# TABLE OF CONTENTS

# PART – 1
# WAREHOUSE PROJECT

# 1 INTRODUCTION

A small manufacturing company wants to expand their space further, so they have produced a set of options and alternatives. We will help them choose the best way to achieve the goal from their options.

| | Alternative | | | | |
|---|---|---|---|---|---|
| | | A1 - Centre | A2 - Suburb | A3 - Shared | A4 - Extend |
| | C1 – Public transport links | Good bus<br>Good rail | Good bus<br>No rail | Poor bus<br>Good rail | Excellent bus<br>Excellent rail |
| | C2 - Parking | Poor | Good | Excellent | Moderate |
| | C3 – Warehouse space | Poor | Excellent | Good | Good |
| | C4 - Security | *** | **** | *** | ** |
| | C5 - Cost | £800,000 | £600,000 | £300,000 | £250,000 |

Fig 1 Alternatives and criteria

As per Fig 1, Four alternatives and five criteria are provided based on which best alternative must be selected. The manufacturing company has done a fair amount of research by finding out all the data related to the provided alternatives concerning all the criteria. We now must choose the best of the above using the data provided and suggest the best viable alternative with proper analysis.

## 1.1 MCDA METHOD

MCDA methods are beneficial in scenarios with multiple conflicting criteria like our current situation. MCDA evaluates the performance of every alternative concerning the criteria and helps us make decisions appropriately. It is a decision support tool that helps set priority for the alternatives. We will use two popular MCDA methods AHP and TOPSIS, to devise the best possible solution to help the manufacturing company acquire extra space. Below is the snapshot of MCDA-related terms and definitions.

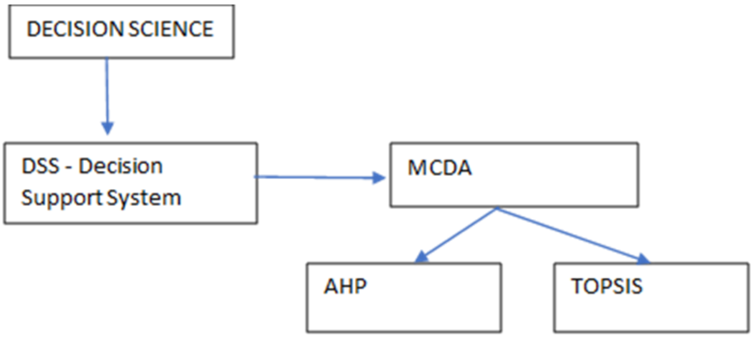


Fig 2 MCDA

AHP & TOPSIS are MCDA methods that we will be discussing in detail for our given case study.

AHP – Analytical Hierarchy Process (AHP). It was developed by Thomas L. Saaty. It calculates priorities for the available criteria and alternatives and helps decide based on paired comparisons.

TOPSIS is abbreviated as Technique for Order Preference by Similarity to Ideal Solution. This method chooses a list of available alternatives whose Euclidean distance is shortest from a positive ideal solution and furthest from a negative ideal solution.

# 2. AHP AND TOPSIS ANALYSIS

### 2.1.1 CRITERIA WEIGHTS REASONING

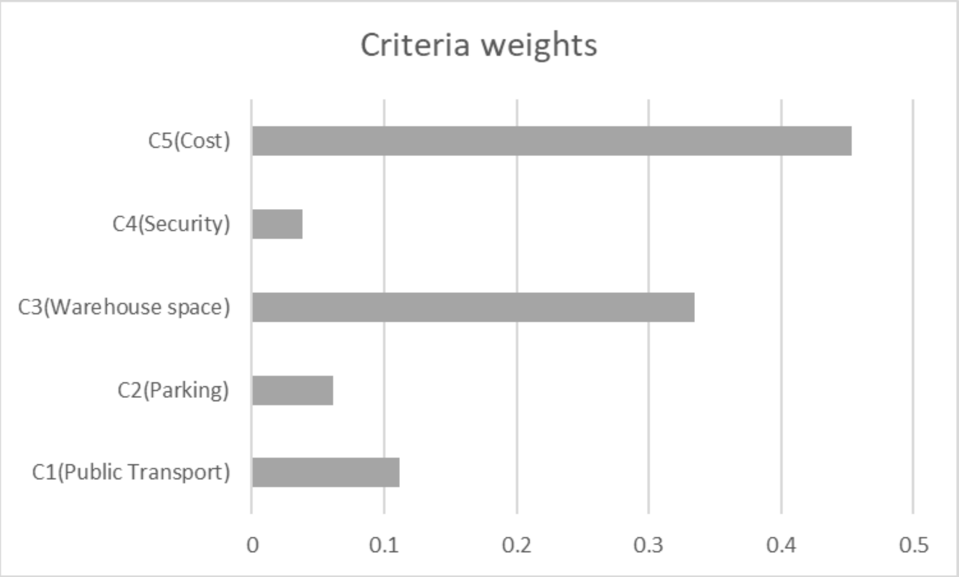


Fig 3 Criteria weights

Considering this is a manufacturing company whose primary goal is to extend space; we need to have enough money to proceed with our goal. So, as Fig 3 states, the cost is given the highest weight here, followed by warehouse space which is our primary goal. Transport is the next priority because a manufacturing company needs to be nearer to the market. It eventually can reduce the transport cost and aim for profit which is our goal. Followed by Transport, we will need a parking space in the evening to transport it to the market once our manufacturing is finished. Hence, parking is ranked fourth. Since this is a small manufacturing company, we can use existing resources for security instead of added resources, so security is ranked last.

We will follow the above weight criteria for our two analyses.

### 2.2.1 ANALYTICAL HEIRARRCHY PROCESS – AHP

Firstly, we will use AHP to find the best possible solution to help us acquire extra space. AHP is a step-by-step process, as explained below.

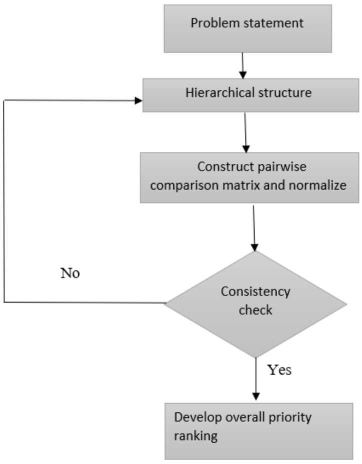

Problem statement
Hierarchical structure
Construct pairwise comparison matrix and normalize
Consistency check
No
Yes
Develop overall priority ranking

Fig 4 AHP process

Before proceeding with the analysis, we will decompose our case study into a hierarchy which is our first step after analysing the problem statement.

