DSA 5103

Intelligent Data Analytics

Homework #1

Library Commands:

```
library(tidyverse)

library(plyr)

library(datasets)
```

Problem 1:

1(a)

Creating a vector with 10 numbers (3, 12, 6, -5, 0, 8, 15, 1, -10, 7) and assigning it to x

Code:

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

x
```

Output:

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

1(b)

Creating a new vector y with 10 elements ranging from the minimum value of x to the maximum value of x

Code:

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

y
```

Output:

```
[1] -10.000000  -7.222222  -4.444444  -1.666667

[5]   1.111111   3.888889   6.666667   9.444444
```

[9] 12.222222 15.000000

1(c)

Computing the sum, mean, standard deviation, variance, mean absolute deviation, quartiles, and quintiles

Code:

#Sum of elements of x

sum(x)

Output:

[1] 37

Code:

#Mean of elements in x

mean(x)

Output

[1] 3.7

Code:

#Standard Deviation of x

sd(x)

Output:

[1] 7.572611

Code:

#Variance of x

var(x)

Output:

[1] 57.34444

Code:

#Mean Absolute Deviation od x

mad(x)

Output:

[1] 5.9304

Code:

#Quartiles for x

quantile(x,c(0.25, 0.5, 0.75,1))

Output:

25%   50%   75% 100%

0.25 4.50 7.75 15.00

Code:

#Quantiles for x

quantile(x,c(.20,.40,.60,.80,1))

Output:

 20% 40% 60% 80% 100%

-1.0 2.2 6.4 8.8 15.0

#Sum of elements of y

sum(y)

Output:

[1] 25

Code:

#Mean of elements in y

mean(y)

Output:

[1] 2.5

Code:

#Standard Deviation of y

sd(y)

Output:

[1] 8.41014

Code:

#Variance of y

var(y)

Output:

[1] 70.73045

Code:

#Mean Absolute Deviation of y

mad(y)

Output:

[1] 10.29583

Code:

#Quartiles for x

quantile(y,c(0.25, 0.5, 0.75,1))

Output:

25%   50%   75% 100%

-3.75 2.50 8.75 15.00

Code:

#Quantiles for x

quantile(y,c(.20,.40,.60,.80,1))

Output:

20%   40%   60%   80%

-5.000000e+00 -1.665335e-15 5.000000e+00 1.000000e+01

100%

1.500000e+01

1(d)

Using sample() to create a new 7 element vector z by to randomly sample from x with replacement

Code:

```
sample(x,7,replace=TRUE)
```

Output:

```
[1]   0   6  15 -10 -10  15   3
```

1(e)

The differences in mean are not significant.

Computing a statistical test for differences in means between the vectors x and y

Code:

```
t.test(x,y)
```

Output:

```
        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
```

1(f)

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

Code:

```
x[order(x)]
t.test(x,y,paired=TRUE)
```

```
        Paired t-test


data:  x and y
t = 0.30858, df = 9, p-value = 0.7647
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -7.596943  9.996943
sample estimates:
mean difference
            1.2
```

1(g)

Creating a logical vector that identifies which numbers in x are negative

Code:

```
z <- x<0
z
```

Output:

```
 [1] FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE
 [9]  TRUE FALSE
```

1(h)

Removing all entries with negative numbers from x

Code:

x[x>=0]

Output:

[1] 3 12 6 0 8 15 1 7

Problem 2:

2(a)

Displaying all rows in X with missing values

Code:

```
col1 <- c(1,2,3,NA,5)

col2 <- c(4,5,6,89,101)

col3 <- c(45,NA,66,121,201)

col4 <- c(14,NA,13,NA,27)

X <- data.frame(rbind (col1,col2,col3,col4))

#displaying all rows in X with missing values

X[!complete.cases(X),]
```

Output:

```
     X1 X2 X3  X4  X5
col1  1  2  3  NA   5
col3 45 NA 66 121 201
col4 14 NA 13  NA  27
```

2(b)(i)

Replacing any 99's in the vector y with 'NA'

Code:

```
y <- c(3,12,99,99,7,99,21)

y[y==99]<-NA

y
```

Output:

[1] 3 12 NA NA 7 NA 21

2(b)(ii)

Counting the number of missing values in y

Code:

sum(is.na(y))

Output:

[1] 3

Problem 3:

3(a)

Using the read.csv() function to read the data into a data frame and calling the data frame college.

