Probability Practice :

Part A :

Visitors to your website are asked to answer a single survey question before they get access to the content on the page. Among all of the users, there are two categories: Random Clicker (RC), and Truthful Clicker (TC). There are two possible answers to the survey: yes and no. Random clickers would click either one with equal probability. You are also giving the >information that the expected fraction of random clickers is 0.3.

After a trial period, you get the following survey results: 65% said Yes and 35% said No.

What fraction of people who are truthful clickers answered yes?

Probability of yes for a random clicker. $P(Y/R) = 0.5$

Fraction of random clicker $P(R) = 0.3$

Fraction of truthful clicker $P(T) = 0.7$

Fraction of yes $P(Y) = 0.65$

Fraction of no $P(N) = 0.35$

Fraction of yes explained as a sum of conditional probability $P(Y) = P(Y/R)xP(R) + P(Y/T)xP(T)$

$0.65 = 0.5x0.3 + P(Y/T)x0.7$

Fraction of truthful clickers that answered yes $P(Y/T) = 0.714$

Part B :

Imagine a medical test for a disease with the following two attributes:

The sensitivity is about 0.993. That is, if someone has the disease, there is a probability of 0.993 that they will test positive. The specificity is about 0.9999. This means that if someone doesn't have the disease, there is probability of 0.9999 that they will test negative. In the general population, incidence of the disease is reasonably rare: about 0.0025% of all people have it (or 0.000025 as a decimal probability).

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

$P(D)$ = Probability of having the disease

$P(T)$ = Probability of testing positive.

$P(N)$ = Probability of testing negative

$P(ND)$ = Probability of not having disease

$P(T/D)$ = Probability of testing positive given that they have the disease

$P(N/ND)$ = Probability of testing negative given that they do not have the disease

$P(D) = 0.000025$

P(T/D) = 0.993

P(N/ND) = 0.9999

We use Bayes theorem to calculate the probability, P(D/T) which is probability of having the disease given that the test is positive.

P(D/T) = P(T/D)xP(D)/ P(T)

P(T) = P(T/D)xP(D) + (1-P(N/ND))x(1-P(D))

P(D/T) = 0.993 x 0.000025/ (0.993 x 0.000025 + (1-0.9999) x (1 - 0.000025))

P(D/T) = 0.1988

If a universal testing policy is implemented, then the chance that they actually have the disease when they are tested positive is 0.198 which is a low value. The probability that the person does not have the disease is very high (1 - 0.000025). Because of this, if a user tests positive, then it is more likely that they do not have the disease than they do.

Green buildings

```
library(ggplot2)
library(lattice)
library(mosaic)

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: mosaicData

## Loading required package: Matrix

##
## The 'mosaic' package masks several functions from core packages in order t
o add additional features.
## The original behavior of these functions should not be affected by this.

##
## Attaching package: 'mosaic'

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

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cov, D, 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(RCurl)

## Loading required package: bitops

set.seed(100)
rm(list=ls()) #Clear Workspace
temp = getURL("https://raw.githubusercontent.com/matt-staton/stat_380/master/
greenbuildings.csv")
greenbuildings = read.csv(text = temp, header=T)
gbuild = greenbuildings

attach(gbuild)
gbuild$Rent_Diff = Rent - cluster_rent
names(gbuild)

##  [1] "CS_PropertyID"     "cluster"           "size"
##  [4] "empl_gr"           "Rent"              "leasing_rate"
##  [7] "stories"           "age"               "renovated"
## [10] "class_a"           "class_b"           "LEED"
## [13] "Energystar"        "green_rating"      "net"
## [16] "amenities"         "cd_total_07"       "hd_total07"
## [19] "total_dd_07"       "Precipitation"     "Gas_Costs"
## [22] "Electricity_Costs" "cluster_rent"      "Rent_Diff"

summary(gbuild)

##  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     Rent_Diff
##  Min.   :0.009487   Min.   :0.01780   Min.   : 9.00   Min.   :-45.9150
##  1st Qu.:0.010296   1st Qu.:0.02330   1st Qu.:20.00   1st Qu.: -2.9650
##  Median :0.010296   Median :0.03274   Median :25.14   Median :  0.0000
##  Mean   :0.011336   Mean   :0.03096   Mean   :27.50   Mean   :  0.9213
##  3rd Qu.:0.011816   3rd Qu.:0.03781   3rd Qu.:34.00   3rd Qu.:  3.2800
##  Max.   :0.028914   Max.   :0.06280   Max.   :71.44   Max.   :191.2800
##
```

```
lm.fit = lm(Rent ~., data = gbuild)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Rent ~ ., data = gbuild)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -5.284e-12 -3.500e-15  2.000e-16  3.200e-15  5.451e-12
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error    t value Pr(>|t|)
## (Intercept)        7.172e-14  1.136e-14  6.311e+00 2.92e-10 ***
```

```
## CS_PropertyID      1.293e-20  1.750e-21  7.387e+00 1.65e-13 ***
## cluster            1.643e-17  3.157e-18  5.204e+00 2.00e-07 ***
## size              -1.023e-19  7.341e-21 -1.393e+01  < 2e-16 ***
## empl_gr            2.160e-15  1.891e-16  1.142e+01  < 2e-16 ***
## leasing_rate       5.162e-16  5.927e-17  8.711e+00  < 2e-16 ***
## stories           -1.470e-16  1.798e-16 -8.180e-01   0.4136
## age                4.262e-16  5.245e-17  8.126e+00 5.14e-16 ***
## renovated         -2.132e-14  2.874e-15 -7.418e+00 1.31e-13 ***
## class_a            3.958e-14  4.878e-15  8.116e+00 5.57e-16 ***
## class_b            6.067e-15  3.812e-15  1.592e+00   0.1115
## LEED               1.309e-14  3.981e-14  3.290e-01   0.7422
## Energystar         5.628e-14  4.243e-14  1.326e+00   0.1847
## green_rating      -7.069e-14  4.266e-14 -1.657e+00   0.0975 .
## net               -2.460e-15  6.597e-15 -3.730e-01   0.7093
## amenities         -4.580e-15  2.801e-15 -1.636e+00   0.1020
## cd_total_07       -1.284e-17  1.628e-18 -7.890e+00 3.42e-15 ***
## hd_total07         3.904e-19  9.994e-19  3.910e-01   0.6961
## total_dd_07               NA         NA         NA       NA
## Precipitation      2.724e-15  1.792e-16  1.520e+01  < 2e-16 ***
## Gas_Costs         -9.165e-12  8.727e-13 -1.050e+01  < 2e-16 ***
## Electricity_Costs  5.655e-12  2.781e-13  2.033e+01  < 2e-16 ***
## cluster_rent       1.000e+00  1.580e-16  6.331e+15  < 2e-16 ***
## Rent_Diff          1.000e+00  1.259e-16  7.946e+15  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.046e-13 on 7797 degrees of freedom
##   (74 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 7.408e+30 on 22 and 7797 DF,  p-value: < 2.2e-16
```

```
confint(lm.fit)
```

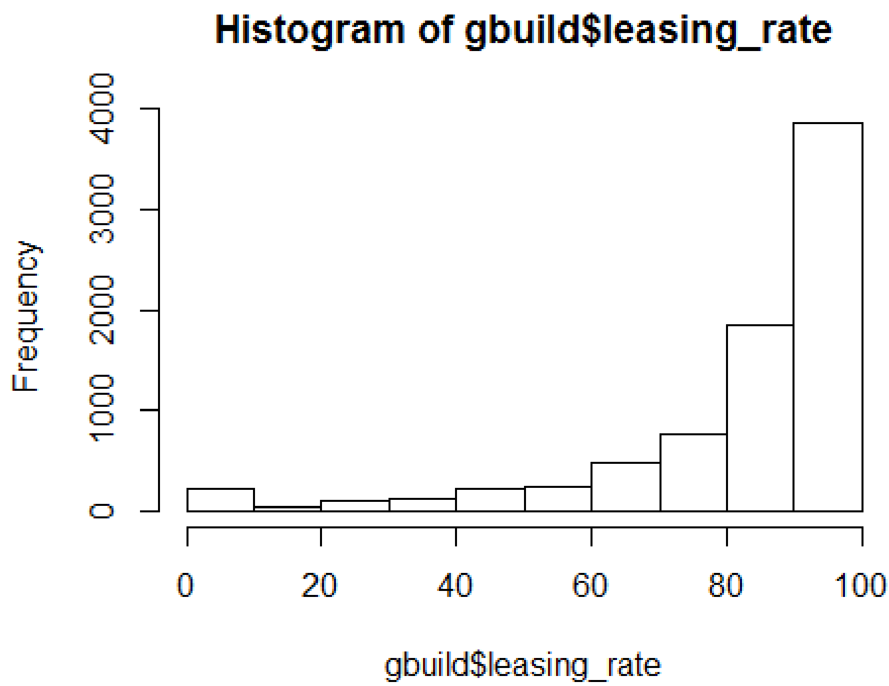
```
##                           2.5 %        97.5 %
## (Intercept)        4.944170e-14  9.399177e-14
## CS_PropertyID      9.497826e-21  1.635919e-20
## cluster            1.023998e-17  2.261785e-17
## size              -1.166696e-19 -8.789077e-20
## empl_gr            1.789213e-15  2.530539e-15
## leasing_rate       4.000657e-16  6.324173e-16
## stories           -4.993372e-16  2.053959e-16
## age                3.233495e-16  5.289668e-16
## renovated         -2.695144e-14 -1.568444e-14
## class_a            3.002321e-14  4.914599e-14
## class_b           -1.405133e-15  1.353961e-14
## LEED              -6.494246e-14  9.113091e-14
## Energystar        -2.689349e-14  1.394475e-13
## green_rating      -1.543178e-13  1.293406e-14
## net               -1.539271e-14  1.047289e-14
## amenities         -1.007010e-14  9.093739e-16
```

```
## cd_total_07       -1.603215e-17 -9.651347e-18
## hd_total07        -1.568757e-18  2.349594e-18
## total_dd_07                  NA            NA
## Precipitation      2.372503e-15  3.075061e-15
## Gas_Costs         -1.087532e-11 -7.453807e-12
## Electricity_Costs  5.109408e-12  6.199704e-12
## cluster_rent       1.000000e+00  1.000000e+00
## Rent_Diff          1.000000e+00  1.000000e+00
```

We can see here controlling for all avaialbe variables that the most significant predictors of price are PropertyID, cluster(ie:location), size, employment growth, cluster Rent, stories, leasing rate, hd-total-07 (total heating days in 2007), precipitation, Gas costs, age, class A, class B, net, amenities and electricity costs.

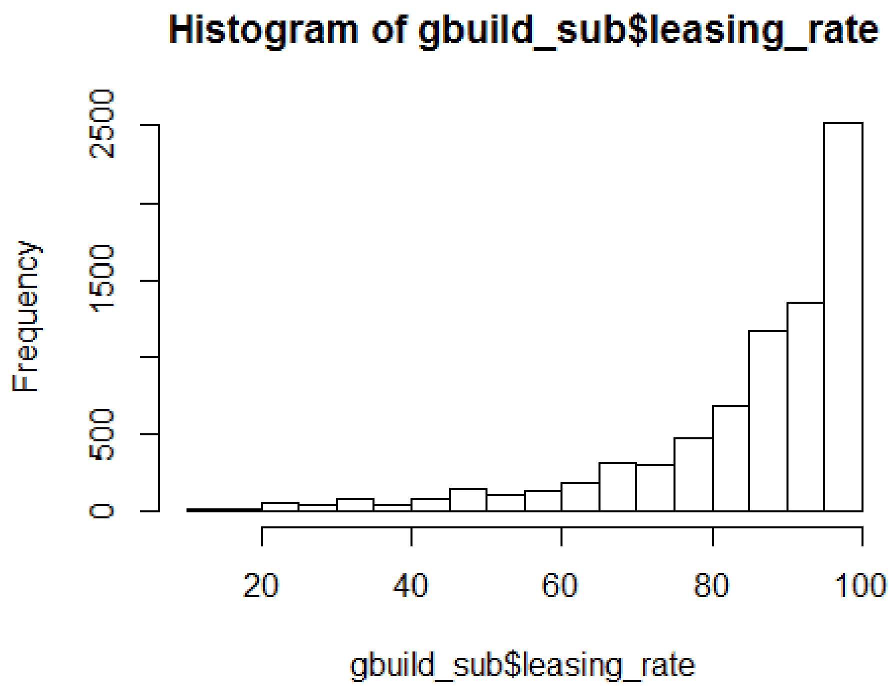
These predictors seem intuitive with the exception of green_rating having almost no predictive power. The green rating has a 95% CI of -6.827645e+00 8.221535e+00, which includes 0, leaving us to accept the null hypothesis that green rating is not statistically significant. Furthermore, I was very surprised to see renovation having very little predictive power, with a 95% confidence interval of -6.493550e-01 3.644221e-01.

```
names(lm.fit)

##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "na.action"     "xlevels"       "call"          "terms"
## [13] "model"

hist(gbuild$leasing_rate,plot=TRUE)
```

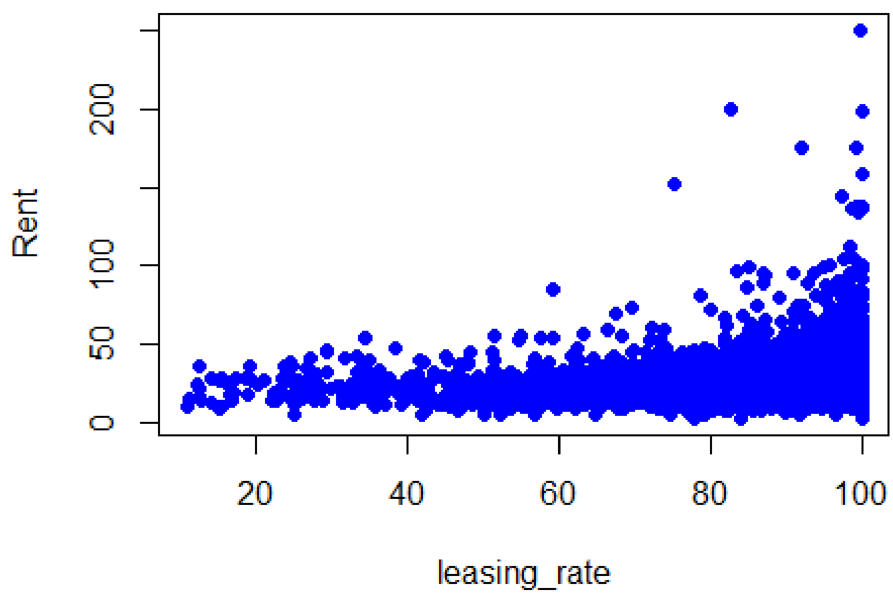


In order to do an apples to apples comparison of the previous analysis I will re-run the model with leasing rates >= 10%. Let's also plot rent as function of leasing rate to understand the effect removing the bottom 10 percentil will have.

