title: "HW2" author: "William Florez" date: "31 de mayo de 2017" github: https://github.com/wiflore output: html_document: default word_document: default ---

HW2

Question 1

A clustering problem for could be *Fixed Income - Treasuries Asset Allocation*

The possible outcome could be:

1. Long run bonds looks better vs Short Term bonds
2. Short run bonds looks better vs Long Term bonds
3. Long run bonds looks and Short Term bonds looks good
4. Long run bonds looks and Short Term bonds looks bad

As a predictors I could use:
CPI actual
CPI Forecast
Central Bank Rate actual
Central Bank Forecast
GDP actual
GDP Forecast
Consumer Sentiment
Private Payrolls
Dividend Yield vs Corporate Bond Yield
US vs German bonds yield spread
Leading indicators
US 2year vs US 10year yield spread

Question 2

```
library(gplots)

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess

library(datasets)
library(ggplot2)

set.seed(123)

data <- iris
head(data)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
x = data[,-5]
y = data$Species
```

```
kcx <- kmeans(x, 3, nstart = 20)
kcx
```

```
## K-means clustering with 3 clusters of sizes 50, 38, 62
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.006000    3.428000     1.462000    0.246000
## 2     6.850000    3.073684     5.742105    2.071053
## 3     5.901613    2.748387     4.393548    1.433871
##
## Clustering vector:
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3
##  [71] 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3
2 2 2
## [106] 2 3 2 2 2 2 2 2 3 3 2 2 2 2 3 2 3 2 3 2 2 3 3 2 2 2 2 2 3 2 2 2
2 3 2
## [141] 2 2 3 2 2 2 3 2 2 3
##
## Within cluster sum of squares by cluster:
## [1] 15.15100 23.87947 39.82097
##  (between_SS / total_SS =  88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
#Check correlation for looking improvements
cor(x)
```

```
##              Sepal.Length Sepal.Width Petal.Length Petal.Width
## Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
## Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
## Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
## Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

```r
#Sepal.Width very uncorrelated variable
#Rerunning without Sepal.width
z = x[,-2]
kcz <- kmeans(z, 3, nstart = 20)
kcz
```

```
## K-means clustering with 3 clusters of sizes 38, 50, 62
##
## Cluster means:
##   Sepal.Length Petal.Length Petal.Width
## 1     6.850000     5.742105    2.071053
## 2     5.006000     1.462000    0.246000
## 3     5.901613     4.393548    1.433871
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2
##  [36] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3
3 3 3
##  [71] 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 3
1 1 1
## [106] 1 3 1 1 1 1 1 1 3 3 1 1 1 1 3 1 3 1 3 1 1 3 3 1 1 1 1 1 3 1 1 1
1 3 1
## [141] 1 1 3 1 1 1 3 1 1 3
##
## Within cluster sum of squares by cluster:
## [1] 20.76579  8.11020 34.46613
##  (between_SS / total_SS =  90.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
#Comparing Results
table(y,kcx$cluster)
```

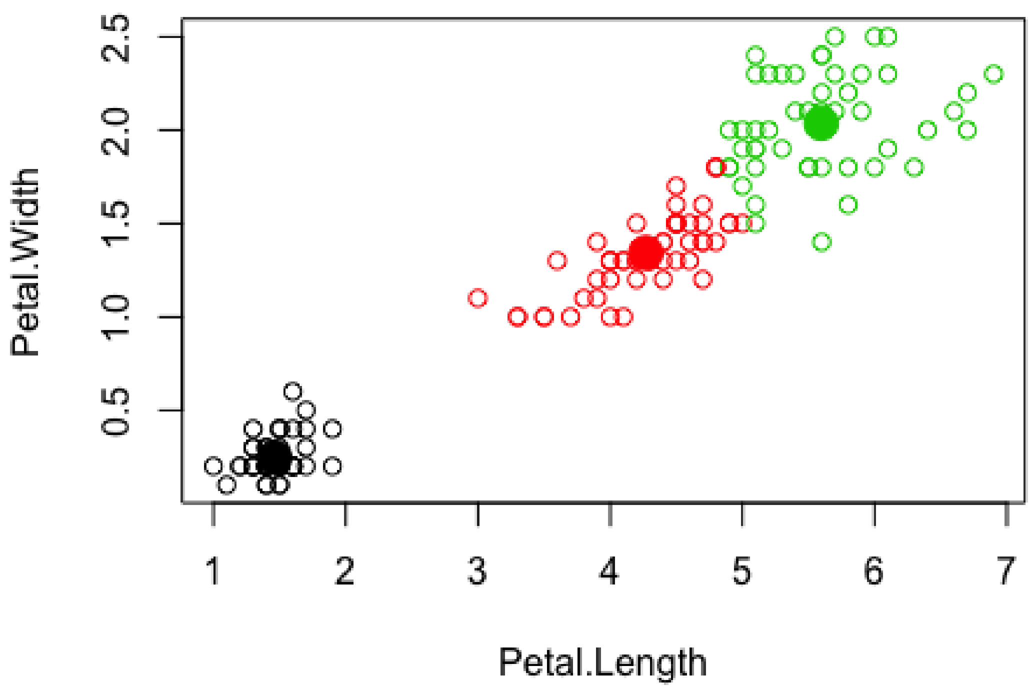
```
##
## y             1  2  3
##   setosa     50  0  0
##   versicolor  0  2 48
##   virginica   0 36 14
```

