94/100
-2 For not submitting PDF

Calderoni-HW1

Ivan Calderoni

8/31/2017

```
# Packages used:

library(moments)
library(plyr)
```

Problem 1: Using R: Vectors

Problem 1 (a):

```
# Create a vector with 10 numbers (3, 12, 6, -5, 0, 8, 15, 1, -10,
# 7) and assign it to x.

x <- c(3, 12, 6, -5, 0, 8, 15, 1, -10, 7)

# print x
x

##  [1]   3  12   6  -5   0   8  15   1 -10   7
```

Problem 1 (b):

```
# Using the seq command, create a new vector y with 10 elements
# ranging from the minimum value of x to the maximum value of x.

y <- seq(min(x), max(x), length.out = 10)

# print y
y

##  [1] -10.000000  -7.222222  -4.444444  -1.666667   1.111111   3.888889
##  [7]   6.666667   9.444444  12.222222  15.000000
```

Problem 1 (c):

```
# Compute the sum, mean, standard deviation, variance, mean
# absolute deviation for x and y.

sum(x)
```

```
## [1] 37
sum(y)
## [1] 25
mean(x)
## [1] 3.7
mean(y)
## [1] 2.5
sd(x)
## [1] 7.572611
sd(y)
## [1] 8.41014
var(x)
## [1] 57.34444
var(y)
## [1] 70.73045
mad(x)
## [1] 5.9304
mad(y)
## [1] 10.29583
```

Problem 1 (d):

```
# Find a package (or packages) that provide the  statistical
# measures skewness and kurtosis. Use the appropriate functions
# from the package to calculate the skewness and kurtosis of x.

skewness(x)
## [1] -0.3123905
kurtosis(x)
## [1] 2.355328
```

Problem 1 (e):

```
# Use t.test() to compute a statistical test for differences in
# means between the vectors x and y. Are the differences in means
# significant?

t.test(x, y)

##
##  Welch Two Sample t-test
##
## data:  x and y
## t = 0.33531, df = 17.805, p-value = 0.7413
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -6.324578  8.724578
## sample estimates:
## mean of x mean of y
##       3.7       2.5

# Since the p-value = 0.7413 is greater than the significance level
# (alpha = 0.05), we conclude that differences in means are not
# significant.
```

Vigneshwaran Dharmarajan: good

Problem 1 (f):

```
# Sort the vector x and re-run the t-test as a paired t-test.

x <- sort(x)

# print x
x

##  [1] -10  -5   0   1   3   6   7   8  12  15

t.test(x, y, paired=TRUE)

##
##  Paired t-test
##
## data:  x and y
## t = 2.164, df = 9, p-value = 0.05868
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.05440584  2.45440584
## sample estimates:
## mean of the differences
##                     1.2
```

```
# Since the p-value = 0.05868 is greater than the significance
# level (alpha = 0.05), we conclude that differences in means are
# not significant.
```

Problem 1 (g):

```
# Create a logical vector that identifies which numbers in x are
# negative.

a <- (x < 0)

# print a
a

##  [1]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Problem 1 (h):

```
# Use this logical vector to remove all entries with negative
# numbers from x. (Make sure to overwrite the vector x so that the
# new vector x has 8 elements!)

x <- x[a==FALSE]

# print x
x

## [1]  0  1  3  6  7  8 12 15
```

Problem 2: Using R: Introductory Data Exploration

Problem 2 (a):

```
# Use the read.csv() function to read the data into a data frame in
# R. Call the dataframe college. Make sure that you have the
# directory set to the correct location for the data (or that the
# data is in the same directory as the RStudio project).

college <- read.csv("college.csv")
```

Problem 2 (b):

```
# now R has given each row a name correponding to the university
rownames(college) <- college [,1]

