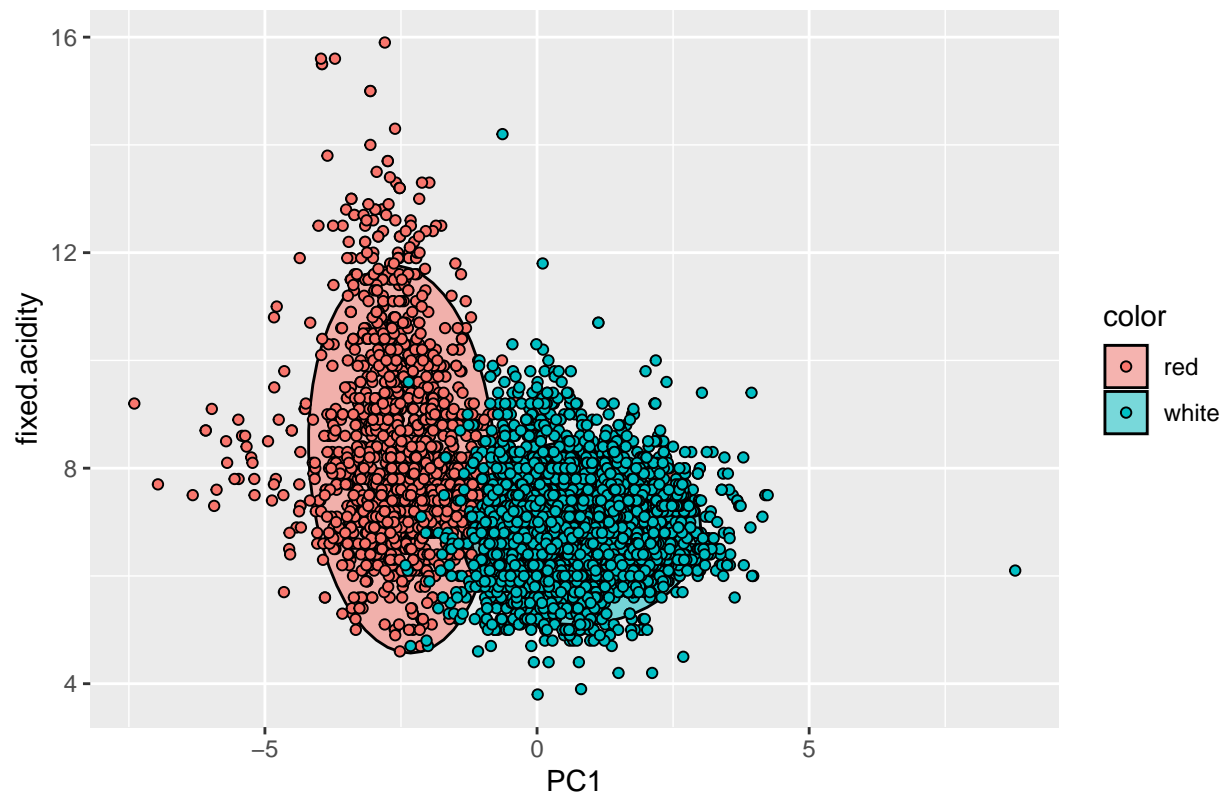


Exercise 4

Question 1: Clustering and PCA

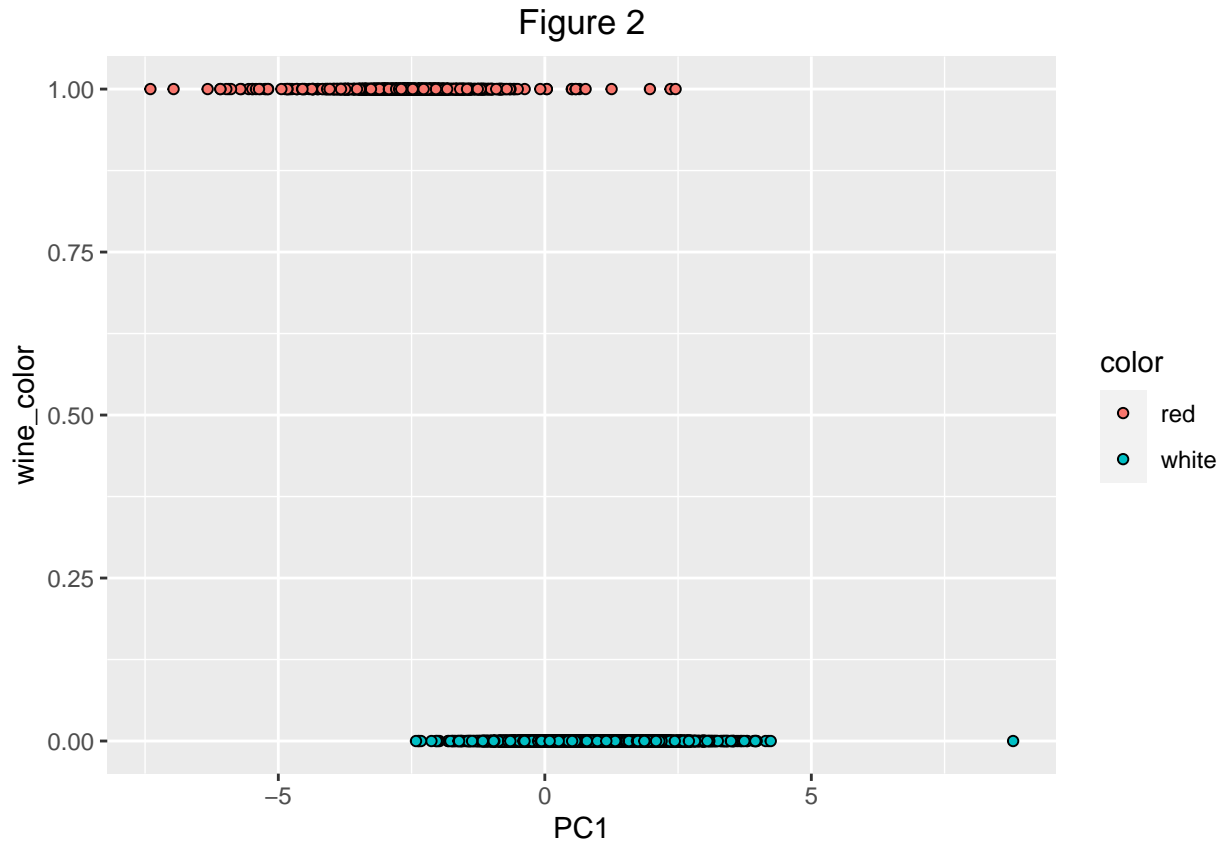
Principal Component Analysis (PCA)

