

Exercise 1

Gaby Lio, Shirley Zhu, Jushira Thelakkat, Yuxin Li

August 10, 2017

Probability practice

Part A: What fraction of people who are truthful clickers answered yes?

- $P(RC)=0.3$
- $P(TC)=0.7$
- $P(Y|RC)=0.5$
- $P(N|RC)=0.5$
- $P(Y)=0.65$
- $P(N)=0.35$

| | RC | TC | Total |
|-------|------|-----|-------|
| Y | 0.15 | 0.5 | 0.65 |
| N | 0.15 | 0.2 | 0.35 |
| Total | 0.3 | 0.7 | 1 |

$$P(Y|TC) = 0.5/0.7 = 71.43\%$$

The fraction of people who answered yes given that they are truthful clickers is 71.43%.

Part B: Suppose someone tests positive. What is the probability that they have the disease? In light of this calculation, do you envision any problems in implementing a universal testing policy for the disease?

$$P(+|D)=0.9993$$

$$P(-|Dc)=0.9999$$

$$P(D) = 0.000025$$

$$P(Dc) = 1-0.000025=0.999975$$

| | + | - | Total |
|-------|--------------|------------|----------|
| D | 2.49825e-05 | 0.00000018 | 0.000025 |
| Dc | 1e-04 | 0.999875 | 0.999975 |
| Total | 0.0001249825 | 0.999875 | 1 |

$$P(+,D) = P(+|D) * P(D) = 2.49825e-05$$

$$P(-,Dc) = P(-|Dc)* P(Dc) = 0.999875$$

$$P(+,Dc) = P(Dc) - P(-,Dc) = 0.999975-0.999875 = 1e-04$$

$$P(+) = P(+,D)+P(+,Dc)=2.49825e-05+1e-04=0.0001249825$$

$$P(D|+) = P(+,D)/P(+) = 2.49825e-05/0.0001249825 = 0.199888 = \mathbf{19.99\%}$$

The probability of that someone has the disease given that they test positive is very low, only 19.99%. If they were to implement a universal testing policy for this disease, most people who test positive will not have

the disease ~ about 80.01% actually. This would cause chaos, and proves that a universal testing policy for this disease is not recommended.

Exploratory Analysis: Green Buildings

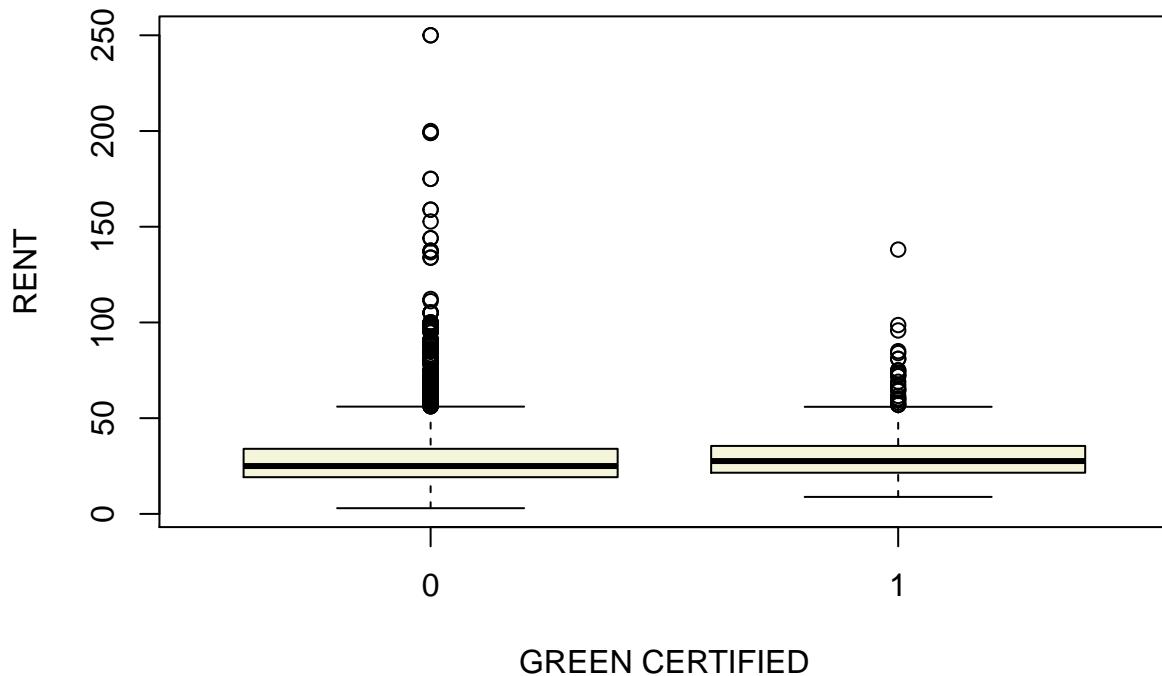
Looking through the stat gurus summary about green buildings we realized it was flawed in many ways. The first thing he did wrong was deciding to remove buildings that had less than 10% occupancy from the dataset. In our analysis we decided to keep these buildings.

We first wanted to check his claims that rent would be higher for a green building, therefore making a green building more profitable, and convincing his boss to build a green building. We created box plots of green building vs. Rent to assert these claims.

```
green_data = read.csv('greenbuildings.csv')

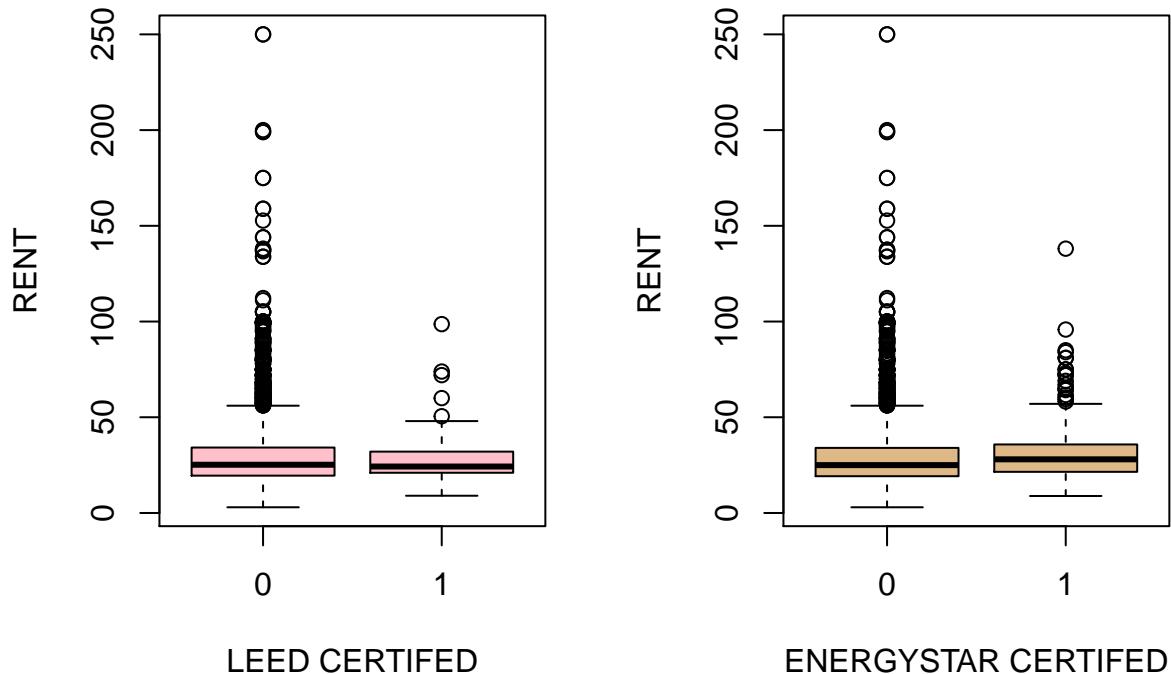
green_data$renovated = as.factor(green_data$renovated)
green_data$class_a = as.factor(green_data$class_a)
green_data$class_b = as.factor(green_data$class_b)
green_data$green_rating = as.factor(green_data$green_rating)
green_data$LEED = as.factor(green_data$LEED)
green_data$Energystar = as.factor(green_data$Energystar)
green_data$net = as.factor(green_data$net)
green_data$amenities = as.factor(green_data$amenities)
green_data$green_rating = as.factor(green_data$green_rating)

plot(green_data$green_rating, green_data$Rent, xlab='GREEN CERTIFIED', ylab='RENT', col='beige')
```



We found that having a green rating only slightly increased the amount of money you would be able to charge tenants for rent. In fact there is barely any variability between the two averages as shown in the box plots above, making the difference not statistically significant. If you go a step further and compare LEED certified vs. not and Energystar certified vs. not you see that in fact LEED energy buildings which are green certified charge a lower rent.

```
# Compare RENT values vs. being LEED or ENERGYSTAR CERTIFIED
par(mfrow = c(1,2))
plot(green_data$LEED, green_data$Rent, xlab='LEED CERTIFIED', ylab='RENT', col='pink')
plot(green_data$Energystar, green_data$Rent, xlab='ENERGYSTAR CERTIFIED', ylab='RENT', col='burlywood')
```



