Exercise 4

Robert Toto

4/18/2021

Problem 1: Clustering & PCA

AFter running both Clustering and PCA analysis, it seems that clustering is the better method for unsupervised learning on this dataset. Using 4 k-means++ clusters on scaled chemical property data, I was able to clearly distinguish between white and red colors with very little misidentification. I used 4 clusters because I needed a binary separation for color (which can be accomplished with an *even* amount of clusters) but also needed to assess grouping by quality, which is not a binary scale. Overall, the clustering did an excellent job of separating wines by color, and a poor job of identifying quality. The PCA scores did a good job of identifying color (not as good as clusters) and also a bad job of identifying quality. My preferred method is therefore clustering, and I will explain why below.

Before analyzing the color and quality by cluster, it is helpful to view the clusters across various chemical properties to see how the wines are separated along these characteristics. The four graphs below show how the four clusters are distributed across various pairings of chemical measurements. Each plot shows that each of the four clusters contains relatively specific chemical qualities, showing four distinct types of wine by chemical makeup. The clusters are not homogeneously interspersed with one another. In Figure 1, showing Chlorides vs pH, cluster 3 clearly contains wines with very low chlorides, while the other clusters contains wines of slightly higher chlorides that varied with pH level. In Figure 2, showing Alcohol vs Density, cluster 4 is a mainly low-alcohol group of wines, while cluster 3 has mainly higher alcohol content wines. In Figure 3, showing Citric Acid vs Sulfates, cluster 2 clearly contains the wines with the lowest citric acid of all the clusters, and cluster 1 contains the few wines with abnormally high sulfates. Finally, Figure 4, showing Free Sulfur Dioxide vs Total Sulfur Dioxide also shows clear groupings. cluster 2 has the lowest sulfur dioxides, cluster 4 has the highest of all the groups, and cluster 3 contains the mid-range sulfur dioxide wines. These preliminary plots show evidence of well-grouped, tight clusters that contain valuable information on distinct chemical groups.

(Note: In the Rmarkdown file, I have commented out the clustering and plotting code and inserted .png files to show these figures. If I were to knit, then the clusters would re-run in a different way, and my description here would not make any sense. If you wish to check the code, please just un-comment it, and it runs perfectly fine. I will also do this for the remaining figures (5-8) below.)

Figure 1

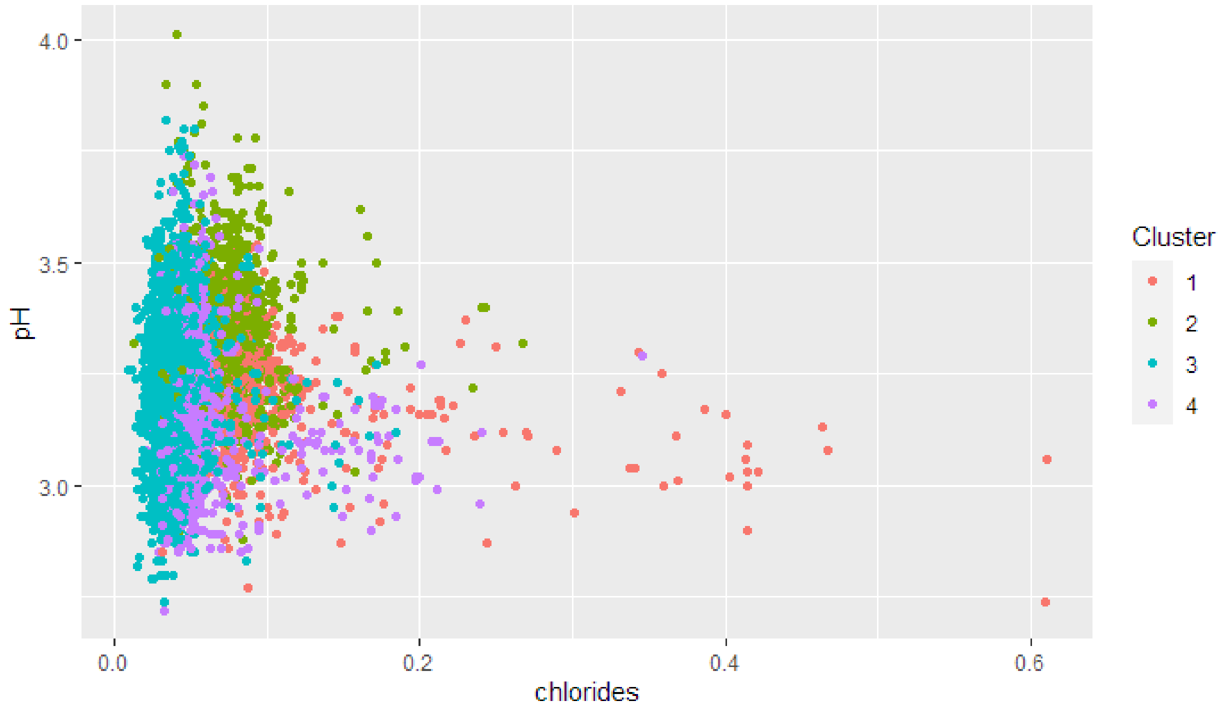


Figure 2

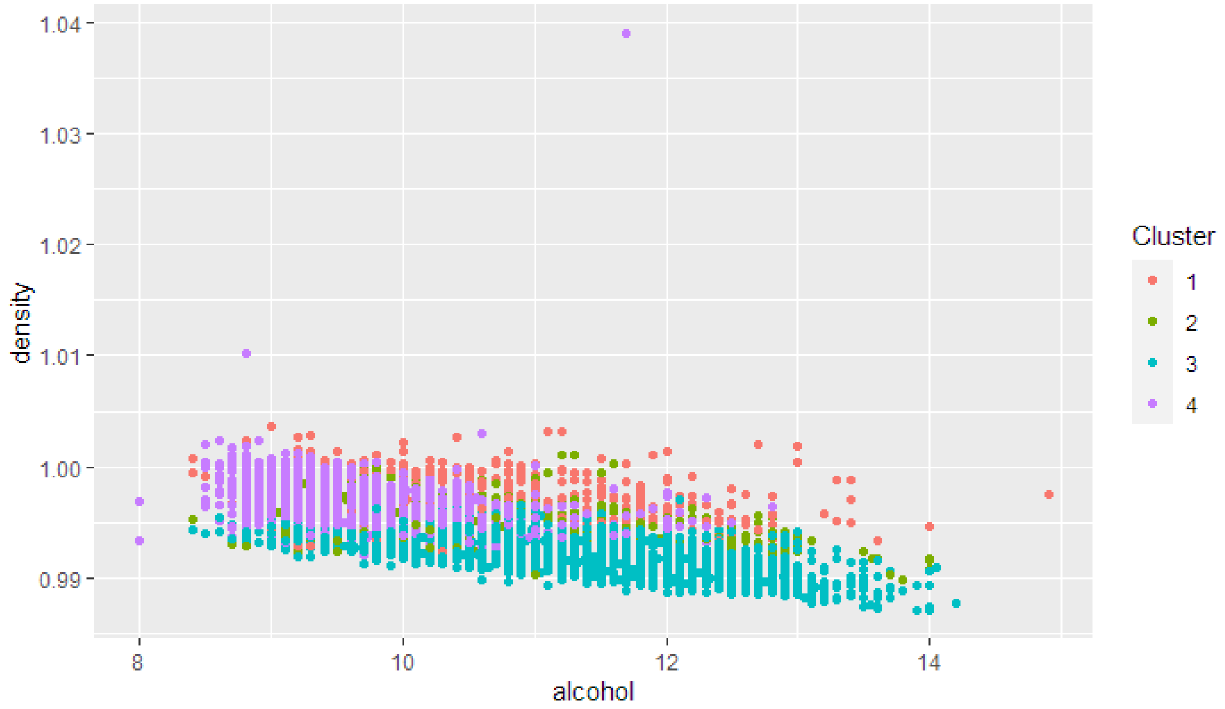


Figure 3

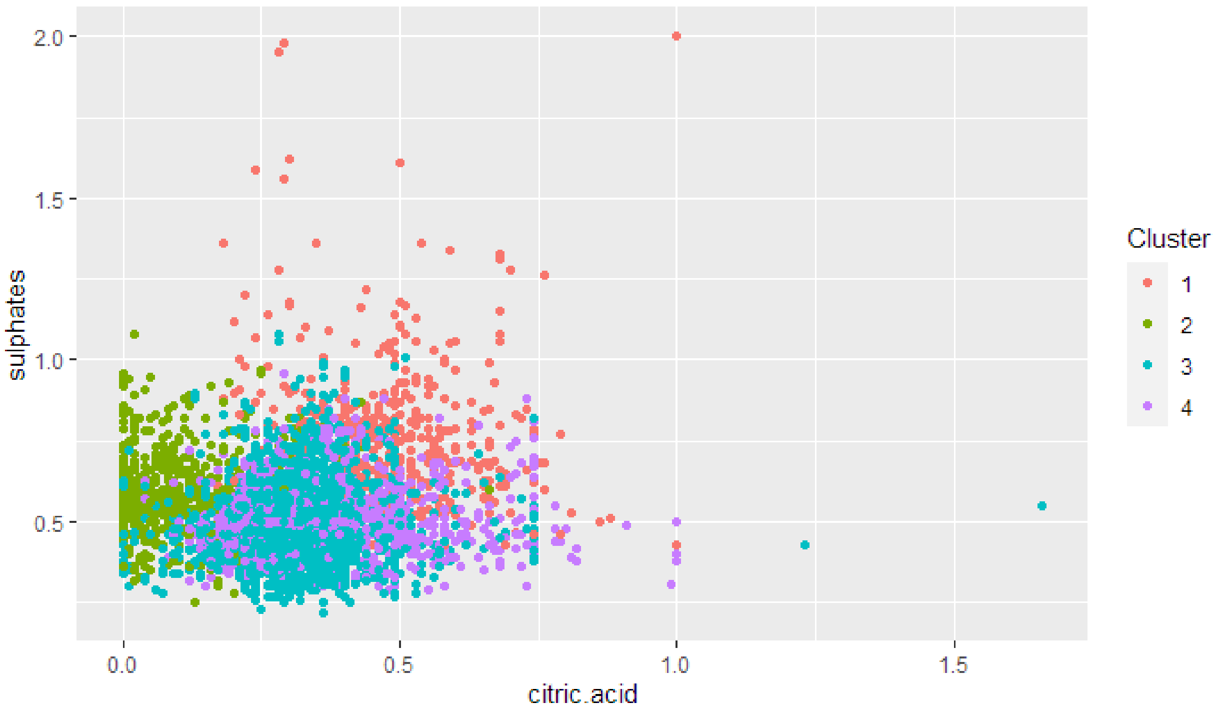
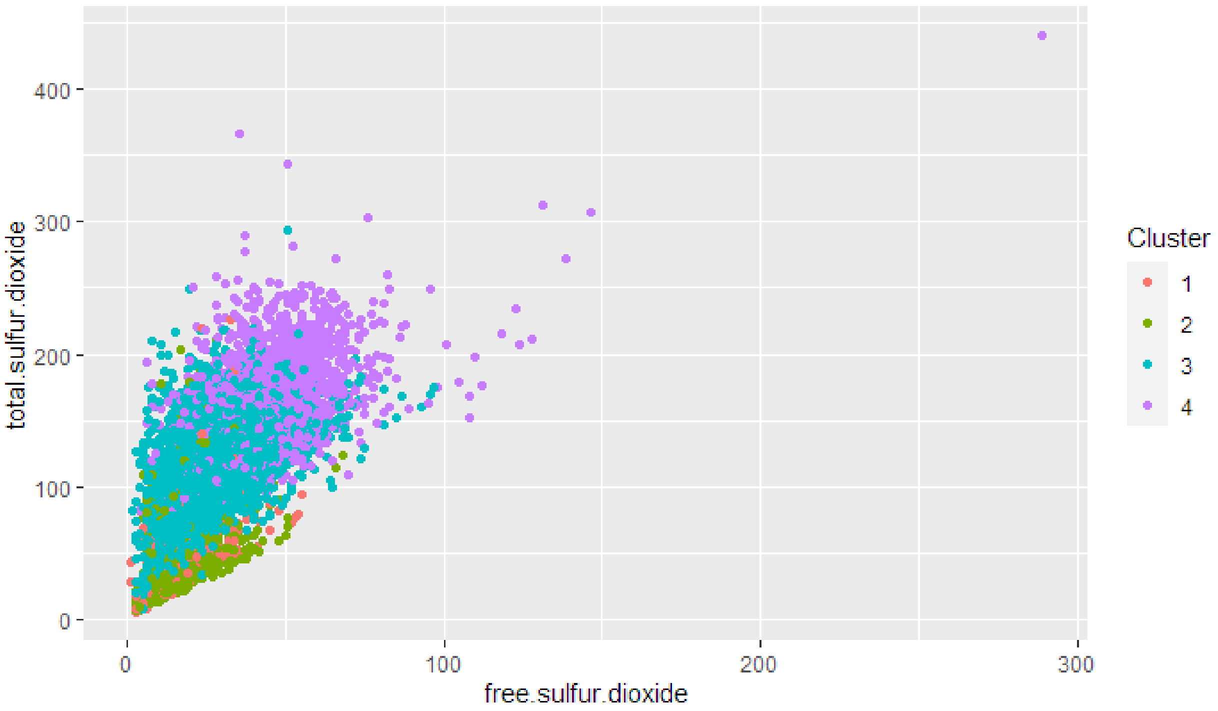


