R_HW1

Monika_Lodha, Charvi_Mittal,Ashwin_baabu_Paramasivan

August 14, 2017

PART A

Let Random Clicker be R
Truthful Clicker be T
We know from the survey result that P(Y)= 0.65 and P(N) = 0.35
Also expected fraction of random clickers is 0.3
Therefore, P(R) = 0.3
Hence, P(T) = 1-0.3 = 0.7 (from total probability)

Given that Random clickers would click either one with equal probability
So, P(Y|R) = 0.5
P(N|R) = 0.5

P(Y|T) = ??
P(Y|T) = (P(Y)*P(T|Y))/P(T)

P(T|Y) is unknown in this equation

P(Y|R) = (P(R|Y)*P(Y))/P(R)*
*0.5 = (P(R|Y)*0.65)/0.3
P(R|Y) = (0.5*0.3)/0.65
P(R|Y) = 0.15/0.65

Therefore, P(T|Y) = 1-(0.15/0.65)
P(T|Y) = 0.5/0.65

P(Y|T) = (P(Y)*P(T|Y))/P(T)*
*= (0.65*0.5)/(0.65*0.7)*
= 5/7

5 out of 7 truthful clickers answered yes.

PART B

Let disease be D
Postive be P
Negative be N
No disease be ND

```
P(P|D) = 0.993
P(N|ND) = 0.9999
P(D) = 0.000025

P(D|P) = ??

P(D|P) = (P(P|D) * P(D))/ P(P)

P(P) = P(P,D) + P(P,ND)
     = P(P|D) * P(D) + P(P|ND) * P(ND)
     = (0.993 * 0.000025) + (0.0001 * 0.999975)
     = 0.000024825 + 0.0000999975
P(P) = 0.0001248225
```

P(D|P) = (0.993 * 0.000025) / 0.0001248225
= .000024825 / 0.0001248225
= 0.1988

The probability that they have the disease if tested positive is 0.1988

There is a huge possibility of false positives due to which this test can't be accepted universally.

Green Buildings

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.3
```

```
library(plotly)
```

```
## Warning: package 'plotly' was built under R version 3.3.3
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

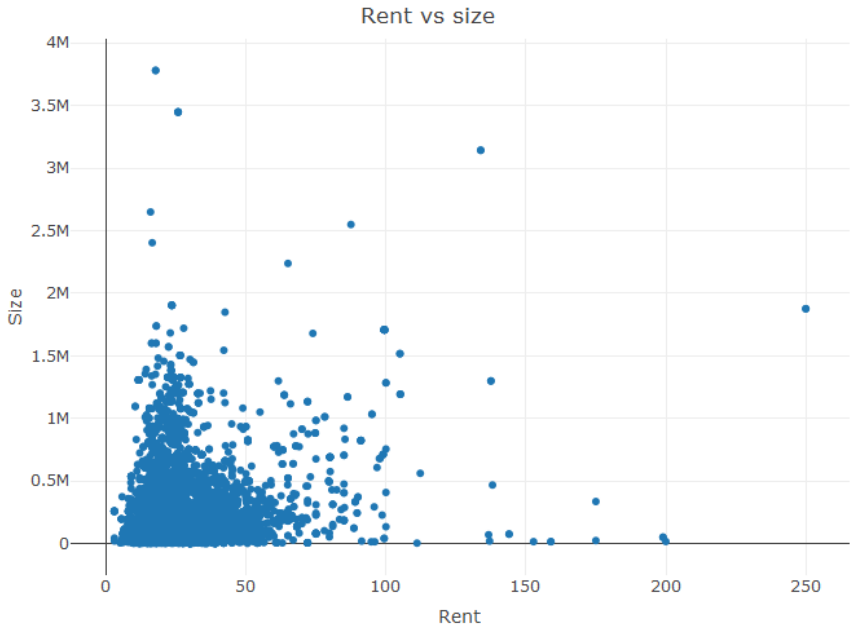
```
####importing csv
green_build<-read.csv('greenbuildings.csv',header = TRUE,stringsAsFactors = FALSE)
green_build$green_rating<-as.character(green_build$green_rating)
green_build$amenities<-as.character(green_build$amenities)
green_build$cluster<-as.character(green_build$cluster)
green_build$net<-as.character(green_build$net)
green_build$renovated<-as.character(green_build$renovated)
#### Let's combine class a and class b
green_build$class<-ifelse(green_build$class_a == 1,'class_a',ifelse(green_build$cl
ass_b == 1,'class_b','class_c'))
####removing unwanted columns
green_build<-green_build[,-c(10,11)]
```

Data cleanup complete

Now let's find out some correlations.

How 'Stories' is related to 'size' of building?

```
p <- plot_ly(green_build, x = green_build$Rent, y = green_build$size) %>%
   layout(
    title = "Rent vs size",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "Size")
    )
```

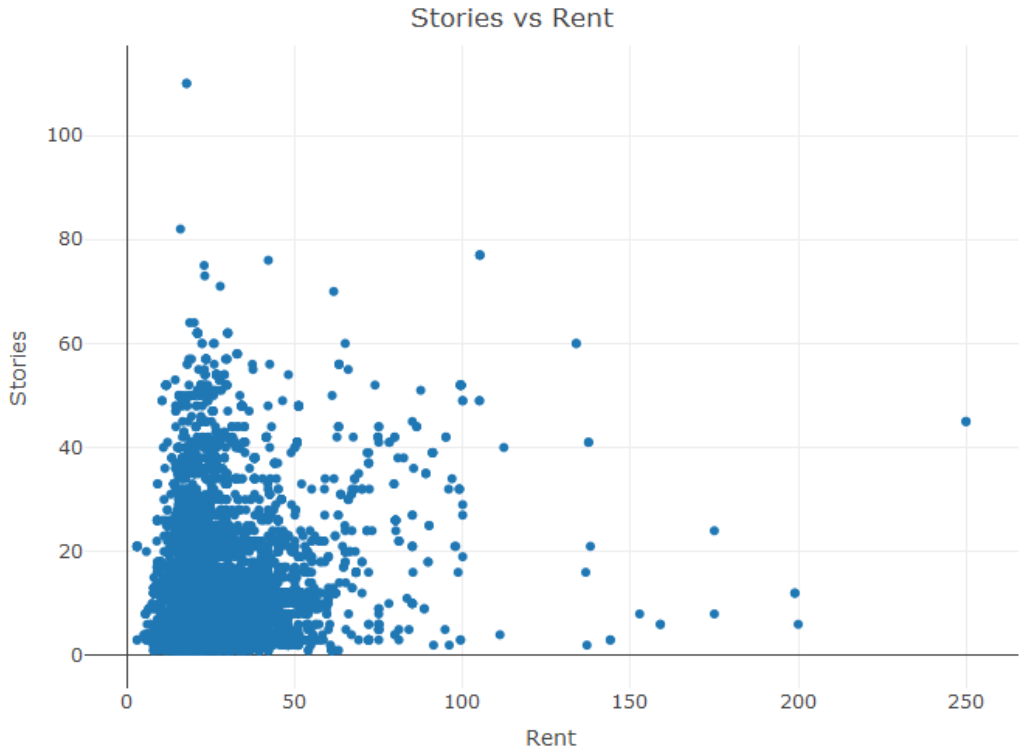


We can see a clear relation between the two. If a building is small in size, rent per square foot can also be less. So it might not be a good estimator of rent for huge building.

Moving on to other variables, and checking relation between few more variables.

Rent and Stories

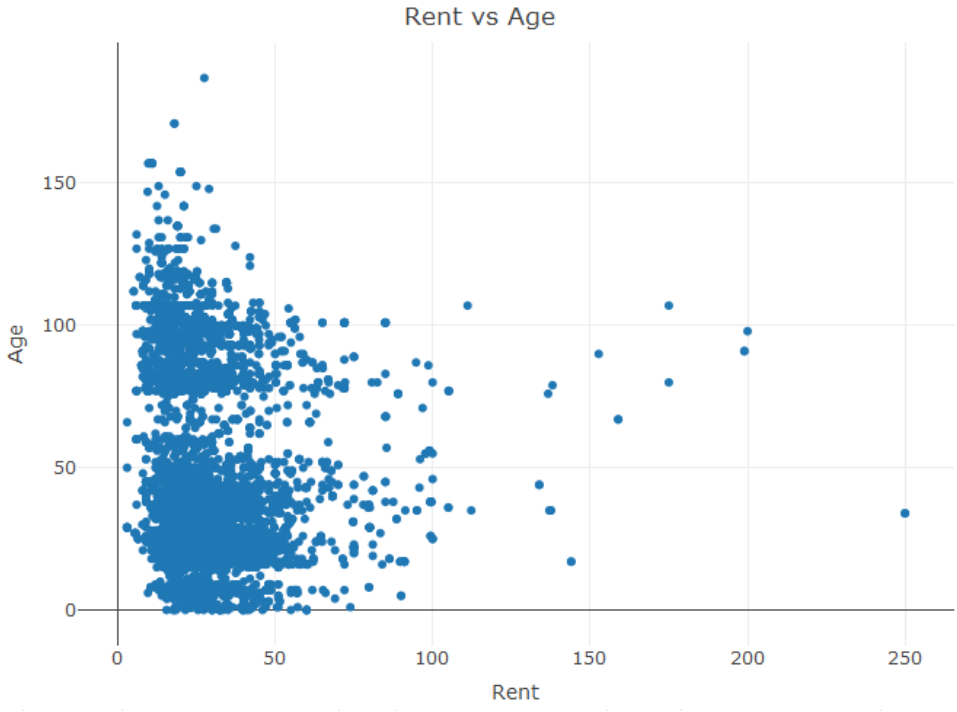
```
temp <- green_build %>% select(stories,Rent)
temp <- na.omit(temp)
p<-plot_ly(temp, x = temp$Rent, y = temp$stories)%>%
   layout(
    title = "Stories vs Rent",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "Stories")
    )
```



Stories can also affect the rent as we can see.

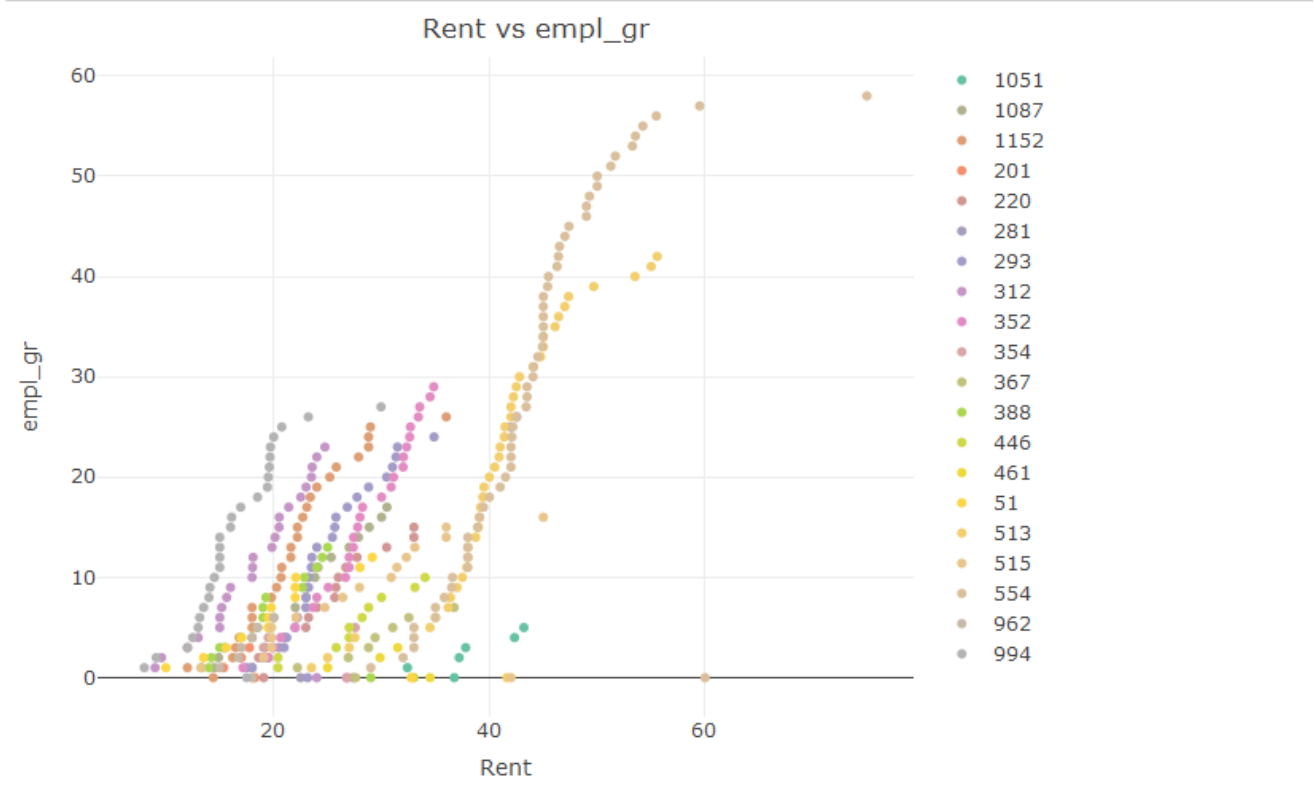
Rent and age

```
temp <- green_build %>% select(age,Rent)
temp <- na.omit(temp)
p<-plot_ly(temp, x = temp$Rent, y = temp$age)%>%
   layout(
    title = "Rent vs Age",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "Age")
    )
```



As the age of the building increases, it's rent goes donw. This effect has not been considered by the Excel guru. The rent has been considered constant over 30 years which might not be a correct assumption.

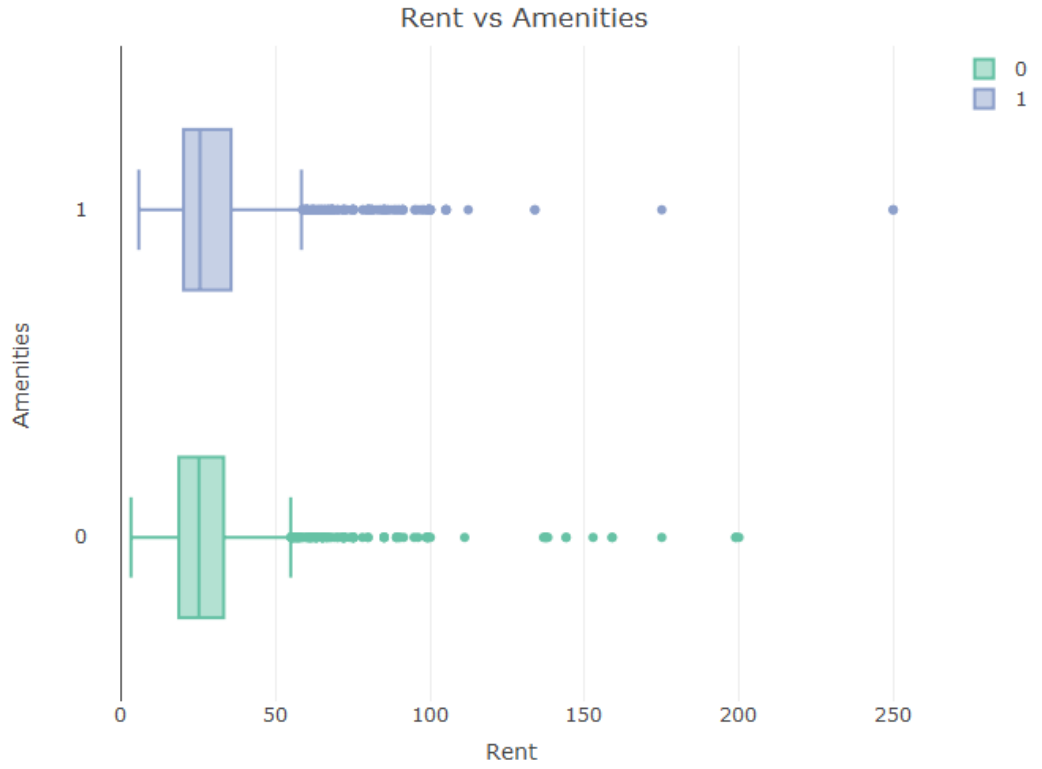
```
set.seed(100)
cluster_random <- sample(green_build$cluster,20)
temp<-green_build[green_build$cluster %in% cluster_random,]
p<-plot_ly(temp, x = temp$Rent, y = temp$empl.gr,color = temp$cluster)%>%
   layout(
    title = "Rent vs empl_gr",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "empl_gr")
    )
```



Here we see a clear clustering pattern while considering emplyment growth rate which tells us that it is an important factor. Rent increases as employment growth increases.

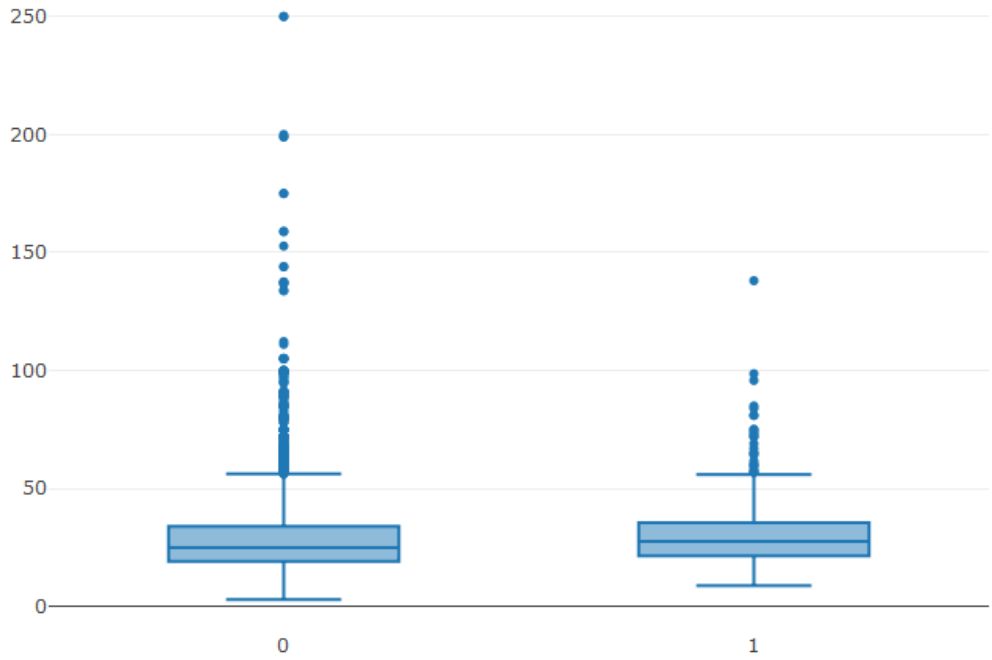
Another case be that the employment growth of a region can decrease with time and rent can go low because of that. That factor has not been considered by the guru which makes his case weak.

```
temp <- green_build %>% select(amenities,Rent)
temp <- na.omit(temp)
p<-plot_ly(temp, x = temp$Rent, color = temp$amenities, type = "box")%>%
   layout(
    title = "Rent vs Amenities",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "Amenities")
    )
```



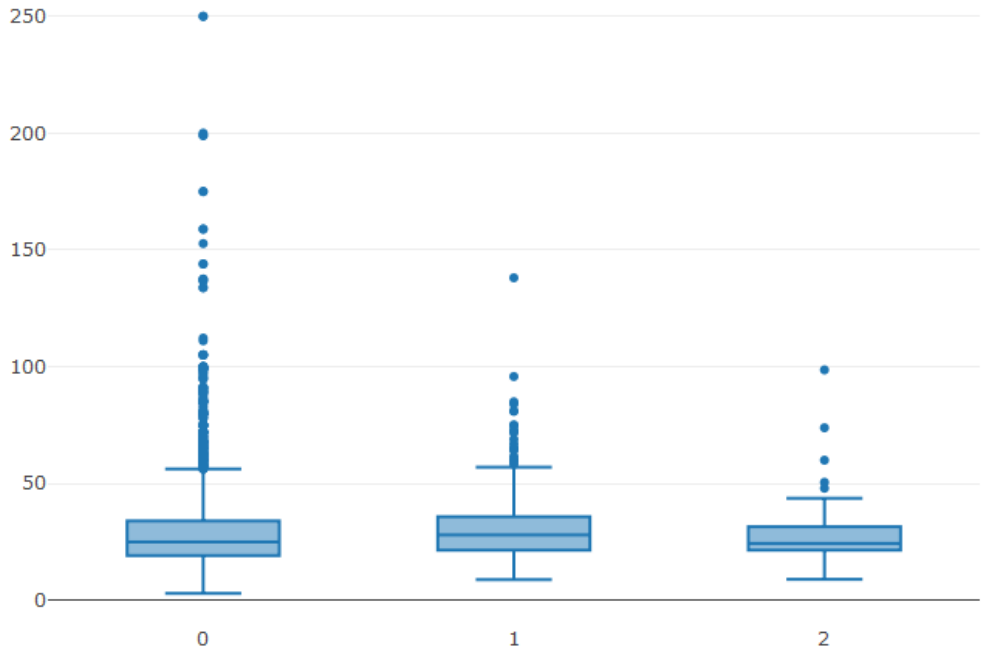
We can see a correlation between amenities and rent. Buildings having atleast one amenity have higher rent on average.

```
green_build$Rating<-ifelse(green_build$Energystar == 1, 1,ifelse(green_build$LEED
== 1,2,0))
#plot_ly(green_build, y=green_build$Rent, x= green_build$green_rating, type = "box"
)
```



We can see that green buildings have high rent compared to non green.

```
#plot_ly(green_build, y=green_build$Rent, x=green_build$Rating,  type = "box")
```

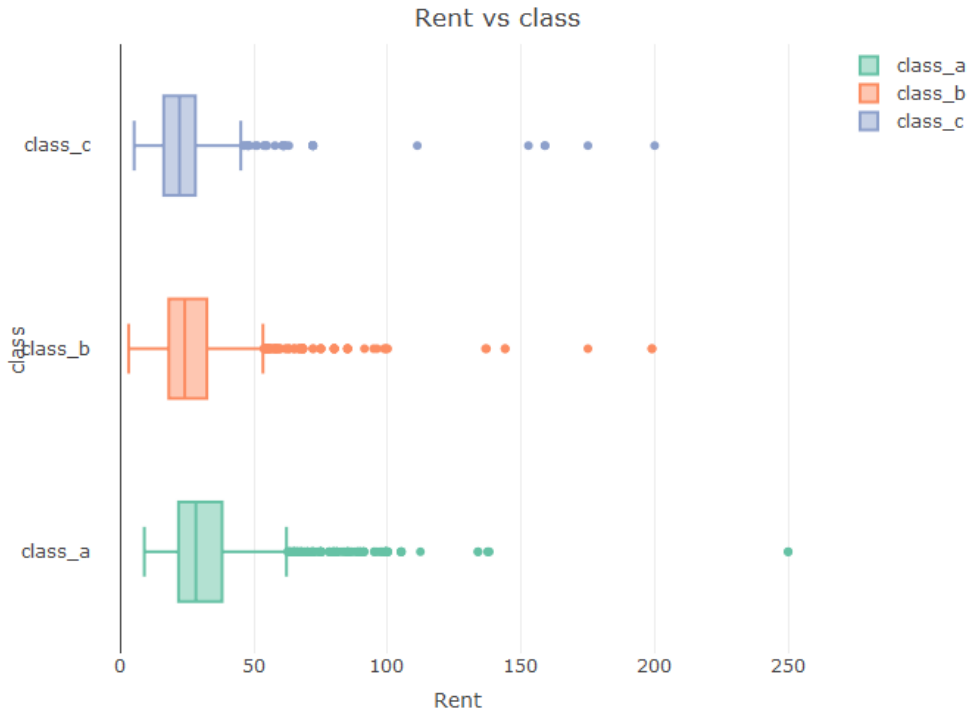


Along with being a green building, it also matters whether it is LEED star rating or ENERGY star rating.

```
temp <- green_build %>% select(class,Rent)
temp <- na.omit(temp)
p<-plot_ly(temp, x = temp$Rent, color = temp$class, type = "box")%>%
   layout(
    title = "Rent vs class",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "class")
    )

summary(green_build)
```

```
##  CS_PropertyID       cluster                size             empl_gr
##  Min.   :      1   Length:7894        Min.   :   1624   Min.   :-24.950
##  1st Qu.: 157452   Class :character   1st Qu.:  50891   1st Qu.:  1.740
##  Median : 313253   Mode  :character   Median : 128838   Median :  1.970
##  Mean   : 453003                      Mean   : 234638   Mean   :  3.207
##  3rd Qu.: 441188                      3rd Qu.: 294212   3rd Qu.:  2.380
##  Max.   :6208103                      Max.   :3781045   Max.   : 67.780
##                                                         NA's   :74
##       Rent          leasing_rate       stories            age
##  Min.   :  2.98   Min.   :  0.00   Min.   :  1.00   Min.   :  0.00
##  1st Qu.: 19.50   1st Qu.: 77.85   1st Qu.:  4.00   1st Qu.: 23.00
##  Median : 25.16   Median : 89.53   Median : 10.00   Median : 34.00
##  Mean   : 28.42   Mean   : 82.61   Mean   : 13.58   Mean   : 47.24
##  3rd Qu.: 34.18   3rd Qu.: 96.44   3rd Qu.: 19.00   3rd Qu.: 79.00
##  Max.   :250.00   Max.   :100.00   Max.   :110.00   Max.   :187.00
##
##   renovated             LEED            Energystar
##  Length:7894        Min.   :0.000000   Min.   :0.00000
##  Class :character   1st Qu.:0.000000   1st Qu.:0.00000
##  Mode  :character   Median :0.000000   Median :0.00000
##                     Mean   :0.006841   Mean   :0.08082
##                     3rd Qu.:0.000000   3rd Qu.:0.00000
##                     Max.   :1.000000   Max.   :1.00000
##
```

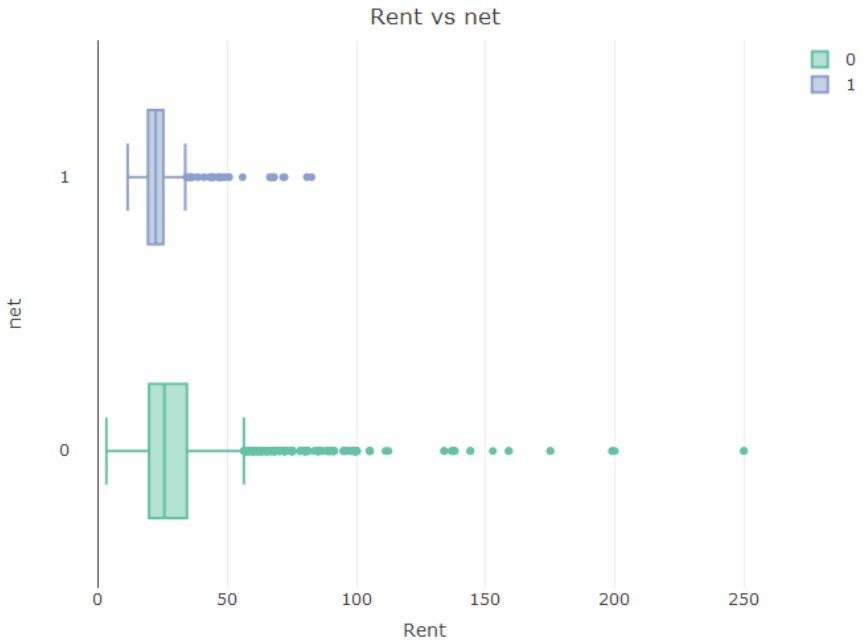
```
##
##  green_rating          net             amenities         cd_total_07
##  Length:7894        Length:7894        Length:7894        Min.   :  39
##  Class :character   Class :character   Class :character   1st Qu.: 684
##  Mode  :character   Mode  :character   Mode  :character   Median : 966
##                                                           Mean   :1229
##                                                           3rd Qu.:1620
##                                                           Max.   :5240
##
##    hd_total07    total_dd_07   Precipitation     Gas_Costs
##  Min.   :   0   Min.   :2103   Min.   :10.46   Min.   :0.009487
##  1st Qu.:1419   1st Qu.:2869   1st Qu.:22.71   1st Qu.:0.010296
##  Median :2739   Median :4979   Median :23.16   Median :0.010296
##  Mean   :3432   Mean   :4661   Mean   :31.08   Mean   :0.011336
##  3rd Qu.:4796   3rd Qu.:6413   3rd Qu.:43.89   3rd Qu.:0.011816
##  Max.   :7200   Max.   :8244   Max.   :58.02   Max.   :0.028914
##
##  Electricity_Costs  cluster_rent      Compare          class
##  Min.   :0.01780   Min.   : 9.00   Min.   :0.0000   Length:7894
##  1st Qu.:0.02330   1st Qu.:20.00   1st Qu.:0.0000   Class :character
##  Median :0.03274   Median :25.14   Median :0.0000   Mode  :character
##  Mean   :0.03096   Mean   :27.50   Mean   :0.1667
##  3rd Qu.:0.03781   3rd Qu.:34.00   3rd Qu.:0.0000
##  Max.   :0.06280   Max.   :71.44   Max.   :2.0000
##
##      Rating
##  Min.   :0.00000
##  1st Qu.:0.00000
##  Median :0.00000
##  Mean   :0.09273
##  3rd Qu.:0.00000
##  Max.   :2.00000
##
```



We can see that Class A has highest rent, followed by class B and then class C because Class A has better quality buildings compared to Class B and Class C. This also shows that Class can be a big contributor in high rent of green buildings.

