## Hierarchical Clustering

Hierarchical clustering is a popular technique used in data mining to identify underlying patterns and structures within a dataset. There are two main types of hierarchical clustering: agglomerative and divisive. Agglomerative clustering starts with individual data points as separate clusters and then iteratively merges them based on their similarity, while divisive clustering begins with the entire dataset as a single cluster and then recursively splits it into smaller subclusters. The clustering algorithm then iterates or splits clusters based on this similarity measure, until all data points are grouped into a single cluster or individual clusters.

It is a versatile and widely used technique in various fields such as biology, marketing, and social science, but it can be computationally intensive and requires careful selection of distance measures and clustering parameters.

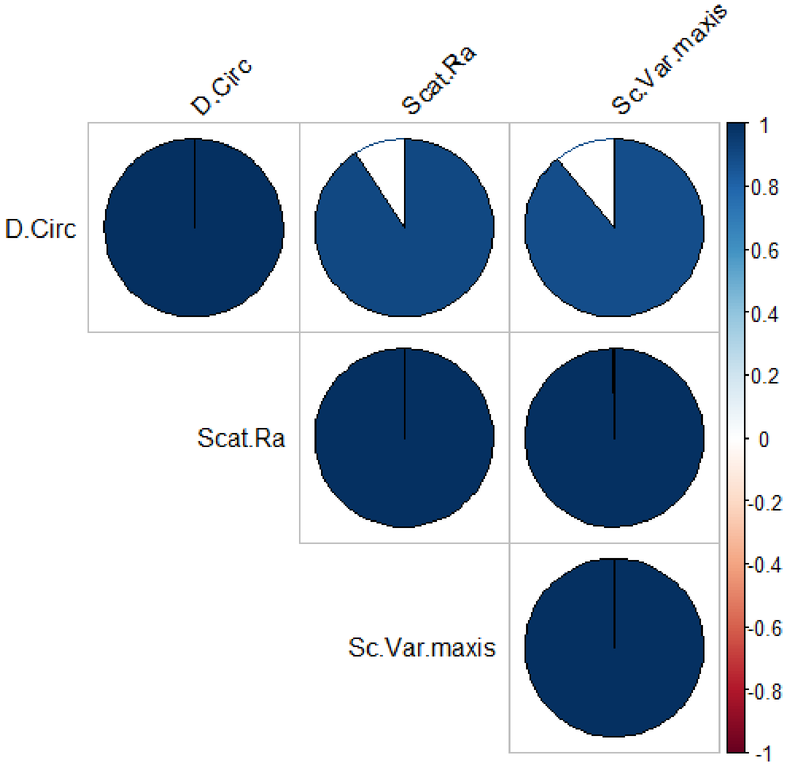
All the data preprocessing and Hierarchical clustering analysis tasks were carried out in RStudio with R computational language.

Loading the dataset to the RStudio. Code is demonstrated as follows

```
# Load the dataset and preprocess
vehicle <- read.csv("C:/Users/Gihan/Documents/Data mining course work/Q2
clustering/vehicle.csv")

# Remove "Samples" and "Class" columns
vehicle <- vehicle[, -c(1, ncol(vehicle))]

 # Normalize the data
vehicle_norm <- scale(vehicle[, -1])

 # Identify and remove outliers using z-score
z_scores <- apply(vehicle_norm, 1, function(x) max(abs(scale(x)))) vehicle_clean
<- vehicle_norm[z_scores < 1,]

# Reduce selected columns
vehicle_reduced <- vehicle_norm[, c("D.Circ", "Scat.Ra", "Sc.Var.maxis")]

#Load the Library
library(colorspace)
```

**Calculating the correlation matrix of each attribute before applying hierarchical clustering methods**

```
# Calculate the correlation matrix
cor_matrix <- cor(vehicle_reduced)
cor_matrix

# Create a correlation plot using the corrplot function
library(corrplot)
corrplot(cor_matrix, method = "pie", type = "upper", tl.col = "black", tl.srt =
45)
```



