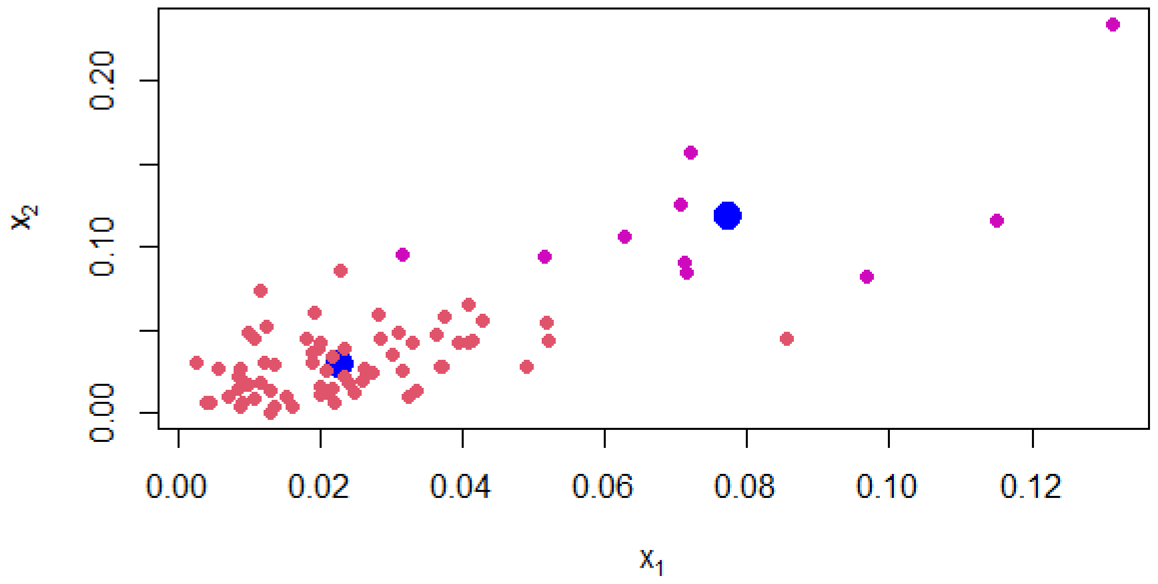
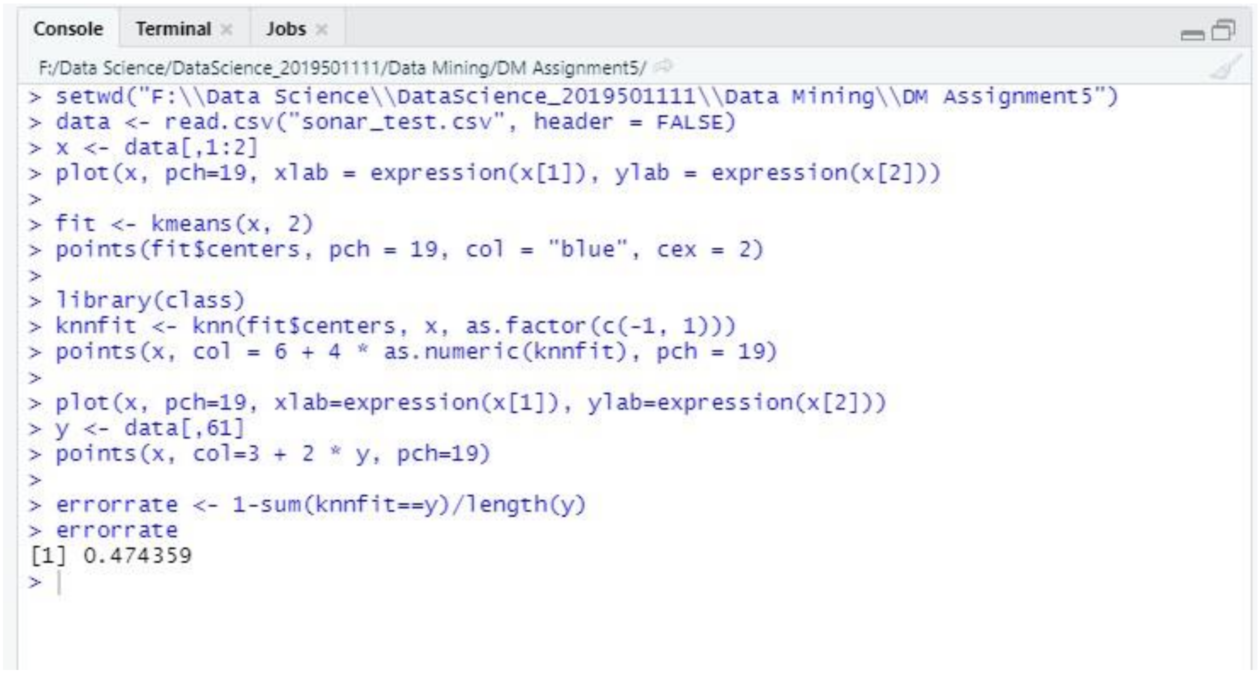
Data Mining Assignment 5

1) Read Chapter 8 (Sections 8.1 and 8.2) and Chapter 2 (Section 2.4).

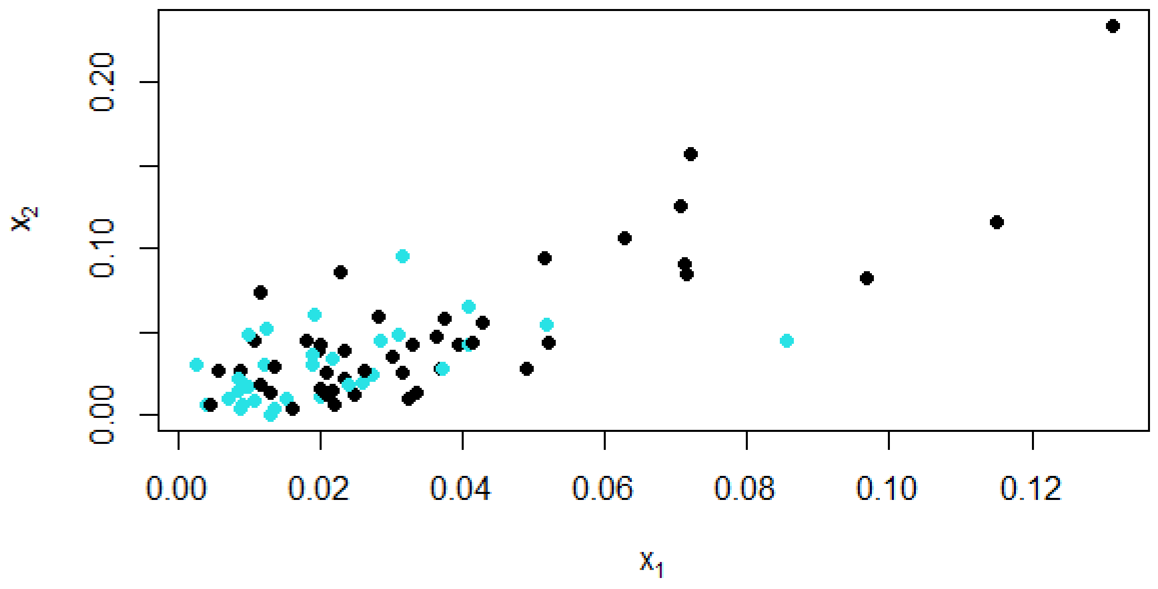
2) Use Kmeans() with all `the default values to find the k=2 solution for the first two columns of the sonar test data. Plot these two columns. Also plot the fitted cluster centers using a different color. Finally use the knn() function to assign the cluster membership for the points to the nearest cluster center. Color the points according to their cluster membership. Show your R commands for doing so.



