Assignment 5

2023-11-13

Install required packages

```
library(tidyverse)  # data manipulation
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# install.packages("factoextra")
library(factoextra)
```

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

```
library(dplyr)
library(ggplot2)
library(cluster)
set.seed(123)
```

Import data set

```
library(readr)
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.
```

```
View(Cereals)
```

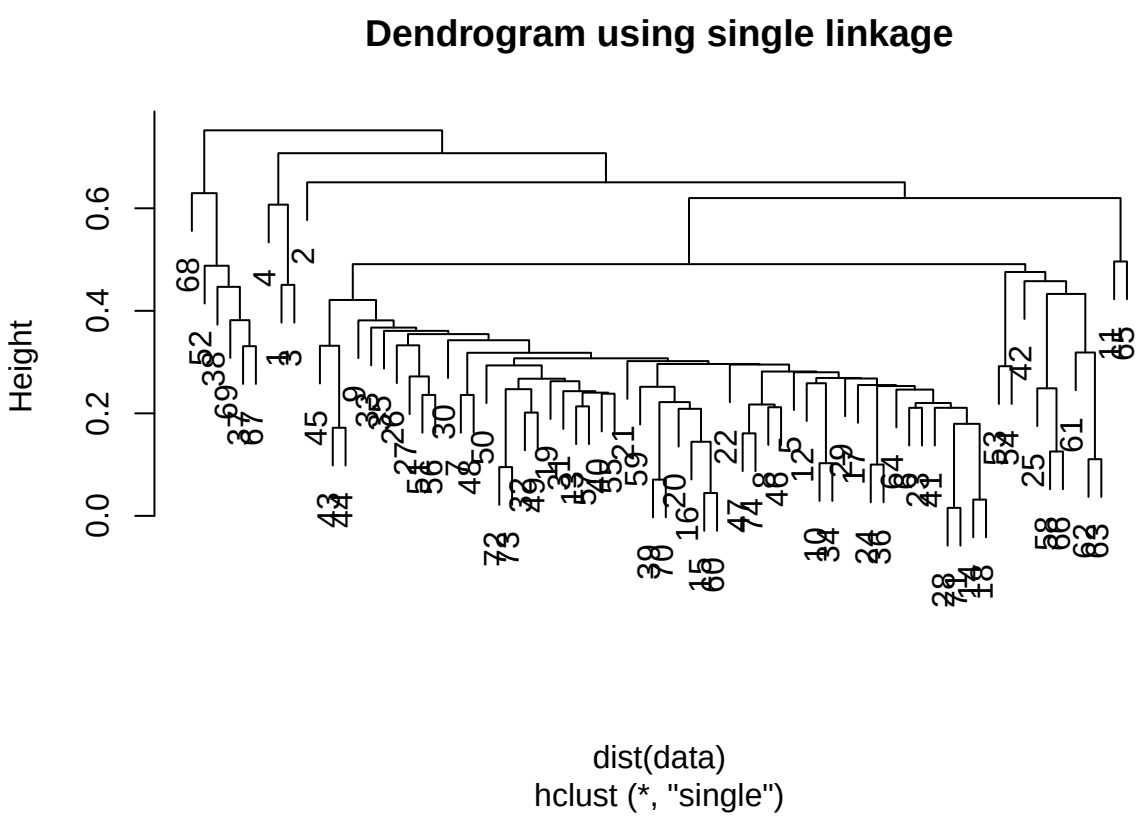
First we process the data and remove all cereals with missing values

