Cluster Analysis

Table of Contents

* Required packages

knitr::opts\_chunk$set(echo = TRUE)  
#install.packages("dplyr","ade4","magrittr","cluster","factoextra","cluster.datasets","xtable","kableExtra","knitr","summarytools")  
knitr::opts\_chunk$set(echo = TRUE)

# Definition of a distance

* A distance function or a metric on , is a function .
* A distance function must satisfy some required properties or axioms.
* There are three main axioms.
* A1. (identity of indiscernibles);
* A2. (symmetry);
* A3. (triangle inequality), where , and are all vectors of .
* We should use the term *dissimilarity* rather than *distance* when not all the three axioms A1-A3 are valid.
* Most of the time, we shall use, with some abuse of vocabulary, the term distance.

# Exercice 1

* Prove that the three axioms A1-A3 imply the non-negativity condition:

# Euclidean distance

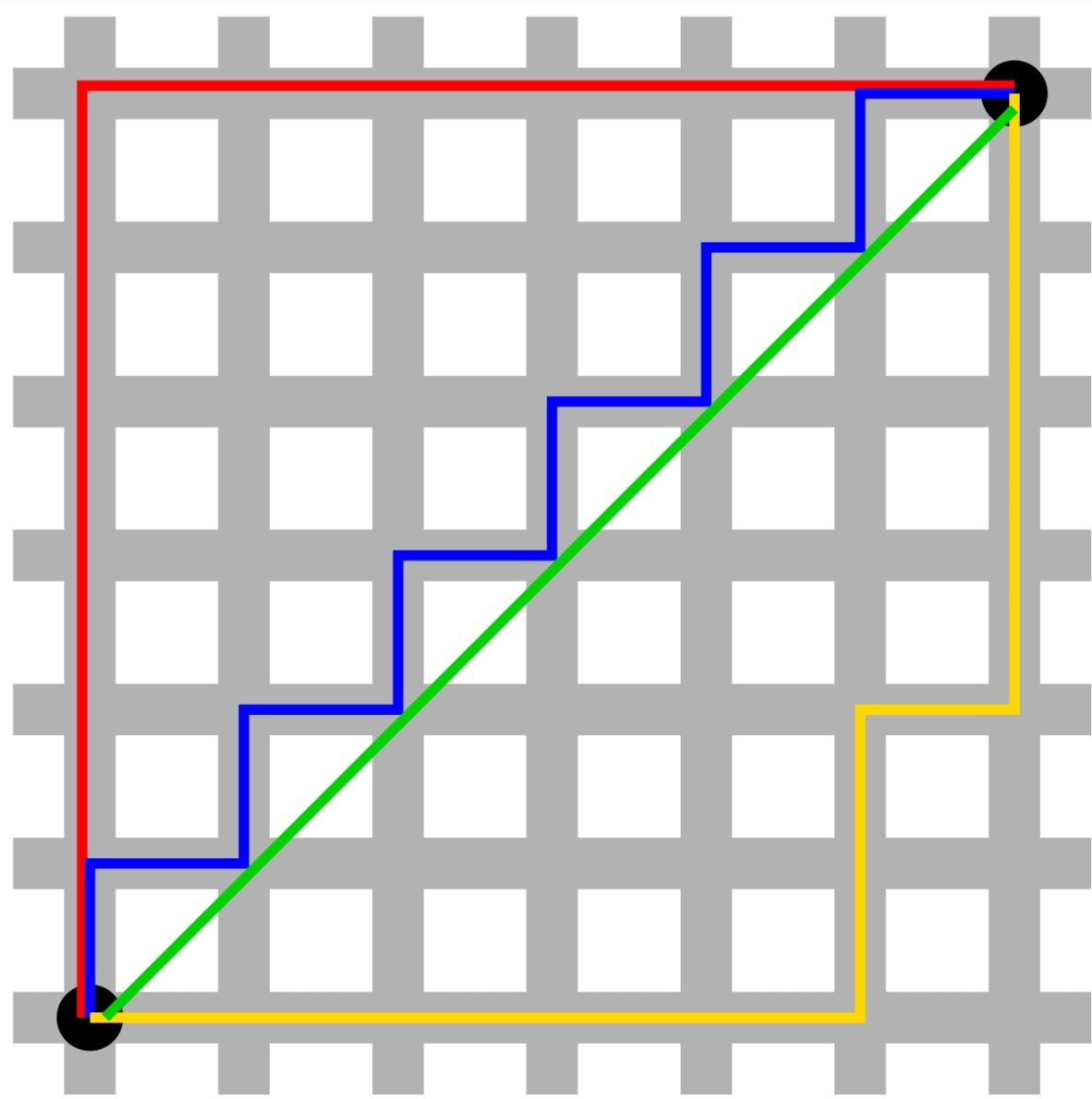
* It is defined by:
* A1-A2 are obvious.
* The proof of A3 is provided below.

# Exercice 2

* Is the squared Euclidian distance a true distance?

# Manhattan distance

* The Manhattan distance also called taxi-cab metric or city-block metric is defined by:
* A1-A2 hold.
* A3 also holds using the fact that for any reals .
* There exists also a weighted version of the Manhattan distance called the Canberra distance.



x = c(0, 0)  
y = c(6,6)  
dist(rbind(x, y), method = "euclidian")

## x  
## y 8.485281

dist(rbind(x, y), method = "euclidian",diag=T,upper=T)

## x y  
## x 0.000000 8.485281  
## y 8.485281 0.000000

6\*sqrt(2)

## [1] 8.485281

dist(rbind(x, y), method = "manhattan")

## x  
## y 12

dist(rbind(x, y), method = "manhattan",diag=T,upper=T)

## x y  
## x 0 12  
## y 12 0

# Canberra distance

* It is defined by:
* Note that the term is not properly defined as: .
* By convention we set that term to be zero in that case.
* The Canberra distance is specially sensitive to small changes near zero.

x = c(0, 0)  
y = c(6,6)  
dist(rbind(x, y), method = "canberra")

## x  
## y 2

6/6+6/6

## [1] 2

# Exercice 3

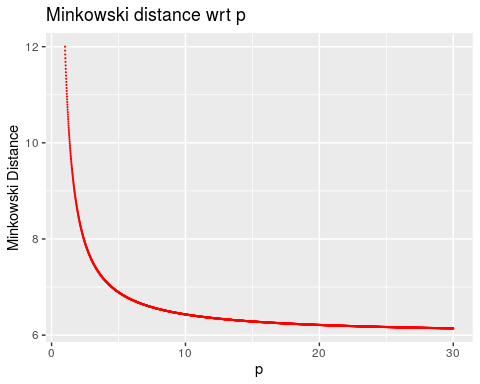
* Prove that the Canberra distance is a true distance, i.e. that it satisfies A1-A3.

# Minkowski distance

* Both the Euclidian and the Manattan distances are special cases of the Minkowski distance which is defined, for , by:
* For , we get the Manhattan distance.
* For , we get the Euclidian distance.
* Let us also define:
* where is known as the -norm or Minkowski norm.
* Note that the Minkowski distance and norm are related by:
* Conversely, we have:

where is the null-vetor of .

library("ggplot2")  
x = c(0, 0)  
y = c(6,6)  
MinkowDist=c() # Initialiser à vide la liste  
for (p in seq(1,30,.01))  
{  
MinkowDist=c(MinkowDist,dist(rbind(x, y), method = "minkowski", p = p))   
}  
  
ggplot(data =data.frame(x = seq(1,30,.01), y=MinkowDist ) , mapping = aes( x=x, y= y))+  
 geom\_point(size=.1,color="red")+  
 xlab("p")+ylab("Minkowski Distance")+ggtitle("Minkowski distance wrt p")



# Exercice 4

Produce a similar graph using “The Economist” theme. Indicate on the graph the Manhattan, the Euclidian distances as well as the Chebyshev distance introduced below.

# Chebyshev distance

* At the limit, we get the Chebyshev distance which is defined by:
* The corresponding norm is:

# Minkowski inequality

* The proof of the triangular inequality A3 is based on the Minkowski inequality:
* For any nonnegative real numbers ; , and for any , we have:
* To prove that the Minkowski distance satisfies A3, notice that
* Since for any reals , we have: , and using the fact that is increasing in , we obtain:
* Applying the Minkowski inequality with and , , we get:

# Exercice 5

To illustrate the Minkowski inequality, draw times two lists of draws from the lognormal distribution with mean and standard-deviation . Illustrate with a graph the gap between the two drawn lists.

# Hölder inequality

* The proof of the Minkowski inequality itself requires the Hölder inequality:
* For any nonnegative real numbers ; , and any with , we have:
* The proof of the Hölder inequality relies on the Young inequality:
* For any , we have
* with equality occuring iff: .
* To prove the Young inequality, one can use the (strict) convexity of the exponential function.
* For any reals , we have:
* We then set: and to get the Young inequality.
* A good reference on inequalities is: [Z. Cvetkovski, Inequalities: theorems, techniques and selected problems, 2012, Springer Science & Business Media](https://book4you.org/book/1228394/3ec0fb).

# Cauchy-Schwartz inequality

* Note that the triangular inequality for the Minkowski distance implies:
* Note that for , we have . The Hölder inequality implies for that special case
* Since the LHS od thes above inequality is greater then , we get the Cauchy-Schwartz inequality
* Using the dot product notation called also scalar product notation: , and the norm notation , the Cauchy-Schwartz inequality is:

# Pearson correlation distance

* The Pearson correlation coefficient is a similarity measure on defined by:
* where is the mean of the vector defined by:
* Note that the Pearson correlation coefficient satisfies P2 and is invariant to any positive linear transformation, i.e.:
* for any .
* The Pearson distance (or correlation distance) is defined by:
* Note that the Pearson distance does not satisfy A1 since for any non-zero vector . It neither satisfies the triangle inequality. However, the symmetry property is fullfilled.

# Cosine correlation distance

* The cosine of the angle between two vectors and is a measure of similarity given by:
* Note that the cosine of the angle between the two centred vectors and coincides with the Pearson correlation coefficient of and , where is a vector of units of .
* The cosine correlation distance is defined by:
* It shares similar properties than the Pearson correlation distance. Likewise, Axioms A1 and A3 are not satisfied.

# Spearman correlation distance

* To calculate the Spearman’s rank-order correlation, we need to map seperately each of the vectors to ranked data values:
* Here, is the rank of among the set of values of .
* We illustrate this transformation with a simple example:
* If , then the rank-order vector is .

x=c(3, 1, 4, 15, 92)  
rank(x)

## [1] 2 1 3 4 5

* The Spearman’s rank correlation of two numerical vectors and is simply the Pearson correlation of the two correspnding rank-order vectors and , i.e. . This measure is is useful because it is more robust against outliers than the Pearson correlation.
* If all the ranks are distinct, it can be computed using the following formula:
* where .
* The spearman distance is then defined by:
* It can be shown that easaly that it is not a proper distance.
* If all the ranks are distinct, we get:

x=c(3, 1, 4, 15, 92)  
rank(x)

## [1] 2 1 3 4 5

y=c(30,2 , 9, 20, 48)  
rank(y)

## [1] 4 1 2 3 5

d=rank(x)-rank(y)  
d

## [1] -2 0 1 1 0

cor(rank(x),rank(y))

## [1] 0.7

1-6\*sum(d^2)/(5\*(5^2-1))

## [1] 0.7

# Exercice 6

* For the two vectors and :
* Calculate the ranks for each vector.
* Deduce the Spearman correlation distance from that ranks.
* Deduce the Spearman correlation distance from the above dispalyed alternative equation.
* Calculate the Spearman correlation distance using the **R** function.

# Kendall tau distance

* The Kendall rank correlation coefficient is calculated from the number of correspondances between the rankings of and the rankings of .
* The number of pairs of observations among observations or values is:
* The pairs of observations and are said to be *concordant* if:
* and to be *discordant* if:
* where returns for positive numbers and negative numbers and otherwise.
* If or (or both), there is a tie.
* The Kendall coefficient is defined by (neglecting ties):
* Let (resp. ) be the number of concordant (resp. discordant) pairs, we have
* The Kendall tau distance is then:
* Remark: the triangular inequality may fail in cases where there are ties.

x=c(3, 1, 4, 15, 92)  
y=c(30,2 , 9, 20, 48)  
tau=0  
for (i in 1:5)  
{   
tau=tau+sign(x -x[i])%\*%sign(y -y[i])  
}  
tau=tau/(5\*4)  
tau

## [,1]  
## [1,] 0.6

cor(x,y, method="kendall")

## [1] 0.6

# Exercice 7

* For the two vectors and :
* List all pairs of coordinates.
* How many pairs are there?
* For each pair and each cector, compute the signs of the differences in coordinates.
* Deduce the Kendall tau coefficient using the above computations.
* Calculate the Kendall tau coefficient using the R function.