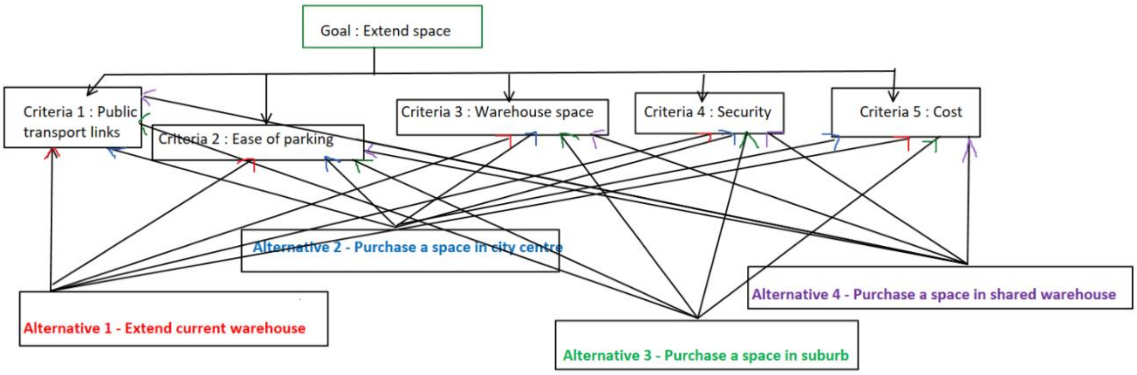
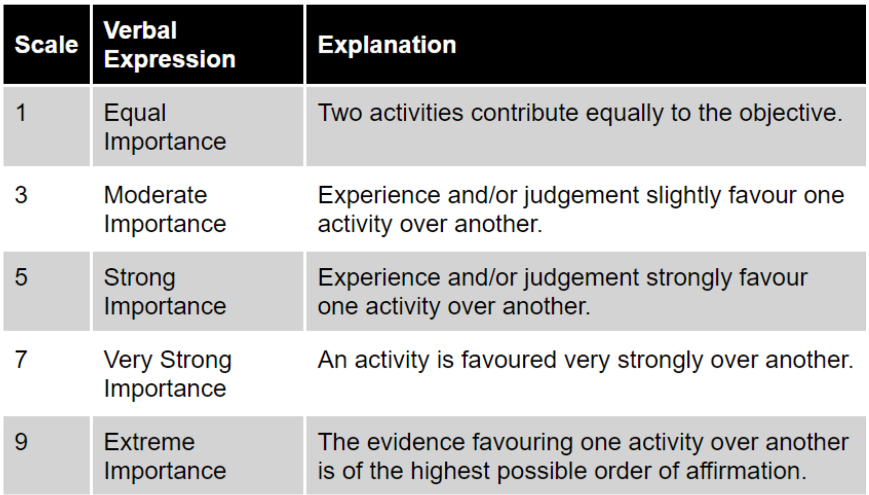

Goal : Extend space
Criteria 1 : Public transport links
Criteria 2 : Ease of parking
Criteria 3 : Warehouse space
Criteria 4 : Security
Criteria 5 : Cost
Alternative 2 - Purchase a space in city centre
Alternative 4 - Purchase a space in shared warehouse
Alternative 1 - Extend current warehouse
Alternative 3 - Purchase a space in suburb

Fig 5 Hierarchical structure

After finalizing the structure, we must construct a pairwise comparison matrix. We need an ideal scale to compare each alternative concerning the criteria. T.L.Saaty developed a scale for such comparisons, which will help us rank our alternatives and criteria.



| Scale | Verbal Expression | Explanation |
|---|---|---|
| 1 | Equal Importance | Two activities contribute equally to the objective. |
| 3 | Moderate Importance | Experience and/or judgement slightly favour one activity over another. |
| 5 | Strong Importance | Experience and/or judgement strongly favour one activity over another. |
| 7 | Very Strong Importance | An activity is favoured very strongly over another. |
| 9 | Extreme Importance | The evidence favouring one activity over another is of the highest possible order of affirmation. |

Fig 6 Saaty's scale

As per Fig 6, we weigh our criteria in our decision matrix, followed by ranking priorities. We perform a pairwise comparison for each criterion and produce a normalized priority vector after finding the geomean values. We prioritize all the criteria separately for the data provided.

Since all the rankings are randomly based on assumptions, we need to perform a consistency check. It is done with the help of a consistency ratio. If the calculated CR is greater than 0.1, we cannot trust our assumptions, and the process is repeated until it is consistent. CR is a Consistent index divided by a random index.

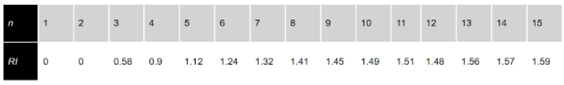
| | |
|---|---|
| CI | $CI = \frac{(\lambda \max - n)}{(n-1)}$ |
| RI | <br>n 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15<br>RI 0 0 0.58 0.9 1.12 1.24 1.32 1.41 1.45 1.49 1.51 1.48 1.56 1.57 1.59 |
| CR | CI / RI |

Fig 7 Consistency check

Once criteria weights are compared and proved consistent, we move on to construct a pairwise comparison matrix for all the five criteria and produce a normalized matrix. Finally, we calculate the final score for each criterion. The final score is the sum-product of each alternative's criteria to the criteria weights calculated initially.

### 2.2.2 AHP OUTPUT ANALYSIS

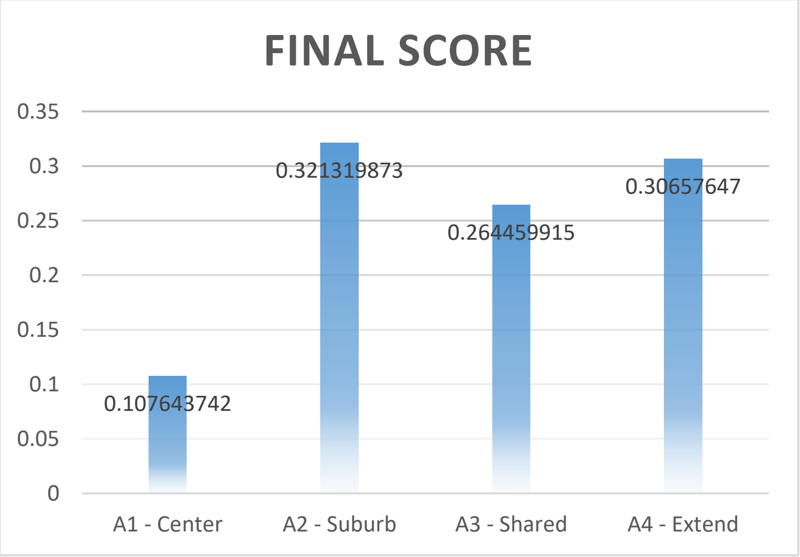


Fig 8 Final score analysis

Looking at the above Fig 8, We can clearly say A2-Suburb has the highest value amongst all. Taking cost as the highest preference followed by warehouse space, Transport, Parking, and security, we are here with clear-cut advice to give to the manufacturing company.

### 2.3.1 TOPSIS

TOPSIS method selects the best alternative whose distance is closest to ideal and farthest from negative ideal solution. Positive ideal solution has the best possible attributes whereas negative has the worst possible attributes. We will find positive and negative ideal solutions for all the alternatives with rest to each criteria provided. It is a step-by-step process as depicted below.

