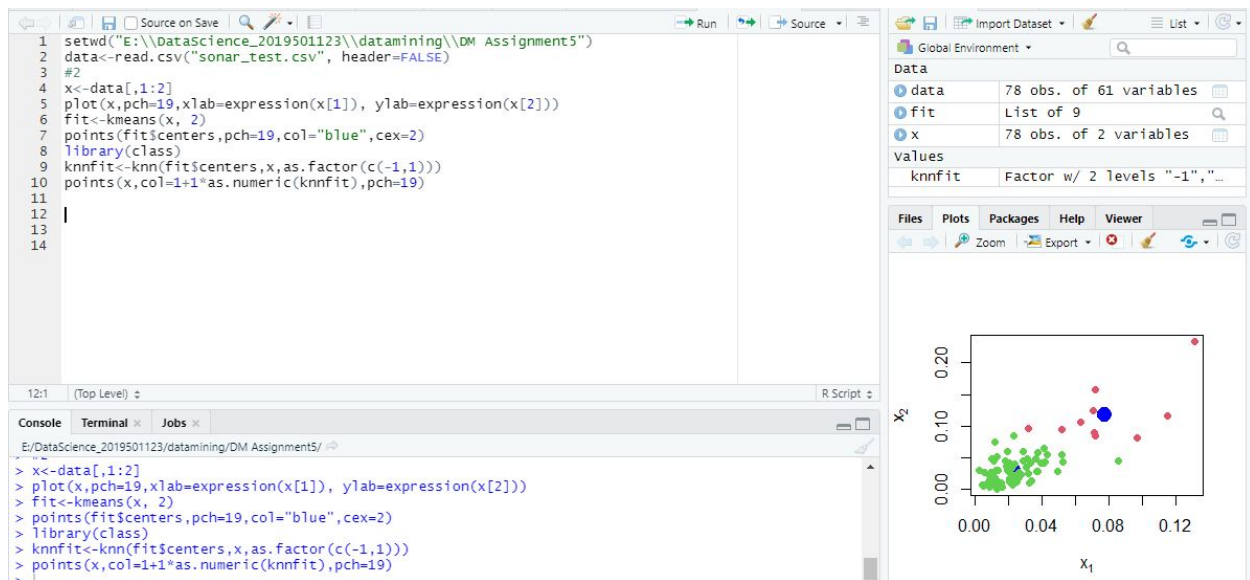


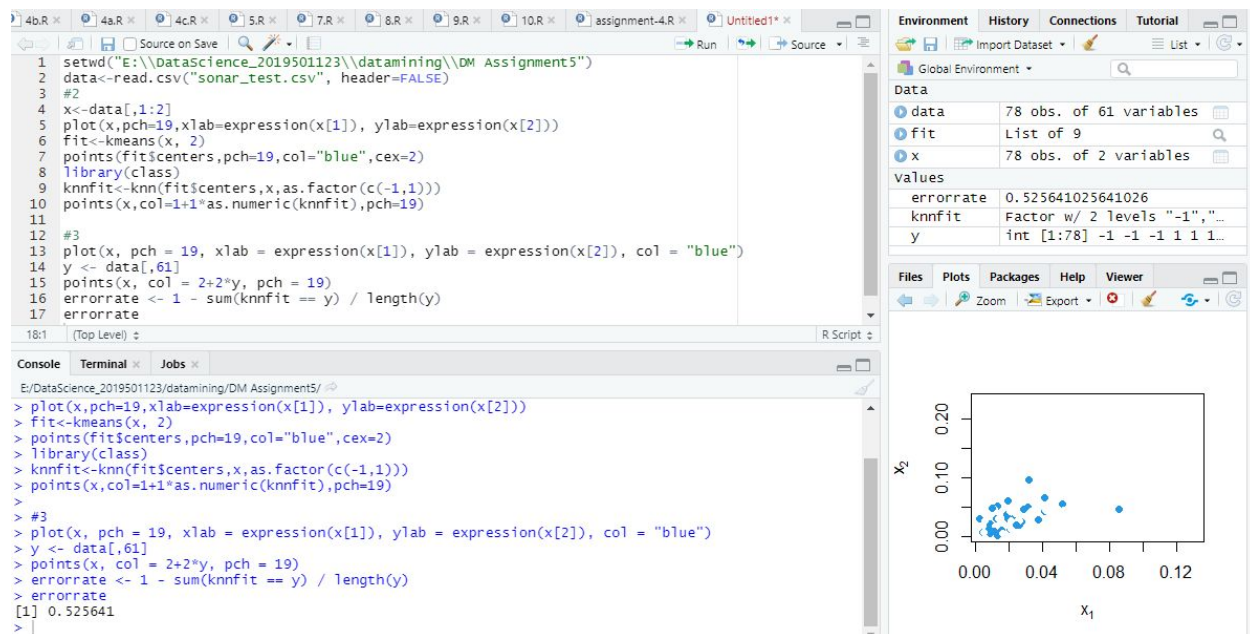
Data Mining Assignment 5

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

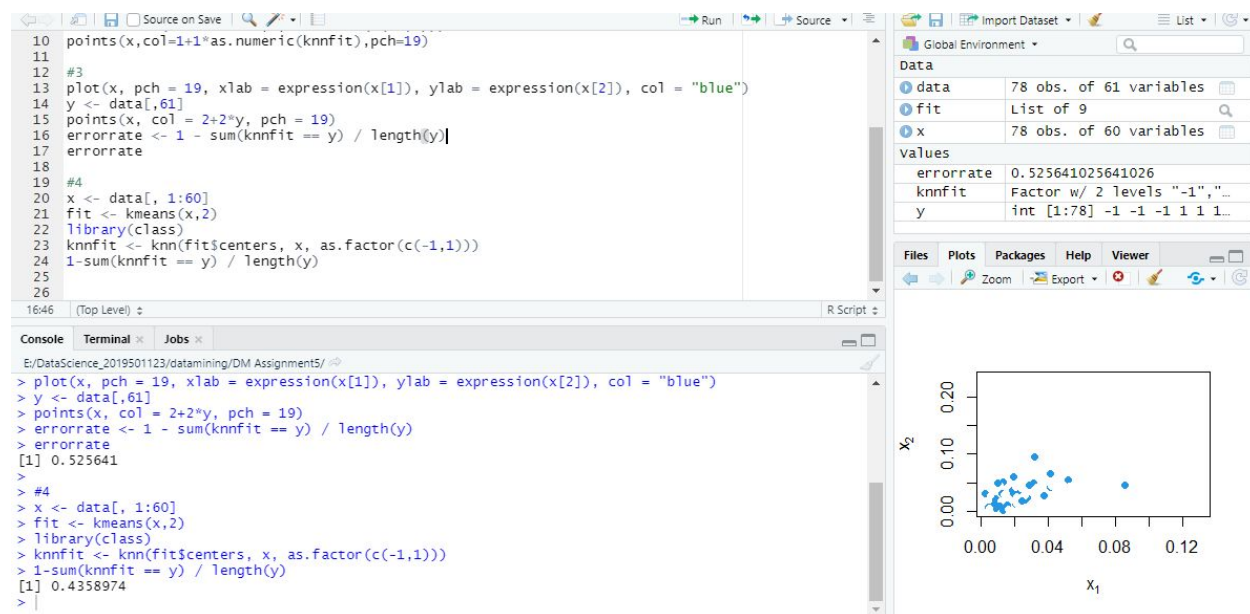
2. Use Kmeans() with all the default values to find the k=2 solution for the first two columns of the sonar test data. Plot these two columns. Also plot the fitted cluster centers using a different color. Finally use the knn() function to assign the cluster membership for the points to the nearest cluster 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.



