Implementing Agglomerative Hierarchical Clustering (AHC) in R from scratch

Complete one task:

2. Implementing agglomerative hierarchical clustering (AHC).

Task 2: Implement agglomerative hierarchical clustering (AHC) in R from first principles.

- You should apply your AHC program to the `NCI microarray` data set.

You need to complete the following parts:

a. Implement AHC with the following linkage functions:
    1. single linkage,
    2. complete linkage,
    3. average linkage and
    4. centroid linkage.

Your output should be a data structure that represents **a dendrogram**.

b. Implement a function `getClusters` that takes
    - a dendrogram and a positive integer K as arguments,
    - and its output is the K clusters obtained by cutting the dendrogram at an appropriate height.

c. In your report, use `the getClusters function` to discuss the performance of AHC with **the four different linkage functions** when applied to the `NCI microarray` dataset.

2.1. Introduction

What is Clustering?

Clustering is the method of dividing objects into sets that are similar, and dissimilar to the objects belonging to another set.

There are two different types of clustering, each divisible into two subsets:

1. Hierarchical clustering
    - Agglomerative
    - Divisive
2. Partial clustering
    - K-means
    - Fuzzy c-means

What is Hierarchical Clustering?

Hierarchical clustering is separating data into groups based on some measure of similarity, finding a way to measure how they are alike and different, and further narrowing down the data.

Types of Hierarchical Clustering

Hierarchical clustering is divided into:

1. Agglomerative
2. Divisive

Divisive Clustering is known as the top-down approach. We take a large cluster and start dividing it into two, three, four, or more clusters.

Agglomerative Clustering is known as a bottom-up approach. Consider it as bringing things together.

What is the Distance Measure?

- Distance measure determines the similarity between two elements and it influences the shape of the clusters.

Some of the ways we can calculate distance measures include:

- Euclidean distance measure
- Squared Euclidean distance measure
- Manhattan distance measure
- Cosine distance measure

In this assignment I am using **Euclidean distance** measure.

$$||a-b||_2 = \sqrt{\sum_i (a_i - b_i)^2}$$

What is Agglomerative Clustering?

- Agglomerate clustering begins with each element as a separate cluster and merges them into larger clusters.

There are many cluster agglomeration methods (i.e, linkage methods). The most common linkage methods are:

1. **Minimum or single linkage:** The distance between two clusters is defined as the minimum value of all pairwise distances between the elements in cluster 1 and the elements in cluster 2. It tends to produce long, “loose” clusters.
2. **Maximum or complete linkage:** The distance between two clusters is defined as the maximum value of all pairwise distances between the elements in cluster 1 and the elements in cluster 2. It tends to produce more compact clusters.
3. **Mean or average linkage:** The distance between two clusters is defined as the average distance between the elements in cluster 1 and the elements in cluster 2.
4. **Centroid linkage:** The distance between two clusters is defined as the distance between the centroid for cluster 1 (a mean vector of length p variables) and the centroid for cluster 2.

Dendrogram the graphical representation of the hierarchical tree.

Steps to Agglomerative Hierarchical Clustering (AHC)

I will follow the steps below to perform agglomerative hierarchical clustering using R software:

1. Import the dataset
2. Preparing the data
    - Normalize the data
3. Calculate Euclidean distance
    - Computing (dis)similarity information between every pair of objects in the data set.
4. Apply AHC using linkage functions and Create a dendrogram
    - Using linkage function to group objects into hierarchical cluster tree, based on the distance information generated at step 1. Objects/clusters that are in close proximity are linked together using the linkage function.
5. Determining where to cut the hierarchical tree into clusters. This creates a partition of the data.

2 (a). Implementing agglomerative hierarchical clustering (AHC)

First, Reading the 'NCI microarray` dataset. Then applying agglomerative hierarchical clustering with different linkage functions.

Step 1. Importing the dataset.

```
ncidata <- read.table("ncidata.txt")
ncidata <- t(ncidata)
any(is.na(ncidata))
```

```
## [1] FALSE
```

```
dim(ncidata)
```

```
## [1]   64 6830
```

Step 2. Preparing the data.

```
# Standardize the data
ncidata <- scale(ncidata)
```

Step 3. Calculate Euclidean distance.

