CSB Trend Analysis

College Scoreboard Trends Analysis

By Thomas Taylor

This analysis looks at the impacts of the release of the college scoreboard (CSB) information of median 10 year post graduation reported earnings in September of 2015. Specifically this looks at google trends data before and after the release of that data as a proxy for how the CSB data impacted undergraduate applications/admissions. A difference in difference approach is taken comparing universities that had high post graduate earnings to those that did not.

```
#load packages


library(readr)

library(tidyverse)

── Attaching packages ───────────────────────────────────────
tidyverse 1.3.2 ──
✔ ggplot2 3.4.0      ✔ dplyr   1.0.10
✔ tibble  3.1.8      ✔ stringr 1.5.0
✔ tidyr   1.2.1      ✔ forcats 0.5.2
✔ purrr   1.0.1
── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()

library(fixest)

Warning: package 'fixest' was built under R version 4.2.3

library(vtable)

Loading required package: kableExtra

Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':

    group_rows

library(rio)
```

Datasets loaded in and cleaned

```r
dflist <- list.files("Data")

Most_Recent_Cohorts_Scorecard_Elements_ <-
read_csv("Data2/Most+Recent+Cohorts+(Scorecard+Elements).csv", show_col_types
= FALSE)

college_list <- read_csv("Data2/id_name_link.csv", show_col_types = FALSE)

merged <- import_list(dflist, rbind=TRUE)
```

```
Warning in FUN(X[[i]], ...): Import failed for trends_up_to_finish.csv

Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.

Warning in FUN(X[[i]], ...): Import failed for trends_up_to_inter_1.csv

Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.

Warning in FUN(X[[i]], ...): Import failed for trends_up_to_inter_2.csv

Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.

Warning in FUN(X[[i]], ...): Import failed for trends_up_to_inter_3.csv

Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.

Warning in FUN(X[[i]], ...): Import failed for trends_up_to_inter_4.csv

Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.

Warning in FUN(X[[i]], ...): Import failed for trends_up_to_inter_5.csv

Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.

Warning in FUN(X[[i]], ...): Import failed for trends_up_to_inter_6.csv

Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.

Warning in FUN(X[[i]], ...): Import failed for trends_up_to_UM.csv
```

```
Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.

Warning in FUN(X[[i]], ...): Import failed for trends_up_to_UPhoenix.csv

Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.

Warning in FUN(X[[i]], ...): Import failed for trends_up_to_UT.csv

Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.

Warning in FUN(X[[i]], ...): Import failed for trends_up_to_UTMB.csv

Warning in structure(out, filename = thisfile): Calling 'structure(NULL, *)'
is deprecated, as NULL cannot have attributes.
  Consider 'structure(list(), *)' instead.
```

