ASSIGNMENT 5

SAI ROHITH GUDA

11-30-2023

Introduction

The data includes 16 variables with observations about various cereal components, such as the cereal's name, manufacturer, calories, protein, fats, sodium, potassium, fiber, and vitamins, along with information about 77 cereals arranged in rows. The Hierarchical Clustering Model, an unsupervised learning algorithm with an arbitrary number of clusters, is used for the analysis. However, make your selection based on a comparison of the various clusters that the data points have created. Using R, the analysis is completed.

Data Loading and Processing

```r
# Load the data
cereals <- read.csv("Cereals.csv")
head(cereals)
```

```
##                         name mfr type calories protein fat sodium fiber carbo
## 1                  100%_Bran   N    C       70       4   1    130  10.0   5.0
## 2          100%_Natural_Bran   Q    C      120       3   5     15   2.0   8.0
## 3                   All-Bran   K    C       70       4   1    260   9.0   7.0
## 4 All-Bran_with_Extra_Fiber   K    C       50       4   0    140  14.0   8.0
## 5             Almond_Delight   R    C      110       2   2    200   1.0  14.0
## 6    Apple_Cinnamon_Cheerios   G    C      110       2   2    180   1.5  10.5
##   sugars potass vitamins shelf weight cups   rating
## 1      6    280       25     3      1 0.33 68.40297
## 2      8    135        0     3      1 1.00 33.98368
## 3      5    320       25     3      1 0.33 59.42551
## 4      0    330       25     3      1 0.50 93.70491
## 5      8     NA       25     3      1 0.75 34.38484
## 6     10     70       25     1      1 0.75 29.50954
```

Data processing is done by omitting the duplicated rows or null values from the data

```r
cereals<-na.omit(cereals)
head(cereals)
```

```
##                         name mfr type calories protein fat sodium fiber carbo
## 1                  100%_Bran   N    C       70       4   1    130  10.0   5.0
## 2          100%_Natural_Bran   Q    C      120       3   5     15   2.0   8.0
## 3                   All-Bran   K    C       70       4   1    260   9.0   7.0
## 4 All-Bran_with_Extra_Fiber   K    C       50       4   0    140  14.0   8.0
## 6   Apple_Cinnamon_Cheerios   G    C      110       2   2    180   1.5  10.5
## 7               Apple_Jacks   K    C      110       2   0    125   1.0  11.0
##   sugars potass vitamins shelf weight cups   rating
## 1      6    280       25     3      1 0.33 68.40297
## 2      8    135        0     3      1 1.00 33.98368
## 3      5    320       25     3      1 0.33 59.42551
## 4      0    330       25     3      1 0.50 93.70491
## 6     10     70       25     1      1 0.75 29.50954
## 7     14     30       25     2      1 1.00 33.17409
```

Data Selection and Normalization

Using the scale() function, the data is normalized by dividing by the standard deviation and subtracting the mean from the numeric columns (in this case, 4 to 12).

```r
# Normalize the data
cereals_norm <- scale(cereals[, 4:12])
head(cereals_norm)
```

```
##     calories    protein        fat     sodium       fiber      carbo     sugars
## 1 -1.8659155  1.3817478  0.0000000 -0.3910227  3.22866747 -2.5001396 -0.2542051
## 2  0.6537514  0.4522084  3.9728810 -1.7804186 -0.07249167 -1.7292632  0.2046041
## 3 -1.8659155  1.3817478  0.0000000  1.1795987  2.81602258 -1.9862220 -0.4836096
## 4 -2.8737823  1.3817478 -0.9932203 -0.2702057  4.87924705 -1.7292632 -1.6306324
## 6  0.1498180 -0.4773310  0.9932203  0.2130625 -0.27881412 -1.0868662  0.6634132
## 7  0.1498180 -0.4773310 -0.9932203 -0.4514312 -0.48513656 -0.9583868  1.5810314
##       potass   vitamins
## 1  2.5605229 -0.1818422
## 2  0.5147738 -1.3032024
## 3  3.1248675 -0.1818422
## 4  3.2659536 -0.1818422
## 6 -0.4022862 -0.1818422
## 7 -0.9666308 -0.1818422
```

Distance Matrix

A dissimilarity matrix is computed using the Euclidean distance metric. This matrix captures the pairwise distances between observations in the normalized dataset.

```r
dissimilarity_matrix <- dist(cereals_norm, method = "euclidean")
```

Hierarchical Cluster Analysis

In statistics and data analysis, hierarchical cluster analysis, or HCA, is a technique used to cluster together related objects or data points. A hierarchical structure with related elements grouped together at varying levels of granularity is the intended method of data organization. This method is especially helpful for examining and displaying the underlying structure of a dataset. An illustration of the outcome of hierarchical clustering is frequently a dendrogram. A dendrogram is a diagram that resembles a tree and shows how the clusters are organized hierarchically. Each node in the tree represents a cluster, and the height at which branches merge represents the dissimilarity between clusters. The distance metric and linkage technique (which determines how far apart clusters are) that are selected can have a big influence on the clustering outcomes.

