Aifaz Gowani - HW1

PART A: Random Clickers and truthful clickers

p(randome clicker saying yes) = .5 p(random clicker saying no) = .5

expected fraction of random clickers = .3 expected fraction of truthful clickers = .7

Total results: YES = .65 No = .35

x = what fraction of people who are truthful clickers that said yes??

Given the prob of yes = (Random clickers)(random clickers saying yes) + (truthful clickers)(x)

```
#     .65 = (.5)(.3) + (.7)(x)
.50 /.7 #  = 'X'
```

```
## [1] 0.7142857
```

```
#  The answer is that about 71.43% of truthful clickers said yes.
```

PART B: Medical test for a disease

-If someone has a disease, there is a prob of .9993 that they will test positive. - If someone does not have a disease, there is a prob of .9999 that they will test negative.

Prob of disease in the population =.000025

x = p(has disease | test positive) = p(test positive | has disease) p(disease) ———————————————————————— p(test positive | has disease) p(disease) + (tests positive | no disease) p(no disease)

```
num = ((.9993)*(.0025))
den = (((.9993)*(.000025))+ ((.0001)*(.999975)))
final_answer = num/ den
```

The final answer: about 19.88% that someone who tests positive actually has the disease. That is very shocking to see because without this Bayes theorem, you would think that the test has accuracy of .9993 which can be misleading if you do not have the right aprroach.

```
setwd('Desktop')
library(mosaic)
```

```
## Loading required package: dplyr
```

```
## Warning: package 'dplyr' was built under R version 3.5.1
```

```
##
## 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

## Loading required package: ggstance

##
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh

##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix

##
## 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 object is masked from 'package:Matrix':
##
##     mean
```

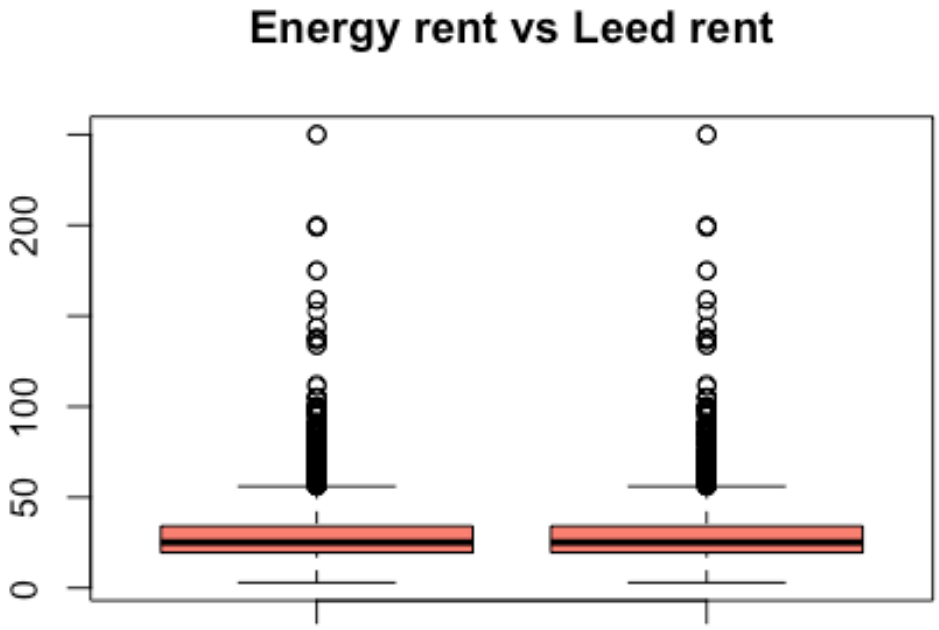
```
## The following object is masked from 'package:ggplot2':
##
##     stat

## 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(ggplot2)

green = read.csv('green.csv')
```

The next step was to extract the green buildings only so we can make meanigful assumptions about those buildings in relative to other buildings.

```
green_only = subset(green, green_rating==1)
dim(green_only)

## [1] 685  23

not_green = subset(green, green_rating == 0)
```

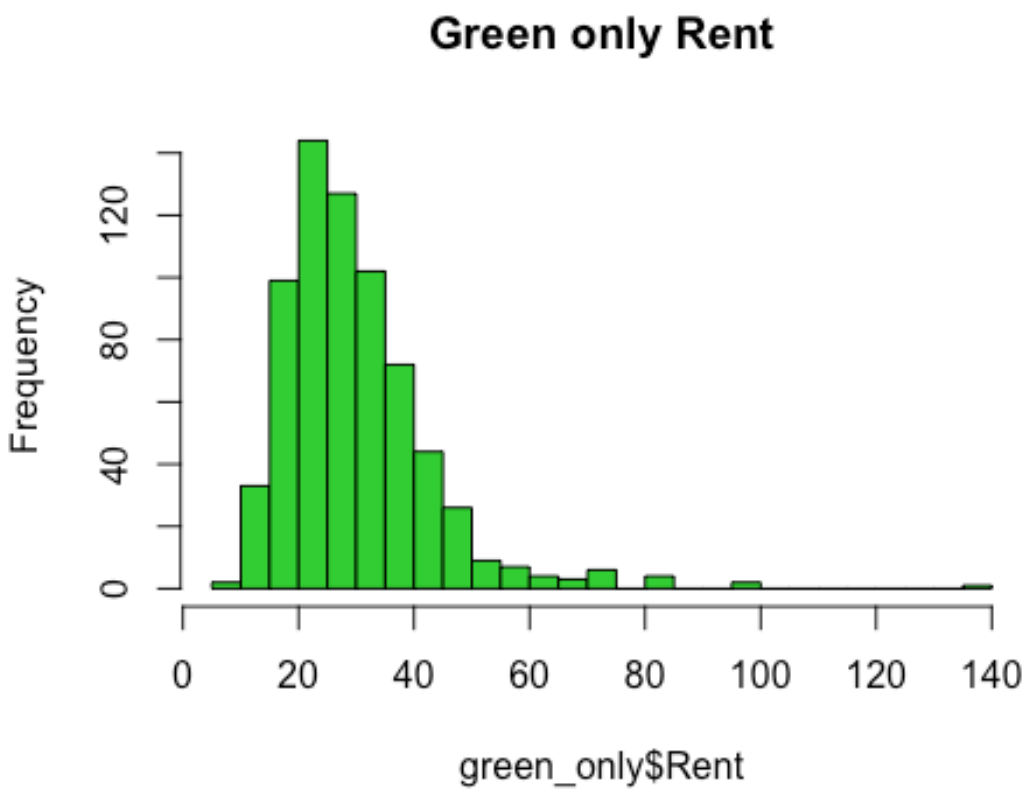
I then wanted to see whether there is any major impact does it play whether you have a LEED certification or a EnergyStar and whether that plays an impact on the rent that the tenants have to pay.

```
attach(green)
#hist(green_only$Rent, green_only$size)
LEED = subset(green, LEED = 1)
Energy = subset(green, Energystar = 1)
boxplot(Energy$Rent,LEED$Rent, col = 'salmon',main ='Energy rent vs Leed
rent')
```

