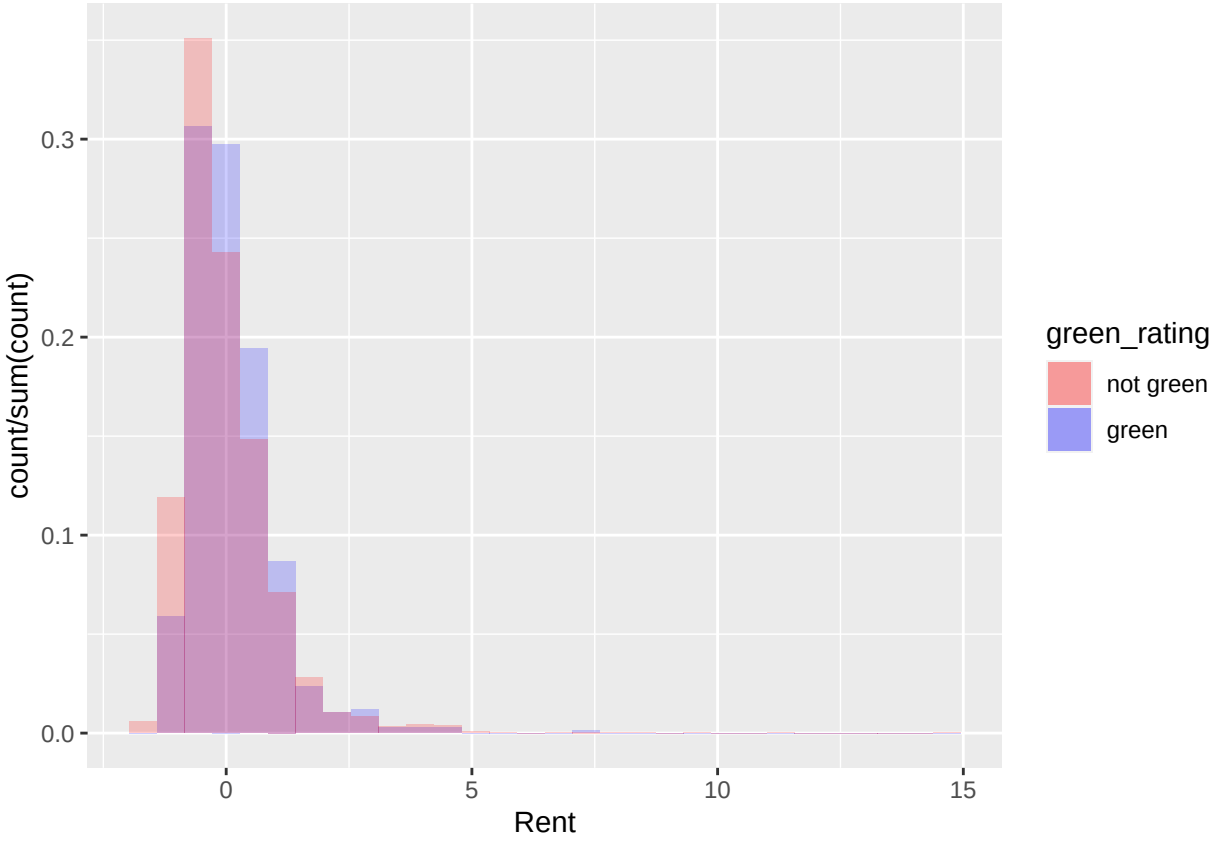
Exercises

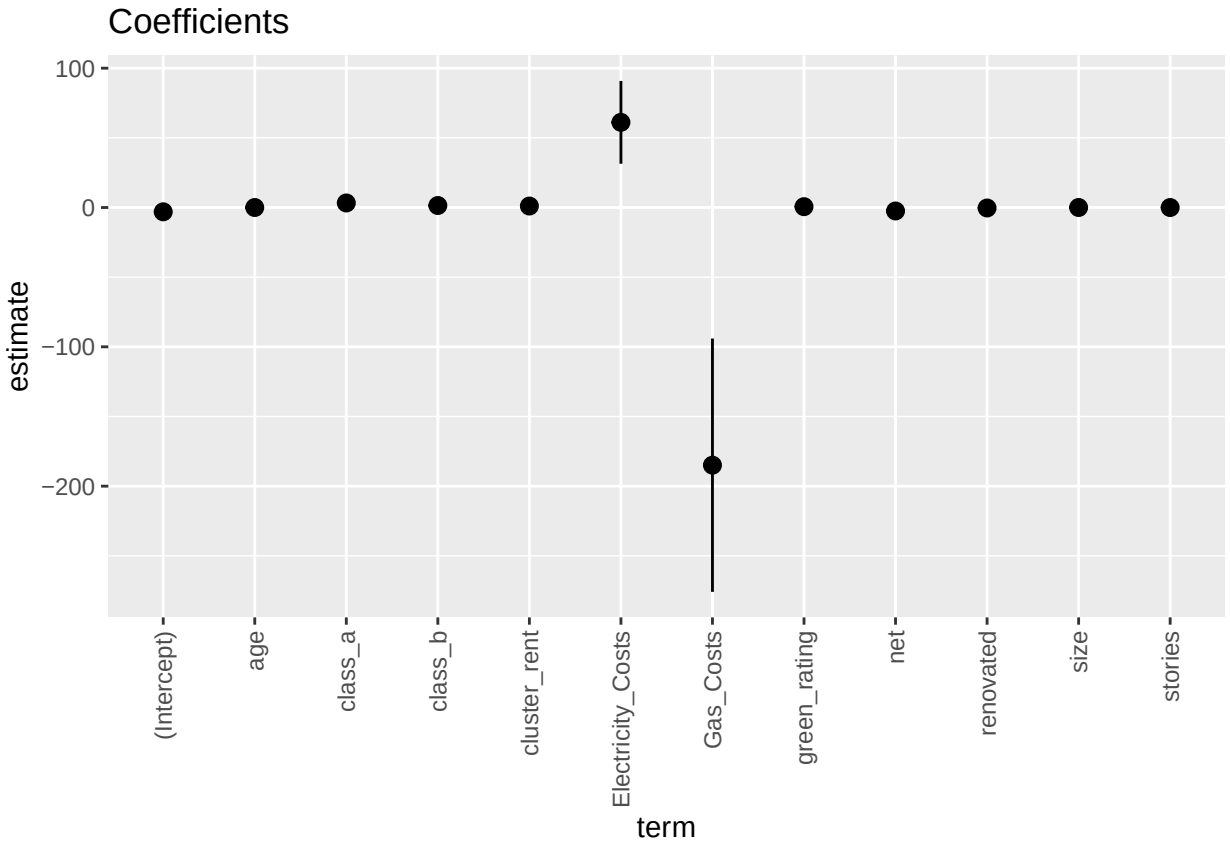
Green Buildings

There are some parts I don't agree with the guru. First of all, the guru is assuming green buildings have a higher rate by looking at green buildings and non-green buildings separately. Looking at the rent frequency histogram, there's no significant distinction between the green and non-green buildings.

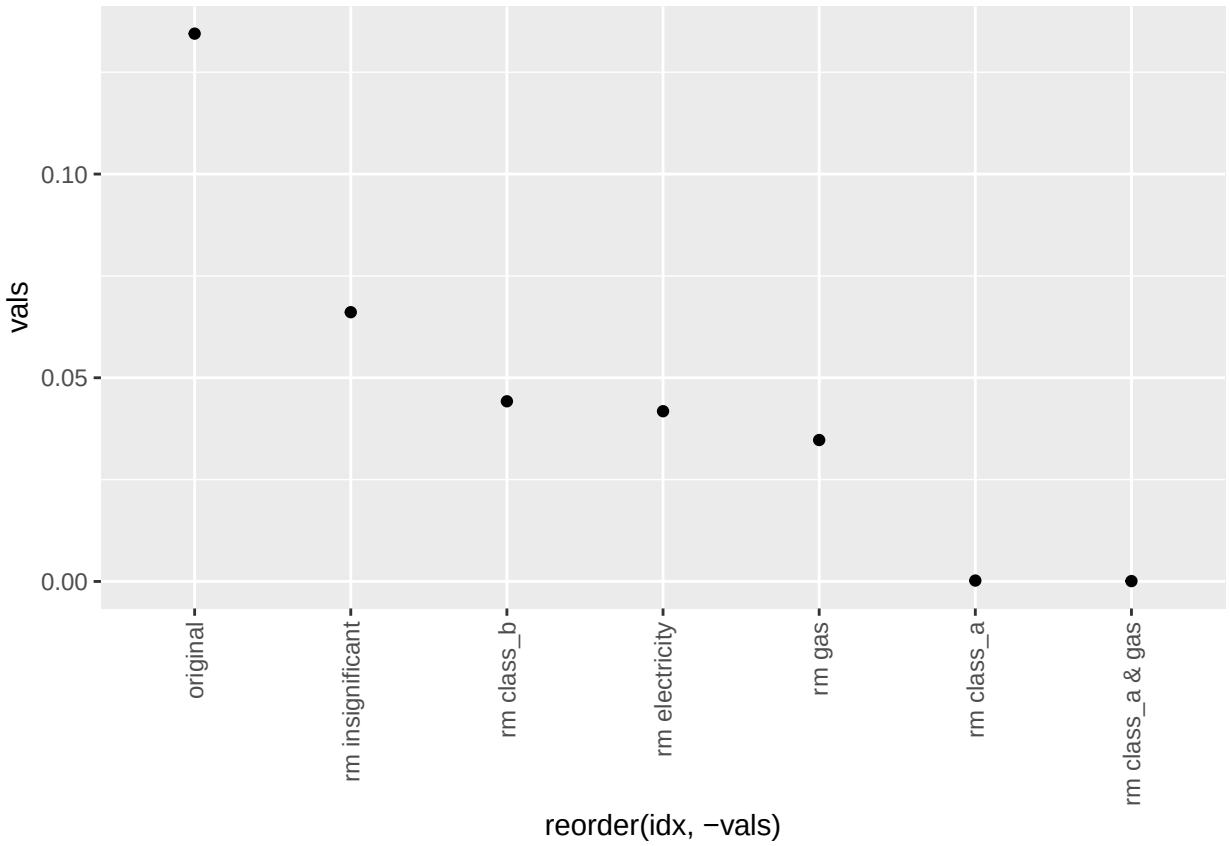
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



