Question 2

```r
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: lattice

## Loading required package: ggplot2

## 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(ggplot2)
library(foreach)
```

```r
greenbuildings=read.csv('greenbuildings.csv')
View(greenbuildings)
attach(greenbuildings)
detach(greenbuildings)

names(greenbuildings)

##  [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"

#### 15 stories + green + over 250,000 ####
mask=greenbuildings$stories>=15 #keeps buildings with exactly 15 stories
fifteen_stories=greenbuildings[mask,]
str(fifteen_stories)

## 'data.frame':    2614 obs. of  23 variables:
##  $ CS_PropertyID    : int  379285 234578 42087 233989 234263 234298 233940
233941 431225 224553 ...
##  $ cluster          : int  1 6 6 6 6 6 6 6 8 8 ...
##  $ size             : int  174307 225895 912011 518578 255305 254920 74595
6 746824 409889 723922 ...
##  $ empl_gr          : num  2.22 4.01 4.01 4.01 4.01 ...
##  $ Rent             : num  40.7 14.8 17 17 18 ...
##  $ leasing_rate     : num  96.6 91 99.3 93.5 95.7 ...
##  $ stories          : int  16 15 31 21 15 15 31 31 20 40 ...
##  $ age              : int  5 24 34 36 25 26 28 29 6 34 ...
##  $ renovated        : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ class_a          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ class_b          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LEED             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Energystar       : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ green_rating     : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ net              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities        : int  1 1 1 1 1 1 1 1 0 0 ...
##  $ cd_total_07      : int  4988 2746 2746 2746 2746 2746 2746 2746 5240 52
40 ...
##  $ hd_total07       : int  58 1670 1670 1670 1670 1670 1670 1670 956 956 .
..
##  $ total_dd_07      : int  5046 4416 4416 4416 4416 4416 4416 4416 6196 61
96 ...
##  $ Precipitation    : num  42.6 25.6 25.6 25.6 25.6 ...
##  $ Gas_Costs        : num  0.0137 0.0101 0.0101 0.0101 0.0101 ...
##  $ Electricity_Costs: num  0.029 0.0289 0.0289 0.0289 0.0289 ...
##  $ cluster_rent     : num  36.8 17.5 17.5 17.5 17.5 ...
```

```
mask2=fifteen_stories$green_rating==1
fifteen_green=fifteen_stories[mask2,]#buildings with 15 stories and green rat
ings
str(fifteen_green)
```

```
## 'data.frame':    273 obs. of  23 variables:
##  $ CS_PropertyID    : int  431225 204299 437486 755727 320838 48101 246750
479467 1029816 86081 ...
##  $ cluster          : int  8 11 13 14 16 22 25 26 28 29 ...
##  $ size             : int  409889 525422 378538 841498 550101 465363 49080
3 1117000 413895 388325 ...
##  $ empl_gr          : num  67.78 1.74 3.27 1.74 1.97 ...
##  $ Rent             : num  30.5 25 26.6 24.5 29 ...
##  $ leasing_rate     : num  97.1 71.1 95.5 99.5 87.8 ...
##  $ stories          : int  20 16 17 40 43 27 20 60 25 22 ...
##  $ age              : int  6 23 22 2 24 19 25 15 22 22 ...
##  $ renovated        : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ class_a          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ class_b          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LEED             : int  0 0 0 1 0 0 0 1 0 0 ...
##  $ Energystar       : int  1 1 1 0 1 1 1 0 1 1 ...
##  $ green_rating     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ net              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ amenities        : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ cd_total_07      : int  5240 1113 2269 1113 130 130 684 1929 1073 3939
...
##  $ hd_total07       : int  956 6001 2382 6001 2739 2739 1419 2891 7171 376
...
##  $ total_dd_07      : int  6196 7114 4651 7114 2869 2869 2103 4820 8244 43
15 ...
##  $ Precipitation    : num  10.5 41.3 40.7 41.3 22.7 ...
##  $ Gas_Costs        : num  0.012 0.0108 0.0138 0.0108 0.0103 0.0103 0.0103
0.0139 0.0102 0.0137 ...
##  $ Electricity_Costs: num  0.0235 0.0233 0.0229 0.0233 0.0378 0.0378 0.037
8 0.021 0.0206 0.029 ...
##  $ cluster_rent     : num  25.5 22 25.1 23.3 34 ...
```

```
mask5=fifteen_green$size>=250000
fifteen_green_size=fifteen_green[mask5,] #15 + green + greater than 250,000


#### less than 15, not green, less than 250,000
mask3=greenbuildings$stories<15
nofifteen=greenbuildings[mask3,]

mask4=nofifteen$green_rating<1
nofifteen_green=nofifteen[mask4,] #less than 15 + no green buildings


mask7=nofifteen_green$size<250000
```

```
nofifteen_green=nofifteen_green[mask7,]#less than 15, no green, less than 250
,000

median(fifteen_green_size$Rent)#35.71 -- 25.25 #if it equaled to 15 exactly

## [1] 25.25

median(nofifteen_green$Rent)#25

## [1] 25
```

A new data frame was created consisting of the buildings that had only green ratings, had 15 or more stories, and were 250,000 square feet or more. In order to gain a better understanding about whether the investment would be worth it, we'd needed to narrow down the information to certain buildings that had similar features. The excel Guru only took into account green vs. not green. After taking the median, it seems as though the Excel guru is overestimating. The median of green buildings are actually 25.2 when we take all these variables into consideration. However, if we were to take the buildings with exactly 15 (not equal to or more), we'd see that the median rises to 35.71, resulting in over 2 million dollars of revenue. There seems to be a large amount of green buildings bringing down the median price in rent. Although this result in revenue seems favorable, there are only 6 of these buildings in the sample. Once again emphasizing the stats Guru over generalization and simplification of the data.