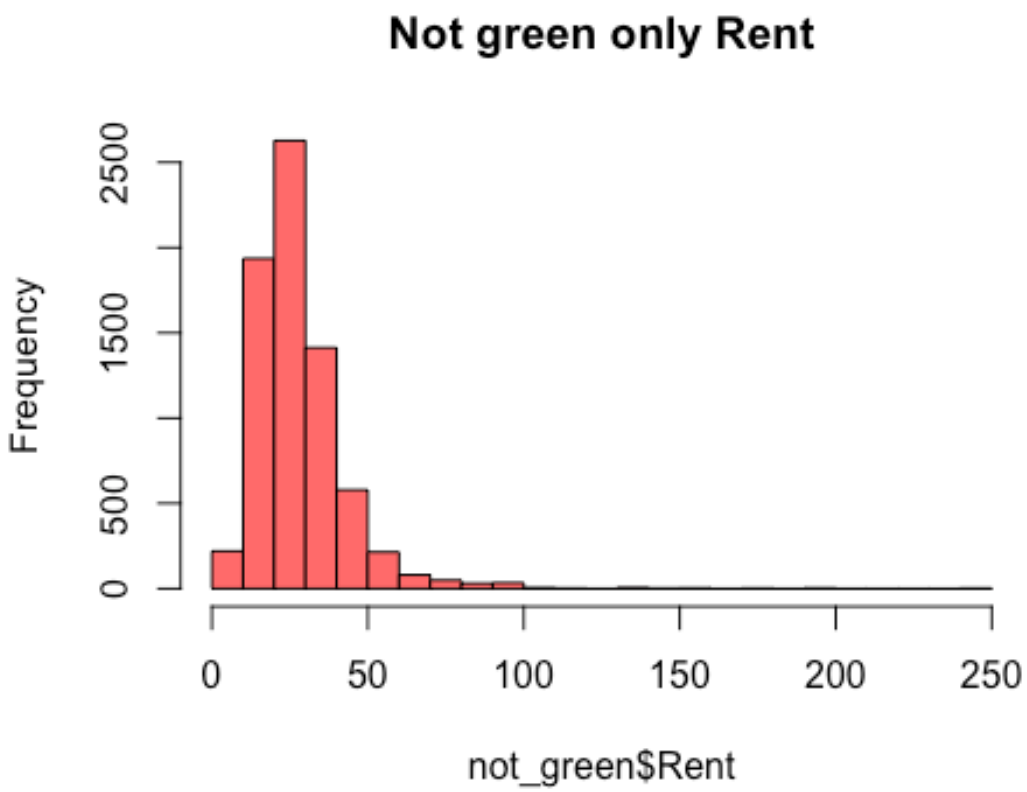


After plotting the box plot above, i was able to identify that there was no real benefit of having one certification over the other and both were providing with equal results.

```
hist(green_only$Rent, 30, col='limegreen',main ='Green only Rent')
```

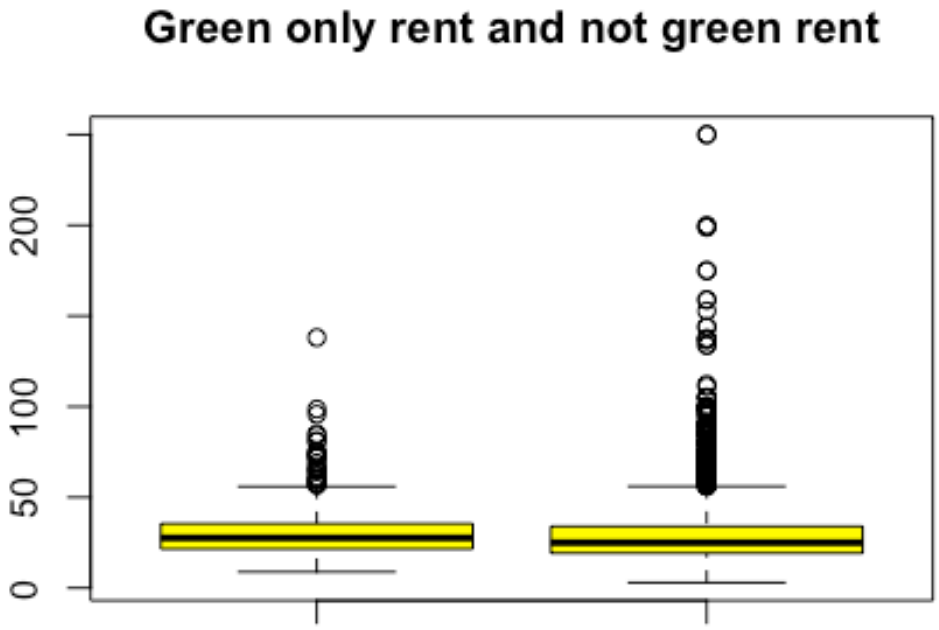



```r
hist(not_green$Rent,30,col = 'indianred1',main ='Not green only Rent')
```



