

## Section 8: Miscellaneous topics

---

Review of important functions covered in this section:

### Functions

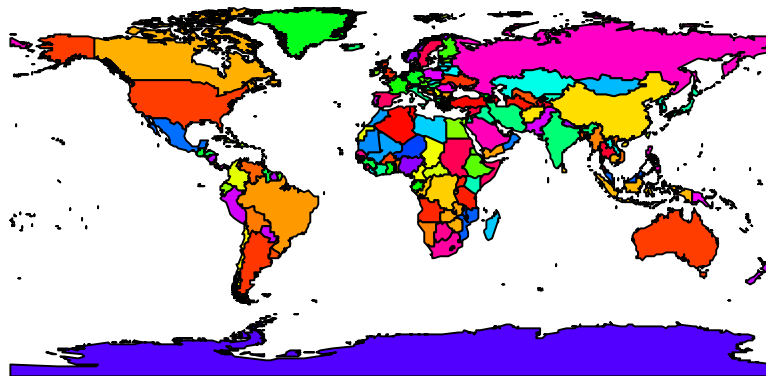
map	Draws geographical maps (maps package)
map.text	Draws geographical maps with labels (maps package)
identify	Identify chosen points in a plot
smooth.map	Smooths aggregated data (maps package)
ibar	Interactive bar chart (iplots package)
ihist	Interactive histogram (iplots package)
iplot	Interactive plot (iplots package)
iabline	Interactive plot line (iplots package)
itext	Interactive text in a plot (iplots package)
iplot.data	Interactive plot data (iplots package)
ipcp	Interactive parallel coordinates plot (iplots package)
require(Rcmdr)	Loads R commander
hclust	Hierarchical clustering (cluster package)
rect.hclust	Draw rectangles around hierarchical clusters (cluster package)
agnes	Agglomerative nesting hierarchical clustering (cluster package)
pltree	Draws clustering trees (cluster package)
as.dendogram	Converts to dendogram (cluster package)
plot	Plots dendogram objects (cluster package)
pvclust	Calculates p-values for hierarchical clustering (pvclust package)
seplot	Shows diagnostic plots for s.e. of p-value (pvclust package)

## Exercise

We will use `mtcars` and some other map data in this section.

- a. Using `maptools` package, draw a colorful world map.

```
install.packages("maptools")  
  
require(maptools)  
  
plot(wrld_simpl, col = rainbow(200))
```



- b. Using maps package, draw the map of China.

```
install.packages("maps")  
require(maps)  
map("world", "China")
```

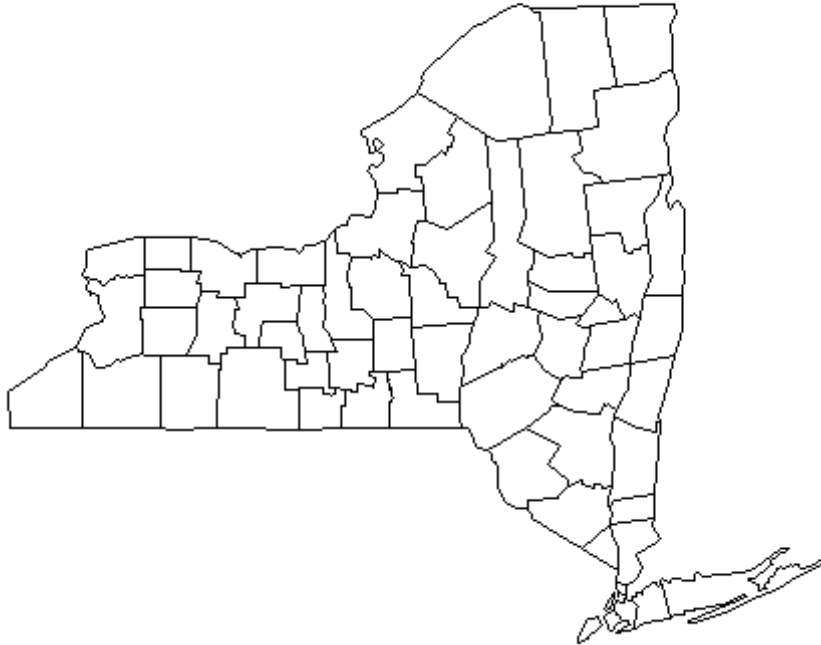


- c. From the world map, how would you identify which country you are pointing at?

```
identify(map("world", fill = TRUE, col = "lightblue"))  
  
# then click in anywhere on the map
```

d. How would you draw map of New York county?

```
map("county", "new york")
```



e. Compute a distance matrix using `mtcars` data.

