Exercises

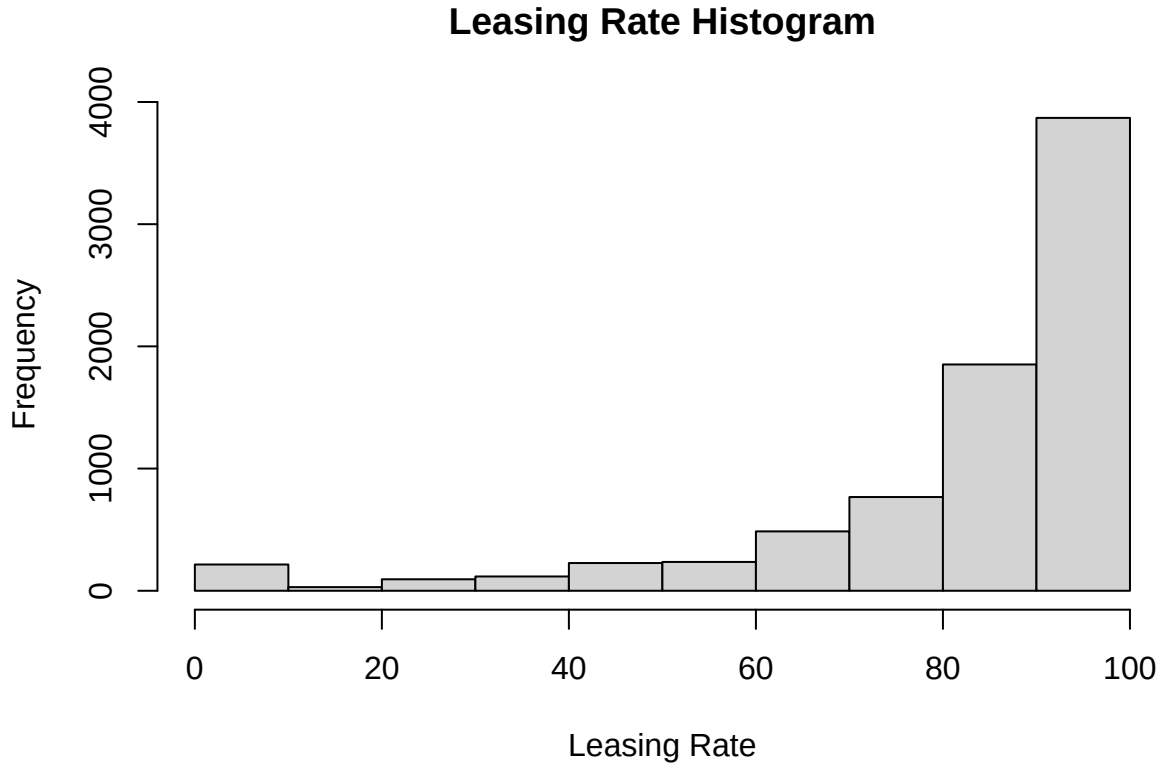
Jacob Pammer, Chandler Wann, Narain Mandyam, Rudraksh Garg

8/16/2021

Git Repo: https://github.com/rudragarg/STA_380_Group_Exercises

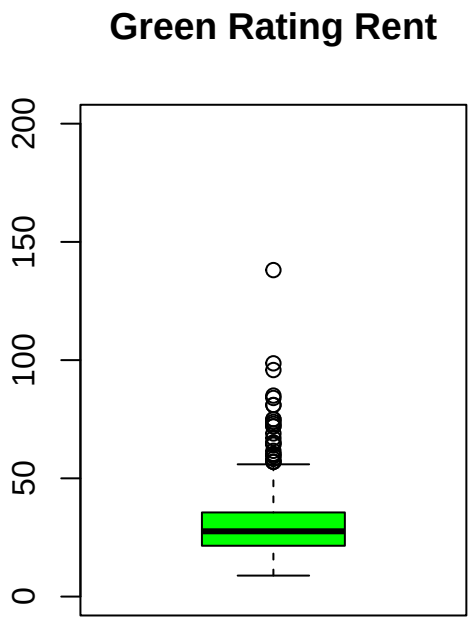
Visual story telling part 1: green buildings

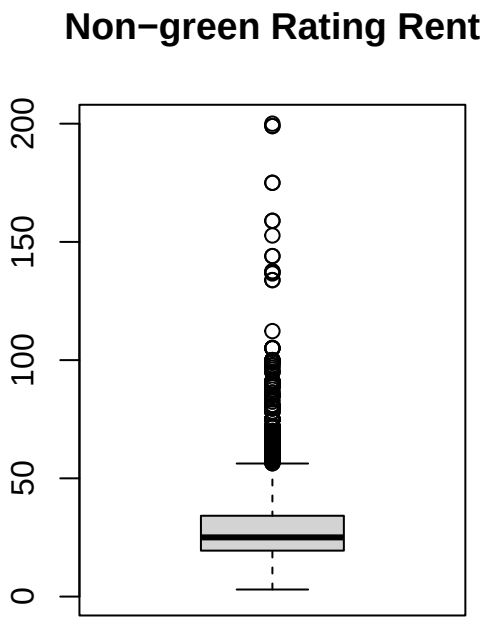
To start, removing records with less than a 10% leasing rate makes sense. The histogram below, shows that these records seem to be special cases that have values far less than the average. Additionally, her theory that these low vacancy places have something wrong with them seems legitimate. Since, this will be a new building, we should assume that it will not have any of these problems.



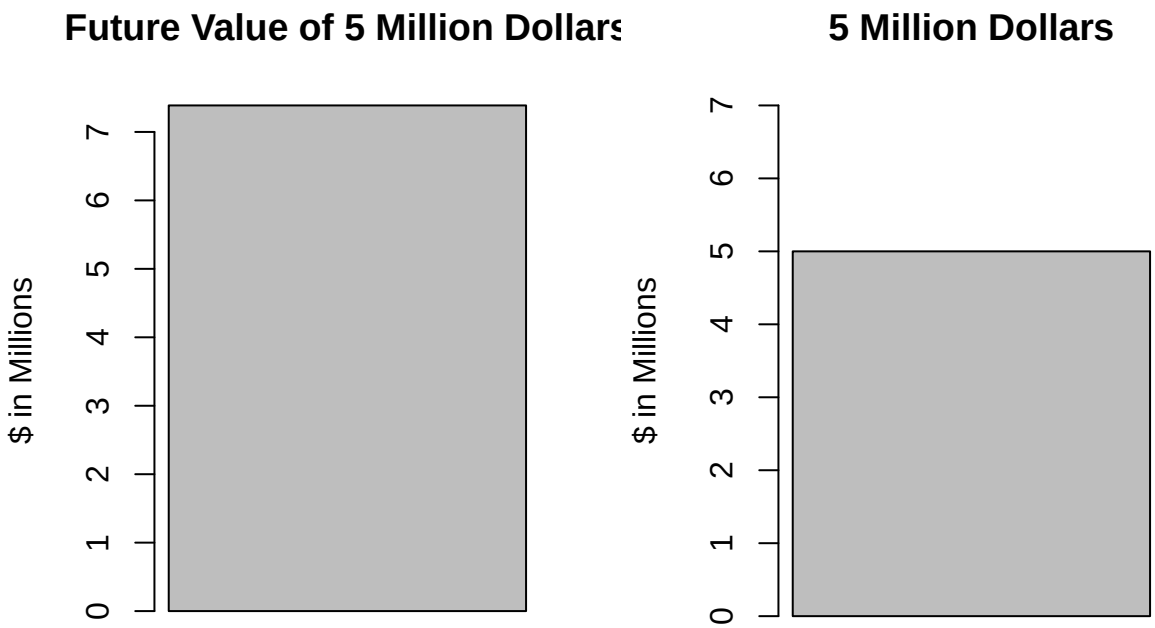
Certain portions of the Guru's analysis were done correctly. For example, the use of medians is a good idea to avoid outliers. However, the analyst failed to account for potential variation in rent prices. As you can see in the side by side plots below, green rating's IQR and non-green rating's IQR cross through the same variable. So a range of estimates would be necessary to estimate a change in profits created by building a

green complex.

