STA 380, Part 2: Exercises

Amber Camilleri

August 19, 2019

Contents

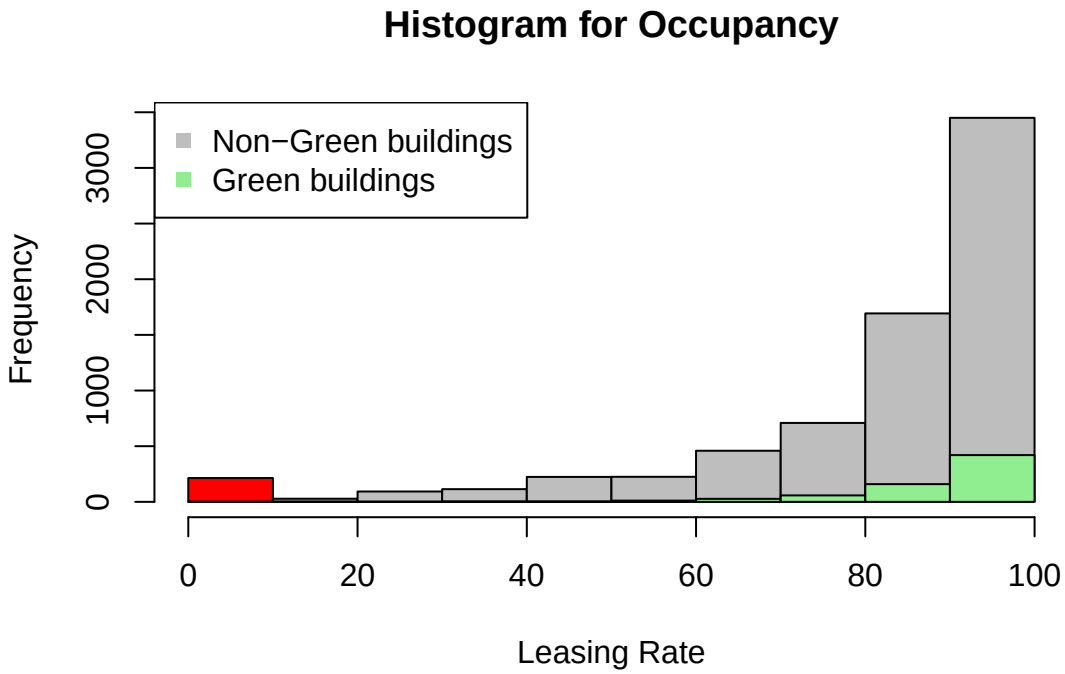
1. Visual Story Telling Part 1: Green Buildings

The following analysis will outline the evidence in support of/ opposition to the conclusions of the on-staff stats guru.

(a) Is it reasonable to remove low occupied outliers?

> *"I noticed that a handful of the buildings in the data set had very low occupancy rates (less than 10% of available space occupied). I decided to remove these buildings from consideration, on the theory that these buildings might have something weird going on with them, and could potentially distort the analysis."*

I looked into the distribution of leasing rate of green buildings and non-green buildings. Interestingly, the distribution of non-green buildings' leasing rate has a shoot up in the range below 10%. Therefore, I hold the same belief that these buildings are "weird" and should be removed from our analysis.




```
## [1] "The number of outliers is 112"
```

```
## [1] "The mean rent of outliers is 103.209285714286 , while the mean rent of all
buildings is 28.5858458132569"
```

```
## [1] "The mean # of stories of outliers is 29.2946428571429 , while the mean # of
stories of all buildings is 13.8299257715848"
```

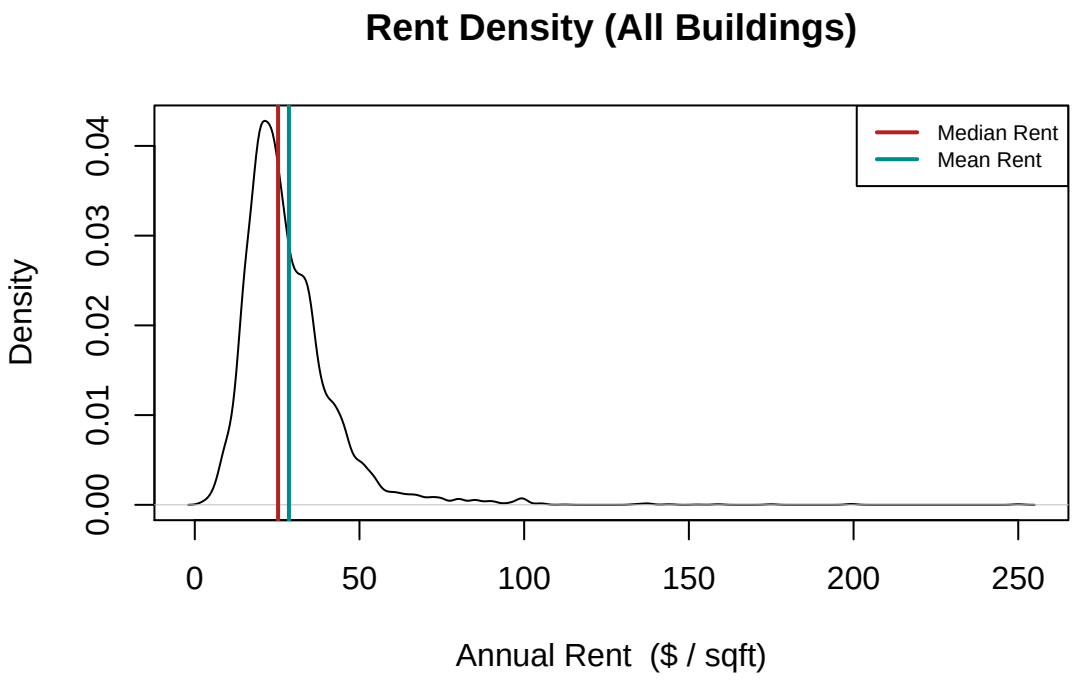
```
## [1] "The number of green buildings among outliers is 7 , while the number of outliers
is 112 . The fraction of outliers that are green is 0.0625"
```

Most of these outliers have significantly higher rent than the average level. And then most of them have specifically high stories level which might be the reason of extremely high rent. Considering I am estimating for a 15-story building. These outliers might be less valuable for analysis. In addition, among these outliers, there exist few green buildings, which means I am not able to compare green and non-green building among these outleirs. In conclusion, I agree with the guru's decision to remove these outliers.

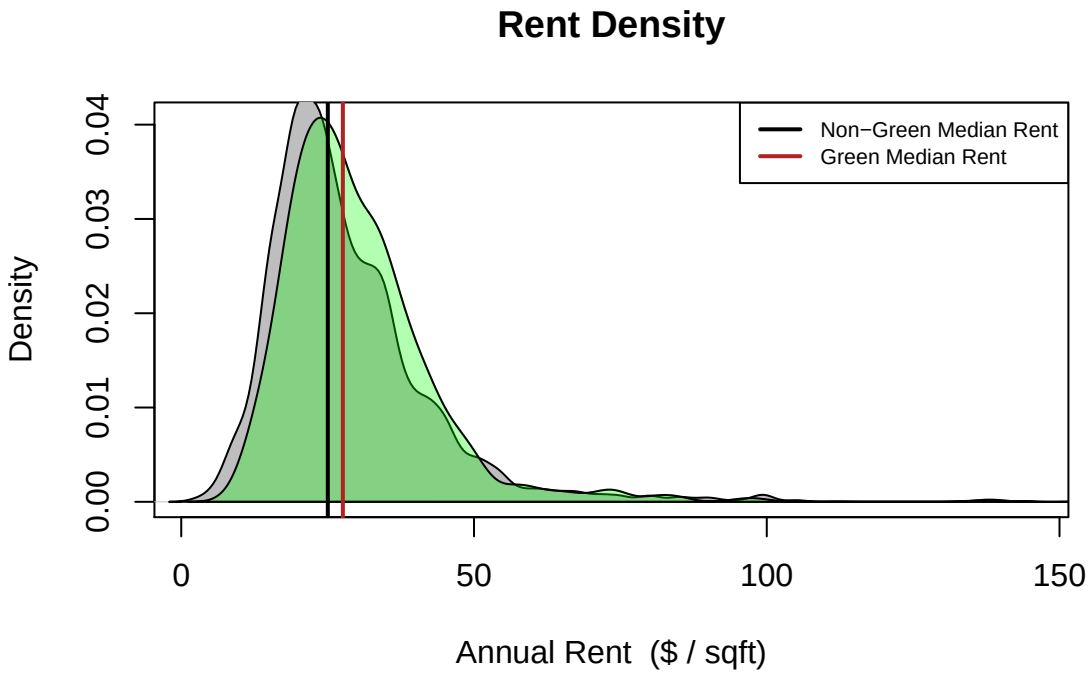
(b) Should We use Median or Mean?

> *"I used the median rather than the mean, because there were still some outliers in the data, and the median is a lot more robust to outliers."*

I decided to use median based on the distribution and size of the dataset. The leasing rate for green buildings is highly left-skewed, so the median is a better estitamation for our building, which is 92.9%.



