Exercise1

Vijai Kasthuri Rangan

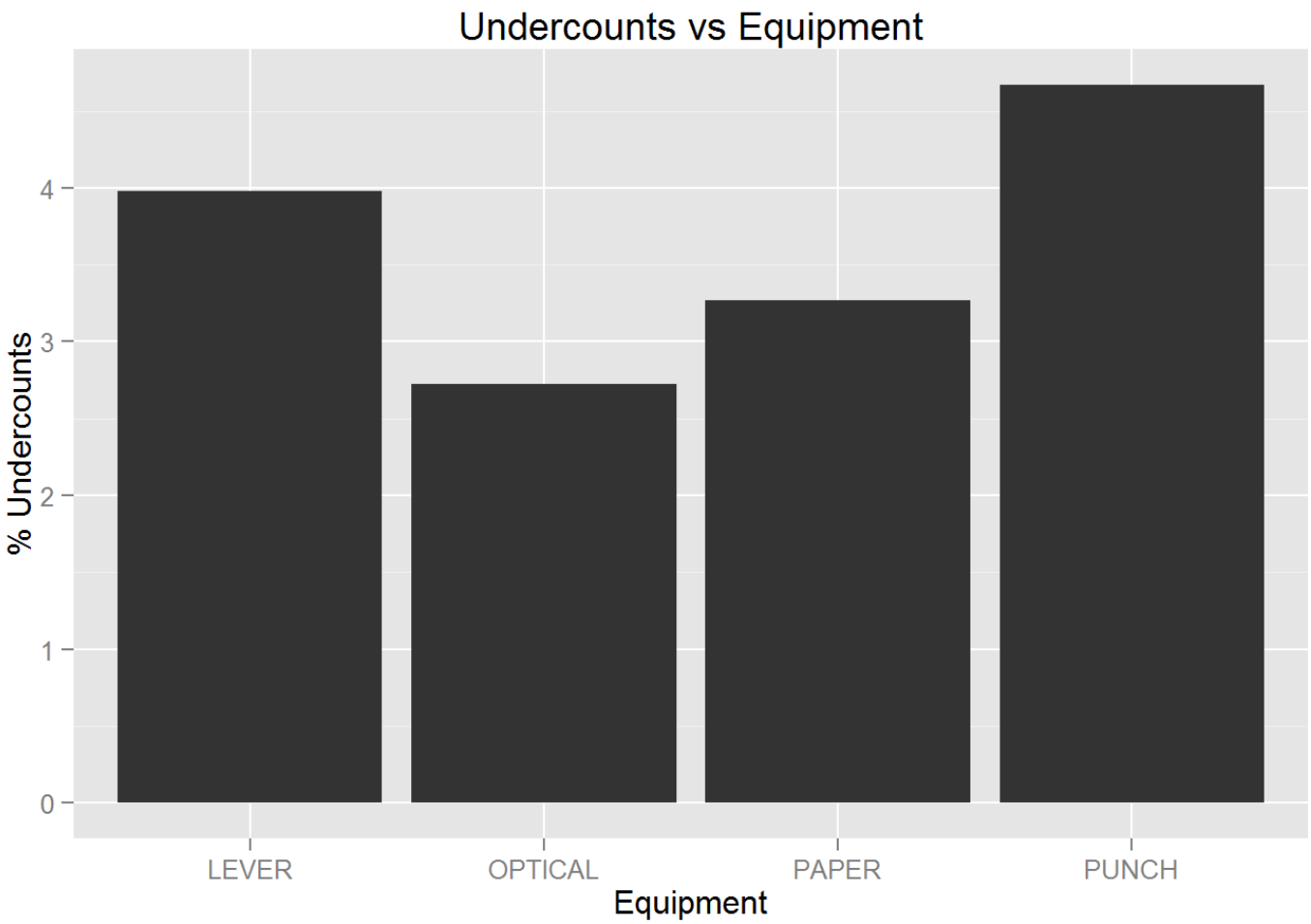
Exploratory Analysis

In order to understand the issue of undercount we first load the data to a data frame. The total number of undercounts is calculated in the different counties by finding the difference between the number of votes and the ballots. A bar plot of the number of undercounts in each county is plotted to understand the number of undervotes recorded in each county.

```
library(ggplot2)

georgia_data = read.csv("C:/Users/Vijai/OneDrive/Github/STA380-04082015/STA380-master/dat
a/georgia2000.csv")
georgia_data["undercount"] = georgia_data["ballots"] - georgia_data["votes"]
```

In order to understand the percentage of undercount across the different equipments used we plot the different equipment against the percentage of undervotes recorded against that equipment.

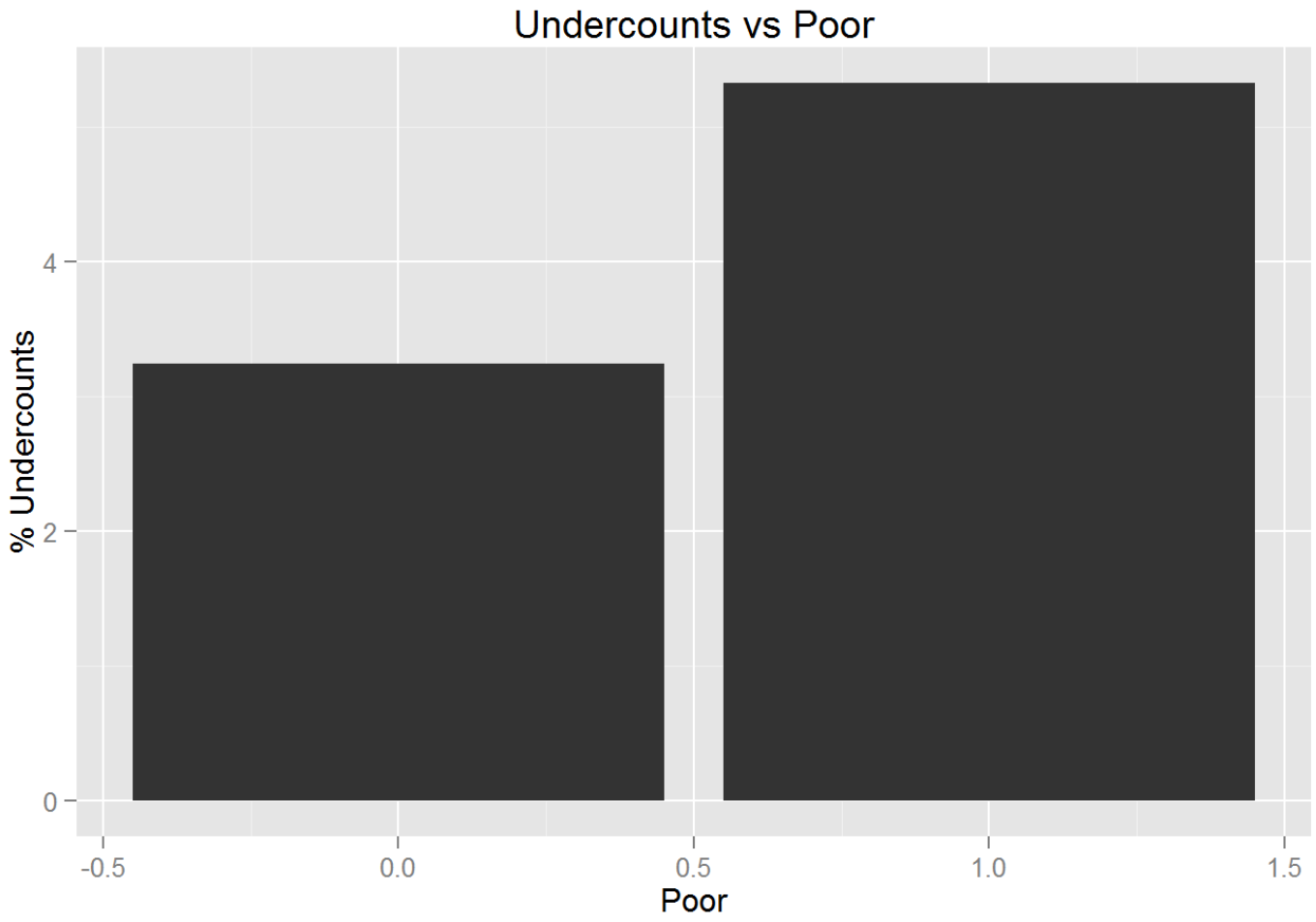
```
##Show Equips that lead to more undercount vote %

ballot_equip = aggregate(ballots~equip, data=georgia_data, sum)
undercount_equip = aggregate(undercount~equip, data=georgia_data, sum)
merged_df_equip = merge(ballot_equip,undercount_equip, by ="equip")
merged_df_equip["ratio"] = merged_df_equip["undercount"]/merged_df_equip["ballots"] * 100

ggplot(merged_df_equip, aes(x = merged_df_equip$equip, y = merged_df_equip$ratio) ) + geo
m_bar(stat = "identity")+ ggtitle("Undercounts vs Equipment") + xlab ( "Equipment") + yla
b ("% Undercounts")
```



The above plot shows that the rate of undercounts vary for each equipment and it is the maximum for the punch equipment.

A similar graph is plotted for the poor people.

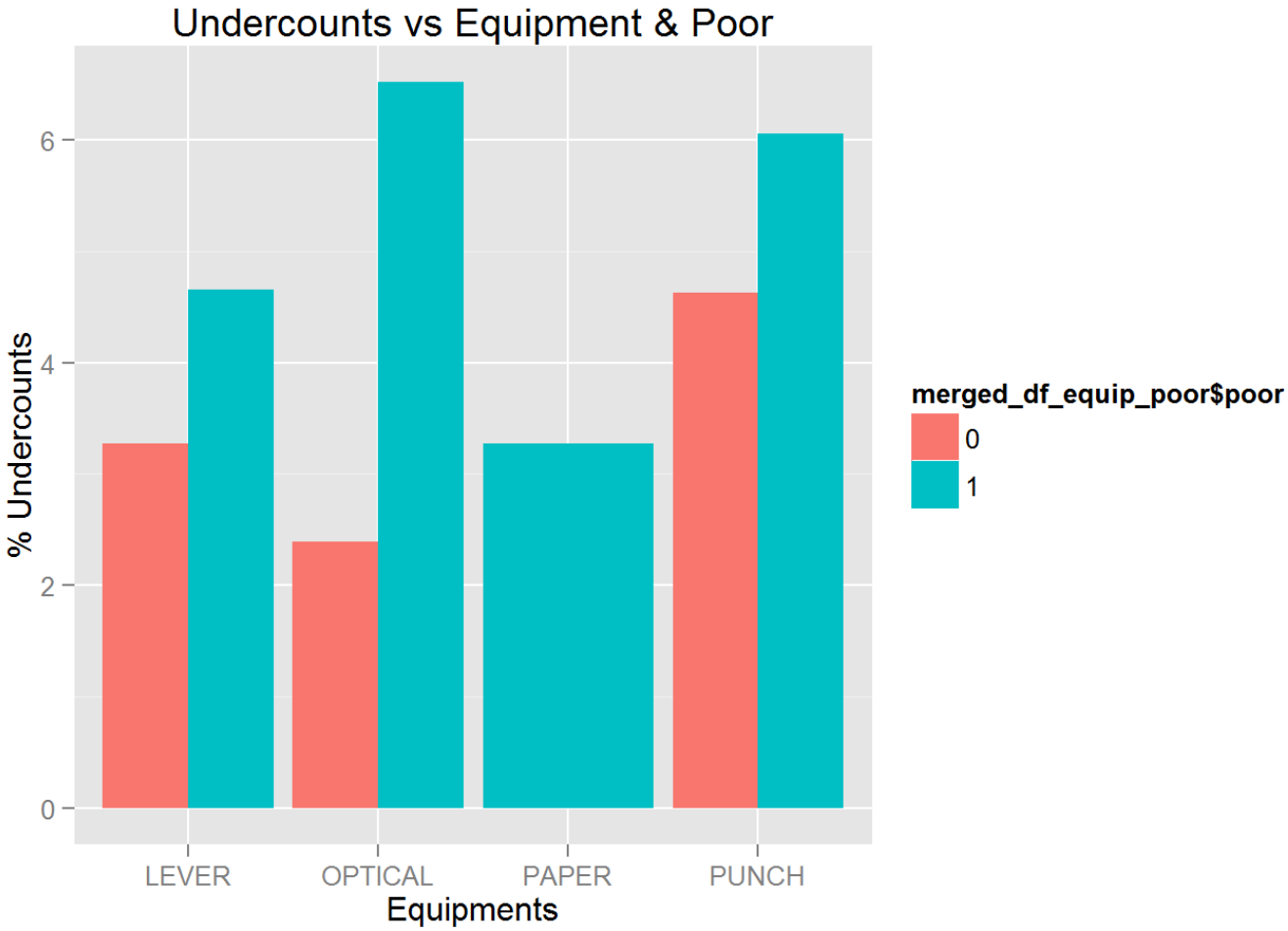
```
##Show Poor have a greater tendency to vote undercount (vote %)

ballot_poor = aggregate(ballots~poor, data=georgia_data, sum)
undercount_poor = aggregate(undercount~poor, data=georgia_data, sum)
merged_df_poor = merge(ballot_poor,undercount_poor, by ="poor")
merged_df_poor["ratio"] = merged_df_poor["undercount"]/merged_df_poor["ballots"] * 100

ggplot(merged_df_poor, aes(x = merged_df_poor$poor, y = merged_df_poor$ratio) ) + geom_ba
r(stat = "identity")+ ggtitle("Undercounts vs Poor") +  xlab ( "Poor") + ylab ("% Underco
unts")
```



The above graph shows that poor people have a greater rate of undervotes.