Figure 1



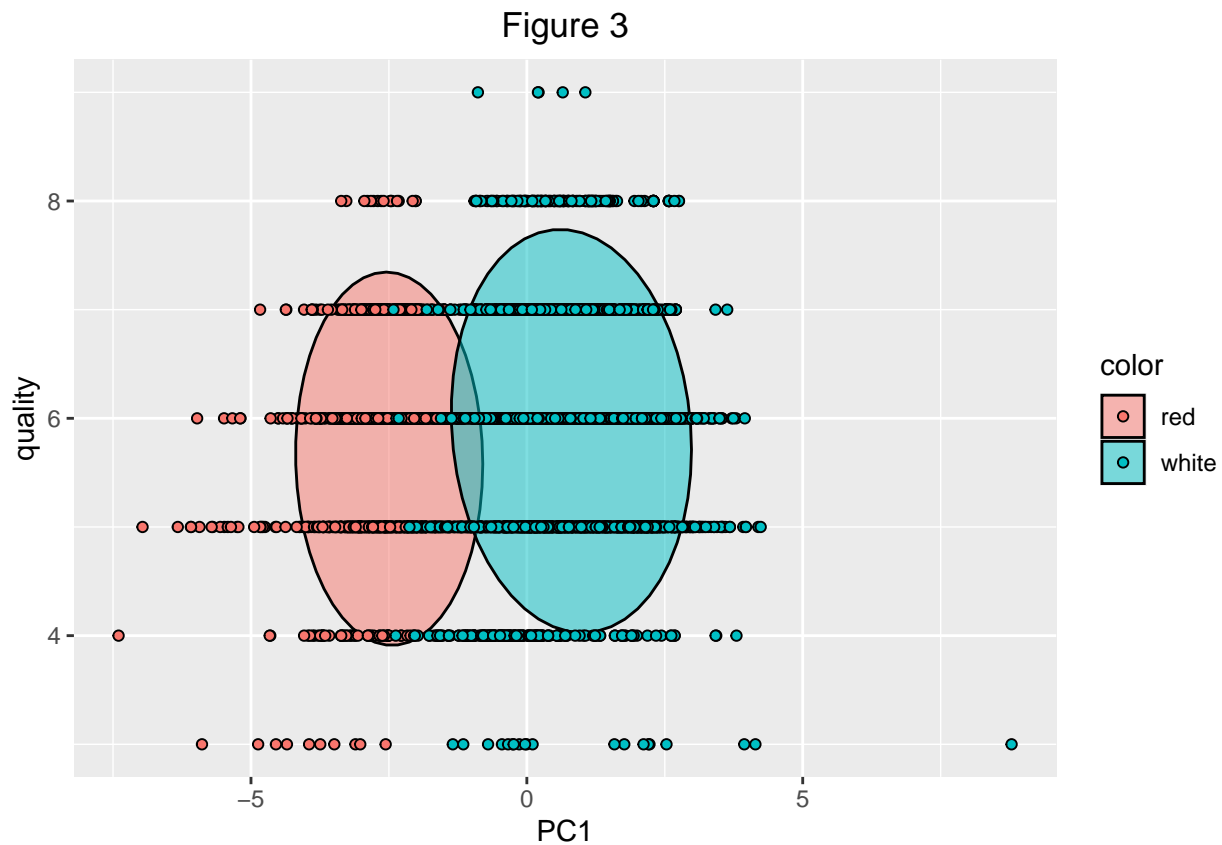
```
##                                [,1]
## fixed.acidity                 -0.41566573
## volatile.acidity              -0.66276622
## citric.acid                   0.26525521
## residual.sugar                0.60212615
## chlorides                     -0.50498499
## free.sulfur.dioxide           0.75007124
## total.sulfur.dioxide          0.84842512
## density                       -0.07821904
## pH                            -0.38065695
## sulphates                     -0.51198691
## alcohol                       -0.18526996
```

Using principal component analysis (PCA), I was able to use a data set consisting of 11 chemical properties of wine to determine whether the observed wine was categorized as red or white. Figure 1 displays the data collected for the chemical property, fixed acidity, and PC1, and determines which of the points are predicted to be white or red. Additionally, fixed acidity and PC1 have a moderate correlation of approximately -0.415. It should also be noted that PC1 is more or less correlated, depending on the chemical property. Total sulfur dioxide has the highest correlation of 0.848, while density has the lowest correlation of -0.078.



```
## [1] -0.8254209
```

Figure 2 displays PC1 and the determined wine_color, red or white. Overall, PCA did a good job at determining the color of wine, with a correlation of -0.825.

