Predictive Modeling Assignment 1

Joey Chen, Jess Lee, Vincent Kuo, Matt Zlotnik

August 8, 2018

Probability practice

Answers - Part A

All the signs of "&&" below denotes intersection.

The following is given: P(RC) = 0.3 P(No|RC) = P(Yes|RC) =0.5 P(Yes) = 0.65

The fraction of people who are TC answered yes is P(yes|TC) - the probability of answering yes conditional on TC. To calculate: P(yes|TC) = P(yes && TC) / P(TC) - P(TC) = 1-P(RC) = 0.7 - P(Yes && RC) = P(RC)*P(Yes|RC) = 0.15 By the Law of Total Probability: - P(Yes && TC) = P(yes) - P(Yes && RC) = 0.65-0.15 = 0.5 Thus we have: P(yes|TC) = P(yes && TC) / P(TC) = 0.5/0.7 = 0.7143 (rounded)

Answer - Part B.

TP denotes being tested positive, while TN denotes being tested negative. RP denotes that in fact positive, while RN denotes otherwisely. Given the following: - P(TP|RP) = 0.993 = P(TP&&RP)/P(RP) - P(TN|RN) = 0.9999 = P(TN&&RN) / P(RN) - P(RP) = 0.000025 - P(RN) = 1-P(RP) = 0.999975 To derive the following: P(RP|TP), which is equivalent to P(RP&&TP)/P(TP)

P(TP&&RP) = 0.993*P(RP) = 0.000024825 P(TP) = P(TP&&RP)+ P(TP&&RN) = 0.0000999975 + 0.000024825 = , of which P(TP&&RN) is derived as following: P(TP&&RN) = P(RN)- P(TN&&RN) = 0.999975 -(P(RN)* P(TN|RN)) =0.999975 - 0.9998750025 = 0.0000999975

Thus, P(RP|TP) = P(RP&&TP)/P(TP) = 0.000024825/0.0001248225 = 0.19888241302 The probability that a patient has disease given he/she is tested positive is only around 0.1989. This method fails to conclude whether a patient has disease and will mis-suggest a positive outcome while the truth is negative.

Exploratory analysis: green buildings

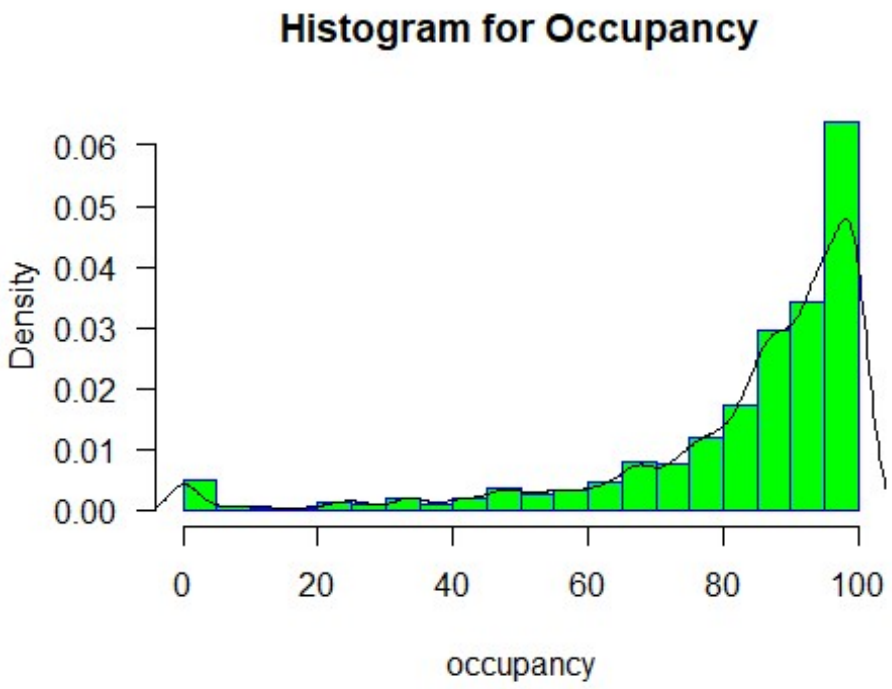
```
library(ggplot2)
library(dplyr)
library(lawstat)
library(vcd)
greenbuildings <- read.csv('../data/greenbuildings.csv', header=TRUE)
na.omit(greenbuildings)
```

To measure impact of green building on market value of building, first, it is a reasonable to compare greenbuilding sample and non-greenbuilding sample. However, for accurate

comparison, features other than 'green_ratings' of each sample should be controlled to be similar to one another.

The staff from the case controlled 'occupancy rate', however, this process turns out to be ineffective. To verify whether 'occupancy rate' is impacting 'rent' projection, we checked correlation between 'occupancy rate(leasing_rate)' and 'rent'.

```
#1.Comparing dataset without occupancy less than 10% to original
hist(greenbuildings$leasing_rate, main="Histogram for Occupancy",
xlab="occupancy",border="blue", col="green",las=1,breaks=16,prob = TRUE)
lines(density(greenbuildings$leasing_rate))
```

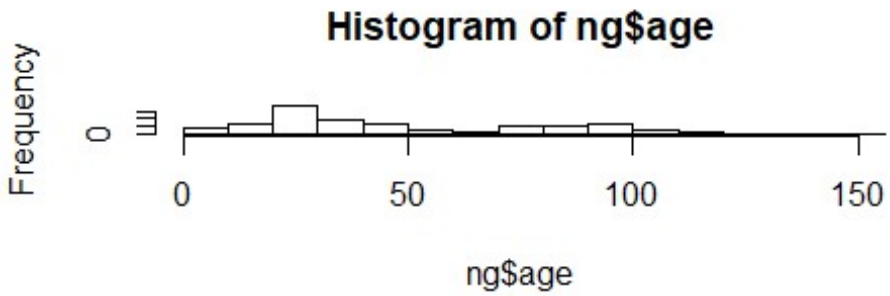


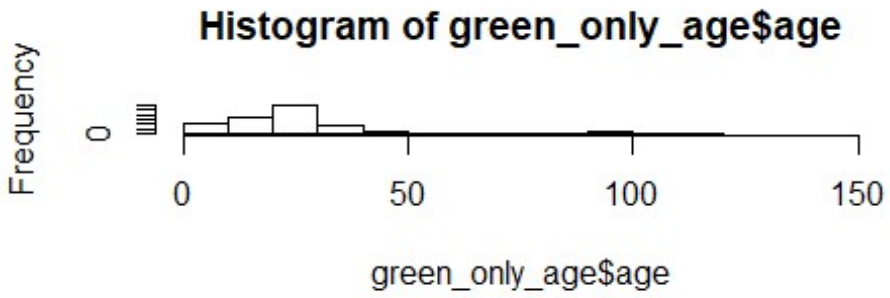
Since the distribution of 'occupancy rate' is skewed, here we used 'spearman' correlation.

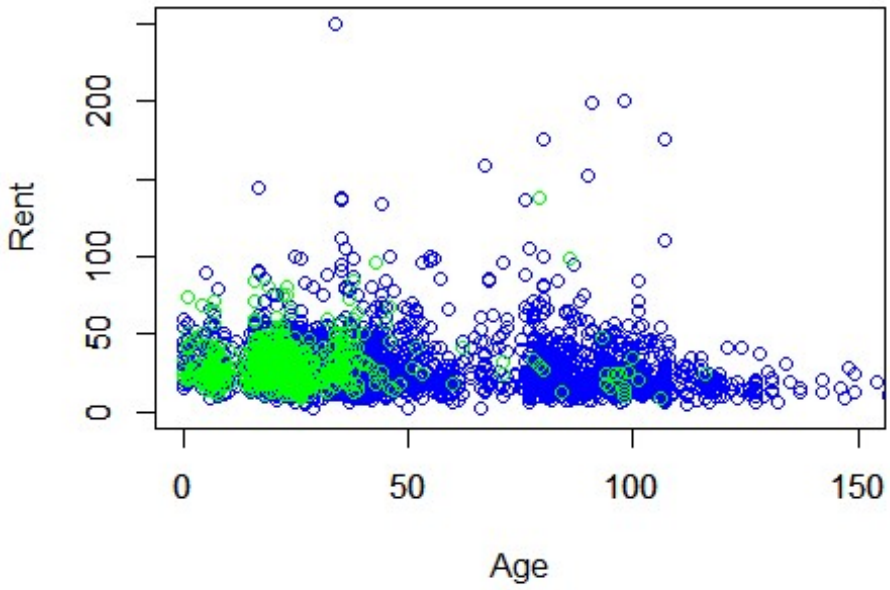
```
more10 <- greenbuildings%>%filter(leasing_rate > 0.1)
#Correlation between 'occupancy rate' and 'Rent' from buildings with
occupancy rate higer than 10%.
a <- cor(more10$leasing_rate, more10$Rent, method='spearman')
print(a)

## [1] 0.2307047

b <- cor(greenbuildings$leasing_rate, greenbuildings$Rent, method='spearman')
print(b)

## [1] 0.2409054
```

The difference between correlation based on processed data, 0.23, and one based on original data, 0.24, is marginal. Occupancy rate is not confounding variable, however, there are few other confounding variables which need to be controlled.