```r
table(y,kcz$cluster)
```

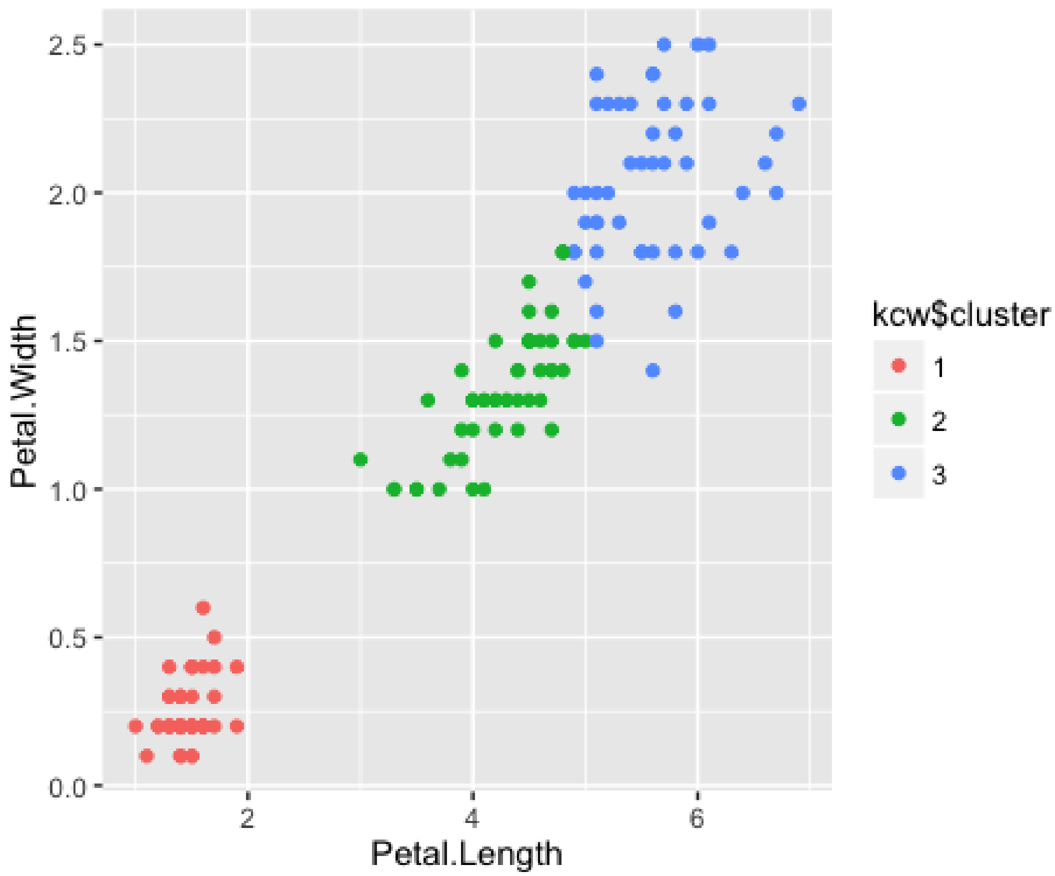
```
##
## y             1  2  3
##   setosa      0 50  0
##   versicolor  2  0 48
##   virginica  36  0 14
```

```
#Nothing Improve. So I will test with the most correlated variables
Petal.Lengh and Petal.Width
w = z[,-1]
kcw <- kmeans(w, 3, nstart = 20)
kcw