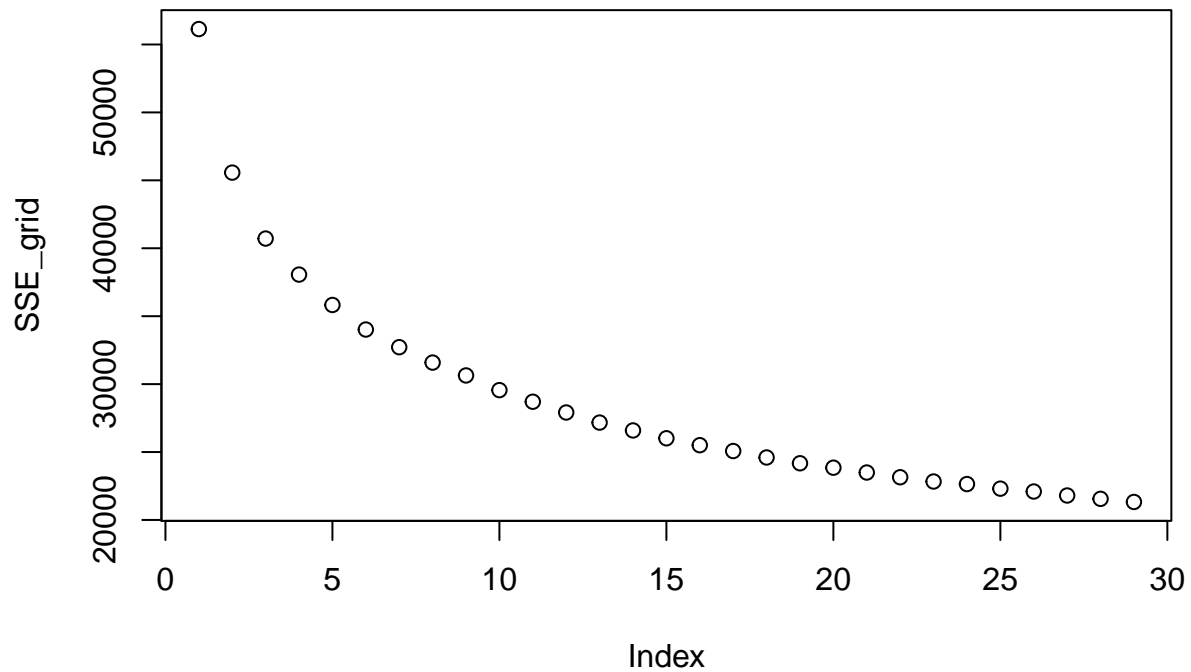


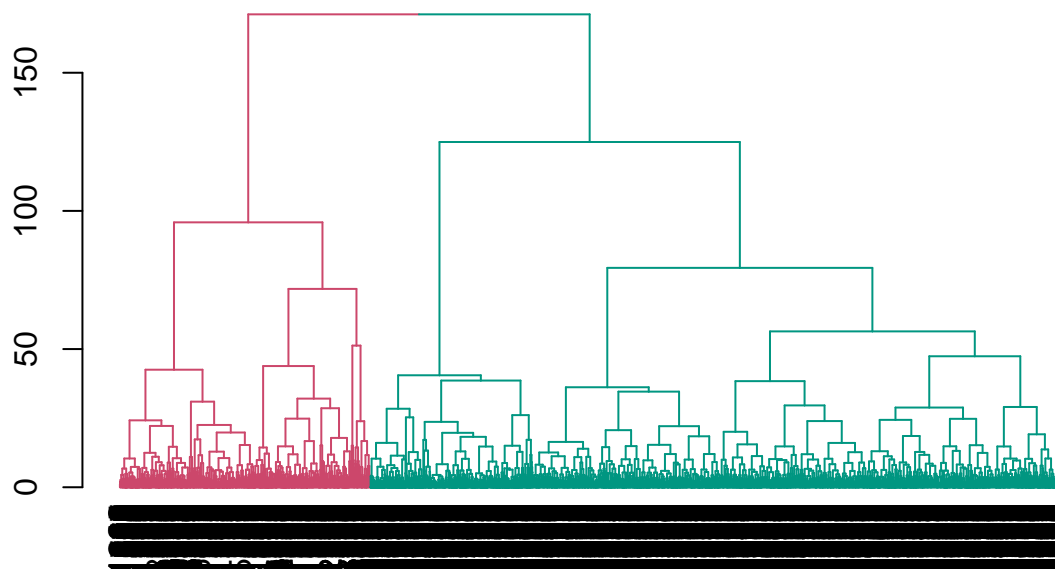
```
## [1] 0.07614681
```

Figure 3 displays the rated quality of wine and PC1 of either red or white wines. PCA did a poor job at determining the rated quality of wine, with a correlation of -0.076.

