Project 1: Wrangling, Exploration, Visualization

SDS322E

Data Wrangling, Exploration, Visualization

General Instructions

Your project will be completed in a single Rmd file called index.Rmd. Clone the following repository, which contains the instructions and template: https://github.com/nathanielwoodward/project1.git

When you have finished your project, save all changes and make sure the project knits to HTML correctly.

Submit a link to the Github repository containing your project to Canvas before the due date. Your project should also be live at your_username.github.io/project1 and linked to from the portfolio page on your website. See the last step (6) in this document for how to set this up. Failure to have this done before the deadline will result in a late deduction.

The text of your project document will provide a narrative structure around your code/output. All results presented must have corresponding code. Any answers/results/plots etc. given without the corresponding R code that generated the result will not be considered. Furthermore, all code contained in your final project document must work correctly (knit early, knit often)! Please do not include any extraneous code (code that produces warnings is acceptable, as long as you understand what the warnings mean!)

Find data:

Find two (!) datasets with one ID variable in common (e.g., dates, times, states, counties, countries, universities, athletes, zipcodes, movies/shows), both with at least 50 observations (i.e., rows) in each. Please think very carefully about whether it makes sense to combine your datasets! If you find one dataset with 50 medical patients and it has their age, and you find another dataset with 50 *different* patients that has their ages, it makes *no* sense to join them based on age (you would just be randomly pairing up different people's data who happened to be the same age).

When combined, the resulting/final dataset must have **at least 4 different variables (at least 3 numeric) in addition to the common variable** (i.e., five variables total).

You can have as many variables as you would like! If you found two datasets that you like but they don't have enough variables, find a third dataset with the same common variable and join all three!

Note that you will need to read these datasets into R at the top of your Rmd file like this, for example:

```r
library(tidyverse)
data1 <- read_csv("/path/to/dataset1.csv")
data2 <- read_csv("/path/to/dataset2.csv")

# or, if you'd prefer to read it in from some remote location

data1 <- read_csv("https://some.website/dataset1.csv")
data2 <- read_csv("https://some.website/dataset2.csv")
```

Guidelines

1. If the datasets are not tidy, you will need to reshape them so that every observation has its own row and every variable its own column. If the datasets are both already tidy, you will make them untidy with `pivot_wider()/spread()` and then tidy them again with `pivot_longer/gather()` to demonstrate your use of the functions. It's fine to wait until you have your descriptives to use these functions (e.g., you might want to `pivot_wider()` to rearrange the data to make your descriptive statistics easier to look at in tabular form); it's fine long as you use them at least once!

   - Depending on your datasets, it might be a good idea to do this before joining. For example, if you have a dataset you like with multiple measurements per year, but you want to join by year, you could average over your numeric variables to get means/year, do counts for your categoricals to get a counts/year, etc.
   - If your data sets are already tidy, demonstrate the use of `pivot_longer()/gather()` and `pivot_wider()/spread()` on all or part of your data at some point in this document (e.g., after you have generated summary statistics in part 3, make a table of them wide instead of long).

2. Join your 2+ separate data sources into a single dataset based on a common ID variable! If you can't find a good pair of datasets to join, you may split one main dataset into two different datasets with a common ID variable in each, and then join them back together based on that common ID, but this is obviously less than ideal!

   - You will document the type of join that you do (left/right/inner/full), including a discussion of how many observations/rows and distinct IDs were in each original dataset, which IDs appeared in one dataset but not the other, how many observations in each dataset were dropped (if any) after doing the join, and why you chose this particular join.

3. Create summary statistics

   - Use *all six* core `dplyr` functions (`filter, select, arrange, group_by, mutate, summarize`) to manipulate and explore your dataset. For mutate, create a new variable that is a function of at least one other variable, preferably using a dplyr vector function (see dplyr cheatsheet). It's totally fine to use the `_if`, `_at`, `_all` versions of mutate/summarize instead (indeed, it is encouraged if you have lots of variables). Use a `stringr` function such as str_detect or str_replace_all with regex at least once.
   - Using `dplyr`, create summary statistics (`mean, sd, var, n, quantile, min, max, n_distinct, cor`, etc) for each of your numeric variables. If you have lots of numeric variables (e.g., 10+), pick a few that are of interest and just summarize based on those. For your categorical variables, report the frequencies/counts of each level. Finally, report the number of missing values (NAs) for each variable.
   - Compute these statistics both overall and after grouping by one or more categorical variables (either together or one-at-a-time; if you have two categorical variables, include at least one statistic based on a grouping of two categorical variables simultaneously). If you do not have any categorical variables, create one using mutate (e.g., with `case_when` or `ifelse`) to satisfy the `group_by` requirements above. Compute at least one summary stat with a user-defined function rather than a built-in one.
   - Report these summary statistics in an easy-to-read tables (e.g., by reshaping, if your original datasets were tidy). Style at least one table using the `gt` or `kable` packages.