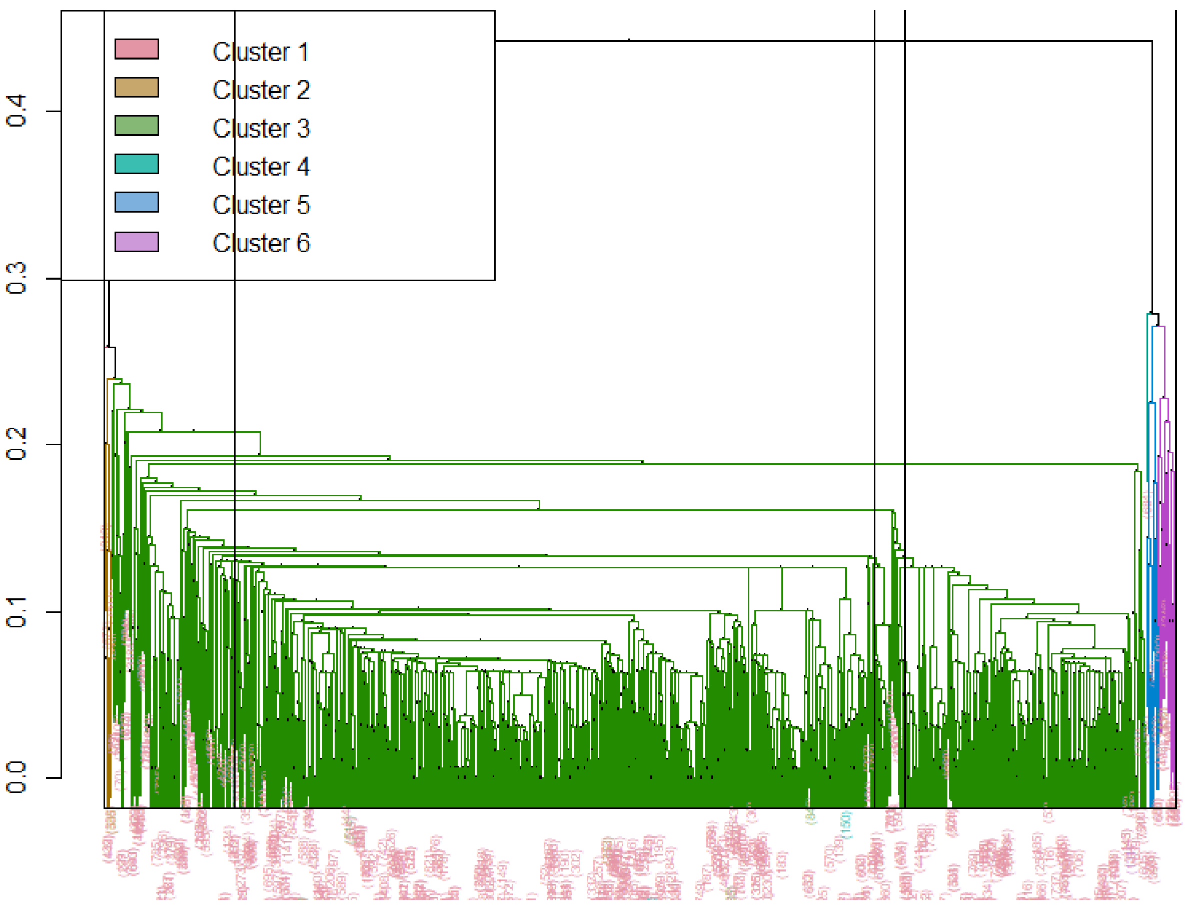
| | D.Circ | Scat.Ra | Sc.Var.maxis |
|---|---|---|---|
| **D.Circ** | 1.0000000 | 0.9072801 | 0.8896611 |
| **Scat.Ra** | 0.9072801 | 1.0000000 | 0.9963180 |
| **Sc.Var.maxis** | 0.8896611 | 0.9963180 | 1.0000000 |

The correlation matrix shows the correlation coefficients between each pair of attributes. The correlation coefficient ranges from -1 to +1, where -1 indicates a perfect negative correlation, 0 indicates no correlation, and +1 indicates a perfect positive correlation.

