PART -B

1. Introduction

The global beer market is projected to grow from $768.17 billion in 2021 to $989.48 billion in 2028 at a CAGR of 3.68% in the forecast period, 2021-2028 (fortune business insights). The beer industry has suffered during the COVID-19 pandemic due to low demand and supply chain issues. The report suggests that the demand will return to pre-covid levels in the next 7 years. BrewDog (a multinational brewery and pub chain) wants to be prepared for the forecasted increase in demand and growth and they want to market beers that customers are more likely to consume. If they are not able to do the marketing according to the customer interests, they will fall behind their competitors.

1.1 The way forward: Cluster Analysis

Cluster analysis is a data-reduction technique designed to uncover subgroups of observations within a dataset (Kabacoff, et al., (2015) p.331). This is done by calculating the similarity and the distance (Euclidean method) between them. There are mainly 2 types of clustering algorithms: Hierarchical and Non- Hierarchical. Based on cluster analysis, Marketing campaigns are tailored to appeal to the customers by BrewDog.

2. Missing Data:

Data Analysts spend 80% of their time on data cleaning and missing data is one of the most common issues when dealing with any dataset during data cleaning. Data can be missing due to incomplete data entry, equipment malfunctions, lost files, The method of data capture being changed, and many other reasons.

The reason being focused on missing values is because these can create a bias in our results. so, this makes our results inaccurate and not reliable.

Missing data is classified as follows:

Missing Completely at Random (MCAR)	If the presence of missing data is unrelated to observed or unobserved values
Missing at Random (MAR)	If the presence of data is related to observed values and missing data pattern can be established
Missing Not at Random (MNAR)	Missingness depends on the unobserved values of the data set

2.1 Handling Missing data

There are various methods to handle missing data such as Deletion, Maximum likelihood estimation, and imputing the values. The figure below describes the methods for handling missing data.

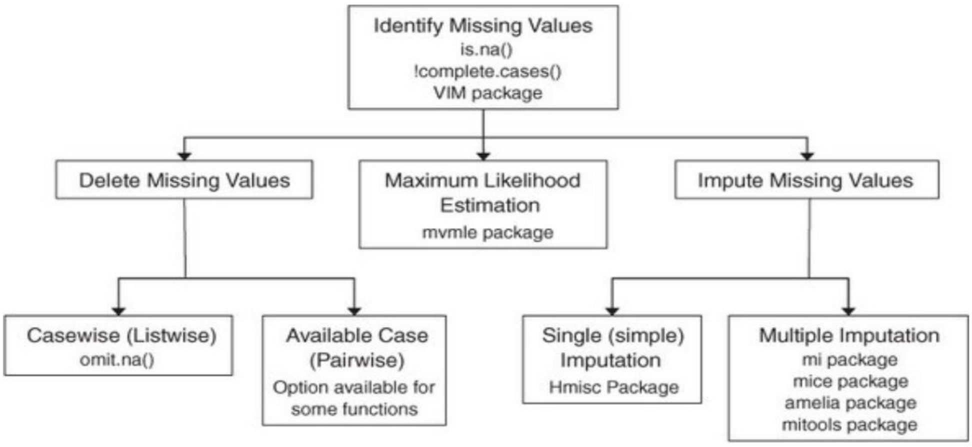


Figure 2.1 Handling missing data methods (Kabacoff, et al., (2015), p.366, Figure 18.1)

2.1.1 Deletion: This is one of the basic and simple method used to handle missing data. In this, you would delete the data either partially or completely based on the analysis. Deletion can be done using **Listwise and Pairwise methods**.

Listwise:

- We can only analyse the data rows where there is complete data for every column and delete every row which consists of a missing value
- It is simple and results could be biased if the data is not MCAR
- In R the function is **complete.cases( )** can be used to retain the data without missing values

Pairwise:

- We can only analyse the data rows where the variables of interest have data present and delete the missing value only if it is required in that specific analysis
- The advantage of this is it uses all the possible information

2.1.2 Maximum likelihood estimate: This method uses each case available data to compute maximum likelihood estimates that are most likely to have resulted in the observed data. We can use *mvmic* package in R for using this method. This method is only applicable to linear models.

