

# Hierarchical Clustering

Hierarchical clustering, also known as hierarchical cluster analysis, is an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other.

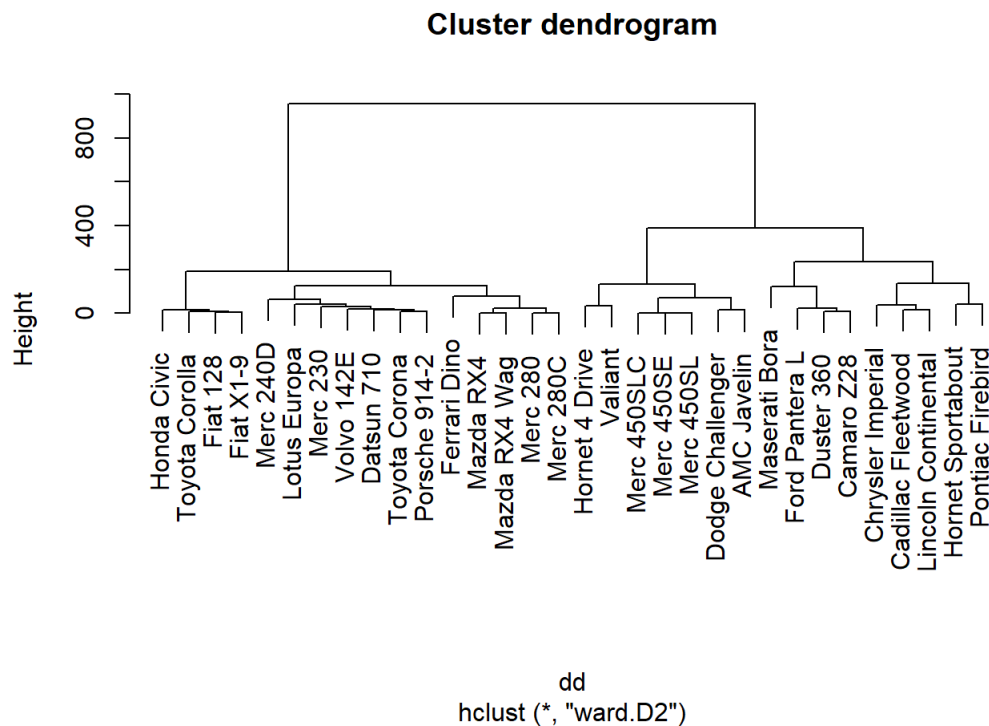
How hierarchical clustering works

Hierarchical clustering starts by treating each observation as a separate cluster. Then, it repeatedly executes the following two steps: (1) identify the two clusters that are closest together, and (2) merge the two most similar clusters. This continues until all the clusters are merged together.

Experiment

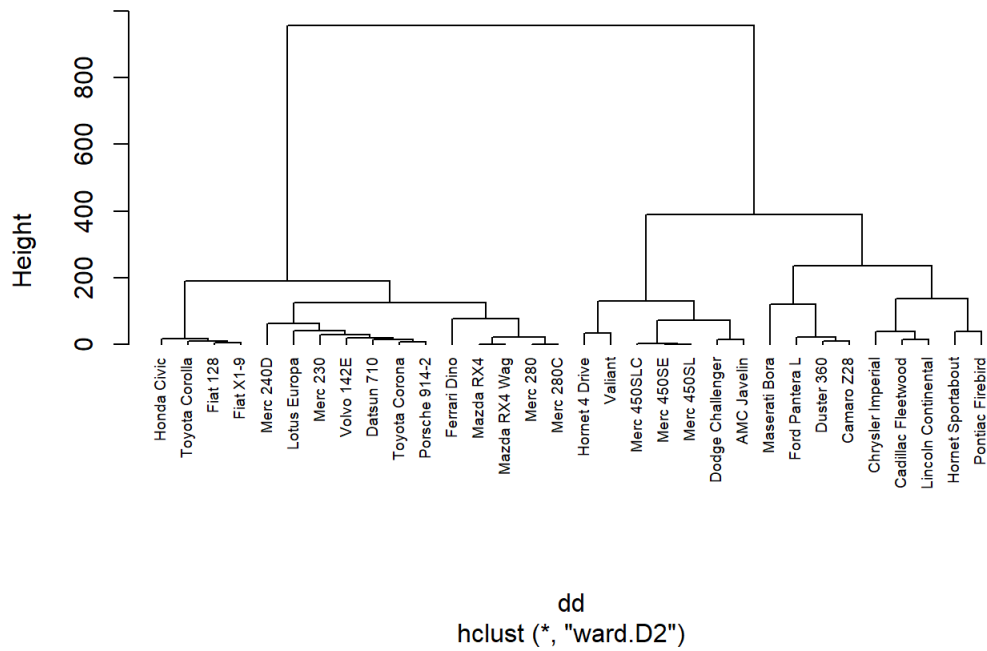
1. Load the mtcars dataset and Compute the distance and hierarchical clustering using `hclust()` and Display the hclustering using `plot()`

```
data(mtcars)
# Compute distances and hierarchical clustering
dd <- dist(mtcars, method = "euclidean")
hc <- hclust(dd, method = "ward.D2")
plot(hc, labels = NULL, hang = 0.1, main = "Cluster dendrogram", sub = NULL,
      xlab = NULL, ylab = "Height")
```



