Data.Exploration.Project.ECON.4110

ECON 4110 : Data Exploration Project Deadline : 5.14.23

Research Question: Among colleges that predominantly grant bachelor's degrees, did the release of the Scorecard shift student interest to high-earnings colleges relative to low-earnings ones (as proxied by Google searches for keywords associated with those colleges)?

1) Import, Clean, and Join Relevant Data

```
# Set wd
# load relevant packages.
library(rio)
library(lubridate)
```

```
Attaching package: 'lubridate'

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
library(stringr)
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✔ dplyr   1.1.2     ✔ readr   2.1.4
✔ forcats 1.0.0     ✔ tibble  3.2.1
✔ ggplot2 3.4.2     ✔ 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(vtable)
```

```
Loading required package: kableExtra

Attaching package: 'kableExtra'

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

```
    group_rows
```

```
library(fixest)
```

```
# Create list of files after setting wd to Rawdata.
files <- list.files(pattern = 'trends_up_to', full.names = TRUE)

# Use import_list to compile the list of files into a dataset.
data <- import_list(files, rbind = TRUE, fill = TRUE)
```

```
# Convert data to date format and aggregate by month.
data <- data %>%
  mutate(week = str_sub(monthorweek, start = 1, end = 10)) %>%
  mutate(week = ymd(week)) %>%
  mutate(month = floor_date(week,"month"))

# Aggregate by schname and keyword and standardize the index varaible.
data <- data %>%
  group_by(schname, keyword) %>%
  mutate(index.standard = ((index-mean(index))/sd(index)))
```

```
# Use import() to read in scorecard data.
Scorecard <- import('Most+Recent+Cohorts+(Scorecard+Elements).csv')
# Use names()[] to change the UNITID and OPEID column names to lower case.
  names(Scorecard)[1] <- "unitid"
  names(Scorecard)[2] <- "opeid"

#use import() to read in the id_name_link.csv file
id_name_link <- import('id_name_link.csv')
```

```
# Merge in the Scorecard Data.
  # Use group_by() and mutate(n = n()) to count how many times each school name pops up i
      # mutate(n = n()) creates a new variable, n, in the data frame that represents the
  # Then, filter() to get rid of any school names that show up more than once.
group_by(id_name_link, schname) %>%
  mutate(schname_count = n()) %>%
  filter(schname_count == 1)
```

```
# A tibble: 3,523 × 4
# Groups:   schname [3,523]
   unitid   opeid schname                                      schname_count
    <int>   <int> <chr>                                               <int>
 1 180203 2517500 aaniiih nakoda college                                  1
 2 222178  353700 abilene christian university                           1
 3 138558  154100 abraham baldwin agricultural college                    1
 4 172866 2050300 academy college                                         1
 5 412173 3346300 academy for nursing and health occupations              1
```

```
 6 108232  753100 academy of art university                          1
 7 475635 4185500 academy of couture art                           1
 8 126182  134500 adams state university                           1
 9 188429  266600 adelphi university                               1
10 188438  286000 adirondack community college                     1
# ℹ 3,513 more rows
```

```
# Use the "schname" variable to link up the Google trends data (data) to id_name_link.
# Then use the "unitid" or "opeid" columns to link THAT to the Scorecard data.
  # inner_join() can perform both of these links.
data_JOIN_id_name_link_BY_schname <- inner_join(data, id_name_link, by = 'schname')
```

```
Warning in inner_join(data, id_name_link, by = "schname"): Detected an unexpected many-
to-many relationship between `x` and `y`.
ℹ Row 2854 of `x` matches multiple rows in `y`.
ℹ Row 3591 of `y` matches multiple rows in `x`.
ℹ If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
Scorecard_JOIN_data_AND_id_name_link_BY_unitid <-
  inner_join(data_JOIN_id_name_link_BY_schname, Scorecard, by = "unitid")
Scorecard_JOIN_data_AND_id_name_link_BY_unitid
```

