STA 380 Exercise

Yingjia Shang, Sharon Liu, Jiaxi Wang

2022-08-04

https://github.com/Sharon-Liu97/STA380_Exercise

Problem 1: Probability Practice

Part A

Given $P(RC) = 0.3 \quad P(TC) = 0.7$

$P(Yes) = 0.65 \quad P(No) = 0.35$

$TP = TC + RC$

$\Rightarrow 0.65 = P(TC \cap Yes) + 0.3 \times 0.5$

$\Rightarrow P(TC \cap Yes) = 0.65 - 0.15 = \underline{\underline{0.5}}$

Part B

Given $P(P|D) = 0.993 \quad P(\text{not } P \mid \text{not } D) = 0.9999$

$P(D) = 0.000025$

$$P(D|P) = \frac{P(D \cap P)}{P(P)} = \frac{P(D) - P(D \cap \text{Not } P)}{P(D) \times P(P|D)}$$

$$= \frac{0.000025 - 0.007 \times 0.000025}{0.000025 \times 0.993 + 0.999975 \times 0.0001}$$

$$= \underline{\underline{0.19885}}$$

Problem 2: Wrangling the Billboard Top 100

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

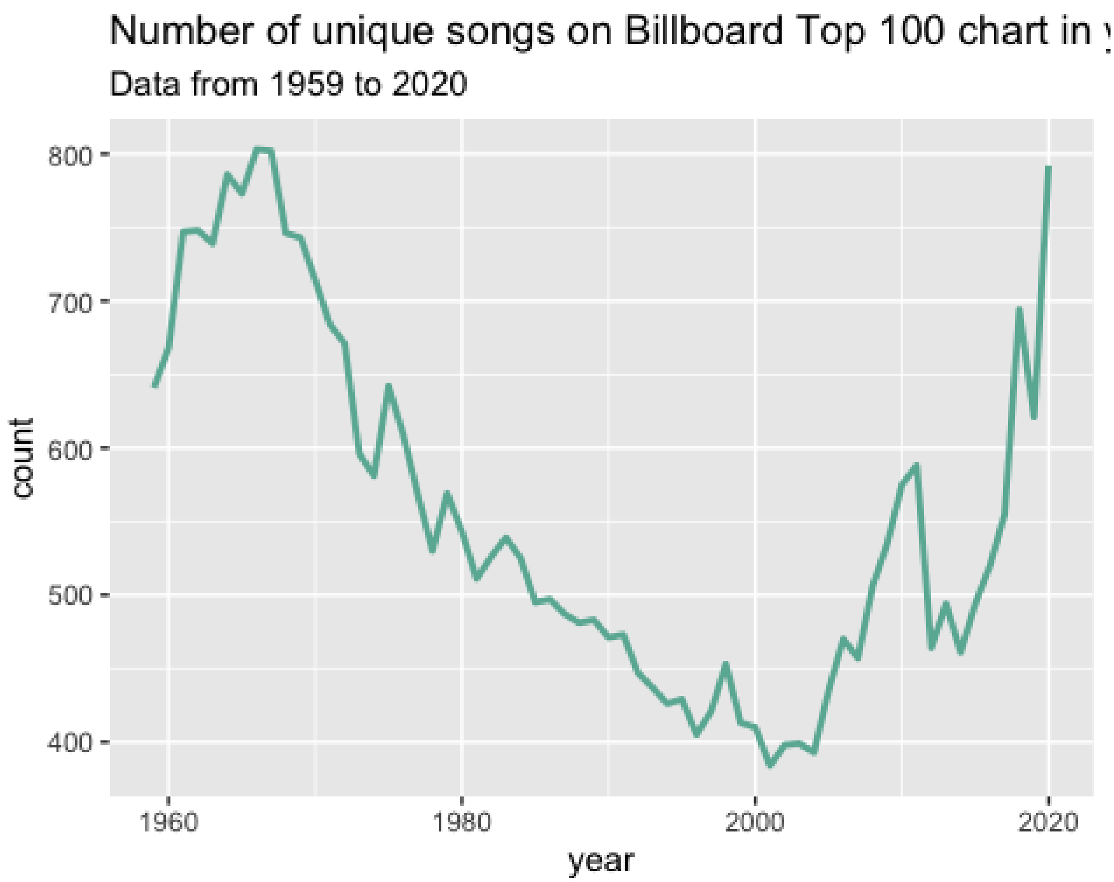
Part A

```
## # A tibble: 10 × 3
## # Groups:   performer, song [10]
##    performer                                song
count
##    <chr>                                    <chr>
<int>
##  1 Imagine Dragons                          Radioactive
87
##  2 AWOLNATION                               Sail
79
##  3 Jason Mraz                               I'm Yours
76
##  4 The Weeknd                               Blinding Lights
76
##  5 LeAnn Rimes                              How Do I Live
69
##  6 LMFAO Featuring Lauren Bennett & GoonRock Party Rock Anthem
68
##  7 OneRepublic                              Counting Stars
68
##  8 Adele                                    Rolling In The Deep
65
##  9 Jewel                                    Foolish Games/You Were Meant…
65
## 10 Carrie Underwood                         Before He Cheats
64
```

This table displays the top ten most popular songs since 1958, measuring by the total number of weeks that this song spent on the Billboard Top 100. The song's title, its performer, and total number of weeks are displayed in this table in descending order. The top ten most popular songs are Radioactive, Sail, I'm Yours, Blinding Lights, How Do I Live, Party Rock Anthem, Counting Stars, Rolling In the Deep, Foolish Games/You Were Meant For Me, and Before He Cheats.

Part B

```
## # A tibble: 62 × 2
##     year count
##    <int> <int>
##  1  1959   641
##  2  1960   668
##  3  1961   747
##  4  1962   748
##  5  1963   739
##  6  1964   786
##  7  1965   773
##  8  1966   803
##  9  1967   802
## 10  1968   746
## # … with 52 more rows
```

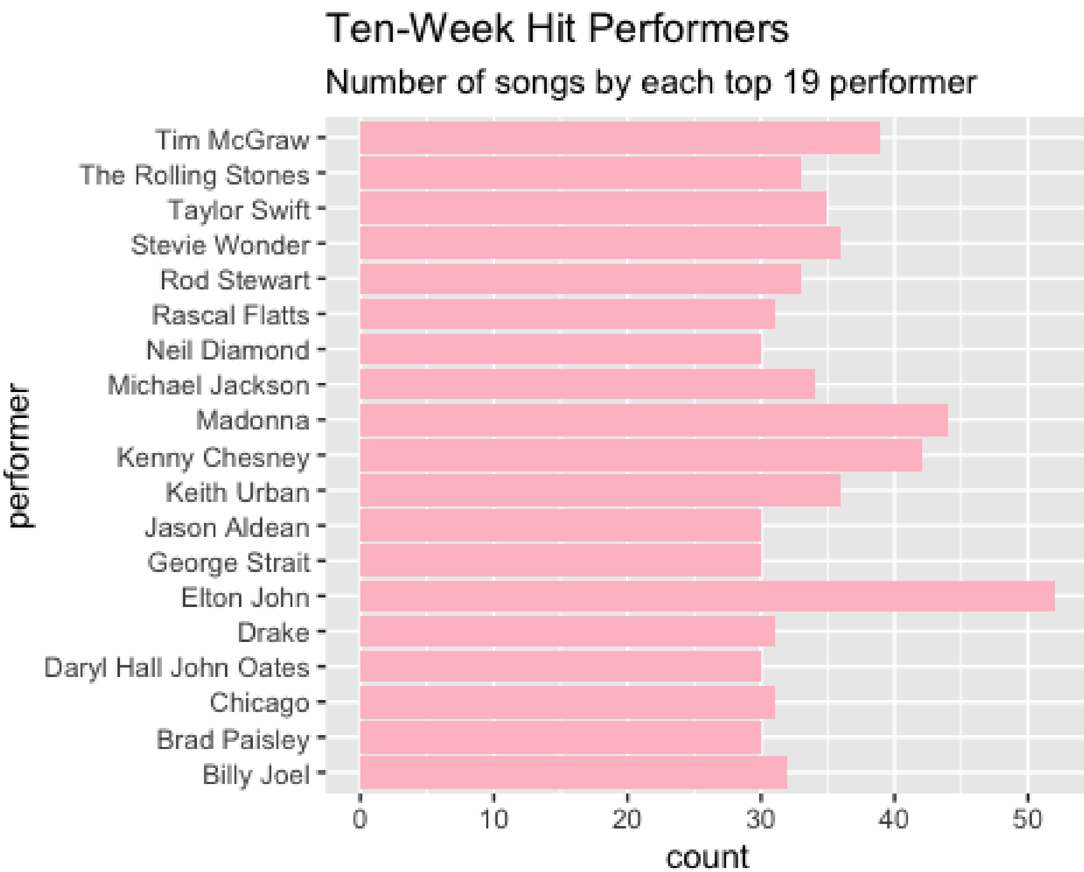


This figure shows the number of unique songs on Billboard Top 100 chart each year, and it displays the trend over years from 1959 to 2020. From the graph, we observe that in late 1960s, the musical diversity has reached at a peak. After late 1960s, the musical diversity started to decrease; from 1970 to 2000 approximately, people's favorite songs tend to be less diverse. Starting from early 2000s, musical diversity emerged again and persisted until now.

Part C

```
## # A tibble: 6,126 × 2
##    performer                                count
##    <chr>                                    <int>
##  1 "? (Question Mark) & The Mysterians"         2
##  2 "'N Sync"                                    8
##  3 "'N Sync & Gloria Estefan"                   1
##  4 "'N Sync Featuring Nelly"                    1
##  5 "'Til Tuesday"                               3
##  6 "\"Groove\" Holmes"                          1
##  7 "\"Little\" Jimmy Dickens"                   1
##  8 "\"Weird Al\" Yankovic"                      4
##  9 "10,000 Maniacs"                             5
## 10 "100 Proof Aged in Soul"                     2
## # … with 6,116 more rows
```

```
## # A tibble: 19 × 2
##    performer              count
##    <chr>                  <int>
##  1 Billy Joel                32
##  2 Brad Paisley              30
##  3 Chicago                   31
##  4 Daryl Hall John Oates     30
##  5 Drake                     31
##  6 Elton John                52
##  7 George Strait             30
##  8 Jason Aldean              30
##  9 Keith Urban               36
## 10 Kenny Chesney             42
## 11 Madonna                   44
## 12 Michael Jackson           34
## 13 Neil Diamond              30
## 14 Rascal Flatts             31
## 15 Rod Stewart               33
## 16 Stevie Wonder             36
## 17 Taylor Swift              35
## 18 The Rolling Stones        33
## 19 Tim McGraw                39
```



This plot displays the ten-week hit performers and number of songs by each of these top 19 performers. From the graph, we observe that Elton John has the most songs on board of more than 50 songs, following by Madonna, Kenny Chesney, and Tim McGraw. Other performers have approximately 30~40 songs on board.

Problem 3: Visual Story Telling Part 1: Green Buildings

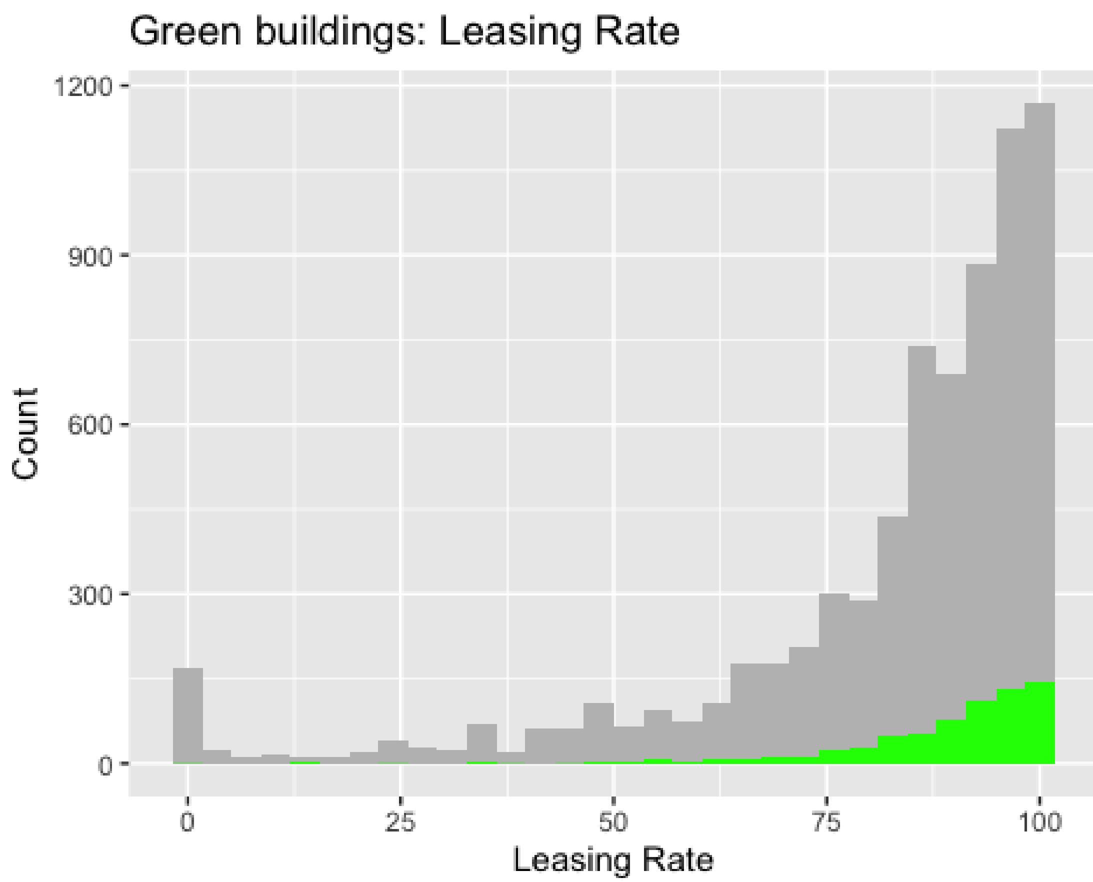
In this problem, our goal is to provide recommendation with solid analysis and insights to the developer of whether she should accept the stats guru's suggestion of paying extra 5% premium for a green certification. We would take the following steps to solve this question: * Provide visualized evidence on guru's suggestion * Visualize the data to determine any other correlation within the dataset * Identify possible confounding variable affecting rent and green status

Looking at the stats guru's analysis, I do not agree with his conclusion with the evidence he provide. I think it is not sufficient to prove that greenbuildings have higher rent overall, as he only considers the simple relationship between rent and green rating and fails to prove that other factors don't directly associate with higher rent.

Leasing Rate

He first removed the outliers according to the leasing rate in the dataset. Let's first visualize the leasing rate. To better visualize the relationship between green buildings and non-green buildings, we will first split the dataset into two groups

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



From the plot, we observe that most green buildings focus on having higher leasing rate, while the leasing rate for non-green buildings are relatively unstable. Additionally, looking at the non-green buildings, we observe that there's a significant amount of data points with a leasing occupancy of lower than 10%. Let's find out how many data points exactly are in this bracket.

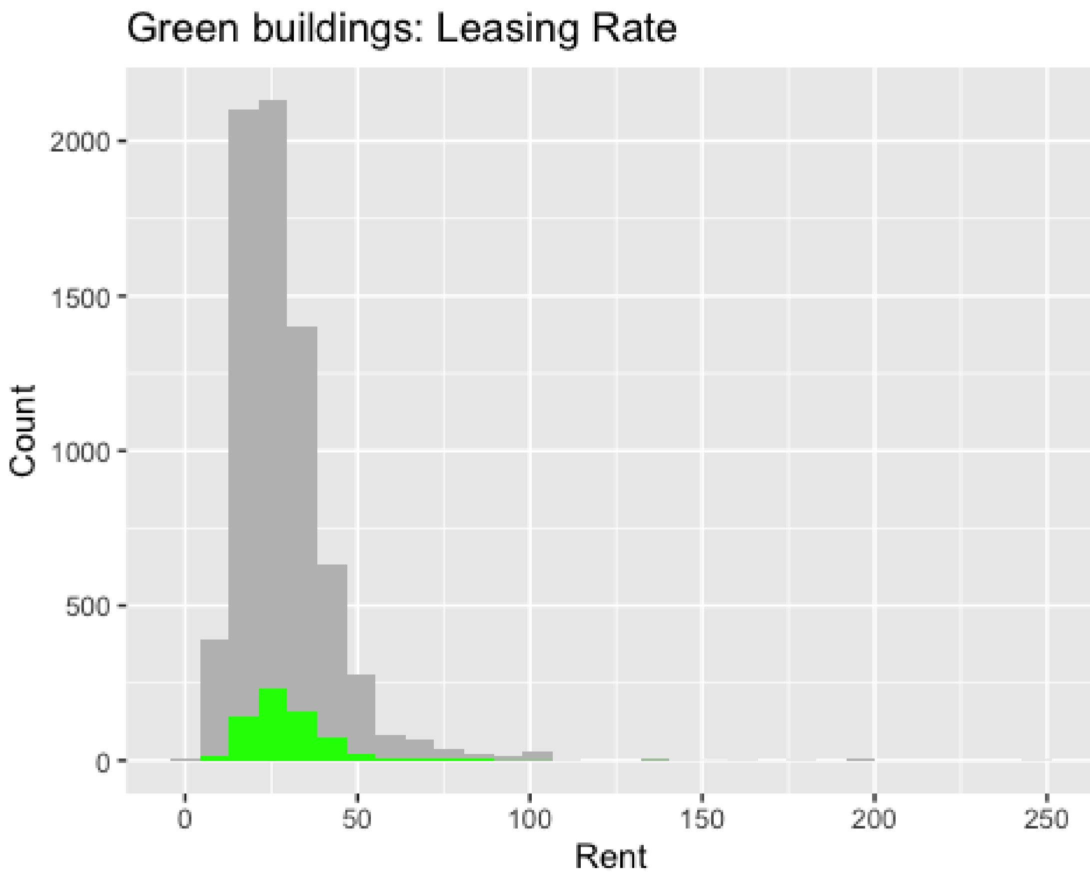
```
## [1] 215
```

There are 215 rows out of 7894 records with a leasing occupancy of lower than 10%. To avoid the possibility of distorting the information, we would include this part of data points in the following analysis.

Rent Distribution

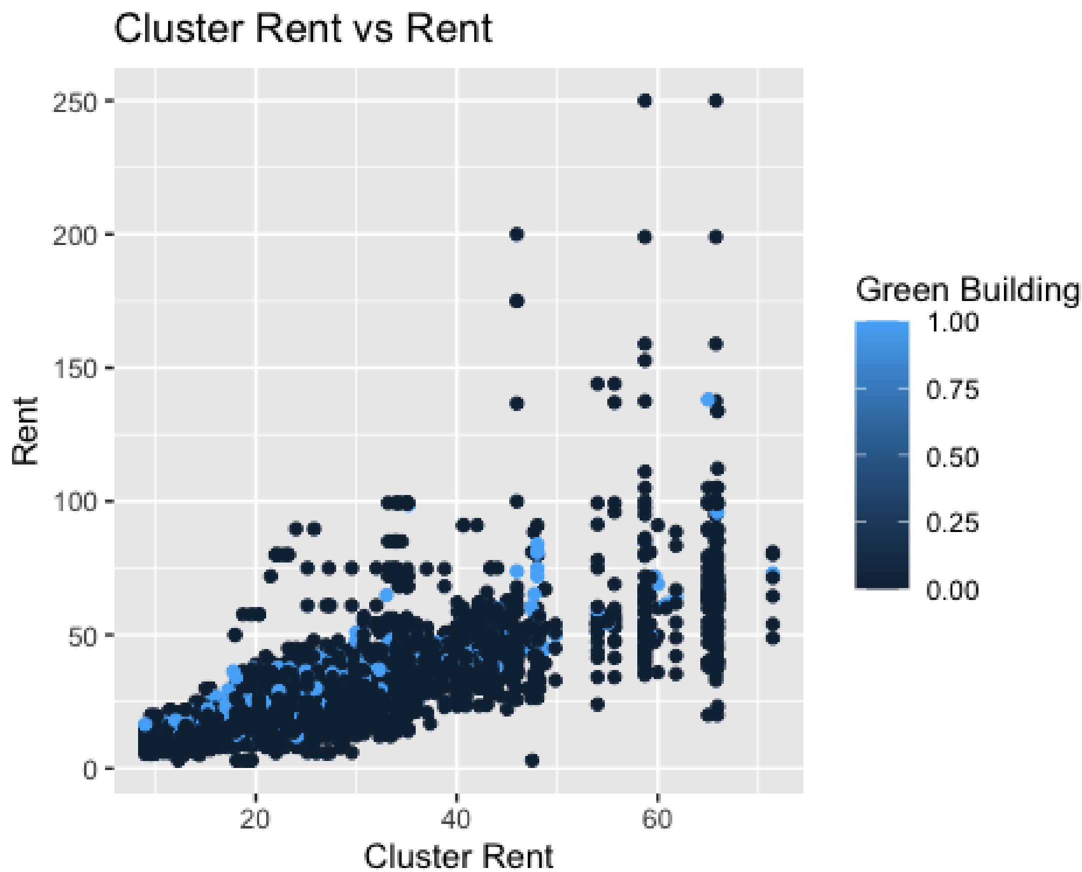
Next, we will visualize the rent distribution of green and non-green buildings.

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



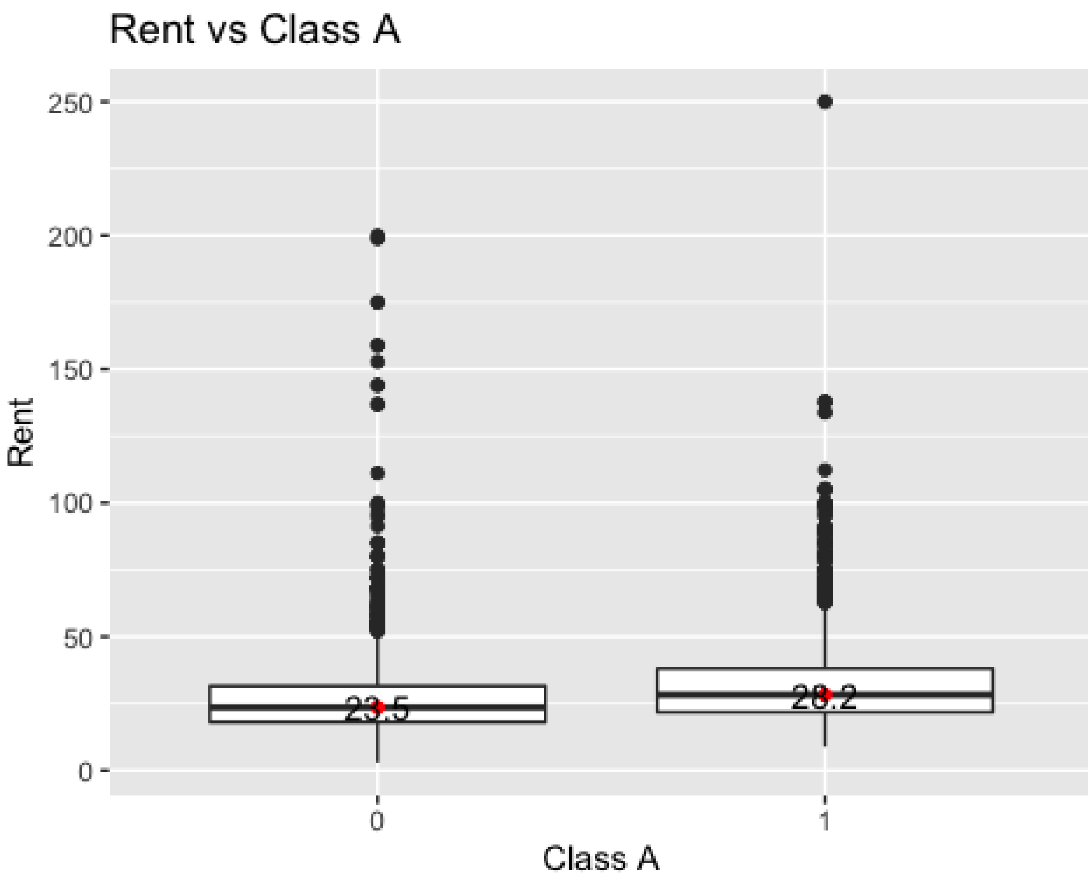
The stats guru compared the median rent for green and non-green buildings and concluded green buildings have a higher median market rent. However, as we observe from this graph, there are a lot of outliers with rent over $75. Since the sample size of greenbuildings is a lot smaller than non-green buildings, there isn't enough evidence to prove green buildings have higher rent than non-green buildings in general. Moreover, since there are many other factors in the dataset, we need to examine the possibility of confounding variables in the relationship between rent and green status.

We will now visualize some relationship between rent and other variables in the dataset to find possible factors affecting rent: #### Cluster

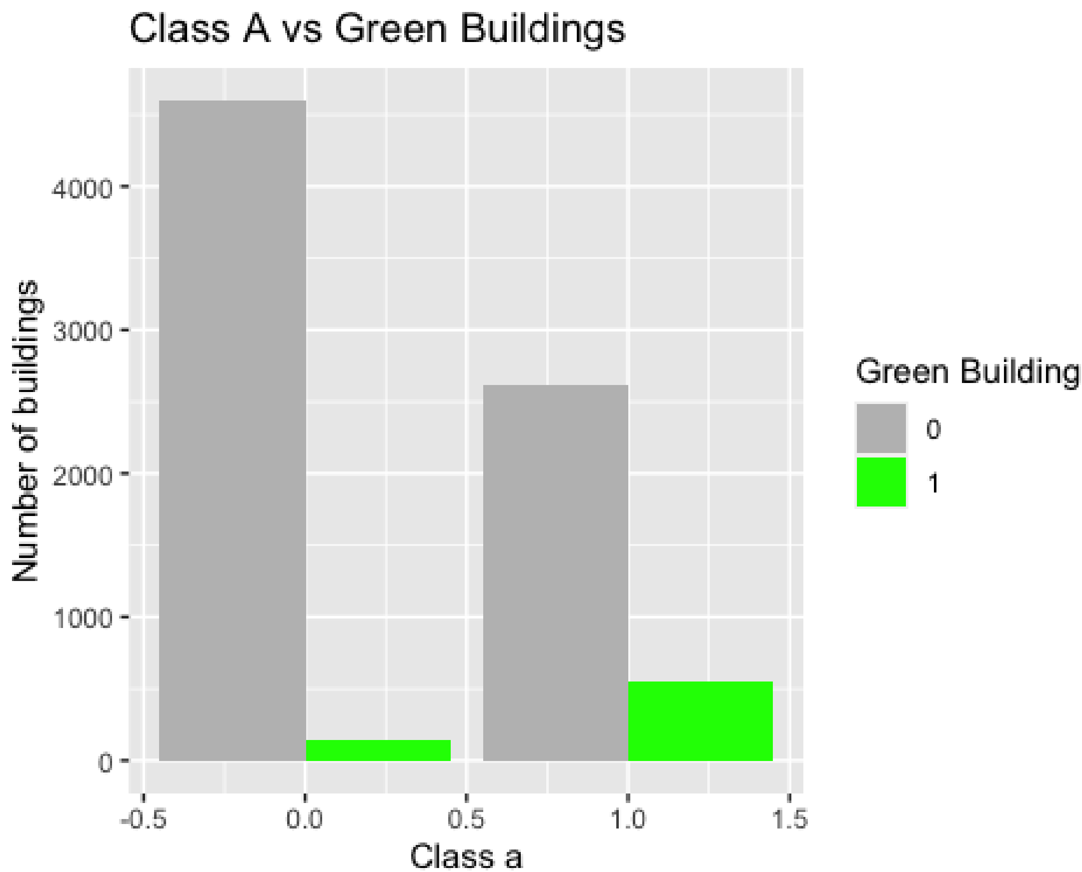


According to the graph, cluster rent is correlated to rent for both green and non-green buildings. For buildings with higher cluster rent, they tend to have higher rent. One possible reason is that the cluster is at a good location, such as near important highways, good school district, or commercial zone. Since the developer is constructing on East Cesar Chavez, just across I-35 from downtown, she can reference the cluster rent of this location.