We can also see that the density plot is right-skewed. Therefore, it might be better to use median to measure the central tendency of the dataset.




```
## [1] "The median market rent in the non-green buildings is $ 25.03 per square foot per
year, while the median market rent in the green buildings is $ 27.6 per square foot per
year: about $ 2.57 more per square foot."
```

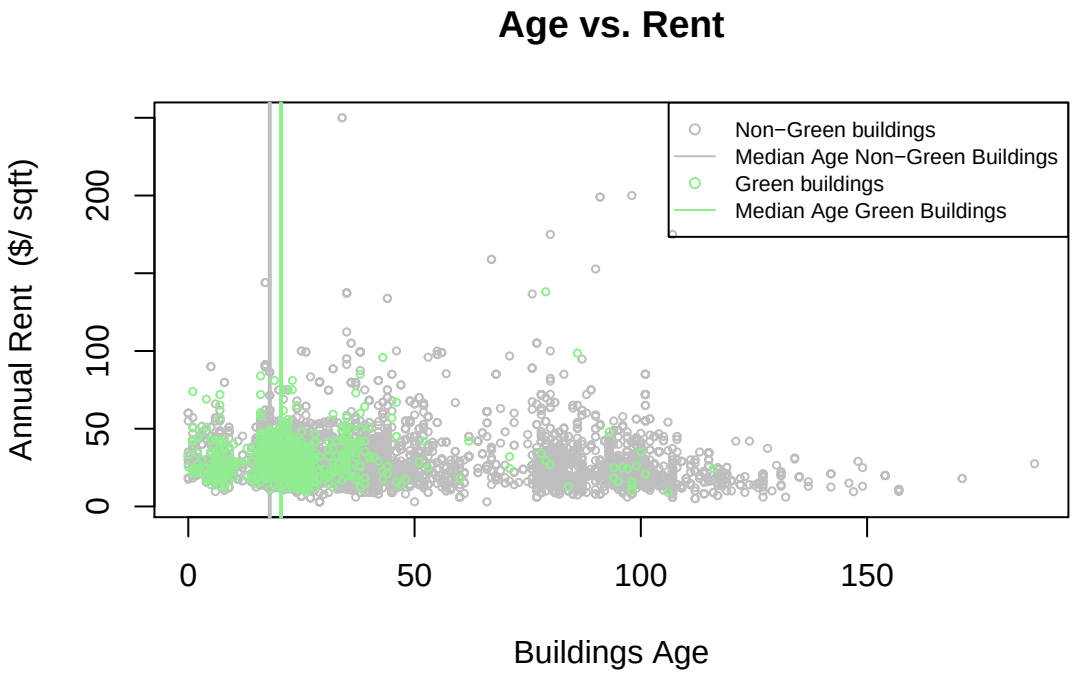
```
## [1] "Because our building would be 250,000 square feet, this would translate into an
additional $ 642500 of extra revenue per year if I build the green building"
```

(c) Confounding Variables

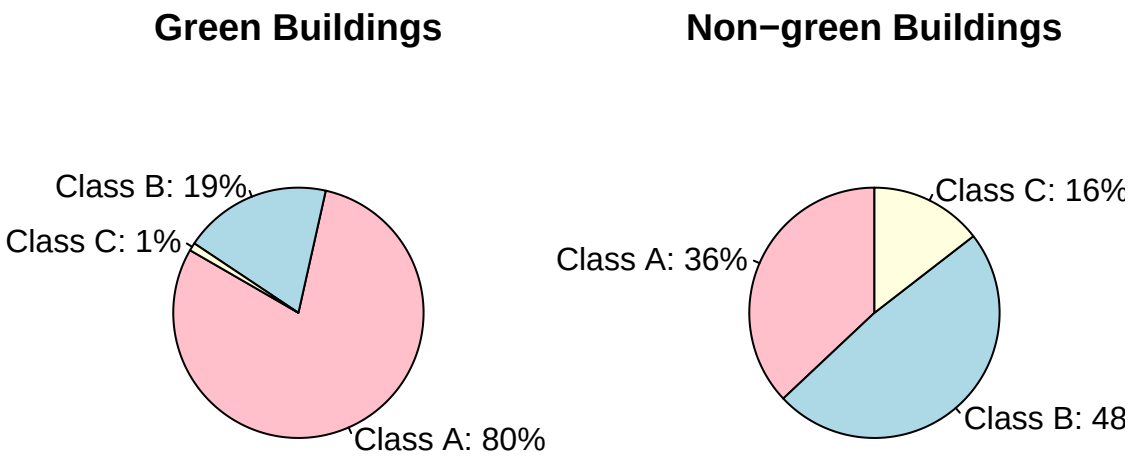
> *"Once I scrubbed these low-occupancy buildings from the data set, I looked at the green buildings and non-green buildings separately. The median market rent in the non-green buildings was $25 per square foot per year, while the median market rent in the green buildings was $27.60 per square foot per year: about $2.60 more per square foot."*

The way the author calculates the premium rent for green buildings is too generic as there are confounding variables. With these confounding variables, we are not sure how the green rating directly influences the rent.

Age is one of the confounding variables. As shown in the plot, green buildings are highly concentrated in the lower range of age, which means they are relatively new, thus having higher rent. I decided to analyze the buildings with ages less than 50.



Another confounding variable is class. As shown in charts below, class A buildings have a definite premium rent over other classes and green buildings have an enormously high percentage falling into class A and class B. Therefore, I removed buildings in class C in our analysis.



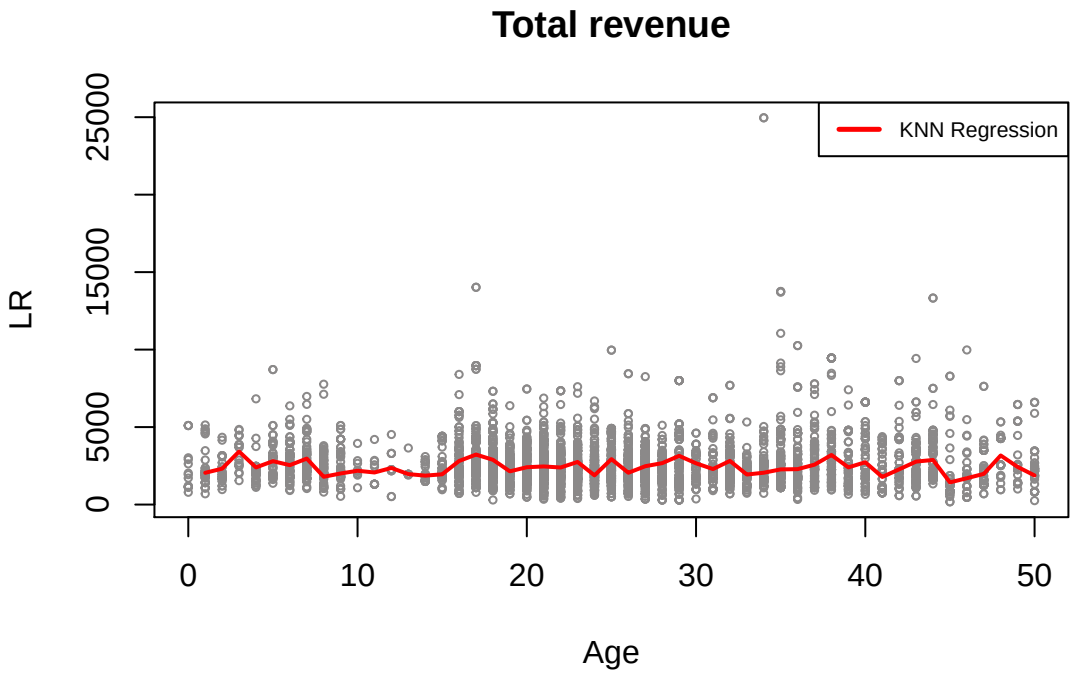
Then I calculate how much the premium in rent is brought by green rating. I first group the buildings based on cluster, and then calculate the difference between the median of green building's rent and that of non-green buildings within the same cluster.

```
## [1] "The mean of difference among all clusters is 2.15776984126984 which is our
expected premium rent."
```

(d) Future Predictions

> *"Based on the extra revenue we would make, we would recuperate these costs in $5000000/650000 = 7.7 years. Even if our occupancy rate were only 90%, we would still recuperate the costs in a little over 8 years."*

It is such a strong assumption that it assumes a constant leasing rate and constant rent over the life cycle of the building. Is that true? I create a new factor, LR, which is leasing_rate $\times$ Rent. With a fixed building size, this feature is proportional to the total leasing revenue. How does it change with age? I selected buildings less than 50 years old and in class A or B and here is the plot:



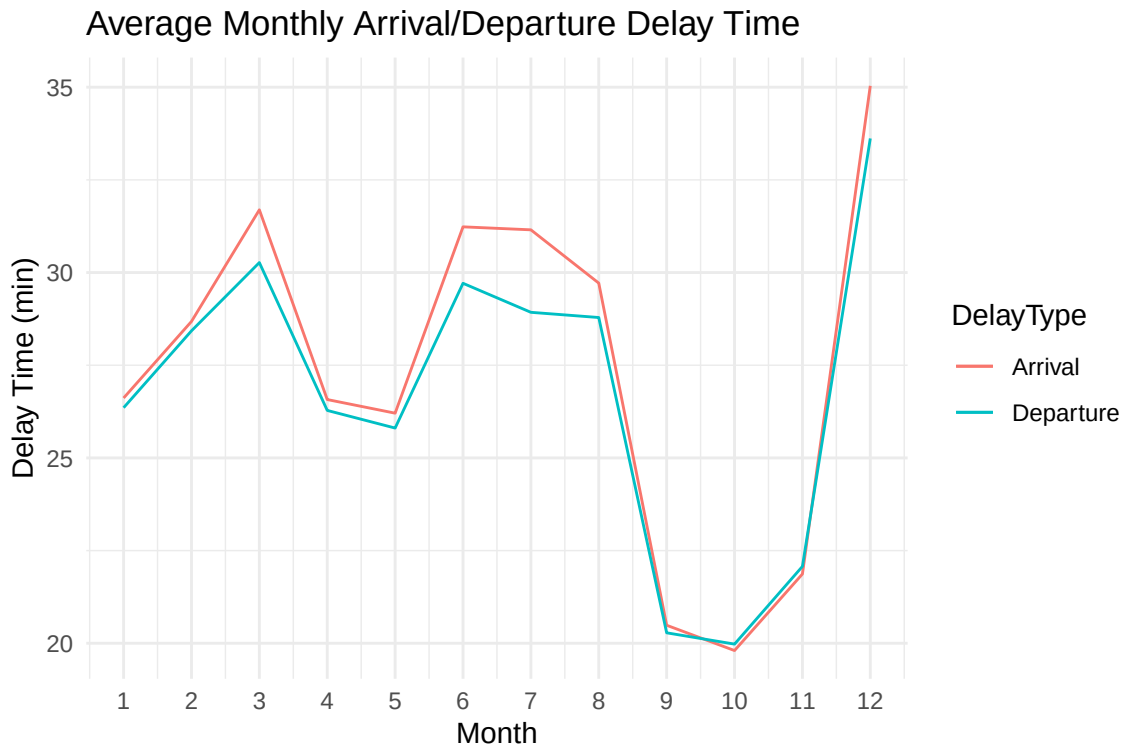
Each building is a gray dot in the plot. I draw a line with KNN (k set to 20) to show a smooth general trend of LR over age. It turns out that the total revenue doesn't go down with an increasing age. Therefore, I could assume that the 2.2 premium for green rating holds for at least the first 50 years.

