Homework 1

Ankur Patel

8/29/2020

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

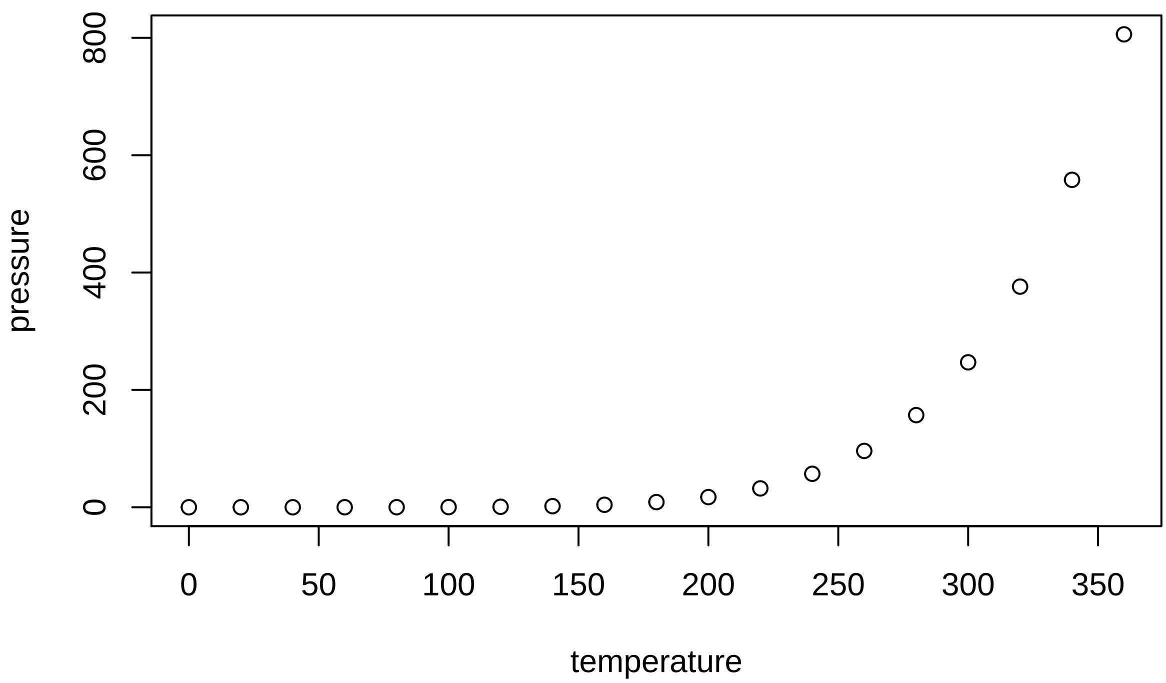
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00  
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Problem 2

Part A

I am entering this class with no knowledge of how to use GitHub and my knowledge of the R terminal commands is also very limited. Although I have experience coding with R and making plots with R, I want to learn how to make my code and plots more presentable. I am hoping this class will be useful not only for my academic development but also for my future professional career, especially when it comes to using version control. My learning objectives are

- To learn how to use GitHub for my academic and professional development
- To learn how to format and professionally present data in scientific papers
- To get a better understanding of using terminal commands

Part B

$$f(x) = \frac{1}{\pi\sigma}\frac{1}{1+(\frac{x-\theta}{\sigma})^2} \text{ for all } x, \theta \text{ and } \sigma > 0 \text{ (Cauchy pdf)} \tag{1}$$

$$f(x) = \frac{1}{\beta}e^{-\frac{x}{\beta}} \text{ for } x > 0, \beta > 0 \text{ (Exponential pdf)} \tag{2}$$

$$f(x) = \frac{1}{B(\alpha,\beta)}x^{\alpha-1}(1-x)^{\beta-1} \text{ for } \alpha, \beta > 0 \text{ and } 0 < x < 1 \text{ (Beta pdf)} \tag{3}$$