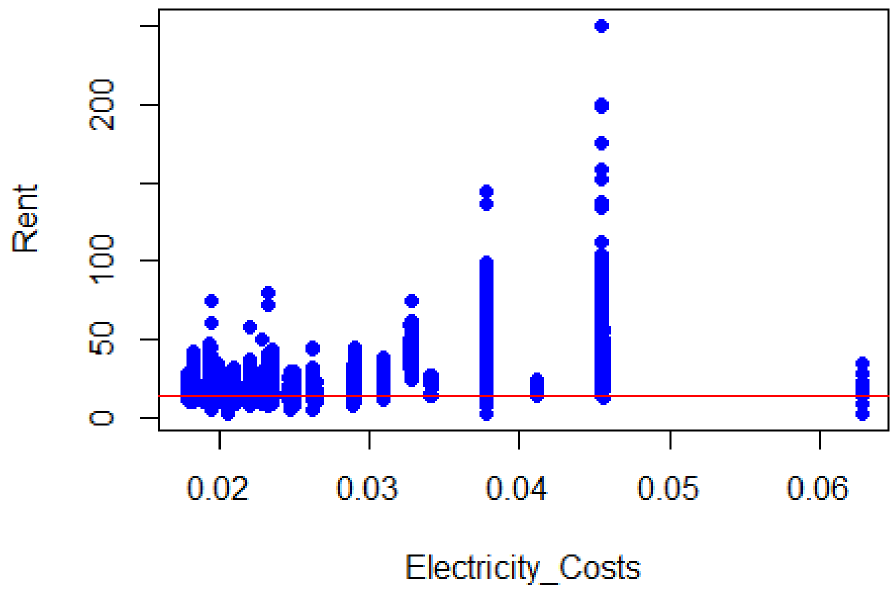
```
gbuild_sub = subset(gbuild, gbuild$leasing_rate >= 10)  #Remove lease rates < 10%

hist(gbuild_sub$leasing_rate,plot=TRUE)
```




```r
#Rent positively correlated with leasing rate
plot(Rent~leasing_rate,data = gbuild_sub, col="blue",pch=16)
```



```r
#Rent positively correlated with electricity costs
plot(Rent~Electricity_Costs,data = gbuild_sub, col="blue",pch=16)

abline(lm(Rent~leasing_rate,data = gbuild_sub),col="red")
```




```r
gbuild_sub$util_index = gbuild_sub$hd_total07*gbuild_sub$Gas_Costs +
    gbuild_sub$cd_total_07 * gbuild_sub$Electricity_Costs
lm.fit2 = lm(Rent ~.+util_index*class_a+cd_total_07*class_a+green_rating*clas
s_a+empl_gr*class_a+empl_gr*green_rating, data = gbuild_sub)
summary(lm.fit2)

##
## Call:
## lm(formula = Rent ~ . + util_index * class_a + cd_total_07 *
##     class_a + green_rating * class_a + empl_gr * class_a + empl_gr *
##     green_rating, data = gbuild_sub)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.221e-12 -1.100e-14  1.000e-15  1.040e-14  2.559e-11
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error    t value Pr(>|t|)
## (Intercept)             1.696e-13  7.403e-14  2.291e+00  0.02202 *
## CS_PropertyID           2.623e-20  5.412e-21  4.847e+00 1.28e-06 ***
## cluster                -2.619e-17  9.068e-18 -2.889e+00  0.00388 **
## size                   -3.519e-20  2.088e-20 -1.685e+00  0.09197 .
```

```
## empl_gr               -3.503e-16  7.524e-16 -4.660e-01  0.64158    
## leasing_rate           3.802e-16  2.150e-16  1.768e+00  0.07704 .  
## stories                2.598e-16  5.126e-16  5.070e-01  0.61235    
## age                    8.216e-17  1.540e-16  5.330e-01  0.59378    
## renovated              1.577e-16  8.302e-15  1.900e-02  0.98484    
## class_a               -1.894e-14  2.036e-14 -9.310e-01  0.35213    
## class_b               -6.154e-15  1.116e-14 -5.510e-01  0.58147    
## LEED                   2.406e-14  1.129e-13  2.130e-01  0.83118    
## Energystar             5.744e-14  1.203e-13  4.770e-01  0.63309    
## green_rating          -3.586e-14  1.223e-13 -2.930e-01  0.76940    
## net                   -2.120e-14  1.886e-14 -1.124e+00  0.26101    
## amenities              7.752e-15  8.077e-15  9.600e-01  0.33719    
## cd_total_07            2.097e-17  1.892e-17  1.108e+00  0.26774    
## hd_total07             7.324e-18  7.790e-18  9.400e-01  0.34718    
## total_dd_07                   NA         NA         NA       NA    
## Precipitation         -5.308e-16  5.195e-16 -1.022e+00  0.30693    
## Gas_Costs              5.085e-12  4.338e-12  1.172e+00  0.24116    
## Electricity_Costs     -5.203e-13  1.156e-12 -4.500e-01  0.65278    
## cluster_rent           1.000e+00  4.620e-16  2.164e+15  < 2e-16 ***
## Rent_Diff              1.000e+00  3.592e-16  2.784e+15  < 2e-16 ***
## util_index            -5.825e-16  5.953e-16 -9.790e-01  0.32781    
## class_a:util_index    -6.081e-16  2.966e-16 -2.050e+00  0.04039 *  
## class_a:cd_total_07    5.131e-17  1.184e-17  4.332e+00 1.50e-05 ***
## class_a:green_rating   3.311e-14  2.966e-14  1.116e+00  0.26427    
## empl_gr:class_a       -2.316e-15  1.017e-15 -2.278e+00  0.02275 *  
## empl_gr:green_rating -5.280e-16  1.330e-15 -3.970e-01  0.69150    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.963e-13 on 7577 degrees of freedom
##   (73 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1 
## F-statistic: 7.118e+29 on 28 and 7577 DF,  p-value: < 2.2e-16
```

```
confint(lm.fit2)
```

```
##                              2.5 %        97.5 %
## (Intercept)           2.444902e-14  3.146911e-13
## CS_PropertyID         1.562321e-20  3.683954e-20
## cluster              -4.396784e-17 -8.417897e-18
## size                 -7.613067e-20  5.742406e-21
## empl_gr              -1.825168e-15  1.124666e-15
## leasing_rate         -4.125658e-17  8.016243e-16
## stories              -7.450957e-16  1.264617e-15
## age                  -2.197758e-16  3.840877e-16
## renovated            -1.611641e-14  1.643183e-14
## class_a              -5.884580e-14  2.096202e-14
## class_b              -2.803693e-14  1.572905e-14
## LEED                 -1.971601e-13  2.452805e-13
## Energystar           -1.784055e-13  2.932774e-13
```

```
## green_rating         -2.756190e-13  2.039027e-13
## net                  -5.818207e-14  1.577347e-14
## amenities            -8.080561e-15  2.358441e-14
## cd_total_07          -1.612036e-17  5.806592e-17
## hd_total07           -7.947044e-18  2.259476e-17
## total_dd_07                     NA             NA
## Precipitation        -1.549050e-15  4.875257e-16
## Gas_Costs            -3.418602e-12  1.358805e-11
## Electricity_Costs    -2.787123e-12  1.746545e-12
## cluster_rent          1.000000e+00  1.000000e+00
## Rent_Diff             1.000000e+00  1.000000e+00
## util_index           -1.749422e-15  5.843577e-16
## class_a:util_index   -1.189638e-15 -2.664067e-17
## class_a:cd_total_07   2.808932e-17  7.452550e-17
## class_a:green_rating -2.502430e-14  9.124126e-14
## empl_gr:class_a      -4.308981e-15 -3.231612e-16
## empl_gr:green_rating -3.135996e-15  2.080056e-15
```

We can see leasing rate has a positive effect on rent prices, we won't do anything with that for now, but we may need to control for that in the future.

Again we see none of the green IV's have any reliable predictave power as it relates to price. The most meaningful predictors remain to be Size,Employment growth, Class A, Class B, Net, HD-total07, Gas-Costs, Electricity-Costs, and Cluster_Rent (IE: Neighboorhood average rent)

The analysis doesn't account for lurking variables by just looking at median rent. It assumes the higher prices of green buildings are due to them being green, when we can clearly see from the regression model, they are not. This is because the regression model shows the marginal effect of each variable, and allows one to control for other factors. Furthermore, in order to reliably predict the price of the building, the developer should input the values of the most important predictors above to estimate. Using the median however was a good idea, because it is more robust to outliers.

Lets remove all statistically insignificant variables using step-wise regression;acknowledging that the coefficients may change slightly

```
lm.fit3 = step(lm.fit2 , scope=formula(lm.fit2), direction="back", k=log(leng
th(gbuild_sub)))
```

```
## Start:  AIC=-438764.6
## Rent ~ CS_PropertyID + cluster + size + empl_gr + leasing_rate +
##     stories + age + renovated + class_a + class_b + LEED + Energystar +
##     green_rating + net + amenities + cd_total_07 + hd_total07 +
##     total_dd_07 + Precipitation + Gas_Costs + Electricity_Costs +
##     cluster_rent + Rent_Diff + util_index + util_index * class_a +
##     cd_total_07 * class_a + green_rating * class_a + empl_gr *
##     class_a + empl_gr * green_rating
```

```
## Warning: attempting model selection on an essentially perfect fit is
## nonsense

##
## Step:  AIC=-438764.6
## Rent ~ CS_PropertyID + cluster + size + empl_gr + leasing_rate +
##     stories + age + renovated + class_a + class_b + LEED + Energystar +
##     green_rating + net + amenities + cd_total_07 + hd_total07 +
##     Precipitation + Gas_Costs + Electricity_Costs + cluster_rent +
##     Rent_Diff + util_index + class_a:util_index + class_a:cd_total_07 +
##     class_a:green_rating + empl_gr:class_a + empl_gr:green_rating

## Warning: attempting model selection on an essentially perfect fit is
## nonsense

##                        Df Sum of Sq    RSS     AIC
## - empl_gr:green_rating  1         0      0 -438767
## - class_a:green_rating  1         0      0 -438766
## <none>                                   0 -438765
## - class_a:util_index    1         0      0 -438763
## - empl_gr:class_a       1         0      0 -438762
## - LEED                  1         0      0 -438760
## - CS_PropertyID         1         0      0 -438753
## - class_a:cd_total_07   1         0      0 -438749
## - Electricity_Costs     1         0      0 -438737
## - leasing_rate          1         0      0 -438736
## - renovated             1         0      0 -438723
## - Energystar            1         0      0 -438715
## - hd_total07            1         0      0 -438714
## - stories               1         0      0 -438711
## - Precipitation         1         0      0 -438655
## - class_b               1         0      0 -438629
## - net                   1         0      0 -438586
## - amenities             1         0      0 -438520
## - Gas_Costs             1         0      0 -438519
## - age                   1         0      0 -438514
## - size                  1         0      0 -438466
## - cluster               1         0      0 -438168
## - cluster_rent          1    411137 411137   30438
## - Rent_Diff             1    680136 680136   34267
##
## Step:  AIC=-438767.6
## Rent ~ CS_PropertyID + cluster + size + empl_gr + leasing_rate +
##     stories + age + renovated + class_a + class_b + LEED + Energystar +
##     green_rating + net + amenities + cd_total_07 + hd_total07 +
##     Precipitation + Gas_Costs + Electricity_Costs + cluster_rent +
##     Rent_Diff + util_index + class_a:util_index + class_a:cd_total_07 +
##     class_a:green_rating + empl_gr:class_a

## Warning: attempting model selection on an essentially perfect fit is
## nonsense
```

```
##                         Df Sum of Sq    RSS     AIC
## - class_a:green_rating  1         0      0 -438769
## <none>                                   0 -438768
## - class_a:util_index    1         0      0 -438766
## - empl_gr:class_a       1         0      0 -438764
## - LEED                  1         0      0 -438763
## - CS_PropertyID         1         0      0 -438755
## - class_a:cd_total_07   1         0      0 -438752
## - Electricity_Costs     1         0      0 -438740
## - leasing_rate          1         0      0 -438739
## - renovated             1         0      0 -438725
## - Energystar            1         0      0 -438718
## - hd_total07            1         0      0 -438717
## - stories               1         0      0 -438714
## - Precipitation         1         0      0 -438658
## - class_b               1         0      0 -438632
## - net                   1         0      0 -438584
## - amenities             1         0      0 -438523
## - Gas_Costs             1         0      0 -438521
## - age                   1         0      0 -438516
## - size                  1         0      0 -438466
## - cluster               1         0      0 -438171
## - cluster_rent          1    411193 411193   30436
## - Rent_Diff             1    680136 680136   34263
##
## Step:  AIC=-438769.7
## Rent ~ CS_PropertyID + cluster + size + empl_gr + leasing_rate +
##     stories + age + renovated + class_a + class_b + LEED + Energystar +
##     green_rating + net + amenities + cd_total_07 + hd_total07 +
##     Precipitation + Gas_Costs + Electricity_Costs + cluster_rent +
##     Rent_Diff + util_index + class_a:util_index + class_a:cd_total_07 +
##     empl_gr:class_a

## Warning: attempting model selection on an essentially perfect fit is
## nonsense

##                        Df Sum of Sq    RSS     AIC
## <none>                                   0 -438770
## - green_rating          1         0      0 -438768
## - class_a:util_index    1         0      0 -438768
## - empl_gr:class_a       1         0      0 -438766
## - LEED                  1         0      0 -438765
## - CS_PropertyID         1         0      0 -438757
## - class_a:cd_total_07   1         0      0 -438753
## - Electricity_Costs     1         0      0 -438742
## - leasing_rate          1         0      0 -438741
## - renovated             1         0      0 -438727
## - Energystar            1         0      0 -438720
## - hd_total07            1         0      0 -438719
## - stories               1         0      0 -438715
```

```
## - Precipitation        1         0      0 -438660
## - class_b              1         0      0 -438634
## - net                  1         0      0 -438591
## - amenities            1         0      0 -438525
## - Gas_Costs            1         0      0 -438523
## - age                  1         0      0 -438518
## - size                 1         0      0 -438469
## - cluster              1         0      0 -438173
## - cluster_rent         1    411770 411770   30443
## - Rent_Diff            1    680334 680334   34262
```

```
summary(lm.fit3)
```

```
## 
## Call:
## lm(formula = Rent ~ CS_PropertyID + cluster + size + empl_gr + 
##     leasing_rate + stories + age + renovated + class_a + class_b + 
##     LEED + Energystar + green_rating + net + amenities + cd_total_07 + 
##     hd_total07 + Precipitation + Gas_Costs + Electricity_Costs + 
##     cluster_rent + Rent_Diff + util_index + class_a:util_index + 
##     class_a:cd_total_07 + empl_gr:class_a, data = gbuild_sub)
## 
## Residuals:
##        Min         1Q     Median         3Q        Max 
## -1.219e-12 -1.120e-14  1.300e-15  1.020e-14  2.559e-11 
## 
## Coefficients:
##                      Estimate Std. Error    t value Pr(>|t|)    
## (Intercept)         1.659e-13  7.400e-14  2.242e+00  0.02496 *  
## CS_PropertyID       2.574e-20  5.408e-21  4.759e+00 1.98e-06 ***
## cluster            -2.849e-17  9.067e-18 -3.142e+00  0.00168 ** 
## size               -3.581e-20  2.088e-20 -1.715e+00  0.08636 .  
## empl_gr            -3.208e-16  7.515e-16 -4.270e-01  0.66951    
## leasing_rate        4.159e-16  2.150e-16  1.935e+00  0.05304 .  
## stories             2.495e-16  5.125e-16  4.870e-01  0.62645    
## age                 8.009e-17  1.537e-16  5.210e-01  0.60240    
## renovated           4.873e-16  8.301e-15  5.900e-02  0.95319    
## class_a            -1.575e-14  2.017e-14 -7.810e-01  0.43503    
## class_b            -6.861e-15  1.115e-14 -6.150e-01  0.53841    
## LEED                2.681e-14  1.128e-13  2.380e-01  0.81219    
## Energystar          6.321e-14  1.202e-13  5.260e-01  0.59899    
## green_rating       -1.755e-14  1.209e-13 -1.450e-01  0.88457    
## net                -2.121e-14  1.886e-14 -1.125e+00  0.26073    
## amenities           7.999e-15  8.073e-15  9.910e-01  0.32183    
## cd_total_07         2.087e-17  1.892e-17  1.103e+00  0.27014    
## hd_total07          7.712e-18  7.788e-18  9.900e-01  0.32212    
## Precipitation      -5.167e-16  5.193e-16 -9.950e-01  0.31977    
## Gas_Costs           4.967e-12  4.336e-12  1.145e+00  0.25209    
## Electricity_Costs  -3.902e-13  1.156e-12 -3.370e-01  0.73576    
## cluster_rent        1.000e+00  4.617e-16  2.166e+15  < 2e-16 ***
```

```
## Rent_Diff           1.000e+00  3.592e-16  2.784e+15  < 2e-16 ***
## util_index         -5.886e-16  5.952e-16 -9.890e-01  0.32279    
## class_a:util_index  -6.352e-16  2.957e-16 -2.148e+00  0.03176 *  
## class_a:cd_total_07  5.244e-17  1.180e-17  4.443e+00 8.99e-06 ***
## empl_gr:class_a     -2.467e-15  9.796e-16 -2.519e+00  0.01180 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.962e-13 on 7579 degrees of freedom
##   (73 observations deleted due to missingness)
## Multiple R-squared:      1,  Adjusted R-squared:      1 
## F-statistic: 7.666e+29 on 26 and 7579 DF,  p-value: < 2.2e-16
```

