STAT 385 Fall 2019 Homework Assignment 01

Tianli Ding

Due by 12:00 PM 09/14/2019

Grader Comment

Efficiency:

The code and document is written with lots of logical thought about the tasks with few unnecessary or unnecessarily verbose lines and is written in a least brute force way.

Correctness:

The files run with no errors and the coding is written in a most accurate way.

Documentation:

The explanations and comments contain reasonable clarity and exhibit mid-tier (to be expected) fundamentals of usefulness, grammar, smoothness, and structure.

Beauty:

The document is more visually appealing and has decent readability. The coding output and visualizations are averagely attractive.

NOTE:

Please note that the summary above applies to the completed work. If the submission was incomplete, the ultimate mark for the submission will be scaled down proportionally.

Please take care to ensure that one's repo includes the image files that are included in one's .rmd file. Otherwise, they may not appear in the graded work that is pushed to the student's git.

Please see comments below for specifics.

Below you will find problems for you to complete as an individual. It is fine to discuss the homework problems with classmates, but cheating is prohibited and will be harshly penalized if detected.

1. With the `data(pressure)` dataset in R, do the following:

a. run the command that shows the help page of the median function

```
help(median)
```

b. show the first 10 rows and 2 columns of the `data(pressure)` dataset

```
head(pressure, 10)
```

```
##    temperature pressure
## 1            0   0.0002
## 2           20   0.0012
## 3           40   0.0060
## 4           60   0.0300
## 5           80   0.0900
## 6          100   0.2700
## 7          120   0.7500
## 8          140   1.8500
## 9          160   4.2000
## 10         180   8.8000
```

c. run the command `head(pressure)`

```
head(pressure)
```

```
##   temperature pressure
## 1           0   0.0002
## 2          20   0.0012
## 3          40   0.0060
## 4          60   0.0300
## 5          80   0.0900
## 6         100   0.2700
```

d. describe the differences between parts **b** and **c**.

In part **b**, we manually set the rows that displayed to 10. While in part **c**, it shows only 6 lines by default.

2. With the `data(cars)` dataset in R, compute the following:

a. the median of the stopping distance

```
median(cars$dist)
```

```
## [1] 36
```

b. the mean of the speed

```
mean(cars$speed)
```

```
## [1] 15.4
```

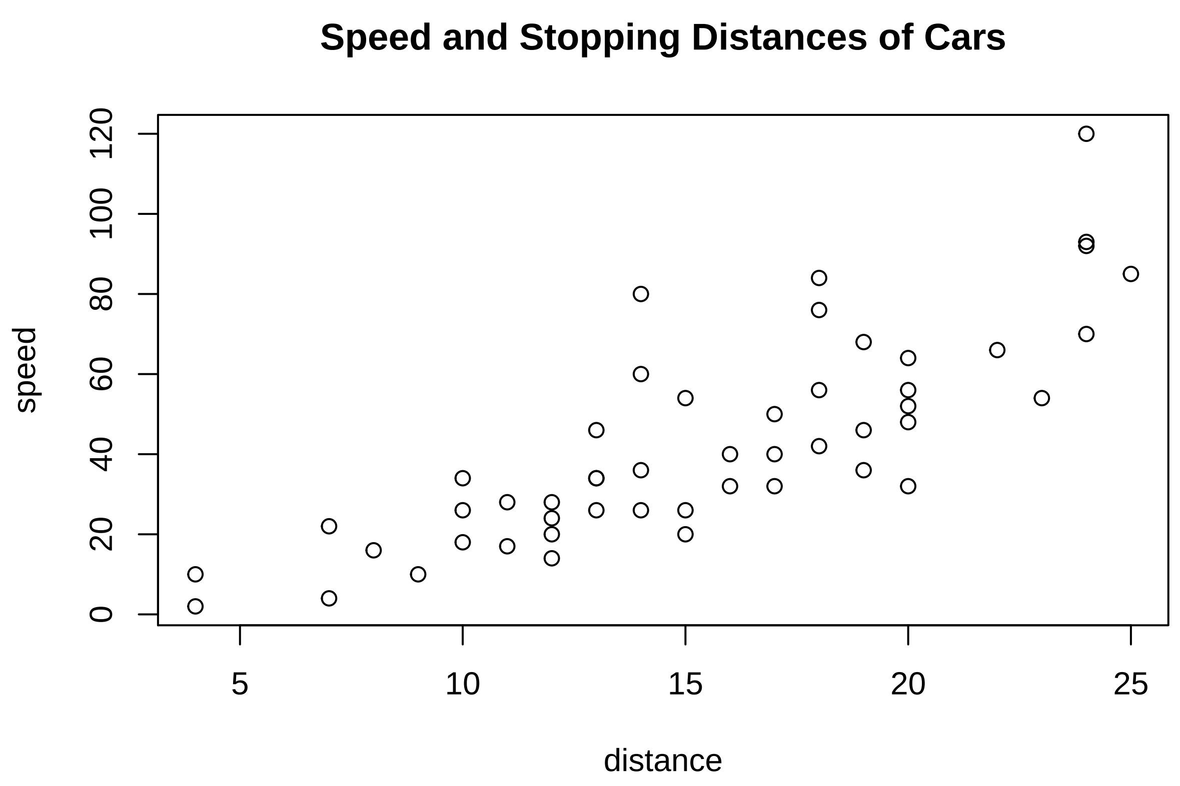
c. the five number summary of both the mean and speed

```r
#the five number summary of both the stopping distance and speed:
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
```

