**R: Hierarchical Clustering**

*Goal:* Performing clustering using hierarchical methods

*Data:* USArrests from Base R

**Table of Contents**

1 --- Data Description and Setup

2 --- Hierarchical Clustering

2 --- Linkage Methods

2 --- Dendrogram

4 --- Gap Statistic

5 --- Final Results

**Data Description and Setup**

In this demo, we will use USArrests from Base R, which we previously used for PCA. Recall that this dataset has four variables. *Murder*, *Assault*, and *Rape* measure the rates of each respective crime in the given state, while *UrbanPop* represents the population of the given state living in the city.

We use two libraries: **cluster** defines our hierarchical clustering procedures, while **factoextra** provides plotting functions for hierarchical clustering.

library(factoextra)

library(cluster)

example <- USArrests

# remove rows with missing values

example <- na.omit(example)

# scale each variable to have a mean of 0 and sd of 1

example <- scale(example)

# view first six rows of dataset

head(example)

Murder Assault UrbanPop Rape

Alabama 1.24256408 0.7828393 -0.5209066 -0.003416473

Alaska 0.50786248 1.1068225 -1.2117642 2.484202941

Arizona 0.07163341 1.4788032 0.9989801 1.042878388

Arkansas 0.23234938 0.2308680 -1.0735927 -0.184916602

California 0.27826823 1.2628144 1.7589234 2.067820292

Colorado 0.02571456 0.3988593 0.8608085 1.864967207

**Hierarchical Clustering**

**Steps for running hierarchical clustering:**

Step 1: Find the appropriate linkage method for your data;

Step 2: Construct dendrogram using the linkage method identified in Step 1

Step 3: Use Gap statistic to choose the optimal number of clusters

Step 4: Attach cluster labels to your data and analyze clusters

**Linkage Methods**

Agglomerative hierarchical clustering can be performed using several different measures of dissimilarity between data points of each cluster:

* **Average:** mean distance
* **Complete:** maximum distance
* **Single**: minimum distance
* **Ward**: minimum variance criterion; minimizes the total within-cluster variance

Here, we run the agglomerative clustering algorithm using each type of linkage and observe the difference between **agglomerative coefficient** which measures how tightly observations are clustered. We select the best linkage method as the one which obtains a coefficient closest to 1.

# define linkage methods to use

linkage <- c( "average", "single", "complete", "ward")

names(linkage) <- c( "average", "single", "complete", "ward")

# agnes: Compute agglomerative hierarchical clustering of the dataset

ac <- function(x) {

agnes(example, method = x)$ac

}

# calculate agglomerative coefficient for each clustering linkage method

sapply(linkage, ac)

average single complete ward

0.7379371 0.6276128 0.8531583 0.9346210

**Dendrogram**

Since Ward’s method yielded the highest agglomerative coefficient, let us use it to construct a cluster model. We can output our cluster results as a **dendogram**.

A dendrogram is a tree-like visual representation of the observations, that allows us to view at once the clustering obtained for each possible number of clusters, from 1 to n.This case study demonstrates which states are most similar in terms of urban population and the three crime rate variables, and how they naturally cluster together in a hierarchy.

The **agnes** function performs agglomerative clustering in R. Some important parameters include:

* *x*: a data frame or dissimilarity matrix of the data to cluster
* *metric*: specifies Euclidean or Manhattan distance (defaults to Euclidean)
* *method*: One of "average", "single", "complete", "ward" which species the linkage method
* *stand*: if specified as TRUE, standardizes the data before clustering

clust <- agnes(example, method = "ward")

pltree(clust, cex = 0.6, hang = -1, main = "Dendrogram")

Chart

Description automatically generated

**Gap Statistic**

Hierarchical clustering produces a dendrogram that presents all possible clusters at once, but to identify the optimal number of clusters, we must decide where to “cut” the dendrogram to group the observations. To do this, we rely on the **gap statistic** which estimates the “gap” between within-cluster dissimilarity produced by the algorithm compared to its expected value.

