HW 2 Responses

Please let me know if you see any way I can improve upon my code, vocabulary, syntax or any answer. I am still very new to this and any help is much appreciated! Code will be highlighted in yellow and output in green. Page 11 on is all the code from R Studio.

Question 4.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a clustering model would be appropriate. List some (up to 5) predictors that you might use.

Answer: Why someone chooses which supplier they will use to procure a material is an example. Some predictors may be: price, time of delivery, shop distance to site, relationship with sales rep, or manufacturers that company distributes.

Question 4.2

The *iris* data set `iris.txt` contains 150 data points, each with four predictor variables and one categorical response. The predictors are the width and length of the sepal and petal of flowers and the response is the type of flower. The data is available from the R library datasets and can be accessed with iris once the library is loaded. It is also available at the UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Iris). *The response values are only given to see how well a specific method performed and should not be used to build the model.*

Use the R function `kmeans` to cluster the points as well as possible. Report the best combination of predictors, your suggested value of k, and how well your best clustering predicts flower type.

Code and Output:

```
#setting up working directory and reading the data.
iris_data <- read.table("4.2irisSummer2018.txt", header = TRUE)
```

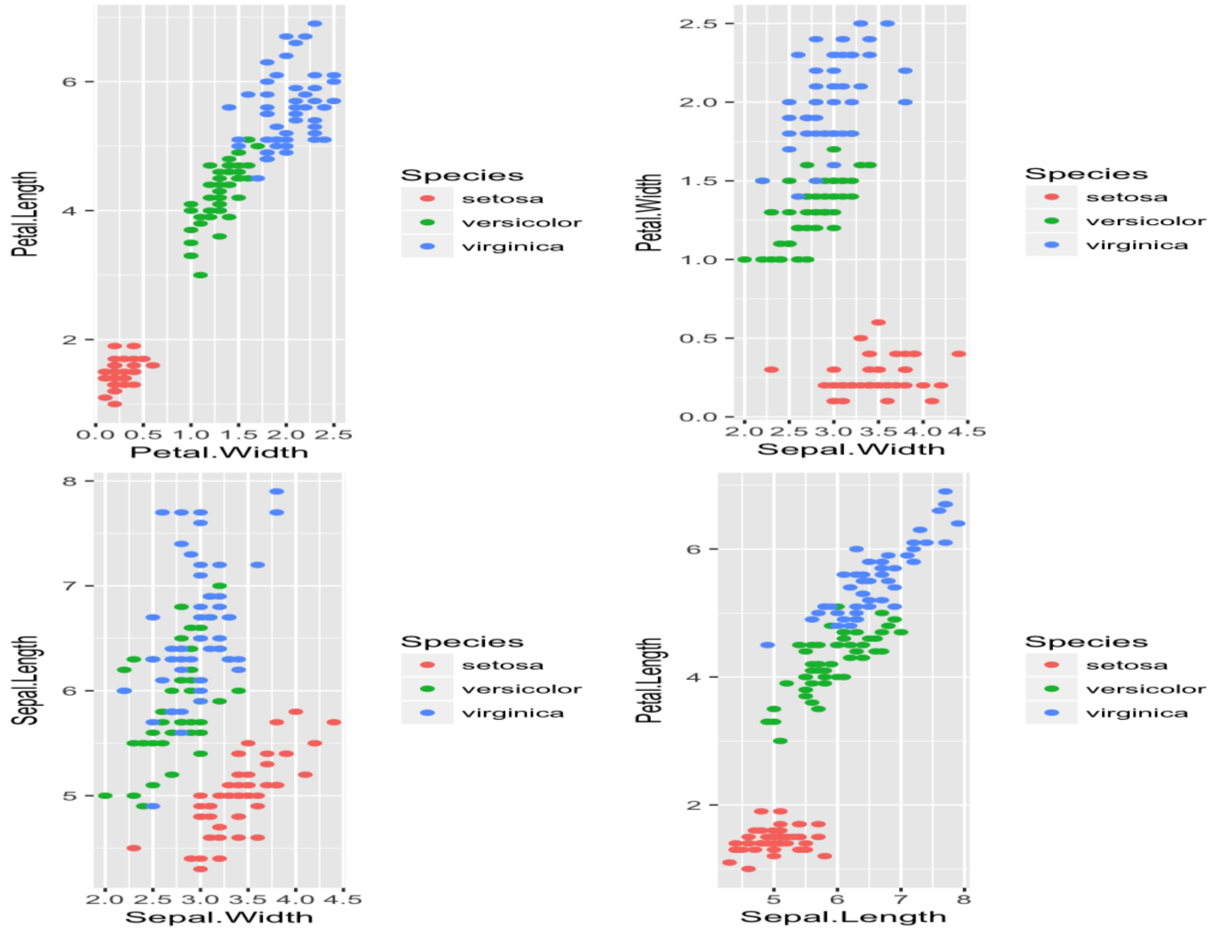
```
#installing the recommended package for this question
install.packages("ggplot2")
library(ggplot2)
```

