

# Cluster Analysis

## Contents

<b>Definition of a distance</b>	<b>3</b>
<b>Exercice 1</b>	<b>3</b>
<b>Euclidean distance</b>	<b>3</b>
<b>Exercice 2</b>	<b>3</b>
<b>Manhattan distance</b>	<b>3</b>
<b>Canberra distance</b>	<b>5</b>
<b>Exercice 3</b>	<b>5</b>
<b>Minkowski distance</b>	<b>5</b>
<b>Exercice 4</b>	<b>6</b>
<b>Chebyshev distance</b>	<b>7</b>
<b>Minkowski inequality</b>	<b>7</b>
<b>Exercice 5</b>	<b>7</b>
<b>Hölder inequality</b>	<b>7</b>
<b>Pearson correlation distance</b>	<b>8</b>
<b>Cosine correlation distance</b>	<b>9</b>
<b>Spearman correlation distance</b>	<b>9</b>
<b>Exercice 6</b>	<b>10</b>
<b>Kendall tau distance</b>	<b>10</b>
<b>Exercice 7</b>	<b>11</b>
<b>Standardization</b>	<b>11</b>
<b>Exercice 8</b>	<b>12</b>
<b>Similarity measures for binary data</b>	<b>13</b>
<b>Exercice 9</b>	<b>16</b>
<b>Exercice 10</b>	<b>19</b>

Nominal variables	19
Gower's dissimilarity	21
Daisy function	24
More on distance matrix computation	25
Visualizing distance matrices	28
Partitioning Clustering	29
K-Means Clustering	29
Exercice 11	30
K-Means	31
K-Means Algorithm	32
Exercice 12	33
<b>K-MEANS APPLIED TO FOOD NUTRIENT DATA</b>	33
Exercice 13	33
More on kmeans() output	36
Exercice 14	37
Exercice 15	37
Exercice 16	38
Visualizing k-means results	38
More on PCA	39
Silhouette	40
Silhouette example	41
Silhouette graph	42
Exercice 17	42
Exercice 18	43
Exercice 19	43
Partitioning Around Medoids (PAM, k-medoids)	43
<i>L</i> and <i>L*</i> clusters	48
• Required packages	
knitr:::opts_chunk\$set(echo = TRUE)	
#install.packages("dplyr", "ade4", "magrittr", "cluster",	
#"factoextra", "cluster.datasets", "xtable", "kableExtra",	
#"knitr", "summarytools")	

## Definition of a distance

- A distance function or a metric on  $\mathbb{R}^m$ ,  $m \geq 1$ , is a function  $d : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ .
- A distance function must satisfy some required properties or axioms.
- There are three main axioms.
- A1.  $d(\mathbf{x}, \mathbf{y}) = 0 \iff \mathbf{x} = \mathbf{y}$  (identity of indiscernibles);
- A2.  $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$  (symmetry);
- A3.  $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$  (triangle inequality), where  $\mathbf{x} = (x_1, \dots, x_m)$ ,  $\mathbf{y} = (y_1, \dots, y_m)$  and  $\mathbf{z} = (z_1, \dots, z_m)$  are all vectors of  $\mathbb{R}^m$ .
- 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:

$$d(\mathbf{x}, \mathbf{y}) \geq 0.$$

## Euclidean distance

- It is defined by:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{j=1}^m (x_j - y_j)^2}.$$

- A1-A2 are obvious.
- The proof of A3 is provided below.

## Exercice 2

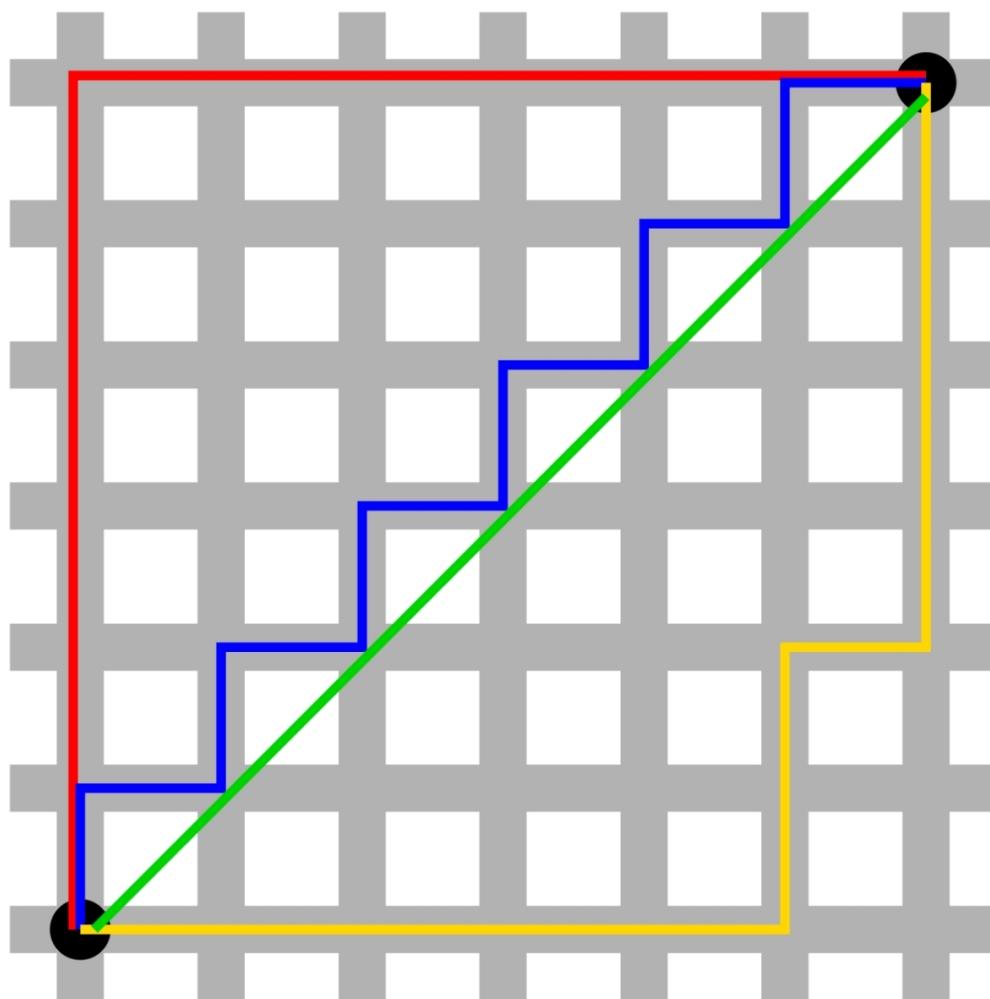
- Is the squared Euclidian distance a true distance?

## Manhattan distance

- The Manhattan distance also called taxi-cab metric or city-block metric is defined by:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^m |x_j - y_j|.$$

- A1-A2 hold.
- A3 also holds using the fact that  $|a + b| \leq |a| + |b|$  for any reals  $a, b$ .
- 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 0.0000000 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:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^m \frac{|x_j - y_j|}{|x_j| + |y_j|}.$$

- Note that the term  $|x_j - y_j|/(|x_j| + |y_j|)$  is not properly defined as:  $x_j = y_j = 0$ .
- 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 Manhattan distances are special cases of the Minkowski distance which is defined, for  $p \geq 1$ , by:

$$d(\mathbf{x}, \mathbf{y}) = \left[ \sum_{j=1}^m |x_j - y_j|^p \right]^{1/p}.$$

- For  $p = 1$ , we get the Manhattan distance.
- For  $p = 2$ , we get the Euclidian distance.
- Let us also define:

$$\|\mathbf{x}\|_p \equiv \left[ \sum_{j=1}^m |x_j|^p \right]^{1/p},$$

- where  $\|\cdot\|_p$  is known as the  $p$ -norm or Minkowski norm.
- Note that the Minkowski distance and norm are related by:

$$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_p.$$

- Conversely, we have:

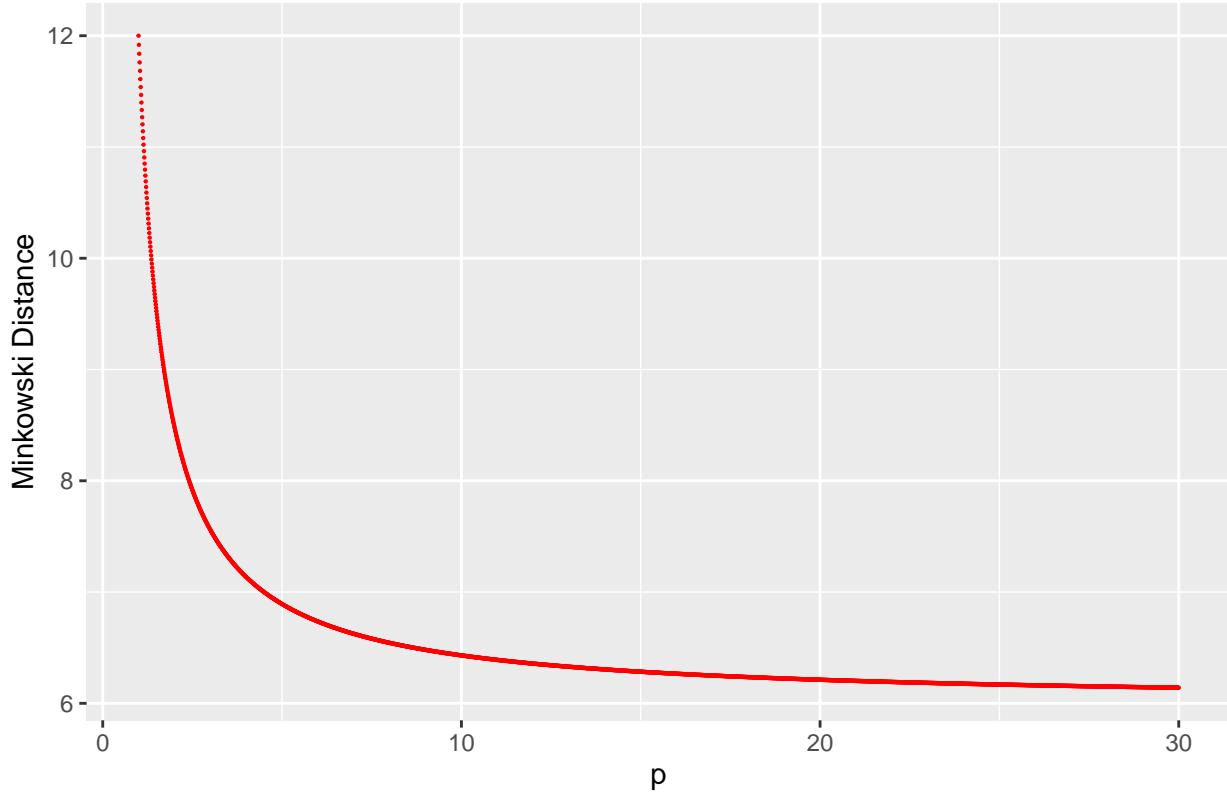
$$\|\mathbf{x}\|_p = d(\mathbf{x}, \mathbf{0}),$$

where  $\mathbf{0}$  is the null-vetor of  $\mathbb{R}^m$ .

```
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")
```