2.1.3 Imputation: It is the process of replacing the missing values by substituting them using statistical methods**.** This method keeps the full sample size, and it is best for precision. The two most common methods for replacing missing data are simple imputation and multiple imputation.

Simple Imputation:

- In this the missing values in a variable are replaced with statistical variables like Mean, Mode, Median
- It is simple and doesn't reduce the sample size available for analysis
- This method underestimates the standard errors and distorts correlations
- Possibility of bias if data is not MCAR

Multiple Imputation:

- It estimates the missing data by repeated simulations.
- This is generally done using the **chained equation approach**
- Algorithms are very complex, but R provides good packages such as *mice, mi, amelia, mitools.*

2.2 Implementing Imputation

We should identify the missing values in the dataset as they may contain Null values, Nan values, and infinity values. We have the **dplyr package** in R, so It is possible to use the **is.na(), is.null() and is.infinite()** functions in R to identify missing, empty and infinite values in the dataset. We should also check for **outliers** if any are present in the variables then remove them and then go ahead with imputing the data.

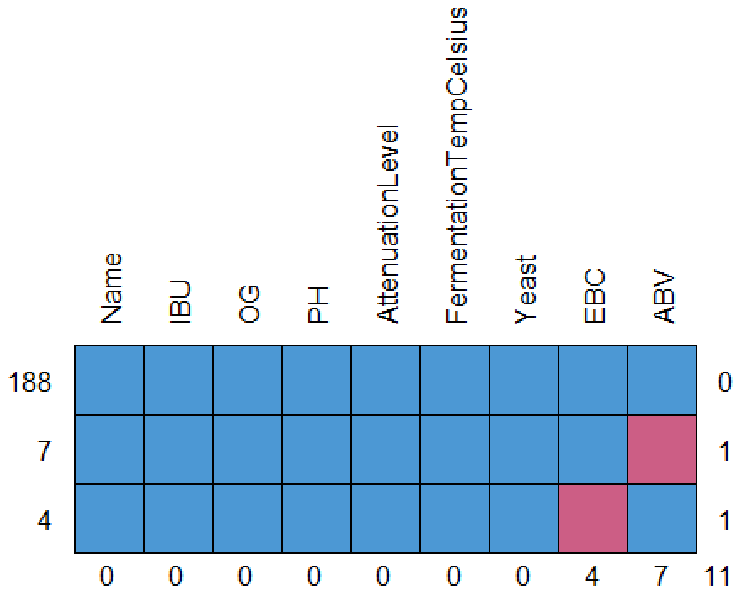


Figure 2.2 Matrix representing the Missing data in the dataset (**11 data missing**)