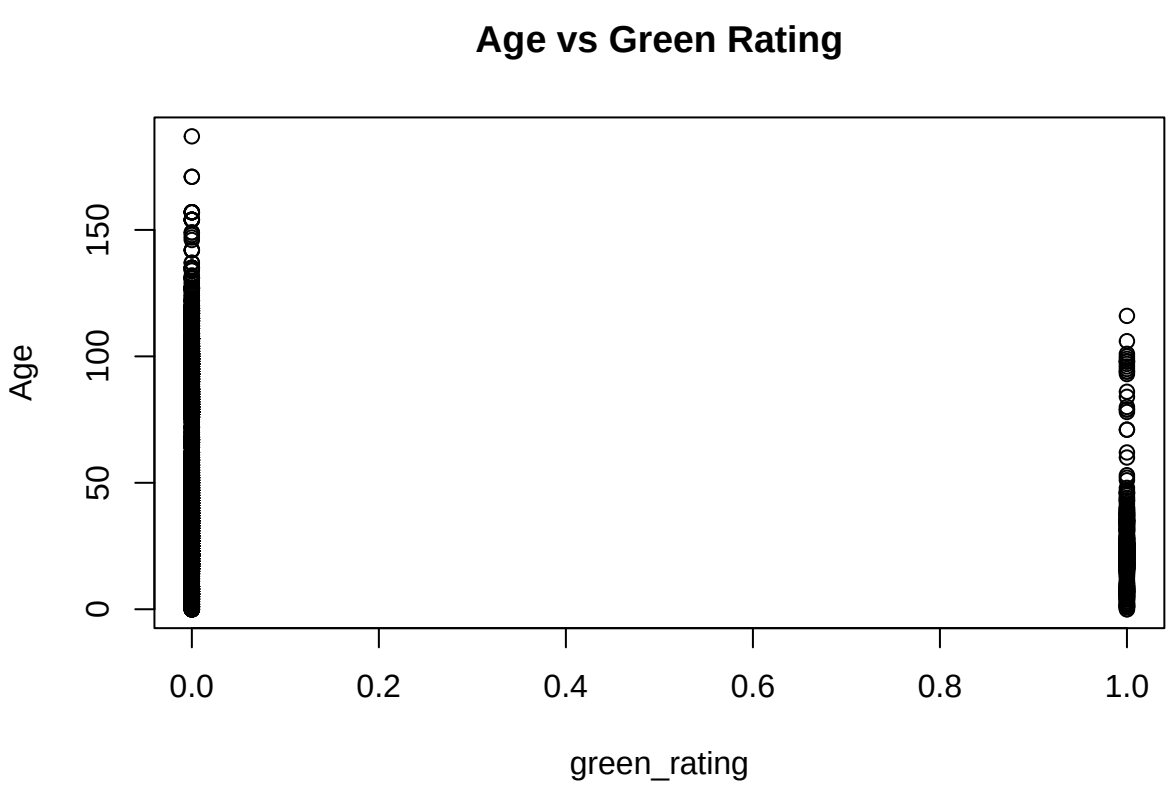




Additionally, the analyst did not account for certain cities being more green then others. Certain clusters will have higher rents regardless of green. Therefore, failing to take this into account could be undercutting additional revenue. Another flaw in her analysis is she failed to account for the future value of the additional expense incurred to make the building green. As shown below the future value of 5 million dollars in 8 years is in the 7 millions. This certainly should be accounted for when analyzing the potential consequences of a green building.

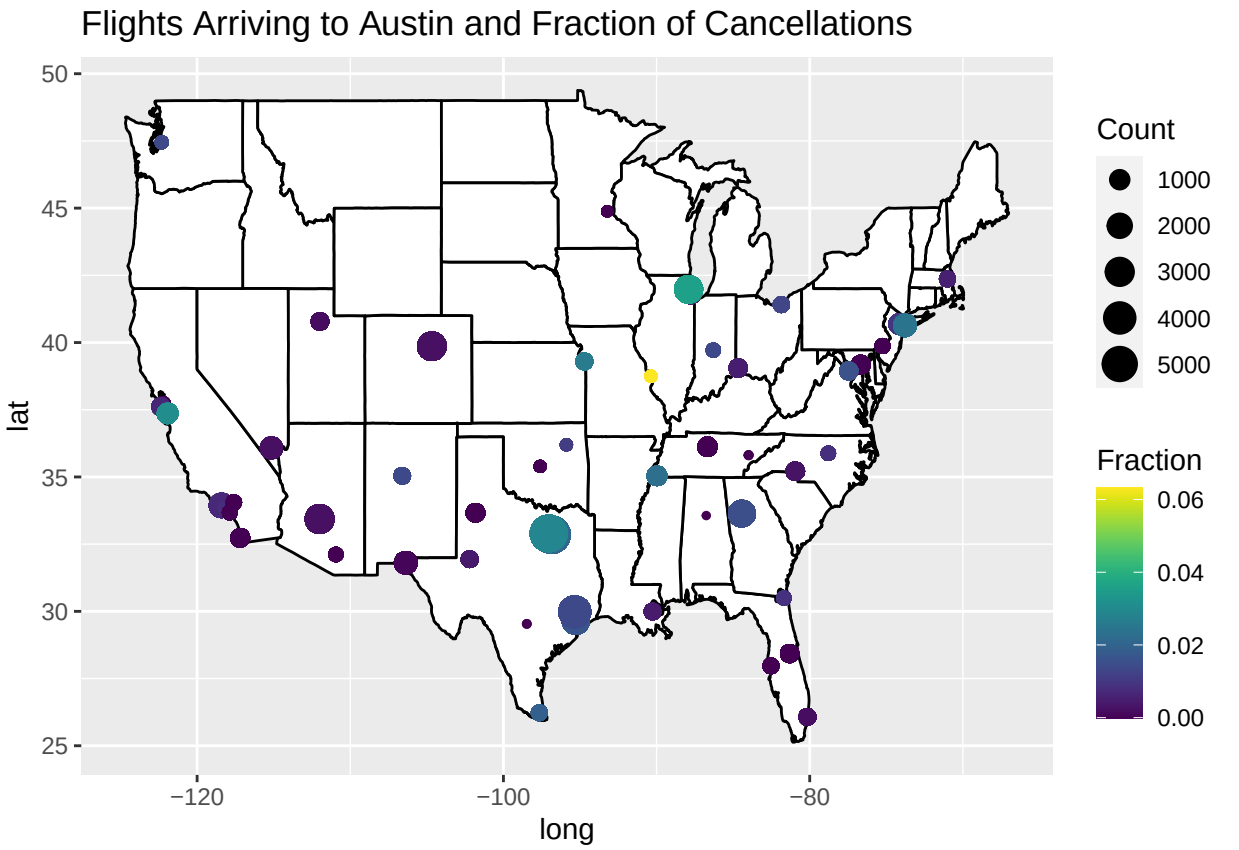


A potential confounding variable is the age of the complex. Since green technology is rather new, it makes sense to remove the non-green rentals whose age are greater than that of the max age of the green buildings. The plot below shows that this value is at about 100 years old.

