Machine Learning Part 2 Exam

Sai Bhargav, Apurva Audi, Vivian Wang, Muskaan Singhania

2022-08-14

GitHub Link to RMD File: https://github.com/muskaansinghania85/ML_Projects/blob/main/ML_Part2_Exam.Rmd

Probability Practice

Part a

P(Y) = 0.65 and P(N) = 0.35 (Y = Yes, N = No)

P(R) = 0.3 and P(T) = 0.7 (R = Random Clicker, T = Truthful Clicker)

P(Y|R) = 0.5 = P(N|R)

P(Y) = P(R) * P(Y|R) + P(T) * P(Y|T)

=> P(Y|T) = (P(Y) - P(R) * P(Y|R))/P(T)

Substituting the values we get:

P(Y|T) = (0.65 - 0.3 * 0.5)/0.7

P(Y|T) = 0.7143 or **71.43%**

Part b

P(P|D) = 0.993, P(N|no D) = 0.9999 (P = positive test, N = negative test, D = have disease, no D = have no disease)

P(D) = 0.000025 => P(no D) = 0.999975

P(D|P) = P(D,P)/P(P) = **P(P|D) * P(D)/P(P)**

In the highlighted expression we know P(P|D), P(D). For P(P):

P(P) = P(P|D) * P(D) + P(P|no D) * P(no D)

=> P(P) = P(P|D) * P(D) + P(P,no D)

=> P(P) = P(P|D) * P(D) + P(no D) - P(N,no D)

=> **P(P) = P(P|D) * P(D) + P(no D) - P(no D) * P(N|no D)**

In the highlighted equation we know all the terms on the right

P(P) = 0.993 * 0.000025 + 0.999975 - 0.999975 * 0.9999 = 0.0001248225

Now:

P(D|P) = 0.993 * 0.000025/0.0001248225

P(D|P) = 0.1989 or **19.89%**

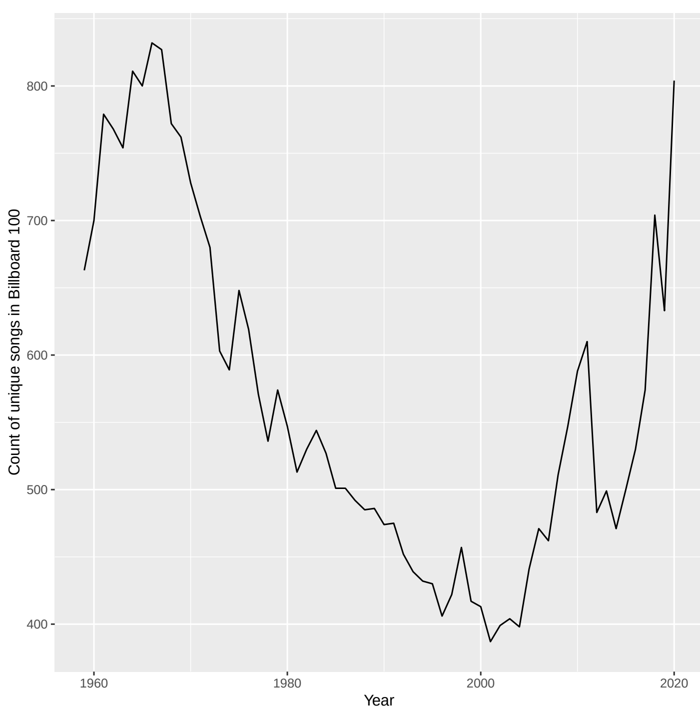
Wrangling the Billboard Top 100

```
##    performer                   song year week week_position
## 1 Patty Duke Don't Just Stand There 1965   29            34
## 2 Patty Duke Don't Just Stand There 1965   30            22
## 3 Patty Duke Don't Just Stand There 1965   31            14
## 4 Patty Duke Don't Just Stand There 1965   32            10
## 5 Patty Duke Don't Just Stand There 1965   33             8
## 6 Patty Duke Don't Just Stand There 1965   34             8
```

Part a

```
## # A tibble: 10 x 3
## # Groups:   performer [10]
##    performer                                 song                          count
##    <chr>                                     <chr>                         <int>
##  1 Imagine Dragons                           Radioactive                      87
##  2 AWOLNATION                                Sail                             79
##  3 Jason Mraz                                I'm Yours                        76
##  4 The Weeknd                                Blinding Lights                  76
##  5 LeAnn Rimes                               How Do I Live                    69
##  6 LMFAO Featuring Lauren Bennett & GoonRock Party Rock Anthem                68
##  7 OneRepublic                               Counting Stars                   68
##  8 Adele                                     Rolling In The Deep              65
##  9 Jewel                                     Foolish Games/You Were Meant~    65
## 10 Carrie Underwood                          Before He Cheats                 64
```

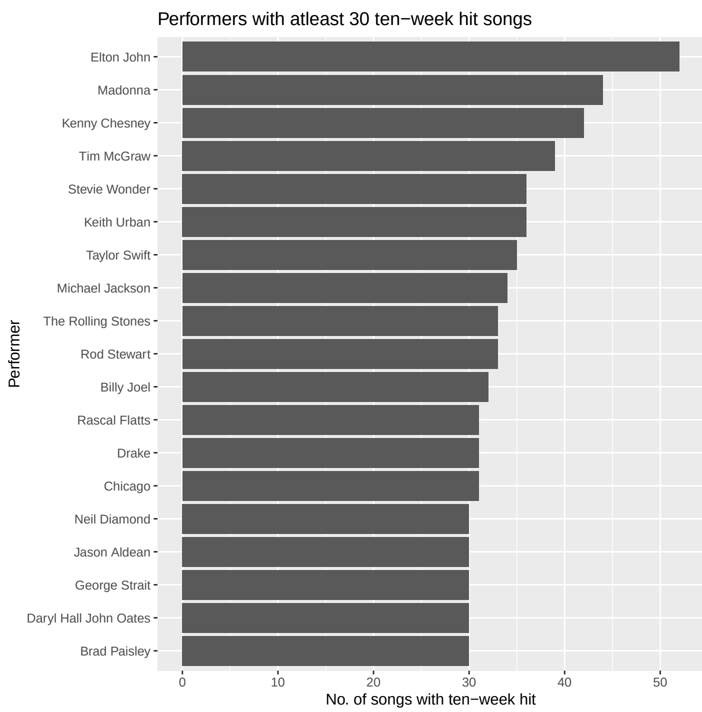
Part b



From the graph we see that the number of unique songs that appeared in the Billboard Top 100 each year initially increased from 1958 to about 1966.

Then it decreased dramatically from that year and does not seem to bounce back up until after 2001. The music diversity is constantly changing and perhaps what we see in the graph represents some kind of cyclical pattern of the music industry. Maybe it tends to have a period of "prosperity" followed by some kind of "depression".

Part c

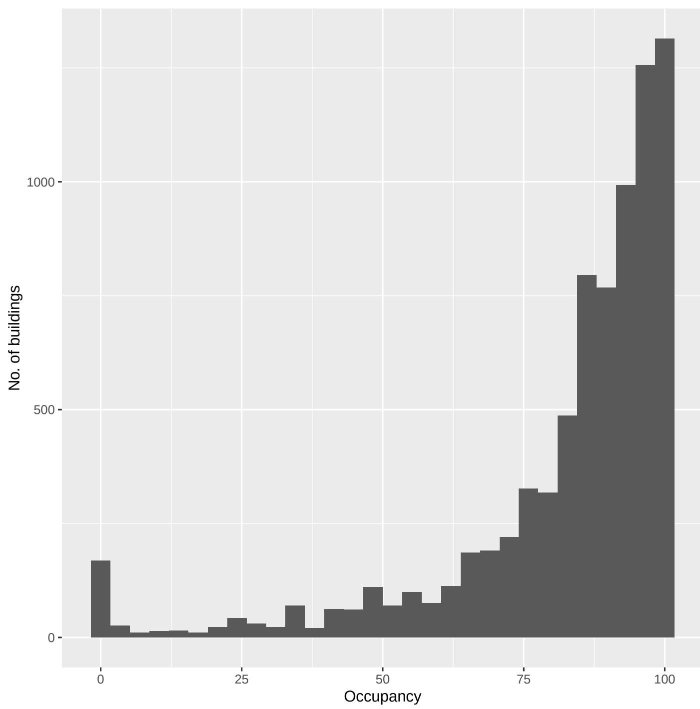


From the graph we see that there are 19 artists who have had at least 30 songs that were "ten-week hits." When you get closer to the top ranking, the range of the number of songs gets much wider. Everyone except the top 4 artists all have pretty similar number of ten-week hits songs. When it gets to Tim McGraw, who is fourth place on the graph, the difference starts to increase. Most noticeably, Elton John, the artist with the highest number of ten-week hits, out-competes other artists on the ranking by a lot. The range is about 22, and Elton John has about 8 more ten-week hits than Madonna, the second place on the list.

Visual story telling part 1: green buildings

```
##   CS_PropertyID cluster   size empl_gr  Rent leasing_rate stories age renovated
## 1        379105       1 260300    2.22 38.56        91.39      14  16         0
## 2        122151       1  67861    2.22 28.57        87.14       5  27         0
## 3        379839       1 164848    2.22 33.31        88.94      13  36         1
## 4         94614       1  93372    2.22 35.00        97.04      13  46         1
## 5        379285       1 174307    2.22 40.69        96.58      16   5         0
## 6         94765       1 231633    2.22 43.16        92.74      14  20         0
##   class_a class_b LEED Energystar green_rating net amenities cd_total_07
## 1       1       0    0          1            1   0         1        4988
## 2       0       1    0          0            0   0         1        4988
## 3       0       1    0          0            0   0         1        4988
## 4       0       1    0          0            0   0         0        4988
## 5       1       0    0          0            0   0         1        4988
```

```
## 6       1       0     0          0            0    0         1        4988
##   hd_total07 total_dd_07 Precipitation  Gas_Costs Electricity_Costs
## 1         58        5046         42.57 0.01370000        0.02900000
## 2         58        5046         42.57 0.01373149        0.02904455
## 3         58        5046         42.57 0.01373149        0.02904455
## 4         58        5046         42.57 0.01373149        0.02904455
## 5         58        5046         42.57 0.01373149        0.02904455
## 6         58        5046         42.57 0.01373149        0.02904455
##   cluster_rent
## 1        36.78
## 2        36.78
## 3        36.78
## 4        36.78
## 5        36.78
## 6        36.78
```

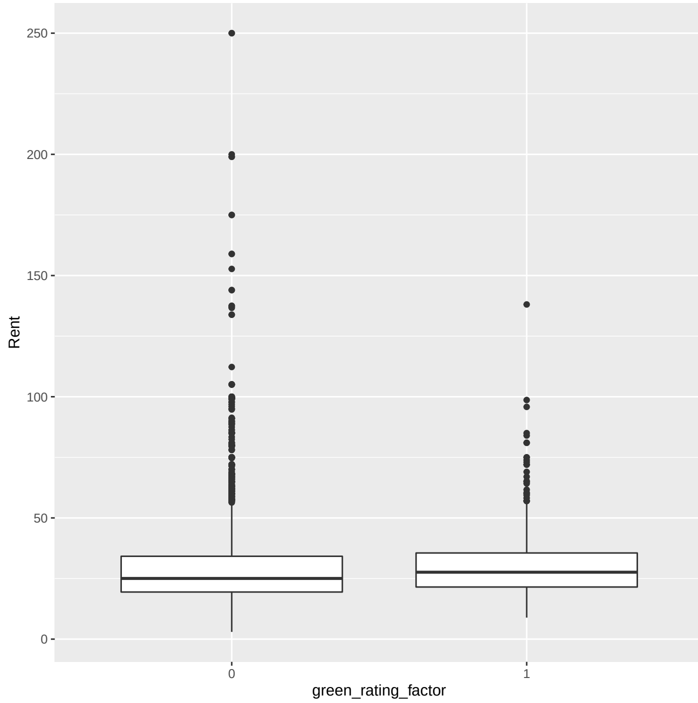


The greenbuildings.csv data has been read in. As seen in the histogram for Occupancy we have some buildings which have very low occupancy (< 10%). So as done by the developer's on-staff, the buildings with occupancy less than 10% will be removed.

```
## The no.of buildings with Occupancy less than 10% is: 215
```

```
## The no.of buildings with Occupancy less than 10% and have green rating: 1
```

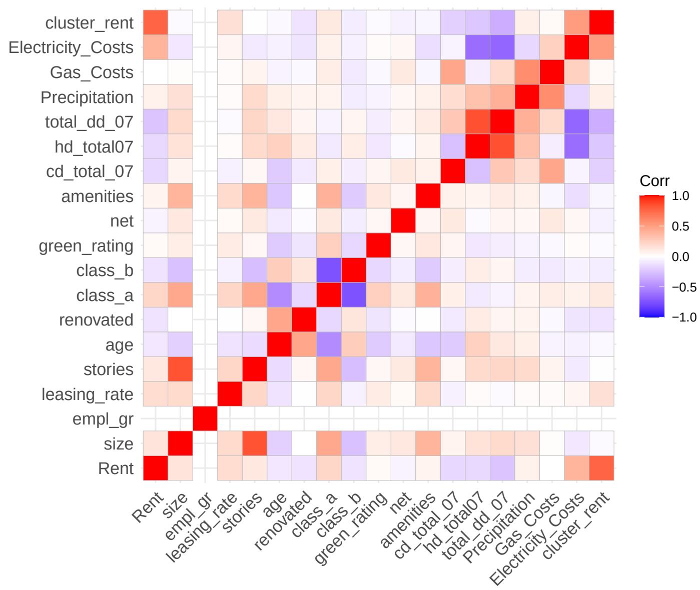
215 buildings have been removed from the dataset due very low. Also, only one of them have green rating. As seen in the and as done by the developer's on-staff, the buildings with occupancy less than 10% have been removed.



We can see from the boxplots that the buildings with green rating do have a slightly higher rent compared to those that do not have green rating.

```
## # A tibble: 2 x 2
##   green_rating median_rent
##          <int>       <dbl>
## 1            0        25.0
## 2            1        27.6
```

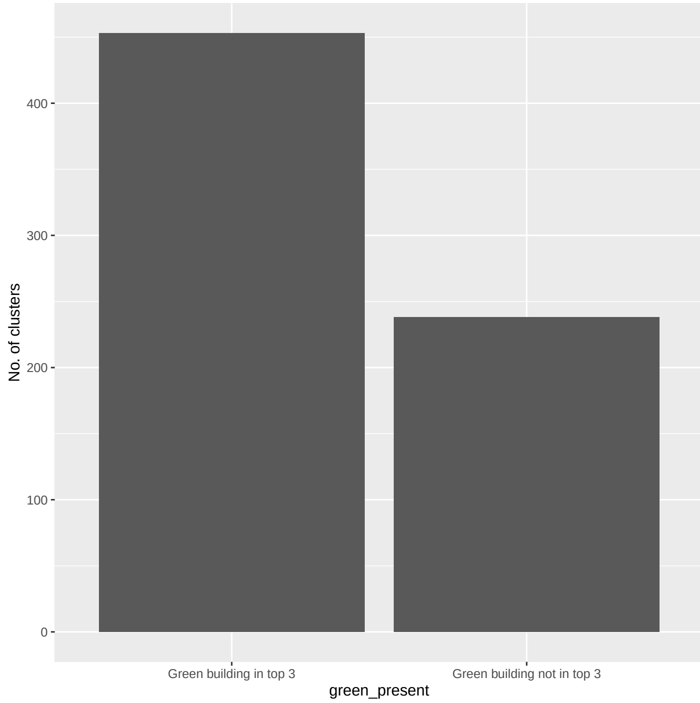
The median values for the green buildings and non-green buildings also seem to be correct. But we need to confirm if the rent is mostly affected by the green rating and no other factors.