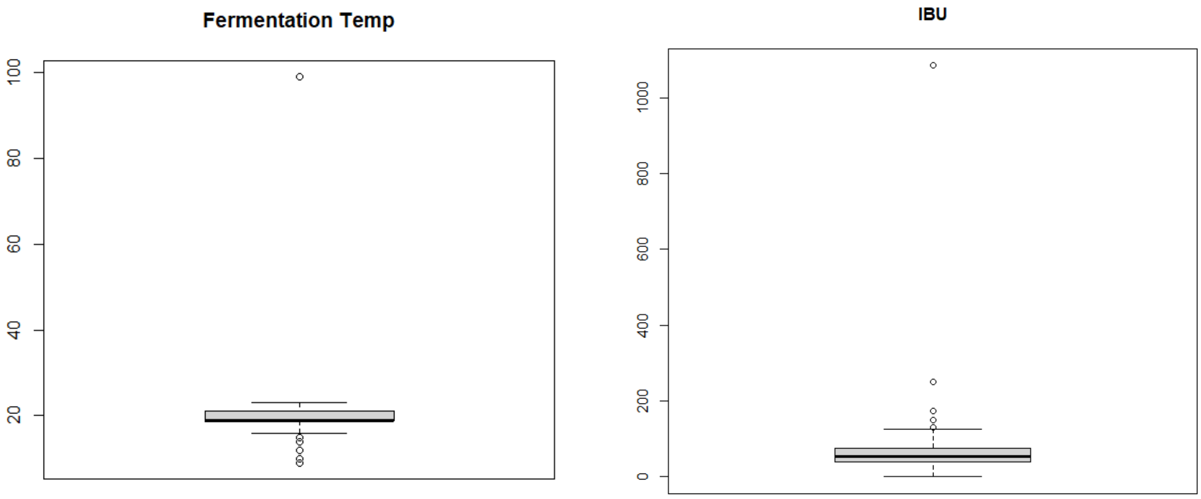


Figure 2.3 Box plot of variables Fermentation Temperature and IBU

From the above Figure, the outliers were removed as they are way above the normal range. In Fermentation Temperature and IBU, the outliers are 99 °C and 1085 respectively which are way above the range and in the remaining variables, all are in the range.

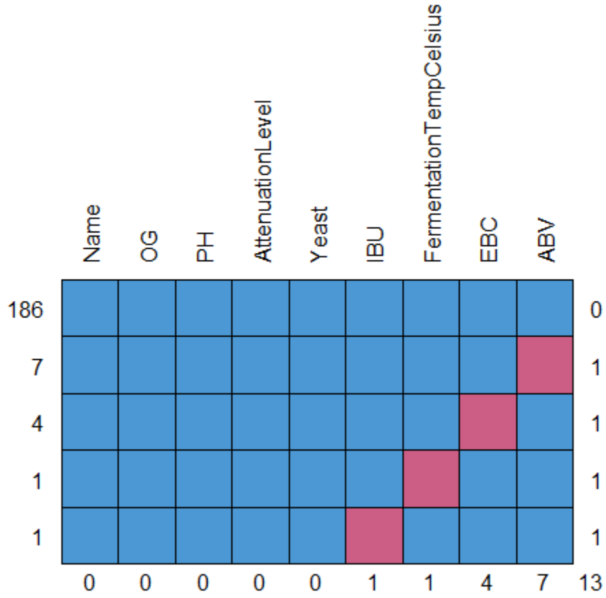


Figure 2.4 Plots representing the missing data after removing outliers (**13 missing data**)

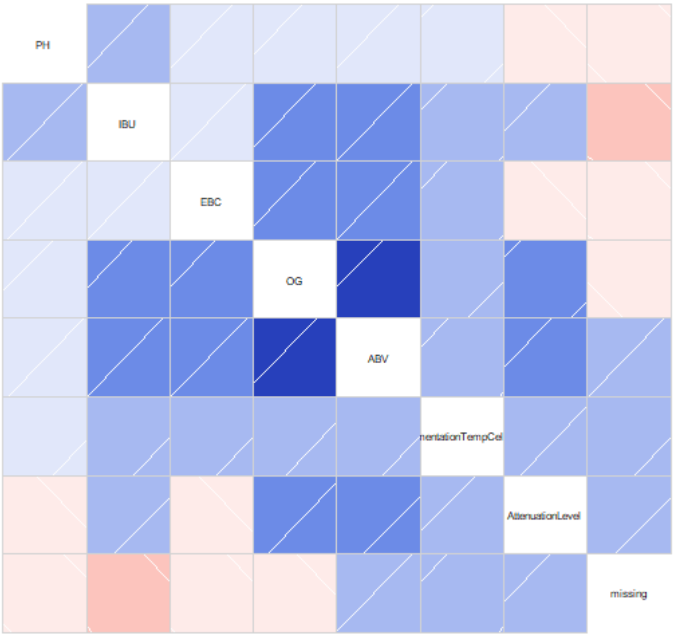


Figure 2.5 correlation matrix generated using Corrgram library in R

The correlation between the missing variables and the variables **EBC (r = -0.13), ABV (r = 0.23), IBU (r =-0.19), and fermentationTempCelsius (r= 0.20).** None of the correlations are visibly large and it suggests that missing data may slightly deviate from MCAR and may be MAR. (Kabacoff, et al., (2015), p.371)

Since there might be a slight possibility of missing data being MAR, so we are using the Multiple Imputation because of the possibility of bias in Simple Imputation

We will use the **mice** package in R, there are multiple methods, but we are going to use Random Forest, CART, and PMM as these are suitable for our missing dataset (numeric). Based on how close these mean values are to the original dataset we are going to select a model out of these 3.