4. Make visualizations (three plots)

   - Create at least three plots of your choice with ggplot that highlight some of the more interesting features of your data.
   - Each plot should have at least two geom layers and at least two variables mapped to aesthetics
   - Each should make use of different geoms (don't make the same kind of plot twice)

- At least one plot should correctly make use of `stat="summary"`
- Each plot should include both a theme and scale modification
- Each should include a supporting paragraph describing the relationships being visualized and any trends that are apparent
- It is fine to include more, but limit yourself to 4. Plots should avoid being redundant!
- Make them pretty! Customize them!

Rubric

Prerequisite: Finding appropriate data from at least two sources per the instructions above: Failure to do this will result in a 0! You will submit a .Rmd file and a knitted document (html/pdf).

0. Introduction (5 pts)

- Write a narrative introductory paragraph or two describing the datasets you have chosen, the variables they contain, how they were acquired, and why they are interesting to you. Expand on potential associations you may expect, if any.

1. Tidying: Rearranging Wide/Long (10 pts)

- Tidy the datasets (using the `tidyr` functions `pivot_longer/gather` and/or `pivot_wider/spread`)
- If you data sets are already tidy, be sure to use those functions somewhere else in your project (e.g., for rearranging summary statistics)
- Document the process (describe in words what was done, no more than a paragraph)

2. Joining/Merging (20 pts)

- Join your datasets into one using a `dplyr` join function on an ID variable (or ID variables) common to both
- Discuss the process in words, including why you chose the join you did
- At a minimum, you should calculate (and your discussion should mention) the number of…
  - (1) total observations/rows in each dataset
  - (2) unique IDs in each dataset
  - (3) IDs that appear in one dataset but not the other (and which those are)
  - (4) IDs the datasets have common
- Discuss the size of the joined dataset and how it relates to the size of the original datasets
- In the joined dataset, note how many observations/rows were dropped, and any potential problems with this

3. Wrangling (46 pts)

- Use all six core `dplyr` functions in the service of generating summary tables/statistics (14 pts)
  - Use mutate at least once to generate a variable that is a function of at least one other variable
  - Use at least one `stringr` function with to do something with regex
- Compute summary statistics for each of your variables using `summarize` (fine to use variants such as `summarize_all`) alone and with `group_by` (if you have more than 10 variables, fine to just focus on 10 of them) (20 pts)
  - Use at least 5 unique functions inside of summarize (e.g., mean, sd)
  - For at least 2 functions, use summarize after grouping by a categorical variable. Create one by dichotomizing a numeric if necessary
  - For at least 1 function, define your own function and use it inside summarize
  - If applicable, at least 1 of these should group by two categorical variables
  - For each categorical variable, provide a table of counts for each level (e.g., by using `n()` inside `summarize` after group_by)

- Discuss the procedure and all (or the most interesting) findings/results in no more than two paragraphs (8 pts)
- Style at least one table with `gt` or `kable` packages (4 pts)

4. Visualizing (39 pts)

- Create 3 effective, polished plots with ggplot (30 pts)
  - Each plot should have at least 2 geometry layers and at least 2 aesthetic mappings
  - Each plot should have a title and clean labeling for all mappings
  - Modify the default theme and scales at least once per plot
  - For at least one plot, add more tick marks (x, y, or both) than are given by default
  - For at least one plot, use the stat="summary" function
- Write a supporting paragraph (for each plot) describing what the plot depicts and any relationships/trends that are apparent (9 pts)

5. Neatness, Holistic/Discretionary Points (5 pts)

- Keep things looking nice! Your project should not knit to more than 30 or so pages (probably closer to 10-20)! You will lose points if you print out your entire dataset(s), have terrible/hard-to-read formatting, etc. Imagine this is a polished report you are giving to your PI or boss to summarize your work researching a topic.