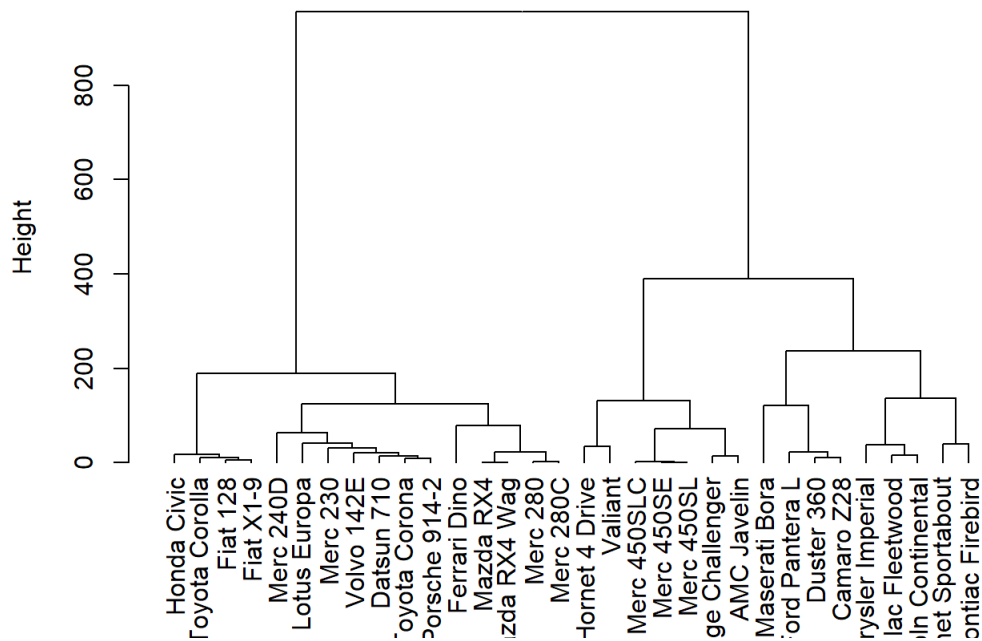
```
#plot(hc)
# Put the labels at the same height: hang = -1
plot(hc, hang = -1, cex = 0.6)
```

