Cluster Analysis

Setia Pramana

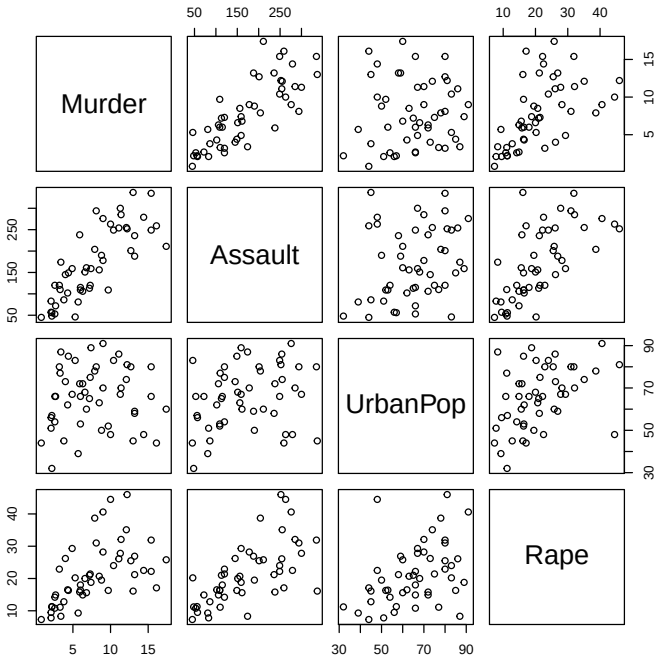
April 11, 2016

1 Data Preparation and Visualitation

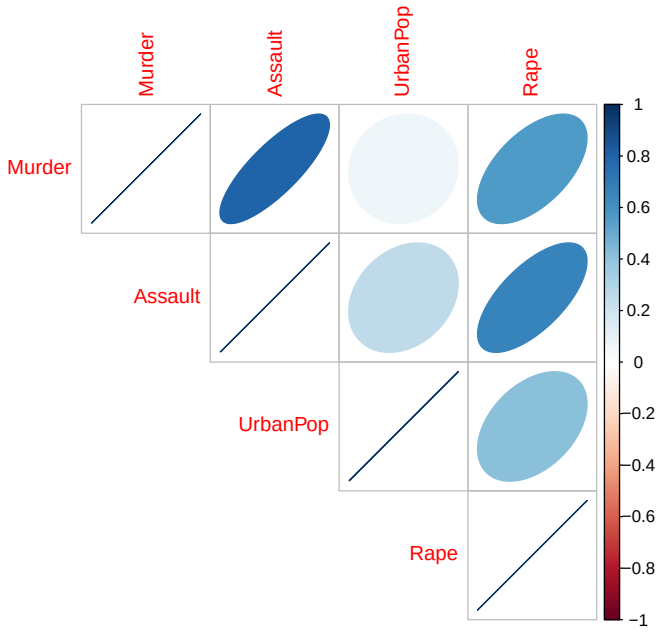
```
> library(corrplot)
> data(USArrests)
> head(USArrests)
```

```
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
```

```
> pairs(USArrests)
```

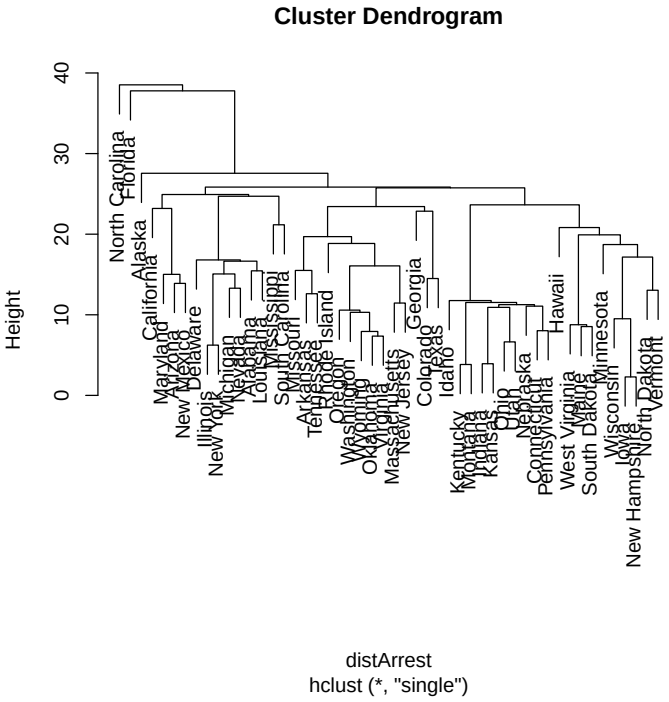



```
> corrplot(cor(USArrests), method = "ellipse", type = "upper")
>
```