In order to understand the trend of which instrument has a greater tendency to cause more undercounts in the poor class, we plot it against the undercount percentage and group it by poor across the different equipment.

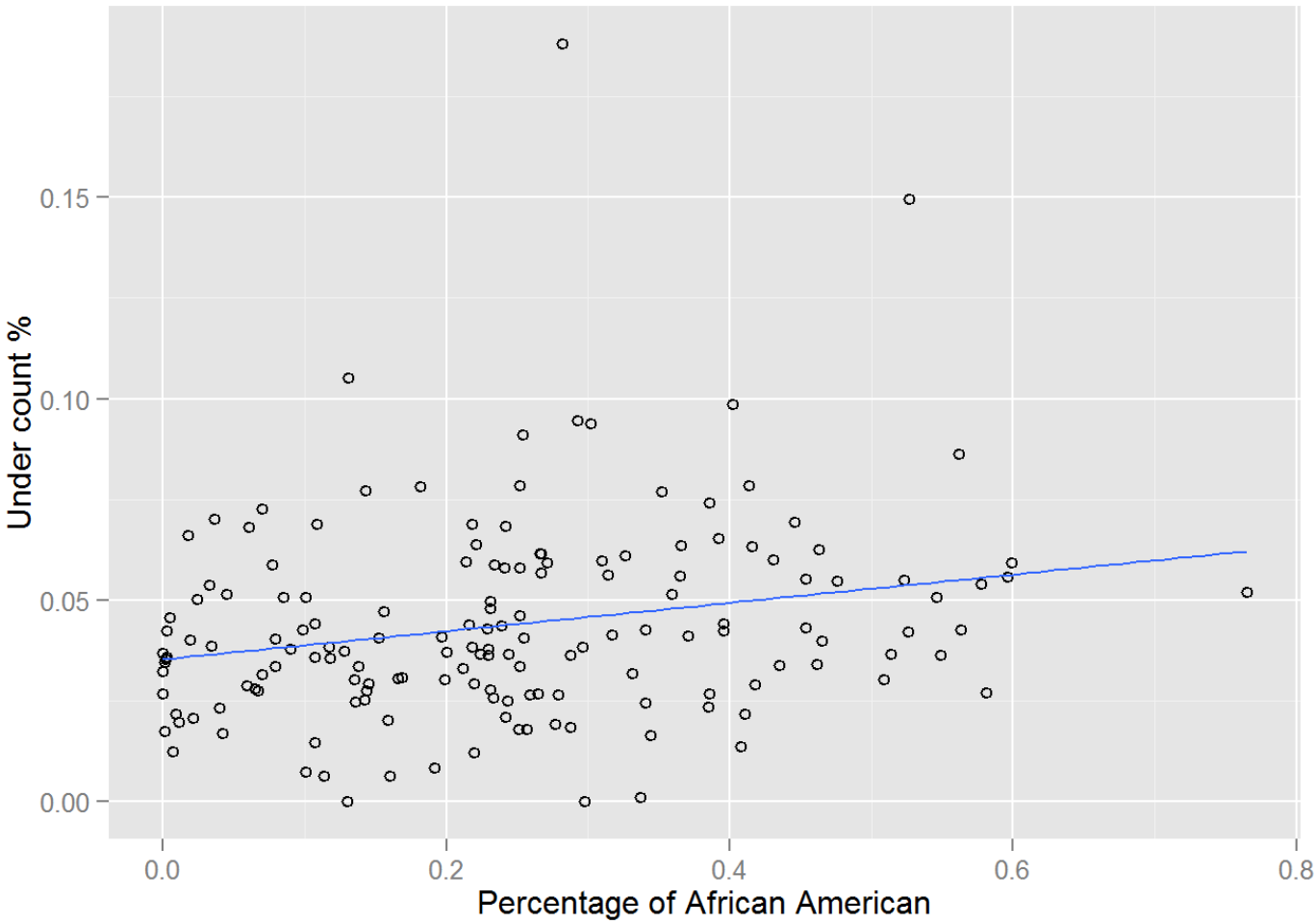
```
#To check the correlation between the poor and the equip that is used in poor major

##Show Equips and Poor that lead to more undercount vote %

ballot_equip_poor = aggregate(ballots~equip+poor, data=georgia_data, sum)
undercount_equip_poor = aggregate(undercount~equip+poor, data=georgia_data, sum)

merged_df_equip_poor = merge(ballot_equip_poor,undercount_equip_poor, by = c("equip","poo
r"))
merged_df_equip_poor["ratio"] = merged_df_equip_poor["undercount"]/merged_df_equip_poo
r["ballots"] * 100
merged_df_equip_poor$poor = factor(merged_df_equip_poor$poor)

qplot(x=merged_df_equip_poor$equip, y=(merged_df_equip_poor$undercount)*100/(merged_df_eq
uip_poor$ballots), xlab = "Equipments", ylab = "% Undercounts", fill=merged_df_equip_poo
r$poor, data=merged_df_equip_poor, geom="bar", stat="identity", position="dodge", legen
d(legend = "Poor")) + ggtitle("Undercounts vs Equipment & Poor")
```



From the above graph we see that the poor have a greater rate of undercounts in optical, which is more than the other equipments.Also one can observe that only poor people use “paper” to vote andhave recorded underount percentage in those as well.

To understand the correlation of the minority African American population on the total number of undercounts we plot a scatter plot to find if there exists any relationship

```
undercountperc = (georgia_data$ballots-georgia_data$votes)/(georgia_data$ballots)
ggplot(georgia_data, aes(y=undercountperc, x=georgia_data$perAA)) + geom_point(shape=1) +
geom_smooth(method=lm, se=FALSE) + xlab("Percentage of African American") + ylab("Under c
ount %")
```



The correlation between the points are weak suggesting a weak correlation between the number of undercounts and African American population

Bootstrapping

To assess the given ETFs, the required libraries are loaded

```
library(mosaic)
```

```
## Loading required package: car
## Loading required package: dplyr
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
##
## Loading required package: lattice
## Loading required package: mosaicData
##
## Attaching package: 'mosaic'
##
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
##
## The following object is masked from 'package:car':
##
##     logit
##
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cov, D, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
##
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

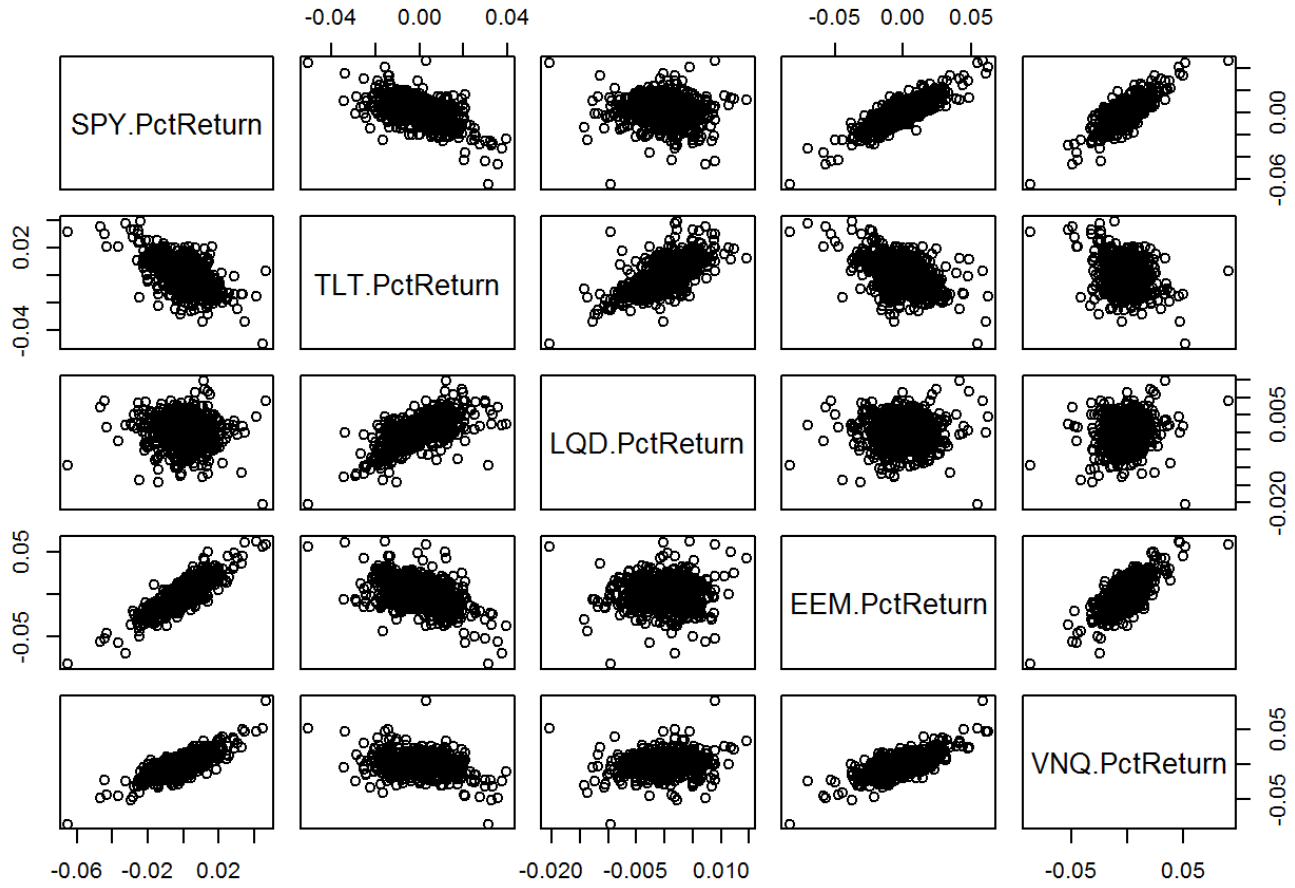
```
library(fImport)
```

```
## Loading required package: timeDate
## Loading required package: timeSeries
```

```
library(foreach)
```

A function is defined as below that calculates the return from a stock based on the change in closing price over a day. The returns are calcuated over the chosen 5 year period for the ETFs - SPY, TLT, LQD, EEM, VNQ.