## K-means clustering with 3 clusters of sizes 50, 52, 48
##
## Cluster means:
##   Petal.Length Petal.Width
## 1     1.462000    0.246000
## 2     4.269231    1.342308
## 3     5.595833    2.037500
##
## Clustering vector:
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
2 2 2
##  [71] 2 2 2 2 2 2 2 3 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3
3 3 3
## [106] 3 2 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3
3 2 3
## [141] 3 3 3 3 3 3 3 3 3 3
##
## Within cluster sum of squares by cluster:
## [1]  2.02200 13.05769 16.29167
##  (between_SS / total_SS =  94.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"

table(y,kcz$cluster)

##
## y             1  2  3
##   setosa      0 50  0
##   versicolor  2  0 48
##   virginica  36  0 14

table(y,kcw$cluster)

##
## y             1  2  3
##   setosa     50  0  0
##   versicolor  0 48  2
##   virginica   0  4 46
```

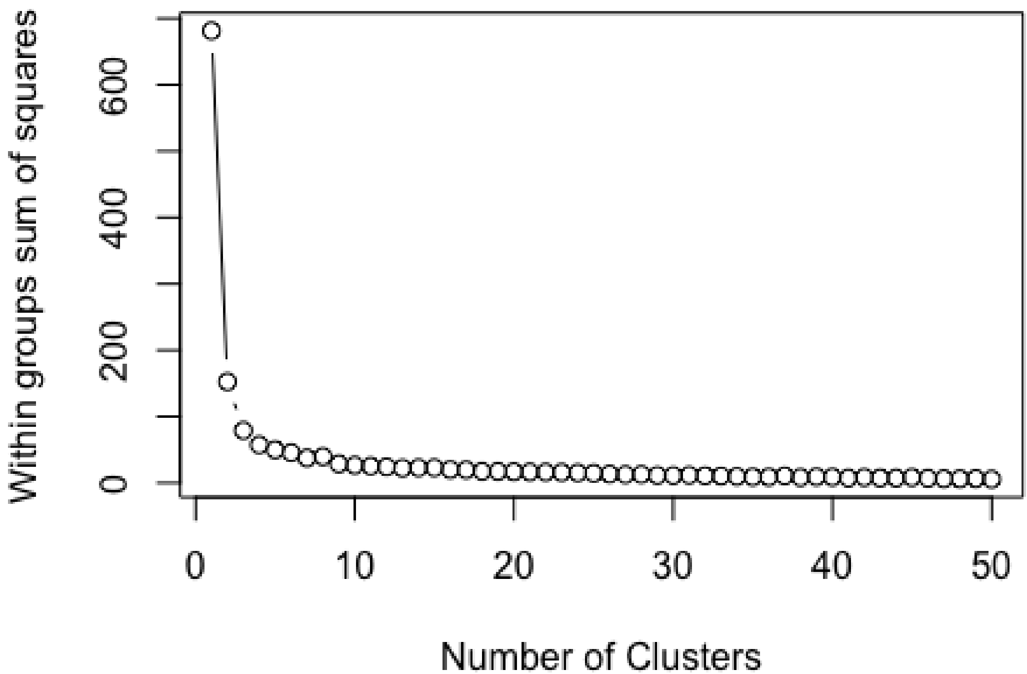
```
#That Looks better

#Ploting
plot(x[c("Petal.Length", "Petal.Width")], col=kcw$cluster)
points(kcw$centers[,c("Petal.Length", "Petal.Width")], col=1:3, pch=20,
cex=3)
```




```
kcw$cluster <- as.factor(kcw$cluster)
ggplot(data, aes(Petal.Length, Petal.Width, color = kcw$cluster)) +
geom_point()
```




