R Notebook

#Loading the libaries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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
```

```
library(cluster)
library(factoextra)
```

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

#Data Preprocessing. Remove all cereals with missing values. #Loading the data

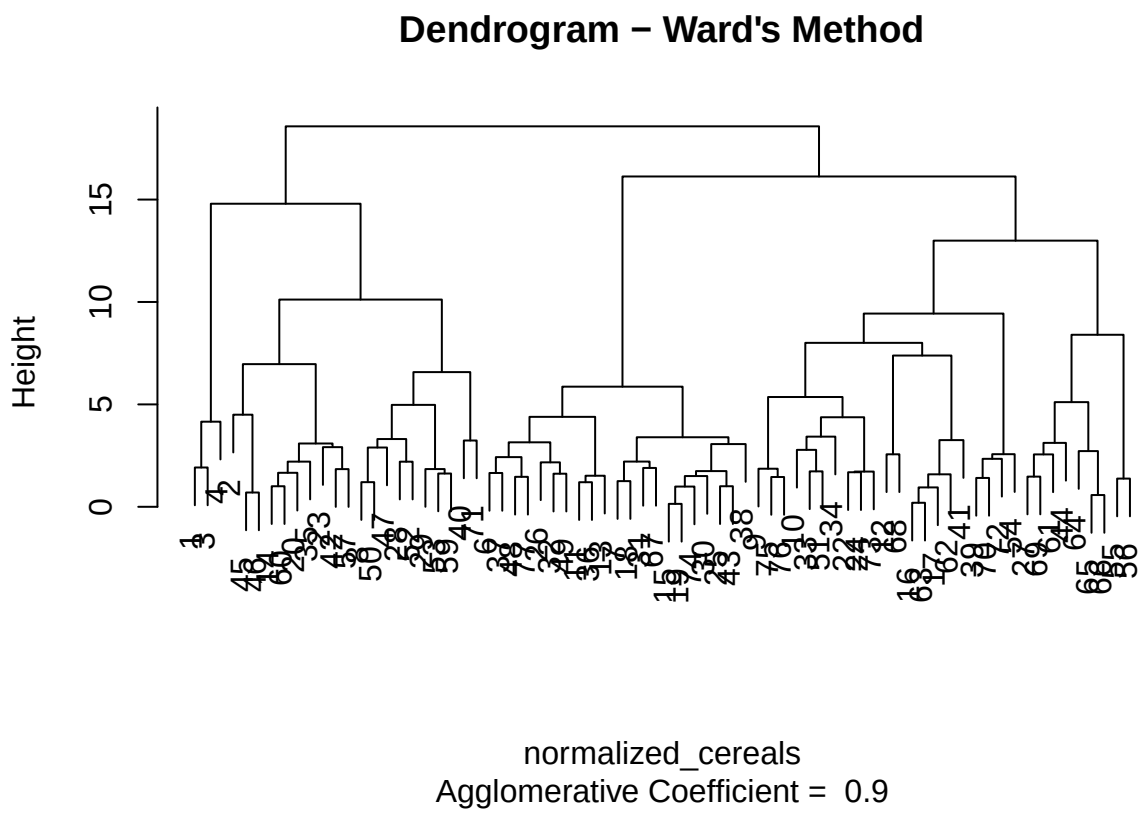
```
# Load the data
cereals <- read.csv("C:/Users/hruth/Desktop/Fundamental of machine learning/Assignment 5/Cereals.csv")

