# Assignment #1

Jiwon Jeon

September 3, 2016

```
#Assignment #1
#Name: Jiwon Jeon
#ISE 5103 Intelligent Data Analytics
#Date: 09/03/2016

#required packages for this assigment
library(lsr)  #provides the statistical measure for Problem 1(c)
library(e1071)  #provides the statistical measures for Problem 1(e)
library(plyr)  #provides the statistical data for Problem 3
library(datasets)  #provides datasets for Problem 4
```

Problem 1: Vectors

```
#Problem 1(a)
x = c(3,12,6,-5,0,8,15,1,-10,7)  #a vector x with 10 numbers
x

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

#Problem 1(b)
y = seq(min(x), max(x), length = 10)  #a vector y with 10 elements
                                      #between minimum x and maximum 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)
sum(x)  #sum of x

## [1] 37

mean(x)  #mean of x

## [1] 3.7

sd(x)  #standard deviation of x

## [1] 7.572611

var(x)  #variance of x

## [1] 57.34444
```

```
aad(x)  #mean absolute deviation of x (uses package 'lsr')

## [1] 5.9

mad(x)  #median absolute deviation of x

## [1] 5.9304

quantile(x)  #quartile of x

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

quantile(x, probs = seq(0,1,0.2))  #quintile of x

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

sum(y)  #sum of y

## [1] 25

mean(y)  #mean of y

## [1] 2.5

sd(y)  #standard deviation of y

## [1] 8.41014

var(y)  #variance of y

## [1] 70.73045

aad(y)  #mean absolute deviation of y (uses package 'lsr')

## [1] 6.944444

mad(y)  #median absolute deviation of y

## [1] 10.29583

quantile(y)  #quartile of y

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

quantile(y, probs = seq(0,1,0.2))  #quintile of y

##            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)
z = sample(x, 7, replace = TRUE)  #a vector z with 7 random numbers
                                  #from x with replacement
z

## [1]  8 15  3 12  6 -5  8

#Problem 1(e): uses package 'e1071'
skewness(x)  #skewness of x

## [1] -0.2667237

kurtosis(x)  #kurtosis of x

## [1] -1.092184

#Problem 1(f)
t.test(x,y)  #statistical test between the vectors 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
```

Mean of x is 3.7 while mean of y is 2.5. The difference in these two means is not significant compared to the range of x and y. For individual statistical test, `t.test(x)` and `t.test(y)` can be used, respectively.

```
#Problem 1(g)
sort(x)  #sorts the vector x in ascending order

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

t.test(x,sort(x))  #t-test for x and sort(x)

## 
##  Welch Two Sample t-test
## 
## data:  x and sort(x)
## t = 0, df = 18, p-value = 1
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.11493  7.11493
## sample estimates:
```

```
## mean of x mean of y
##       3.7       3.7
```