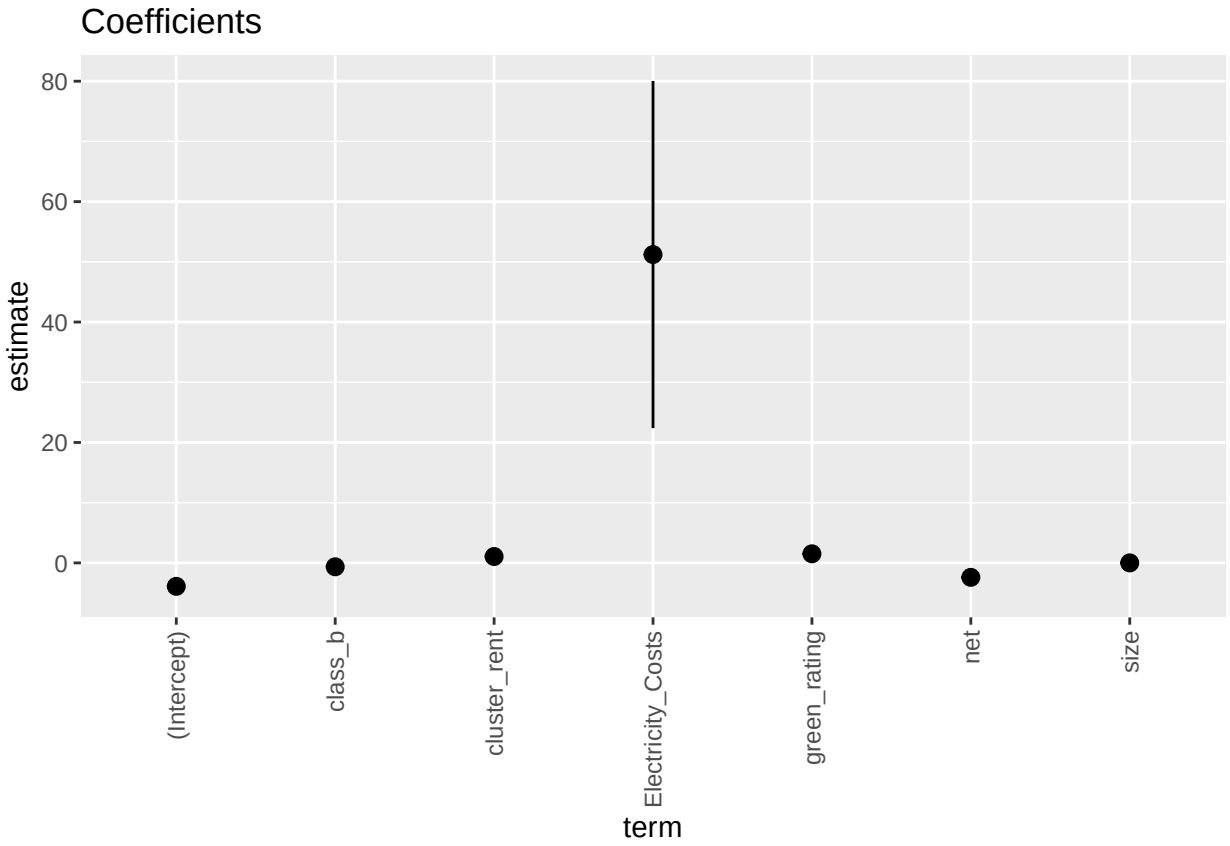
In addition, a linear regression on whole dataset shows that there's no significant correlation between rent and green_rating, as green_rating has a p value of 0.13. However, there's a chance that some of the variables are confounders for the relationship between rent and green, given the disproportionate coefficient plot.



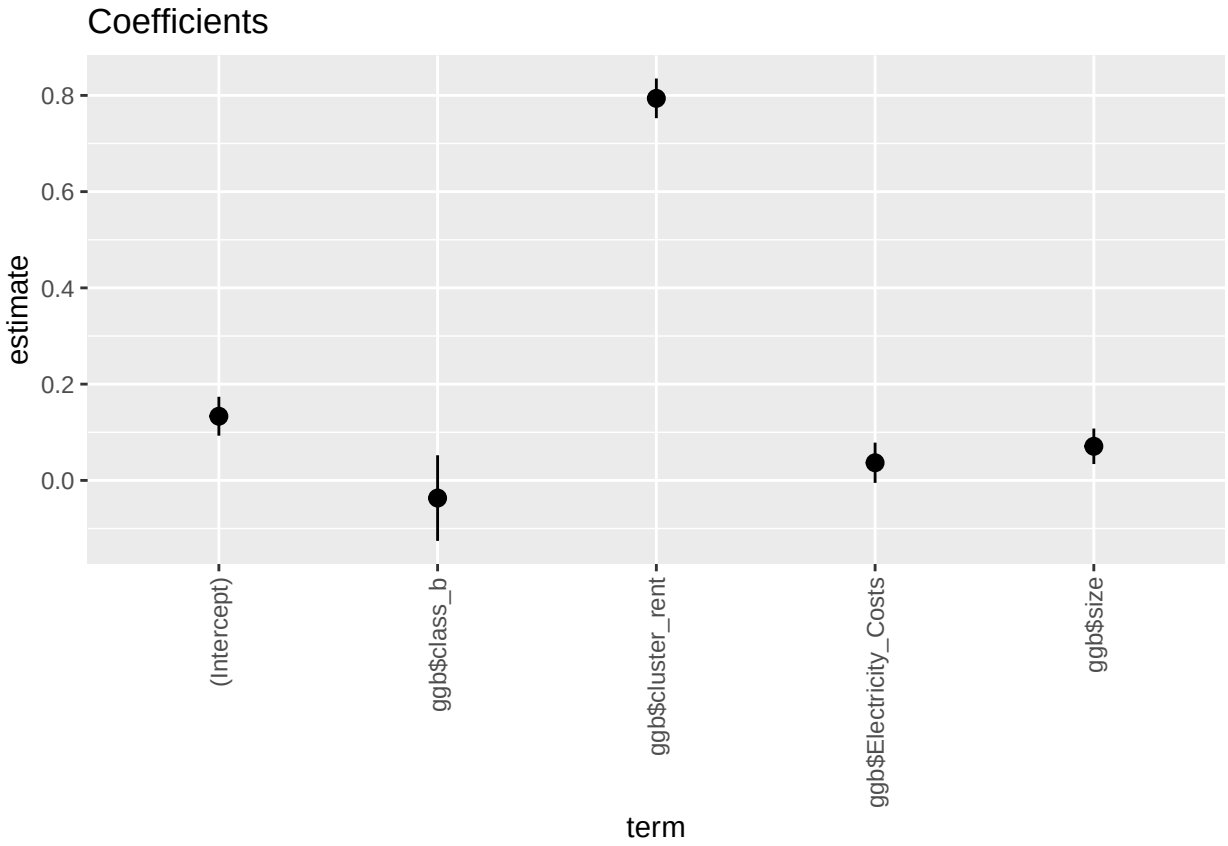
As we remove the insignificant variables and gas, the p value of green drops from 0.134 to 0.035. So green_rating might be correlated with gas as green buildings should have lower recurring costs by design. But the most significant confounder is class_a as the p value of green drops from 0.134 to 0.000219.



Taking out both will make the p value of green 5.32e-05 and result in a seemingly more reasonable coefficient plot.

