HW03 - DS6306-402

Dan Crouthamel

May 21, 2019

```
knitr::opts_chunk$set(echo = TRUE)
library(ggplot2)
```

Question 1 - GitHub Cloning

Using Git, clone the following GitHub repository to your local machine: https://github.com/caesar0301/awesome-public-datasets. In RMarkdown, please show the code (commented out, as it's not R syntax) that you used to create a new directory, navigate to the appropriate directory, and clone the repository to it. One Git command per line, please.

```
# mkdir 'Unit 3'
# cd 'Unit 3'/
# git clone https://github.com/caesar0301/awesome-public-datasets
```

Question 2 - Data Summary

From this aforementioned cloned repo, please extract titanic.csv.zip. To be clear, this does not have to be done in Git or command line.

a. In R, please read in titanic.csv via either read.table() or read.csv(), assigning it to df. This dataset follows the passengers aboard the Titanic, including their fees paid, rooms rented, and survivorship status.

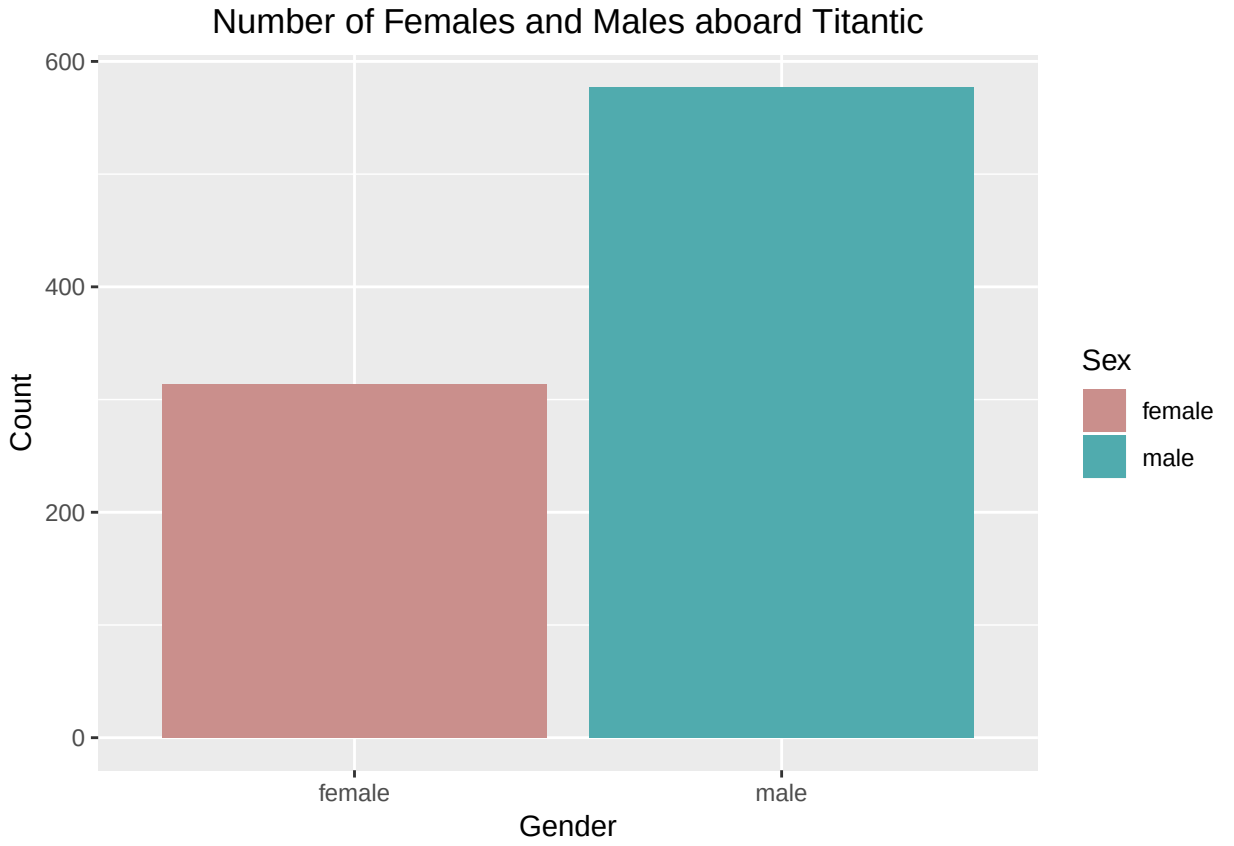
```
df <- read.csv("awesome-public-datasets/Datasets/titanic.csv")
```

b. Output the respective count of females and males aboard the Titanic. Plot the frequency of females and males. Be sure to give an accurate title and label the axes.

```
maleCount <- sum(df$Sex=='male')
femaleCount <- sum(df$Sex=='female')
# There are `r femaleCount` females and `r maleCount` aboard the Titantic.
```