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:

$$d(\mathbf{x}, \mathbf{y}) = \max_{j=1, \dots, n} (|x_j - y_j|) = \lim_{p \rightarrow \infty} \left[ \sum_{j=1}^n |x_j - y_j|^p \right]^{1/p}.$$

- The corresponding norm is:

$$\|\mathbf{x}\|_\infty = \max_{j=1, \dots, n} (|x_j|).$$

## Minkowski inequality

- The proof of the triangular inequality A3 is based on the Minkowski inequality:
- For any nonnegative real numbers  $a_1, \dots, a_m; b_1, \dots, b_m$ , and for any  $p \geq 1$ , we have:

$$\left[ \sum_{j=1}^m (a_j + b_j)^p \right]^{1/p} \leq \left[ \sum_{j=1}^m a_j^p \right]^{1/p} + \left[ \sum_{j=1}^m b_j^p \right]^{1/p}.$$

- To prove that the Minkowski distance satisfies A3, notice that

$$\sum_{j=1}^m |x_j - z_j|^p = \sum_{j=1}^m |(x_j - y_j) + (y_j - z_j)|^p.$$

- Since for any reals  $x, y$ , we have:  $|x + y| \leq |x| + |y|$ , and using the fact that  $x^p$  is increasing in  $x \geq 0$ , we obtain:

$$\sum_{j=1}^m |x_j - z_j|^p \leq \sum_{j=1}^m (|x_j - y_j| + |y_j - z_j|)^p.$$

- Applying the Minkowski inequality with  $a_j = |x_j - y_j|$  and  $b_j = |y_j - z_j|$ ,  $j = 1, \dots, n$ , we get:

$$\sum_{j=1}^m |x_j - z_j|^p \leq \left( \sum_{j=1}^m |x_j - y_j|^p \right)^{1/p} + \left( \sum_{j=1}^m |y_j - z_j|^p \right)^{1/p}.$$

## Exercice 5

To illustrate the Minkowski inequality, draw 100 times two lists of 100 draws from the lognormal distribution with mean 1600 and standard-deviation 300. 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  $a_1, \dots, a_m; b_1, \dots, b_m$ , and any  $p, q > 1$  with  $1/p + 1/q = 1$ , we have:

$$\sum_{j=1}^m a_j b_j \leq \left[ \sum_{j=1}^m a_j^p \right]^{1/p} \left[ \sum_{j=1}^m b_j^q \right]^{1/q}$$

- The proof of the Hölder inequality relies on the Young inequality:
- For any  $a, b > 0$ , we have

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q},$$

with equality occurring iff:  $a^p = b^q$ .

- To prove the Young inequality, one can use the (strict) convexity of the exponential function.
- For any reals  $x, y$ , we have:

$$e^{\frac{x}{p} + \frac{y}{q}} \leq \frac{e^x}{p} + \frac{e^y}{q}.$$

- We then set:  $x = p \ln a$  and  $y = q \ln b$  to get the Young inequality.
- A good reference on inequalities is: Z. Cvetkovski, Inequalities: theorems, techniques and selected problems, 2012, Springer Science & Business Media.

## # Cauchy-Schwartz inequality

- Note that the triangular inequality for the Minkowski distance implies:

$$\sum_{j=1}^m |x_j| \leq \left[ \sum_{j=1}^m |x_j|^p \right]^{1/p}.$$

- Note that for  $p = 2$ , we have  $q = 2$ . The Hölder inequality implies for that special case

$$\sum_{j=1}^m |x_j y_j| \leq \sqrt{\sum_{j=1}^m x_j^2} \sqrt{\sum_{j=1}^m y_j^2}.$$

- Since the LHS of the above inequality is greater than  $|\sum_{j=1}^m x_j y_j|$ , we get the Cauchy-Schwartz inequality

$$|\sum_{j=1}^m x_j y_j| \leq \sqrt{\sum_{j=1}^m x_j^2} \sqrt{\sum_{j=1}^m y_j^2}.$$

- Using the dot product notation called also scalar product notation:  $\mathbf{x} \cdot \mathbf{y} = \sum_{j=1}^m x_j y_j$ , and the norm notation  $\|\cdot\|_2$ , the Cauchy-Schwartz inequality is:

$$|\mathbf{x} \cdot \mathbf{y}| \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2.$$

## Pearson correlation distance

- The Pearson correlation coefficient is a similarity measure on  $\mathbb{R}^m$  defined by:

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{\sum_{j=1}^m (x_j - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_{j=1}^m (x_j - \bar{x})^2 \sum_{j=1}^m (y_j - \bar{y})^2}},$$

where  $\bar{x}$  is the mean of the vector  $\mathbf{x}$  defined by:

$$\bar{x} = \frac{1}{n} \sum_{j=1}^m x_j,$$

- Note that the Pearson correlation coefficient satisfies P2 and is invariant to any positive linear transformation, i.e.:
$$\rho(\alpha \mathbf{x}, \mathbf{y}) = \rho(\mathbf{x}, \mathbf{y}),$$
for any  $\alpha > 0$ .

- The Pearson distance (or correlation distance) is defined by:

$$d(\mathbf{x}, \mathbf{y}) = 1 - \rho(\mathbf{x}, \mathbf{y}).$$

- Note that the Pearson distance does not satisfy A1 since  $d(\mathbf{x}, \mathbf{x}) = 0$  for any non-zero vector  $\mathbf{x}$ . It neither satisfies the triangle inequality. However, the symmetry property is fulfilled.

## Cosine correlation distance

- The cosine of the angle  $\theta$  between two vectors  $\mathbf{x}$  and  $\mathbf{y}$  is a measure of similarity given by:

$$\cos(\theta) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|_2 \|\mathbf{y}\|_2} = \frac{\sum_{j=1}^m x_j y_j}{\sqrt{\sum_{j=1}^m x_j^2 \sum_{j=1}^m y_j^2}}.$$

- Note that the cosine of the angle between the two centred vectors  $\mathbf{x} - \bar{\mathbf{x}}\mathbf{1}$  and  $\mathbf{y} - \bar{\mathbf{y}}\mathbf{1}$  coincides with the Pearson correlation coefficient of  $\mathbf{x}$  and  $\mathbf{y}$ , where  $\mathbf{1}$  is a vector of units of  $\mathbb{R}^m$ .
- The cosine correlation distance is defined by:

$$d(\mathbf{x}, \mathbf{y}) = 1 - \cos(\theta).$$

- 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 separately each of the vectors to ranked data values:

$$\mathbf{x} \rightarrow \text{rank}(\mathbf{x}) = (x_1^r, \dots, x_m^r).$$

- Here,  $x_j^r$  is the rank of  $x_j$  among the set of values of  $\mathbf{x}$ .
- We illustrate this transformation with a simple example:
- If  $\mathbf{x} = (3, 1, 4, 15, 92)$ , then the rank-order vector is  $\text{rank}(\mathbf{x}) = (2, 1, 3, 4, 5)$ .

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

```
## [1] 2 1 3 4 5
```

- The Spearman's rank correlation of two numerical vectors  $\mathbf{x}$  and  $\mathbf{y}$  is simply the Pearson correlation of the two corresponding rank-order vectors  $\text{rank}(\mathbf{x})$  and  $\text{rank}(\mathbf{y})$ , i.e.  $\rho(\text{rank}(\mathbf{x}), \text{rank}(\mathbf{y}))$ . This measure is useful because it is more robust against outliers than the Pearson correlation.
- If all the  $n$  ranks are distinct, it can be computed using the following formula:

$$\rho(\text{rank}(\mathbf{x}), \text{rank}(\mathbf{y})) = 1 - \frac{6 \sum_{j=1}^m d_j^2}{n(n^2 - 1)},$$

where  $d_j = x_j^r - y_j^r$ ,  $j = 1, \dots, n$ .

- The spearman distance is then defined by:

$$d(\mathbf{x}, \mathbf{y}) = 1 - \rho(\text{rank}(\mathbf{x}), \text{rank}(\mathbf{y})).$$

- It can be shown that easaly that it is not a proper distance.
- If all the  $n$  ranks are distinct, we get:

$$d(\mathbf{x}, \mathbf{y}) = \frac{6 \sum_{j=1}^m d_j^2}{n(n^2 - 1)}.$$

```
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  $\mathbf{x} = (22, 34, 1, 12, 25, 56, 7)$  and  $\mathbf{y} = (2, 64, 12, 2, 22, 5, 8)$  :
- 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  $\mathbf{x}$  and the rankings of  $\mathbf{y}$ .
- The number of pairs of observations among  $n$  observations or values is:

$$\binom{n}{2} = \frac{n(n-1)}{2}.$$

- The pairs of observations  $(x_i, x_j)$  and  $(y_i, y_j)$  are said to be *concordant* if:

$$\text{sign}(x_j - x_i) = \text{sign}(y_j - y_i),$$

and to be *discordant* if:

$$\text{sign}(x_j - x_i) = -\text{sign}(y_j - y_i),$$

where  $\text{sign}(\cdot)$  returns 1 for positive numbers and  $-1$  negative numbers and 0 otherwise.

- If  $x_j = x_j$  or  $y_j = y_j$  (or both), there is a tie.
- The Kendall  $\tau$  coefficient is defined by (neglecting ties):

$$\tau = \frac{1}{n(n-1)} \sum_{j=1}^n \sum_{j=1}^m \text{sign}(x_j - x_j) \text{sign}(y_j - y_j).$$

- Let  $n_c$  (resp.  $n_d$ ) be the number of concordant (resp. discordant) pairs, we have

$$\tau = \frac{2(n_c - n_d)}{n(n-1)}.$$

- The Kendall tau distance is then:

$$d(\mathbf{x}, \mathbf{y}) = 1 - \tau.$$

- 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  $\mathbf{x} = (22, 34, 1, 12, 25, 56, 7)$  and  $\mathbf{y} = (2, 64, 12, 2, 22, 5, 8)$ :
- 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 unitless variables.
- A well known method is the z-score transformation.
- Let  $\mathbf{v} \equiv (v_1, \dots, v_n)$  a vector of measurements recorded for  $n$  individuals or objects.

$$\mathbf{v} \rightarrow \left( \frac{v_1 - \bar{\mathbf{v}}}{s_{\mathbf{v}}}, \dots, \frac{v_n - \bar{\mathbf{v}}}{s_{\mathbf{v}}} \right),$$

where  $\bar{\mathbf{v}}, s_{\mathbf{v}}$  are the sample mean and standard-deviation, respectively, given by:

$$\bar{\mathbf{v}} = \frac{1}{n} \sum_{i=1}^n v_i, \quad s_{\mathbf{v}} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (v_i - \bar{\mathbf{v}})^2}.$$

- The transformed variable will have a mean of 0 and a variance of 1.
- 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

```

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

**Table 3 Age (in years) and Height (in centimeters) of Four People**

Person	Age (yr)	Height (cm)
A	35	190
B	40	190
C	35	160
D	40	160

Figure 1: From Kaufman & Rousseeuw, p. 6-8

- 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  $n$  characters.
- \*The presence of the character is coded by 1 and the absence by 0.
- We have have at our disposal two vectors.
- $\mathbf{x}$  is observed for a first individual (or object).
- $\mathbf{y}$  is observed for a second individual.
- We can then calculate the following four statistics:

$$a = \mathbf{x} \cdot \mathbf{y} = \sum_{j=1}^m x_j y_j.$$

$$b = \mathbf{x} \cdot (\mathbf{1} - \mathbf{y}) = \sum_{j=1}^m x_j (1 - y_j).$$

$$c = (\mathbf{1} - \mathbf{x}) \cdot \mathbf{y} = \sum_{j=1}^m (1 - x_j) y_j.$$

$$d = (\mathbf{1} - \mathbf{x}) \cdot (\mathbf{1} - \mathbf{y}) = \sum_{j=1}^m (1 - x_j) (1 - y_j).$$

- The counts of matches are  $a$  for (1, 1) and  $d$  for (0, 0);
- The counts of mismatches are  $b$  for (1, 0) and  $c$  for (0, 1).
- Note that obviously:  $a + b + c + d = n$ .
- This gives a very useful  $2 \times 2$  association table.