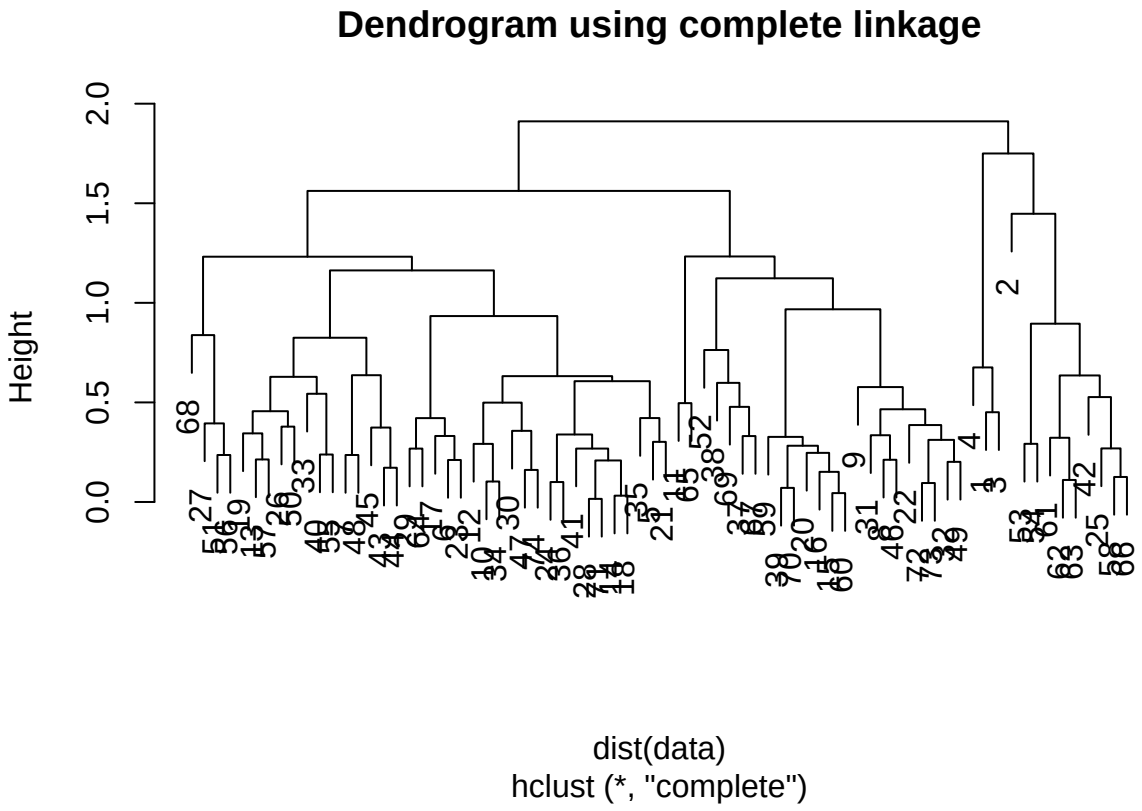
```r
# remove all missing values
cereals_data <- na.omit(Cereals)

#normalize the data
min_max_normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}

cluster_columns <- c("calories", "protein", "fat", "sodium", "fiber", "carbo", "sugars", "potass", "vita
cereals_data[cluster_columns] <- lapply(cereals_data[cluster_columns], min_max_normalize)
print(cereals_data)
```

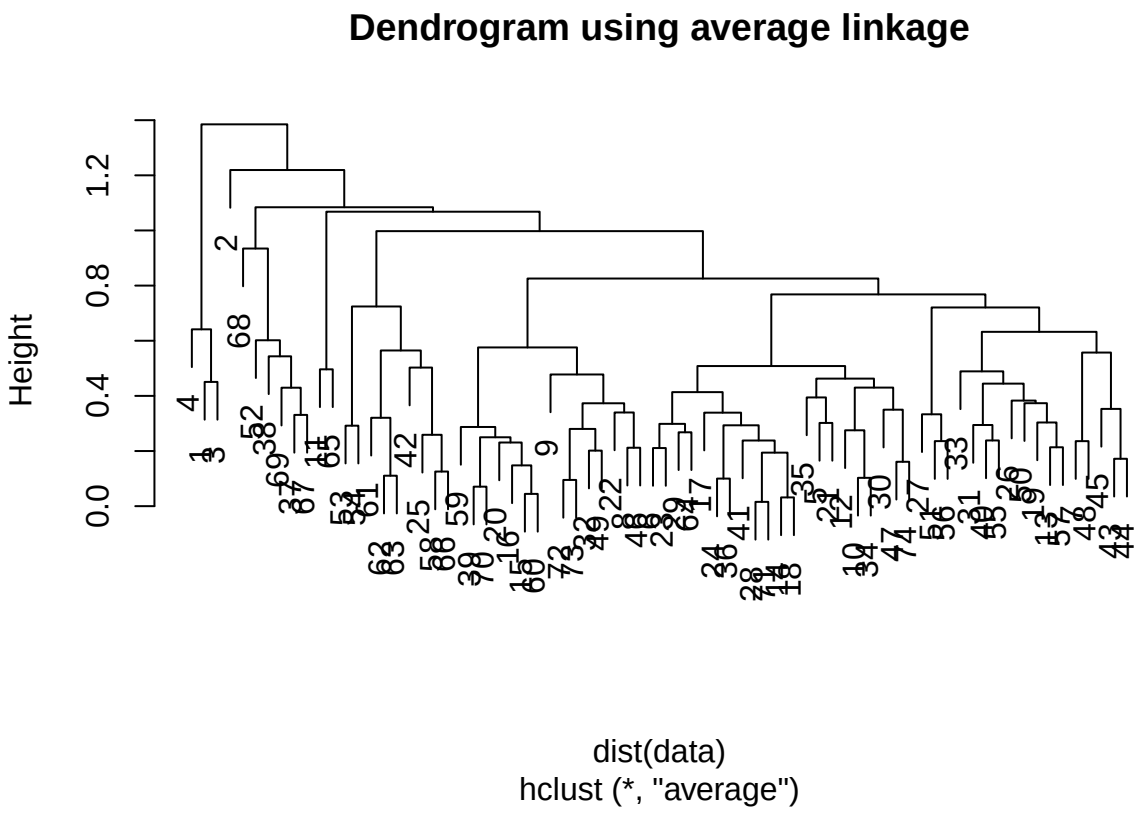
```
## # A tibble: 74 x 16
##    name     mfr   type  calories protein   fat sodium  fiber carbo sugars potass
##    <chr>    <chr> <chr>    <dbl>   <dbl> <dbl>  <dbl>  <dbl> <dbl>  <dbl>  <dbl>
##  1 100%_Br~ N     C        0.182     0.6   0.2 0.406  0.714  0      0.4   0.841
##  2 100%_Na~ Q     C        0.636     0.4   1   0.0469 0.143  0.167  0.533 0.381
##  3 All-Bran K     C        0.182     0.6   0.2 0.812  0.643  0.111  0.333 0.968
##  4 All-Bra~ K     C        0         0.6   0   0.438  1      0.167  0     1
##  5 Apple_C~ G     C        0.545     0.2   0.4 0.562  0.107  0.306  0.667 0.175
##  6 Apple_J~ K     C        0.545     0.2   0   0.391  0.0714 0.333  0.933 0.0476
##  7 Basic_4  G     C        0.727     0.4   0.4 0.656  0.143  0.722  0.533 0.270
##  8 Bran_Ch~ R     C        0.364     0.2   0.2 0.625  0.286  0.556  0.4   0.349
##  9 Bran_Fl~ P     C        0.364     0.4   0   0.656  0.357  0.444  0.333 0.556
## 10 Cap'n'C~ Q     C        0.636     0     0.4 0.688  0      0.389  0.8   0.0635
## # i 64 more rows
## # i 5 more variables: vitamins <dbl>, shelf <dbl>, weight <dbl>, cups <dbl>,
## #   rating <dbl>
```

