Assignment 4

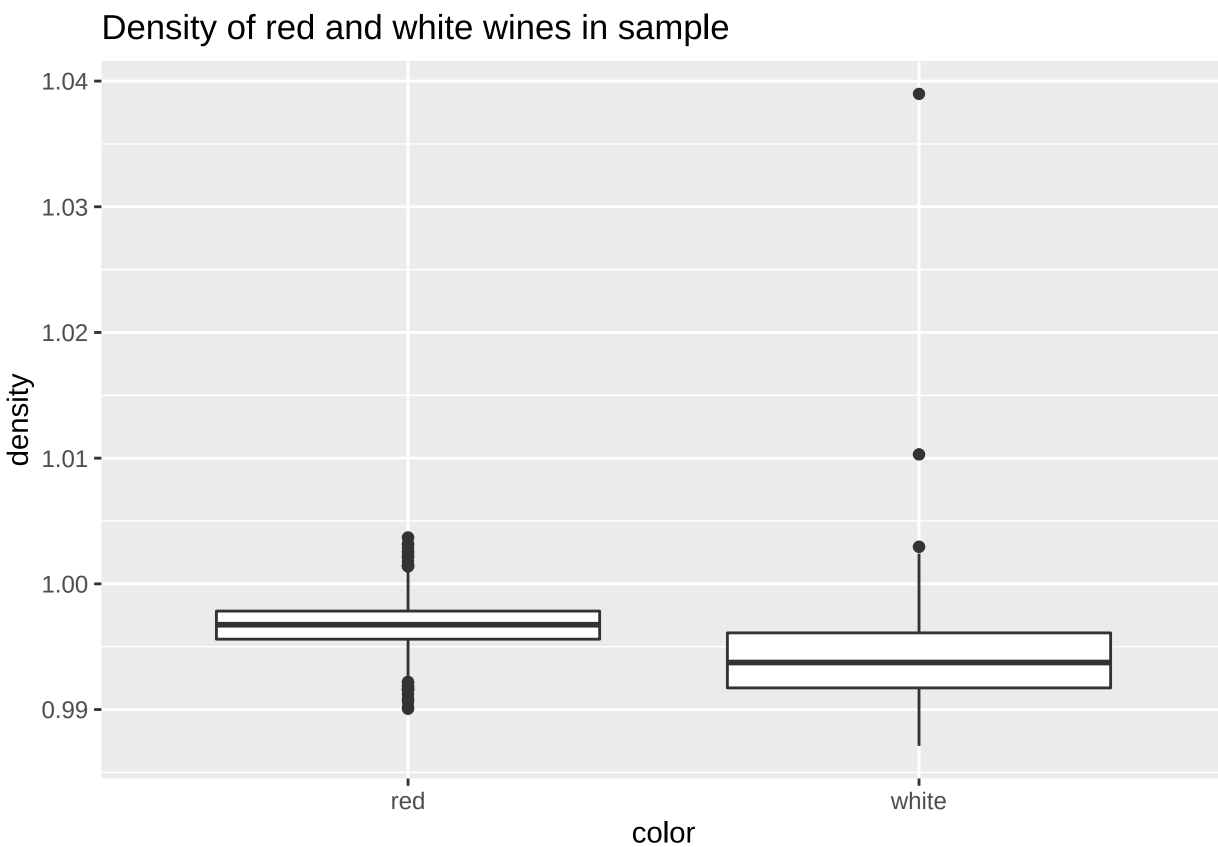
Maria Gilbert

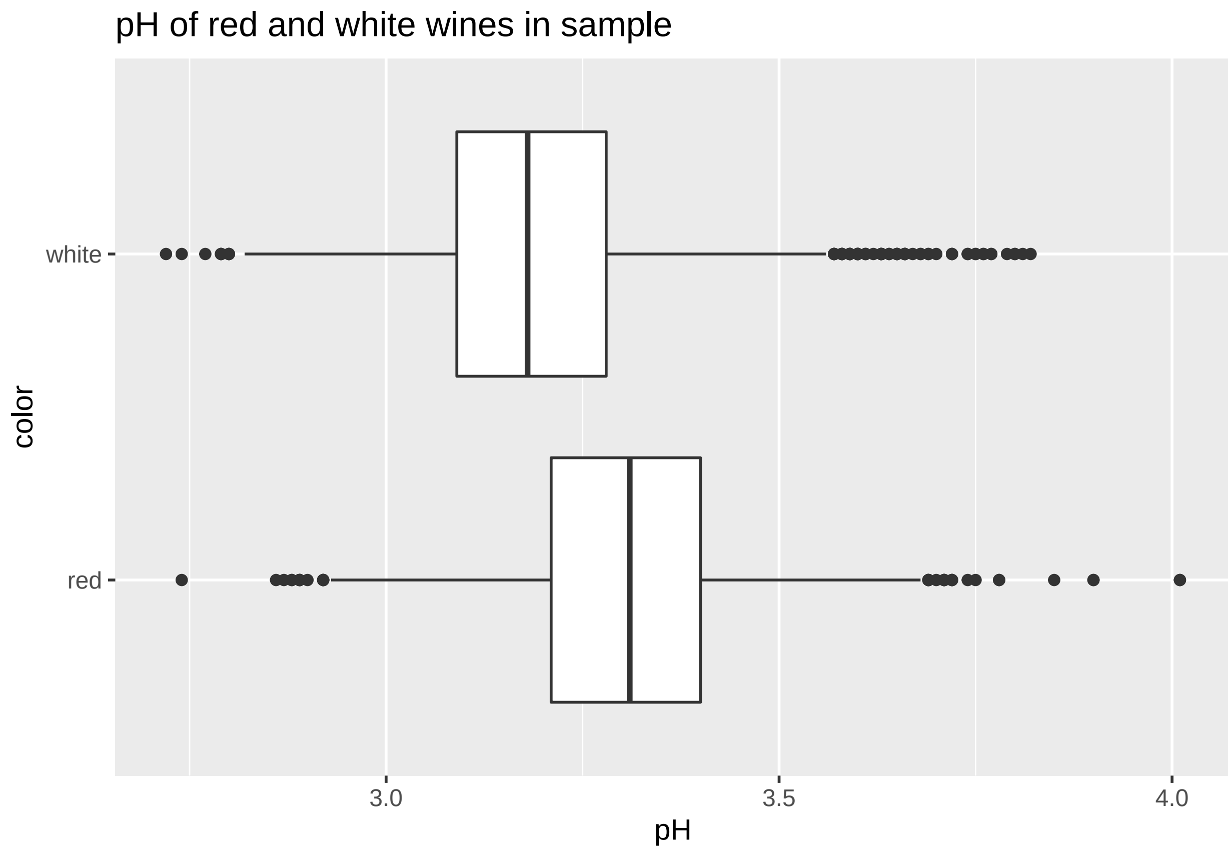
5/7/2021

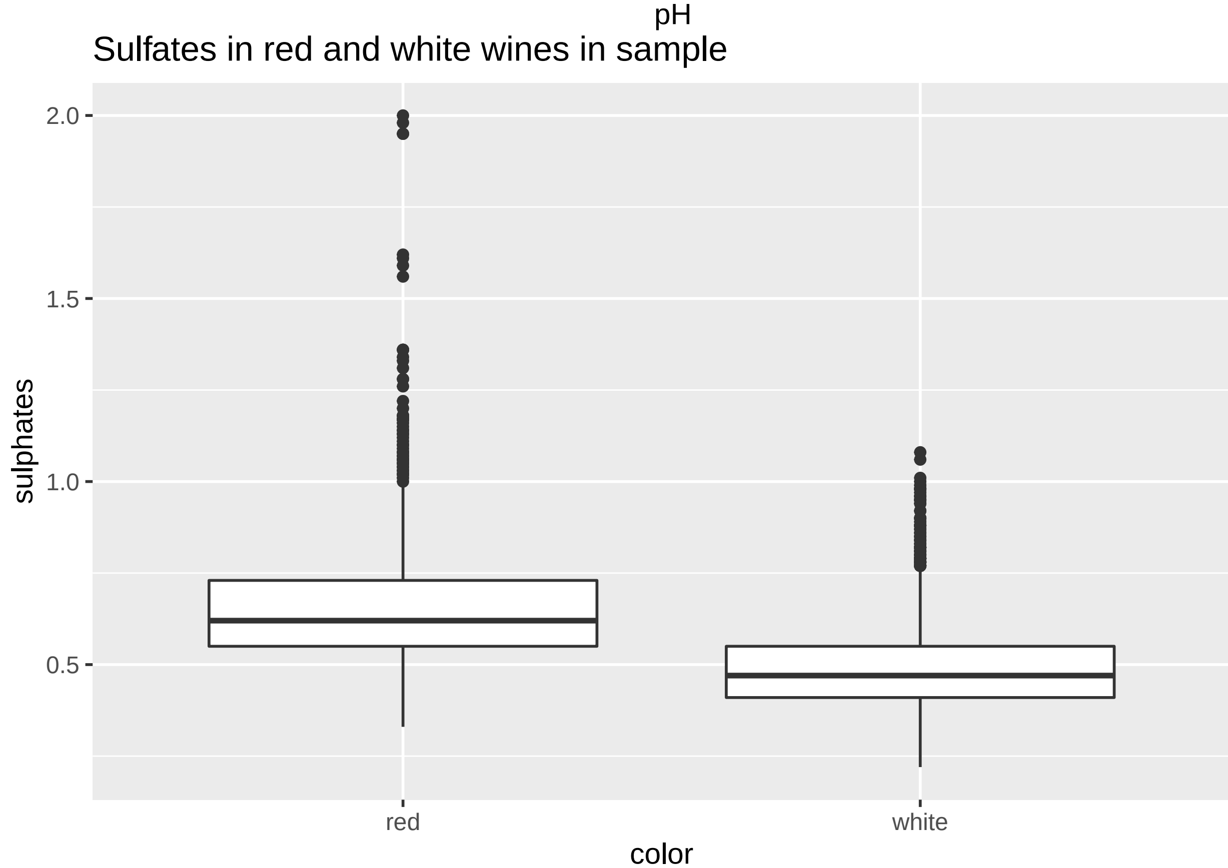
Clustering and PCA

Using data on 6500 different bottles of *vinho verde* wine from Northern Portugal, my goal is to use unsupervised learning to find a pattern that can predict whether a wine is red or white. My data includes 4898 white wines and 1599 red wines, with information on 11 chemical properties, including fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol, as well as an indicator of quality ranging from 1-10.

First, I want to see the relationships between density, pH, sulphates and color, using boxplots.







I am also interested in seeing the actual correlations between wine color and all the other variables in the data set, to see which are the strongest indicators of whether a wine is white or red. The following table shows the correlation between color and all other variables:

```
##                             White         Red
## Fixed acidity        -0.48673983 -0.65303559
## Volatile acidity      0.18739650  0.34882101
## Citric acid          -0.51267825  0.47164366
## Residual sugar        