Diana & Spectral Clustering

Bhuvan Kapoor [23MDT1003]

2024-03-19

# Diana  
library(datasets)  
library(cluster)  
library(factoextra)

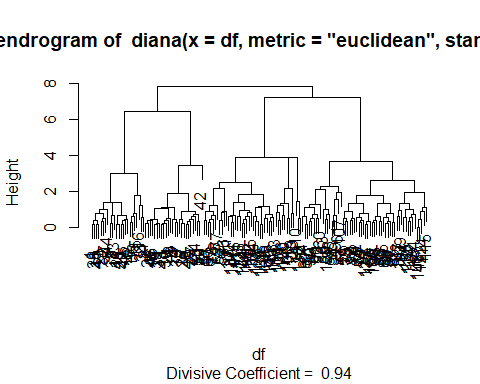
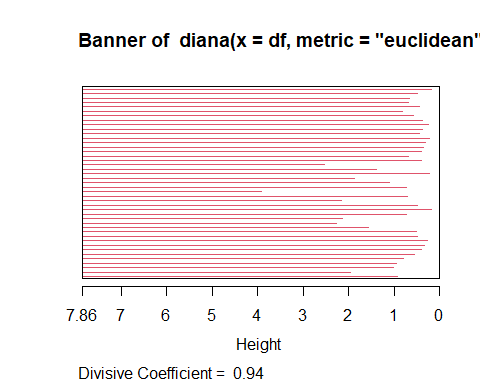
## Loading required package: ggplot2

## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa

library(purrr)  
  
df <- iris[,1:4]  
dv <- diana(df, metric = "euclidean", stand = T)  
print(dv)

