Homework1

Jack Cheng

8/6/2017

R Markdown

This is the homework 1 for the second part of STA380 in Red McCombs business school.

Statistics Questions:

Question 1

From the question we know that:

P(RC)=0.3

P(TC)=1-P(RC)=0.7 since TC is the complement of RC

P(Y)=0.65

P(Y|RC)=0.5

Where RC denotes that the clicker is a random clicker, TC denotes the clicker is a truthful clicker and Y denotes the result is yes.

And we want to know P(Y|TC).

Solution:

P(Y,RC)=P(Y|RC)*P(RC)=0.5*0.3=0.15

P(Y,TC)=P(Y)-P(Y,RC)=0.65-0.15=0.5 since TC is the complement of RC

so P(Y|TC)=P(Y,TC)/P(TC)=0.5/0.7=0.7142857

Question 2

From the question we know that:

P(P|D)=0.993

P(N|Dc)=0.9999

P(D)=0.000025

Where D denotes with desease, Dc denotes no desease, P denotes positive and N denotes negative.

We want to know: P(D|P)

Solution:

since we know Dc is the complement of D

so P(Dc)=1-P(D)=0.999975 and P(P)=P(Dc,P)+P(D,P)

and N is the complement of P

so P(P|Dc)=1-P(N|Dc)=0.0001

P(D|P)=P(D,P)/P(P)

=(P(P|D)*P(D))/(P(D,P)+P(Dc,P))

=(P(P|D)*P(D))/(P(P|D)*P(D)+P(Dc,P)*P(Dc))

=0.993*0.000025/(0.993*0.000025+0.0001*0.999975)=0.1988824

Which is really high!

That is to say though the sensitivity and specificity of the test is really good, due to the fact that the prior probability of desease is so low as 0.000025, the false positive rate is still really high. This kind of implementing a universal testing policy for the disease will lead to panic and chaos.

Exploratory analysis: green buildings

Question:

Actually, in order to find out the answer to this question and to determine whether having this new house as a green certificated house is a great option, we finished a lot of exploratory analysis focusing on comparing the greenhouse to the non green ones in a way that control the other factors constant. The most intuitive way of finishing this task would be running a regression, however if you run a regression you actually exclude all the other factors that may link to green house that contribute to the price, which is not the real scenario. In other words, we consider many features that associated with green house that can contribute to the price as some important effects that help us decide whether to have this certificate. So in this case we don't want to actually exclude these effects to find out the 'pure' effect of green house as we always do in a regression analysis, what we will do is just to compare them in a reasonable way. That is the whole picture and instructive idea of our work below:

```r
green_house = read.csv("~/Desktop/UT Austin/Predictive learning/greenbuildings.csv")
green_only = subset(green_house, green_rating==1)
green.rent=rep(NA,length(unique(green_house$cluster)))
avg.rent = rep(NA,length(unique(green_house$cluster)))
for (i in 1:length(unique(green_house$cluster))){
  index = which(green_house$cluster==unique(green_house$cluster)[i])
  green.rent[i]=green_house$Rent[index]
  avg.rent[i]=green_house$cluster_rent[index]
}

avg.increase = mean(green.rent-avg.rent)
5000000/(2.999978*250000)
```

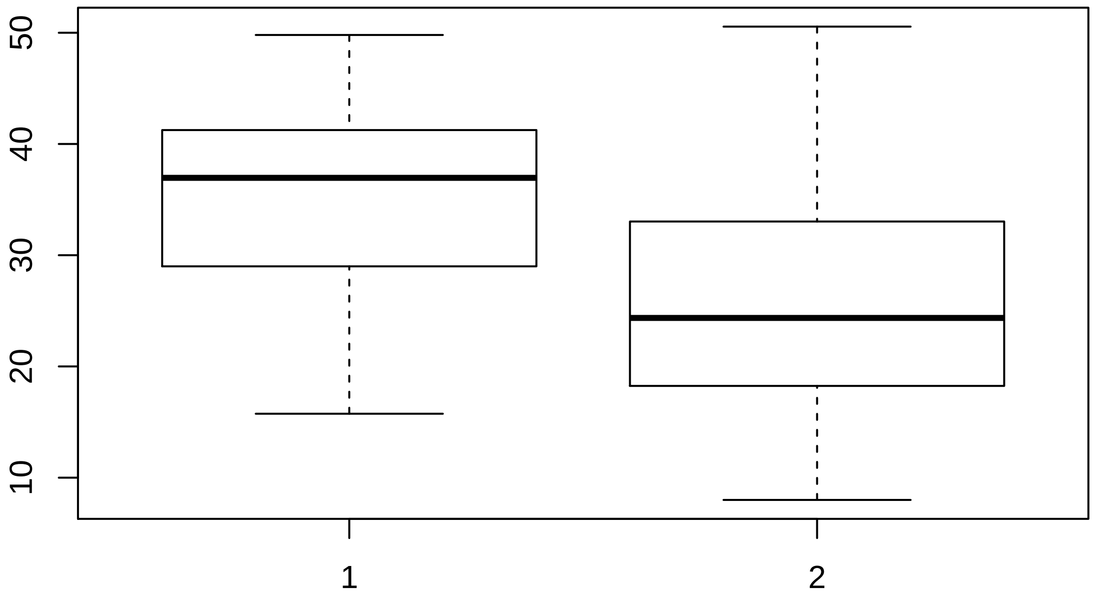
```
## [1] 6.666716
```

```r
# we need around 6.666716 years to recuperate the premium cost.
```

We need to take into account about the indormation we do have about this house and only compare the houses in this way and leave the other factors as a set of different factor levels at different probabilities. Which is we look at the dataset we have as a random sample that could represent the whole population. Since there are a lot of feathers about the house that we don't know, what our group assume is just leave them as a probability distribution that the whole dataset can just be a well representative of. As for the information that we do know about this house, we compare the relative houses that have the same feature values.

So what do we actually know about the building? First, it is with 15 story, and it is planning to be build so its age must be really small. Furthermore, we extract all building with story 15 and find out those buildings with relatively low age (since in the raw dataset there is rarely small number of ralatively new house so we loosen our condition a little bit)

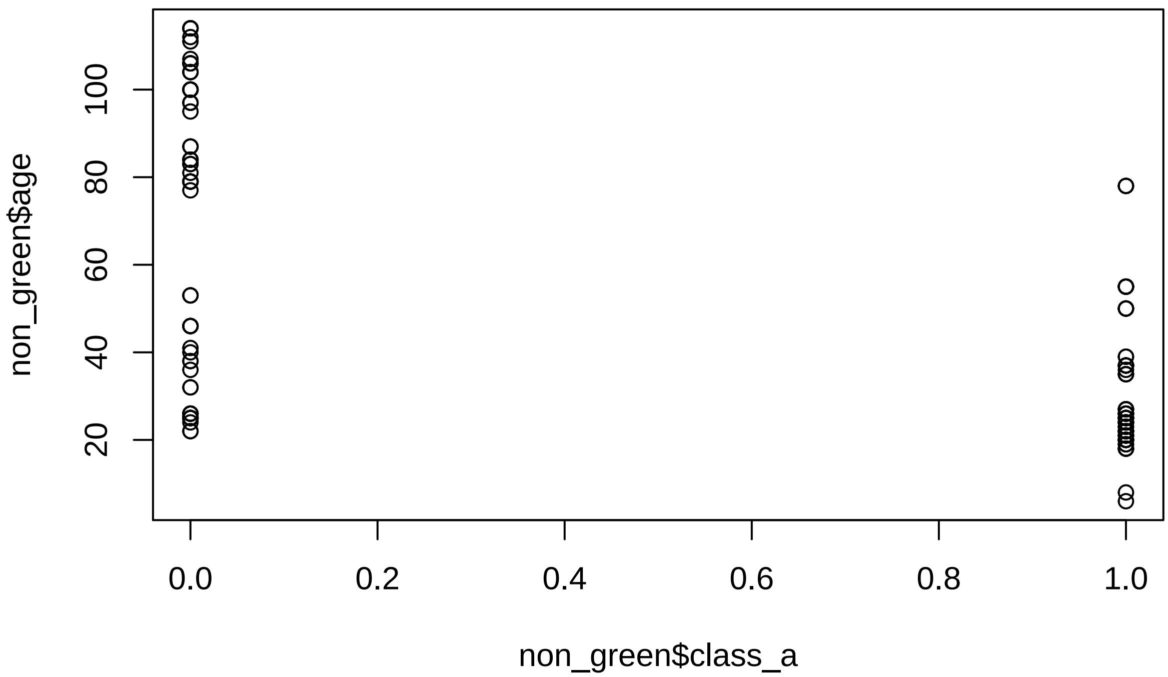
```
story_15 = subset(green_house, stories==15)
non_green = subset(story_15, green_rating!=1)
non_green_avg = mean(subset(non_green)$Rent)
green=subset(story_15, green_rating==1)
green_avg=mean(green$Rent)
boxplot(green$Rent, non_green$Rent)
```



```
green_avg-non_green_avg
```

```
## [1] 8.668808
```

```
plot(non_green$class_a,non_green$age)
```



```
lala1=non_green[which(non_green$age<=20),]
lala2=subset(lala1, class_a==1 | class_b==1)
mean(lala2$Rent)
```

```
## [1] 32.50467
```

```
mean(green$Rent)
```

```
## [1] 34.954
```

```
mean(lala2$leasing_rate)
```

```
## [1] 90.07
```

