STA 380 Homework 1

Henry Chang, Tiffany Sung, Joseph Chin

8/8/2018

Problem 1 Probability Practice

Part A

Given the following: Prob(RC) = 0.3 Prob(Yes|RC) = Prob(No|RC) = 0.5 Prob(TC) = 0.7 Prob(Yes) = 0.65 Prob(No) = 0.35

We want to know Prob(Yes|TC) = ?

Solution: Prob(Yes) = Prob(Yes|RC) ∗ Prob(RC) + Prob(Yes|TC) ∗ Prob(TC) 0.65 = 0.5 ∗ 0.3 + Prob(Yes|TC) ∗ 0.7

```
(0.65-0.3*0.5) / 0.7
## [1] 0.7142857
```

Part B

The probability of having disease given test positive: D: having disease, ND: not having disease P: getting postivie in the test, N: getting negative in the test

P(D|P) = P(P|D) ∗ P(D) / (P(P|D) ∗ P(D) + P(P|N) ∗ P(N))

```
(0.993) * 0.000025 / ((0.993) * 0.000025 + 0.0001 * 0.999975)
## [1] 0.1988824
```

Problem of the test lies in having too many “False Positive”. Which is giving too many positive on tests for those who don’t have a desease. This kind of implementing a universal testing policy for the disease will lead to panic and chaos.

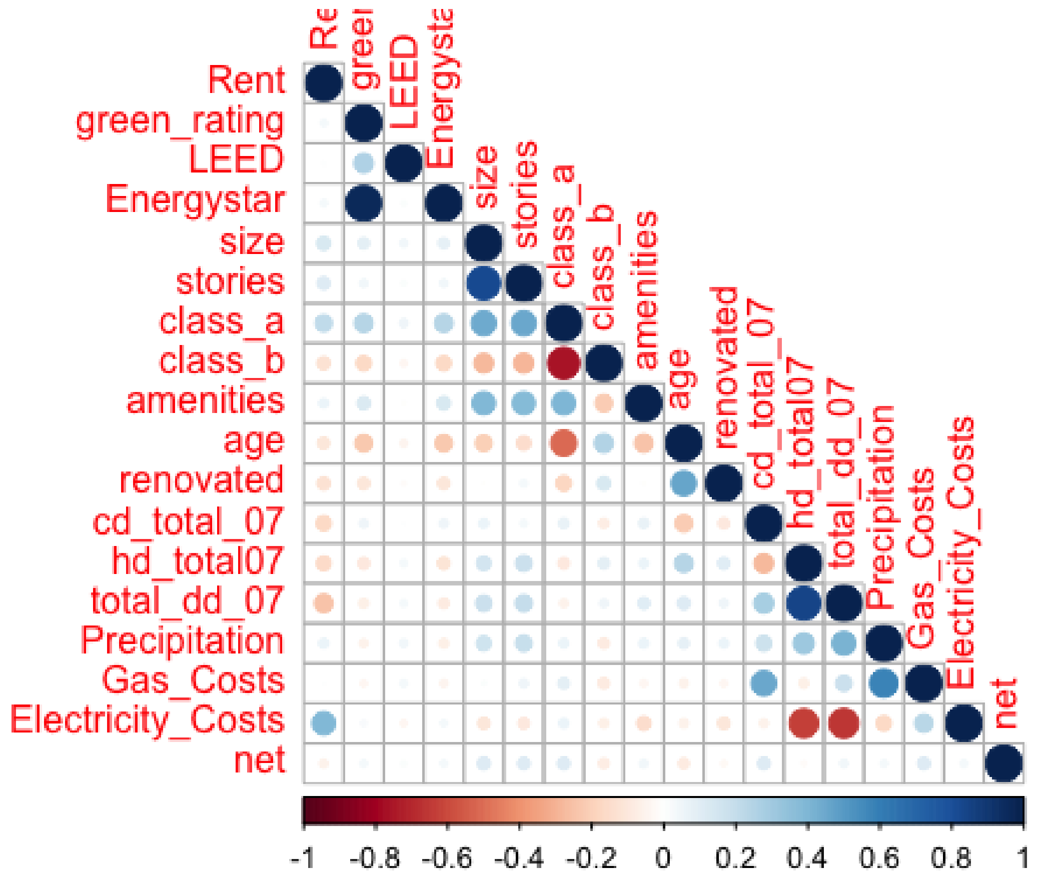
Problem 2 Green Buildings

```
library(corrplot)
## corrplot 0.84 loaded
library(ggplot2)
plot_data =
read.csv(url("https://raw.githubusercontent.com/jgscott/STA380/master/data/gr
eenbuildings.csv"))
#colnames(plot_data)
re_order_var = c("Rent", "green_rating", "LEED","Energystar", "size",
"stories", "class_a",
 "class_b", "amenities",  "age", "renovated", "cd_total_07",
"hd_total07","total_dd_07", "Precipitation","Gas_Costs" ,
"Electricity_Costs", "net")
```

After reading our data into R. I reordered the variables by their categories to make correlation plots easier to read. Our new variable list starts with rent, green building certificate, characteristics of a building, weather indicators, ends with energy and other cost.

Variable correlation plot

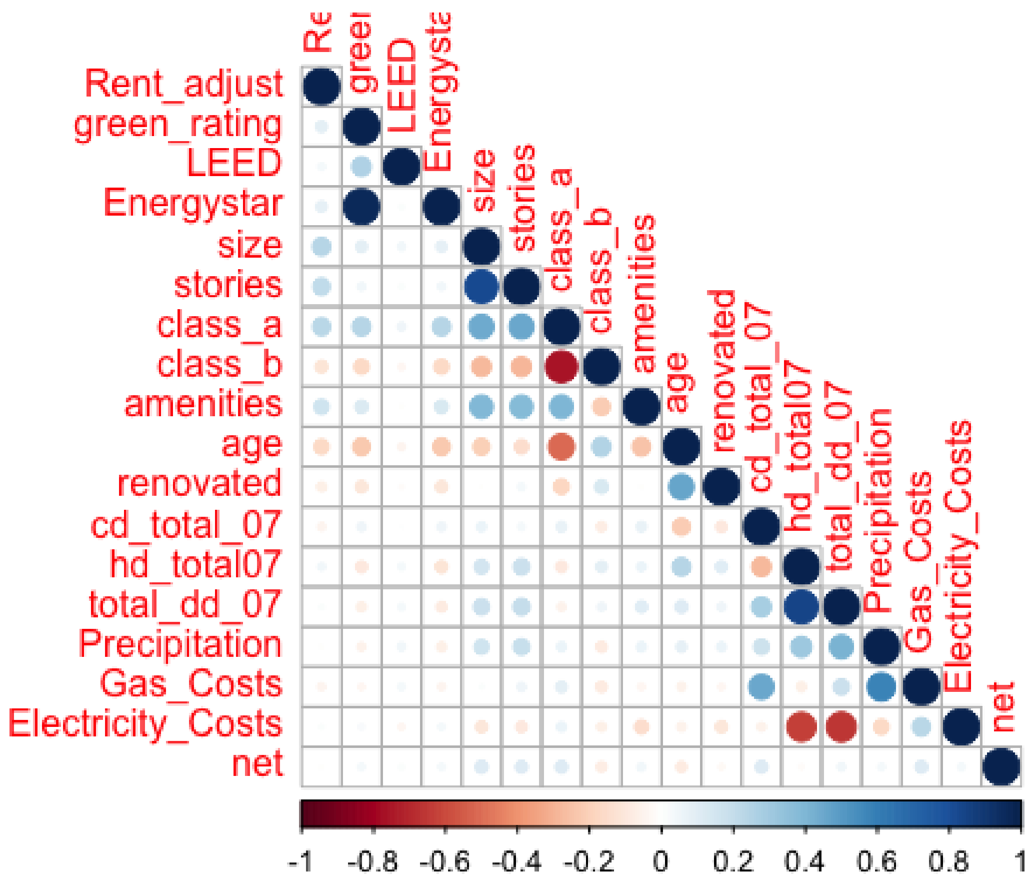
```
corrplot(cor(plot_data[re_order_var]), type = "lower")
```



The correlations seems reasonable: rent postiviely correlated to good characteristics of a building and negatively correlated to worse building conditions and less favorable weather conditions. However, “cluster_rent” provides aggregated information by cluster. Which cannot be included in our correlation plot but provides valuable information.

Therefore, we created varaiable “Rent_adjust” by dividing rent using its cluster’s rent. Which tells us the house’s rent proportional to the cluster’s rent.

```
plot_data["Rent_adjust"] = plot_data["Rent"] / plot_data["cluster_rent"]
corrplot(cor(plot_data[c("Rent_adjust", re_order_var[-1])]), type = "lower")
```



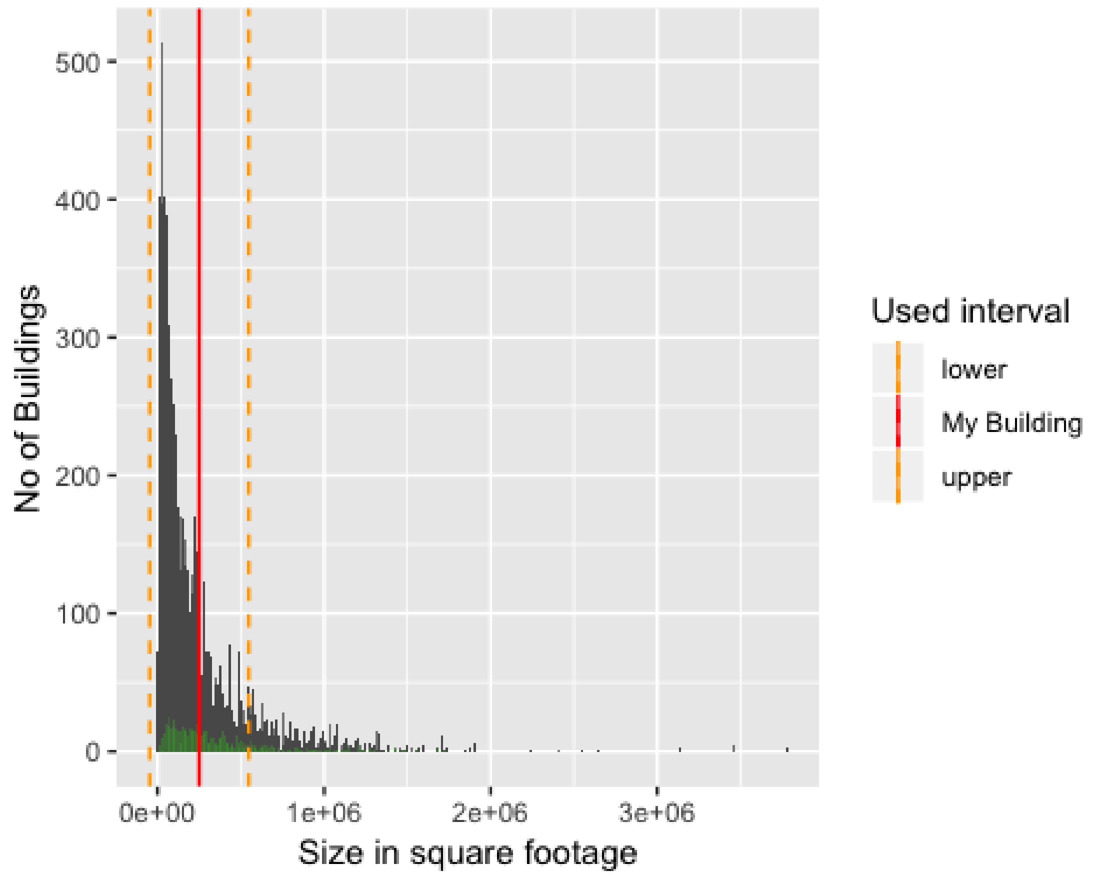
After replacing Rent using “Rent_adjust”, we found that correlations between “Rent_adjust” and weather related variables has dissappeared. Which seems to tell us that the effect of weather on rent are reflected on the cluster’s rent.

Examine rent difference between green and non-green buildings having similar characteristic with our new building.

After we have studied our variables, we look into information we have on our new building : size, age, and stories to help us decide whether we should build a green building.