Besides revenue, the cost for green buildings could potentially be higher than non-green ones. Without enough information to quantify that, I assume that the extra cost is about 5% of total revenue. The median of green buildings' rent is \$30, which makes the cost \$1.5.

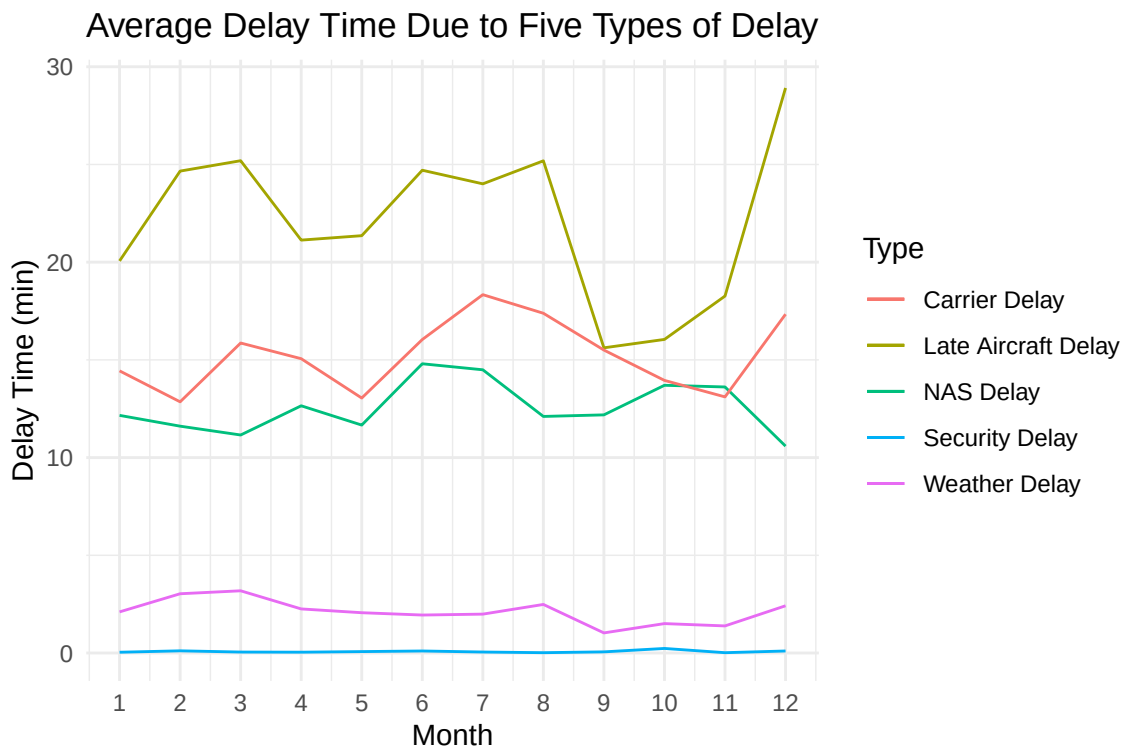
Therefore, the annual extra revenue from green rating is (\$2.2-\$1.5) $\times$ 92.9% $\times$ 250,000 (size) = \$162,757 and it needs \$5,000,000 / \$162,757 = 30 years to recuperate the extra cost. 30 years as the payback period of an investment is too long and makes the company exposed to industry fluctuations and external risks. I would suggest not building the green building.

2. Visual Story Telling Part 2: Flights at ABIA

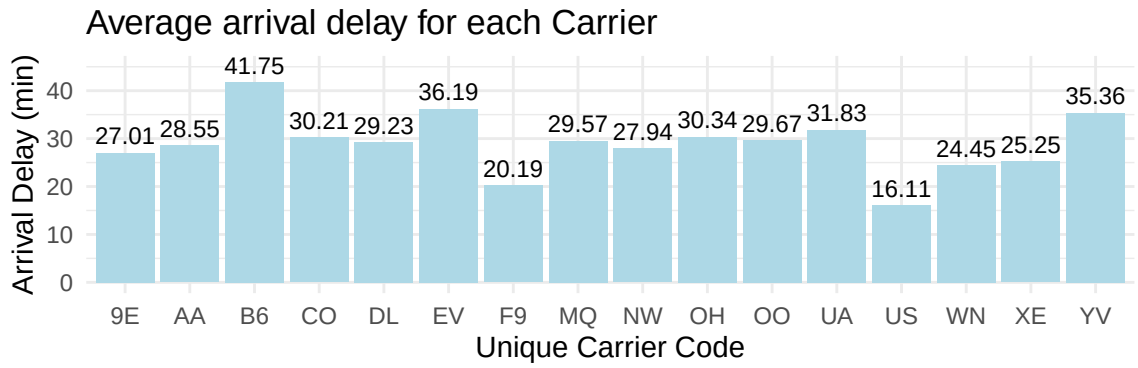
(a) Delay Times

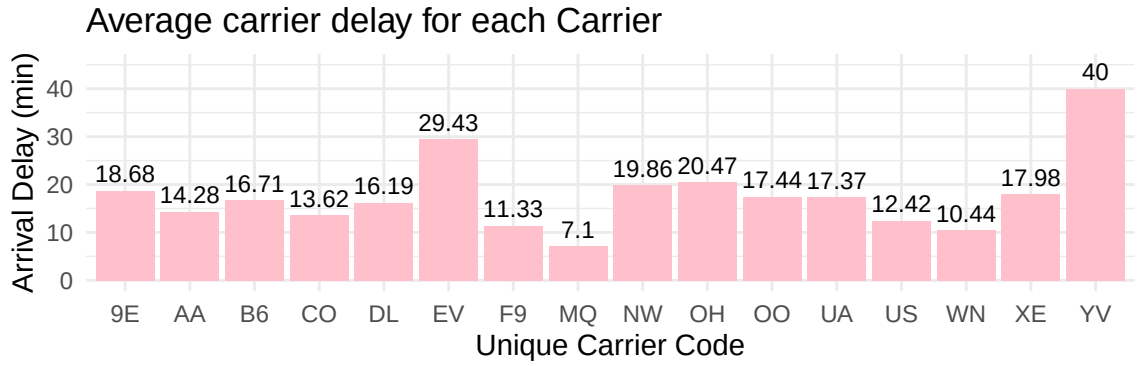


Apparently, months with a higher average arrival delay tend to have a higher average departure delay. This may be because that given flight time period stays the same, departure delays will always lead to arrival delays.



According to this plot, I can see late aircraft delay is much higher than any other delay types except for September and October. Overall, The most delays occur in the winter months, probably because of incliment weather.





From these plots I can see that the higher average arrival delay for each carrier may not be due to their own

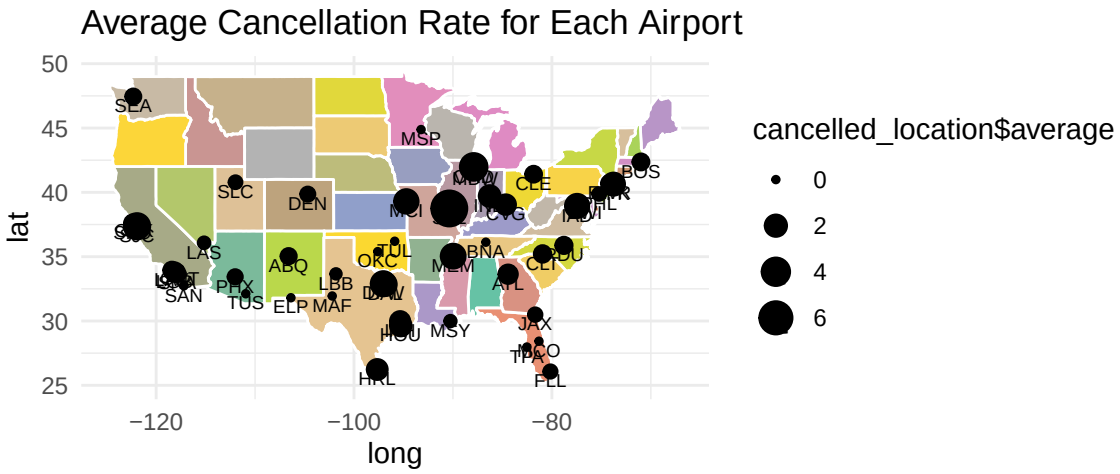
fault. Airline YV and EV have both higher average arrival delay and average carrier delay, so it may be unwise to choose from these two carriers.

AircraftDelay has the largest delay time among all types of reasons. STL also ranks the first among this group.

(b) Cancellation Rate

Therefore, based on the average cancellation rate and the average total delay time for different airports, We can determine that STL is the worst airport to fly in. But one concern is that STL has 95 flights from Austin in our dataset, which is actually not a big destination. If considering the total number of flights, then ORD is also a bad destination.

Plot of Cancellation Rate by Airport (size corresponds to cancllation rate):



Top 5 airports with the average cancellation rate: STL, ORD, SJC, DFW, MEM

3. Portfolio Modeling

I will construct three different portfolios of exchange-traded funds, or ETFs, and use bootstrap resampling to analyze the short-term tail risk of these portfolios.

(a) Characterize the risk/return properties of the five asset classes

My assets:

```
## [1] "SPY" "TLT" "LQD" "EEM" "VNQ"
```

Expected Return of each asset:

ClCl.LQD	ClCl.TLT	ClCl.EEM	ClCl.VNQ	ClCl.SPY
7.08e-05	0.0001961	0.0001969	0.0002627	0.0003005

Now, rank the five asset from lowest return to highest return based on sample mean.

Standard deviation of return for each asset:

ClCl.LQD	ClCl.TLT	ClCl.SPY	ClCl.EEM	ClCl.VNQ
0.005121	0.008973	0.0123	0.01943	0.0205

Here, I rank them from lowest risk to highest risk based on sample standard deviations of the assets. Then we got a rough risk ranking of these ETFs: EEM> VNQ> SPY> TLT> LQD.