Variable name	Dataset mean (without Missing values)	Imputed values mean using 3 different methods		
		CART	Predictive Mean Matching (PMM)	Random Forest(RF)
ABV	7.675	7.613	7.719	7.683
EBC	71.66	70.6	70.62	71.85
IBU	62.34	62.46	62.11	62.38
FermentationTempCelsius	18.95	18.95	18.96	18.95

Figure 2.6 Imputed values mean comparison with different methods used in the model

The value of **m = 10** i.e., the number of data sets to be generated from the existing dataset, and **maxit =10** i.e., the number of iterations for calculating imputation.

From above Figure 2.6, we can infer that **Random Forest is the best method** for imputation as it is closely matching with the original, and from Figure 2.7 we can infer that the **distribution almost remains the same** for imputed values using RF.

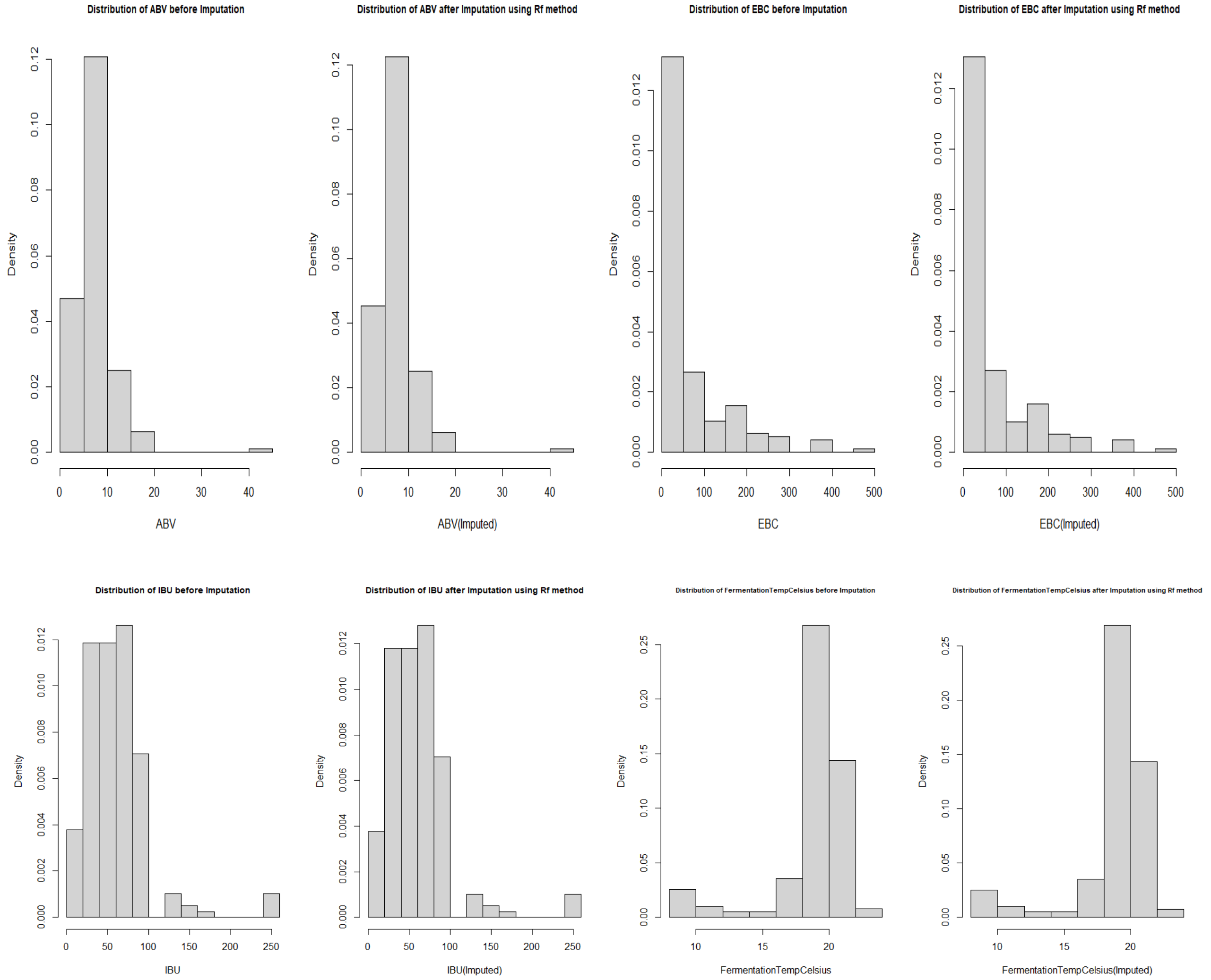
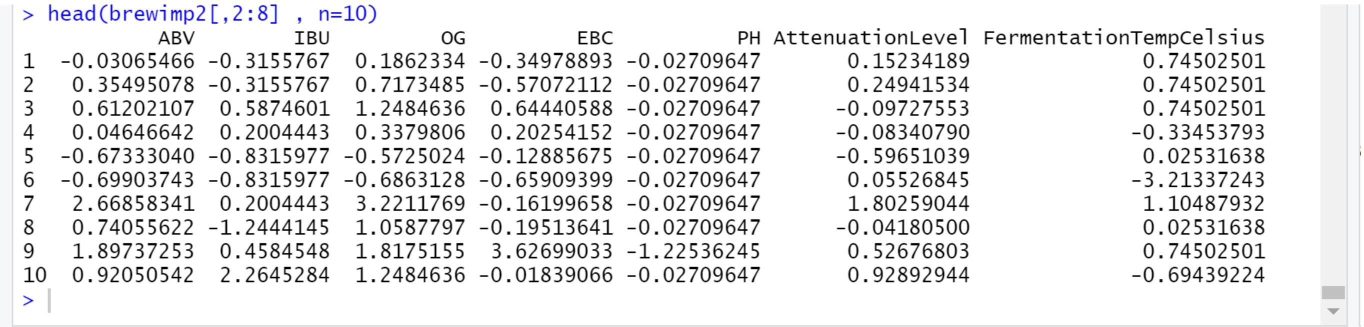