# Standardization

* Variables or measurements are often standardized before calculating dissimilarities.
* Standardization converts the original variables into uniteless variables.
* A well known method is the z-score transformation.
* Let a vector of measurements recrded for individuals or objects.
* where are the sample mean and standard-deviation, respectively, given by:
* The transformed variable will have a mean of and a variance of .
* The result obtained with Pearson correlation measures and standardized Euclidean distances are comparable.
* For other methods, see: [Milligan, G. W., & Cooper, M. C. (1988). A study of standardization of variables in cluster analysis. *Journal of classification*, *5*(2), 181-204](https://booksc.org/book/6755563/66358a)

v=c(3, 1, 4, 15, 92)  
w=c(30,2 , 9, 20, 48)  
(v-mean(v))/sd(v)

## [1] -0.5134116 -0.5647527 -0.4877410 -0.2053646 1.7712699

scale(v)

## [,1]  
## [1,] -0.5134116  
## [2,] -0.5647527  
## [3,] -0.4877410  
## [4,] -0.2053646  
## [5,] 1.7712699  
## attr(,"scaled:center")  
## [1] 23  
## attr(,"scaled:scale")  
## [1] 38.9551

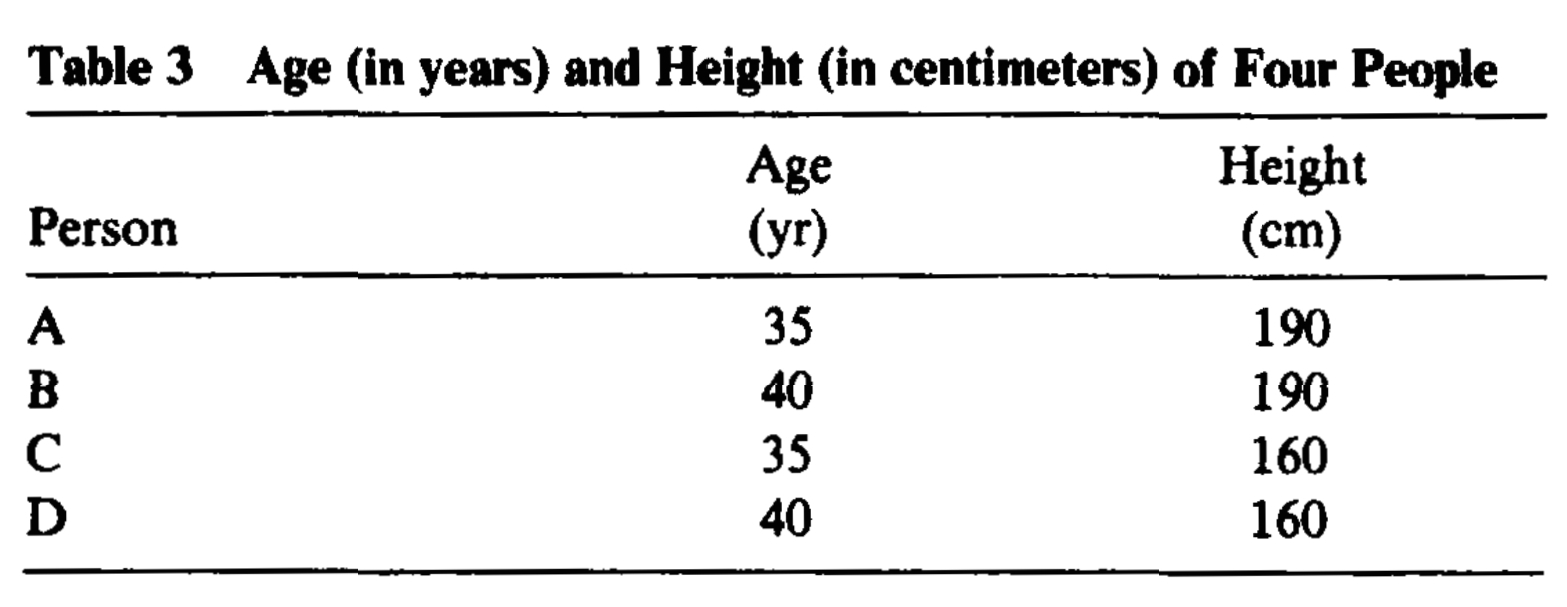
(w-mean(w))/sd(w)

## [1] 0.45263128 -1.09293895 -0.70654639 -0.09935809 1.44621214

scale(w)

## [,1]  
## [1,] 0.45263128  
## [2,] -1.09293895  
## [3,] -0.70654639  
## [4,] -0.09935809  
## [5,] 1.44621214  
## attr(,"scaled:center")  
## [1] 21.8  
## attr(,"scaled:scale")  
## [1] 18.11629

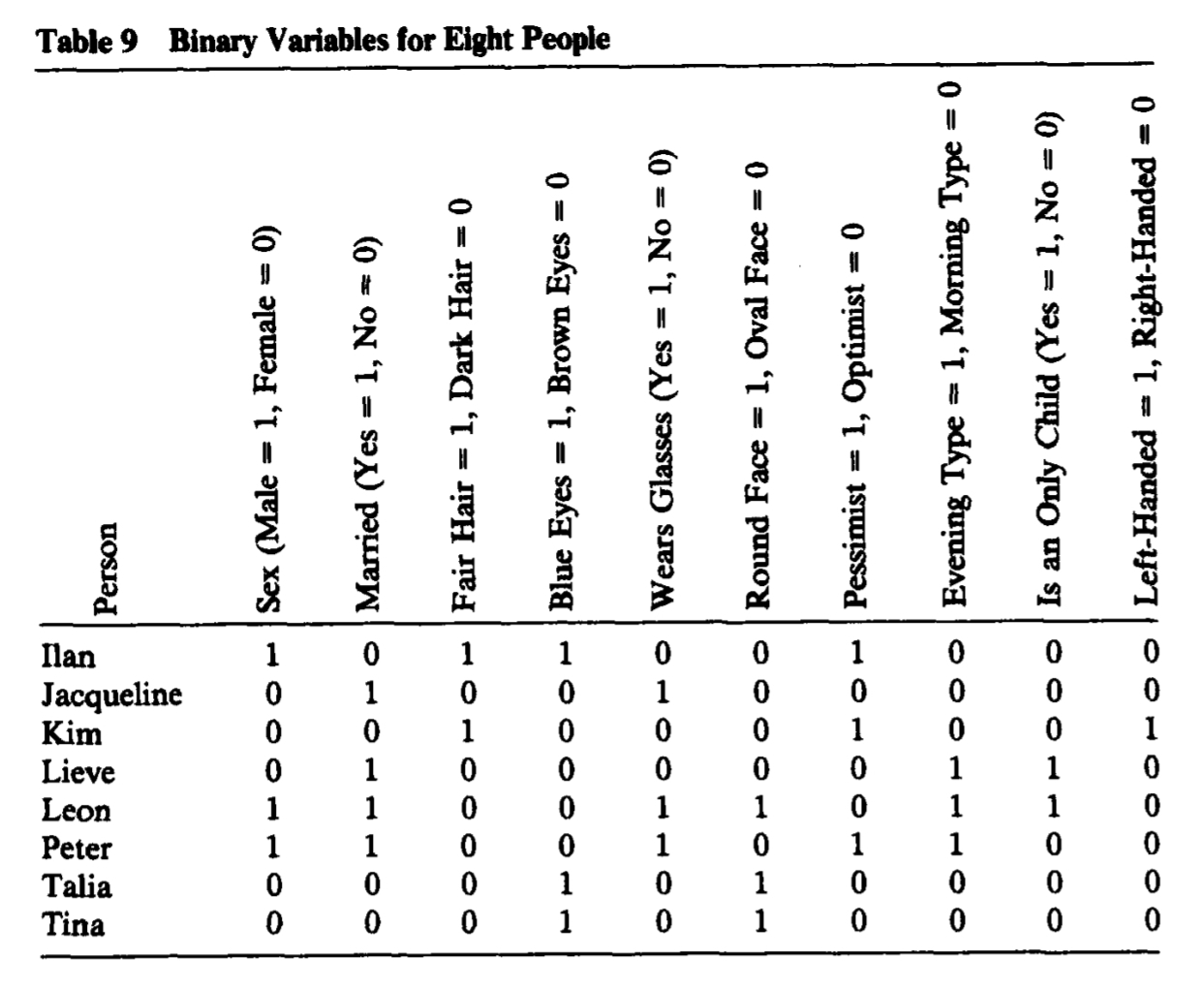
# Exercice 8

* Consider the following example 
* Plot the data using a nice scatter plot.
* Transform the Height from centimeters (cm) into feet (ft).
* Display your data in a table.
* Plot the data within a new scatter plot.
* What do you observe?
* Standardize the two variables Age and Height.
* Display your data in a table.
* Plot the standardized data within a new scatter plot.
* Conclude.

# Similarity measures for binary data

* A common simple situation occurs when all information is of the presence/absence of 2-level qualitative characters.
* We assume there are characters.
* \*The presence of the character is coded by and the absence by 0.
* We have have at our disposal two vectors.
* is observed for a first individual (or object).
* is observed for a second individual.
* We can then calculate the following four statistics:
* The counts of matches are for and for ;
* The counts of mismatches are for and for .
* Note that obviously: .
* This gives a very useful association table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Second individual |  |  |
|  |  | 1 | 0 | *Totals* |
| **First individual** | 1 |  |  |  |
|  | 0 |  |  |  |
| *Totals* |  |  |  |  |

 Table from [Kaufman, L., & Rousseeuw, P. J. (2009). Finding groups in data: an introduction to cluster analysis (Vol. 344). John Wiley & Sons](https://book4you.org/book/669277/d2cf59)

* The data shows people (individuals) and binary variables:
* Sex, Married, Fair Hair, Blue Eyes, Wears Glasses, Round Face, Pessimist, Evening Type, Is an Only Child, Left-Handed.

data=c(  
1,0,1,1,0,0,1,0,0,0,  
0,1,0,0,1,0,0,0,0,0,  
0,0,1,0,0,0,1,0,0,1,  
0,1,0,0,0,0,0,1,1,0,  
1,1,0,0,1,1,0,1,1,0,  
1,1,0,0,1,0,1,1,0,0,  
0,0,0,1,0,1,0,0,0,0,  
0,0,0,1,0,1,0,0,0,0  
)  
data=data.frame(matrix(data, nrow=8,byrow=T))  
row.names(data)=c("Ilan","Jacqueline","Kim","Lieve","Leon","Peter","Talia","Tina")  
names(data)=c("Sex", "Married", "Fair Hair", "Blue Eyes", "Wears Glasses", "Round Face", "Pessimist", "Evening Type", "Is an Only Child", "Left-Handed")

* We are comparing the records for Ilan with Talia.

library(knitr)  
library(xtable)  
library(stargazer)  
library(texreg)  
library(kableExtra)  
library(summarytools)

## Warning in fun(libname, pkgname): couldn't connect to display ":0"

set.seed(893)  
datat<-as.data.frame(t(data))  
datat=lapply(datat,as.factor)  
Ilan=datat$Ilan  
Talia =datat$Talia  
print(ctable(Ilan,Talia,prop = 'n',style = "rmarkdown"))

### Cross-Tabulation

#### Ilan \* Talia

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Talia | 0 | 1 | Total |
| Ilan |  |  |  |  |
| 0 |  | 5 | 1 | 6 |
| 1 |  | 3 | 1 | 4 |
| Total |  | 8 | 2 | 10 |

* Therefore: .
* Note that interchanging Ilan and Talia would permute and while leaving and unchanged.
* A good similarity or dissimilarity coefficient must treat and symmetrically.
* A similarity measure is denoted by: .
* The corresponding distance is then defined as:
* Alternatively, we have:
* A list of some of the similarity measures that have been suggested for binary data is shown below.
* An more complete list can be found in: [Gower, J. C., & Legendre, P. (1986). Metric and Euclidean properties of dissimilarity coefficients. *Journal of classification*, *3*(1), 5-48](https://booksc.org/book/6755353/c44198).

|  |  |  |
| --- | --- | --- |
| Coefficient |  |  |
| Simple matching |  |  |
| Jaccard |  |  |
| Rogers and Tanimoto (1960) |  |  |
| Gower and Legendre (1986) |  |  |
| Gower and Legendre (1986) |  |  |

* To calculate these coefficients, we use the function: [dist.binary().](https://www.rdocumentation.org/packages/ade4/versions/1.7-16/topics/dist.binary) available in the **ade4** package.
* All the distances in the **ade4** package are of type .

library(ade4)  
a=1  
b=3  
c=1  
d=5  
dist.binary(data[c("Ilan","Talia"),],method=2)^2

Ilan

Talia 0.4

1-(a+d )/(a+b+c+d)

[1] 0.4

dist.binary(data[c("Ilan","Talia"),],method=1)^2

Ilan

Talia 0.8

1-a/(a+b+c)

[1] 0.8

dist.binary(data[c("Ilan","Talia"),],method=4)^2

Ilan

Talia 0.5714286

1-(a+d )/(a+2\*(b+c)+d)

[1] 0.5714286

# One Gower coefficient is missing  
dist.binary(data[c("Ilan","Talia"),],method=5)^2

Ilan

Talia 0.6666667

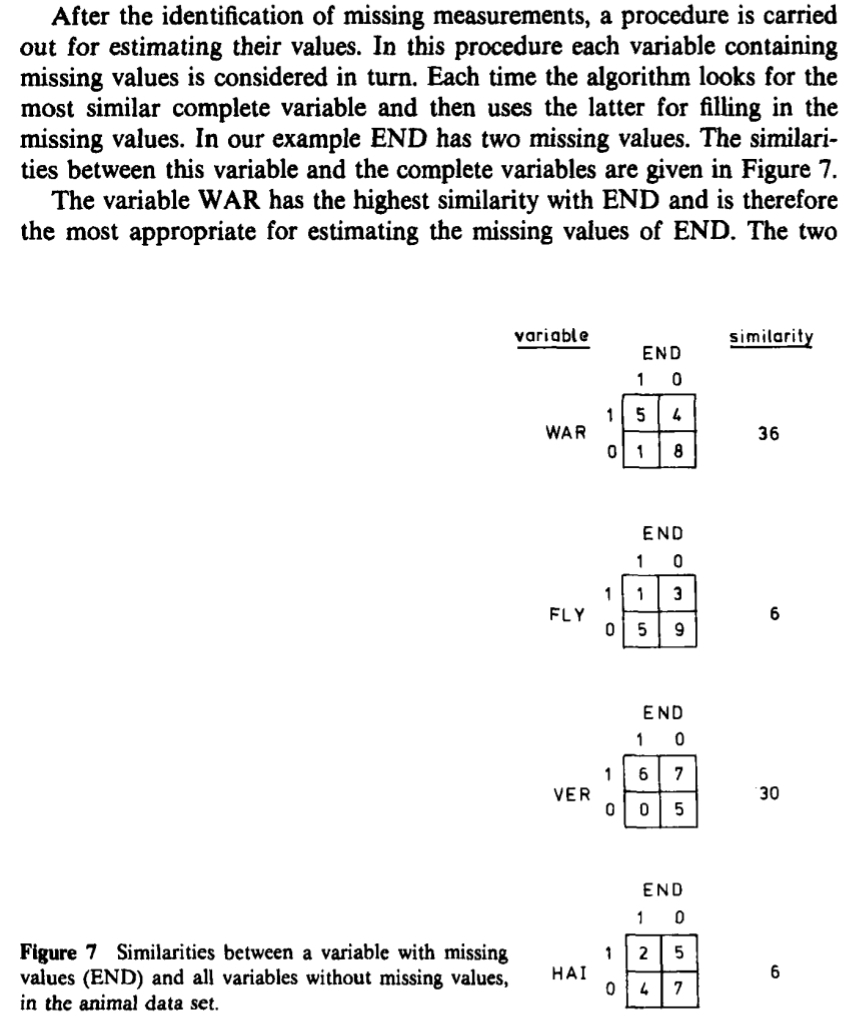
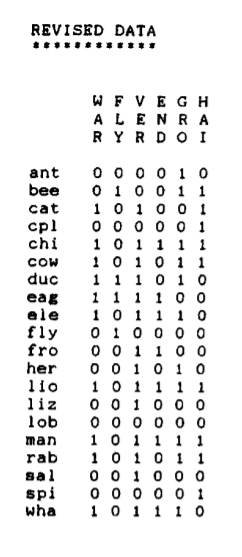
1-2\*a/(2\*a+b+c)

[1] 0.6666667

* The reason for such a large number of possible measures has to do with the apparent uncertainty as to how to deal with the count of zero-zero matches .
* The measues embedding are sometimes called symmetrical.
* The other measues are called assymmetrical.
* In some cases, of course, zero\_zero matches are completely equivalent to one–one matches, and therefore should be included in the calculated similarity measure.
* An example is gender, where there is no preference as to which of the two categories should be coded zero or one.
* But in other cases the inclusion or otherwise of is more problematic; for example, when the zero category corresponds to the genuine absence of some property, such as wings in a study of insects.