d. the plot of the stopping distance (vertical axis) versus the speed (horizontal axis) making sure the axes are written exactly as “distance” and “speed”, while the title as “Speed and Stopping Distances of Cars”.

```r
plot(cars$speed, cars$dist, xlab = "distance", ylab = "speed", main = "Speed and Stopping Distances of C
```



3. With the `data(iris)` dataset in R, do the following calculations:

a. the sum of the 25th and 26th versicolor iris’s sepal lengths divided by 2

```r
which(iris$Species == 'versicolor')[25]
```

```
## [1] 75
```

```r
which(iris$Species == 'versicolor')[26]
```

```
## [1] 76
```

```r
(iris$Sepal.Length[75] + iris$Sepal.Length[76])/2
```

```
## [1] 6.5
```

b. the sum of all petal lengths among all species of iris that are between 0 cm and 3 cm

```r
sum = 0
for (i in 1:length(iris$Petal.Length)) {
  if(iris$Petal.Length[i] <= 3)
    sum = sum + iris$Petal.Length[i]
  i = i+1
}
sum
```

```
## [1] 76.1
```

c. assuming x equals the 10th virginica iris, then $4 \cdot x/2^3$

```r
x = iris[which(iris$Species == 'virginica'),]
x = data.frame(iris[110,1:4])
x/(2^3)*4
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width
## 110          3.6         1.8         3.05        1.25
```

d. assuming x equals the sum of all setosa sepal widths, then $4 \cdot (x/2)^3$.

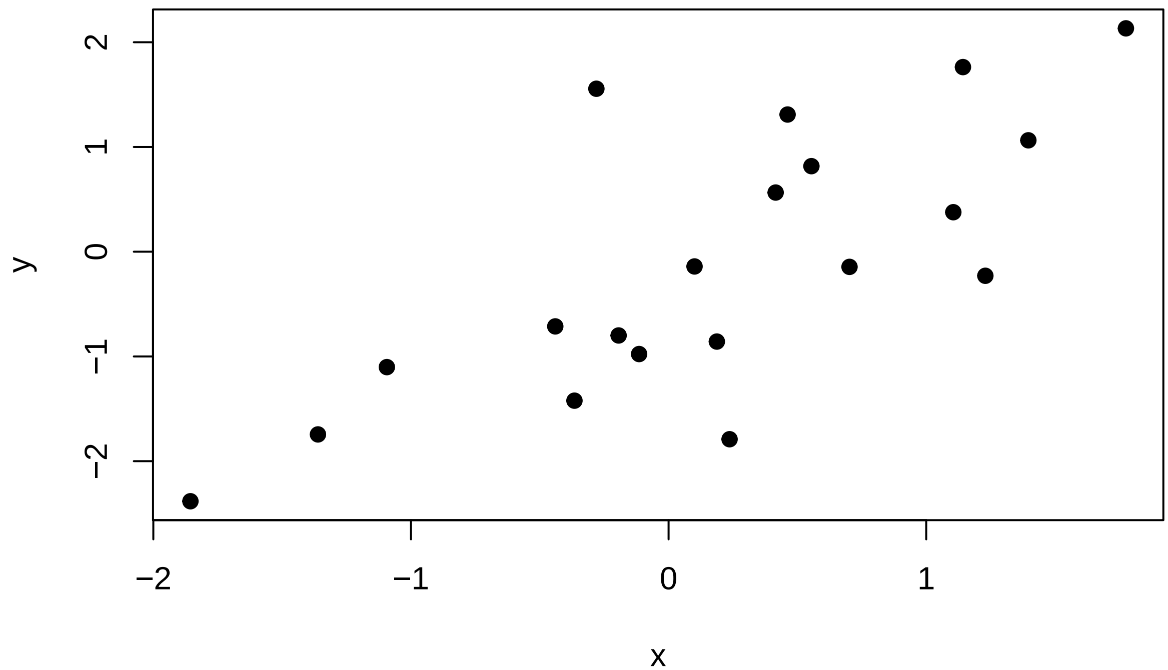
```r
x2 = iris[which(iris$Species == "setosa"),]
x2 = sum(x2$Sepal.Width)
x2/(2^3)*4
```

```
## [1] 85.7
```

