Midterm Project - Data Cleaning

Shreeya Sampat

# Data Exploration Project:

In this project I will be compiling together some messy real-world data, and then designing an analysis to answer a research question.

### Research Question:

The College Scorecard was released at the start of September 2015. **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)?

### The Data:

The ZIP file Data\_Exploration\_Rawdata.zip contains a number of files:

* trends\_up\_to\_....csv: These are files generated using Google Trends. They are the Google Trends index for each keyword for the given monthorweek. Each keyword (indexed with keynum) is selected to be reflective of a university in the United States, given by schname. There are multiple files because the Google Trends API will kick you off if you make too many requests, and you have to start again.
* Most+Recent+Cohorts+(Scorecard+Elements).csv: This is data from the College Scorecard, a simple dataset that contains lots of information about United States colleges and the students that graduate from them. The variable names aren’t super helpful but they are documented in CollegeScorecardDataDictionary-09-08-2015.csv
* id\_name\_link.csv, which can be used to match colleges as identified in the Scorecard data (by unitid and opeid / UNITID and OPEID) with colleges as identified in the Google Trends data (by schname). The join functions will be helpful (see help(join) after loading the **tidyverse**)

# Load necessary libraries  
library(rio)  
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(lubridate)

Attaching package: 'lubridate'

The following objects are masked from 'package:base':  
  
 date, intersect, setdiff, union

library(stringr)  
library(ggplot2)  
library(fixest)  
library(broom)

# Use list.files(), probably with the full.names = TRUE option, to create a vector of filenames of all the trends\_up\_to\_ files are  
# Get all filenames for the Google Trends data  
trend\_files <- list.files(  
 # Folder where your files are  
 path = "Lab3\_Rawdata",  
 # Matches all Google Trends files  
 pattern = "^trends\_up\_to\_.\*\\.csv$",   
 full.names = TRUE   
)  
  
# Then, use import\_list() in the rio package to read in that vector of filenames, using rbind = TRUE to bind all the results together into a single dataset  
# Import all trends CSVs into a single dataset  
trends\_raw <- import\_list(trend\_files, rbind = TRUE)

Warning in (function (input = "", file = NULL, text = NULL, cmd = NULL, :  
Stopped early on line 1562. Expected 6 fields but found 5. Consider fill=TRUE  
and comment.char=. First discarded non-empty line: <<11,yti career institute -  
york,yti.edu,2,>>

Warning in (function (input = "", file = NULL, text = NULL, cmd = NULL, :  
Stopped early on line 1095. Expected 6 fields but found 5. Consider fill=TRUE  
and comment.char=. First discarded non-empty line: <<9,heidelberg  
university,heidelberg.edu,2,>>

Warning in (function (input = "", file = NULL, text = NULL, cmd = NULL, :  
Stopped early on line 1094. Expected 6 fields but found 5. Consider fill=TRUE  
and comment.char=. First discarded non-empty line: <<8,mount vernon nazarene  
university,mvnu.edu,2,>>

Warning in (function (input = "", file = NULL, text = NULL, cmd = NULL, :  
Stopped early on line 3280. Expected 6 fields but found 5. Consider fill=TRUE  
and comment.char=. First discarded non-empty line: <<41,potomac state college  
of west virginia university,potomac state college of west virginia  
university,1,>>

# Quick check  
glimpse(trends\_raw)

Rows: 1,198,340  
Columns: 7  
$ schid <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0"…  
$ schname <chr> "young harris college", "young harris college", "young har…  
$ keyword <chr> "young harris college", "young harris college", "young har…  
$ keynum <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…  
$ monthorweek <chr> "2013-03-31 - 2013-04-06", "2013-04-07 - 2013-04-13", "201…  
$ index <int> 34, 36, 45, 45, 100, 42, 38, 38, 33, 40, 43, 39, 29, 29, 3…  
$ `\_file` <chr> "Lab3\_Rawdata/trends\_up\_to\_finish.csv", "Lab3\_Rawdata/tren…

# Aggregating the Google Trends data  
  
# You're probably going to want to get an actual date variable to do aggregation and modeling. monthorweek is a string  
  
# Extract Start of Week and Convert to Date  
# Use str\_sub to get the first ten characters out of the monthorweek variable  
# Then, load the lubridate package and use the ymd() function to turn that into an actual usable date  
trends\_clean <- trends\_raw %>%  
 mutate(  
 week = str\_sub(monthorweek, 1, 10),   
 week = ymd(week)   
 )  
  
# If you want to aggregate the dates further, say to do months rather than weeks, you can use the floor\_date() function from lubridate. Setting the units to 'month', for example, will "round down" all dates in the same month to be the first of the month  
trends\_clean <- trends\_clean %>%  
 mutate(month = floor\_date(week, unit = "month"))  
  
glimpse(trends\_clean)

Rows: 1,198,340  
Columns: 9  
$ schid <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0"…  
$ schname <chr> "young harris college", "young harris college", "young har…  
$ keyword <chr> "young harris college", "young harris college", "young har…  
$ keynum <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…  
$ monthorweek <chr> "2013-03-31 - 2013-04-06", "2013-04-07 - 2013-04-13", "201…  
$ index <int> 34, 36, 45, 45, 100, 42, 38, 38, 33, 40, 43, 39, 29, 29, 3…  
$ `\_file` <chr> "Lab3\_Rawdata/trends\_up\_to\_finish.csv", "Lab3\_Rawdata/tren…  
$ week <date> 2013-03-31, 2013-04-07, 2013-04-14, 2013-04-21, 2013-04-2…  
$ month <date> 2013-03-01, 2013-04-01, 2013-04-01, 2013-04-01, 2013-04-0…

# Aggregating  
  
# We can't aggregate the index variable as-is since they're all on different scales  
# Use group\_by() and mutate() to standardize the index variable by school name and keyword (subtract the mean of index and then divide the result by the standard deviation of index, calculating both of those within school name and keyword)  
  
# Standardize the Index by schname and keynum  
trends\_standardized <- trends\_clean %>%  
 mutate(schname = stringr::str\_squish(stringr::str\_to\_lower(schname))) %>%  
 filter(  
 !is.na(schname), schname != "",  
 !is.na(month),  
 !is.na(index),  
 index >= 0, index <= 100  
 ) %>%  
 group\_by(keynum) %>%  
 mutate(  
 index\_std = as.numeric(scale(index)),   
 index\_std = ifelse(is.na(index\_std), 0, index\_std)  
 )  
  
# Now, a one-unit change in the standardized index can be understood and interpreted as a one-standard-deviation change in search interest  
   
# Now, if you want, you can use group\_by() and summarize() to aggregate your standardized index to the keyword-month level, or school-week level, or school-month level, or whatever you want  
  
# Aggregate to college-month  
trends\_agg <- trends\_standardized %>%  
 group\_by(schname, month) %>%  
 summarise(  
 avg\_std\_index = mean(index\_std, na.rm = TRUE),  
 .groups = "drop"  
 )  
  
glimpse(trends\_agg)

Rows: 91,112  
Columns: 3  
$ schname <chr> "aaniiih nakoda college", "aaniiih nakoda college", "aan…  
$ month <date> 2013-03-01, 2013-04-01, 2013-05-01, 2013-06-01, 2013-07…  
$ avg\_std\_index <dbl> 1.13905440, -0.96066157, -0.93536379, -0.40916991, -0.31…

dir.create("data", showWarnings = FALSE)  
rio::export(trends\_agg, "clean\_trends\_month\_school.rds")  
cat("Saved cleaned file as clean\_trends\_month\_school.rds\n")

Saved cleaned file as clean\_trends\_month\_school.rds