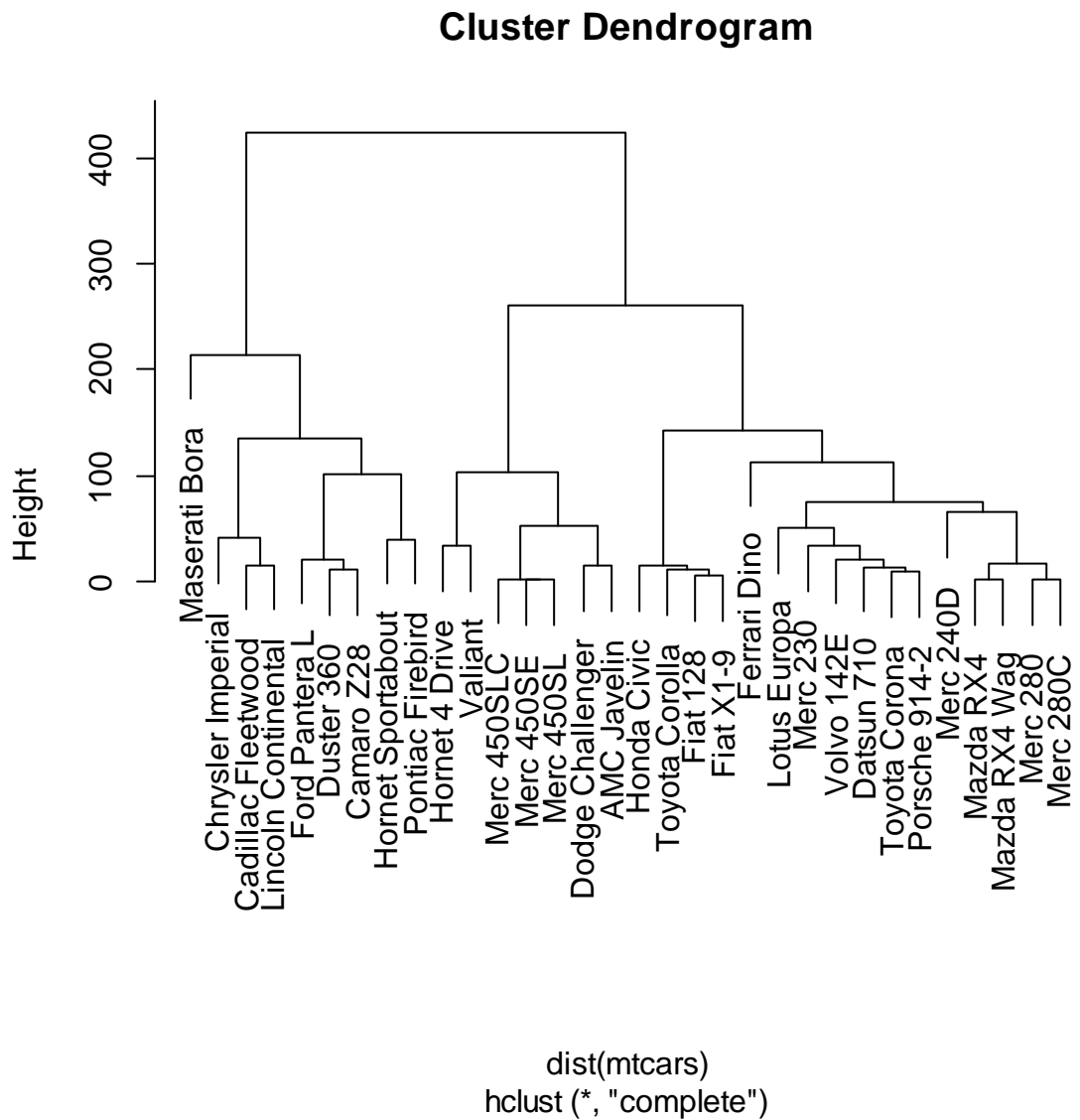
```
dist.mat = dist(mtcars)
```

f. Perform Hierarchical clustering on this distance matrix.

```
hcl = hclust(dist(mtcars), method = "complete")
```

g. Plot the dendrogram.

