treatment = if\_else(actual\_date > cutoff\_date, FALSE, TRUE))%>%  
 mutate(treatment = if\_else(actual\_date < cutoff\_date, 'Pre Release', 'Post Release'))

index\_means\_and\_sd <- combined\_data\_joined %>%  
 group\_by(schname, keyword)%>%  
 mutate(standard\_index =   
 (index - mean(index, na.rm = TRUE)) / (sd(index, na.rm = TRUE))  
 )  
   
index\_means\_and\_sd<- index\_means\_and\_sd%>%  
 group\_by(unitid, actual\_date)%>%  
 summarise(mean\_weekly\_index = mean(standard\_index, na.rm = TRUE), .groups = 'drop')%>%  
 ungroup()

combined\_data\_index <- combined\_data\_joined%>%  
 inner\_join(index\_means\_and\_sd, by = c('unitid' = 'unitid', 'actual\_date' = 'actual\_date'))

combined\_scorecard <- combined\_data\_index%>%  
 inner\_join(scorecard\_data, by = c('unitid' = 'UNITID'))

filtered\_combined <- combined\_scorecard%>%  
 filter(PREDDEG == 3)%>%  
 mutate(reported\_earning = as.numeric(`md\_earn\_wne\_p10-REPORTED-EARNINGS`))  
  
filtered\_combined <- filtered\_combined%>%   
 drop\_na()

mean\_earnings <- mean(filtered\_combined$reported\_earning, na.rm = TRUE)  
std\_earnings <- sd(filtered\_combined$reported\_earning, na.rm = TRUE)

filtered\_combined <- filtered\_combined %>%  
 mutate(high\_income = ifelse(reported\_earning >= mean\_earnings + std\_earnings, TRUE, FALSE))%>%  
 mutate(low\_income = ifelse(reported\_earning <= mean\_earnings - std\_earnings, TRUE, FALSE))%>%  
 mutate(income = ifelse(reported\_earning >= mean\_earnings + std\_earnings, "High Income", "Low Income"))  
  
filtered\_combined <- filtered\_combined%>%  
 filter(  
 reported\_earning < mean\_earnings - std\_earnings | reported\_earning > mean\_earnings + std\_earnings)  
  
filtered\_combined <- filtered\_combined %>%  
 mutate(  
 Month = month(ymd(month)),  
 Season = case\_when(  
 Month %in% c(12, 1, 2) ~ "Winter",  
 Month %in% c(3, 4, 5) ~ "Spring",  
 Month %in% c(6, 7, 8) ~ "Summer",  
 Month %in% c(9, 10, 11) ~ "Fall",  
 TRUE ~ NA\_character\_   
 )  
 )  
  
filter\_more <- filtered\_combined%>%  
 filter(actual\_date < month\_cutoff)

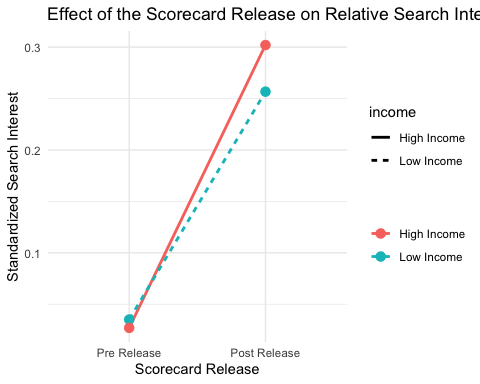
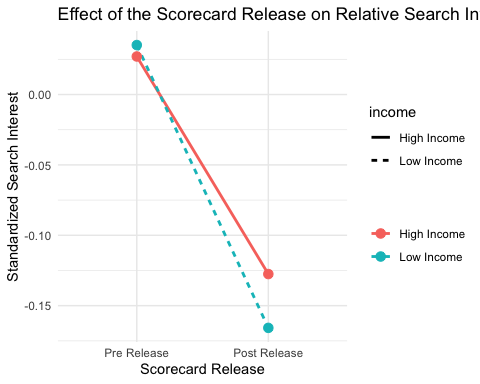
group\_means <- filtered\_combined %>%  
 group\_by(post\_treatment, high\_income, treatment, income)%>%  
 summarise(mean\_standard\_index = mean(mean\_weekly\_index, na.rm = TRUE), .groups = 'drop')%>%  
 ungroup()  
  
group\_means\_short\_run <- filter\_more %>%  
 group\_by(post\_treatment, high\_income, treatment, income)%>%  
 summarise(mean\_standard\_index = mean(mean\_weekly\_index, na.rm = TRUE), .groups = 'drop')%>%  
 ungroup()  
  
group\_means\_season <- filtered\_combined%>%  
 group\_by(post\_treatment, high\_income, treatment, income, Season)%>%  
 summarise(mean\_standard\_index = mean(mean\_weekly\_index, na.rm = TRUE), .groups = 'drop')%>%  
 ungroup()  
  