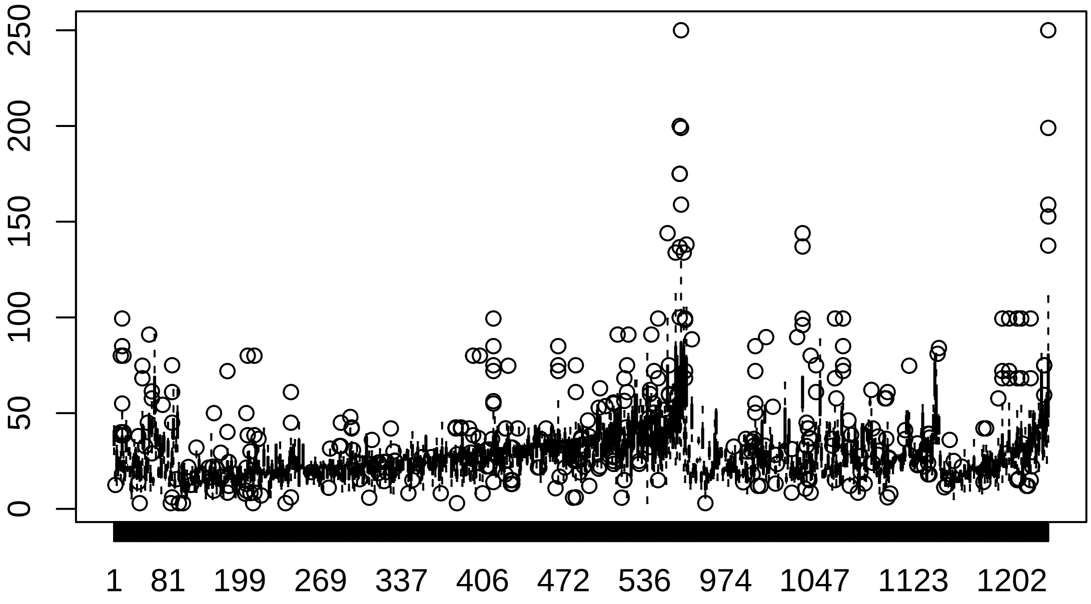
```
mean(green$leasing_rate)
```

```
## [1] 86.015
```

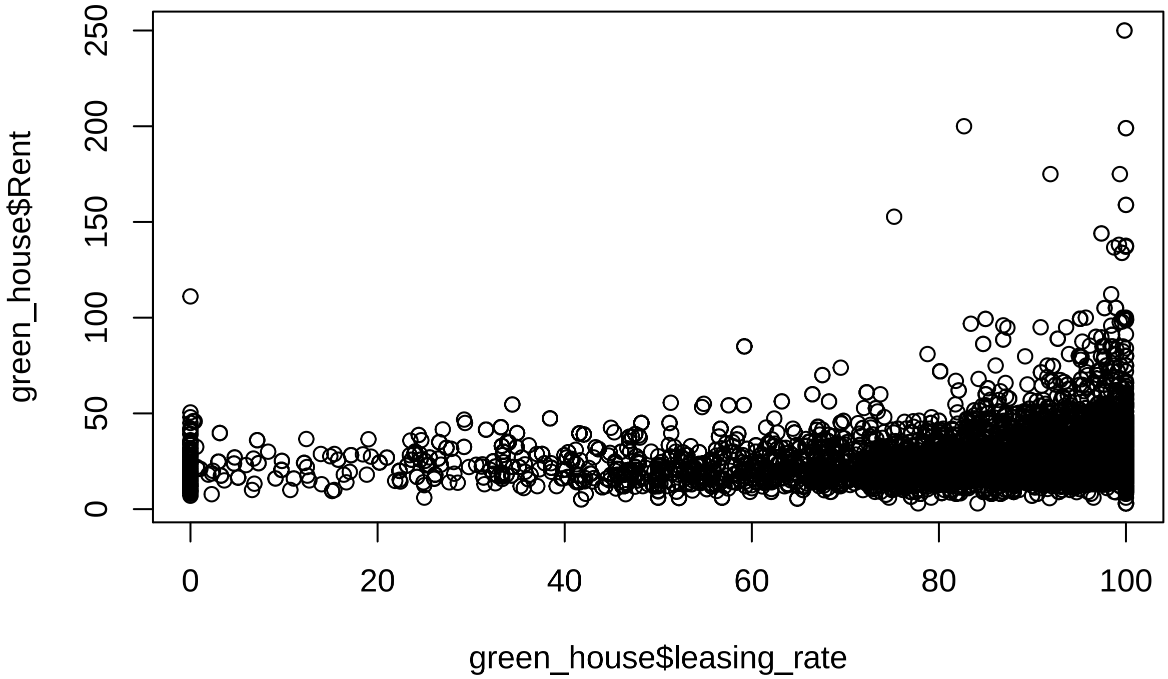
```
year_green=5000000/(250000*mean(green$leasing_rate)*0.01*(mean(green$Rent)-mean(lala2$Rent)))
year_green
```

```
## [1] 9.493097
```

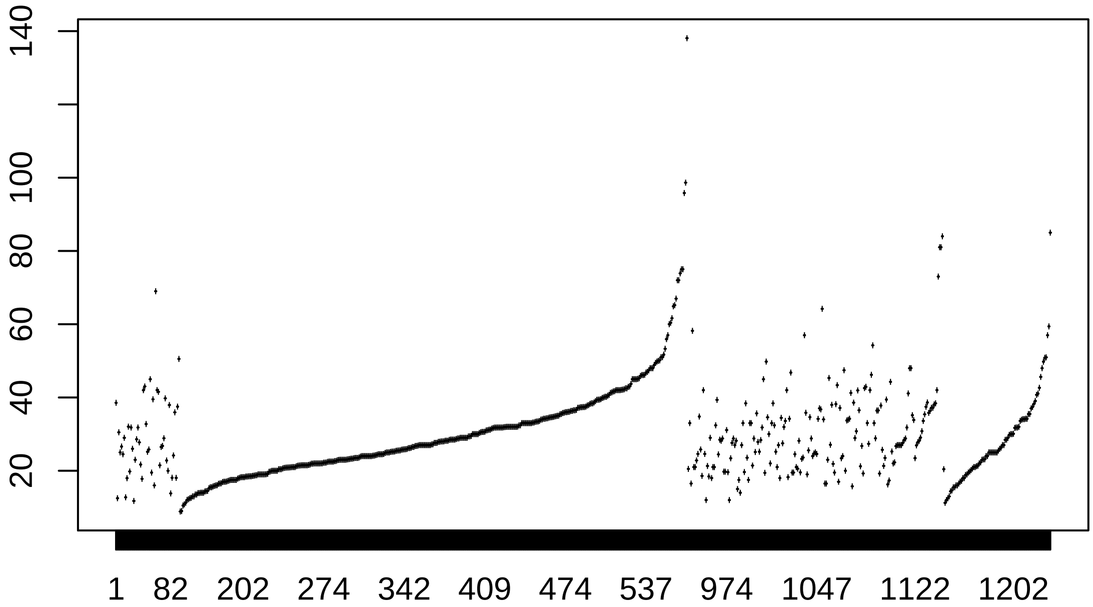
```
boxplot(Rent~cluster, green_house)
```



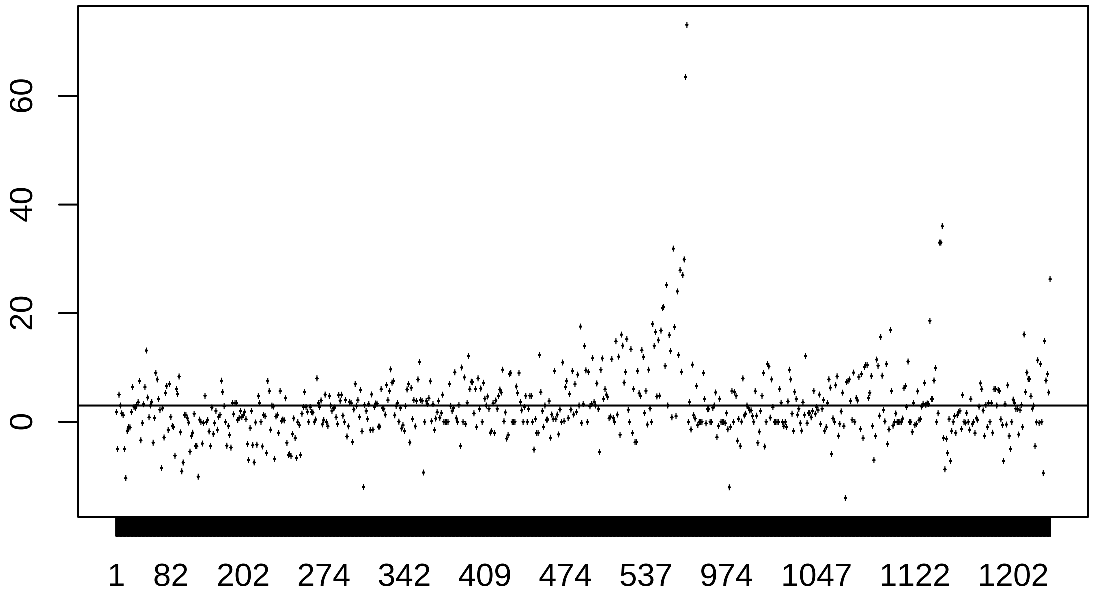
```
plot(green_house$leasing_rate,green_house$Rent)
```



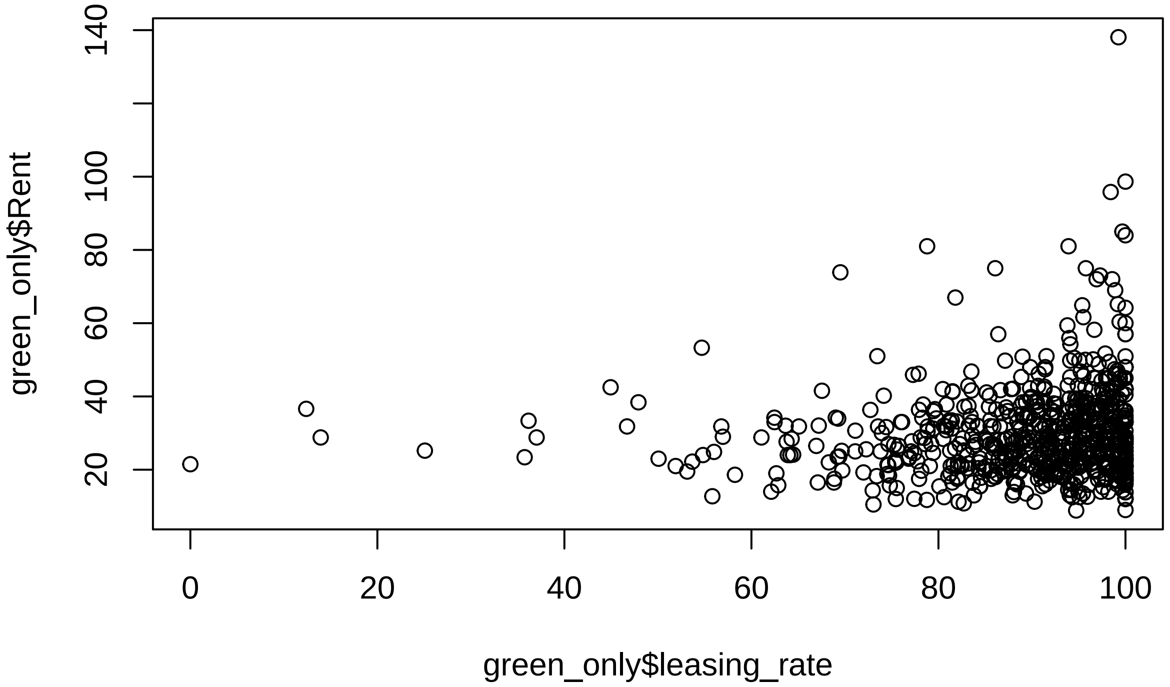
```
boxplot(Rent~cluster, green_only)
```



```
green_only[,'profit_per_ft']=green_only[,'Rent']-green_only[,'cluster_rent']
boxplot(green_only$profit_per_ft~green_only$cluster)
abline(h=mean(green.rent-avg.rent))
```



```r
plot(green_only$leasing_rate, green_only$Rent)
```



Here we can see we actually need 9.493097 years to get the investment on green certification back, which is differently from the estimation the guru provided.