		Second individual		
		1	0	Totals
First individual	1	$a$	$b$	$a + b$
	0	$c$	$d$	$c + d$
Totals		$a + c$	$b + d$	$n$

- The data shows 8 people (individuals) and 10 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,
```

**Table 9 Binary Variables for Eight People**

Person	Sex (Male = 1, Female = 0)	Married (Yes = 1, No = 0)	Fair Hair = 1, Dark Hair = 0	Blue Eyes = 1, Brown Eyes = 0	Wears Glasses (Yes = 1, No = 0)	Round Face = 1, Oval Face = 0	Pessimist = 1, Optimist = 0	Evening Type = 1, Morning Type = 0	Is an Only Child (Yes = 1, No = 0)	Left-Handed = 1, Right-Handed = 0
Ilan	1	0	1	1	0	0	1	0	0	0
Jacqueline	0	1	0	0	1	0	0	0	0	0
Kim	0	0	1	0	0	0	1	0	0	1
Lieve	0	1	0	0	0	0	0	1	1	0
Leon	1	1	0	0	1	1	0	1	1	0
Peter	1	1	0	0	1	0	1	1	0	0
Talia	0	0	0	1	0	1	0	0	0	0
Tina	0	0	0	1	0	1	0	0	0	0

Figure 2: Table from Kaufman, L., & Rousseeuw, P. J. (2009). Finding groups in data: an introduction to cluster analysis (Vol. 344). John Wiley & Sons

```

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", "Morning Type", "Is an Only Child", "Left-Handed", "Right-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:  $a = 1$ ,  $b = 3$ ,  $c = 1$ ,  $d = 5$ .
- Note that interchanging Ilan and Talia would permute  $b$  and  $c$  while leaving  $a$  and  $d$  unchanged.
- A good similarity or dissimilarity coefficient must treat  $b$  and  $c$  symmetrically.
- A similarity measure is denoted by:  $s(\mathbf{x}, \mathbf{y})$ .
- The corresponding distance is then defined as:

$$d(\mathbf{x}, \mathbf{y}) = 1 - s(\mathbf{x}, \mathbf{y}).$$

- Alternatively, we have:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{1 - s(\mathbf{x}, \mathbf{y})}.$$

- A list of some of the similarity measures  $s(\mathbf{x}, \mathbf{y})$  that have been suggested for binary data is shown below.
- A 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.

Coefficient	$s(\mathbf{x}, \mathbf{y})$	$d(\mathbf{x}, \mathbf{y}) = 1 - s(\mathbf{x}, \mathbf{y})$
Simple matching	$\frac{a+d}{a+b+c+d}$	$\frac{b+c}{a+b+c+d}$
Jaccard	$\frac{a}{a+b+c}$	$\frac{b+c}{a+b+c}$
Rogers and Tanimoto (1960)	$\frac{a+d}{a+2(b+c)+d}$	$\frac{2(b+c)}{a+2(b+c)+d}$
Gower and Legendre (1986)	$\frac{2(a+d)}{2(a+d)+b+c}$	$\frac{b+c}{2(a+d)+b+c}$
Gower and Legendre (1986)	$\frac{2a}{2a+b+c}$	$\frac{b+c}{2a+b+c}$

- To calculate these coefficients, we use the function: `dist.binary()`. available in the **ade4** package.
- All the distances in the **ade4** package are of type  $d(\mathbf{x}, \mathbf{y}) = \sqrt{1 - s(\mathbf{x}, \mathbf{y})}$ .

```
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  $d$ .
- The measures embedding  $d$  are sometimes called symmetrical.
- The other measures 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  $d$  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.
- 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.

After the identification of missing measurements, a procedure is carried out for estimating their values. In this procedure each variable containing missing values is considered in turn. Each time the algorithm looks for the most similar complete variable and then uses the latter for filling in the missing values. In our example END has two missing values. The similarities between this variable and the complete variables are given in Figure 7.

The variable WAR has the highest similarity with END and is therefore the most appropriate for estimating the missing values of END. The two

<u>variable</u>	<u>END</u>	<u>similarity</u>								
WAR	<table border="1"> <tr> <td>1</td><td>0</td></tr> <tr> <td>1</td><td>5</td><td>4</td></tr> <tr> <td>0</td><td>1</td><td>8</td></tr> </table>	1	0	1	5	4	0	1	8	36
1	0									
1	5	4								
0	1	8								
FLY	<table border="1"> <tr> <td>1</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>3</td></tr> <tr> <td>0</td><td>5</td><td>9</td></tr> </table>	1	0	1	1	3	0	5	9	6
1	0									
1	1	3								
0	5	9								
VER	<table border="1"> <tr> <td>1</td><td>0</td></tr> <tr> <td>1</td><td>6</td><td>7</td></tr> <tr> <td>0</td><td>0</td><td>5</td></tr> </table>	1	0	1	6	7	0	0	5	30
1	0									
1	6	7								
0	0	5								
HAI	<table border="1"> <tr> <td>1</td><td>0</td></tr> <tr> <td>1</td><td>2</td><td>5</td></tr> <tr> <td>0</td><td>4</td><td>7</td></tr> </table>	1	0	1	2	5	0	4	7	6
1	0									
1	2	5								
0	4	7								

**Figure 7** Similarities between a variable with missing values (END) and all variables without missing values, in the animal data set.

**REVISED DATA**

\*\*\*\*\*

	W	F	V	E	G	H
	A	L	E	N	R	A
	R	Y	R	D	O	I

ant	0	0	0	0	1	0
bee	0	1	0	0	1	1
cat	1	0	1	0	0	1
cpl	0	0	0	0	0	1
chi	1	0	1	1	1	1
cow	1	0	1	0	1	1
duc	1	1	1	0	1	0
eag	1	1	1	1	0	0
ele	1	0	1	1	1	0
fly	0	1	0	0	0	0
fro	0	0	1	1	0	0
her	0	0	1	0	1	0
lio	1	0	1	1	1	1
liz	0	0	1	0	0	0
lob	0	0	0	0	0	0
man	1	0	1	1	1	1
rab	1	0	1	0	1	1
sal	0	0	1	0	0	0
spi	0	0	0	0	0	1
wha	1	0	1	1	1	0

## 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.

## Nominal variables

- We previously studied above binary variables which can only take on two states coded 0, 1.
- 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.
- Let  $M$  be the number of states and code the outcomes as  $1, \dots, M$ .
- We may choose  $1 = \text{blue}$ ,  $2 = \text{brown}$ ,  $3 = \text{green}$ , and  $4 = \text{grey}$ .
- These states are not ordered in any way
- One strategy would be creating a new binary variable for each of the  $M$  nominal states.
- Then to put it equal to 1 if the corresponding state occurs and to 0 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  $\mathbf{x}, \mathbf{y}$ , are both  $m$  nominal records for two individuals,
- Let define the function:

$$\delta(x_j, y_j) \equiv \begin{cases} 0, & \text{if } x_j = y_j; \\ 1, & \text{if } x_j \neq y_j. \end{cases}$$

- Let  $N_{a+d}$  be the number of attributes of the two individuals on which the two records match:

$$N_{a+d} = \sum_{j=1}^m [1 - \delta(x_j, y_j)].$$

- Let  $N_{b+c}$  be the number of attributes on which the two records do not match:

$$N_{b+c} = \sum_{j=1}^m \delta(x_j, y_j).$$

- Let  $N_d$  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:

$$d(\mathbf{x}, \mathbf{y}) = \frac{\sum_{j=1}^m \delta(x_j, y_j)}{n} = \frac{N_{a+d}}{N_{a+d} + N_{b+c}}.$$

- Note that simple matching has exactly the same meaning as in the preceding section.
- For more details, see GAN, Guojun, MA, Chaoqun, et WU, Jianhong. Data clustering: theory, algorithms, and applications. Society for Industrial and Applied Mathematics, 2020.

**Table 6.2.** Some matching coefficients for nominal data.

Measure	$s(\mathbf{x}, \mathbf{y})$	Weighting of matches, mismatches
Russell and Rao	$\frac{N_{a+d} - N_d}{N_{a+d} + N_{b+c}}$	Equal weights
Simple matching	$\frac{N_{a+d}}{N_{a+d} + N_{b+c}}$	Equal weights
Jaccard	$\frac{N_{a+d} - N_d}{N_{a+d} - N_d + N_{b+c}}$	Equal weights
Unnamed	$\frac{2N_{a+d}}{2N_{a+d} + N_{b+c}}$	Double weight for matched pairs
Dice	$\frac{2N_{a+d} - 2N_d}{2N_{a+d} - 2N_d + N_{b+c}}$	Double weight for matched pairs
Rogers-Tanimoto	$\frac{N_{a+d}}{N_{a+d} + 2N_{b+c}}$	Double weight for unmatched pairs
Unnamed	$\frac{N_{a+d} - N_d}{N_{a+d} - N_d + 2N_{b+c}}$	Double weight for unmatched pairs
Kulczynski	$\frac{N_{a+d} - N_d}{N_{b+c}}$	Matched pairs excluded from denominator
Unnamed	$\frac{N_{a+d}}{N_{b+c}}$	Matched pairs excluded from denominator

Figure 3: From: GAN et al. (2007)

## 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:

$$d(\mathbf{x}, \mathbf{y}) = \frac{\sum_{j=1}^m w_j \delta(x_j, y_j)}{\sum_{j=1}^m w_j}.$$

- The weight  $w_j$  is put equal to 1 when both measurements  $x_j$  and  $y_j$  are nonmissing,
- The number  $\delta(x_j, y_j)$  is the contribution of the  $j$ th measure or variable to the dissimilarity measure.
- If the  $j$ th measure is nominal, we take

$$\delta(x_j, y_j) \equiv \begin{cases} 0, & \text{if } x_j = y_j; \\ 1, & \text{if } x_j \neq y_j. \end{cases}$$

- If the  $j$ th measure is interval-scaled, we take instead:

$$\delta(x_j, y_j) \equiv \frac{|x_j - y_j|}{R_j},$$

where  $R_j$  is the range of variable  $j$  over the available data.

- Consider the following data set:

object	variable							
	1	2	3	4	5	6	7	8
Begonia	0	1	1	4	3	15	25	15
Broom	1	0	0	2	1	3	150	50
Camellia	0	1	0	3	3	1	150	50
Dahlia	0	0	1	4	2	16	125	50
Forget-me-not	0	1	0	5	2	2	20	15
Fuchsia	0	1	0	4	3	12	50	40
Geranium	0	0	0	4	3	13	40	20
Gladiolus	0	0	1	2	2	7	100	15
Heather	1	1	0	3	1	4	25	15
Hydrangea	1	1	0	5	2	14	100	60
Iris	1	1	1	5	3	8	45	10
Lily	1	1	1	1	2	9	90	25
Lily-of-the-valley	1	1	0	1	2	6	20	10
Peony	1	1	1	4	2	11	80	30
Pink Carnation	1	0	0	3	2	10	40	20
Red Rose	1	0	0	4	2	18	200	60
Scotch Rose	1	0	0	2	2	17	150	60
Tulip	0	0	1	2	1	5	25	10

Table 1: Flower dataset.

From: Struyf, A., Hubert, M., & Rousseeuw, P. (1997). Clustering in an object-oriented environment. *Journal of Statistical Software*, 1(4), 1-30.

- 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 (Tuberclule): 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:

```
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")
```



Figure 4: Begonia



Figure 5: Genêt

```

        )
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

```

$$\frac{|1 - 0| + |0 - 1| + |0 - 1| + 1 + |1 - 3|/2 + |3 - 15|/17 + |150 - 25|/180 + |50 - 15|/50}{8} \approx 0.8875408$$

## Daisy function

---

**daisy** *Dissimilarity Matrix Calculation*

---

### Description

Compute all the pairwise dissimilarities (distances) between observations in the data set. The original variables may be of mixed types. In that case, or whenever `metric = "gower"` is set, a generalization of Gower's formula is used, see 'Details' below.

### Usage

```
daisy(x, metric = c("euclidean", "manhattan", "gower"),
      stand = FALSE, type = list(), weights = rep.int(1, p),
      warnBin = warnType, warnAsym = warnType, warnConst = warnType,
      warnType = TRUE)
```

Cluster package description available at this link.

```

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

### USArrests

From datasets v3.6.2  
by R-core R-core@R-project.org 99.99th Percentile

#### Violent Crime Rates By US State

This data set contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas.

**Keywords** [datasets](#)

#### Usage

USArrests

#### Note

`usArrests` contains the data as in McNeil's monograph. For the `UrbanPop` percentages, a review of the table (No. 21) in the Statistical Abstracts 1975 reveals a transcription error for Maryland (and that McNeil used the same "round to even" rule that R's `round()` uses), as found by Daniel S Coven (Arizona).

See the example below on how to correct the error and improve accuracy for the '<n>.5' percentages.

#### Format

A data frame with 50 observations on 4 variables.

[1]	Murder	numeric	Murder arrests (per 100,000)
[2]	Assault	numeric	Assault arrests (per 100,000)
[3]	UrbanPop	numeric	Percent urban population

#### References

McNeil, D. R. (1977) *Interactive Data Analysis*. New York: Wiley.

- 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()`.

```

knitr::kable(USArrests,caption = "USArrest data set",
digits = 1,align=c("c","c","c","c"),booktabs = TRUE)%>%
kable_styling(latex_options = "HOLD_position")

```

Table 4: USArrest data set

	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
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	24.0
Indiana	7.2	113	65	21.0
Iowa	2.2	56	57	11.3
Kansas	6.0	115	66	18.0
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	9.0	178	70	28.2
Montana	6.0	109	53	16.4
Nebraska	4.3	102	62	16.5
Nevada	12.2	252	81	46.0
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	13.0	337	45	16.1
North Dakota	0.8	45	44	7.3
Ohio	7.3	120	75	21.4
Oklahoma	6.6	151	68	20.0
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	4.0	145	73	26.2
West Virginia	5.7	81	39	9.3
Wisconsin	2.6	26	53	10.8
Wyoming	6.8	161	60	15.6

```

##  kable_classic(latex_options = "hold_position")

set.seed(123)
ss <- sample(1:50, 15)
df <- USArrests[ss, ]
df.scaled <- scale(df)
knitr::kable(df.scaled, caption = "Sample of USArrest data set", digits = 4, align=c("c", "c", "c"), booktabs
kable_classic(latex_options = "hold_position")

```

Table 5: Sample of USArrest data set

	Murder	Assault	UrbanPop	Rape
New Mexico	0.5851	1.0230	0.2251	0.6110
Iowa	-1.7022	-1.5476	-0.6892	-1.4389
Indiana	-0.4591	-0.9078	-0.1266	-0.4829
Arizona	-0.2354	1.1240	0.9284	0.5026
Tennessee	1.0326	-0.0659	-0.5486	0.0986
Texas	0.9083	0.0801	0.9284	-0.0394
Oregon	-1.0309	-0.3914	0.0141	0.3351
West Virginia	-0.8320	-1.2670	-1.9552	-1.6360
Missouri	-0.0116	-0.1781	0.2251	0.2267
Montana	-0.7575	-0.9527	-0.9706	-0.9362
Nebraska	-1.1801	-1.0312	-0.3376	-0.9264
California	-0.0116	0.9220	1.7020	1.4487
South Carolina	1.3309	0.9557	-1.3222	-0.3351
Nevada	0.7840	0.6526	0.9987	1.9809
Florida	1.5796	1.5843	0.9284	0.5913

- 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=2)

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.

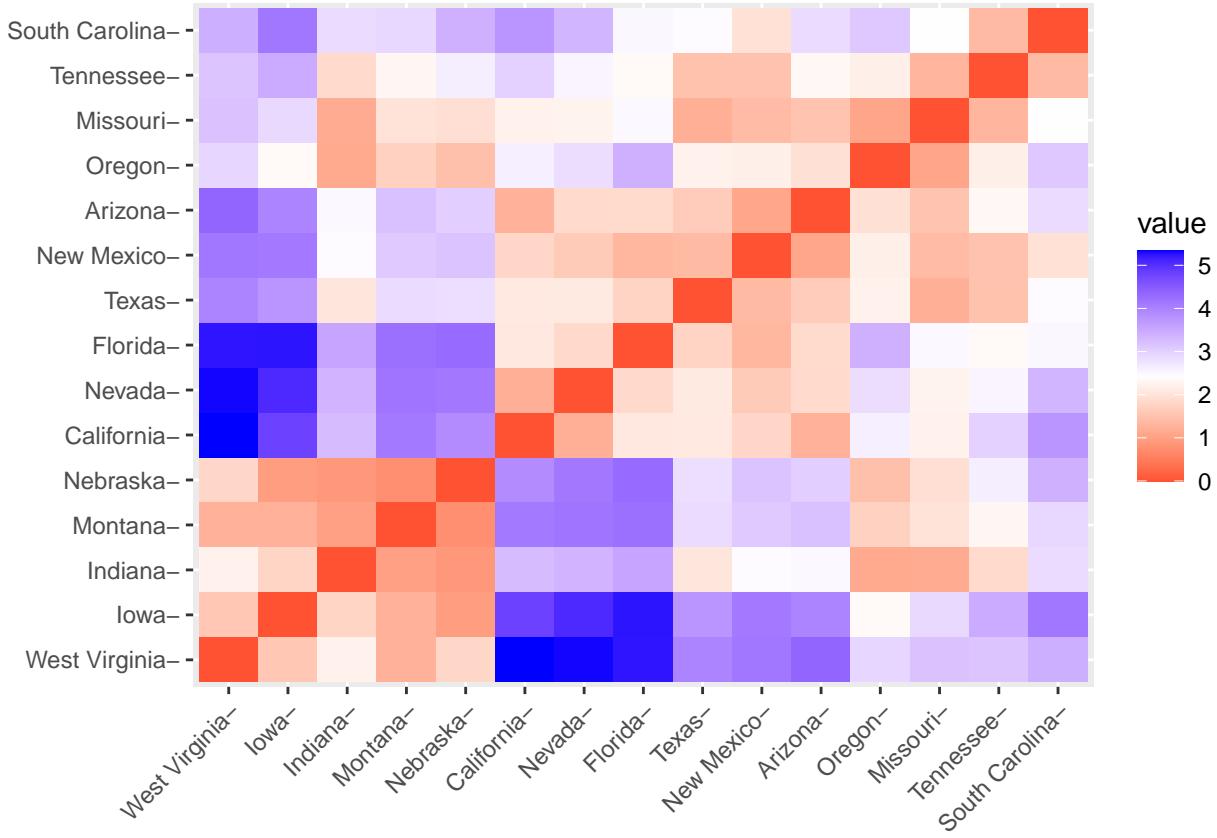
```
library("factoextra")  
dist.cor <- get_dist(df.scaled, method = "pearson")  
round(as.matrix(dist.cor)[1:3, 1:3], 1)
```

```
##           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:
  - 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.
  - 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.
  - 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.

**Table 4.1 Nutrients in Meat, Fish, and Fowl**  
*[The Yearbook of Agriculture 1959 (The United States Department of Agriculture, Washington, D.C.) p. 244.]* The quantity used is always 3 ounces.

	Food Energy (Calories)	Protein (Grams)	Fat (Grams)	Calcium (Milli Grams)	Iron (Milli Grams)
BB Beef, braised	340	20	28	9	2.6
HR Hamburger	245	21	17	9	2.7
BR Beef, roast	420	15	39	7	2.0
BS Beef, steak	375	19	32	9	2.6
BC Beef, canned	180	22	10	17	3.7
CB Chicken, broiled	115	20	3	8	1.4
CC Chicken, canned	170	25	7	12	1.5
BH Beef heart	160	26	5	14	5.9
LL Lamb leg, roast	265	20	20	9	2.6
LS Lamb shoulder, roast	300	18	25	9	2.3
HS Smoked ham	340	20	28	9	2.5
PR Pork roast	340	19	29	9	2.5
PS Pork simmered	355	19	30	9	2.4
BT Beef tongue	205	18	14	7	2.5
VC Veal cutlet	185	23	9	9	2.7
FB Bluefish, baked	135	22	4	25	.6
AR Clams, raw	70	11	1	82	6.0
AC Clams, canned	45	7	1	74	5.4
TC Crabmeat, canned	90	14	2	38	.8
HF Haddock, fried	135	16	5	15	.5
MB Mackerel, broiled	200	19	13	5	1.0
MC Mackerel, canned	155	16	9	157	1.8
PF Perch, fried	195	16	11	14	1.3
SC Salmon, canned	120	17	5	159	0.7
DC Sardines, canned	180	22	9	367	2.5
UC Tuna, canned	170	25	7	7	1.2
RC Shrimp, canned	110	23	1	98	2.6

- 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 converts the variables (with the exception of Fat) in percentage of food delivery.
- For e.g., the first (BB) line is obtained in the following way:
- $340/3200 = 11\%$ (Food Energy).
- $20/70 = 29\%$ (Protein).
- $0.009/0.8 = 1\%$  (Calcium).
- $2.6/10 = 26\%$  (Iron).
- 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  $n$  objects and  $k$  clusters,  $k \leq n$ .
- Our purpose is to partition the  $n$  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.
- Example of functions in R.