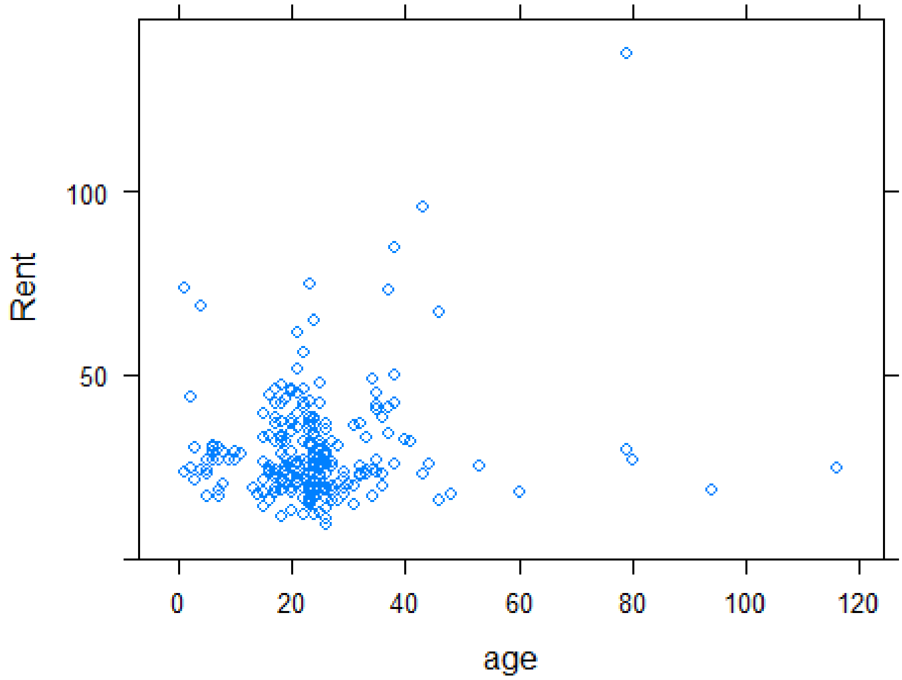
```
names(greenbuildings)

##  [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"

lm.fit=lm(Rent~.,data=fifteen_green_size)
summary(lm.fit) #regression model

##
## Call:
## lm(formula = Rent ~ ., data = fifteen_green_size)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.209  -4.085  -0.218   3.203  44.415
##
## Coefficients: (2 not defined because of singularities)
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -2.288e+01  9.942e+00  -2.301  0.02231 *
## CS_PropertyID       4.278e-07  1.129e-06   0.379  0.70514
## cluster             8.468e-04  1.153e-03   0.734  0.46346
```

```
## size                 3.303e-06  3.545e-06   0.932  0.35251
## empl_gr              9.836e-02  7.037e-02   1.398  0.16360
## leasing_rate         8.000e-02  4.903e-02   1.632  0.10418
## stories             -4.741e-02  8.560e-02  -0.554  0.58025
## age                  9.777e-02  4.476e-02   2.184  0.02998 *
## renovated           -1.496e+00  1.286e+00  -1.163  0.24611
## class_a              6.102e+00  5.070e+00   1.203  0.23007
## class_b              1.215e+00  5.350e+00   0.227  0.82050
## LEED                 3.140e+00  4.116e+00   0.763  0.44630
## Energystar           3.005e+00  4.502e+00   0.667  0.50518
## green_rating                NA         NA      NA       NA
## net                 -6.229e-01  1.593e+00  -0.391  0.69619
## amenities           -5.816e+00  1.808e+00  -3.217  0.00149 **
## cd_total_07          8.972e-04  5.926e-04   1.514  0.13144
## hd_total07           7.950e-04  3.708e-04   2.144  0.03311 *
## total_dd_07                 NA         NA      NA       NA
## Precipitation        1.879e-01  6.973e-02   2.694  0.00759 **
## Gas_Costs           -6.284e+02  3.295e+02  -1.907  0.05780 .
## Electricity_Costs    2.204e+02  1.120e+02   1.967  0.05043 .
## cluster_rent         1.137e+00  6.331e-02  17.963  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.783 on 222 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.7998, Adjusted R-squared:  0.7818
## F-statistic: 44.35 on 20 and 222 DF,  p-value: < 2.2e-16
```

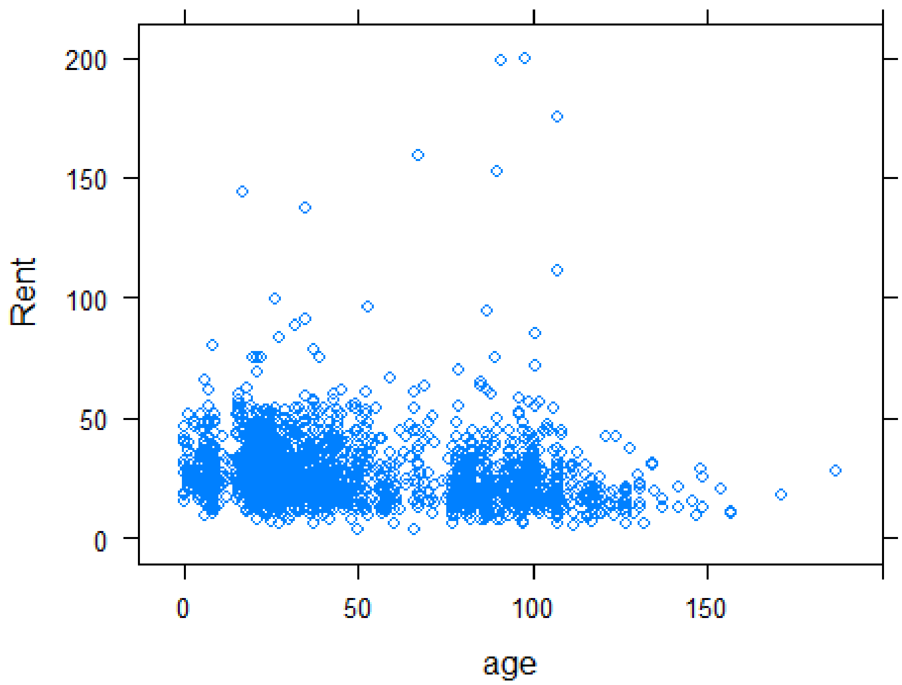
```
green.f=factor(fifteen_green_size$green_rating, levels=c(0,1), labels=c("Not
Green", "Green"))

class_a.f=factor(fifteen_green_size$class_a, levels=c(0,1), labels=c("Not Cla
ss A", "Class A"))

class_b.f=factor(fifteen_green_size$class_a, levels=c(0,1), labels=c("Not Cla
ss B", "Class B"))

amenities.f=factor(fifteen_green_size$amenities, levels=c(0,1),labels=c("No A
menities", "Amenities"))

amenities.f2=factor(nofifteen_green$amenities, levels=c(0,1),labels=c("No Ame
nities", "Amenities"))
```