Typical techniques for linking consist of:

- Single Linkage: Calculates the separation between the two clusters' nearest members.
- Complete Linkage: Measures the separation between the two clusters' furthest members, or complete linkage.
- Average Linkage: The average distance between each pair of members in the two clusters is measured by the Average Linkage.
- Ward's Method: Reduces the variance within each cluster using Ward's method.

Because of its adaptability, hierarchical clustering can be used with a wide range of data types, including mixed, categorical, and numerical datasets. It is widely used in many domains where knowing the natural grouping of data is crucial, such as biology for classifying species, marketing for customer segmentation, and many more.

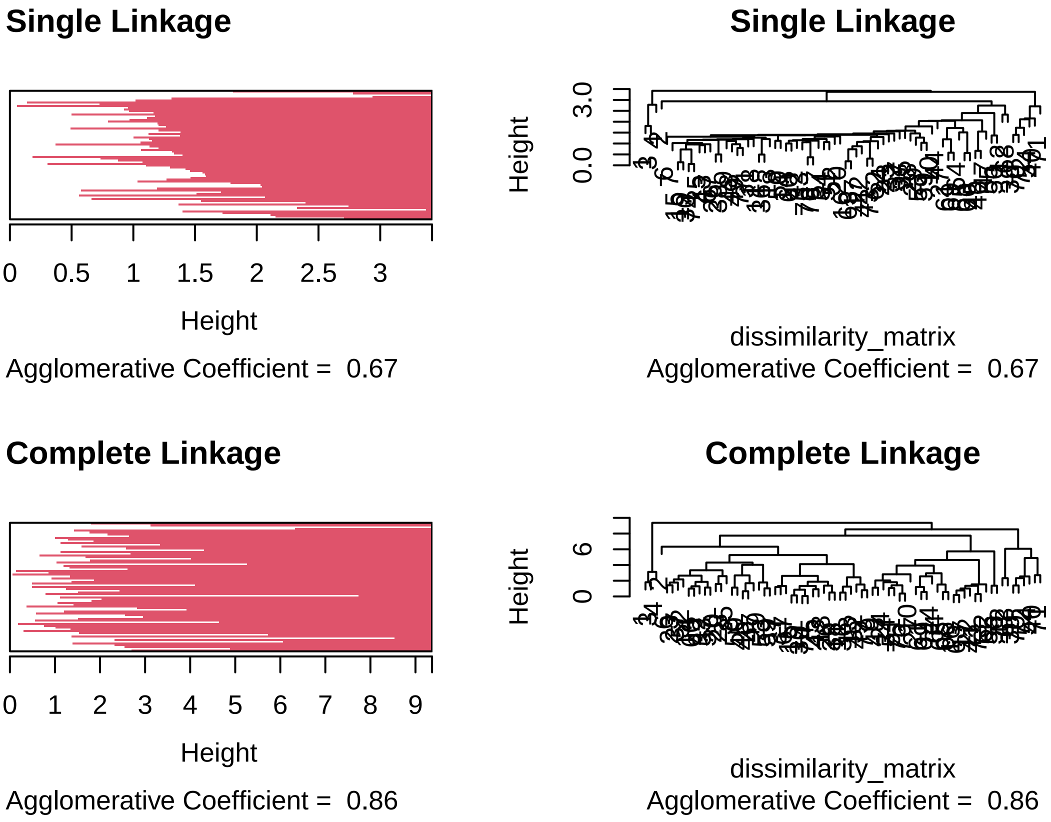
```r
agnes_single <- agnes(dissimilarity_matrix, method = "single")
agnes_complete <- agnes(dissimilarity_matrix, method = "complete")
agnes_average <- agnes(dissimilarity_matrix, method = "average")
agnes_ward <- agnes(dissimilarity_matrix, method = "ward")
```

The Agnes function from the "cluster" library is used to carry out hierarchical clustering, employing the single, complete, average, and Ward's method linkage techniques.