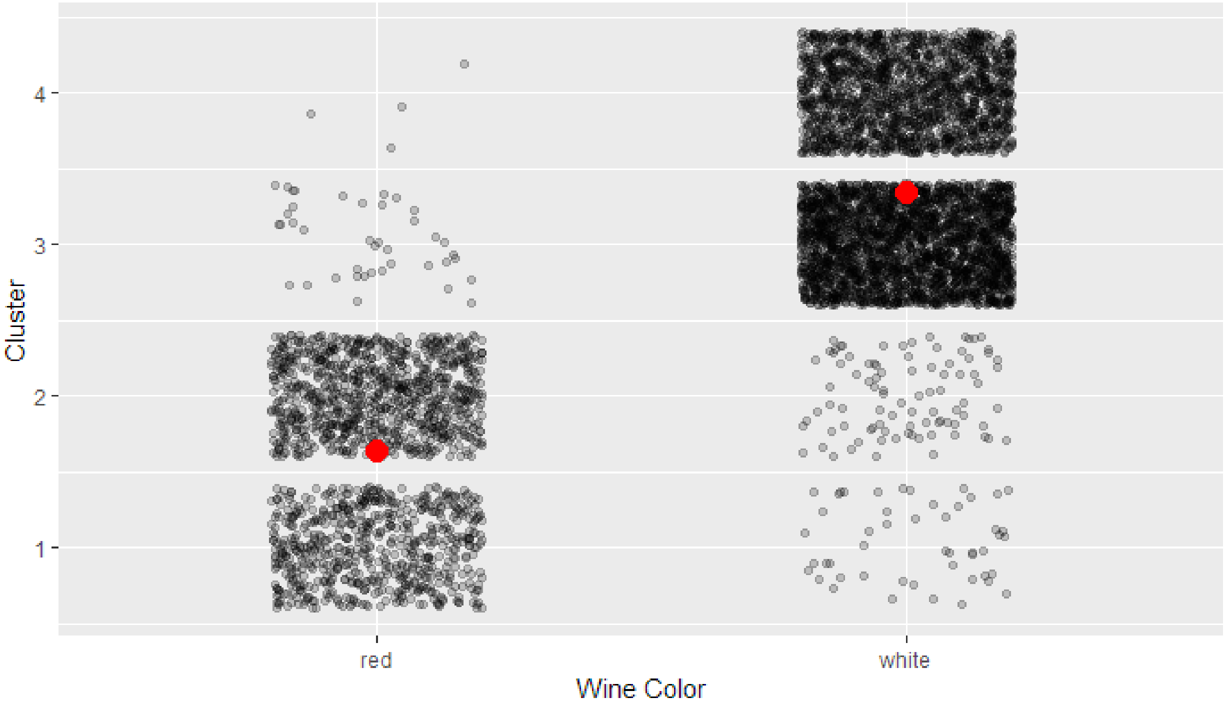
Figure 4

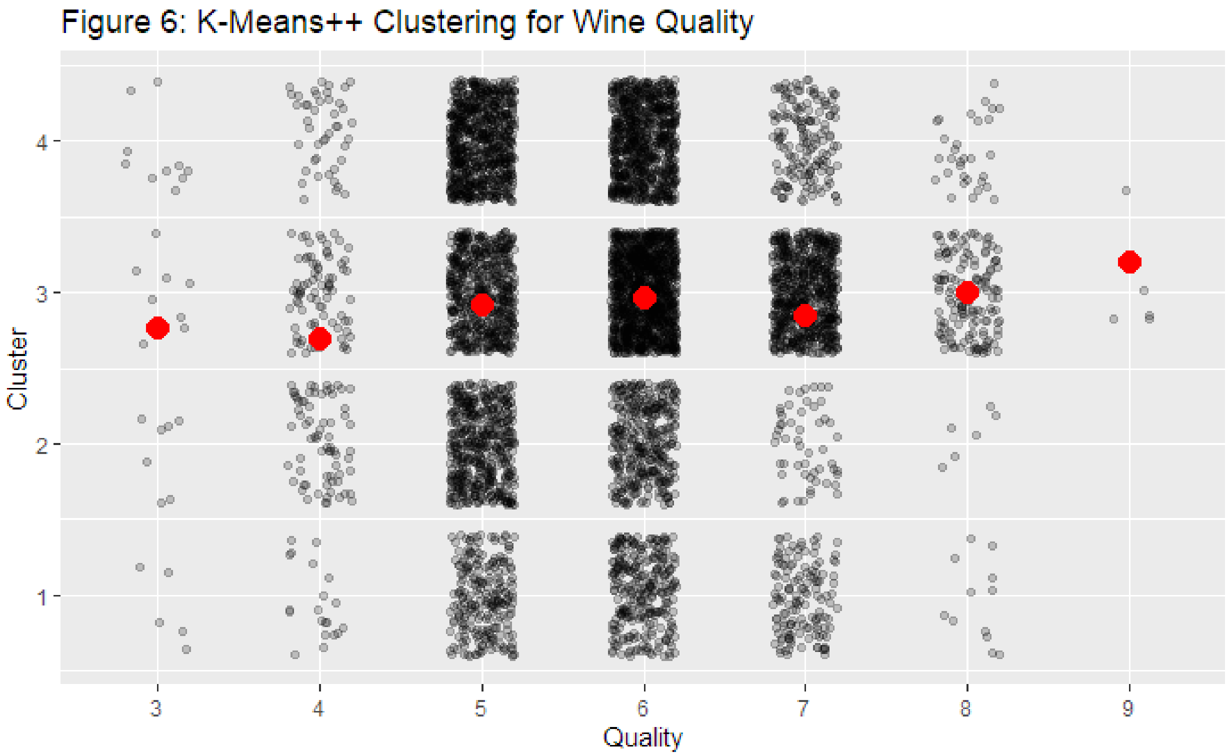


In Figure 5 below, the 4 clusters are plotted against wine color. The visual evidence is striking, demonstrating a clear distinction between red and white within the clusters. Clusters 1 and 2 contain nearly all the red wines, while clusters 3 and 4 contain nearly all the white wines. There are only a handful of red wines in clusters 3 and 4, and there are only a handful of white wines in clusters 1 and 2. Overall, these clusters do an excellent job of grouping the wines into their correct colors using only the chemical properties information.

However, as evidenced in FIgure 6, the clusters do *not* do a good job of distinguishing quality. It is worth mentioning that when 10 clusters are used, there is no improvement. Each cluster contains wines from nearly every quality group. There is a little bit of quality information captured when comparing cluster 1 to cluster 3. Cluster 1 contains mainly wines of quality 7 and below, while cluster 3 contains mainly wines of quality 5 and above. Other than this, there is very little quality-level information distinguished by clustering. This may not be surprising since the clusters did an excellent job of distinguishing red from white, and there is no relationship between color and quality. Red and white wines can both be of low or high quality. So, the clusters' predictive information is tied to color, which is unrelated to quality, and therefore contain little information on quality. Due to this trade-off, the clustering was not capable of distinguishing between higher and lower quality wines, but did an excellent job distinguishing color.

Figure 5: K-Means++ Clustering for Wine Color





For comparison, I created 2 PC scores to categorize the wines by color and quality. THe PCA did *not* do as good a job at classifying the wines by color, compared to the clusters. There is more visual ambiguity in the two PC scores across the colors. Figure 7 shows that the first PC score does an okay job of separating wines by color, but with a lot more overlap than the clusters, which were far more distinct. The second score (PC2) is homogeneously distributed within the PC1 distinction, adding no additional classification information. In Figure 8, the PC scores show no capacity to distinguish wines by quality. Both PC scores contain wines of all quality levels, and PC1 gives roughly the same distribution of values across all quality scores. Overall, the PC scores do a poor job of distinguishing quality and a good job of distinguishing color. However, the clusters do a far better job of of distinguishing color and a slightly better job for quality. Therefore, the clustering technique is superior for this analysis.