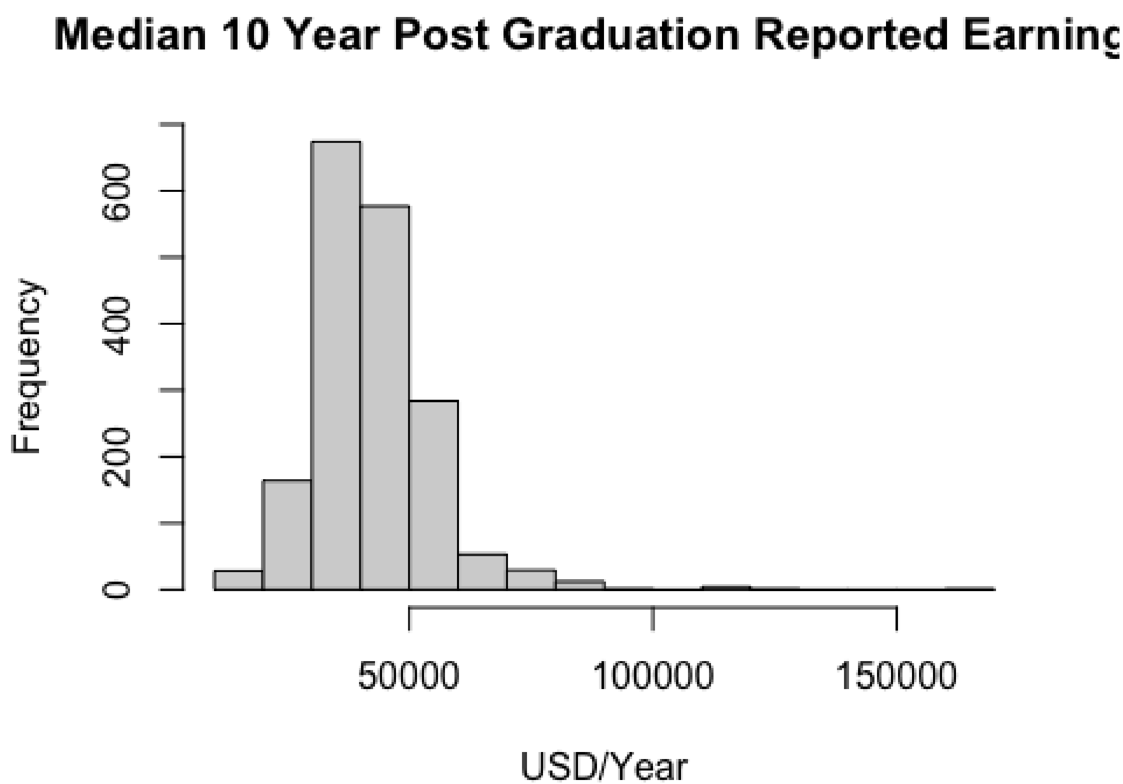
```
g <- college_list %>% group_by(schname) %>% filter(n()<2)


cohrtfilt <- Most_Recent_Cohorts_Scorecard_Elements_ %>% filter(UNITID %in%
g$unitid)

#I merged all the trends data into one giant csv to make it nicer to work
with

filteredtrends <- merged %>% filter(schname %in% g$schname)

#remove bigger lists to save space
rm(college_list, Most_Recent_Cohorts_Scorecard_Elements_, merged)

#filter for just primarily bachelor granting universities
cohrtfilt <- cohrtfilt %>% filter(PREDDEG == 3)
```

Spiting Universities by Earnings:

For this analysis I decided to go with the average (42,300) Median 10 Year Post Graduation Reported Earnings because I did not want to diminish the weight of high outliers in earnings. On the assumption that those are likely to be the most impacted by the release of the College Scoreboard data. At the same time I decided to not go higher than the average to preserve at least some parity in sample size.

```
em <- (cohrtfilt$`md_earn_wne_p10-REPORTED-EARNINGS`)

em2 <- as.numeric(em)
```

```
Warning: NAs introduced by coercion

median_earnings10yr_mean <- mean(na.omit(em2))

hist(em2, main = "Median 10 Year Post Graduation Reported Earnings", xlab =
"USD/Year", ylab = "Frequency")
```



Trends data setup

Given the self relativistic nature of Google Trends data it can be difficult to compare a large number of different searches which may all be at different scales. For this analysis a normalization technique was used in order to compare trend indices (taking the mean value of each keyword and subtracting that value from each index, then dividing by the standard deviation). This system is a good estimation, but a better rework would be to have some dummy search that you compare all the searches to. It can be a bit tricky to pick a good dummy search because as seen in Figure 1 if you choose a dummy search that is too popular it can cause data compression issues. Finding an ideal dummy search can be a bit tricky but if you can find a good one the upside is you can be much more precise in your cross keyword comparisons. An especially effective strategy can be to pick a dummy search that you have a good estimate as to what the number of searches at peak actually was.

Figure 1: Sample Keyword trend with poor dummy search choice.