### Hierarchical Clustering - Performing Single Method

Single linkage is a linkage method used in hierarchical clustering, where the distance between two clusters is defined as the minimum distance between any two objects in the two clusters. The two clusters that have the smallest distance between them are merged into a new cluster at each iteration, resulting in long, chain-like clusters that are sensitive to outliers and noise.

```
# Perform hierarchical clustering using Single linkage method
d_vehicle <- dist(vehicle_reduced)
hc_vehicle <- hclust(d_vehicle, method = "single")

# Create a dendrogram object and modify it for single method
library(dendextend)
dend <- as.dendrogram(hc_vehicle)

 # Order the dendrogram to match the order of the rows in vehicle_reduced
dend <- rotate(dend, 1:nrow(vehicle_reduced))

# Color the branches based on three clusters
dend <- color_branches(dend, k = 6)  labels_colors(dend) <-
rainbow_hcl(6)[sort_levels_values(cutree(hc_vehicle, k = 6))]  # Color the
labels based on the clusters

# Add parentheses to the labels
labels(dend) <- paste("(", labels(dend), ")", sep = "")
# Adjust the height of the dendrogram
dend <- hang.dendrogram(dend, hang_height = 0.1)

# Reduce the size of the labels
dend <- set(dend, "labels_cex", 0.5)

# Plot the dendrogram with a legend
par(mar = c(3, 3, 3, 7))
plot(dend,
    main = "Clustered Vehicle data set using Single Linkage Method",
    horiz = FALSE, nodePar = list(cex = .007))
legend("topleft", legend = c("Cluster 1", "Cluster 2", "Cluster 3","Cluster
4","Cluster 5","Cluster 6"), fill = rainbow_hcl(6))

# Perform hierarchical clustering using Average linkage method
d_vehicle_avg <- dist(vehicle_reduced)
hc_vehicle_avg <- hclust(d_vehicle_avg, method = "average")
rect.hclust(hc_vehicle_avg, k=4, border="black")
```

