Leeds University Business School

UNIVERSITY OF LEEDS

Assessed Coursework Coversheet

For use with *individual* assessed work

Student ID Number:	2	0	1	5	7	6	9	5	6
Module Code:	LUBS5308M								
Module Title:	Business Analytics and Decision Science								
Module Leader:	Richard Hodgett								
Declared Word Count:	3000								

Please Note:

Your declared word count must be accurate, and should not mislead. Making a fraudulent statement concerning the work submitted for assessment could be considered academic malpractice and investigated as such. If the amount of work submitted is higher than that specified by the word limit or that declared on your word count, this may be reflected in the mark awarded and noted through individual feedback given to you.

It is not acceptable to present matters of substance, which should be included in the main body of the text, in the appendices ("appendix abuse"). It is not acceptable to attempt to hide words in graphs and diagrams; only text which is strictly necessary should be included in graphs and diagrams.

By submitting an assignment you confirm you have read and understood the University of Leeds **Declaration of Academic Integrity** (http://www.leeds.ac.uk/secretariat/documents/academic_integrity.pdf).

LUBS5308M

Business Analytics and Decision Science Coursework

Part – I

Report on Expansion Advice

By – 201576956

Introduction:

When you grow as a company, there is always a need for expansion. So, an approach for advice is made by a small manufacturing company on how they should expand, and which is the best way to acquire more space based on certain alternatives and criteria. So, on applying two different MCDA methods, AHP and TOPSIS, on the data provided by the company, one of the alternatives showed results in favor of the companies' interest, and now it is up to them to comply with it. This report contains information about these findings and the way these were achieved.

Data Characterization:

The company provided data in the form of a table that outlined various options they considered, such as acquiring land in the city center or the suburbs or extending the existing warehouse, taking various criteria into account. For the two analyses mentioned above, respective tables are created based on the data. Figure 1 depicts the original data, while figures 2 and 3 display the data tables for AHP and TOPSIS analyses.

Figure 2 contains the normalized values from the data in figure 1 for each criterion and the normalized weights for the same. As, Figure 1 shows that different criteria have different ranges of values. As a result, comparing these values straight up may not yield the desired result. Accordingly, the values of various criteria are normalized to make them comparable. The process is like that in figure 3.

		Alternatives			
		A1 (Centre)	A2 (Suburb)	A3 (Shared)	A4 (Extend)
Criteria	C1 (Public transport links)	Good bus and rail links	Good bus links but no rail link	Poor bus links but good rail link	Excellent bus and rail links
	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

Figure 1: Data provided by the company

	PUBLIC TRANSPORT LINKS	PARKING	WAREHOUSE SPACE	SECURITY	COST
A1(CENTRE)	0.2928	0.0954	0.0886	0.227	0.4103
A2(SUBURB	0.0715	0.2772	0.4337	0.4236	0.3077
A3(SHARED	0.1137	0.4673	0.2389	0.227	0.1538
A4(EXTEND	0.5219	0.1601	0.2389	0.1223	0.1282

	WEIGHTS
PUBLIC TRANSPORT LINKS	0.4185
PARKING	0.0618
WAREHOUSE SPACE	0.2625
SECURITY	0.0973
COST	0.1599

Figure 2: Data table made for AHP analysis

Column1	PUBLIC TRANSPORT LINKS	PARKING	WAREHOUSE SPACE	SECURITY	COST
WEIGHT	0.4185	0.0618	0.2625	0.0973	0.1599
A1(CENTRE)	6	2	2	6	800,000
A2(SUBURB)	4	6	8	8	600,000
A3(SHARED)	5	8	6	6	300,000
A4(EXTENT)	8	4	6	4	250,000

Figure 3: Data table made for TOPSIS analysis

Data Analysis and Findings:

The following section discusses the approach to performing AHP and TOPSIS on the data provided by the company. The findings of this approach will also be shared, but first, let's see what kind of MCDA approach AHP and TOPSIS are:

- AHP: In order to analyze and solve complex problems, Analytic Hierarchy Process, or AHP, is used. The pairwise comparison method is an approach to multi-criteria decision making (MCDA) where one compares each alternative with itself and the others to determine which is better and by how much. AHP can be computed using two methods: eigenvectors and geometric means.

- TOPSIS: TOPSIS stands for Technique for Order Preference by Similarity to Ideal Solution. It is an MCDA method based on the notion that the best alternative should have the shortest distance from the positive ideal solution and the longest distance from the negative ideal solution. The person conducting this analysis chooses the positive and negative ideal solutions. As in the AHP method, weights are involved in this technique too. These normalized weights, however, are derived from the AHP method.

After defining some of the methods above, let’s discuss the various stages involved in each of these MCDA methods. Let’s start with AHP first:

- In AHP, the first step is to assign weights to the different criteria under study. These weights may differ from person to person as it depends on which criteria are most important to the person doing the analysis. Check figure 18 for the various weights of each criterion with respect to each other.