Code:

#Reading the data into a data frame in R

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

#Calling the data frame college

college

3(b)

Code:

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

#Eliminate the first column in the data where the names are stored

college <- college [,-1]

Output:

	X	Private	Apps	Accept	Enroll	Top10perc	Top25perc	F.Undergrad
Abilene Christian University	Abilene Christian University	Yes	1660	1232	721	23	52	288
Adelphi University	Adelphi University	Yes	2186	1924	512	16	29	268
Adrian College	Adrian College	Yes	1428	1097	336	22	50	10
Agnes Scott College	Agnes Scott College	Yes	417	349	137	60	89	5
Alaska Pacific University	Alaska Pacific University	Yes	193	146	55	16	44	2

3(c)(i)

Using the summary() function to produce a numerical summary of the variables in the data set

Code:

summary(college)

Output:

```
  Private             Apps          Accept
 Length:777         Min.   :   81   Min.   :   72
 Class :character   1st Qu.:  776   1st Qu.:  604
 Mode  :character   Median : 1558   Median : 1110
                    Mean   : 3002   Mean   : 2019
                    3rd Qu.: 3624   3rd Qu.: 2424
                    Max.   :48094   Max.   :26330
    Enroll       Top10perc       Top25perc      F.Undergrad
 Min.   :  35   Min.   : 1.00   Min.   :  9.0   Min.   :  139
 1st Qu.: 242   1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992
 Median : 434   Median :23.00   Median : 54.0   Median : 1707
 Mean   : 780   Mean   :27.56   Mean   : 55.8   Mean   : 3700
 3rd Qu.: 902   3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005
 Max.   :6392   Max.   :96.00   Max.   :100.0   Max.   :31643
  P.Undergrad        Outstate       Room.Board
 Min.   :    1.0   Min.   : 2340   Min.   :1780
 1st Qu.:   95.0   1st Qu.: 7320   1st Qu.:3597
 Median :  353.0   Median : 9990   Median :4200
 Mean   :  855.3   Mean   :10441   Mean   :4358
 3rd Qu.:  967.0   3rd Qu.:12925   3rd Qu.:5050
 Max.   :21836.0   Max.   :21700   Max.   :8124
    Books           Personal         PhD
```

```
 Min.   :  96.0   Min.   : 250   Min.   :  8.00
 1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
 Median : 500.0   Median :1200   Median : 75.00
 Mean   : 549.4   Mean   :1341   Mean   : 72.66
 3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
 Max.   :2340.0   Max.   :6800   Max.   :103.00
    Terminal       S.F.Ratio      perc.alumni
 Min.   : 24.0   Min.   : 2.50   Min.   : 0.00
 1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00
 Median : 82.0   Median :13.60   Median :21.00
 Mean   : 79.7   Mean   :14.09   Mean   :22.74
 3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00
 Max.   :100.0   Max.   :39.80   Max.   :64.00
     Expend       Grad.Rate
 Min.   : 3186   Min.   : 10.00
 1st Qu.: 6751   1st Qu.: 53.00
 Median : 8377   Median : 65.00
 Mean   : 9660   Mean   : 65.46
 3rd Qu.:10830   3rd Qu.: 78.00
 Max.   :56233   Max.   :118.00
```

3(c)(ii)

Accessing help for the pairs function and then using pairs to produce a scatterplot matrix of the first ten columns