Bootstrapping

```r
library(mosaic)
```

```
## 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: ggformula
```

```
## Loading required package: ggplot2
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
## Loading required package: mosaicData
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, 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
```

```r
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:dplyr':
##
##     first, last
## Loading required package: TTR
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
library(foreach)
mystocks = c("SPY", "TLT", "LQD","EEM","VNQ")
getSymbols(mystocks)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

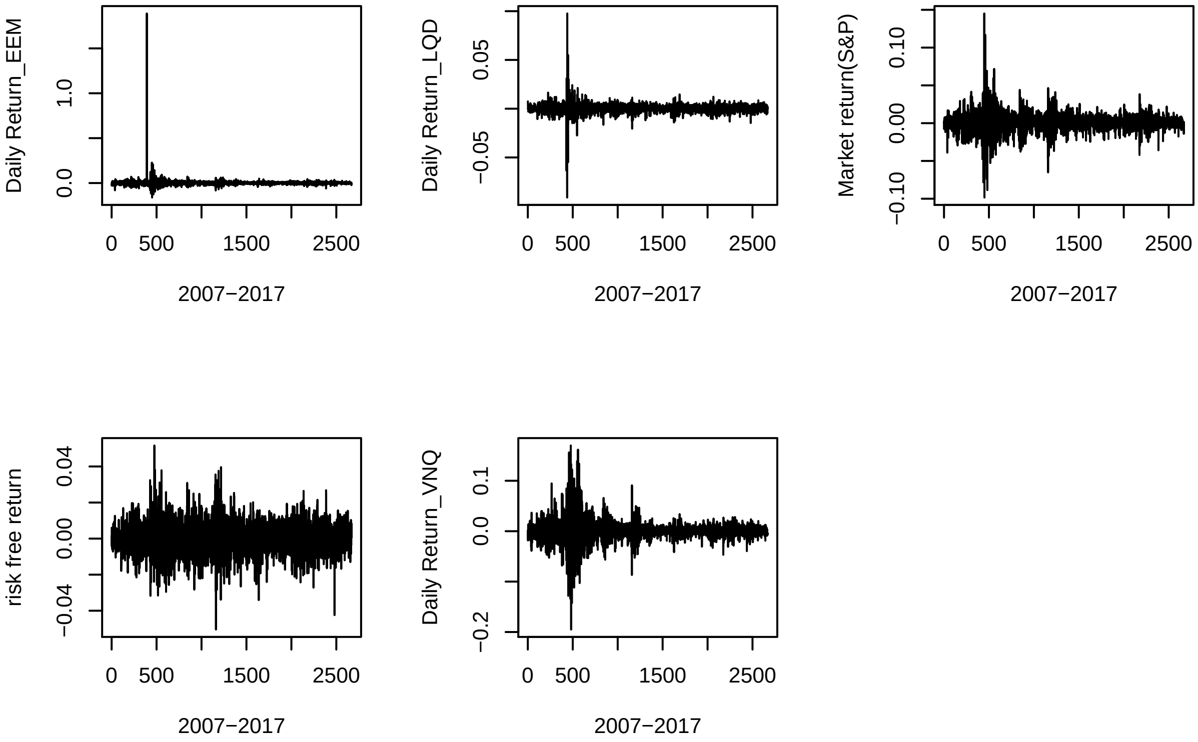
```
##
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).
```

```
## Warning: LQD contains missing values. Some functions will not work if
## objects contain missing values in the middle of the series. Consider using
## na.omit(), na.approx(), na.fill(), etc to remove or replace them.
```

```
## [1] "SPY" "TLT" "LQD" "EEM" "VNQ"
```

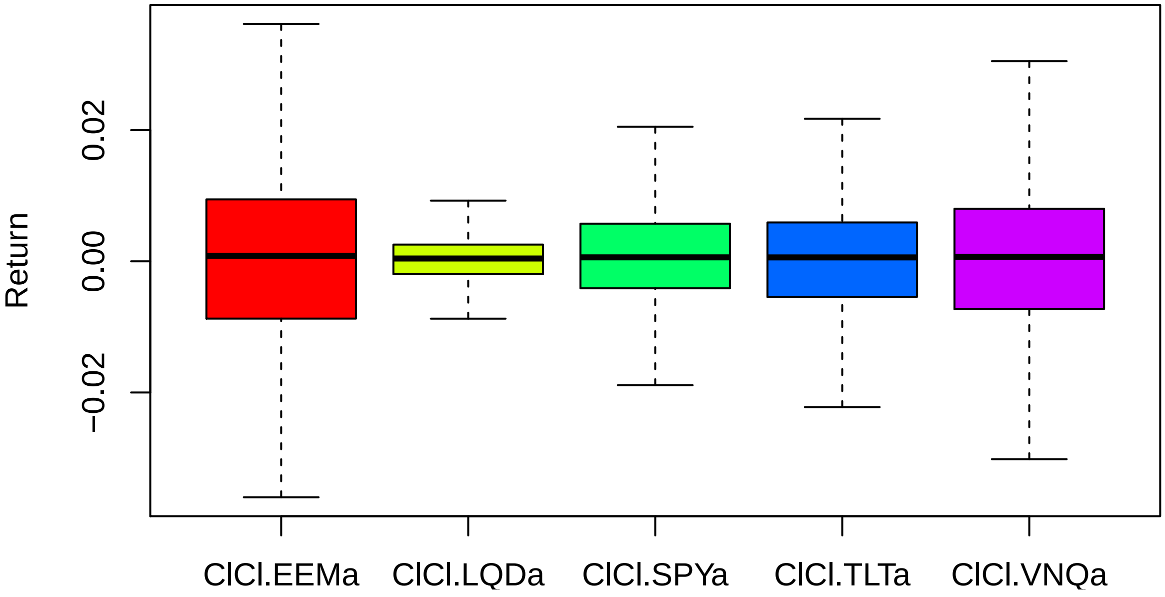
```r
EEMa = adjustOHLC(EEM)
LQDa = adjustOHLC(LQD)
SPYa = adjustOHLC(SPY)
TLTa = adjustOHLC(TLT)
VNQa = adjustOHLC(VNQ)

all_returns = cbind(ClCl(EEMa),ClCl(LQDa),ClCl(SPYa),ClCl(TLTa),ClCl(VNQa))
all_returns = as.matrix(na.omit(all_returns))
par(mfrow=c(2,3))
plot(all_returns[,1], type='l',xlab="2007-2017",ylab="Daily Return_EEM")
plot(all_returns[,2], type='l',xlab="2007-2017",ylab="Daily Return_LQD")
plot(all_returns[,3], type='l',xlab="2007-2017",ylab="Market return(S&P)")
plot(all_returns[,4], type='l',xlab="2007-2017",ylab="risk free return")
plot(all_returns[,5], type='l',xlab="2007-2017",ylab="Daily Return_VNQ")
```



First let us just to explore the data by looking at the mean and variance of each asset to get a roughly idea about their risk return properties:

```
boxplot(all_returns,outline=FALSE,col=rainbow(5),ylab='Return')
```



```
library(plotly)
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:mosaic':
##
##     do
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
plot_ly(type='box', yaxis= list(range = c(-0.5, 0.5))) %>%
  add_boxplot(y = all_returns[,3], name = 'SPY') %>%
  add_boxplot(y = all_returns[,4], name = 'TLT') %>%
  add_boxplot(y = all_returns[,2], name = 'LQD') %>%
  add_boxplot(y = all_returns[,1], name = 'EEM') %>%
  add_boxplot(y = all_returns[,5], name = 'VNQ') %>%
  layout(
    yaxis = list(range = c(-0.5,0.5)))
```

We can see through the graph that EEM and VNQ are with high volatility and rather high return and the TLT is just the most robust way of investing. Then let us see the sharp ratio for those assets first to get a more through measurement of these assets:

- There are many ways to measure the performance of a certain asset. Here we choose the sharp ratio, Jensen's alpha and treynor ratio as examples.

Note that sharp ratio is the ratio between extra mean return exceed risk free asset and the volitility of asset.

First use TLT as an approximation to the risk free asset. Calculate the extra return of each asset:

```
EEMa = adjustOHLC(EEM)
LQDa = adjustOHLC(LQD)
SPYa = adjustOHLC(SPY)
TLTa = adjustOHLC(TLT)
VNQa = adjustOHLC(VNQ)
EEM_extra_return=all_returns[,1]-all_returns[,4]
LQD_extra_return=all_returns[,2]-all_returns[,4]
Market_extra_return=all_returns[,3]-all_returns[,4]
VNQ_extra_return=all_returns[,5]-all_returns[,4]

EEM_SD=sd(EEM_extra_return)
LQD_SD=sd(LQD_extra_return)
Market_SD=sd(Market_extra_return)
VNQ_SD=sd(VNQ_extra_return)

SR_EEM=mean(EEM_extra_return)/EEM_SD
SR_LQD=mean(LQD_extra_return)/LQD_SD
SR_Market=mean(Market_extra_return)/Market_SD
SR_VNQ=mean(VNQ_extra_return)/VNQ_SD