# Exercice 9

* Use the data set *animals* available in the package *cluster*.
* This data set was first used in this textbook [KAUFMAN, Leonard et ROUSSEEUW, Peter J. Finding groups in data: an introduction to cluster analysis. John Wiley & Sons, 2009](https://book4you.org/book/669277/d2cf59).
* Identify the missing measurements.
* Explain the way how KAUFMAN and ROUSSEEUW, pp. 296-297 treat the missing measurements.
* Compute a distance matrix for the completed data.
* Propose a graphical way to represent that distance matrix.
* Which group of animals look close?
* Change the method of calculating and observe if it has some effect of the graph.

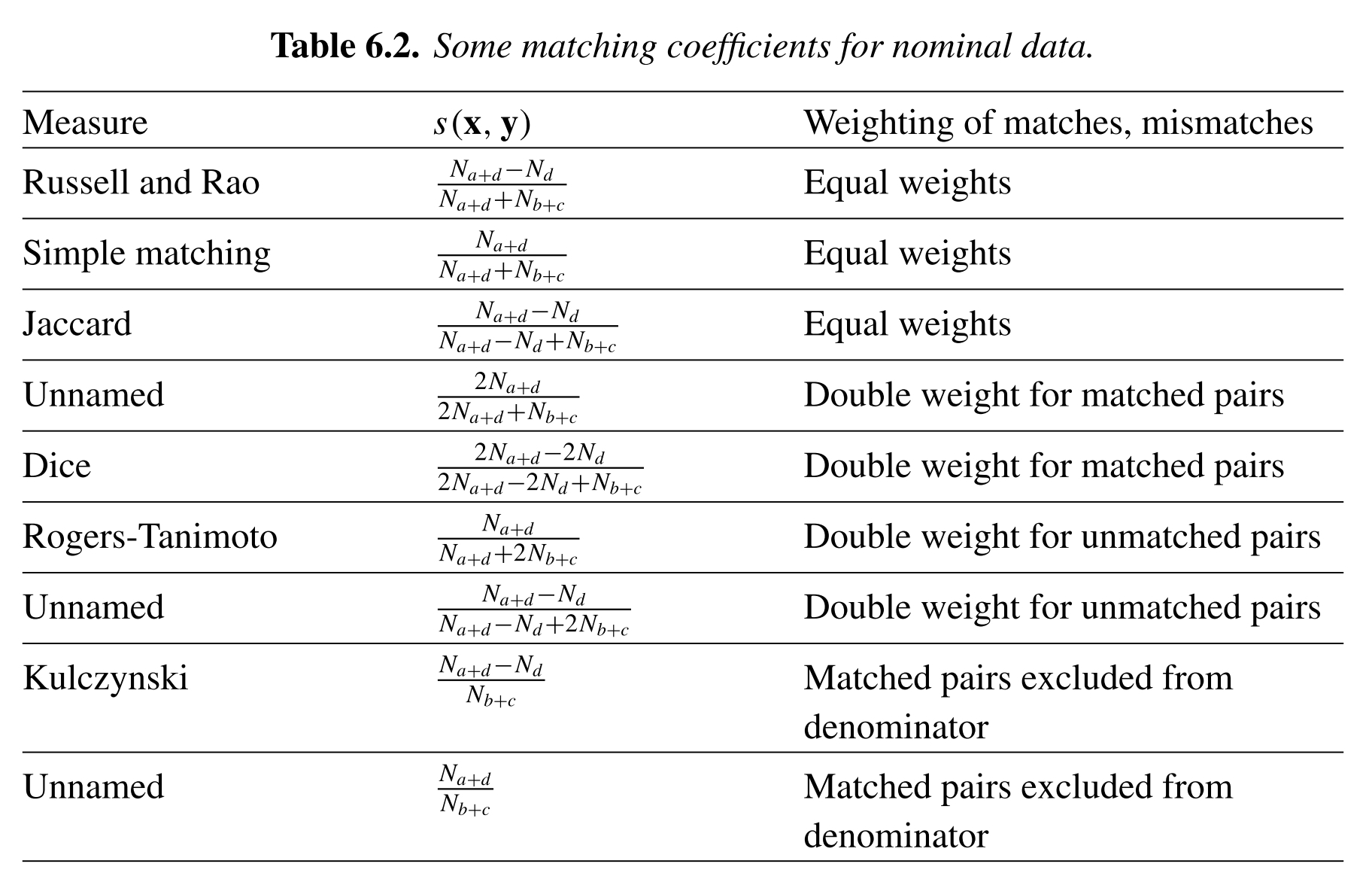
 

# Exercice 10

* Prove that the distances based on the Simple Matching coefficient and the Jaccard coefficient satisfy A3.
* Prove that the distances proposed by Gower and Legendre (1986) do not satisfy A3.
* Hint: Proofs and counterexamples have to be adapted from in the paper: [Gower, J. C., & Legendre, P. (1986). Metric and Euclidean properties of dissimilarity coefficients. *Journal of classification*, *3*(1), 5-48](https://booksc.org/book/6755353/c44198).

# Nominal variables

* We previously studied above binary variables which can only take on two states coded .
* We generalize this approach to nominal variables which may take on more than two states.
* Eye’s color may have for example four states: blue, brown, green, grey.
* Le be the number of states and code the outcomes as .
* We may choose and .
* These states are not ordered in any way
* One strategy would be creating a new binary variable for each of the nominal states.
* Then to put it equal to if the corresponding state occurs and to otherwise.
* After that, one could resort to one of the dissimilarity coefficients of the previous subsection.
* The most common way of measuring the similarity or dissimilarity between two objects through categorial variables is the simple matching approach.
* If are both nominal records for two individuals,
* Let define the function:
* Let be the number of attributes of the two individuals on which the two records match:
* Let be the number of attributes on which the two records do not match:
* Let be the number of attributes on which the two records match in a “not applicable” category.
* The distance corresponding to the simple matching approach is:
* Note that simple matching has exactly the same meaning as in the preceding section.

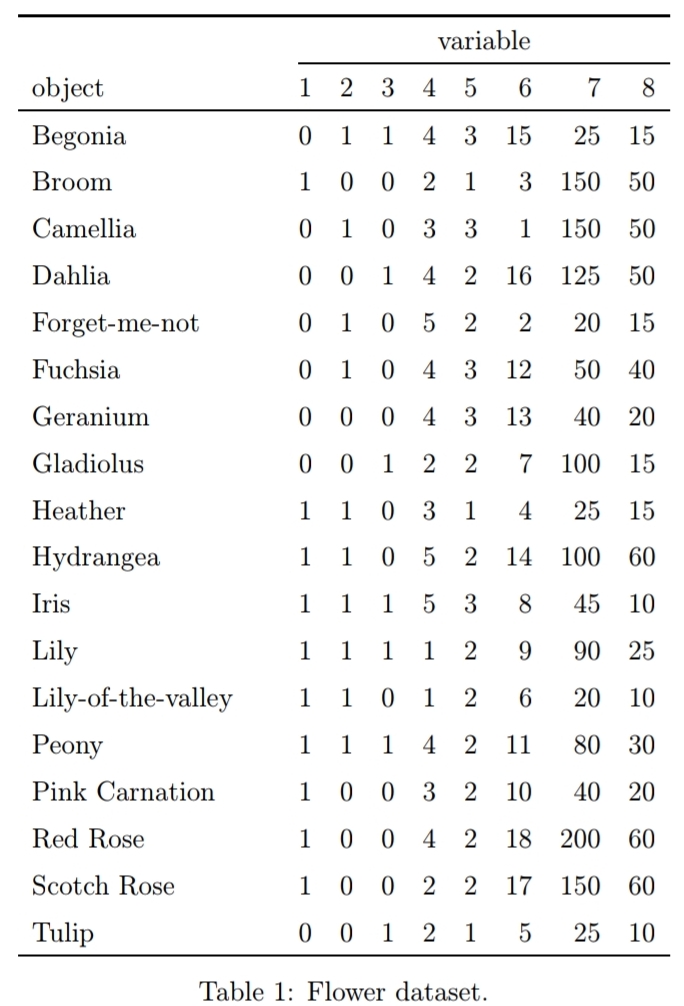


From: GAN et al

* For more details, see [GAN, Guojun, MA, Chaoqun, et WU, Jianhong. Data clustering: theory, algorithms, and applications. Society for Industrial and Applied Mathematics, 2020.](https://book4you.org/book/716744/c8c51f)

# Gower’s dissimilarity

* Gower’s coefficient is a dissimilarity measure specifically designed for handling mixed attribute types or variables.
* See: GOWER, John C. A general coefficient of similarity and some of its properties. *Biometrics*, 1971, p. 857-871.
* The coefficient is calculated as the weighted average of attribute contributions.
* Weights usually used only to indicate which attribute values could actually be compared meaningfully.
* The formula is:
* The wheight is put equal to when both measurements and are nonmissing,
* The number is the contribution of the th measure or variable to the dissimilarity measure.
* If the th measure is nominal, we take
* If the th measure is interval-scaled, we take instead:
* where is the range of variable over the available data.
* Consider the following data set:

 Data from: [Struyf, A., Hubert, M., & Rousseeuw, P. (1997). Clustering in an object-oriented environment. *Journal of Statistical Software*, *1*(4), 1-30](https://www.jstatsoft.org/article/view/v001i04).

* The dataset contains 18 flowers and 8 characteristics:

1. Winters: binary, indicates whether the plant may be left in the garden when it freezes.
2. Shadow: binary, shows whether the plant needs to stand in the shadow.
3. Tubers (Tubercule): asymmetric binary, distinguishes between plants with tubers and plants that grow in any other way.
4. Color: nominal, specifies the flower’s color (1=white, 2=yellow, 3= pink, 4=red, 5= blue).
5. Soil: ordinal, indicates whether the plant grows in dry (1), normal (2), or wet (3) soil.
6. Preference: ordinal, someone’s preference ranking, going from 1 to 18.
7. Height: interval scaled, the plant’s height in centimeters.
8. Distance: interval scaled, the distance in centimeters that should be left between the plants.

* The dissimilarity between Begonia and Broom (Genêt) can be calculated as follows:

 *Begonia*

 *Genêt*

library(cluster)  
library(dplyr)  
data <-flower %>%   
rename(Winters=V1,Shadow=V2,Tubers=V3,Color=V4,Soil=V5,Preference=V6,Height=V7,Distance=V8) %>%  
mutate(Winters=recode(Winters,"1"="Yes","0"="No"),  
 Shadow=recode(Shadow,"1"="Yes","0"="No"),  
 Tubers=recode(Tubers,"1"="Yes","0"="No"),  
 Color=recode(Color,"1"="white", "2"="yellow", "3"= "pink", "4"="red", "5"="blue"),  
 Soil=recode(Soil,"1"="dry", "2"="normal", "3"= "wet")  
 )   
row.names(data)=c("Begonia","Broom","Camellia","Dahlia","Forget-me-not","Fuchsia",  
 "Geranium", "Gladiolus","Heather","Hydrangea","Iris","Lily","Lily-of-the-valley",  
 "Peony","Pink Carnation","Red Rose","Scotch Rose","Tulip")

res=lapply(data,class)   
res=as.data.frame(res)  
res[1,] %>%   
knitr::kable()

Winters

Shadow

Tubers

Color

Soil

Preference

Height

Distance

factor

factor

factor

factor

ordered

ordered

numeric

numeric

flower[1:2,]