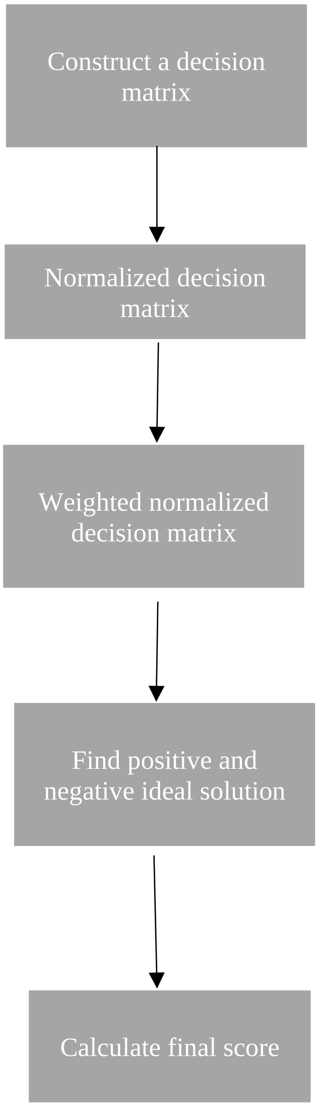

Construct a decision matrix
Normalized decision matrix
Weighted normalized decision matrix
Find positive and negative ideal solution
Calculate final score

Fig 9 TOPSIS flow

As Fig 9 suggests, we start by constructing the decision matrix. The weights for each criterion are decided based on a standard scale. We use the below scale for calculating weights for Transport.

| | |
|---|---|
| Excellent | **5** |
| Good | **4** |
| Moderate | **3** |
| Poor | **2** |
| Worse | **1** |

Fig 10 Scale

| Public transport links | Numerical Weightage | Average |
|---|---|---|
| **Good bus and rail links** | 4,4 | 4 |
| **Good bus links but no rail link** | 4,0 | 2 |
| **Poor bus links but good rail link** | 2,4 | 3 |
| **Excellent bus and rail links** | 5,5 | 5 |

Fig 11 Transport weights

Since, the above values are descriptive we have used Fig 11 for scaling the Transport criteria. This process suggests the alternative with excellent bus and rail link is the best followed by good bus and good rail links based on the average value calculated. For parking criteria, we use Fig 10 for assigning weights. Security and cost are assigned weights directly without any conversions since they are quantitative values. We create the decision matrix with the help of the weights assigned. Now, the next step is to normalize the decision matrix. We divide each weight with the root sum squared for each criterion to normalise the matrix. We generate a weighted normalized matrix where values are multiplied with its corresponding weights.

Now proceeding to the most important part in TOPSIS, finding positive and negative ideal solution. Firstly, we calculate the positive ideal solution with the below conditions.

| C1 - TRANSPORT (max) | C2 - PARKING (max) | C3 - WAREHOUSE SPACE (max) | C4 - SECURITY (max) | C5 - COST (min) |
|---|---|---|---|---|

The conditions mentioned above are decided based on the feasibility of getting a better solution. Our goal being expanding space, we need to have maximum values for three of the criteria with less cost. Accordingly, the criterions are selected for positive ideal solution and scores are calculated.

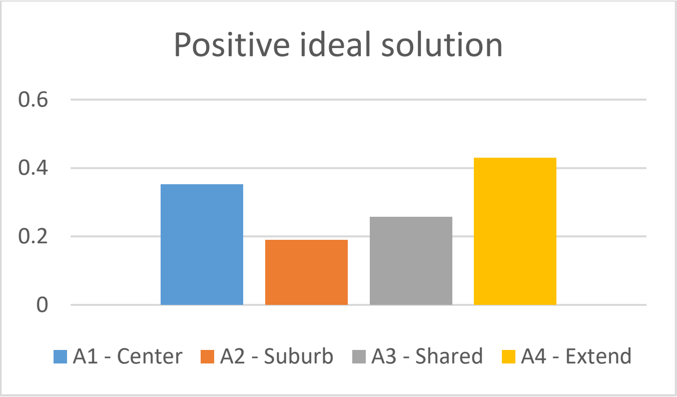


Fig 12 Positive solution

Similarly, we calculate the negative ideal solution by selecting those conditions with less possibility of achieving our goal.

| C1 - TRANSPORT (min) | C2 - PARKING (min) | C3 - WAREHOUSE SPACE (min) | C4 - SECURITY (min) | C5 - COST (max) |
|---|---|---|---|---|

The above condition will not help us in achieving our goal. So, this is by far the worst possible solution to acquire more space. Decision matrix and scores are calculated.

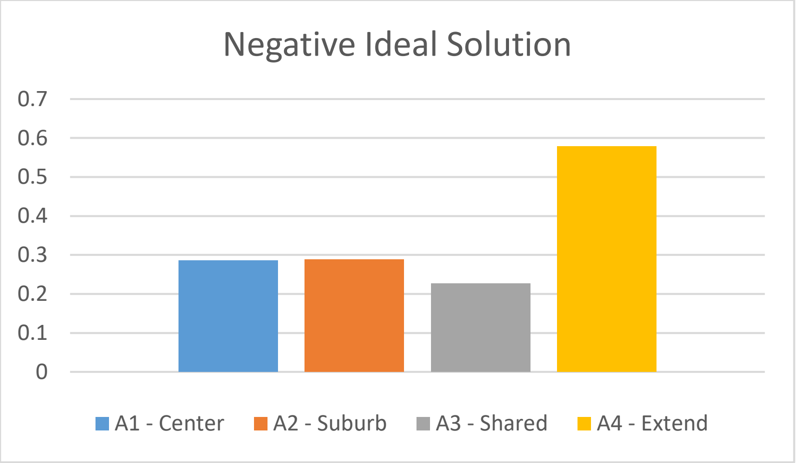


Fig 13 Negative solution

We produce the final scores using the above-calculated values. With the help of this, we can find the best alternative which is closest to ideal solution

### 2.3.2 TOPSIS ANALYSIS

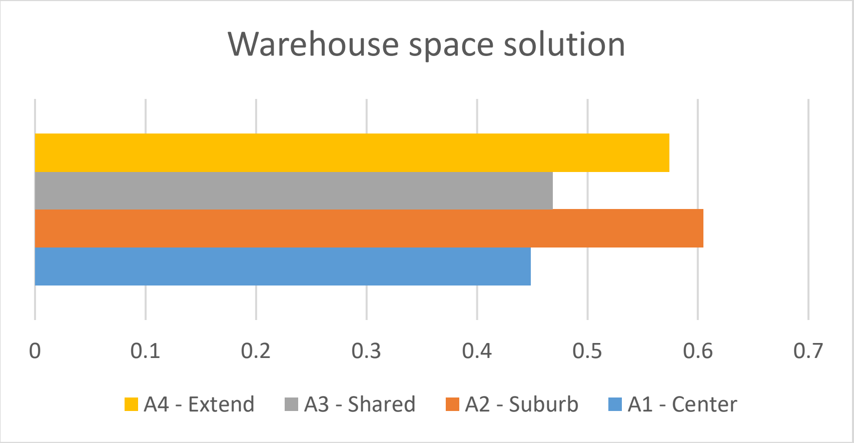


Fig 14 Output

As Fig 14 suggests, A2 is the best possible solution closest to the positive ideal solution and furthest from the negative ideal solution. So, we suggest the company purchase a space in the suburb as it is the best performing alternative. The next suggestion would be to go with Extending their current warehouse.

