Exercises 1

Wenying Hu, Xueru Rong, Zhaoshun (Jason) Su

August 9, 2018

Probability practice

Part A

$$P(RC) = 0.3$$
$$P(TC) = 1 - 0.3 = 0.7$$
$$P(Yes) = 0.65$$
$$P(No) = 0.35$$
$$P(Yes\ and\ RC) = P(Yes|RC) \times P(RC) = 0.5 * 0.3 = 0.15$$
$$P(No\ and\ RC) = P(RC) - P(Yes\ and\ RC) = 0.3 - 0.15 = 0.15$$
$$P(Yes|TC) = \frac{P(Yes\ and\ TC)}{P(TC)} = \frac{P(Yes) - P(Yes\ and\ RC)}{0.7} = \frac{0.65 - 0.15}{0.7} = 0.714$$

Part B

$$P(Pos|D) = 0.993$$
$$P(Neg|D') = 0.9999$$
$$P(Pos|D') = 1 - P(N|D') = 0.0001$$
$$P(D) = 0.000025$$
$$P(D') = 1 - P(D) = 0.999975$$
$$P(D|Pos) = \frac{P(D\ and\ Pos)}{P(Pos)} = \frac{P(D\ and\ Pos)}{P(D\ and\ Pos) + P(D'\ and\ Pos)} = \frac{P(Pos|D) \times P(D)}{P(Pos|D) \times P(D) + P(Pos|D') \times P(D')}$$
$$= \frac{0.993 \times 0.000025}{0.993 \times 0.000025 + 0.0001 \times 0.999975} = 0.1985$$

Problems in implementing a universal testing policy:

1. The probability of having the disease when getting positive result is actually not high. The result might cause unnecessary panic;
2. All assumed probabilities need to be very accurate which might be hard to achieve in reality.

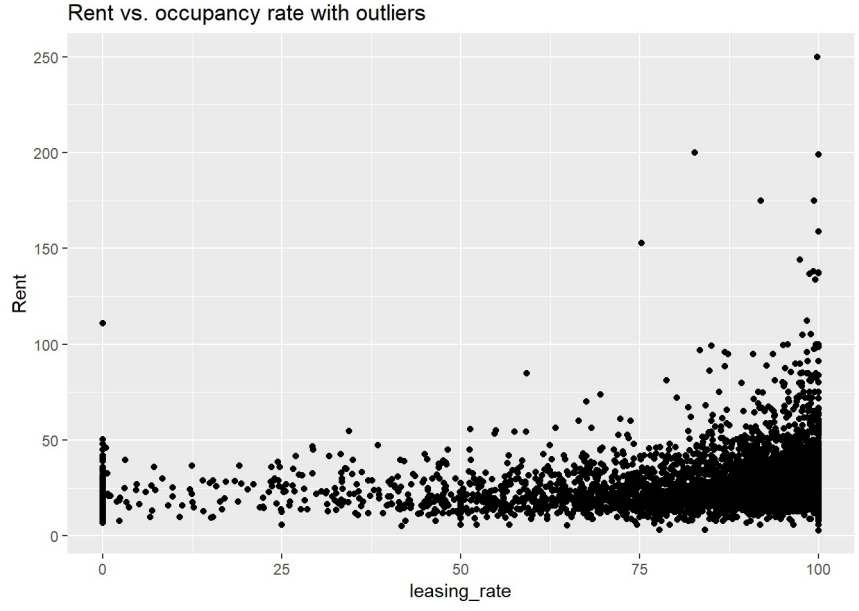
Exploratory analysis: green buildings

```
library(ggplot2)
dat=read.csv("greenbuildings.csv")
```

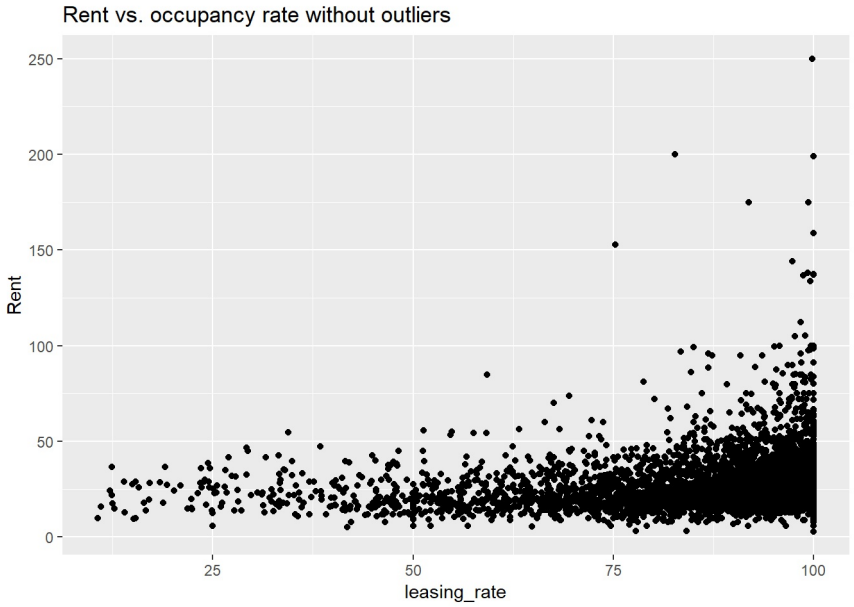
1. Is it reasonable to remoeve low occupied outliers?

“I decided to remove these buildings from consideration, on the theory that these buildings might have something weird going on with them, and could potentially distort the analysis.”

```
par(mfrow=c(1,2))
greenbuilding = dat[,-c(12,13)]
greenbuilding$green_rating = factor(greenbuilding$green_rating)
greenbuilding$cluster = factor(greenbuilding$cluster)
ggplot(greenbuilding, aes(x=leasing_rate, y=Rent)) + geom_point()+ggtitle("Rent vs. occupancy rate with outlier
s")
```



```
mask = which(greenbuilding$leasing_rate > 10)
greenbuilding = greenbuilding[mask,]
ggplot(greenbuilding, aes(x=leasing_rate, y=Rent)) + geom_point()+ggtitle("Rent vs. occupancy rate without outl
iers")
```

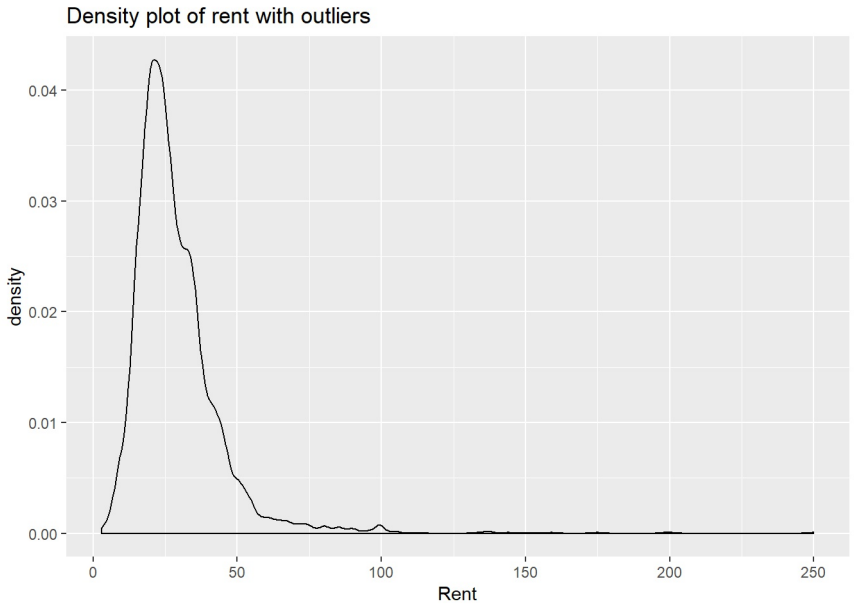


From the scatterplot we can see that there is a specific pattern when x is small. Therefore, it is correct to remove the outliers.