## Cluster Dendrogram



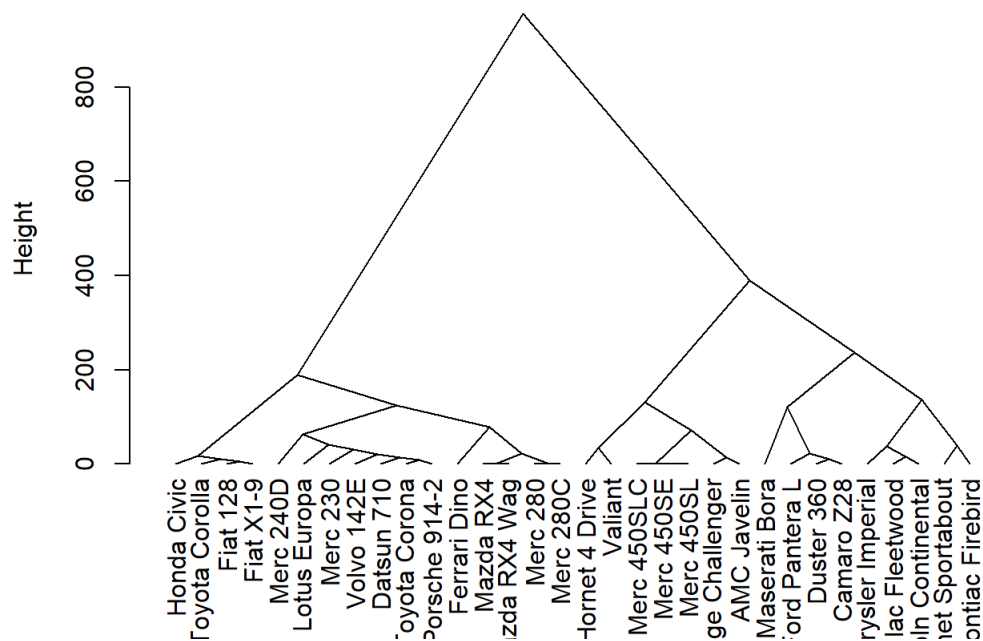
2.Display the hierarchical clustering results dendrogram and plot the result

```
# Convert hclust into a dendrogram and plot
hcd <- as.dendrogram(hc)
# Default plot
plot(hcd, type = "rectangle", ylab = "Height")
```



3. Construct the Triangle plot

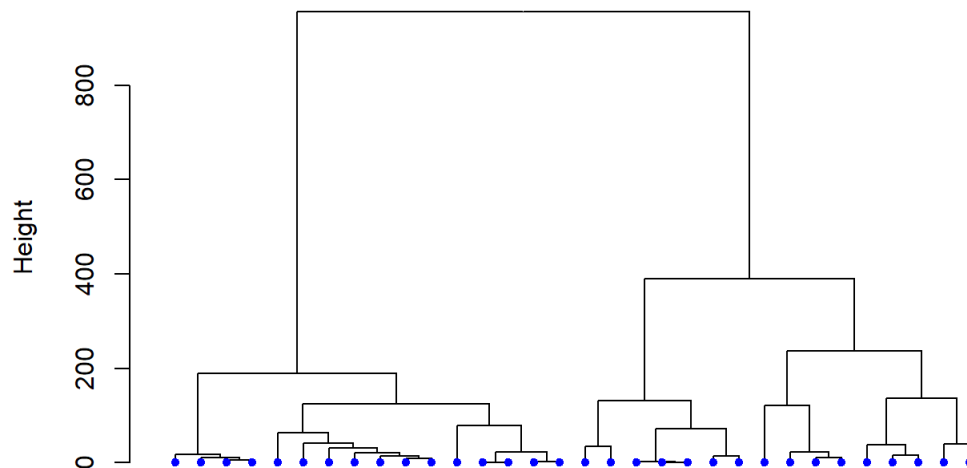
```
# Triangle plot
plot(hcd, type = "triangle", ylab = "Height")
```