SR_EEM
```

```
## [1] 0.01670746
```

```
SR_LQD
```

```
## [1] -0.009331467
```

```
SR_Market
```

```
## [1] 0.002448236
```

```
SR_VNQ
```

```
## [1] 0.004535166
```

Actually here we see that LQD have a lower mean return than risk free asset, that probably suggest us not to invest in this asset since this performance is really terrible. EEM got the highest sharp ratio.

Then we look at Jensen's alpha:

Jensen's alpha is a measurement of the return after adjusting by taking risk into account.

Fit the data with a CAPM model first:

```
lmEEM=lm(EEM_extra_return~Market_extra_return)
lmLQD=lm(LQD_extra_return~Market_extra_return)
lmVNQ=lm(VNQ_extra_return~Market_extra_return)
coef(lmEEM)
```

```
##         (Intercept) Market_extra_return
##        0.0006889502        1.1845306651
```

```
coef(lmLQD)
```

```
##         (Intercept) Market_extra_return
##       -9.716354e-05        3.552807e-01
```

```
coef(lmVNQ)
```

```
##         (Intercept) Market_extra_return
##        6.475704e-05        1.157159e+00
```

The alpha and beta are intercept and beta in this particular case.

We can see that EEM get the highest alpha while VNQ get the lowest.

Then let us look at the treynor ratio:

```
TR_EEM=mean(EEM_extra_return)/coef(lmEEM)[2]
TR_LQD=mean(LQD_extra_return)/coef(lmLQD)[2]
TR_VNQ=mean(VNQ_extra_return)/coef(lmVNQ)[2]

TR_EEM
```

```
## Market_extra_return
##        0.0006279864
```

```
TR_LQD
```

```
## Market_extra_return
##       -0.0002271204
```

```
TR_VNQ
```

```
## Market_extra_return
##        0.0001023256
```

We can see here that EEM is with the highest treynor ratio.

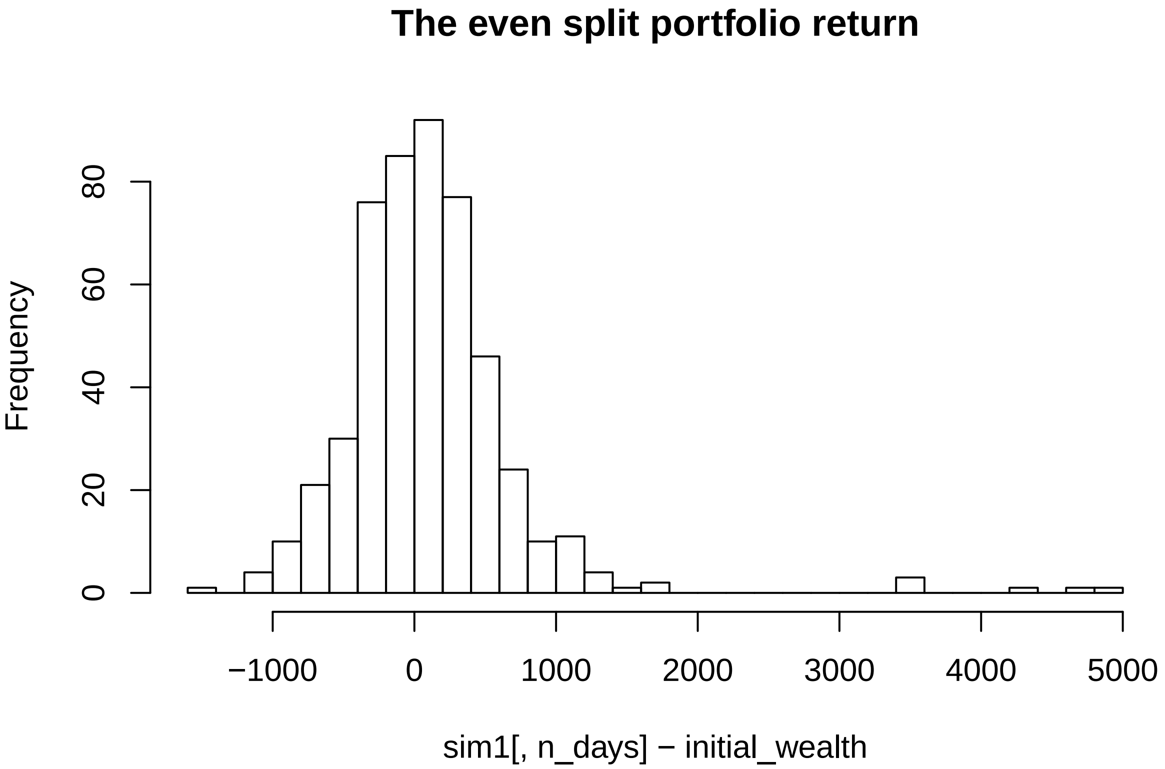
Now we begin the boostrapping simulation.

The even split one

```
set.seed(888)
initial_wealth = 10000
sim1 = foreach(i=1:500, .combine='rbind') %do% {
    total_wealth = initial_wealth
    weights = c(0.2, 0.2, 0.2, 0.2, 0.2)
    holdings = weights * total_wealth
    n_days = 20
    wealthtracker = rep(0, n_days)
    for(today in 1:n_days) {
        return.today = resample(all_returns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
    }
    wealthtracker
}
mean(sim1[,n_days]- initial_wealth)
```

```
## [1] 101.0998
```

```
hist(main = 'The even split portfolio return',sim1[,n_days]- initial_wealth, breaks=30)
```

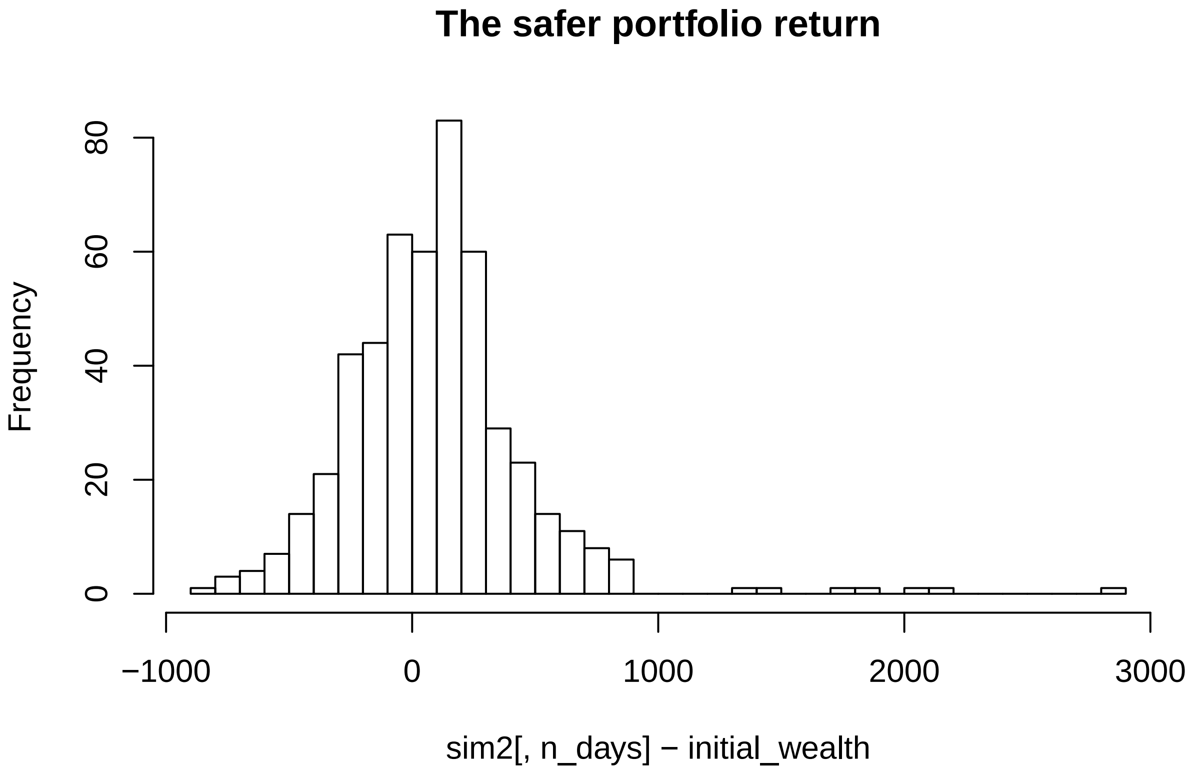


The safer portfolio Next we have our safe portfolio as we invest more on the TLT and since we analysis that LQD is inefficient so we do not invest in it.

```
set.seed(888)
initial_wealth = 10000
sim2 = foreach(i=1:500, .combine='rbind') %do% {
    total_wealth = initial_wealth
    weights = c(0.1, 0, 0.1, 0.7, 0.1)
    holdings = weights * total_wealth
    n_days = 20
    wealthtracker = rep(0, n_days)
    for(today in 1:n_days) {
        return.today = resample(all_returns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
    }
    wealthtracker
}
mean(sim2[,n_days]- initial_wealth)
```