```
YahooPricesToReturns = function(series) {
  mycols = grep('Adj.Close', colnames(series))
  closingprice = series[,mycols]
  N = nrow(closingprice)
  percentreturn = as.data.frame(closingprice[2:N,]) / as.data.frame(closingprice[1:(N-
1),]) - 1
  mynames = strsplit(colnames(percentreturn), '.', fixed=TRUE)
  mynames = lapply(mynames, function(x) return(paste0(x[1], ".PctReturn")))
  colnames(percentreturn) = mynames
  as.matrix(na.omit(percentreturn))
}


mystocks = c("SPY", "TLT", "LQD", "EEM", "VNQ")
myprices = yahooSeries(mystocks, from='2011-01-01', to='2015-07-30')

myreturns = YahooPricesToReturns(myprices)

pairs(myreturns)
```

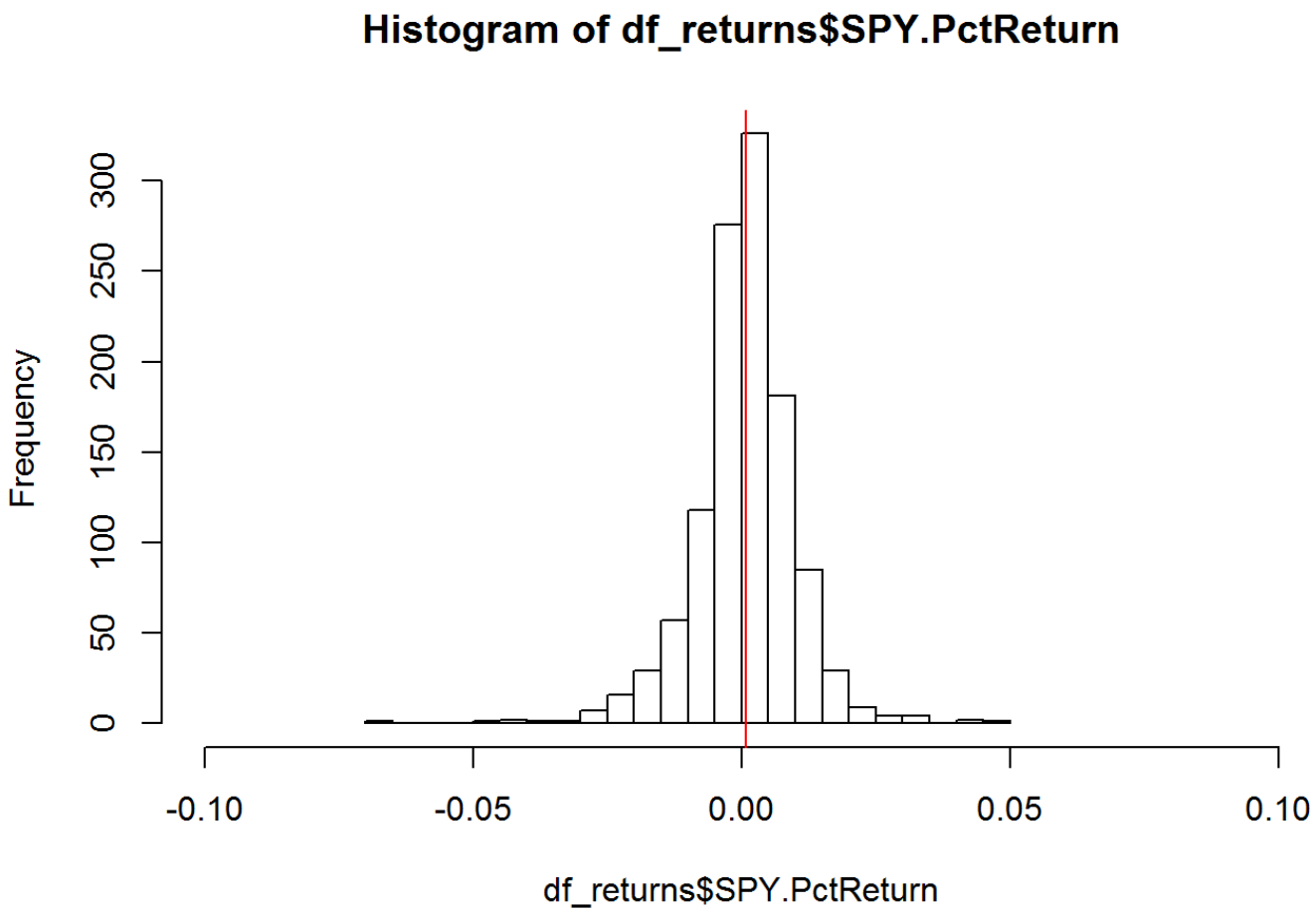


The total assets are initially taken to be $100000 and an equal split of the portfolio is used to assess the risk of each ETF.

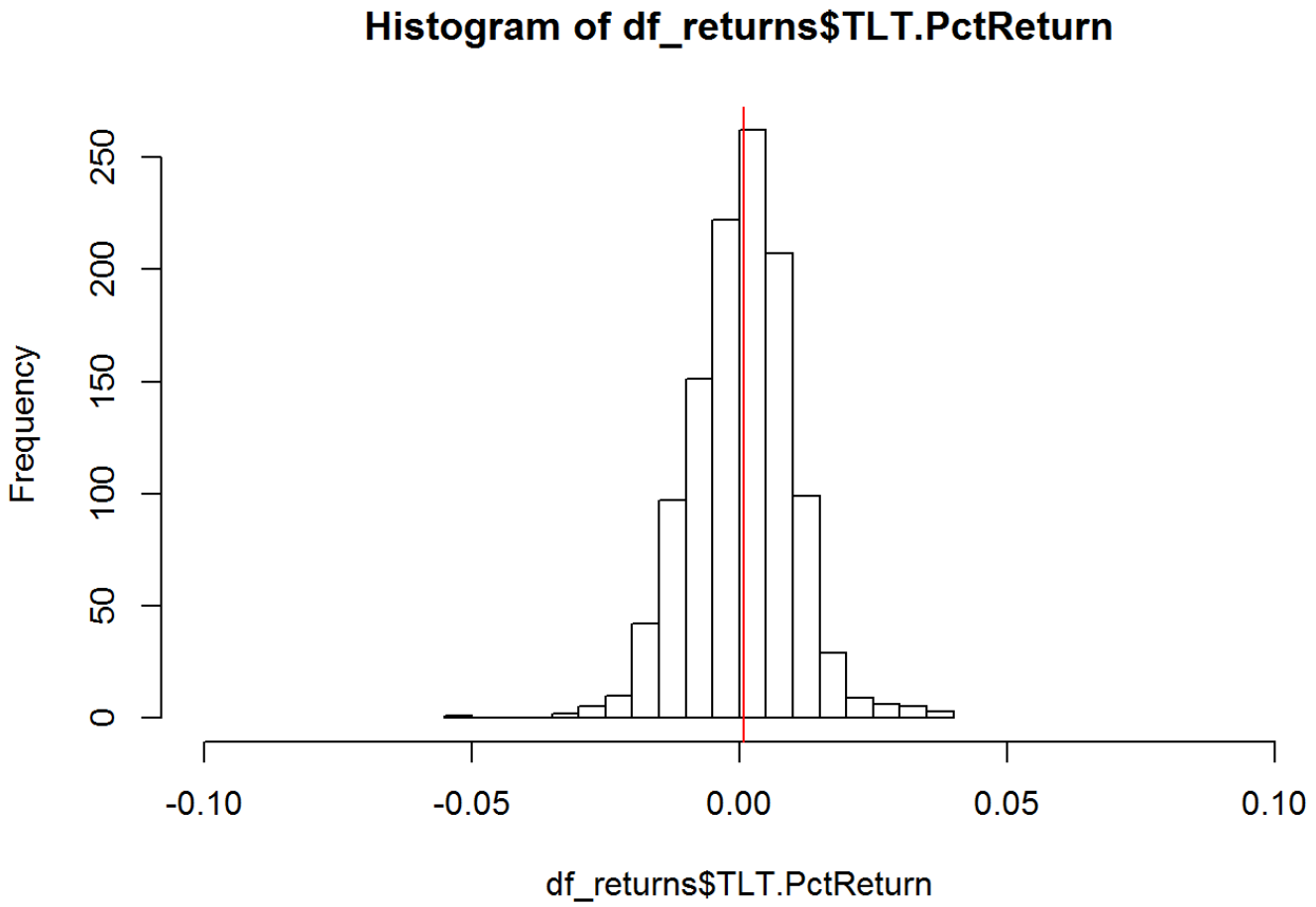
```
total_assets = 100000

weights = c(0.2, 0.2, 0.2, 0.2, 0.2)

holdings = weights * total_assets

par(mfrow = c(1,1))

#the returns for the assessts are calculated.
df_returns = data.frame(myreturns)