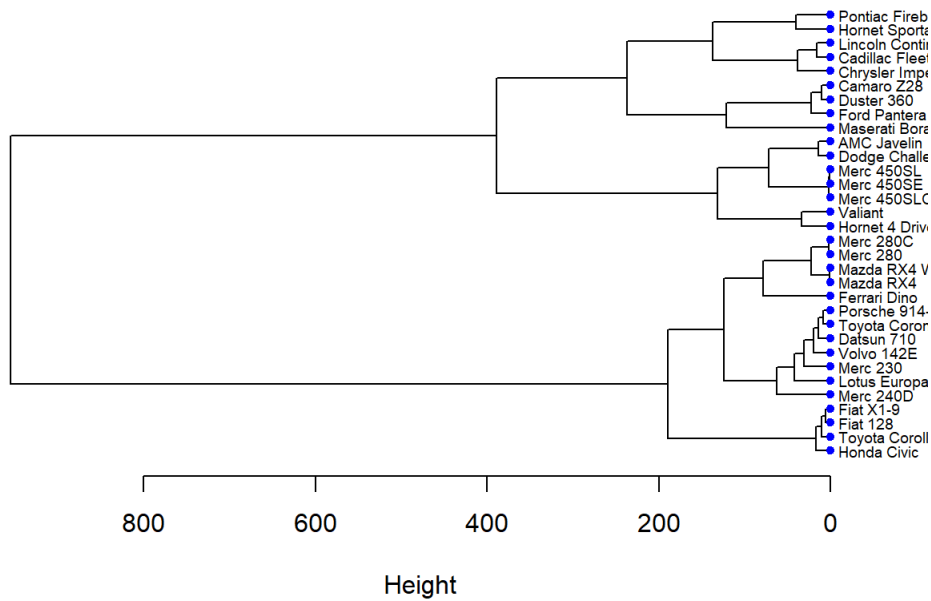
4. creation of nodePar and customizing the visualization

```
# Define nodePar
nodePar <- list(lab.cex = 0.6, pch = c(NA, 19),
               cex = 0.7, col = "blue")

# Customized plot; remove labels
plot(hcd, ylab = "Height", nodePar = nodePar, leaflab = "none")
```



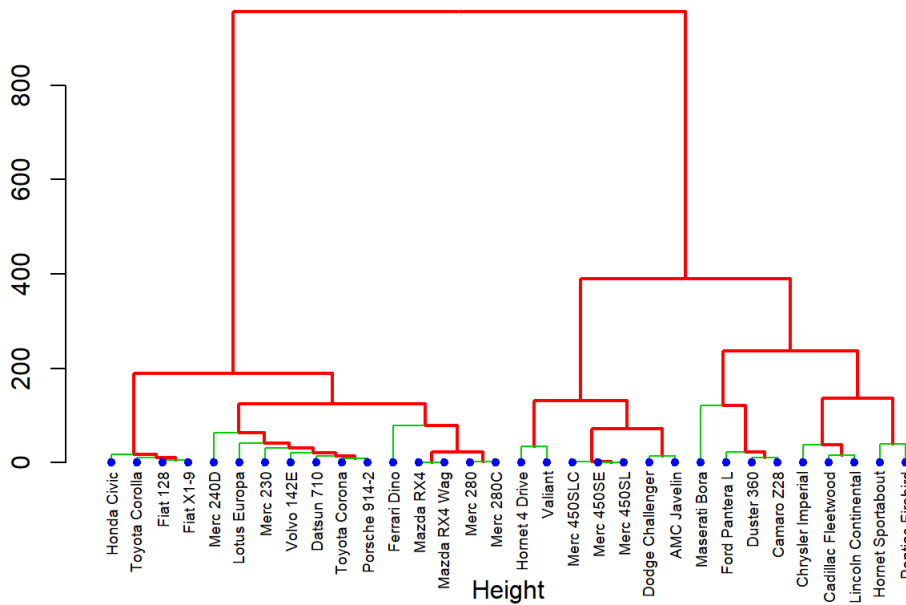
```
# Horizontal plot
plot(hcd, xlab = "Height",
     nodePar = nodePar, horiz = TRUE)
```



5. Change edge color based

on the clustering

```
# Change edge color
plot(hcd, xlab = "Height", nodePar = nodePar,
     edgePar = list(col = 2:3, lwd = 2:1))
```