0.70035716 -0.39064532
## Chlorides            -0.32912865 -0.48721797
## Free sulfur dioxide   0.03296955  0.11932328
## Total sulfur dioxide  0.48673983  0.65303559
## Density              -0.18739650 -0.34882101
## pH                    0.51267825 -0.47164366
## Sulfates             -0.70035716  0.39064532
## Alcohol               0.32912865  0.48721797
## Quality              -0.03296955 -0.11932328
```

I find something interesting here, which is that there appears to be a very similar effect of residual sugar and sulfates on whether a wine is white or red. However, the correlation between residual sugar and sulfates is not as high as I would expect given this observation. This correlation is:

```
## [1] -0.1859274
```

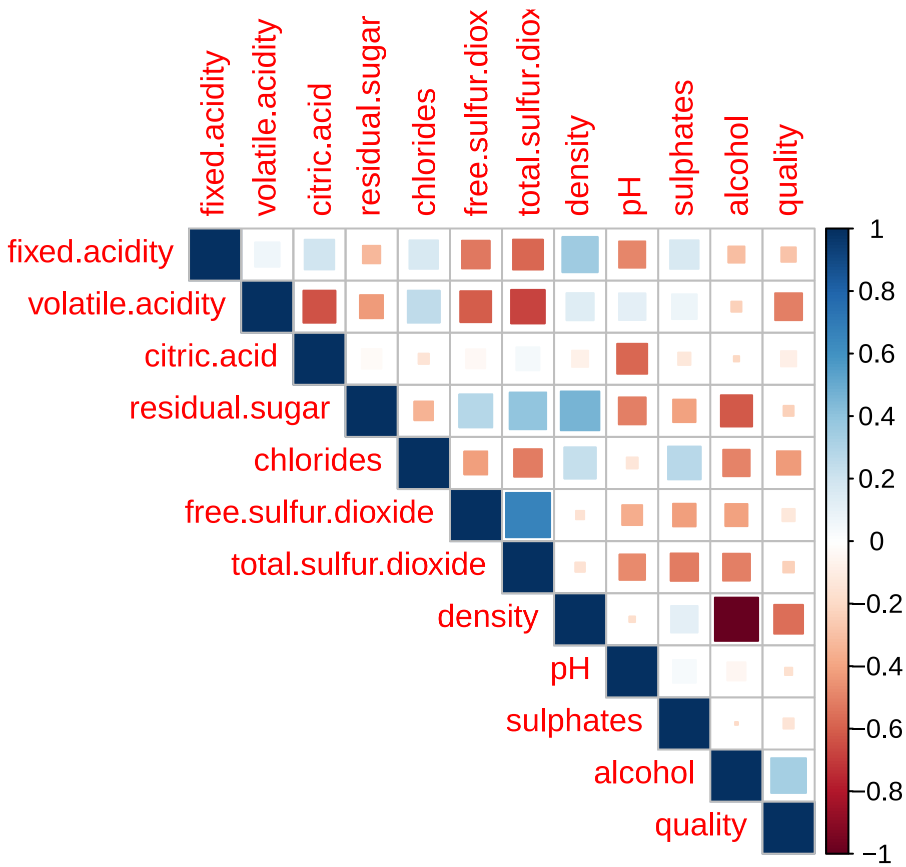
Similarly, I notice a similar effect between alcohol and free sulfur dioxide. This correlation is:

```
## [1] -0.1798384
```

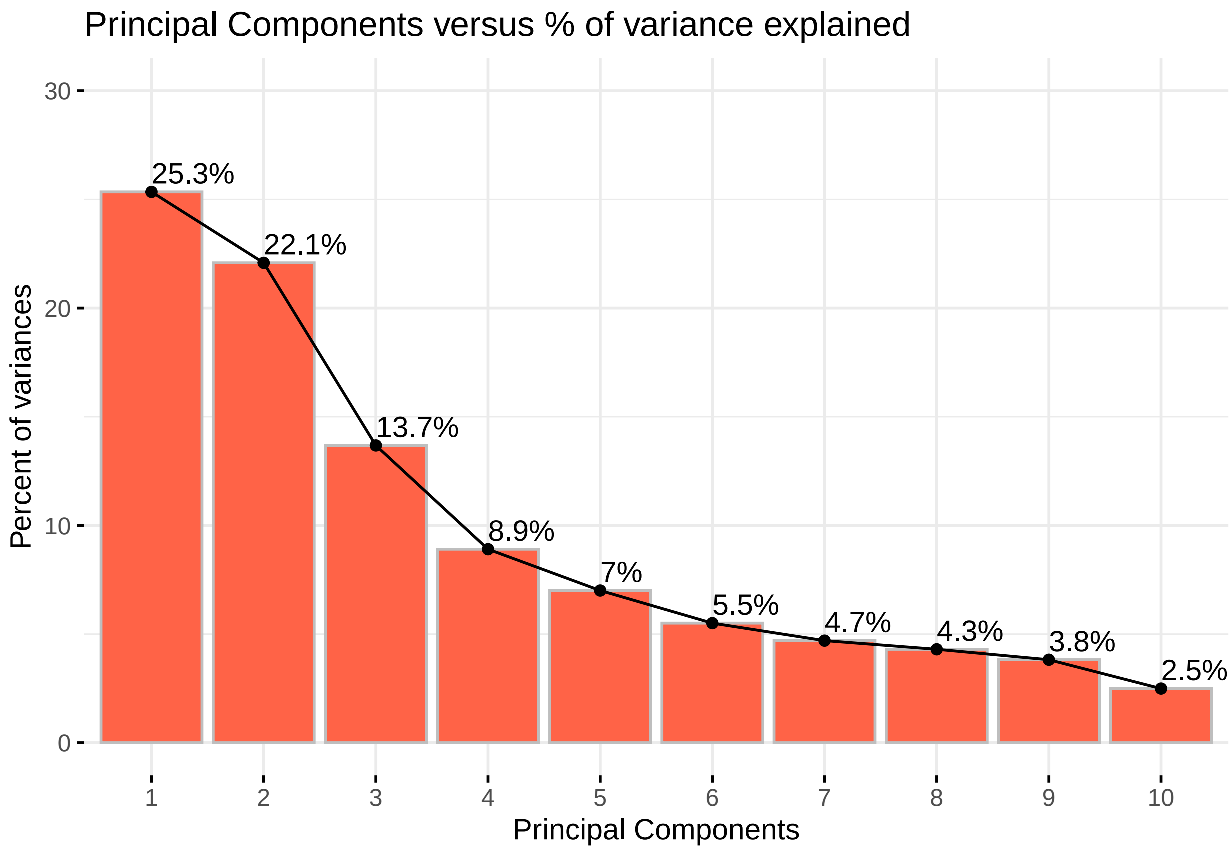
Now, I know that wine color is probably most related to residual sugar, sulfates, fixed acidity, citric acid, total sulfur dioxide, alcohol, pH, and chlorides. Less important factors are quality, free sulfur dioxide, volatile acidity, and density.