The excel Guru needs to improve his analysis by making several variables in the data into factors. He is generalizing the situatoin because One can't solely consider one variable (such as the green aspect) to be the only factor that makes an impact on rent. With the new data frame (15 stories, green, and over 250,000 sq ft), a regression model displayed the statistically significant variables: age, amenities, hd_total07, precipitation, and cluster_rent. Below are several plots that show the importance of these variables on rent and compare the impacts of several variables on the two data sets. The buildings that are in the not green data set have less amenities than that of the buildings that are clustering into the green set. If you take a look at the graphs below, you can see that there are many in the green set that have amenities. So, in this case, we don't know for certain if the market or rent price is being driven by amenities or the "green" aspect of the building itself.

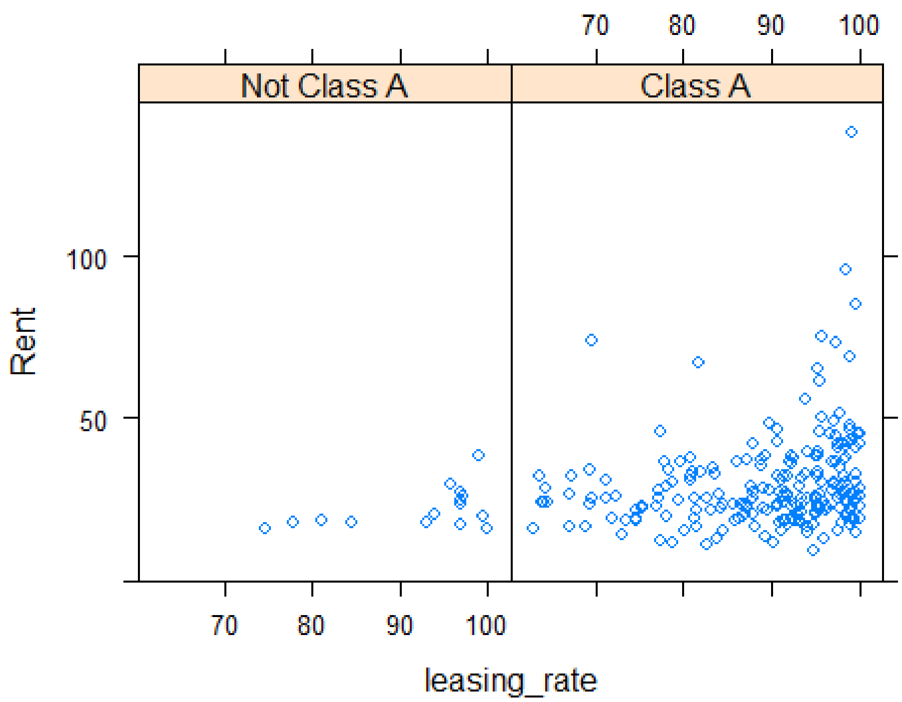
```
xyplot(Rent~age,data=fifteen_green_size)
```




```
xyplot(Rent~age,data=nofifteen_green)
```

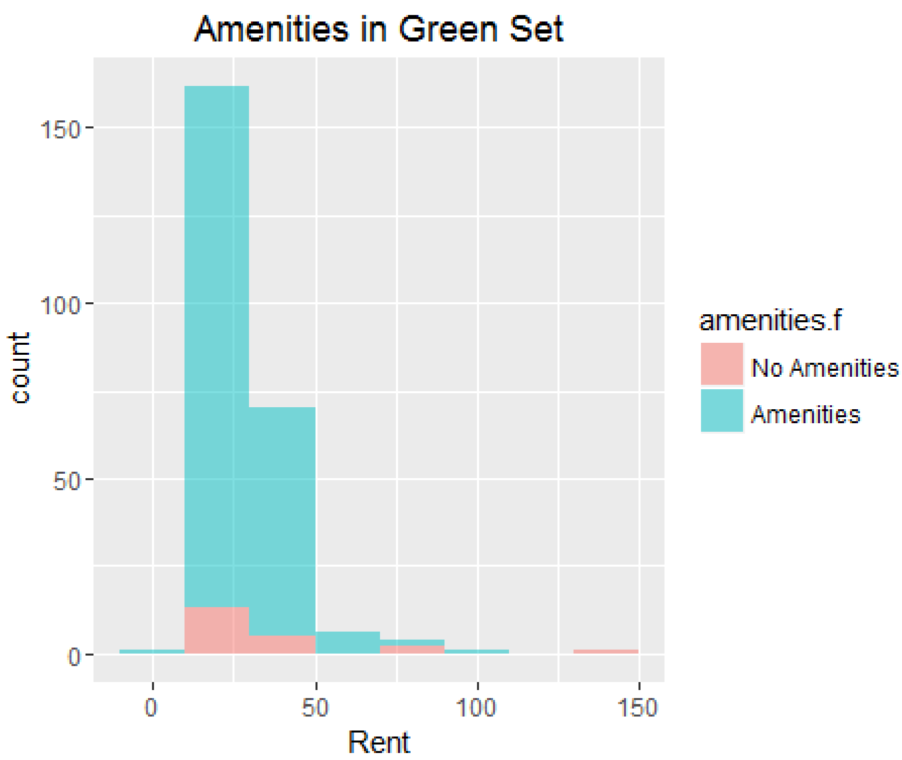



```
xyplot(Rent~leasing_rate|class_a.f,data=fifteen_green_size)
```