par(mfrow = c(5,1))
```

In order to understand the risk involved in investing in a particular ETF, we plot the histograms and the also calculate the standard deviations of each of the ETF. The wider the histogram or bigger the standard deviation, greater is the risk involved in investing in that ETF.

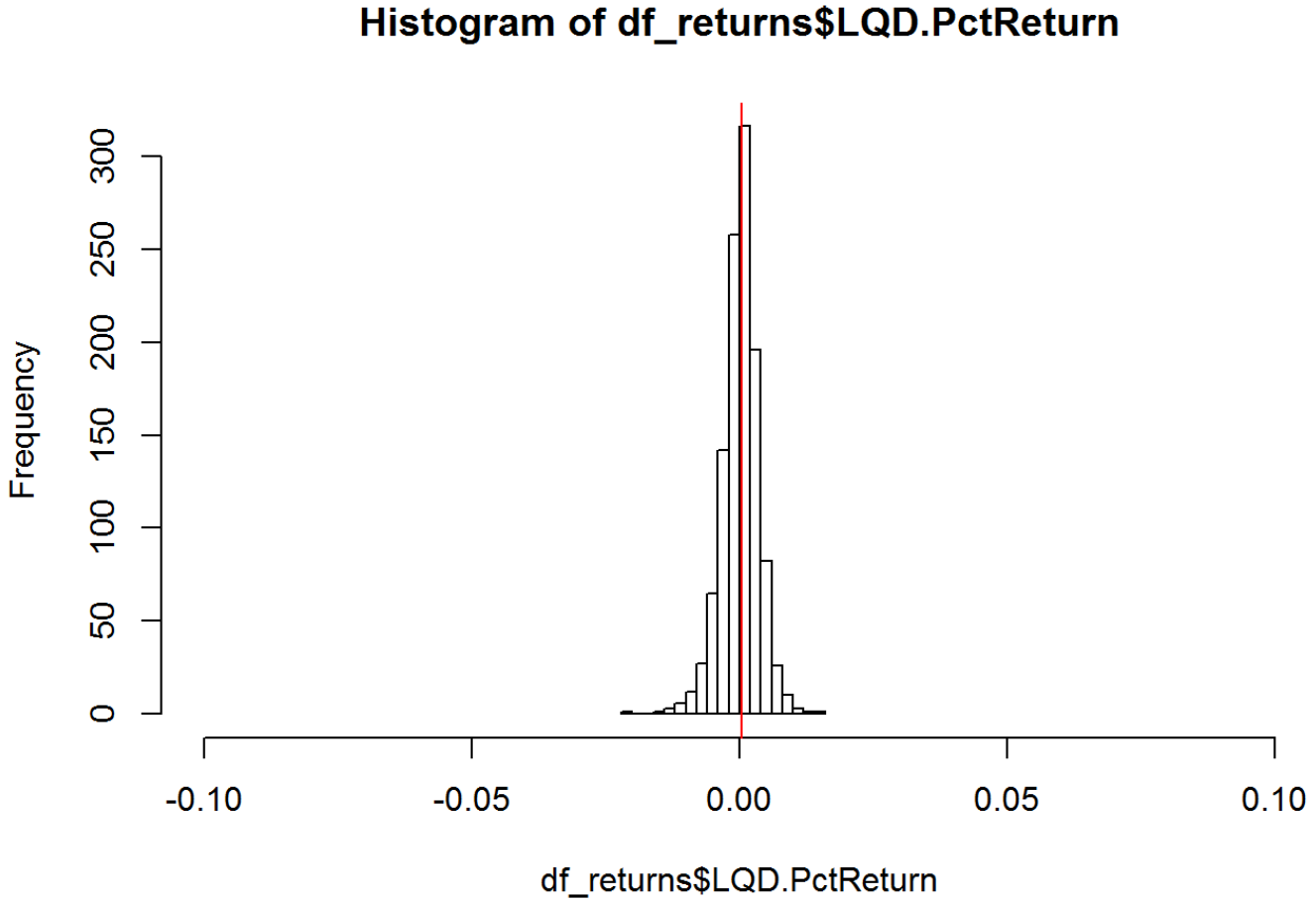
```
hist(df_returns$SPY.PctReturn, breaks = 20, xlim = c(-0.1,0.1))
abline(v=median(df_returns$SPY.PctReturn), col = "red")
```



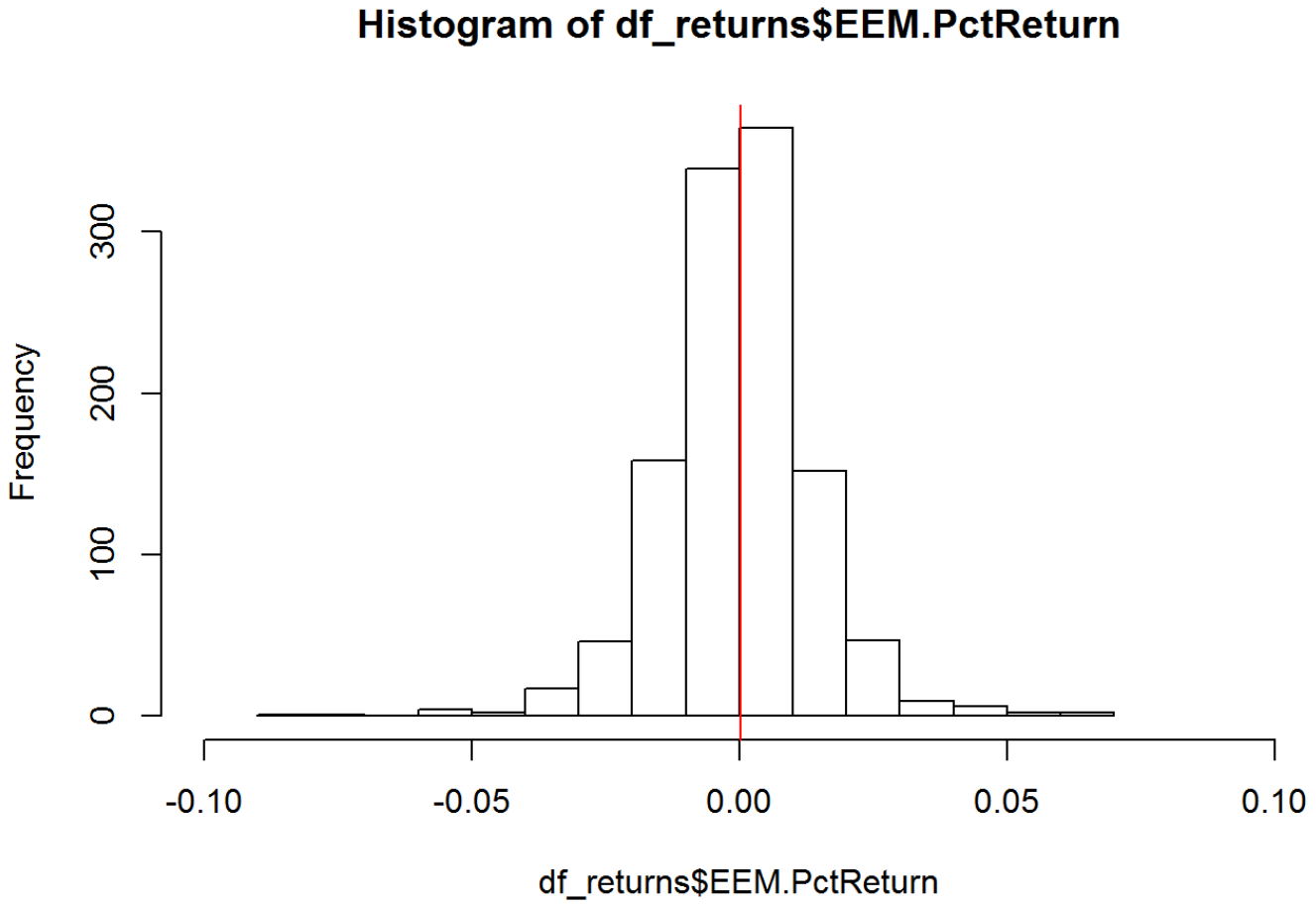
```
hist(df_returns$TLT.PctReturn, breaks = 20, xlim = c(-0.1,0.1))
abline(v=median(df_returns$TLT.PctReturn), col = "red")
```



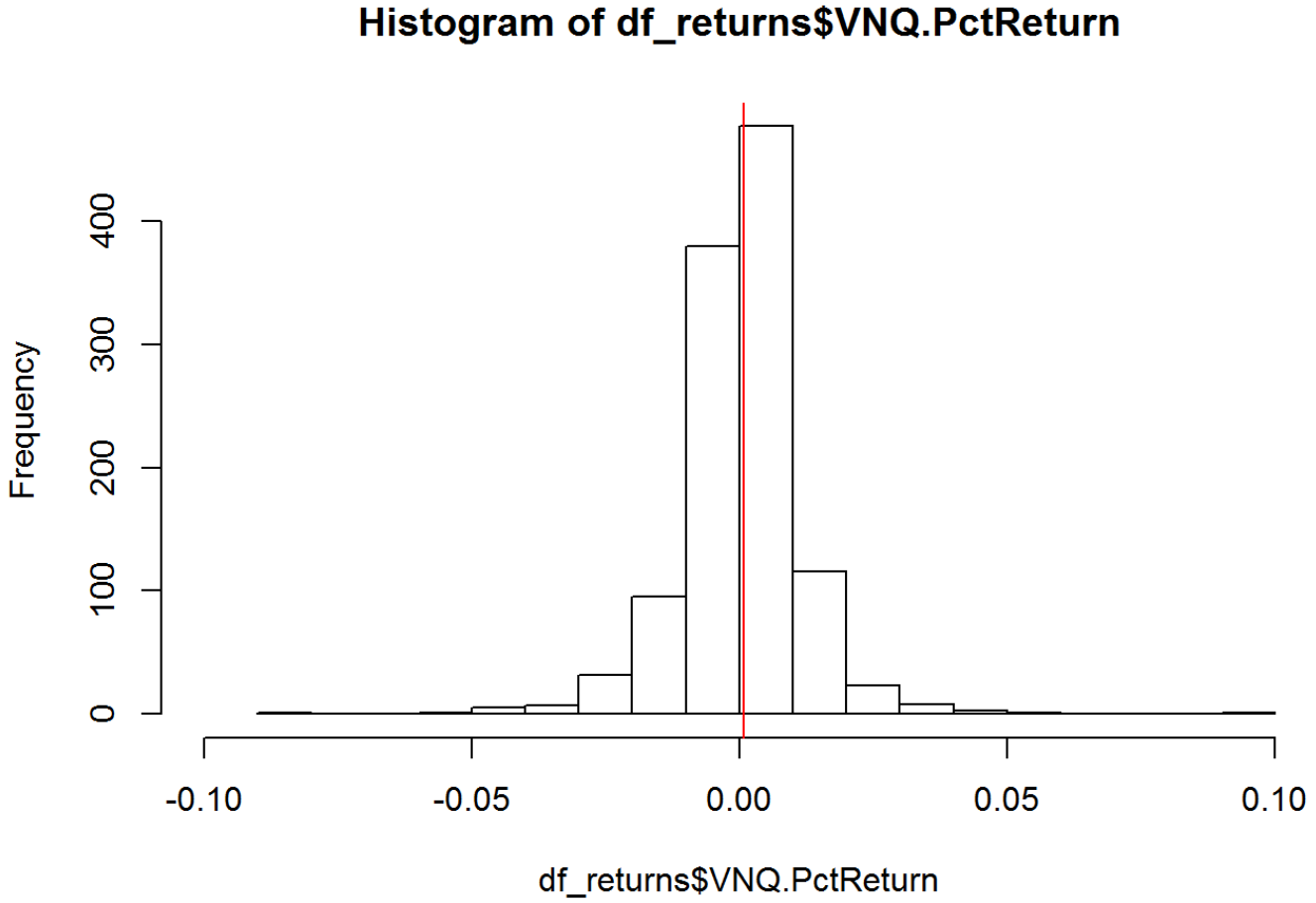

```
hist(df_returns$LQD.PctReturn, breaks = 20, xlim = c(-0.1,0.1))
abline(v=median(df_returns$LQD.PctReturn), col = "red")
```




```
hist(df_returns$EEM.PctReturn, breaks = 20, xlim = c(-0.1,0.1))
abline(v=median(df_returns$EEM.PctReturn), col = "red")
```




```
hist(df_returns$VNQ.PctReturn, breaks = 20, xlim = c(-0.1,0.1))
abline(v=median(df_returns$VNQ.PctReturn), col = "red")
```




```
summary(df_returns)
```

```
##  SPY.PctReturn        TLT.PctReturn        LQD.PctReturn
##  Min.   :-0.0651232   Min.   :-0.0504495   Min.   :-0.0205232
##  1st Qu.:-0.0037197   1st Qu.:-0.0056139   1st Qu.:-0.0017559
##  Median : 0.0007579   Median : 0.0007862   Median : 0.0004592
##  Mean   : 0.0005644   Mean   : 0.0003935   Mean   : 0.0002067
##  3rd Qu.: 0.0053892   3rd Qu.: 0.0063600   3rd Qu.: 0.0022252
##  Max.   : 0.0464992   Max.   : 0.0396555   Max.   : 0.0146677
##  EEM.PctReturn        VNQ.PctReturn
##  Min.   :-8.337e-02   Min.   :-0.0868671
##  1st Qu.:-8.291e-03   1st Qu.:-0.0048738
##  Median : 2.576e-04   Median : 0.0008889
##  Mean   :-6.086e-05   Mean   : 0.0004989
##  3rd Qu.: 7.747e-03   3rd Qu.: 0.0064609
##  Max.   : 6.240e-02   Max.   : 0.0910393
```

```
#return the standard deviation of the ETF.
sapply(df_returns, function(x) sd(x))
```

```
## SPY.PctReturn TLT.PctReturn LQD.PctReturn EEM.PctReturn VNQ.PctReturn
##   0.009420369   0.009596946   0.003486944   0.013854692   0.011457092
```

Both the histograms and the Standard Deviations are clear indicators of the obeservations made below.

We see that the standard deviation of these tickers are as - EEM>VNQ>TLT>SPY>LQD. The same trend is also observed in the histograms. The wider the plots the more risky. Therefore we can consider - Safest ETF -> LQD. Safe/relatively risky ETF -> SPY, TLT. Risky ETF -> EEM, VNQ.

Therefore shares for the risky and the non risy ones can be split based on relative riskiness of the ETF (standard deviation)

Splitting the window for the plots.