This already points in the direction of discrediting the gurus claim of a green building being able to charge tenants more for rent. Another major mistake the GURU did was that he only analyzed the data without looking at possible other confounding variables (i.e. age, stories, amenities, net, etc.). When just analyzing Rent vs. green or not green, you are not taking into account other effects variables have. We decided to run a linear regression to see, if holding all other variables constant, being a green building had a significant impact on rent.

```
lmgreen = lm(green_data$Rent ~ ., data=green_data)
summary(lmgreen)

##
## Call:
## lm(formula = green_data$Rent ~ ., data = green_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -53.753  -3.581  -0.526   2.491 173.916
```

```

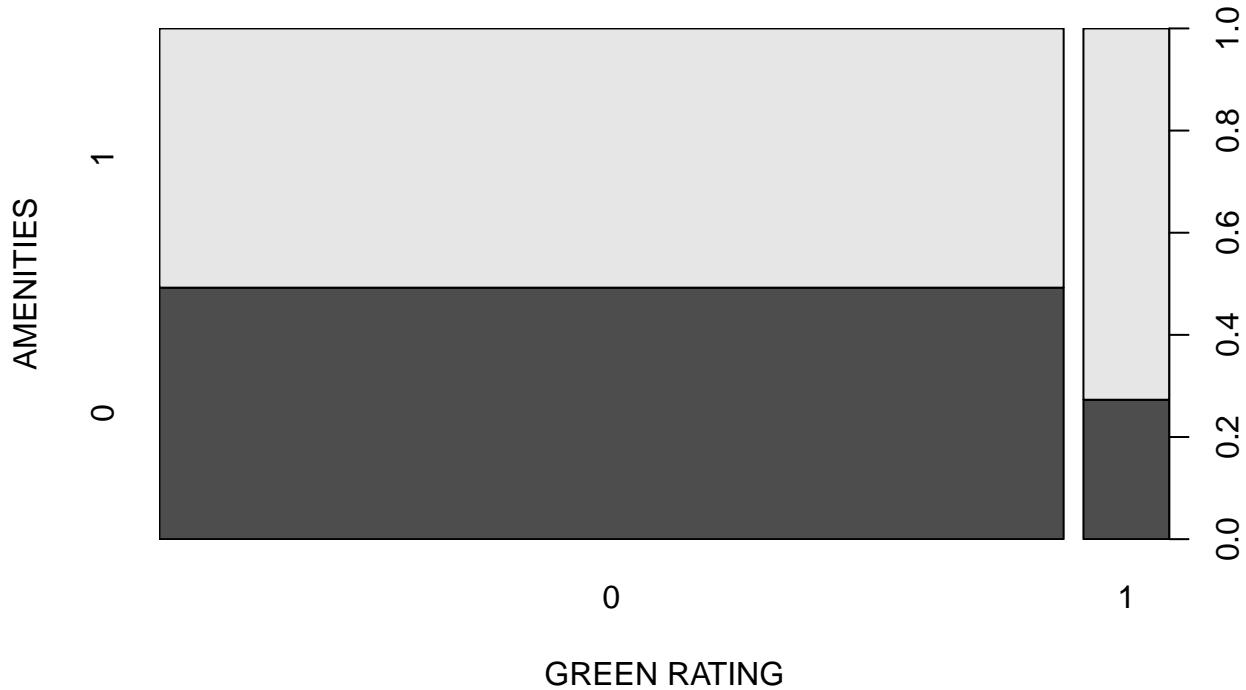
## 
## Coefficients: (1 not defined because of singularities)
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           -8.315e+00  1.018e+00 -8.167 3.67e-16 ***
## CS_PropertyID        2.959e-07  1.574e-07  1.879 0.060241 .  
## cluster              7.532e-04  2.840e-04  2.653 0.008006 ** 
## size                 6.741e-06  6.561e-07 10.276 < 2e-16 ***
## empl_gr              6.450e-02  1.700e-02  3.794 0.000149 *** 
## leasing_rate         9.454e-03  5.332e-03  1.773 0.076247 .  
## stories              -3.472e-02 1.617e-02 -2.147 0.031823 *  
## age                  -1.249e-02 4.717e-03 -2.649 0.008096 ** 
## renovated1           -1.425e-01 2.586e-01 -0.551 0.581681  
## class_a1             2.872e+00  4.377e-01  6.563 5.63e-11 *** 
## class_b1             1.186e+00  3.427e-01  3.462 0.000539 *** 
## LEED1                1.877e+00  3.582e+00  0.524 0.600318  
## Energystar1          -2.127e-01 3.818e+00 -0.056 0.955572  
## green_rating1         6.969e-01  3.839e+00  0.182 0.855929  
## net1                 -2.559e+00 5.929e-01 -4.316 1.61e-05 *** 
## amenities1            6.703e-01 2.519e-01  2.661 0.007802 ** 
## cd_total_07           -1.248e-04 1.464e-04 -0.852 0.394005  
## hd_total07            5.354e-04  8.972e-05  5.967 2.52e-09 *** 
## total_dd_07           NA       NA      NA      NA      
## Precipitation          4.830e-02 1.611e-02  2.997 0.002735 ** 
## Gas_Costs              -3.559e+02 7.842e+01 -4.538 5.76e-06 *** 
## Electricity_Costs     1.886e+02 2.493e+01  7.563 4.38e-14 *** 
## cluster_rent            1.008e+00 1.421e-02 70.949 < 2e-16 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 9.413 on 7798 degrees of freedom
##   (74 observations deleted due to missingness)
## Multiple R-squared:  0.6126, Adjusted R-squared:  0.6116 
## F-statistic: 587.2 on 21 and 7798 DF,  p-value: < 2.2e-16

```

As you can see from the output, when holding all other variables constant, having a green_rating was not significant in affecting Rent at all. Neither was a building having Energystar or LEED certifications (i.e. being green buildings). Other things that had a significant impact on rent included which cluster they belonged in, and each buildings size, age, class, net,amenities, perception costs, heating days, gas costs, and electricity costs.

We then dived deeper into these insights by seeing if green buildings tended to have amenities thus increasing the price of rent.

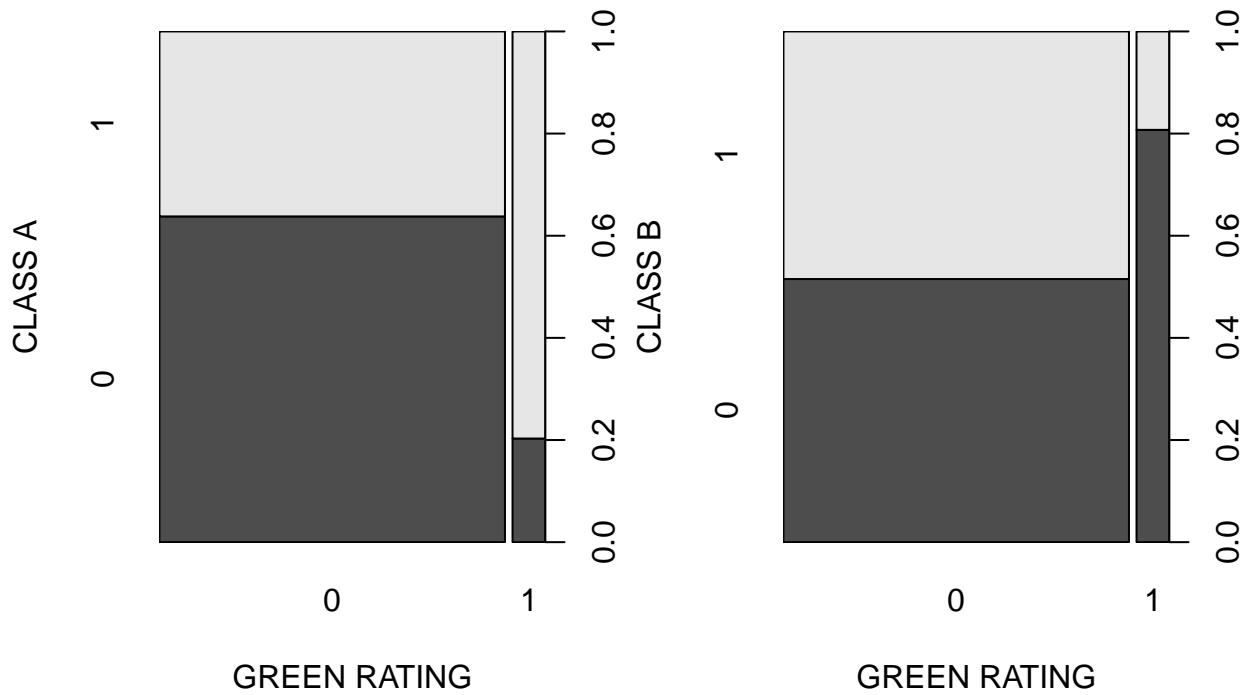
```
plot(green_data$green_rating, green_data$amenities, xlab='GREEN RATING', ylab='AMENITIES')
```



As you can see from the plot above, about 70% of green buildings have amenties which means this could be influencing the higher price of rent.

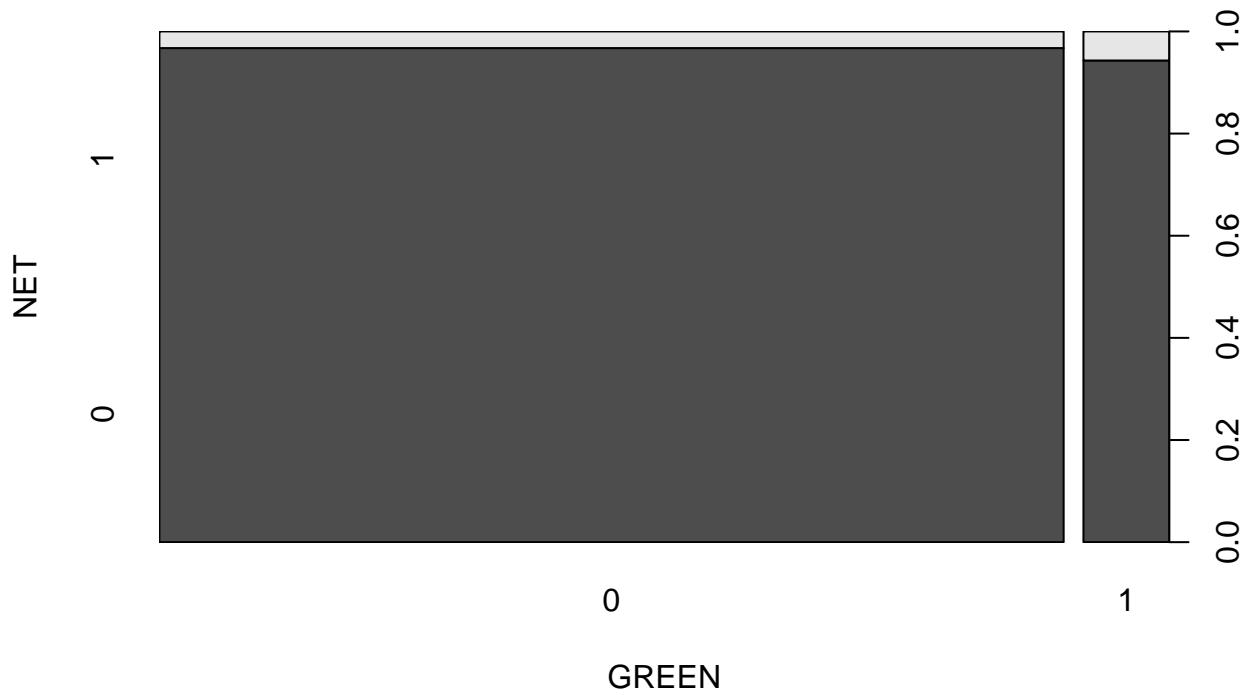
We ran the same type of analysis for the variables net, class a, and class b. You can see that most green buildings are class A (about 80%) and few are Class B (about 20%). This proves that the upcharge in price is likely due to the building being class A and not green.

```
par(mfrow=c(1,2))
plot(green_data$green_rating, green_data$class_a, xlab='GREEN RATING', ylab='CLASS A')
plot(green_data$green_rating, green_data$class_b, xlab='GREEN RATING', ylab='CLASS B')
```