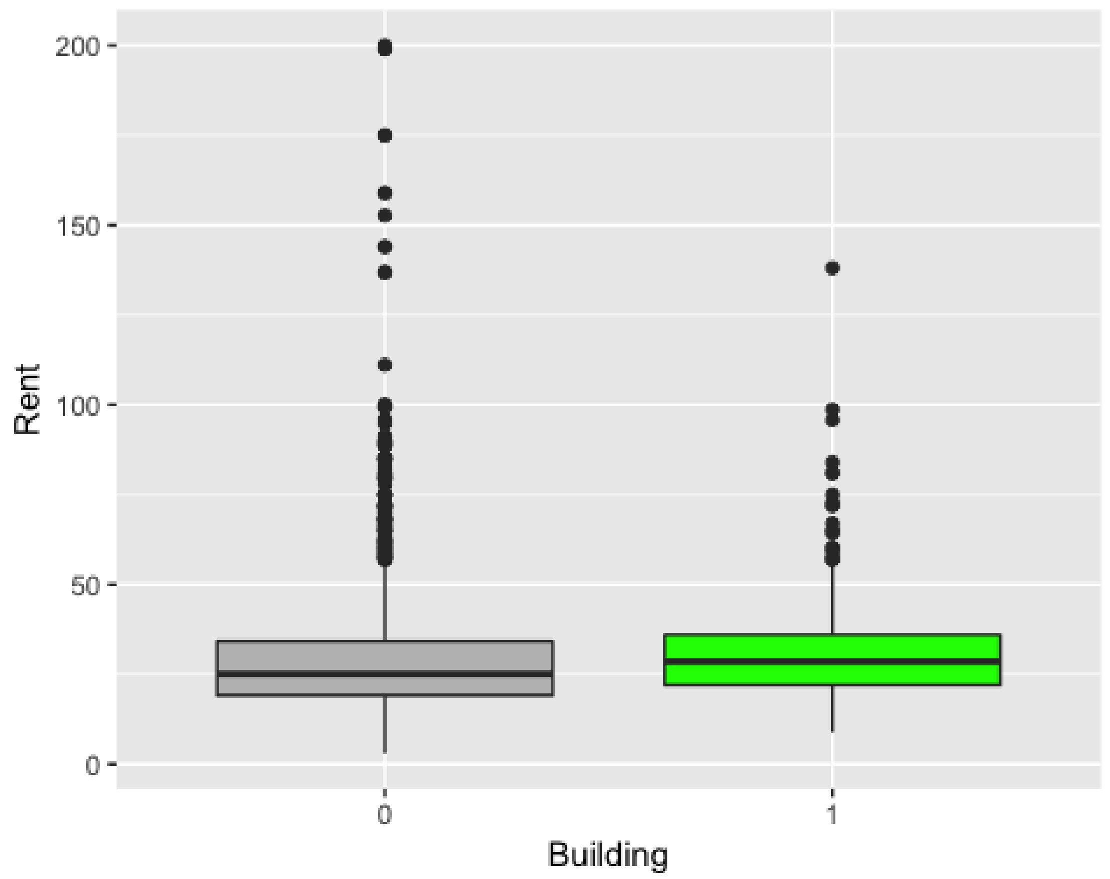
Size

```
size_sd = sd(plot_data$size)
ggplot(plot_data, aes(x = size)) +
  geom_histogram(binwidth = 10000) +
  geom_histogram(data=subset(plot_data, green_rating == "1"),binwidth =
10000,fill = "green", alpha = 0.2) +
  labs( x = "Size in square footage", y ="No of Buildings" ) +
  geom_vline(aes(xintercept = 250000, color = "My Building")) +
  geom_vline(aes(xintercept = 250000 + size_sd, color = "upper") , linetype =
"dashed") +
  geom_vline(aes(xintercept = 250000 - size_sd, color = "lower"), linetype =
"dashed") +
  scale_color_manual(name="Used interval", values=c(`My Building` = "red",
upper = "orange", lower = "orange"))
```



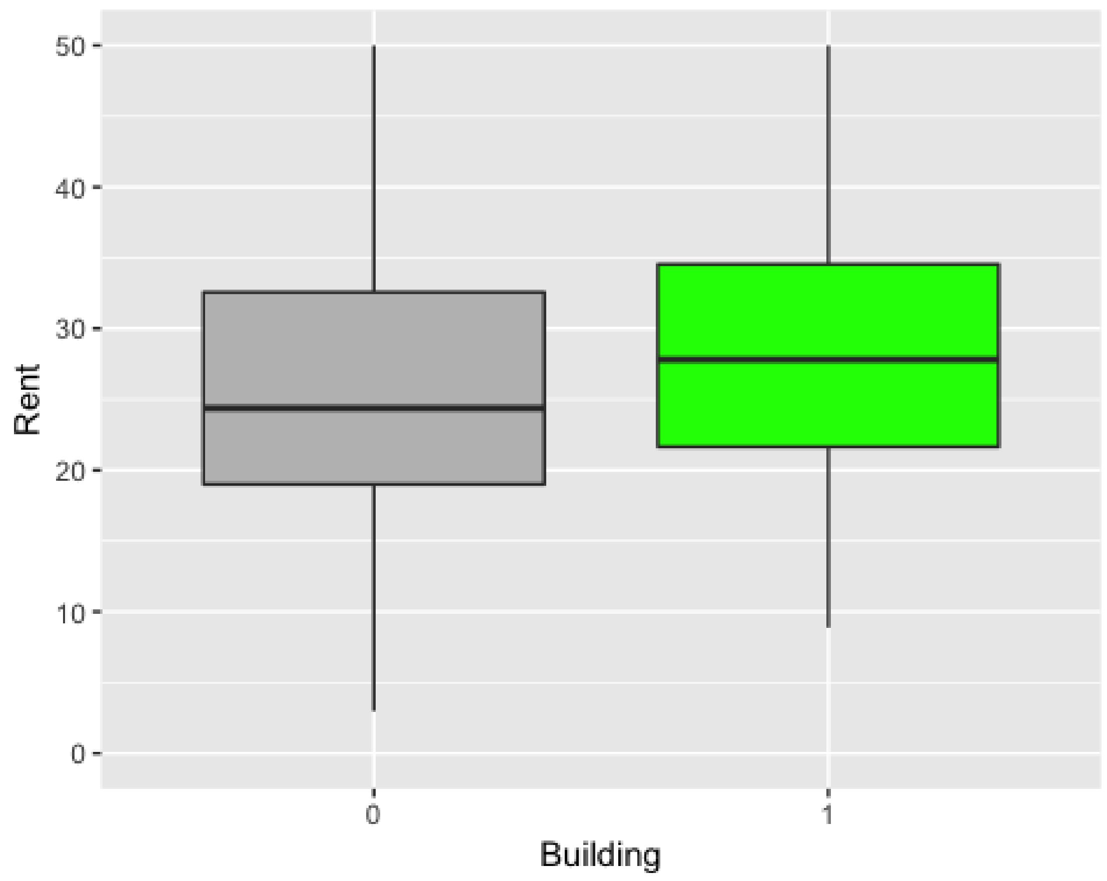
We plot the “size” variable’s distribution of our entire data along with the green building’s size distribution colored in green. The vertical red line shows the our building’s size : 250000. And the orange dash line shows interval of one standard deviation, in which are the data points we are carrying to our boxplot.

```
size_explore = plot_data[plot_data$size <=  250000 + size_sd, ]
size_explore = size_explore[size_explore$size >= 250000 - size_sd, ]
size_explore$green_rating = as.factor(size_explore$green_rating)
ggplot(size_explore, aes(x = green_rating, y = Rent) ) +
  geom_boxplot(fill = c("grey", "green")) +
  #scale_y_continuous(limit = c(0, 50)) +
  labs(x = "Building", y = "Rent")
```



Using the data points having less than one standard deviation on “size” with our new building, we plotted the rent difference between green and non-green buildings.

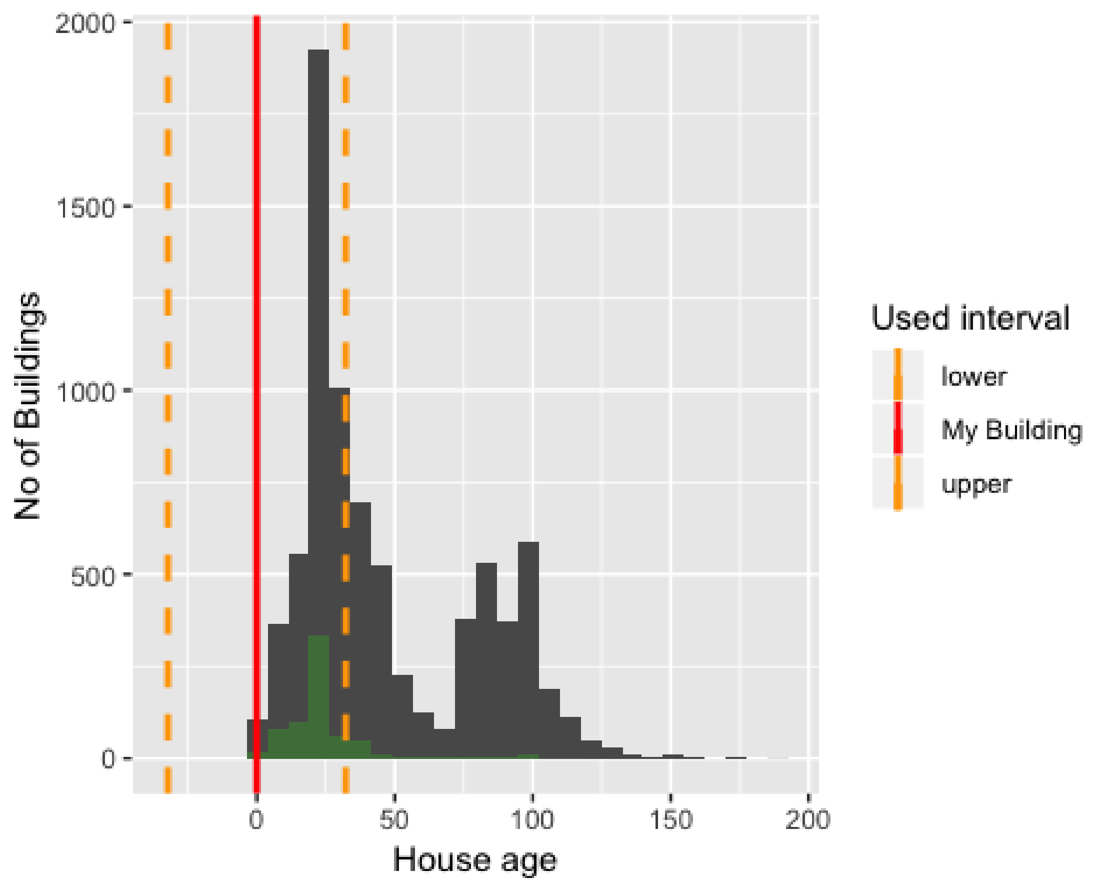
```
size_explore = plot_data[plot_data$size <=  250000 + size_sd, ]
size_explore = size_explore[size_explore$size >= 250000 - size_sd, ]
size_explore$green_rating = as.factor(size_explore$green_rating)
ggplot(size_explore, aes(x = green_rating, y = Rent) ) +
  geom_boxplot(fill = c("grey", "green")) +
  scale_y_continuous(limit = c(0, 50)) +  # 363 among 6998 have rent > 50
  labs(x = "Building", y = "Rent")
```



We zoom in our plot by resetting Y limit. We found that 25, 50, 75 rent quantile of green buildings are all higher than non-green buildings.

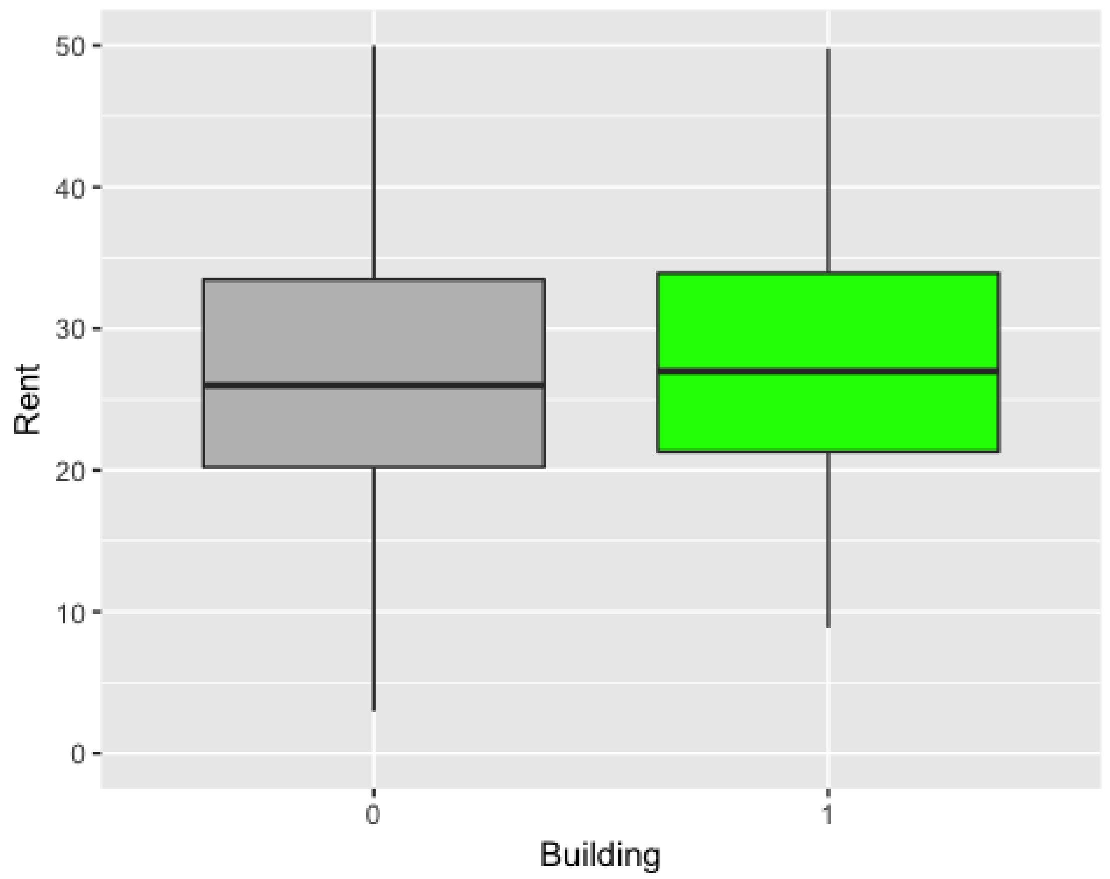
Age

```r
age_sd = sd(plot_data$age)
ggplot(plot_data, aes(x = age)) +
  geom_histogram() +
  geom_histogram(data=subset(plot_data, green_rating == "1"),fill = "green",
alpha = 0.2) +
  labs( x = "House age", y ="No of Buildings" ) +
  geom_vline(aes(xintercept = 0, color = "My Building"), size = 1) +
  geom_vline(aes(xintercept = 0 + age_sd, color = "upper") , linetype =
"dashed", size = 1) +
  geom_vline(aes(xintercept = 0 - age_sd, color = "lower"), linetype =
"dashed", size = 1) +
  scale_color_manual(name="Used interval", values=c(`My Building` = "red",
                                                    upper = "orange",
                                                    lower = "orange"))
```



We plot the “age” variable’s distribution along with “age”’s distribution among green buildings. We found new buildings (age = 0) only consist of a small portion of our data.