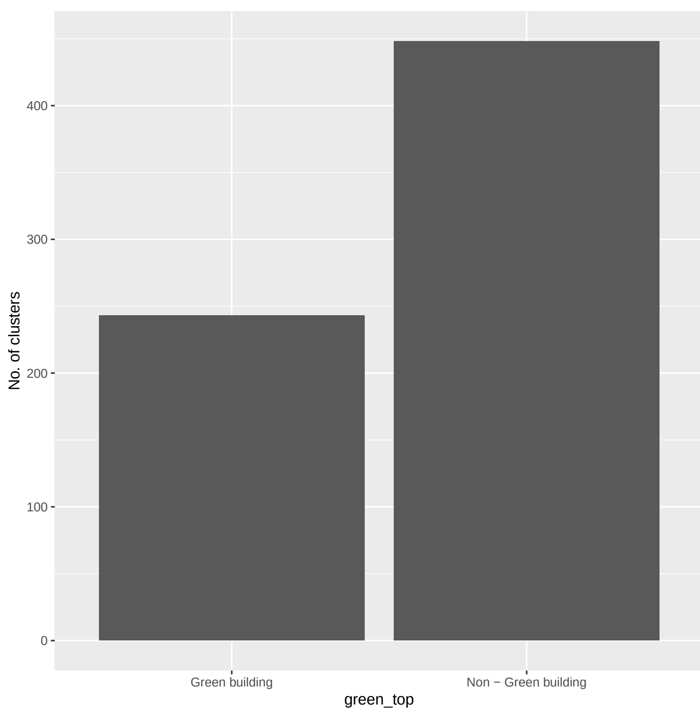
From the correlation plot, we can see that *Rent* is highly correlated with *cluster_rent*. This means that the rent of a building depends on the area in which the building is located. So we need to see if, in each cluster the green building has the highest or atleast top 3 in terms of rent.

```
## # A tibble: 10 x 2
##    cluster green_present
##      <int>         <int>
##  1       1             1
##  2       6             0
##  3       8             0
##  4      11             1
##  5      13             1
##  6      14             0
##  7      16             0
##  8      18             0
##  9      20             0
## 10      22             0
```

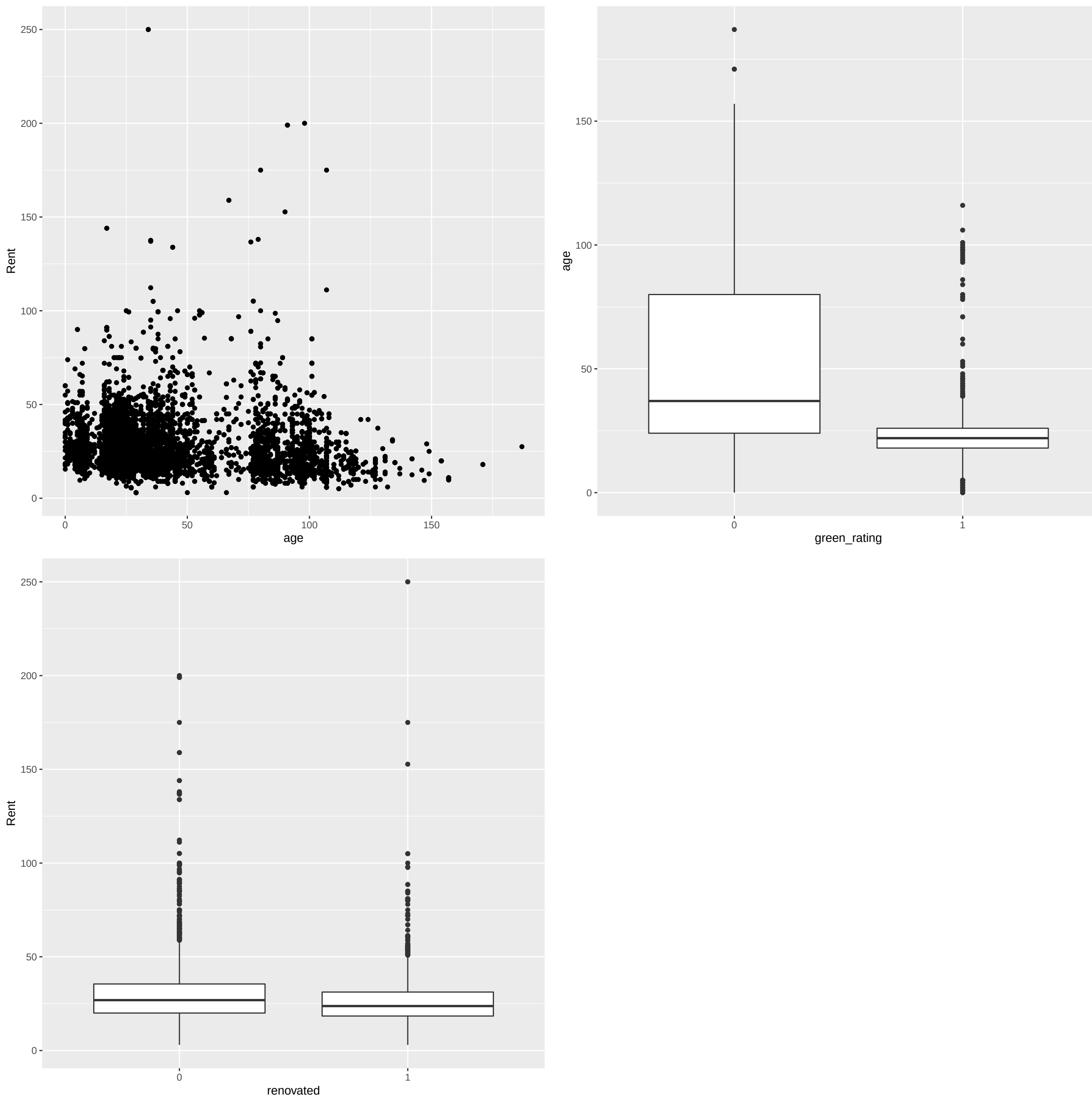
We created a table of cluster number and a column called *green_present* which has two inputs i.e., **0** and **1**. 1 implies that the green building present in the corresponding cluster is in the top 3 in terms of the rent. O implies that the green building is not in the top 3.



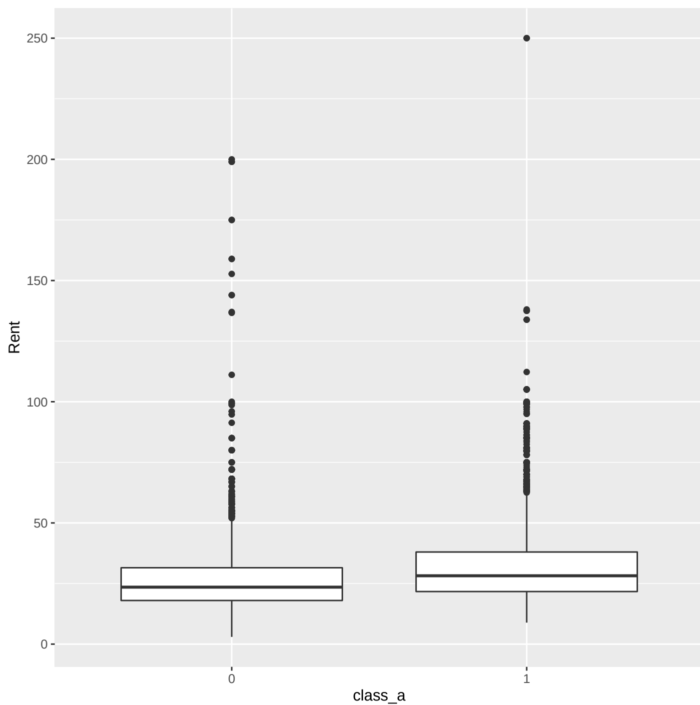
In maximum number of clusters green building is in the top 3 buildings in terms of rent. But there is a considerable number of clusters in which the green building is not even in the top 3. Let us see in how many clusters we have the green building actually at the top.



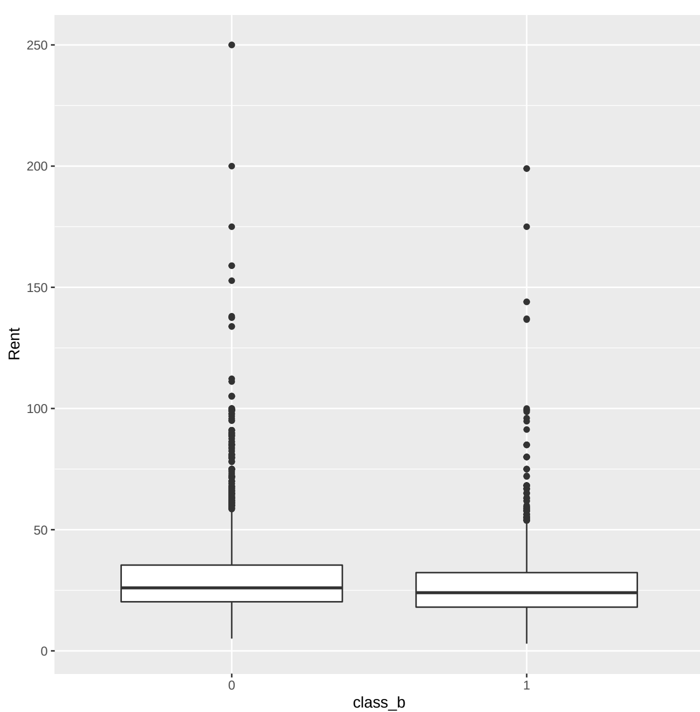
As we can see, more clusters have a non-green building with the highest rent. This confirms that having a green rating does not necessarily lead to higher rents. Also, we can see that *age*, *class-a*, *class-b* and *renovated* columns have a possibility of confounding relationships with *Rent* and *green_rating*.




```
##          green_rating
## renovated    0    1
##         0 4359  539
##         1 2850  146
```




```
##          green_rating
## class_a    0    1
##       0 4598  139
##       1 2611  546
```




```
##          green_rating
## class_b    0    1
```

```
##        0 3714  553
##        1 3495  132
```

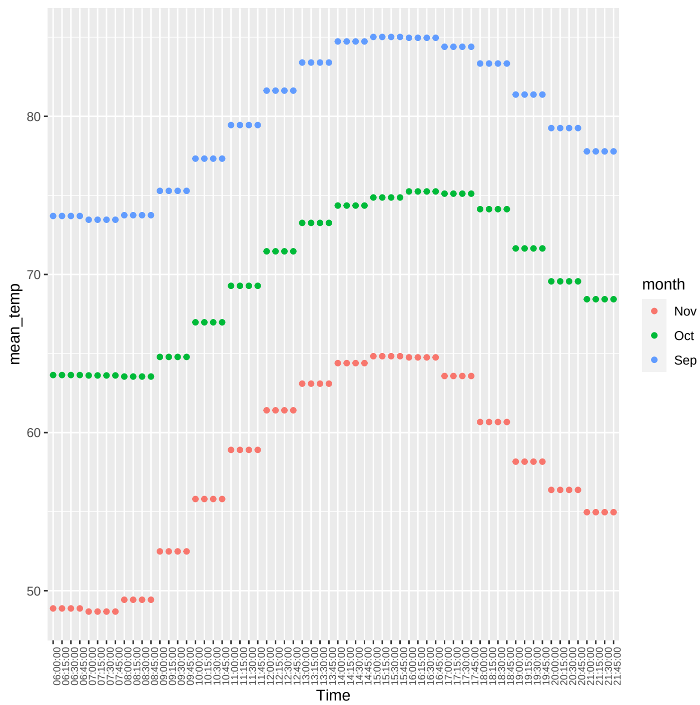
We can see from the graphs that *green_rating* and *Rent* have similar correlations with *age*, *class-a*, *class-b* and *renovated*.

- As *age* increases *Rent* goes down. In the box plot between *green_rating* and *age*, *green_rating* **0** has higher median *age*.
- From the confusion matrix between *renovated* and *green_rating* we see that the probability of having a *green_rating* 1 goes down from **0.08** to **0.05** given the condition that *renovated* is **1**. This implies that if a building has undergone substantial renovations in its lifetime then it is far from having a green rating. That is again connected to the age of the building. Older buildings are usually renovated
- *Class-a* buildings have higher rents. The condition that the given building is *class-a* rated increases the probability of it being a green-rated building from **0.08** to **0.173**
- *Class-b* buildings have lower rents. The condition that the given building is *class-b* rated decreases the probability of it being a green-rated building from **0.08** to **0.03**.

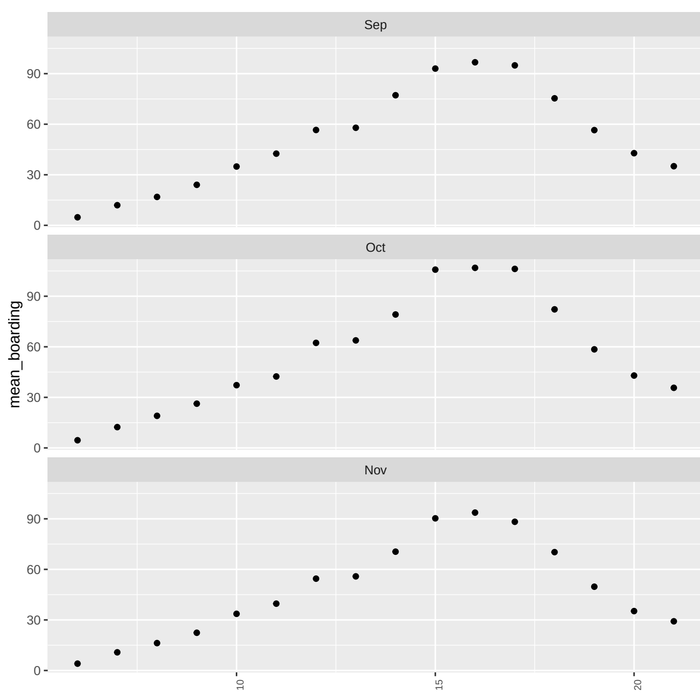
Hence, we can conclude that there are confounding variables which might give the impression that green-rating leads to higher rents.