Mean of x and mean of sort(x) are the same; 3.7.

```
#Problem 1(h)
x<0  #a logical vector to identify negative numbers in x

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

#Problem 1(i):
x = x[x>=0]  #removes the negative numbers from x
x

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

Problem 2: Introductory data exploration

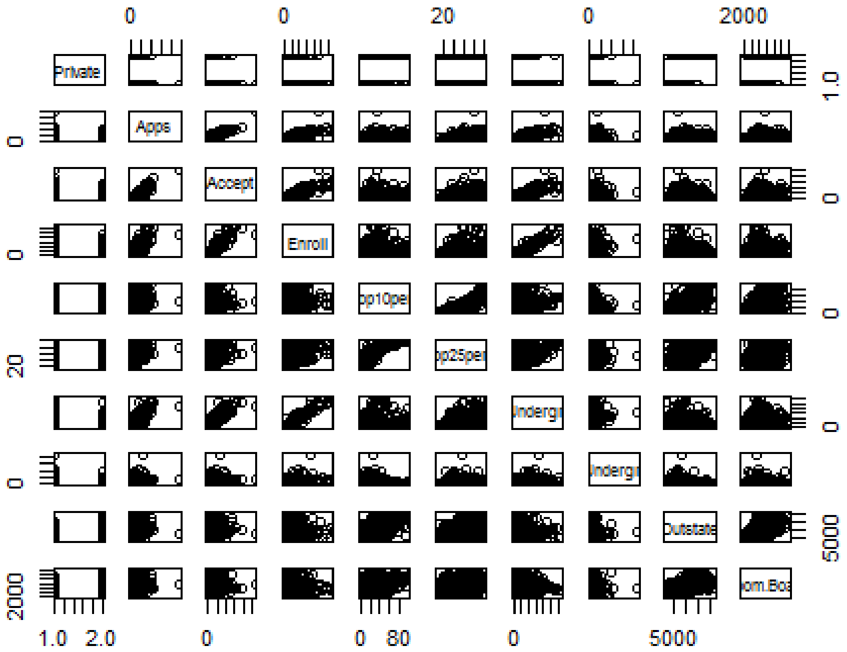
```
#Problem 2(a)
college = read.csv("college.csv", header = TRUE)  #reads the data file "Colle
ge.csv"
college = data.frame(college)  #loads the data as data frame

#Problem 2(b)
rownames(college) = college[,1]  #displays the row.names with the name
                                 #in the first column
View(college)  #views the data
college = college[,-1]  #removes the generated column for row.names
View(college)

#Problem 2(c).i:
summary(college)  #produces a numerical summary

##  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
?pairs  #help for the pairs()
```

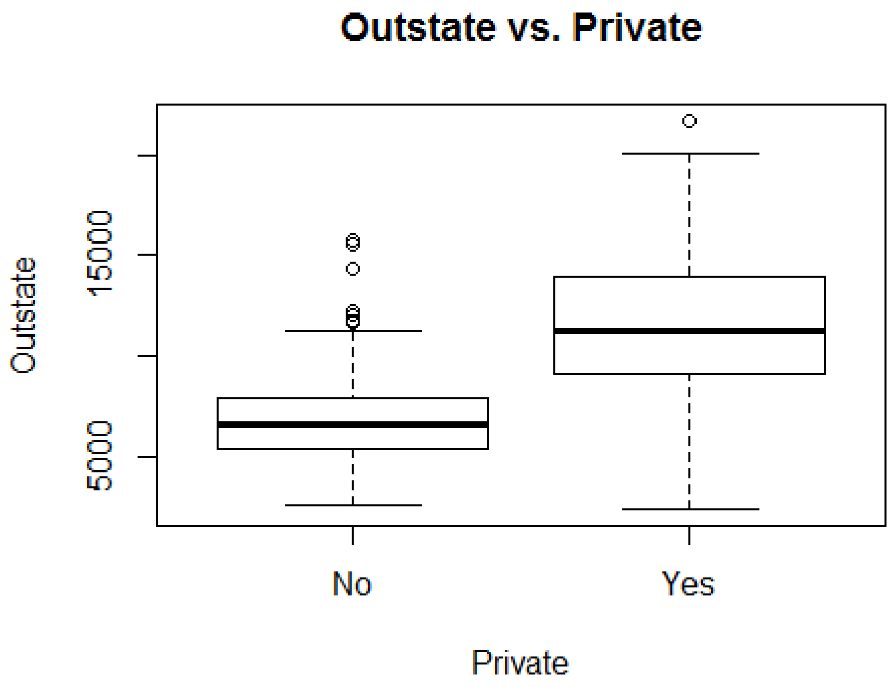
```
## starting httpd help server ...
```

```
##  done
```

```
pairs(college[,1:10])  #produces a scatterplot matrix of the first ten column
s
```