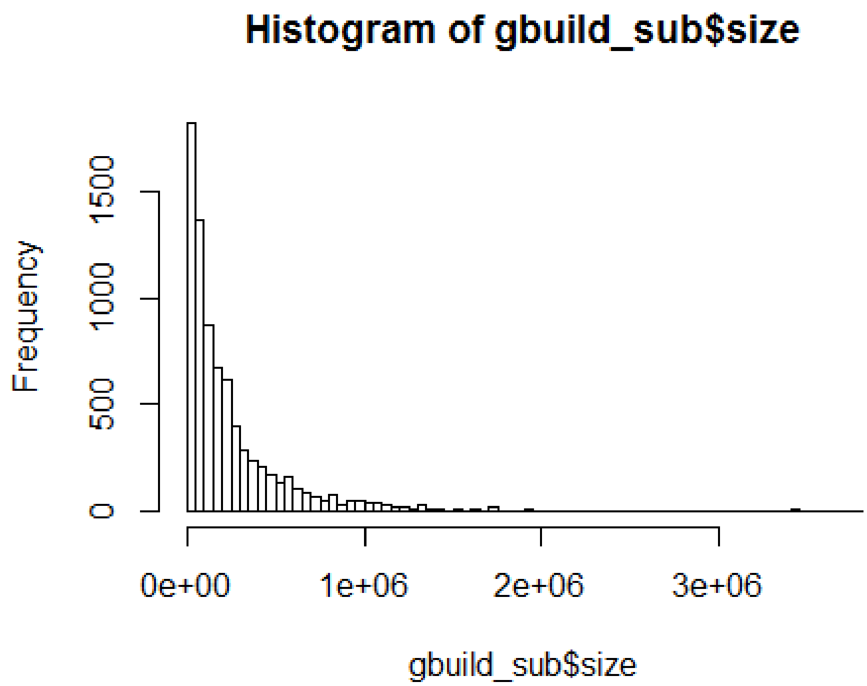
```
confint(lm.fit3)
```

```
##                             2.5 %        97.5 %
## (Intercept)          2.088064e-14  3.110011e-13
## CS_PropertyID        1.513610e-20  3.633968e-20
## cluster             -4.626273e-17 -1.071480e-17
## size                -7.674500e-20  5.118619e-21
## empl_gr             -1.793931e-15  1.152382e-15
## leasing_rate        -5.447144e-18  8.373433e-16
## stories             -7.551991e-16  1.254132e-15
## age                 -2.212730e-16  3.814592e-16
## renovated           -1.578538e-14  1.676003e-14
## class_a             -5.529029e-14  2.379531e-14
## class_b             -2.872016e-14  1.499860e-14
## LEED                -1.943457e-13  2.479583e-13
## Energystar          -1.724164e-13  2.988363e-13
## green_rating        -2.545419e-13  2.194381e-13
## net                 -5.818189e-14  1.575771e-14
## amenities           -7.827249e-15  2.382491e-14
## cd_total_07         -1.622360e-17  5.795607e-17
## hd_total07          -7.555322e-18  2.297868e-17
## Precipitation       -1.534598e-15  5.012440e-16
## Gas_Costs           -3.533754e-12  1.346727e-11
## Electricity_Costs   -2.656771e-12  1.876338e-12
## cluster_rent         1.000000e+00  1.000000e+00
## Rent_Diff            1.000000e+00  1.000000e+00
## util_index          -1.755330e-15  5.782299e-16
## class_a:util_index  -1.214864e-15 -5.546056e-17
## class_a:cd_total_07  2.930674e-17  7.558264e-17
## empl_gr:class_a     -4.387713e-15 -5.469624e-16
```

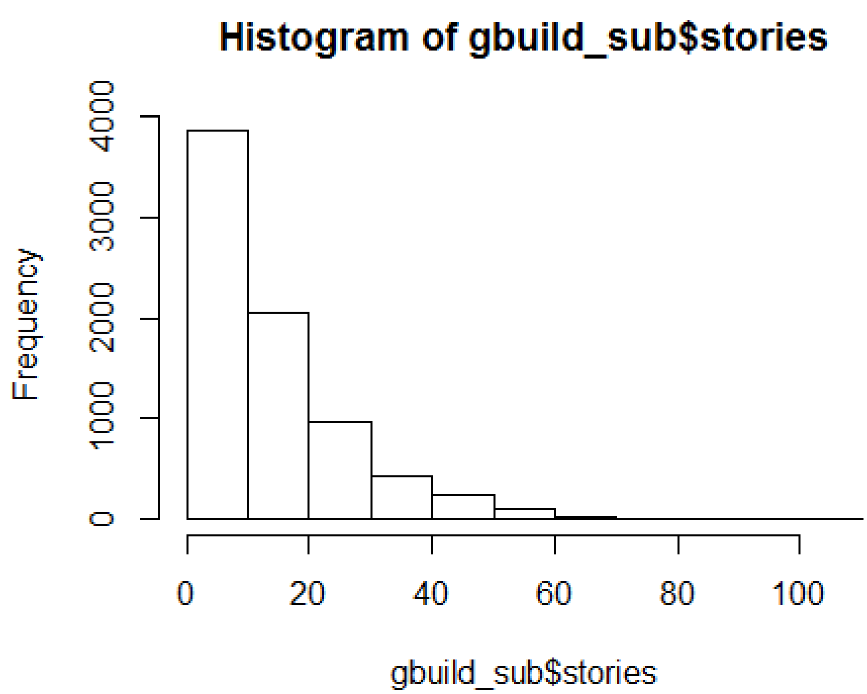
Surprisingly LEED remained in the model, however we fail to reject the null hypothesis that it is significant at the 95% level.

First we will begin by breaking the important continuous variables into manageable buckets. This will also serve us well to see the distribution of buildings across different ranges of values and setup our cross tab tables coming up. We will also limit the data to cities with positive employment growth, since Austin has one of the best economies in the country.

```
gbuild_sub$sizeCategory = cut(gbuild_sub$size, breaks = c(rep(0:20)*200000))
gbuild_sub$storiesCategory = cut(gbuild_sub$stories, breaks = c(rep(0:12)*10)
)
gbuild_sub$empl_grCategory = cut(gbuild_sub$empl_gr, breaks = c(rep(0:6)))
gbuild_sub$ageCategory = cut(gbuild_sub$age, breaks = c(rep(0:10)*20))
gbuild_sub$Electricity_CostsCategory = cut(gbuild_sub$Electricity_Costs,
                                           breaks = c(seq(0.00,0.07, by=0.01)
))
gbuild_sub$Gas_CostsCategory = cut(gbuild_sub$Gas_Costs,
                                           breaks = c(seq(0.00,0.03, by=0.005
)))
gbuild_sub$total_dd_07Category = cut(gbuild_sub$total_dd_07,
                                           breaks = c(seq(0.00,9000, by=2000)
))
gbuild_sub$cd_total_07Category = cut(gbuild_sub$cd_total_07,
                                           breaks = c(seq(0.00,6000, by=600))
)
attach(gbuild_sub)

## The following objects are masked from gbuild:
##
##     age, amenities, cd_total_07, class_a, class_b, cluster,
##     cluster_rent, CS_PropertyID, Electricity_Costs, empl_gr,
##     Energystar, Gas_Costs, green_rating, hd_total07, leasing_rate,
##     LEED, net, Precipitation, renovated, Rent, size, stories,
##     total_dd_07

hist(gbuild_sub$size, breaks=75) #Good dispersion in target range
```




```
hist(gbuild_sub$stories)   #Good dispersion in target range
```

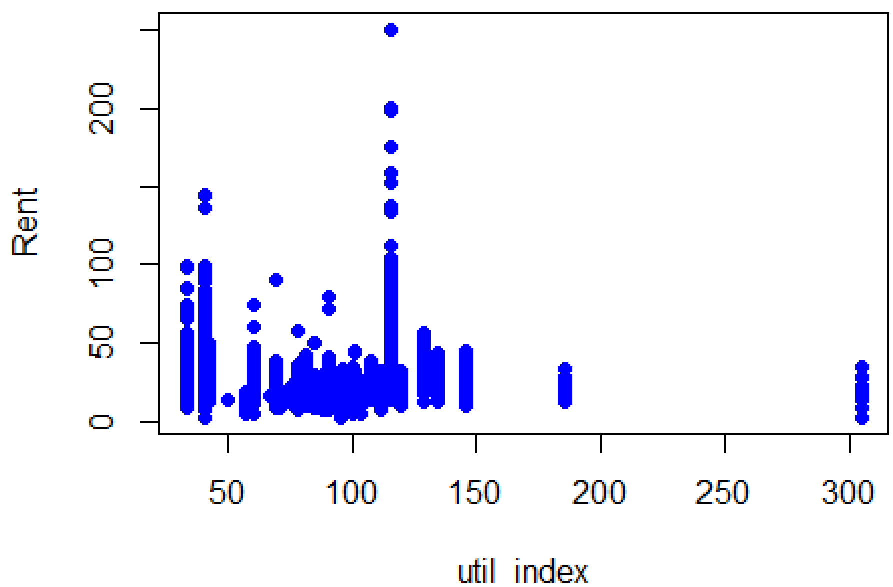


There appears to be a good distribution of buildings over the top predictors, so I'm not concerned about extrapolating outside of the observed ranges.

Next I inspect the theory that lower utility costs are the driver of higher rent prices in green buildings. In order to do this I create a feature called util_index, which is the sum of the products of gas costs and heating days and electric costs and cooling days. This feauture will allow us to measure the expense of HVAC in a single variable.

We begin by examing rents for green and non-green buildings as two different series across the full range of util_index:

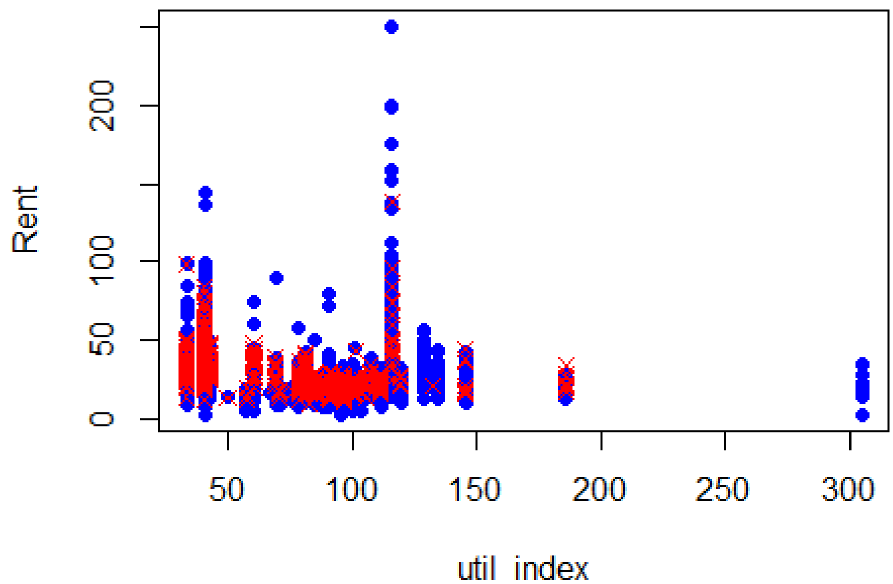
```
gbuild_sub$util_index = gbuild_sub$hd_total07*gbuild_sub$Gas_Costs +
    gbuild_sub$cd_total_07 * gbuild_sub$Electricity_Costs
#Bucket util_index
gbuild_sub$util_indexCategory = cut(gbuild_sub$util_index,
                                    breaks = c(seq(0.00,200, by=25)))
#Util index negatively correlated with rent; Higher utilities = lower rent
plot(Rent~util_index,data = gbuild_sub, col="blue",pch=16)
```




```
#Normalize util_index in case we need it
gbuild_sub$util_index_norm = gbuild_sub$util_index/mean(gbuild_sub$util_index
)

summary(gbuild_sub)

##   CS_PropertyID        cluster              size             empl_gr
##   Min.   :      1   Min.   :   1.0   Min.   :   2378   Min.   :-24.950
##   1st Qu.: 157426   1st Qu.: 272.0   1st Qu.:  52000   1st Qu.:  1.740
##   Median : 313238   Median : 479.0   Median : 132417   Median :  1.970
##   Mean   : 435335   Mean   : 590.1   Mean   : 239465   Mean   :  3.188
```