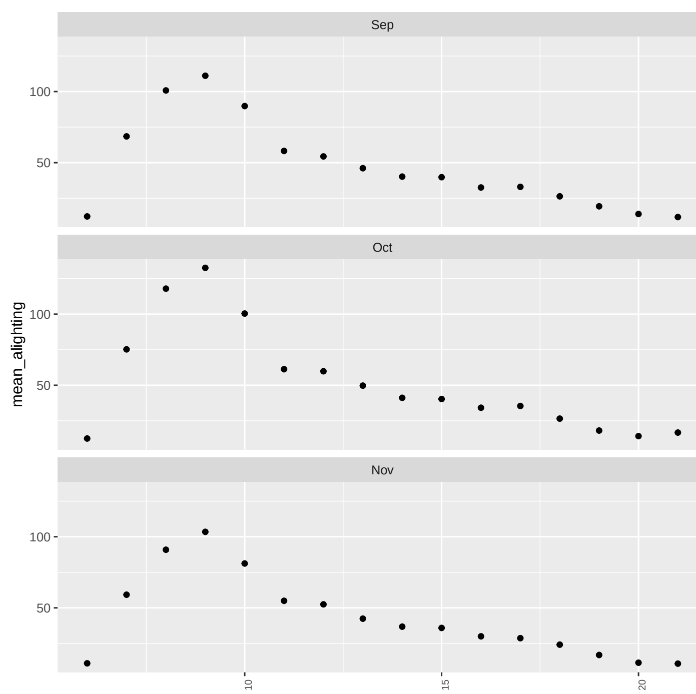
Visual story telling part 2: Capital Metro data

```
##             timestamp boarding alighting day_of_week temperature hour_of_day
## 1 2018-09-01 06:00:00        0         1         Sat       74.82           6
## 2 2018-09-01 06:15:00        2         1         Sat       74.82           6
## 3 2018-09-01 06:30:00        3         4         Sat       74.82           6
## 4 2018-09-01 06:45:00        3         4         Sat       74.82           6
## 5 2018-09-01 07:00:00        2         4         Sat       74.39           7
## 6 2018-09-01 07:15:00        4         4         Sat       74.39           7
##   month weekend
## 1   Sep weekend
## 2   Sep weekend
## 3   Sep weekend
## 4   Sep weekend
## 5   Sep weekend
## 6   Sep weekend
```



The above scatter plot is between time of the day and mean temperature at that time of the day in a particular month. We can clearly see that in all the three months the temperature starts at lowest point at 6 in the morning, rises to a peak in the afternoon and cools down throughout the evening and night. Also, we can see that Sep is hotter than October and October is hotter than November. This represents the arrival of winter season in November.

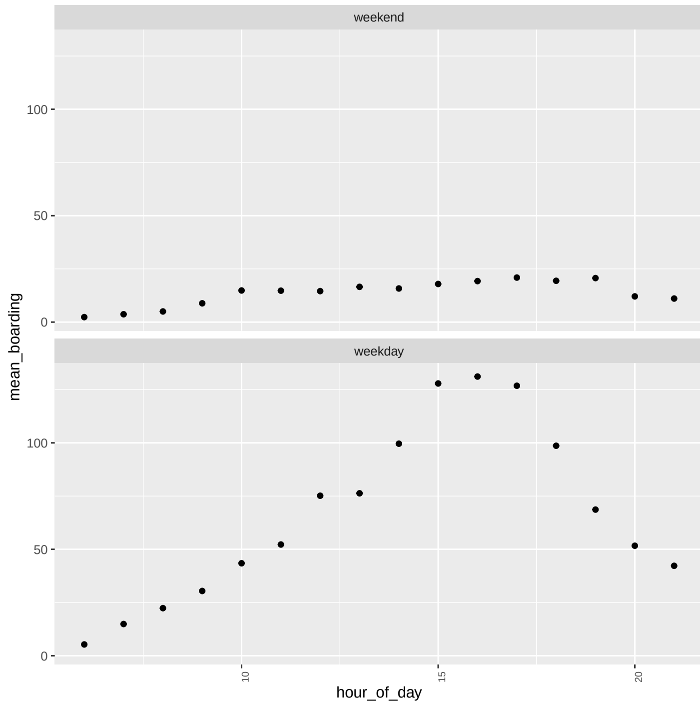


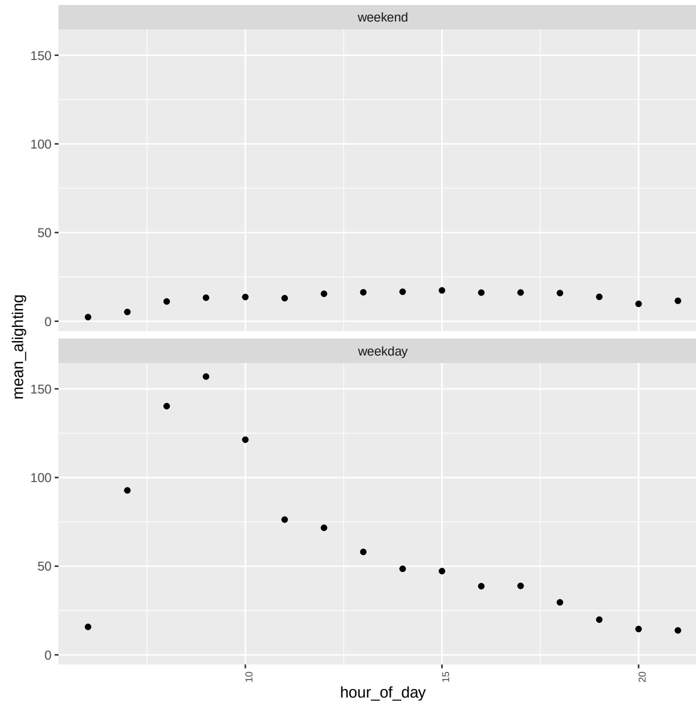


- Most of the boarding of the Capital Metro buses at the UT campus is happening in the evening hours when classes are over for most of the students and they are leaving from the campus
- Most of the alighting of the Capital Metro buses at the UT campus is happening in the morning hours

when classes are going to start for most of the students and they are coming to the campus

We have the data from September to November. In all the three months we see similar trend for alighting and boarding. But, these trends might not be repeated in December because of the winter holidays which will lead to lower student population around the campus. But we do have weekends data which can represent December month.





We can see that during the weekends there is not much activity as there are no classes at UT on weekends.

Portfolio modeling

In this exercise I am going to take three different portfolios. The first portfolio will have 5 ETFs which are considered to be very safe. In the second portfolio, I am going to consider 5 ETFs which are considered to be highly risky. And in the third I am going to take 3 safe and 2 risky ETFs. In all the portfolios, we are starting with an initial capital of $100,000 and checking the final return at the end of 20 days.

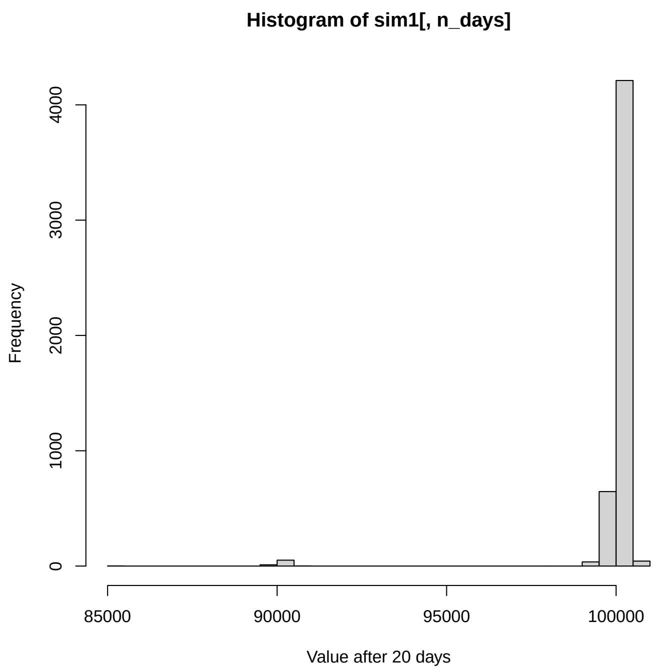
Portfolio 1

The ETFs considered are *SPDR Bloomberg 1-3 Month T-Bill (BIL)*, *iShares Short Treasury Bond (SHV)*, *Invesco Ultra Short Duration (GSY)*, *Goldman Sachs Access Treasury 0-1 Year (GBIL)* and *SPDR SSGA Ultra Short Term Bond (ULST)*. These ETFs are considered to be very safe and not much variation is expected in terms of returns.

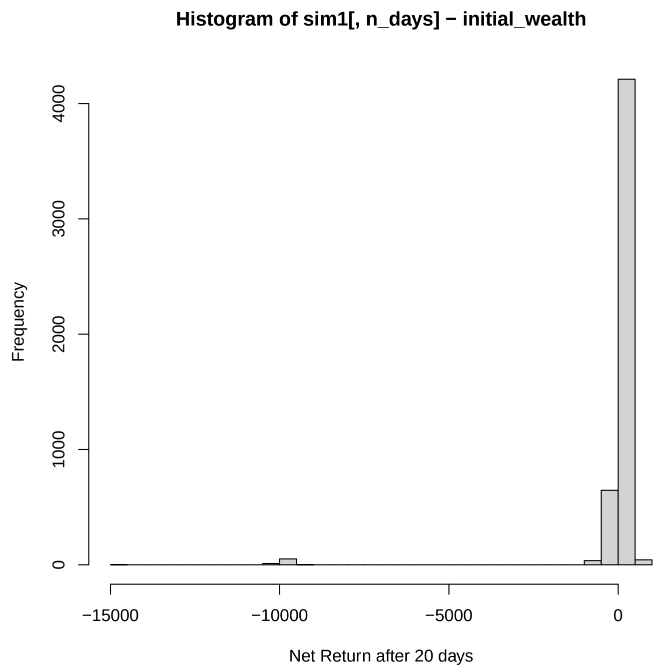
```
##                ClCl.BILa     ClCl.SHVa     ClCl.GSYa    ClCl.GBILa
## 2017-01-04  0.0000000000 -9.065531e-05  0.0000000000 -1.998201e-05
## 2017-01-05 -0.0002187664 -9.066352e-05  0.0003991219 -4.003479e-05
## 2017-01-06  0.0002188143  2.719427e-04 -0.0003989627 -2.398685e-04
## 2017-01-09  0.0000000000  1.813033e-04  0.0000000000  3.999700e-04
## 2017-01-10 -0.0002187664 -9.063972e-05  0.0000000000  0.000000e+00
```

```
## 2017-01-11  0.0002188143  0.000000e+00  0.0001995210  9.990005e-05
##               ClCl.ULSTa
## 2017-01-04  0.0014940488
## 2017-01-05 -0.0004972650
## 2017-01-06  0.0002487065
## 2017-01-09  0.0004974136
## 2017-01-10  0.0000000000
## 2017-01-11 -0.0004971663
```

As we can see, the change in the closing prices of these ETFs is very low. Let's use Bootstrap resampling and see the variation in the next 20 days.




```
## The mean value we have at the end of 20 days is  99963.82
```




```
## The mean total variation is  -36.17581
```

```
## VaR is  222.5169
```

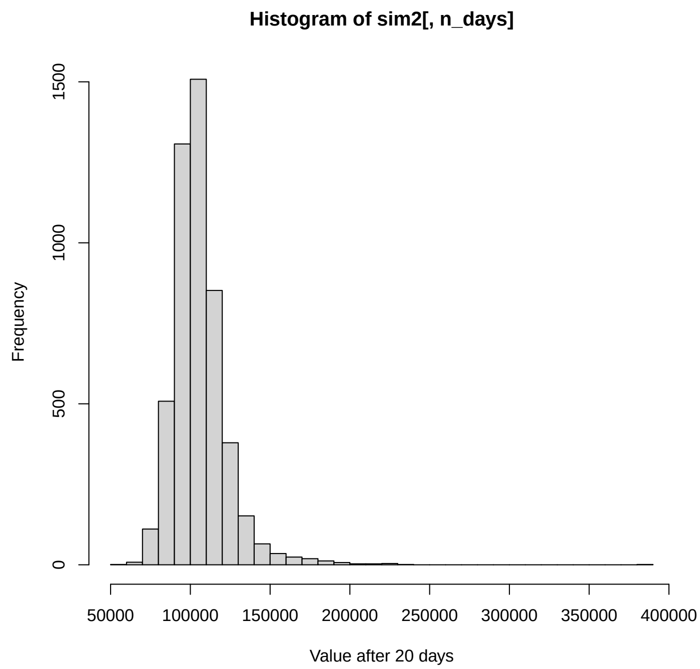
As we were expecting, the variation in returns for this portfolio is very low and hence we get a low VaR. Also, the histogram for the returns has a low standard deviation which also corroborates to the fact that the ETFs in this portfolio are safe.

Portfolio 2

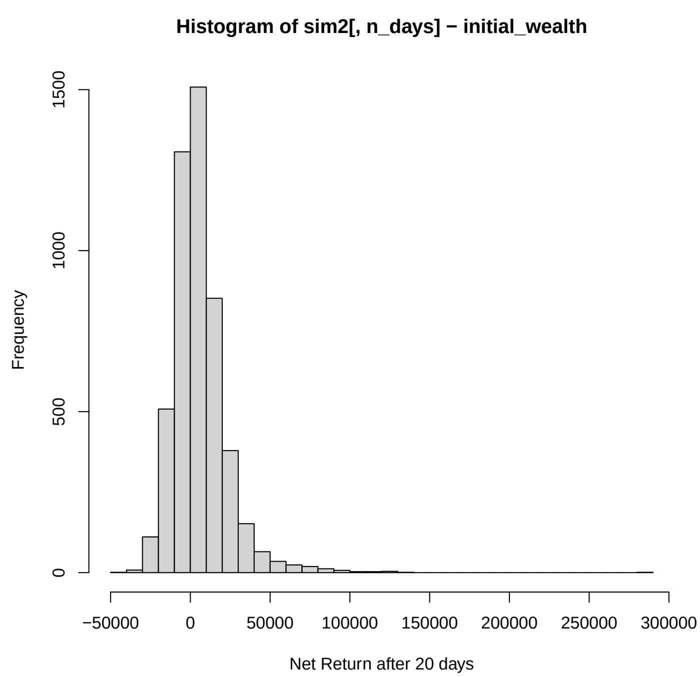
The ETFs considered are *ProShares UltraPro QQQ (TQQQ)*, *ProShares Ultra QQQ (QLD)*, *Direxion Daily S&P 500 Bull 3x Shares (SPXL)*, *Direxion Daily S&P 500 Bull 2x Shares (SPUU)* and *Direxion Daily 20+ Year Treasury Bull 3x Shares (TMF)*. These ETFs are considered to be highly volatile.

```
##             ClCl.TQQQa   ClCl.QLDa    ClCl.SPXLa   ClCl.SPUUa    ClCl.TMFa
## 2017-01-04 0.017036299 0.010501084  0.0177224757  0.021128671  0.012588944
## 2017-01-05 0.017731896 0.011182650 -0.0023218788  0.000000000  0.046486541
## 2017-01-06 0.025355872 0.017537980  0.0112782313  0.008905186 -0.029442147
## 2017-01-09 0.010122863 0.006696674 -0.0094707024  0.000000000  0.026077700
## 2017-01-10 0.005726672 0.004362050 -0.0008936466 -0.009865031 -0.003112137
## 2017-01-11 0.007686832 0.004668838  0.0083176821 -0.002359727  0.010405880
```

We can see that the daily variation of these ETFs are high compared to the Portfolio 1 ETFs that we have seen before.




```
## The mean value we have at the end of 20 days is  105787.7
```




```
## The mean total variation is  5787.724
```

```
## VaR is  15898.38
```

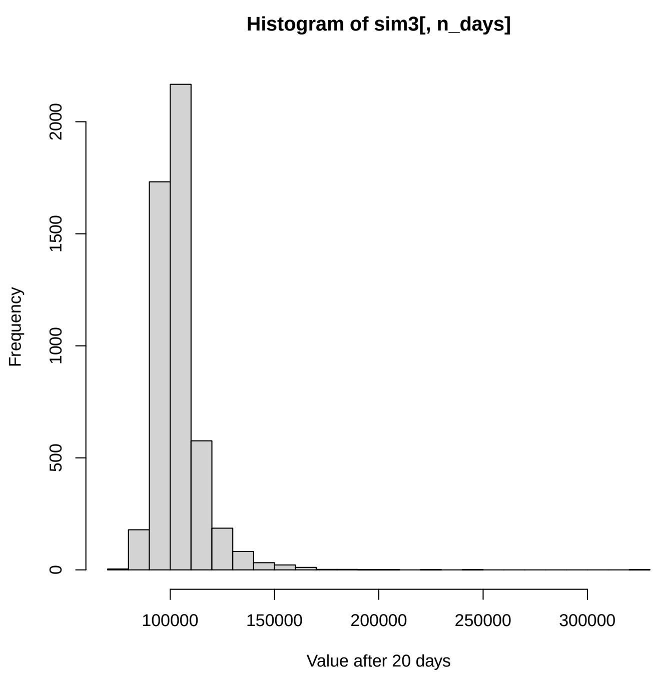
The variation in returns for this portfolio is high and hence we get a high VaR. Also, the histogram for the returns has a high standard deviation which also corroborates to the fact that the ETFs in this portfolio are volatile.