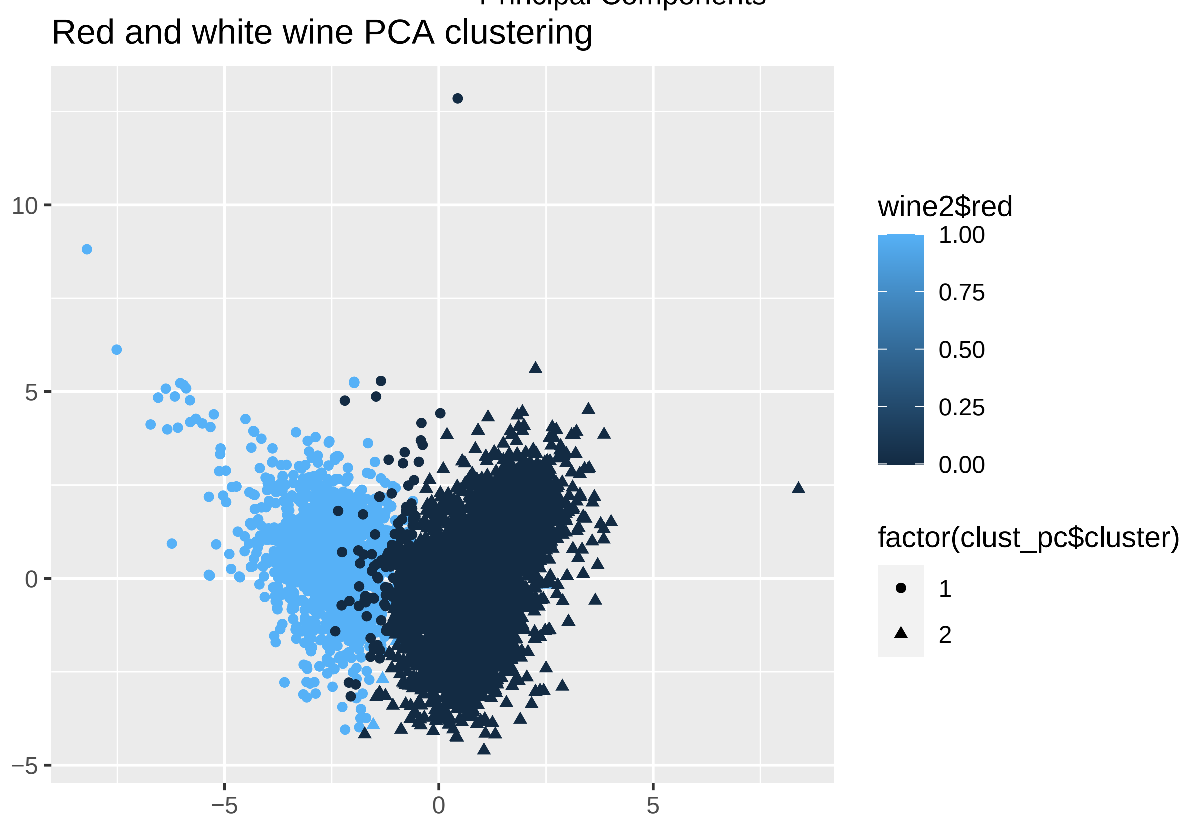
PCA: Principal Components Analysis

Here, I will use Principal Components Analysis to find the components which can predict whether a wine is white or red. Instead of using the qualities that I previously found to have relatively high correlations with wine color, I will be allowing the algorithm to organize the information from the data.




```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.7440 1.6278 1.2812 1.03374 0.91679 0.81265 0.75088
## Proportion of Variance 0.2535 0.2208 0.1368 0.08905 0.07004 0.05503 0.04699
## Cumulative Proportion  0.2535 0.4743 0.6111 0.70013 0.77017 0.82520 0.87219
##                           PC8    PC9    PC10    PC11    PC12
## Standard deviation     0.7183 0.6770 0.54682 0.47706 0.18107
## Proportion of Variance 0.0430 0.0382 0.02492 0.01897 0.00273
## Cumulative Proportion  0.9152 0.9534 0.97830 0.99727 1.00000
```

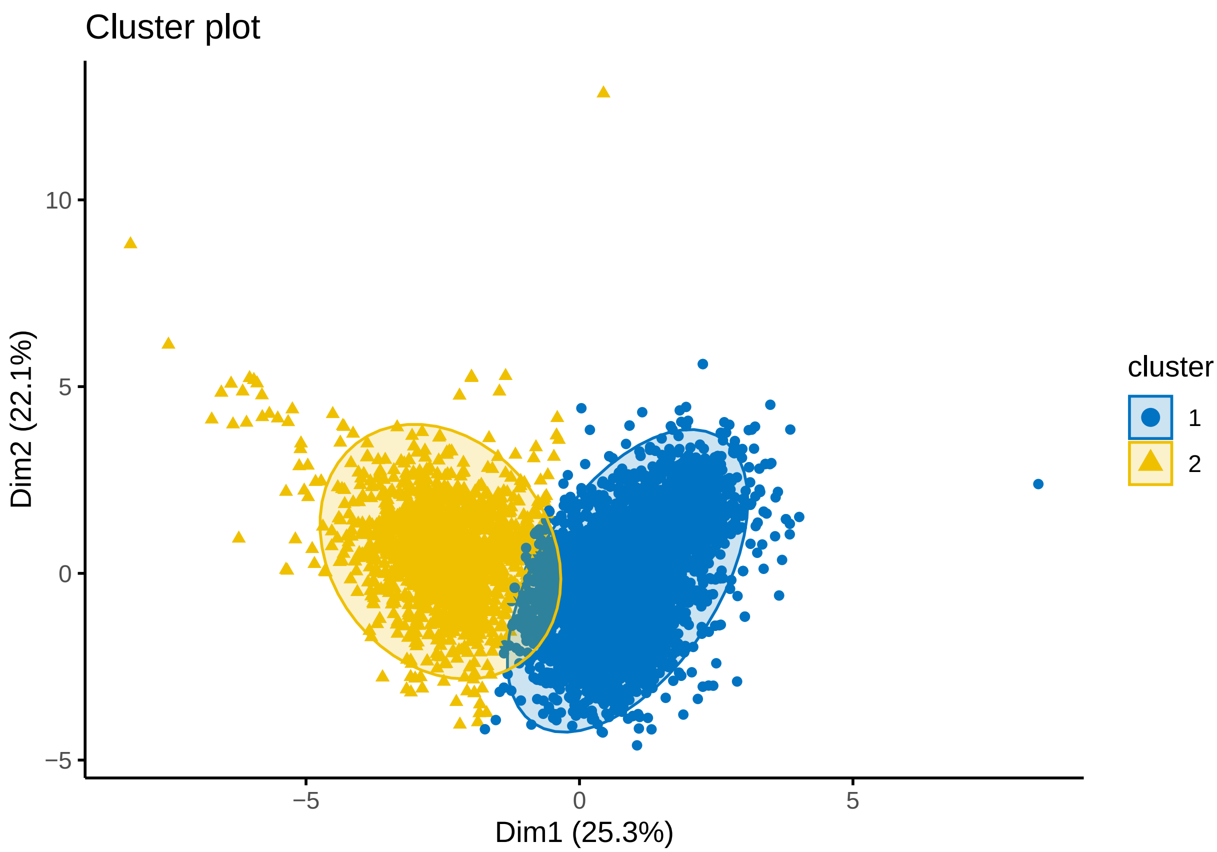

Principal Components versus % of variance explained
Percent of variances
Principal Components
25.3%
22.1%
13.7%
8.9%
7%
5.5%
4.7%
4.3%
3.8%
2.5%


Red and white wine PCA clustering
wine2$red
factor(clust_pc$cluster)