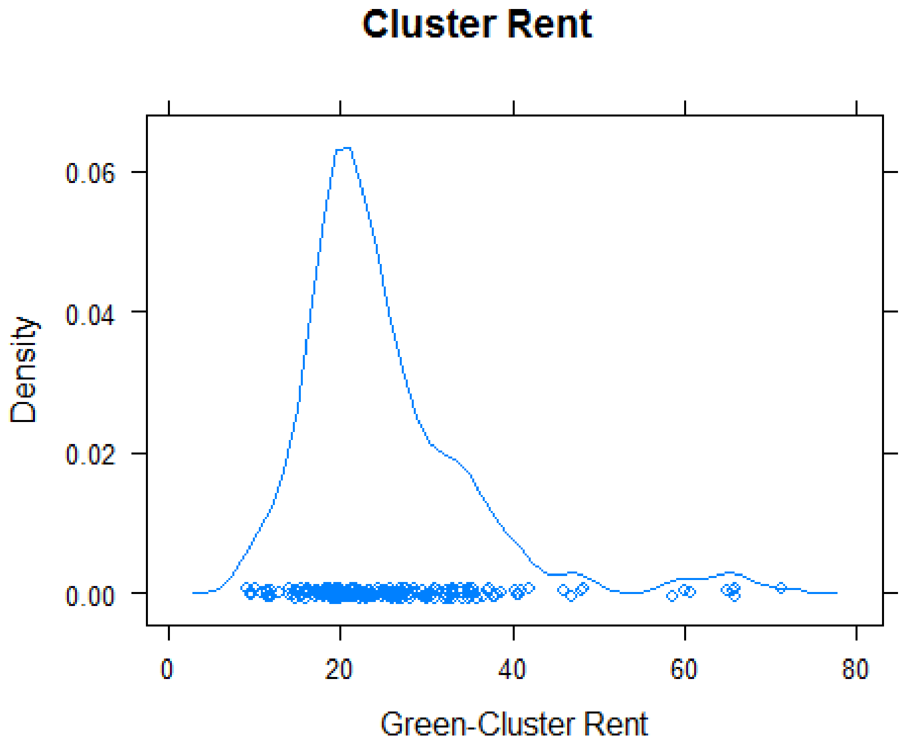



```
xyplot(Rent~leasing_rate|class_b.f,data=fifteen_green_size)
```

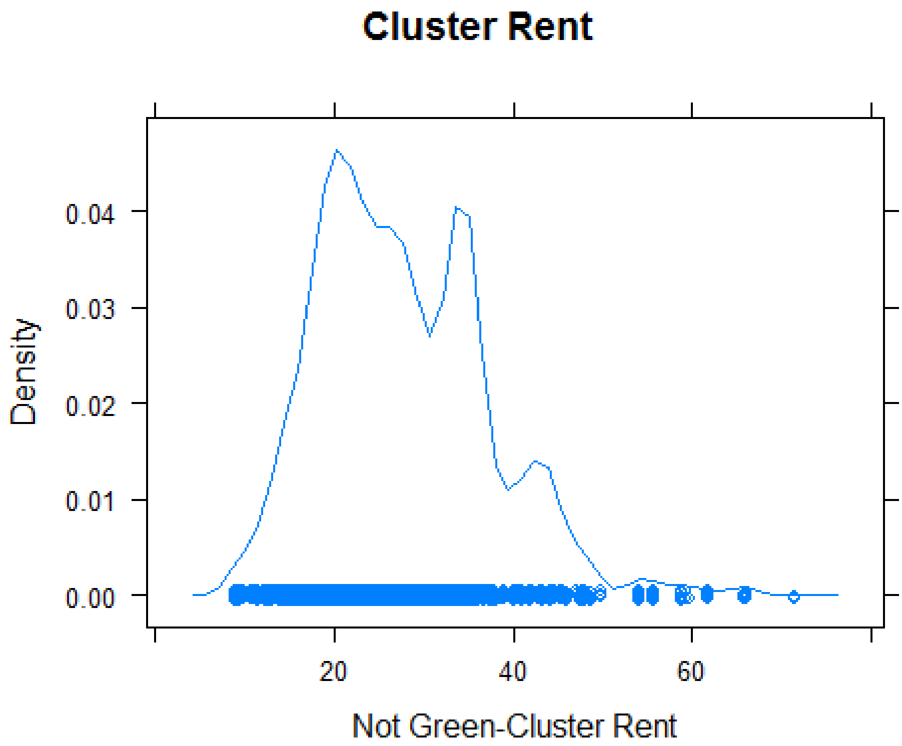

```
qplot(Rent, data=fifteen_green_size, fill=amenities.f, alpha=I(.5), binwidth=
20, main="Amenities in Green Set")
```