	PUBLIC TRANSPORT LINKS	PARKING	WAREHOUSE SPACE	SECURITY	COST
PUBLIC TRANSPORT LINKS	1	5	2	4	3
PARKING	0.2	1	0.25	0.5	0.333333333
WAREHOUSE SPACE	0.5	4	1	3	2
SECURITY	0.25	2	0.333333333	1	0.5
COST	0.333333333	3	0.5	2	1

Figure 18: Depicts the weights of each criterion w.r.t each other

- In the next step, a similarity matrix is drawn, which is basically a matrix where each criterion is compared to the other criteria and itself based on their weights. The similarity matrix for weights of the data used above can be seen in figure 4.

	PUBLIC TRANSPORT LINKS	PARKING	WAREHOUSE SPACE	SECURITY	COST
PUBLIC TRANSPORT LINKS	1	5	2	4	3
PARKING	0.2	1	0.25	0.5	0.333333333
WAREHOUSE SPACE	0.5	4	1	3	2
SECURITY	0.25	2	0.333333333	1	0.5
COST	0.333333333	3	0.5	2	1

Figure 4: Similarity matrix for weights in AHP

- After the similarity matrix is ready, it is squared, and the values are stored in another table-like structure. Once the required table (which is the square of the similarity matrix) is formulated, a summation of each row is calculated, and the values are stored in another column, named Sum. The values in the "Sum" column are then normalized, and the normalized values are stored in a column named "Normalized Sum". Refer to Figure 5 to visualize the scenario.

	PUBLIC TRANSPORT LINKS	PARKING	WAREHOUSE SPACE	SECURITY	COST		SUM		NORMALIZED SUM
PUBLIC TRANSPORT LINKS	1	5	2	4	3				
PARKING	0.2	1	0.25	0.5	0.333333333				
WAREHOUSE SPACE	0.5	4	1	3	2				
SECURITY	0.25	2	0.333333333	1	0.5				
COST	0.333333333	3	0.5	2	1				
PUBLIC TRANSPORT LINKS	5	35	8.083333333	22.5	13.66666667		84.2500		0.4193
PARKING	0.761111111	5	1.233333333	3.216666667	2.016666667		12.2278		0.0609
WAREHOUSE SPACE	3.216666667	22.5	5	14	8.333333333		53.0500		0.2640
SECURITY	1.233333333	8.083333333	1.916666667	5	3.083333333		19.3167		0.0961
COST	2.016666667	13.66666667	3.083333333	8.333333333	5		32.1000		0.1597
							200.9444		

Figure 5: Shows the Sum and Normalized Sum columns along with the square of our similarity matrix.

- Repeat the above steps until the same values for the “Normalized Sum” column for two consecutive steps are derived. These values can then be considered as normalized values for weights.
- Follow the same pattern of steps as above for each criterion in the data. Calculate the similarity matrix for each criterion while comparing the different alternatives to each other and themselves for that criterion. The normalized sum for each criterion has to be calculated and stored. Refer figure 6 to see how we calculate normalized sum for a particular criterion.

	A1(CENTRE)	A2(SUBURB)	A3(SHARED)	A4(EXTEND)		SUM		NORMALIZED SUM
A1(CENTRE)	1	4	3	0.5				
A2(SUBURB	0.25	1	0.5	0.16666667				
A3(SHARED	0.333333333	2	1	0.2				
A4(EXTEND)	2	6	5	1				
A1(CENTRE)	4	17	10.5	2.26666667		33.7667		0.2934
A2(SUBURB	1	4	2.583333333	0.55833333		8.1417		0.0708
A3(SHARED	1.566666667	6.533333333	4	0.9		13.0000		0.1130
A4(EXTEND)	7.166666667	30	19	4		60.1667		0.5228
						115.0750		

Figure 6: Shows how the matrix looks for a particular criterion when performing AHP

- Calculate the normalized sum of each criterion and form a matrix where the rows are the different available alternatives, and the columns are the different criteria. Input the normalized values calculated for each criterion in the above steps into the new matrix. The weights of each criterion that were calculated at first are then multiplied to individual criteria in each row and then the sum of rows is calculated. This gives a score for each alternative, and the alternative with the highest score is chosen to be the best alternative. The process we used to calculate these values is called the Eigenvector Method.

Completing these steps will show that for the data under discussion AHP suggests that extending the current warehouse would be the best alternative for the company. Refer to figure 7 to visualize the results.