```
#Problem 2(c).iii
plot(college$Private, college$Outstate, xlab = "Private", ylab = "Outstate",
     main = "Outstate vs. Private")  #boxplots of Outstate vs. Private
```

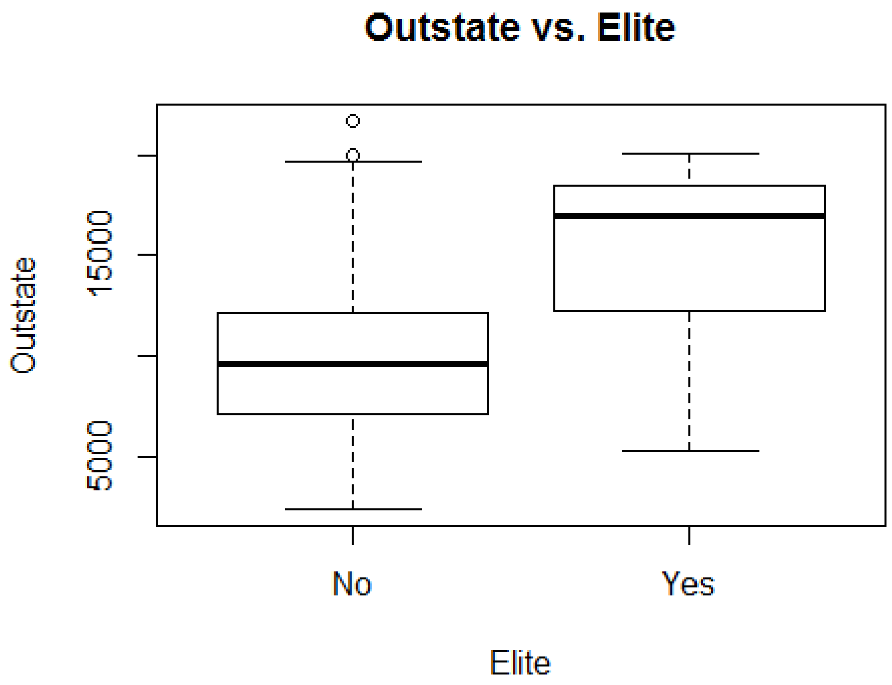


```
#Problem 2(c).iv
#creates 777 replicated value of "No", and names the vector as Elite
Elite = rep("No", nrow(college))

#creates a logical vector to see if the proportion of students from
#the top 10% of their high school classes exceeds 50%.
#If the condition is TRUE, replaces "No" to "Yes"
Elite [college$Top10perc > 50] = "Yes"

#converts a vector of Elite into a factor to recognize "Yes" or "No"
#in column of data frame
Elite = as.factor(Elite)

#finishes creating a new qualitative variable, Elite, by combining
#the data frame college and Elite
college = data.frame(college, Elite)