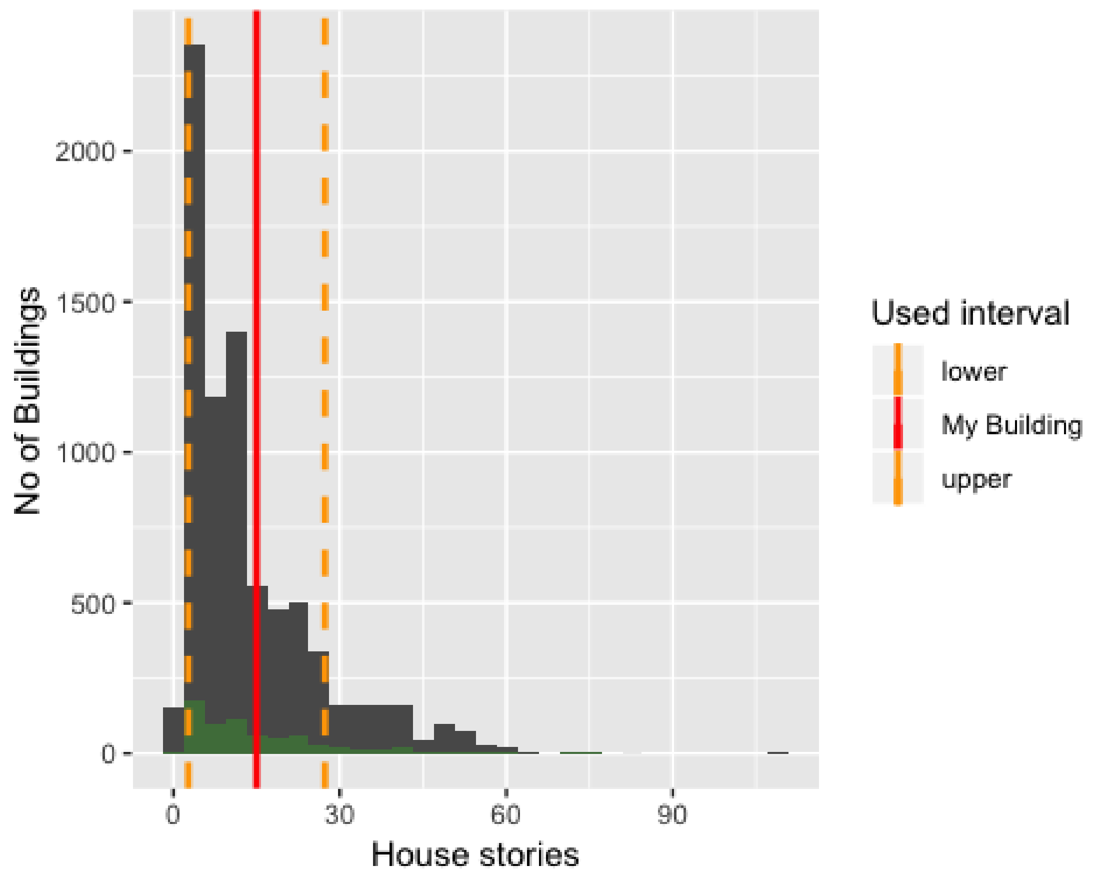
```
age_explore = plot_data[plot_data$age <=  age_sd, ]
age_explore$green_rating = as.factor(age_explore$green_rating)
ggplot(age_explore, aes(x = green_rating, y = Rent)) +
      geom_boxplot(fill = c("grey", "green")) +
      scale_y_continuous(limit = c(0, 50)) + #173 among 3740 have rent > 50
      labs(x = "Building", y = "Rent")
```



Green buildings have higher rents in relatively new buildings.

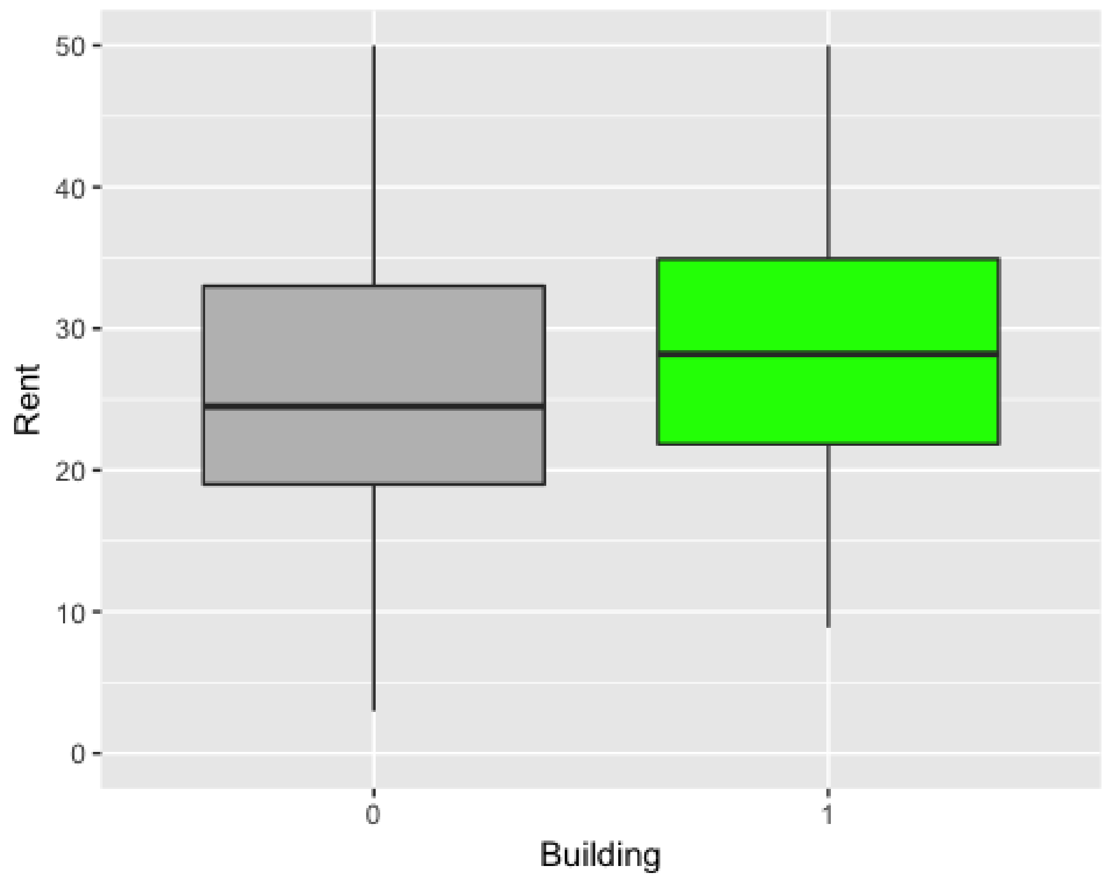
Stories

```
stories_sd = sd(plot_data$stories)
ggplot(plot_data, aes(x = stories)) +
  geom_histogram() +
  geom_histogram(data=subset(plot_data, green_rating == "1"),fill = "green",
alpha = 0.2) +
  labs( x = "House stories", y ="No of Buildings" ) +
  geom_vline(aes(xintercept = 15, color = "My Building"), size = 1) +
  geom_vline(aes(xintercept = 15 + stories_sd, color = "upper") , linetype =
"dashed", size = 1) +
  geom_vline(aes(xintercept = 15 - stories_sd, color = "lower"), linetype =
"dashed", size = 1) +
  scale_color_manual(name="Used interval", values=c(`My Building` = "red",
                                                    upper = "orange",
                                                    lower = "orange"))
```



Under the distribution of “stories”, we can see that our new buildings story is around 70 quantile among our data set.

```
stories_explore = plot_data[plot_data$stories <= 15 + stories_sd, ]
stories_explore = stories_explore[stories_explore$stories >= 15 - stories_sd,
]
stories_explore$green_rating = as.factor(stories_explore$green_rating)
ggplot(stories_explore, aes(x = green_rating, y = Rent)) +
  geom_boxplot(fill = c("grey", "green")) +
  scale_y_continuous(limit = c(0, 50)) +  #173 among 3740 have rent > 50
  labs(x = "Building", y = "Rent")
```



Green buildings can bring higher rent in our stories interval.

Validation using regression

On plots above we obsereved higher rents from green buildings under different condition controled. Before coming up with our final advice, it's always good to check the result of linear regression, which is good at controling variables we consider might affect our rent.

We used Rent_adjust as dependent variable to control the effect from clusters. Then added class_a to control the raised rent from high-quality buildings .

```
summary(lm( Rent_adjust ~ log(size) + age + stories+ class_a +green_rating ,
            data = plot_data))

##
## Call:
## lm(formula = Rent_adjust ~ log(size) + age + stories + class_a +
##     green_rating, data = plot_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9002 -0.1413 -0.0251  0.0899  3.4442
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.6550154  0.0454151  14.423  < 2e-16 ***
## log(size)     0.0307891  0.0042013   7.328 2.56e-13 ***
## age          -0.0005894  0.0001151  -5.121 3.11e-07 ***
## stories       0.0014500  0.0004035   3.594 0.000328 ***
## class_a       0.0528742  0.0088073   6.003 2.02e-09 ***
## green_rating  0.0333510  0.0117117   2.848 0.004416 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2797 on 7888 degrees of freedom
## Multiple R-squared:  0.08143,    Adjusted R-squared:  0.08085
## F-statistic: 139.9 on 5 and 7888 DF,  p-value: < 2.2e-16
```

Green building provides 3.3% increase on local cluster rent per square foot. (Our building 250,000 square foot, extra cost for green: 5,000,000)

Problem 3 Bootstrapping

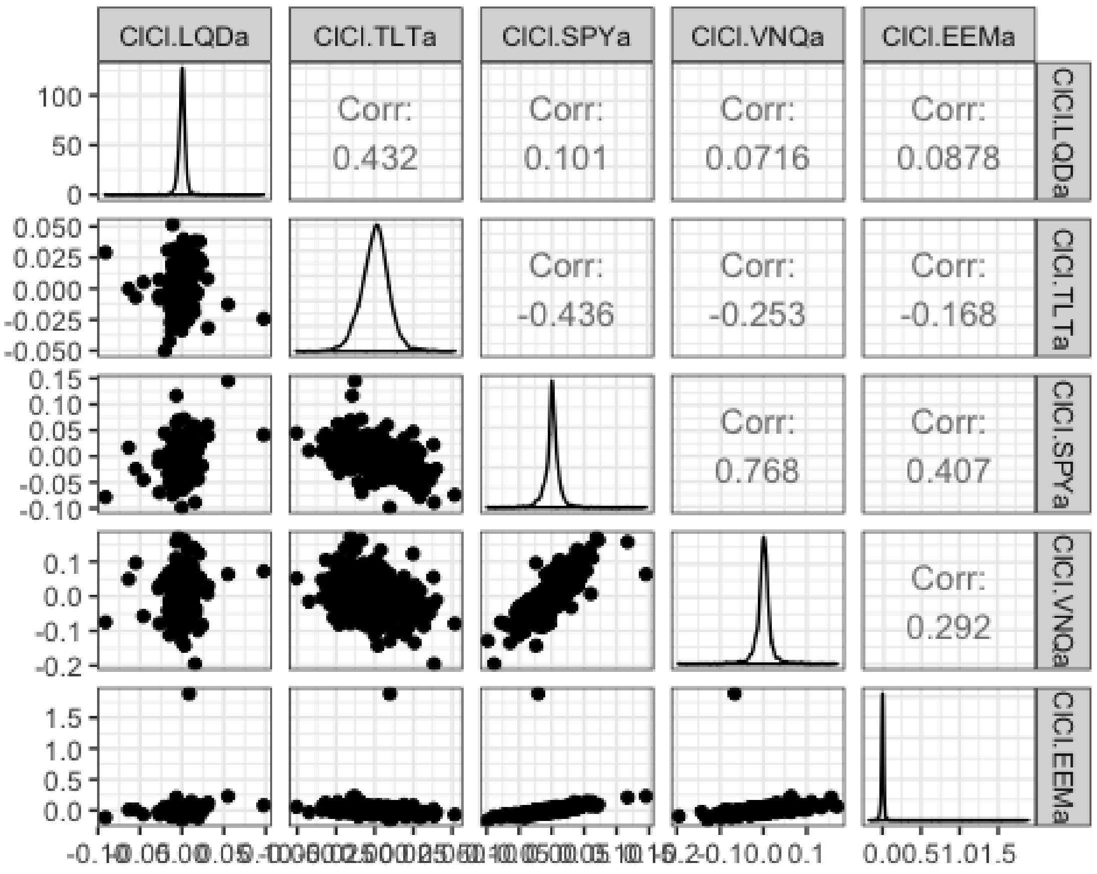
```
library(mosaic)
library(quantmod)
library(foreach)
library(GGally)

# Import the five ETFs
mystocks = c("SPY", "TLT", "LQD", "EEM", "VNQ")
myprices = getSymbols(mystocks, from = "2007-01-01")
getSymbols(mystocks)
## [1] "SPY" "TLT" "LQD" "EEM" "VNQ"

#Get the adjusted closing price of each ETF
for(ticker in mystocks) {

  expr = paste0(ticker, "a = adjustOHLC(", ticker, ")")
  eval(parse(text=expr))
}

# Combine all the returns in a matrix
all_returns = cbind(ClCl(LQDa),ClCl(TLTa),ClCl(SPYa), ClCl(VNQa), ClCl(EEMa))
all_returns = as.matrix(na.omit(all_returns))

head(all_returns)
##                 ClCl.LQDa    ClCl.TLTa     ClCl.SPYa    ClCl.VNQa
## 2007-01-04  0.0075152938  0.006063328  0.0021221123  0.001296655
## 2007-01-05 -0.0006526807 -0.004352668 -0.0079763183 -0.018518518
## 2007-01-08 -0.0002798843  0.001793566  0.0046250821  0.001451392
## 2007-01-09  0.0001866169  0.000000000 -0.0008498831  0.012648208
## 2007-01-10 -0.0013063264 -0.004475797  0.0033315799  0.012880523
## 2007-01-11 -0.0030832664 -0.005844712  0.0043804651  0.012973706
##                ClCl.EEMa
## 2007-01-04 -0.013809353
## 2007-01-05 -0.029238205
## 2007-01-08  0.007257535
## 2007-01-09 -0.022336235
## 2007-01-10 -0.002303160
## 2007-01-11  0.012650112
```

```
# The sample correlation matrix
cor(all_returns)
```