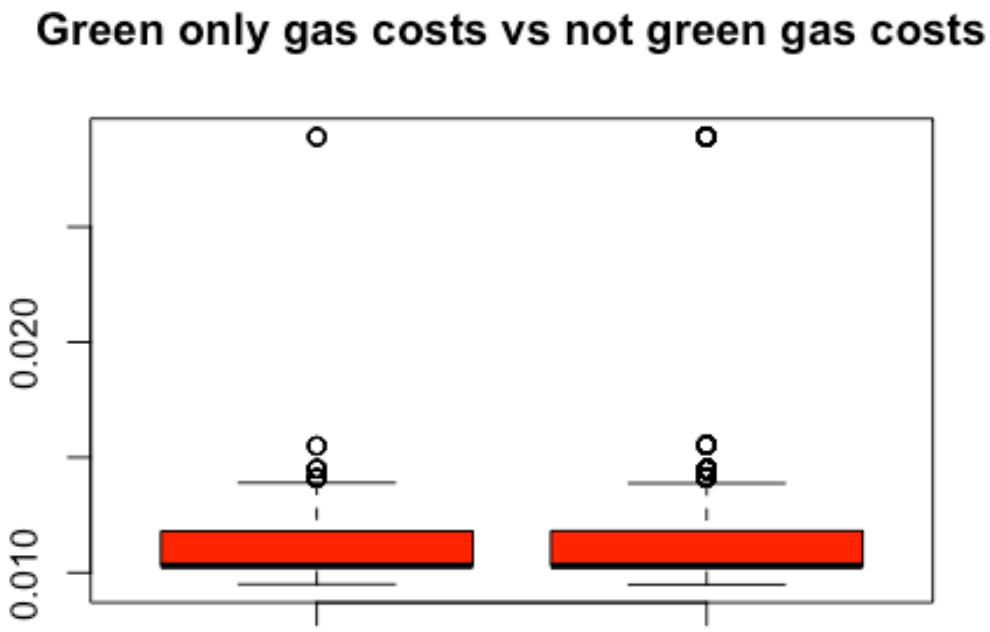
I wanted to understand the distribution of rent for both: the green building and non green buildings. Therefore I decided to create a historgram. The amount is already being graphed in dollars per square foot. Throughout this exploratory data analysis, I was experimenting with what would be an ideal grpah to convey a message through graphs. the grpahs above allow us to look at their spread individually, however face problems when the user has to compare the two because of the separate x/y axis. Therefore, the box plot shown below was a better alternative.

```
boxplot(green_only$Rent,not_green$Rent,col='yellow', main = 'Green only rent
and not green rent')
```



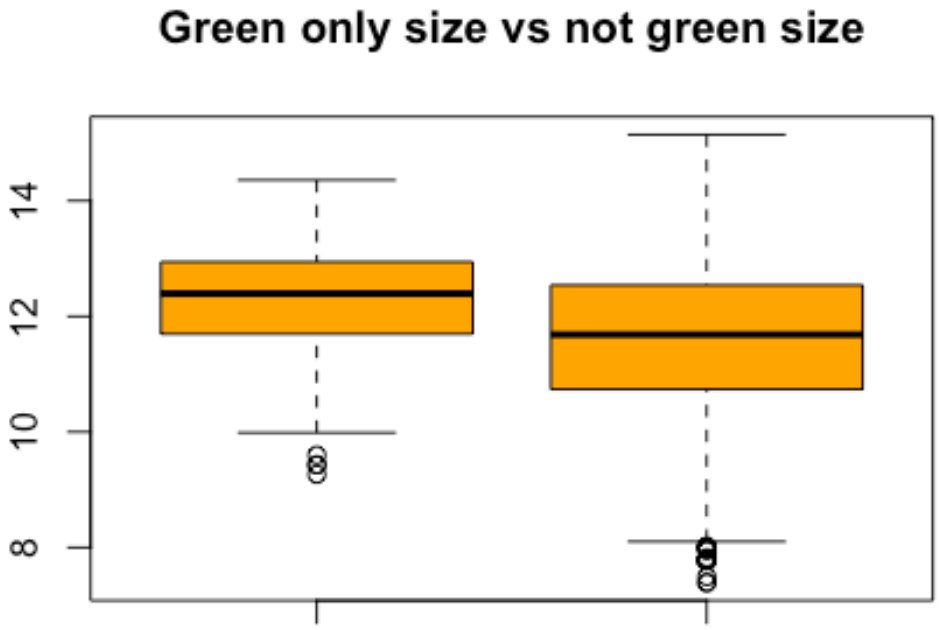
After plotting the box plot above, you can see that the spread of non-green buildings is much larger than it is for green buildings and the median distribution for green building's rent tend to be little higher.

```
boxplot(green_only$Gas_Costs,not_green$Gas_Costs,col = ('red'),main ='Green
only gas costs vs not green gas costs')
```



The entire purpose of the green buildings is to be environmentally concious and help less conumption of things that can be potentially harmful. One of these things that these buildings focus on is primarily reduction of gas usage. My intuition was that this will allow the gas costs to be definetly lower than thos buildings that are considered to be non-gree. However, this was not the case and they were paying about equivalently for their gas costs no matter what type of building that they were living in as shown above.

```
boxplot(log(green_only$size),log(not_green$size),col='orange',main ='Green
only size vs not green size')
```