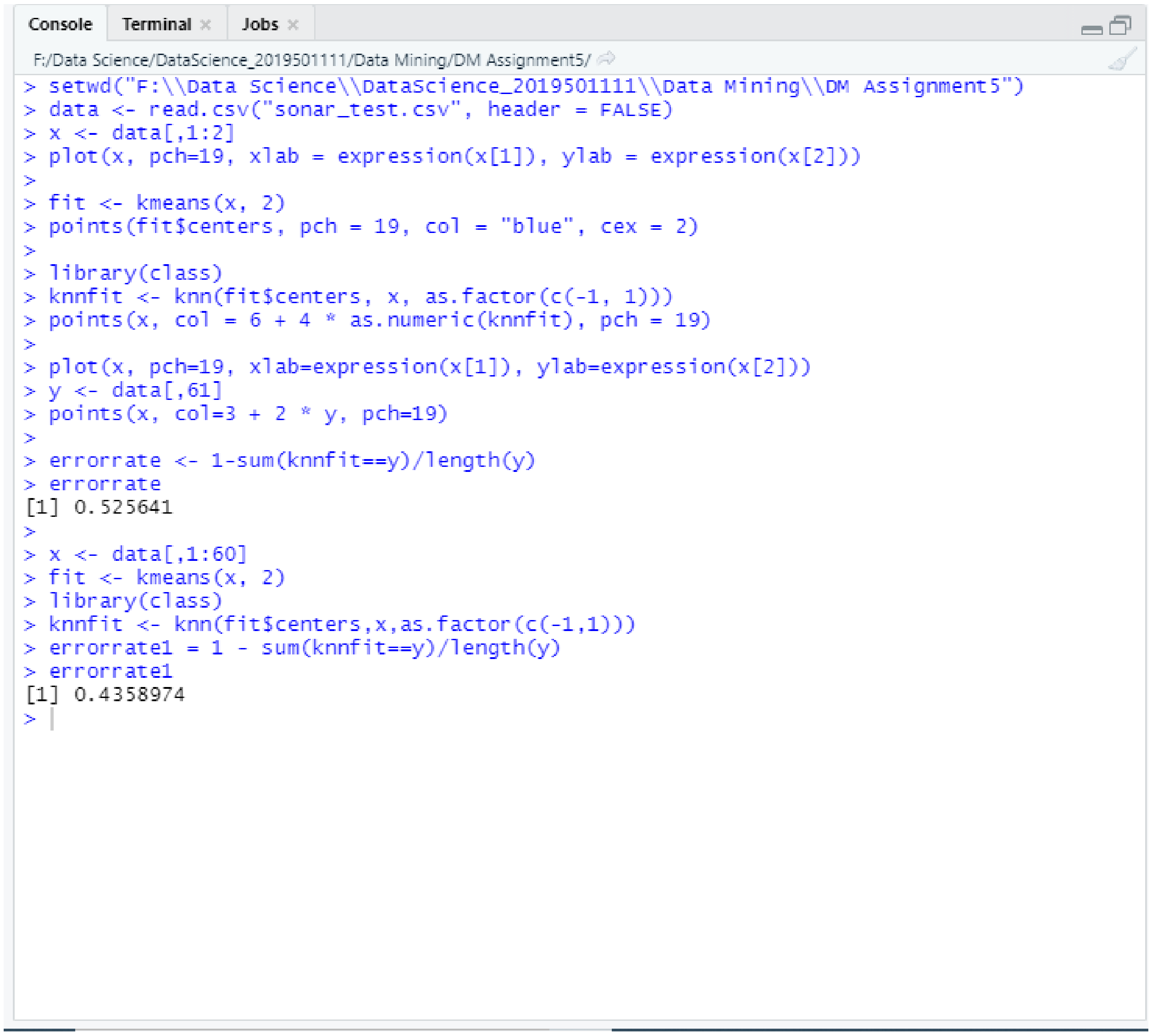
3) Graphically compare the cluster memberships from the previous problem to the actual labels in the test data. Also compute the misclassification error that would result if you used your clustering rule to classify the data. Show your R commands for doing so.


Console
Terminal
Jobs
F:/Data Science/DataScience_2019501111/Data Mining/DM Assignment5/

```
> setwd("F:\\Data Science\\DataScience_2019501111\\Data Mining\\DM Assignment5")
> data <- read.csv("sonar_test.csv", header = FALSE)
> x <- data[,1:2]
> plot(x, pch=19, xlab = expression(x[1]), ylab = expression(x[2]))
>
> fit <- kmeans(x, 2)
> points(fit$centers, pch = 19, col = "blue", cex = 2)
>
> library(class)
> knnfit <- knn(fit$centers, x, as.factor(c(-1, 1)))
> points(x, col = 6 + 4 * as.numeric(knnfit), pch = 19)
>
> plot(x, pch=19, xlab=expression(x[1]), ylab=expression(x[2]))
> y <- data[,61]
> points(x, col=3 + 2 * y, pch=19)
>
> errorrate <- 1-sum(knnfit==y)/length(y)
> errorrate
[1] 0.474359
> |
```



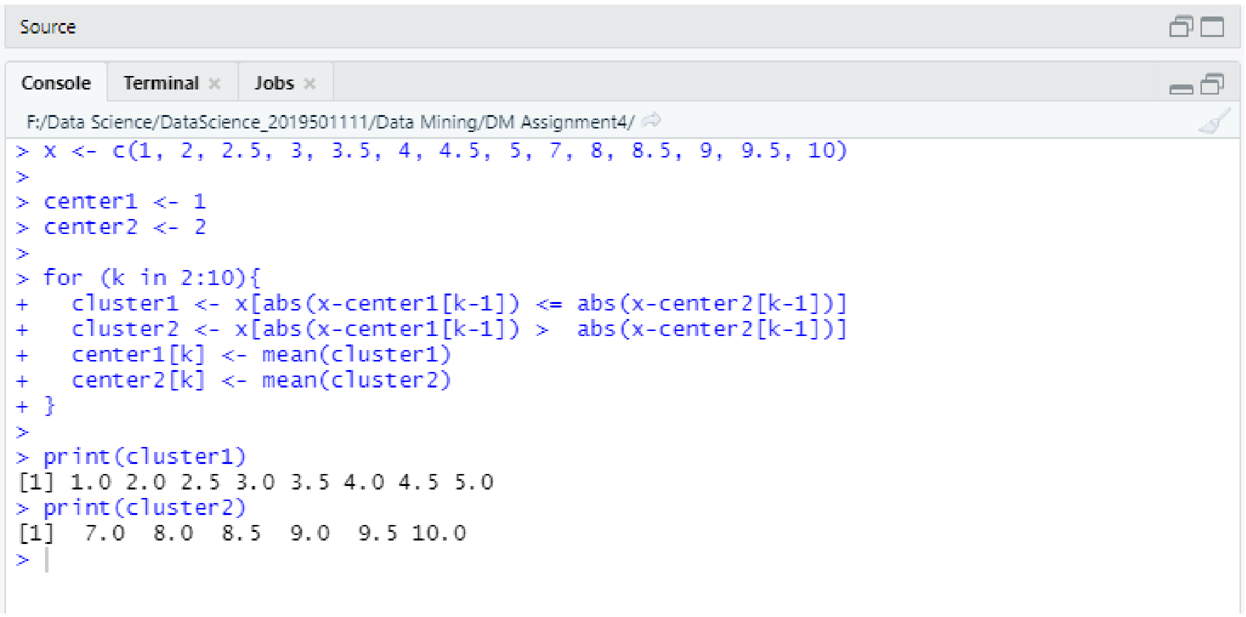
4) Repeat the previous problem using all 60 columns. Show your R commands for doing so.