Hierarchical Clustering

As a separate approach, I decided to create a cluster dendrogram by using hierarchical clustering. First, the euclidean method was used to find the distances between each point. Next, using the found distances and the Ward method, a cluster dendrogram was produced. The Ward method was used to create the dendrogram because it produced a dendrogram that was much more well-balance relative to other used methods. Lastly, I applied the elbow method and determined that $k = 2$ was the optimal number of clusters for the data set, which can be observed on the following graph.





```
## cluster
##      1      2
## 1741 4756

## [1] 0.9274149

## [1] -0.1193233
```

The resulting cluster dendrogram produced is well-balanced. Moreover, the branches were given to separate colors corresponding to the resulting clustering so as to give a better visualization. Cluster 1 and cluster 2 contain 1471 and 4757 wines, respectively. This aligns well with the actual amounts of red and white wines in the data set; 1599 red wines and 4898 white wines. And indeed, the correlation of the clusters and the data set is 0.927, proving that hierarchical clustering is more effective method in distinguishing between red and white wines. Conversely, hierarchical clustering wasn't very successful in predicting wine quality, a correlation of -0.119, but was also more successful in distinguishing wine quality relative to PCA.

Question 2: Market Segmentation

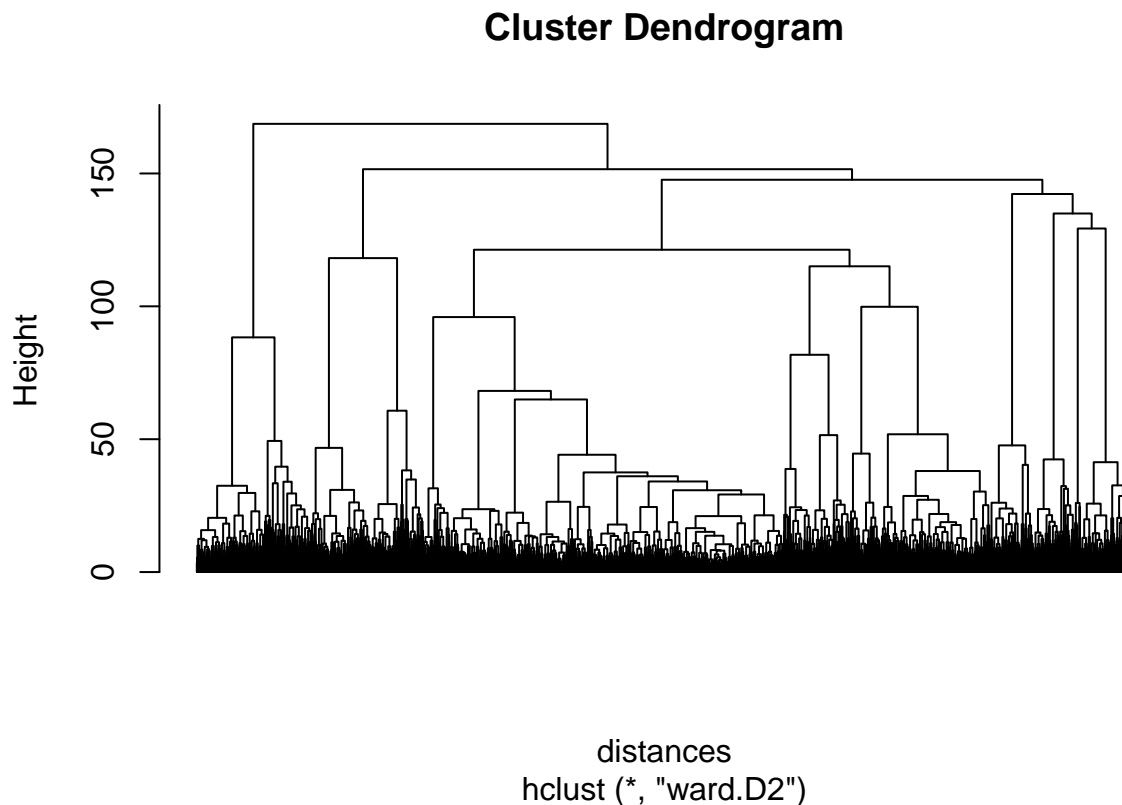
I. Overview

A sample of Twitter followers and corresponding tweets were taken for the company, NutrientH2O. The tweets were grouped into a data set of 36 categories. The objective is to analyze the data set and determine a possible correlation among twitter followers and their subsequent tweets, and use the results for better marketing techniques in promoting products made by NutrientH2O.