Portfolio 3

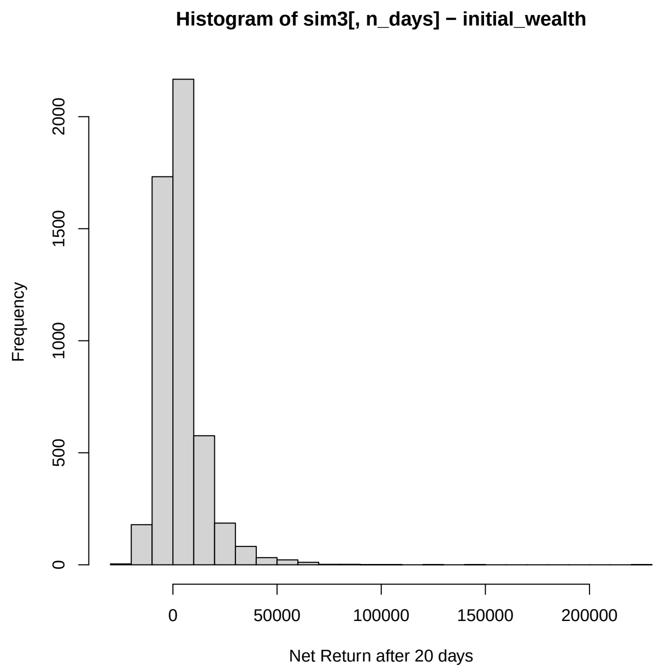
The ETFs considered are *ProShares UltraPro QQQ (TQQQ)*, *ProShares Ultra QQQ (QLD)*, *SPDR Bloomberg 1-3 Month T-Bill (BIL)*, *iShares Short Treasury Bond (SHV)*, *Invesco Ultra Short Duration (GSY)*. We have 2 volatile and 3 safe ETFs in this portfolio.

```
##             ClCl.TQQQa   ClCl.QLDa     ClCl.BILa     ClCl.SHVa     ClCl.GSYa
## 2017-01-04 0.017036299 0.010501084  0.0000000000 -9.065531e-05  0.0000000000
## 2017-01-05 0.017731896 0.011182650 -0.0002187664 -9.066352e-05  0.0003991219
## 2017-01-06 0.025355872 0.017537980  0.0002188143  2.719427e-04 -0.0003989627
## 2017-01-09 0.010122863 0.006696674  0.0000000000  1.813033e-04  0.0000000000
## 2017-01-10 0.005726672 0.004362050 -0.0002187664 -9.063972e-05  0.0000000000
## 2017-01-11 0.007686832 0.004668838  0.0002188143  0.000000e+00  0.0001995210
```

We see that the variation in the closing prices for the two risky portfolios are in another scale compared to the three safe portfolios.




```
## The mean value we have at the end of 20 days is  103952.8
```




```
## The mean total variation is  3952.85
```

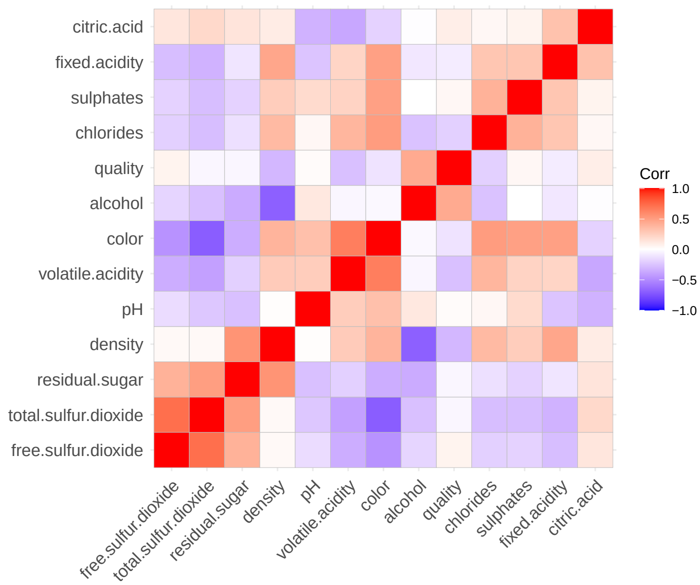
```
## VaR is  9035.16
```

For this mixed portfolio, we can see that the VaR is lower compared to the *Portfolio 2* which had all the risky ETFs. We can also see that the histogram for the final return has more spread compared to the one in *Portfolio 1* but lower spread compared to the one in *Portfolio 2*.

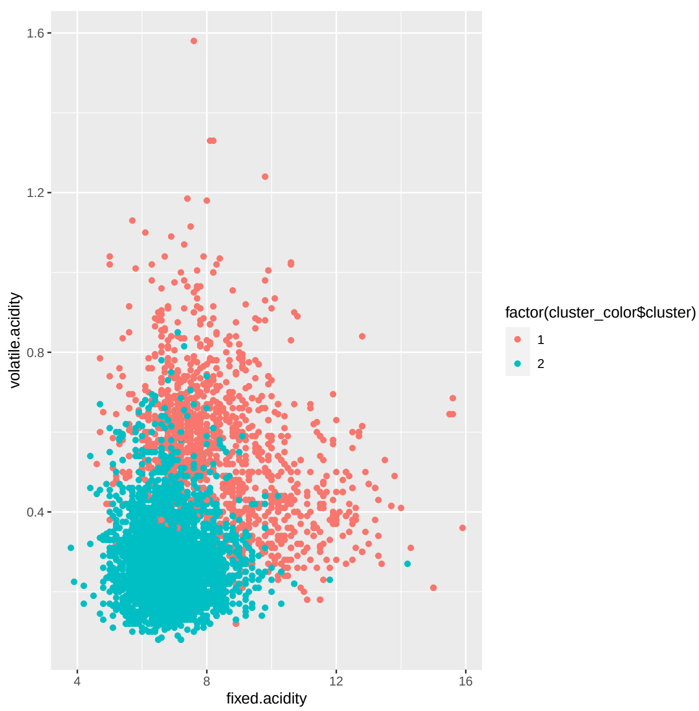
Clustering and PCA

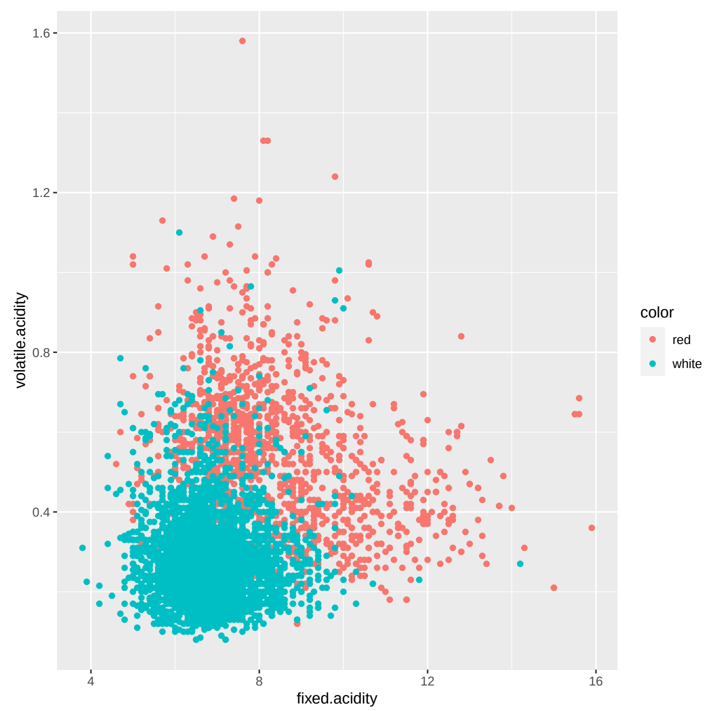
```
##   fixed.acidity volatile.acidity citric.acid residual.sugar chlorides
## 1           7.4             0.70        0.00            1.9     0.076
## 2           7.8             0.88        0.00            2.6     0.098
## 3           7.8             0.76        0.04            2.3     0.092
## 4          11.2             0.28        0.56            1.9     0.075
## 5           7.4             0.70        0.00            1.9     0.076
## 6           7.4             0.66        0.00            1.8     0.075
##   free.sulfur.dioxide total.sulfur.dioxide density   pH sulphates alcohol
## 1                  11                   34  0.9978 3.51      0.56     9.4
## 2                  25                   67  0.9968 3.20      0.68     9.8
## 3                  15                   54  0.9970 3.26      0.65     9.8
## 4                  17                   60  0.9980 3.16      0.58     9.8
## 5                  11                   34  0.9978 3.51      0.56     9.4
## 6                  13                   40  0.9978 3.51      0.56     9.4
##   quality color
## 1       5   red
## 2       5   red
## 3       5   red
```

```
## 4        6   red
## 5        5   red
## 6        5   red
```

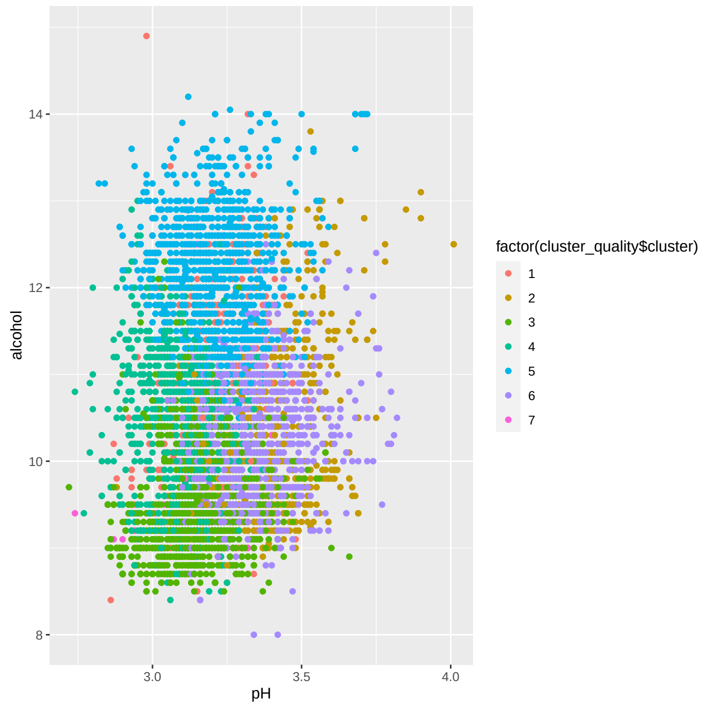


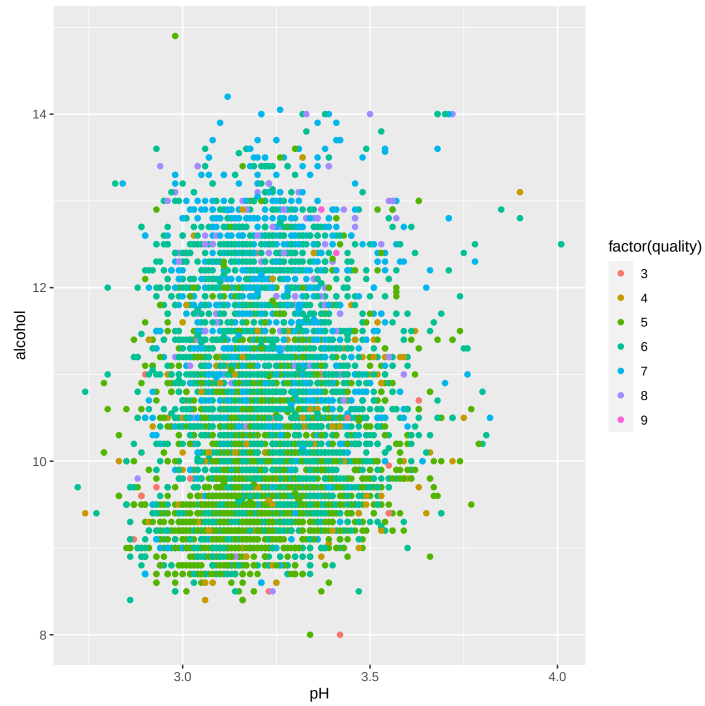
The *color* column has been modified to binary representation. If the wine is red it will be **1** else **0**. Now, since all the columns are numerical a correlation plot has been created to see the correlation between the columns of interest (*color* and *quality*). We notice for *color* column has high positive correlation with *volatile.acidity* and *fixed.acidity*.





The first scatter plot is between *fixed.acidity* and *volatile.acidity* columns from the Wine data. These two columns are selected because they both have a positive correlation with color column. Which means for higher values, the wine is going to be red wine. The colors are based on the clusters each data point is assigned to using K-means clustering. The second plot is between the same attributes but the colors are based on whether the datapoint corresponds to a red wine or a white wine. When we compare the two plots we can see that with a few exceptions, the datapoints have been clustered correctly.





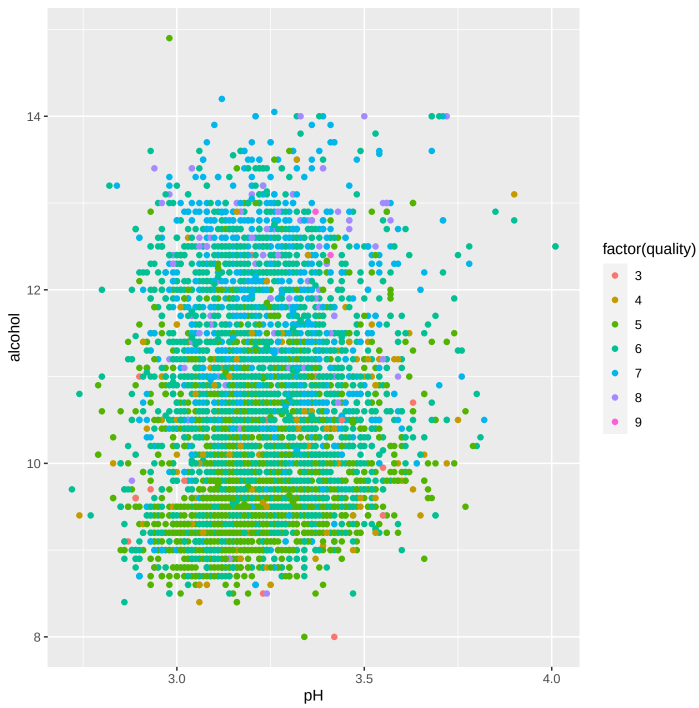
The plots are between *pH* and *alcohol* but in the first plot, the color coding is based on the clusters each of the data points have been assigned to. The color coding the second plot is based on the quality rating of the wine. From the plots we can see that the k-means clustering did not do a good job in clustering the wines to correct qualities. We will use PCA (Principal Components Analysis) to see what components come up.