Console Terminal Jobs

F:/Data Science/DataScience_2019501111/Data Mining/DM Assignment5/

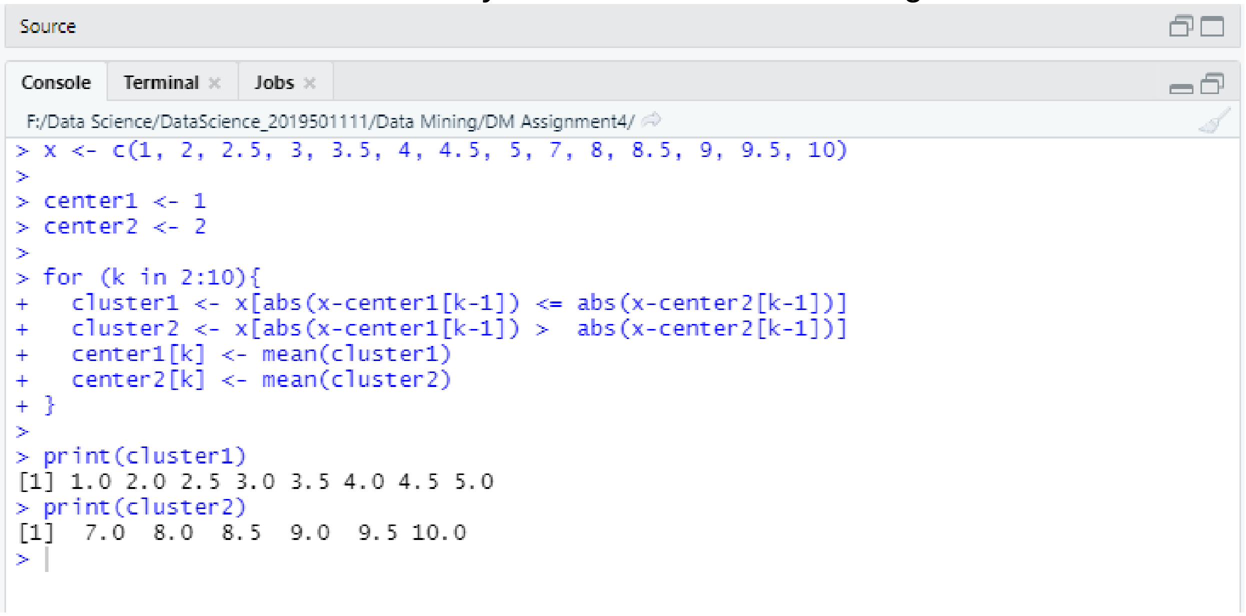
```
> setwd("F:\\Data Science\\DataScience_2019501111\\Data Mining\\DM Assignment5")
> data <- read.csv("sonar_test.csv", header = FALSE)
> x <- data[,1:2]
> plot(x, pch=19, xlab = expression(x[1]), ylab = expression(x[2]))
>
> fit <- kmeans(x, 2)
> points(fit$centers, pch = 19, col = "blue", cex = 2)
>
> library(class)
> knnfit <- knn(fit$centers, x, as.factor(c(-1, 1)))
> points(x, col = 6 + 4 * as.numeric(knnfit), pch = 19)
>
> plot(x, pch=19, xlab=expression(x[1]), ylab=expression(x[2]))
> y <- data[,61]
> points(x, col=3 + 2 * y, pch=19)
>
> errorrate <- 1-sum(knnfit==y)/length(y)
> errorrate
[1] 0.525641
>
> x <- data[,1:60]
> fit <- kmeans(x, 2)
> library(class)
> knnfit <- knn(fit$centers,x,as.factor(c(-1,1)))
> errorrate1 = 1 - sum(knnfit==y)/length(y)
> errorrate1
[1] 0.4358974
>
```

5) Consider the one dimensional data set given x←c(1,2,2.5,3,3.5,4,4.5,5,7,8,8.5,9,9.5,10). Starting with initial cluster center values of 1 and 2 carry out algorithm 10 until convergence by hand for k=2 clusters. Show all your work for each step and be sure to say specifically which points are in each cluster at each step.


Source
Console
Terminal
Jobs
F:/Data Science/DataScience_2019501111/Data Mining/DM Assignment4/

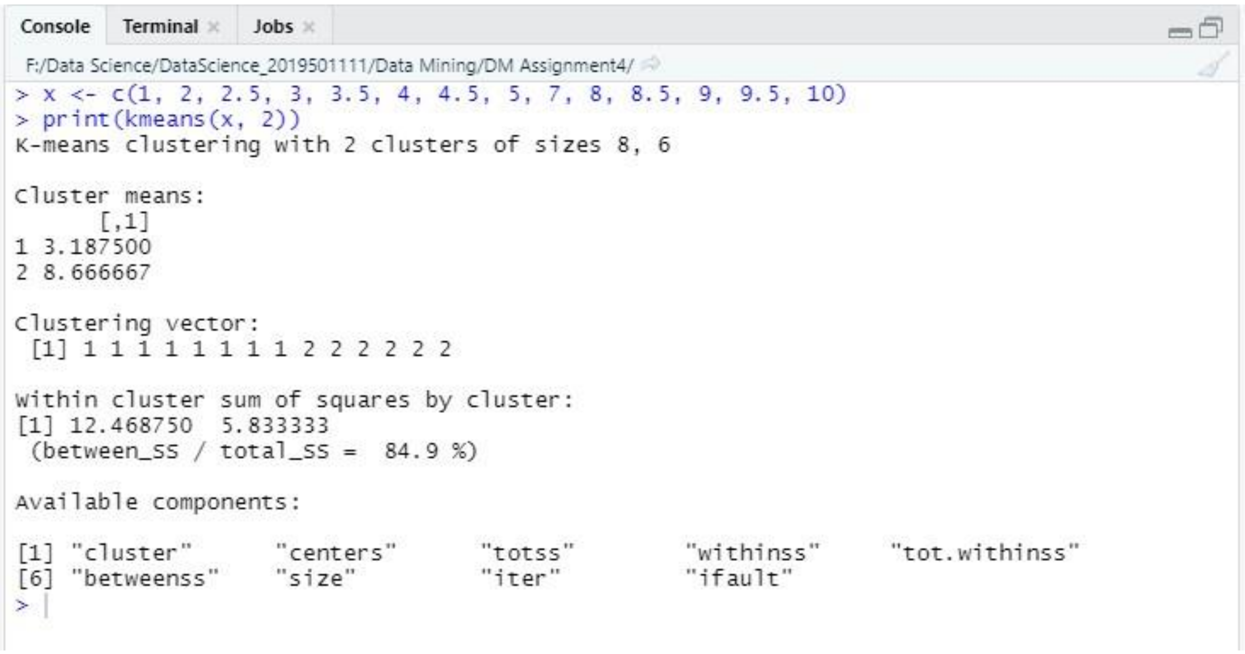
```
> x <- c(1, 2, 2.5, 3, 3.5, 4, 4.5, 5, 7, 8, 8.5, 9, 9.5, 10)
>
> center1 <- 1
> center2 <- 2
>
> for (k in 2:10){
+   cluster1 <- x[abs(x-center1[k-1]) <= abs(x-center2[k-1])]
+   cluster2 <- x[abs(x-center1[k-1]) >  abs(x-center2[k-1])]
+   center1[k] <- mean(cluster1)
+   center2[k] <- mean(cluster2)
+ }
>
> print(cluster1)
[1] 1.0 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> print(cluster2)
[1]  7.0  8.0  8.5  9.0  9.5 10.0
> 
```