Hence, just having a green building is not a sole predictor of rent.

```
temp <- green_build %>% select(net,Rent)
temp <- na.omit(temp)
p<-plot_ly(temp, x = temp$Rent, color = temp$net, type = "box")%>%
   layout(
    title = "Rent vs net",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "net")
    )
```

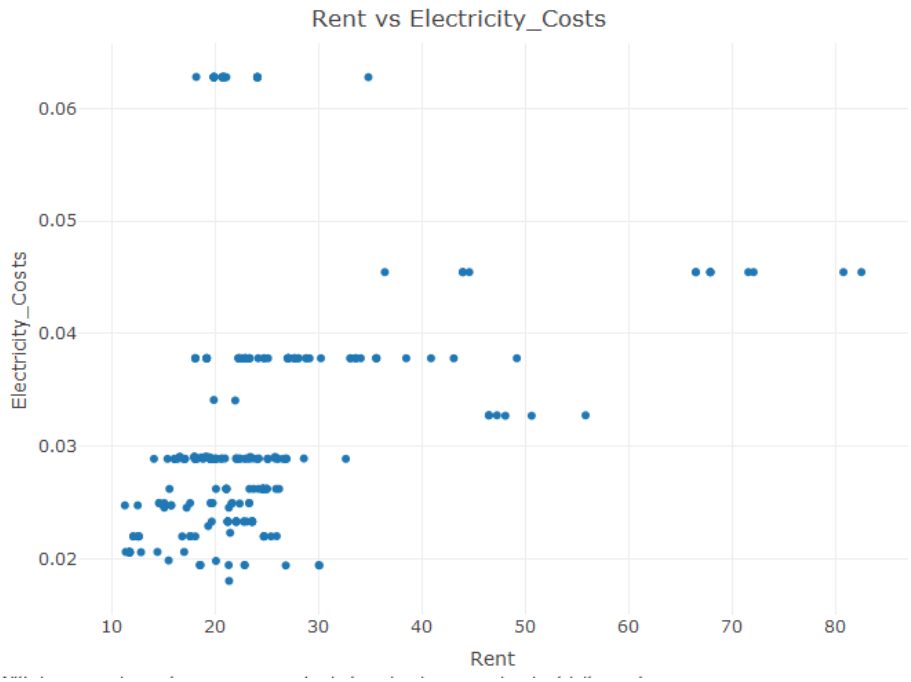


Clearly, if the tenants are paying the utility bill on their own, the rent they have to pay will be lower. But that doesn't give any insight about whether it's going to tell us whether green building is better or not.

```
temp <- green_build %>% filter(net == 1)
```

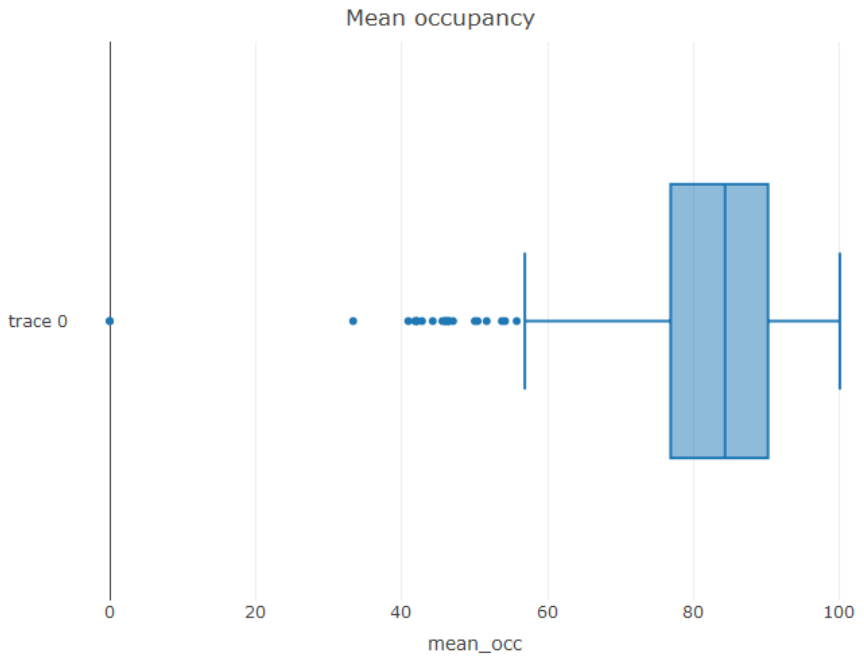
```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

```
temp1 <- temp %>% select(Rent,Electricity_Costs)
p<-plot_ly(temp1, x = temp1$Rent, y = temp1$Electricity_Costs)%>%
   layout(
    title = "Rent vs Electricity_Costs",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "Electricity_Costs")
    )
```



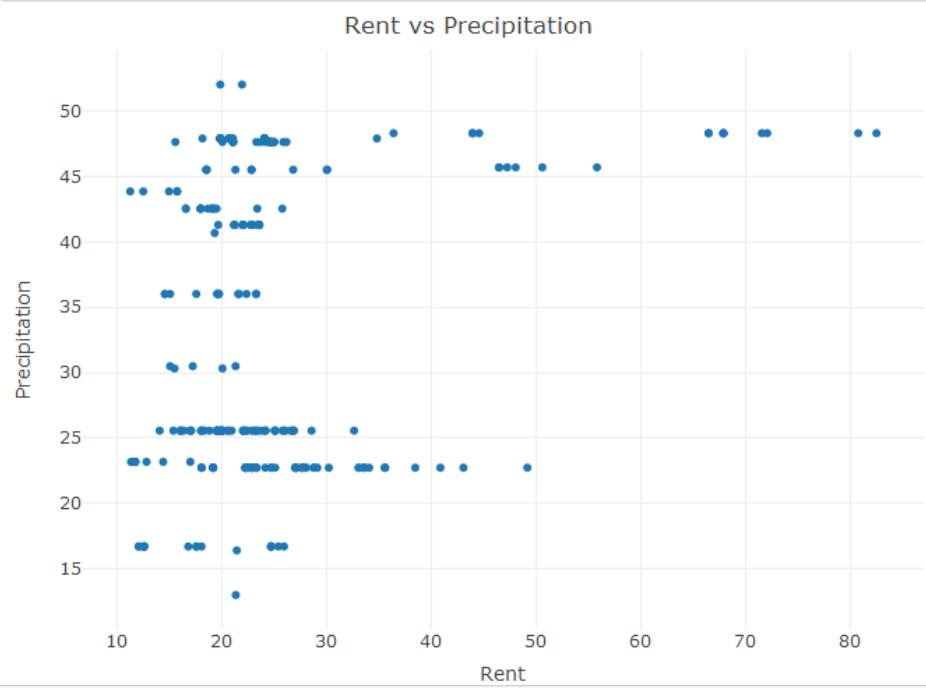
With incraese in rent, we see a gradual stepwise increase in electricity costs.

```
green_build_occ <- green_build %>% select(cluster,leasing_rate) %>% group_by(cluste
r) %>%  summarise(mean_occ = mean(leasing_rate))
p<-plot_ly(green_build_occ, x = green_build_occ$mean_occ, type = "box")%>%
   layout(
    title = "Mean occupancy",
      xaxis = list(title = "mean_occ")
    )
```

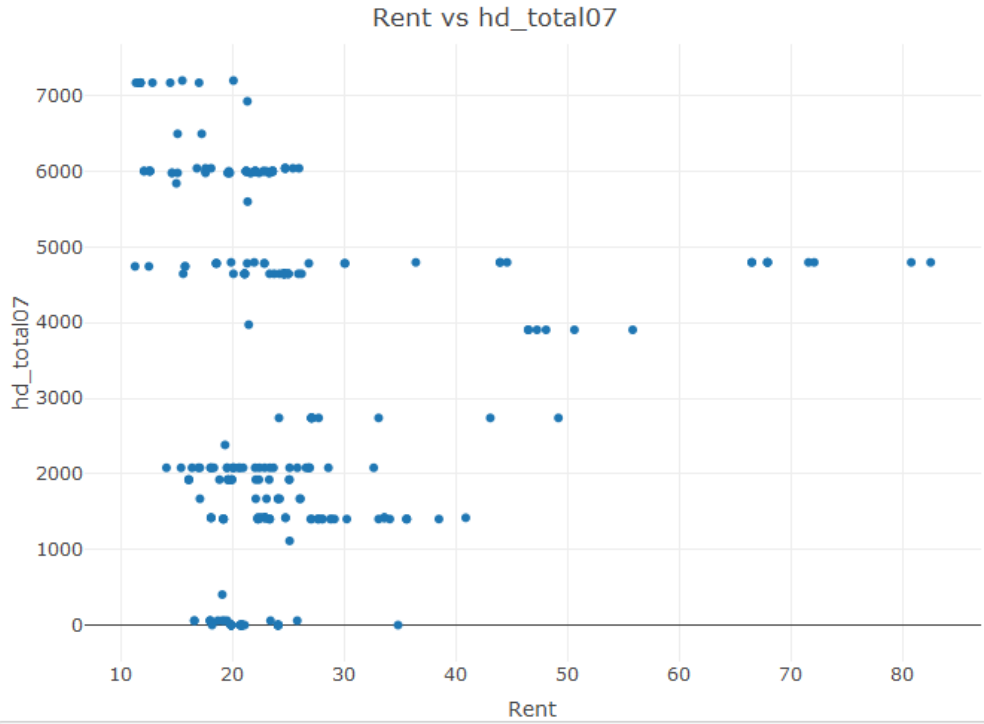


The mean of occupancy from the above box plot is 84% which shows that the prediction of minimum occupancy of 90% is a bit on the higher end. We can assume the occupancy to be around 75% for prediction purpose.

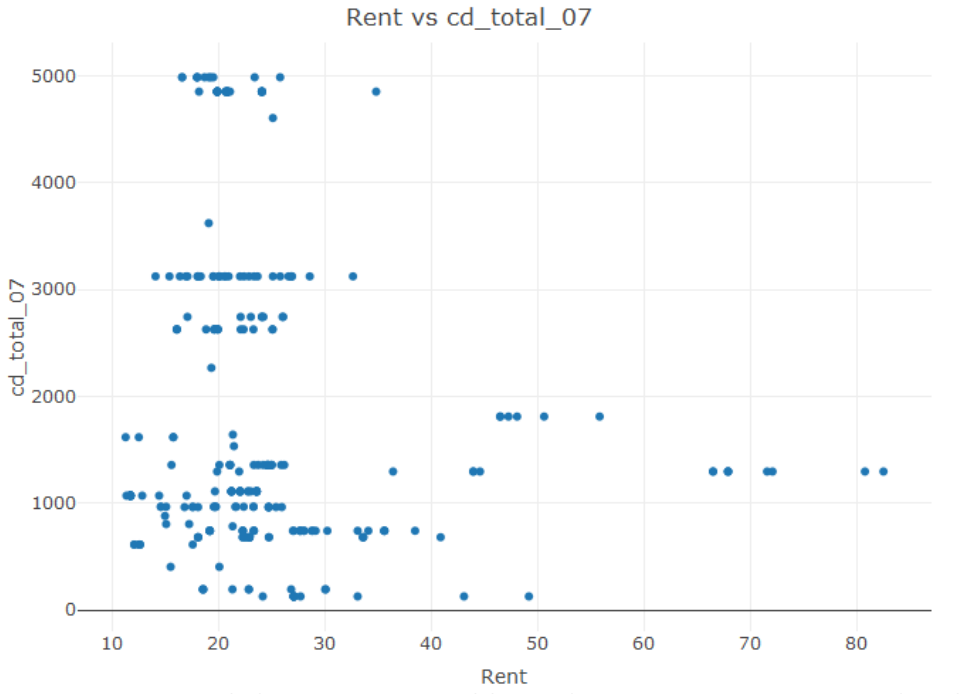
```
temp1 <- temp %>% select(Rent,Precipitation)
p<-plot_ly(temp1, x = temp1$Rent, y = temp1$Precipitation)%>%
   layout(
    title = "Rent vs Precipitation",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "Precipitation")
    )
```




```
temp1 <- temp %>% select(Rent,hd_total07)
p<-plot_ly(temp1, x = temp1$Rent, y = temp1$hd_total07)%>%
   layout(
    title = "Rent vs hd_total07",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "hd_total07")
    )
```



```
temp1 <- temp %>% select(Rent,cd_total_07)
p<-plot_ly(temp1, x = temp1$Rent, y = temp1$cd_total_07)%>%
   layout(
    title = "Rent vs cd_total_07",
      xaxis = list(title = "Rent"),
      yaxis = list(title = "cd_total_07")
    )
```



We see an increase in rent with an increase in Precipitation, Heating days and cooling days (ignoring the outliers)

After checking for all the correlations and variable reductions, we decided to go ahead with employment growth rate, stories, Precipitation and heating and cooling days.

Hence we look for these values for our new building on East Cesar Chavez.

Now we try to classify our new building into a cluster. First removing missing values

```
greendata=green_build[which(!(is.na(green_build$empl_gr))),]
```

Next we classify the building using least rmse

```
table = greendata %>% group_by(cluster) %>% summarise(mean_empl_gr = mean(empl_gr),
                      mean_cd = mean(cd_total_07),
                      mean_hd = mean(hd_total07),
                      mean_ppt = mean(Precipitation),
                      mean_stories = mean(stories))
table['stories'] = 15
table['emply_growth']   = 2.8
table['Precipitation_new']  = 34.25
table['Heating_degree_days'] =  1541
table['Cooling_degree_days']    = 3130
x1 = max(table[,2])-min(table[,2])
x2 = max(table[,3])-min(table[,3])
x3 = max(table[,4])-min(table[,4])
x4 = max(table[,5])-min(table[,5])
x5 = max(table[,6])-min(table[,6])
View(table)
table['rmse1']= ((table[,2]-table[,8])/x1)^2
table['rmse2']= ((table[,3]-table[,11])/x2)^2
table['rmse3']= ((table[,4]-table[,10])/x3)^2
table['rmse4']= ((table[,5]-table[,9])/x4)^2
table['rmse5']= ((table[,6]-table[,7])/x5)^2

table['final_rmse']=sqrt((table[,12]+table[,13]+table[,14]+table[,15]+table[,16])/
5)
max(table['final_rmse'])
```

```
## [1] 0.4637667
```

```
min(table['final_rmse'])
```

```
## [1] 0.09069835
```

```
newdata <- table[order(table$final_rmse),]
View(newdata)
final_data = newdata %>% select(cluster,final_rmse)
final_cluster= green_build %>% filter(cluster == 394)
View(final_cluster)
```

In the final cluster, we can see that green building has the highest rent but we can also see that the buildings having highest rent are also Class A buildings. As we have seen above, CLass and rent have high correlation which can be a cause for high rent.This relation was not considered in Excel guru's exploration which can be misleading.

The maximum for non green building in cluster 394 is $25.46 and for green building it's $29.37 which gives us extra $3.91 dollars per square foot of the building. Which is higher than what the Excel guru estimated. Hence, we can get the investment back in (5000000/(2500003.91.75)) 6.8 years considering the occupancy of 75%.

But there are few assumptions that need to be considered here:
The employment growth rate will not go down. In 30 years, employment growth can go up or down. It is assumed constant. Also an important factor to be considered is the time value of money and the rent in a particular geographic location changes depending on the development of that location and what impact it goes through, in the years. As there is no sufficient details provided on these parameters it is safe to

assume our hypothesis will hold in the given conditions.

BOOTSTRAPPING

```
library(mosaic)
```

```
## Warning: package 'mosaic' was built under R version 3.3.3
```

```
## Warning: package 'ggformula' was built under R version 3.3.3
```

```
## Warning: package 'mosaicData' was built under R version 3.3.3
```

```
library(quantmod)
```

```
## Warning: package 'quantmod' was built under R version 3.3.3
```

```
## Warning: package 'xts' was built under R version 3.3.3
```

```
## Warning: package 'zoo' was built under R version 3.3.3
```

```
## Warning: package 'TTR' was built under R version 3.3.3
```

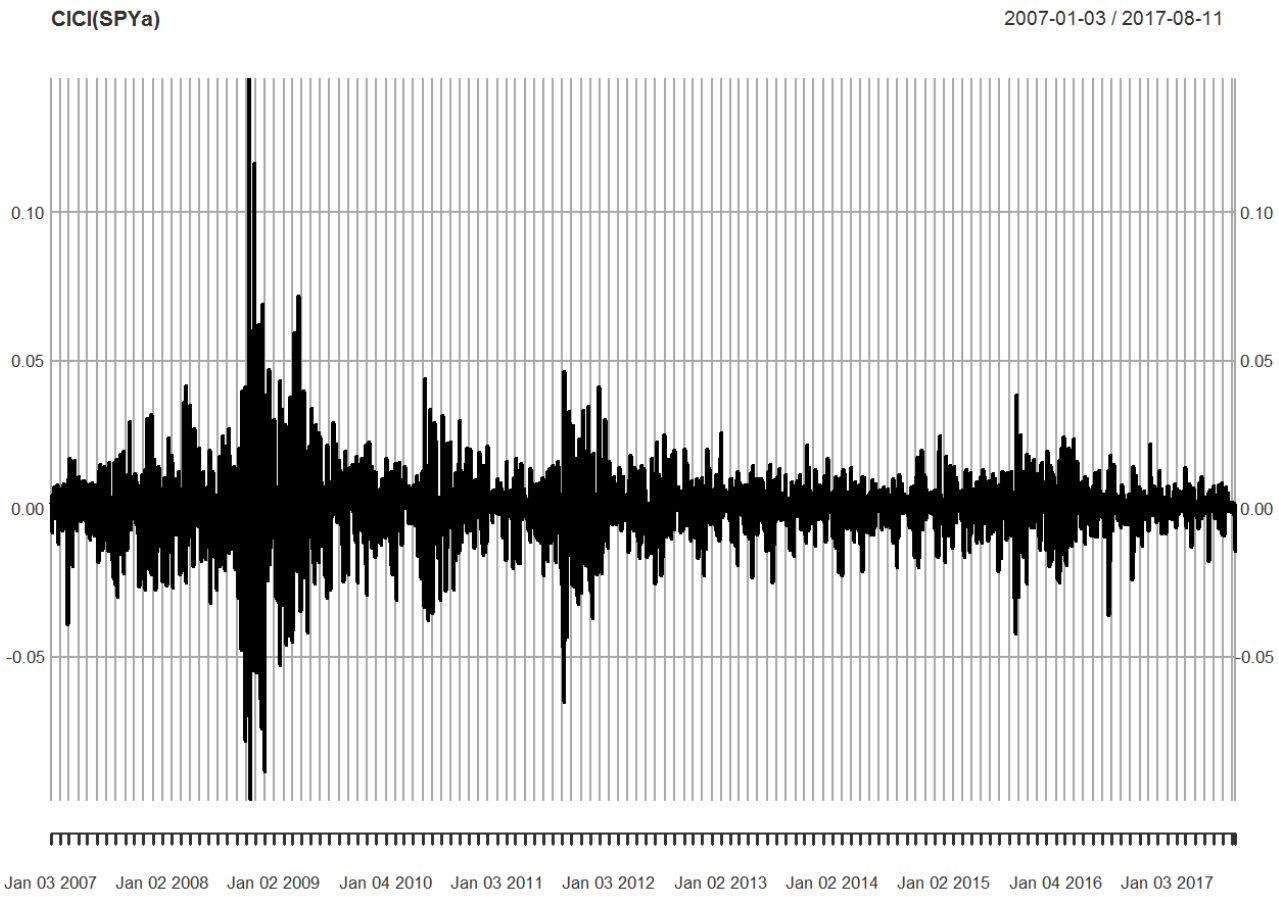
```
library(foreach)
```

```
## Warning: package 'foreach' was built under R version 3.3.3
```

```
library(plotly)
```

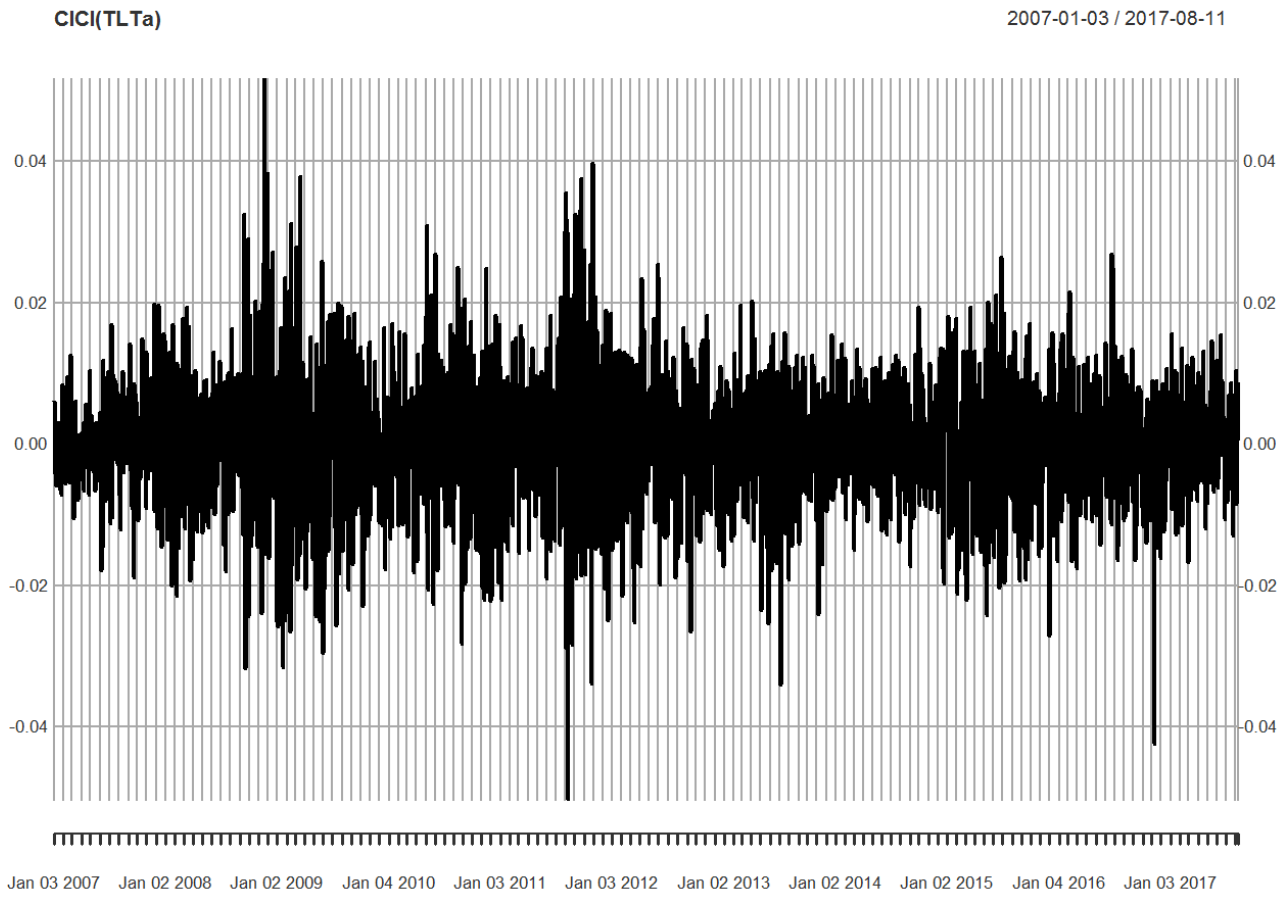
```
mystocks = c("SPY", "TLT", "LQD", "EEM", "VNQ")
myprices = getSymbols(mystocks, from = "2007-01-01")

