HW1

Lina Lee

8/29/2020

Problem 1

R is an open source, community built, programming platform. Not only is there a plethra of useful web based resources, there also exist in-R tutorials. Please do both of the Primers labeled as The Basics on Rstudio.cloud.

I completed the Primers.

Problem 2

Part A

In this new Rmarkdown file, please type a paragraph about what you are hoping to get out of this class. Include at least 3 specific desired learning objectives in list format

- To understand how to use R to do proper analysis for given data as well as generate plots
- To learn how to display plots based on data more effectively to readers
- To practice to make a well written report using latex

Part B

To this, add 3 density functions (Appendix Cassella & Berger) in centered format with equation number, i.e. format this as you would find in a journal.

$$f(x|\alpha,\beta) = \frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1}, \quad 0 \leq x \leq 1, \quad \alpha \geq 0, \quad \beta \geq 0 \tag{1}$$

$$f(x|\beta) = \frac{1}{\beta}exp^{-\frac{x}{\beta}}, \quad 0 \leq x < \infty, \quad \beta \geq 0 \tag{2}$$

$$f(x|\alpha,\beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}}x^{\alpha-1}exp^{-\frac{x}{\beta}}, \quad 0 \leq x < \infty, \quad \alpha \geq 0, \quad \beta \geq 0 \tag{3}$$

Problem 3

1. Rule 1: For Every Result, Keep Track of How It Was Produced

We should record the details of steps that we did to get the final result from raw data including pre-processing and post-processing steps as well as the program we used. Based on my previous research experience, keeping a record of all the details for each step is forgettable. This is because research usually has been done over a long period. Therefore organization of research results and data is very important.

2. Rule 2: Avoid Manual Data Manipulation Steps
   This analysis does not include any manual manipulation of the given data. Manual manipulations will lead to error as well as reproduction risk. some data is very complicated and unorganized, it is very hard to clean the data without manual manipulation sometimes. In this case, we may use a minimum of manual manipulation.

3. Rule 3: Archive the Exact Versions of All External Programs Used
   We should record names and versions of the program we used for analysis because input and output format changes between versions or some programs have specific requirements to installed program and package. we may not notice an update of the program version, so we may miss recording updated version of program. we need to check the version update sometimes.

4. Rule 4: Version Control All Custom Scripts
   Since small changes in the program brings a huge difference in consequence. We should record exactly the same code script to reproduce the result. Therefore, tracking changes in the program is really important. we can track changes in code by using a version control system like Subversion, Git, or Mercuria. If I use a version control system, there is no many difficulties that I will have in this step.

5. Rule 5: Record All Intermediate Results, When Possible in Standardized Formats
   If we record all intermediate results, we can easily find and modify faults in each step. Also, it allows us to find the causes of discrepancy in reproductions. If research includes many intermediate steps, saving all the intermediate results will require huge storage. then, we need to divide the results into several groups and save each group separately or we can use a cloud that has a big storage size.

6. Rule 6: For Analyses That Include Randomness, Note Underlying Random Seeds
   When analyses include randomness, results may be slightly different by each run. Random seed should be recorded to exact reproduction of results in the future. I don't have many challenges in this step since I can easily record random seed.

7. Rule 7: Always Store Raw Data behind Plots
   If we store raw data underlying plots, we can easily modify plots without doing the whole analysis again. Also, underlying data as well as storing values both need to be saved. In this step, I don't have any challenges.

8. Rule 8: Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected
   we can incorporate debug outputs in the source code of scripts and programs. we can include the link of the permanent output of all underlying data in the main result, so we can easily see the output of all underlying data if we need it.

9. Rule 9: Connect Textual Statements to Underlying Results
   We need to connect a given textual interpretation to the underlying results when the textual interpretation is initially made. so we can avoid errors made by connecting the statements with the exact underlying results later. If we used several data to do the analysis at a stage of research, we may miss one file. we should check whether we include all the data files information in text documents.

