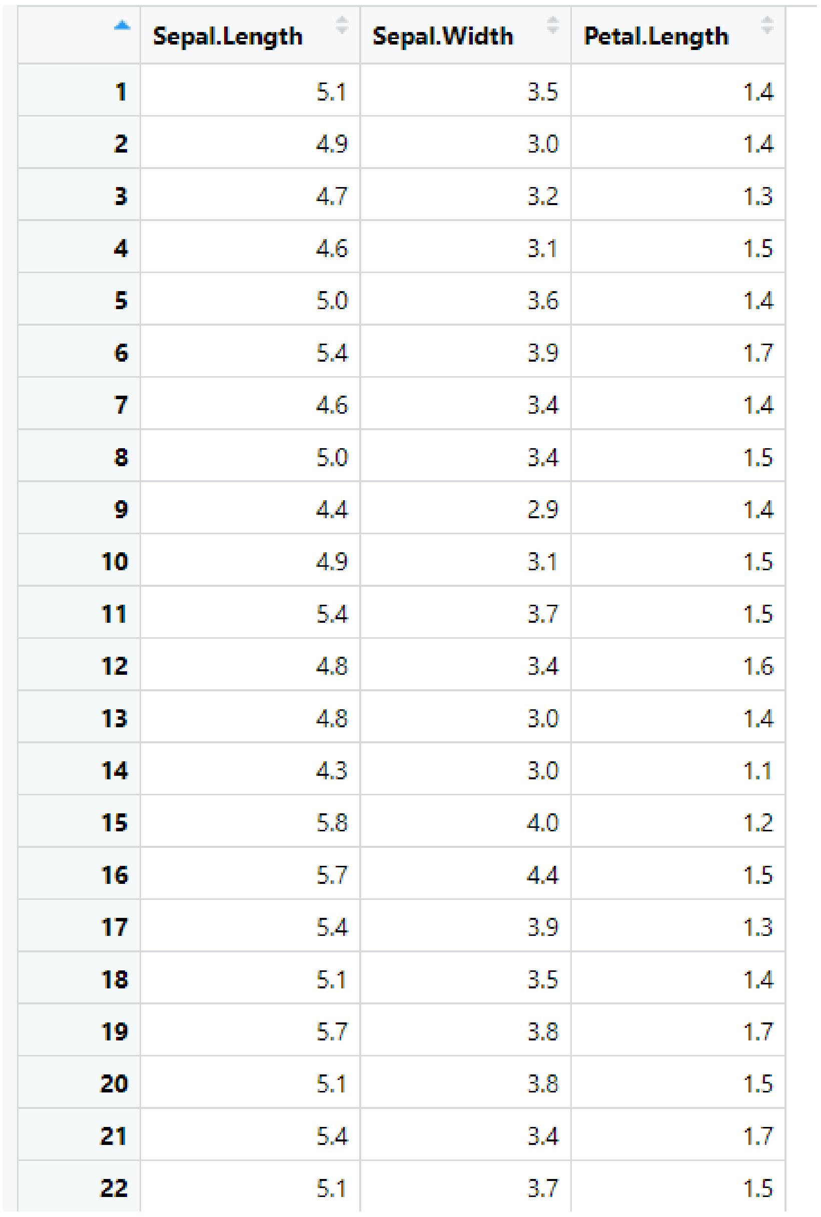
Name: **Varun Sudhir**

Reg No : **21BDS0040**

Exploratory Data Analysis Problem Set 3

Write R programs to implement the agglomerative hierarchical clustering using own values and datasets.

Dataset (Iris dataset) :



	Sepal.Length	Sepal.Width	Petal.Length
1	5.1	3.5	1.4
2	4.9	3.0	1.4
3	4.7	3.2	1.3
4	4.6	3.1	1.5
5	5.0	3.6	1.4
6	5.4	3.9	1.7
7	4.6	3.4	1.4
8	5.0	3.4	1.5
9	4.4	2.9	1.4
10	4.9	3.1	1.5
11	5.4	3.7	1.5
12	4.8	3.4	1.6
13	4.8	3.0	1.4
14	4.3	3.0	1.1
15	5.8	4.0	1.2
16	5.7	4.4	1.5
17	5.4	3.9	1.3
18	5.1	3.5	1.4
19	5.7	3.8	1.7
20	5.1	3.8	1.5
21	5.4	3.4	1.7
22	5.1	3.7	1.5

