STAS380 Exercise 1 - Raksha Pai, Kyle Katzen, Bhavana Vijay, Jake Schmidt

August 9, 2017

Probability Practice

Part A

$Prob(RC) = 0.3$

$Prob(RC/Yes) = Prob(RC/No) = 0.5$

$Prob(TC) = 0.7$

$Prob(Yes) = 0.65$

$Prob(No) - 0.35$

$Prob(TC/Yes) =?$

$Prob(Yes) = Prob(RC/Yes) * Prob(RC) + Prob(TC/Yes) * Prob(TC)$

$0.65 = 0.5 * 0.3 + Prob(TC/Yes) * 0.7$

$Prob(TC/Yes) = **0.7142857**$

Therefore, the fraction of people who are truthful clickers and answered yes is 0.71

Part B

$Prob(Positive/Disease) = 0.993$

$Prob(Negative/NoDisease) = 0.9999$

$Prob(Disease) = 0.000025$

$Prob(Disease/Positive) = Prob(Disease)*Prob(Positive/Disease)/[Prob(Disease)*Prob(Positive/Disease)+ Prob(NoDisease) * Prob(Positive/NoDisease)$

$Prob(Disease/Positve) = 0.000025 * 0.993/[0.000025 * 0.993 + (1 - 0.000025) * (1 - 0.9999)]$

$Prob(Disease/Positve) = **0.1988824**$

Therefore, The probability of someone having a disease having tested positive is about 20%

In implementing a universal testing policy, there might be problems as the test accurately testing a disease positive is only about 1/5ths. Hence, there are about 80% positive results that are false positives.

Exploratory analysis: green buildings

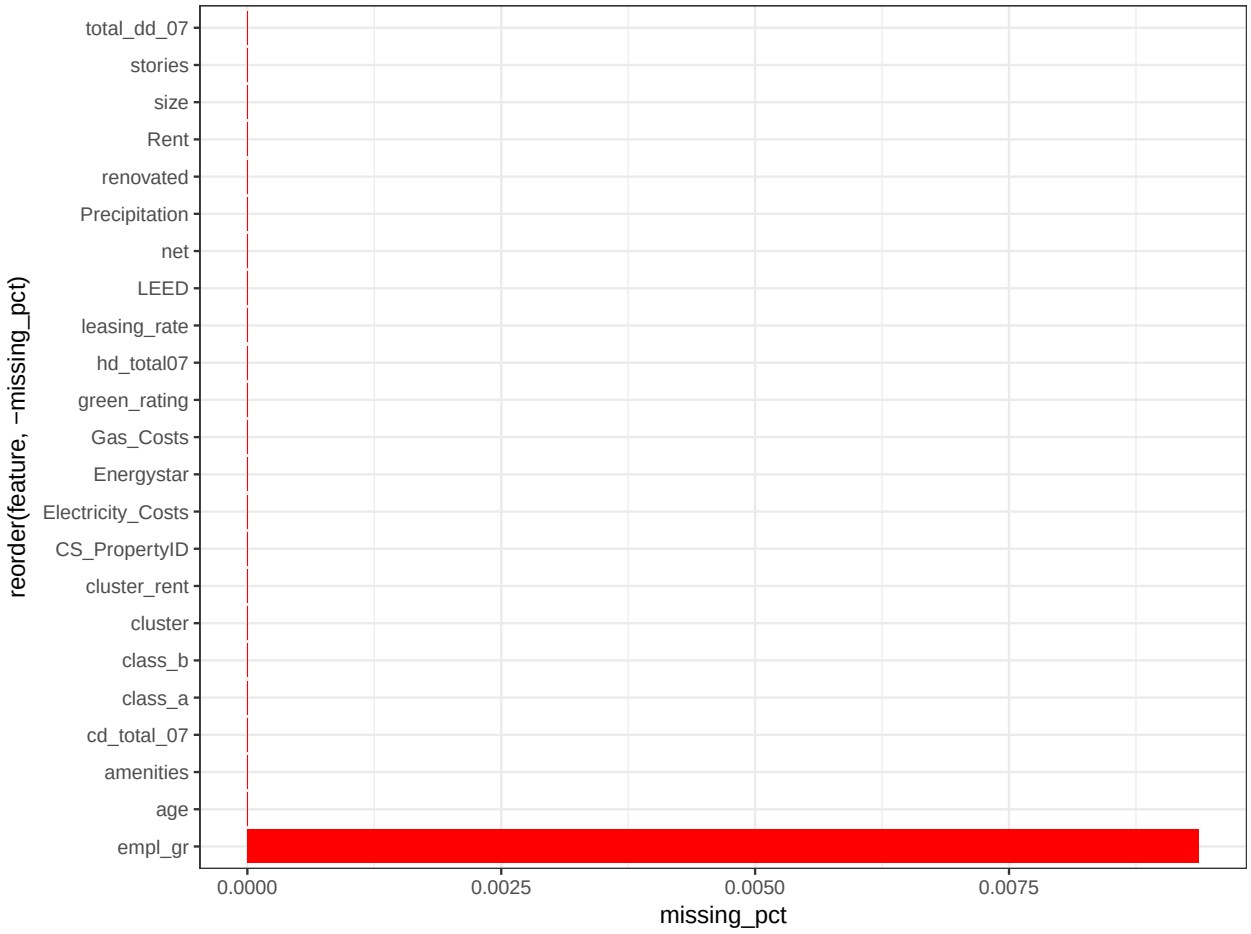
Reading in the data and plotting missing values

```
green = read.csv(url("https://raw.githubusercontent.com/jgscott/STA380/master/data/greenbuildings.csv"))
summary(green)
```

```
 CS_PropertyID        cluster            size            empl_gr
 Min.   :      1   Min.   :   1.0   Min.   :   1624   Min.   :-24.950
 1st Qu.: 157452   1st Qu.: 272.0   1st Qu.:  50891   1st Qu.:  1.740
 Median : 313253   Median : 476.0   Median : 128838   Median :  1.970
 Mean   : 453003   Mean   : 588.6   Mean   : 234638   Mean   :  3.207
 3rd Qu.: 441188   3rd Qu.:1044.0   3rd Qu.: 294212   3rd Qu.:  2.380
 Max.   :6208103   Max.   :1230.0   Max.   :3781045   Max.   : 67.780
                                                      NA's   :74
      Rent         leasing_rate       stories            age
 Min.   :  2.98   Min.   :  0.00   Min.   :  1.00   Min.   :  0.00
 1st Qu.: 19.50   1st Qu.: 77.85   1st Qu.:  4.00   1st Qu.: 23.00
 Median : 25.16   Median : 89.53   Median : 10.00   Median : 34.00
 Mean   : 28.42   Mean   : 82.61   Mean   : 13.58   Mean   : 47.24
 3rd Qu.: 34.18   3rd Qu.: 96.44   3rd Qu.: 19.00   3rd Qu.: 79.00
 Max.   :250.00   Max.   :100.00   Max.   :110.00   Max.   :187.00

   renovated          class_a          class_b            LEED
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.000000
 Median :0.0000   Median :0.0000   Median :0.0000   Median :0.000000
 Mean   :0.3795   Mean   :0.3999   Mean   :0.4595   Mean   :0.006841
 3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.000000
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000000

   Energystar       green_rating          net            amenities
 Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
 1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
 Median :0.00000   Median :0.00000   Median :0.00000   Median :1.0000
 Mean   :0.08082   Mean   :0.08677   Mean   :0.03471   Mean   :0.5266
 3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
 Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000

  cd_total_07     hd_total07    total_dd_07   Precipitation
 Min.   :  39   Min.   :   0   Min.   :2103   Min.   :10.46
 1st Qu.: 684   1st Qu.:1419   1st Qu.:2869   1st Qu.:22.71
 Median : 966   Median :2739   Median :4979   Median :23.16
 Mean   :1229   Mean   :3432   Mean   :4661   Mean   :31.08
 3rd Qu.:1620   3rd Qu.:4796   3rd Qu.:6413   3rd Qu.:43.89
 Max.   :5240   Max.   :7200   Max.   :8244   Max.   :58.02

   Gas_Costs        Electricity_Costs  cluster_rent
 Min.   :0.009487   Min.   :0.01780   Min.   : 9.00
 1st Qu.:0.010296   1st Qu.:0.02330   1st Qu.:20.00
 Median :0.010296   Median :0.03274   Median :25.14
 Mean   :0.011336   Mean   :0.03096   Mean   :27.50
 3rd Qu.:0.011816   3rd Qu.:0.03781   3rd Qu.:34.00
 Max.   :0.028914   Max.   :0.06280   Max.   :71.44
```

```r
green$green_rating = as.factor(green$green_rating)

missing_values <- green %>% summarize_all(funs(sum(is.na(.))/n()))

missing_values <- gather(missing_values, key = "feature", value = "missing_pct")
```

```
missing_values %>% ggplot(aes(x = reorder(feature, -missing_pct), y = missing_pct)) +
    geom_bar(stat = "identity", fill = "red") + coord_flip() + theme_bw()
```

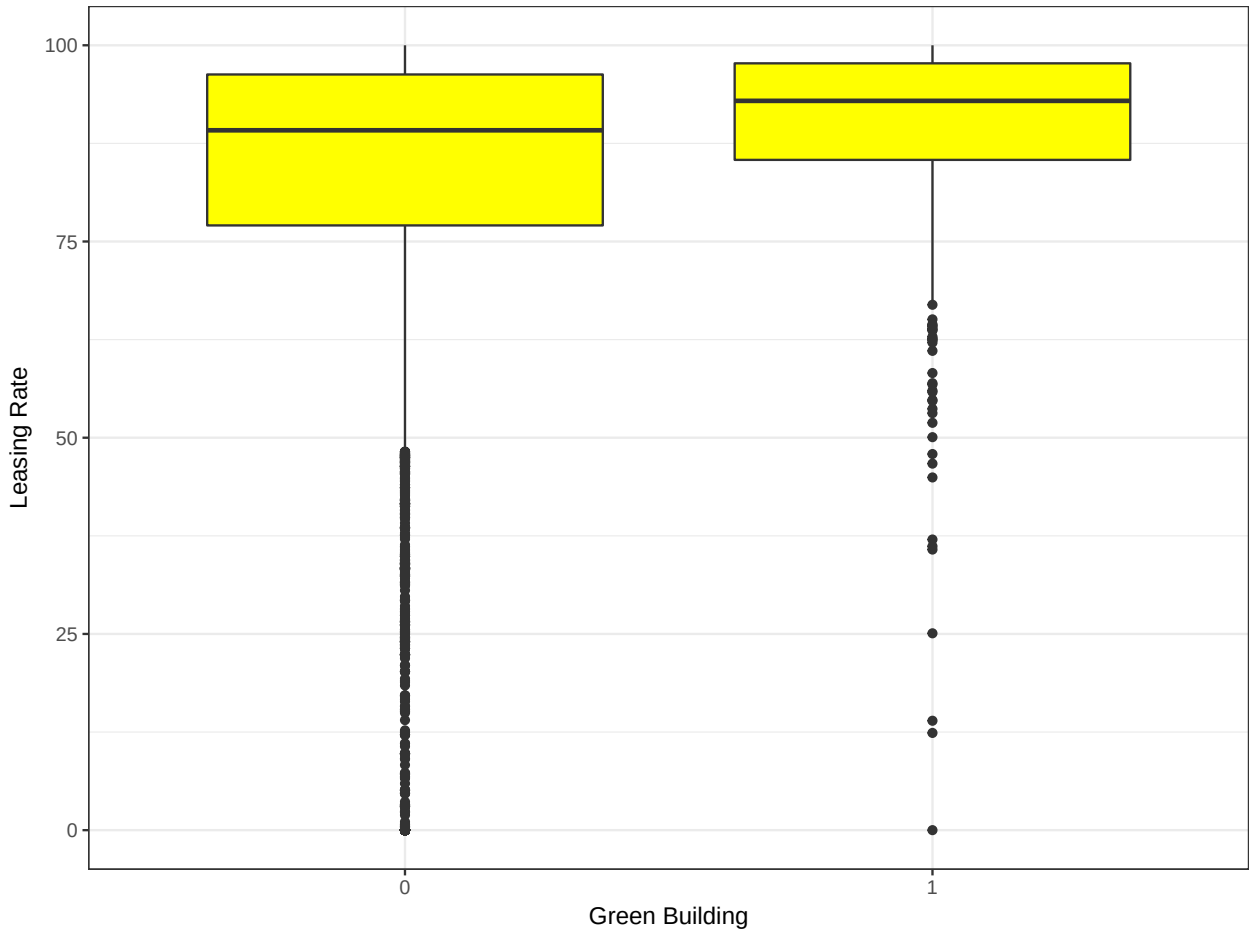


Only empl_gr has 70 missing values. Let's fill it up by the mean.

```
green$empl_gr[is.na(green$empl_gr)] = mean(green$empl_gr, na.rm = TRUE)
```

Data Cleaning - Outliers | Leasing Rate

```
ggplot(green, aes(x = green$green_rating, y = green$leasing_rate), fill = green$green_rating) +
    geom_boxplot(fill = "yellow") + labs(x = "Green Building", y = "Leasing Rate") +
    theme_bw()
```



There are some buildings which have a leasing rate of less than 10% and these are the outliers.

Removing outliers based on leasing rate

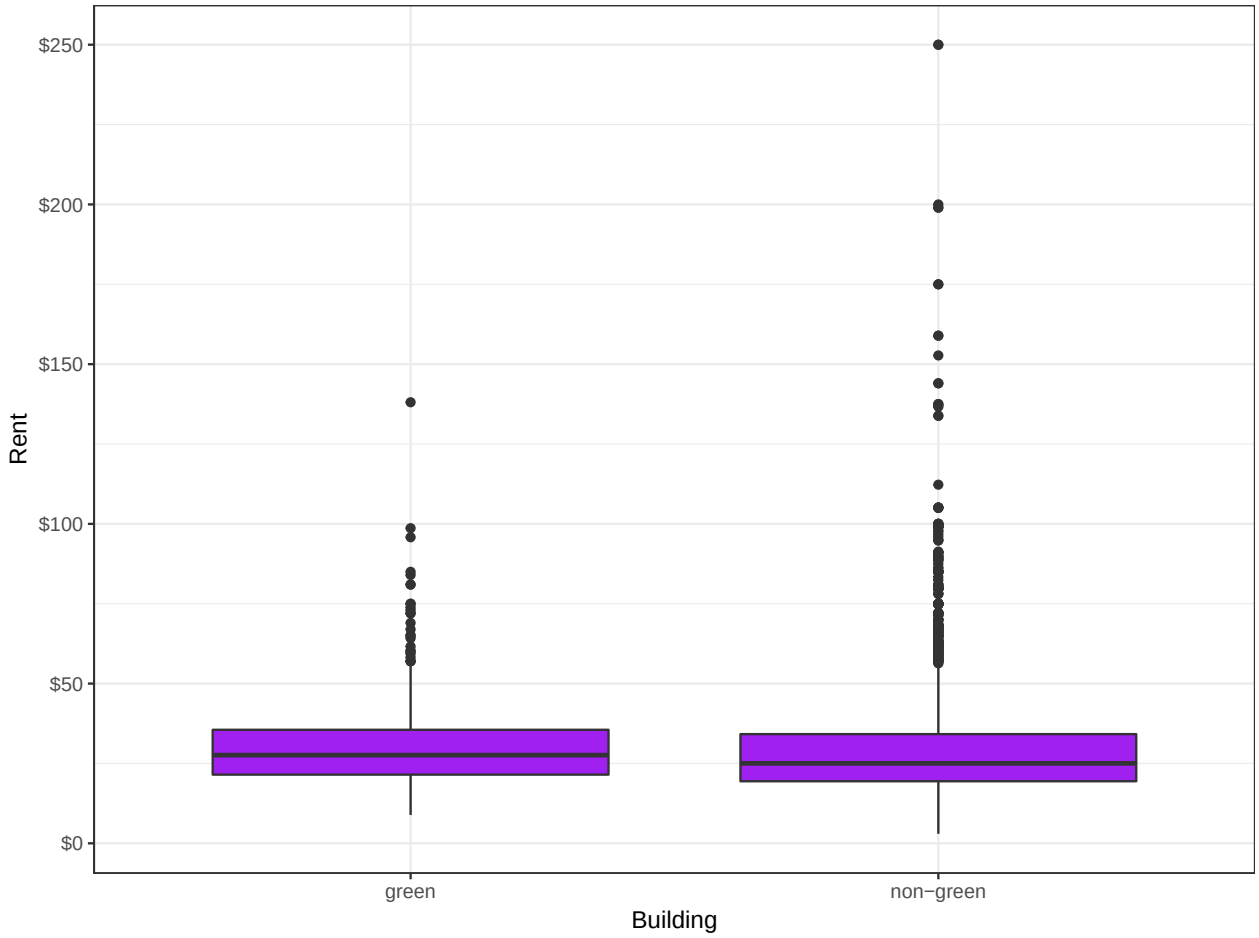
```
data = green[!green$leasing_rate < 10, ]
```

```
data$green_rating_type[data$green_rating == 1] <- "green"
data$green_rating_type[data$green_rating == 0] <- "non-green"
data$green_rating_type = as.factor(data$green_rating_type)
```

Rent (per sq footage)

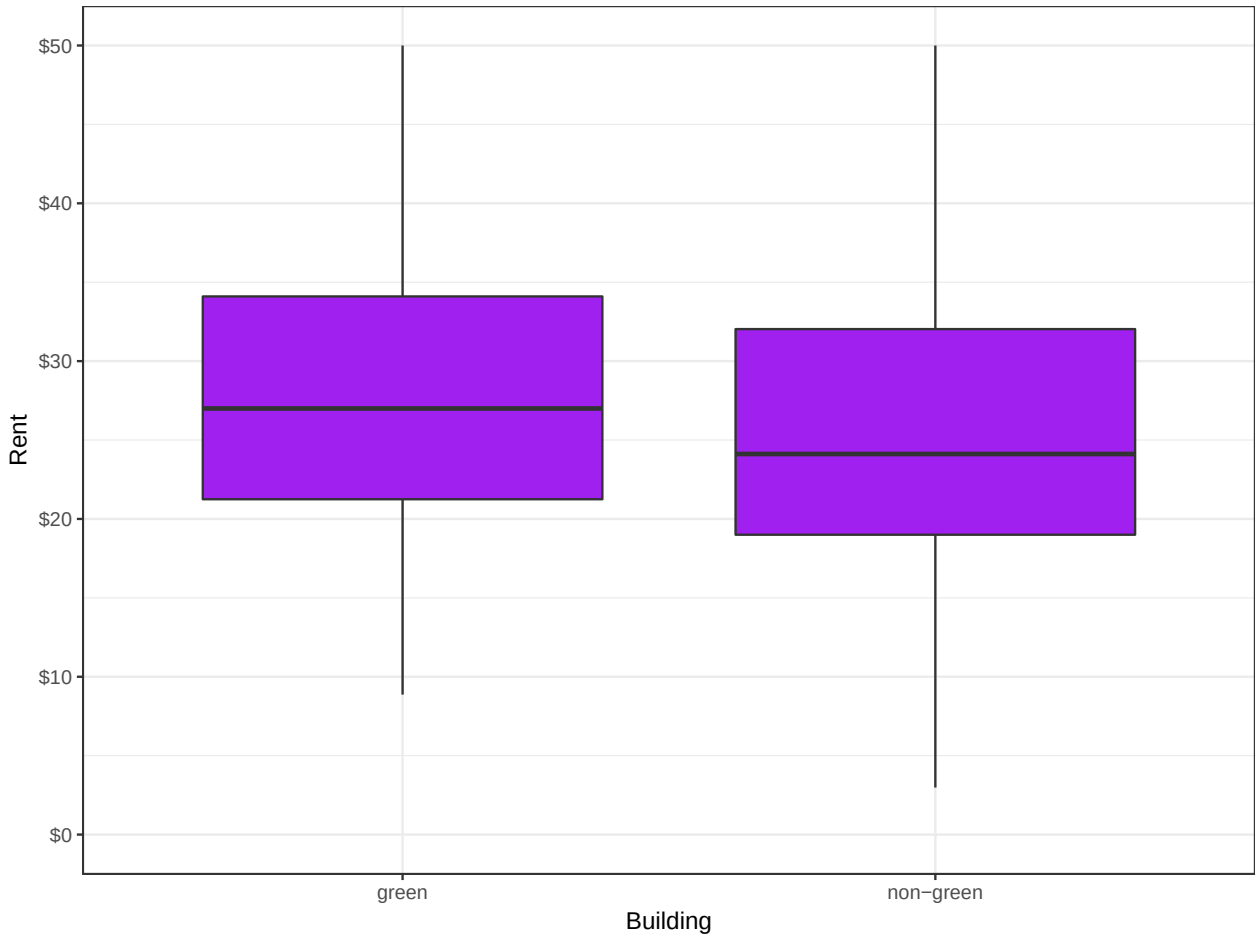
With Outliers

```
ggplot(data, aes(x = data$green_rating_type, y = data$Rent)) + geom_boxplot(fill = "purple") +
    scale_y_continuous(labels = dollar_format()) + labs(x = "Building", y = "Rent") +
    theme_bw()
```



Without Outliers

```
ggplot(data, aes(x = data$green_rating_type, y = data$Rent)) + geom_boxplot(fill = "purple") +
    scale_y_continuous(labels = dollar_format(), limit = c(0, 50)) + labs(x = "Building",
    y = "Rent") + theme_bw()
```



The median rent of Green building is 27.6 and that of a Non-Green Building is 25 per square feet.