2. Should we use median or mean?

"I used the median rather than the mean, because there were still some outliers in the data, and the median is a lot more robust to outliers."

```
ggplot(greenbuilding,aes(x=Rent))+geom_density()+ggtitle("Density plot of rent with outliers")
```




```
mask2 = which(greenbuilding$Rent > 75)
paste("The number of outliers is",length(mask2))
```

```
## [1] "The number of outliers is 112"
```

```
outliers = greenbuilding[mask2,]
remain = greenbuilding[-mask2,]
```

We decided to use median or mean mainly based on the distribution and size of the dataset. From the first plot, we could see that the number of outliers is not small and the density plot is right-skewed. Therefore, it might be better to use median to measure the central tendency of the dataset. However, without the outliers the distribution of the dataset is quiet normally distributed which might be caused by its

large size. So let's try to dig into these outliers to check whether it is reasonable to remove them. We could also see that if we are able to remove them, the distribution is much more normally ditributed.

```
library(gridExtra)

paste("The mean of rent of outliers is ",mean(outliers$Rent),", while the mean of rent of all buildings is ",me
an(greenbuilding$Rent))
```

```
## [1] "The mean of rent of outliers is  103.209285714286 , while the mean of rent of all buildings is  28.5858
458132569"
```

```
paste("The mean of stories of outliers is ",mean(outliers$stories),", while the mean of stories of all building
s is ",mean(greenbuilding$stories))
```

```
## [1] "The mean of stories of outliers is  29.2946428571429 , while the mean of stories of all buildings is  1
3.8299257715848"
```

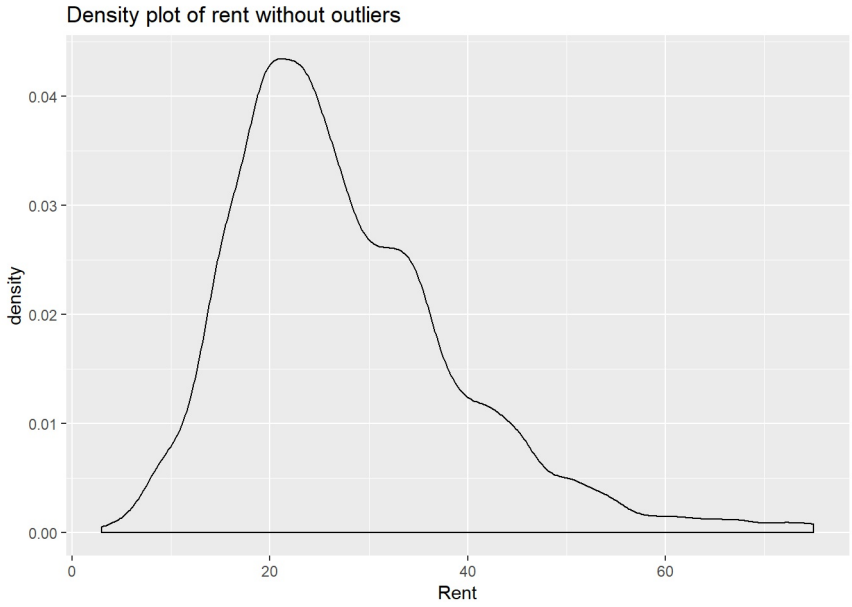
```
paste("The number of green buildings among outliers is ",sum(outliers$green_rating=="1"),", while the number of
 outliers is ",dim(outliers)[1],". The fraction is ",sum(outliers$green_rating=="1")/dim(outliers)[1])
```

```
## [1] "The number of green buildings among outliers is  7 , while the number of outliers is  112 . The fractio
n is  0.0625"
```

```
rent_set = data.frame(ca=c("Outliers","All"),
                      rent=c(mean(outliers$Rent),mean(greenbuilding$Rent)))
p1=ggplot(data=rent_set,aes(x=ca,y=rent,fill=ca))+geom_bar(stat="identity")+ggtitle("Comparison of Rent")

story_set = data.frame(ca=c("Outliers","All"),
                       story=c(mean(outliers$stories),mean(greenbuilding$stories)))
p2=ggplot(data=story_set,aes(x=ca,y=story,fill=ca))+geom_bar(stat="identity")+ggtitle("Comparison of Story")

number_set = data.frame(ca=c("Green","Non Green"),
                        number=c(sum(outliers$green_rating=="1"),dim(outliers)[1]-sum(outliers$green_rating=="1")
))
p3=ggplot(number_set,aes(x="",y=number,fill=ca))+geom_bar(width = 1, stat = "identity")+coord_polar("y", start=
0)+ggtitle("GB Among Outliers")

grid.arrange(p1,p2,p3, ncol=3)
```

From the graphs, we can see that first, most of these outliers have significantly higher rent than the average level. And then most of them have specifically high stories level which might be the reason of extremely high rent. Considering we are estimating for a 15-story building. These outliers might be less valuable for analysis. In addition, among these outliers, there exist few green buildings, which means we are not able to compare green and non-green building among these outleirs.In conclusion, we might not miss insight even if we remove these outliers.

```
ggplot(remain,aes(x=Rent))+geom_density()+ggtitle("Density plot of rent without outliers")
```

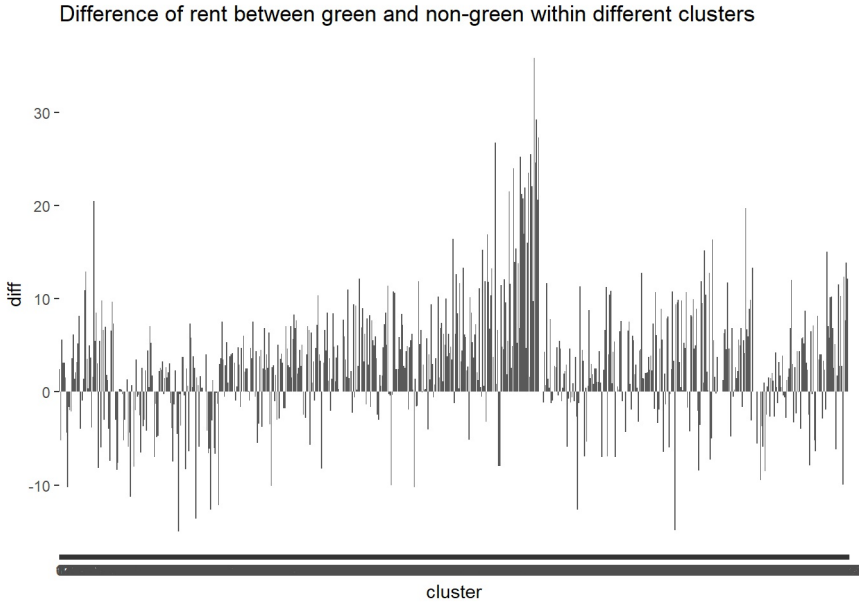


After removing these outliers, the distribution of rent approximates to normal distribution and the data size is still large enough. In this case, we are able to use “mean” to estimate central tendency in later analysis.

3. Should we just compare the mean rent of buildings from all areas?

“The median market rent in the non-green buildings was $25 per square foot per year, while the median market rent in the green buildings was $27.60 per square foot per year”

```
library(dplyr)

mean_rent_green = remain %>% group_by(cluster) %>%
  summarise(diff=mean(Rent[green_rating==1])-mean(Rent[green_rating==0]))

ggplot(data=mean_rent_green, aes(x=cluster, y=diff)) +
  geom_bar(stat="identity",position=position_dodge())+ggtitle("Difference of rent between green and non-green w
ithin different clusters")
```



From the result we can see that the difference of median price of green and non-green buildings vary a lot. Therefore, we need to find out which cluster this building belongs to and estimate the price again based on its cluster.

4. Future Estimation

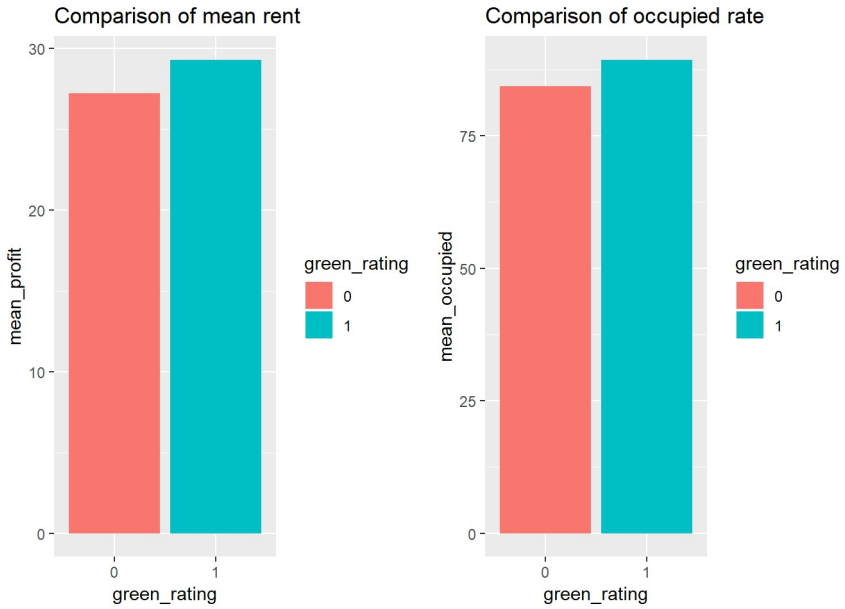
“The median market rent in the non-green buildings was $25 per square foot per year, while the median market rent in the green buildings was $27.60 per square foot per year: about $2.60 more per square foot.”