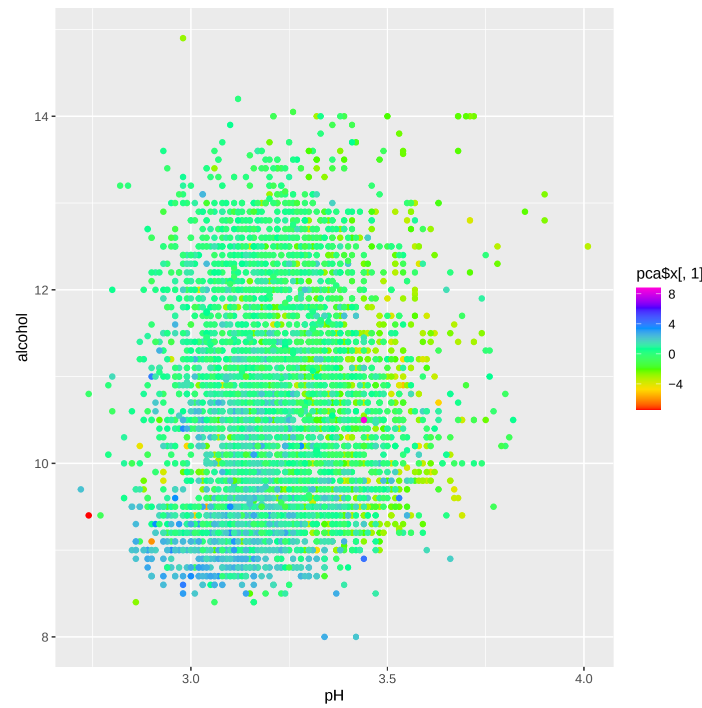
```
## Importance of first k=1 (out of 11) components:
##                           PC1
## Standard deviation     1.7407
## Proportion of Variance 0.2754
## Cumulative Proportion  0.2754
```

```
## [1] "Coefficients of x variables to get Principal Component 1"
```

```
##        fixed.acidity     volatile.acidity          citric.acid
##          -0.23879890          -0.38075750           0.15238844
##       residual.sugar            chlorides  free.sulfur.dioxide
##           0.34591993          -0.29011259           0.43091401
## total.sulfur.dioxide              density                   pH
##           0.48741806          -0.04493664          -0.21868644
##            sulphates              alcohol
##          -0.29413517          -0.10643712
```

We see that from the coefficents of x variables for PC1 it seems like if it is positive it is mostly white wine else it is a red wine. This can be interpreted by comparing the coefficient values and the correlation between color and other features. For example, the coefficient for *total.sulfur.dioxide* is positive and the highest in magnitude. If we see in the correlation matrix plot for *color* and *total.sulfur.dioxide* has a very negative correlation. The direction is opposite as PC1 is measuring the whiteness of the wine whereas the color column used for the correlation matrix is taking red wine as 1. A scatter plot between fixed.acidity and volatile.acidity has been created twice. Once with the color of the points determined by the type of wine column (red or white). In the second graph, the color of the points is determined by the value of PC1. The type of wine is clearly visible in the plot with the PC1 based gradient coloring.





We again plotted two scatter plots but this time it is between *pH* and *alcohol*. The first graph is color coded by the quality rating of the wine. The second graph is gradient color coded based on PC1 value. Compared to K-means clustering the quality of wines is more accurately divided by PCA with one Principal Component.

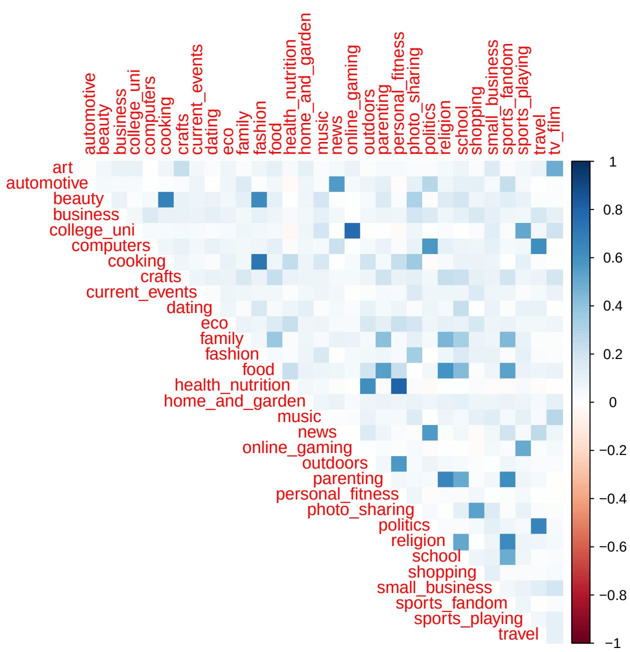
Market Segmentation

```
##            X chatter current_events travel photo_sharing uncategorized tv_film
## 1 hmjoe4g3k       2              0      2             2             2       1
## 2 clk1m5w8s       3              3      2             1             1       1
## 3 jcsovtak3       6              3      4             3             1       5
## 4 3oeb4hiln       1              5      2             2             0       1
## 5 fd75x1vgk       5              2      0             6             1       0
## 6 h6nvj91yp       6              4      2             7             0       1
##   sports_fandom politics food family home_and_garden music news online_gaming
## 1             1        0    4      1               2     0    0             0
## 2             4        1    2      2               1     0    0             0
## 3             0        2    1      1               1     1    1             0
## 4             0        1    0      1               0     0    0             0
## 5             0        2    0      1               0     0    0             3
## 6             1        0    2      1               1     1    0             0
##   shopping health_nutrition college_uni sports_playing cooking eco computers
## 1        1               17           0              2       5   1         1
## 2        0                0           0              1       0   0         0
## 3        2                0           0              0       2   1         0
## 4        0                0           1              0       0   0         0
## 5        2                0           4              0       1   0         1
## 6        5                0           0              0       0   0         1
##   business outdoors crafts automotive art religion beauty parenting dating
## 1        0        2      1          0   0        1      0         1      1
```

```
## 2         1         0      2          0   0         0      0         0      1
## 3         0         0      2          0   8         0      1         0      1
## 4         1         0      3          0   2         0      1         0      0
## 5         0         1      0          0   0         0      0         0      0
## 6         1         0      0          1   0         0      0         0      0
##   school personal_fitness fashion small_business spam adult
## 1      0               11       0              0    0     0
## 2      4                0       0              0    0     0
## 3      0                0       1              0    0     0
## 4      0                0       0              0    0     0
## 5      0                0       0              1    0     0
## 6      0                0       0              0    0     0
```

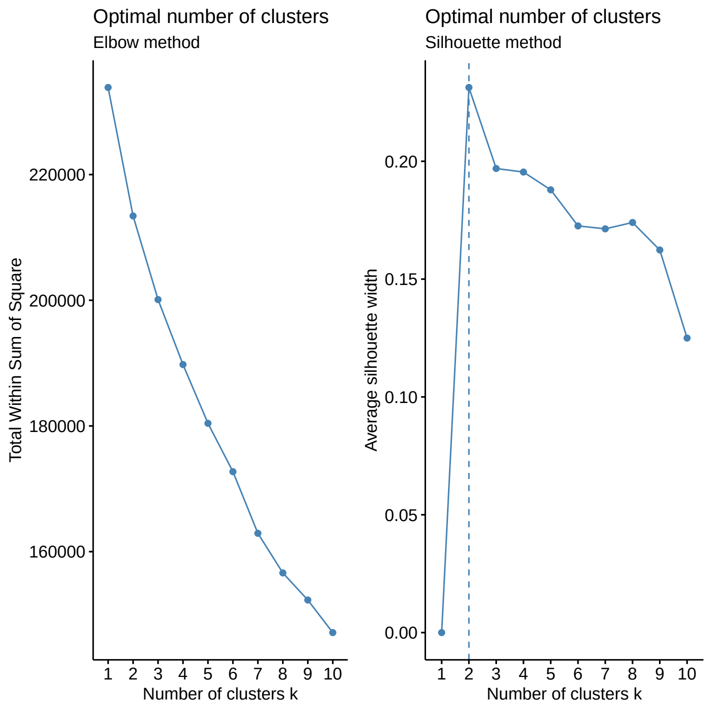
We have dropped the irrelevant columns i.e Chatter and uncategorized since they wouldn't give out any information to make the segment

We have also dropped users who have been identified as spam or adult even once since the question mentions that even though they have filtered out, there could have been some slip through and we don't want these bots to skew our analysis



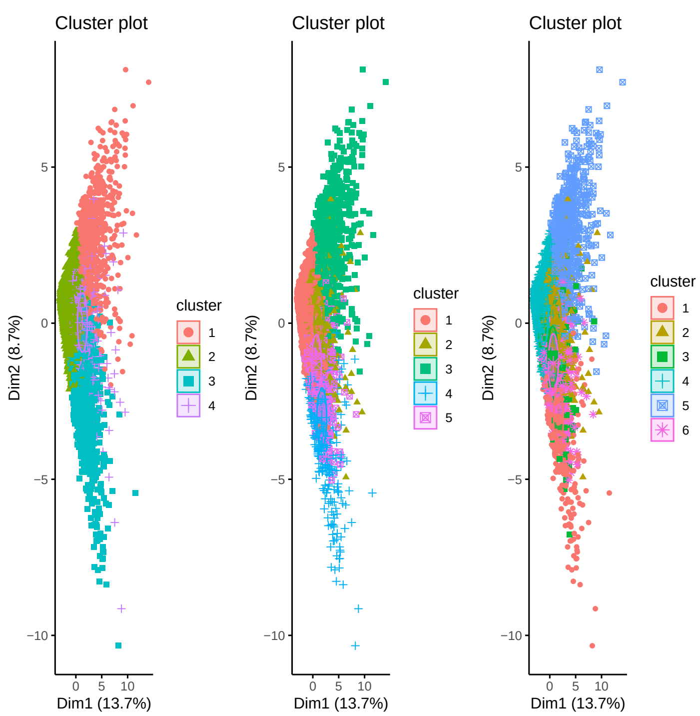
We can see **correlated categories** above. For example, it makes sense that users who are into health nutrition are also into personal fitness. We're sure there are more correlated categories that would become more evident once we perform cluster analysis

We have normalized the data to perform clustering since it's distance based algorithm



It isn't very clear from the elbow plot as to what number of clusters is the most optimal. Silhoutte method recommends two clusters, but for our use case that's too small a number

Forming the clusters



After visually examining the plot above, we notice that the **green squares in the 6 clusters plot don't seem to make a significant cluster**. Even the though the clusters are more evident in the 4 clusters plot, I think it's better to **move forward with**

5 clusters as any insignificant cluster after profiling can be discarded

We assigning features to clusters based on the frequency of posts for a specific category as compared to mean of other clusters

```
##                           k
##  [1,] "travel"           "2"
##  [2,] "tv_film"          "2"
##  [3,] "politics"         "2"
##  [4,] "news"             "2"
##  [5,] "computers"        "2"
##  [6,] "business"         "2"
##  [7,] "automotive"       "2"
##  [8,] "dating"           "2"
##  [9,] "sports_fandom"    "3"
## [10,] "food"             "3"
## [11,] "family"           "3"
## [12,] "home_and_garden"  "3"
## [13,] "crafts"           "3"
## [14,] "religion"         "3"
## [15,] "parenting"        "3"
## [16,] "school"           "3"
## [17,] "current_events"   "4"
## [18,] "photo_sharing"    "4"
## [19,] "music"            "4"
## [20,] "online_gaming"    "4"
## [21,] "shopping"         "4"
## [22,] "college_uni"      "4"
## [23,] "sports_playing"   "4"
## [24,] "cooking"          "4"
## [25,] "art"              "4"
## [26,] "beauty"           "4"
## [27,] "fashion"          "4"
## [28,] "small_business"   "4"
## [29,] "health_nutrition" "5"
## [30,] "eco"              "5"
## [31,] "outdoors"         "5"
## [32,] "personal_fitness" "5"
```

One of clusters got wiped out since it didn't have any significant result

From the output above, we came up with the following the customer profiles

Health Conscious Segment: They seem to be enthusiastic about Health, Nutrition and Fitness
Family Focused Segment: Their interests are usually surrounding schooling, parenting, religion, home and garden
Millenials/ Gen-Zers: They are into things like music, shopping, online shopping, photo sharing, beauty, fashion, college uni, playing sports, start ups, online gaming etc
Knowledgeable Segment: They like tweeting about business, news, politics, automative, travel & tv/film

The Reuters Corpus

Data was received in a format where a train folder contained folders for 50 authors and each author folder contained 50 documents for each author. It also contained a test folder with a similar structure with the same 50 authors and 50 other documents written by each author.

Method

First we read the author folder names and looped through the contents to get the document information and content. This was done for both train and test sets. With this data we made a corpus document, which is a text mining document. We then mapped five content transformers to make everything lowercase and remove numbers, punctuation, excess whitespace and stop words. Next, we created a document term matrix that is a data frame like structure that comprises rows for each document and columns for all terms that appear across documents. Once this was created, sparse terms were removed on the training and test document term matrices using 91% and 96% thresholds respectively (these thresholds are somewhat arbitrary and different values could be used). Finally, we removed terms that occur in the test data but not in the train data.