Important Features

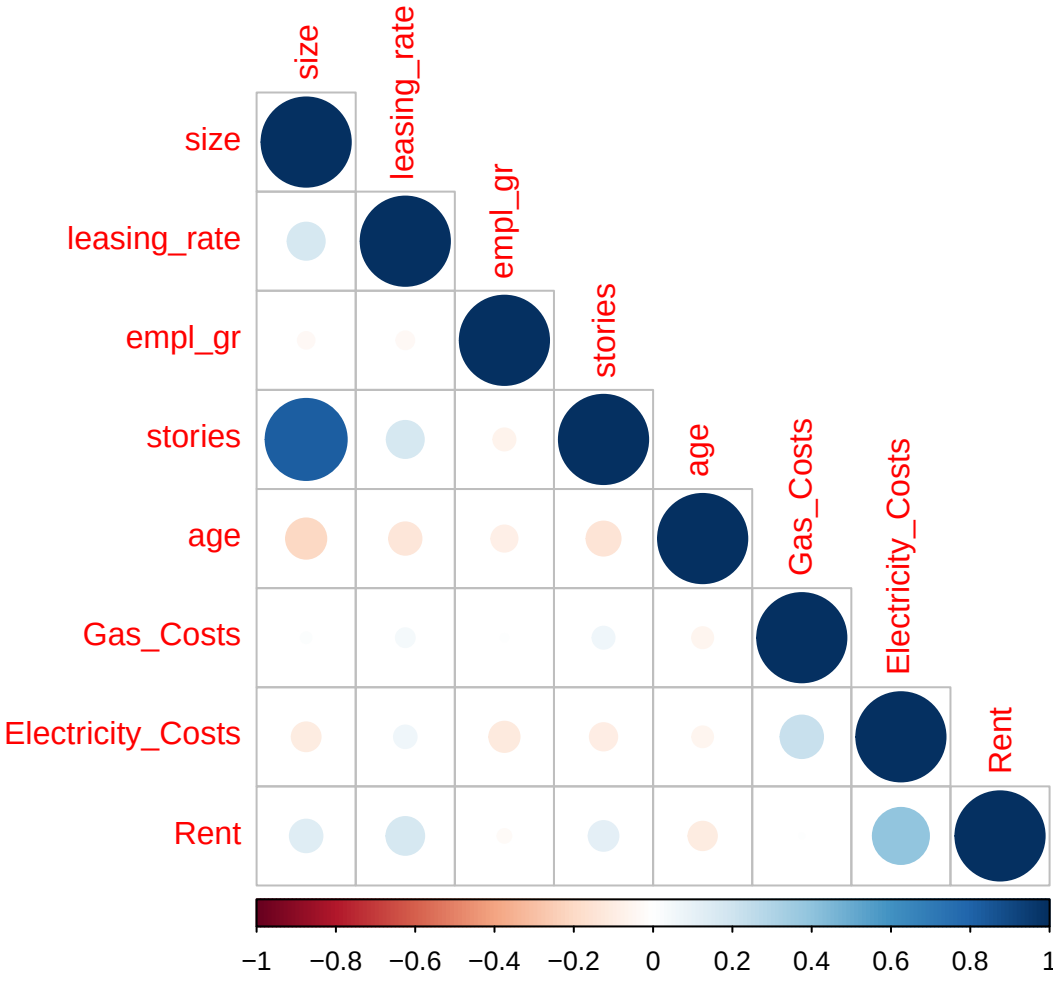
```
fit = lm(Rent ~ size + leasing_rate + stories + age + class_a + class_b + green_rating +
    amenities + Gas_Costs + Electricity_Costs + renovated, data = data)
summary(fit)
```

```
Call:
lm(formula = Rent ~ size + leasing_rate + stories + age + class_a +
    class_b + green_rating + amenities + Gas_Costs + Electricity_Costs +
    renovated, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-34.131  -7.757  -1.878   4.878 199.720

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       1.074e-01  1.243e+00   0.086 0.931110
size              6.079e-06  9.171e-07   6.629 3.61e-11 ***
leasing_rate      1.056e-01  9.383e-03  11.254  < 2e-16 ***
```

```
stories           -2.794e-02  2.258e-02  -1.237 0.216033
age                3.397e-02  6.376e-03   5.329 1.02e-07 ***
class_a            6.165e+00  6.199e-01   9.945  < 2e-16 ***
class_b            1.992e+00  4.931e-01   4.039 5.43e-05 ***
green_rating1     -1.929e+00  5.568e-01  -3.464 0.000536 ***
amenities          2.578e-01  3.516e-01   0.733 0.463370
Gas_Costs         -6.989e+02  6.630e+01 -10.541  < 2e-16 ***
Electricity_Costs  7.257e+02  1.878e+01  38.649  < 2e-16 ***
renovated         -2.874e+00  3.596e-01  -7.991 1.53e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.27 on 7667 degrees of freedom
Multiple R-squared:  0.2325,    Adjusted R-squared:  0.2314
F-statistic: 211.2 on 11 and 7667 DF,  p-value: < 2.2e-16
```

Correlation

```
vars <- data[c("size", "leasing_rate", "empl_gr", "stories", "age", "Gas_Costs",
    "Electricity_Costs", "Rent")]
corrplot(cor(vars, use = "complete.obs"), type = "lower")
```



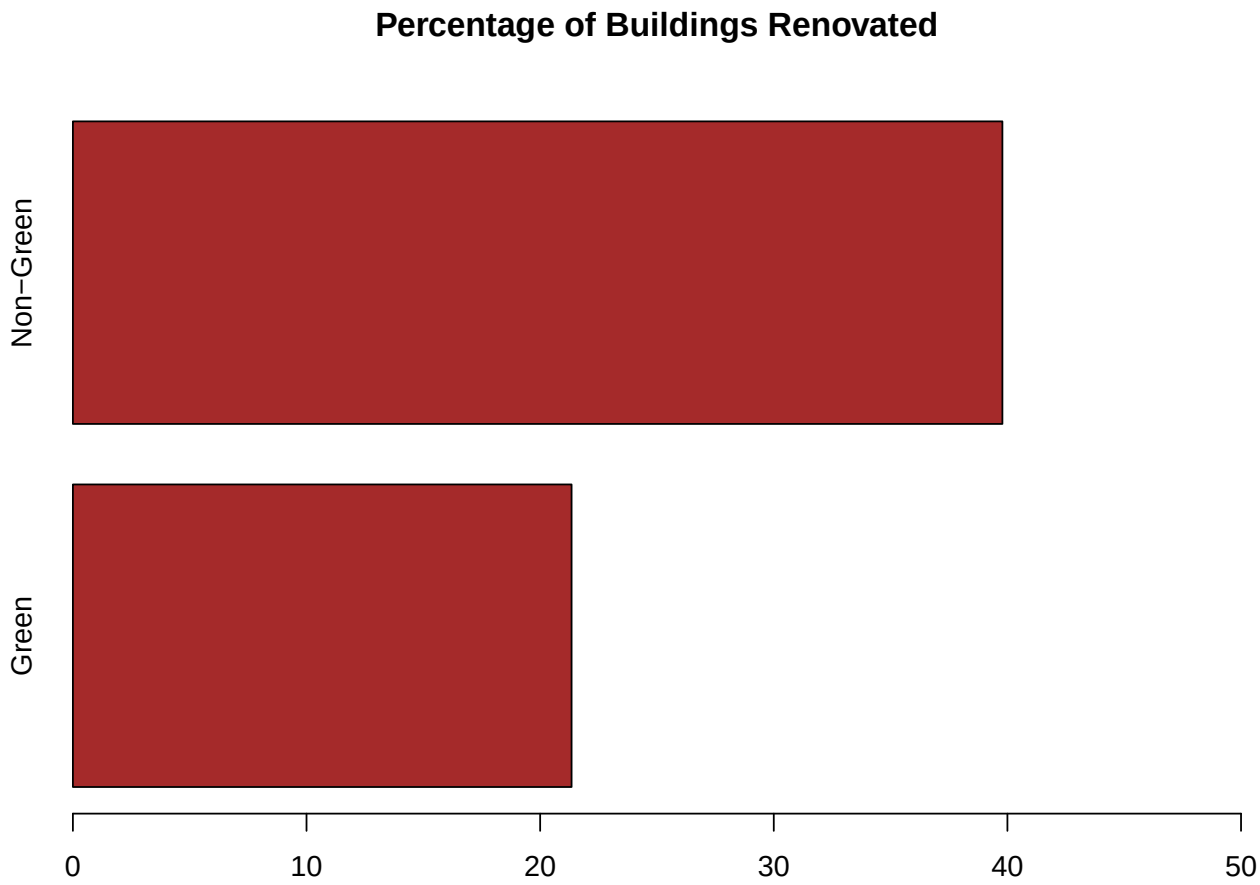
We see that rent has a correlation with leasing rate, electricity costs.

Renovation

```
gr = filter(data, green_rating == "1")
gr1 = nrow(filter(gr, renovated == 1))
green_renovated = gr1/nrow(gr)
green_renovated_perc = green_renovated * 100
```

```
ng = filter(data, green_rating == "0")
ng1 = nrow(filter(ng, renovated == 1))
ngreen_renovated = ng1/nrow(ng)
ngreen_renovated_perc = ngreen_renovated * 100
```

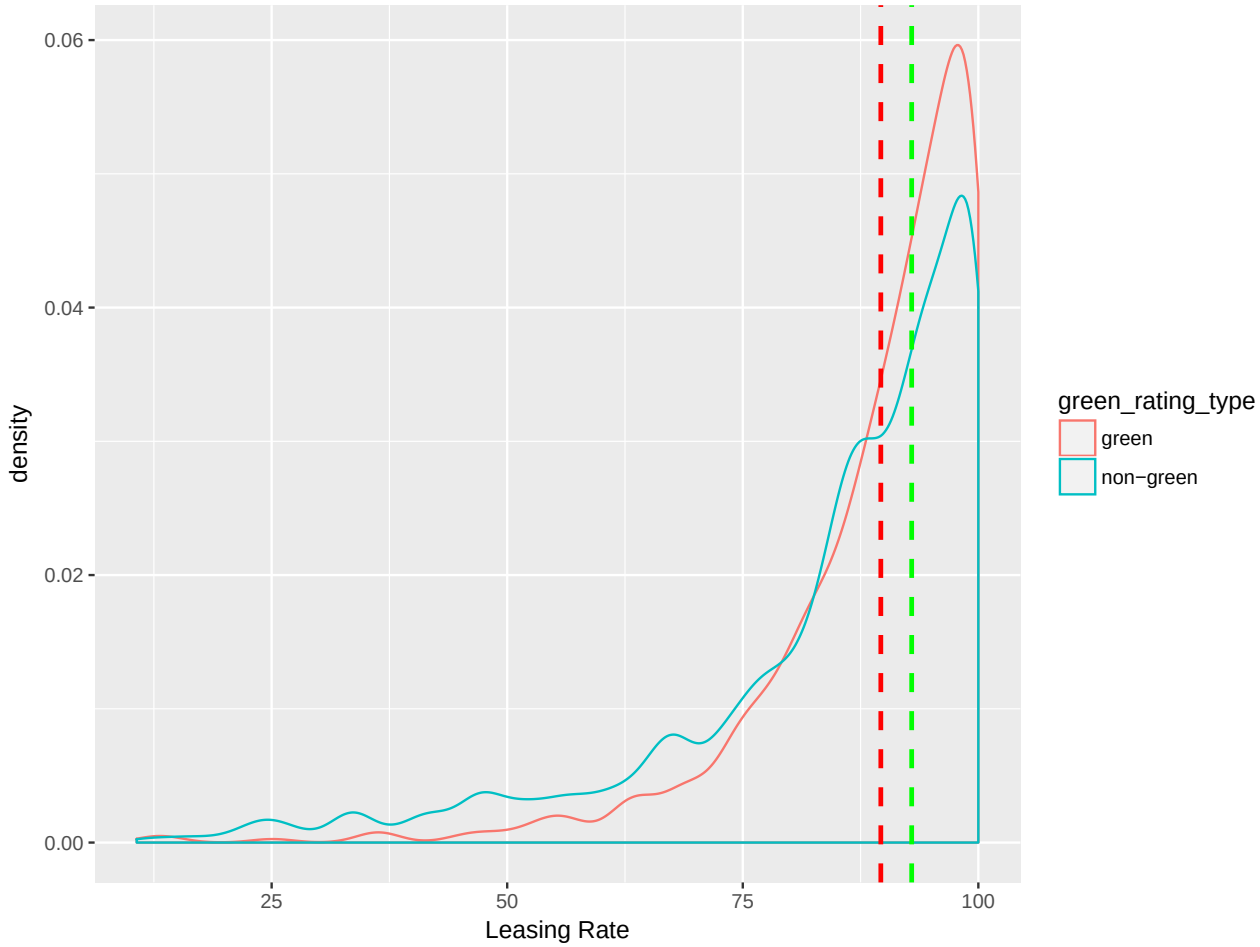
```
counts = c(green_renovated_perc, ngreen_renovated_perc)
barplot(counts, main = "Percentage of Buildings Renovated", horiz = T, names.arg = c("Green",
    "Non-Green"), xlim = c(0, 50), col = "brown")
```



Looks like non-green building renovate more than green-buildings which would add to the costs. But it's crucial to note the fact that age is not controlled in this analysis. Green buildings are relatively newer than non-green buildings.

Leasing Rate

```
ggplot(data, aes(x = leasing_rate, color = green_rating_type)) + geom_density() +
    geom_vline(aes(xintercept = 92.925), color = "green", linetype = "dashed",
        lwd = 1) + geom_vline(aes(xintercept = 89.65), color = "red", linetype = "dashed",
    lwd = 1) + labs(x = "Leasing Rate")
```



The median leasing rate of green buildings is higher (93) than non-green ones (89) which would translate to more revenue per square feet inspite of the high median rent.

The Residential Market

```
d = data.frame(unclass(table(data$cluster, data$green_rating_type)))
d$id <- row.names(d)

cluster = d[d$green == 0, ]$id

rent1 = data[data$cluster == c("38", "141", "209", "237", "279", "1146", "1147"),
    ]$cluster_rent
median(rent1)
```

```
[1] 19.39
```

```
rent2 = data[data$cluster != c("38", "141", "209", "237", "279", "1146", "1147"),
    ]$cluster_rent
median(rent2)
```

```
[1] 25.2
```

Looking at clusters which do not have any green building in the neighborhood and comparing it to clusters which do have atleast one green building, we see that the median rent for the former is lesser than the latter. We can see this as an oppurtunity of residential economic growth.

Gas Costs

```
round(median(ng$Gas_Costs), 4)
```

```
[1] 0.0103
```

```
round(median(gr$Gas_Costs), 4)
```

```
[1] 0.0103
```

Gas costs are similiar in both the buildings.

Electricity Costs

```
round(median(ng$Electricity_Costs), 4)
```

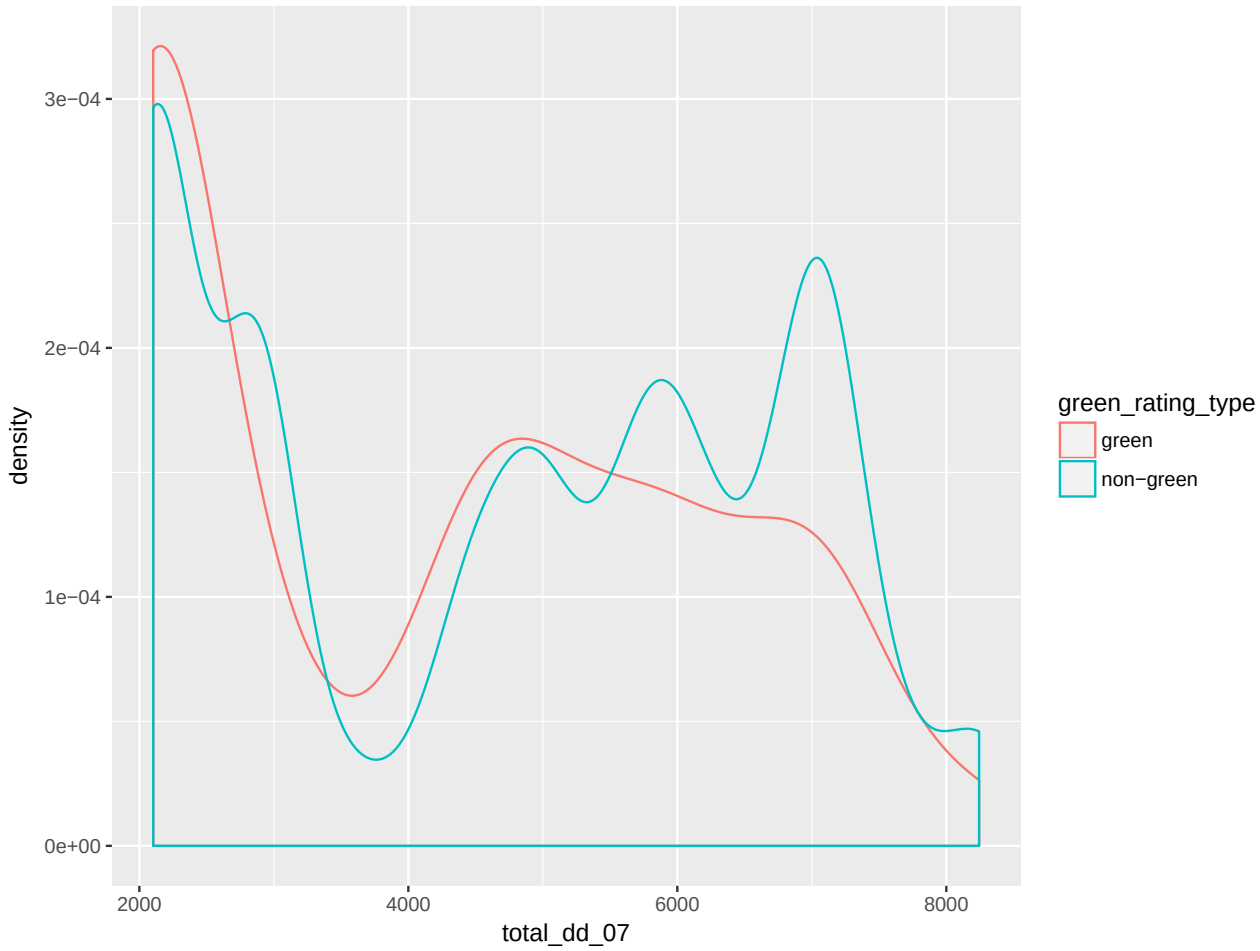
```
[1] 0.0327
```

```
round(median(gr$Electricity_Costs), 4)
```

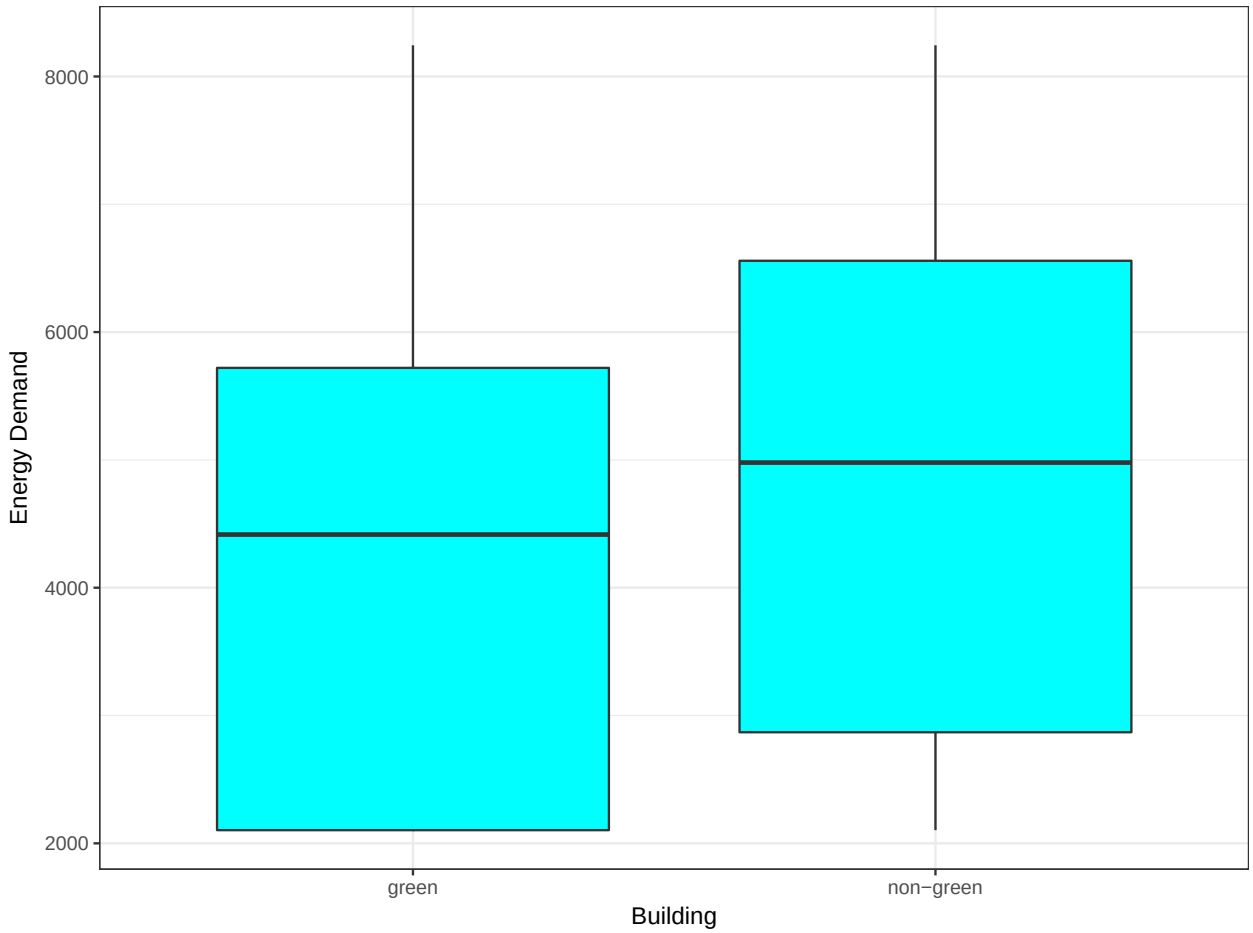
```
[1] 0.0341
```

Energy Demand

```
ggplot(data, aes(x = total_dd_07, color = green_rating_type)) + geom_density()
```



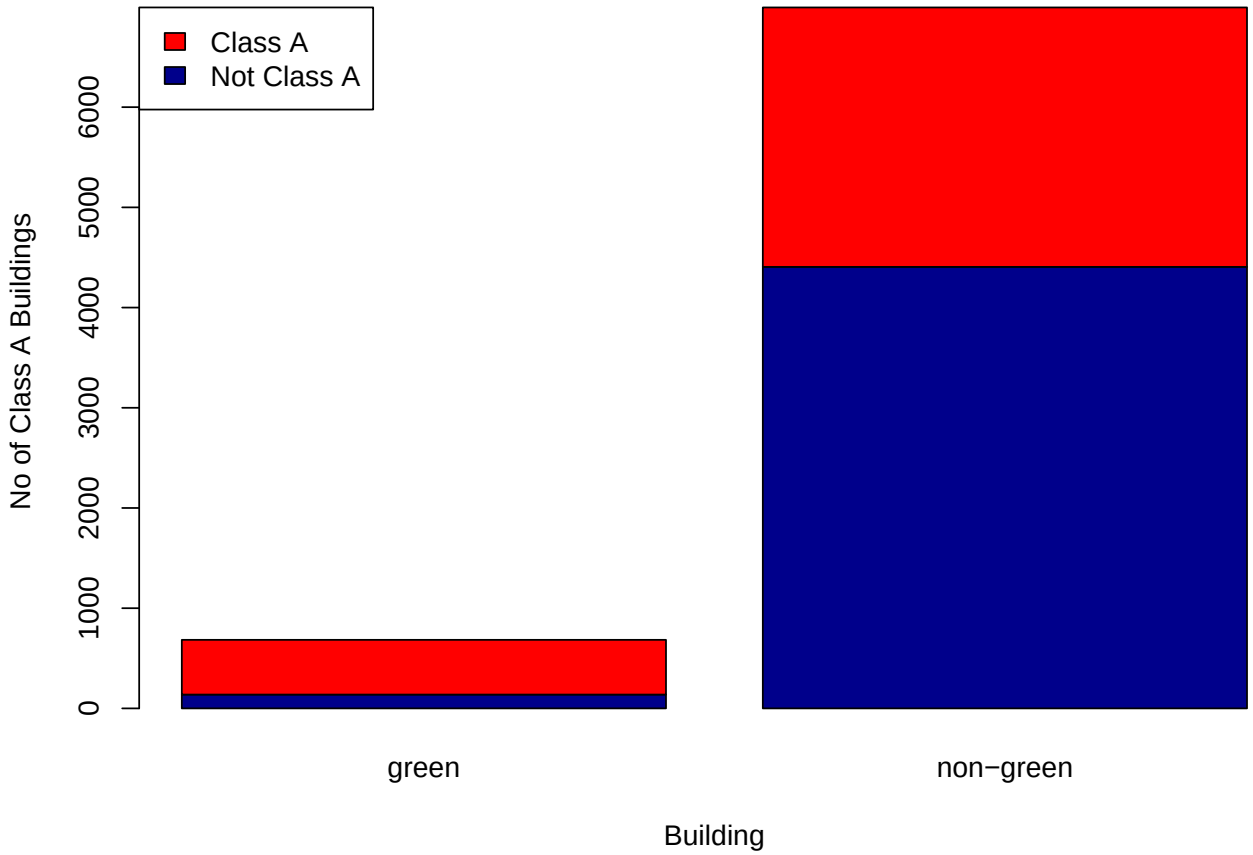

```
ggplot(data, aes(x = data$green_rating_type, y = data$total_dd_07)) + geom_boxplot(fill = "cyan") +
    labs(x = "Building", y = "Energy Demand") + theme_bw()
```



Electricity costs are slightly higher in the green buildings even though the energy demand is more in the non-green buildings. These costs are calculated for a building's geographic area and not building specific. Maybe the vicinity has a higher demand for energy which would relate to more costs.