 # deletes the first data column of college
college <- college [,-1]
```

Problem 2 (c) (i):

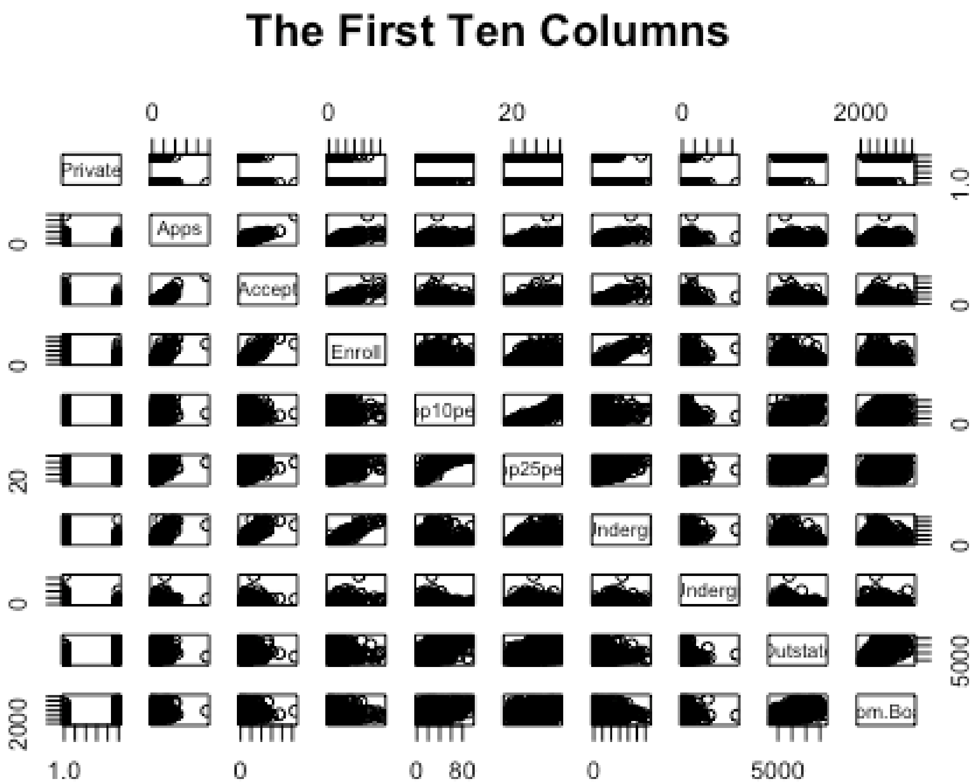
```r
# Use the summary() function to produce a numerical summary of the
# variables in the data set.

summary(college)

##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##             Median : 1558   Median : 1110   Median : 434   Median :23.00
##             Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
##             3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
##             Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
##    Top25perc      F.Undergrad     P.Undergrad         Outstate
##  Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
##  Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
##  Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
##  3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
##  Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
##    Room.Board       Books           Personal         PhD
##  Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
##  Median :4200   Median : 500.0   Median :1200   Median : 75.00
##  Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
##  3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
##  Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
##     Terminal       S.F.Ratio       perc.alumni        Expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
##  1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
##  Median : 82.0   Median :13.60   Median :21.00   Median : 8377
##  Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
##  3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
##  Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
##    Grad.Rate
##  Min.   : 10.00
##  1st Qu.: 53.00
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

Problem 2 (c) (ii):

```r
# Access help for the pairs function and then use pairs to produce
# a scatterplot matrix of the first ten columns. Recall that you
# can reference the first ten columns of a matrix A using A[,1:10].

# ?pairs
```

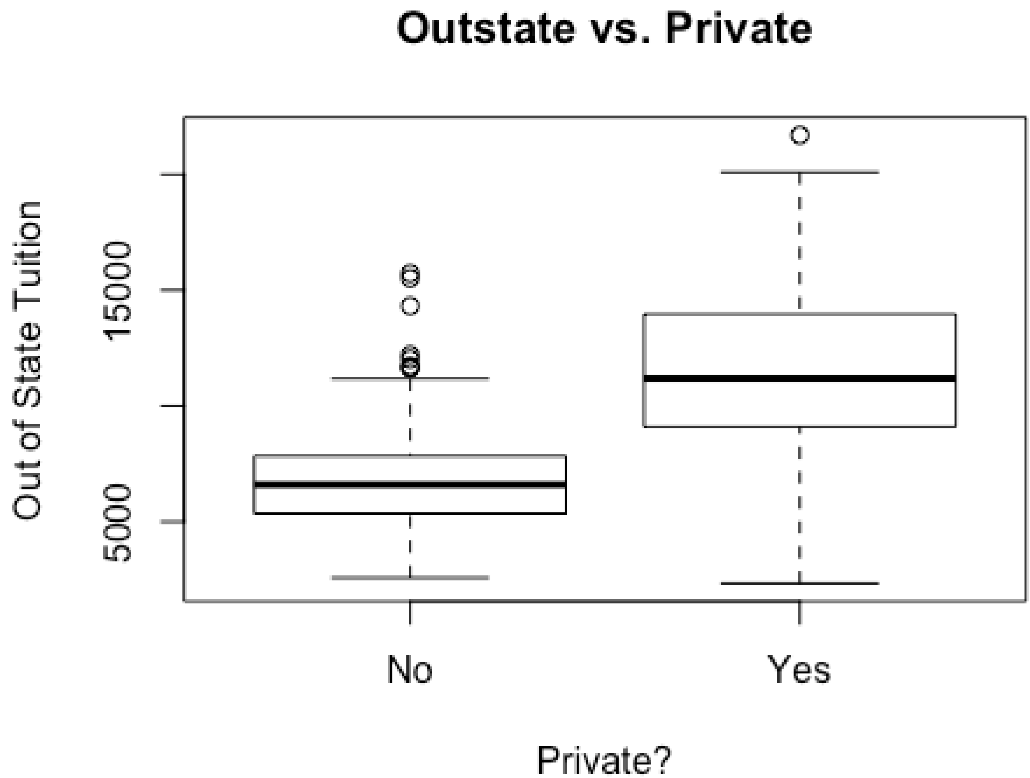
```r
pairs(college[,1:10], labels = colnames(college), main = "The First Ten Columns")
```