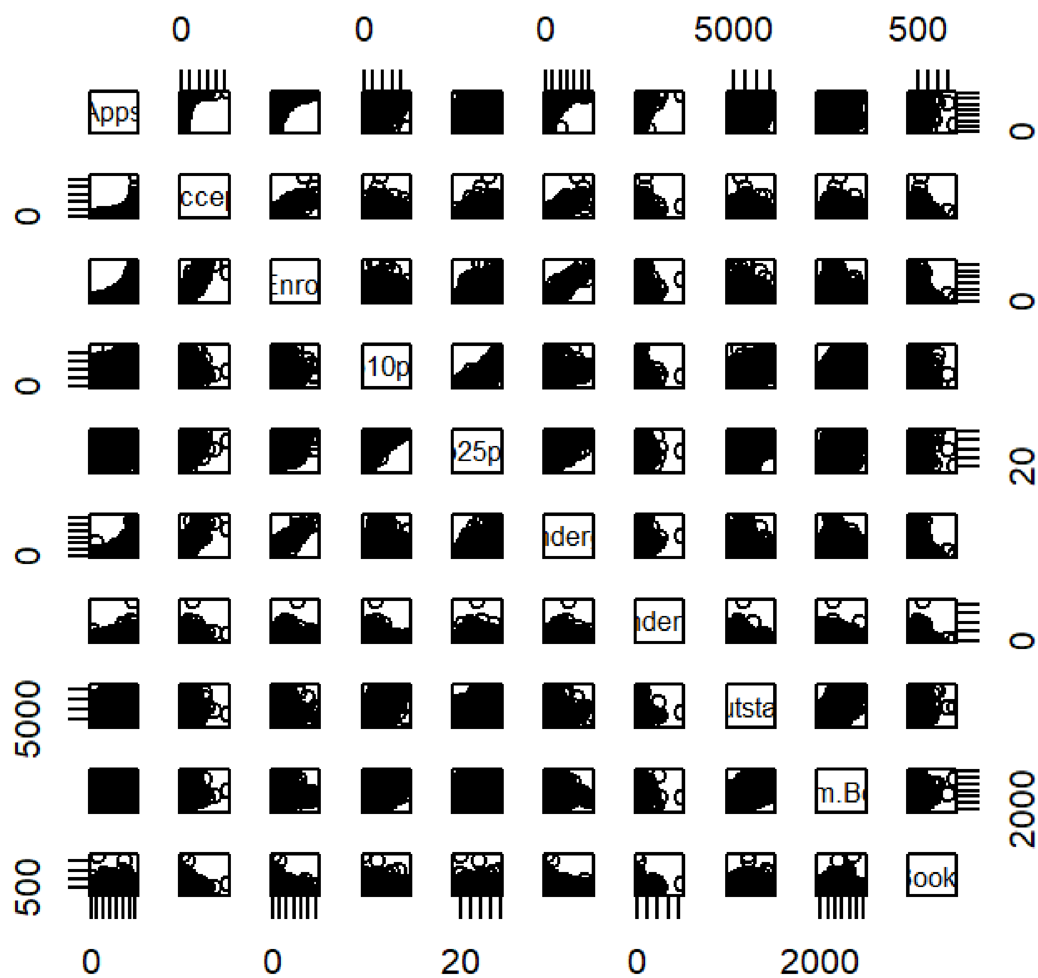
Code:

```
help("pairs")

college[,1] = as.numeric(factor(college[,1]))
```

pairs(college[,1:10])

Output:



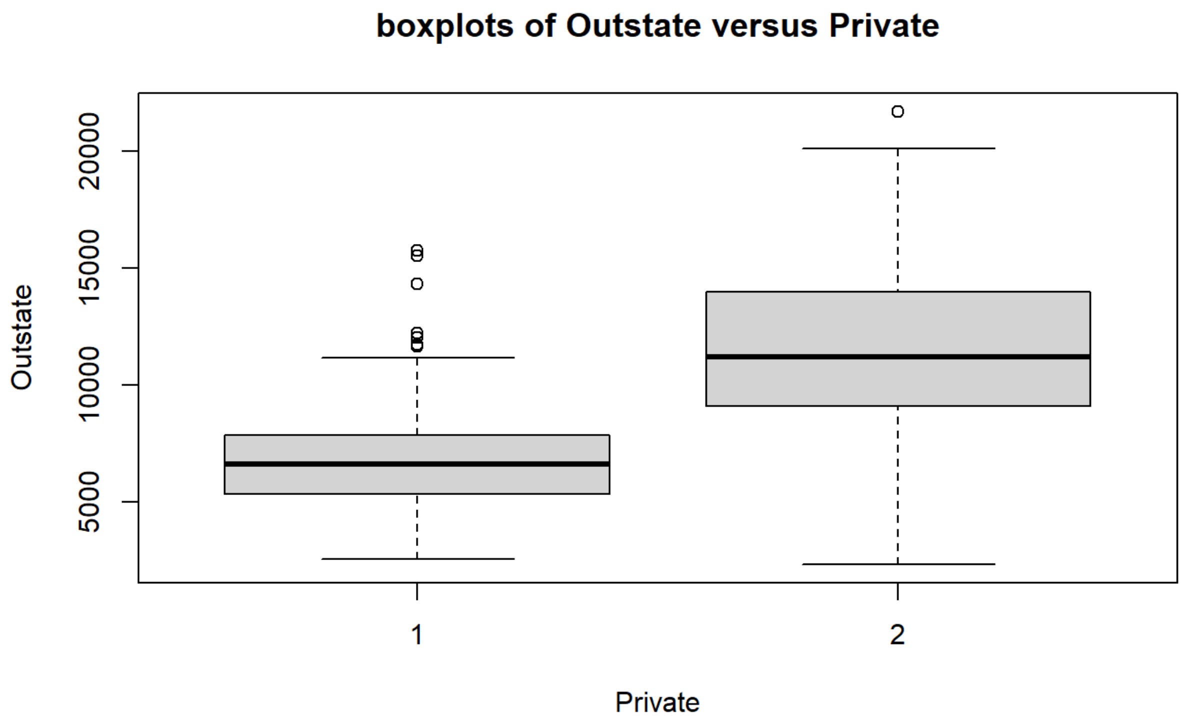
3(c)(iii)