Quality of Buildings

```
counts = table(data$class_a, data$green_rating_type)
barplot(counts, xlab = "Building", ylab = "No of Class A Buildings", col = c("darkblue",
    "red"), legend = c("Not Class A", "Class A"), args.legend = list(x = "topleft"))
```

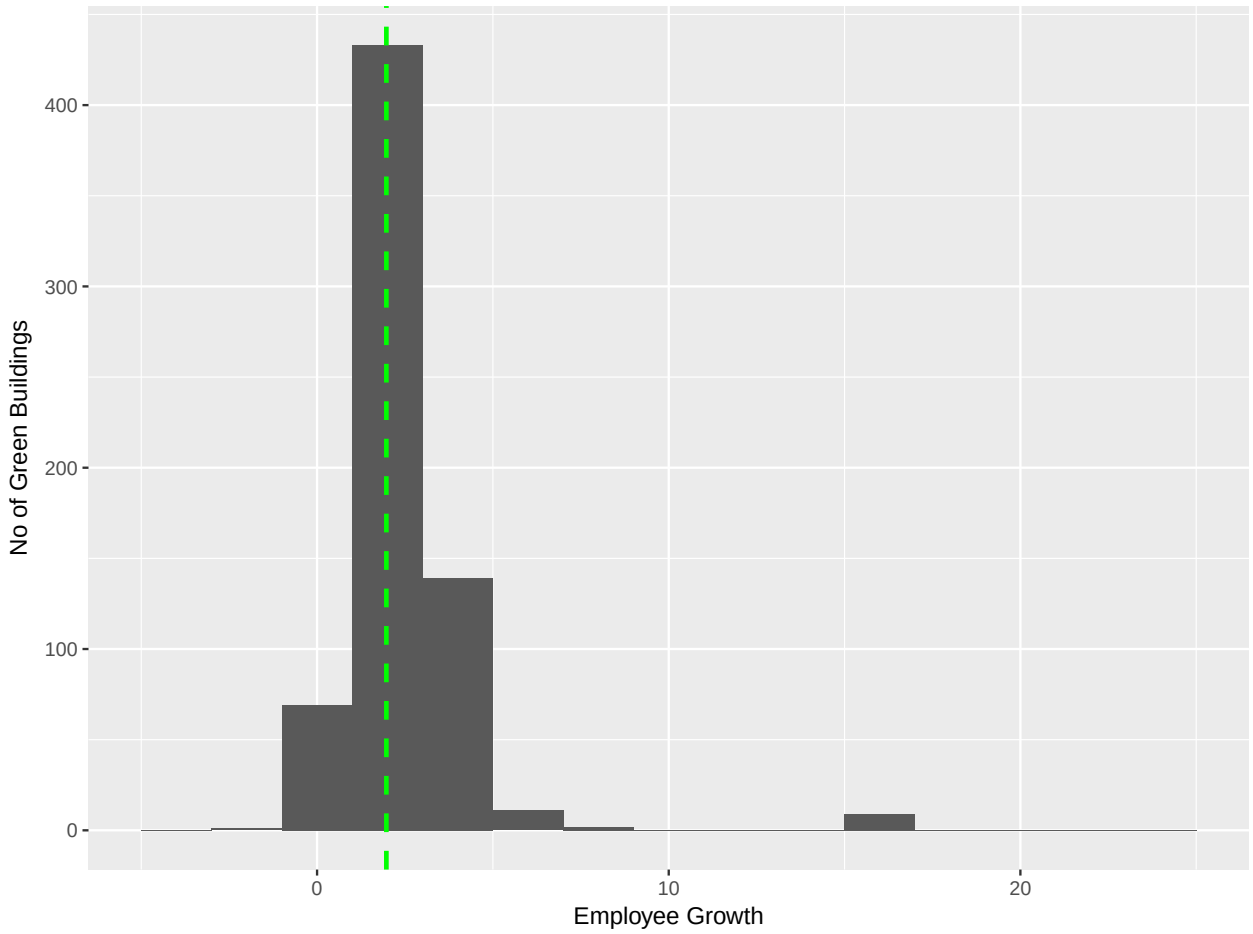


Green buildings have a much higher percentage of class_A buildings than non-green ones. This means that green buildings have a better reputation.

Employee Growth

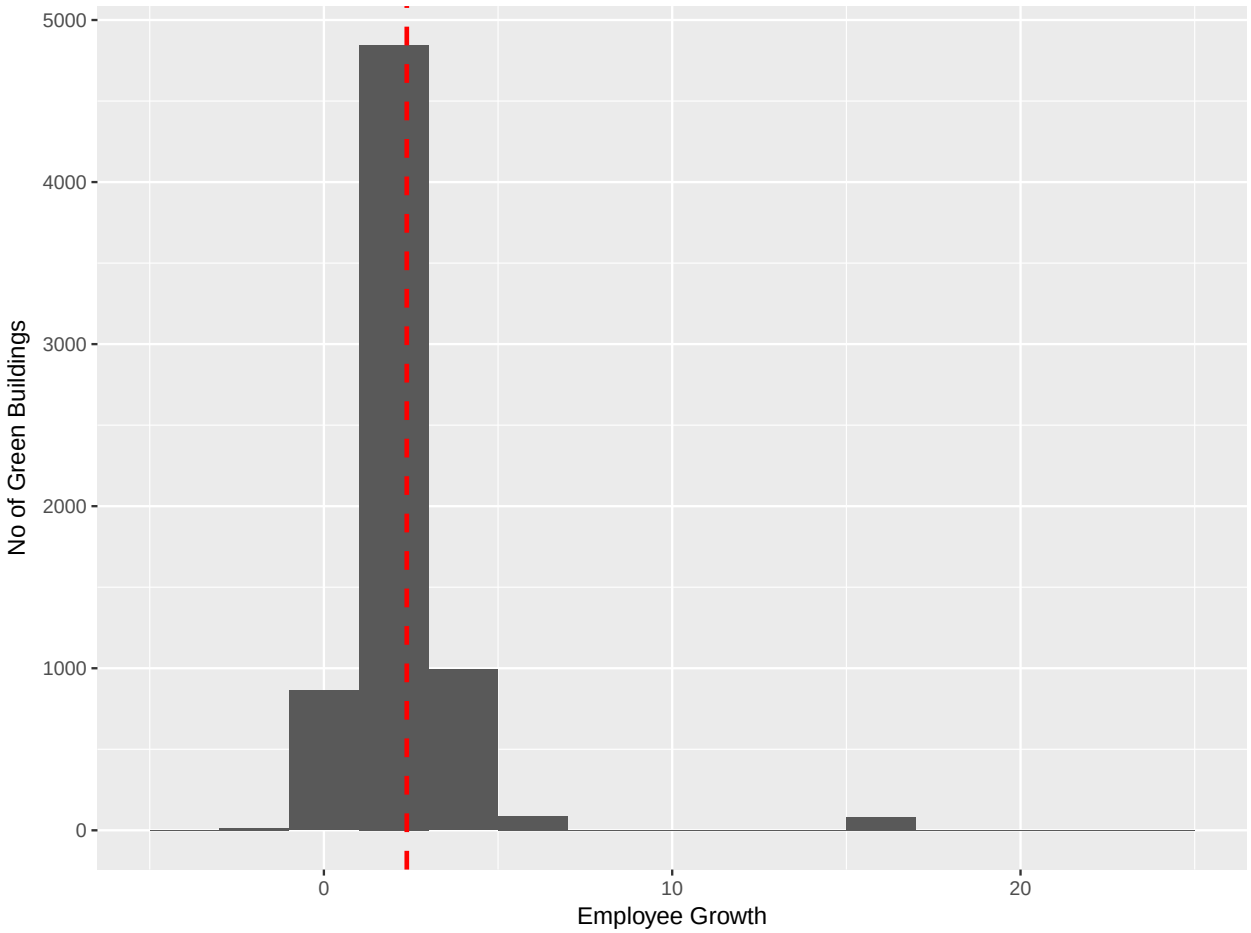
Green Buildings

```
ggplot(gr, aes(x = empl_gr)) + geom_histogram(binwidth = 2) + labs(x = "Employee Growth",
    y = "No of Green Buildings") + geom_vline(xintercept = 1.97, color = "green",
    linetype = "dashed", lwd = 1) + scale_x_continuous(limits = c(-5, 25))
```



Non - Green Buildings

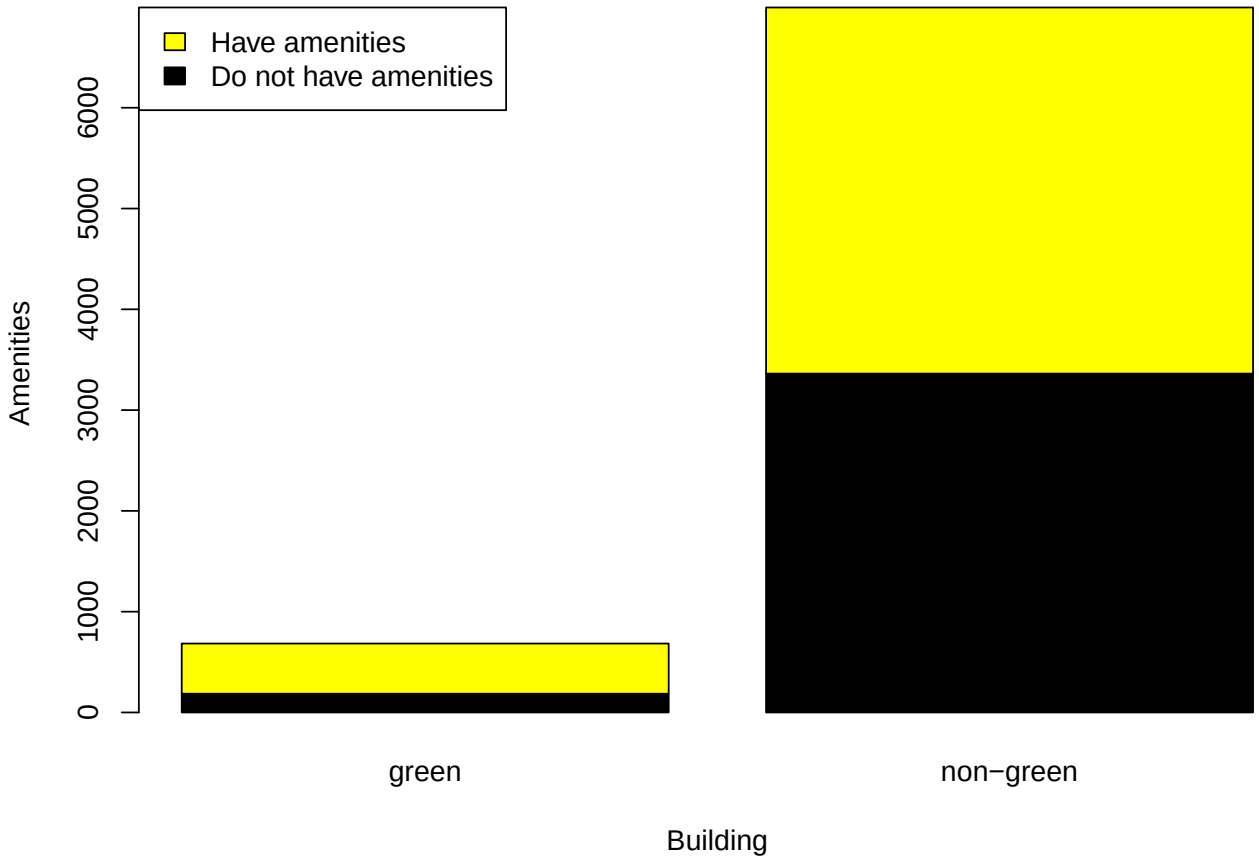
```
ggplot(ng, aes(x = empl_gr)) + geom_histogram(binwidth = 2) + labs(x = "Employee Growth",
    y = "No of Green Buildings") + geom_vline(xintercept = 2.38, color = "red",
    linetype = "dashed", lwd = 1) + scale_x_continuous(limits = c(-5, 25))
```



The median employee growth in the non-green building is 1.97 and the in the green building is 2.38 which substantiates the fact that better indoor environments result in higher employee productivity.

Amenities

```
counts1 = table(data$amenities, data$green_rating_type)
barplot(counts1, xlab = "Building", ylab = "Amenities", col = c("black", "yellow"),
    legend = c("Do not have amenities", "Have amenities"), args.legend = list(x = "topleft"))
```



A vast majority of the green buildings have amenities, and hence might charge a premium from potential tenants.

Let's now decide if investing in a green building is a good financial move or not

```
fin = read.csv("/Users/Jake/Downloads/npv-example.csv", stringsAsFactors = F)
fin$ROI = as.integer(fin$ROI)
fin$Building = as.factor(fin$Building)
str(fin)
```

```
'data.frame':   60 obs. of  6 variables:
 $ Index    : int  0 1 2 3 4 5 6 7 8 9 ...
 $ Year     : chr  "year 1" "year 2" "year 3" "year 4" ...
 $ Cash.Flow: num  6250000 6375000 6502500 6632550 6765201 ...
 $ NPV      : num  6250000 6071429 5897959 5729446 5565748 ...
 $ ROI      : int  -93750000 -87678571 -81780612 -76051166 -70485418 -65078692 -59826443 -54724259 -497
 $ Building : Factor w/ 2 levels "green","non-green": 2 2 2 2 2 2 2 2 2 2 ...
```

Assuming a steady increase of 2% in rent every year, we have calcuated the cash flow for 30 years.

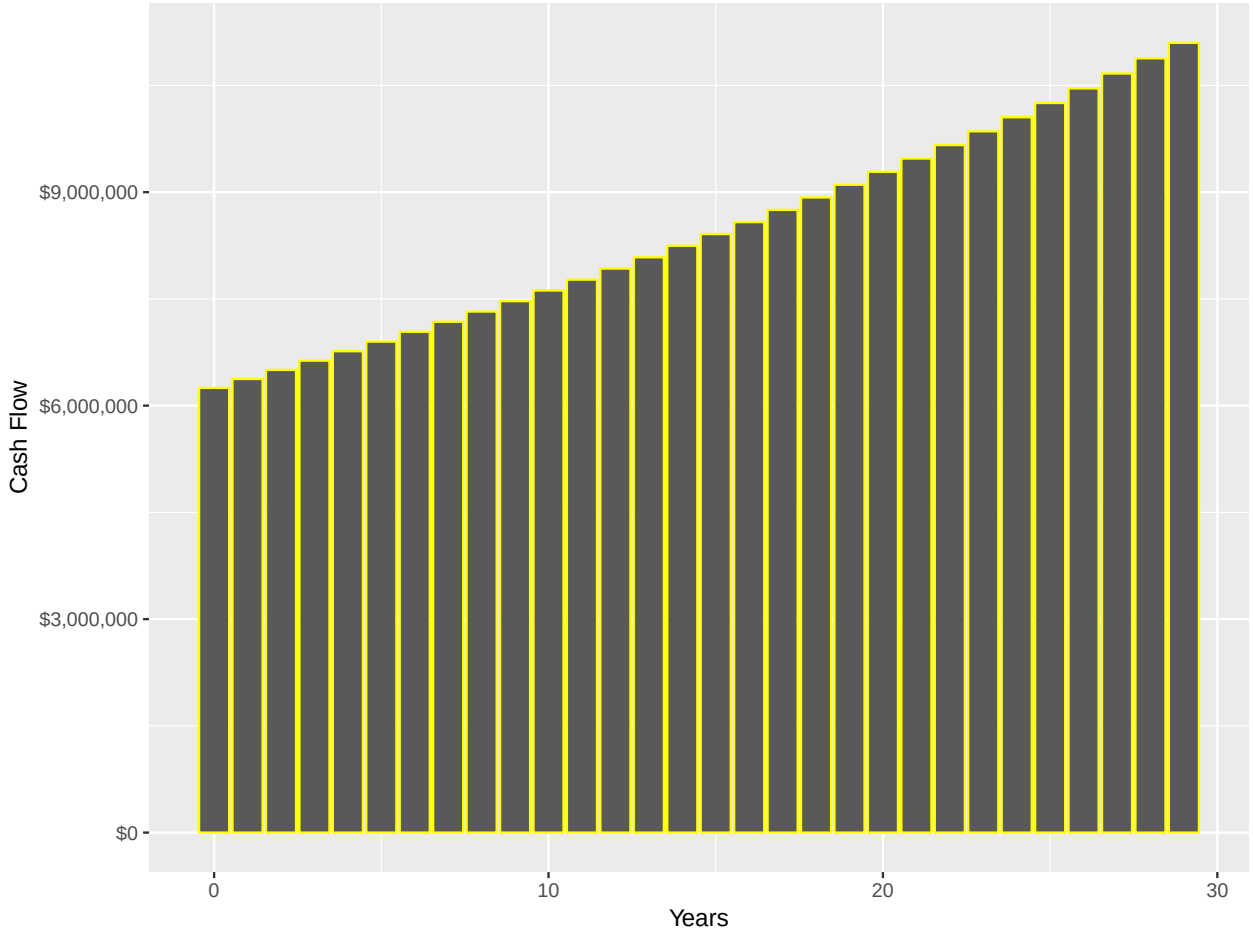
The discount rate is held steady at 5% for both the cases.

```
f1 = filter(fin, Building == "non-green")
f2 = filter(fin, Building == "green")
```

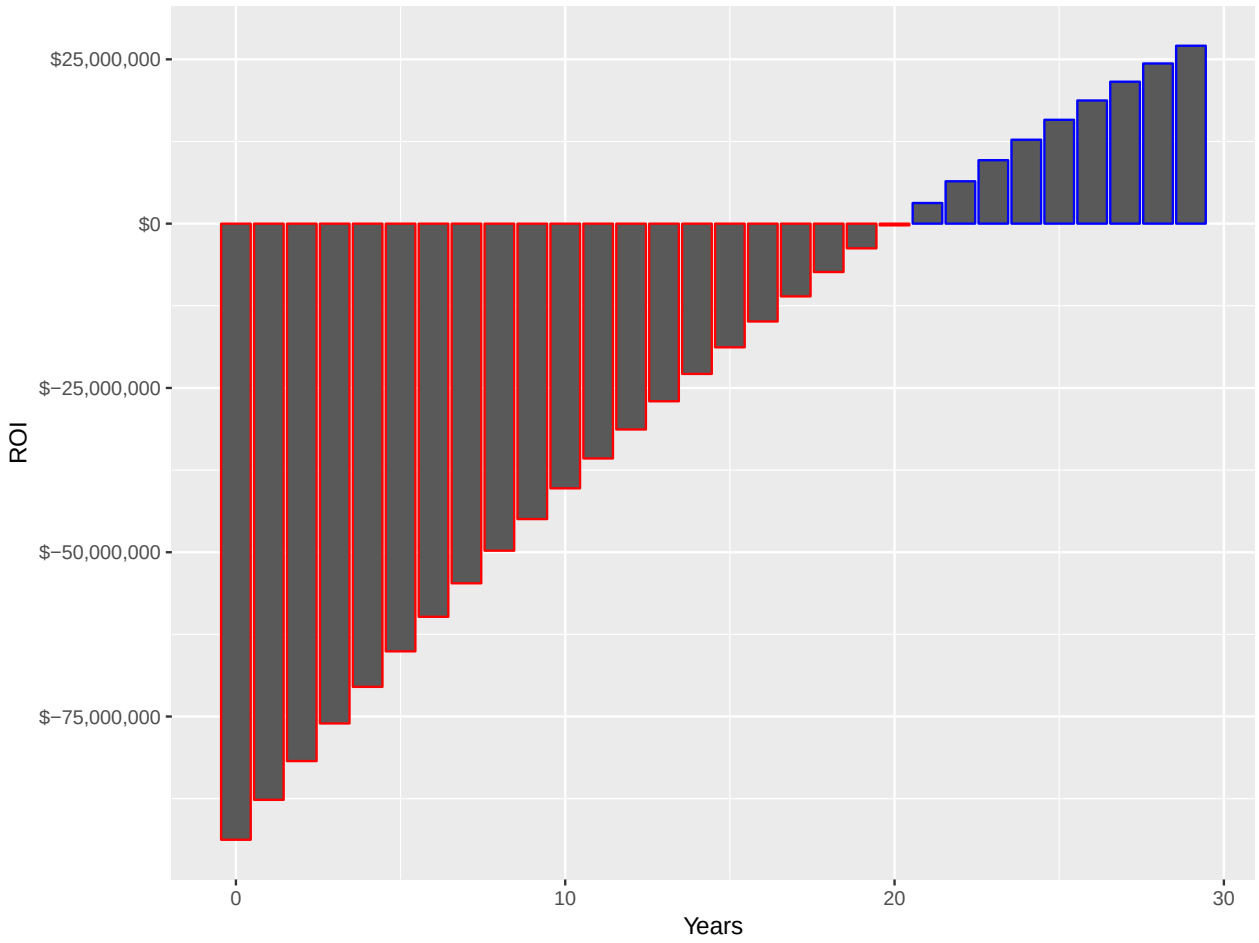
What is the Net Present Value (NPV) of the investment Y-O-Y over 30 years?