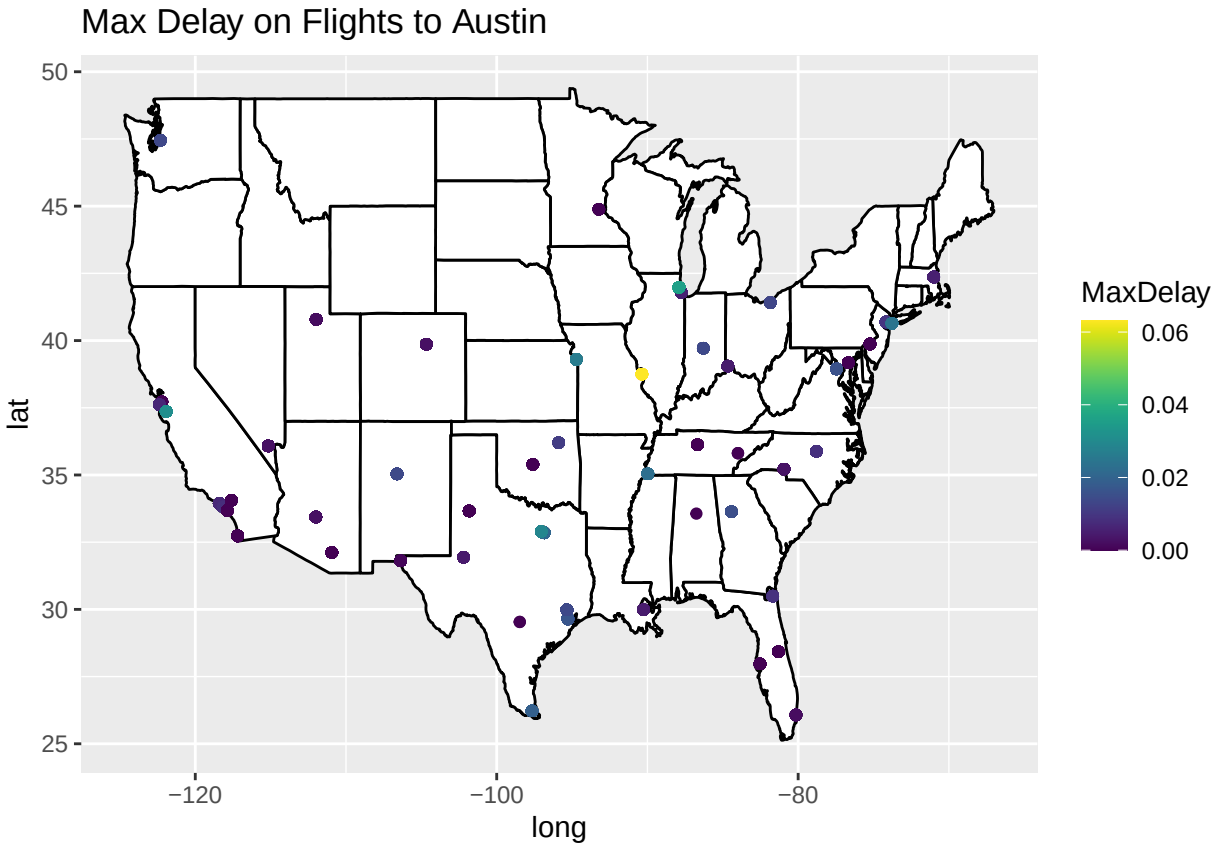


Even though there are many flaws in this study, choosing a green building could be a viable option, but further analysis that address our issues and confounding variables would be needed to be ensure that going green is a good decision.

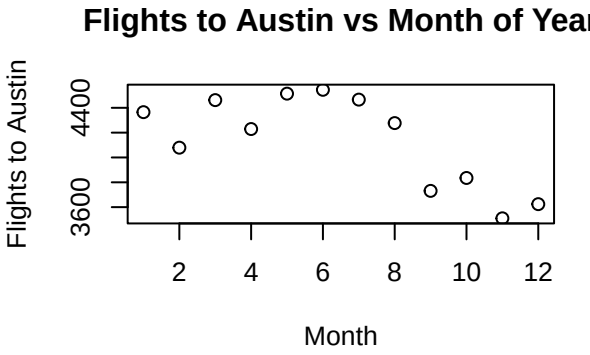
Visual story telling part 2: flights at ABIA

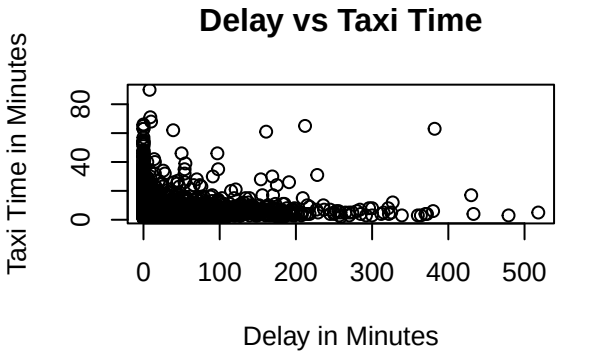


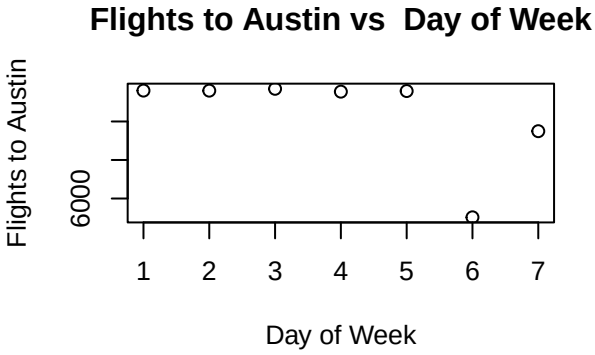
Above is a count of the origin locations of all the flights to Austin for the year 2021. As expected the most flights come from the Texas area.

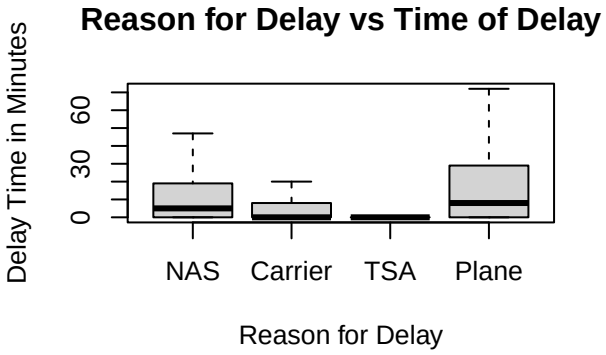


Above is the max delay for flights to Austin. It is interesting to note that Chicago had the largest delay.









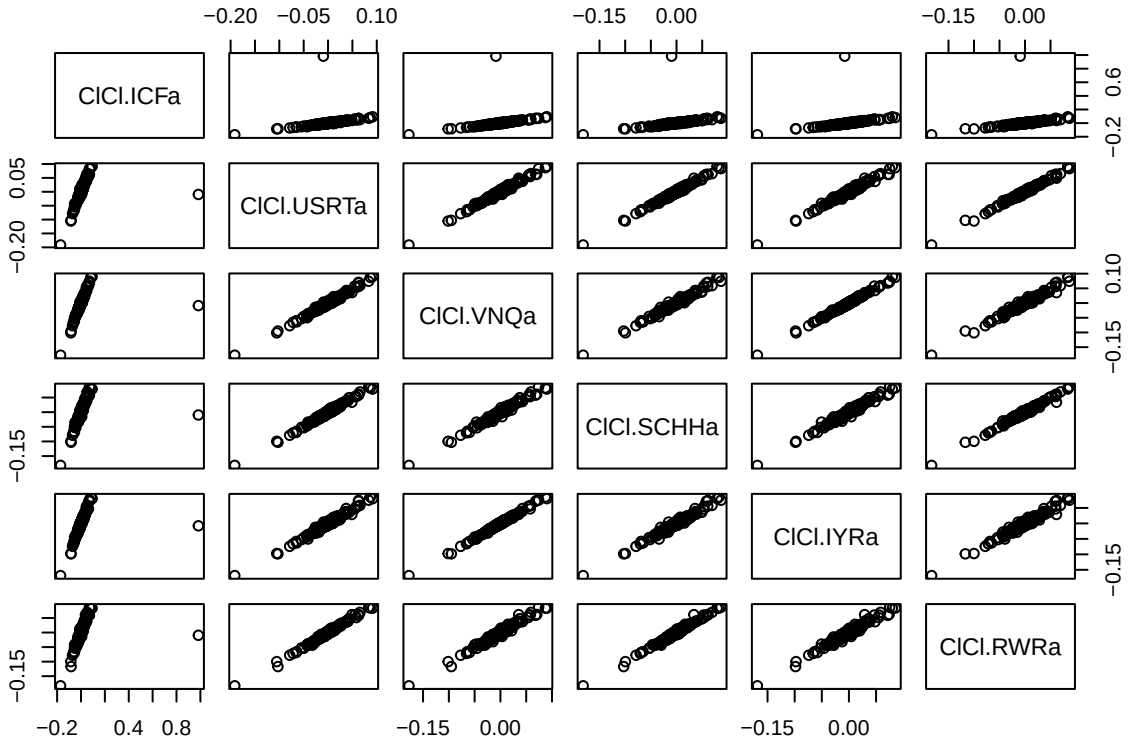
In the above plots we see lower flights to Austin in the months from September - December. We also see as delay increases, taxi time tends to decrease at a logarithmic rate. In the third plot we see that most days have the same amount of flights except for Saturday which is the value 6. In the last graph we see that the largest reason for delay comes from plane issues, followed by NAS which is delay due to weather adjusted for how the airport treats other weather issues.

Portfolio modeling

We selected 6 REIT ETFs to create a prediction portfolio. Modeling for the next 20 days, finding different weights was the goal. We looked at weightings based off of the 5yr, 3yr, and 1 yr returns for the portfolio. The goal was to find which of these portfolio weight combos would beat the straightline weight of each ETF. Each ETF did well in its own right, and certain did better than others based on the time frame used.