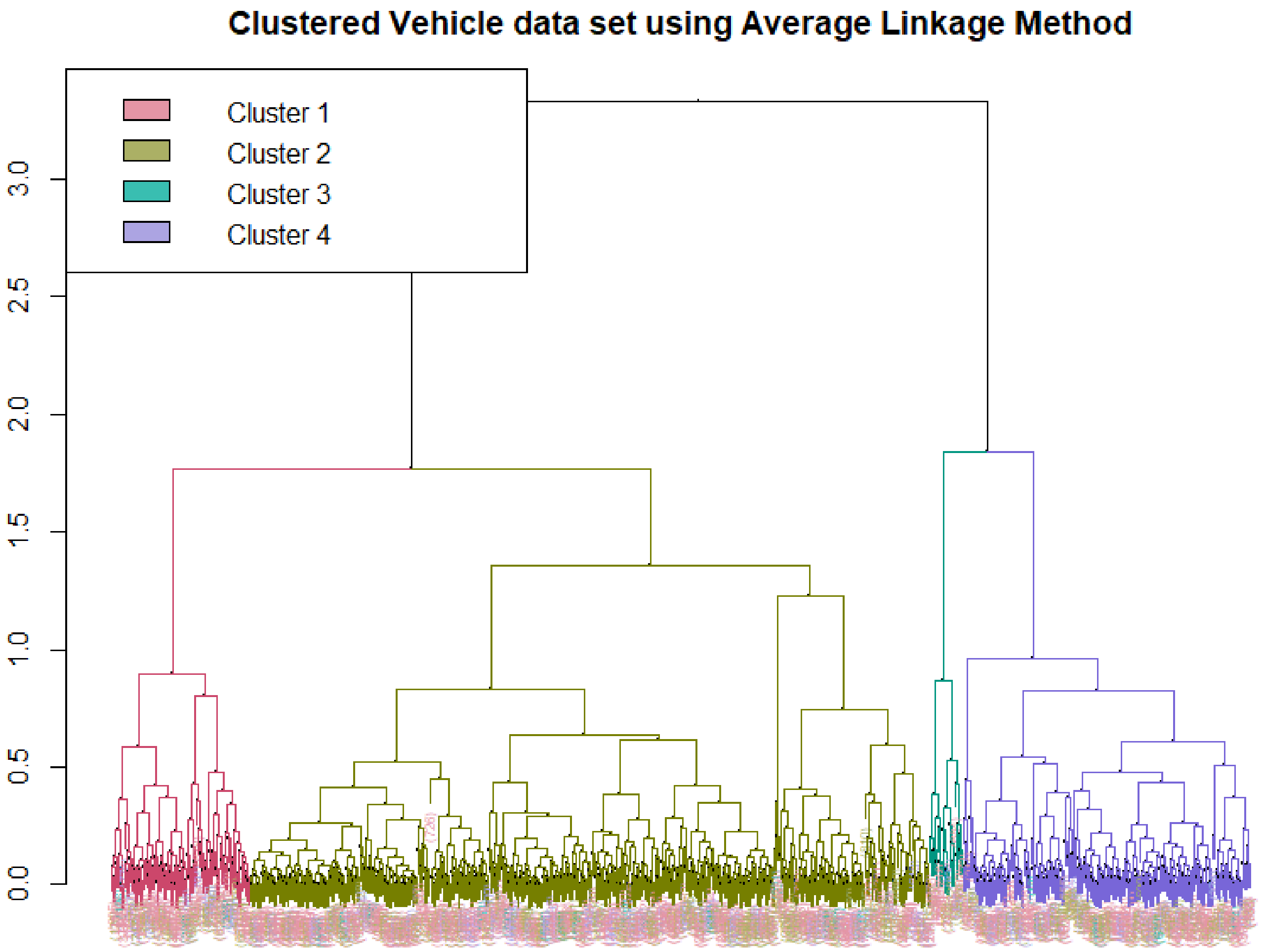


Clustered Vehicle data set using Single Linkage Method

### Hierarchical Clustering - Performing Average Method

Average linkage is a method used to merge two clusters with the smallest average distance between them, resulting in more balanced and compact clusters. However, it can be sensitive to outliers and noise.

