Lab 0

James Duncan (based on materials by Tiffany Tang, Zoe Vernon, Rebecca Barter, Yuval Benjamini, Jessica Li, Adam Bloniarz, and Ryan Giordano)

Due 9/4/2020

Note: This lab is not representative of the labs that you will receive in this class. Future labs will be significantly more open-ended and difficult.

This lab will not be for a grade; you do not have to complete the lab if you don't want to, but you do need to submit *something* on GitHub (even if it is a blank `lab0.Rmd` and `lab0.pdf` file). This lab is an opportunity to make sure that you know how to submit your assignments, and for you to learn a little bit of Git/GitHub and R/tidyverse. If you do not have extensive experience using R/tidyverse previously, I recommend attempting to complete this lab.

Install R and RStudio

Install R from CRAN (https://cran.r-project.org/) and RStudio from RStudio (https://www.rstudio.com/products/RStudio/).

Install the tidyverse package in R

In the RStudio console, install the tidyverse package

```r
# you only ever have to run the following once:
install.packages("tidyverse")
```

The best resource at the moment for learning the tidyverse is the book R for Data Science (http://r4ds.had.co.nz/) by Garrett Grolemund and Hadley Wickham. For more advanced topics, Advanced R (https://adv-r.hadley.nz/) by Hadley Wickham is a nice reference. I also find the tidyverse website (https://www.tidyverse.org/) helpful, but it is probably not the place to start learning.

The tidyverse is actually a bundle of packages:

- `ggplot2` for visualization
- `dplyr` for data manipulation (SQL-style)
- `tidyr` for reshaping data (wide-form to long-form and vice verse)
- `readr` for loading data from a variety of formats
- `purr` for performing functional programming operations (e.g. maps to replace for-loops)
- `tibble` a more flexible alternative to data frames

The most important packages are `ggplot2` and `dplyr`, so if you decide to learn anything, learn these!

Other useful packages include:

- `lubridate` for dealing with dates
- `forcats` for dealing with factors

When writing code, you should follow the Google R Style Guide (https://google.github.io/styleguide/Rguide.xml), which is a slight modification of the Tidyverse Style Guide (https://style.tidyverse.org/). Please take a look at the Google R Style Guide as well as Part 1 of the Tidyverse Style Guide.

Analysis Instructions

Write up a report conducting the following analyses using R Markdown (if you prefer markdown) or R Sweave (if you prefer raw LaTeX). Note that both R Markdown and R Sweave can both can handle LaTeX equations contained within $ (inline) or $$ (new line) symbols.

This walkthrough will be a quick overview of important functions/tools that you may find useful in future labs. If you are not familiar with R/Tidyverse, this lab is highly recommended.

Loading the data

1. If you have not set up your Github account (which is totally OK, as I will walk you through this process during the first lab section on August 28), download the `statecoord.txt` file from Bcourses.
2. If you have already set up your Github account, clone my stat-215a-fall-2020 repo by typing in the terminal (`git clone https://github.com/jpdunc23/stat-215a-fall-2020`) to get the class materials and data for this lab. These will live in the `lab0/` folder. If you have already cloned this repo, you can instead just pull any changes from the stat-215a-fall-2020 github repo (`git pull`).
3. Open RStudio and load the data USArrests in R (`data("USArrests")`).
4. Load the `statecoord.txt` data file into R.
5. Load in libraries from tidyverse via `library(tidyverse)`.

Manipulating the data

1. Merge the two datasets together into a single data frame (using the `join()` functions from `dplyr`. Type `?dplyr::full_join`), and name the resulting data frame `arrests`. Check that this worked correctly.

Visualizing the data

1. Plot “Murder” vs “Assault” using `ggplot()` and the `geom_point()` function. What do you see?
2. Plot “Rape” vs “urban population” using `ggplot()` and `geom_point()`. There should be an outlier. Mark the outlier with a different color.
3. Re-make these plots with the state names instead of the points (use `geom_text()`). Do you notice anything interesting?
4. Challenge exercise: Plot a map of the US colouring each state by its “Murder” rate. Check out `geom_polygon()`

Regression

You can fit a linear regression using the `lm()` function (or manually if you’d prefer!).

1. Remove the “murder” and “assault” columns from the `arrests` data frame (use `dplyr::select()`).
2. Fit a linear regression of urban population on “Rape”.
3. Plot predicted values versus the residuals. Do you see any trends?
4. Replot “Rape” vs urban pop and draw a blue line with the predicted responses.
5. Now refit without the outlier and add a red line on the same plot.

6. Compare the lines. Are the linear responses a good description of the data?
7. Make a publishable graph. Add a header (`ggtitle`), axis labels (`xlab` and `ylab`) and customize the legend (`scale_color_manual`).

Submit the lab

When you have completed Lab 0 (within a folder called `lab0/`), add, commit and push your changes to your stat-215-a Github repository.

The `lab0/` folder (a sub-folder of `stat-215-a/`) should have the following structure:

```
lab0/
  data/
  documents/
  lab0.Rmd
  lab0.pdf
  lab0_blind.Rmd
  lab0_blind.pdf
  R/
  other/
```

Again, you do not have to complete this lab, but at the very minimum, you must push a blank lab0.Rmd and lab0.pdf file to GitHub, so I can make sure I can see your repository for future labs.