Training files : + Clean up the file names in training set + Renaming the articles + Creating vector for documents

Testing files : + Clean up the file names in testing set + Renaming the articles + Creating vector for documents

After creating a vector of all documents, we create a corpus for text mining:

Creating a text mining corpus for train and transforming train corpus by:

- Converting all text to lowercase
- Removing all numbers to enable efficient text mining
- Remove all punctuation like ',' '.'
- Remove extra white-spaces
- Removing common stop words as part of list "en"

Creating corpus for test documents

Document-Term Matrix for train and test corpus

```
## [1] "DocumentTermMatrix"    "simple_triplet_matrix"
```

Inspecting the entries of Document Term Matrix for train

```
## <<DocumentTermMatrix (documents: 10, terms: 20)>>
## Non-/sparse entries: 48/152
## Sparsity           : 76%
## Maximal term length: 11
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs access accounts agencies also announced bogus business called character
##   1       1        1        1    1         1     2        2      1         4
##   10      4        0        0    1         0     0        1      0         4
##   2       0        0        0    2         1     0        1      0         4
##   3       2        0        0    0         0     0        0      0         4
##   4       0        0        0    0         1     0        1      0         4
##   5       0        0        0    0         1     0        1      0         4
##   6       0        0        0    0         0     0        0      0         4
##   7       0        0        1    1         0     0        0      1         4
##   8       0        0        0    0         0     0        1      0         4
##   9       0        0        1    0         0     0        1      0         4
##     Terms
## Docs charged
```

```
## 1         1
## 10        0
## 2         0
## 3         0
## 4         0
## 5         0
## 6         0
## 7         0
## 8         1
## 9         1
```

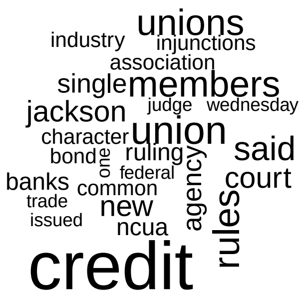
Frequently occurring words (count of atleast 1000) in Train corpus

```
##  [1] "also"          "business"      "character"     "datetimestamp"
##  [5] "description"   "group"         "heading"       "hour"
##  [9] "isdst"         "language"
```

Finding words whose count correlates with a specific word Interesting to observe that human character is depicted by authors as apolitical, rivaling etc with most words negatively portraying their character with each word of association more than 0.5

```
## $character
##      ancestry    apolitical         arrow       catwalk  connotations
##          0.52          0.52          0.52          0.52          0.52
##  cosmopolitan       descend      diaspora       esquire fastemerging
##          0.52          0.52          0.52          0.52          0.52
##     fiftyfour    halftruths         heung   highsociety      kongborn
##          0.52          0.52          0.52          0.52          0.52
##          kuan    limthongul    megatrends      migrated      naisbitt
##          0.52          0.52          0.52          0.52          0.52
##    newsstands     pictogram      piercing      polemics    rediscover
##          0.52          0.52          0.52          0.52          0.52
##     rivalling    sandwiched   sensibility         shing         shuyi
##          0.52          0.52          0.52          0.52          0.52
##     socialite        sondhi thailandbased   upandcoming       visuals
##          0.52          0.52          0.52          0.52          0.52
##         vogue    wellheeled           yew         zhong
##          0.52          0.52          0.52          0.52
```

Building a word cloud to represent the words that have appeared atleast 50 times in the 15th author i.e. Jan Lapotka. The words depit that his writings are around Finance and new practices in banking and trade.




```
## [1] "Document Term Matrix for train"
```

```
## <<DocumentTermMatrix (documents: 2500, terms: 32570)>>
## Non-/sparse entries: 537861/80887139
## Sparsity           : 99%
## Maximal term length: 40
## Weighting          : term frequency (tf)
```

```
## [1] "Document Term Matrix for test"
```

```
## <<DocumentTermMatrix (documents: 2500, terms: 33373)>>
## Non-/sparse entries: 545286/82887214
## Sparsity           : 99%
## Maximal term length: 45
## Weighting          : term frequency (tf)
```

Dropping terms that occur less frequently leading to a long tail of rare terms

Removing words from train and test DT Matrix which have word count as 0 in more than 90% of documents in train and test.

TF-IDF weights

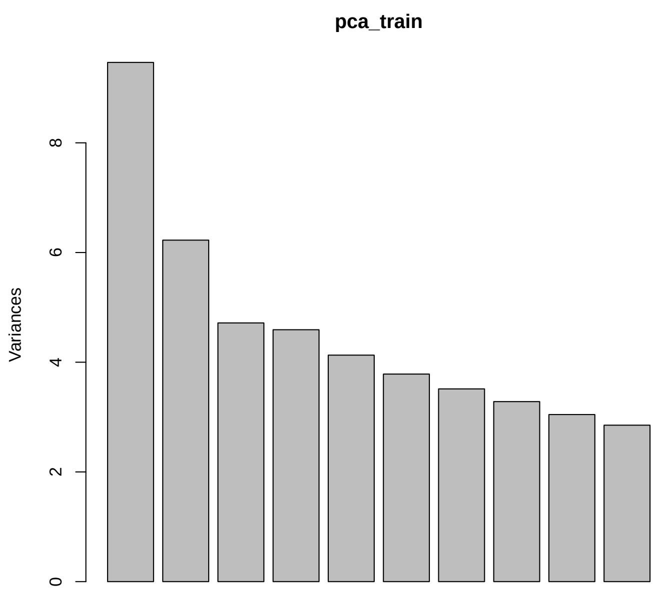
```
## <<DocumentTermMatrix (documents: 2500, terms: 364)>>
## Non-/sparse entries: 163877/746123
## Sparsity           : 82%
## Maximal term length: 14
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

Compare documents with terms

```
## <<DocumentTermMatrix (documents: 1, terms: 364)>>
## Non-/sparse entries: 47/317
## Sparsity           : 87%
## Maximal term length: 14
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
## Sample             :
##     Terms
## Docs commission   computer   consumer       cthe       home  investors
##    1 0.08741916 0.04296506 0.08696628 0.03974902 0.04176277 0.09997993
##     Terms
## Docs     later        may      place technology
##    1 0.0428196 0.06864041 0.04095772 0.04095772
```

Dimensionality Reduction

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   8.806  10.702  11.417  13.243  39.129
```



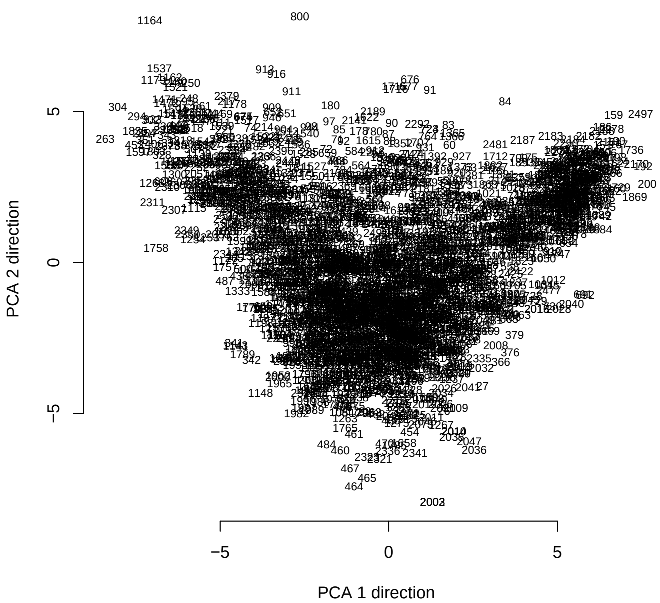
This plot clearly indicates that the first two PCA trains explain maximum variability in the data and hence can be used for our analysis.

Description of Loadings in the PCA train 1 and 2

```
##      china    beijing    chinese      share    million     chinas    quarter
##  0.1712778  0.1698391  0.1565943 -0.1550767 -0.1477759  0.1444136 -0.1425643
##   earnings  analysts     profit    profits    percent    analyst     leader
## -0.1387713 -0.1352297 -0.1326969 -0.1318931 -0.1312685 -0.1257244  0.1226952
##       hong  officials   official       kong        net  political       rose
##  0.1208738  0.1196299  0.1184599  0.1173703 -0.1141872  0.1138774 -0.1124834
## government    results     shares    billion
##  0.1092414 -0.1092371 -0.1065199 -0.1036792
```

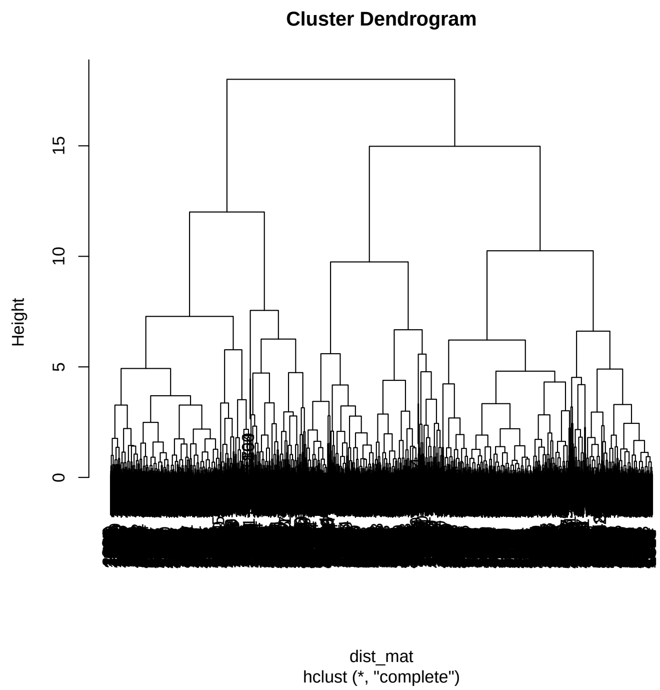
```
##          corp     companies           new communications          deal
##    -0.1507010    -0.1490099    -0.1366834    -0.1358493    -0.1348559
##        chinas         china       beijing       company      forecast
##     0.1311536     0.1305471     0.1302783    -0.1301624     0.1285871
##        profit       chinese          will           inc      services
##     0.1278436     0.1269668    -0.1223854    -0.1220210    -0.1127963
##         offer     customers          half       results       percent
##    -0.1110132    -0.1104692     0.1104282     0.1096930     0.1087669
##      industry          last           net        period          rise
##    -0.1077910     0.1062837     0.1037816     0.1018239     0.1011414
```

Each document into a single pair of numbers – massive dimensionality reduction



Distance Matrix using PCA scores

- Forming hierarchical cluster of size 5 using Euclidean distance




```
##   51   53   54   56   59   60   61   62   63   64   65   67   69   70   71   72
##   51   53   54   56   59   60   61   62   63   64   65   67   69   70   71   72
##   75   76   80   83   84   85   86   87   88   89   90   91   92   94   95   96
##   75   76   80   83   84   85   86   87   88   89   90   91   92   94   95   96
##   97   99  100  139  151  152  153  155  156  157  158  159  161  162  163  164
##   97   99  100  139  151  152  153  155  156  157  158  159  161  162  163  164
##  165  166  167  168  169  170  171  172  173  175  176  177  178  179  180  181
##  165  166  167  168  169  170  171  172  173  175  176  177  178  179  180  181
##  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  198
##  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  198
##  199  200  203  226  232  268  269  314  327  349  414  415  429  444  478  523
##  199  200  203  226  232  268  269  314  327  349  414  415  429  444  478  523
##  551  552  553  554  555  556  561  563  564  567  568  571  572  576  578  579
##  551  552  553  554  555  556  561  563  564  567  568  571  572  576  578  579
##  584  587  588  589  590  593  594  595  596  597  601  621  653  654  655  657
##  584  587  588  589  590  593  594  595  596  597  601  621  653  654  655  657
##  659  662  663  669  670  671  673  676  677  678  683  684  689  693  694  695
##  659  662  663  669  670  671  673  676  677  678  683  684  689  693  694  695
##  696  698  702  703  704  706  707  708  709  711  712  713  714  716  717  718
##  696  698  702  703  704  706  707  708  709  711  712  713  714  716  717  718
##  719  720  722  723  724  725  726  727  731  732  733  734  735  736  737  739
##  719  720  722  723  724  725  726  727  731  732  733  734  735  736  737  739
##  740  741  742  743  744  745  746  747  749  750  753  754  755  756  757  761
##  740  741  742  743  744  745  746  747  749  750  753  754  755  756  757  761
##  762  763  764  765  766  767  768  771  772  776  779  780  781  782  783  784
##  762  763  764  765  766  767  768  771  772  776  779  780  781  782  783  784
##  785  792  794  796  860  864  901  902  906  908  910  912  920  921  922  923
##  785  792  794  796  860  864  901  902  906  908  910  912  920  921  922  923
##  925  926  927  928  929  930  931  933  934  936  937  938  939  940  942  943
##  925  926  927  928  929  930  931  933  934  936  937  938  939  940  942  943
##  954  978  979  992  993  999 1007 1008 1010 1011 1013 1017 1021 1025 1026 1027
```

```
##  954  978  979  992  993  999 1007 1008 1010 1011 1013 1017 1021 1025 1026 1027
## 1032 1035 1042 1043 1047 1049 1050 1051 1127 1181 1185 1188 1189 1305 1307 1309
## 1032 1035 1042 1043 1047 1049 1050 1051 1127 1181 1185 1188 1189 1305 1307 1309
## 1326 1328 1338 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363
## 1326 1328 1338 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363
## 1364 1365 1366 1367 1368 1371 1372 1373 1374 1375 1376 1377 1381 1382 1383 1384
## 1364 1365 1366 1367 1368 1371 1372 1373 1374 1375 1376 1377 1381 1382 1383 1384
## 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400
## 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400
## 1401 1402 1403 1404 1405 1413 1417 1420 1422 1433 1438 1443 1445 1446 1449 1458
## 1401 1402 1403 1404 1405 1413 1417 1420 1422 1433 1438 1443 1445 1446 1449 1458
## 1483 1486 1503 1509 1513 1524 1525 1544 1567 1568 1601 1602 1603 1604 1605 1606
## 1483 1486 1503 1509 1513 1524 1525 1544 1567 1568 1601 1602 1603 1604 1605 1606
## 1608 1610 1612 1613 1614 1615 1616 1617 1619 1621 1622 1623 1624 1625 1627 1628
## 1608 1610 1612 1613 1614 1615 1616 1617 1619 1621 1622 1623 1624 1625 1627 1628
## 1629 1630 1631 1633 1634 1635 1637 1641 1642 1643 1647 1648 1650 1685 1701 1702
## 1629 1630 1631 1633 1634 1635 1637 1641 1642 1643 1647 1648 1650 1685 1701 1702
## 1703 1704 1706 1707 1709 1710 1711 1712 1715 1716 1718 1719 1720 1721 1722 1723
## 1703 1704 1706 1707 1709 1710 1711 1712 1715 1716 1718 1719 1720 1721 1722 1723
## 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739
## 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739
## 1741 1742 1743 1744 1746 1747 1748 1749 1750 1780 1852 1853 1854 1855 1856 1857
## 1741 1742 1743 1744 1746 1747 1748 1749 1750 1780 1852 1853 1854 1855 1856 1857
## 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1869 1870 1871 1872 1873 1874
## 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1869 1870 1871 1872 1873 1874
## 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890
## 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890
## 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1920 2103 2112 2114 2119 2120
## 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1920 2103 2112 2114 2119 2120
## 2121 2122 2124 2125 2128 2129 2132 2133 2135 2136 2138 2139 2140 2141 2142 2145
## 2121 2122 2124 2125 2128 2129 2132 2133 2135 2136 2138 2139 2140 2141 2142 2145
## 2148 2149 2150 2151 2152 2153 2154 2155 2156 2158 2159 2160 2161 2162 2163 2164
## 2148 2149 2150 2151 2152 2153 2154 2155 2156 2158 2159 2160 2161 2162 2163 2164
## 2165 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2180 2181 2182 2183 2184
## 2165 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2180 2181 2182 2183 2184
## 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2198 2199 2200 2251
## 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2198 2199 2200 2251
## 2252 2253 2254 2255 2256 2257 2258 2259 2262 2263 2264 2267 2268 2269 2270 2271
## 2252 2253 2254 2255 2256 2257 2258 2259 2262 2263 2264 2267 2268 2269 2270 2271
## 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2287 2288
## 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2287 2288
## 2289 2290 2291 2292 2294 2295 2296 2297 2298 2299 2300 2380 2384 2409 2410 2430
## 2289 2290 2291 2292 2294 2295 2296 2297 2298 2299 2300 2380 2384 2409 2410 2430
## 2451 2452 2458 2460 2463 2464 2465 2468 2470 2471 2472 2473 2474 2475 2476 2478
## 2451 2452 2458 2460 2463 2464 2465 2468 2470 2471 2472 2473 2474 2475 2476 2478
## 2479 2480 2481 2482 2483 2485 2486 2487 2488 2490 2491 2493 2494 2495 2497 2498
## 2479 2480 2481 2482 2483 2485 2486 2487 2488 2490 2491 2493 2494 2495 2497 2498
## 2499 2500
## 2499 2500
```