```
# A tibble: 1,617,730 × 133
# Groups:   schname, keyword [9,914]
   schid schname  keyword keynum monthorweek index `_file` week       month
   <chr> <chr>    <chr>    <int> <chr>       <int> <chr>   <date>     <date>
 1 0     young h… young …      1 2013-03-31…    34 ./tren… 2013-03-31 2013-03-01
 2 0     young h… young …      1 2013-04-07…    36 ./tren… 2013-04-07 2013-04-01
 3 0     young h… young …      1 2013-04-14…    45 ./tren… 2013-04-14 2013-04-01
 4 0     young h… young …      1 2013-04-21…    45 ./tren… 2013-04-21 2013-04-01
 5 0     young h… young …      1 2013-04-28…   100 ./tren… 2013-04-28 2013-04-01
 6 0     young h… young …      1 2013-05-05…    42 ./tren… 2013-05-05 2013-05-01
 7 0     young h… young …      1 2013-05-12…    38 ./tren… 2013-05-12 2013-05-01
 8 0     young h… young …      1 2013-05-19…    38 ./tren… 2013-05-19 2013-05-01
 9 0     young h… young …      1 2013-05-26…    33 ./tren… 2013-05-26 2013-05-01
10 0     young h… young …      1 2013-06-02…    40 ./tren… 2013-06-02 2013-06-01
# ℹ 1,617,720 more rows
# ℹ 124 more variables: index.standard <dbl>, unitid <int>, opeid.x <int>,
#   opeid.y <int>, opeid6 <int>, INSTNM <chr>, CITY <chr>, STABBR <chr>,
#   INSTURL <chr>, NPCURL <chr>, HCM2 <int>, PREDDEG <int>, CONTROL <int>,
#   LOCALE <chr>, HBCU <chr>, PBI <chr>, ANNHI <chr>, TRIBAL <chr>,
#   AANAPII <chr>, HSI <chr>, NANTI <chr>, MENONLY <chr>, WOMENONLY <chr>,
#   RELAFFIL <chr>, SATVR25 <chr>, SATVR75 <chr>, SATMT25 <chr>, …
```

2) Modify Data for Regression

```
# Filter for Bachelor Degree Universities using the variable name PREDDEG and set it to 3
Scorecard_JOIN_data_AND_id_name_link_BY_unitid <-
  filter(Scorecard_JOIN_data_AND_id_name_link_BY_unitid, PREDDEG == 3)
```

```
# Define high, average, low income
income.mean <- Scorecard_JOIN_data_AND_id_name_link_BY_unitid$`md_earn_wne_p10-REPORTED-E
  as.numeric %>%
  na.omit()%>%
  mean()
```

```
Warning in na.omit(.): NAs introduced by coercion
```

```
income.sd <- Scorecard_JOIN_data_AND_id_name_link_BY_unitid$`md_earn_wne_p10-REPORTED-EAR
  as.numeric() %>%
  na.omit() %>%
  sd()
```

```
Warning in na.omit(.): NAs introduced by coercion
```

```
income.high <- income.mean+income.sd
income.low <- income.mean-income.sd
tibble(income.low, income.mean, income.high)
```

```
# A tibble: 1 × 3
  income.low income.mean income.high
       <dbl>       <dbl>       <dbl>
1     31843.      43449.      55055.
```

```
#Define and clean variables of interest
income <- as.numeric(Scorecard_JOIN_data_AND_id_name_link_BY_unitid$`md_earn_wne_p10-REPO
```

```
Warning: NAs introduced by coercion
```

```
schname <- Scorecard_JOIN_data_AND_id_name_link_BY_unitid$schname
index <- as.numeric(Scorecard_JOIN_data_AND_id_name_link_BY_unitid$index.standard)
Date.bymonth <- as.Date(Scorecard_JOIN_data_AND_id_name_link_BY_unitid$month)
```

3) Create Regression Data Frame

```
# New df with only variables of interest
Plot.df <- data.frame(
  Date.bymonth = Date.bymonth,
  University.Name = schname,
  University.Mean.Income = income,
```

```
  University.Interest = index
)
```

```
# eliminate rows with NA values (privacy protected)
Plot.df <- Plot.df %>%
  filter(!is.na(University.Mean.Income) & !is.na(University.Interest))
```

```
# Date of Scorecard Introduction : 2015-09
# Create a new column "Scorecard.pre.post" based on the date condition
Plot.df <- Plot.df %>%
  mutate(Scorecard.pre.post = case_when(
    Date.bymonth <= as.Date("2015-09-12") ~ "Before",
    Date.bymonth > as.Date("2015-09-12") ~ "After"
  ))
```