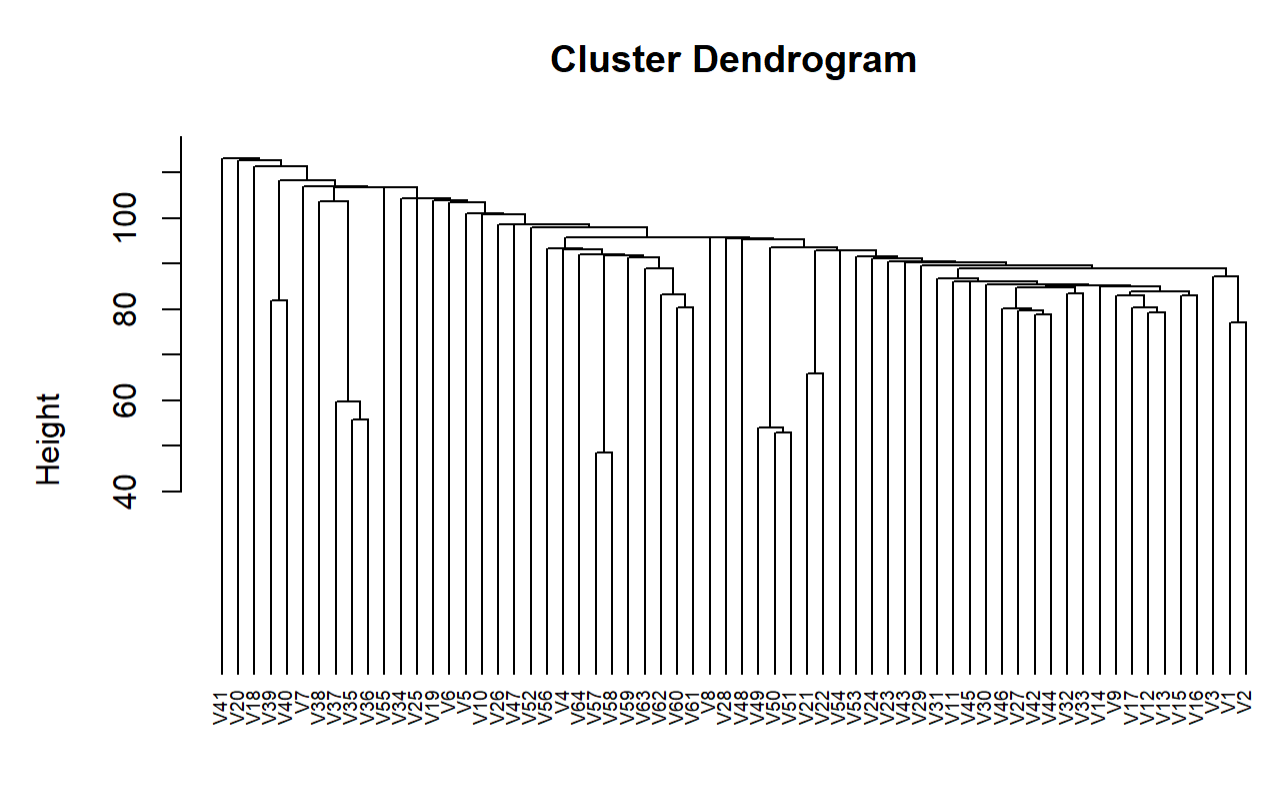
```
# Finding distance matrix
ahc.dist <- dist(ncidata, method = "euclidean")
as.matrix(ahc.dist)[1:5, 1:5]
```

```
##           V1        V2        V3        V4       V5
## V1   0.00000  77.04594  87.30561 103.18322 113.7230
## V2  77.04594   0.00000  88.89531 106.64318 116.1610
## V3  87.30561  88.89531   0.00000  95.79984 101.0443
## V4 103.18322 106.64318  95.79984   0.00000 107.0625
## V5 113.72295 116.16097 101.04429 107.06253   0.0000
```

Step 4. Applying AHC using linkage functions and Create dendrograms.