```
#Is this were a unsupervised problem the best way to determine the K is
looking the sum of squares

wss = NULL
##Looking best k for unsupervised problem
for (i in 1:50) wss[i] <- sum(kmeans(x,
                                     centers = i)$withinss)
plot(1:50, wss, type = "b", xlab = "Number of Clusters",
     ylab = "Within groups sum of squares")
```



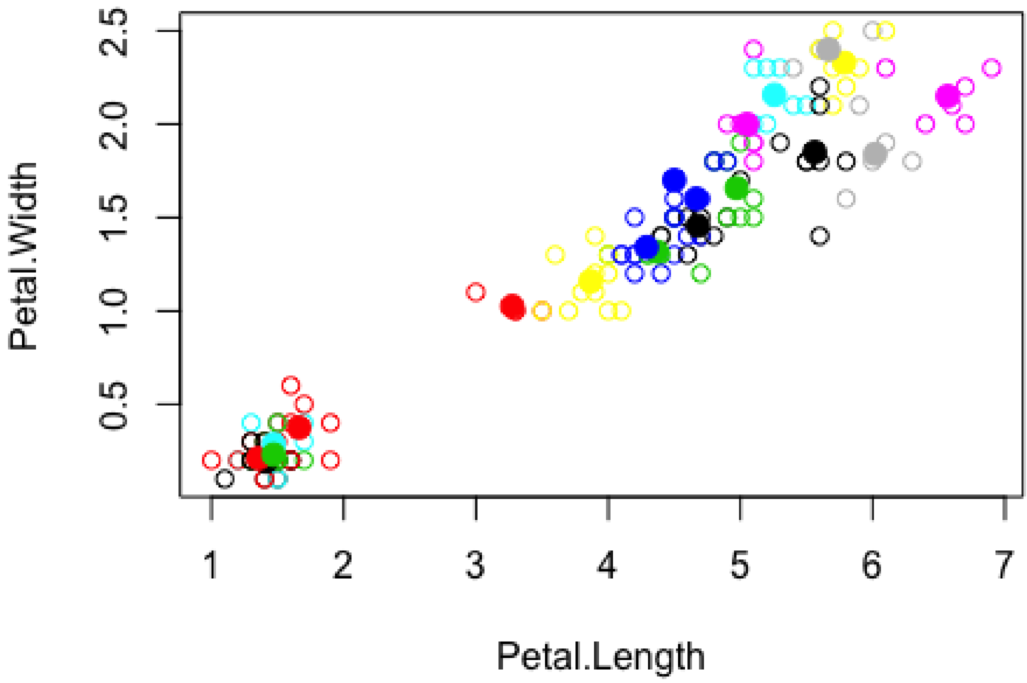

```
#Clustering with K = 20
kc <- kmeans(x,20)
kc

## K-means clustering with 20 clusters of sizes 16, 9, 6, 1, 7, 6, 7, 5,
9, 4, 10, 9, 7, 6, 12, 3, 8, 8, 7, 10
##
## Cluster means:
##    Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1      4.668750    3.025000     1.412500   0.1937500
## 2      4.877778    3.444444     1.355556   0.2111111
## 3      6.216667    2.650000     4.366667   1.3166667
## 4      4.900000    2.500000     4.500000   1.7000000
## 5      5.528571    4.042857     1.471429   0.2857143
## 6      7.716667    3.166667     6.566667   2.1500000
## 7      6.785714    3.242857     5.785714   2.3285714
## 8      7.240000    2.980000     6.020000   1.8400000
## 9      6.722222    3.000000     4.677778   1.4555556
## 10     5.000000    2.300000     3.275000   1.0250000
## 11     5.270000    3.500000     1.470000   0.2300000
## 12     6.100000    3.100000     4.666667   1.6000000
## 13     6.671429    3.085714     5.257143   2.1571429
## 14     5.766667    2.750000     5.050000   2.0000000
```

```
## 15     5.625000    2.541667     3.866667   1.1583333
## 16     6.266667    3.366667     5.666667   2.4000000
## 17     6.400000    2.800000     5.562500   1.8500000
## 18     5.037500    3.587500     1.662500   0.3750000
## 19     6.200000    2.600000     4.971429   1.6571429
## 20     5.640000    2.880000     4.290000   1.3400000
##
## Clustering vector:
##   [1] 11  1  1  1  2  5  2  2  1  1 11  2  1  1  5  5  5 11  5 18 11
18  2
##  [24] 18 18  1 18 11 11  1  1 11  5  5  1  2 11  2  1 11  2  1  1 18
18  1
##  [47] 18  1 11  2  9 12  9 15  9 20 12 10  9 15 10 20 15 12 15  9 20
15  3
##  [70] 15 12  3 19  3  3  9  9  9 12 15 15 15 15 19 20 12  9  3 20 15
20 12
##  [93] 15 10 20 20 20  3 10 20 16 14  8 17  7  6  4  8 17  7 13 17 13
14 14
## [116] 13 17  6  6 19  7 14  6 19  7  8 19 12 17  8  8  6 17 19 17  6
16 17
## [139] 12 13  7 13 14  7  7 13 19 13 16 14
##
## Within cluster sum of squares by cluster:
##  [1] 1.7312500 0.6688889 0.8850000 0.0000000 0.8342857 1.8500000
0.8085714
##  [8] 0.4200000 0.7933333 0.2950000 0.4630000 0.8600000 0.5571429
0.4433333
## [15] 1.5075000 0.2200000 1.0187500 0.7012500 0.6514286 0.7330000
##  (between_SS / total_SS =  97.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
table(y,kc$cluster)
```