There are 314 females and 577 aboard the Titantic.

```
ggplot(data=df, aes(x=Sex, fill=Sex)) +
  geom_bar() +
  labs(title = "Number of Females and Males aboard Titantic", x = "Gender", y = "Count") +
  scale_fill_hue(c = 40) +
  theme(plot.title = element_text(hjust = 0.5))
```



c. Please use one apply function (to review: swirl() modules 11, 12) to output the means of Age, Fare, and Survival. Make sure the output is a real number for all three means.

```
sapply(df[c("Age","Fare","Survived")], mean, na.rm=TRUE)
```

```
##        Age       Fare   Survived
## 29.6991176 32.2042080  0.3838384
```

Question 3 - Function Building

You research sleep and just got your first data set. Later, you'll have another dataset with the same column names, so you want to create a helper function that you can analyze this dataset and the next. Load sleep_data_01.csv (found at http://talklab.psy.gla.ac.uk/L1_labs/lab_1/homework/index.html). Questions 3A through 3D should be answered in function(x){}. 3E can be outside of the function.

a. Create objects for the median Age, the minimum and maximum Duration of sleep, and the mean and standard deviation of the Rosenberg Self Esteem scale (RSES). You may need to specify a few options like in Problem and live session.

b. Create a data.frame object called report: it should consist of the median age, the RSES mean and standard deviation respectively divided by five (since there are five questions and these scores are summed), and the range of Duration (the statistical definition of range; it should be a single number.)

c. Change the column names of this data.frame to MedianAge, SelfEsteem, SE_SD, and DurationRange.

d. Round the report to at most 2 digits: leave this as the closing line to the function.

e. Finally, run the function on your sleep data to show the output.

```
sleepData <- read.csv("sleep_data_01.csv")
sleepDataReport <- function(sleepData) {
  # Part A
  # DOS = Duration of Sleep
  # RSES = Rosenberg Self Esteem Scale
  medianAge <- median(sleepData$Age, na.rm = TRUE)
  minDOS <- min(sleepData$Duration, na.rm = TRUE)
  maxDOS <- max(sleepData$Duration, na.rm = TRUE)
  meanRSES <- mean(sleepData$RSES, na.rm = TRUE)
  sdRSES <- sd(sleepData$RSES, na.rm = TRUE)

  # Part B
  report <- data.frame(medianAge, (meanRSES / 5), (sdRSES / 5), (maxDOS - minDOS))

  # Part C
  names(report) <- c("MedianAge", "SelfEsteem", "SE_SD", "DurationRange")

  # Part D
  sapply(report, round, digits = 2)
}

# Part E
sleepDataReport(sleepData)
```

```
##     MedianAge    SelfEsteem         SE_SD DurationRange
##         14.00          3.62          1.24          7.00
```

Question 4 - FiveThirtyEight Data

Navigate on GitHub to https://github.com/rudeboybert/fivethirtyeight and read README.md. It will include everything you need.

a. Install the fivethirtyeight package.

```
# This was done using RStudio GUI interface. It can also be done via the command line
# using install.packages
library(fivethirtyeight)
```

b. In the listing of Data sets in package 'fivethirtyeight,' assign the 22nd data set to an object 'df.'

The ReadMe file on Github suggests using data(package = "fivethirtyeight") to see a list of all datasets.

```
the538sets <- data(package = "fivethirtyeight")
```

If you view the sets, you'll see it's a list of length 4 and the 3rd element labeled 'results' is a matrix. Then use the head command to take a peak at the structure of that matrix. The 'Item' column provides the dataset name that can be referenced when the libray is loaded. For this problem, we are interested in the 22nd entry.

```
names(the538sets)
```

```
## [1] "title"   "header"  "results" "footer"
```

```
class(the538sets$results)
```

```
## [1] "matrix"
```

```
head(the538sets$results)
```

```
##      Package           LibPath
## [1,] "fivethirtyeight" "/home/bsharp/R/x86_64-pc-linux-gnu-library/3.6"
## [2,] "fivethirtyeight" "/home/bsharp/R/x86_64-pc-linux-gnu-library/3.6"
## [3,] "fivethirtyeight" "/home/bsharp/R/x86_64-pc-linux-gnu-library/3.6"
## [4,] "fivethirtyeight" "/home/bsharp/R/x86_64-pc-linux-gnu-library/3.6"
## [5,] "fivethirtyeight" "/home/bsharp/R/x86_64-pc-linux-gnu-library/3.6"
## [6,] "fivethirtyeight" "/home/bsharp/R/x86_64-pc-linux-gnu-library/3.6"
##      Item
## [1,] "US_births_1994_2003"
## [2,] "US_births_2000_2014"
## [3,] "ahca_polls"
## [4,] "airline_safety"
## [5,] "antiquities_act"
## [6,] "avengers"
##      Title
## [1,] "Some People Are Too Superstitious To Have A Baby On Friday The 13th"
## [2,] "Some People Are Too Superstitious To Have A Baby On Friday The 13th"
## [3,] "American Health Care Act Polls"
## [4,] "Should Travelers Avoid Flying Airlines That Have Had Crashes in the Past?"
## [5,] "Trump Might Be The First President To Scrap A National Monument"
## [6,] "Joining The Avengers Is As Deadly As Jumping Off A Four-Story Building"
```

```
the538sets$results[22,"Item"]
```

```
##                   Item
## "college_recent_grads"
```

