STA380.Ex1

Catherine McNabb

August 10, 2018

Probability Practice

Part A

Let RC stand for Random Clicker and TC stand for Truthful Clicker. We can calculate the fraction of people who are truthful clickers who answered yes using Bayes' Rule:

P(Yes) = .65 P(No) = .35

P(RC) = .3 P(TC) = .7

P(No|RC) = .5 P(Yes|RC) = .5

Ex: 100 people 65 answer yes, 30 are random clickers, and half of those are random clickers who answer yes. So, P(yes|RC) is .15.

That means 65-15 (50) people are truthful clickers who answered yes.

That is 50/70 percent of the truthful clickers who answer yes, or 71%.

The Bayes Theorem formula that applies would go like this:

P(yes|TC) =

$$\frac{P(Yes) - P(yes|RC)}{P(TC)}$$

```
truthful.yes = 50/70
print(truthful.yes)

## [1] 0.7142857
```

Part B

Let D stand for has disease; let ND stand for no disease.

P(D) = .000025 P(ND) = .999975

P(D|yes) = .993 P(ND|no) = .9999

P(yes|D) =

$$\frac{P(D|yes) * P(D)}{P(D|yes)P(D) + P(ND)P(ND|no)}$$

P(yes|D) =

$$\frac{(.993)(.000025)}{(.993)(.000025) + (.999975)(.0001)}$$

```
probability = (.993*.000025)/((.993*.000025) + (.999975*.0001))
probability

## [1] 0.1988824
```

The chance of getting a positive test when having the disease is 19.89%.

This is pretty bad, because if you have the disease, you most definitely want to test positive for it so you can treat it. Even though the test seems pretty accurate, it's not accurate where it counts, which is diagnosing the disease for people who need it.

Exploratory Analysis: Green Buildings

First, I will read in the data to explore.

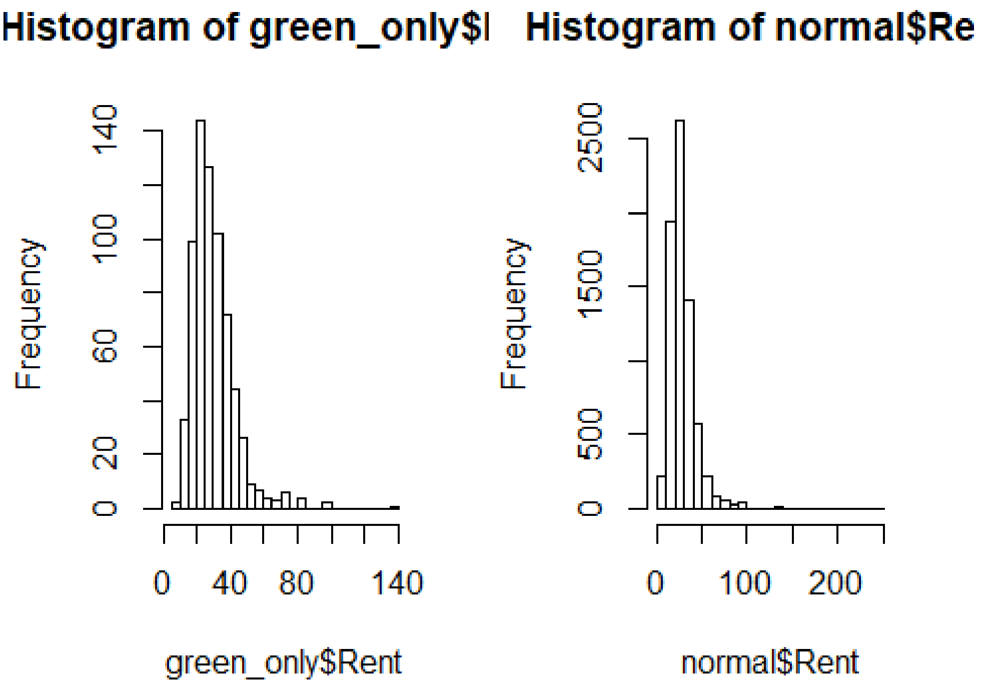
```
library(mosaic)

## Warning: package 'mosaic' was built under R version 3.5.1

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

## Warning: package 'ggplot2' was built under R version 3.5.1

## Warning: package 'ggstance' was built under R version 3.5.1

## Warning: package 'mosaicData' was built under R version 3.5.1

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

Then, I will extract the buildings with green ratings, so I can compare a green-only list to a not-green building list.

```
## dimensions are:  685 23
```

Let's look at the distributions of the rent of the two lists, as well as the average rent per sq foot:

```
par(mfrow=c(1,2))

hist(green_only$Rent, 25)
median(green_only$Rent)

## [1] 27.6

hist(normal$Rent, 25)
```



```
median(normal$Rent)

## [1] 25
```

The distributions both have long tails, but the long tail of the rent price for normal buildings is MUCH longer and weirder than the long tail of the price for green buildings per square foot. Because of the weird distribution, I agree with the 'guru', and I also used median for the usual price of a square foot rather than the mean.

As noted in the assignment, the green is about $2.60 more than the "normal" buildings.

Another problem with this data is that green buildings are probably built more in progressive cities like NY or SF where the price per square foot is more expensive anyways. So maybe we should compare the green only rent to the cluster rent price instead of the overall price.

```
median(green_only$Rent)-median(green_only$cluster_rent)

## [1] 2.225
```

This shows a difference of $2.25, so let's go with that instead of the $2.60 figure.

Now, we compare the size of the buildings. Presumable, bigger size means more tenants and therefore more rent money. Ours is apparently planned to be 250,000 feet, but it would be interesting to know for similar buildings.

```
green_only_size = mean(green_only$size)
green_only_size
```

```
## [1] 325781.3

normal_size = mean(normal$size)
normal_size

## [1] 225977.3
```

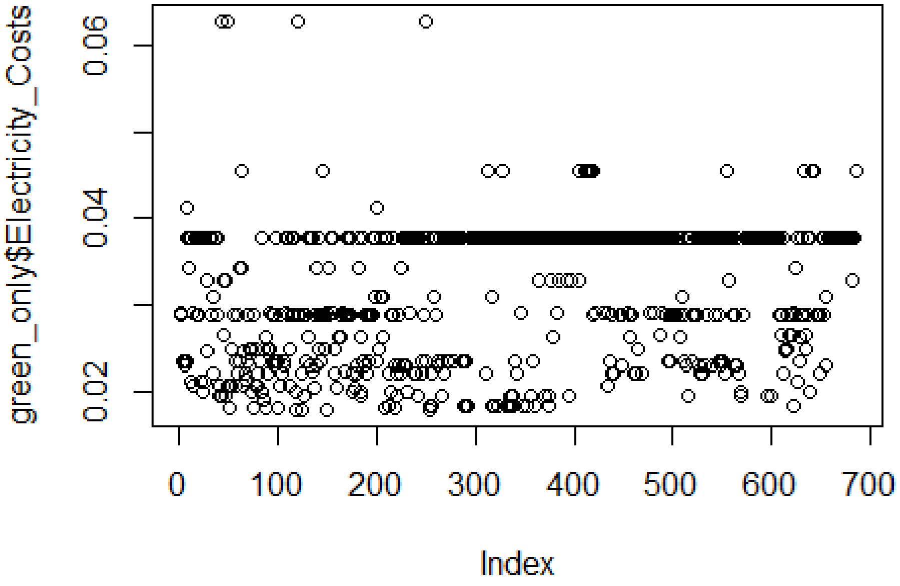
The size of a green building is much larger than a normal building on average. But, maybe the normal buildings have a higher leasing rate, so the extra size doesn't matter. Let's see.