Instead of revenue, we might be more interested in the future profit, which means we should also consider about the cost and the occupancy rate.

```
remain["profit"]=remain["Rent"]-remain["Gas_Costs"]-remain["Electricity_Costs"]
remain=remain[,-c(19,20)]

profit_set = remain %>% group_by(green_rating) %>% summarise(mean_profit=mean(profit),mean_occupied=mean(leasing_rate))
p1=ggplot(data=profit_set,aes(x=green_rating,y=mean_profit,fill=green_rating))+geom_bar(stat="identity")+ggtitle("Comparison of mean rent")
p2=ggplot(data=profit_set,aes(x=green_rating,y=mean_occupied,fill=green_rating))+geom_bar(stat="identity")+ggtitle("Comparison of occupied rate")
profit_set
```

```
## # A tibble: 2 x 3
##   green_rating mean_profit mean_occupied
##   <fct>              <dbl>         <dbl>
## 1 0                   27.3          84.3
## 2 1                   29.3          89.3
```

```
grid.arrange(p1,p2, ncol=2)
```



Based on the results above, a normal building might gain an additional

$$250000 * (29.3163 * 0.8935 - 27.2548 * 0.8429) = 805261$$

dollars of extra revenue per year if we buil the green building.

Therefore, we might be able to recuperate the extra $5 million cost in around 6.2 years. Since the building will be able to earn rents for more than 30 years, “going green” is probably a good investment.

Bootstrapping

Setting up the dataset including these five ETFs from 2007-01-01

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

#Import the needed stocks data from 2007-01-01
stocks_list = c("SPY","TLT","LQD","EEM","VNQ")
getSymbols(stocks_list,from="2007-01-01")
```

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

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

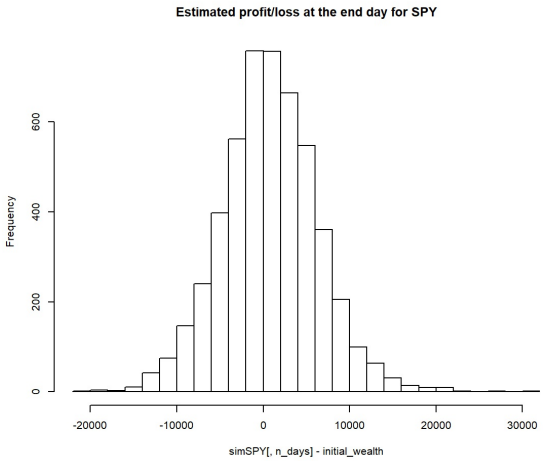
Estimate the risk/return properties of the five major asset classes. Here we just show the process of handling the SPY because the processes of other ETFs are the same. We will just store the result and display them together in the latter part.

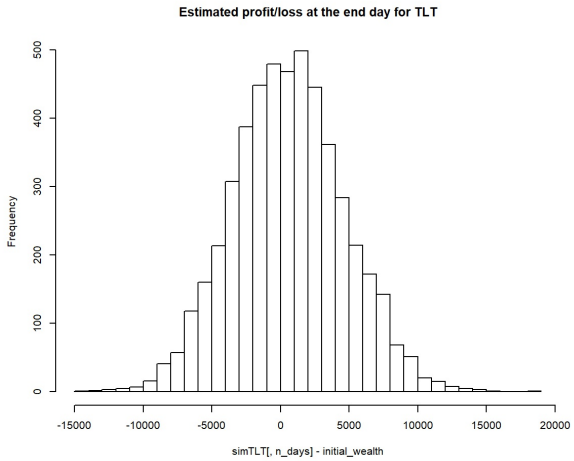
```
set.seed("666666")

all_returns = ClCl(SPYa)

all_returns = as.matrix(na.omit(all_returns))

initial_wealth = 100000
simSPY = foreach(i=1:5000, .combine='rbind') %do% {
    total_wealth = initial_wealth
    holdings = total_wealth
    n_days = 20
    wealthtracker = rep(0, n_days)
    for(today in 1:n_days) {
        return.today = resample(all_returns,1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
    }
    wealthtracker
}



# Calculate 5% value at risk
SPYVar=quantile(simSPY[,n_days], 0.05) - initial_wealth
SPYprofit = quantile(simSPY[,n_days], 0.90) - initial_wealth
SPYmean=mean(simSPY[,n_days])
```

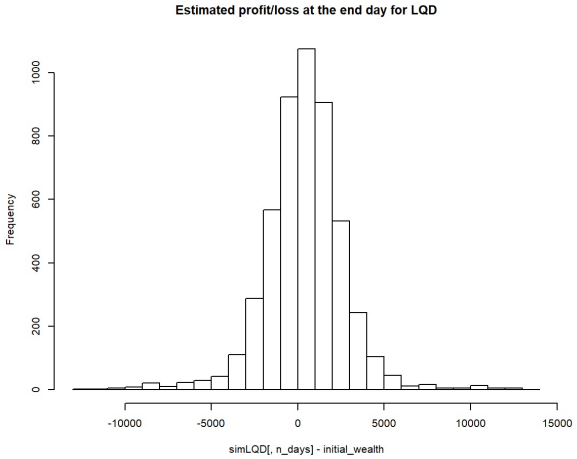
```
t <- matrix(c(SPYVar,SPYprofit,TLTVar,TLTprofit,LQDVar,LQDprofit,EEMVar,EEMprofit,VNQVar,VNQprofit),ncol=2,byro
w=TRUE)
colnames(t) <- c("5% Value at Risk","The top 10% potential profit")
rownames(t) <- c("SPY","TLT","LQD","EEM","VNQ")
t <- as.table(t)
t
```

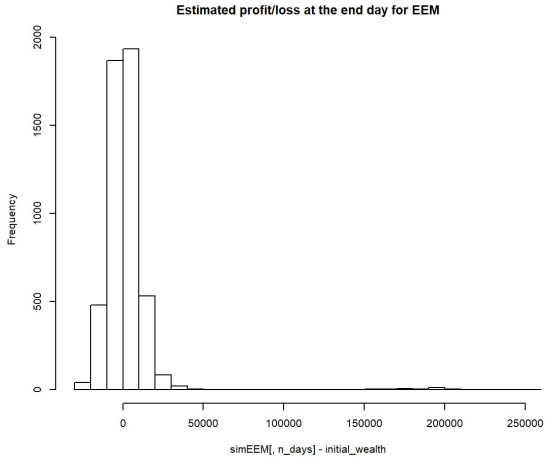
```
##     5% Value at Risk The top 10% potential profit
## SPY        -8388.289                    7527.351
## TLT        -5999.653                    5939.240
## LQD        -3028.365                    2872.259
## EEM       -13558.980                   11992.560
## VNQ       -13729.160                   12246.862
```

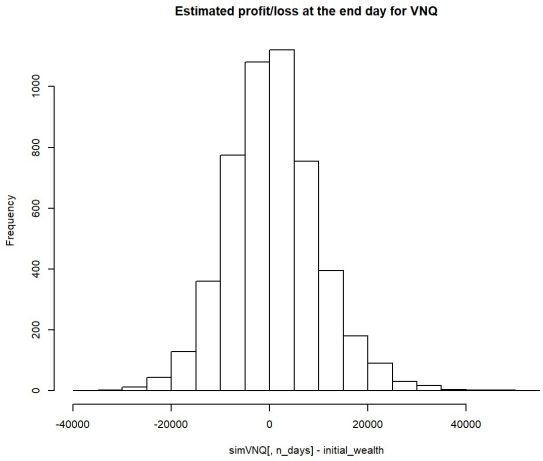
```
par(mfrow=c(3,2))
hist(simSPY[,n_days]- initial_wealth, breaks=30,main="Estimated profit/loss at the end day for SPY")
hist(simTLT[,n_days]- initial_wealth, breaks=30,main="Estimated profit/loss at the end day for TLT")
hist(simLQD[,n_days]- initial_wealth, breaks=30,main="Estimated profit/loss at the end day for LQD")
hist(simEEM[,n_days]- initial_wealth, breaks=30,main="Estimated profit/loss at the end day for EEM")
hist(simVNQ[,n_days]- initial_wealth, breaks=30,main="Estimated profit/loss at the end day for VNQ")