Figure 2.7 Distributions of original and Imputed variables

3. Clustering Algorithm

Step One: Choosing the variables, in our dataset, we will be choosing Alcohol by volume (**ABV**), International Bitterness Units (**IBU**), Original Gravity (**OG**), Colour Units from the European Brewery Convention (**EBC**), **pH** (Acid & Base Scale), **Attenuation Level**, **FermentationtempCelsius** and **Yeast type** as these represent the variability of beers.

Step Two: Scaling the data to nullify the impact of the large range of data. We will be using the **scale()** function



```
> head(brewimp2[,2:8] , n=10)
           ABV        IBU         OG         EBC          PH AttenuationLevel FermentationTempCelsius
1  -0.03065466 -0.3155767  0.1862334 -0.34978893 -0.02709647       0.15234189              0.74502501
2   0.35495078 -0.3155767  0.7173485 -0.57072112 -0.02709647       0.24941534              0.74502501
3   0.61202107  0.5874601  1.2484636  0.64440588 -0.02709647      -0.09727553              0.74502501
4   0.04646642  0.2004443  0.3379806  0.20254152 -0.02709647      -0.08340790             -0.33453793
5  -0.67333040 -0.8315977 -0.5725024 -0.12885675 -0.02709647      -0.59651039              0.02531638
6  -0.69903743 -0.8315977 -0.6863128 -0.65909399 -0.02709647       0.05526845             -3.21337243
7   2.66858341  0.2004443  3.2211769 -0.16199658 -0.02709647       1.80259044              1.10487932
8   0.74055622 -1.2444145  1.0587797 -0.19513641 -0.02709647      -0.04180500              0.02531638
9   1.89737253  0.4584548  1.8175155  3.62699033 -1.22536245       0.52676803              0.74502501
10  0.92050542  2.2645284  1.2484636 -0.01839066 -0.02709647       0.92892944             -0.69439224
> 
```

Figure 3.1 Scaled variables

Step Three: Checking for **outliers** and removing them as they can **misrepresent** the results (Refer to Figure 2.3)

Step Four: Calculate the Euclidean distance between the variables using the **daisy()** function.