Class

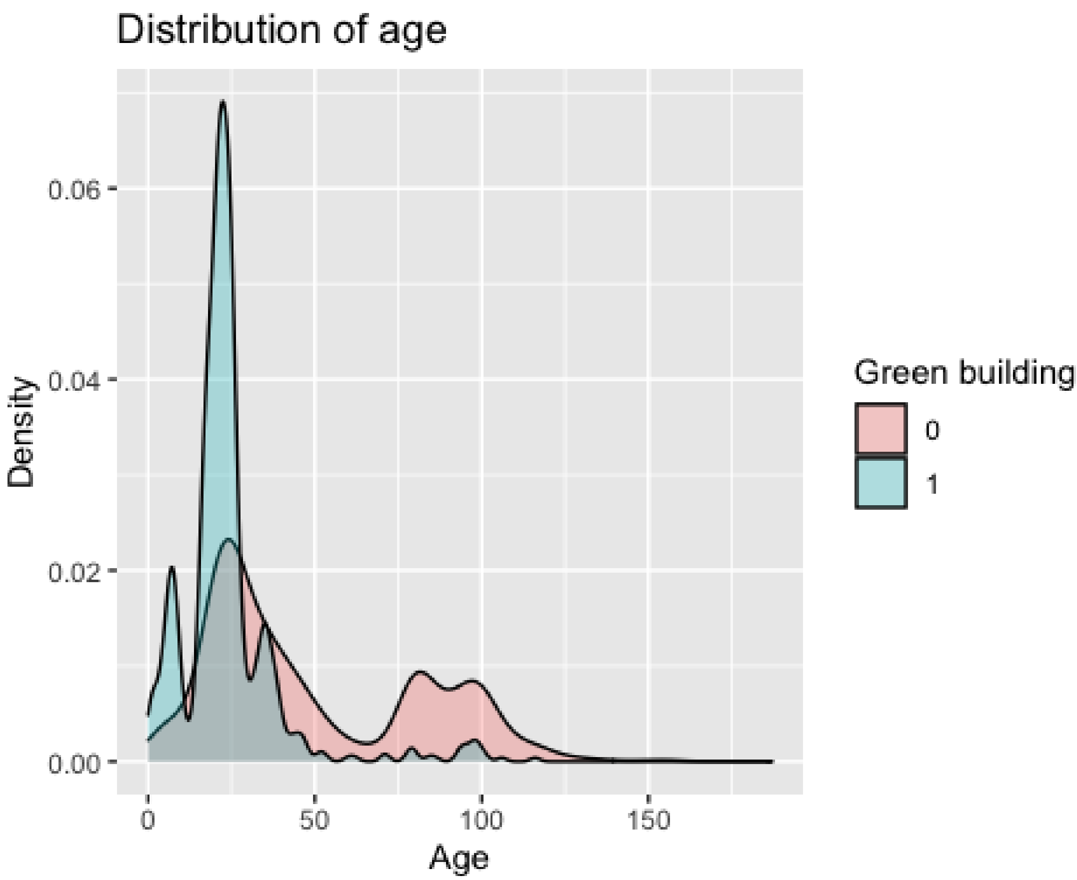


Class A buildings have a median rent of $28.2, and non-Class A buildings have a median rent of 23.5 dollars. Class A buildings is higher by 5 dollars than non-Class A buildings.

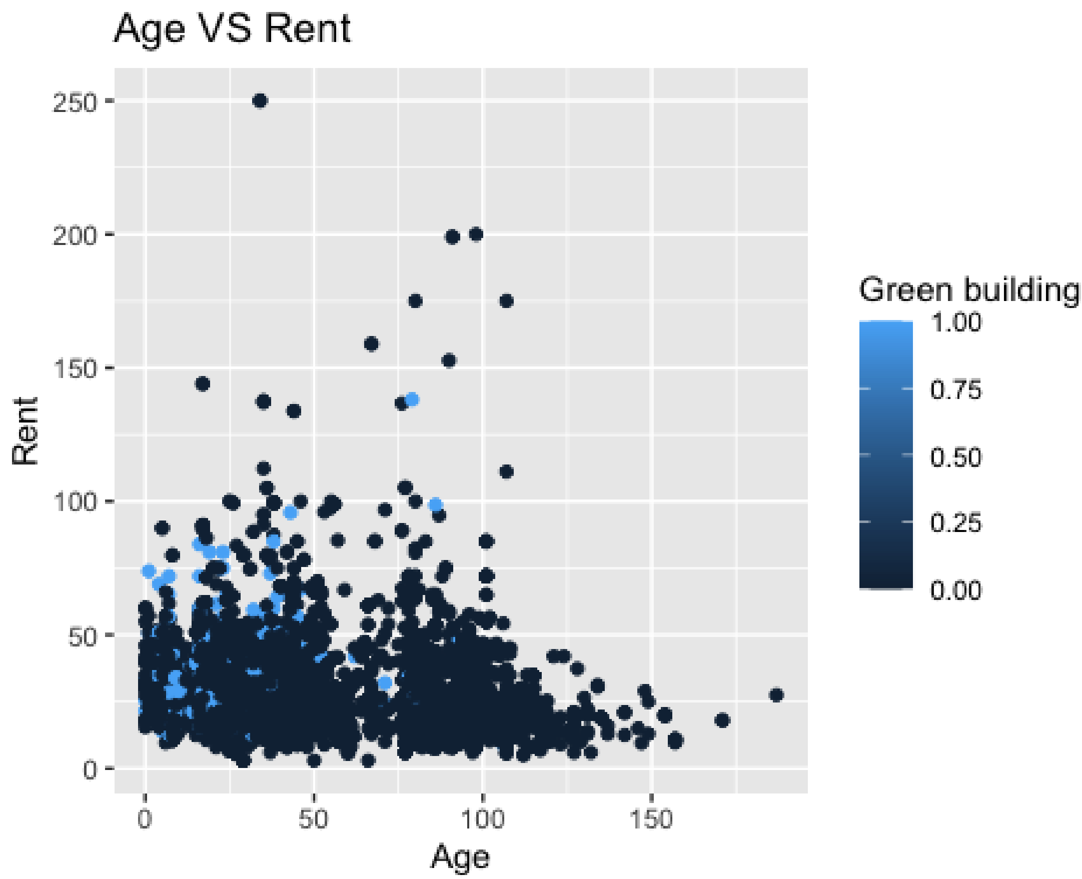


Looking at the relationship, more green buildings belong to Class A. In the prompt, the developer didn’t mention the class of the building. Since Class A buildings have higher rent in general, she should also focus on this feature.

Rent

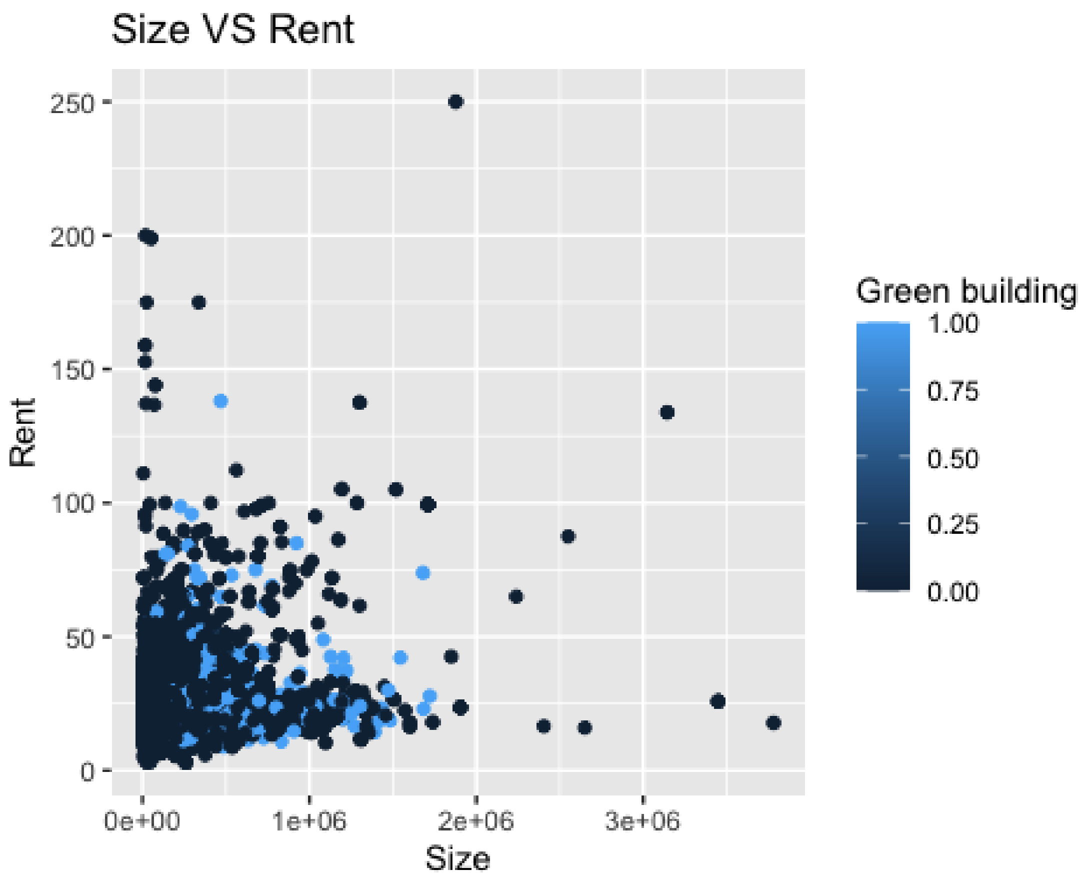


Looking at the distribution of age, most of the green buildings are younger than non-green buildings.



Although green buildings are younger in general, there isn't a clear trend showing the correlation between age and rent.

Size



Looking at the graph, rent is slightly correlated with size. Larger size can result in higher rent.

Insights

By observing the above confounding variables, it is hard to conclude that the increase in rent per square foot as analyzed by the stats guru is purely caused by the building's green rating. Thus, if we would to find out whether green rating truly leads to higher rent per square foot, we should hold the confounding variables of buildings with different green ratings at the same level. With all other variables held constant, we can compare the median rent and conclude whether green rating is the factor that cause the rent to be higher.

For example, for the Austin real-estate developer building, she should consider the rent for buildings with a size ranging from around 200K to 300K square feet.

```
## Median value for green buildings (20k-30k sqft)= 28.82

## Median value for non-green buildings (20k-30k sqft)= 27.95
```

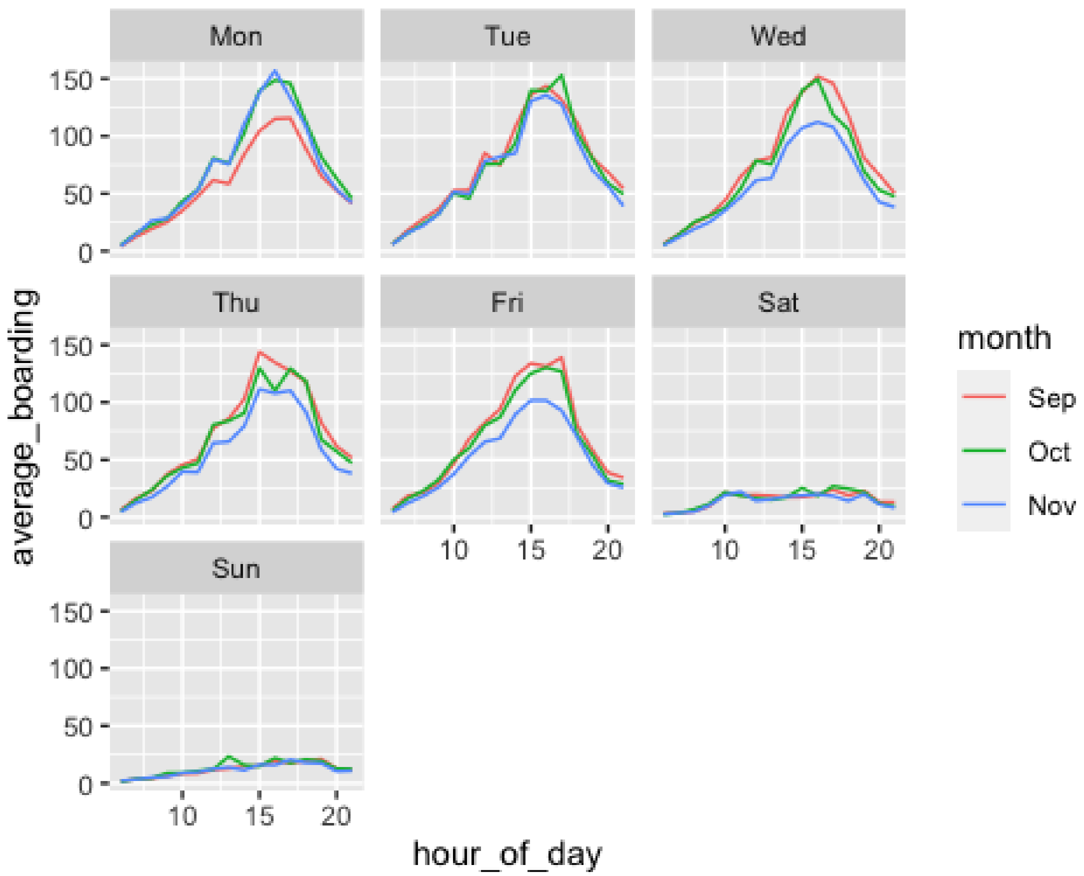
Green buildings' rent are at premium of approximately 1 dollar.

Conclusion

In conclusion, the guru's analysis is not accurate because he fails to include the effect of other confounding variables affecting rent. We would suggest that the developer focus on the location and cluster of the building and whether it can be a class-a building. Additionally, with the same range of size, green buildings are at a small amount of premium. After accounting all these confounding variables, the developer can decided whether it is worthy to pay the 5% premium for a green certificate.

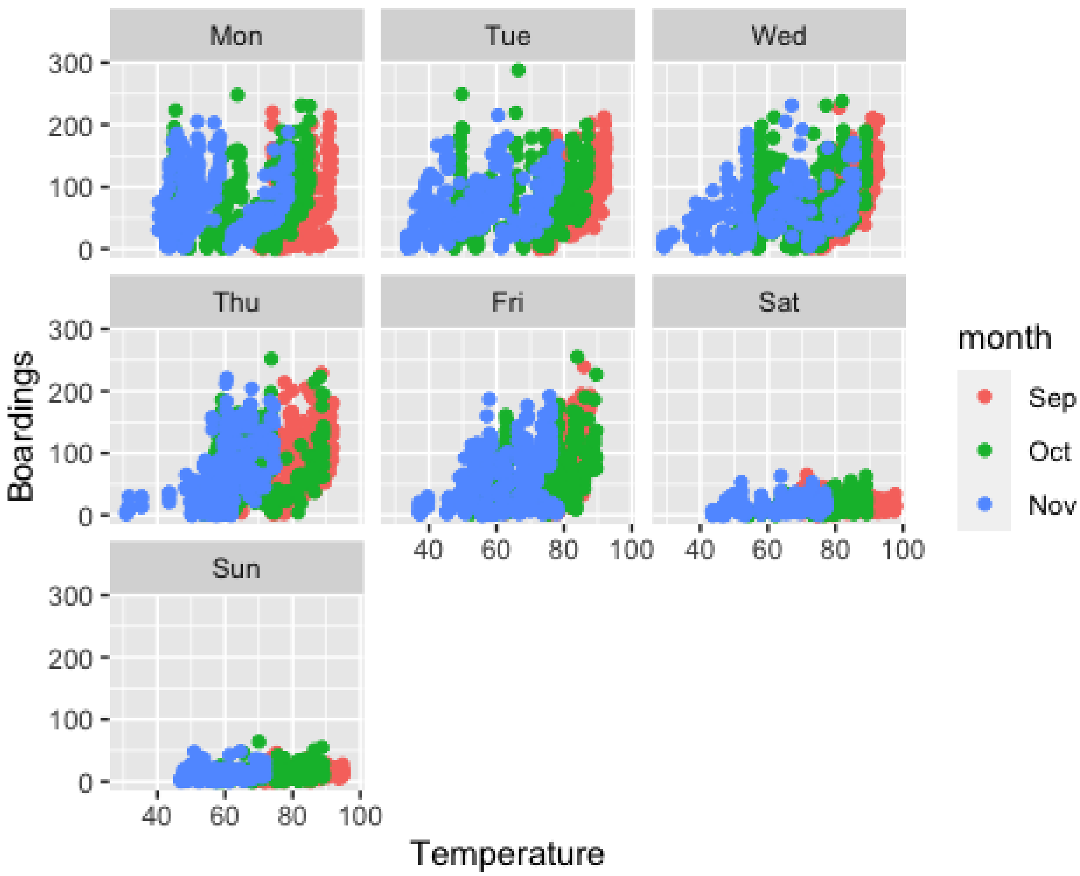
Problem 3: Visual Story Telling Part 2: Capital Metro Data

```
## `summarise()` has grouped output by 'hour_of_day', 'day_of_week'. You can
## override using the `.groups` argument.
```



This line graph represents the average number of people boarding any Capital Metro around UT in each 15-minute window throughout the day, grouped by month, and faceted by day of week. In this graph, the x-axis represents the specific hour of a day, the y-axis represents the average number of boarding. The graph is faceted by each day of the week, and each facet contains three lines representing different months. According to the graph, the peak hours for boarding are approximately the same for weekdays, which is 15 to 17 o'clock. There are less people on board on weekends compared with weekdays, but the peak hours over the weekend are approximately 19-20 o'clock. Moreover, we also noticed that the

September line for Monday November line on Wednesday, Thursday, and Friday are lower than the others. One possible explanation for this is that we have Labor Day and Thanksgiving holidays in these two months on these days. Many students are not commuting on these holidays. Since the y-axis is calculated by average, the overall mean is brought down by the lower number over the holidays.



This graph represent the total boarding number on any Capital Metro around UT based on temperature, grouped by month, faceted by day of week. The x-axis represents the temperature, the y-axis represents the total number of boarding. The graph is faceted by each day of the week, and each facet contains three different colors of points clusters representing different months. While holding other variables constant, temperature doesn't seem to affect the number of boarding. Therefore, temperature doesn't seem to be an important factor affecting number of students boarding. Still, the number of boarding is generally higher on Weekdays than on Weekends. Moreover, we observed an interesting patterns in these data points: while the temperature vs number of boardings relationship holds almost constant in each month, there are differences between general temperatures over different months. This is mainly due to the decrease in temperature from September to November.

Problem 4: Portfolio Modeling

```
## Loading required package: xts
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

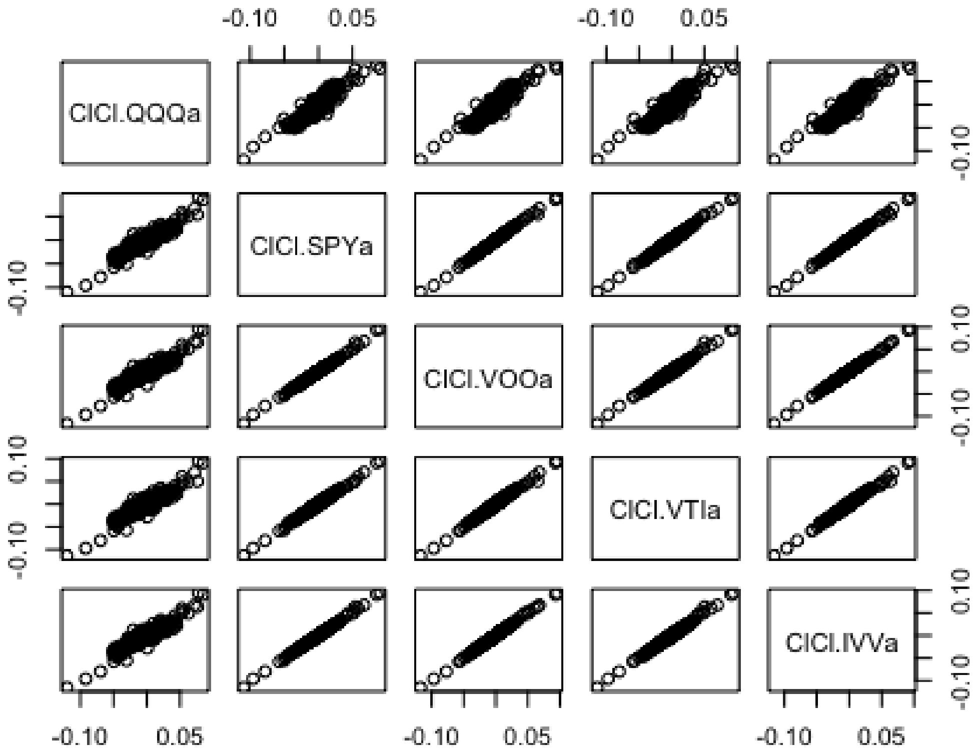
##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

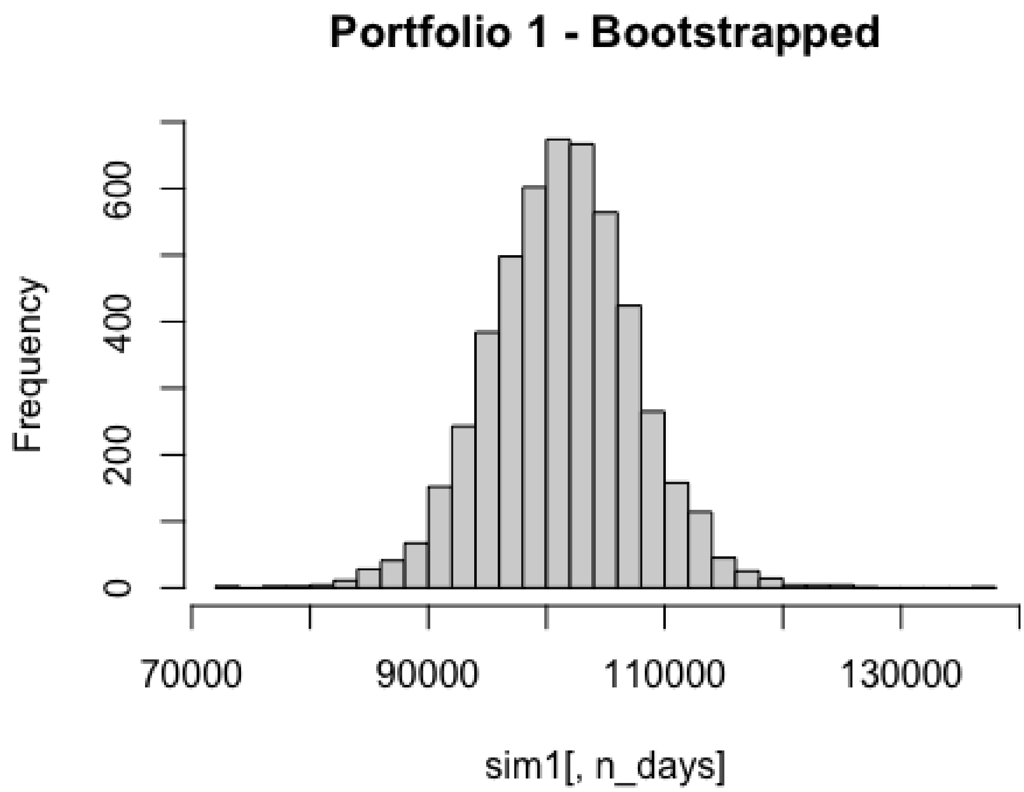
Portfolio 1: Safe Portfolio

The ETFs in portfolio 1 are all from large cap growth equity ETFs. These ETFs invest in growth company stocks that are believed to have a large market capitalization size, which means they are safer and more stable. * 20% SPY is considered one of the safest and largest ETFs. * 20% QQQ offers exposure to NASDAQ and has become one of the most popular exchange-traded products. * 20% VOO tracks S&P 500 Index, more diverse than most * 20% VTI attacts investors looking for simplified portforlio and minimized rebalancing obligations * 20% IVV tracks the S&P 500 Index, which includes many large and well known US firms; offers cheap and relatively balanced exposure to world’s largest companies.

```
##            SPY.Open SPY.High  SPY.Low SPY.Close SPY.Volume SPY.Adjusted
## 2017-08-14 225.1761 226.2763 225.1394  226.0471   73291900     226.0471
## 2017-08-15 226.4505 226.4689 225.6987  226.0196   55242700     226.0196
## 2017-08-16 226.5697 226.9915 225.9646  226.4139   56715500     226.4139
## 2017-08-17 225.7721 226.1021 222.8839  222.8839  128490400     222.8839
## 2017-08-18 222.7097 223.8925 222.0679  222.5355  136748000     222.5355
## 2017-08-21 222.4713 222.9847 221.7286  222.7097   65469700     222.7097
```

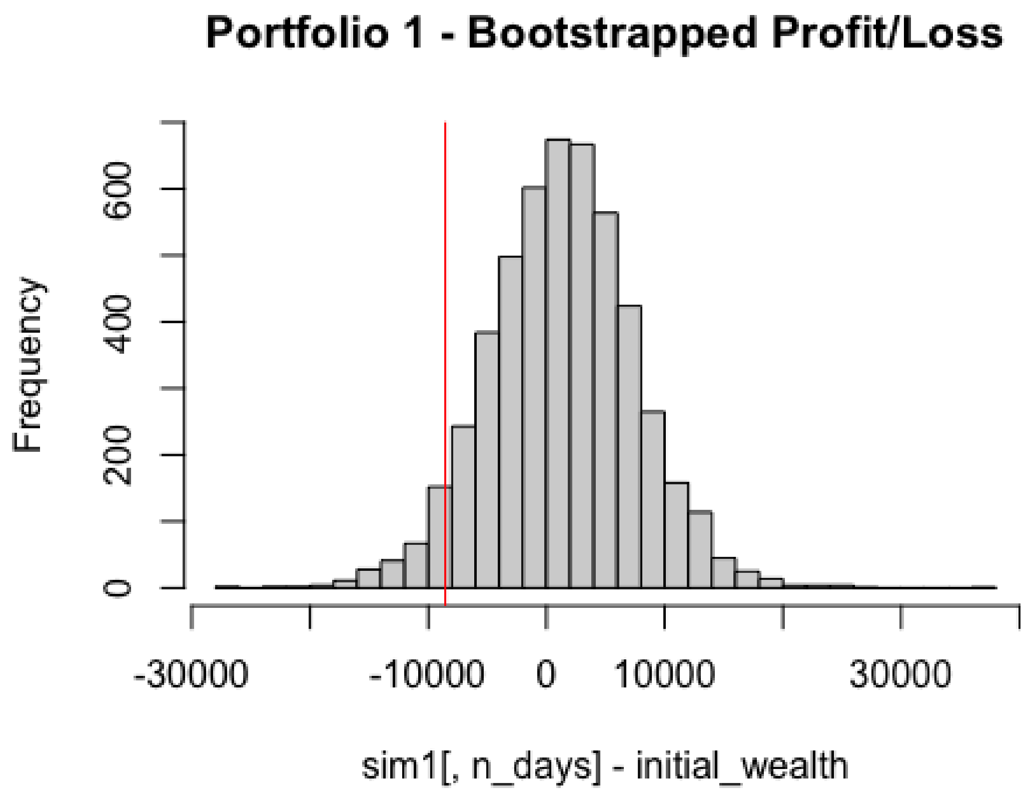


Looking at the correlation pair plots, we observe that all the ETFs are highly correlated. If one of them goes up, the other ones would go up as well. It implies that these ETFs are mostly likely following the market trend.