par(mfrow=c(3,3))
```



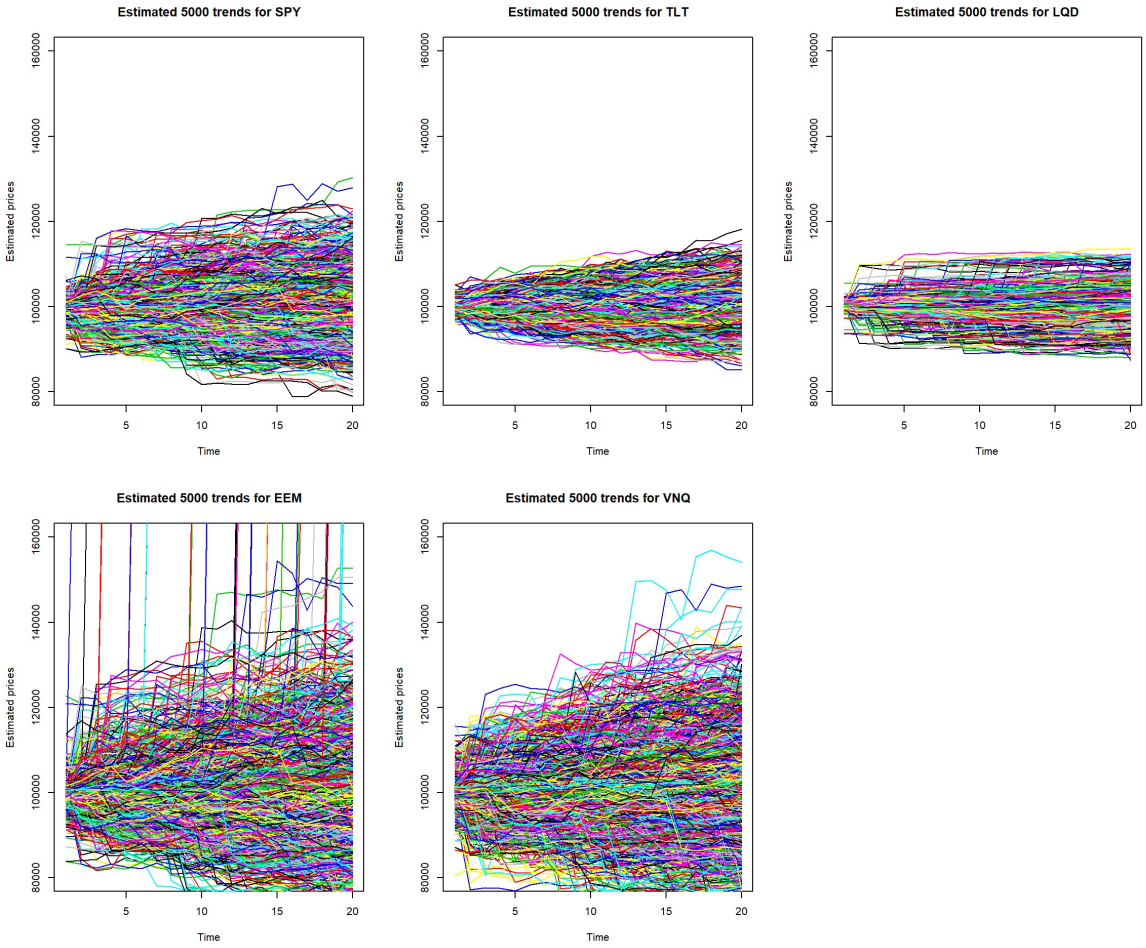








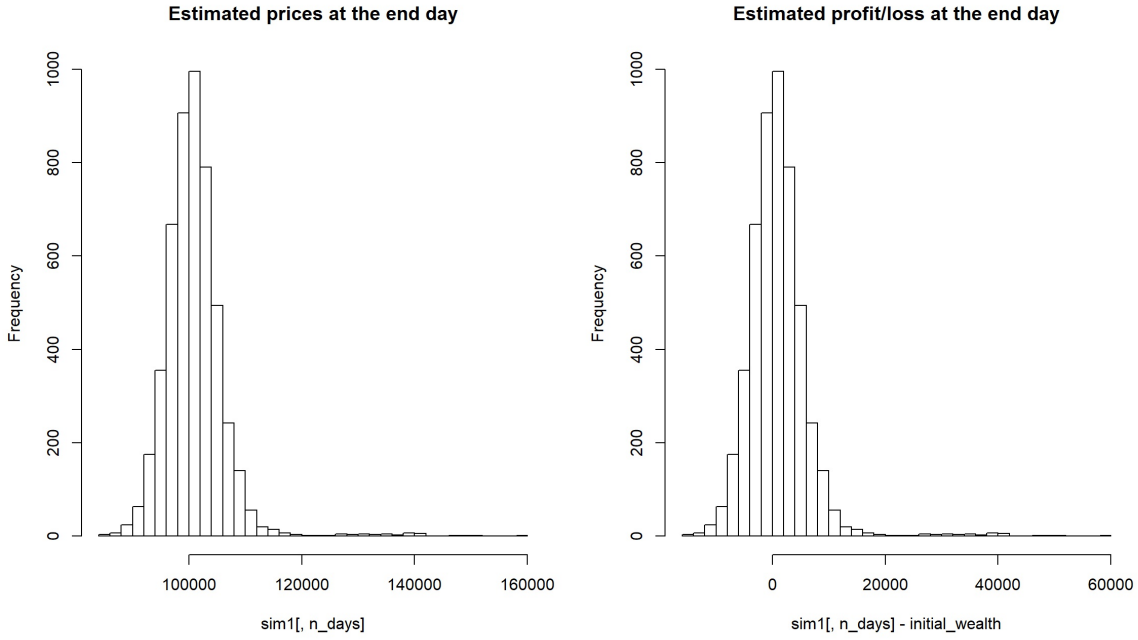

```
x=1:n_days
plot(x,simSPY[1,],xlab="Time",ylab="Estimated prices",type = "l",ylim=c(80000,160000),main="Estimated 5000 tren
ds for SPY")
for(i in 2:5000){
  lines(x,simSPY[i,],col=i)
}

plot(x,simTLT[1,],xlab="Time",ylab="Estimated prices",type = "l",ylim=c(80000,160000),main="Estimated 5000 tren
ds for TLT")
for(i in 2:5000){
  lines(x,simTLT[i,],col=i)
}

plot(x,simLQD[1,],xlab="Time",ylab="Estimated prices",type = "l",ylim=c(80000,160000),main="Estimated 5000 tren
ds for LQD")
for(i in 2:5000){
  lines(x,simLQD[i,],col=i)
}
plot(x,simEEM[1,],xlab="Time",ylab="Estimated prices",type = "l",ylim=c(80000,160000),main="Estimated 5000 tren
ds for EEM")
for(i in 2:5000){
  lines(x,simEEM[i,],col=i)
}
plot(x,simVNQ[1,],xlab="Time",ylab="Estimated prices",type = "l",ylim=c(80000,160000),main="Estimated 5000 tren
ds for VNQ")
for(i in 2:5000){
  lines(x,simVNQ[i,],col=i)
}
```



Observing the distribution of estimated prices and potential trends for these five ETFs, we could conclude that the EEM and VNQ are much more risky than other three ETFs. Among these three, LQD is the safest one with the smalleset variance, followed by TLT. And SPY's variance is the median of these five ETFs.

The even split

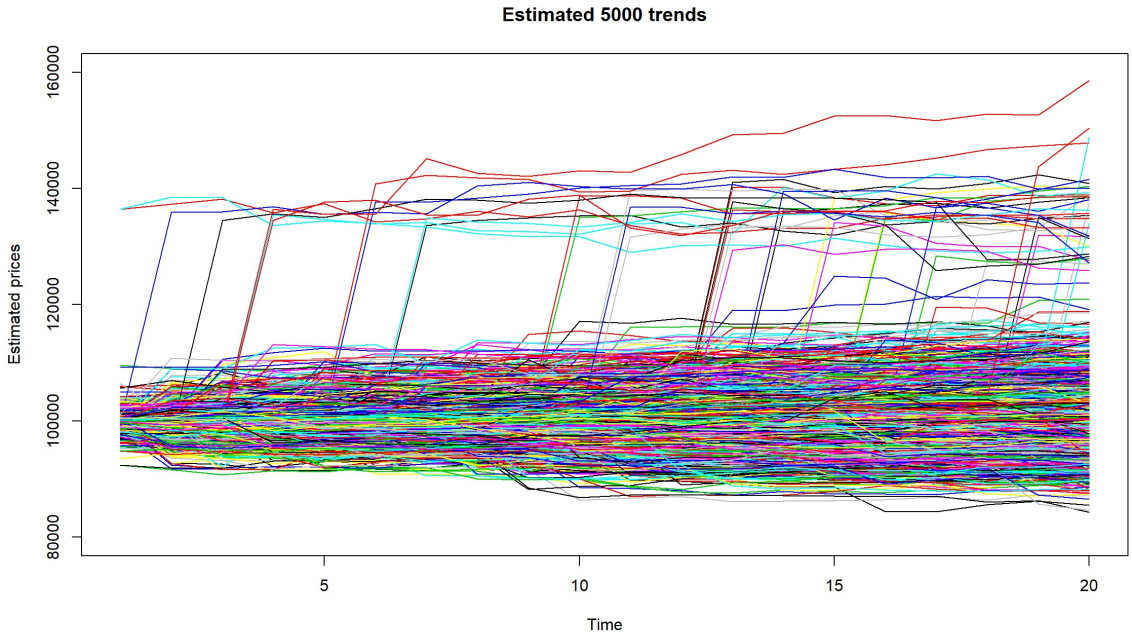
```
set.seed("666666")