Figure 2: Sample Keyword with better Dummy Search

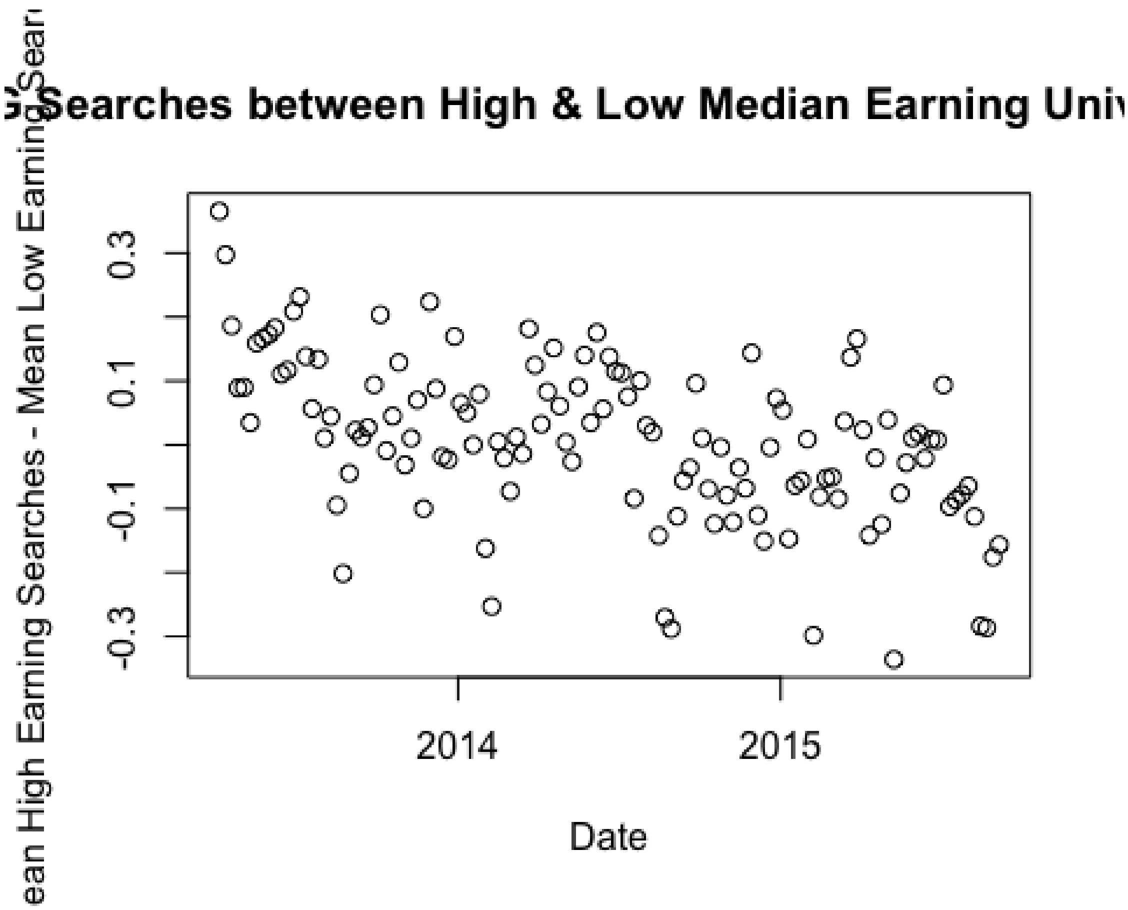
Figure 3: Example of how searches can be compared relative to peak of same dummy search, here Duke and UAB can be compared directly.

```
#remove na values from trends
filteredtrends <- na.omit(filteredtrends)

#normalize trends (+0.00000000000001 introduced to avoid any divide by 0
errors)
filteredtrends <- filteredtrends %>% group_by(keyword) %>%
  mutate(index_normed = ((index-mean(index))/(sd(index)+0.000000000001)))

#get dates in readable format, for the rest of the anlaysis end dates will
primarily be used but note each period is one week
filteredtrends <- filteredtrends %>% mutate(end_date =

as.Date(substr(monthorweek, 14, 23), format="%Y-%m-%d"))

filteredtrends <- filteredtrends %>% mutate(start_date =

as.Date(substr(monthorweek, 1, 10), format="%Y-%m-%d"))

#split colleges into low and high earning by whether their 10 year median
earnings for grads are above or below the mean
high_earning <- cohrtfilt %>%
filter(as.numeric(`md_earn_wne_p10-REPORTED-EARNINGS`)>median_earnings10yr_me
an)

Warning in mask$eval_all_filter(dots, env_filter): NAs introduced by coercion

low_earning <- cohrtfilt %>%
filter(as.numeric(`md_earn_wne_p10-REPORTED-EARNINGS`)<median_earnings10yr_me
an)

Warning in mask$eval_all_filter(dots, env_filter): NAs introduced by coercion

high_univ <-g %>% filter(unitid %in% high_earning$UNITID)

low_univ <-g %>% filter(unitid %in% low_earning$UNITID)
```