Converting to dataframe

After removing sparseness

```
## [1] "Dataframe of train"
```

```
## [1] 2500  364
```

```
## [1] "Dataframe of test"
```

```
## [1] 2500  363
```

Models

Naive Bayes Classifier

First we ran Naive-Bayes to predict authors. The data was changed to matrices to run the Naive-Bayes library and had predictive accuracy of approximately 33.28% . This uses the naivebayes library in R.

```
## [1] 0.0204
```

Random Forest Classifier

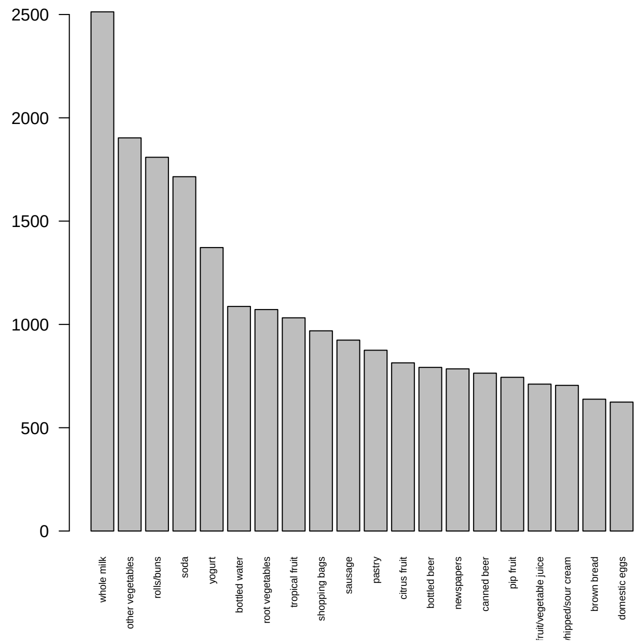
Secondly we ran a Random Forest model with 1,250 trees and an m of 20. To run this in R, we had to change some terms in the train and test data sets so they did not conflict with special terms in R. The Random Forest model had predictive accuracy of approximately 54.12% . This uses the randomForest library in R.

```
##           1             2             3             4             5             6 
##  ronPressman    rlPenhaul eresePoletti  ronPressman    rlPenhaul  ronPressman 
## 50 Levels: adDorfman ahamEarnshaw anCrosby atherScoffield ... Winterbottom
```

```
## [1] 0.0228
```

Association Rule Mining

```
## 
## abrasive cleaner artif. sweetener   baby cosmetics        baby food 
##               35               32                6                1 
##             bags    baking powder 
##                4              174
```



```
## transactions as itemMatrix in sparse format with
##  15296 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.01677625
##
## most frequent items:
##       whole milk other vegetables       rolls/buns             soda
##             2513             1903             1809             1715
##           yogurt          (Other)
##             1372            34055
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4
## 3485 2630 2102 7079
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   2.835   4.000   4.000
##
## includes extended item information - examples:
##             labels
## 1 abrasive cleaner
## 2 artif. sweetener
## 3   baby cosmetics
##
## includes extended transaction information - examples:
##   transactionID
## 1             1
## 2             2
## 3             3
```

```
## Apriori
```

```
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5   0.005      1
##  maxlen target  ext
##       5  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 76
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 15296 transaction(s)] done [0.00s].
## sorting and recoding items ... [101 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [118 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
##       lhs                                    rhs                  support
## [1]   {}                                  => {soda}               0.112120816
## [2]   {}                                  => {rolls/buns}         0.118266213
## [3]   {}                                  => {other vegetables}   0.124411611
## [4]   {}                                  => {whole milk}         0.164291318
## [5]   {butter milk}                       => {whole milk}         0.005033996
## [6]   {onions}                            => {root vegetables}    0.005295502
## [7]   {onions}                            => {other vegetables}   0.007452929
## [8]   {onions}                            => {whole milk}         0.005360879
## [9]   {berries}                           => {other vegetables}   0.005164749
## [10]  {berries}                           => {whole milk}         0.005230126
## [11]  {hamburger meat}                    => {other vegetables}   0.006210774
## [12]  {hamburger meat}                    => {whole milk}         0.005818515
## [13]  {dessert}                           => {whole milk}         0.006603033
## [14]  {cream cheese }                     => {yogurt}             0.005033996
## [15]  {chocolate}                         => {soda}               0.005360879
## [16]  {chicken}                           => {other vegetables}   0.007975941
## [17]  {chicken}                           => {whole milk}         0.006341527
## [18]  {frozen vegetables}                 => {whole milk}         0.005687762
## [19]  {canned beer}                       => {soda}               0.006537657
## [20]  {beef}                              => {citrus fruit}       0.005099372
## [21]  {beef}                              => {root vegetables}    0.008695084
## [22]  {root vegetables}                   => {beef}               0.008695084
## [23]  {beef}                              => {other vegetables}   0.008302824
## [24]  {beef}                              => {whole milk}         0.008172071
## [25]  {curd}                              => {yogurt}             0.007649059
## [26]  {curd}                              => {other vegetables}   0.007060669
## [27]  {curd}                              => {whole milk}         0.012617678
## [28]  {margarine}                         => {rolls/buns}         0.005491632
## [29]  {butter}                            => {yogurt}             0.006210774
## [30]  {butter}                            => {other vegetables}   0.008237448
## [31]  {butter}                            => {whole milk}         0.014382845
## [32]  {pork}                              => {root vegetables}    0.006733787
```

```
## [33]  {pork}                       => {other vegetables}   0.009283473
## [34]  {pork}                       => {whole milk}         0.008695084
## [35]  {frankfurter}                => {sausage}            0.006472280
## [36]  {sausage}                    => {frankfurter}        0.006472280
## [37]  {frankfurter}                => {tropical fruit}     0.005557008
## [38]  {frankfurter}                => {rolls/buns}         0.006210774
## [39]  {frankfurter}                => {other vegetables}   0.007060669
## [40]  {frankfurter}                => {whole milk}         0.008237448
## [41]  {bottled beer}               => {bottled water}      0.006929916
## [42]  {bottled beer}               => {soda}               0.008302824
## [43]  {brown bread}                => {pastry}             0.005033996
## [44]  {brown bread}                => {rolls/buns}         0.006276151
## [45]  {brown bread}                => {whole milk}         0.006733787
## [46]  {domestic eggs}              => {rolls/buns}         0.008172071
## [47]  {domestic eggs}              => {whole milk}         0.008172071
## [48]  {fruit/vegetable juice}      => {bottled water}      0.005753138
## [49]  {fruit/vegetable juice}      => {soda}               0.009348849
## [50]  {shopping bags}              => {soda}               0.006406904
## [51]  {whipped/sour cream}         => {yogurt}             0.009741109
## [52]  {yogurt}                     => {whipped/sour cream} 0.009741109
## [53]  {whipped/sour cream}         => {rolls/buns}         0.005360879
## [54]  {whipped/sour cream}         => {other vegetables}   0.008302824
## [55]  {whipped/sour cream}         => {whole milk}         0.011440900
## [56]  {pip fruit}                  => {citrus fruit}       0.008172071
## [57]  {citrus fruit}               => {pip fruit}          0.008172071
## [58]  {pip fruit}                  => {sausage}            0.006210774
## [59]  {sausage}                    => {pip fruit}          0.006210774
## [60]  {pip fruit}                  => {tropical fruit}     0.012683054
## [61]  {tropical fruit}             => {pip fruit}          0.012683054
## [62]  {pip fruit}                  => {root vegetables}    0.008106695
## [63]  {root vegetables}            => {pip fruit}          0.008106695
## [64]  {pip fruit}                  => {other vegetables}   0.010917887
## [65]  {pip fruit}                  => {whole milk}         0.012552301
## [66]  {pastry}                     => {soda}               0.007256799
## [67]  {pastry}                     => {rolls/buns}         0.010198745
## [68]  {pastry}                     => {whole milk}         0.009414226
## [69]  {citrus fruit}               => {sausage}            0.006929916
## [70]  {sausage}                    => {citrus fruit}       0.006929916
## [71]  {citrus fruit}               => {tropical fruit}     0.012486925
## [72]  {tropical fruit}             => {citrus fruit}       0.012486925
## [73]  {citrus fruit}               => {root vegetables}    0.008695084
## [74]  {root vegetables}            => {citrus fruit}       0.008695084
## [75]  {citrus fruit}               => {yogurt}             0.006733787
## [76]  {citrus fruit}               => {other vegetables}   0.012813808
## [77]  {other vegetables}           => {citrus fruit}       0.012813808
## [78]  {citrus fruit}               => {whole milk}         0.012813808
## [79]  {sausage}                    => {tropical fruit}     0.008172071
## [80]  {tropical fruit}             => {sausage}            0.008172071
## [81]  {sausage}                    => {root vegetables}    0.007322176
## [82]  {root vegetables}            => {sausage}            0.007322176
## [83]  {sausage}                    => {rolls/buns}         0.010787134
## [84]  {sausage}                    => {other vegetables}   0.012617678
## [85]  {other vegetables}           => {sausage}            0.012617678
## [86]  {sausage}                    => {whole milk}         0.012552301
```

```
## [87]  {bottled water}                      => {soda}              0.014644351
## [88]  {soda}                               => {bottled water}     0.014644351
## [89]  {bottled water}                      => {rolls/buns}        0.008564331
## [90]  {tropical fruit}                     => {root vegetables}   0.010983264
## [91]  {root vegetables}                    => {tropical fruit}    0.010983264
## [92]  {tropical fruit}                     => {yogurt}            0.008172071
## [93]  {tropical fruit}                     => {other vegetables}  0.015494247
## [94]  {other vegetables}                   => {tropical fruit}    0.015494247
## [95]  {tropical fruit}                     => {whole milk}        0.018305439
## [96]  {whole milk}                         => {tropical fruit}    0.018305439
## [97]  {root vegetables}                    => {other vegetables}  0.025366109
## [98]  {other vegetables}                   => {root vegetables}   0.025366109
## [99]  {root vegetables}                    => {whole milk}        0.022620293
## [100] {whole milk}                         => {root vegetables}   0.022620293
## [101] {yogurt}                             => {rolls/buns}        0.011898536
## [102] {rolls/buns}                         => {yogurt}            0.011898536
## [103] {yogurt}                             => {other vegetables}  0.015886506
## [104] {other vegetables}                   => {yogurt}            0.015886506
## [105] {yogurt}                             => {whole milk}        0.024254707
## [106] {whole milk}                         => {yogurt}            0.024254707
## [107] {soda}                               => {rolls/buns}        0.014252092
## [108] {rolls/buns}                         => {soda}              0.014252092
## [109] {rolls/buns}                         => {whole milk}        0.018305439
## [110] {whole milk}                         => {rolls/buns}        0.018305439
## [111] {other vegetables}                   => {whole milk}        0.040860356
## [112] {whole milk}                         => {other vegetables}  0.040860356
## [113] {other vegetables, root vegetables} => {whole milk}        0.008172071
## [114] {root vegetables, whole milk}        => {other vegetables}  0.008172071
## [115] {other vegetables, whole milk}       => {root vegetables}   0.008172071
## [116] {other vegetables, yogurt}           => {whole milk}        0.006341527
## [117] {whole milk, yogurt}                 => {other vegetables}  0.006341527
## [118] {other vegetables, whole milk}       => {yogurt}            0.006341527
##       confidence coverage   lift      count
## [1]   0.1121208  1.00000000 1.0000000 1715
## [2]   0.1182662  1.00000000 1.0000000 1809
## [3]   0.1244116  1.00000000 1.0000000 1903
## [4]   0.1642913  1.00000000 1.0000000 2513
## [5]   0.2800000  0.01797856 1.7042897   77
## [6]   0.2655738  0.01993985 3.7893810   81
## [7]   0.3737705  0.01993985 3.0043055  114
## [8]   0.2688525  0.01993985 1.6364374   82
## [9]   0.2415902  0.02137814 1.9418623   79
## [10]  0.2446483  0.02137814 1.4891129   80
## [11]  0.2905199  0.02137814 2.3351508   95
## [12]  0.2721713  0.02137814 1.6566381   89
## [13]  0.2767123  0.02386245 1.6842785  101
## [14]  0.1974359  0.02549686 2.2011512   77
## [15]  0.1680328  0.03190377 1.4986761   82
## [16]  0.2890995  0.02758891 2.3237343  122
## [17]  0.2298578  0.02758891 1.3990868   97
## [18]  0.1839323  0.03092312 1.1195500   87
## [19]  0.1308901  0.04994770 1.1674019  100
## [20]  0.1511628  0.03373431 2.8405234   78
## [21]  0.2577519  0.03373431 3.6777739  133
```

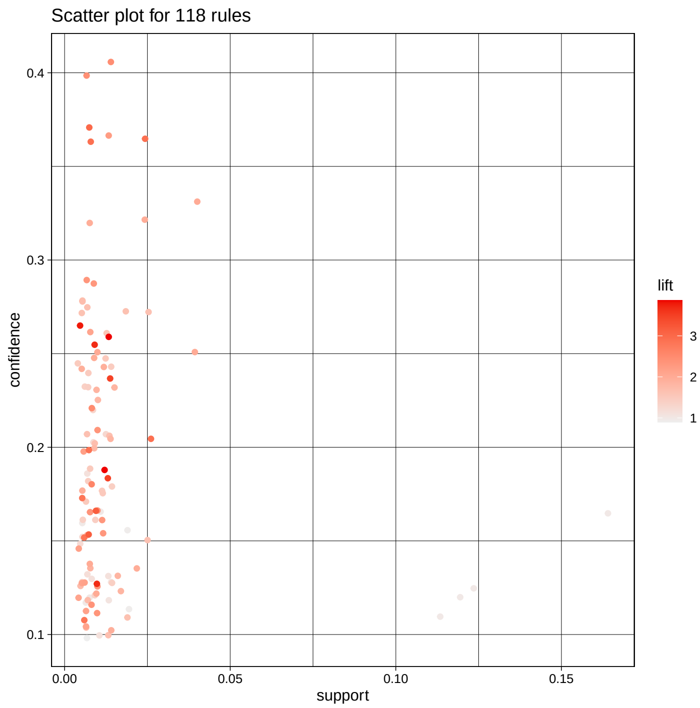
```
## [22]  0.1240672  0.07008368 3.6777739  133
## [23]  0.2461240  0.03373431 1.9783044  127
## [24]  0.2422481  0.03373431 1.4745031  125
## [25]  0.2232824  0.03425732 2.4893063  117
## [26]  0.2061069  0.03425732 1.6566530  108
## [27]  0.3683206  0.03425732 2.2418751  193
## [28]  0.1458333  0.03765690 1.2330938   84
## [29]  0.1743119  0.03563023 1.9433493   95
## [30]  0.2311927  0.03563023 1.8582885  126
## [31]  0.4036697  0.03563023 2.4570363  220
## [32]  0.1816578  0.03706851 2.5920135  103
## [33]  0.2504409  0.03706851 2.0130028  142
## [34]  0.2345679  0.03706851 1.4277559  133
## [35]  0.1706897  0.03791841 2.8256158   99
## [36]  0.1071429  0.06040795 2.8256158   99
## [37]  0.1465517  0.03791841 2.1721465   85
## [38]  0.1637931  0.03791841 1.3849526   95
## [39]  0.1862069  0.03791841 1.4967003  108
## [40]  0.2172414  0.03791841 1.3222937  126
## [41]  0.1338384  0.05177824 1.8833412  106
## [42]  0.1603535  0.05177824 1.4301852  127
## [43]  0.1206897  0.04171025 2.1097931   77
## [44]  0.1504702  0.04171025 1.2723010   96
## [45]  0.1614420  0.04171025 0.9826570  103
## [46]  0.2003205  0.04079498 1.6938102  125
## [47]  0.2003205  0.04079498 1.2193007  125
## [48]  0.1237693  0.04648274 1.7416521   88
## [49]  0.2011252  0.04648274 1.7938255  143
## [50]  0.1011352  0.06334990 0.9020198   98
## [51]  0.2113475  0.04609048 2.3562475  149
## [52]  0.1086006  0.08969665 2.3562475  149
## [53]  0.1163121  0.04609048 0.9834766   82
## [54]  0.1801418  0.04609048 1.4479504  127
## [55]  0.2482270  0.04609048 1.5108951  175
## [56]  0.1680108  0.04864017 3.1571161  125
## [57]  0.1535627  0.05321653 3.1571161  125
## [58]  0.1276882  0.04864017 2.1137644   95
## [59]  0.1028139  0.06040795 2.1137644   95
## [60]  0.2607527  0.04864017 3.8647995  194
## [61]  0.1879845  0.06746862 3.8647995  194
## [62]  0.1666667  0.04864017 2.3781095  124
## [63]  0.1156716  0.07008368 2.3781095  124
## [64]  0.2244624  0.04864017 1.8041915  167
## [65]  0.2580645  0.04864017 1.5707739  192
## [66]  0.1268571  0.05720450 1.1314326  111
## [67]  0.1782857  0.05720450 1.5074949  156
## [68]  0.1645714  0.05720450 1.0017050  144
## [69]  0.1302211  0.05321653 2.1556952  106
## [70]  0.1147186  0.06040795 2.1556952  106
## [71]  0.2346437  0.05321653 3.4778203  191
## [72]  0.1850775  0.06746862 3.4778203  191
## [73]  0.1633907  0.05321653 2.3313653  133
## [74]  0.1240672  0.07008368 2.3313653  133
## [75]  0.1265356  0.05321653 1.4107062  103
```