Non - Green Buildings

```
ggplot(f1, aes(Index, Cash.Flow)) + geom_col(color = "yellow") + scale_y_continuous(labels = dollar_for
    labs(x = "Years", y = "Cash Flow")
```



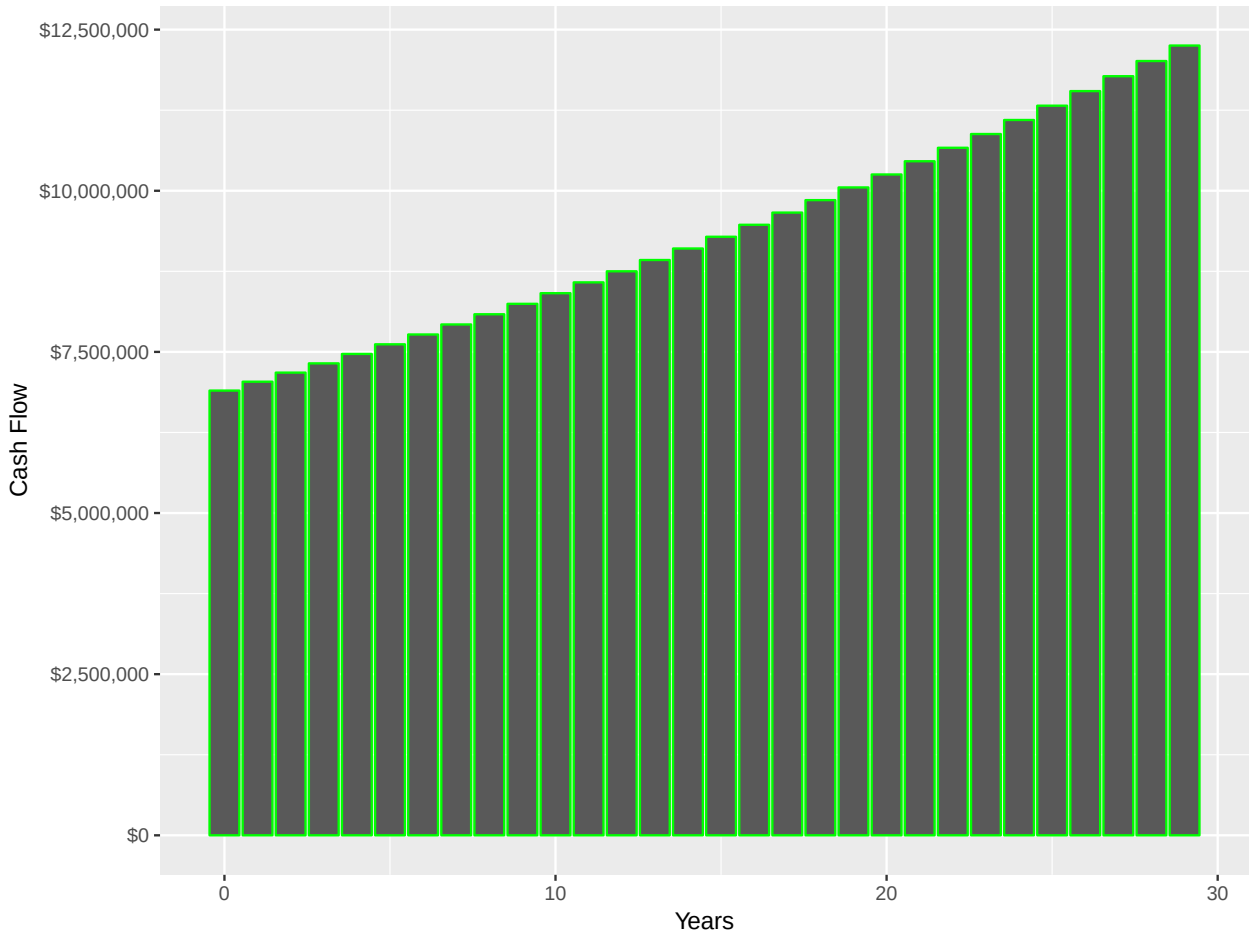

```
x = f1$ROI
cols <- c("red", "blue")[(x > 0) + 1]
ggplot(f1, aes(Index, ROI)) + geom_col(color = cols) + scale_y_continuous(labels = dollar_format()) +
    labs(x = "Years", y = "ROI")
```



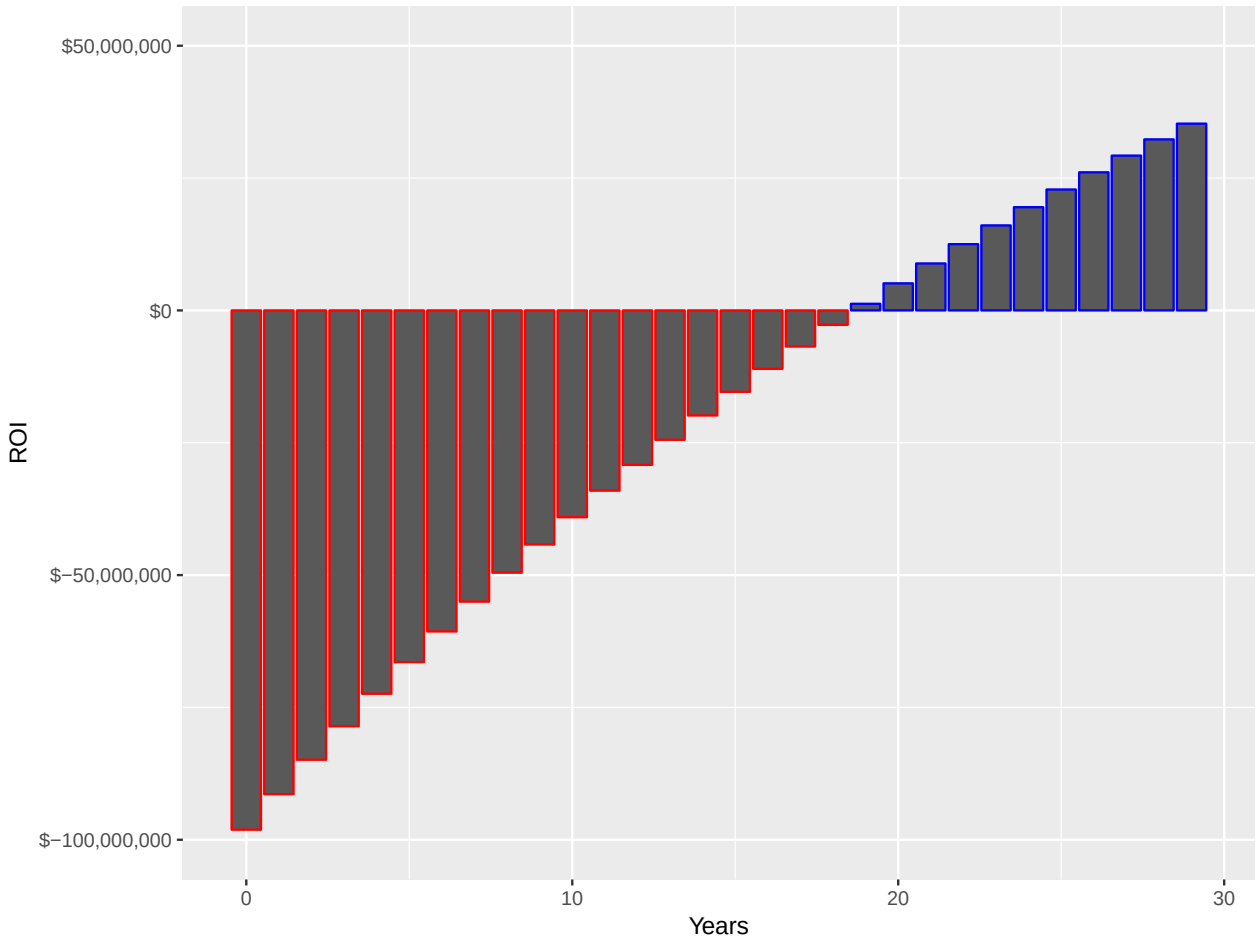
For a period of 30 years, the ROI of the investment of 100M for a non-green building with a median rent of 25$ per sq footage at 2% rent hike every year and 5% discount rate is 27,070,076 (approx 27M dollars). Breakeven will happen at year 22.

Green Buildings

```
ggplot(f2, aes(Index, Cash.Flow)) + geom_col(color = "green") + scale_y_continuous(labels = dollar_forma
    labs(x = "Years", y = "Cash Flow")
```



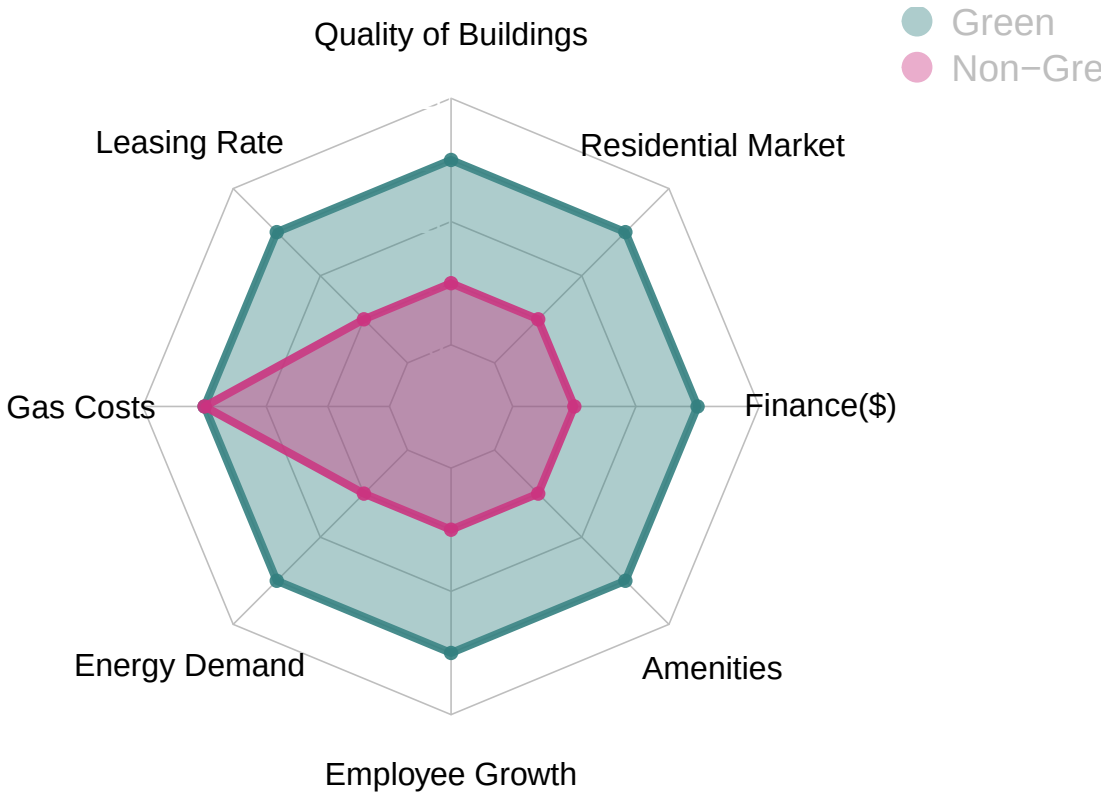

```
y = f2$ROI
cols <- c("red", "blue")[(y > 0) + 1]
ggplot(f2, aes(Index, ROI)) + geom_col(color = cols) + scale_y_continuous(labels = dollar_format(),
    limits = c(-1e+08, 5e+07)) + labs(x = "Years", y = "ROI")
```



For a period of 30 years, the ROI of the investment of 105M for a green building with a median rent of 27.6$ per sq footage at 2% rent hike every year and 5% discount rate is 35,285,364 (approx 35M dollars). Breakeven will happen at year 19.

Conclusion

```
library(fmsb)
data_plot = data.frame(matrix(c(15,5,15,5,15,15,15,5,15,5,15,5,15,5,15,5),ncol=8,nrow=2))
colnames(data_plot)=c("Quality of Buildings" ,"Leasing Rate" , "Gas Costs" , "Energy Demand","Employee
rownames(data_plot)=c("Green","Non-Green")
data_plot=rbind(rep(20,8) , rep(0,8) , data_plot)
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9) )
colors_in=c( rgb(0.2,0.5,0.5,0.4), rgb(0.8,0.2,0.5,0.4) , rgb(0.7,0.5,0.1,0.4) )
radarchart( data_plot  , axistype=1 ,
    #custom polygon
    pcol=colors_border , pfcol=colors_in , plwd=4 , plty=1,
    #custom the grid
    cglcol="grey", cglty=1, axislabcol="white", caxislabels=seq(0,20,5),cglwd=0.8,
    #custom labels
    vlcex=1
    )
legend(x=1.4, y=1.4, legend = rownames(data_plot[-c(1,2),]), bty = "n", pch=20 , col=colors_in , text.c
```



Based on all the factors that we see above, building a green building is a better investment

Boostrapping

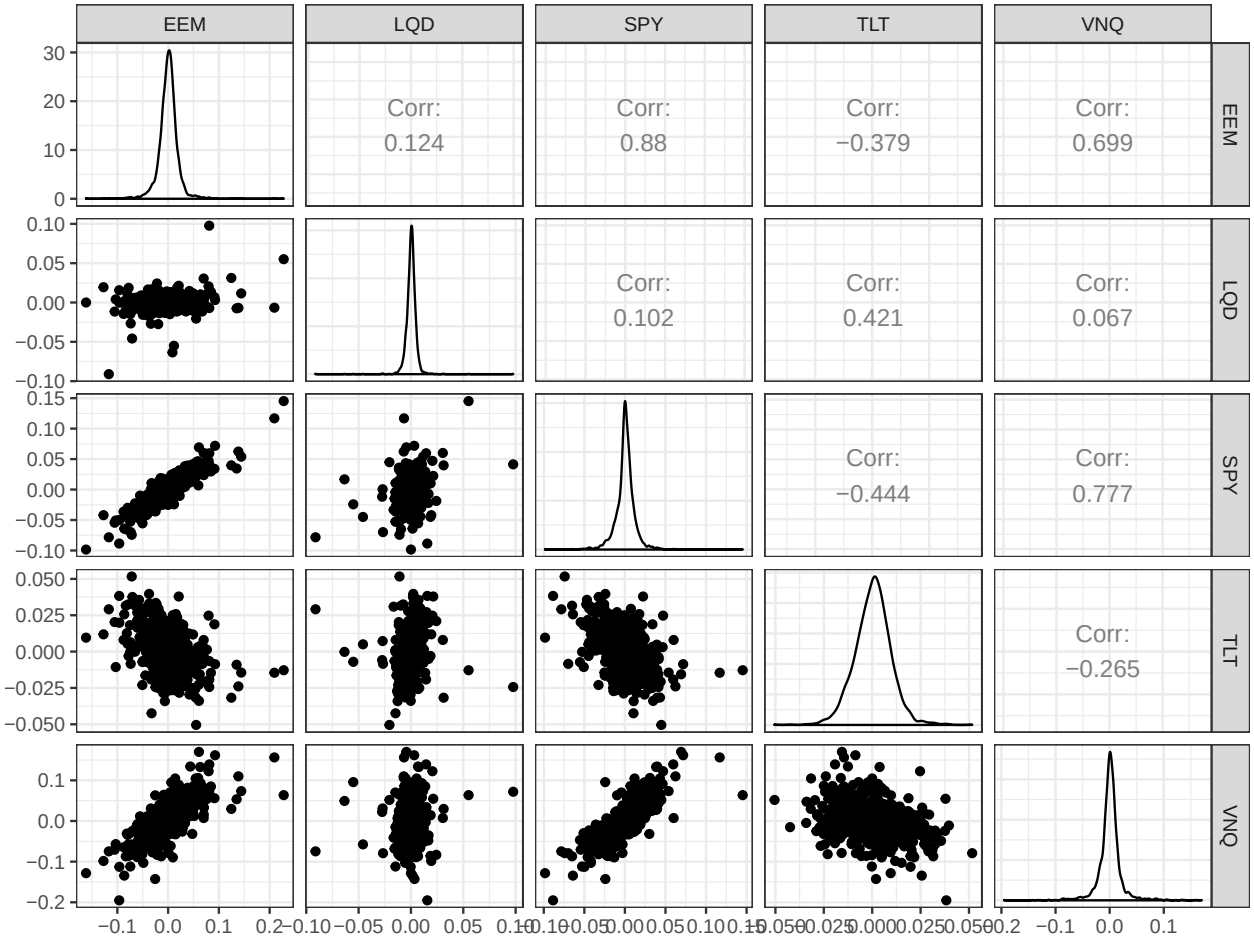
```
getSymbols(c("SPY", "TLT", "LQD", "EEM", "VNQ"))
```

```
'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"
```

```
EEM <- ClCl(adjustOHLC(EEM, use.Adjusted = T))
LQD <- ClCl(adjustOHLC(LQD, use.Adjusted = T))
SPY <- ClCl(adjustOHLC(SPY, use.Adjusted = T))
```

```
TLT <- ClCl(adjustOHLC(TLT, use.Adjusted = T))
VNQ <- ClCl(adjustOHLC(VNQ, use.Adjusted = T))
etfs <- as_tibble(rownames_to_column(data.frame(cbind(EEM, LQD, SPY, TLT, VNQ)),
    "ts")) %>% mutate(ts = lubridate::ymd(ts))
etfs[is.na(etfs)] <- 0
colnames(etfs) <- c("ts", "EEM", "LQD", "SPY", "TLT", "VNQ")

etfs %>% select(-ts) %>% ggpairs() + theme_bw()
```




```
total_wealth = 1e+05
```

```
# Annual volatility (standard deviation of daily returns * square root of
# number of trading days per year)
etfs[, -1] %>% sapply(sd) * sqrt(252)
```

```
       EEM        LQD        SPY        TLT        VNQ
0.32659149 0.08574862 0.20300264 0.14918785 0.34892254
```

```
# Annual return (mean of daily returns * number of trading days per year)
etfs[, -1] %>% sapply(mean) * 252
```

```
       EEM        LQD        SPY        TLT        VNQ
0.08218641 0.05773369 0.09260839 0.07766903 0.10997011
```

We see, as expected, that the debt funds are less volatile and therefore have lower annual returns. From 2007,

a balanced portfolio of these assets would behave as follows.

```
get_rtns <- function(weights, assets, initial_wealth) {
    holdings = weights * initial_wealth
    n_days = nrow(assets)
    wealthtracker = rep(0, n_days)
    for (today in 1:n_days) {
        return.today = assets[today, -1]
        holdings = holdings + holdings * return.today
        total_wealth = sum(holdings)
        wealthtracker[today] = total_wealth
        holdings = total_wealth * weights  #daily rebalance
    }
    data.frame(ts = assets[, 1], rtns = wealthtracker)
}

weights_uni <- rep(0.2, 5)
rtns <- get_rtns(weights_uni, etfs, 1e+05)
rtns %>% ggplot(aes(ts, rtns)) + geom_line() + theme_bw()
```


```
Delt(rtns[1, "rtns"], rtns[nrow(rtns), "rtns"])
```

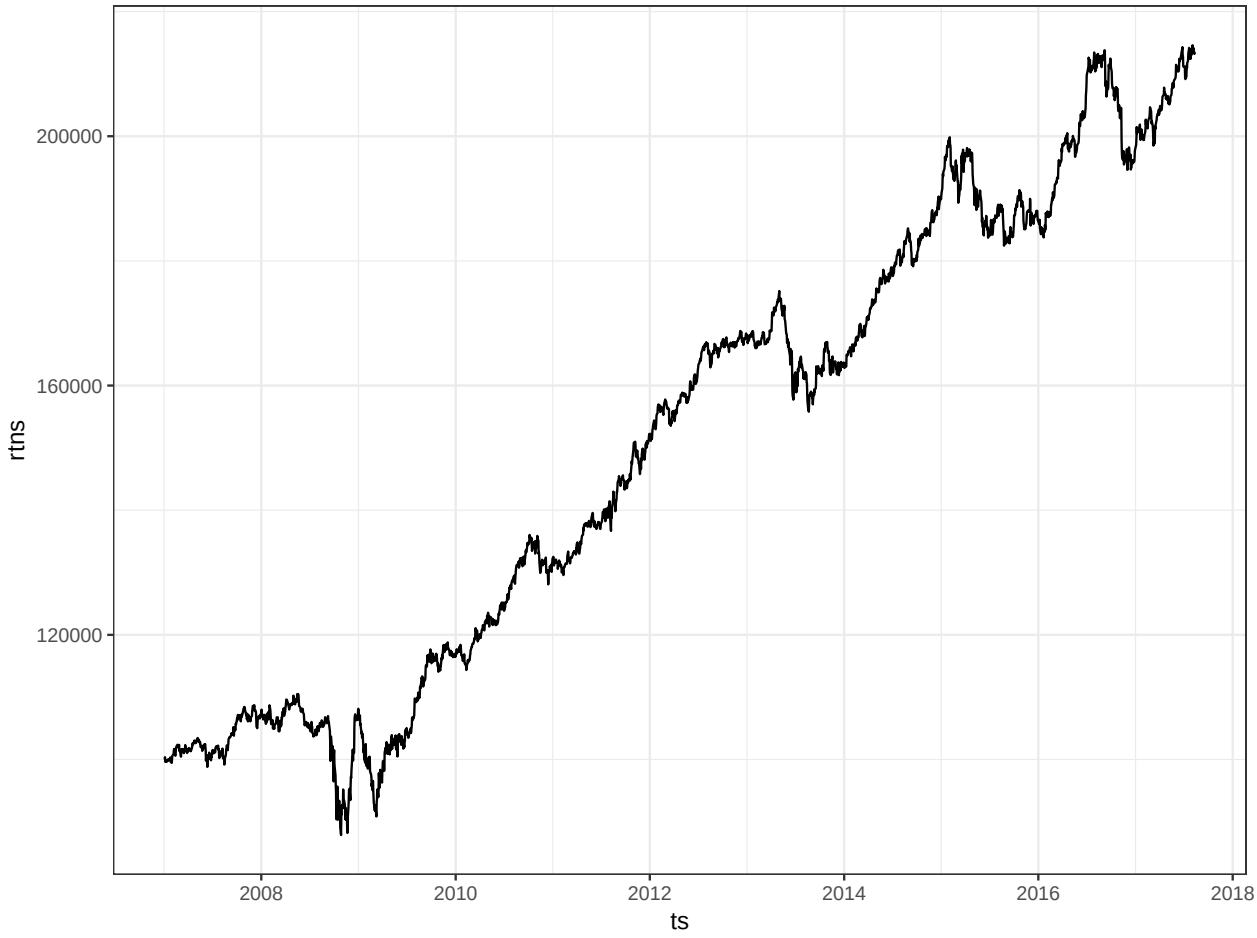
```
     Delt.0.arithmetic
[1,]          1.138207
```

```
sd(na.omit(Delt(rtns[, "rtns"]))) * sqrt(252)
```

```
[1] 0.15741
```

We see an evenly-weighted portfolio would have returned 114% over the backtest period, with an annual volatility of 15.7%. Let's construct another portfolio weighted towards the lower-volatility bond funds.

```
weights_safe <- c(0.1, 0.35, 0.1, 0.35, 0.1)
rtns <- get_rtns(weights_safe, etfs, 1e+05)
rtns %>% ggplot(aes(ts, rtns)) + geom_line() + theme_bw()
```




```
Delt(rtns[1, "rtns"], rtns[nrow(rtns), "rtns"])
```

```
     Delt.0.arithmetic
[1,]          1.133335
```

```
sd(na.omit(Delt(rtns[, "rtns"]))) * sqrt(252)
```

```
[1] 0.09389372
```

So we only got a 113% return but had an annual volatility of only 9%. Wow! Thats as much as what we observed with the uniform portfoio

```
weights_agg <- c(0.3, 0.05, 0.3, 0.05, 0.3)
rtns <- get_rtns(weights_agg, etfs, 1e+05)
rtns %>% ggplot(aes(ts, rtns)) + geom_line() + theme_bw()
```


```
Delt(rtns[1, "rtns"], rtns[nrow(rtns), "rtns"])
```

```
     Delt.0.arithmetic
[1,]         0.9559096
```

```
sd(na.omit(Delt(rtns[, "rtns"]))) * sqrt(252)
```

```
[1] 0.240805
```

With a more aggressive risk profile, we make 96% but at an annual 24% volatility.