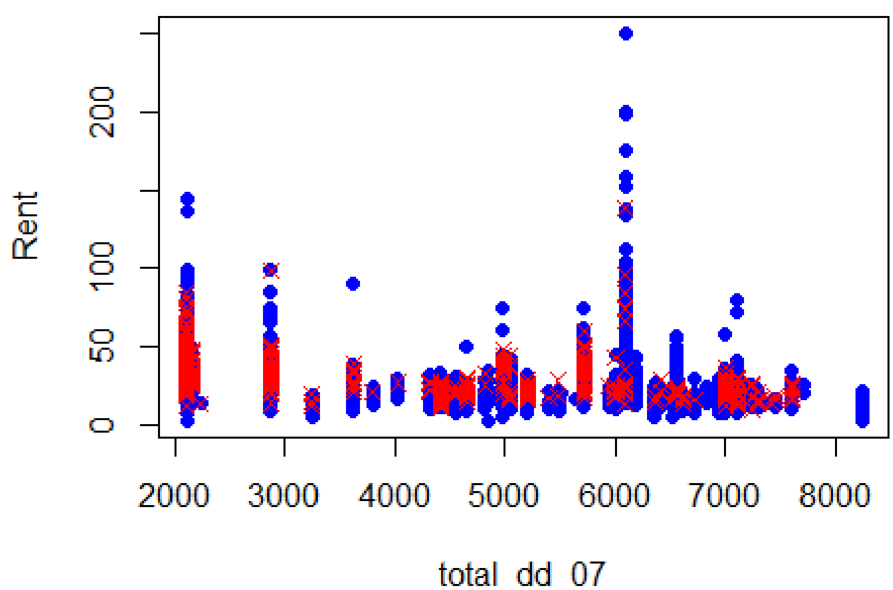
```
##  3rd Qu.: 440780   3rd Qu.:1044.0   3rd Qu.: 302375   3rd Qu.:  2.380
##  Max.   :6208103   Max.   :1230.0   Max.   :3781045   Max.   : 67.780
##                                                       NA's   :73
##       Rent         leasing_rate       stories            age
##  Min.   :  2.98   Min.   : 10.68   Min.   :  1.00   Min.   :  0.00
##  1st Qu.: 19.50   1st Qu.: 79.51   1st Qu.:  4.00   1st Qu.: 23.00
##  Median : 25.29   Median : 90.24   Median : 10.00   Median : 34.00
##  Mean   : 28.59   Mean   : 84.88   Mean   : 13.83   Mean   : 47.04
##  3rd Qu.: 34.20   3rd Qu.: 96.66   3rd Qu.: 20.00   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   :0.3814   Mean   :0.4083   Mean   :0.4587   Mean   :0.007032
##  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.   :0.000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :1.000
##  Mean   :0.08295   Mean   :0.08907   Mean   :0.03555   Mean   :0.538
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.000
##
##   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   :1217   Mean   :3440   Mean   :4657   Mean   :31.10
##  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     Rent_Diff
##  Min.   :0.009487   Min.   :0.01780   Min.   : 9.00   Min.   :-45.9150
##  1st Qu.:0.010296   1st Qu.:0.02330   1st Qu.:20.25   1st Qu.: -2.9100
##  Median :0.010296   Median :0.03274   Median :25.20   Median :  0.0000
##  Mean   :0.011329   Mean   :0.03095   Mean   :27.60   Mean   :  0.9903
##  3rd Qu.:0.011816   3rd Qu.:0.03781   3rd Qu.:34.15   3rd Qu.:  3.3300
##  Max.   :0.028914   Max.   :0.06280   Max.   :71.44   Max.   :191.2800
##
##    util_index              sizeCategory  storiesCategory empl_grCategory
##  Min.   : 33.12   (0,2e+05]       :4746   (0,10] :3867    (0,1]: 916
##  1st Qu.: 40.47   (2e+05,4e+05]  :1530   (10,20]:2049    (1,2]:2970
##  Median : 78.53   (4e+05,6e+05]  : 672   (20,30]: 963    (2,3]:2307
##  Mean   : 75.78   (6e+05,8e+05]  : 297   (30,40]: 420    (3,4]: 624
##  3rd Qu.: 96.17   (8e+05,1e+06]  : 192   (40,50]: 245    (4,5]: 434
##  Max.   :304.83   (1e+06,1.2e+06]: 117   (50,60]: 106    (5,6]:  98
```

```
##                    (Other)        : 125   (Other):  29    NA's : 330
##    ageCategory   Electricity_CostsCategory    Gas_CostsCategory
##  (20,40] :3183   (0,0.01]   :   0          (0,0.005]   :   0
##  (0,20]  :1287   (0.01,0.02]: 826          (0.005,0.01]: 554
##  (80,100]:1250   (0.02,0.03]:2917          (0.01,0.015]:6993
##  (40,60] : 886   (0.03,0.04]:3535          (0.015,0.02]:  45
##  (60,80] : 553   (0.04,0.05]: 314          (0.02,0.025]:   0
##  (Other) : 503   (0.05,0.06]:   0          (0.025,0.03]:  87
##  NA's    :  17   (0.06,0.07]:  87
##     total_dd_07Category        cd_total_07Category util_indexCategory
##  (0,2e+03]    :   0     (600,1.2e+03]    :3523     (25,50]  :2872
##  (2e+03,4e+03]:3044     (0,600]          :1667     (75,100] :2428
##  (4e+03,6e+03]:2087     (1.8e+03,2.4e+03]: 823     (100,125]:1406
##  (6e+03,8e+03]:2308     (1.2e+03,1.8e+03]: 695     (50,75]  : 626
##  NA's         : 240     (2.4e+03,3e+03]  : 424     (125,150]: 217
##                         (4.8e+03,5.4e+03]: 259     (Other)  :  43
##                         (Other)          : 288     NA's     :  87
##  util_index_norm
##  Min.   :0.4370
##  1st Qu.:0.5341
##  Median :1.0364
##  Mean   :1.0000
##  3rd Qu.:1.2692
##  Max.   :4.0228
##

g_green = subset(gbuild_sub, green_rating==1)
g_green = g_green[complete.cases(g_green),]
g_ngreen = subset(gbuild_sub, green_rating==0)
g_ngreen = g_ngreen[complete.cases(g_green),]

plot(Rent~util_index,data = g_ngreen, col="blue",pch=16)
points(Rent~util_index,data = g_green, col="red",pch=4)
```

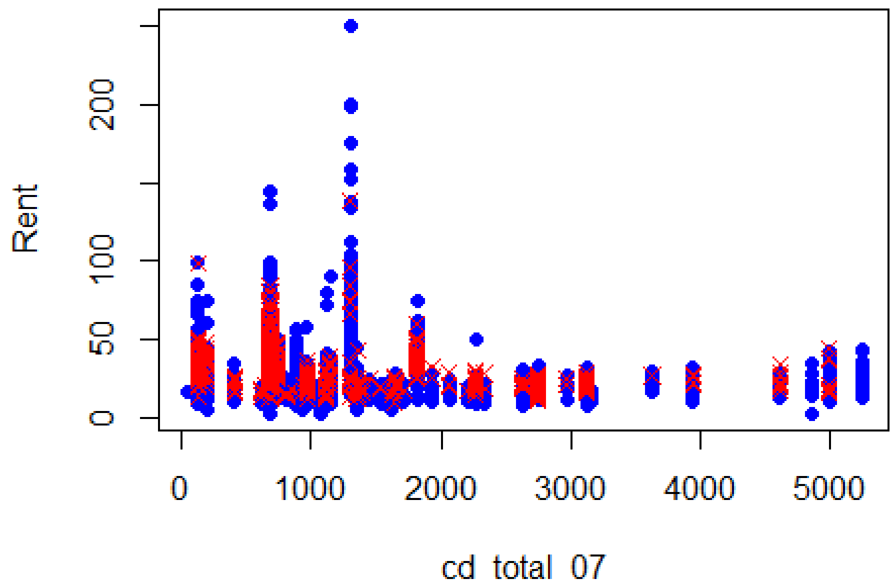


Well there isn't much to take from this graph. Let's try to do the same thing across total-dd_07-days:

```
plot(Rent~total_dd_07,data = g_ngreen, col="blue",pch=16)
points(Rent~total_dd_07,data = g_green, col="red",pch=4)
```



```r
plot(Rent~cd_total_07,data = g_ngreen, col="blue",pch=16)
points(Rent~cd_total_07,data = g_green, col="red",pch=4)
```



It appears green buildings are highly concentrated in milder climates. Lets look at a cross tab of green building frequencies by util_index and classs.

```r
freq =xtabs(~green_rating+class_a+util_indexCategory, data = gbuild_sub)
freq

## , , util_indexCategory = (0,25]
## 
##             class_a
## green_rating    0    1
##            0    0    0
##            1    0    0
## 
## , , util_indexCategory = (25,50]
## 
##             class_a
## green_rating    0    1
##            0 1624  951
##            1   70  227
## 
## , , util_indexCategory = (50,75]
## 
##             class_a
## green_rating    0    1
##            0  400  191
```

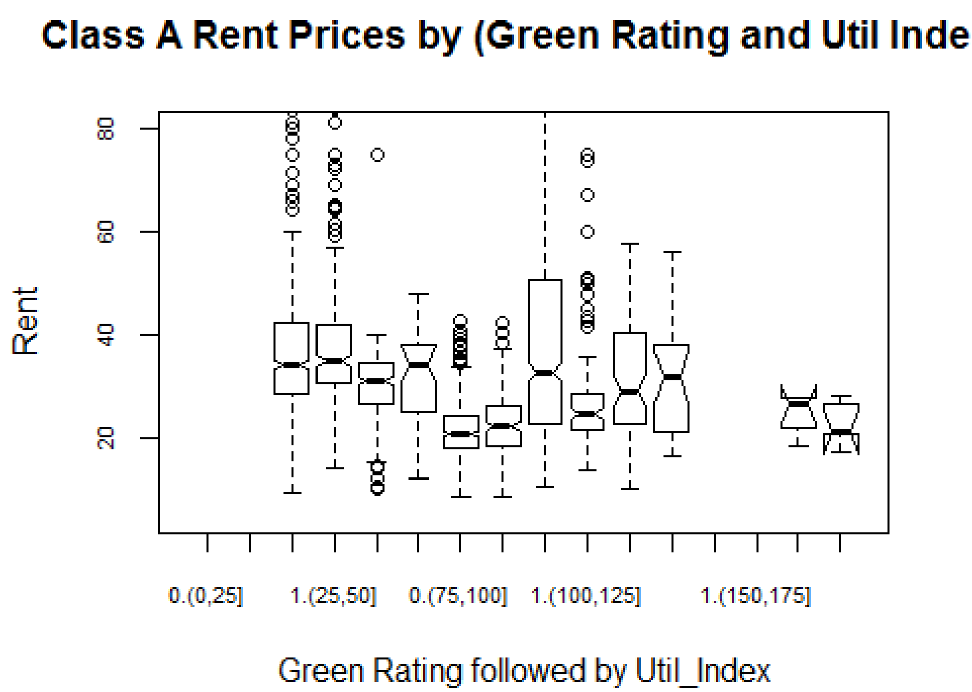
```
##            1    7   28
##
## , , util_indexCategory = (75,100]
##
##             class_a
## green_rating    0    1
##            0 1518  704
##            1   42  164
##
## , , util_indexCategory = (100,125]
##
##             class_a
## green_rating    0    1
##            0  714  588
##            1   12   92
##
## , , util_indexCategory = (125,150]
##
##             class_a
## green_rating    0    1
##            0   94   94
##            1    1   28
##
## , , util_indexCategory = (150,175]
##
##             class_a
## green_rating    0    1
##            0    0    0
##            1    0    0
##
## , , util_indexCategory = (175,200]
##
##             class_a
## green_rating    0    1
##            0   28    6
##            1    4    5
```

The table above seems to indicate green buildings are highly concentrated in class a buildings. That would be a good reason why they appear to rent for more money. The table shows 3 splits of the util_index: 0-75,75-150,150-225

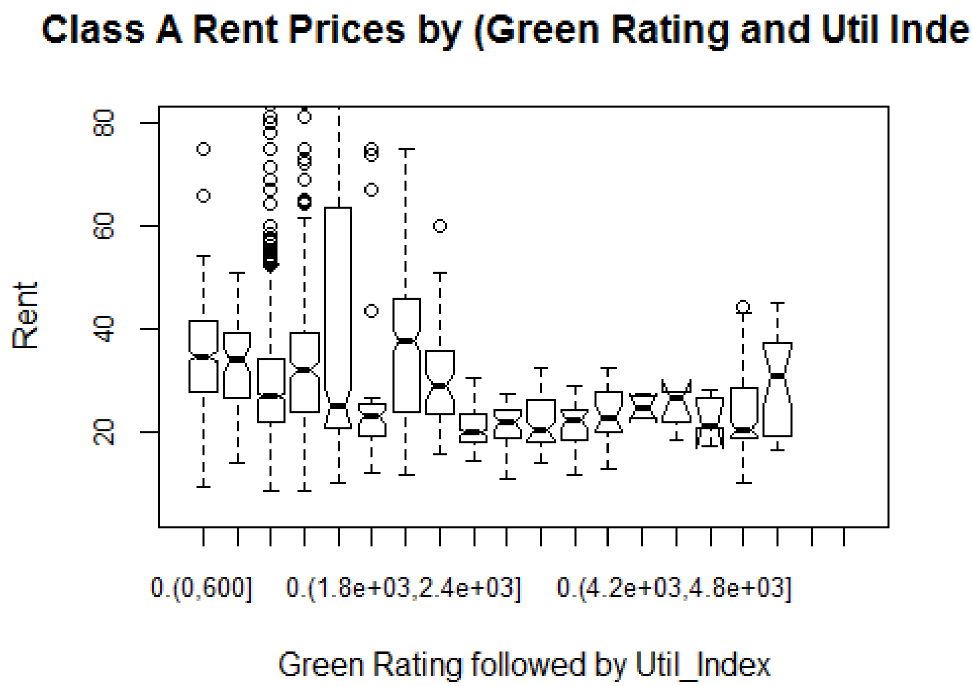
For those respective bins, 255 of the 332 green buildings or 76.8%, are found in class a buildings. 284/339 or 83.8% of green buildings are class a in the second bin. And 5/9 of the green buildings in the last bin are class A. It seems we've found something here. What if the higher rent prices for green buildings were a reflection of the class of the building instead of the green rating?