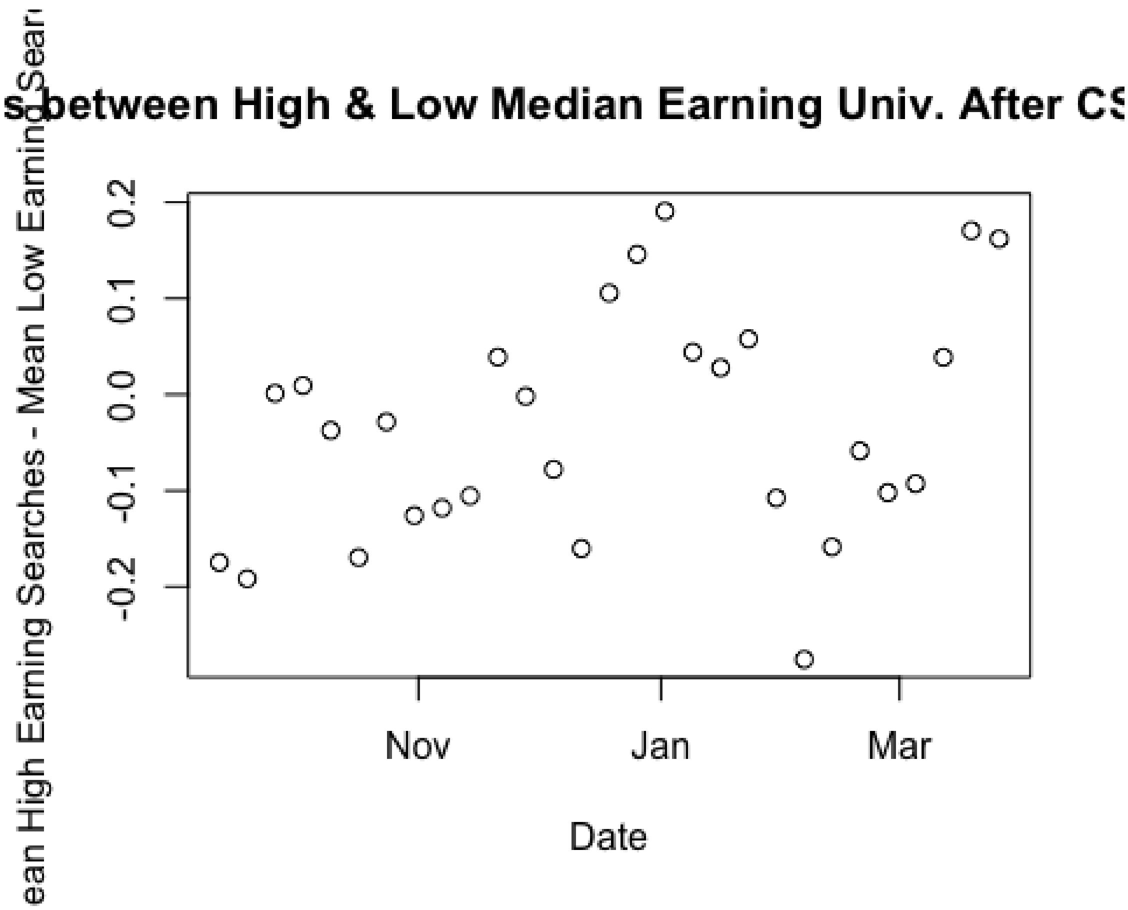
```
high_trends <- filteredtrends %>% filter(schname %in% high_univ$schname)


low_trends <- filteredtrends %>% filter(schname %in% low_univ$schname)

#remove unfiltered dataframes
rm(filteredtrends)
rm(g)
rm(cohrtfilt)

#date college scoreboard goes public
sb_date <- as.Date("2015-09-08", format="%Y-%m-%d")

#split trends data by before and after csb date

high_before <- high_trends %>% filter(end_date < sb_date)

high_after <- high_trends %>% filter(end_date >= sb_date)

low_before <- low_trends %>% filter(end_date < sb_date)

low_after <- low_trends %>% filter(end_date >= sb_date)
```

