Mukhtar_Assignment5_MIS_64060

Contents

Problem Statement 1
Data Preparation 1
Load Data Set and Libraries 2
Review Data Structure 2
Data Preprocessing 4
Assignment Task A 5
Assignment Task B 14
Assignment Task C 17
Assignment Task D 24

Problem Statement

Hierarchical Clustering The dataset Cereals.csv includes nutritional information, store display, and consumer ratings for 77 breakfast cereals.

Data Preparation

```
getwd()
```

```
## [1] "C:/Users/Mukht/OneDrive/Desktop/Kent State University/College of Business Admin-Bus. Analytics I
```

```
setwd("C:\\Users\\Mukht\\OneDrive\\Desktop\\Kent State University\\College of Business Admin-Bus. Analy

Assignment5<-read.csv("Cereals.csv")
str(Assignment5)
```

```
## 'data.frame':    77 obs. of  16 variables:
##  $ name    : chr  "100%_Bran" "100%_Natural_Bran" "All-Bran" "All-Bran_with_Extra_Fiber" ...
##  $ mfr     : chr  "N" "Q" "K" "K" ...
##  $ type    : chr  "C" "C" "C" "C" ...
##  $ calories: int  70 120 70 50 110 110 110 130 90 90 ...
##  $ protein : int  4 3 4 4 2 2 2 3 2 3 ...
##  $ fat     : int  1 5 1 0 2 2 0 2 1 0 ...
##  $ sodium  : int  130 15 260 140 200 180 125 210 200 210 ...
##  $ fiber   : num  10 2 9 14 1 1.5 1 2 4 5 ...
```

```
##  $ carbo   : num  5 8 7 8 14 10.5 11 18 15 13 ...
##  $ sugars  : int  6 8 5 0 8 10 14 8 6 5 ...
##  $ potass  : int  280 135 320 330 NA 70 30 100 125 190 ...
##  $ vitamins: int  25 0 25 25 25 25 25 25 25 25 ...
##  $ shelf   : int  3 3 3 3 3 1 2 3 1 3 ...
##  $ weight  : num  1 1 1 1 1 1 1 1.33 1 1 ...
##  $ cups    : num  0.33 1 0.33 0.5 0.75 0.75 1 0.75 0.67 0.67 ...
##  $ rating  : num  68.4 34 59.4 93.7 34.4 ...
```

```
head(Assignment5)
```

```
##                        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
```

Load Data Set and Libraries

As a matter of priority, we would load all of the requisite packages that would be needed for this problem. Specifically, “ISLR”, “caret”, “dplyr”, “tidyverse”, “factoextra”, “ggplot2”, “proxy”, “NbClust”, “ppclust”, “dendextend”, and “cluster” would be loaded for the purpose of solving all the problems in question.

Next, we will import the “cereal” data set into the RStudio environment.

```
# Import data set from BlackBoard into the RStudio environment
Assignment5<- read.csv("cereals.csv")
```

Review Data Structure

A summary of the data set will be displayed to review the data set.

```
# Review first few rows of the data set
head(Assignment5)
```

```
##                        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
```

```
# Investigate the structure of the data set
str(Assignment5)
```

```
## 'data.frame':    77 obs. of  16 variables:
##  $ name    : chr  "100%_Bran" "100%_Natural_Bran" "All-Bran" "All-Bran_with_Extra_Fiber" ...
##  $ mfr     : chr  "N" "Q" "K" "K" ...
##  $ type    : chr  "C" "C" "C" "C" ...
##  $ calories: int  70 120 70 50 110 110 110 130 90 90 ...
##  $ protein : int  4 3 4 4 2 2 2 3 2 3 ...
##  $ fat     : int  1 5 1 0 2 2 0 2 1 0 ...
##  $ sodium  : int  130 15 260 140 200 180 125 210 200 210 ...
##  $ fiber   : num  10 2 9 14 1 1.5 1 2 4 5 ...
##  $ carbo   : num  5 8 7 8 14 10.5 11 18 15 13 ...
##  $ sugars  : int  6 8 5 0 8 10 14 8 6 5 ...
##  $ potass  : int  280 135 320 330 NA 70 30 100 125 190 ...
##  $ vitamins: int  25 0 25 25 25 25 25 25 25 25 ...
##  $ shelf   : int  3 3 3 3 3 1 2 3 1 3 ...
##  $ weight  : num  1 1 1 1 1 1 1 1.33 1 1 ...
##  $ cups    : num  0.33 1 0.33 0.5 0.75 0.75 1 0.75 0.67 0.67 ...
##  $ rating  : num  68.4 34 59.4 93.7 34.4 ...
```

```
# Investigate the summary of the data set
summary(Assignment5)
```

```
##      name               mfr                type              calories
##  Length:77          Length:77          Length:77          Min.   : 50.0
##  Class :character   Class :character   Class :character   1st Qu.:100.0
##  Mode  :character   Mode  :character   Mode  :character   Median :110.0
##                                                           Mean   :106.9
##                                                           3rd Qu.:110.0
##                                                           Max.   :160.0
##
##     protein            fat            sodium          fiber
##  Min.   :1.000   Min.   :0.000   Min.   :  0.0   Min.   : 0.000
##  1st Qu.:2.000   1st Qu.:0.000   1st Qu.:130.0   1st Qu.: 1.000
##  Median :3.000   Median :1.000   Median :180.0   Median : 2.000
##  Mean   :2.545   Mean   :1.013   Mean   :159.7   Mean   : 2.152
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:210.0   3rd Qu.: 3.000
##  Max.   :6.000   Max.   :5.000   Max.   :320.0   Max.   :14.000
##
##      carbo           sugars          potass          vitamins
##  Min.   : 5.0   Min.   : 0.000   Min.   : 15.00   Min.   :  0.00
##  1st Qu.:12.0   1st Qu.: 3.000   1st Qu.: 42.50   1st Qu.: 25.00
```

```
##  Median :14.5   Median : 7.000   Median : 90.00   Median : 25.00
##  Mean   :14.8   Mean   : 7.026   Mean   : 98.67   Mean   : 28.25
##  3rd Qu.:17.0   3rd Qu.:11.000   3rd Qu.:120.00   3rd Qu.: 25.00
##  Max.   :23.0   Max.   :15.000   Max.   :330.00   Max.   :100.00
##  NA's   :1      NA's   :1        NA's   :2
##      shelf           weight          cups           rating
##  Min.   :1.000   Min.   :0.50   Min.   :0.250   Min.   :18.04
##  1st Qu.:1.000   1st Qu.:1.00   1st Qu.:0.670   1st Qu.:33.17
##  Median :2.000   Median :1.00   Median :0.750   Median :40.40
##  Mean   :2.208   Mean   :1.03   Mean   :0.821   Mean   :42.67
##  3rd Qu.:3.000   3rd Qu.:1.00   3rd Qu.:1.000   3rd Qu.:50.83
##  Max.   :3.000   Max.   :1.50   Max.   :1.500   Max.   :93.70
##
```

Data Preprocessing

The data will be scaled prior to removing the NA values from the data set.

```
# Create duplicate of data set for preprocessing
Assignment5_scaled <- Assignment5
# Scale the data set prior to placing it into a clustering algorithm
Assignment5_scaled[ , c(4:16)] <- scale(Assignment5[ , c(4:16)])
# Remove NA values from data set
Assignment5_preprocessed <- na.omit(Assignment5_scaled)

# Review the scaled data set with NA's removed
head(Assignment5_preprocessed)
```

```
##                        name mfr type   calories    protein         fat
## 1                 100%_Bran   N    C -1.8929836  1.3286071 -0.01290349
## 2         100%_Natural_Bran   Q    C  0.6732089  0.4151897  3.96137277
## 3                   All-Bran   K    C -1.8929836  1.3286071 -0.01290349
## 4 All-Bran_with_Extra_Fiber   K    C -2.9194605  1.3286071 -1.00647256
## 6   Apple_Cinnamon_Cheerios   G    C  0.1599704 -0.4982277  0.98066557
## 7               Apple_Jacks   K    C  0.1599704 -0.4982277 -1.00647256
##       sodium       fiber      carbo     sugars     potass   vitamins      shelf
## 1 -0.3539844  3.29284661 -2.5087829 -0.2343906  2.5753685 -0.1453172  0.9515734
## 2 -1.7257708 -0.06375361 -1.7409943  0.2223705  0.5160205 -1.2642598  0.9515734
## 3  1.1967306  2.87327158 -1.9969238 -0.4627711  3.1434645 -0.1453172  0.9515734
## 4 -0.2346986  4.97114672 -1.7409943 -1.6046739  3.2854885 -0.1453172  0.9515734
## 6  0.2424445 -0.27354112 -1.1011705  0.6791317 -0.4071355 -0.1453172 -1.4507595
## 7 -0.4136273 -0.48332864 -0.9732057  1.5926539 -0.9752315 -0.1453172 -0.2495930
##       weight       cups     rating
## 1 -0.1967771 -2.1100340  1.8321876
## 2 -0.1967771  0.7690100 -0.6180571
## 3 -0.1967771 -2.1100340  1.1930986
## 4 -0.1967771 -1.3795303  3.6333849
## 6 -0.1967771 -0.3052601 -0.9365625
## 7 -0.1967771  0.7690100 -0.6756899
```

```
#Tidy data
Assignment5<-Assignment5 %>%
```

```
  pivot_longer(4:14, names_to = "content", values_to = "values")
Assignment5
```

```
## # A tibble: 847 x 7
##    name      mfr   type   cups rating content  values
##    <chr>     <chr> <chr> <dbl>  <dbl> <chr>     <dbl>
##  1 100%_Bran N     C      0.33   68.4 calories     70
##  2 100%_Bran N     C      0.33   68.4 protein       4
##  3 100%_Bran N     C      0.33   68.4 fat           1
##  4 100%_Bran N     C      0.33   68.4 sodium      130
##  5 100%_Bran N     C      0.33   68.4 fiber        10
##  6 100%_Bran N     C      0.33   68.4 carbo         5
##  7 100%_Bran N     C      0.33   68.4 sugars        6
##  8 100%_Bran N     C      0.33   68.4 potass      280
##  9 100%_Bran N     C      0.33   68.4 vitamins     25
## 10 100%_Bran N     C      0.33   68.4 shelf         3
## # ... with 837 more rows
```