Is it always right that greener apartmartments tend to have a higher size than those buildings that are not? Upon investigating this relationship, I found that it is true that those buildings that are considered to be greener, tend to have a large size than those who are considered to be non-green

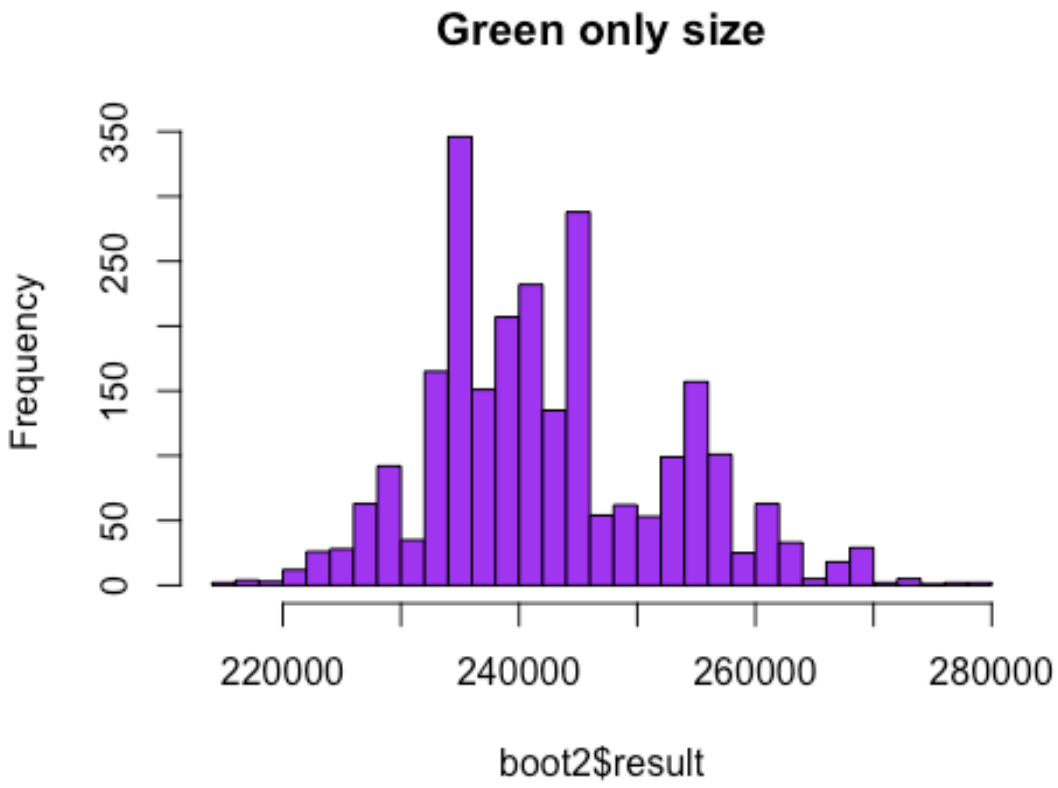
To expand my research more about the true median of the size of the buildings, I believed that boostrapping would be a great method to do 2500 samples and get close to the truth. Upon running the boostrap, I received that the median size is between 2255728 and 263765 square feet with a 95% confidence interval.

Below is presented also a graph to provide a visual of the size within the buildings and specifically to show that this is not a normal distribution. This could be mainly because of the location of buildings within a specific area and proximity to downtown etc.

```
##### Bootstrap the median
median(green_only$size)

## [1] 241150

boot2 = do(2500)*{
  median(resample(green_only)$size)
}
head(boot2)
```

```
##   result
## 1 255294
## 2 239250
## 3 256739
## 4 246625
## 5 238437
## 6 233225
```

```
hist(boot2$result, 30, col='purple', main ='Green only size')
```

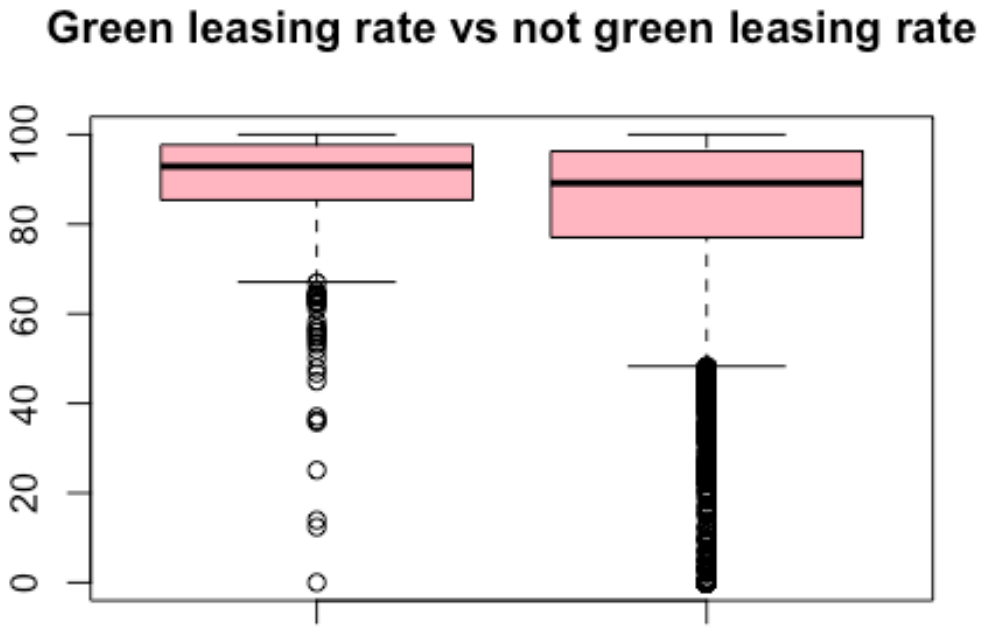



```
confint(boot2)
```

```
##     name  lower  upper level     method estimate
## 1 result 225500 264149  0.95 percentile   241150
```

My last appraoch was to look at whether the buildings that are considered to be particularly greener buildings tend to have a higher occupancy rate. This will be a key factor because if a building decides to go green then it should at least get its returns back in the form of having a higher occupany rate and that is what is show in the box plot below.

```
boxplot(green_only$leasing_rate,not_green$leasing_rate,col='lightpink', main
='Green leasing rate vs not green leasing rate')
```



As you can see above that the buildings that are tend to be considered as greener buildings, have a higher occupancy rate that thos that are non-green.