From the above tables, I can classify our assets into different categories. Any assets below the 3rd rank will be given a score low. Those above the third rank will be given a score high, and the middle rank will be given a score medium.

SPY - High return/ Medium risk

TLT - Medium return/ Low risk

LQD - Low return/ Low risk

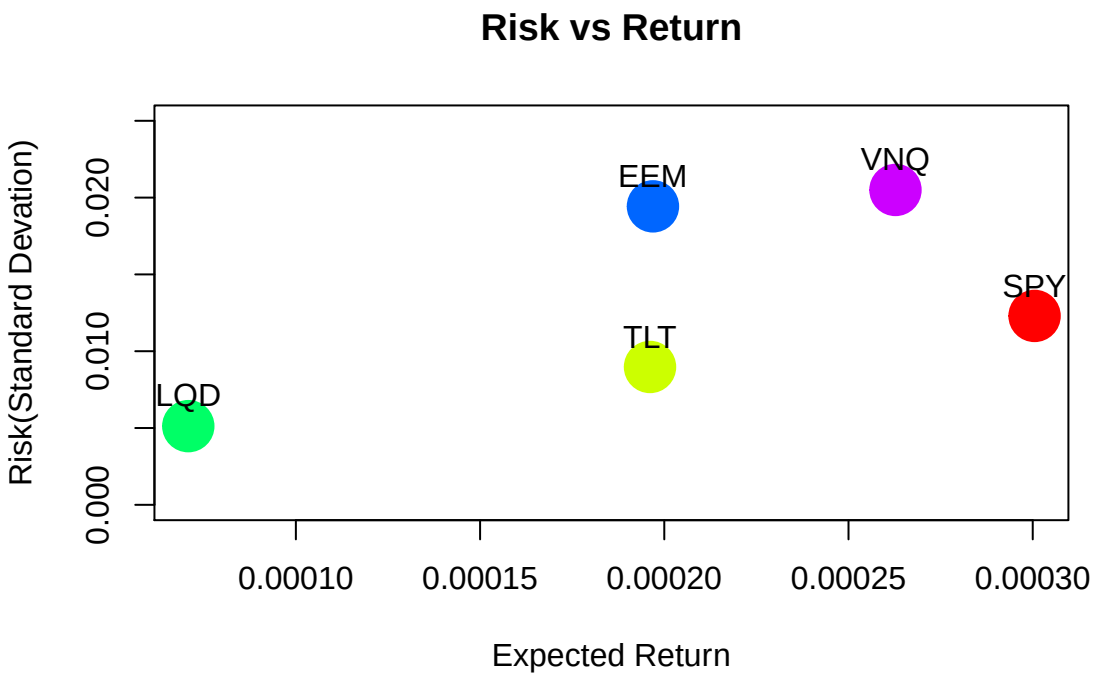
EEM - Low return/ High risk

VNQ - High return/ High risk

Correlation between assets' returns:

	ClCl.SPY	ClCl.TLT	ClCl.LQD	ClCl.EEM	ClCl.VNQ
ClCl.SPY	1	-0.4357	0.09861	0.8665	0.7531
ClCl.TLT	-0.4357	1	0.4428	-0.3694	-0.2418
ClCl.LQD	0.09861	0.4428	1	0.1197	0.07991
ClCl.EEM	0.8665	-0.3694	0.1197	1	0.6801
ClCl.VNQ	0.7531	-0.2418	0.07991	0.6801	1

I will decide our not only on asset's expected return and standard deviation but also on its correlation with other assets. On one hand, if an asset has high positive correlation with another asset, that means they will make a riskier combination. On the other hand, if an asset has negative correlation with another asset, they will make a safer combination.



(b) Bootstrapping

Setting values for our simulation:

I have 100,000 to invest, and I will do our simulation for 20 days.

For each of the strategy, I will adjust the weight accordingly.

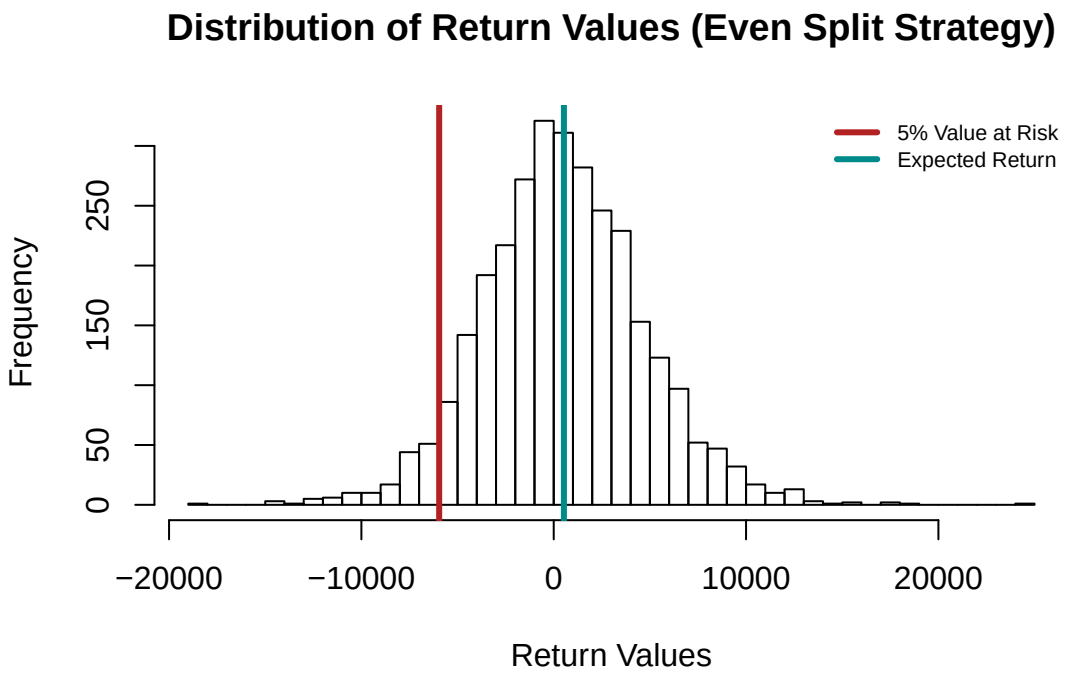
(b1) Even Split Strategy

For this first porfolio strategy, I will assign equal weights to all five assets.

Weight of each stock for the even split strategy:

SPY	TLT	LQD	EEM	VNQ
0.2	0.2	0.2	0.2	0.2

Distribution of return values for even split strategy:



Value at Risk at 5%	Expected Return	Standard Deviation of Return
-5956	529.4	4196

This shows us that if investors invest for 20 traiding days for this portfolio, 5 percent of the time they will suffer a loss of 5956. However, on average, they will receive around 530.

Quantile Values:

0%	25%	50%	75%	100%
-18364	-2131	424.4	3120	24347

The table suggests that the return value in for 20 trading days can range from a loss of 18364 to a gain of 24347.

(b2) Safe Strategy

For this strategy, I will look at our classification of the five assets and choose those with low risk properties. I will also include one medium risk asset. For the weight, I will use 1/standard deviation as the coefficients and normalize them to add up to 1. SPY, TLT, and LQD are the three chosen assets.

To find the safe portfolio, we can:
1.use the funds with smaller variances (also relatively lower returns) like LQD, TLT, and SPY
2.choose the funds that have low correlations between them, especially consider using TLT together with other funds (hedging)

Therefore, we try allocating 40% of asset in LQD, 30% of asset in TLT, 30% of asset in SPY

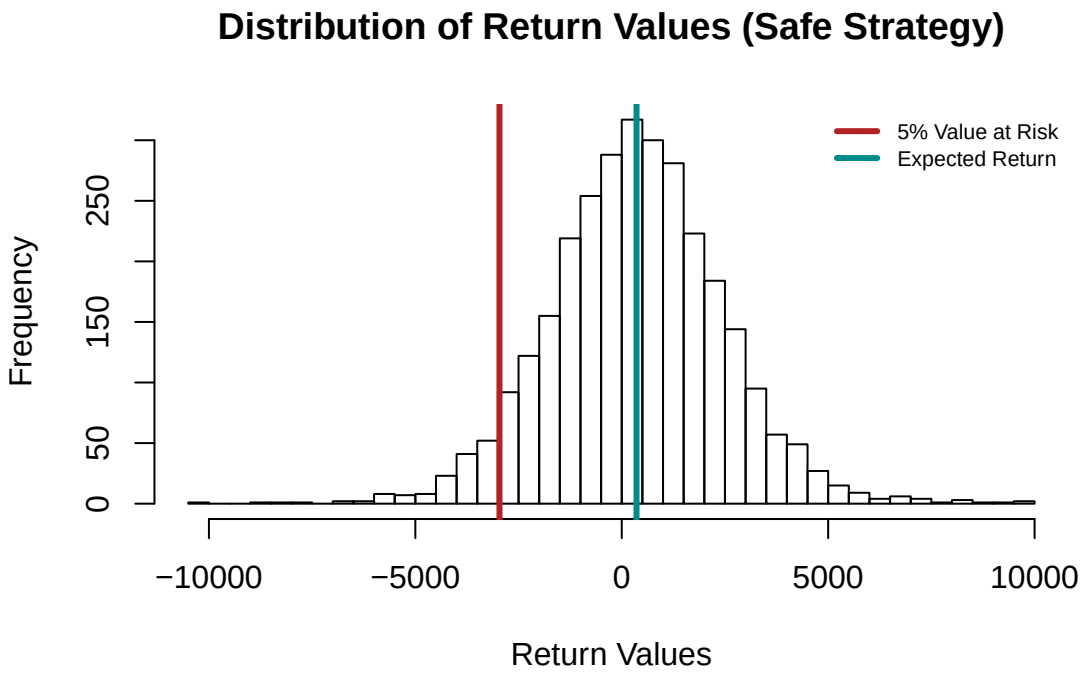
	ClCl.SPY	ClCl.TLT	ClCl.LQD	ClCl.EEM	ClCl.VNQ
ClCl.SPY	1	-0.4357	0.09861	0.8665	0.7531
ClCl.TLT	-0.4357	1	0.4428	-0.3694	-0.2418
ClCl.LQD	0.09861	0.4428	1	0.1197	0.07991
ClCl.EEM	0.8665	-0.3694	0.1197	1	0.6801
ClCl.VNQ	0.7531	-0.2418	0.07991	0.6801	1

Among the asset that has low standard deviation, I chose SPY, TLT, and LQD in our safe strategy because SPY and TLT have -0.44 correlation coefficient, suggesting a negative correlation. LQD and SPY have almost 0 correlation coefficient. Finally, LQD and TLT have about 0.4 correlation coefficient. It might seems counterintuitive at first that I pick this asset. However, other combinations will include an asset that has high correlation with SPY. As a result, I select LQD, TLT, and SPY.