After pre-processing and scaling the data, the total number of observations went from 77 to 74. Therefore, there were only 3 records with an "NA" value.

Assignment Task A

"Apply hierarchical clustering to the data using Euclidean distance to the normalized measurements. Use Agnes to compare the clustering from single linkage, complete linkage, average linkage, and Ward. Choose the best method."

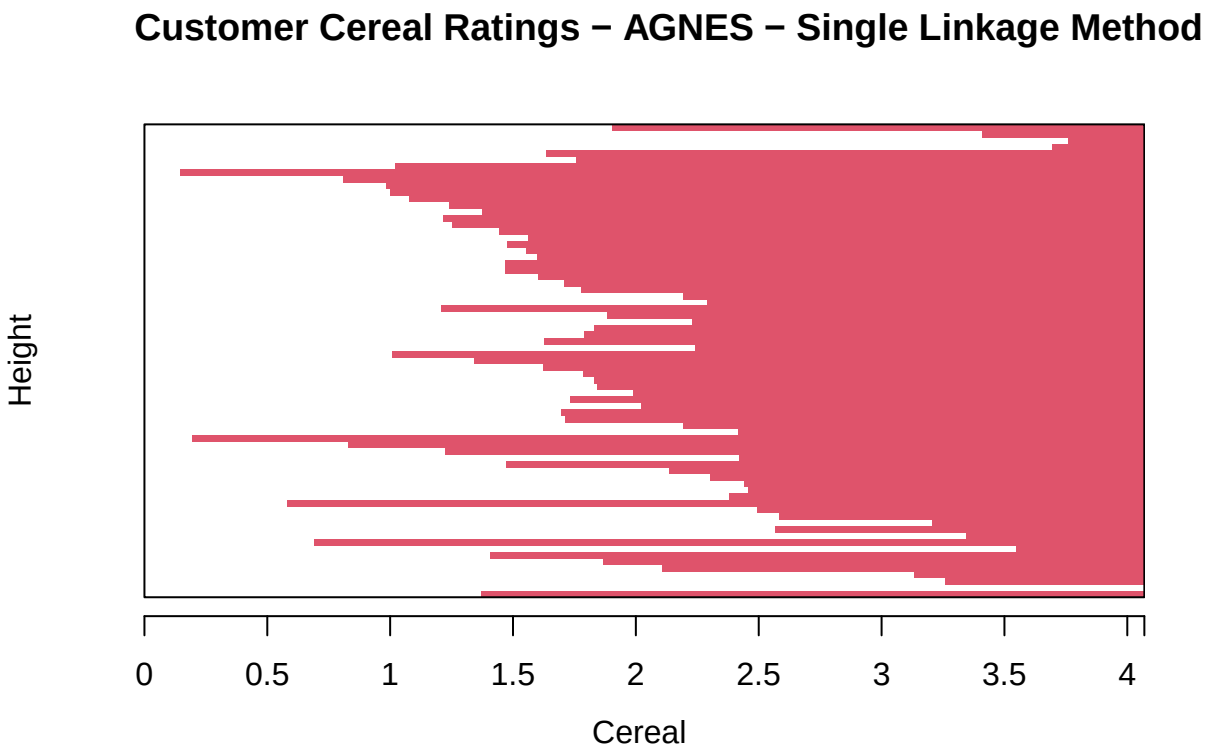
Single Linkage:

```
# Create the dissimilarity matrix for the numeric values in the data set via Euclidean distance measure
Assignment5_d_euclidean <- dist(Assignment5_preprocessed[ , c(4:16)], method = "euclidean")
# Perform hierarchical clustering via the single linkage method
ag_hc_single <- agnes(Assignment5_d_euclidean, method = "single")
# Plot the results of the different methods
plot(ag_hc_single,
     main = "Customer Cereal Ratings - AGNES - Single Linkage Method",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 1,
     cex = 0.55,
     hang = -1)
```

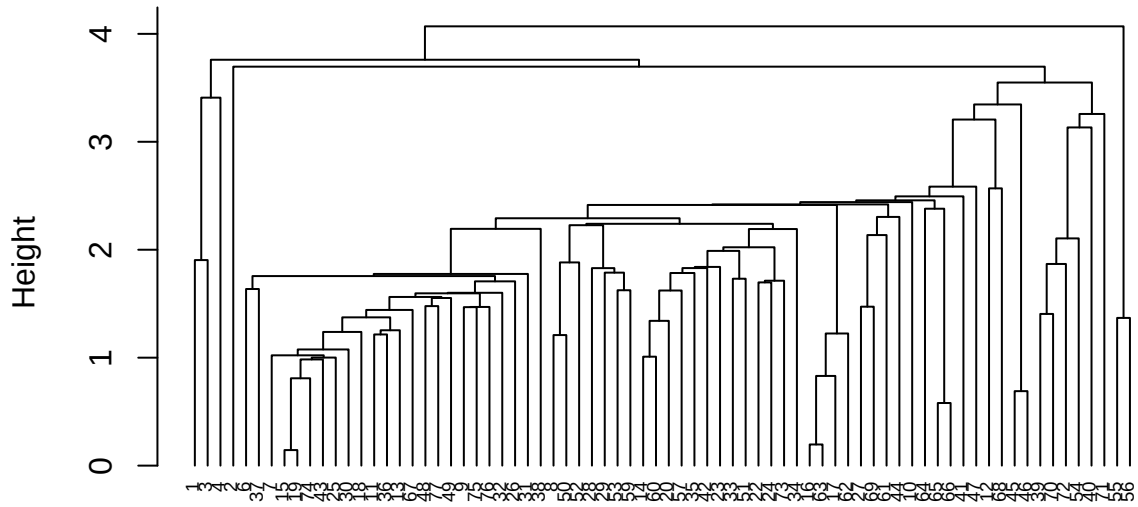
```
## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical
## parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"
## is not a graphical parameter
```

```
## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a
## graphical parameter
```



Agglomerative Coefficient = 0.61



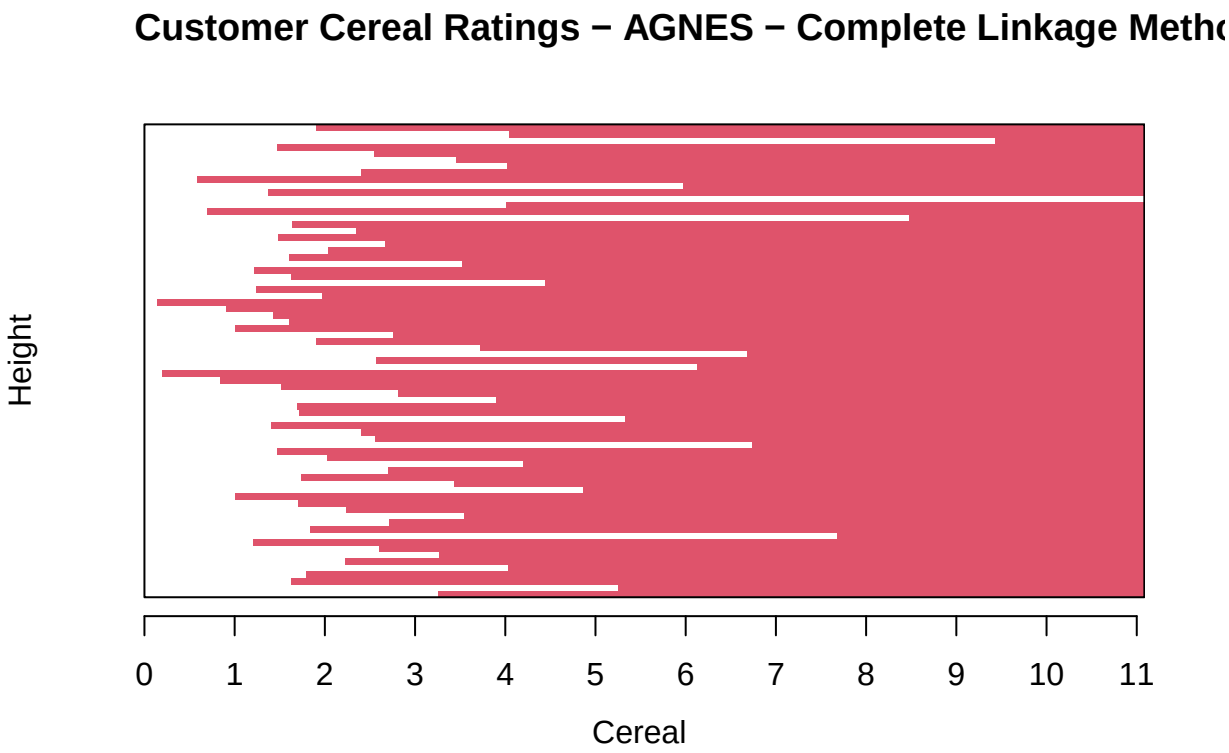
Complete Linkage:

```
# Perform hierarchical clustering via the complete linkage method
ag_hc_complete <- agnes(Assignment5_d_euclidean, method = "complete")
# Plot the results of the different methods
plot(ag_hc_complete,
     main = "Customer Cereal Ratings - AGNES - Complete Linkage Method",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 1,
     cex = 0.55,
     hang = -1)
```