```
# Create a set of variables that are 'High'  if the school is high income, 'Low' if low i
# neither high nor Low income, and 'Low' if the school is low income.
Plot.df <- Plot.df %>%
  mutate(University.Income.Category = case_when(
    University.Mean.Income >= income.high ~ 'High',
    University.Mean.Income >= income.low ~ 'Medium',
    University.Mean.Income <= income.low ~ 'Low')
    )
```

```
# Summarize mean interest over time per group. Save to new data frame to be used for OLS
Plot.df2 <- Plot.df %>% group_by(Date.bymonth, University.Income.Category) %>%
  summarize(Group.Mean.Interest = mean(University.Interest))
```

```
`summarise()` has grouped output by 'Date.bymonth'. You can override using the
`.groups` argument.
```

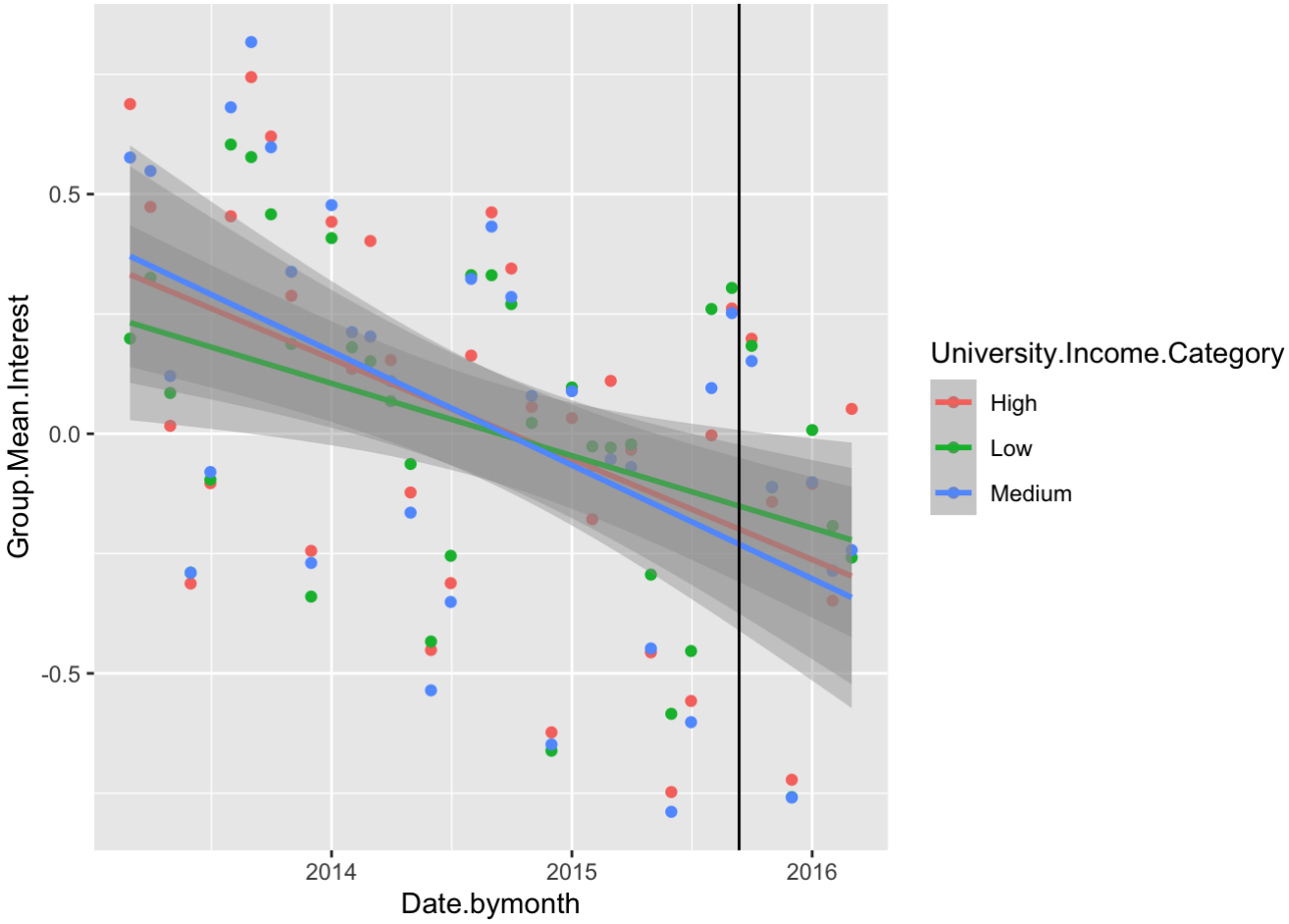
```
# Date of Scorecard Introduction : 2015-09
# Create a new column "Scorecard.pre.post" based on the date condition
Plot.df2 <- Plot.df2 %>% mutate(Scorecard.pre.post = case_when(
    Date.bymonth <= as.Date("2015-09-12") ~ "Before" ,
    Date.bymonth > as.Date("2015-09-12") ~ "After"
    ))
```

4) Plot Data