Confounding variable 1: 'age'







Age range of greenbuidlings is smaller than that of non-greenbuildings. Moreover, the number of non-greenbuildings based on age are somehow polarized; frequency concetrated at under 50 year-old and over 80 year-old.

```
rent_green = lm(Rent~., data=g)
rent_non_green = lm(Rent~., data=ng)
summary(rent_green)
```

```
##
## Call:
## lm(formula = Rent ~ ., data = g)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.845  -3.301  -0.473   2.761  60.139
##
## Coefficients: (2 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -8.369e+00  4.713e+00  -1.776   0.0762 .
## CS_PropertyID      6.200e-07  5.868e-07   1.057   0.2910
## cluster            1.748e-03  6.967e-04   2.509   0.0123 *
## size               4.898e-06  2.368e-06   2.069   0.0390 *
## empl_gr            7.097e-02  3.562e-02   1.992   0.0468 *
## leasing_rate       3.792e-02  2.091e-02   1.814   0.0702 .
## stories           -1.511e-02  5.103e-02  -0.296   0.7673
## age                1.188e-02  2.006e-02   0.592   0.5540
## renovated         -3.854e-01  6.896e-01  -0.559   0.5764
## class_a            1.162e+00  2.641e+00   0.440   0.6602
## class_b           -1.185e-01  2.628e+00  -0.045   0.9640
## LEED               2.395e+00  2.520e+00   0.950   0.3423
## Energystar         3.275e-01  2.679e+00   0.122   0.9027
## green_rating              NA         NA      NA       NA
## net               -7.755e-01  1.110e+00  -0.699   0.4850
## amenities         -1.648e+00  6.491e-01  -2.539   0.0113 *
## cd_total_07       -1.091e-04  3.277e-04  -0.333   0.7392
## hd_total07         2.737e-04  2.249e-04   1.217   0.2240
## total_dd_07               NA         NA      NA       NA
## Precipitation      4.971e-02  3.997e-02   1.244   0.2141
## Gas_Costs         -4.060e+02  2.132e+02  -1.904   0.0574 .
## Electricity_Costs  1.389e+02  5.583e+01   2.488   0.0131 *
## cluster_rent       1.107e+00  3.299e-02  33.541   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.517 on 658 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.7531, Adjusted R-squared:  0.7456
## F-statistic: 100.3 on 20 and 658 DF,  p-value: < 2.2e-16
```

```
summary(rent_non_green)
```

```
##
## Call:
## lm(formula = Rent ~ ., data = ng)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.560  -3.646  -0.527   2.509 173.791
##
## Coefficients: (4 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -8.319e+00  1.075e+00  -7.738 1.15e-14 ***
## CS_PropertyID      2.869e-07  1.643e-07   1.746 0.080797 .
## cluster            7.090e-04  3.037e-04   2.335 0.019572 *
## size               6.853e-06  6.850e-07  10.003  < 2e-16 ***
## empl_gr            6.440e-02  1.856e-02   3.470 0.000523 ***
## leasing_rate       8.117e-03  5.560e-03   1.460 0.144360
## stories           -3.458e-02  1.720e-02  -2.010 0.044492 *
## age               -1.272e-02  4.936e-03  -2.577 0.009995 **
## renovated         -1.643e-01  2.750e-01  -0.597 0.550361
## class_a            2.867e+00  4.590e-01   6.247 4.41e-10 ***
## class_b            1.149e+00  3.533e-01   3.253 0.001149 **
## LEED                      NA         NA      NA       NA
## Energystar                NA         NA      NA       NA
## green_rating              NA         NA      NA       NA
## net               -2.741e+00  6.546e-01  -4.187 2.86e-05 ***
## amenities          8.227e-01  2.688e-01   3.060 0.002221 **
## cd_total_07       -1.382e-04  1.592e-04  -0.868 0.385296
## hd_total07         5.611e-04  9.630e-05   5.826 5.92e-09 ***
## total_dd_07               NA         NA      NA       NA
## Precipitation      5.072e-02  1.733e-02   2.926 0.003447 **
## Gas_Costs         -3.578e+02  8.382e+01  -4.269 1.99e-05 ***
## Electricity_Costs  1.949e+02  2.714e+01   7.182 7.57e-13 ***
## cluster_rent       9.998e-01  1.539e-02  64.949  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.633 on 7122 degrees of freedom
##   (68 observations deleted due to missingness)
## Multiple R-squared:  0.6039, Adjusted R-squared:  0.6029
## F-statistic: 603.4 on 18 and 7122 DF,  p-value: < 2.2e-16
```

According to linear regression result, age does not impact upon greenbuildings' statistically significantly, however, age impacts rent among non-greenbuildings. Thus we controlled the range of age to under 40 and checked median.

```
#Based on initial histogram, cut out data based on age under 40.
g40 <- g%>%filter(age<40)
ng40 <- ng%>%filter(age<40)
#Then compared median of orginal data and age-controlled data
print(median(ng40$Rent)-median(g40$Rent))
```

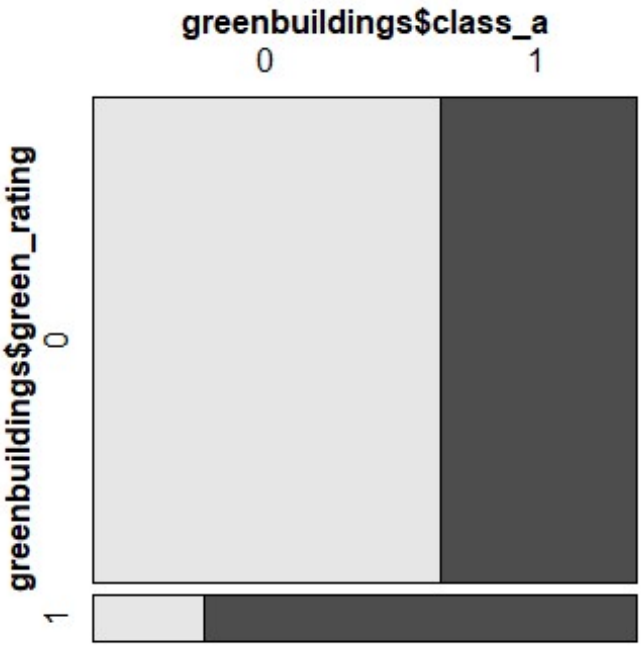
```
## [1] -1.2
```

```
print(median(ng$Rent)-median(g$Rent))
```

```
## [1] -2.6
```

Comparing to median from original data, after age control, median changed by 1.4 per square, which would significantly change whole rent. If 'age' was not confounding variable, the median after processing should have not been changed. Thus, first confounding variable 'age' should be controlled and to do so, we will filter out buildings with age lower than 40.

Confounding variable 2: 'class'

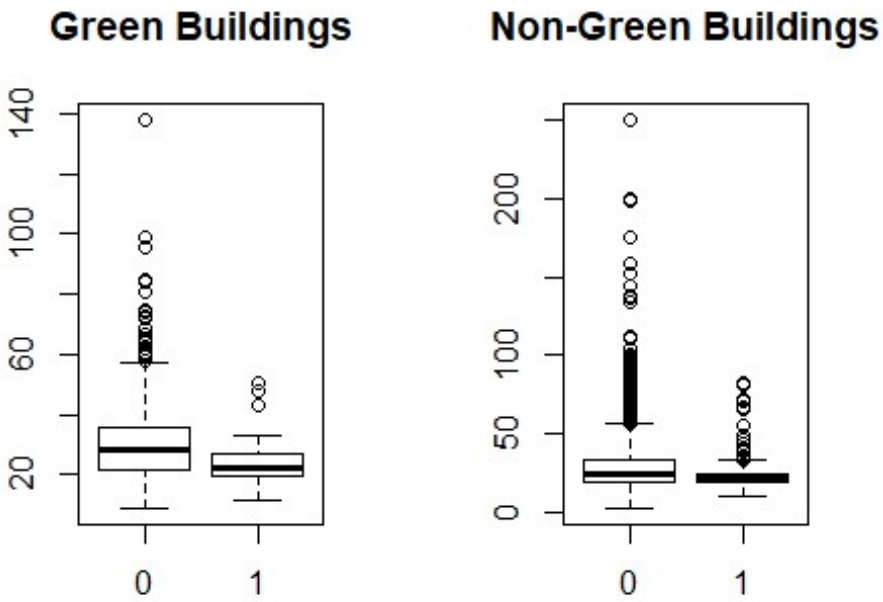



```
## [1] 0.7970803
```

```
## [1] 0.3621862
```

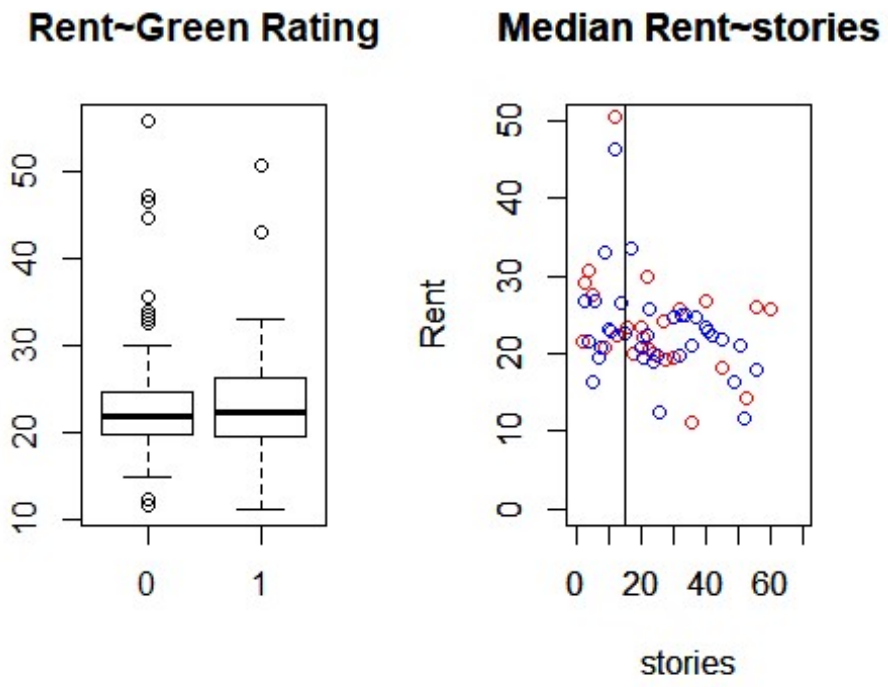
Green buildings has much higher portion of a-class(about 80%) comparing to non green buildings(about 36%). Considering the fact that buildings with a-class tend to have higher rent, variable 'class-a' has high potential to work as a confounding variable and impact the result significantly. Thus, both greenbuildings and non-greenbuildins data should control 'class_a', and here, as a way to control, we suggest to consider only a-class buildings.