Lets dig deeper, and look at a few boxplots of rent by green rating for only class A buildings & non-class A buildings

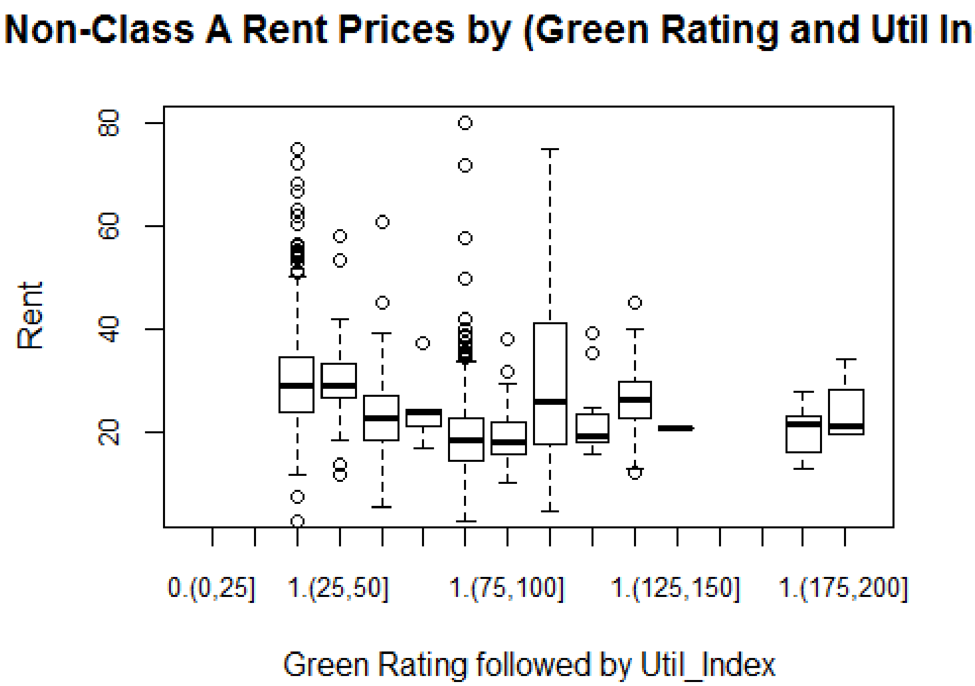
```
gbuild_sub_A = subset(gbuild_sub, class_a == 1)
gbuild_sub_NotA = subset(gbuild_sub, class_a == 0)
boxplot(Rent ~ green_rating+util_indexCategory, data = gbuild_sub_A,
        xlab= "Green Rating followed by Util_Index",
        ylab="Rent",notch=TRUE,ylim=c(5,80),cex.axis=.7)

## Warning in bxp(structure(list(stats = structure(c(NA, NA, NA, NA, NA, NA,
:
## some notches went outside hinges ('box'): maybe set notch=FALSE

        title("Class A Rent Prices by (Green Rating and Util Index)")
```




```
boxplot(Rent ~ green_rating+cd_total_07Category, data = gbuild_sub_A,
        xlab= "Green Rating followed by Util_Index",
        ylab="Rent",notch=TRUE,ylim=c(5,80),cex.axis=.8)

## Warning in bxp(structure(list(stats = structure(c(9.6, 28.2, 34.84, 41.5,
:
## some notches went outside hinges ('box'): maybe set notch=FALSE

        title("Class A Rent Prices by (Green Rating and Util Index)")
```




```
#boxplot(Rent ~ green_rating+total_dd_07Category, data = gbuild_sub_A, xlab=
"Green Rating followed by total degree days", ylab="Rent",notch=TRUE,ylim=c(5
,80),cex.axis=.8)
boxplot(Rent ~ green_rating+util_indexCategory,
        data = gbuild_sub_NotA, xlab= "Green Rating followed by Util_Index",y
lab="Rent",
        notch=FALSE,ylim=c(5,80),cex.axis=.8)
        title("Non-Class A Rent Prices by (Green Rating and Util Index)")
```




```
g_control1 = subset(gbuild_sub_A, net==1 & leasing_rate <= 80 & empl_gr > 0)
freq =xtabs(~green_rating+empl_grCategory+util_indexCategory, data = g_contro
l1)
freq

## , , util_indexCategory = (0,25]
##
##             empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0     0     0     0     0     0     0
##            1     0     0     0     0     0     0
##
## , , util_indexCategory = (25,50]
##
##             empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0     0     3     3     0     0     0
##            1     0     2     0     0     0     0
##
## , , util_indexCategory = (50,75]
##
##             empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0     0     0     0     0     0     0
##            1     0     0     0     0     0     0
##
## , , util_indexCategory = (75,100]
```

```
##
##              empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0     0     6     0     0     0     2
##            1     0     0     1     1     0     0
##
## , , util_indexCategory = (100,125]
##
##              empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0     1     0     0     6     0     0
##            1     0     0     0     2     0     0
##
## , , util_indexCategory = (125,150]
##
##              empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0     0     0     7     0     0     0
##            1     0     0     1     0     0     0
##
## , , util_indexCategory = (150,175]
##
##              empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0     0     0     0     0     0     0
##            1     0     0     0     0     0     0
##
## , , util_indexCategory = (175,200]
##
##              empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0     0     0     0     0     0     0
##            1     0     0     0     0     0     0
```

```
rent_sum =xtabs(Rent~green_rating+empl_grCategory+util_indexCategory, data =
g_control1)
avg_rent = rent_sum/freq
avg_rent
```

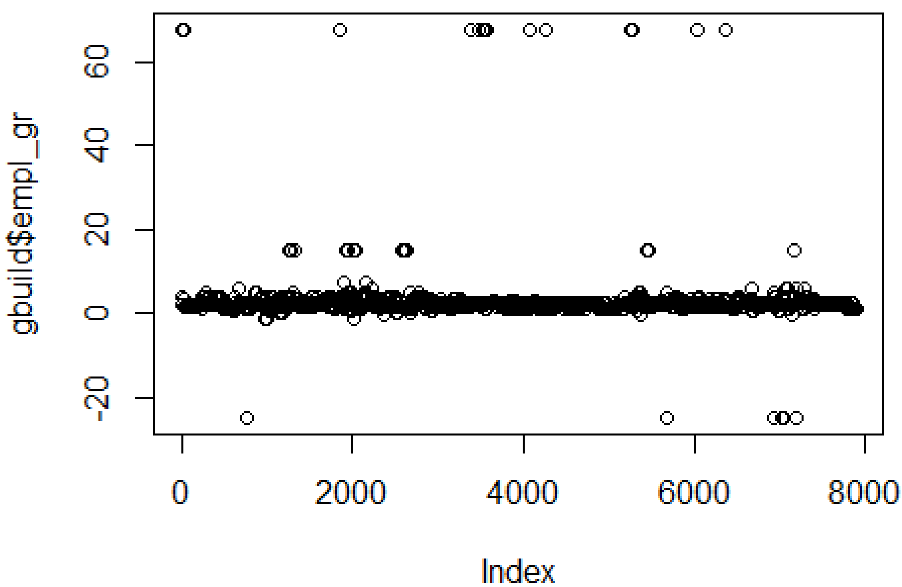
```
## , , util_indexCategory = (0,25]
##
##              empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0
##            1
##
## , , util_indexCategory = (25,50]
##
##              empl_grCategory
## green_rating (0,1]    (1,2]    (2,3] (3,4] (4,5] (5,6]
```

```
##           0       28.33333 33.51333
##           1       30.31500
##
## , , util_indexCategory = (50,75]
##
##             empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##           0
##           1
##
## , , util_indexCategory = (75,100]
##
##             empl_grCategory
## green_rating (0,1]    (1,2]    (2,3]    (3,4] (4,5]    (5,6]
##           0       23.50000                         25.00000
##           1                14.34000 19.25000
##
## , , util_indexCategory = (100,125]
##
##             empl_grCategory
## green_rating    (0,1] (1,2] (2,3]    (3,4] (4,5] (5,6]
##           0 23.65000             18.77167
##           1                      20.92000
##
## , , util_indexCategory = (125,150]
##
##             empl_grCategory
## green_rating (0,1] (1,2]    (2,3] (3,4] (4,5] (5,6]
##           0             17.84429
##           1             18.61000
##
## , , util_indexCategory = (150,175]
##
##             empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##           0
##           1
##
## , , util_indexCategory = (175,200]
##
##             empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##           0
##           1
```

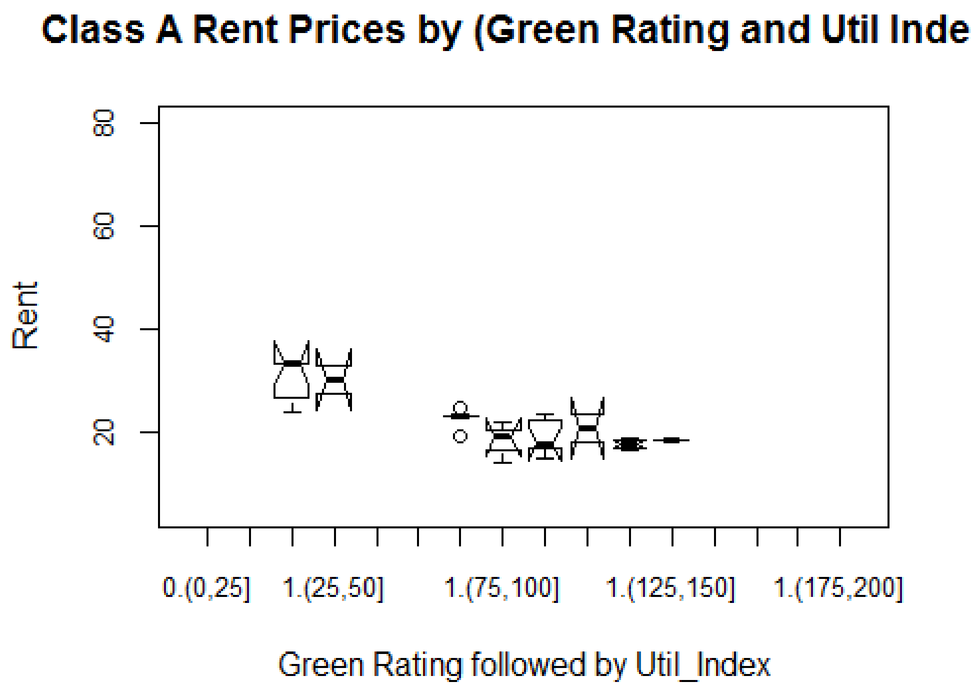
At this point we can see green buildings are highly correlated and that some green buildings in certain utility_index bins do rent for a premium. The last table is particularly interesting. Here we can see that green buildings generally only sell for a premium in modest to high growth cities.

Net pricing wasn't indicated as a strong predictor but let's do the same excercise controlling for non-net leases and some other features correlated with price.

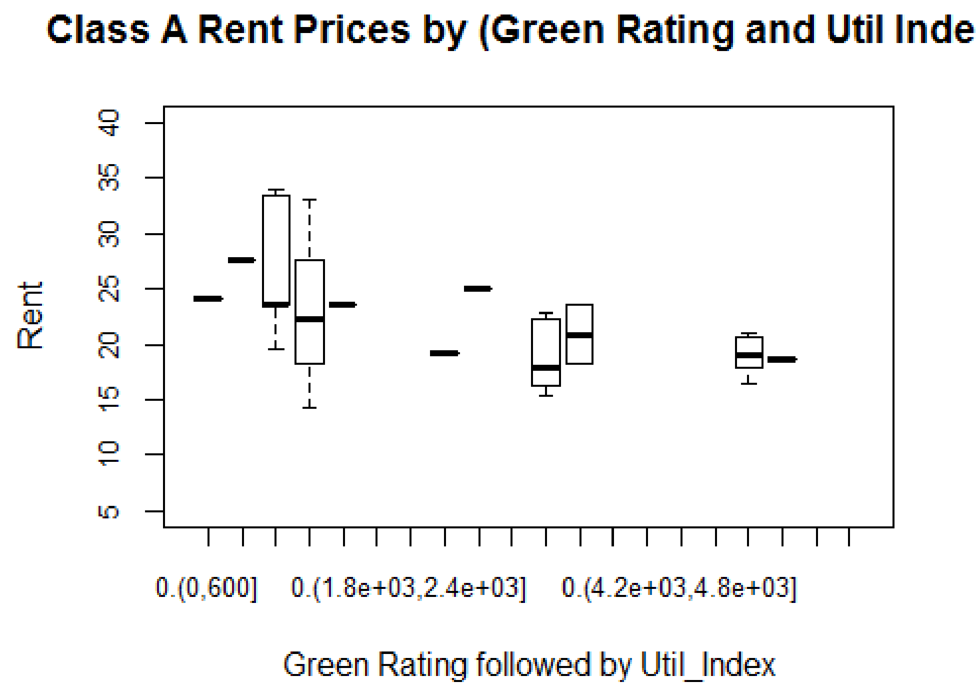
```
#Control for important lurking variables
g_control1 = subset(gbuild_sub_A, net==1 & leasing_rate <= 80 & empl_gr > 0)
g_control2 = subset(gbuild_sub_A, net==1 & leasing_rate <= 80 & empl_gr <= 0)

plot(gbuild$empl_gr)
```



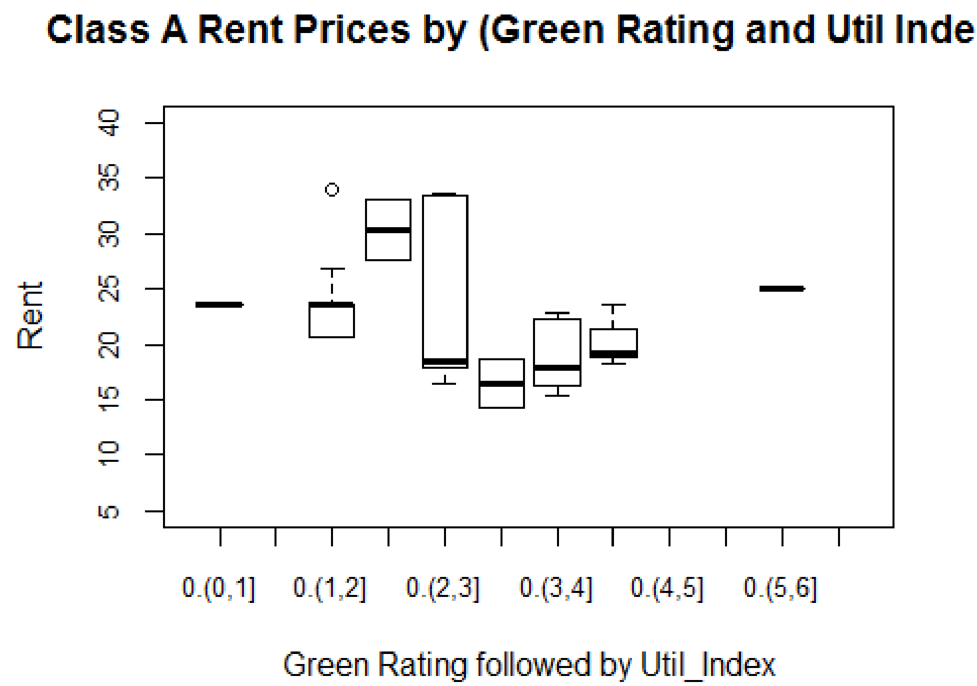

```
boxplot(Rent ~ green_rating+util_indexCategory, data = g_control1,
        xlab= "Green Rating followed by Util_Index",
        ylab="Rent",notch=TRUE,ylim=c(5,80),cex.axis=.8)

## Warning in bxp(structure(list(stats = structure(c(NA, NA, NA, NA, NA, NA,
:
## some notches went outside hinges ('box'): maybe set notch=FALSE

        title("Class A Rent Prices by (Green Rating and Util Index)")
```