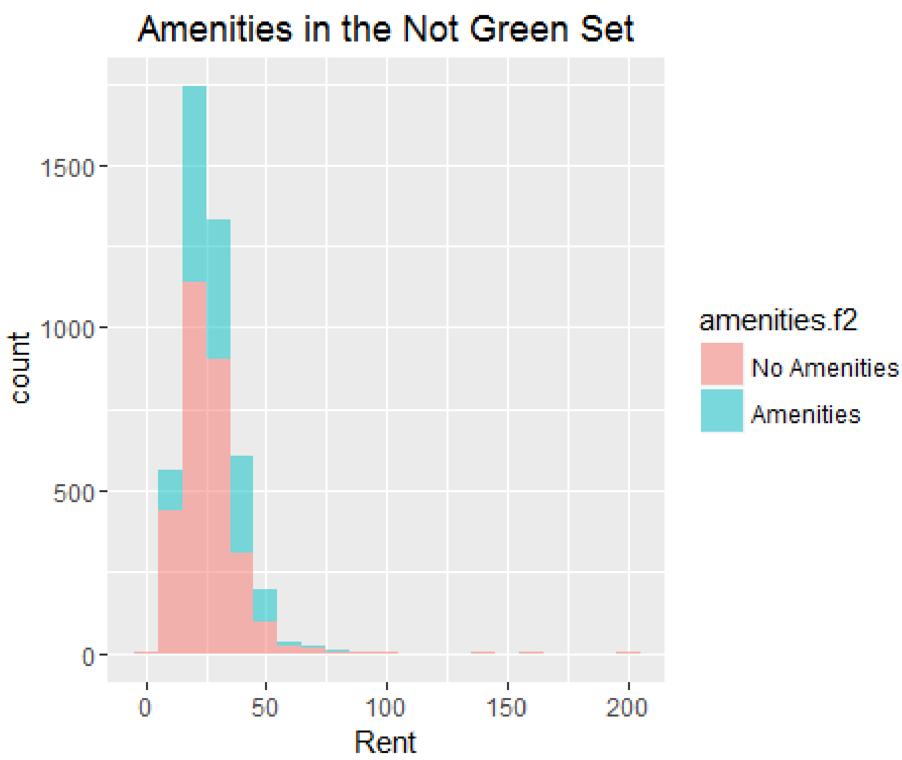
```
densityplot(~fifteen_green_size$cluster_rent, main="Cluster Rent", xlab="Green-Cluster Rent")
```




```
densityplot(~nofifteen_green$cluster_rent, main="Cluster Rent", xlab="Not Green-Cluster Rent")
```




```r
qplot(Rent, data=nofifteen_green, fill=amenities.f2, alpha=I(.5), binwidth=10
, main="Amenities in the Not Green Set")
```



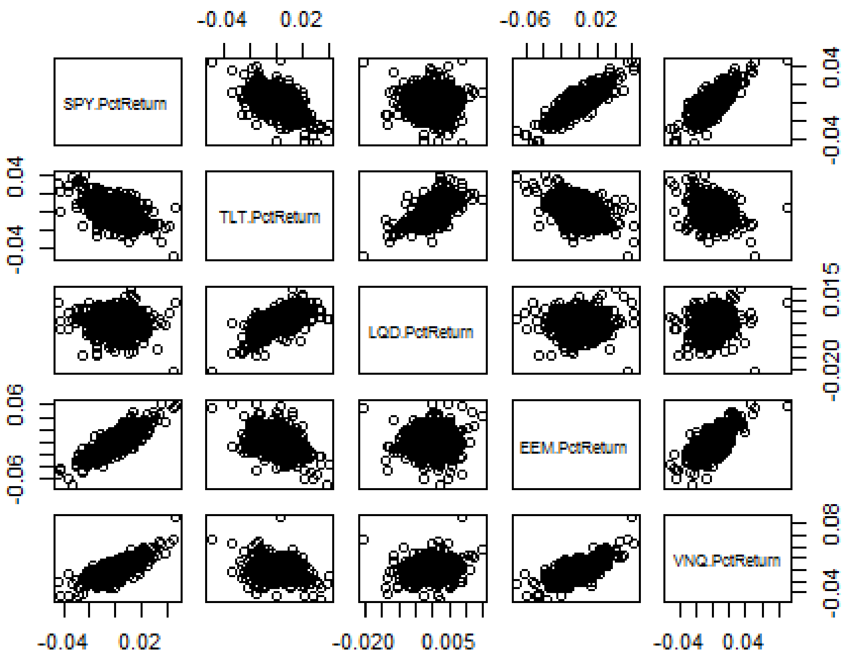
```r
median(fifteen_green_size$cluster_rent)

## [1] 22.5

median(nofifteen_green$cluster_rent)

## [1] 26.69
```

Assuming the new building would be considered "class a", the excel Guru needs to take variability in rent into account for this factor. Additionally, the median cluster rent for those in the green set are much lower than those in the not green set. Cluster rent is based off of local market, so it'd be beneficial to figure out the areas of these different clusters in order to compare it to the Austin housing market.
His conclusion can be improved in several ways, as explained. He needs to improve his numbers and have more information to back up his claims since there are more variables that seem to have an association with rent than soley whether or not the building is green certified, which is important consideration to take into account since the investment is so large.