```
# Plot data
#gg point plot
Plot.point <- ggplot(Plot.df2, aes(x = Date.bymonth, y = Group.Mean.Interest, color = Uni
  geom_point() +
  geom_smooth(method = 'lm') +
```

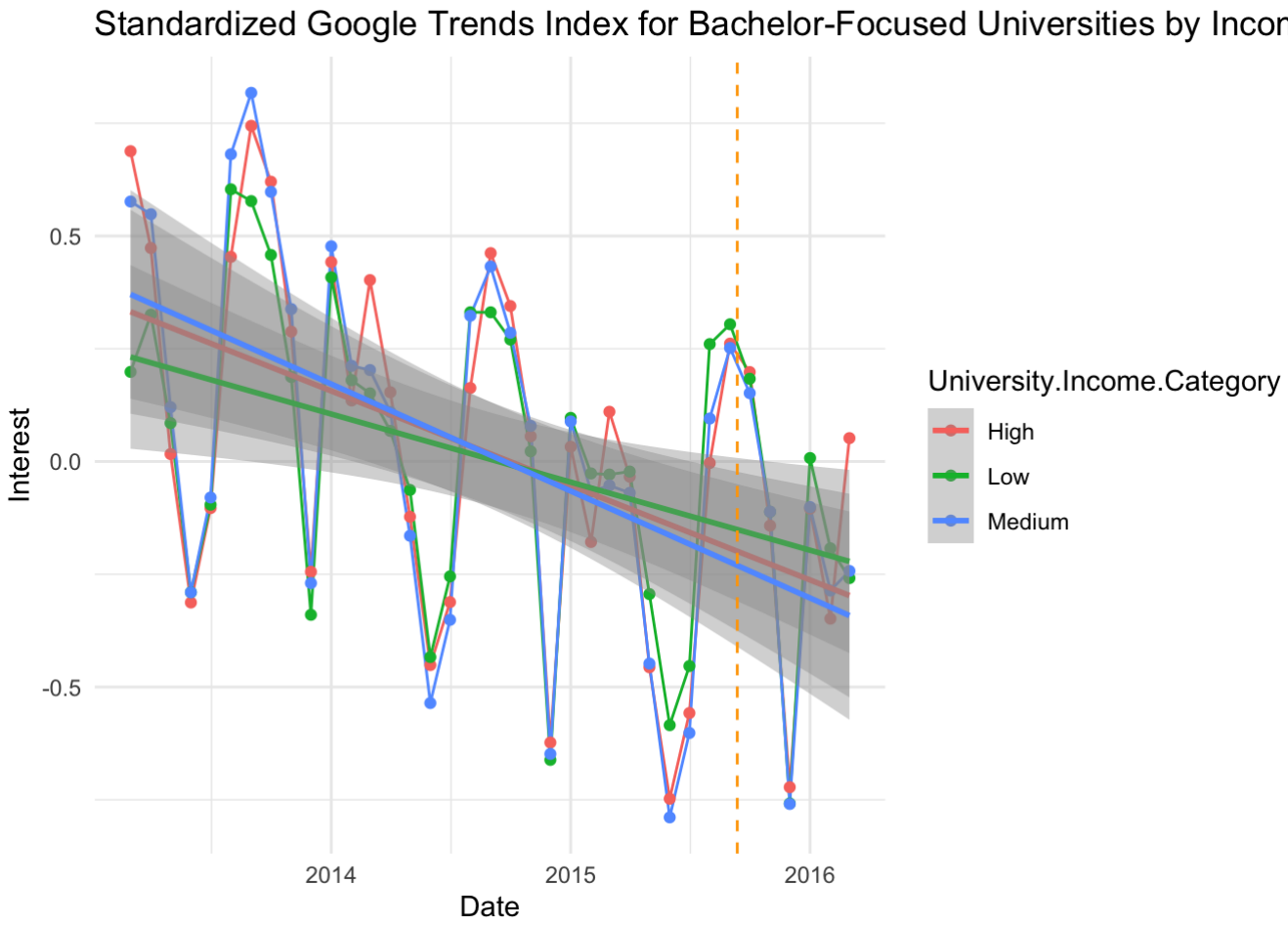
```
  geom_vline(xintercept = ymd('2015-09-12'))
Plot.point
```

