STAT 209: Project 2 Description

NOTE: As with project 1, please have *exactly one* person in your group create project repo through the GitHub Classroom link here (https://classroom.github.com/g/i2NtnOm9), and when asked to create a team, please name your team simply `group1`, `group2`, etc. (no capital letters, spaces, or punctuation). Then, the other team members should use the same link and join the existing group. This will do a few things: (1) It gives me easy access to your repos without you explicitly having to invite me, as well as the ability to check out your repos via script without much if any hand-tuning,and (2) It makes your repository private, since it is part of my `ocstats` organization which has some extra features.

Goals

The goal of this project is to leverage your newfound data-wrangling skills to enable you to create data visualizations that you would not have been able to create without the prerequisite wrangling.

Required skills

- Proficiency with the `ggplot2` package
- Proficiency with the "five main verbs" of the `dplyr` package
- Proficiency with join operations from the `dplyr` package
- Proficiency with the restructuring verbs `gather` and `spread` from the `tidyr` package
- Proficiency with the GitHub workflow

Relevant background and resources

- Chs. 2-5 of the textbook
- Slides and labs through Lab 10
- `ggplot2` Quick Reference (http://colindawson.net/stat209/resources/ggplot2-quick-reference.pdf)
- RMarkdown Quick Reference (http://colindawson.net/stat209/resources/rmarkdown-quick-reference.pdf)
- `dplyr` Quick Reference (http://colindawson.net/stat209/resources/dyplyr-quick-reference.pdf)
- `tidyr` Quick Reference (http://colindawson.net/stat209/resources/tidyr-quick-reference.pdf)

Instructions

The final writeup

The final result of the project will be much the same as in Project 1: Your group will work together to write a blog post that contains one or more data graphics that tell the reader something interesting about the domain that the data comes from. The same examples as before apply:

- Typical ages of people with common female names (https://fivethirtyeight.com/features/how-to-tell-someones-age-when-all-you-know-is-her-name/)
- Quality of various flights and airports (https://fivethirtyeight.com/features/fastest-airlines-fastest-airports/)
- Learning about New Yorkers' lives via taxi and Uber records (http://toddwschneider.com/posts/analyzing-1-1-billion-nyc-taxi-and-uber-trips-with-a-vengeance/)
- People admitted to the ER for wall-punching (https://qz.com/582720/americas-most-prolific-wall-punchers-charted/)

Since you have more tools at your disposal to slice and dice data in creative ways, the writeup can be a little bit longer than before: say up to 1000 words (not counting the Methods section).

The suggested structure of the writeup is much the same as before:

- An introduction that sets up the context and introduces the datasets and their sources. Tell the reader what the units of observation ("cases") are, and what the relevant variables are. Don't just point the reader to the data via a link: describe it in the text! My impression was that the intros were a bit terse last time (probably because I alluded to "an introductory paragraph"), so I encourage you to give a bit more context this time.
- For each graphic you include, a paragraph or two discussing what the graphic shows, including a concise "takehome" message in one or two sentences
- A "methodology" section (separate from the body of the main blog post, but included in the same `.Rmd` as an "Appendix" section) that explains both the data-wrangling component (how did you get your data in a form such that you could visualize what you wanted to visualize) and the visualization component (why did you choose the type of graph (geometry) you did, why did you choose the aesthetic mapping you did, why did you choose the color scheme you did, etc.). Note: In addition to describing your choices in the methodology section of the writeup, you should include comments in your code explaining what each step does and why.
- A discussion section that ties together the insights from the various views of the data you have created, and suggests open questions that were not possible to answer in the scope of this project (either because the relevant data was not available, or because of a technical hurdle that we have not yet learned enough to overcome)

Data-Wrangling Requirements

- (At least some of) your visualizations should pull together data from multiple data tables via some form of `*_join()` operation.
- In addition, your project should make non-trivial use of the five main "single table" verbs: `filter()`, `select()`, `mutate()`, `arrange()`, and `summarize()` (likely together with `group_by()`). The use of these verbs should be natural and motivated by the desired visualization, but I envision that you will make use of at least one `group_by()` and `summarize()`, as well as at least two other verbs in combination.