Vigneshwaran Dharmarajan: good



Problem 2c (iii):

```r
# Use the plot() function to produce side-by-side boxplots of
# Outstate versus Private. Label the axes and main title
# appropriately.

plot(college$Outstate ~ college$Private, main  ="Outstate vs. Private", xlab
= "Private?", ylab = "Out of State Tuition")
```



Problem 2 (c) (iv):

```
# Using the following bit of code you will create a new qualitative
# variable, called Elite by binning the Top10perc variable. That
# is, Elite will classify the universities into two groups based on
# whether or not the proportion of students coming from the top 10%
# of their high school classes exceeds 50%. Add comments to each
# line below explaining what the corresponding code is doing and
# then run the code.

# creates a new row in college and replicate "No" 777 times under
# the new row "Elite"
Elite <- rep ("No", nrow(college))

# changes the value from "No" to "Yes" if the college has a higher
# Top10per value than 50
Elite[college$Top10perc > 50] <- "Yes"

# now Elite has 2 levels "no", "yes" categorized as 1 and 2
Elite <- as.factor(Elite)
```

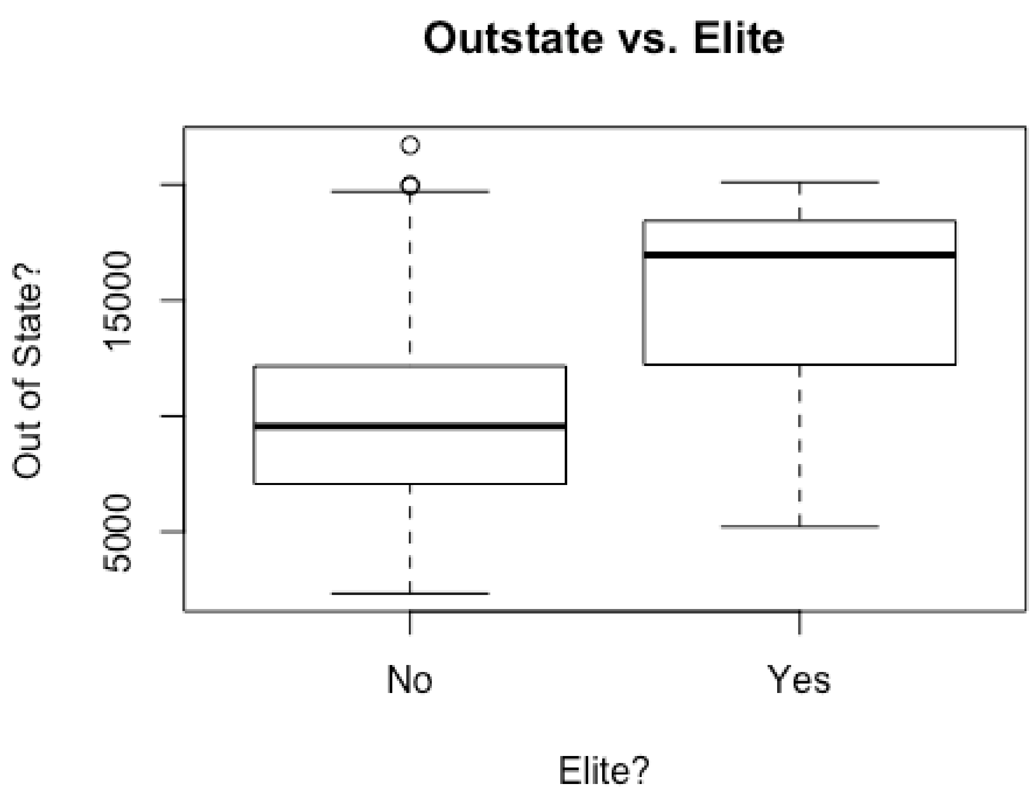
```r
# adds the new data to the college data frame
college <- data.frame(college, Elite)
```