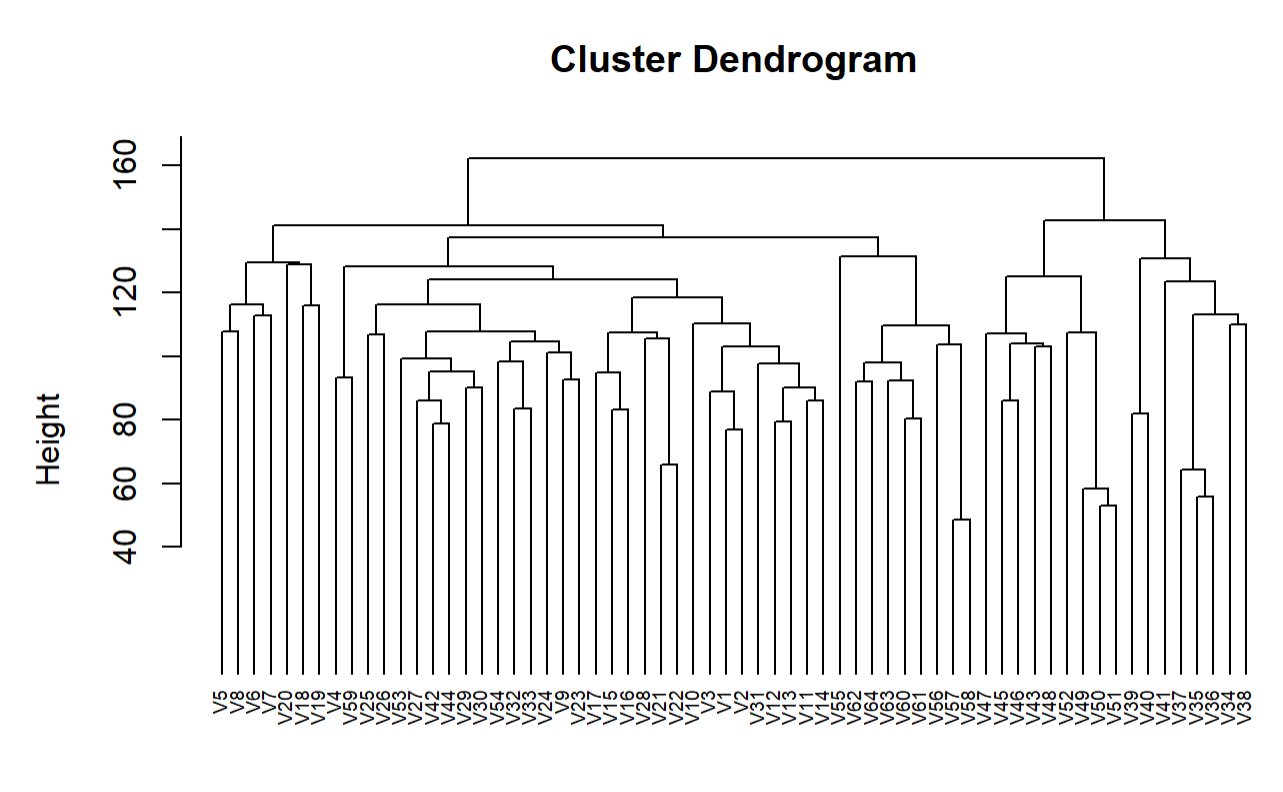
```
# methods to assess
m <- c("single", "complete", "average", "centroid")

# function to compute linkage functions.
AHC <- function(x) {
    for (i in x) {
        ahc.di <- dist(ncidata, method = "euclidean")
        ahc.med <- hclust(ahc.di, method = i)
        print(ahc.med)
        plot(ahc.med, hang = -1, cex = 0.6)
    }
}

AHC(m)
```

```
##
## Call:
## hclust(d = ahc.di, method = i)
##
## Cluster method   : single
## Distance         : euclidean
## Number of objects: 64
```



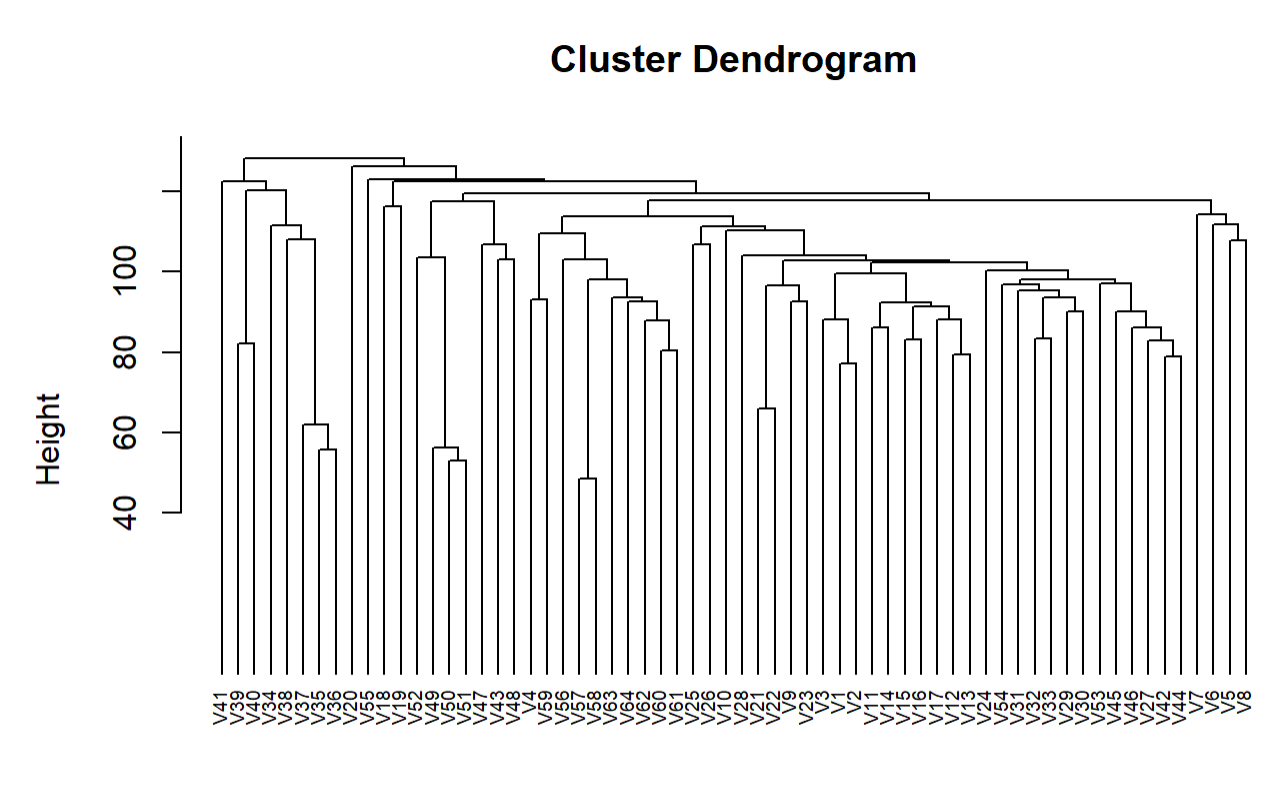
ahc.di
hclust (*, "single")

```
##
## Call:
## hclust(d = ahc.di, method = i)
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 64
```