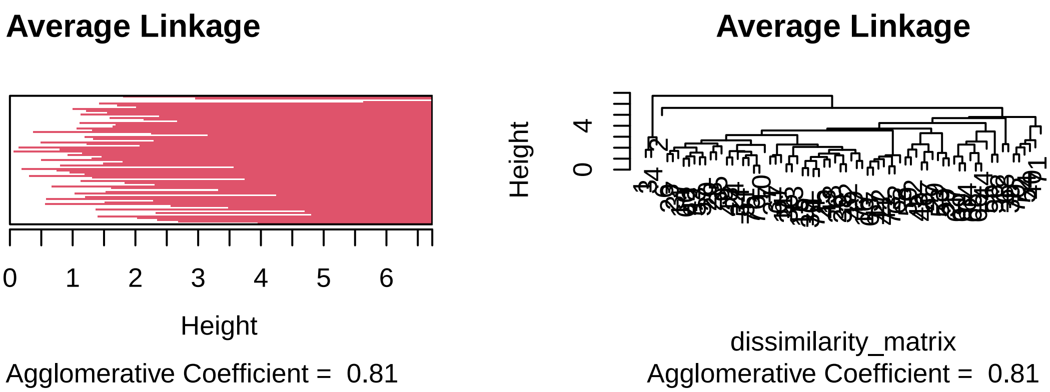
Visualization by dendgram

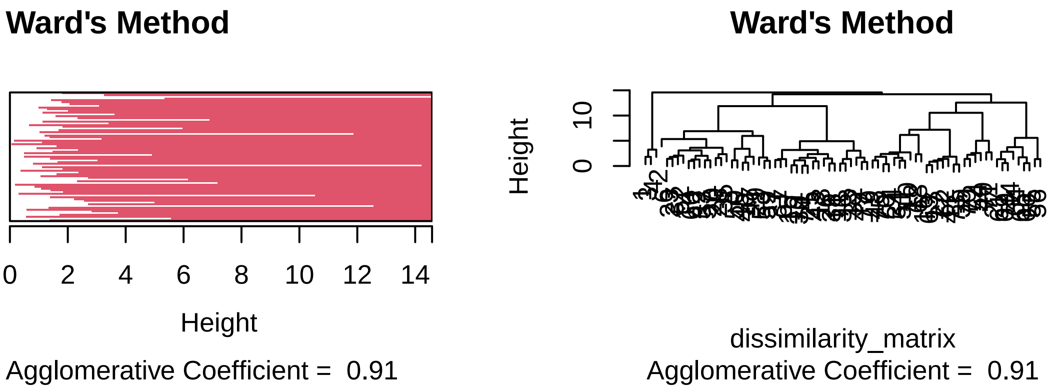
Dendrograms for each linkage method are plotted in a 2x2 layout using the par() function and plot().

```r
# Plot the dendrograms
par(mfrow = c(2, 2))
plot(agnes_single, main = "Single Linkage")
plot(agnes_complete, main = "Complete Linkage")
```



```r
plot(agnes_average, main = "Average Linkage")
plot(agnes_ward, main = "Ward's Method")
```





Selecting the number of Clusters

For complete linkage, the dendrogram is chopped at a height of three, and the resulting cluster assignments are kept in c.complete. The number of observations in each cluster can be printed using the table() function.

```
# Cut the dendrogram at a height of 3 and obtain the cluster assignments
c.complete <- cutree(agnes_complete, k = 3)
# Print the number of observations in each cluster
table(c.complete)
```

```
## c.complete
##  1  2  3
##  3 63  8
```

```
# Cut the dendrogram at a height of 3 and obtain the cluster assignments
c.single <- cutree(agnes_single, k = 3)
# Print the number of observations in each cluster
table(c.single)
```

```
## c.single
##  1  2  3
##  3 65  6
```

```
# Cut the dendrogram at a height of 3 and obtain the cluster assignments
c.average <- cutree(agnes_average, k = 3)
# Print the number of observations in each cluster
table(c.average)
```

```
## c.average
##  1  2  3 
##  3  1 70
```

```
# Cut the dendrogram at a height of 3 and obtain the cluster assignments
c.ward <- cutree(agnes_ward, k = 3)
# Print the number of observations in each cluster
table(c.ward)
```

```
## c.ward
##  1  2  3 
##  3 39 32
```