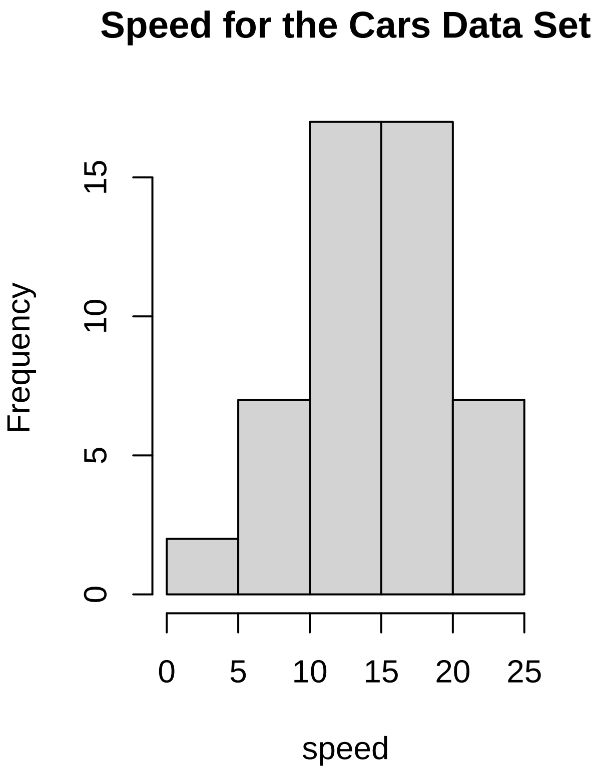
Problem 3

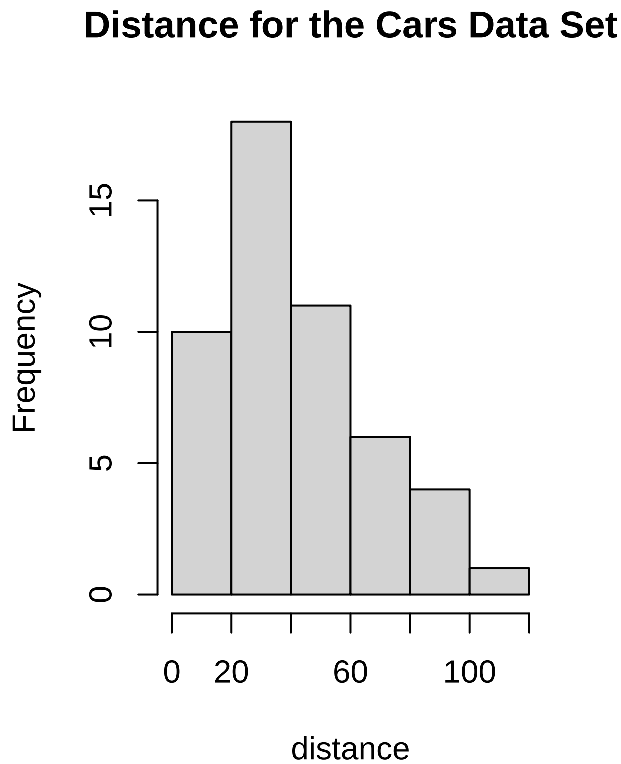
1. Keep track of how every result was produced.
   - Challenge: It is difficult to know which results will be of most importance
2. Try not to manually manipulate data
   - Challenge: Some datasets by their nature may require manual data manipulation
3. Keep track of and store the exact versions of the external programs used
   - Challenge: Versions of the programs used may change very frequently which will make it time consuming to archive the versions
4. Version control all custom scripts
   - Challenge: This may prove difficult if multiple people are working on the same script, and there needs to be close coordination and verification among team members
5. Use standardized formats to record intermediate results
   - Challenge: Storage space for intermediate records may not be available when very large output files are generated.
6. Record the random seeds used
7. Store the raw data behind plots
   - Challenge: The raw data might be messy or cumbersome to handle. It will also take up storage space that may be very limited
8. Store the detailed values making up a summary in such a way that layers of information can be inspected
   - Challenge: There may be many layers of information and storing the files could require storage space that isn't available
9. Connect statements and explanations in the paper to results