# Remove rows with missing values
cereals <- na.omit(cereals)
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
## 6   Apple_Cinnamon_Cheerios   G    C      110       2   2    180   1.5  10.5
## 7                Apple_Jacks   K    C      110       2   0    125   1.0  11.0
##   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
## 6     10     70       25     1      1 0.75 29.50954
## 7     14     30       25     2      1 1.00 33.17409
```

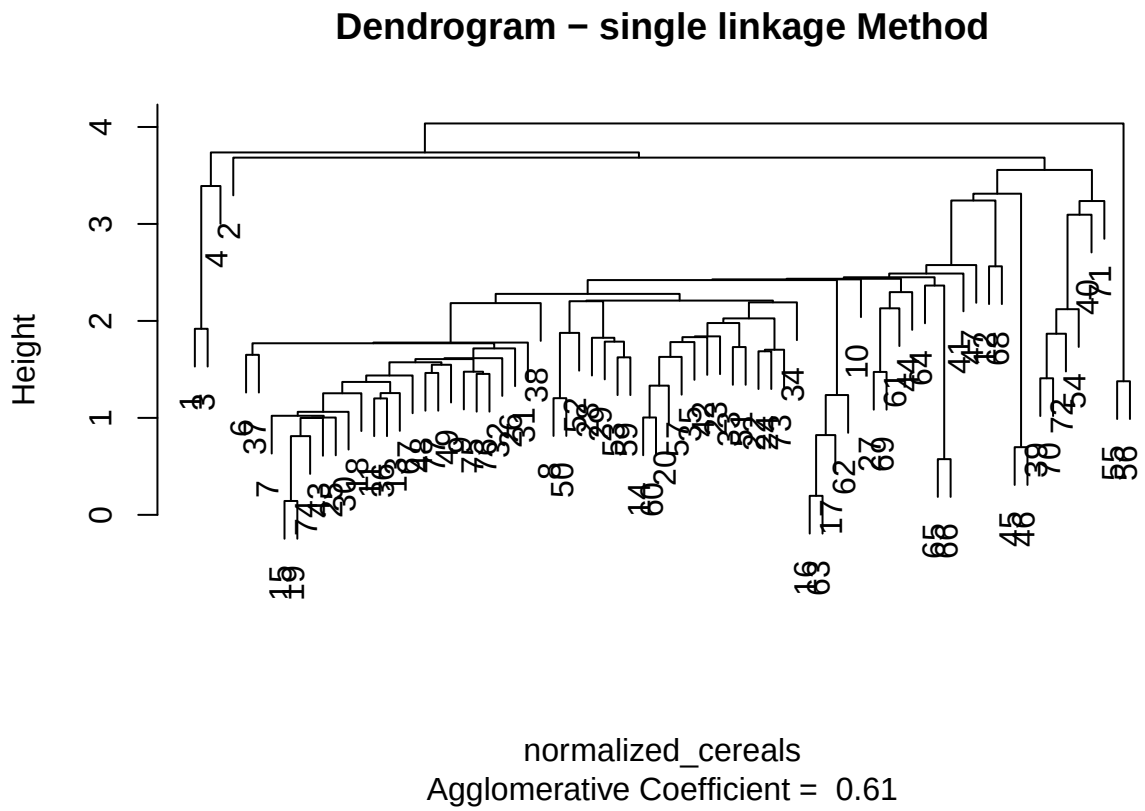
#Explanation #loads a dataset named “Cereals.csv” from the specified file path.Then,it removes rows with missing values from the cereals dataset using the na.omit() function, which deletes any rows containing NA or missing values.Finally,it displays the first few rows of the cleaned cereals dataset using the head() function to inspect the data.

#Question #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.

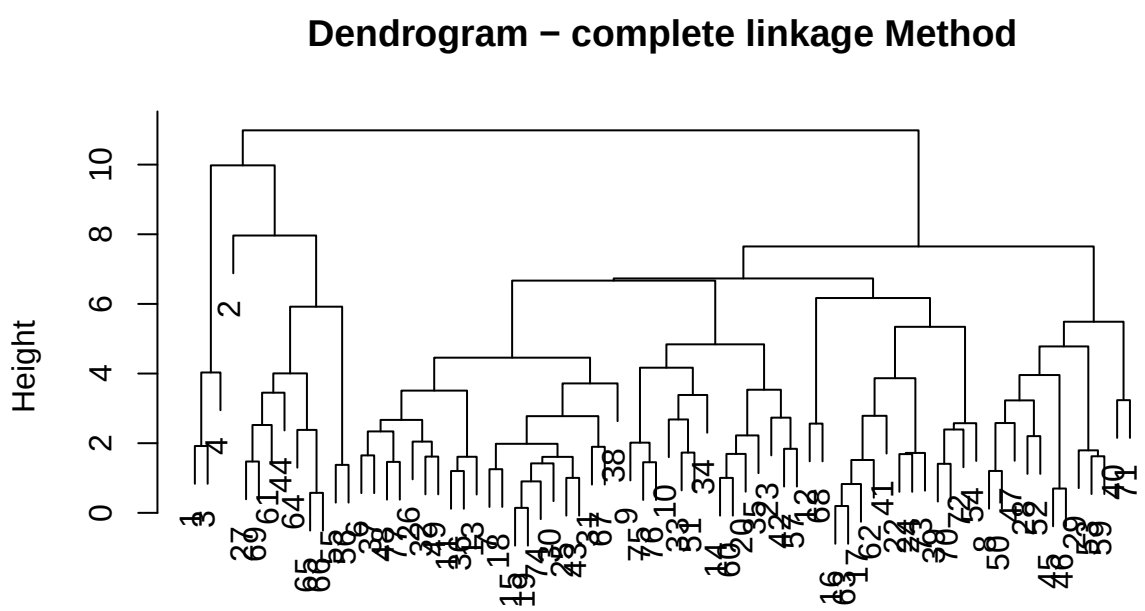
```
# Hierarchical Clustering
# Identify numeric columns
numeric_columns <- sapply(cereals, is.numeric)

# Normalize only the numeric columns
normalized_cereals <- scale(cereals[, numeric_columns])

# Apply hierarchical clustering with Ward's method
ward_cluster <- agnes(normalized_cereals, method = "ward")
# Apply hierarchical clustering with single linkage
single_clustering <- agnes(normalized_cereals, method = "single")
# Apply hierarchical clustering with complete linkage
complete_clustering <- agnes(normalized_cereals, method = "complete")
# Apply hierarchical clustering with average linkage
average_clustering <- agnes(normalized_cereals, method = "average")