Extracting Information from Clusters

Noe aggregate each method of clusters to exttract information using the mean method to make the comparison easy

```
# Extract information from each cluster for complete linkage
cluster_means_complete <- aggregate(cereals_norm, by = list(Cluster = c.complete), FUN = mean)
cluster_means_single <- aggregate(cereals_norm, by = list(Cluster = c.single), FUN = mean)
cluster_means_average <- aggregate(cereals_norm, by = list(Cluster = c.average), FUN = mean)
cluster_means_ward <- aggregate(cereals_norm, by = list(Cluster = c.ward), FUN = mean)

head(cluster_means_complete)
```

```
##   Cluster    calories    protein        fat      sodium      fiber
## 1       1 -2.20187108  1.3817478 -0.3310734  0.17279012  3.6413124
## 2       2  0.05383072 -0.1822392  0.0315308 -0.09856876 -0.1510907
## 3       3  0.40178472  0.9169781 -0.1241525  0.71143272 -0.1756529
##           carbo      sugars      potass   vitamins
## 1 -2.0718749196 -0.78948236  2.98378133 -0.1818422
## 2  0.0001102354  0.09172246 -0.13019146 -0.2886384
## 3  0.7760849908 -0.42625847 -0.09366023  2.3412183
```

```
head(cluster_means_single)
```

```
##   Cluster    calories     protein         fat      sodium      fiber
## 1       1 -2.20187108  1.38174776 -0.33107342  0.17279012  3.6413124
## 2       2  0.05678418 -0.07691407  0.03056062 -0.05924053 -0.1550206
## 3       3  0.48577362  0.14236189 -0.16553671  0.55537739 -0.1412658
##         carbo      sugars      potass   vitamins
## 1 -2.07187492 -0.78948236  2.98378133 -0.1818422
## 2  0.01410335  0.05284413 -0.13422250 -0.2853524
## 3  0.88315115 -0.17773687 -0.03781363  3.1822385
```

```r
head(cluster_means_average)
```

```
##   Cluster   calories     protein         fat      sodium       fiber      carbo
## 1       1 -2.2018711  1.38174776 -0.33107342  0.17279012  3.64131237 -2.0718749
## 2       2  0.6537514  0.45220836  3.97288104 -1.78041856 -0.07249167 -1.7292632
## 3       3  0.0850266 -0.06567788 -0.04256658  0.01802926 -0.15502065  0.1134984
##        sugars    potass    vitamins
## 1 -0.78948236  2.9837813 -0.18184220
## 2  0.20460407  0.5147738 -1.30320244
## 3  0.03091204 -0.1352303  0.02641041
```

```r
head(cluster_means_ward)
```

```
##   Cluster   calories    protein        fat       sodium      fiber      carbo
## 1       1 -2.2018711  1.3817478 -0.3310734  0.172790124  3.6413124 -2.0718749
## 2       2  0.4599309 -0.1913189  0.4838765 -0.016180105 -0.1677174 -0.4312919
## 3       3 -0.3541153  0.1036311 -0.5586864  0.003520429 -0.1369674  0.7198753
##       sugars      potass   vitamins
## 1 -0.7894824  2.98378133 -0.1818422
## 2  0.7222349 -0.06404118 -0.2105950
## 3 -0.8062098 -0.20167931  0.2737104
```

Complete Linkage Method

Cluster 1:

- Low calories
- High protein
- Low fat
- High fiber
- Low carbohydrates
- Low sugars
- High potassium
- High vitamins

Cluster 2: Moderate values for most variables.

Cluster 3: Moderate values with a potential emphasis on higher fat.

Single Linkage Method

Cluster 1: Similar characteristics to Cluster 1 of the Complete Linkage method.

Cluster 2: Moderate values for most variables.

Cluster 3: Moderate values with a potential emphasis on higher fat.

Average Linkage Method

Cluster 1: Similar characteristics to Cluster 1 of the Complete Linkage method.

Cluster 2:

- High calories
- Moderate protein
- High fat
- Low fiber
- High carbohydrates
- Low sugars
- Moderate potassium
- Low vitamins

Cluster 3:

- Low calories
- Low protein
- Low fat
- Low fiber
- Low carbohydrates
- Low sugars
- Low potassium
- Low vitamins

Ward's Method

Cluster 1: Similar characteristics to Cluster 1 of the Complete Linkage method.

Cluster 2: Moderate values for most variables.

Cluster 3: Moderate values with a potential emphasis on higher fat.