6. GitHub and GitHub Pages

- Push your cloned project1 folder up to your GitHub. It should contain index.Rmd (your finished project code), the knitted html file (index.html) and optionally, the instructions files. Here's how:

1. Create a remote repository (on your GitHub) called project1.

2. In the terminal, run the following code, changing your username accordingly

```
cd ~/project1 #cd to your project1 directory
git add .
git commit -m "message"
git remote set-url origin https://github.com/YOUR_USERNAME/project1.git
git push origin main
```

- Once it is up there, go to your remote repository and then to Settings > Options and scroll down to GitHub pages. Alternately, go to https://github.com/YOUR__USERNAME/project1/settings/pages. Then, select the main branch and click save. Finally, go back to your homepage repository, go to the Portfolio.Rmd page, and update the link for project1 to YOUR__USERNAME.github.io/project1. Add, commit, and push your updated homepage to GitHub for deployment.
- Your project MUST be live at YOUR__USERNAME.github.io/project1 to receive a grade. Failure to do this before the due date will result in a late deduction.

Where do I find data?

OK, brace yourself!

You can choose ANY datasets you want that meet the above criteria for variables and observations. I'm just sitting here but off the top of my head, if you are into amusement parks, you could look at amusement-park variables, including ticket sales per day etc.; then you could join this by date in weather data. If you are interested in Game of Thrones, you could look at how the frequency of mentions of character names (plus other character variables) and the frequency of baby names in the USA... You could even take your old Biostats data and merge in new data (e.g., based on a Google forms timestamp).

You could engage in some "me-search": You can request your Spotify data or download Netflix viewing activity, Amazon purchase history, etc. You can use your Google Fit/Fitbit/Apple watch data, etc. These can be combined (e.g., with each other, with other data sources).

You can make it as serious as you want, or not, but keep in mind that you will be incorporating this project into a portfolio webpage for your final in this course, so choose something that really reflects who you are, or something that you feel will advance you in the direction you hope to move career-wise, or something that you think is really neat. On the flip side, regardless of what you pick, you will be performing all the same tasks, so it doesn't end up being that big of a deal.

If you are totally clueless and have no direction at all, log into the server and type

```
data(package = .packages(all.available = TRUE))
```

This will print out a list of **ALL datasets in ALL packages** installed on the server (a ton)! Scroll until your eyes bleed! Actually, do not scroll that much... To start with something more manageable, just run the command on your own computer, or just run `data()` to bring up the datasets in your current environment. To read more about a dataset, do `?packagename::datasetname`.

If it is easier for you, and in case you don't have many packages installed, a list of R datasets from a few common packages (also downloadable in CSV format) is given at the following website: https://vincentarelbundock.github.io/Rdatasets/datasets.html (including types/numbers of variables in each)

- A good package to download for fun/relevant data is `fivethiryeight`. Just run `install.packages("fivethirtyeight"` `load the packages with`library(fivethirtyeight)`, run`data()‘, and then scroll down to view the datasets. Here is an online list of all 127 datasets (with links to the 538 articles). Lots of sports, politics, current events, etc: https://cran.r-project.org/web/packages/fivethirtyeight/vignettes/fivethirtyeight.html
- If you have already started to specialize (e.g., ecology, epidemiology) you might look at discipline-specific R packages (vegan, epi, respectively). We will be using some tools from these packages later in the course, but they come with lots of data too, which you can explore according to the directions above
- However, you *emphatically DO NOT* have to use datasets available via R packages! In fact, I would much prefer it if you found the data from completely separate sources and brought them together (a much more realistic experience in the real world)! You can even reuse data from your SDS328M project, provided it shares a variable in common with other data which allows you to merge the two together (e.g., if you still had the timestamp, you could look up the weather that day: https://www.wunderground.com/history/). If you work in a research lab or have access to old data, you could potentially merge it with new data from your lab!
- Here is a curated list of interesting datasets (read-only spreadsheet format): https://docs.google.com/spreadsheets/d/1wZhPLMCHKJvwOkP4juclhjFgqIY8fQFMemwKL2c64vk/edit
- Here is another great compilation of datasets: https://github.com/rfordatascience/tidytuesday
- Another good general place to look: https://www.kaggle.com/datasets
- Here is the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php
  - See also https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research#Biological_data
- To help narrow your search down or to see interesting variable ideas, check out https://www.tylervigen.com/spurious-correlations. This is the spurious correlations website, and it is fun, but if you look at the bottom of each plot you will see sources for the data. This is a good place to find very general data (or at least get a sense of where you can scrape data together from)!

- If you are interested in medical data, check out www.countyhealthrankings.org
- If you are interested in scraping UT data, the university makes *loads* of data public (e.g., beyond just professor CVs and syllabi). Check out all the data that is available in the statistical handbooks: https://reports.utexas.edu/statistical-handbook

Broader data sources: Data.gov 186,000+ datasets!

Social Explorer is a nice interface to Census and American Community Survey data (more user-friendly than the government sites). May need to sign up for a free trial.

U.S. Bureau of Labor Statistics

U.S. Census Bureau

Gapminder, data about the world.

...