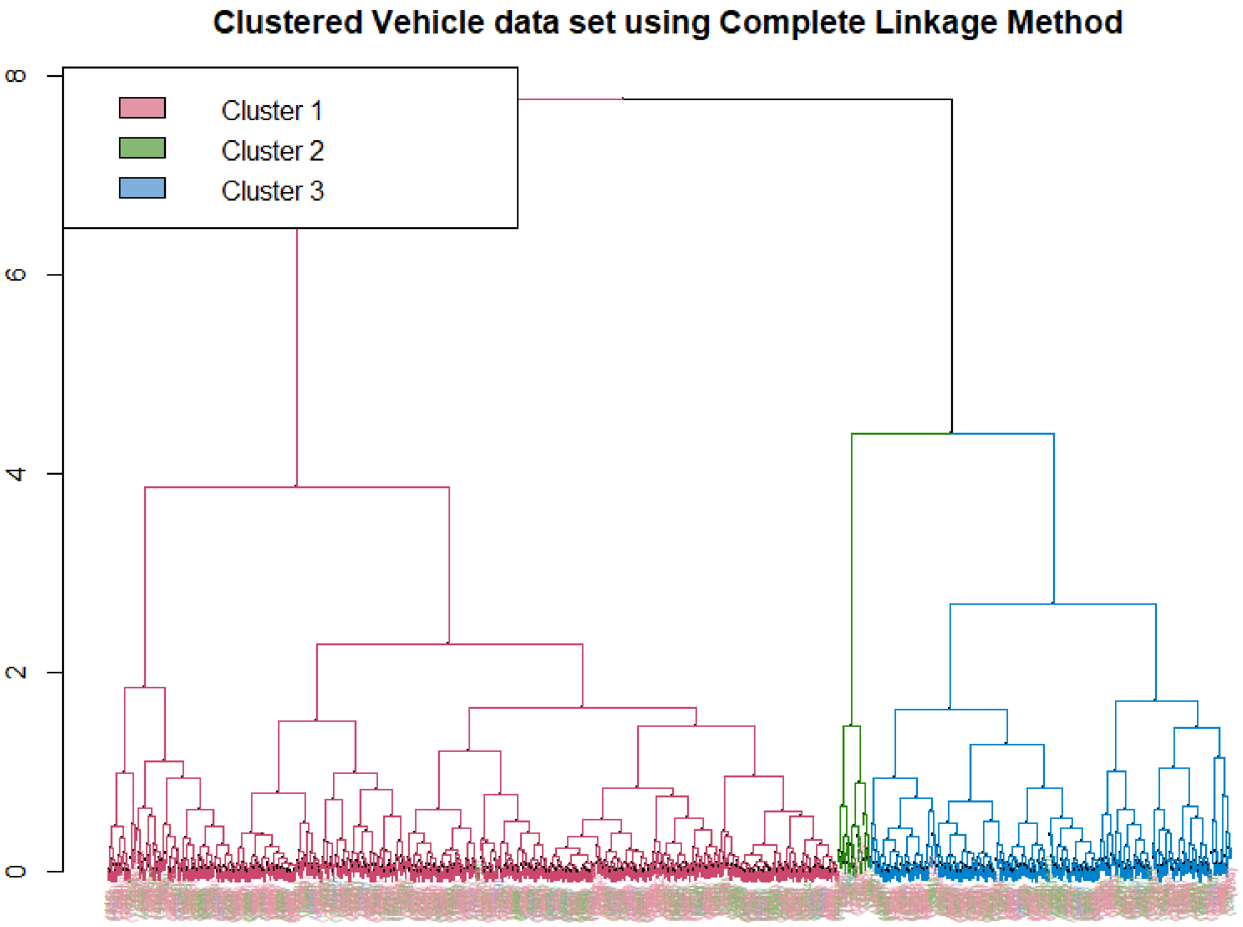
```
# Create a dendrogram object and modify it for average linkage method
dend_avg <- as.dendrogram(hc_vehicle_avg)

 # Order the dendrogram to match the order of the rows in vehicle_reduced
dend_avg <- rotate(dend_avg, 1:nrow(vehicle_reduced))

 # Color the branches based on three clusters
dend_avg <- color_branches(dend_avg, k = 4)

 # Color the labels based on the clusters
labels_colors(dend_avg) <-
rainbow_hcl(4)[sort_levels_values(cutree(hc_vehicle_avg, k = 4))]

# Add parentheses to the labels
labels(dend_avg) <- paste("(", labels(dend_avg), ")", sep = "")

 # Adjust the height of the dendrogram
dend_avg <- hang.dendrogram(dend_avg, hang_height = 0.1)

 # Reduce the size of the labels
dend_avg <- set(dend_avg, "labels_cex", 0.5)

# Plot the dendrogram with a legend for average linkage method
par(mar = c(3, 3, 3, 7))
plot(dend_avg,
    main = "Clustered Vehicle data set using Average Linkage Method",
    horiz = FALSE, nodePar = list(cex = .007))
legend("topleft", legend = c("Cluster 1", "Cluster 2", "Cluster 3","Cluster 4"),
fill = rainbow_hcl(4))

# Perform hierarchical clustering using Complete linkage method
d_vehicle_comp <- dist(vehicle_reduced)
hc_vehicle_comp <- hclust(d_vehicle_comp, method = "complete")
```