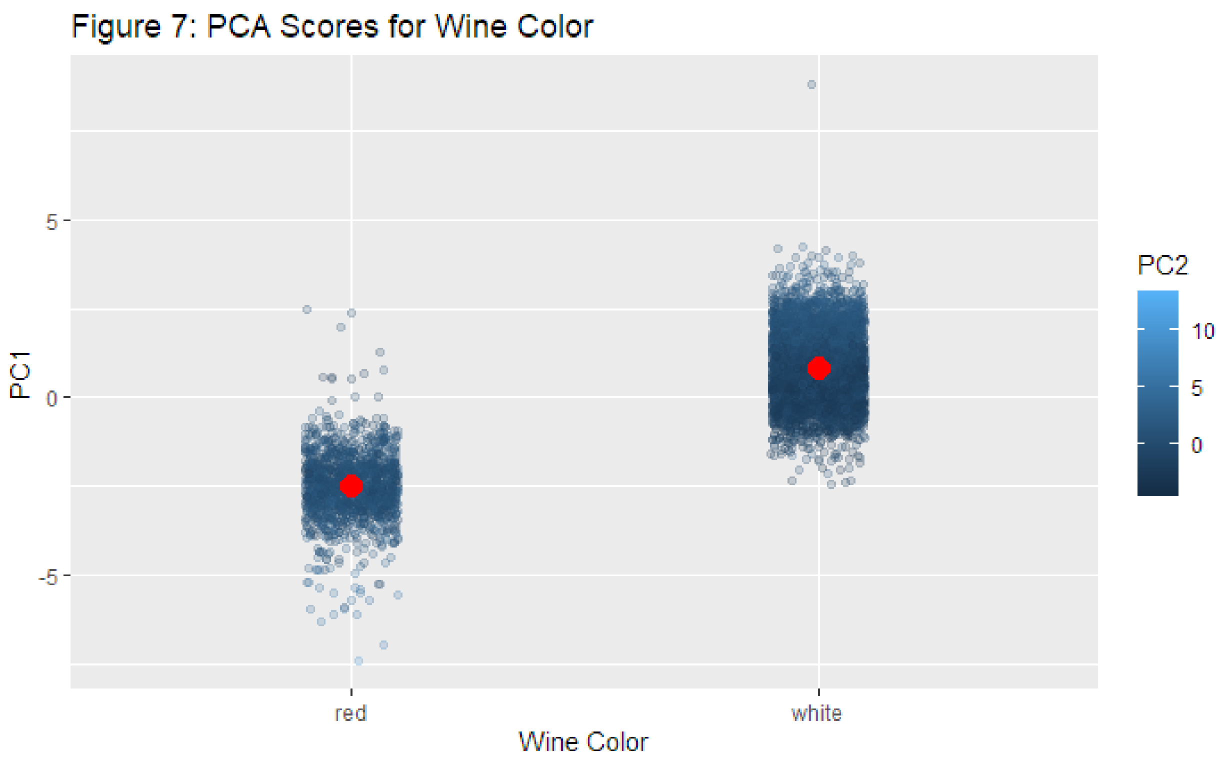


Figure 7: PCA Scores for Wine Color
PC1
5
0
-5
red
white
Wine Color
PC2
10
5
0

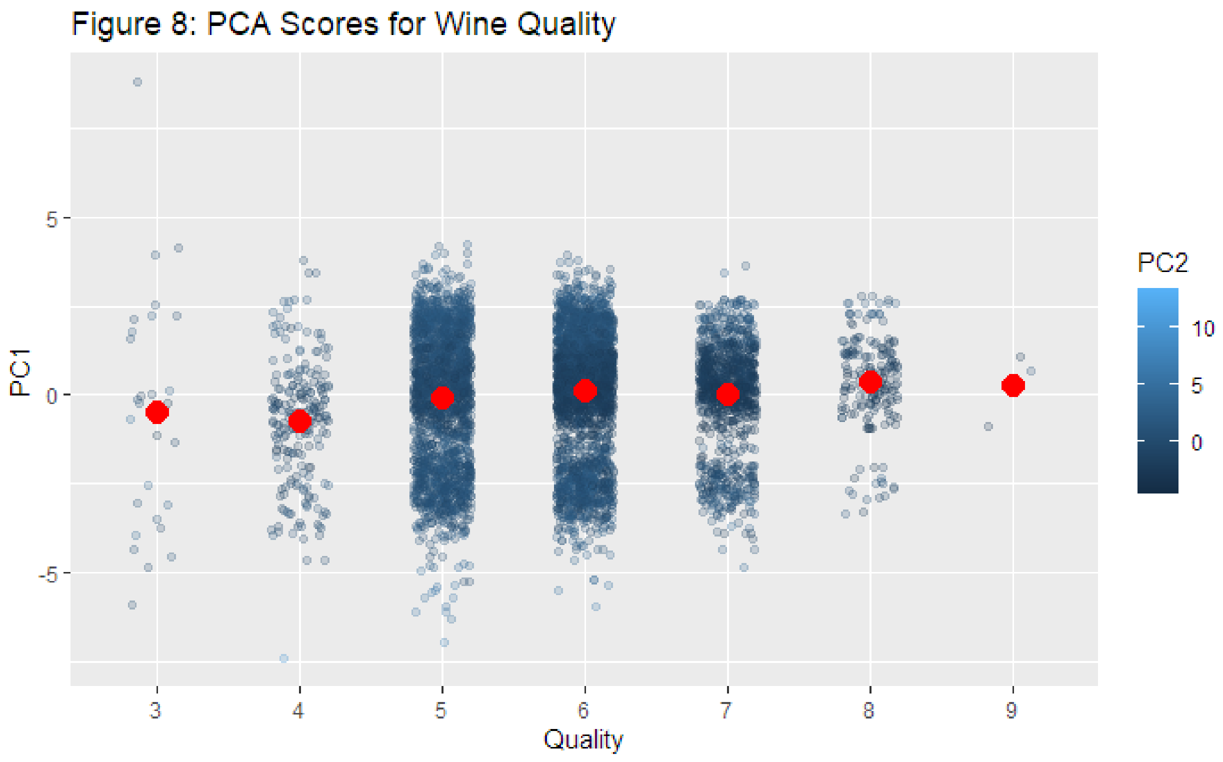


Figure 8: PCA Scores for Wine Quality
PC1
5
0
-5
3
4
5
6
7
8
9
Quality
PC2
10
5
0

Problem 2: Market Segmentation

Pre-Processing

I chose to use k-means++ clustering to identify market segments from the users data. I used an elbow plot to first choose the optimal number of clusters. The elbow plot below demonstrates the trade-off between within-distance (cluster tightness) and complexity. As you increase K, you get improvements in cluster within-distance, which is desirable. However, these improvements are *decreasing*, and as you increase K, you incur a cost of more complexity. If you have too may clusters (very high K), you may be over-fitting by dividing your data into unnecessary categories. The goal is to minimize withing-distance without over-fitting with too many clusters. The optimal K, therefore, is at the *bend* in the elbow, where you are getting decreasing improvements in within-distance while maintaining a manageable (non-excessive) K.

The optimal cluster count is roughly 7 clusters with a within-cluster sum of squares of about 200,000. However, after plotting the clusters by markets, it appears that a slightly lower cluster amount of 5 is more useful.

Plotting the clusters by related market pairs (e.g., computers and online_gaming) reveals how the five clusters separate individuals into usable market segments. At five cluster, the within-cluster distance is slightly larger, at 215,000, but the clusters are more informative to the segmentation.

Report for NutrientH20

After grouping each user into groups based on their expressions of interest in different markets, I have developed a set of five market segments. While there is overlap in individuals' interest across many markets, the segments I have produced can provide some distinct groupings of users interests. To optimize your marketing, you can use these groupings to target distinct market segments. Each of the 5 market segment group I have identified can be referred to as a cluster. Below are some visualizations to understand these groups.