```
## [1] 94.73552
```

```
hist(main = 'The safer portfolio return',sim2[,n_days]- initial_wealth, breaks=30)
```



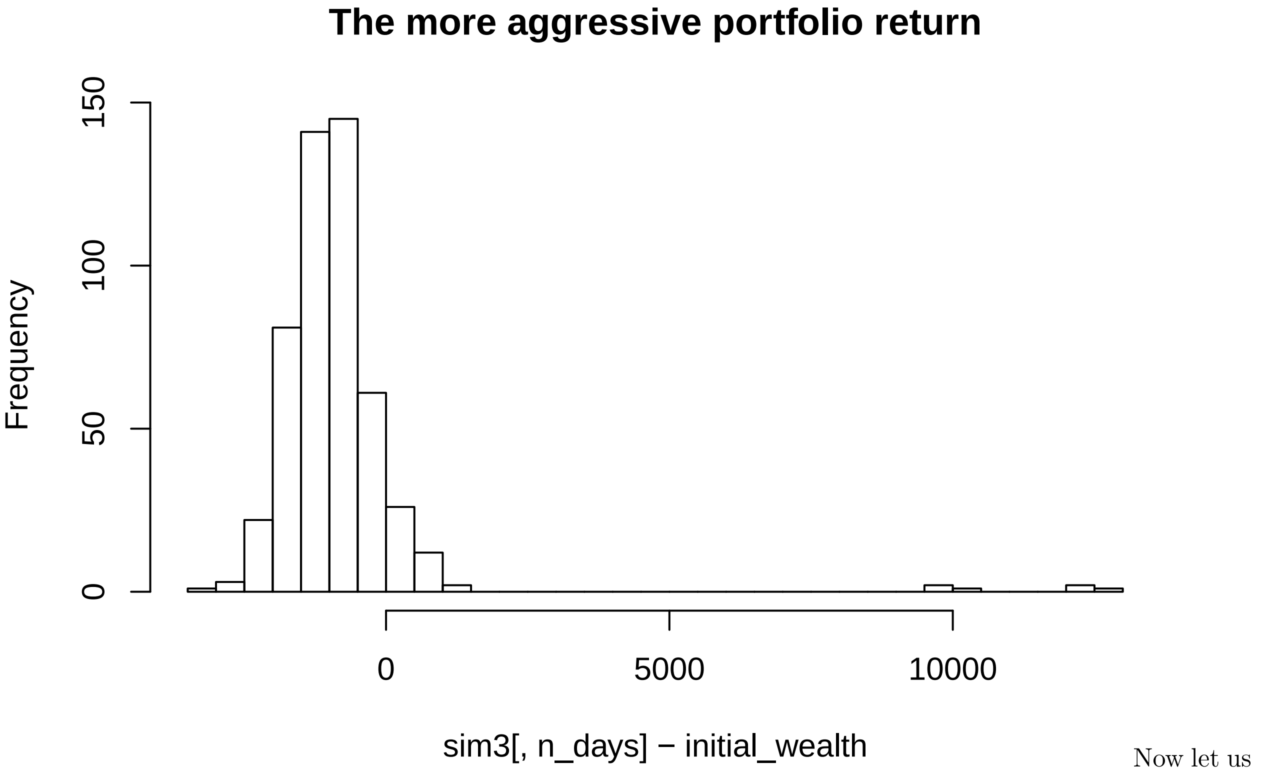
The more aggressive one.

In this one we choose to invest in more EEM which is a more risky asset.

```
set.seed(888)
initial_wealth = 10000
sim3 = foreach(i=1:500, .combine='rbind') %do% {
    total_wealth = initial_wealth
    weights = c(0.6, 0, 0.1, 0.1, 0.1)
    holdings = weights * total_wealth
    n_days = 20
    wealthtracker = rep(0, n_days)
    for(today in 1:n_days) {
        return.today = resample(all_returns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
    }
    wealthtracker
}
mean(sim3[,n_days]- initial_wealth)
```

```
## [1] -835.7462
```

```
hist(main = 'The more aggressive portfolio return',sim3[,n_days]- initial_wealth, breaks=30)
```



Now let us look at the value at risk at 5% level of these portfolios:

```
quantile(sim1[,n_days], 0.05) - initial_wealth
```

```
##        5%
## -669.3861
```

```
quantile(sim2[,n_days], 0.05) - initial_wealth
```

```
##        5%
## -403.6357
```

```
quantile(sim3[,n_days], 0.05) - initial_wealth
```

```
##        5%
## -2001.593
```

As we can see, here we have the value at risk of 5% for these three different portfolio with different style, and these numbers make sence as the aggressive one have the highest risk and the safe one is with the lowest risk.

Market segmentation

```
library(pander)
library(ggplot2)
library(LICORS)
library(foreach)
library(mosaic)
library(gridExtra)
library(wordcloud)
```

Data Cleaning

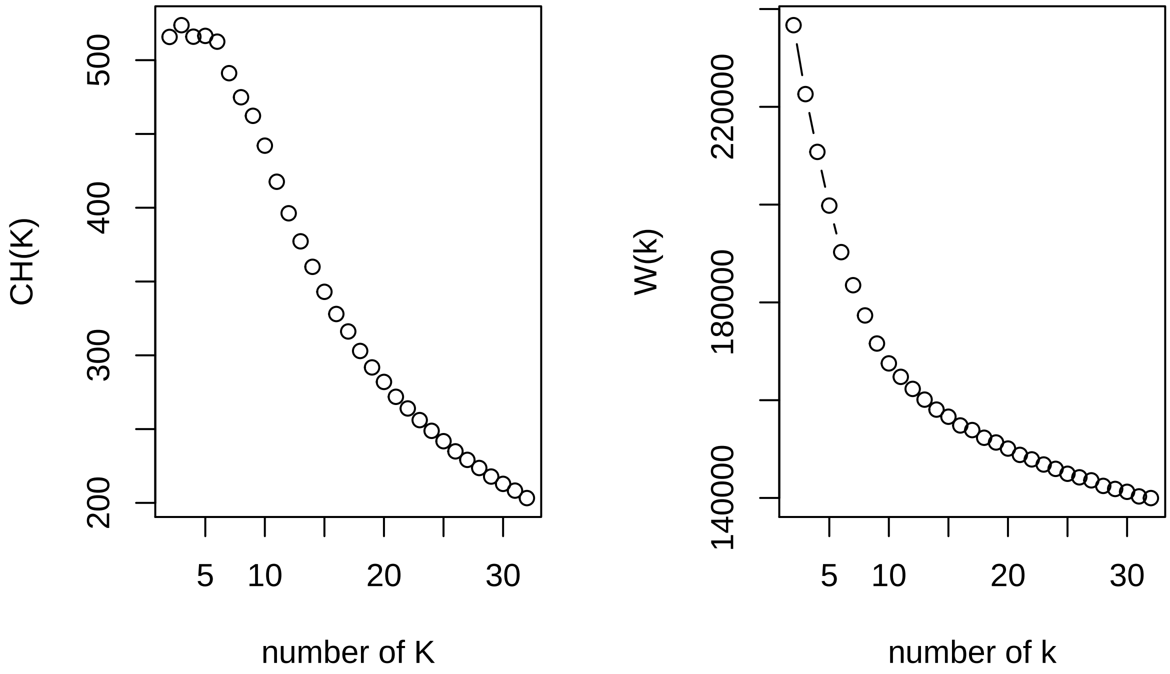
```
sm = read.csv("~/Desktop/UT Austin/Predictive learning/social_marketing.csv")
sm_1 = subset(sm, select = -c(X,uncategorized, spam, adult, chatter))
sm_sub=sm_1/rowSums(sm_1)
sm_new = scale(sm_sub, center=TRUE, scale=TRUE)
```

K-means Clustering

```
list= rep(NA, dim(sm_new)[2]-1)
list2= rep(NA, dim(sm_new)[2]-1)

set.seed(1)
for ( i in 2:dim(sm_new)[2]){
  list[i-1]=kmeans(sm_new, i , nstart = 50)$betweenss/ kmeans(sm_new, i , nstart = 50)$tot.withinss *(d
  list2[i-1]=kmeans(sm_new, i , nstart = 50)$tot.withinss
}

par(mfrow=c(1,2))
plot(list ~ c(2:32), type='b', xlab = 'number of K', ylab = 'CH(K)')
plot(list2 ~ c(2:32), type='b',xlab='number of k',ylab='W(k)')
```



So based on the graphs above, we are able to see the relationship between number of K and CH(K) and W(K) respectively. As the number of K increases, both values of CH(K) and W(K) decrease. Since we could only have 32 clusters for 32 variables, we tried different K from k=2 to k=32 and find out that CH(K) and W(K) are at minimum when K=32. However, this does not make sense to us since K=32 is too large and we would have same variables in different clusters.

K means for k=2 to k=8

```
set.seed(1)
kmeans_sm2<- kmeans(sm_new, 2, nstart = 50)
print(apply(kmeans_sm2$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                2
##  [1,] "sports_fandom" "health_nutrition"
##  [2,] "politics"      "cooking"
##  [3,] "religion"      "personal_fitness"
##  [4,] "travel"        "outdoors"
##  [5,] "college_uni"   "fashion"
##  [6,] "parenting"     "beauty"
##  [7,] "family"        "eco"
##  [8,] "automotive"    "dating"
##  [9,] "tv_film"       "music"
## [10,] "news"          "food"
```

```
kmeans_sm2$size
```

```
## [1] 5384 2498
```

```
set.seed(1)
kmeans_sm3<- kmeans(sm_new, 3, nstart = 50)
print(apply(kmeans_sm3$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                 2                3
##  [1,] "politics"       "religion"      "health_nutrition"
##  [2,] "college_uni"    "parenting"     "cooking"
##  [3,] "travel"         "sports_fandom" "personal_fitness"
##  [4,] "tv_film"        "food"          "outdoors"
##  [5,] "shopping"       "school"        "fashion"
##  [6,] "news"           "family"        "beauty"
##  [7,] "current_events" "crafts"        "eco"
##  [8,] "online_gaming"  "beauty"        "dating"
##  [9,] "photo_sharing"  "eco"           "music"
## [10,] "automotive"     "automotive"    "food"
```

```
kmeans_sm3$size
```

```
## [1] 4169 1330 2383
```

```
set.seed(1)
kmeans_sm4<- kmeans(sm_new, 4, nstart = 50)
print(apply(kmeans_sm4$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1               2                3                4
##  [1,] "religion"      "politics"       "cooking"        "health_nutrition"
##  [2,] "parenting"     "college_uni"    "fashion"        "personal_fitness"
##  [3,] "sports_fandom" "travel"         "beauty"         "outdoors"
##  [4,] "food"          "shopping"       "photo_sharing"  "cooking"
##  [5,] "school"        "news"           "music"          "eco"
##  [6,] "family"        "tv_film"        "dating"         "food"
##  [7,] "crafts"        "online_gaming"  "business"       "dating"
##  [8,] "beauty"        "current_events" "small_business" "home_and_garden"
##  [9,] "eco"           "photo_sharing"  "sports_playing" "crafts"
## [10,] "automotive"    "automotive"     "school"         "sports_playing"
```