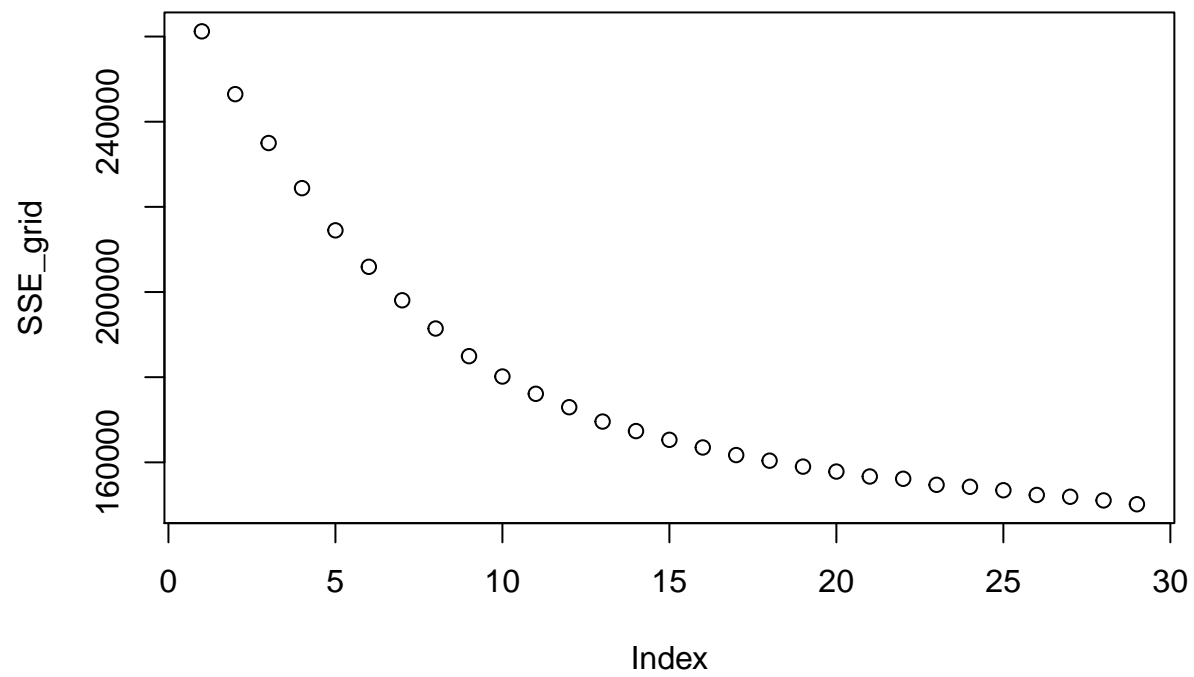
II. Data and Model

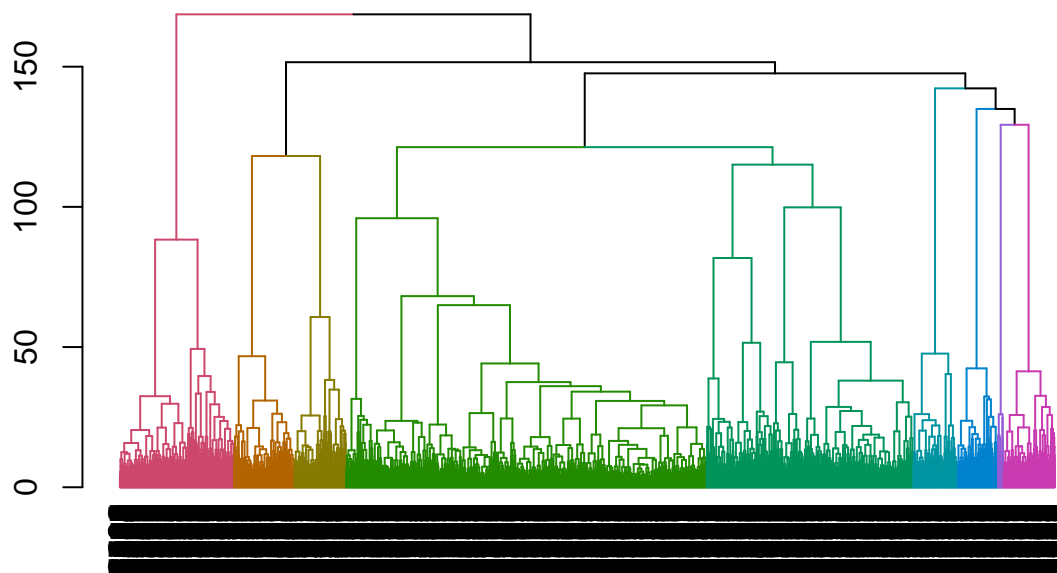
To accomplish this objective, I decided to create a cluster dendrogram by using hierarchical clustering. First, the euclidean method was used to find the distances between each point. Next, using the found distances and the Ward method, a cluster dendrogram was produced. The Ward method was used to create the dendrogram because it produced a dendrogram that was much more well-balance relative to other used methods.