#Problem 2(c).v
summary(college)
```

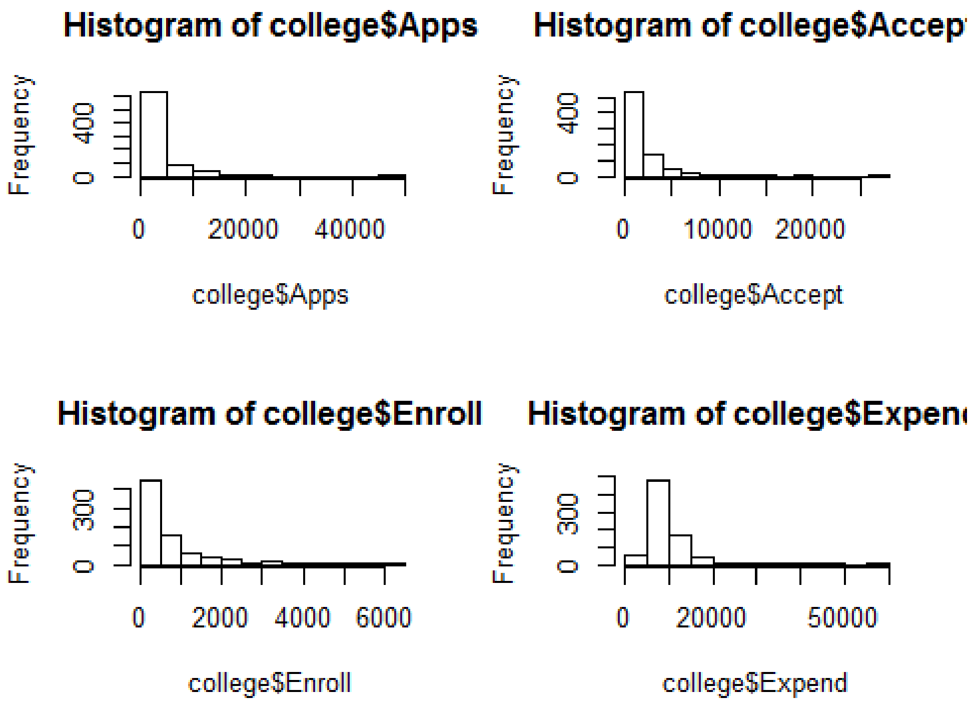
```
##  Private        Apps          Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
##             Median : 1558   Median : 1110   Median : 434   Median :23.00
##             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      Elite
##  Min.   : 10.00   No :699
##  1st Qu.: 53.00   Yes: 78
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

There are 78 elite universities out of 777 universities.

```
#Problem 2(c).vi
plot(college$Elite, college$Outstate, xlab = "Elite", ylab = "Outstate",
     main = "Outstate vs. Elite")  #boxplots of Outstate vs. Elite
```

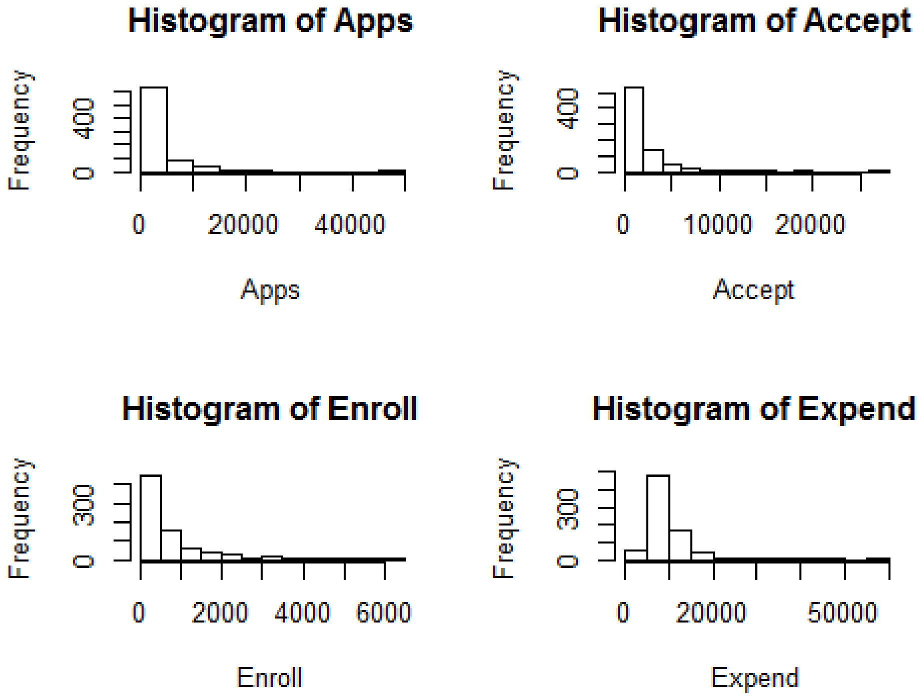


```
#Problem 2(c).vii
par(mfrow=c(2,2))  #divides the print window into four regions
hist(college$Apps)  #histogram for number of applications received
hist(college$Accept)  #histogram for number of applicants accepted
hist(college$Enroll)  #histogram for number of new students enrolled
hist(college$Expend)  #histogram for instructional expenditure/student
```