```
green_only_leaserate = mean(green_only$leasing_rate)
green_only_leaserate

## [1] 89.2819

normal_leaserate = mean(normal$leasing_rate)
normal_leaserate

## [1] 81.97206
```

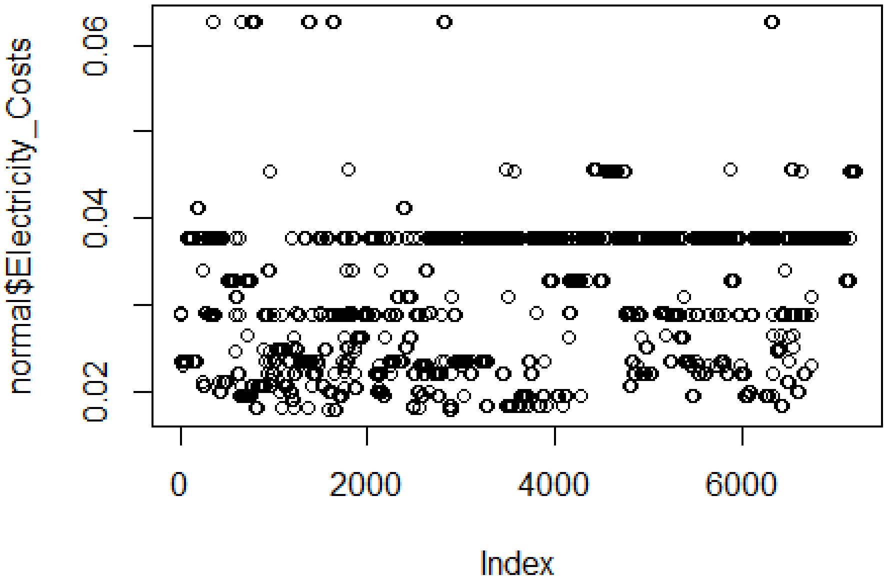
Nope! We can see this is not true. The green buildings are typically 89% leased, while the other buildings are just about 82% leased. If the green buildings are on average larger and the STILL are leasing at a much higher rate, then that bodes well for our building.

Perhaps also the electicty costs are cheaper for green buildings, because they tend to utilize natural lighting.

```
plot(green_only$Electricity_Costs)
```



```
plot(normal$Electricity_Costs)
```



Here, we can see that the plots look almost exactly the same, so we can assume this is generally NOT true.

Back to the analysis from the 'Excel guru' though, I think most of what he said holds up. I think there are a couple additional considerations here that we came up with. First of all, the price is actually closer to only $2.20 difference per square foot instead of $2.60, so that makes it a lot more attractive to tenants even though we won't make as much. Second, and this is not backed up by statistical evidence here, but green buildings and taking climate change into consideration is more important with businesses than ever, so it could have a marginal impact on the bottom line for the business as well. I'd rather work in a green building and work with companies that are aware of their footprints. Additionally, I found that the green buildings are more likely to be leased than the normal buildings, so I am able to agree with the guru's assessment at hoping for a 90% full building to recoup the losses.

Between the two analsyes and the trend toward "greener" decisions, I agree with the guru's assessment that the new building should indeed be green.

amenities

Bootstrapping

Upload the needed libraries and set seed:

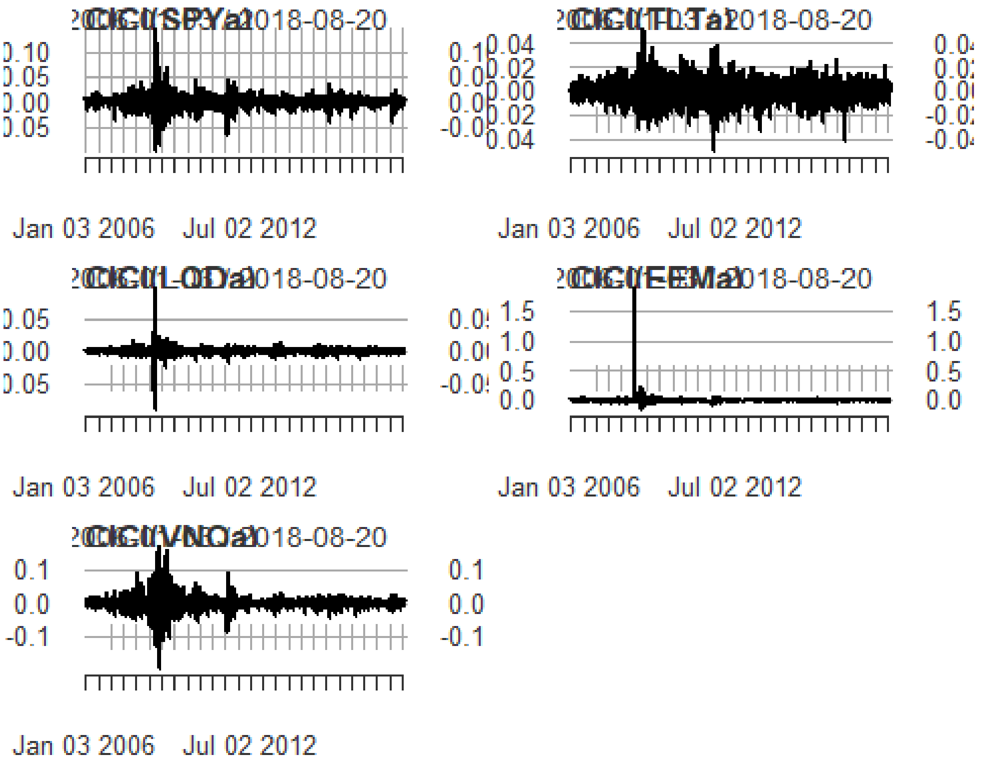
```
library(mosaic)
library(quantmod)

## 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)
set.seed(345)
```

Import the relevant info for the given tickers

```
## '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).