all_returns = cbind(    ClCl(SPYa),
                                ClCl(TLTa),
                                ClCl(LQDa),
                                ClCl(EEMa),
                                ClCl(VNQa))

all_returns = as.matrix(na.omit(all_returns))

initial_wealth = 100000
sim1 = foreach(i=1:5000, .combine='rbind') %do% {
    total_wealth = initial_wealth
    weights = c(0.2, 0.2, 0.2, 0.2, 0.2)
    holdings = weights * total_wealth
    n_days = 20
    wealthtracker = rep(0, n_days)
    for(today in 1:n_days) {
        return.today = resample(all_returns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
    }
    wealthtracker
}

par(mfrow=c(1,2))

hist(sim1[,n_days], 30, main="Estimated prices at the end day")

#profit/loss
hist(sim1[,n_days]- initial_wealth, breaks=30,main="Estimated profit/loss at the end day")
```




```
# Calculate 5% value at risk
evenmean=mean(sim1[,n_days])
evenVar=quantile(sim1[,n_days], 0.05) - initial_wealth
paste("The 5% value at risk is ",evenVar)
```

```
## [1] "The 5% value at risk is  -6167.09355643528"
```

```
evenprofit = quantile(sim1[,n_days], 0.90) - initial_wealth

par(mfrow=c(1,1))

x=1:n_days
plot(x,sim1[1,],xlab="Time",ylab="Estimated prices",type = "l",ylim=c(80000,160000),main="Estimated 5000 trends
")
for(i in 2:5000){
  lines(x,sim1[i,],col=i)
}
```

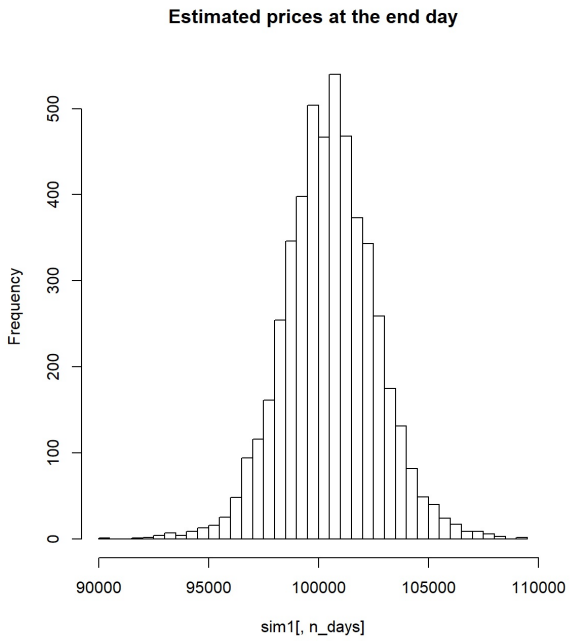


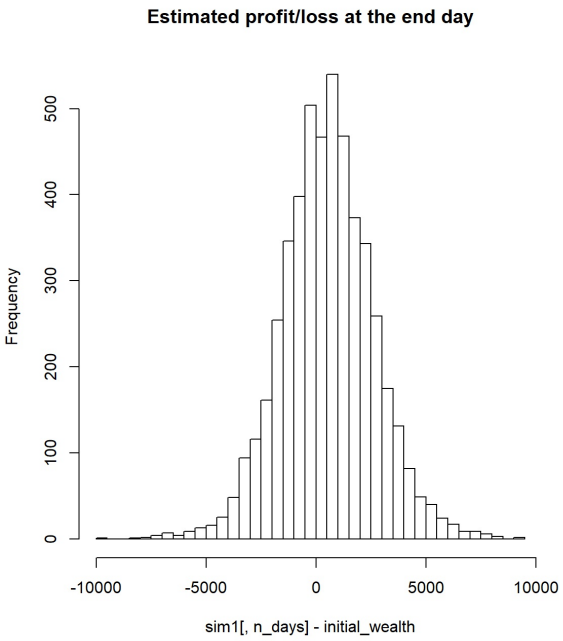
From the graphs we can see that the even portfolio's distribution is similar to the distribution of SPY, the median of these five ETFs. And it has more outliers with high prices, which might be caused by the investment in EEM and VNQ.

The safer choice

From the results, we can see that SPY,TLT and LQD, especially LQD, are relatively safer choices. Therefore, we allocate 0.2,0.3,0.5 to SPY,TLT and LQD

```
set.seed("666666")

all_returns = cbind(    ClCl(SPYa),
                                ClCl(TLTa),
                                ClCl(LQDa))

all_returns = as.matrix(na.omit(all_returns))

initial_wealth = 100000
sim1 = foreach(i=1:5000, .combine='rbind') %do% {
    total_wealth = initial_wealth
    weights = c(0.2, 0.3, 0.5)
    holdings = weights * total_wealth
    n_days = 20
    wealthtracker = rep(0, n_days)
    for(today in 1:n_days) {
        return.today = resample(all_returns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
    }
    wealthtracker
}

par(mfrow=c(1,2))

hist(sim1[,n_days], 30, main="Estimated prices at the end day")

#profit/loss
hist(sim1[,n_days]- initial_wealth, breaks=30,main="Estimated profit/loss at the end day")
```

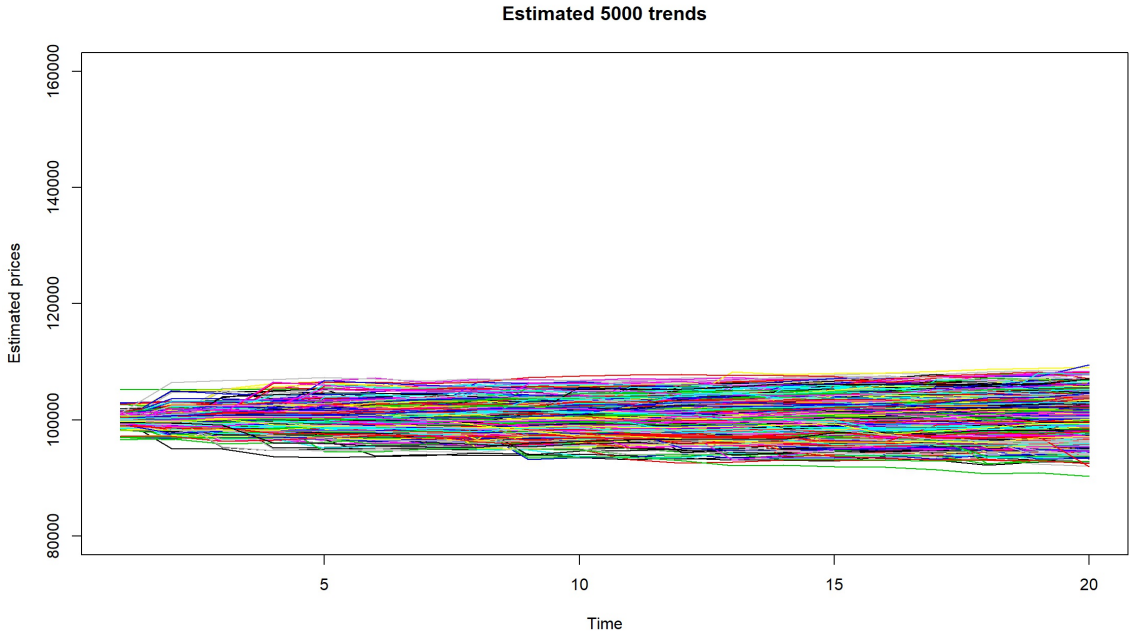