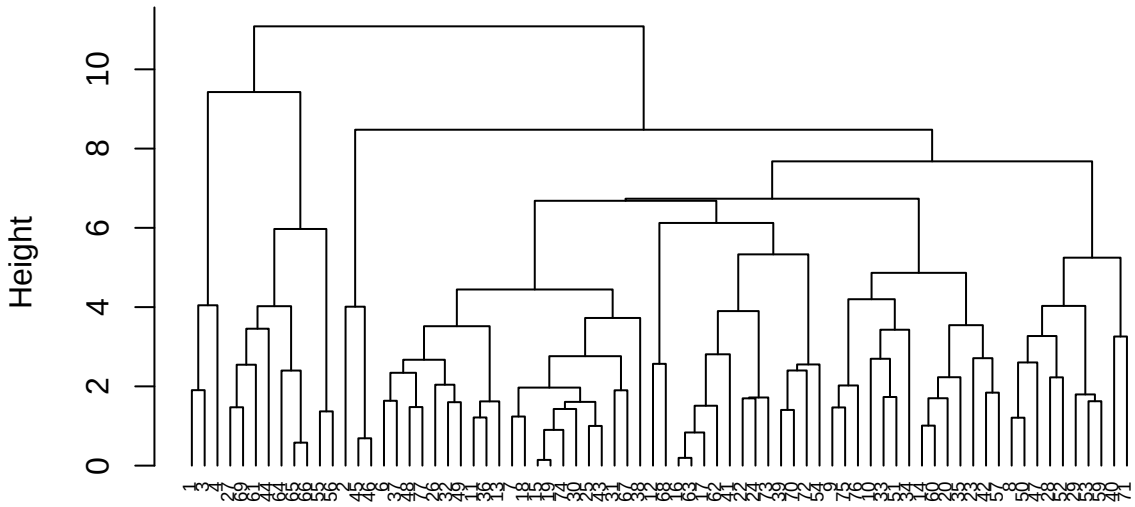
```
## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical
## parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"
## is not a graphical parameter
```

```
## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a
## graphical parameter
```



Agglomerative Coefficient = 0.84



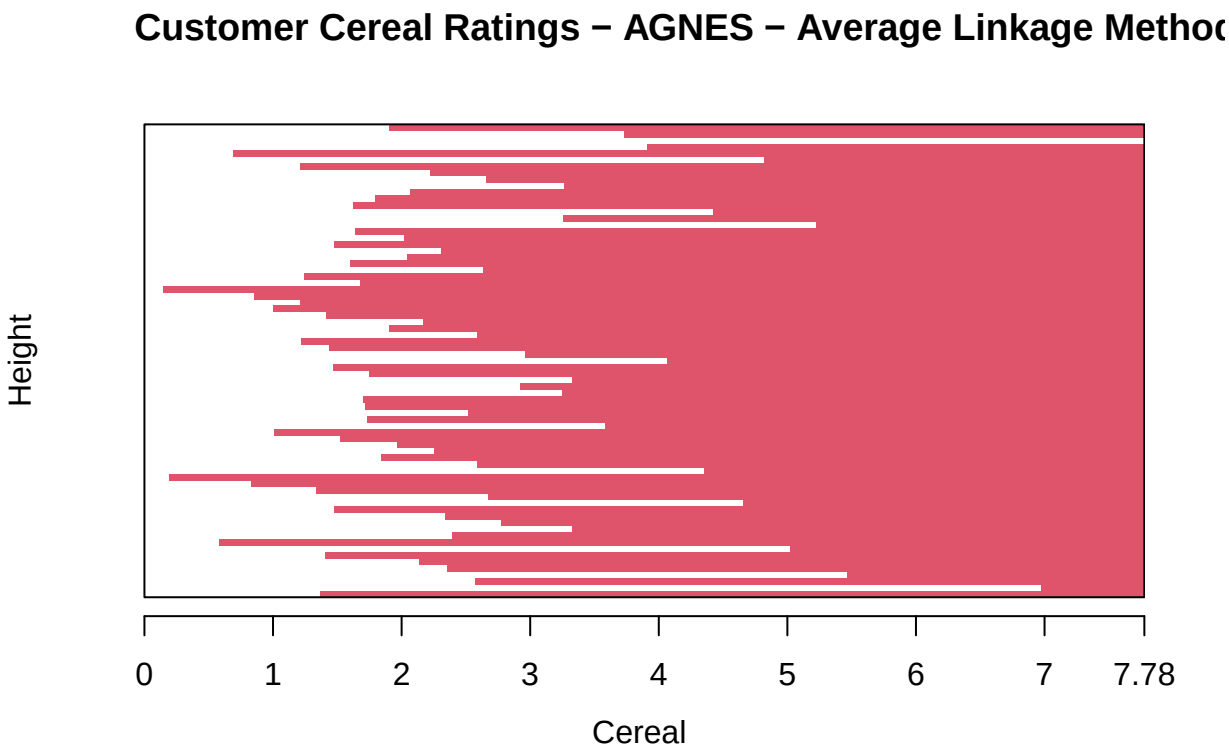
Average Linkage:

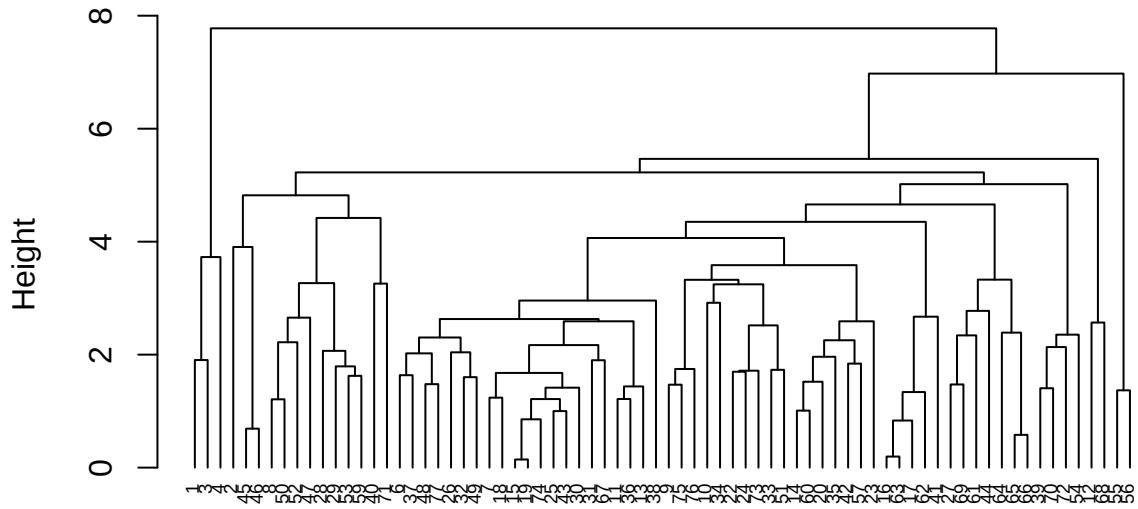
```
# Perform hierarchical clustering via the average linkage method
ag_hc_average <- agnes(Assignment5_d_euclidean, method = "average")
# Plot the results of the different methods
plot(ag_hc_average,
     main = "Customer Cereal Ratings - AGNES - Average Linkage Method",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 1,
     cex = 0.55,
     hang = -1)
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical
## parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"
## is not a graphical parameter
```

```
## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a
## graphical parameter
```