```
par(mfrow = c(3,1))
```

The equal split approach - Investing with equal share on all ETFs. Monte Carlo simulation is done to resample 20 observations (for 2 weeks)

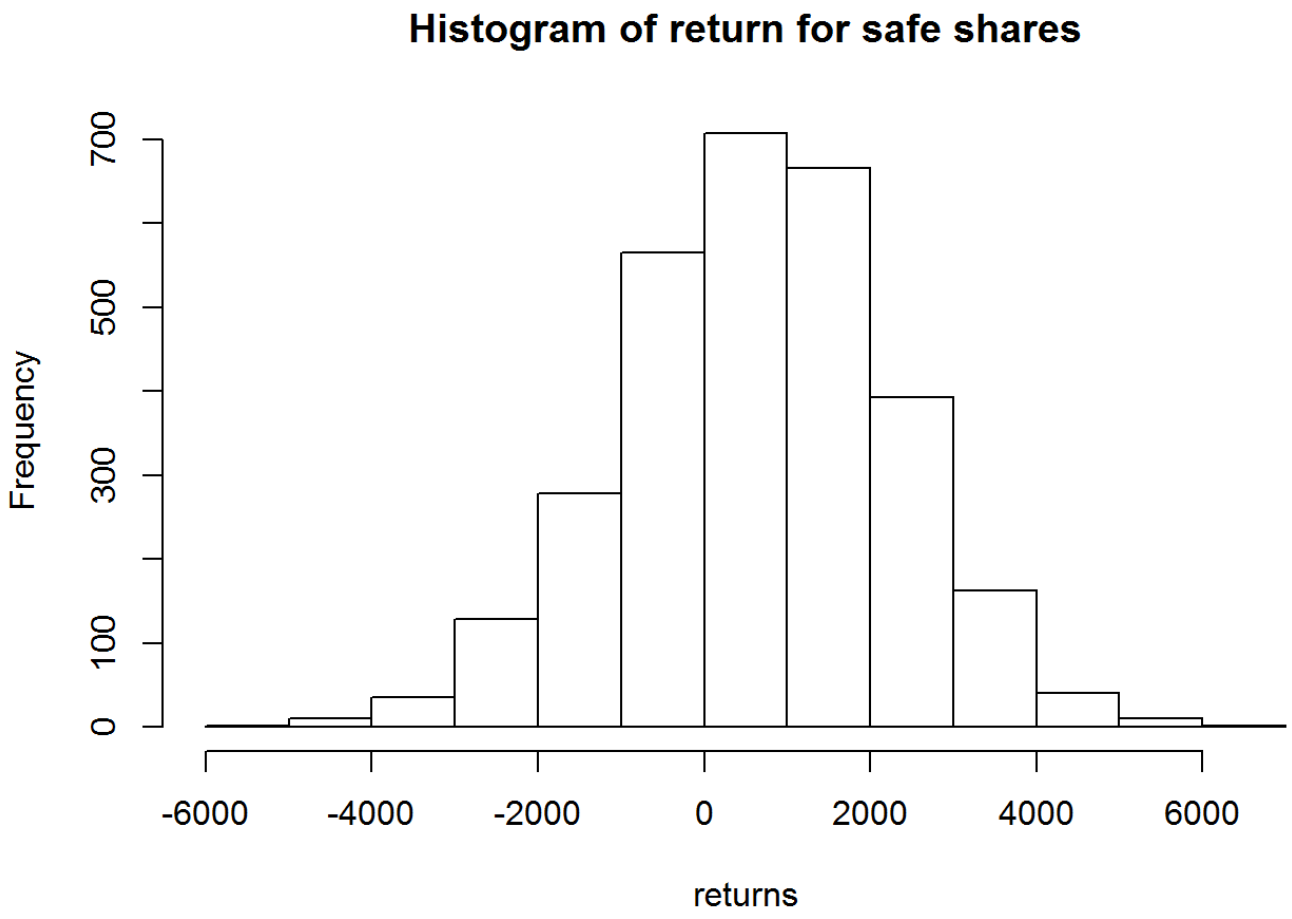
```
n_days = 20
sim1 = foreach(i=1:3000, .combine='rbind') %do% {
    total_assets = 100000
  weights = c(0.2, 0.2, 0.2, 0.2, 0.2)
    holdings = weights * total_assets
    assetstracker = rep(0, n_days) # Set up a placeholder to track total wealth
    for(today in 1:n_days) {

        return.today = resample(myreturns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_assets = sum(holdings)
        assetstracker[today] = total_assets
    }
    assetstracker
}


# Profit/loss
hist(sim1[,n_days]- 100000, main = "Histogram of return for equal shares", xlab = "return
s")
```


```
# Calculate 5% value at risk
quantile(sim1[,n_days], 0.05) - 100000
```

```
##        5%
## -3699.615
```

- The safe ETFs* The three stocks that were identified as safe or relatively less risky were split with weights of 0.25, 0.25, 0.5 based on their standard deviation.

```
n_days = 20
sim2 = foreach(i=1:3000, .combine='rbind') %do% {
    total_assets = 100000
  weights_safe = c(0.25, 0.25, 0.5, 0, 0)
    holdings = weights_safe * total_assets
    assetstracker_safe = rep(0, n_days) # Set up a placeholder to track total wealth
    for(today in 1:n_days) {

        return.today = resample(myreturns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_assets = sum(holdings)
        assetstracker_safe[today] = total_assets
    }
    assetstracker_safe
}

# Profit/loss
hist(sim2[,n_days]- 100000, main = "Histogram of return for safe shares", xlab = "return
s")
```

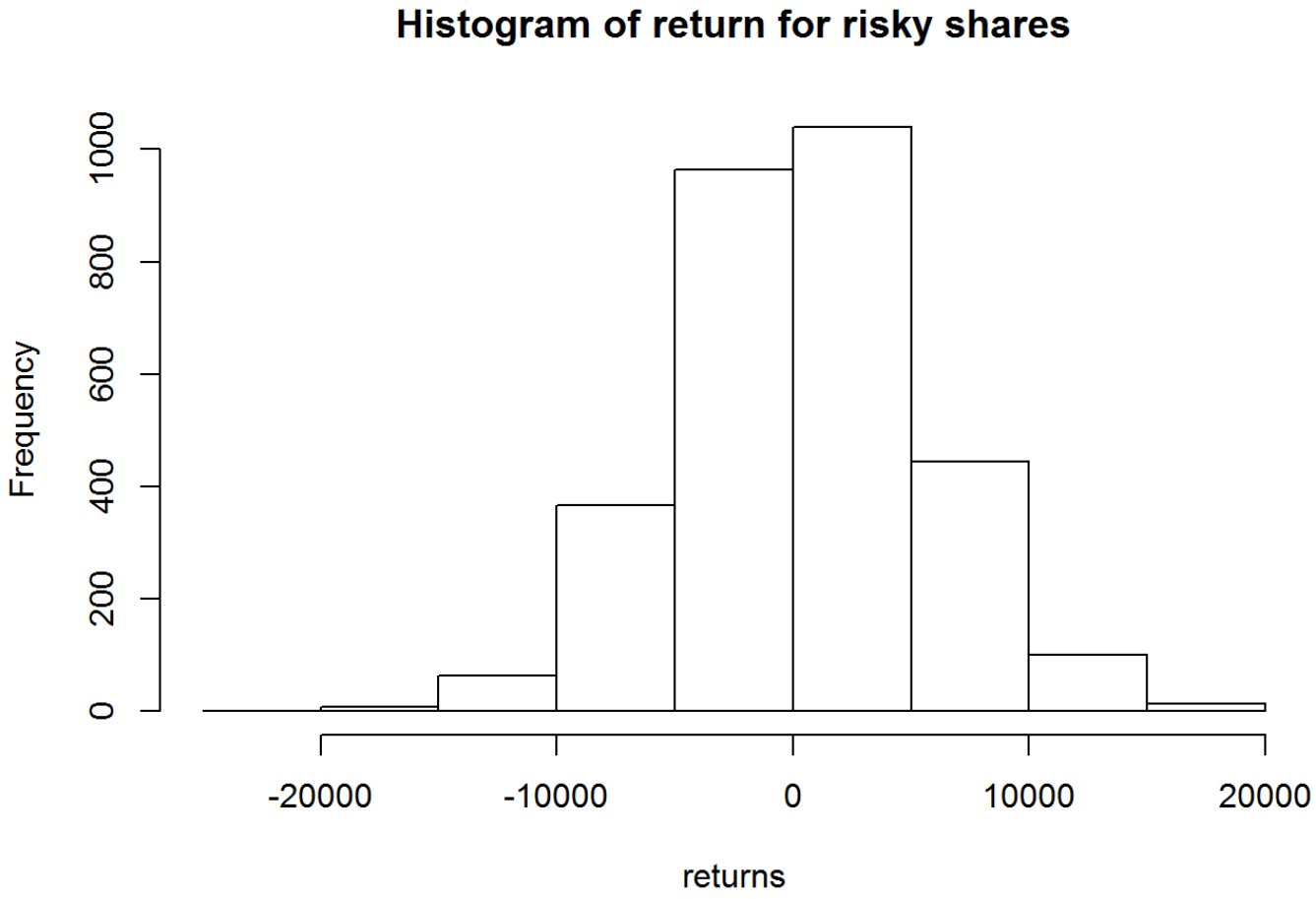



```
# Calculate 5% value at risk
quantile(sim2[,n_days], 0.05) - 100000
```

```
##        5%
## -2120.784
```

- The risky ETFs* The risky stocks that were identified were given weights of 0.5, 0.5 based on their standard deviation.

```
n_days = 20
sim3 = foreach(i=1:3000, .combine='rbind') %do% {

  total_assets = 100000
  weights_risky = c(0, 0, 0, 0.5, 0.5)
    holdings = weights_risky * total_assets
    assetstracker_risky = rep(0, n_days) # Set up a placeholder to track total wealth
    for(today in 1:n_days) {
        return.today = resample(myreturns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_assets = sum(holdings)
        assetstracker_risky[today] = total_assets
    }
    assetstracker_risky
}

# Profit/loss
hist(sim3[,n_days]- 100000, main = "Histogram of return for risky shares", xlab = "return
s")
```




```
# Calculate 5% value at risk
quantile(sim3[,n_days], 0.05) - 100000
```

```
##        5%
## -8022.335
```

From the plots of the histograms of returns over the 2 week period above and the risk calculated at 5% level one could identify that the risk involved in investing in ETFs - EEM (most risky), VNQ is higher than investing in LQD (safest), SPY, TLT.

The safest stock has risk at 5% estimate < equal portfolio < risky portfolio

** Clustering and PCA **

The given data - wine.csv is loaded.

```
library(ggplot2)

wine_data = read.csv("C:/Users/Vijai/OneDrive/Github/STA380-04082015/STA380-master/data/w
ine.csv")
head(wine_data)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                   34  0.9978 3.51      0.56     9.4
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
## 5                  11                   34  0.9978 3.51      0.56     9.4
## 6                  13                   40  0.9978 3.51      0.56     9.4
##   quality color
## 1       5   red
## 2       5   red
## 3       5   red
## 4       6   red
## 5       5   red
## 6       5   red
```

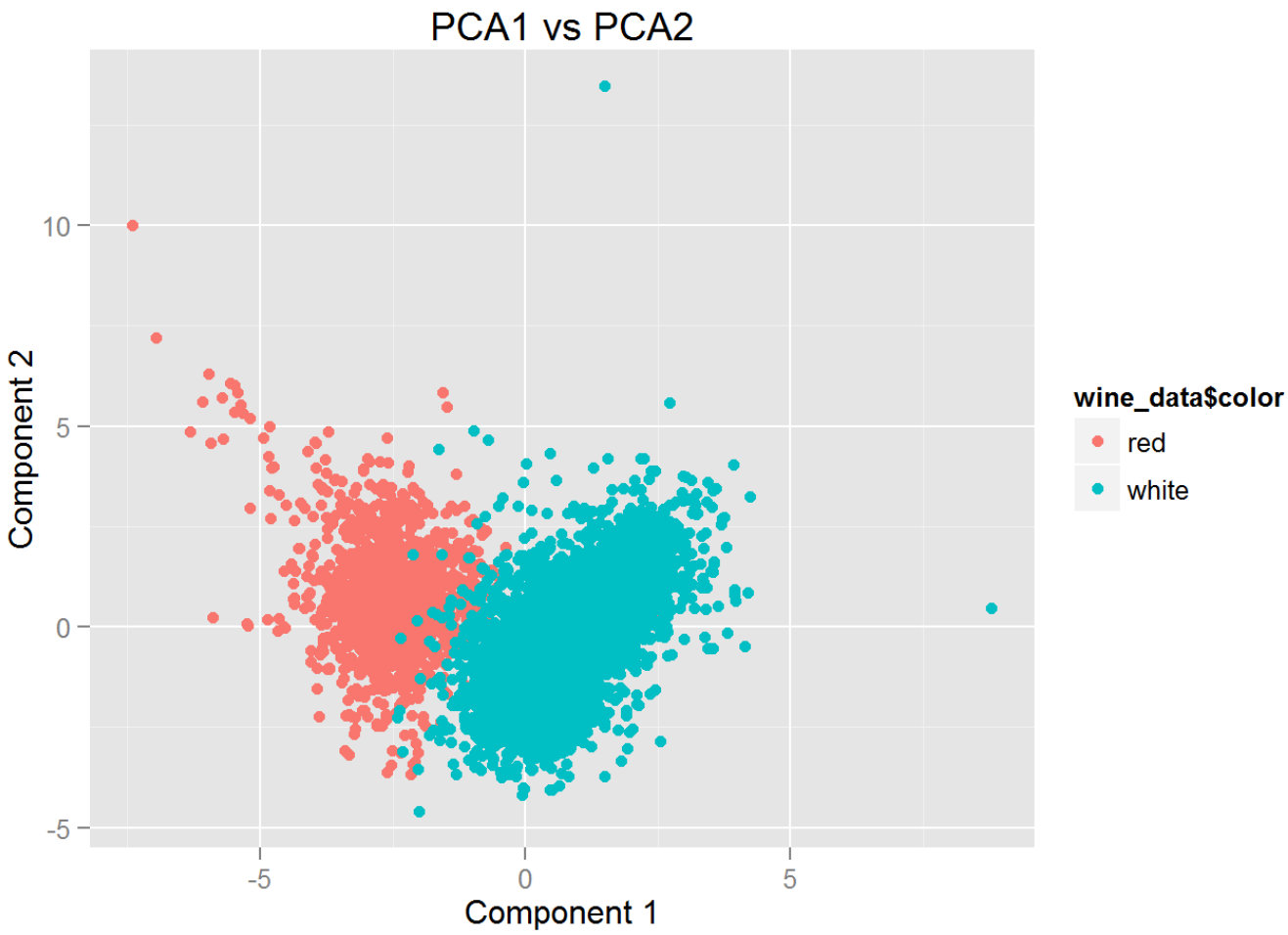
Trimming the data off qualty and type of wine.