The 'college_recent_grads' is a dataset that can be loaded by name, or by the last command above.

```
df <- get(the538sets$results[22,"Item"])
# or this works too since we know the name, df <- get("college_recent_grads")
```

c. Use a more detailed list of the data sets to write out the URL in a comment to the related news story.

From the ReadMe file we can use the following (shown in comments). This bring up a help page and the URL for 'college_recent_grads' can be found.

```
# vignette("fivethirtyeight", package = "fivethirtyeight")
# http://fivethirtyeight.com/features/the-economic-guide-to-picking-a-college-major/
```

d. Using R command(s), give the dimensions and column names of this data frame.

```
dim(df)
```

```
## [1] 173  21
```

```
colnames(df)
```

```
##  [1] "rank"                        "major_code"
##  [3] "major"                       "major_category"
##  [5] "total"                       "sample_size"
##  [7] "men"                         "women"
##  [9] "sharewomen"                  "employed"
## [11] "employed_fulltime"           "employed_parttime"
## [13] "employed_fulltime_yearround" "unemployed"
## [15] "unemployment_rate"           "p25th"
## [17] "median"                      "p75th"
## [19] "college_jobs"                "non_college_jobs"
## [21] "low_wage_jobs"
```

Question 5 - Data Summary

Use your newly assigned data frame from question 4 for this question.

a. Write an R command that gives you the column names of the data frame. Right after that, write one that counts the number of columns but not rows. Hint: The number should match one of your numbers in Question 1d for dimensions.

```
colnames(df)
```

```
##  [1] "rank"                          "major_code"
##  [3] "major"                         "major_category"
##  [5] "total"                         "sample_size"
##  [7] "men"                           "women"
##  [9] "sharewomen"                    "employed"
## [11] "employed_fulltime"             "employed_parttime"
## [13] "employed_fulltime_yearround"   "unemployed"
## [15] "unemployment_rate"             "p25th"
## [17] "median"                        "p75th"
## [19] "college_jobs"                  "non_college_jobs"
## [21] "low_wage_jobs"
```

```
ncol(df)
```

```
## [1] 21
```

b. Generate a count of each unique major_category in the data frame. I recommend using libraries to help. To be clear, this should look like a matrix or data frame containing the major_category and the frequency it occurs in the dataset. Assign it to major_count.

```
library(dplyr)
major_count <- tally(group_by(df,major_category))
major_count
```

```
## # A tibble: 16 x 2
##    major_category                            n
##    <chr>                                 <int>
##  1 Agriculture & Natural Resources          10
##  2 Arts                                      8
##  3 Biology & Life Science                   14
##  4 Business                                 13
##  5 Communications & Journalism               4
##  6 Computers & Mathematics                  11
##  7 Education                                16
##  8 Engineering                              29
##  9 Health                                   12
## 10 Humanities & Liberal Arts                15
## 11 Industrial Arts & Consumer Services       7
## 12 Interdisciplinary                         1
## 13 Law & Public Policy                       5
## 14 Physical Sciences                        10
## 15 Psychology & Social Work                  9
## 16 Social Science                            9
```