Use the plot() function to produce side-by-side boxplots of Outstate versus Private

Code

college$Private<-as.factor(college$Private) plot(college$Private,college$Outstate,main= "boxplots of Outstate versus Private",xlab='Private',ylab='Outstate')

Output



3(c)(iv)

#Making all the values to "No" using rep function

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

#If the Top10perc variable is greater than 50 then we make it Yes

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

#Used to change the character to factor

Elite <- as.factor (Elite)

#Joining the elite column to college

college <- data.frame(college ,Elite)

3(c)(v)

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

Code:

```
summary(college$Elite[college$Elite=="Yes"])
```

Output:

```
No Yes
  0  78
```

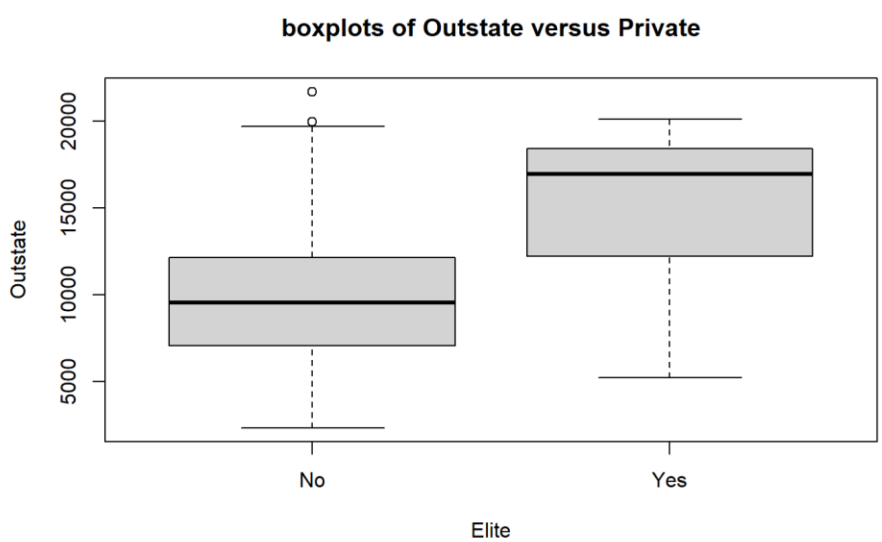
3(c)(vi)

Producing side-by-side boxplots of Outstate versus Elite

Code

```
college$Elite<-as.factor(college$Elite)
plot(college$Elite,college$Outstate,main="boxplots of Outstate versus Private",xlab='Elite',ylab='Outstate')
```

Output



3(c)(vii)

Dividing the print window into 4 screens and using the hist() function to produce histograms.

Code

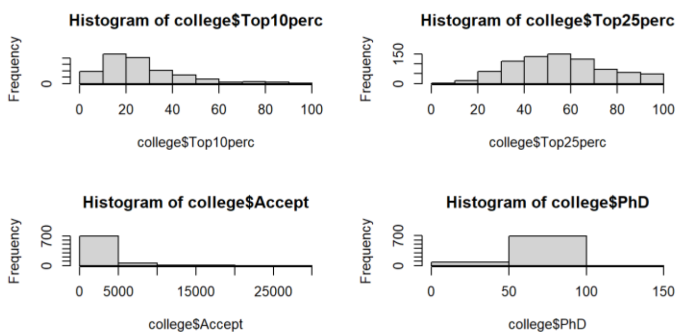
```
hist(college$Top10perc,breaks = 10)

hist(college$Top25perc, breaks = 7)

hist(college$Accept,breaks = 5)

hist(college$PhD,breaks = 3)
```

Output:



Problem 4:

4(a)

Code:

```
#Loading the data frame baseball in the plyr package

install.packages("plyr")

library(plyr)

#Getting information about the data set and definitions for the variables

?baseball

baseball
```

4(b)

Code

```
#Setting sf to 0 for players before 1954

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

#Setting missing values in hbp to 0

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

#Exclude all player records with fewer than 50 ab

baseball <- subset(baseball,subset=ab>=50)

baseball

4(c)

Making all the values to 0 using rep function and Computing on base percentage in the variable obp then adding the column to data frame

Code