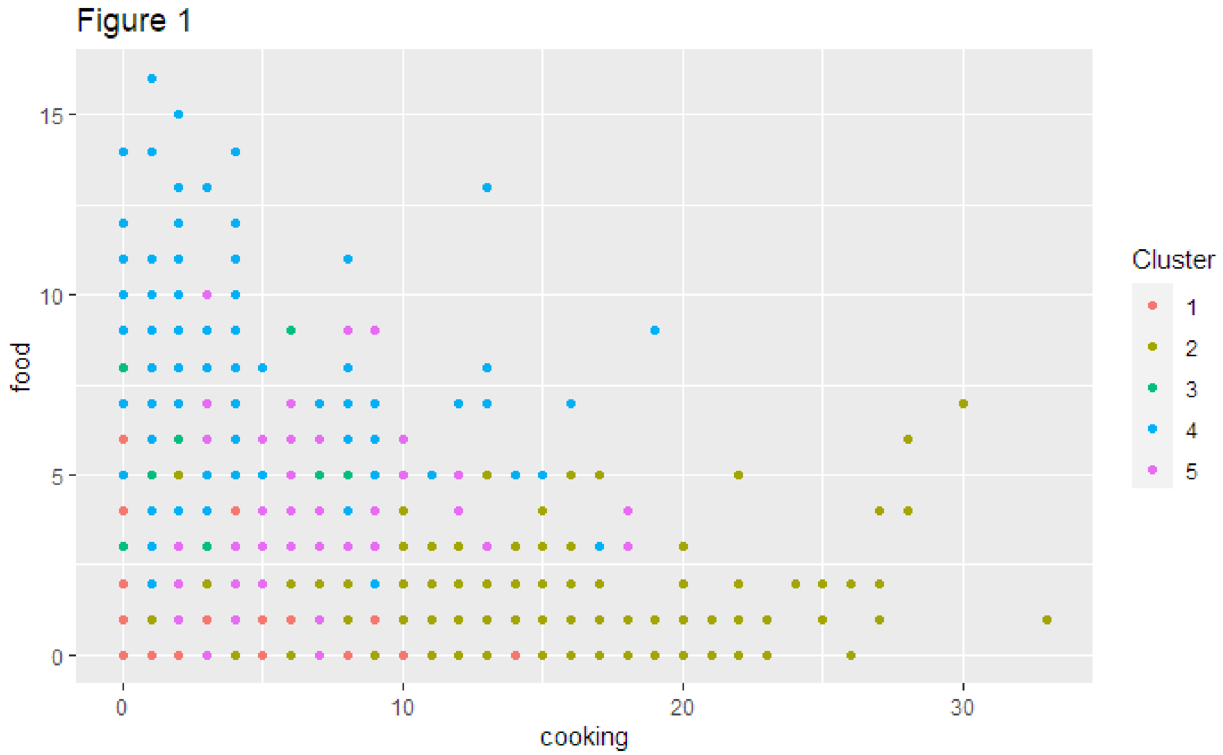


Figure 2

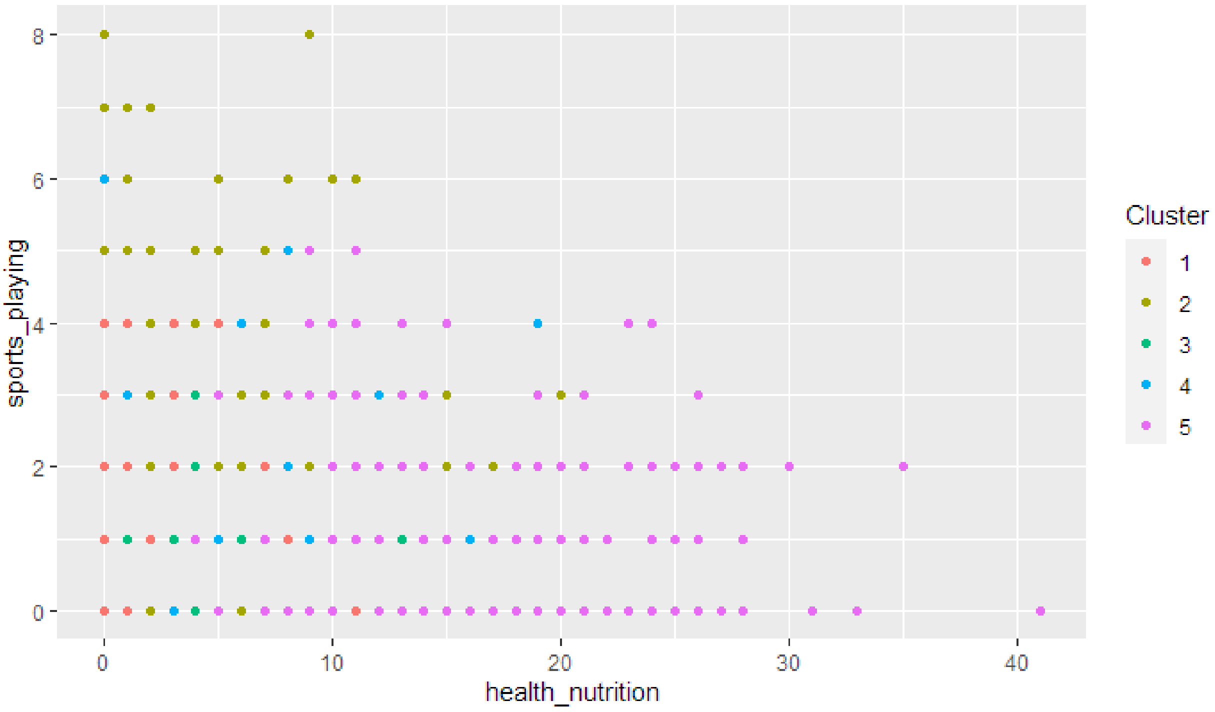


Figure 3

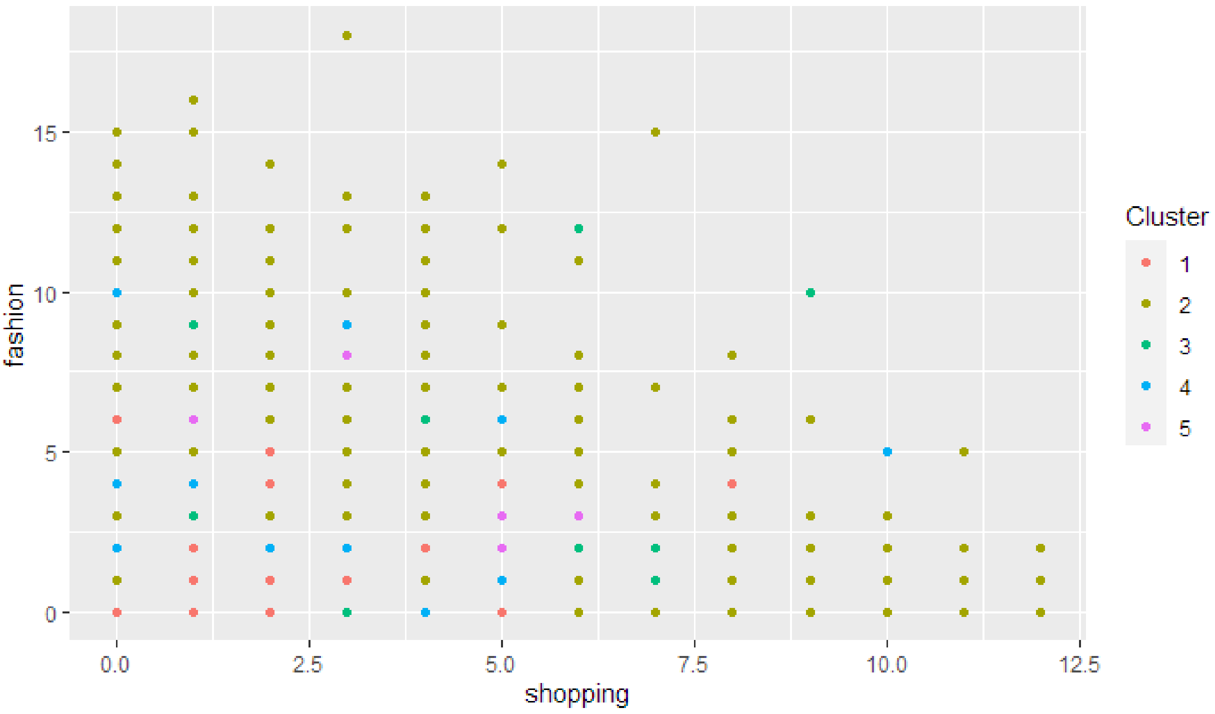


Figure 4

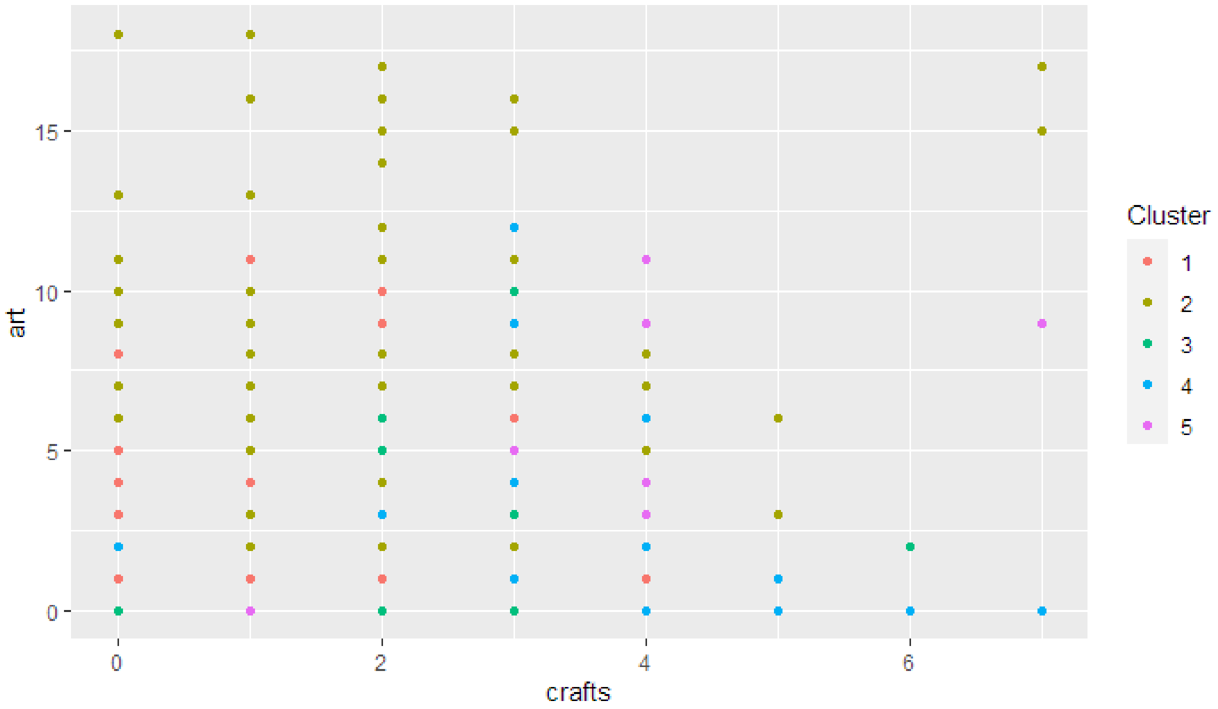


Figure 5

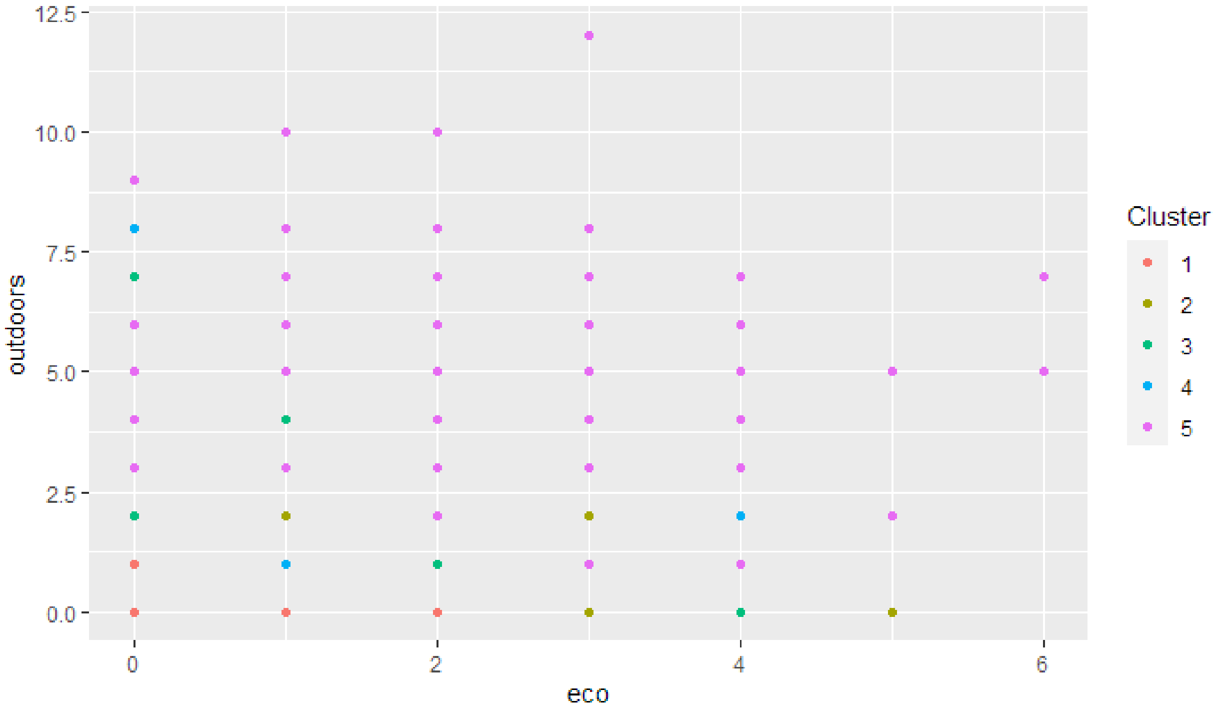


Figure 6

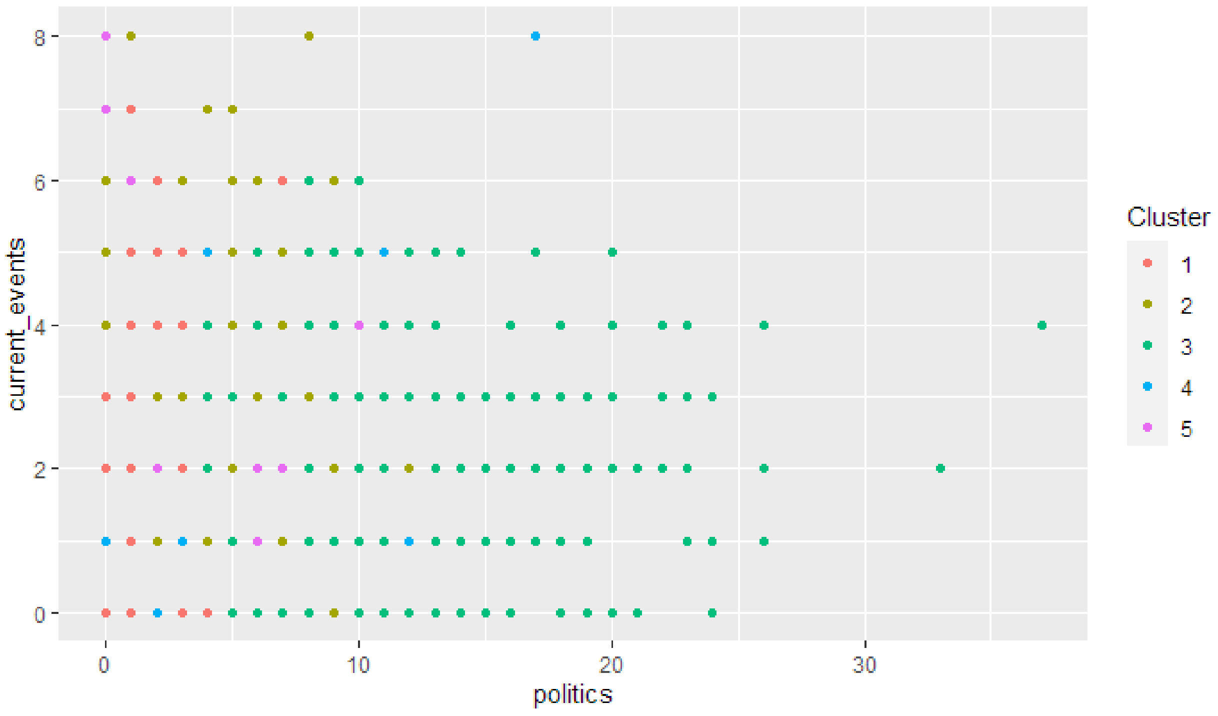


Figure 7

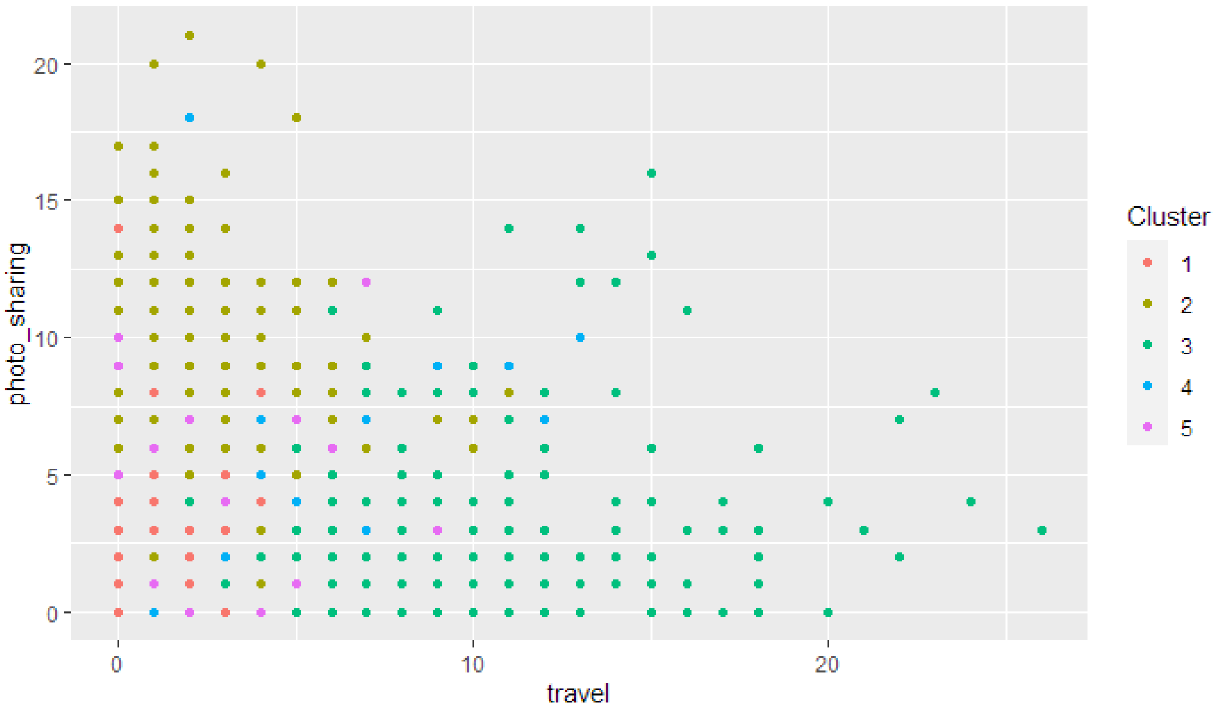
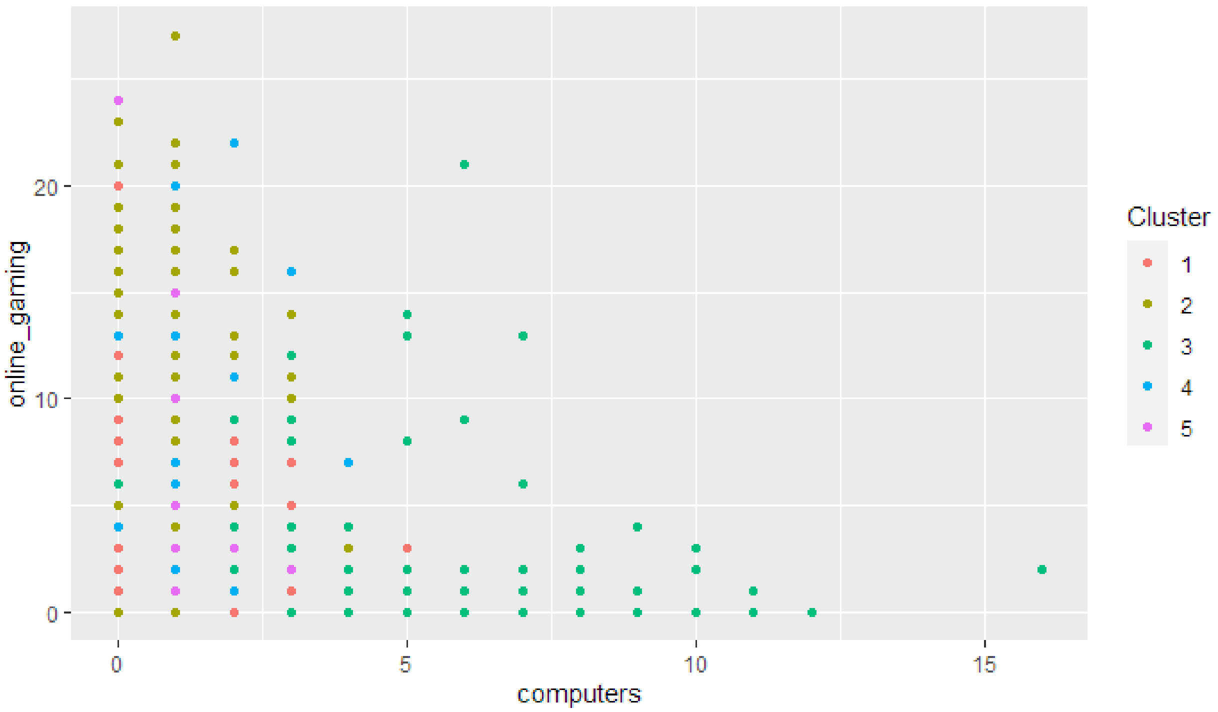