K-means clustering

K-means clustering is a method of centroid-based clustering, where clusters are represented by a central vector or centroid. This method organizes the data into k clusters. Since I am trying to see a pattern to predict whether a wine is red or white, I will use k = 2 for this exercise.

```
## List of 9
##  $ cluster     : int [1:6497] 2 2 2 2 2 2 2 2 2 2 ...
##  $ centers     : num [1:2, 1:12] -0.283 0.82 -0.4 1.159 0.116 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:12] "fixed.acidity" "volatile.acidity" "citric.acid" "residual.sugar" ...
##  $ totss       : num 77952
##  $ withinss    : num [1:2] 42240 20238
##  $ tot.withinss: num 62478
##  $ betweenss   : num 15474
##  $ size        : int [1:2] 4829 1668
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```



Now that I have identified 2 clusters, I want to see how many red versus white wines are present in each of the 2 clusters.

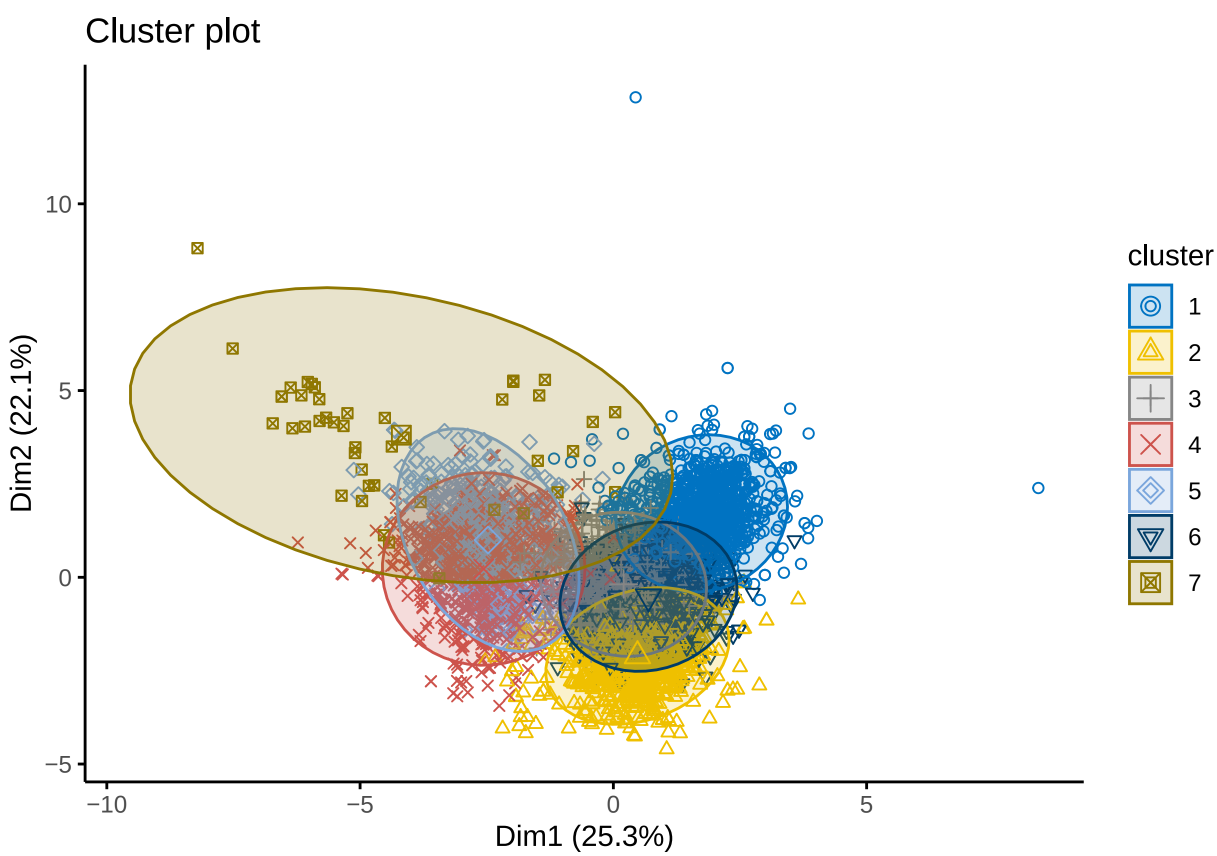
```
##              wine$color
## wine$cluster  red white
##            1   25  4804
##            2 1574    94
```

Given 1,599 red wines and 4,898 white wines, and the above data table, I can use Bayes' Theorem to evaluate how closely my K-means clustering can predict the color of a wine. Overall within this data set there is a 24.6% of any one randomly chosen wine being red, and a 75.4% chance of being white. Within the 1,668 wines in cluster 2, there is a 94.7% chance on any of randomly chosen wine being red. Within the 4849 wines in cluster 1, there is a 99.4% chance of any randomly chosen wine being white.

K-means clustering for wine quality

Now, I will apply the same method, using k = 7 clusters to represent quality categories of 3, 4, 5, 6, 7, 8, and 9.

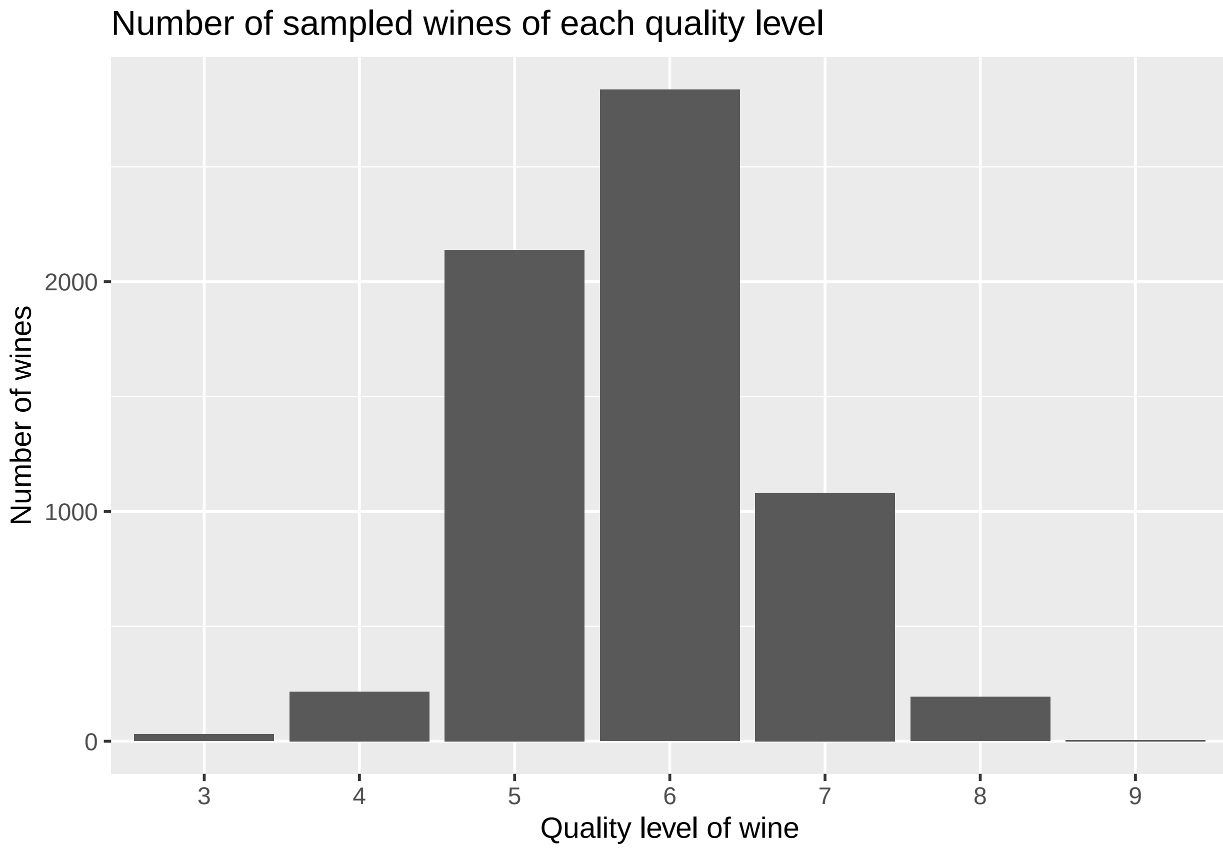
```
## List of 9
##  $ cluster     : int [1:6497] 4 4 4 5 4 4 4 4 4 4 ...
##  $ centers     : num [1:7, 1:12] -0.17248 -0.44114 0.00883 0.11537 2.10081 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:7] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:12] "fixed.acidity" "volatile.acidity" "citric.acid" "residual.sugar" ...
##  $ totss       : num 77952
##  $ withinss    : num [1:7] 9163 6855 6060 5889 4806 ...
##  $ tot.withinss: num 39042
##  $ betweenss   : num 38910
##  $ size        : int [1:7] 1494 1348 1084 971 584 967 49
##  $ iter        : int 5
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```




```
##              wine$quality
```

```
## wine$cluster7   3   4   5   6   7   8   9
##             1   8  27 668 654 119  18   0
##             2   0   0  21 572 604 146   5
##             3   9  92 561 410  12   0   0
##             4   6  73 507 343  40   2   0
##             5   3   8 158 267 137  11   0
##             6   1  14 196 574 166  16   0
##             7   3   2  27  16   1   0   0
```

In the data set, we have 30 wines of quality 3, 216 wines of quality 4, 2138 wines of quality 5, 2836 wines of quality 6, 1079 wines of quality 7, 193 wines of quality 8, and 5 wines of quality 9.