## 3. RECOMMENDATION AND CONCLUSION

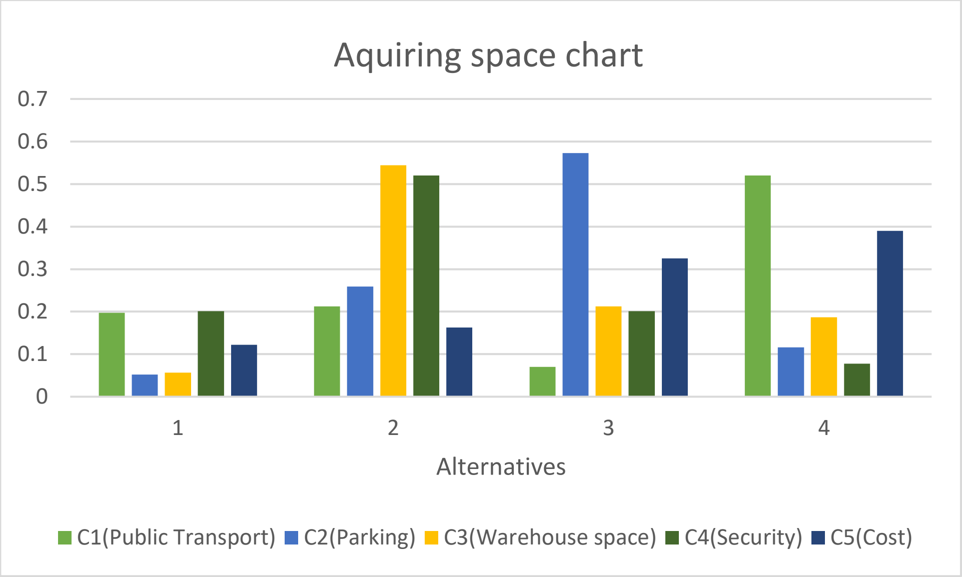


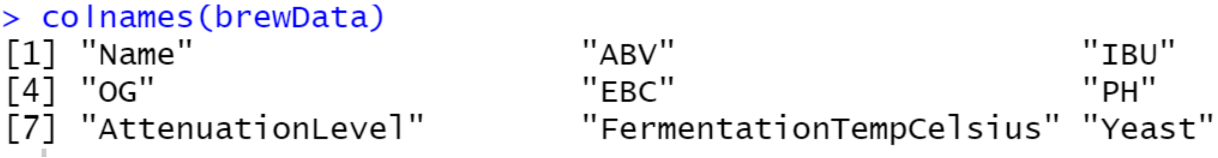
Fig 15 Analysis

By combining both results, we can conclude that purchasing a space in the suburb is the best option to acquire more space. Hence, we advise the manufacturing company to go with it. It is the only alternative providing maximum warehouse space and the highest security. It also has excellent transport and a parking facility. Overall, the cost is significantly less as well. The next possible suggestion would be to extend their current warehouse. The company is advised not to purchase a space in the city centre as it has the least possible score. It might also cost them a lot since the city centre is a prominent place, transportation costs would be huge, and getting a parking space would be difficult.

# PART – 2
# BEER MARKETING STRATEGY

# 1 INTRODUCTION

Brew Dog's data set is provided with several missing data points. The data is about 196 types of beers. The company wants to market similar beers to their customer as per their interest. So, we will help them identify all the missing values and cluster them accordingly using clustering algorithms. The following factors are provided to us to help us with the process.



```
> colnames(brewData)
[1] "Name"                       "ABV"                        "IBU"
[4] "OG"                         "EBC"                        "PH"
[7] "AttenuationLevel"           "FermentationTempCelsius"    "Yeast"
```

To cluster the data, we need to have zero missing values. So, we will proceed with missing data handling followed by clustering.

# 2 HANDLING MISSING DATA

It is essential to handle missing data prior to any analysis we do. It is because any statistical results based on this data would be biased. We will handle missing data with the help of R. This process is a step-by-step approach, as shown below.

| | |
|---|---|
| STEP 1 | Identify the missing data |
| STEP 2 | Identify the cause |
| STEP 3 | Decide DELETE / REPLACE |
| STEP 4 | Evaluate the data after STEP 3 |

## 2.1 IDENTIFYING MISSING DATA

The R plot mentioned below explains how much missing data are present in our given dataset.

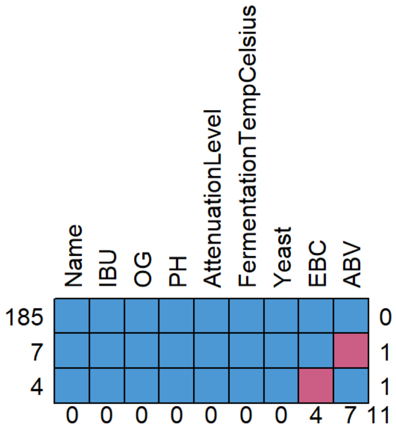


Figure 1 Missing data

Figure 1 clearly shows that 4 EBC and 7 ABV are missing. In contrast, all the other factors do not have any missing values. So, we can deduce 11 missing data in our dataset and 185 completed data.

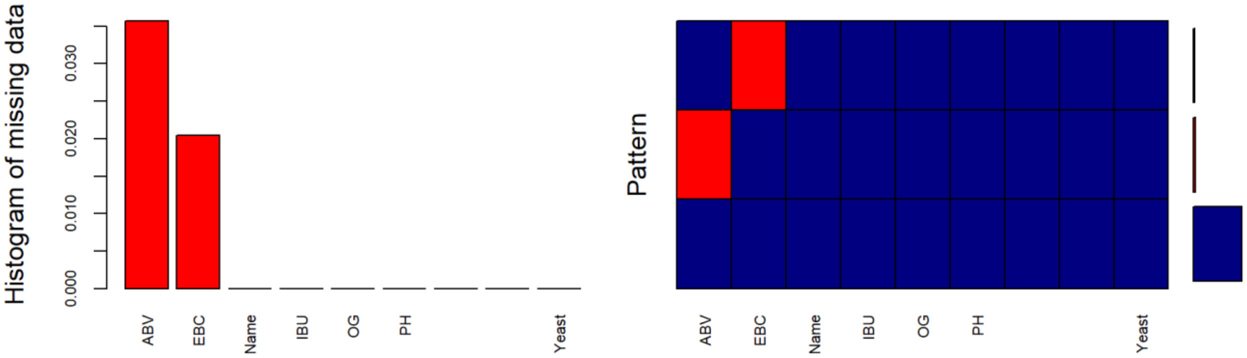


Figure 2 Histogram

The plot in Figure 2 states that almost 94% of the data is complete, with 3.6% missing ABV data and 2% missing EBC data.

