STA 380 Assignment Part 2

Rohitashwa Chakraborty, Meha Mehta, Dipali Pandey, Sahitya Sundar Raj Vijayanagar

08-16-2021

Github link: https://github.com/rohitashwachaks/sta380-part2/blob/main/data/STA%20380%20Assignment%20Part%202.Rmd

Visual story telling part 1: green buildings

No we do not agree with the inference of the stat-guru.

While the guru's explanation sounds logical, the guru relies heavily on the assumption that the *green rating* of a building is the sole driver behind the rent of a property.

A simple linear regression run on the rent premium against green building indicates that it is statistically very significant.

```
##                Estimate Std. Error  t value     Pr(>|t|)
## (Intercept)   0.7119434  0.1156967 6.153532 7.947381e-10
## green_rating1 2.4124581  0.3927573 6.142363 8.524150e-10
```

Coefficient of Green_Rating: 2.4125; P-Value: 8e-10

However, our model suffers from an extremely high bias and has a very low R^2. This is a clear indicator that our model is inadequate.

Thus we try looking for confounding variables.

```
## 1 ) MLR, green-building rating with CS_PropertyID : Non Confounding
## 2 ) MLR, green-building rating with cluster : Non Confounding
## 3 ) MLR, green-building rating with size : Non Confounding
## 4 ) MLR, green-building rating with empl_gr : Non Confounding
## 5 ) MLR, green-building rating with leasing_rate : Non Confounding
## 6 ) MLR, green-building rating with stories : Non Confounding
## 7 ) MLR, green-building rating with age : Non Confounding
## 8 ) MLR, green-building rating with renovated : Non Confounding
## 9 ) MLR, green-building rating with class_a : Confounding
## 10 ) MLR, green-building rating with class_b : Non Confounding
## 11 ) MLR, green-building rating with net : Non Confounding
## 12 ) MLR, green-building rating with amenities : Non Confounding
## 13 ) MLR, green-building rating with cd_total_07 : Non Confounding
## 14 ) MLR, green-building rating with hd_total07 : Non Confounding
## 15 ) MLR, green-building rating with total_dd_07 : Non Confounding
## 16 ) MLR, green-building rating with Precipitation : Non Confounding
## 17 ) MLR, green-building rating with Gas_Costs : Non Confounding
## 18 ) MLR, green-building rating with Electricity_Costs : Non Confounding
```

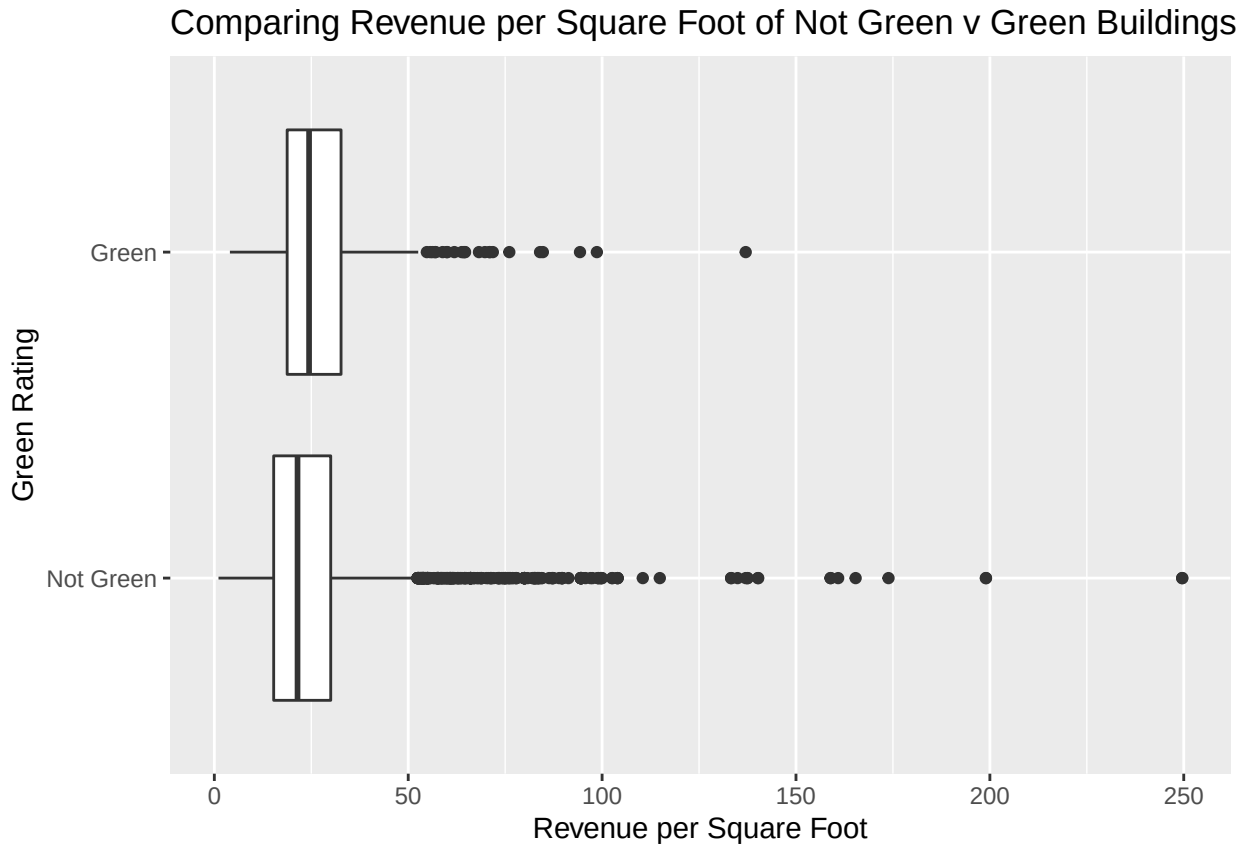
Therefore, It seems that whether a building is rated as a ***Class-A*** listing is actually an underlying confounding variable with the *Rent* and *Green Ratings*

THis is not surprising since a *Class-A* listing is the most desirable property and will be superior to its neighbourhood competition in terms of not only amenities and services but will also be technologically superior and therefore will have lower costs. All these factors will raise the price and also increase the chances that the property qualifies as a *Green* building.

```
##                 Estimate Std. Error   t value
## (Intercept)   -0.7401308  0.1405214 -5.267032
## green_rating1  0.6688835  0.3979776  1.680706
## class_a1       4.0091931  0.2286928 17.530914
```

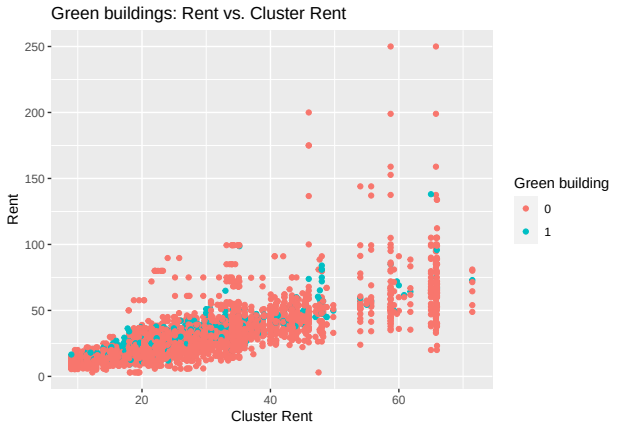
The T-statistic shows that at a 95% confidence, *Green Rating* is in fact not statistically significant in determining the rent!

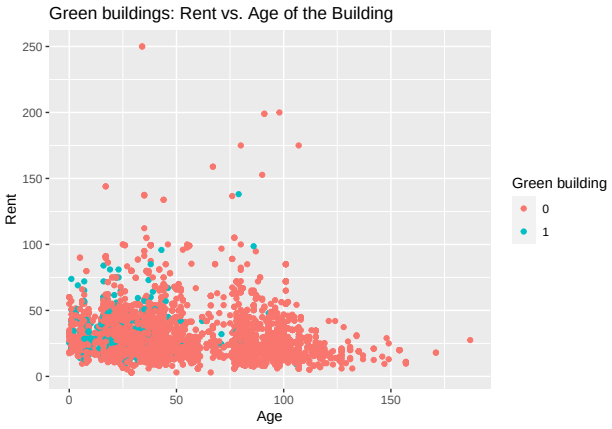
It is the Class-A rating which determines the rent instead!!

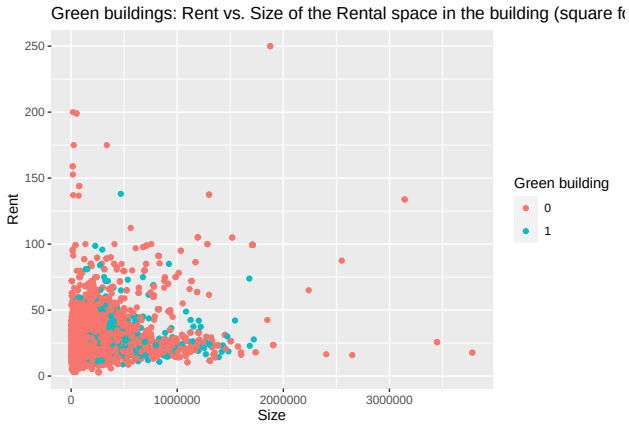


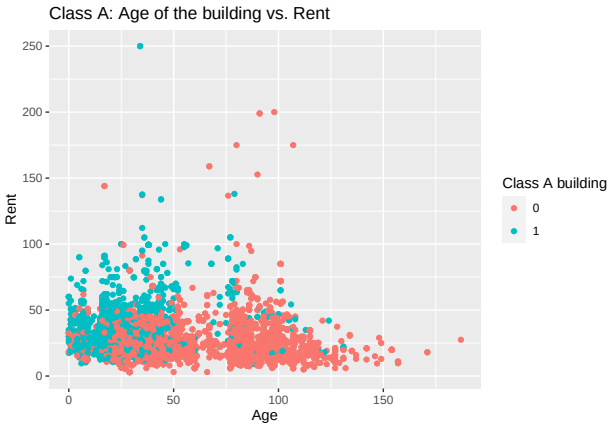
The medians (black lines on the box-plots) and means (the red stars) of green and non-green building revenue per square foot show that green buildings have higher revenue. The non -green buildings have a lower mean than the green buildings

Visualizations



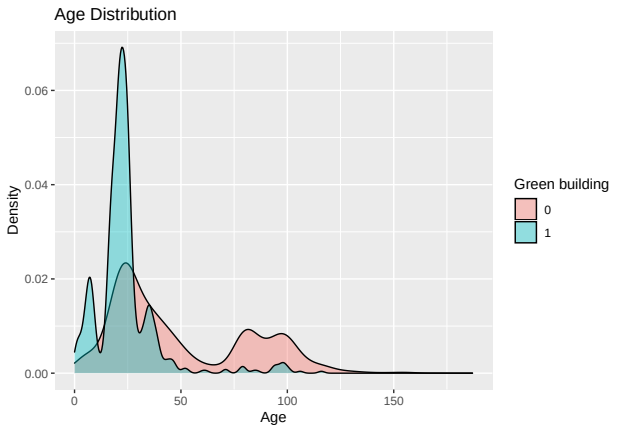


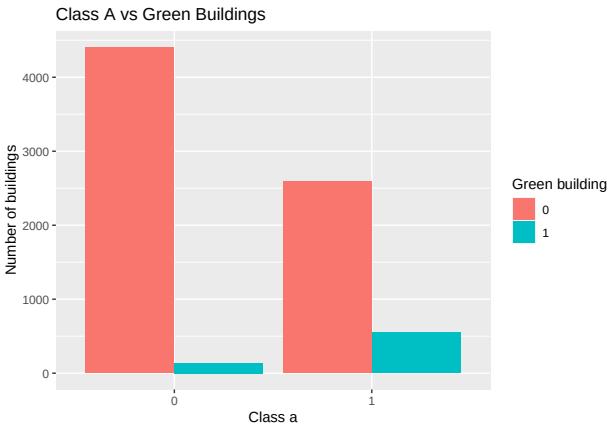




##Inference

- There is a correlation between rent and the cluster rent
- The size of the rental space in the building is also correlated with the Rent
- A Class buildings appear to be younger
- Age does not not seem to have a high correlation with rent
- Class A buildings have higher rent

