Lecture 7: Data Wrangling with dplyr

Problem Set for Lecture 7

These questions will assist in bolstering your understanding of the material in Lecture 7. Emphasis should be placed on having the correct code/output as well as communication. The deliverable should be a knitted RMarkdown document to pdf without any code running off the page. You will be able to present these in office hours as an oral assessment interview along with the problem set for Lecture 5 and Lecture 6.

***** First, make sure you have read the Lecture Material's writeup *****

1. Use the `wine` dataset in the `asbio` library to answer the following questions. If you are having trouble accessing this library/dataset send me an email and I can send you a csv file to import to work in order to carry out the following questions. When you are working these steps out, only display the first 10 observations to show your results.

   (a) Rename the columns in dplyr to something that makes more sense (read the documentation!!!) and then save this as a new dataframe so you do not need to "carry" the renaming code to each future step. You will use this new dataframe in all of the following parts.

   (b) Filter the dataset so only the observations of wine quality 5, 6, and 8 are present.

   (c) Only display the columns which deal with Y through X4 and X10 in the original naming system

   (d) Create a new column which looks at the alcohol content variable and divides the vector in half. The bottom 50% of alcohol content values should be labeled "Below Average", and the top 50% of alcohol content values should be labeled "Above Average".

   (e) Reorder the dataset so that the citric acid content is showing from high to low. Save this dataset to a dataframe and use it for all of the problems below.

   (f) Create a Box-plot in ggplot based on the wine quality and the residual sugar content

2. Use the `flights` dataset in the `nycflights13` library, write the code to answer the following questions. When you are working these steps out, only display the first 10 observations to show your results.

   (a) Display only the following columns: `year`, `month`, `day`, any column containing the pattern "arr", `carrier`, `dest`, and `distance`.

   (b) Rename the columns so that we have a column called `arrival`, `scheduled_arrival`, and `minutes_delayed`.

(c) Filter the dataset so only the "9E", "AA", "B6", "DL", "EV", "MQ", "UA", and "US" carriers are being shown.

(d) Omit all observations which contain a missing value

(e) Create a new column which is a logical values based on whether the flight was delayed by 1 minute or more.

(f) Create a barplot in ggplot which plots the carrier on the x-axis and the y-axis is the percentage of times the arrival time was considered delayed

(g) Saving the dataframe from part (e), use the aggregate function to determine which destination was delayed by the most minutes on average and then sort this from the most delayed to the least delayed (you should get "CAE" being delayed 41.76 minutes on average)

3. Use the `USArrests` dataset in R, write the code to answer the following questions. When you are working these steps out, only display the first 10 observations to show your results.

(a) Load the `USArrests` dataset and convert it to a dataframe called `arrests`. Add a column called `State` using the row names, then remove the row names so `State` is a regular column at the front of the other columns.

(b) Use `mutate()` to create two new columns:

i. `TotalRate` = `Murder` + `Assault` + `Rape`

ii. `ViolentClass` which labels a state as `"High"` if `TotalRate` is above the median `TotalRate`, and `"Low"` otherwise. Make sure this is saved as a factor

(c) Filter the dataset so that you only keep states where the `UrbanPop > 60` AND (the `Murder > 8` OR the `Rape > 20`). Save this filtered result as `arrests2`.

(d) Create a scatterplot using `ggplot2` which looks at UrbanPop and Murder rates based on the Violent Class category for the `arrests2` dataframe. Include a linear line of best fit based on the category.

(e) Use `pull()` to extract the `Murder` column as a vector from the `arrests2` dataframe, then compute the mean murder rate. Save the result as `mean_murder`.

(f) Using `mean_murder`, add another column to `arrests2` called `AboveMeanMurder` that is `TRUE` if `Murder` is above the mean, and `FALSE` otherwise.

(g) How many states have `AboveMeanMurder == TRUE` within the `arrests2` dataset? (determine this without using the summarize() or group_by() functions). You should get 9 total states.