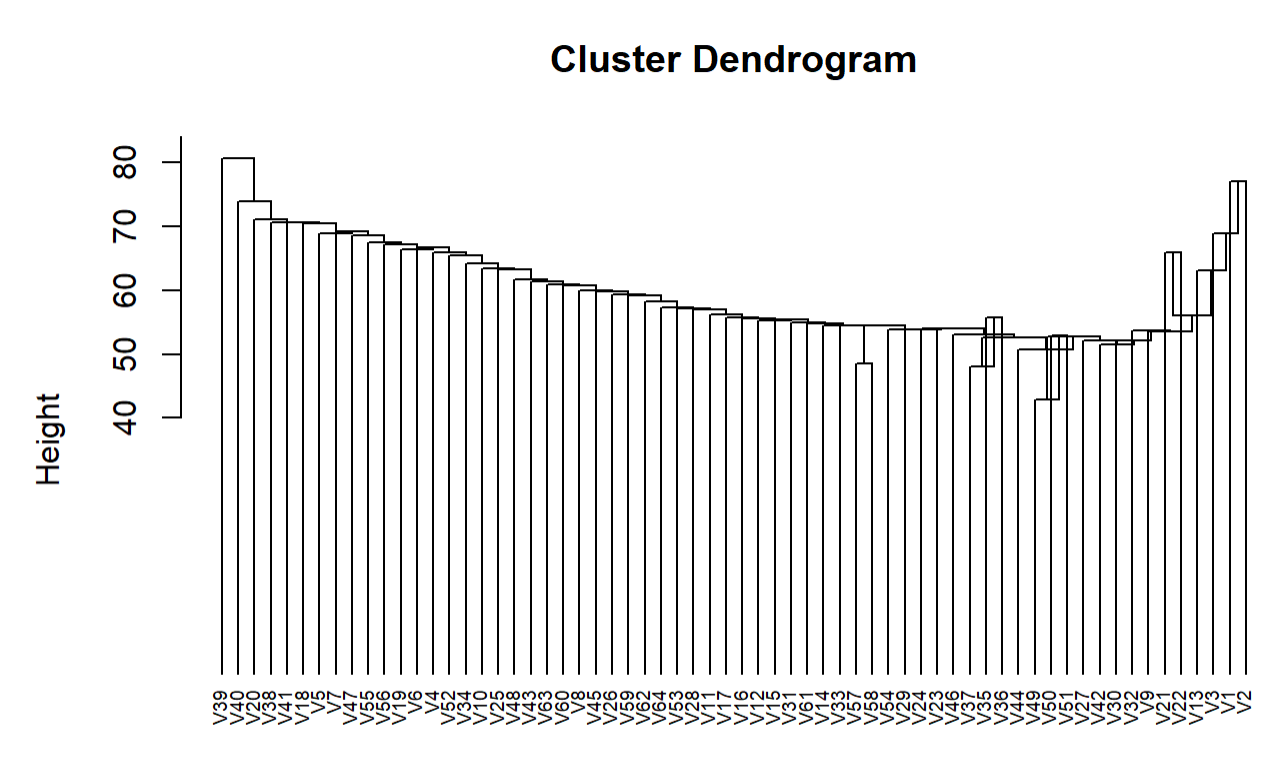
ahc.di
hclust (*, "complete")

```
##
## Call:
## hclust(d = ahc.di, method = i)
##
## Cluster method   : average
## Distance         : euclidean
## Number of objects: 64
```



ahc.di
hclust (*, "average")

```
##
## Call:
## hclust(d = ahc.di, method = i)
##
## Cluster method   : centroid
## Distance         : euclidean
## Number of objects: 64
```



ahc.di
hclust (*, "centroid")

Implementing agglomerative hierarchical clustering (AHC) with single linkage.

```
# Agglomerative Hierarchical Clustering for single linkage
start.time <- Sys.time()
ahc.single <- hclust(ahc.dist, method="single")
ahc.single
```