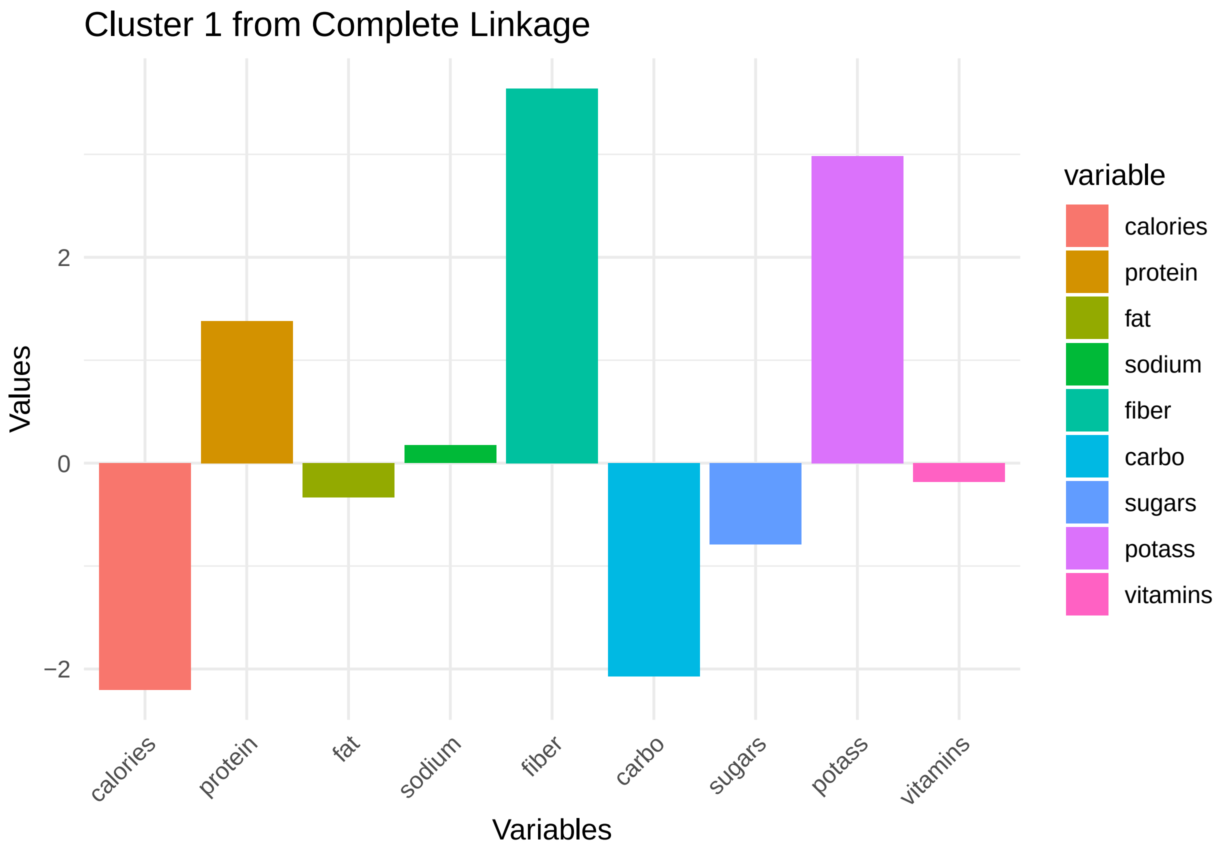
### Conclusion

Based on the criteria of “Low Calories, High Protein, Low Fat, High Fiber, Low Carbohydrates, Low Sugars, High Potassium, High Vitamins,” Cluster 1 from the Complete Linkage method seems to match these criteria the closest. It has low calories, high protein, low fat, high fiber, low carbohydrates, low sugars, high potassium, and high vitamins.

```r
# Extract data for Cluster 1 from complete linkage
cluster1_complete <- subset(cluster_means_complete, Cluster == 1)

# Melt the data for better plotting with ggplot
melted_data <- melt(cluster1_complete, id.vars = "Cluster")

# Create a bar plot using ggplot
ggplot(melted_data, aes(x = variable, y = value, fill = variable)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Cluster 1 from Complete Linkage",
       x = "Variables", y = "Values") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



## Conclusion Extracting the data from the cluster of the best cereals that will be allowed to the school’s canteen

```r
# Extract data for Cluster 1 from complete linkage
cluster1_complete <- subset(cereals, c.complete == 1)
```

```r
# Display the data
head(cluster1_complete)
```

```
##                          name mfr type calories protein fat sodium fiber carbo
## 1                 100%_Bran   N    C       70       4   1    130    10     5
## 3                  All-Bran   K    C       70       4   1    260     9     7
## 4 All-Bran_with_Extra_Fiber   K    C       50       4   0    140    14     8
##   sugars potass vitamins shelf weight cups   rating
## 1      6    280       25     3      1 0.33 68.40297
## 3      5    320       25     3      1 0.33 59.42551
## 4      0    330       25     3      1 0.50 93.70491
```