Data Exploration Project

Nick Huntington-Klein

2023-03-08

Data Exploration Project

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

We haven't done a lot of data cleaning in this class, but there is some in this project. The Data Cleaning Bonus Slides avialable on Canvas will help.

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**)

Notes about the Data

- There are multiple keywords per university, and the number of keywords is not the same across universities
- Google Trends shows the popularity of a given search term in Google over time.
- Google Trends indices are comparable *only to themselves*. That is, if the term "boston university" increases from 15 to 16 from one week to the next, and the term "seattle university" also went from 15 to 16, then we know that they both increased, but we don't know if the amount of increase was the same (even though they both increased by 1, that "1" could mean different amounts of searching for Boston U and Seattle U), or that they're currently both getting the same amount of attention (even though they both end up with 16, that could mean wildly

different amounts of searches for Boston and for Seattle). You can make the indices reasonably comparable by standardizing them (for each term, subtract the mean and divide by the standard deviation).

- There are multiple universities that share the same `schname`, which will mess up your ability to link Scorecard data to Google Trends data (or distinguish different universities within Google Trends). In the real world this problem would cause you many, many hours of work to distinguish them (it's possible, based on the order they appear in the Google Trends data). But instead for this assignment I will allow you to just **drop all universities that share an exact name with another university**.

The Analysis

The College Scorecard isn't just data for us - it's also treatment! The College Scorecard is a public-facing website that contains important information about colleges, including how much its graduates earn. This information has traditionally been very difficult to find.

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)?

You will need to produce at least one regression and one graph for your analysis, and explain them.

Things to Think About for Analysis

You're going to have to make some choices! Be sure to explain why you made them in your writeup.

- There is a variable in the Scorecard with information about the median earnings of graduates ten years after graduation for each college. But how can we define "high-earning" and "low-earning" colleges? There's not a single answer - be ready to defend your choice.
- What *level* should the data be at? You can leave the data as is, with one row per week per keyword. Or `group_by` and `summarize` to put things to one week per college, or one month per college, or one mont per keyword, etc. etc.
- How should the regression model be designed to answer the question (transformations and functional form? Standard error adjustments? etc.), and how can we interpret the results once we have them?

IMPORTANT PROJECT NOTES

- This project *will take some time*. It will take you *time* to put the data together. You will probably hit a snag and need to do some part of your work over again. *This is how all data projects work.* So give yourself time to make a mistake, or hit a wall, and need to go back to fix something. **If you start this project the week it is due, you will almost certainly be scrambling and turn in a poor product.** Like you would with any on-the-job project, give yourself a reasonable time frame. Try to have the data prepared two weeks ahead. Then the analysis one week ahead. Then the writeup you can do the week it's due. You won't need a whole week to do the writeup, you say? Probably not! But

this gives you the chance for the data preparation to take a little longer than expected… and the analysis to take a little longer than expected. **With any project you work on in your career, try to pad the schedule so you have time to fail.**

- Like many projects you will see in your career, this one asks you to answer a question. In this case it's "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" **Make sure your analysis actually answers this question.** It's tempting to just throw together a bunch of related variables and call that a model. But most of the time this won't actually answer the question you're interested in! Your boss won't be happy if they've told you to spend three weeks answering X and you come back with a report that instead answers A, B, and C. When you've got a model, ask yourself: (a) how will the results to this model answer the research question? and (b) does this model make sense? You can help yourself out here by interpreting your coefficients carefully. Without an interaction term this might be "This coefficient means that for every one-unit increase in A, controlling for the other variables in the model, we expect a (something)-unit increase in B." Does that sentence provide information about the research question? Continue that sentence with "and so the introduction of the Google Scorecard (does what?)" Also be sure your model makes sense! The dependent variable should be the outcome. When you "control for other variables" are those the variables you want to control for? Think this through! Use your common sense - think about what your model says and ask yourself if that's what you want! We've covered how to put a model together. You can also consult the Regression Flowchart (https://theeffectbook.net/ch-StatisticalAdjustment.html#turning-a-causal-diagram-into-a-regression). The textbook has plenty more on this topic.