## V1 V2 V3 V4 V5 V6 V7 V8  
## 1 0 1 1 4 3 15 25 15  
## 2 1 0 0 2 1 3 150 50

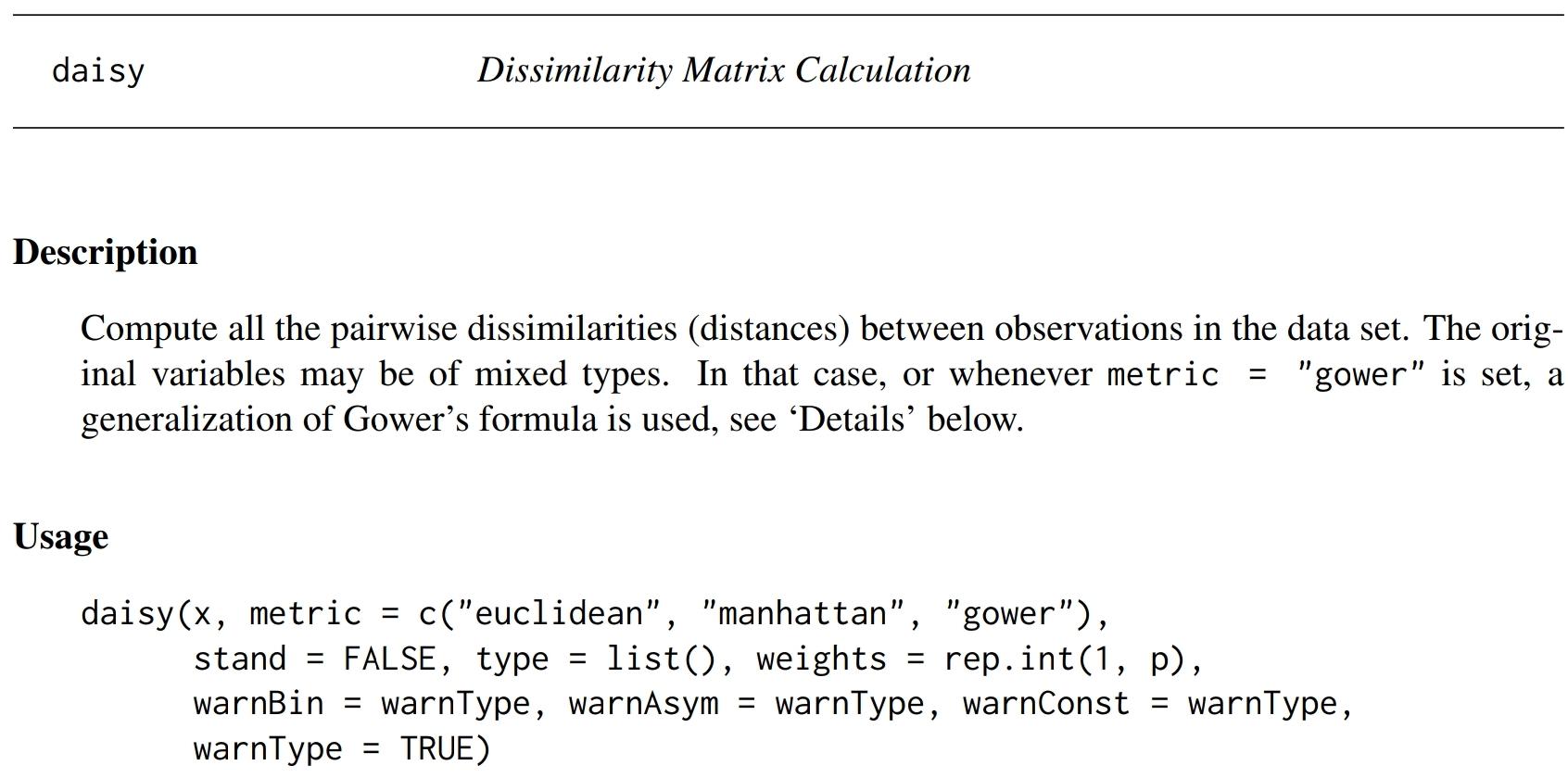
max(data$Height)-min(data$Height)

## [1] 180

max(data$Distance)-min(data$Distance)

## [1] 50

# Daisy function

 [Cluster package description available at this link](https://www.google.com/search?q=daisy+package+r&oq=daisy+&aqs=chrome.1.69i57j35i19i39j35i39j69i59j46i67i433.4466j0j7&client=tablet-android-samsung-nf-rev1&sourceid=chrome-mobile&ie=UTF-8).

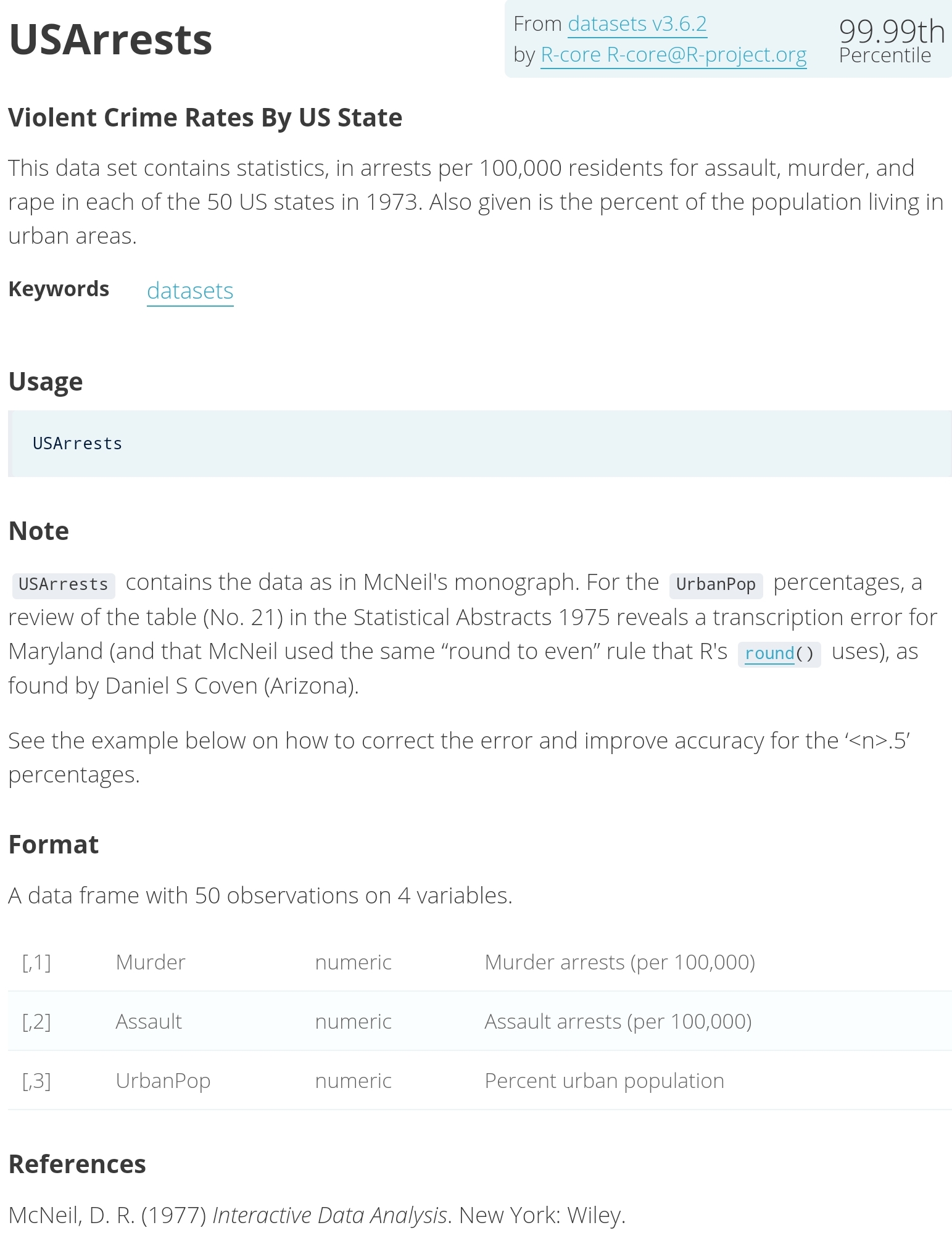
library(cluster)  
(abs(1-0)+abs(0-1)+abs(0-1)+1+abs(1-3)/2+abs(3-15)/17+abs(150-25)/180+abs(50-15)/50)/8

## [1] 0.8875408

dist<-daisy(data[,1:8],metric = "Gower")  
as.matrix(dist)[1:2,1:2]

## Begonia Broom  
## Begonia 0.0000000 0.8875408  
## Broom 0.8875408 0.0000000

# More on distance matrix computation



* We use a subset of the data by taking 15 random rows among the 50 rows in the data set.
* We are using the function sample().
* We standardize the data using the function scale().

stargazer(USArrests,header=TRUE, type='html',summary=FALSE,digits=1)

Murder

Assault

UrbanPop

Rape

Alabama

13.2

236

58

21.2

Alaska

10

263

48

44.5

Arizona

8.1

294

80

31

Arkansas

8.8

190

50

19.5

California

9

276

91

40.6

Colorado

7.9

204

78

38.7

Connecticut

3.3

110

77

11.1

Delaware

5.9

238

72

15.8

Florida

15.4

335

80

31.9

Georgia

17.4

211

60

25.8

Hawaii

5.3

46

83

20.2

Idaho

2.6

120

54

14.2

Illinois

10.4

249

83

24

Indiana

7.2

113

65

21

Iowa

2.2

56

57

11.3

Kansas

6

115

66

18

Kentucky

9.7

109

52

16.3

Louisiana

15.4

249

66

22.2

Maine

2.1

83

51

7.8

Maryland

11.3

300

67

27.8

Massachusetts

4.4

149

85

16.3

Michigan

12.1

255

74

35.1

Minnesota

2.7

72

66

14.9

Mississippi

16.1

259

44

17.1

Missouri

9

178

70

28.2

Montana

6

109

53

16.4

Nebraska

4.3

102

62

16.5

Nevada

12.2

252

81

46

New Hampshire

2.1

57

56

9.5

New Jersey

7.4

159

89

18.8

New Mexico

11.4

285

70

32.1

New York

11.1

254

86

26.1

North Carolina

13

337

45

16.1

North Dakota

0.8

45

44

7.3

Ohio

7.3

120

75

21.4

Oklahoma

6.6

151

68

20

Oregon

4.9

159

67

29.3

Pennsylvania

6.3

106

72

14.9

Rhode Island

3.4

174

87

8.3

South Carolina

14.4

279

48

22.5

South Dakota

3.8

86

45

12.8

Tennessee

13.2

188

59

26.9

Texas

12.7

201

80

25.5

Utah

3.2

120

80

22.9

Vermont

2.2

48

32

11.2

Virginia

8.5

156

63

20.7

Washington

4

145

73

26.2

West Virginia

5.7

81

39

9.3

Wisconsin

2.6

53

66

10.8

Wyoming

6.8

161

60

15.6

set.seed(123)  
ss <- sample(1:50,15)   
df <- USArrests[ss, ]   
df.scaled <- scale(df)   
stargazer(df.scaled,header=TRUE, type='html',summary=FALSE,digits=1)

Murder

Assault

UrbanPop

Rape

New Mexico

0.6

1.0

0.2

0.6

Iowa

-1.7

-1.5

-0.7

-1.4

Indiana

-0.5

-0.9

-0.1

-0.5

Arizona

-0.2

1.1

0.9

0.5

Tennessee

1.0

-0.1

-0.5

0.1

Texas

0.9

0.1

0.9

-0.04

Oregon

-1.0

-0.4

0.01

0.3

West Virginia

-0.8

-1.3

-2.0

-1.6

Missouri

-0.01

-0.2

0.2

0.2

Montana

-0.8

-1.0

-1.0

-0.9

Nebraska

-1.2

-1.0

-0.3

-0.9

California

-0.01

0.9

1.7

1.4

South Carolina

1.3

1.0

-1.3

-0.3

Nevada

0.8

0.7

1.0

2.0

Florida

1.6

1.6

0.9

0.6

* The R functions for computing distances.

1. dist() function accepts only numeric data.
2. get\_dist() function [factoextra package] accepts only numeric data. it supports correlation-based distance measures.
3. daisy() function [cluster package] is able to handle other variable types (nominal, ordinal, …).

* Remark: All these functions compute distances between rows of the data.
* Remark: If we want to compute pairwise distances between variables, we must transpose the data to have variables in the rows.
* We first compute Euclidian distances

dist.eucl <- dist(df.scaled, method = "euclidean",upper = TRUE)  
  
  
stargazer(as.data.frame(as.matrix(dist.eucl)[1:3, 1:3]),header=TRUE, type='html',summary=FALSE,digits=1)

New Mexico

Iowa

Indiana

New Mexico

0

4.1

2.5

Iowa

4.1

0

1.8

Indiana

2.5

1.8

0

round(sqrt(sum((df.scaled["New Mexico",]-df.scaled["Iowa",])^2)),1)

[1] 4.1

round(sqrt(sum((df.scaled["New Mexico",]-df.scaled["Indiana",])^2)),1)

[1] 2.5

round(sqrt(sum((df.scaled["Iowa",]  
-df.scaled["Indiana",])^2)),1)

[1] 1.8

* We also compute correlation based distances.

library("factoextra")  
dist.cor <- get\_dist(df.scaled, method = "pearson")  
round(as.matrix(dist.cor)[1:3, 1:3], 1)

## New Mexico Iowa Indiana  
## New Mexico 0.0 1.7 2.0  
## Iowa 1.7 0.0 0.3  
## Indiana 2.0 0.3 0.0

round(1-cor(df.scaled["New Mexico",],df.scaled["Iowa",]),1)

## [1] 1.7

round(1-cor(df.scaled["New Mexico",],df.scaled["Indiana",]),1)

## [1] 2

round(1-cor(df.scaled["Iowa",],df.scaled["Indiana",]),1)

## [1] 0.3

# Visualizing distance matrices

* A simple solution for visualizing the distance matrices is to use the function fviz\_dist() [factoextra package].
* Other specialized methods will be described later on.

library(factoextra)  
fviz\_dist(dist.eucl)