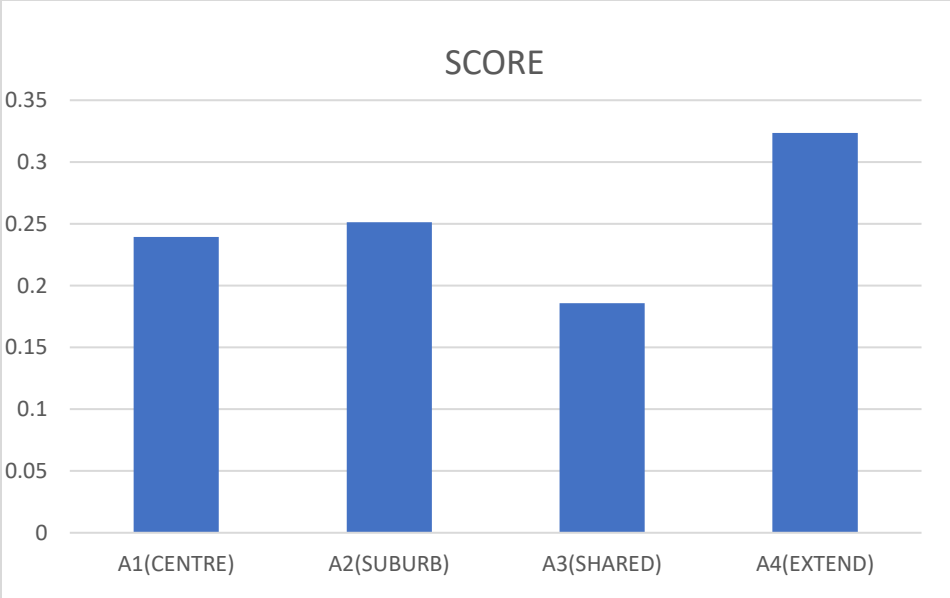

Figure 7: Results after performing AHP showing extend as the best alternative

Now, let's see what results TOPSIS offers and what can be inferred from those results. The various steps involved are:

- The TOPSIS matrix can be seen in figure 8 below where, the rows consist of all the alternatives presented in the given data along with an extra row called weights that consists of the same normalized weights from the AHP. The columns consist of all the criterions mentioned in the original data in figure 1.

Column1	PUBLIC TRANSPORT LINKS	PARKING	WAREHOUSE SPACE	SECURITY	COST
WEIGHT	0.4185	0.0618	0.2625	0.0973	0.1599
A1(CENTRE)	6	2	2	6	800,000
A2(SUBURB)	4	6	8	8	600,000
A3(SHARED)	5	8	6	6	300,000
A4(EXTENT)	8	4	6	4	250,000

Figure 8: The TOPSIS matrix

- While forming this matrix, certain assumptions were made:
    1. Values 2, 4, 6, 8, in the dataset mark the importance of a criterion for different alternatives in the data ranging from poor to excellent.
    2. For the public transport links column in the matrix, an average of values from figure 9 were taken to formulate respective values for the complex decision making.

	PUBLIC TRANSPORT LINKS	
TYPES	BUS	RAIL
NO	2	2
POOR	4	4
GREAT	6	6
EXCELLENT	8	8

Figure 9: Value range for different types of public transport links ranging from no links to excellent links

- The first step is to normalize the decision matrix using the vector normalization formula in figure 10.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_i x_{ij}^2}}$$

.

Figure 10: vector normalization formula where Xij is score of alternative i w.r.t j (courtesy: Wikipedia)

- Performing the above computation and then multiplying the normalized values with the criteria weights in our data we get the following normalized matrix which can be seen in figure 11.

Column1	PUBLIC TRANSPORT LINKS	PARKING	WAREHOUSE SPACE	SECURITY	COST
A1(CENTRE)	0.504977623	0.1825713	0.168989269	0.48664911	0.7451943
A2(SUBURB)	0.336651748	0.5477138	0.675957075	0.64886548	0.5588957
A3(SHARED)	0.420814686	0.7302851	0.506967807	0.48664911	0.2794479
A4(EXTENT)	0.673303497	0.3651426	0.506967807	0.32443274	0.2328732

Figure 11: Decision matrix after the above computation with the normalized values of each criterion

- Using the weighted normalized matrix above, the positive ideal solution or PIS (marked in green) and negative ideal solution or NIS in figure 12 (marked in red) can be determined for each criterion. In four of the criteria, higher values are better, while for cost (which is minimizing), lower is better.

Column1	PUBLIC TRANSPORT LINKS	PARKING	WAREHOUSE SPACE	SECURITY	COST
A1(CENTRE)	0.211333135	0.0112829	0.044359683	0.04735096	0.1191566
A2(SUBURB)	0.140888757	0.0338487	0.177438732	0.06313461	0.0893674
A3(SHARED)	0.176110946	0.0451316	0.133079049	0.04735096	0.0446837
A4(EXTENT)	0.281777513	0.0225658	0.133079049	0.03156731	0.0372364

Figure12: Normalized weighted matrix with Positive and Negative ideal points marked

- Compute the distance of each alternative from the PIS using the formula shown in figure 13 and the distance from the NIS using the formula shown in figure 14. Finally compute the final score for each alternative using the formula shown in figure 15. The different values and the final score can be seen in figure 16.

$$S_i^* = \sqrt{\sum \left(V_{ij} - V_j^*\right)^2}$$

Figure 13: Formula to calculate PIS

$$S_i^- = \sqrt{\sum \left(V_{ij} - V_j^-\right)^2}$$

Figure 14: Formula to calculate NIS

$$Final\ Score = S_i^- / \left(S_i^* + S_i^-\right)$$

Figure 15: Formula to calculate the final score (Image Courtesy: BADS class notes)

						Si*	
PIS	0.00496241	0.0011457	0.017710033	0.00024912	0.0067109	0.1754372	A1(CENTRE)
	0.019849642	0.0001273	0	0	0.0027176	0.1506472	A2(SUBURB)
	0.011165423	0	0.001967781	0.00024912	5.546E-05	0.1159215	A3(SHARED)
	0	0.0005092	0.001967781	0.00099649	0	0.0589363	A4(EXTENT)
						Si-	
NIS	0.00496241	0	0	0.00024912	0	0.072191	A1(CENTRE)
	0	0.0005092	0.017710033	0.00099649	0.0008874	0.1417855	A2(SUBURB)
	0.001240603	0.0011457	0.007871126	0.00024912	0.0055462	0.1266996	A3(SHARED)
	0.019849642	0.0001273	0.007871126	0	0.0067109	0.1859005	A4(EXTENT)

Figure 16: The final scores of PIS and NIS

- The result of the above computation can be seen in figure 17. From these scores, TOPSIS approach suggests that the company should extend its current warehouse like AHP approach.