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

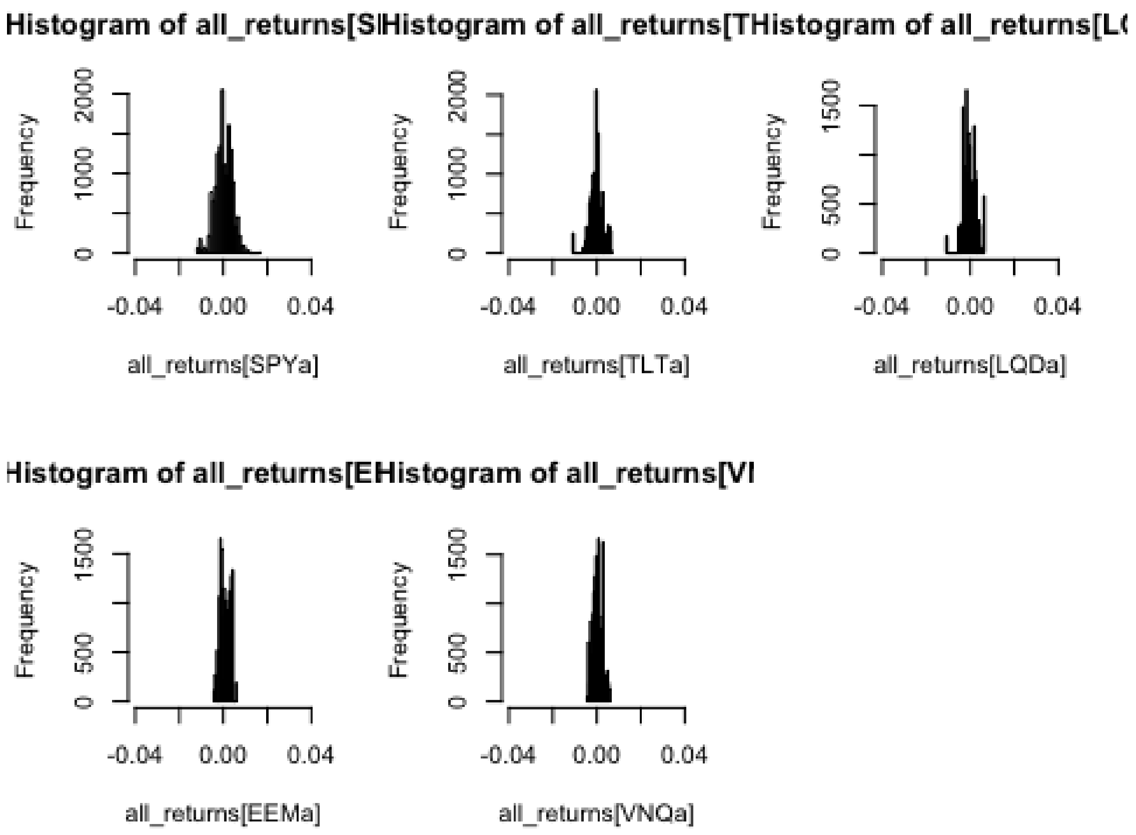
From the correlation plot, we can see that SPY is highly correlated with VNQ and EEM which indicates that emerging-market and real estate relatively more dominant by the market condition. Also, we noticed that TLT is negatively correlated with SPY. This can be explained as when the stock market goes up, people tend to sell treasury bonds to buy stocks, and when it goes down, people buy treasury bonds because they are safe.

```
ggpairs(as.data.frame(all_returns))+theme_bw()
```



From the mean and variance of each ETFs, we can see that EEM, VNQ have relatively higher return, but higher risk consequently. Thus, we can then assume EEM and VNQ is a risker investment compare to the other ETFs. This is also very intuitive because investing in bonds is less risky than investing in emerging-market or real estate which are both highly volatile.

```
#The mean and variance
sort(apply(all_returns,2,mean),decreasing = TRUE)

##    ClCl.EEMa    ClCl.VNQa    ClCl.SPYa    ClCl.TLTa    ClCl.LQDa
## 0.0009813682 0.0004157265 0.0003981241 0.0002787785 0.0002095494

cat("\n")

sort(apply(all_returns,2,var),decreasing = TRUE)

##    ClCl.EEMa    ClCl.VNQa    ClCl.SPYa    ClCl.TLTa    ClCl.LQDa
## 1.616072e-03 4.467609e-04 1.546740e-04 8.369447e-05 2.718333e-05

par(mfrow=c(2,3))
hist(all_returns[SPYa], 30, xlim=c(-0.04, 0.04))
hist(all_returns[TLTa], 30, xlim=c(-0.04, 0.04))
hist(all_returns[LQDa], 30, xlim=c(-0.04, 0.04))
hist(all_returns[EEMa], 30, xlim=c(-0.04, 0.04))
hist(all_returns[VNQa], 30, xlim=c(-0.04, 0.04))
```



Evenly Weighted Portfolio First, we construct a portfolio which the initial wealth is evenly distributed into each ETFs.

```
set.seed(1)
initial_wealth = 100000
sim_even_split = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0.2, 0.2, 0.2, 0.2, 0.2) #seperate evenly
  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 #p+r(stock)
    total_wealth = sum(holdings)#p+r(portfolio)
    wealthtracker[today] = total_wealth
    holdings = total_wealth * weights #daily rebalance
  }
  wealthtracker
}
```

With an even weighted portfolio, we have a 0.9% return and 20-day 5% VaR at -6300.89

```
even_mean <- mean(sim_even_split[,n_days])
even_return <- (mean(sim_even_split[,n_days])-initial_wealth)/initial_wealth
even_5_perc <- quantile(sim_even_split[,n_days], 0.05) - initial_wealth

# Print out results
cat("For an even split investment of $100,000,\nExpected holdings:",
even_mean, "\nReturn Rate:", even_return, "\n5% VaR:", even_5_perc)
```

```
## For an even split investment of $100,000,
## Expected holdings: 101009.1
## Return Rate: 0.01009115
## 5% VaR: -5941.083
```

Safer Portfolio Then, according to the mean and variance analysis of each ETFs, we want to construct a more risk-adverse portfolio by weighted toward the less-volatile ETFs for instance, SPY,TLT and LQD. We decided not to put any weight on EEM and VNQ because accoding to the correlation plot, these two ETF has relatively higher correlation with SPY which is the market index. If we put them in the portfolio, it would not diverse and reduce the risk.

```
set.seed(1)
initial_wealth = 100000
sim_safe_split = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0.5, 0.3, 0.2, 0, 0) # 0.2 SPY, 0.3 TLT, 0.5 LQD
  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
    holdings = total_wealth * weights
  }
  wealthtracker
}
```

With a safer portfolio, we have a lower 20-day 5% VaR at -3002.014, however, this also means a lower return at 0.5%.

```
safe_mean <- mean(sim_safe_split[,n_days])
safe_return <- (mean(sim_safe_split[,n_days])-initial_wealth)/initial_wealth
safe_5_perc <- quantile(sim_safe_split[,n_days], 0.05) - initial_wealth

# Print out results
cat("For an safer split investment of $100,000,\nExpected
holdings:",safe_mean, "\nReturn Rate:", safe_return, "\n5% VaR:",
safe_5_perc)
```

```
## For an safer split investment of $100,000,
## Expected holdings: 100544.5
## Return Rate: 0.005445058
## 5% VaR: -2784.43
```

Aggressive Portfolio For the aggressive portfolio, we merely place our initial wealth on the risker investment EEM and VNQ. We did not place any holdings on the other ETFs because if we added them in the portfolio, it would diversify the risk and make the portfolio less aggressive.

```
set.seed(1)
initial_wealth = 100000
sim_risky_split = foreach(i=1:5000, .combine='rbind') %do% {
  total_wealth = initial_wealth
  weights = c(0, 0, 0, 0.3, 0.7) #0.7 EEM, 0.3 VNQ
  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
    holdings = total_wealth * weights
  }
  wealthtracker
}
```

With an aggressive portfolio, we got a 1.65% return but follow by a really high 20-days 5% VaR at -12596.79

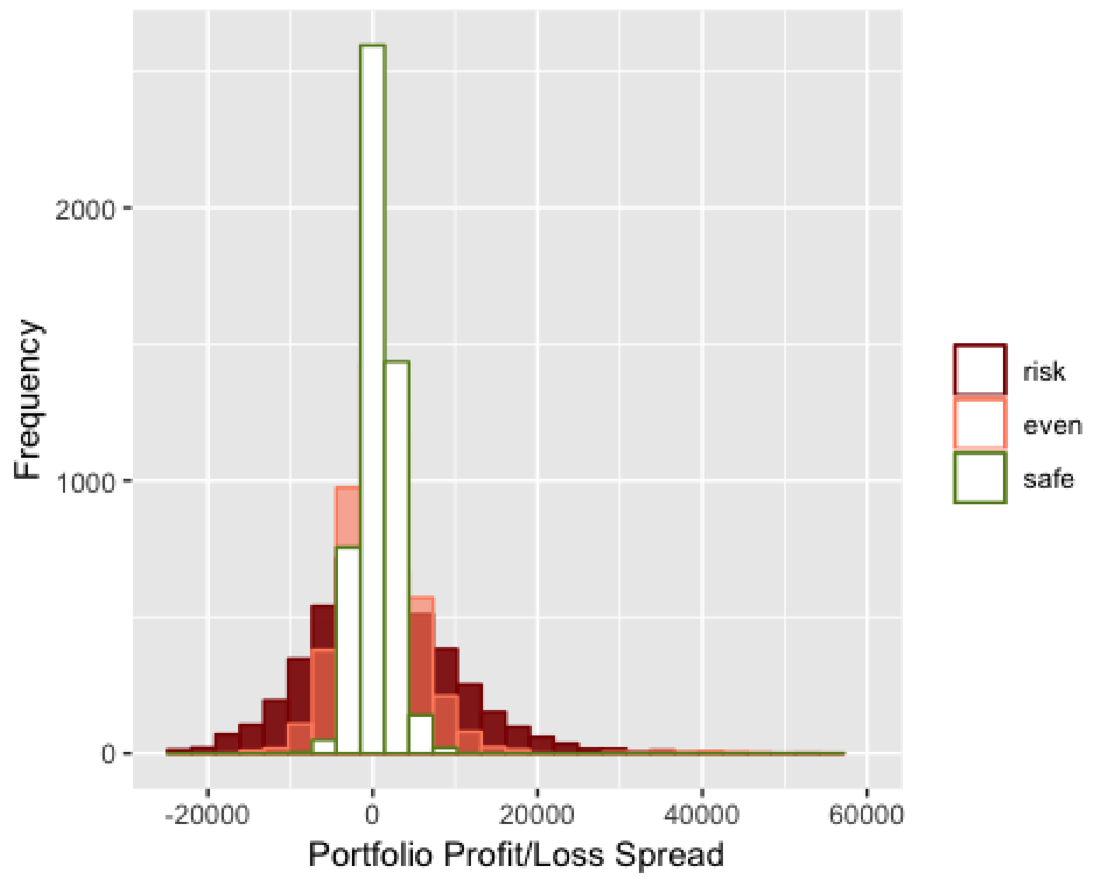
```
risky_mean <- mean(sim_risky_split[,n_days])
risky_return <- (mean(sim_risky_split[,n_days])-
initial_wealth)/initial_wealth
risky_5_perc <- quantile(sim_risky_split[,n_days], 0.05) - initial_wealth

# Print out results
cat("For a risky split investment of $100,000,\nExpected holdings:",
risky_mean, "\nReturn Rate:", risky_return, "\n5% VaR:", risky_5_perc)
```

```
## For a risky split investment of $100,000,
## Expected holdings: 101864.7
## Return Rate: 0.01864712
## 5% VaR: -12422.34
```

With the histogram below, again, we prove that portfolio that place higher proportions on ETFs with higher variance could be more volatile. The 'risk' portfolio has more chances of getting higher returns and also more loss. The ‘safe’ is the lower-volatility portfolio and as expected, it has lower chances of getting big profit or big loss.