6) Repeat the previous problem by writing a loop and verify that the final answer is the same and show your R commands for doing so.


Source
Console
Terminal
Jobs
F:/Data Science/DataScience_2019501111/Data Mining/DM Assignment4/

```
> x <- c(1, 2, 2.5, 3, 3.5, 4, 4.5, 5, 7, 8, 8.5, 9, 9.5, 10)
>
> center1 <- 1
> center2 <- 2
>
> for (k in 2:10){
+   cluster1 <- x[abs(x-center1[k-1]) <= abs(x-center2[k-1])]
+   cluster2 <- x[abs(x-center1[k-1]) >  abs(x-center2[k-1])]
+   center1[k] <- mean(cluster1)
+   center2[k] <- mean(cluster2)
+ }
>
> print(cluster1)
[1] 1.0 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> print(cluster2)
[1]  7.0  8.0  8.5  9.0  9.5 10.0
> 
```

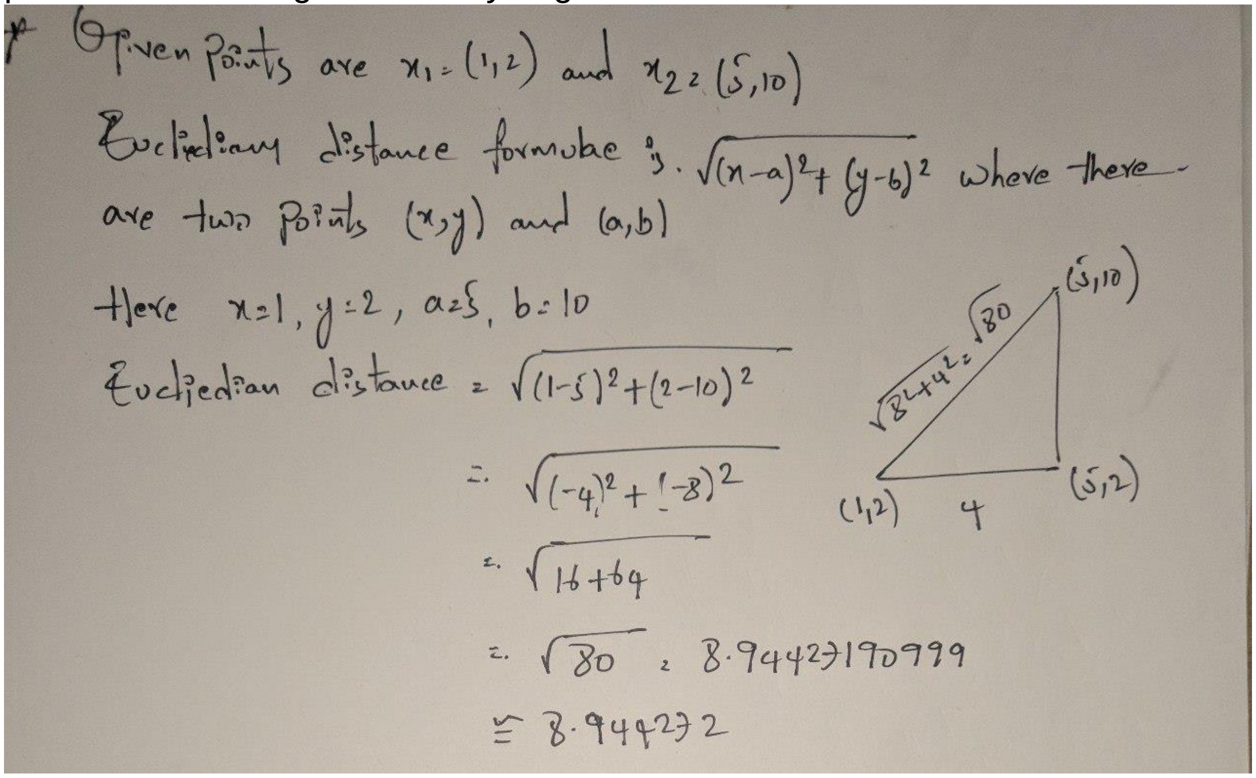
7) Verify that the kmeans function gives the same solution for the previous problem when you use all of the default values and show your R commands for doing so.


Console
Terminal
Jobs
F:/Data Science/DataScience_2019501111/Data Mining/DM Assignment4/

```
> x <- c(1, 2, 2.5, 3, 3.5, 4, 4.5, 5, 7, 8, 8.5, 9, 9.5, 10)
> print(kmeans(x, 2))
K-means clustering with 2 clusters of sizes 8, 6

Cluster means:
      [,1]
1 3.187500
2 8.666667

Clustering vector:
 [1] 1 1 1 1 1 1 1 1 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1] 12.468750  5.833333
 (between_SS / total_SS =  84.9 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
> |
```

8) Consider the points x1<-c(1,2) and x2<-c(5,10).

a) Compute the (Euclidean) distance by hand. Show your work and include a picture of the triangle for the Pythagorean Theorem.