```
##
## y             1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
20
##   setosa     16  9  0  0  7  0  0  0  0  0 10  0  0  0  0  0  0  8  0
0
##   versicolor  0  0  6  0  0  0  0  0  9  4  0  7  0  0 12  0  0  0  2
10
##   virginica   0  0  0  1  0  6  7  5  0  0  0  2  7  6  0  3  8  0  5
0
```

```r
plot(x[c("Petal.Length", "Petal.Width")], col=kc$cluster)
points(kc$centers[,c("Petal.Length", "Petal.Width")], col=1:20, pch=20,
cex=2)
```




```
#This is hard to analyze and looks overfitting
```

To summarize

The best combination of predictors are: Petal.Length and Petal.Width

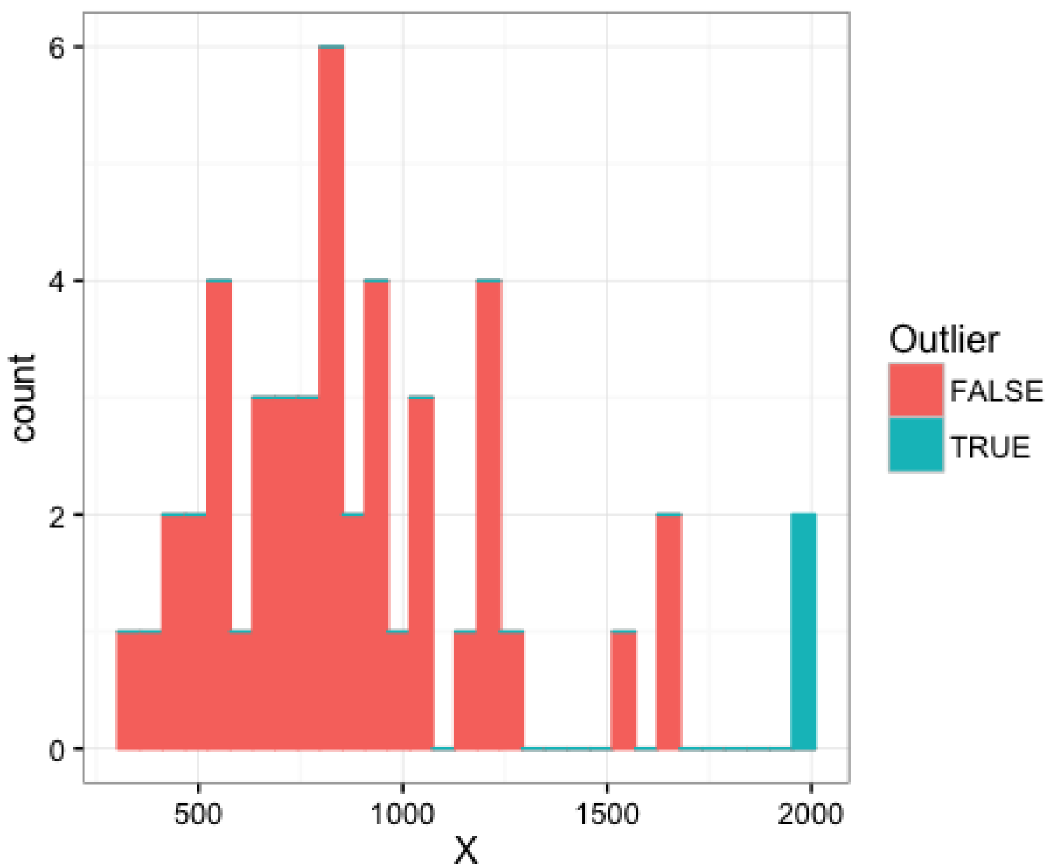
My suggested value of k = 3, because we already know there are 3 classes. Evaluating ebow chart a k betweetn 20-30 could reduce error squares but looks overfitting and hard to interpreted

How well your best clustering predicts flower type: 96% of accuracy

Question 3