Confounding variable 3: 'net'



'0' represents building including amenities in Rent and '1' represents Rent withoug amenities. Boxplots are describing that impact of net on rent varies of greenbuildings is not similar to that of non-greenbuildings. Since 'net' is a confounding variable, here, we decided to control data by including only 'net=1' data because it has less outliers in both cases, greenbuildings and non-greebuildings.

Comparing rent based on confounding-variable-controlled samples and simple expected return estimation



According to box plot on the left, greenbuidlings('1') earn higher rents in average than non-greenbuidings('0'). There is a chance to have extremely high rent return for non-greenbuilding owners, however, greenbuilding owners have more chance to have relatively higher return in average. Also, especially in case of 15-story green building from the case, it is expected to have about $21 per square rent income in average. However, in prespective of stories of building, it seems that there is no significant amount of premium for green-buildings('red dot') compared to non-green buildings('blue dot'). Since variable 'stories' has less statistics significance, more information about target greenbuilding(building for investment) is required and revenue-projection should be done from perspectives from those information.

Bootstrapping

```
rm(list=ls())
library(mosaic)
library(quantmod)
library(foreach)

mystocks = c("SPY","TLT","LQD","EEM","VNQ")
getSymbols(mystocks)

for(ticker in mystocks) {
  expr = paste0(ticker, "a = adjustOHLC(", ticker, ")")
```

```
  eval(parse(text=expr))
}

all_returns = cbind(ClCl(SPYa),ClCl(TLTa),ClCl(LQDa),ClCl(EEMa),ClCl(VNQa))
all_returns = as.matrix(na.omit(all_returns))
```

Now we have all five assets' daily returns. We want to understand the risk return properties of these assets. We will use variance of the return to capture risk of the assets. If the variance of the return is high, than this ETF might be a high risk asset.

```
# risk/return properties
etf_var <- apply(all_returns,2,var)
etf_mean <- apply(all_returns,2,mean)

etf_var[which.min(etf_var)]

##    ClCl.LQDa
## 2.718333e-05

etf_mean[which.min(etf_mean)]

##    ClCl.LQDa
## 0.0002095494

# LQD is the safest

etf_var[which.max(etf_var)]

##   ClCl.EEMa
## 0.001616072

etf_mean[which.max(etf_mean)]

##    ClCl.EEMa
## 0.0009813682

# EEM is the most aggressive
```

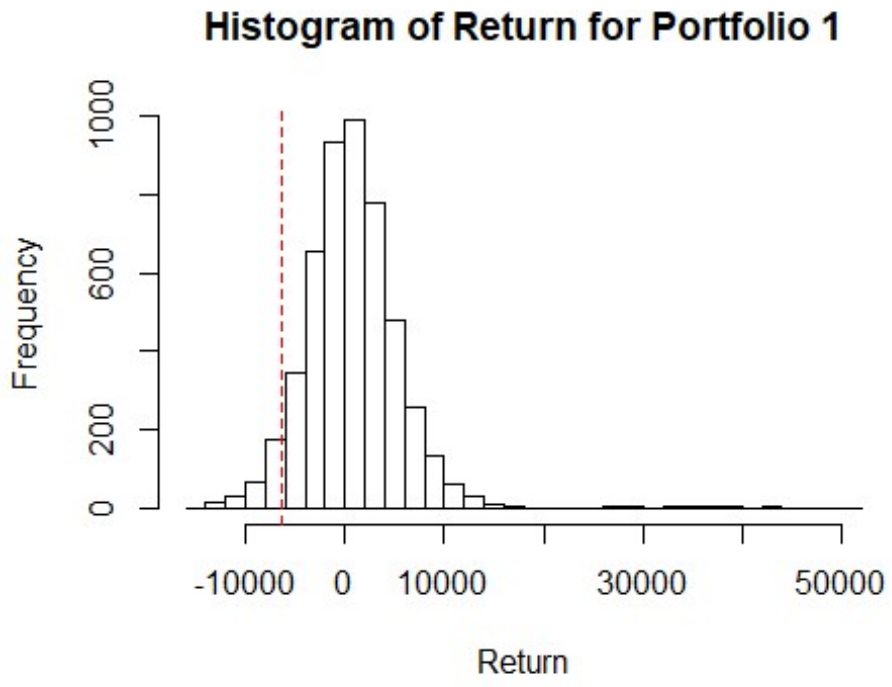
Now we know LQD has the lowest risk and the lowest return, so LQD is the safest ETF. On the other hand, EEM has the highest average return but has the highest risk. After understanding those properties, we should start to consider our asset allocation. Because ETFs might have correlation, the safest strategy will be not to choose highly correlated ETFs combination to distribute the risk. So, we want to know the correlation among all the ETFs.

```
cor(all_returns)

##            ClCl.SPYa  ClCl.TLTa  ClCl.LQDa   ClCl.EEMa   ClCl.VNQa
## ClCl.SPYa  1.0000000 -0.4362148 0.10133468  0.40676425  0.76813129
## ClCl.TLTa -0.4362148  1.0000000 0.43237319 -0.16758098 -0.25332123
## ClCl.LQDa  0.1013347  0.4323732 1.00000000  0.08784764  0.07156075
```

```
## ClCl.EEMa  0.4067643 -0.1675810 0.08784764  1.00000000  0.29228612
## ClCl.VNQa  0.7681313 -0.2533212 0.07156075  0.29228612  1.00000000
```

Portfolio 1

```
set.rseed(99)
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.today = resample(all_returns, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return.today
    total_wealth = sum(holdings)
    wealthtracker[today] = total_wealth
  }
  wealthtracker
}
return1 = mean(sim1[,n_days])
hist(sim1[,n_days]- initial_wealth, breaks=30,main = "Histogram of Return for
Portfolio 1",xlab = "Return")
VaR1 = quantile(sim1[,n_days], 0.05) - initial_wealth
abline(v=VaR1,col="red",lty=2)
```




```
cat("Portfolio 1: final wealth = ",return1,"\n")
```

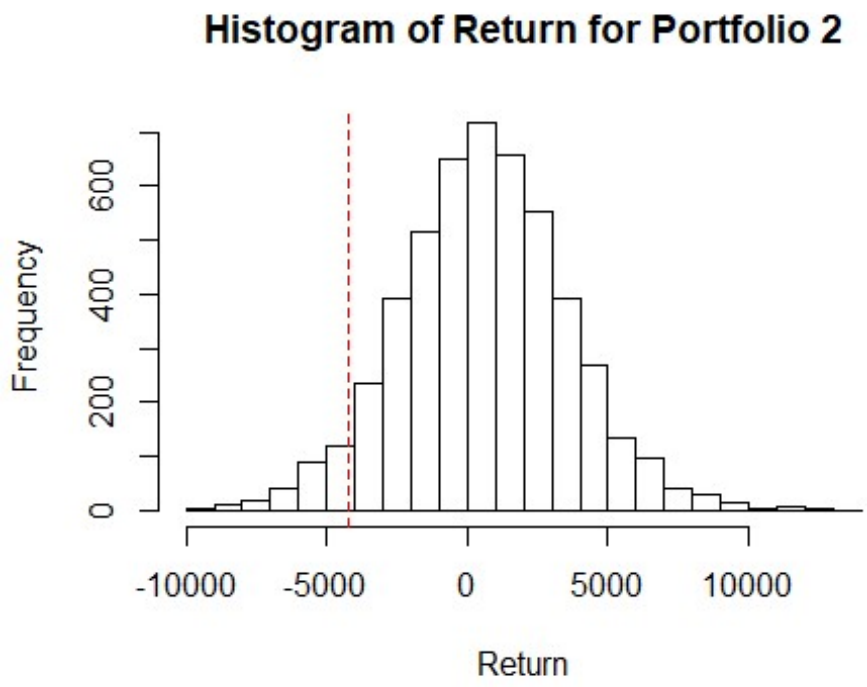
```
## Portfolio 1: final wealth =  100911.2

cat("Portfolio 1: 5% level Value at Risk = ",VaR1)

## Portfolio 1: 5% level Value at Risk =  -6253.86
```

For Portfolio 1, we evenly split assets in each of the five ETFs. The red line on the histogram is the 5% VaR. After using bootstrap to simulate 4-week trading day, the final wealth is 100911.2 and the 5% level VaR is -6253.86. Let's use another stratgey in Portfolio 2.