```
wine_trim = wine_data[,c(-12,-13)]

#getting a sense of the data
head(wine_trim)
```

```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                   34  0.9978 3.51      0.56     9.4
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
## 5                  11                   34  0.9978 3.51      0.56     9.4
## 6                  13                   40  0.9978 3.51      0.56     9.4
```

Scaling the numeric variables and applying the Principal Component Analysis on the data frame. The scores and loadings are stored afte PCA. The scores help in identifying the segments in the data when plotted with the color of the wine that is given in the data.

```
# Run PCA
pc1 = prcomp(wine_trim, scale=TRUE)
loadings = pc1$rotation
scores = pc1$x

qplot(scores[,1], scores[,2], color=wine_data$color, xlab='Component 1', ylab='Component
2') + ggtitle ("PCA1 vs PCA2")
```



The plots of the two component (PC1, PC2) scores and colored by the type of wine gives a clear distinction of tTwo groups/segmentations along component 1. Therefore PC1 is a good component to identify the data where the segmentation of wine are distinct.

To analyze the relative importance of the varriables we retrieve the top loadings from the PCA. This

```
o1 = order(loadings[,1], decreasing = T)
colnames(wine_trim)[head(o1,3)]
```

```
## [1] "total.sulfur.dioxide" "free.sulfur.dioxide"  "residual.sugar"
```

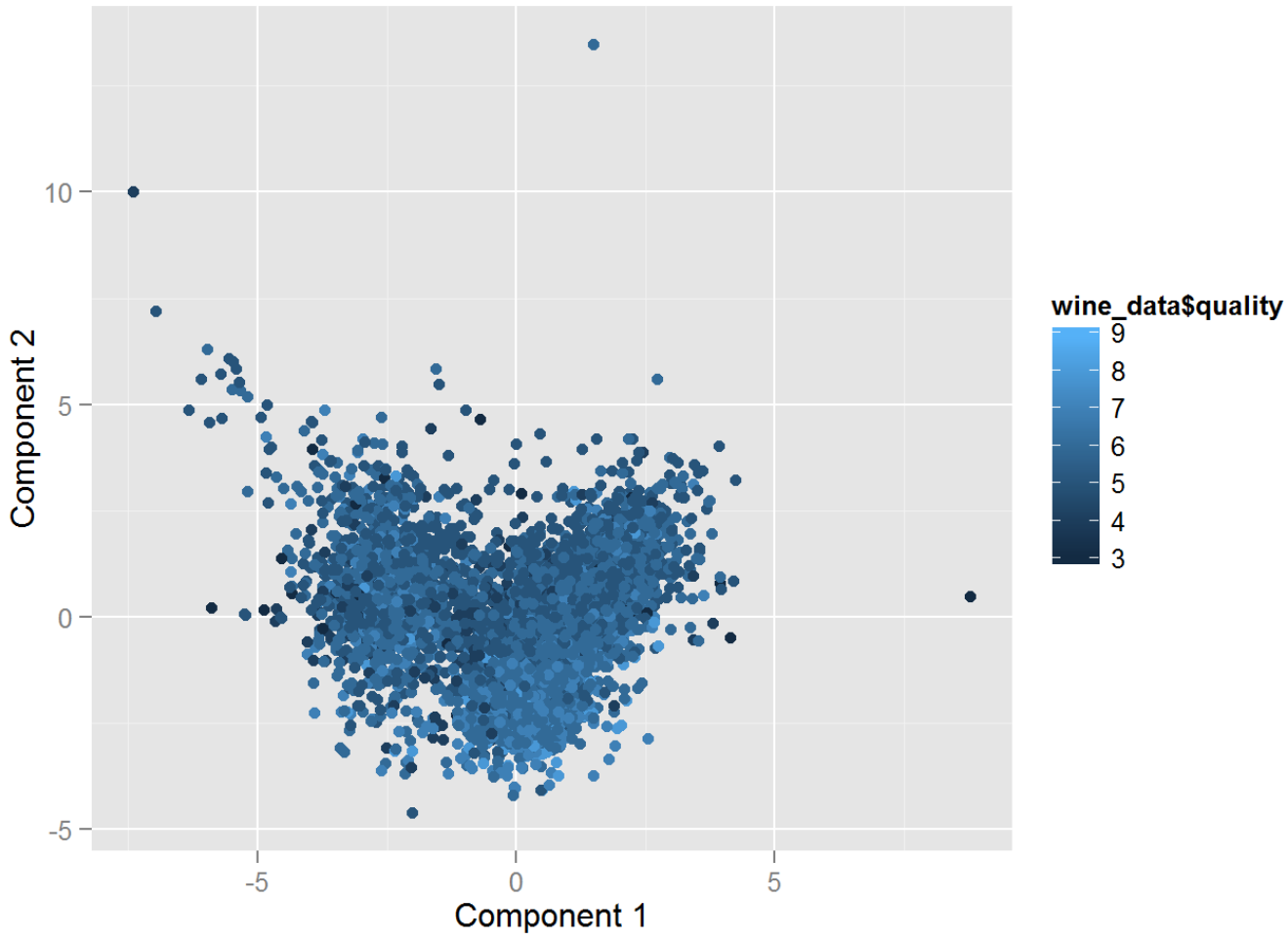
```
colnames(wine_trim)[tail(o1,3)]
```

```
## [1] "chlorides"        "sulphates"        "volatile.acidity"
```

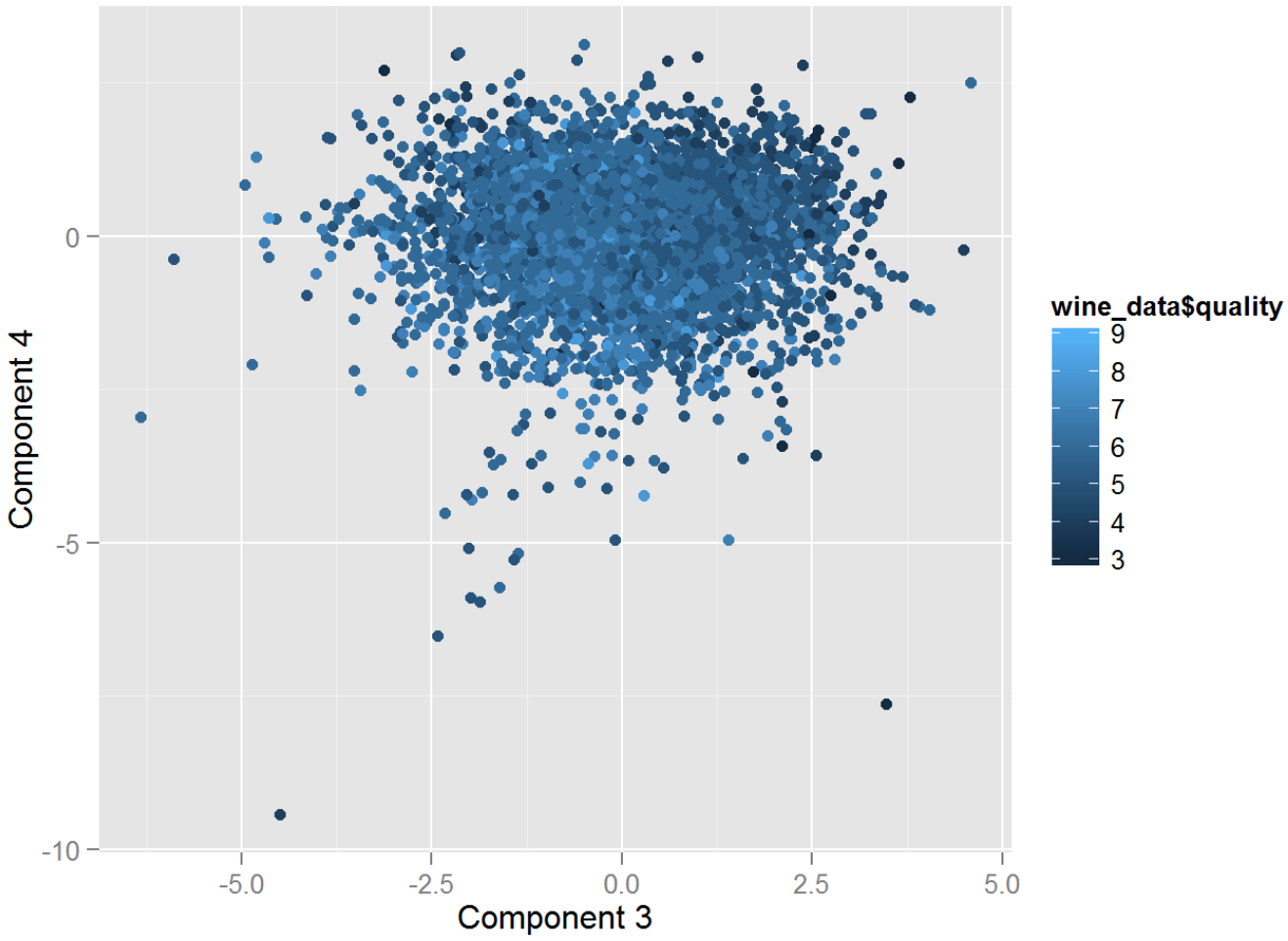
After dimension reduction we see that there are 6 factors that have been identified as important and from the above output indicates that total sulphur dioxide, free sulphur dioxide, residual sugar are significant indicators of the color of wine.

To investigate whether PCA provides any help in identifying/segmenting the quality of wine we plot a scatter plot of different components. We observe that the plots do not clearly identify any clear segments and hence we could conclude that PCA does not help us much in investigating the quality of the wine.

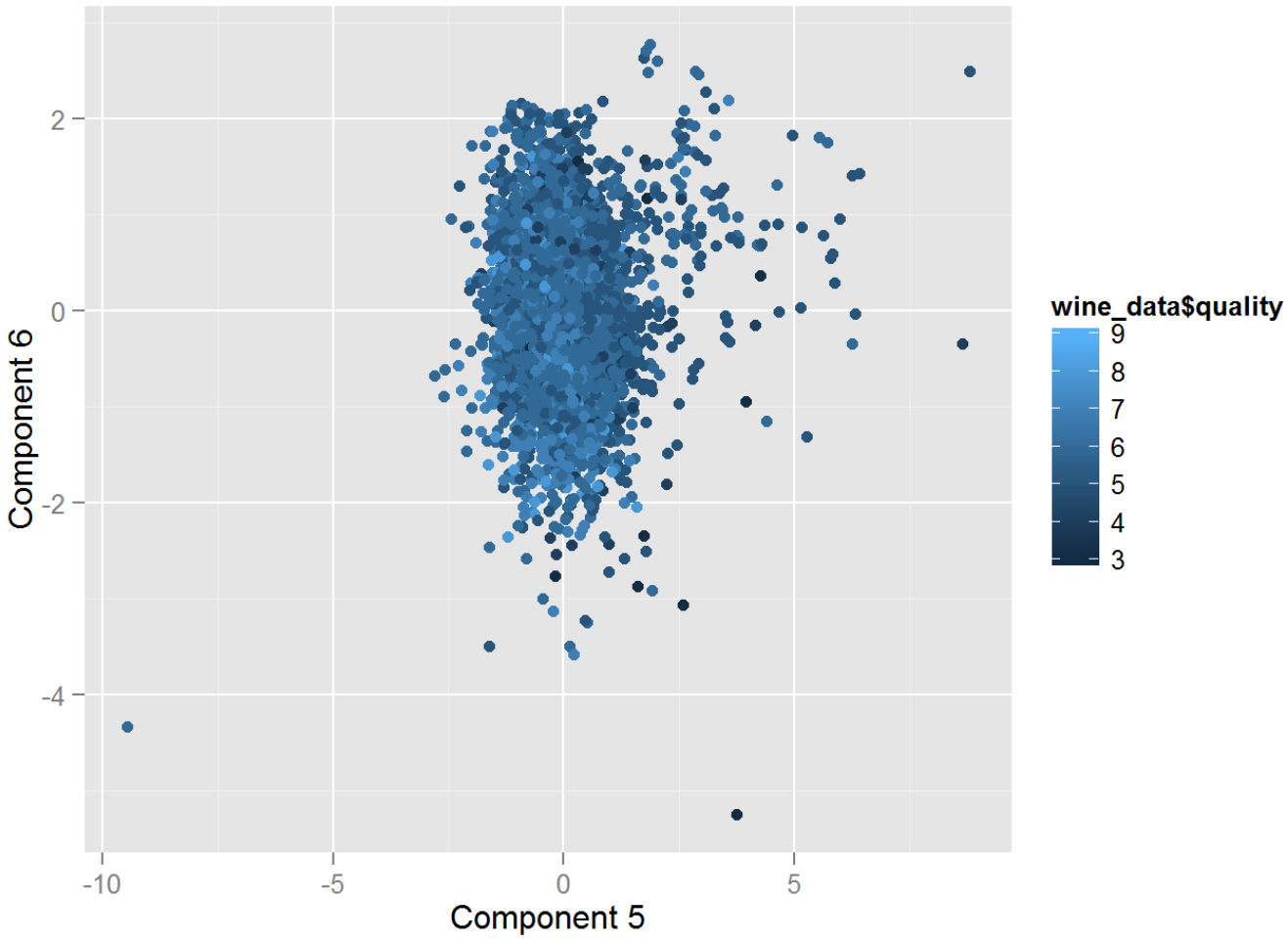
```
qplot(scores[,1], scores[,2], color=wine_data$quality, xlab='Component 1', ylab='Componen
t 2')
```



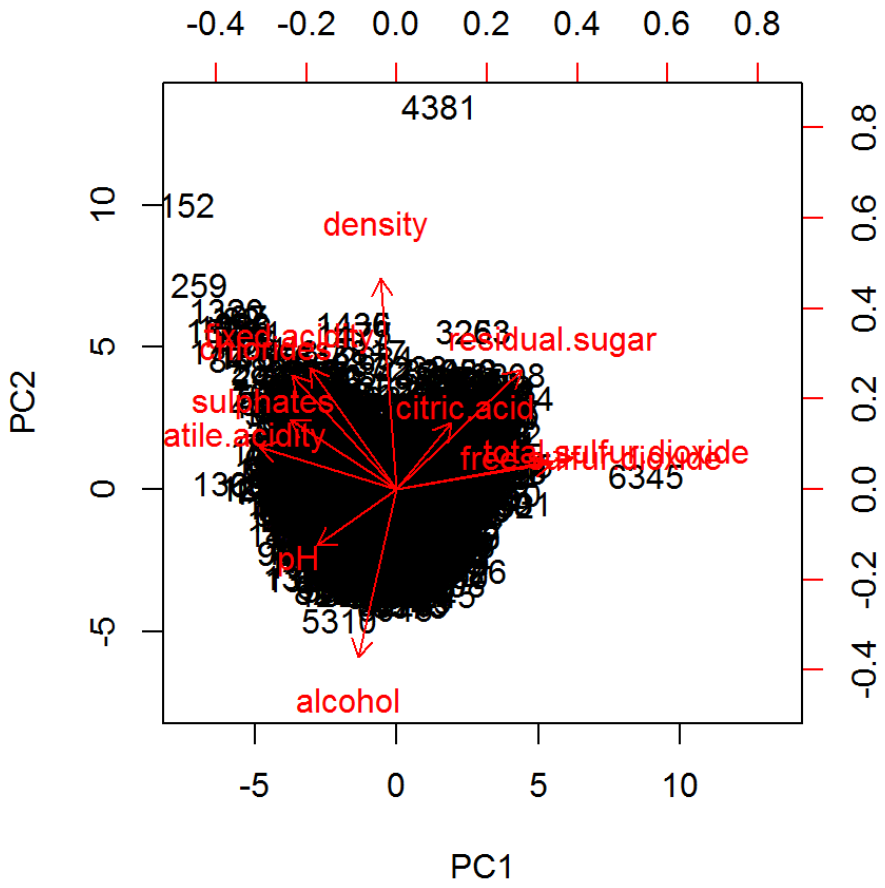