```
library(outliers)
library(ggplot2)
library(RCurl)

## Loading required package: bitops

data <- getURL("http://www.statsci.org/data/general/uscrime.txt")
data = read.table(text = data, header = TRUE)

X <- data$Crime
grubbs.flag <- function(x) {
```

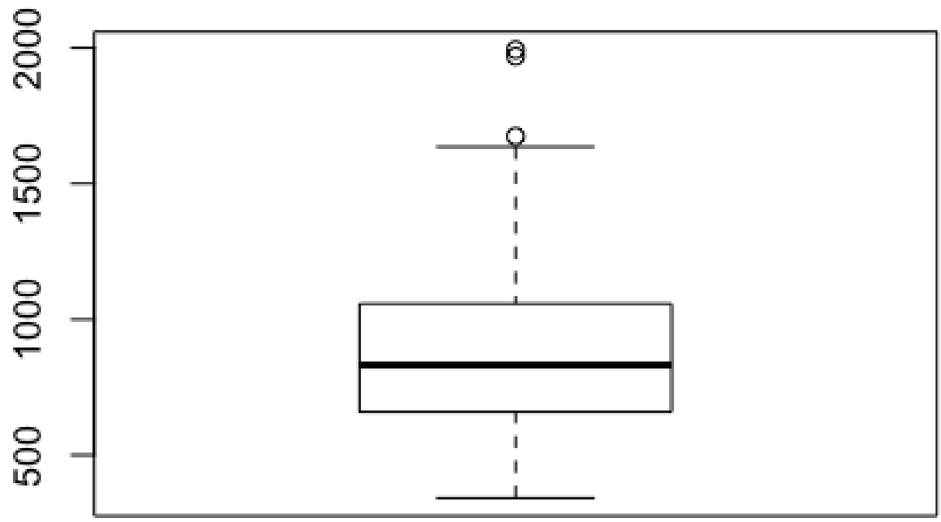
```
  outliers <- NULL
  test <- x
  grubbs.result <- grubbs.test(test)
  pv <- grubbs.result$p.value
  while(pv < 0.1) {
    outliers <-
c(outliers,as.numeric(strsplit(grubbs.result$alternative," ")[[1]][3]))
    test <- x[!x %in% outliers]
    grubbs.result <- grubbs.test(test)
    pv <- grubbs.result$p.value
  }
  return(data.frame(X=x,Outlier=(x %in% outliers)))
}

# Plot the outliers highlighted in colour:

ggplot(grubbs.flag(X),aes(x=X,color=Outlier,fill=Outlier))+
  geom_histogram(binwidth=diff(range(X))/30)+
  theme_bw()
```



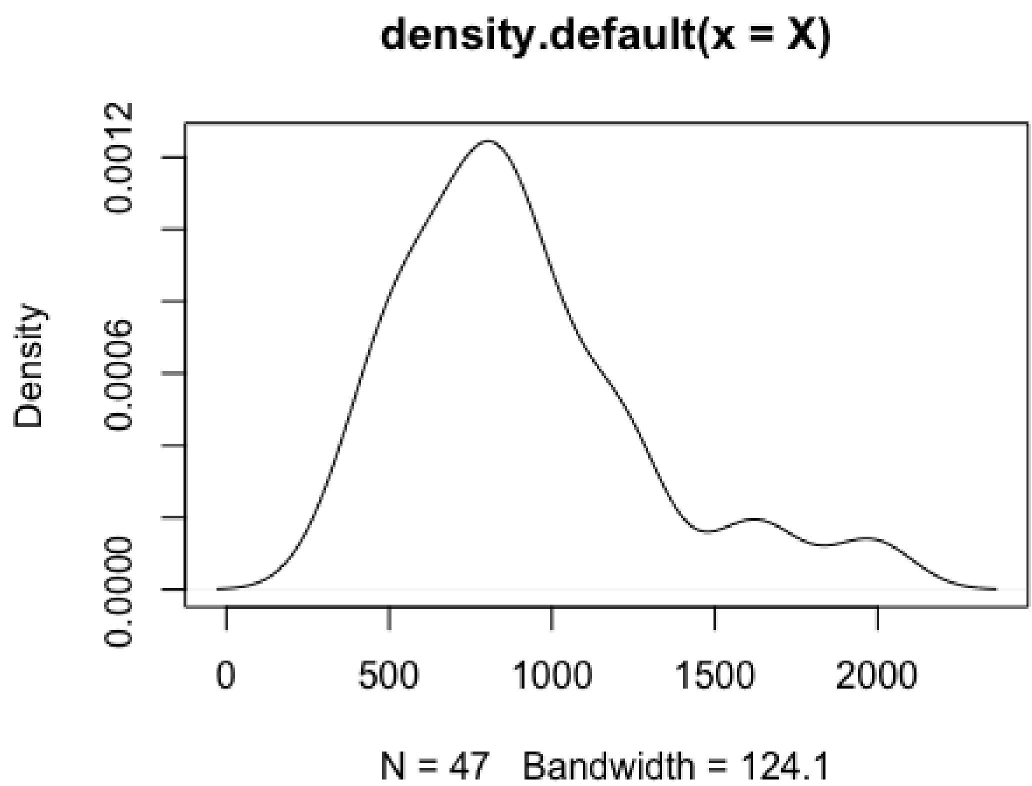