## Merge:  
## [,1] [,2]  
## [1,] -102 -143  
## [2,] -8 -40  
## [3,] -11 -49  
## [4,] -10 -35  
## [5,] -129 -133  
## [6,] -18 -41  
## [7,] -128 -139  
## [8,] -3 -48  
## [9,] -1 -28  
## [10,] -81 -82  
## [11,] -20 -47  
## [12,] -12 -25  
## [13,] -2 -26  
## [14,] 8 -30  
## [15,] -121 -144  
## [16,] -5 -38  
## [17,] 4 -31  
## [18,] -89 -96  
## [19,] -66 -87  
## [20,] -137 -149  
## [21,] -64 -79  
## [22,] -14 -39  
## [23,] -56 -100  
## [24,] -6 -17  
## [25,] -67 -85  
## [26,] 2 -29  
## [27,] 7 -150  
## [28,] -36 -50  
## [29,] -68 -83  
## [30,] -21 -32  
## [31,] -58 -94  
## [32,] -117 -138  
## [33,] -112 -124  
## [34,] -113 -140  
## [35,] 18 -97  
## [36,] -70 -90  
## [37,] -59 -75  
## [38,] 13 -13  
## [39,] 38 -46  
## [40,] 9 6  
## [41,] 14 -4  
## [42,] 11 -45  
## [43,] 21 -92  
## [44,] 23 -95  
## [45,] -72 -98  
## [46,] -52 -57  
## [47,] -51 -53  
## [48,] -108 -131  
## [49,] -9 22  
## [50,] 19 -76  
## [51,] 42 -22  
## [52,] -69 -120  
## [53,] 26 -27  
## [54,] 15 -141  
## [55,] 39 17  
## [56,] -7 12  
## [57,] 1 -122  
## [58,] -118 -132  
## [59,] -80 -93  
## [60,] -106 -136  
## [61,] 30 -37  
## [62,] 45 -74  
## [63,] 41 -43  
## [64,] 36 -91  
## [65,] -55 37  
## [66,] -105 -148  
## [67,] -65 35  
## [68,] 34 -146  
## [69,] 33 -127  
## [70,] -62 43  
## [71,] -84 -135  
## [72,] -33 -34  
## [73,] -54 10  
## [74,] -111 -125  
## [75,] -101 20  
## [76,] 53 -24  
## [77,] 24 -19  
## [78,] 54 -142  
## [79,] -73 -147  
## [80,] -104 32  
## [81,] 44 29  
## [82,] 16 -23  
## [83,] 31 -99  
## [84,] 3 51  
## [85,] 75 -116  
## [86,] 50 -78  
## [87,] -126 -130  
## [88,] 69 -134  
## [89,] 64 59  
## [90,] 76 28  
## [91,] 66 68  
## [92,] 52 -88  
## [93,] 40 61  
## [94,] -119 -123  
## [95,] 67 25  
## [96,] 70 27  
## [97,] 77 -15  
## [98,] -103 48  
## [99,] 78 -145  
## [100,] 46 -86  
## [101,] 65 62  
## [102,] 100 -71  
## [103,] 56 90  
## [104,] 47 86  
## [105,] 57 -114  
## [106,] 79 -109  
## [107,] 74 99  
## [108,] 93 -44  
## [109,] -63 92  
## [110,] 71 88  
## [111,] 55 63  
## [112,] 111 49  
## [113,] 80 5  
## [114,] 105 -115  
## [115,] 73 89  
## [116,] 113 91  
## [117,] 104 -77  
## [118,] 98 87  
## [119,] -16 72  
## [120,] 115 -60  
## [121,] 118 60  
## [122,] 106 110  
## [123,] 81 95  
## [124,] 101 96  
## [125,] 85 107  
## [126,] 82 84  
## [127,] 108 126  
## [128,] -110 58  
## [129,] 83 -61  
## [130,] 97 119  
## [131,] 124 123  
## [132,] 125 116  
## [133,] 112 103  
## [134,] 129 -107  
## [135,] 122 114  
## [136,] 120 109  
## [137,] 121 94  
## [138,] 117 102  
## [139,] 136 134  
## [140,] 138 132  
## [141,] 131 135  
## [142,] 127 130  
## [143,] 133 -42  
## [144,] 139 141  
## [145,] 137 128  
## [146,] 140 145  
## [147,] 142 143  
## [148,] 146 144  
## [149,] 147 148  
## Order of objects:  
## [1] 1 28 18 41 21 32 37 44 5 38 23 11 49 20 47 45 22 6  
## [19] 17 19 15 16 33 34 2 26 13 46 10 35 31 3 48 30 4 43  
## [37] 9 14 39 7 12 25 8 40 29 27 24 36 50 42 51 53 66 87  
## [55] 76 78 77 52 57 86 71 101 137 149 116 111 125 121 144 141 142 145  
## [73] 104 117 138 129 133 105 148 113 140 146 103 108 131 126 130 106 136 119  
## [91] 123 110 118 132 54 81 82 70 90 91 80 93 60 63 69 120 88 58  
## [109] 94 99 61 107 55 59 75 72 98 74 62 64 79 92 128 139 150 56  
## [127] 100 95 68 83 65 89 96 97 67 85 73 147 109 84 135 112 124 127  
## [145] 134 102 143 122 114 115  
## Height:  
## [1] 0.1588972 0.3522506 0.1588972 0.7429182 0.3297390 0.4669556 0.9669998  
## [8] 1.3809909 0.2103351 0.6355887 1.3809909 0.1454428 0.6606155 0.1648695  
## [15] 0.3594459 0.4296625 3.0192253 0.2559596 0.6059300 0.7882515 1.7926087  
## [22] 1.2369804 0.5529290 6.4781819 0.1937329 0.3522506 0.3522506 0.4536115  
## [29] 0.1519449 0.2198535 1.0728684 0.1588972 0.1937329 0.3545399 0.4766915  
## [36] 1.0779525 0.4114114 0.2462107 1.8862618 0.4583332 0.1919697 0.8857861  
## [43] 0.1454428 0.2978408 0.4397070 0.5949431 0.7145130 0.3233341 3.4251727  
## [50] 7.8596012 0.3857276 0.8875159 0.2448257 0.4114114 0.6658535 1.2244864  
## [57] 2.1580112 0.3857276 0.8451091 0.8794758 2.5047168 0.5908284 0.2462107  
## [64] 0.6643595 1.3663360 0.5858987 0.9472968 0.1937329 0.4471717 0.6094103  
## [71] 0.8228228 1.8447002 0.6147461 0.3306353 1.0802711 0.1519449 1.1669624  
## [78] 0.4896514 0.7177927 0.3367706 0.5305736 3.9045934 0.8073257 0.3857276  
## [85] 1.2338777 0.6792586 1.3026772 0.4646184 2.1474805 0.7594923 3.9045934  
## [92] 1.4153532 0.4624022 7.2186132 0.5768914 0.1648695 1.0900745 0.3429247  
## [99] 0.4845355 0.7131863 0.4646184 1.2760181 2.1189644 0.9933036 0.4324143  
## [106] 0.7199354 2.2570873 0.3306353 0.6586028 1.5332193 1.9406320 3.6954491  
## [113] 0.4883862 0.3485209 0.8743087 0.3823246 0.4729988 1.3432207 0.5461094  
## [120] 0.2462107 0.3694613 0.7795459 0.1588972 0.3177944 1.7974229 0.2559596  
## [127] 0.3823246 0.6185631 0.3297390 1.3321596 0.5289768 0.2198535 0.3367706  
## [134] 0.7725288 0.2908856 2.5843574 0.6111389 0.9369708 1.3159809 0.5513950  
## [141] 1.0026342 0.3312947 0.5461094 0.7036296 1.9434767 0.0000000 0.4607023  
## [148] 0.9048444 1.0900180  
## Divisive coefficient:  
## [1] 0.9408201  
##   
## Available components:  
## [1] "order" "height" "dc" "merge" "diss" "call" "data"

