Assignment #1

Saied Hosseinipoor

August 30, 2016

Libraries

The following libraries was used in this assignment:

```
library(e1071)                # package for calculaiton of skewness and kurtosis
library(plyr)                 # package including data set for problem #3
library(datasets)             # package including "quakes" data set for problem #4
```

if the package "e1071" is not istalled please run install.packages("e1071").

Problems

Problems #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)    # define a vector with specific numbers and assign to x
x
```

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

Problem 1(b)

Using the commands seq, min, and max with one line of code 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)    # Creat vector y consist of 10 element from min(x) to max(x
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, quartiles, and quintiles for x and y.

```r
sum(x); sum(y)                # summation elements for "x" and "y"
```

```
## [1] 37
```

```
## [1] 25
```

```r
mean(x); mean(y)              # mean values for "x" and "y"
```

```
## [1] 3.7
```

```
## [1] 2.5
```

```r
sd (x); sd (y)                # standard deviation values for "x" and "y"
```

```
## [1] 7.572611
```

```
## [1] 8.41014
```

```r
var(x); var(y)                # Variances for "x" and "y"
```

```
## [1] 57.34444
```

```
## [1] 70.73045
```

```r
mad(x); mad(y)                # Mean absolute deviation values for "x" and "y"
```

```
## [1] 5.9304
```

```
## [1] 10.29583
```

```r
quantile(x); quantile(y)      # Quartiles for "x" and "y"
```

```
##     0%    25%    50%    75%   100%
## -10.00   0.25   4.50   7.75  15.00
```

```
##     0%    25%    50%    75%   100%
## -10.00  -3.75   2.50   8.75  15.00
```

```r
quantile(x, probs = seq(0, 1, 0.2)); quantile(y, probs = seq(0, 1, 0.2))    # Quintiles for "x" and "y"
```

```
##    0%   20%   40%   60%   80%  100%
## -10.0  -1.0   2.2   6.4   8.8  15.0
```

```
##            0%           20%           40%           60%           80%
## -1.000000e+01 -5.000000e+00 -1.665335e-15  5.000000e+00  1.000000e+01
##          100%
##  1.500000e+01
```

Problem 1(d)

Create a new 7 element vector z by using R to randomly sample from x with replacement.

```r
z = sample(x, 7, replace = T)
```

Problem 1(e)

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.

```r
skewness(x)                 # Skewness of vector "x" from package e1071
```

```
## [1] -0.2667237
```

```r
kurtosis(x)                 # Kurtosis of vector "x" from package e1071
```

```
## [1] -1.092184
```

Problem 1(f)

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

```r
t.test(x, y)                # t-test to compare the mean of "x" and "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
```

The reults shows the difference between two sets are not significant. The p-values for the test is %76 which is very higher than %5 (or any acceptable reference values). The confidence interval is also (-2.89, 10.89) which contains zero.

Problem 1(g)

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

```r
t.test(sort(x), y, paired = T)          # t-test to compare the mena of sorted "x" and "y"
```

```
##
##  Paired t-test
##
## data:  sort(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
```

Problem 1(h)

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

```
neg_x = x < 0                # Logical vector for x < 0
neg_x
```

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

Problem 1(i)

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[!neg_x]                # Recreat "x" vector with non-negative elemets of old "x"
x
```

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

Problem #2: Using R: Introductory data exploration

> This exercise relates to the College data set, which can be found in the file "College.csv" in D2L. The file contains a number of variables for 777 different universities and colleges in the US. The variables are:

Private : Public/private indicator

Apps : Number of applications received

Accept : Number of applicants accepted

Enroll : Number of new students enrolled

Top10perc : New students from top 10

Top25perc : New students from top 25

F.Undergrad : Number of full-time undergraduates

P.Undergrad : Number of part-time undergraduates

Outstate : Out-of-state tuition

Room.Board : Room and board costs

Books : Estimated book costs

Personal : Estimated personal spending

PhD : Percent of faculty with Ph.D.s

Terminal : Percent of faculty with terminal degree

S.F.Ratio : Student/faculty ratio

perc.alumni : Percent of alumni who donate

Expend : Instructional expenditure per student

Grad.Rate : Graduation rate

Before reading the data into R, it can be viewed in Excel or a text editor.