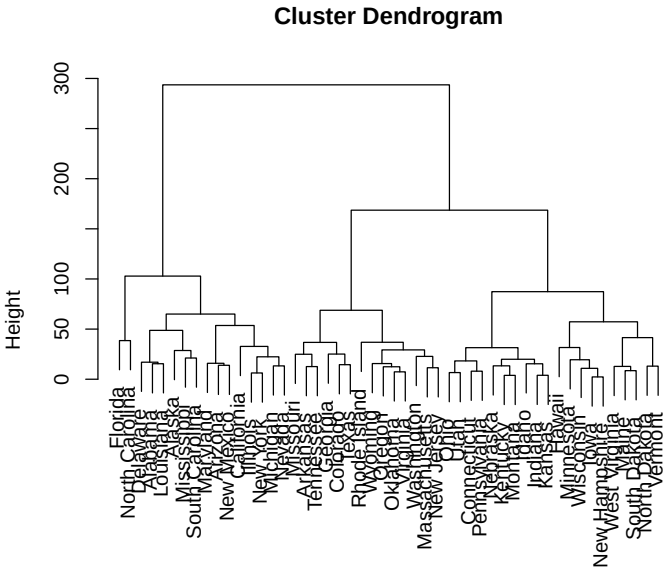


2 Hierachical CLustering

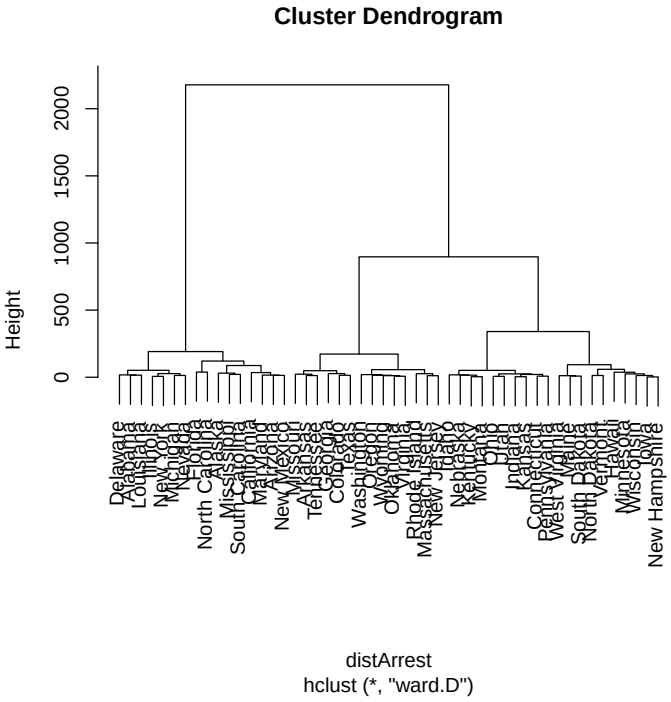
```
> ################################
> ### Hierachical CLustering  ##
> ################################
>
>
>
> distArrest <- dist (USArrests)
> ### Distance Methods Could be:  ##
> ### "euclidean", "maximum", "manhattan", "canberra" ##
> ### "binary" or "minkowski" ##
>
>
> # Clustering
>
> plot(res1 <- hclust(distArrest , method="single")  )
> ## method :  "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (=
>
>
```



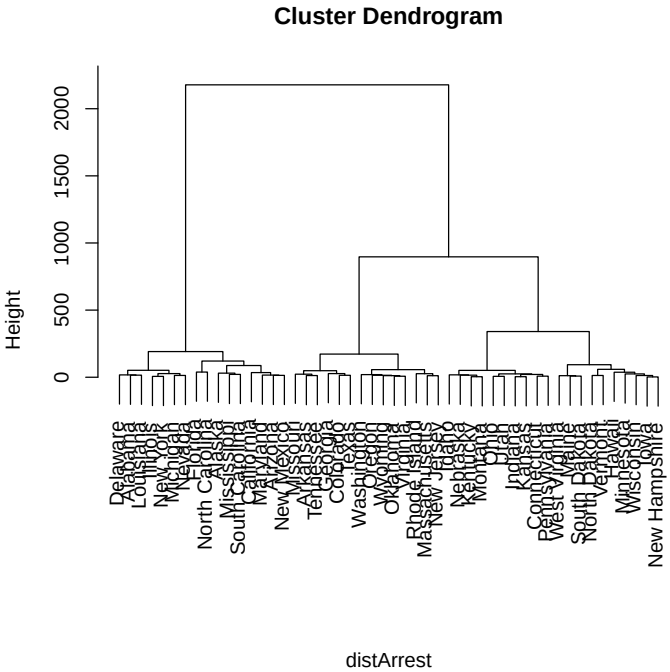

```
> ### max DIstance
> plot(res2 <- hclust(distArrest , method="complete")  )
>
```