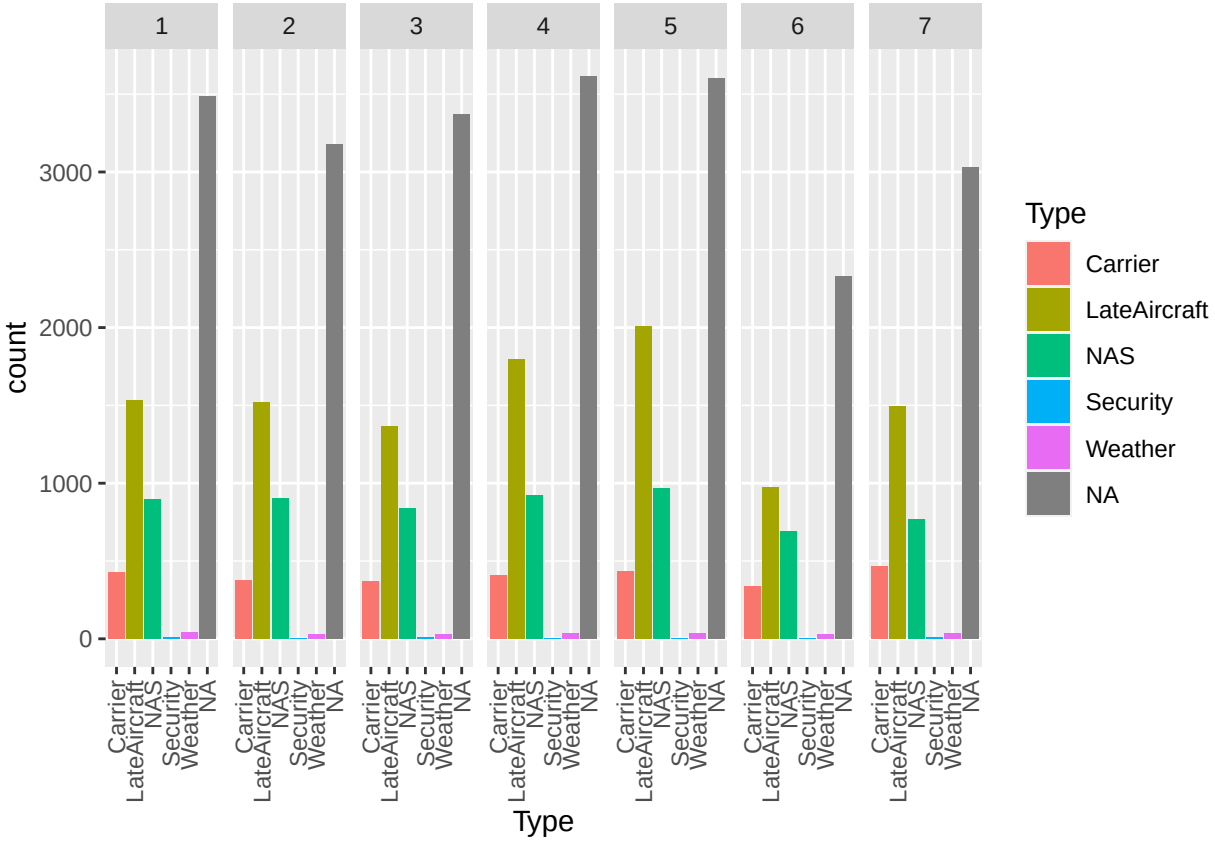


From the models, we can see that only size and cluster_rent are significantly correlated with rent and cluster_rent has a very positive relationship with rent.

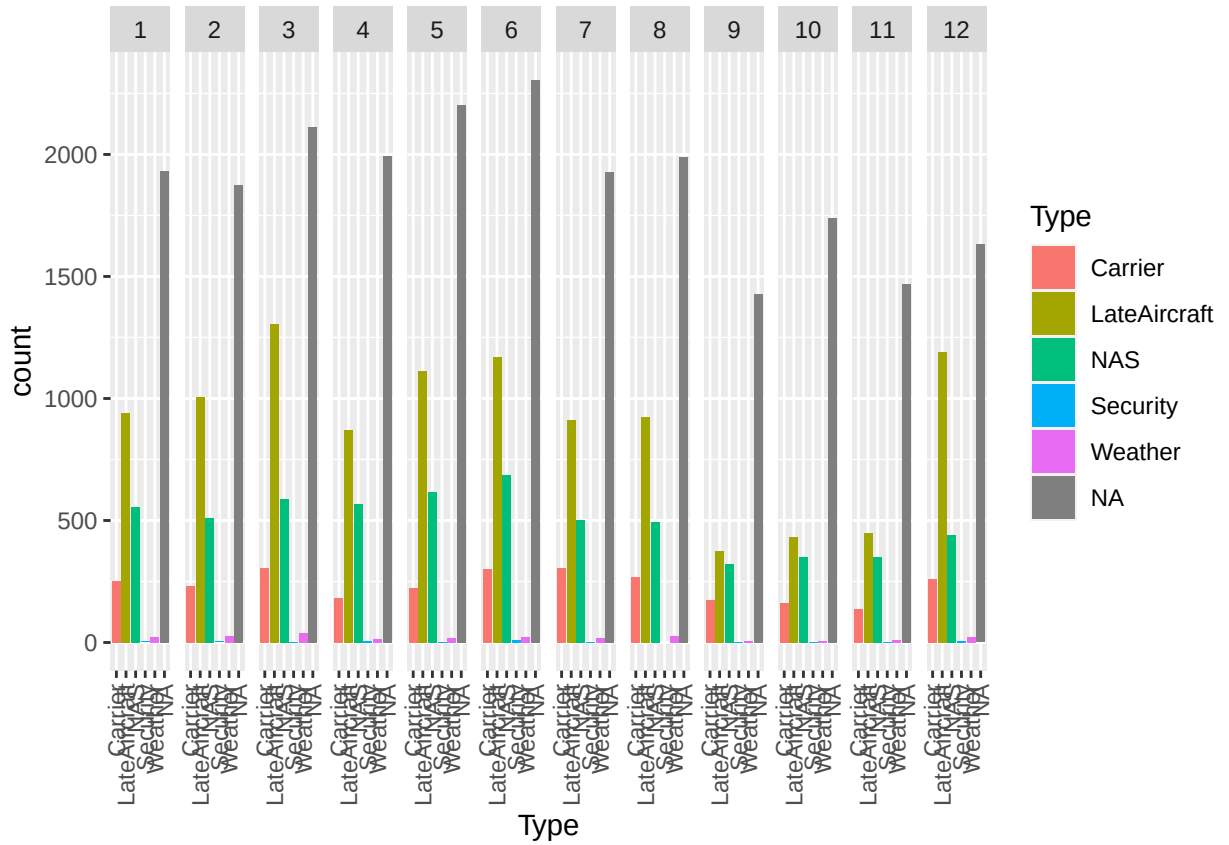


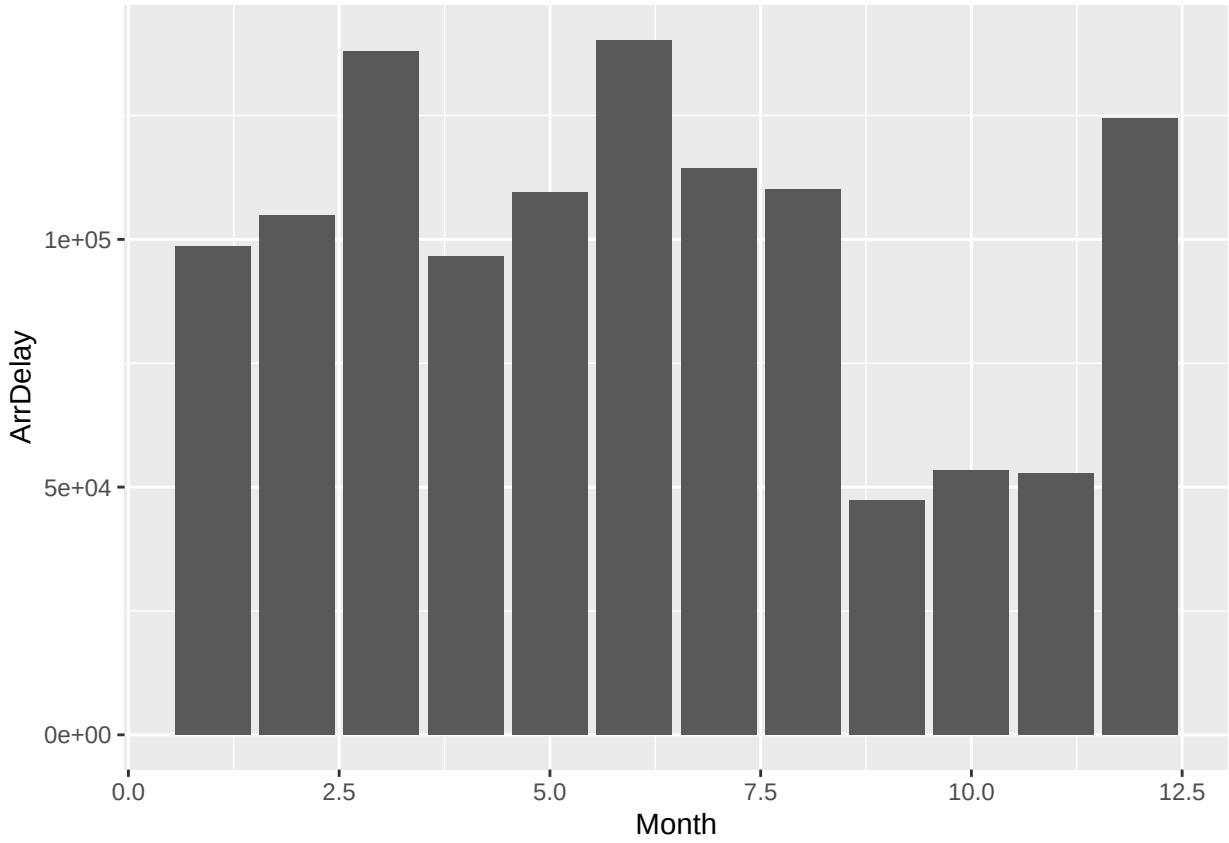
Since the new building would be in Austin and the database is nationwide, the best way to improve the prediction is to use the nearest clusters to predict rent.