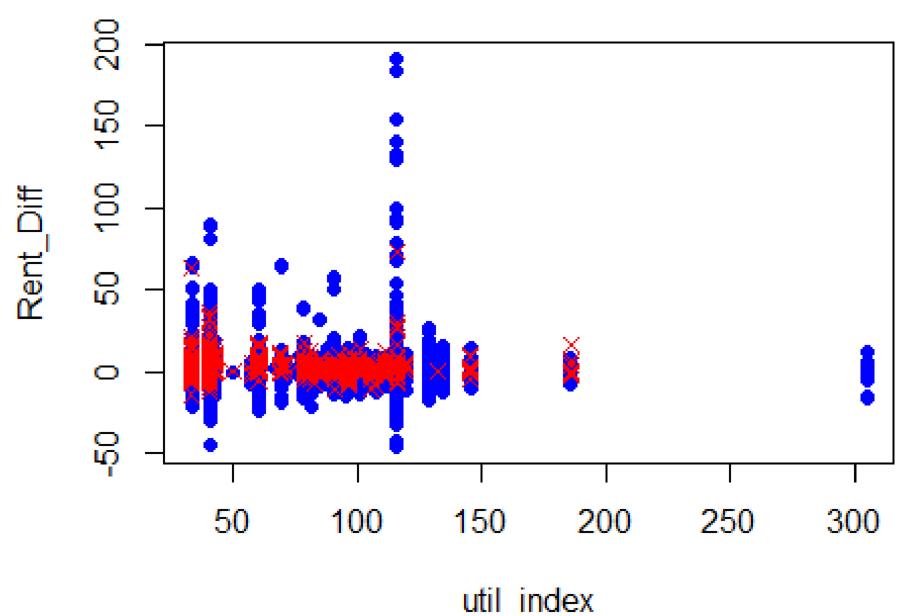

```r
boxplot(Rent ~ green_rating+cd_total_07Category, data = g_control1,
        xlab= "Green Rating followed by Util_Index",
        ylab="Rent",ylim=c(5,40),cex.axis=.8)
        title("Class A Rent Prices by (Green Rating and Util Index)")
```




```
boxplot(Rent ~ green_rating+empl_grCategory, data = g_control1,
        xlab= "Green Rating followed by Util_Index",
        ylab="Rent",ylim=c(5,40),cex.axis=.8)
        title("Class A Rent Prices by (Green Rating and Util Index)")
```




```r
plot(Rent_Diff~util_index,data = g_ngreen, col="blue",pch=16)
points(Rent_Diff~util_index,data = g_green, col="red",pch=4)
```



```
freq =xtabs(~green_rating+class_a+util_indexCategory, data = gbuild_sub)
rent_sum =xtabs(Rent~green_rating+class_a+util_indexCategory, data = gbuild_s
ub)
avg_rent = rent_sum/freq
avg_rent
```

```
## , , util_indexCategory = (0,25]
##
##              class_a
## green_rating 0 1
##            0
##            1
##
## , , util_indexCategory = (25,50]
##
##              class_a
## green_rating        0        1
##            0 30.76389 36.94904
##            1 31.60100 37.39678
##
## , , util_indexCategory = (50,75]
##
##              class_a
## green_rating        0        1
##            0 23.24820 31.24958
##            1 24.31571 31.23036
##
## , , util_indexCategory = (75,100]
##
##              class_a
## green_rating        0        1
##            0 19.39635 21.75456
##            1 19.52500 22.79463
##
## , , util_indexCategory = (100,125]
##
##              class_a
## green_rating        0        1
##            0 31.18513 40.77844
##            1 22.53083 30.21543
##
## , , util_indexCategory = (125,150]
##
##              class_a
## green_rating        0        1
##            0 26.40766 31.10670
##            1 20.90000 31.30321
##
## , , util_indexCategory = (150,175]
##
```

```
##              class_a
## green_rating 0 1
##            0
##            1
##
## , , util_indexCategory = (175,200]
##
##              class_a
## green_rating        0        1
##            0 20.44107 25.05500
##            1 24.12000 23.05200
```

```
gbuild_notnet = subset(gbuild_sub,net==0)
gbuild_net = subset(gbuild_sub,net==1)
```

```
g_cntrl_notnet = subset(gbuild_notnet, class_a == 1 | class_b == 1 & age <= 3
0 &
                        (empl_gr >= 1 & empl_gr <= 3) &
                        (stories >= 5 & stories <= 25) &
                        (size <= 300000 & size >= 200000))
```

```
freq =xtabs(~green_rating+empl_grCategory+class_a, data = gbuild_sub)
freq
```

```
## , , class_a = 0
##
##              empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0   491  1925  1166   331   261    52
##            1    20    39    46     9    13     1
##
## , , class_a = 1
##
##              empl_grCategory
## green_rating (0,1] (1,2] (2,3] (3,4] (4,5] (5,6]
##            0   359   876   877   214   119    35
##            1    46   130   218    70    41    10
```

```
rent_sum =xtabs(Rent~green_rating+empl_grCategory+class_a, data = gbuild_sub)
avg_rent = rent_sum/freq
avg_rent
```

```
## , , class_a = 0
##
##              empl_grCategory
## green_rating    (0,1]    (1,2]    (2,3]    (3,4]    (4,5]    (5,6]
##            0 31.16255 27.85325 25.59782 19.44486 17.98046 18.21058
##            1 26.38250 29.19846 29.17696 19.60889 18.52154 10.51000
##
## , , class_a = 1
##
```

```
##              empl_grCategory
## green_rating    (0,1]    (1,2]    (2,3]    (3,4]    (4,5]    (5,6]
##            0 35.87744 36.09933 32.41568 23.99907 20.45521 21.53857
##            1 32.96804 31.20923 35.11009 24.44957 21.04268 22.44500
```

```
freq =xtabs(~green_rating+total_dd_07Category, data = gbuild_sub)
rent_sum =xtabs(Rent~green_rating+total_dd_07Category, data = gbuild_sub)
avg_rent = rent_sum/freq
avg_rent
```

```
##              total_dd_07Category
## green_rating (0,2e+03] (2e+03,4e+03] (4e+03,6e+03] (6e+03,8e+03]
##            0                32.41796      27.41713      26.10320
##            1                35.60505      25.70507      25.92176
```

```
freq =xtabs(~green_rating+net+util_indexCategory, data = gbuild_sub)
rent_sum =xtabs(Rent~green_rating+net+util_indexCategory, data = gbuild_sub)
avg_rent = rent_sum/freq
avg_rent
```

```
## , , util_indexCategory = (0,25]
##
##              net
## green_rating 0 1
##            0
##            1
##
## , , util_indexCategory = (25,50]
##
##              net
## green_rating        0        1
##            0 33.18523 26.26608
##            1 36.10089 31.93600
##
## , , util_indexCategory = (50,75]
##
##              net
## green_rating        0        1
##            0 26.02386 17.39692
##            1 30.32531 24.75000
##
## , , util_indexCategory = (75,100]
##
##              net
## green_rating        0        1
##            0 20.16530 19.48324
##            1 22.21375 20.95214
##
## , , util_indexCategory = (100,125]
##
##              net
```

```
## green_rating        0        1
##           0 35.73954 31.22422
##           1 29.80478 26.26857
##
## , , util_indexCategory = (125,150]
##
##             net
## green_rating        0        1
##           0 29.23134 19.32667
##           1 31.82778 19.02000
##
## , , util_indexCategory = (150,175]
##
##             net
## green_rating 0 1
##           0
##           1
##
## , , util_indexCategory = (175,200]
##
##             net
## green_rating        0        1
##           0 21.14091 25.03000
##           1 23.52667
```

```
freq
```

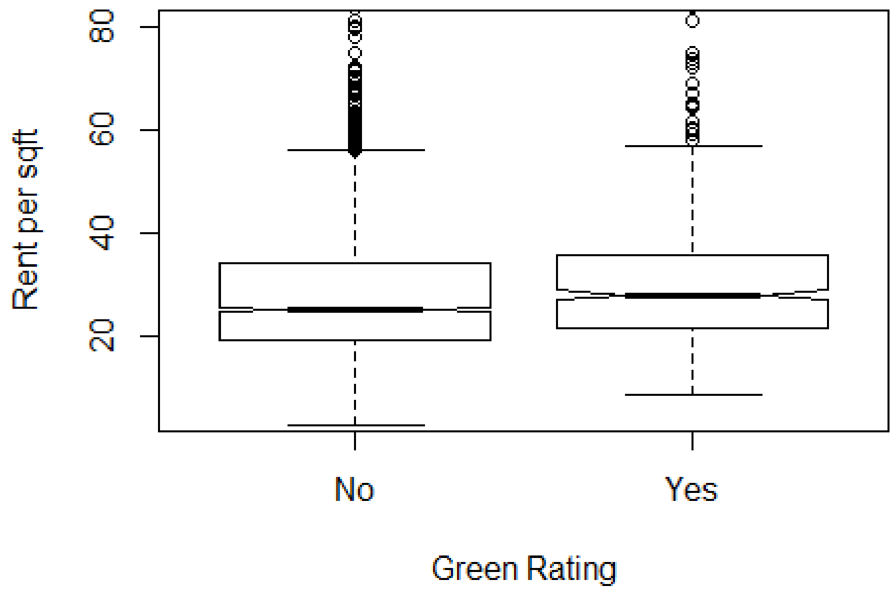
```
## , , util_indexCategory = (0,25]
##
##             net
## green_rating    0    1
##           0    0    0
##           1    0    0
##
## , , util_indexCategory = (25,50]
##
##             net
## green_rating    0    1
##           0 2524   51
##           1  292    5
##
## , , util_indexCategory = (50,75]
##
##             net
## green_rating    0    1
##           0  578   13
##           1   32    3
##
## , , util_indexCategory = (75,100]
##
```

```
##              net
## green_rating    0    1
##            0 2151   71
##            1  192   14
##
## , , util_indexCategory = (100,125]
##
##              net
## green_rating    0    1
##            0 1238   64
##            1   90   14
##
## , , util_indexCategory = (125,150]
##
##              net
## green_rating    0    1
##            0  179    9
##            1   27    2
##
## , , util_indexCategory = (150,175]
##
##              net
## green_rating    0    1
##            0    0    0
##            1    0    0
##
## , , util_indexCategory = (175,200]
##
##              net
## green_rating    0    1
##            0   33    1
##            1    9    0
```

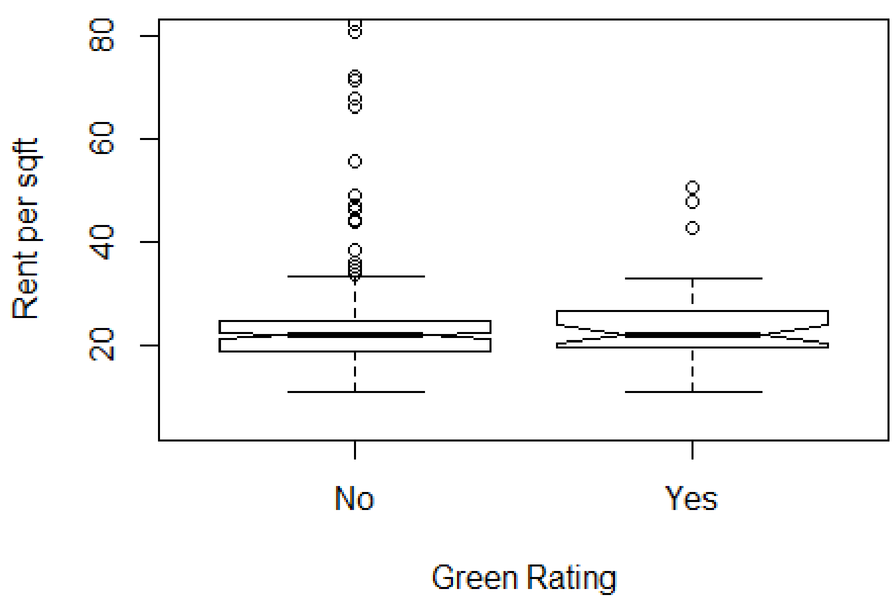
plot(cluster_rent~util_index, data = subset(gbuild_sub,green_rating==1),col="blue")
points(cluster_rent~util_index, data = subset(gbuild_sub,green_rating==0),col="red")

plot(Rent_norm~util_index, data = subset(gbuild_sub,green_rating==1),col="blue",pch=16)
points(Rent_norm~util_index, data = subset(gbuild_sub,green_rating==0),col="red", pch=4)

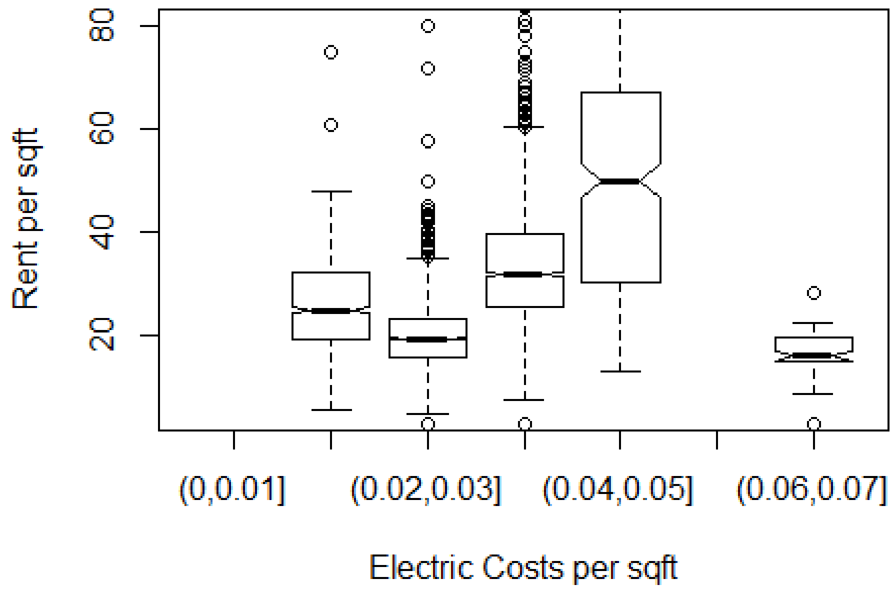
```

Now lets examine rent by utility costs, after we control for some features. Let's assume the building will be class A, with median employment growth (2), roughly 250,000 sqft, and less than 10 years old.