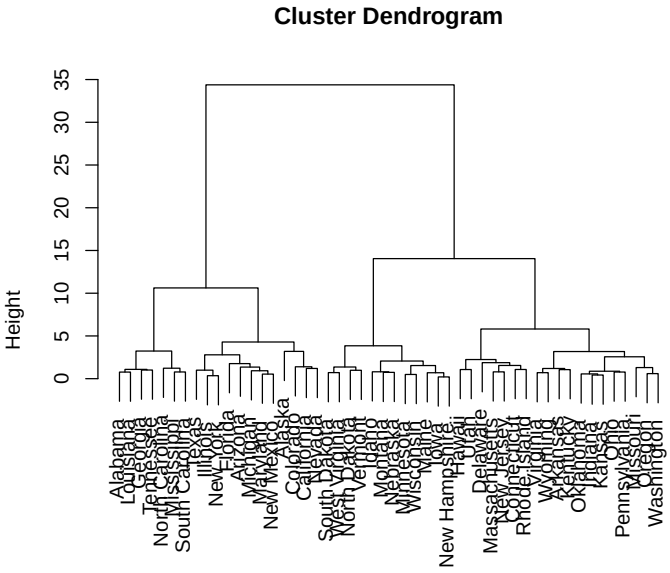

```
> ## Average Distance ##
>
> plot(res3 <- hclust(distArrest , method="average")  )
```




```
> ## Average Distance ##
>
> plot(res3.a <- hclust(distArrest , method="ward.D")  )
```




```
> ## Rescaling the data ###
>
> USArrScale <- scale (USArrests, scale=T)
> distArrSc <- dist (USArrScale )
> plot(res4 <- hclust(distArrSc , method="ward.D")  )
>
```

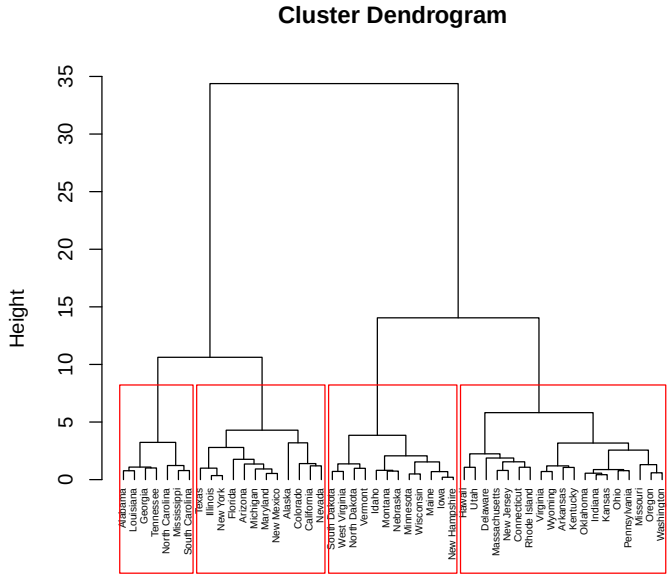



```
> plot(res4, hang = -1, cex = 0.5)
> ## rectangle for 4 clusters ##
>
> rect.hclust(res4, k = 4)
> ## the clustering member
> cutree(res4, h = 4)
```

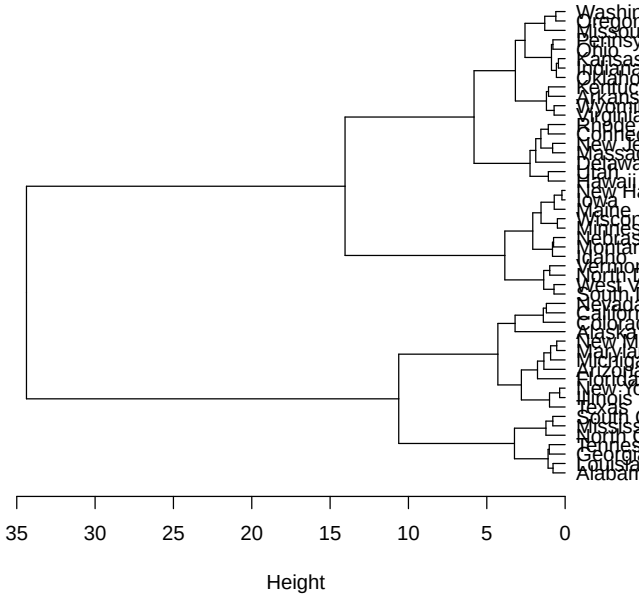
Alabama	Alaska	Arizona	Arkansas	California
1	2	3	4	2
Colorado	Connecticut	Delaware	Florida	Georgia
2	5	5	3	1
Hawaii	Idaho	Illinois	Indiana	Iowa
5	6	3	4	6
Kansas	Kentucky	Louisiana	Maine	Maryland
4	4	1	6	3
Massachusetts	Michigan	Minnesota	Mississippi	Missouri
5	3	6	1	4
Montana	Nebraska	Nevada	New Hampshire	New Jersey
6	6	2	6	5
New Mexico	New York	North Carolina	North Dakota	Ohio
3	3	1	6	4
Oklahoma	Oregon	Pennsylvania	Rhode Island	South Carolina

```
             4              4              4              5              1
South Dakota      Tennessee          Texas           Utah        Vermont
             6              1              3              5              6
      Virginia     Washington  West Virginia      Wisconsin        Wyoming
             4              4              6              6              4
```