OVERALL: After looking at the data about the rent distribution, size of the apartments, leasing occupancy of these buildings etc., I would support that the decision of the analyst was a valid one which would lead to a profit down the road after the intial costs of becoming a green building is satisfied. This will also help tenants gain a recognition as environmentally concious and promote their quality of life as well.

BOOTSTRAPPING QUESTION:

Download several years of daily data on these ETFs, using the functions in the quantmod package, as we used in class. Go back far enough historically so that you get both good runs and bad runs of stock-market performance. Now explore the data and come to an understanding of the risk/return properties of these assets. Then consider three portfolios:

the even split: 20% of your assets in each of the five ETFs above. something that seems safer than the even split, comprising investments in at least three classes. You choose the allocation, and you can certainly invest in more than three assets if you want. (You can even choose different ETFs if you want.) something more aggressive (again, you choose the allocation) comprising investments in at least two classes/assets. By more aggressive, I mean a portfolio that looks like it has a chance at higher returns, but also involves more

risk of loss. Suppose there is a notional $100,000 to invest in one of these portfolios. Write a brief report that:

marshals appropriate evidence to characterize the risk/return properties of the five major asset classes listed above. outlines your choice of the “safe” and “aggressive” portfolios. uses bootstrap resampling to estimate the 4-week (20 trading day) value at risk of each of your three portfolios at the 5% level. compares the results for each portfolio in a way that would allow the reader to make an intelligent decision among the three options. You should assume that your portfolio is rebalanced each day at zero transaction cost. That is, if you’re aiming for 50% SPY and 50% TLT, you always redistribute your wealth at the end of each day so that the 50/50 split is retained, regardless of that day’s appreciation/depreciation.

```
library(mosaic)
library(quantmod) # stands for quantiative modeling

## 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)


#### Now use a bootstrap approach
#### With more stocks

mystocks = c("SPY", "TLT", "LQD", "EEM", "VNQ")
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")
## and getOption("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).
```

```
# A chunk of code for adjusting all stocks
# creates a new object addind 'a' to the end
# For example, WMT becomes WMTa, etc
for(ticker in mystocks) {
  expr = paste0(ticker, "a = adjustOHLC(", ticker, ")")
  eval(parse(text=expr))
}

head(SPYa)
```

```
##            SPY.Open SPY.High  SPY.Low SPY.Close SPY.Volume SPY.Adjusted
## 2007-01-03 112.3644 112.8462 111.0373  111.6693   94807600     111.6693
## 2007-01-04 111.5587 112.2064 111.0689  111.9062   69620600     111.9062
## 2007-01-05 111.6377 111.6930 110.8873  111.0136   76645300     111.0136
## 2007-01-08 111.2348 111.7009 110.7846  111.5271   71655000     111.5271
## 2007-01-09 111.6219 111.8510 110.9031  111.4323   75680100     111.4323
## 2007-01-10 111.0452 111.8273 110.8241  111.8035   72428000     111.8035
```

The first part of this question wanted us to read the files that's contain data from several years ago. I decided to get all the data from the year 2007 on wards because this will allow me to incorporate the downfall of stocks such as S&P 500 during the great recession. The next step was to adjust for dividends and stock splits that may have occurred after that point that data was gathered.

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

head(all_returns)
```

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

```
## 2007-01-04  0.001296655
## 2007-01-05 -0.018518518
## 2007-01-08  0.001451392
## 2007-01-09  0.012648208
## 2007-01-10  0.012880523
```

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

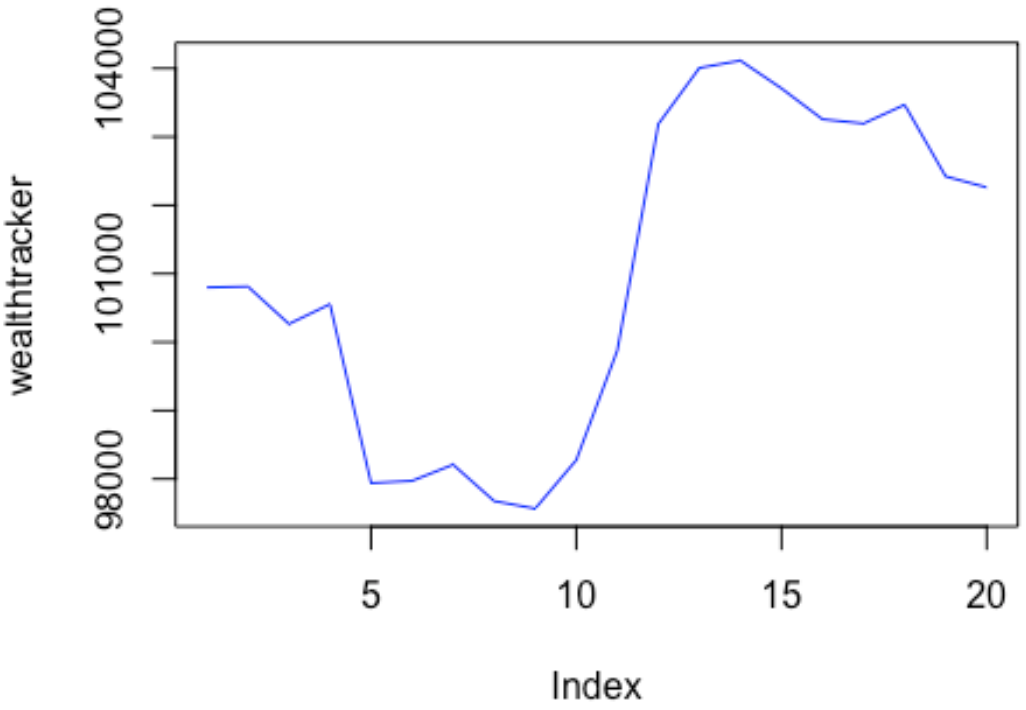
```
set.seed(1)
```

the second part was to combine all the stocks that were adjusted into a simple matrix. we also have to take into account community imported data that there're no missing values and if there are missing values then get rid of them and that is exactly what I did above.