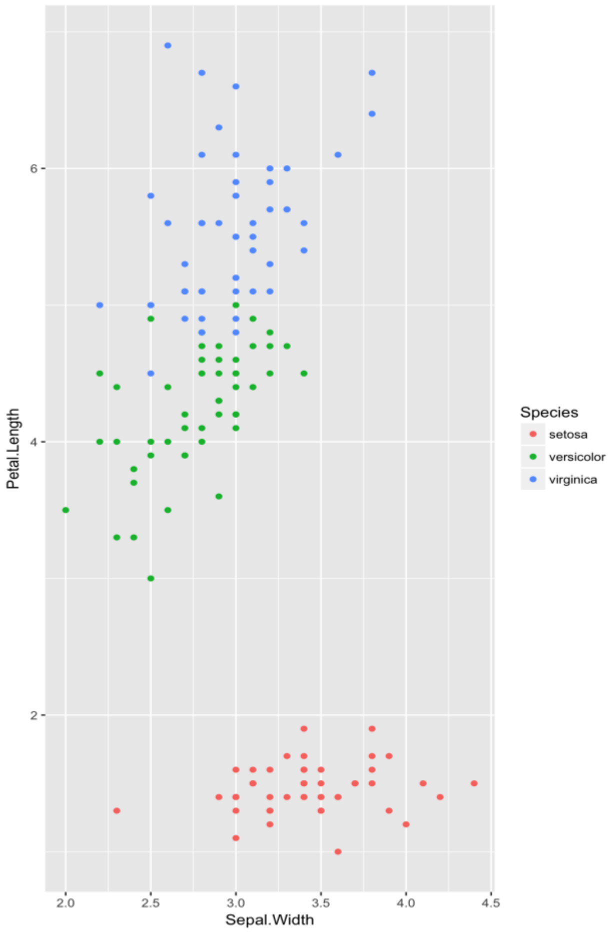
#plotted each predictor to give a visual of the data.