Weight of each stock for the safe strategy:

SPY	TLT	LQD	EEM	VNQ
0.2096	0.2872	0.5032	0	0

Distribution of return values for safe strategy:



Value at Risk at 5%	Expected Return	Standard Deviation of Return
-2962	357.9	2088

This shows us that if investors invest for 20 traiding days for this portfolio, 5 percent of the time they will suffer a loss of 3218. However, on average, they will receive around 261.

Quantile Values:

0%	25%	50%	75%	100%
-10140	-966.4	338.7	1647	9571

The table suggests that the return value in for 20 trading days can range from a loss of 9046 to a gain of 8677.

(b3) Aggressive Strategy

As for discovering the aggressive portfolio, we have following strategies:
1.use the funds with the biggest variances (also the highest returns) like EEM, VNQ, and SPY
2.choose the funds that have high correlations between them, specifically we should exclude TLT
3.choose only very few funds so that the risk can not be shared
Thus here we allocate 80% of asset in EEM and 20% of asset in VNQ

	ClCl.SPY	ClCl.TLT	ClCl.LQD	ClCl.EEM	ClCl.VNQ
ClCl.SPY	1	-0.4357	0.09861	0.8665	0.7531
ClCl.TLT	-0.4357	1	0.4428	-0.3694	-0.2418
ClCl.LQD	0.09861	0.4428	1	0.1197	0.07991
ClCl.EEM	0.8665	-0.3694	0.1197	1	0.6801
ClCl.VNQ	0.7531	-0.2418	0.07991	0.6801	1

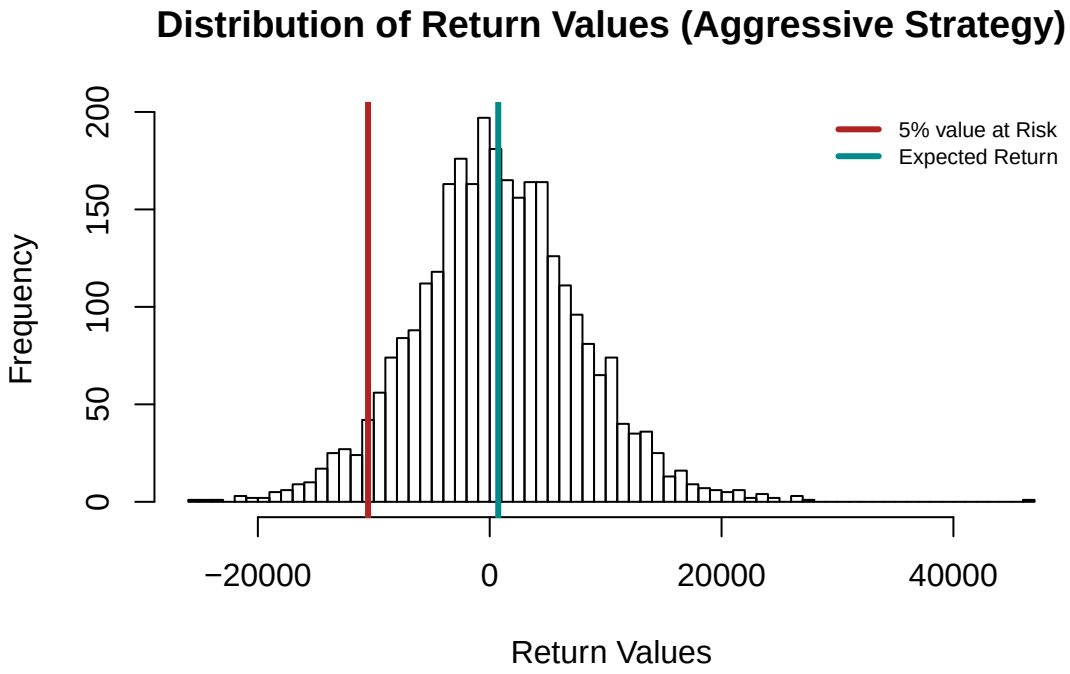
For this strategy, I will not be as diversified as the safe strategy. Also, I will look mainly at assests which have high returns with moderate to high risks. Coefficients will be adjusted based on the expected return values.

I chose SPY and VNQ because they both have high returns and moderate to high risk. They also have a positive correlation of 0.753, meaning that they are risky but can yield high returns.

Weight of each stock for the aggressive strategy:

SPY	TLT	LQD	EEM	VNQ
0.4665	0	0	0	0.5335

Distribution of return values for agressive strategy:



Value at Risk at 5%	Expected Return	Standard Deviation of Return
-10494	737.5	7114

This shows us that if investors invest for 20 traiding days for this portfolio, 5 percent of the time they will suffer a loss of 2962. However, on average, they will receive around 358.

Quantile Values:

0%	25%	50%	75%	100%
-25197	-3752	512.9	5106	46186

The table suggests that the return value in for 20 trading days can range from a loss of 10140 to a gain of 9571.

(d) Summary

Table 15: Table continues below

	Value at Risk at 5%	Expected Return
Split Strategy	-5956	529.4
Safe Strategy	-2962	357.9
Aggressive Strategy	-10494	737.5

	Standard Deviation of Return
Split Strategy	4196
Safe Strategy	2088
Aggressive Strategy	7114

4. Market Segmentation

(a) Data Pre-Processing

I delete the following four categories: 'spam','adult','uncategorize', and 'chatter in order to make our market segmentation more meaningful, spam, adult, uncategorize, and chatter. Since adult and spam are categories that are supposed to be filtered and contain improper contents, and uncategorize and chatter have no special meanings. Therefore, I delete these four columns.

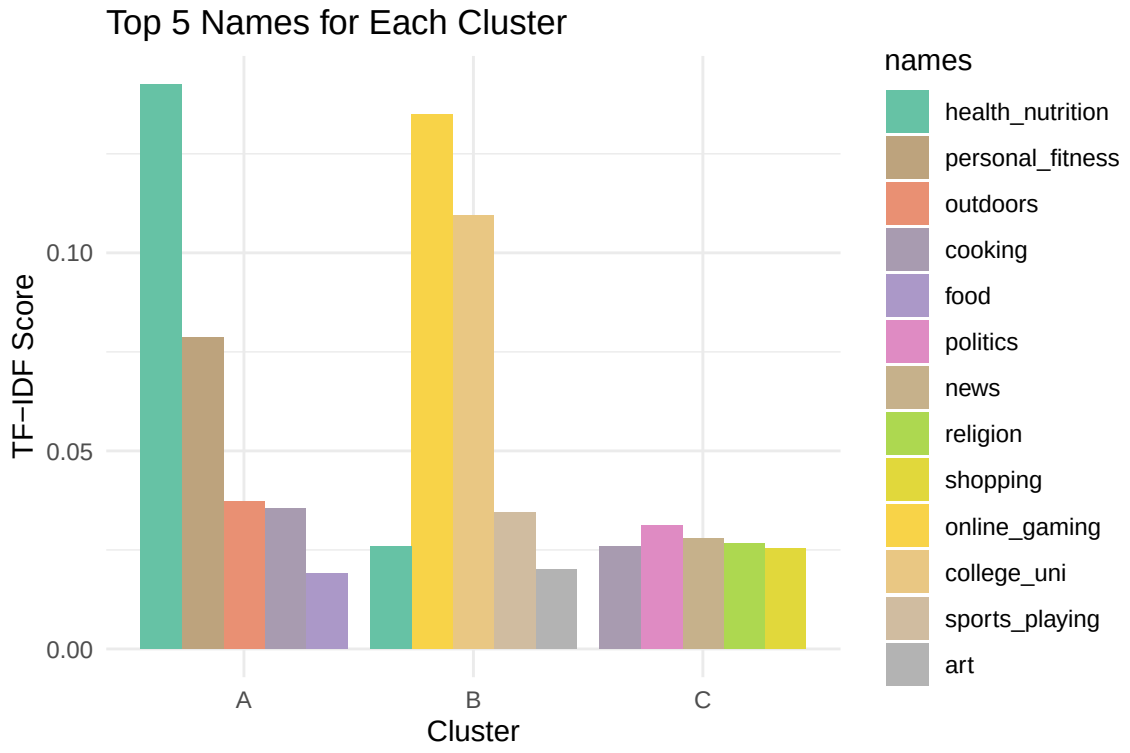
Since certain high-frequency terms have little discriminating power like photo-sharing, I use TF-IDF to recalculate the weight of each term for every follower. TF stands for term-frequency, measuring how frequent a term occurs in a follower's tweets: the more frequent a term occurs, the more important it is to the follower; IDF stands for inverse-document-frequency, measuring how frequent the term occurs in the whole dataset: the more frequent a term occurs, the less important it is to every follower.

I use 'cosine' as a measurement for the similarity. It calculates the cosine of the angle between two vectors It measures difference in orientation instead of magnitude. For example, I have 3 follower A,B,C with features like A={'travelling':10,'cooking':5}, B={'travelling':20,'cooking':10}, C={'travelling':10,'cooking':12}, I would consider A more similar with B than C even though A and C are 'closer'.

(b) Define Market Segment

I will define a "market segment" as a cluster of correlated interests.

By looking at different outputs of different Ks, I chose k=3 as our final parameter since its output makes more sense to us.