```
library(multicool)
```

```
## Loading required package: Rcpp
```

**Table 4.2 Nutrients in Meat, Fish, and Fowl**  
 As a percentage of recommended daily allowances.

	Food Energy	Protein	Fat (Grams)	Calcium	Iron
Beef, braised	11	29	28	1	26
Hamburger	8	30	17	1	27
Beef, roast	13	21	39	1	20
Beef, steak	12	27	32	1	26
Beef, canned	6	31	10	2	37
Chicken, broiled	4	29	3	1	14
Chicken, canned	5	36	7	2	15
Beef, heart	5	37	5	2	59
Lamb leg, roast	8	29	20	1	26
Lamb shoulder, roast	9	26	25	1	25
Ham, smoked	11	29	28	1	25
Pork roast	11	27	29	1	25
Pork simmered	11	27	30	1	25
Beef tongue	6	26	14	1	25
Veal cutlet	6	33	9	1	27
Bluefish, baked	4	31	4	3	6
Clams, raw	2	16	1	10	60
Clams, canned	1	10	1	9	54
Crabmeat, canned	3	20	2	5	8
Haddock, fried	4	23	5	2	5
Mackerel, broiled	6	27	13	1	10
Mackerel, canned	5	23	9	20	18
Perch, fried	6	23	11	2	13
Salmon, canned	4	24	5	20	7
Sardines, canned	6	31	9	46	25
Tuna, canned	5	36	7	1	12
Shrimp, canned	3	33	1	12	26

Figure 6: Data

```
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  $e$ .
- 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  $e$  is minim.
- If the neighborhoods are very large, it is cheaper computationally to move to the first partition discovered in the neighborhood where  $e$  is reduced from its present value.
- A number of stopping rules are possible.
- For example, the search stops when  $e$  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 ( $n = 5$ ) beef foods  $\{BB, BR, BS, BC, BH\}$  into three clusters ( $k = 3$ ).
- 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  $\mathbf{x}_i \equiv (x_i^1, \dots, x_i^m)$  the vector of values for the object  $i$ ,  $i = 1, \dots, n$ .
- The variables are assumed scaled.
- The partition has  $k$  disjoint clusters:  $C_1, \dots, C_k$ , which are the indices of the objects in the various clusters.
- Let  $n_l$  be the number of objects in cluster  $C_l$ .
- Each of the  $n$  objects lies in just one of the  $k$  clusters.
- Note that  $\sum_{l=1}^k n_l = n$ .
- The vector of means over the objects in cluster  $C_l$  is denoted by  $\bar{\mathbf{x}}_l$ , with

$$\bar{\mathbf{x}}_l \equiv \frac{1}{n_l} \sum_{i \in C_l} \mathbf{x}_i = (\bar{x}_l^1, \dots, \bar{x}_l^m), \quad l = 1, \dots, k,$$

where

$$\bar{x}_l^j \equiv \frac{\sum_{i \in C_l} x_i^j}{n_l}, \quad j = 1, \dots, m; \quad l = 1, \dots, k.$$

- The distance between the object  $j$  and the cluster  $l$  is  $d(\mathbf{x}_i, \bar{\mathbf{x}}_l)$ , where  $d$  is taken to be the Euclidian distance

$$d(\mathbf{x}_i, \bar{\mathbf{x}}_l) = \|\mathbf{x}_i - \bar{\mathbf{x}}_l\| = \left[ \sum_{j=1}^m (x_i^j - \bar{x}_l^j)^2 \right]^{1/2}, \quad i = 1, \dots, m; \quad l = 1, \dots, k.$$

where  $\|\cdot\|$  is the Euclidian norm.

- The error of the partition is taken to be

$$e = \sum_{l=1}^k \sum_{i \in C_l} \|\mathbf{x}_i - \bar{\mathbf{x}}_l\|^2.$$

- Alternatively, we have

$$e = \sum_{i=1}^n \sum_{j=1}^m \|\mathbf{x}_i - \bar{\mathbf{x}}_{l(j)}\|^2,$$

where  $l(i)$  is the index of the cluster of object  $i$ .

- The general procedure is to search for a partition with a small error  $e$  by moving cases from one cluster to another.
- The search ends when no such movement reduces  $e$ .
- **STEP 1.** Assume initial clusters. Compute the cluster means  $\bar{\mathbf{x}}_l$  and the initial error  $e$ .
- **STEP 2.** For the first object, compute for every cluster  $l$

$$\Delta e = \frac{n_l d^2(\mathbf{x}_1, \bar{\mathbf{x}}_l)}{n_l + 1} - \frac{n_{l(1)} d^2(\mathbf{x}_1, \bar{\mathbf{x}}_{l(1)})}{n_{l(1)} - 1}, \quad l = 1, \dots, k, \quad l \neq l(1).$$

- It corresponds to the error variation in transferring the first object from cluster  $l(1)$  to which it belongs to cluster  $l$ .
- If the minimum of this quantity over all  $l \neq l(1)$  is negative, transfer the first case from cluster  $l(1)$  to this minimal  $l$ .
- Adjust the cluster means of  $l(1)$  and the minimal  $l$  and add the error variation (which is negative) to  $e$ .
- **STEP 3.** Repeat STEP 2 for each object  $i$  such that  $2 \leq i \leq n$ .
- **STEP 4.** If 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:

$$\Delta e = \frac{n_l d^2(\mathbf{x}_1, \bar{\mathbf{x}}_l)}{n_l + 1} - \frac{n_{l(1)} d^2(\mathbf{x}_1, \bar{\mathbf{x}}_{l(1)})}{n_{l(1)} - 1}, \quad l = 1, \dots, k, \quad l \neq l(1).$$

## 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 ( $m = 8$ ) are partitioned into three clusters ( $k = 3$ ).

## Exercice 13

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

Table 4.3 Application of K-Means Algorithm to Food Data

	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

INITIAL CLUSTER MEANS			e = 154.9
	Energy	Protein	Calcium
1. HR, CB	8.5	25	1
2. HR, BS	10	28.5	1
3. BB, BC, CC, BH	6.75	33.25	1.5

FIRST CHANGE			e = 108.2
	Energy	Protein	Calcium
1. HR, CB	8.5	25	1
2. HR, BS, BB	10.33	28.67	1
3. BC, CC, BH	5.33	34.67	1.67

SECOND CHANGE			e = 61.4
	Energy	Protein	Calcium
1. BR	13	21	1
2. HR, BS, BB	10.33	28.67	1
3. BC, CC, BH, CB	5	33.25	1.5

STEP 1. A quick initial clustering, which often works well, is based on the case sums. Suppose these are denoted by  $\text{SUM}(J)$ , having minimum value MIN and maximum value MAX. To obtain  $K$  initial clusters, set case  $I$  into the  $J$ th cluster, where  $J$  is the integral part of  $K[\text{SUM}(J) - \text{MIN}]/(\text{MAX} - \text{MIN}) + 1$ . Here the case sums are 41, 39, 35, 40, 41, 34, 42, and 44. The corresponding clusters are 3, 2, 1, 2, 3, 1, 3, and 3. Thus the initial partition is (BR CB) (HR BS) (BB BC CC BH). The values of  $B(L, J)$  ( $1 \leq L \leq 3, 1 \leq J \leq 3$ ) are next computed. For example,  $B(1, 1)$ , the mean of cases in the first cluster for the first variable, equals  $(13 + 4)/2 = 8.5$ . (See Table 4.3 for more.) The error for the initial partition is the sum of squared distances of cases from their cluster means,

$$\begin{aligned} e[P(8, 3)] &= (11 - 8.5)^2 + (29 - 33.25)^2 + (1 - 1.5)^2 + (8 - 10)^2 \\ &\quad + (30 - 28.5)^2 + (1 - 1)^2 + (13 - 8.5)^2 + (21 - 25)^2 \\ &\quad + (1 - 1)^2 + (12 - 10)^2 + (27 - 28.5)^2 + (1 - 1)^2 \\ &\quad + (6 - 6.75)^2 + (31 - 33.25)^2 + (2 - 1.5)^2 + (8.5 - 4)^2 \\ &\quad + (29 - 25)^2 + (1 - 1)^2 + (5 - 6.75)^2 + (36 - 33.25)^2 \\ &\quad + (1 - 1.5)^2 + (5 - 6.75)^2 + (37 - 33.25)^2 + (2 - 1.5)^2 \\ &= 154.9. \end{aligned}$$

The first three squares are the squared distance of BB from its cluster mean (6.75, 33.25, 1.5), and so on.

STEP 2. For the first case, the distances to clusters are

$$\begin{aligned} D(1, 1)^2 &= (11 - 8.5)^2 + (29 - 25)^2 + (1 - 1)^2 = 22.25, \\ D(1, 2)^2 &= 1.25, \\ D(1, 3)^2 &= 36.4. \end{aligned}$$

The increase in error in transferring the first case to cluster 1 is  $2 \times 22.5/3 - 4 \times 36.4/3$ , and that to cluster 2 is  $2 \times 1.25/3 - 4 \times 36.4/3$ . The cluster that is best for the first case is thus the second cluster, and the error reduction is 47.7. The new value of  $e[P(8, 3)]$  is thus 154.9 - 47.7 = 108.2.

It is necessary to update the means of clusters 2 and 3, since cluster 2 has gained the first case and cluster 3 has lost it. For example,

$$\begin{aligned} B(2, 1) &= (11 + 2 \times 10)/3 = 10.33, \\ B(2, 2) &= (29 + 2 \times 28.5)/3 = 28.67, \\ B(2, 3) &= (1 + 2 \times 1)/3 = 1.00. \end{aligned}$$

STEP 3. Repeating Step 1 on all cases, for case 2, cluster 2 is far closer than any other, and case 2 remains in cluster 2, with no change taking place. Continuing, no change takes place until case 6, which moves to cluster 3. No further changes occur in this pass.