III. Results

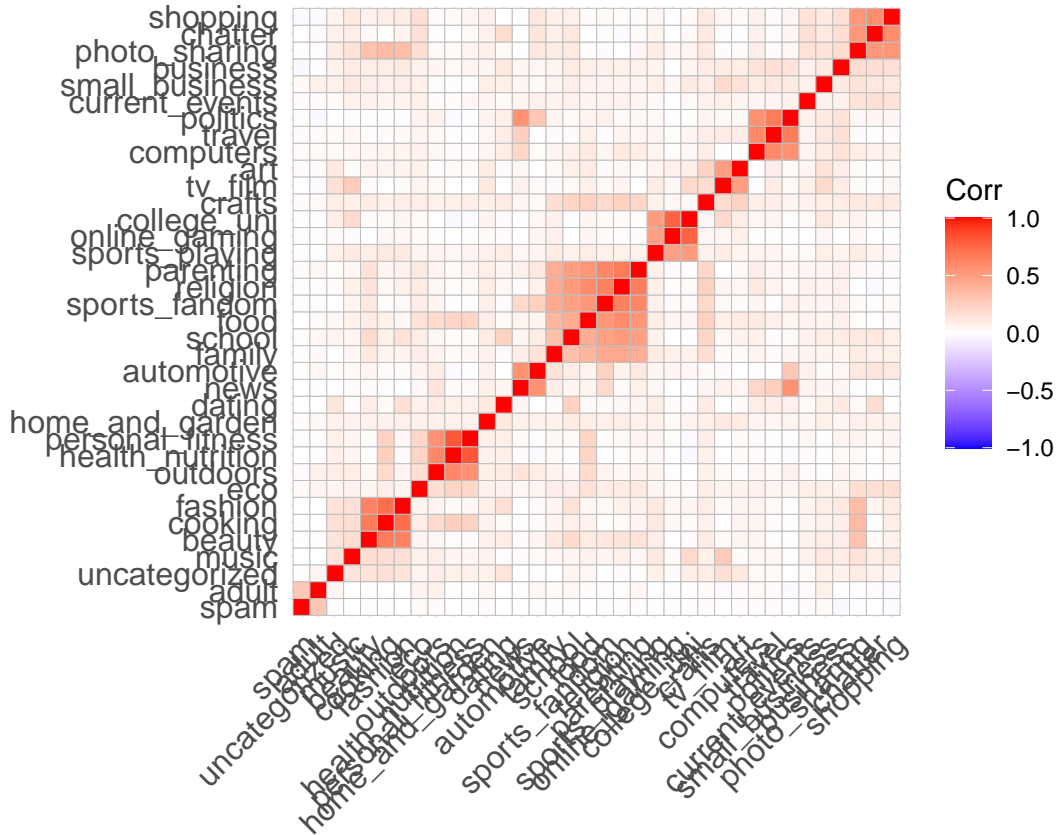


The resulting cluster dendrogram is well-balanced. The next step is to find the optimal k to cut the tree. To accomplish this, I used the elbow method in the following graph and observed $k = 9$ would be optimal.





The next cluster dendrogram is a replica of the first, but with the colored branches corresponding to the created cluster of $k = 9$ so as to create a better visualization of the process.

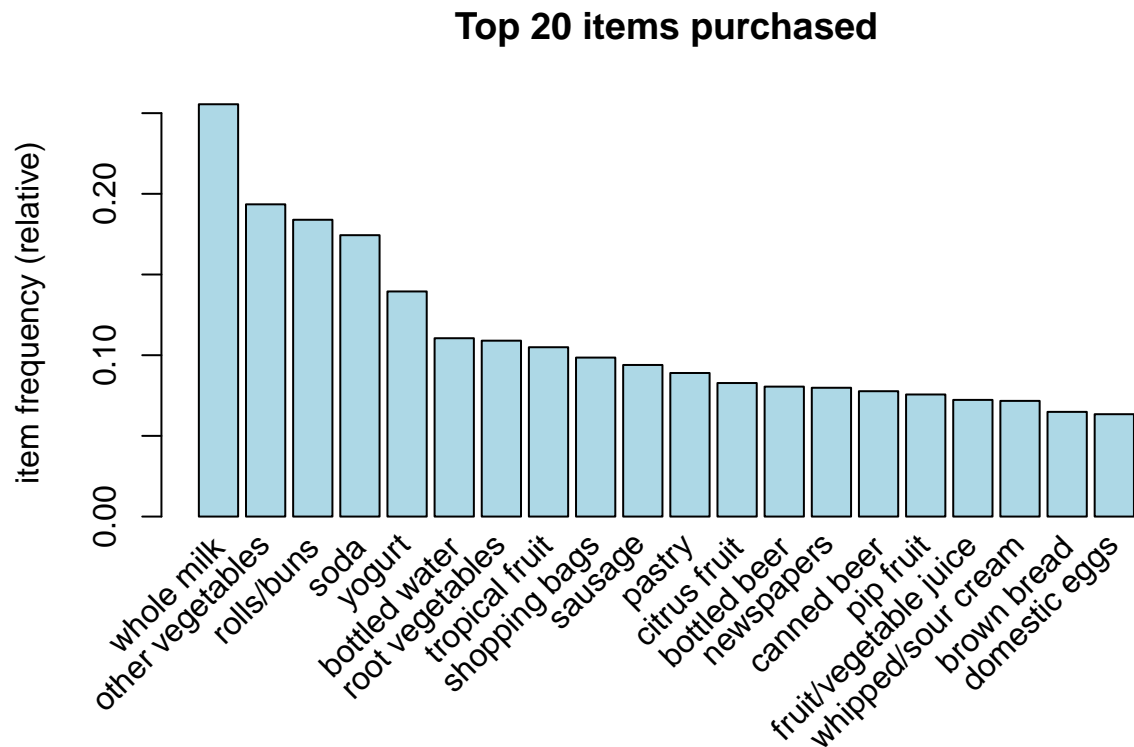


The final plot displays the correlation of the listed categories of tweets.

IV. Conclusion

After using hierarchical clustering to analyze the data set, there are several resulting market segments. The largest market segment lies in the cluster of Twitter followers that tweet topics that fall within the sports_fandom category. These followers also tend to tweet topics that also fall within the parenting, religion, food, school, and family category. These tweets could be scrutinized as to be more family-oriented in nature, and NutrientH20 could benefit through an increase in sales if it marketed itself as a family-oriented company. There was also a smaller and more obvious clustering in which contained tweets categorized as outdoors, health_nutrition, and personal_fitness. Moreover, spam and adult tweets resulted in the same cluster, which shouldn't be too surprising since both are most likely the result of Twitter bots.