Customer Cereal Ratings − AGNES − Average Linkage Method
Height
0
1
2
3
4
5
6
7
7.78
Cereal
Agglomerative Coefficient = 0.78



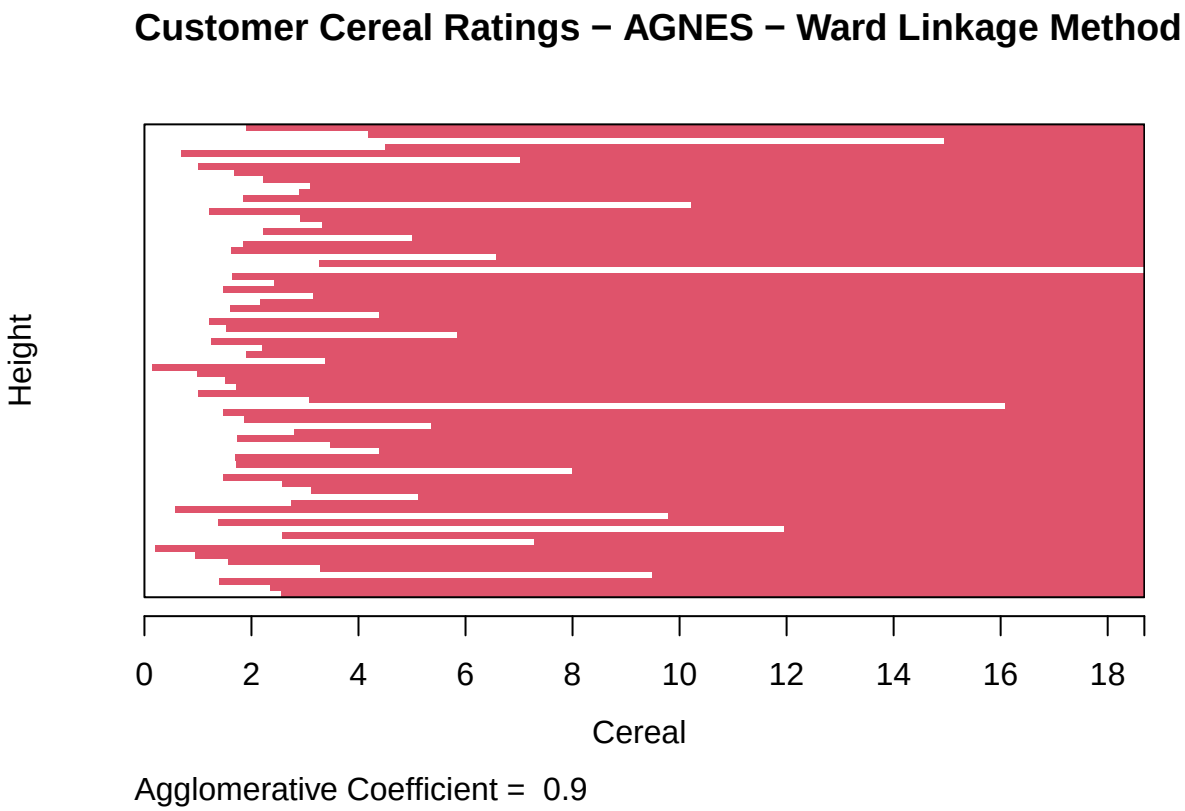
Ward Method:

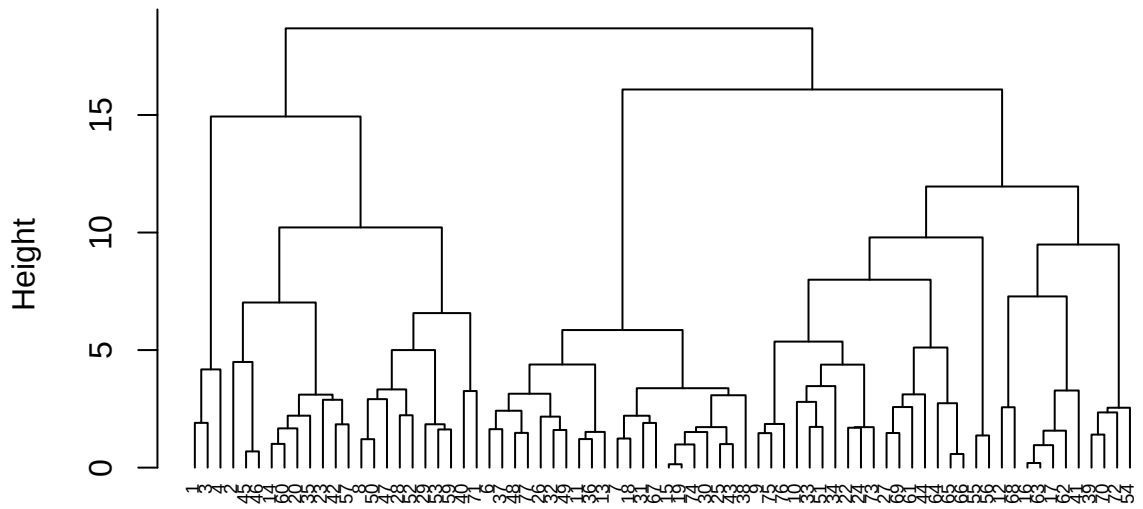
```
# Perform hierarchical clustering via the ward linkage method
ag_hc_ward <- agnes(Assignment5_d_euclidean, method = "ward")
# Plot the results of the different methods
plot(ag_hc_ward,
     main = "Customer Cereal Ratings - AGNES - Ward Linkage Method",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 1,
     cex = 0.55,
     hang = -1,)
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical
## parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"
## is not a graphical parameter
```

```
## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a
## graphical parameter
```

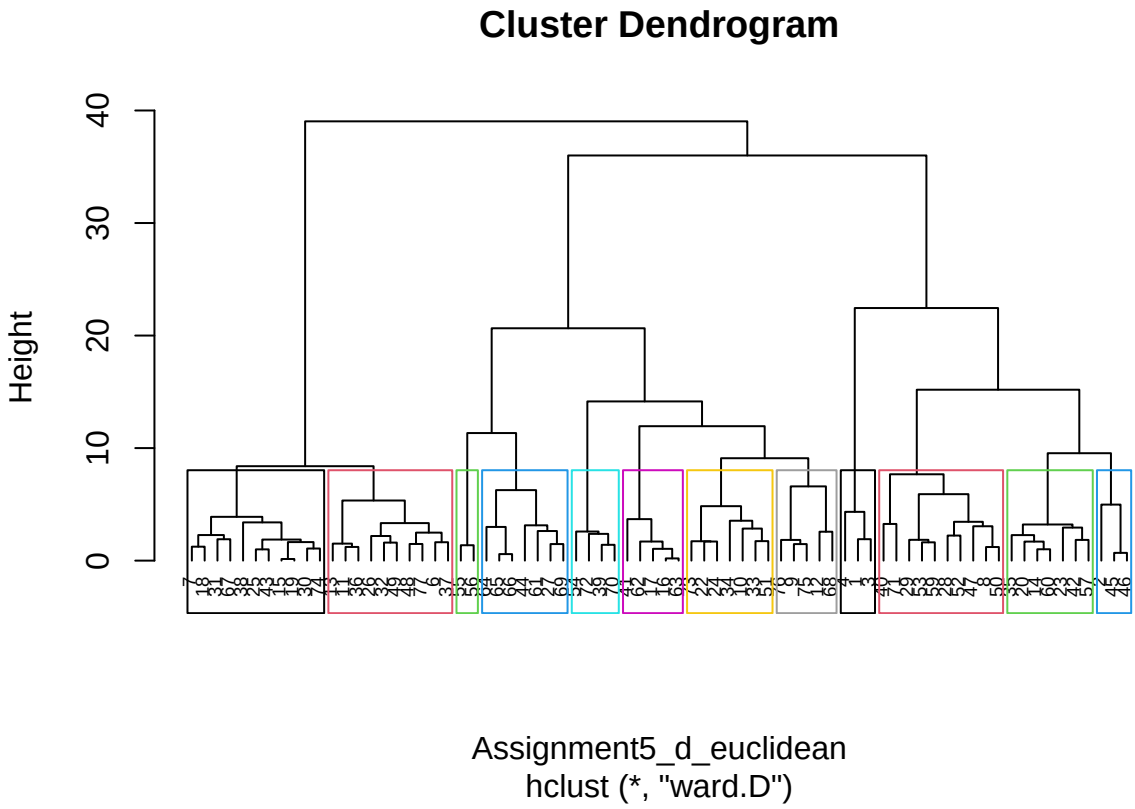

Customer Cereal Ratings − AGNES − Ward Linkage Method
Height
0
2
4
6
8
10
12
14
16
18
Cereal
Agglomerative Coefficient = 0.9




```
# Plot the results of the different methods
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.1.2
```

```
ag_hc_ward<-hclust(Assignment5_d_euclidean, method = "ward.D")
plot(ag_hc_ward, cex = 0.6, hang = -1)
rect.hclust(ag_hc_ward, k = 12, border = 1:12)
```



The best clustering method would be based on the agglomerative coefficient that is returned from each method. The close the value is to 1.0, the closer the clustering structure is. Therefore, the method with the value closest to 1.0 will be chosen.