Plot Histogram of Portfolios

```r
even = sim_even_split[,n_days]- initial_wealth
safe = sim_safe_split[,n_days]- initial_wealth
risk = sim_risky_split[,n_days]- initial_wealth

plot_df = data.frame(risk,even, safe)

ggplot(data = plot_df) +
    geom_histogram(aes(risk,color = 'risk'), fill="red4",alpha =
0.9,position="identity")+
    geom_histogram(aes(even,color = 'even'),fill = 'salmon1',alpha =
0.6,position="identity") +
    geom_histogram(aes(safe,color = 'safe'),fill = 'white',alpha =
1.0,position="identity")+
    xlim(c(-25000,60000))+
    xlab('Portfolio Profit/Loss Spread')+
    ylab('Frequency')+
    scale_colour_manual("",
                      breaks = c("risk", "even", "safe"),
                      values = c("salmon1", "red4","olivedrab4"))
```



Evaluate Portfolio Performance

```
get_return = function(initial_wealth,weights,all_returns){

    sim_split = foreach(i = 1:5000, .combine = 'rbind') %do% {
        total_wealth = initial_wealth
        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 #p+r(stock)
            total_wealth = sum(holdings)#p+r(portfolio)
            wealthtracker[today] = total_wealth
            holdings = total_wealth * weights #daily rebalance
        }
        wealthtracker
    }
    return(quantile(sim_split[,n_days], 0.05) - initial_wealth)
}

#weight_even = rep(0.2,5)
#even = get_return(100000,weight_uni,all_returns)
```

Once again, by randomly tuning the weight on those 5 ETFs, we can observe that if placing more holdings on the higher-volatility ETFs, we would expect it to lose more money of the portfolios during the worst 5% of possible 20-day periods.

```r
set.seed(1)
weight = runif(5)
weight = weight / sum(weight)
for (i in 1:10){
    weight = runif(5)
    weight = weight / sum(weight)
    initial_wealth = 100000
    VaR = get_return(initial_wealth,weight,all_returns)
    cat('The portfolio weight is: ')
    print(weight)
    cat('VaR of the portfolio is: ')
    print(VaR)
}
```

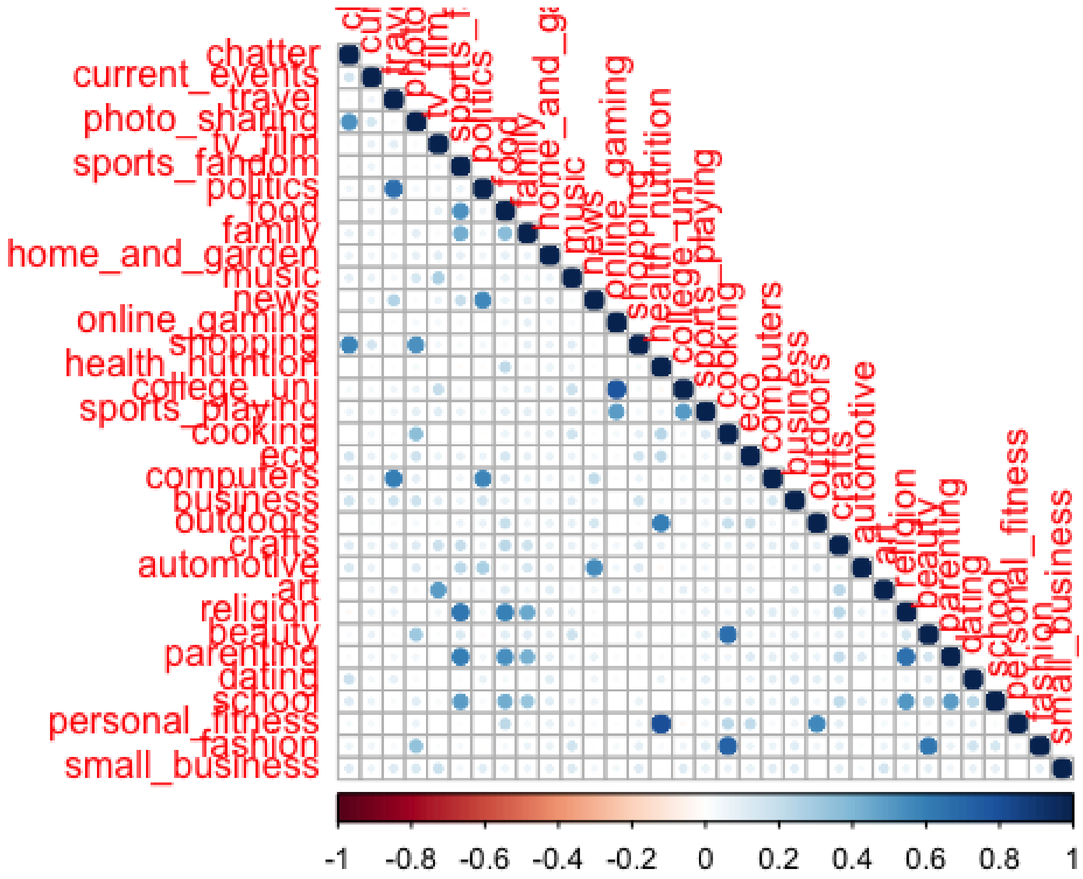
```
## The portfolio weight is: [1] 0.28120698 0.29569494 0.20683781 0.19692041
0.01933986
## VaR of the portfolio is:        5%
## -4208.322
## The portfolio weight is: [1] 0.06473905 0.45353950 0.19643722 0.00748451
0.27779972
## VaR of the portfolio is:        5%
## -4711.766
## The portfolio weight is: [1] 0.2317894 0.1058172 0.1834601 0.2476991
0.2312342
## VaR of the portfolio is:        5%
## -7038.451
## The portfolio weight is: [1] 0.37746475 0.15186376 0.19061732 0.18721150
0.09284268
## VaR of the portfolio is:        5%
## -4958.259
## The portfolio weight is: [1] 0.001673649 0.352725081 0.217737384
0.140157414 0.287706472
## VaR of the portfolio is:        5%
## -6268.753
## The portfolio weight is: [1] 0.0005496292 0.2491131647 0.3524609624
0.3284036533 0.0694725904
## VaR of the portfolio is:       5%
## -7292.21
## The portfolio weight is: [1] 0.18054499 0.08488587 0.24893533 0.28819775
0.19743606
## VaR of the portfolio is:        5%
## -8010.587
## The portfolio weight is: [1] 0.2102807 0.1492866 0.1785018 0.2136476
0.2482834
## VaR of the portfolio is:        5%
## -6928.782
## The portfolio weight is: [1] 0.19093565 0.39039666 0.06379952 0.03689110
0.31797707
## VaR of the portfolio is:       5%
## -4800.12
## The portfolio weight is: [1] 0.22974677 0.40298065 0.07596301 0.11979554
0.17151403
## VaR of the portfolio is:        5%
## -4200.626
```

Problem 4 Market segmentation

```
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms
library(factoextra) # clustering algorithms & visualization
library(corrplot)
```

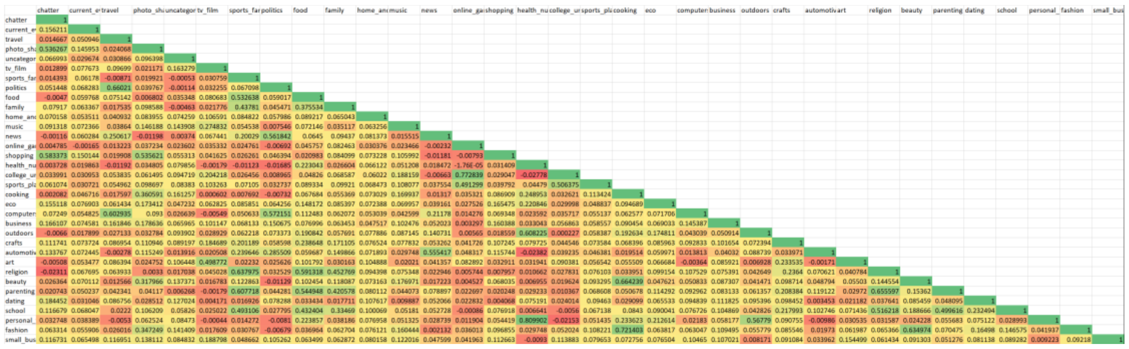
Data Exploration

```
# read file
social <-
read.csv(url("https://raw.githubusercontent.com/jgscott/STA380/master/data/so
cial_marketing.csv"), row.names = 1)
#drop columns spam, adult, and uncategorized from file
social = subset(social, select = -c(spam,adult,uncategorized))

corrplot(cor(social), type = "lower")
```



From the correlation plot, we can identidy several variables with high correlation. For example: personal_fitness and health_nutrition.

```r
knitr::include_graphics("https://raw.githubusercontent.com/tiffblahthegiraffe/STA380_HW/master/plot.png")
```



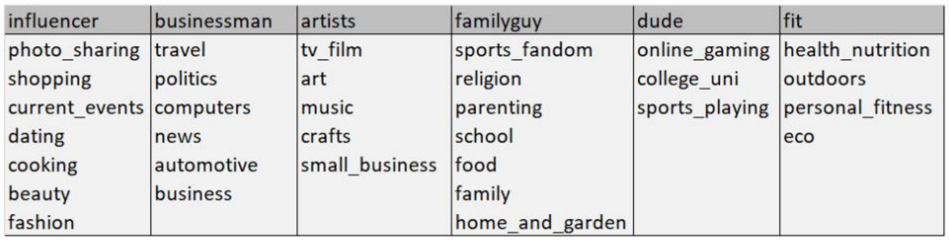
	chatter	current_e	travel	photo_shi	uncategor	tv_film	sports_far	politics	food	family	home_anc	music
chatter	1											
current_e	0.156211	1										
travel	0.014667	0.050946	1									
photo_sha	0.536267	0.145953	0.024068	1								
uncategor	0.066993	0.029674	0.030866	0.096398	1							
tv_film	0.012899	0.077673	0.09699	0.021171	0.163279	1						
sports_far	0.014393	0.06178	-0.00871	0.019921	-0.00053	0.030759	1					
politics	0.051448	0.068283	0.66021	0.039767	-0.00114	0.032255	0.067098	1				
food	-0.0047	0.059768	0.075142	0.006802	0.035348	0.080683	0.532638	0.059017	1			
family	0.07917	0.063367	0.017535	0.098588	-0.00463	0.021776	0.43781	0.045471	0.375534	1		
home_anc	0.070158	0.053511	0.040932	0.083955	0.074259	0.106591	0.084822	0.057986	0.089217	0.065043	1	
music	0.091318	0.072366	0.03864	0.146188	0.143908	0.274832	0.054538	0.007546	0.072146	0.035117	0.063256	1
news	-0.00116	0.060284	0.250617	-0.01198	0.00374	0.067441	0.20029	0.561842	0.0645	0.09437	0.081373	0.015515
online_ga	0.004785	-0.00165	0.013223	0.037234	0.023602	0.035332	0.024761	-0.00692	0.045757	0.082463	0.030376	0.023466
shopping	0.583373	0.150144	0.019908	0.535621	0.055313	0.041625	0.026261	0.046394	0.020983	0.084099	0.073228	0.105992
health_nu	0.003728	0.019863	-0.01192	0.034805	0.079856	-0.00179	-0.01123	-0.01685	0.223043	0.026604	0.066122	0.051208
college_ur	0.033991	0.030953	0.053835	0.061495	0.094719	0.204218	0.026456	0.008965	0.04826	0.068587	0.06022	0.188159
sports_pla	0.061074	0.030721	0.054962	0.098697	0.08383	0.103263	0.07105	0.032737	0.089334	0.09921	0.068473	0.108077
cooking	0.002082	0.046716	0.017597	0.360591	0.161257	0.000602	0.007692	-0.00732	0.067684	0.055369	0.073029	0.169937
eco	0.155118	0.076903	0.061434	0.173412	0.047232	0.062825	0.085851	0.064256	0.148172	0.085397	0.072388	0.069957
computer	0.07249	0.054825	0.602935	0.093	0.020639	-0.00549	0.050633	0.572151	0.112483	0.062072	0.053039	0.042599
business	0.166107	0.074581	0.161846	0.178636	0.065965	0.101147	0.068133	0.150675	0.076996	0.063453	0.047517	0.102476
outdoors	-0.0066	0.017899	0.027133	0.032784	0.093902	0.028929	0.062218	0.073373	0.190842	0.057691	0.077886	0.087145
crafts	0.111741	0.073724	0.086954	0.110946	0.089197	0.184689	0.201189	0.058598	0.238648	0.171105	0.076524	0.077832
automotiv	0.133767	0.072445	-0.00278	0.115249	0.013916	0.020508	0.239646	0.285509	0.059687	0.149866	0.071893	0.029748
art	-0.00508	0.053477	0.086394	0.024752	0.106448	0.498772	0.02232	0.025626	0.101792	0.030163	0.104888	0.02021
religion	-0.02311	0.067695	0.063933	0.0033	0.017038	0.045028	0.637975	0.032529	0.591318	0.452769	0.094398	0.075348
beauty	0.026364	0.070112	0.012566	0.317966	0.137371	0.016783	0.122863	-0.01129	0.102454	0.118087	0.073163	0.176971
parenting	0.020743	0.050237	0.042341	0.04117	0.006268	-0.00179	0.607718	0.044281	0.544948	0.420578	0.080112	0.044073
dating	0.184452	0.031046	0.086756	0.028512	0.127024	0.004171	0.016926	0.078288	0.033434	0.017711	0.107617	0.009887
school	0.116679	0.068047	0.0222	0.106209	0.05826	0.025022	0.493106	0.027795	0.432404	0.33469	0.100069	0.05181
personal_	0.032748	0.038389	-0.0053	0.062524	0.08473	-0.00044	0.014272	-0.0081	0.223857	0.038186	0.076958	0.051325
fashion	0.063314	0.055906	0.026016	0.347249	0.141409	0.017609	0.030767	-0.00679	0.036964	0.062704	0.076121	0.160444
small_bus	0.116731	0.065498	0.116951	0.138112	0.084832	0.188798	0.048662	0.105262	0.063499	0.062872	0.080158	0.122016