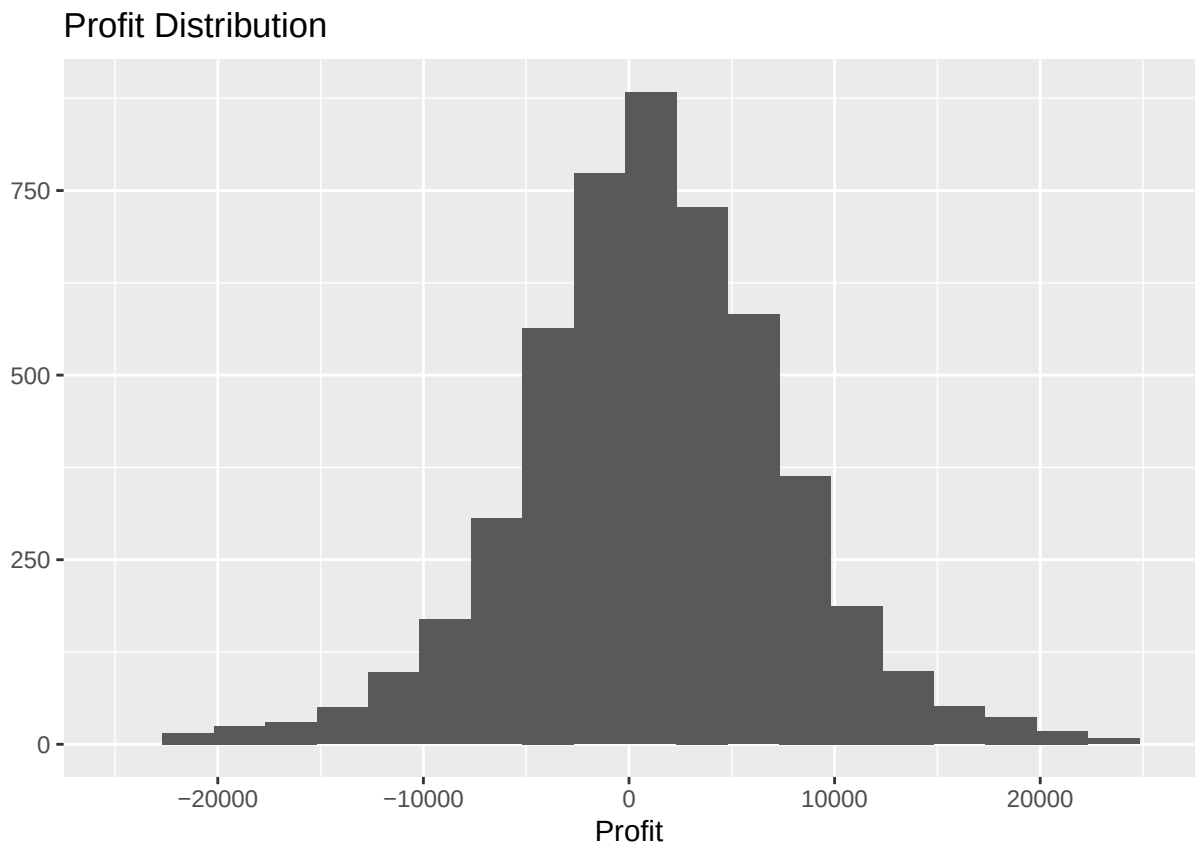
ETFs

- ICF - iShares Cohen & Steers REIT ETF;
- USRT - iShares Core U.S. REIT ETF;
- VNQ - Vanguard Real Estate Index Fund ETF;
- SCHH - Schwab U.S. REIT ETF;
- IYR - iShares U.S. Real Estate ETF;
- RWR - SPDR Dow Jones REIT ETF;



This pairwise plot demonstrates the relationships between each ETF. Clearly, they are all positively correlated. This is due to the ETFs being contained within the same industry, real estate. Real estate is a great way to hedge against inflation and diversify your overall portfolio; however it would be highly unusual for real estate to make up the entirety of your investments.

Total wealth after 20 days in all 6 ETFs, equally weighted.



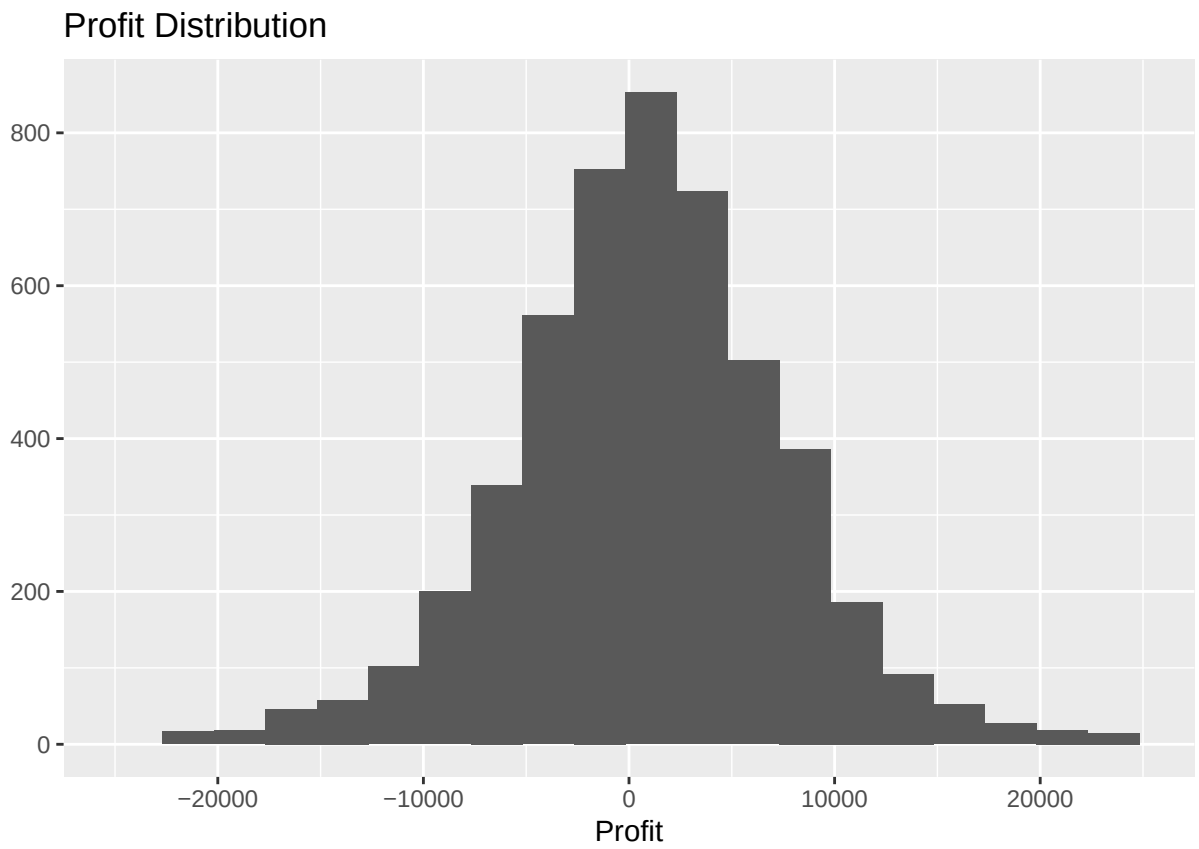
This is the Profit plot of the evenly weighted holdings of all 6 ETF REITs. This method gave a mean of $1,067. And a range from -$8,535 to $10,669.

Portfolio 1

Weighted based on 1yr returns

Representing the possible range of total wealth after 20 days in the market.

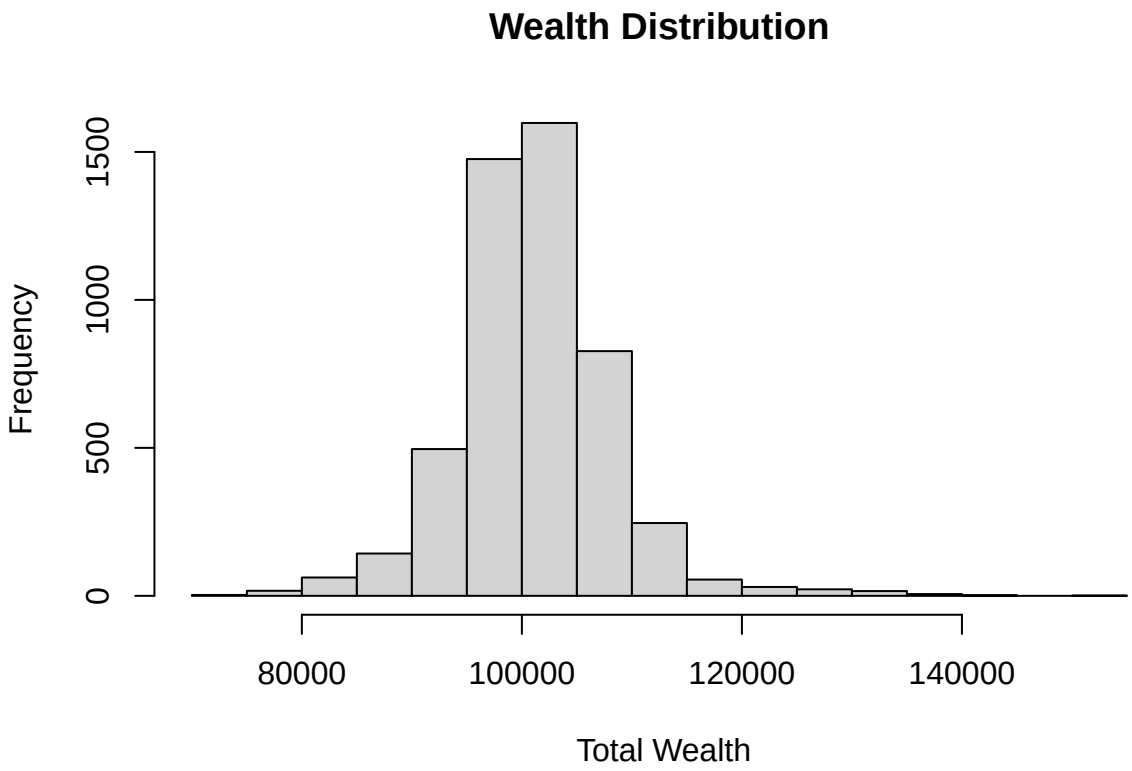

Wealth Distribution
Frequency
0
500
1000
1500
80000
100000
120000
140000
Total Wealth