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

As with the stocks in class, ETFs can split and have dividends, so let’s adjust for those.

Look at all close to close changes to see variability of the ETFs.



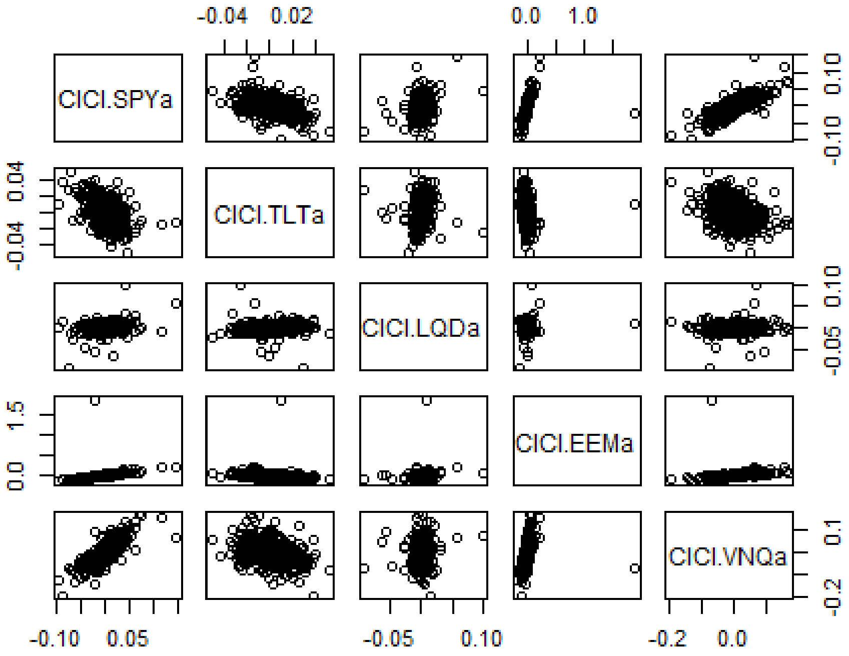
As expected, there is a lot of variability around the Recession in 2008 for all types of ETFs. Otherwise, the emerging market equities and corporate bonds hold very little variability in the past 12 years.

Now, I'll put the returns from the ETFs in one matrix and see what the data looks like using the "head" function.

```
##               ClCl.SPYa    ClCl.TLTa     ClCl.LQDa    ClCl.EEMa
## 2006-01-03           NA           NA            NA           NA
## 2006-01-04 0.0047356434  0.002397047  0.0000000000  0.009066095
## 2006-01-05 0.0006283896 -0.001195663  0.0008312154  0.005087725
## 2006-01-06 0.0083215970 -0.001850016 -0.0015687551  0.020893860
## 2006-01-09 0.0025693086  0.000654143  0.0001848799  0.009811162
## 2006-01-10 0.0010094742 -0.008062737 -0.0026797357 -0.012014198
##              ClCl.VNQa
## 2006-01-03          NA
## 2006-01-04 0.007406205
## 2006-01-05 0.007678500
## 2006-01-06 0.009403340
## 2006-01-09 0.008512737
## 2006-01-10 0.006051871
```

Let's look at the relationship between the pairs of the ETFs.

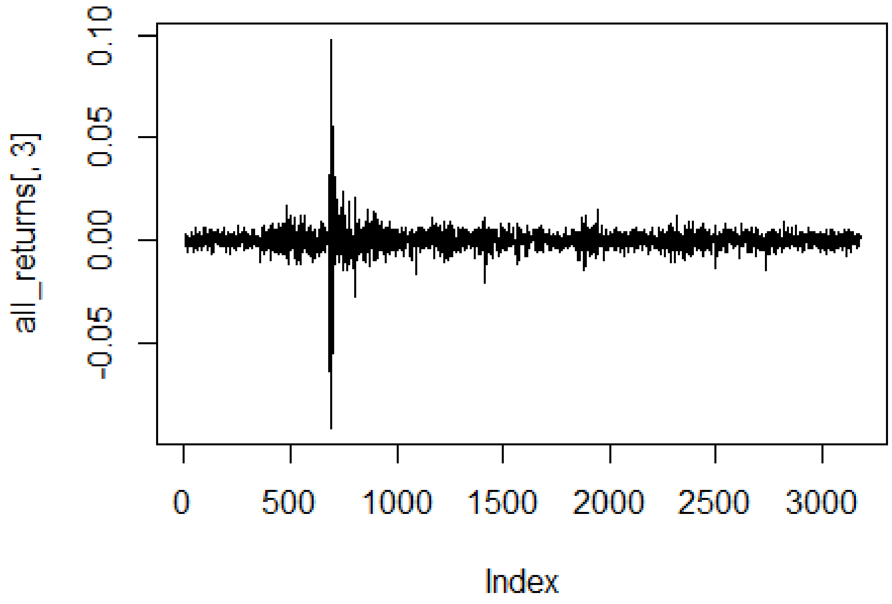
```
pairs(all_returns)
```



We can see that the emerging market equities seems to have a not very variable relationship with the other ETFs, which is inline with what we noted above.

Now I'll pull a plot of all the returns combined in a matrix over time.

```
plot(all_returns[,3], type='l')
```



We can see that again, the variability is really mostly around 2008-2009 when the global economy was pretty week. Otherwise, there are much smaller dips and hills in the dataset.

Let’s look at the correlations

```
cor(all_returns)
```

```
##              ClCl.SPYa  ClCl.TLTa  ClCl.LQDa   ClCl.EEMa   ClCl.VNQa
## ClCl.SPYa  1.0000000 -0.4259381 0.10319281  0.41401685  0.76479827
## ClCl.TLTa -0.4259381  1.0000000 0.44118254 -0.16429890 -0.24634498
## ClCl.LQDa  0.1031928  0.4411825 1.00000000  0.08885671  0.07347216
## ClCl.EEMa  0.4140168 -0.1642989 0.08885671  1.00000000  0.29530802
## ClCl.VNQa  0.7647983 -0.2463450 0.07347216  0.29530802  1.00000000
```

Some of the ETFs are highly correlated and some are not. For example, the stock exchange and real estate market are, and that makes sense because they’re both tied to the health of the US economy. However, treasury bonds are negatively correlated with the performance of the US stocks and emerging markets. This is probably because when the economy is doing well, people will invest in riskier funds, like those in emerging markets and the stock market, but when the economy is doing poorly, people will be more likely to invest in something dependable, like US Treasury bonds.

Now, I have $100,000 to invest, and I need to dedide which portfolio to use.