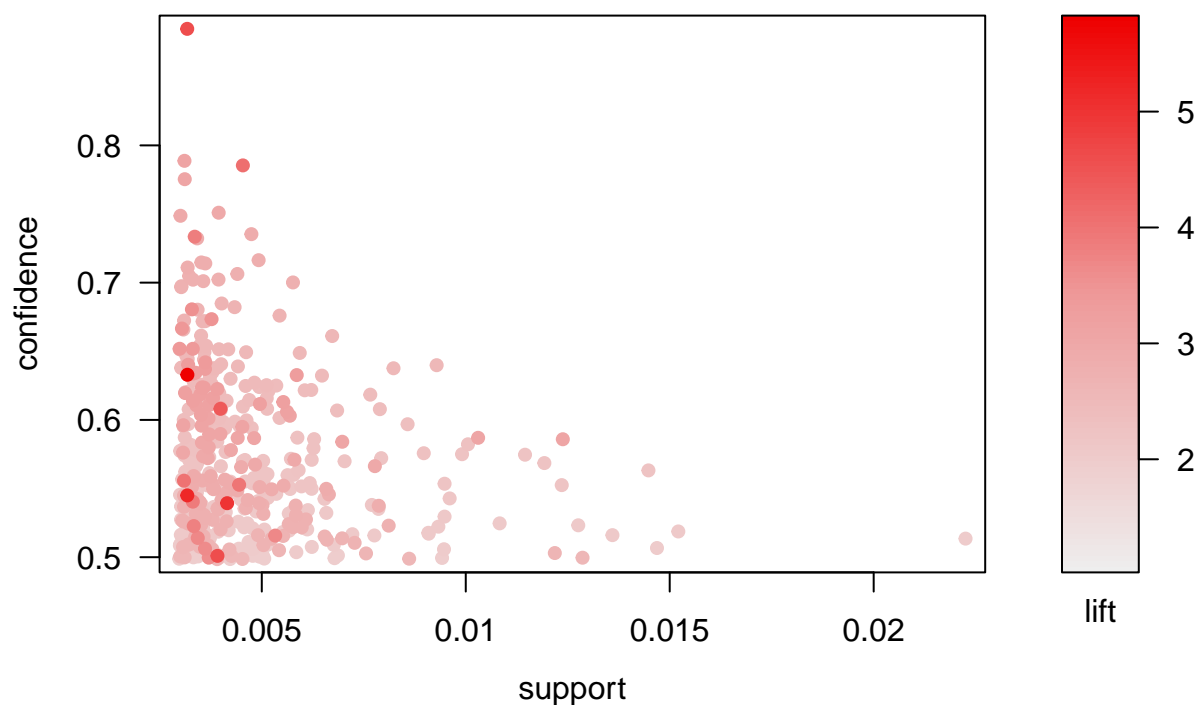
Question 3: Association Rules for Grocery Purchases



```
## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport maxtime support minlen
##          0.5    0.1    1 none FALSE                TRUE      5  0.003    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: 29
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [136 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [421 rule(s)] done [0.00s].
## creating S4 object ... done [0.00s].

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

Scatter plot for 421 rules

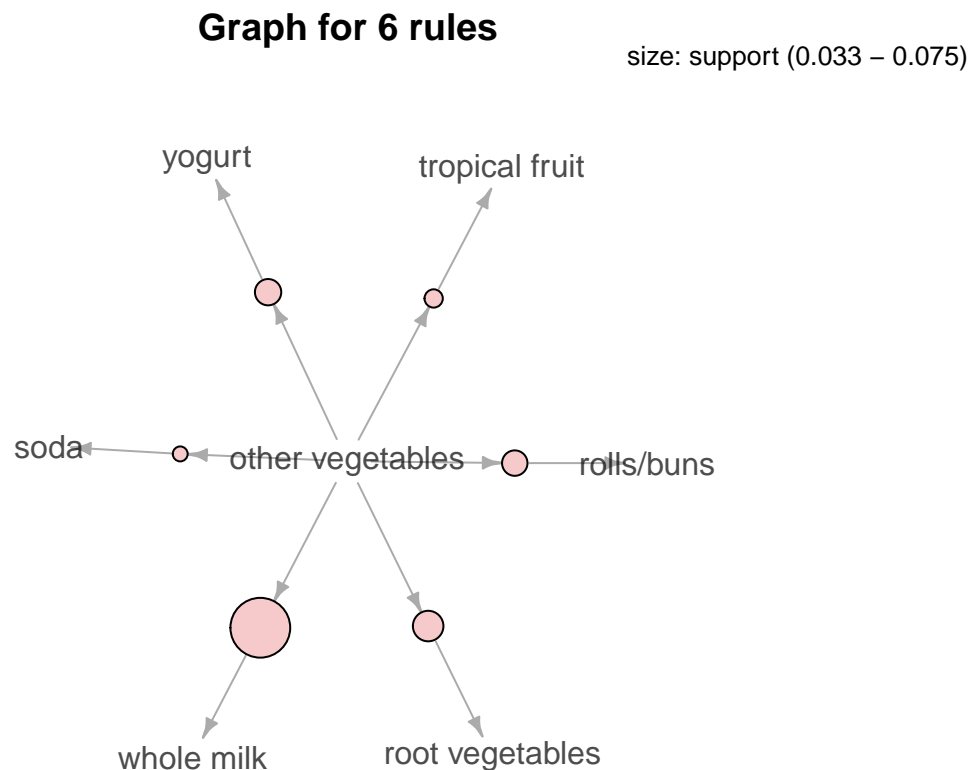


Upon observation of this scatter plot, I noticed a break point along the confidence axis at 0.65, and chose this as an appropriate level to further research the data. Moreover, lift > 3, was also an important metric drawn from this plot. Next, using lift > 3 and confidence > 0.65, I constructed the following table. At this criteria, the most frequent rhs items are “other vegetables” and “whole milk”. The lhs items tend to be the most common household items, such as root vegetables, cheese, brown bread, butter, and yogurt.

##	lhs	rhs	support	confidence	coverage	lift	count
## [1]	{root vegetables, sliced cheese}	=> {other vegetables}	0.003762074	0.6727273	0.005592272	3.476759	37
## [2]	{brown bread, whipped/sour cream}	=> {other vegetables}	0.003050330	0.6521739	0.004677173	3.370536	30
## [3]	{onions, root vegetables, whole milk}	=> {other vegetables}	0.003253686	0.6808511	0.004778851	3.518744	32
## [4]	{brown bread, other vegetables, root vegetables}	=> {whole milk}	0.003152008	0.7750000	0.004067107	3.033078	31
## [5]	{margarine, root vegetables, whole milk}	=> {other vegetables}	0.003253686	0.6530612	0.004982206	3.375122	32
## [6]	{butter, tropical fruit, yogurt}	=> {other vegetables}	0.003050330	0.6666667	0.004575496	3.445437	30
## [7]	{butter, root vegetables, yogurt}	=> {whole milk}	0.003050330	0.7894737	0.003863752	3.089723	30