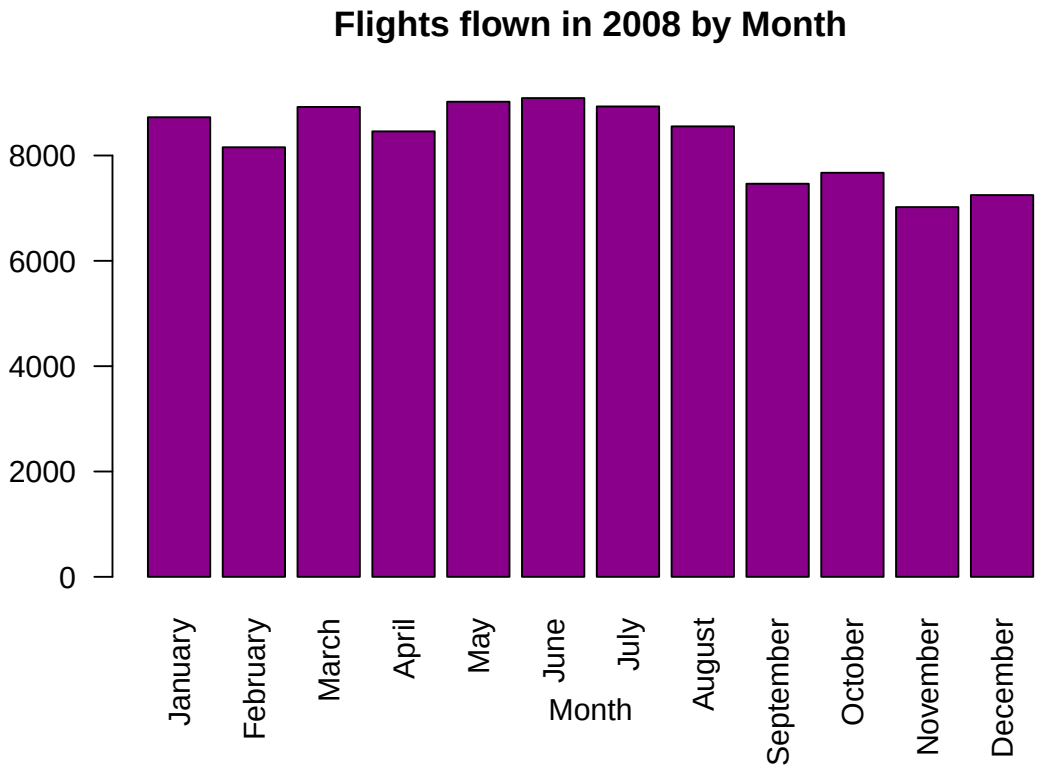




Visual story telling part 2: flights at ABIA

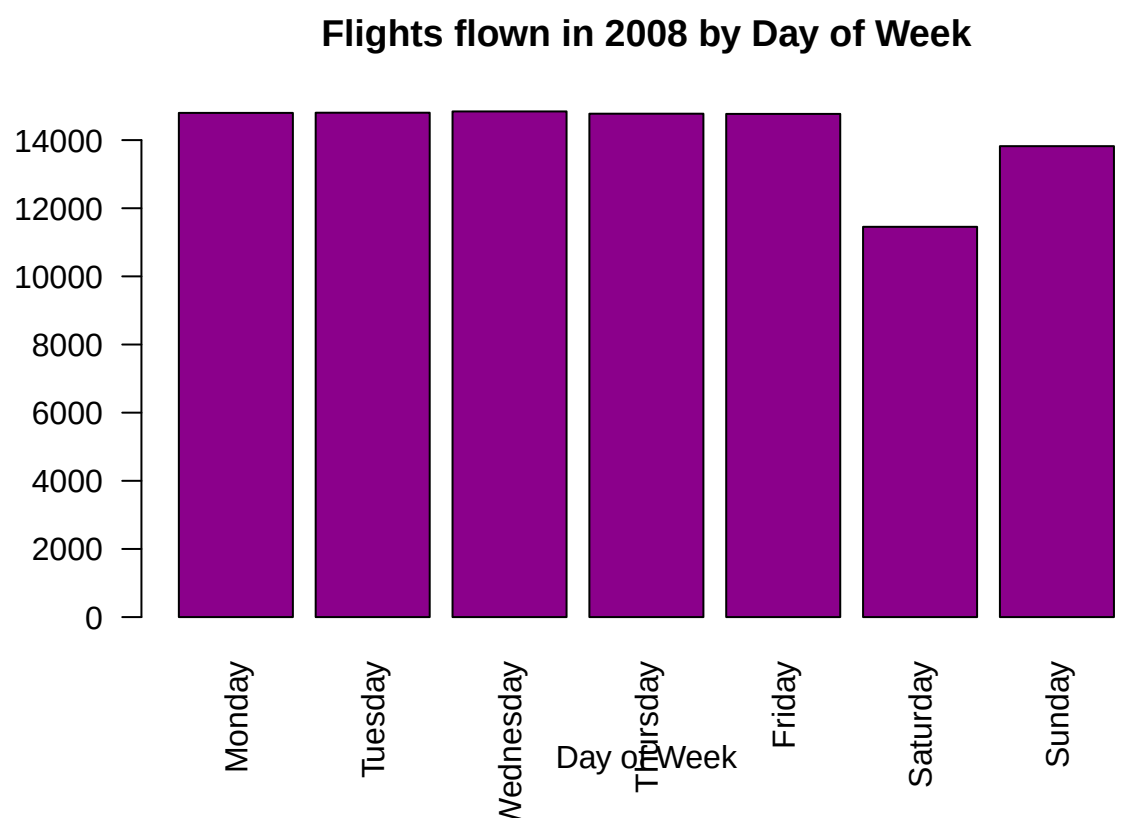
Flights flown by Month

- Number of flights flying during the last 4 months of the year much lesser than other months



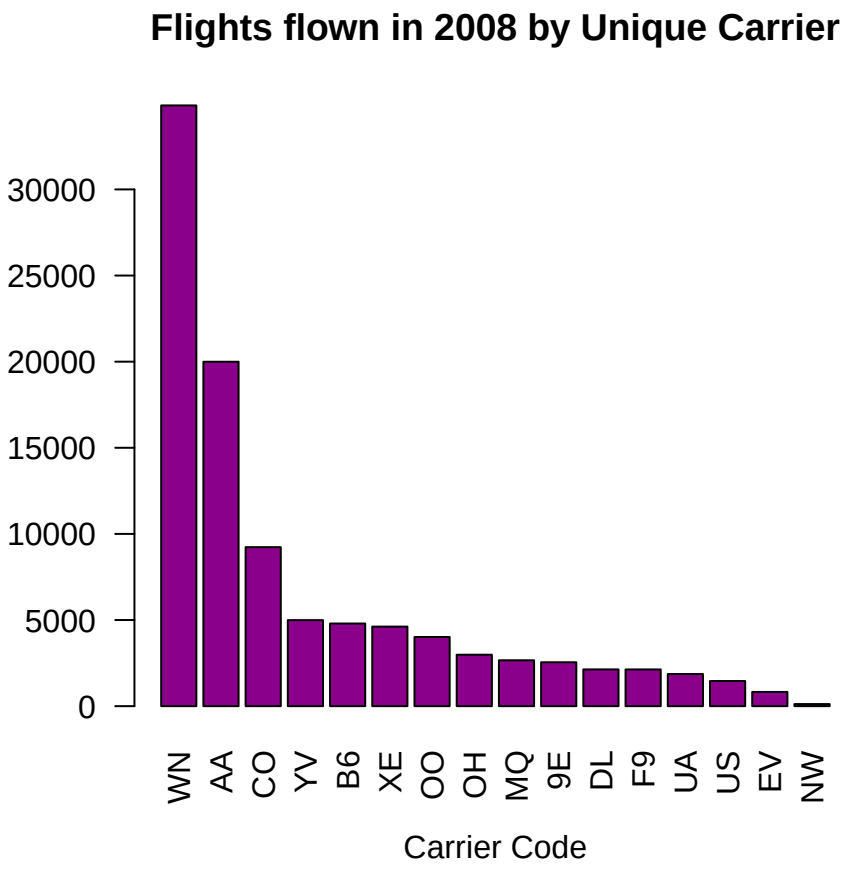
Flights flown by Day of week

- Comparatively lesser flights flown on Saturdays than other days



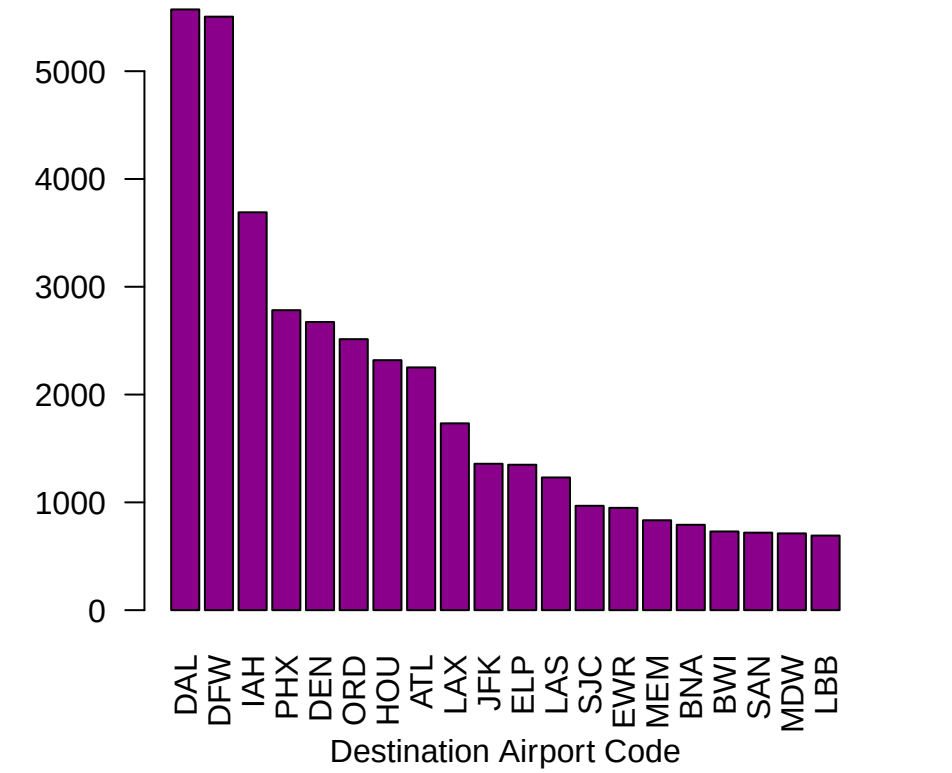
Flights flown by Carrier

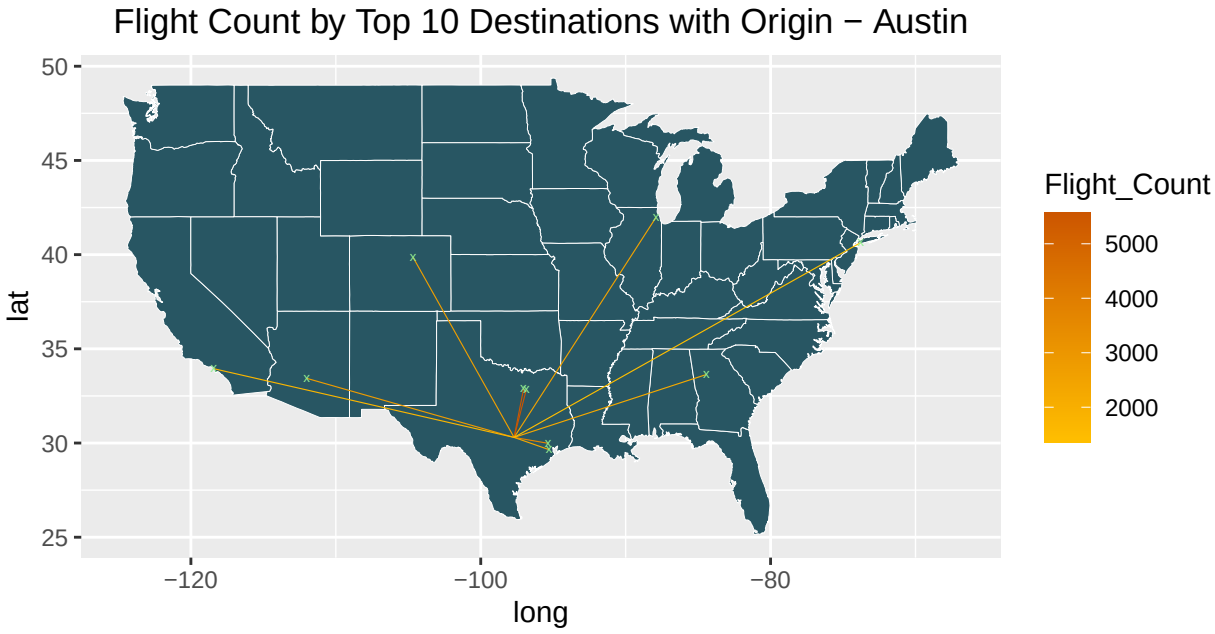
- Southwest Airlines leads in the number of flights flown in 2008, followed by American Airlines



Top 20 destinations in 2008 for flights flown from Austin

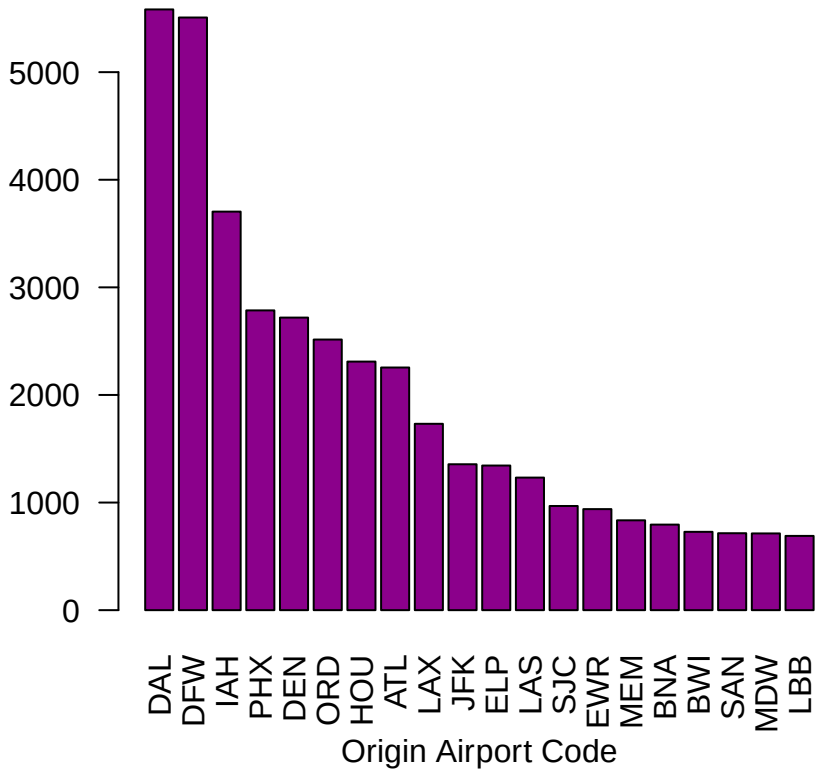
- Top destinations: Dallas, Houston, Phoenix, Denver


Flights flown from Austin in 2008 by Top 20 Destinatio
5000
4000
3000
2000
1000
0
DAL
DFW
IAH
PHX
DEN
ORD
HOU
ATL
LAX
JFK
ELP
LAS
SJC
EWR
MEM
BNA
BWI
SAN
MDW
LBB
Destination Airport Code

