

# Assignment\_5

2025-11-11

## Loading packages

```
library(readr)
library(dplyr)

## 
## 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

library(class)
library(gmodels)
library(tidyverse)

## — Attaching core tidyverse packages ————— tidyverse
2.0.0 —
## ✓forcats    1.0.1      ✓stringr    1.5.2
## ✓ggplot2    4.0.0      ✓tibble      3.3.0
## ✓lubridate  1.9.4      ✓tidyrm    1.3.1
## ✓purrr     1.1.0

## — Conflicts —————
tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()   masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(factoextra)

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

library(flexclust)
library(stats)
library(cluster)
library(caret)

## Loading required package: lattice
## 
## Attaching package: 'lattice'
```

```

## 
## The following objects are masked from 'package:flexclust':
## 
##     barchart, bwplot, densityplot, histogram
## 
## 
## Attaching package: 'caret'
## 
## The following object is masked from 'package:purrr':
## 
##     lift

```

## Importing the dataset

```

cereals <- read_csv("./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.

summary(cereals)      #77 observations

##      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 procesing

```

cereals <- na.omit(cereals) #removing the observation with null values
summary(cereals)           #3 observations have been removed, now the total
is 74

##      name            mfr            type            calories
##  Length:74          Length:74          Length:74          Min.   : 50
##  Class :character   Class :character   Class :character  1st Qu.:100
##  Mode  :character   Mode  :character   Mode  :character  Median  :110
##                                         Mean   :107
##                                         3rd Qu.:110
##                                         Max.   :160
##      protein          fat            sodium          fiber            carbo
##  Min.    :1.000   Min.    :0       Min.    :  0.0  Min.    : 0.000   Min.   :
##  1st Qu.:2.000   1st Qu.:0       1st Qu.:135.0  1st Qu.: 0.250   1st
##  Qu.:12.00
##  Median  :2.500   Median  :1       Median  :180.0  Median  : 2.000   Median
##  :14.50
##  Mean    :2.514   Mean    :1       Mean    :162.4  Mean    : 2.176   Mean
##  :14.73
##  3rd Qu.:3.000   3rd Qu.:1       3rd Qu.:217.5  3rd Qu.: 3.000   3rd
##  Qu.:17.00
##  Max.    :6.000   Max.    :5       Max.    :320.0  Max.    :14.000   Max.
##  :23.00
##      sugars          potass          vitamins        shelf
##  Min.    : 0.000   Min.    :15.00   Min.    :  0.00  Min.    :1.000
##  1st Qu.: 3.000   1st Qu.:41.25   1st Qu.: 25.00  1st Qu.:1.250
##  Median  : 7.000   Median  :90.00   Median  : 25.00  Median  :2.000
##  Mean    : 7.108   Mean    :98.51   Mean    : 29.05  Mean    :2.216
##  3rd Qu.:11.000   3rd Qu.:120.00  3rd Qu.: 25.00  3rd Qu.:3.000
##  Max.    :15.000   Max.    :330.00  Max.    :100.00 Max.    :3.000
##      weight          cups           rating
##  Min.    :0.500   Min.    :0.2500   Min.    :18.04
##  1st Qu.:1.000   1st Qu.:0.6700   1st Qu.:32.45
##  Median  :1.000   Median  :0.7500   Median  :40.25
##  Mean    :1.031   Mean    :0.8216   Mean    :42.37

```

```

## 3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:50.52
## Max.     :1.500   Max.      :1.5000   Max.     :93.70

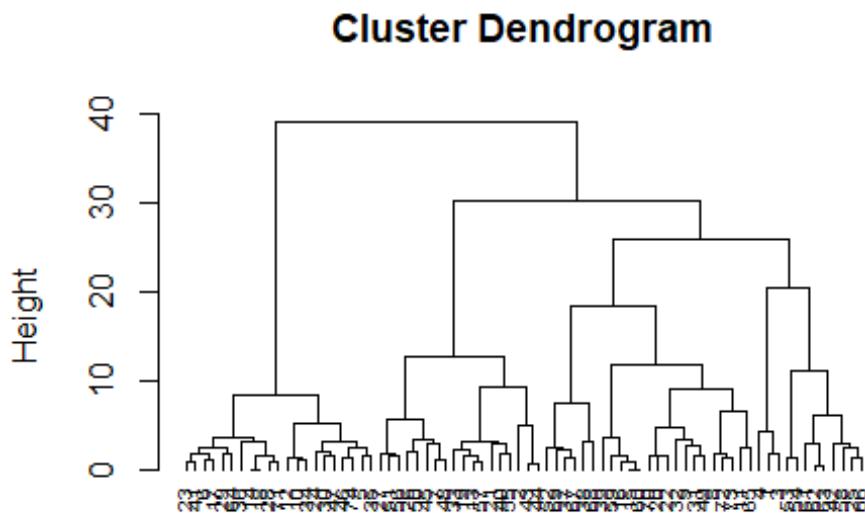
#from the summary, it is clear that the value of sodium, potassium will influence the model due to its high magnitude nature, if not normalized