Step Five: Selecting cluster algorithm

Agglomerative hierarchical clustering is used. The reasons are as follows

- the size of the dataset is not large,
- Lack of prior knowledge of the K value (no of clusters)

We will be using **agnes()** because of the presence of other variable types and the method is **ward** for calculating distance.

Step Six: Calculate the number of clusters using the nbclust

Step Seven: Visualize the results using a dendrogram.

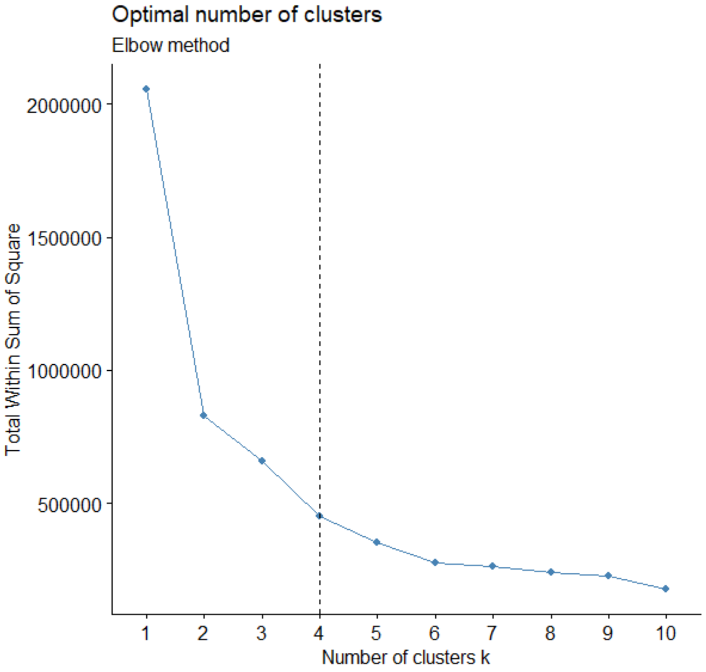


Figure 3.2 Optimal Number of clusters

3.1 Dendrogram

As per Figure 3.2, the dendrogram has been divided into 4 clusters

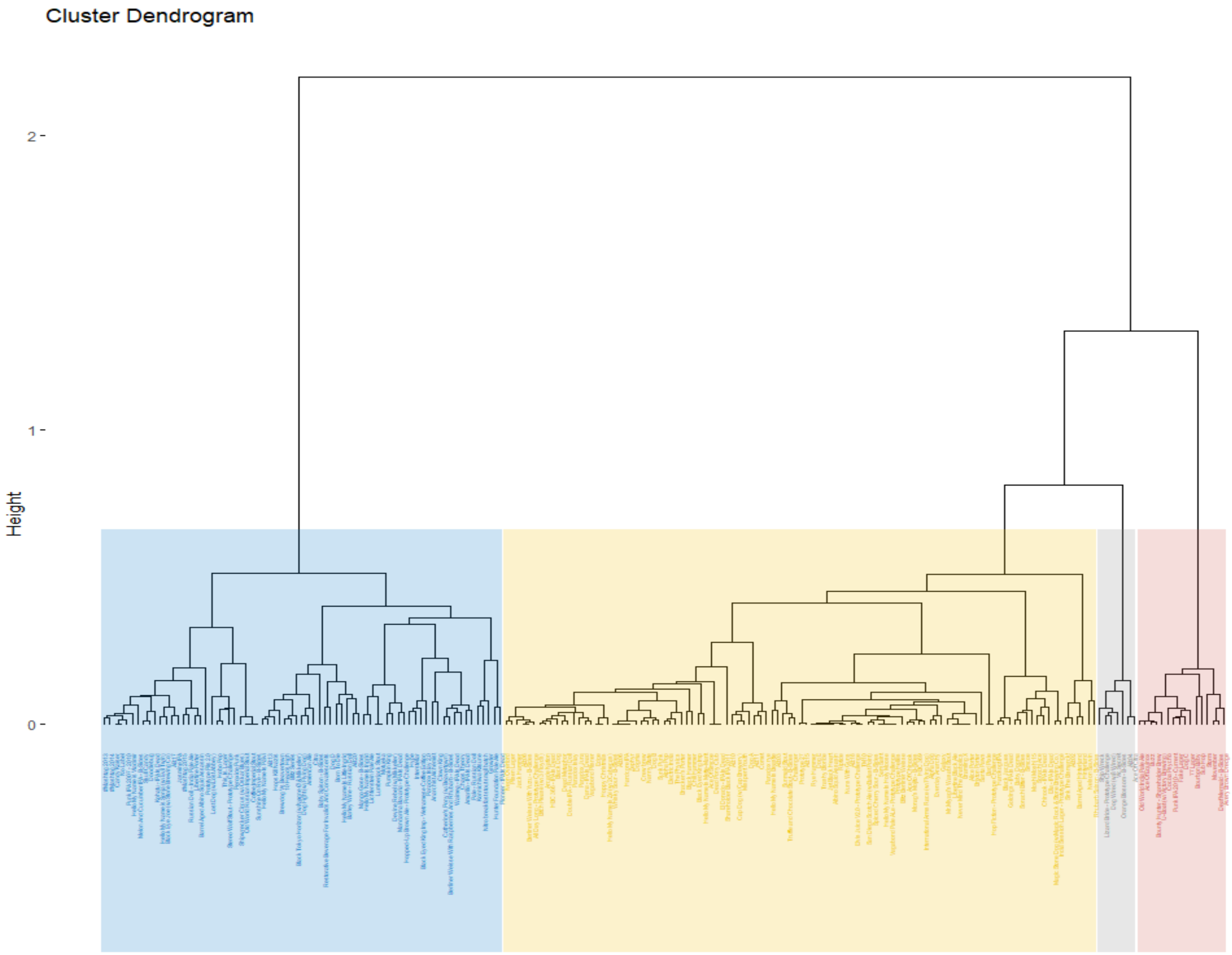


Figure 3.3 Dendrogram with 4 clusters

The clades(branches) arrangement will tell us about the variables that are most similar to each other and the height of it will explain the similarities or dissimilarities from each other. Based on this we can interpret the following,

- cluster 1 (Blue) is different from the other clusters because of the vertical height of the clade.
- Cluster 2(Yellow) and cluster 3(grey) are more similar to each other

Variable	Cluster 1 (Blue)	Cluster 2 (yellow)	Cluster 3 (Grey)	Cluster 4 (Red)