Again, I can use Bayes' theorem to evaluate how well the clustering algorithm lines up with wine quality. The following table shows the % chances of a wine in any given cluster being of a certain quality level.

```
##                     3           4           5           6           7
## Cluster 1 0.005354752 0.018072289 0.447121821 0.437751004 0.079651941
## Cluster 2 0.000000000 0.000000000 0.015578635 0.424332344 0.448071217
## Cluster 3 0.008302583 0.084870849 0.517527675 0.378228782 0.011070111
## Cluster 4 0.006179197 0.075180227 0.522142122 0.353244078 0.041194645
## Cluster 5 0.005136986 0.013698630 0.270547945 0.457191781 0.234589041
## Cluster 6 0.001034126 0.014477766 0.202688728 0.593588418 0.171664943
## Cluster 7 0.061224490 0.040816327 0.551020408 0.004220309 0.020408163
##                     8           9
## Cluster 1 0.012048193 0.000000000
## Cluster 2 0.108308605 0.003709199
## Cluster 3 0.000000000 0.000000000
## Cluster 4 0.002059732 0.000000000
```

```
## Cluster 5 0.018835616 0.000000000
## Cluster 6 0.016546019 0.000000000
## Cluster 7 0.000000000 0.000000000
```

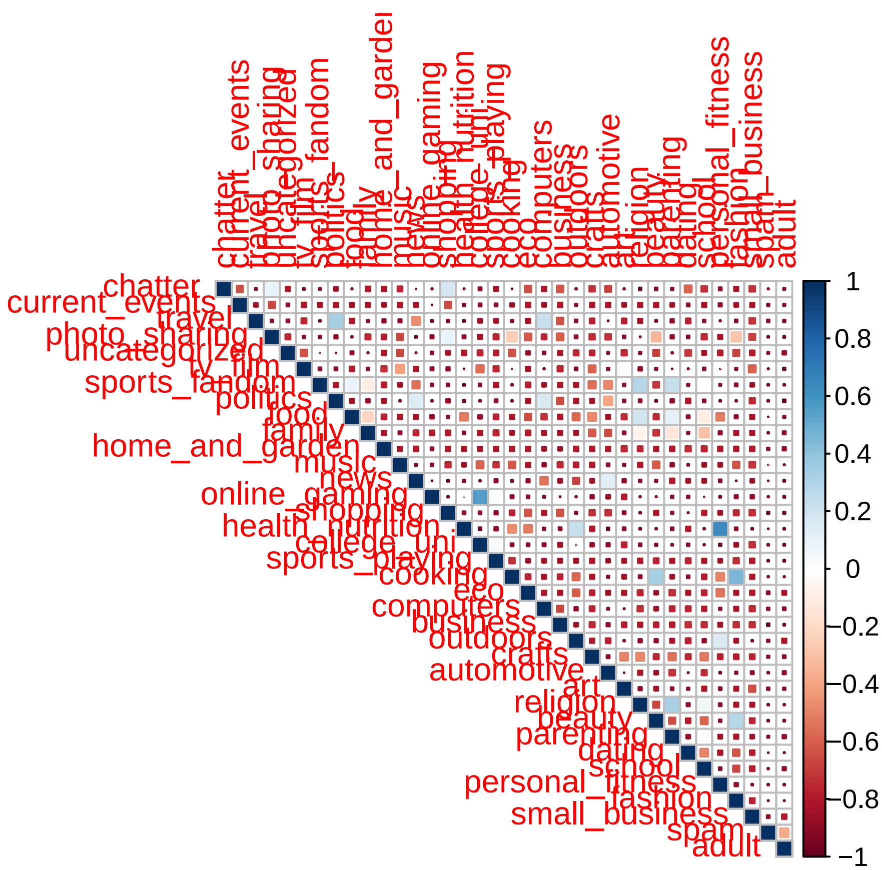
Market Segmentation

Using data on Twitter activity from 7882 randomly selected users, I would like to use clustering to find patterns in the data. This data includes how much interaction users had that is categorized as chatter, current events, travel, photo sharing, tv and film, sports fans, politics, food, family, home and garden, music, news, online gaming, shopping, health and nutrition, college and universities, playing sports, cooking, eco, computers, business, outdoors, crafts, automotive, art, religion, beauty, parenting, dating, school, personal fitness, fashion, small business, adult, spam, and uncategorized material.

The first thing I want to is make sure that I am filtering out users who have either 0 values for all of these content categories, as well as those who have 0 values for all those except spam, adult, and/or uncategorized. I am doing this because I believe these users are probably bots and won't be relevant in my analysis.

It turns out there were ZERO users who had either all 0 values or all 0 values except for spam, adult, and uncategorized.

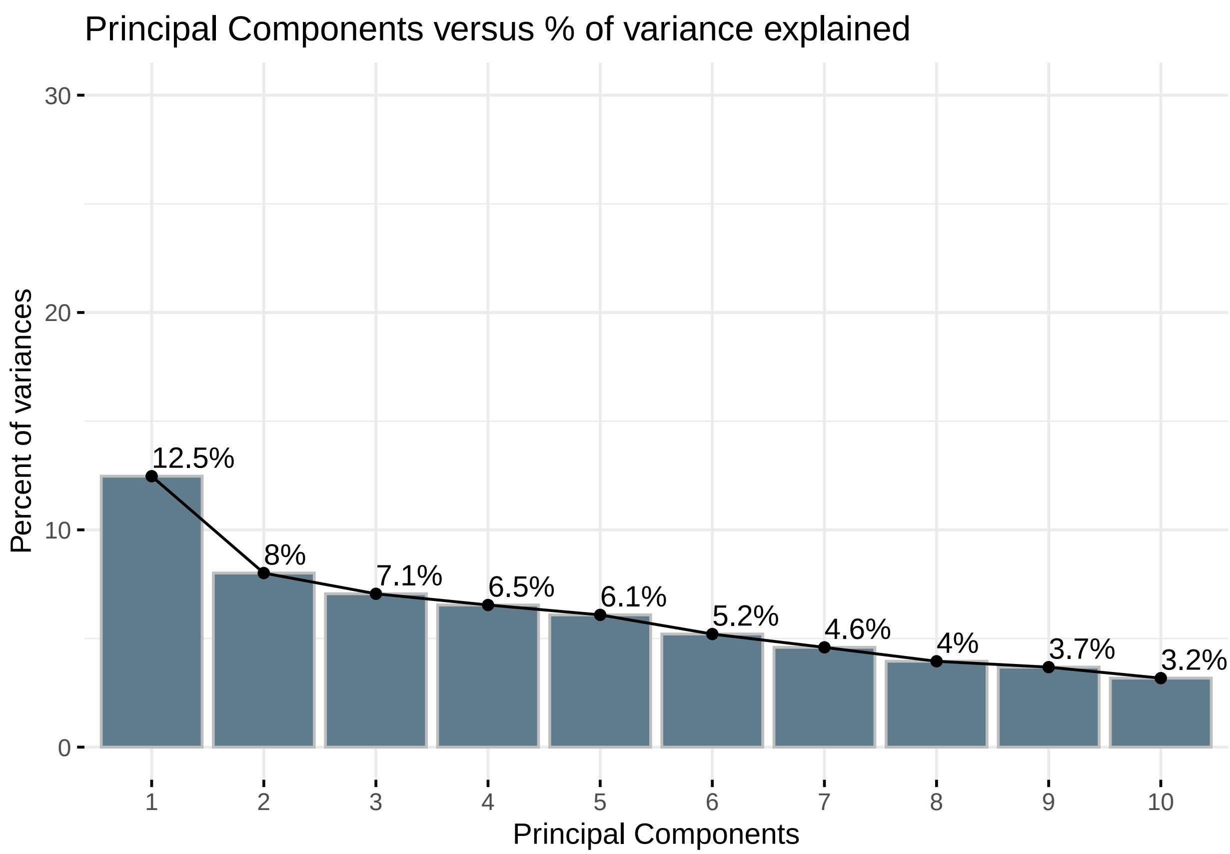
It can be expected that certain categories will be correlated with each other. For example, family and parenting or news and current events. The following figure shows each category's correlation with the others.