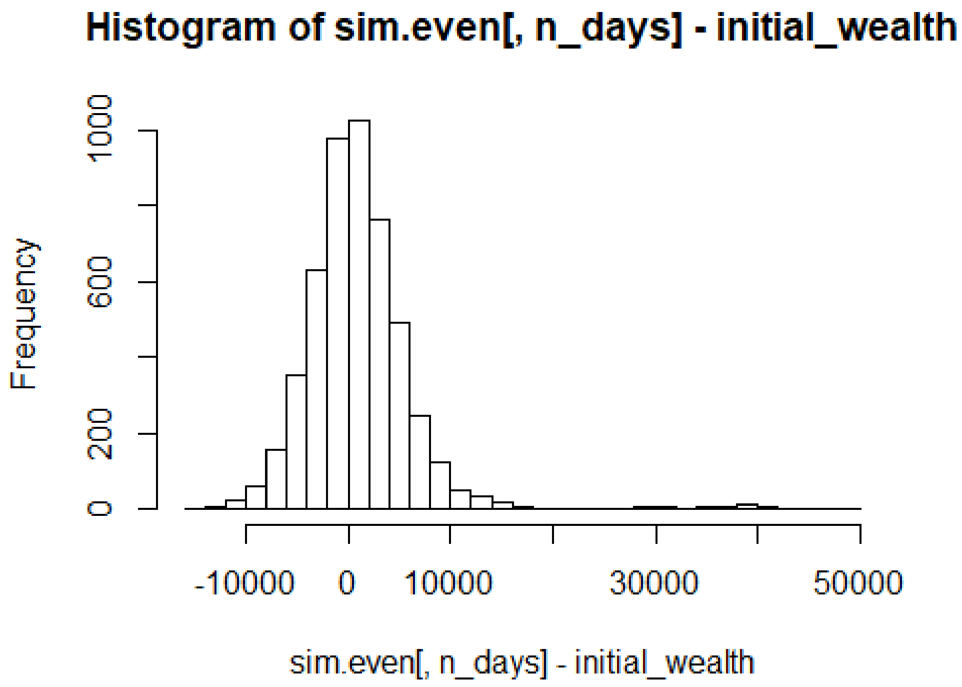
Portfolio #1 - Even split

I will be bootstrapping over the past 4 weeks (20 days), with equal weights on all 5 ETFs after setting the seed.

```
##               [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## result.1 100533.51 101532.82 100996.59 101718.94 101183.65 102117.42
## result.2  94846.91  95379.31  96665.93  97002.82  98207.39  97900.28
## result.3 100277.91  99886.46  99610.45  99389.76 100074.29  98764.28
## result.4 100223.80 100438.60 101343.12 101302.43 101492.33 101260.40
## result.5  99886.89  99277.44  99171.66  98706.60  99039.52  99158.00
## result.6  99522.03  99730.68  98638.21  98175.60  97643.24  98061.54
##               [,7]      [,8]      [,9]     [,10]     [,11]     [,12]
## result.1 102867.81 101716.08 102091.79 102714.64 102714.88 102018.74
## result.2  97873.40  98164.03  98303.91  98212.79  98865.29  98902.42
## result.3  99010.05  98835.94  98833.54  99666.06 100431.67 100718.22
## result.4 100725.21 101072.46 102403.17 103442.01 103664.08 103503.93
## result.5  96009.18  96705.04  96784.05  97052.99  96408.08  95146.05
## result.6  97504.27  97914.29  97976.72  98730.36  99583.56  99532.83
##              [,13]     [,14]     [,15]     [,16]     [,17]     [,18]
## result.1 101827.03 101897.94 102340.61 102322.38 102443.03 103544.81
## result.2  97819.45  98117.57 102271.05 102544.25 102903.87 103512.87
## result.3 100345.51 100110.75  99872.45  99636.37  99664.40  99106.01
## result.4 103959.36 103139.47 102861.30 102924.82 101814.49 101144.41
## result.5  95379.15  93969.18  94496.40  93696.78  94253.33  95073.93
## result.6 100629.22 101549.55 101842.62 101973.90 102371.16 102254.20
##              [,19]     [,20]
## result.1 104018.94 103519.30
## result.2 104076.45 103049.26
## result.3  99724.68  99245.67
## result.4 100749.58 100927.89
## result.5  94940.97  94787.25
## result.6 101797.22 102148.62
```

Looking at the head values of the simulation, we can already see that the results are variable.

Now let’s see the histogram:



Using boostrapping and then creating a histogram of gains and losses, we can see that the most likely outcome with this investment is a small gain. There is a longer tail on the right side of the histogram though, where we can see that there is also a chance of a big “win” in the market with this porfolio. Most likely though, you will either gain or lose a bit of money.

```
mean(sim.even[,n_days])

## [1] 100928.8
```

We can also see the mean for this histogram is $100,991, which agrees with my assessment of the histogram above, where you are more likely to gain money, but probably just a bit.

Now, looking at the 5% level belwo, we can see that 95% of the time you will have better returns than -$6,000, which is good. Out of $100,000, that isn’t that bad of a loss, but you of course still do not want any loss. The good news is that there is only a 5% chance of that loss happening and a 95% chance that there will be even less of a loss or a gain.

```
quantile(sim.even[,n_days], 0.05) - initial_wealth

##        5%
## -5921.591
```