Problem 2 (c) (v):

```r
# Use the summary() function to see how many elite universities
# there are.

summary(college$Elite)

##  No Yes
## 699  78
```

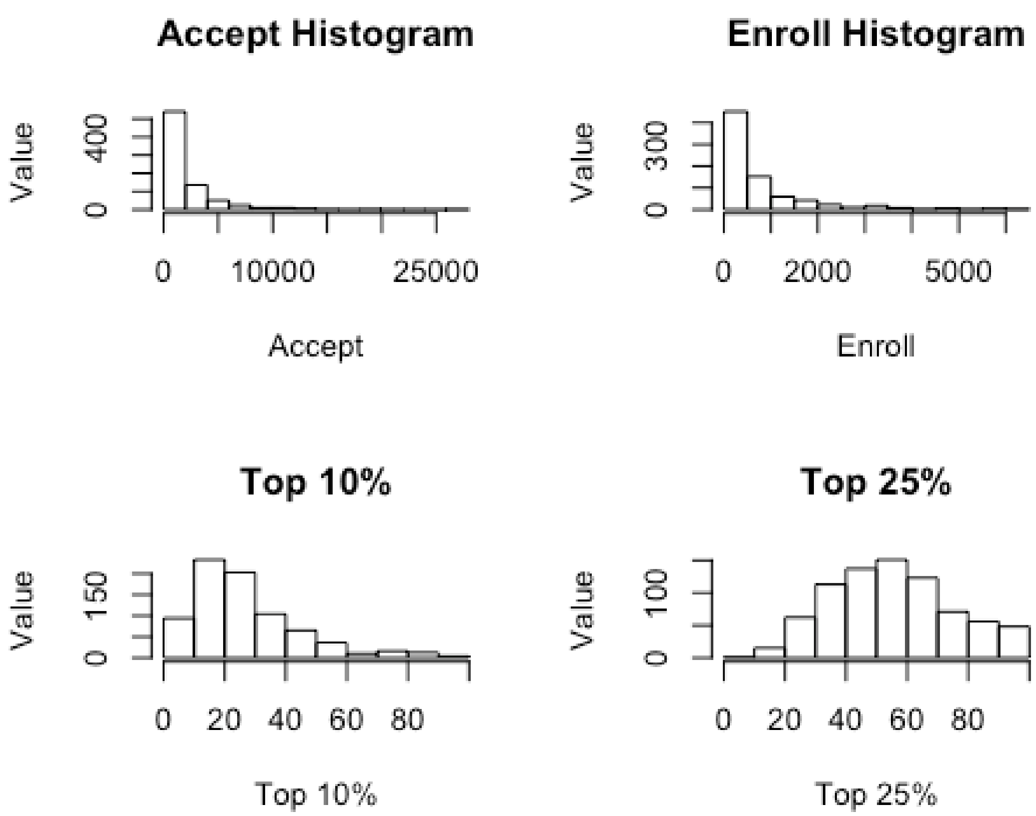
Problem 2 (c) (vi):

```r
# Now use the plot() function to produce side-by-side boxplots of
# Outstate versus Elite. Label the axes and main title
# appropriately.

plot(college$Outstate ~ college$Elite, main  ="Outstate vs. Elite", xlab =
"Elite?", ylab = "Out of State?")
```



Problem 2 (c) (vii):

```r
# Use the hist() function to produce some histograms with differing
# numbers of bins for a few of the quantitative variables. You may
# find the command par(mfrow=c(2,2)) useful: it will divide the
# print window into four regions so that four plots can be made
# simultaneously. Modifying the arguments to this function will
# divide the screen in other ways.

par(mfrow=c(2,2))

hist(college$Accept, main = "Accept Histogram", xlab = "Accept", ylab =
"Value")
hist(college$Enroll, main = "Enroll Histogram", xlab = "Enroll", ylab =
"Value")
hist(college$Top10perc, main = "Top 10%", xlab = "Top 10%", ylab = "Value")
hist(college$Top25perc, main = "Top 25%", xlab = "Top 25%", ylab = "Value")
```