```
qplot(scores[,3], scores[,4], color=wine_data$quality, xlab='Component 3', ylab='Componen
t 4')
```




```
qplot(scores[,5], scores[,6], color=wine_data$quality, xlab='Component 5', ylab='Componen
t 6')
```




```
##There is not sufficient information that the PCA provides to segment quality.

biplot(pc1, scale=0)
```




```
#Checkng the variability explained by each PCA
pc1$sdev^2
```

```
##  [1] 3.0298686 2.4938260 1.5563470 0.9705521 0.7198749 0.6073117 0.5231588
##  [8] 0.5015103 0.3370240 0.2276958 0.0328308
```

We use K Means to investigate the same data set to see if the clustering technique provides a better segmentation for color and quality.

Scaling the data to plot using k means.

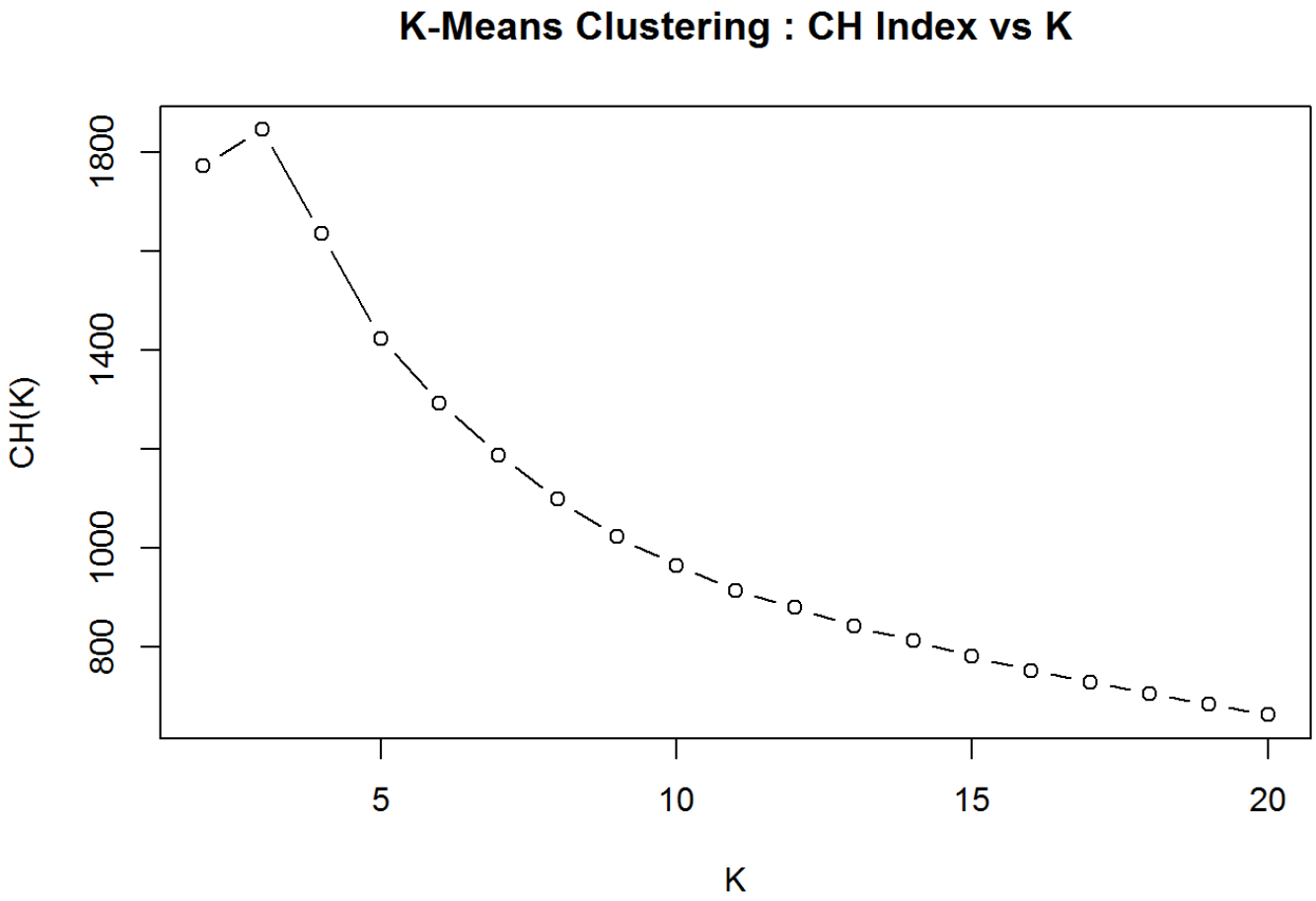
```
wine_scaled <- scale(wine_trim, center=TRUE, scale=TRUE)
```

To compute the ideal value of k we run a loop through different values of k and identify the one that we would need by calculating the highest CH index.

```
#To compute the value of k
n= dim(wine_scaled)[1]
ch = numeric(length=19)

length(ch)
```

```
## [1] 19
```

```
options(warn=-1)

for(i in 2:20){
  set.seed = 10
  kmean = kmeans(wine_scaled, centers=i, nstart=50)
  ch[i-1] = (sum(kmean$betweenss)/(i-1))/(sum(kmean$withinss)/(n-i))
}

plot(2:20, ch, xlab='K', ylab='CH(K)', type='b', main='K-Means Clustering : CH Index vs
K' )
```




```
#peaks at 3
```

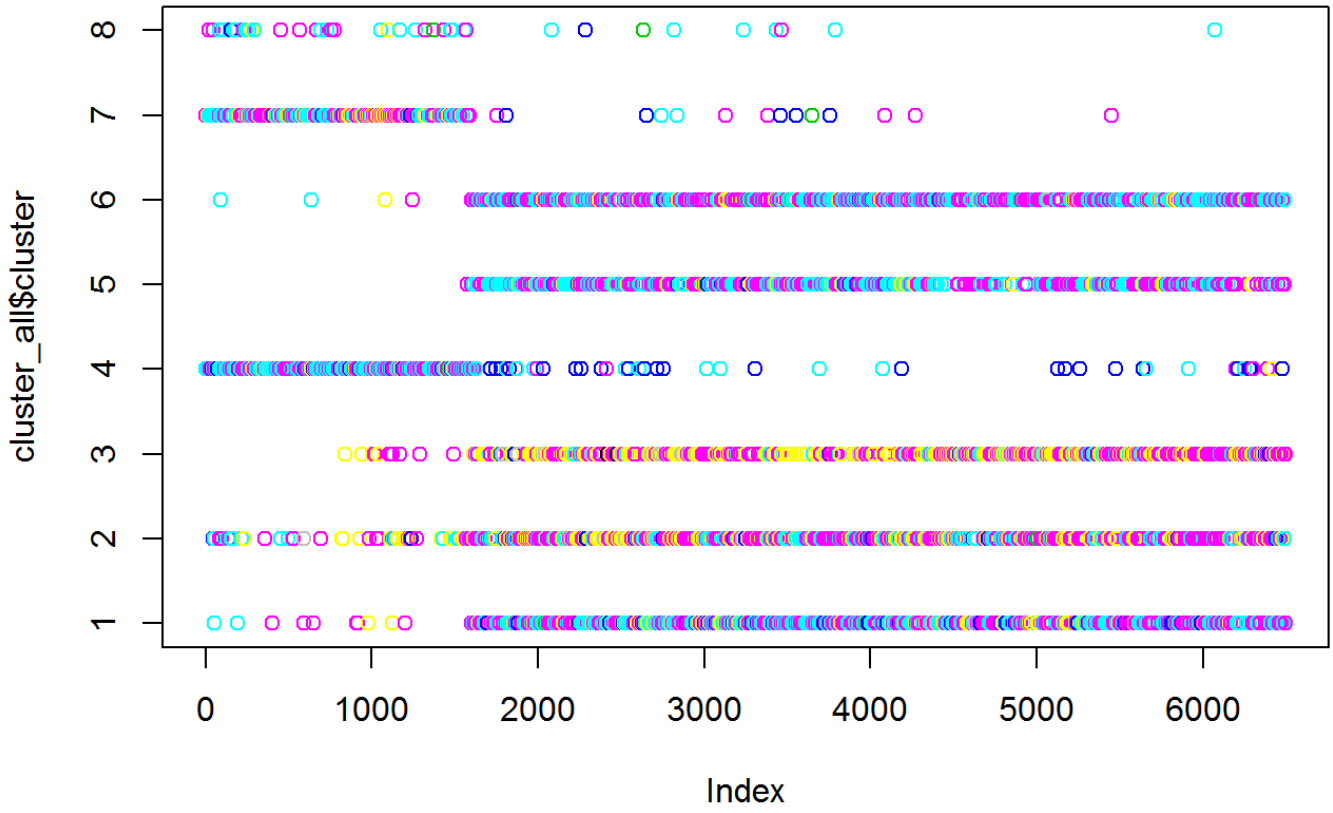
We find from the graph that the value of CH index decreases sharply until 8 and then steadies. This is differenct from the 2 segments that one could identify as the red and white wine. But the CH Index decreases rapidly for higher values of k. This is also an indication of a segmentation due to quality.

We run the kmeans for a value of 8 as below.