```
# Now loop over 4 trading weeks
# approach 1
total_wealth = 100000
weights = c(0.2, 0.2, 0.2, 0.2, 0.2)
holdings = weights * total_wealth
n_days = 20
wealthtracker = rep(0, n_days) # Set up a placeholder to track total wealth
for(today in 1:n_days) {
  weights = c(0.2, 0.2, 0.2, 0.2, 0.2)
  return.today = resample(all_returns, 1, orig.ids=FALSE)
  holdings = total_wealth * weights
  holdings = holdings + holdings*return.today
  total_wealth = sum(holdings)
  wealthtracker[today] = total_wealth
}
total_wealth
```

```
## [1] 102261
```

```
plot(wealthtracker, type='l',col='blue')
```



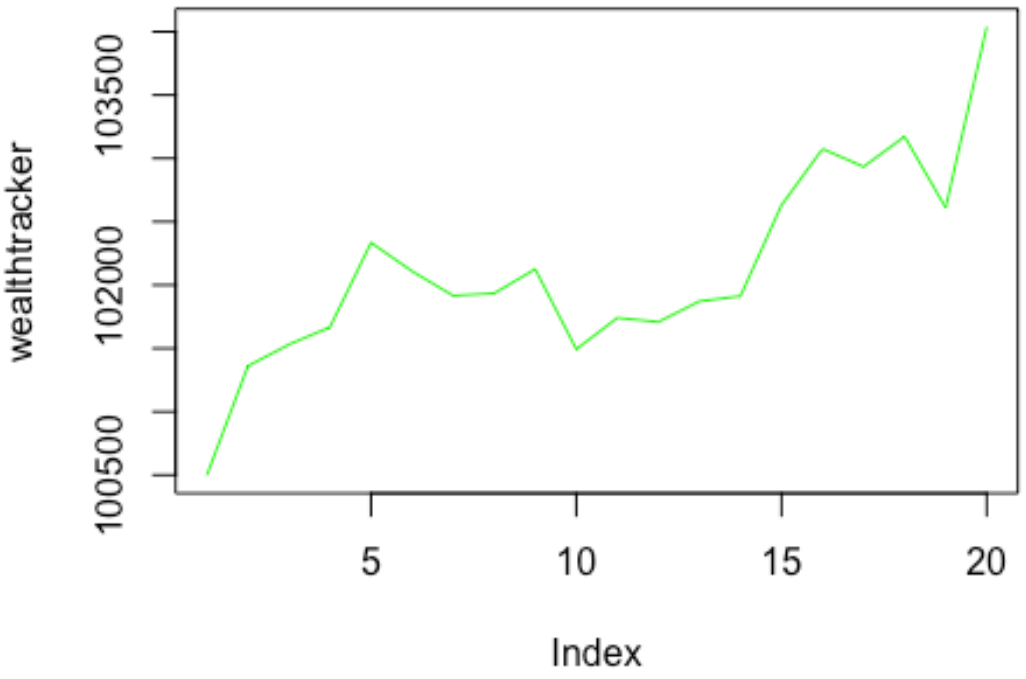
We started out with 100k and our goal was to allocate resources in such a way that profits are maximized. Another thing to note is that the days needed here are 20 trading days which is something that i adjusted for above. In approach number one, we decided to allocate equal amount of our wealth in each one of the stock tickers. By doing so we wanted to see if we will be able to mitigate risk. We wanted to make sure that the allocation of weights remain the same day in and day out that is the reason that i placed weights inside the for loop along within holdings so it can be reassigned and use again the next day with new accumulated allocation. As you can see that there was a great profit this time of about 10% when we alloacted our resources evenly.

```
# approach 2
total_wealth = 100000
weights = c(0.05, 0.3, 0.3, 0.3, 0.05)
holdings = weights * total_wealth
n_days = 20
wealthtracker = rep(0, n_days) # Set up a placeholder to track total wealth
for(today in 1:n_days) {
  weights = c(0.05, 0.3, 0.3, 0.3, 0.05)
  holdings = total_wealth * weights
  return.today = resample(all_returns, 1, orig.ids=FALSE)
  holdings = holdings + holdings*return.today
  total_wealth = sum(holdings)
```