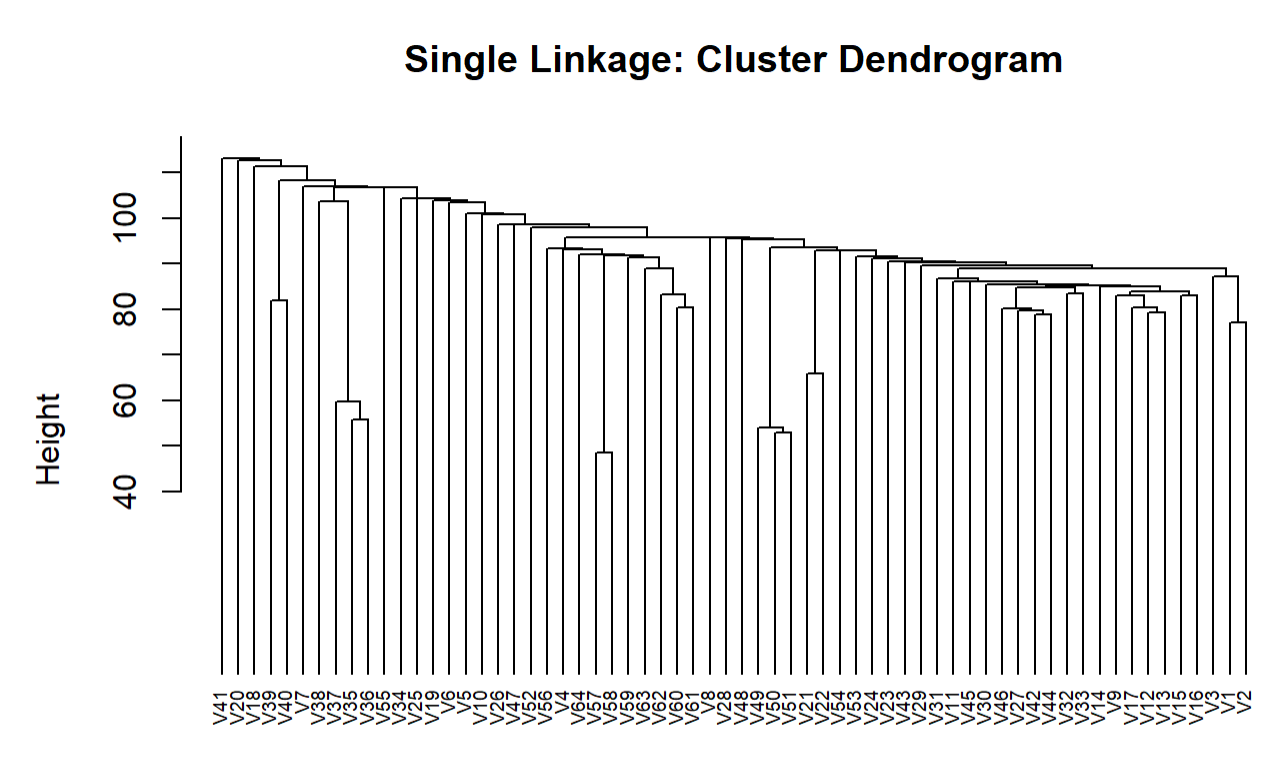
```
##
## Call:
## hclust(d = ahc.dist, method = "single")
##
## Cluster method   : single
## Distance         : euclidean
## Number of objects: 64
```

```
end.time <- Sys.time()
print(end.time - start.time)
```

```
## Time difference of 0 secs
```

Plotting a dendrogram for single linkage.

```
# Plotting a dendrogram for single linkage
plot(ahc.single, hang = -1, cex = 0.6, main="Single Linkage: Cluster Dendrogram")
```



ahc.dist
hclust (*, "single")

Implementing agglomerative hierarchical clustering (AHC) with complete linkage.

```
# Agglomerative Hierarchical Clustering for complete linkage
start.time <- Sys.time()
ahc.complete <- hclust(ahc.dist , method="complete")
ahc.complete
```