```
## [8] {root vegetables,
##      tropical fruit,
##      whipped/sour cream} => {other vegetables} 0.003355363 0.7333333 0.004575496 3.789981 33
## [9] {citrus fruit,
##      root vegetables,
##      tropical fruit}      => {other vegetables} 0.004473818 0.7857143 0.005693950 4.060694 44
## [10] {citrus fruit,
##       root vegetables,
##       tropical fruit,
##       whole milk}         => {other vegetables} 0.003152008 0.8857143 0.003558719 4.577509 31
```

```
## Warning in plot.rules(rules_soda_1, method = "graph", interactive = FALSE, : The
## parameter interactive is deprecated. Use engine='interactive' instead.
```



Lastly, I created a graph that provides the likelihood of a consumer purchasing other items, given that the consumer purchased “other vegetables”. The most likely of which is whole milk, and shouldn’t be too surprising since whole milk was the most common purchased item in the data set. Furthermore, consumers that purchased other vegetables, are least likely to purchase soda, indicting these particular consumers may be more health conscious than most.

Question 4: Author Attribution

```
## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)): transformation
## drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(removeNumbers)):
## transformation drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(removeNumbers)):
## transformation drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(removePunctuation)):
## transformation drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(stripWhitespace)):
## transformation drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(removeWords),
## stopwords("SMART")): transformation drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)): transformation
## drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(removeNumbers)):
## transformation drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(removePunctuation)):
## transformation drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(stripWhitespace)):
## transformation drops documents

## Warning in tm_map.SimpleCorpus(., content_transformer(removeWords),
## stopwords("SMART")): transformation drops documents

## Warning in TermDocumentMatrix.SimpleCorpus(x, control): custom functions are
## ignored

## <<DocumentTermMatrix (documents: 2500, terms: 644)>>
## Non-/sparse entries: 184397/1425603
## Sparsity          : 89%
## Maximal term length: 18
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)

## Warning in TermDocumentMatrix.SimpleCorpus(x, control): custom functions are
## ignored

## <<DocumentTermMatrix (documents: 2500, terms: 660)>>
## Non-/sparse entries: 188410/1461590
## Sparsity          : 89%
## Maximal term length: 18
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

```

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'robert'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'august'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'form'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'numbers'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'proposal'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'completed'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'release'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'currency'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'dollar'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'overseas'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'signed'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'corps'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'weak'. Did you use factors
## with numeric labels for training, and numeric values for new data?

```

```

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'approved'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'family'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'limited'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'single'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'health'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'push'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'secretary'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'vote'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'regulatory'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'calls'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'theyre'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'opposition'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'savings'. Did you use
## factors with numeric labels for training, and numeric values for new data?

```

```

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'plant'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'remaining'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'win'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'active'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'countrys'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'gains'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'index'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'strategic'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'series'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'australian'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'toronto'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'quoted'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'campaign'. Did you use
## factors with numeric labels for training, and numeric values for new data?

```



```

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'body'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'owns'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'combined'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'resources'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'pacific'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'asian'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'offers'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'profitable'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'france'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'southern'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'cchina'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'human'. Did you use factors
## with numeric labels for training, and numeric values for new data?

## Warning in predict.naiveBayes(naive_bayes, DTM_test[, -ncol(DTM_test)]): Type
## mismatch between training and new data for variable 'dollars'. Did you use
## factors with numeric labels for training, and numeric values for new data?

## [1] 992

```