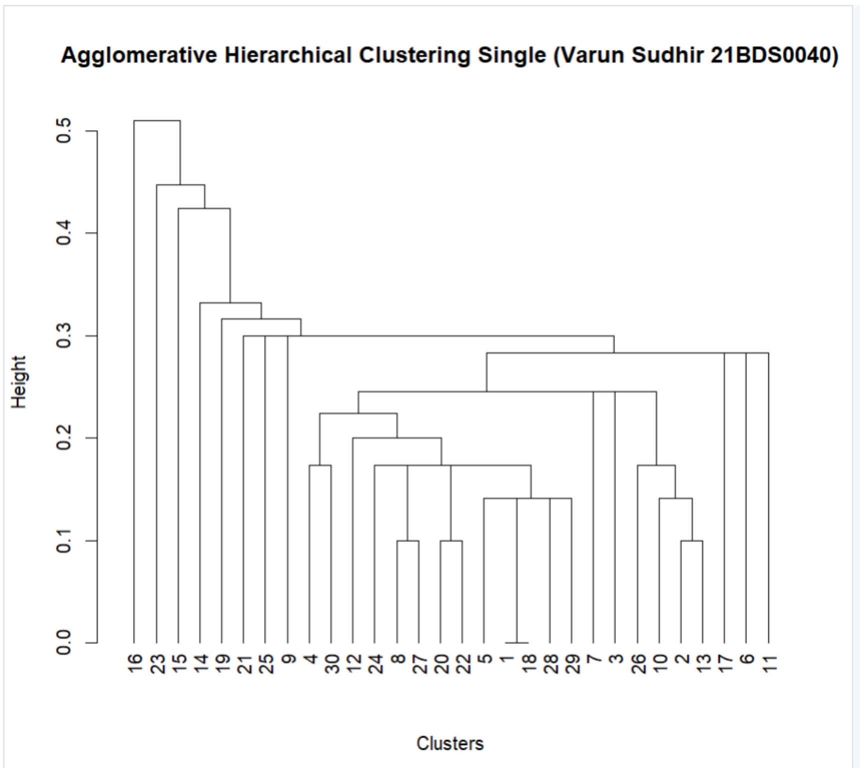
Code:

```
# Function to compute the Euclidean distance between two data points
euclidean_distance <- function(x, y) {
  sqrt(sum((x - y)^2))
}

# Function to create the distance matrix
compute_distance_matrix <- function(data) {
  n <- nrow(data)
  dist_matrix <- matrix(0, n, n)

  for (i in 1:(n-1)) {
    for (j in (i+1):n) {
      dist_matrix[i, j] <- euclidean_distance(data[i,], data[j,])
      dist_matrix[j, i] <- dist_matrix[i, j]
    }
  }

  return(dist_matrix)
}

# Function to compute single linkage (minimum distance between clusters)
single_linkage <- function(cluster1, cluster2, dist_matrix) {
  min_dist <- Inf
  for (i in cluster1) {
    for (j in cluster2) {
      if (dist_matrix[i, j] < min_dist) {
        min_dist <- dist_matrix[i, j]
      }
    }
  }
  return(min_dist)
}

# Function to compute complete linkage (maximum distance between clusters)
complete_linkage <- function(cluster1, cluster2, dist_matrix) {
  max_dist <- -Inf
  for (i in cluster1) {
    for (j in cluster2) {
      if (dist_matrix[i, j] > max_dist) {
        max_dist <- dist_matrix[i, j]
      }
    }
  }
  return(max_dist)
}

# Function to compute average linkage (average distance between clusters)
```