ABIA



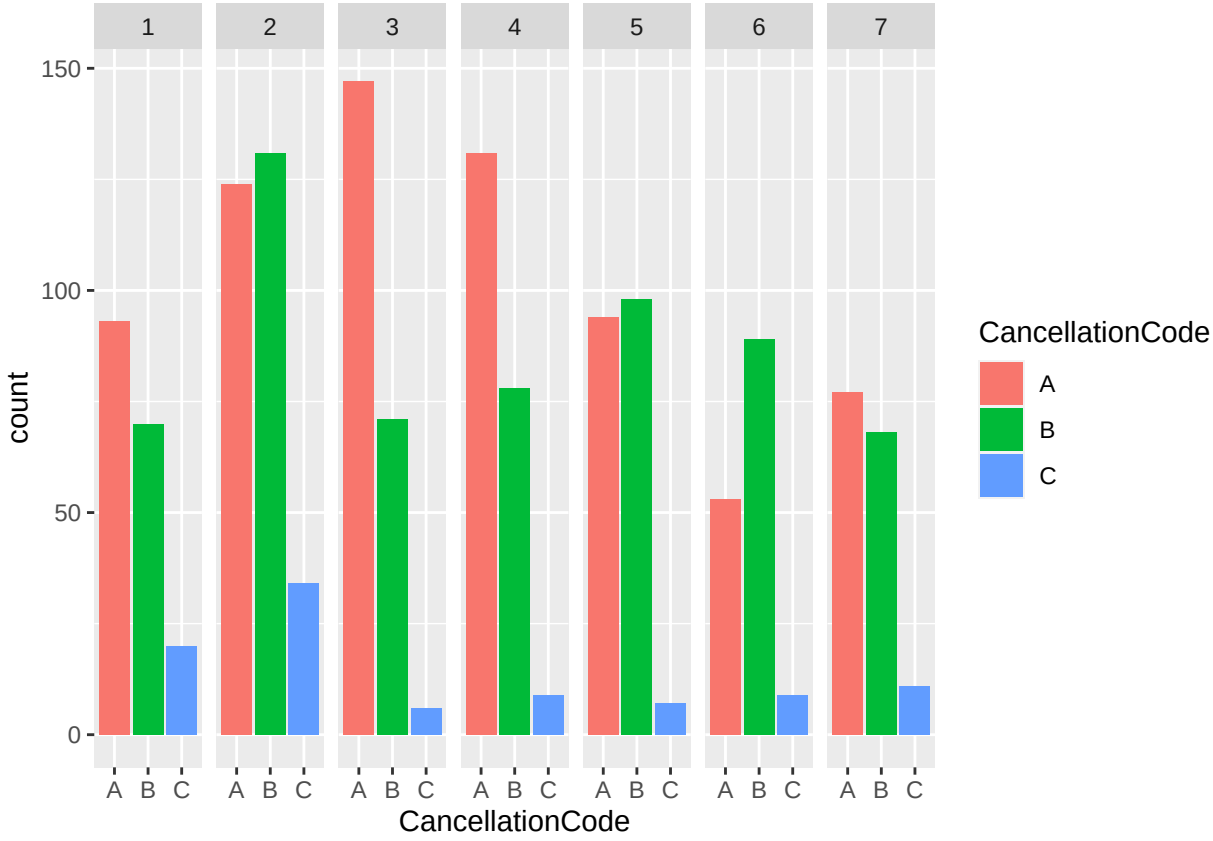
The flights into and out of Austin had some interesting delay patterns. When we look at the number of flights that arrived late during each day of the week, we can see that Late Aircraft is the most frequent kind of delays and security was the least frequent. There's a big drop in Late Aircraft delays from Fridays to Saturdays, which could be because there are fewer flights during the weekend and less traffic in the air and on the ground. So the airplanes need not wait for their turns to use the runway and take off. The proportion of the other kinds of delays are stable throughout the week, which makes sense since they're either related to weather or rare events.

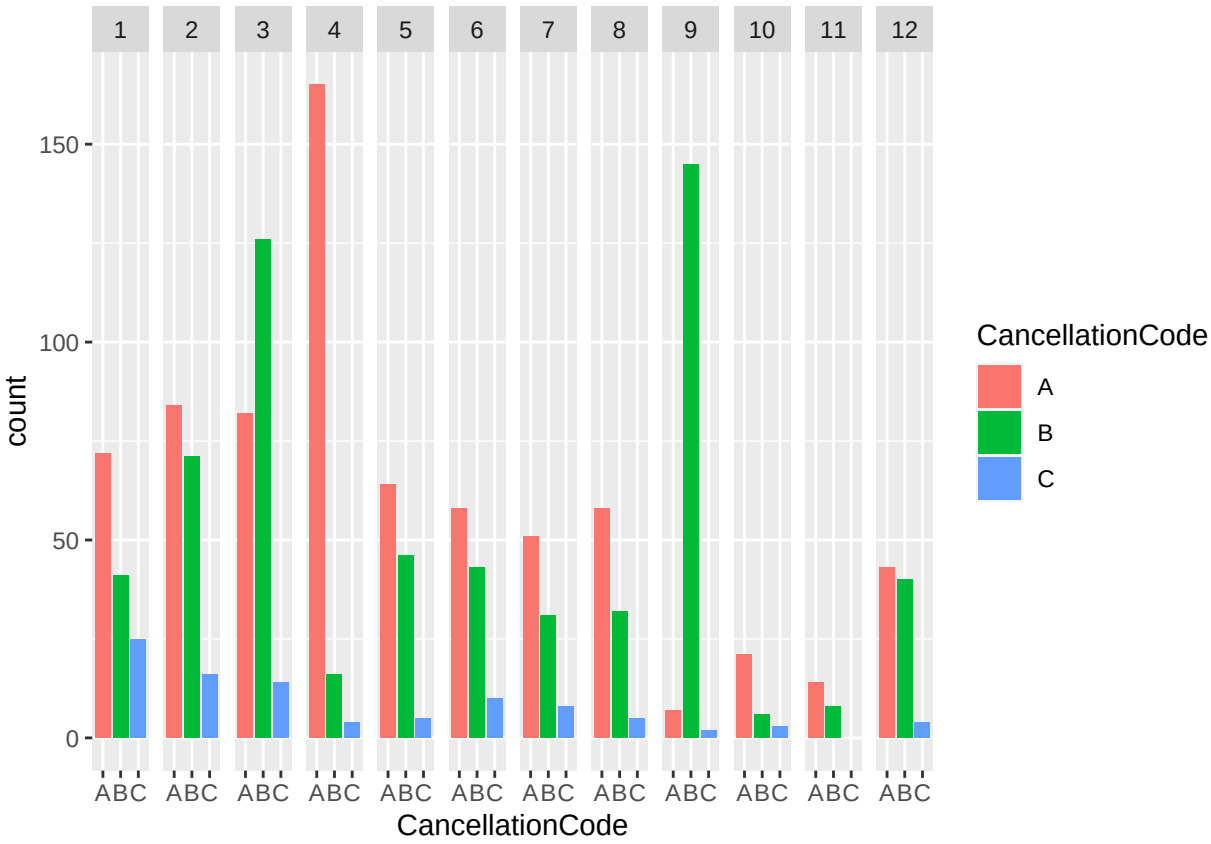




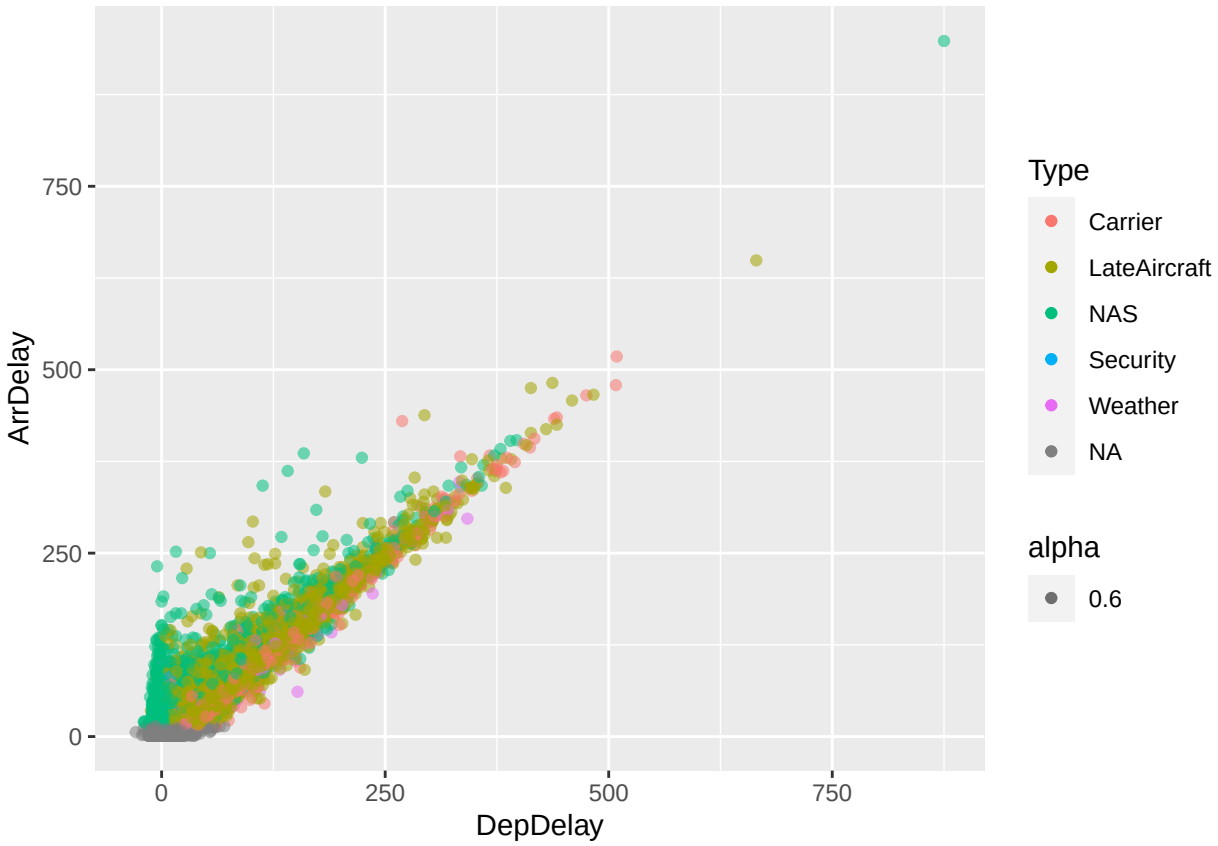
Looking at delays throughout the year, we can see that the number of delays are significantly low

from September to November and has abnormal highs in March and June. The low in Fall could be that the weather in Austin are more stable or better for flying in this season. Or people may travel to Austin less in this quarter and go home for holiday reasons. The high in March is apparently due to the SXSW Festival and the one in June can be explained by the increasing traffic in the air as a result of the urge to travel in the beginning of summer holidays. Most types of delays swing with this trend, which could mean that they are internally related in some way.