Single Linkage: 0.61 Complete Linkage: 0.84 Average Linkage: 0.78 Ward Method: 0.90

As a result, the Ward method will be chosen as the best clustering model in this problem.

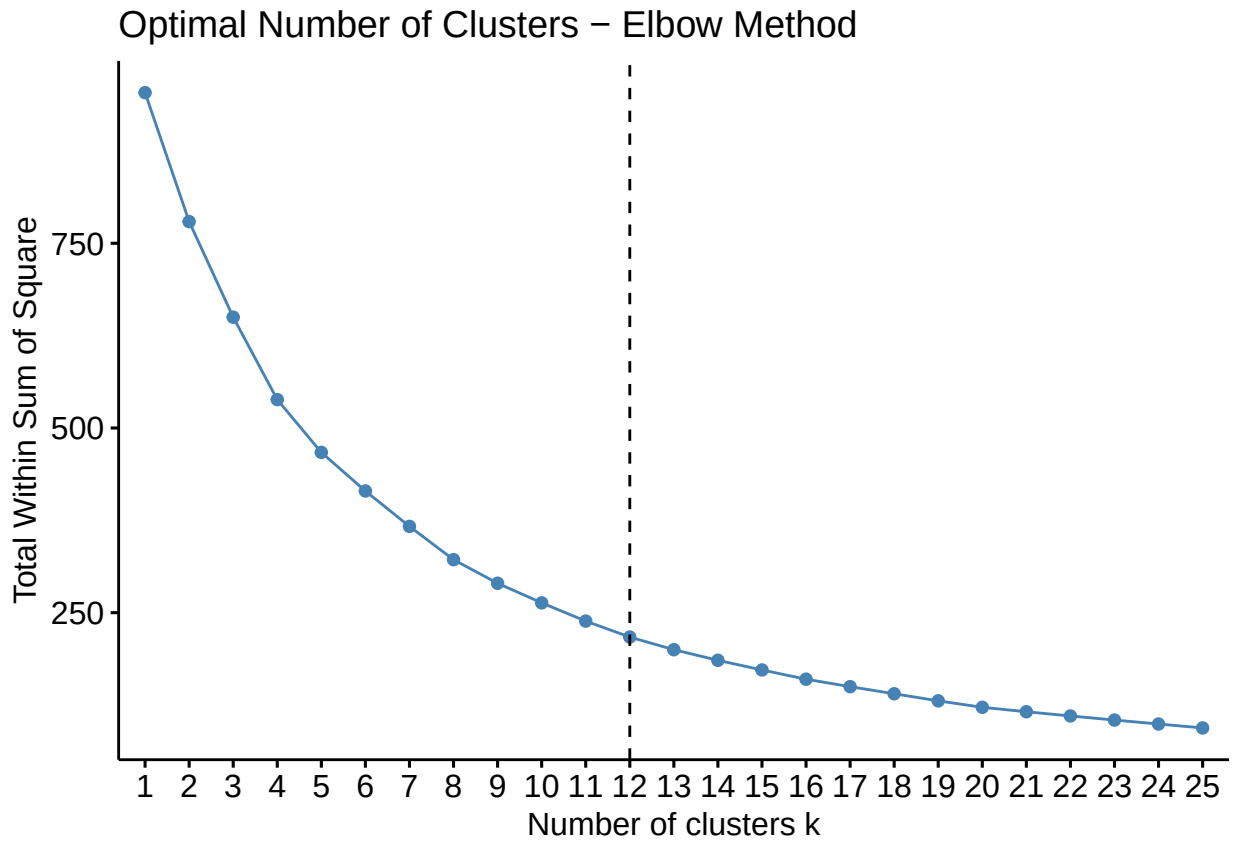
Assignment Task B

"How many clusters would you choose?"

To determine the appropriate number of clusters, we will use the elbow and silhouette methods.

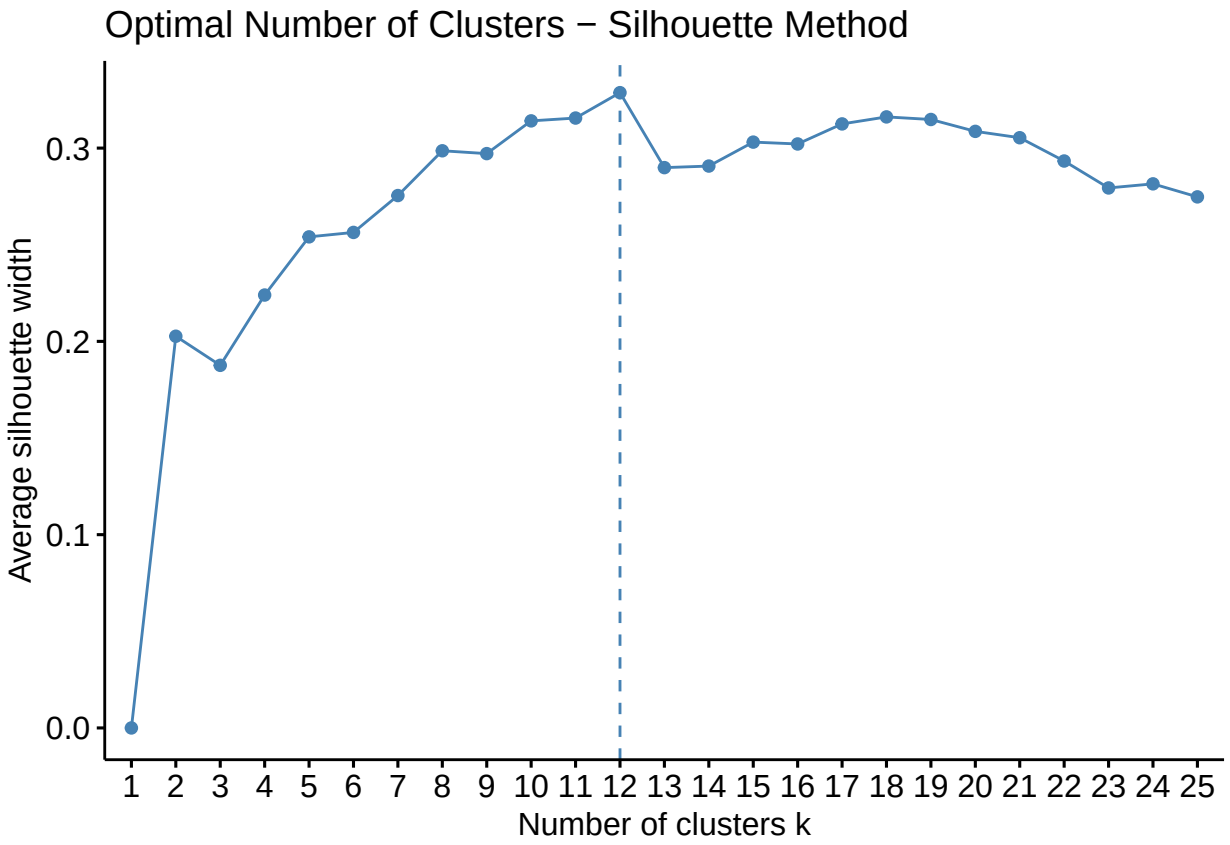
Elbow Method:

```
# Determine the optimal number of clusters for the dataset via the Elbow method
fviz_nbclust(Assignment5_preprocessed[ , c(4:16)], hcut, method = "wss", k.max = 25) +
  labs(title = "Optimal Number of Clusters - Elbow Method") +
  geom_vline(xintercept = 12, linetype = 2)
```



Silhouette Method:

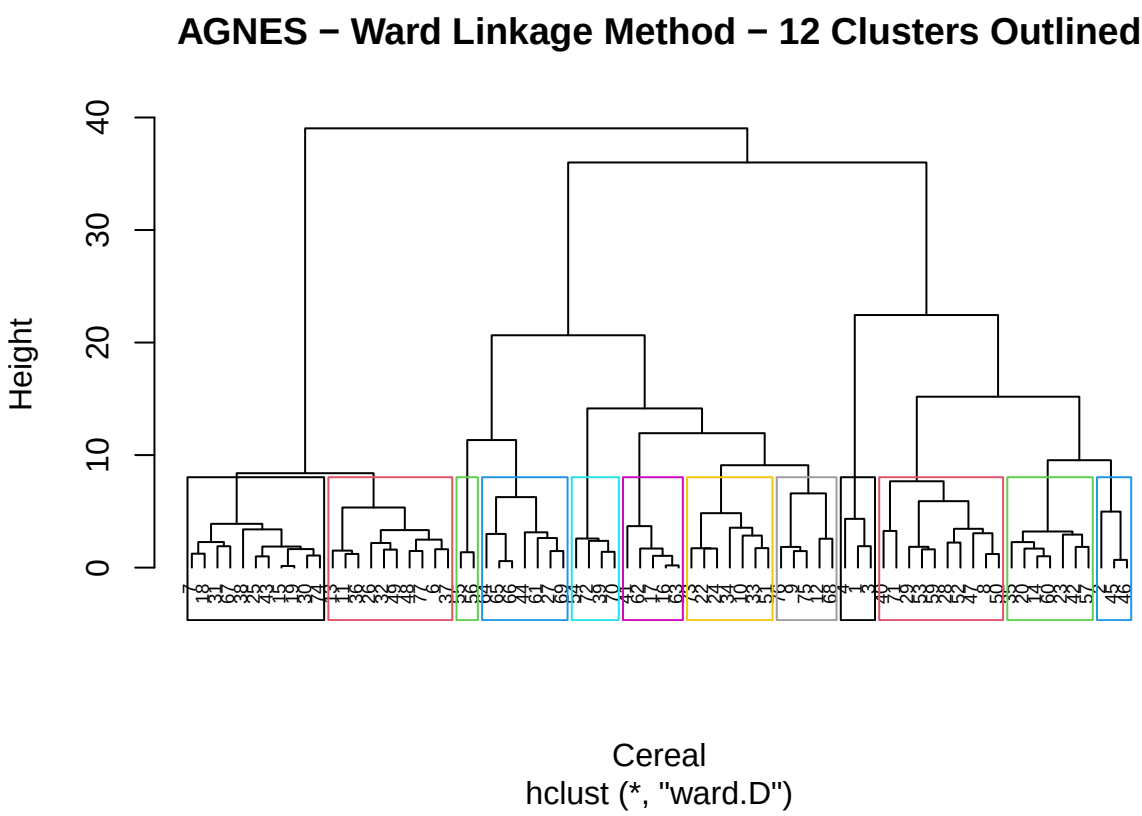
```
# Determine the optimal number of clusters for the dataset via the silhouette method
fviz_nbclust(Assignment5_preprocessed[, c(4:16)],
                                hcut,
                                method = "silhouette",
                                k.max = 25) +
  labs(title = "Optimal Number of Clusters - Silhouette Method")
```



Based on the agreement of the silhouette and elbow method, the appropriate number of clusters would be 12 in this case.

Below we will outline the 12 clusters on the hierarchical tree

```
# Plot of the Ward hierarchical tree with the 12 clusters outlined for reference
plot(ag_hc_ward,
     main = "AGNES - Ward Linkage Method - 12 Clusters Outlined",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 1,
     cex = 0.55,
     hang = -1)
rect.hclust(ag_hc_ward, k = 12, border = 1:12)
```



Assignment Task C

"Comment on the structure of the clusters and on their stability. Hint: To check stability, partition the data and see how well clusters formed based on one part apply to the other part. To do this: 1. Cluster partition A 2. Use the cluster centroids from A to assign each record in partition B (each record is assigned to the cluster with the closest centroid). 3. Assess how consistent the cluster assignments are compared to the assignments based on all the data"

All Data Assigned Clusters:

The assigned clusters for all data sets will be in "Assignment5_preprocessed_1":

```
# Cut the tree into 12 clusters for analysis
ward_clusters_12 <- cutree(ag_hc_ward, k = 12)
# Add the assigned cluster to the preprocessed data set
Assignment5_preprocessed_1<-cbind(cluster = ward_clusters_12, Assignment5_preprocessed)
```

Partition Data:

To check stability of clusters, the data set will be split into a 70/30 partition. The 70% will be used to create cluster assignments again, and then the remaining 30% will be assigned based on their closest centroid.

```
# Set the seed for randomized functions
set.seed(123)
# Split the data into 70% partition A and 30% partition B
Assignment5Index <- createDataPartition(Assignment5_preprocessed$protein, p=0.3, list = F)
```

```
Assignment5_preprocessed_PartitionB<- Assignment5_preprocessed[Assignment5Index, ]
Assignment5_preprocessed_PartitionA<- Assignment5_preprocessed[-Assignment5Index,]
summary(Assignment5_preprocessed_PartitionA)
```

```
##      name               mfr                type              calories
##  Length:50          Length:50          Length:50          Min.   :-2.9195
##  Class :character   Class :character   Class :character   1st Qu.:-0.3533
##  Mode  :character   Mode  :character   Mode  :character   Median : 0.1600
##                                                           Mean   : 0.1292
##                                                           3rd Qu.: 0.1600
##                                                           Max.   : 2.7262
##     protein               fat               sodium              fiber
##  Min.   :-1.411645   Min.   :-1.0065   Min.   :-1.904699   Min.   :-0.90290
##  1st Qu.:-0.498228   1st Qu.:-1.0065   1st Qu.:-0.294341   1st Qu.:-0.90290
##  Median :-0.041519   Median :-0.0129   Median : 0.182802   Median :-0.16865
##  Mean   :-0.004982   Mean   :-0.0129   Mean   :-0.002091   Mean   :-0.03019
##  3rd Qu.: 0.415190   3rd Qu.:-0.0129   3rd Qu.: 0.689766   3rd Qu.: 0.35582
##  Max.   : 3.155442   Max.   : 3.9614   Max.   : 1.554588   Max.   : 4.97115
##      carbo              sugars            potass            vitamins
##  Min.   :-1.99692   Min.   :-1.6047   Min.   :-1.11726   Min.   :-1.26426
##  1st Qu.:-0.71728   1st Qu.:-0.9195   1st Qu.:-0.83321   1st Qu.:-0.14532
##  Median :-0.07745   Median : 0.1082   Median :-0.12309   Median :-0.14532
##  Mean   : 0.01468   Mean   : 0.1265   Mean   : 0.01468   Mean   : 0.05609
##  3rd Qu.: 0.75432   3rd Qu.: 1.1359   3rd Qu.: 0.30299   3rd Qu.:-0.14532
##  Max.   : 2.09795   Max.   : 1.8210   Max.   : 3.28549   Max.   : 3.21151
##      shelf              weight             cups              rating
##  Min.   :-1.45076   Min.   :-0.1968   Min.   :-2.11003   Min.   :-1.75286
##  1st Qu.:-1.45076   1st Qu.:-0.1968   1st Qu.:-0.56308   1st Qu.:-0.82251
##  Median :-0.24959   Median :-0.1968   Median : 0.08148   Median :-0.25743
##  Mean   :-0.05741   Mean   : 0.1847   Mean   : 0.05140   Mean   :-0.08343
##  3rd Qu.: 0.95157   3rd Qu.:-0.1968   3rd Qu.: 0.76901   3rd Qu.: 0.44081
##  Max.   : 0.95157   Max.   : 3.1260   Max.   : 2.18705   Max.   : 3.63339
```

```
summary(Assignment5_preprocessed_PartitionB)
```

```
##      name               mfr                type              calories
##  Length:24          Length:24          Length:24          Min.   :-2.9195
##  Class :character   Class :character   Class :character   1st Qu.:-0.4816
##  Mode  :character   Mode  :character   Mode  :character   Median : 0.1600
##                                                           Mean   :-0.2463
##                                                           3rd Qu.: 0.2883
##                                                           Max.   : 2.2129
##     protein               fat               sodium             fiber
##  Min.   :-1.41165   Min.   :-1.0065   Min.   :-1.9047   Min.   :-0.90290
##  1st Qu.:-0.49823   1st Qu.:-1.0065   1st Qu.:-0.2645   1st Qu.:-0.48333
##  Median :-0.04152   Median :-0.0129   Median : 0.4214   Median :-0.06375
##  Mean   :-0.07958   Mean   :-0.0129   Mean   : 0.1033   Mean   : 0.09359
##  3rd Qu.: 0.41519   3rd Qu.: 0.2355   3rd Qu.: 0.6301   3rd Qu.: 0.35582
##  Max.   : 1.32861   Max.   : 1.9742   Max.   : 1.9124   Max.   : 3.29285
##      carbo              sugars            potass             vitamins
##  Min.   :-2.50878   Min.   :-1.6047   Min.   :-1.18827   Min.   :-1.264260
##  1st Qu.:-0.52533   1st Qu.:-0.9195   1st Qu.:-0.76220   1st Qu.:-0.145317
```

```
##  Median :-0.07745   Median :-0.2344   Median :-0.15859   Median :-0.145317
##  Mean   :-0.08812   Mean   :-0.2058   Mean   :-0.03728   Mean   :-0.005449
##  3rd Qu.: 0.30644   3rd Qu.: 0.5078   3rd Qu.: 0.26748   3rd Qu.:-0.145317
##  Max.   : 1.58609   Max.   : 1.3643   Max.   : 2.57537   Max.   : 3.211511
##      shelf             weight             cups              rating
##  Min.   :-1.4508   Min.   :-3.5195   Min.   :-2.45380   Min.   :-1.6261
##  1st Qu.:-0.2496   1st Qu.:-0.1968   1st Qu.:-0.64903   1st Qu.:-0.4105
##  Median : 0.9516   Median :-0.1968   Median :-0.30526   Median :-0.1001
##  Mean   : 0.1508   Mean   :-0.3601   Mean   :-0.09936   Mean   : 0.1093
##  3rd Qu.: 0.9516   3rd Qu.:-0.1968   3rd Qu.: 0.76901   3rd Qu.: 0.6920
##  Max.   : 0.9516   Max.   : 1.9963   Max.   : 2.91755   Max.   : 1.8322
```