```
## [1] 101299.9

## [1] 1299.933
```




```
##        5%
## -8556.006
```

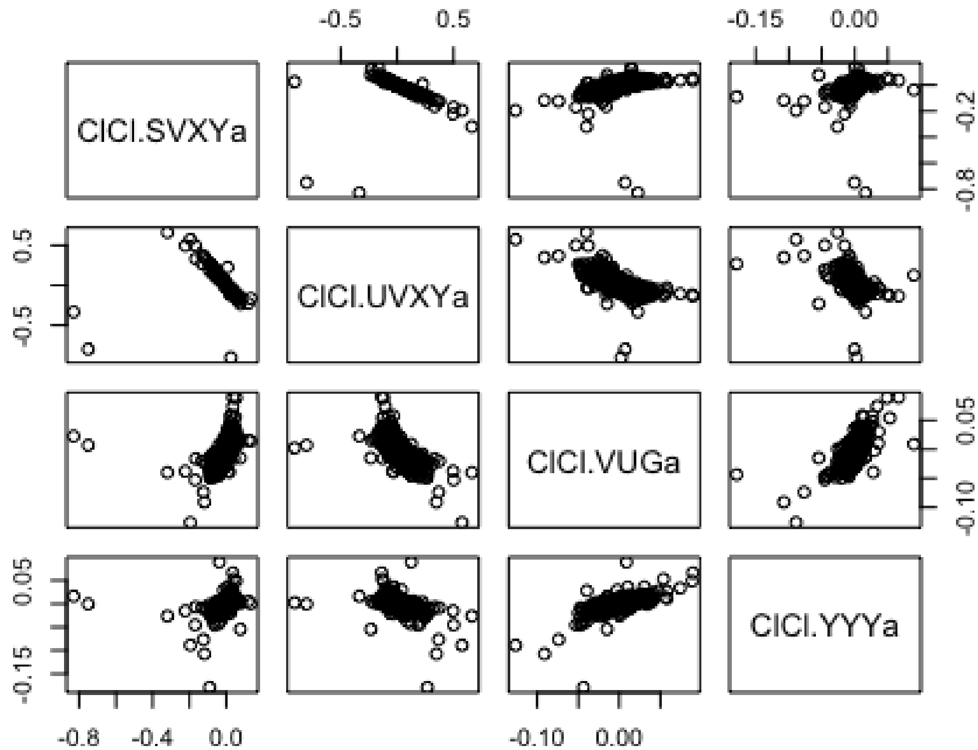
According to the histogram above (Profit/Loss), we observe that there is a chance for a loss after the 4-week period. However, with the possibility of loss, the mean earnings still result in a profit of **approximately $1299.93**. For this portfolio, the 4-week VaR at 5% level is **approximately -$8556.006**

Portfolio 2: Aggressive Portfolio

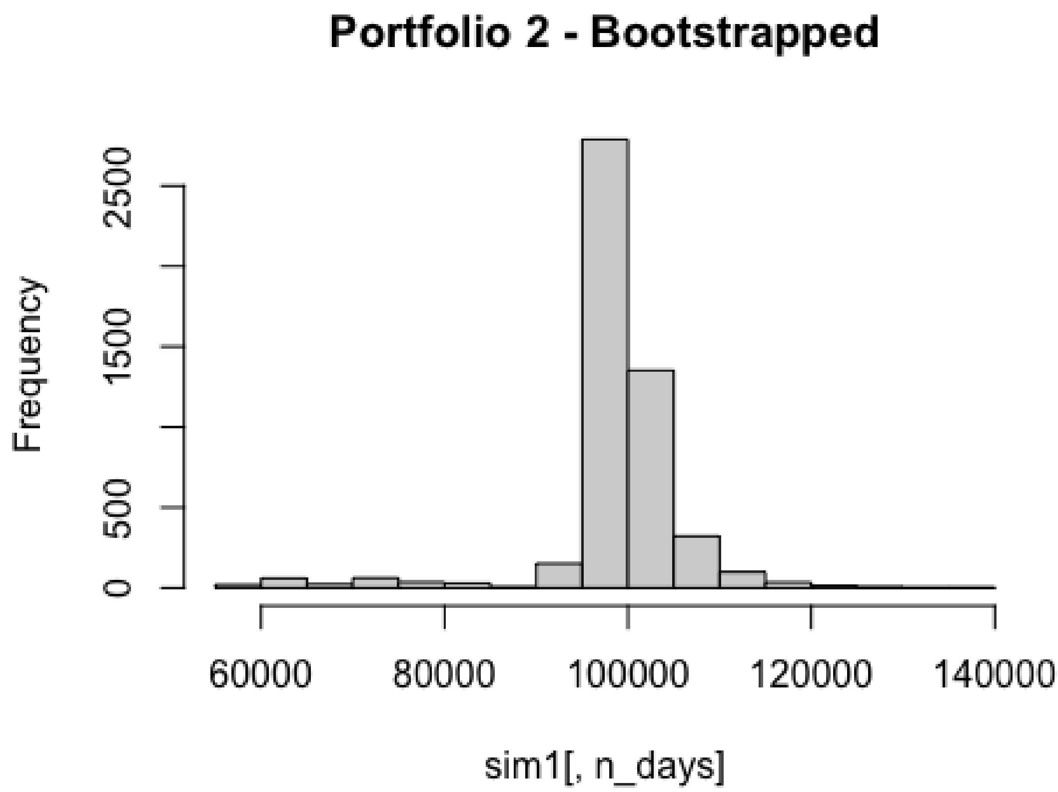
The ETFs in portfolio 2 takes a more aggressive approach in capturing market trends and profit from market volatility. * 25% SVXY * 25% UVXY * 25% VUG * 25% YYY

```
##            SVXY.Open SVXY.High SVXY.Low SVXY.Close SVXY.Volume SVXY.Adjust
ed
## 2017-08-14   1232.80   1295.20  1232.48    1293.12     2003575        323.
28
## 2017-08-15   1327.20   1327.20  1274.40    1299.52     1543700        324.
88
## 2017-08-16   1300.32   1319.68  1289.60    1309.28     1642775        327.
32
## 2017-08-17   1274.08   1292.16  1088.00    1091.84     5669800        272.
96
```

```
## 2017-08-18   1133.44   1186.56  1094.88    1130.56     4463550        282.
64
## 2017-08-21   1140.00   1178.56  1114.56    1175.52     2448350        293.
88
```

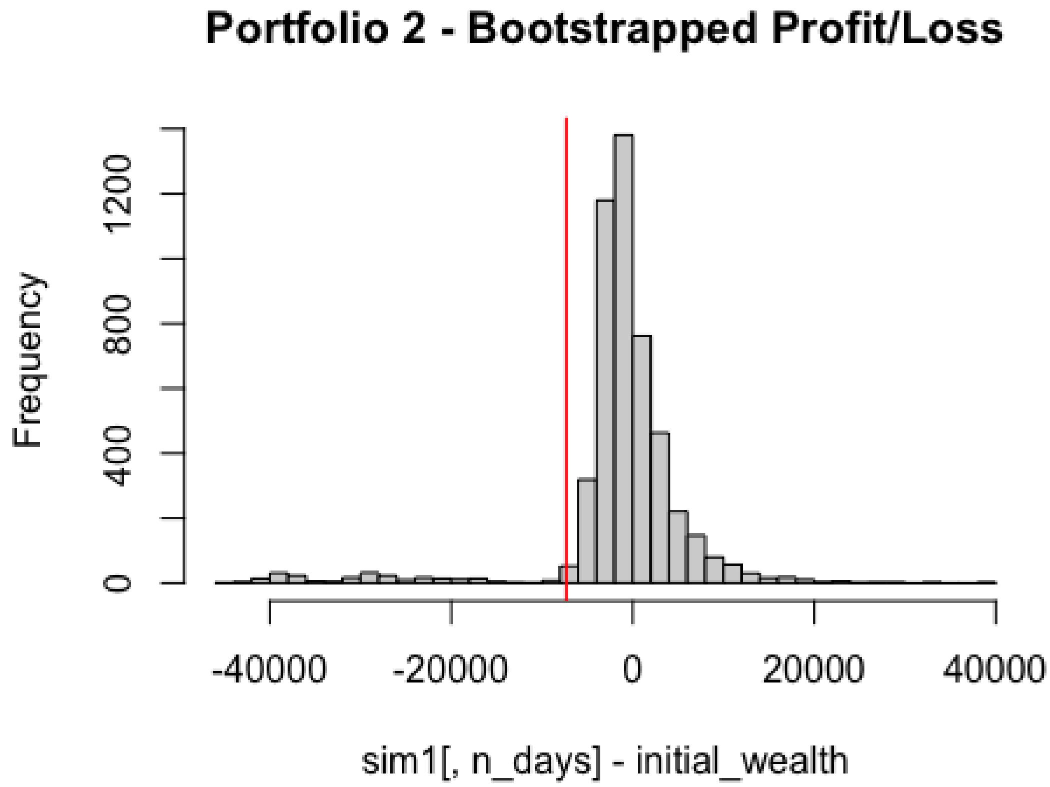


From this pair plot matrix, we observe that the ETFs in this portfolio is less correlated than the ones in portfolio 1. Thus, we would expect higher volatility in this portfolio compared with portfolio 1.




```
## [1] 98741.79

## [1] -1258.214
```

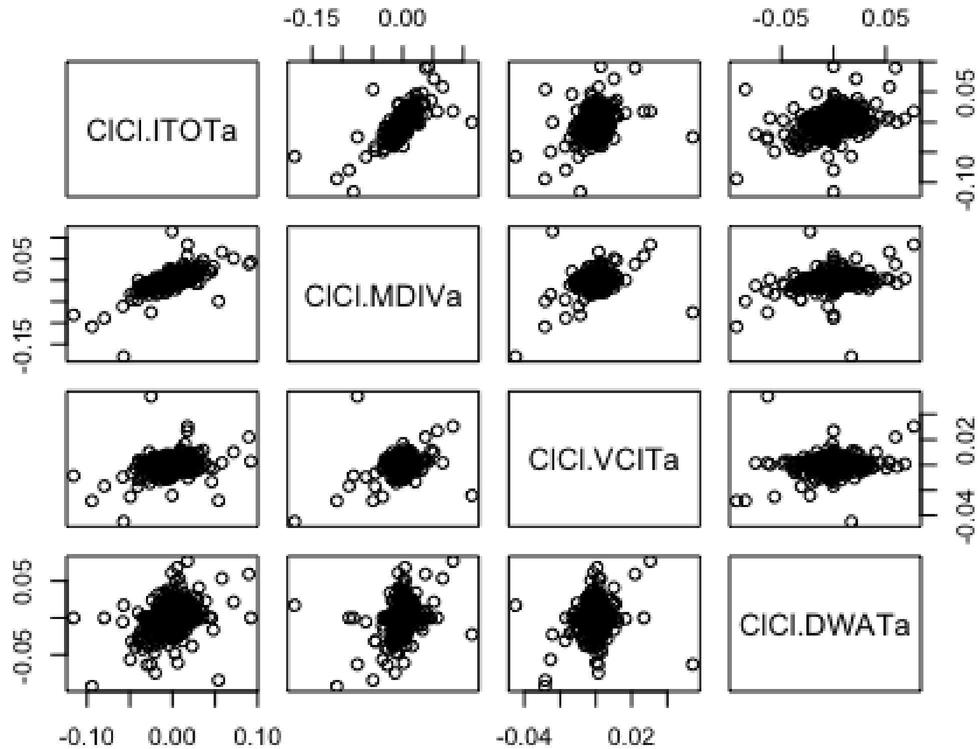



```
##        5%
## -7328.289
```

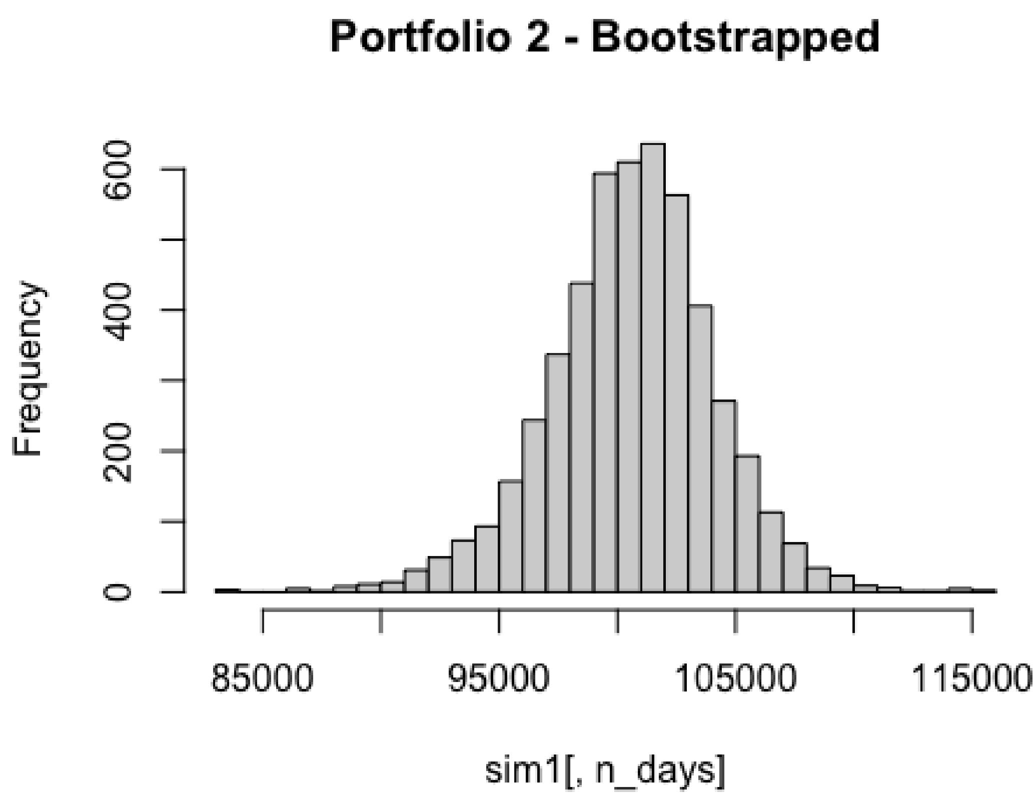
According to the histogram above (Profit/Loss), we observe that there is still a chance for a loss after the 4-week period. However, for this portfolio, the mean earnings seems to result in a loss of **approximately $1258.21**. In addition, for this portfolio, the 4-week VaR at 5% level is **approximately -$7328.29**, which is slightly lower than Portfolio 1. By observing the histogram, we can further analyze that, compared with the first portfolio, this portfolio is more likely to result in a loss. According the the plot, there seems to be a small amount of outlier on the left side of the graph, implying there is a greater chance for this portfolio to generate a loss between 20000 to 40000 dollars

Portfolio 3: Diversified Portfolio

The ETFs in portfolio 3 consider ETFs exposed to multiple asset classes to avoid the risk of market volatility, and we want to observe whether this portfolio can yield higher returns compared with safe ETFs. * 25% ITOT * 25% MDIV * 25% VCIT * 25% DWAT

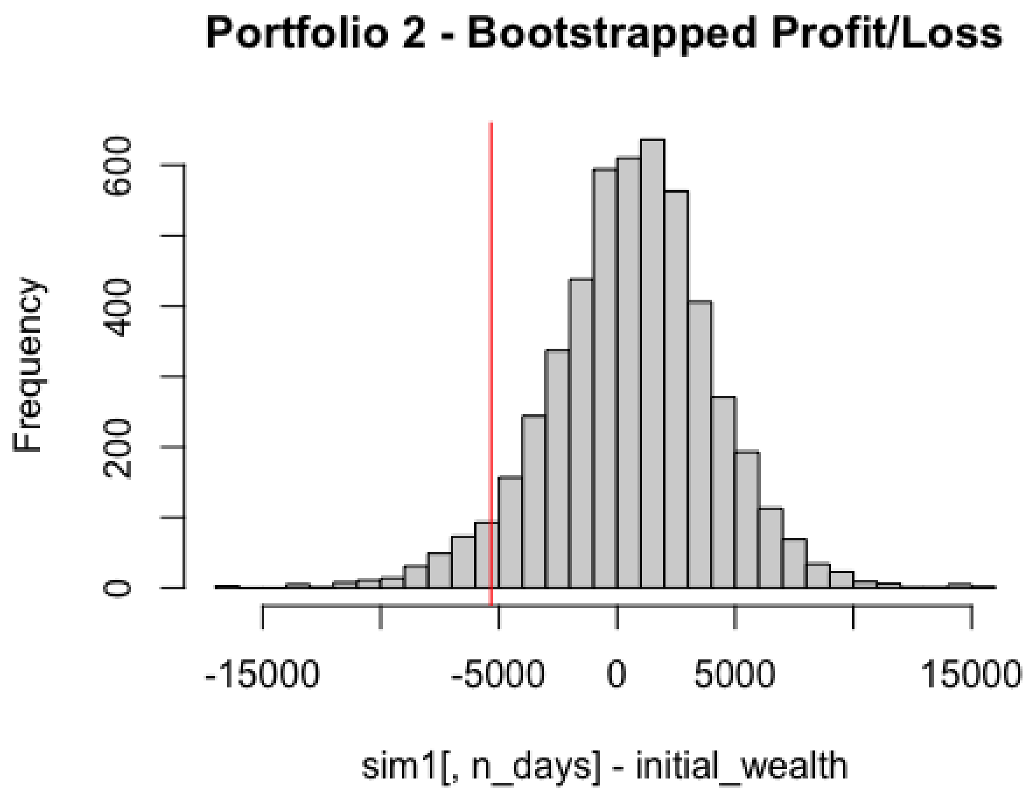


The ETFs in this portfolio are less correlated than the ones in portfolio 1 and portfolio 2. The coefficients for a few of relationships are almost horizontal.