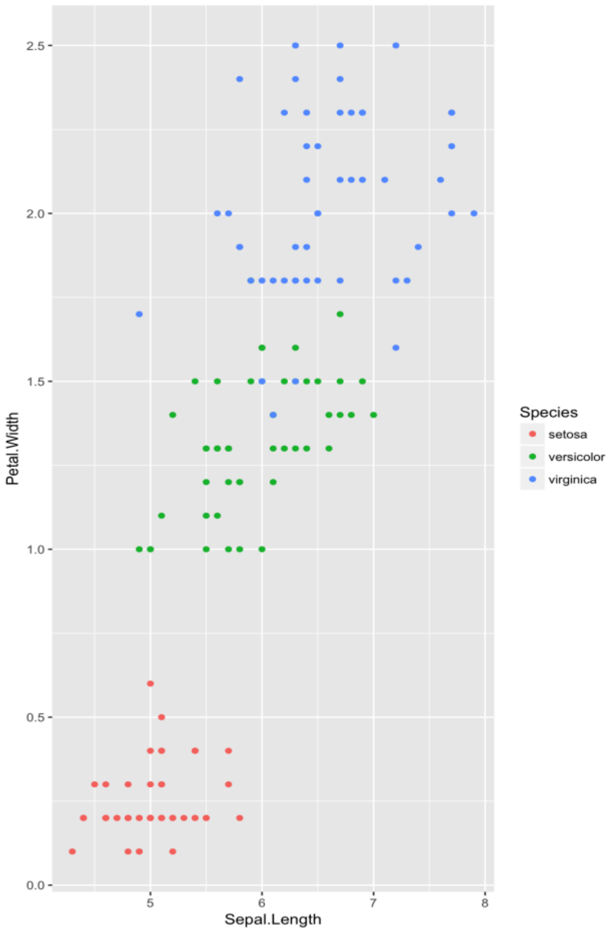
```
# Petal width v.  Petal length
P1 <- ggplot(iris_data, aes(Petal.Width, Petal.Length, color = Species))+ geom_point()
#Sepal Width vs. Sepal Length
P2 <- ggplot(iris_data, aes(Sepal.Width, Sepal.Length, color = Species)) + geom_point()
#Sepal Width vs. Petal Width
P3  <- ggplot(iris_data, aes(Sepal.Width, Petal.Width, color = Species))+geom_point()
#Sepal Length vs. Petal Length
```

```
P4 <- ggplot(iris_data, aes(Sepal.Length, Petal.Length, color = Species)) + geom_point()
#Sepal Width vs. Petal Length
P5 <- ggplot(iris_data, aes(Sepal.Width, Petal.Length, color = Species)) + geom_point()
#Sepal Length Vs. Petal Width
P6 <- ggplot(iris_data, aes(Sepal.Length, Petal.Width, color = Species)) +geom_point()
```

#Below is the output from the multiplot function. See code on last pages for setup of function





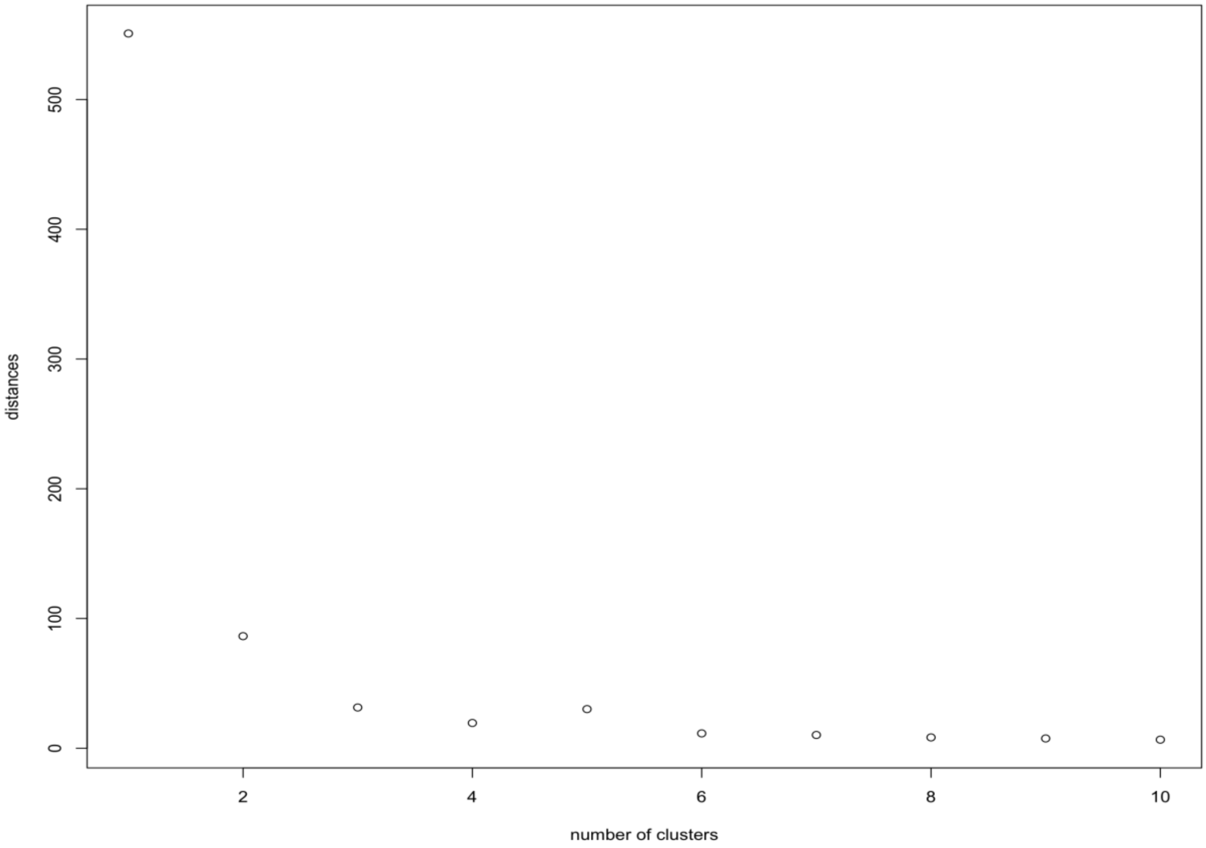


```
#find good number of clusters tried with 25 and 15. 10 gave much more clear picture.
distances <- rep(1,10)

#setting up an elbow diagram to find best k value or number of clusters. I plotted all predictor
#combinations this seemed to be the best at 3:4.
for (k_clusters in 1:10){
  clusters <- kmeans(iris_data[,3:4], k_clusters)
  distances[k_clusters] <- clusters$tot.withinss
}


#plotting the elbow diagram
plot(distances, xlab = "number of clusters")
```