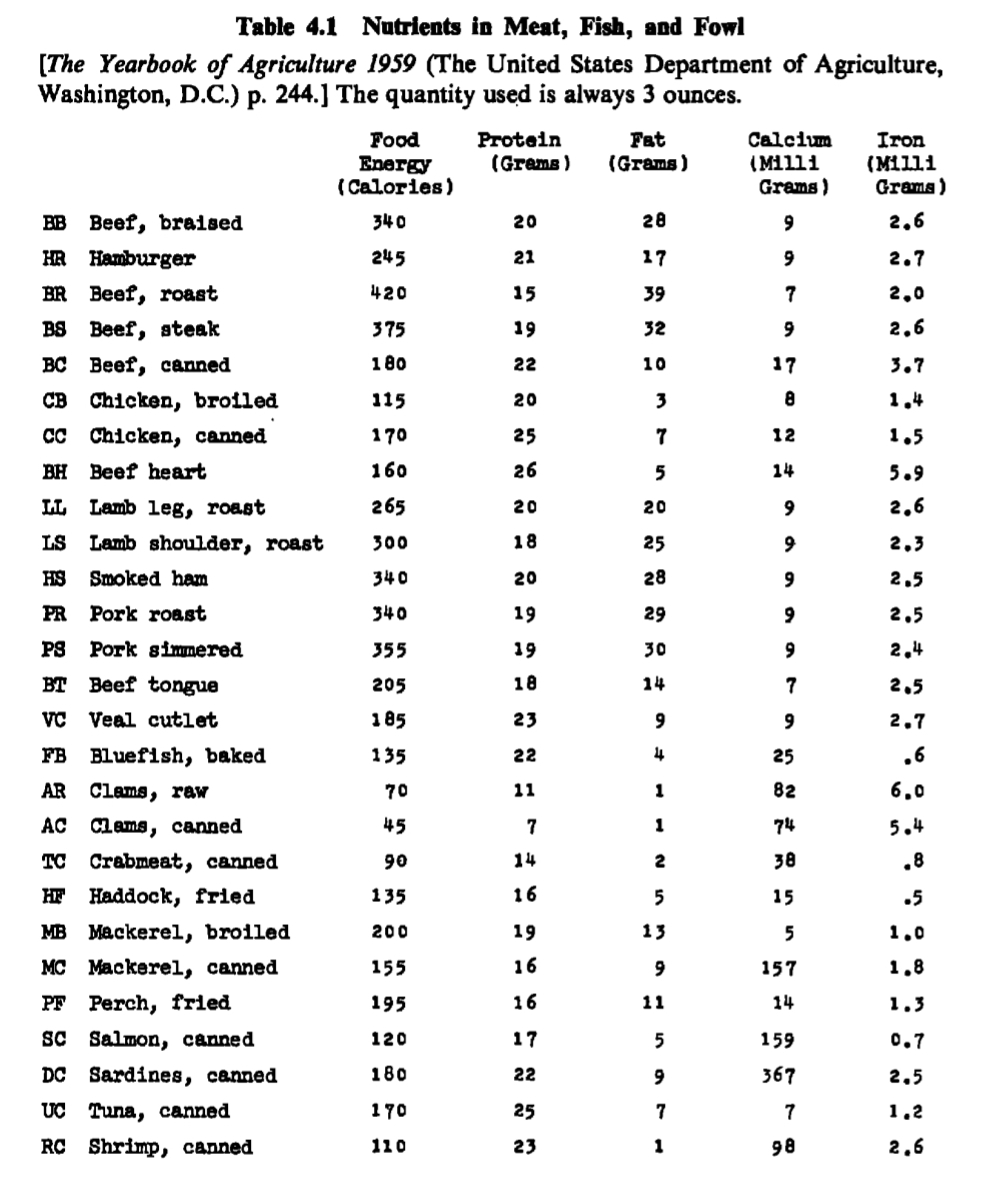


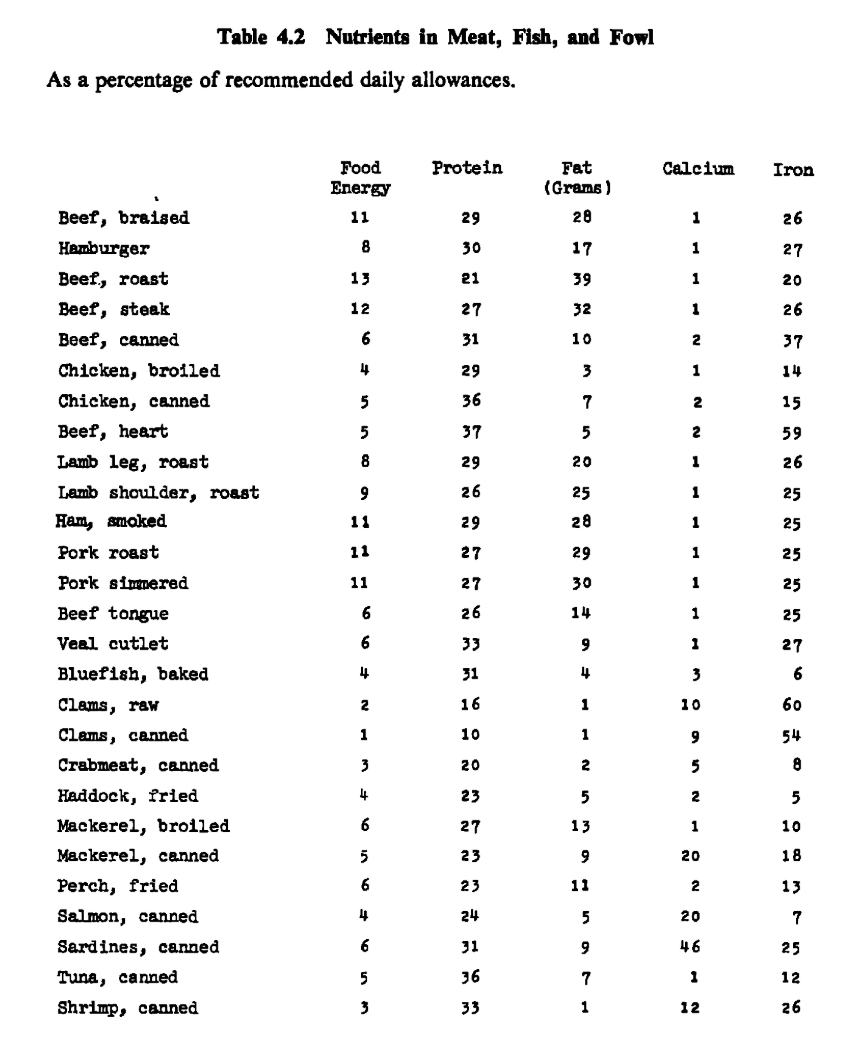
# Partitioning Clustering

* Partitioning clustering are clustering methods used to classify observations within a data set, into multiple groups based on their similarity.
* The algorithms require the analyst to specify the number of clusters to be generated.
* We describe the commonly used partitioning clustering, including:

1. K-means clustering (MacQueen, 1967), in which, each cluster is represented by the center or means of the data points belonging to the cluster. The K-means method is sensitive to anomalous data points and outliers.
2. K-medoids clustering or PAM (Partitioning Around Medoids, Kaufman & Rousseeuw, 1990), in which, each cluster is represented by one of the objects in the cluster. PAM is less sensitive to outliers compared to k-means.
3. CLARA algorithm (Clustering Large Applications), which is an extension to PAM adapted for large data sets.

# K-Means Clustering

* The description of the algorithm is based on:
* HARTIGAN, John A. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.
* The data used by the author are provided below. 
* The principal nutrients in meat, fish, and fowl are listed.
* Recall that 1oz= 28.34952g.
* Estimated daily dietary allowances are: food energy (3200 cal), protein (70 g), calcium (0.8 g), and iron (10 mg).
* Table 4.2 convents the variables (with the exception of Fat) in percentage of food delivery.



Data

* For e.g., the first (BB) ligne is obtained in the following way:
* .
* .
* .
* .
* An argument could be made that iron is less important than calories or protein and so should be given less weight or ignored entirely.
* There are objects and clusters, .
* Our purpose is to partition the objects (here foods) so that objects within clusters are close and objects in different clusters are distant.
* Each cluster contains at least one object and each object belongs to only one cluster.
* There is a very large number of possible partitions.

# Exercice 11

* What is the number of possible partitions?
* Hint use the Stirling numbers of the second kind [Wikipedia](https://en.m.wikipedia.org/wiki/Stirling_numbers_of_the_second_kind).
* Example of functions in R.

library(multicool)

## Loading required package: Rcpp

Stirling2(8,3)

## [1] 966

Stirling2(8,1)

## [1] 1

Stirling2(3,2)

## [1] 3

# K-Means

* The discordance between the data and a given partition is denoted by .
* We use the technique of local optimization.
* A neighborhood of partitions is defined for each ption.
* Starting from an initial partition, search through a set of partitions at each step.
* Move from the partition to a partition in its neighborhood for which is minim.
* If the neighborhoods are very large, it is cheaper computationally to move to the first partition discovered in the neighborhood where is reduced from its present value.
* A number of stopping rules are possible.
* For example, the search stops when is not reduced by movement to the neighborhood.
* The present partition is locally optimal in that it is the best partition in its neighborhood.
* Consider partitions of the five () beef foods into three clusters ().
* Totally, there are 25 such partitions.

library(gmp)  
Stirling2(5,3)

## Big Integer ('bigz') :  
## [1] 25

* A plausible neighborhood for a partition is the set of partitions obtained by transferring an object from one cluster to another.
* For the partition (BB BR) (BS) (BC BH), the neighborhood consists of the following ten partitions:

1. (BR) (BB BS) (BC BH)
2. (BR) (BS) (BB BC BH)
3. (BB) (BR BS) (BC BH)
4. (BB) (BS) (BR BC BH)
5. (BB BR BS) O (BC BH)
6. (BB BR) O (BS BC BH)
7. (BB BR BC) (BS) (BH)
8. (BB BR) (BS BC) (BH)
9. (BB BR BH) (BS) (BC)
10. (BB BR) (BS BH) (BC)

# K-Means Algorithm

* Let the vector of values for the object ,
* The variables are assumed scaled.
* The partition has disjoint clusters: , which are the indices of the objects in the various clusters.
* Let be the number of objects in cluster .
* Each of the objects lies in just one of the clusters.
* Note that .
* The vector of means over the objects in cluster is denoted by , with
* where
* The distance between the object and the cluster is , where is taken to be the Euclidian distance
* where is the Euclidian norm.
* The error of the partition is taken to be
* Alternatively, we have

where is the index of the cluster of object .

* The general procedure is to search for a partition with a small error by moving cases from one cluster to another.
* The search ends when no such movement reduces .
* **STEP 1**. Assume initial clusters. Compute the cluster means and the initial error .
* **STEP 2**. For the first object, compute for every cluster
* It corresponds to the error variation in transferring the first object from cluster to which it belongs to cluster .
* If the minimum of this quantity over all is negative, transfer the first case from cluster to this minimal .
* Adjust the cluster means of and the minimal and add the error variation (which is negative) to .
* **STEP 3**. Repeat STEP 2 for each object such that .
* **STEP 4**. lf no movement of an object from one cluster to another occurs for any case, stop. Otherwise, return to STEP 2.

# Exercice 12

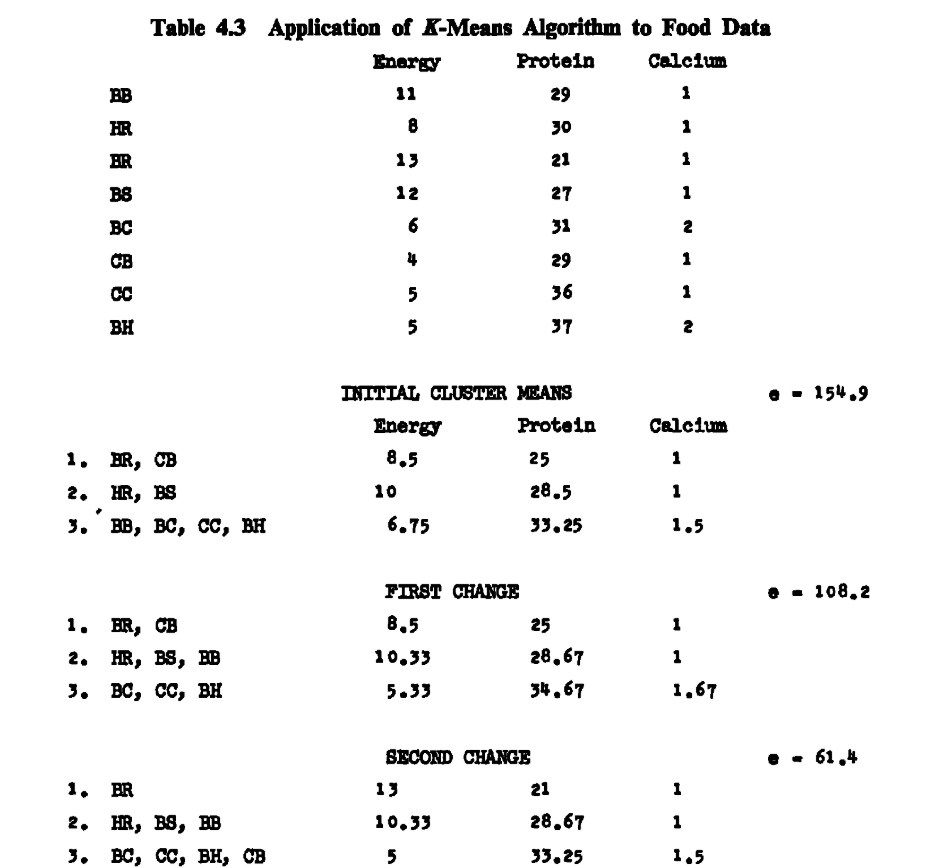
Prove that the error variation is indeed given by:

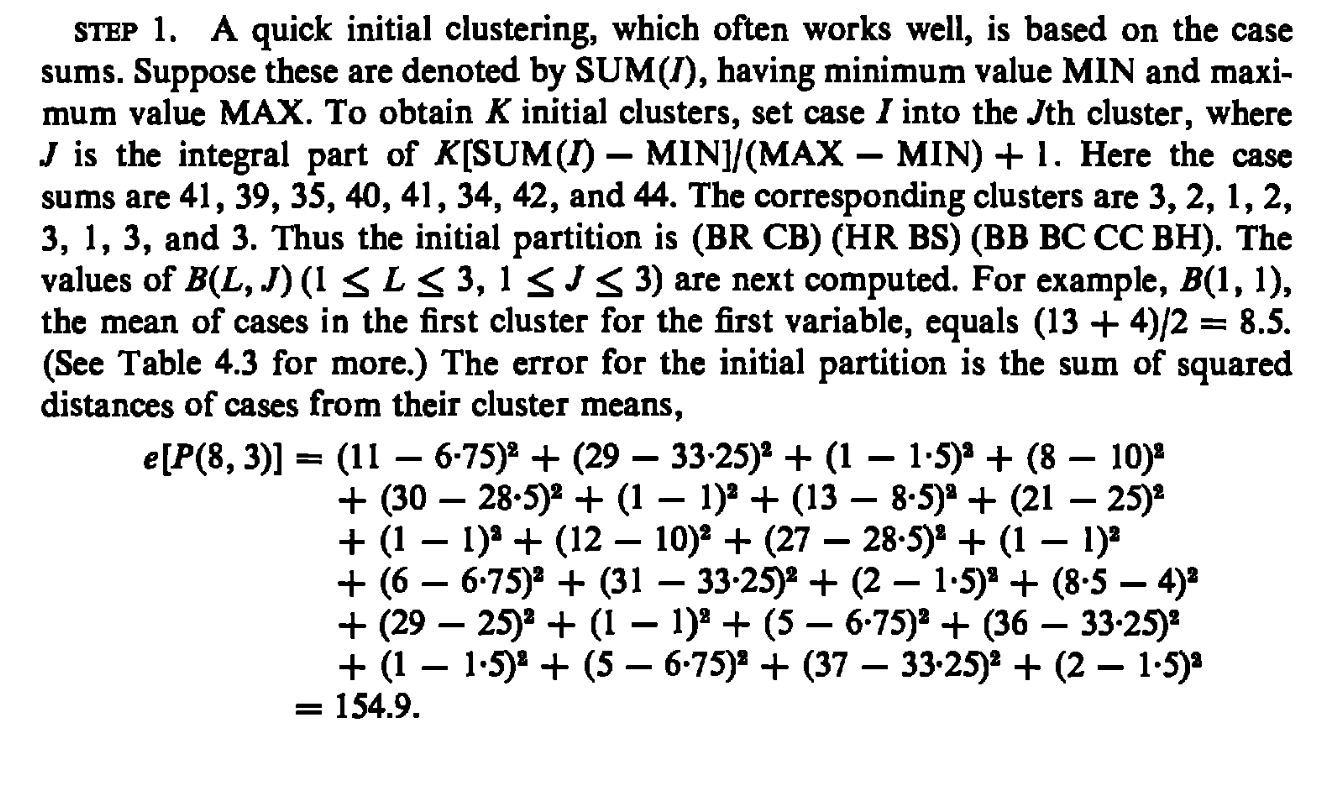
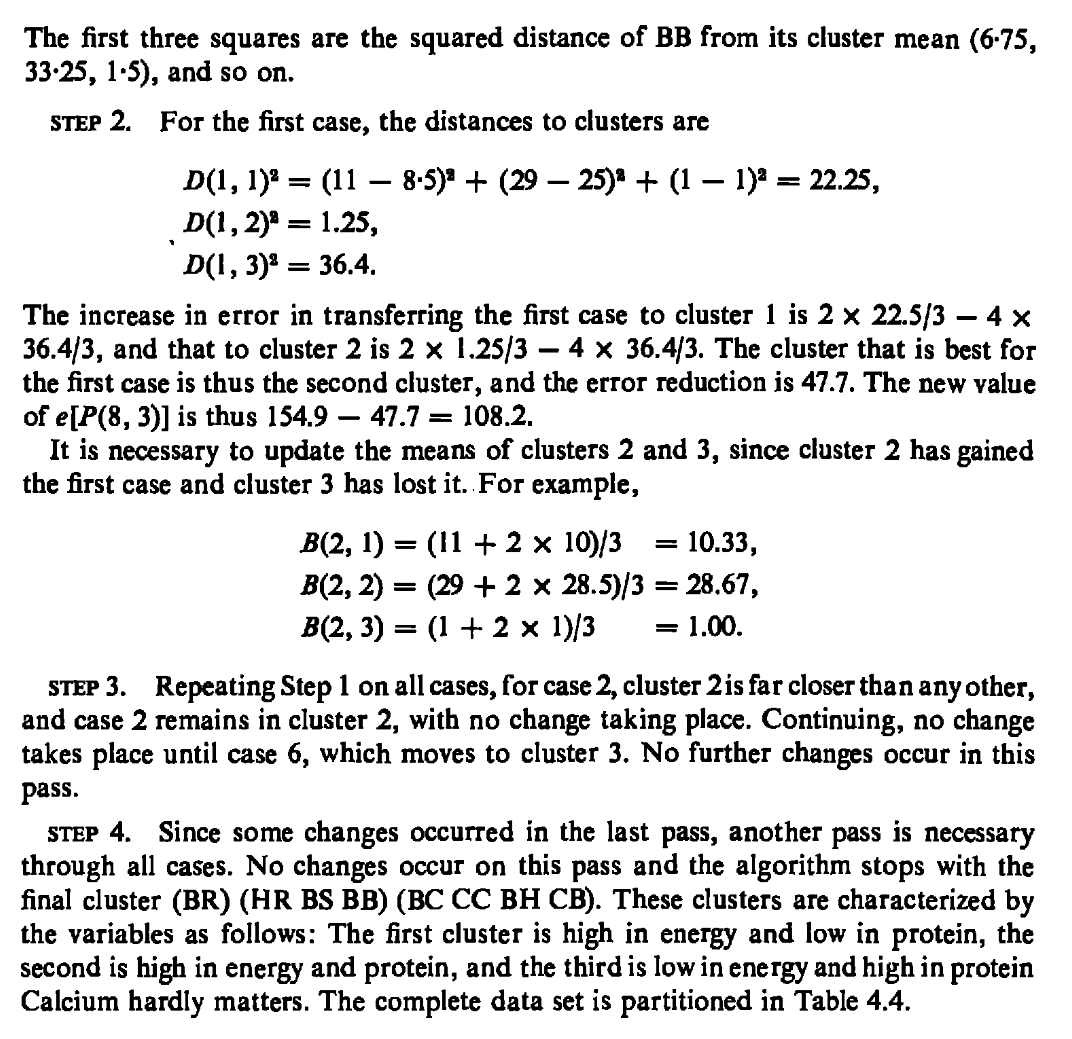
# K-MEANS APPLIED TO FOOD NUTRIENT DATA

* Only the first eight foods will be considered.
* Only three variables, food energy, protein, and calcium as a percentage of recommended daily allowances are used.
* The eight foods are partitioned into three clusters ().

# Exercice 13

* Explain in details the k-means algorithm based on the following pages of Hartigan (1975).



#library("cluster.datasets")  
#write.csv(rda.meat.fish.fowl.1959,"Hartigandat%a1.csv")  
df<-read.csv("Hartigandata1.csv")  
print(df)

## X name energy protein fat calcium iron  
## 1 1 Braised beef 11 29 28 1 26  
## 2 2 Hamburger 8 30 17 1 27  
## 3 3 Roast beef 18 21 39 1 20  
## 4 4 Beefsteak 12 27 32 1 26  
## 5 5 Canned beef 6 31 10 2 37  
## 6 6 Broiled chicken 8 29 3 1 14  
## 7 7 Canned chicken 5 36 7 2 15  
## 8 8 Beef heart 5 37 5 2 59  
## 9 9 Roast lamb leg 8 29 20 1 26  
## 10 10 Roast lamb shoulder 9 26 25 1 23  
## 11 11 Smoked ham 11 29 28 1 25  
## 12 12 Pork roast 11 27 29 1 25  
## 13 13 Pork simmered 11 27 30 1 24  
## 14 14 Beef tongue 6 26 14 1 25  
## 15 15 Veal cutlet 6 33 9 1 27  
## 16 16 Baked bluefish 4 31 4 3 6  
## 17 17 Raw clams 2 16 1 10 60  
## 18 18 Canned clams 1 10 1 9 54  
## 19 19 Canned crabmeat 3 20 2 5 8  
## 20 20 Fried haddock 4 23 5 2 5  
## 21 21 Broiled mackerel 6 27 13 1 10  
## 22 22 Canned mackerel 5 23 9 20 18  
## 23 23 Fried perch 6 23 11 2 13  
## 24 24 Canned salmon 4 24 5 20 7  
## 25 25 Canned sardines 6 31 9 46 25  
## 26 26 Canned tuna 5 36 7 1 12  
## 27 27 Canned shrimp 3 33 1 12 26

df<-df[1:8,c(3,4,6)]  
df

## energy protein calcium  
## 1 11 29 1  
## 2 8 30 1  
## 3 18 21 1  
## 4 12 27 1  
## 5 6 31 2  
## 6 8 29 1  
## 7 5 36 2  
## 8 5 37 2

# The data set contains some errors   
df[3,1]<-13 # Error in line 3  
df[6,1]<-4 # Error at line 6  
df[7,3]<-1 # Error at line 7  
df

## energy protein calcium  
## 1 11 29 1  
## 2 8 30 1  
## 3 13 21 1  
## 4 12 27 1  
## 5 6 31 2  
## 6 4 29 1  
## 7 5 36 1  
## 8 5 37 2

rownames(df)<-c("BB","HR","BR","BS","BC","CB","CC","BH")  
df

## energy protein calcium  
## BB 11 29 1  
## HR 8 30 1  
## BR 13 21 1  
## BS 12 27 1  
## BC 6 31 2  
## CB 4 29 1  
## CC 5 36 1  
## BH 5 37 2

colnames(df)<-c("Energy","Protein","Calcium")  
df

## Energy Protein Calcium  
## BB 11 29 1  
## HR 8 30 1  
## BR 13 21 1  
## BS 12 27 1  
## BC 6 31 2  
## CB 4 29 1  
## CC 5 36 1  
## BH 5 37 2

# More on kmeans() output

km.res<-kmeans(df[1:8,],3,iter.max = 100)  
km.res$cluster

## BB HR BR BS BC CB CC BH   
## 3 3 2 3 1 1 1 1

km.res$centers

## Energy Protein Calcium  
## 1 5.00000 33.25000 1.5  
## 2 13.00000 21.00000 1.0  
## 3 10.33333 28.66667 1.0

km.res$totss

## [1] 267.5

sum((df[1:8,]$Energy-mean(df[1:8,]$Energy))^2)+  
sum((df[1:8,]$Protein-mean(df[1:8,]$Protein))^2)+  
sum((df[1:8,]$Calcium-mean(df[1:8,]$Calcium))^2)

## [1] 267.5

7\*var(df[1:8,]$Energy)+7\*var(df[1:8,]$Protein)+7\*var(df[1:8,]$Calcium)

## [1] 267.5

km.res$withinss

## [1] 47.75000 0.00000 13.33333

km.res$tot.withinss

## [1] 61.08333

km.res$betweenss

## [1] 206.4167

km.res$size

## [1] 4 1 3

km.res$iter

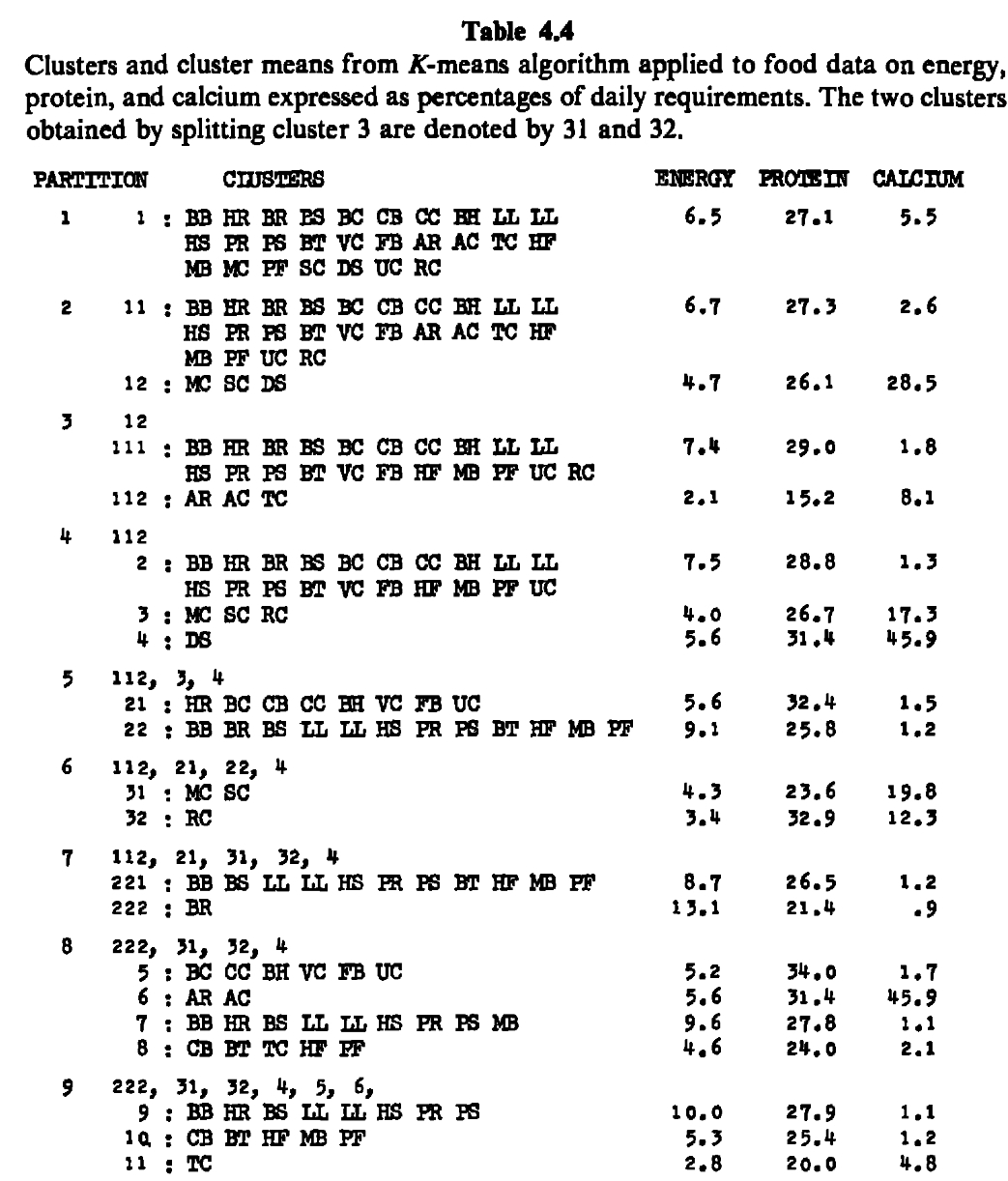
## [1] 1

# Exercice 14

* Produce a heatmap for the data by rearranging the lines according to the found clusters (k=3).

# Exercice 15

* Produce the Table 4.4 of Hartigan (1975).



# Exercice 16

* Use ggplot2 to draw an x-y plot with the number of clusters on the x-axis and the error on the y-axis.

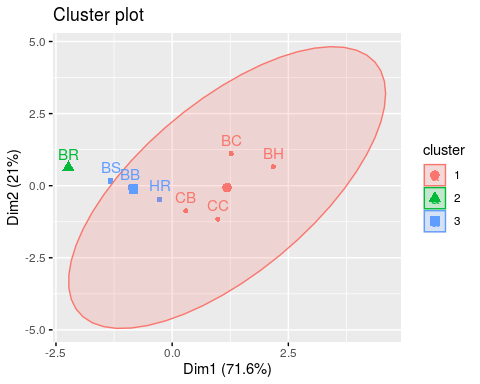
**Remark:** The location of a knee is generally considered as an appropriate number of clusters.

# Visualizing k-means results

* The results are visualized using fviz\_cluster() function.
* It draws a scatter plot of data points colored by cluster numbers.
* If the data contains more than 2 variables, the Principal Component Analysis (PCA) algorithm is used to reduce the dimensionality of the data.
* The first two principal dimensions are used to plot the data.