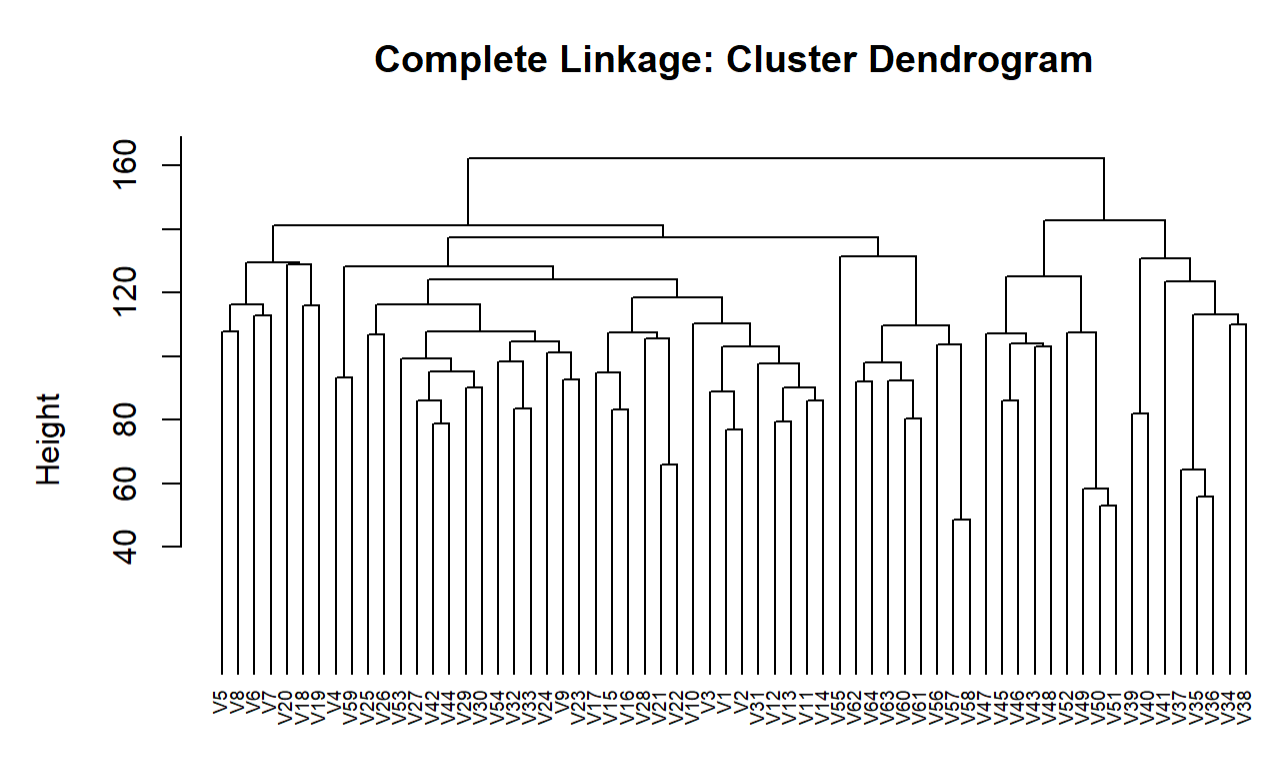
```
##
## Call:
## hclust(d = ahc.dist, method = "complete")
##
## Cluster method   : complete
## Distance         : euclidean
## Number of objects: 64
```

```
end.time <- Sys.time()
print(end.time - start.time)
```

```
## Time difference of 0.004117012 secs
```

Plotting a dendrogram for complete linkage.

```
# Plotting a dendrogram for complete linkage
plot(ahc.complete, hang = -1, cex = 0.6, main="Complete Linkage: Cluster Dendrogram")
```



ahc.dist
hclust (*, "complete")

Implementing agglomerative hierarchical clustering (AHC) with average linkage.

```
# Agglomerative Hierarchical Clustering for average linkage
start.time <- Sys.time()
ahc.average <- hclust(ahc.dist , method="average" )
ahc.average
```

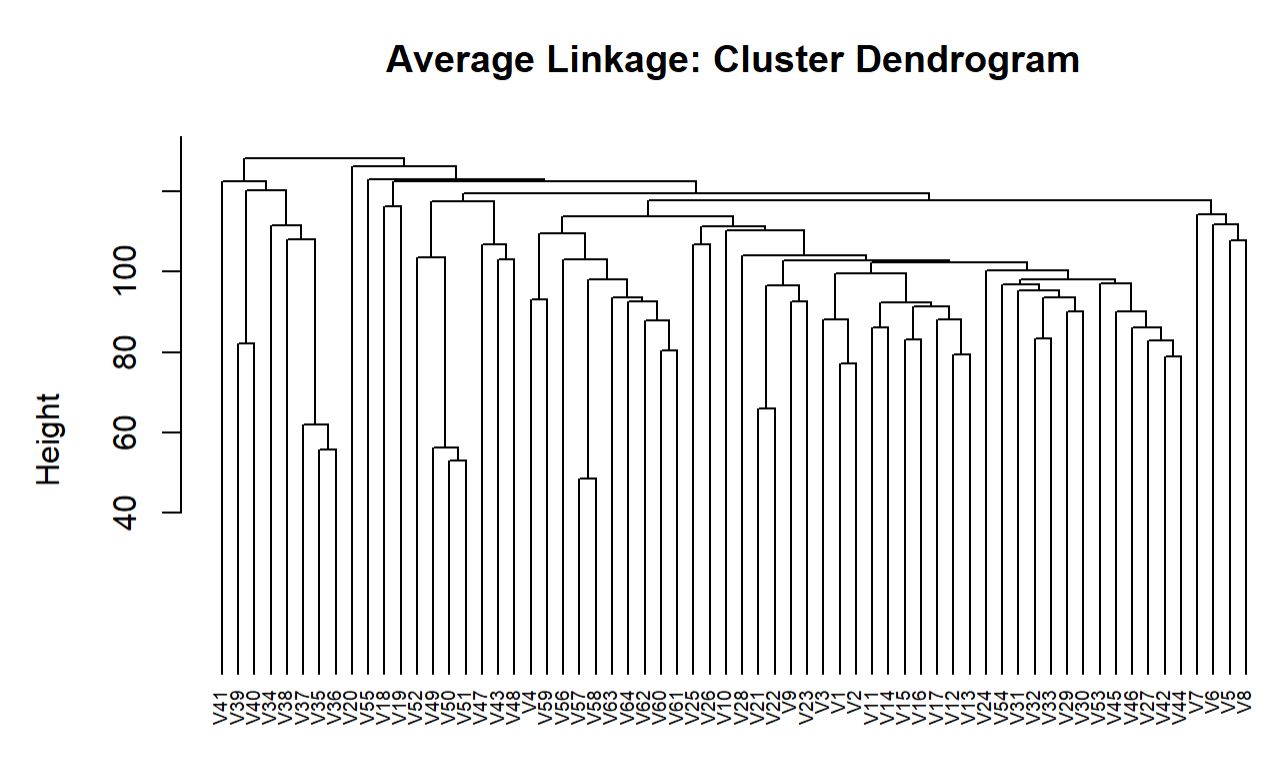
```
##
## Call:
## hclust(d = ahc.dist, method = "average")
##
## Cluster method   : average
## Distance         : euclidean
## Number of objects: 64
```

```
end.time <- Sys.time()
print(end.time - start.time)
```

```
## Time difference of 0.002038002 secs
```