A1(CENTRE)	0.291529672	WORST
A2(SUBURB)	0.484848321	
A3(SHARED)	0.522211867	
A4(EXTENT)	0.759283166	BEST

Figure 17: Results of TOPSIS Approach

Advice to the company:

The results from both AHP and TOPSIS show that the best advice for the company is to extend their current warehouse. Figure 1 supports this claim as well as the current warehouse already has excellent bus and rail links, with good space and moderate parking. Extending the current warehouse would help the company cut their costs significantly and thus expand their business profitably. The company can consider making its security arrangements better for the present warehouse.

The step by step working of both the AHP and TOPSIS approaches have been discussed in the Data Analysis and Findings section of this report in case someone wants to refer to something or has an objection to the results or advice given above.

LUBS5308M

Business Analytics and Decision Science Coursework

Part – II

Beer Clustering for BrewDog

By – 201576956

Introduction:

A company needs to know their customers well, what they like and dislike. BrewDog, a beer company, seeks to cluster its beers so that they can use this information to market similar beers to their customers. The report describes how the clustering technique is used with the company's data to show what conclusions can be drawn based on the clustered results. This report also discusses how the missing datasets in the data were handled.

Data Characterization:

The data provided by BrewDog is a CSV file with 9 columns and 196 rows of different kinds of beers. The columns are:

- Name: Contains the name of different beers.
- ABV: Alcohol by volume figure for the beer.
- IBU: International Bitterness Units for the respective beer.
- OG: Original Gravity for a beer.
- EBC: Color units from the European Brewery Convention.
- PH: Acid & Base scale.
- Attenuation Level
- FermentationTempCelcius: Fermentation temperature in Celsius.
- Yeast Type

The data has 185 rows with non-null values and 11 rows with some null values in one of the columns. Upon analysis of the dataset more, than 7 null values can be seen in the ABV column, whereas 4 null values are present in the EBC column. Figure 1 helps to visualize the number of missing values in the data. Figure 2 shows how the data is.

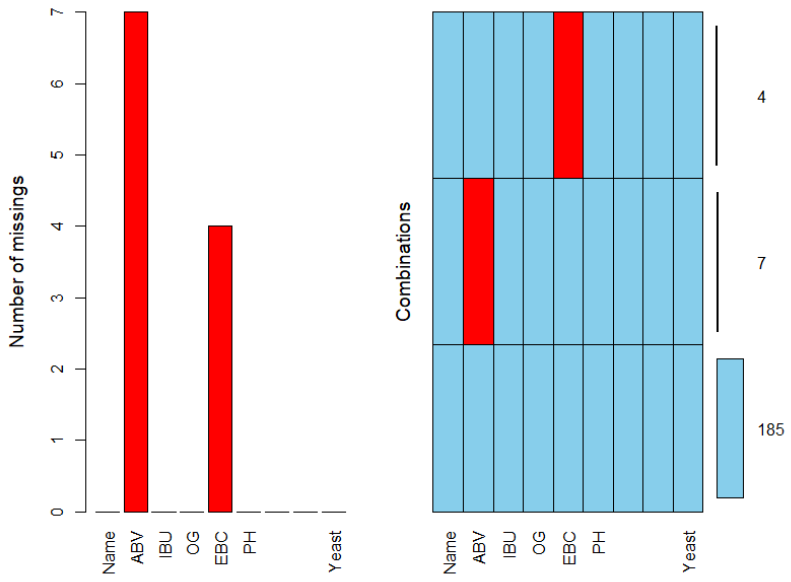

Figure 1: Depicts the number of missing data in each column

	Name	ABV	IBU	OG	EBC	PH	AttenuationLevel	FermentationTempCelsius	Yeast
1	Buzz	4.50	60.0	1044.0	20.0	4.4	75.00	19	Wyeast 1056 - American Ale
2	Trashy Blonde	4.10	41.5	1041.7	15.0	4.4	76.00	18	Wyeast 1056 - American Ale
3	Berliner Weisse With Yuzu - B-Sides	4.20	8.0	1040.0	8.0	3.2	83.00	21	Wyeast 1056 - American Ale
4	Pilsen Lager	6.30	55.0	1060.0	30.0	4.4	80.00	9	Wyeast 2007 - Pilsen Lager
5	Avery Brown Dredge	7.20	59.0	1069.0	10.0	4.4	67.00	10	Wyeast 2007 - Pilsen Lager
6	Electric India	7.50	38.0	1045.0	15.0	4.4	88.90	22	Wyeast 3711 - French Saison
7	Fake Lager	4.70	40.0	1046.0	12.0	4.4	78.00	10	Wyeast 2007 - Pilsen Lager
8	Bramling X	7.50	75.0	1068.0	22.0	4.4	80.90	19	Wyeast 1056 - American Ale
9	Misspent Youth	7.30	30.0	1079.0	120.0	4.4	74.70	19	Wyeast 1056 - American Ale
10	Arcade Nation	5.30	60.0	1052.0	200.0	4.2	77.00	19	Wyeast 1056 - American Ale
11	Movember	4.50	50.0	1047.0	140.0	5.2	74.50	19	Wyeast 1056 - American Ale
12	Alpha Dog	4.50	42.0	1046.0	62.0	4.4	72.80	22	Wyeast 1056 - American Ale