Portfolio 2

For Portfolio 2, we want to use a safer strategy. We already have some understanding of risk properties about all five ETFs, and for safer allocation, we want to drop EEM due to the high risk. Furthermore, we put more weight on the low risk ETFs (TLT & LQD).

```
set.rseed(99)
initial_wealth = 100000
sim2 = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0.2, 0.3, 0.3, 0, 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
}

return2 = mean(sim2[,n_days])
hist(sim2[,n_days]- initial_wealth, breaks=30,main = "Histogram of Return for
Portfolio 2",xlab = "Return")
VaR2 = quantile(sim2[,n_days], 0.05) - initial_wealth
abline(v=VaR2,col="red",lty=2)
```




```
cat("Portfolio 2: final wealth = ",return2,"\n")
```

```
## Portfolio 2: final wealth =  100617.6
```

```
cat("Portfolio 2: 5% level Value at Risk = ",VaR2)
```

```
## Portfolio 2: 5% level Value at Risk =  -4222.945
```

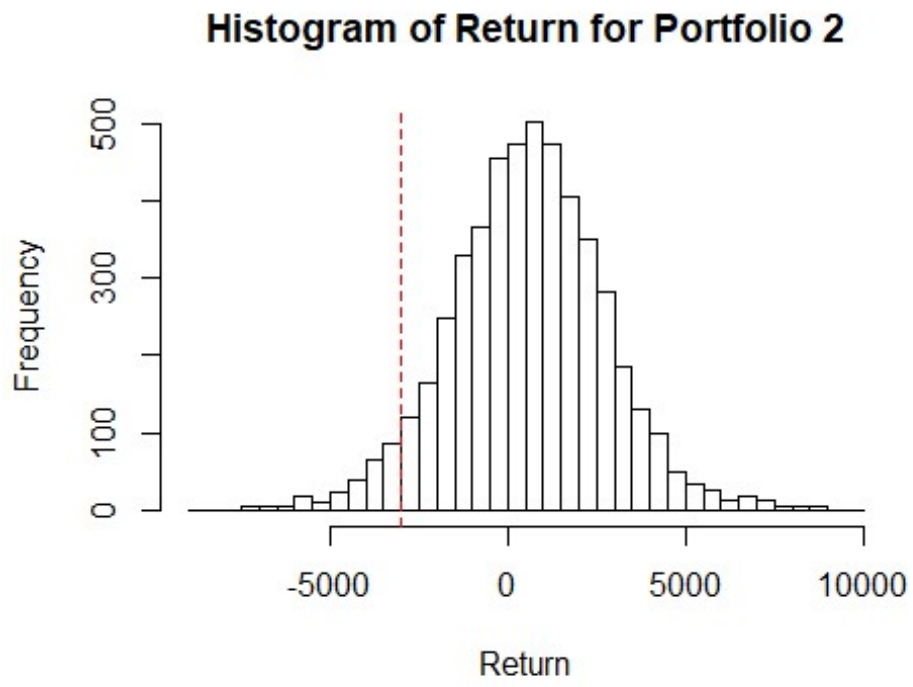
The result shows us that we will have less return but also have less risk compared to Portfolio 1. We will have 290 dollars less profit but the VaR is at -4222.945 dollars. However, we think this is not the safest strategy. We talked about the correlation among ETFs and we think we should take this into account.

```
cor(all_returns)
```

```
##            ClCl.SPYa  ClCl.TLTa  ClCl.LQDa   ClCl.EEMa   ClCl.VNQa
## ClCl.SPYa  1.0000000 -0.4362148 0.10133468  0.40676425  0.76813129
## ClCl.TLTa -0.4362148  1.0000000 0.43237319 -0.16758098 -0.25332123
## ClCl.LQDa  0.1013347  0.4323732 1.00000000  0.08784764  0.07156075
## ClCl.EEMa  0.4067643 -0.1675810 0.08784764  1.00000000  0.29228612
## ClCl.VNQa  0.7681313 -0.2533212 0.07156075  0.29228612  1.00000000
```

In the correlation matrix, we can see that SPY and VNQ has a really high correlation. If we want to take the safest allocation strategy, we should avoid choosing ETFs with high correlation to prevent risk. Thus, now we use another allocation tht we think is safer. We drop VNQ due to the high correlation with SPY, and run the bootstrap.

```r
set.rseed(99)
initial_wealth = 100000
sim2_new = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0.3, 0.3, 0.4, 0, 0)
  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
}

return2_new = mean(sim2_new[,n_days])
hist(sim2_new[,n_days]- initial_wealth, breaks=30,main = "Histogram of Return
for Portfolio 2",xlab = "Return")
VaR2_new = quantile(sim2_new[,n_days], 0.05) - initial_wealth
abline(v=VaR2_new,col="red",lty=2)
```




```r
cat("Portfolio 2: final wealth = ",return2_new,"\n")

## Portfolio 2: final wealth =  100577.8
```

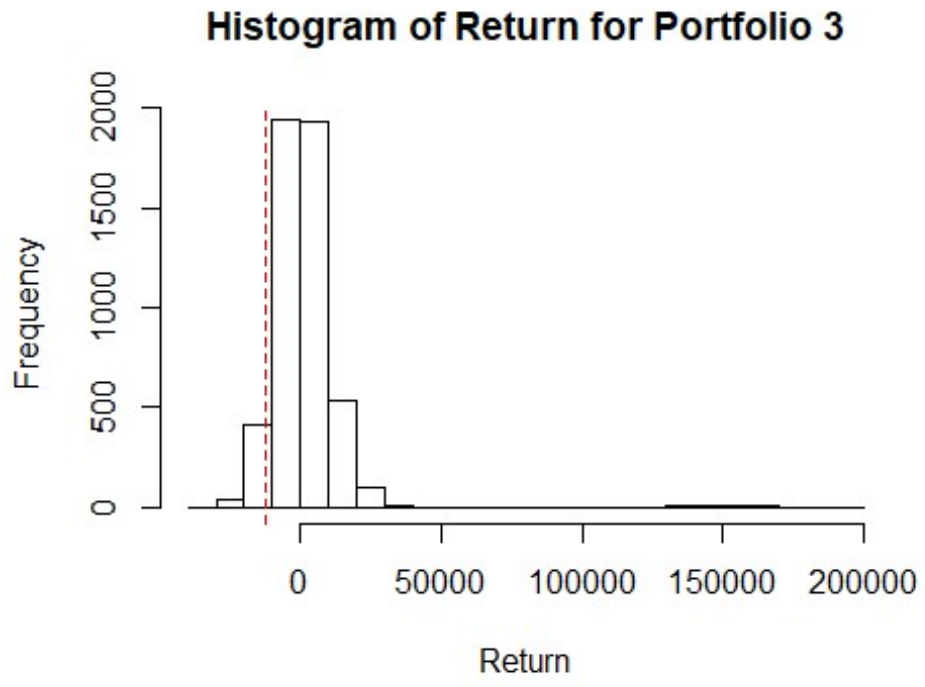
```
cat("Portfolio 2: 5% level Value at Risk = ",VaR2_new)

## Portfolio 2: 5% level Value at Risk =  -3005.832
```

As we can see, the final wealth is about 40 dollars less but VaR is about 1200 dollars more! And we can see that the histogram converges to the middle. It means that we are getting a lower risk result by sacrificing higher return. In conclusion, we should not only consider ETFs variance but also consider the correlation between ETFs. Now, let's think of a aggressive strategy.

Portfolio 3

When it comes to more aggressive strategy, we plan to put most of our assets on high-return and high-risk ETFs. The reason why we choose EEM and VNQ is that they have the top 2 high average return. Besides, these two ETFs' correlation is about 0.3 and this time we want the correlation to be high. The reason is that we want both of the ETFs to grow together and make more profits more aggressively.

```
set.rseed(99)
initial_wealth = 100000
sim3 = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0,0,0,0.8,0.2)
  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
}

return3 = mean(sim3[,n_days])
hist(sim3[,n_days]- initial_wealth, breaks=30,main = "Histogram of Return for
Portfolio 3",xlab = "Return")
VaR3 = quantile(sim3[,n_days], 0.05) - initial_wealth
abline(v=VaR3,col="red",lty=2)
```




```
cat("Portfolio 3: final wealth = ",return3,"\n")
```

```
## Portfolio 3: final wealth =  101755
```

```
cat("Portfolio 3: 5% level Value at Risk = ",VaR3)
```

```
## Portfolio 3: 5% level Value at Risk =  -12648.89
```

As we can see, the histogram is right-skewed. We get several high return results but we do have some big loss too. The final wealth we have is 101755, almost 850 dollars more than Portfolio 1. However, we do have a -12648.89 5% VaR. In conclusion, we recommend readers to choose between Portfolio 2 and Portfolio 3. Choosing Portfolio 1 will yield small return and prevent small risk. If readers want to choose a safer allocation strategy, Portfolio 2 provides them a reasonable profits with a low risk. If readers try to be aggressive, Portfolio 3 will yield the most return but have more risk. However, we want to emphasize that we should always consider the correlation among the assets. People can always distribute the risk by choosing small correlation ETFs.

Market Segmentation

PCA

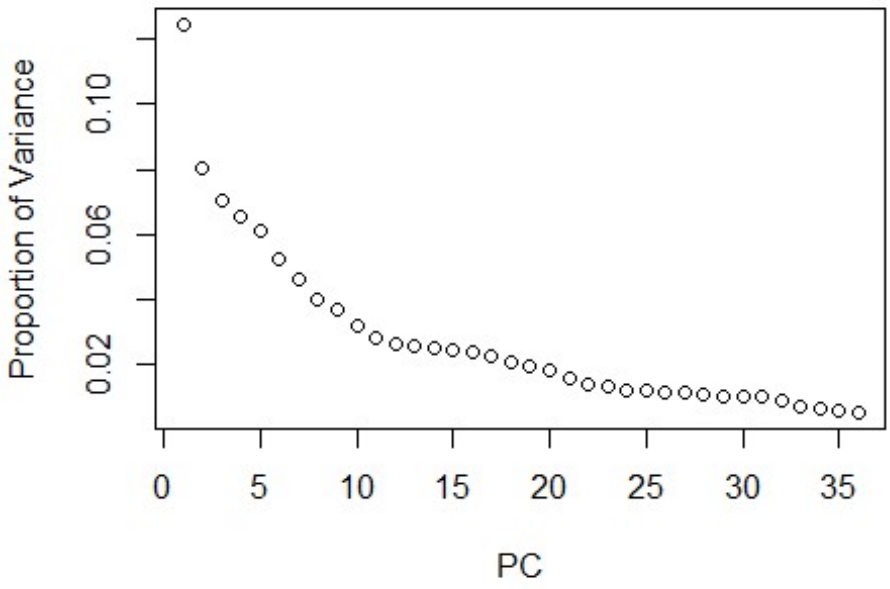
To try and find interesting market segments, we primarily used two modeling techniques. First, we tried to use Principal Components Analysis to try and analyze groupings of all 36 variables in the dataset.