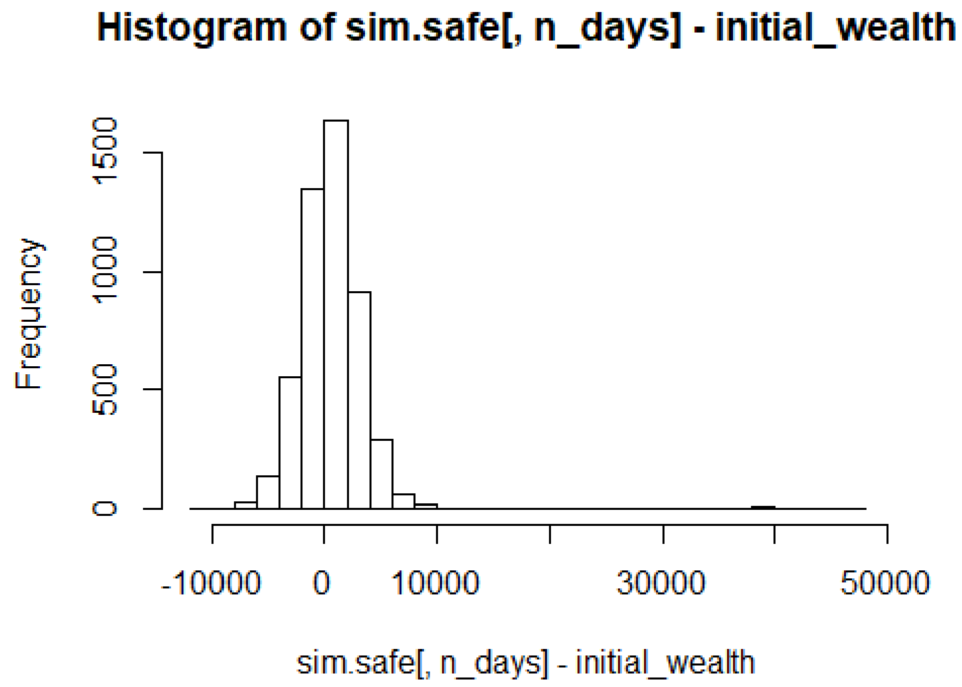
Portfolio #2 - Safe Option

For the “safe” option, I will invest 40% of my money into treasury bonds, 40% into investment-grade corporate bonds, and 20% into emerging market equities.

```
##               [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## result.1  99953.43 100276.17  99723.41  99361.12  99047.17  98464.21
## result.2 100210.91 100118.48  99871.49  99566.84 100111.06 101069.20
## result.3 100007.05  99534.78  99621.99 100063.92  99798.80  99428.23
## result.4  99586.70  99359.07  99429.82  99597.78  98579.56  97729.18
## result.5 100037.91 100147.26 100704.70 101095.50 101223.21 100992.65
## result.6 102785.72 102802.32 102591.66 102763.08 103590.19 103733.11
##               [,7]      [,8]      [,9]     [,10]     [,11]    [,12]
## result.1  98874.88  99750.42 100542.95 100686.85 100896.83 100679.2
## result.2 100478.66 100036.51 100100.22 100121.93 100182.85 100308.3
## result.3  99305.11  99011.70 100074.83 100217.22 100200.01 100128.0
## result.4  97516.24  97638.35  97943.32  97851.99  97560.23  97352.4
## result.5 101480.73 101666.69 100880.40 100043.61 100484.24 100940.1
## result.6 103713.44 103103.78 102742.53 102888.63 103087.67 102841.8
##              [,13]     [,14]     [,15]     [,16]     [,17]     [,18]
## result.1 101085.64 101204.17 101513.46 101720.39 102054.15 102486.30
## result.2 100630.03 100844.19 101179.55 101511.52 102582.33 102254.08
## result.3 100372.33 100322.78 100094.99 100364.72 100733.41 100874.05
## result.4  97969.95  97772.76  97595.69  97383.35  96824.08  98020.77
## result.5 101172.39 101511.47 102010.10 101631.32 102083.49 102220.79
## result.6 103783.27 104023.90 103879.34 103924.14 104216.93 103851.21
##              [,19]     [,20]
## result.1 102529.54 101840.85
## result.2 102804.42 102318.75
## result.3 101325.61 101464.19
## result.4  97481.84  97620.09
## result.5 101350.64 101354.93
## result.6 104178.48 104314.06
```

Looking at the head of the safe porfolio, I can’t really tell a difference, so let’s look at the histogram and see if there is something there.

```
hist(sim.safe[,n_days]- initial_wealth, breaks=30)
```



It definitely appears that you have more of a chance of gaining than losing. Moreover, there's a bit of a wacky long tail here, with just a few of the simulations gaining a LOT of money, but none losing a LOT of money.

```
mean(sim.safe[,n_days])

## [1] 100698.5
```

The mean gain for this safe simulation was $100,784, which is a bit less than the even split. This is actually as expected, because of the old saying "high risk, high reward". In this case, we went with "low risk, low reward".

Now, looking at the 5% chance on the left tail, we can see that there is a 95% chance that we lose less than -$3,000 or that we gain money. This is half the amount that we were at risk of losing at the same percentage for the evenly split porfolio, which again makes sense with the "low risk, low reward" mantra. This is the low risk part.

```
quantile(sim.safe[,n_days], 0.05) - initial_wealth

##        5% 
## -3440.112
```