# Adjust for splits and dividends
SPYa = adjustOHLC(SPY)
TLTa = adjustOHLC(TLT)
LQDa = adjustOHLC(LQD)
EEMa = adjustOHLC(EEM)
VNQa = adjustOHLC(VNQ)
```

```
plot(ClCl(SPYa))
```



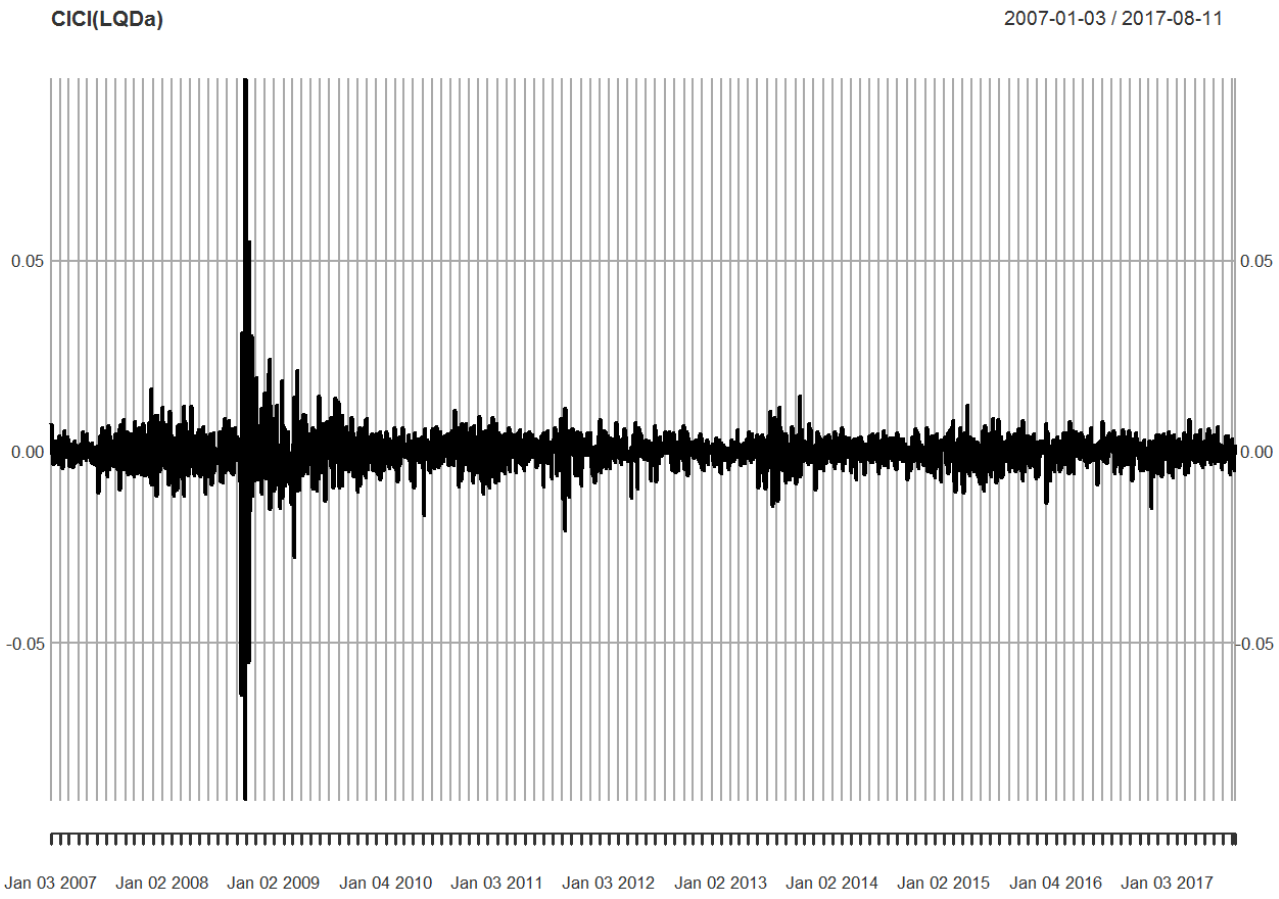
There is moderate risk/return assiciated with SPY as the percent difference between everyday closing price does not seem to be too high

```
plot(ClCl(TLTa))
```



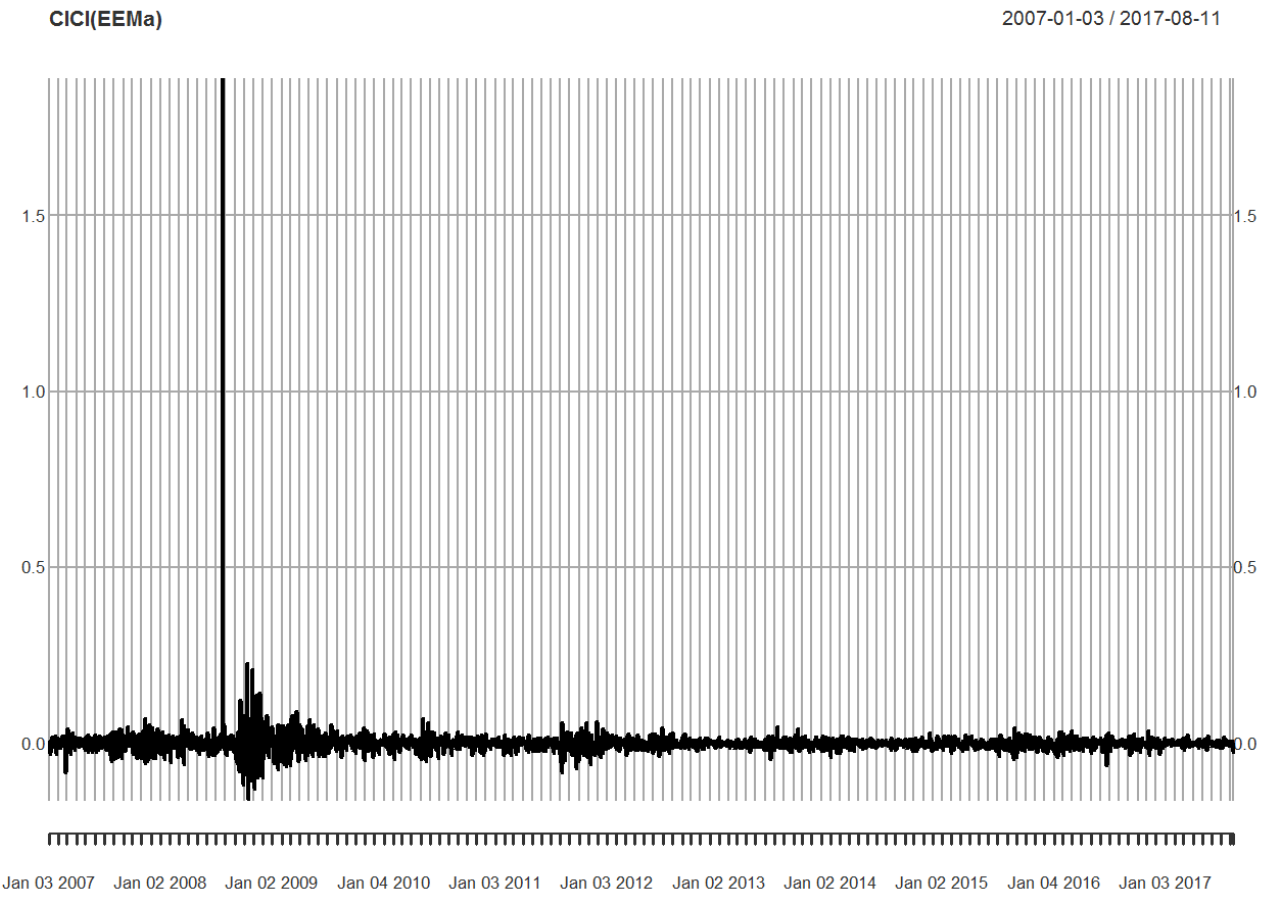
TLT follows the most fluctuating trend with a large difference between closing prices everyday. Hence, there is higher risk/return associated with such assets

```
plot(ClCl(LQDa))
```



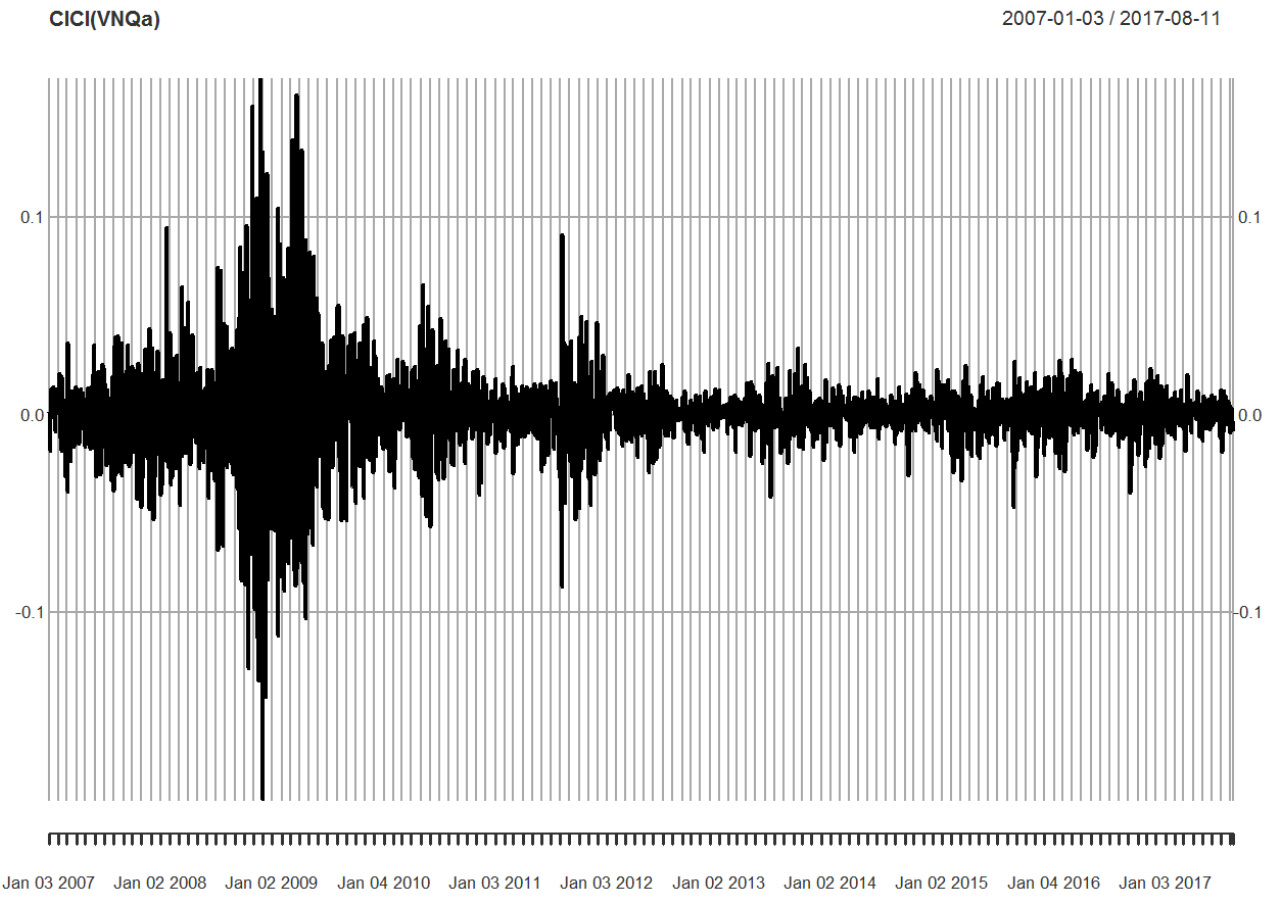
Trend does not seem to be fluctuating, low risk/return

```
plot(ClCl(EEMa))
```



Trend seems almost flat. Hence, lowest risk/return associated

```
plot(ClCl(VNQa))
```



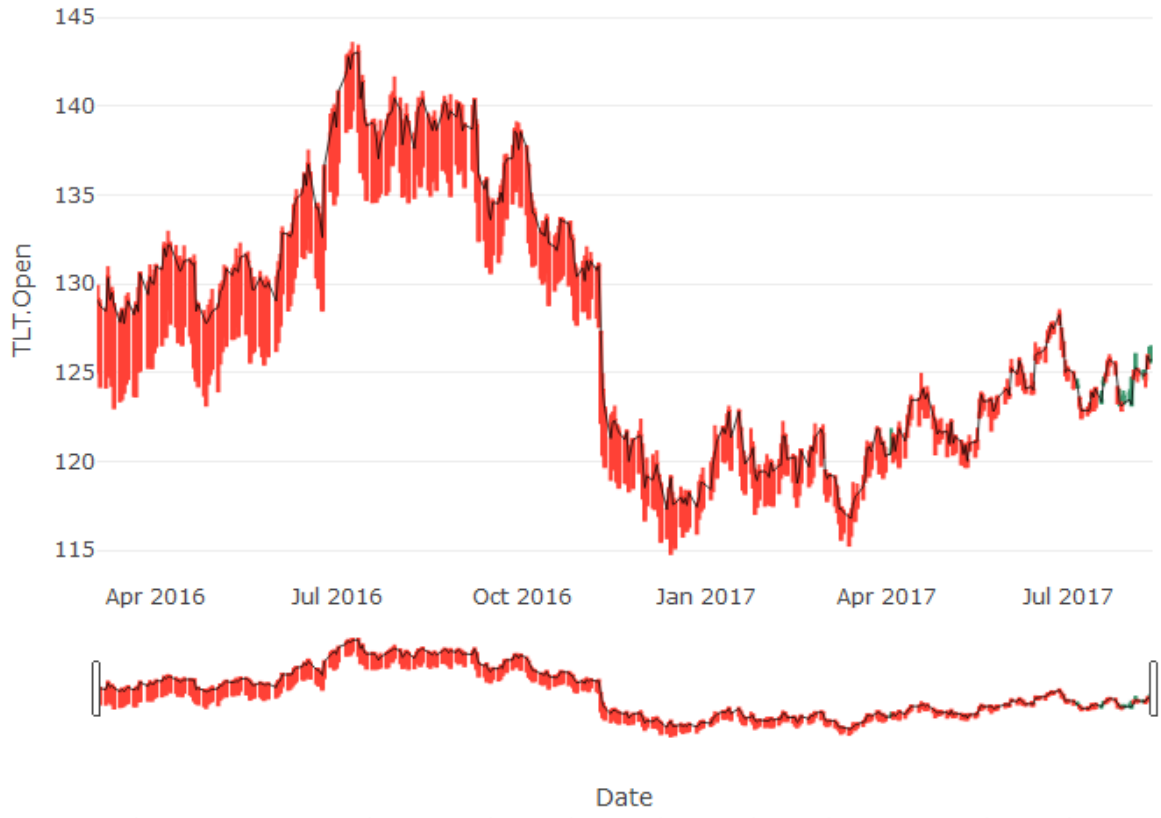
Behaves just like SPY, moderate risk/return associated

Performance of assets over last year

```
df <- data.frame(Date=index(SPYa),coredata(SPYa))
df <- tail(df, 365)
p <- df %>%
  plot_ly(x = ~Date, type="candlestick",
          open = ~SPY.Open, close = ~SPY.Close,
          high = ~SPY.High, low = ~SPY.Low) %>%
  add_lines(y = ~SPY.Open, line = list(color = 'black', width = 0.75)) %>%
  layout(showlegend = FALSE)
```

Trend shows asset price increasing over time with moderate risk/return

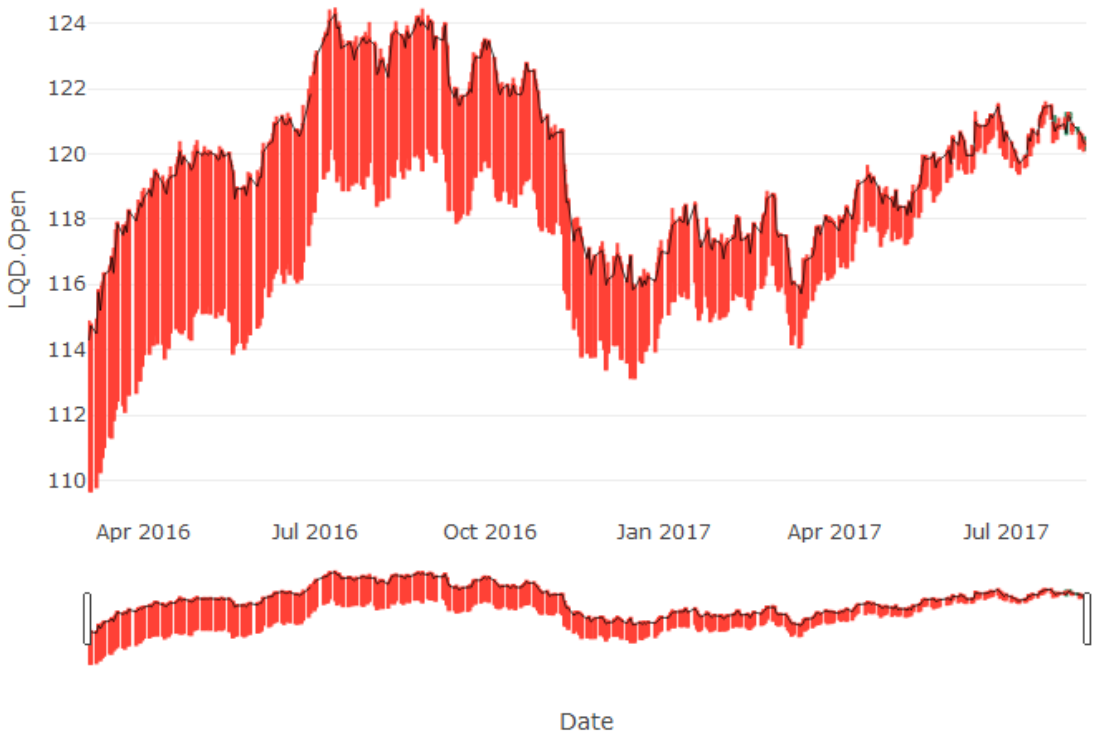
```
df <- data.frame(Date=index(TLTa),coredata(TLTa))
df <- tail(df, 365)
p <- df %>%
  plot_ly(x = ~Date, type="candlestick",
          open = ~TLT.Open, close = ~TLT.Close,
          high = ~TLT.High, low = ~TLT.Low) %>%
  add_lines(y = ~TLT.Open, line = list(color = 'black', width = 0.75)) %>%
  layout(showlegend = FALSE)
```



TLT does not show an increasing trend over time. Although it seems to be stabilizing from the past couple of

months

```
df <- data.frame(Date=index(LQDa),coredata(LQDa))
df <- tail(df, 365)
p <- df %>%
  plot_ly(x = ~Date, type="candlestick",
          open = ~LQD.Open, close = ~LQD.Close,
          high = ~LQD.High, low = ~LQD.Low) %>%
  add_lines(y = ~LQD.Open, line = list(color = 'black', width = 0.75)) %>%
  layout(showlegend = FALSE)
```

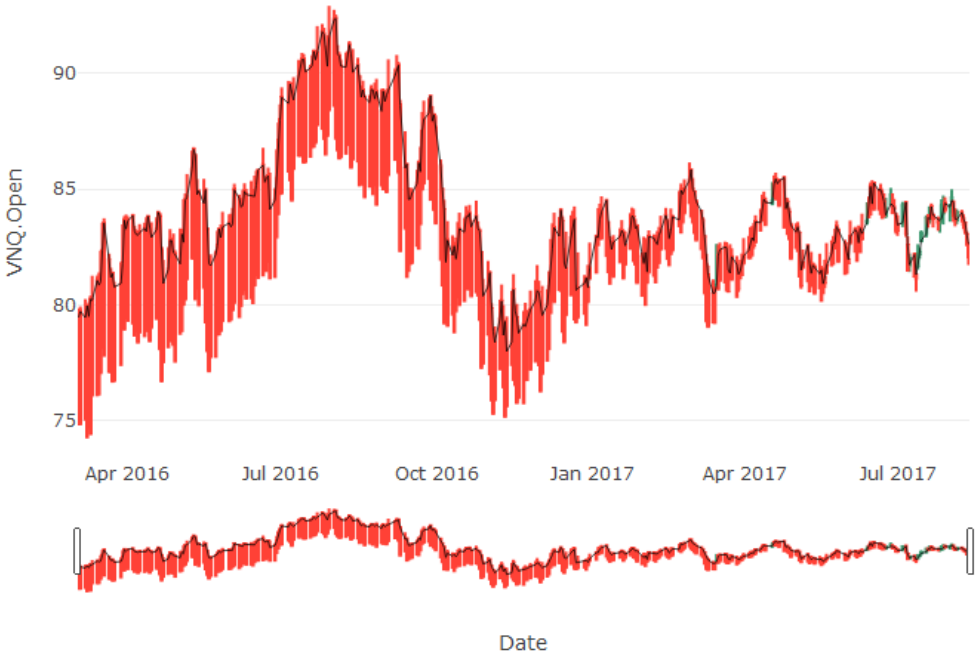


LQD is gaining for the past months giving moderate returns

```
df <- data.frame(Date=index(EEMa),coredata(EEMa))
df <- tail(df, 365)
p <- df %>%
  plot_ly(x = ~Date, type="candlestick",
          open = ~EEM.Open, close = ~EEM.Close,
          high = ~EEM.High, low = ~EEM.Low) %>%
  add_lines(y = ~EEM.Open, line = list(color = 'black', width = 0.75)) %>%
  layout(showlegend = FALSE)
```

EEM shows a steep increase in price over time

```
df <- data.frame(Date=index(VNQa),coredata(VNQa))
df <- tail(df, 365)
p <- df %>%
  plot_ly(x = ~Date, type="candlestick",
          open = ~VNQ.Open, close = ~VNQ.Close,
          high = ~VNQ.High, low = ~VNQ.Low) %>%
  add_lines(y = ~VNQ.Open, line = list(color = 'black', width = 0.75)) %>%
  layout(showlegend = FALSE)
```



Plotting risk/return of individual assets

SPY

```
set.seed(100)
all_returns_spy = cbind(ClCl(SPYa))
all_returns_spy = as.matrix(na.omit(all_returns_spy))
initial_wealth = 100000
sim1_spy = foreach(i=1:1000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(1)
  holdings = weights * total_wealth
  n_days = 20
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return = mosaic::resample(all_returns_spy, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
  }
  wealthtracker
}

mean(sim1_spy[,n_days])
```

```
## [1] 100772.9
```

```
sd(sim1_spy[,n_days])
```

```
## [1] 5686.956
```

TLT

```
set.seed(100)
all_returns_tlt = cbind(ClCl(TLTa))
all_returns_tlt = as.matrix(na.omit(all_returns_tlt))
initial_wealth = 100000
sim1_tlt = foreach(i=1:1000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(1)
  holdings = weights * total_wealth
  n_days = 20
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return = mosaic::resample(all_returns_tlt, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
  }
  wealthtracker
}

mean(sim1_tlt[,n_days])
```

```
## [1] 100769.5
```

```
sd(sim1_tlt[,n_days])
```

```
## [1] 4261.106
```

LQD

```
set.seed(100)
all_returns_lqd = cbind(ClCl(LQDa))
all_returns_lqd = as.matrix(na.omit(all_returns_lqd))
initial_wealth = 100000
sim1_lqd = foreach(i=1:1000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(1)
  holdings = weights * total_wealth
  n_days = 20
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return = mosaic::resample(all_returns_lqd, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
  }
  wealthtracker
}

mean(sim1_lqd[,n_days])
```

```
## [1] 100533.3
```

```
sd(sim1_lqd[,n_days])
```

```
## [1] 2534.052
```

EEM

```
set.seed(100)
all_returns_eem = cbind(ClCl(EEMa))
all_returns_eem = as.matrix(na.omit(all_returns_eem))
initial_wealth = 100000
sim1_eem = foreach(i=1:1000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(1)
  holdings = weights * total_wealth
  n_days = 20
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return = mosaic::resample(all_returns_eem, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
  }
  wealthtracker
}

mean(sim1_eem[,n_days])
```

```
## [1] 102336.3
```

```
sd(sim1_eem[,n_days])
```

```
## [1] 22011.83
```

VNQ

```
set.seed(100)
all_returns_vnq = cbind(ClCl(VNQa))
all_returns_vnq = as.matrix(na.omit(all_returns_vnq))
initial_wealth = 100000
sim1_vnq = foreach(i=1:1000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(1)
  holdings = weights * total_wealth
  n_days = 20
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return = mosaic::resample(all_returns_vnq, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
  }
  wealthtracker
}

mean(sim1_vnq[,n_days])
```

```
## [1] 100875.4
```

```
sd(sim1_vnq[,n_days])
```

```
## [1] 9940.88
```

```
data.frame('Portfolio' = c('SPY','TLT','LQD','EEM','VNQ'), 'Mean' = c(mean(sim1_sp
y[,n_days]),mean(sim1_tlt[,n_days]),mean(sim1_lqd[,n_days]),mean(sim1_eem[,n_days]
),mean(sim1_vnq[,n_days])),
              'Standard Deviation' = c(sd(sim1_spy[,n_days]),sd(sim1_tlt[,n_days])
,sd(sim1_lqd[,n_days]),sd(sim1_eem[,n_days]),
                                       sd(sim1_vnq[,n_days])))
```