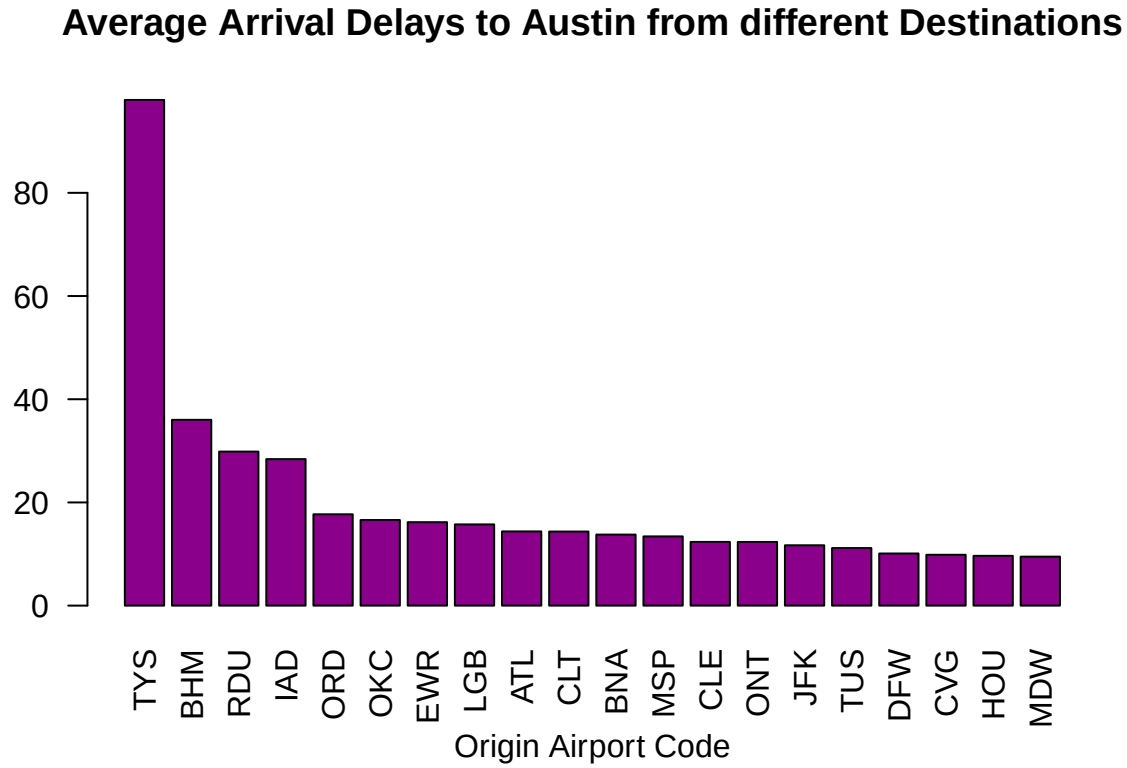


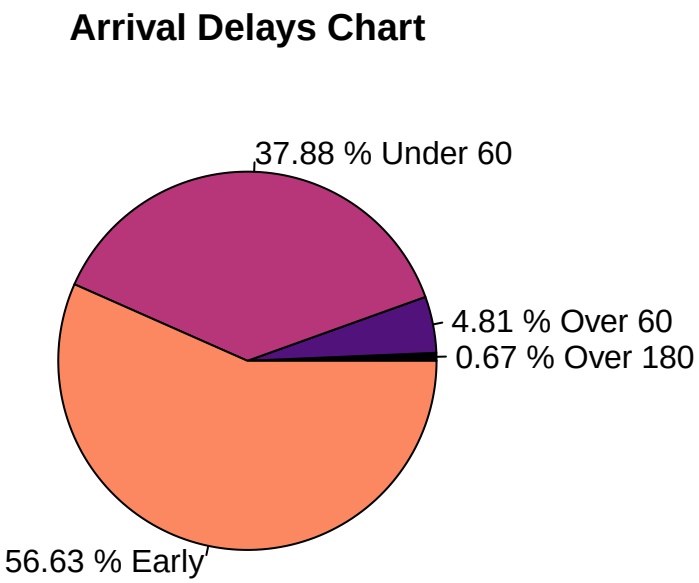
Top 20 Origins in 2008 for flights flown to Austin

- Top Origins: Dallas, Houston, Phoenix, Denver

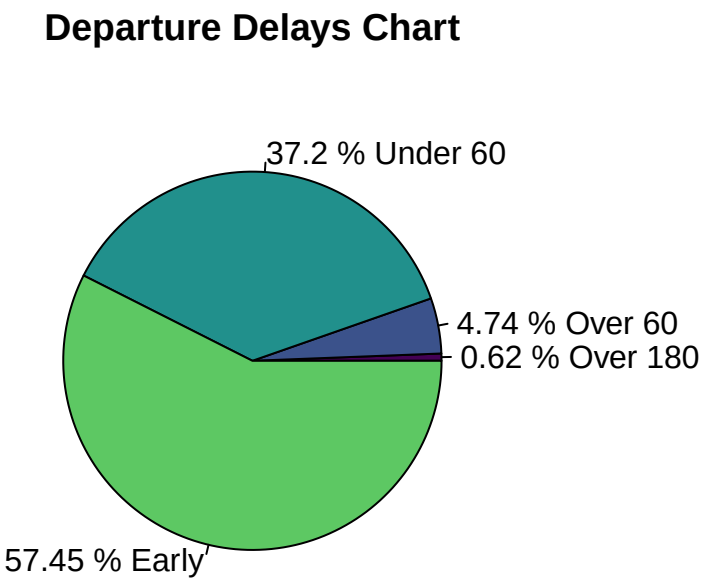

Flights flown to Austin in 2008 by Top 20 Origins
5000
4000
3000
2000
1000
0
DAL
DFW
IAH
PHX
DEN
ORD
HOU
ATL
LAX
JFK
ELP
LAS
SJC
EWR
MEM
BNA
BWI
SAN
MDW
LBB
Origin Airport Code

Plotting average arrival delays from different origins





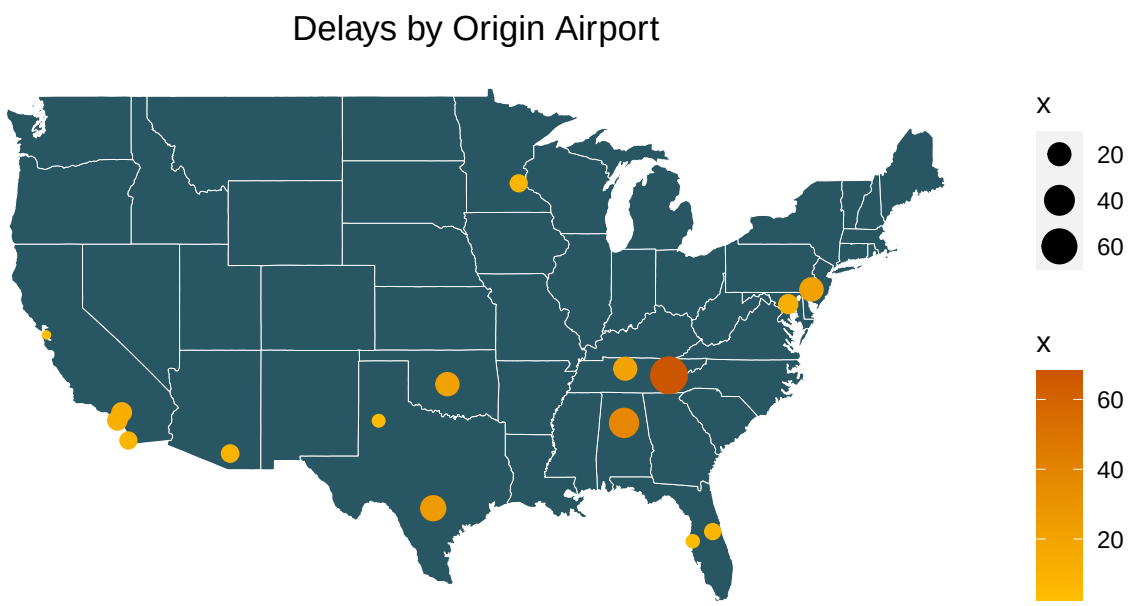
- Flight arrival delay statistics



- Flight departure delay statistics

Flight Delays by Origin Airport

- To Determine which airports have maximum average delays for flights to ABIA



Portfolio modeling

Three ETF portfolios of different asset classes were invested into to test their respective four-trading-week Value at Risk at the 5% level using past five years of data (starting from 2016-01-01). The three classes picked were HY Bonds, Corporate (Investment Grade) Bonds and Government Bonds. $100,000 were allocated for each of the three portfolios containing five ETFs with equal weights (20% in each ETF) which would rebalance daily i.e. each of the ETFs in a portfolio will always comprise 20% of the total wealth at the beginning of each trading day.

High Yield Bonds Portfolio

- For this portfolio, the US traded high yield bond ETFs were selected. This portfolio contains the follow ETFs:
- HYG: Beta = 0.37; Weight = 20%
- JNK: Beta = 0.40; Weight = 20%
- USHY: Beta = 0.40; Weight = 20%
- SRLN: Beta = 0.34; Weight = 20%
- BKLN: Beta = 0.37; Weight = 20%
- Weighted average Beta of the portfolio is 0.376.

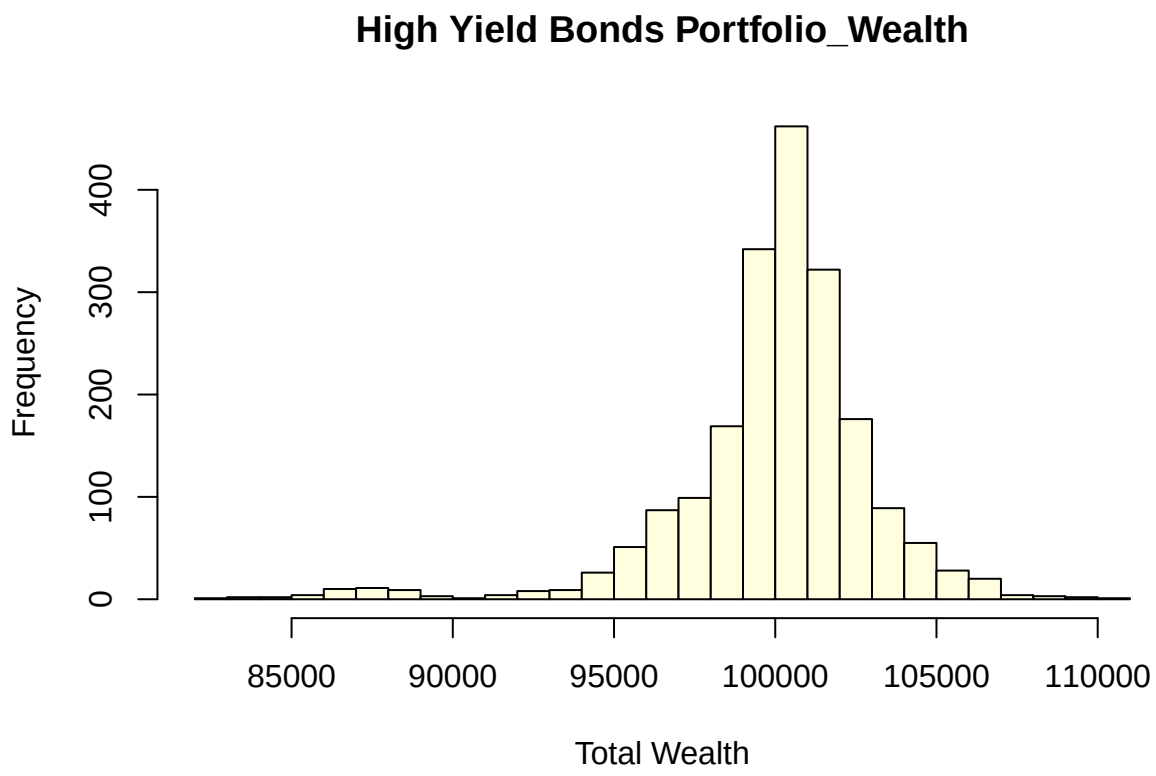
- This portfolio was selected as High Yield Bonds ETFs offer investors exposure to debt issued by below investment grade corporations. These ETFs invest in junk bonds, senior loans, as well as international below investment grade debt.These products have offered numerous ways for investors to take advantage of this space. High yields can be a great addition to a yield-hungry portfolio, as they can offer yields into the double digits for those willing to take on the risks that come along with it. The high returns come from riskier bond choices who have to pay out higher ratios to compensate investors for high risks. This means that the holdings of these ETFs will have higher chances of defaults, and could potentially leave investors out to dry. But for those who have done their homework on the holdings of a particular "junk" bond fund have the ability to generate strong returns from these powerful products.

Investment Grade Bonds Portfolio

- For this portfolio, the US traded investment grade bond ETFs were selected.
- VCIT: Beta = 0.18; Weight = 20%
- LQD: Beta = 0.19; Weight = 20%
- VCSH: Beta = 0.08; Weight = 20%
- IGSB: Beta = 0.08; Weight = 20%
- IGIB: Beta = 0.20; Weight = 20%
- Weighted average Beta of the portfolio is 0.146.
- This portfolio was selected to gain exposure to investment grade corporate bonds, which provides low risk, stable return profile as they are invested in investment-grade credit (highly) rated corporates. However, the returns are greater than the government bonds. Bonds included in these funds can feature varying maturities and are issued by companies from multiple industries.