```
grubbs.flag(X)
```

```
##        X Outlier
## 1   791   FALSE
## 2  1635   FALSE
## 3   578   FALSE
## 4  1969    TRUE
## 5  1234   FALSE
## 6   682   FALSE
## 7   963   FALSE
## 8  1555   FALSE
## 9   856   FALSE
## 10  705   FALSE
## 11 1674   FALSE
## 12  849   FALSE
## 13  511   FALSE
## 14  664   FALSE
## 15  798   FALSE
## 16  946   FALSE
## 17  539   FALSE
## 18  929   FALSE
## 19  750   FALSE
## 20 1225   FALSE
## 21  742   FALSE
## 22  439   FALSE
## 23 1216   FALSE
## 24  968   FALSE
## 25  523   FALSE
## 26 1993    TRUE
## 27  342   FALSE
## 28 1216   FALSE
## 29 1043   FALSE
## 30  696   FALSE
## 31  373   FALSE
## 32  754   FALSE
## 33 1072   FALSE
## 34  923   FALSE
## 35  653   FALSE
## 36 1272   FALSE
## 37  831   FALSE
## 38  566   FALSE
## 39  826   FALSE
## 40 1151   FALSE
## 41  880   FALSE
## 42  542   FALSE
## 43  823   FALSE
## 44 1030   FALSE
## 45  455   FALSE
## 46  508   FALSE
## 47  849   FALSE
```

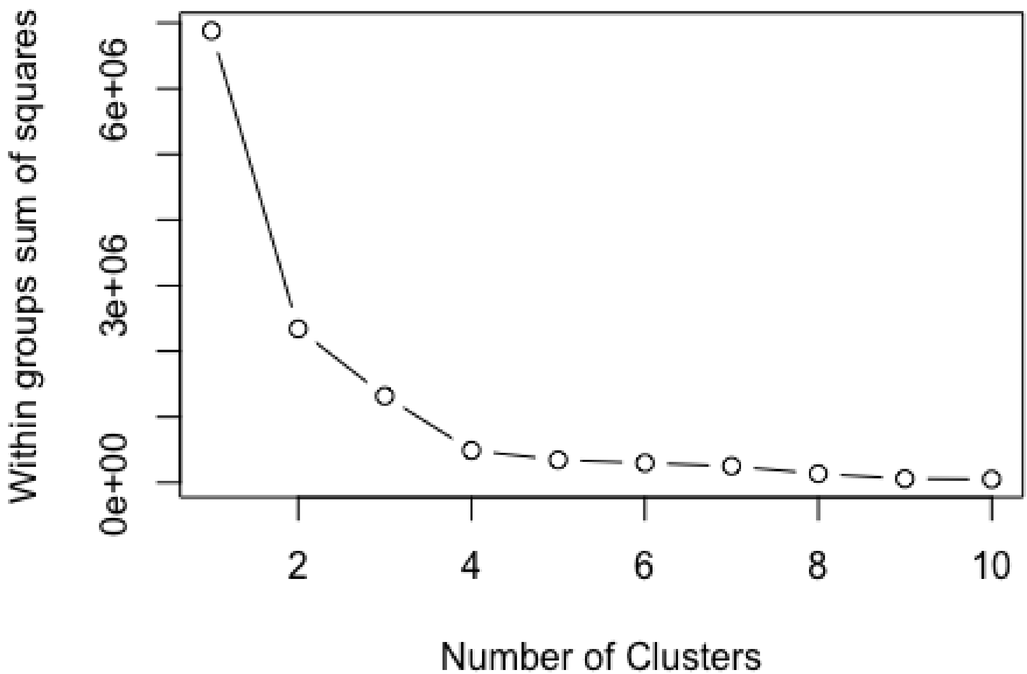
```
d = density(X)
boxplot(X)
```