Re-Run Clustering with Partitioned Data:

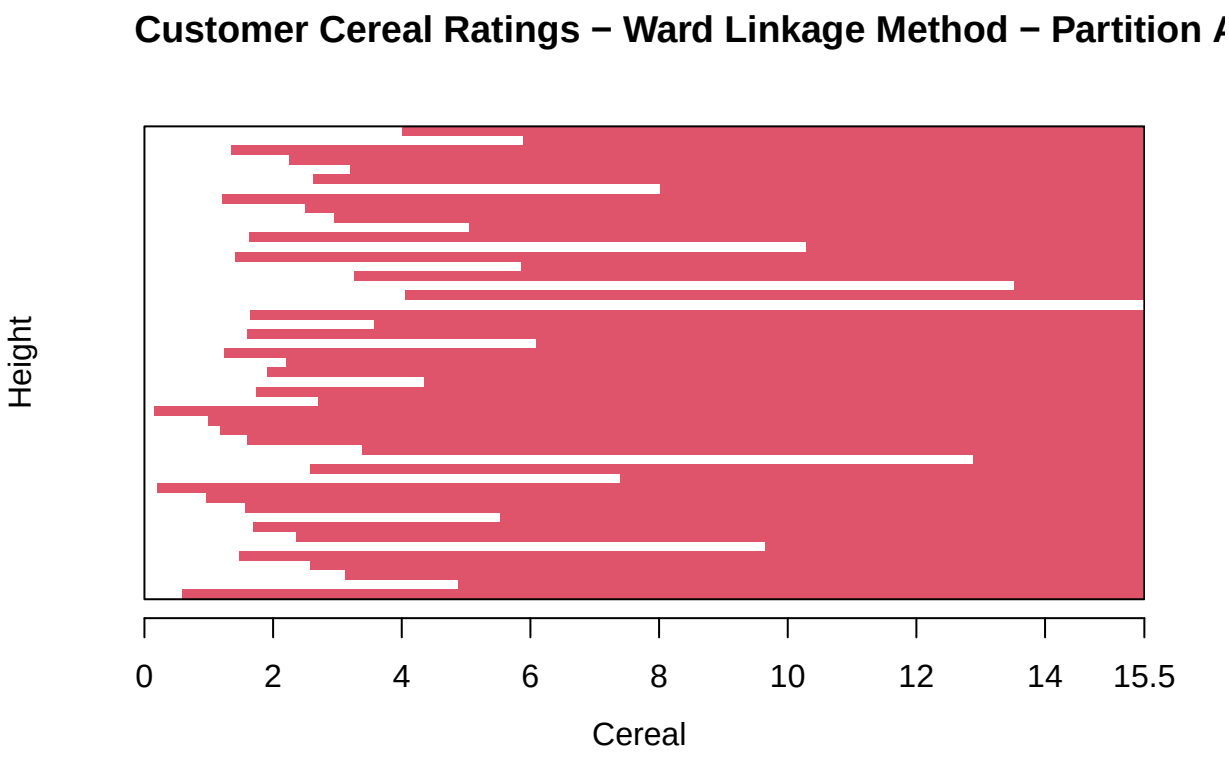
For the purposes of this task, we will assume the same K value (12) and ward clustering method to determine the stability of the clusters. We will then assign clusters to the nearest points in Partition B (for clusters 1 to 12).

```r
# Create the dissimilarity matrix for the numeric values in the partitioned data set via Euclidean dist
Assignment5_d_euclidean_A <- dist(Assignment5_preprocessed_PartitionA[ , c(4:16)], method = "euclidean")
# Perform hierarchical clustering via the ward linkage method on partitioned data
ag_hc_ward_A <- agnes(Assignment5_d_euclidean_A, method = "ward")
# Plot the results of the different methods
plot(ag_hc_ward_A,
     main = "Customer Cereal Ratings - Ward Linkage Method - Partition A",
     xlab = "Cereal",
     ylab = "Height",
     cex.axis = 1,
     cex = 0.55,
     hang = -1)
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): "hang" is not a graphical
## parameter
```

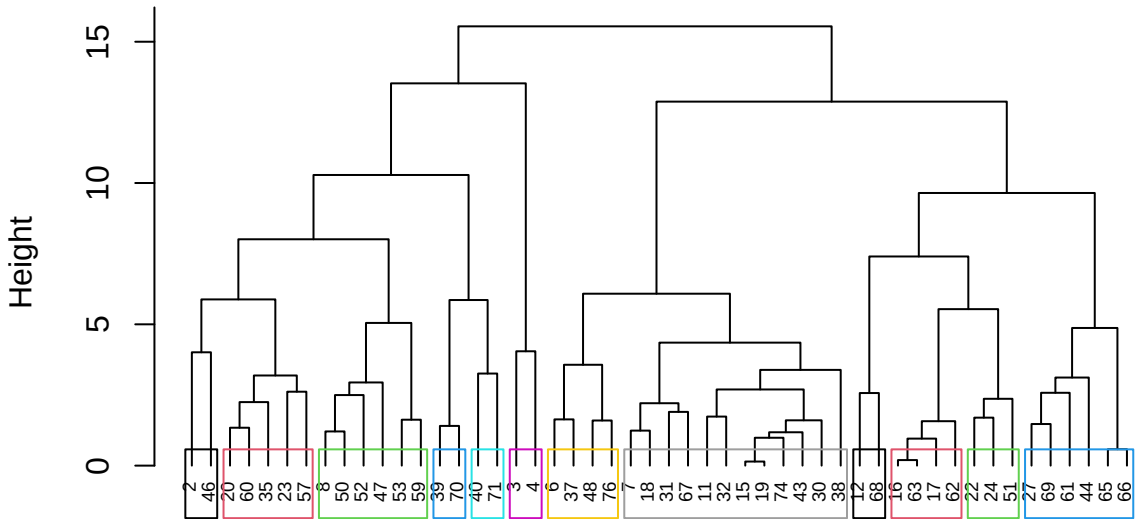
```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "hang"
## is not a graphical parameter
```

```
## Warning in axis(1, at = at.vals, labels = lab.vals, ...): "hang" is not a
## graphical parameter
```



Agglomerative Coefficient = 0.88

```
rect.hclust(ag_hc_ward_A, k = 12, border = 1:12)
```




```
# Cut the tree into 12 clusters for analysis
ward_clusters_12_A <- cutree(ag_hc_ward_A, k = 12)
# Add the assigned cluster to the preprocessed data set
Assignment5_preprocessed_A <- cbind(cluster = ward_clusters_12_A, Assignment5_preprocessed_PartitionA)
```

The centroids for each of the clusters will need to be calculated, so we can find the closest centroid for the data points in partition B.

```
# Find the centroids for the re-ran Ward hierarchical clustering
ward_Centroids_A <- aggregate(Assignment5_preprocessed_A[, 5:17], list(Assignment5_preprocessed_A$cluste
ward_Centroids_A <- data.frame(Assignment5 = ward_Centroids_A[, 1], Centroid = rowMeans(ward_Centroids_A
ward_Centroids_A <- ward_Centroids_A$Centroid
summary(ward_Centroids_A)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.250040 -0.141338  0.008143  0.145452  0.396052  0.935190
```

```
# Calculate Centers of Partition B data set
Assignment5_preprocessed_PartitionB_centers <- data.frame(Assignment5_preprocessed_PartitionB[, 1:3], Ce
summary(Assignment5_preprocessed_PartitionB_centers)
```

```
##      name               mfr                type               Center
##  Length:24          Length:24          Length:24          Min.   :-1.01345
##  Class :character   Class :character   Class :character   1st Qu.:-0.19341
```

```
##  Mode   :character   Mode   :character   Mode   :character   Median :-0.04209
##                                                              Mean   :-0.05216
##                                                              3rd Qu.: 0.15576
##                                                              Max.   : 0.63939
```

```
# Calculate the distance between the centers of partition A and the values of partition B
B_to_A_centers <- dist(ward_Centroids_A, Assignment5_preprocessed_PartitionB_centers$Center, method = "
# Assign the clusters based on the minimum distance to cluster centers
Assignment5_preprocessed_B <- cbind(cluster = c(4,8,7,3,5,6,7,11,11,10,8,5,10,1,10,1,4,12,12,7,7,1,4,9)
# Combine partitions A and B for comparision to original clusters
Assignment5_preprocessed_2 <- rbind(Assignment5_preprocessed_A, Assignment5_preprocessed_B)
Assignment5_preprocessed_1 <- Assignment5_preprocessed_1[order(Assignment5_preprocessed_1$name), ]
Assignment5_preprocessed_2 <- Assignment5_preprocessed_2[order(Assignment5_preprocessed_2$name), ]
```