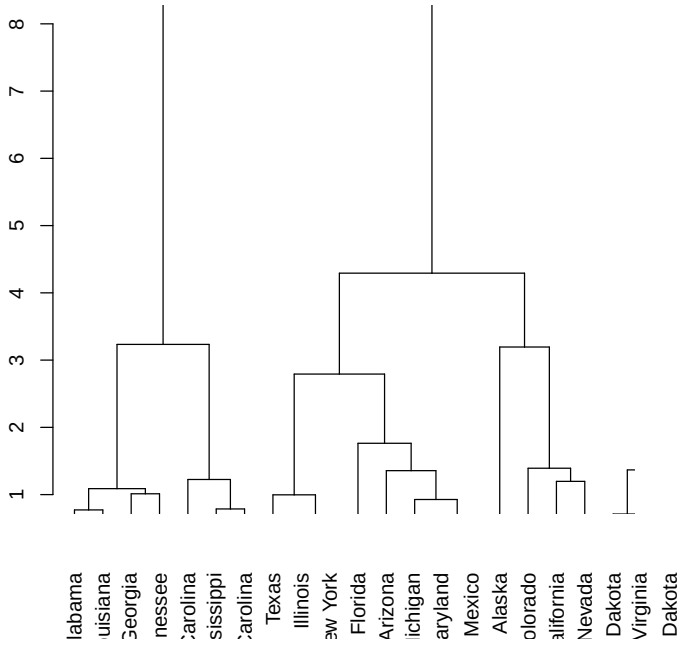
>




```
> # Convert hclust into a dendrogram and plot
> res4.a <- as.dendrogram(res4)
> # Horizontal plot
> plot(res4.a ,  xlab = "Height", horiz = TRUE)
```




```
> # Zoom in to the first dendrogram
> plot(res4.a, xlim = c(1, 20), ylim = c(1,8))
>
>
```



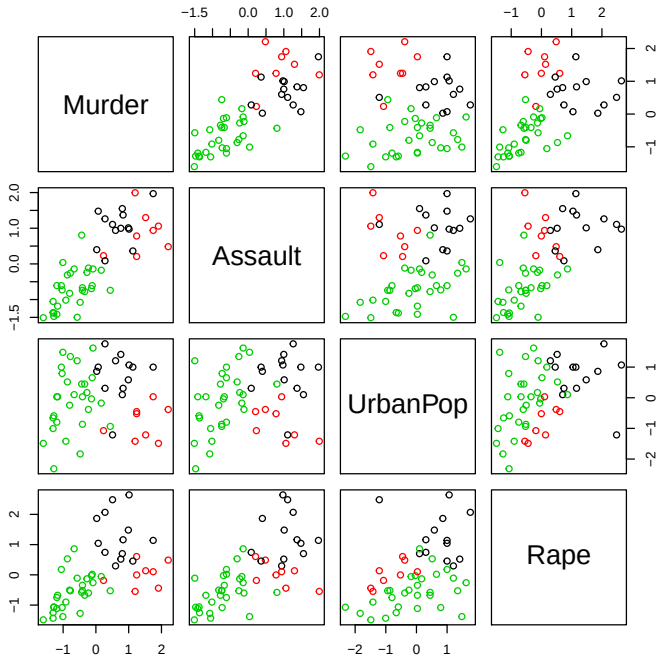
3 K-Means CLustering

```
> ###############
> ### K Means ###
> ###############
>
>
> kMres <- kmeans(USArrScale , centers=3)
> kMres $cluster
```

```
       Alabama         Alaska        Arizona       Arkansas     California
             2              1              1              2              1
      Colorado    Connecticut       Delaware        Florida        Georgia
             1              3              3              1              2
        Hawaii          Idaho       Illinois        Indiana           Iowa
             3              3              1              3              3
        Kansas       Kentucky      Louisiana          Maine       Maryland
             3              3              2              3              1
 Massachusetts       Michigan      Minnesota    Mississippi       Missouri
             3              1              3              2              1
       Montana       Nebraska         Nevada  New Hampshire     New Jersey
```

```
          3           3           1           3           3
 New Mexico    New York North Carolina North Dakota        Ohio
          1           1           2           3           3
   Oklahoma      Oregon Pennsylvania Rhode Island South Carolina
          3           3           3           3           2
South Dakota   Tennessee       Texas        Utah     Vermont
          3           2           1           3           3
   Virginia  Washington West Virginia   Wisconsin     Wyoming
          3           3           3           3           3
```