```
# Calculate 5% value at risk
safermean=mean(sim1[,n_days])
saferVar=quantile(sim1[,n_days], 0.05) - initial_wealth
paste("The 5% value at risk is ",saferVar)
```

```
## [1] "The 5% value at risk is  -2870.62687328462"
```

```
saferprofit = quantile(sim1[,n_days], 0.90) - initial_wealth

par(mfrow=c(1,1))

x=1:n_days
plot(x,sim1[1,],xlab="Time",ylab="Estimated prices",type = "l",ylim=c(80000,160000),main="Estimated 5000 trends
")
for(i in 2:5000){
  lines(x,sim1[i,],col=i)
}
```

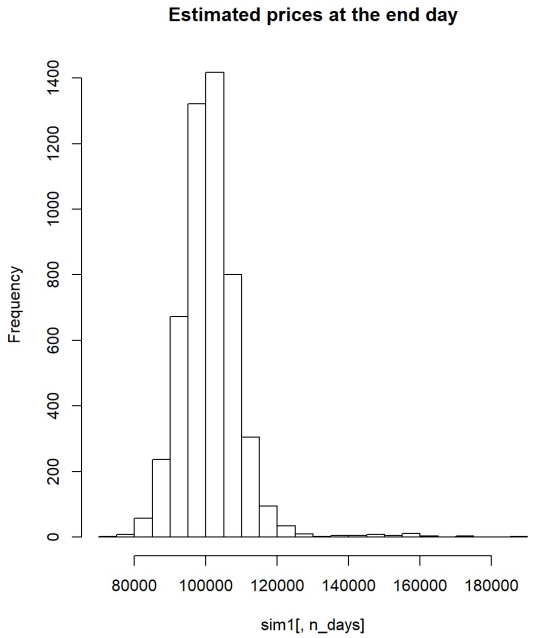


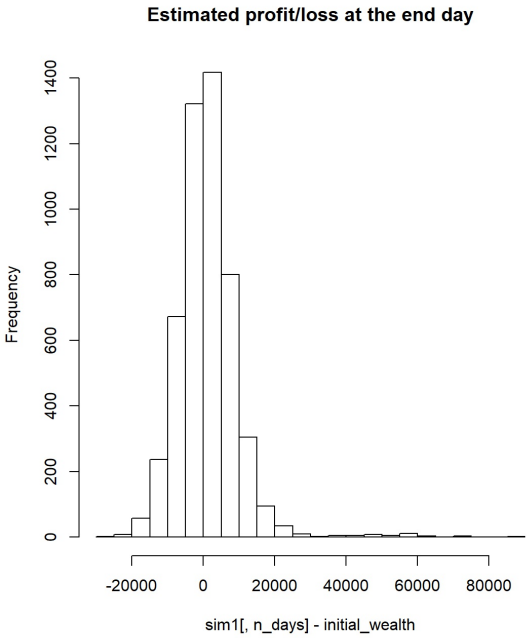
From the graphs, we can see the the safer portfolio has much smaller variance even than TLT, just a bit larger than LQD. And it has less outliers with low or high prices than LQD. The probability of significant loss is small but the probability of high return is also small.

The more aggressive choice

From the result, we can see that EEM and VNQ are more risky choices, but we still want to keep the SPY to control the risk within acceptable range. Therefore, we allocate 0.4,0.3,0.3 to SPY,EEM and VNQ.

```
set.seed("666666")

all_returns = cbind(    ClCl(SPYa),
                                ClCl(EEMa),
                                ClCl(VNQa))

all_returns = as.matrix(na.omit(all_returns))

initial_wealth = 100000
sim1 = foreach(i=1:5000, .combine='rbind') %do% {
    total_wealth = initial_wealth
    weights = c(0.4,0.3,0.3)
    holdings = weights * total_wealth
    n_days = 20
    wealthtracker = rep(0, n_days)
    for(today in 1:n_days) {
        return.today = resample(all_returns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
    }
    wealthtracker
}

par(mfrow=c(1,2))

hist(sim1[,n_days], 30, main="Estimated prices at the end day")

#profit/loss
hist(sim1[,n_days]- initial_wealth, breaks=30,main="Estimated profit/loss at the end day")
```

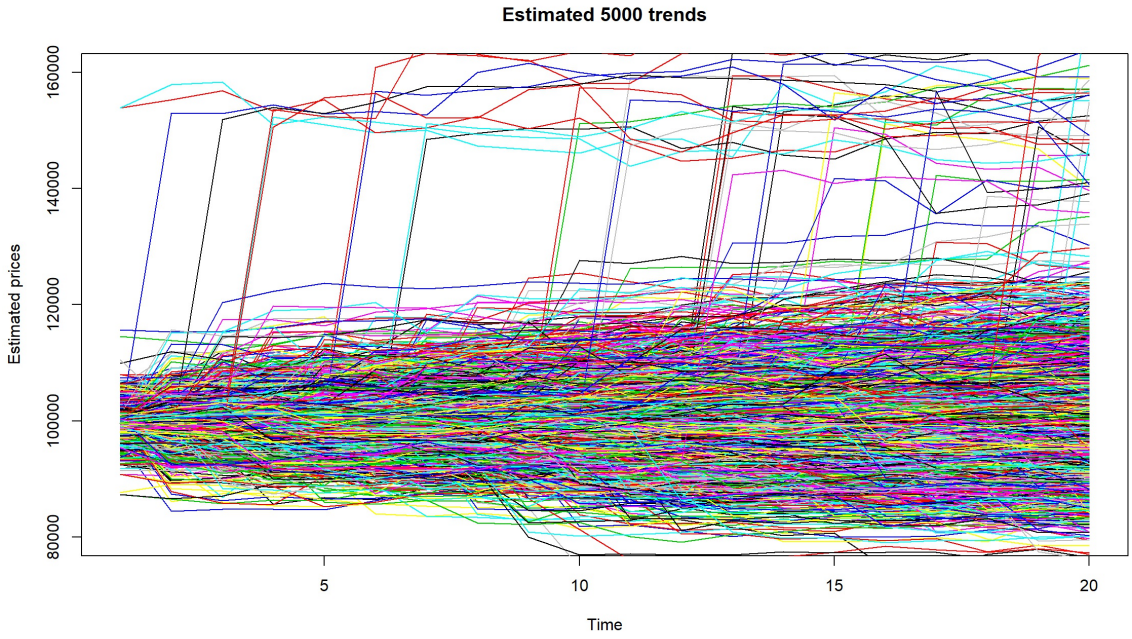





```
# Calculate 5% value at risk
aggmean=mean(sim1[,n_days])
aggVar=quantile(sim1[,n_days], 0.05) - initial_wealth
paste("The 5% value at risk is ",aggVar)
```

```
## [1] "The 5% value at risk is  -10659.5986819229"
```

```
aggprofit = quantile(sim1[,n_days], 0.90) - initial_wealth

par(mfrow=c(1,1))

x=1:n_days
plot(x,sim1[1,],xlab="Time",ylab="Estimated prices",type = "l",ylim=c(80000,160000),main="Estimated 5000 trends
")
for(i in 2:5000){
  lines(x,sim1[i,],col=i)
}
```



From the graphs, we can see that this portfolio has much larger variance.It has chance to gain significant high return, but it is also more risky than other portfolios.

Comparison of all portfolios for decision making

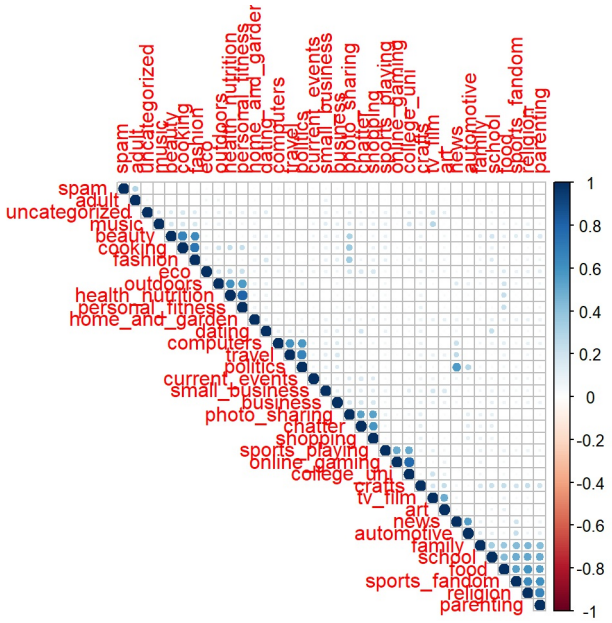
```
t <- matrix(c(SPYVar,SPYmean,SPYprofit,TLTVar,TLTmean,TLTprofit,LQDVar,LQDmean,LQDprofit,EEMVar,EEMmean,EEMprof
it,VNQVar,VNQmean,VNQprofit,evenVar,evenmean,evenprofit,saferVar,safermean,saferprofit,aggVar,aggmean,aggprofit
),ncol=3,byrow=TRUE)
colnames(t) <- c("5% Value at Risk","Expected return","The top 10% potential profit")
rownames(t) <- c("SPY","TLT","LQD","EEM","VNQ","Even","Safer","Aggressive")
t <- as.table(t)
t
```