```
##   Portfolio     Mean Standard.Deviation
## 1       SPY 100772.9           5686.956
## 2       TLT 100769.5           4261.106
## 3       LQD 100533.3           2534.052
## 4       EEM 102336.3          22011.827
## 5       VNQ 100875.4           9940.880
```

VNQ has moderate risk/return associated with it as is presented by results for the past 1 year

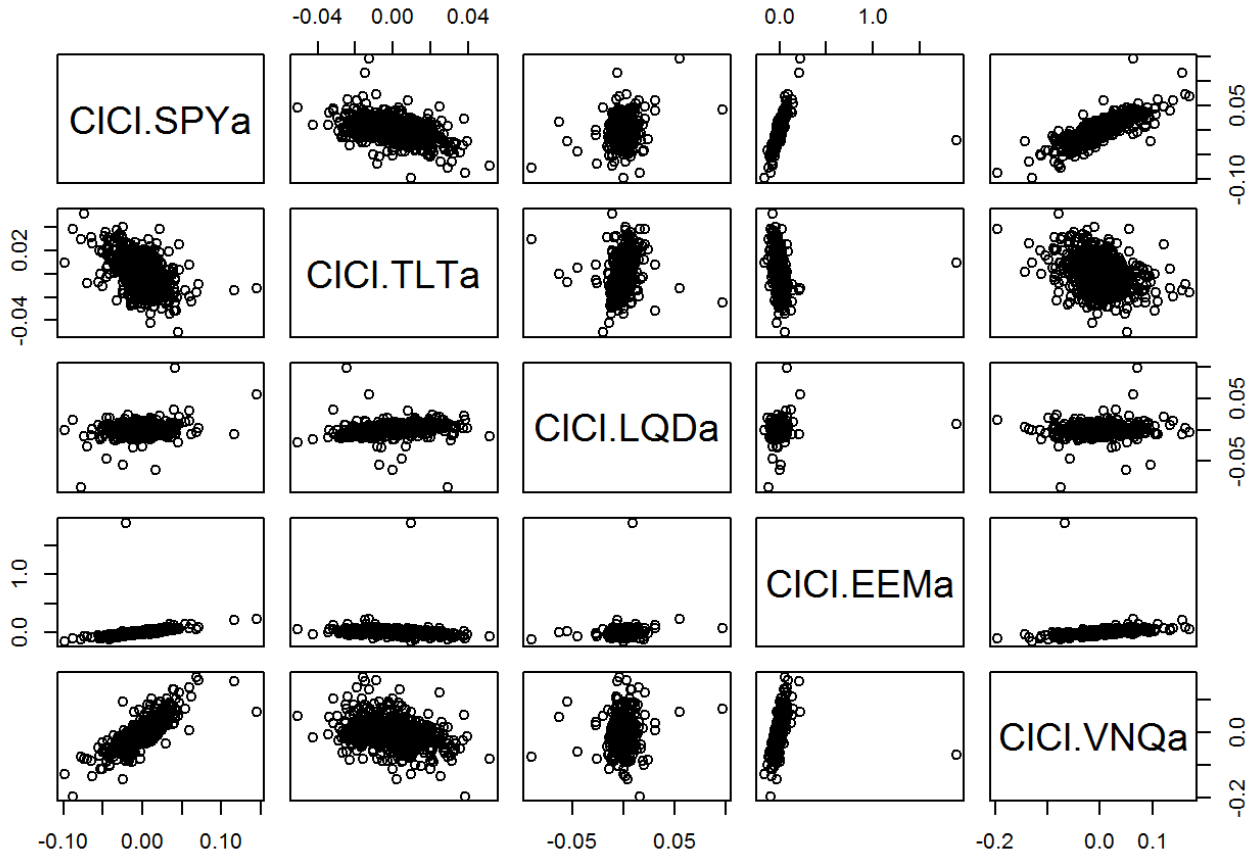
Hence the following portfolio exhibit different porperties 1. Equity(SPY): Gives moderate risk/return with returns increasing over time
2. Treasury bonds (TLT): This follows a fluctuating trend with higher risk/return associated with it.
3. Corporate bonds (LQD): LQD gives low/moderate returns with stable plots 4. Equities (EEM): Shows a fluctuating trend and is highly volatile. Gives highest returns with highest deviation
5. Real Estate (VNQ): this behaves just like SPY with moderate risk/return associated with it.

Correlation

```
# Combine close to close changes in a single matrix
all_returns = cbind(ClCl(SPYa),ClCl(TLTa),ClCl(LQDa),ClCl(EEMa),ClCl(VNQa))
head(all_returns)
```

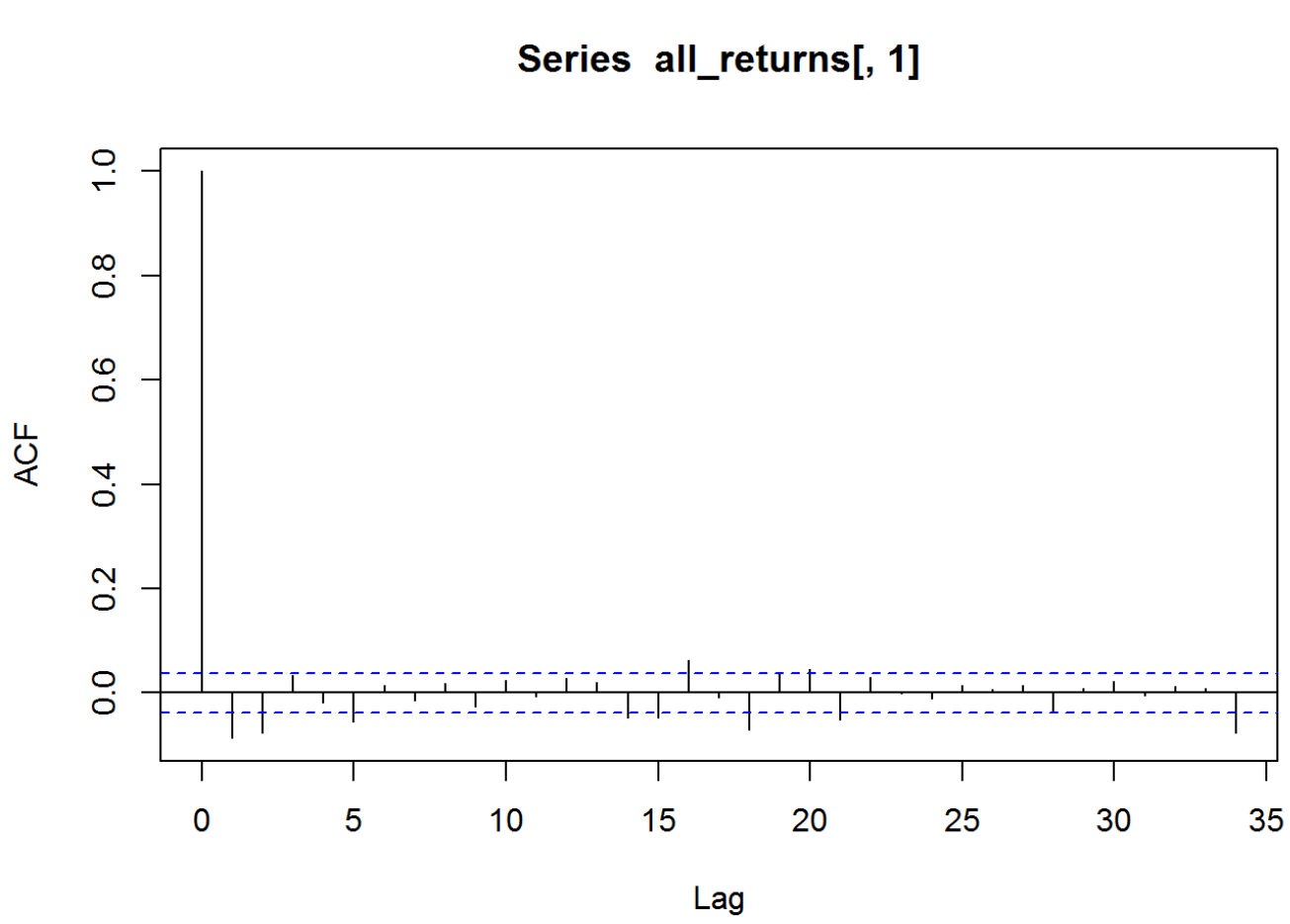
```
##                ClCl.SPYa    ClCl.TLTa     ClCl.LQDa    ClCl.EEMa
## 2007-01-03            NA           NA             NA           NA
## 2007-01-04  0.0021221123  0.006063328  0.0075152938 -0.013809353
## 2007-01-05 -0.0079763183 -0.004352668 -0.0006526807 -0.029238205
## 2007-01-08  0.0046250821  0.001793566 -0.0002798843  0.007257535
## 2007-01-09 -0.0008498831  0.000000000  0.0001866169 -0.022336235
## 2007-01-10  0.0033315799 -0.004475797 -0.0013063264 -0.002303160
##               ClCl.VNQa
## 2007-01-03           NA
## 2007-01-04  0.001296655
## 2007-01-05 -0.018518518
## 2007-01-08  0.001451392
## 2007-01-09  0.012648208
## 2007-01-10  0.012880523
```

```
all_returns = as.matrix(na.omit(all_returns))
# These returns can be viewed as draws from the joint distribution
pairs(all_returns)
```

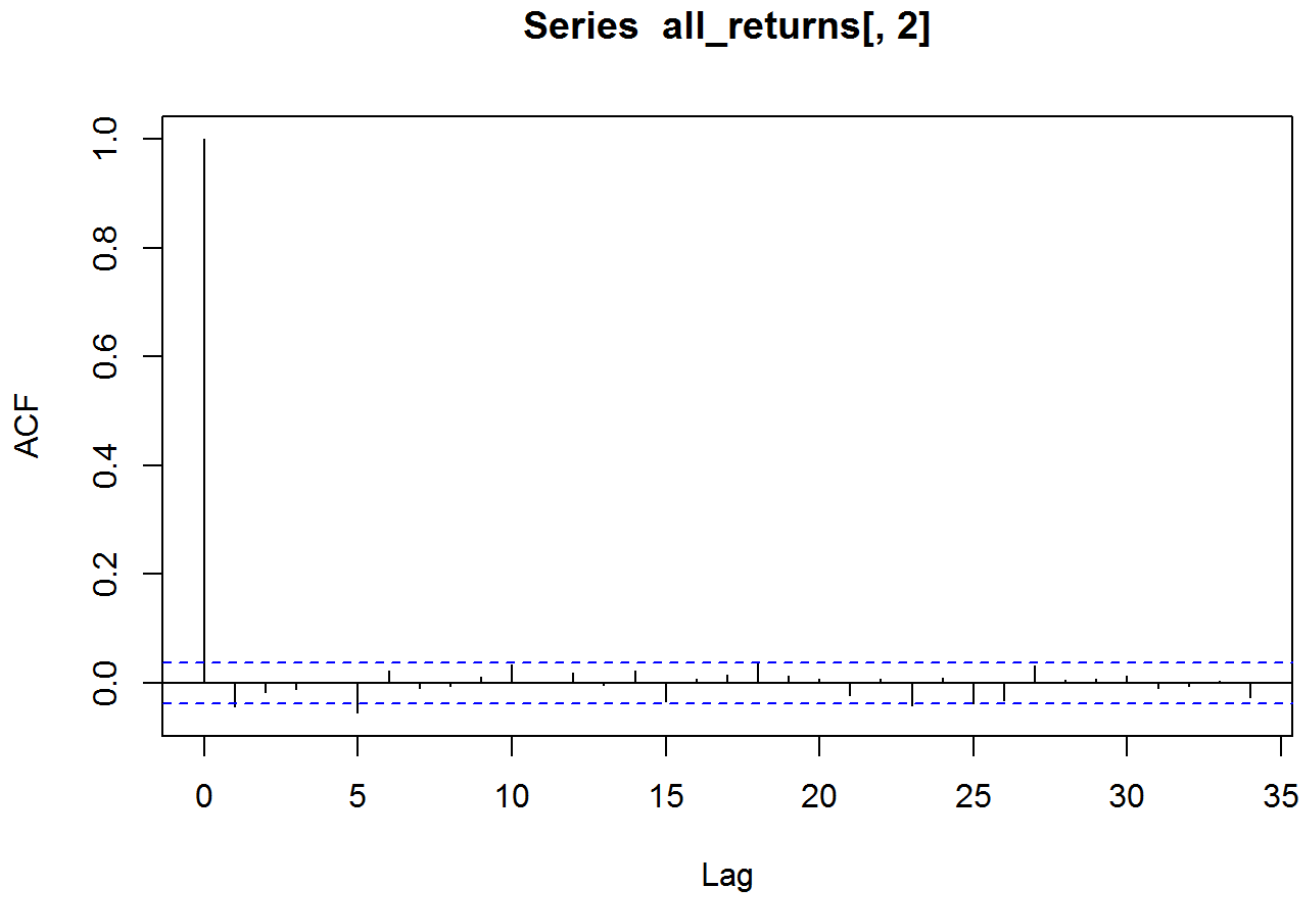


SPY and VNQ share a high positive correlation

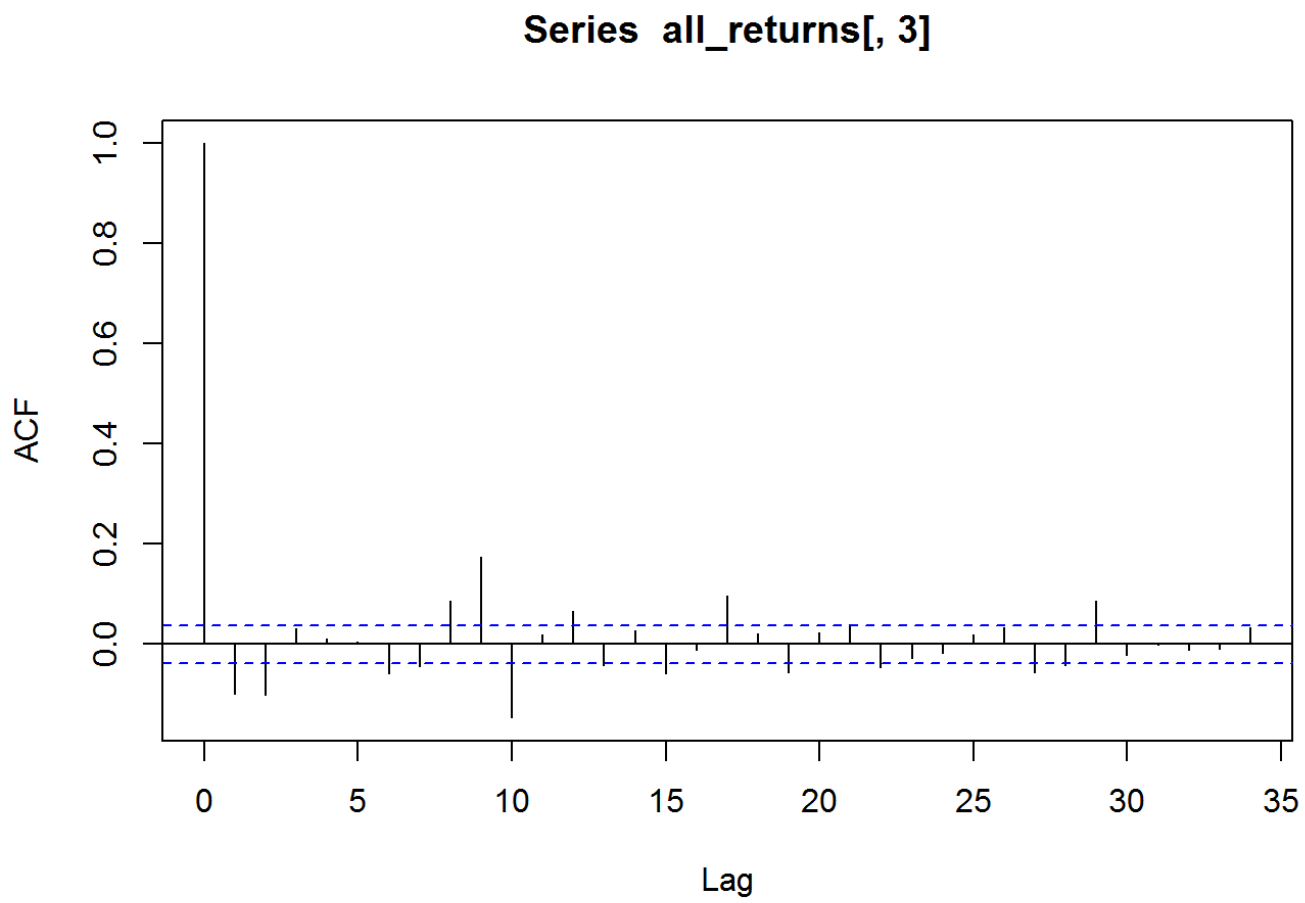
```
# An autocorrelation plot: nothing there
acf(all_returns[,1]) ### how functions perform as a function of time, any correlati
on
```



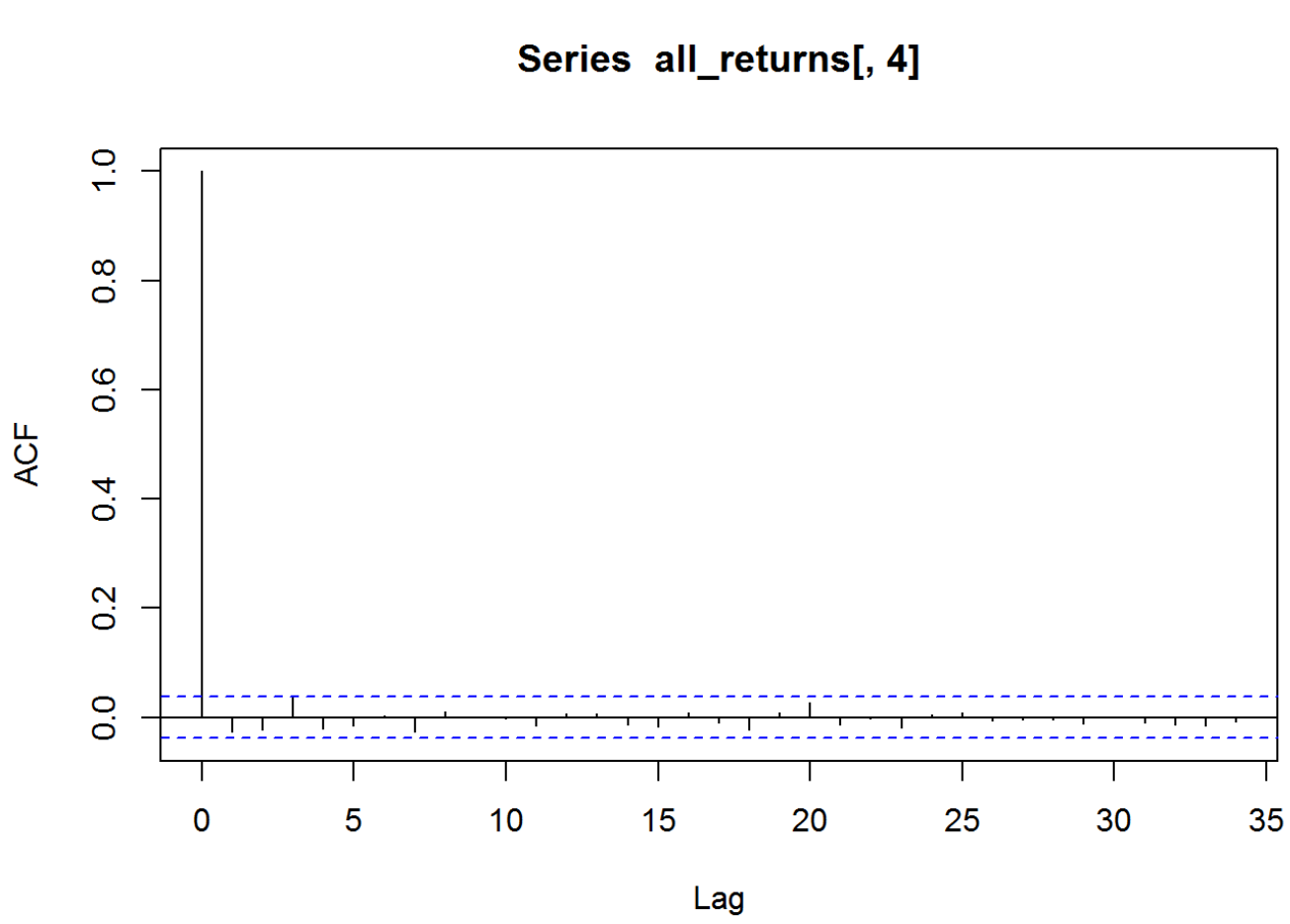

```
acf(all_returns[,2])
```



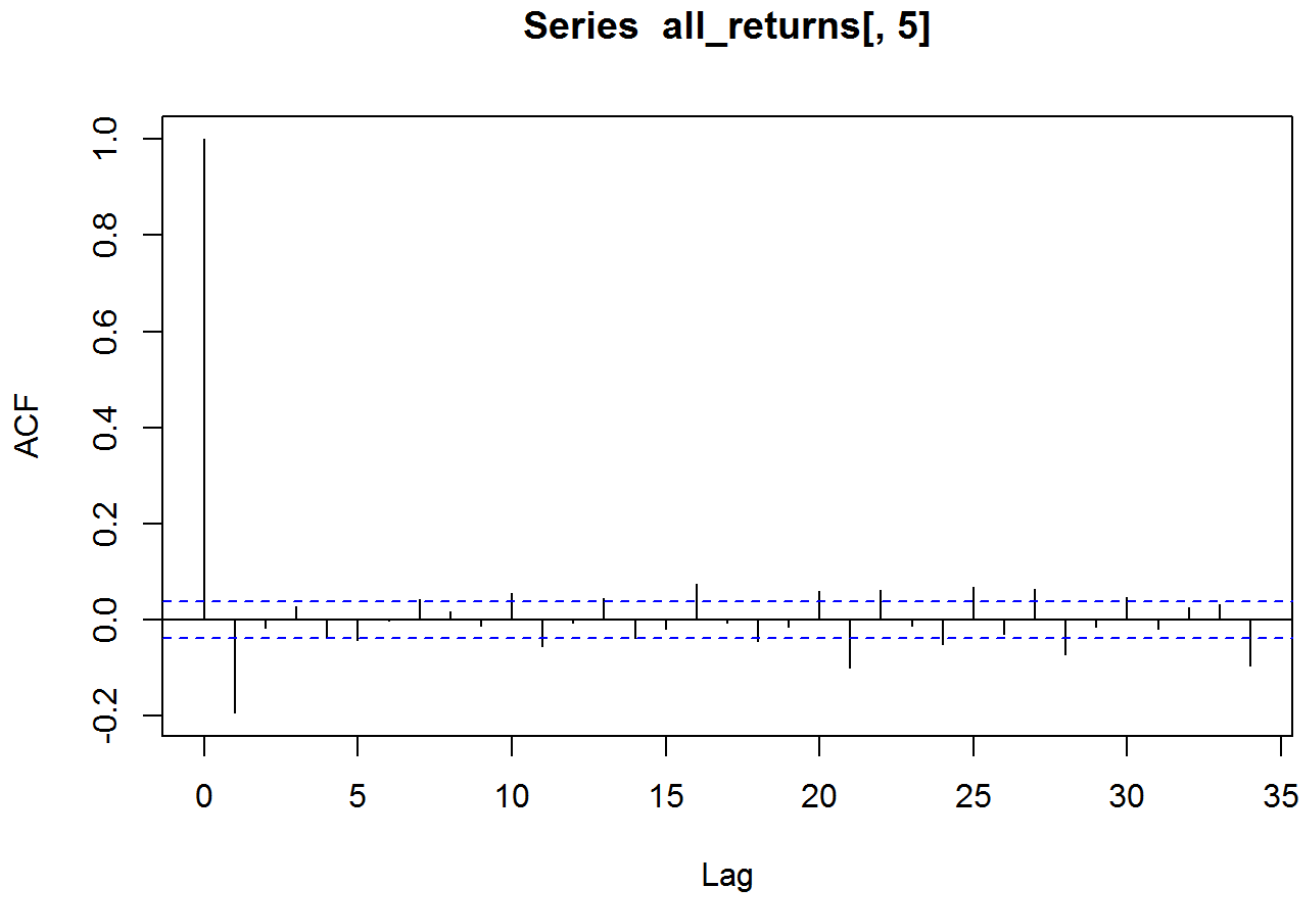
```
acf(all_returns[,3])
```




```
acf(all_returns[,4])
```




```
acf(all_returns[,5])
```



None of the assets are time dependent or are related to it'previous performance

```
# The sample correlation matrix
cor(all_returns) ### if any stocks are related to each other, give high or low retu
rns
```