	news	online_ga	shopping	health_nu	college_ur	sports_pla	cooking	eco	computer	business	outdoors
news	1										
online_ga	-0.00232	1									
shopping	-0.01181	-0.00793	1								
health_nu	0.018472	-1.76E-05	0.031409	1							
college_ur	-0.00663	0.772839	0.029047	-0.02778	1						
sports_pla	0.037554	0.491299	0.039792	0.04479	0.506375	1					
cooking	0.01317	0.035321	0.086909	0.248953	0.032621	0.113424	1				
eco	0.039161	0.027526	0.165475	0.220846	0.029998	0.048837	0.094689	1			
computer	0.21178	0.014276	0.069348	0.023592	0.035717	0.055137	0.062577	0.071706	1		
business	0.052023	0.003297	0.160388	0.033043	0.056863	0.058557	0.090454	0.069033	0.145387	1	
outdoors	0.140731	0.00565	0.018559	0.608225	0.000227	0.058387	0.192634	0.174811	0.043039	0.050914	1
crafts	0.053262	0.041726	0.107245	0.079725	0.044546	0.073584	0.068396	0.085963	0.092833	0.101654	0.072394
automotiv	0.555417	0.048317	0.115744	-0.02382	0.039235	0.046381	0.019514	0.059971	0.013813	0.04032	0.088739
art	0.041357	0.082892	0.032911	0.031941	0.090381	0.056542	0.055509	0.066684	-0.00364	0.085921	0.006928
religion	0.022946	0.005744	0.007957	0.010662	0.027831	0.076103	0.033951	0.099154	0.107529	0.075391	0.042649
beauty	0.017223	0.004527	0.068035	0.006955	0.019624	0.093295	0.664239	0.047621	0.050833	0.087307	0.041471
parenting	0.078897	0.022697	0.020248	0.029233	0.010367	0.068608	0.050678	0.114292	0.092962	0.083133	0.061357
dating	0.052066	0.022832	0.004068	0.075191	0.024014	0.09463	0.029099	0.065533	0.094839	0.111825	0.095396
school	0.052728	-0.00086	0.076918	0.006641	-0.0056	0.067138	0.0843	0.090041	0.076726	0.104869	0.042826
personal_	0.028739	0.011904	0.054419	0.809902	-0.02153	0.051435	0.233623	0.212614	0.02183	0.058177	0.56779
fashion	0.002132	0.036013	0.096855	0.029748	0.052024	0.108221	0.721403	0.063817	0.063047	0.109495	0.055779
small_bus	0.047599	0.041963	0.112663	-0.0093	0.113883	0.079653	0.072756	0.076504	0.10465	0.107021	0.008171

	crafts	automotiv	art	religion	beauty	parenting	dating	school	personal_	fashion	small_bus
crafts	1										
automotiv	0.033971	1									
art	0.233535	-0.00171	1								
religion	0.2364	0.070621	0.040784	1							
beauty	0.098714	0.048794	0.05503	0.144554	1						
parenting	0.208384	0.119122	0.02972	0.655597	0.15362	1					
dating	0.098452	0.003453	0.021182	0.037641	0.085459	0.048095	1				
school	0.217993	0.102746	0.071436	0.516218	0.188666	0.499616	0.232494	1			
personal_	0.090755	-0.00986	0.030535	0.031587	0.024228	0.055683	0.075122	0.028993	1		
fashion	0.085546	0.01973	0.061987	0.065366	0.634974	0.070475	0.16498	0.146575	0.041937	1	
small_bus	0.091084	0.033962	0.154499	0.061434	0.091303	0.051276	0.081138	0.089282	0.009223	0.09218	1

With excel, we looked further into the correlations between variables, and decided to manually group several highly correlated variables together into six hypothetical groups/clusters, as displayed below:

```r
knitr::include_graphics("https://raw.githubusercontent.com/tiffblahthegiraffe/STA380_HW/master/group.png")
```



influencer	businessman	artists	familyguy	dude	fit
photo_sharing	travel	tv_film	sports_fandom	online_gaming	health_nutrition
shopping	politics	art	religion	college_uni	outdoors
current_events	computers	music	parenting	sports_playing	personal_fitness
dating	news	crafts	school		eco
cooking	automotive	small_business	food		
beauty	business		family		
fashion			home_and_garden		

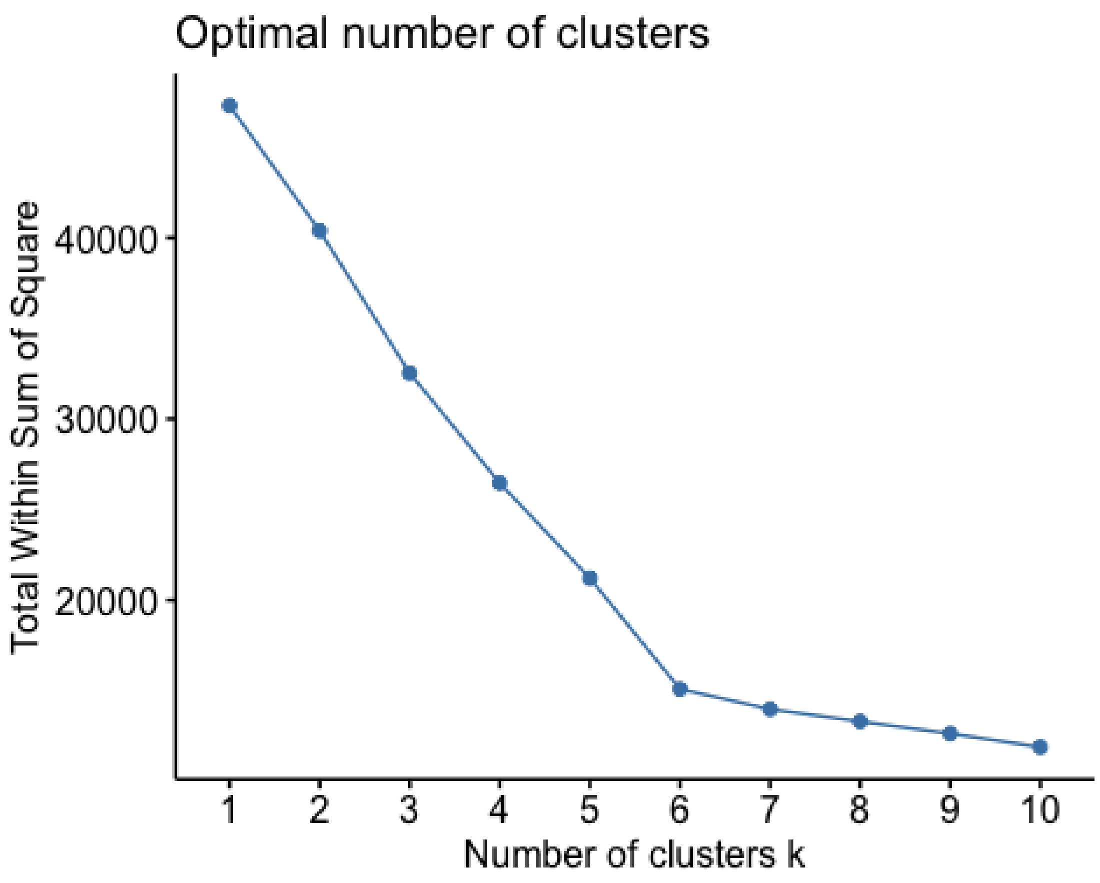
Let's dig into each group a little more: Group 1 is influencer, whom can be seem as the social media savvy millennials that like sharing their lifestyle online Group 2 represents businessman, a group of corporate people that enjoy politics, news, and business Group 3 are artists that talks about art, music and films on social media Group 4 captures familyguy. These people are very family oriented, with interests including religion, family, and parenting Group 5 is "dude." Think about a college male that likes online gaming and sports. Group 6 represents "fit." These people care about their fitness, shape, nutrition consumption, and hope to stay active

Now, with six hypothetical groups, we want to use clustering to see if our correlation-based groups serve as a group proxy as NutrientH20's market segmentation

K Means

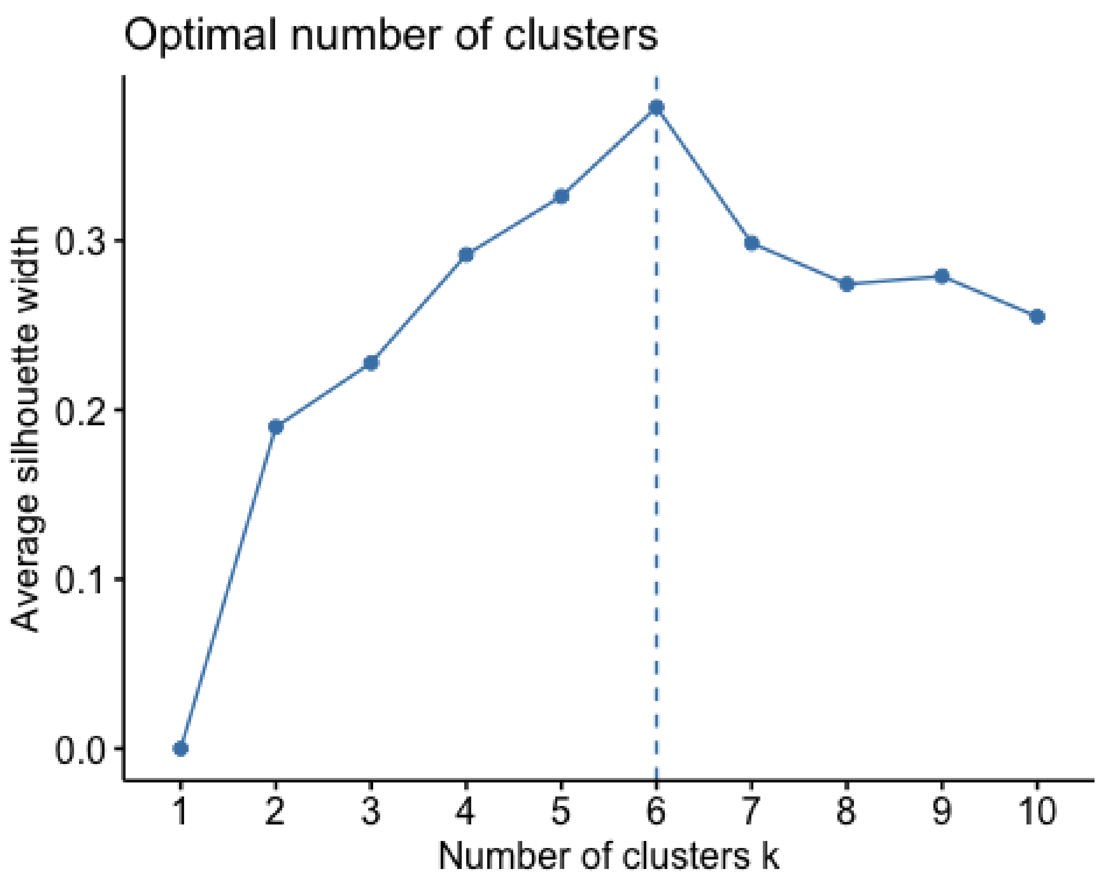
```
#create hypothetical groups in dataframe
social["influencer"] =
social$chatter+social$photo_sharing+social$shopping+social$current_events+soc
ial$dating + social$cooking+social$beauty+social$fashion
social["businessman"] = social$travel + social$politics +social$computers
+social$news + social$automotive +social$business
social['artists'] = social$tv_film +social$art +social$music +social$crafts
+social$small_business
social['familyguy'] = social$sports_fandom + social$religion
+social$parenting +social$school +social$food +social$family
+social$home_and_garden
social['dude'] = social$online_gaming +social$college_uni
+social$sports_playing
social['fit']
=social$outdoors+social$health_nutrition+social$personal_fitness+social$eco

#select six groups and form new dataframe
social_new = social[, c(34:39)]

#normalize social_new for better comparison between rows
social_norm = social_new/rowSums(social_new)

#scale social_norm
social_scaled <- scale(social_norm, center=TRUE, scale=TRUE)
```

```r
#select K
set.seed(123)
fviz_nbclust(social_scaled, kmeans, method = "wss")
```



From elbow method, it is concluded that k=6 gives us best clustering result

```r
set.seed(123)
fviz_nbclust(social_scaled, kmeans, method = "silhouette")
```



Again, K=6 returns best clusting result.

To triple check, we calculate CH

```
set.seed(123)
for (i in 2:10){
  final <- kmeans(social_scaled, centers = i , nstart = 25)
  B = final$betweenss
  final$withinss
  W = final$tot.withinss
  B/W
  n = nrow(social_scaled)
  k=i
  CH = (B/(k-1))/(W/(n-k))
  cat("k=", k, ", CH:", CH, "\n")

}
```

```
## k= 2 , CH: 1652.997
## k= 3 , CH: 1907.703
## k= 4 , CH: 2224.62
## k= 5 , CH: 2747.85
## k= 6 , CH: 3358.518
## k= 7 , CH: 3126.006
## k= 8 , CH: 2900.054
## k= 9 , CH: 2730.181
## k= 10 , CH: 2609.697
```

When K=6, CH reches its max of 3358.518 From the methods above, we get a preliminary idea that our six manually selected groups might be a good proxy for market segmentation

```
set.seed(123)
#use K=6 to run kmeans
final <- kmeans(social_scaled, centers =6 , nstart = 25)

set.seed(123)
#display centers of seven clusters to see how they are allocated
print(final$centers)

##   influencer businessman    artists  familyguy        dude        fit
## 1 -0.3958139  -0.1868781  2.2283920 -0.2565413  0.02980669 -0.3856151
## 2 -0.5158493  -0.4328407 -0.3624055 -0.3525595 -0.31408302  1.8864271
## 3 -0.6836458  -0.4425627 -0.1639833 -0.4226420  2.83906155 -0.4246151
## 4 -0.6435441  -0.4117771 -0.2058447  2.0067311 -0.30223302 -0.3872227
## 5 -0.7161993   1.9344438 -0.3028690 -0.1884180 -0.28954096 -0.4019451
## 6  1.1261119  -0.3096098 -0.2503628 -0.4000180 -0.25278227 -0.3889869
```

From the result, we can see that our hypothetical grouping method works well! Each of the six groups represents a distinct demographics.

Cluster 1 represents “artists.” Variables include: tv_film, art, music, crafts, small_business

Cluster 2 is “fit.” These people are the outdoor enthuiast that care about health_nutrition, outdoors, personal_fitness, and eco.

Cluster 3 centers on “dude,” a proxy for people who enjoy topics like online_gaming, college_uni, sports_playing.

Cluster 4 centers on “familyguy.” This category captures people who are “family oriented” and enojoy taking about topics like sports_fandom, religion, parenting, school, food, family, and home_and_garden.