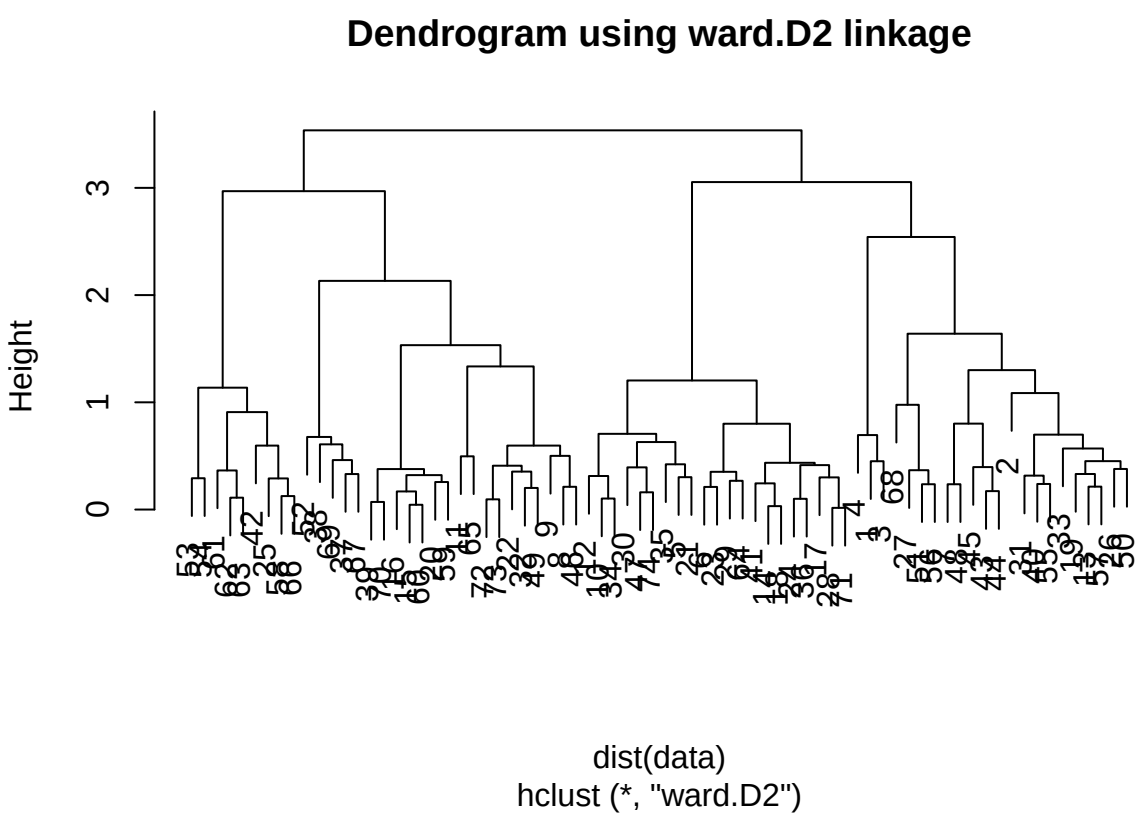
1. Apply hierarchical clustering to the data using Euclidean distance to the normalized measurements. Use Agnes to compare the clustering from single linkage, complete linkage, average linkage, and Ward. Choose the best method. How many clusters would you choose?

```r
methods <- c("single", "complete", "average", "ward.D2")

perform_clustering <- function(method, data) {
  hc <- hclust(dist(data), method = method)
  plot(hc, main = paste("Dendrogram using", method, "linkage"))
}

for (method in methods) {
  perform_clustering(method, cereals_data[cluster_columns])
}
```


