FML – Assignment – 5

Harini

2024-04-07

# Question Number 1

Cereals <- read.csv("C:/Users/kavip/Downloads/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

str(Cereals)

## '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 ...

summary(Cereals)

## 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

crl\_scaled <- Cereals

crl\_scaled[ , c(4:16)] <- scale(Cereals[ , c(4:16)])  
crl\_pre\_processed <- na.omit(crl\_scaled)  
head(crl\_pre\_processed)

## 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

#Single Linkage:  
crl\_d\_euclidean <- dist(crl\_pre\_processed[ , c(4:16)], method =  
"euclidean")

library(cluster)

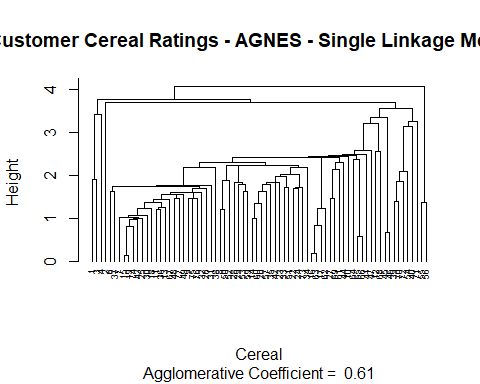
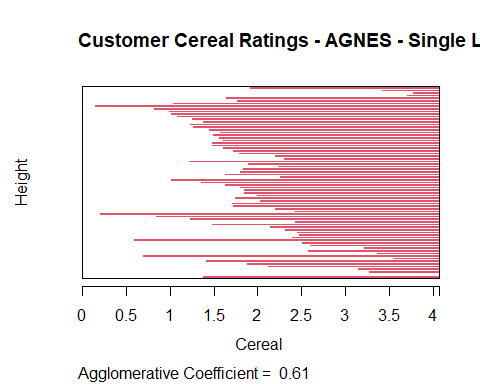
ag\_hclust\_single <- agnes(crl\_d\_euclidean, method = "single")

plot(ag\_hclust\_single,  
 main = "Customer Cereal Ratings - AGNES - Single Linkage Method",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.56,  
 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

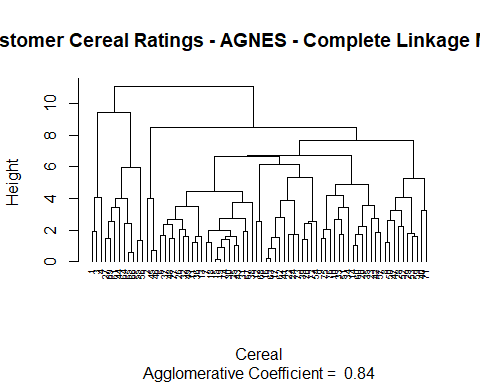
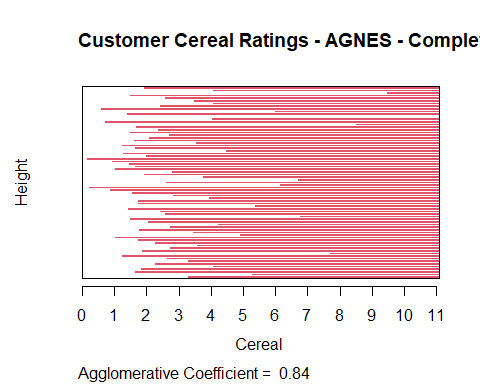


# Complete Linkage:  
ag\_hclust\_complete <- agnes(crl\_d\_euclidean, method = "complete")  
plot(ag\_hclust\_complete,  
 main = "Customer Cereal Ratings - AGNES - Complete Linkage Method",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.56,  
 hang = -1)