Figure 8



Cluster 1 contains the miscellaneous users how have no distinct interests in any particular market. In each of the figures below, cluster-1 individuals are sparsely represented in each market and show no particular trend, except a generally low interest in politics. To optimize your marketing, I would recommend *not* targeting these individuals because they represent an ambiguous market segment and are unlikely to respond to targeting. This frees the advertising budget for well-defined segments.

Cluster 2 has a high interest in cooking, fashion, and shopping (Figures 1 and 3). WHile NutrientH20 does not currently have plans to associate its brand with the fashion industry, individuals interested in fashion are often susceptible to *lifestyle* marketing where a product promotes a certain type of lifestyle. In this case, NutrientH20 could associate its drink products with the lifestyles of fashionable young adults. For instance, an advertisement with Jaden Smith or Kiley Jenner could be a good way to target this group. These individuals are already interested in fashion, and by associating NutrientH20s product with this industry through a celebrity opinion leader, NutrientH20 could make its products desirable to this group.

Cluster 3 has a very strong interest in politics, current events, and travel (Figures 6 and 7). NutrientH20 could target these users by promoting its products in the users' Twitter feed alongside news organization accounts, political pundit accounts, and travel accounts like National Geographic. THese individuals likely follow such accounts to stay tuned-in to politics,l news, and travel. By advertising next to these accounts, NutrientH20 can successfully target these users to increase sales.

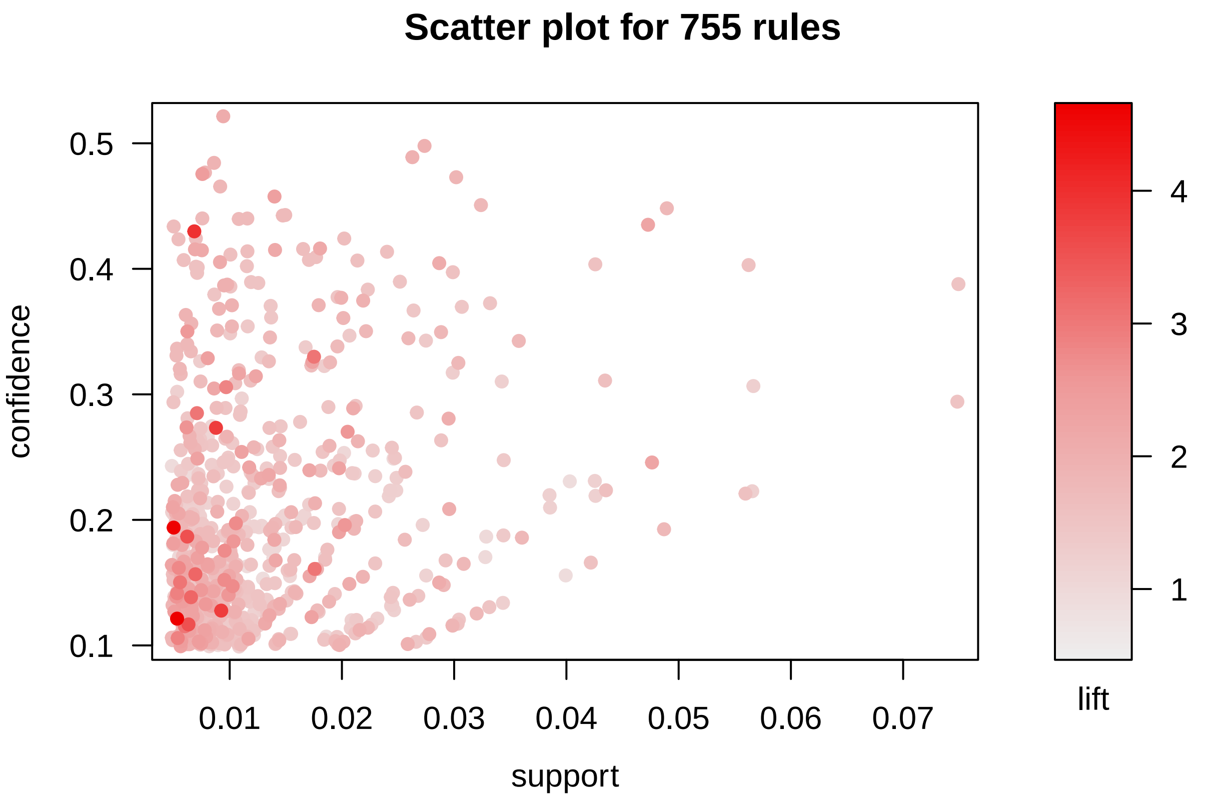
Cluster 4 has a a very high interest in food but a low interest in cooking (Figure 1). Since these individuals are likely to eat out at restaurants rather than cook at home, NutrientH20 could advertise to these individuals that their beverage products are available at trendy restaurants, like Cava and Panera.

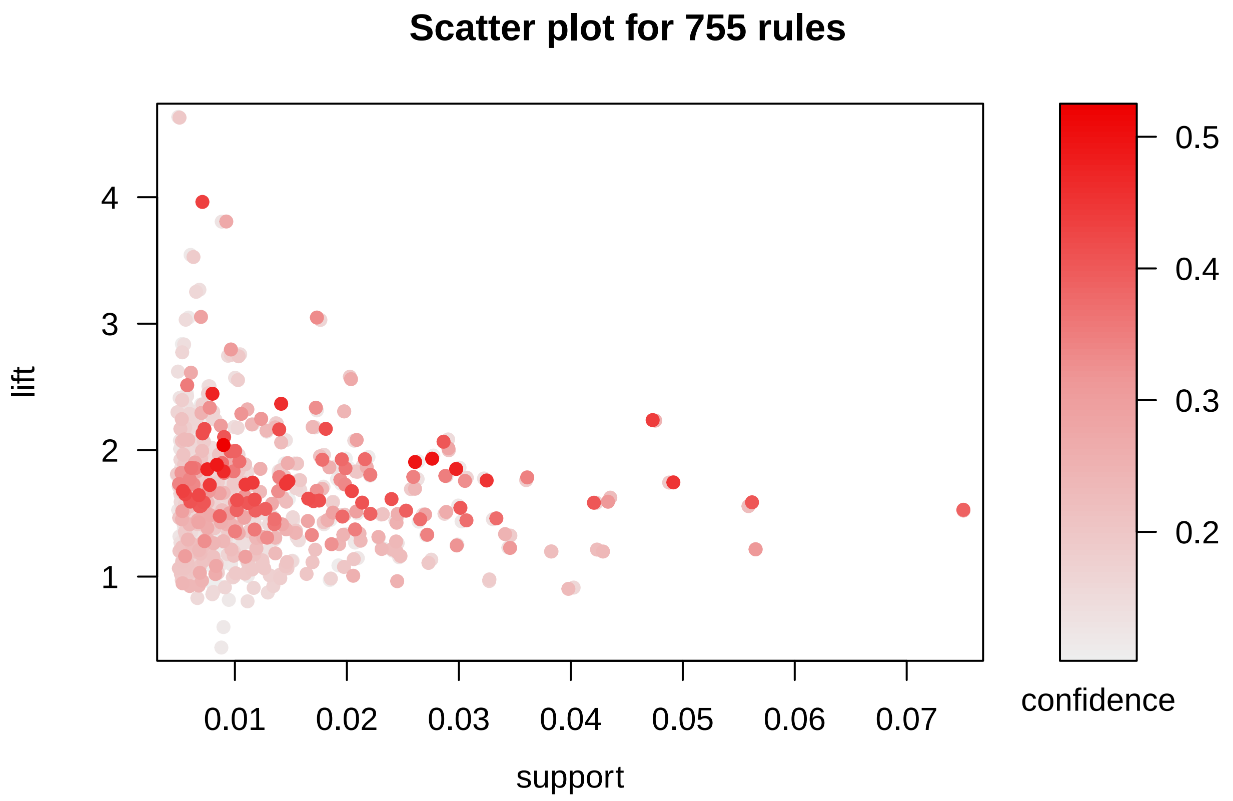
Cluster 5 has a high interest in health nutrition, the outdoors, and eco-friendly behavior (Figures 2 and 5). These individuals represent are a highly desirable customer segment because their interests are already aligned closely with NutrientH20's lifestyle brand image. NutrientH20 sells healthy, sporty beverages that provide carbohydrates and electrolytes to fit, adventurous, and youthful people. NutrientH20 should devote a substantial portion of its marketing budget to targeting these individuals by advertising how NutrientH20 enables a healthy and nutritious lifestyle by making fresh, eco-friendly, hydrating products for its adventurous customers.

Problem 3: Association Rules for Grocery Purchases

After removing rules for carts with either no antecedent or no consequent (blank LHS or RHS), there are 755 usable *rules* from the shopping cart data. Each rule represents an item bought in association with another other item and shows the descriptive statistics on how strong the association is (lift & confidence). Only 39 rules have lift above 3.5, meaning that the confidence of the consequent item in relation to its antecedent would be 3.5 times higher than its support. Only 9 rules have confidence above 45%, meaning that the consequent is purchased 45% of the time if the antecedent is purchased. There are no rules that have both a lift above 3.5 and confidence above 45%.