```
## [1] 100599.5

## [1] 599.5008
```




```
##       5%
## -5336.995
```

According to the histogram above (Profit/Loss), we observe that there is still a chance for a loss after the 4-week period. However, for this portfolio, the mean earnings seems to result in a profit of **approximately $599.5**. Moreover, the range of this histogram becomes smaller, as it implies that the volatility of both profit and loss is smaller. Thus, this portfolio is not as risky as the previous two. In addition, for this portfolio, the 4-week VaR at 5% level is **approximately -$5336.995**, which is the lowest among three portfolio. Although the potential profit is not very high, this portfolio has the least risks among all three.

Conclusion

- Portfolio 1 results in a possible profit of approximately 1299.93. The 4-week VaR at 5% level is approximately -8556.006
- Portfolio 2 results in a possible loss of approximately 1258.21. The 4-week VaR at 5% level is approximately -7328.29
- Portfolio 3 results in a possible profit of approximately 599.5. The 4-week VaR at 5% level is approximately -5336.995 Portfolio 2 seems to be the riskest portfolio among all, since it has a more volatile profit/loss histogram, and the graph is more screwing towards the loss side. Portfolio 1 has the highest value of 4-week VaR at 5% level, meaning there's a 0.05 probability that the portfolio will fall in value by

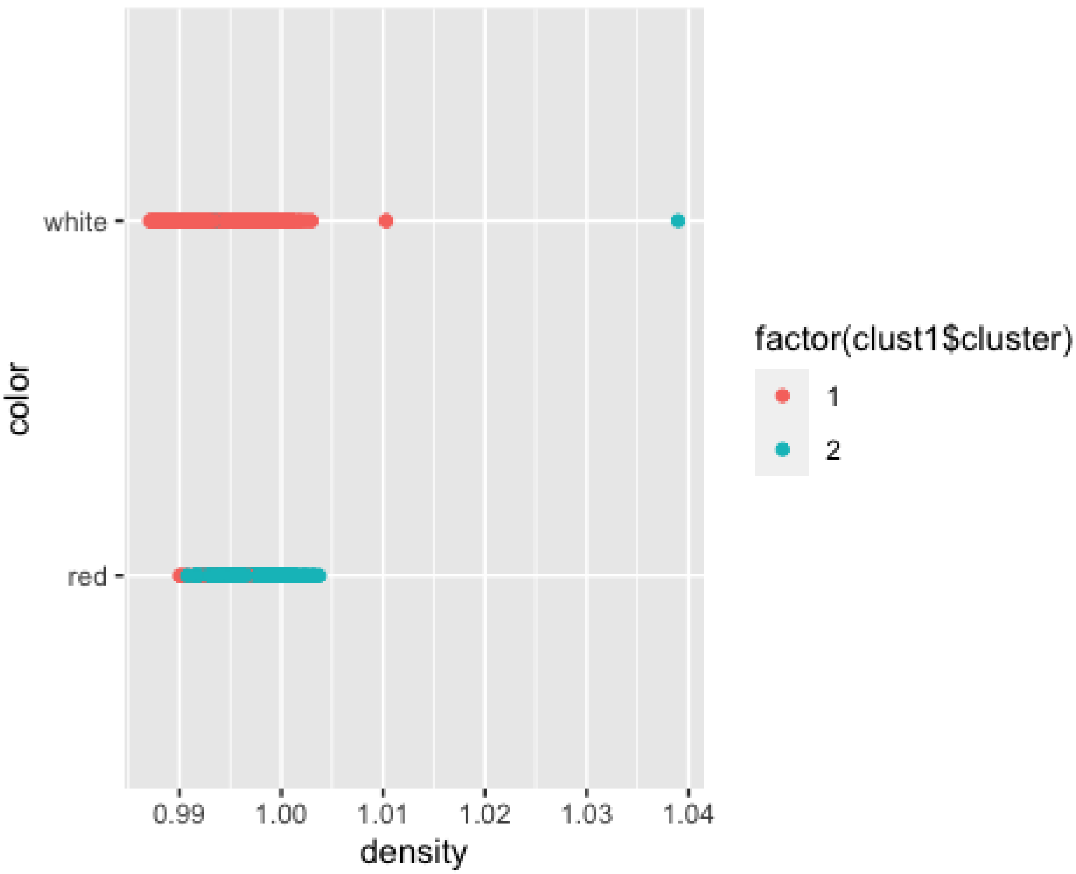
more than 8000 if there's no trading. Compared with portfolio 1, although portfolio 3 has a relatively lower possible profit, the VaR is much smaller than that of portfolio 1. The possible reason for this trend is that Portfolio 3 was able diversify the market risk by soothing out high volatility and capture the profits at the same time. We would suggest that investors diversify their portfolio to balance risks and returns.

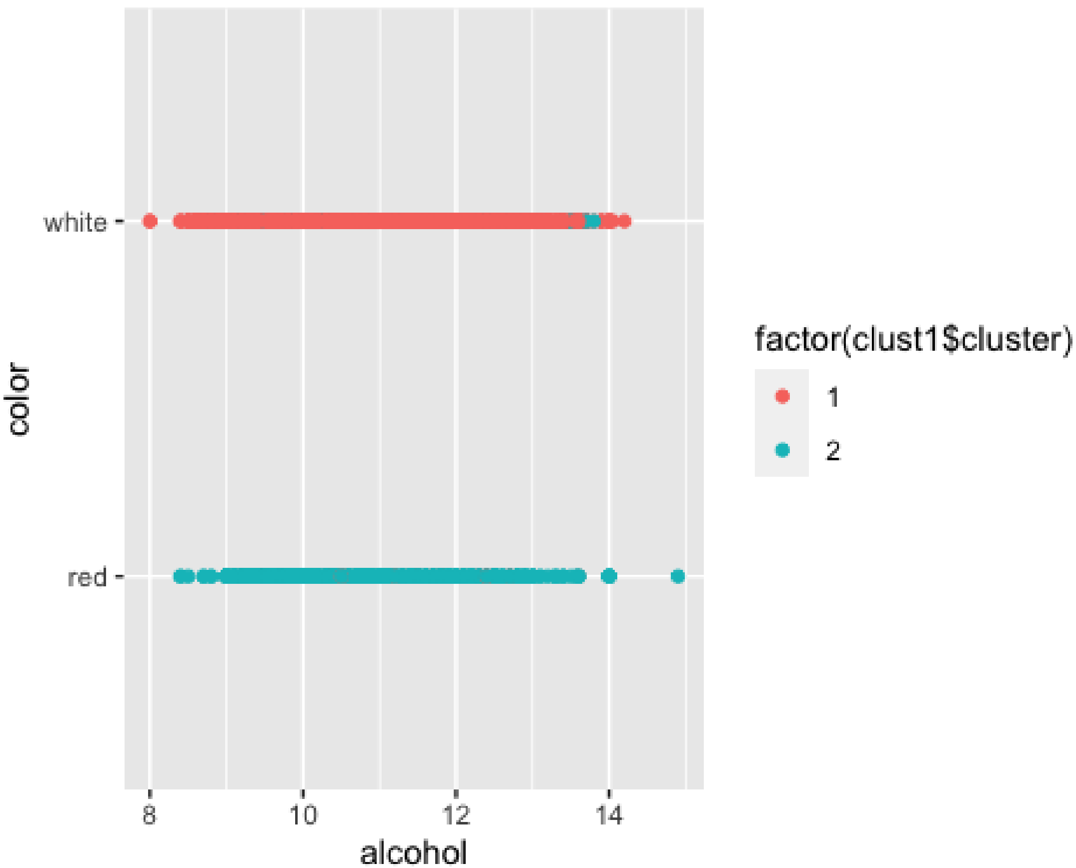
Problem 5: Clustering and PCA

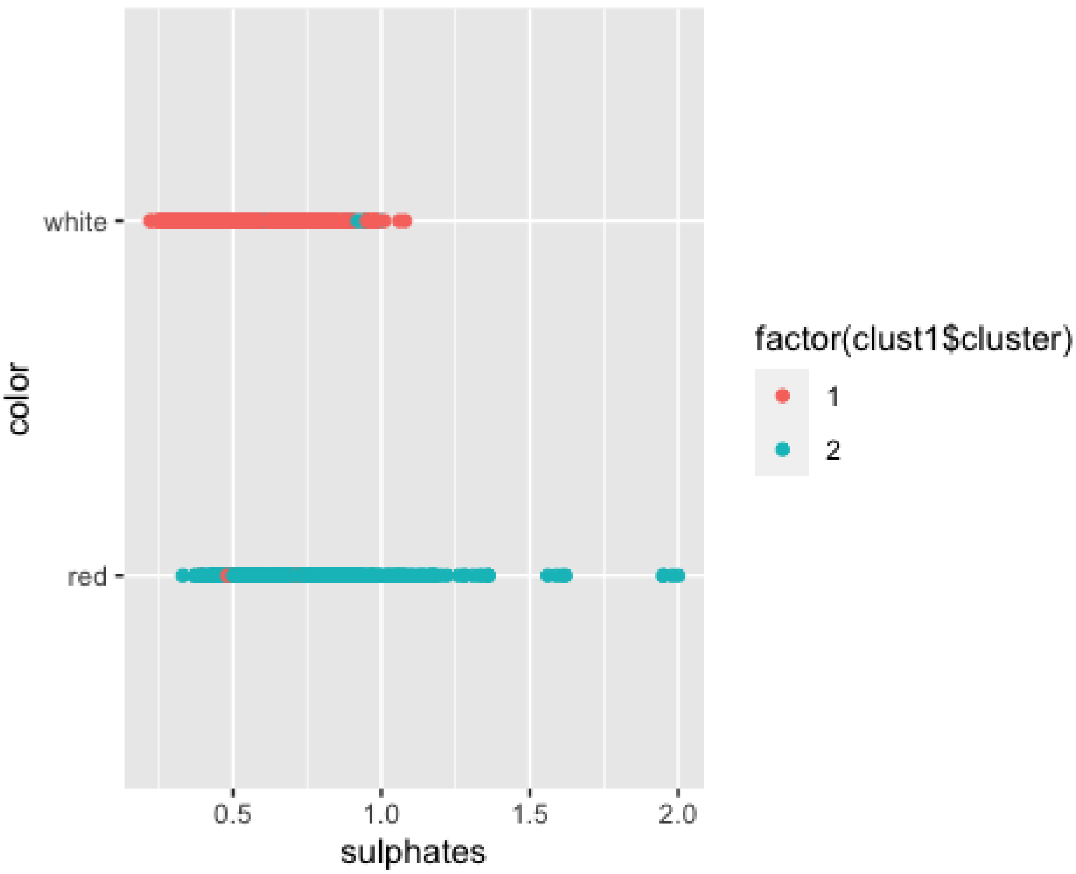
In order to analyze based on the 11 chemical properties, we created a new dataframe with only the 11 variables, excluding the last two outcome variables. #### Clustering

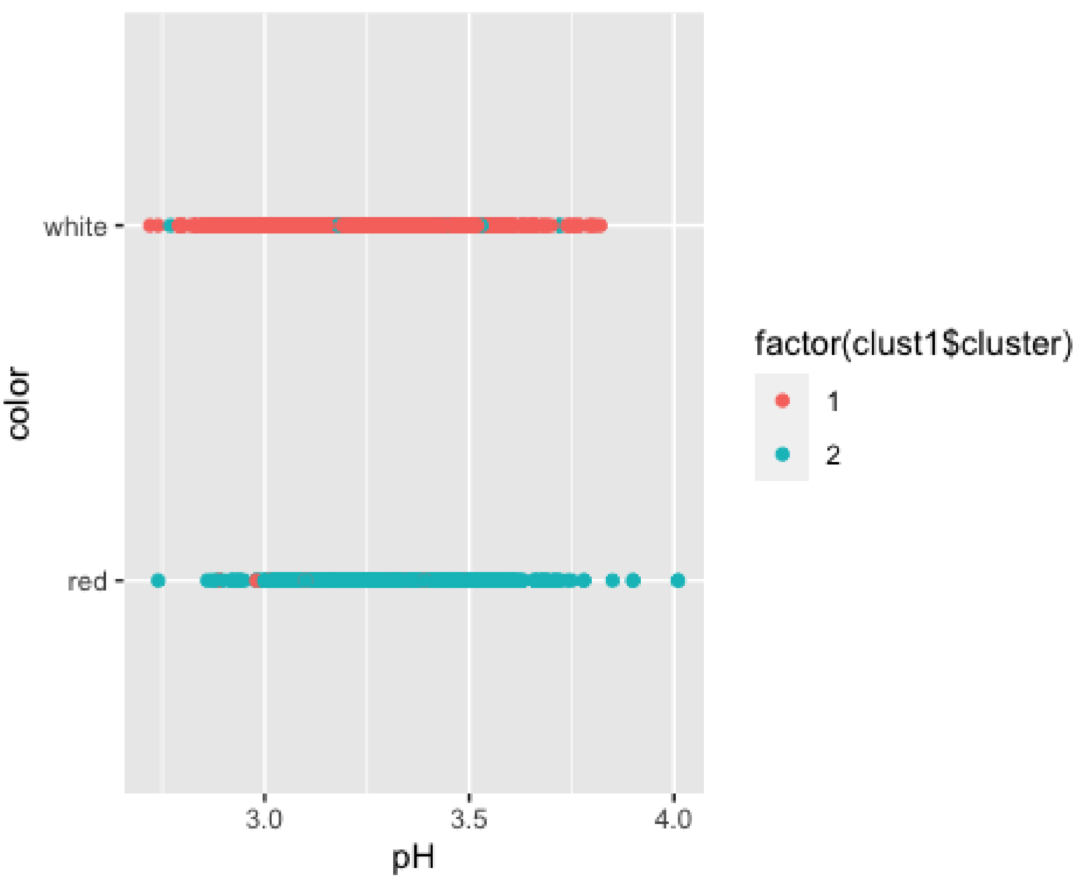
First, before actually running any clutering models, we scaled and centered the data. Then, we clustered the dataset into two parts, running k-means with 2 clusters and 20 starts to examine the **color of wine**.

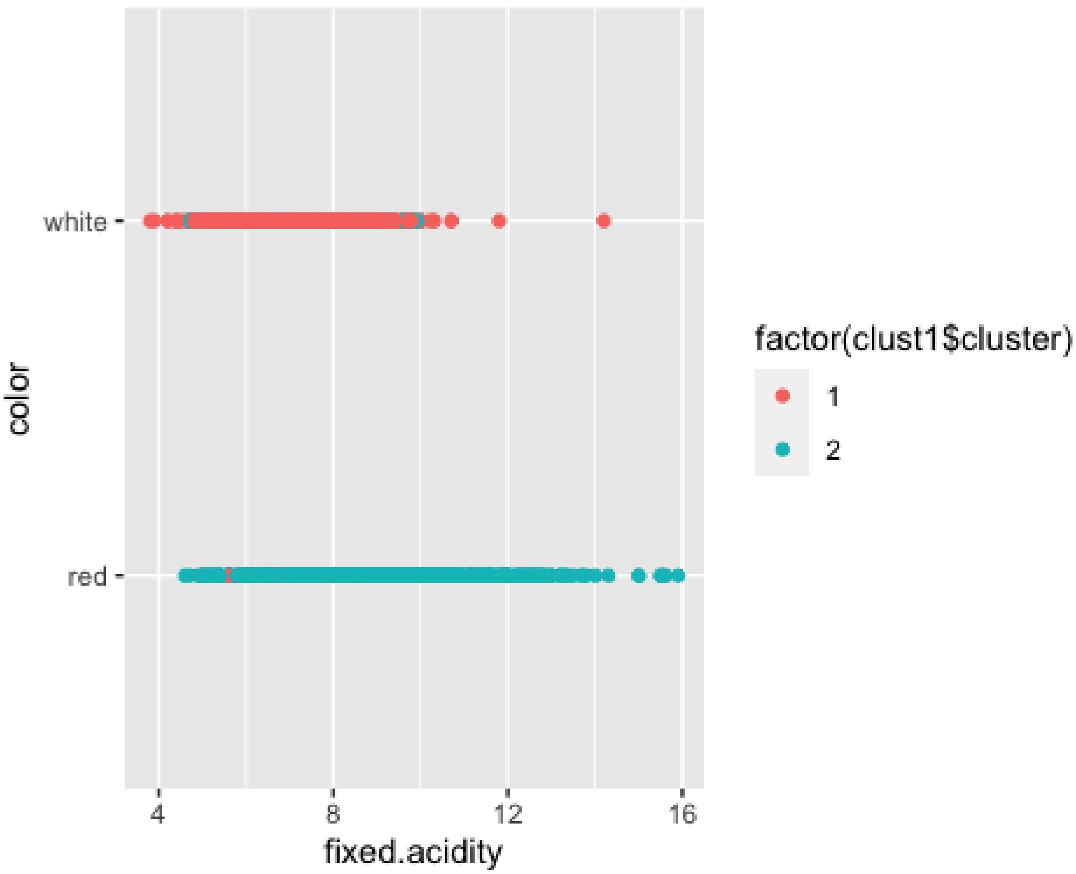
```
##                wine$color
## clust1$cluster  red white
##              1   24  4830
##              2 1575    68
```


white
red
color
0.99
1.00
1.01
1.02
1.03
1.04
density
factor(clust1$cluster)
1
2

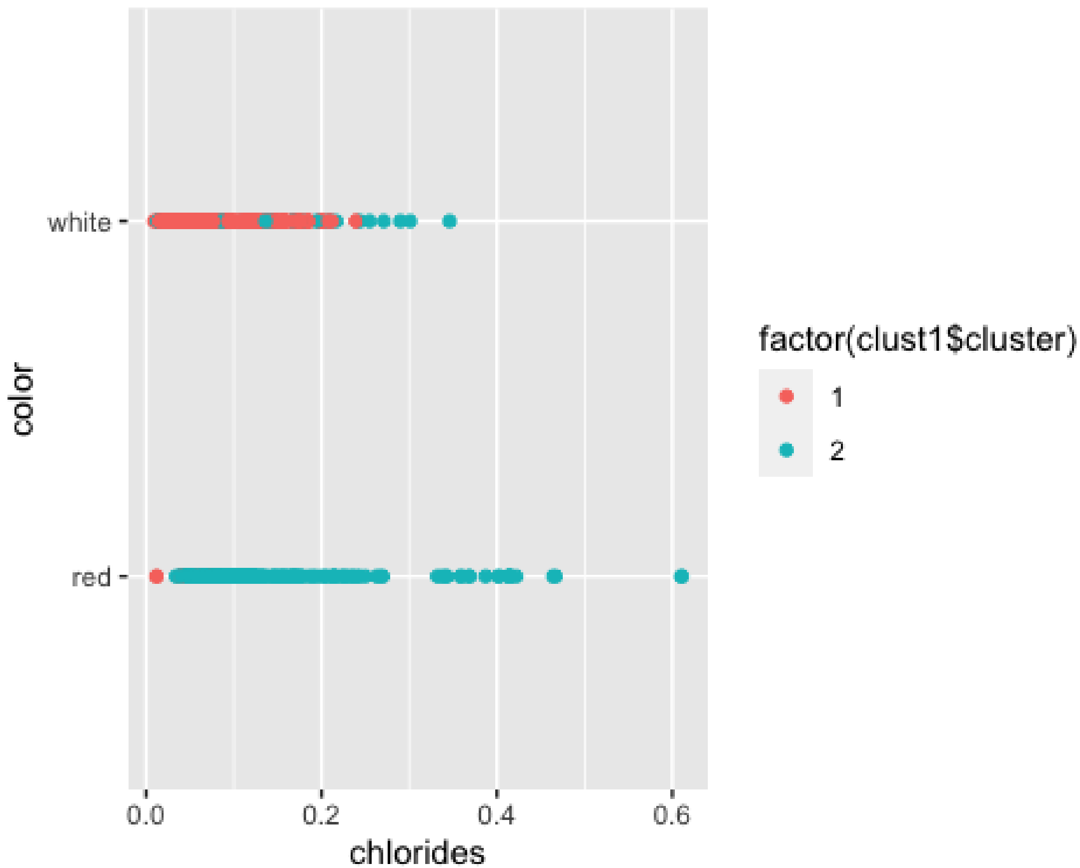

white
red
color
8
10
12
14
alcohol
factor(clust1$cluster)
1
2


white
red
color
0.5
1.0
1.5
2.0
sulphates
factor(clust1$cluster)
1
2


white
red
color
3.0
3.5
4.0
pH
factor(clust1$cluster)
1
2



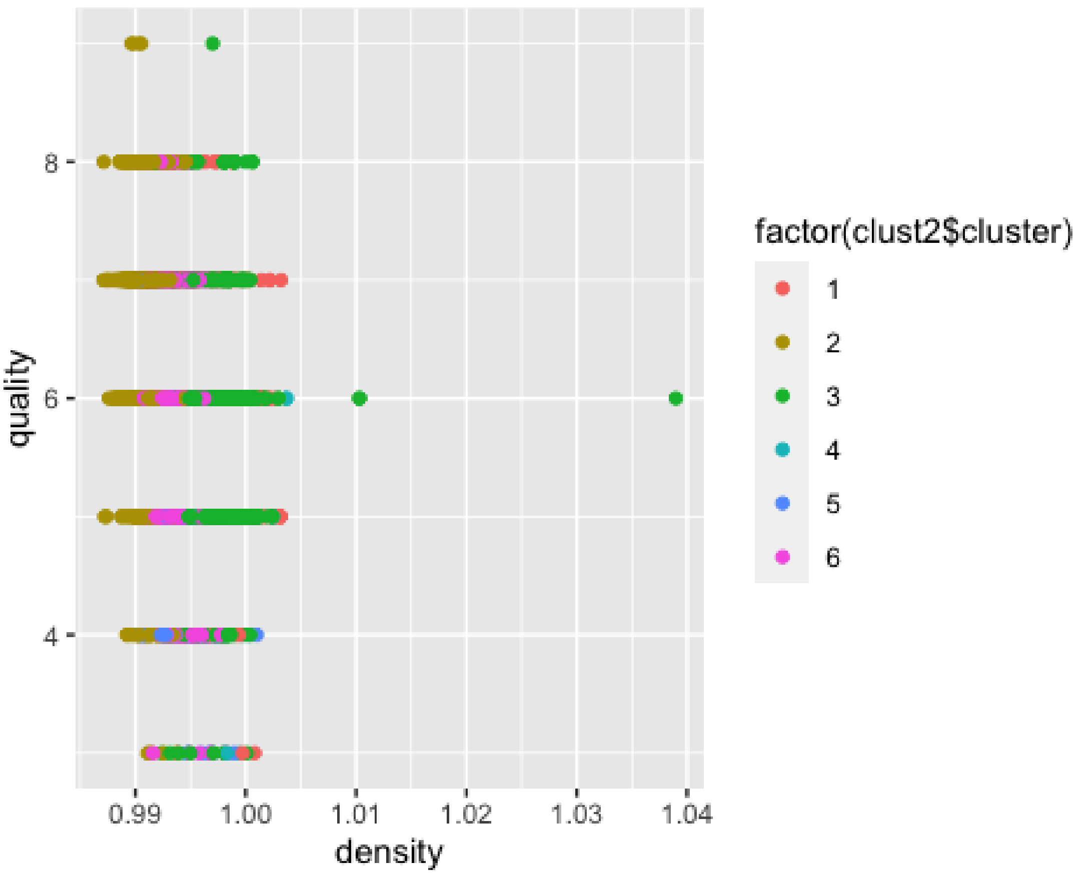
white
red
color
4
8
12
16
fixed.acidity
factor(clust1$cluster)
1
2



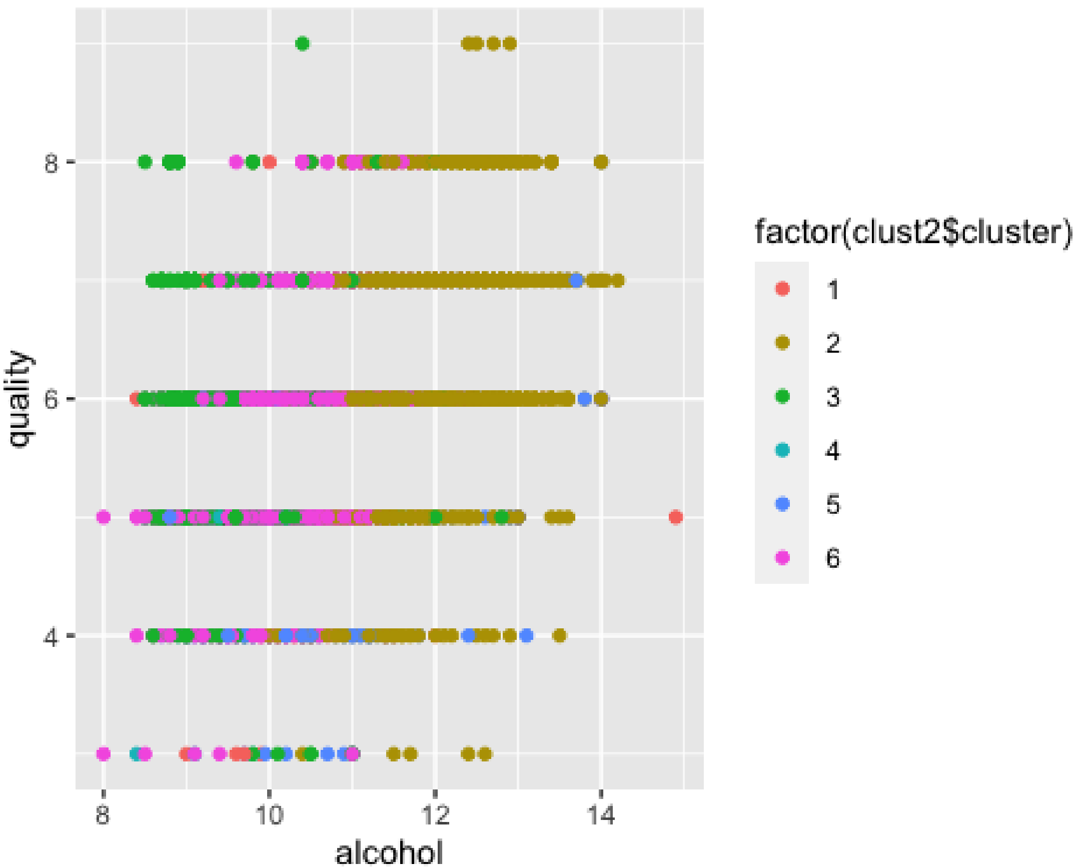
The section above included a table for predicted results and some graphical proves. From the table, we observed that most of the cluster 1 data points are clustered as red wine, and most of the data points in cluster 2 are categorized as white wine. We can observe the same pattern from the graphs. With a total of 11 variables, we selected 6 of them: density, pH, alcohol, chlorides, fixed.acidity, and sulphate to observe whether the clustering models can distinguish between red and white wine. By observing each of these six graphs, we can conclude that most cluster 1 points are categorized as red wine, and most of cluster 2 points are categorized as white wine, which is corresponding to our results from the table.

Next, we ran k-means with 6 clusters and 20 starts to examine the **quality of wine**.

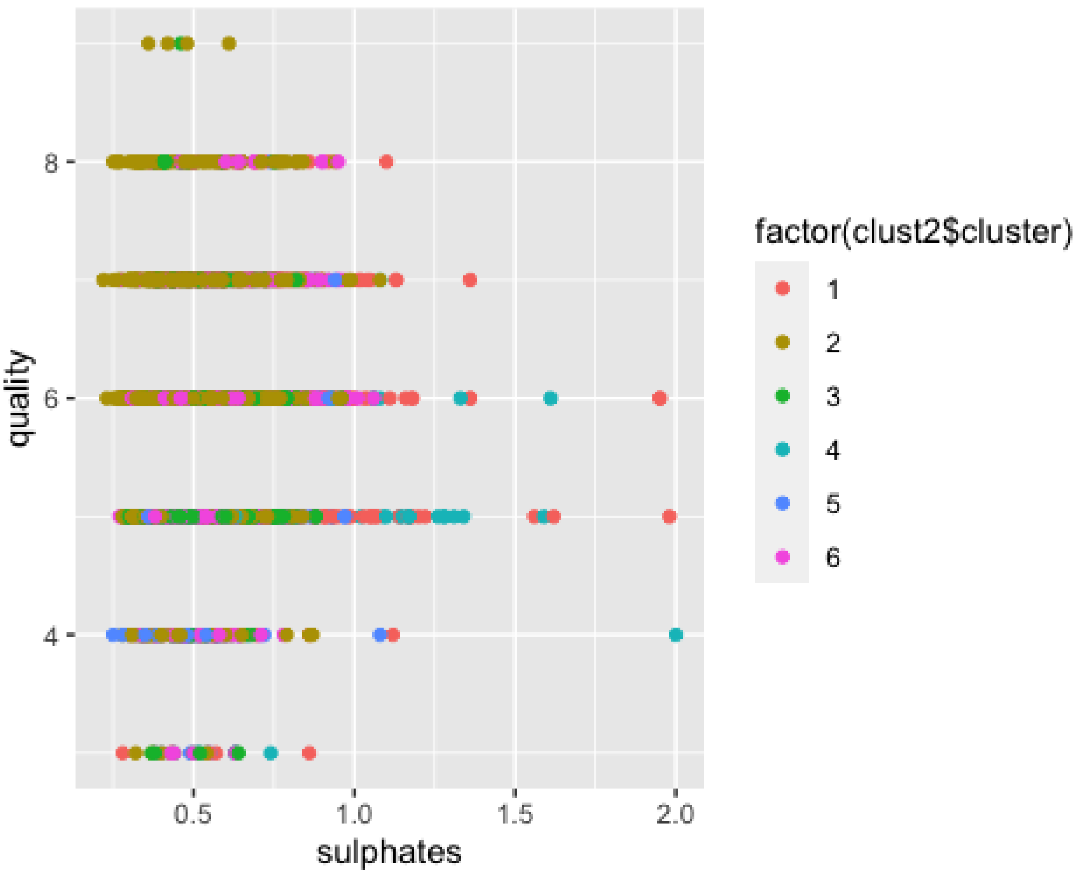
```
##                wine$quality
## clust2$cluster   3   4   5   6   7   8   9
##              1   4  21 203 266 141  14   0
##              2   5  41 198 750 531 112   4
##              3   7  29 673 685 132  27   1
##              4   3   2  54  44   2   0   0
##              5   6  67 466 352  48   3   0
##              6   5  56 544 739 225  37   0
```

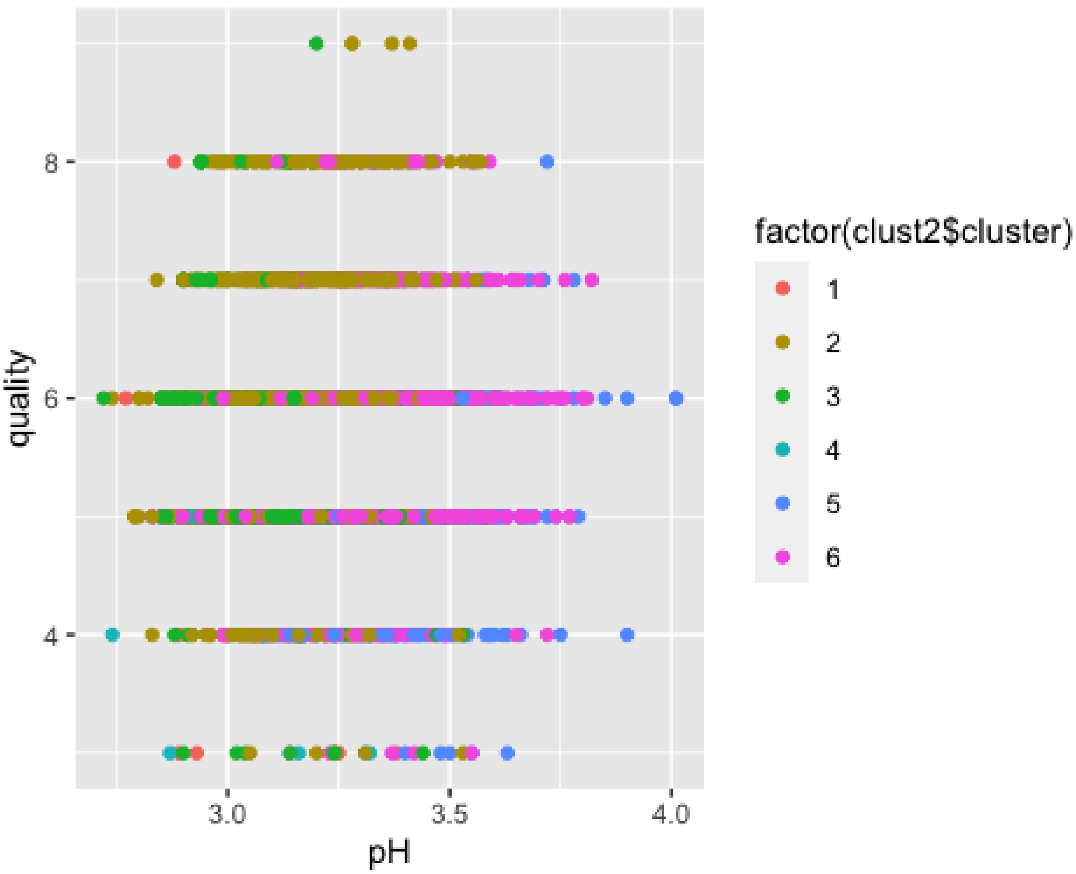


factor(clust2$cluster)
1
2
3
4
5
6
quality
8
6
4
density
0.99
1.00
1.01
1.02
1.03
1.04

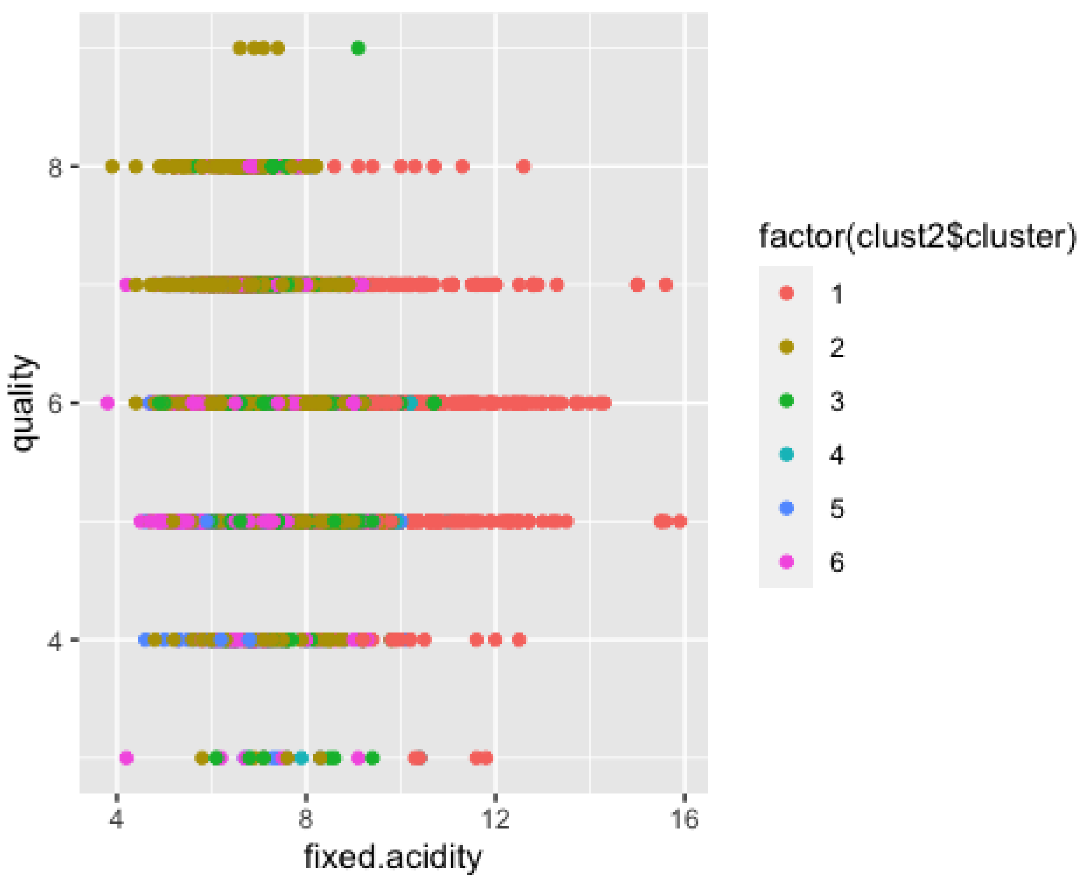


quality
alcohol
8
6
4
8
10
12
14
factor(clust2$cluster)
1
2
3
4
5
6

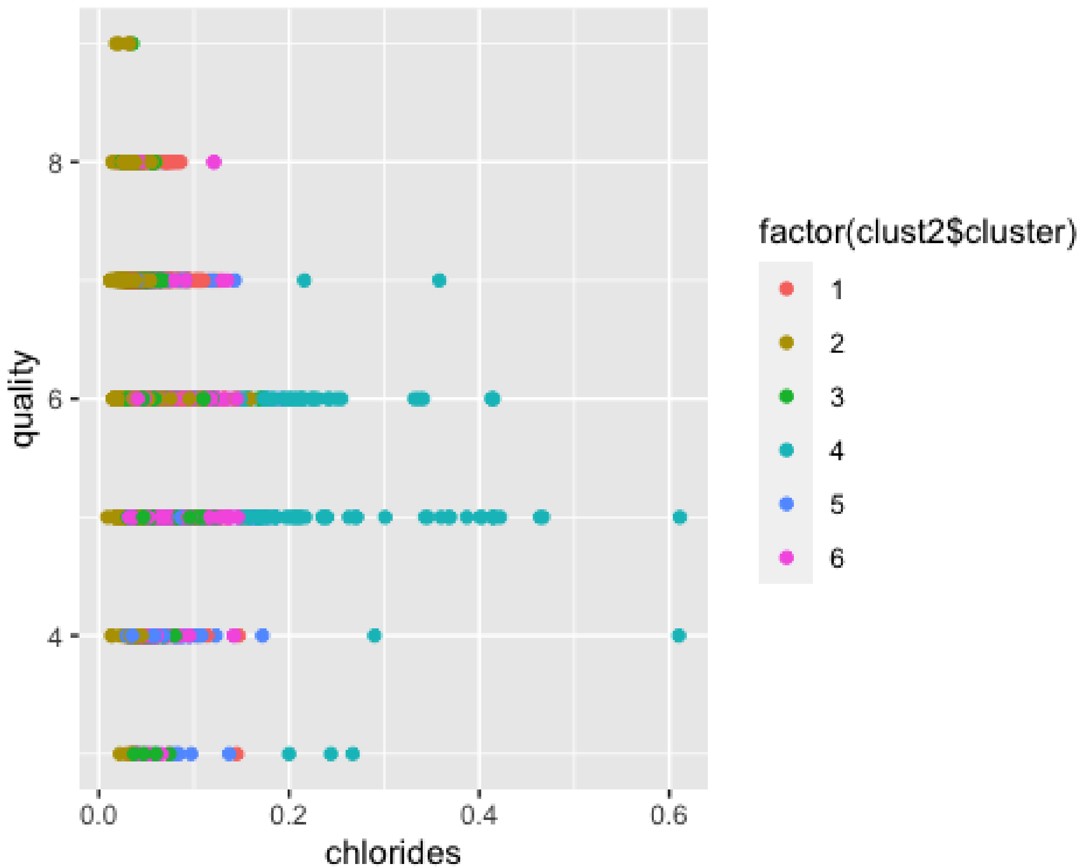

quality
8
6
4
0.5
1.0
1.5
2.0
sulphates
factor(clust2$cluster)
1
2
3
4
5
6



quality
8
6
4
3.0
3.5
4.0
pH
factor(clust2$cluster)
1
2
3
4
5
6



quality
8
6
4
4
8
12
16
fixed.acidity
factor(clust2$cluster)
1
2
3
4
5
6

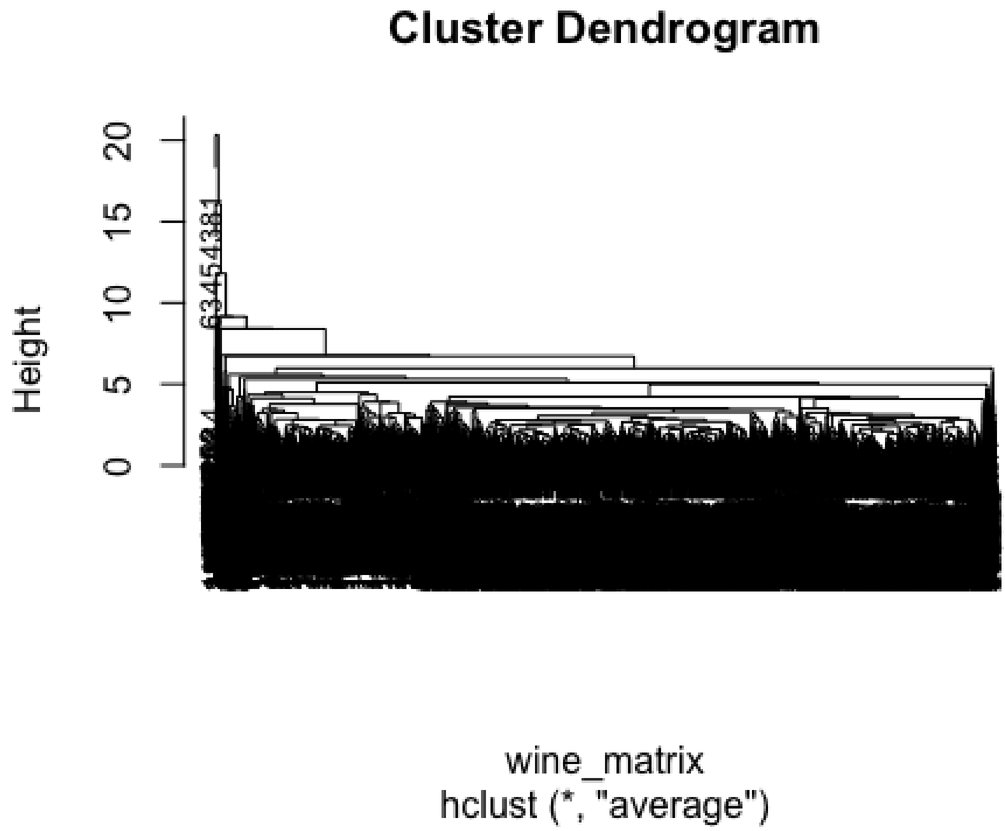


We have also included a table and graphs (with the same variables) for this part as well. However, from both the table and the graphs, we can observe that the data points in each cluster are distributed in different level of qualities. It is very hard for us to observe any patterns of which cluster is majorly categorized as which quality level.

Hierarchical Clustering

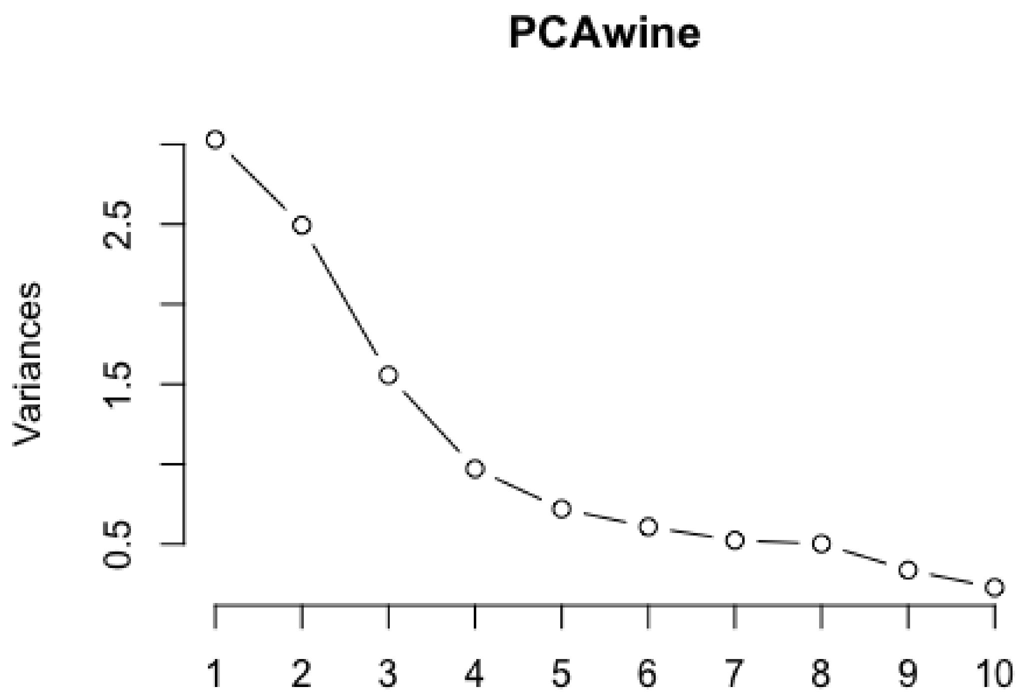
We have also tried hierarchical clustering. However, the resulting clusters are extremely imbalanced, so we would not discuss this model any further.