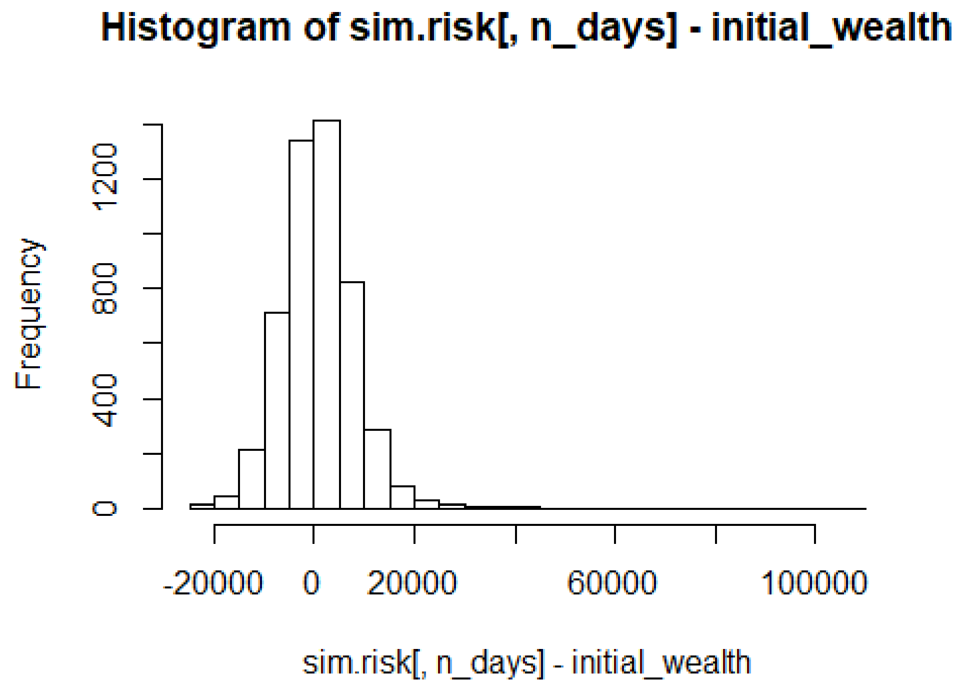
Porfolio #3 - Aggressive Option

For the "aggressive" option, I will invest 40% in US domestic equities, 40% in real estate, and 20% in emerging market equities.

```
##                [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## result.1 104477.23 104787.04 104445.14 107531.72 108550.10 109102.27
## result.2 100004.46  99763.59  96522.63 101811.98 101275.65 102363.31
## result.3 100430.23  97652.99  97409.74  97094.68  96825.61  96796.03
## result.4  99510.98  98859.83  98944.62  99116.76  99712.38  99903.43
## result.5 100196.51 100139.69  96886.52  94512.41  96364.73  96626.49
## result.6  99811.62  99571.21  99910.25  98783.31  96726.13  97321.97
##                [,7]      [,8]      [,9]     [,10]     [,11]     [,12]
## result.1 109655.67 109755.19 108463.64 110201.89 109072.12 108564.59
## result.2 102429.71 102714.06  99012.88  98948.89 101691.70 101897.73
## result.3  96260.11  96513.84  96606.88  96730.46  97509.63  97813.24
## result.4 100160.49  99951.66 100296.06 100046.23  97039.38  96435.82
## result.5  97577.07  97432.71  98012.36  97454.59  97392.19  98743.02
## result.6  98252.08  98337.37  94947.56  95984.05  97190.89  96561.08
##               [,13]     [,14]     [,15]     [,16]     [,17]     [,18]
## result.1 107085.55 106995.71 110895.69 112420.01 110702.01 110888.21
## result.2 103931.31 104578.02 104359.44 104528.24 107133.70 105923.77
## result.3  98206.12  97643.34  97489.40  97199.02  94189.72  93231.83
## result.4  95977.41  96626.93  97294.42  97157.07  98467.26  98457.98
## result.5  98918.58  98261.17 100117.15 100253.48  95727.76  95669.05
## result.6  96235.79  95540.45  92612.29  93060.69  93987.69  94542.56
##               [,19]     [,20]
## result.1 110475.94 110500.41
## result.2 106208.02 105502.16
## result.3  93555.76  92481.90
## result.4 105030.05 105602.35
## result.5  94433.66  97041.58
## result.6  95902.42  97500.63
```

At first glance, I can see at least one row where the end wealth is in the $800,000’s, which I don’t believe I saw for either of the previous porfolio scenarios.

```
hist(sim.risk[,n_days]- initial_wealth, breaks=30)
```



Looking at the risky histogram, it seems somewhat similar to the first scenario, where the histogram is somewhat normal, but instead of a few random marks way on the right side, there is more of a connected tail to the main histogram area.

```
mean(sim.risk[,n_days])

## [1] 100890.5
```

The mean return is the highest we've seen, actually at $100,972. I presume this is from the long tail mostly, because looking at the histogram, it looks like there's a chance to lose about $200,000, which would be very bad.

```
quantile(sim.risk[,n_days], 0.05) - initial_wealth

##        5% 
## -10250.64
```

This is confirmed at the 5% quantile, which says that there is a 5% chance of losing about $10,500, which is much worse than the other options as expected.

Summary

Looking at the mean returns and 5% risks, I think it would be easy for an investor to make a decision between the three portfolios depending on how much risk she is comfortable with. we saw that the "high risk, high reward" mantra held true between the three porfolios, and risk increased with the more aggressive porfolios.

Market Segmentation

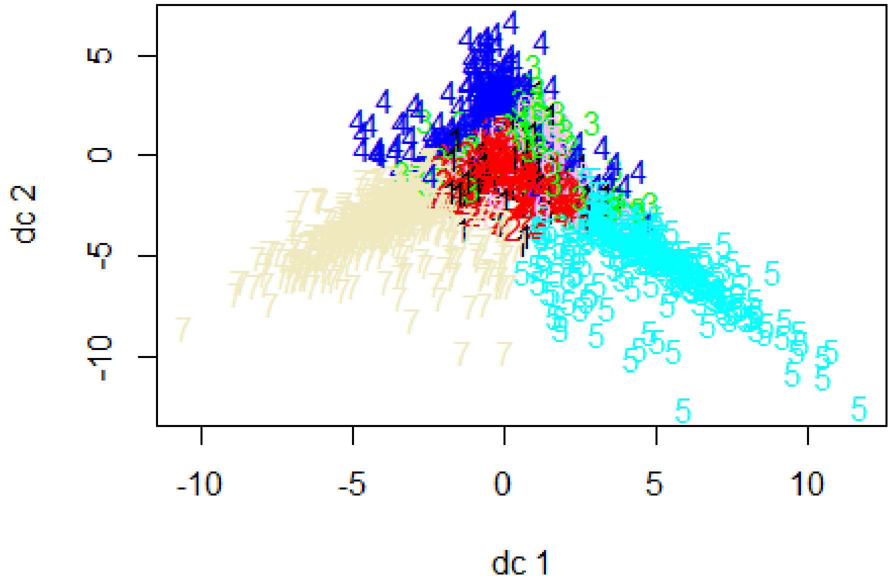
Load the correct libraries.

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

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

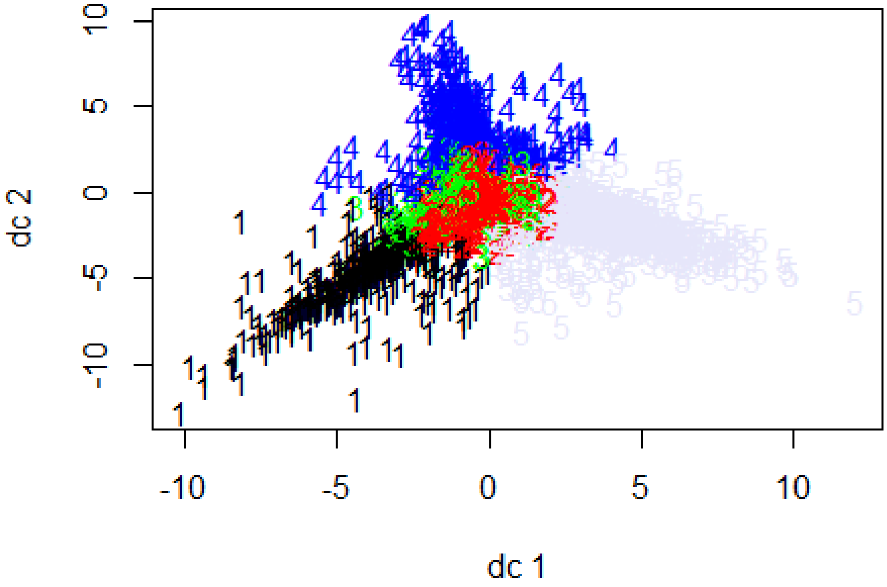
Read in data.

K Means Clustering

Now, I'm going to set up the data to make the tweets the dependent variables, and create a k means plot with 7 clusters, because that's a reasonable amount of gropus to segment customers for marketing.