According to the PCA summary as well as the plot below, we realized that, with only 2 dimensions, the explained variance is only 20%.

```
set.seed(2)
data_scaled = scale(data[,-1], center = T, scale = T)
pca <- prcomp(data_scaled, scale = T)
summary(pca)
```

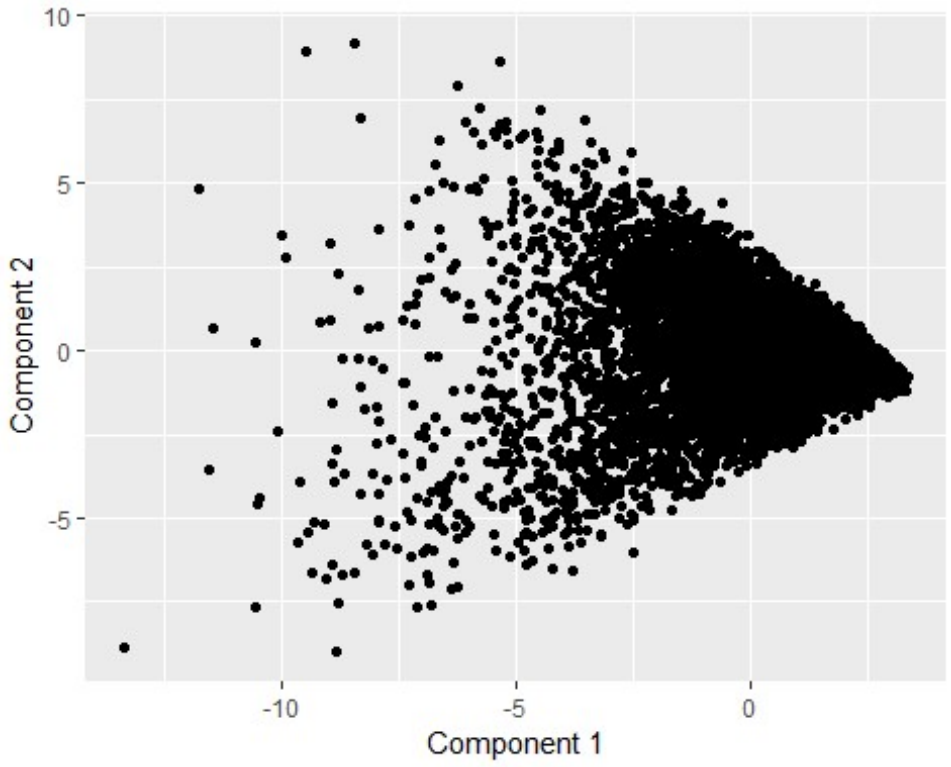
```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6
## Standard deviation     2.1186 1.69824 1.59388 1.53457 1.48027 1.36885
## Proportion of Variance 0.1247 0.08011 0.07057 0.06541 0.06087 0.05205
## Cumulative Proportion  0.1247 0.20479 0.27536 0.34077 0.40164 0.45369
##                           PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     1.28577 1.19277 1.15127 1.06930 1.00566 0.96785
## Proportion of Variance 0.04592 0.03952 0.03682 0.03176 0.02809 0.02602
## Cumulative Proportion  0.49961 0.53913 0.57595 0.60771 0.63580 0.66182
##                          PC13    PC14    PC15    PC16   PC17    PC18
## Standard deviation     0.96131 0.94405 0.93297 0.91698 0.9020 0.85869
## Proportion of Variance 0.02567 0.02476 0.02418 0.02336 0.0226 0.02048
## Cumulative Proportion  0.68749 0.71225 0.73643 0.75979 0.7824 0.80287
##                          PC19    PC20    PC21    PC22    PC23    PC24
## Standard deviation     0.83466 0.80544 0.75311 0.69632 0.68558 0.65317
## Proportion of Variance 0.01935 0.01802 0.01575 0.01347 0.01306 0.01185
## Cumulative Proportion  0.82222 0.84024 0.85599 0.86946 0.88252 0.89437
##                          PC25    PC26    PC27    PC28    PC29    PC30
## Standard deviation     0.64881 0.63756 0.63626 0.61513 0.60167 0.59424
## Proportion of Variance 0.01169 0.01129 0.01125 0.01051 0.01006 0.00981
## Cumulative Proportion  0.90606 0.91735 0.92860 0.93911 0.94917 0.95898
##                          PC31   PC32    PC33    PC34    PC35    PC36
## Standard deviation     0.58683 0.5498 0.48442 0.47576 0.43757 0.42165
## Proportion of Variance 0.00957 0.0084 0.00652 0.00629 0.00532 0.00494
## Cumulative Proportion  0.96854 0.9769 0.98346 0.98974 0.99506 1.00000
```

```
loadings = pca$rotation
scores = pca$x
s = summary(pca)
plot(s$importance[2,],xlab='PC', ylab='Proportion of Variance')
```



In the first principal dimension, most of observations concentrate on the right hand side. As the top loadings are about spam/adult/gamine/college, we conclude these may be two kinds of uses: spam/porn bots that aren't excluded and undergrad gamer. As to the left hand side, with significantly fewer observations, those features do not yield clear information either. In the second principal component, we can clearly see that the higher value side is the people who love fashion/dressing/life styles and are willing to share. The lower value side of the second dimension is almost the same with the first dimension. Last but not least, the correlation of the first and second component is apparent: the higher the value in first component, the lower the variance of observations in the second dimension, centralzing at the middle (where is close to 0). Base on these facts, we consider that the second dimension is not powerful in explaining data, and is even worse when observations have higher value in first dimension, say those bots and undergrad gamers.

```
qplot(scores[,1], scores[,2], xlab='Component 1', ylab='Component 2')
```



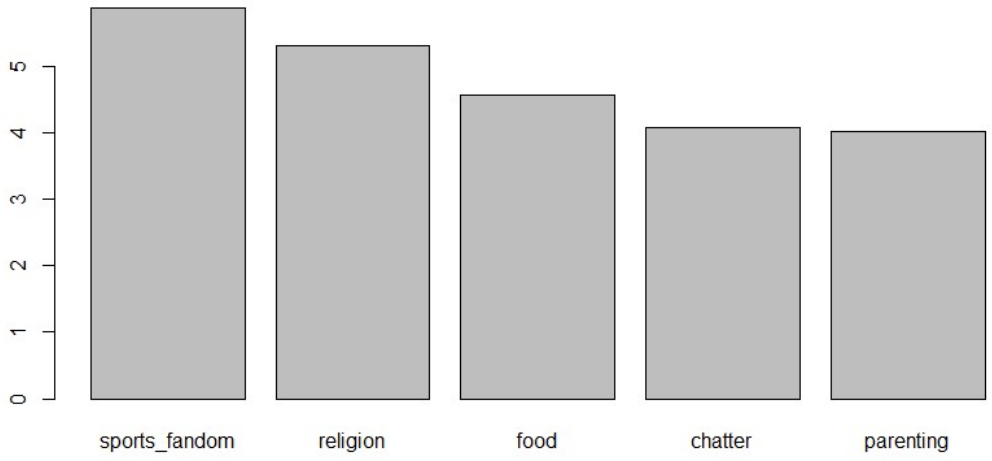

```
o1 = order(loadings[,1], decreasing=TRUE)
colnames(data_scaled)[head(o1,5)]

## [1] "spam"          "adult"         "online_gaming" "college_uni"
## [5] "uncategorized"

colnames(data_scaled)[tail(o1,5)]

## [1] "school"        "sports_fandom" "parenting"     "food"
## [5] "religion"

o2 = order(loadings[,2], decreasing=TRUE)
colnames(data_scaled)[head(o2,5)]

## [1] "cooking"       "photo_sharing" "fashion"       "shopping"
## [5] "beauty"

colnames(data_scaled)[tail(o2,5)]

## [1] "school"        "food"          "parenting"     "religion"
## [5] "sports_fandom"
```

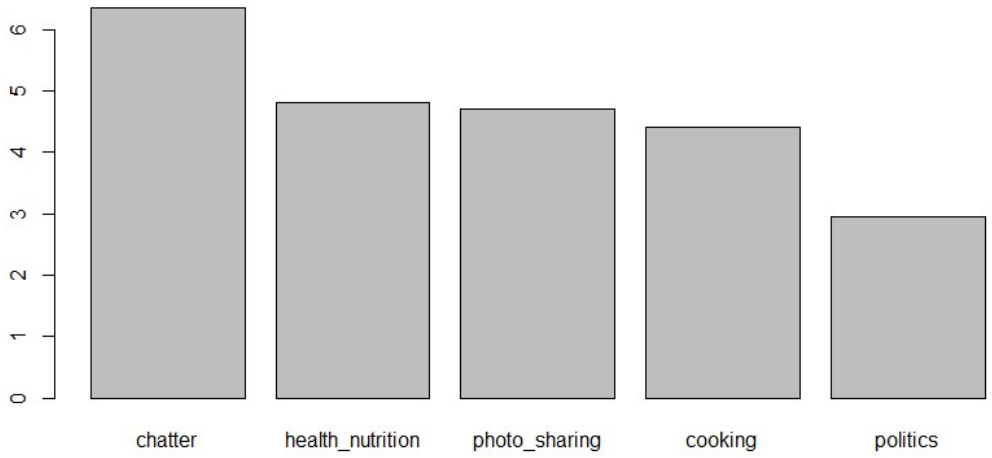
After digging through the loadings and analyzing the scatterplot of the first two principal components, we realized that PCA would most likely not be the most effective way to find interesting market clusters.