```
cluster_all <- kmeans(wine_scaled, centers=8, nstart=50)
names(cluster_all)
```

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

```
plot(cluster_all$cluster, col = wine_data$quality)
```



The above observations indicate that PCA is a better dimension reduction method as it reduces the number of features and hence the complexity of the problem and also helps in visualizing the distinct segments that we would have in terms of the type of the wine. Visualizing the quality with the plot using above we see that the quality and the clusters are not indicative of the segments as they do not have the same color in a cluster.

** Market Segmentation **

The data for market segmentation is loaded from social_marketing.csv

```
##Social Marketing

par(mfrow=c(1,1))

sm = read.csv("C:/Users/Vijai/OneDrive/Github/STA380-04082015/STA380-master/data/social_m
arketing.csv")

head(sm)
```

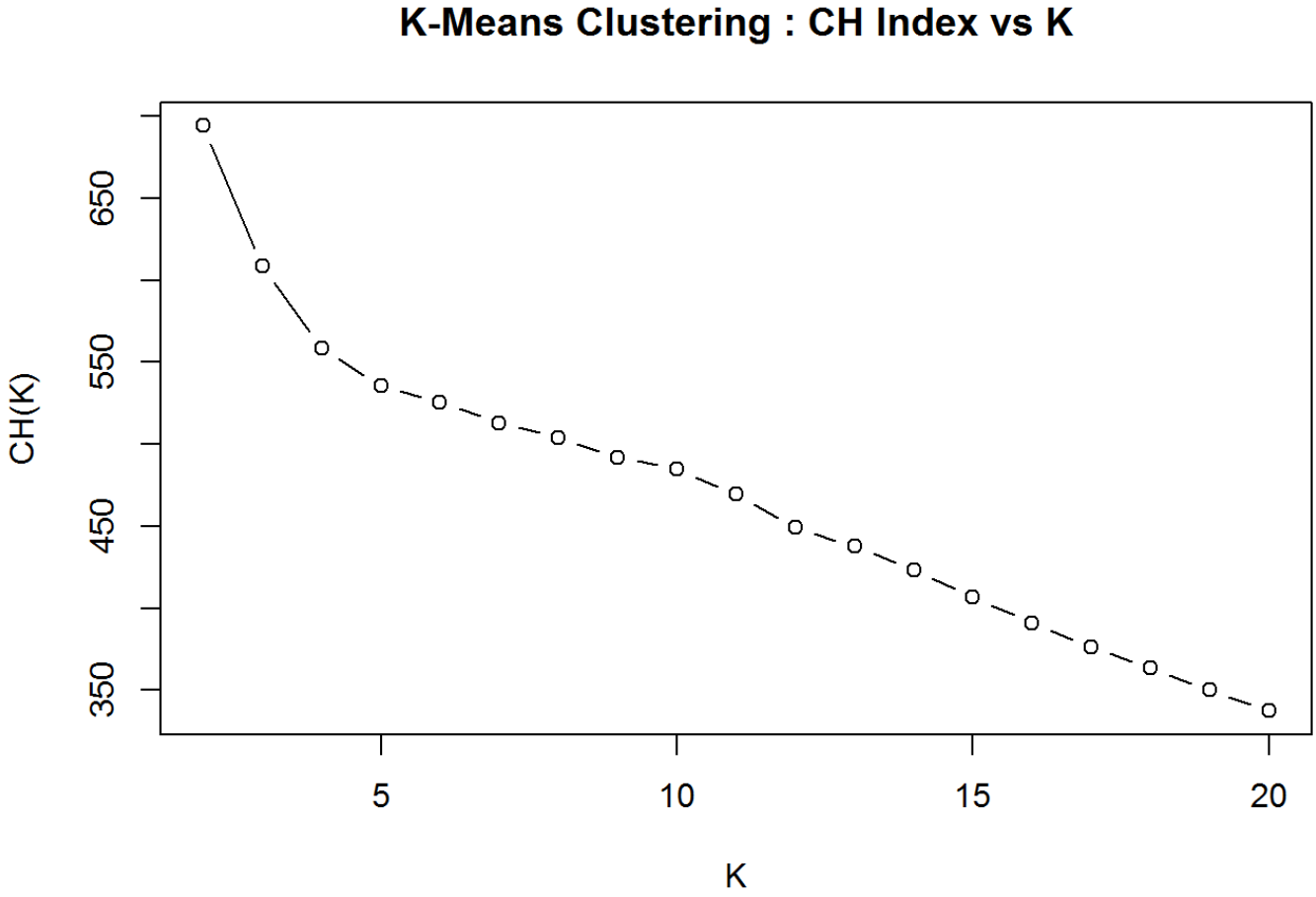
```
##           X chatter current_events travel photo_sharing uncategorized
## 1 hmjoe4g3k       2              0      2             2             2
## 2 clk1m5w8s       3              3      2             1             1
## 3 jcsovtak3       6              3      4             3             1
## 4 3oeb4hiln       1              5      2             2             0
## 5 fd75x1vgk       5              2      0             6             1
## 6 h6nvj91yp       6              4      2             7             0
##   tv_film sports_fandom politics food family home_and_garden music news
## 1       1             1        0    4      1               2     0    0
## 2       1             4        1    2      2               1     0    0
## 3       5             0        2    1      1               1     1    1
## 4       1             0        1    0      1               0     0    0
## 5       0             0        2    0      1               0     0    0
## 6       1             1        0    2      1               1     1    0
##   online_gaming shopping health_nutrition college_uni sports_playing
## 1             0        1               17           0              2
## 2             0        0                0           0              1
## 3             0        2                0           0              0
## 4             0        0                0           1              0
## 5             3        2                0           4              0
## 6             0        5                0           0              0
##   cooking eco computers business outdoors crafts automotive art religion
## 1       5   1         1        0        2      1          0   0        1
## 2       0   0         0        1        0      2          0   0        0
## 3       2   1         0        0        0      2          0   8        0
## 4       0   0         0        1        0      3          0   2        0
## 5       1   0         1        0        1      0          0   0        0
## 6       0   0         1        1        0      0          1   0        0
##   beauty parenting dating school personal_fitness fashion small_business
## 1      0         1      1      0               11       0              0
## 2      0         0      1      4                0       0              0
## 3      1         0      1      0                0       1              0
## 4      1         0      0      0                0       0              0
## 5      0         0      0      0                0       0              1
## 6      0         0      0      0                0       0              0
##   spam adult
## 1    0     0
## 2    0     0
## 3    0     0
## 4    0     0
## 5    0     0
## 6    0     0
```

Some preprocessing of data is done to remove any columns/rows that would not be right inifluencers of the segmentation process. For example we would not need the uncategorized column as this does not convey a category for segmenting the data.

```
##remove the rows that are spam and adult as we do not our data to be influenced by these observations.

sm=sm[-(sm$spam >= 0 & sm$adult>=0),]

##also remove uncategorized column as this column does not categorize the data

sm = sm[,-6]
```

The CH Index is calculated to indentify the k to be used for k means clustering process

```
#Identify the number of clusters using kmeans CH Index

sm_scaled <- scale(sm[,-1], center=TRUE, scale=TRUE)

sm_trim = sm[,-1]

#To compute the value of k
n= dim(sm_scaled)[1]
ch = numeric(length=20)

set.seed = 1234

for(i in 2:20){

  kmean = kmeans(sm_scaled, centers=i, nstart=50)
  ch[i-1] = (sum(kmean$betweenss)/(i-1))/(kmean$tot.withinss/(n-i))
}

plot(2:20, ch[1:19], xlab='K', ylab='CH(K)', type='b', main='K-Means Clustering : CH Inde
x vs K' )
```



From the above graph we see that the knee of the graph lies close to 6. Hence choosing k = 6 and running k means. Retrieve the clusters after k means and compare with the original data frame by subsetting the dataframes and identifying the significant features in each cluster.

```
set.seed = 1112

cluster_all <- kmeans(sm_scaled, centers=6, nstart=50)
names(cluster_all)
```

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

```
cluster1 = cluster_all$cluster

sm_trim = sm

sm_trim$cluster <- cluster1
```

For Cluster 1

```
cluster1 = subset(sm_trim,cluster == 1)

tail(sort(sapply(cluster1[,-37],mean)))
```

```
##        school     parenting       chatter          food      religion
##      2.708661      4.061680      4.269029      4.566929      5.261155
## sports_fandom
##      5.900262
```

The cluster is a group of younger people who would be interested in sports, online_gaming and college goers.

Cluster 2

```
cluster2 = subset(sm_trim,cluster == 2)

tail(sort(sapply(cluster2[,-37],mean)))
```

```
## health_nutrition           travel         shopping   current_events
##         1.094460         1.099537         1.276539         1.445156
##    photo_sharing          chatter
##         2.282940         4.315383
```

We find that the data shows features that identify a pattern of health conscious market probably in their late teens to early 20s

Cluster 3

```
cluster3 = subset(sm_trim,cluster == 3)

tail(sort(sapply(cluster3 [,-37],mean)))
```

```
##     computers photo_sharing       chatter          news        travel
##      2.472222      2.538012      4.545322      5.311404      5.605263
##      politics
##      8.942982
```

This cluster has low values of mean and is recorded against travel, shopping, current_events. There are groups that are similar to group in cluster 3 but also involves non-professionals.

Cluster 4

```
cluster4 = subset(sm_trim,cluster == 4)

tail(sort(sapply(cluster4[,-37],mean)))
```

```
## health_nutrition           beauty          chatter          fashion
##         2.310835         3.923623         4.914742         5.577265
##    photo_sharing          cooking
##         6.175844        10.959147
```

The group of people under cluster 4 can be identified as those interested in fashion, beauty and cooking. These can be classified as women of age group [18-35]

clsuter 5

```
cluster5 = subset(sm_trim,cluster == 5)

tail(sort(sapply(cluster5[,-37],mean)))
```

```
##        tv_film sports_playing  photo_sharing        chatter  online_gaming
##       1.942857       2.562637       2.876923       4.580220       9.246154
##    college_uni
##      10.336264
```

We find that the segment of observations here are married men and women who relate to food, parenting, school and sports_fandom

Clsuter 6

```
cluster6 = subset(sm_trim,cluster == 6)

tail(sort(sapply(cluster6[,-37],mean)))
```

```
##    photo_sharing         outdoors          cooking          chatter
##         2.693002         2.739278         3.267494         4.405192
## personal_fitness health_nutrition
##         6.446953        11.994357
```

The group of people under cluster 3 could be classified as those interested in travel, policits and news. Could be identified with people of mid age professional group.

We find that phot_sharing and chatter are two variables that appear in all the clusters, meaning that these variables are common for all the clusters and are significant because they almost all posts have a photo associated with them and the segmented as chatter.