Again, we note the different volatilities in each asset: the REIT and equity funds are more volatile, and portfolios with higher proportions of them are thus more volatile. The inverse is also true. What's strange is that, over the selected time period and particular assets, the low-volatility assets outperforms the high-volatility assets. This makes sense because in the time period of 10 years there is a huge crash that makes the volatile high-risk stocks a bigger loss. The crash in 2008, affected the riskiest portfolio a lot more than the safe portfolios. Hence, we see a lower return in a 10 year time frame observed above.

```
set.seed(42)
VaR <- function(initial_wealth, weights) {
    sim1 = foreach(i = 1:500, .combine = "rbind") %do% {
        total_wealth = initial_wealth
        holdings = weights * total_wealth
        n_days = 20
        wealthtracker = rep(0, n_days)
        for (today in 1:n_days) {
            return.today = resample(etfs[, -1], 1, orig.ids = FALSE)
```

```
            holdings = holdings + holdings * return.today
            total_wealth = sum(holdings)
            wealthtracker[today] = total_wealth
        }
        wealthtracker
    }
    quantile(sim1[, n_days], 0.05) - initial_wealth
}
```

The 20-day 5% VaR for the different portfolios:

```
VaR(1e+05, weights_uni)
```

```
       5%
-6169.602
```

```
VaR(1e+05, weights_safe)
```

```
       5%
-3985.984
```

```
VaR(1e+05, weights_agg)
```

```
       5%
-10285.38
```

Conclusion

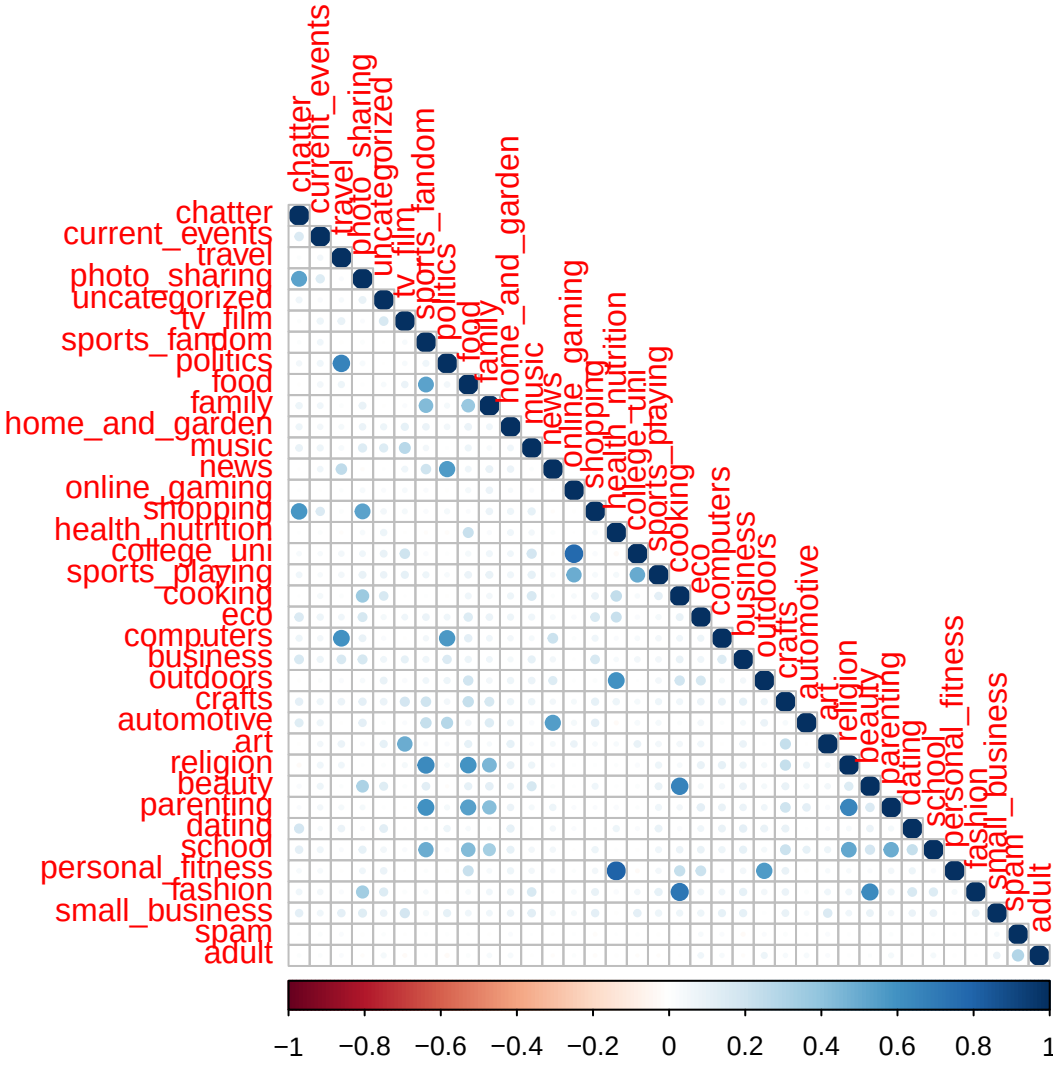
We see that, as expected, the lower-volatility portfolio has the lowest 20-day 5% VaR, meaning we would expect it to lose the least money of the portfolios during the worst 5% of possible 20-day periods.

This is a good example, of different scenarios of portfolios performing under a crash, where we observe a balanced or a safe portfolio making better returns than a more volatile portfolio which is vulnerable to a crisis in the market.

Market Segmentation

First, lets do some exploratory data analysis by looking at correlations of the different types

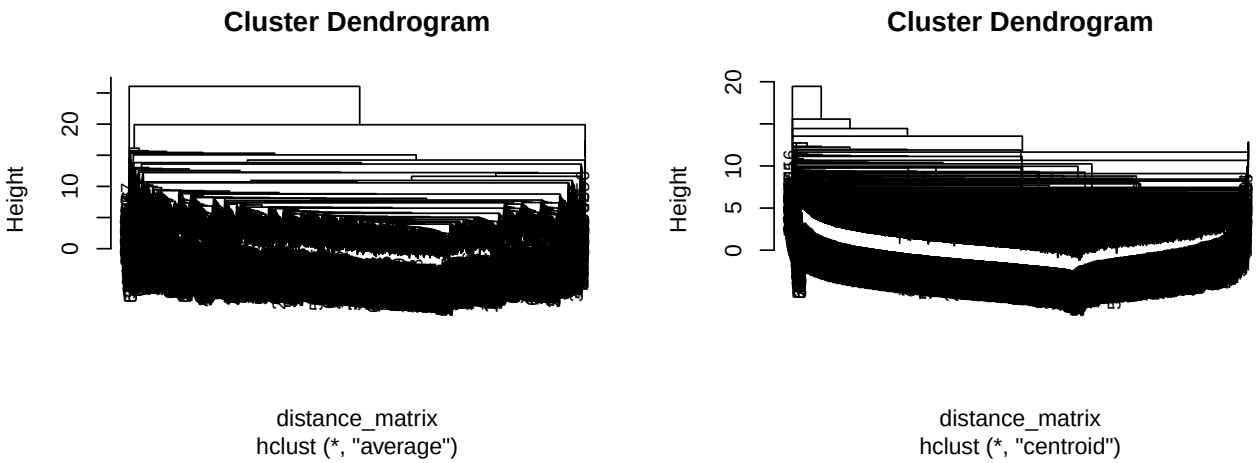
```
soc_mkt = soc_mkt_all[, -1]
corr_mat = cor(soc_mkt)
corrplot(corr_mat, type = "lower")
```

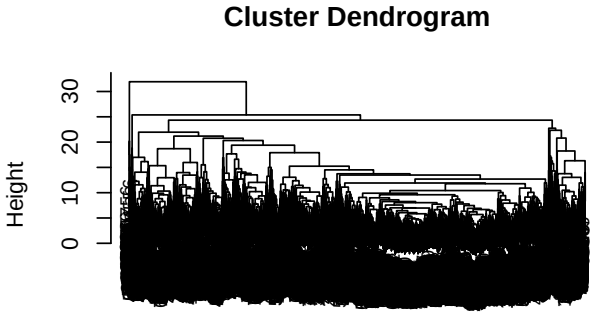


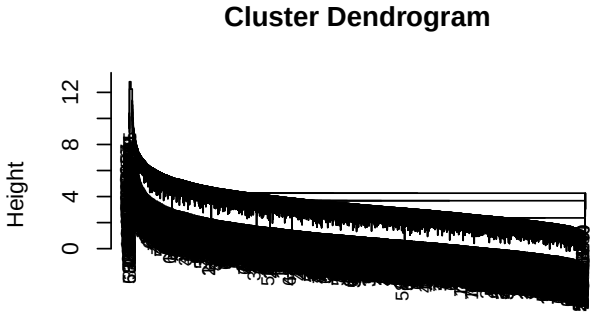
Looking at the correlation plot, dots that are darker indicate a strong relationship. Results here, seem to be intuitive and in the right direction. The highest correlation is between Health-Nutrition and Personal Fitness, as they are related topics, tweeting about these topic is highly possible. There also seems to be high correlation between Cooking and Fashion, and Cooking and Beauty, which intuitively make sense. We observe a relationship between Online Gaming and College/University often occur together, as a lot of college students are interested in gaming this also conforms with human intuition.

Now we can try our hand at clustering and confirm our hypothesis.

Heirarchical Clustering








```
   1    2    3    4    5
7867   10    2    1    2

   1    2    3    4    5
7876    2    1    2    1

   1    2    3    4    5    6    7    8    9   10
 487 5628  284  859  130   16  410   49    9   10

   1    2    3    4    5
7831   47    2    1    1
```

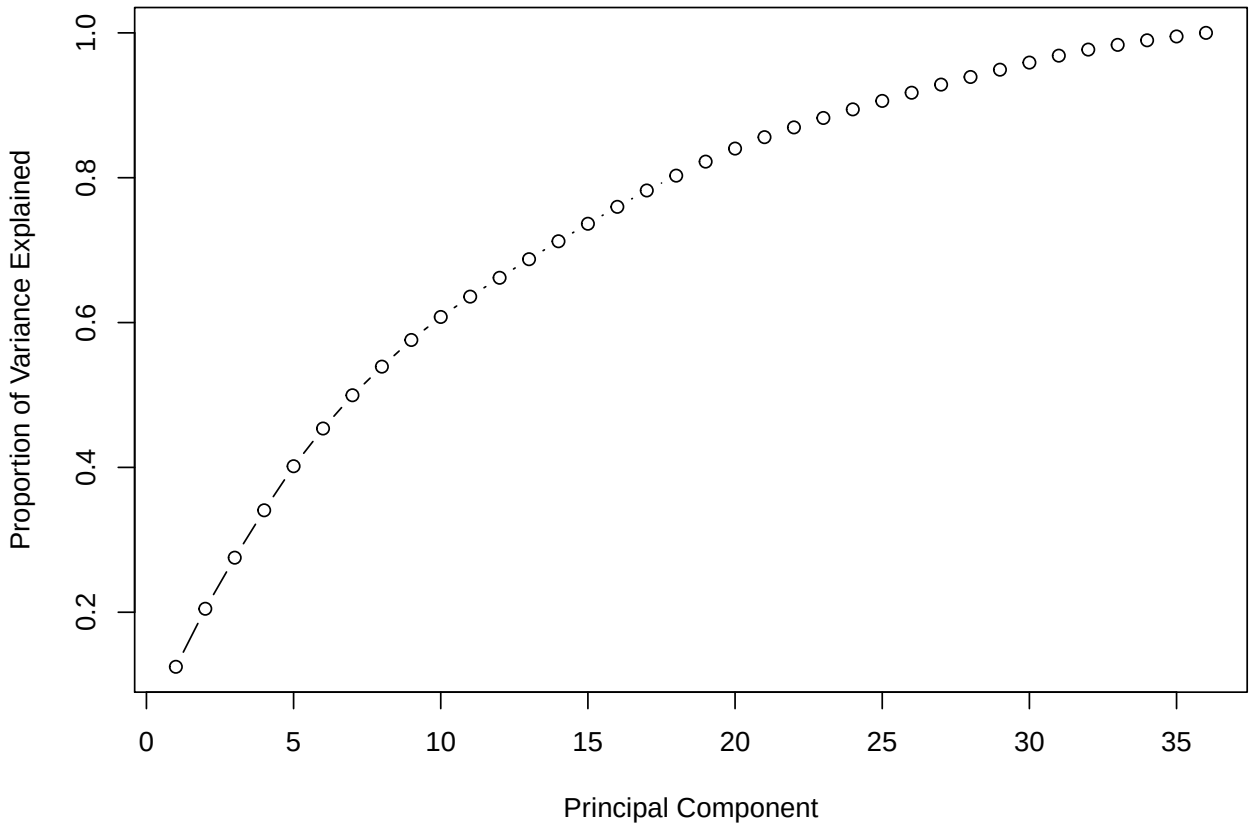
Hierarchical clustering results are hard to interpret and even after we cut the trees to 5. Hierarchical clustering for this data is not useful since there arent tiers or levels among the distribution and so it ends up grouping incorrectly into a single cluster only.

Next lets try Principle Component Analysis

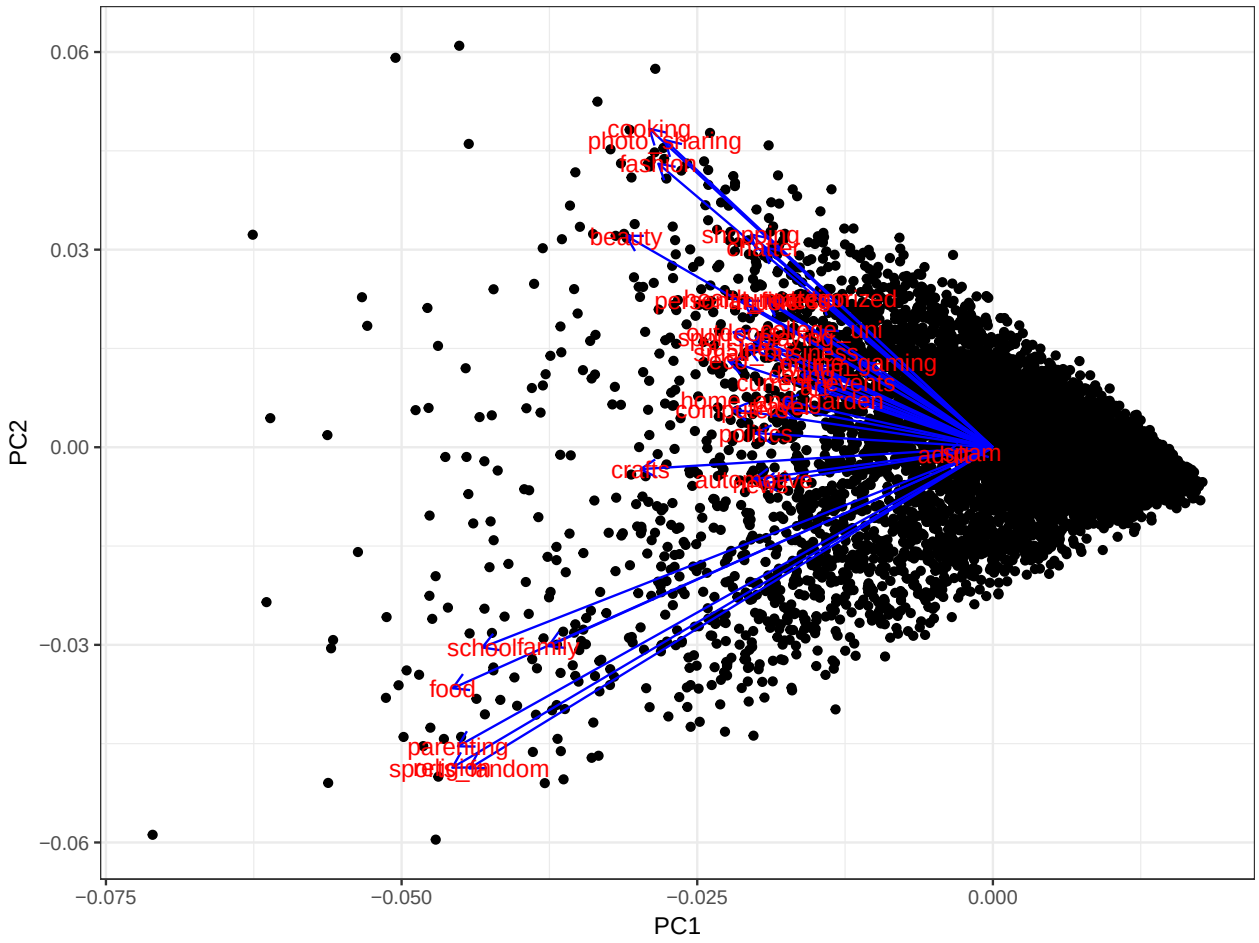
PCA

```
## PCA ##

pc1 = prcomp(as.matrix(soc_mkt_scaled), scale. = TRUE)
# compute standard deviation of each principal component
std_dev <- pc1$sdev
```

```r
# compute variance
pr_var <- std_dev^2
# proportion of variance explained
prop_varex <- pr_var/sum(pr_var)
par(mfrow = c(1, 1))
plot(cumsum(prop_varex), xlab = "Principal Component", ylab = "Proportion of Variance Explained",
    type = "b")
```




```r
loadings = pc1$rotation
scores = pc1$x
# autoplot(pc1,loadings=TRUE)
autoplot(pc1, loadings = TRUE, loadings.colour = "blue", loadings.label = TRUE,
    loadings.label.size = 4) + theme_bw()
```



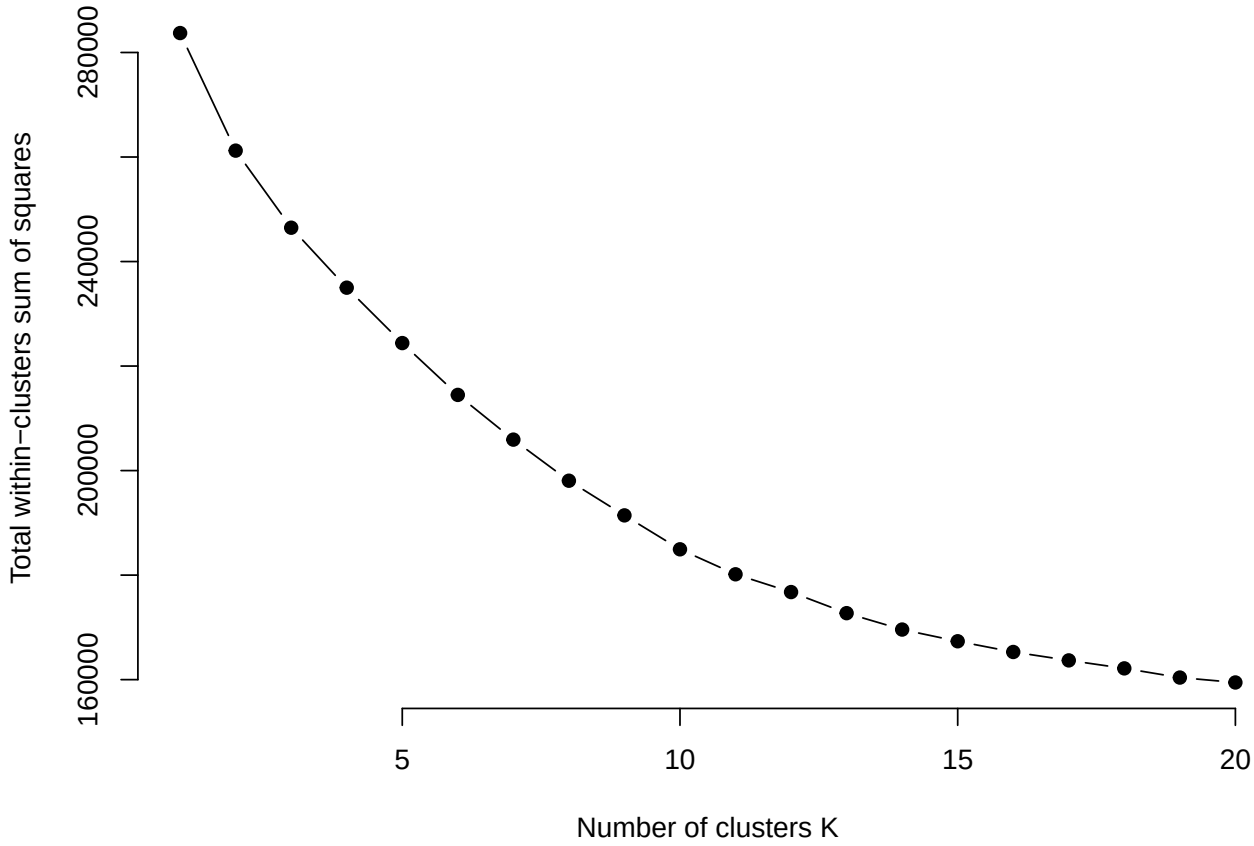
80% of the variance is explained by about 15 principle components. 50% of the variance is covered with just 6 PCs

Next, we will try clustering and try to visualize these clusters using PCs to validate them.