```
kmeans_sm4$size
```

```
## [1] 1314 4085  948 1535
```

```
set.seed(1)
kmeans_sm5<- kmeans(sm_new, 5, nstart = 50)
print(apply(kmeans_sm5$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                2                3                4
##  [1,] "college_uni"    "cooking"        "religion"       "politics"
##  [2,] "shopping"       "fashion"        "parenting"      "news"
##  [3,] "photo_sharing"  "beauty"         "sports_fandom"  "automotive"
##  [4,] "online_gaming"  "photo_sharing"  "food"           "travel"
##  [5,] "tv_film"        "music"          "school"         "computers"
##  [6,] "current_events" "dating"         "family"         "sports_fandom"
##  [7,] "sports_playing" "business"       "crafts"         "outdoors"
##  [8,] "art"            "small_business" "beauty"         "business"
##  [9,] "music"          "school"         "eco"            "dating"
## [10,] "small_business" "sports_playing" "small_business" "family"
##       5
##  [1,] "health_nutrition"
##  [2,] "personal_fitness"
##  [3,] "outdoors"
##  [4,] "cooking"
##  [5,] "eco"
##  [6,] "food"
##  [7,] "dating"
##  [8,] "home_and_garden"
##  [9,] "crafts"
## [10,] "art"
```

```
kmeans_sm5$size
```

```
## [1] 2914  926 1247 1318 1477
```

```
set.seed(1)
kmeans_sm6<- kmeans(sm_new, 6, nstart = 50)
print(apply(kmeans_sm6$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                2                3
##  [1,] "politics"       "religion"       "shopping"
##  [2,] "news"           "parenting"      "photo_sharing"
##  [3,] "travel"         "sports_fandom"  "current_events"
##  [4,] "automotive"     "food"           "tv_film"
##  [5,] "computers"      "school"         "art"
##  [6,] "sports_fandom"  "family"         "music"
##  [7,] "outdoors"       "crafts"         "business"
##  [8,] "dating"         "beauty"         "home_and_garden"
##  [9,] "business"       "eco"            "small_business"
## [10,] "small_business" "small_business" "eco"
##       4                  5                 6
##  [1,] "health_nutrition" "online_gaming"   "cooking"
##  [2,] "personal_fitness" "college_uni"     "fashion"
##  [3,] "outdoors"         "sports_playing"  "beauty"
##  [4,] "cooking"          "art"             "photo_sharing"
##  [5,] "food"             "tv_film"         "music"
##  [6,] "eco"              "small_business"  "dating"
##  [7,] "dating"           "family"          "business"
##  [8,] "home_and_garden"  "dating"          "school"
##  [9,] "crafts"           "music"           "sports_playing"
## [10,] "art"              "home_and_garden" "small_business"
```

```
kmeans_sm6$size
```

```
## [1] 1269 1209 2410 1454  657  883
```

```
set.seed(1)
kmeans_sm7<- kmeans(sm_new, 7, nstart = 50)
print(apply(kmeans_sm7$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                 2                 3                4
##  [1,] "photo_sharing"   "online_gaming"   "cooking"        "politics"
##  [2,] "shopping"        "college_uni"     "fashion"        "news"
##  [3,] "current_events"  "sports_playing"  "beauty"         "automotive"
##  [4,] "eco"             "art"             "photo_sharing"  "travel"
##  [5,] "business"        "small_business"  "music"          "computers"
##  [6,] "home_and_garden" "family"          "dating"         "sports_fandom"
##  [7,] "small_business"  "dating"          "school"         "outdoors"
##  [8,] "music"           "home_and_garden" "business"       "dating"
##  [9,] "family"          "tv_film"         "sports_playing" "business"
## [10,] "automotive"      "crafts"          "small_business" "family"
##       5                6                  7
##  [1,] "religion"       "health_nutrition" "tv_film"
##  [2,] "parenting"      "personal_fitness" "art"
##  [3,] "sports_fandom"  "outdoors"         "dating"
##  [4,] "food"           "cooking"          "music"
##  [5,] "school"         "food"             "small_business"
##  [6,] "family"         "eco"              "crafts"
##  [7,] "crafts"         "dating"           "current_events"
##  [8,] "beauty"         "home_and_garden"  "college_uni"
##  [9,] "eco"            "crafts"           "home_and_garden"
## [10,] "small_business" "business"         "business"
```

```
kmeans_sm7$size
```

```
## [1] 1509  617  866 1229 1174 1415 1072
```

```
set.seed(1)
kmeans_sm8<- kmeans(sm_new, 8, nstart = 50)
print(apply(kmeans_sm8$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                 2                 3
##  [1,] "online_gaming"   "photo_sharing"   "travel"
##  [2,] "college_uni"     "shopping"        "politics"
##  [3,] "sports_playing"  "current_events"  "computers"
##  [4,] "art"             "eco"             "news"
##  [5,] "small_business"  "business"        "business"
##  [6,] "family"          "home_and_garden" "dating"
##  [7,] "dating"          "small_business"  "small_business"
##  [8,] "tv_film"         "music"           "crafts"
##  [9,] "home_and_garden" "family"          "eco"
## [10,] "crafts"          "crafts"          "current_events"
##       4                5                 6
##  [1,] "religion"       "tv_film"         "health_nutrition"
##  [2,] "parenting"      "art"             "personal_fitness"
##  [3,] "sports_fandom"  "dating"          "outdoors"
##  [4,] "food"           "music"           "cooking"
##  [5,] "school"         "small_business"  "food"
##  [6,] "family"         "crafts"          "eco"
##  [7,] "crafts"         "current_events"  "dating"
##  [8,] "beauty"         "college_uni"     "home_and_garden"
##  [9,] "eco"            "home_and_garden" "crafts"
## [10,] "small_business" "business"        "business"
##       7                 8
##  [1,] "news"            "cooking"
##  [2,] "automotive"      "fashion"
##  [3,] "politics"        "beauty"
##  [4,] "sports_fandom"   "photo_sharing"
##  [5,] "outdoors"        "music"
##  [6,] "family"          "dating"
##  [7,] "home_and_garden" "school"
##  [8,] "parenting"       "sports_playing"
##  [9,] "school"          "small_business"
## [10,] "current_events"  "business"
```

```
kmeans_sm8$size
```

```
## [1]  606 1463  664 1167 1075 1389  668  850
```

K means ++ for k=2 to k=8

```
set.seed(1)
kmeanspp_sm2<- kmeanspp(sm_new, 2)
print(apply(kmeanspp_sm2$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1               2
##  [1,] "sports_fandom" "health_nutrition"
##  [2,] "politics"      "cooking"
##  [3,] "religion"      "personal_fitness"
##  [4,] "travel"        "outdoors"
##  [5,] "college_uni"   "fashion"
##  [6,] "parenting"     "beauty"
##  [7,] "family"        "eco"
##  [8,] "automotive"    "dating"
##  [9,] "tv_film"       "music"
## [10,] "news"          "food"
```

```
kmeanspp_sm2$size
```

```
## [1] 5384 2498
```

```
set.seed(1)
kmeanspp_sm3<- kmeanspp(sm_new, 3)
print(apply(kmeanspp_sm3$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                  2               3
##  [1,] "health_nutrition" "religion"      "politics"
##  [2,] "cooking"          "parenting"     "college_uni"
##  [3,] "personal_fitness" "sports_fandom" "travel"
##  [4,] "outdoors"         "food"          "tv_film"
##  [5,] "fashion"          "school"        "shopping"
##  [6,] "beauty"           "family"        "news"
##  [7,] "eco"              "crafts"        "current_events"
##  [8,] "dating"           "beauty"        "online_gaming"
##  [9,] "music"            "eco"           "photo_sharing"
## [10,] "food"             "automotive"    "automotive"
```

```
kmeanspp_sm3$size
```

```
## [1] 2383 1330 4169
```

```
set.seed(1)
kmeanspp_sm4<- kmeanspp(sm_new, 4)
print(apply(kmeanspp_sm4$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                 2                 3
##  [1,] "religion"        "politics"        "photo_sharing"
##  [2,] "parenting"       "news"            "college_uni"
##  [3,] "sports_fandom"   "travel"          "shopping"
##  [4,] "food"            "automotive"      "fashion"
##  [5,] "school"          "computers"       "online_gaming"
##  [6,] "family"          "sports_fandom"   "tv_film"
##  [7,] "crafts"          "outdoors"        "beauty"
##  [8,] "beauty"          "business"        "sports_playing"
##  [9,] "eco"             "home_and_garden" "music"
## [10,] "home_and_garden" "dating"          "cooking"
##       4
##  [1,] "health_nutrition"
##  [2,] "personal_fitness"
##  [3,] "outdoors"
##  [4,] "cooking"
##  [5,] "eco"
##  [6,] "food"
##  [7,] "dating"
##  [8,] "home_and_garden"
##  [9,] "crafts"
## [10,] "art"
```

```
kmeanspp_sm4$size
```

```
## [1] 1289 1384 3698 1511
```

```
set.seed(1)
kmeanspp_sm5<- kmeanspp(sm_new, 5)
print(apply(kmeanspp_sm5$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                 2                 3
##  [1,] "religion"        "politics"        "photo_sharing"
##  [2,] "parenting"       "news"            "fashion"
##  [3,] "sports_fandom"   "automotive"      "shopping"
##  [4,] "food"            "travel"          "online_gaming"
##  [5,] "school"          "computers"       "cooking"
##  [6,] "family"          "sports_fandom"   "college_uni"
##  [7,] "crafts"          "outdoors"        "beauty"
##  [8,] "beauty"          "business"        "sports_playing"
##  [9,] "eco"             "small_business"  "dating"
## [10,] "home_and_garden" "home_and_garden" "current_events"
##       4                  5
##  [1,] "health_nutrition" "tv_film"
##  [2,] "personal_fitness" "art"
##  [3,] "outdoors"         "music"
##  [4,] "cooking"          "small_business"
##  [5,] "food"             "crafts"
##  [6,] "eco"              "current_events"
##  [7,] "dating"           "college_uni"
##  [8,] "home_and_garden"  "home_and_garden"
##  [9,] "crafts"           "business"
## [10,] "business"         "travel"
```