- As appropriate, you should take advantage of the ability to reshape your data as appropriate for your visualizations using `gather()` and/or `spread()` operations.
- Think about places where you are carrying out repetitive work, and consider whether your workflow could benefit from the use of custom functions or iterative constructions (such as `do()` or `lapply()`).

Data Visualization Requirements

Your writeup should consist of at least as many graphs as there are group members; but there's a good chance you'll want to include more than that (however, quality is more important than quantity). You will need to make a judgment call about when it is better to overlay information in one graph and when it is better to put the information in separate graphs.

As before, you can split up the work however you choose, but a natural split might be for each person to be responsible for one or two graphs.

Naturally the graphs must be created within your `.Rmd` using `ggplot2`, and you should endeavor to do whatever wrangling is necessary in a separate pipeline, so that the visualization pipeline is as well structured and concise (and readable!) as possible.

Data

You can use any data source you want, except that you may *not* use a dataset we have used in a lab, nor can you use a dataset that is included in an R package (at least, not without running it by me first). For full credit, you will need to join data from more than one table. The tables you join need not be from the same source; for example, you could join data about the same geographical entities, or time periods, etc., from different sources.

Some possible sources for data are (as before):

- The federal government's Data.gov (http://data.gov) site
- The American Psychological Association (http://apa.org/research/responsible/data-links.aspx)
- The data science competition Kaggle (https://www.kaggle.com/datasets)
- The UC Irvine machine learning repository (http://archive.ics.uci.edu/ml/index.php)
- The Economics Network (http://www.economicsnetwork.ac.uk/data_sets)
- The Gapminder (https://www.gapminder.org/data/) database

The GitHub workflow

- See the top of this document for instructions on initializing your GitHub repo. Make sure you create the repo from the link provided; otherwise I will not see it!
- Next, one person should creating a new RStudio project from version control, pasting in the repo's URL (ending in `.git`).
- That person should create a new master `.Rmd` file, and commit and push all the new files to GitHub. For consistency, name the `.Rmd` file simply `project2.Rmd`.
- All other team members should then pull the repo.
- Each time you sit down to work on the project, *pull* before you do anything else. This will save you headaches.
- Whenever you make an edit *to any file* and want to save it, *pull first*, then *stage* and *commit*. If you're ready to share it with your group, perform a *push*.
- If you get an error upon pulling, likely it is because a file you have edited was changed by someone else, and GitHub couldn't figure out how to reconcile the changes. You may need to go into the file and manually resolve the changes, then commit to merge them in the repo.

- If you get an error upon committing or pushing, you may have forgotten to pull first. If not, you may need to resolve a conflict manually by going into the relevant file(s) and manually editing them to merge the changes. **If this happens, notify your group members that you are undertaking a manual merge, so they do not continue to make edits in the mean time!**
- Make sure you have coordinated who is doing what when with your group, to minimize the above sorts of problems.

Turning in your project

Collaboration on your project should take place via GitHub commits. Your `.Rmd` source should be part of a GitHub repo from its inception, and changes should be recorded via commits from the account of the person who made the edit. Everyone in the group must make at least one commit.

Your final submission will consist of the `.Rmd` source, compiled `.html`, and any other files needed for the `.Rmd` to compile successfully. For example, if you are reading in the data from a `.csv` file stored locally (that is, in your RStudioPro server account), commit this file, and make sure that you are using a relative path to the file when you read in the data. If you are reading the dataset directly from an R package or from a URL, this is not necessary.

Whatever state those files are in at the deadline is what I will grade.

Grading Rubric

This project is worth a total of 10% of the course grade. There is a group component and an individual component to the grade, each weighted equally (5% each).

The typical division of labor is that each group member is individually responsible for at least one graphic, along with the part of the writeup and methodology section directly pertaining to that graph, and the group as a whole works together to write and edit the general introduction and conclusion, along with any components of the Methods section that pertain to the project as a whole. Your group may choose to divide the work differently, but be sure that each person is involved in the topic selection and planning stage, the coding component, the “general audience” writing element, and the “technical writing” element.