```
  wealthtracker[today] = total_wealth
}
total_wealth
```

```
## [1] 104038.8
```

```
plot(wealthtracker, type='l',col='green')
```



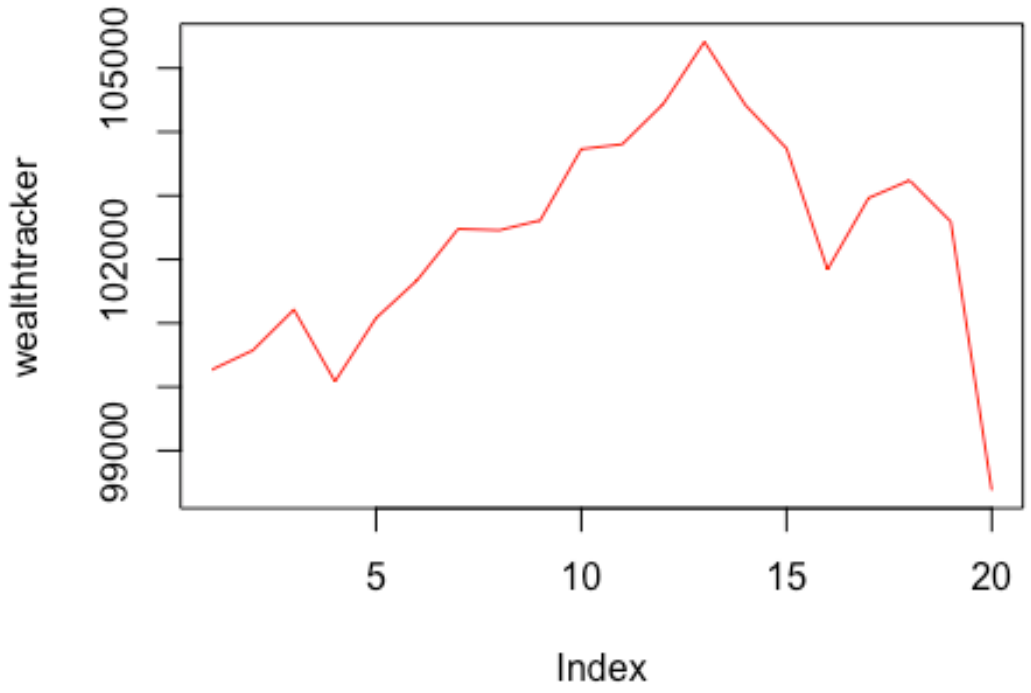
In the approach number two, we wanted a strategy that was a little bit even safer then splitting the entire portfolio into 20% for each stock ticker. for this approach I decided to invest 30% in three different stocks and about 5% each and the remaining two stocks. I assumed that this approach would provide me with a higher gain however this is not the case and average increase with this approach was only about 2% compared to approach number one where the return was almost 10%.

```
# approach 3
total_wealth = 100000
weights = c(0.5, 0.0, 0.0, 0.0, 0.5)
holdings = weights * total_wealth
n_days = 20
wealthtracker = rep(0, n_days) # Set up a placeholder to track total wealth
for(today in 1:n_days) {
  weights = c(0.5, 0.0, 0.0, 0.0, 0.5)
```

```
  holdings = total_wealth * weights
  return.today = resample(all_returns, 1, orig.ids=FALSE)
  holdings = holdings + holdings*return.today
  total_wealth = sum(holdings)
  wealthtracker[today] = total_wealth
}
total_wealth
```

```
## [1] 98377.52
```

```
plot(wealthtracker, type='l',col='red')
```



Using the third approach which was to allocator funds 50-50 amongst two stocks, I was able to get better result in the end but earning about 3.5% compared to 2% which was achieved during approach number two. although this was one and a half % higher than approach number two, I believe that over long term this is not a good approach because if these two markets perform poorly then entire investment suffers. however, if you allocate the funds among several different markets then is a higher chance of not losing money when it comes to Bear market.

Overall: I got the highest results where the funds were equivalently split among the five markets and that makes sense. When one market goes down, since the other one is in a different category, it is unlikely that market will also go down because of being in a

different industry. I would invest money where it is evenly split in the market therefore it can cover the costs of a downfall of another market and allocate resources safely compared to the other approaches where the benefit was significantly lower because of the weight places on certain markets was higher.

Market Segmentation

Your task to is analyze this data as you see fit, and to prepare a (short!) report for NutrientH20 that identifies any interesting market segments that appear to stand out in their social-media audience. You have complete freedom in deciding how to pre-process the data and how to define "market segment." (Is it a group of correlated interests? A cluster? A latent factor? Etc.) Just use the data to come up with some interesting, well-supported insights about the audience.

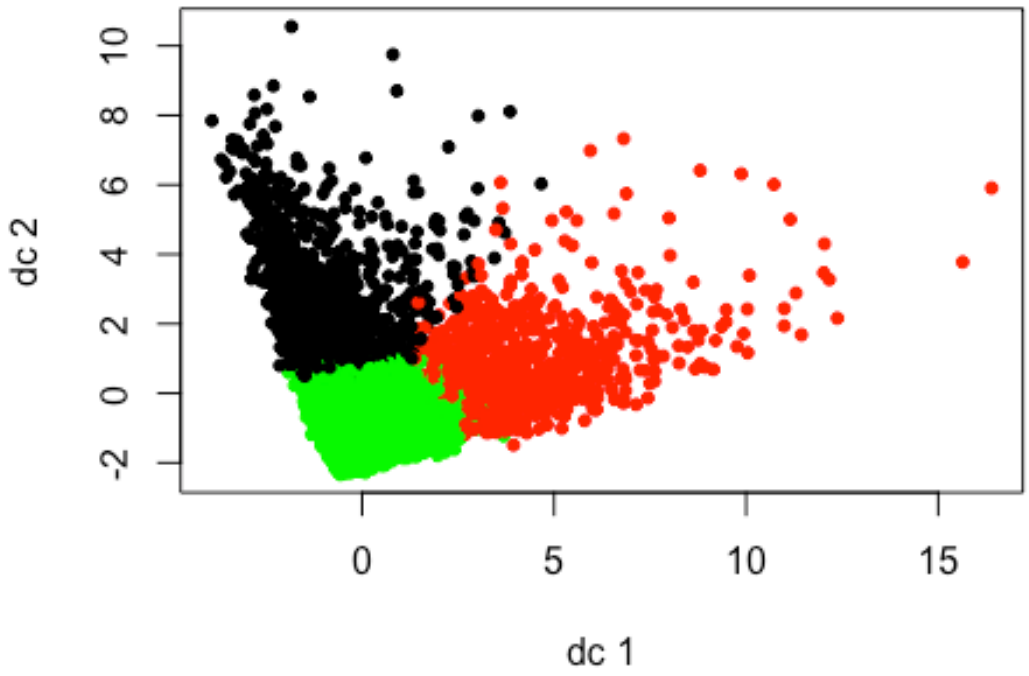
```
social_marketing <- read.csv("Desktop/social_marketing.csv", row.names=1)
#head(social_marketing, 10)