pre\_treat\_low <- group\_means %>% filter(post\_treatment == FALSE, high\_income == FALSE) %>% pull(mean\_standard\_index)  
pre\_treat\_high <- group\_means %>% filter(post\_treatment == FALSE, high\_income == TRUE) %>% pull(mean\_standard\_index)  
post\_treat\_low <- group\_means %>% filter(post\_treatment == TRUE, high\_income == FALSE) %>% pull(mean\_standard\_index)  
post\_treat\_high<- group\_means %>% filter(post\_treatment == TRUE, high\_income == TRUE) %>% pull(mean\_standard\_index)  
  
pre\_treat\_low\_sr <- group\_means\_short\_run %>% filter(post\_treatment == FALSE, high\_income == FALSE) %>% pull(mean\_standard\_index)  
pre\_treat\_high\_sr <- group\_means\_short\_run %>% filter(post\_treatment == FALSE, high\_income == TRUE) %>% pull(mean\_standard\_index)  
post\_treat\_low\_sr <- group\_means\_short\_run %>% filter(post\_treatment == TRUE, high\_income == FALSE) %>% pull(mean\_standard\_index)  
post\_treat\_high\_sr<- group\_means\_short\_run %>% filter(post\_treatment == TRUE, high\_income == TRUE) %>% pull(mean\_standard\_index)  
   
did\_est <- ((pre\_treat\_low - post\_treat\_low)-(pre\_treat\_high - post\_treat\_high))  
did\_est\_sr <- ((pre\_treat\_low\_sr - post\_treat\_low\_sr)- (pre\_treat\_high\_sr - post\_treat\_high\_sr))  
  
print(did\_est)

## [1] 0.04635644

print(did\_est\_sr)

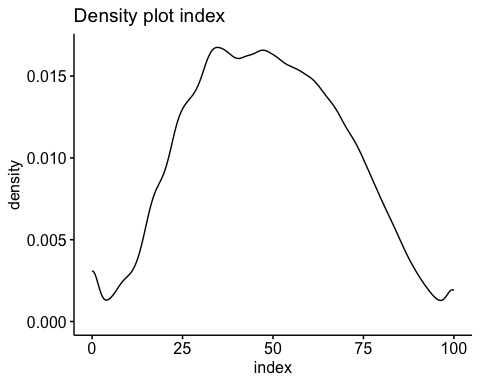
## [1] 0.05341031

## 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.

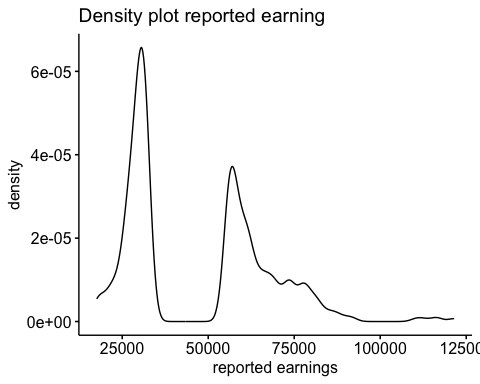


## model model\_sr  
## Dependent Var.: mean\_weekly\_index mean\_weekly\_index  
##   
## Constant 0.0351\*\*\* (0.0028) 0.0351\*\*\* (0.0028)  
## high\_incomeTRUE -0.0081\* (0.0040) -0.0081\* (0.0039)  
## post\_treatmentTRUE -0.2009\*\*\* (0.0067) 0.2216\*\*\* (0.0159)  
## high\_incomeTRUE x post\_treatmentTRUE 0.0464\*\*\* (0.0095) 0.0534\* (0.0227)  
## Fixed-Effects: ------------------- ------------------  
## Season No No  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
## S.E. type IID IID  
## Observations 163,157 138,803  
## R2 0.00870 0.00348  
## Within R2 -- --  
##   
## model\_seasonal  
## Dependent Var.: mean\_weekly\_index  
##   
## Constant   
## high\_incomeTRUE -0.0081 (0.0541)  
## post\_treatmentTRUE -0.2844\*\* (0.0271)  
## high\_incomeTRUE x post\_treatmentTRUE 0.0463 (0.0702)  
## Fixed-Effects: ------------------  
## Season Yes  
## \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
## S.E. type by: Season  
## Observations 163,157  
## R2 0.07207  
## Within R2 0.01707  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

ggdensity(filtered\_combined$index,   
 main = "Density plot index",  
 xlab = "index")



ggdensity(filtered\_combined$reported\_earning,   
 main = "Density plot reported earning",  
 xlab = "reported earnings")



sample\_test <- sample\_n(filtered\_combined, 4900)  
  
shapiro.test(sample\_test$reported\_earning)

##   
## Shapiro-Wilk normality test  
##   
## data: sample\_test$reported\_earning  
## W = 0.88369, p-value < 2.2e-16

shapiro.test(sample\_test$index)

##   
## Shapiro-Wilk normality test  
##   
## data: sample\_test$index  
## W = 0.99224, p-value = 1.019e-15

#this was to check and make sure that the data I'm working with is normally distributed.