#Below is the output from using predictors 3 and 4 (petal length and width)



#looking at the graph three looks to be the best predictor

#scaling the data to try kmeans when it is scaled

```
#scale the data. only using 2 through 5 for clustering from the data. This never seemed to
#worked for me.
scaled <- scale(iris_data[,2:5])
#output from the scaling
Error in colMeans(x, na.rm = TRUE) : 'x' must be numeric

#Initialize  data vectors
irisCluster <- vector(mode="list", length=10)
TotalSS <- rep(0, length=10)
TotalWithinSS <- rep(0, length=10)
kvalues <- rep(0, length=10)
```

```
#performing kmeans clustering
for (k in 1:10){
  irisCluster[[k]] <- kmeans(scaled, k, nstart = 20)
  TotalSS[k] <- irisCluster[[k]]$totss
  TotalWithinSS[k] <- irisCluster[[k]]$tot.withinss
  kvalues[k] <- k
}
```

Answer:

From using the kmeans algorithm the best k value seems to be 3. The best predictors would be petal length and width. Three is the elbow of the number of clusters v. distance graph. Distance does not get much if any closer after having more than 3 clusters.

Question 5.1

Using crime data from the file `uscrime.txt` (http://www.statsci.org/data/general/uscrime.txt, description at http://www.statsci.org/data/general/uscrime.html), test to see whether there are any outliers in the last column (number of crimes per 100,000 people). Use the `grubbs.test` function in the `outliers` package in R.