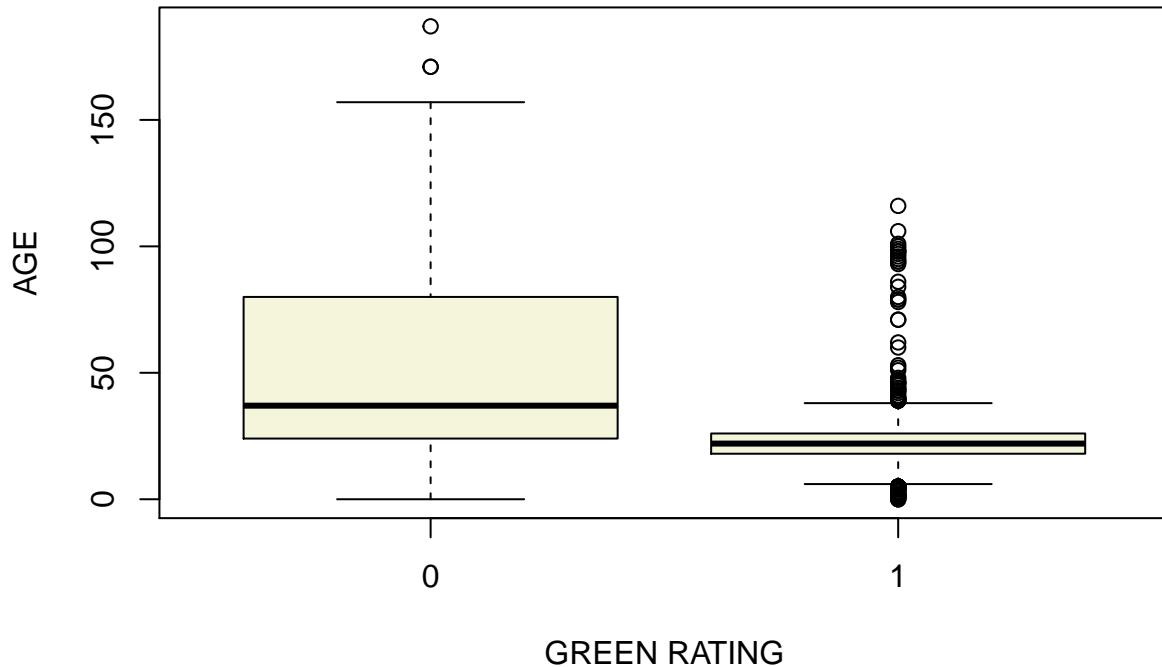
The same goes for the variable net. As you can see below, most green buildings do not have to pay for their own utilities, it is included in the rent costs. Thus adding another confounding variable to why Rent prices could be higher.

```
plot(green_data$green_rating, green_data$net, xlab='GREEN', ylab='NET')
```



Lastly, it is told to us in the problem that the building will be 15 stories and will be new. When looking at the relative age of green buildings, they are much lower than non-green buildings.

```
plot(green_data$green_rating, green_data$age, xlab='GREEN RATING', ylab='AGE', col='beige')
```



This is because green buildings are a newer concept, and did not exist a while ago. From the linear regression output we can see that Age is a significant variable, and if the building is newer, the rent will tend to be higher than if the building was old. Since most green buildings are newer than non-green buildings this could be another factor affecting the rent price.

Overall, there are too many confounding factors that effect the price of Rent. The guru solely basing his argument on the fact that green buildings have a higher rent is a wrong assumption, and therefore invalidates his analysis.

He also miscalculated the premium one could charge for having a green building. Holding all other variables constant, the premium is only \$0.07, much smaller than what the guru proposed per square foot. This means that it would take way longer than 8 years to pay off the building. This calculation although erroneous, still does not matter though because the variable was insignificant when other variables were used in the analysis.

Taking all other variables into account, the rent price is higher due to many other variables, and not just the fact that the building is green. If we had more data, such as which location cluster the building would fall into, we may be able to predict if the rent of the building would be higher. Since we only know that it will be new and have 15 stories, there is not much more we can predict and the developer should not listen to the gurus analysis.

Bootstrapping

```
library(mosaic)
## Loading required package: dplyr
##
```

```

## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
## Loading required package: lattice
## Loading required package: ggformula
## Loading required package: ggplot2
##
## New to ggformula? Try the tutorials:
##   learnr::run_tutorial("introduction", package = "ggformula")
##   learnr::run_tutorial("refining", package = "ggformula")
## Loading required package: mosaicData
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features. The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
##
## Attaching package: 'mosaic'
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median,
##     prop.test, quantile, sd, t.test, var
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
library(quantmod)

## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'

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

## The following objects are masked from 'package:dplyr':
##

```

```

##      first, last
## Loading required package: TTR
## Version 0.4-0 included new data defaults. See ?getSymbols.
library(foreach)

#Import the ETFs and use getSymbols to get their prices from 2007

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

## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data.getOption("getSymbols.env")
## andgetOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
##
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).

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

for(ticker in mystocks) {
  expr = paste0(ticker, "a = adjustOHLC(", ticker, ")")
  eval(parse(text=expr))
}

# Combine all the returns into a single matrix

all_returns = cbind( C1C1(SPYa),
                     C1C1(TLTa),
                     C1C1(LQDa),
                     C1C1(EEMa),
                     C1C1(VNQa))
head(all_returns)

##          C1C1.SPYa    C1C1.TLTa    C1C1.LQDa    C1C1.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  0.0000000000  0.0001866169 -0.022336235
## 2007-01-10  0.0033315799 -0.004475797 -0.0013063264 -0.002303160
##          C1C1.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))

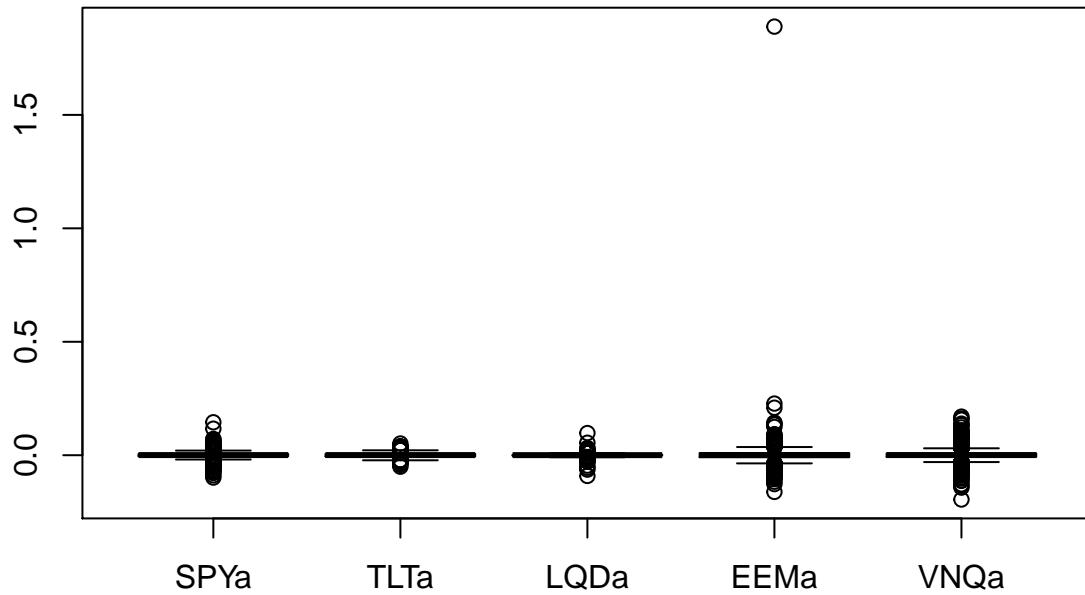
# The sample correlation matrix
cor(all_returns)

##          C1C1.SPYa  C1C1.TLTa  C1C1.LQDa  C1C1.EEMa  C1C1.VNQa
## C1C1.SPYa  1.0000000 -0.4434811  0.10255294  0.40329628  0.77671377
## C1C1.TLTa -0.4434811  1.0000000  0.42152831 -0.16861612 -0.26520718
## C1C1.LQDa  0.1025529  0.4215283  1.00000000  0.08800942  0.06703847
## C1C1.EEMa  0.4032963 -0.1686161  0.08800942  1.00000000  0.29131651
## C1C1.VNQa  0.7767138 -0.2652072  0.06703847  0.29131651  1.00000000

boxplot(all_returns, main="Daily Return Distribution by Investment Type", names= c("SPYa", "TLTa", "LQDa",