Dendrogram using single linkage
Height
0.0
0.2
0.4
0.6
dist(data)
hclust (*, "single")


Dendrogram using complete linkage
Height
0.0
0.5
1.0
1.5
2.0
dist(data)
hclust (*, "complete")


Dendrogram using average linkage
Height
0.0
0.4
0.8
1.2
dist(data)
hclust (*, "average")




```
clusters_ward <- cutree(hclust(dist(cereals_data[cluster_columns]), method = "ward.D2"), k = 4)

cereals_data$Cluster_Ward <- clusters_ward
```

2. How many clusters would you chose?

```
k = 4
```

3. Comment on the structure of the clusters and on their stability.

```
train_indices <- sample(1:nrow(cereals_data), size = nrow(cereals_data) / 2)
train_data <- cereals_data[train_indices, ]
test_data <- cereals_data[-train_indices, ]

train_centroids <- aggregate(train_data[cluster_columns], by=list(Cluster=train_data$Cluster_Ward), mea

find_closest_centroid <- function(record, centroids) {
  distances <- apply(centroids[, -1], 1, function(centroid) sum((record - centroid)^2))
  return(which.min(distances))
}

test_data$Cluster_Assigned <- apply(test_data[cluster_columns], 1, function(x) find_closest_centroid(x,

original_test_clusters <- test_data$Cluster_Ward
```

```r
## Assess how consistent the cluster assignments are compared to the assignments based on all the data.

consistency <- mean(test_data$Cluster_Assigned == original_test_clusters)
consistency_percentage <- consistency * 100
print(paste("Consistency: ", consistency_percentage, "%"))
```

```
## [1] "Consistency:  94.5945945945946 %"
```

The consistency of the cluster assignments is of 94.59%. This means that 94.59% of the data in the testing sets was assigned to the same cluster when using cereals_data. This indicates high consistency and that the data set is not highly sensitive to variations in the data.

4. The elementary public schools would like to choose a set of cereals to include in their daily cafeterias. Every day a different cereal is offered, but all cereals should support a healthy diet. For this goal, you are requested to find a cluster of "healthy cereals." Should the data be normalized? If not, how should they be used in the cluster analysis?

```r
healthy_cereals <- aggregate(cereals_data[cluster_columns], by=list(Cluster=cereals_data$Cluster_Ward),
print(healthy_cereals)
```

```
##   Cluster  calories   protein        fat      sodium      fiber     carbo
## 1       1 0.5909091 0.4636364 0.36363636 0.491477273 0.30519481 0.4368687
## 2       2 0.5541126 0.1047619 0.20000000 0.526785714 0.04081633 0.4126984
## 3       3 0.5041322 0.3363636 0.10909091 0.710227273 0.11688312 0.7525253
## 4       4 0.2929293 0.2888889 0.02222222 0.005208333 0.15079365 0.5740741
##      sugars    potass  vitamins
## 1 0.5575758 0.5050505 0.2727273
## 2 0.7650794 0.1027967 0.2500000
## 3 0.2424242 0.1904762 0.4204545
## 4 0.1555556 0.2398589 0.1111111
```

The data was normlized to perform this cluster analysis. We found four clusters

```r
# Compared to the other 3 clusters, Cluster #4 is the healthiest. It's low on calories, protein, fat, s

# Print the names of cereals in this cluster.
healthy_cereals <- cereals_data[cereals_data$Cluster_Ward == 4, ]
print(healthy_cereals$name)
```

```
## [1] "Frosted_Mini-Wheats"       "Maypo"
## [3] "Puffed_Rice"               "Puffed_Wheat"
## [5] "Raisin_Squares"            "Shredded_Wheat"
## [7] "Shredded_Wheat_'n'Bran"    "Shredded_Wheat_spoon_size"
## [9] "Strawberry_Fruit_Wheats"
```