STEP 4. Since some changes occurred in the last pass, another pass is necessary through all cases. No changes occur on this pass and the algorithm stops with the final cluster (BR) (HR BS BB) (BC CC BH CB). These clusters are characterized by the variables as follows: The first cluster is high in energy and low in protein, the second is high in energy and protein, and the third is low in energy and high in protein. Calcium hardly matters. The complete data set is partitioned in Table 4.4.

```
#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 5.000000 33.25000 1.5
## 2 13.000000 21.00000 1.0
## 3 10.333333 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).

**Table 4.4**  
Clusters and cluster means from K-means algorithm applied to food data on energy, protein, and calcium expressed as percentages of daily requirements. The two clusters obtained by splitting cluster 3 are denoted by 31 and 32.

PARTITION	CLUSTERS	ENERGY	PROTEIN	CALCIUM
1	1 : BB HR BR BS BC CB CG HH LL LL HS PR PS BT VC FB AR AC TC HF MB MC PF SC DS UC RC	6.5	27.1	5.5
2	11 : BB HR BR BS BC CB CG HH LL LL HS PR PS BT VC FB AR AC TC HF MB PF UC RC 12 : MC SC DS	6.7	27.3	2.6
3	12 111 : BB HR BR BS BC CB CG HH LL LL HS PR PS BT VC FB HF MB PF UC RC 112 : AR AC TC	4.7	26.1	28.5
4	112 2 : BB HR BR BS BC CB CG HH LL LL HS PR PS BT VC FB HF MB PF UC 3 : MC SC RC 4 : DS	7.4	29.0	1.8
5	112, 3, 4 21 : HR BC CB CG HH VC FB UC 22 : BB BR BS LL LL HS PR PS BT HF MB PF	2.1	15.2	8.1
6	112, 21, 22, 4 31 : MC SC 32 : RC	7.5	28.8	1.3
7	112, 21, 31, 32, 4 221 : BB BS LL LL HS PR PS BT HF MB PF 222 : BR	4.0	26.7	17.3
8	222, 31, 32, 4 5 : BC CC BH VC FB UC 6 : AR AC 7 : BB HR BS LL LL HS PR PS MB 8 : CB BT TC HF PF	5.6	31.4	45.9
9	222, 31, 32, 4, 5, 6, 9 : BB HR BS LL LL HS PR PS 10 : CB BT HF MB PF 11 : TC	9.6	27.8	1.1
		10.0	27.9	1.1
		5.3	25.4	1.2
		2.8	20.0	4.8

## 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.

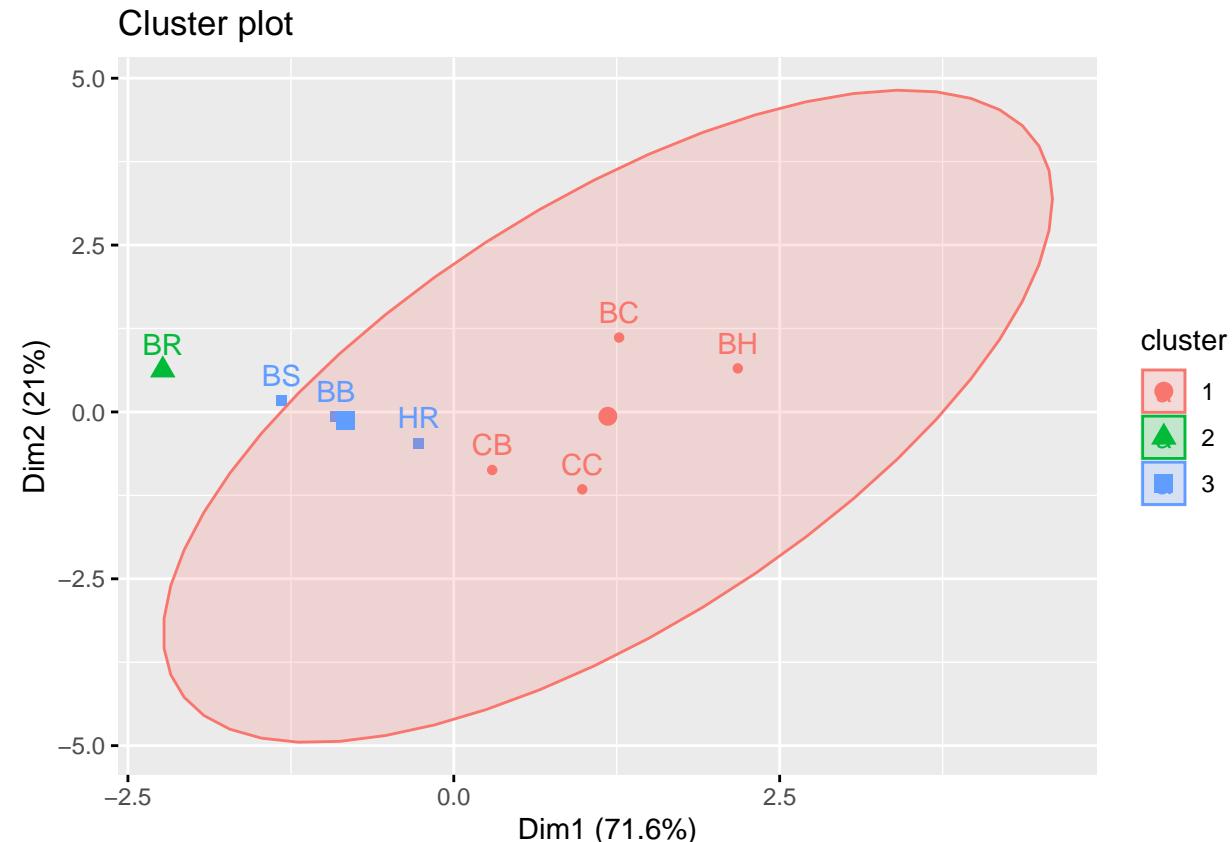
## Visualizing k-means results

- 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
```



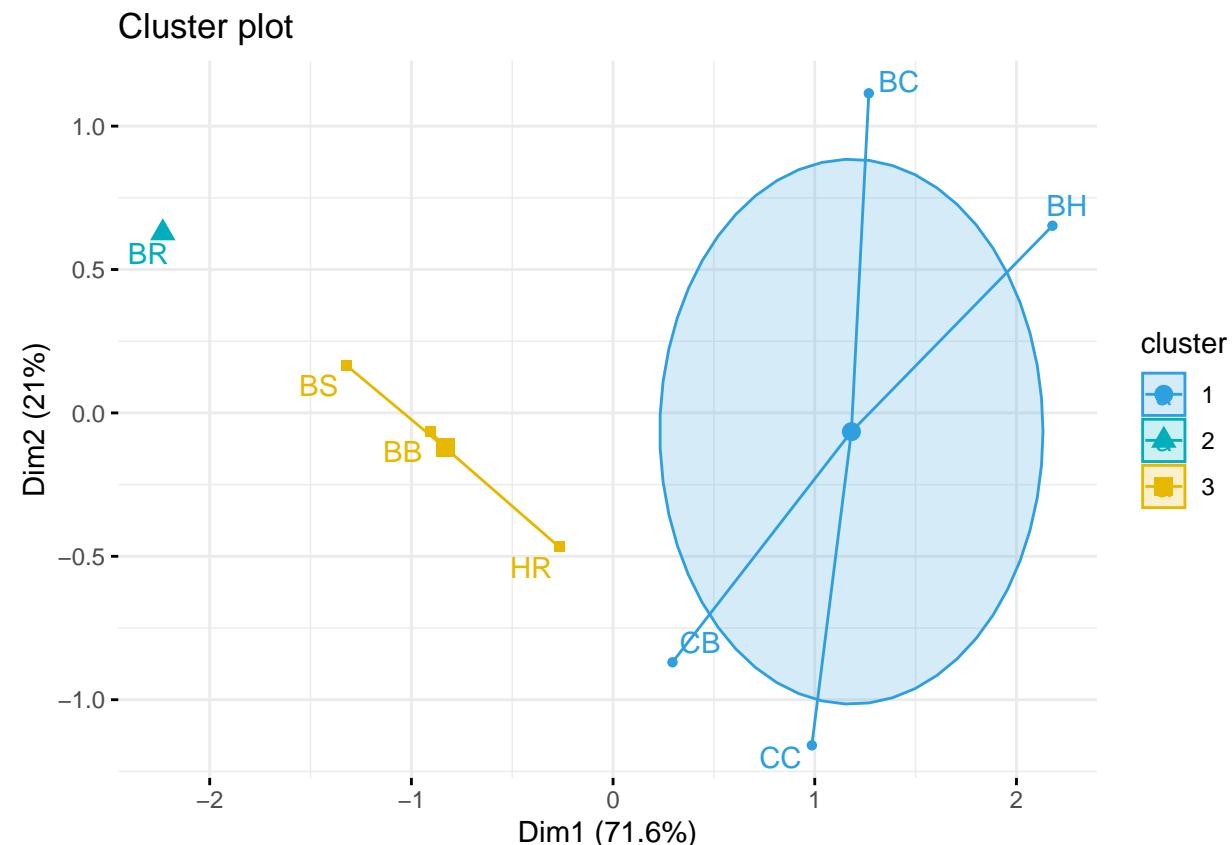
```
fviz_cluster(km.res, data = df[1:8,],
palette = c("#2E9FDF", "#OOAFBB", "#E7B800"),
ellipse.type = "euclid", # Concentration ellipse
star.plot = TRUE, # Add segments from centroids to items
repel = TRUE, # Avoid label overplotting (slow)
```

```

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)

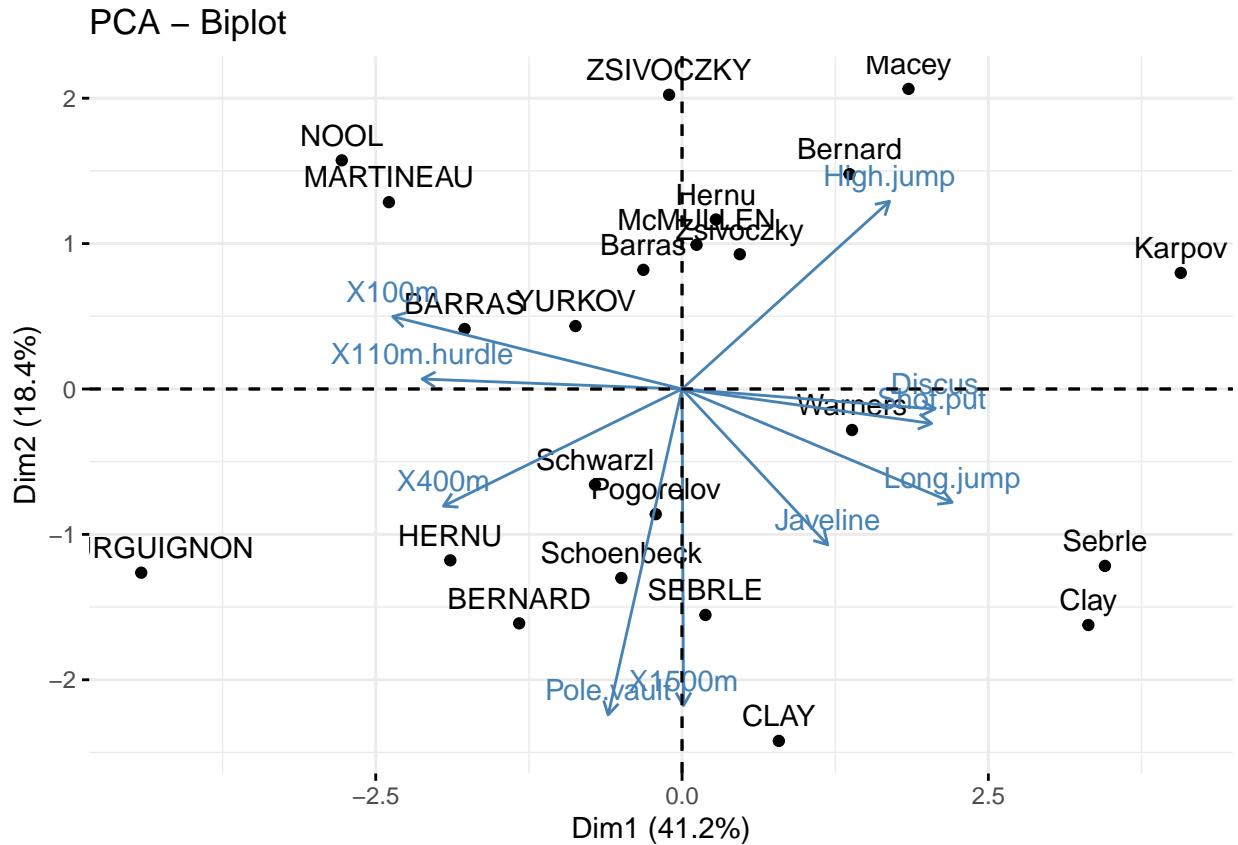
## Importance of components:
##                 PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation 2.0308 1.3559 1.1132 0.90523 0.83759 0.65029 0.55007
## Proportion of Variance 0.4124 0.1839 0.1239 0.08194 0.07016 0.04229 0.03026