## 2.2 IDENTIFYING THE CAUSE

Now that we have identified all the missing data, the next step is to find its cause. We are creating a new column with all the missing values. We use a correlation matrix to find the relation between all the columns given. In R, we will use the corrgram function to draw a correlation matrix.

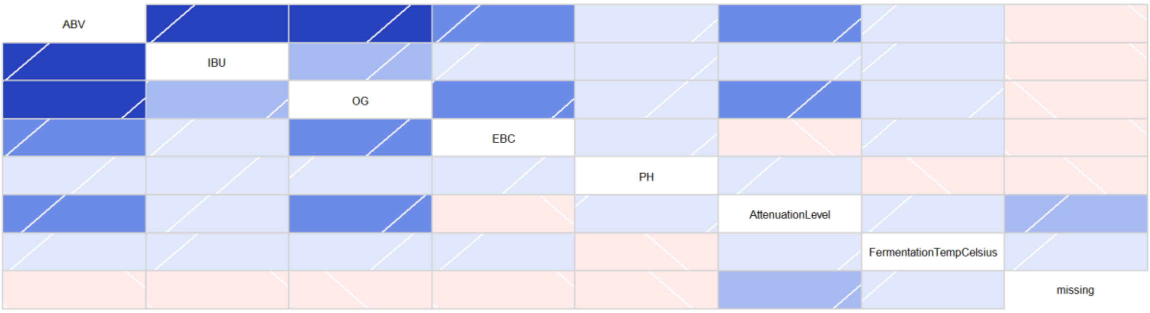


Figure 3 Correlation matrix

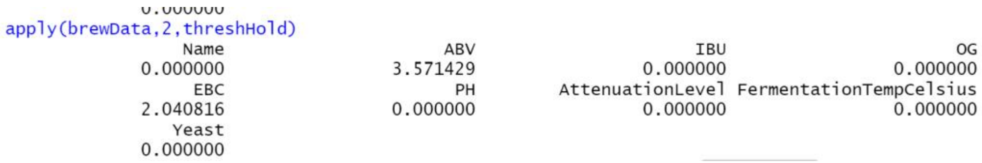
With the help of Fig 3, we can tell whether the values are co-related or not using shaded colours. The more intense blue is, the more linked they are. In comparison, red signifies less dependency amongst both. So, we can see that ABV and EBC correlate with other variables since it has blue shades. If there is less than a 5% missing value and it is missing at random, we can safely assume that the data is MCAR (2021).

## 2.3 DECIDE DELETE OR REPLACE

We decide to choose “replace” over “delete” since losing available data points might tweak the information and result (6 Different Ways to Compensate for Missing Data (Data Imputation with examples), 2021).

To replace, we are using the mice package in R. We chose the mice package over other methods because small quick solutions might bias our data. It can also handle diverse types of variables (R Packages | Impute Missing Values In R, 2021).

Before imputing, we check the data points with our safe threshold limit, which should not exceed 5 per cent. If it does, we are sampling out that data from imputation. The below result shows the missing threshold percentage.



```
                   0.000000
apply(brewData,2,threshHold)
                       Name                       ABV                       IBU                        OG
                   0.000000                  3.571429                  0.000000                  0.000000
                        EBC                        PH          AttenuationLevel FermentationTempCelsius
                   2.040816                  0.000000                  0.000000                  0.000000
                      Yeast
                   0.000000
```

3.5 % of the missing value is ABV, and the rest 2.04% is EBC. So, we proceed with mice imputation. We use predictive mean matching for multiple imputations since this will not lead to implausible values. It is because values are imputed based on other units' observation (Predictive Mean Matching Imputation (Example in R), 2021)

.

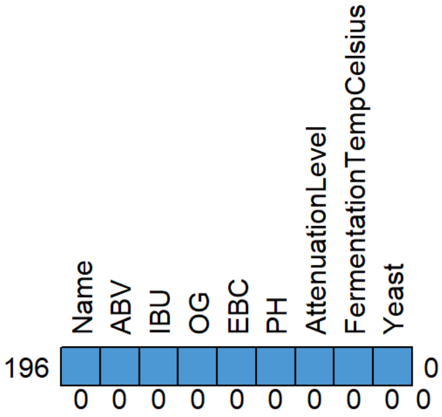


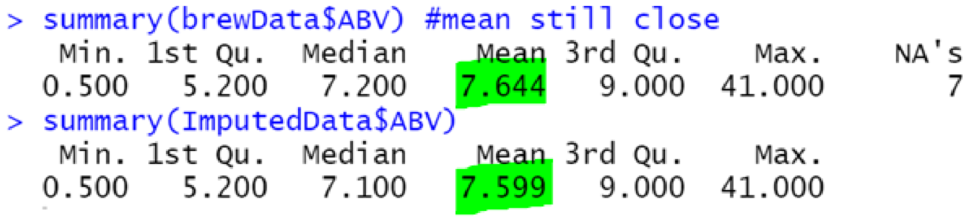
Figure 4 Imputed value

Figure 4 clearly shows no missing data left. So, we can confidently say all the data was imputed successfully. There are 196 complete data now for us to proceed with clustering.

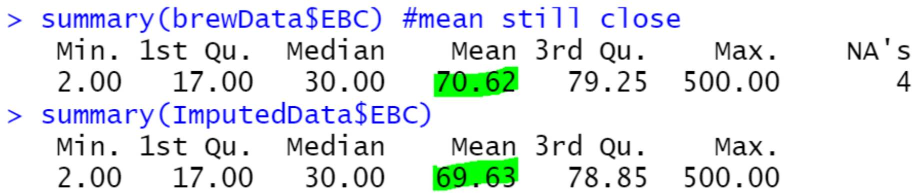
## 2.4 EVALUATING IMPUTED DATA

Now that data are imputed, we must ensure that the imputed data does not affect the data in any way. For this, we need to compare our imputed data with the old one and ensure there are no such deviations or impacts because of it. We use the below-mentioned methods to check.

### 2.4.1 COMPARING MEAN



```
> summary(brewData$ABV) #mean still close
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  0.500   5.200   7.200   7.644   9.000  41.000       7 
> summary(ImputedData$ABV)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.500   5.200   7.100   7.599   9.000  41.000 
```



```
> summary(brewData$EBC) #mean still close
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
   2.00   17.00   30.00   70.62   79.25  500.00       4 
> summary(ImputedData$EBC)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   2.00   17.00   30.00   69.63   78.85  500.00 
```

The above code snippet clearly states that there is not much difference in the mean before and after imputation for both the missing values.