Problem 2(c).vii can be written as following code to trim the names of axes and main title:

```
attach(college)  #attaches data to the R search path to access simply with th
eir names

## The following object is masked _by_ .GlobalEnv:
##
##     Elite

par(mfrow=c(2,2))
hist(Apps)
hist(Accept)
hist(Enroll)
hist(Expend)
```



Problem 3: Manipulating data in data frames

```
#Problem 3(a): uses package 'plyr'
baseball = data.frame(baseball)
?baseball

#Problem 3(b)
baseball$sf[baseball$year < 1954] = 0  #sets sacrifice flies (sf) to 0 before
 1954
```

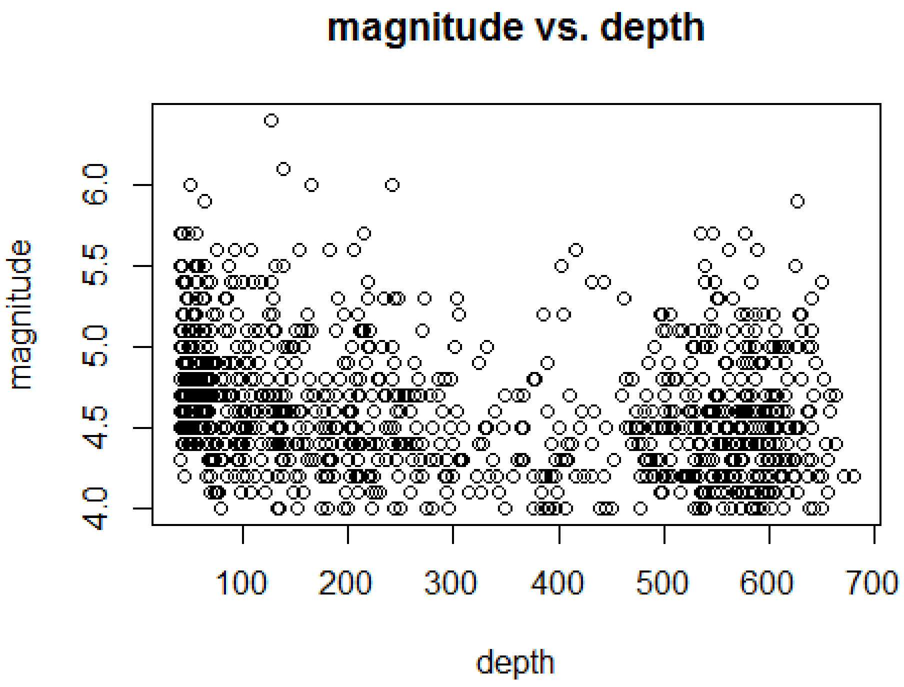
This work also can be done using `for()` and `if()` loop as below:

```
for(i in 1:nrow(baseball)){
  if(baseball$year[i] < 1954){
    baseball$sf[i] = 0
  }
}

baseball$hbp[is.na(baseball$hbp)] = 0  #sets missings in hit by pitch (hbp) t
o 0
baseball = baseball[baseball$ab >= 50,]  #excludes all player records with
                                         #fewer than 50 at bats(ab)

#Problem 3(c)
#calculates on base percentage in the variable obp
baseball$obp = (baseball$h + baseball$bb + baseball$hbp)/
  (baseball$ab + baseball$bb + baseball$hbp + baseball$sf)
```

```
#Problem 3(d)
#sorts the data in descending order
baseball_order = baseball[order(-baseball$obp),]
#prints year, id (name), and obp for top five records
baseball = print(baseball_order[1:5, c("year", "id", "obp")])

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

Problem 4: `aggregate()` function

```
#Problem 4(a): uses package 'datasets'
quakes = data.frame(quakes)