```
boxplot(Rent ~ green_rating, data = gbuild_notnet, names= c("No","Yes"),title
="Non-Net Leases", xlab= "Green Rating", ylab="Rent per sqft",notch=TRUE, yli
m=c(5,80))
```
```



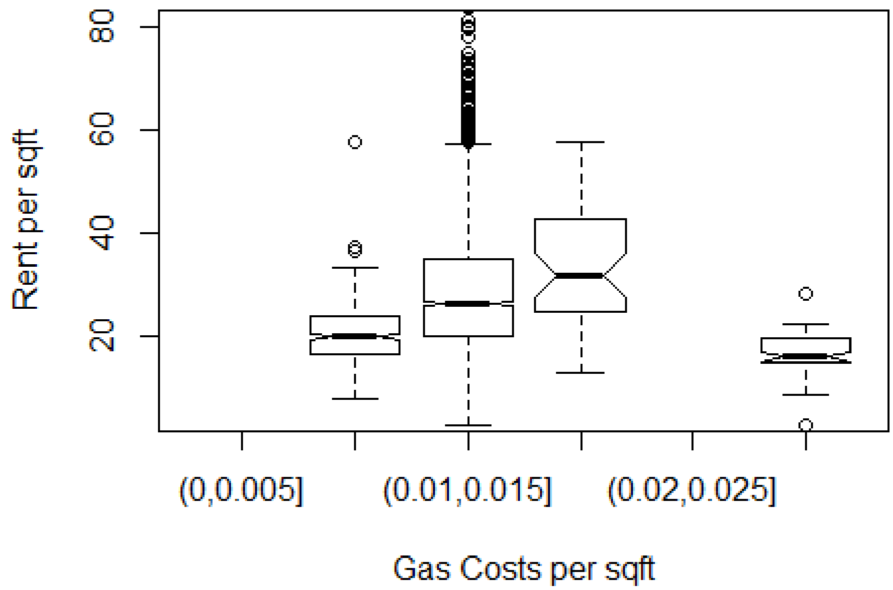

```
boxplot(Rent ~ green_rating, data = gbuild_net, names= c("No","Yes"),title="Net Leases", xlab= "Green Rating", ylab="Rent per sqft",notch=TRUE, ylim=c(5,80))
```



```
boxplot(Rent ~ Electricity_CostsCategory, data = gbuild_notnet, xlab= "Electr
ic Costs per sqft", ylab="Rent per sqft",notch=TRUE,ylim=c(5,80))
```




```
boxplot(Rent ~ Gas_CostsCategory, data = gbuild_notnet, xlab= "Gas Costs per
sqft", ylab="Rent per sqft",notch=TRUE,ylim=c(5,80))
```




```
xtabs(~green_rating+net,data = gbuild_sub)

##               net
## green_rating    0    1
##             0 6761  234
##             1  645   39
```

Conclusion -

In conclusion, we have shown how risky and unreliable the former analysts recommendations were. By not using regression to control for the other features the recommendations was wreckless. We have attempted to isolate the effect of the green buildings outside of regression. While we weren't able to isolate the effect of green buildings completely, we believe its quite evident its highly correlated with other features that drive up rent, such as class a buildings and high employment growth cities. At a minimum we have shown that green buildings effect is not consistent throughout the data set, and that its unwise to generalize.

Bootstrapping :

The value at risk and returns of each portfolio gives us a measure of how "safe" or "risky" an asset is.

```
suppressMessages(library(mosaic))
suppressMessages(library(fImport))
suppressMessages(library(foreach))
```

```
mystocks = c("SPY","TLT","LQD","EEM","VNQ")
myprices = yahooSeries(mystocks, from='2010-01-01', to='2016-07-30')


# A helper function for calculating percent returns from a Yahoo Series
YahooPricesToReturns = function(series) {
    mycols = grep('Adj.Close', colnames(series))
    closingprice = series[,mycols]
    N = nrow(closingprice)
    percentreturn = as.data.frame(closingprice[2:N,]) / as.data.frame(closing
price[1:(N-1),]) - 1
    mynames = strsplit(colnames(percentreturn), '.', fixed=TRUE)
    mynames = lapply(mynames, function(x) return(paste0(x[1], ".PctReturn")))
    colnames(percentreturn) = mynames
    as.matrix(na.omit(percentreturn))
}

myreturns = YahooPricesToReturns(myprices)
```

marshals appropriate evidence to characterize the risk/return properties of the five major asset classes listed above.

We will use bootstrap sampling to calculate the returns for each asset. The code below does bootstrapping for SPY alone. Similarly we implement the code for all assets.

```
sim_SPY = foreach(i=1:5000, .combine='rbind') %do% {
  totalwealth = 100000
  n_days = 20
  weights_even = c(1.0, 0.0, 0.0, 0.0, 0.0)
  holdings = weights_even * totalwealth
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return.today = resample(myreturns, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return.today
    totalwealth = sum(holdings)
    wealthtracker[today] = totalwealth
    holdings = weights_even * totalwealth
  }
  wealthtracker
}
```

The average returns for SPY over 20 days is

```
mean(sim_SPY[,n_days])

## [1] 101076.9
```

5% value at risk for SPY is :

```
quantile(sim_SPY[,n_days], 0.05) - 100000
```

```
##        5%
## -6183.525
```

outlines your choice of the "safe" and "aggressive" portfolios.

We derived a table like the one below to identify the assets as safe and aggresive based on their loss at risk and average returns.

uses bootstrap resampling to estimate the 4-week (20 trading day) value at risk of each of your three portfolios at the 5% level

Even split portfolio :

```
sim_even = foreach(i=1:500, .combine='rbind') %do% {
  totalwealth = 100000
  n_days = 20
  weights_even = c(0.2, 0.2, 0.2, 0.2, 0.2)
  holdings = weights_even * totalwealth
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return.today = resample(myreturns, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return.today
    totalwealth = sum(holdings)
    wealthtracker[today] = totalwealth
    holdings = weights_even * totalwealth
  }
  wealthtracker
}
```

Average return for even split portfolio is

```
return_even <- mean(sim_even[,n_days])
```

5% value at risk for even split portfolio

```
risk_even <- quantile(sim_even[,n_days], 0.05) - 100000
```

Safe portfolio - The safe portfolio will use the safest assets - SPY, TLT and LQD (at least 3 classes required)The safe assets are those that have low risk. We are choosing to invest about 80% of our wealth into SPY because SPY has the highest returns among the three and has medium to low risk :

```
sim_safe = foreach(i=1:500, .combine='rbind') %do% {
  totalwealth = 100000
  n_days = 20
  weights_even = c(0.8, 0.1, 0.1, 0.0, 0.0)
  holdings = weights_even * totalwealth
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return.today = resample(myreturns, 1, orig.ids=FALSE)
```

```
    holdings = holdings + holdings*return.today
    totalwealth = sum(holdings)
    wealthtracker[today] = totalwealth
    holdings = weights_even * totalwealth
  }
  wealthtracker
}
```

Average return for safe split portfolio is :

```
return_safe <- mean(sim_safe[,n_days])
```

5% value at risk for safe split portfolio :

```
risk_safe <- quantile(sim_safe[,n_days], 0.05) - 100000
```

Aggresive portfolio : In our 'Aggressive portfolio', we have chosen the assets that give the highest returns irrespective of the risk involved. EEM, VNQ are the two assets that gave us the highest returns. So, our aggressive portfolio includes EEM and VNQ. We are choosing to invest in EEM and VNQ in the ratio 3:7 because VNQ offers higher returns than EEM and we want to maximize our returns.

```
sim_high = foreach(i=1:500, .combine='rbind') %do% {
  totalwealth = 100000
  n_days = 20
  weights_even = c(0.0, 0.0, 0.0, 0.3, 0.7)
  holdings = weights_even * totalwealth
  wealthtracker = rep(0, n_days)
  for(today in 1:n_days) {
    return.today = resample(myreturns, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return.today
    totalwealth = sum(holdings)
    wealthtracker[today] = totalwealth
    holdings = weights_even * totalwealth
  }
  wealthtracker
}
```

Average return for aggressive portfolio is

```
return_aggressive <- mean(sim_high[,n_days])
```

5% value at risk for aggresive portfolio :

```
risk_aggressive <- quantile(sim_high[,n_days], 0.05) - 100000
```

compares the results for each portfolio in a way that would allow the reader to make an intelligent decision among the three options.

Conclusion

Average returns over a 20 day period for the three portfolios :

Even :

```
return_even

## [1] 100769.3
```

Safe :

```
return_safe

## [1] 101402
```

Aggressive :

```
return_aggressive

## [1] 101438.9
```

Loss at risk for the three portfolios :

Even :

```
risk_even

##        5%
## -4040.989
```

Safe :

```
risk_safe

##        5%
## -3946.606
```

Aggressive :

```
risk_aggressive

##        5%
## -7142.911
```

So, from the above estimations of risk and returns, if an investor is willing to be aggressive, then he stands to gain a lot in the returns and his loss at risk is also the highest among the three portfolios.

The safe portfolio does not yield higher returns than even portfolio and the loss at risk is also higher for safe portfolio as compared to the loss at risk value for even portfolio.

So, it is more beneficial to invest in an even portfolio.

Problem 3 :

Market segmentation

Inital Set-up and Loading the Data:

```
# Change to required path


library(flexclust)

## Loading required package: grid

## Loading required package: modeltools

## Loading required package: stats4

##
## Attaching package: 'modeltools'

## The following object is masked from 'package:RCurl':
##
##     clone

library(ggplot2)
library(reshape2)
library(corrplot)
library(corrgram)


mkt_seg = read.csv("C:/MSBA/James Scott Statistics/STA380-master/STA380-maste
r/data/social_marketing.csv",header=T)
str(mkt_seg)

## 'data.frame':    7882 obs. of  37 variables:
##  $ X              : Factor w/ 7882 levels "123pxkyqj","12grikctu",..: 372
0 2540 4096 596 3197 3609 4749 6518 7418 4917 ...
##  $ chatter         : int  2 3 6 1 5 6 1 5 6 5 ...
##  $ current_events  : int  0 3 3 5 2 4 2 3 2 2 ...
##  $ travel          : int  2 2 4 2 0 2 7 3 0 4 ...
##  $ photo_sharing   : int  2 1 3 2 6 7 1 6 1 4 ...
##  $ uncategorized   : int  2 1 1 0 1 0 0 1 0 0 ...
##  $ tv_film         : int  1 1 5 1 0 1 1 1 0 5 ...
##  $ sports_fandom   : int  1 4 0 0 0 1 1 1 0 9 ...
##  $ politics        : int  0 1 2 1 2 0 11 0 0 1 ...
##  $ food            : int  4 2 1 0 0 2 1 0 2 5 ...
##  $ family          : int  1 2 1 1 1 1 0 0 2 4 ...
##  $ home_and_garden : int  2 1 1 0 0 1 0 0 1 0 ...
##  $ music           : int  0 0 1 0 0 1 0 2 1 1 ...
##  $ news            : int  0 0 1 0 0 0 1 0 0 0 ...
##  $ online_gaming   : int  0 0 0 0 3 0 0 1 2 1 ...
##  $ shopping        : int  1 0 2 0 2 5 1 3 0 0 ...
##  $ health_nutrition: int  17 0 0 0 0 0 1 1 22 7 ...
```

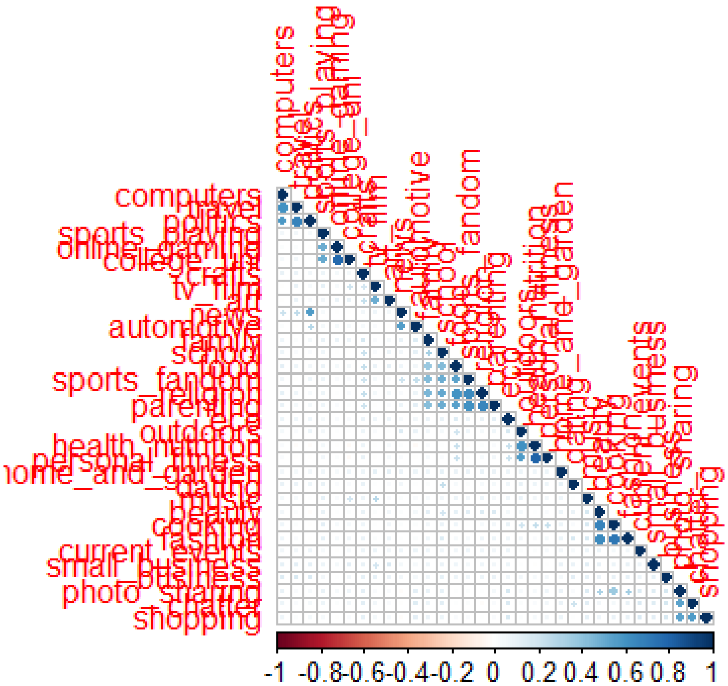
```
##  $ college_uni     : int  0 0 0 1 4 0 1 0 1 4 ...
##  $ sports_playing  : int  2 1 0 0 0 0 1 0 0 1 ...
##  $ cooking         : int  5 0 2 0 1 0 1 10 5 4 ...
##  $ eco             : int  1 0 1 0 0 0 0 0 2 1 ...
##  $ computers       : int  1 0 0 0 1 1 1 1 1 2 ...
##  $ business        : int  0 1 0 1 0 1 3 0 1 0 ...
##  $ outdoors        : int  2 0 0 0 1 0 1 0 3 0 ...
##  $ crafts          : int  1 2 2 3 0 0 0 1 0 0 ...
##  $ automotive      : int  0 0 0 0 0 1 0 1 0 4 ...
##  $ art             : int  0 0 8 2 0 0 1 0 1 0 ...
##  $ religion        : int  1 0 0 0 0 0 1 0 0 13 ...
##  $ beauty          : int  0 0 1 1 0 0 0 5 5 1 ...
##  $ parenting       : int  1 0 0 0 0 0 0 1 0 3 ...
##  $ dating          : int  1 1 1 0 0 0 0 0 0 0 ...
##  $ school          : int  0 4 0 0 0 0 0 0 1 3 ...
##  $ personal_fitness: int  11 0 0 0 0 0 0 0 12 2 ...
##  $ fashion         : int  0 0 1 0 0 0 0 4 3 1 ...
##  $ small_business  : int  0 0 0 0 1 0 0 0 1 0 ...
##  $ spam            : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ adult           : int  0 0 0 0 0 0 0 0 0 0 ...
```

From looking at the various columns in the dataset, we decided to drop the columns spam and adult since they do not give us real insights into user preferences. In addition, we also combined the columns chatter and uncategorized into one since they represent the tweets that dont fit into any category.

```
mkt_seg_junk = mkt_seg[,-c(36,37)]
mkt_seg_junk$chatter = mkt_seg_junk$uncategorized + mkt_seg_junk$chatter
mkt_seg_junk = mkt_seg_junk[,-6] # Removing uncategorized

# Without the id column
mkt_seg_no_id = mkt_seg_junk[,-1]
```

To see if any of the variables are related, we plotted correlations. corrplot was used since it allows for easier and cleaner visualization of relationships.