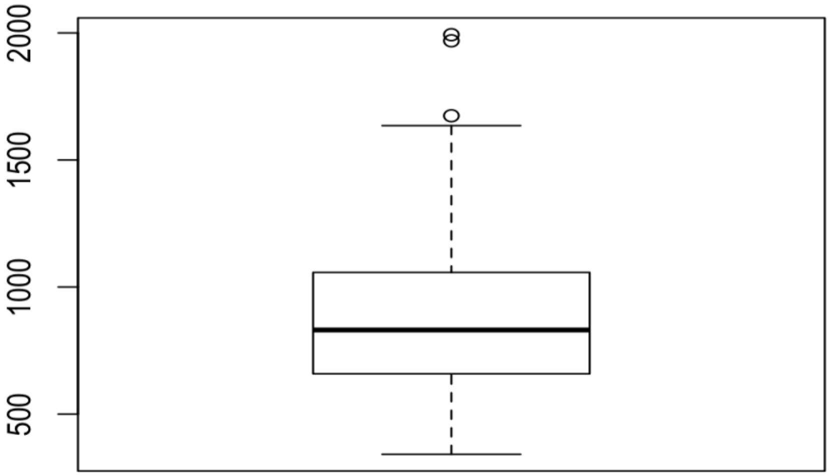
Code and Output:

```
#installing package recommended to use for question 5.1
install.packages("outliers")
library(outliers)
```

```
#setting up working directory and assigning data to be read
crime_data <- read.table("5.1uscrimeSummer2018.txt", stringsAsFactors = FALSE, header =
TRUE)
```

```
#using a box plot to understand where outliers may be
boxplot(crime)
```

```
#Below is the output
```



```
#grubbs test is used to find outliers


#two tailed test is 11 one tailed is 10.  Boxplot shows outliers may only appear to be on the high side.

#did two tailed, one tailed high and low test

#two tailed test
grubbs.test(crime_data$Crime, type=11)

#output from the grubbs test two tailed
        Grubbs test for two opposite outliers

data:  crime_data$Crime
G = 4.26880, U = 0.78103, p-value = 1
alternative hypothesis: 342 and 1993 are outliers

#grubbs test for one outlier on the low side
grubbs.test(crime_data$Crime, type=10, opposite = TRUE)

#output for one tailed test on low side
        Grubbs test for one outlier

data:  crime_data$Crime
G = 1.45590, U = 0.95292, p-value = 1
alternative hypothesis: lowest value 342 is an outlier

#one tailed test
grubbs.test(crime_data$Crime, type=10)

#output from the grubbs test for one outlier
Grubbs test for one outlier

data:  crime_data$Crime
G = 2.81290, U = 0.82426, p-value = 0.07887
alternative hypothesis: highest value 1993 is an outlier
```

Answer: For this question I made the assumption that we would reject the null hypothesis if the p-value < or = to .05. For the two tailed and one tailed lower grubbs test the p-value is 1. We would certainly fail to reject the null hypothesis for these two tests. For the one outlier grubbs test it gives a p-value of .07887 for the higher value 1993. This is close to .05, but using value of p must be less than or equal to .05 to reject the null hypothesis, this would not be an outlier. From running the test and using the assumption p must be less than .05 to reject the null hypothesis, there would be no outliers.

Question 6.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a Change Detection model would be appropriate. Applying the CUSUM technique, how would you choose the critical value and the threshold?

Answer: **Change detection could be used to help understand cost per month at our chemical site. The threshold could be found by looking at your average expected monthly spend or the average of the past monthly spend from prior data. After you find this you could find how much you usually go over the spend and see what the leadership wants to set as the threshold based on the budget. To help find the C you could see average fluctuation from T on past data to understand what is a significant change. Critical value would more than likely be zero for this scenario.**

Question 6.2

1. Using July through October daily-high-temperature data for Atlanta for 1996 through 2015, use a CUSUM approach to identify when unofficial summer ends (i.e., when the weather starts cooling off) each year. You can get the data that you need from the file temps.txt or online, for example at http://www.iweathernet.com/atlanta-weather-records or https://www.wunderground.com/history/airport/KFTY/2015/7/1/CustomHistory.html . You can use R if you'd like, but it's straightforward enough that an Excel spreadsheet can easily do the job too.

Answer: Please see attached excel sheet tab 6.2.1 for more information. Using CUSUM I came up with an unofficial summer ends date of September 16th. I used a Mu as the average of July-August temperature for each year. I was thinking this would be more of an accurate depiction of the summer temperature. C was set at 0 and T at 90 based off my observations. I assume this could be optimized, but looking at a few values, I found this to be the most accurate in excel. Below is a quick chart of the outcome of each year.