Top 5 Names per Cluster:

A	B	C
health_nutrition	online_gaming	politics

A	B	C
personal_fitness	college_uni	news
outdoors	sports_playing	religion
cooking	health_nutrition	cooking
food	art	shopping

From the topics of high TFIDF-socres in the clusters, I can infer that first cluster represents people who care a lot about health and fitness, mostly likely to be well-educated people and housewives; the second cluster represents college/high school students; the third cluster represents people who care about current events, most likely working people.

(c) Marketing Strategy for Each Group:

- *Cluster one:* I recommend the company could post some healthy cooking recipes which use company's products, and the company can cooperate with some famous chefs to promote their products.
- *Cluster two:* The company should launch interesting social media campaigns to attract this market segment, such as campaigns combining simple gaming and promotions together.
- *Cluster three:* The company can sponsor some social events or even make some political contributions to improce their social exposures on newspaper, TV and news website to target this group.

5. Author Attribution

(a) Data Pre-Proccessing

```
## NULL
```

```
## <<DocumentTermMatrix (documents: 2500, terms: 3325)>>
## Non-/sparse entries: 376957/7935543
## Sparsity           : 95%
## Maximal term length: 20
## Weighting          : term frequency (tf)
```

```
##    Mode   FALSE    TRUE
## logical   29264    3325
```

There are 32,589 words in document-term matrix for the test data, however there are only 3325 words which are also common in the train data set. So, let us drop the remaining words for the classification problem. This is however not an optimal solution, as I am dropping many words.

```
##    Mode    TRUE
## logical    3325
```

Create the TF-IDF matrix for test and train data:

3325 words are still high to conduct classification. Thus I will reduce the dimensions using Principal Component Analysis. I will run the PCAs on the train data set and take the top words which explain 75% of the variability in data.

Run PCA on TF-IDF to reduce the number of words:

Based on the PCA on train data, I can say that 330 words define 75% of the variability. Using the PCAs on the train data set, I predicted the PCAs on test data set. I will use these data sets for our classification of the authors.

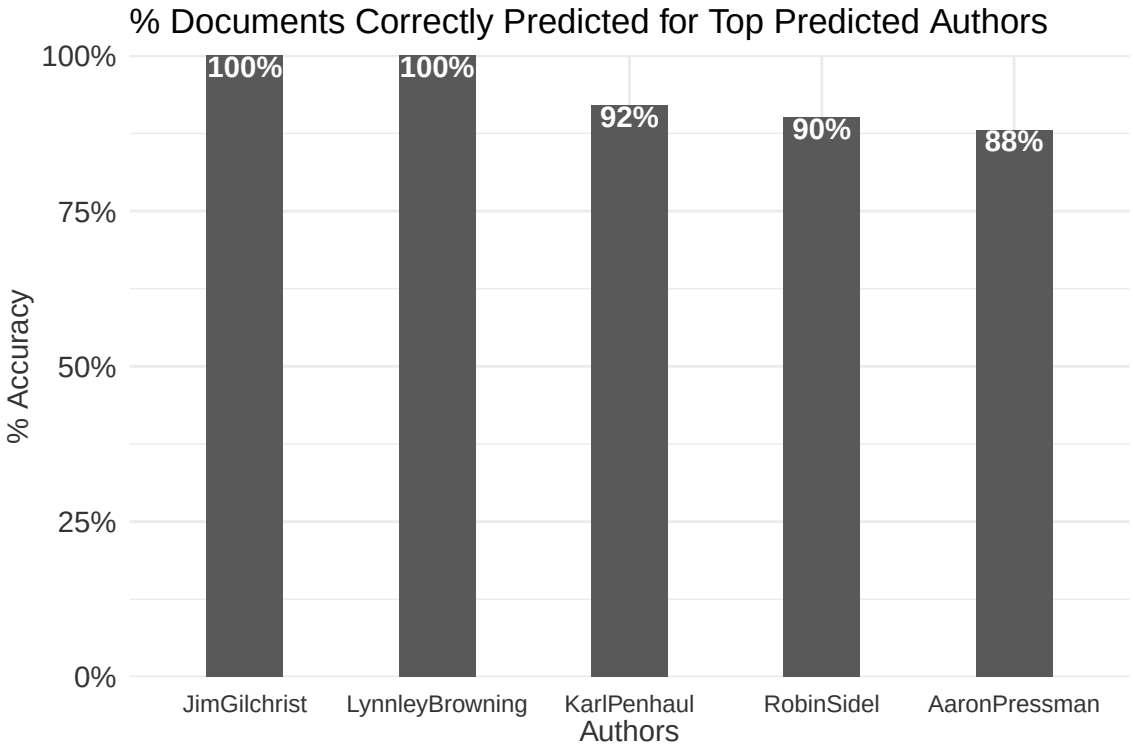
(b) Classification - 1: Random Forest.

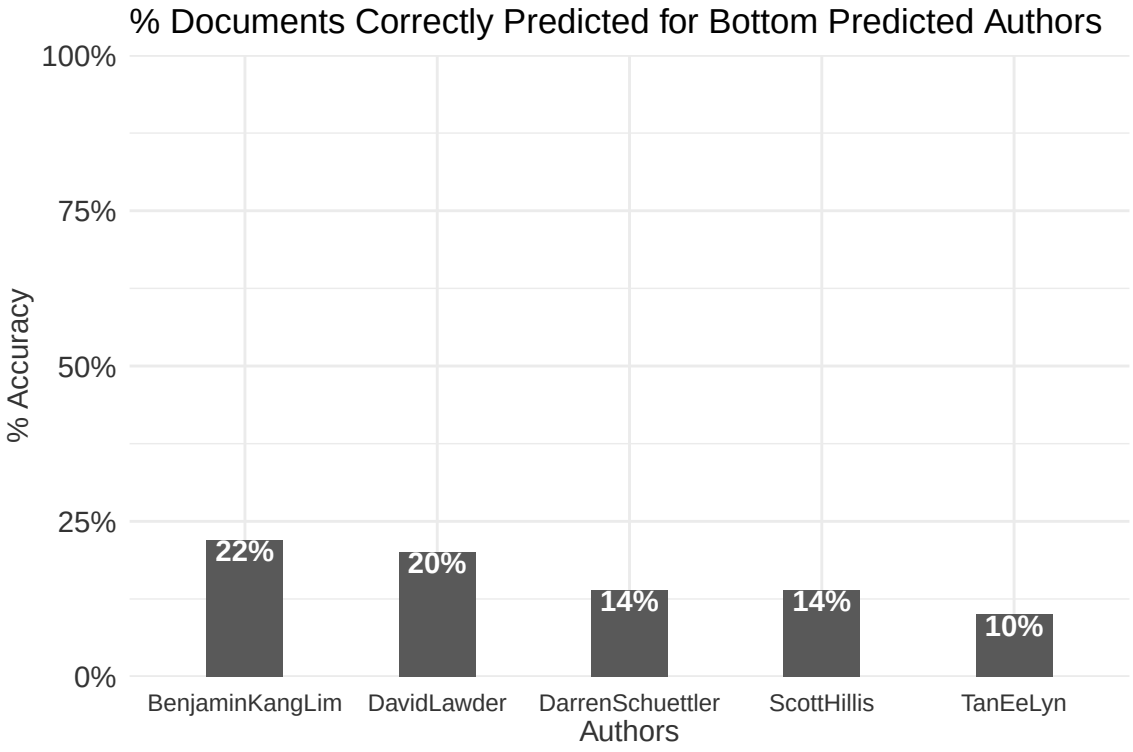
After running Random Forest I have an acuracy of 59%.

```
## [1] 0.5896
```

However, let's look at how the accuracy varies for different authors:

Authors	Correct_Predictions	percentage_accuracy
JimGilchrist	50	1
LynnleyBrowning	50	1
KarlPenhaul	46	0.92
RobinSidel	45	0.9
AaronPressman	44	0.88
MatthewBunce	43	0.86
SimonCowell	43	0.86
FumikoFujisaki	41	0.82
JoWinterbottom	41	0.82
NickLouth	41	0.82





Authors most *correctly* predicted by Random forest: JimGilchrist, LynnleyBrowning, KarlPenhaul, RobinSidel, MatthewBunce, NickLouth.

Authors most *incorrectly* predicted: TanEeLyn, ScottHillis, EdnaFernandes, BenjaminKangLim, DarrenSchuettler.

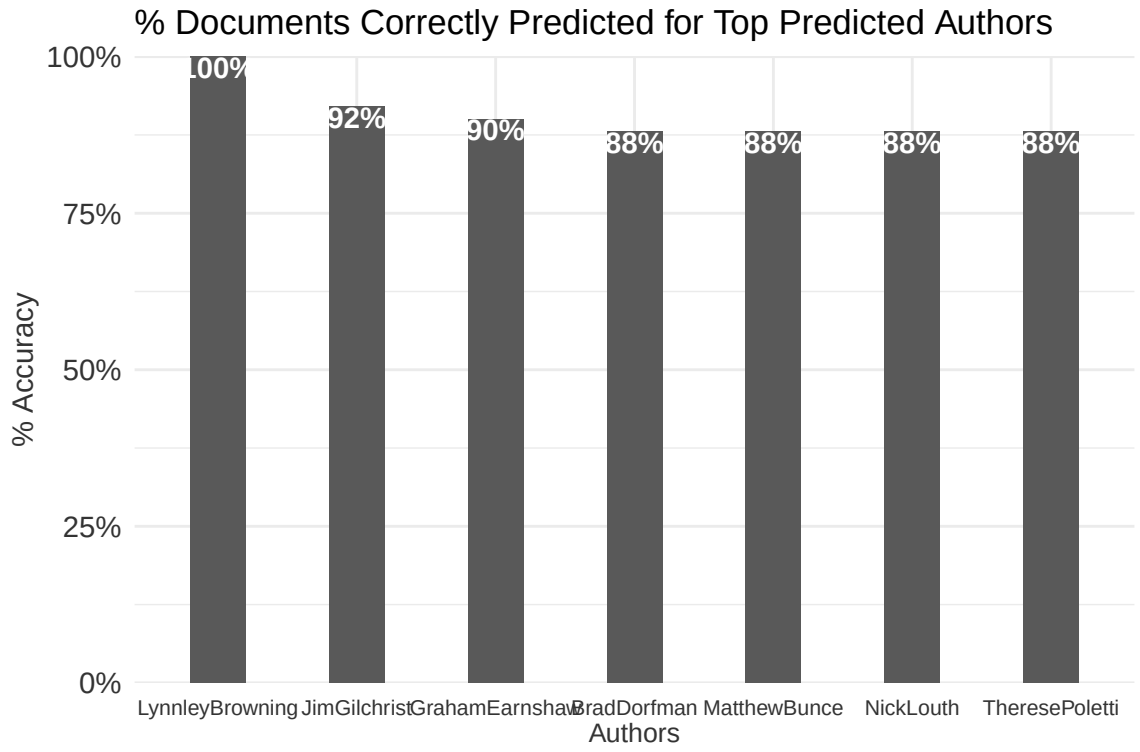
(c) Classification - 2: Support Vector Machine (SVM)