(5,10)
$\sqrt{8^2+4^2} = \sqrt{80}$
(1,2)
4
(5,2)

* Given Points are $x_1 = (1,2)$ and $x_2 = (5,10)$

Euclidiany distance formube is. $\sqrt{(x-a)^2+(y-b)^2}$ where there are two Points (x,y) and (a,b)

Here $x=1, y=2, a=5, b=10$

Euclidian distance $= \sqrt{(1-5)^2+(2-10)^2}$

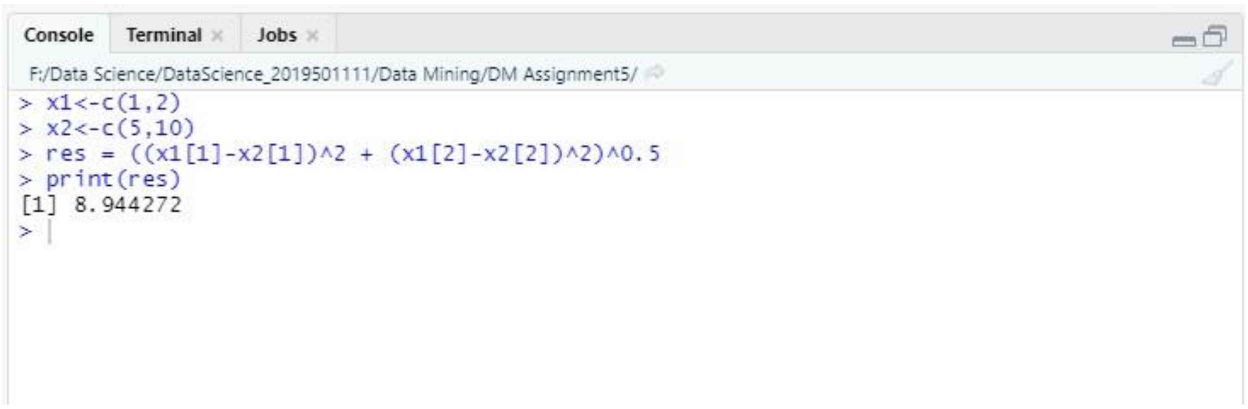
$= \sqrt{(-4)^2+(-8)^2}$

$= \sqrt{16+64}$

$= \sqrt{80} = 8.94427190999$

$\cong 8.944272$

b) Verify that the dist function in R gives the same value as you got in part a. Show your R commands for doing so.



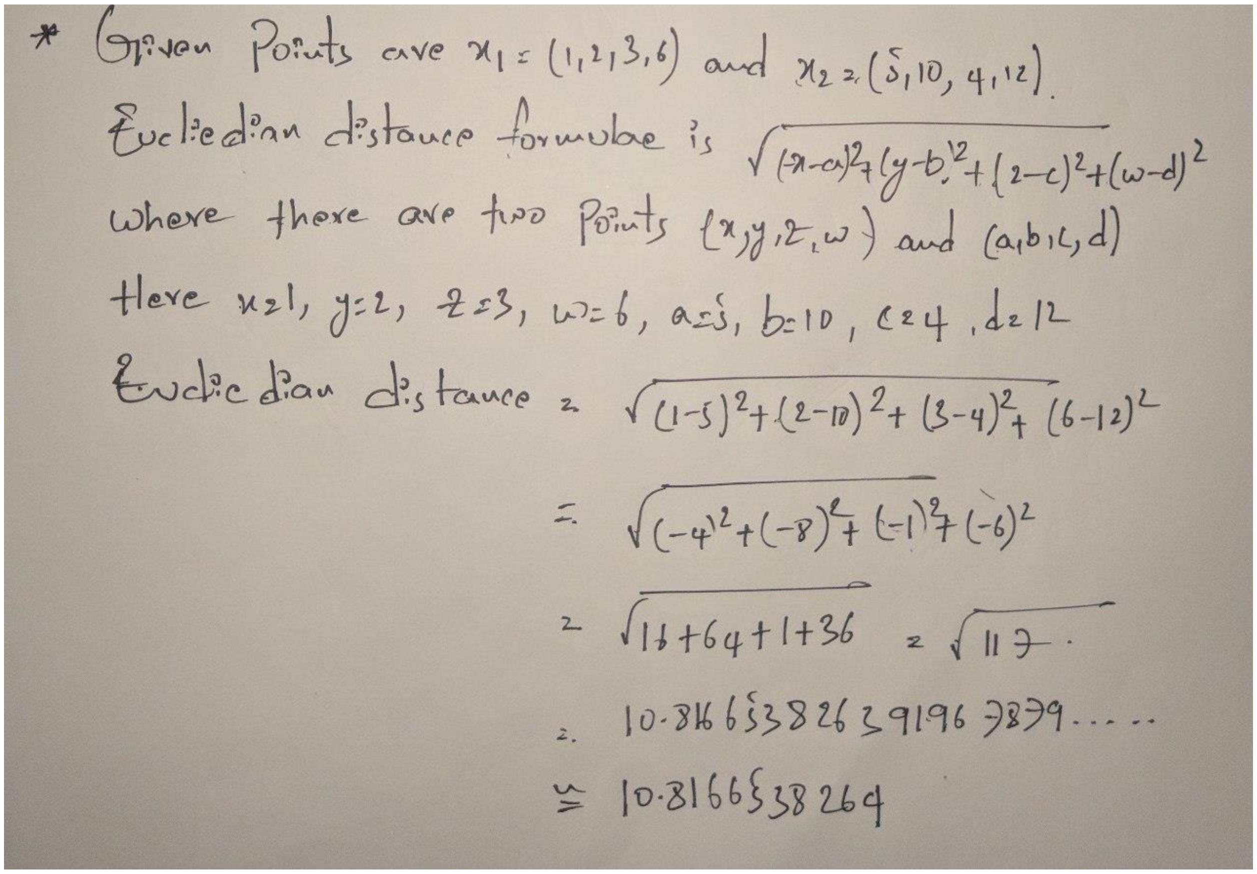
Console Terminal Jobs

F:/Data Science/DataScience_2019501111/Data Mining/DM Assignment5/

```
> x1<-c(1,2)
> x2<-c(5,10)
> res = ((x1[1]-x2[1])^2 + (x1[2]-x2[2])^2)^0.5
> print(res)
[1] 8.944272
> |
```

9) Consider the points x1<-c(1,2,3,6) and x2<-c(5,10,4,12).

a) Compute the (Euclidean) distance by hand. Show your work.



* Given Points are $x_1 = (1,2,3,6)$ and $x_2 = (5,10,4,12)$.

Euclidean distance formulae is $\sqrt{(x-a)^2+(y-b)^2+(z-c)^2+(w-d)^2}$

where there are two Points (x,y,z,w) and (a,b,c,d)

Here $x=1$, $y=2$, $z=3$, $w=6$, $a=5$, $b=10$, $c=4$, $d=12$

Euclidean distance $= \sqrt{(1-5)^2+(2-10)^2+(3-4)^2+(6-12)^2}$

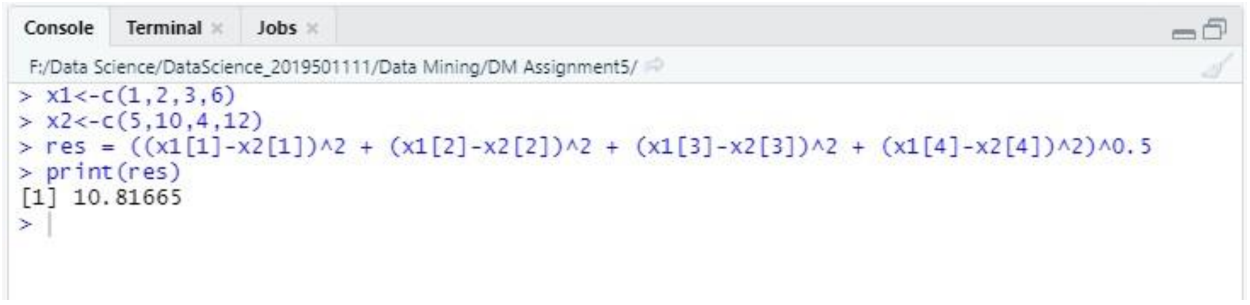
$= \sqrt{(-4)^2+(-8)^2+(-1)^2+(-6)^2}$

$= \sqrt{16+64+1+36} = \sqrt{117}$.