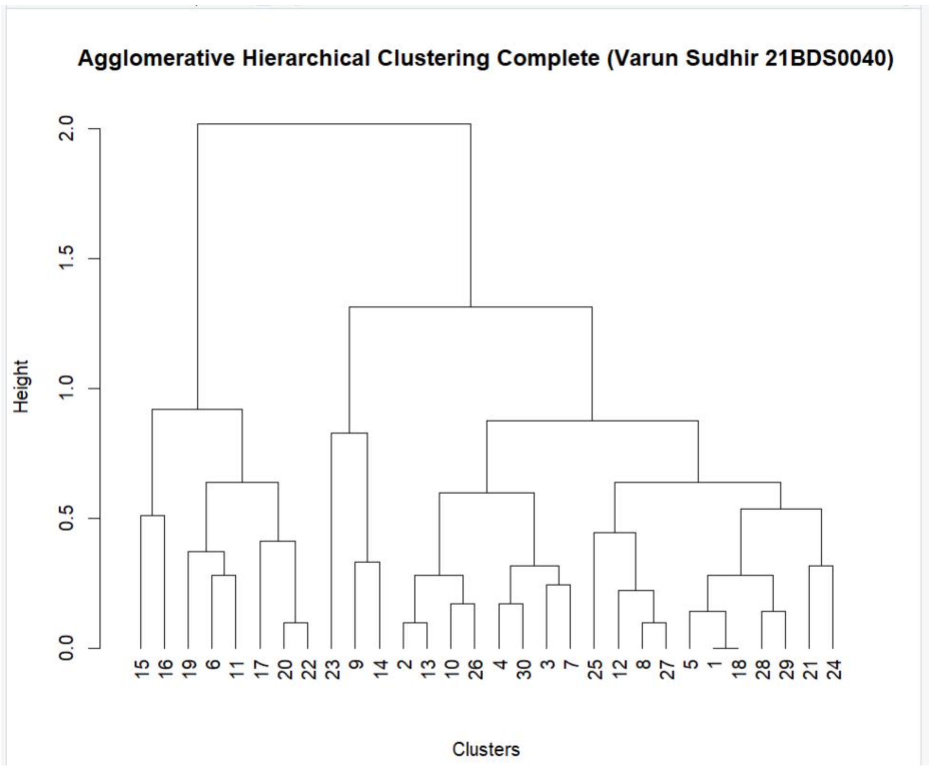
```r
average_linkage <- function(cluster1, cluster2, dist_matrix) {
  total_dist <- 0
  count <- 0
  for (i in cluster1) {
    for (j in cluster2) {
      total_dist <- total_dist + dist_matrix[i, j]
      count <- count + 1
    }
  }
  return(total_dist / count)
}

agglomerative_clustering <- function(data, linkage_method) {
  n <- nrow(data)
  dist_matrix <- compute_distance_matrix(data)
  clusters <- as.list(1:n)
  merge_matrix <- matrix(0, n - 1, 2)
  heights <- numeric(n - 1)
  cluster_index <- -(1:n)

  step <- 1
  while (length(clusters) > 1) {
    min_dist <- Inf
    to_merge <- c()
    for (i in 1:(length(clusters)-1)) {
      for (j in (i+1):length(clusters)) {
        dist_ij <- linkage_method(clusters[[i]], clusters[[j]], dist_matrix)
        if (dist_ij < min_dist) {
          min_dist <- dist_ij
          to_merge <- c(i, j)
        }
      }
    }

    merged_cluster <- c(clusters[[to_merge[1]]], clusters[[to_merge[2]]])
    clusters <- clusters[-to_merge]
    clusters <- append(clusters, list(merged_cluster))
    merge_matrix[step, ] <- c(cluster_index[to_merge[1]],
cluster_index[to_merge[2]])
    heights[step] <- min_dist
    cluster_index <- cluster_index[-to_merge]
    cluster_index <- c(cluster_index, step)

    step <- step + 1
  }

  return(list(merge_matrix = merge_matrix, heights = heights))
}
```

```r
plot_dendrogram <- function(merge_history, n, linkage_choice) {
  merge_matrix <- merge_history$merge_matrix
  heights <- merge_history$heights

  hclust_data <- list(
    merge = merge_matrix,
    height = heights,
    order = 1:n,
    labels = NULL,
    method = "agglomerative",
    dist.method = "euclidean"
  )

  class(hclust_data) <- "hclust"

  plot(as.dendrogram(hclust_data),
       main = paste("Agglomerative Hierarchical Clustering", linkage_choice,
"(Varun Sudhir 21BDS0040)"),
       ylab = "Height",
       xlab = "Clusters")
}

data <- iris[1:30, c("Sepal.Length", "Sepal.Width", "Petal.Length")]

linkage_choice <- "Average"

# Select the corresponding linkage function based on user input
linkage_method <- switch(linkage_choice,
                         "Single" = single_linkage,
                         "Complete" = complete_linkage,
                         "Average" = average_linkage)

# Perform agglomerative clustering
merge_history <- agglomerative_clustering(data, linkage_method)

# Plot the dendrogram
plot_dendrogram(merge_history, nrow(data), linkage_choice)
```