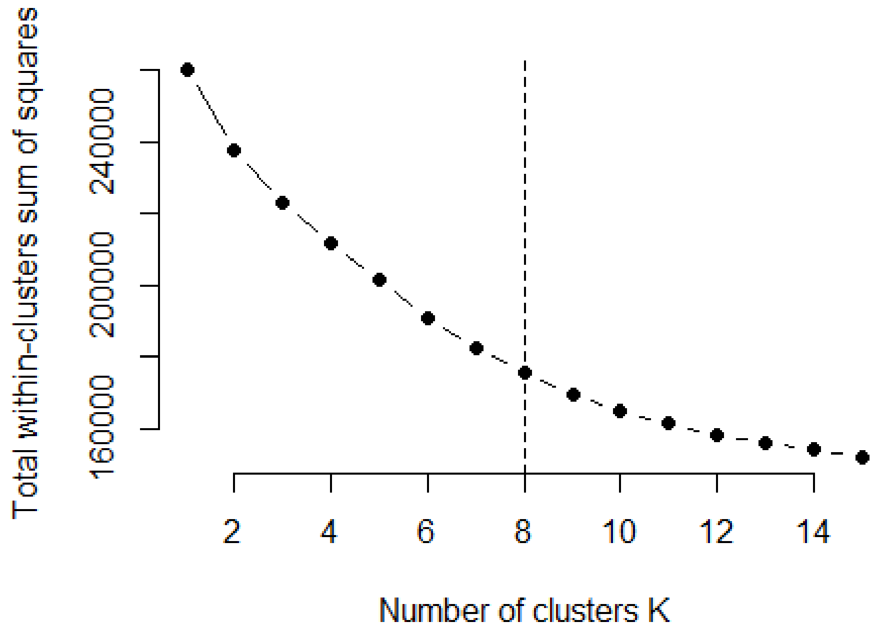
```
# Looking at correlations between variables
corr_matrix = cor(mkt_seg_no_id)
corrplot(corr_matrix, type="lower", order="hclust")
```



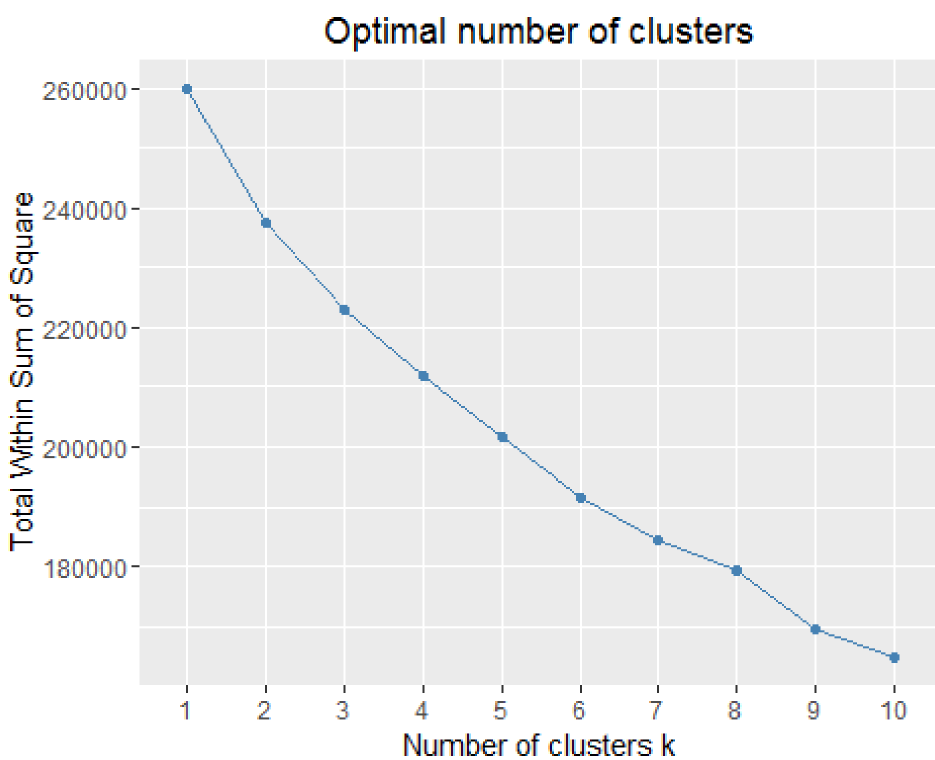
From the corrplot, it seems like there are likely to be about 4-8 clusters.

To find the optimal number of clusters, we implemented the Elbow method for k means clustering after scaling and centering the data.

```r
library(factoextra)
library(cluster)
library(NbClust)

# scaling before clustering
mkt_seg_scale <- scale(mkt_seg_no_id, center=TRUE, scale=TRUE)

set.seed(5)
# Calculating wss till k=15
k.max <- 15
data <- mkt_seg_scale
wss <- sapply(1:k.max,
        function(k){kmeans(data, k, nstart=10 )$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")
abline(v = 8, lty =2)
```

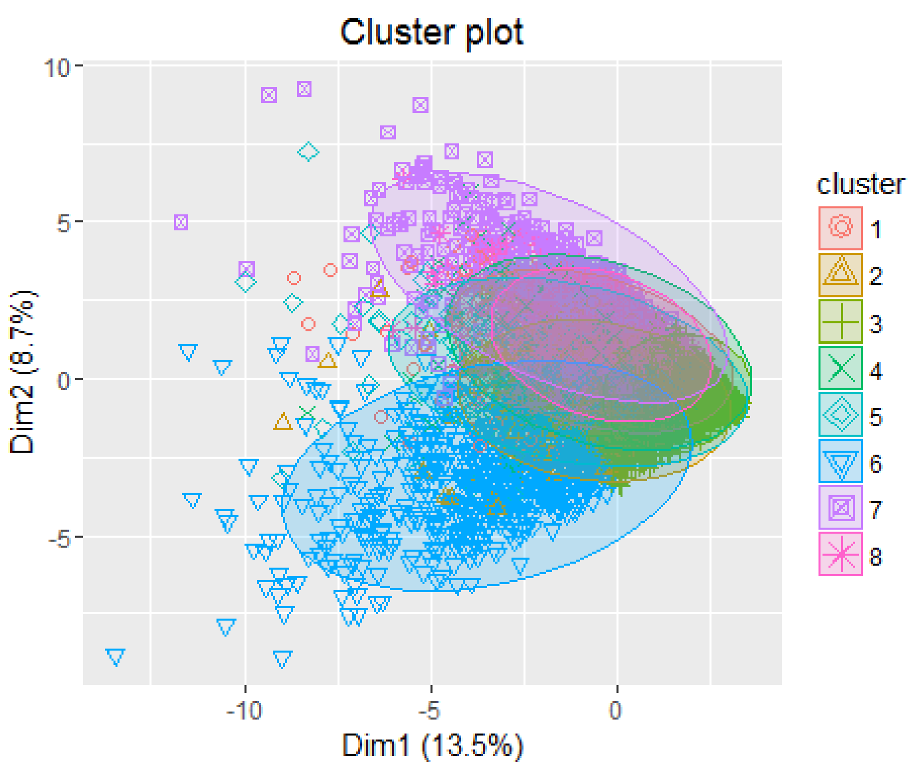



```r
# Cross checking with factoextra package
fviz_nbclust(data, kmeans, method = "wss")
```



From the plots, the optimal number of clusters is 8. We chose 8 since it minimises wss to an acceptable value and will not have too many clusters that will it hard to interpret. We will use k means clustering with 8 clusters.

```
# K-means clustering
set.seed(10)
km_seg <- kmeans(mkt_seg_scale, 8, nstart = 30)

# k-means group number of each observation
clust_obs <- km_seg$cluster
table(clust_obs)

## clust_obs
##    1    2    3    4    5    6    7    8
##  794  440 3480  371  364  698  504 1231

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




```
# Identifying where the centers of the clusters are
clusters_cent = km_seg$centers
imp_fact = t(clusters_cent)

# Separating by cluster and only taking important features
cluster_1 = imp_fact[which(abs(imp_fact[,1])>=0.4),1]
cluster_2 = imp_fact[which(abs(imp_fact[,2])>=0.4),2]
```

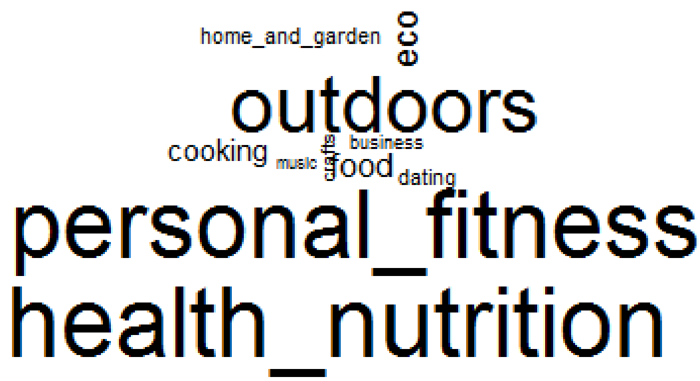
```
cluster_3 = imp_fact[which(abs(imp_fact[,3])>=0.4),3]
names(cluster_3) = c("photo_sharing") # since only 1 variable
cluster_4 = imp_fact[which(abs(imp_fact[,4])>=0.4),4]
cluster_5 = imp_fact[which(abs(imp_fact[,5])>=0.4),5]
cluster_6 = imp_fact[which(abs(imp_fact[,6])>=0.4),6]
cluster_7 = imp_fact[which(abs(imp_fact[,7])>=0.4),7]
cluster_8 = imp_fact[which(abs(imp_fact[,8])>=0.4),8]

# Seeing how the clusters turned out
cluster_1

##             food health_nutrition          cooking              eco
##        0.4577940        2.2002089        0.4025928        0.5419657
##         outdoors personal_fitness
##        1.7114549        2.1673465

cluster_2

## sports_fandom      politics          news    automotive
##     0.6580063     1.2176757     2.6381607     2.5800077

cluster_3

## photo_sharing
##    -0.4056307

cluster_4

##  online_gaming    college_uni sports_playing
##       3.497844       3.267583       2.149292

cluster_5

##         travel       politics           news      computers       business
##      3.2262928      3.0806307      1.1301224      2.8876695      0.5512909
## small_business
##      0.4158731

cluster_6

## sports_fandom          food        family        crafts      religion
##     2.0833636     1.8417506     1.5028191     0.7294984     2.2793038
##     parenting        school
##     2.1526532     1.6869280

cluster_7

## photo_sharing         music       cooking        beauty       fashion
##     1.2267360     0.5271313     2.8124735     2.5708935     2.6659507

cluster_8
```

```
##        chatter  photo_sharing        tv_film          music       shopping
##      1.2243274      0.8780005      0.5222805      0.4020824      1.1005971
##       business            art small_business
##      0.4458287      0.4056572      0.4204943
```

From the above results, we can drop cluster 3 since it has only 1 category and cluster 7 already has similar features.

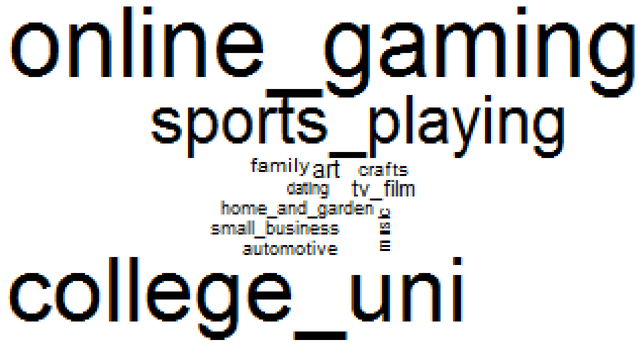
Now, plotting important features of each cluster in a wordcloud.

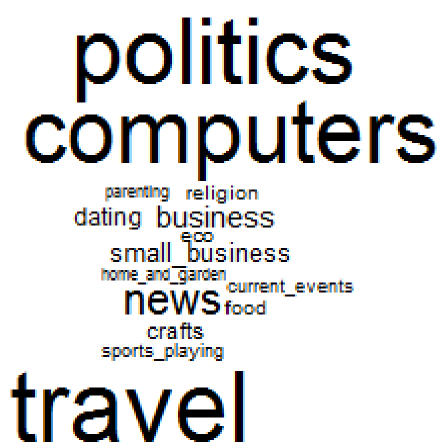
```
par(mfrow=c(1,1))
library(wordcloud)

## Loading required package: RColorBrewer

for (i in c(1,2,4,5,6,7,8)) { # skipping cluster 3
wordcloud(colnames(mkt_seg_scale), km_seg$centers[i,], min.freq=0, max.words=
100, scale=c(3,.5))
}
```

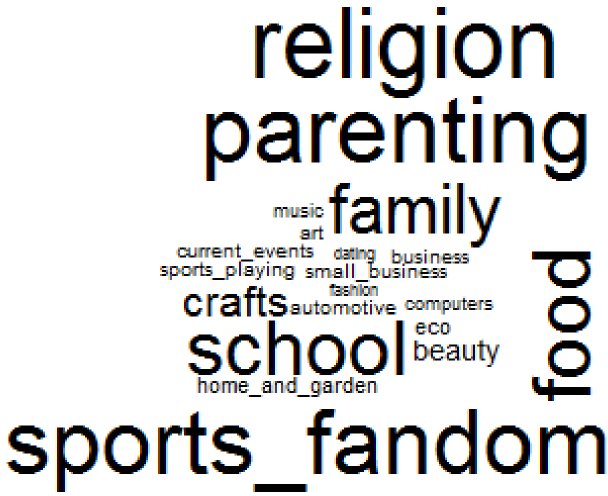


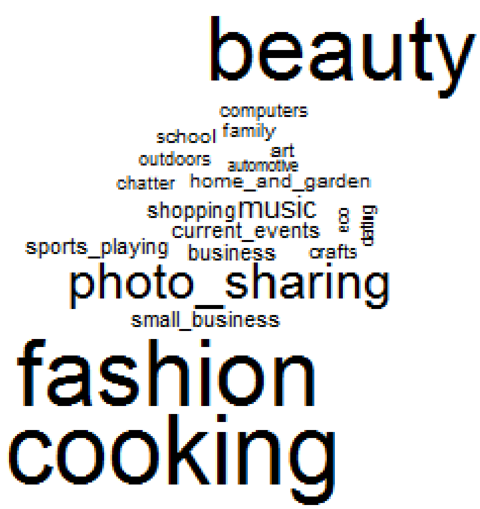
home_and_garden
eco
outdoors
cooking
music
crafts
business
food
dating
personal_fitness
health_nutrition

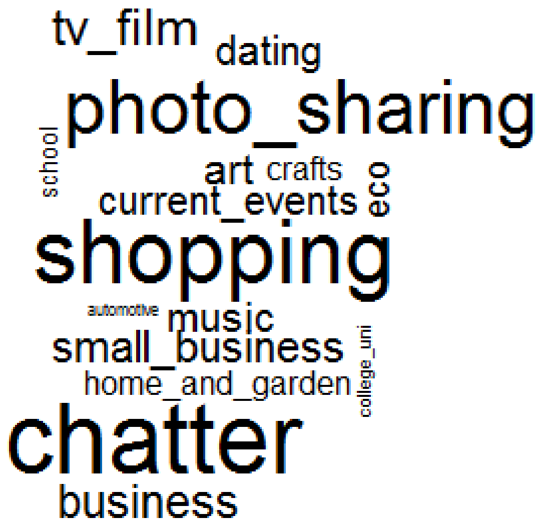
automotive
politics
family
home_and_garden
tv_film
news
outdoors
current_events
school
parenting
sports_fandom


online_gaming
sports_playing
family
art
crafts
dating
tv_film
home_and_garden
small_business
music
automotive
college_uni


politics
computers
parenting
religion
dating
business
eco
small_business
home_and_garden
current_events
news
food
crafts
sports_playing
travel


religion
parenting
music
family
art
current_events
dating
business
sports_playing
small_business
fashion
crafts
automotive
computers
food
eco
school
beauty
home_and_garden
sports_fandom


beauty
computers
school
family
art
outdoors
automotive
chatter
home_and_garden
shopping
music
eco
dating
current_events
sports_playing
business
crafts
photo_sharing
small_business
fashion
cooking



From the clusters obtained, the market segments obtained are the following:

Cluster 1 - Health conscious users
Cluster 2 - Users with more (stereotypical) masculine interests
Cluster 3 - Youngsters (Cluster 3 was dropped and numbers of all others were changed accordingly)
Cluster 4 - Businessmen/Business women
Cluster 5 - Family oriented users
Cluster 6 - Users with more (stereotypical) feminine interests
Cluster 7 - Miscellaneous

These market segments are valuable to NutrientH20 because they now have a better understanding of their customer base by getting a fair idea of what age groups their customers are in, what phase of life they are going through and their hobbies/interests. They can tune their messaging strategy to have customized messages and promotions going out to people based on these interests.