cereals.num <- cereals[, 4:16] #creating dataset only with numeric variables

```

## Task 1: Hierarchical clustering and Agnes

```
#Applying hierarchical clustering to the data using Euclidean distance to the  
normalized measurements.  
df <- scale(cereals.num)      #normalizing data  
dist <- dist(df, method="euclidean")    #calculating euclidean distance matrix  
  
hc.ward <- hclust(dist, method="ward")  
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"  
  
plot(hc.ward, cex=0.6, hang=-1)
```



```
dist  
hclust(*, "ward.D")
```

```
#Agnes clustering with different linkage methods, using the normalized data
hc_single <- agnes(df, method="single")
hc_complete <- agnes(df, method="complete")
hc_average <- agnes(df, method="average")
hc_ward <- cluster::agnes(df, method="ward")
#comparing agglomerative coefficients
print(hc_single$ac)
```

```

## [1] 0.6067859
print(hc_complete$ac)
## [1] 0.8353712
print(hc_average$ac)
## [1] 0.7766075
print(hc_ward$ac)
## [1] 0.9046042

#Even if I use the euclidean distance data with agnes instead of the only normalized data, it shows the same result.
hc_ward1 <- agnes(dist, method="ward")
print(hc_ward1$ac)

## [1] 0.9046042

#From agglomerative coefficient values, ward linkage is the best method for this dataset, as it shows the highest value, which is 0.9046042, among all four methods.

```

## Task 2: Choosing the cluster number

```

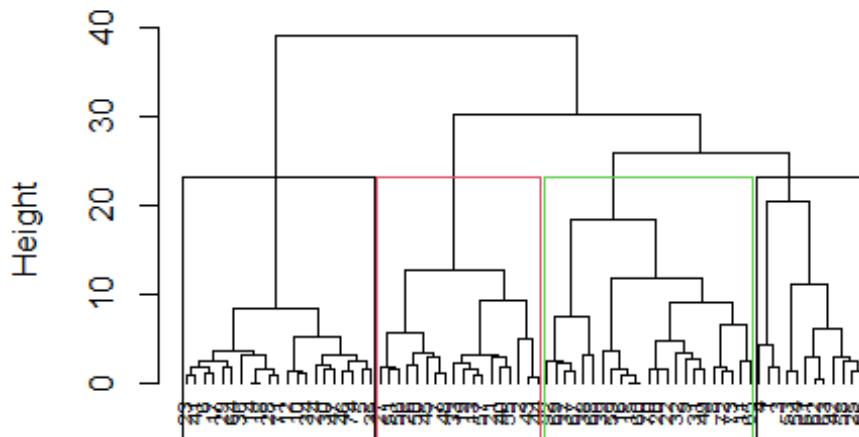
#From the lecture slides, we learned that agnes ( ) and hclust ( ) functions behave very similarly, so I am using ward linkage here to find the cluster number based on euclidean distance matrix
hc.ward <- hclust(dist, method="ward")

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

plot(hc.ward, cex=0.6, hang=-1)
#visualization of clusters
rect.hclust(hc.ward, k=4, border=1:3)

```

## Cluster Dendrogram



```
dist  
hclust (*, "ward.D")
```

```
#adding the cluster number with the original dataset  
df <- as.data.frame(df)  
df.clusters <- cutree(hc.ward, k = 4)  
df$cluster <- df.clusters  
table(df.clusters)  
  