```
`geom_smooth()` using formula = 'y ~ x'
```




```
# gg point and line plot
Plot.line <- ggplot(Plot.df2, aes(x = Date.bymonth, y = Group.Mean.Interest, color = Univ
  geom_line() +
  geom_point() +
  geom_smooth(method = 'lm') +
  geom_vline(xintercept = as.Date('2015-09-12'), color = "orange", linetype = "dashed") +
  theme_minimal() +
  labs(title = 'Standardized Google Trends Index for Bachelor-Focused Universities by Inc
Plot.line
```

```
`geom_smooth()` using formula = 'y ~ x'
```



6) Run Regression

```
# OLS Regression
# regress index on scorecard.
OLS.Reg.Plot.df2 <- Plot.df2 %>% feols(Group.Mean.Interest ~ i(Scorecard.pre.post, ref =
etable(OLS.Reg.Plot.df2)
```

```
                            OLS.Reg.Plot.df2
Dependent Var.:           Group.Mean.Interest

Constant                        0.0525 (0.0379)
Scorecard.pre.post = After  -0.2495** (0.0942)
___________________________ ___________________
S.E. type                                   IID
Observations                                111
R2                                      0.06049
Adj. R2                                 0.05187
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Control for time, and Income Category
```

```
OLS.Reg.Plot.df2 <- Plot.df2 %>% feols(Group.Mean.Interest ~ i(Scorecard.pre.post, ref =
```

```
The variables 'University.Income.Category::High', 'University.Income.Category::Medium'
and 'University.Income.CategoryMedium:i(Scorecard.pre.post, ref = "After")' have been
removed because of collinearity (see $collin.var).
```

```
etable(OLS.Reg.Plot.df2)
```

```
                                                                OLS.Reg.Plot.df2
Dependent Var.:                                               Group.Mean.Interest

Constant                                                        -0.1778 (0.1521)
i(factor_var=Scorecard.pre.post,ref="After")                    0.2851. (0.1661)
University.Income.CategoryLow                                   -0.0104 (0.2150)
University.Income.CategoryMedium                                -0.0470 (0.2150)
University.Income.CategoryHigh x i(Scorecard.pre.post,ref="After")  -0.0524 (0.2349)
University.Income.CategoryLow x i(Scorecard.pre.post,ref="After")   -0.0546 (0.2349)
_____________________________________________                 __________________
S.E. type                                                                  IID
Observations                                                               111
R2                                                                     0.06130
Adj. R2                                                                0.01661
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpret Regression and Concludes

Interpret Coefficients of Interest:

For the coefficient “University.Income.CategoryHigh x i(Scorecard.pre.post,ref=‘After’)”:

- The estimated coefficient of -0.0524 suggests that, when Scorecard.pre.post is in its “After” form, there is a negative interaction effect between the “High” level of University.Income.Category and Group.Mean.Interest.
- This means that for universities classified as “High” in terms of University.Income.Category, the effect of Scorecard.pre.post taking the “After” form is associated with a decrease in Group.Mean.Interest of, compared to the reference level (“Before”).

For the coefficient “University.Income.CategoryLow x i(Scorecard.pre.post,ref=‘After’)”:

- The estimated coefficient of -0.0546 suggests that, when Scorecard.pre.post takes the “After” form, there is a negative interaction effect between the “Low” level of University.Income.Category and Group.Mean.Interest.
- This means that for institutions classified as “Low” in terms of University.Income.Category, the effect of Scorecard.pre.post being in its “After” form is associated with a decrease in Group.Mean.Interest, compared to the reference level (“Before”).

However, none of the coefficients are statistically significant at the generally accepted levels, signifying that this effect is likely due to random chance, and the scorecard had no significant effect on searches.