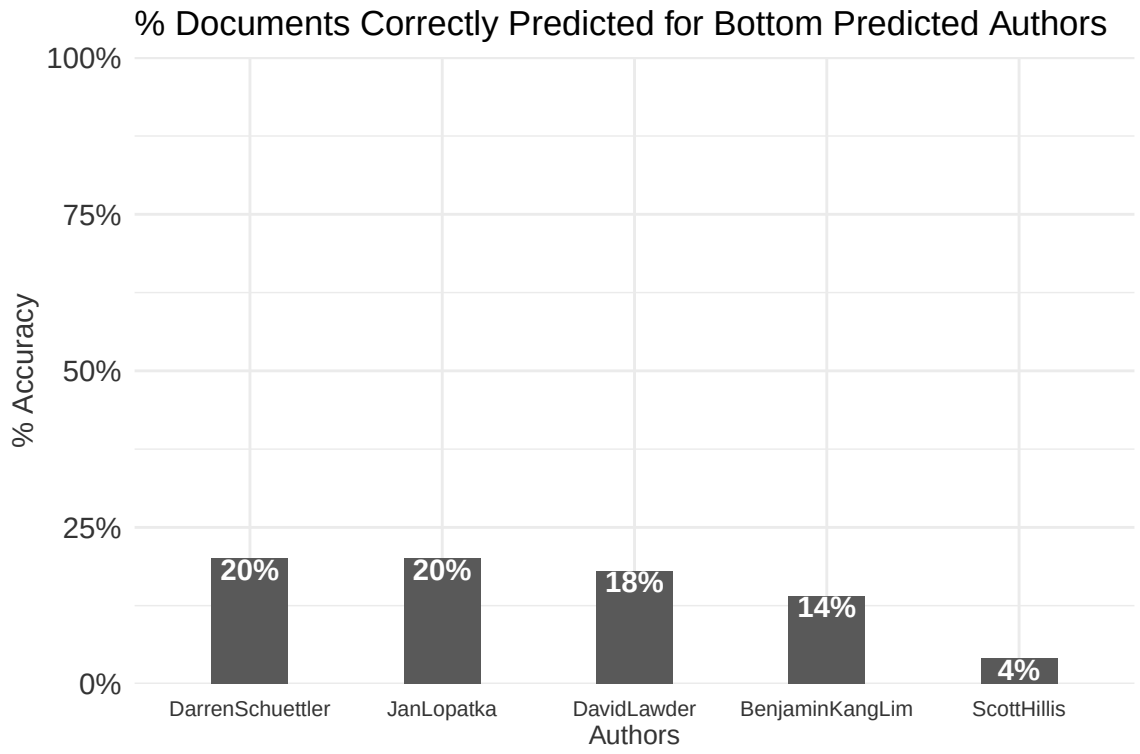
```
## [1] 0.5692
```

From SVM as well I get an accuracy of 57%.

Interestingly, I get similar accuraies in the SVM model as well. Additionally, the run time for SVM is much lower than Random Forest algorithm.

Authors	Correct_Predictions	percentage_accuracy
LynnleyBrowning	50	1
JimGilchrist	46	0.92
GrahamEarnshaw	45	0.9
BradDorfman	44	0.88
MatthewBunce	44	0.88
NickLouth	44	0.88
TheresePoletti	44	0.88
FumikoFujisaki	40	0.8
PeterHumphrey	40	0.8
KirstinRidley	39	0.78





Authors most *correctly* predicted by SVM: LynnleyBrowning, JimGilchrist, GrahamEarnshaw, BradDorfman, MatthewBunce, NickLouth, TheresePoletti.

Authors most *incorrectly* predicted: ScottHillis, DarrenSchuettler, DavidLawder, JanLopatka, BenjaminKangLim.

(d) Summary

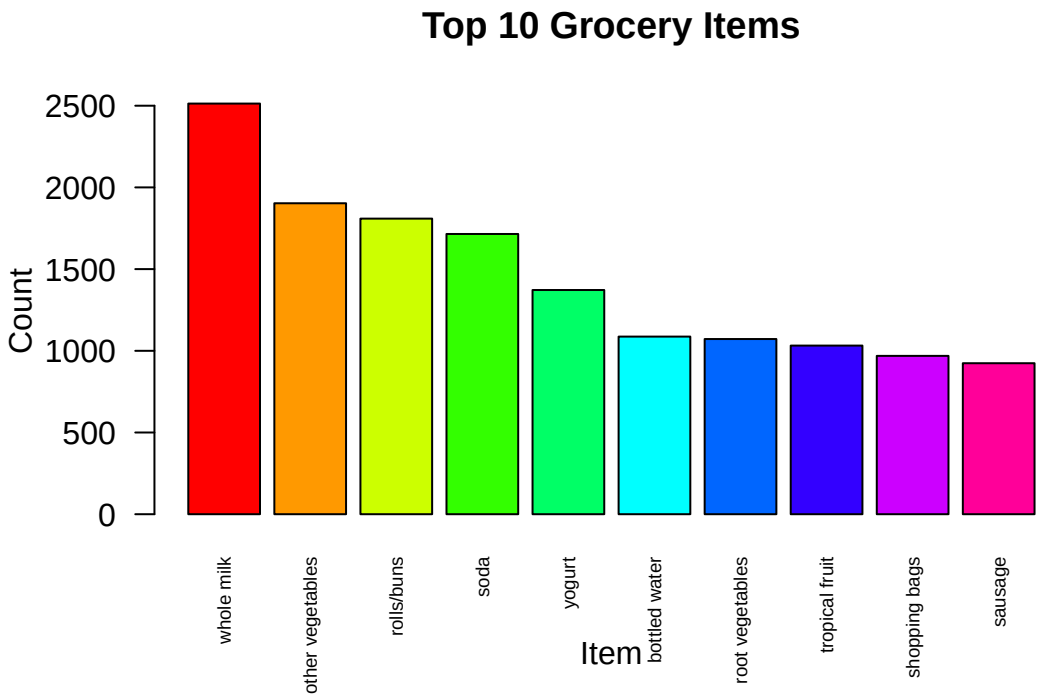
Overall, the outputs of the 2 model does give a similar accuracy of ~58%. While this is not impressive, I do get a lot of authors which have very hgh accuracies in both the models. Overall accuracy seems to be low as there are some authors who are not predicted that well. One potential reason behind this could be that I have dropped quite a few words form the train and test data sets. However, there are many more words in the test data (which if incorporated could improve accuracies).

6. Association Rule Mining

(a) Data Pre-Proccessing

user	value
1	citrus fruit
1	semi-finished bread
1	margarine
1	ready soups
2	tropical fruit
2	yogurt

The table above shows the head of transaction dataframe before splitting it by transactions.



(b) Apriori Algorithm

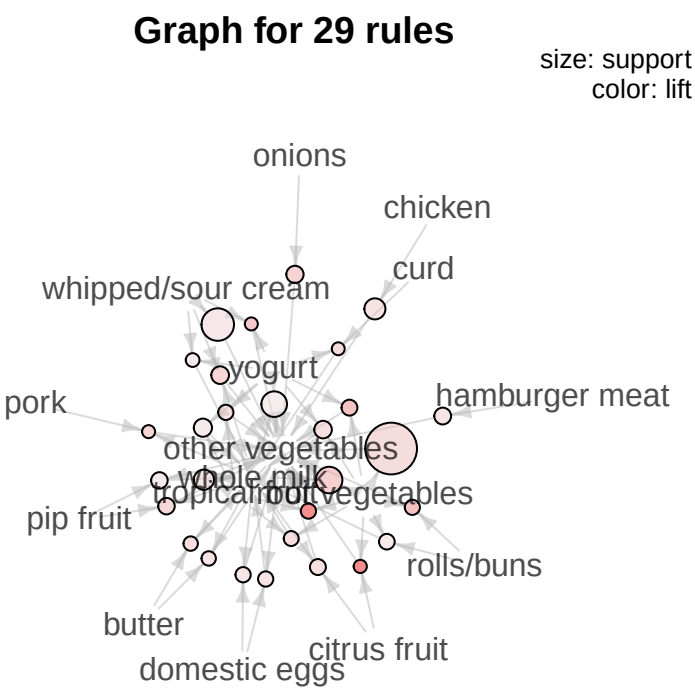
The Apriori Algorithm expects a list of baskets in a special format. In this case, one "transaction" of items per user.