```
plot(d)
```




```
boxplot.stats(X)$out

## [1] 1969 1674 1993

#The initial conclusion is that highest-crime is a outlier with a
cofident of
#let see clustering for better understading

wss = NULL
##Looking best k for unsupervised problem
for (i in 1:10) wss[i] <- sum(kmeans(X,
                                     centers = i)$withinss)
plot(1:10, wss, type = "b", xlab = "Number of Clusters",
     ylab = "Within groups sum of squares")
```




```
#ebow chart suggets k = 4 so let cluster the data

k = 4
kc = kmeans(X, centers = k)
kc

## K-means clustering with 4 clusters of sizes 13, 20, 5, 9
##
## Cluster means:
##        [,1]
## 1  514.8462
## 2  828.0500
## 3 1765.2000
## 4 1162.1111
##
## Clustering vector:
##  [1] 2 3 1 3 4 2 2 3 2 2 3 2 1 1 2 2 1 2 2 4 2 1 4 2 1 3 1 4 4 2 1 2 4
2 1
## [36] 4 2 1 2 4 2 1 2 4 1 1 2
##
## Within cluster sum of squares by cluster:
## [1] 108717.69 150476.95 162880.80  66890.89
##  (between_SS / total_SS =  92.9 %)
```

```
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
clusters = kc$cluster
testing = data
testing = data.frame(scale(testing))
testing$Clusters = factor(clusters, levels = 1:k,
                          labels = letters[1:k])
table(clusters)
```

```
## clusters
##  1  2  3  4
## 13 20  5  9
```

```
aggregate(data, by = list(testing$Clusters),FUN = mean)
```

```
##   Group.1        M        So       Ed       Po1       Po2        LF
## 1       a 13.63077 0.2307692 10.33077  6.384615  6.000000 0.5572308
## 2       b 14.14500 0.5000000 10.40000  8.055000  7.665000 0.5590000
## 3       c 13.30000 0.2000000 11.38000 12.960000 12.080000 0.5826000
## 4       d 13.85556 0.2222222 10.81111 10.066667  9.488889 0.5598889
##         M.F      Pop        NW         U1       U2   Wealth     Ineq
## 1  98.13077 23.76923  7.269231 0.09900000 3.300000 4744.615 19.96923
## 2  98.12500 30.70000 12.995000 0.09415000 3.390000 5235.000 19.60500
## 3 100.24000 64.80000 10.880000 0.09120000 3.720000 6066.000 17.78000
## 4  97.86667 52.66667  7.388889 0.09566667 3.377778 5580.000 19.02222
##         Prob     Time     Crime
## 1 0.05856931 25.08466  514.8462
## 2 0.05083475 25.81493  828.0500
## 3 0.02867960 28.70008 1765.2000
## 4 0.03242233 29.35584 1162.1111
```

```
#There are not a clusters near to highst data crime
```

Is the lowest-crime city an outlier? No because the distrubution of the crime has more density to the right, so is considered a more normal data

Is the highest-crime city an outlier? Could be, the answer for a first insight is yes, the point is very distant from the median. I look more information clustering the data and clusters did't shows that highest-crime is near to ones of them. However is necessary a more depper analysis to determinate if is really a outlier/rare or should be consider as a event to take into account.

Question 4

I could use CUSUM with the differences of returns between a stock and its peers. For example, **the critical value** is the difference of the cummulative return of APPLE vs its peers (AMAZON, GOOGLE, SAMSUNG, ETC) cross some point it point that something it is happening with the company and could be a signal to buy or sell subject to if cross is negative or positive.

Question 5 - look excel file

```
library(outliers)
library(ggplot2)
library(RCurl)

data <-
getURL("https://d37djvu3ytnwxt.cloudfront.net/assets/courseware/v1/592f3be3e90d2bdfe6a69f62374a1250/asset-v1:GTx+ISYE6501x+2T2017+type@asset+block/temps.txt")
data = read.table(text = data, header = TRUE)

plot(data)
```