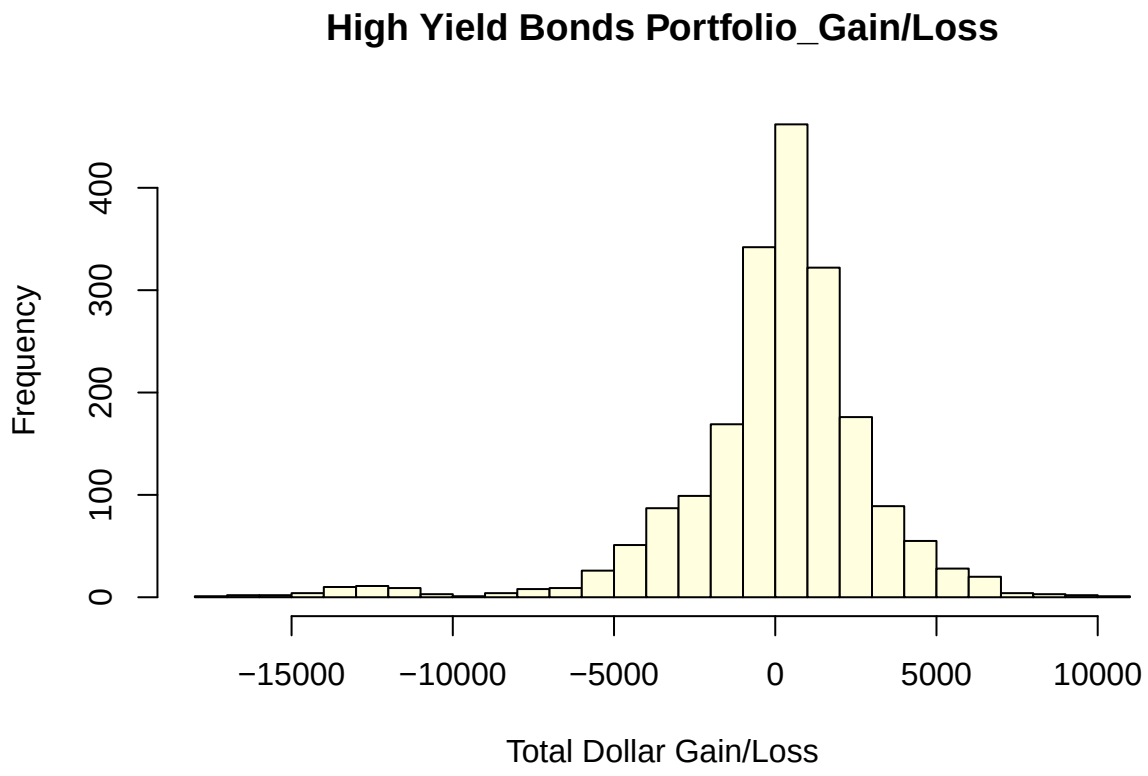
Government Bonds Portfolio

- For this portfolio, the US traded government bond ETFs were selected. This portfolio contains the follow ETFs:
- SHY: Beta = -0.03; Weight = 20%
- TLT: Beta = -0.25; Weight = 20%
- GOVT: Beta = -0.09; Weight = 20%
- SHV: Beta = -0.01; Weight = 20%
- IEF: Beta = -0.13; Weight = 20%
- Weighted average Beta of the portfolio is -0.102
- This portfolio was selected to gain exposure to fixed income securities issued by government agencies. Bonds featured in these ETFs include U.S. Treasuries of varying maturities, floating rate Treasury bonds, and TIPS. The aim for this portfolio is to earn stable return while taking a limited amount of risk. The bonds are mostly either backed by the US Treasury or state governments, the default risk is near zero implying an extremely low risk as indicated by their negative Betas.




```
## [1] 100009.3
```

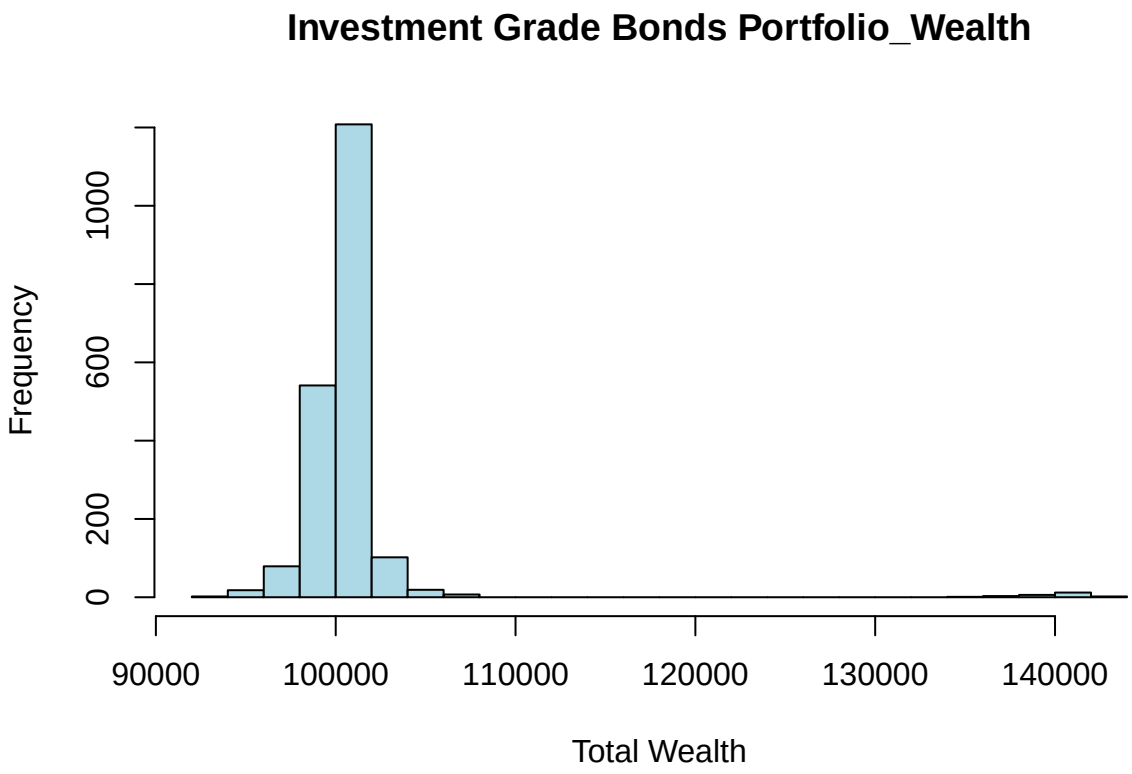
```
## [1] 9.349063
```




```
##        5%
## -4688.555
```

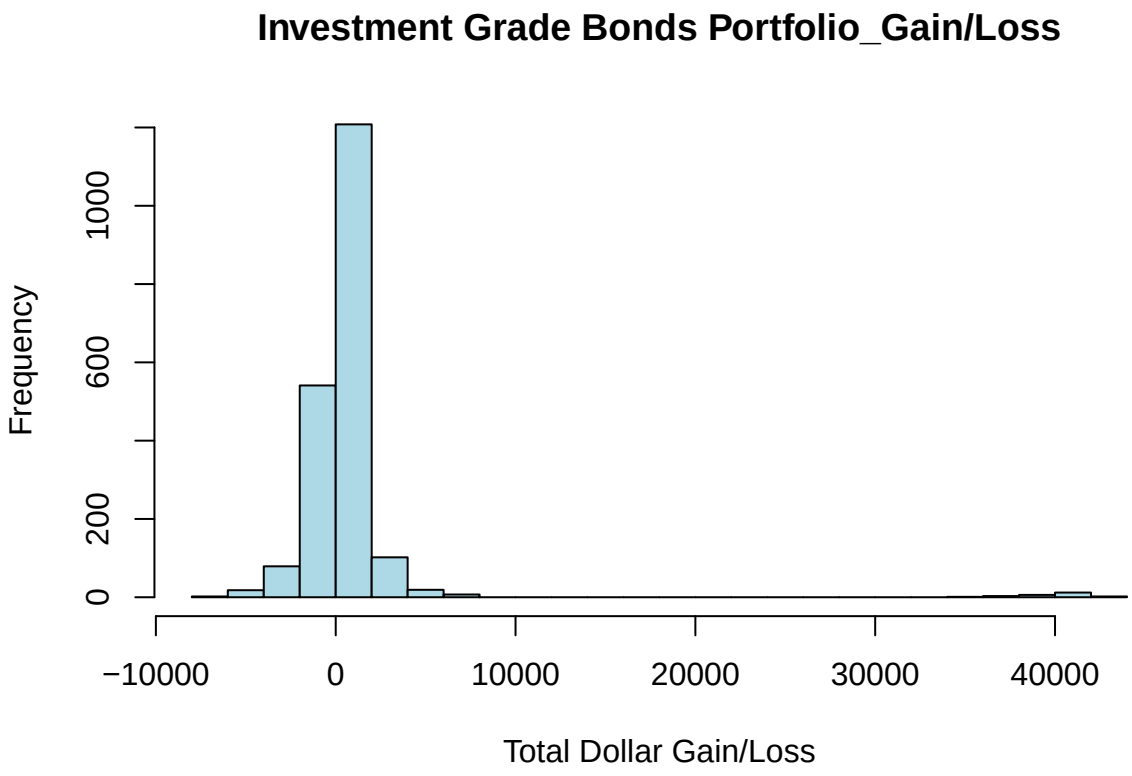
```
##          5%
## -0.04688555
```

VaR=4.688555%




```
## [1] 100836.5
```

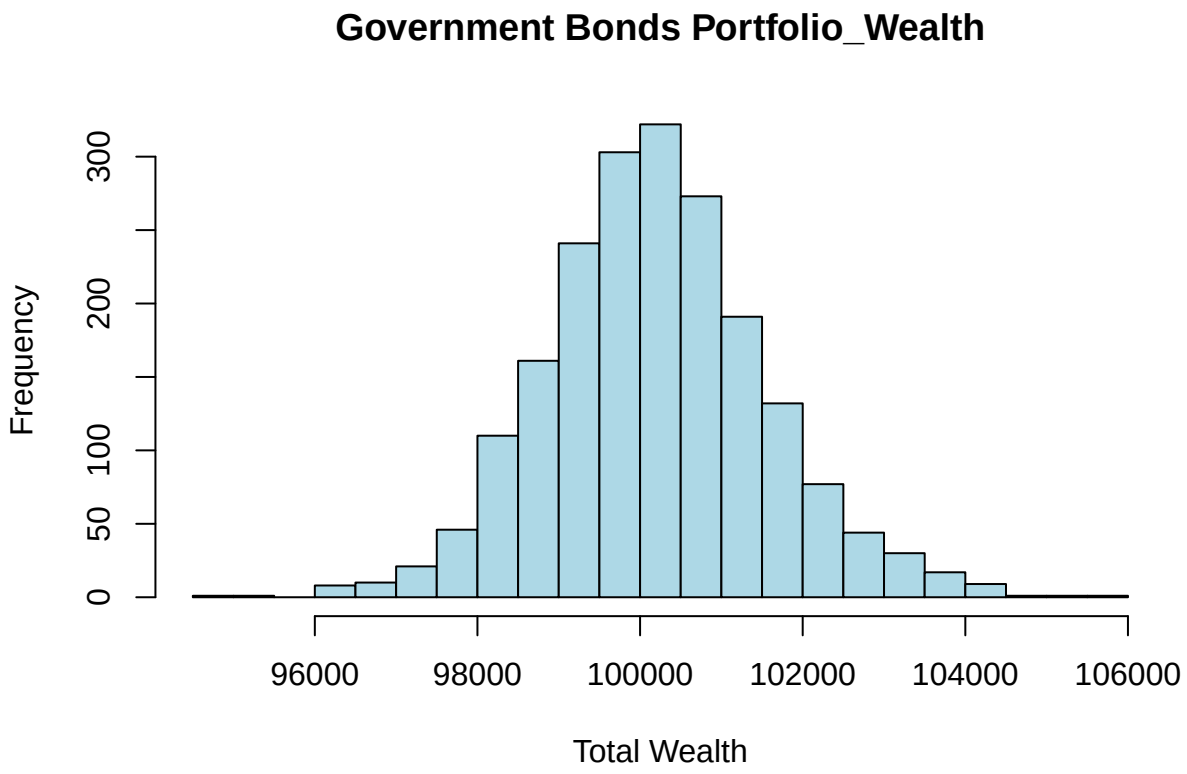
```
## [1] 836.5093
```




```
##        5%
## -1956.064
```

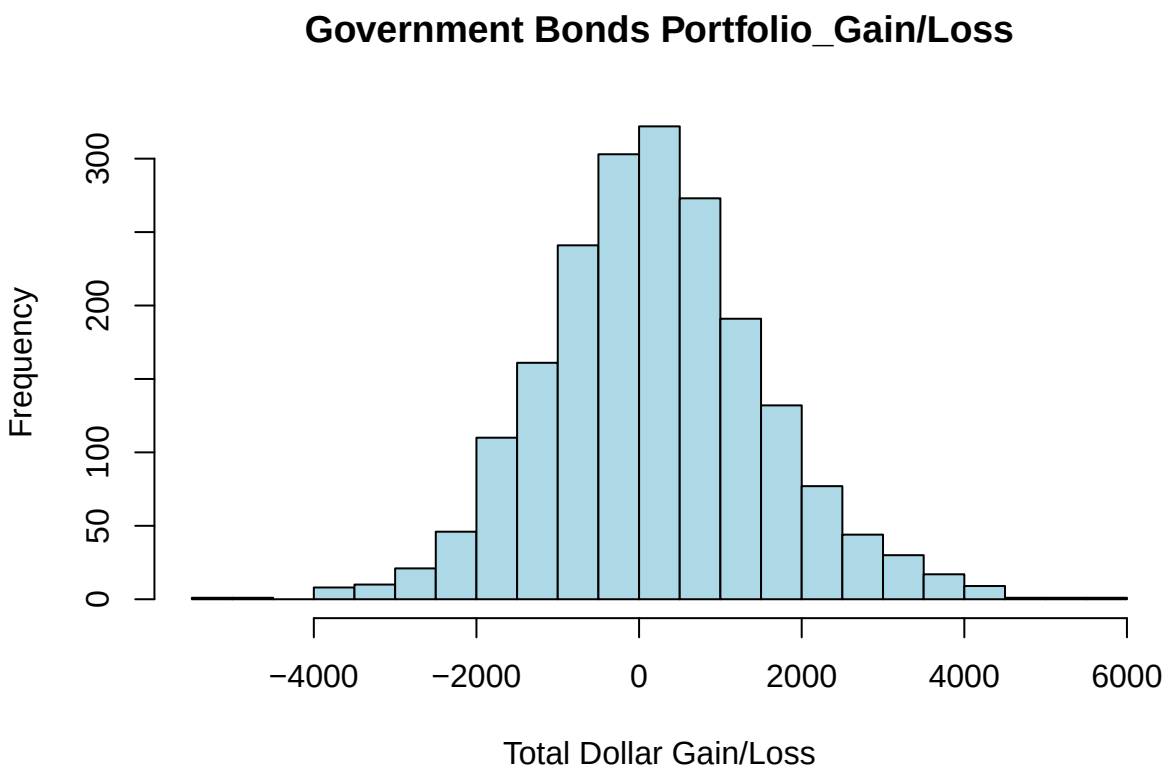
```
##          5%
## -0.01956064
```

VaR=-1.956064%




```
## [1] 100188.4
```

```
## [1] 188.3696
```




```
##        5%
## -1948.256
```

```
##          5%
## -0.01948256
```

-0.01948256 -1.948256%

- 5% Value at Risk for the three portfolios were:
- High Yield Bond Portfolio: -$6,997.02 or -4.69%
- Investment Grade Bond Portfolio: -1.96%
- Government Bond Portfolio: -1.95%
- Investing $100,000 in each of the portfolios for 20 trading days with daily rebalancing, there is a 5% chance for our total gain/loss for (1) the High Yield Bond Portfolio to be less than -4.69%, (2) the Invesment Grade Bond Portfolio to be less than -1.96%, (3) the Government Bond Portfolio to be 1.95%. It is clearly seen that the High Yield portfolio has chances of higher loss due to higher risk involved. The Investment Grade and Government Bond Portfolios appear to have comparatively low and similar loss chances as they are low risk, with the Governemnt one having the least Value at Risk.