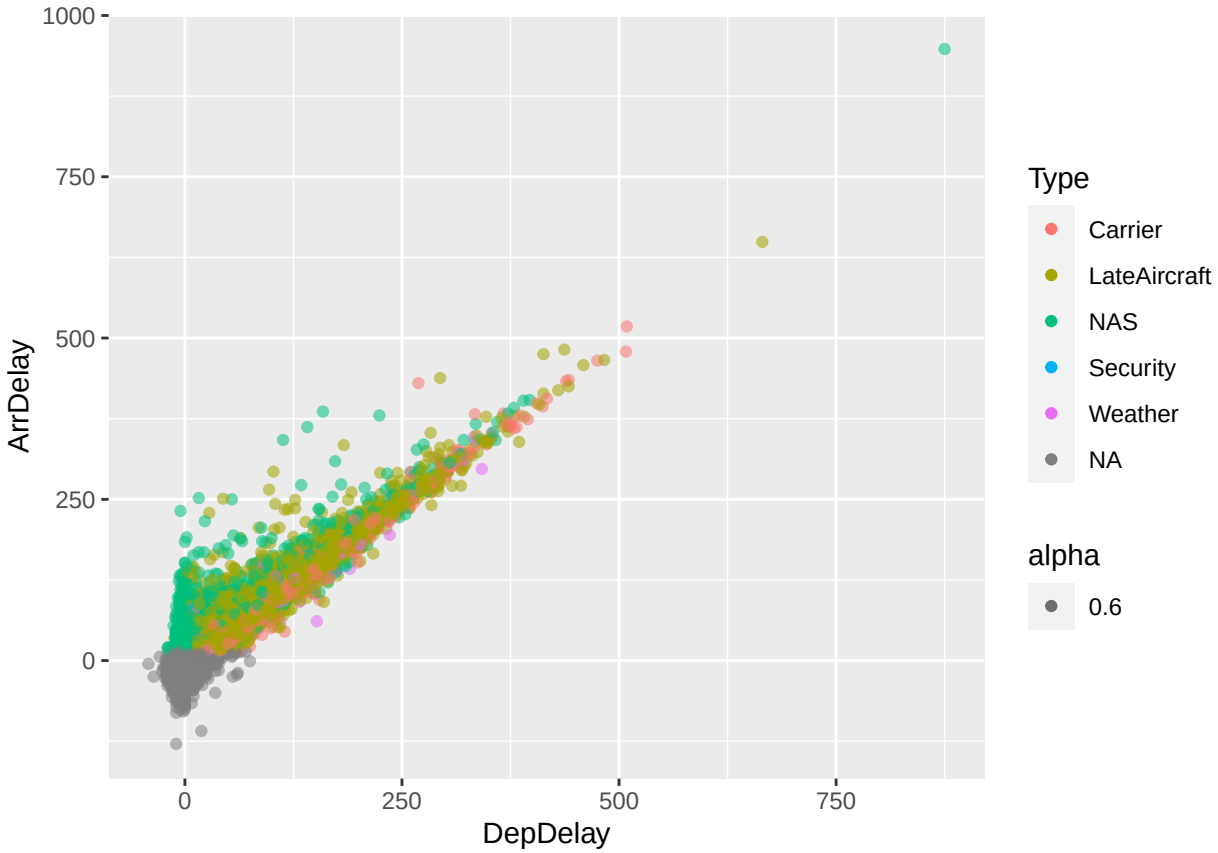




Looking at cancellation patterns, it is surprising that carrier cancellation seems to happen particularly often in April, which has a moderate temperature. Maybe vehical issues and device issues are not the leading factors here. On the contrary, weather cancellation is high in September as we'd expect. It's not uncommon to have some storms in that month. However, weather cancellation is also frequent in March. We will probably need more geographic knowledge to explain that.

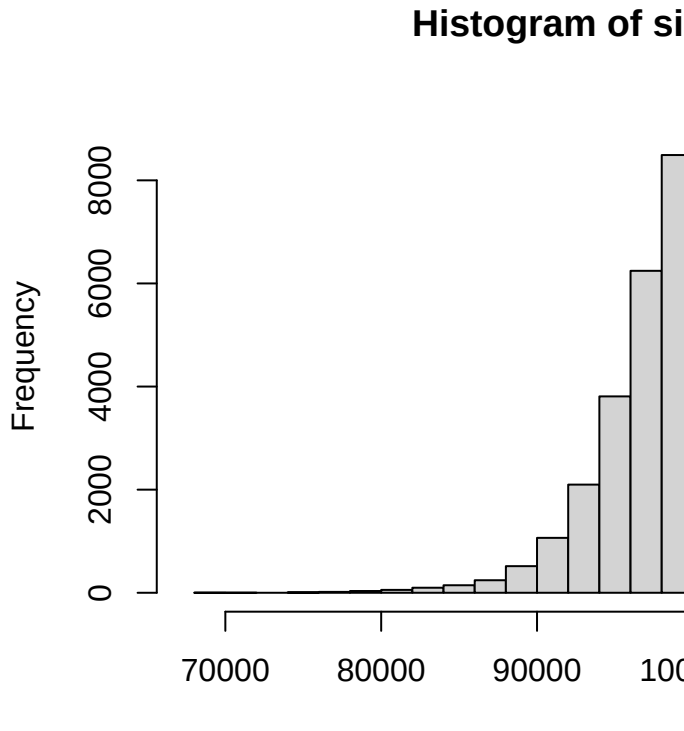



```
## Warning: Removed 1601 rows containing missing values (geom_point).
```

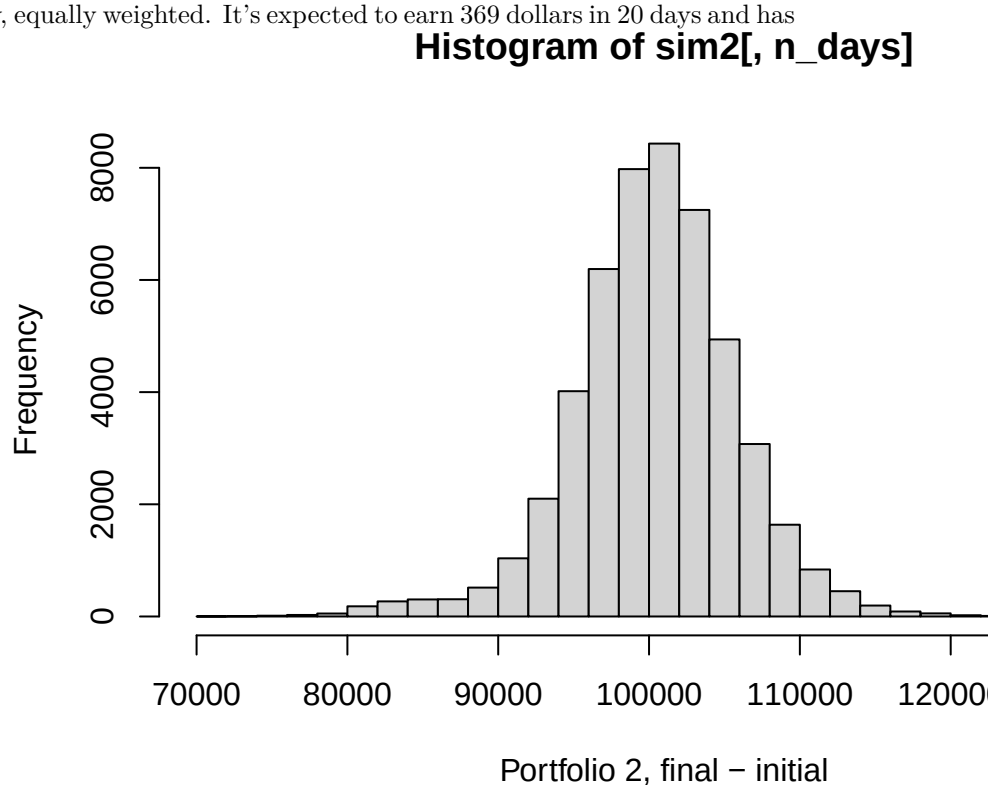


Lastly, I tried to explore the relationship among departure delay, arrival delay, and the type of delay. The first graph looks at only the late arrivals. It's not surprising that a late arrival tend to a result of a late departure. However, some flights that departed early might also arrive late because of "NAS" reasons, where a type of weather delay "could be reduced with corrective action by the airports or the Federal Aviation Administration"(bts.gov). So even though many flights left on time or early, their arrival delayed for hours. The second graph looks at all departure and arrival times. We can see a dense region where many flights, though departed only on time, or sometimes even late, managed to arrive early.

