IST387 – Week 3 / More Data Frames / Functions

Reminders of things to practice from last week:

Make a data frame	`data.frame()`
Row index of max/min	`which.max()  which.min()`
Sort value or order rows	`sort()  order()`
Descriptive statistics	`mean() sum() max()`

This Week: Often, when you get a dataset, it is not in the format you want. You can (and should) use code to refine the dataset to become more useful. As Chapter 6 of Introduction to Data Science mentions, this is called "data munging." In this homework, you will read in a dataset from the web and work on it (in a data frame) to improve its usefulness.

Step 1: Use *read_csv( )* to read a CSV file from the web into a data frame:

A. Use R code to read directly from a URL on the web. Store the dataset into a new dataframe, called *dfStates*. The URL is:

"https://raw.githubusercontent.com/CivilServiceUSA/us-states/master/data/states.csv"

Hint: Make sure to use *read_csv( )*, not *read.csv( )*. Check the help to compare them.

Step 2: Create a new data frame that only contains states with Twitter URLs:

B. Use *View( ), head( ), and tail ( )* to examine the *dfStates* data frame. **Add a block comment that briefly describes what you see.**

C. Create a selector variable that has TRUE if a state is missing its Twitter URL: *noTwitter <- is.na(dfStates$twitter_url)*

D. Use the *table( )* command to summarize the contents of *noTwitter*. **Write a comment interpreting what you see.**

E. Create a new data frame that contains only the states with Twitter URLs: *twitterStates <- dfStates[!noTwitter,]*

F. Use the *dim( )* command on *twitterStates* to confirm that the data frame contained 35 observations and 19 columns/variables.

Step 3: Calculate the mean for each of the three numeric variables.

G. The data frame contains three numeric variables. You can remind yourself of what they are by looking at the output of *str(twitterStates)*. Calculate the mean for each of the numeric variables.

H. Write a comment, noting the mean population for *twitterStates*.

Expert Mode!!! Create another data frame containing the 15 states that do not have Twitter URLs. Find out the mean population of those 15 states. Compare that to the answer you recorded for problem H.

Step 4: Extract the Twitter handle from the URL.

I. Use the *gsub()* command to remove the beginning part of the URL from the Twitter URLs. This command should work most of the time:
gsub("https://twitter.com/","", twitterStates$twitter_url)

J. Take a close look at the output from the *gsub( )* command in problem I. **Explain the cause of the incorrect results in a comment**.

K. Assign the results of the *gsub( )* command to a new variable on the data frame. Note that you do not have to repair the problems that you explained in problem J:
twitterStates$handle <- gsub("https://twitter.com/","", twitterStates$twitter_url)

Step 5: Create a function to extract Twitter handles:

L. The following function should work most of the time. Make sure to run this code before trying to test it. That is how you make the new function known to R. **Add comments to each line explaining what it does:**

```
getTwitterHandleFromURL <- function(URL) {
 fixTry1 <- gsub("https://twitter.com/","", URL)
 fixTry2 <- gsub("http://twitter.com/","", fixTry1)
 fixTry3 <- gsub("http://www.twitter.com/","", fixTry2)
 return(fixTry3)
}
```

M. Run your new function on the Twitter URLs. Make sure to use a comment to explain the cause of any problems that remain unfixed:
getTwitterHandleFromURL(twitterStates$twitter_url)

N. Assign the results of problem M to a variable on the data frame:
twitterStates$handle <- getTwitterHandleFromURL(twitterStates$twitter_url)

Expert Mode!!! Write a comment in your code that briefly describes an applied project where you could use the data frames and variables you just created.