K-means

We used elbow method for K-means to get the vaue of k that understands cluster composition.

```
k.max <- 20  # Maximal number of clusters
wss <- sapply(1:k.max, function(k) {
    kmeans(soc_mkt_scaled, k, nstart = 10, iter.max = 50)$tot.withinss
})
plot(1:k.max, wss, type = "b", pch = 19, frame = FALSE, xlab = "Number of clusters K",
    ylab = "Total within-clusters sum of squares")
```

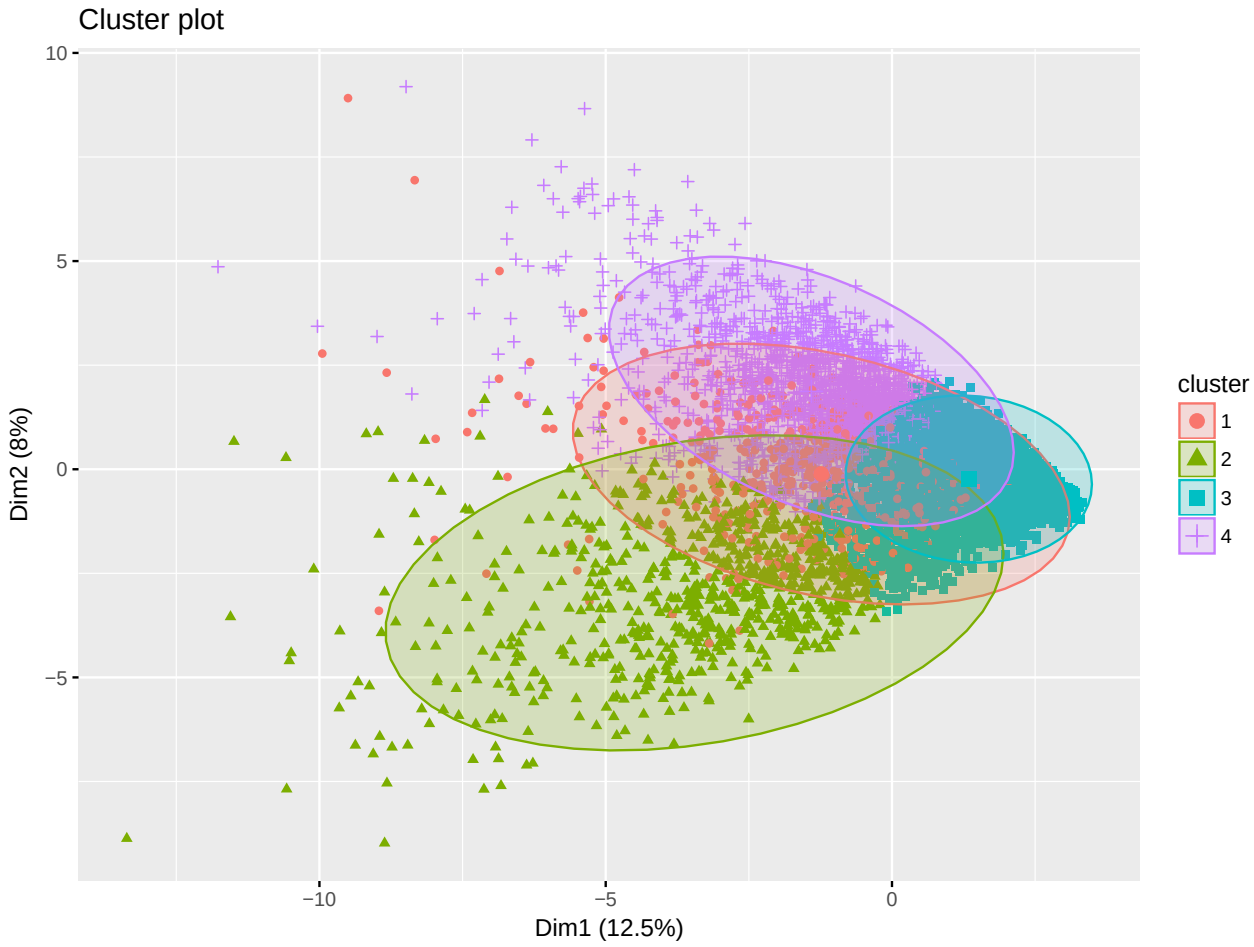


The elbow starts to tilt at about 8 clusters. That should be our answer.

Below we will look at 4 clusters first to see why lesser custers give us no good answer.

Trying with 4 clusters:

```
# K-means clustering with 4
set.seed(1)
km.res1 <- kmeans(soc_mkt_scaled, 4, nstart = 25)
# k-means group number of each observation

# Visualize k-means clusters
fviz_cluster(km.res1, data = soc_mkt_scaled, geom = "point", stand = FALSE,
    frame.type = "norm")
```




```
clusters_pars = km.res1$centers
transposed = t(clusters_pars)
cluster_1 = transposed[which(abs(transposed[, 1]) >= 0.4), 1]
cluster_2 = transposed[which(abs(transposed[, 2]) >= 0.4), 2]
cluster_3 = transposed[which(abs(transposed[, 3]) >= 0.4), 3]
cluster_4 = transposed[which(abs(transposed[, 4]) >= 0.4), 4]
```

```
    travel   politics       news  computers automotive
  1.768714   2.375881   1.941556   1.549122   1.122043

sports_fandom          food        family        crafts      religion
    2.0214691     1.8089459     1.4618754     0.6915247     2.2295333
    parenting        school
    2.1003788     1.6298873

numeric(0)

         chatter    photo_sharing    uncategorized            music
       0.5492028        0.7981651        0.4156293        0.4847336
        shopping health_nutrition   sports_playing          cooking
       0.5762315        0.6590026        0.4000114        0.8742163
             eco         outdoors           beauty personal_fitness
       0.4171848        0.5438048        0.6516625        0.6722260
         fashion
       0.7667315
```

```
fact_clust <- factor(km.res1$cluster)
summary(fact_clust)
```

```
   1    2    3    4
 714  768 4563 1837
```

With 4 clusters, one cluster is getting grouped generally with very weak relationships to the center. This cluster has 4563 observations and that is 4563 observations which are wasted into a general group. We suspect this could be the influence of spam and chatter. To see more detailed clusters we will next see 8 clusters.

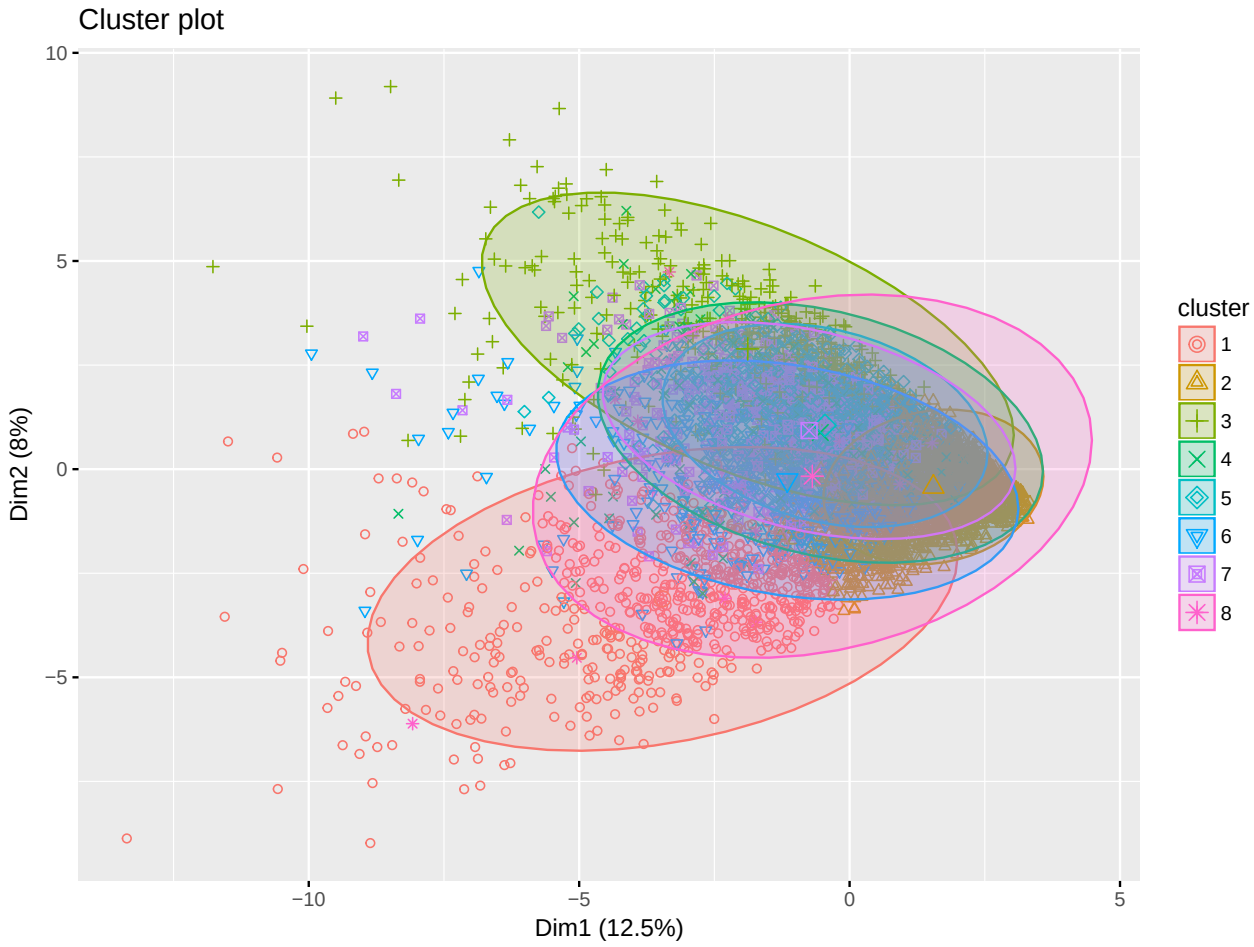
Trying with 8 clusters to see if we can fine tune without over splitting

Get the X values without the ids and scale them

```
soc_mkt_x_scaled <- soc_mkt_all %>% select(-X) %>% scale() %>% as.data.frame()
```

Run kmeans functions

```
# k-means group number of each observation
sm_kmeans8 <- kmeans(soc_mkt_x_scaled, 8, iter.max = 50, nstart = 50)

# Visualize k-means clusters
fviz_cluster(sm_kmeans8, data = soc_mkt_scaled, geom = "point", stand = FALSE,
    frame.type = "norm")
```



```
clusters_pars = sm_kmeans8$centers
transposed = t(clusters_pars)
cluster_1 = transposed[which(abs(transposed[, 1]) >= 0.5), 1]
cluster_2 = transposed[which(abs(transposed[, 2]) >= 0.5), 2]
cluster_3 = transposed[which(abs(transposed[, 3]) >= 0.5), 3]
cluster_4 = transposed[which(abs(transposed[, 4]) >= 0.5), 4]
cluster_5 = transposed[which(abs(transposed[, 5]) >= 0.5), 5]
cluster_6 = transposed[which(abs(transposed[, 6]) >= 0.5), 6]
cluster_7 = transposed[which(abs(transposed[, 7]) >= 0.5), 7]
cluster_8 = transposed[which(abs(transposed[, 8]) >= 0.5), 8]
```

```
sports_fandom          food        family        crafts      religion
    2.0679074     1.8422809     1.5030332     0.7163354     2.2590238
    parenting        school
    2.1499150     1.6727291
```

```
numeric(0)
```

```
photo_sharing         music       cooking        beauty       fashion
    1.2128710     0.5290238     2.7779173     2.5561090     2.6469317
```

```
 online_gaming    college_uni sports_playing
      3.498681       3.272318       2.177727
```

```
      chatter photo_sharing       tv_film      shopping
    1.2094136     0.8714549     0.5044661     1.1076373
```

```
    travel   politics       news  computers automotive
  1.896260   2.481778   2.004493   1.668934   1.053455
```

```
health_nutrition              eco         outdoors personal_fitness
       2.1937620        0.5265472        1.7197451        2.1597033
```

```
     spam     adult
12.418865  3.750222
```

```
fact_clust <- factor(sm_kmeans8$cluster)
summary(fact_clust)
```

```
   1    2    3    4    5    6    7    8
 703 3561  513  368 1270  617  801   49
```

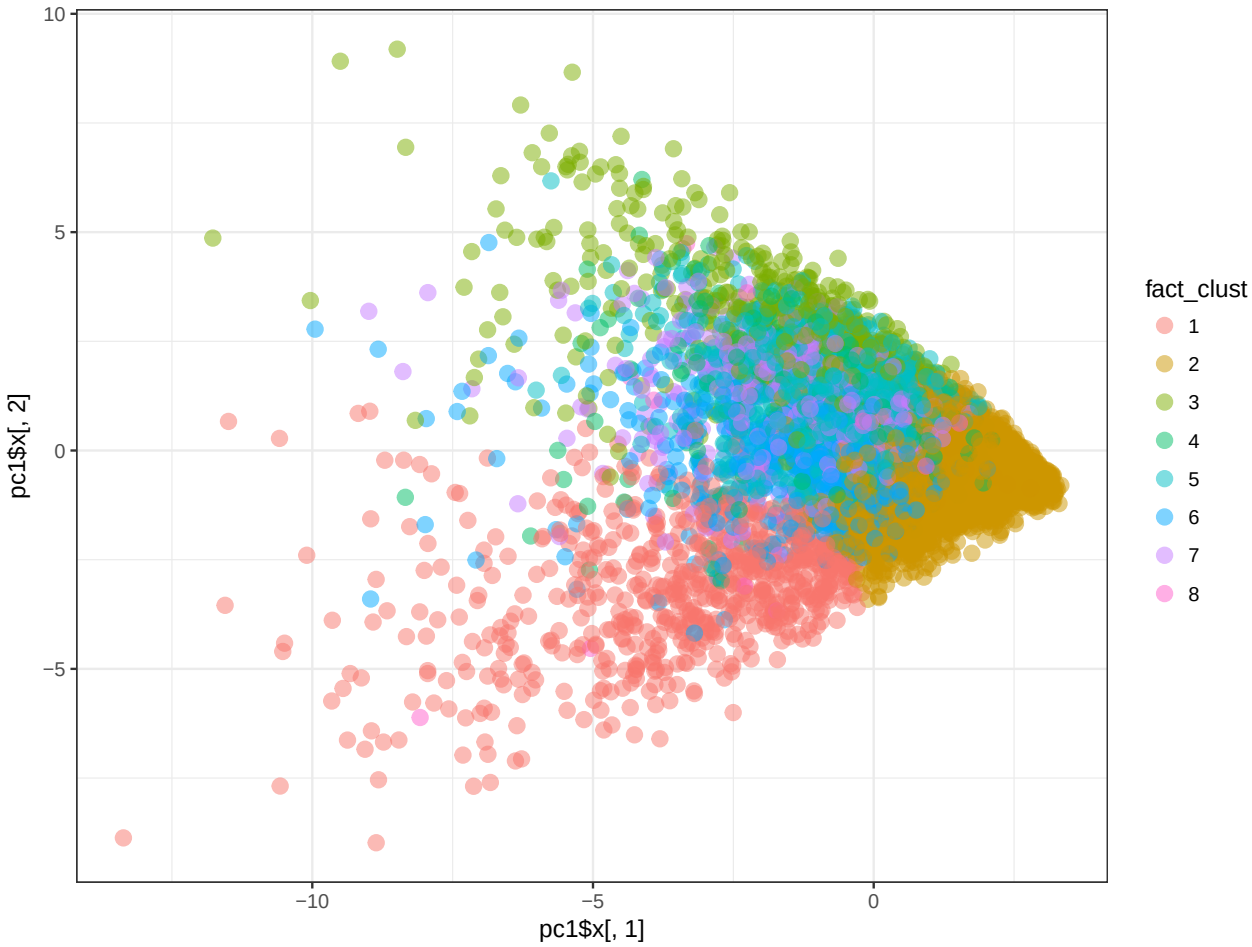
The cluster composition we got from eight clusters makes sense intuitively and is interpretable. We earlier saw a huge cluster getting generalized. Here we again have a group of 3561 users who are not strongly biased or opinionated about any specific genre of topic. The clusters of 8 did do a good job singling out a spam and adult cluster from the more specific clusters which can do a good job with targetted marketing.

Lets see our 8 clusters on the principle component scatter plots we saw earlier:

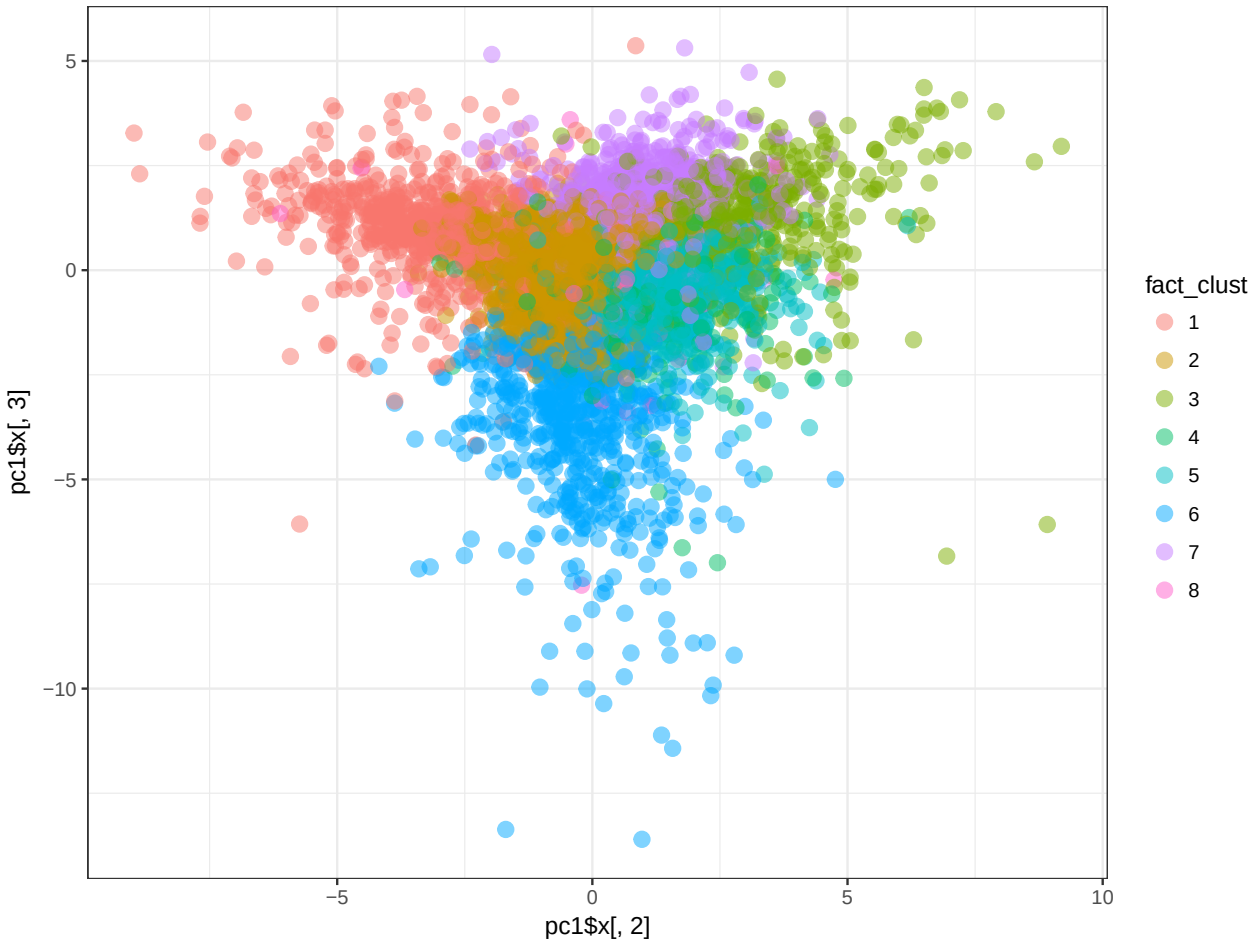
```
par(mfrow = c(1, 1))

fact_clust <- factor(sm_kmeans8$cluster)
summary(fact_clust)
```

```
   1    2    3    4    5    6    7    8
 703 3561  513  368 1270  617  801   49
```

```
ggplot(pc1, aes(x = pc1$x[, 1], y = pc1$x[, 2], col = fact_clust)) + geom_point(size = 3,
    alpha = 0.5) + theme_bw()
```




```
ggplot(pc1, aes(x = pc1$x[, 2], y = pc1$x[, 3], col = fact_clust)) + geom_point(size = 3,
    alpha = 0.5) + theme_bw()
```



These scatterplots seem to confirm our clusters very well. 8 clusters seems to give us specific target groups by successfully removing the spammers into one of the clusters.