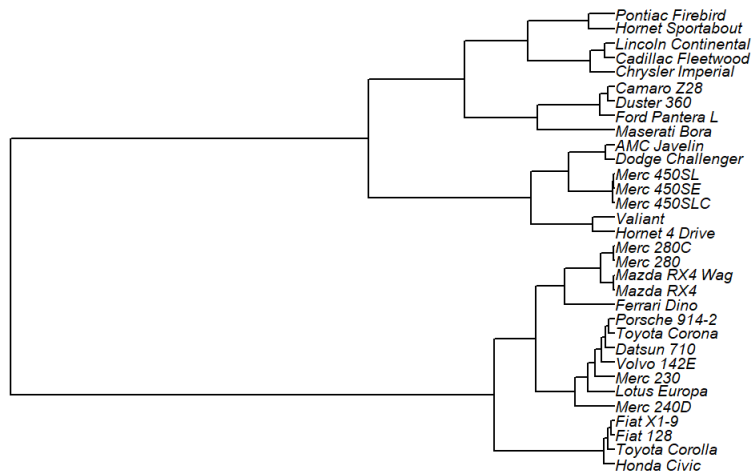


6. Display the cluster results using ape package

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

```
## Warning: package 'ape' was built under R version 3.3.3
```

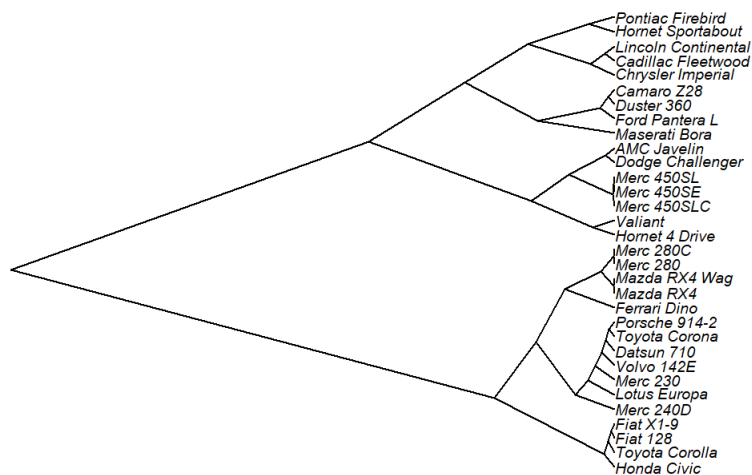
```
# Default plot
plot(as.phylo(hc), cex = 0.6, label.offset = 0.5)
```



## 7. Display the cluster using Cladogram.

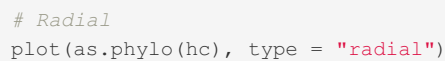
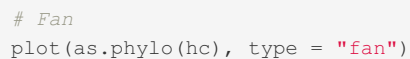
A cladogram is used by a scientist studying phylogenetic systematics to visualize the groups of objects being compared, how they are related, and their most common ancestors.

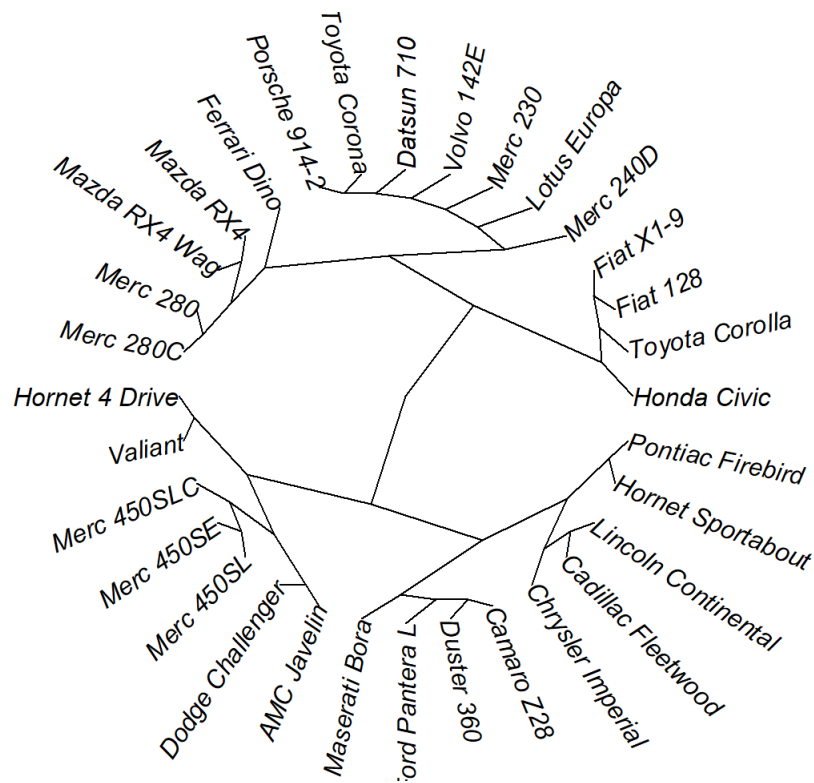
```
# Cladogram
plot(as.phylo(hc), type = "cladogram", cex = 0.6,
     label.offset = 0.5)
```