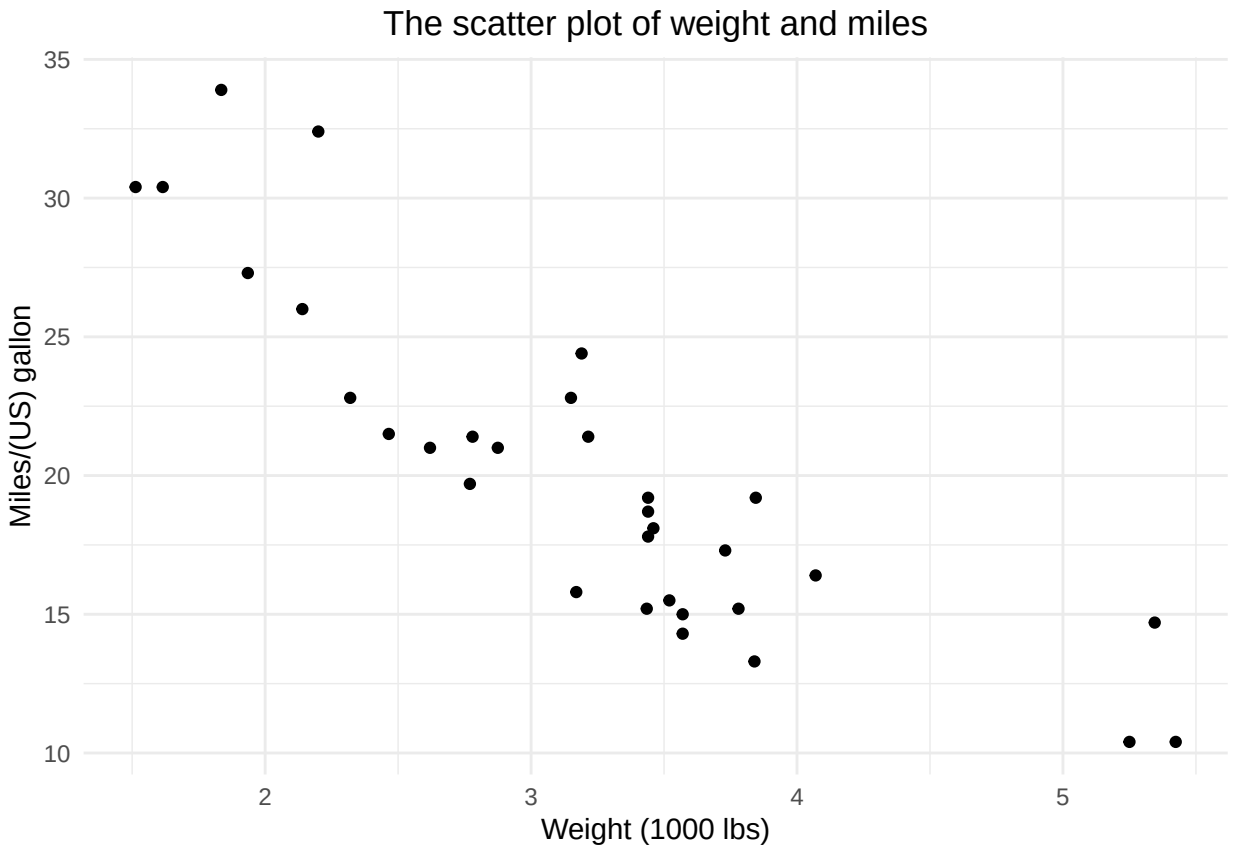
10. Rule 10: Provide Public Access to Scripts, Runs, and Results
    We should provide main data, source code, program version, and intermediate results to the public. We should also provide more methodology details if anyone requests them. In this step, there is no many challenges that I can have.

Problem 4

Please create and include a basic scatter plot and histogram of an internal R dataset.