```
##     1     2     3     4     5     6
##  6462     6    25     2     1     1
```

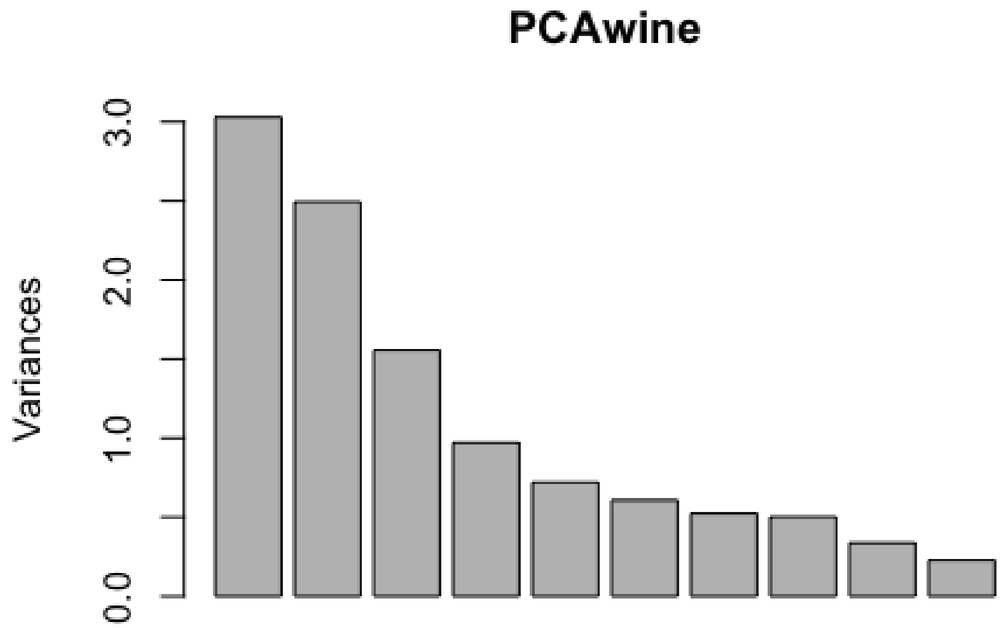


PCA

Next, we used the PCA model to observe the common factors by creating new uncorrelated variables which maximize variance.



PCAwine
Variances
0.5
1.5
2.5
1
2
3
4
5
6
7
8
9
10



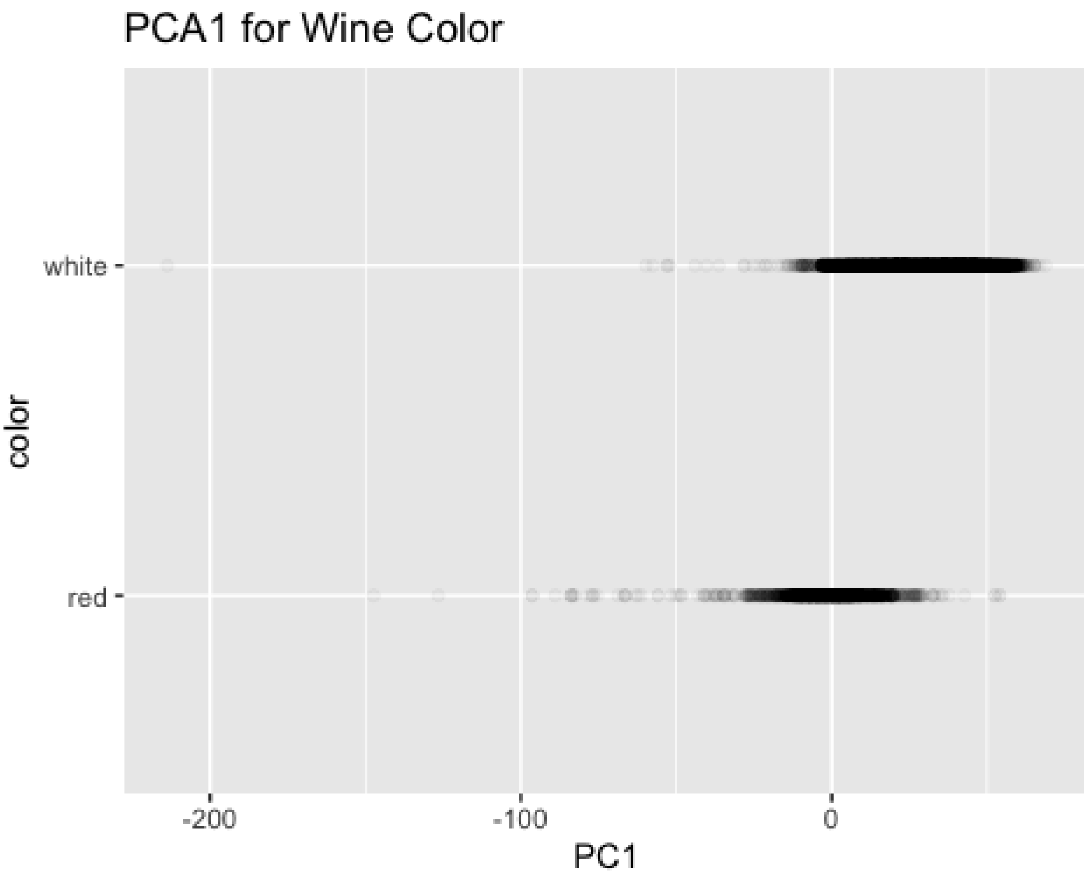

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC
7
## Standard deviation     1.7407 1.5792 1.2475 0.98517 0.84845 0.77930 0.7233
0
## Proportion of Variance 0.2754 0.2267 0.1415 0.08823 0.06544 0.05521 0.0475
6
## Cumulative Proportion  0.2754 0.5021 0.6436 0.73187 0.79732 0.85253 0.9000
9
##                            PC8     PC9   PC10    PC11
## Standard deviation     0.70817 0.58054 0.4772 0.18119
## Proportion of Variance 0.04559 0.03064 0.0207 0.00298
## Cumulative Proportion  0.94568 0.97632 0.9970 1.00000
```

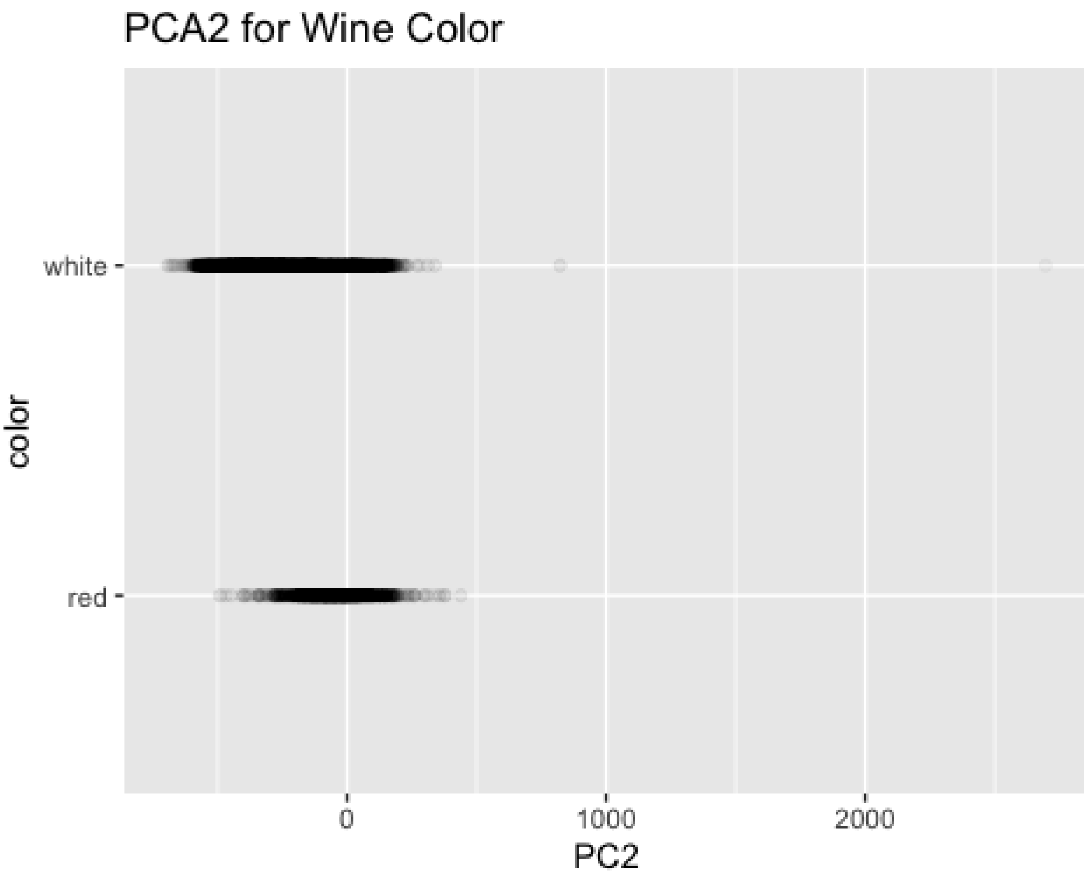
We first ran the pca model and find the summary of the dimensionality reduced summaries. We didn't set a specific number of PCA variables to be analyzed because we wanted to observe the overall variances for each variable. After getting the result, we decided to include the first four PCA variables that explain approximately 75% of the variance. Then, we used these PCA variables to categorize wine color and quality by graphing out the results.

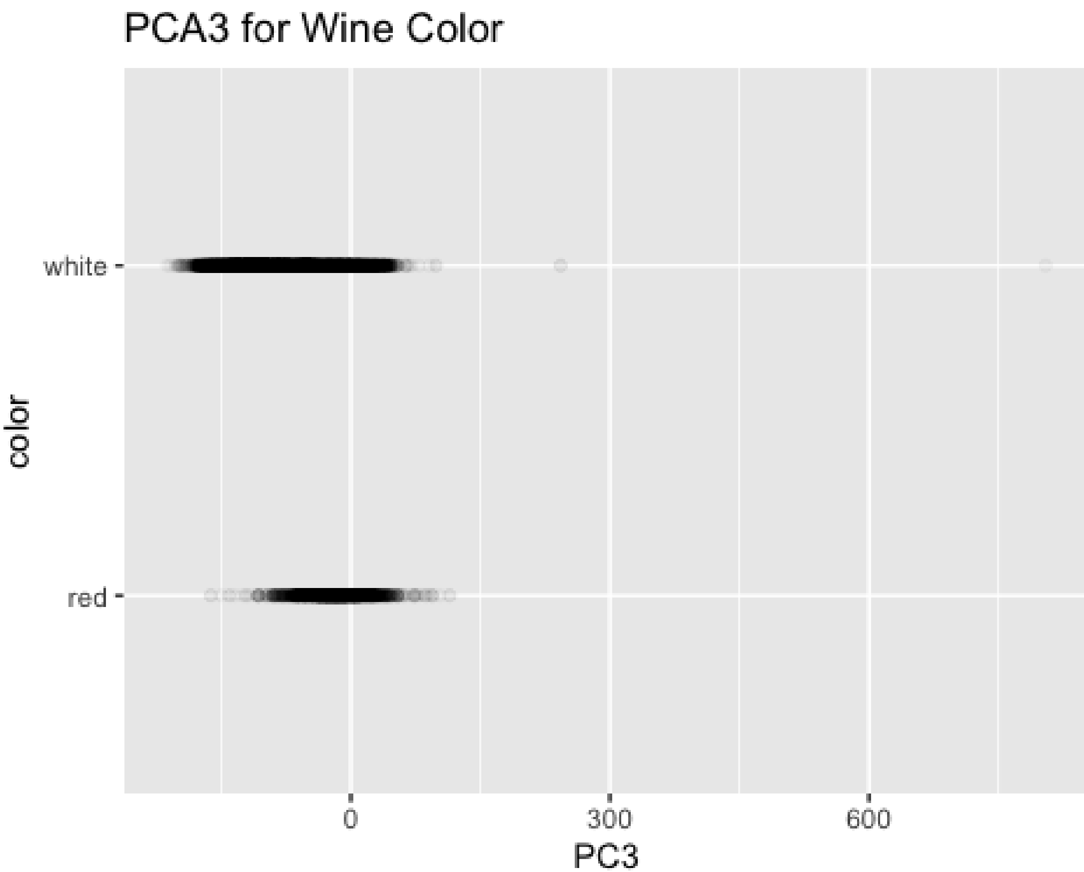
```
##                        PC1   PC2   PC3   PC4
## fixed.acidity        -0.24  0.34 -0.43  0.16
```

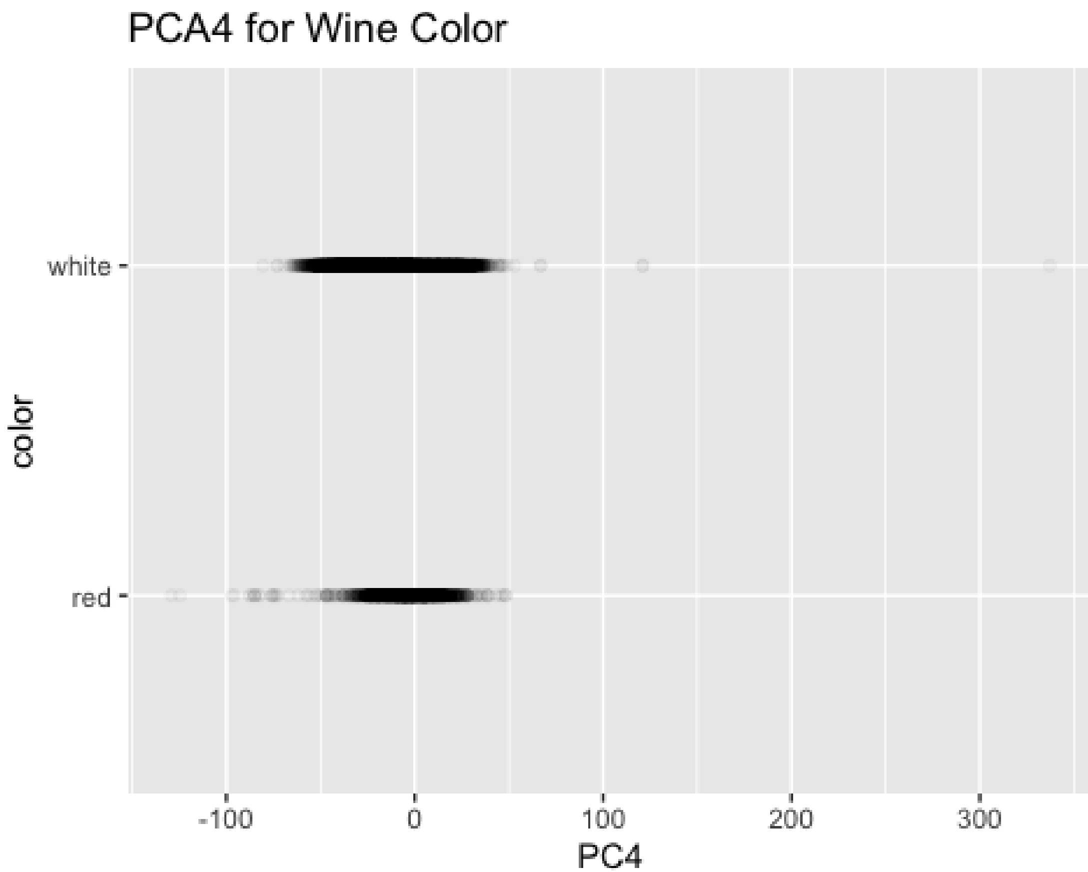
```
## volatile.acidity     -0.38  0.12  0.31  0.21
## citric.acid           0.15  0.18 -0.59 -0.26
## residual.sugar        0.35  0.33  0.16  0.17
## chlorides            -0.29  0.32  0.02 -0.24
## free.sulfur.dioxide   0.43  0.07  0.13 -0.36
## total.sulfur.dioxide  0.49  0.09  0.11 -0.21
## density              -0.04  0.58  0.18  0.07
## pH                   -0.22 -0.16  0.46 -0.41
## sulphates            -0.29  0.19 -0.07 -0.64
## alcohol              -0.11 -0.47 -0.26 -0.11
```

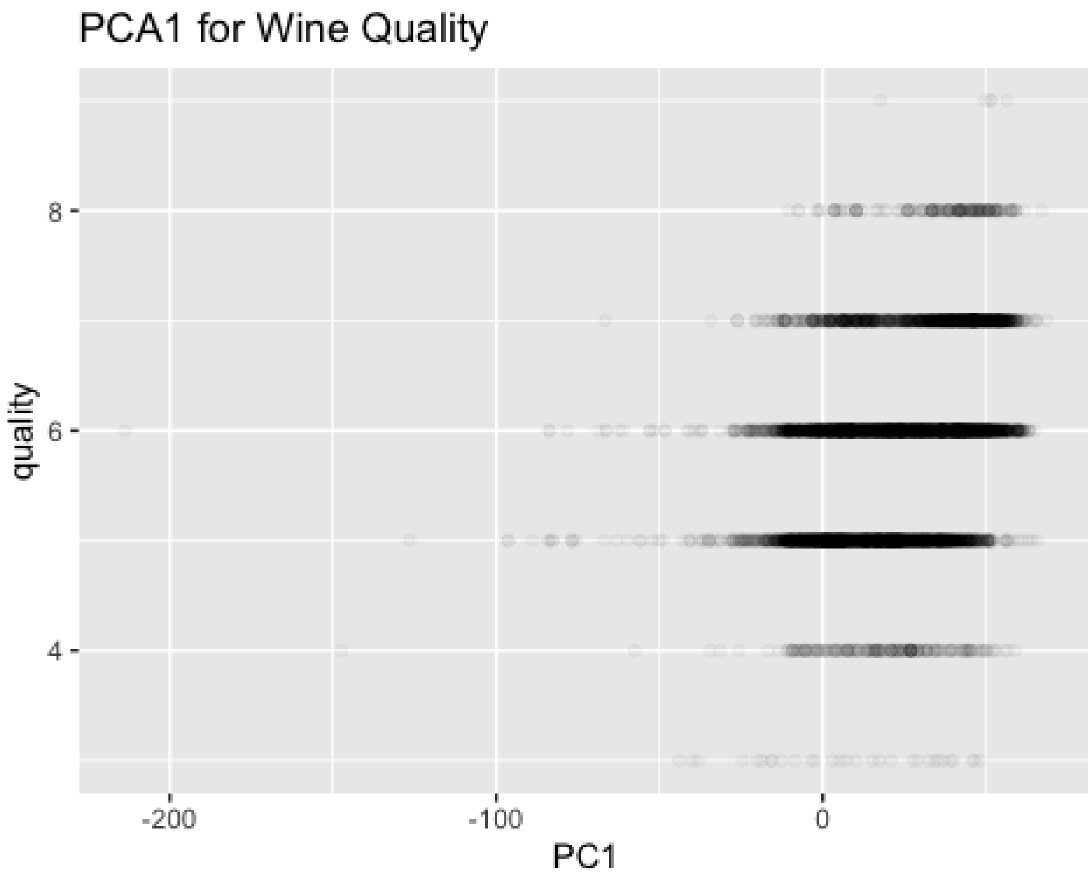
Looking at the summary table from PC1 to PC4, we can observe some patterns and similarities between different summary variables. For example, the PCA variable values are similar for sulphates and chlorides except for PC4. Moreover, the PCA variable values for free.sulfur.dioxide and total.sulfur.dioxide are very similar. One possible explanation is that these two chemical properties might be very similar to each other.


PCA1 for Wine Color
white
red
color
-200
-100
0
PC1

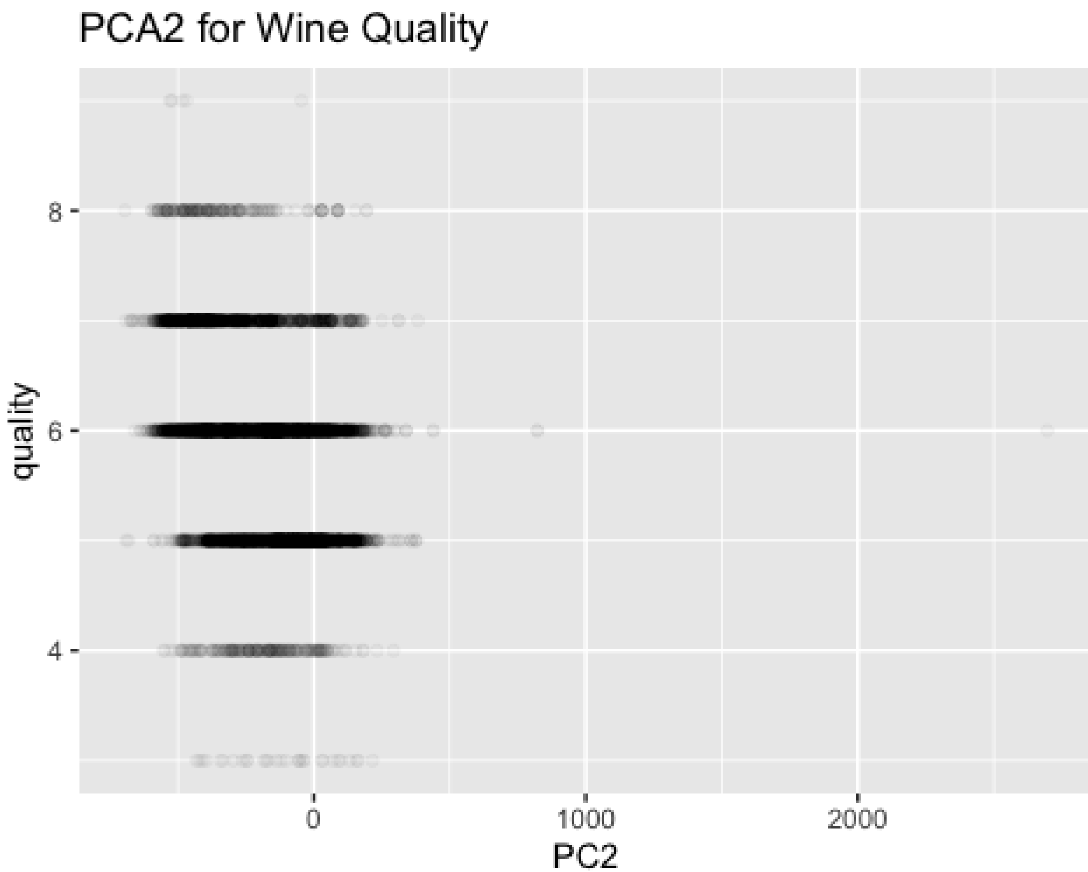

PCA2 for Wine Color
white
red
color
0
1000
2000
PC2

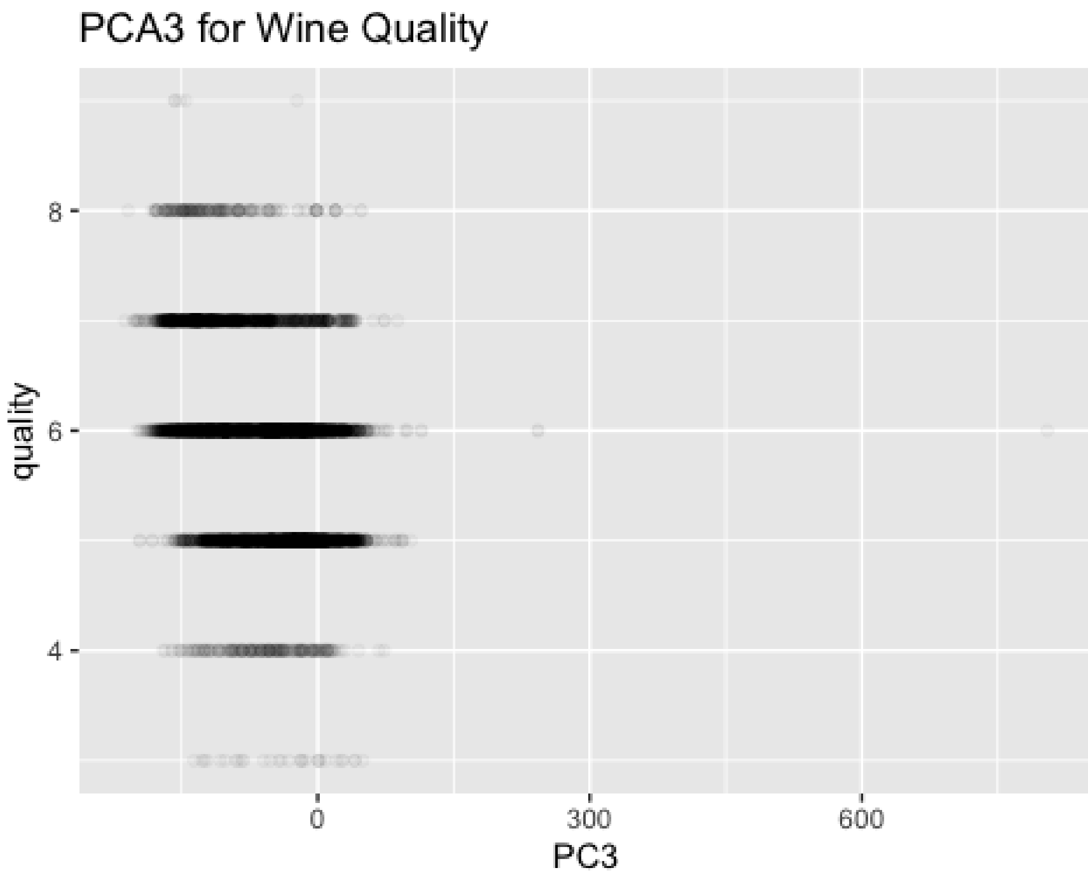

PCA3 for Wine Color
white
red
color
0
300
600
PC3

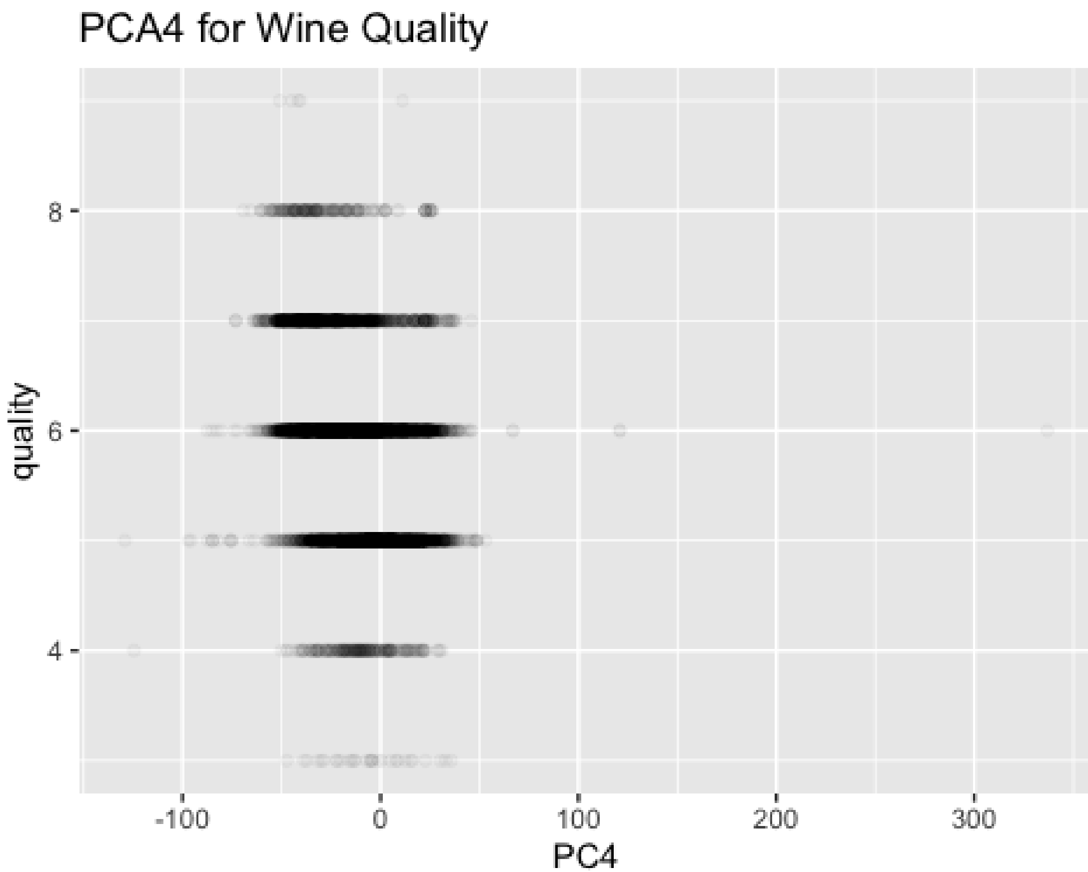

PCA4 for Wine Color
white
red
color
-100
0
100
200
300
PC4



PCA1 for Wine Quality
quality
8
6
4
-200
-100
0
PC1


PCA2 for Wine Quality
quality
8
6
4
0
1000
2000
PC2


PCA3 for Wine Quality
quality
8
6
4
0
300
600
PC3



Looking at the plots for each PCA variable and wine color, the range for red and white wine in PC4 overlaps with each other, making us harder to determine the correlation between PCA variables and distinguishing wine color. For PC1, the range for white wine is higher and wider than the range for red wine, but a significant portion still overlaps. For PC2 and PC3, the distribution for white wine is generally lower than the distribution of red wine. Additionally, the ranges for white wine from PC1 to PC3 are generally larger than the ranges for red wine. For the plots for each PCA variable and wine quality, the ranges for different wine quality level in each PCA variable greatly overlaps with each other, so we cannot distinguish the patterns for any quality level in terms of any PCA variable.