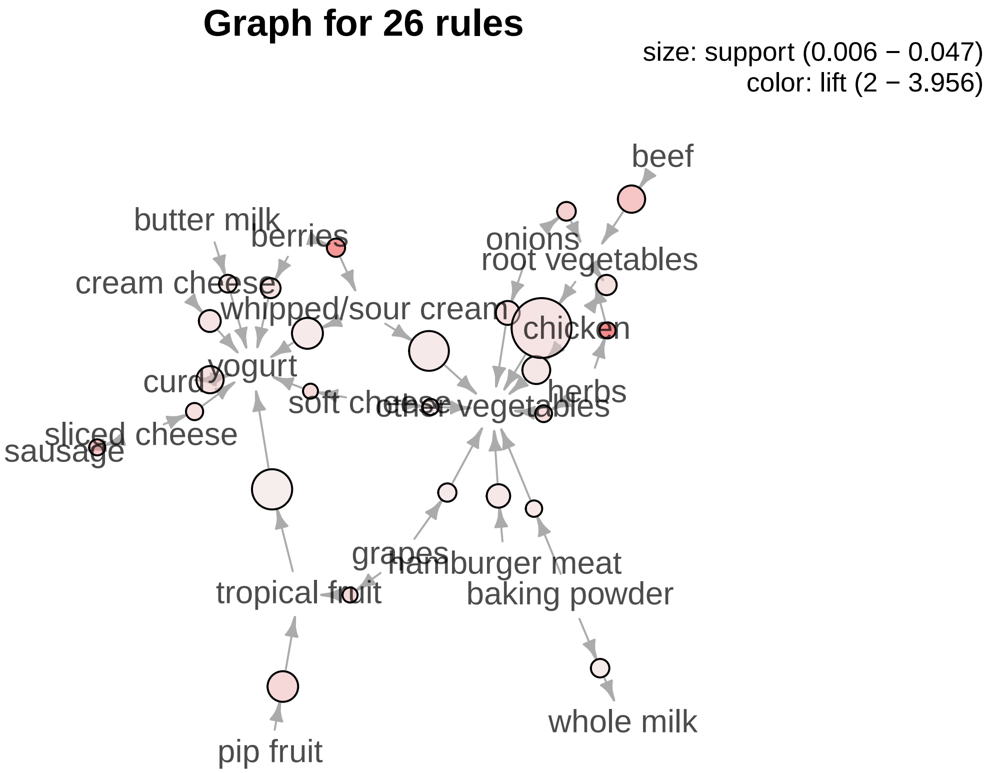
Therefore, to set thresholds for lift and confidence, it is necessary to visualize these values for all 755 rules. Plots for confidence vs support and for support vs lift are shown below.





To isolate the product associations that the grocery store could promote to maximize revenue, I will set the threshold for confidence at 25% to capture strong association purchases and lift at 2 to capture the subset with confidence at least double the consequent's support. I set the confidence threshold at 25% because the confidence vs support plot above that most rules fall below this value, and there are a smattering of high-lift rules above this level. The goal is to capture a subset that is not too large to focus on to drive associative sales. I chose the lift threshold of 2 because the lift vs support plot shows a small grouping of rules above a lift of 2 that appear to be valuable high-lift outliers that also fall above the confidence threshold. These two thresholds leave a subset of 26 rules with very strong purchasing associations through which the store could bolster marketing and thus sales. For instance, seeing that sausage has a strong rule with sliced cheese may be unexpected. But knowing this, the store can now place these items closer together to promote greater lift since there is already high association. Below are the 26 rules in the subset that the store should focus on. Also shown is a key visualization fo the network of item associations.

```
##      lhs                   rhs                  support     confidence
## [1]  {herbs}            => {root vegetables}    0.007015760 0.4312500
## [2]  {herbs}            => {other vegetables}   0.007727504 0.4750000
## [3]  {baking powder}    => {other vegetables}   0.007320793 0.4137931
## [4]  {baking powder}    => {whole milk}         0.009252669 0.5229885
## [5]  {soft cheese}      => {yogurt}             0.005998983 0.3511905
## [6]  {soft cheese}      => {other vegetables}   0.007117438 0.4166667
## [7]  {grapes}           => {tropical fruit}     0.006100661 0.2727273
## [8]  {grapes}           => {other vegetables}   0.009049314 0.4045455
## [9]  {butter milk}      => {yogurt}             0.008540925 0.3054545
## [10] {sliced cheese}    => {sausage}            0.007015760 0.2863071
## [11] {sliced cheese}    => {yogurt}             0.008032537 0.3278008
## [12] {onions}           => {root vegetables}    0.009456024 0.3049180
## [13] {onions}           => {other vegetables}   0.014234875 0.4590164
## [14] {berries}            => {whipped/sour cream} 0.009049314 0.2721713
## [15] {berries}            => {yogurt}             0.010574479 0.3180428
## [16] {hamburger meat}     => {other vegetables}   0.013828165 0.4159021
## [17] {cream cheese}       => {yogurt}             0.012404677 0.3128205
## [18] {chicken}            => {root vegetables}    0.010879512 0.2535545
## [19] {chicken}            => {other vegetables}   0.017895272 0.4170616
## [20] {beef}               => {root vegetables}    0.017386884 0.3313953
## [21] {curd}               => {yogurt}             0.017285206 0.3244275
## [22] {whipped/sour cream} => {yogurt}             0.020742247 0.2893617
## [23] {whipped/sour cream} => {other vegetables}   0.028876462 0.4028369
## [24] {pip fruit}          => {tropical fruit}     0.020437214 0.2701613
## [25] {tropical fruit}     => {yogurt}             0.029283172 0.2790698
## [26] {root vegetables}    => {other vegetables}   0.047381800 0.4347015
##      coverage   lift     count
## [1]  0.01626843 3.956477  69
## [2]  0.01626843 2.454874  76
## [3]  0.01769192 2.138547  72
## [4]  0.01769192 2.046793  91
## [5]  0.01708185 2.517462  59
## [6]  0.01708185 2.153398  70
## [7]  0.02236909 2.599101  60
## [8]  0.02236909 2.090754  89
## [9]  0.02796136 2.189610  84
## [10] 0.02450432 3.047435  69
## [11] 0.02450432 2.349797  79
## [12] 0.03101169 2.797452  93
## [13] 0.03101169 2.372268 140
## [14] 0.03324860 3.796886  89
## [15] 0.03324860 2.279848 104
## [16] 0.03324860 2.149447 136
## [17] 0.03965430 2.242412 122
## [18] 0.04290798 2.326221 107
## [19] 0.04290798 2.155439 176
## [20] 0.05246568 3.040367 171
## [21] 0.05327911 2.325615 170
## [22] 0.07168277 2.074251 204
## [23] 0.07168277 2.081924 284
## [24] 0.07564820 2.574648 201
## [25] 0.10493137 2.000475 288
## [26] 0.10899847 2.246605 466
```



The network plot is highly informative of the associative relationships of these high-lift, high-confidence rules. The node color saturation denotes lift, and the node size denotes support. The arrows denote antecedent to consequent relationships. (Note that the locations of sub-regions may not appear as described here due to knitting. The network will be the same, but its orientation on the XY plane may be changed.)The top-left region shows yogurt at the center of sub-network. Yogurt is a consequent to buttermilk, cream cheese, soft cheese, and sour cream. It appears to be a diary region, but it also contains berries and a connection to a small fruit region in the top-right. This implies there is marketing opportunity around yogurt including other dairy purchases and fruit purchases. Interestingly, there is a small offshoot from the sliced cheese node in the diary region that is connected to sausage, which has no other connections in this subset. In the lower-right there is a larger sub-region with vegetables at the center. Surrounding vegetables are various antecedents including poultry, meat, and other vegetables like onions and root vegetables. This region could be a dinner region for meat and vegetables with a few offshoots, like baking powder and grapes. Overall, this sub-region implies that there is marketing opportunity in exploiting product affinities surrounding typical dinner substrates (meat and vegetables). This may be especially profitable since the meat item (large circle in lower-left region) already has very high support, indicated by its circle size. By co-advertising its consequents in this region through point-of-purchase marketing, the store can exploit the fact that meat is a high-frequency purchase and persuade people to buy affiliated goods with high lift.

Probelm 4: Author Attribution

Separate training and testing corpuses were provided to predict authorship of documents. Both corpuses contain 2,500 documents (50 authors by 50 documents each). My first pre-processing step was to *tokenize* these corpuses. I used 5 tokenization processes: (1) transform all words to lower-case, (2) remove numbers from the text, (3) remove punctuation from the text, (4) remove/split-on blank spaces, and (5) remove *stop words* (i.e., filler words: is, a, of, been, etc.).

I then created separate train and test Document-Term Matrices (DTMs). Each DTM contained 2,500 documents (50 authors by 50 articles) by 31,423 terms. Inspecting the DTMs revealed that tokenization was successful in producing discrete, actual words (i.e., access, accounts, agencies, announced, bogus, business, etc).

Since most of these terms only occur in one document, the DTMs were highly sparse. Therefore, it was essential to keep only the terms that show up across multiple documents, as these will be terms useful for author attribution prediction. Therefore, I removed terms in the training DTM that showed up zero times in more than 99% of training corpus documents. This step reduced the amount of terms to just 3,076. Finally, to simplify prediction on the test set, I restricted the terms in the test set to only those 3,076 defined in the reduced training set.

I next standardized the observations in each DTM using a combination of two weighting techniques: Term-Frequency Weighting (TF) and Inverse-Document Frequency Weighting (IDF). With TF weighting, I accounted for the fact that some documents are longer than others and therefore have greater frequency of certain terms relative to shorter documents. This weighting reduces the impact of long documents' terms to be equivalent to that of shorter documents. Using IDF weighting, I down-weighted certain terms that show up frequently across all documents but that are not actually useful. For instance, since these documents are from corpus of news articles, there will be certain frequent terms that are common to news articles but that are not idiosyncratic to author and therefore should be removed to improve prediction and reduce noise. These weighting methods alter the value assigned to each token in each document in the DTMs and resulted in matrices of the same size as before (2,500 by 3,076) but with newly-weighted observations.

Since the weighted matrices still contained somewhat high sparsity (many observations of 0), I used Principal Components Analysis (PCA) to reduce the matrix dimensions to 100 principal components scores. All 2,500 documents were assigned 100 scores (PC1 to PC100) related to the composition of 3,076 terms in each document. This enables documents to be compared for commonality and difference by how close their scores are while reducing the matrix sparsities. Using these scores, it is possible to determine which documents are similar to one another and thereby help predict authorship of each document using a prediction model. The summary of the PC scores shown below demonstrates that roughly 24% of the variation in the 3,076 terms is cumulatively explained by the PC scores.