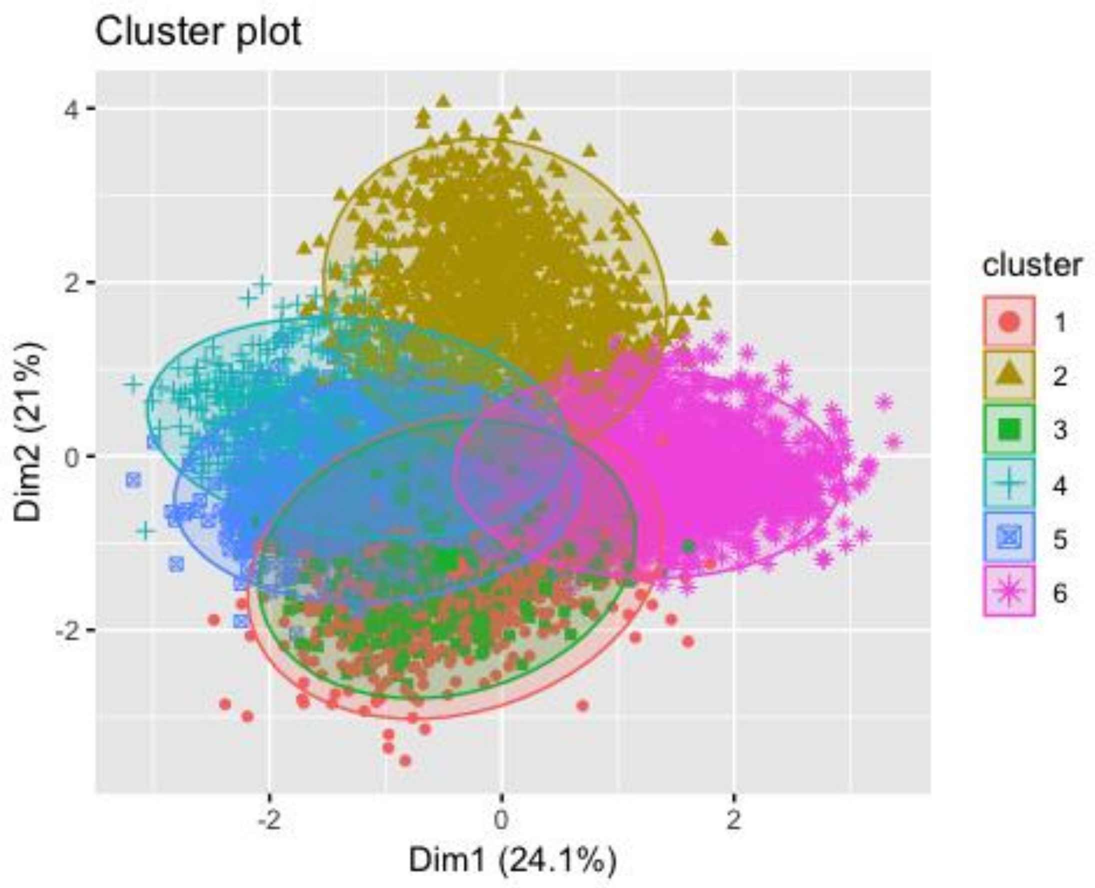
Cluster 5 centers heavily on the businessman group, which includes people who like talking about topics like travel, travel, politics, computers, news, automotive, business.

Cluster 6 captures “influencers,” which can be thought as the millennials that like sharing lifestyle related topics on social media. Topics include:photo_sharing, shopping, current_events, dating, cooking, beauty, fashion.

```
#lets see how many people belong in each group with this line of code:
final$size

## [1]  841 1362  627 1133 1205 2714
```

cluster 1 (artists) has 841 people cluster 2 (fit) has 1362 people cluster 3 (dude) represents 627 people cluster 4 (familyguy) includes 1133 people cluster 5 (businessman) has 1205 people cluster 6 (influencer) includes 2714 people

```
#kmeans plot to show 6 clusters
fviz_cluster(final, data = social_scaled, ellipse.type = "norm", stand =
TRUE, geom = "point")
```



The plot shows that our six groups are separated in a fairly clear fashion.

PCA

Now, let us move on to PCA to further analyze our hypothetical groups.

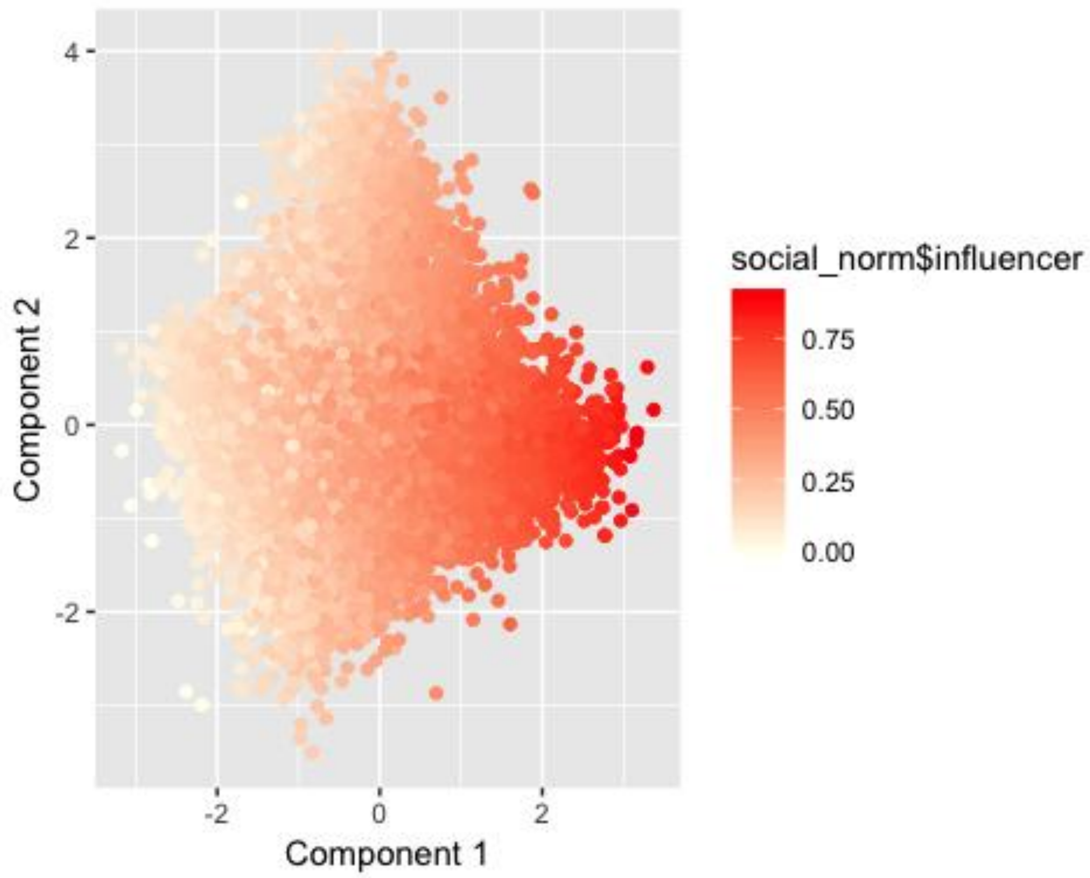
```
#Run PCA with 5 ranks
pc1 = prcomp(social_norm, scale=TRUE, rank=5)
loadings = pc1$rotation
scores = pc1$x

#several biplots show the first two PCs and how these groups are segmented

q1 = qplot(scores[,1], scores[,2], color= social_norm$influencer ,
xlab='Component 1', ylab='Component 2')
q2 = qplot(scores[,1], scores[,2], color = social_norm$businessman,
```

```r
xlab='Component 1', ylab='Component 2')
q3 = qplot(scores[,1], scores[,2], color = social_norm$artists,
xlab='Component 1', ylab='Component 2')
q4 = qplot(scores[,1], scores[,2], color = social_norm$familyguy,
xlab='Component 1', ylab='Component 2')
q5 = qplot(scores[,1], scores[,2], color = social_norm$dude, xlab='Component
1', ylab='Component 2')
q6 = qplot(scores[,1], scores[,2], color = social_norm$fit, xlab='Component
1', ylab='Component 2')
```

These plots showcase where each hypothetical group belong in the PCA two-domention result. A plot represent a twitter user, and the more red it is, the the more percentage of the user's posts relate to the corresponding group. In other words, the more red it is, the more the user belongs to a hypothetical group we created.

The influencer

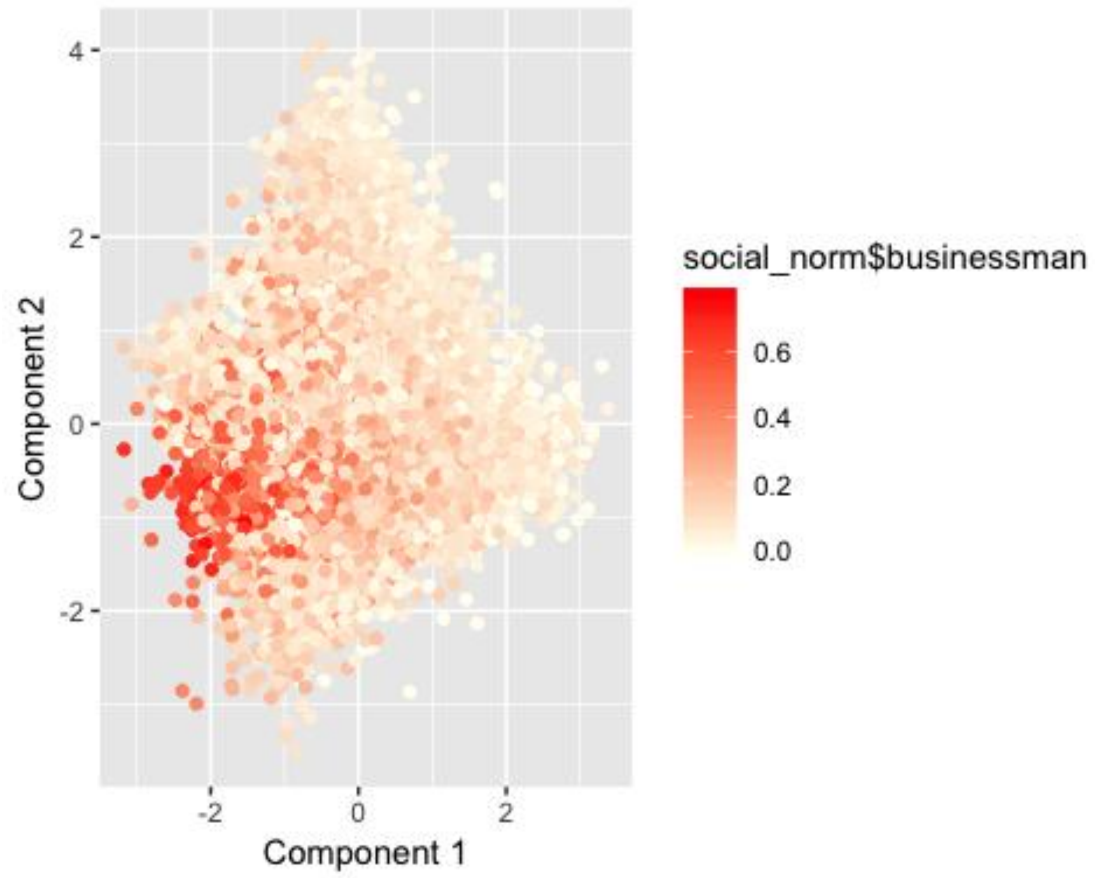
```r
#influencer
q1+scale_color_gradient(low="ivory", high="red")
```



This plot shows where influencer sits in this dimension.

The businessman

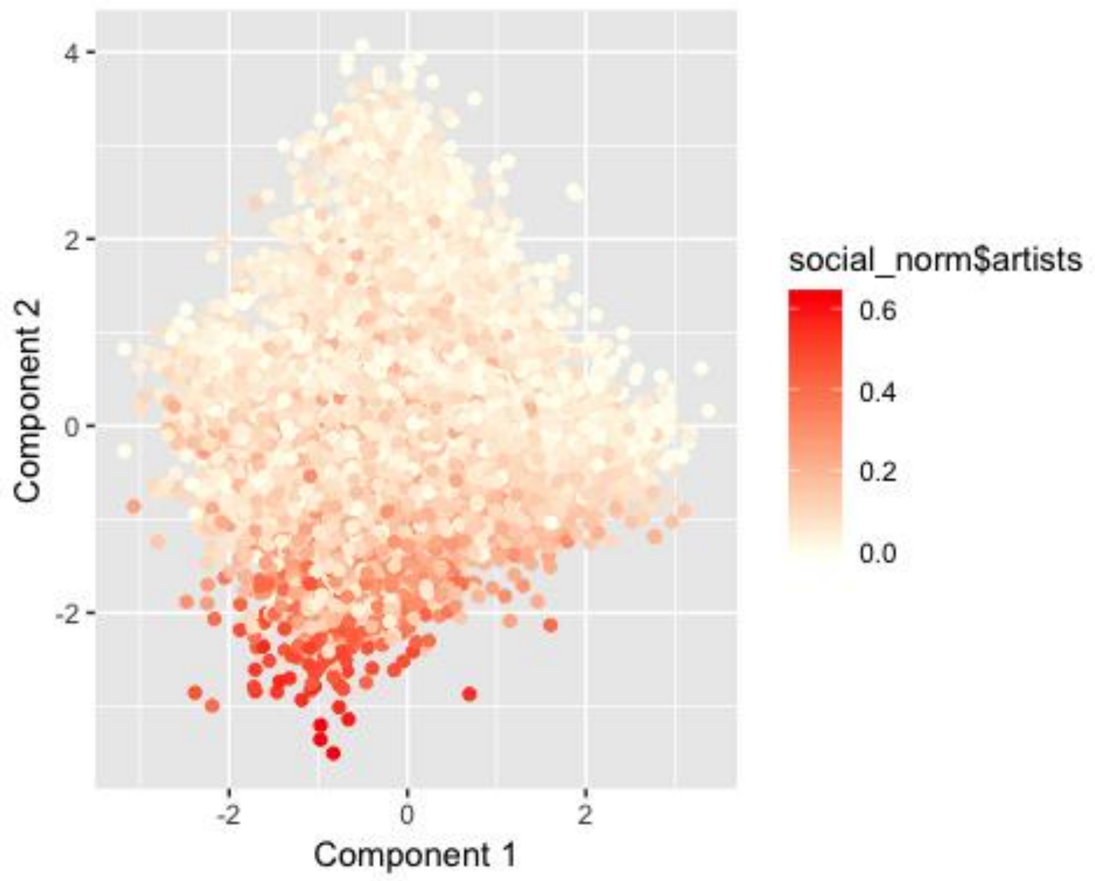
```r
#Businessman
q2+scale_color_gradient(low="ivory", high="red")
```



This plot shows businessman (red) versus others (white).