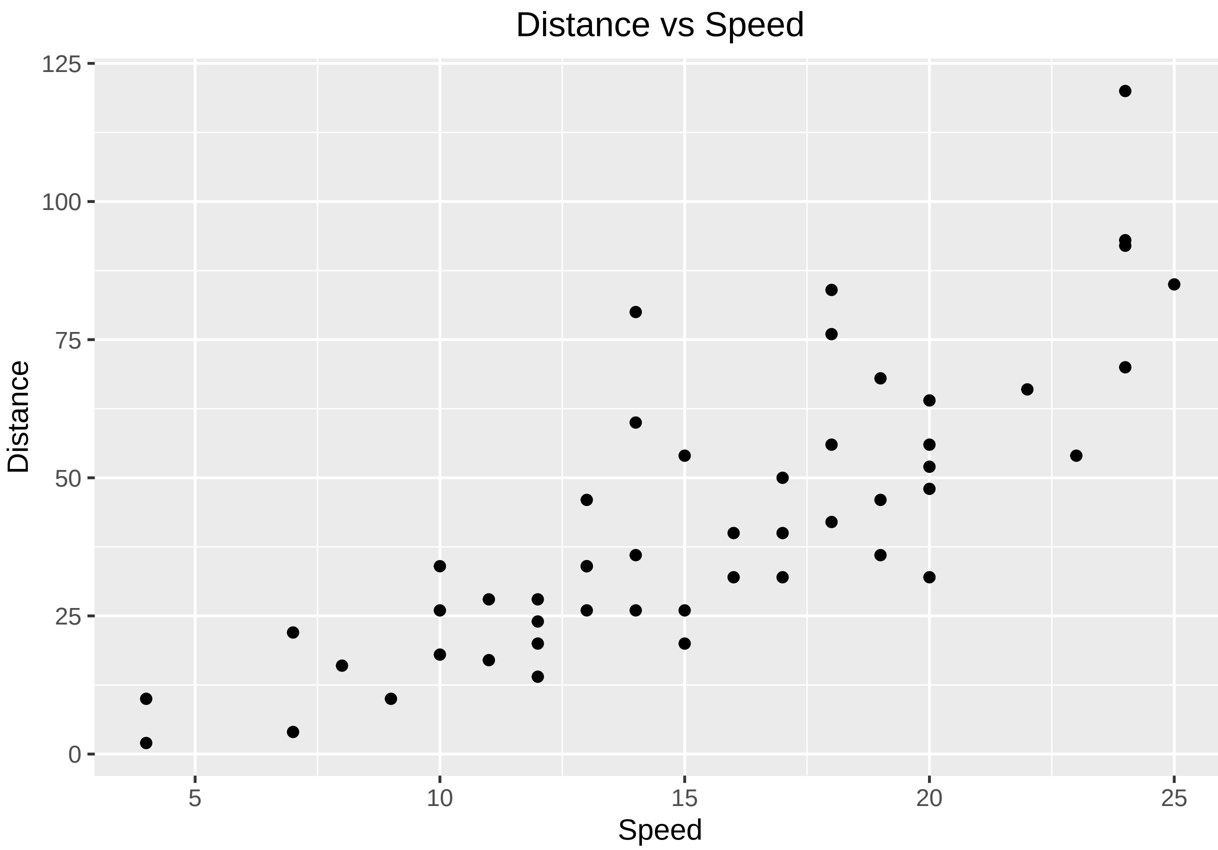
- Challenge: A given statement may be connected to many results across different versions which could overload the reader with having to look at different versions

10. Give the public access to scripts, runs, and results

Problem 4






Distance vs Speed
Distance
Speed
125
100
75
50
25
0
5
10
15
20
25