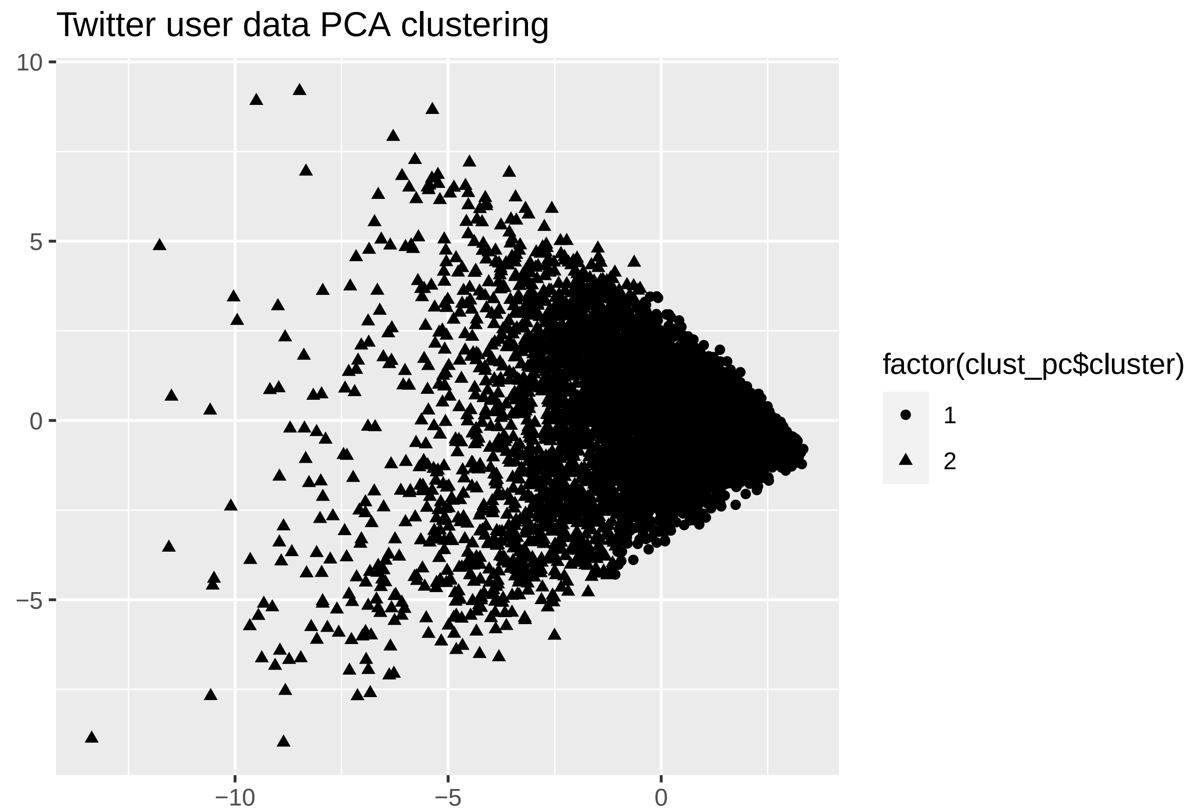


This plot is almost too big to understand, so thankfully we have PCA and K-means to better organize this data. I will try PCA as well as K-means testing out several different values of K to see which one seems most appropriate.

PCA

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.1186 1.69824 1.59388 1.53457 1.48027 1.36885 1.28577
## Proportion of Variance 0.1247 0.08011 0.07057 0.06541 0.06087 0.05205 0.04592
## Cumulative Proportion  0.1247 0.20479 0.27536 0.34077 0.40164 0.45369 0.49961
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     1.19277 1.15127 1.06930 1.00566 0.96785 0.96131 0.94405
## Proportion of Variance 0.03952 0.03682 0.03176 0.02809 0.02602 0.02567 0.02476
## Cumulative Proportion  0.53913 0.57595 0.60771 0.63580 0.66182 0.68749 0.71225
##                           PC15    PC16   PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.93297 0.91698 0.9020 0.85869 0.83466 0.80544 0.75311
## Proportion of Variance 0.02418 0.02336 0.0226 0.02048 0.01935 0.01802 0.01575
## Cumulative Proportion  0.73643 0.75979 0.7824 0.80287 0.82222 0.84024 0.85599
##                           PC22    PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     0.69632 0.68558 0.65317 0.64881 0.63756 0.63626 0.61513
## Proportion of Variance 0.01347 0.01306 0.01185 0.01169 0.01129 0.01125 0.01051
## Cumulative Proportion  0.86946 0.88252 0.89437 0.90606 0.91735 0.92860 0.93911
##                           PC29    PC30    PC31   PC32    PC33    PC34    PC35
## Standard deviation     0.60167 0.59424 0.58683 0.5498 0.48442 0.47576 0.43757
## Proportion of Variance 0.01006 0.00981 0.00957 0.0084 0.00652 0.00629 0.00532
## Cumulative Proportion  0.94917 0.95898 0.96854 0.9769 0.98346 0.98974 0.99506
##                           PC36
## Standard deviation     0.42165
## Proportion of Variance 0.00494
## Cumulative Proportion  1.00000
```

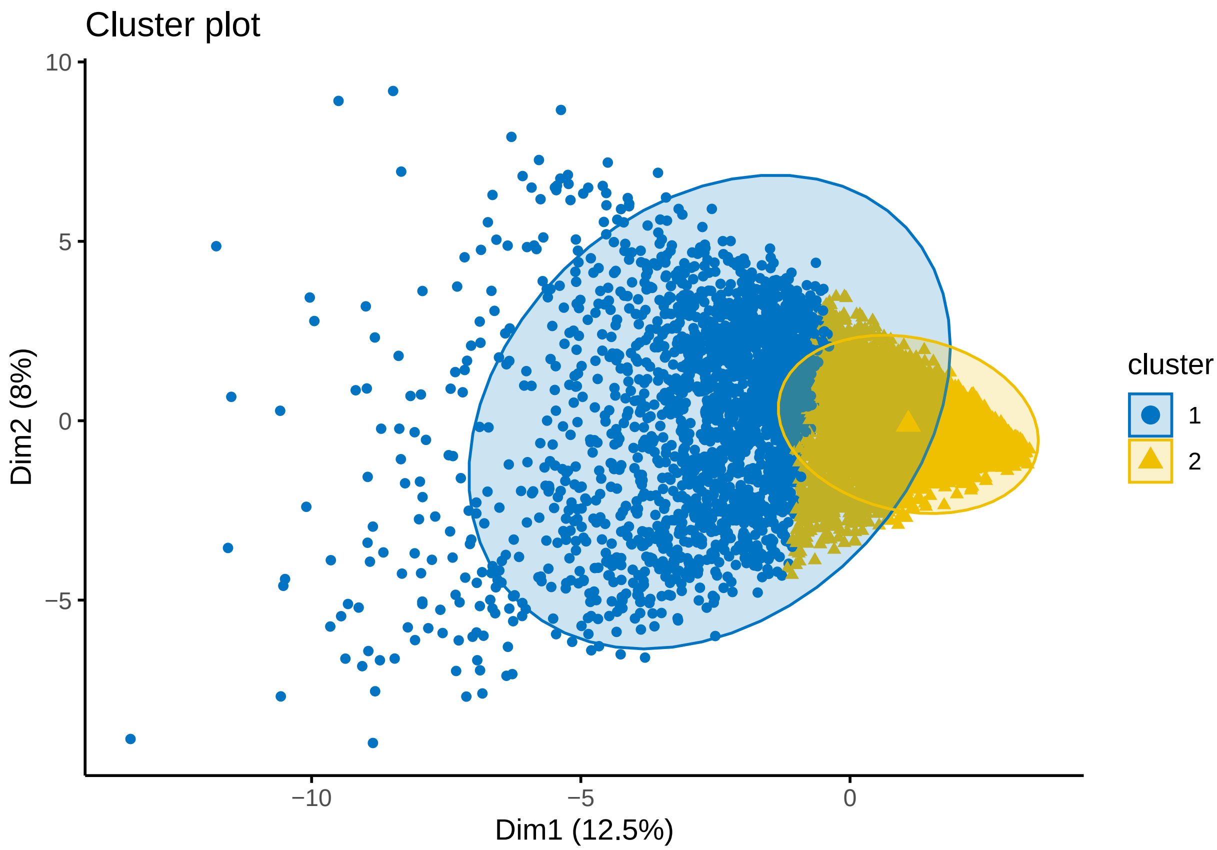




Unfortunately, principal components do not seem to do a very good job of explaining variances in this exercise, with the top 10 principal compoments only explaining 61% of the variance. It is not terrible, but I think that K-means will do a little better.

K-means clustering

```
## List of 9
##  $ cluster     : int [1:7882] 1 2 2 2 2 2 2 2 1 1 ...
##  $ centers     : num [1:2, 1:36] 0.361 -0.15 0.241 -0.1 0.302 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:36] "chatter" "current_events" "travel" "photo_sharing" ...
##  $ totss       : num 283716
##  $ withinss    : num [1:2] 143578 117642
##  $ tot.withinss: num 261220
##  $ betweenss   : num 22496
##  $ size        : int [1:2] 2311 5571
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