Market segmentation

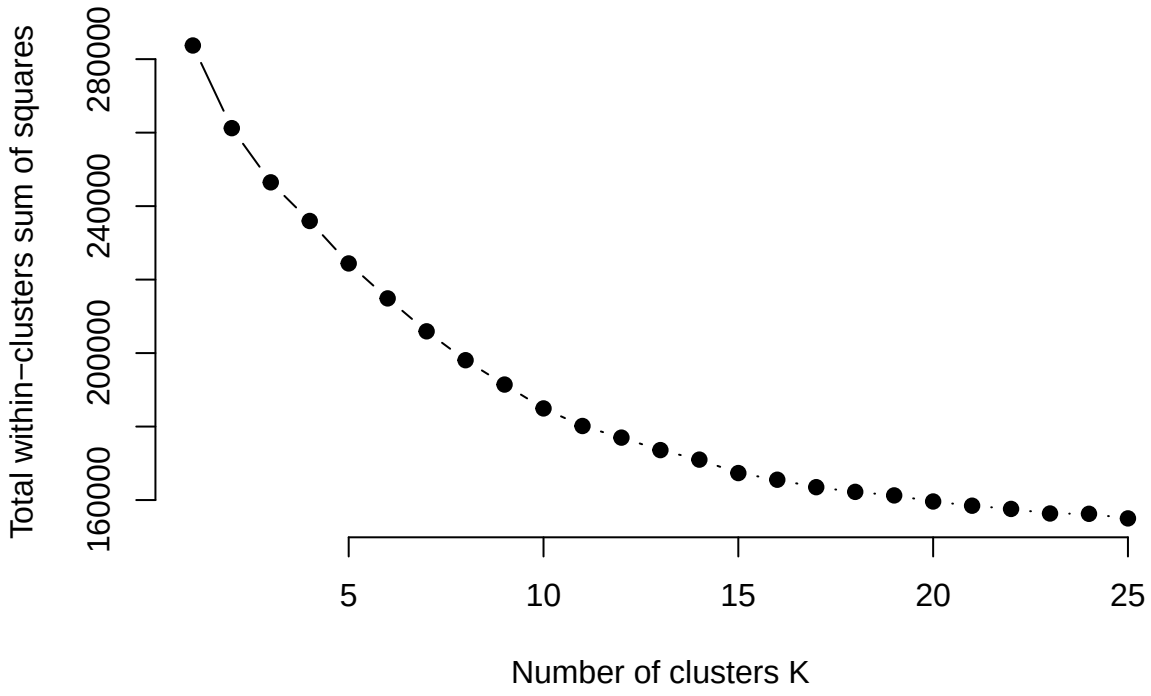
Market Segmentation

Data Pre-Processing

In our Dataset, we observe the following:

- *X* column is essentially the Turk's ID. Thus, this column has been set as the row index and dropped consequently
- Mean-normalised all columns and scaled them to a 0-1 rage *(Performed Z-transformation)* to make the measurements comparable

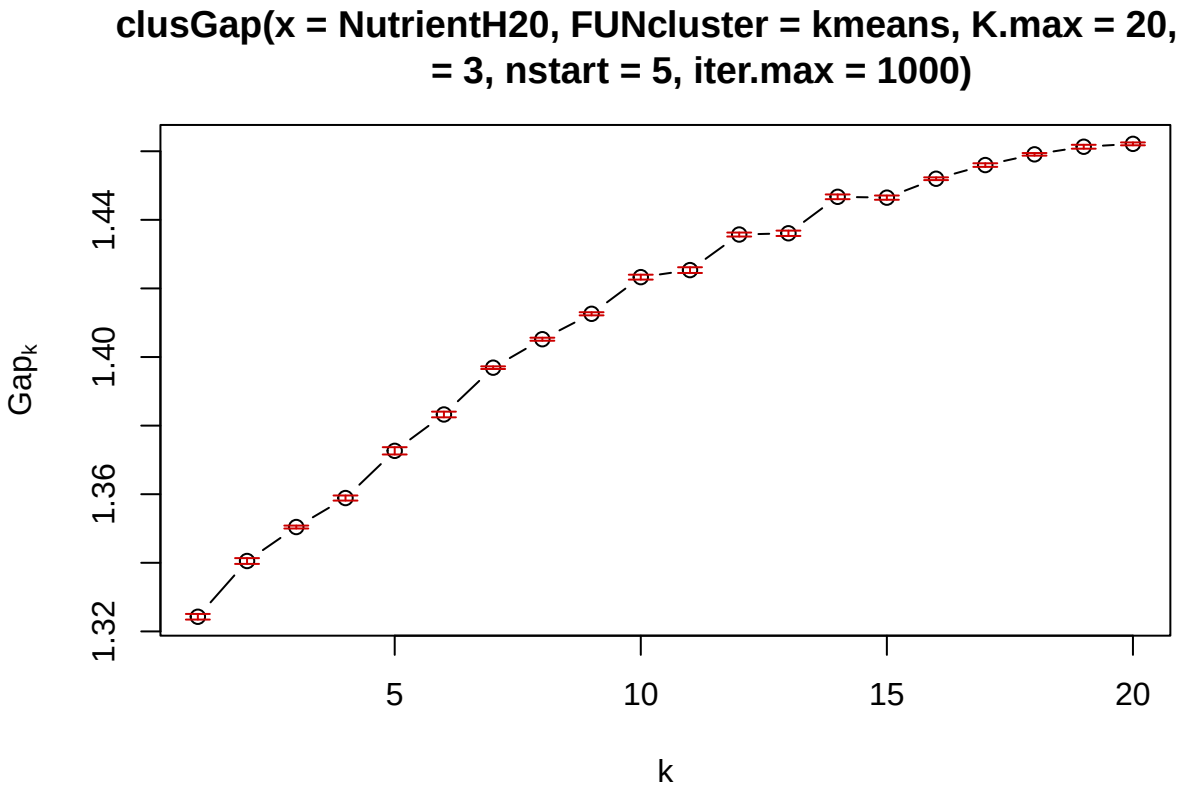
Elbow method



The Elbow Plot above indicates a kink at $\boldsymbol{K = 14}$. This could be a good estimate of the number of clusters we go choose.

However, This method is too *visual* and the differences are not clearly discernible. Thus, let us re-validate our results with the calculation of the **GAP** metric.