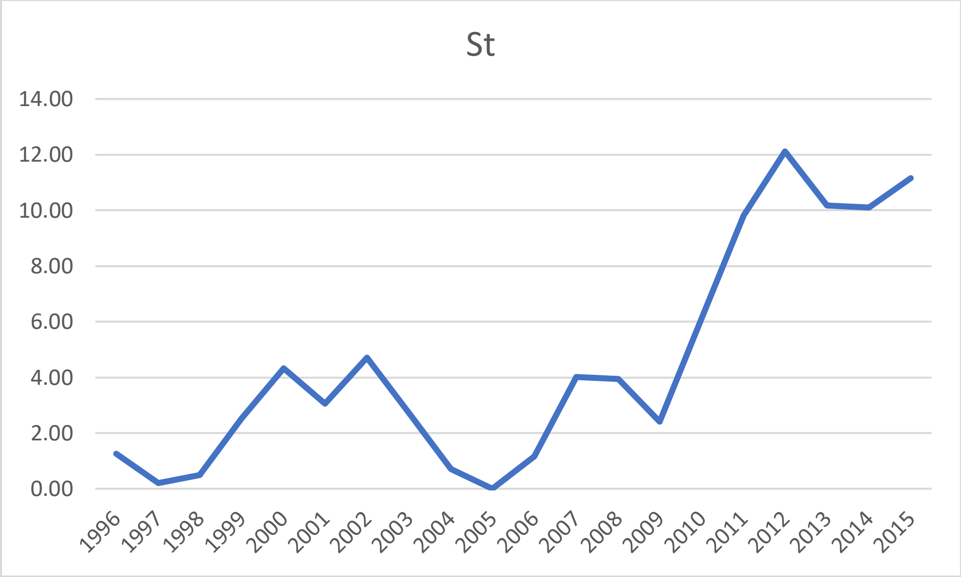
	Summer end Date	Month	Identifier (August 1(1)-October 31(92))
1996	1st	September	32
1997	26th	September	57
1998	12th	September	43
1999	20th	September	51
2000	2nd	September	33
2001	19th	September	50
2002	21st	September	52
2003	22nd	September	53
2004	6th	September	37
2005	7th	October	68
2006	6th	September	37
2007	15th	September	46
2008	13th	September	44
2009	9th	September	40
2010	29th	September	60
2011	9th	September	40
2012	28th	August	28
2013	30th	September	61
2014	27th	September	58
2015	12th	September	43
		Average	47
			September 16th

2. Use a CUSUM approach to make a judgment of whether Atlanta's summer climate has gotten warmer in that time (and if so, when).

Answer: For more information please see the 6.2.2 tab on the excel attachment. I used the average temperature for each year between 7/1-9/15 based off my answer for question 6.2.1 for Xt. Mu was calculated using the first ten years (1996-2005) average temperature. I was hoping this would show some change in the last ten years of the data if there is a steady increase. I set T=10 as 2 data points being over 91 moved the St graph very far. I do not believe using the CUSUM method with this few of data points you can accurately say it increased. However, according to my CUSUM model It got warmer in 2012 as Atlanta had a third year in a row of average temperatures above 89. Below is a quick chart.

Mu	T	C
87.40	10	0

Year	Xt (Average temperature 7/1 - 9/15)	Xt-Mu	St
1996	88.65	1.25	1.25
1997	86.35	-1.05	0.19
1998	87.70	0.30	0.49
1999	89.44	2.04	2.53
2000	89.21	1.81	4.34
2001	86.12	-1.29	3.05
2002	89.05	1.65	4.70
2003	85.42	-1.99	2.71
2004	85.40	-2.00	0.71
2005	86.69	-0.71	0.00
2006	88.56	1.16	1.16
2007	90.26	2.86	4.01
2008	87.32	-0.08	3.94
2009	85.87	-1.53	2.40
2010	91.12	3.71	6.12
2011	91.09	3.69	9.81
2012	89.71	2.31	12.12
2013	85.45	-1.95	10.17
2014	87.34	-0.06	10.10
2015	88.45	1.05	11.16