plot(dv)



for (i in 2:5){  
 dvcut <- cutree(as.hclust(dv), k=i)  
 print(table(dvcut))  
}

## dvcut  
## 1 2   
## 50 100   
## dvcut  
## 1 2 3   
## 50 44 56   
## dvcut  
## 1 2 3 4   
## 24 26 44 56   
## dvcut  
## 1 2 3 4 5   
## 24 26 32 56 12

## Spectral Clustering  
  
library(igraph)

##   
## Attaching package: 'igraph'

## The following objects are masked from 'package:purrr':  
##   
## compose, simplify

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

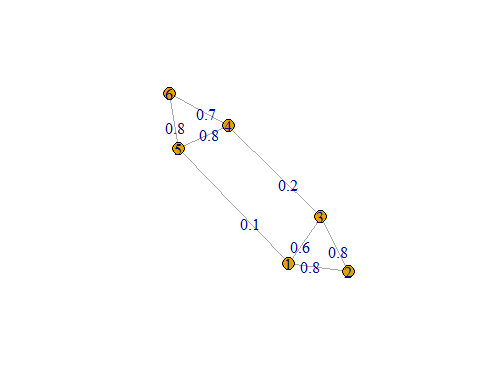
A <- rbind(c(0,0.8,0.6,0,0.1,0),c(0.8,0,0.8,0,0,0),c(0.6,0.8,0,0.2,0,0),c(0,0,0.2,0,0.8,0.7),c(0.1,0,0,0.8,0,0.8),c(0,0,0,0.7,0.8,0))  
A

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 0.0 0.8 0.6 0.0 0.1 0.0  
## [2,] 0.8 0.0 0.8 0.0 0.0 0.0  
## [3,] 0.6 0.8 0.0 0.2 0.0 0.0  
## [4,] 0.0 0.0 0.2 0.0 0.8 0.7  
## [5,] 0.1 0.0 0.0 0.8 0.0 0.8  
## [6,] 0.0 0.0 0.0 0.7 0.8 0.0

ig <- graph.adjacency(A, mode = 'undirected', weighted = T)

## Warning: `graph.adjacency()` was deprecated in igraph 2.0.0.  
## ℹ Please use `graph\_from\_adjacency\_matrix()` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

plot(ig, edge.label = E(ig)$weight)



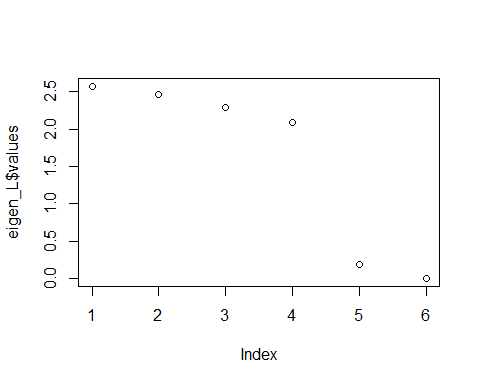
D <- diag(apply(A,1,sum))  
D

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 1.5 0.0 0.0 0.0 0.0 0.0  
## [2,] 0.0 1.6 0.0 0.0 0.0 0.0  
## [3,] 0.0 0.0 1.6 0.0 0.0 0.0  
## [4,] 0.0 0.0 0.0 1.7 0.0 0.0  
## [5,] 0.0 0.0 0.0 0.0 1.7 0.0  
## [6,] 0.0 0.0 0.0 0.0 0.0 1.5