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")      # Reads data from file "college.csv" and assign to data fram "col
```

Problem 2(b)

Look at the data using RStudio. You should notice that the

rst column is just the name of each university. We dont really want R to treat this as data. However, it may be handy to have these names for later. Try the following commands:

rownames (college) <- college [,1] View (college)

You should see that there is now a row.names column with the name of each university recorded. This means that R has given each row a name corresponding to the appropriate university. R will not try to perform calculations on the row names. However, we still need to eliminate the first column in the data where the names are stored. Try

college <- college [,-1]

and then view the data (either with the View command or clicking on the college data frame in the RStudio workspace window) Now you should see that the

rst data column is Private.

```
rownames(college) <- college[,1]      # Use the first column of data as row's name
View (college )                       # Display the content of the dataframe
college <- college[,-1]               # Remove the first column's data
```

Problem 2(c)

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

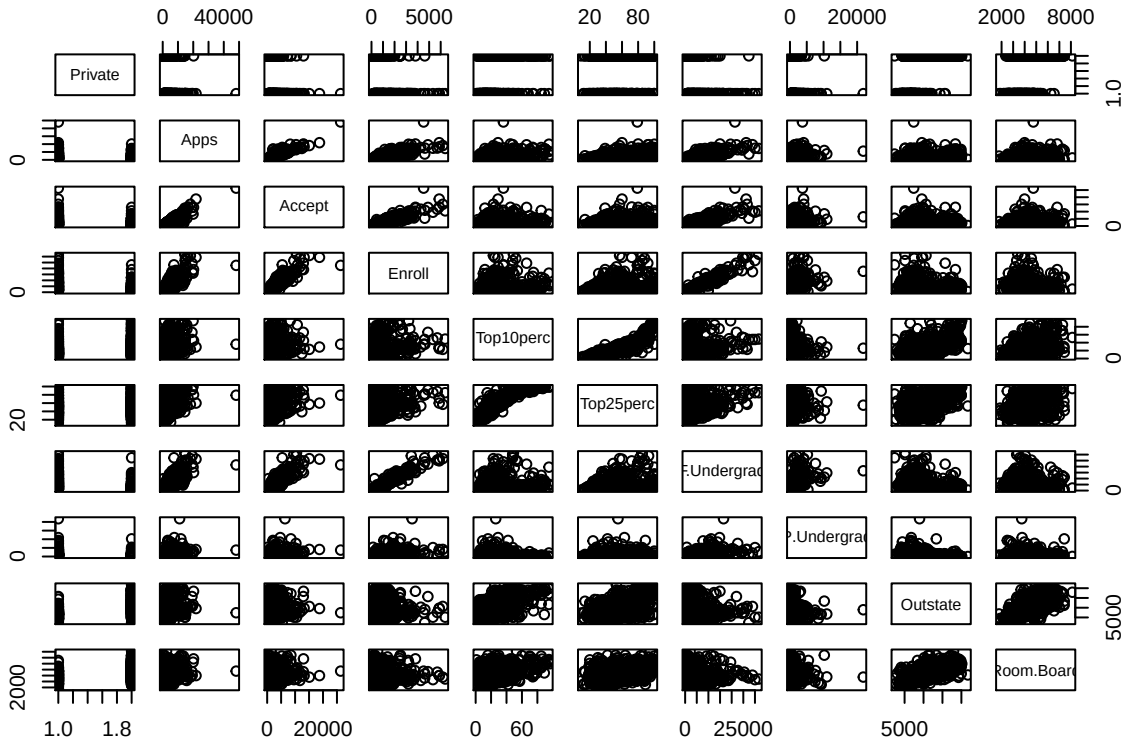
```
summary(college)                      # Shows the numerical summaries for "college"
```