```

```

## 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  $k$  clusters.
- For  $i = 1, \dots, n$ , let:

$$a_i = \frac{1}{n_{l(i)} - 1} \sum_{i' \in i \in C_{l(i)} \setminus \{i\}} d(i, i'),$$

be the mean distance between  $i$  and all the points of the same cluster.

- If  $n_{l(i)} = 1$ , we set  $a_i = 0$ .
- It is interpreted as a measure of how well  $i$  is assigned to its cluster (smaller the value, better is the assignment).
- Let also

$$b_i = \min_{l \neq l(i)} \frac{1}{|n_l|} \sum_{i' \in C_l} d(i, i'),$$

be the “neighboring cluster” of  $i$ .

- We now define a *silhouette* of  $i$  as

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \text{ if } n_{l(i)} > 1,$$

and

$$s_i = 0 \text{ if } n_{l(i)} = 1.$$

- Alternatively, we have

$$s_i = \begin{cases} 1 - a_i/b_i, & \text{if } a_i < b_i; \\ 0, & \text{if } a_i = b_i; \\ b_i/a_i - 1, & \text{if } a_i > b_i. \end{cases}$$

- From the above definition it is clear that

$$-1 \leq s_i \leq 1$$

- For  $s_i$  to be close to 1 we require  $a_i \ll b_i$ .
- If  $s_i$  is close to  $-1$ ,  $i$  is more appropriately clustered in its neighbouring cluster.
- An  $s_i$  near zero means that the  $i$  is on the border of two natural clusters.

## Silhouette example

```
library(knitr)
knitr::opts_chunk$set(echo = TRUE)
km.res<-kmeans(df[1:8,],3,iter.max = 100)
slobj<-silhouette(km.res$cluster,dist(df[1:8,]))
row.names(slobj)<-row.names(df[1:8,])

knitr::kable(slobj[,], caption = "Silhouettes values for the food example",digits = 4,col.names = c("Cluster", "Neighbo", "Silhouette width"))
kable_classic(latex_options = "hold_position")%>%
row_spec(1:2,background = "#2EFEF7")%>%
row_spec(3:5,background = "#F3F781")%>%
row_spec(6:8,background = "#FA58F4")
```

Table 6: Silhouettes values for the food example

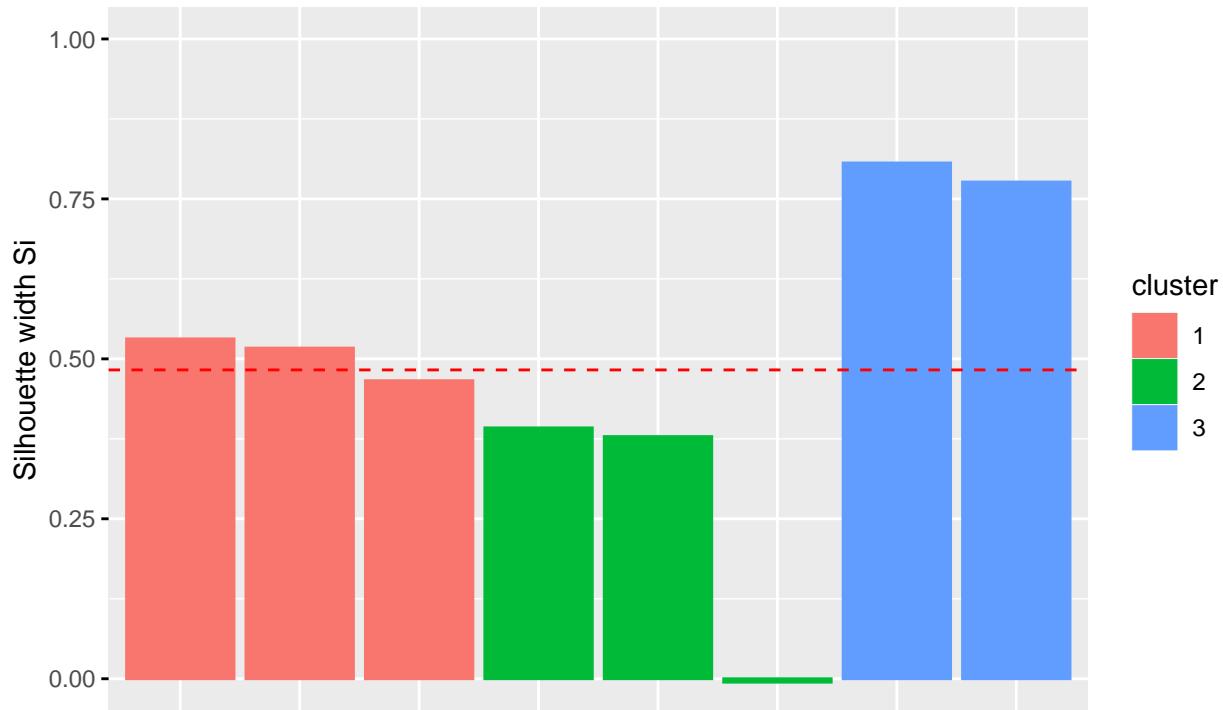
	Cluster	Neighbor	Silhouette width
BB	2	1	-0.0053
HR	1	2	0.4659
BR	2	1	0.3785
BS	2	1	0.3921
BC	1	3	0.5168
CB	1	3	0.5312
CC	3	1	0.7764
BH	3	1	0.8062

## Silhouette graph

```
silhouette(km.res$cluster,dist(df[1:8,]))%>%
fviz_silhouette()