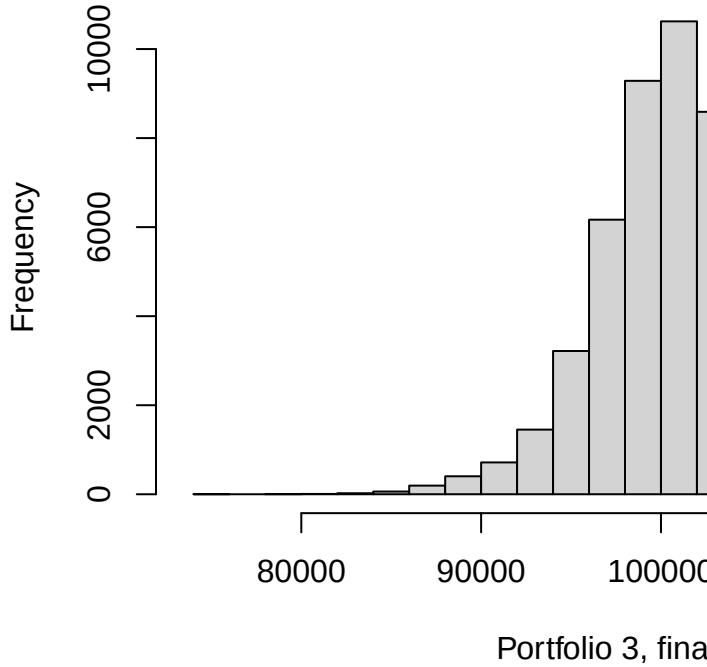
Portfolio modeling



I picked 11 EFTs from 3 categories: high-yield, energy, and all-cap.
Porfolio 1 has 3 high-yields and 1 energy, equally weighted. It's expected to earn 369 dollars in 20 days and has

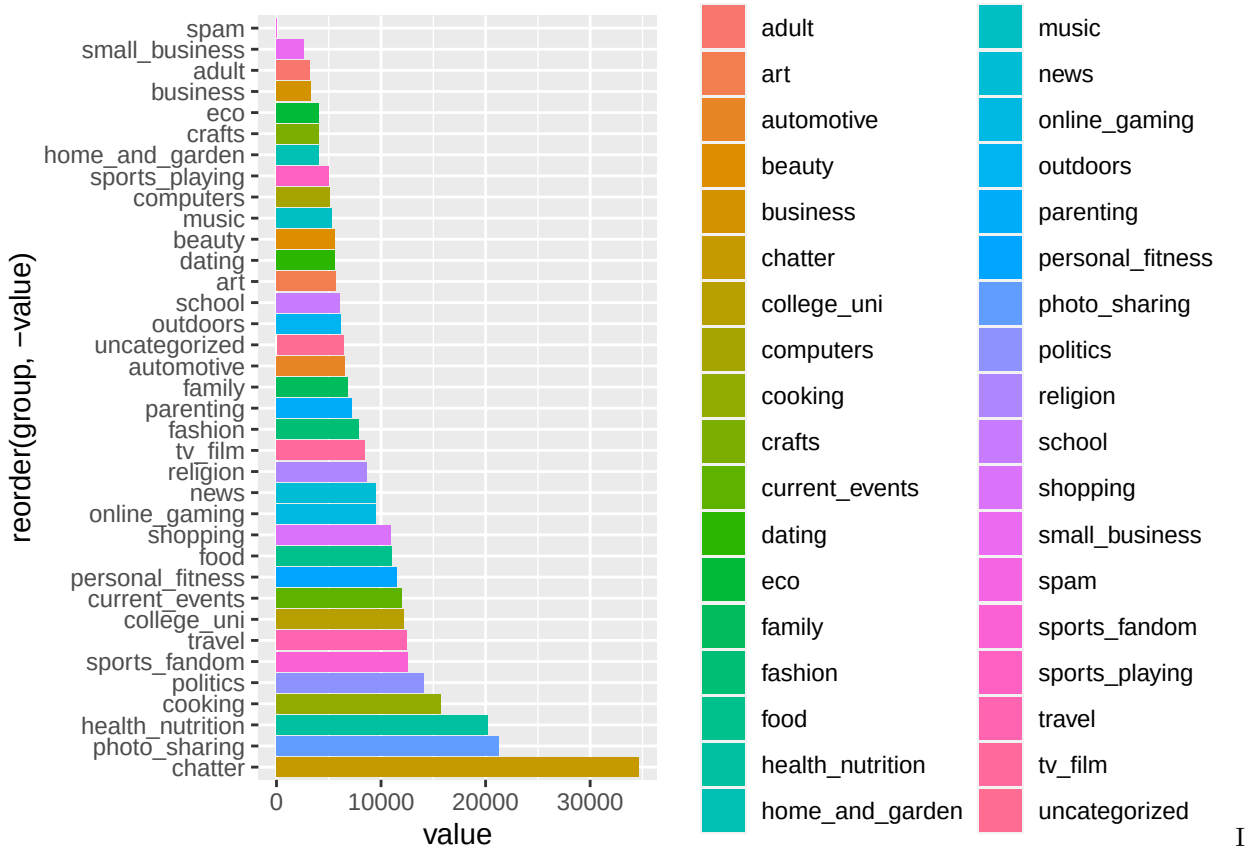


a tail risk of 7641 dollars at the 5% level.
Portfolio 2 has 1 high-yield, 1 energy, and 2 all-caps, equally weighted. It's estimated to earn 336 dollars on av-

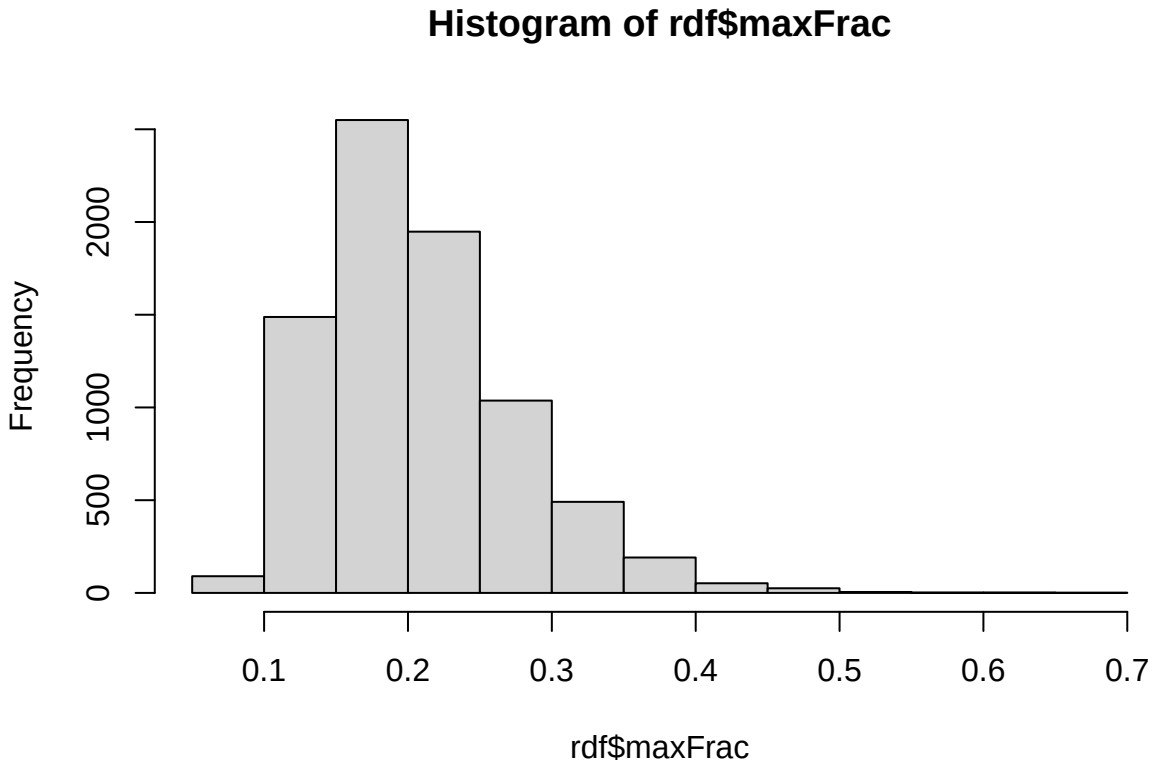


erage in 20 days and has a tail risk of 8447 dollars at the 5% level.
Portfolio 3 has 1 all-cap (40%), 1 high-yield(30%), and 1 energy(30%). It's expected to earn 492 dollars on average in 20 days and has a tail risk of 6383 dollars at the 5% level. Overall, Portfolio 3 has a higher earning on average and a lower tail risk. Portfolio 2 has the lowest expected earning and highest tail risk.