Showing 1 to 12 of 196 entries, 9 total columns

Figure 2: BrewDog CSV file data

Data Wrangling, Imputation:

In the data characterization section of this report, detailed information about the missing values in the data is presented. Moving ahead with clustering without doing anything about the missing datasets might be a bad idea. Also, removing the rows containing the missing values might seem to make our lives easy but, since it's a small dataset, doing that might not be a good option. There are usually two ways to deal with missing datasets apart from the one discussed above: Simple Imputation and Multiple Imputation.

Let's see how these imputation techniques work over a dataset:

- Simple Imputation: In this method, replacing the missing value with the mean, median, or mode values of the column can be done.
- Multiple Imputation: This method involves performing a simple imputation for every missing value and then setting back a variable to its null state when the imputation is complete. After this, Regression is performed on the column to forecast the value of the missing variable. This step is repeated for every single missing value in the column. This process can be done easily in R programming using the **mice()** function.

Now, lets start with the imputation process over the dataset:

- The best way to make the choice of which imputation method to go with is to check out the distribution of the columns the imputation process must be done on and check if there are any inconsistencies in the data column. Figure 3 below displays the distribution of ABV and EBC columns from the given data using histograms.

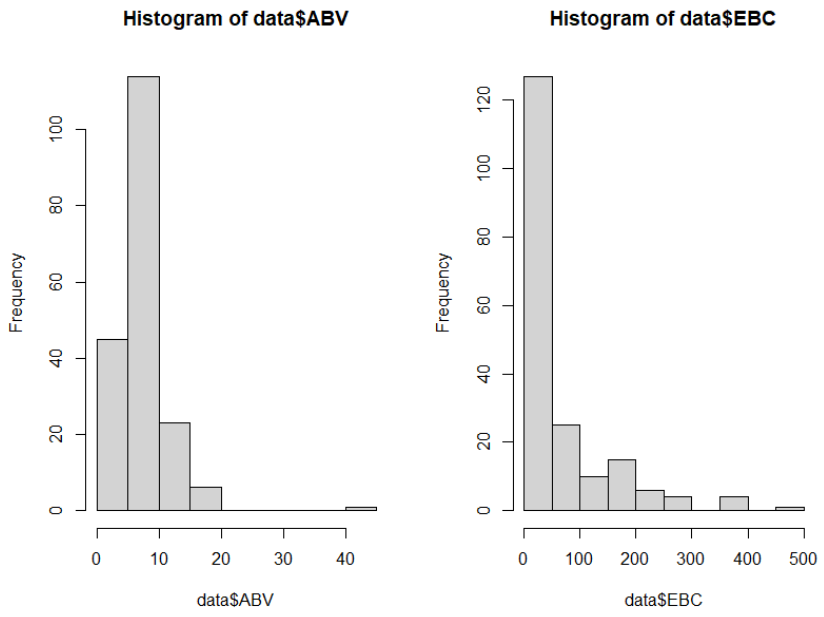

Figure 3: Histograms for ABV column and EBC column of the dataset, showing the distribution of data for the respective columns along with the inconsistencies in the value range.

- Looking at this histogram, it seems like using Multiple Imputations is a better choice as figure 3 shows the presence of an outlier value in the ABV column and a non-normal distribution for the EBC column. Using simple imputations might produce odd results since the mean, median, and mode values can differ largely from the actual values in that column, thus filling up NA values with unrealistic numbers.

- As a result of applying multiple imputations on the two columns (ABV and EBC), there are no more null values remaining. Following imputations, the null values in the ABV column display a value of 7.50, and the null values in the EBC column display a value of 40.0. These changes are seen when the summary of the two columns is compared before and after computation (Figure 4).

```
> summary(data$EBC)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
   2.00   17.00   30.00   70.62   79.25  500.00       4
>
> summary(mi$EBC)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2.00   17.75   30.00   69.99   78.85  500.00
> summary(data$ABV)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
  0.500   5.200   7.200   7.644   9.000  41.000       7
>
> summary(mi$ABV)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.500   5.200   7.200   7.638   8.575  41.000
>
```

Figure 4: Shows the change in range of values for columns ABV and EBC before and after imputation.

Data Clustering and Analysis:

A clean dataset is now at hand with no null values once the imputation process is complete. Now, moving on to the second objective of this report, which is to group the beers into different clusters. In general, clustering means grouping those objects together that have similar traits to one another into a single cluster that is distinct from the traits of another cluster. Various clustering methods are discussed below:

- Hierarchical Clustering: These clustering techniques tend to build a hierarchy of clusters for comparison. It's usually of two types:
    1. Agglomerative: It's the bottom-up approach.
    2. Divisive: It's a top to bottom approach.