The Writeup

You don't have to write a lot here. Just make an RMarkdown file that performs your analysis and displays the results, and in which you explain your analysis.

Be sure to:

- Include at least one regression and one graph
- Explain why you are performing the analysis you are performing, and the choices you made in putting it together
- Explain how your analysis addresses the research question
- Any additional analyses you did that led you to design your main analysis that way (i.e. "I graphed Y vs. X and it looked nonlinear so I added a polynomial term" - you could even include this additional analysis if you like)
- Explain what we should conclude, *in real world terms*, based on your results

There's no minimum or maximum length. I expect most analyses will be somewhere around two pages of text

Your Project

You will be turning in a link to a GitHub repo containing all your files. See Happy Git with R (https://happygitwithr.com/). If you just want to do the whole thing on your computer and then use the GitHub website's drag-and-drop file uploader, that's fine. Make sure you've given me access to your repo when you share it (my GitHub username is NickCH-K (https://github.com/NickCH-K/)).

Your repo should be an R project and should contain folders for, at a minimum, “data” and “code”. If you save your processed data it should go in a “processed_data” folder. (Note, if your working directory is “code”, you can access a file in “data” with `../data/'`).

The uploaded Git project should include a *rendered* Quarto document for your analysis in addition to its source code. Rendering to HTML, PDF, or Word are all acceptable. This knitted document can be in the same folder as your `.QMD` file, or you can have it render to the main folder, or to its own “analysis” or “paper” folder.

- If the data files won’t fit on the repo, you can leave them out.
- Make sure that the code chunks are *visible* in your knitted document
- If you want to have your data preparation code in its own script, and your analysis code in a separate RMarkdown, that’s fine. Be sure both of them are in the repo!

Notes about your code:

- Your code **should not contain any references to an absolute filepath.** Instead, set the working directory properly before running your code. You can “move up” a folder by using ‘../’. For example, if your working directory is in the ‘code/’ folder and your data is in ‘data/mydata.Rdata’, you can get to it using ‘../data/mydata.Rdata’.
- The data cleaning code may take a while to run and slow down your knitting. I expect most of you will probably want to have one code file (which you’ll only need to run once) that cleans your data and `export()` s it in clean form, and another file that just loads that file up and does all your analysis.
- When reading in the `trends_up_to` files, you **may not** explicitly write out every single one of the filenames. Instead, use `import_list()` from **rio** with the `rbind=TRUE` option, possibly passing it a list of filenames made using `list.files()`.
- You can isolate the first ten characters of a variable `x` using `str_sub(x,1,10)`. Just sayin’.

Grading Rubric

- 5% Satisfying the requirements listed on this page: Link to a GitHub repo with the appropriate file structure and a *knitted* document, including at least one regression and at least one graph, including explanatory text, following the directions throughout this document
- 20% Data cleaning: I will be reviewing your data cleaning code, checking for any mistakes that will lead to improperly-cleaned data in your analysis.
- 25% Choice and justification of analyses: Selecting and properly performing analyses that answer the research question well. Also, *explaining why you chose these analyses.*
- 20% Narrow interpretation of results: Properly interpreting, in text, the individual analyses you did. This particular grading category focuses on things like properly interpreting coefficients or individual results. Don’t be afraid to tell me the obvious - if you have a graph with time on the x-axis and a line going up, also saying “this graph shows an increase in (something) over time” in the text is a good idea. You don’t need to completely describe and interpret *every* coefficient in your model, but do include text that interprets the important ones.
- 20% Broad interpretation of results: Drawing reasonable conclusions about the research question that come specifically from your results. It’s a very good idea to have something at the end of the paper to the effect of “The impact of the Scorecard was (something), as we can tell

from (my analyses), since they show that (how we can come to that conclusion specifically from your results).” If you can do a good job filling that in, you’re in a good position.

- 10% Clear and engaging writing