Conclusion

By running clustering and PCA models, we have tried to use these two different models to find relationships between the 11 chemical properties and try to categorize wine color and quality. According to the summary tables and graphs, we think that clustering model makes more sense for this data. From both the result table and the graphs, we can observe that clustering model did a good job of distinguishing red and white wines. However, although it was pretty accurate on distinguish the wine color, it doesn't seem capable of distinguishing between wines from different quality level. Moreover, the distinguishing power of PCA model on wine quality doesn't seem accurate as well.

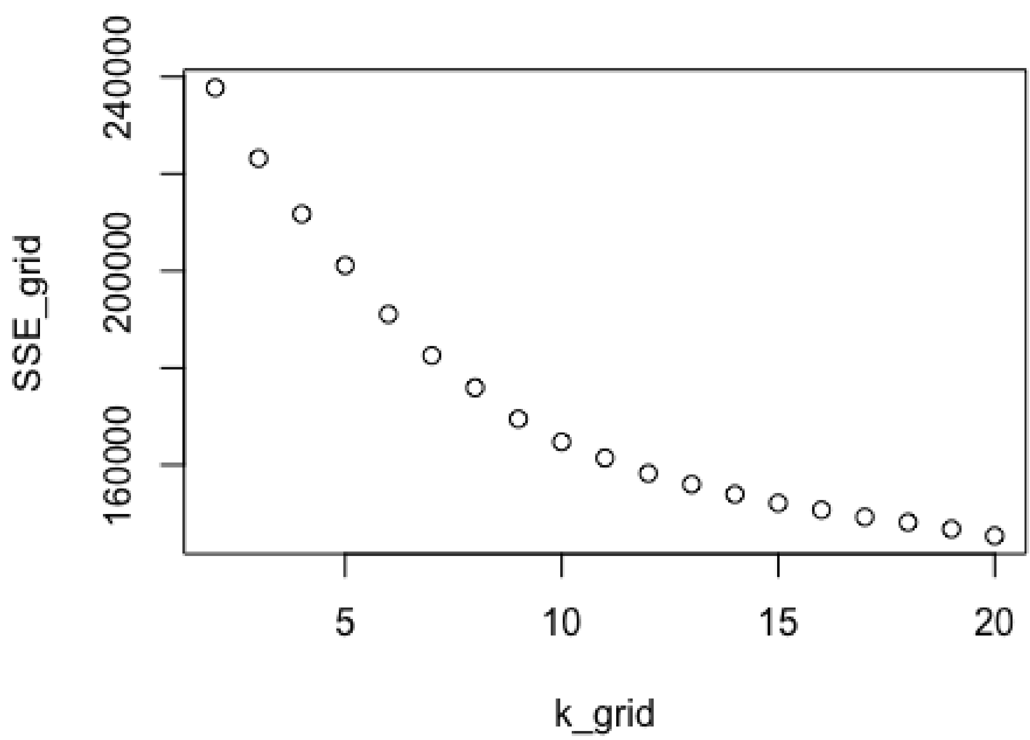
Problem 6: Market Segmentation

Read the social marketing file

For this exercise, we are trying to segment the market using two different clustering method in order to split the users in the market and assess the preference of users in each cluster. With this information, we will prepare a report based on analyzing the segment and understanding our audience better. #### We start with removing adult, spam, uncategorized columns, then we scale & center the data

```
## [1] 7882   34
```

Now we need to decide the optimal number of K. Here we will use elbow plot and CH index. If the results from these two techniques don't match, we will hand pick the optimal K value.




```
## Warning: did not converge in 10 iterations

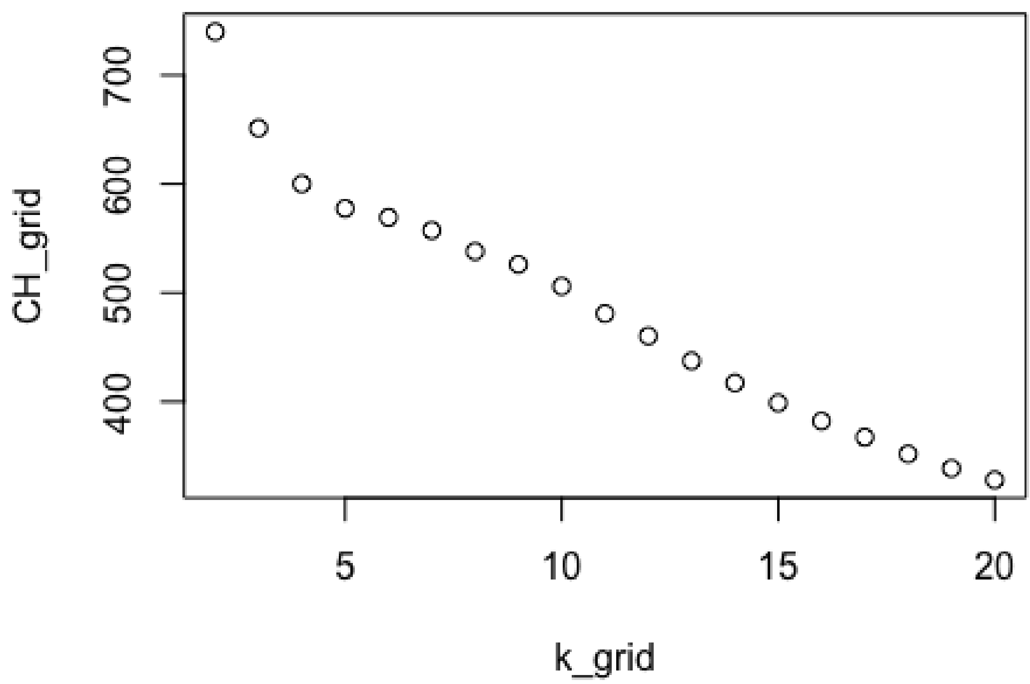
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

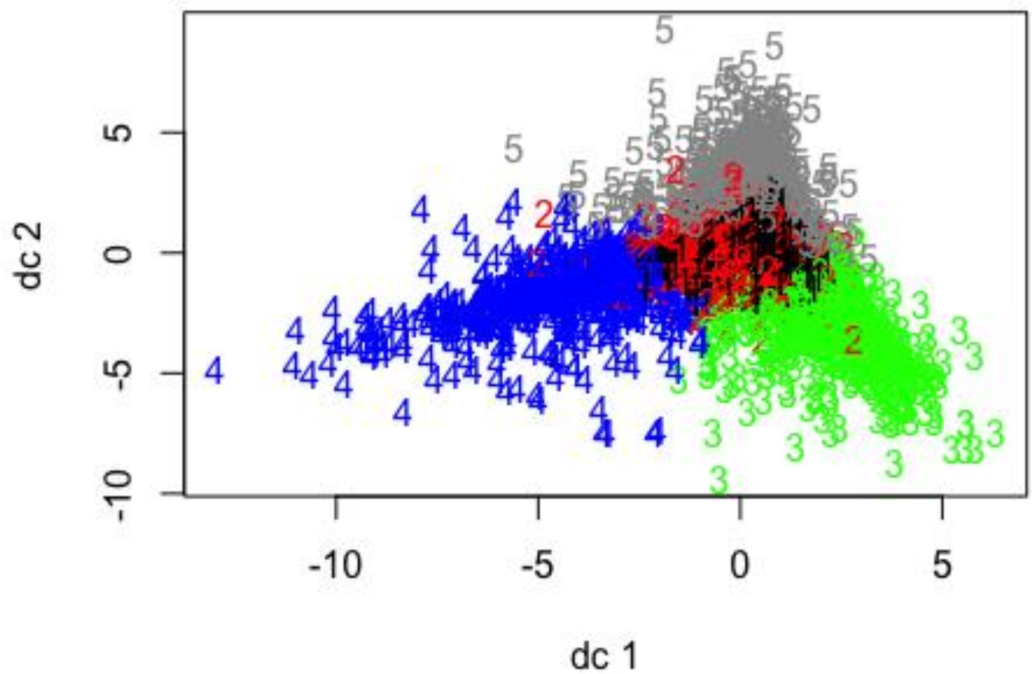
## Warning: did not converge in 10 iterations
```

```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```



By observing the results from the two plots, we decide to use the number of clusters as 5.



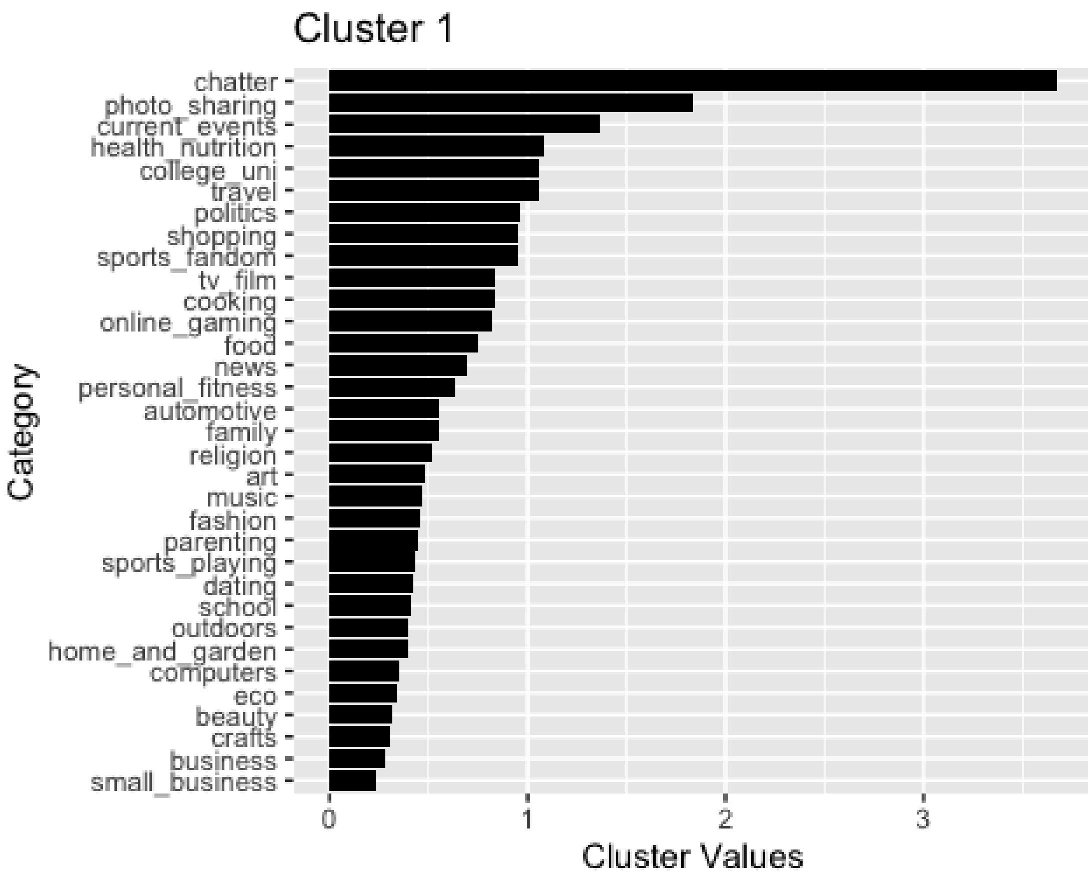