Graphing the data and looking at visual trends

```
#average normalized search per week
ha_means <- aggregate(high_after$index_normed, list(high_after$end_date),
FUN=mean)

hb_means <- aggregate(high_before$index_normed, list(high_before$end_date),
FUN=mean)


la_means <- aggregate(low_after$index_normed, list(low_after$end_date),
FUN=mean)

lb_means <- aggregate(low_before$index_normed, list(low_before$end_date),
FUN=mean)

plot(hb_means$Group.1, (hb_means$x - lb_means$x), xlab = "Date", ylab = "Mean
High Earning Searches - Mean Low Earning Searches" )
title("Diff in AVG Searches between High & Low Median Earning Univ. Prior to
CSB")
```



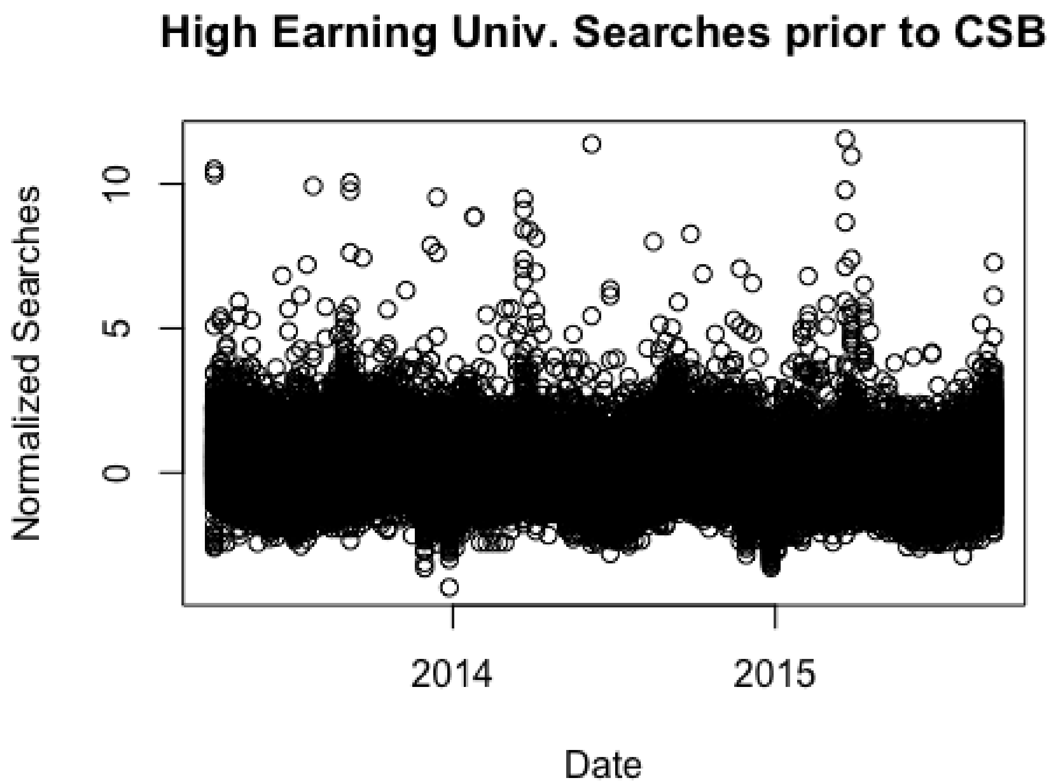

```
plot(ha_means$Group.1, (ha_means$x - la_means$x),  xlab = "Date", ylab =
"Mean High Earning Searches - Mean Low Earning Searches")
title("Diff in AVG Searches between High & Low Median Earning Univ. After CSB
Release in Sept. 2015")
```