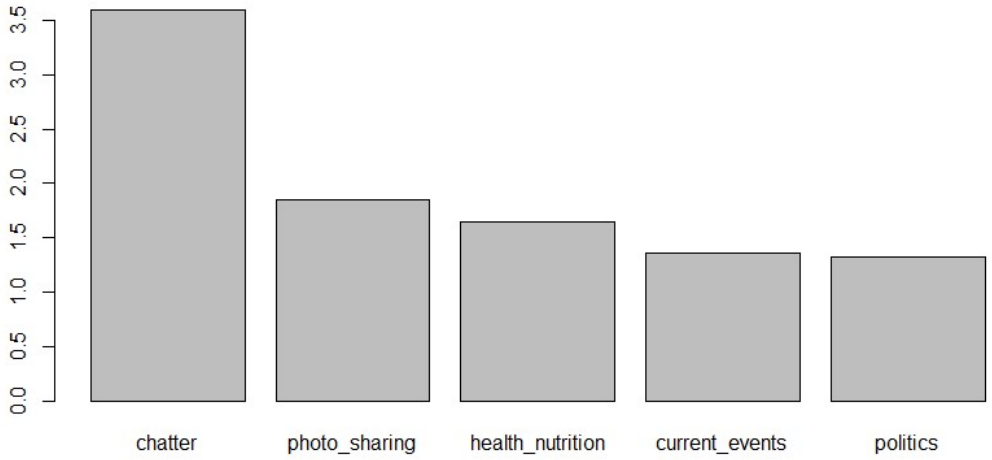
K-Menas

Next, we used k-means clustering analysis to try and group the users into different clusters. A primary struggle in using the K-means process was choosing how many clusters, K, we should create in the data.

```
set.seed(2)
data_scaled = scale(data[,-1], center = T, scale = T)

#kmeanspp modeling, k =3
clust2 = kmeanspp(data_scaled, k=3, nstart=25)
obs_ci = data.frame(matrix(ncol = 36, nrow = 3))
obs_c = data.frame()
for (i in 1:3){
  c = which(clust2$cluster == i)
  obs_c= data[c,]
  obs_ci[i,] = apply(obs_c[,-1],2,mean)
}
colnames(obs_ci)<- names(apply(obs_c[,-1],2,mean))

##number of observations in a cluster
for(i in 1:3){
  print(length(which(clust2$cluster == i)))
}

## [1] 808
## [1] 2158
## [1] 4916

#plot top 5 for each cluster
for (i in 1:3){
  plotV <- sort(obs_ci[i,], decreasing = T)[1:5]
  barplot(as.vector(as.matrix(plotV)),names.arg = names(plotV))
}
```


0
1
2
3
4
5
sports_fandom
religion
food
chatter
parenting

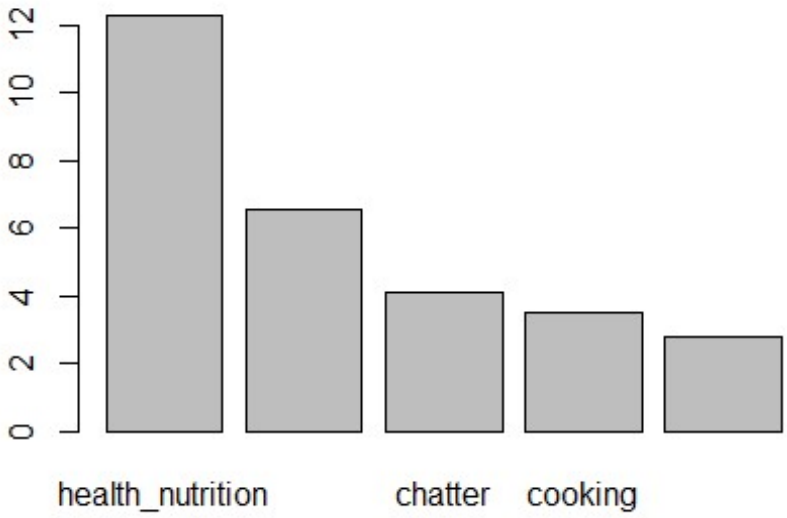

0
1
2
3
4
5
6
chatter
health_nutrition
photo_sharing
cooking
politics

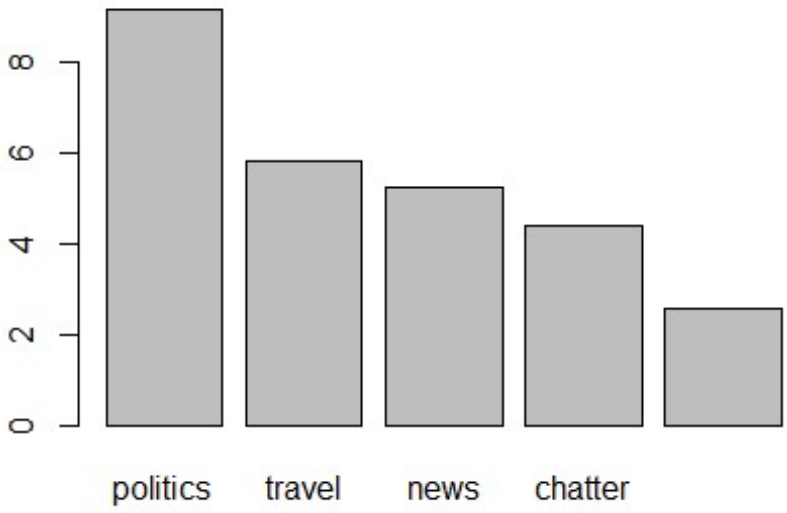


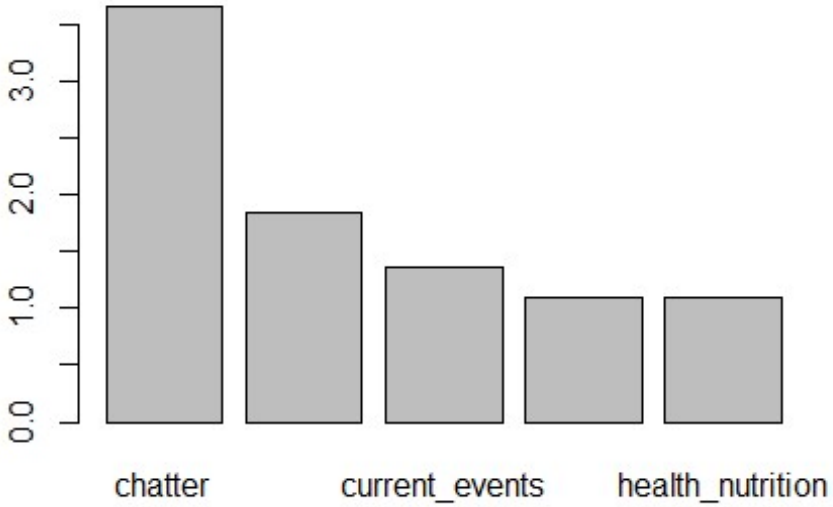
We decided to start with three clusters and work up to more clusters until we deemed the number of clusters too complicated for reasonable interpretation and analysis. When looking at three clusters, we noticed a peculiar occurrence that two of the three clusters shared the same three most common interests shown by their accounts. These two clusters we deemed to be the health-and-fitness group, as their interests included chatter, health, nutrition, and photo sharing. The third cluster in this model showed interest in sports fandom, religion, and food. As such, we have deemed this cluster the "Texans", as those three interests primarily describe the general population in the state of Texas.

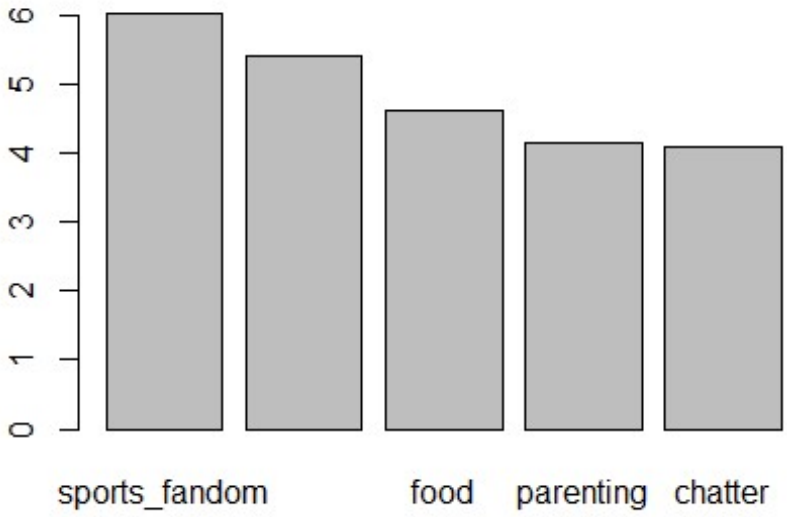
```
#kmeanspp modeling, k =5
clust2 = kmeanspp(data_scaled, k=5, nstart=25)
obs_ci = data.frame(matrix(ncol = 36, nrow = 5))
obs_c = data.frame()
for (i in 1:5){
  c = which(clust2$cluster == i)
  obs_c= data[c,]
  obs_ci[i,] = apply(obs_c[,-1],2,mean)
}
colnames(obs_ci)<- names(apply(obs_c[,-1],2,mean))

##number of observations in a cluster
for(i in 1:5){
  print(length(which(clust2$cluster == i)))
}

## [1] 870
## [1] 670
## [1] 4172
## [1] 740
## [1] 1430
```