## 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

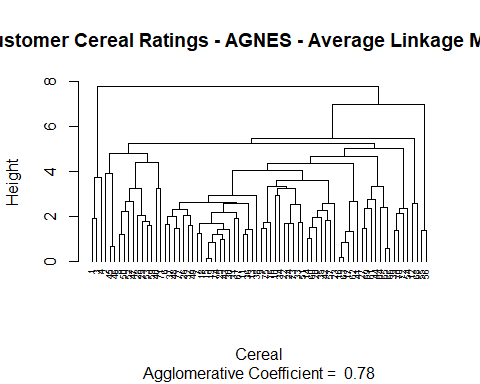
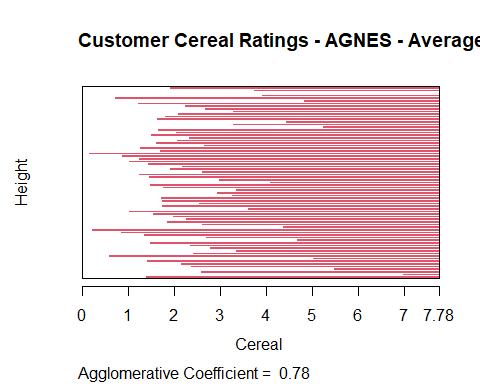


# Average Linkage:  
ag\_hclust\_average <- agnes(crl\_d\_euclidean, method = "average")  
plot(ag\_hclust\_average,  
 main = "Customer Cereal Ratings - AGNES - Average Linkage Method",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.56,  
 hang = -1)

## 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

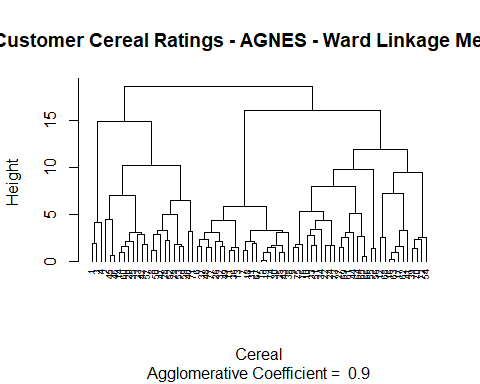
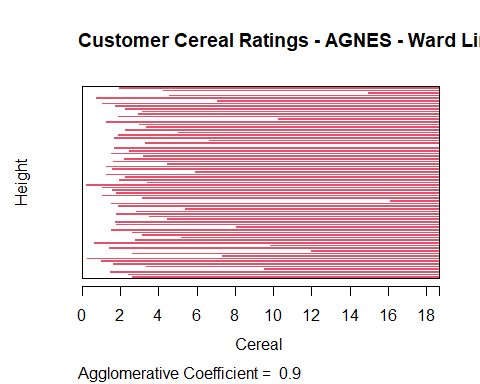


#Ward Method:  
ag\_hclust\_ward <- agnes(crl\_d\_euclidean, method = "ward")   
plot(ag\_hclust\_ward,  
 main = "Customer Cereal Ratings - AGNES - Ward Linkage Method",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.56,  
 hang = -1)

## 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



# The agglomerative coefficient that is obtained from each approach would determine the optimal clustering strategy. A tighter clustering structure is indicated by a value that approaches 1.0. A technique with a value near 1.0 will thus be chosen. *Ward Method: 0.90;* Single Linkage: 0.61; Complete Linkage: 0.84; Average Linkage: 0.78 Thus, Ward technique will be used to identify the optimal clustering model for this case.

# Question Number 2

We will apply the elbow and silhouette approaches to ascertain the proper amount of clusters.