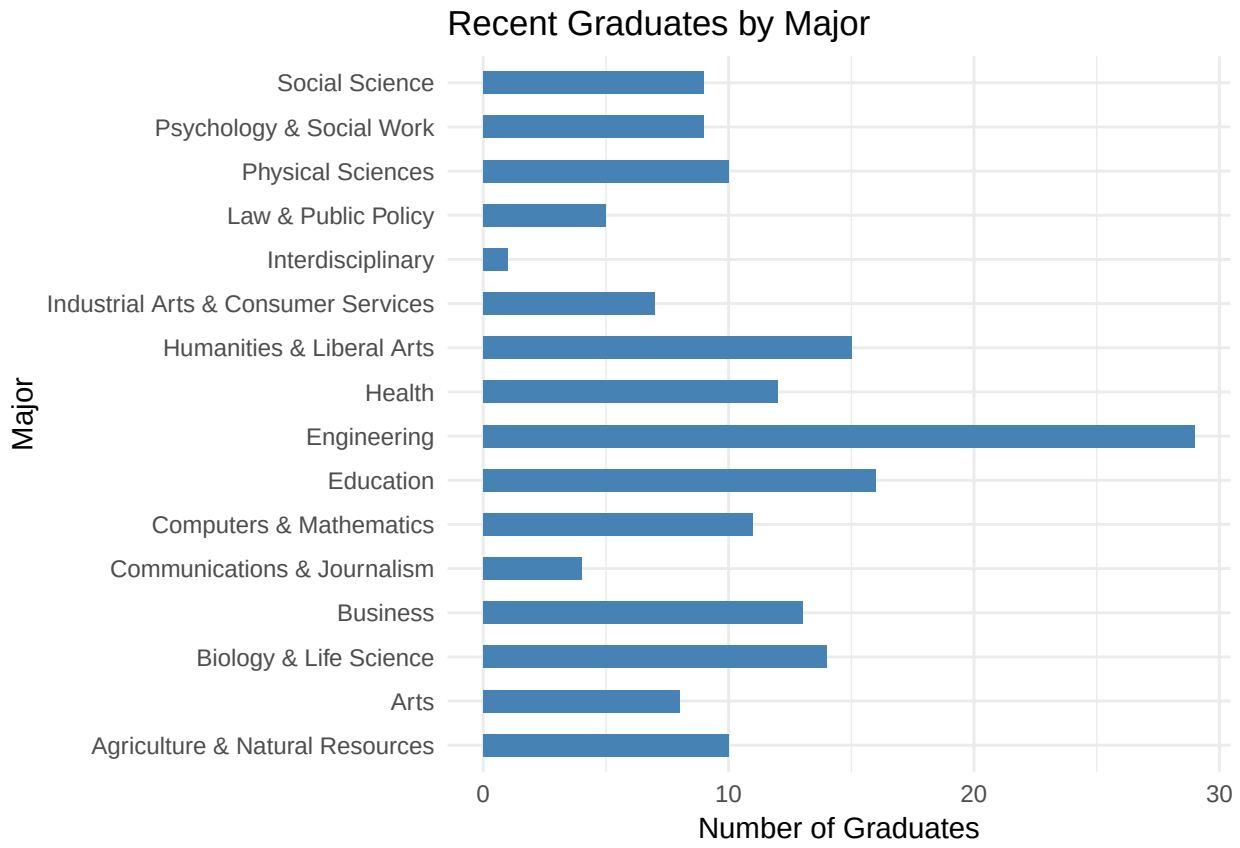
```
# Could have also done df %>% group_by(major_category) %>% tally()
```

Above I decided to go with the dplyr package and the tally command.

c. To make things easier to read, put par(las=2) before your plot to make the text perpendicular to the axis. Make a barplot of major_count. Make sure to label the title with something informative (check the vignette if you need), label the x and y axis, and make it any color other than grey. Assign the major_category labels to their respective bar. Flip the barplot horizontally so that bars extend to the right, not upward. All of these options can be done in a single pass of barplot(). Note: It's okay if it's wider than the preview pane.

*Please note, I used ggplot instead. Hope that is OK!

```
ggplot(data=major_count, aes(x=major_count$major_category, y=major_count$n)) +
  geom_bar(stat="identity", width=0.5, fill="steelblue") + coord_flip() +
  labs(title = "Recent Graduates by Major", x = "Major", y = "Number of Graduates") +
  theme(plot.title = element_text(hjust = 0.5)) +
  theme_minimal()
```



d. Write the fivethirtyeight data to a csv file. Make sure that it does not have row labels.

```
write.csv(df, file = "recent_college_grads.csv", row.names = FALSE)
```

Question 6 - Data Summary

a. Start a new repository on GitHub for your SMU MSDS homework. On your local device, make sure there is a directory for Homework at the minimum; you are welcome to add whatever you would like to this repo in addition to your requirements here.

b. Create a README.md file which explains the purpose of the repository, the topics included, the sources for the material you post, and contact information in case of questions. Remember, the one in the root directory should be general. You are welcome to make short READMEs for each assignment

individually in other folders.

c. In one (or more) of the nested directories, post your RMarkdown script, HTML file, and data from 'fivethirtyeight.' Make sure that in your README or elsewhere that you credit fivethirtyeight in some way.

d. In your RMarkdown script, please provide the link to this GitHub so the grader can see it.

```
## https://github.com/bSharpCyclist/MSDS-6306---Intro-To-Data-Science.git
```