Question 3 There are 5 asset classes: US domestic equities, US Treasury Bonds, Investment-grade corporate bonds, Emerging-market equities and Real estate. The two most risky of the assets are emerging-market equities and real estate. For the aggressive portfolio, we did an even 50/50 split of those two asset classes. US domestic equities, US Treasury bonds and Investment-grade corporate bonds are the most risk averse. For the safe portfolio we used 30% domestic equities, 40% Treasury bonds and 30% Investment-grade corporate bonds. I used a 30/40/30 split to make it possible to hedge against risk by not losing too much from any part of my portfolio.

```r
library(fImport)

## Loading required package: timeDate

## Loading required package: timeSeries

library(mosaic)
library(foreach)


#Create Portfolio
Portfolio = c("SPY", "TLT", "LQD", "EEM", "VNQ")
Prices = yahooSeries(Portfolio, from='2011-08-07', to='2016-08-07')


YahooPricesToReturns = function(series) {
    cols = grep('Adj.Close', colnames(series))
    closingprice = series[,cols]
    N = nrow(closingprice)
    percentreturn = as.data.frame(closingprice[2:N,])/as.data.frame(closingpr
ice[1:(N-1),]) - 1
    names = strsplit(colnames(percentreturn), '.', fixed=TRUE)
```

```
    names = lapply(names, function(x) return(paste0(x[1], ".PctReturn")))
    colnames(percentreturn) = names
    as.matrix(na.omit(percentreturn))
}

Returns = YahooPricesToReturns(Prices)
pairs(Returns)
```




```
cor(Returns)

##                 SPY.PctReturn TLT.PctReturn LQD.PctReturn EEM.PctReturn
## SPY.PctReturn       1.0000000    -0.5339894  -0.113386220   0.828821210
## TLT.PctReturn      -0.5339894     1.0000000   0.723976609  -0.436298713
## LQD.PctReturn      -0.1133862     0.7239766   1.000000000  -0.005922508
## EEM.PctReturn       0.8288212    -0.4362987  -0.005922508   1.000000000
## VNQ.PctReturn       0.7469290    -0.2471027   0.136649925   0.661773120
##                 VNQ.PctReturn
## SPY.PctReturn       0.7469290
## TLT.PctReturn      -0.2471027
## LQD.PctReturn       0.1366499
## EEM.PctReturn       0.6617731
## VNQ.PctReturn       1.0000000

set.seed(23)
even_split = foreach(i=1:5000, .combine='rbind') %do%{
  wealth = 100000
  weights = c(.2, .2, .2, .2, .2)
```

```
  holdings = weights * wealth
  days = 20
  tracker = rep(0,days)
  for(today in 1:days){
    today_return = resample(Returns, 1, orig.ids=FALSE)
    holdings = holdings + holdings*today_return
    wealth = sum(holdings)
    tracker[today] = wealth
  }
  wealth
}

set.seed(23)
safe = foreach(i=1:5000, .combine='rbind') %do%{
  wealth = 100000
  weights = c(.3, .4, .3, .0, .0)
  holdings = weights * wealth
  days = 20
  tracker = rep(0, days)
  for(today in 1:days){
    today_return = resample(Returns, 1, orig.ids=FALSE)
    holdings = holdings + holdings*today_return
    wealth = sum(holdings)
    tracker[today] = wealth
  }
  wealth
}


set.seed(23)
risk = foreach(i=1:5000, .combine='rbind') %do% {
  wealth = 100000
  weights = c(.0, .0, .0, .5, .5)
  holdings = weights * wealth
  days = 20
  tracker = rep(0, days)
  for(today in 1:days){
    today_return = resample(Returns, 1, orig.ids=FALSE)
    holdings = holdings + holdings*today_return
    wealth = sum(holdings)
    tracker[today] = wealth
  }
  wealth
}
```

Question 4

R Markdown

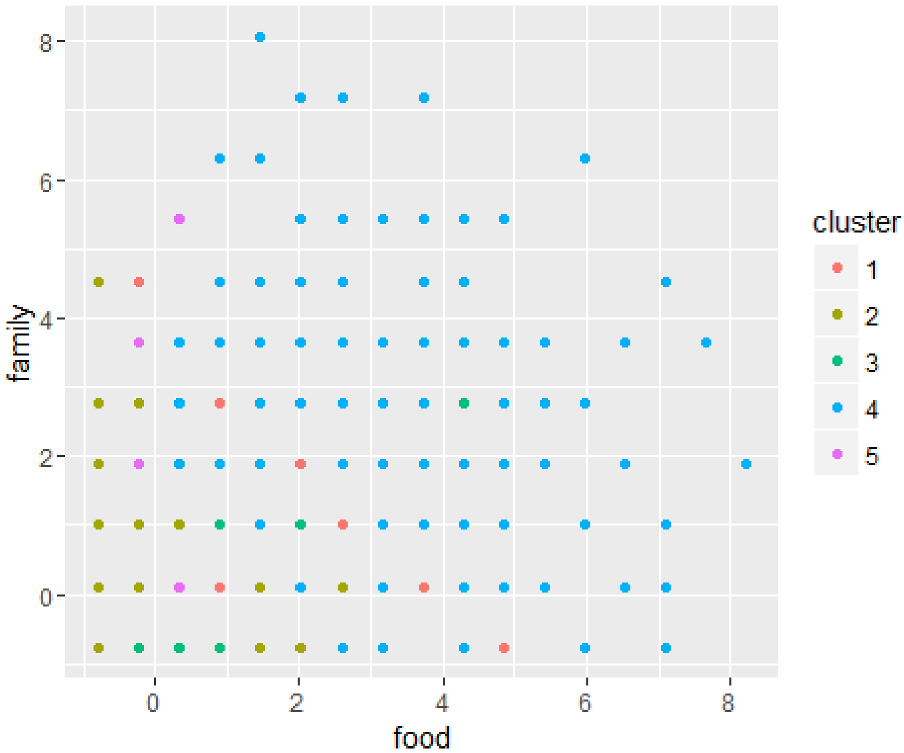
Including Plots

You can also embed plots, for example:

```
library(ggplot2)
mydata = read.csv('social_marketing.csv')
mydata = mydata[,-1]
str(mydata)
```