```
kmeanspp_sm5$size
```

```
## [1] 1249 1331 2928 1451  923
```

```
set.seed(1)
kmeanspp_sm6<- kmeanspp(sm_new, 6)
print(apply(kmeanspp_sm6$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                  2                3
##  [1,] "health_nutrition" "religion"       "online_gaming"
##  [2,] "personal_fitness" "parenting"      "college_uni"
##  [3,] "outdoors"         "sports_fandom"  "sports_playing"
##  [4,] "cooking"          "food"           "art"
##  [5,] "food"             "school"         "tv_film"
##  [6,] "eco"              "family"         "small_business"
##  [7,] "dating"           "crafts"         "family"
##  [8,] "home_and_garden"  "beauty"         "dating"
##  [9,] "crafts"           "eco"            "music"
## [10,] "art"              "small_business" "home_and_garden"
##       4                5                6
##  [1,] "cooking"        "politics"       "shopping"
##  [2,] "fashion"        "news"           "photo_sharing"
##  [3,] "beauty"         "travel"         "current_events"
##  [4,] "photo_sharing"  "automotive"     "tv_film"
##  [5,] "music"          "computers"      "art"
##  [6,] "dating"         "sports_fandom"  "music"
##  [7,] "business"       "outdoors"       "business"
##  [8,] "school"         "dating"         "home_and_garden"
##  [9,] "sports_playing" "business"       "small_business"
## [10,] "small_business" "small_business" "eco"
```

```
kmeanspp_sm6$size
```

```
## [1] 1454 1209  657  883 1269 2410
```

```
set.seed(1)
kmeanspp_sm7<- kmeanspp(sm_new, 7)
print(apply(kmeanspp_sm7$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                2                  3
##  [1,] "religion"       "health_nutrition" "politics"
##  [2,] "parenting"      "personal_fitness" "news"
##  [3,] "sports_fandom"  "outdoors"         "automotive"
##  [4,] "food"           "cooking"          "travel"
##  [5,] "school"         "food"             "computers"
##  [6,] "family"         "eco"              "sports_fandom"
##  [7,] "crafts"         "dating"           "outdoors"
##  [8,] "beauty"         "home_and_garden"  "dating"
##  [9,] "eco"            "crafts"           "business"
## [10,] "small_business" "art"              "family"
##       4                 5                 6
##  [1,] "online_gaming"   "tv_film"         "photo_sharing"
##  [2,] "college_uni"     "art"             "shopping"
##  [3,] "sports_playing"  "music"           "current_events"
##  [4,] "art"             "small_business"  "eco"
##  [5,] "small_business"  "crafts"          "business"
##  [6,] "family"          "college_uni"     "home_and_garden"
##  [7,] "home_and_garden" "current_events"  "small_business"
##  [8,] "dating"          "home_and_garden" "family"
##  [9,] "tv_film"         "business"        "music"
## [10,] "crafts"          "travel"          "crafts"
##       7
##  [1,] "fashion"
##  [2,] "cooking"
##  [3,] "beauty"
##  [4,] "dating"
##  [5,] "photo_sharing"
##  [6,] "music"
##  [7,] "school"
##  [8,] "business"
##  [9,] "sports_playing"
## [10,] "small_business"
```

```
kmeanspp_sm7$size
```

```
## [1] 1181 1412 1243  609  926 1542  969
```

```
set.seed(1)
kmeanspp_sm8<- kmeanspp(sm_new, 8)
print(apply(kmeanspp_sm8$centers,1,function(x) colnames(sm_new)[order(x, decreasing=TRUE)[1:10]]))
```

```
##       1                2                  3
##  [1,] "religion"       "health_nutrition" "cooking"
##  [2,] "parenting"      "personal_fitness" "fashion"
##  [3,] "sports_fandom"  "outdoors"         "beauty"
##  [4,] "food"           "cooking"          "photo_sharing"
##  [5,] "school"         "food"             "music"
##  [6,] "family"         "eco"              "dating"
##  [7,] "crafts"         "dating"           "school"
##  [8,] "beauty"         "home_and_garden"  "sports_playing"
##  [9,] "eco"            "crafts"           "small_business"
## [10,] "small_business" "business"         "business"
##       4                 5                6
##  [1,] "news"            "travel"         "tv_film"
##  [2,] "automotive"      "politics"       "art"
##  [3,] "politics"        "computers"      "dating"
##  [4,] "sports_fandom"   "news"           "music"
##  [5,] "outdoors"        "business"       "small_business"
##  [6,] "family"          "dating"         "crafts"
##  [7,] "home_and_garden" "small_business" "current_events"
##  [8,] "parenting"       "crafts"         "college_uni"
##  [9,] "school"          "eco"            "home_and_garden"
## [10,] "current_events"  "current_events" "business"
##       7                 8
##  [1,] "photo_sharing"   "online_gaming"
##  [2,] "shopping"        "college_uni"
##  [3,] "current_events"  "sports_playing"
##  [4,] "eco"             "art"
##  [5,] "business"        "small_business"
##  [6,] "home_and_garden" "family"
##  [7,] "small_business"  "dating"
##  [8,] "music"           "tv_film"
##  [9,] "family"          "home_and_garden"
## [10,] "crafts"          "crafts"
```

```
kmeanspp_sm8$size
```

```
## [1] 1167 1389  850  668  664 1075 1463  606
```

After we look at the k mean ++ method, we concluded that when k = 5, the clustering result makes more sense to us. So we decided to go with this result.

```
sum(kmeanspp_sm2$withinss)
```

```
## [1] 236700.4
```

```
sum(kmeanspp_sm3$withinss)
```

```
## [1] 222600.8
```

```
sum(kmeanspp_sm4$withinss)
```

```
## [1] 210836.1
```

```
sum(kmeanspp_sm5$withinss)
```

```
## [1] 203575.8
```

```
sum(kmeanspp_sm6$withinss)
```

```
## [1] 190279
```

```
sum(kmeanspp_sm7$withinss)
```

```
## [1] 183564.9
```

```
sum(kmeanspp_sm8$withinss)
```

```
## [1] 177331.5
```

```
sm_distance_matrix = dist(sm_new, method='euclidean')

hier_sm = hclust(sm_distance_matrix, method='average')
cluster1 = cutree(hier_sm, k=5)

hier_sm2 = hclust(sm_distance_matrix, method='complete')
cluster2 = cutree(hier_sm2, k=5)

hier_sm3 = hclust(sm_distance_matrix, method='single')
cluster3 = cutree(hier_sm3, k=5)
summary(factor(cluster1))
```

```
##    1    2    3    4    5
## 7877    1    1    2    1
```

```
summary(factor(cluster2))
```

```
##    1    2    3    4    5
## 7735  138    2    5    2
```

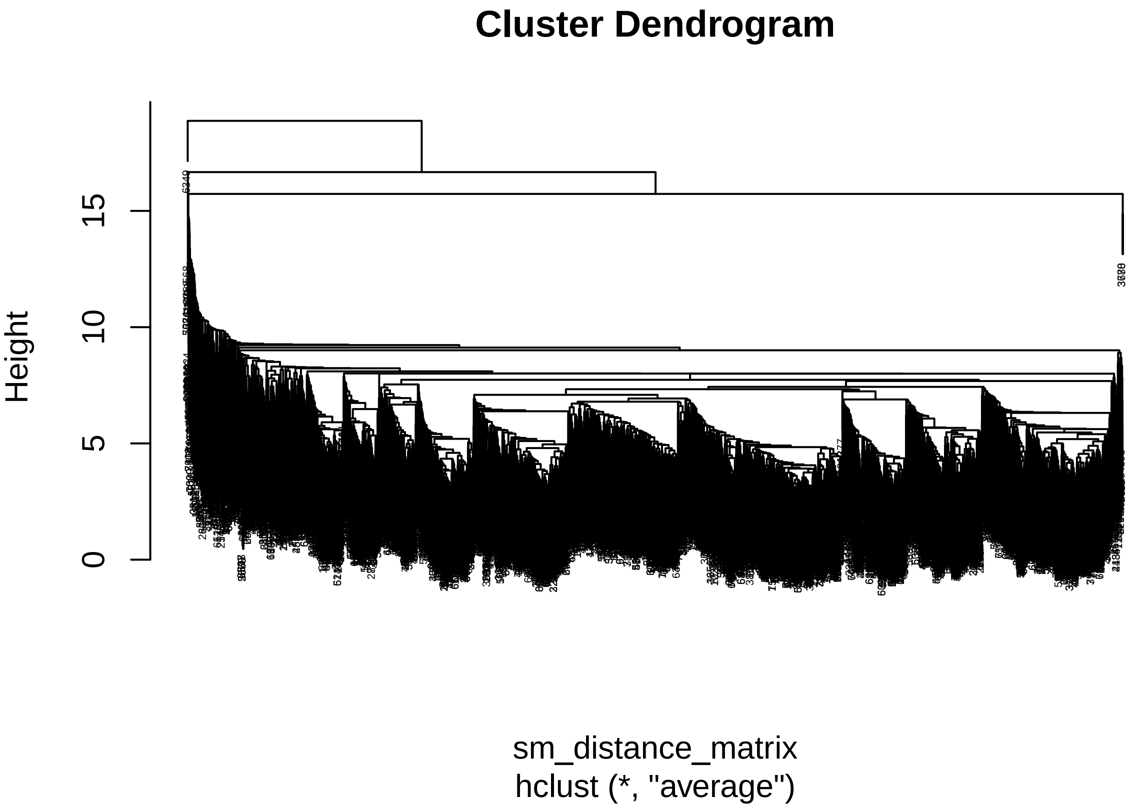
```
summary(factor(cluster3))
```