## df.clusters  
## 1 2 3 4  
## 12 18 21 23  
  
#From the tree structure, 3 or 4 clusters would be a good separation of data.  
If cut the tree into 5 or more clusters then the chances of getting small  
groups are high. Also, then I will need to jump a larger vertical gaps.  
Finally, between those choice, I will proceed with cluster 4.
```

### Task 3: Stability checking

```
set.seed(246)  
# Step 1: Partitioning the data (e.g., 70% training, 30% test)  
train.index <- createDataPartition(1:nrow(df), p = 0.7, list = FALSE)  
train.data=df[train.index,]      #70%  
test.data=df[-train.index,]     #30%  
  
#clustering partition A (training data)  
train.data.dist <- dist(train.data, method="euclidean")  
hc.clusterA <- hclust(train.data.dist, method="ward") #making cluster  
partition A using Ward Linkage
```

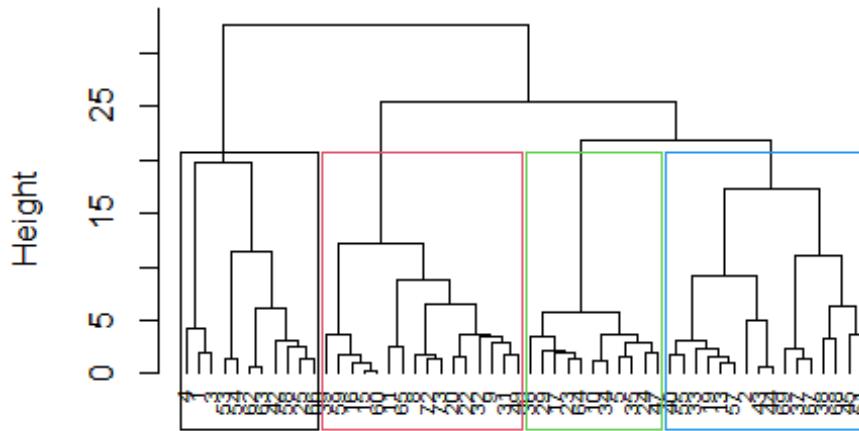
```

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

#doing this part for my visualization
plot(hc.clusterA, cex=0.6, hang=-1)
rect.hclust(hc.clusterA,k=4,border=1:4)

```

## Cluster Dendrogram



```

train.data.dist
hclust (*, "ward.D")

```

```

#calculating training data cluster centroids
train_clusters <- cutree(hc.clusterA, k = 4)
train.data <- as.data.frame(train.data)
train.data$cluster <- train_clusters
centroid.A <- aggregate(train.data, by = list(Cluster = train_clusters),
mean)

#Assigning each record in B (test data) to nearest centroid
assign_to_cluster <- function(obs, centers) {
  distances <- apply(centers, 1, function(c) sqrt(sum((obs - c)^2)))
  which.min(distances)
}
test_prediction <- apply(test.data, 1, assign_to_cluster, centers =
centroid.A)

## Warning in obs - c: longer object length is not a multiple of shorter
## object
## length

```

```

#getting actual clusters of test dataset
fulldata.cluster <- hclust(dist, method="ward")

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

clusters <- cutree(as.hclust(fulldata.cluster), k = 4)
test_clusters <- clusters[-train.index]

#Assess stability using Adjusted Rand Index (ARI)
ari <- randIndex(test_prediction, test_clusters)
cat("Adjusted Rand Index (stability measure):", ari, "\n")

## Adjusted Rand Index (stability measure): 0.4808743

#The ARI value is 0.4808743 which represents somewhat stable relation among
clusters

```

## Task 4: Healthy diet cereal clustering

*#To make the cluster of healthy diets, I am using the hierarchical clustering and ward linkage method. The code will be almost same as task-1*

```

diet <- cereals
diet <- na.omit(diet) #removing the observation with null values
summary(diet)          #from the summary, it is clear that the value of
potassium will influence the model due to its high magnitude nature, if not
normalized