X = social_marketing
#View(X)
set.seed(12)
X = scale(X[,2:36], center=TRUE, scale=TRUE)
X = na.omit(X)
```

The first step was to import the dataset but simply doing this won't allow us to begin our exploratory analysis, we would have to make the variables standardize and normalize them by the use of z-scores or any other form of normalization. That is what i have done in the later part of the code presented above. You also need to get rid of the values which are missing because this can lead you to gather and make inferences based on wrong information.

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

This was just for my knowledge to learn a bit more about the data set by getting the center of the data set original and the dataset with the one that was scaled.

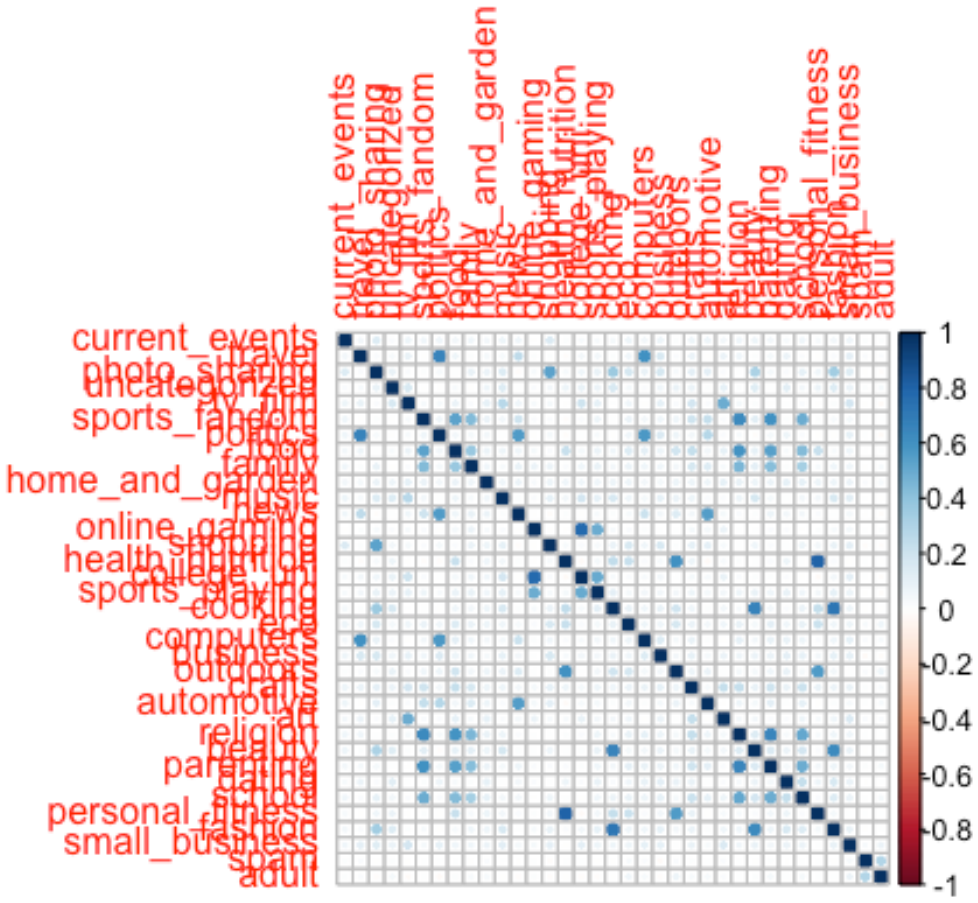
```
library(cluster)
library(fpc)
clust1 = kmeans(X, 3,iter.max = 10000)
#head(clust1)
plotcluster(X, clust1$cluster,pch=20)
```




```
#clust1$centers
#names(clust1)
```

After grouping the points into three different categories and applying kmeans with the iteration of 10,000, it was easier to see that pattern was developing. after experimenting with several different clusters, the results were most clear when the cluster was at three. the plot above helps us explain the distance that is between different clusters. this can you allow us to use the different types of methods in order to capture the essence of how far an attribute is from another cluster and predict whether an attribute belongs to a specific cluster with similar traits. clustering is specifically helpful because it groups the predictors and allow us to infer whether predictor is a member of specific group and guess its attributes in correlation to others attributes in their group. these three groups are split between activities, business and other aspects of life such as as family etc. the activities group includes things such as traveling, Photo sharing, shopping while business includes things such as computers, Business, automotive, and family category includes things such as family, religion, music etc.

```
library(corrplot)

## corrplot 0.84 loaded

a = cor(X)
corrplot(a,method = 'circle')
```



From the exploration of the market segment data, I was particularly interested how certain variables interact with other variables in the data set. Interesting attractions include college universities and online gaming. you can assume that students who are young also are more likely to play games and this can be particularly seen amongst college students data set which just reinforces this idea. another intuitive variable interaction that was noticeable amongst the variables was personal fitness and health care nutrition. Fashion and cooking was also highly and positively correlated which would mean that people who are into cooking are also fashionable. several other interactions were also striking such politics and news which simply means that people who are into politics will keep up with the news and that is again something that is intuitive and interesting to see.