  Hierarchical clustering, especially the agglomerative type, has the following general procedure of cluster formation:

  I. A distance between the clusters needs to be defined.
  II. The points need to be set into their own cluster.
  III. The final motive is to have just one single cluster at the end and to do that:
    - Keep calculating the distance between all clusters.
    - Merge the two closest clusters.
  IV. Observe the sequence of operations.

- K-means Clustering: It is an unsupervised technique that helps to group data points based on the similarity or closeness between each data point and the number of clusters one wants to form. There are some limitations to K-means, like choosing the K value or the number of clusters one wants and the fact that it is dependent on seeds. Also, K-means does not work well on datasets with categorical values, hence not useful for this report.

One might get scared looking at the complexity of these clustering techniques. Well, R programming makes clustering look much easier than it is with the help of different packages available in it. Now, let's see the various steps involved to perform Clustering in R programming:

- The first step is to import all the required packages and install them into the R script. Figure 5 shows the packages used for this analysis.

```
install.packages("fastcluster")
library("fastcluster")
install.packages("NbClust")
library("NbClust")
install.packages("cluster")
library("cluster")
```

Figure 5: Shows the required R packages

- Then, using the “hclust” function and ward distance measuring technique, a dendrogram is created that displays the hierarchical clustering technique. If the number of clusters required is set to 3, the clustering can be observed accordingly in figure 6. One important thing to be noted here is that this clustering was done only on the columns with numerical data and not on the categorical data of our dataset.

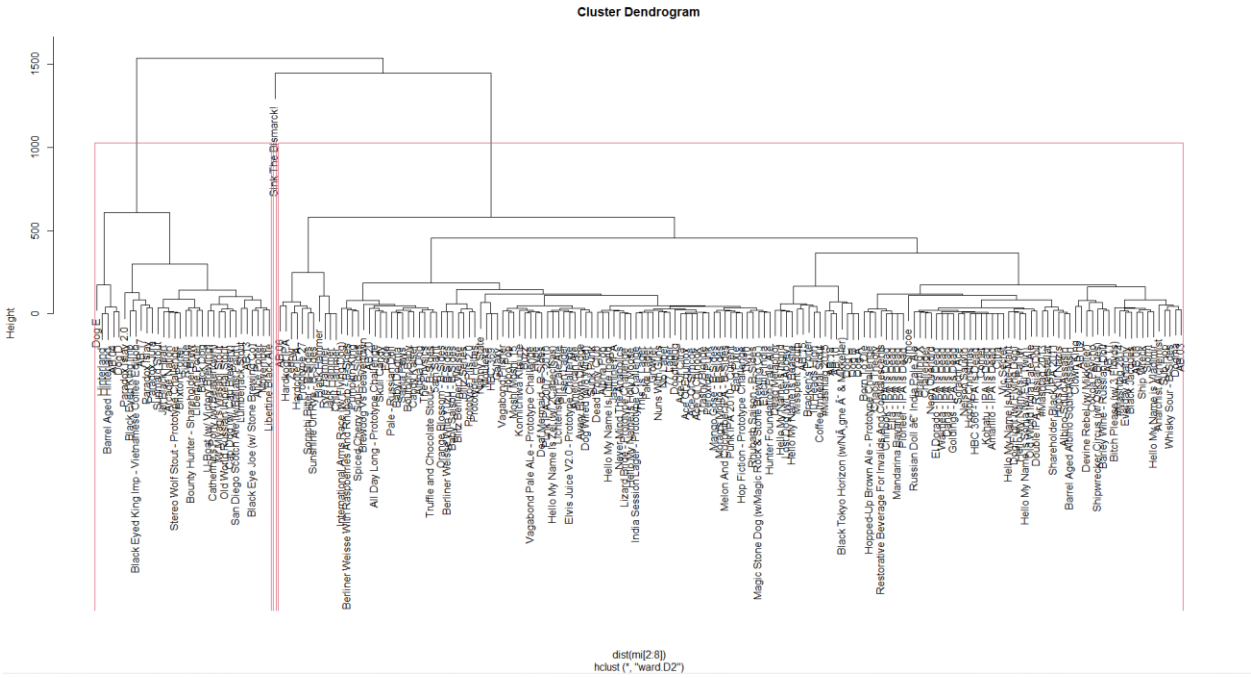

Figure 6: Dendrogram using hclust function

- The above technique of hierarchical clustering cannot be used because the resulting dendrogram shows that one individual beer type has been detected as a cluster using this technique. Also, the fact that a single element is so high on a dendrogram depicts that the clustering didn’t happen properly.
- Another approach towards hierarchical clustering is using the **Daisy** functions. Using the Daisy function in hierarchical clustering enable one to apply clustering over categorical data as well which means, one can apply clustering over the whole dataset. So, on applying the daisy function over the data, and then using the agglomerative clustering along with ward method gives us a dendrogram (figure 7). From the dendrogram below, some points can be made:
    1. This is a much better dendrogram than the one above.
    2. Such high value of the agglomerative coefficient for the dendrogram suggests that the clustering of the data has been done well as the value is closer to 1 for the coefficient.
    3. The Y-axis of a dendrogram depicts the height at which clustering takes place. Two distinct clusters are distinctly visible in the dendrogram but on a closer look to the left side cluster, it appears that the height difference between the clusters just below the cluster formed the height between 1.0 and 1.5 is much bigger to that of the major

cluster on the right formed closer to 0.5.Weel, with that information, we ca say that the two clusters on the left hand side can in fact be considered as distinct clusters for our dataset. Also, it's a fact that, larger the height between the last formed clusters, better is the clustering. So, in total 3 major clusters can be made in this data after interpreting the dendrogram.