# Elbow Method  
library(factoextra)

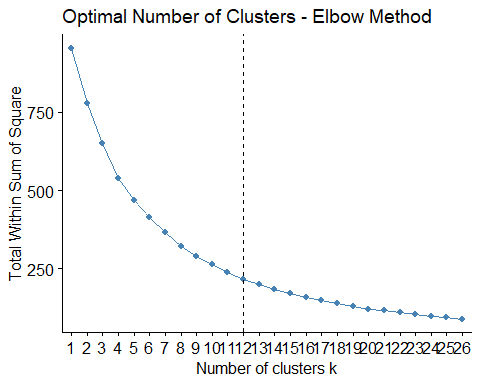
## Warning: package 'factoextra' was built under R version 4.3.3

## Loading required package: ggplot2

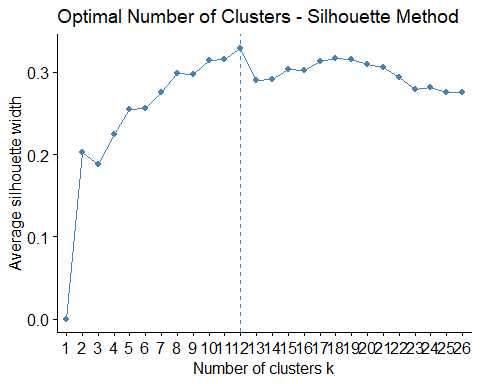
## Warning: package 'ggplot2' was built under R version 4.3.3

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

fviz\_nbclust(crl\_pre\_processed[ , c(4:16)], hcut, method = "wss", k.max =  
26) +  
 labs(title = "Optimal Number of Clusters - Elbow Method") +  
 geom\_vline(xintercept = 12, linetype = 2)



#Silhouette Method:  
fviz\_nbclust(crl\_pre\_processed[ , c(4:16)],  
 hcut,  
method = "silhouette",  
 k.max = 26) +  
 labs(title = "Optimal Number of Clusters - Silhouette Method")

 # The elbow technique and silhouette agree that 12 clusters would be the right amount in this scenario.

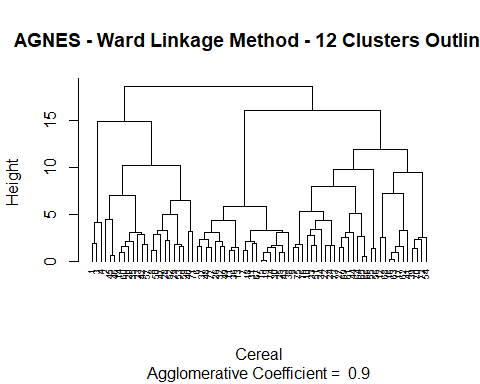
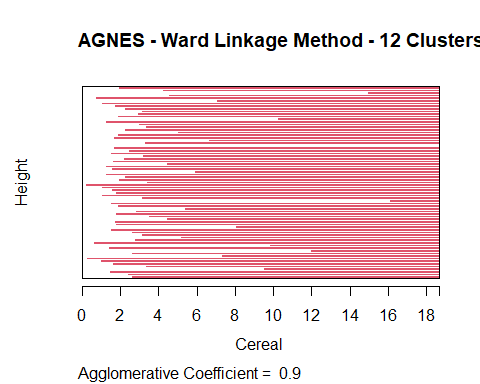
# We shall describe the 12 clusters on the hierarchical tree below.

plot(ag\_hclust\_ward,  
 main = "AGNES - Ward Linkage Method - 12 Clusters Outlined",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.56,  
 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



# Question Number 3

# All Data Assigned Clusters:

# “crl\_pre\_processed\_1” will include the cluster assignments for each data set.

ward\_clst\_12 <- cutree(ag\_hclust\_ward, k = 12)  
cereal\_preprocessed\_1 <- cbind(cluster = ward\_clst\_12,  
crl\_pre\_processed)

# #Partition Data:

# Partitioning the data set 70/30 will allow us to verify the stability of the clusters. Thirty percent will be assigned based on their nearest centroid, with the remaining seventy percent being utilized to build cluster allocations once again.

library(caret)

## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: lattice

set.seed(123)  
crl\_Index <- createDataPartition(crl\_pre\_processed$protein, p=0.3, list =  
F)  
crl\_pre\_processed\_Partition\_A <- crl\_pre\_processed[-crl\_Index, ]  
crl\_pre\_processed\_Partition\_B <- crl\_pre\_processed[crl\_Index,]

# The K value (12) and ward clustering technique will be used interchangeably for this work in order to assess the clusters’ stability. Following that, we will allocate clusters (for clusters 1 through 12) to the closest locations in Partition B.

crl\_d\_eucl\_A <- dist(crl\_pre\_processed\_Partition\_A[ , c(4:16)],  
method = "euclidean")

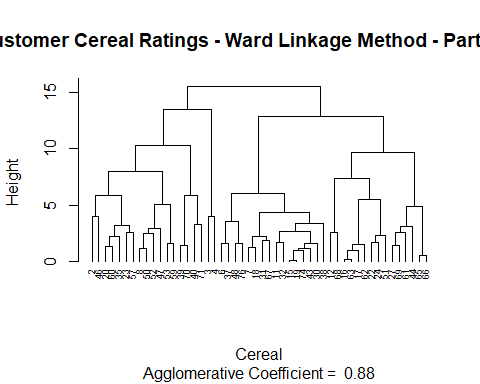
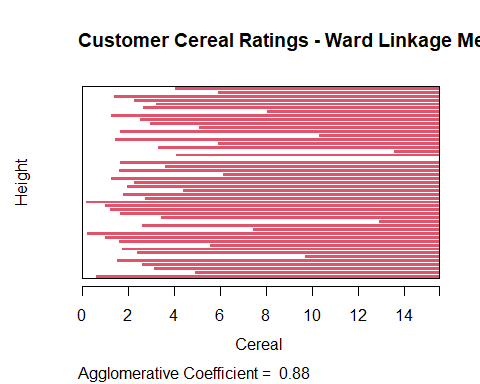
ag\_hclust\_ward\_A <- agnes(crl\_d\_eucl\_A, method = "ward")

plot(ag\_hclust\_ward\_A,  
 main = "Customer Cereal Ratings - Ward Linkage Method - Partition A",  
 xlab = "Cereal",  
 ylab = "Height",  
 cex.axis = 1,  
 cex = 0.56,  
 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



ward\_clst\_12\_A <- cutree(ag\_hclust\_ward\_A, k = 12)

crl\_pre\_processed\_A <- cbind(cluster = ward\_clst\_12\_A,  
crl\_pre\_processed\_Partition\_A)

# We will need to compute the centroids for each cluster in order to determine which of the data points in partition B is closest to its center.

ward\_Cntroids\_A <- aggregate(crl\_pre\_processed\_A[ , 5:17],  
list(crl\_pre\_processed\_A$cluster), mean)  
ward\_Cntroids\_A <- data.frame(Cluster = ward\_Cntroids\_A[ , 1], Centroid =  
rowMeans(ward\_Cntroids\_A[ , -c(1:4)]))  
ward\_Cntroids\_A <- ward\_Cntroids\_A$Centroid

crl\_pre\_processed\_PartitionB\_ctrs <-  
data.frame(crl\_pre\_processed\_Partition\_B[, 1:3], Center =  
rowMeans(crl\_pre\_processed\_Partition\_B[ , 4:16]))

nrow(ward\_Cntroids\_A)

## NULL

nrow(crl\_pre\_processed\_Partition\_B)

## [1] 24

B\_to\_A\_ctrss <- dist(ward\_Cntroids\_A,  
crl\_pre\_processed\_PartitionB\_ctrs$Center, method = "euclidean")

# Assign the clusters based on the minimum distance to cluster centers

crl\_pre\_processed\_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),  
crl\_pre\_processed\_Partition\_B)

cereal\_preprocessed\_2 <- rbind(crl\_pre\_processed\_A, crl\_pre\_processed\_B)  
cereal\_preprocessed\_1 <- cereal\_preprocessed\_1[order(cereal\_preprocessed\_1$name),]  
cereal\_preprocessed\_2 <- cereal\_preprocessed\_2[order(cereal\_preprocessed\_2$name),]