4. Using the Instructor Notes from Week 03, address the following:

a. run the code that appears in the "Try This at Home" section at the very bottom

```r
set.seed(13) ## set the random seed
x <- rnorm(n = 20) ## generate predictor variable
y <- rnorm(n = 20, mean = x) ## generate response variable
plot(y ~ x, pch = 19) ## plot (x,y) pairs
```

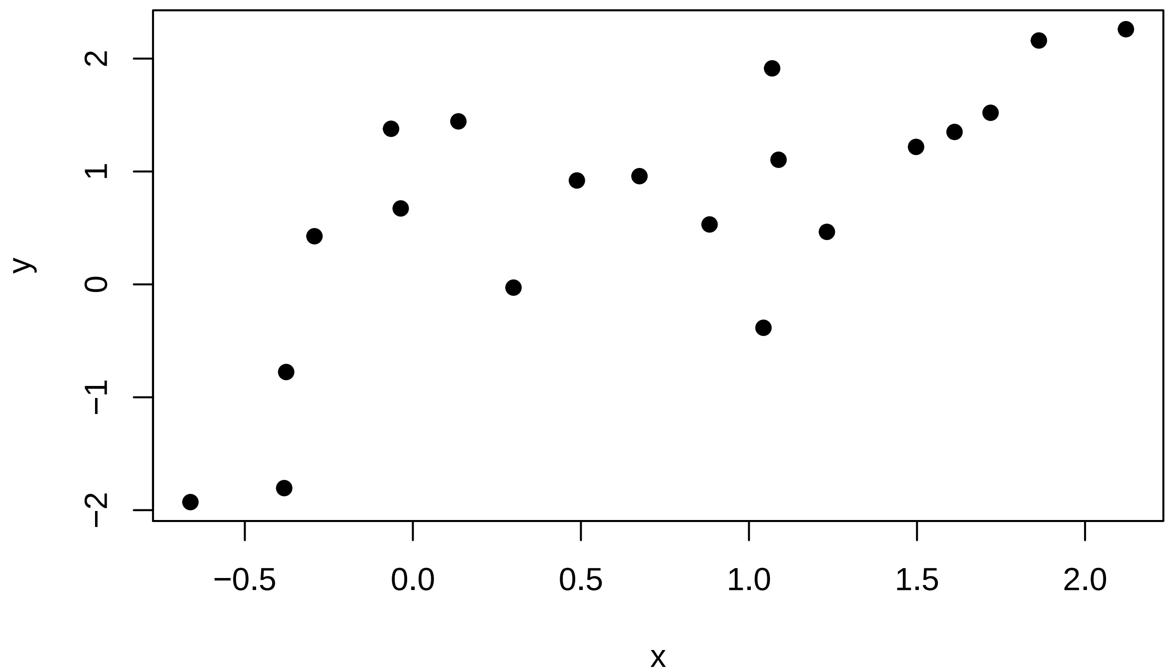



```r
dat <- data.frame(y = y, x = x)
model <- lm(y ~ x, data = dat)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71229 -0.54021 -0.03768  0.48680  2.16449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3210     0.2021  -1.588 0.129697
## x             1.0263     0.2168   4.733 0.000166 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8869 on 18 degrees of freedom
## Multiple R-squared:  0.5544, Adjusted R-squared:  0.5297
## F-statistic:  22.4 on 1 and 18 DF,  p-value: 0.0001661
```

b. change the seed number in the set.seed() command from 13 to 14. Then, re-run the code in part **a**.

```r
set.seed(14) ## set the random seed
x <- rnorm(n = 20) ## generate predictor variable
y <- rnorm(n = 20, mean = x) ## generate response variable
plot(y ~ x, pch = 19) ## plot (x,y) pairs
```




```r
dat <- data.frame(y = y, x = x)
model <- lm(y ~ x, data = dat)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ x, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40606 -0.33953 -0.05979  0.52395  1.46208
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.01928    0.24659  -0.078 0.938533
## x            0.99125    0.22947   4.320 0.000413 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.8407 on 18 degrees of freedom
## Multiple R-squared:  0.509,  Adjusted R-squared:  0.4817
## F-statistic: 18.66 on 1 and 18 DF,  p-value: 0.0004126
```

c. Is there a difference between the results of parts **a** and **b**? If so, describe some of the obvious differences.

Yes, the results from part a and b are different. One of the obvious difference is that different seeds gives the different random value.

d. Why is it important to set the random seed at the beginning of a simulation?

Since we want to make sure that everytime we are running the simulation, we get the same result by setting a fixed seed. If we do not set a seed, then it will generate different result which is hard for us to check the answer.

5. Find images that address the following:

a. a screenshot that shows your personal repo in GitHub Enterprise

b. a screenshot of the Git terminal (or shell) open within RStudio

c. a screenshot of RStudio with the 4 panes basic such that there's a Git tab in the top right pane, and the files in the repo in the bottom right pane

d. the push/pull pop up window of RStudio with the final commit message of your homework submission.