Vigneshwaran Dharmarajan: -1: you should change the number of bins

Problem 3: Using R: Manipulating Data in Data Frames

Problem 3 (a):

```r
# Load the data frame baseball in the plyr package. Use ?baseball
# to get information about the data set and definitions for the
# variables.

data("baseball")

??baseball
```

Problem 3 (b):

```r
# You will calculate the on base percentage for each player, but
# first clean up the data:

# Before 1954, sacrifice flies were counted as part of sacrifice
# hits, so for players before 1954, sacrifice flies (i.e. the
# variable sf) should be set to 0.

baseball$sf[baseball$year < 1954] <- 0

# Hit by pitch (the variable hbp) is often missing - set these
# missings to 0.

baseball$hbp[is.na(baseball$hbp)] <- 0

# Exclude all player records with fewer than 50 at bats (the
# variable ab).

baseball <- baseball[!(baseball$ab < 50),]
```

Problem 3 (c):

```r
# Compute on base percentage in the variable obp according to the
# formula:

obp = (baseball$h + baseball$bb + baseball$hbp) / (baseball$ab + baseball$bb
+ baseball$hbp + baseball$sf)

baseball <- data.frame(baseball, obp)
```

Problem 3 (d):

```r
# Sort the data based on the computed obp and print the year,
# player name, and on base percentage for the top five records
```

```r
# based on this value.

head(baseball[order(baseball$obp, decreasing=TRUE), c(1,2,23)], 5)
```

```
##              id year       obp
## 84983 bondsba01 2004 0.6094003
## 82594 bondsba01 2002 0.5816993
## 29489 willite01 1941 0.5528053
## 7772  mcgrajo01 1899 0.5474860
## 19883  ruthba01 1923 0.5445402
```

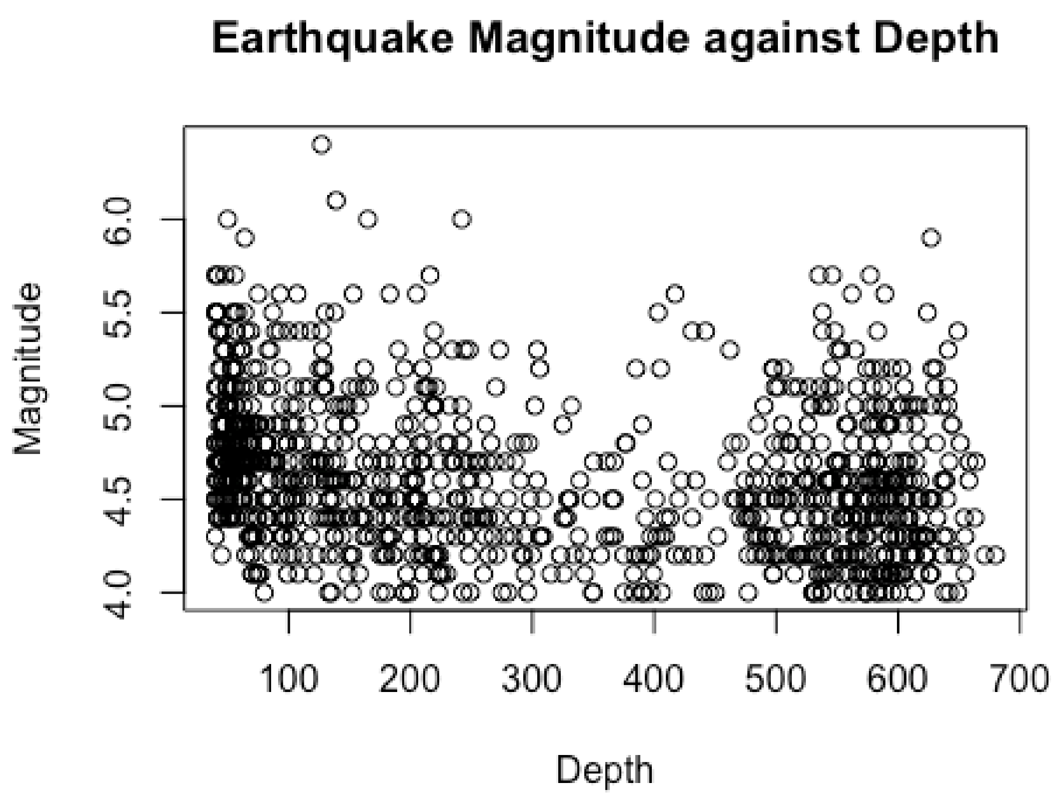
Problem 4: Using R: aggregate() function