```
## [1] 201103.6

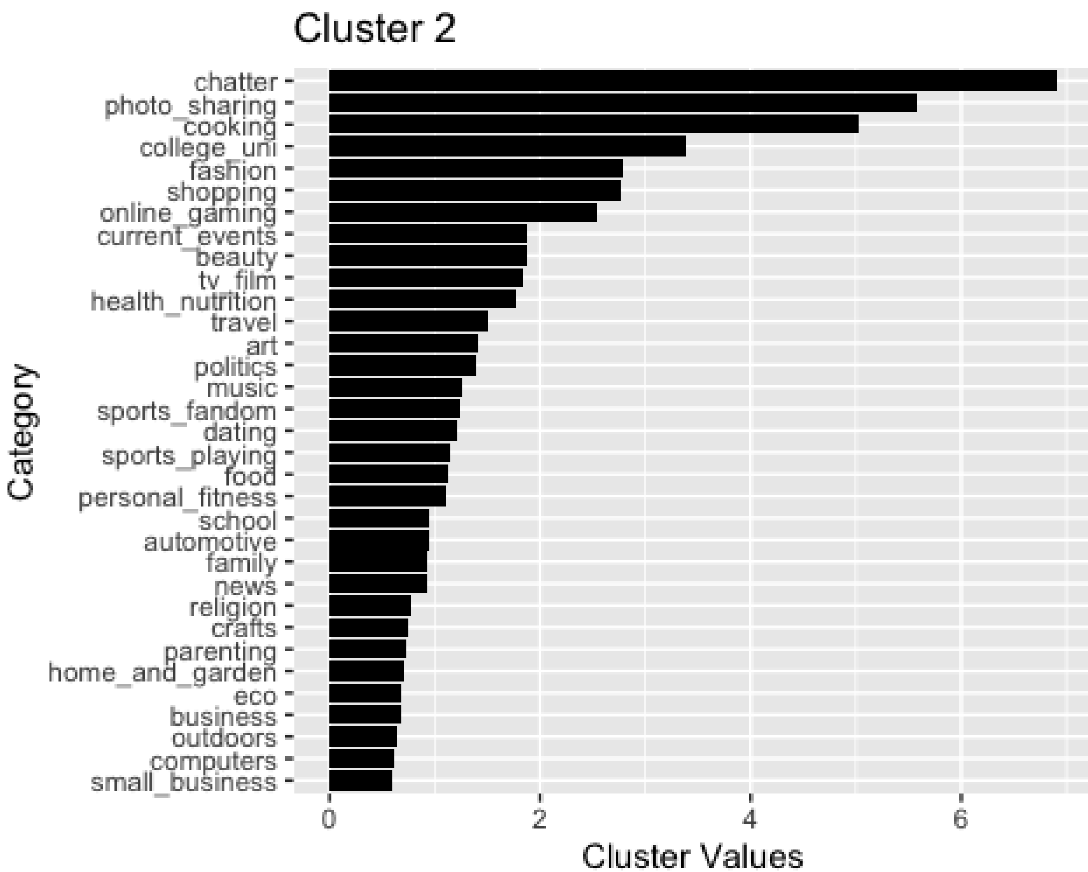
## [1] 58969.44

##  [1]  2  4  5  6 11 14 15 16 19 20 22 23 24 25 26 27 29 31 35 36
```

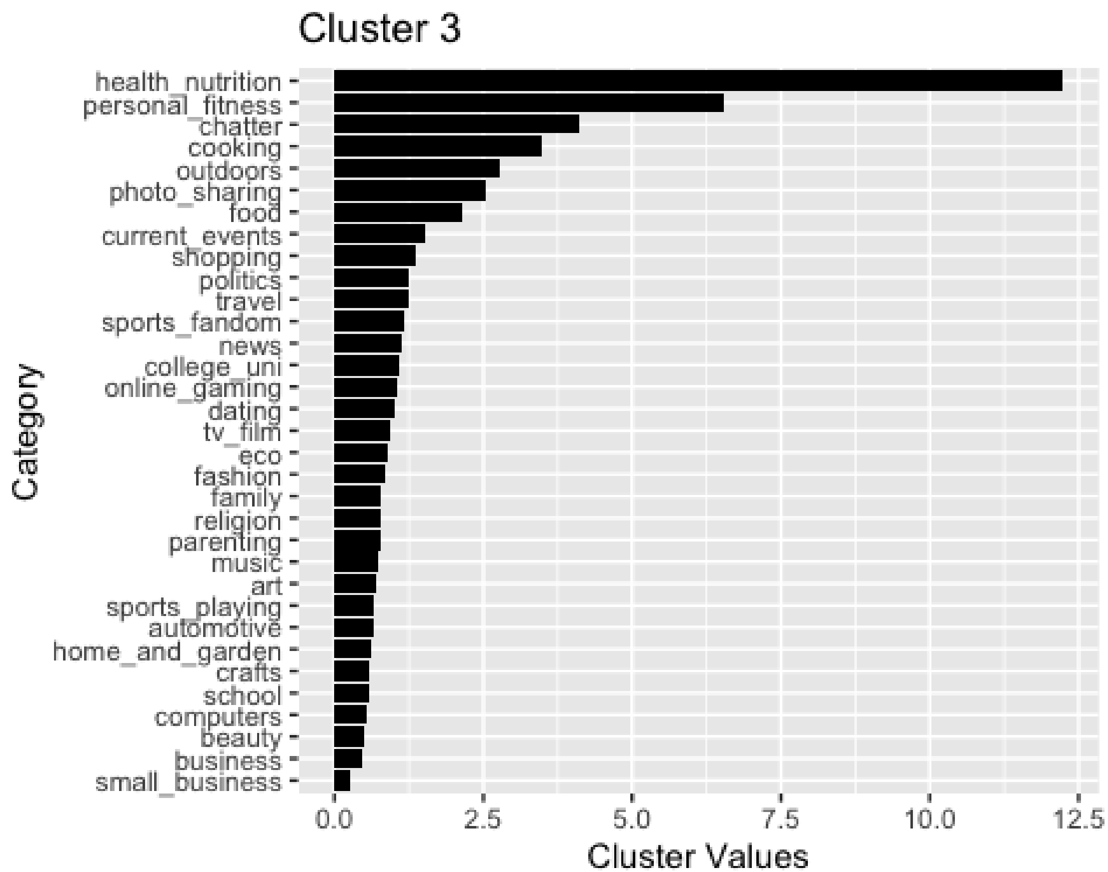
We plot the data points in different colors and obtained a total withiness of 201103.6 and betweeness of 58969.44.

We have successfully categorized each data point into a cluster. Now, in order to better understand our audience, we begin analyzing the key features representing each cluster.

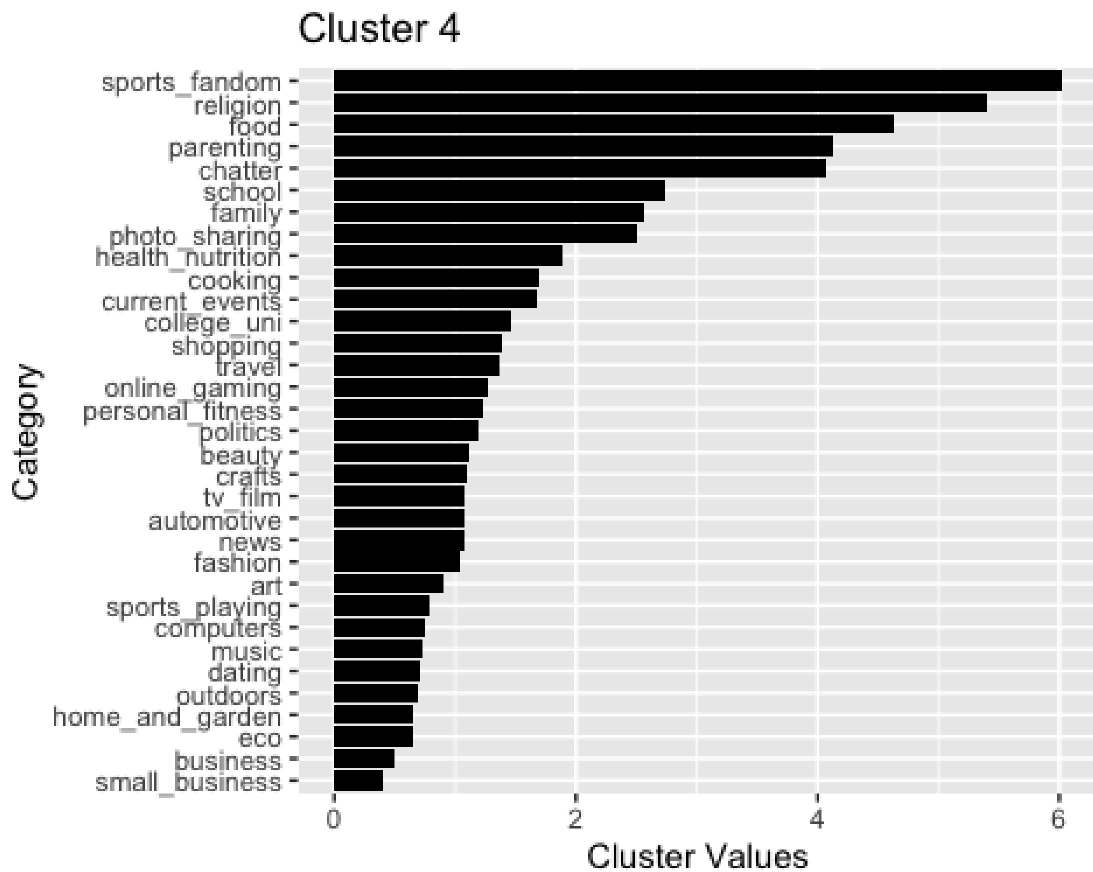

Cluster 1
Category
chatter
photo_sharing
current_events
health_nutrition
college_uni
travel
politics
shopping
sports_fandom
tv_film
cooking
online_gaming
food
news
personal_fitness
automotive
family
religion
art
music
fashion
parenting
sports_playing
dating
school
outdoors
home_and_garden
computers
eco
beauty
crafts
business
small_business
0
1
2
3
Cluster Values



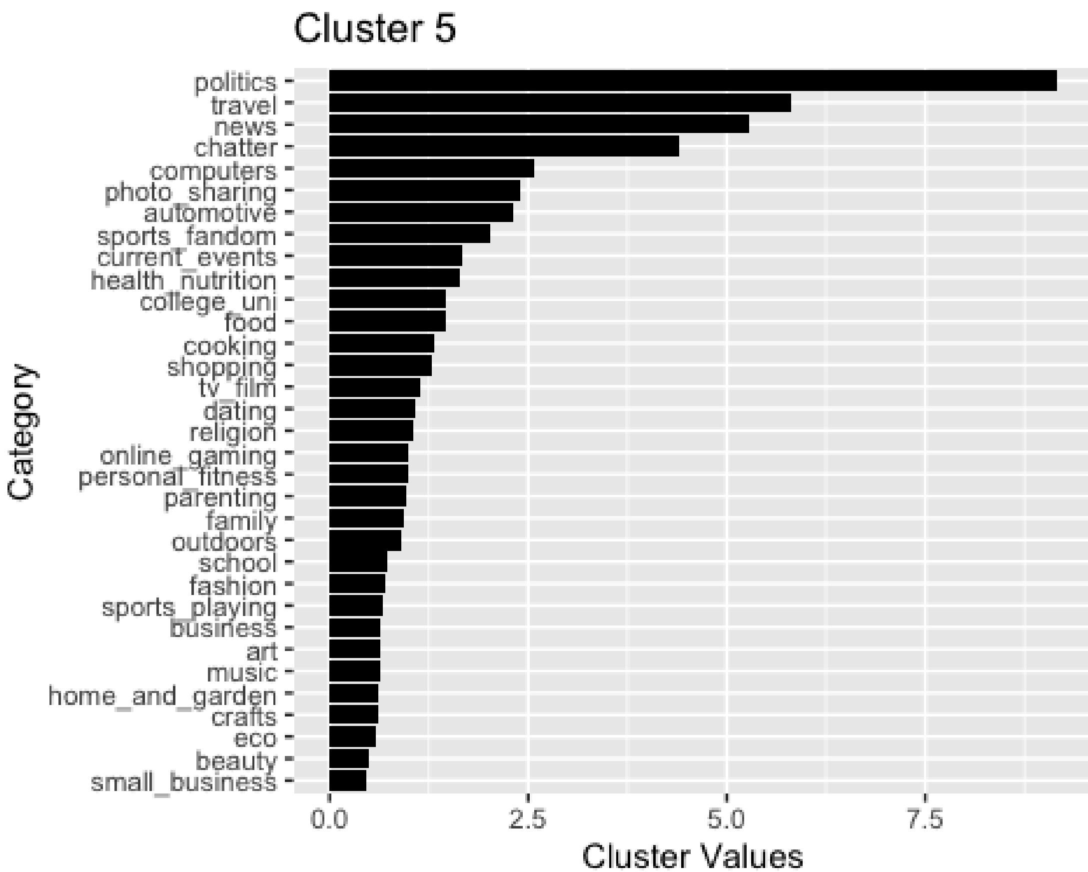
Cluster 2
chatter
photo_sharing
cooking
college_uni
fashion
shopping
online_gaming
current_events
beauty
tv_film
health_nutrition
travel
art
politics
music
sports_fandom
dating
sports_playing
food
personal_fitness
school
automotive
family
news
religion
crafts
parenting
home_and_garden
eco
business
outdoors
computers
small_business
Category
0
2
4
6
Cluster Values



Cluster 3
health_nutrition
personal_fitness
chatter
cooking
outdoors
photo_sharing
food
current_events
shopping
politics
travel
sports_fandom
news
college_uni
online_gaming
dating
tv_film
eco
fashion
family
religion
parenting
music
art
sports_playing
automotive
home_and_garden
crafts
school
computers
beauty
business
small_business
Category
0.0
2.5
5.0
7.5
10.0
12.5
Cluster Values



Cluster 4
sports_fandom
religion
food
parenting
chatter
school
family
photo_sharing
health_nutrition
cooking
current_events
college_uni
shopping
travel
online_gaming
personal_fitness
politics
beauty
crafts
tv_film
automotive
news
fashion
art
sports_playing
computers
music
dating
outdoors
home_and_garden
eco
business
small_business
Category
0
2
4
6
Cluster Values



The five market segments we found are characterized by the following features: * Cluster 1: chatter, photo sharing, cooking * Cluster 2: sports fandom, religion, food * Cluster 3: health nutrition, personal fitness, chatter * Cluster 4: politics, travel, news * Cluster 5: chatter, photo sharing, current events

- Cluster 1 is mostly represented by users who care more about posting photos and cooking contents. So it's reasonable for us to advertise about recipes or shows about food to them. Since they have the characteristics of both photo-sharing and cooking at the same time, we can also send them advertisement about pretty food filter, food pictures, and photo-taking tips.

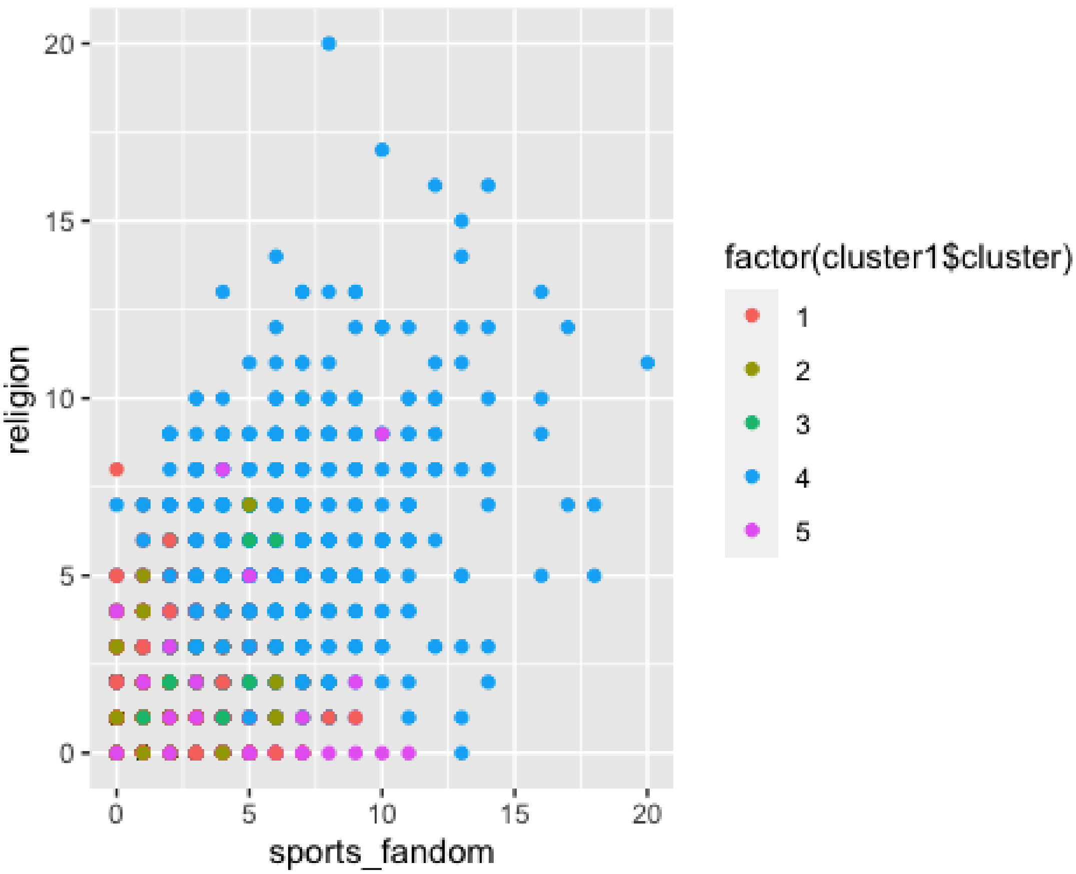
- Cluster 2 is represented by sports fans who are also religious and love food content. We could recommend religious related cuisines/food recipes and sports. Additionally, we can consider about advertising on famous religious restaurants and or sports events around them.

- Cluster 3 is made up of active users who enjoy personal fitness and healthy lifestyle. For products, we can consider advertising fitness-related products like protein bars, organic/fresh food to them. We can also send them some posts written by fitness-related bloggers, healthy food recipes; or we can link these users together, since most of them are also chatters. By linking these users together, we are simulating the network effects within the company to create extra values.

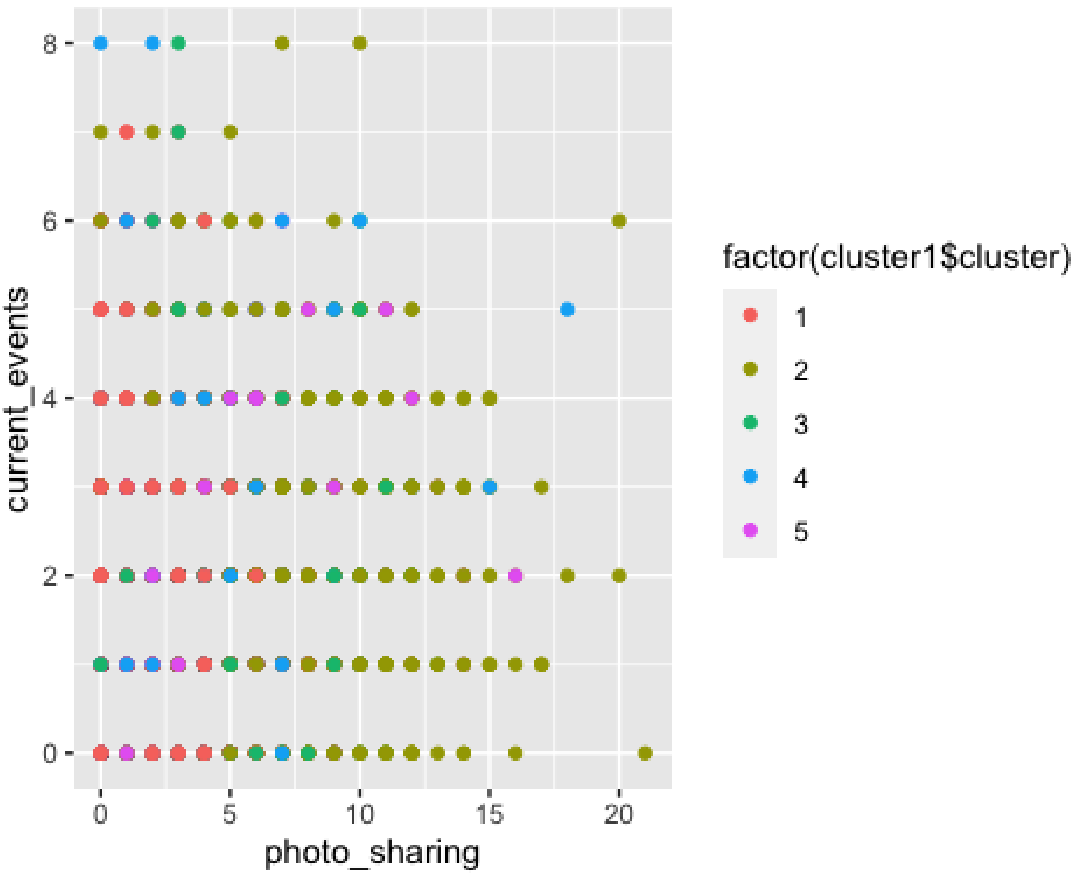
- Cluster 4 is represented by users who love politics, watching news, and travelling. The content they like might be related to political shows and news report about foreign countries, so we can increase the political-related posts they are exposed to. Since they also like traveling, we can also advertise them with traveling bloggers and videos or increase the advertisement related to hotels, flight tickets, or theme park admission tickets.

- Cluster 5 is characterized by users who share photos very often and care a lot about current events. They could be interested in news so we can utilize social media to reach out to them by increasing the amount of current events locally or nationally they receive.

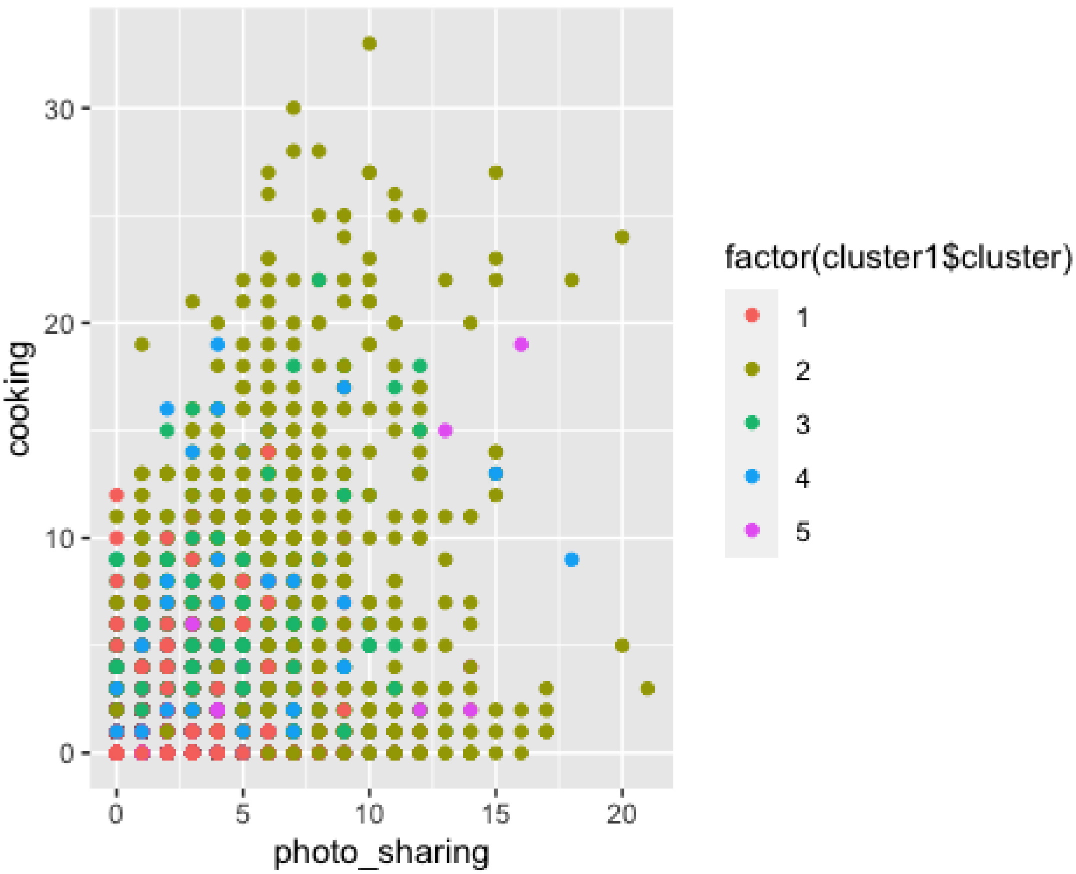
To better solidify our insights above, we also used some of their top characteristics to observe the distribution of clustering in these feature for each cluster.



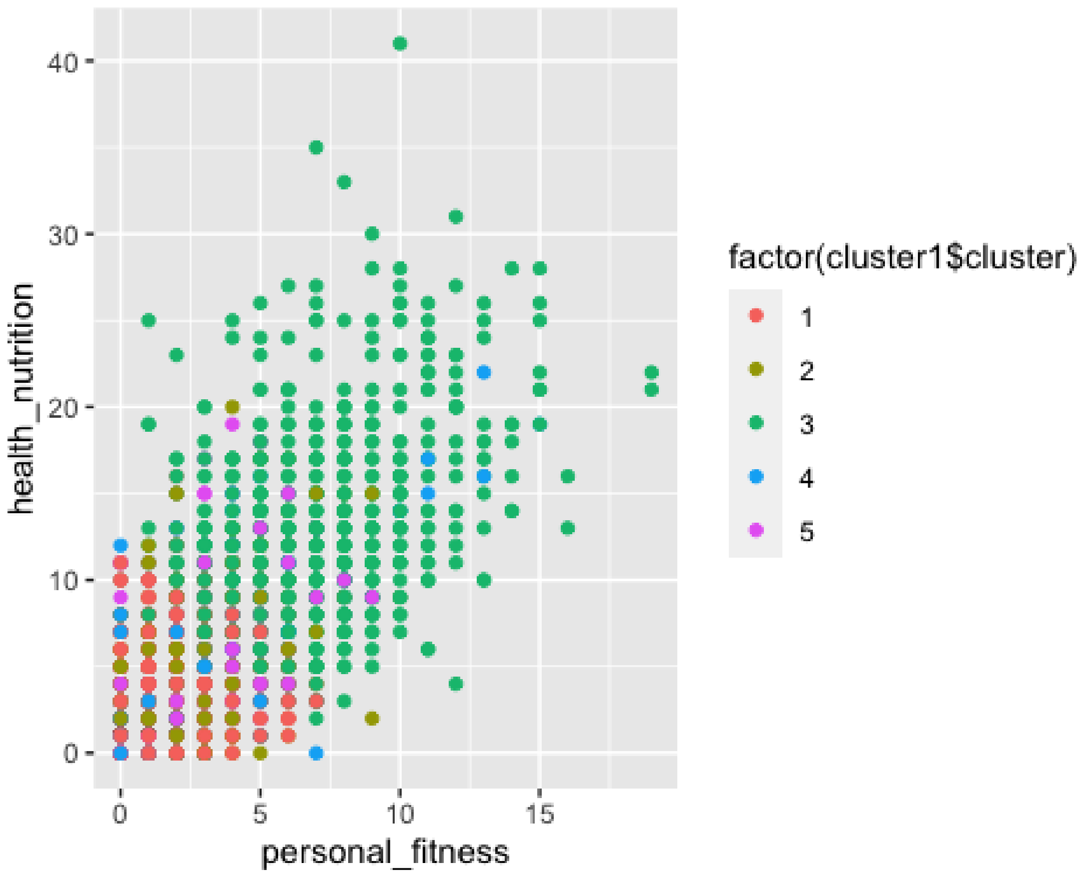
religion
sports_fandom
0
5
10
15
20
factor(cluster1$cluster)
1
2
3
4
5



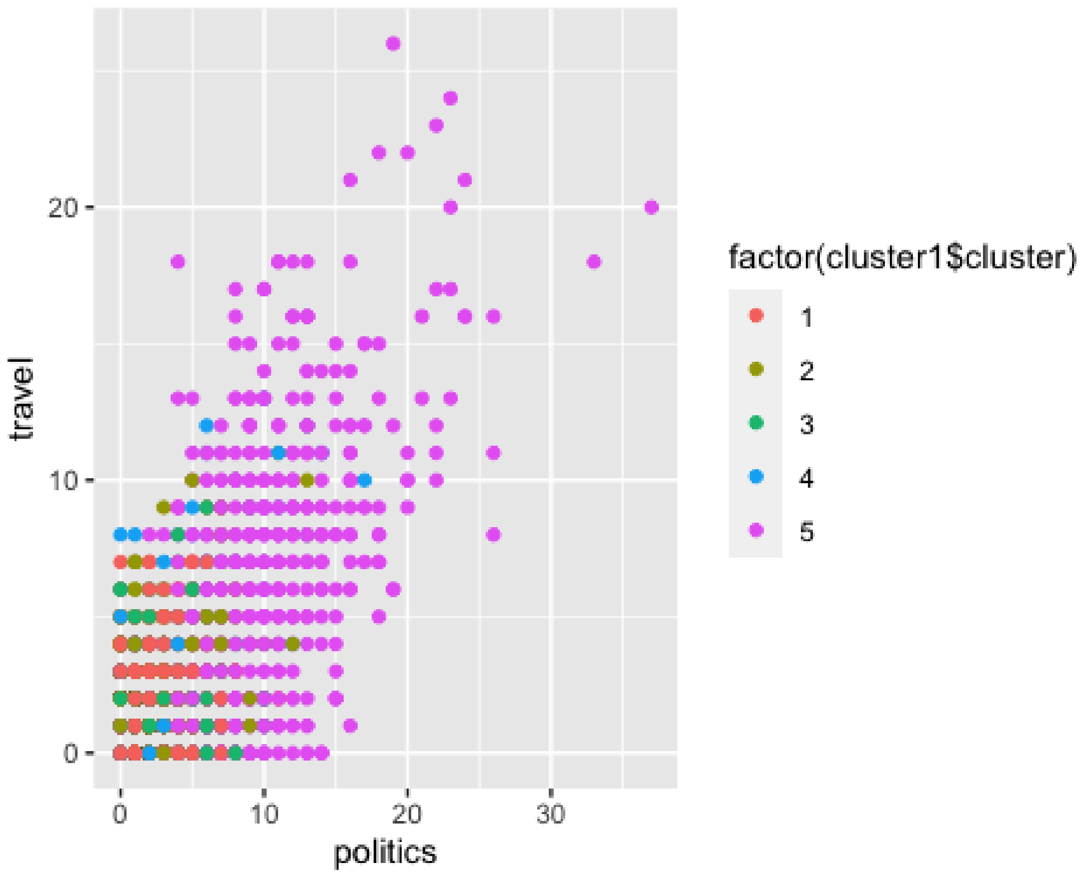
current_events
photo_sharing
factor(cluster1$cluster)
1
2
3
4
5
0
2
4
6
8
0
5
10
15
20



cooking
30
20
10
0
0
5
10
15
20
photo_sharing
factor(cluster1$cluster)
1
2
3
4
5



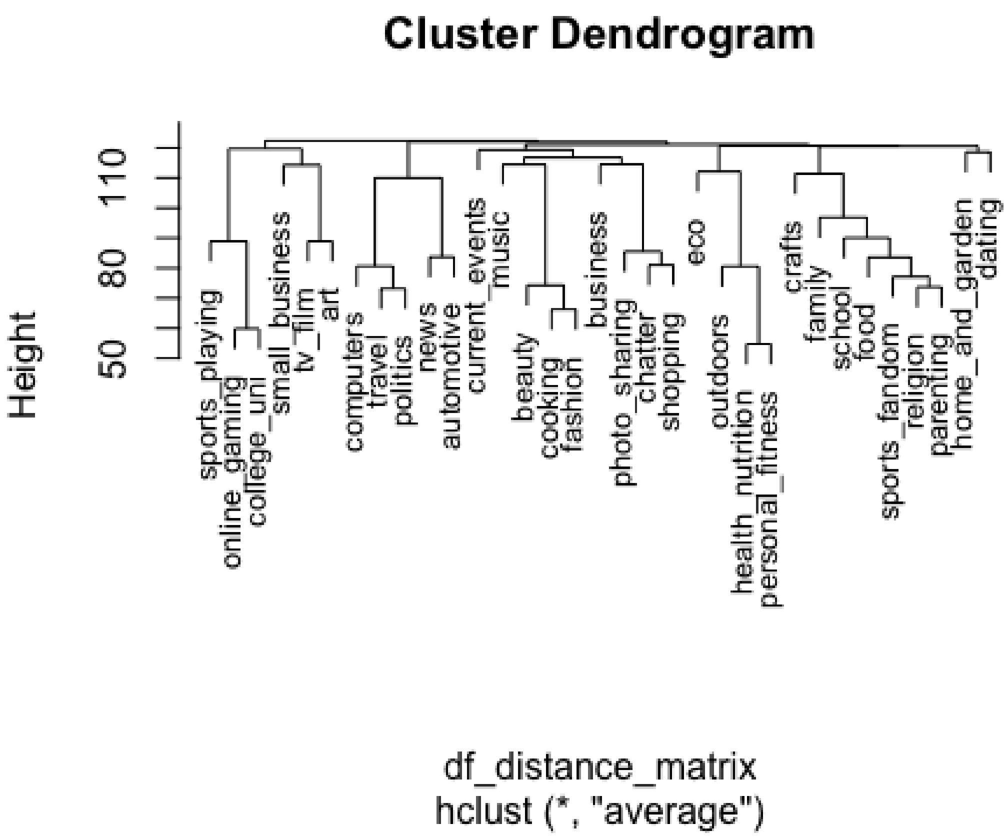
health_nutrition
personal_fitness
0
10
20
30
40
0
5
10
15
factor(cluster1$cluster)
1
2
3
4
5



By observing the scatter plots, we can observe that: * The yellow dots, representing cluster 2, are users who interested in sports fandom and religion. * The orange dots, representing cluster 1, are users who interested in photo sharing and cooking. * The green dots, representing cluster 3, are users who interested in personal fitness and health nutrition. * The blue dots, representing cluster 4, are users who interested in politics and travel * However, the clustering in the second graph is ambiguous, we observe that the characteristics related to this graph are current events and photo sharing; however, there are a lot of orange dots as well as purple dots. One possible explanation for this is that users represented by orange dots also share the characteristic of photo sharing. From the analysis above, we can observe that the relationships we find out in these graphs mostly match with our findings with clusters we mentioned earlier.

Hierarchical Clustering

Additionally, we ran a hierarchical clustering model to compare the findings from k-means clustering.




```
## 1 2 3 4 5
## 9 5 6 9 4
```

By examing the tree diagram above, we can identify the following market segments: * People who love sports, video gaming, interested in college/universities, small businesses, film and art. This cluster of people are likely to be high school or college students who still have time to enjoy gaming, sports, and other media, but also need to start working. * People care about computers, traveling, politics, news, automotive. This cluster of people are likely to be people who already started working or professionals in the industry. * People who care about current events, music, beauty, cooking, fashion, photo sharing, and shopping. This cluster of people are likely to be young females who have a certain amount of purchasing power to support their interests in fashion, beauty, and cooking. * People who like outdoors, health nutrition, and personal fitness. This cluster is similar to the cluster 2 of k-means clustering. They are likely to be athelets, fitness bloggers, or models who care about living a healthy life. * People who care about family, school, food, sports fandom, relition, parenting, home and garden, and dating. This cluster of people are also likely to be young people but at a lower age than college students, who still live with their parents.

Most of them might be taken care by their parents, and their purchasing power might be very limited.

In conclusion, kmeans and hierachical clustering give us very similar maket segments. Such analysis allows us to drive insights that can help he company to send the right message to correct people because they now have a better understanding of specific groups of customers. But the analysis needs to be continued updating because people do change their preferences over time.

Problem 7: The Reuters corpus

7.1 Problem Statement:

In this exercise, we are predicting the author of an article based on the model trained by the c50train directory in the Reuters C50 Corpus. We are observing and comparing the results we get from different models.

7.2 Approach:

7.2.1 Import necessary packages

7.2.2 Read and clean train and test files

7.2.3 Data preprocsessing: Tokenization + Doc-Term Matrix

7.2.3.1 Tokenization

Steps to take: -Convert all characters to lower cases -Remove extra white space -Remove numbers -Remove punctuation -Remove stopwords

```
## <<DocumentTermMatrix (documents: 2500, terms: 32241)>>
## Non-/sparse entries: 473695/80128805
## Sparsity           : 99%
## Maximal term length: 40
## Weighting          : term frequency (tf)
```

```
## <<DocumentTermMatrix (documents: 2500, terms: 660)>>
## Non-/sparse entries: 224397/1425603
## Sparsity           : 86%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

```
## <<DocumentTermMatrix (documents: 2500, terms: 33048)>>
## Non-/sparse entries: 480577/82139423
## Sparsity           : 99%
## Maximal term length: 45
## Weighting          : term frequency (tf)
```

```
## <<DocumentTermMatrix (documents: 2500, terms: 676)>>
## Non-/sparse entries: 228410/1461590
## Sparsity           : 86%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

7.2.3.2 Ensuring identical test and train datasets

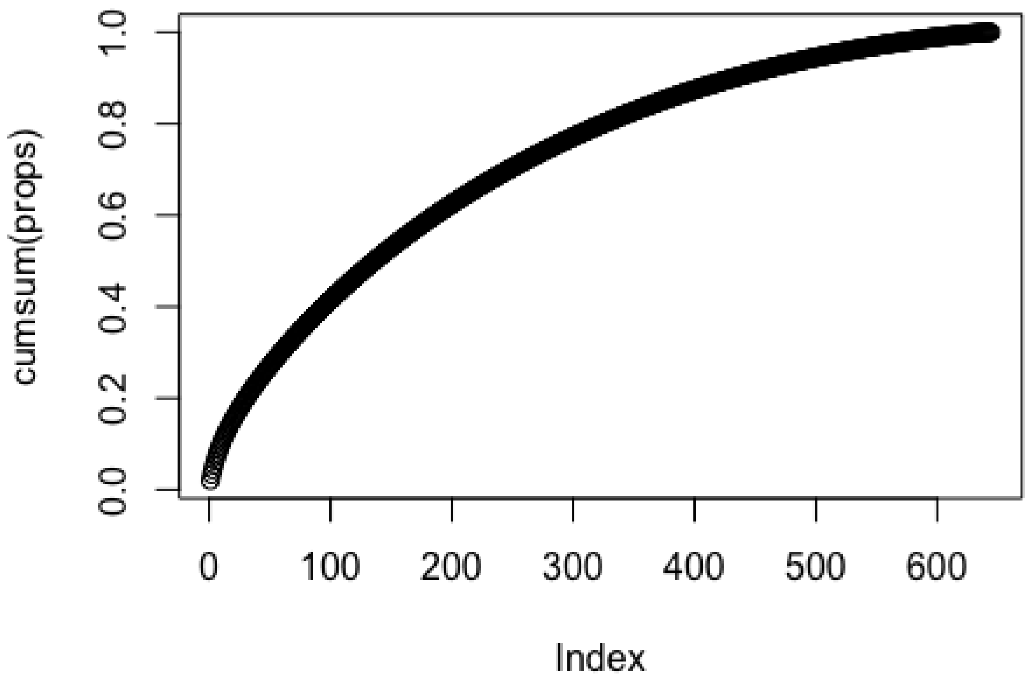
```
## <<DocumentTermMatrix (documents: 2500, terms: 660)>>
## Non-/sparse entries: 225031/1424969
## Sparsity           : 86%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

For the train data, after the pre-processing step, there are 2500 documents and 32241 words, with the sparsity 99%. We then dropped the term which only appears once or twice in the documents, trying to get rid of the long tail. Hence we removed those terms that have count 0 in at least 95% of docs. And it gives us 660 terms and Sparsity 89%. For the test data, after the pre-processing step, there are 2500 documents and 33048 words, with the sparsity 99%. We then redo the matrix process to make sure both train and test have 660 terms.

7.2.4 PCA to reduce dimension

7.2.4.1 Extract principle components

7.2.4.2 Choose Number of components



From the graph we can see that 200 principles can give us 60% of variance explained, so we will stop at 200/2500 documents.

7.2.4.3 Format and prepare the train and test data

7.3 Modeling:

7.3.1 Random Forest

```
```{r echo=FALSE,message=FALSE,error=FALSE}
set.seed(1)
RF_model<-randomForest(as.factor(author)~.,data=train, mtry=14,importance=TRUE)
predict_RF<-predict(RF_model,data=test)
predicted<-predict_RF
actual<-as.factor(test$author)
RFresult<-as.data.frame(cbind(actual,predicted))
RFresult$flag<-ifelse(RFresult$actual==RFresult$predicted,1,0)
sum(RFresult$flag)/nrow(RFresult)
```

[1] 0.7336
```

Note: This part of code might crash sometimes, depends on the different results from different seeds. The screenshot here is showing that we had this part of code work in our workspace, having a result of 0.7336.

7.3.2 KNN

```
## [1] 0.0196
```

7.3.3 Naive Bayes

```
## [1] 0.0336
```

7.4 Result

We conducted 3 different classifiers on the train data, which are Random Forest, KNN, and naive bayes. The accuracy for Random Forest classifier is 73.36%, the accuracy for KNN is 3.24%, and the accuracy for naive bayes is 4%. The random forest classifier performs way more better than the other two models. We have noticed that, while we have a reasonable level of accuracy for the random forest model, our accuracies for the other two models are extremely low, so the predicting power of these two models are insignificant. One possible explanation for this extremely low accuracy is that, since we used PCA components to reduce the dimensions, the values between different PCA variables are large. When we run KNN or Naive Bayes, it is hard to find a predictive trend within these variables. ## 7.5 Conclusion We are trying to predict the author of the article, and by using the PCA method and the Random Forest classifier, we have 73.36% accuracy on predicting the correct author based on the article.