```
##            ClCl.SPYa  ClCl.TLTa  ClCl.LQDa   ClCl.EEMa   ClCl.VNQa
## ClCl.SPYa  1.0000000 -0.4436811 0.10258006  0.40342388  0.77663305
## ClCl.TLTa -0.4436811  1.0000000 0.42142782 -0.16877436 -0.26527461
## ClCl.LQDa  0.1025801  0.4214278 1.00000000  0.08802841  0.06701479
## ClCl.EEMa  0.4034239 -0.1687744 0.08802841  1.00000000  0.29135938
## ClCl.VNQa  0.7766330 -0.2652746 0.06701479  0.29135938  1.00000000
```

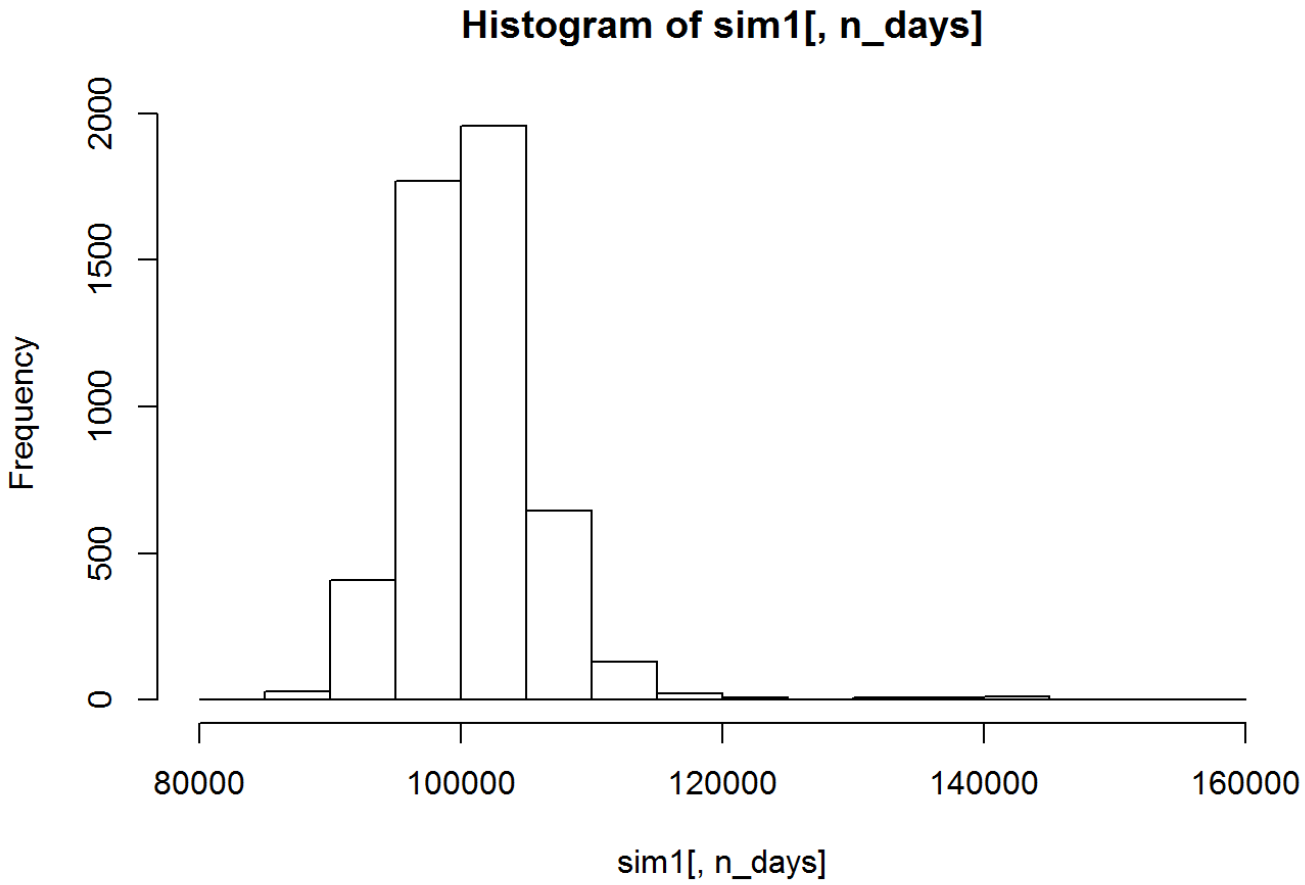
Keeping SPY and VNQ together for a portfolio increases votality as they move in same direction.
Such portfolio could result into either really huge profits or loss.
A safe portfolio should have variables with low correlation because even if one asset is losing, the other having low correlation might not move in the same direction. Hence, overall we might not face such huge loss or gains.
TLT shares low correlation with almost all other assets. Hence, it could be a good asset to include as a part of safe portfolio. Bonds(TNT, LQD) have low correlation with equity (SPY, EEM). Hence these together could be a part of safe portfolio. Also, real estate has low correlation with stock market which leads to low exposure to risk and high yields.

Even split

```
set.seed(100)
all_returns = cbind(ClCl(SPYa),ClCl(TLTa),ClCl(LQDa),ClCl(EEMa),ClCl(VNQa))
all_returns = as.matrix(na.omit(all_returns))
initial_wealth = 100000
sim1 = foreach(i=1:5000, .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 = mosaic::resample(all_returns, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
  }
  wealthtracker
}

head(sim1)
```

```
##               [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## result.1 100598.84 100973.03 100948.95  99304.34  98749.47  98814.41
## result.2 100606.24 101244.10 101051.76 101076.65 101176.06  98604.08
## result.3 101823.09 101358.32 101646.38 101892.42 101270.66 101148.40
## result.4 102781.32 102636.67 102543.80 101861.60 100093.38 100755.97
## result.5  99485.53  99205.78  99061.98  99056.53  99996.16  99533.85
## result.6 100262.04 100480.88  99163.95  99466.84  99380.81  99915.42
##               [,7]      [,8]      [,9]     [,10]     [,11]     [,12]
## result.1  99086.66  99506.09  99718.60  95401.50  96171.52  95688.18
## result.2  98654.03  99320.86  99395.54  99777.23 100152.11 100364.48
## result.3 101536.83 101180.99 104372.40 103970.40 105865.72 107375.77
## result.4 100159.35  98587.05  98282.06  98147.57  98455.52  97220.55
## result.5  99602.89 100299.04 100557.52 100629.67 100793.20 102547.08
## result.6 100075.11 100182.54 100258.41 100016.36  99455.90  99084.22
##              [,13]     [,14]     [,15]     [,16]     [,17]     [,18]
## result.1  95919.15  96148.61  95976.52  96613.57  96589.15  97239.45
## result.2 100539.92 100722.93 100557.95 101073.72  94292.64  93930.71
## result.3 107906.77 108777.93 108865.92 110249.19 110629.81 111012.56
## result.4  96787.52  96886.63  96983.28  95989.59  96471.12  96485.54
## result.5 101485.45 100122.17  99605.51  99822.06 100260.84 100104.54
## result.6  98083.96  98670.13  98529.66  99474.93 100344.18 100862.78
##              [,19]     [,20]
## result.1  97657.44  97891.80
## result.2  93748.85  92766.42
## result.3 111164.72 112371.61
## result.4  96513.06  97318.98
## result.5 100068.37 100448.16
## result.6  99043.78  98927.78
```

```
hist(sim1[,n_days], 25)
```



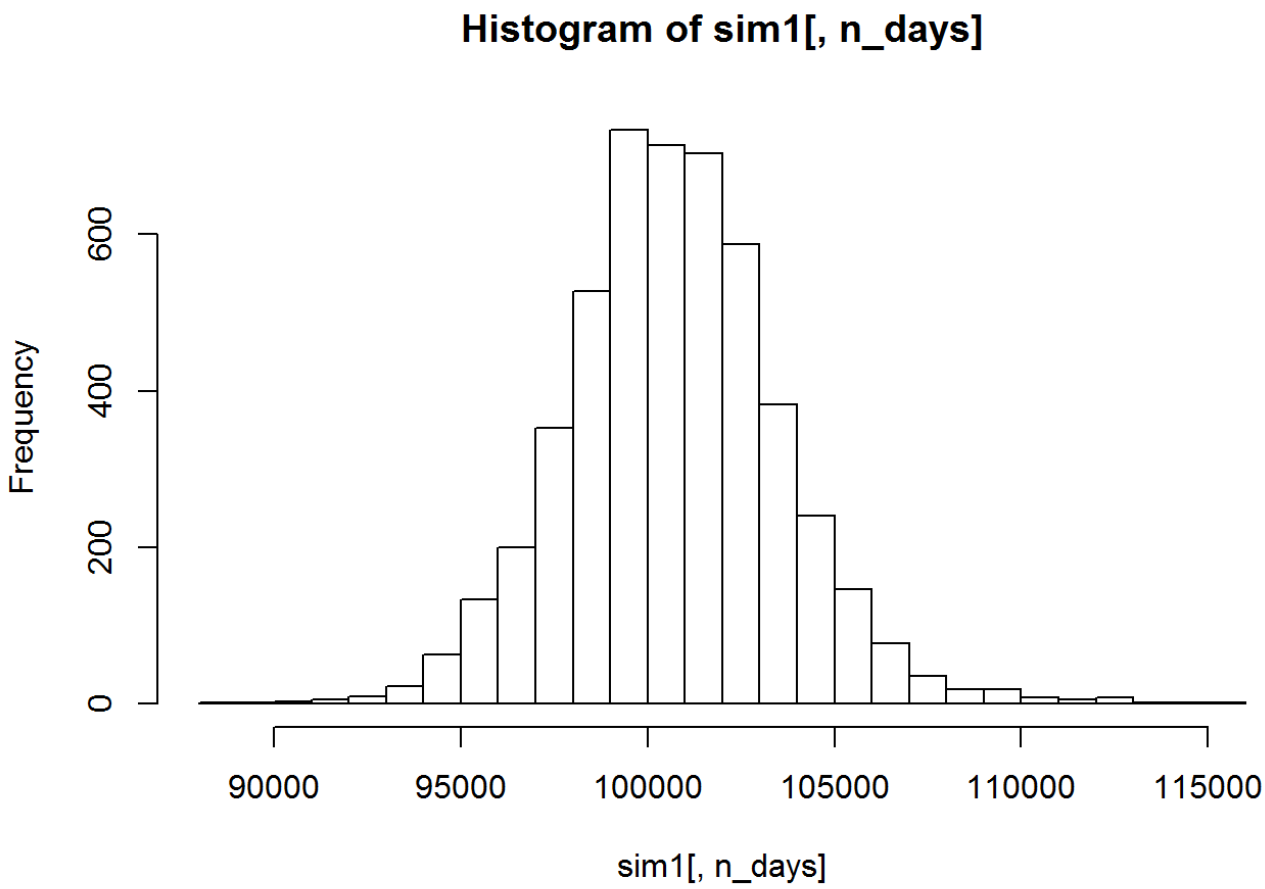

```
even_sd<-sd(sim1[,n_days])
# Profit/loss
even_mean<-mean(sim1[,n_days])
#hist(sim1[,n_days]- initial_wealth, breaks=30)

# Calculate 5% value at risk
even_var<-quantile(sim1[,n_days], 0.05) - initial_wealth
####money being lost 5% of the time
```

Safe split

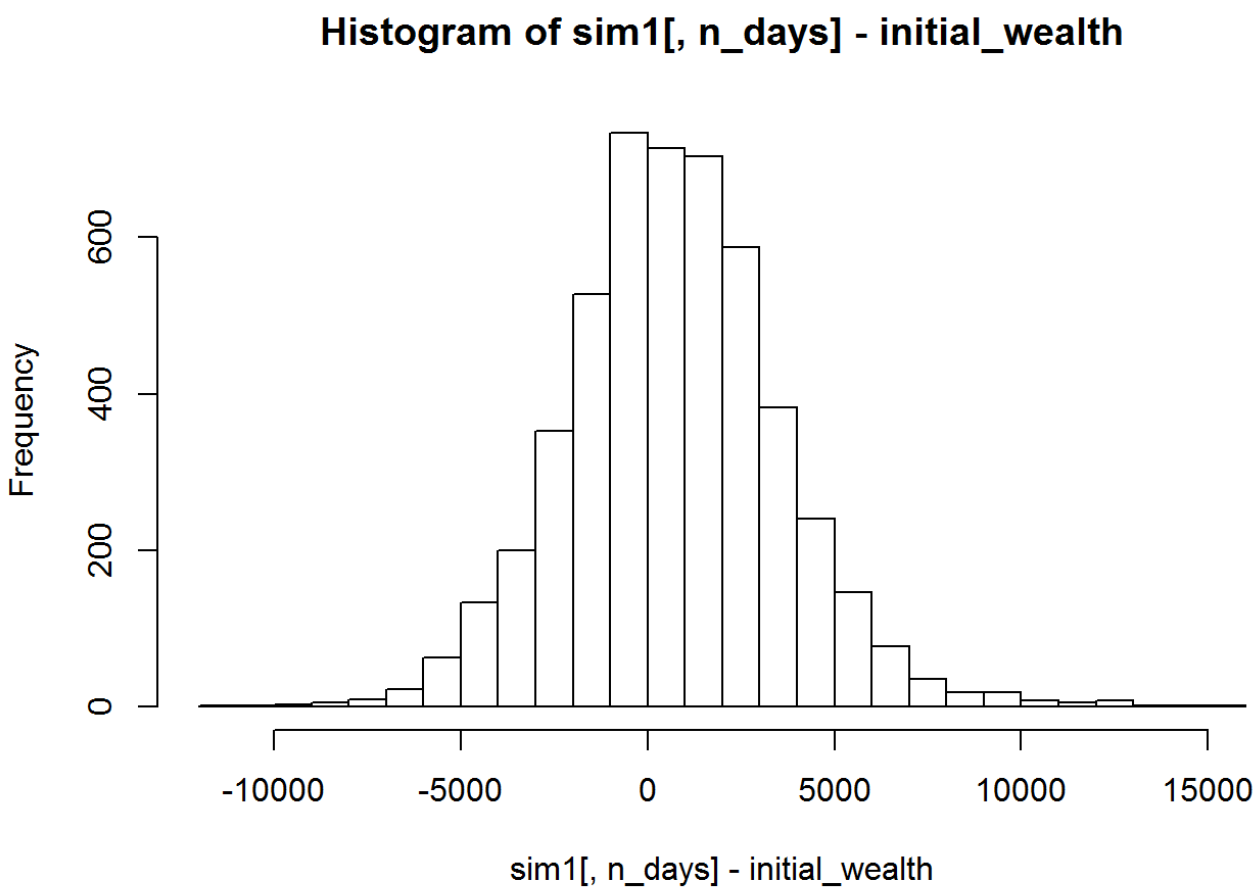
```
set.seed(100)
all_returns_safe = cbind(ClCl(LQDa),ClCl(TLTa),ClCl(SPYa))
all_returns_safe = as.matrix(na.omit(all_returns_safe))
initial_wealth = 100000
sim1 = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0.25,0.25,0.5)
  holdings = weights * total_wealth
  n_days = 20
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return = mosaic::resample(all_returns_safe, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
  }
  wealthtracker
}

head(sim1)
```

```
##                [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## result.1 100404.16 100550.64 100560.56  99089.42  98980.58  99062.31
## result.2 100488.34 100897.14 100325.77 100661.32 100838.06  98845.44
## result.3 101298.21 101140.80 101018.31 101356.19 101292.76 101156.57
## result.4 101780.52 101807.26 101749.03 101523.95 100873.09 101500.42
## result.5  99926.16  99706.46  99781.77  99696.56 100173.19  99664.76
## result.6 100119.63 100354.49  99308.23  99209.46  99283.13  99612.53
##                [,7]      [,8]      [,9]     [,10]     [,11]     [,12]
## result.1  99322.29  99810.86  99922.43  97495.48  98078.66  97821.87
## result.2  98902.56  99571.33  99526.58  99625.33  99826.58  99908.46
## result.3 101488.65 101479.15 104113.22 104279.68 105620.52 106364.35
## result.4 101360.14 100851.00 101176.14 101024.76 101341.01 100285.87
## result.5  99067.69  99390.19  99600.62  98999.86  99311.34 100732.79
## result.6 100003.69 100139.61 100227.48 100078.57  99727.74  99261.46
##               [,13]     [,14]     [,15]     [,16]     [,17]     [,18]
## result.1  97732.45  97891.51  97793.77  98268.65  98229.15  98845.16
## result.2 100218.06 100373.31 100363.12 100741.94  97419.45  96927.38
## result.3 106370.52 106611.22 106996.24 108212.80 108575.50 109211.82
## result.4  99970.22 100082.25 100335.46  99651.95  99633.25  99566.97
## result.5 100233.78  99583.01  99220.33  99450.75  99958.32  99611.11
## result.6  98852.42  99216.75  99044.88  99513.28 100125.56 100391.80
##               [,19]     [,20]
## result.1  99150.26  99472.41
## result.2  97084.69  96174.49
## result.3 109281.51 110569.67
## result.4  99453.61  99688.81
## result.5  99739.97  99944.27
## result.6  99219.46  99123.78
```

```
hist(sim1[,n_days], 25)
```




```
safe_sd<-sd(sim1[,n_days])
# Profit/loss
safe_mean<-mean(sim1[,n_days])
hist(sim1[,n_days]- initial_wealth, breaks=30)
```



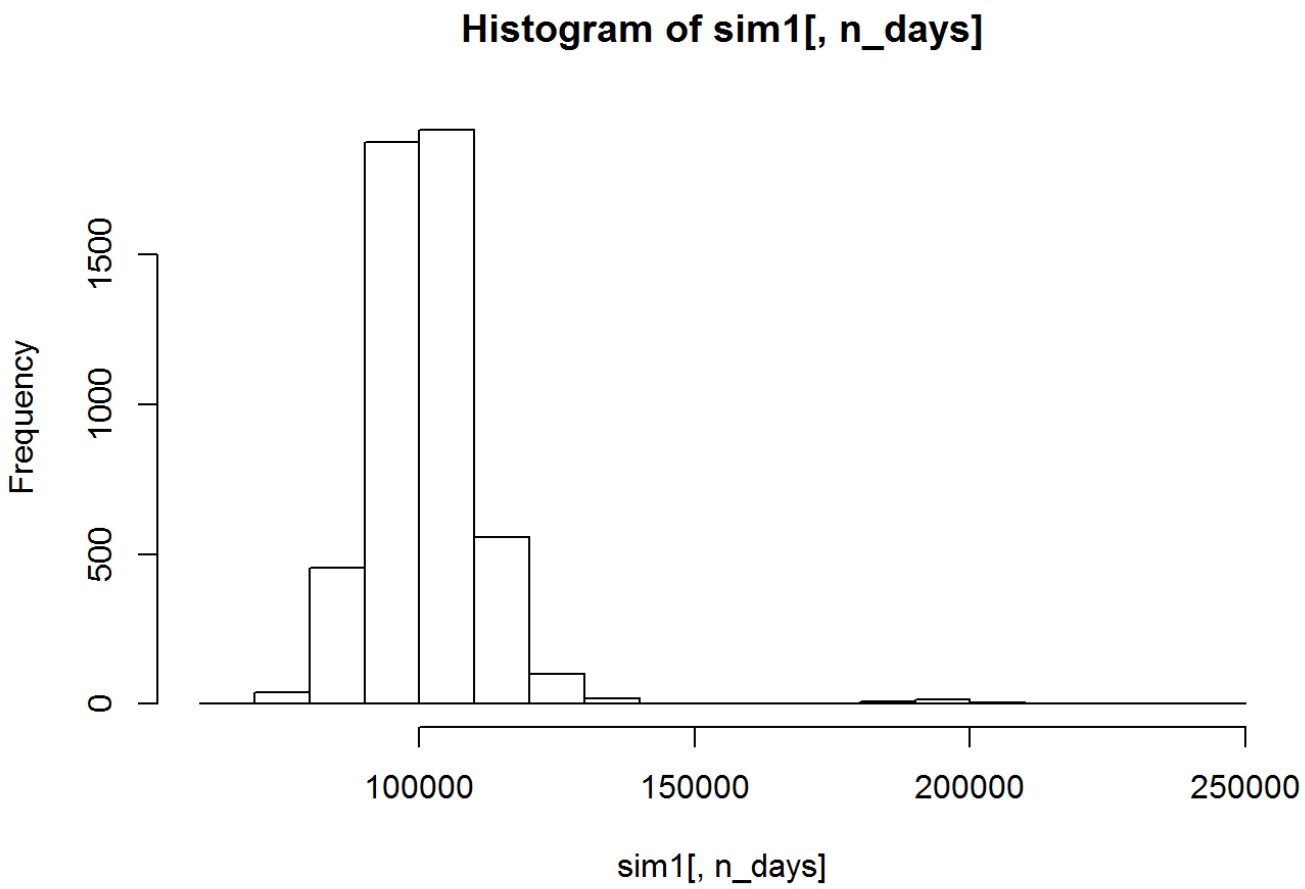

```
# Calculate 5% value at risk
safe_var<-quantile(sim1[,n_days], 0.05) - initial_wealth
```

aggresive portfolio

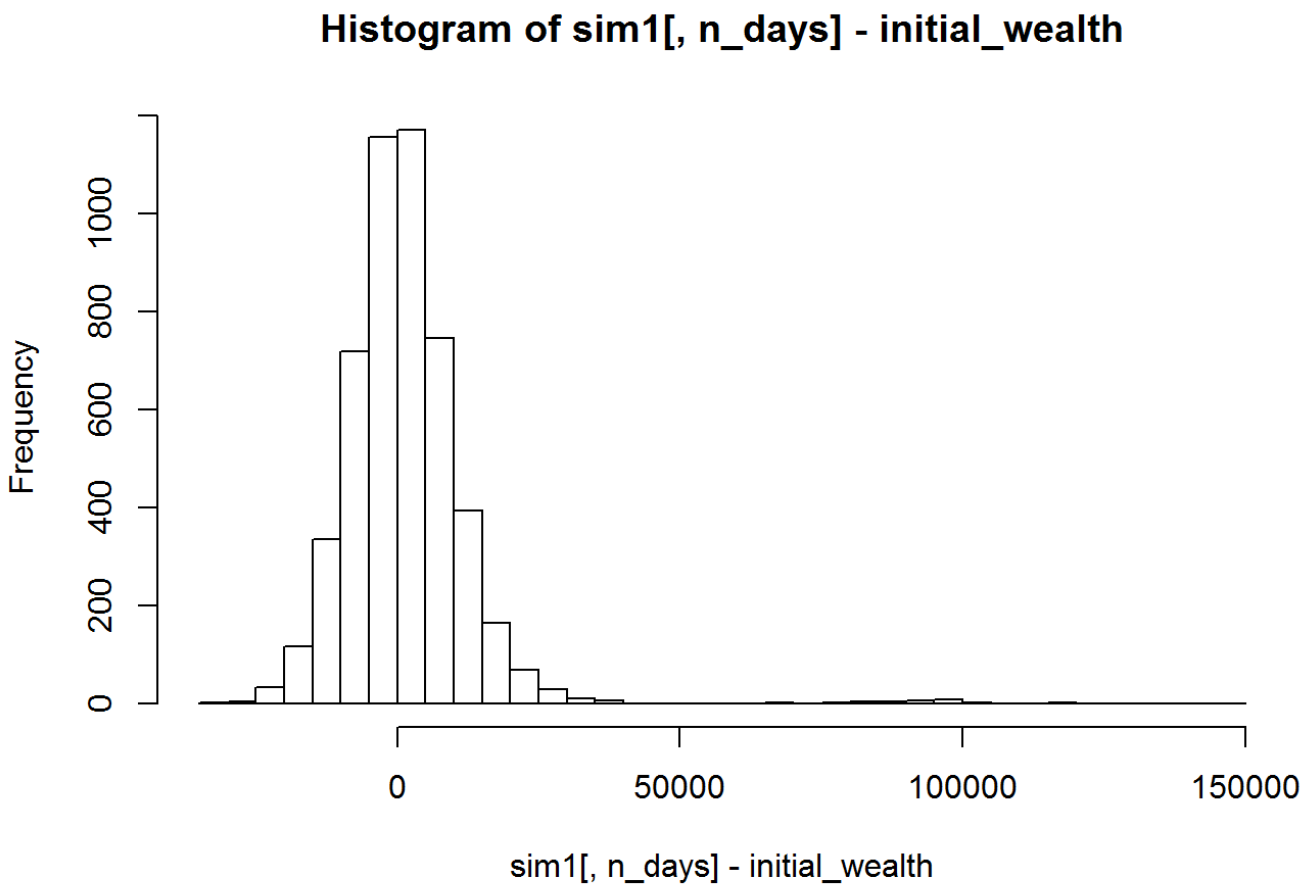
```
set.seed(100)
all_returns_agg = cbind(ClCl(VNQa),ClCl(EEMa))
all_returns_agg = as.matrix(na.omit(all_returns_agg))
initial_wealth = 100000
sim1 = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0.5,0.5)
  holdings = weights * total_wealth
  n_days = 20
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return = mosaic::resample(all_returns_agg, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
  }
  wealthtracker
}

head(sim1)
```

```
##               [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## result.1  99286.12 101979.74 102041.78  99386.27 102959.77 101935.13
## result.2 101202.43 101193.18 100490.85  99224.30  99759.83  95620.15
## result.3 100864.18 101457.17 102634.92 101401.49 100922.33  99979.19
## result.4  99438.81  99845.67  99548.23  98946.51  94496.57  92412.79
## result.5 100684.45 101102.00 100810.24 100998.85 102850.72 101754.57
## result.6  98059.28  98550.49  96839.10  97776.53  97886.65  96900.72
##               [,7]      [,8]      [,9]     [,10]     [,11]     [,12]
## result.1 101825.65 101708.15 101953.04  93029.33  91876.22  90663.62
## result.2  95807.78  95949.02  95162.08  96096.05  96856.80  97423.53
## result.3 100498.82 101403.61 105247.73 105490.05 109250.06 111914.04
## result.4  89803.89  85558.71  84150.65  85090.12  86187.43  87792.03
## result.5 102870.70 102926.81 102433.59 101584.94 101465.83 103983.14
## result.6  97056.70  96586.65  96576.58  98087.21  97940.57  98479.93
##              [,13]     [,14]     [,15]     [,16]     [,17]     [,18]
## result.1  91596.70  91929.88  92772.50  93394.13  93455.73  94107.01
## result.2  97908.18  98013.73  98074.99  96589.07  82536.57  84494.16
## result.3 113239.77 114425.63 111857.77 113428.68 114158.76 115462.81
## result.4  87612.00  87772.09  89143.55  90093.87  88709.74  87239.48
## result.5 105947.58 108856.89 107954.16 108493.37 108836.10 109298.82
## result.6  96238.31  95431.35  95455.68  97214.80  98429.33  97960.35
##              [,19]     [,20]
## result.1  93711.34  93974.92
## result.2  83850.08  83649.15
## result.3 115908.65 117352.74
## result.4  85254.36  87368.32
## result.5 109030.97 108341.29
## result.6  99843.00  98206.78
```