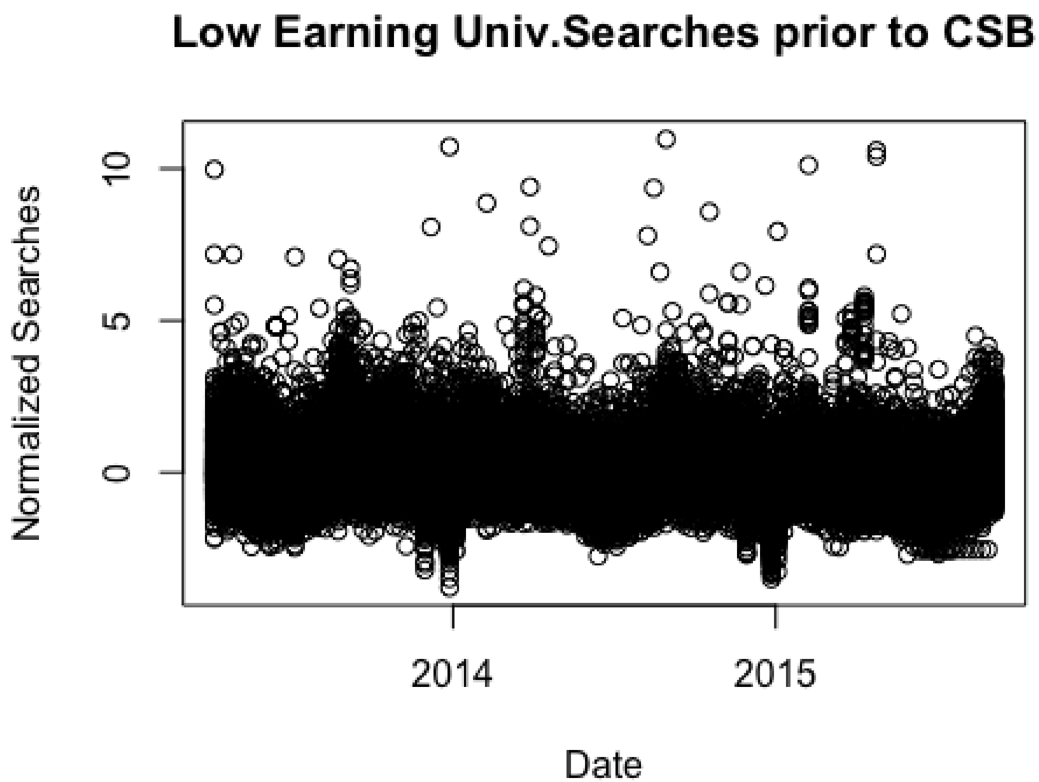
Seasonality:

One important element to factor in when looking at the results on this analysis is that both the above mean earnings and below mean earnings universities showed strong seasonal patterns. Intuitively this makes a lot of sens because college applications are a rather seasonal activity and it makes sense that annually searches would be higher in the fall and winter and lower in spring and summer. As such it is important to keep i mind that linear estimation models will likely not fit this data particularly well, however as the degree of seasonality is not likely to be correlated with either low or high mean earnings it does not need to be included in the regressions.

