Assignment\_5

2025-06-23

# Load necessary libraries  
if (!require("cluster")) install.packages("cluster", dependencies = TRUE)

## Loading required package: cluster

## Warning: package 'cluster' was built under R version 4.4.3

if (!require("dplyr")) install.packages("dplyr", dependencies = TRUE)

## Loading required package: dplyr

## Warning: package 'dplyr' was built under R version 4.4.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

if (!require("readr")) install.packages("readr", dependencies = TRUE)

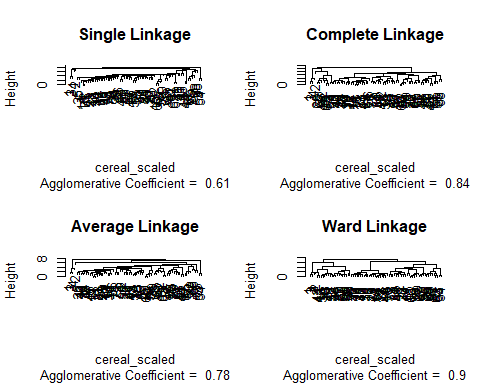
## Loading required package: readr

library(cluster)  
library(dplyr)  
library(readr)  
  
# Load the dataset  
cereal\_df <- read\_csv("C:/Users/arkha/Downloads/Cereals.csv")

## Rows: 77 Columns: 16

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (3): name, mfr, type  
## dbl (13): calories, protein, fat, sodium, fiber, carbo, sugars, potass, vita...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Answer: All cereals with missing values are removed using na.omit().  
cereal\_df\_clean <- na.omit(cereal\_df)  
  
# Drop non-numeric columns (name, mfr, type)  
cereal\_numeric <- cereal\_df\_clean %>% select(-name, -mfr, -type)  
  
# Normalize the numeric data  
# Answer: Yes, normalization is necessary because the variables (e.g., sodium, fiber, etc.)  
# are on different scales. Without normalization, large-scale features would dominate the distance calculations.  
cereal\_scaled <- scale(cereal\_numeric)  
  
# Apply AGNES with different linkage methods  
agnes\_single <- agnes(cereal\_scaled, method = "single")  
agnes\_complete <- agnes(cereal\_scaled, method = "complete")  
agnes\_average <- agnes(cereal\_scaled, method = "average")  
agnes\_ward <- agnes(cereal\_scaled, method = "ward")  
  
# Plot dendrograms for comparison  
par(mfrow = c(2, 2))  
plot(agnes\_single, which.plots = 2, main = "Single Linkage")  
plot(agnes\_complete, which.plots = 2, main = "Complete Linkage")  
plot(agnes\_average, which.plots = 2, main = "Average Linkage")  
plot(agnes\_ward, which.plots = 2, main = "Ward Linkage")



# Based on the dendrograms, Ward’s method was chosen as the best clustering approach.It creates well-separated and balanced clusters.  
  
  
# I decided to cut the tree into 4 clusters based on visual observation of Ward's dendrogram.  
cluster\_labels <- cutree(agnes\_ward, k = 4)  
  
# Attach cluster info to the data  
cereal\_df\_clean$cluster\_id <- cluster\_labels  
  
# View cluster sizes  
table(cereal\_df\_clean$cluster\_id)

##   
## 1 2 3 4   
## 3 20 21 30

# Split data into training (70%) and testing (30%)  
set.seed(101)  
n\_rows <- nrow(cereal\_scaled)  
train\_idx <- sample(1:n\_rows, size = 0.7 \* n\_rows)  
  
data\_train <- cereal\_scaled[train\_idx, ]  
data\_test <- cereal\_scaled[-train\_idx, ]  
  
# Cluster partition A (training)  
agnes\_train <- agnes(data\_train, method = "ward")  
train\_clusters <- cutree(agnes\_train, k = 4)  
  
# Calculate centroids of training clusters  
train\_centroids <- aggregate(data\_train, by = list(cluster = train\_clusters), FUN = mean)  
  
# Function to assign test observations to nearest centroid  
assign\_to\_cluster <- function(obs, centroids) {  
 distances <- apply(centroids[, -1], 1, function(center) sum((obs - center)^2))  
 return(which.min(distances))  
}  
  
# Assign test observations to nearest cluster  
test\_assignments <- apply(data\_test, 1, assign\_to\_cluster, centroids = train\_centroids)  
  
# Cluster full data to compare assignments  
agnes\_full <- agnes(cereal\_scaled, method = "ward")  
full\_clusters <- cutree(agnes\_full, k = 4)  
true\_test\_clusters <- full\_clusters[-train\_idx]  
  
# Compare predicted vs. actual cluster labels in test set  
# This comparison checks how consistent cluster assignments from training are with the full dataset.  
# The table shows the overlap between predicted clusters (from training centroids) and actual clusters (from full data).  
table(Predicted = test\_assignments, Actual = true\_test\_clusters)

## Actual  
## Predicted 1 2 3 4  
## 1 0 0 0 6  
## 2 2 2 0 1  
## 3 0 0 5 0  
## 4 0 7 0 0

# Summarize clusters to inspect their nutritional profile  
cluster\_summary <- cereal\_df\_clean %>%  
 group\_by(cluster\_id) %>%  
 summarise(  
 avg\_calories = mean(calories),  
 avg\_protein = mean(protein),  
 avg\_fat = mean(fat),  
 avg\_sugars = mean(sugars),  
 avg\_fiber = mean(fiber)  
 )  
  
print(cluster\_summary)

## # A tibble: 4 × 6  
## cluster\_id avg\_calories avg\_protein avg\_fat avg\_sugars avg\_fiber  
## <int> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 63.3 4 0.667 3.67 11   
## 2 2 124 3.15 1.95 9.35 3.1   
## 3 3 111. 1.52 1 11.3 0.571  
## 4 4 97.3 2.63 0.4 3.03 1.8

# The healthiest cereals would have high fiber and protein and low sugar and fat.  
# By inspecting the summary, I can identify the cluster that best fits these criteria.  
# This cluster would be recommended for elementary schools offering healthy breakfast options.