#Problem 4(b)
#scatter plot of magnitude vs. depth
plot(quakes$depth, quakes$mag, xlab = "depth", ylab = "magnitude", main = "ma
gnitude vs. depth")
```

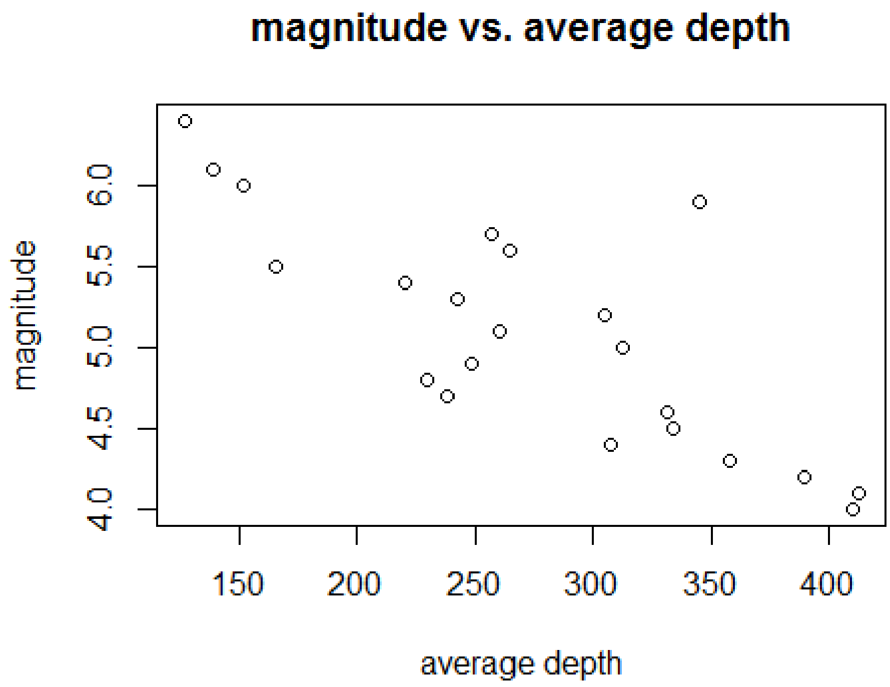


```
#Problem 4(c)
#computes the average earthquake depth for each magnitude level
quakeAvgDepth = aggregate(x = quakes$depth, by = list(quakes$mag), FUN = "mea
n")
```

```r
#Problem 4(d)
names(quakeAvgDepth)[1] = "magnitude"
names(quakeAvgDepth)[2] = "average depth"
```

This work also can be done using `colnames()`:

```r
colnames(quakeAvgDepth) = c("magnitude", "average depth")

quakeAvgDepth

##    magnitude average depth
## 1        4.0      410.0652
## 2        4.1      412.4000
## 3        4.2      389.8778
## 4        4.3      357.9294
## 5        4.4      307.1188
## 6        4.5      333.6729
## 7        4.6      331.2970
## 8        4.7      238.2959
## 9        4.8      229.4615
## 10       4.9      248.3148
## 11       5.0      313.0426
## 12       5.1      260.9302
## 13       5.2      304.6552
## 14       5.3      242.8095
## 15       5.4      220.6500
## 16       5.5      165.3571
## 17       5.6      264.8889
## 18       5.7      257.5000
## 19       5.9      345.5000
## 20       6.0      152.3333
## 21       6.1      139.0000
## 22       6.4      127.0000

#Problem 4(e)
#scatter plot of magnitude vs. average depth
plot(quakeAvgDepth$`average depth`, quakeAvgDepth$magnitude,
     xlab = "average depth", ylab = "magnitude", main = "magnitude vs. averag
e depth")
```



```
#Problem 4(f)
```

From the plot of magnitude vs. average depth, it can be said that the magnitude and average depth of earthquake has fairly inverse linear relationship. However, it is difficult to find this tendency from the plot of magnitude vs. depth.