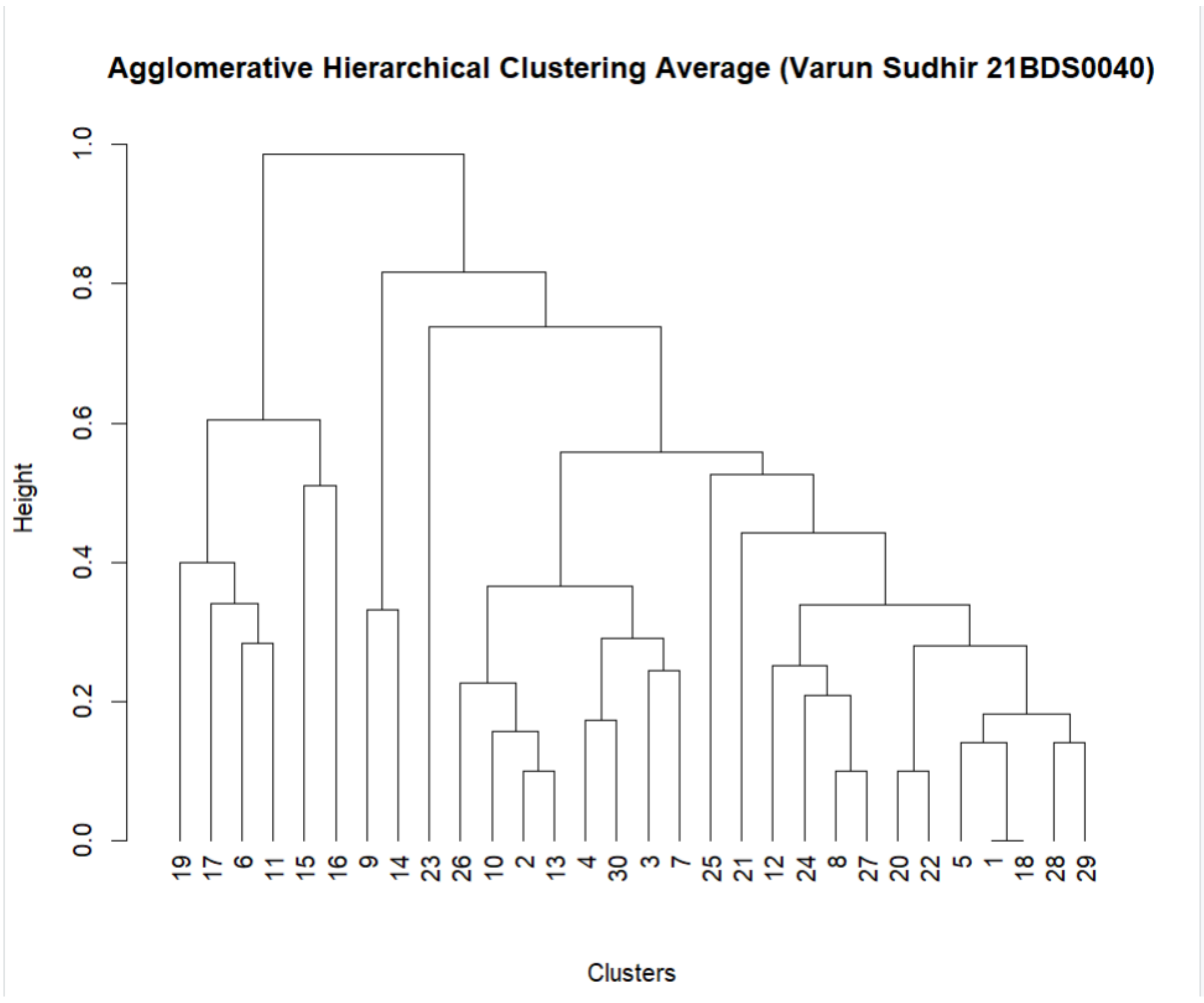
Output (Single – linkage) :



Output (Complete – linkage)



Output (Average linkage)


Agglomerative Hierarchical Clustering Average (Varun Sudhir 21BDS0040)
Height
Clusters