$= 10.816653826391967879.....$

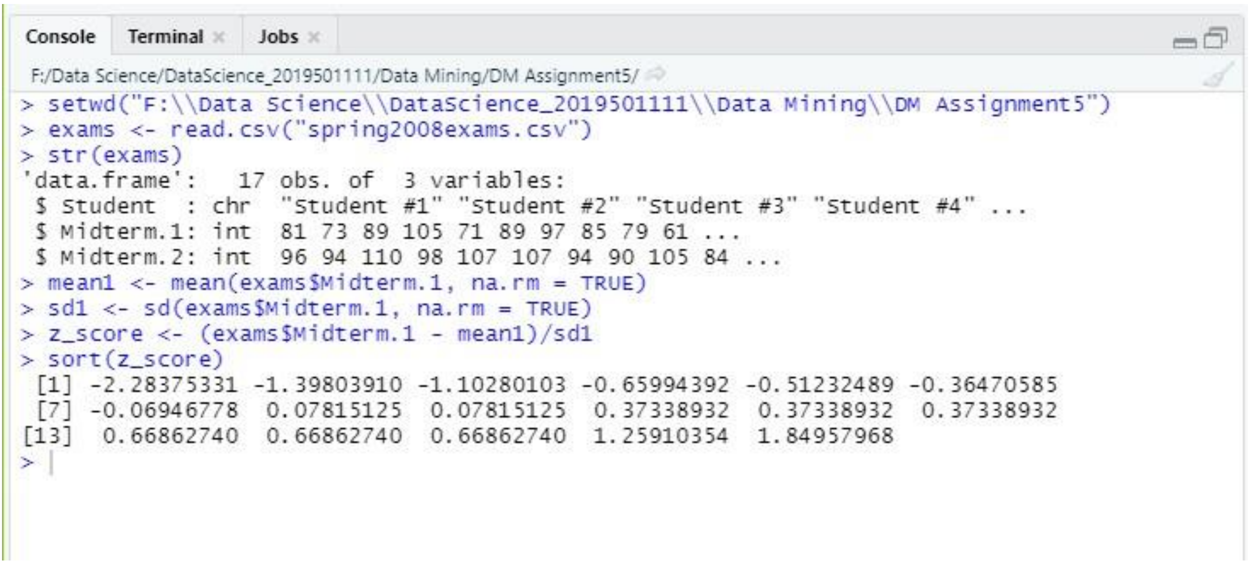
$\simeq 10.8166538264$

b) Verify that the dist function in R gives the same value as you got in part a. Show your R commands for doing so.

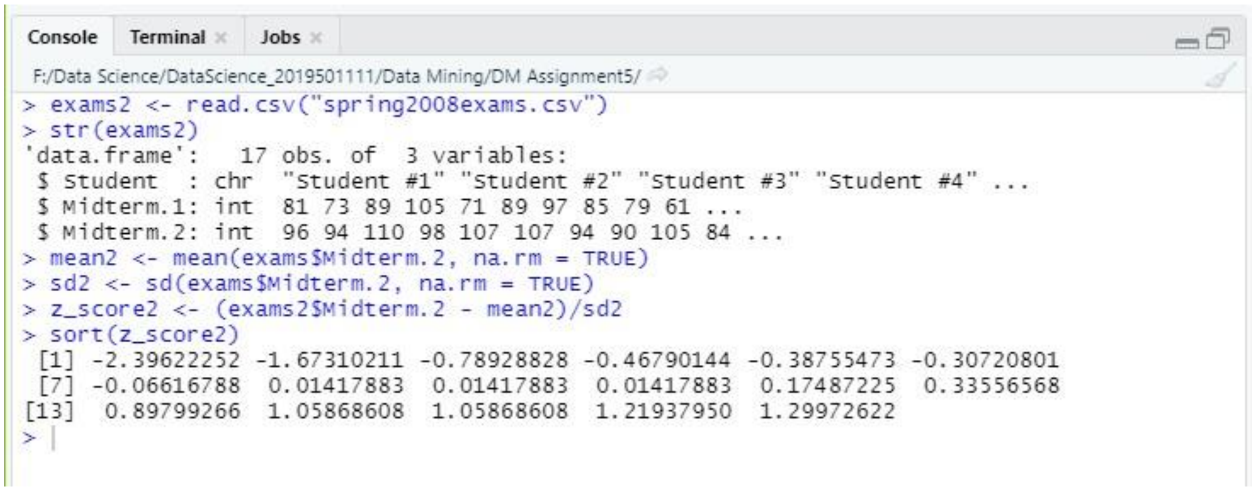

```
> x1<-c(1,2,3,6)
> x2<-c(5,10,4,12)
> res = ((x1[1]-x2[1])^2 + (x1[2]-x2[2])^2 + (x1[3]-x2[3])^2 + (x1[4]-x2[4])^2)^0.5
> print(res)
[1] 10.81665
> 
```

10) Read Chapter 10.

11) Use a z score cut off of 3 to identify any outliers using the grades for the first midterm at www.stats202.com/spring2008exams.csv. Are there any outliers according to the z=+/-3 rule? What is the value of the largest z score and what is the value of the smallest (most negative) z score? Show your R commands.


```
> setwd("F:\\Data Science\\DataScience_2019501111\\Data Mining\\DM Assignment5")
> exams <- read.csv("spring2008exams.csv")
> str(exams)
'data.frame':	17 obs. of  3 variables:
 $ Student  : chr  "Student #1" "Student #2" "Student #3" "Student #4" ...
 $ Midterm.1: int  81 73 89 105 71 89 97 85 79 61 ...
 $ Midterm.2: int  96 94 110 98 107 107 94 90 105 84 ...
> mean1 <- mean(exams$Midterm.1, na.rm = TRUE)
> sd1 <- sd(exams$Midterm.1, na.rm = TRUE)
> z_score <- (exams$Midterm.1 - mean1)/sd1
> sort(z_score)
 [1] -2.28375331 -1.39803910 -1.10280103 -0.65994392 -0.51232489 -0.36470585
 [7] -0.06946778  0.07815125  0.07815125  0.37338932  0.37338932  0.37338932
[13]  0.66862740  0.66862740  0.66862740  1.25910354  1.84957968
> 
```