Code for HW:

```
#ctrl L clears the console
#good practice to always use this function to clear out ecerything
rm(list = ls())


##################################################################
#####################Question 4.2#################################
##################################################################


iris_data <- read.table("4.2irisSummer2018.txt", header = TRUE)
# set working directory through the session tab. Ctrl+Shift+H also will change working directory.
#view lets you see the data
View(iris_data)
#head and tail will give you a piece of the data
#command enter will run the current line/
head(iris_data)
tail(iris_data)
#summary will give you the summary of the data.
summary(iris_data)

#Setting up multiplot function
multiplot <- function(..., plotlist=NULL, file, cols=1, layout=NULL) {
  library(grid)

  # Make a list from the ... arguments and plotlist
  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  # If layout is NULL, then use 'cols' to determine layout
  if (is.null(layout)) {
    # Make the panel
    # ncol: Number of columns of plots
    # nrow: Number of rows needed, calculated from # of cols
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
  }

  if (numPlots==1) {
    print(plots[[1]])
  } else {
    # Set up the page
    grid.newpage()
    pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

    # Make each plot, in the correct location
    for (i in 1:numPlots) {
      # Get the i,j matrix positions of the regions that contain this subplot
      matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

      print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                      layout.pos.col = matchidx$col))
    }
  }
}
```

```
#ggplot2 is the recommended library to use.
#installing the package
install.packages("ggplot2")
library(ggplot2)
library(grid)

# Petal width v.  Petal length  maybe the best
P1 <- ggplot(iris_data, aes(Petal.Width, Petal.Length, color = Species))+ geom_point()
#Sepal Width vs. Sepal Length
P2 <- ggplot(iris_data, aes(Sepal.Width, Sepal.Length, color = Species)) + geom_point()
#Sepal Width vs. Petal Width
P3  <- ggplot(iris_data, aes(Sepal.Width, Petal.Width, color = Species))+geom_point()
#Sepal Length vs. Petal Length
P4 <- ggplot(iris_data, aes(Sepal.Length, Petal.Length, color = Species)) + geom_point()
#Sepal Width vs. Petal Length
P5 <- ggplot(iris_data, aes(Sepal.Width, Petal.Length, color = Species)) + geom_point()
#Sepal Length Vs. Petal Width
P6 <- ggplot(iris_data, aes(Sepal.Length, Petal.Width, color = Species)) +geom_point()


#plotted each on individually using plot.
plot(P1)

#plotting using the multiplot function that was set up earlier
multiplot(P1, P2, P3, P4, cols = 2)
multiplot(P5,P6, cols = 2)
```

```
#find good number of clusters
distances <- rep(1,10)


#elbow of the diagram tells you tge best clusters. You are using all 4 predictors here.
for (k_clusters in 1:10){
  clusters <- kmeans(iris_data[,3:4], k_clusters)
  distances[k_clusters] <- clusters$tot.withinss
}


plot(distances, xlab = "number of clusters")

iris_data[,2:5]
#scale the data. only using 2 through 5 for clsutering from the data.
scaled<-scale(iris_data[,2:5])
#Initialize data vectors
irisCluster <- vector(mode="list", length=10)
TotalSS <- rep(0, length=10)
TotalWithinSS <- rep(0, length=10)
kvalues <- rep(0, length=10)
#performing kmeans clustering
for (k in 1:10){
  irisCluster[[k]] <- kmeans(scaled, k, nstart = 20)
  TotalSS[k] <- irisCluster[[k]]$totss
  TotalWithinSS[k] <- irisCluster[[k]]$tot.withinss
  kvalues[k] <- k
}
```

```
###################################################################
####################Questions 5.1#################################
###################################################################


#installing the recommended package to use
install.packages("outliers")
library(outliers)

#choosing the data to read
crime_data <- read.table("5.1uscrimeSummer2018.txt", header = TRUE)

#setting up what wwe need to plot
crime <- crime_data[,"Crime"]

#using a box plot to understand where outliers may be
boxplot(crime)

#I just wanted to see what this would do
boxplot(crime_data)

#sampling the data
head(crime_data)

#seeing what the summary looked like
summary(crime_data)

#grubbs test is used to find outliers
#two tailed test is 11 one tailed is 10.

#testing for one outlier
grubbs.test(crime_data$Crime, type=10)

#testing for two tailed outlier
grubbs.test(crime_data$Crime, type=11)

#testing for outlier on low side
grubbs.test(crime_data$Crime, type=10, opposite = TRUE)




#####################################################################
#######################Question 6.2 in excel#########################
#####################################################################
```