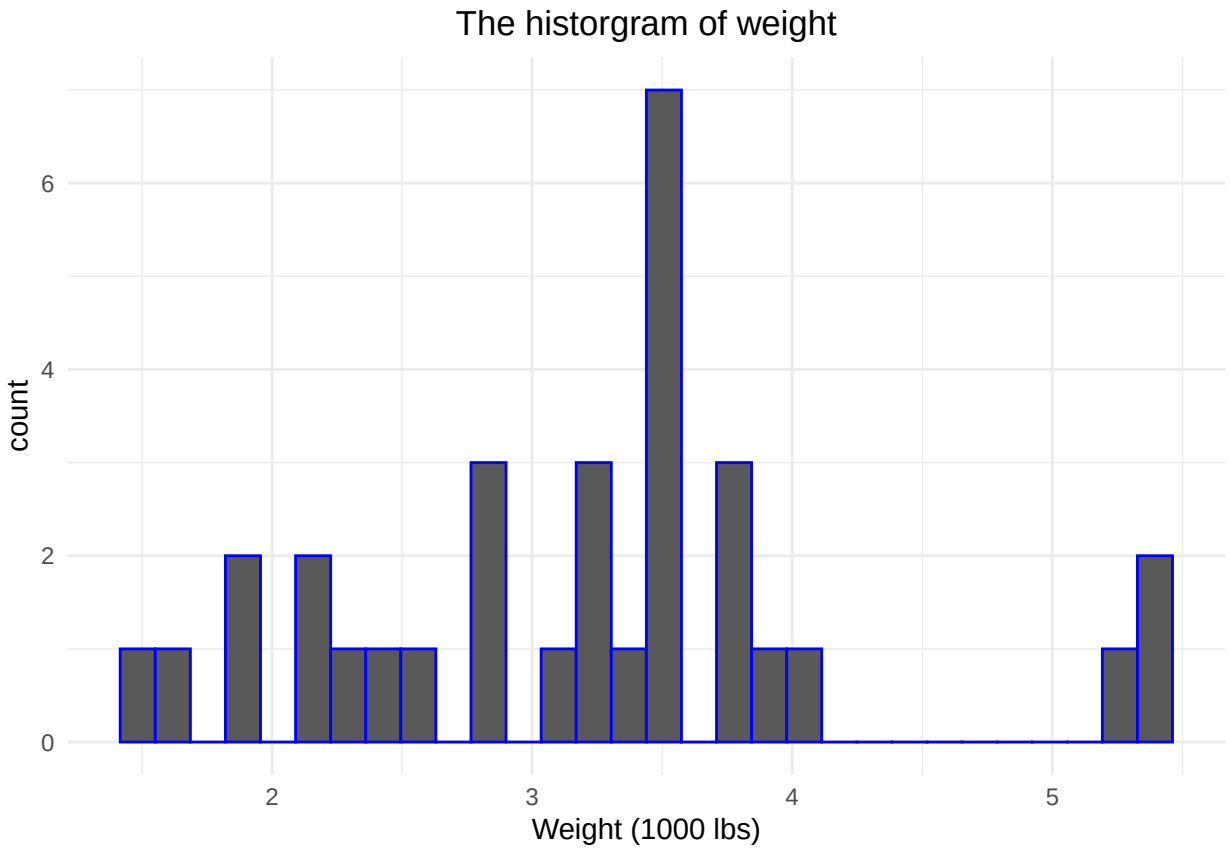
The data "mtcars" was used for creating plots. The data was generated from the 1974 Motor Trend US magazine.The data include fuel consumption and 10 automobiles characteristics of 32 automobiles (1973–74 models).

```
library(ggplot2)
# library(help="datasets")

# ?mtcars


ggplot()+
  geom_point(aes(x=wt,y=mpg),data=mtcars)+
  labs(x="Weight (1000 lbs)",y="Miles/(US) gallon",title="The scatter plot of weight and miles")+
  theme_minimal()+
    theme(plot.title = element_text(hjust = 0.5))
```



There is negative linear trend between and weight and miles that car travel per gallon of feul. As weight increases, miles per gallon decreases. That is, the heavier car are able to travel less miles per gallon than the lighter cars.

```r
ggplot()+
  geom_histogram(aes(x=wt),col="blue",data=mtcars)+
  labs(x="Weight (1000 lbs)",title="The historgram of weight")+
  theme_minimal()+
  theme(plot.title = element_text(hjust = 0.5))
```



The histogram shows the frequency by weight of 32 observations. 3500lbs has the highest frequency. Weight of most cars are below 4000lbs except 3 cars with higher than 5000lbs.