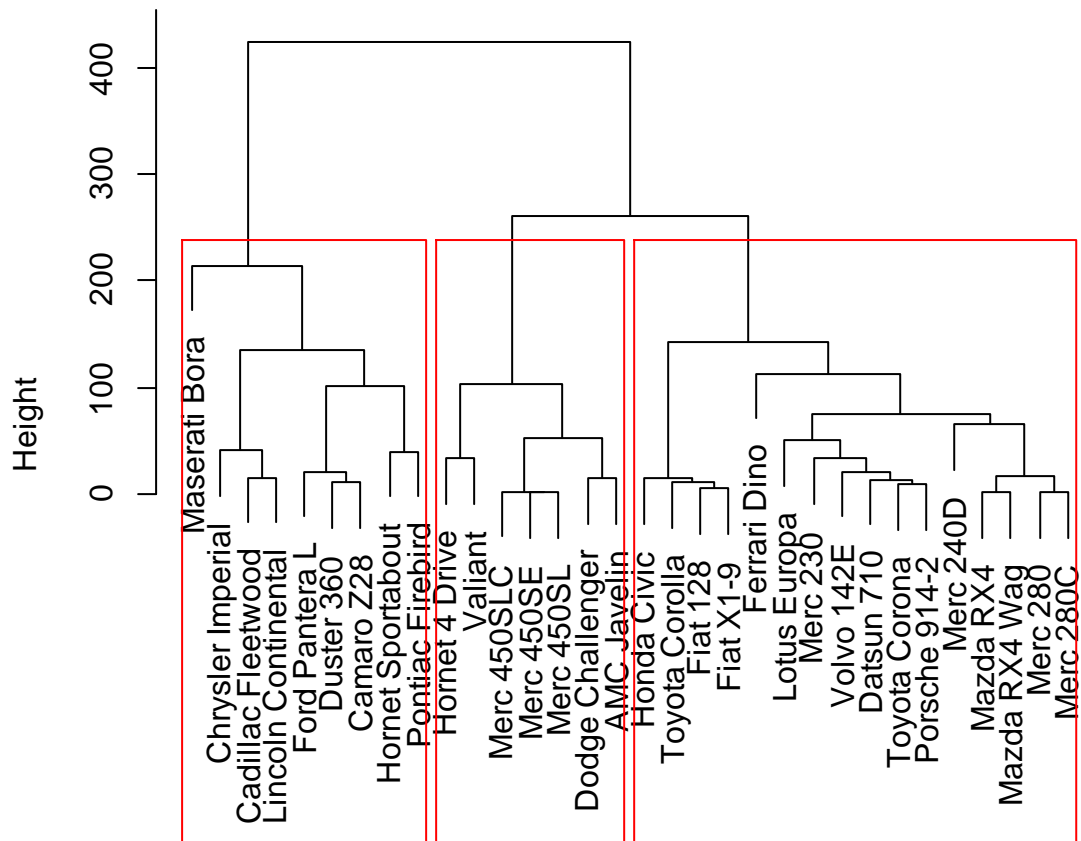
```
plot(hcl)
```



- h. Apply rectangles around groups created based on data.

```
rect.hclust(hcl, k = 4, border = "red")
```

## Cluster Dendrogram



```
dist(mtcars)
hclust (*, "complete")
```

- i. Compute agglomerative hierarchical clustering of the mtcars dataset.