```
obp<- rep(0, nrow(baseball))

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

baseball<-data.frame(baseball,obp)

baseball
```

4(d)

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

Code

```
b_obp<-baseball[order(-baseball$obp),]

b_obp

b_obp[1:5,c("year","id","obp")]
```

Output

```
      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 5:

5(a)

Loading the quakes data from the datasets package

Code

```
install.packages("datasets")

library(datasets)

quakes
```

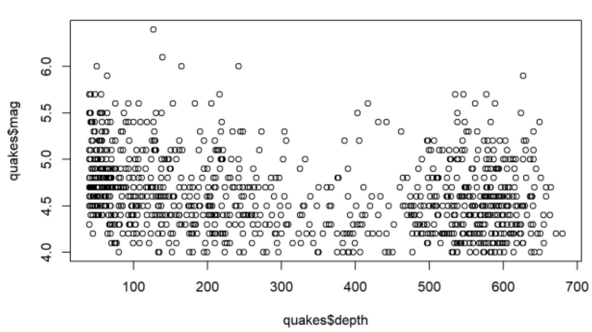
5(b)

Plotting the recorded earthquake magnitude against the earthquake depth

Code

```
plot(quakes$depth, quakes$mag,)
```

Output



5(c)

Using aggregate to compute the average earthquake depth for each magnitude level and storing these results in a new data frame named quakeAvgDepth

Code

```
quakeAvgDepth <- data.frame(aggregate(quakes$depth,by = list(quakes$mag),FUN= mean))
```

5(d)

Changing column name in quakeAvgDepth

Code

```
names(quakeAvgDepth)[names(quakeAvgDepth) == 'Group.1']  <-  'AgMag'
names(quakeAvgDepth)[names(quakeAvgDepth) == 'x'] <- ‘AgDep’
```

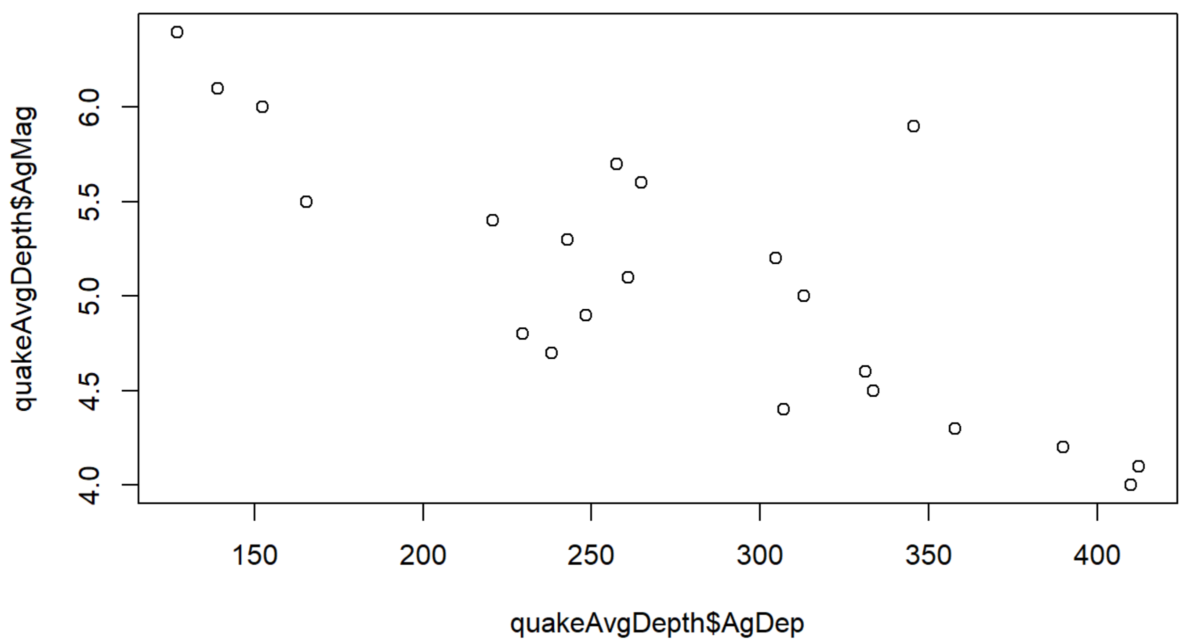
5(e)

Plotting between average depth to magnitude

Code

```
plot(quakeAvgDepth$AgDep, quakeAvgDepth$AgMag)
```

Output



5(f)

In the first plot high and low depths have more magnitude. In the second plot we can say that depth and magnitude were inversely proportional