```

Daily Return Distribution by Investment Type



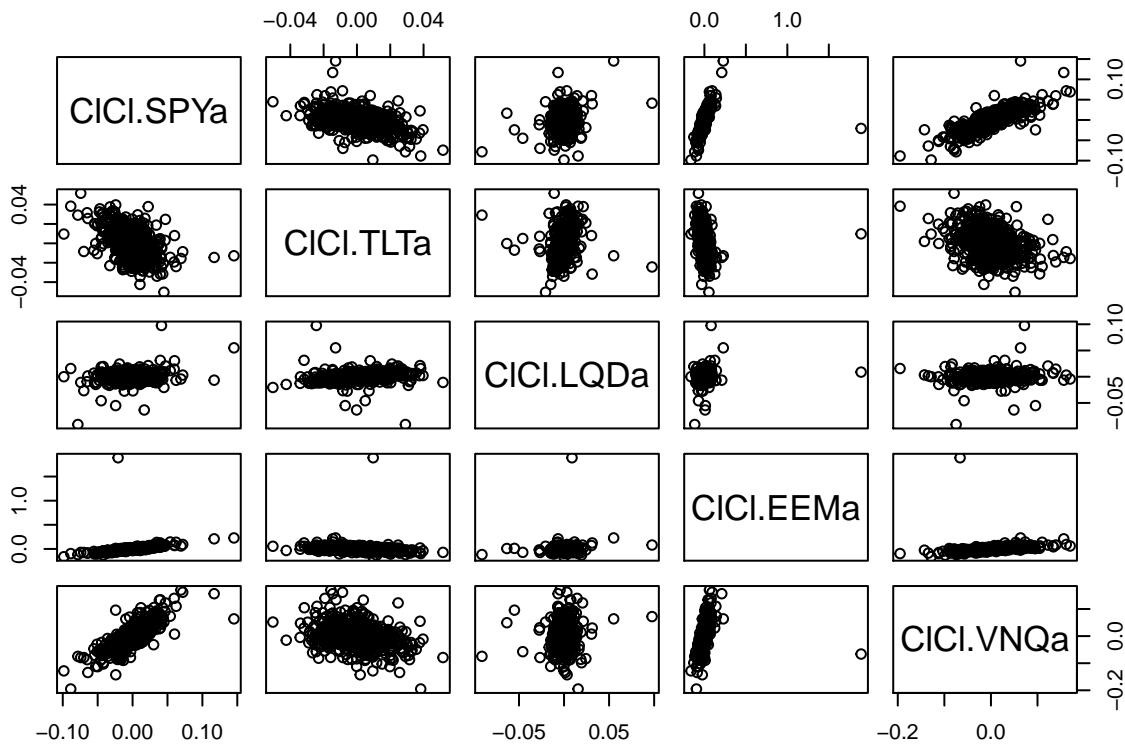
Looking at the boxplots, the safest investments are Investment-grade corporate bonds (LQD), US Treasury bonds (TLT) and US domestic equities (QPY) . Their inter-quartile range (IQR, which captures the middle 50% of the data) is $< +/- 0.005$.

The riskier investments are emerging market equities (EEM) and real estate (VNQ), with IQRs between $\sim +/- 0.01$. There is a tradeoff between return and risk, with the higher risk investments yielding higher potential returns.

```

# Compute the returns from the closing prices
pairs(all_returns)

```



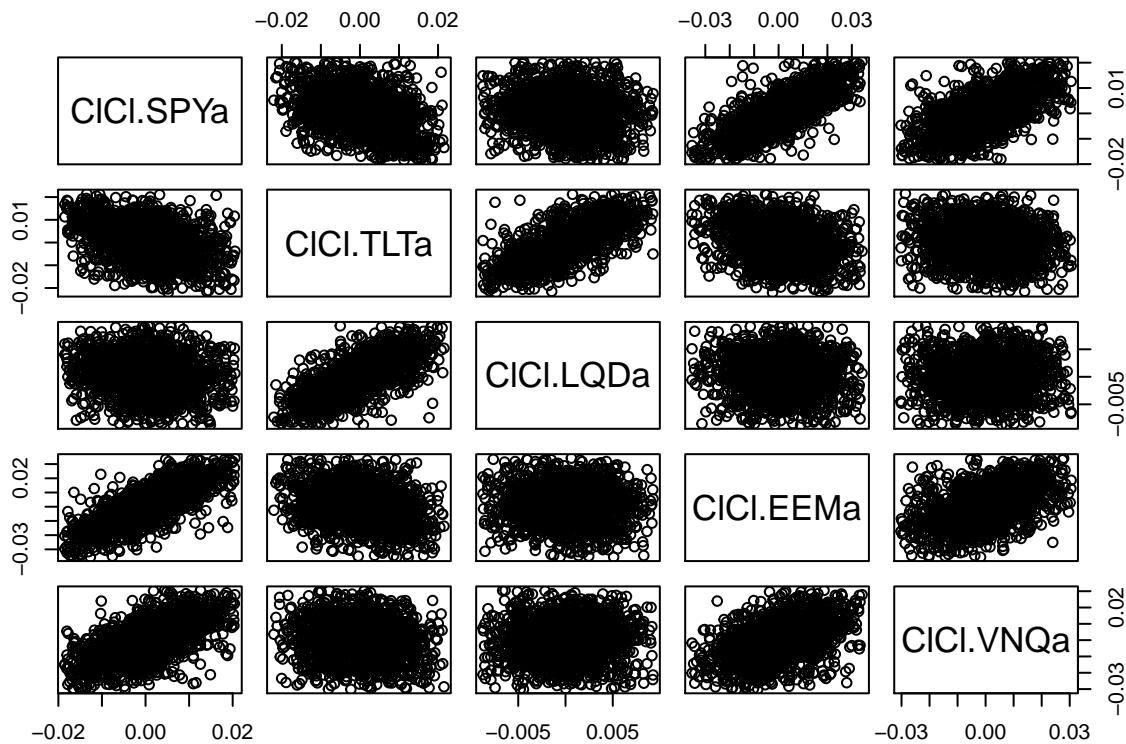
We found out there are outliers in the datapoint, especially in the CLCL data that are affecting the viewing ad interpretation of the graphs. Thus, we removed the extreme outliers.

```
#define a function to remove outliers
remove_outliers <- function(x, na.rm = TRUE, ...) {
  qnt <- quantile(x, probs=c(.25, .75), na.rm = na.rm, ...)
  H <- 1.5 * IQR(x, na.rm = na.rm)
  y <- x
  y[x < (qnt[1] - H)] <- NA
  y[x > (qnt[2] + H)] <- NA
  y
}

# Compute the returns from the closing prices
n = seq(1,dim(all_returns)[2])
all_clean_returns <- data.frame(0)
for (i in n){
  y <- remove_outliers(all_returns[,i])
  all_clean_returns = cbind(all_clean_returns,y)
  names(all_clean_returns)[names(all_clean_returns) == 'y'] <- colnames(all_returns)[i]
}
all_clean_returns = as.matrix(na.omit(all_clean_returns[,2:6]))
```

Let's plot the cleaned dataset.

```
pairs(all_clean_returns)
```



By looking at the pairs plot, we can estimate that SPY and TLT are negatively correlated; SPY and EEM, VNQ has a very positive correlation with each other. LQD has a positive correlation with TLT but not much with SPY,EEM or VNQ.

Consider the correlation between SPY and EEM and VNQ, we will pick one out of the three with the least standard deviation(lowest risk). Since SPY and TLT negatively correlated, we will keep both of them in a safe profile, thus, when SPY goes down, TLT will go up to balance out the risk and vice versa. Since LQD has the lowest standard deviation, LQD will be included in the safe profile as well.

Since the dropped-duplicates dataset is only for plotting purposes, and the outliers should be properly sampled in the future simulations, we will continue to work with the `all_returns` dataframe.

```
# Get the standard deviations and Value at risk for each of the five ETFs for comparision

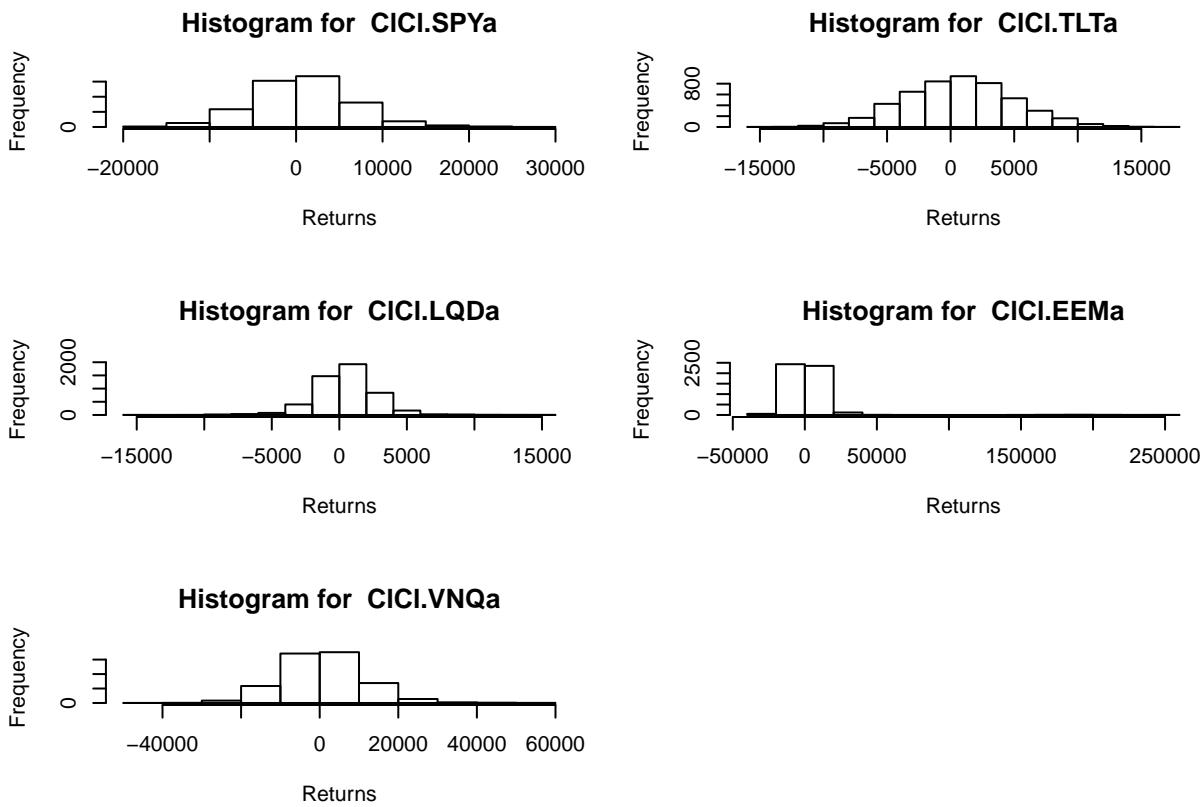
VaR_all=NULL
mean_all=NULL
sd_all=NULL
par(mfrow=c(3,2))
n_days=20
set.seed(1)
for (j in 1:ncol(all_returns)){
  # Now simulate many different possible years
  sim = foreach(i=1:5000, .combine='rbind') %do% {
    totalwealth = 100000
    weights = 1
```

```

holdings = weights * totalwealth
wealthtracker = rep(0, n_days) # Set up a placeholder to track total wealth
for(today in 1:n_days) {
  return.today = resample(all_returns[,j], 1, orig.ids=FALSE)
  holdings = holdings + holdings*return.today
  totalwealth = sum(holdings)
  wealthtracker[today] = totalwealth
}
wealthtracker
}

hist(sim[,n_days]- 100000, main = paste('Histogram for ',colnames(all_returns)[j],sep=" "), xlab = "R
# Calculate 5% value at risk
VaR_all[j] = quantile(sim[,n_days], 0.05) - 100000
mean_all[j] = mean(sim[,n_days])
sd_all[j] = sd(sim[,n_days])
}

```



```

VaR_all
## [1] -8457.867 -6112.391 -2971.231 -13418.160 -14965.009
mean_all
## [1] 100780.3 100693.6 100464.9 101894.7 100822.1

```

```

sd_all

## [1] 5729.955 4259.425 2348.712 18794.654 10216.446

```

To further support our previous findings, we see that EEM and VNQ have very high standard deviations (standard deviation being a measure of risk) and value at risk values compared to LQD, TLT and SPY. So if an investor is looking for a high return, he/she should invest in these ETFs which have the highest risk.