### 2.4.2 COMPARING HISTOGRAM

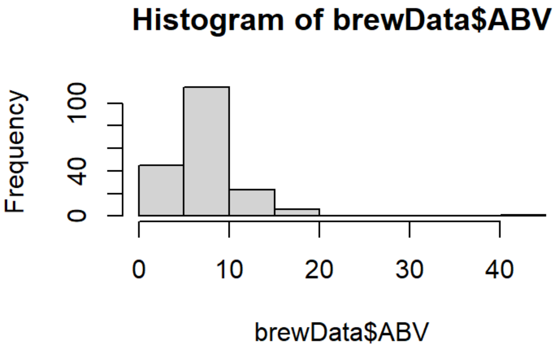


Figure 5 Initial ABV

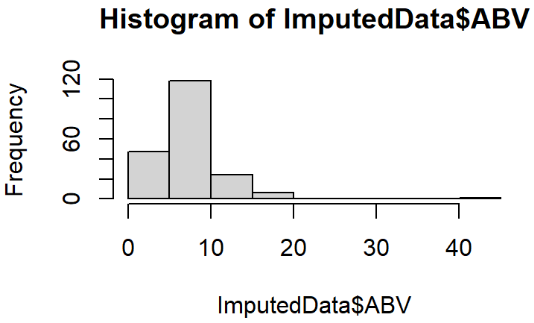


Figure 6 Imputed ABV

We cannot see any considerable difference between the two figures plotted above. We can conclude that the imputed ABV values did not impact anything. Similarly, we can compare for EBC as well.

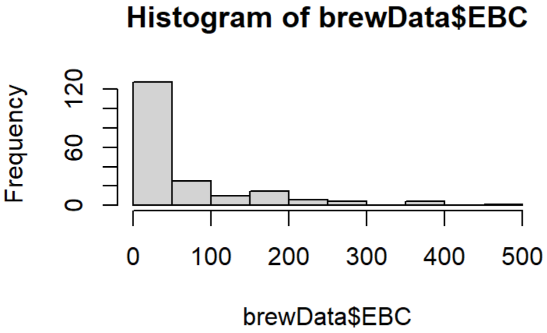


Figure 7 Initial EBC

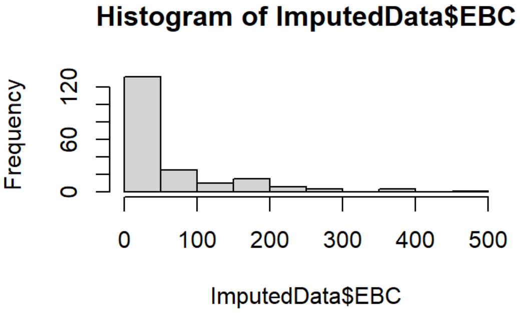
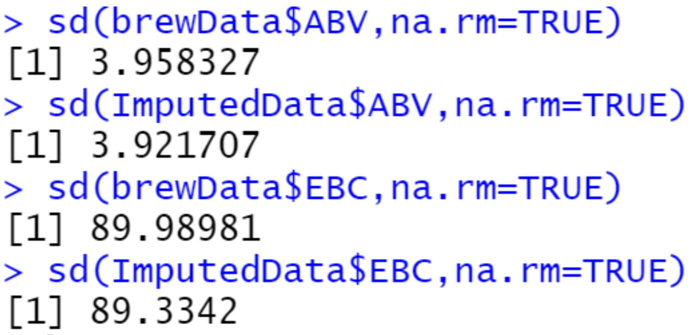


Figure 8 Imputed EBC

### 2.4.3 COMPARING STANDARD DEVIATION



```
> sd(brewData$ABV,na.rm=TRUE)
[1] 3.958327
> sd(ImputedData$ABV,na.rm=TRUE)
[1] 3.921707
> sd(brewData$EBC,na.rm=TRUE)
[1] 89.98981
> sd(ImputedData$EBC,na.rm=TRUE)
[1] 89.3342
```

The above code snippet depicts not much deviation even after imputing the values for both ABV and EBC.

## 2.5 IMPUTATION CONCLUSION

We have imputed all 11 missing values using the MICE method with the help of PMM. We also confirmed that imputed values did not affect the data frame much. It resulted in 196 completed data. We are good to proceed further with clustering.

# 3 CLUSTERING

The company wants to sell its customers similar beers according to their interests. To do this, we are helping them using a technique called clustering. Clustering is an identical grouping of data by calculating its similarity. It is of two types, Hierarchical and non-Hierarchical. For our case study, we have chosen Hierarchical over the other since it is more interpretable and provides more information. Also, the number of clusters can be easily determined using dendrograms.

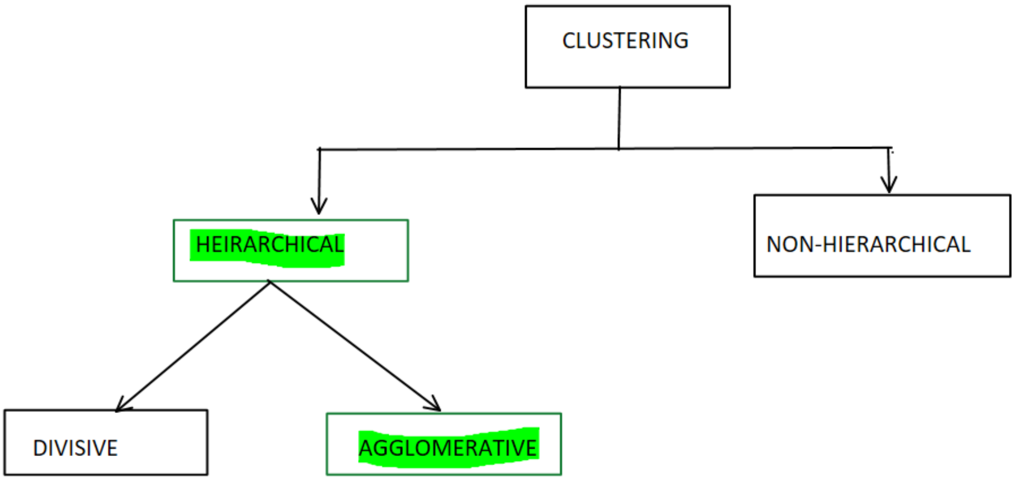


Figure 9 Clustering hierarchy

Hierarchical clustering can be formed from top-down or bottom-top using two techniques – Agglomerative and divisive. We are opting agglomerative over divisive since it is suitable for identifying small clusters (How to Perform Hierarchical Clustering using R | R-bloggers, 2021). Before we get started with clustering, we need to figure out the distance matrix which calculates distance between all the data points using various methods. For our case study, we chose Ward’s method over the others since it produces better cluster hierarchies. The other main reason is, it is less prone to outliers than the other (How to Perform Hierarchical Clustering using R | R-bloggers, 2021).

### 3.1 CLUSTERING PROCESS

We have categorical data in our dataset. So, we use the daisy method in R to compute the dissimilarity matrix. This method automatically uses Euclidean measures for numerical data and Gower’s distance for categorical data. Gower’s distance measures the dissimilarity of two factors. Now that we have our dissimilarity matrix, we can proceed with clustering with the help of the Agnes method. We produce the following dendrogram to understand in detail the clustering process.