Plotting a dendrogram for average linkage.

```
# plotting a dendrogram for average linkage
plot(ahc.average,hang = -1, cex = 0.6, main="Average Linkage: Cluster Dendrogram")
```



ahc.dist
hclust (*, "average")

Implementing agglomerative hierarchical clustering (AHC) with centroid linkage.

```
# Agglomerative Hierarchical Clustering for centroid linkage
start.time <- Sys.time()
ahc.centroid <- hclust(ahc.dist, method="centroid")
ahc.centroid
```

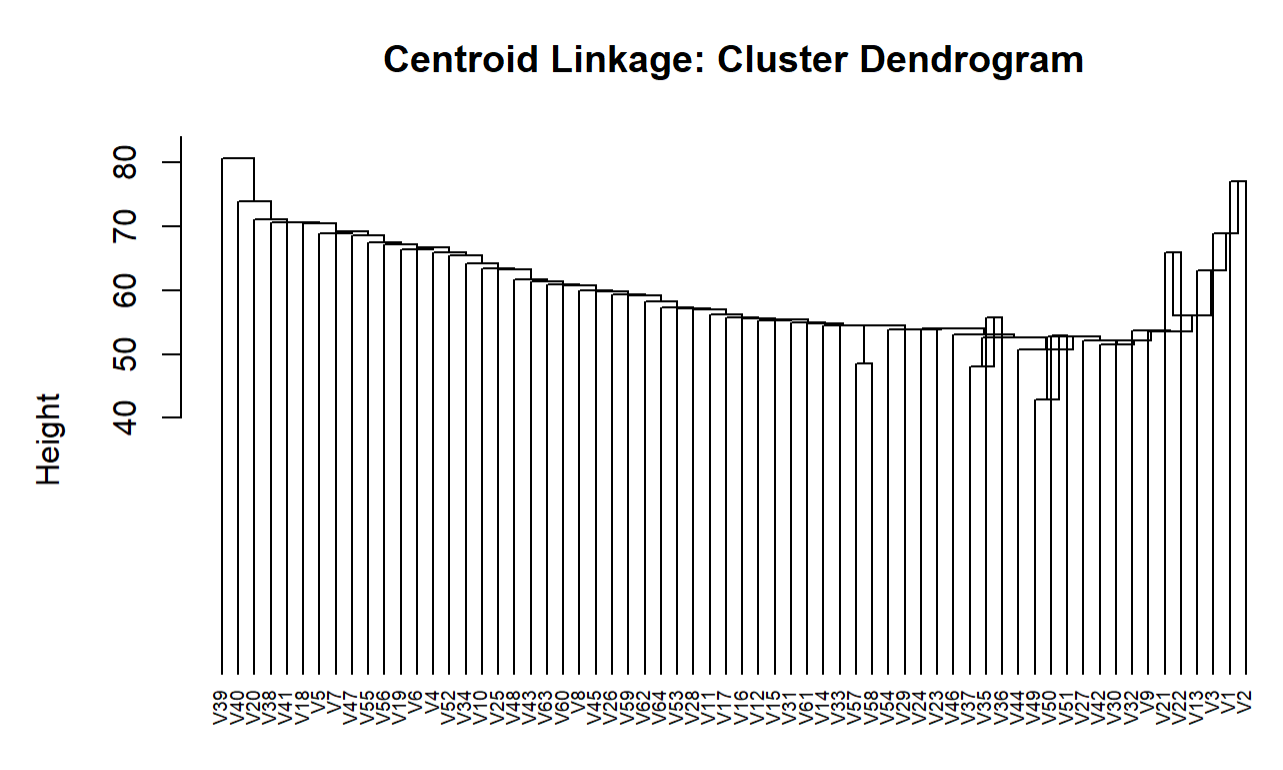
```
##
## Call:
## hclust(d = ahc.dist, method = "centroid")
##
## Cluster method   : centroid
## Distance         : euclidean
## Number of objects: 64
```

```
end.time <- Sys.time()
print(end.time - start.time)
```

```
## Time difference of 0.004064798 secs
```

plotting a dendrogram for centroid linkage.

```
# plotting a dendrogram for centroid linkage
plot(ahc.centroid, hang = -1, cex = 0.6, main="Centroid Linkage: Cluster Dendrogram")
```