We also see that LQD seems to be the ETF with the least value at risk and least standard deviation. So we will look to give this ETF a relatively higher weight in our safe portfolio.

Even Split Portfolio

Here we give equal weights of 0.2 each to all the five ETFs.

```

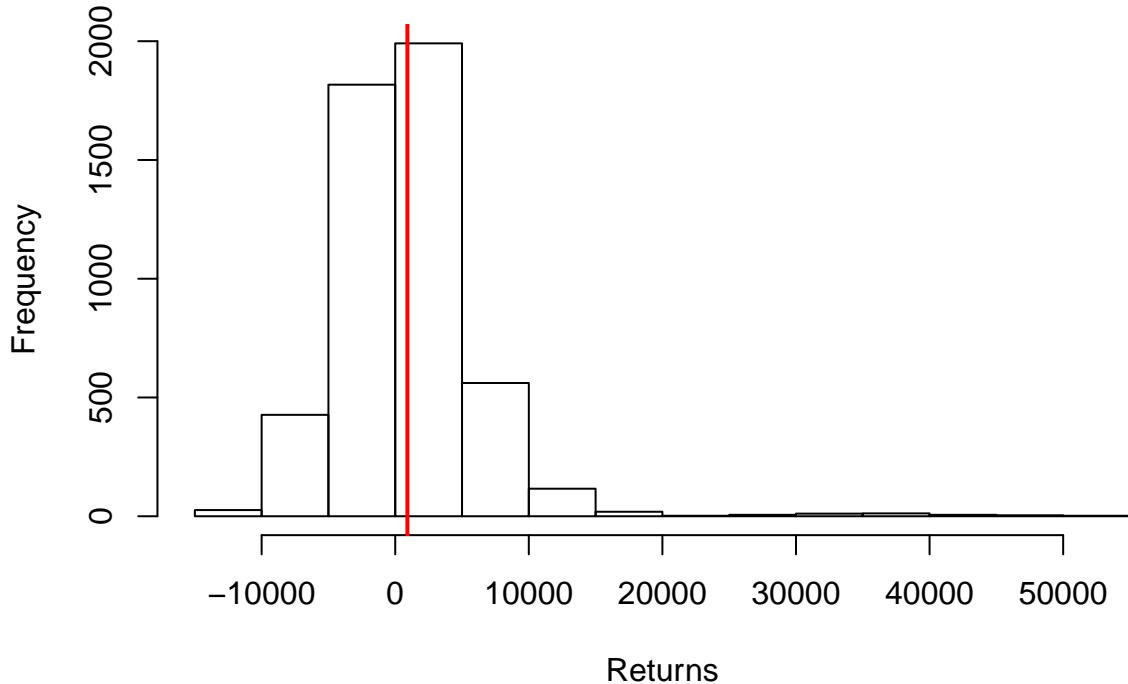
initialwealth=100000
sim_even = foreach(i=1:5000, .combine='rbind') %do% {
totalwealth_even = 100000
n_days = 20
weights_even = c(0.2, 0.2, 0.2, 0.2, 0.2)
holdings_even = weights_even * totalwealth_even
wealthtracker_even = rep(0, n_days) # Set up a placeholder to track total wealth

for(today in 1:n_days) {
return.today = resample(all_returns, 1, orig.ids=FALSE)
holdings_even = holdings_even + holdings_even*return.today
totalwealth_even = sum(holdings_even)
wealthtracker_even[today] = totalwealth_even
}
wealthtracker_even

hist(sim_even[,n_days]- 100000, main = "Histogram of returns - Even Split", xlab = "Returns")
abline(v=mean(sim_even[,n_days]- 100000), col="red", lwd=2)

```

Histogram of returns – Even Split



Safer Portfolio

For a safer choice than the even split above, we decided on only investing in the three safest investments; US domestic markets, US Treasury bonds, and corporate bonds. This will minimize the chance of a big loss, but will also limit potential for large gains. Historically the bond market has been less vulnerable to price swings or volatility than the stock market.

A safer stock would be the one which does not vary much due to changes in other stocks and has positive returns. In this case, LQD is one such stock with close to zero correlation with all other stocks except TLT.

Also, among LQD, TLT and SPY, LQD has the least standard deviation (least risk) so we give a higher weight of 0.5 to LQD and lesser weight of 0.2 to SPY.

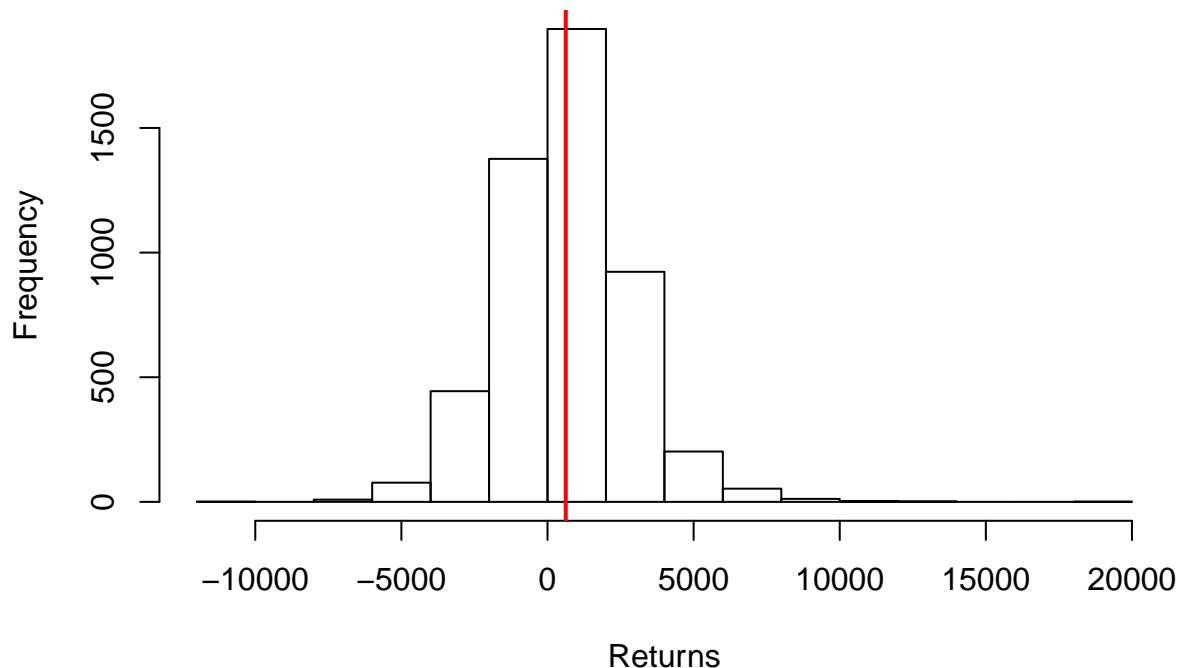
```
initialwealth=100000
sim_safe = foreach(i=1:5000, .combine='rbind') %do% {
  totalwealth_safe = 100000
  n_days = 20
  weights_safe = c(0.2, 0.3, 0.5, 0, 0)
  holdings_safe = weights_safe * totalwealth_safe
  wealthtracker_safe = rep(0, n_days) # Set up a placeholder to track total wealth
  for(today in 1:n_days) {
    return.today = resample(all_returns, 1, orig.ids=FALSE)
    holdings_safe = holdings_safe + holdings_safe*return.today
    totalwealth_safe = sum(holdings_safe)
    wealthtracker_safe[today] = totalwealth_safe
  }
}
```

```

wealthtracker_safe
}
hist(sim_safe[,n_days]- 100000, main = "Histogram of returns - Safe", xlab = "Returns")
abline(v=mean(sim_safe[,n_days]- 100000), col="red", lwd=2)

```

Histogram of returns – Safe



Riskier Portfolio

For the risky portfolio, we decided on investing an even split in the two riskiest investments from above; emerging market equities (EEM) and real estate (VNX) . This will maximize the chance at a big return, but will also open the possibility for severe losses.

```

initialwealth=100000
sim_risk = foreach(i=1:5000, .combine='rbind') %do%
totalwealth_risk = 100000
n_days = 20
weights_risk = c(0, 0, 0, 0.5, 0.5)
holdings_risk = weights_risk * totalwealth_risk
wealthtracker_risk = rep(0, n_days) # Set up a placeholder to track total wealth
for(today in 1:n_days) {
return.today = resample(all_returns, 1, orig.ids=FALSE)
holdings_risk = holdings_risk + holdings_risk*return.today
totalwealth_risk = sum(holdings_risk)
wealthtracker_risk[today] = totalwealth_risk
}
wealthtracker_risk