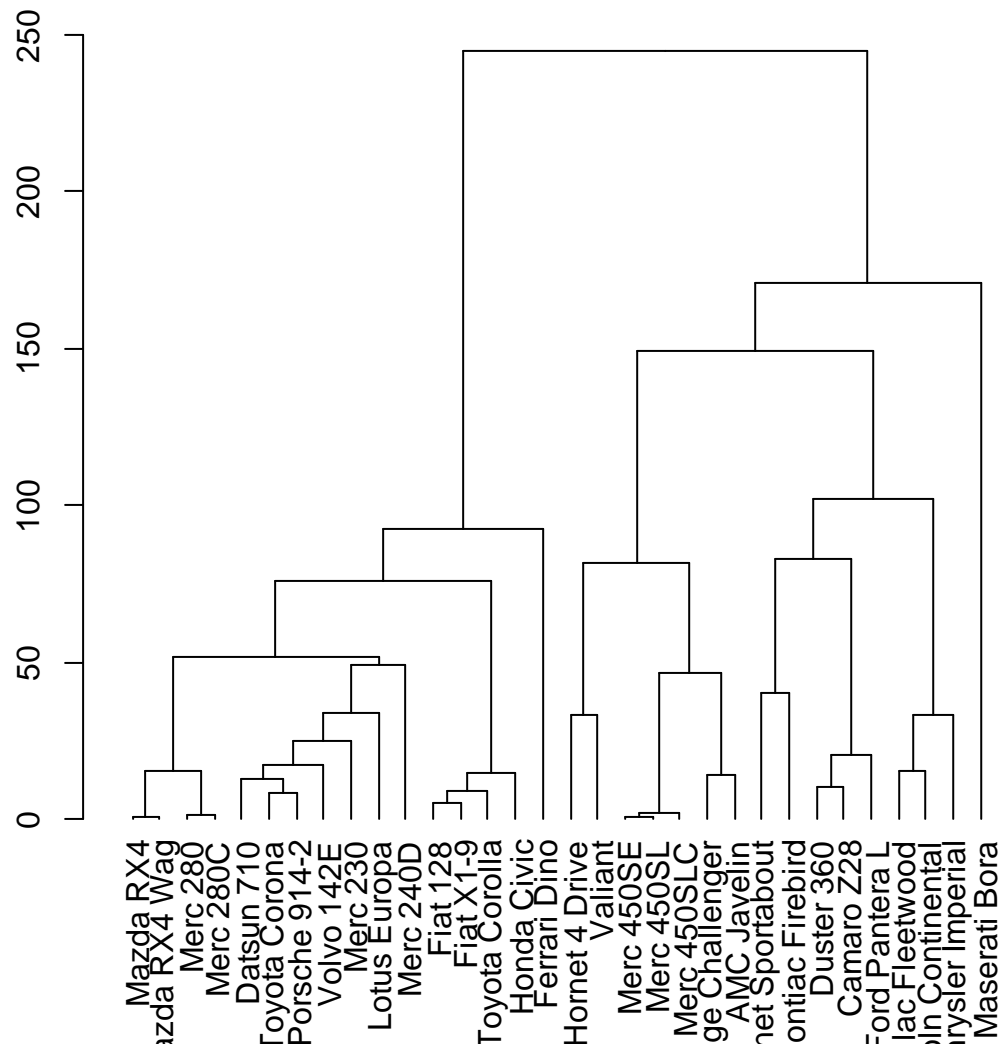
```
install.packages("cluster")

require(cluster)

cluster = agnes(mtcars)
```

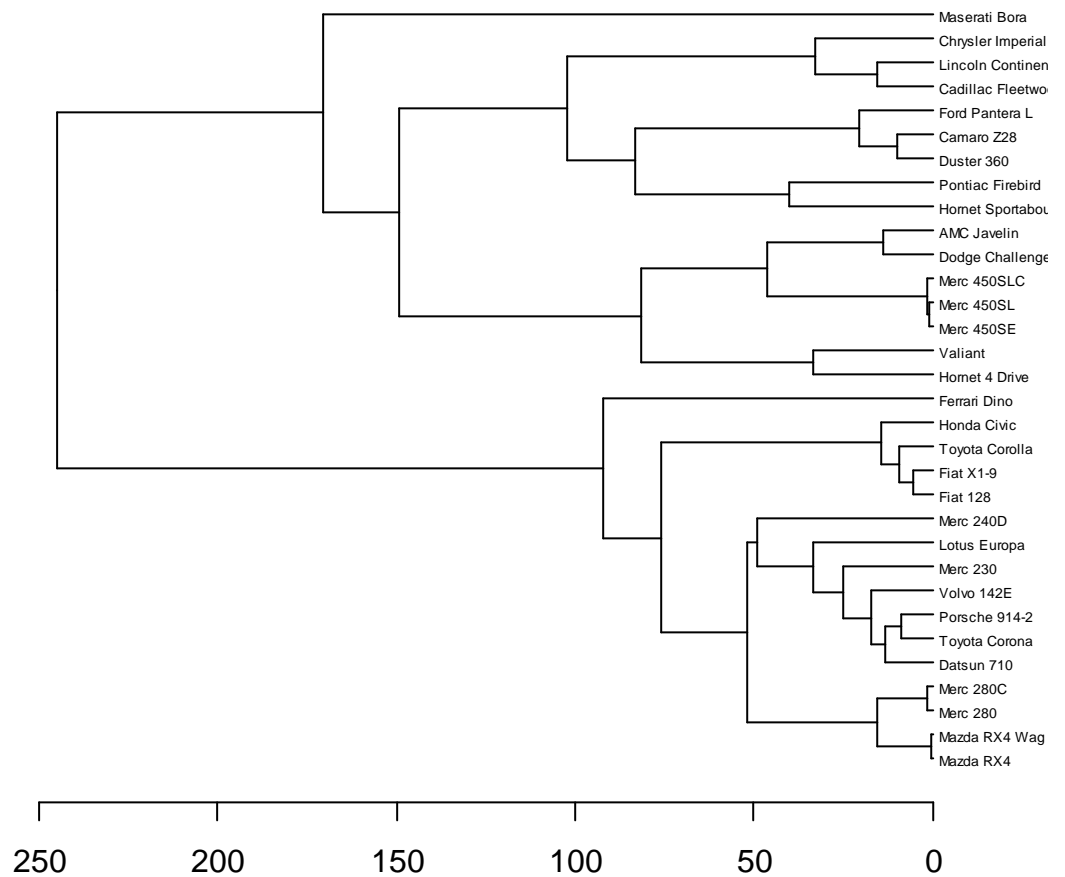
j. Draw a dendrogram of the above clustering.

```
dend = as.dendrogram(as.hclust(cluster))  
plot(dend)
```