```
##    1    2    3    4    5
## 7878    1    1    1    1
```

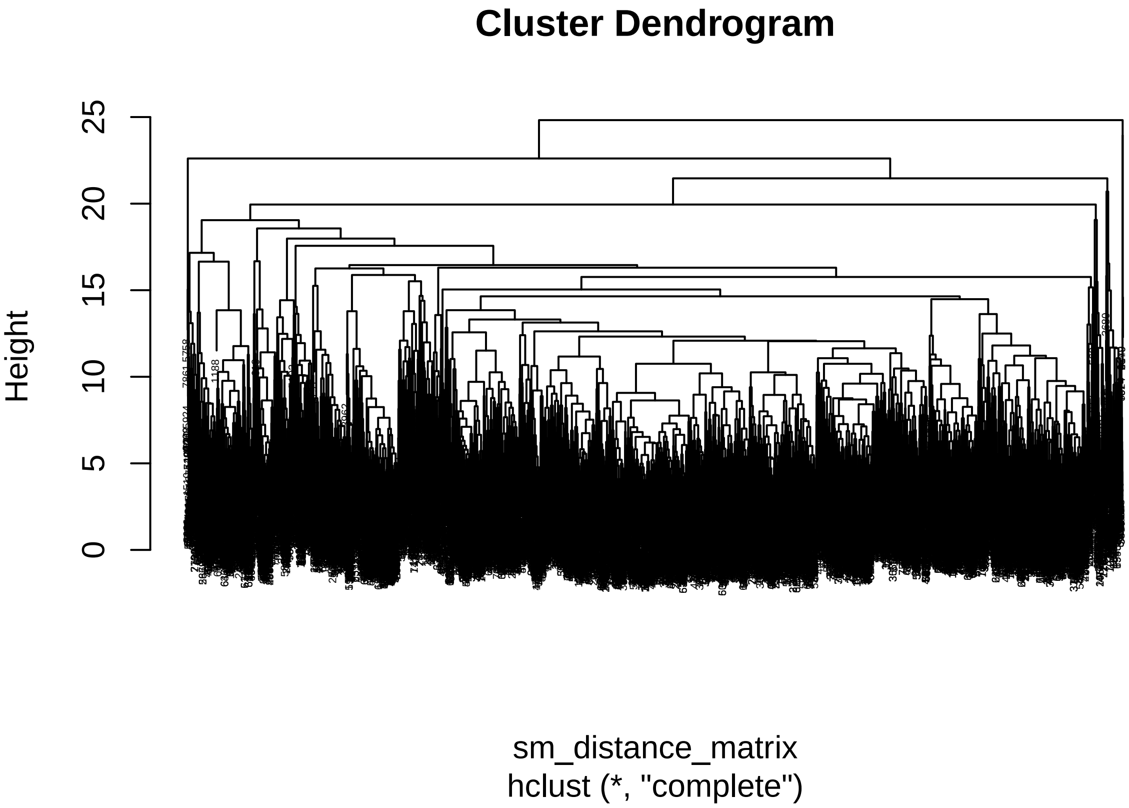
We tried hcluster for k=5 for min, max and average methods and find out that hcluster does not work very well in this case because most of the data points lie on cluster 1.

So we just give up using this hcluster model due to its significantly inbalance cluster result that obviously will not work for the user segmentation task.

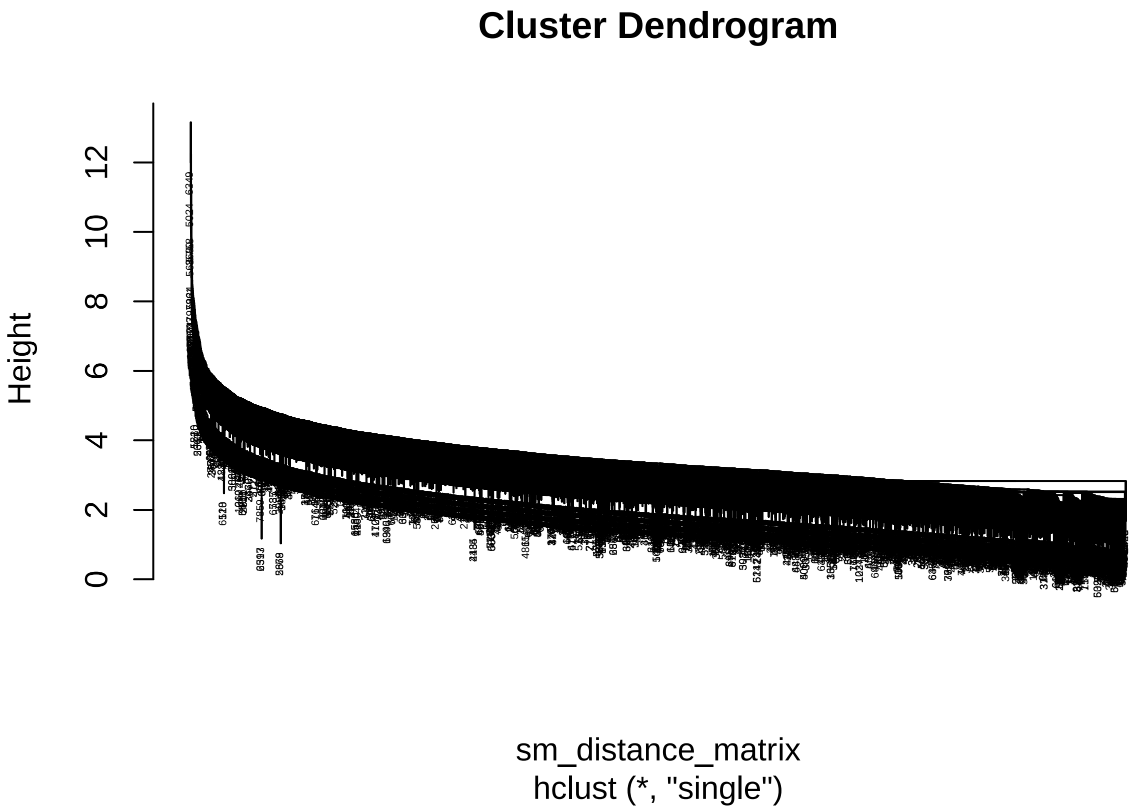
```
plot(hier_sm, cex=0.3)
```



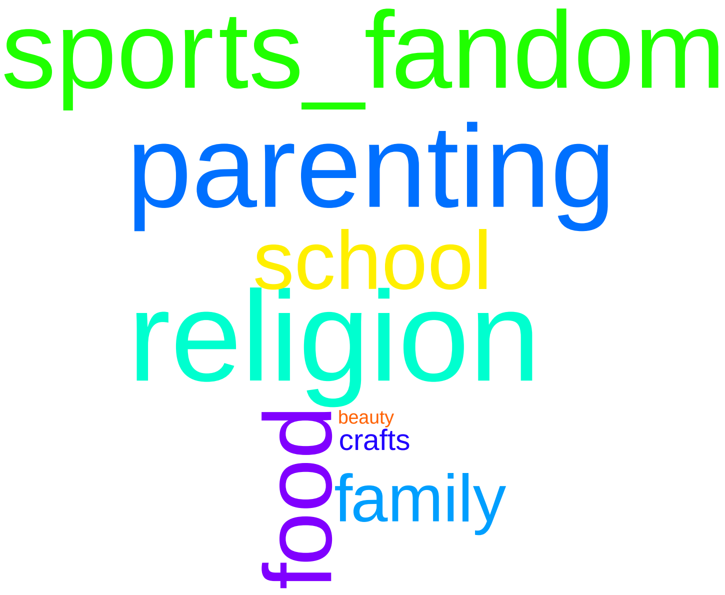
```
plot(hier_sm2, cex=0.3)
```



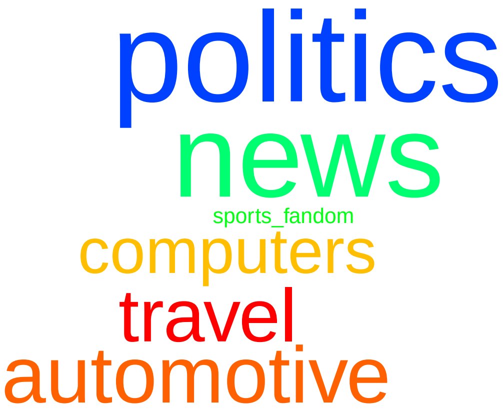
```r
plot(hier_sm3, cex=0.3)
```



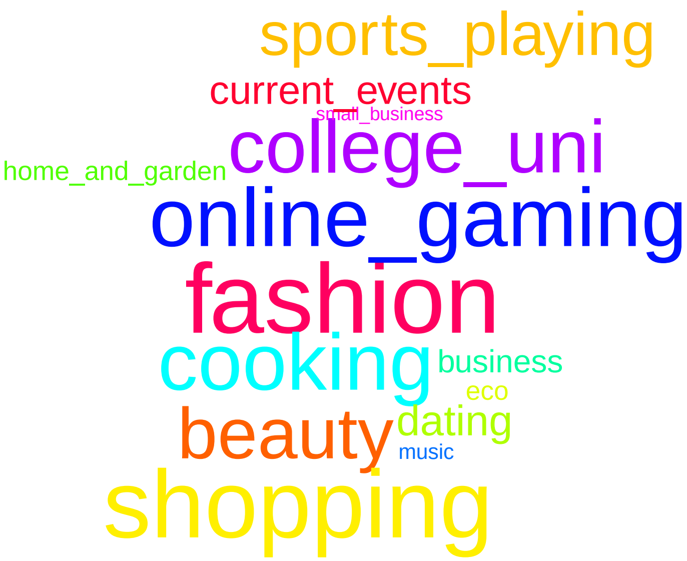
```r
set.seed(1)
require("RColorBrewer")
wordcloud(colnames(sm_sub),
          kmeanspp_sm5$centers[1,],min.freq=0,
          max.words=1000,
          colors = rainbow(32),ordered.colors = TRUE)
```



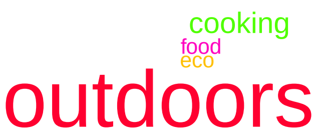
```
wordcloud(colnames(sm_sub),
          kmeanspp_sm5$centers[2,],min.freq=0,
          max.words=1000,
          colors = rainbow(32),ordered.colors = TRUE)
```



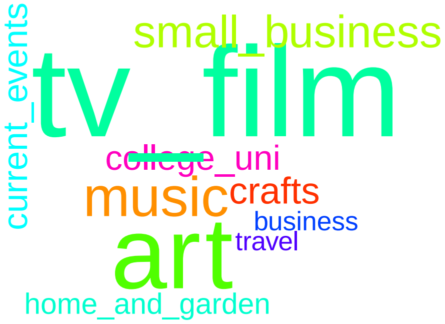
```
wordcloud(colnames(sm_sub),
          kmeanspp_sm5$centers[3,],
          min.freq=0,max.words=1000,
          colors = rainbow(32),ordered.colors = TRUE)
```



```
wordcloud(colnames(sm_sub),
          kmeanspp_sm5$centers[4,],min.freq=0,
          max.words=1000,
          colors = rainbow(32),ordered.colors = TRUE)
```



```
wordcloud(colnames(sm_sub),
          kmeanspp_sm5$centers[5,],min.freq=0,
          max.words=1000,
          colors = rainbow(32),ordered.colors = TRUE)
```


tv_film
art
small_business
music
current_events
college_uni
crafts
business
travel
home_and_garden