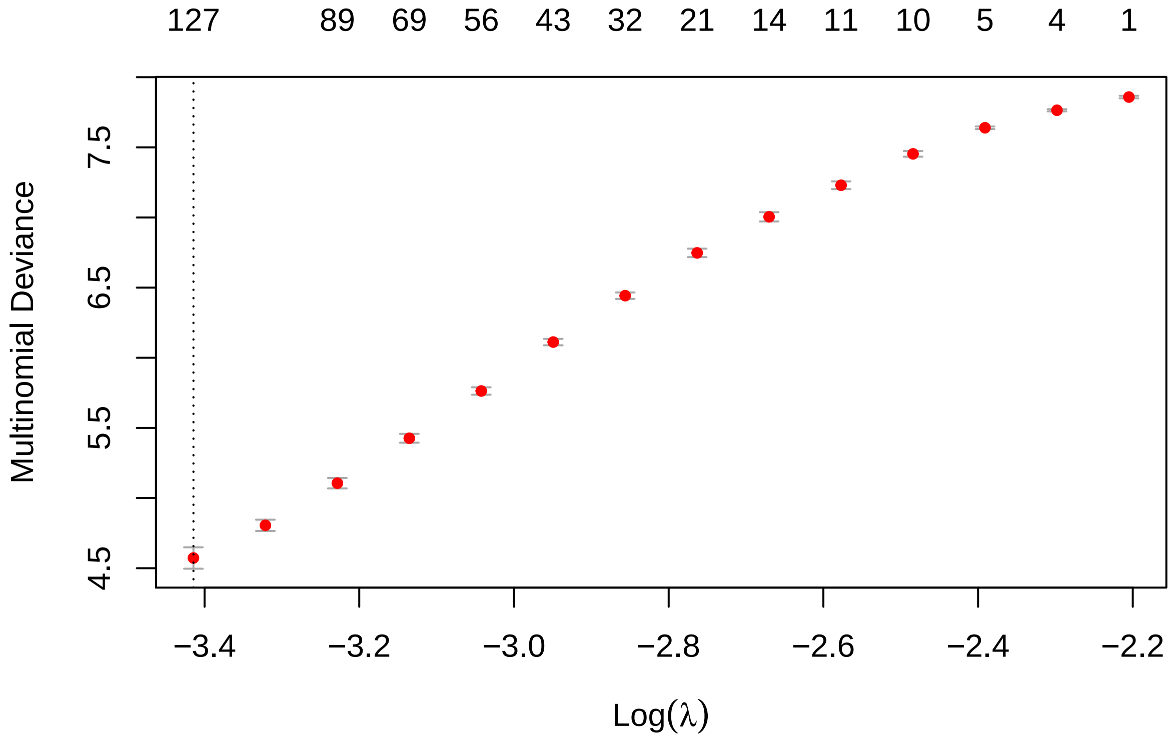
```
## Importance of first k=100 (out of 264) components:
##                            PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.80192 2.28319 2.02413 1.94914 1.85113 1.83527 1.72443
## Proportion of Variance 0.02974 0.01975 0.01552 0.01439 0.01298 0.01276 0.01126
## Cumulative Proportion  0.02974 0.04948 0.06500 0.07939 0.09237 0.10513 0.11640
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     1.68425 1.60689 1.57840 1.56807 1.53897 1.50958 1.48511
## Proportion of Variance 0.01075 0.00978 0.00944 0.00931 0.00897 0.00863 0.00835
## Cumulative Proportion  0.12714 0.13692 0.14636 0.15567 0.16464 0.17328 0.18163
##                          PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     1.4801 1.45954 1.43349 1.42900 1.41509 1.39467 1.37413
## Proportion of Variance 0.0083 0.00807 0.00778 0.00774 0.00759 0.00737 0.00715
## Cumulative Proportion  0.1899 0.19800 0.20578 0.21352 0.22110 0.22847 0.23562
##                           PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     1.36131 1.35469 1.34660 1.33406 1.32678 1.31783 1.31297
## Proportion of Variance 0.00702 0.00695 0.00687 0.00674 0.00667 0.00658 0.00653
## Cumulative Proportion  0.24264 0.24959 0.25646 0.26320 0.26987 0.27645 0.28298
##                           PC29    PC30    PC31    PC32   PC33    PC34    PC35
## Standard deviation     1.29830 1.29144 1.28498 1.28087 1.2789 1.26453 1.26252
## Proportion of Variance 0.00638 0.00632 0.00625 0.00621 0.0062 0.00606 0.00604
## Cumulative Proportion  0.28936 0.29568 0.30194 0.30815 0.3144 0.32040 0.32644
##                          PC36    PC37    PC38    PC39   PC40    PC41    PC42
## Standard deviation     1.2582 1.25632 1.24909 1.23860 1.2378 1.23364 1.22307
## Proportion of Variance 0.0060 0.00598 0.00591 0.00581 0.0058 0.00576 0.00567
## Cumulative Proportion  0.3324 0.33842 0.34433 0.35014 0.3559 0.36170 0.36737
##                           PC43   PC44    PC45    PC46    PC47    PC48    PC49
## Standard deviation     1.21937 1.2154 1.21069 1.20391 1.19708 1.19250 1.19002
## Proportion of Variance 0.00563 0.0056 0.00555 0.00549 0.00543 0.00539 0.00536
## Cumulative Proportion  0.37300 0.3786 0.38415 0.38964 0.39507 0.40046 0.40582
##                           PC50    PC51    PC52    PC53    PC54    PC55    PC56
## Standard deviation     1.18544 1.17798 1.16934 1.16653 1.16374 1.15858 1.15444
## Proportion of Variance 0.00532 0.00526 0.00518 0.00515 0.00513 0.00508 0.00505
## Cumulative Proportion  0.41114 0.41640 0.42158 0.42673 0.43186 0.43695 0.44200
##                           PC57    PC58   PC59    PC60    PC61    PC62    PC63
## Standard deviation     1.14667 1.14156 1.1373 1.13553 1.13237 1.12450 1.12379
## Proportion of Variance 0.00498 0.00494 0.0049 0.00488 0.00486 0.00479 0.00478
## Cumulative Proportion  0.44698 0.45191 0.4568 0.46170 0.46655 0.47134 0.47613
##                           PC64    PC65   PC66    PC67    PC68    PC69    PC70
## Standard deviation     1.11770 1.11497 1.1145 1.11291 1.10669 1.09916 1.09684
## Proportion of Variance 0.00473 0.00471 0.0047 0.00469 0.00464 0.00458 0.00456
## Cumulative Proportion  0.48086 0.48557 0.4903 0.49496 0.49960 0.50418 0.50874
##                          PC71    PC72    PC73    PC74   PC75    PC76    PC77
## Standard deviation     1.0901 1.08823 1.08565 1.07992 1.0772 1.07187 1.06931
## Proportion of Variance 0.0045 0.00449 0.00446 0.00442 0.0044 0.00435 0.00433
## Cumulative Proportion  0.5132 0.51772 0.52219 0.52661 0.5310 0.53535 0.53968
##                           PC78    PC79    PC80    PC81   PC82    PC83    PC84
## Standard deviation     1.06470 1.06267 1.06052 1.05686 1.0525 1.04651 1.04252
## Proportion of Variance 0.00429 0.00428 0.00426 0.00423 0.0042 0.00415 0.00412
## Cumulative Proportion  0.54398 0.54826 0.55252 0.55675 0.5609 0.56509 0.56921
##                           PC85    PC86    PC87   PC88    PC89    PC90    PC91
## Standard deviation     1.04122 1.03661 1.03060 1.0276 1.02387 1.02231 1.02146
## Proportion of Variance 0.00411 0.00407 0.00402 0.0040 0.00397 0.00396 0.00395
## Cumulative Proportion  0.57331 0.57738 0.58141 0.5854 0.58938 0.59334 0.59729
##                           PC92    PC93    PC94    PC95    PC96    PC97    PC98
## Standard deviation     1.01372 1.01145 1.00826 1.00486 0.99747 0.99611 0.99503
## Proportion of Variance 0.00389 0.00388 0.00385 0.00382 0.00377 0.00376 0.00375
## Cumulative Proportion  0.60118 0.60506 0.60891 0.61273 0.61650 0.62026 0.62401
##                           PC99   PC100
## Standard deviation     0.99280 0.98983
## Proportion of Variance 0.00373 0.00371
## Cumulative Proportion  0.62774 0.63145
```

Multinomial Logit Model

I created two models, the first of which was a Multinomial Logit Model where each author (outcome) is read as a factor, and each of 100 PC scores is a predictor variable. I combined the outcome matrix of 2500 authors with the DTM training set matrix of 100 PC scores so that the model could read the covariates and outcome from the same data frame. The coefficients of the multinomial logit model are shown below.

Multinomial Lasso Model

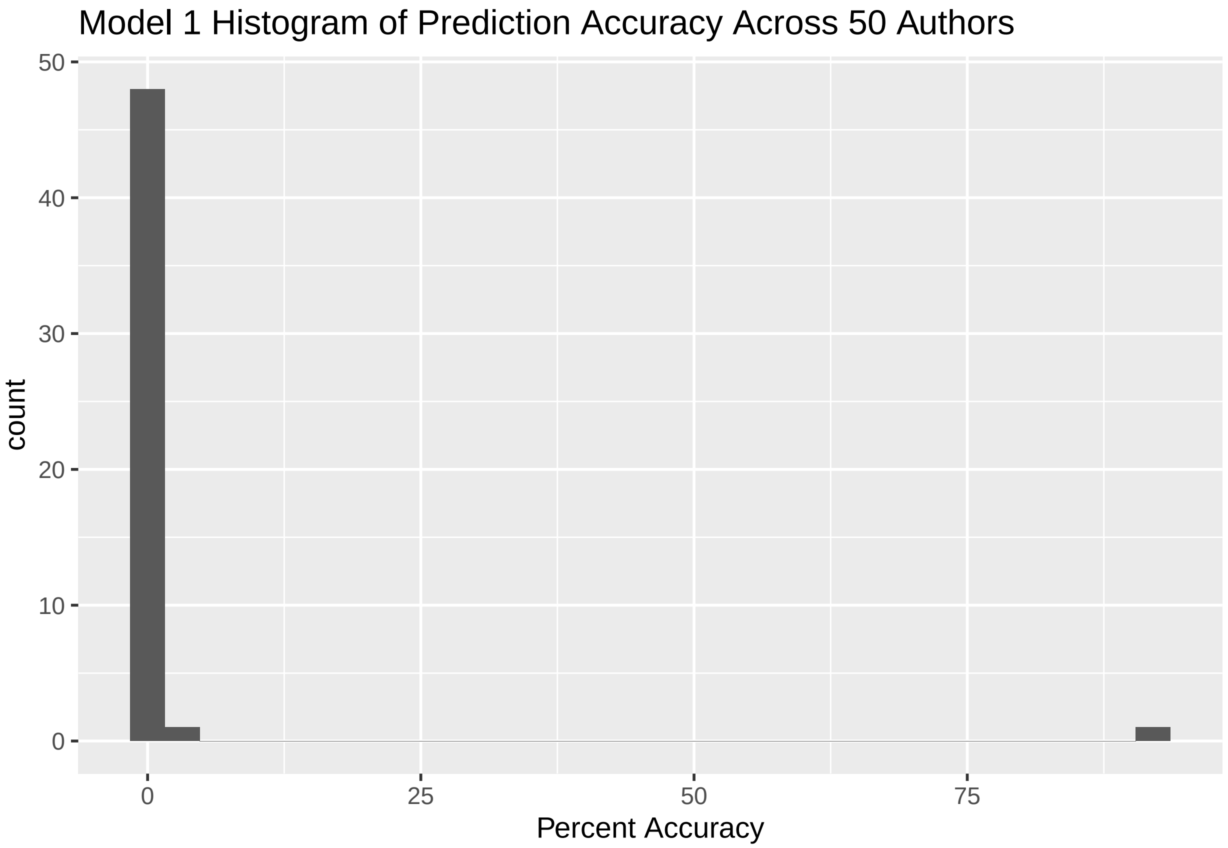
The second model is a cross-validated Multinomial Lasso Model. Instead of using all 100 PC scores like in the first model, Instead of using weighted, PC-scored covariates, I used the raw terms from the training DTM dictionary to create the model, and predicted on the raw terms of the test DTM. The reason is explained below after the plot of the lasso model just underneath here:

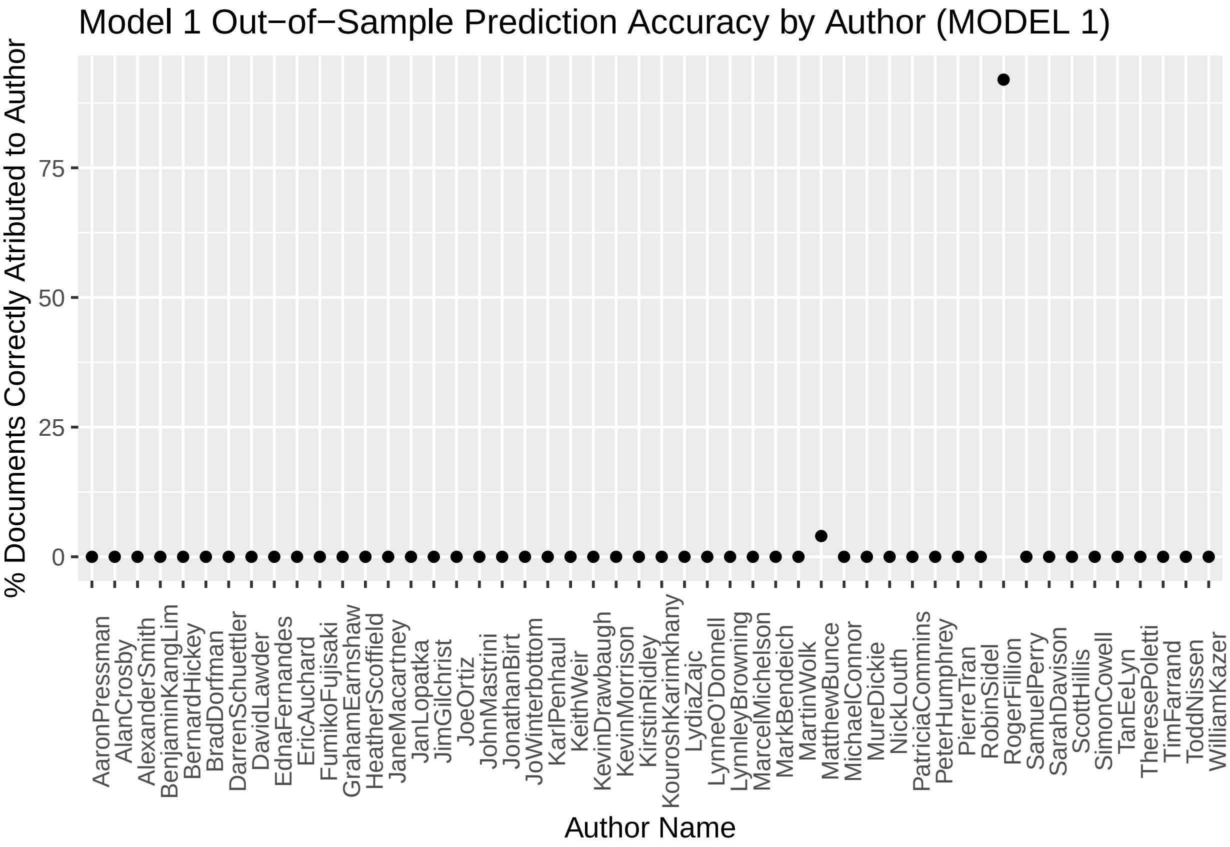


Before I show the out-of-sample prediction accuracy results, I should mention that I had an issue with restricting the vocabulary of my test DTM to include the same terms as my training DTM. WHen you tokenize the training corpus and testing corpus, you get different *vocabularies* or *dictionaries* of terms. In order for prediction to be effective, it is important that you restrict the test set vocabulary to only that of he training set, so that you are using the same covariates (words) to predict outcomes on the test set as are available in the test set. This was not the case here. My restriction code (DTM_test = DocumentTermMatrix(corpus_test, control = list(dictionary=Terms(DTM_train)))) only restricted my testing DTM to have the same *amount* of terms, not the same terms as the training DTM. Therefore, my prediction accuracy is extremely low since the training and testing matrices are dissimilar. This is also why the PCA analysis did not help–it produced PC scores based on totally different vocabularies. In fact, any correct predictions are likely due more to chance. There is nothing wrong with the model, but rather the inputs. And this proved to be an unassailable problem. I scoured the internet and reached out for help to figure out why this code was not working, but as far as I can tell, it is the right line of code, but it just does not do what it is meant to do. So, please take into the account that I can identify the main reason my prediction accuracy is so low, have tried to fix it, and am unable to despite exhausting every resource.

Model 1 Results

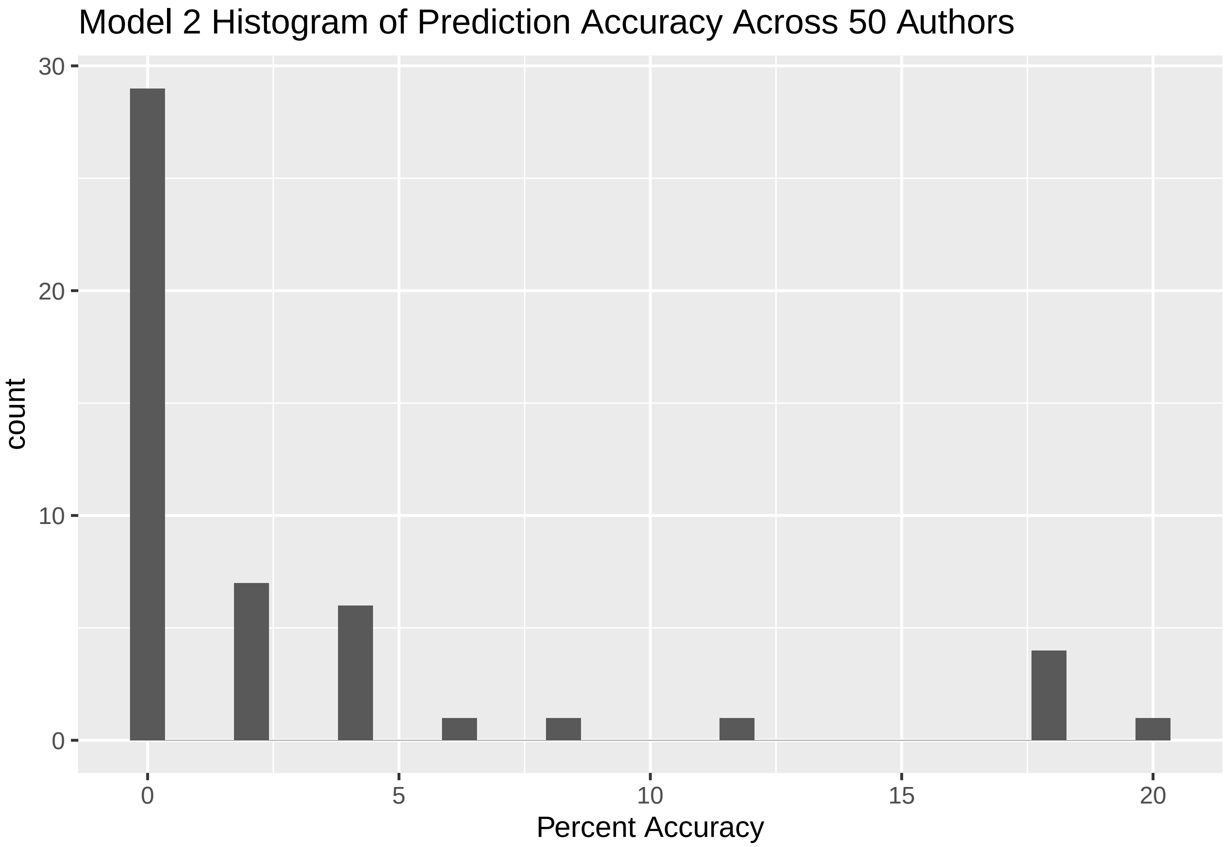
The prediction accuracy for the first model (multinomial with 100 PC scores) is about 1.92%. This means that out of all 2,500 documents whose authorships I predicted in the test set, only 48 were correct. Below are a histogram and scatterplot of prediction accuracies across all authors. Since each author had 50 documents, there were 50 opportunities to correctly predict authorship for each author. The histogram shows that most predictions overall were wrong (0), and the scatterplot shows that i predicted authorship of at least one document correctly for only two authors (Michael Connor and Roger Fillion). The model did an excellent job of assigning correct authorship to Roger Fillion, and a dismal job for all 49 other authors. Again, this failure is attributed to the different dictionaries of the training and testing sets, which could not be rectified. If the training and testing sets had the same set of terms, these predictions would be much improved. So, I suffer a technical limitation, not a methodological failure.

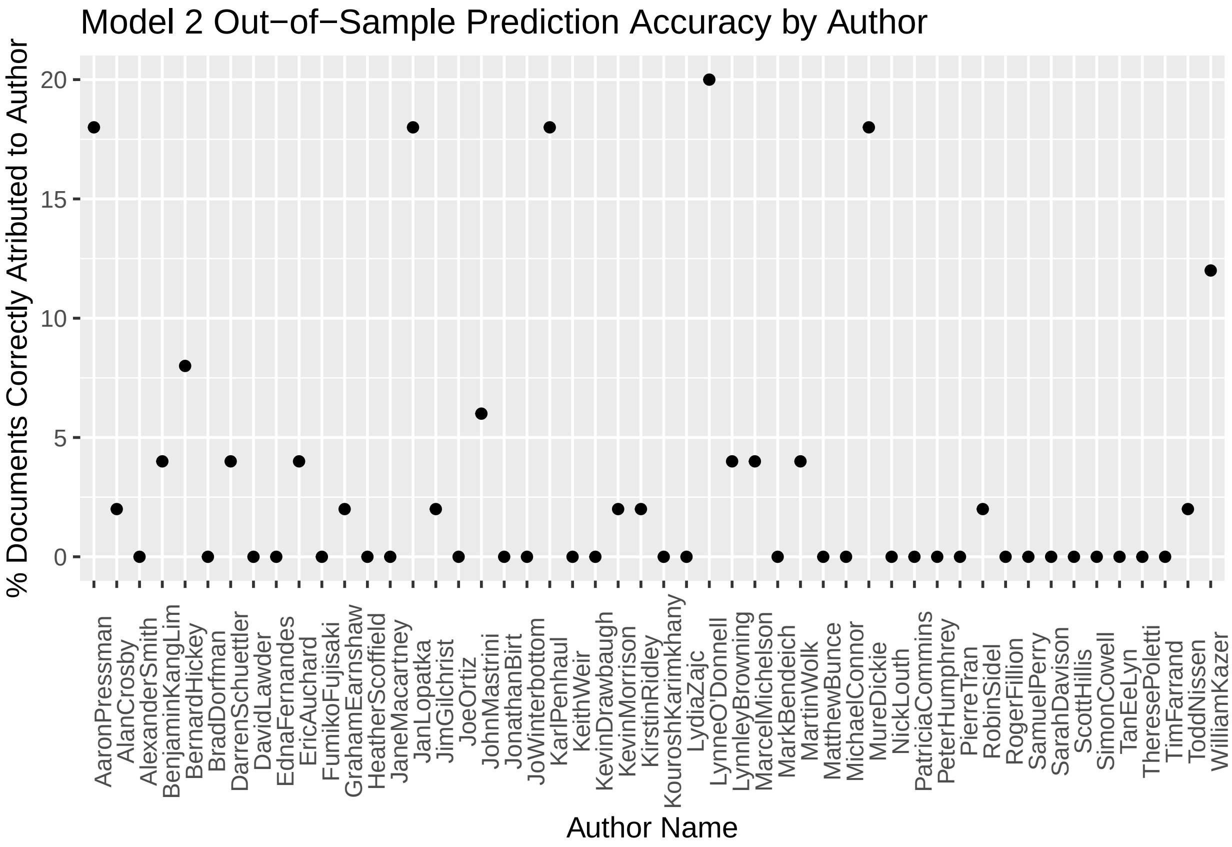




Model 2 Results

The prediction accuracy for the second model (lasso multinomial) is much better than model 1. This is because I did not use PC scores to summarize the terms. By leaving the raw terms in both the training and testing sets for prediction, I was able to capture information from the subset of terms that actually overlap between the training and testing set. Again, this relates back to my core problem of different dictionaries. If I could have a common dictionary across both the training and testing sets, then the PCA dimensionality reduction may have actually been useful. Insofar as I am stuck with different dictionaries due to technical issue, the happenstance overlap of certain terms in the raw dictionaries is the best means of predicting authorship. Below, you can see that using the raw terms as predictor variables enabled far more correct predictions (though still very little overall), resulting in an average authorship prediction accuracy of 3.12%–nearly 3 times greater that with PC scores. The scatterplot shows that instead of just 2 authors with some correct predictions, this model correctly predicts at least one document for 21 authors. The remaining 29 authors did not have a single correct document assigned during prediction. The ability of this model to capture some predictive information despite totally different dictionaries implies that there are likely some high-frequency terms that were present in both dictionaries. The rest of the terms in the testing set, because they were not restricted to the training set dictionary, simply were not usable, and the terms in the training set that were not in the testing set created immense predictive noise. If dictionary restriction could have been accomplished, this noise would have been removed, and the PC scores could have been useful.


Model 2 Histogram of Prediction Accuracy Across 50 Authors
count
30
20
10
0
0
5
10
15
20
Percent Accuracy



I should reiterate that I actually got as high as 11% accuracy when choosing two random words from the terms list in the DTM to use as predictor variables. However, since the DTM is so sparse, the model could not handle any more words, let alone the entire set (hundreds). So dimensionality reduction was required. I felt pressured to use a more complex methodology (dimensionality reduction) since the directions called for this rather than to go with the simpler model just described that produced better predictions. My main problem here is in restricting the test DTM to the dictionary of the training DTM, as discussed. I have tried many different versions of the statement that is supposed to do this (DocumentTermMatrix(corpus_train, control = list(dictionary=Terms(DTM_train)))), however it never works. Therefore, my training and testing DTMs have different terms. I am trying to use covariates from the training matrix that are not in the testing matrix to to out-of-sample prediction, so naturally, the prediction will be poor. Unless I can get this dictionary-restriction code to work (there is no clear reason why it should not), I will continue to get very poor prediction accuracy.