The artist

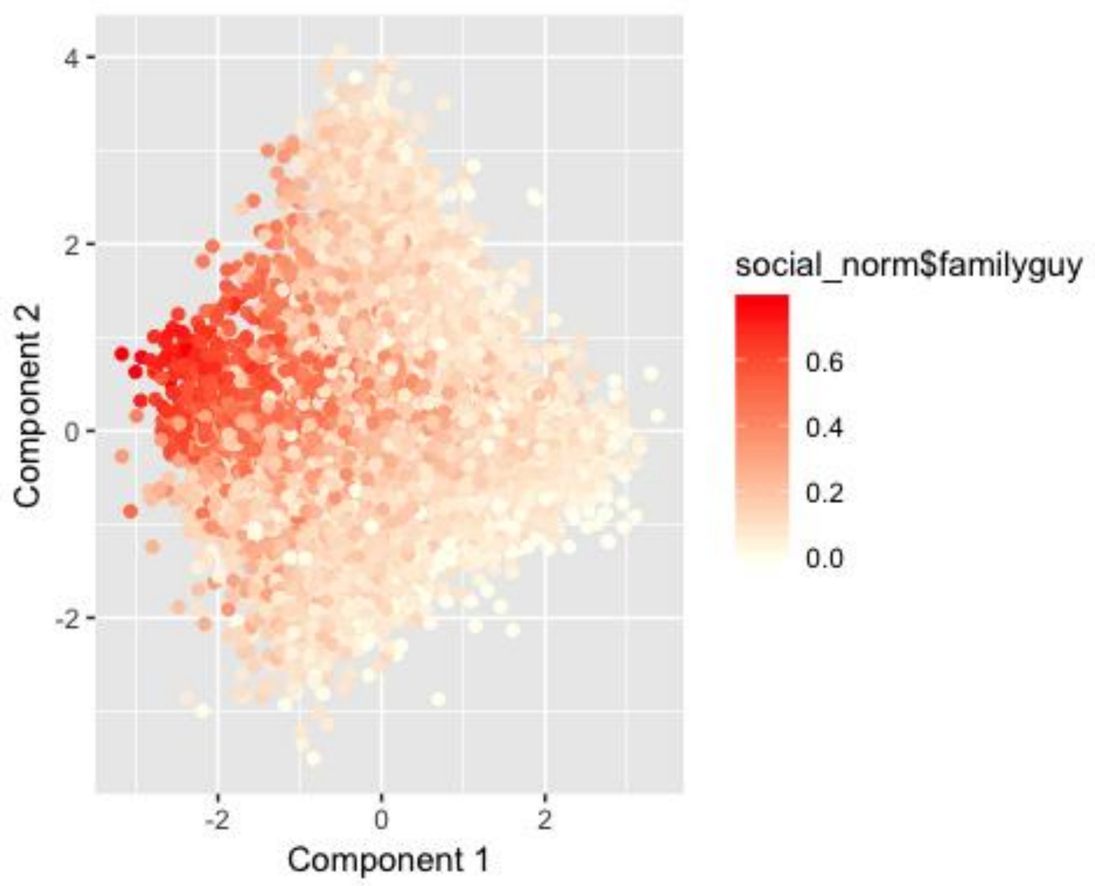
```r
#Artists
q3+scale_color_gradient(low="ivory", high="red")
```



This plot points out those where are artists are.

The familyguy

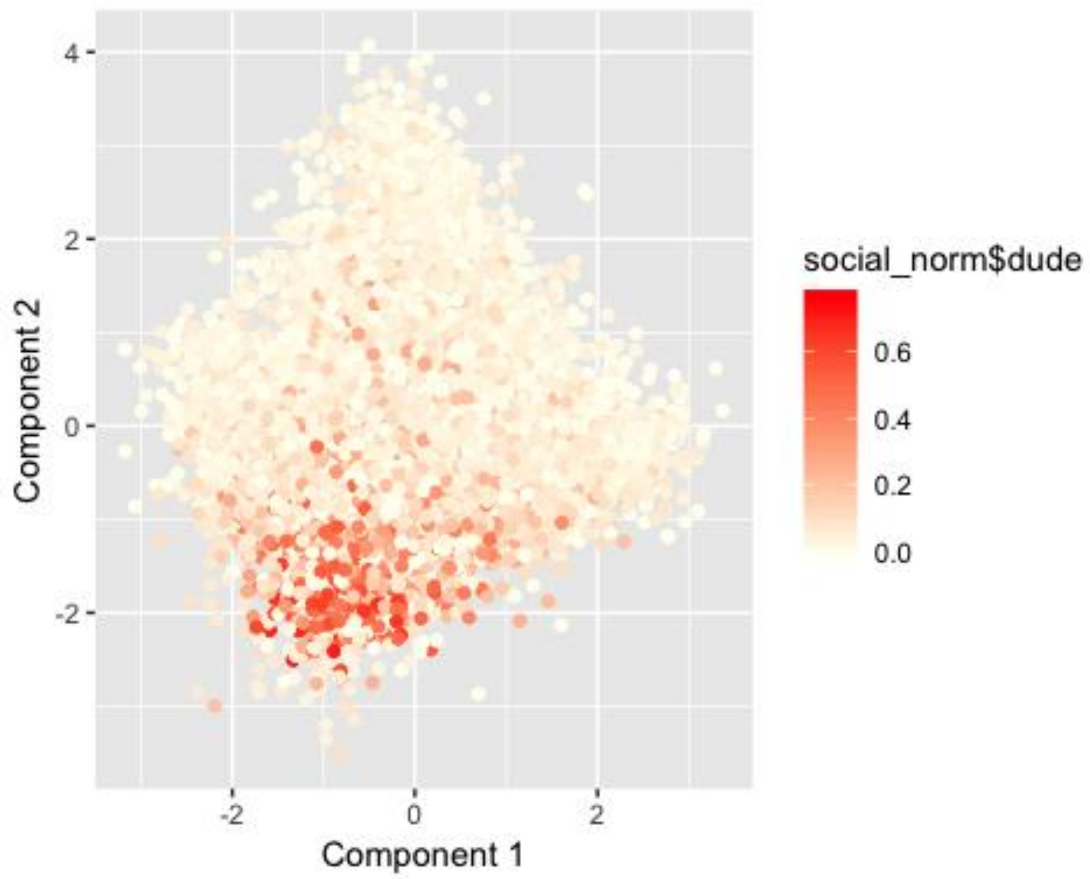
```
#Familyguy
q4+scale_color_gradient(low="ivory", high="red")
```



This plot identifies the familyguy group.

The dude

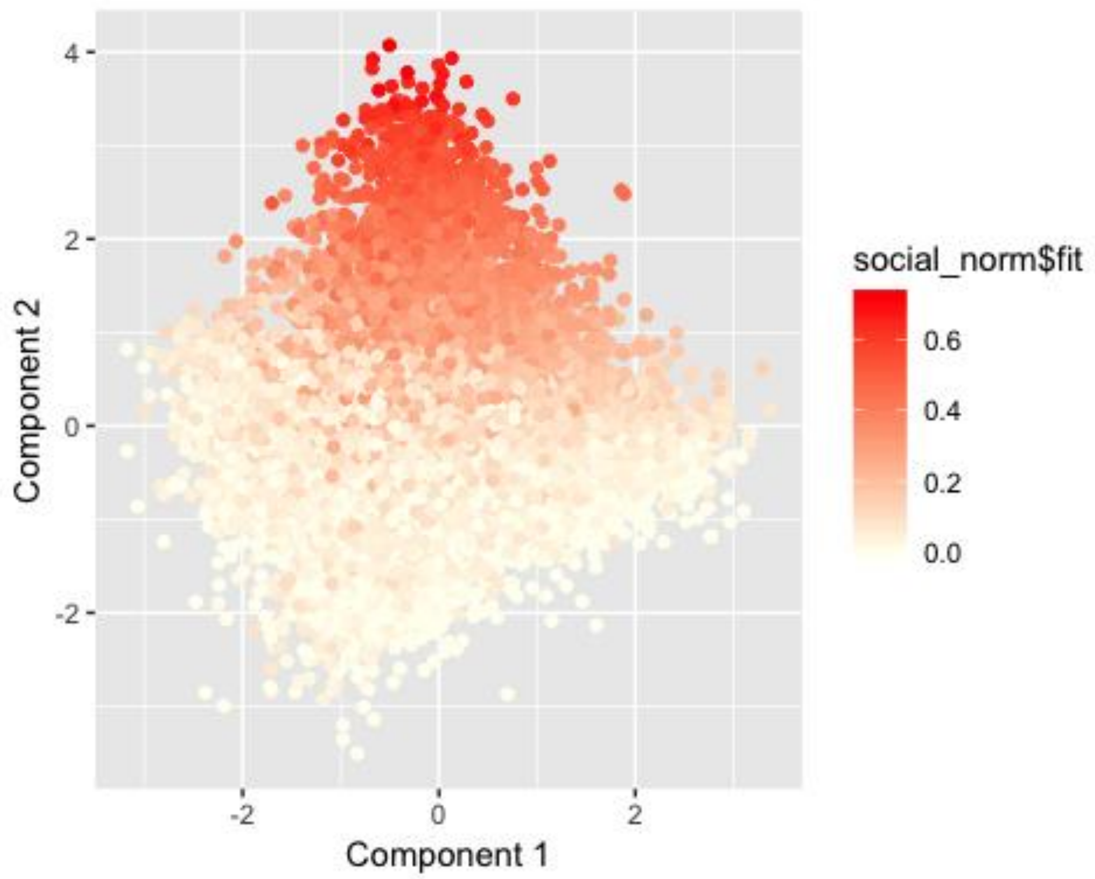
```
#Dude
q5+scale_color_gradient(low="ivory", high="red")
```



This plot shows where “dude” sit related to others

The fit

```
#Fit
q6+scale_color_gradient(low="ivory", high="red")
```



This plot clearly identifies the "fit" group

**Interpretation: component1 does a great job separating influencer (right) from familyguy and businessman (left of graph) Component 2 is great at separating "fit" (upper) from "artists" and "dude" (lower)

```
o1 = order(loadings[,1], decreasing=TRUE)
colnames(social_norm)[head(o1,2)]

## [1] "influencer" "fit"
```

from the formula below, it is clear that PC1, 2 has the ability to separate influencer and fit out from the rest of the data; result aligns with plots

```
loadings

##                      PC1        PC2         PC3         PC4         PC5
## influencer   0.78837083 -0.1120214  0.18897859 -0.18558798 -0.04929072
## businessman -0.35333354 -0.2033397  0.63810200  0.50052420 -0.02318891
## artists     -0.12683750 -0.4489010 -0.30143877 -0.05915333  0.79131622
## familyguy   -0.47330634  0.1299075  0.08223908 -0.74344082 -0.15779690
## dude        -0.11307903 -0.3699373 -0.61327767  0.24497233 -0.54891098
## fit         -0.02708018  0.7686758 -0.28874700  0.31435302  0.21130124
##                    PC6
## influencer  0.5416062
## businessman 0.4190132
## artists     0.2487053
## familyguy   0.4180165
## dude        0.3360585
## fit         0.4261025
```

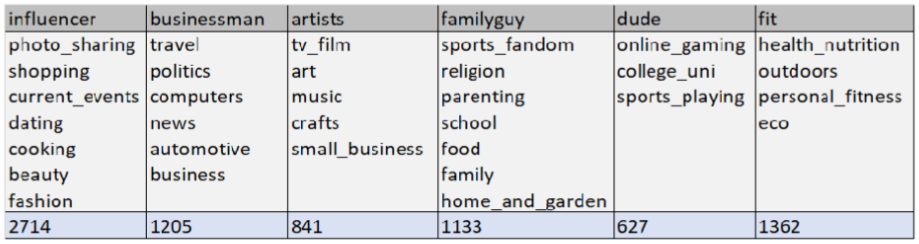
Looking at the vectors, we attain the same results.

Conclusion

From kmeans clustering and PCA, we can conclude that our hypothecial grouping method works very well in identifying users with different interests on social media, or “socialgraphics.” This output can help NutrientH20 better target its audience and focus their social media marketing efforts on a more defined and targeted group of people.

Once again, the groups, variables included, and number of peolpe in each group:

```
knitr::include_graphics("https://raw.githubusercontent.com/tiffblahthegiraffe/STA380_HW/master/final.png")
```



influencer	businessman	artists	familyguy	dude	fit
photo_sharing	travel	tv_film	sports_fandom	online_gaming	health_nutrition
shopping	politics	art	religion	college_uni	outdoors
current_events	computers	music	parenting	sports_playing	personal_fitness
dating	news	crafts	school		eco
cooking	automotive	small_business	food		
beauty	business		family		
fashion			home_and_garden		
2714	1205	841	1133	627	1362