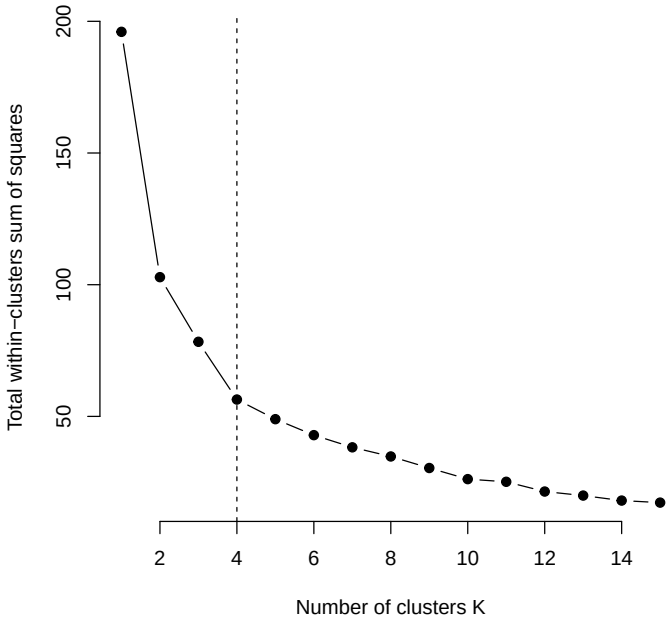
```
> #install.packages("fpc")
>
> pairs(USArrScale , col=c(1:3)[kMres $cluster])
>
```




```
> ## different starting centroids ##
> ## give slightly different result ##
>
> set.seed(12)
> kMres1 <- kmeans(USArrScale , centers=4)
> head(kMres1 $cluster)
```

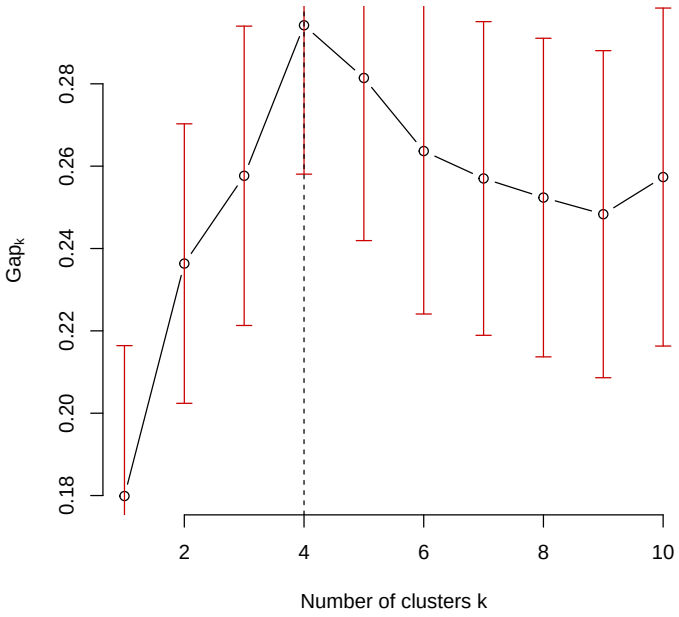
```
   Alabama     Alaska    Arizona   Arkansas California   Colorado
         1          4          4          1          4          4
```

```
> kMres2 <- kmeans(USArrScale , centers=4)
> head(kMres2 $cluster)

   Alabama     Alaska    Arizona   Arkansas California   Colorado
         4          2          1          4          1          1

> table(kMres1 $cluster,kMres2 $cluster)

     1  2  3  4
  1  0  0  0  8
  2  0  0 13  0
  3  0  0 16  0
  4 12  1  0  0

>
>
>

> set.seed(12)
> # Compute and plot wss for k = 2 to k = 15
> k.max <- 15 # Maximal number of clusters
> ## get the total within-cluster variation for each k ##
> wss <- sapply(1:k.max,
+         function(k){kmeans(USArrScale, k, nstart=10 )$tot.withinss})
> plot(1:k.max, wss,
+        type="b", pch = 19, frame = FALSE,
+        xlab="Number of clusters K",
+        ylab="Total within-clusters sum of squares")
> abline(v = 4, lty =2)
>
```