##   cluster size ave.sil.width
## 1       1     3        0.50
## 2       2     3        0.26
## 3       3     2        0.79
```

Clusters silhouette plot  
Average silhouette width: 0.48



## 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..

Country	Dissimilarities to other countries										
	BEL	BRA	CHI	CUB	EGY	FRA	IND	ISR	USA	USS	YUG
BRA	5.58										
CHI	7.00	6.50									
CUB	7.08	7.00	3.83								
EGY	4.83	5.08	8.17	5.83							
FRA	2.17	5.75	6.67	6.92	4.92						
IND	6.42	5.00	5.58	6.00	4.67	6.42					
ISR	3.42	5.50	6.42	6.42	5.00	3.92	6.17				
USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75			
USS	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17		
YUG	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67	
ZAI	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92

- The author obtain the following two figures.

Fig. 2. Silhouettes of a clustering with  $k = 2$  of the twelve countries data of Table 1.

Fig. 3. Silhouettes of a clustering with  $k = 3$  of the twelve countries data.

- Your task is to produce the same figures using modern dataviz tools.
  - What did you learn from these 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  $k$  representative objects among  $n$  objects.
  - These objects should represent various aspects of the structure of the data.
  - In the PAM algorithm, the representative objects are called *medoids*.
  - After finding a set of  $k$  representative objects, the  $k$  clusters are constructed by assigning each object of the data set to the nearest representative object.

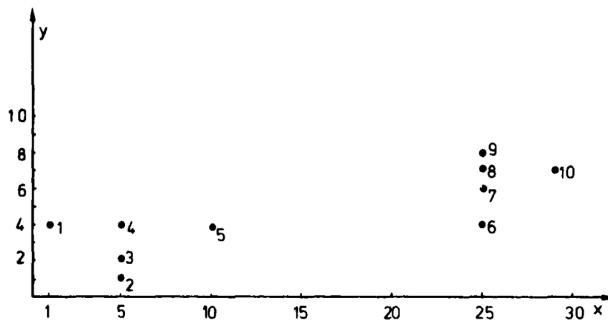
- To illustrate algorithm, consider a data set of ten objects ( $n = 10$ ) and two variables ( $m = 2$ ).

**Table 1 Coordinates of the Objects of the Example of Figure 1**

Number	x Coordinate	y Coordinate
1	1.0	4.0
2	5.0	1.0
3	5.0	2.0
4	5.0	4.0
5	10.0	4.0
6	25.0	4.0
7	25.0	6.0
8	25.0	7.0
9	25.0	8.0
10	29.0	7.0

Figure 7: From Kaufman & Rousseeuw, p. 68-72

- Suppose the data set must be divided into two clusters ( $k = 2$ ).
- The algorithm first considers possible choices of two representative objects.
- Then, it constructs the clusters around these representative objects.



**Figure 1** Two-dimensional example with 10 objects.

- As an example, suppose objects 1 and 5 are the selected representative objects.
- In Table 2, the 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.

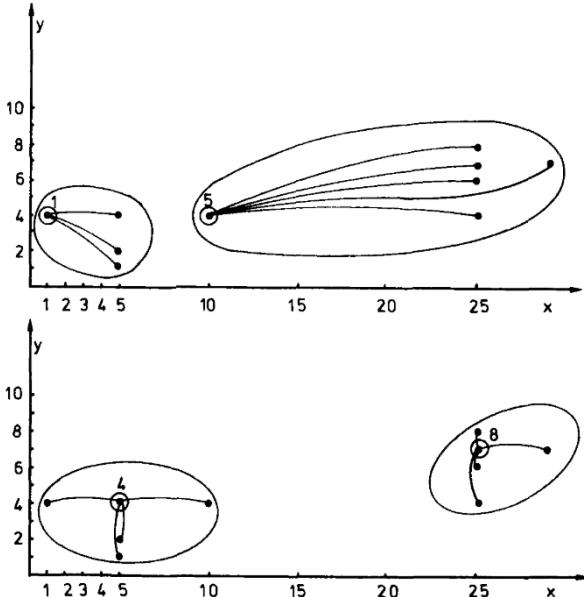
**Table 2 Assignment of Objects to Two Representative Objects**

Object Number	Dissimilarity from Object 1	Dissimilarity from Object 5	Minimal Dissimilarity	Closest Representative Object
1	0.00	9.00	0.00	1
2	5.00	5.83	5.00	1
3	4.47	5.39	4.47	1
4	4.00	5.00	4.00	1
5	9.00	0.00	0.00	5
6	24.00	15.00	15.00	5
7	24.08	15.13	15.13	5
8	24.19	15.30	15.30	5
9	24.33	15.52	15.52	5
10	28.16	19.24	19.24	5
Average 9.37				

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

**Table 3 Assignment of Objects to Two Other Representative Objects**

Object Number	Dissimilarity from Object 4	Dissimilarity from Object 8	Minimal Dissimilarity	Closest Representative Object
1	4.00	24.19	4.00	4
2	3.00	20.88	3.00	4
3	2.00	20.62	2.00	4
4	0.00	20.22	0.00	4
5	5.00	15.30	5.00	4
6	20.00	3.00	3.00	8
7	20.10	1.00	1.00	8
8	20.22	0.00	0.00	8
9	20.40	1.00	1.00	8
10	24.19	4.00	4.00	8
Average 2.30				



**Figure 2** Clusterings corresponding to the selections described in Tables 2 and 3.

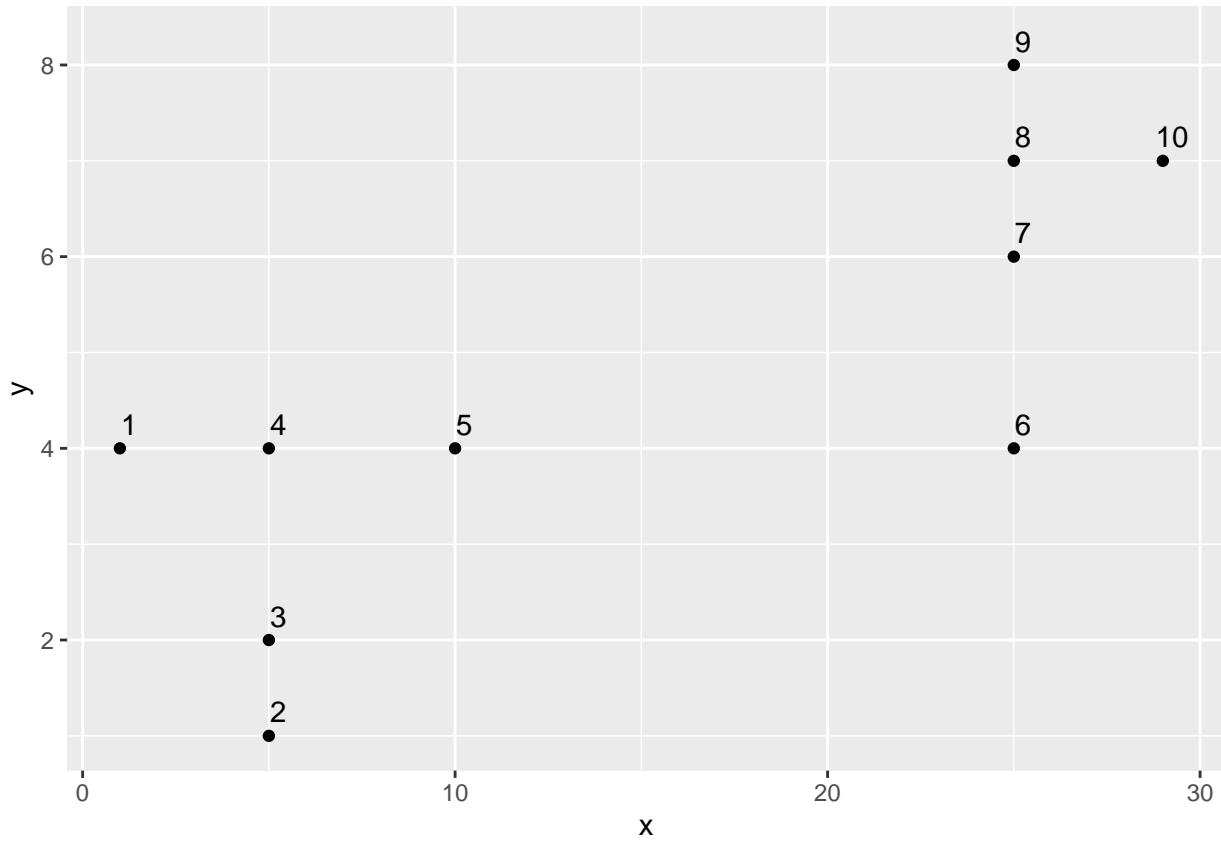
- The average dissimilarity for the case objects 4 and 8 are selected is 2.30, which is considerably less than the value of 9.37, found when 1 and 5 were the representative objects.
- Alternatively a PAM program can be used by entering a matrix of dissimilarities between objects.
- The algorithms are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

```
x=c(1,5,5,5,10,25,25,25,29)
```

```
y=c(4,1,2,4,4,4,6,7,8,7)
```

```
df<-data.frame(x,y)
```

```
ggplot(df,aes(x=x,y=y,label=row.names(df)))+geom_point()+geom_text(label=row.names(df),nudge_x = 0.25, check_overlap = T)
```



```

dist(df,upper=T,diag=T)%>%
as.matrix()~>M
cbind(M[,c(1,5)],apply(M[,c(1,5)],1,min))%>%
data.frame()~>T2
T2$val=0
T2$val[which(T2[,1]==apply(M[,c(1,5)],1,min))]<~-1
T2$val[which(T2[,2]==apply(M[,c(1,5)],1,min))]<~-5
names(T2)~<-c("Dist. wrt 1","Dist. wrt 5","Min. dist.","Closest rep. obj.")

kable(T2,digit =2,align = "lccc",booktabs = TRUE)%>%
kable_classic(latex_options = "hold_position")

```

Dist. wrt 1	Dist. wrt 5	Min. dist.	Closest rep. obj.
0.00	9.00	0.00	1
5.00	5.83	5.00	1
4.47	5.39	4.47	1
4.00	5.00	4.00	1
9.00	0.00	0.00	5
24.00	15.00	15.00	5
24.08	15.13	15.13	5
24.19	15.30	15.30	5
24.33	15.52	15.52	5
28.16	19.24	19.24	5

```

mean(T2$`Min. dist.`)

## [1] 9.36615

dist(df,upper=T,diag=T)%>%
as.matrix() -> M
cbind(M[,c(4,8)],apply(M[,c(4,8)],1,min))%>%
data.frame() -> T3
T3$val=0
T3$val[which(T3[,1]==apply(M[,c(4,8)],1,min))]->1
T3$val[which(T3[,2]==apply(M[,c(4,8)],1,min))]->5
names(T3)<-c("Dist. wrt 4","Dist. wrt 8","Min. dist.","Closest rep. obj.")

kable(T3,digit =2,align = "lccc",booktabs = TRUE)%>%
kable_classic(latex_options = "hold_position")

```

	Dist. wrt 4	Dist. wrt 8	Min. dist.	Closest rep. obj.
4.00	24.19	4	1	
3.00	20.88	3	1	
2.00	20.62	2	1	
0.00	20.22	0	1	
5.00	15.30	5	1	
20.00	3.00	3	5	
20.10	1.00	1	5	
20.22	0.00	0	5	
20.40	1.00	1	5	
24.19	4.00	4	5	

```

mean(T3$`Min. dist.`)

## [1] 2.3

PAM<-pam(df,2)
PAM$medoids

##      x y
## [1,] 5 2
## [2,] 25 7

PAM$id.med

## [1] 3 8

PAM$clustering

## [1] 1 1 1 1 1 2 2 2 2 2

PAM$objective

##      build      swap
## 3.421612 2.185730

PAM$isolation

##  1 2
## L* L*
```

```
## Levels: no L L*
```

## *L* and *L*<sup>\*</sup> clusters

The clustering characteristics:

- whether a cluster is a singleton
- whether a cluster is isolated: Two types of isolated clusters are considered in the program, *L*-clusters and *L*<sup>\*</sup>-clusters. They have

HOW TO USE THE PROGRAM PAM

83

the following definitions (see Gordon, 1981):

*C* is an *L*-cluster if for each object *i* belonging to *C*:

$$\max_{j \in C} d(i, j) < \min_{h \notin C} d(i, h)$$

*C* is an *L*<sup>\*</sup>-cluster if:

$$\max_{i, j \in C} d(i, j) < \min_{l \in C, h \notin C} d(l, h) \quad (5)$$

It is clear that *L*<sup>\*</sup>-clusters are also *L*-clusters. It should be observed that the property of being isolated depends both on the internal structure of a cluster and its position relative to other clusters. When dividing the data into two clusters, it can happen that one is isolated and the other is not.

- the diameter and separation of each cluster: the left-hand side of Eq. (5) is called the diameter of cluster *C* and the right-hand side its separation.
- in each cluster, the average dissimilarity of the objects to the medoid and the maximum dissimilarity of any object to the medoid.

Dist. wrt 3	Dist. wrt 8	Min. dist.	Closest rep. obj.
4.47	24.19	4.47	3
1.00	20.88	1.00	3
0.00	20.62	0.00	3
2.00	20.22	2.00	3
5.39	15.30	5.39	3
20.10	3.00	3.00	8
20.40	1.00	1.00	8
20.62	0.00	0.00	8
20.88	1.00	1.00	8
24.52	4.00	4.00	8

```
## [1] 2.18573
```

```
##      1     2     3     4     5
## 9.000000 5.830952 5.385165 5.000000 9.000000
##      1     2     3     4     5
## 24.000000 20.22375 20.09975 20.00000 15.00000
```

```
##      1      2      3      4      5
## TRUE TRUE TRUE TRUE TRUE
## [1] 9
## [1] 15
## [1] TRUE
##      6      7      8      9      10
## 5.000000 4.123106 4.000000 4.123106 5.000000
##      6      7      8      9      10
## 15.00000 15.13275 15.29706 15.52417 19.23538
##      6      7      8      9      10
## TRUE TRUE TRUE TRUE TRUE
## [1] 5
## [1] 15
## [1] TRUE
```