##      name           mfr        type       calories
##  Length:74      Length:74      Length:74      Min.   : 50
##  Class :character Class :character Class :character 1st Qu.:100
##  Mode  :character Mode  :character Mode  :character Median  :110
##                                         Mean   :107
##                                         3rd Qu.:110
##                                         Max.   :160
##      protein         fat        sodium      fiber       carbo
##  Min.   :1.000   Min.   :0   Min.   : 0.0   Min.   : 0.000   Min.   :
##  1st Qu.:2.000   1st Qu.:0   1st Qu.:135.0  1st Qu.: 0.250   1st
##  Qu.:12.00
##  Median :2.500   Median :1   Median :180.0  Median : 2.000   Median
##  :14.50
##  Mean   :2.514   Mean   :1   Mean   :162.4  Mean   : 2.176   Mean
##  :14.73
##  3rd Qu.:3.000   3rd Qu.:1   3rd Qu.:217.5  3rd Qu.: 3.000   3rd
##  Qu.:17.00
##  Max.   :6.000   Max.   :5   Max.   :320.0  Max.   :14.000   Max.
##  :23.00
##      sugars        potass      vitamins      shelf
##  Min.   : 0.000   Min.   :15.00   Min.   : 0.00   Min.   :1.000
##  1st Qu.: 3.000   1st Qu.:41.25   1st Qu.:25.00  1st Qu.:1.250
##  Median : 7.000   Median : 90.00   Median :25.00  Median :2.000

```

```

##   Mean    : 7.108    Mean    : 98.51    Mean    : 29.05    Mean    : 2.216
## 3rd Qu.:11.000    3rd Qu.:120.00    3rd Qu.: 25.00    3rd Qu.: 3.000
## Max.    :15.000    Max.    :330.00    Max.    :100.00    Max.    :3.000
##      weight       cups        rating
## Min.    :0.500    Min.    :0.2500    Min.    :18.04
## 1st Qu.:1.000    1st Qu.:0.6700    1st Qu.:32.45
## Median :1.000    Median :0.7500    Median :40.25
## Mean    :1.031    Mean    :0.8216    Mean    :42.37
## 3rd Qu.:1.000    3rd Qu.:1.0000    3rd Qu.:50.52
## Max.    :1.500    Max.    :1.5000    Max.    :93.70

diet.num <- diet[, 4:16] #removing the non-numeric variables

diet <- scale(diet.num) #normalizing data
summary(diet)          # now all variables are in same scale

##      calories       protein        fat        sodium
## Min.    :-2.8738    Min.    :-1.40687    Min.    :-0.9932    Min.    :-1.9616
## 1st Qu.:-0.3541    1st Qu.:-0.47733    1st Qu.:-0.9932    1st Qu.:-0.3306
## Median : 0.1498    Median :-0.01256    Median : 0.0000    Median : 0.2131
## Mean    : 0.0000    Mean    : 0.00000    Mean    : 0.0000    Mean    : 0.0000
## 3rd Qu.: 0.1498    3rd Qu.: 0.45221    3rd Qu.: 0.0000    3rd Qu.: 0.6661
## Max.    : 2.6695    Max.    : 3.24083    Max.    : 3.9729    Max.    : 1.9045
##      fiber         carbo        sugars        potass
## Min.    :-0.89778   Min.    :-2.50014   Min.    :-1.6306   Min.    :-1.1783
## 1st Qu.:-0.79462   1st Qu.:-0.70143   1st Qu.:-0.9424   1st Qu.:-0.8079
## Median :-0.07249   Median :-0.05903   Median :-0.0248   Median :-0.1201
## Mean    : 0.00000   Mean    : 0.00000   Mean    : 0.0000   Mean    : 0.0000
## 3rd Qu.: 0.34015   3rd Qu.: 0.58337   3rd Qu.: 0.8928   3rd Qu.: 0.3031
## Max.    : 4.87925   Max.    : 2.12512   Max.    : 1.8104   Max.    : 3.2660
##      vitamins       shelf        weight       cups
## Min.    :-1.3032    Min.    :-1.4617    Min.    :-3.4600    Min.    :-2.4251
## 1st Qu.:-0.1818    1st Qu.:-1.1612    1st Qu.:-0.2008    1st Qu.:-0.6432
## Median :-0.1818    Median :-0.2599    Median :-0.2008    Median :-0.3038
## Mean    : 0.0000    Mean    : 0.0000    Mean    : 0.0000   Mean    : 0.0000
## 3rd Qu.:-0.1818    3rd Qu.: 0.9420    3rd Qu.:-0.2008    3rd Qu.: 0.7568
## Max.    : 3.1822    Max.    : 0.9420    Max.    : 3.0583    Max.    : 2.8780
##      rating
## Min.    :-1.7336
## 1st Qu.:-0.7071
## Median :-0.1510
## Mean    : 0.0000
## 3rd Qu.: 0.5807
## Max.    : 3.6578