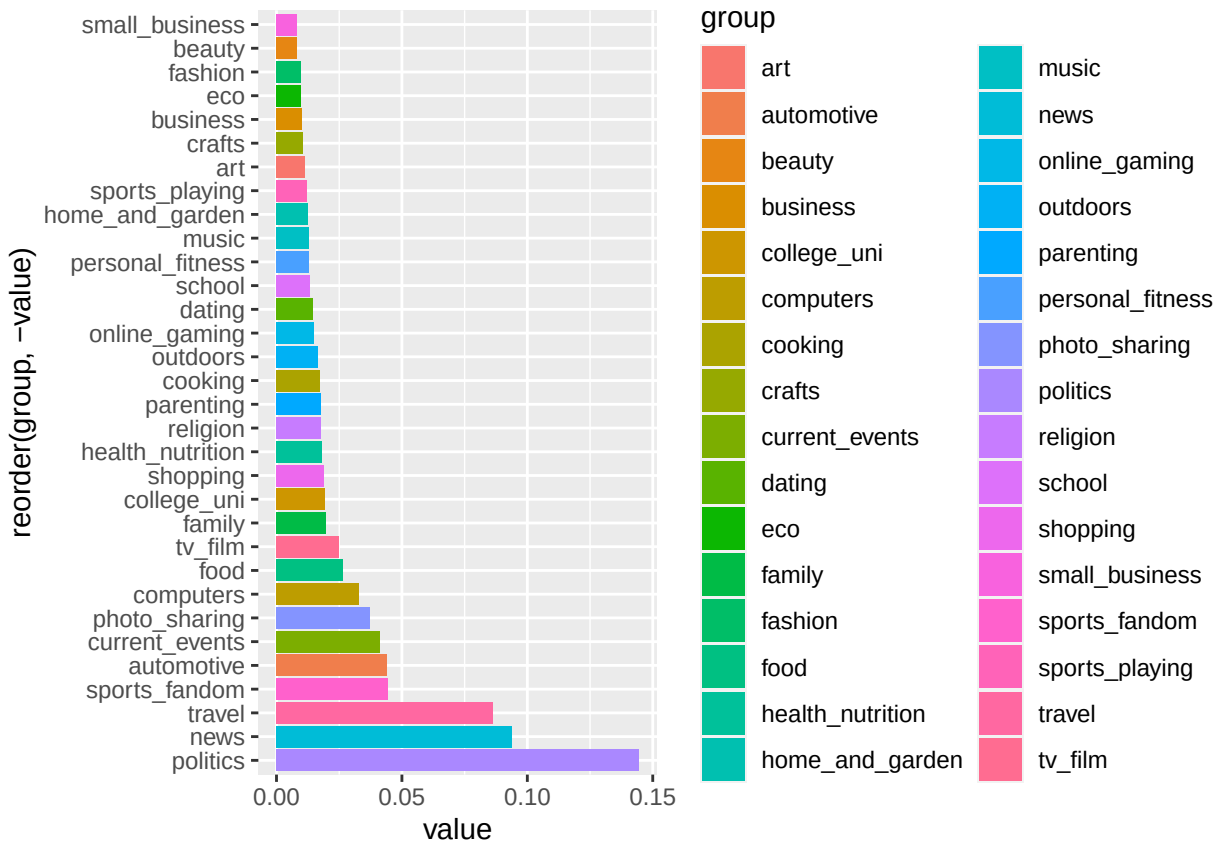
Market Segmentation



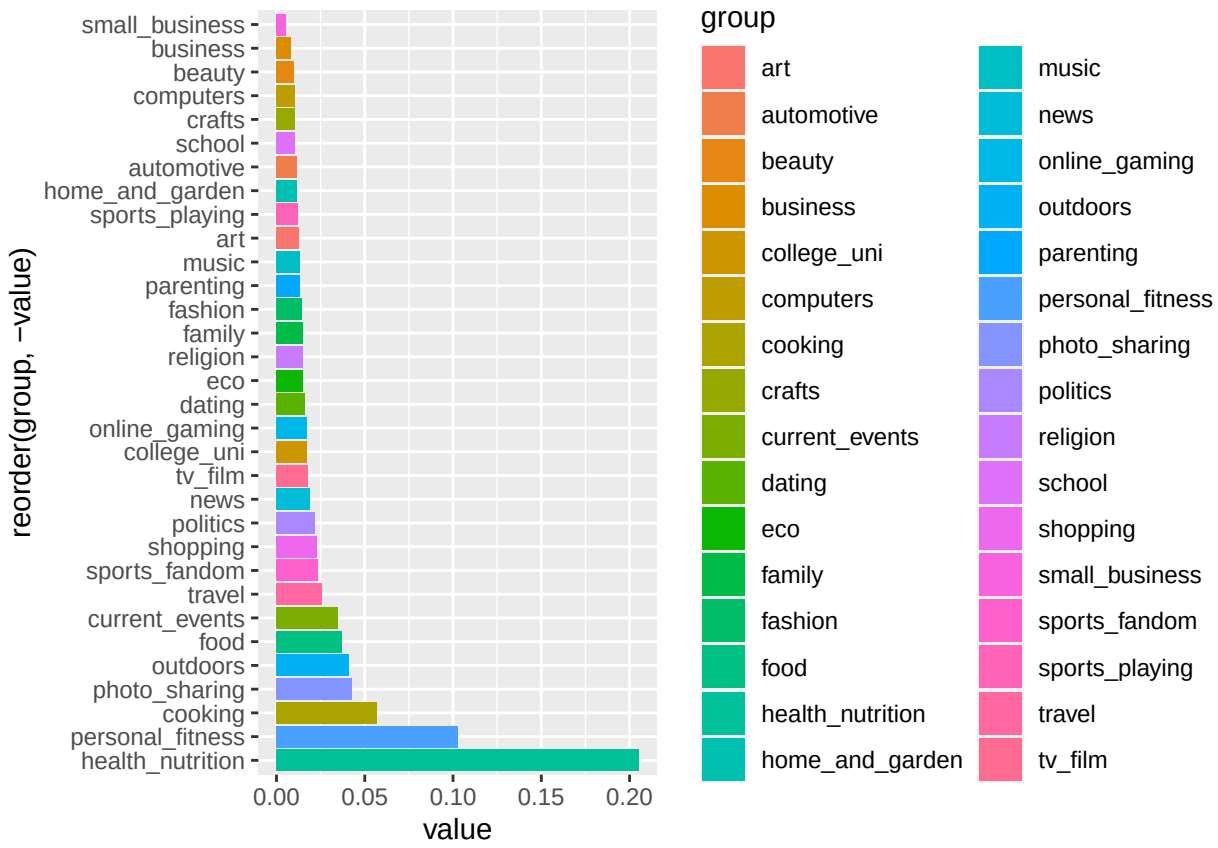
I first summed the number of times each category label appears in a tweet and sorted it. This is not very helpful. I decided to use fractions instead of counts since some users may tweet a lot while others don't. So the fraction shows more about the user.



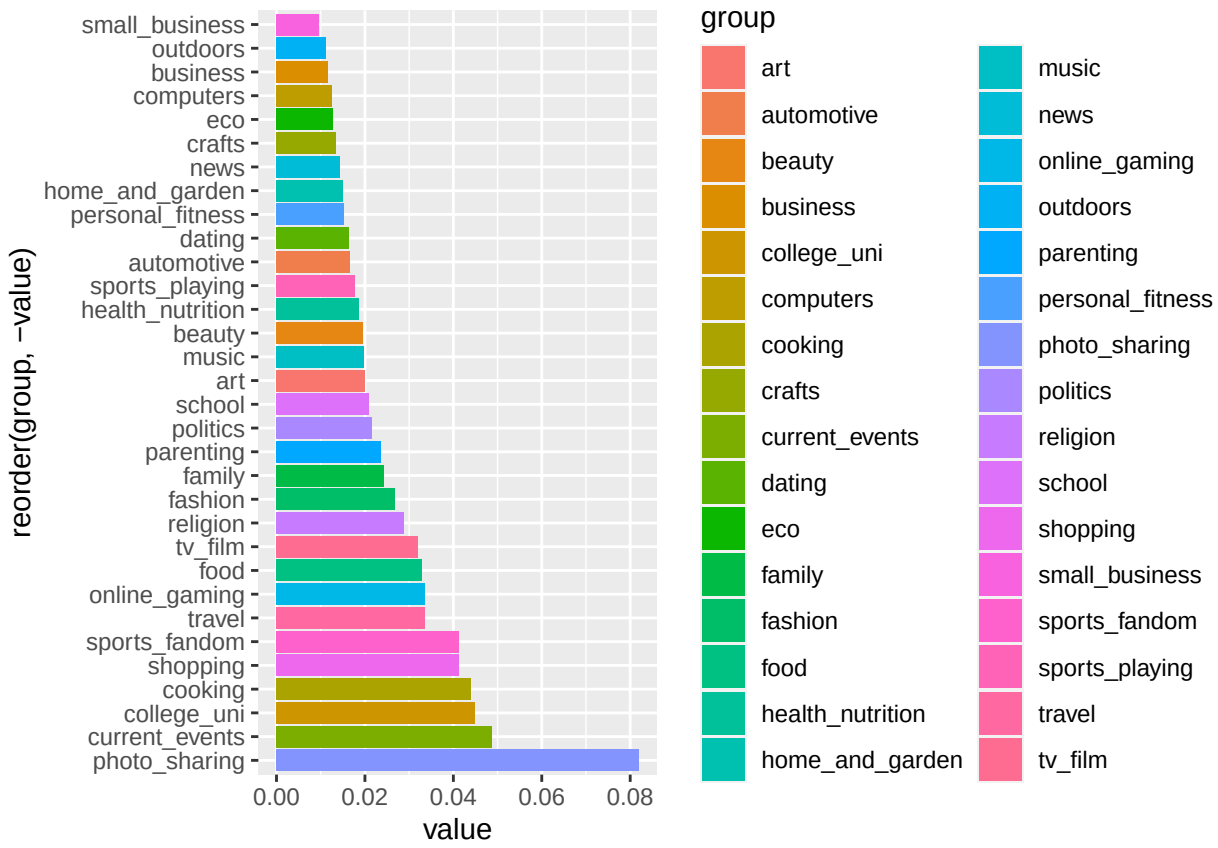
Given the histogram of the fractions we can decide that 40% is large enough to decide if the user is a bot so I excluded them. It seems to me that clustering is a better choice here since we don't have a y variable to predict. I decided to do 3 clusters based on Gap's result. The Gap code is commented out as it takes too long to knit.



Top categories in cluster 1 include photo_sharing, current_events, college, cooking, shopping, sports, etc. To me, it looks like they are young adults, specifically college students or rising college students.



Top categories in cluster 2 include politics, news, travel, sports_fandom, automotive, etc. To me, it looks like they are styreotypical men who care about politics, what's happening in the world, and sports and cars.



Top categories in cluster 3 include health, personal_fitness, cooking, photo_sharing, outdoors, food, etc. To me, it looks like they care about physical health and physique. They'd probably buy organic food or go to wholefoods.