```
##  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
```

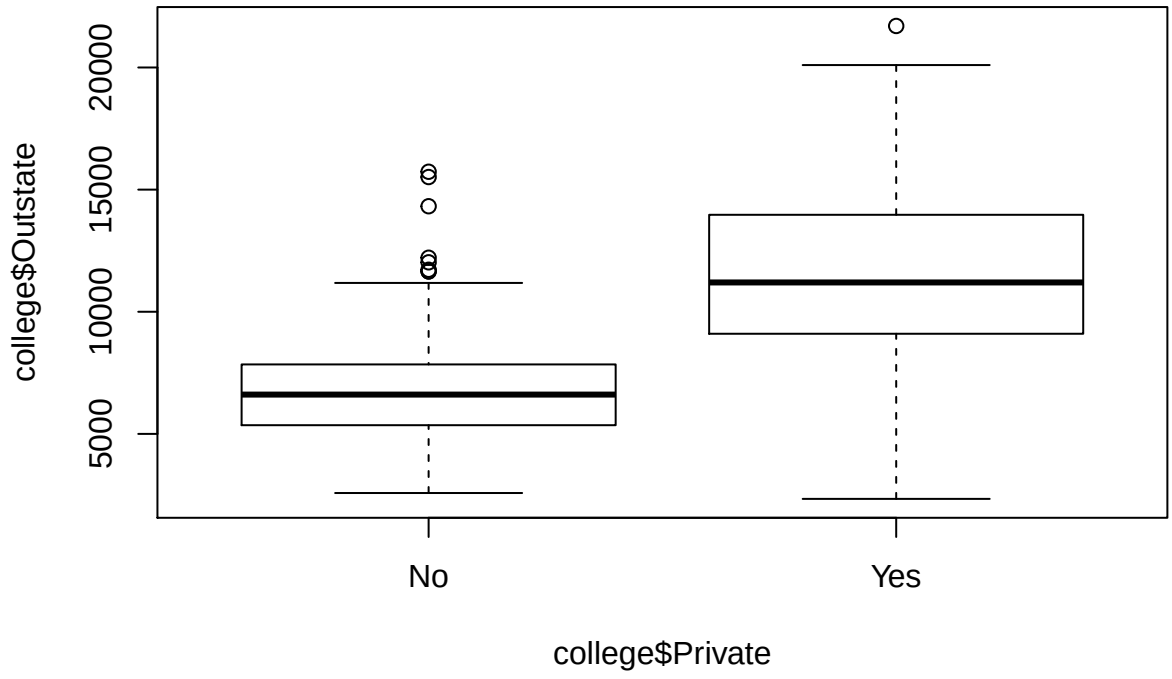
ii. Access help for the pairs function and then use pairs to produce a scatterplot matrix of the first ten columns. Recall that you can reference the
rst ten columns of a matrix A using **A[,1:10]**.

```
help("pairs")                          # Shows the description for "pairs" function
pairs(college[,1:10])                  # Used pairs fucntion to produce a scatterplot matrix for the fir
```



iii. 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)  # Side-by-side boxplot for Outstate vs. Private
```



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.

Elite <- rep ("No", nrow(college))

Elite [college$Top10perc >50] <- "Yes"

Elite <- as.factor (Elite)

college <- data.frame(college ,Elite)

```
Elite <- rep("No", nrow(college))          # Creates a vector in size of rows of the college filling wit
Elite [college$Top10perc > 50] <- "Yes"    # Find the Top10prec greater than 50 and replace the correspo
Elite <- as.factor(Elite)                  # Converts the "Elite" vector into a factor vector with 2 lev
college <- data.frame(college, Elite)      # Adds "Elite" as a new column to the "College" data frame
```

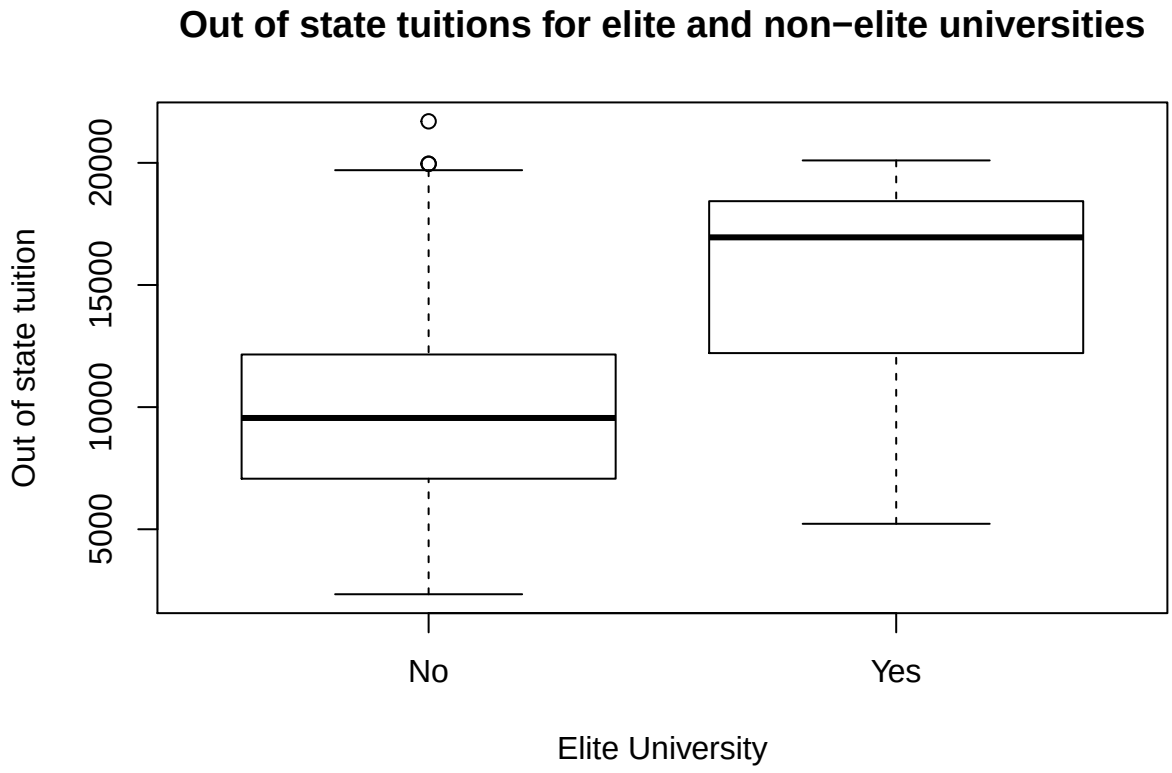
v. Use the summary() function to see how many elite universities there are.

```
summary(college$Elite)                     # Shows the number of teh elite and non-elite colleges
```

```
##  No Yes
## 699  78
```