# Visualize Dendrogram for Ward's method
plot(ward_cluster, which.plots = 2, main = "Dendrogram - Ward's Method")
```



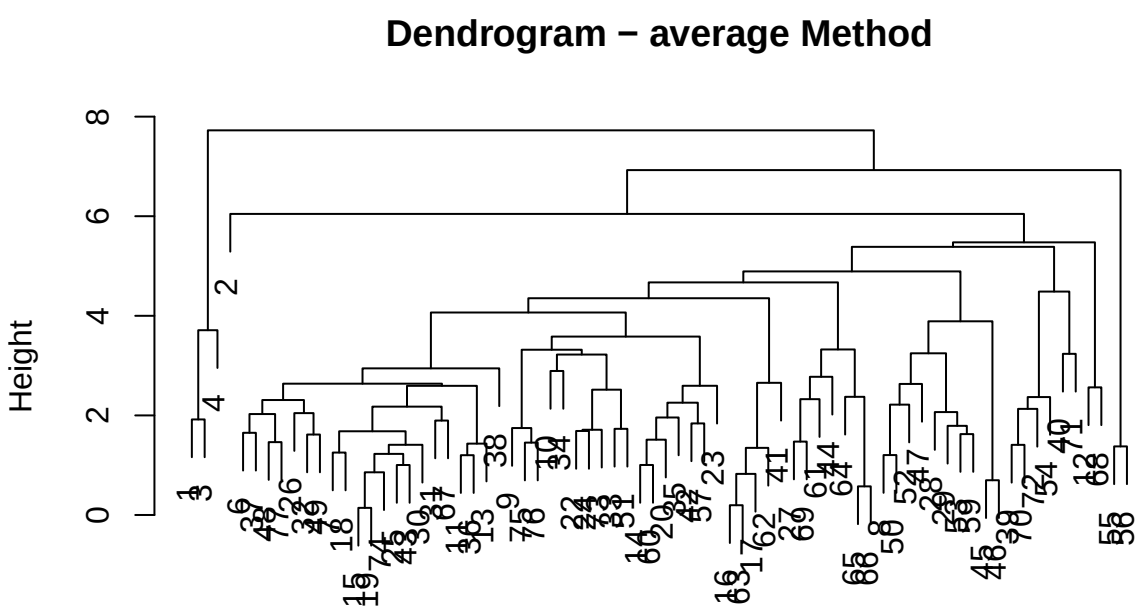
```r
# Visualize Dendrogram for Single linkage  method
plot(single_clustering, which.plots = 2, main = "Dendrogram - single linkage Method")
```




```r
# Visualize Dendrogram for Complete linkage method
plot(complete_clustering, which.plots = 2, main = "Dendrogram - complete linkage Method")
```




```
# Visualize Dendrogram for Average linkage method
plot(average_clustering, which.plots = 2, main = "Dendrogram - average Method")
```



#Explanation #This code helps in exploring hierarchical clustering using different linkage methods and visualizing the resulting dendrograms, which can assist in understanding the structure and relationships within the data.Out of all the plots the dendrogram resulting from Ward's method is good choice for visualization.

#Question #How many clusters would you choose? 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 otherpart.To do this: Cluster partition A Use the cluster centroids from A to assign each record in partition B (each recordis assigned to the cluster with the closest centroid). Assess how consistent the cluster assignments are compared to the assignments based on all the data

```
# Cluster Stability and Healthy Cereals
# Create cluster partitions A and B
set.seed(123)
partition_A <- sample(1:2, nrow(normalized_cereals), replace = TRUE)
partition_B <- 3 - partition_A

# Fit cluster on partition A
cluster_A <- cutree(ward_cluster, k = 3)

# Use cluster centroids from A to assign records in partition B
cluster_B <- cluster_A[partition_B]

# Assess cluster consistency
consistency <- sum(cluster_A == cluster_B) / length(cluster_B)

# Print or visualize the results
```

```r
cat("Cluster Consistency:", consistency, "\n")
```

```
## Cluster Consistency: 0.3108108
```

##Explanation ##This code randomly partitions the dataset into two partitions (A and B), fits a hierarchical clustering model on partition A, and then assigns the clusters from partition A to partition B to assess cluster consistency. Finally, it prints the cluster consistency value.Two cluster partitions A and B using a random sampling method, fits a cluster model on partition A, assigns records in partition B to clusters based on the centroids obtained from partition A, calculates the cluster consistency between partitions A and B.

#Question #The elementary public schools would like to choose a set of cereals to include in theirdaily cafeterias. Every day a different cereal is offered, but all cereals should support ahealthy 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
# Identify healthy cereals cluster
healthy_cereals_cluster <- cluster_A[which(cereals$sugars < 5 & cereals$fiber > 5)]

# Print or visualize the results
cat("Healthy Cereals Cluster:", healthy_cereals_cluster, "\n")
```

```
## Healthy Cereals Cluster: 1
```

#Explanation #Identifies healthy cereals by filtering the cluster assignments from partition A (cluster_A) based on specific criteria related to the cereals dataset.It selects cereals that have less than 5 grams of sugars (cereals$sugars < 5$) and more than 5 grams of fiber (cereals$fiber > 5$).The resulting cluster assignments for these healthy cereals are stored in the variable healthy_cereals_cluster. Finally, it prints the cluster assignments of the identified healthy cereals using the cat() function.