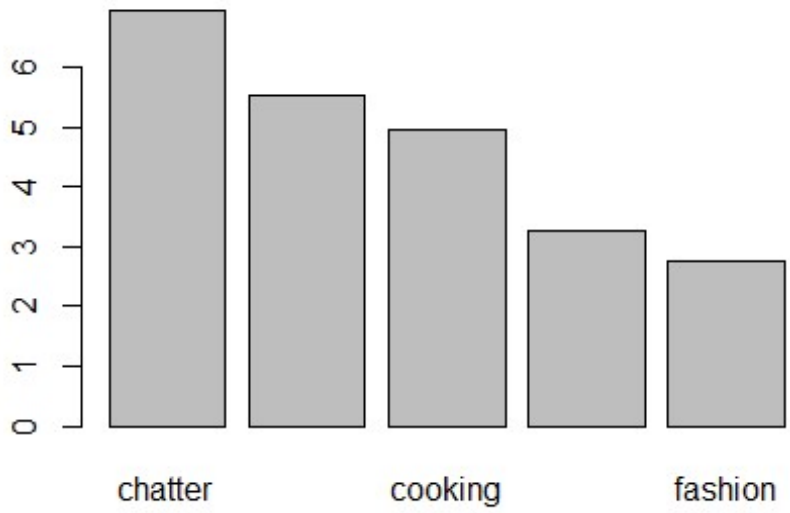
```r
#plot top 5 for each cluster
for (i in 1:5){
  plotV <- sort(obs_ci[i,], decreasing = T)[1:5]
  barplot(as.vector(as.matrix(plotV)),names.arg = names(plotV))
}
```


0
2
4
6
8
10
12
health_nutrition
chatter
cooking


0
2
4
6
8
politics
travel
news
chatter

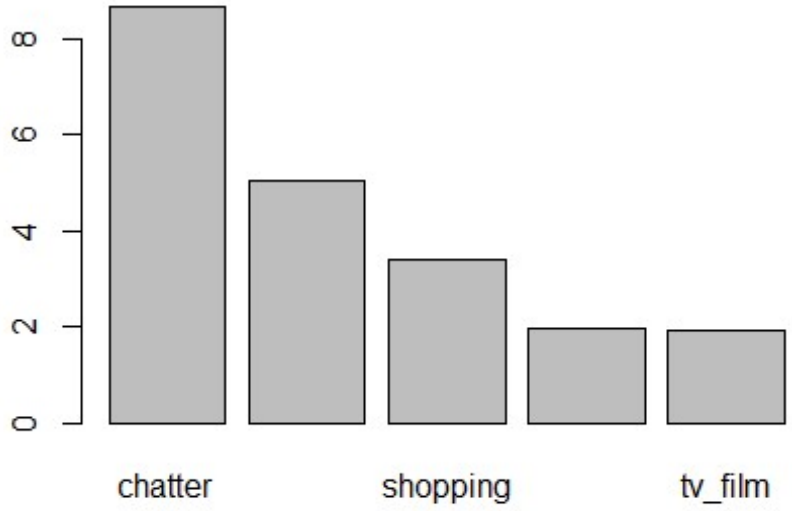

0.0
1.0
2.0
3.0
chatter
current_events
health_nutrition

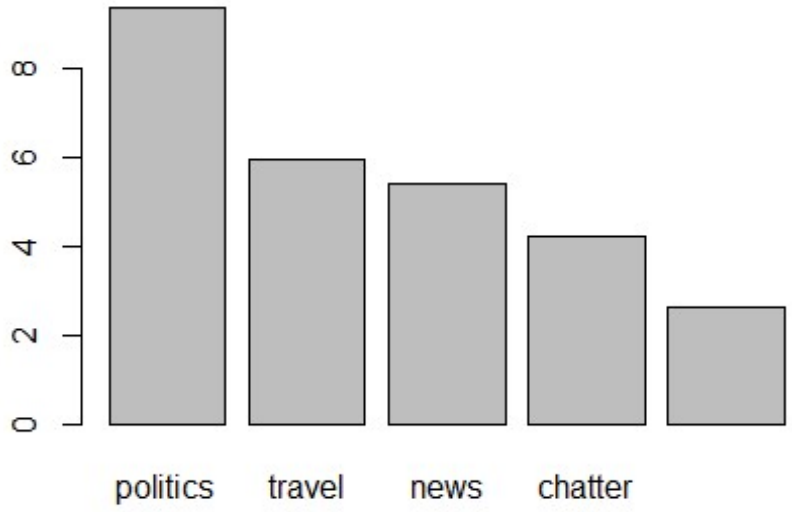

0
1
2
3
4
5
6
sports_fandom
food
parenting
chatter

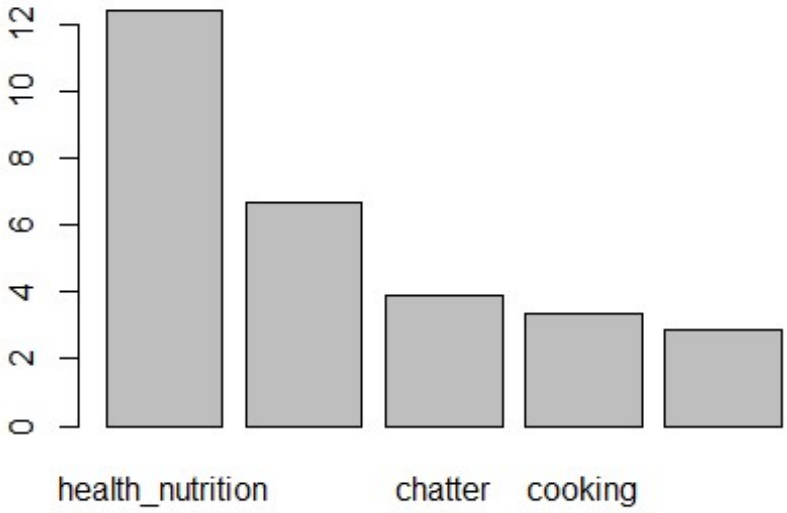


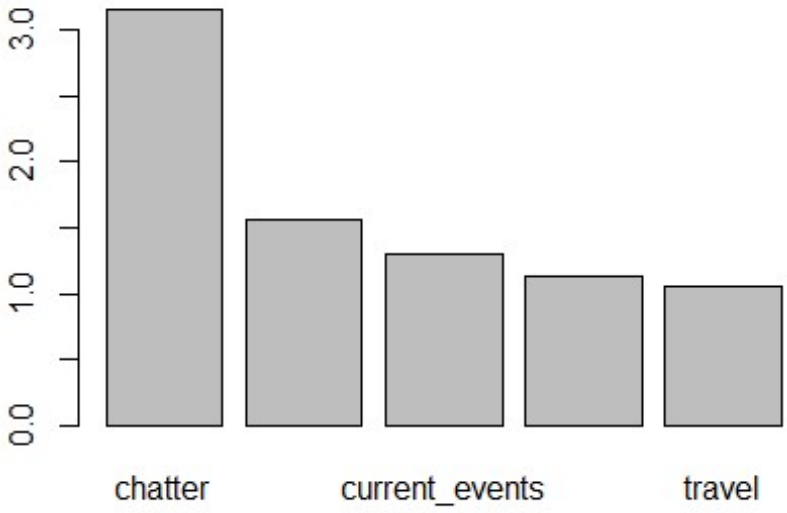

```
#kmeanspp modeling, k =7
clust2 = kmeanspp(data_scaled, k=7, nstart=25)
obs_ci = data.frame(matrix(ncol = 36, nrow = 7))
obs_c = data.frame()
for (i in 1:7){
  c = which(clust2$cluster == i)
  obs_c= data[c,]
  obs_ci[i,] = apply(obs_c[,-1],2,mean)
}
colnames(obs_ci)<- names(apply(obs_c[,-1],2,mean))

##number of observations in a cluster
for(i in 1:7){
  print(length(which(clust2$cluster == i)))
}

## [1] 1281
## [1] 618
## [1] 805
## [1] 3583
## [1] 371
## [1] 517
## [1] 707

#plot top 5 for each cluster
for (i in 1:7){
  plotV <- sort(obs_ci[i,], decreasing = T)[1:5]
```

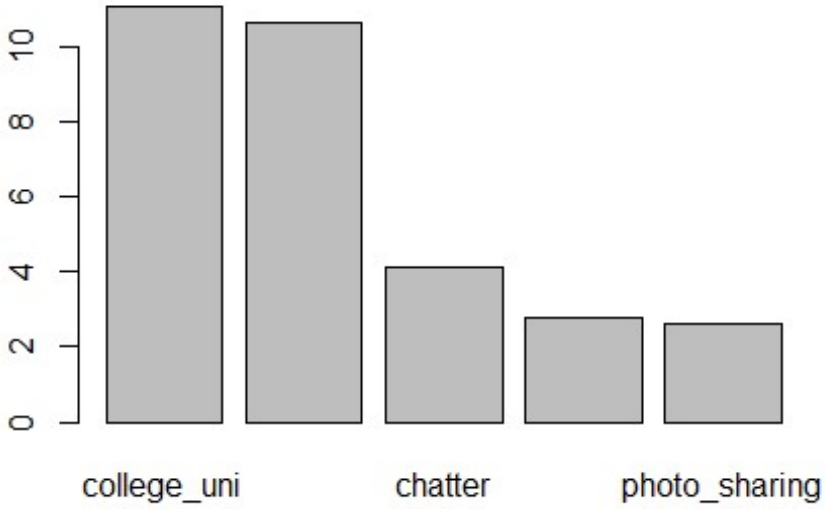
```
  barplot(as.vector(as.matrix(plotV)),names.arg = names(plotV))
}
```