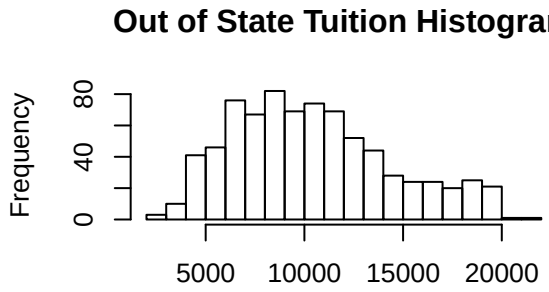
vi. Now use the plot() function to produce side-by-side boxplots of Outstate versus Elite. Label the axes and main title appropriately.

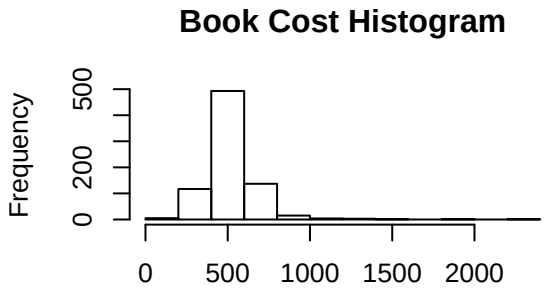
```
plot(college$Outstate~college$Elite,
     xlab = "Elite University",
     ylab = "Out of state tuition",
     main = "Out of state tuitions for elite and non-elite universities"
     )  # Side-by-side boxplot for Outstate vs. Elite
```

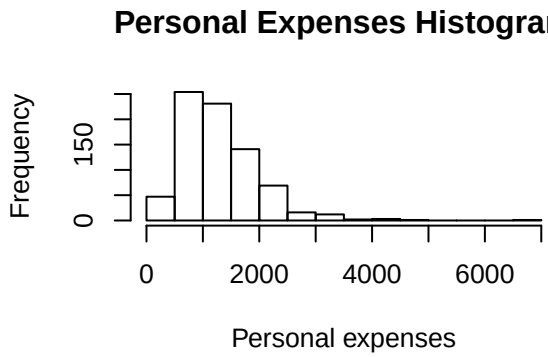


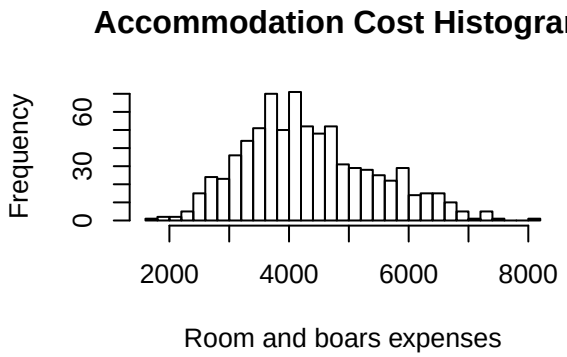
vii. Use the **hist()** function to produce some histograms with differing numbers of bins for a few of the quantitative variables. You may find the command **par(mfrow=c(2,2))** useful: it will divide the print window into four regions so that four plots can be made simultaneously. Modifying the arguments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

```
par(mfrow = c(2,2))        # Divides the screen into 4 winows
# Draw histograms for out of state tuitions, book costs, personal expenses, and accommodation costs.
hist(college$Outstate, xlab = "Out of state tuition", main = "Out of State Tuition Histogram", breaks =
hist(college$Books, xlab = "Estimted book costs", main = "Book Cost Histogram")
hist(college$Personal, xlab = "Personal expenses", main = "Personal Expenses Histogram", breaks = 15)
hist(college$Room.Board, xlab = "Room and boars expenses", main = "Accommodation Cost Histogram", breaks
```