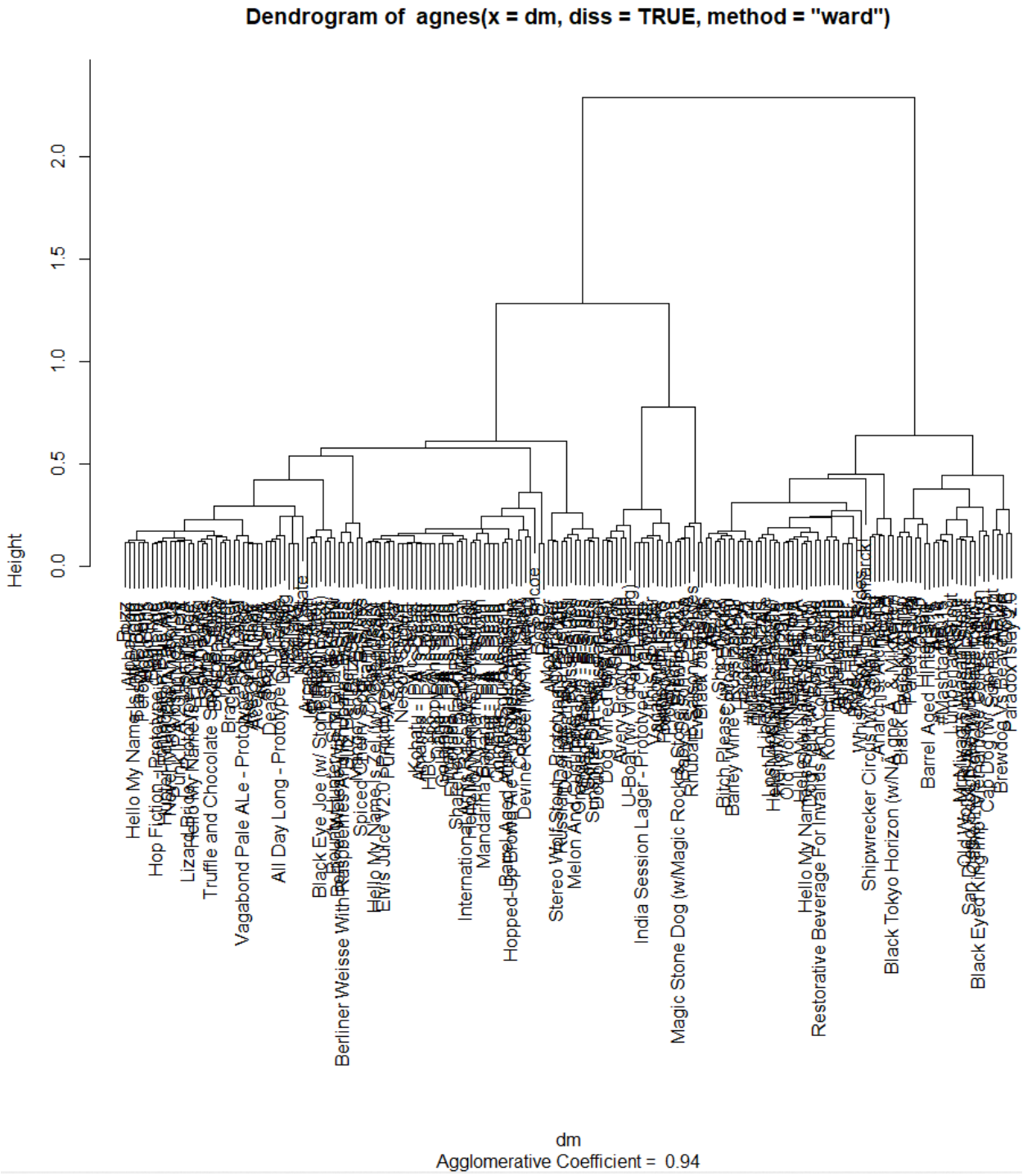

Figure 7: Dendrogram using the Daisy function and the agglomerative approach

- Since three clusters can be formed over our data, a data frame is created where an extra column has been added to our original data that displays the cluster number each beer belongs to. Check figure 8 and 9 for better visualization.