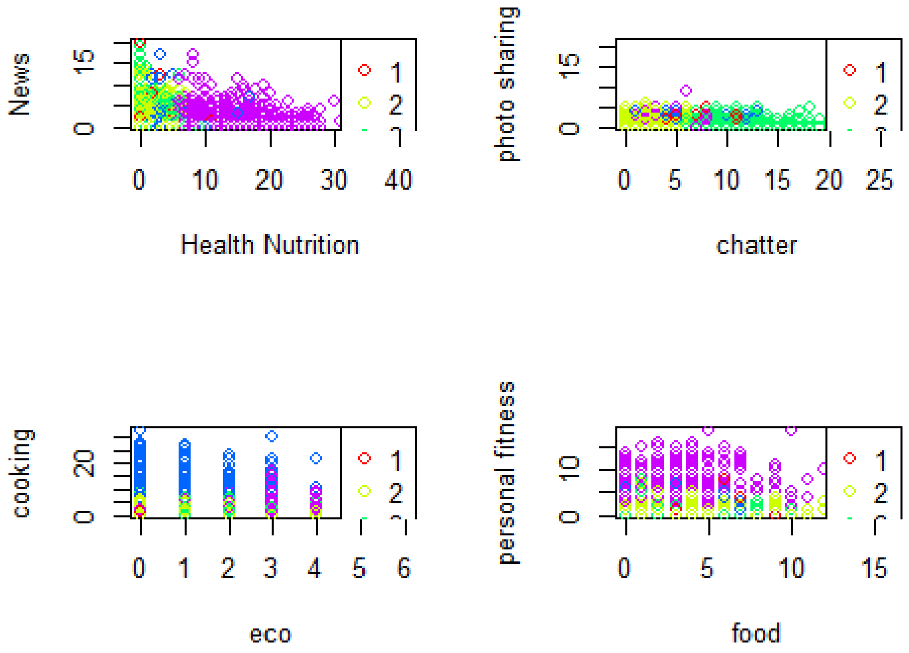


Seven is clearly way too many clusters and non-sensical. It seems that there really should only be 4 or 5. We should proabably do 5 because segementing customers into 4 groups is not super helpful for marketing



Five looks fine, with three very clear clusters and two around the center.

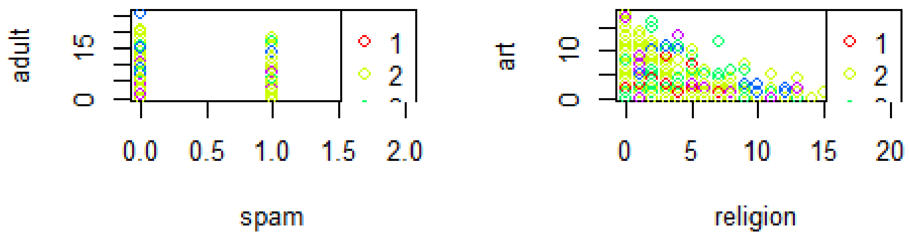
Let's look at some plots to describe the clusters in ways which might be helpful for segmentation:

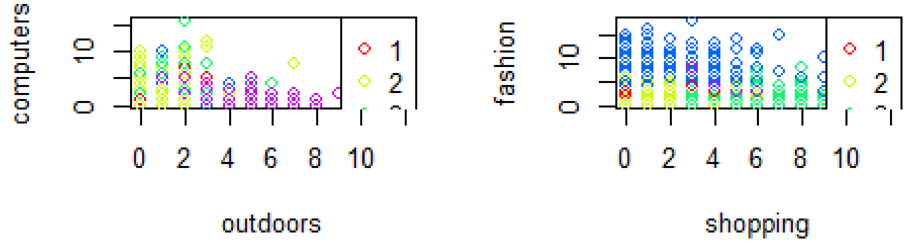


Right away we can see that cluster 5 has a lot of tweets about health nutrition and personal fitness, and a good bit about food as well. This group seems likely to be a good fit for our product. Let's call cluster 5 "Fitness Folks", because it is often helpful to come up with a catchy name for each segmentation. Here, I am segmenting based on interests.

Another group that pops out is cluster 4 - these people tweet a lot about cooking, but I think we can find more info about them.

Cluster 3 could be interesting as well - these people are high in chatter and have a good amount of news as well, so they could be worth looking into for spam tweets. Let's look a bit more.

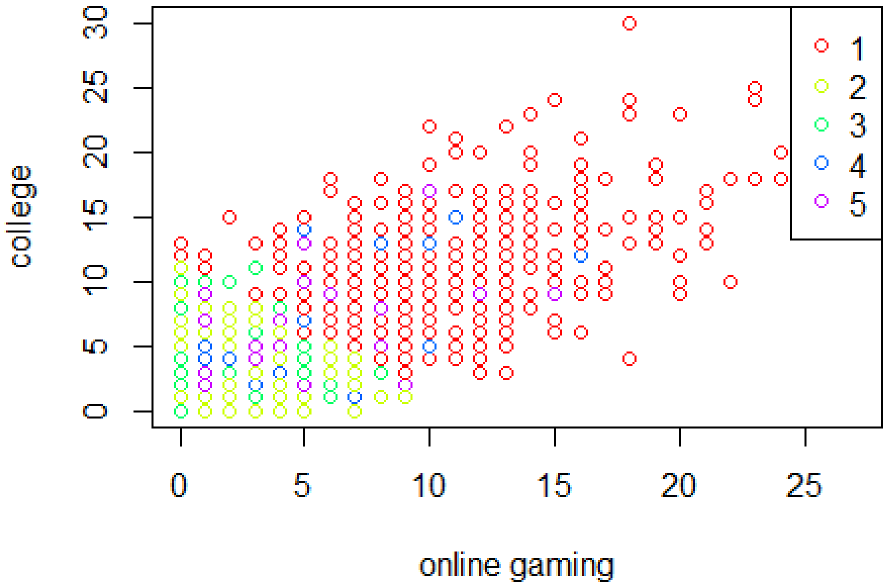




Cluster 4 stands out here for fashion and shopping. This fits in with the results above that showed they were interested in cooking. Perhaps these people are mostly lifestyle bloggers or "influencers" who try to tweet about the lastest trends. We'll go with that and call cluster 4 "Lifestyle Bloggers".

Cluster 3 is not mostly spam as I suspected, but it is a lot of shopping accounts. These could be other brands that follow the NutrientH20 brand. We'll call cluster 3 "Corporate".

That only leaves 1 and 2 to figure out. Cluster 3 seems to be all over the place, including a lot of adult and spam and art content, but also computers. For now, we'll consider Cluster 2 "Random/Spam". This even makes sense because cluster 2 was centered in the plot cluster graph. That leaves cluster 1, which did not seem to stand out in any of the graphs we have looked at. I want to try one more plot to check on cluster 1:



As I suspected, a LOT of 1's popped up here. Let's consider them "Gamers".

Okay, so we have our clusters now:

1 - Gamers

2 - Random/Spam

3 - Corporate

4 - Lifestyle Bloggers

5 - Fitness Folks

This is more than enough information to do at least some targeted marketing. We know who could be "influencers" and convince others to use our brand, and we know hobbies of others, like the gamers. If our nutrientH20 product is a health drink, we also know that the fitness folks could be very helful to target, as they may be interested in a healthy product.