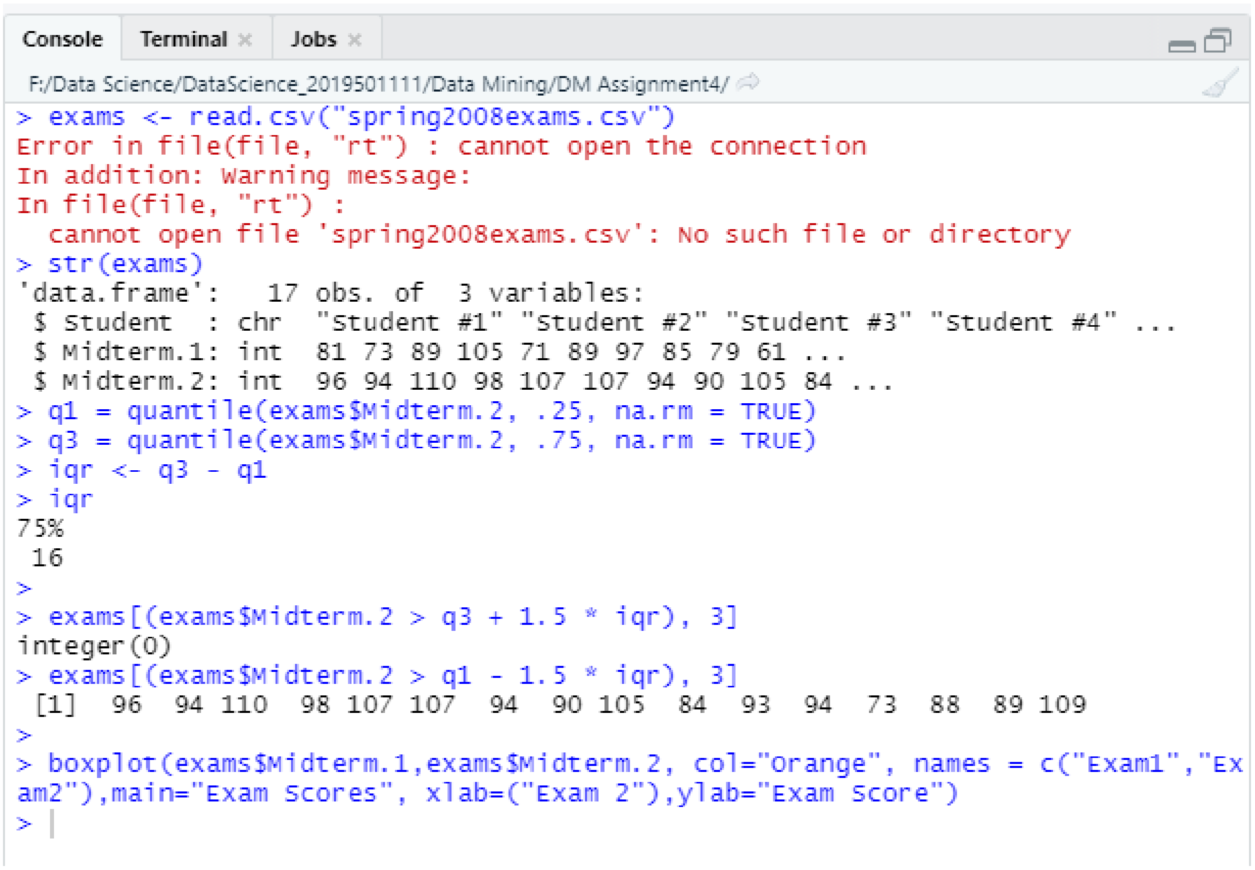
12) Use a z score cut off of 3 to identify any outliers using the grades for the second midterm at www.stats202.com/spring2008exams.csv. Are there any outliers according to the z=+/-3 rule? What is the value of the largest z score and what is the value of the smallest (most negative) z score? Show your R commands.


Console
Terminal
Jobs
F:/Data Science/DataScience_2019501111/Data Mining/DM Assignment5/

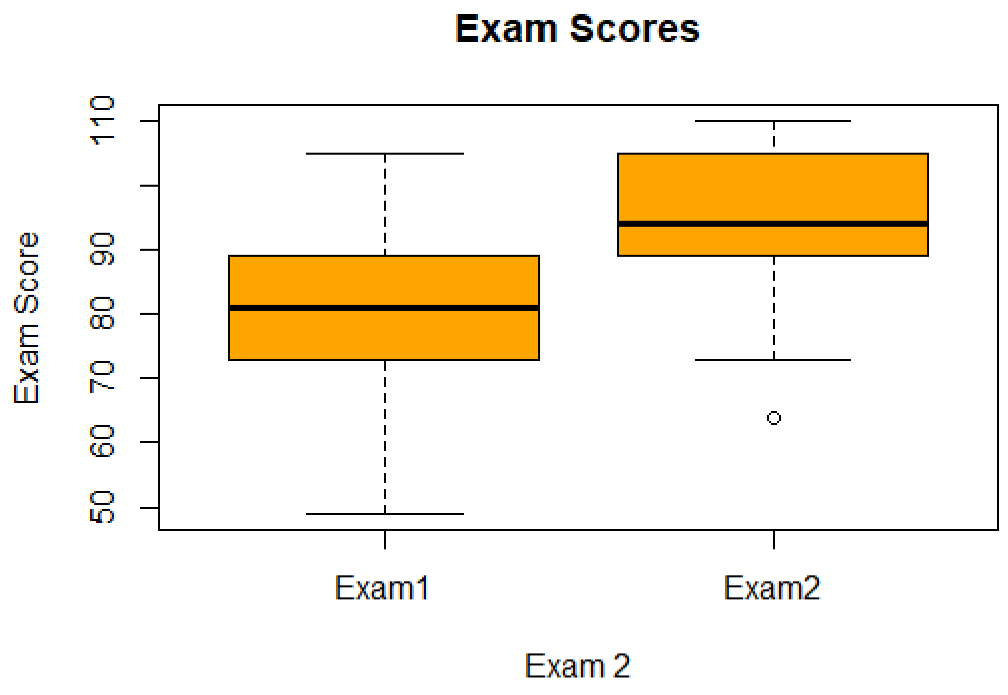
```
> exams2 <- read.csv("spring2008exams.csv")
> str(exams2)
'data.frame':   17 obs. of  3 variables:
 $ Student  : chr  "Student #1" "Student #2" "Student #3" "Student #4" ...
 $ Midterm.1: int  81 73 89 105 71 89 97 85 79 61 ...
 $ Midterm.2: int  96 94 110 98 107 107 94 90 105 84 ...
> mean2 <- mean(exams$Midterm.2, na.rm = TRUE)
> sd2 <- sd(exams$Midterm.2, na.rm = TRUE)
> z_score2 <- (exams2$Midterm.2 - mean2)/sd2
> sort(z_score2)
 [1] -2.39622252 -1.67310211 -0.78928828 -0.46790144 -0.38755473 -0.30720801
 [7] -0.06616788  0.01417883  0.01417883  0.01417883  0.17487225  0.33556568
[13]  0.89799266  1.05868608  1.05868608  1.21937950  1.29972622
> |
```

13) Compute the count of each ip address (1st column) in the data stats202log.txt, then use a z score cut off of 3 to identify any outliers for these counts using Excel for the user agent column of the data at www.stats202.com/stats202log.txt. (The user agent column is the second to last column and the value for it in the first row is "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322)"). What user agents are identified as outliers using the z=+/-3 rule on the counts of the user agents? What are the z scores for these outliers? (You do not need to show any work for this problem because you are using Excel.)