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



5. Consider the one dimensional data set given $x \leftarrow c(1, 2, 2.5, 3, 3.5, 4, 4.5, 5, 7, 8, 8.5, 9, 9.5, 10)$. 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.

The screenshot shows the RStudio environment with the following components:

- Script Editor:** Contains R code for K-means clustering. The code defines a vector `x` with 10 values, initializes two centers, and iteratively updates them based on the nearest neighbor assignment.
- Environment Pane:** Displays the current state of variables:

Variable	Class	Value
center1	num	[1:10] 1 1 2.12 2.93 ...
center2	num	[1:10] 2 5.88 6.9 8.1...
cluster1	num	[1:8] 1 2 2.5 3 3.5 4...
cluster2	num	[1:6] 7 8 8.5 9 9.5 10
k	10L	
x	num	[1:14] 1 2 2.5 3 3.5 ...
- Console:** Shows the execution output of the script, including the final cluster assignments and the updated center values.

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

```
20 x <- data[, 1:60]
21 fit <- kmeans(x,2)
22 library(class)
23 knnfit <- knn(fit$centers, x, as.factor(c(-1,1)))
24 1-sum(knnfit == y) / length(y)
25
26 #5
27 x <- c(1,2,2.5,3,3.5,4,4.5,5,5,7,8,8.5,9,9.5,10)
28 center1 <- 1
29 center2 <- 2
30 for (k in 2:10){
31   cluster1 <- x[abs(x - center1[k-1]) <= abs(x - center2[k-1])]
32   cluster2 <- x[abs(x - center1[k-1]) > abs(x - center2[k-1])]
33   center1[k] <- mean(cluster1)
34   center2[k] <- mean(cluster2)
35 }
36 print(cluster1)
37 print(cluster2)
38
39
```

38:1 (Top Level) R Script

Console Terminal Jobs

E:/DataScience_2019501123/datamining/DM Assignment5/

```
> setwd("E:\\DataScience_2019501123\\datamining\\DM Assignment5")
> #5
> x <- c(1,2,2.5,3,3.5,4,4.5,5,5,7,8,8.5,9,9.5,10)
> 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.

```
38
39 #7
40 km <- kmeans(x,2)
41 print(km)
42 |
```

42:1 (Top Level) ↕ R Scr

Console Terminal × Jobs ×

E:/DataScience_2019501123/datamining/DM Assignment5/ ↗