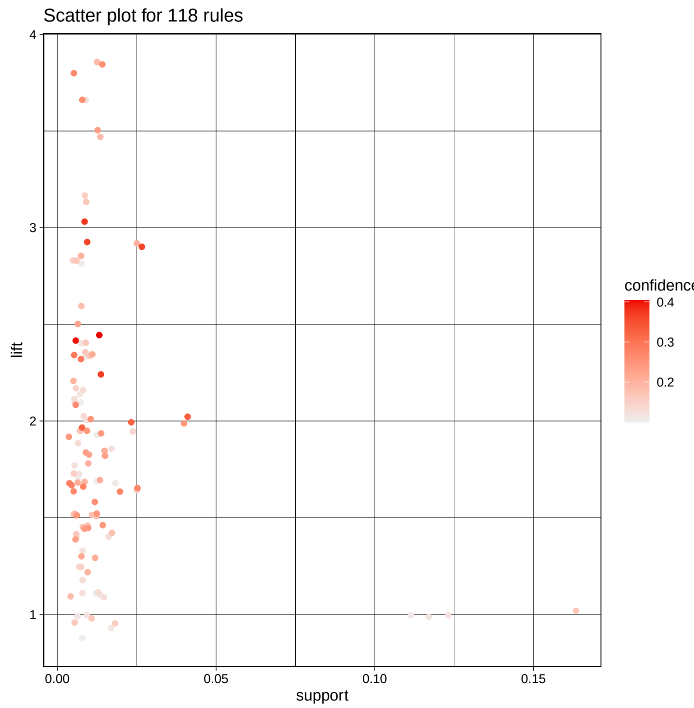
```
## [76]  0.2407862  0.05321653 1.9354001  196
## [77]  0.1029953  0.12441161 1.9354001  196
## [78]  0.2407862  0.05321653 1.4656054  196
## [79]  0.1352814  0.06040795 2.0051008  125
## [80]  0.1211240  0.06746862 2.0051008  125
## [81]  0.1212121  0.06040795 1.7295341  112
## [82]  0.1044776  0.07008368 1.7295341  112
## [83]  0.1785714  0.06040795 1.5099108  165
## [84]  0.2088745  0.06040795 1.6788984  193
## [85]  0.1014188  0.12441161 1.6788984  193
## [86]  0.2077922  0.06040795 1.2647790  192
## [87]  0.2060718  0.07106433 1.8379438  224
## [88]  0.1306122  0.11212082 1.8379438  224
## [89]  0.1205152  0.07106433 1.0190161  131
## [90]  0.1627907  0.06746862 2.3228046  168
## [91]  0.1567164  0.07008368 2.3228046  168
## [92]  0.1211240  0.06746862 1.3503740  125
## [93]  0.2296512  0.06746862 1.8458982  237
## [94]  0.1245402  0.12441161 1.8458982  237
## [95]  0.2713178  0.06746862 1.6514435  280
## [96]  0.1114206  0.16429132 1.6514435  280
## [97]  0.3619403  0.07008368 2.9092164  388
## [98]  0.2038886  0.12441161 2.9092164  388
## [99]  0.3227612  0.07008368 1.9645663  346
## [100] 0.1376840  0.16429132 1.9645663  346
## [101] 0.1326531  0.08969665 1.1216480  182
## [102] 0.1006081  0.11826621 1.1216480  182
## [103] 0.1771137  0.08969665 1.4236107  243
## [104] 0.1276931  0.12441161 1.4236107  243
## [105] 0.2704082  0.08969665 1.6459066  371
## [106] 0.1476323  0.16429132 1.6459066  371
## [107] 0.1271137  0.11212082 1.0748099  218
## [108] 0.1205086  0.11826621 1.0748099  218
## [109] 0.1547816  0.11826621 0.9421170  280
## [110] 0.1114206  0.16429132 0.9421170  280
## [111] 0.3284288  0.12441161 1.9990636  625
## [112] 0.2487067  0.16429132 1.9990636  625
## [113] 0.3221649  0.02536611 1.9609371  125
## [114] 0.3612717  0.02262029 2.9038421  125
## [115] 0.2000000  0.04086036 2.8537313  125
## [116] 0.3991770  0.01588651 2.4296899   97
## [117] 0.2614555  0.02425471 2.1015364   97
## [118] 0.1552000  0.04086036 1.7302764   97

##     lhs                   rhs                support     confidence coverage
## [1] {onions}           => {root vegetables} 0.005295502 0.2655738  0.01993985
## [2] {beef}             => {root vegetables} 0.008695084 0.2577519  0.03373431
## [3] {root vegetables} => {beef}             0.008695084 0.1240672  0.07008368
## [4] {pip fruit}        => {tropical fruit}  0.012683054 0.2607527  0.04864017
## [5] {tropical fruit}  => {pip fruit}        0.012683054 0.1879845  0.06746862
##     lift     count
## [1] 3.789381  81
## [2] 3.677774 133
## [3] 3.677774 133
```

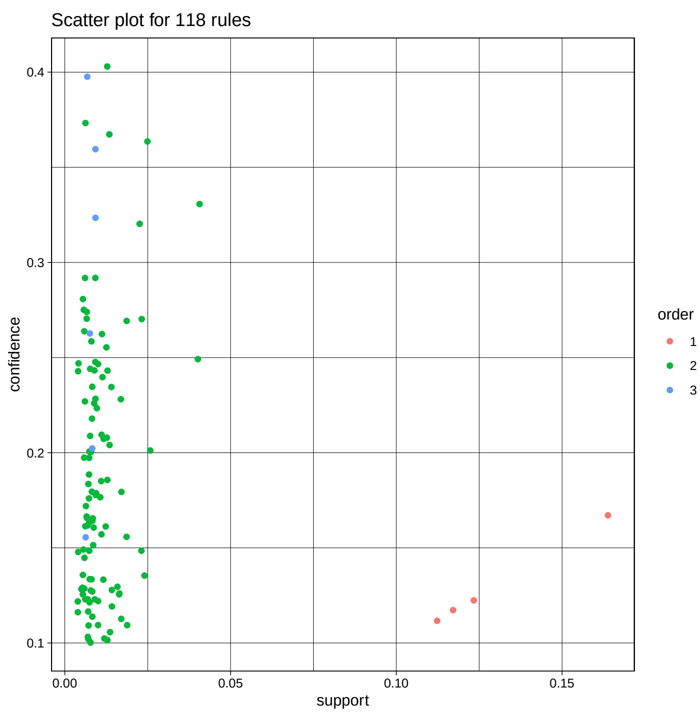
```
## [4] 3.864800 194
## [5] 3.864800 194
```

```
##     lhs                                   rhs                support
## [1] {onions}                           => {other vegetables} 0.007452929
## [2] {curd}                             => {whole milk}       0.012617678
## [3] {butter}                           => {whole milk}       0.014382845
## [4] {root vegetables}                  => {other vegetables} 0.025366109
## [5] {root vegetables}                  => {whole milk}       0.022620293
## [6] {other vegetables}                 => {whole milk}       0.040860356
## [7] {other vegetables, root vegetables} => {whole milk}       0.008172071
## [8] {root vegetables, whole milk}      => {other vegetables} 0.008172071
## [9] {other vegetables, yogurt}         => {whole milk}       0.006341527
##     confidence coverage   lift     count
## [1] 0.3737705  0.01993985 3.004306 114
## [2] 0.3683206  0.03425732 2.241875 193
## [3] 0.4036697  0.03563023 2.457036 220
## [4] 0.3619403  0.07008368 2.909216 388
## [5] 0.3227612  0.07008368 1.964566 346
## [6] 0.3284288  0.12441161 1.999064 625
## [7] 0.3221649  0.02536611 1.960937 125
## [8] 0.3612717  0.02262029 2.903842 125
## [9] 0.3991770  0.01588651 2.429690  97
```

```
##     lhs                            rhs                support     confidence
## [1] {onions}                    => {other vegetables} 0.007452929 0.3737705
## [2] {root vegetables}           => {other vegetables} 0.025366109 0.3619403
## [3] {root vegetables, whole milk} => {other vegetables} 0.008172071 0.3612717
##     coverage   lift     count
## [1] 0.01993985 3.004306 114
## [2] 0.07008368 2.909216 388
## [3] 0.02262029 2.903842 125
```



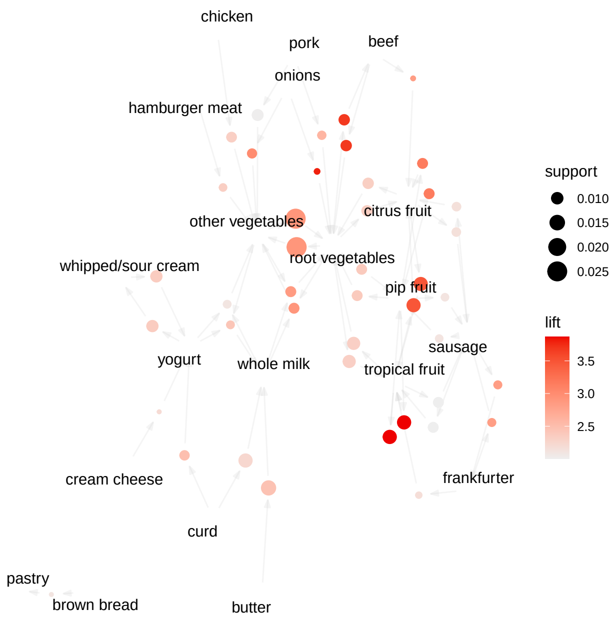





```
##     lhs                    rhs                  support    confidence coverage
## [1] {}                  => {soda}               0.11212082 0.1121208  1.0000000
## [2] {}                  => {rolls/buns}         0.11826621 0.1182662  1.0000000
## [3] {}                  => {other vegetables}   0.12441161 0.1244116  1.0000000
## [4] {}                  => {whole milk}         0.16429132 0.1642913  1.0000000
## [5] {other vegetables} => {whole milk}          0.04086036 0.3284288  0.1244116
## [6] {whole milk}        => {other vegetables}   0.04086036 0.2487067  0.1642913
##     lift      count
```

```
## [1] 1.000000 1715
## [2] 1.000000 1809
## [3] 1.000000 1903
## [4] 1.000000 2513
## [5] 1.999064  625
## [6] 1.999064  625
```

```
## set of 43 rules
##
## rule length distribution (lhs + rhs):sizes
##  2  3
## 39  4
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   2.000   2.093   2.000   3.000
##
## summary of quality measures:
##     support           confidence        coverage            lift
##  Min.   :0.005034   Min.   :0.1028   Min.   :0.01589   Min.   :2.005
##  1st Qu.:0.006472   1st Qu.:0.1328   1st Qu.:0.03426   1st Qu.:2.222
##  Median :0.008172   Median :0.1817   Median :0.04864   Median :2.378
##  Mean   :0.009134   Mean   :0.2056   Mean   :0.04909   Mean   :2.651
##  3rd Qu.:0.009741   3rd Qu.:0.2593   3rd Qu.:0.06394   3rd Qu.:2.909
##  Max.   :0.025366   Max.   :0.4037   Max.   :0.12441   Max.   :3.865
##      count
##  Min.   : 77.0
##  1st Qu.: 99.0
##  Median :125.0
##  Mean   :139.7
##  3rd Qu.:149.0
##  Max.   :388.0
##
## mining info:
##        data ntransactions support confidence
##  grocstrans         15296   0.005        0.1
##                                                                                      call
##  apriori(data = grocstrans, parameter = list(support = 0.005, confidence = 0.1, maxlen = 5))
```



From the bar graph we see the **top 20 most purchased grocery items**. These are common goods that people buy on a regular basis like **milk, vegetables, soda, water, and fruit**.

When looking at the subset of the **association rules where lift > 3.5**, we see that there are only **five pairs of items**, which makes sense because we are picking a high threshold for lift. These are pairs where there is a strong "relationship" between the LHS and the RHS. This means that buyers are much more likely to buy the RHS items if they buy the LHS items. So the five pairs filtered are **(Onions, root vegetables), (beef, root vegetables), (root vegetables, beef), (pip fruit, tropical fruit), (tropical fruit, pip fruit)**. Out of these 5, **two are duplicates**. So there are only **three pairs of items that have a lift greater than 3.5**. When looking at the subset where **confidence is greater than 0.3**, we see that there are **9 pairs of items**. It's interesting to note that for each of these pairs, the **confidence is greater than the support**. In other words, these **pairs are more like complementary goods**, as opposed to the example we looked at in class where coffee and tea are substitutes. When looking at a **combination of lift greater than 2.5 and confidence greater than 0.3, we see that for all of them support is low, confidence is high and lift is high**.

Futhermore, they are all pointing to the same item, other vegetables. When plotting the rules, we see that high lift rules usually have low support. The two-key plot illustrates that **order 3 rules tend to have high confidence**. This makes sense because these are probably **popular grocery items** that people purchase other things with. When looking at pairs of items where **support is greater than 0.035, we see only six pairs**. **4 of them have empty LHS's**. **For the remaining two**, confidence is higher than support and lift is at about 2, indicating these are **complementary goods**. And these two are actually **duplicate pairs**. However, it's interesting that even though their support is the same, they have slightly different confidence values. **From the plot of the subset where confidence > 0.02, support > 0.005, and lift > 2, we can see that the different kind of meats are clustered together, same thing with dairy products, and vegetables**.