```
hist(sim1[,n_days], 25)
```




```
aggressive_sd<-sd(sim1[,n_days])
# Profit/loss
aggressive_mean<-mean(sim1[,n_days])
hist(sim1[,n_days]- initial_wealth, breaks=30)
```




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

```
portfolio<-data.frame('Portfolio_type' = c('Even','Safe','Aggresive'),'Mean' = c(e
ven_mean,safe_mean,aggressive_mean),'Variance' = c(even_sd,safe_sd,aggressive_sd),
'VAR'=c(even_var,safe_var,agg_var))
portfolio
```

```
##   Portfolio_type     Mean  Variance        VAR
## 1           Even 101046.3  5485.654  -6356.321
## 2           Safe 100701.5  2884.230  -3941.895
## 3      Aggresive 101507.7 12172.590 -13117.104
```

Market Segmentation

```
library(plotly)
library(ggplot2)
library(LICORS)
```

```
## Warning: package 'LICORS' was built under R version 3.3.3
```

```
library(foreach)
library(mosaic)
social_mktg<-read.csv('social_marketing.csv',header = TRUE,stringsAsFactors = FALSE
,row.names=1)
names(social_mktg)
```

```
##  [1] "chatter"          "current_events"   "travel"
##  [4] "photo_sharing"    "uncategorized"    "tv_film"
##  [7] "sports_fandom"    "politics"         "food"
## [10] "family"           "home_and_garden"  "music"
## [13] "news"             "online_gaming"    "shopping"
## [16] "health_nutrition" "college_uni"      "sports_playing"
## [19] "cooking"          "eco"              "computers"
## [22] "business"         "outdoors"         "crafts"
## [25] "automotive"       "art"              "religion"
## [28] "beauty"           "parenting"        "dating"
## [31] "school"           "personal_fitness" "fashion"
## [34] "small_business"   "spam"             "adult"
```

Remove columns for categories which won't be targeted

```
social_mktg<-social_mktg[,-c(1,5,35,36)]
```

```
x<-colSums(social_mktg)
```

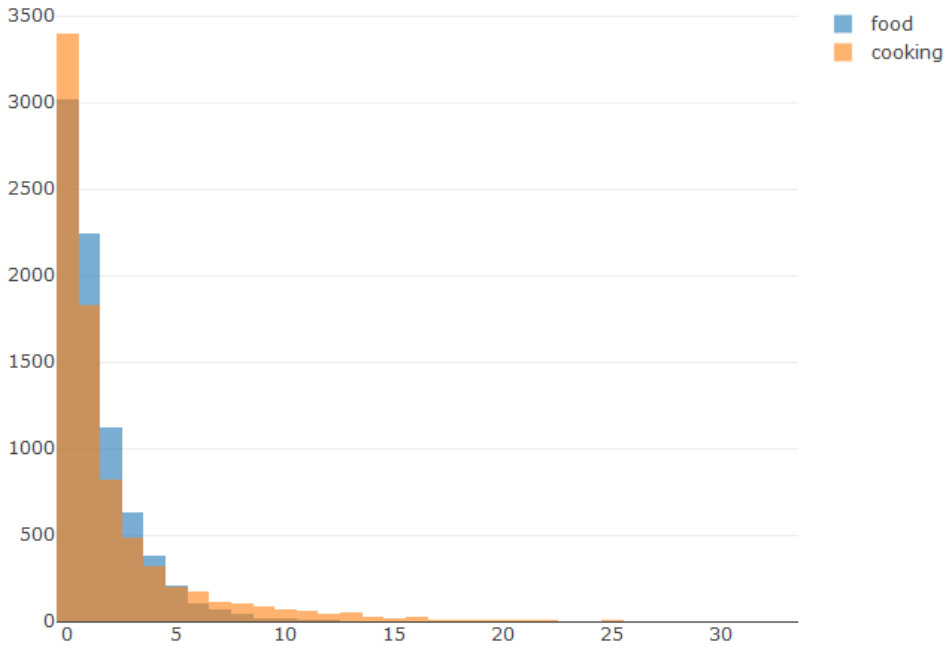
Counting the total number of tweets about each particular category

Photo sharing was tweeted on the most number of times while small_business was the least

```
correlation<-cor(social_mktg,method = 'pearson')
```

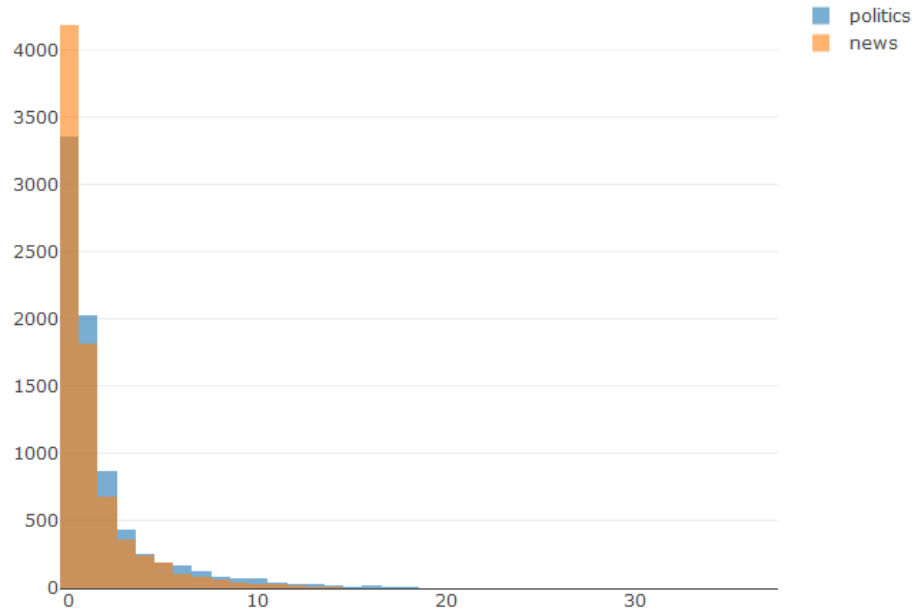
To check if any related variables are tweeted about together and group them

```
p1<- plot_ly(alpha = 0.6) %>%
  add_histogram(x = social_mktg$food,name = 'food') %>%
  add_histogram(x = social_mktg$cooking, name = 'cooking') %>%
  layout(barmode = "overlay")
```



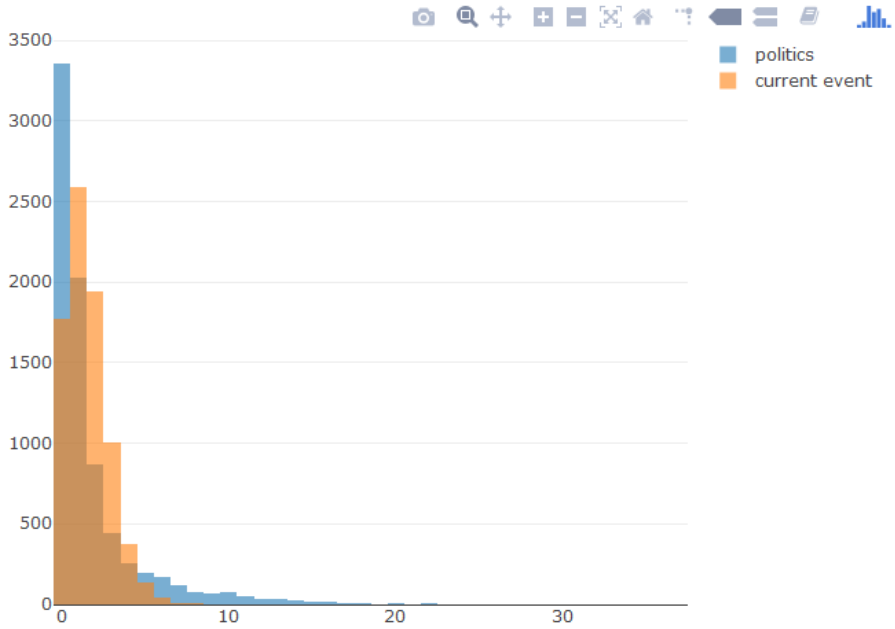
Very less correlation between cooking and food(correlation=0.06768445), as seen from the plot too, they are not tweeted together

```
p2<- plot_ly(alpha = 0.6) %>%
  add_histogram(x = social_mktg$politics, name = 'politics') %>%
  add_histogram(x = social_mktg$news, name = 'news') %>%
  layout(barmode = "overlay")
```



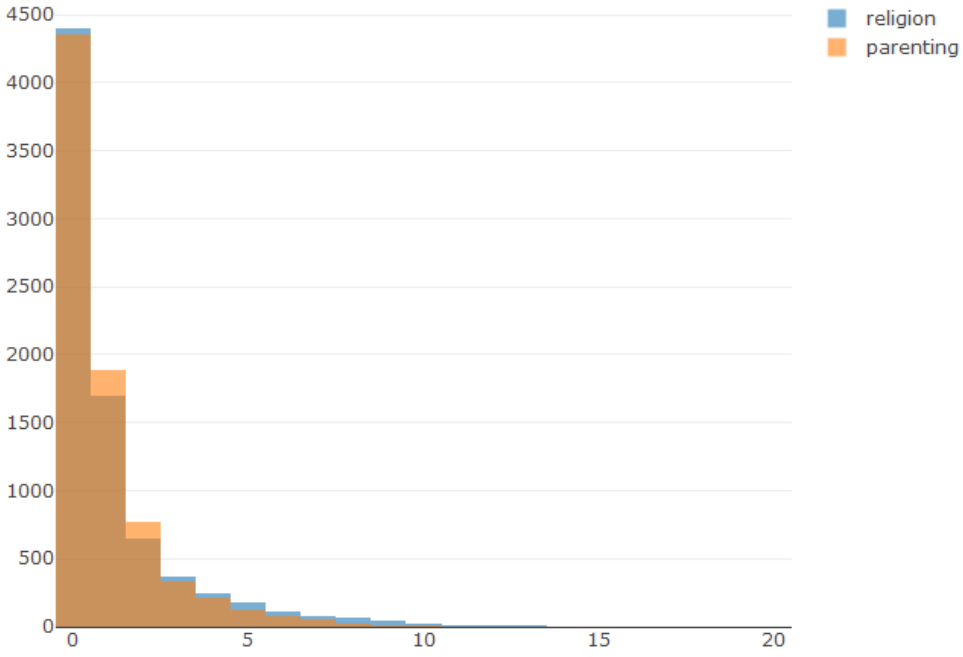
Politics and news have a very high correlation(0.5618422) as seen from the plot

```
p3<- plot_ly(alpha = 0.6) %>%
  add_histogram(x = social_mktg$politics, name = 'politics') %>%
  add_histogram(x = social_mktg$current_events, name = 'current event') %>%
  layout(barmode = "overlay")
```



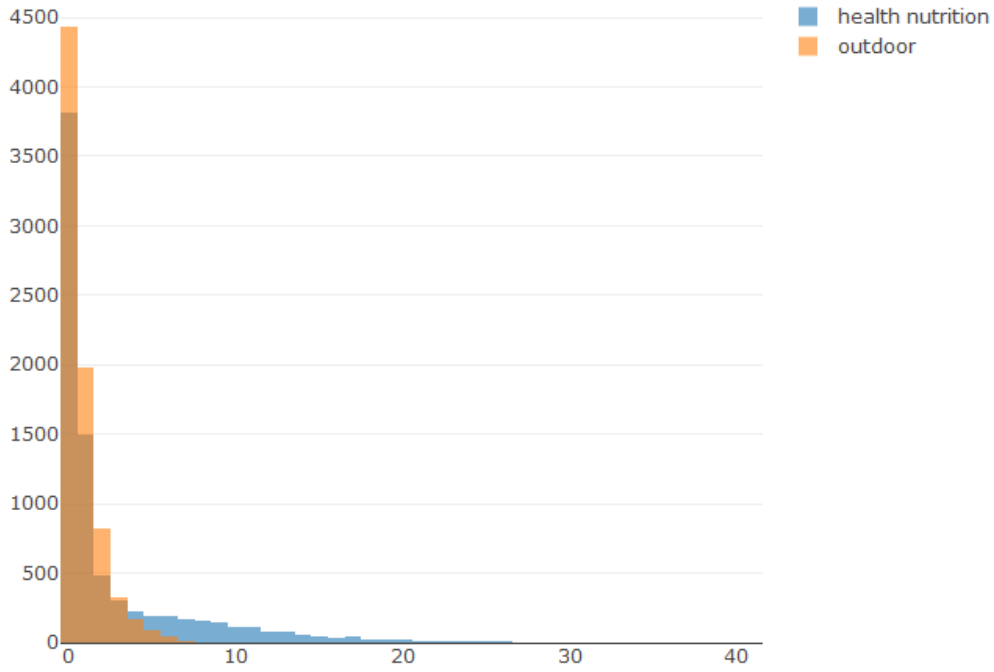
Very less correlation between politics and current events(correlation=0.06828273), as seen from the plot too, they are not tweeted together

```
p4<- plot_ly(alpha = 0.6) %>%
  add_histogram(x = social_mktg$religion, name = 'religion') %>%
  add_histogram(x = social_mktg$parenting, name = 'parenting') %>%
  layout(barmode = "overlay")
```



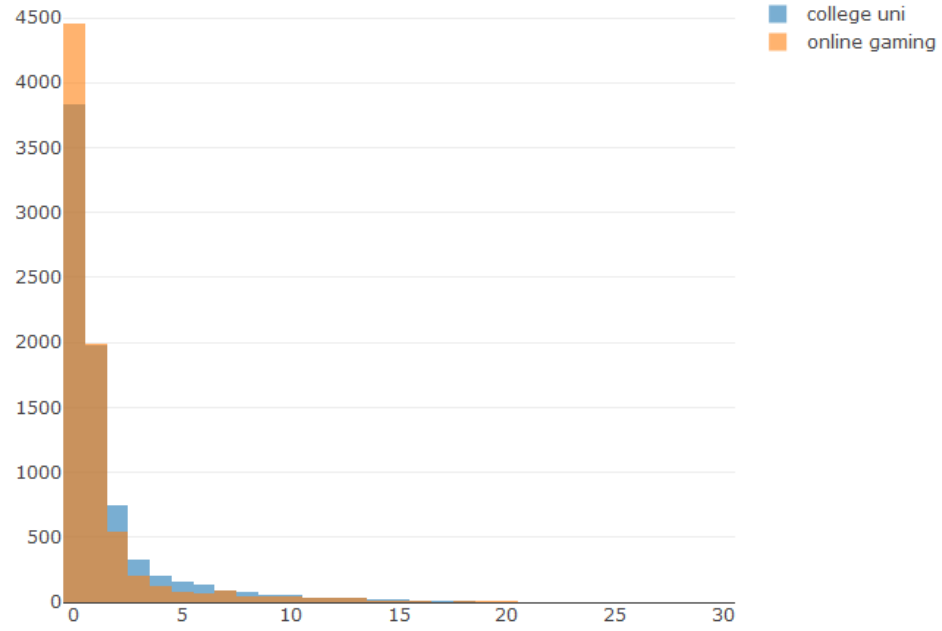
Religion and parenting have a very high correlation(0.6555973) as seen from the plot

```
p5<- plot_ly(alpha = 0.6) %>%
  add_histogram(x = social_mktg$health_nutrition, name = 'health nutrition') %>%
  add_histogram(x = social_mktg$outdoors, name = 'outdoor') %>%
  layout(barmode = "overlay")
```



####Health nutrition and outdoors have a very high correlation(0.6082254) as seen from the plot

```
p6<- plot_ly(alpha = 0.6) %>%
  add_histogram(x = social_mktg$college_uni, name = 'college uni') %>%
  add_histogram(x = social_mktg$online_gaming, name = 'online gaming') %>%
  layout(barmode = "overlay")
```



College_uni and online gaming have a very high correlation(0.7728393) as seen from the plot

```
attach(social_mktg)
social_mktgs= scale(social_mktg, center=TRUE, scale=TRUE)
```

Extract the centers and scales from the rescaled data (which are named attributes)

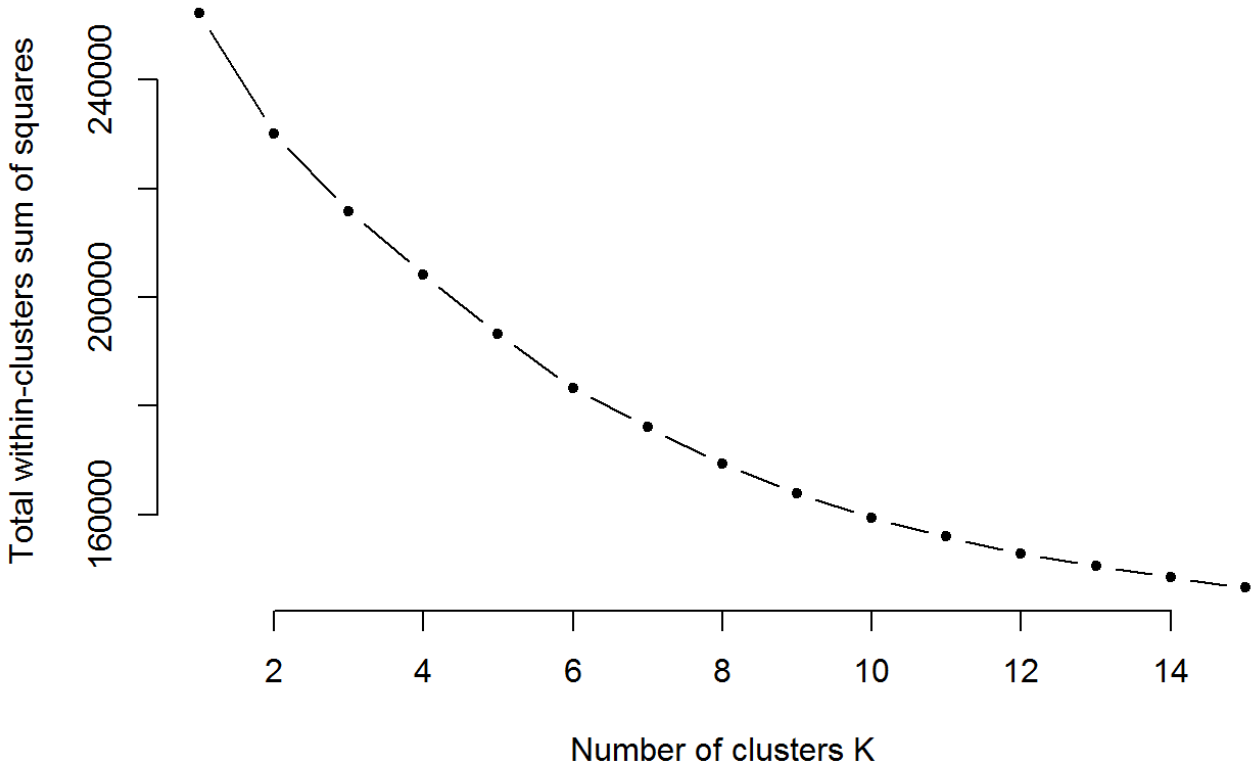
```
mu = attr(social_mktgs,"scaled:center")
sigma = attr(social_mktgs,"scaled:scale")
```

Compute and plot wss for k = 2 to k = 15 through elbow method to find optimal K value

```
k.max <- 15
data <- social_mktgs
sumofsquares <- sapply(1:k.max,
              function(k){kmeans(data, k, nstart=30,iter.max = 15 )$tot.withinss})
sumofsquares
```

```
##  [1] 252192.0 230121.4 215873.7 204176.7 193297.4 183176.1 176078.5
##  [8] 169350.0 163954.3 159314.6 155944.8 152723.3 150467.0 148405.4
## [15] 146550.2
```

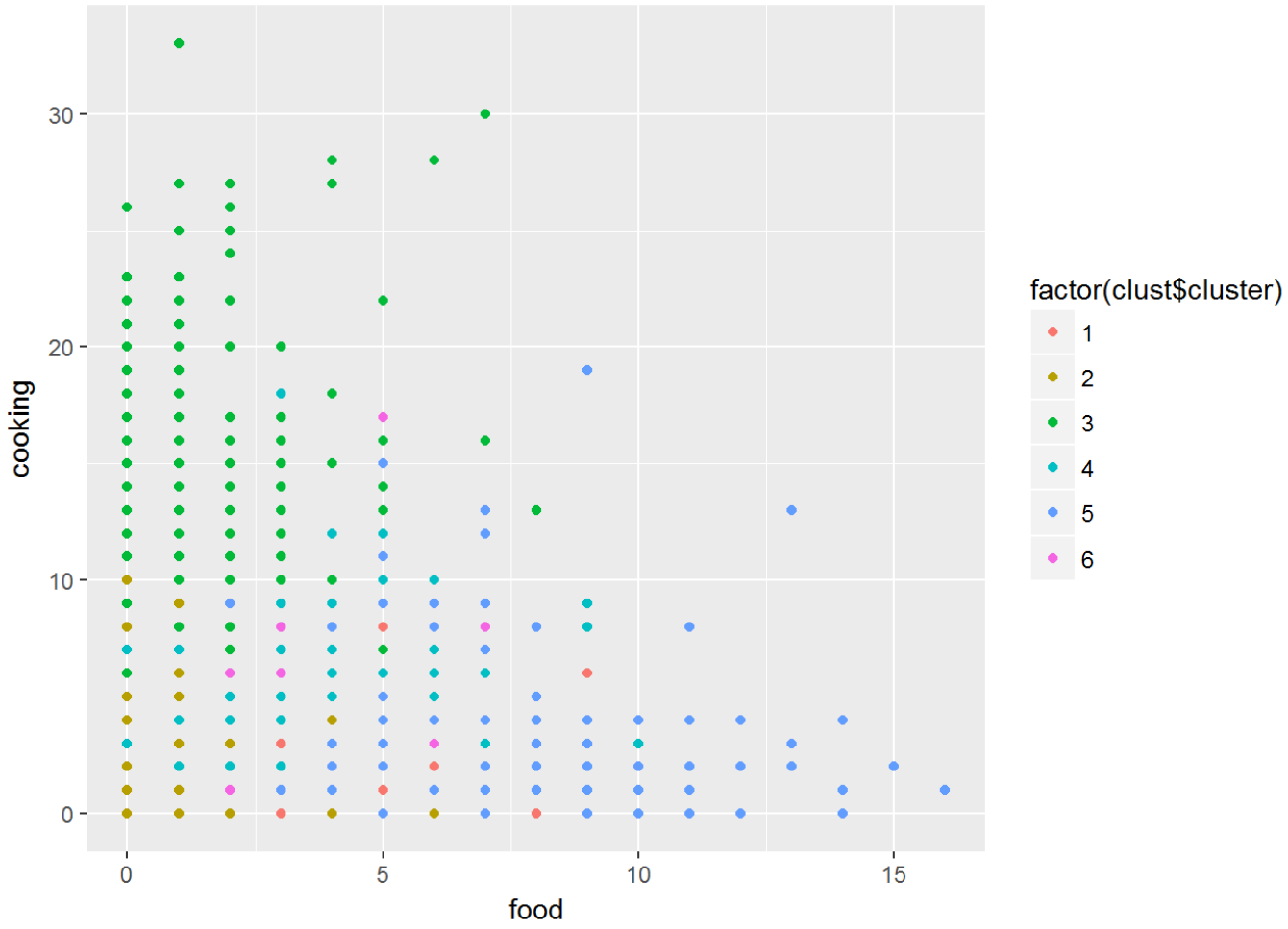
```
plot(1:k.max, sumofsquares,
     type="b", pch = 20, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```



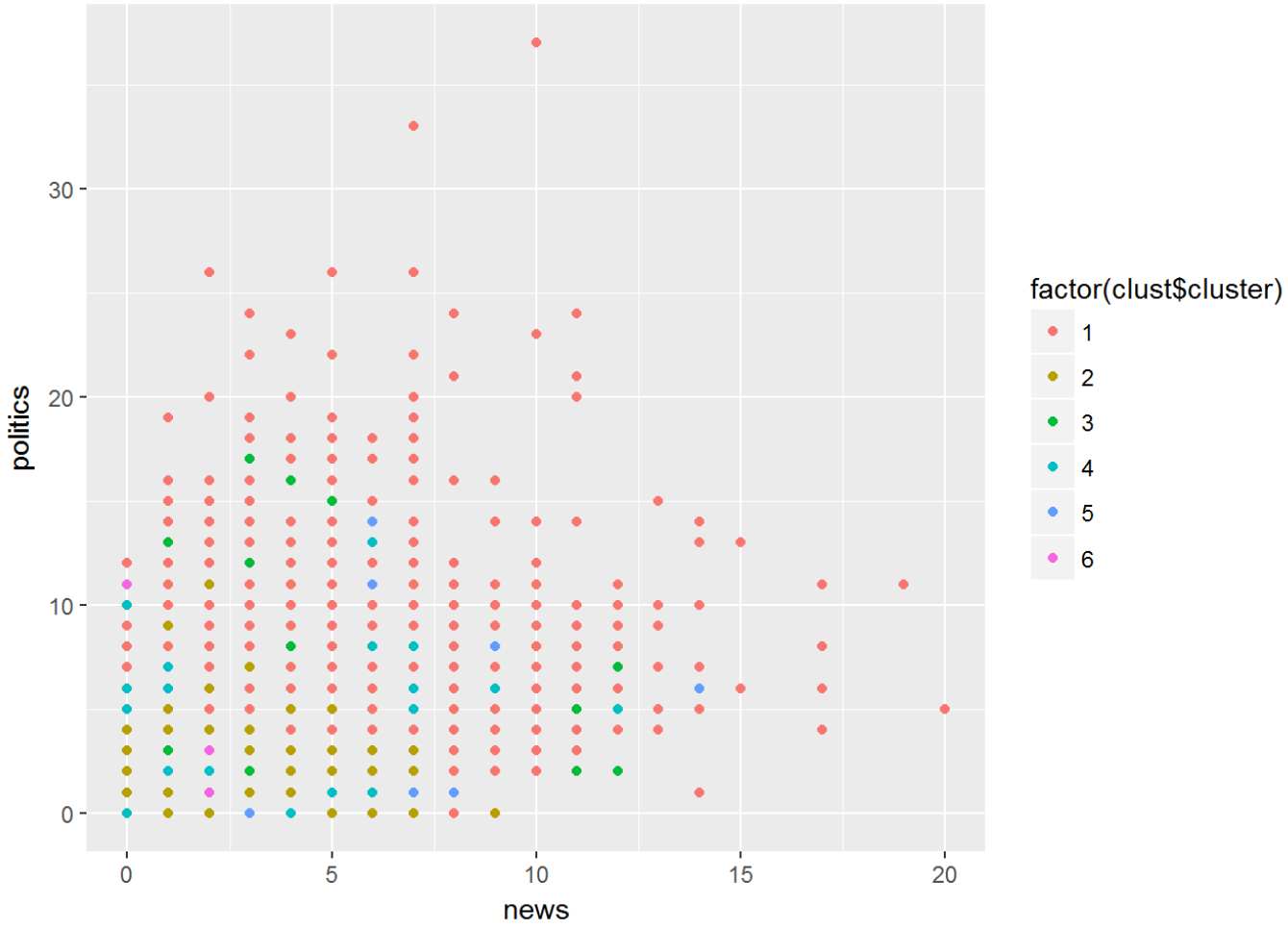
From the plot it is visible that the K value of 6 is the best choice for the given dataset

```
set.seed(12345)
clust = kmeanspp(social_mktgs, k=6, nstart=25)
```