```
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 %)
```

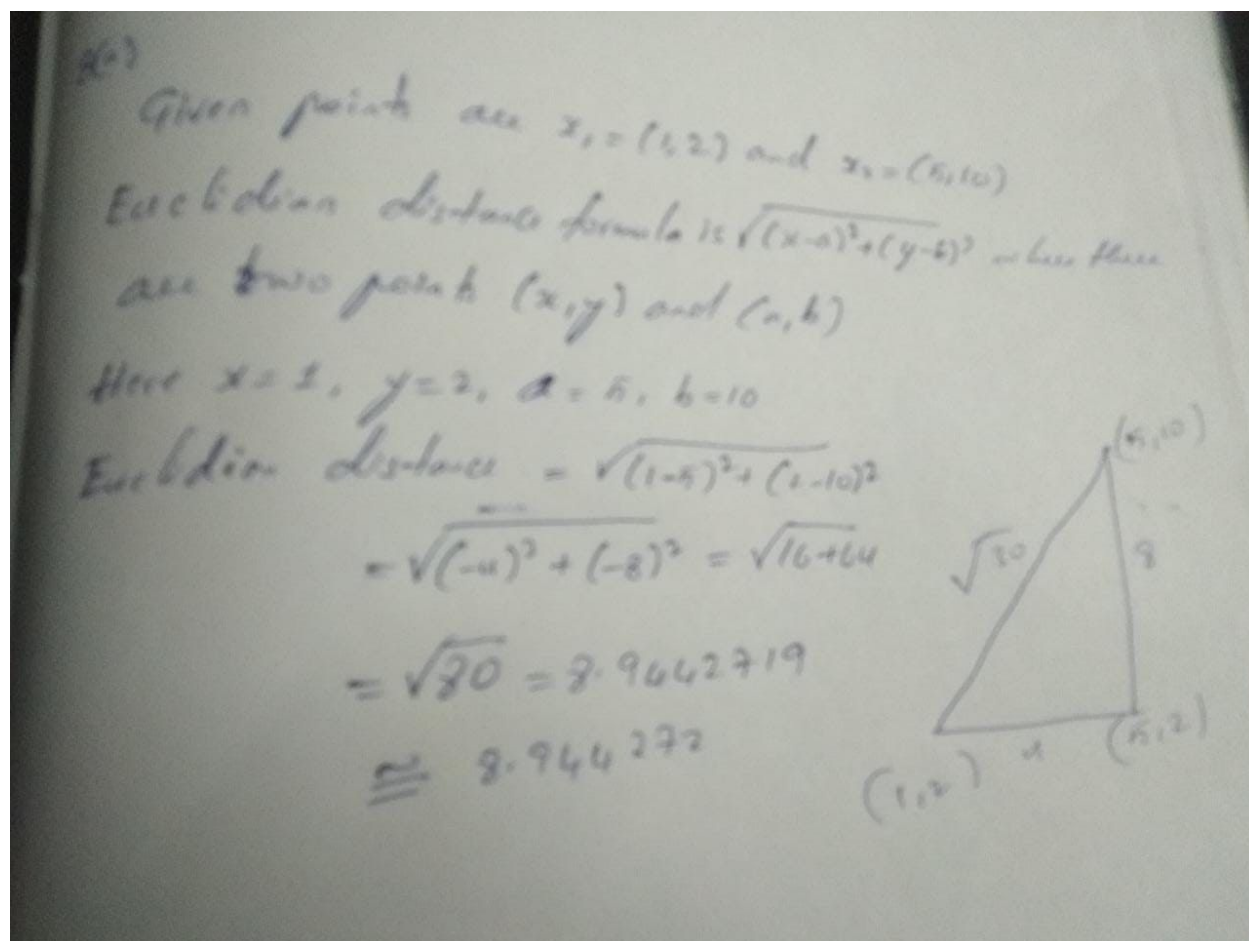
Available components:

```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
```

> |

8) Consider the points $x_1 \leftarrow c(1,2)$ and $x_2 \leftarrow c(5,10)$.

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



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.

The screenshot shows the RStudio environment with the following components:

- Source Editor:** Contains R code for a clustering algorithm and a distance calculation.

```
26 #5
27 x <- c(1,2,2.5,3,3.5,4,4.5,5,7,8,8.5,9,9.5,10)
28 center1 <- 1
29 center2 <- 2
30 for (k in 2:10){
31   cluster1 <- x[abs(x - center1[k-1]) <= abs(x - center2[k-1])]
32   cluster2 <- x[abs(x - center1[k-1]) > abs(x - center2[k-1])]
33   center1[k] <- mean(cluster1)
34   center2[k] <- mean(cluster2)
35 }
36 print(cluster1)
37 print(cluster2)
38
39 #7
40 km <- kmeans(x,2)
41 print(km)
42
43 #8(b)
44 x1 <- c(1,2)
45 x2 <- c(5,10)
46 res = ((x1[1] - x2[1]) ^ 2 + (x1[2] - x2[2]) ^ 2) ^ 0.5
47 print(res)
48
```
- Global Environment:** Displays the current values of variables.

Values	
res	8.944271909999916
x1	num [1:2] 1 2
x2	num [1:2] 5 10
- Console:** Shows the execution output.