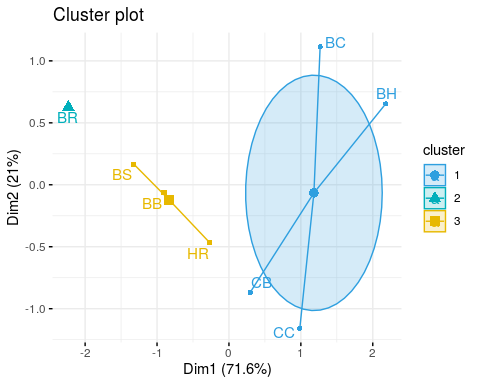
library(ggplot2)  
library(factoextra)  
  
fviz\_cluster(km.res, df, ellipse.type = "norm")

## Too few points to calculate an ellipse  
## Too few points to calculate an ellipse



fviz\_cluster(km.res, data = df[1:8,],  
palette = c("#2E9FDF", "#00AFBB", "#E7B800"),  
ellipse.type = "euclid", # Concentration ellipse  
star.plot = TRUE, # Add segments from centroids to items  
repel = TRUE, # Avoid label overplotting (slow)  
ggtheme = theme\_minimal()  
)

## Too few points to calculate an ellipse  
## Too few points to calculate an ellipse



pca=prcomp(df[1:8,], scale = TRUE)  
summary(pca)

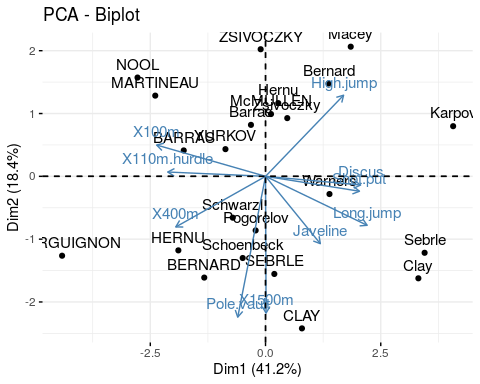
## Importance of components:  
## PC1 PC2 PC3  
## Standard deviation 1.4655 0.7938 0.47119  
## Proportion of Variance 0.7159 0.2100 0.07401  
## Cumulative Proportion 0.7159 0.9260 1.00000

# More on PCA

data(decathlon2)  
decathlon.active <- decathlon2[1:23, 1:10]  
res.pca <- prcomp(decathlon.active, scale = TRUE)  
summary(res.pca)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 2.0308 1.3559 1.1132 0.90523 0.83759 0.65029 0.55007  
## Proportion of Variance 0.4124 0.1839 0.1239 0.08194 0.07016 0.04229 0.03026  
## Cumulative Proportion 0.4124 0.5963 0.7202 0.80213 0.87229 0.91458 0.94483  
## PC8 PC9 PC10  
## Standard deviation 0.52390 0.39398 0.3492  
## Proportion of Variance 0.02745 0.01552 0.0122  
## Cumulative Proportion 0.97228 0.98780 1.0000

fviz\_pca\_biplot(res.pca)



# Silhouette

* Assume the data are clustered into clusters.
* For , let:
* be the mean distance between and all the points of the same cluster.
* If , we set .
* It is interpreted as a measure of how well is assigned to its cluster (smaller the value, better is the assignment).
* Let also
* be the “neighboring cluster” of .
* We now define a *silhouette* of as
* and
* Alternatively, we have
* From the above definition it is clear that
* For to be close to 1 we require .
* If is close to , is more appropriately clustered in its neighbouring cluster.
* An near zero means that the is on the border of two natural clusters.

# Silhouette example

Table with kable

cluster

neighbor

sil\_width

HR

1

2

0.4658716

BC

1

3

0.5168427

CB

1

3

0.5312237

BB

2

1

-0.0053370

BR

2

1

0.3785264

BS

2

1

0.3920856

CC

3

1

0.7764139

BH

3

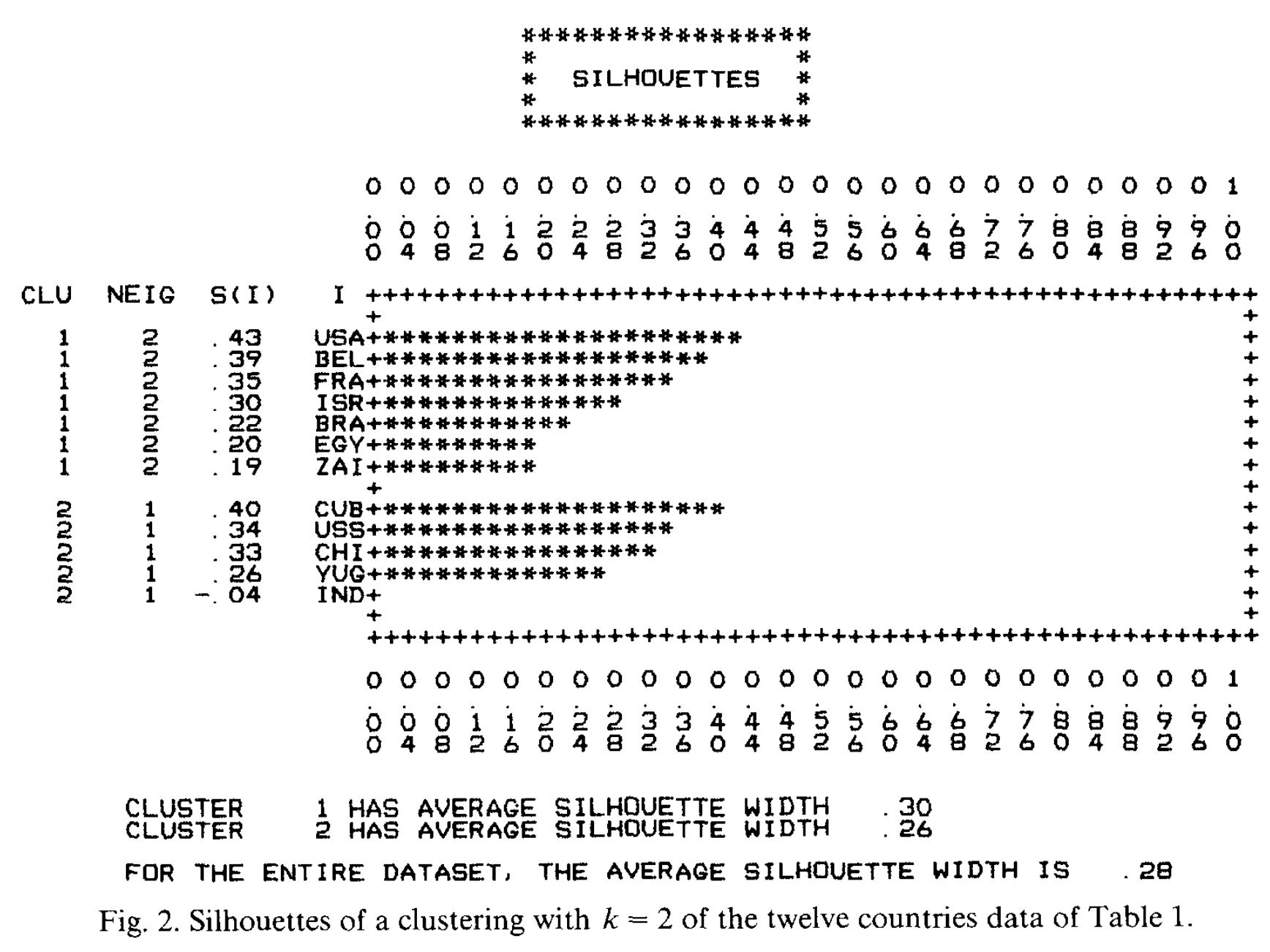
1

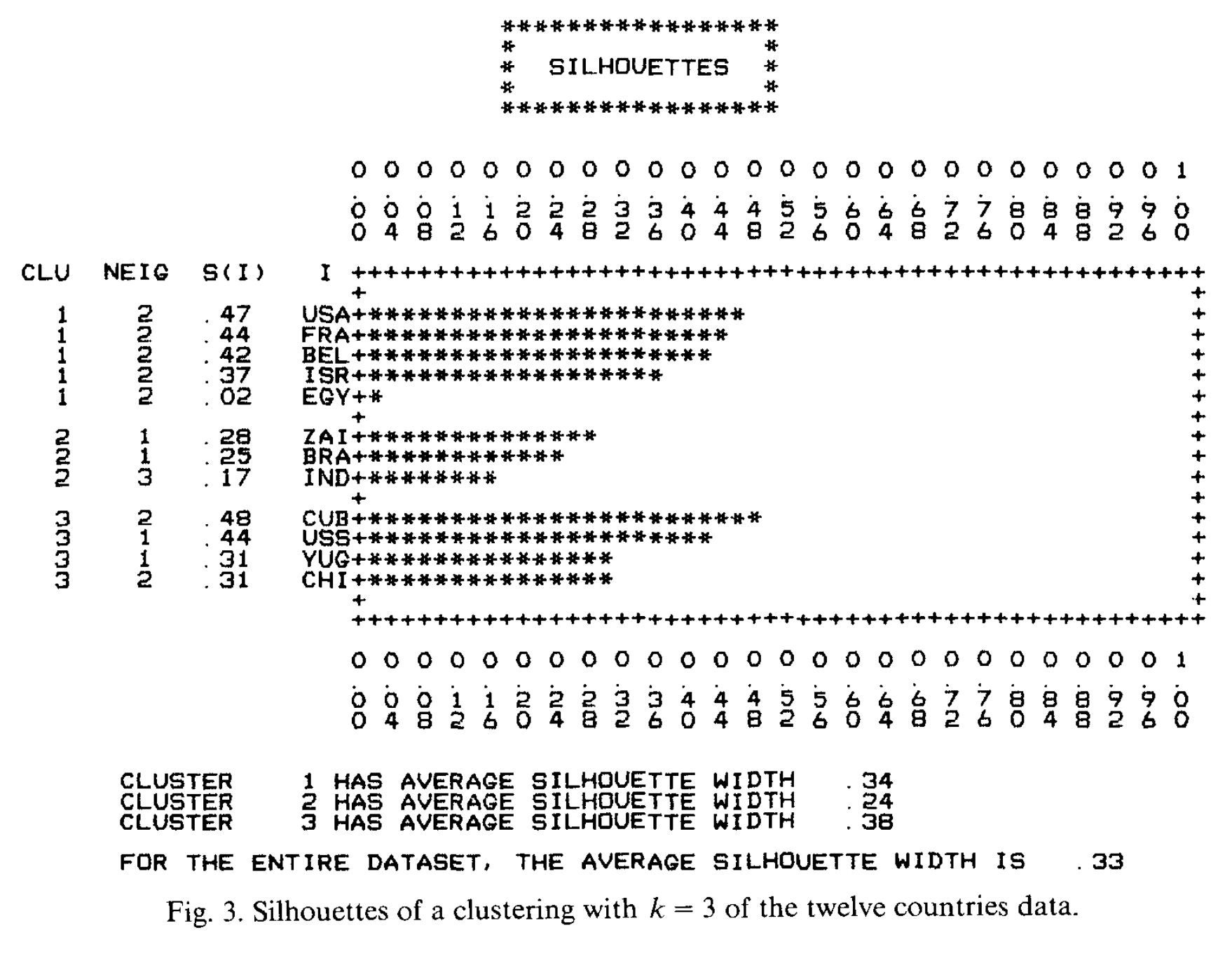
0.8061654

# Exercice 17

* We are using the data of the paper which introduces the **Silhouette** method some years ago: [ROUSSEEUW, Peter J. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics, 1987, vol. 20, p. 53-65.](https://booksc.org/book/3006616/528309).

 \* The author obtain the following two figures.





* Your task is to produce the same figures using modern dataviz tools.
* What did you learn from theses graphs?

# Exercice 18

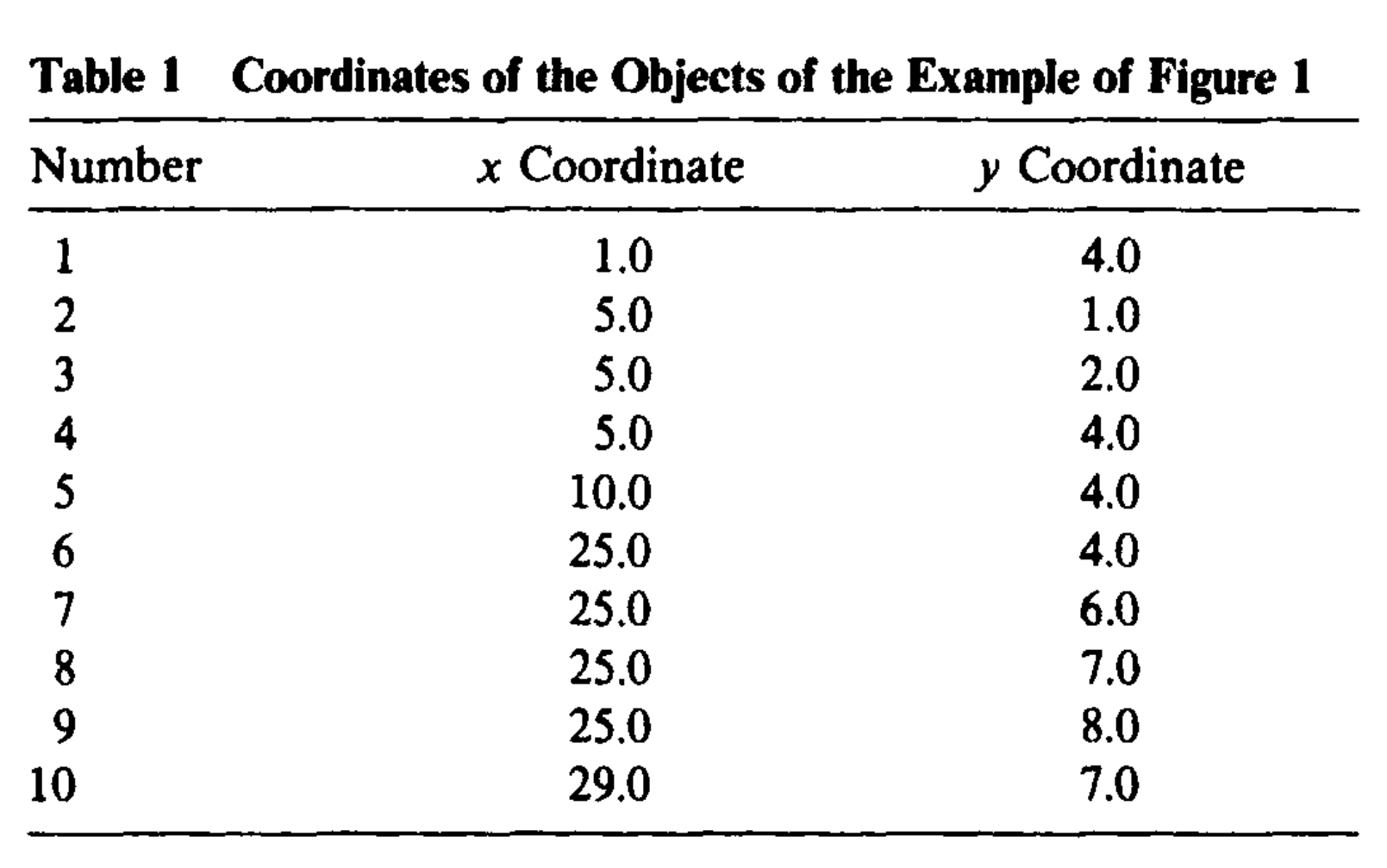
* Provide a Silhouette graph for the Hartigan (75) data