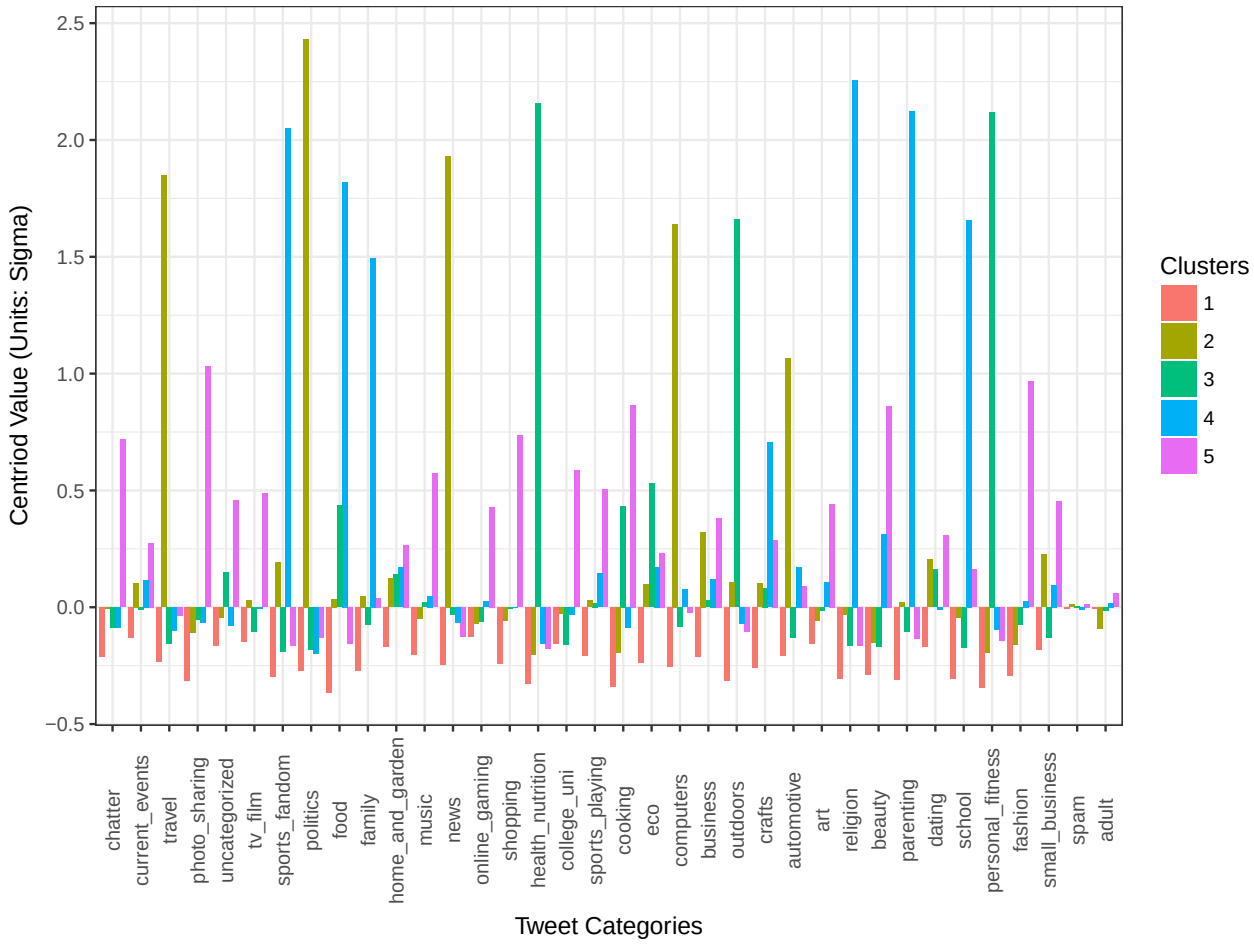
For visualizing these clusters into groups better, we have more graphs!

Melt the centers into long form for use later in visualization

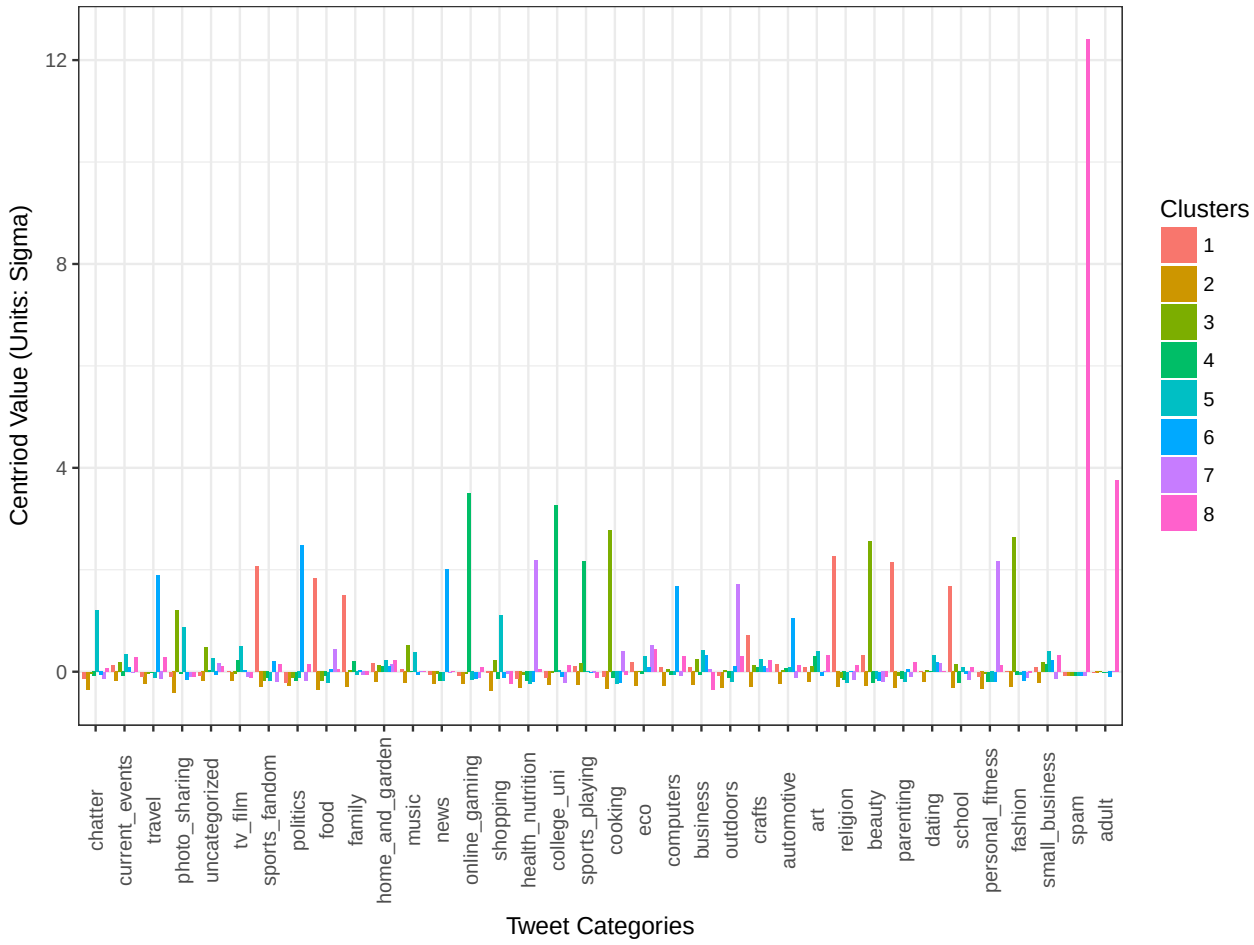
```
library(reshape2)

long5 <- melt(sm_kmeans5$centers)
long8 <- melt(sm_kmeans8$centers)
```

Now compare the centroids by category with a bargraph

```
library(ggplot2)
library(ggthemes)

ggplot(long5) + geom_col(aes(x = Var2, y = value, fill = as.factor(Var1)), position = "dodge") +
    xlab("Tweet Categories") + ylab("Centriod Value (Units: Sigma)") + theme_bw() +
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) + labs(fill = "Clusters")
```

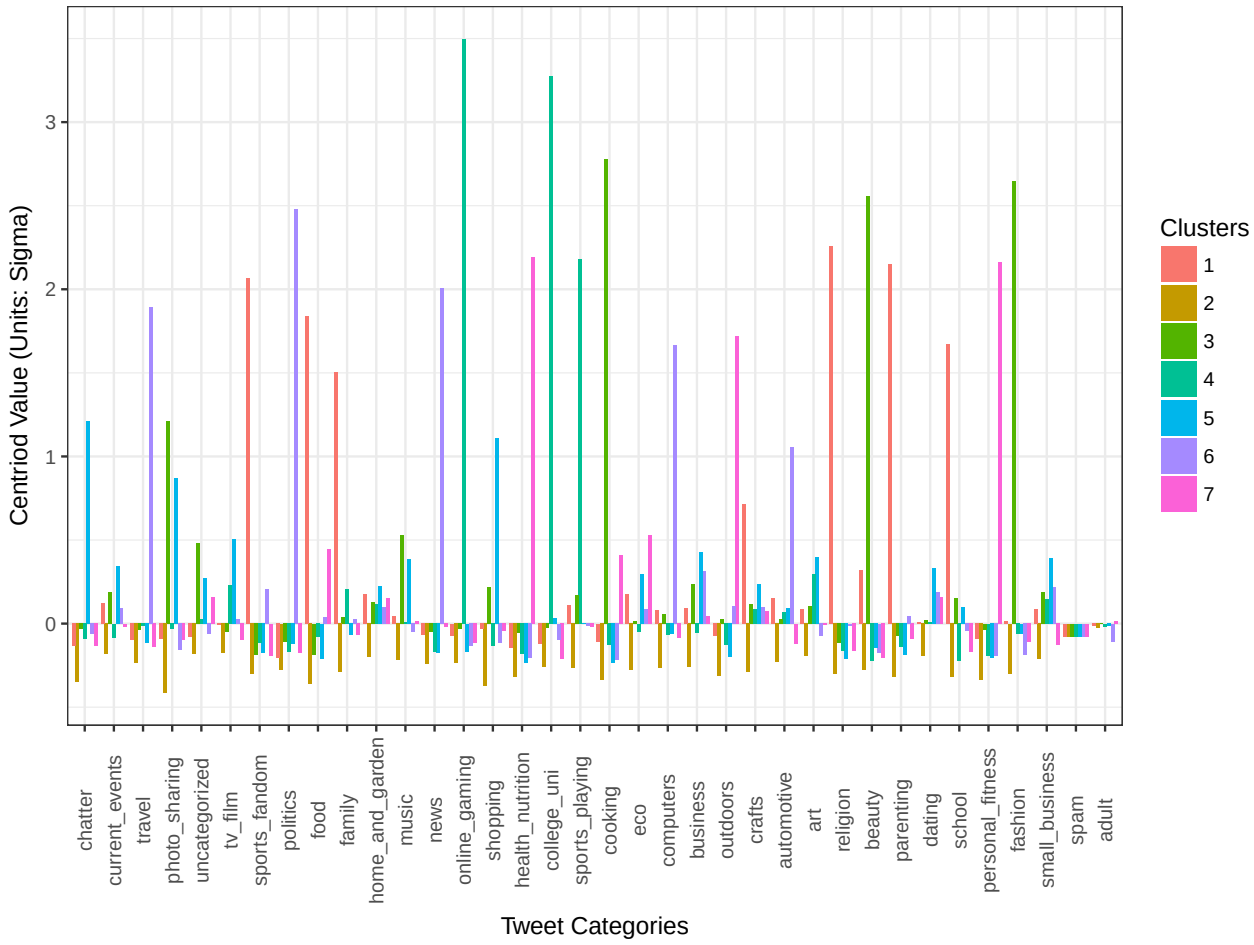



```
ggplot(long8) + geom_col(aes(x = Var2, y = value, fill = as.factor(Var1)), position = "dodge") +
    xlab("Tweet Categories") + ylab("Centriod Value (Units: Sigma)") + theme_bw() +
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) + labs(fill = "Clusters")
```

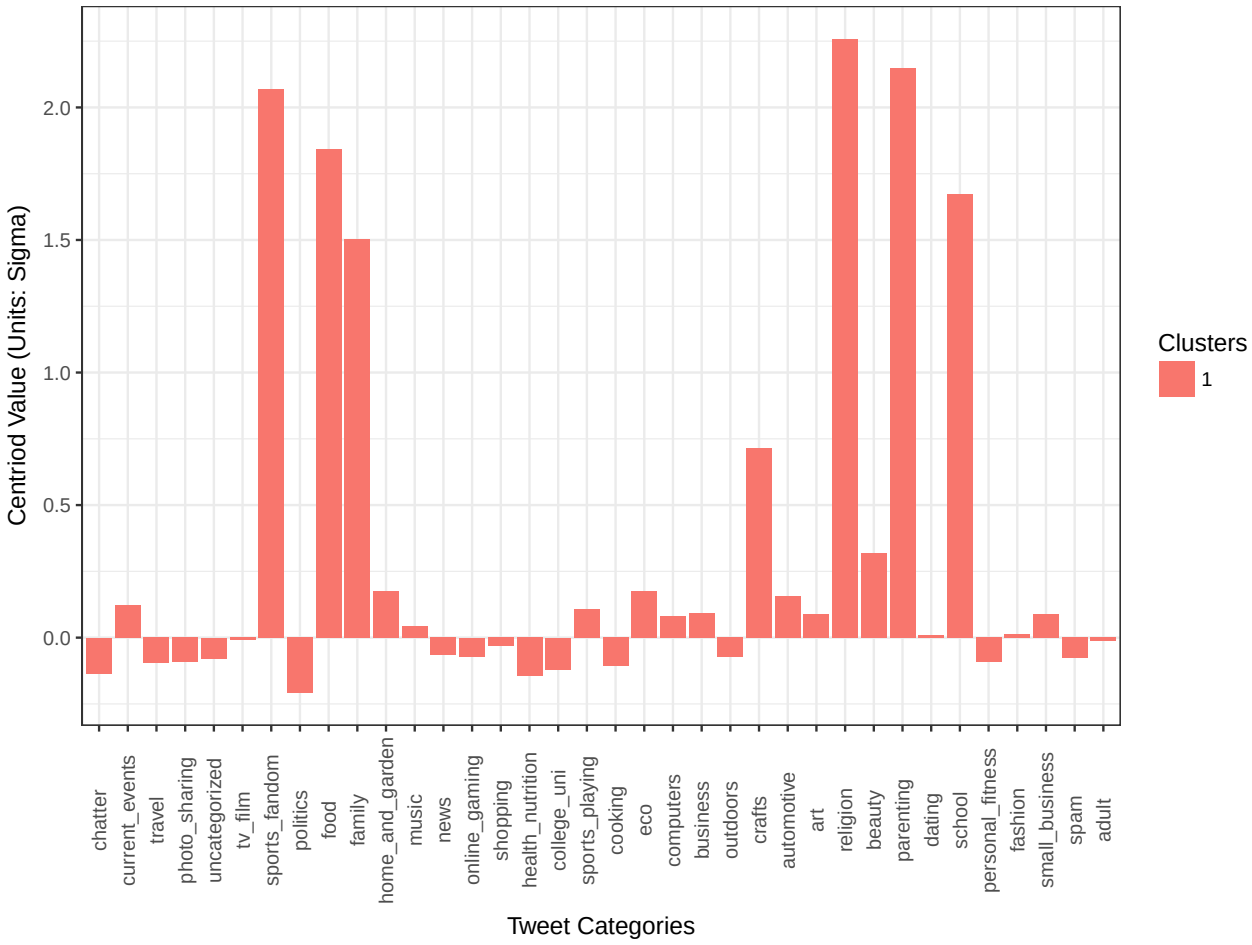


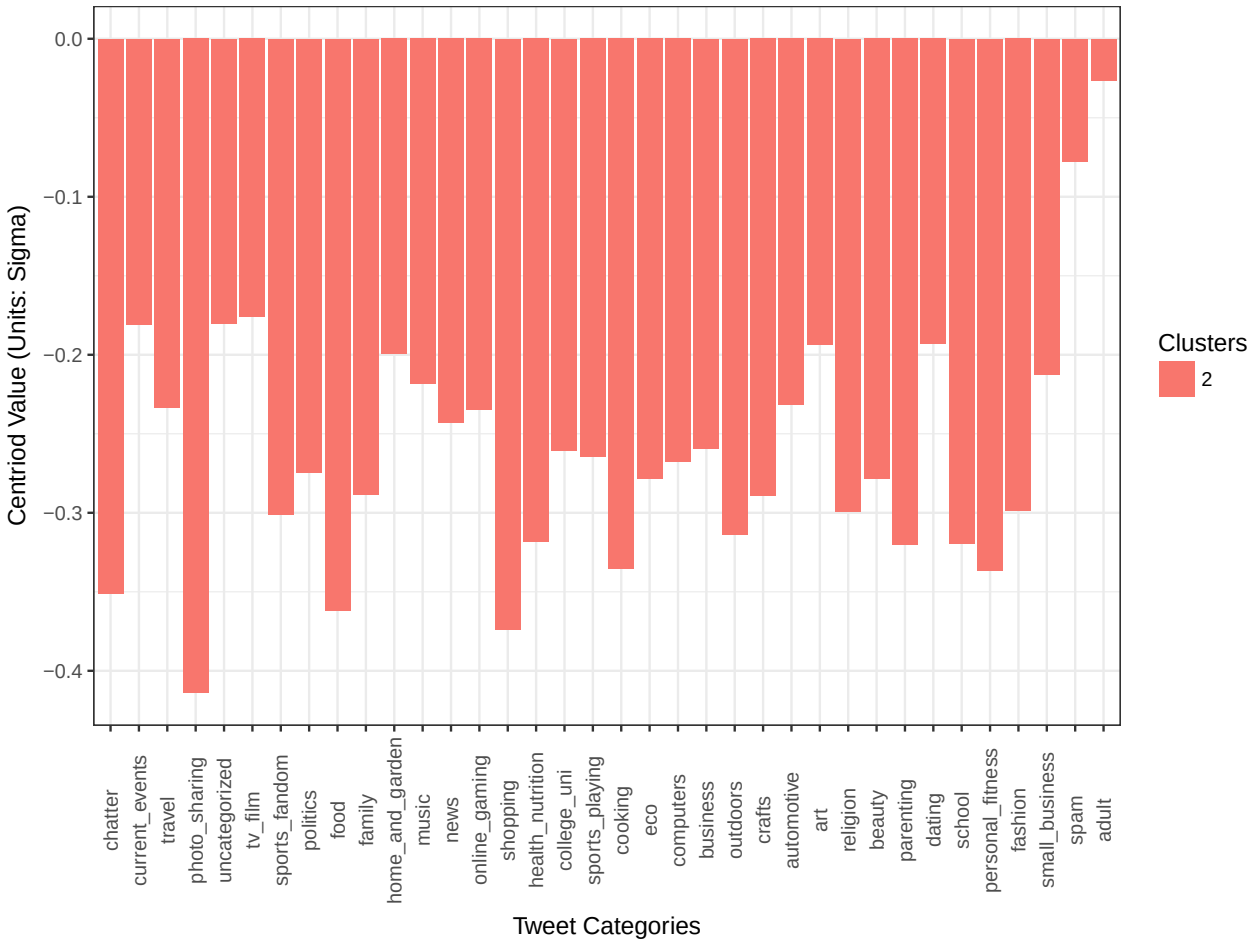
As you can see with 8 clusters kmeans is able to create a cluster that is mostly spammers. Before 8, the spammers were spread throughout the other clusters. There is no reason to market to them, so it is important that we identify who they are. Right now though, they are skewing our otherwise lovely graph, so lets drop them and get to the clusters that we're interested in.

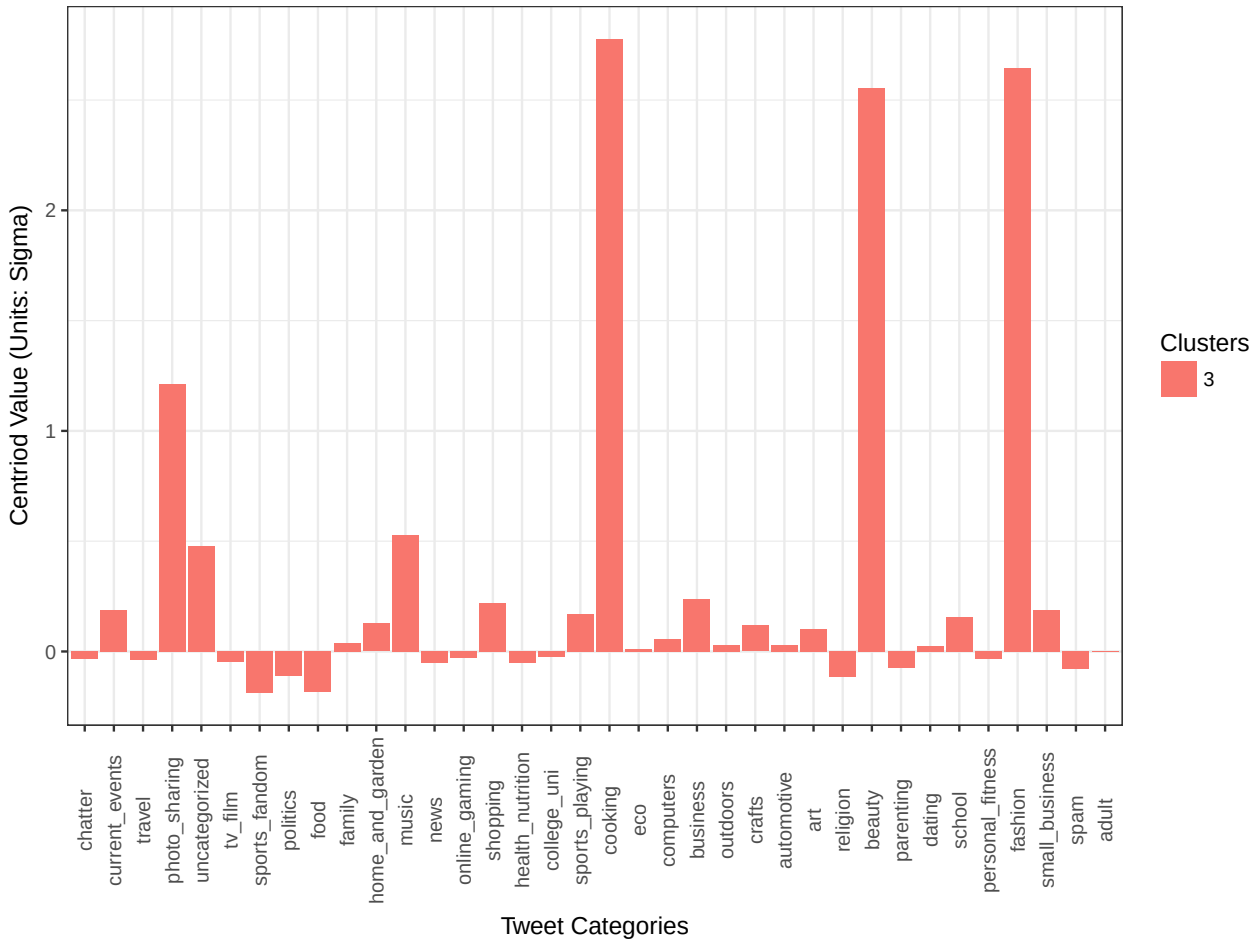
```
long8mod <- long8[-which(long8$Var1 == 8), ]

ggplot(long8mod) + geom_col(aes(x = Var2, y = value, fill = as.factor(Var1)),
    position = "dodge") + xlab("Tweet Categories") + ylab("Centriod Value (Units: Sigma)") +
    theme_bw() + theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
    labs(fill = "Clusters")
```