Clustered Vehicle data set using Average Linkage Method
Cluster 1
Cluster 2
Cluster 3
Cluster 4
0.0
0.5
1.0
1.5
2.0
2.5
3.0

### Hierarchical Clustering - Performing Complete Method

Complete linkage uses the maximum distance between two clusters to merge them into a new cluster, resulting in compact and spherical clusters that are less sensitive to outliers and noise. However, it can also produce small, disconnected clusters.

```
# Create a dendrogram object and modify it for complete linkage method
dend_comp <- as.dendrogram(hc_vehicle_comp)

 # Order the dendrogram to match the order of the rows in vehicle_reduced
dend_comp <- rotate(dend_comp, 1:nrow(vehicle_reduced))

# Color the branches based on three clusters
dend_comp <- color_branches(dend_comp, k = 3)

# Color the labels based on the clusters
labels_colors(dend_comp) <-
rainbow_hcl(3)[sort_levels_values(cutree(hc_vehicle_comp, k = 3))]

# Add parentheses to the labels
labels(dend_comp) <- paste("(", labels(dend_comp), ")", sep = "")

 # Adjust the height of the dendrogram
dend_comp <- hang.dendrogram(dend_comp, hang_height = 0.1)

# Reduce the size of the labels
dend_comp <- set(dend_comp, "labels_cex", 0.5)

# Plot the dendrogram with a legend for complete linkage method
par(mar = c(3, 3, 3, 7))
plot(dend_comp,
    main = "Clustered Vehicle data set using Complete Linkage Method",
    horiz = FALSE, nodePar = list(cex = .007))
legend("topleft", legend = c("Cluster 1", "Cluster 2", "Cluster 3"), fill =
rainbow_hcl(3))
```