Gap Index

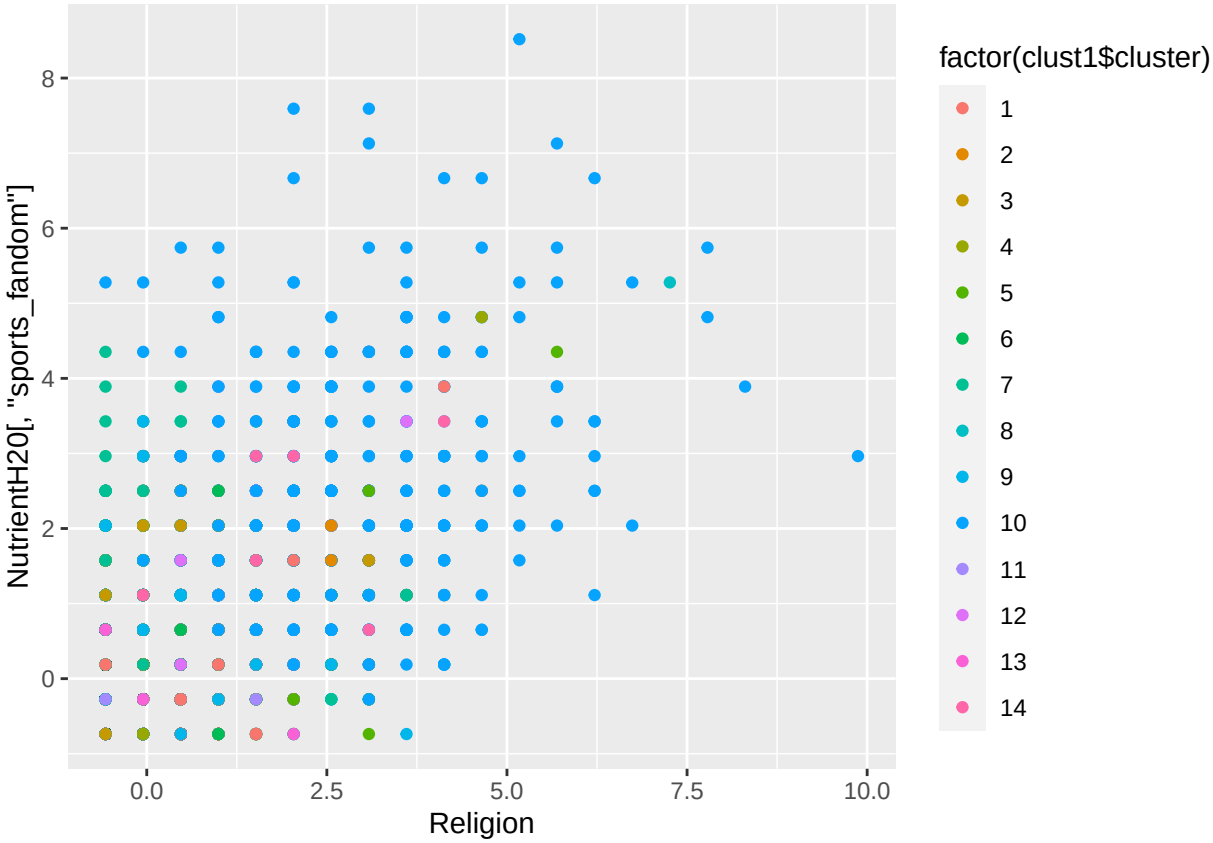


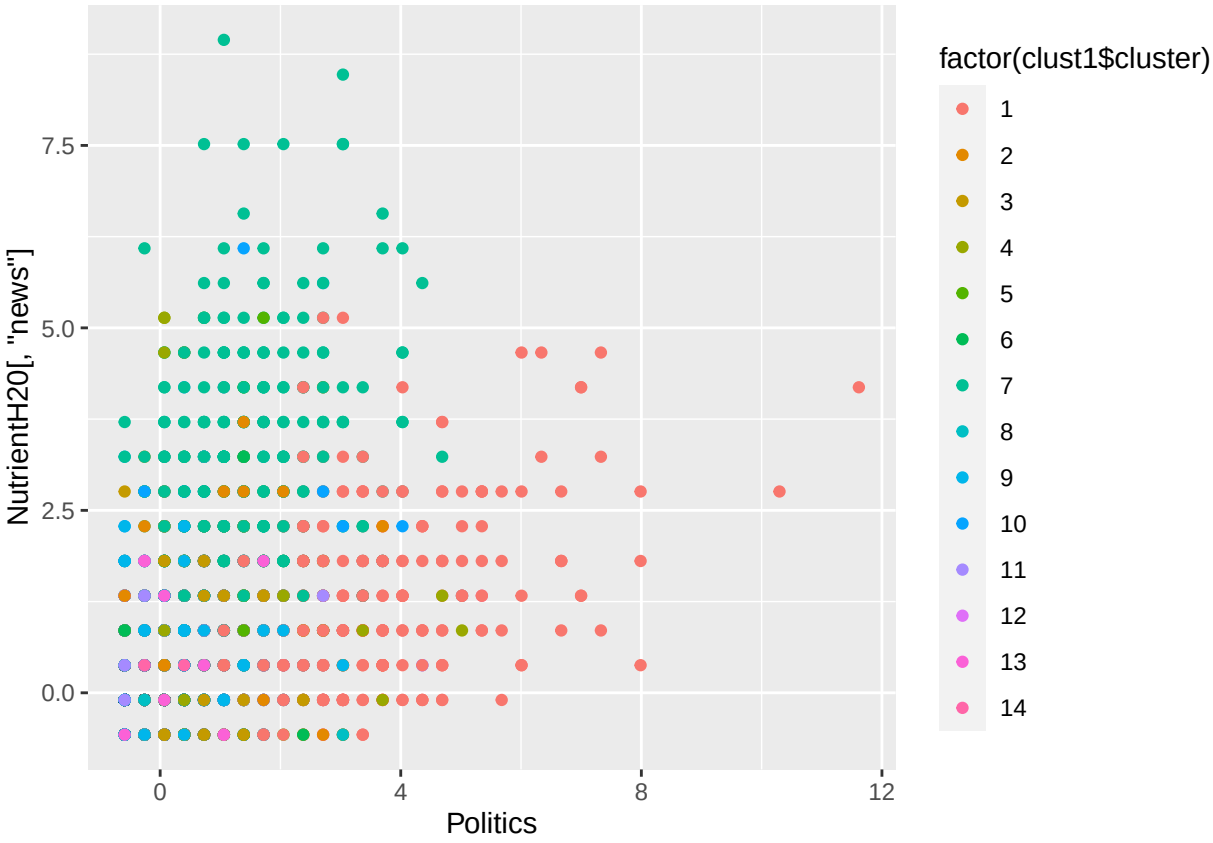

```
##           logW   E.logW      gap       SE.sim
##  [1,] 9.666055 10.99032 1.324268 0.0008224680
##  [2,] 9.611718 10.95224 1.340521 0.0008446234
##  [3,] 9.583689 10.93412 1.350427 0.0004244282
##  [4,] 9.561569 10.92045 1.358884 0.0007422304
##  [5,] 9.538090 10.91074 1.372646 0.0010602093
##  [6,] 9.518623 10.90187 1.383246 0.0008250187
##  [7,] 9.498126 10.89502 1.396896 0.0003843231
##  [8,] 9.482833 10.88803 1.405194 0.0004427242
##  [9,] 9.469945 10.88255 1.412604 0.0004554329
## [10,] 9.454182 10.87747 1.423289 0.0007160178
## [11,] 9.447812 10.87315 1.425342 0.0008341239
## [12,] 9.433375 10.86909 1.435713 0.0005789388
## [13,] 9.429150 10.86523 1.436080 0.0007802581
## [14,] 9.414951 10.86167 1.446714 0.0006919865
## [15,] 9.411458 10.85793 1.446474 0.0006214817
## [16,] 9.402572 10.85456 1.451986 0.0003996070
## [17,] 9.395897 10.85187 1.455969 0.0005255290
## [18,] 9.389830 10.84893 1.459097 0.0003837243
## [19,] 9.384525 10.84584 1.461313 0.0005703963
## [20,] 9.381393 10.84354 1.462150 0.0004201965

##
```

```
##
## Optimum cluster size is: 14
```

From the above plots and output statistic, we see that the local optima of the Cluster Gap plot is achieved at $\boldsymbol{K = 14}$ clusters. Thus, we go ahead with dividing the consumer base in 14 clusters for targeted marketing.

Running K-Means with 14 clusters





```
##
## Cluster Whithinness: 11122.7847934064 10237.0217657893 18304.1104205402 14996.9611695742 6190.420127

##
## Total Whithinness: 170419.12158895

##
## Cluster Betweenness: 113296.87841105
```

Author attribution

- Building a dictionary from training data (half of ReutersC50)

```
## <<DocumentTermMatrix (documents: 2500, terms: 32669)>>
## Non-/sparse entries: 619802/81052698
## Sparsity           : 99%
## Maximal term length: 40
## Weighting          : term frequency (tf)

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

## <<DocumentTermMatrix (documents: 2500, terms: 3393)>>
## Non-/sparse entries: 422971/8059529
## Sparsity           : 95%
## Maximal term length: 40
## Weighting          : term frequency (tf)
```

Pre-processing steps:

- Convert to lowercase
- Numbers removed
- Punctuation removed
- Unnecessary white space trimmed

Data reduced to 2500 documents with 32,669 terms

- Stop words removed

Data reduced to 32,570 terms.

- Sparse terms removed

Data reduced to 3393 terms.

- Word counts changed to TF-IDF weights.
- Process repeated for testing data. 3448 terms in the testing data compared to 3393 terms in the training data. We must consider how to treat these new words.

```
## <<DocumentTermMatrix (documents: 2500, terms: 33472)>>
## Non-/sparse entries: 628611/83051389
## Sparsity           : 99%
## Maximal term length: 45
## Weighting          : term frequency (tf)
```

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

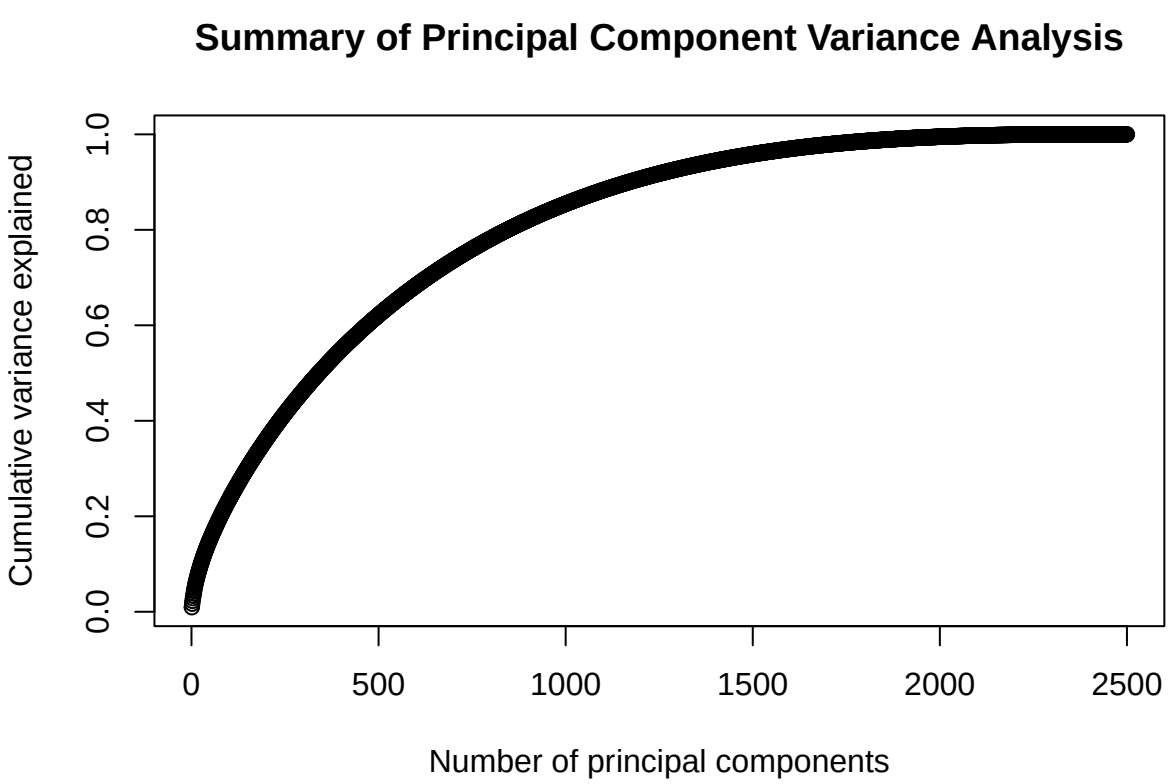
```
## <<DocumentTermMatrix (documents: 2500, terms: 3448)>>
## Non-/sparse entries: 428509/8191491
## Sparsity           : 95%
## Maximal term length: 39
## Weighting          : term frequency (tf)
```

- 55 new terms observed in the testing data
- new terms ignored since they constitute less than 2% of the training data

```
## <<DocumentTermMatrix (documents: 2500, terms: 3393)>>
## Non-/sparse entries: 379314/8103186
## Sparsity           : 96%
## Maximal term length: 40
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

The training set now has 3393 predictors. Principal Component Analysis (PCA) is performed to reduce the number of predictors as follows:

- Eliminate columns that have 0 entries. This eliminates columns where the term is not found in any data in the test or train set. This ensures the term lists are identical and only use columns common between train and test data, leaving 8,317,500 elements in both the train and test matrices.
- Once the data is in the same format, use PCA to choose the number of principal components.



- Choice of number of components is based on desired level of goodness of fit
- At 1000 principal components ~80% of the variance is explained
- Data pre-processing is complete and models can be run

Four models run to predict author attribution:

- KNN
- Random Forest
- Naive Bayes
- Logistic regression

KNN

KNN model run with k = 1, other values of k did not improve the testing or training accuracy.

```
## [1] 0.326
```

- Maximum testing accuracy obtained with KNN : 32.6%

Random Forest

- Random Forest model run with m=31 (square root of 1000, the number of total variables).

```
## [1] 0.784
```

- Maximum testing accuracy obtained with Random Forest : 78.4%
- Large improvement over KNN

Naive Bayes

- Naive Bayes model used to predict the testing data from a training model.

```
## [1] 0.3144
```

- Maximum testing accuracy obtained with Naive Bayes : 31.44%
- Reduction in accuracy from KNN

Multinomial Logistic Regression

- Lastly, multinomial logistic regression performed with only the first 18 Principal Components

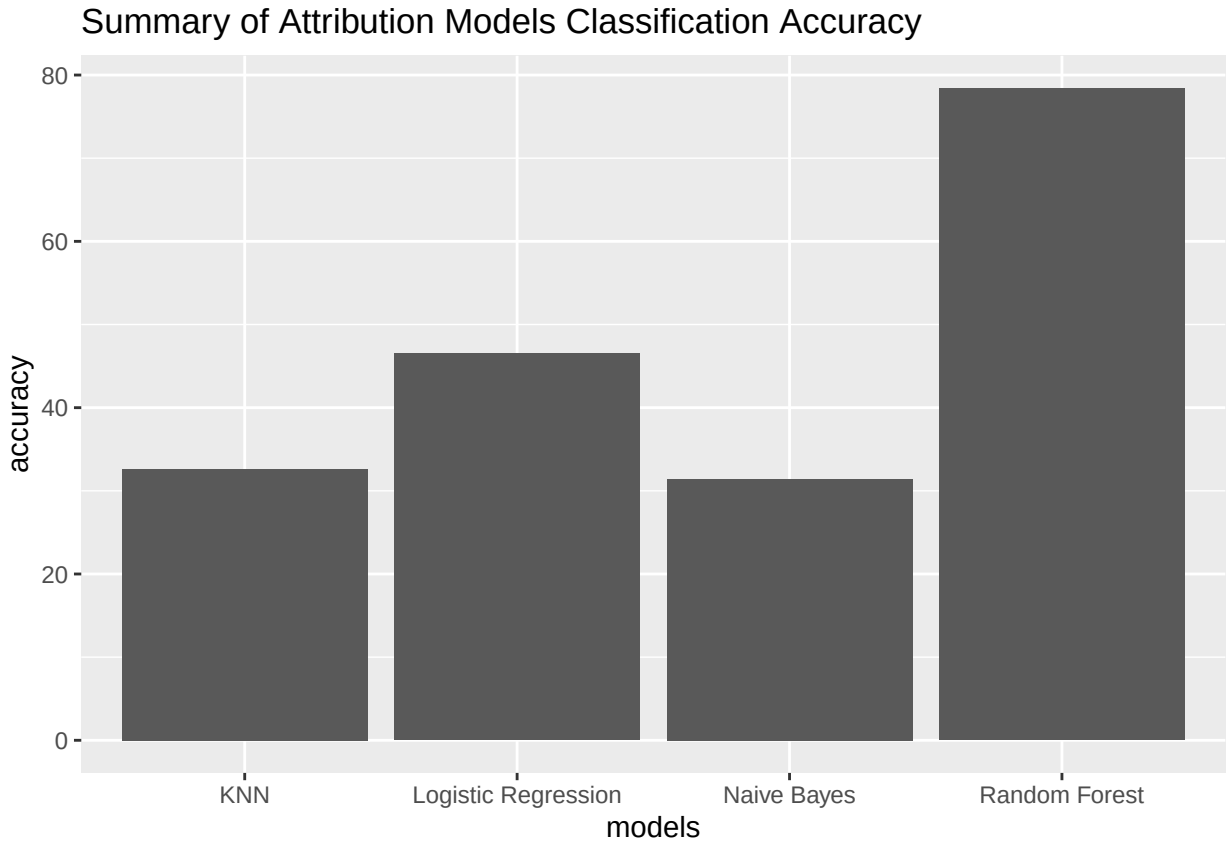
```
## # weights:  1000 (931 variable)
## initial  value 9780.057514
## iter  10 value 3978.610720
## iter  20 value 3027.473914
## iter  30 value 2744.784042
## iter  40 value 2654.854912
## iter  50 value 2552.955018
## iter  60 value 2433.970187
## iter  70 value 2345.527733
## iter  80 value 2305.565551
## iter  90 value 2285.592449
## iter 100 value 2269.532712
## final  value 2269.532712
## stopped after 100 iterations
```

```
## [1] 0.4652
```

- Maximum testing accuracy obtained with Logistic Regression : 46.52%
- Better than KNN or Naive Bayes
- Lower than Random Forest

Comparing the accuracy of the various models:

```
##                models accuracy
## 1       Random Forest    78.40
## 2 Logistic Regression    46.52
## 3                 KNN    32.60
## 4         Naive Bayes    31.40
```



In summary, the random forest model has the highest classification accuracy of ~78.4% on the testing dataset.

Association rule mining

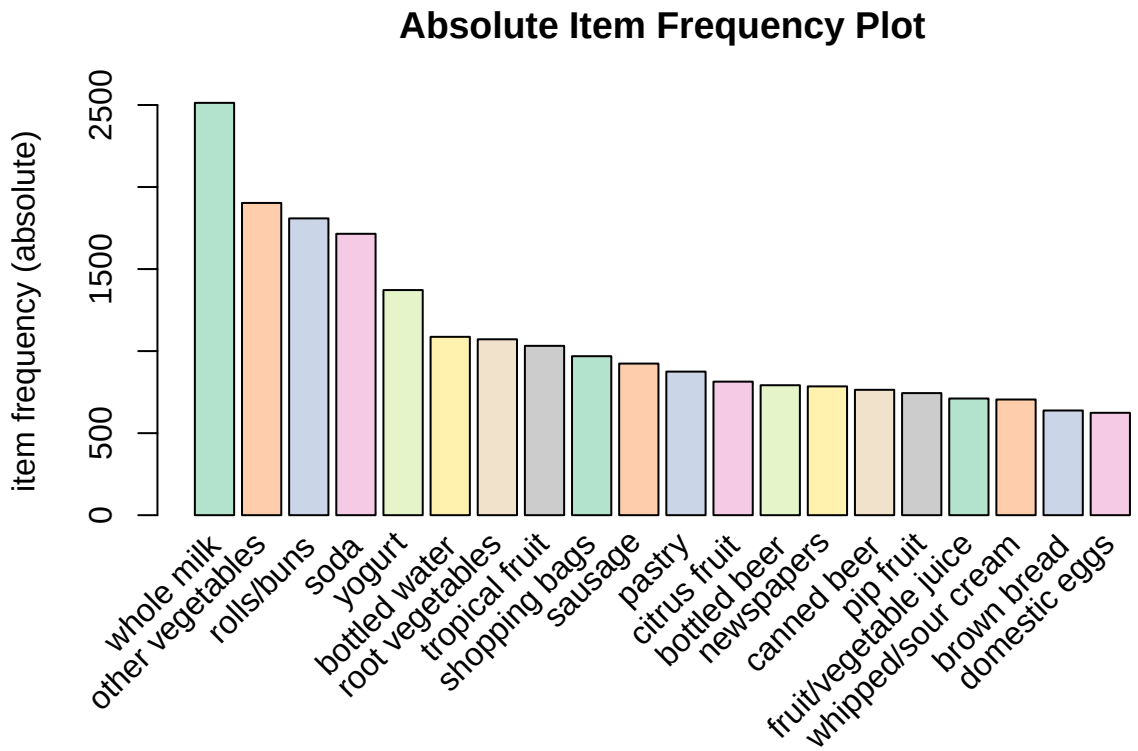
Load the dataset from 'groceries.txt' as transactions of format basket. As shown, there are 2159 transactions with just one item in the basket, 1643 transactions with two items in the basket, and so on.

```
## transactions as itemMatrix in sparse format with
##  9835 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.02609146
##
## 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    5    6    7    8    9   10   11   12   13   14   15   16
## 2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55   46
##   17   18   19   20   21   22   23   24   26   27   28   29   32
##   29   14   14    9   11    4    6    1    1    1    1    3    1
##
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   4.409   6.000  32.000
##
## includes extended item information - examples:
##             labels
## 1 abrasive cleaner
## 2 artif. sweetener
## 3   baby cosmetics
```

Item Frequency Plot

Create the Item frequency plot to indicate the frequencies for different bought items. As shown, whole milk is the most frequently bought grocery item, followed by other vegetables and rolls/buns.



Association mining rules

Creating association mining rules by randomly selecting minimum support as 0.001, confidence as 0.8, and max length=10, results in 410 rules.

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target  ext
```

```
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 9
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [157 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.01s].
## writing ... [410 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
##      lhs                          rhs                    support confidence    coverage      lift cou
## [1]  {liquor,
##       red/blush wine}         => {bottled beer}     0.001931876  0.9047619 0.002135231 11.235269
## [2]  {cereals,
##       curd}                   => {whole milk}       0.001016777  0.9090909 0.001118454  3.557863
## [3]  {cereals,
##       yogurt}                 => {whole milk}       0.001728521  0.8095238 0.002135231  3.168192
## [4]  {butter,
##       jam}                    => {whole milk}       0.001016777  0.8333333 0.001220132  3.261374
## [5]  {bottled beer,
##       soups}                  => {whole milk}       0.001118454  0.9166667 0.001220132  3.587512
## [6]  {house keeping products,
##       napkins}                => {whole milk}       0.001321810  0.8125000 0.001626843  3.179840
## [7]  {house keeping products,
##       whipped/sour cream}     => {whole milk}       0.001220132  0.9230769 0.001321810  3.612599
## [8]  {pastry,
##       sweet spreads}          => {whole milk}       0.001016777  0.9090909 0.001118454  3.557863
## [9]  {curd,
##       turkey}                 => {other vegetables} 0.001220132  0.8000000 0.001525165  4.134524
## [10] {rice,
##       sugar}                  => {whole milk}       0.001220132  1.0000000 0.001220132  3.913649
```

Looking at the above result, 100% of customers who bought {rice,sugar} also bought whole milk. Similarly, 90.48% of customers who bought {liquor, red/blush wine} also bought bottled beer.

Trying more stricter rules with conf=0.9 and shorter rules with maxlen=3, results in only 10 rules as shown below:

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target  ext
##       3  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
```

```
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 9
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [157 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [10 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
##      lhs                          rhs                    support confidence    coverage      lift cou
## [1]  {liquor,
##       red/blush wine}         => {bottled beer}     0.001931876  0.9047619 0.002135231 11.235269    1
## [2]  {cereals,
##       curd}                   => {whole milk}       0.001016777  0.9090909 0.001118454  3.557863    1
## [3]  {bottled beer,
##       soups}                  => {whole milk}       0.001118454  0.9166667 0.001220132  3.587512    1
## [4]  {house keeping products,
##       whipped/sour cream}     => {whole milk}       0.001220132  0.9230769 0.001321810  3.612599    1
## [5]  {pastry,
##       sweet spreads}          => {whole milk}       0.001016777  0.9090909 0.001118454  3.557863    1
## [6]  {rice,
##       sugar}                  => {whole milk}       0.001220132  1.0000000 0.001220132  3.913649    1
## [7]  {bottled water,
##       rice}                   => {whole milk}       0.001220132  0.9230769 0.001321810  3.612599    1
## [8]  {canned fish,
##       hygiene articles}       => {whole milk}       0.001118454  1.0000000 0.001118454  3.913649    1
## [9]  {grapes,
##       onions}                 => {other vegetables} 0.001118454  0.9166667 0.001220132  4.737476    1
## [10] {hard cheese,
##       oil}                    => {other vegetables} 0.001118454  0.9166667 0.001220132  4.737476    1
```

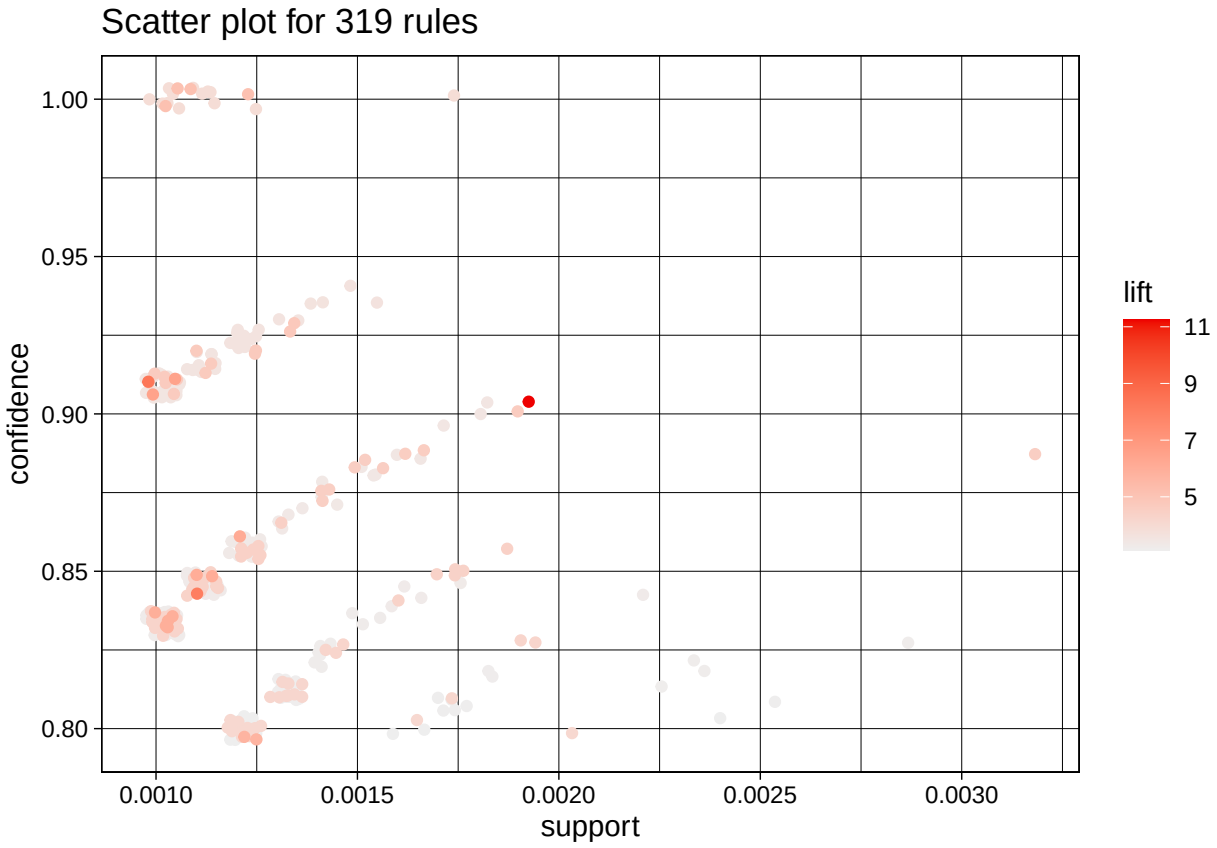
Visualizations

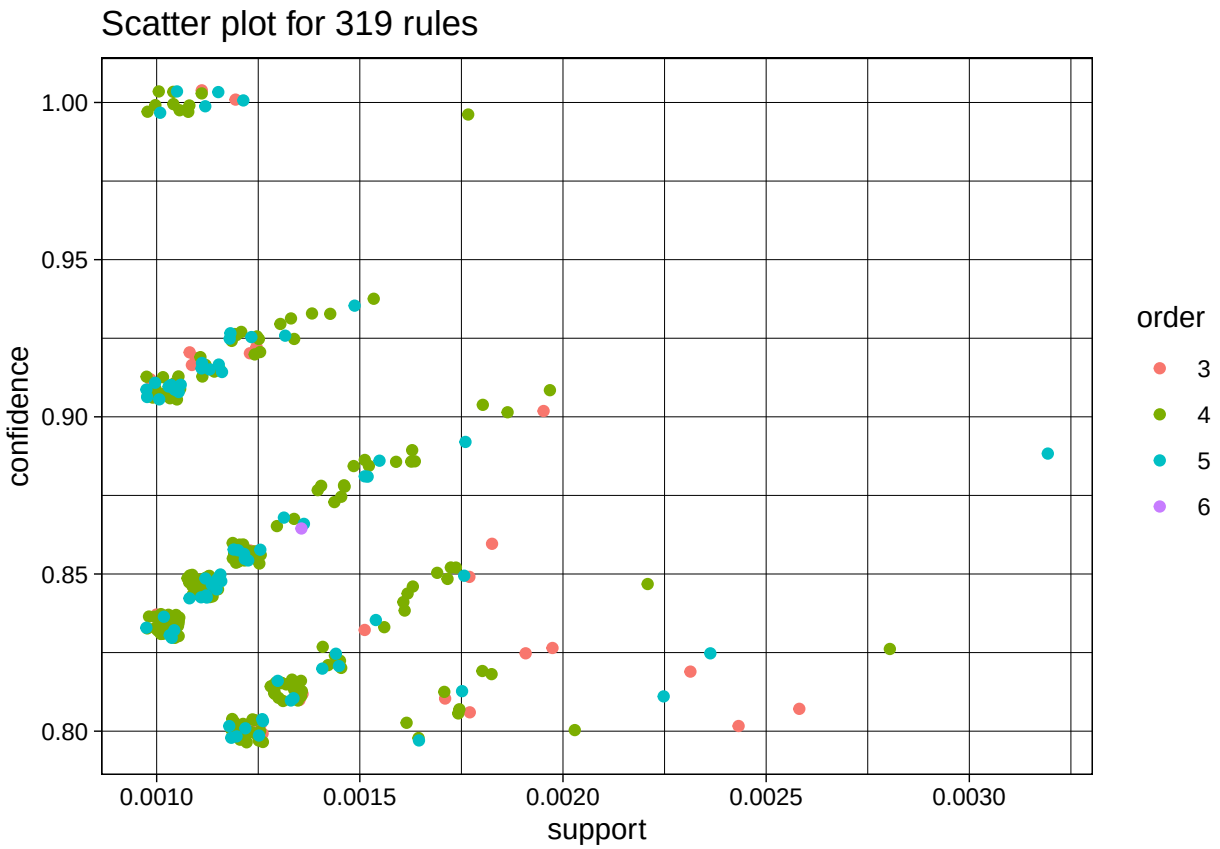
Considering the 410 association rules created by minimum support as 0.001, confidence as 0.8, and max length=10, the next step included removing subsets of larger rules, which resulted in a total of 319 rules.