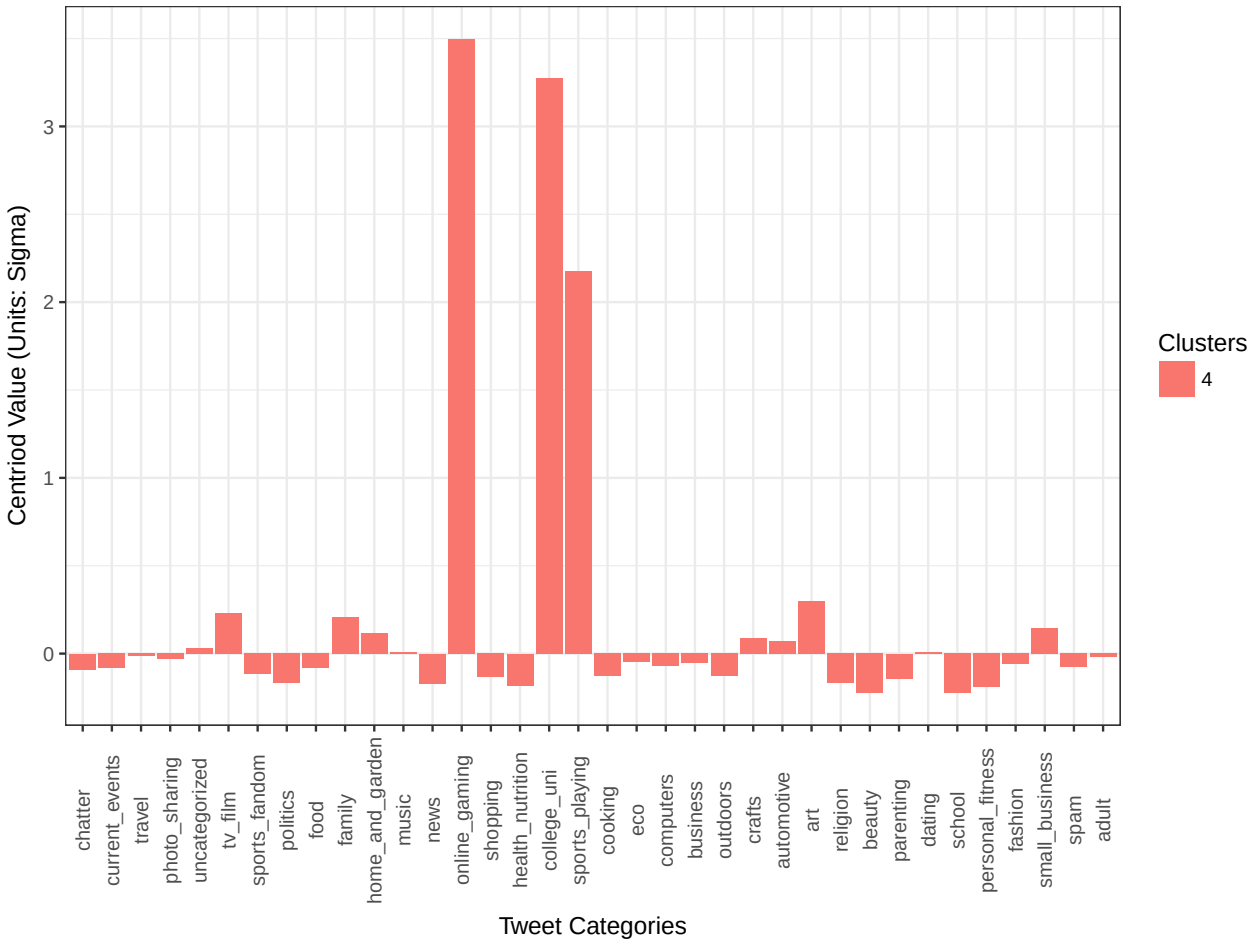


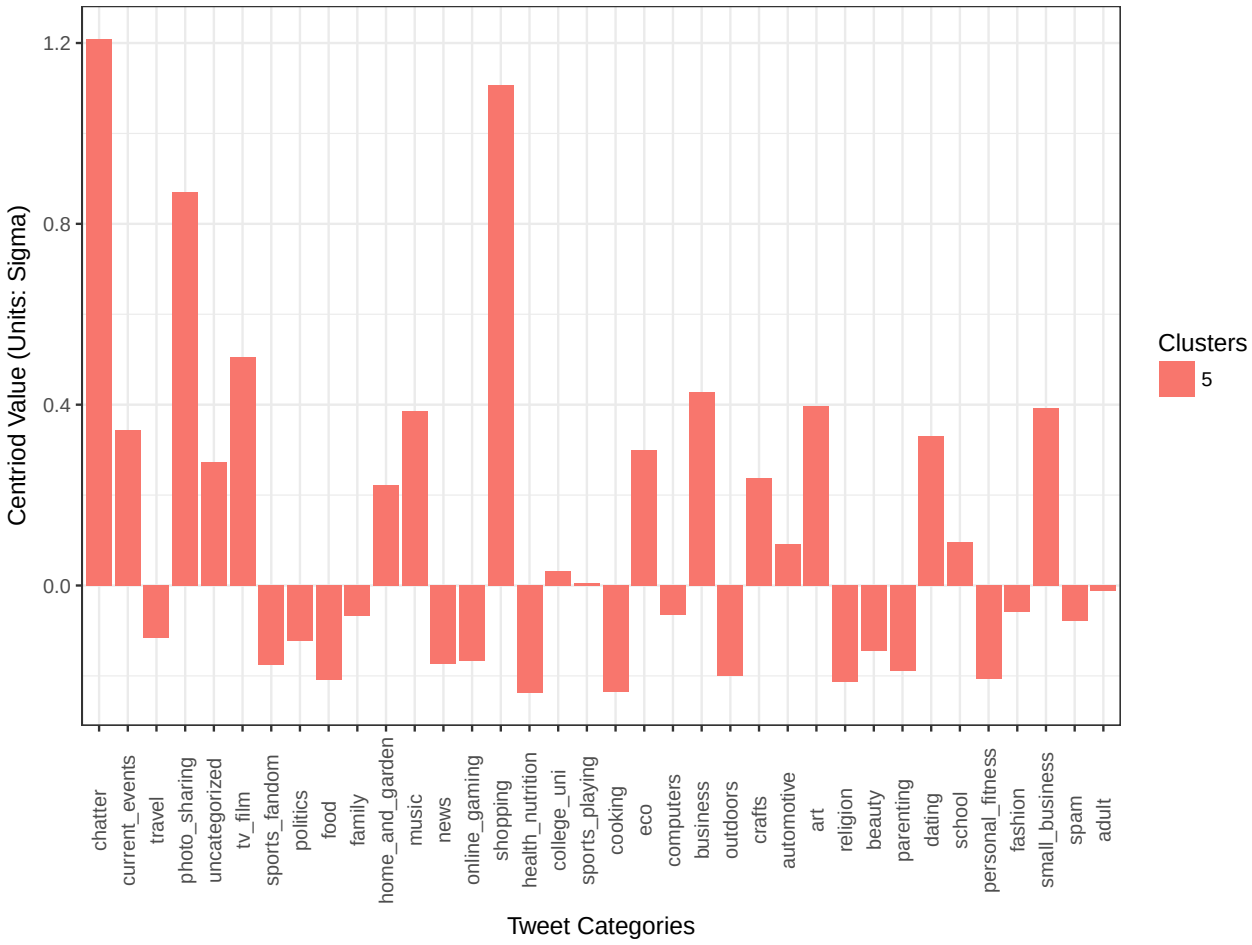
Ah, much better. As you can see, some clusters have centroids whose values for certain categories is really far away from the mean. This indicates that the preferences of the people in those clusters have identifiable interests that we can market to. This visualization is a nice high level overview, but If we look at a cluster we are interested in like 2 you find that its a little difficult to detrimine that they are intersted in school, family, parenting, religion etc. I can barely tell the difference in colors between the light green and the green. So lets look at them individually.

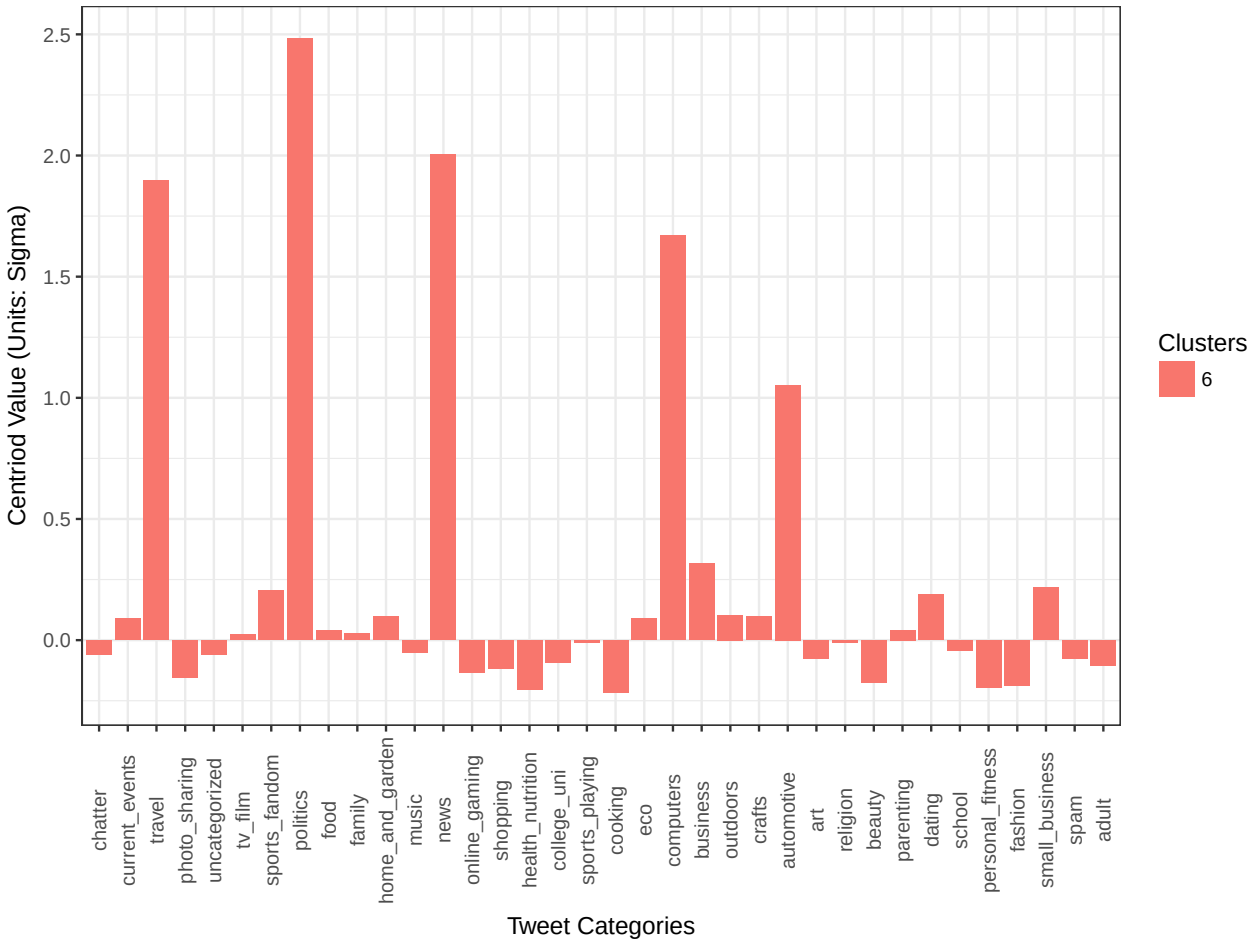

Centriod Value (Units: Sigma)
2.0
1.5
1.0
0.5
0.0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
1

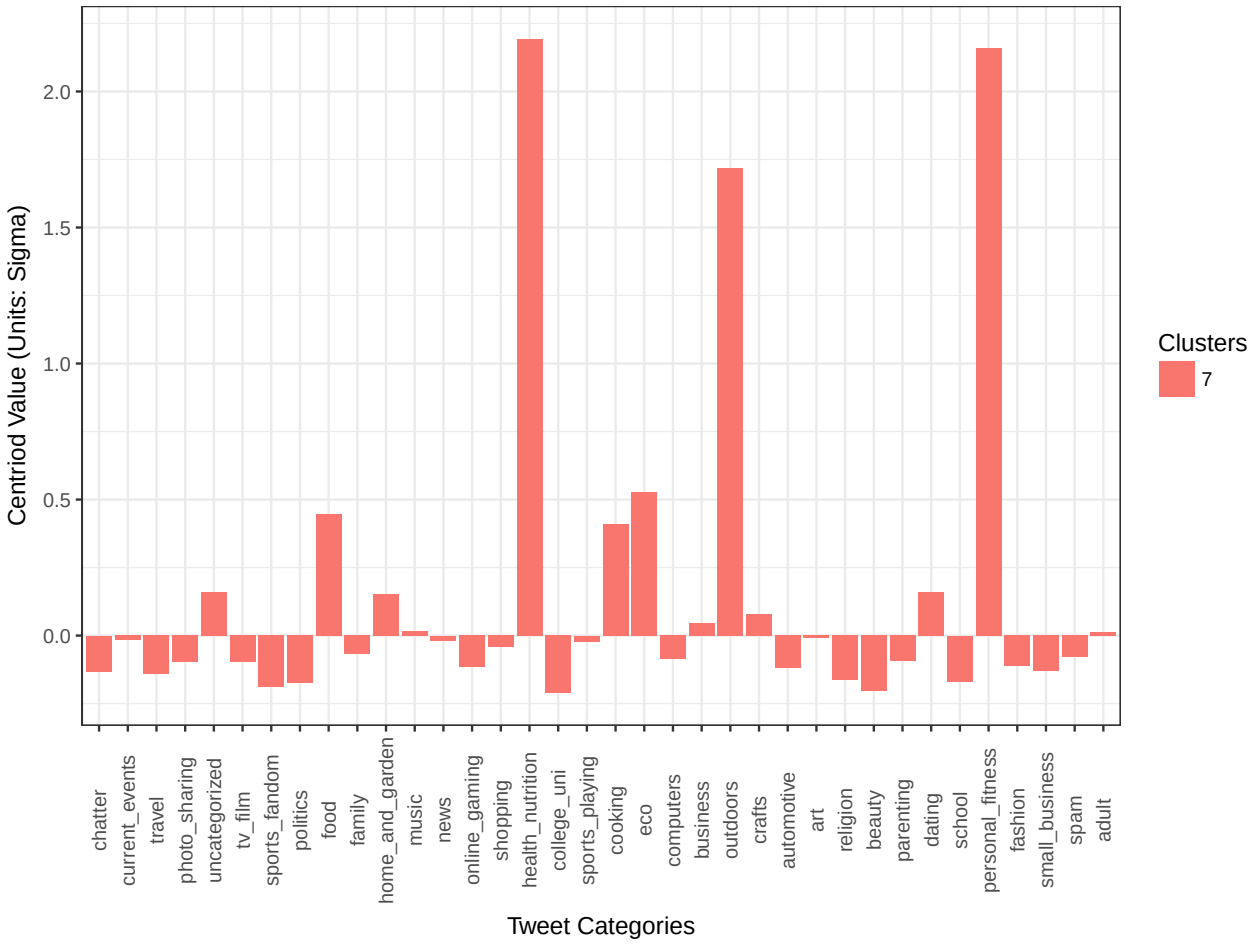

Centriod Value (Units: Sigma)
0.0
−0.1
−0.2
−0.3
−0.4
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
2

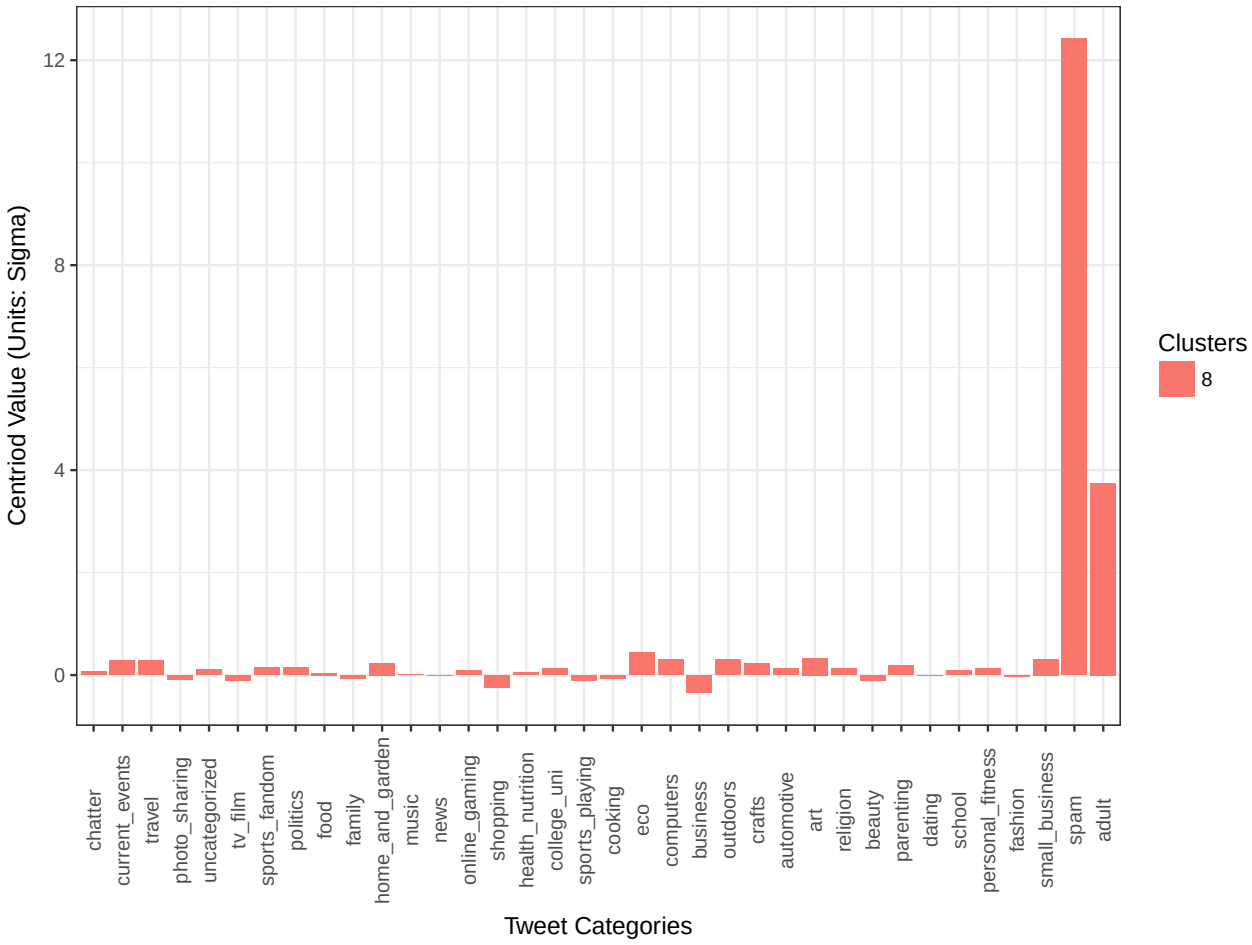

Centriod Value (Units: Sigma)
2
1
0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
3


Centriod Value (Units: Sigma)
3
2
1
0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
4


Centriod Value (Units: Sigma)
1.2
0.8
0.4
0.0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
5


Centriod Value (Units: Sigma)
2.5
2.0
1.5
1.0
0.5
0.0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
6


Centriod Value (Units: Sigma)
2.0
1.5
1.0
0.5
0.0
chatter
current_events
travel
photo_sharing
uncategorized
tv_film
sports_fandom
politics
food
family
home_and_garden
music
news
online_gaming
shopping
health_nutrition
college_uni
sports_playing
cooking
eco
computers
business
outdoors
crafts
automotive
art
religion
beauty
parenting
dating
school
personal_fitness
fashion
small_business
spam
adult
Tweet Categories
Clusters
7



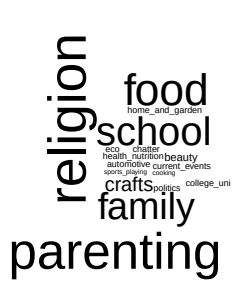
Perfect. Cluster one is our spammers. Cluster two is our religious parents interested in school, food, and watching sports. Cluster 3 is interested in tech, news, politics cars, and travel. And so on. . .

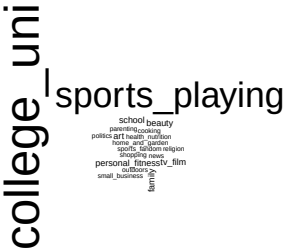
Lastly, in order to visualize the prominent characteristics of each cluster, we used word cloud.

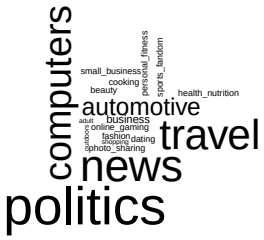
```
# # A word cloud
par(mfrow = c(2, 4))
library(wordcloud)
```

```
Loading required package: RColorBrewer
```

```
for (i in 1:8) {
    wordcloud(labels(sm_kmeans8$centers[i, ]), abs(sm_kmeans8$centers[i, ]),
        min.freq = 0.1, max.words = 50, scale = c(3, 0.3))
}
```







Conclusion

We found 8 distinct clusters to be the best way to visualize these users and target them for marketing. Below are their characterestics in no special order.

- 1) 3561 users. These users tweet about everything or they are inactive. They dont specifically tweet about anything topical. We could use these customers along with any other cluster if more users are required in the campaign
- 2) 513 users. This cluster is a young female cluster. They are active on social media, they like beauty and fashion.
- 3) 617 users. This is a corporate male cluster. They are into computers, politics, news and they travel.
- 4) 801 users. These group of people are big outdoor enthusiasts.
- 5) 368 users. Young college coing people who are into sports.
- 6) 1270 users. Seems like a young demographic that is into a bunch of stuff but mainly into shopping and being active of social media. This cluster can be grouped with cluster 2 for a bigger population of young users.
- 7) 49 users. Prominent spammers who talk alot about adult stuff.
- 8) 703 users. This is a typical parent group. Into religion, school, family and food.

These clusters can be leveraged for specific targetting yeilding higher read rates and conversions.