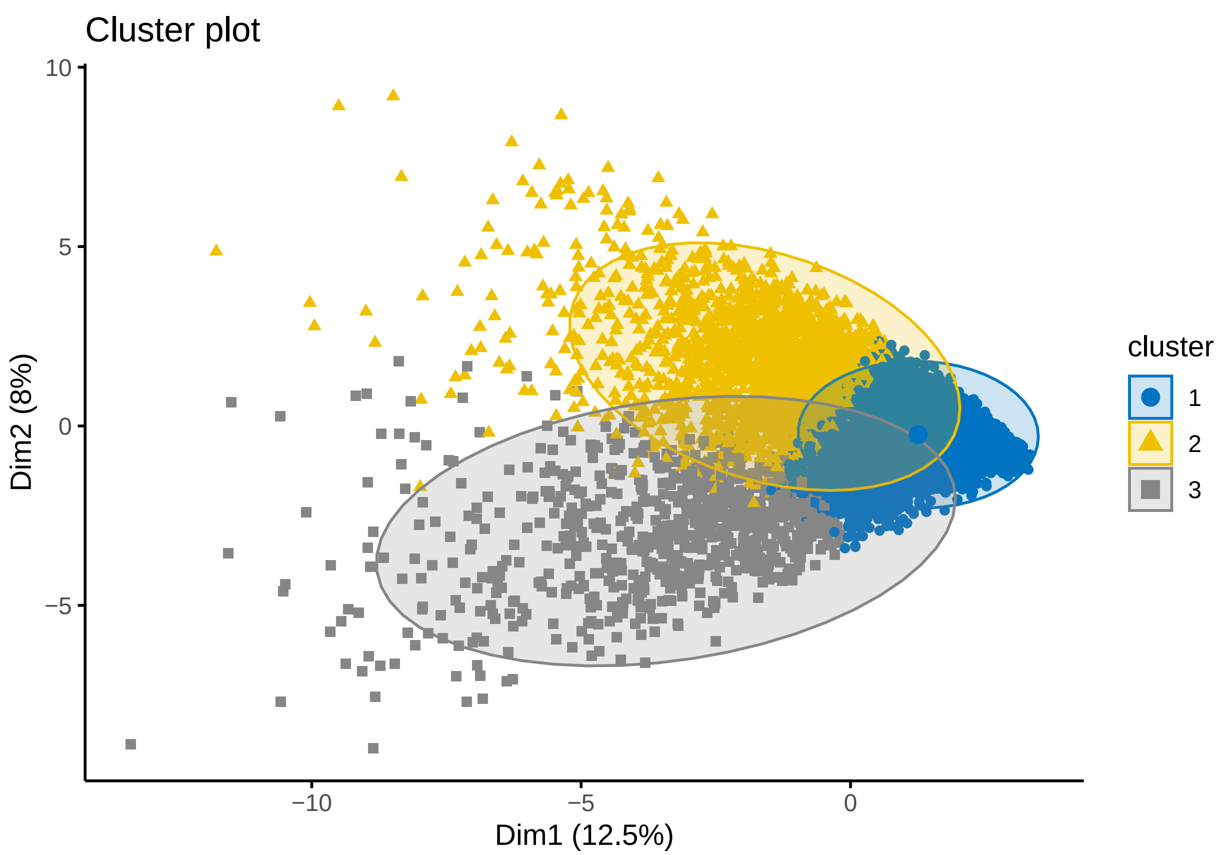



```
##                            Cluster 1   Cluster 2
## Chatter                  5.672868888 3.870220786
## Current Events           1.832107313 1.399389697
## Travel                   2.274340113 1.299048645
## Photo Sharing            4.131544786 2.101597559
## Uncategorized            1.061445262 0.709926405
## TV/film                  1.586758979 0.856040208
## Sports fandom            3.013846820 1.005026028
## Politics                 2.719601904 1.402441213
## Food                     2.687581134 0.862322743
## Family                   1.549545651 0.579429187
## Home and garden          0.749891822 0.749891822
## Music                    1.064041540 0.519655358
## News                     1.819558633 0.950816729
## Online Gaming            1.868022501 0.935379645
## Shopping                 2.086975335 1.099982050
## Health and Nutrition     4.167892687 1.903248968
## College and Universities 2.480311553 1.163345898
## Sports playing           1.028991778 0.477472626
## Cooking                  3.888792730 1.213965177
## Eco                      0.807442666 0.389876144
## Computers                1.090004327 0.466164064
## Business                 0.675897880 0.318434751
## Outdoors                 1.278234531 0.577095674
## Crafts                   0.936391173 0.936391173
## Automotive               1.249242752 0.655896607
## Art                      1.245781047 0.508705798
```

```
## Religion                    2.430549546 0.541554479
## Beauty                      1.487234963 0.380721594
## Parenting                   1.969710082 0.486447675
## Dating                      1.235828646 0.493089212
## School                      1.592816962 0.425417340
## Personal fitness            2.409779316 1.068928379
## Fashion                     2.008221549 0.576916173
## Small business              0.547382086 0.248788368
## Spam                        0.009086975 0.005385030
## Adult                       0.519688447 0.355052953
```

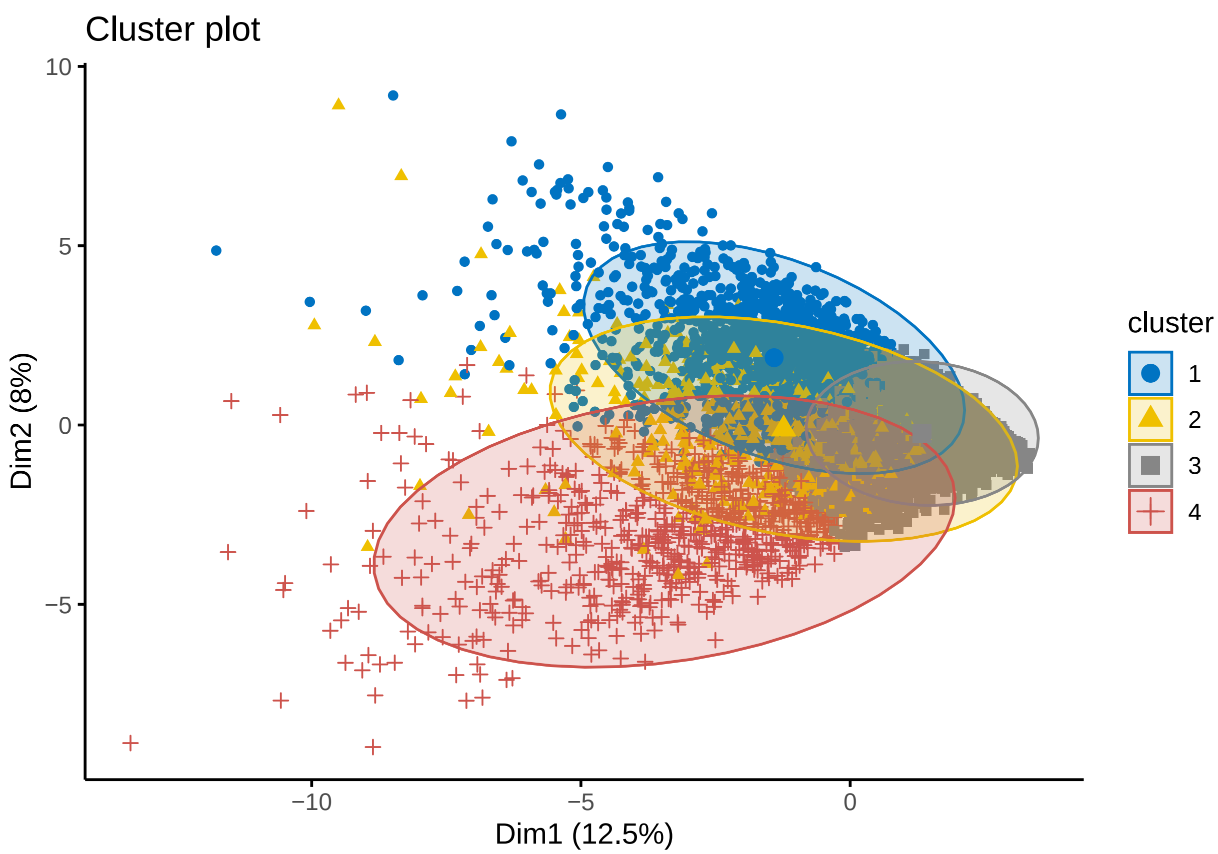
The main conclusion I can draw from this is that Cluster 1 constitutes more active Twitter users, while Cluster 2 constitutes less active users.

```
## List of 9
##  $ cluster     : int [1:7882] 2 1 2 1 1 1 1 2 2 3 ...
##  $ centers     : num [1:3, 1:36] -0.2279 0.5565 -0.0919 -0.1331 0.2528 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:36] "chatter" "current_events" "travel" "photo_sharing" ...
##  $ totss       : num 283716
##  $ withinss    : num [1:3] 92423 120704 33344
##  $ tot.withinss: num 246472
##  $ betweenss   : num 37244
##  $ size        : int [1:3] 4924 2150 808
##  $ iter        : int 4
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```