Author Attribution

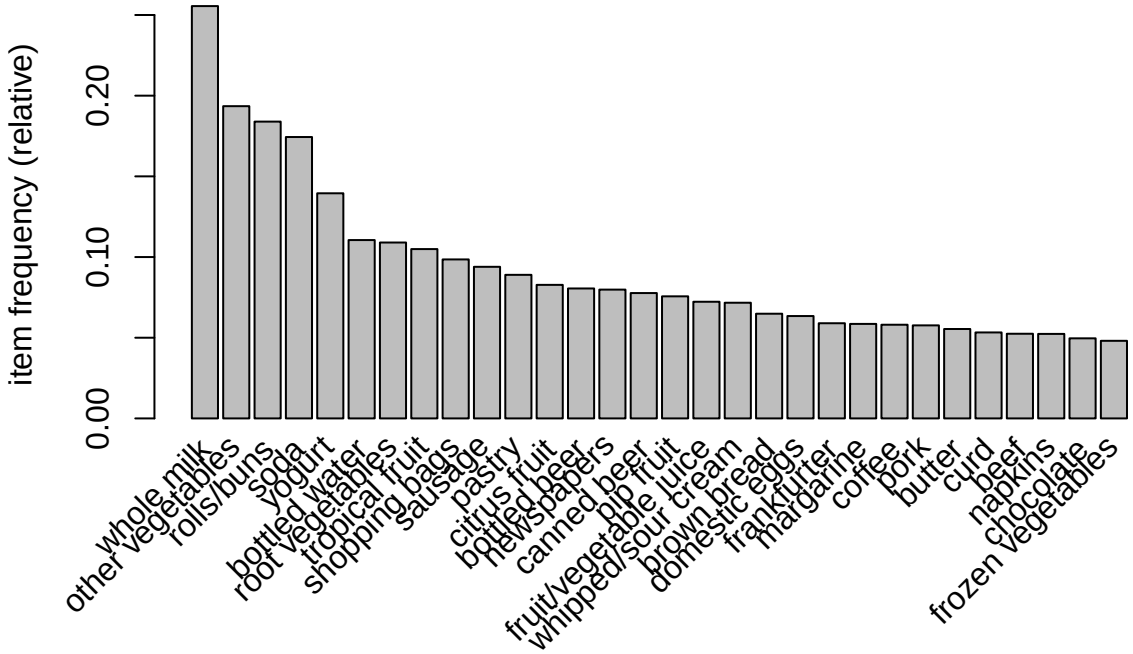
First read in all folders in C50train. Each folder represents an author so for each author, get the author name from folder name, and read in all documents in that folder and add them to the training set for x. Meanwhile we add 50 author names to the training set for y to match the size of trainx.

Then create the corpus witthh the training set and clean the file name, do the pre-processing and tokenization, build DTM, and remove sparse terms, as shown in lecture. After we make the weighted DTM into a matrix, we do the same for the test folder.

We build the DTM for test set with the dictionary of the DTM of train set.

Then we use PCA to reduce dimension. I did 30 summaries because bigger ranks broke my laptop, and that suggests using PC1 to PC10 are enough. We ignore the words that are in test set but not in train set by only fitting the model with intersection of the train and test sets. Finally, we use knn to see our accuracy. (The pca code is commented out as it takes too long to knit).

Association Rule



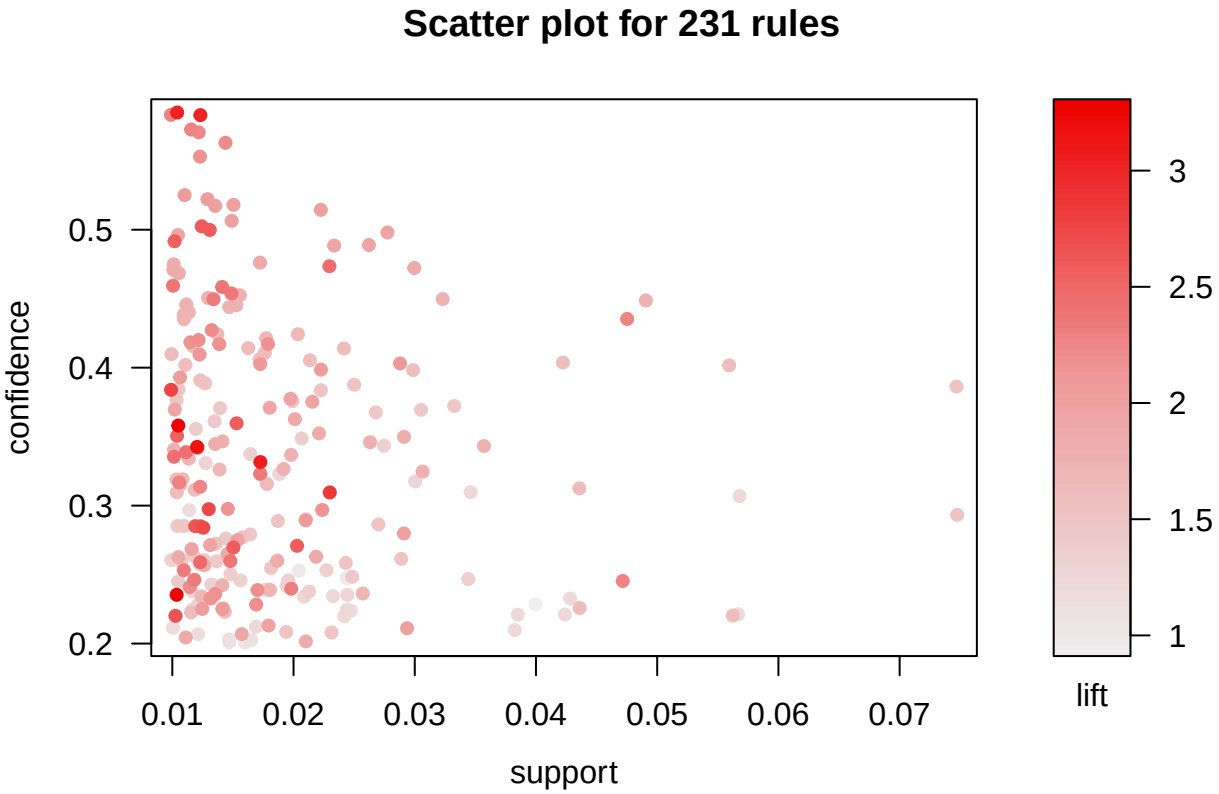

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.2    0.1    1 none FALSE            TRUE       5    0.05      2
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 491
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
## sorting and recoding items ... [28 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [6 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
##     lhs                   rhs                support    confidence coverage
## [1] {yogurt}           => {whole milk}       0.05602440 0.4016035  0.1395018
## [2] {whole milk}       => {yogurt}           0.05602440 0.2192598  0.2555160
## [3] {rolls/buns}       => {whole milk}       0.05663447 0.3079049  0.1839349
## [4] {whole milk}       => {rolls/buns}       0.05663447 0.2216474  0.2555160
## [5] {other vegetables} => {whole milk}       0.07483477 0.3867578  0.1934926
## [6] {whole milk}       => {other vegetables} 0.07483477 0.2928770  0.2555160
##     lift     count
```

```
## [1] 1.571735 551
## [2] 1.571735 551
## [3] 1.205032 557
## [4] 1.205032 557
## [5] 1.513634 736
## [6] 1.513634 736
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.2    0.1    1 none FALSE            TRUE       5    0.01      2
##  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: 98
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
## sorting and recoding items ... [88 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 done [0.01s].
## writing ... [231 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



Given the frequency plot we can see that the item frequency smooths down to around 0.05, so I decided to set support = 0.05. For simplicity, I set length to 2 so we only look at 2 items. And a arbitrary confidence=0.2. The generated rules are mostly dairies including milk and yogurt, and buns that go with milk. It is pretty likely that who buys milk will buy yogurt and vegetables and vice versa. This makes sense since they are commodities that constantly need restock bc go bad easily.

To see more rules, I lowered support to 0.01 which would give 125 rules. That is too many so I raised it to 0.02 and got 72 rules. Now we can see proteins, drinks, fruit and other commodities that people buy frequently.

There's only 1 rule with support = 0.02 and confidence = 0.5 so I set confidence to 0.4 and got 15 rules.

```
rule3 = apriori(grotrans,parameter=list(support=0.02, confidence=.4, minlen=2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.4    0.1    1 none FALSE            TRUE       5    0.02      2
##  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: 196
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
```

```
## sorting and recoding items ... [59 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [15 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
inspect(rule3)
```

```
##      lhs                                   rhs                 support
## [1]  {frozen vegetables}                => {whole milk}       0.02043721
## [2]  {beef}                             => {whole milk}       0.02125064
## [3]  {curd}                             => {whole milk}       0.02613116
## [4]  {margarine}                        => {whole milk}       0.02419929
## [5]  {butter}                           => {whole milk}       0.02755465
## [6]  {domestic eggs}                    => {whole milk}       0.02999492
## [7]  {whipped/sour cream}               => {other vegetables} 0.02887646
## [8]  {whipped/sour cream}               => {whole milk}       0.03223183
## [9]  {tropical fruit}                   => {whole milk}       0.04229792
## [10] {root vegetables}                  => {other vegetables} 0.04738180
## [11] {root vegetables}                  => {whole milk}       0.04890696
## [12] {yogurt}                           => {whole milk}       0.05602440
## [13] {other vegetables,root vegetables} => {whole milk}       0.02318251
## [14] {root vegetables,whole milk}       => {other vegetables} 0.02318251
## [15] {other vegetables,yogurt}          => {whole milk}       0.02226741
##      confidence coverage   lift     count
## [1]  0.4249471  0.04809354 1.663094 201
## [2]  0.4050388  0.05246568 1.585180 209
## [3]  0.4904580  0.05327911 1.919481 257
## [4]  0.4131944  0.05856634 1.617098 238
## [5]  0.4972477  0.05541434 1.946053 271
## [6]  0.4727564  0.06344687 1.850203 295
## [7]  0.4028369  0.07168277 2.081924 284
## [8]  0.4496454  0.07168277 1.759754 317
## [9]  0.4031008  0.10493137 1.577595 416
## [10] 0.4347015  0.10899847 2.246605 466
## [11] 0.4486940  0.10899847 1.756031 481
## [12] 0.4016035  0.13950178 1.571735 551
## [13] 0.4892704  0.04738180 1.914833 228
## [14] 0.4740125  0.04890696 2.449770 228
## [15] 0.5128806  0.04341637 2.007235 219
```

This case is like a combination of rule 1 and 2 and shows relationships between whole milk and other perishables. The takeaway might be that people buy whole milk the most often.