Problem 4 (a):

```r
# Load the quakes data from the datasets package.

data("quakes")
```

Problem 4 (b):

```r
# Plot the recorded earthquake magnitude against the earthquake
# depth using the plot command.

plot(quakes$mag ~ quakes$depth, main = "Earthquake Magnitude against Depth",
xlab = "Depth", ylab = "Magnitude")
```



Problem 4 (c):

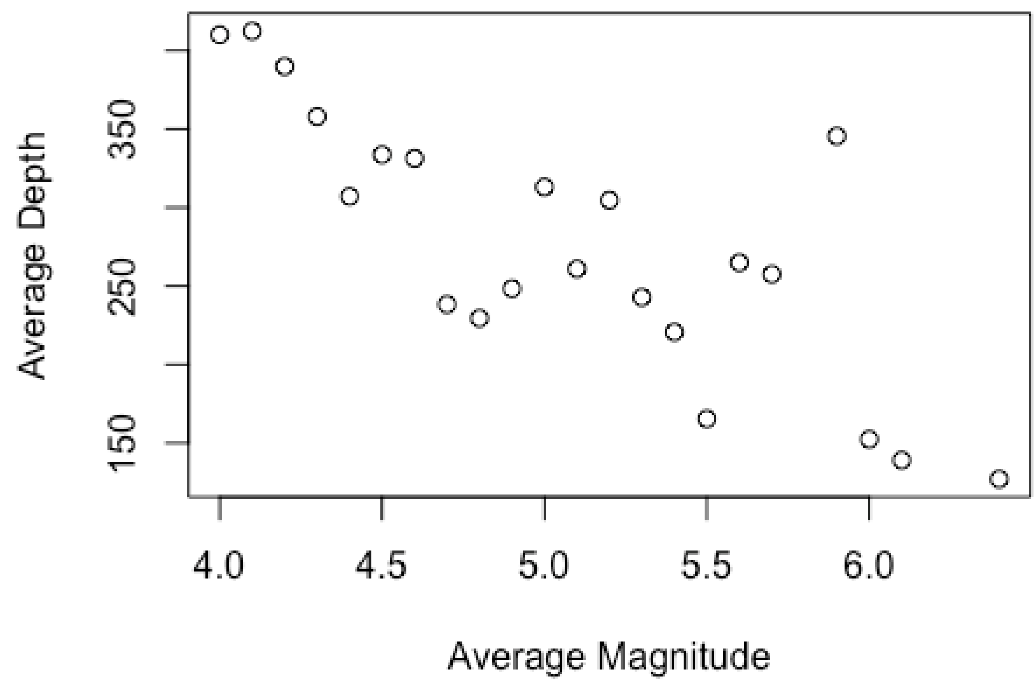
```
# Use aggregate to compute the average earthquake depth for each
# magnitude level. Store these results in a new data frame named
# quakeAvgDepth.

quakeAvgDepth <- aggregate(depth~mag, data=quakes, FUN=mean)
```

Problem 4 (d):

```
# Rename the variables in quakeAvgDepth to something meaningful.

colnames(quakeAvgDepth)[1] <- "MagnitudeInterval"
colnames(quakeAvgDepth)[2] <- "DepthInterval"
```

Problem 4 (e):

```
# Plot the magnitude vs. the average depth.

plot(quakeAvgDepth$MagnitudeInterval, quakeAvgDepth$DepthInterval, xlab =
"Average Magnitude", ylab = "Average Depth")
```



Problem 4 (f):

```
# From the two plots, do you think there is a relationship between
# earthquake depth and magnitude?

# It seems to be on average that the greater the depth, the lesser
# the magnitude on average. And the level of depth decreases, the
# average magnitude increases overall.
```

Vigneshwaran Dharmarajan: -3 Too general. Be more explanatory: which plot suggests this? are there any differences between the two of them or can you reach to the same conclusion from both?