L <- D-A  
L

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] 1.5 -0.8 -0.6 0.0 -0.1 0.0  
## [2,] -0.8 1.6 -0.8 0.0 0.0 0.0  
## [3,] -0.6 -0.8 1.6 -0.2 0.0 0.0  
## [4,] 0.0 0.0 -0.2 1.7 -0.8 -0.7  
## [5,] -0.1 0.0 0.0 -0.8 1.7 -0.8  
## [6,] 0.0 0.0 0.0 -0.7 -0.8 1.5

eigen\_L <- eigen(L)  
plot(eigen\_L$values)



eigen\_L$vectors

## [,1] [,2] [,3] [,4] [,5] [,6]  
## [1,] -0.10598786 -0.3788245748 -0.30546656 0.64690880 0.4084006 0.4082483  
## [2,] -0.21517718 0.7063206485 0.30450981 -0.01441501 0.4418249 0.4082483  
## [3,] 0.36782805 -0.3884382495 0.04461661 -0.63818761 0.3713186 0.4082483  
## [4,] -0.61170644 -0.0009962363 -0.45451793 -0.33863293 -0.3713338 0.4082483  
## [5,] 0.65221488 0.3509689196 -0.30495543 0.16645901 -0.4050475 0.4082483  
## [6,] -0.08717145 -0.2890305075 0.71581349 0.17786774 -0.4451628 0.4082483

k <- 2  
z <- eigen\_L$vectors[,5:6]  
z

## [,1] [,2]  
## [1,] 0.4084006 0.4082483  
## [2,] 0.4418249 0.4082483  
## [3,] 0.3713186 0.4082483  
## [4,] -0.3713338 0.4082483  
## [5,] -0.4050475 0.4082483  
## [6,] -0.4451628 0.4082483

# swapping the z matrix  
z <- cbind(eigen\_L$vectors[,6],eigen\_L$vectors[,5])  
# or z[,c(1,2)] = z[,c(2,1)]  
z

## [,1] [,2]  
## [1,] 0.4082483 0.4084006  
## [2,] 0.4082483 0.4418249  
## [3,] 0.4082483 0.3713186  
## [4,] 0.4082483 -0.3713338  
## [5,] 0.4082483 -0.4050475  
## [6,] 0.4082483 -0.4451628

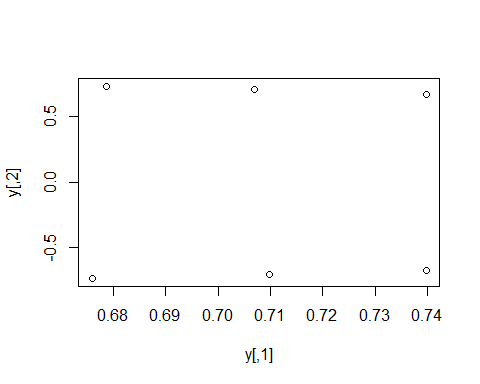
# Normalizing the Z matrix  
denominator <- sqrt(apply(z^2,1,sum))  
  
denorminator2 <- replicate(k, expr = denominator)  
denorminator2

## [,1] [,2]  
## [1,] 0.5774580 0.5774580  
## [2,] 0.6015612 0.6015612  
## [3,] 0.5518552 0.5518552  
## [4,] 0.5518654 0.5518654  
## [5,] 0.5750914 0.5750914  
## [6,] 0.6040171 0.6040171

y <- z/denorminator2  
y

## [,1] [,2]  
## [1,] 0.7069749 0.7072386  
## [2,] 0.6786479 0.7344637  
## [3,] 0.7397743 0.6728551  
## [4,] 0.7397606 -0.6728702  
## [5,] 0.7098842 -0.7043185  
## [6,] 0.6758887 -0.7370037

plot(y)



# Apply KMeans on y  
spec\_cluster <- kmeans(y, centers = 2, nstart = 100)  
spec\_cluster$cluster

## [1] 2 2 2 1 1 1

spec\_cluster$size

## [1] 3 3

spec\_cluster$centers

## [,1] [,2]  
## 1 0.7085111 -0.7047308  
## 2 0.7084657 0.7048525