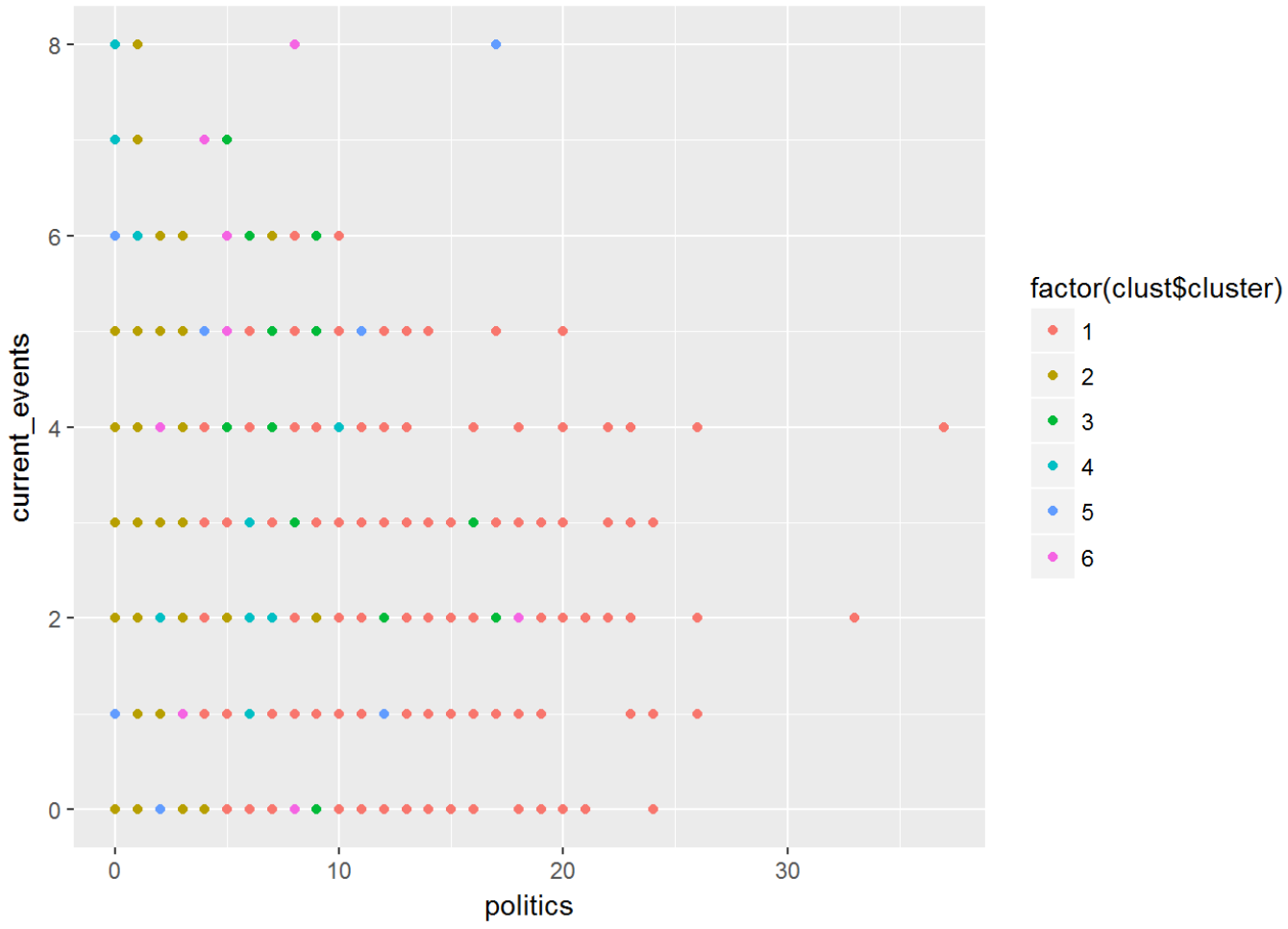
```
qplot(food, cooking, data=social_mktg, color=factor(clust$cluster))
```



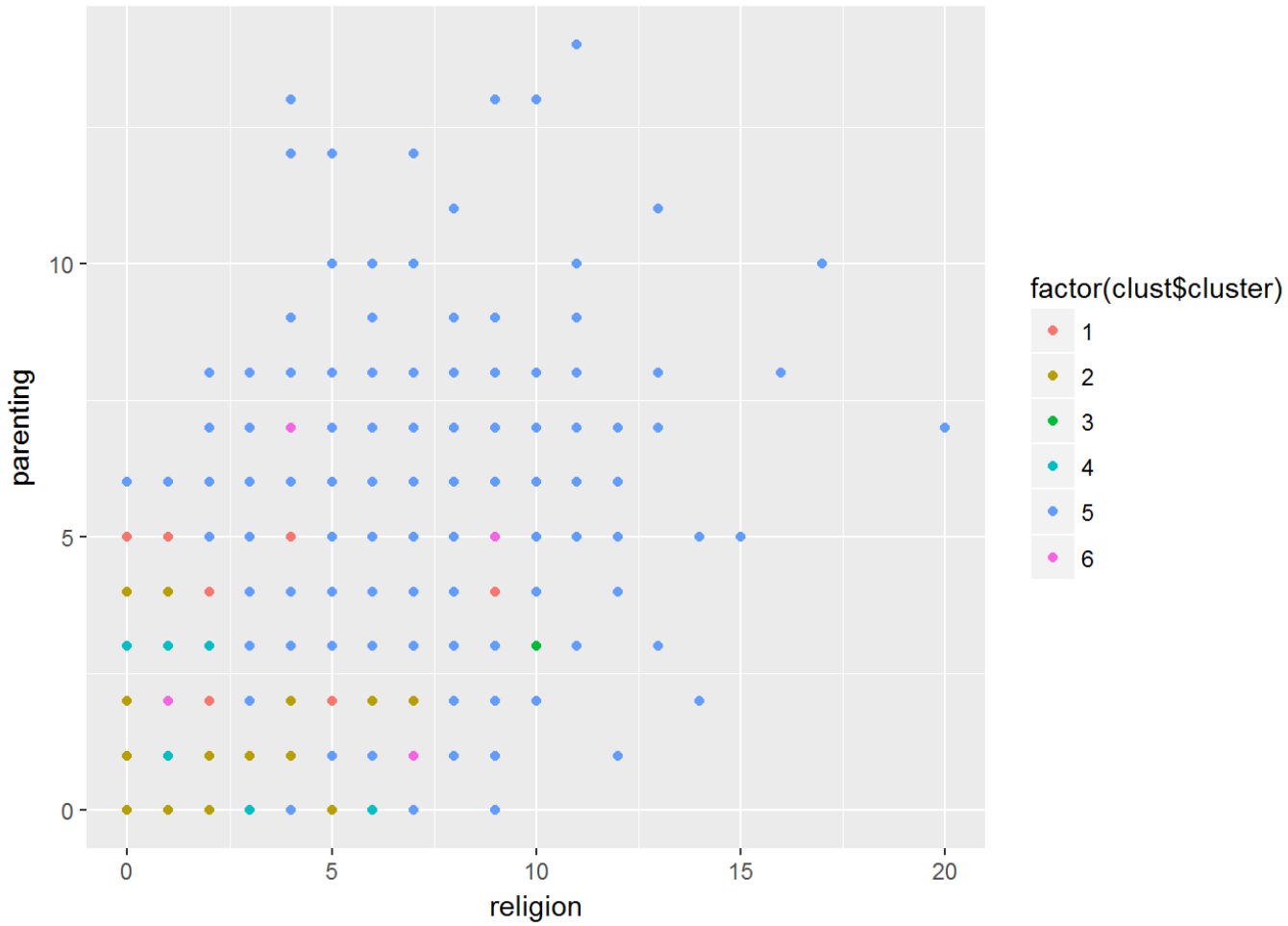

```
qplot(news, politics, data=social_mktg, color=factor(clust$cluster))
```



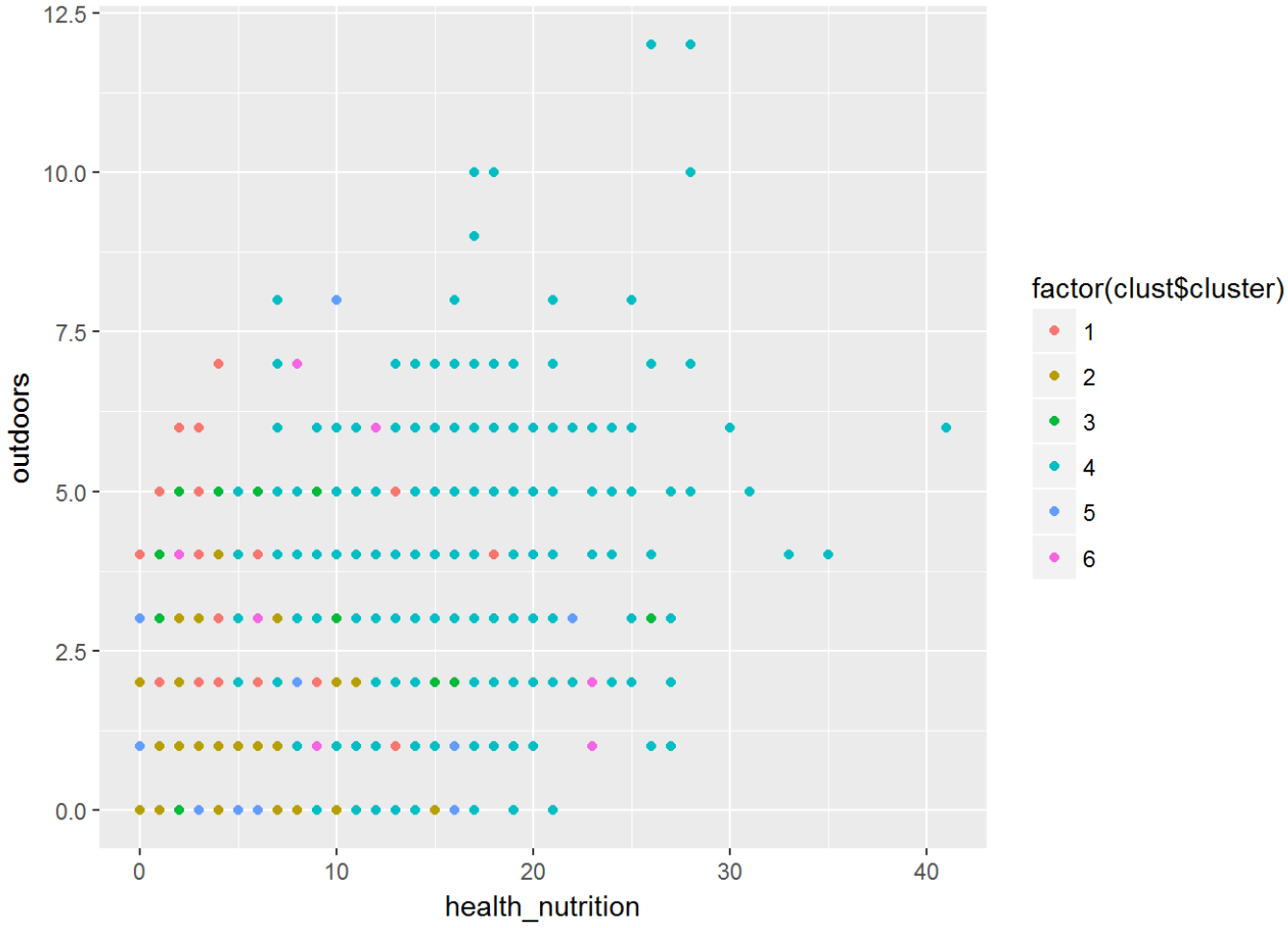

```
qplot(politics, current_events, data=social_mktg, color=factor(clust$cluster))
```



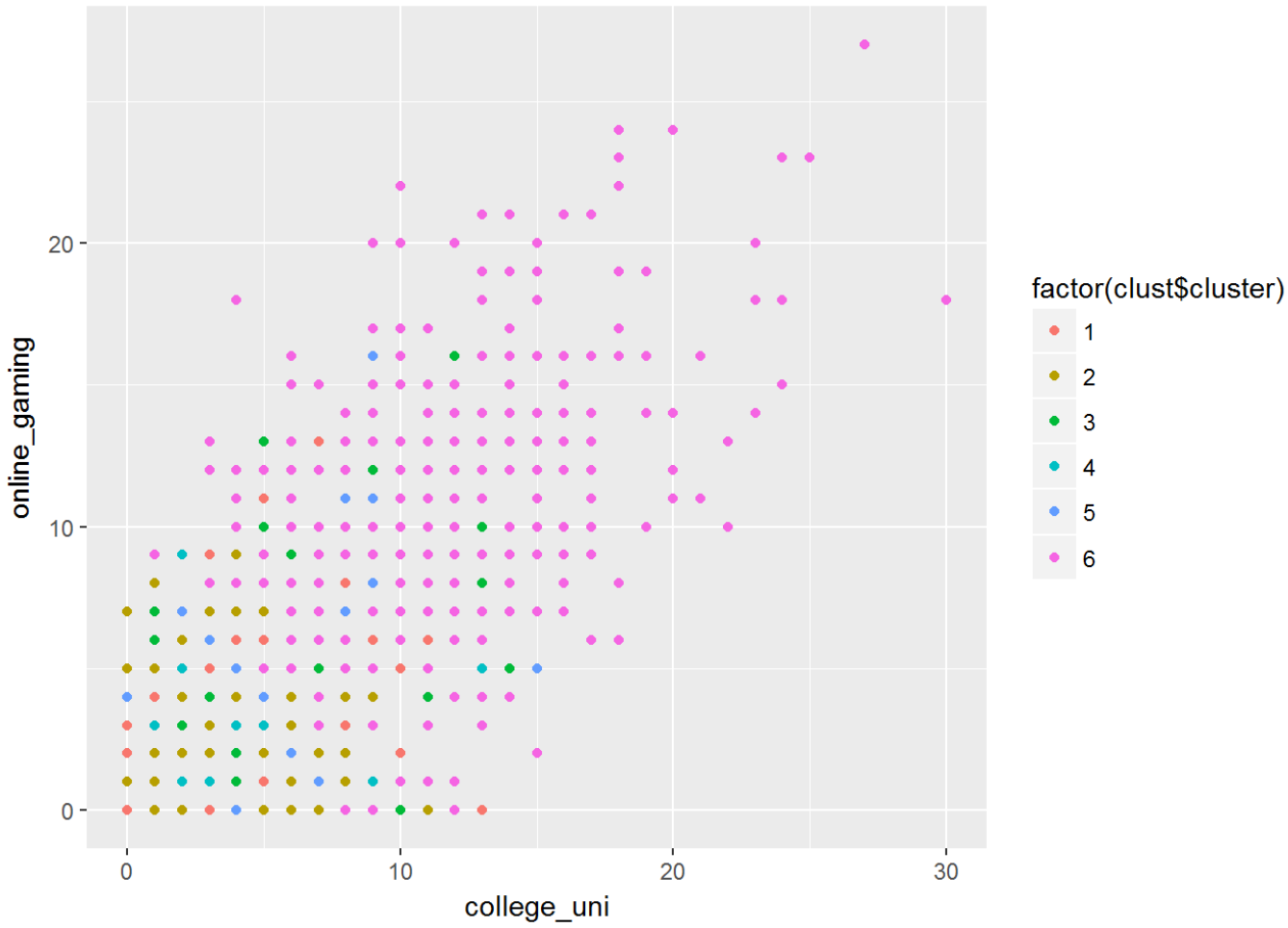
```
qplot(religion, parenting, data=social_mktg, color=factor(clust$cluster))
```




```
qplot(health_nutrition, outdoors, data=social_mktg, color=factor(clust$cluster))
```




```
qplot(college_uni, online_gaming, data=social_mktg, color=factor(clust$cluster))
```

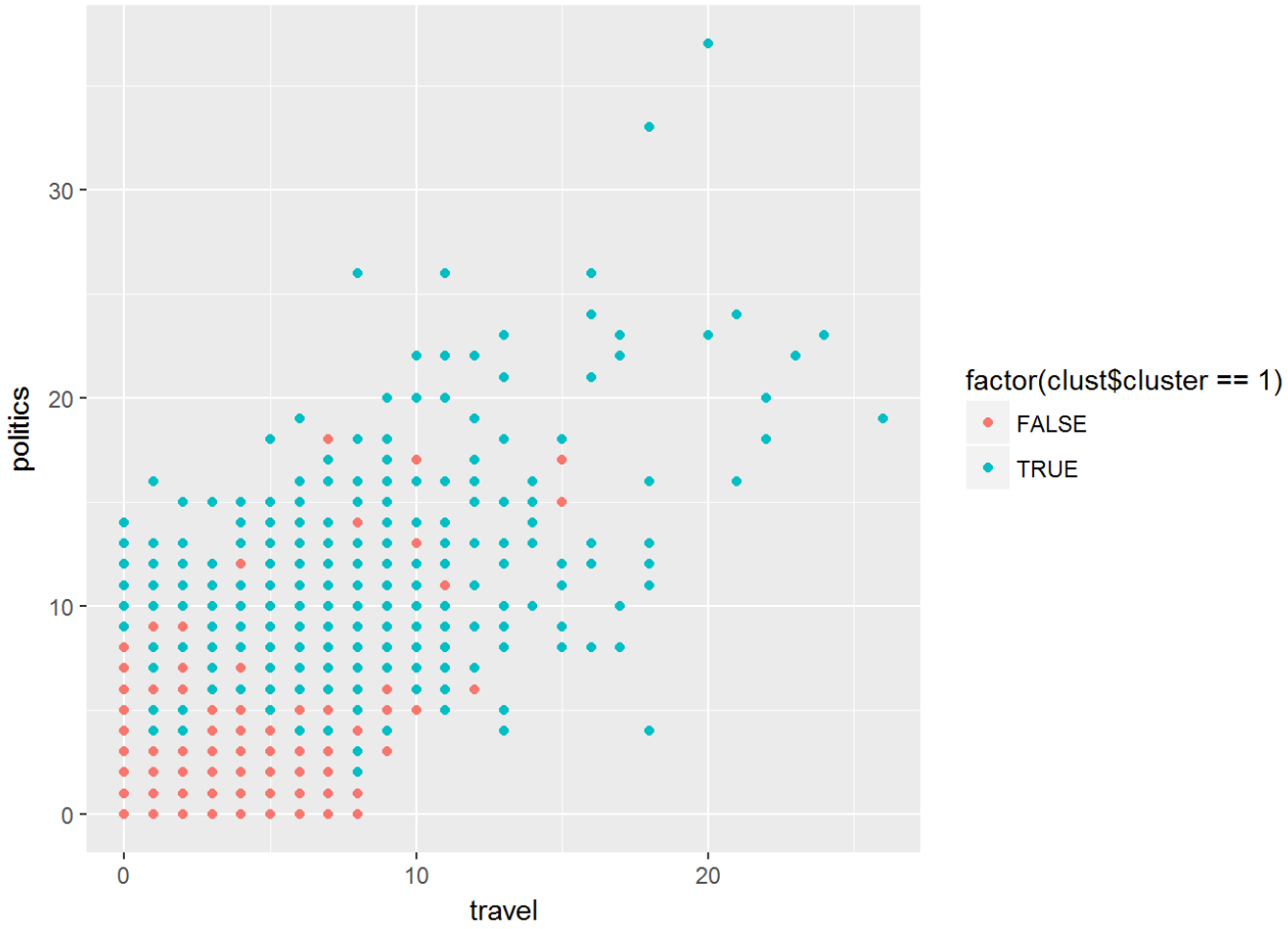


As seen from the clusters pattern, quite similar to the correlation plots plotted earlier, outdoors & health nutrition, politics & news, parenting & religion, college_uni & online_gaming are all in the same clusters mostly. Whereas cooking & food as also politics &current events seem to be in rather different clusters.

Segmenting market into 6 different cluster based on the interests(tweets) of the people

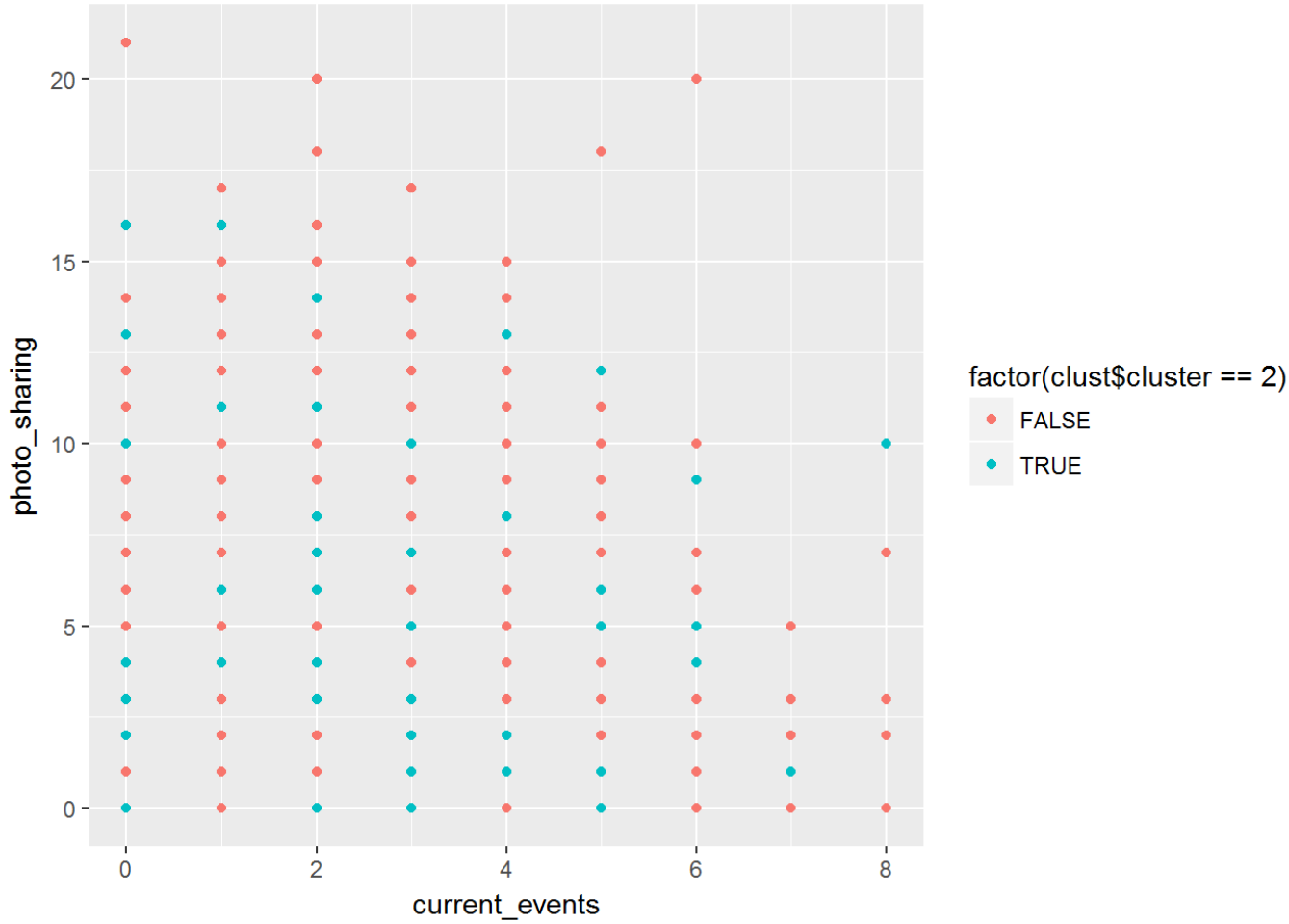
```
qplot(travel,politics,computers, data=social_mktg, color=factor(clust$cluster==1))
```

```
## Warning: Ignoring unknown parameters: NA
```