k. Draw the same dendrogram horizontally.

```
plot(dend, horiz = TRUE, nodePar = list(lab.cex = 0.4, pch =  
""))
```





- ```
install.packages("pvclust")

require(pvclust)

fit = pvclust(mtcars, method.hclust="ward",
method.dist="euclidean", nboot = 50)

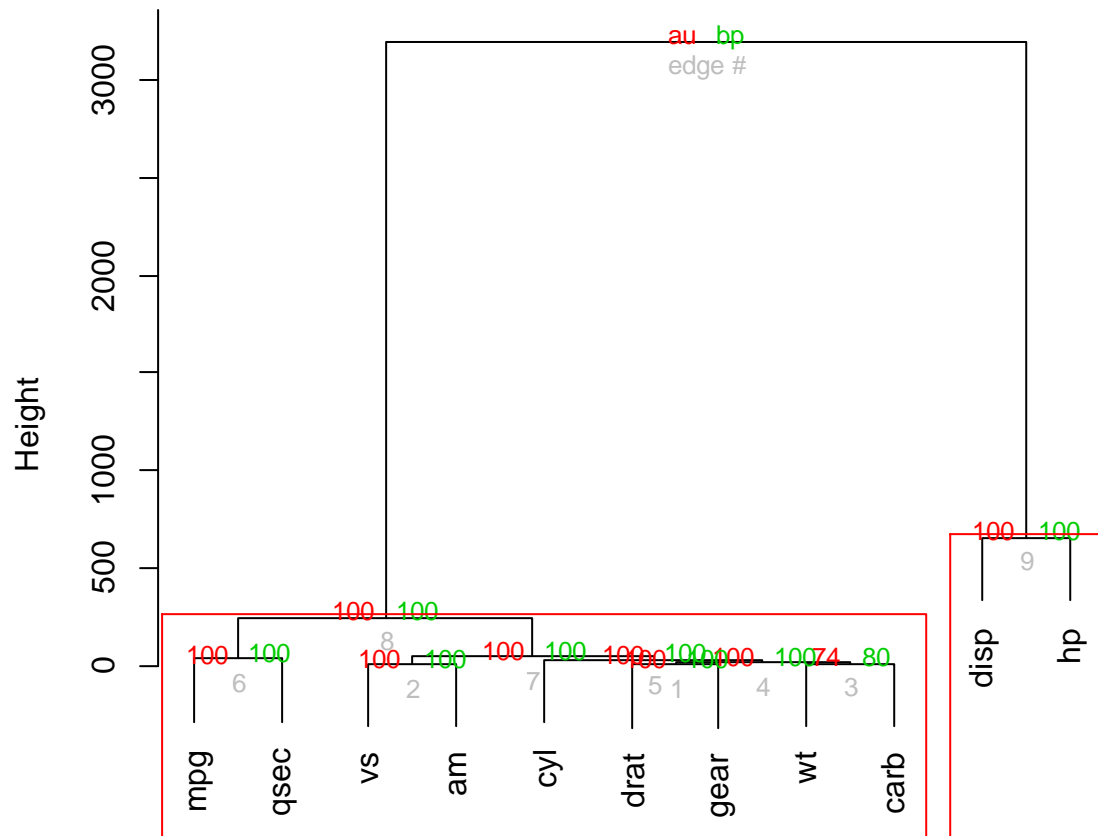
plot(fit)
```

Distance: euclidean  
Cluster method: ward

m. Add rectangles around groups highly supported by the mtcars data.

```
pvrect(fit, alpha=.95)
```

### Cluster dendrogram with AU/BP values (%)



Distance: euclidean  
Cluster method: ward

n. Draw diagnostic plot for SE of p-value obtained above.

```
seplot(fit)
```