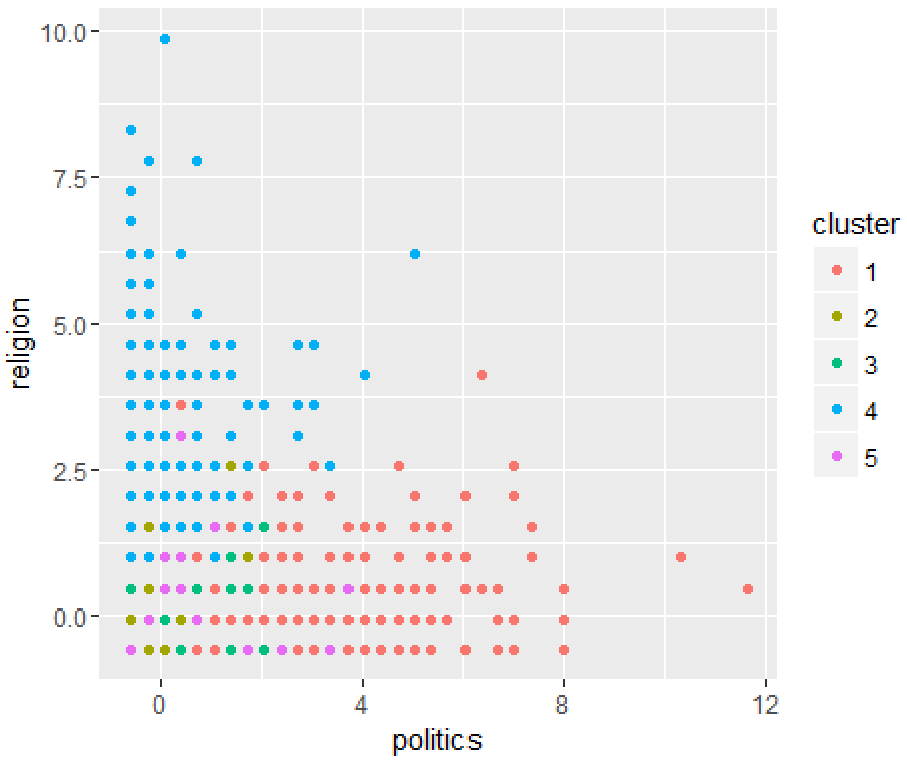
```
## 'data.frame':    7882 obs. of  36 variables:
##  $ chatter         : int  2 3 6 1 5 6 1 5 6 5 ...
##  $ current_events  : int  0 3 3 5 2 4 2 3 2 2 ...
##  $ travel          : int  2 2 4 2 0 2 7 3 0 4 ...
##  $ photo_sharing   : int  2 1 3 2 6 7 1 6 1 4 ...
##  $ uncategorized   : int  2 1 1 0 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 ...

J = scale(mydata, center = TRUE, scale = TRUE)

H20 = kmeans(J, 5, nstart = 55)

df = data.frame(J)
df$cluster = factor(H20$cluster)
ggplot(data = df, aes(x=food, y=family, color=cluster)) + geom_point()
```

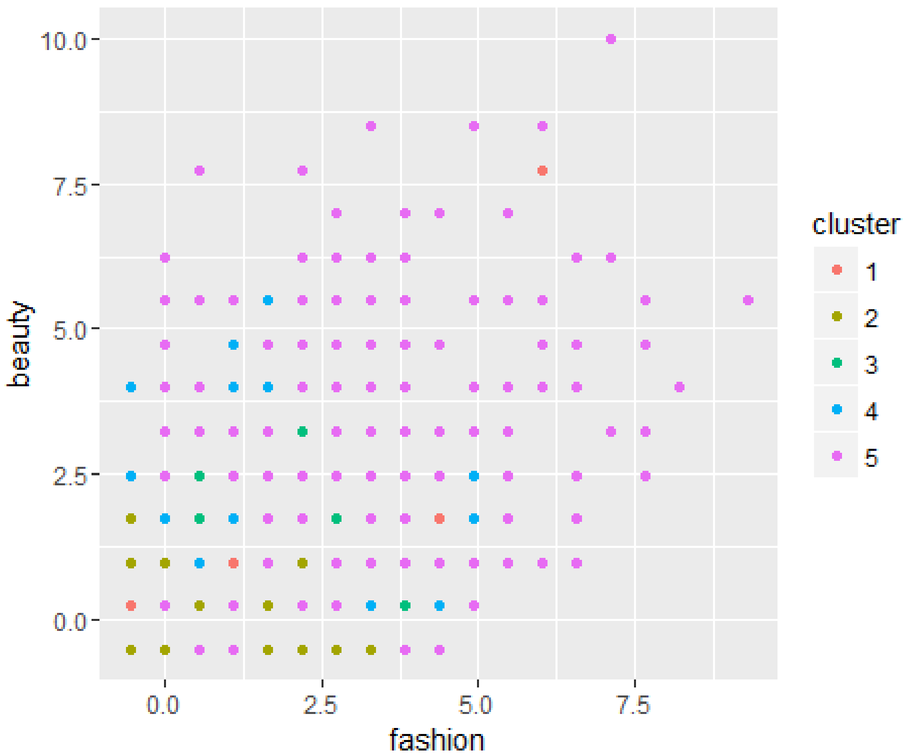



```
print(apply(H20$centers,1,function(x) colnames(J)[order(x, decreasing = TRUE)
[1:6]]))

##      1            2                3                  4
## [1,] "politics"   "adult"          "health_nutrition" "religion"
## [2,] "news"       "spam"           "personal_fitness" "parenting"
## [3,] "travel"     "online_gaming"  "outdoors"         "sports_fandom"
## [4,] "computers"  "current_events" "eco"              "food"
## [5,] "automotive" "tv_film"        "food"             "school"
## [6,] "business"   "college_uni"    "cooking"          "family"
##      5
## [1,] "photo_sharing"
## [2,] "fashion"
## [3,] "cooking"
## [4,] "beauty"
```