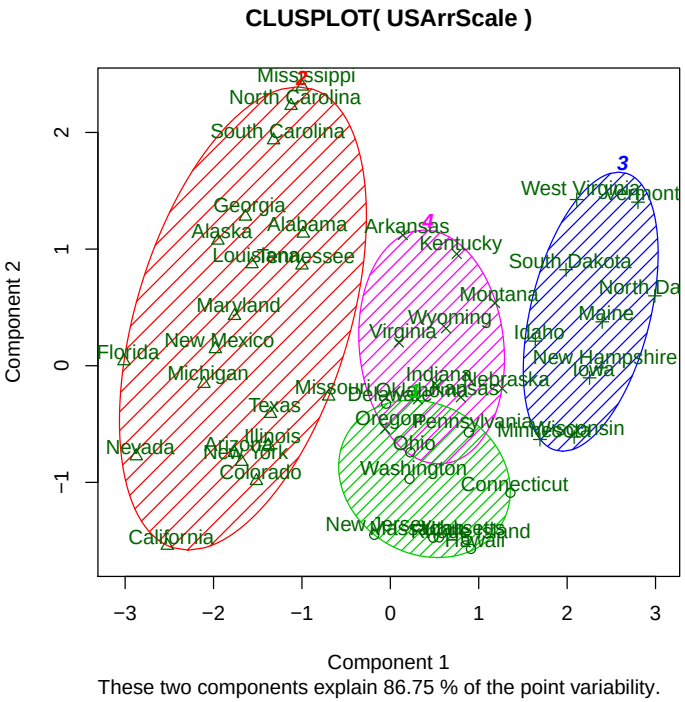



```
> # Compute gap statistic
> library(cluster)
> set.seed(123)
> gap_stat <- clusGap(USArrScale, FUN = kmeans, nstart = 25,
+                     K.max = 10, B = 50)
> # Print the result
> print(gap_stat, method = "firstmax")

Clustering Gap statistic ["clusGap"].
B=50 simulated reference sets, k = 1..10
 --> Number of clusters (method 'firstmax'): 4
          logW   E.logW       gap     SE.sim
 [1,] 3.458369 3.638250 0.1798804 0.03653200
 [2,] 3.135112 3.371452 0.2363409 0.03394132
 [3,] 2.977727 3.235385 0.2576588 0.03635372
 [4,] 2.826221 3.120441 0.2942199 0.03615597
 [5,] 2.738868 3.020288 0.2814197 0.03950085
 [6,] 2.669860 2.933533 0.2636730 0.03957994
 [7,] 2.598748 2.855759 0.2570109 0.03809451
 [8,] 2.531626 2.784000 0.2523744 0.03869283
 [9,] 2.468162 2.716498 0.2483355 0.03971815
[10,] 2.394884 2.652241 0.2573567 0.04104674
```

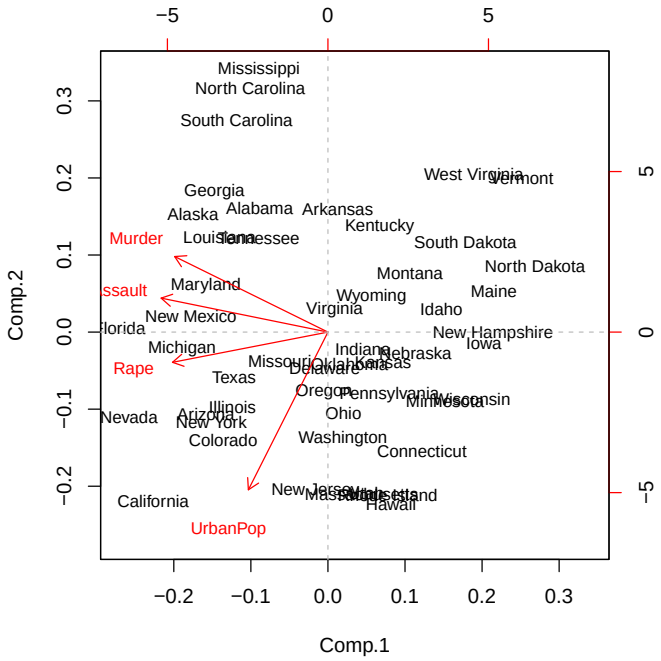
```
> # Base plot of gap statistic
> plot(gap_stat, frame = FALSE, xlab = "Number of clusters k")
> abline(v = 4, lty = 2)
>
```




```
> # Cluster Plot against 1st 2 principal components
> kMres2 <- kmeans(USArrScale , centers=4)
> library(cluster)
> clusplot(USArrScale, kMres2$cluster, color=TRUE, shade=TRUE,
+           labels=2, lines=0)
>
>
```




```
> pcares <- princomp(USArrScale, cor=T)
> biplot(pcares,cex=0.8)
> abline(h = 0, v = 0, lty = 2, col = 8)
>
```