Problem #3: Using R: Manipulating data in data frames

Problem 3(a)

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

```
?baseball                  # Shows the help and description about basegall dataset fron plyr package
```

Problem 3(b)

You will calculate the *on base percentage* for each player, but first clean up the data: * Before 1954, sacrifice fies 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                  # Set 0 "sacrifies flies" for players beore 1954
```

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

```
baseball$hbp[!complete.cases(baseball$hbp)] <- 0  # Set 0 missing values for "Hit by pitch"
```

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

```
baseball <- subset(baseball, baseball$ab>=50)      # Excludes all player records with fewer than 50 at b
```

Problem 3(c)

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

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

```
obp = with(baseball, (h + bb + hbp) / (ab + bb + hbp + sf)) # Compute on base percentage
baseball = data.frame(baseball, obp)    # Adds "obp" as data in a new column
```

Problem 3(d)

Sort the data based on the computed obp and print the year, player name, and on base percentage for the top five records based on this value.

```
baseball <- baseball[order(-baseball$obp),]       # Sorts the sata set based on "obp"h
head(baseball[,c("id", "year", "obp")], n =5)   # Displays the top 5 on base percentage
```

```
##              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

The aggregate function is very useful method in R and allows you to easily compute statistics (such as the mean) for different groupings, e.g. if you have a set of data for students which contains both demographic and grade information; to compute the mean class grade by gender, you could use the aggregate command. To complete this problem, you will need to look up information on how to use aggregate. You can use the built-in R documentation, look for help online, or both.

Problem 4(a)

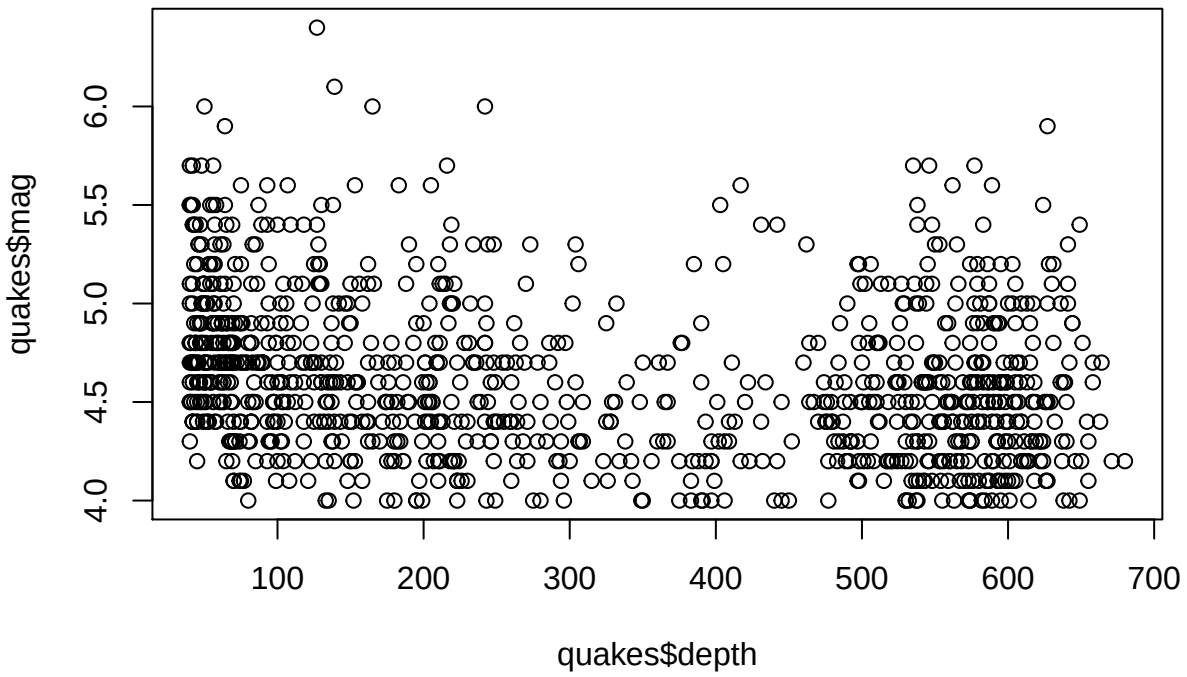
Load the quakes data from the datasets package.

```
#library(datasets)              # package including "quakes" data set for problem #4
```