```
##                        Cluster 1 Cluster 2 Cluster 3
## Chatter                3.5944354 6.3627907 4.0742574
## News                   0.9335906 1.8302326 1.2004950
## Family                 0.5678310 0.9372093 2.4727723
## Health and Nutrition 1.6401300 4.8376744 2.1757426
## Business               0.2940699 0.6855814 0.5123762
```

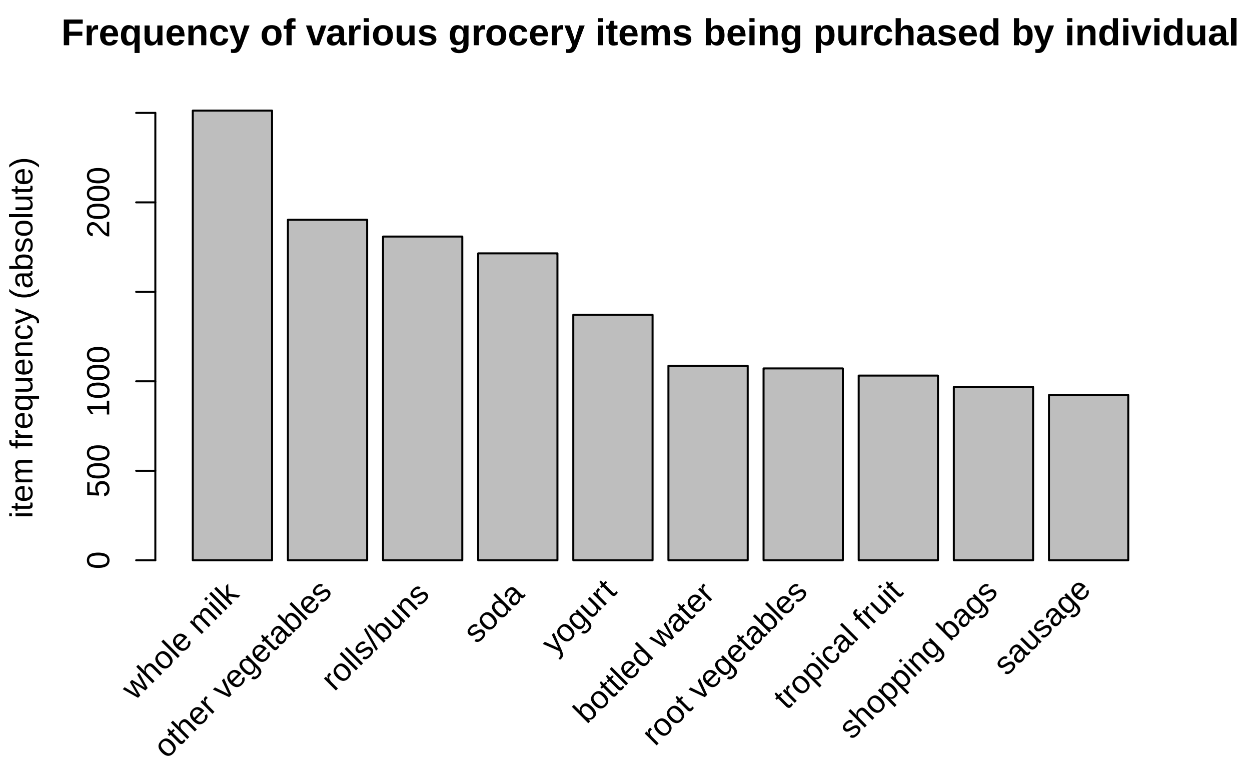
```
## List of 9
##  $ cluster     : int [1:7882] 1 3 1 3 3 3 2 1 1 4 ...
##  $ centers     : num [1:4, 1:36] 0.5514 0.0017 -0.2073 -0.082 0.2123 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:36] "chatter" "current_events" "travel" "photo_sharing" ...
##  $ totss       : num 283716
##  $ withinss    : num [1:4] 92031 31395 80557 31012
##  $ tot.withinss: num 234995
##  $ betweenss   : num 48721
##  $ size        : int [1:4] 1830 714 4570 768
##  $ iter        : int 7
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```




```
##                  Cluster 1 Cluster 2 Cluster 3 Cluster 4
## TV/film          1.6715847 1.1428571 0.8212254 1.0520833
## Sports fans      1.2109290 2.0420168 0.9433260 5.9622396
## Family           0.9103825 0.9299720 0.5566740 2.5195312
## Eco              0.8344262 0.5910364 0.3474836 0.6523438
## Personal fitness 3.0863388 1.1890756 0.8656455 1.3945312
```

Association rules for grocery purchases

```
## transactions as itemMatrix in sparse format with
##  15297 rows (elements/itemsets/transactions) and
##  173 columns (items) and a density of 0.0163888
##
## most frequent items:
##       whole milk other vegetables       rolls/buns             soda
##             2513             1903             1809             1715
##           yogurt          (Other)
##             1372            34059
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4
## 3485 2630 2102 7080
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   2.835   4.000   4.000
##
## includes extended item information - examples:
##             labels
## 1 abrasive cleaner
## 2 artif. sweetener
## 3   baby cosmetics
```



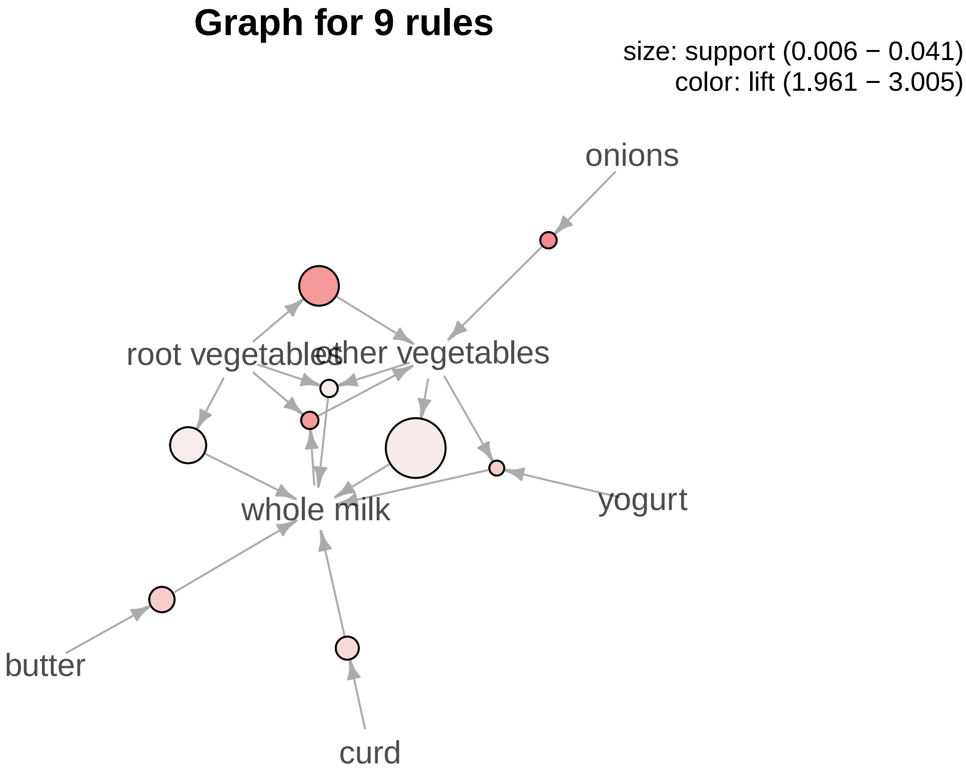

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
```

```
##         0.3    0.1    1 none FALSE            TRUE       5   0.005      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: 76
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[173 item(s), 15297 transaction(s)] done [0.01s].
## sorting and recoding items ... [101 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [9 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
## set of 9 rules
##
## rule length distribution (lhs + rhs):sizes
## 2 3
## 6 3
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   2.000   2.333   3.000   3.000
##
## summary of quality measures:
##     support           confidence        coverage            lift
##  Min.   :0.006341   Min.   :0.3222   Min.   :0.01589   Min.   :1.961
##  1st Qu.:0.008172   1st Qu.:0.3284   1st Qu.:0.02262   1st Qu.:1.999
##  Median :0.012617   Median :0.3619   Median :0.03426   Median :2.430
##  Mean   :0.016220   Mean   :0.3602   Mean   :0.04647   Mean   :2.430
##  3rd Qu.:0.022619   3rd Qu.:0.3738   3rd Qu.:0.07008   3rd Qu.:2.904
##  Max.   :0.040858   Max.   :0.4037   Max.   :0.12440   Max.   :3.005
##      count
##  Min.   : 97.0
##  1st Qu.:125.0
##  Median :193.0
##  Mean   :248.1
##  3rd Qu.:346.0
##  Max.   :625.0
##
## mining info:
##        data ntransactions support confidence
##  groceries2         15297   0.005        0.3
```

From this analysis, I can tell that: - 40.4% of those who bought butter also bought whole milk - 39.9% of those who bought other vegetables and yogurt also bought whole milk - 37.4% of those who bought onions also bought other vegetables - 36.8% of those who bought curd also bought whole milk - 36.2% of those who bought root vegetables also bought other vegetables - 36.1% of those who bought root vegetables and whole milk also bought other vegetables - 32.8% of those who bought other vegetables also bought whole milk - 32.3% of those who bought root vegetables also bought whole milk - 32.2% of those who bought other vegetables and root vegetables also bought whole milk

Basically, everyone is buying vegetables and whole milk. We can see that these items are fairly central to the network in the following graph.



Author Attribution

Using text analysis, I would like to create an unsupervised learning algorithm that will help to use text content to predict which author wrote a certain article.

I tried this problem for way too much time. I could not get it to work. This assignment was absolutely insane especially due less than one week before the project is due.