## 8. Display the cluster using unrooted,Fan, radial formats

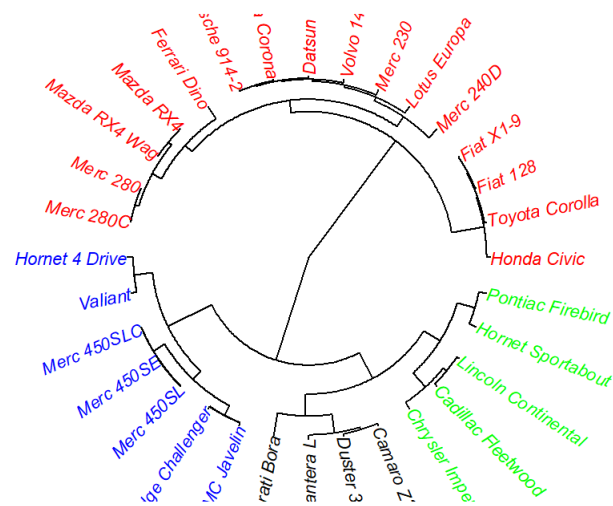
```
# Unrooted
plot(as.phylo(hc), type = "unrooted", cex = 0.6,
     no.margin = TRUE)
```





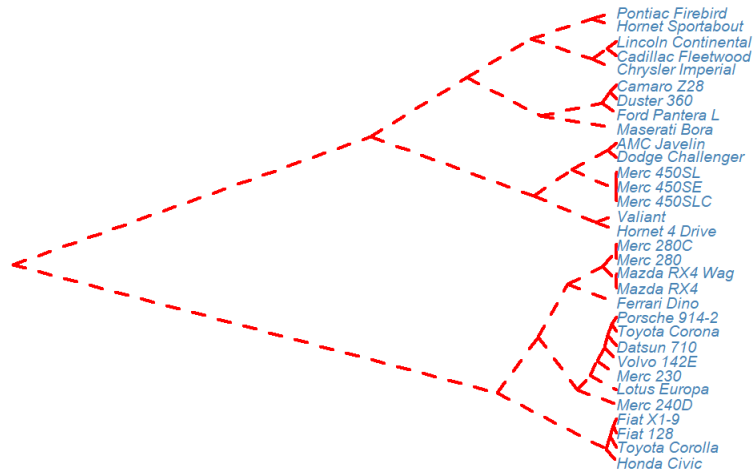
9. Cut and Plot the Cut the dendrogram into 4 clusters and color the clusters

```
# Cut the dendrogram into 4 clusters
colors = c("red", "blue", "green", "black")
clus4 = cutree(hc, 4)
plot(as.phylo(hc), type = "fan", tip.color = colors[clus4],
     label.offset = 1, cex = 0.7)
```



10. Change the appearance, edges and label of the cluster results

```
# Change the appearance
# change edge and label (tip)
plot(as.phylo(hc), type = "cladogram", cex = 0.6,
     edge.color = "red", edge.width = 2, edge.lty = 2,
     tip.color = "steelblue")
```