Problem 4(b)

Plot the recorded earthquake magnitude against the earthquake depth using the plot command.

```
par(mfrow=c(1,1))                       # Reset display windows
plot(quakes$mag~quakes$depth)           # Plots earthquake magnetude agianst the depth
```



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(quakes$depth, list("Magnetude level"= quakes$mag), mean)
```

Problem 4(d)

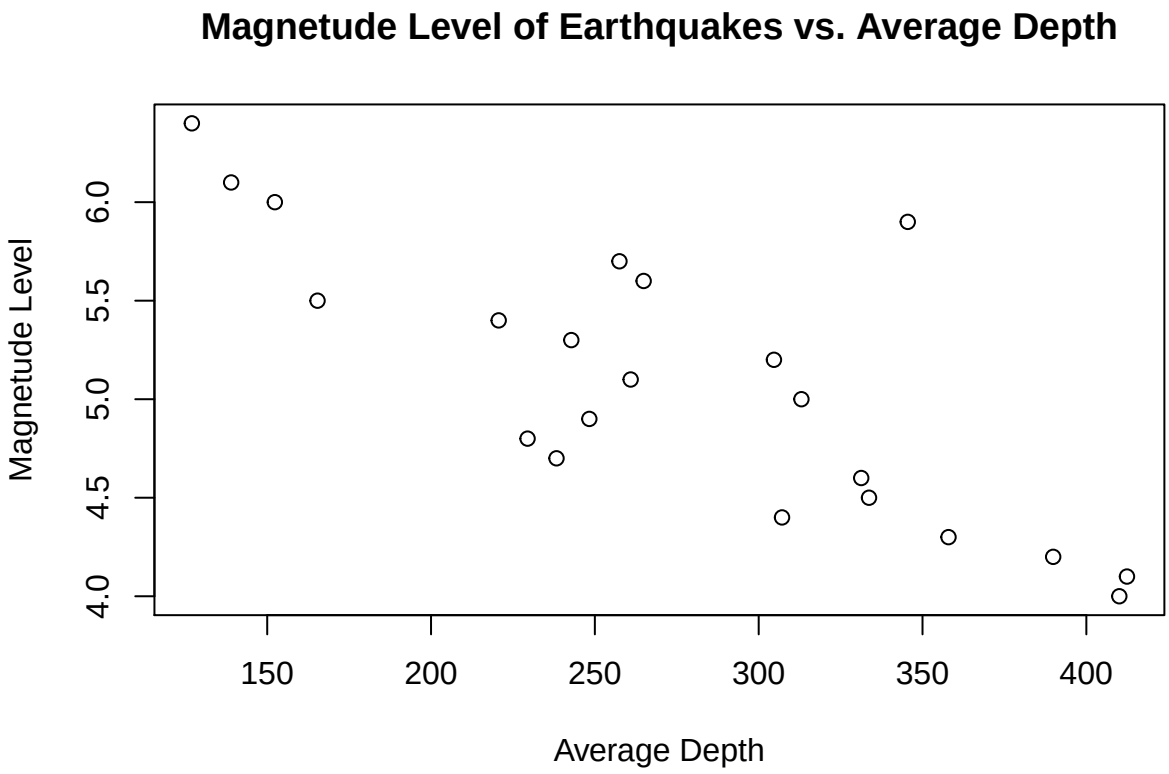
Rename the variables in quakeAvgDepth to something meaningful.

```
names(quakeAvgDepth) <- c("Magnetude Level", "Average Depth")
```

Problem 4(e)

Plot the magnitude vs. the average depth.

```
plot(quakeAvgDepth$`Magnetude Level` ~ quakeAvgDepth$`Average Depth`,
     xlab = "Average Depth",
     ylab = "Magnetude Level",
     main = "Magnetude Level of Earthquakes vs. Average Depth")
```



Problem 4(f)

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

The plot shows that increament in the depth has reverse effect on the earthquakes. In the other words we can say that if an earthquake occurs near to the surface, there is a higher chance to be a stronger one.