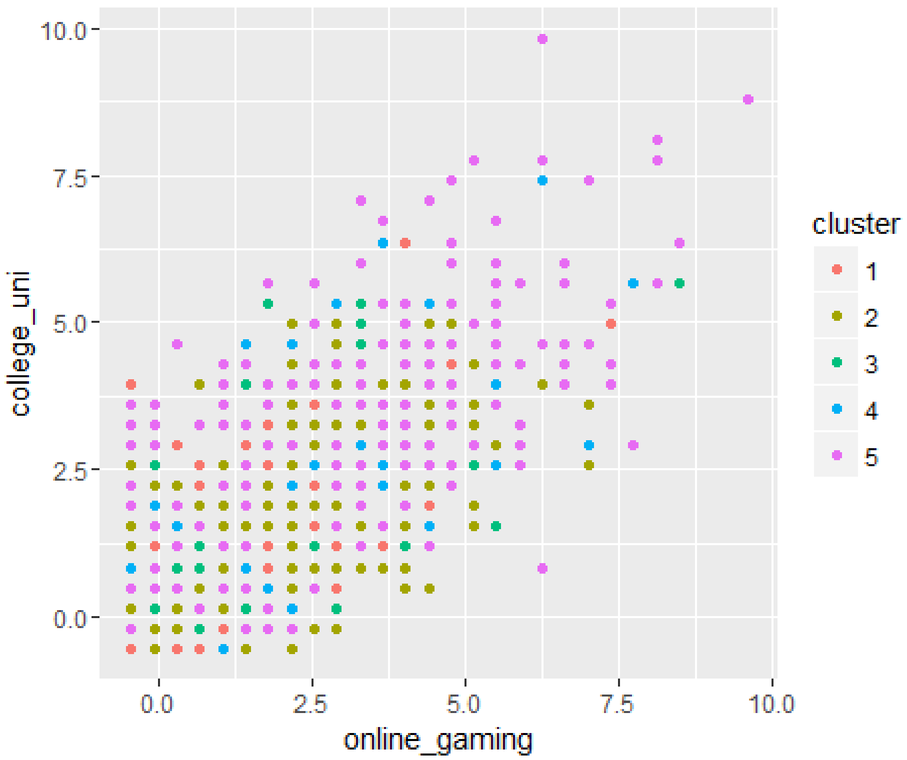
```
## [5,] "shopping"
## [6,] "chatter"

ggplot(data = df, aes(x=politics, y=religion, color=cluster)) + geom_point()
```




```
print(apply(H20$centers,1,function(x) colnames(J)[order(x, decreasing = TRUE)
[1:6]]))

##      1            2                3                  4
## [1,] "politics"   "adult"          "health_nutrition" "religion"
## [2,] "news"       "spam"           "personal_fitness" "parenting"
## [3,] "travel"     "online_gaming"  "outdoors"         "sports_fandom"
## [4,] "computers"  "current_events" "eco"              "food"
## [5,] "automotive" "tv_film"        "food"             "school"
## [6,] "business"   "college_uni"    "cooking"          "family"
##      5
## [1,] "photo_sharing"
## [2,] "fashion"
## [3,] "cooking"
## [4,] "beauty"
## [5,] "shopping"
## [6,] "chatter"

ggplot(data = df, aes(x=fashion, y=beauty, color=cluster)) + geom_point()
```




```
print(apply(H20$centers,1,function(x) colnames(J)[order(x, decreasing = TRUE)
[1:6]]))

##      1            2                3                  4
## [1,] "politics"   "adult"          "health_nutrition" "religion"
## [2,] "news"       "spam"           "personal_fitness" "parenting"
## [3,] "travel"     "online_gaming"  "outdoors"         "sports_fandom"
## [4,] "computers"  "current_events" "eco"              "food"
## [5,] "automotive" "tv_film"        "food"             "school"
## [6,] "business"   "college_uni"    "cooking"          "family"
##      5
## [1,] "photo_sharing"
## [2,] "fashion"
## [3,] "cooking"
## [4,] "beauty"
## [5,] "shopping"
## [6,] "chatter"

ggplot(data = df, aes(x=online_gaming, y=college_uni, color=cluster)) + geom_
point()
```




```
print(apply(H20$centers,1,function(x) colnames(J)[order(x, decreasing = TRUE)
[1:6]]))

##      1            2                3                  4
## [1,] "politics"   "adult"          "health_nutrition" "religion"
## [2,] "news"       "spam"           "personal_fitness" "parenting"
## [3,] "travel"     "online_gaming"  "outdoors"         "sports_fandom"
## [4,] "computers"  "current_events" "eco"              "food"
## [5,] "automotive" "tv_film"        "food"             "school"
## [6,] "business"   "college_uni"    "cooking"          "family"
##      5
## [1,] "photo_sharing"
## [2,] "fashion"
## [3,] "cooking"
## [4,] "beauty"
## [5,] "shopping"
## [6,] "chatter"
```

We divided the data into 5 market segments, which are actually clusters where the 5 most talked about topics are employed in each. For the first plot, we arbitrarily chose the two variables of interest to be food and family. We took twitter users from each these 5 clusters, separating them by color and plotting them in a scatterplot with food on the x-axis and family on the y-axis. This arrangement enabled us to see how much each user in each cluster talked about one of the two variables in relation to another. An interesting observation was that cluster 2 had by far the highest tweets about both food and family among users, with most of the dots located in the upper right hand corner of the plot. This would make sense, as both food and family are among the 5 most talked about variables in cluster 2. For the plot we obtained when running on a religion (y axis) and politics (x axis) scatterplot, we obtained a different cluster distribution. Cluster 4 talked the most about religion by far, but very little about politics. By contrast, cluster 2 talked the most about politicis, but not particularly much about religion. This is once again logically explained by the fact that religion is the most talked about subject in cluser 4, whereas politics is most frequently mentioned in cluster 2. We included more models, each with their own set of clusters and plots indicating which topics are most popular among which clusters. The model thus allows us to find useful insights regarding the most popular subjects among each cluster, enabling the company to adjust their marketing campaigns accordingly per market segment.