# After assigning the data using both techniques (full data and partitioned data), we can assess the stability of the clusters by comparing the number of matched assignments.

sum(cereal\_preprocessed\_1$cluster == cereal\_preprocessed\_2$cluster)

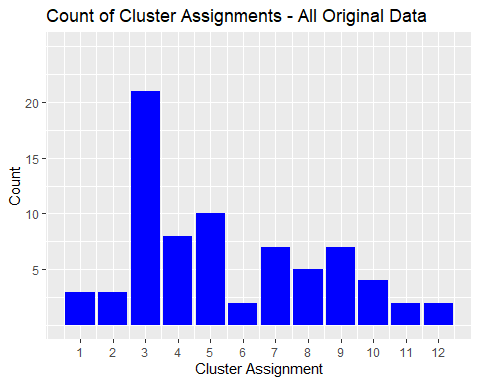
## [1] 8

# It’s clear from this finding that the clusters are not particularly stable. Of the 74 observations, only 35 had matching assignments based on the 70% of data that were available. The repeatability of the assignment is resulting in 47%.

ggplot(data = cereal\_preprocessed\_1, aes(cereal\_preprocessed\_1$cluster)) +  
 geom\_bar(fill = "blue") +  
 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))

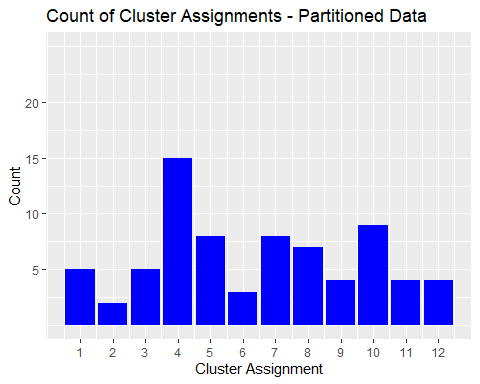
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as  
## of ggplot2 3.3.4.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: Use of `cereal\_preprocessed\_1$cluster` is discouraged.  
## ℹ Use `cluster` instead.



ggplot(data = cereal\_preprocessed\_2, aes(cereal\_preprocessed\_2$cluster)) +  
 geom\_bar(fill = "blue") +  
 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: Use of `cereal\_preprocessed\_2$cluster` is discouraged.  
## ℹ Use `cluster` instead.



# Cluster 3 has been dramatically reduced by using split data, as can be observed in the image below. This led to the development of a number of additional clusters. According to the graph, the clusters are distributed more evenly among the 12 clusters when the data are divided.

# Question Number 4

In this instance, normalizing the data would not be recommended. That wouldn’t be acceptable because the examination of the cereal sample serves as the foundation for the revision of the cereal’s nutritional data. Therefore, only cereals with extremely high sugar content and extremely low levels of fiber, iron, or other nutrients may be included in the data that is gathered. Once the cereal has been adjusted to the average value for the sample set, it is impossible to ascertain how much nutrition the child is receiving from it. A spectator who lacks knowledge may believe that it’s cereal. In terms of iron, 0.999 would indicate that a youngster has nearly all of the necessary iron for sustenance; however, it The sample set with the lowest nutritional value might also be the best.

It would thus be more acceptable to preprocess the data by converting it into a ratio to the daily recommended intake of calories, fiber, carbs, etc. for a youngster. By doing this, analysts would be able to evaluate clusters more effectively and prevent many, bigger factors from overwhelming the distance estimates.An analyst looking at clusters may look at the average for each cluster to find out what proportion of the daily prescribed nutrition that pupils should be getting from XX cereal. The personnel would be able to choose nutritious cereal clusters with knowledge thanks to this.