```

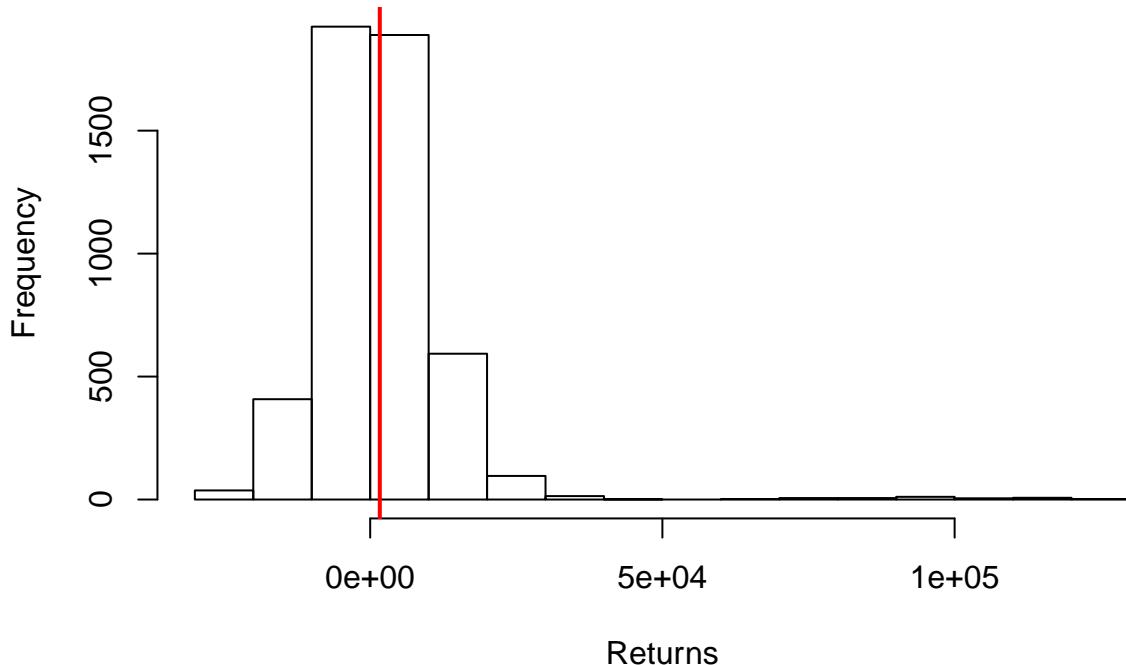
```

}

hist(sim_risk[,n_days]- 100000, main = "Histogram of returns - Aggressive", xlab = "Returns")
abline(v=mean(sim_risk[,n_days]- 100000), col="red", lwd=2)

```

Histogram of returns – Aggressive



```

names = c("Even", "Safe", "Aggressive")
average = c(mean(sim_even[,20]), mean(sim_safe[,20]), mean(sim_risk[,20]))
profit_prob = c(sum(sim_even[,20]>100000)/5000, sum(sim_safe[,20]>100000)/5000, sum(sim_risk[,20]>100000)/5000)
VaR = c((quantile(sim_even[,n_days], 0.05) - 100000), (quantile(sim_safe[,n_days], 0.05) - 100000), (quantile(sim_risk[,n_days], 0.05) - 100000))

data.frame(names, VaR)

```

```

##      names      VaR
## 1    Even -6242.313
## 2     Safe -2892.832
## 3 Aggressive -12890.714

```

The aggressive portfolio is definitely the riskiest with the highest absolute value at risk. The largest percentage of the portfolio value that one might lose over a given time period is 13000 dollars for a risky portfolio, to a 5% degree of certainty.

The safe portfolio has the least absolute value at risk. The largest percentage of the portfolio value that one might lose over a given time period is just 2900 dollars for a safe portfolio, to a 5% degree of certainty.

```
data.frame(names, average, profit_prob)
```

```

##      names   average profit_prob
## 1    Even 100912.3      0.5460
## 2     Safe 100624.5      0.6186

```

```
## 3 Aggresive 101642.5      0.5264
```

Average simulated values of the portfolios and the probability of making a profit can help make investment decisions.

The aggressive portfolio which gave equal weights to the two riskiest ETFs has the highest average return (high risk-high return) but only 52% of portfolios would result in a profit.

The safe portfolio that gave maximum weight to LQD and lesser weights to TLT and SPY, has least average profit, but 61% of portfolios would result in a profit.

So the investor faces this risk-return tradeoff at the portfolio level while considering investment decisions.
Market Segmentation

We decided to use clustering to see if we could find the different market segments for the company. As for the data pre-processing, we did not remove any variables but made sure to center and scale the variables before we ran the K-means regression. We figured that K-means would be the simplest way to identify different segments in the market through clustering.

```
library(ggplot2)
library(LICORS) # for kmeans++
library(foreach)
library(mosaic)

socialmarketing = read.csv('social_marketing.csv', header=TRUE)
dim(socialmarketing)

## [1] 7882 37

# Center and scale the data
X = socialmarketing[,-(1:1)]
X = scale(X, center=TRUE, scale=TRUE)
summary(X)

##      chatter      current_events      travel      photo_sharing
##  Min. :-1.2464  Min. :-1.2028  Min. :-0.6935  Min. :-0.9873
##  1st Qu.:-0.6797 1st Qu.:-0.4147 1st Qu.:-0.6935 1st Qu.:-0.6212
##  Median :-0.3963  Median :-0.4147  Median :-0.2560  Median :-0.2551
##  Mean   : 0.0000  Mean   : 0.0000  Mean   : 0.0000  Mean   : 0.0000
##  3rd Qu.: 0.4537 3rd Qu.: 0.3733 3rd Qu.: 0.1816 3rd Qu.: 0.4771
##  Max.   : 6.1208  Max.   : 5.1019  Max.   :10.6824  Max.   : 6.7008
##  uncategorized      tv_film      sports_fandom      politics
##  Min. :-0.8687  Min. :-0.64522  Min. :-0.7377  Min. :-0.59009
##  1st Qu.:-0.8687 1st Qu.:-0.64522 1st Qu.:-0.7377 1st Qu.:-0.59009
##  Median : 0.1998  Median :-0.04237  Median :-0.2749  Median :-0.26018
##  Mean   : 0.0000  Mean   : 0.00000  Mean   : 0.0000  Mean   : 0.00000
##  3rd Qu.: 0.1998 3rd Qu.:-0.04237 3rd Qu.: 0.1879 3rd Qu.: 0.06973
##  Max.   : 8.7482  Max.   : 9.60325  Max.   : 8.5177  Max.   :11.61664
##      food          family      home_and_garden      music
##  Min. :-0.7871  Min. :-0.7628  Min. :-0.7068  Min. :-0.6595
##  1st Qu.:-0.7871 1st Qu.:-0.7628 1st Qu.:-0.7068 1st Qu.:-0.6595
##  Median :-0.2239  Median : 0.1202  Median :-0.7068  Median :-0.6595
##  Mean   : 0.0000  Mean   : 0.0000  Mean   : 0.0000  Mean   : 0.0000
##  3rd Qu.: 0.3393 3rd Qu.: 0.1202 3rd Qu.: 0.6506 3rd Qu.: 0.3114
##  Max.   : 8.2242  Max.   : 8.0668  Max.   : 6.0803  Max.   :11.9617
##      news          online_gaming      shopping      health_nutrition
##  Min. :-0.57385  Min. :-0.4498  Min. :-0.7681  Min. :-0.57099
##  1st Qu.:-0.57385 1st Qu.:-0.4498 1st Qu.:-0.7681 1st Qu.:-0.57099
```

```

## Median :-0.57385 Median :-0.4498 Median :-0.2153 Median :-0.34858
## Mean : 0.00000 Mean : 0.0000 Mean : 0.0000 Mean : 0.00000
## 3rd Qu.:-0.09784 3rd Qu.:-0.0777 3rd Qu.: 0.3376 3rd Qu.: 0.09625
## Max. : 8.94642 Max. : 9.5968 Max. : 5.8660 Max. : 8.54794
## college_uni sports_playing cooking eco
## Min. :-0.5348 Min. :-0.6552 Min. :-0.582583 Min. :-0.6656
## 1st Qu.:-0.5348 1st Qu.:-0.6552 1st Qu.:-0.582583 1st Qu.:-0.6656
## Median :-0.1897 Median :-0.6552 Median :-0.291032 Median :-0.6656
## Mean : 0.0000 Mean : 0.0000 Mean : 0.000000 Mean : 0.0000
## 3rd Qu.: 0.1555 3rd Qu.: 0.3699 3rd Qu.: 0.000518 3rd Qu.: 0.6336
## Max. : 9.8202 Max. : 7.5456 Max. : 9.038575 Max. : 7.1294
## computers business outdoors crafts
## Min. :-0.5503 Min. :-0.6113 Min. :-0.6471 Min. :-0.6315
## 1st Qu.:-0.5503 1st Qu.:-0.6113 1st Qu.:-0.6471 1st Qu.:-0.6315
## Median :-0.5503 Median :-0.6113 Median :-0.6471 Median :-0.6315
## Mean : 0.0000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000
## 3rd Qu.: 0.2975 3rd Qu.: 0.8330 3rd Qu.: 0.1797 3rd Qu.: 0.5927
## Max. :13.0153 Max. : 8.0545 Max. : 9.2745 Max. : 7.9380
## automotive art religion beauty
## Min. :-0.6074 Min. :-0.4448 Min. :-0.57207 Min. :-0.531
## 1st Qu.:-0.6074 1st Qu.:-0.4448 1st Qu.:-0.57207 1st Qu.:-0.531
## Median :-0.6074 Median :-0.4448 Median :-0.57207 Median :-0.531
## Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.000
## 3rd Qu.: 0.1245 3rd Qu.: 0.1689 3rd Qu.:-0.04983 3rd Qu.: 0.222
## Max. : 8.9083 Max. :10.6010 Max. : 9.87273 Max. :10.012
## parenting dating school personal_fitness
## Min. :-0.60800 Min. :-0.3988 Min. :-0.6461 Min. :-0.6079
## 1st Qu.:-0.60800 1st Qu.:-0.3988 1st Qu.:-0.6461 1st Qu.:-0.6079
## Median :-0.60800 Median :-0.3988 Median :-0.6461 Median :-0.6079
## Mean : 0.000000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000
## 3rd Qu.: 0.05191 3rd Qu.: 0.1622 3rd Qu.: 0.1955 3rd Qu.: 0.2237
## Max. : 8.63074 Max. :13.0666 Max. : 8.6112 Max. : 7.2915
## fashion small_business spam
## Min. :-0.545049 Min. :-0.5441 Min. :-0.07769
## 1st Qu.:-0.545049 1st Qu.:-0.5441 1st Qu.:-0.07769
## Median :-0.545049 Median :-0.5441 Median :-0.07769
## Mean : 0.000000 Mean : 0.0000 Mean : 0.00000
## 3rd Qu.: 0.001873 3rd Qu.: 1.0736 3rd Qu.:-0.07769
## Max. : 9.299557 Max. : 9.1623 Max. :23.93529
## adult
## Min. :-0.2224
## 1st Qu.:-0.2224
## Median :-0.2224
## Mean : 0.0000
## 3rd Qu.:-0.2224
## Max. :14.1151

# Extract the centers and scales from the rescaled data (which are named attributes)
mu = attr(X,"scaled:center")
sigma = attr(X,"scaled:scale")

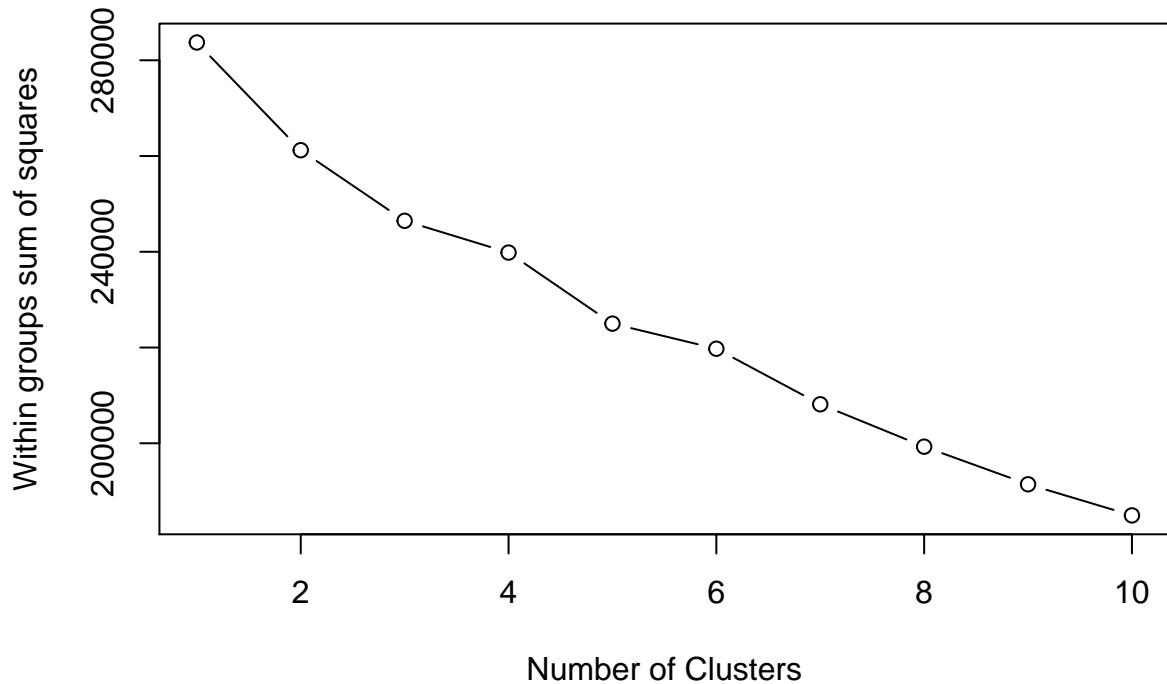
```