14) Identify any outliers more than 1.5 IQR's above the 3rd quartile or below the 1st quartile. Verify that these are the same outliers found by the boxplot function using the grades for the second midterm at www.stats202.com/spring2008exams.csv. Show your R commands and include the boxplot. Are any of the grades for the second midterm outliers by this rule? If so, which ones?

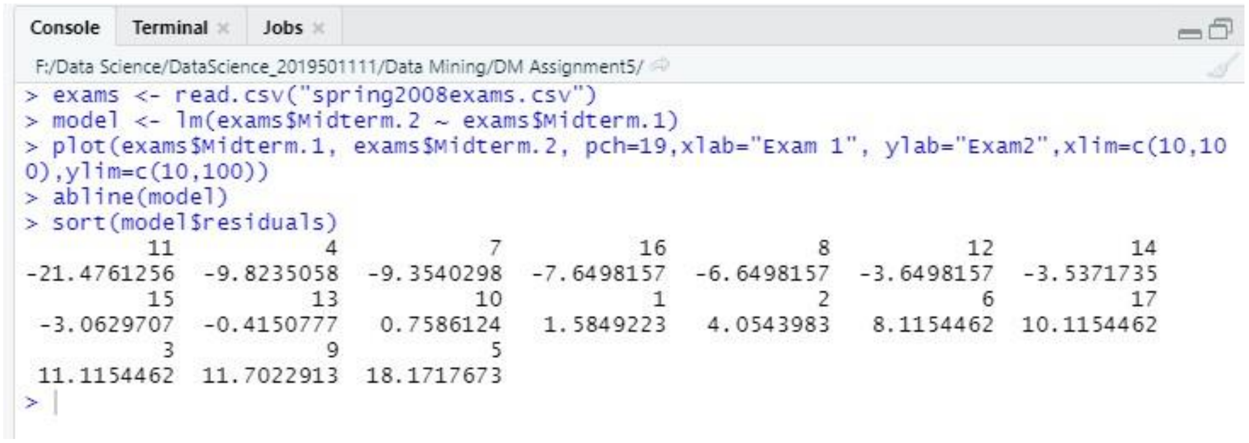

Console
Terminal
Jobs
F:/Data Science/DataScience_201950111/Data Mining/DM Assignment4/

```
> exams <- read.csv("spring2008exams.csv")
Error in file(file, "rt") : cannot open the connection
In addition: Warning message:
In file(file, "rt") :
  cannot open file 'spring2008exams.csv': No such file or directory
> str(exams)
'data.frame':	17 obs. of  3 variables:
 $ Student  : chr  "Student #1" "Student #2" "Student #3" "Student #4" ...
 $ Midterm.1: int  81 73 89 105 71 89 97 85 79 61 ...
 $ Midterm.2: int  96 94 110 98 107 107 94 90 105 84 ...
> q1 = quantile(exams$Midterm.2, .25, na.rm = TRUE)
> q3 = quantile(exams$Midterm.2, .75, na.rm = TRUE)
> iqr <- q3 - q1
> iqr
75%
 16
>
> exams[(exams$Midterm.2 > q3 + 1.5 * iqr), 3]
integer(0)
> exams[(exams$Midterm.2 > q1 - 1.5 * iqr), 3]
 [1]  96  94 110  98 107 107  94  90 105  84  93  94  73  88  89 109
>
> boxplot(exams$Midterm.1,exams$Midterm.2, col="Orange", names = c("Exam1","Ex
am2"),main="Exam Scores", xlab=("Exam 2"),ylab="Exam Score")
> |
```



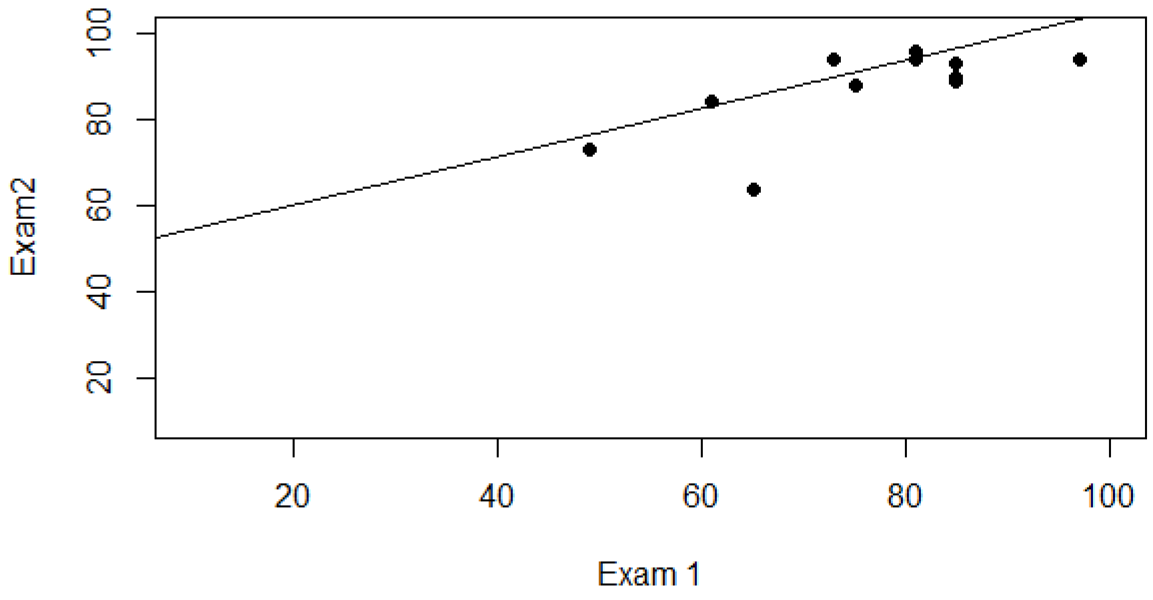
15) Use functions to fit a least squares regression model which predicts the

exam 2 score as a function of the exam 1 score for the data spring2008exams.csv. Plot the fitted line and determine for which points the fitted exam 2 values are the furthest from the actual values using the model residuals using the midterm grades at www.stats202.com/spring2008exams.csv. Be sure to include the plot. Which student # had the largest POSITIVE residual? Show your R commands.



F:/Data Science/DataScience_2019501111/Data Mining/DM Assignment5/

```
> exams <- read.csv("spring2008exams.csv")
> model <- lm(exams$Midterm.2 ~ exams$Midterm.1)
> plot(exams$Midterm.1, exams$Midterm.2, pch=19,xlab="Exam 1", ylab="Exam2",xlim=c(10,10
0),ylim=c(10,100))
> abline(model)
> sort(model$residuals)
         11           4           7          16           8          12          14 
-21.4761256  -9.8235058  -9.3540298  -7.6498157  -6.6498157  -3.6498157  -3.5371735 
         15          13          10           1           2           6          17 
 -3.0629707  -0.4150777   0.7586124   1.5849223   4.0543983   8.1154462  10.1154462 
          3           9           5 
 11.1154462  11.7022913  18.1717673 
> |
```


Exam2
Exam 1