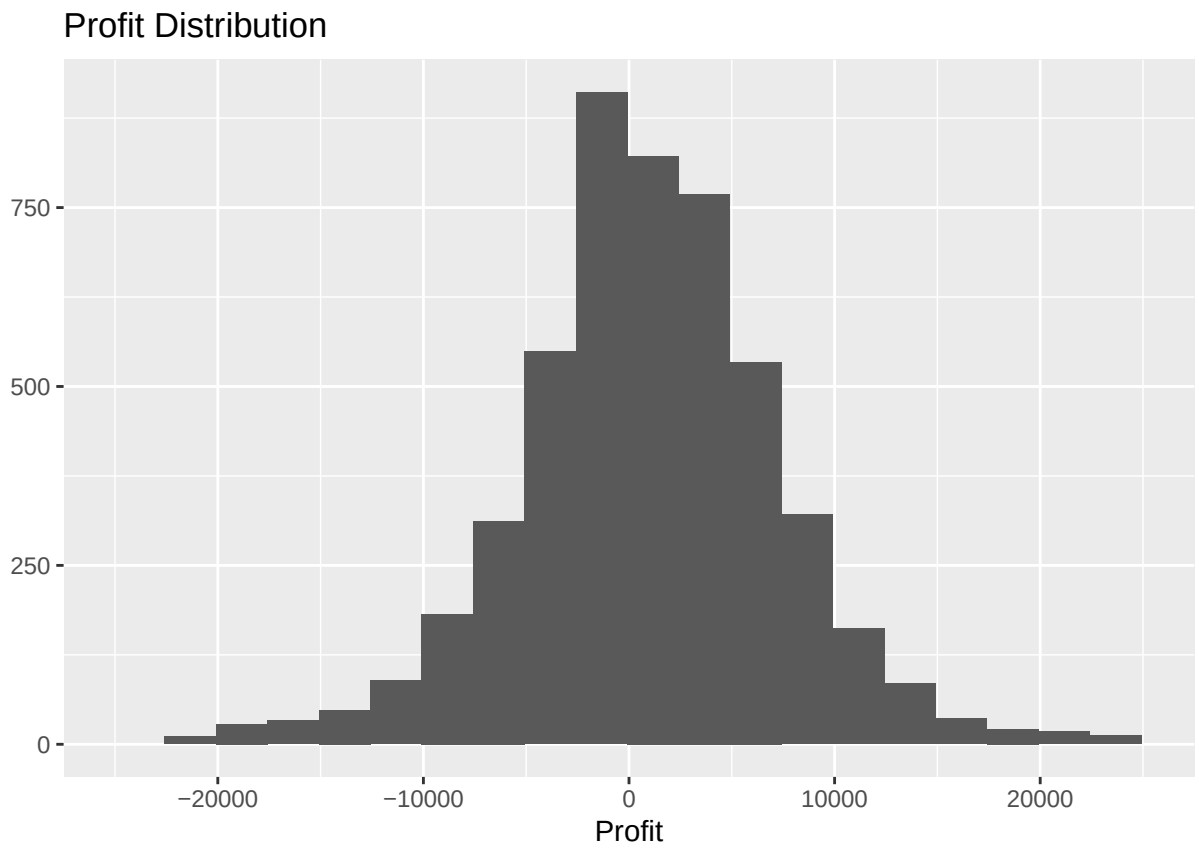
This model demonstrates the total possible profit of this portfolio after 20 days in the market. profit will remain between -$8,886 and +$11,063 with 95% confidence with the mean at $1,089.

Portfolio 2

Weighted based on 5yr returns



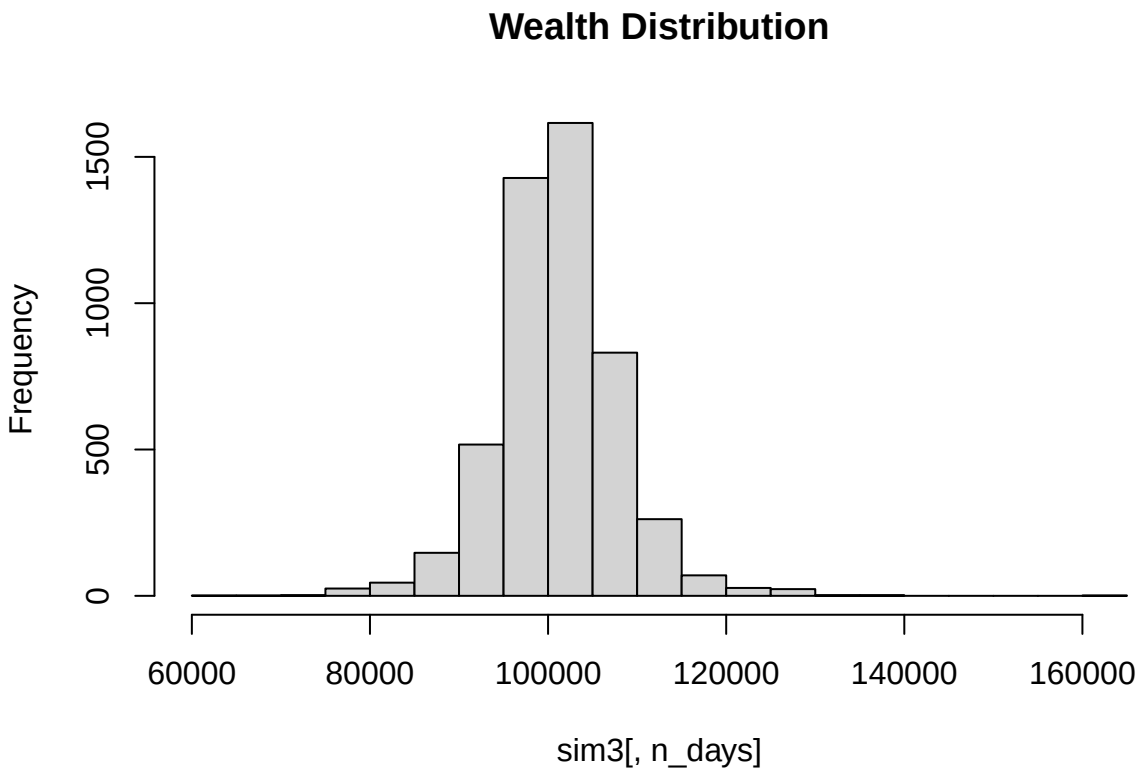
The total wealth in the portfolio 2 based off of weighting on the 5yr return.