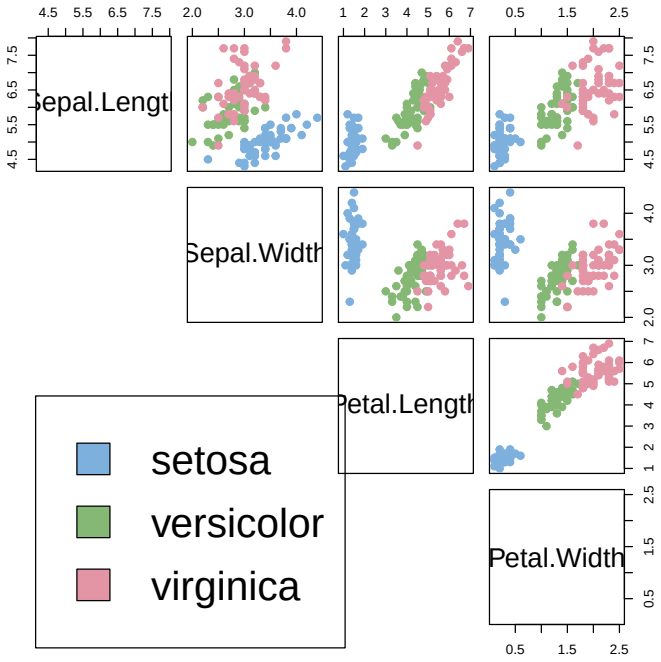


4 More Examples

4.1 Iris Dataset

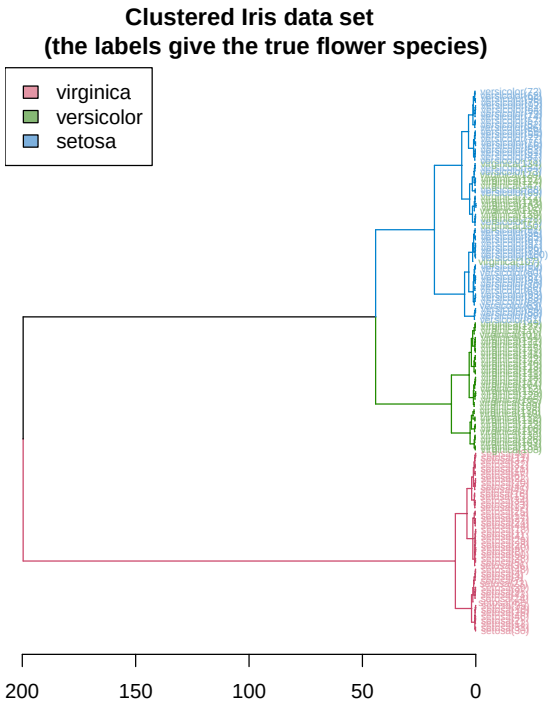
We use Iris Data as example:

```
> data(iris)
> iris2 <- iris[,-5]
> species_labels <- iris[,5]
> library(colorspace) # get nice colors
> species_col <- rev(rainbow_hcl(3))[as.numeric(species_labels)]
> # Plot a SPLOM:
> pairs(iris2, col = species_col,
+       lower.panel = NULL,
+        cex.labels=2, pch=19, cex = 1.2)
> # Add a legend
> par(xpd = TRUE)
> legend(x = 0.05, y = 0.4, cex = 2,
+    legend = as.character(levels(species_labels)),
+     fill = unique(species_col))
> par(xpd = NA)
>
```




```
> d_iris <- dist(iris2)
> #hc_iris <- hclust(d_iris, method = "complete")
> hc_iris <- hclust(d_iris, method = "ward.D")
> iris_species <- rev(levels(iris[,5]))
> #install.packages("dendextend")
> library(dendextend)
> dend <- as.dendrogram(hc_iris)
> # order it the closest we can to the order of the observations:
> dend <- rotate(dend, 1:150)
> # Color the branches based on the clusters:
> dend <- color_branches(dend, k=3) #, groupLabels=iris_species)
> # Manually match the labels, as much as possible, to the real classification of the flower
> labels_colors(dend) <-
+    rainbow_hcl(3)[sort_levels_values(
+       as.numeric(iris[,5])[order.dendrogram(dend)]
+    )]
> # We shall add the flower type to the labels:
> labels(dend) <- paste(as.character(iris[,5])[order.dendrogram(dend)],
+                            "(",labels(dend),")",
+                            sep = "")
> # We hang the dendrogram a bit:
> dend <- hang.dendrogram(dend,hang_height=0.1)
```

```
> # reduce the size of the labels:
> # dend <- assign_values_to_leaves_nodePar(dend, 0.5, "lab.cex")
> dend <- set(dend, "labels_cex", 0.5)
> # And plot:
> par(mar = c(3,3,3,7))
> plot(dend,
+      main = "Clustered Iris data set
+      (the labels give the true flower species)",
+      horiz =  TRUE,  nodePar = list(cex = .007))
> legend("topleft", legend = iris_species, fill = rainbow_hcl(3))
>
```

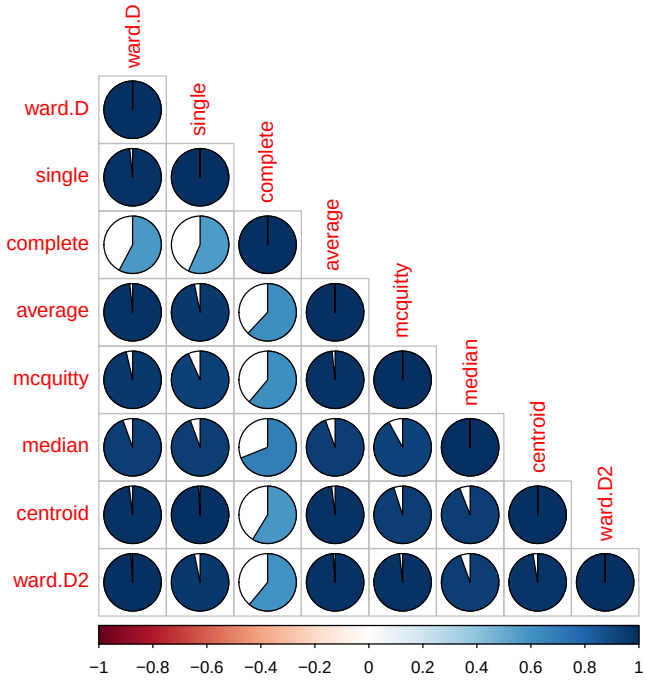


The same can be presented in a circular layout:

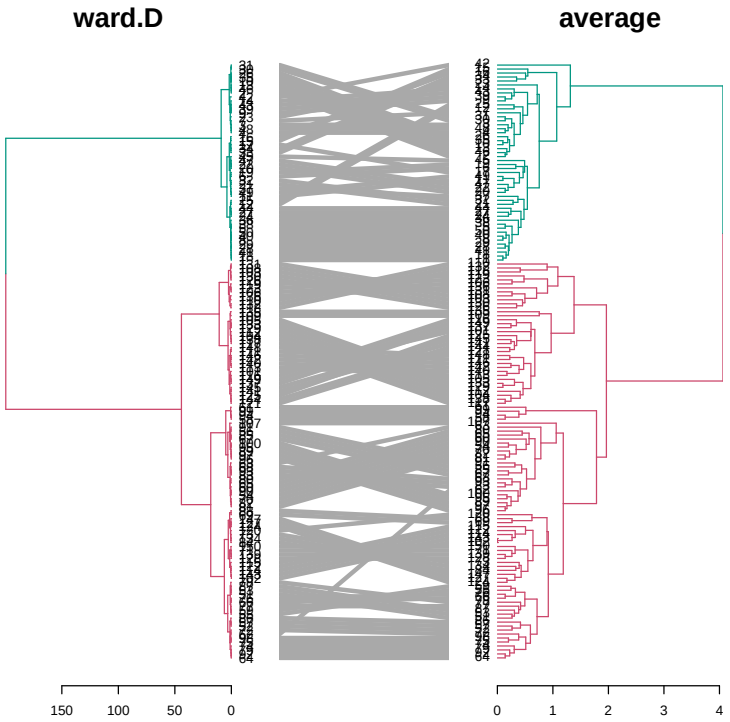
```
> # install.packages('circlize')
>
> # Requires that the circlize package will be installed
> require(circlize)
> par(mar = rep(0,4))
> circlize_dendrogram(dend)
```

```
> hclust_methods <- c("ward.D", "single", "complete", "average", "mcquitty",
+         "median", "centroid", "ward.D2")
> iris_dendlist <- dendlist()
> for(i in seq_along(hclust_methods)) {
+    hc_iris <- hclust(d_iris, method = hclust_methods[i])
+    iris_dendlist <- dendlist(iris_dendlist, as.dendrogram(hc_iris))
+ }
> names(iris_dendlist) <- hclust_methods
> iris_dendlist

$ward.D
'dendrogram' with 2 branches and 150 members total, at height 199.6205

$single
'dendrogram' with 2 branches and 150 members total, at height 1.640122

$complete
'dendrogram' with 2 branches and 150 members total, at height 7.085196

$average
'dendrogram' with 2 branches and 150 members total, at height 4.062683
```

```
$mcquitty
'dendrogram' with 2 branches and 150 members total, at height 4.497283

$median
'dendrogram' with 2 branches and 150 members total, at height 2.82744

$centroid
'dendrogram' with 2 branches and 150 members total, at height 2.994307

$ward.D2
'dendrogram' with 2 branches and 150 members total, at height 32.44761

attr(,"class")
[1] "dendlist"

> iris_dendlist_cor <- cor.dendlist(iris_dendlist)
> iris_dendlist_cor

            ward.D    single  complete   average  mcquitty    median  centroid
ward.D   1.0000000 0.9836838 0.5774013 0.9841333 0.9641103 0.9451815 0.9809088
single   0.9836838 1.0000000 0.5665529 0.9681156 0.9329029 0.9444723 0.9903934
complete 0.5774013 0.5665529 1.0000000 0.6195121 0.6107473 0.6889092 0.5870062
average  0.9841333 0.9681156 0.6195121 1.0000000 0.9828015 0.9449422 0.9801444
mcquitty 0.9641103 0.9329029 0.6107473 0.9828015 1.0000000 0.9203374 0.9499123
median   0.9451815 0.9444723 0.6889092 0.9449422 0.9203374 1.0000000 0.9403569
centroid 0.9809088 0.9903934 0.5870062 0.9801444 0.9499123 0.9403569 1.0000000
ward.D2  0.9911648 0.9682507 0.6096286 0.9895131 0.9829977 0.9445832 0.9737886
           ward.D2
ward.D   0.9911648
single   0.9682507
complete 0.6096286
average  0.9895131
mcquitty 0.9829977
median   0.9445832
centroid 0.9737886
ward.D2  1.0000000

> library(corrplot)
> corrplot(iris_dendlist_cor, "pie", "lower")
>
```




```
> # The `which` parameter allows us to pick the elements in the list to compare
> iris_dendlist %>% dendlist(which = c(1,4)) %>% ladderize %>%
+     set("branches_k_color", k=2) %>%
+     tanglegram(faster = TRUE)
>
>
```


ward.D
average
150
100
50
0
0
1
2
3
4