Group grade: Baseline Criteria (3/5 credit)

Wrangling:

- information from at least two different data tables is integrated in the same graph
- the basic data-wrangling verbs are employed to productive effect (you might not use all of them, but you should use them when it is called for)

Visualization:

- at least two related graphics are included
- the graphics are generated by the code embedded in the `.Rmd` (not included from an external file)

Markdown:

- the `.Rmd` compiles successfully
- Code, unnecessary messages, and raw R output (other than the plots) are suppressed from the `.html` output

Version Control:

- there is a GitHub record of commits, with informative commit messages

Writing:

- the motivation and goals of the blog post are laid out clearly at the beginning of the writeup
- suitable datasets are chosen, and the visualizations chosen fit together to illustrate interesting features about the data
- a description of the dataset is provided in the introduction, along with contextual information
- there is a description of "big picture" insights gained from considering the visualizations as a set
- the post is a reasonable length (~1000 words of text, not including the Methodology section)

Communicating Methodology:

- a description of the technical elements of your project is included in a separate methodology section at the end

Group grade: Finishing touches (5/5 credit)

- The above, plus:

Wrangling:

- Data is converted to "long" or "wide" format using `gather()` and `spread()` if it would be useful to do so
- The data wrangling pipeline is elegant; messy workarounds are avoided where possible (they may not be completely avoidable, but should be minimized)
- Copy/paste repetition of similar code is avoided in favor of functions and/or iterative constructions

Visualization:

- The choices made are effective and allow information to be conveyed clearly and efficiently

Writing:

- The writeup is organized well, making it easy to follow your narrative, and looks professional and polished
- The methodology section is clearly organized, and leaves the reader with enough understanding of your process that, given sufficient familiarity with data-wrangling and visualization, they could reproduce much of your work without seeing your code.

Code Style:

- The code in the `.Rmd` file is well documented, and a consistent coding style is followed (e.g., line breaks and indentation are used in an intentional and purposeful way to improve code readabilty; a consistent variable naming scheme is followed)

Group grade: Above and beyond (6/5 credit)

Wrangling:

- The data wrangling process seems particularly elegant, including some particularly clever and non-obvious uses of wrangling verbs

Writing:

- The writeup could be mistaken for an article in a prominent data journalism outlet

Visualization:

- The graphics used collectively convey particularly surprising or subtle aspects of the data that would have been difficult to notice with any single view

Individual grade: Baseline Criteria (3/5 credit)

Good Faith:

- Your group members report that you were a meaningful participant in planning the project, and made meaningful contributions to the "collective" parts of the project.
- You are present for all in-class workshop and presentation days, or if is is unavoidable for you to miss one, you make up for it by doing some extra work for your group outside class
- You are present for all out-of-class group planning sessions, or if you have to miss one, you make up for it later

Wrangling:

- The data-wrangling you did for your graph is clean and efficient, and your code is readable.

Visualization:

- Your graphic includes relevant context (title, axis labels, etc.)
- The graphic you were responsible for is constructed using clean `ggplot2` code

Writing:

- The interpretation of your individual graphic fits with what is shown in the graphic, and the graphic illustrates what you say it does
- Your contributions to the writing consist of complete, grammatical English sentences
- The visual (aesthetic) mapping is motivated in your piece of the Methods section

Version Control:

- Your individual contributions are clearly documented with commits from your account in GitHub

Individual grade: Full Contributor (5/5 credit)

Good Faith:

- Your group members report that you were at least a co-equal participant in all phases of planning and execution of the project

Wrangling:

- Your graph required some creative combinations of wrangling verbs.

Visualization:

- Your graphic displays aspects of the data that are not easily seen in the other graphics in the writeup
- Your graphic employs thoughtful choices for geometries, aesthetic mappings, color palettes

Writing:

- Your interpretation exhibits insight into non-trivial aspects of the data
- Your writing is clean, concise, and easy to follow.

Individual grade: Above and beyond (6/5 credit)

Wrangling:

- Your wrangling pipeline is particularly elegant – you found creative and effective ways to combine wrangling elements to suit your needs that aren't just parallels to things we've done in class and labs.

Visualization:

- Your individual graphic displays especially original, subtle or surprising insights into the data