11. Install the package `ggdendro`, `ggplot2` and improve your cluster result and

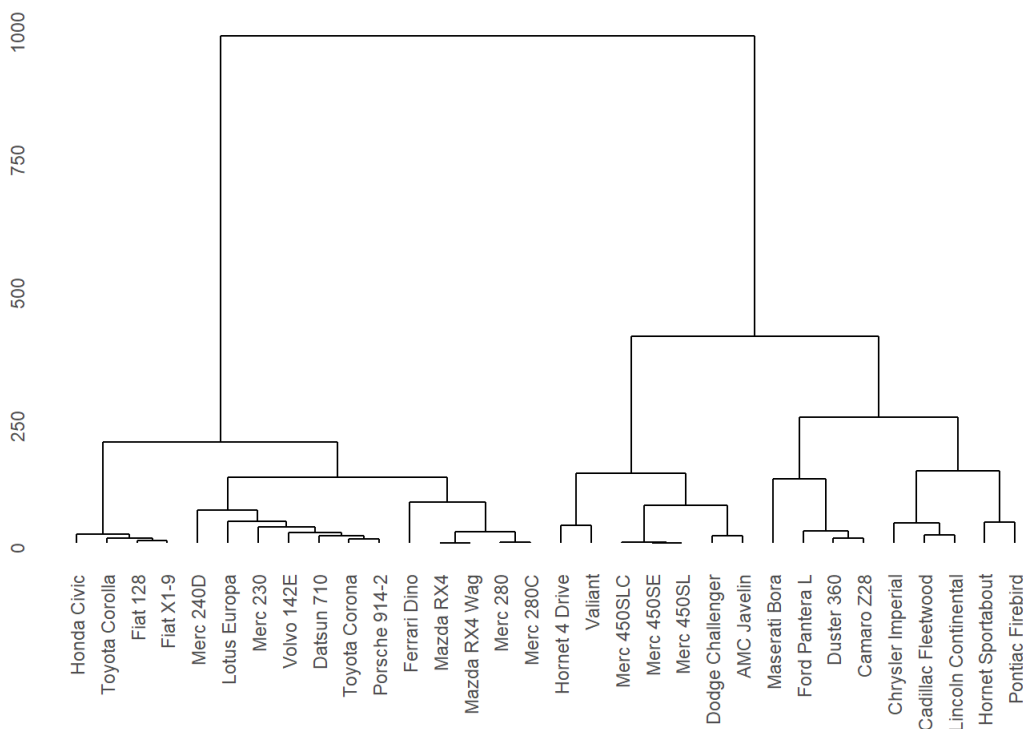
```
#install.packages("ggdendro")
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
library("ggdendro")
```

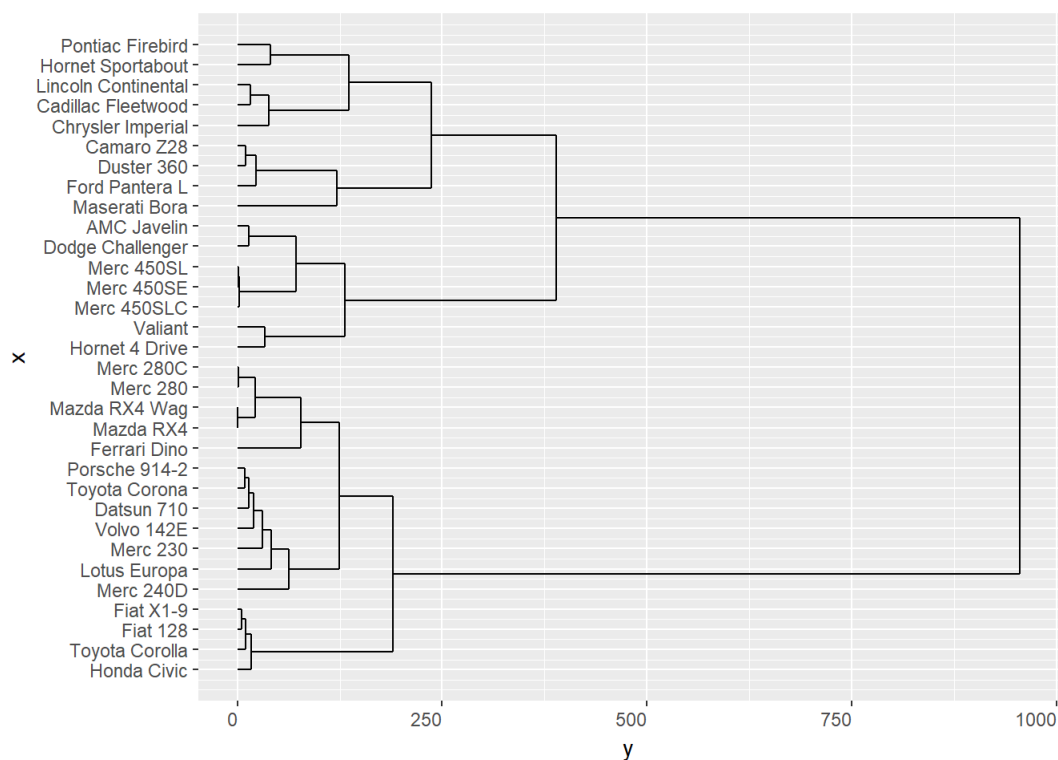
```
## Warning: package 'ggdendro' was built under R version 3.3.3
```

```
# Visualization using the default theme named theme_dendro()
ggdendrogram(hc)
```





```
# Rotate the plot and remove default theme
ggdendrogram(hc, rotate = TRUE, theme_dendro = FALSE)
```