```
E:/DataScience_2019501123/datamining/DM Assignment5/
> setwd("E:\\DataScience_2019501123\\datamining\\DM Assignment5")
> #8(b)
> 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 $x1 \leftarrow c(1,2,3,6)$ and $x2 \leftarrow c(5,10,4,12)$.

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

(a)
 → Given points are $x_1 = (1, 2, 3, 6)$ and $x_2 = (5, 10, 4, 12)$
 Euclidean distance formula 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 = $\sqrt{(1-5)^2 + (2-10)^2 + (3-4)^2 + (6-12)^2}$
 $= \sqrt{(-4)^2 + (-8)^2 + (-1)^2 + (-6)^2}$
 $= \sqrt{16 + 64 + 1 + 36} = \sqrt{117} = 10.816653826391$
 ≈ 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.


```

33 center1[k] <- mean(cluster1)
34 center2[k] <- mean(cluster2)
35 }
36 print(cluster1)
37 print(cluster2)
38
39 #7
40 km <- kmeans(x,2)
41 print(km)
42
43 #8(b)
44 x1 <- c(1,2)
45 x2 <- c(5,10)
46 res = ((x1[1] - x2[1]) ^ 2 + ) ^ 0.5
47 print(res)
48
49 #9(b)
50 x1 <-c(1,2,3,6)
51 x2 <-c(5,10,4,12)
52 dist = ((x1[1] - x2[1]) ^ 2 + (x1[2] - x2[2]) ^ 2 + (x1[3] - x2[3]) ^ 2 + (x1[4] - x2[4]) ^ 2)^0.5
53 print(dist)
54

```

Global Environment

values	
dist	10.816653826392
x1	num [1:4] 1 2 3 6
x2	num [1:4] 5 10 4 12

Files Plots Packages Help Viewer

Console Terminal Jobs

```

E:\DataScience_2019501123\datamining\DM Assignment5/
> setwd("E:\\DataScience_2019501123\\datamining\\DM Assignment5")
> #9(b)
> x1 <-c(1,2,3,6)
> x2 <-c(5,10,4,12)
> dist = ((x1[1] - x2[1]) ^ 2 + (x1[2] - x2[2]) ^ 2 + (x1[3] - x2[3]) ^ 2 + (x1[4] - x2[4]) ^ 2)^0.5
> print(dist)
[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 $z = \pm 3$ 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.

```

42
43 #8(b)
44 x1 <- c(1,2)
45 x2 <- c(5,10)
46 res = ((x1[1] - x2[1]) ^ 2 + ) ^ 0.5
47 print(res)
48
49 #9(b)
50 x1 <-c(1,2,3,6)
51 x2 <-c(5,10,4,12)
52 dist = ((x1[1] - x2[1]) ^ 2 + (x1[2] - x2[2]) ^ 2 + (x1[3] - x2[3]) ^ 2 + (x1[4] - x2[4]) ^ 2)^0.5
53 print(dist)
54
55 #11
56 exams <- read.csv("spring2008exams.csv")
57 str(exams)
58 mean1 <- mean(exams$Midterm.1, na.rm = TRUE)
59 sd1 <- sd(exams$Midterm.1, na.rm = TRUE)
60 z_score <- (exams$Midterm.1 - mean1) / sd1
61
62 sort(z_score)
63

```

Global Environment

Data

exams 17 obs. of 3 variables

values	
mean1	79.9411764705882
sd1	13.5483882262582
z_score	num [1:17] 0.0782 -0.5123...

Files Plots Packages Help Viewer

Console Terminal Jobs

```

E:\DataScience_2019501123\datamining\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 -0.06946778
[8] 0.07815125 0.07815125 0.37338932 0.37338932 0.37338932 0.66862740 0.66862740
[15] 0.66862740 1.25910354 1.84957968
>

```

Largest = 1.84957968 and smallest = -2.28375331 and No outliers are found.

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 $z = \pm 3$ 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.