Problem 8: Association Rule Mining

In this exercise, we are finding out any assocation rules and relationships among products that are commonly purchased together. After reading in the data as a table, there are a maximum of 4 variables in each row. Since each row representing a shopping basket, it means we can have a maximum of 4 products in each basket. However, the number of products in each basket is different, meaning if there are not four products in any specific basket, there are missing values in this row. In order to process the association rule code, we have to first clean the data into a executable form. We will split the data into a list of products for each customer

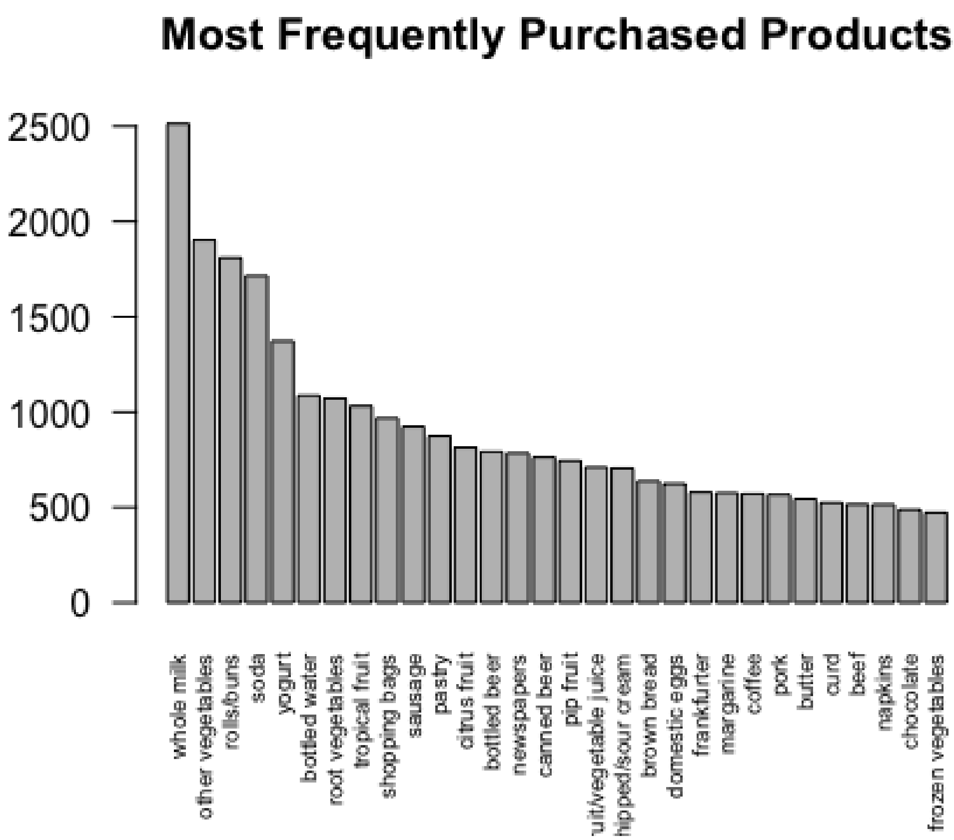
```
## 'data.frame':    43367 obs. of  2 variables:
##  $ customer: int  1 1 1 1 2 2 2 3 4 4 ...
##  $ value   : chr  "citrus fruit" "semi-finished bread" "margarine" "ready soups" ...
```

```
##     customer        value
##  Min.   :    1   Length:43367
##  1st Qu.: 3814   Class :character
##  Median : 7620   Mode  :character
##  Mean   : 7650
##  3rd Qu.:11482
##  Max.   :15296
```

In order to find the association shopping patterns among customers, we find the top 30 most frequently purchased products (same as finding a list of artists in the playlist example).

```
##    Length     Class      Mode 
##     43367 character character 

##  [1] "citrus fruit"             "semi-finished bread"     
##  [3] "margarine"                "ready soups"             
##  [5] "tropical fruit"           "yogurt"                  
##  [7] "coffee"                   "whole milk"              
##  [9] "pip fruit"                "yogurt"                  
## [11] "cream cheese "            "meat spreads"            
## [13] "other vegetables"         "whole milk"              
## [15] "condensed milk"           "long life bakery product"
## [17] "whole milk"               "butter"                  
## [19] "yogurt"                   "rice"                    
## [21] "abrasive cleaner"         "rolls/buns"              
## [23] "other vegetables"         "UHT-milk"                
## [25] "rolls/buns"               "bottled beer"            
## [27] "liquor (appetizer)"       "pot plants"              
## [29] "whole milk"               "cereals"
```



This graph displays the top 30 most frequently purchased products in the dataset. The most popular items are whole milk, other vegetables, rolls/buns, soda, and yogurt. Now, we will apply the apriori method to find the association rules related to these products.

```
## transactions as itemMatrix in sparse format with
##  15296 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.01677625
##
## most frequent items:
##       whole milk other vegetables       rolls/buns             soda
##             2513             1903             1809             1715
##           yogurt          (Other)
##             1372            34055
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4
## 3485 2630 2102 7079
##
##    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
##
## includes extended transaction information - examples:
##   transactionID
## 1             1
## 2             2
## 3             3
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5    0.01      1
##  maxlen target  ext
##       4  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 152
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 15296 transaction(s)] done [0.00s].
## sorting and recoding items ... [71 item(s)] done [0.00s].
```

```
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [45 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

##      lhs                     rhs                  support    confidence
## [1]  {}                   => {soda}               0.11212082 0.1121208
## [2]  {}                   => {rolls/buns}         0.11826621 0.1182662
## [3]  {}                   => {other vegetables}   0.12441161 0.1244116
## [4]  {}                   => {whole milk}         0.16429132 0.1642913
## [5]  {curd}               => {whole milk}         0.01261768 0.3683206
## [6]  {butter}             => {whole milk}         0.01438285 0.4036697
## [7]  {whipped/sour cream} => {whole milk}         0.01144090 0.2482270
## [8]  {pip fruit}          => {tropical fruit}     0.01268305 0.2607527
## [9]  {tropical fruit}     => {pip fruit}          0.01268305 0.1879845
## [10] {pip fruit}          => {other vegetables}   0.01091789 0.2244624
## [11] {pip fruit}          => {whole milk}         0.01255230 0.2580645
## [12] {pastry}             => {rolls/buns}         0.01019874 0.1782857
## [13] {citrus fruit}       => {tropical fruit}     0.01248692 0.2346437
## [14] {tropical fruit}     => {citrus fruit}       0.01248692 0.1850775
## [15] {citrus fruit}       => {other vegetables}   0.01281381 0.2407862
## [16] {other vegetables}   => {citrus fruit}       0.01281381 0.1029953
## [17] {citrus fruit}       => {whole milk}         0.01281381 0.2407862
## [18] {sausage}            => {rolls/buns}         0.01078713 0.1785714
## [19] {sausage}            => {other vegetables}   0.01261768 0.2088745
## [20] {other vegetables}   => {sausage}            0.01261768 0.1014188
## [21] {sausage}            => {whole milk}         0.01255230 0.2077922
## [22] {bottled water}      => {soda}               0.01464435 0.2060718
## [23] {soda}               => {bottled water}      0.01464435 0.1306122
## [24] {tropical fruit}     => {root vegetables}    0.01098326 0.1627907
## [25] {root vegetables}    => {tropical fruit}     0.01098326 0.1567164
## [26] {tropical fruit}     => {other vegetables}   0.01549425 0.2296512
## [27] {other vegetables}   => {tropical fruit}     0.01549425 0.1245402
## [28] {tropical fruit}     => {whole milk}         0.01830544 0.2713178
## [29] {whole milk}         => {tropical fruit}     0.01830544 0.1114206
## [30] {root vegetables}    => {other vegetables}   0.02536611 0.3619403
## [31] {other vegetables}   => {root vegetables}    0.02536611 0.2038886
## [32] {root vegetables}    => {whole milk}         0.02262029 0.3227612
## [33] {whole milk}         => {root vegetables}    0.02262029 0.1376840
## [34] {yogurt}             => {rolls/buns}         0.01189854 0.1326531
## [35] {rolls/buns}         => {yogurt}             0.01189854 0.1006081
## [36] {yogurt}             => {other vegetables}   0.01588651 0.1771137
## [37] {other vegetables}   => {yogurt}             0.01588651 0.1276931
## [38] {yogurt}             => {whole milk}         0.02425471 0.2704082
## [39] {whole milk}         => {yogurt}             0.02425471 0.1476323
## [40] {soda}               => {rolls/buns}         0.01425209 0.1271137
## [41] {rolls/buns}         => {soda}               0.01425209 0.1205086
## [42] {rolls/buns}         => {whole milk}         0.01830544 0.1547816
## [43] {whole milk}         => {rolls/buns}         0.01830544 0.1114206
## [44] {other vegetables}   => {whole milk}         0.04086036 0.3284288
```

```
## [45] {whole milk}        => {other vegetables} 0.04086036 0.2487067
##      coverage   lift     count
## [1]  1.00000000 1.000000 1715
## [2]  1.00000000 1.000000 1809
## [3]  1.00000000 1.000000 1903
## [4]  1.00000000 1.000000 2513
## [5]  0.03425732 2.241875  193
## [6]  0.03563023 2.457036  220
## [7]  0.04609048 1.510895  175
## [8]  0.04864017 3.864800  194
## [9]  0.06746862 3.864800  194
## [10] 0.04864017 1.804191  167
## [11] 0.04864017 1.570774  192
## [12] 0.05720450 1.507495  156
## [13] 0.05321653 3.477820  191
## [14] 0.06746862 3.477820  191
## [15] 0.05321653 1.935400  196
## [16] 0.12441161 1.935400  196
## [17] 0.05321653 1.465605  196
## [18] 0.06040795 1.509911  165
## [19] 0.06040795 1.678898  193
## [20] 0.12441161 1.678898  193
## [21] 0.06040795 1.264779  192
## [22] 0.07106433 1.837944  224
## [23] 0.11212082 1.837944  224
## [24] 0.06746862 2.322805  168
## [25] 0.07008368 2.322805  168
## [26] 0.06746862 1.845898  237
## [27] 0.12441161 1.845898  237
## [28] 0.06746862 1.651444  280
## [29] 0.16429132 1.651444  280
## [30] 0.07008368 2.909216  388
## [31] 0.12441161 2.909216  388
## [32] 0.07008368 1.964566  346
## [33] 0.16429132 1.964566  346
## [34] 0.08969665 1.121648  182
## [35] 0.11826621 1.121648  182
## [36] 0.08969665 1.423611  243
## [37] 0.12441161 1.423611  243
## [38] 0.08969665 1.645907  371
## [39] 0.16429132 1.645907  371
## [40] 0.11212082 1.074810  218
## [41] 0.11826621 1.074810  218
## [42] 0.11826621 0.942117  280
## [43] 0.16429132 0.942117  280
## [44] 0.12441161 1.999064  625
## [45] 0.16429132 1.999064  625
```

From the results above, we find out that there are 45 rules that meets the threshold. In this rule set, we used support of 0.01, confidence of 0.1, and maxlen of 4. Since support represents the percentage of groups that contain all the items listed in the rule, setting a support threshold of 0.01, we would like to find the rules that are relatively common while not being too strict to include a good amount of rules in the first run. Using a confidence level of 0.1, we would like to keep the conditional result of rule at a 10% level. We will further adjust the support and confidence to assess different rules associated with the threshold. We will keep maxlen at 4 since there are 4 items in a basket at max.

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.1    0.1    1 none FALSE            TRUE       5    0.02      1
##  maxlen target  ext
##       4  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 305
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 15296 transaction(s)] done [0.00s].
## sorting and recoding items ... [42 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [12 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
##      lhs                    rhs                  support    confidence covera
ge
## [1]  {}                  => {soda}              0.11212082 0.1121208  1.0000
0000
## [2]  {}                  => {rolls/buns}        0.11826621 0.1182662  1.0000
0000
## [3]  {}                  => {other vegetables} 0.12441161 0.1244116  1.0000
0000
## [4]  {}                  => {whole milk}        0.16429132 0.1642913  1.0000
0000
## [5]  {root vegetables}  => {other vegetables} 0.02536611 0.3619403  0.0700
8368
## [6]  {other vegetables} => {root vegetables}  0.02536611 0.2038886  0.1244
1161
## [7]  {root vegetables}  => {whole milk}        0.02262029 0.3227612  0.0700
8368
## [8]  {whole milk}       => {root vegetables}  0.02262029 0.1376840  0.1642
9132
## [9]  {yogurt}           => {whole milk}        0.02425471 0.2704082  0.0896
9665
## [10] {whole milk}       => {yogurt}            0.02425471 0.1476323  0.1642
```

```
9132
## [11] {other vegetables} => {whole milk}       0.04086036 0.3284288  0.1244
1161
## [12] {whole milk}       => {other vegetables} 0.04086036 0.2487067  0.1642
9132
##      lift     count
## [1]  1.000000 1715
## [2]  1.000000 1809
## [3]  1.000000 1903
## [4]  1.000000 2513
## [5]  2.909216  388
## [6]  2.909216  388
## [7]  1.964566  346
## [8]  1.964566  346
## [9]  1.645907  371
## [10] 1.645907  371
## [11] 1.999064  625
## [12] 1.999064  625
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.2    0.1    1 none FALSE            TRUE       5    0.02      1
##  maxlen target  ext
##       4  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 305
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 15296 transaction(s)] done [0.00s].
## sorting and recoding items ... [42 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [6 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
##     lhs                     rhs                support    confidence coverag
e
## [1] {root vegetables}  => {other vegetables} 0.02536611 0.3619403  0.07008
368
## [2] {other vegetables} => {root vegetables}  0.02536611 0.2038886  0.12441
161
## [3] {root vegetables}  => {whole milk}       0.02262029 0.3227612  0.07008
368
## [4] {yogurt}           => {whole milk}       0.02425471 0.2704082  0.08969
```

```
665
## [5] {other vegetables} => {whole milk}       0.04086036 0.3284288  0.12441
161
## [6] {whole milk}       => {other vegetables} 0.04086036 0.2487067  0.16429
132
##     lift     count
## [1] 2.909216 388
## [2] 2.909216 388
## [3] 1.964566 346
## [4] 1.645907 371
## [5] 1.999064 625
## [6] 1.999064 625
```

```
##      lhs                     rhs                support    confidence covera
ge
## [1]  {}                   => {soda}             0.11212082 0.1121208  1.0000
0000
## [2]  {}                   => {rolls/buns}       0.11826621 0.1182662  1.0000
0000
## [3]  {}                   => {other vegetables} 0.12441161 0.1244116  1.0000
0000
## [4]  {}                   => {whole milk}       0.16429132 0.1642913  1.0000
0000
## [5]  {tropical fruit}   => {other vegetables} 0.01549425 0.2296512  0.0674
6862
## [6]  {other vegetables} => {tropical fruit}   0.01549425 0.1245402  0.1244
1161
## [7]  {tropical fruit}   => {whole milk}       0.01830544 0.2713178  0.0674
6862
## [8]  {whole milk}       => {tropical fruit}   0.01830544 0.1114206  0.1642
9132
## [9]  {root vegetables}  => {other vegetables} 0.02536611 0.3619403  0.0700
8368
## [10] {other vegetables} => {root vegetables}  0.02536611 0.2038886  0.1244
1161
## [11] {root vegetables}  => {whole milk}       0.02262029 0.3227612  0.0700
8368
## [12] {whole milk}       => {root vegetables}  0.02262029 0.1376840  0.1642
9132
## [13] {yogurt}           => {other vegetables} 0.01588651 0.1771137  0.0896
9665
## [14] {other vegetables} => {yogurt}           0.01588651 0.1276931  0.1244
1161
## [15] {yogurt}           => {whole milk}       0.02425471 0.2704082  0.0896
9665
## [16] {whole milk}       => {yogurt}           0.02425471 0.1476323  0.1642
9132
## [17] {rolls/buns}       => {whole milk}       0.01830544 0.1547816  0.1182
6621
## [18] {whole milk}       => {rolls/buns}       0.01830544 0.1114206  0.1642
```

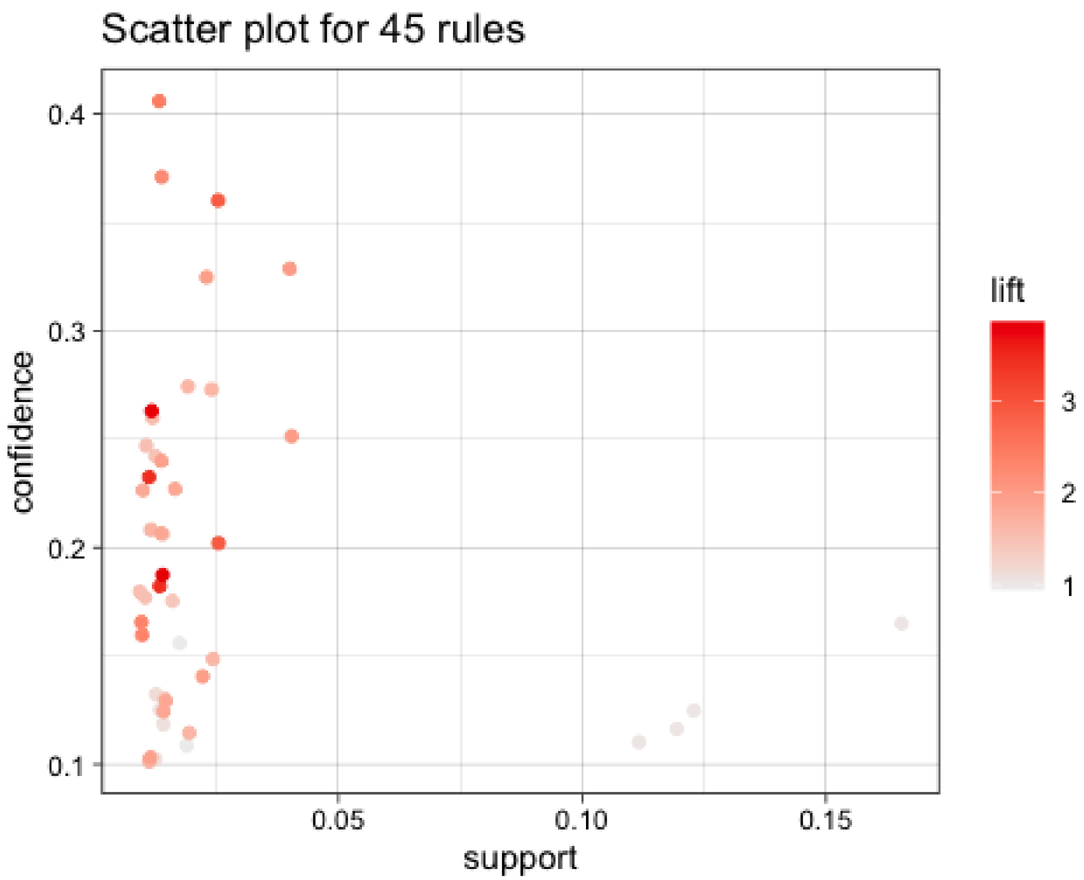
```
9132
## [19] {other vegetables} => {whole milk}       0.04086036 0.3284288  0.1244
1161
## [20] {whole milk}       => {other vegetables} 0.04086036 0.2487067  0.1642
9132
##      lift     count
## [1]  1.000000 1715
## [2]  1.000000 1809
## [3]  1.000000 1903
## [4]  1.000000 2513
## [5]  1.845898  237
## [6]  1.845898  237
## [7]  1.651444  280
## [8]  1.651444  280
## [9]  2.909216  388
## [10] 2.909216  388
## [11] 1.964566  346
## [12] 1.964566  346
## [13] 1.423611  243
## [14] 1.423611  243
## [15] 1.645907  371
## [16] 1.645907  371
## [17] 0.942117  280
## [18] 0.942117  280
## [19] 1.999064  625
## [20] 1.999064  625
```

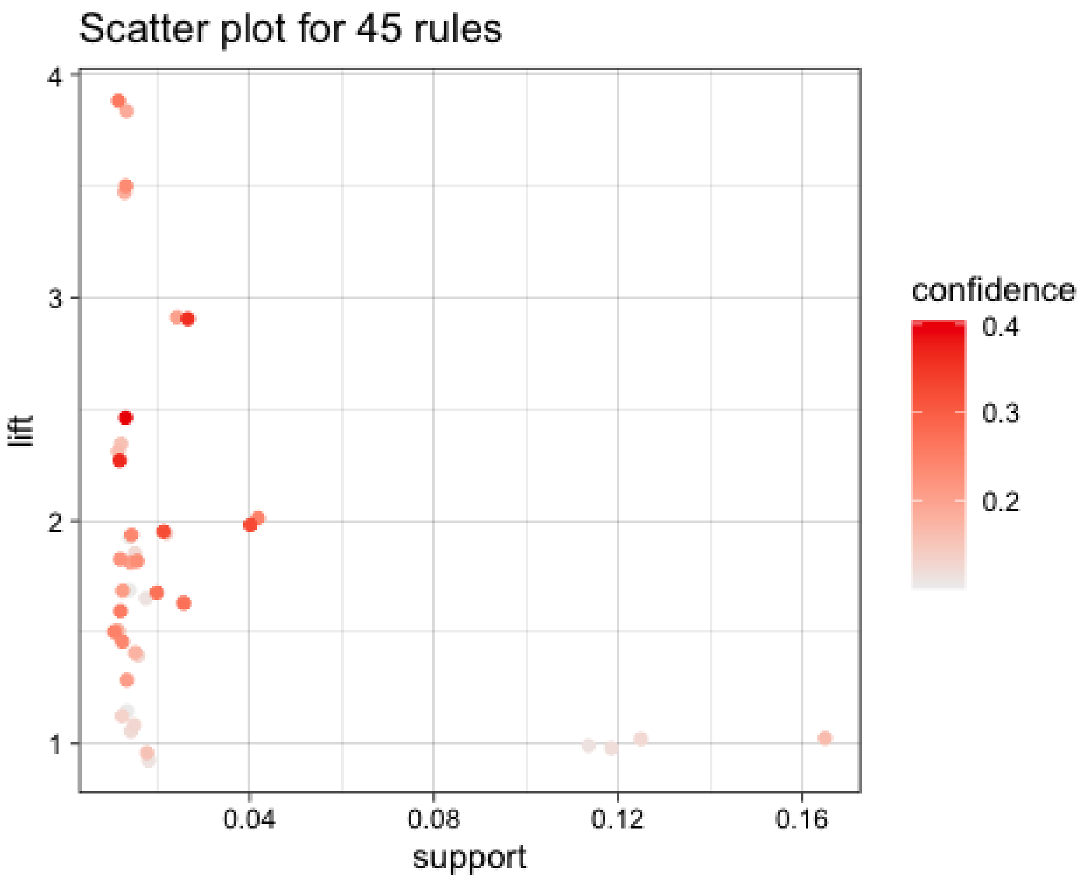
```
##     lhs                    rhs                support    confidence coverag
e
## [1] {curd}             => {whole milk}       0.01261768 0.3683206  0.03425
732
## [2] {butter}           => {whole milk}       0.01438285 0.4036697  0.03563
023
## [3] {root vegetables}  => {other vegetables} 0.02536611 0.3619403  0.07008
368
## [4] {root vegetables}  => {whole milk}       0.02262029 0.3227612  0.07008
368
## [5] {other vegetables} => {whole milk}       0.04086036 0.3284288  0.12441
161
##     lift     count
## [1] 2.241875 193
## [2] 2.457036 220
## [3] 2.909216 388
## [4] 1.964566 346
## [5] 1.999064 625
```

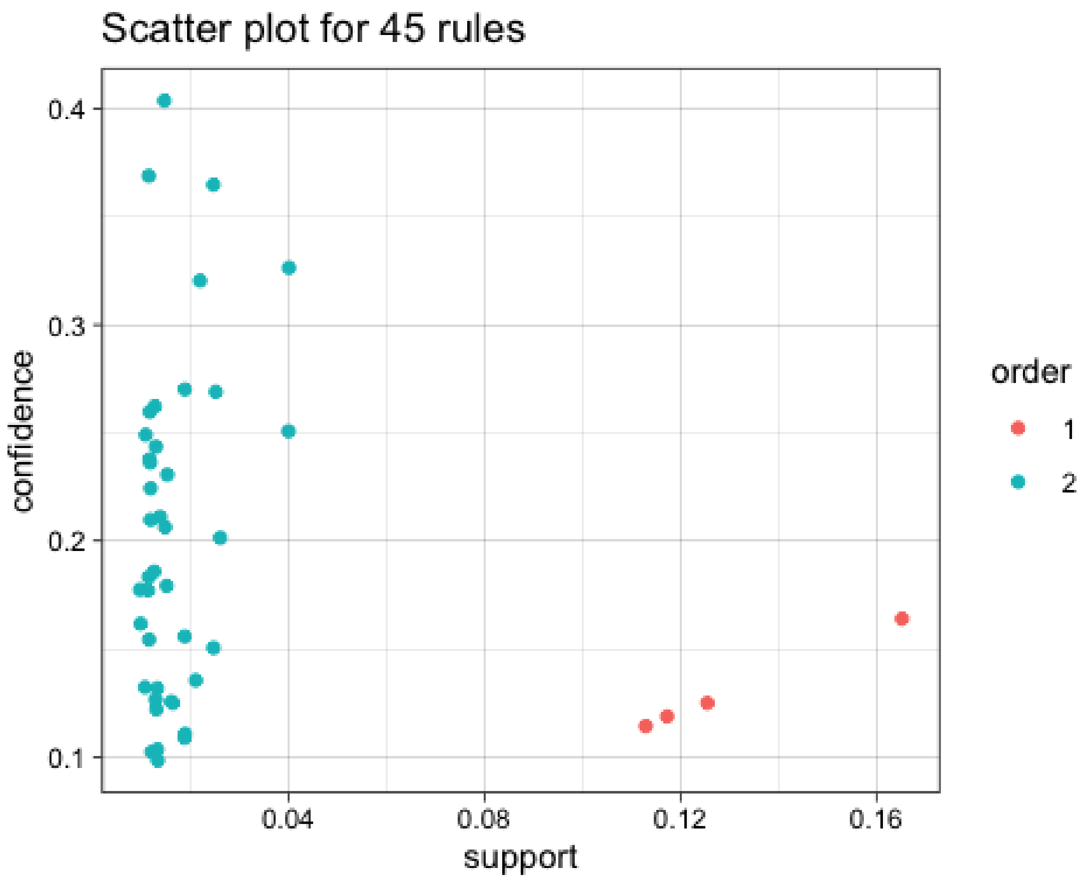
```
##     lhs                  rhs                support    confidence coverage
## [1] {pip fruit}      => {tropical fruit} 0.01268305 0.2607527  0.04864017
## [2] {tropical fruit} => {pip fruit}      0.01268305 0.1879845  0.06746862
## [3] {citrus fruit}   => {tropical fruit} 0.01248692 0.2346437  0.05321653
```

```
## [4] {tropical fruit} => {citrus fruit}   0.01248692 0.1850775  0.06746862
##      lift    count
## [1] 3.86480 194
## [2] 3.86480 194
## [3] 3.47782 191
## [4] 3.47782 191
```

We have increased the support to 0.02, and the resulting association rules come down to 12 different combination. And after raising the confidence to 0.2, the number of resulting association rules has been eliminated to 6. We have also tried adjusting the lift threshold. We set the lift threshold to be 3. Lift is a measure of how much more likely would the result hold given the condition as compared to a customer drawn at random. After setting this threshold, the number of rules come down to 4, and they are all related to pip fruit, tropical fruit, and citrus fruit. For example, if a customer buys pip fruit, he or she is more likely to buy tropical fruit with a lift of 3.86.



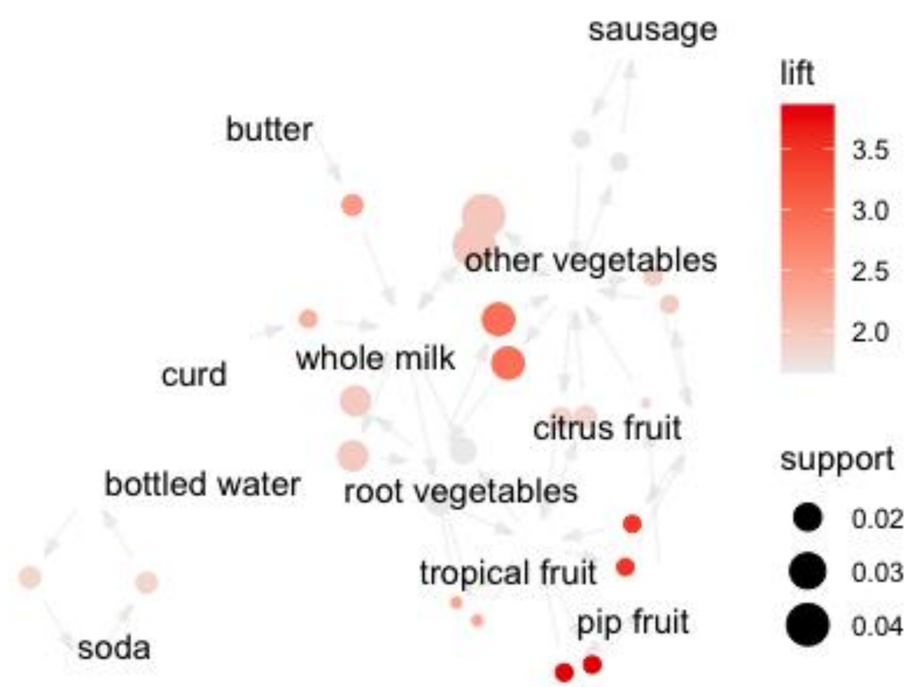

Scatter plot for 45 rules
lift
support
confidence
4
3
2
1
0.04
0.08
0.12
0.16
0.4
0.3
0.2



Every dot on the plot represents an association rule. From the support vs confidence plot, we observed that high lift rules tend to have lower support. However, with the limited amount of rules, the correlation is not very obvious. From the two-key plot, we observe that the level 1 rules tend to have higher support and lower confidence, and level 2 rules tend to have lower support values.

```
## set of 45 rules
##
## rule length distribution (lhs + rhs):sizes
##  1  2
##  4 41
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   1.911   2.000   2.000
##
## summary of quality measures:
##     support           confidence        coverage             lift
##  Min.   :0.01020   Min.   :0.1006   Min.   :0.03426   Min.   :0.9421
##  1st Qu.:0.01255   1st Qu.:0.1277   1st Qu.:0.06041   1st Qu.:1.4236
##  Median :0.01438   Median :0.1786   Median :0.08970   Median :1.6789
##  Mean   :0.02654   Mean   :0.1959   Mean   :0.17099   Mean   :1.8383
##  3rd Qu.:0.02262   3rd Qu.:0.2408   3rd Qu.:0.12441   3rd Qu.:1.9991
##  Max.   :0.16429   Max.   :0.4037   Max.   :1.00000   Max.   :3.8648
```

```
##      count
##  Min.   : 156
##  1st Qu.: 192
##  Median : 220
##  Mean   : 406
##  3rd Qu.: 346
##  Max.   :2513
##
## mining info:
##              data ntransactions support confidence
##  groceries_trans         15296    0.01        0.1
##
call
##  apriori(data = groceries_trans, parameter = list(support = 0.01, confiden
ce = 0.1, maxlen = 4))
```



In conclusion, we found out that whole milk, root vegetables, and other vegetables are highly associated products, while they are also the top frequently bought items. These items are more related to basic and daily necessities. As the marketing strategy, in order to generate higher sales, we can try to place the items in the same association rule closer to boost customers' sales on these products. In addition, many of these products can be complementary, such as yogurt and fruit, fruit and vegetables, cheese and milk, etc. Thus, grocery stores can place these items closer so that when customers are shopping around, they are more likely to buy the other when they buy one. By learning the purchasing pattern from association rules in shopping basket, grocery stores can further design their marketing strategies and shelf arrangement.