ABV	10.496197	6.050476	5.818750	7.757143
IBU	83.14789	53.74762	41.56250	30.28571
ECB	119.85211	48.72000	29.62500	12.28571

Table 3.1 Mean values of clusters

Based on cluster information we have the data regarding the beer classification

Yeast type	Number of Beers
Wyeast 1056 - American Ale	105
Wyeast 1272 - American Ale II	71
Wyeast 2007 - Pilsen Lager	16
Wyeast 3711 - French Saison	7

Table 3.2 Beer classification according to Yeast type

Both statistical and visual analysis strongly supports the above inferences.

References

1.T. Saaty*, A scaling method for priorities in hierarchical structures*, Journal of Mathematical Psychology, 15 (1977), Pg 234-281.

2. T. Saaty, *The Analytic Hierarchy Process*, McGraw-Hill, New York, 1980.

3. Fortune Business Insights. 2022, *Market Research Report* [Online],[Accessed 6 Jan 2023],

Available From: https://www.fortunebusinessinsights.com/beer-market-102489

4. Kabacoff, Robert I.. *R in Action: Data Analysis and Graphics with R*, Manning Publications Co. LLC, 2015.

ProQuest Ebook Central,[Accessed 7 January 2023],

Available From: http://ebookcentral.proquest.com/lib/leeds/detail.action?docID=6642825.