```
##            5% Value at Risk Expected return The top 10% potential profit
## SPY                -8388.289      100759.214                     7527.351
## TLT                -5999.653      100597.012                     5939.240
## LQD                -3028.365      100392.989                     2872.259
## EEM               -13558.980      102036.121                    11992.560
## VNQ               -13729.160      100750.798                    12246.862
## Even               -6167.094      100907.227                     6099.977
## Safer              -2870.627      100527.441                     3113.661
## Aggressive        -10659.599      101139.761                     9825.424
```

Market Segmentation

Clustering

```
library(corrplot)
social=read.csv('marketing.csv')
# correlation of interests
interests=social[2:37]
interests_s=scale(interests)
corrplot(cor(interests_s),type = 'upper',order = 'hclust')
```

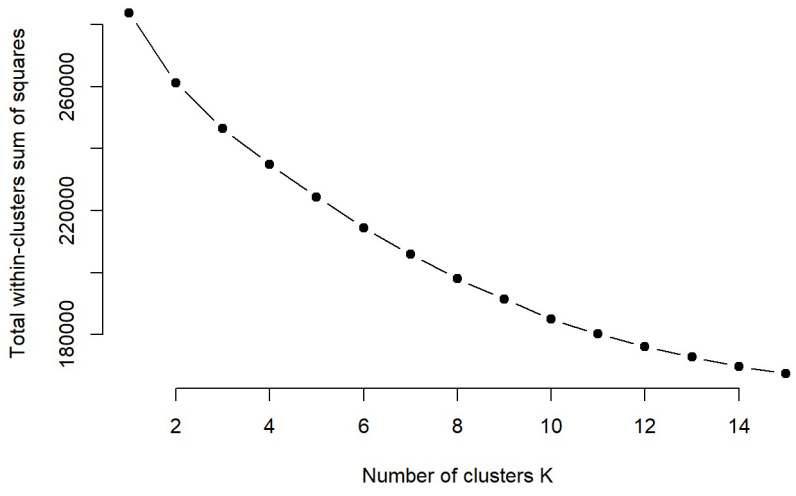


From the correlation plot, we can see that politics and news, politics and travel and computer, online game and coffee uni, beauty and cooking and fashion, parenting and family along with school and sports fandom and religion, photo-sharing and chatter and shopping are the interest groups that are highly correlated.

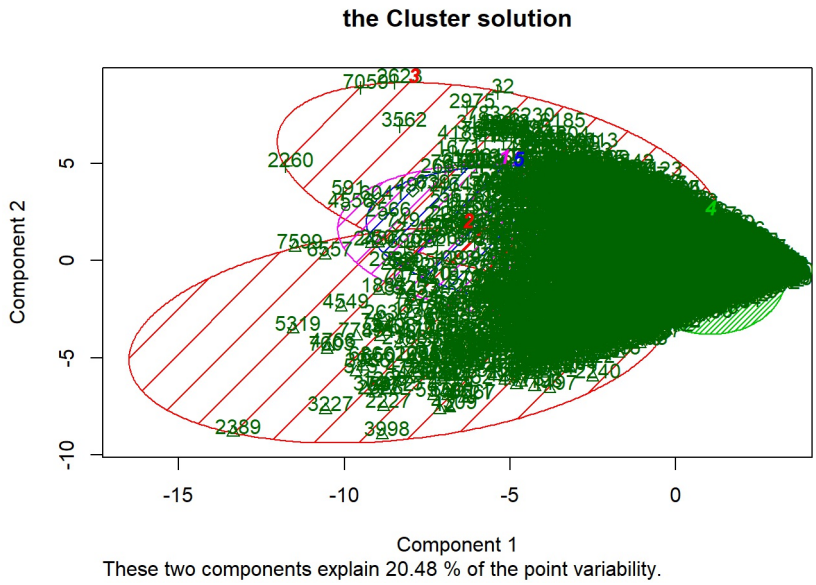
```
# clustering
# the number of clusters using elbow method
set.seed(666666)
k.max=15
wss=sapply(1:k.max,function(k){kmeans(interests_s, k, nstart=50,iter.max = 15 )$tot.withinss})
wss
```

```
##  [1] 283716.0 261219.5 246471.6 234995.5 224399.0 214480.6 205922.7
##  [8] 198062.2 191432.1 184936.3 180158.6 176097.6 172721.5 169583.6
## [15] 167343.3
```

```
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```




```
# we find 5 clusters is the best
cluster=kmeans(interests_s,5)


library(cluster)
clusplot(interests_s,cluster$cluster,main='the Cluster solution',color=TRUE, shade=TRUE, labels=2, lines=0)
```



K-means

Using K-means, we were able to cluster 5 groups of users that share the common interest or similar feature. However, in our case, we are trying to find the market segment—finding groups of features rather than the similar users, and K-means in our cases, cannot visualize groups of features. Besides, K-means is not able to lower the dimensions of as PCA does. Thus, we decided to further use PCA to probe the groups of correlate features, which help firm form advertising campaign more sharply to its target audience.

PCA

```
social=read.csv('marketing.csv')
# correlation of interests
interests=social[,c(1,3:5,7:37)]
rownames(interests) = interests$X
interests = interests[,-1]
colnames(interests)
```

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

Since “chatter” and “uncategorized” variables are sometimes used sparingly, we removed the two variables in our analysis.

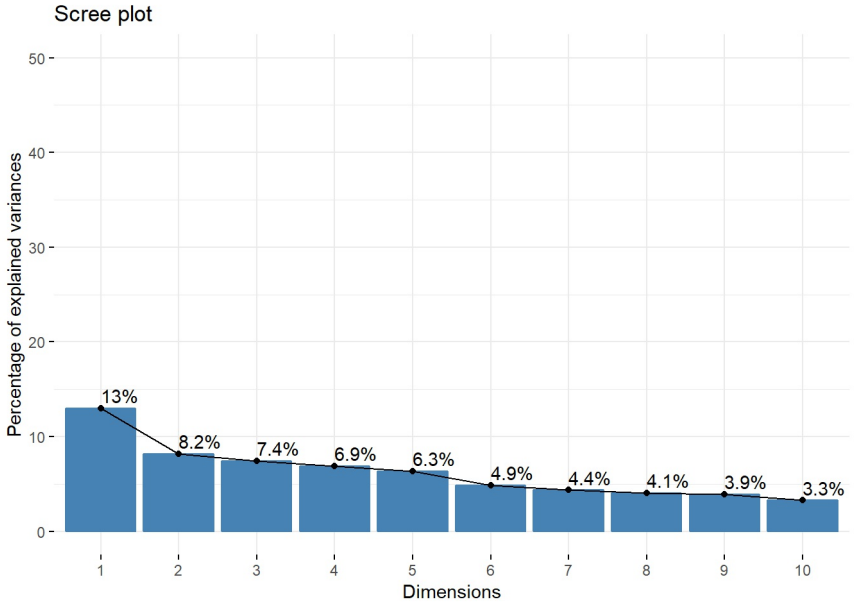
```
library("FactoMineR")
library("factoextra")
library(corrplot)

m.pca = PCA(interests, graph = FALSE)
```