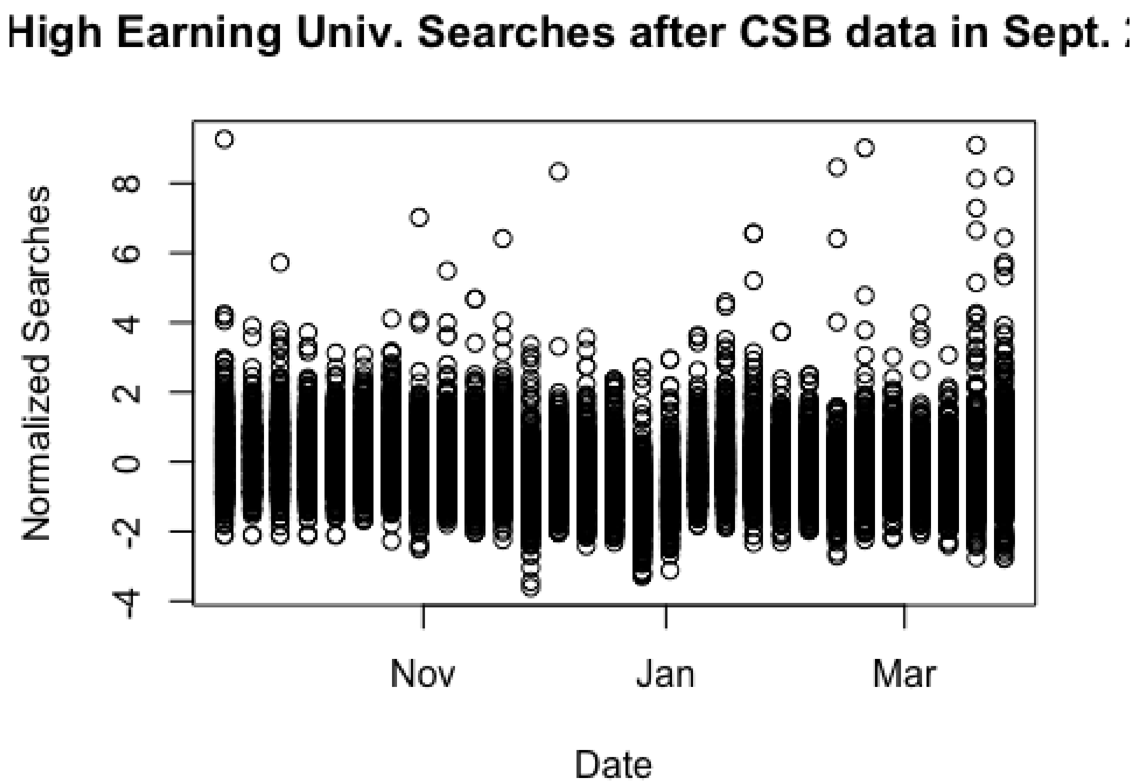
```
#nonaveraged plots

plot(high_before$end_date, high_before$index_normed, xlab = "Date", ylab =
"Normalized Searches")
title("High Earning Univ. Searches prior to CSB")
```



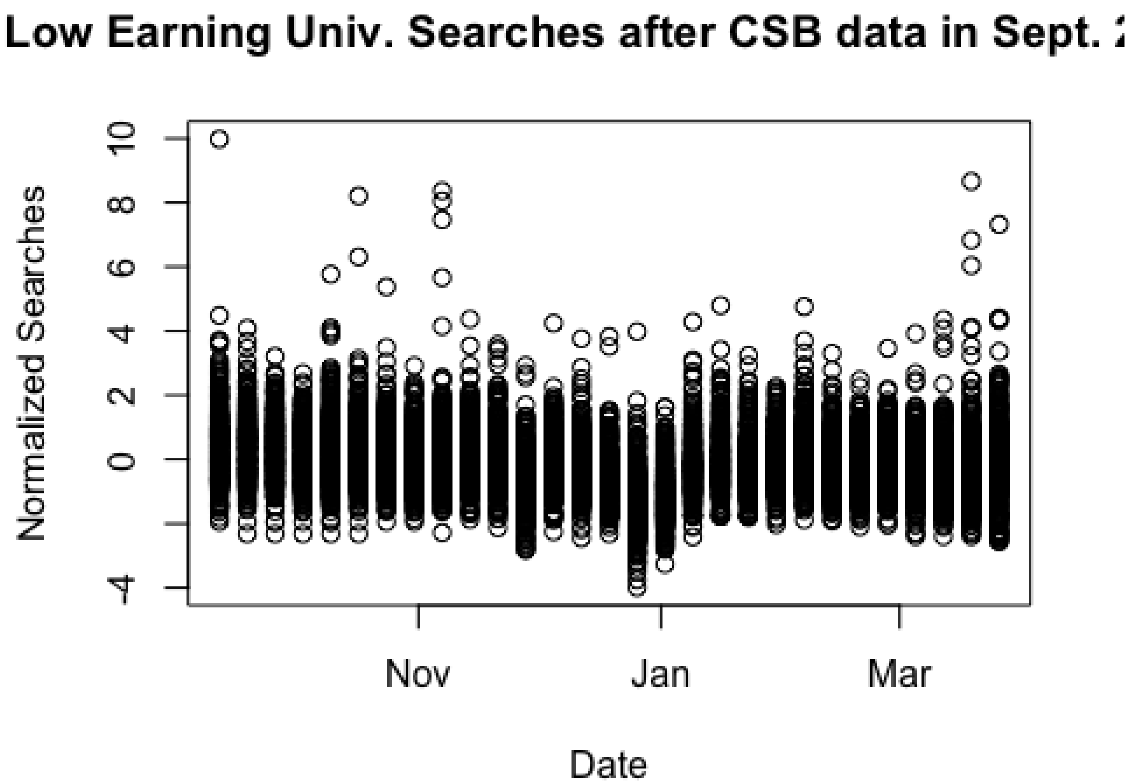

```
plot(low_before$end_date, low_before$index_normed, xlab = "Date", ylab =
"Normalized Searches")
title("Low Earning Univ.Searches prior to CSB")
```




```
plot(high_after$end_date, high_after$index_normed,  xlab = "Date", ylab =
"Normalized Searches")
title("High Earning Univ. Searches after CSB data in Sept. 2015")
```




```r
plot(low_after$end_date, low_after$index_normed,  xlab = "Date", ylab =
"Normalized Searches")
title("Low Earning Univ. Searches after CSB data in Sept. 2015")
```