## 12. Plot the clusterresults using tangrogram

```
#install(dendextend)
library(dendextend)
```

```
##
## -----
## Welcome to dendextend version 1.13.2
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))
## -----
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:ggdendro':
##
##   theme_dendro
```

```
## The following objects are masked from 'package:ape':
##
##   ladderize, rotate
```

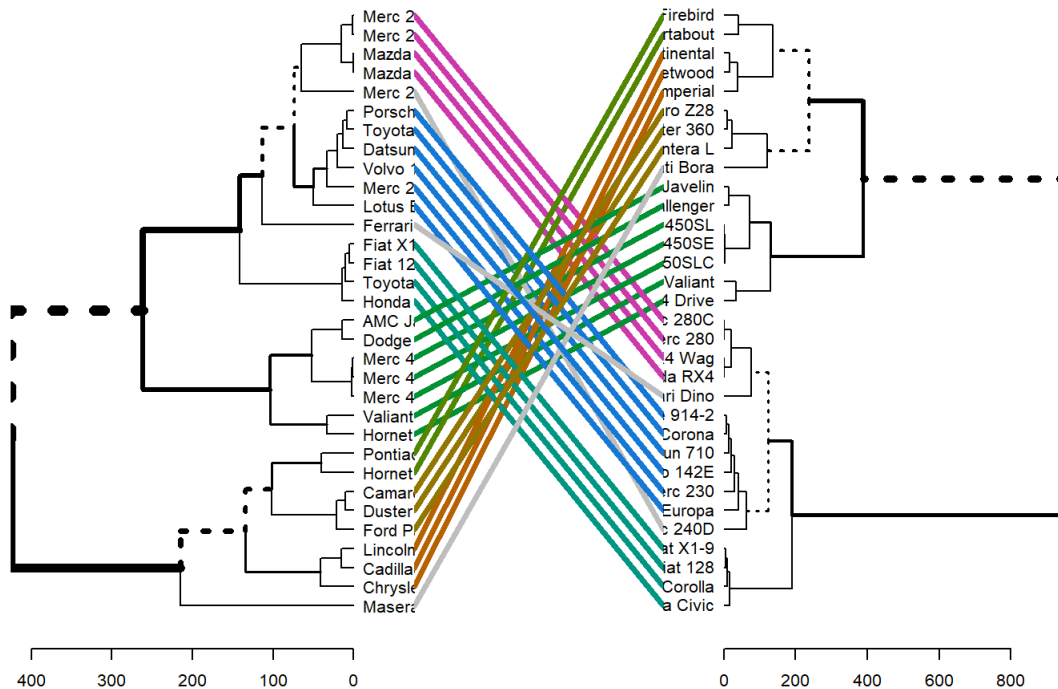
```
## The following object is masked from 'package:stats':
##
##   cutree
```

```

hc1 <- hclust(dd, method = "complete")
hc2 <- hclust(dd, method = "ward.D2")
# Create two dendrograms
dend1 <- as.dendrogram (hc1)
dend2 <- as.dendrogram (hc2)

tanglegram(dend1,dend2)

```



Inference: The output displays “unique” nodes, with a combination of labels/items not present in the other tree, highlighted with dashed lines.

Conclusion: A variety of functions exists in R for visualizing and customizing dendrogram and used in this program