```

51 x2 <- c(5,10,4,12)
52 dist = ((x1[1] - x2[1]) ^ 2 + (x1[2] - x2[2]) ^ 2 + (x1[3] - x2[3]) ^ 2 + (x1[4] - x2[4]) ^ 2) ^ 0.5
53 print(dist)
54
55 #11
56 exams <- read.csv("spring2008exams.csv")
57 str(exams)
58 mean1 <- mean(exams$Midterm.1, na.rm = TRUE)
59 sd1 <- sd(exams$Midterm.1, na.rm = TRUE)
60 z_score <- (exams$Midterm.1 - mean1) / sd1
61
62 sort(z_score)
63
64 #12
65 exams <- read.csv("spring2008exams.csv")
66 str(exams)
67 mean2 <- mean(exams$Midterm.2, na.rm = TRUE)
68 sd2 <- sd(exams$Midterm.2, na.rm = TRUE)
69 z_score2 <- (exams$Midterm.2 - mean2) / sd2
70
71 sort(z_score2)
72

```

```

> 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 ...
> mean2 <- mean(exams$Midterm.2, na.rm = TRUE)
> sd2 <- sd(exams$Midterm.2, na.rm = TRUE)
> z_score2 <- (exams$Midterm.2 - mean2) / sd2
> sort(z_score2)