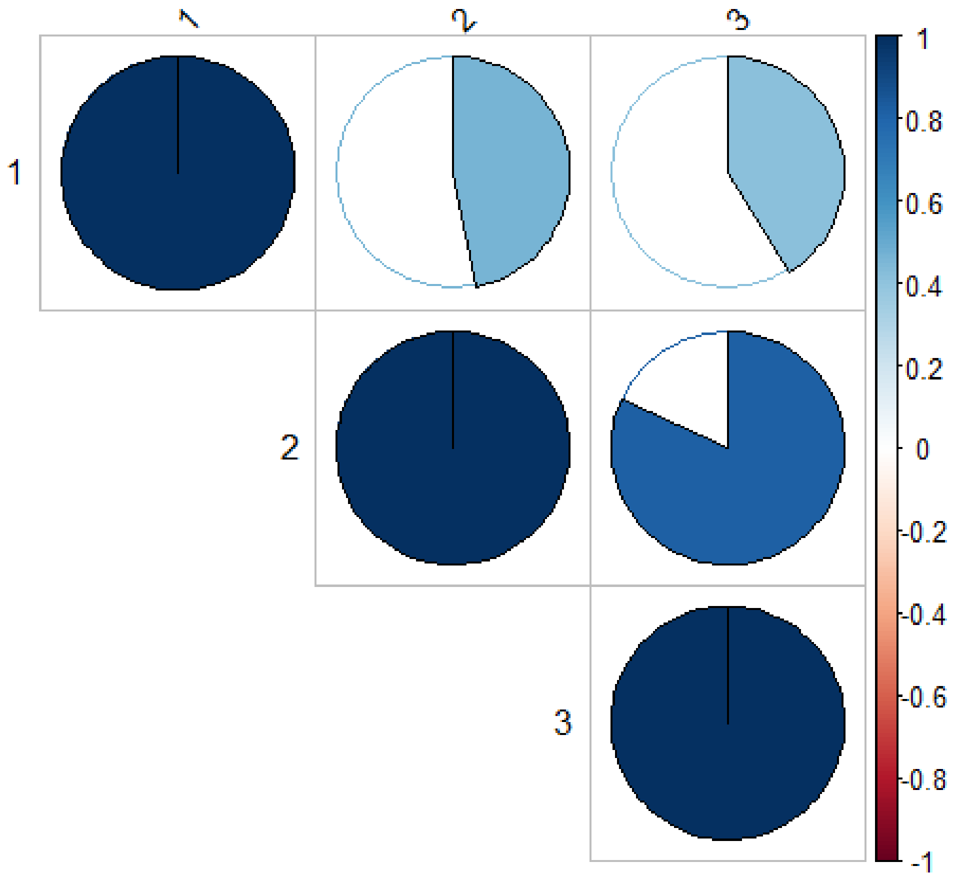

Clustered Vehicle data set using Complete Linkage Method
Cluster 1
Cluster 2
Cluster 3
8
6
4
2
0

### Calculating Cophenetic Correlation

Cophenetic correlation measures the relationship between the distances derived from hierarchical clustering and the pairwise distances of the initial data elements. It has a value between 0 and 1, with larger numbers showing that the hierarchical clustering method better maintains the initial distances.

```
library(dendextend)
# Perform hierarchical clustering using Single linkage method
d_vehicle <- dist(vehicle_reduced)
hc_vehicle <- hclust(d_vehicle, method = "single")

# Calculate the cophenetic correlation for the single linkage method
cc_single <- cophenetic(hc_vehicle)

# Perform hierarchical clustering using Average linkage method
d_vehicle_avg <- dist(vehicle_reduced)
hc_vehicle_avg <- hclust(d_vehicle_avg, method = "average")

# Calculate the cophenetic correlation for the average linkage method
cc_avg <- cophenetic(hc_vehicle_avg)

# Perform hierarchical clustering using Complete linkage method
d_vehicle_comp <- dist(vehicle_reduced)
hc_vehicle_comp <- hclust(d_vehicle_comp, method = "complete")

# Calculate the cophenetic correlation for the complete linkage method
cc_comp <- cophenetic(hc_vehicle_comp)

# Create a dendlist object containing the three dendrograms
dendlist <- dendlist(dend, dend_avg, dend_comp)

# Calculate the correlation between the cophenetic distances of the three
dendrograms
correlation <- cor.dendlist(dendlist, method = "cophenetic")
correlation

# Plot the correlation matrix
corrplot(correlation, method = "pie", type = "upper", tl.col = "black", tl.srt =
45)
```

In order to calculate the Cophenetic Correlation above coding steps followed.



**Cophentic Correlation Result with Column Names**

| | Simple | Average | Complete |
|---|---|---|---|
| Simple | 1.0000000 | 0.4683845 | 0.4100251 |
| Average | 0.4683845 | 1.0000000 | 0.8196372 |
| Complete | 0.4100251 | 0.8196372 | 1.0000000 |

Since the Simple method's Cophenetic correlation coefficient is 1, it flawlessly maintains the initial pairwise distances of the data points. This is due to the fact that the Simple method computes the distance between the two closest data points at each algorithmic stage.

The Cophenetic correlation coefficient for the Average method is 0.4683845, showing that it maintains the original distances less effectively than the Simple method. At each stage of the algorithm, the Average method calculates the distance between the centroids of two clusters, which can result in some loss of information about the initial pairwise distances.

The Cophenetic correlation coefficient for the Complete method is 0.4100251, the lowest of the three methods. When using the Complete method, more information about the initial pairwise distances may be lost because each stage of the algorithm computes the distance between the two most distant data points.

Overall, the Cophenetic correlation coefficients indicate that the Simple method, followed by the Average method, and then the Complete method, is the best option for maintaining the initial pairwise distances in hierarchical clustering. However, the particular objectives of the analysis and the properties of the data being analyzed ultimately determine the clustering technique to be used.