Because the research question asks how the change in search interest of high income colleges relative to low income ones was effected by the publishing of the College Scorecard, I elected to choose a difference in difference methodology for my analysis. Difference in difference is usually used when there is a control group, and in that instance it will tell you the effect of the treatment on the treated group. In this case, because there is no control group what I’m doing isn’t really a difference in differences approach, but my analysis will still output the impact on one group relative to another, which is aligned with the research question.

Standardizing the index based on keyword and school name means that a one unit change in standardized index means one standard deviation change in search interest. A value of zero means that the index that week is the same as the overall mean index which would mean there is no change in search interest compared to the schools historical search interest for that week. This allows us to aggregate the change in search interest for schools with varying levels of baseline search interest, and compare how they change over time. I decided to aggregate by unitid, which is unique to each university, and by actual\_date, which is the week in which the observation was recorded. This will allow me in the future to better isolate the effect of the scorecard release both immediately after its release, and in a longer run.

After tying together the scorecard data, I filtered for schools that predominately grant bachelors degrees. Then, because the research question revolves around the post graduation earnings, I created three dummy variables for income. In this case, I decided that the mean of all reported incomes could function as a reference to define lower and higher earning colleges. The first would track if the income for a particular school was below average and assign True to observations that met the criteria, the next would track if it was above average and do the same, and the final one would simply label the observation as “high income”, or “low income”. The reason I did this was to give myself more options in case I wanted to try different types of analysis. I made sure to filter out any colleges that were within one standard deviation from the mean, as doing so would help isolate the significantly higher and lower income colleges which are the subject of the research question.

One of my final steps before running my regression was to find the difference in difference manually as the difference in differences estimate should be the equal to the coefficient on my interaction term. If they match, I know I have not made any glaring mistakes in my data cleaning.

For my analysis I decided to run three regressions. The first would look at the impact of the College Scoreboard on the relative mean index by week for high and low earning colleges. This regression has no limitations on time so the effect shown in the regression shows the change in interest for the entire post treatment period relative to the pre treatment period for high and low earning colleges. I decided to use the equation for difference in difference that we learned in class for the reasons I have explained above. The second regression is the same as the first but only takes post release observations for one month after the release of the scorecard. This is to see if the effect of the scorecard release changed over time, and to what extent. In the final regression I ran I included seasonal controls as fixed effects to try and control for the variation within school caused by seasons. I would assume that interest would likely be higher in the fall and spring as students are applying to school and accepting admission during those times and would likely be more engaged in college research.

In my analysis I found that the introduction of the College Scorecard increased search activity on Google Trends for colleges with high earning graduates by 4.6% of a standard deviation in search interest relative to what it did for colleges with low earning graduates, with a standard of error of 0.00095 standard deviations. This result comes from the interaction term in my regression between high\_incomeTrue and post\_treatmentTrue. Overall, search interest in both high earning colleges and low earning colleges fell relative to their pre release mean search interest, which indicates that in the long run, the effect of the College Scoreboard was negligible in its impact on overall search interest, but did have a modest effect on the relative search interests of high earning colleges and low earning colleges.

My second regression yielded similar results, but had a slightly larger effect on the relative increase for high earning colleges to low earning colleges, being a difference of about 5.3% of a standard deviation. Whats also interesting about the one month analysis is that the coefficient on on post\_treatmentTrue is positive, whereas in my frist regression it was negative. The graph also shows that unlike in my first regression, when compared to their pre release means, the search interest in both low earning and high earning colleges rose. That being said, this month was also October, which is when many high school seniors would working with their college applications, which would also explain the increase in search interest. That being said, the search interest for high earning colleges increased by 5.3% of a standard deviation relative to low earning colleges, which is likely attributable to the scorecard release.

My third regression is more for fun than for anything else. I wanted to try and include fixed effects for season as a way to control for the variation in search interest caused by the season trends in high school seniors that could have explained the results in my previous regression. The within R2 suggests that season is only responsible for about 1.7% of the variation in search interest within income. This suggest to me that it is probably a needless control, although the R2 is higher than the other regressions which could suggest that more of the variation in search index is explained by including season as a fixed effect. But overall, the coefficient on the itneraction is not statistically significant which leads me to believe that including Season Fixed Effects makes th regression less accurate.

To summarize, among colleges that predominantly grant bachelors degrees, the release of the the College Scorecard Data increased search interest for colleges with high earning graduates by 4.6% of a standard deviation relative to colleges with low earning graduates. This shift suggests a modest impact of the scorecard on student interest towards colleges with high earning graduates.