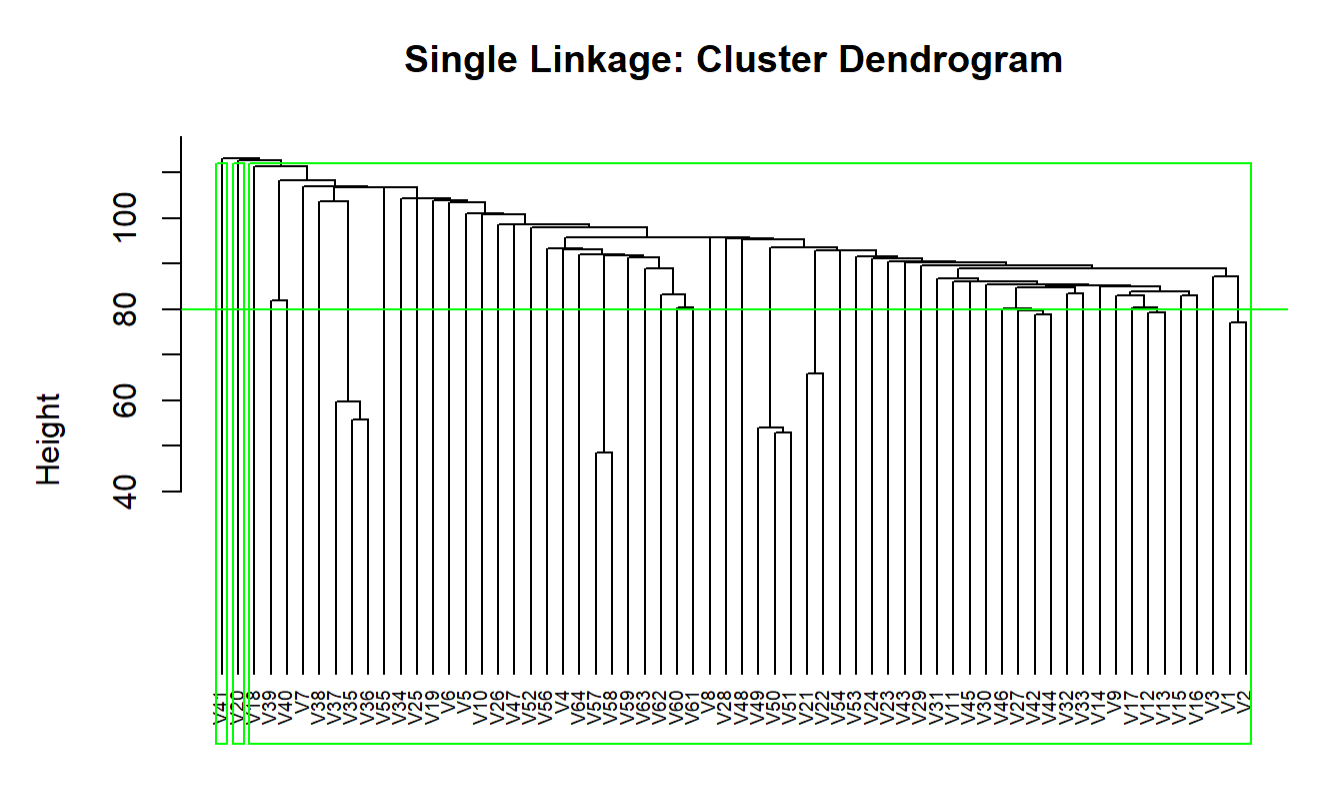
ahc.dist
hclust (*, "centroid")

2 (b). Implementing a function getClusters

```
## Choosing no. of clusters
# Cutting tree by height
plot(ahc.single, hang = -1, cex = 0.6, main="Single Linkage: Cluster Dendrogram")
abline(h = 80, col = "green")
# Cutting tree by no. of clusters
fit <- cutree(ahc.single, k = 3 )

table(fit)
```

```
## fit
##  1  2  3
## 62  1  1
```

```
rect.hclust(ahc.single, k = 3, border = "green")
```



ahc.dist
hclust (*, "single")

References

1. https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering/
2. https://en.wikipedia.org/wiki/Hierarchical_clustering
3. https://www.simplilearn.com/tutorials/data-science-tutorial/hierarchical-clustering-in-r?&utm_medium=Description&utm_source=youtube