	Name	ABV	IBU	OG	EBC	PH	AttenuationLevel	FermentationTempCelsius
25	Nanny State	0.5	55.0	1007.0	30.0	4.4	28.60	19
148	Doodlebug	2.5	35.0	1027.0	10.0	4.2	70.40	19
130	Edge	2.7	36.0	1033.0	57.0	4.4	70.80	19
150	All Day Long - Prototype Challenge	2.8	30.0	1032.5	42.0	4.4	63.10	22
159	Blitz Berliner Weisse	3.0	8.0	1007.0	9.0	3.2	82.50	19
183	Blitz Series	3.2	8.0	1007.0	8.0	3.2	78.00	21
20	Skull Candy	3.5	33.0	1038.0	50.0	4.4	68.40	19
28	Berliner Weisse With Raspberries And Rhubarb - B-Sides	3.6	8.0	1040.0	40.0	3.2	83.00	21
76	Dead Pony Club	3.8	35.0	1040.0	25.0	4.4	70.00	19
117	Orange Blossom - B-Sides	3.8	20.0	1039.0	6.0	5.2	87.00	20
59	Pale - Russian Doll	4.0	35.0	1041.0	45.0	5.2	75.60	19
79	Peroxide Punk	4.0	40.0	1039.0	18.0	4.4	76.90	19

Figure 8: Depicts the first half of the dataset

	Yeast	cutree.clust..k...3.
25	Wyeast 1056 - American Ale	1
148	Wyeast 1056 - American Ale	1
130	Wyeast 1056 - American Ale	1
150	Wyeast 1056 - American Ale	1
159	Wyeast 1056 - American Ale	1
183	Wyeast 1056 - American Ale	1
20	Wyeast 1056 - American Ale	1
28	Wyeast 1056 - American Ale	1
76	Wyeast 1056 - American Ale	1
117	Wyeast 1056 - American Ale	1
59	Wyeast 1056 - American Ale	1
79	Wyeast 1056 - American Ale	1

Figure 9: Depicts the other half of the dataset with the cluster number each beer belongs to

Conclusion:

The beers were successfully clustered using the hierarchical clustering approach. To get more information about which cluster does each beer belong to, run the following R code in the appendix section below.

Appendix:

The R-Code:

#Inputting the data set.

```
data <- read.csv("C:/Users/Priyanshu/Downloads/BrewdogNew.csv", header=TRUE, stringsAsFactors = T)
```

#Checking the number of Complete cases.

```
sum(complete.cases(data))
```

```
#Number of incomplete cases.
sum(!complete.cases(data))

#Importing a required package to check the number of missing values in the data set.
install.packages("VIM",dependencies = T)
library("VIM")
aggr(data, numbers=TRUE, prop=FALSE)

#8 missing values for ABV and 5 missing values for EBC.

#Checking the summary of the data.
summary(data)
unique(data$Yeast)

#Some Visualizations.
hist(data$ABV)
hist(data$EBC)

install.packages("corrgram")
library("corrgram")
corrgram(data)

#Creating an extra column for the missing values.
missdata <- data
missdata$missing <- as.numeric(!complete.cases(data))
corrgram(missdata)

missdata$missing
```

```
#Applying Imputation.
install.packages("mice")
library("mice")

imi <- mice( subset(data, select = c('Name', 'ABV', 'IBU','OG', 'EBC', 'PH','AttenuationLevel', 'FermentationTempCelsius', 'Yeast')), m = 5, maxit = 10)

mi <- complete(imi)

#Checking the changes made to the data due to the imputation process.
summary(data$ABV)

summary(mi$ABV)
summary(data$EBC)

summary(mi$EBC)

#Visualizing the difference in the ABV column in the data set using histogram.
hist(mi$ABV)
hist(data$ABV, breaks = 6)

summary(mi)
summary(data)

#Visualizing the difference in the ABV column in the data set using boxplot.
boxplot(data$ABV)
boxplot(mi$ABV)
```

```
#Importing the important libraries for clustering.
install.packages("fastcluster")
library("fastcluster")
install.packages("NbClust")
library("NbClust")
install.packages("cluster")
library("cluster")

#hierarchical clustering.
eurc <- hclust(dist(mi[2:8]), "ward.D2")
eurc

plot(eurc, labels=mi$Name)
rect.hclust(eurc, 3)

#K-means Clustering.
res <- NbClust(mi[2:8], min.nc=2, max.nc=15, method="ward.D2")

res$Best.nc
res$Best.partition

# k-means clustering on the data
km <- kmeans(mi[2:8], 3)
# Call back clustering information
km

# See how data has been clustered
eddf <- data.frame(mi, km$cluster)
```

```
# Sort data based on cluster number
eddf[order(eddf[3]),]

# Plot ABV vs EBC.
plot(mi[c("ABV","EBC")], col=km$cluster)
# Add labels
text(mi$ABV,mi$EBC, mi$Name, cex=0.6, pos=4)

#Hierarchical clustering using the Daisy function and Agglomeration technique.
dm <- daisy(mi[1:9])
clust <- agnes(dm, diss = TRUE, method="ward")
plot(clust, labels=mi$Name, which.plots= 2, cex = 1)
abline(h = 0.42, col = 'darkgreen')
bb = hclust(dm)
plot(bb)
dm
cluster <- data.frame(mi, cutree(clust, k=3))
cluster[order(cluster[2]),]
table(cutree(clust, k=3), mi$Yeast)
table(cutree(clust, k=3), mi$ABV)

library(dendextend)
hc_single <- clust
hc_tt <- eurc
hc_single <- as.dendrogram(hc_single)
hc_tt <- as.dendrogram(hc_tt)
tanglegram (hc_single, hc_tt)
```

```
dm <- daisy(mi[1:9])
clust <- agnes(dm, diss = TRUE, method="ward")
plot(clust, labels=mi$Name, which.plots= 2, cex = 1)
```