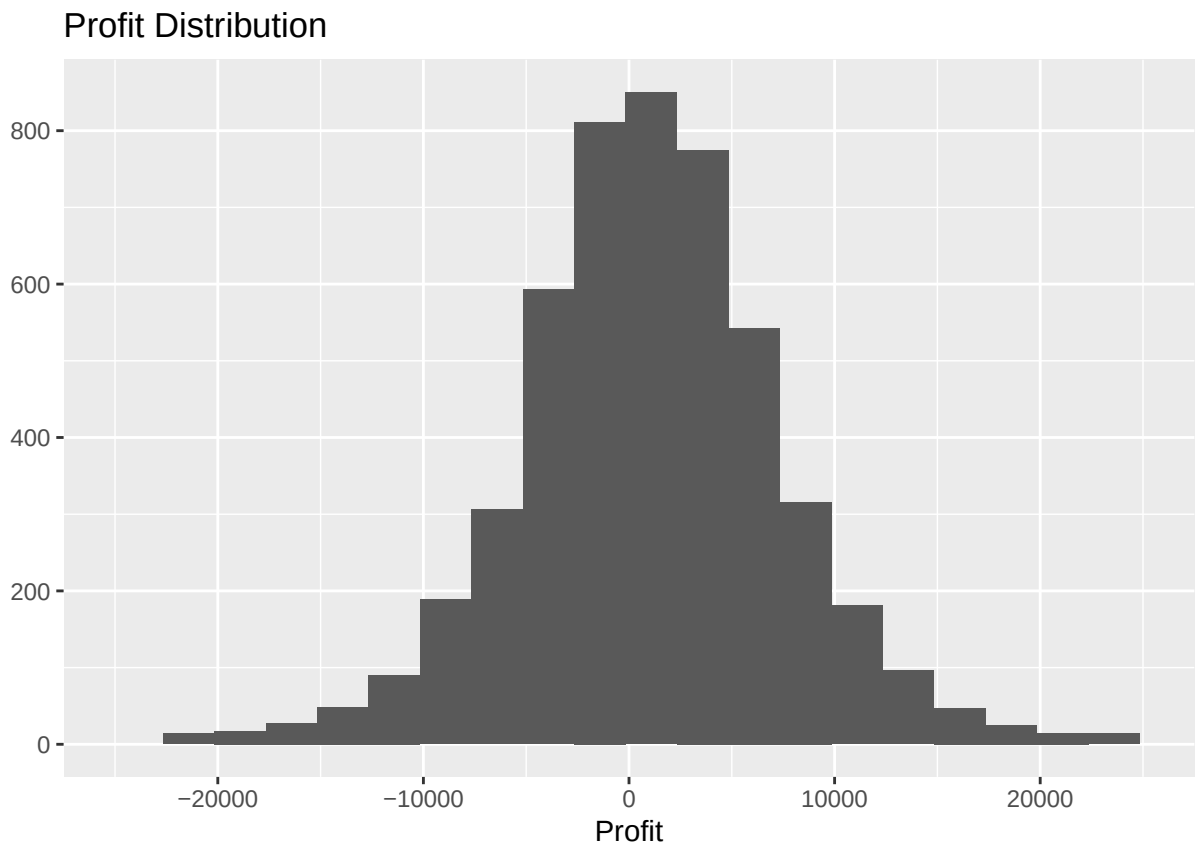
Profit range for portfolio 2 will remain between -$8,441 and +$10,773 with 95% confidence with a mean of $1,166.

Portfolio 3

Weighted based on 3yr returns



Wealth from portfolio 3 based off of the 3yr return.



Profit will remain between -$8,374 and +$10,551 with 95% confidence with a mean of $1,088.

Conclusion

These three portfolios, weighted based on different years of returns, create different risk and reward deviations. Although they are similar, the weighting is varied for each ETF. ETFs are diversified in nature, therefore diversifying in ETFs should not make one's portfolio any safer.

However, examining the 1 yr profit distribution we see a slight slant to the positive side. Perhaps indicating that certain investments have become popular as of late. This method beat the straighline weighting process; so did the 3yr and the 5yr.

If one portfolio weight set had to be choosen then the 5yr return would have to be the winner. It shows consistent earnings over the long run and it has the highest mean ($1,166) for the distribution to center off of. The 5 yr matches both statistically and intuitively. This model also beat the mean of the straight line weighted model by about 10%.

Market segmentation

In this problem, lets look at the social_marketing dataset, and try to discover any insights.

First, lets create a "category" column, that will put each user into a category based on the max number of tweets they have in a particular category:

```
## # A tibble: 35 x 2
## # Groups:   Category [35]
##    Category             n
##    <chr>            <int>
```

```
##  1 chatter           2538
##  2 health_nutrition  1130
##  3 cooking            541
##  4 politics           439
##  5 photo_sharing      418
##  6 sports_fandom      337
##  7 college_uni        323
##  8 online_gaming      267
##  9 travel             229
## 10 news               227
## # ... with 25 more rows
```

After doing this, we can see that the number one category, by a long shot, is chatter. This category is not really useful for understanding the market, as there are active users who could fall into many different spheres of twitter, so lets remove this category, and run the analysis again.

```
## # A tibble: 34 x 2
## # Groups:   Category [34]
##    Category             n
##    <chr>            <int>
##  1 health_nutrition  1271
##  2 photo_sharing     1250
##  3 cooking            603
##  4 politics           548
##  5 current_events     494
##  6 sports_fandom      461
##  7 travel             395
##  8 college_uni        369
##  9 online_gaming      318
## 10 news               274
## # ... with 24 more rows
```

As we can see, health_nutrition, photo-sharing, and cooking are the to categories of these engaged users. This makes sense, as health_nutrition is a core value of VitaminWat... erm I mean NutrientH20's brand. One insight that NutrientH20 could take away is to start a photosharing campaign, that might engage their users who alraedy love to photo share. Another insight could be to advertise on the cooking channel, or target audiences in the cooking social-media verse.

Author attribution

```
## Number of unique training words:  1411
```

```
## Number of unique testing words:  1414
```

```
## [1] "Different words between training and testing set:"
```

```
##  [1] "returned"   "worried"    "expensive"  "smith"      "programmes"
##  [6] "served"     "condition"  "invested"   "video"      "dissidents"
## [11] "troubled"   "rating"     "gms"
```

```
## Number of different words between training and testing set:  13
```

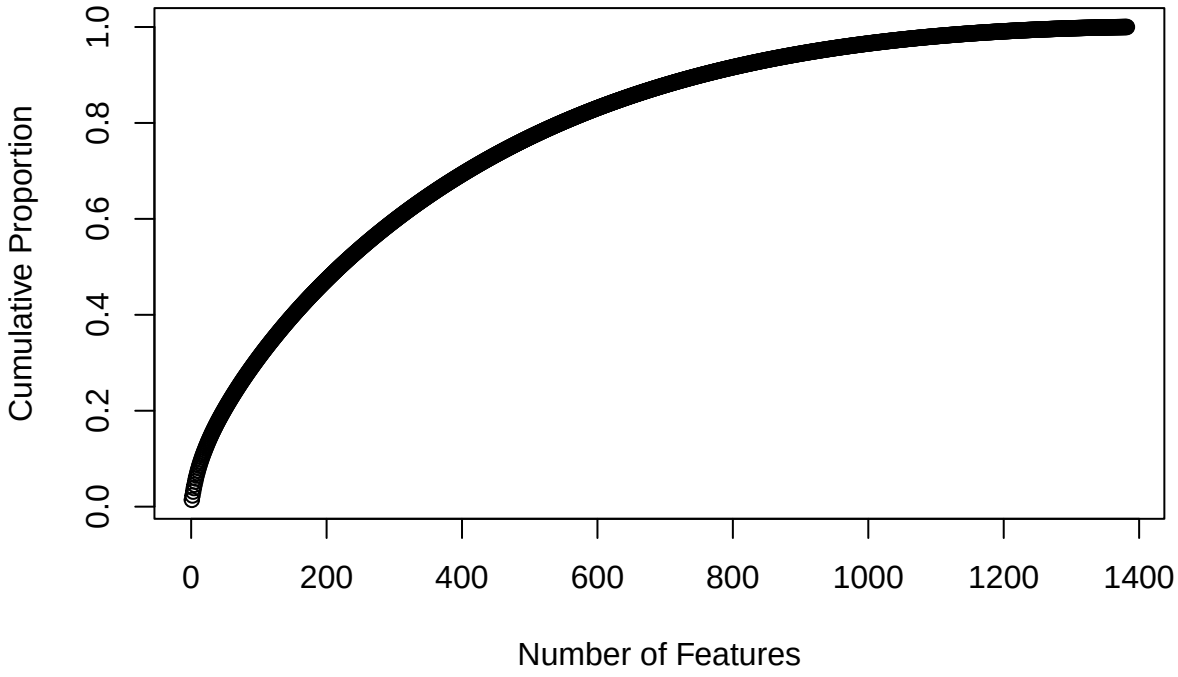
```
## [1] "Without Laplace Smoothing - Naive Bayes Results:"
```

```
## Accuracy:     0.38
## Precision:    0.39
## Recall:       0.85
## F-measure:    0.41
```

```
## [1] "With Laplace Smoothing - Naive Bayes Results:"
```

```
## Accuracy:     0.38
## Precision:    0.39
## Recall:       0.85
## F-measure:    0.41
```

```
## [1] "PCA Plot:"
```




```
## [1] "Random Forest Results:"
```

```
## Accuracy:     0.98
## Precision:    0.98
## Recall:       0.98
## F-measure:    0.98
```