First, we try to get the eigenvalues to measure the variation of each dimensions. Therefore we are able to choose how many dimensions we need.

```
m.eigmal = get_eigenvalue(m.pca)
m.eigmal
```

```
##        eigenvalue variance.percent cumulative.variance.percent
## Dim.1   4.4071010       12.9620618                    12.96206
## Dim.2   2.7717959        8.1523408                    21.11440
## Dim.3   2.5322109        7.4476792                    28.56208
## Dim.4   2.3392479        6.8801410                    35.44222
## Dim.5   2.1535539        6.3339821                    41.77620
## Dim.6   1.6527377        4.8609931                    46.63720
## Dim.7   1.4797375        4.3521692                    50.98937
## Dim.8   1.3779902        4.0529124                    55.04228
## Dim.9   1.3235869        3.8929026                    58.93518
## Dim.10  1.1078949        3.2585143                    62.19370
## Dim.11  0.9866567        2.9019313                    65.09563
## Dim.12  0.9285924        2.7311542                    67.82678
## Dim.13  0.9229439        2.7145410                    70.54132
## Dim.14  0.8736930        2.5696854                    73.11101
## Dim.15  0.8502193        2.5006450                    75.61165
## Dim.16  0.8245305        2.4250896                    78.03674
## Dim.17  0.7282774        2.1419924                    80.17874
## Dim.18  0.6950441        2.0442473                    82.22298
## Dim.19  0.6491969        1.9094027                    84.13239
## Dim.20  0.5672877        1.6684931                    85.80088
## Dim.21  0.4817393        1.4168803                    87.21776
## Dim.22  0.4685476        1.3780813                    88.59584
## Dim.23  0.4257021        1.2520650                    89.84791
## Dim.24  0.4207699        1.2375584                    91.08546
## Dim.25  0.4052348        1.1918671                    92.27733
## Dim.26  0.3992087        1.1741432                    93.45147
## Dim.27  0.3783546        1.1128077                    94.56428
## Dim.28  0.3590644        1.0560718                    95.62035
## Dim.29  0.3527595        1.0375280                    96.65788
## Dim.30  0.3032964        0.8920483                    97.54993
## Dim.31  0.2346946        0.6902782                    98.24021
## Dim.32  0.2283127        0.6715079                    98.91172
## Dim.33  0.1919201        0.5644708                    99.47619
## Dim.34  0.1780965        0.5238134                   100.00000
```

```
fviz_eig(m.pca, addlabels = TRUE, ylim = c(0, 50))
```

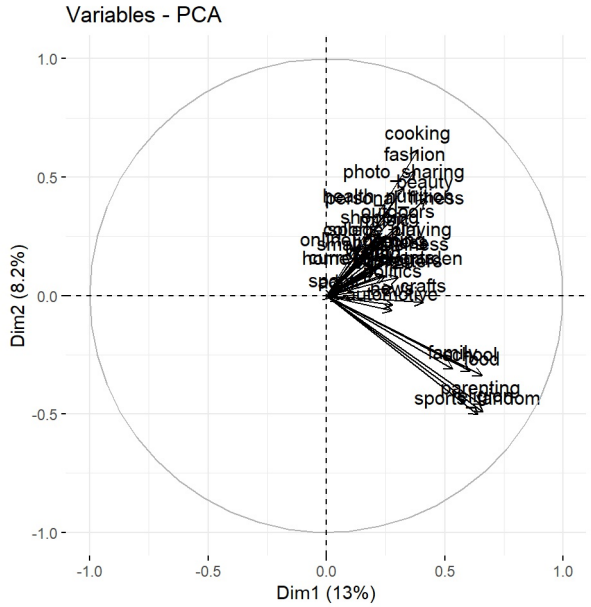


With PCA, we found 5 components can explain most of the variation in the original data from the plot. Together with those 5 dimensions, we have 41.8% representation of the total variable. One interesting point is that in the first dimension, the coefficient are positive, and in the second dimension, several features stand out.

```
m.var = get_pca_var(m.pca)
head(m.var$coord)
```

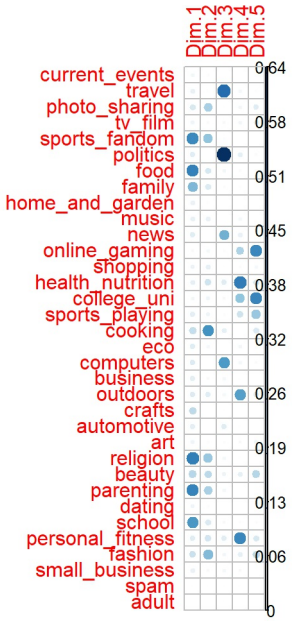
```
##                   Dim.1       Dim.2       Dim.3       Dim.4       Dim.5
## current_events 0.1924697  0.09006604  0.07400009  0.03331178 -0.05598353
## travel         0.2487706  0.08329267  0.70153656 -0.16330764 -0.08096071
## photo_sharing  0.3282786  0.45457789 -0.04120140  0.19579308 -0.29117912
## tv_film        0.1967814  0.13108807  0.15788800  0.19055815  0.26211247
## sports_fandom  0.6396073 -0.50273591 -0.11190856  0.04649573 -0.03014398
## politics       0.2784382  0.03436426  0.80051665 -0.25323168 -0.13127913
```

```
fviz_pca_var(m.pca, col.var = "black")
```

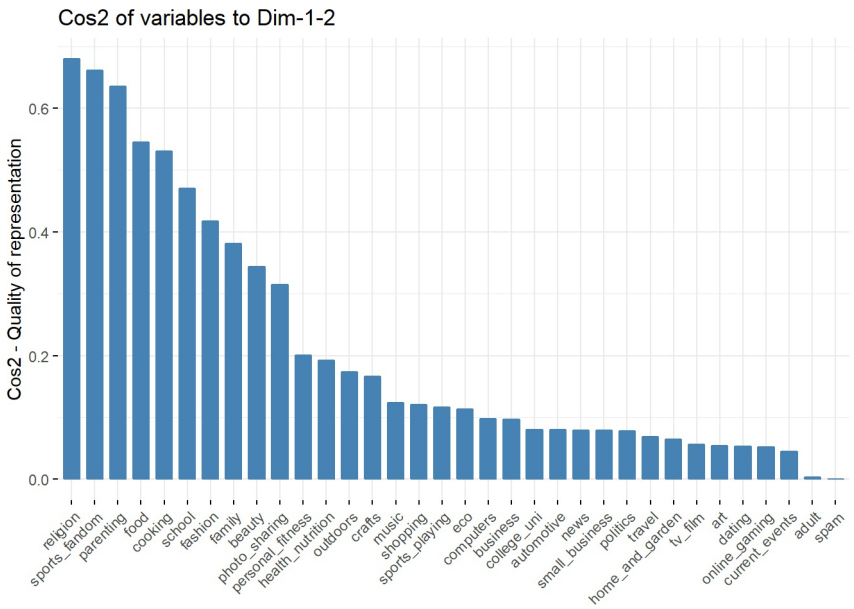


With PCA, we found 5 components can explain most of the variation in the original data from the plot. Together with those 5 dimensions, we have 41.8% representation of the total variable. One interesting point is that in the first dimension, the coefficient are positive, and in the second dimension, several features stand out.

```
corrplot(m.var$cos2, is.corr=FALSE)
```

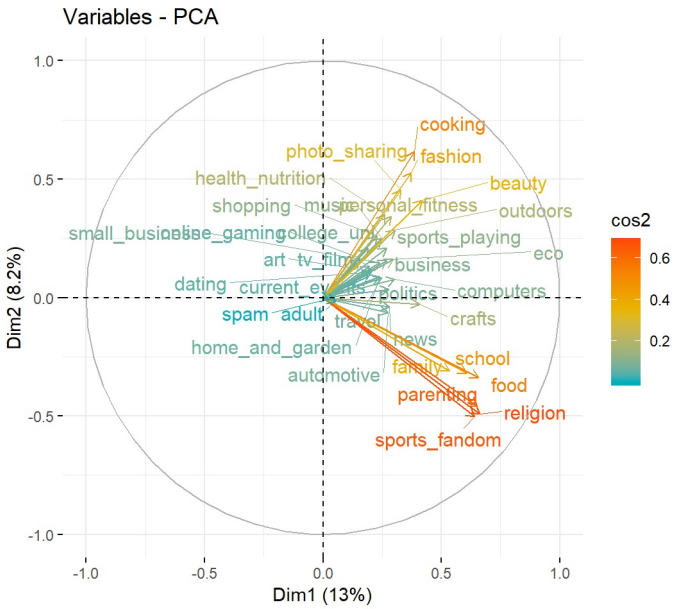



```
fviz_cos2(m.pca, choice = "var", axes = 1:2)
```



To better assist with visualization of the graph above, this graph validates our findings that sports_fandom, food, family, religion, parenting, cooking and school stands out in dimension 1 and 2. On the contrary, current events, dating, online-gaming, art etc. are the least significant variable that can represent the twitter follower.

```
fviz_pca_var(m.pca, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE # Avoid text overlapping
             )
```



In this plot, we combine the coordinates and significant levels of each variables to directly show the market segments that stand out.

Based on these analysis, a possible interesting fact is that most of these twitter followers are relatively elder or already have family, because they focus more on traditional topics, such as religion, sports, parenting and cooking. From another aspect, online gaming, tv&film and dating, which are popular topics among younger people, are not as representative as above segments.