First we decided to make a plot of the sum of squares vs. the different choices of K to find the optimal K in which to use in our kmeans ++ model. Using the elbow method we found that 4-6 were the optimal numbers for K.

```

### finding the optimal K###
set.seed(2)
mydata <- X
wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
for (i in 2:10) wss[i] <- sum(kmeans(mydata,
                                         centers=i)$withinss)
plot(1:10, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")

```



We ran a kmeans++ model using $k=4$, $k=5$, and $k=6$. Below you can see the outputs for each one. It turns out the using $K=4$ and $K=5$, did not give us enough clusters, and that when we used $K=6$ we could see distinct clusters that differed from one another.

```

### optimal output from above shows either 3 or 6 for k so lets try both
set.seed(1)
clust5 = kmeanspp(X, k=5, nstart=25)

```

```

#cluster 3 centers
clust5$center[1]*sigma + mu #chatter, photo_sharing, not active

```

```

##          chatter current_events      travel photo_sharing
## 4.272972973    1.445945946   1.112266112   2.274220374
## uncategorized        tv_film sports_fandom      politics
## 0.729937630    1.036798337   0.977130977   1.006860707
##          food       family home_and_garden      music
## 0.778586279    0.595634096   0.444282744   0.575467775

```

```

##          news   online_gaming      shopping health_nutrition
## 0.686486486 1.159459459 1.252182952 1.067359667
## college_uni sports_playing      cooking       eco
## 1.502494802 0.549272349 0.844074844 0.388357588
## computers      business      outdoors      crafts
## 0.372557173 0.337006237 0.402079002 0.371725572
## automotive        art      religion      beauty
## 0.592515593 0.657588358 0.527442827 0.336798337
## parenting      dating      school personal_fitness
## 0.463201663 0.528690229 0.462577963 0.645322245
## fashion      small_business      spam      adult
## 0.509979210 0.284615385 0.006652807 0.417879418

```

```
clust5$center[2,]*sigma + mu #cooking, fashion, beauty, chatter, photo sharing
```

```

##          chatter current_events      travel photo_sharing
## 5.377880184 1.764976959 1.509984639 6.129032258
## uncategorized      tv_film      sports_fandom      politics
## 1.288786482 1.153609831 1.165898618 1.411674347
## food      family      home_and_garden      music
## 1.105990783 0.929339478 0.652841782 1.278033794
## news      online_gaming      shopping health_nutrition
## 1.024577573 1.528417819 2.195084485 2.152073733
## college_uni sports_playing      cooking       eco
## 2.070660522 0.964669739 10.038402458 0.589861751
## computers      business      outdoors      crafts
## 0.738863287 0.635944700 0.807987711 0.643625192
## automotive        art      religion      beauty
## 0.917050691 0.943164363 0.850998464 3.648233487
## parenting      dating      school personal_fitness
## 0.784946237 1.184331797 1.033794163 1.314900154
## fashion      small_business      spam      adult
## 5.247311828 0.543778802 0.003072197 0.439324117

```

```
clust5$center[3,]*sigma + mu #chatter, photo sharing, personal fitness, health nutrition
```

```

##          chatter current_events      travel photo_sharing
## 4.377825619 1.553283100 1.232508073 2.672766416
## uncategorized      tv_film      sports_fandom      politics
## 0.976318622 1.032292788 1.163616792 1.226049516
## food      family      home_and_garden      music
## 2.120559742 0.791173305 0.636167922 0.762109795
## news      online_gaming      shopping health_nutrition
## 1.087190527 1.196986006 1.473627557 11.907427341
## college_uni sports_playing      cooking       eco
## 1.342303552 0.696447793 3.256189451 0.913885899
## computers      business      outdoors      crafts
## 0.551130248 0.472551130 2.699677072 0.594187298
## automotive        art      religion      beauty
## 0.675995694 0.753498385 0.763186222 0.418729817
## parenting      dating      school personal_fitness
## 0.759956943 1.013993541 0.585575888 6.373519914
## fashion      small_business      spam      adult
## 0.792249731 0.290635091 0.006458558 0.425188375

```

```

clust5$center[4,]*sigma + mu #chatter, photo sharing, parenting, sports fandom, food, family

##          chatter current_events      travel photo_sharing
## 4.258598726    1.672611465    1.355414013    2.630573248
## uncategorized      tv_film sports_fandom     politics
## 0.761783439    1.115923567    5.868789809    1.168152866
##          food       family home_and_garden      music
## 4.532484076    2.485350318    0.657324841    0.757961783
##          news online_gaming      shopping health_nutrition
## 1.040764331    1.289171975    1.470063694    1.852229299
## college_uni sports_playing      cooking      eco
## 1.538853503    0.797452229    1.577070064    0.656050955
## computers        business outdoors      crafts
## 0.741401274    0.500636943    0.701910828    1.080254777
## automotive        art religion      beauty
## 1.047133758    0.898089172    5.242038217    1.077707006
## parenting         dating school personal_fitness
## 4.019108280    0.770700637    2.685350318    1.191082803
## fashion small_business      spam      adult
## 1.001273885    0.410191083    0.006369427    0.405095541

clust5$center[5,]*sigma + mu #chatter, photo_sharing, news, politics, travel,

##          chatter current_events      travel photo_sharing
## 4.536067893    1.654879774    5.588401697    2.516265912
## uncategorized      tv_film sports_fandom     politics
## 0.782178218    1.220650636    2.004243281    8.882602546
##          food       family home_and_garden      music
## 1.445544554    0.923620934    0.615275813    0.637906648
##          news online_gaming      shopping health_nutrition
## 5.241867044    1.176803395    1.380480905    1.674681754
## college_uni sports_playing      cooking      eco
## 1.673267327    0.700141443    1.261669024    0.596888260
## computers        business outdoors      crafts
## 2.473833098    0.663366337    0.919377652    0.649222065
## automotive        art religion      beauty
## 2.325318246    0.751060820    1.016973126    0.463932107
## parenting         dating school personal_fitness
## 0.936350778    1.049504950    0.708628006    1.001414427
## fashion small_business      spam      adult
## 0.656294201    0.475247525    0.008486563    0.240452617

#checking to see sum of squares
clust5$tot.withinss

## [1] 224580
clust5$betweenss

## [1] 59136.04
options(scipen=999)

set.seed(1)
clust4 = kmeanspp(X, k=4, nstart=25)
#cluster 4 centers
clust4$center[1,]*sigma + mu # food, sports fandom, religion, parenting

```

```

##      chatter current_events      travel photo_sharing
## 4.109375000 1.679687500 1.342447917 2.548177083
## uncategorized      tv_film sports_fandom politics
## 0.746093750 1.052083333 5.962239583 1.186197917
##      food      family home_and_garden music
## 4.609375000 2.519531250 0.648437500 0.726562500
##      news online_gaming shopping health_nutrition
## 1.039062500 1.272135417 1.404947917 2.182291667
## college_uni sports_playing cooking eco
## 1.454427083 0.766927083 1.733072917 0.652343750
##      computers business outdoors crafts
## 0.743489583 0.503906250 0.748697917 1.080729167
##      automotive art religion beauty
## 1.050781250 0.884114583 5.364583333 1.106770833
##      parenting dating school personal_fitness
## 4.104166667 0.664062500 2.704427083 1.394531250
##      fashion small_business spam adult
## 1.040364583 0.389322917 0.006510417 0.425781250

clust4$center[2,]*sigma + mu # not active

##      chatter current_events      travel photo_sharing
## 3.66717724 1.37221007 1.06236324 1.88402626
## uncategorized      tv_film sports_fandom politics
## 0.67242888 0.82122538 0.94332604 0.95448578
##      food      family home_and_garden music
## 0.79846827 0.55667396 0.40612691 0.47833698
##      news online_gaming shopping health_nutrition
## 0.69102845 0.93654267 0.98008753 1.52582057
## college_uni sports_playing cooking eco
## 1.15536105 0.45207877 0.92910284 0.34748359
##      computers business outdoors crafts
## 0.35536105 0.28971554 0.49059081 0.32253829
##      automotive art religion beauty
## 0.54288840 0.48927790 0.51597374 0.31947484
##      parenting dating school personal_fitness
## 0.44923414 0.42954048 0.40131291 0.86564551
##      fashion small_business spam adult
## 0.46673961 0.23216630 0.00547046 0.38052516

clust4$center[3,]*sigma + mu #travel, politics, news

##      chatter current_events      travel photo_sharing
## 4.404761905 1.656862745 5.627450980 2.445378151
## uncategorized      tv_film sports_fandom politics
## 0.782913165 1.142857143 2.042016807 8.990196078
##      food      family home_and_garden music
## 1.460784314 0.929971989 0.610644258 0.633053221
##      news online_gaming shopping health_nutrition
## 5.284313725 1.138655462 1.301120448 2.029411765
## college_uni sports_playing cooking eco
## 1.532212885 0.707282913 1.406162465 0.591036415
##      computers business outdoors crafts
## 2.476190476 0.644257703 1.001400560 0.607843137