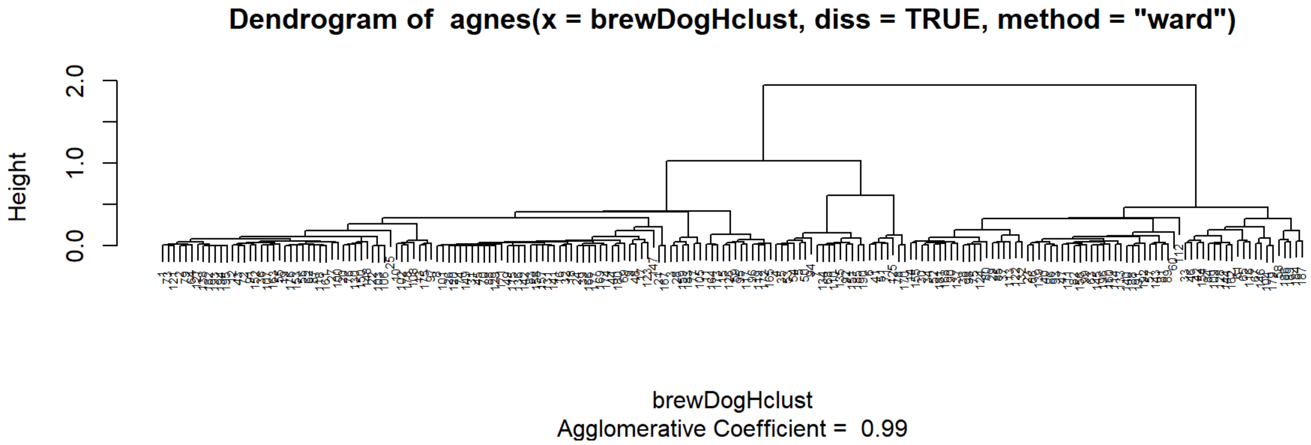
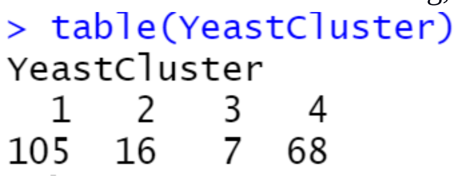


Figure 10 Dendrogram

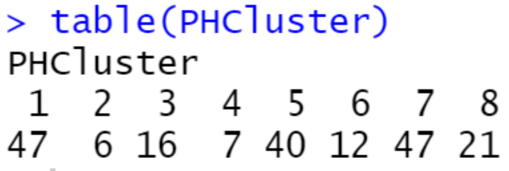
We can cluster the data using three main factors for marketing purposes.

- Yeast
- PH
- Fermentation Temperature

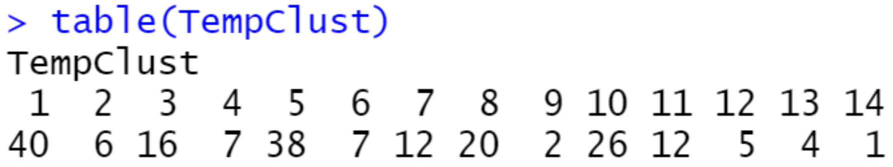
If we consider Yeast for clustering, we will need four clusters, as mentioned below.



```
> table(YeastCluster)
YeastCluster
  1   2   3   4
105  16   7  68
```



```
> table(PHCluster)
PHCluster
 1  2  3  4  5  6  7  8
47  6 16  7 40 12 47 21
```



```
> table(TempClust)
TempClust
 1  2  3  4  5  6  7  8  9 10 11 12 13 14
40  6 16  7 38  7 12 20  2 26 12  5  4  1
```

Let us take Yeast as an example and check the dendrogram with four as the cut-off point.

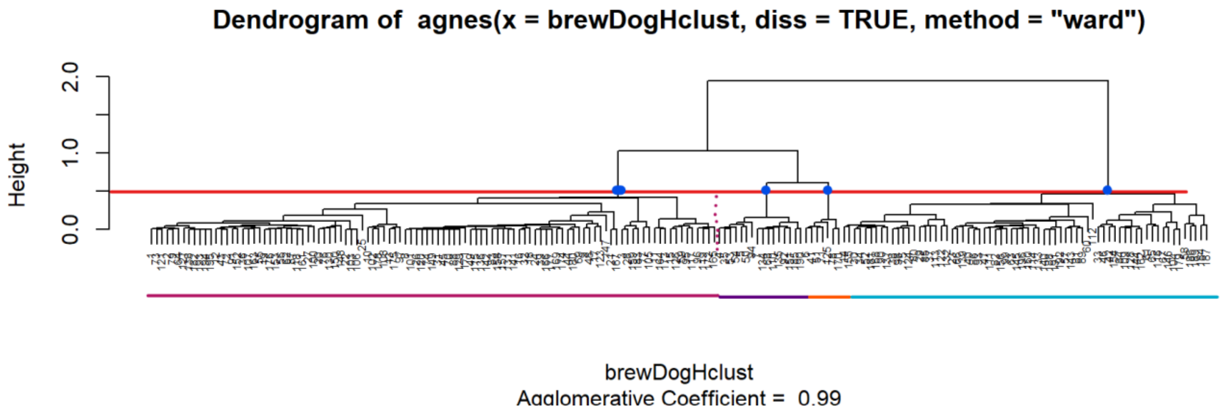


Figure 11 Four clusters

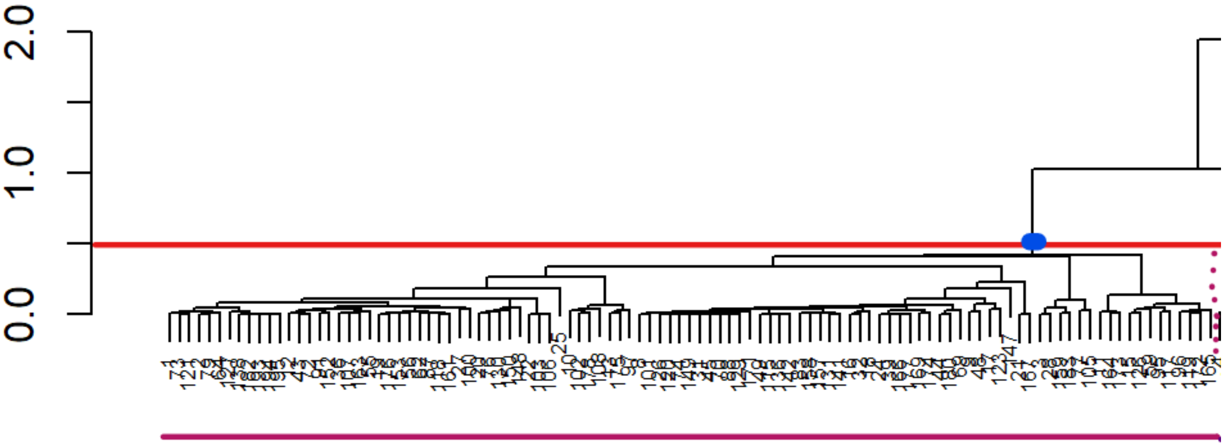


Figure 12 Cluster 1

Let us take four data points randomly from this cluster and check for the yeast value. Similarly, we check for all the available clusters.