Now that the data has been assigned by both methods (full data and partitioned data), we can compare the number of matching assignments to see the stability of the clusters.

```
sum(Assignment5_preprocessed_1$cluster == Assignment5_preprocessed_2$cluster)
```

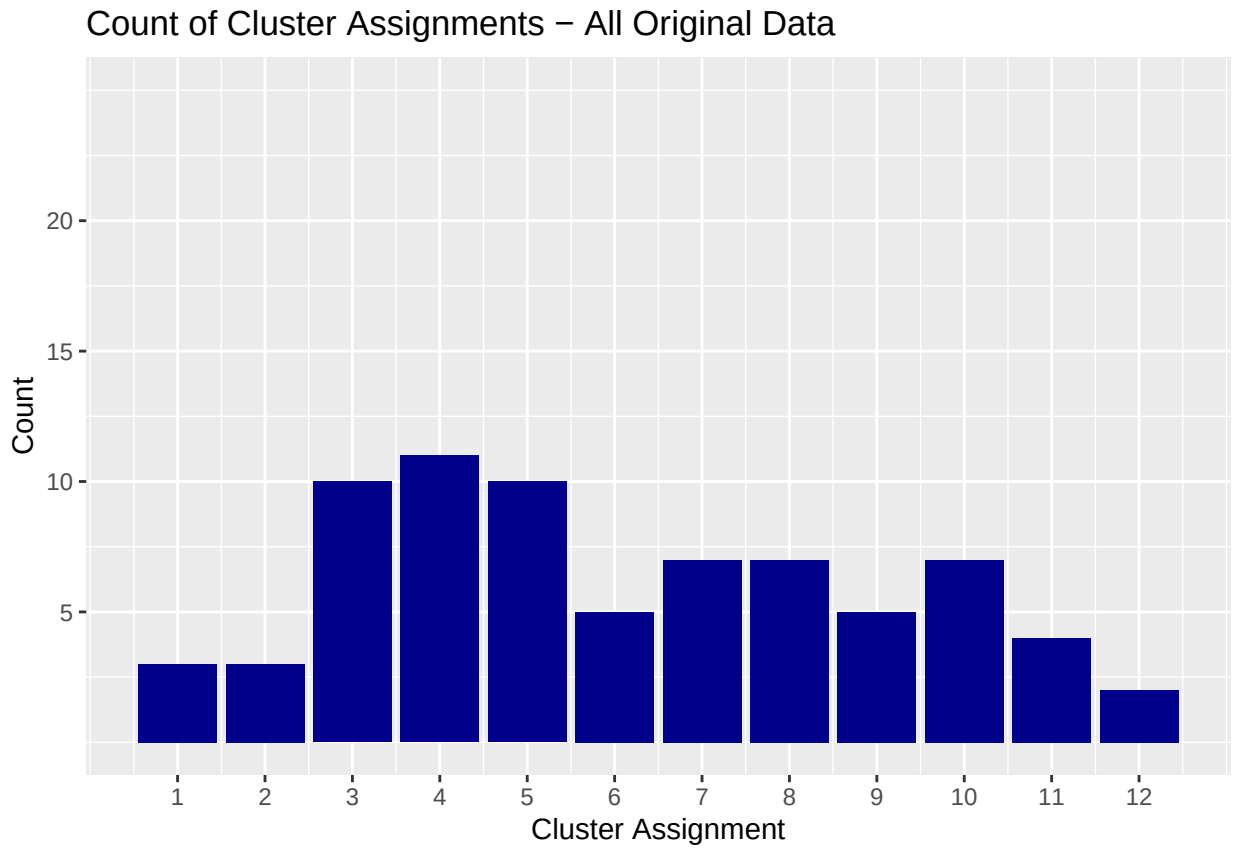
```
## [1] 38
```

From this result, it can be stated that the clusters are not very stable. With 70% of the data available, the resulting assignments were only identical for 38 out of the 74 observations. This results in a % repeatability of assignment.

```
# Visualize the cluster assignments to see any difference between the two
# Plot of original hierarchical clustering algorithm
ggplot(data = Assignment5_preprocessed_1, aes(Assignment5_preprocessed_1$cluster)) +
  geom_bar(fill = "blue4") +
  labs(title="Count of Cluster Assignments - All Original Data") +
  labs(x="Cluster Assignment", y="Count") +
  guides(fill=FALSE) +
  scale_x_continuous(breaks=c(1:12)) +
  scale_y_continuous(breaks=c(5,10,15,20), limits = c(0,25))
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

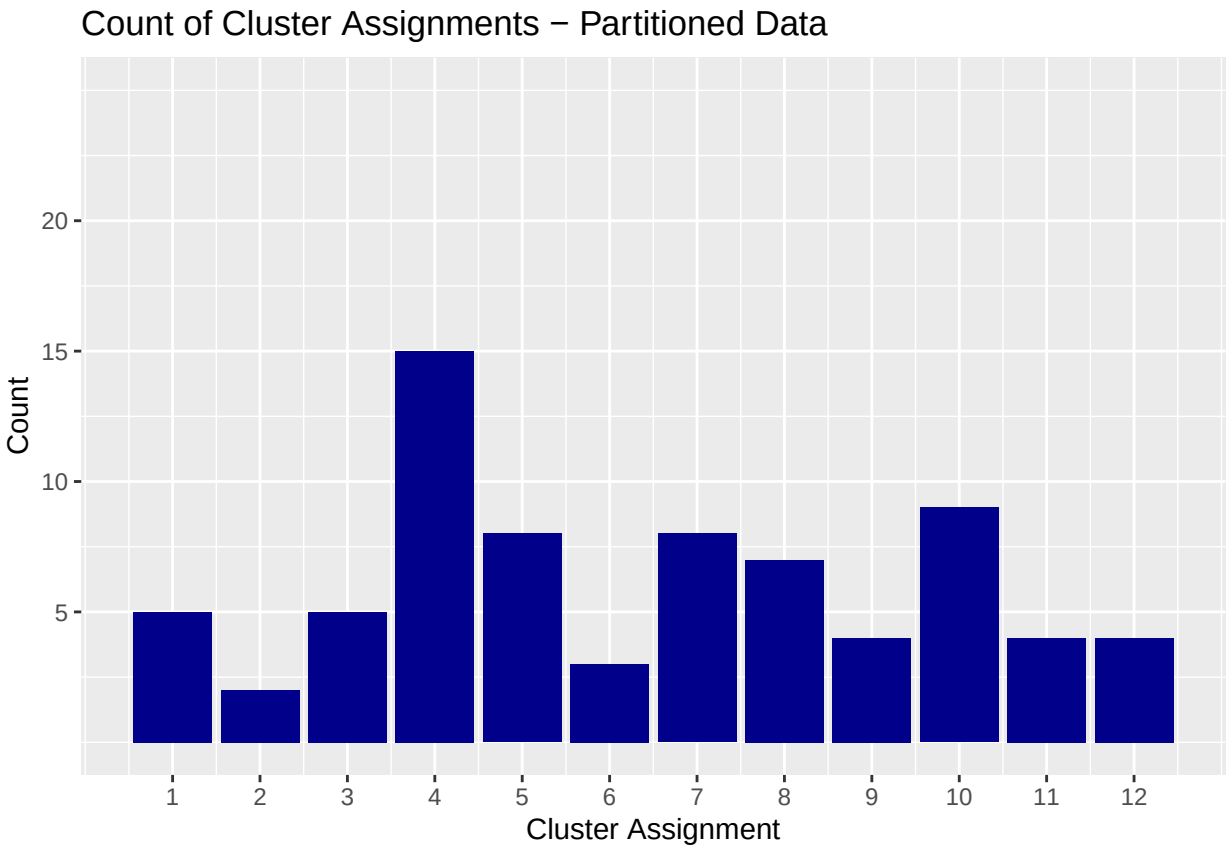
```
## Warning: Use of `Assignment5_preprocessed_1$cluster` is discouraged. Use
## `cluster` instead.
```




```
# Plot of algorithm that was partitioned prior to assigning the remaining data
ggplot(data = Assignment5_preprocessed_2, aes(Assignment5_preprocessed_2$cluster)) +
  geom_bar(fill = "blue4") +
  labs(title="Count of Cluster Assignments - Partitioned Data") +
  labs(x="Cluster Assignment", y="Count") +
  guides(fill=FALSE) +
  scale_x_continuous(breaks=c(1:12)) +
  scale_y_continuous(breaks=c(5,10,15,20), limits = c(0,25))
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

```
## Warning: Use of `Assignment5_preprocessed_2$cluster` is discouraged. Use
## `cluster` instead.
```



Visually, we can see that Cluster 3 significantly shrunk when using the partitioned data. As a result, several of the other clusters became larger as a result. From the chart, it appears the clusters are more evenly distributed across the 12 clusters when the data is partitioned.

Assignment Task D

The answer could be found in the summary, please.