```
## [1] 91
```

```
##      lhs                          rhs                    support confidence    coverage      lift cou
## [1]  {liquor,
##       red/blush wine}         => {bottled beer}     0.001931876  0.9047619 0.002135231 11.235269    1
## [2]  {cereals,
##       curd}                   => {whole milk}       0.001016777  0.9090909 0.001118454  3.557863    1
## [3]  {cereals,
##       yogurt}                 => {whole milk}       0.001728521  0.8095238 0.002135231  3.168192    1
## [4]  {butter,
##       jam}                    => {whole milk}       0.001016777  0.8333333 0.001220132  3.261374    1
## [5]  {bottled beer,
##       soups}                  => {whole milk}       0.001118454  0.9166667 0.001220132  3.587512    1
```

```
## [6]  {house keeping products,
##       napkins}                => {whole milk}       0.001321810  0.8125000 0.001626843  3.179840
## [7]  {house keeping products,
##       whipped/sour cream}     => {whole milk}       0.001220132  0.9230769 0.001321810  3.612599
## [8]  {pastry,
##       sweet spreads}          => {whole milk}       0.001016777  0.9090909 0.001118454  3.557863
## [9]  {curd,
##       turkey}                 => {other vegetables} 0.001220132  0.8000000 0.001525165  4.134524
## [10] {rice,
##       sugar}                  => {whole milk}       0.001220132  1.0000000 0.001220132  3.913649
```





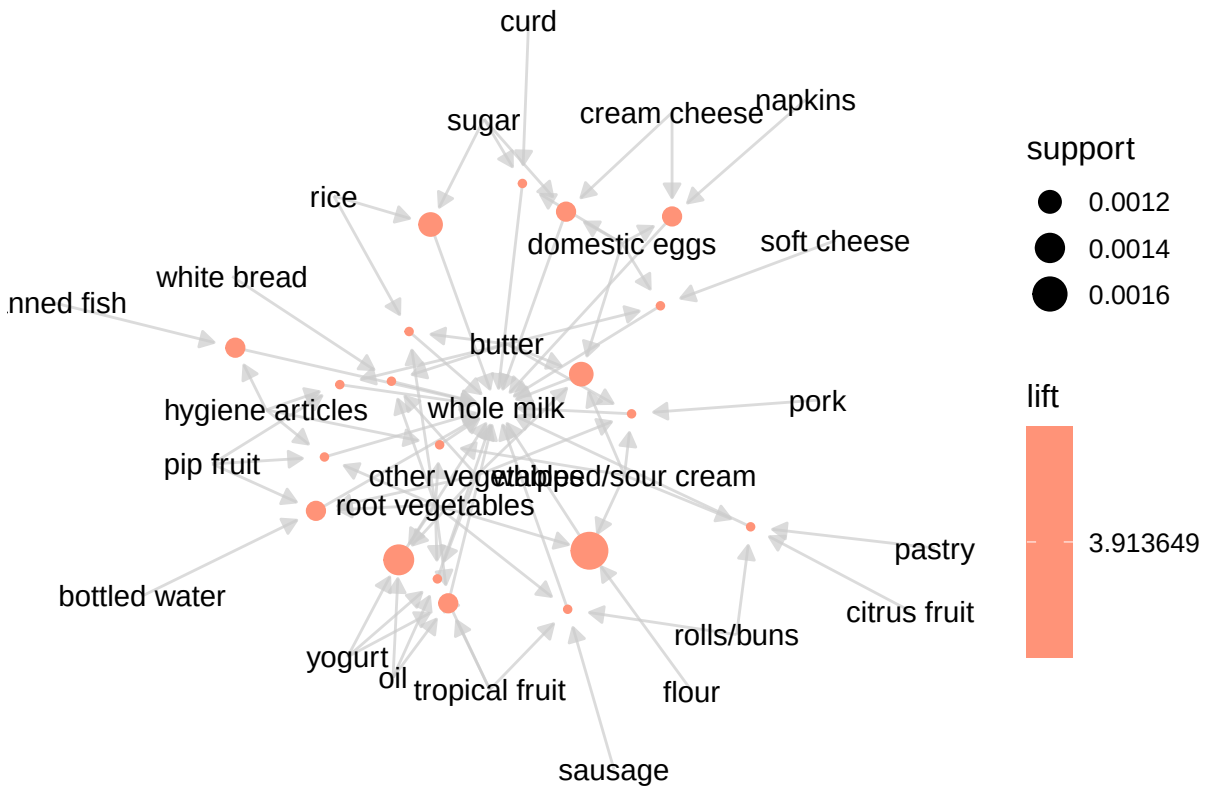
Recommendation rules

Looking at the most bought item in the item frequency list, i.e.m, 'whole milk', it is possible to find the items that are most likely to be bought before buying whole milk by using appearance. Further,using conf=1, will indicate the items where 100% of customers bought whole milk, after buying these items.

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##           1    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 9
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [157 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.01s].
```

```
## writing ... [20 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

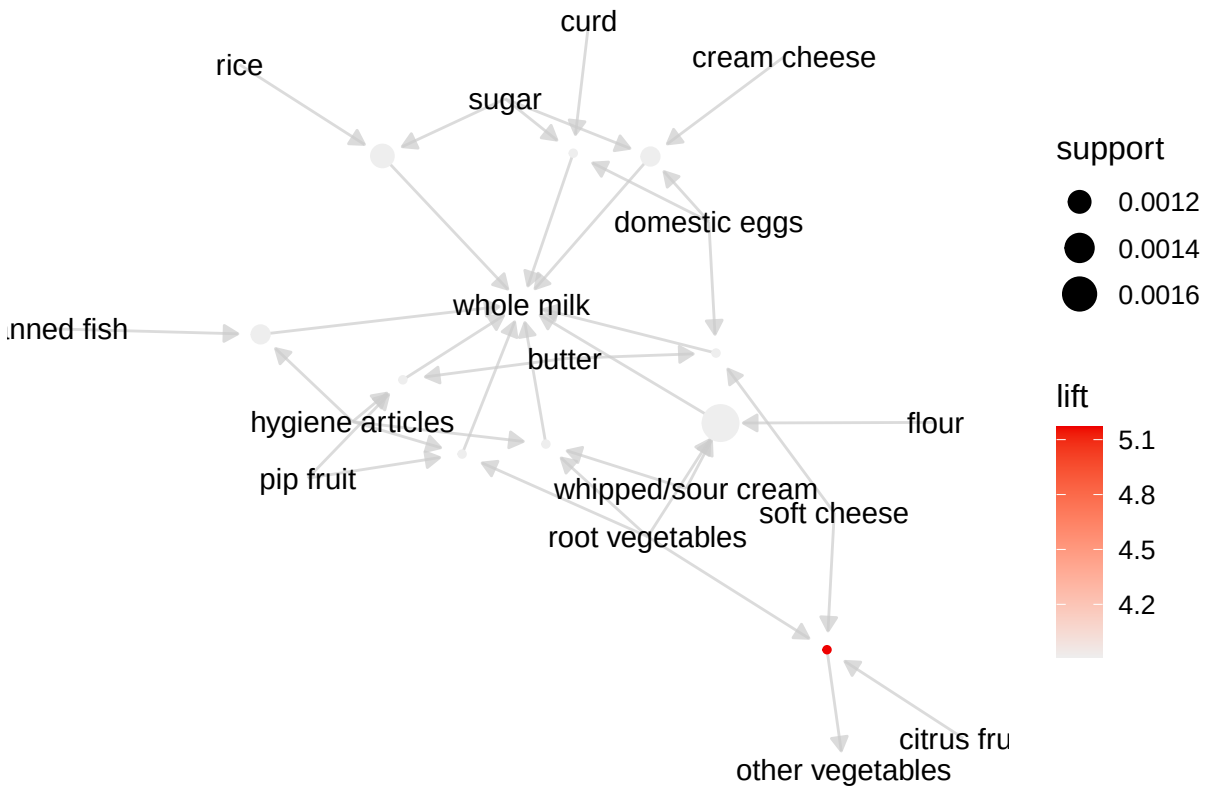
```
##      lhs                     rhs              support confidence    coverage     lift count
## [1]  {rice,
##       sugar}              => {whole milk} 0.001220132          1 0.001220132 3.913649    12
## [2]  {canned fish,
##       hygiene articles}   => {whole milk} 0.001118454          1 0.001118454 3.913649    11
## [3]  {butter,
##       rice,
##       root vegetables}    => {whole milk} 0.001016777          1 0.001016777 3.913649    10
## [4]  {flour,
##       root vegetables,
##       whipped/sour cream} => {whole milk} 0.001728521          1 0.001728521 3.913649    17
## [5]  {butter,
##       domestic eggs,
##       soft cheese}        => {whole milk} 0.001016777          1 0.001016777 3.913649    10
## [6]  {butter,
##       hygiene articles,
##       pip fruit}          => {whole milk} 0.001016777          1 0.001016777 3.913649    10
## [7]  {hygiene articles,
##       root vegetables,
##       whipped/sour cream} => {whole milk} 0.001016777          1 0.001016777 3.913649    10
## [8]  {hygiene articles,
##       pip fruit,
##       root vegetables}    => {whole milk} 0.001016777          1 0.001016777 3.913649    10
## [9]  {cream cheese,
##       domestic eggs,
##       sugar}              => {whole milk} 0.001118454          1 0.001118454 3.913649    11
## [10] {curd,
##       domestic eggs,
##       sugar}              => {whole milk} 0.001016777          1 0.001016777 3.913649    10
```



As shown above, 100% of customers who bought {rice,sugar}, {canned fish,hygiene articles}, {butter,rice,root vegetables}, etc. have bought whole milk, Similarly, we find 20 such antecedents for whole milk.

Sorting rules based on confidence

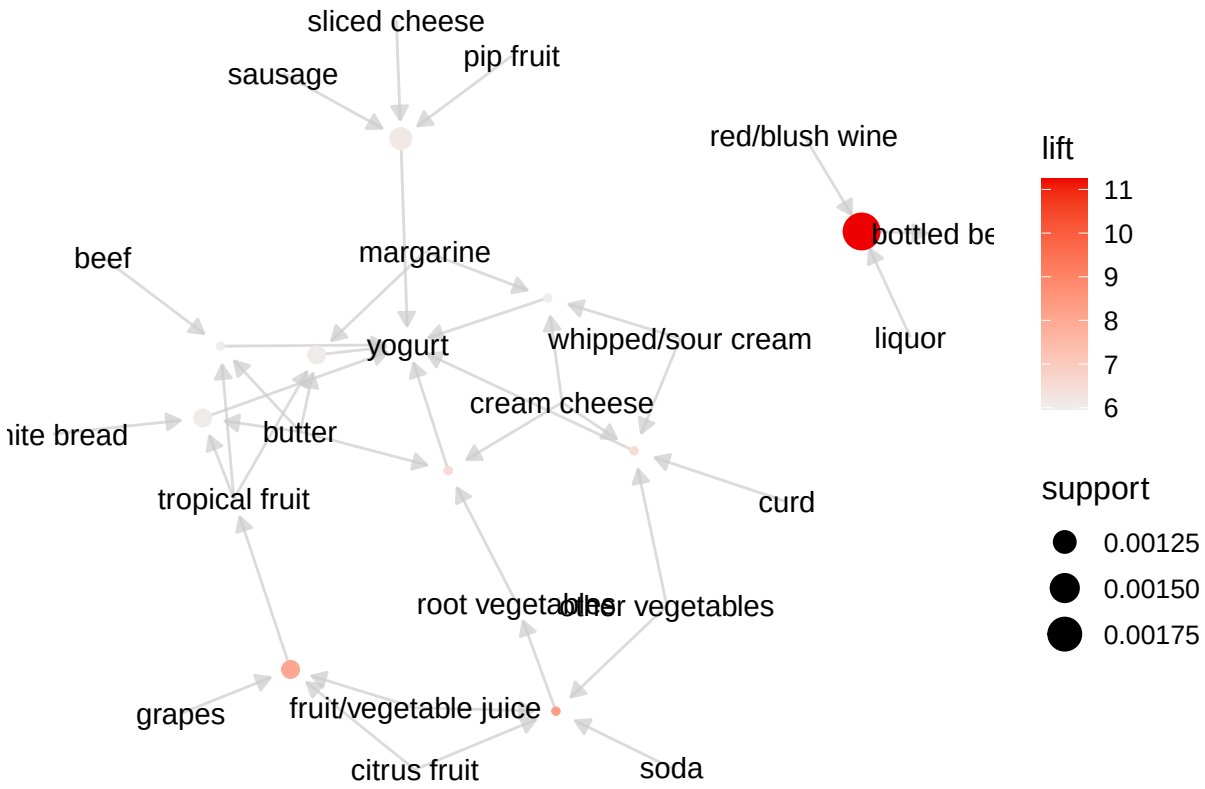
```
##     lhs                     rhs                    support confidence    coverage     lift count
## [1] {rice,
##      sugar}              => {whole milk}       0.001220132          1 0.001220132 3.913649    12
## [2] {canned fish,
##      hygiene articles}   => {whole milk}       0.001118454          1 0.001118454 3.913649    11
## [3] {flour,
##      root vegetables,
##      whipped/sour cream} => {whole milk}       0.001728521          1 0.001728521 3.913649    17
## [4] {butter,
##      domestic eggs,
##      soft cheese}        => {whole milk}       0.001016777          1 0.001016777 3.913649    10
## [5] {citrus fruit,
##      root vegetables,
##      soft cheese}        => {other vegetables} 0.001016777          1 0.001016777 5.168156    10
## [6] {butter,
##      hygiene articles,
##      pip fruit}          => {whole milk}       0.001016777          1 0.001016777 3.913649    10
```



A confidence of 1 indicates that whenever the items on the antecedent are bought, 100% of customers bought the item(s) on the consequent.

Sorting rules based on lift

```
##     lhs                          rhs                   support confidence    coverage      lift count
## [1] {liquor,
##      red/blush wine}        => {bottled beer}    0.001931876  0.9047619 0.002135231 11.235269    19
## [2] {citrus fruit,
##      fruit/vegetable juice,
##      other vegetables,
##      soda}                  => {root vegetables} 0.001016777  0.9090909 0.001118454  8.340400    10
## [3] {citrus fruit,
##      fruit/vegetable juice,
##      grapes}                => {tropical fruit}  0.001118454  0.8461538 0.001321810  8.063879    11
## [4] {butter,
##      cream cheese,
##      root vegetables}       => {yogurt}          0.001016777  0.9090909 0.001118454  6.516698    10
## [5] {cream cheese,
##      curd,
##      other vegetables,
##      whipped/sour cream}    => {yogurt}          0.001016777  0.9090909 0.001118454  6.516698    10
## [6] {pip fruit,
##      sausage,
##      sliced cheese}         => {yogurt}          0.001220132  0.8571429 0.001423488  6.144315    12
```



A rule with lift of 11.23 for {liquore,red/blush wine}->{bottled beer} indicates that the items in the antecedent and consequent are ~11 times more likely to be bought together than bought individually.