Project Objectives

This is a solo project where you will create a shiny app to allow the user to explore some data!

Setting Things Up: Creating the Repo & Signing up for an Account on shinyapps.io

The first step is to create a github repo. All project work should be done within this repo so we can track your activity. You should have at least eight commits on the repo or else you will lose credit. This commits should be spaced out as you work on the project. When you add an element to your app and it works, push up the change!

We won't go through the usual process of creating a webpage for this repo.

You should have a `README.md` file on your repo with a brief description of the app and its purpose the (you can edit this file in RStudio or the github web editor).

You'll do all your work on github and, with your final version, use RStudio to deploy your app on shinyapps.io. This requires you to sign up for an account with them. You can use your NC State Google account, that's probably the easiest thing to do. You should sign up for the **free tier** that allows five applications and 25 active hours (unless you'd like to pay or already have an account).

You should turn in a link to your deployed shiny app and a link to your github repo.

Find Data You Are Interested In

For this project I'm going to let you choose your data from the list below:

- Census microdata (using the `tidycensus` package or your own functionality - remember this data has weights that must be accounted for. You could download a large chunk of data and then read that in for the purposes of your app rather than querying the API within the app.)
- Melbourne Housing Data
- Seoul Bike Sharing Data
- Store sales data
- Banking Data - combine the training and test sets if you use this
- NFL play-by-play Data
- NBA play-by-play
- Mobile Device Usage

If you have another data source you'd like to use, please let me know and we can discuss.

Prepare for Your App

Once you have your data selected, you'll want to have some ideas about what might be interesting to explore with the data. **Create a `.R` or `.qmd` file (this should be included in your repo)** where you do the following in a 'static' manner:

- Produce numerical and graphical summaries to investigate the data. These should include:
  - One-way and Two-way contingency tables

  - Numerical summaries (means, medians, sds, etc.) for quantitative variables at levels of categorical variables
  - Create at least six plots.
    * At least four of these should display multivariate information via the type of graph, by utilizing coloring, grouping, etc.
    * All plots should have nice labels and titles.
    * Some kind of faceting should be used somewhere.
    * At least two plots should be a plot that we didn't cover in class (say a heatmap or something like that - depends on your data - lots of good examples to consider here https://exts.ggplot2.tidyverse.org/gallery/)

App Requirements

Ok, now you're going to put what you did there into a shiny app! First, some requirements to consider:

- Your app will use a sidebar layout (I don't care if you use the built in one or if you use the one from `bslib` or another package).
  - On the sidebar, you will have widgets that allow the user to subset the data
    * Choose (at least) two categorical variables they can subset from. If there are groups of categories that make sense to have them choose for a given variable, that's fine. That is, they don't need to be able to choose any level of the each variable to subset. The user should be able to select all levels as well.
    * Give the user a way to select a numeric variable. When selected, use a dynamic UI method to create a slider (with two values) so they can subset on that variable if they choose.
    * Repeat the previous for a second numeric variable.
    * Use an `actionButton()` that, when pressed, subsets the data according to the selections made on the sidebar. (The data should not update unless this button is pressed.) **You should be using either a `reactive()` or `reactiveValues()` object on the server side to subset your data appropriately and pass it to subsequent sections of the main panel (see below).**
- In the main panel, you should have different tabs outlined below (I don't care which package or functions you use to create the tabs, that's up to you.)
  - An `About` tab (the default tab to start on). The tab should
    * Describe the purpose of the app
    * Briefly discuss the data and its source - providing a link to more information about the data
    * Tell the user the purpose of the sidebar and each tab (main panel section) in the app
    * Include a picture related to the data (for instance, if the data was about the world wildlife fund, you might include a picture of their logo)
  - A `Data Download` tab. The tab should
    * Display the data using `DT::dataTableOutput()` with `DT::renderDataTable()`
    * The data should be subsetted when the user selects a subset (and clicks the previously mentioned button)
    * Save the (possibly subsetted) data as a file (`.csv` is fine but whatever you'd like) by using a `downloadButton()` (see here)
  - A `Data Exploration` tab. The tab will allow the user to obtain the numeric and graphical summaries noted from 'Prepare for Your App' section

    * I want you to design how this part looks - whether you use subtabs, dynamic UI elements, etc.
    * The data should be subsetted when the user selects a subset (and clicks the previously mentioned button)
    * The user should be able to choose to display the categorical data summaries or the numeric variable summaries (you can choose to display graphs and numbers together or separately).
    * The user should be able to select which (categorical or numeric) variables they are summarizing and which variables are modifying the summary where appropriate.
        · For instance, they should be able to select numeric variable to find the means, medians, sds, etc. and select the categorical variable that these summaries should be found across. For plots, they should be able to select which variable might be on the x or y axis, the variable used for coloring, those kinds of things!
    * You should account for any error messages that pop up while changing the widgets (as done in the notes) and use loading spinners or other methods to display for any plot that takes a while to render.

- Lastly, you should have some dynamic text used at some point. That is, somewhere you should be rendering text from the server side.

Submission

You should submit your two URLs: shinyapps.io URL and the github repo URL.

Notes

- `aes_string()` is useful when trying to use columns selected by the user in a `ggplot()` plot since the columns are often returned as a string (this is no longer recommended but I don't mind if you use this)
- Similarly, `!!sym(input$...)` can also be useful to deal with this type of issue
- `get()` comes in handy some times as well

Rubric for Grading (total = 100 points)

Item	Points	Notes
Sidebar subsetting	20	Worth either 0, 2.5, . . . , 20
About tab	10	Worth either 0, 2.5, . . . , 10
Data download tab	10	Worth either 0, 2.5, . . . , 10
Data Exploration tab	35	Worth either 0, 2.5, . . . , 35
Static file for data exploration	10	Worth either 0, 2.5, . . . , 10
GitHub readme	5	Worth either 0 or 5
Deployment on shinyapps.io	10	Worth either 0 or 10

Notes on grading:

- For each item in the rubric, your grade will be lowered one level for each each error (syntax, logical, or other) in the code and for each required item that is missing or lacking a description.

- If your work was not completed and documented using your github repo you will lose up to 50 points.
- You should use Good Programming Practices when coding (see wolfware). If you do not follow GPP you can lose up to 50 points on the project.