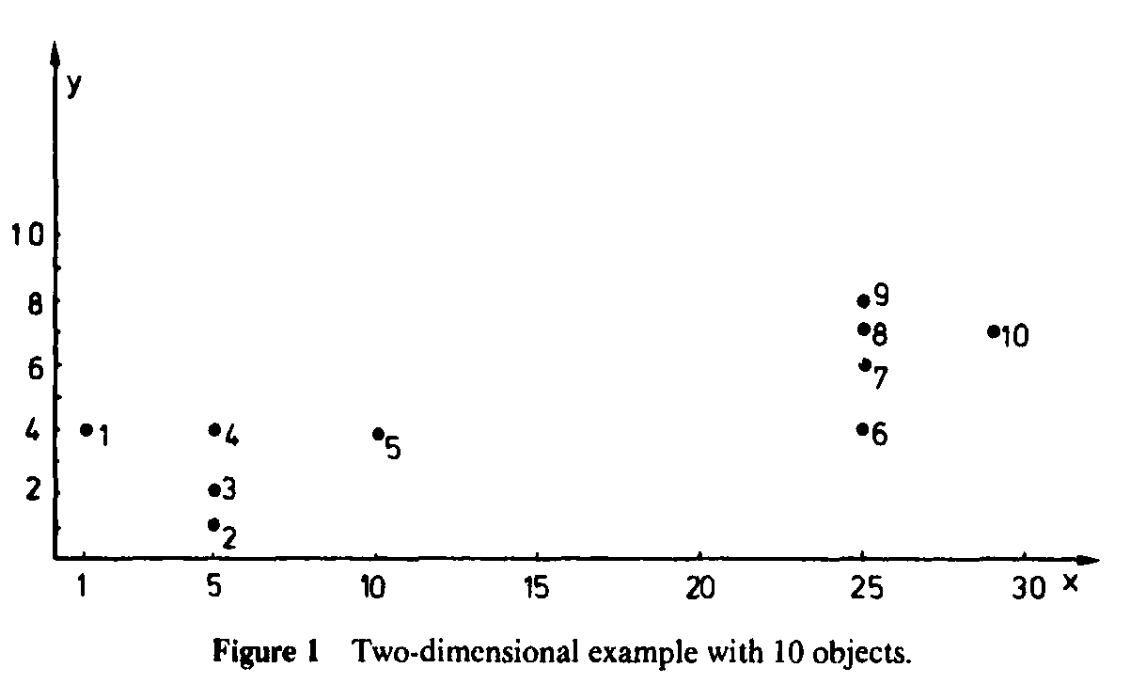
# Exercice 19

* Provide a k-means clustering analysis of the US Arrest data

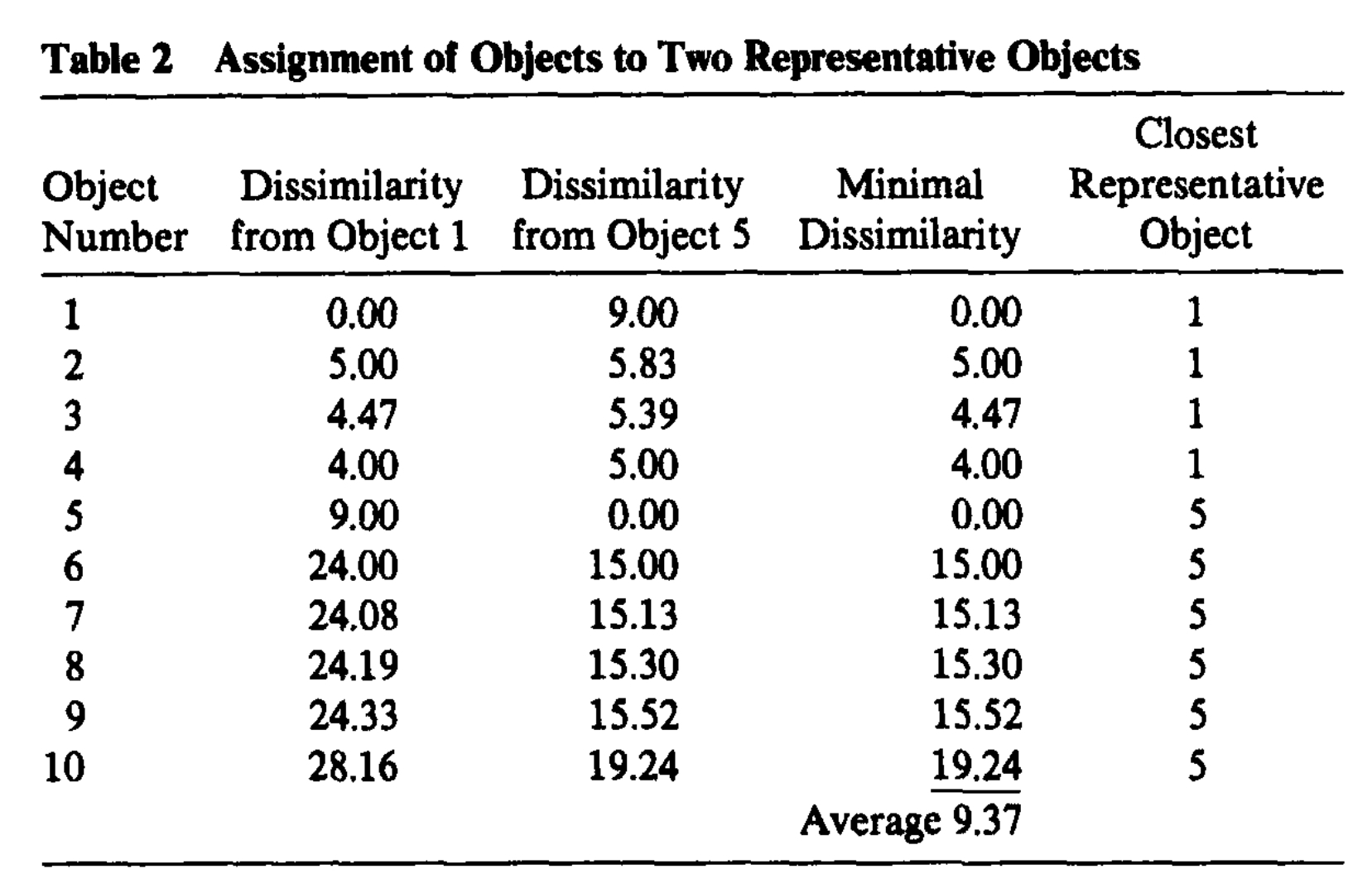
# Partitioning Around Medoids (PAM, k-medoids))

* The algorithm used in the program PAM search for representative objects among the objects of the data set.
* These objects should represent various aspects of the structure of the data.
* In the PAM algorithm the representative objects are the so-called medoids of the clusters.
* After finding a set of representative objects, the clusters are constructed by assigning each object of the data set to the nearest representative object.
* To illustrate the PAM algorithm, consider the data set containing 10 objects () and two variables ().
* Suppose the data set must be divided into two subsets or clusters ().
* The algorithm first considers possible choices of two representative objects and then constructs the clusters around these representative objects.

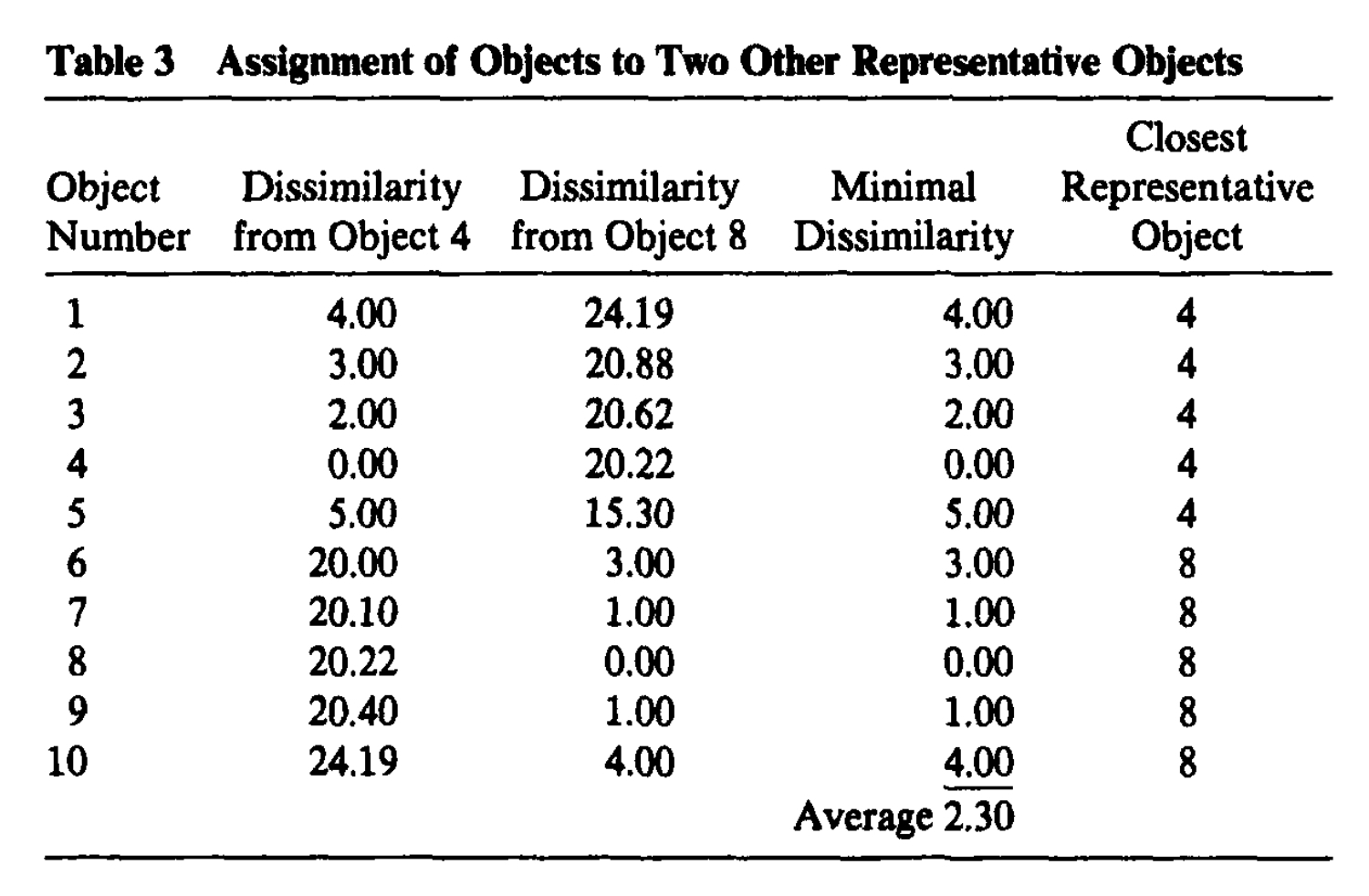


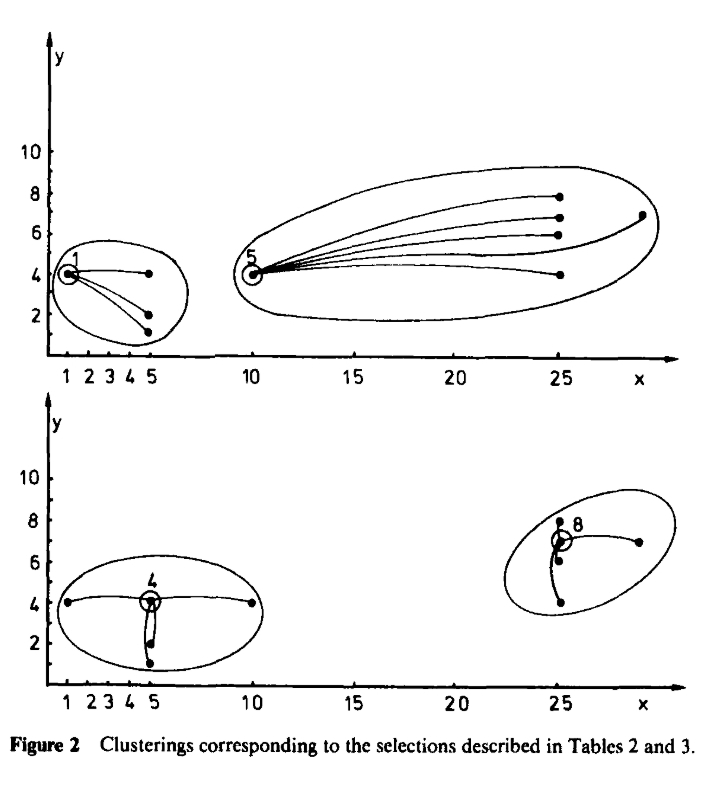


* As an example, suppose objects 1 and 5 are the selected representative objects.
* In Table 2 the dissimilarities (Euclidean distance) from each of the objects to the two selected objects are given, as well as the smallest of these two dissimilarities and the corresponding representative object.
* The average dissimilarity is 9.37.



* In Table 3 the assignment is carried out for the case objects 4 and 8 are selected as representative objects.





* The average dissimilarity for the case objects 4 and 8 are selected is 2.30, which is considerably less than the value of 9.37, found when 1 and 5 were the representative objects.
* Alternatively a PAM program can be used by entering a matrix of dissimilarities between objects.
* The algorithms are rather sophisticated and are not detailed here.