The number of clusters with the highest gap statistic is considered optimal. Below, we calculate the gap statistic for up to 10 clusters and plot the results (with standard error). The results display the following variables:

* *logW* – within-cluster dissimilarity
* *E.logW* – expected within-cluster dissimilarity
* *gap* – E.logW – logW (the gap statistic)
* *SE*.*sim* – standard error of the bootstrapped gap statistic

#calculate gap statistic for each number of clusters (up to 10 clusters)

gap\_stat <- clusGap(example, FUN = hcut, nstart = 25, K.max = 10, B = 50)

gap\_stat

logW E.logW gap SE.sim

[1,] 3.458369 3.631331 0.1729617 0.04024086

[2,] 3.142869 3.381104 0.2382354 0.03866028

[3,] 2.978518 3.250034 0.2715162 0.04374389

[4,] 2.833430 3.135966 0.3025368 0.04940110

[5,] 2.759739 3.040603 0.2808640 0.04950267

[6,] 2.689377 2.954023 0.2646462 0.04926910

[7,] 2.616505 2.871907 0.2554023 0.04825944

[8,] 2.556388 2.799210 0.2428215 0.04935223

[9,] 2.478796 2.728880 0.2500836 0.04888044

[10,] 2.408612 2.663169 0.2545569 0.04801010

#produce plot of gap statistic vs. number of clusters

fviz\_gap\_stat(

gap\_stat,

linecolor = "red",

maxSE = list(method = "firstSEmax", SE.factor = 0)

)

Chart, line chart

Description automatically generated

**Final Results**

From the gap statistic, we see that the optimal number of clusters is 4. Now, we will perform the clustering, cut the tree into 4 clusters, and examine the resulting labels and sizes of each cluster.

The final results show that while cluster 3 is the largest (19 states) and cluster 1 is the smallest (7 states), there are no trivial clusters. Cluster 4 has the lowest average crime rates and urban population, cluster 1 has a relatively high murder rate compared to other crimes, and cluster 2 has the highest average rates of rape, assault, and urban population.

Hence, by outputting cluster labels, we can analyze the differences between clusters.

#compute distance matrix

d <- dist(example, method = "euclidean")

# perform hierarchical clustering using Ward's method

final\_clust <- hclust(d, method = "ward.D2" )

# cut the dendrogram into 4 clusters based on the Gap Statistic

clusters <- cutree(final\_clust, k=4)

# find number of observations in each cluster

table(clusters)

clusters

1 2 3 4

7 12 19 12

# append cluster labels to original data

final\_example <- cbind(USArrests, cluster = clusters)

# display first 20 rows of final data

head(final\_example, 20)

Murder Assault UrbanPop Rape cluster

Alabama 13.2 236 58 21.2 1

Alaska 10.0 263 48 44.5 2

Arizona 8.1 294 80 31.0 2

Arkansas 8.8 190 50 19.5 3

California 9.0 276 91 40.6 2

Colorado 7.9 204 78 38.7 2

Connecticut 3.3 110 77 11.1 3

Delaware 5.9 238 72 15.8 3

Florida 15.4 335 80 31.9 2

Georgia 17.4 211 60 25.8 1

Hawaii 5.3 46 83 20.2 3

Idaho 2.6 120 54 14.2 4

Illinois 10.4 249 83 24.0 2

Indiana 7.2 113 65 21.0 3

Iowa 2.2 56 57 11.3 4

Kansas 6.0 115 66 18.0 3

Kentucky 9.7 109 52 16.3 3

Louisiana 15.4 249 66 22.2 1

Maine 2.1 83 51 7.8 4

Maryland 11.3 300 67 27.8 2

# find mean values for each cluster

aggregate(final\_example, by=list(cluster=final\_example$cluster), mean)

cluster Murder Assault UrbanPop Rape cluster

1 1 14.671429 251.2857 54.28571 21.68571 1

2 2 10.966667 264.0000 76.50000 33.60833 2

3 3 6.210526 142.0526 71.26316 19.18421 3

4 4 3.091667 76.0000 52.08333 11.83333 4