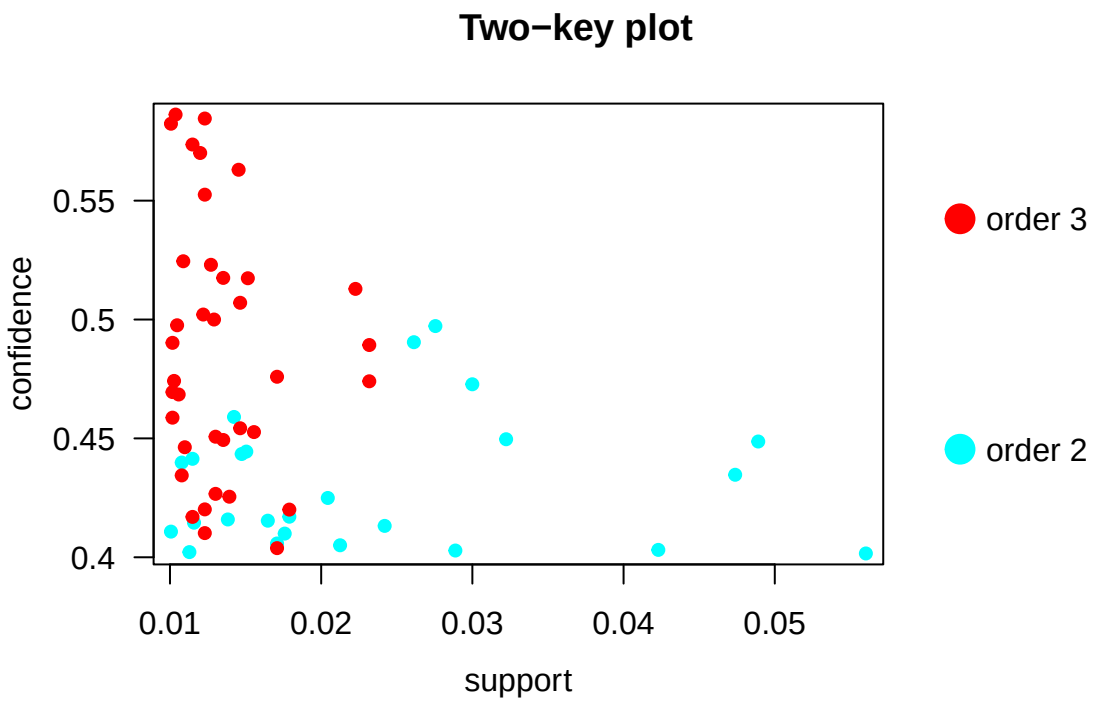
```
## [1] "There are  9835 transactions."
```

I ran a loop with values for support ranging from 0.009 to 0.05 and confidence from 0.2 to 0.5. For these different combinations, I looked for that one that gave us max average lift, which means that there is a high association betwwen the items in the basket. Our goal was to get a high lift value with maximum support. The results I got were best for support= 0.009 and confidence =0.5 with a max average lift of 2.2255.However, increasing the support will ensure higher transactions containing items of interest. The trade off here could be the decrease in lift, which what I see here. But, a slightly higher support ensures many more transactions/rules with a minimum effect on lift. Thus, I decided to choose support to be in between the two values, at 0.01 and confidence a little lower at 0.4.

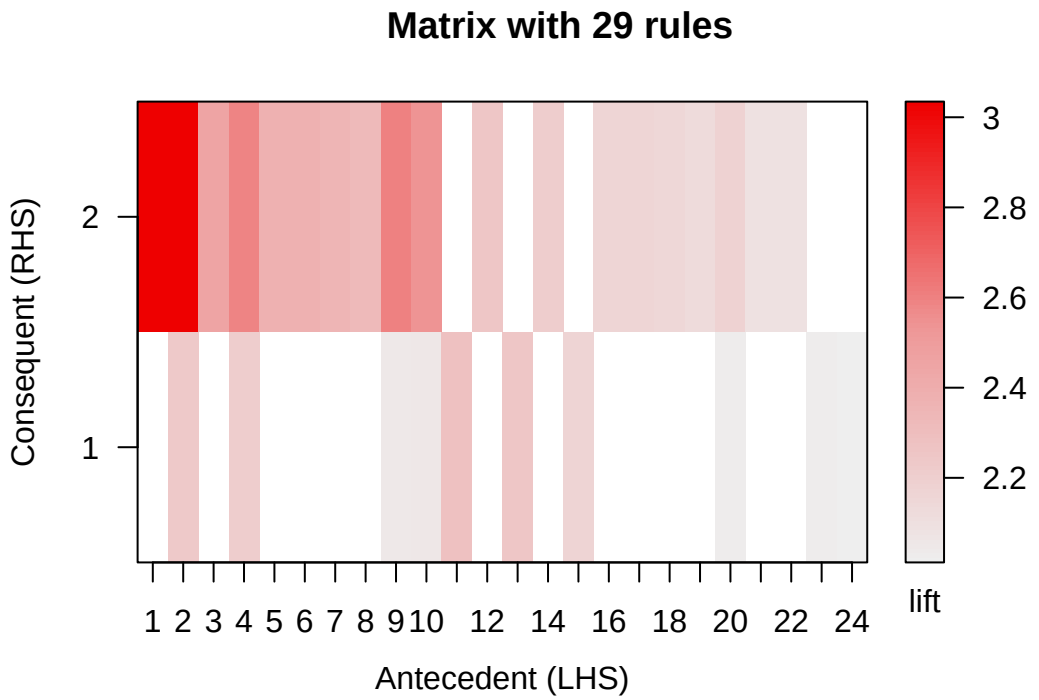
I then again ran the model with chosen values of support and confidence and took a subset of only those rules whose lift was greater than 2 since the mean was very close to 2, it could have given us less associated rules as well. This gave us set of 29 rules with strong association. Out of the groups, whole milk seems to come up the most followed by other vegetables. A large % of people with various baskets are almost always interested in buying whole milk and/or other vegetables.

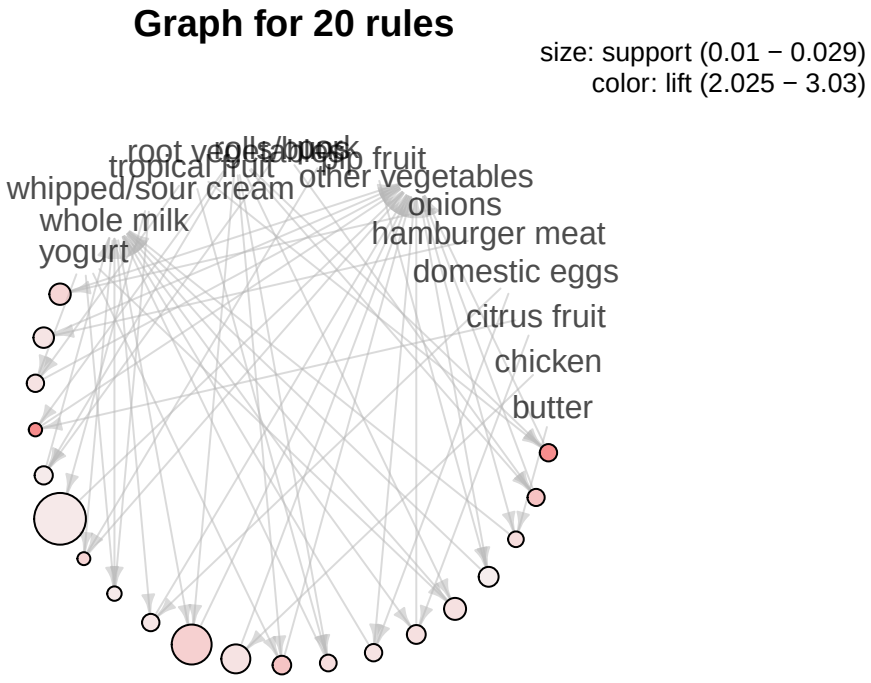
```
## Available control parameters (with default values):
## main  =  Graph for 29 rules
## nodeColors    =  c("#66CC6680", "#9999CC80")
## nodeCol   =  c("#EE0000FF", "#EE0303FF", "#EE0606FF", "#EE0909FF", "#EE0C0CFF", "#EE0F0FFF", "#EE121
## edgeCol   =  c("#474747FF", "#494949FF", "#4B4B4BFF", "#4D4D4DFF", "#4F4F4FFF", "#515151FF", "#535353
## alpha     =  0.5
## cex   =  1
## itemLabels    =  TRUE
## labelCol  =  #000000B3
## measureLabels     =  FALSE
## precision     =  3
## layout    =  NULL
## layoutParams  =  list()
## arrowSize     =  0.5
## engine    =  igraph
## plot  =  TRUE
## plot_options  =  list()
## max   =  100
## verbose   =  FALSE
```






```
## Itemsets in Antecedent (LHS)
##  [1] "{citrus fruit,root vegetables}"   "{root vegetables,tropical fruit}"
##  [3] "{root vegetables,whole milk}"     "{root vegetables,yogurt}"
##  [5] "{onions}"                         "{pork,whole milk}"
##  [7] "{whipped/sour cream,whole milk}"  "{pip fruit,whole milk}"
##  [9] "{rolls/buns,root vegetables}"     "{whipped/sour cream,yogurt}"
## [11] "{curd,yogurt}"                    "{root vegetables}"
## [13] "{butter,other vegetables}"        "{citrus fruit,whole milk}"
## [15] "{domestic eggs,other vegetables}" "{chicken}"
## [17] "{butter,whole milk}"              "{hamburger meat}"
## [19] "{domestic eggs,whole milk}"       "{tropical fruit,yogurt}"
## [21] "{tropical fruit,whole milk}"      "{whipped/sour cream}"
## [23] "{other vegetables,pip fruit}"     "{other vegetables,yogurt}"
## Itemsets in Consequent (RHS)
## [1] "{whole milk}"       "{other vegetables}"
```





The visualizations above gives us the strength f the associations. The first graph gives us a depiction of the importance of the various basket items. Whole milk and other vegetables that came us to be most common are in the middle with branches extending outwards to other items. The next one gives us a two-key plot, not for only the subset but the whole set of values as a function of support and confidence. The final graphis a matrix representation of the matrix of rules with the color scale showing the lift. I can match these to the

lift values above and get the exact items in the basket.

(c) Choice of parameters

I chose support= 0.009 because higher levels of support gave too few rules for us to inspect. I chose confidence = 0.5 because I want to make sure that if iem on rhs appears, item on lhs will also appear. However, this only accounts for how popular the items on rhs are, but not those on the lhs. If rhs items appear regularly in general, there is a greater chance that items on the rhs will contain items on the lhs. To account for this bias, I select our final itemlists based on lift since lift measures how likely item on lhs is purchased when item rhs is purchased. For these chosen values of support and lift we get a max average lift of 2.2255. Therefore, I sort the items by lift and rank the top 20 rules, which is the result generated by the algorithm.

(d) Recommendation

This information would be valuable to store managers planning inventory for these perishable items. Also, this information can be used for product placement strategy. Grouping items frequently bought together on the same shelf or aisle.