```

```

##      automotive          art        religion        beauty
## 2.362745098 0.679271709 1.023809524 0.512605042
##    parenting         dating       school personal_fitness
## 0.960784314 1.047619048 0.722689076 1.189075630
##    fashion small_business        spam        adult
## 0.731092437 0.473389356 0.008403361 0.238095238
clust4$center[4,]*sigma + mu #cooking, fashion, shopping, health_nutrition

##      chatter current_events        travel photo_sharing
## 6.344808743 1.795628415 1.414754098 4.886885246
## uncategorized          tv_film sports_fandom politics
## 1.203825137 1.671584699 1.210928962 1.314754098
##    food           family home_and_garden music
## 1.520765027 0.910382514 0.718032787 1.179234973
##    news online_gaming        shopping health_nutrition
## 0.968852459 1.889617486 2.439344262 5.539344262
## college_uni sports_playing        cooking eco
## 2.580327869 1.026229508 5.010382514 0.834426230
## computers business        outdoors crafts
## 0.630054645 0.636612022 1.440983607 0.725683060
## automotive          art religion beauty
## 0.855737705 1.263934426 0.778688525 1.574863388
##    parenting         dating       school personal_fitness
## 0.749180328 1.301639344 0.887431694 3.086338798
##    fashion small_business        spam        adult
## 2.404918033 0.520765027 0.008196721 0.515300546

#checking to see sum of squares
clust4$tot.withinss

## [1] 234995.5
clust4$betweenss

## [1] 48720.5

set.seed(1)
clust6 = kmeanspp(X, k=6, nstart=25)
#cluster 6 centers
clust6$center[1,]*sigma + mu #chatter, photo sharing

##      chatter current_events        travel photo_sharing
## 4.328492849 1.444664466 1.099229923 2.296149615
## uncategorized          tv_film sports_fandom politics
## 0.728272827 1.003080308 0.970517052 1.010341034
##    food           family home_and_garden music
## 0.769416942 0.573157316 0.440044004 0.562596260
##    news online_gaming        shopping health_nutrition
## 0.692409241 0.588778878 1.278987899 1.091529153
## college_uni sports_playing        cooking eco
## 0.908910891 0.421122112 0.862926293 0.389658966
## computers business        outdoors crafts
## 0.373817382 0.339053905 0.401760176 0.363256326
## automotive          art religion beauty
## 0.580858086 0.622002200 0.526732673 0.354015402
##    parenting         dating       school personal_fitness

```

```

##      0.458525853      0.543234323      0.477227723      0.659845985
##      fashion    small_business          spam           adult
##      0.514851485      0.277667767      0.006820682      0.416501650

```

```
clust6$center[2,]*sigma + mu #online gaming, college universities
```

```

##      chatter   current_events       travel photo_sharing
##      4.482517483      1.487179487      1.573426573      2.818181818
##      uncategorized      tv_film sports_fandom politics
##      0.913752914      1.699300699      1.335664336      1.307692308
##      food      family home_and_garden music
##      1.247086247      1.079254079      0.613053613      0.955710956
##      news      online_gaming shopping health_nutrition
##      0.797202797      9.694638695      1.365967366      1.783216783
##      college_uni sports_playing cooking eco
##      10.564102564      2.613053613      1.482517483      0.489510490
##      computers      business outdoors crafts
##      0.585081585      0.417249417      0.659673660      0.603729604
##      automotive      art religion beauty
##      0.909090909      1.233100233      0.811188811      0.442890443
##      parenting      dating school personal_fitness
##      0.675990676      0.748251748      0.512820513      1.025641026
##      fashion    small_business      spam           adult
##      0.899766900      0.461538462      0.009324009      0.445221445

```

```
clust6$center[3,]*sigma + mu #health nutrition, personal fitness
```

```

##      chatter   current_events       travel photo_sharing
##      4.354729730      1.559684685      1.244369369      2.654279279
##      uncategorized      tv_film sports_fandom politics
##      0.966216216      0.984234234      1.163288288      1.255630631
##      food      family home_and_garden music
##      2.129504505      0.773648649      0.636261261      0.739864865
##      news      online_gaming shopping health_nutrition
##      1.106981982      0.841216216      1.458333333      12.010135135
##      college_uni sports_playing cooking eco
##      0.933558559      0.604729730      3.281531532      0.918918919
##      computers      business outdoors crafts
##      0.561936937      0.470720721      2.740990991      0.588963964
##      automotive      art religion beauty
##      0.663288288      0.740990991      0.762387387      0.424549550
##      parenting      dating school personal_fitness
##      0.761261261      1.038288288      0.596846847      6.438063063
##      fashion    small_business      spam           adult
##      0.800675676      0.293918919      0.006756757      0.416666667

```

```
clust6$center[4,]*sigma + mu # sports fandom, parenting, religion
```

```

##      chatter   current_events       travel photo_sharing
##      4.238219895      1.679319372      1.349476440      2.629581152
##      uncategorized      tv_film sports_fandom politics
##      0.752617801      1.090314136      5.888743455      1.166230366
##      food      family home_and_garden music
##      4.565445026      2.492146597      0.643979058      0.744764398
##      news      online_gaming shopping health_nutrition
##      1.040575916      1.006544503      1.469895288      1.854712042

```

```

##      college_uni sports_playing      cooking      eco
## 1.229057592     0.743455497  1.587696335 0.659685864
##      computers       business      outdoors      crafts
## 0.731675393     0.502617801  0.691099476 1.085078534
##      automotive        art      religion      beauty
## 1.049738220     0.870418848  5.252617801 1.090314136
##      parenting        dating      school personal_fitness
## 4.049738220     0.776178010  2.698952880 1.191099476
##      fashion small_business        spam      adult
## 1.005235602     0.404450262  0.005235602 0.409685864

```

`clust6$center[5,]*sigma + mu # politics, travel, news`

```

##      chatter current_events      travel photo_sharing
## 4.548387097     1.667155425  5.612903226 2.541055718
## uncategorized      tv_film sports_fandom      politics
## 0.775659824     1.199413490  2.014662757 8.960410557
##      food      family home_and_garden      music
## 1.441348974     0.913489736  0.611436950 0.640762463
##      news online_gaming      shopping health_nutrition
## 5.318181818     0.828445748  1.379765396 1.639296188
##      college_uni sports_playing      cooking      eco
## 1.318181818     0.629032258  1.259530792 0.593841642
##      computers       business      outdoors      crafts
## 2.473607038     0.670087977  0.916422287 0.640762463
##      automotive        art      religion      beauty
## 2.347507331     0.718475073  1.030791789 0.473607038
##      parenting        dating      school personal_fitness
## 0.947214076     1.068914956  0.725806452 1.000000000
##      fashion small_business        spam      adult
## 0.668621701     0.483870968  0.005865103 0.236070381

```

`clust6$center[6,]*sigma + mu # fashion, cooking, beauty, photo sharing`

```

##      chatter current_events      travel photo_sharing
## 4.996515679     1.778745645  1.494773519 6.118466899
## uncategorized      tv_film sports_fandom      politics
## 1.296167247     1.085365854  1.174216028 1.442508711
##      food      family home_and_garden      music
## 1.081881533     0.918118467  0.639372822 1.261324042
##      news online_gaming      shopping health_nutrition
## 1.059233449     1.066202091  2.078397213 2.280487805
##      college_uni sports_playing      cooking      eco
## 1.538327526     0.817073171  10.811846690 0.578397213
##      computers       business      outdoors      crafts
## 0.733449477     0.621951220  0.824041812 0.639372822
##      automotive        art      religion      beauty
## 0.904181185     0.947735192  0.869337979 3.878048780
##      parenting        dating      school personal_fitness
## 0.822299652     0.991289199  1.001742160 1.351916376
##      fashion small_business        spam      adult
## 5.564459930     0.506968641  0.003484321 0.437282230

```

`#checking to see sum of squares`

`clust6$tot.withinss`

```
## [1] 214481.4  
clust6$betweenss
```

```
## [1] 69234.64
```

Using k=6, we found the following clusters to represent the following market segments:

1. Cluster 1

- Focused around those who just used a lot of chatter or photo sharing, and did not really focus on any specific topic when tweeting. They also did not seem to be using twitter a lot since their counts were low in every topic.

2. Cluster 2

- Focused around people who mentioned online gaming or college universities a lot. We figures that this could be a young male population consisting of 16-22 year olds who are active on twitter.

3. Cluster 3

- Focused around people who talked a lot about health nutrition and personal fitness. This customer segment could possibly be young adults or adults who are very into fitness and staying healthy, and who regulary attend the gym and eat nutrious foods.

4. Cluster 4

- Focused around sports fandom, parenting and religion. This customer segment more liekly than not represents the parents of families who have children, therefore consisting of an older adult crowd.

5. Cluster 5

- Focused around those who mentioned politics, travel, automotive, computers and news. This customer segment probably consits of older men who are educated and probably have more money.

6. Cluster 6

- Focused on cooking, fashion, and beauty. This cusomter segment probably represents mothers or adult/young adult women.

Below we plotted some of the key variables that represent every customer segment using K=6.

```
# qplot is in the ggplot2 library  
qplot(chatter, photo_sharing,data=socialmarketing, size=I(3), color=factor(clust6$cluster))
```