| No | Name | Yeast |
|---|---|---|
| 10 | Arcade Nation | Wyeast 1056 - American Ale |
| 25 | Nanny State | Wyeast 1056 - American Ale |
| 123 | AB:03 | Wyeast 1056 - American Ale |
| 147 | Simcoe | Wyeast 1056 - American Ale |

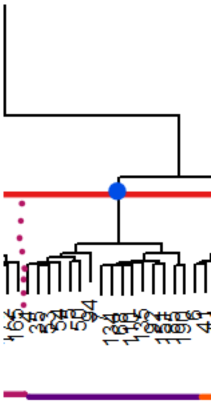


Figure 13 Cluster 2

| No | Name | Yeast |
|---|---|---|
| 94 | U-Boat (w/ Victory Brewing) | Wyeast 2007 - Pilsen Lager |
| 135 | This. Is. Lager | Wyeast 2007 - Pilsen Lager |
| 92 | Vagabond Pilsner | Wyeast 2007 - Pilsen Lager |
| 190 | Prototype Helles | Wyeast 2007 - Pilsen Lager |

Figure 14 Cluster 3

| No | Name | Yeast |
|---|---|---|
| 125 | Rhubarb Saison - B-Sides | Wyeast 3711 - French Saison |
| 78 | Everday Anarchy | Wyeast 3711 - French Saison |
| 170 | Black Jacques | Wyeast 3711 - French Saison |
| 41 | TM10 | Wyeast 3711 - French Saison |

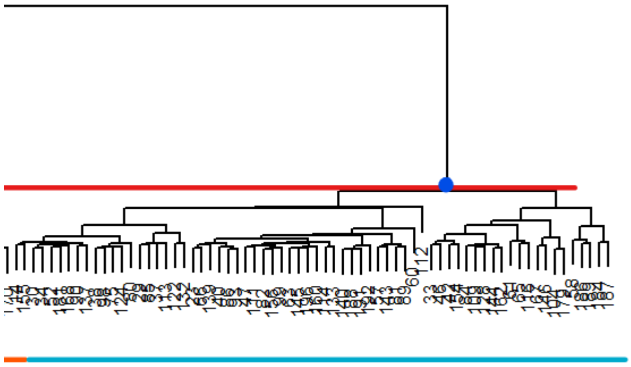


Figure 15 Cluster 4

| No | Name | Yeast |
|---|---|---|
| 33 | AB:17 | Wyeast 1272 - American Ale II |
| 124 | AB:13 | Wyeast 1272 - American Ale II |
| 112 | Sink The Bismarck! | Wyeast 1272 - American Ale II |
| 60 | Whisky Sour - B-Sides | Wyeast 1272 - American Ale II |

We suggest the manufacturing company cluster based on Yeast since it is a vital factor for categorizing beer types. Customer's preference also changes concerning the Yeast content. Some prefer Ale while others might prefer lager. So, this might help in similar marketing beers to customers based on their taste preference. Next suggestion would be to consider PH and temperature.

# 4 REFERENCES

- R-bloggers. 2021. *How to Perform Hierarchical Clustering using R | R-bloggers*. [online] Available at: <https://www.r-bloggers.com/2017/12/how-to-perform-hierarchical-clustering-using-r/> [Accessed 16 December 2021].

- 2021. [online] Available at: <https://medium.com/@yogesh_khurana/how-to-handle-missing-values-5dc70e805720.> [Accessed 16 December 2021].

- Medium. 2021. *6 Different Ways to Compensate for Missing Data (Data Imputation with examples)*. [online] Available at: <https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779> [Accessed 16 December 2021].

- Analytics Vidhya. 2021. *R Packages | Impute Missing Values in R*. [online] Available at: <https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/> [Accessed 16 December 2021].

- Statistics Globe. 2021. *Predictive Mean Matching Imputation (Example in R)*. [online] Available at: <https://statisticsglobe.com/predictive-mean-matching-imputation-method/> [Accessed 16 December 2021].

# 5 APPENDIX A

## 5.1 AHP ANALYSIS CRITERIA WEIGHTS IMPORTANCE SCALE

Weights:

| Criteria ranking | |
|---|---|
| C5(Cost) | 1 |
| C3(Warehouse space) | 2 |
| C1(Public Transport) | 3 |
| C2(Parking) | 4 |
| C4(Security) | 5 |

Transport:

| Criteria ranking | |
|---|---|
| A4 - Extend | 1 |
| A2 - Suburb | 2 |
| A1 - Centre | 3 |
| A3 - Shared | 4 |

Parking:

| Criteria ranking | |
|---|---|
| A3 - Excellent | 1 |
| A2 - Good | 2 |
| A4 - Moderate | 3 |
| A1 - Poor | 4 |

Warehouse space:

| Criteria ranking | |
|---|---|
| A2 - Excellent | 1 |
| A3 - Good | 2 |
| A4 - Good | 2 |
| A1 - Poor | 3 |

Security:

| Criteria ranking | |
|---|---|
| A2 - **** | 1 |
| A3 - *** | 2 |
| A1 - *** | 2 |
| A4 - ** | 3 |

## 5.2 TOPSIS EXTERNAL SCALES USED

Weights:

| ……. | C1 - TRANSPORT | C2 - PARKING | C3 - WAREHOUSE SPACE | C4 - SECURITY | C5 - COST |
|---|---|---|---|---|---|
| Weight | 0.15 | 0.1 | 0.3 | 0.05 | 0.4 |

As per the above table, Highest weight is given to cost since it plays a vital role in helping us acquire more space and in maintaining funds. The priority will be for warehouse space since that is our goal. This is followed by transport, parking and security which are the other major factor for any manufacturing company.

Scale:

| Scale | Excellent | Good | Moderate | Poor | Worse |
|---|---|---|---|---|---|
| Weightage | 5 | 4 | 3 | 2 | 1 |

Transport**:**

| Public transport links | Numerical Weightage | Average/Ratings |
|---|---|---|
| Good bus and rail links | 4,4 | 4 |
| Good bus links but no rail link | 4,0 | 2 |
| Poor bus links but good rail link | 2,4 | 3 |
| Excellent bus and rail links | 5,5 | 5 |

Parking and warehouse space:

| Parking/Warehouse | Weightage |
|---|---|
| Poor | 2 |
| Good | 4 |
| Excellent | 5 |
| Moderate | 3 |

## 5.3 IMPUTATION AND CLUSTERING

The R code used for performing imputation and clustering has been attached below with proper comments for every line of code.

BADS_Imputation_Clustering.R