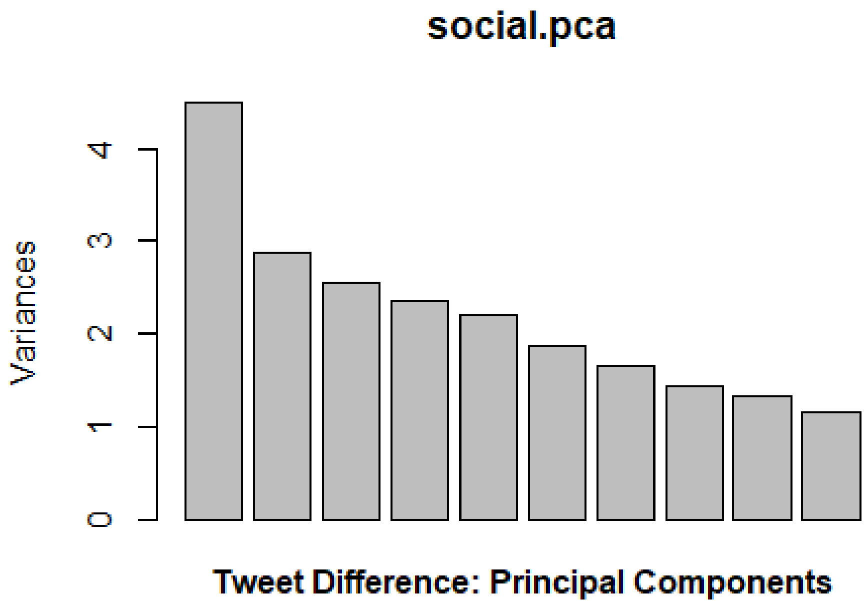
And, for cluster 3, we know we need to investigate a little further before making decisions on them.

HClust

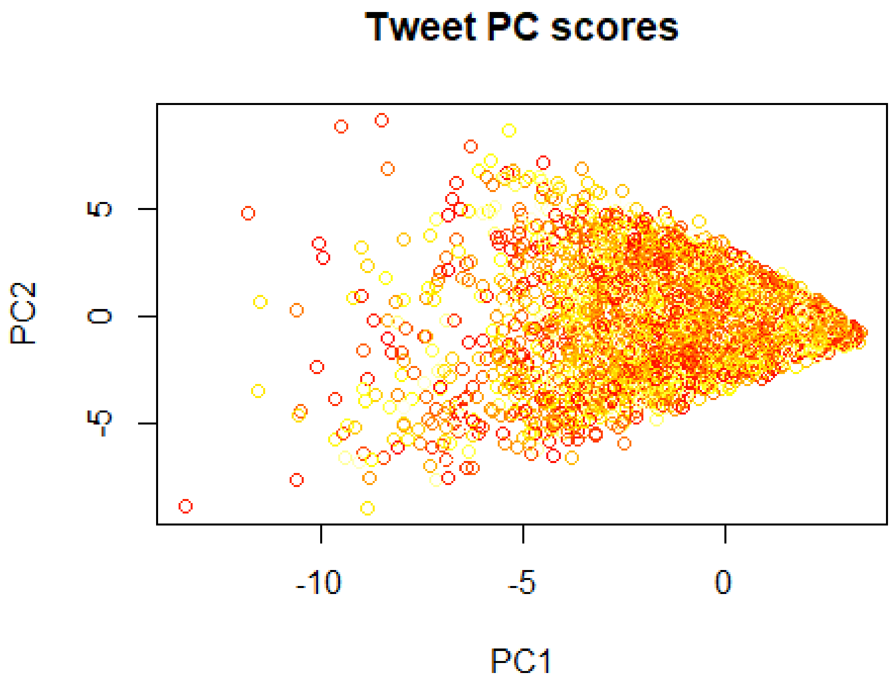
I think the K Means worked pretty well for us, but let's look at some other models just to be sure.

Set up:

Here, let's plot the variances that can be explained by each principal component.



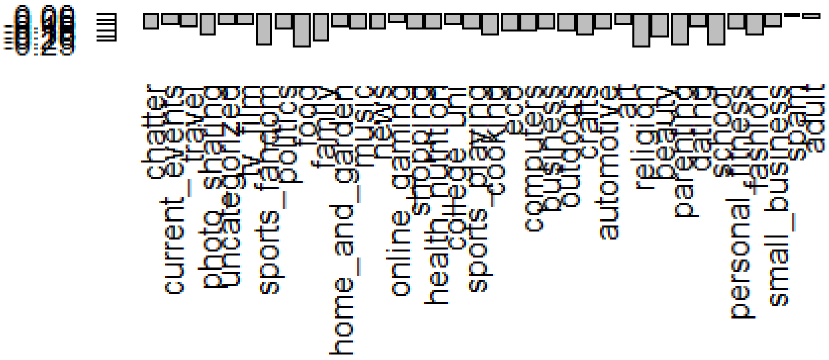
We can see that most of the variance is explained in the first principal component, and then there's diminishing returns from there on out. Let's use two.

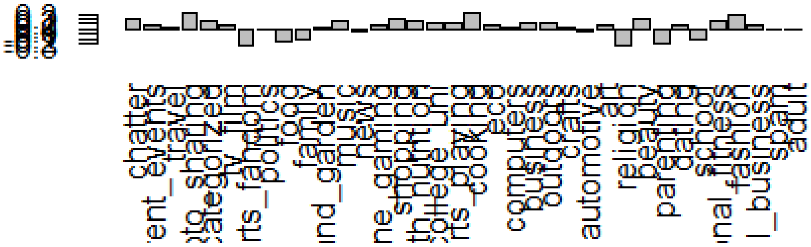


With this graph plotting the first two principal components, we can see that most points are clustered really close in a positive value. The rest are pretty evenly spread out in the highly negative area.

Let’s look at what might be grouping these points:

```
par(mfrow=c(2,1))
barplot(social.pca$rotation[,1], las=2)
barplot(social.pca$rotation[,2], las=2)
```





For the first graph, it seems that personal tweets - parenting, sports, religion are the most negative. These are personal, but they are also some of the most controversial subjects. That could be some sort of category - people who are active with very personal subjects are in this category of negative variance.

Spam and adult tweets (which are spam) are the least negative, so they are not personal at all. I think these seem like pretty safe categories for the first principle component.

For the second principle component, the same personal subjects are negative, but photo-sharing, cooking, fashion and beauty are the most positive. This is a hard distinction to make, because those are personal too. Perhaps the negative tweets are for friends, while the positive tweets are for the public at large. Again, more like what an "influencer" would tweet, while the negative group would be what any social media user would tweet.

What these categories are basically is HOW people use twitter, whereas the kmeans is based more on interests.

In conclusion, I would hand the k means clusters to the NutrientH20 group with the conclusions supported above in the k means section.