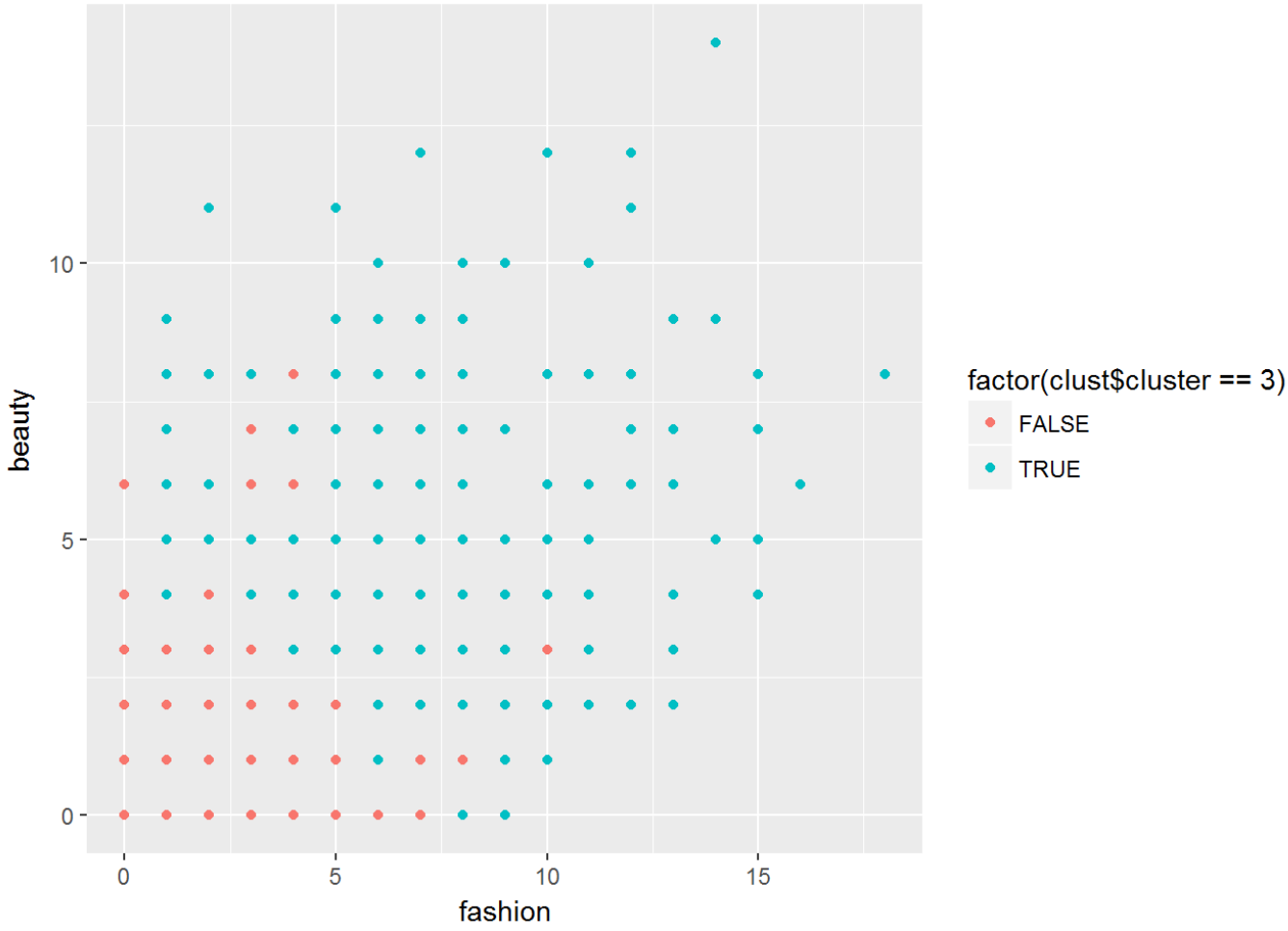

```
qplot(current_events,photo_sharing,tv_film, data=social_mktg, color=factor(clust$cluster==2))
```

```
## Warning: Ignoring unknown parameters: NA
```



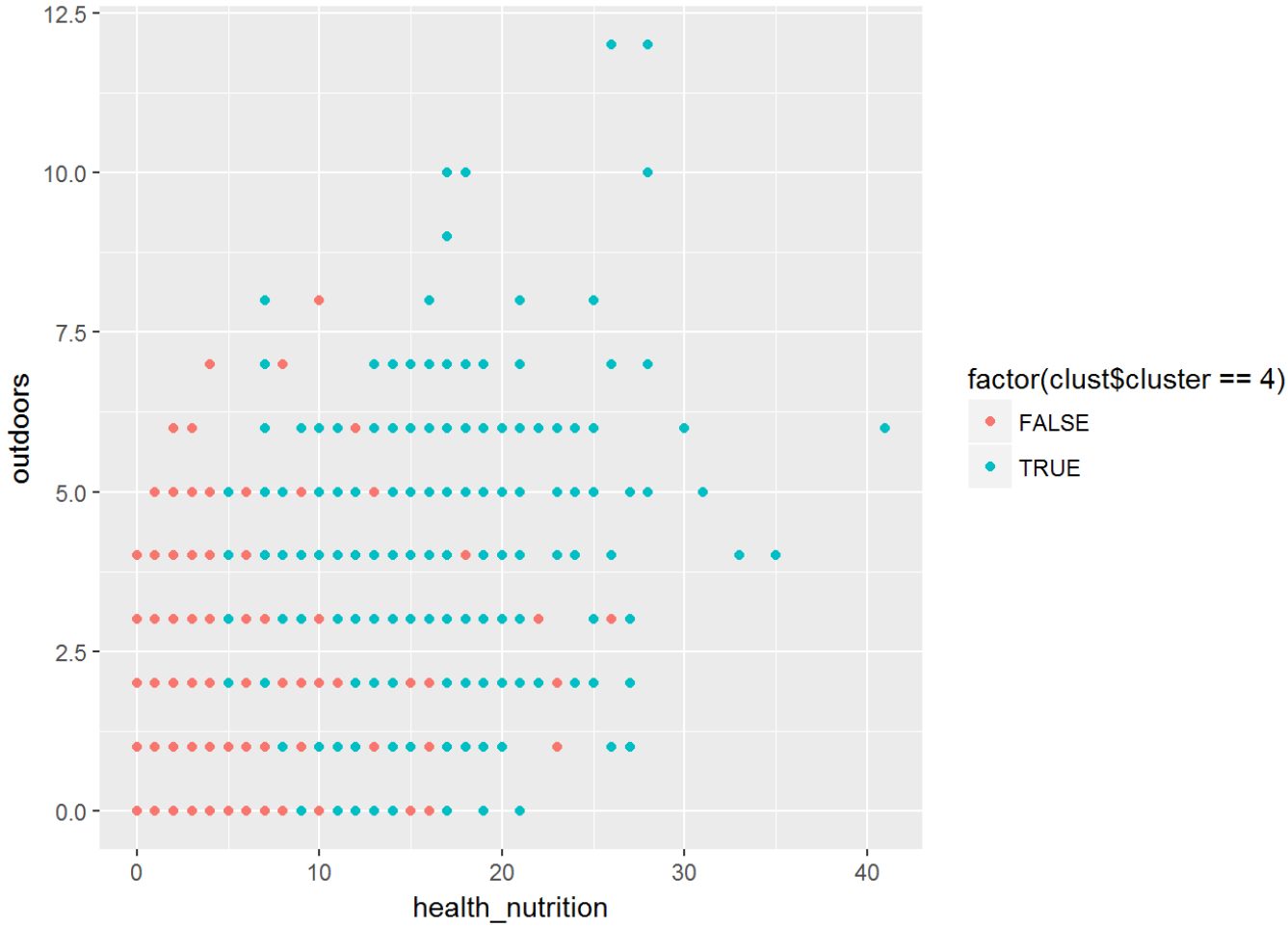

```
qplot(fashion,beauty,personal_fitness,shopping,dating, data=social_mktg, color=factor(clust$cluster==3))
```

```
## Warning: Ignoring unknown parameters: NA
```



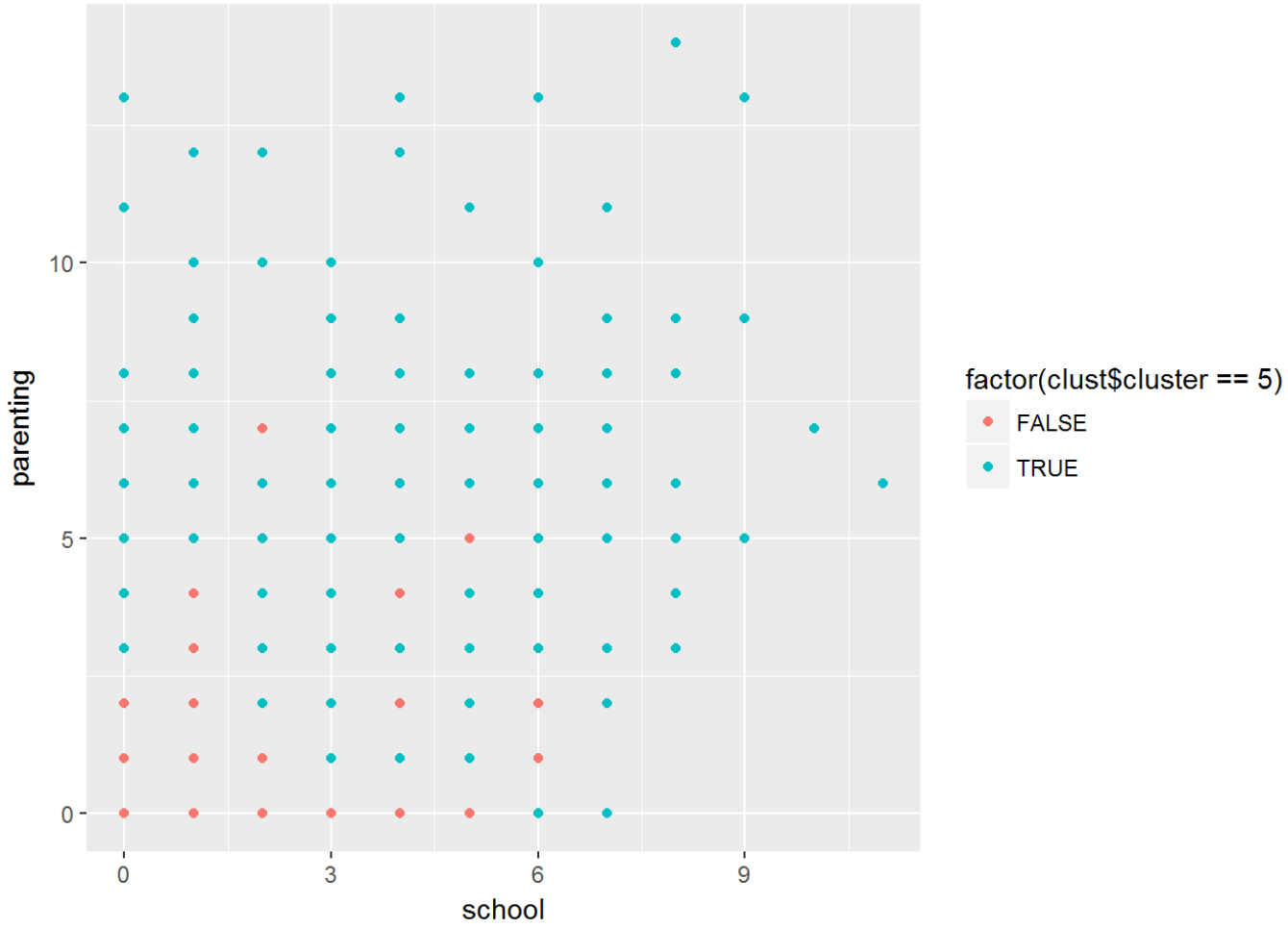

```
qplot(health_nutrition,outdoors,personal_fitness,  data=social_mktg, color=factor(c
lust$cluster==4))
```

```
## Warning: Ignoring unknown parameters: NA
```



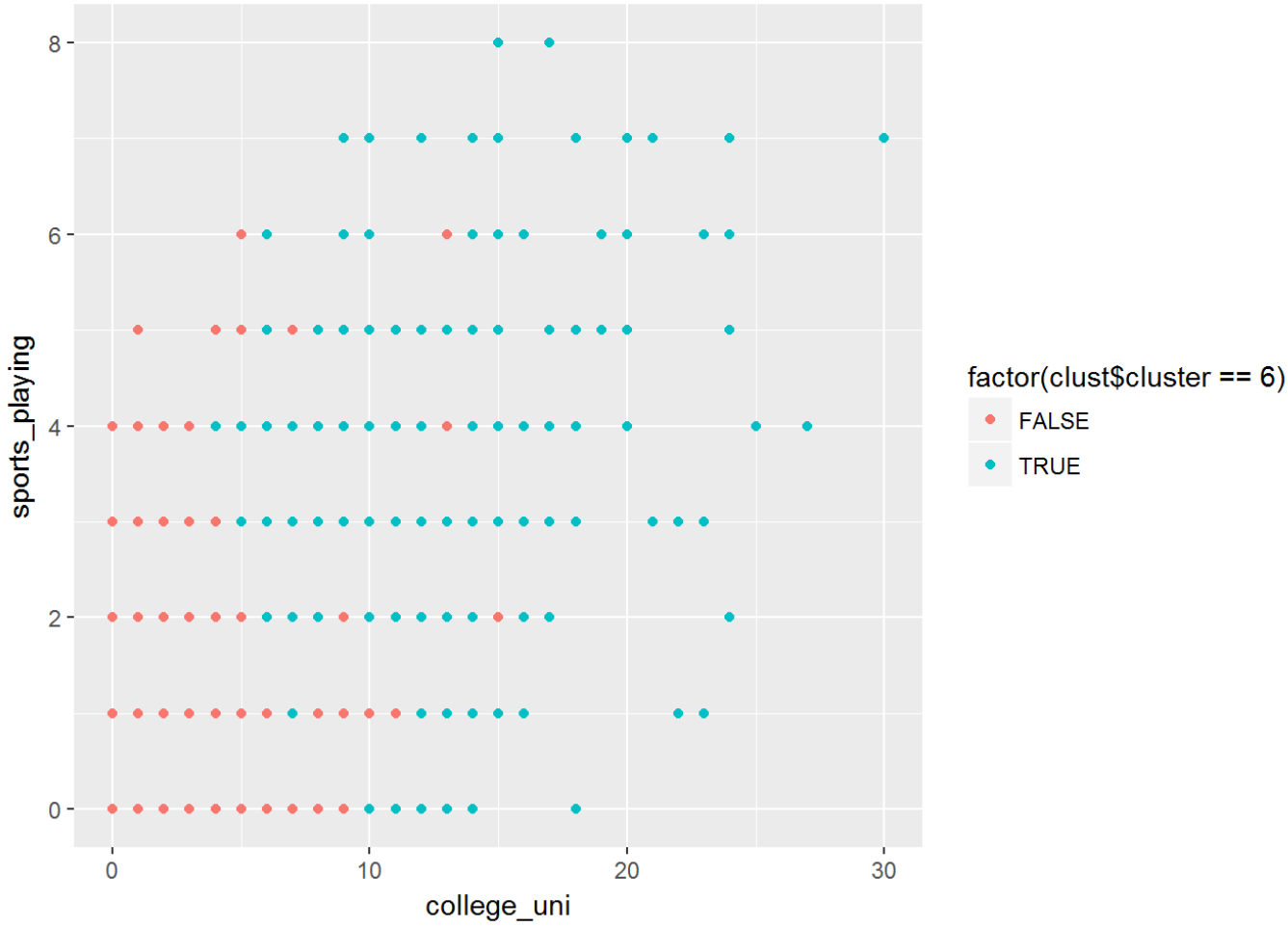

```
qplot(school,parenting,religion,food,family, data=social_mktg, color=factor(clust$cluster==5))
```

```
## Warning: Ignoring unknown parameters: NA
```




```
qplot(college_uni,sports_playing,online_gaming, data=social_mktg, color=factor(clust$cluster==6))
```

```
## Warning: Ignoring unknown parameters: NA
```




```
clust$center[1,]*sigma + mu
```

```
##   current_events           travel    photo_sharing          tv_film
##        1.6622999        5.6084425        2.5429403        1.2110626
##    sports_fandom         politics             food           family
##        2.0101892        8.9330422        1.4425036        0.9126638
##  home_and_garden            music             news    online_gaming
##        0.6128093        0.6390102        5.2940320        0.8558952
##          shopping health_nutrition      college_uni   sports_playing
##        1.3857351        1.6652111        1.3318777        0.6273654
##            cooking              eco        computers         business
##        1.2605531        0.5997089        2.4745269        0.6710335
##          outdoors           crafts       automotive              art
##        0.9184862        0.6433770        2.3362445        0.7510917
##          religion           beauty        parenting           dating
##        1.0291121        0.4759825        0.9432314        1.0655022
##             school personal_fitness          fashion   small_business
##        0.7248908        1.0072780        0.6768559        0.4861718
```

```
clust$center[2,]*sigma + mu
```

```
##   current_events           travel    photo_sharing          tv_film
##        1.4446903        1.0975664        2.2818584        0.9871681
##    sports_fandom         politics             food           family
##        0.9723451        1.0123894        0.7690265        0.5721239
##  home_and_garden            music             news    online_gaming
##        0.4384956        0.5519912        0.6942478        0.5803097
##         shopping health_nutrition      college_uni   sports_playing
##        1.2774336        1.0951327        0.8880531        0.4141593
##          cooking              eco        computers         business
##        0.8495575        0.3882743        0.3734513        0.3365044
##         outdoors           crafts       automotive              art
##        0.4024336        0.3623894        0.5809735        0.6221239
##         religion           beauty        parenting           dating
##        0.5254425        0.3491150        0.4584071        0.5471239
##           school personal_fitness          fashion   small_business
##        0.4769912        0.6590708        0.5148230        0.2761062
```

```
clust$center[3,]*sigma + mu
```

```
##   current_events           travel    photo_sharing          tv_film
##        1.7591623        1.4938918        6.1169284        1.0226876
##    sports_fandom         politics             food           family
##        1.1326353        1.4048866        1.0453752        0.9109948
##  home_and_garden            music             news    online_gaming
##        0.6212914        1.2146597        1.0506108        1.0471204
##         shopping health_nutrition      college_uni   sports_playing
##        2.0209424        2.2844677        1.4607330        0.8080279
##          cooking              eco        computers         business
##       10.9057592        0.5776614        0.7277487        0.6125654
##         outdoors           crafts       automotive              art
##        0.8132635        0.6352531        0.9057592        0.9057592
##         religion           beauty        parenting           dating
##        0.8446771        3.8935428        0.8062827        0.9441536
##           school personal_fitness          fashion   small_business
##        0.9842932        1.3560209        5.5305410        0.4851658
```

```
clust$center[4,]*sigma + mu
```

```
##   current_events           travel    photo_sharing          tv_film
##        1.5519187        1.2381490        2.6896163        0.9887133
##    sports_fandom         politics             food           family
##        1.1659142        1.2550790        2.1331828        0.7787810
##  home_and_garden            music             news    online_gaming
##        0.6467269        0.7347630        1.1038375        0.8510158
##         shopping health_nutrition      college_uni   sports_playing
##        1.4683973       12.0067720        0.9480813        0.6151242
##          cooking              eco        computers         business
##        3.2663657        0.9209932        0.5575621        0.4683973
##         outdoors           crafts       automotive              art
##        2.7415350        0.5936795        0.6625282        0.7449210
##         religion           beauty        parenting           dating
##        0.7607223        0.4243792        0.7652370        1.0349887
##           school personal_fitness          fashion   small_business
##        0.5959368        6.4537246        0.7945824        0.2945824
```

```
clust$center[5,]*sigma + mu
```

```
##   current_events           travel    photo_sharing          tv_film
##        1.6802097        1.3446920        2.6304063        1.0498034
##    sports_fandom         politics             food           family
##        5.8951507        1.1677588        4.5661861        2.4980341
##  home_and_garden            music             news    online_gaming
##        0.6461337        0.7313237        1.0340760        1.0117955
##         shopping health_nutrition      college_uni   sports_playing
##        1.4823067        1.8505898        1.2005242        0.7391874
##          cooking              eco        computers         business
##        1.6159895        0.6605505        0.7313237        0.5032765
##         outdoors           crafts       automotive              art
##        0.6815203        1.0825688        1.0498034        0.8754915
##         religion           beauty        parenting           dating
##        5.2555701        1.1009174        4.0589777        0.7942333
##           school personal_fitness          fashion   small_business
##        2.7090433        1.1887287        1.0275229        0.4010485
```

```
clust$center[6,]*sigma + mu
```

```
##   current_events           travel    photo_sharing          tv_film
##        1.5298013        1.5452539        2.8697572        1.9403974
##    sports_fandom         politics             food           family
##        1.3421634        1.2737307        1.2693157        1.0551876
##  home_and_garden            music             news    online_gaming
##        0.6158940        1.1368653        0.7902870        9.2516556
##         shopping health_nutrition      college_uni   sports_playing
##        1.4017660        1.7262693       10.3554084        2.5673289
##          cooking              eco        computers         business
##        1.4746137        0.4856512        0.5717439        0.4503311
##         outdoors           crafts       automotive              art
##        0.6710817        0.5960265        0.8896247        1.1876380
##         religion           beauty        parenting           dating
##        0.8476821        0.4547461        0.6732892        0.7373068
##           school personal_fitness          fashion   small_business
##        0.5253863        0.9955850        0.8962472        0.4944812
```

Summary: Cluster 1 consists of people who are interested in travel, sports fandom, politics, computers, which indiacte the well informed class of the society

Cluster 2 consists of people with mixed interests in different fields such as art, sports fandom, photosharing,current fandom and travel

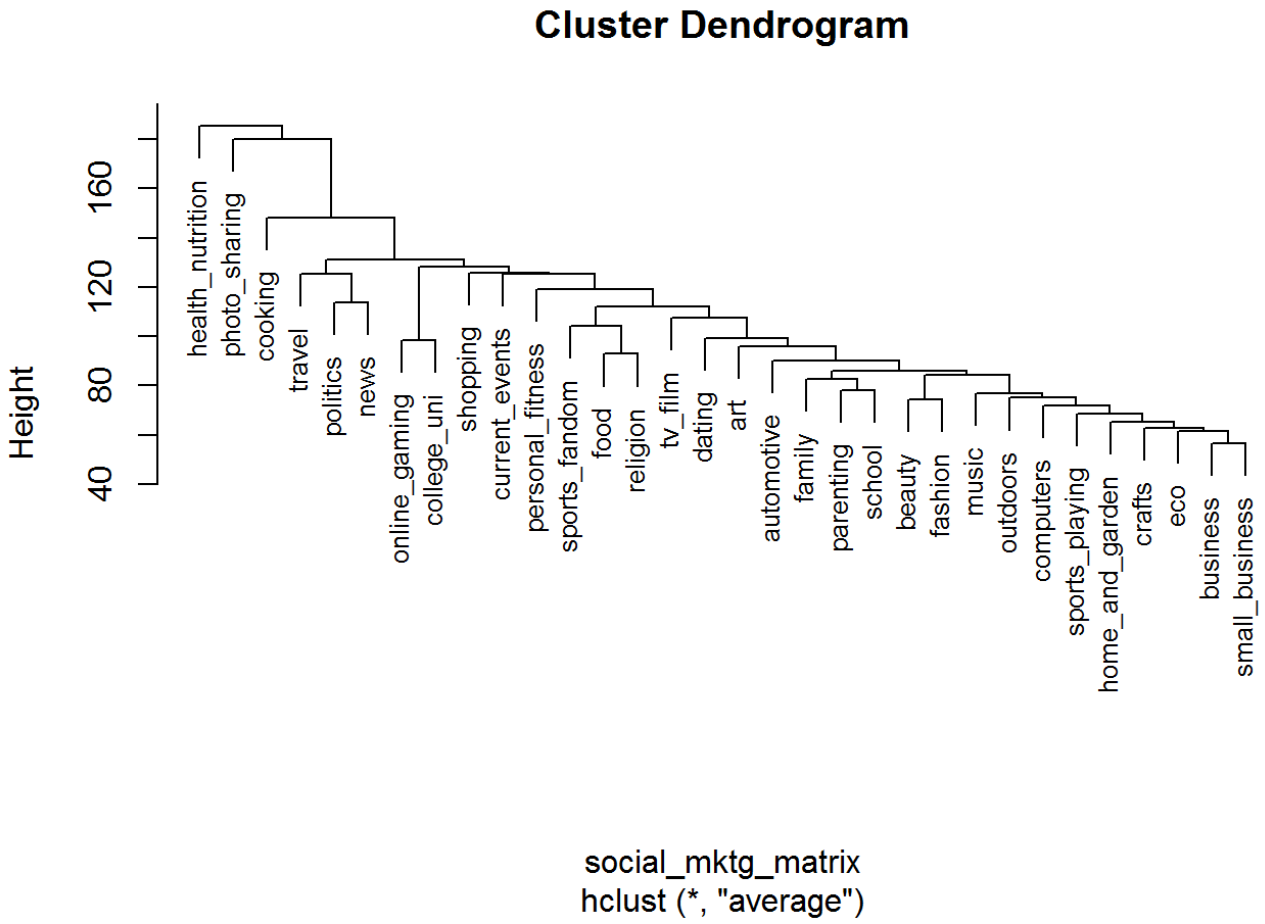
Cluster 3 consists of woman majorly as the tweets are majorly in fashion, beauty, personal fitness, shopping and dating.

Cluster 4 consists of people who are health conscious as it indicates people with interest in health nutrition, outdoors and personal fitness

Cluster 5 consists of parents mostly as the tweets are mostly in the categories of school, parenting, religion, food, family.

Cluster 6 represents mostly college students as tweets are mostly about college_university, sports playing and online gaming

```
s3<-t(social_mktg)
s3scaled <- scale(s3, center=TRUE, scale=TRUE)
social_mktg_matrix = dist(s3scaled, method='euclidean' )
hier_social_mktg = hclust(social_mktg_matrix, method='average')
plot(hier_social_mktg, cex=0.8)
```




```
cluster1 = cutree(hier_social_mktg, k=3)
cluster1
```

```
##   current_events           travel    photo_sharing          tv_film
##                1                1                2                1
##    sports_fandom         politics             food           family
##                1                1                1                1
##  home_and_garden            music             news    online_gaming
##                1                1                1                1
##         shopping   health_nutrition      college_uni   sports_playing
##                1                3                1                1
##          cooking              eco        computers         business
##                1                1                1                1
##         outdoors           crafts       automotive              art
##                1                1                1                1
##         religion           beauty        parenting           dating
##                1                1                1                1
##           school  personal_fitness          fashion   small_business
##                1                1                1                1
```

Similar to the segmentation obtained using K means, the hierarchical trees validates the same.