In order to predict the authorship, we would have to build a model. For this problem, we decided that we can build 2 different models: Naive Bayes and Random Forests. The data is provided by text files, located on the path: data/ReutersC50. In this folder, there is a train set (C50train) and a test set (C50test). Before reading in text, we would need to pull out needed data, such as the files that contain the text to add to the corpus and the author who's text is in the file. When iterating through the list of files, we save the file name and the author, or the label, would be the directory that we are searching through replicated for the number of text files that author has. Once we do this for both the train and test set, we can then use readerPlain, to read in the text of the files in english. This complied list of text can we converted into a simple corpus. The words in the corpus are not standardized and they are not clean, therefore, we needed to conduct some text preprocessing. To clean the text, we converted all text to lower case, removed numbers, removed punctuation, removed excess white space, and got rid of stopwords using the SMART set. This text is now ready to be tokenized. To tokenized, we used a Document Term Matrix. This matrix would list out all documents along the rows and unique words in the corpus along the columns with the values in the matrix being the count

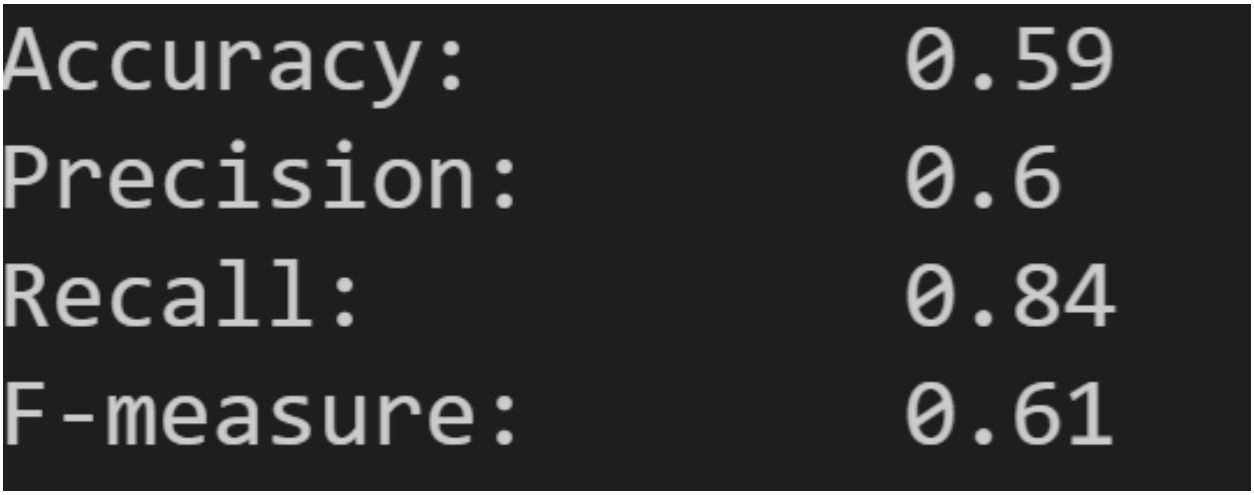
of that unique word appearing in that a certain document. We also needed to get rid of rare cases so we removed a certain level of sparse terms (.975, .95) as well. Our training text is now ready. As for our training labels, they do not to be changed or cleaned; they can be used in the model as is. We conduct the exact same progress for our test data as well.

Our first model is Naive Bayes. With Naive Bayes, we got an accuracy of 39% which is better than the baseline accuracy of random selection of 1/50 or 2%. This model is good start to our goal of predicting. We wanted to improved our Naive Bayes model. After following the calculations behind Naive Bayes, we saw that unknown test terms that are not seen in the train set have the potential of lowering our accuracy. We then looked into how to handle these unknown values. One thought was to add an UNK (unknown) token to the tokenizer to default all unknown words to a constant (or even a guessing function depending on the context of unknown words as seen in Word2Vec/Fasttext word embeddings based on potential word roots/similarities), similar to how an UNK token can be added to a BERT transformer tokenizer. However, that would probably result in handling many corner cases within the Document Term Matrix. We then explored other options such as Laplace Smoothing. Laplace smoothing will add one (or other constant) to the number of instances a word appears and add one (or other constant) times the number of instances a word appears to the number of total words. After applying Laplace (with many constants), we did not see any difference. This could be because of larger dimensional of the data or that there are not many unknown words in the test set. This lead us to explore the data more. We compared the unique words in each set; there are only 13 differing words between the train and test set. Therefore, unknown words do not cause much of a change as they are less than 1% of all words and therefore can be ignored in the context of this problem. To improve our predictions, we looked towards other models.

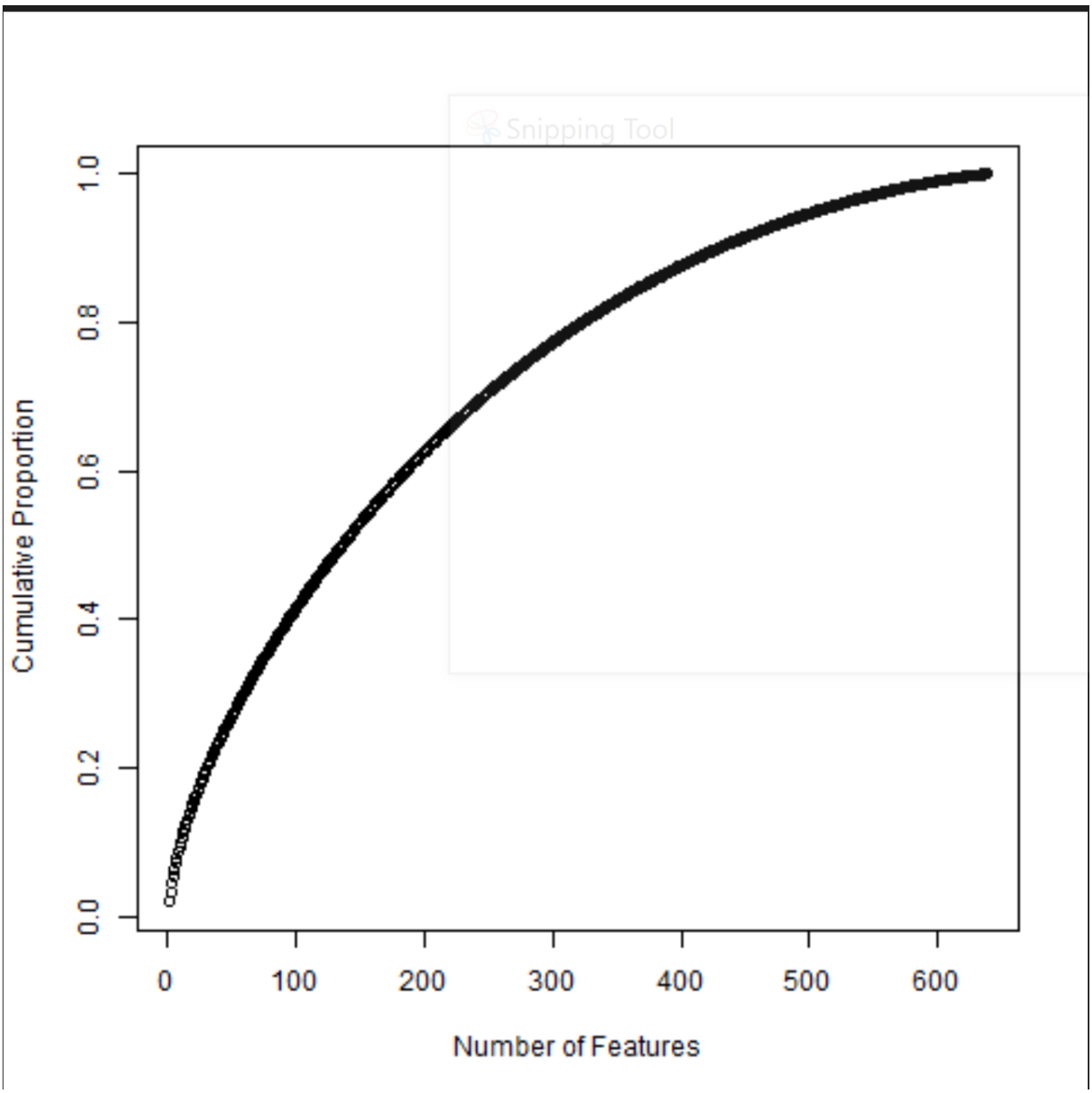
Our second model is Random Forests with PCA. To run Random Forests, we had to conduct some additional formatting and preprocessing. As discovered in the Naive Bayes model, there are some unknown words in the data. These unknown words are also unknown features between the test and train set. Random Forests needs to be aware of all features in order to generate predictions. To do, we got the intersect, or common words, between train and test set vocabularies. We kept only these similar words for our training matrix and started to conduct PCA. For PCA, we would have to get the tf-idf values and run it through PCA. Looking at the PCA plot of the number of features and the Cumulative Variance, we can see many interesting points. At around 200 features, around 50% of the features explain the variance and at around 1000 features, around 100% of the variance is explained. We do not need to run 1400 features, if around 1000 features will also yield similar results. Therefore, we can train with only the first 1000 features. Looking at the results, we see that, without leaking the training dataset, the model is giving 98% correct accuracy. As for the other values, because the data set has balanced classes, there is no need to interpret them in this context however they are also equal to 98%. This is probably a better model probably because Random Forests has the ability to define sets of similar words under a branch and, therefore, can isolate authors much better.