dist <- dist(diet, method="euclidean") #calculating euclidean distance matrix

#performing hierarchical clustering with ward linkage as it's agglomerative

```

```

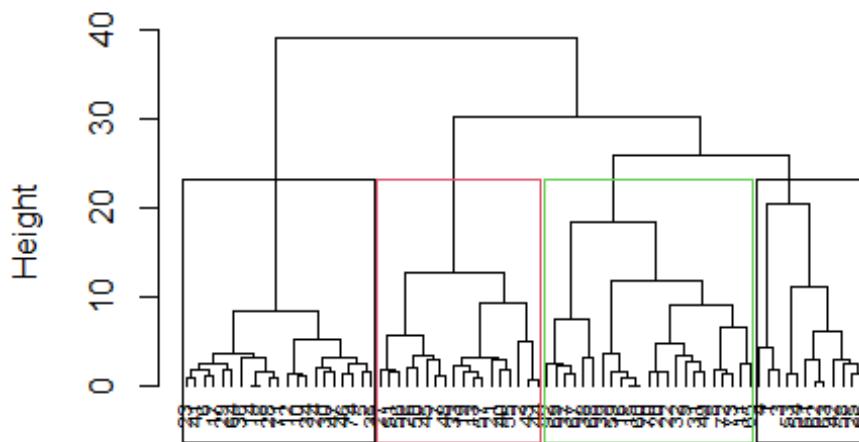
coefficient value is high (got from task-1)
hc.ward <- hclust(dist, method="ward")

## The "ward" method has been renamed to "ward.D"; note new "ward.D2"

plot(hc.ward, cex=0.6, hang=-1)
rect.hclust(hc.ward,k=4,border=1:3)

```

## Cluster Dendrogram



```

dist
hclust (*, "ward.D")

```

```

#adding the cluster number with the scaled dataset
diet <- as.data.frame(df)
diet.clusters <- cutree(hc.ward, k = 4)
diet$cluster <- diet.clusters
#some cluster summary
table(df.clusters)

## df.clusters
## 1 2 3 4
## 12 18 21 23

cereal.diet <- aggregate(diet, by = list(diet$cluster), FUN = mean)

#I will use the protein, fiber, vitamins, fat and potassium variables to
interpret the clusters to select the best set of cereals
cereal.diet %>% group_by(cluster) %>% select(calories, protein, fiber,
vitamins, fat, potass, sodium, sugars, carbo)

## Adding missing grouping variables: `cluster`

```

```

## # A tibble: 4 × 10
## # Groups:   cluster [4]
##   cluster calories protein fiber vitamins     fat potass sodium sugars
carbo
##   <dbl>      <dbl>    <dbl>  <dbl>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
<dbl>
## 1      1 -1.49     0.297  0.890   -0.649 -0.745  0.662 -1.41  -1.02  -
0.402
## 2      2  0.766    0.607  0.386   -0.244  1.05   0.730 -0.123  0.459  -
0.302
## 3      3  0.198   -0.920 -0.662   -0.182  0       -0.742  0.121  0.958  -
0.542
## 4      4 -0.00355  0.210  -0.162    0.696 -0.432 -0.240  0.723 -0.703
0.941

```

*#The following conclusions were obtained based on the averages of each cluster:*

*#Cluster 1 = High Protein, High Fiber, High Potassium*

*#Cluster 2 = High Calories, High Protein, High Fiber, High Fat, High Potassium, High Sugars*

*#Cluster 3 = High Calories, No Fat, High Potassium, High Carbohydrates*

*#Cluster 4 = High Protein, High Vitamins, High Sodium, High Carbohydrates*

*#The choice of health cereals will depend on what nutrition we want to provide the children. For example, without no doubt high protein, high vitamins, high fiber cereals are good for children, at the same time carbohydrate, calories, sodium, fat are good for children too but to a certain extent, for which I need expert help.*

*#Finally, I am in support of normalizing datasets. Otherwise Variables with larger numerical ranges (e.g., sodium, potassium) would influence the distance calculations.*