[1] -2.39622252 -1.67310211 -0.78928828 -0.46790144 -0.38755473 -0.30720801 -0.06616788
[8] 0.01417883 0.01417883 0.01417883 0.17487225 0.33556568 0.89799266 1.05868608
[15] 1.05868608 1.21937950 1.29972622
>

```

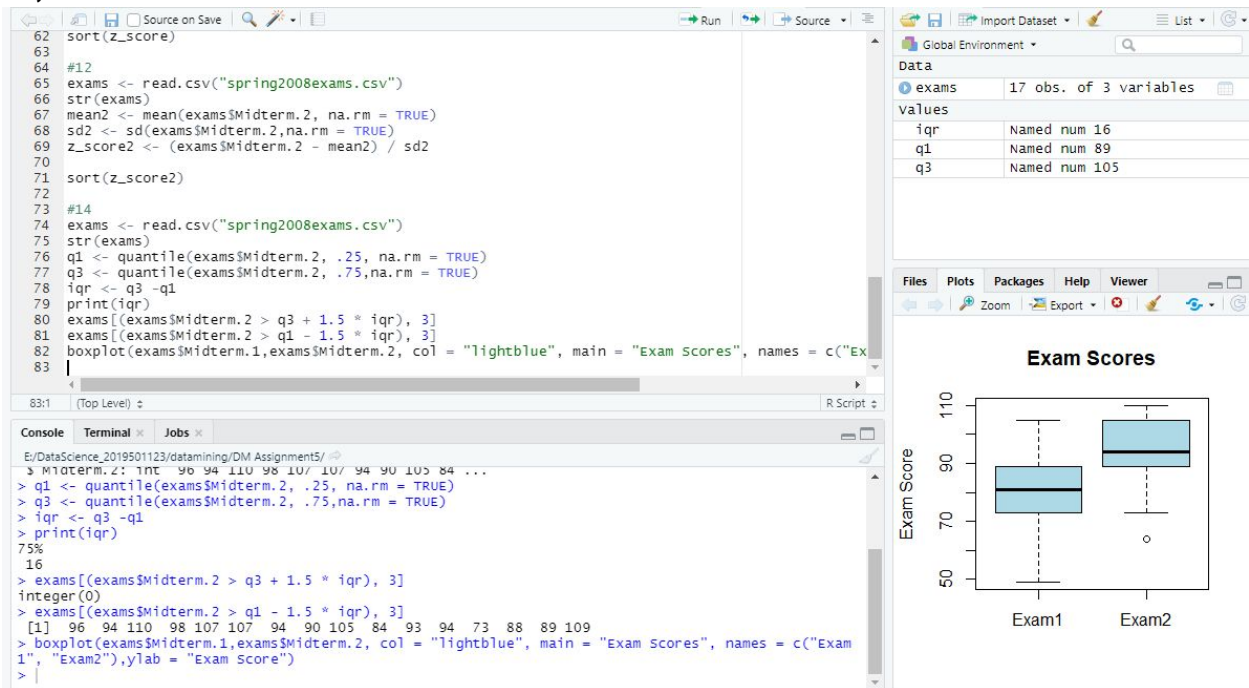
Largest = 1.29972622 and smallest = -2.39622252 and No outliers found

13) Repeat In Class Exercise #60 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 $z = \pm 3$ 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.)

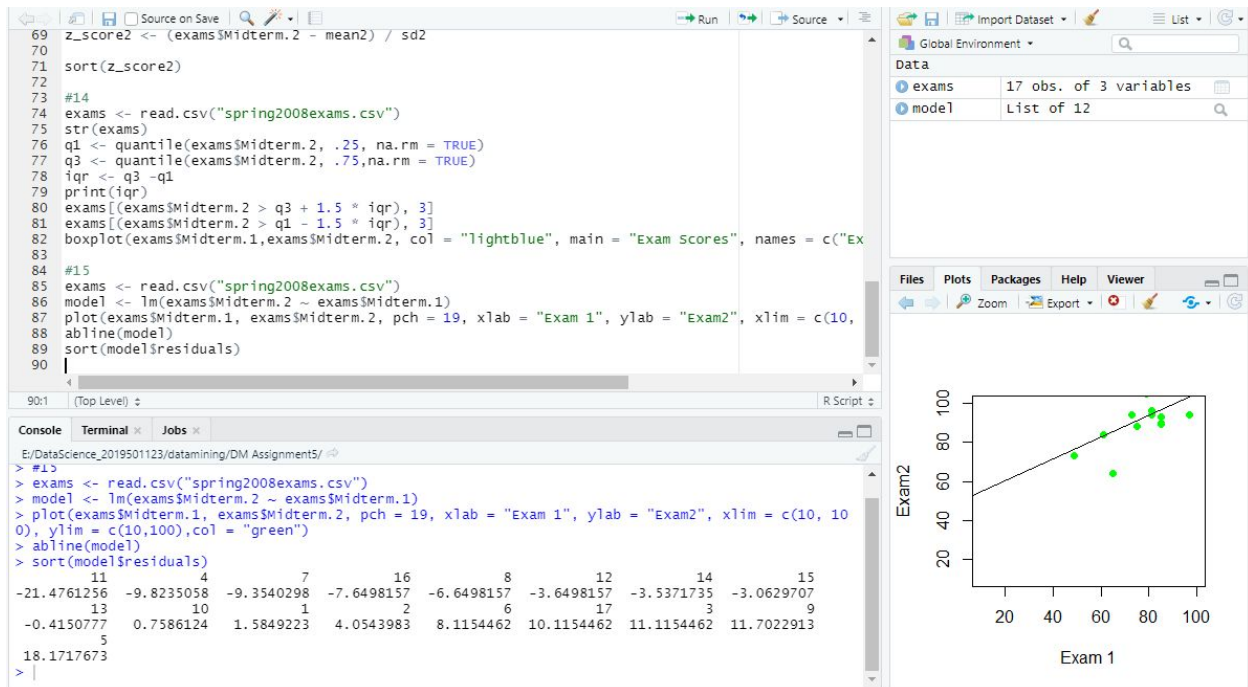
The only value which has outlier is with IP address 65.57.245.11 with a z-score of 8.426135321

Filename: [Question13_stats202Zscores.xlsx](#)

14) Repeat In Class Exercise #61 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?



15) Repeat In Class Exercise #62 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.



5th student has highest residual = 18.1717673