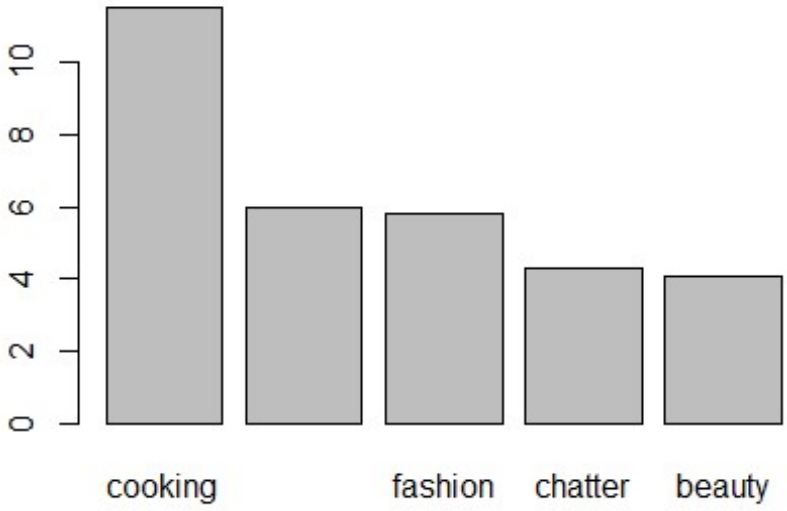


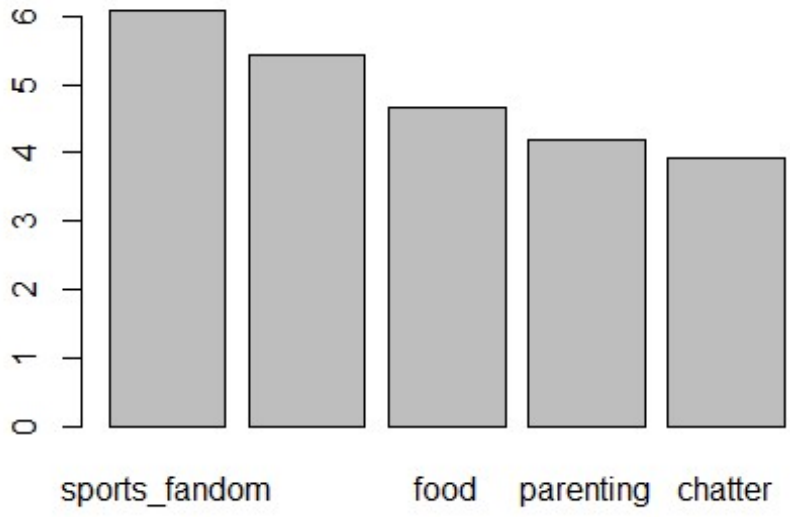



0
2
4
6
8
10
12
health_nutrition
chatter
cooking


0.0
1.0
2.0
3.0
chatter
current_events
travel


0
2
4
6
8
10
college_uni
chatter
photo_sharing


0
2
4
6
8
10
cooking
fashion
chatter
beauty

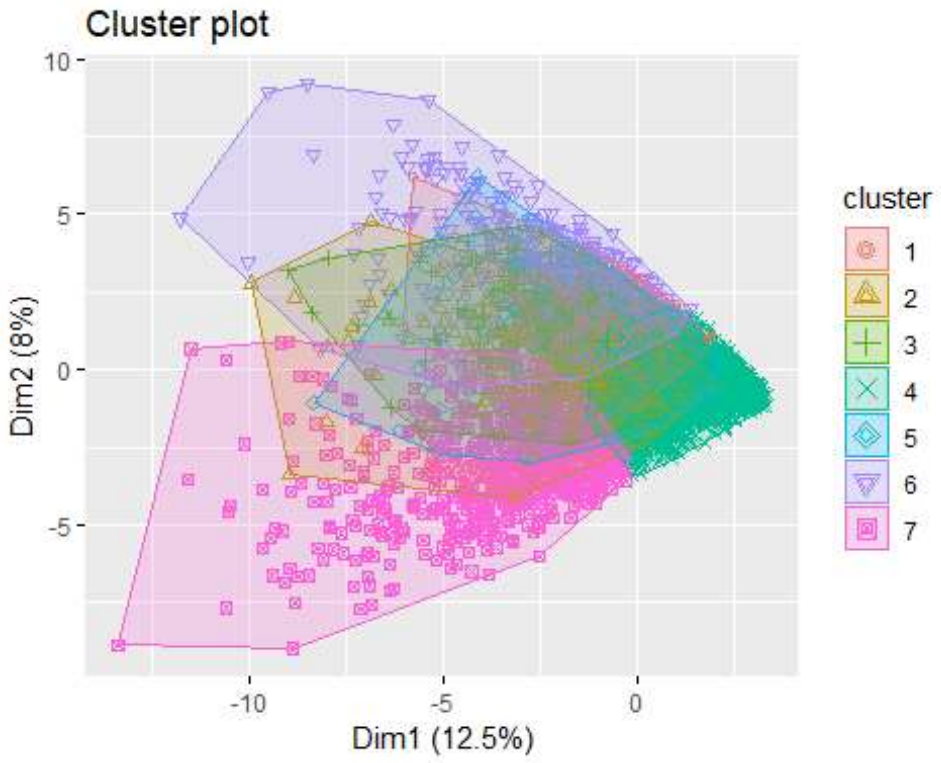


Realizing that these three clusters could not properly cover the whole dataset, we increased the number of clusters. After trying several values of K, we finally settled on seven clusters. Of these clusters, we think that "NurtientH20", a company founded on the principles of health and active lifestyles, should target "Global Citizens", a cluster focused on travel, news, and politics; "Fitness Addicts", a cluster focused on health, nutrition, and personal fitness; and "Young Homemakers", a cluster focused mainly on cooking, photo sharing, and fashion. We believe that if NutrientH20 follows our recommendations and focuses on these three specific market segments we discovered in our clustering analysis, they will increase their revenues without having to sharply increase advertising budgets. [Appendix] For further exploration to the data, we tried to find a way to plot the 7 clusters on PCA graph. And this is our findings.

```
library(factoextra)

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at
https://goo.gl/13EFCZ

library(tidyverse)

## -- Attaching packages --------------------------------- tidyverse 1.2.1 --

## v tibble  1.4.2     v purrr   0.2.5
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0

## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x purrr::accumulate() masks foreach::accumulate()
```

```
## x mosaic::count()     masks dplyr::count()
## x purrr::cross()      masks mosaic::cross()
## x mosaic::do()        masks dplyr::do()
## x tidyr::expand()     masks Matrix::expand()
## x tidyr::fill()       masks VGAM::fill()
## x dplyr::filter()     masks stats::filter()
## x xts::first()        masks dplyr::first()
## x dplyr::lag()        masks stats::lag()
## x xts::last()         masks dplyr::last()
## x mosaic::stat()      masks ggplot2::stat()
## x mosaic::tally()     masks dplyr::tally()
## x purrr::when()       masks foreach::when()
```

```
fviz_cluster(clust2, data = data_scaled,geom = "point")
```




```
options(dplyr.width = Inf)
mm<-data %>%
     mutate(Cluster = clust2$cluster) %>%
     group_by(Cluster) %>%
     summarise_all("mean")
m = as.data.frame(mm)
m = m[,3:38]

sort(m[6,],decreasing = T)
```

```
##    cooking photo_sharing  fashion  chatter   beauty health_nutrition
## 6 11.53191      6.003868 5.841393 4.276596 4.087041         2.359768
```

```
##   shopping current_events   travel college_uni politics personal_fitness
## 6 1.781431        1.77176 1.504836    1.466151 1.454545         1.400387
##   uncategorized    music sports_fandom online_gaming     news     food
## 6      1.261122 1.220503      1.187621      1.117988 1.092843 1.075435
##     tv_film    school    family      art  religion automotive  outdoors
## 6 0.9845261 0.9477756 0.9071567 0.893617 0.8723404  0.8646035 0.8394584
##   sports_playing parenting    dating computers home_and_garden    crafts
## 6      0.8104449 0.8065764 0.7485493 0.7156673       0.6112186 0.6112186
##   business       eco small_business     adult        spam
## 6 0.582205 0.5241779      0.4584139 0.4487427 0.003868472
```

```
sort(m[7,],decreasing = T)
```

```
##   sports_fandom religion     food parenting  chatter   school   family
## 7      6.077793   5.4314 4.664781  4.181047 3.925035 2.763791 2.561528
##   photo_sharing health_nutrition current_events cooking   travel shopping
## 7      2.442716         1.923621       1.683168 1.62942 1.369165 1.329562
##   personal_fitness college_uni politics   beauty   crafts     news
## 7         1.243281    1.196605  1.15983 1.125884 1.104668 1.062235
##    tv_film automotive  fashion online_gaming       art sports_playing
## 7 1.053748   1.043847 1.018388      1.014144 0.8670438      0.7454031
##   uncategorized computers    dating     music  outdoors home_and_garden
## 7     0.7411598 0.7411598 0.7270156 0.7213579 0.6958982       0.6492221
##         eco  business     adult small_business        spam
## 7 0.6449788 0.4879774 0.4200849      0.3917963 0.005657709
```

From this graph, we could know more about the PC. Although PC1 cannot tell us any story, PC2 seperates cluster 6 and cluster 7 pretty clearly. Looking into cluster 6, we found that they are the group of people who like cooking, photo sharing and fashion, and we called them “Young Homemakers”. Group 7 happened to be the “Texans”. Thus, we could say people with high PC2 scores might be a “Young Homemakers” and people with low PC2 scores might be a “Texans” in our clusters.