The March madness issue.

One thing that stands out when looking at the data is the is the high number of outliers in March and April. This is well after the traditional college application season has ended and seems to only affect some schools. My hypothesis is that this is an effect of the annual college basketball "March Maddness" tournament. This is an event that draws in a much larger search audience than potential college applicants, and results in a lot more searches for colleges that make the tournament, especially colleges that make upsets or deep runs. Unfortunately, this is something that is not included in the analyzed dataset and does have a correlation with 10 year median earnings (three of the Final four teams in the 2016 Men's tournament were above the mean 10 year median earnings and all four of the women's were above mean as well.). College football may have similar effects in late November to early January but with all colleges searches increasing during those times that can be more difficult to tell by just looking at the data visually.

```
#combine back into one df for regression and add in Factors
low_before <- low_before %>% mutate(earn_type = 'Low')
low_before <- low_before %>% mutate(sbd = 'Before')

low_after <- low_after %>% mutate(earn_type = 'Low')
low_after <- low_after %>% mutate(sbd = 'After')
```

```
high_before <- high_before %>% mutate(earn_type = 'High')
high_before <- high_before %>% mutate(sbd = 'Before')

high_after <- high_after %>% mutate(earn_type = 'High')
high_after <- high_after %>% mutate(sbd = 'After')


big_Df <- rbind(low_before, low_after, high_before, high_after)
```

Regression and Interpretation

```
Search_diff_in_diff <- feols(index_normed ~ earn_type + sbd + sbd:earn_type,
data = big_Df)

etable(Search_diff_in_diff, vcov = "hetero")

                         Search_diff_in_diff
Dependent Var.:                 index_normed

Constant                  -0.1417*** (0.0089)
earn_typeLow                0.0343** (0.0131)
sbdBefore                  0.1740*** (0.0099)
earn_typeLow x sbdBefore   -0.0421** (0.0145)
________________________ ___________________
S.E. type                Heteroskedast.-rob.
Observations                         127,452
R2                                   0.00370
Adj. R2                              0.00367
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Searches for high earning universities increased by 0.06 normalized units more than low earning universities after the CSB data was publicly released. However, while this increase is significant at the 95% confidence level from the R2 value this difference in difference only explains less than one percent of the variation in searches. On a scale of normalized searches varying between -5 and 15 this is not a particularly substantively significant even though it is statistically significant at a 99% confidence level.

Conclusions:

From the analysis conducted here I could not conclude that the public release of the college scoreboard 10 year median earnings had any impact on searches for colleges. While the coefficient on high earning universities was positive and significantly different from 0 at a 95% confidence level, it was not particularly substantive and the regression was fairly explanatorily weak. However a future analysis with dummy comparative searches, controls for college sport searches, and a larger after sample size could result in a more complete analysis.