It is also interesting to note that when running sparse terms of .95, the naive bayes model accuracy increased and when using 500 features for PCA on random forests, the random forest accuracy decreased slightly. This is seen below with the first image being the Naive Bayes results, second image being the PCA plot, and third image being the Random Forests results.

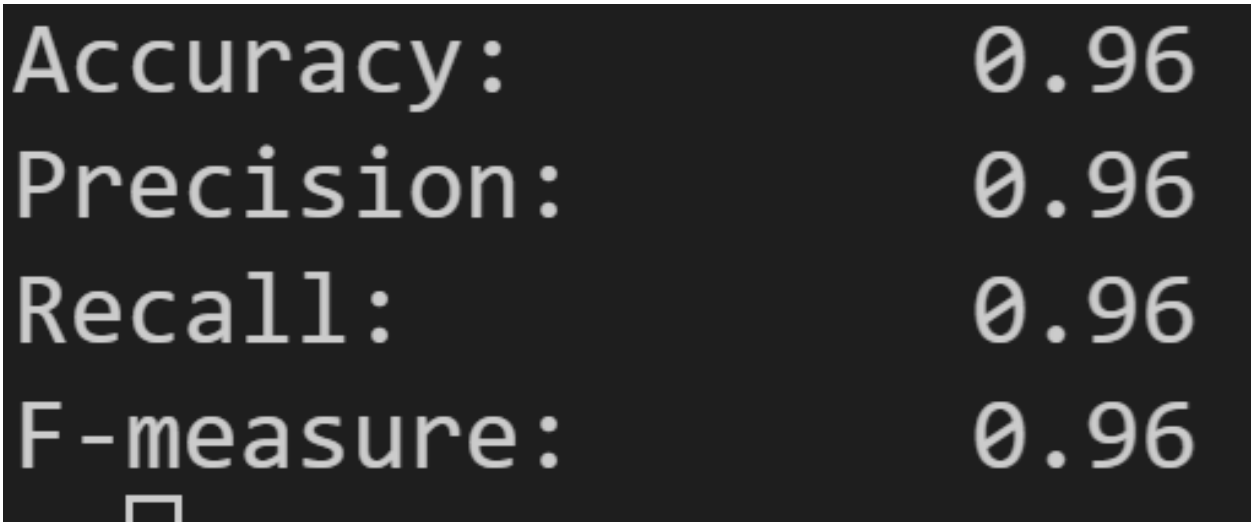
Naive Bayes Results:



PCA Plot:



Random Forests Results:



Association rule mining

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    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 ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9783 transaction(s)] done [0.01s].
## sorting and recoding items ... [154 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [736 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
## [1] "High Support"
```

```
##     lhs    rhs                support   confidence coverage lift count
## [1] {}  => {soda}              0.1181642 0.1181642  1        1    1156
## [2] {}  => {yogurt}            0.1131555 0.1131555  1        1    1107
## [3] {}  => {rolls/buns}        0.1389144 0.1389144  1        1    1359
## [4] {}  => {other vegetables} 0.1438209 0.1438209  1        1    1407
## [5] {}  => {whole milk}        0.1879791 0.1879791  1        1    1839
```

```
## [1] "High Lift"
```

```
##     lhs                      rhs              support     confidence coverage
## [1] {specialty fat}      => {margarine}      0.001124399 0.3055556  0.003679853
## [2] {softener}           => {detergent}      0.001022181 0.2040816  0.005008689
## [3] {popcorn}            => {salty snack}    0.001533272 0.2631579  0.005826434
## [4] {liquor}             => {bottled beer}   0.002248799 0.3188406  0.007053051
## [5] {liquor (appetizer)} => {canned beer}    0.001431054 0.2121212  0.006746397
## [6] {salt}               => {sugar}          0.001737708 0.1666667  0.010426250
## [7] {frozen dessert}     => {frozen meals}   0.001328836 0.1274510  0.010426250
## [8] {flour}              => {sugar}          0.002964326 0.1746988  0.016968210
## [9] {grapes,pip fruit}   => {tropical fruit} 0.001226618 0.4800000  0.002555453
##     lift      count
## [1]  5.726533 11
## [2] 13.490072 10
## [3]  8.908213 15
## [4]  6.525559 22
## [5]  5.748426 14
## [6]  5.381188 17
## [7]  5.007442 13
## [8]  5.640522 29
## [9]  5.914156 12
```

```
## [1] "High Confidence"
```

```
##     lhs                            rhs                support     confidence
## [1] {nuts/prunes}               => {whole milk}       0.001124399 0.3666667
## [2] {herbs}                     => {other vegetables} 0.005315343 0.3714286
## [3] {grapes,pip fruit}          => {tropical fruit}   0.001226618 0.4800000
## [4] {citrus fruit,grapes}       => {tropical fruit}   0.001226618 0.3870968
## [5] {dessert,root vegetables}   => {other vegetables} 0.001226618 0.5000000
## [6] {curd,root vegetables}      => {whole milk}       0.001431054 0.3589744
## [7] {butter,root vegetables}    => {whole milk}       0.002248799 0.4313725
## [8] {citrus fruit,domestic eggs} => {whole milk}      0.001022181 0.3571429
## [9] {sausage,soda}              => {rolls/buns}       0.002146581 0.3559322
##     coverage    lift     count
## [1] 0.003066544 1.950571 11
```

```
## [2] 0.014310539 2.582577 52
## [3] 0.002555453 5.914156 12
## [4] 0.003168762 4.769481 12
## [5] 0.002453235 3.476546 12
## [6] 0.003986507 1.909650 14
## [7] 0.005213125 2.294789 22
## [8] 0.002862108 1.899907 10
## [9] 0.006030870 2.562240 21
```

```
## [1] "Low Support, High Confidence"
```

```
##     lhs                           rhs                 support     confidence
## [1] {grapes,pip fruit}        => {tropical fruit}    0.001226618 0.4800000
## [2] {dessert,root vegetables} => {other vegetables} 0.001226618 0.5000000
## [3] {butter,root vegetables}  => {whole milk}       0.002248799 0.4313725
##     coverage    lift     count
## [1] 0.002555453 5.914156 12
## [2] 0.002453235 3.476546 12
## [3] 0.005213125 2.294789 22
```

In order to get interesting insights into the association rule mining, we needed to format the data properly. First, an association can only be made if there are at least 2 items. In the dataset provided, items are listed from column 1 to column 4. The number of items bought correlates to the number of columns filled in from left to right. Therefore, if a transaction has a missing items in column 2, then they are also missing items in column 3 and 4. If that is the case, then the only filled in column is column 1 which means that the transaction only has one item and needs to be removed from the data set. To do this, we locate all rows that have the column 2 empty and remove them.

Next, in order to replicate our dataset similar to the one done in class, we had to perform a table pivot on rows. We wanted the data frame to contain each item on its own row along with the row number which would represent the transaction number. This means that there will be many of the same row number value in the transaction column. We also wanted to remove all rows that do not have an item associated with it because of the pivot.

Then, we factorize the transactions and create a list based on the transaction number and the items that were bought in the transaction in order to format the data to be converted to an arules type.

When converting to arules and running the apriori algorithm (with a support of .001 and confidence of .1) on it, we can then see some interesting insights:

- The most stable groceries that are bought independently (highest support) are soda, yogurt, rolls/buns, other vegetables, and whole milk. The highest support is .1879791 with whole milk. This makes sense as milk is usually the most brought item at the markets.
- When looking at the associations that have the most lift, we see that the highest lift value is a 13.490072 is at softener -> detergent. This makes sense as these are both laundry items. Other rules are strongly related to items to each other such as popcorn and salty snack (salty snacks), liquor and bottled beer (alcohol), and flour and sugar (baking). Out of this set, the only association that has 2 items on the left side and one on the right is {grapes, pip fruit} -> tropical fruit.
- When looking at the associations that have the most confidence or conditional probability between items, we see that the highest confidence is .5 with {dessert,root vegetables} -> other vegetables. This could be warranted by the fact that people buy both dessert and vegetables in order to counteract their unhealthy diet choices.
- When looking at the associations that have high confidence and low support, we see that dessert,root vegetables -> other vegetables is seen again. This set shows associations of items that were rarely bought but, when bought, they were usually bought together.