Cluster Analysis

Contents

- Required packages

```
knitr::opts_chunk$set(echo = TRUE)
#install.packages("dplyr","ade4","magrittr","cluster","factoextra","cluster.datasets","xtable","kableEx
knitr::opts_chunk$set(echo = TRUE)
```

Definition of a distance

- A distance function or a metric on $\mathbb{R}^m$, $m \geq 1$, is a function $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$.
- 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 $\mathbf{x} = (x_1, \cdots, x_m)$, $\mathbf{y} = (y_1, \cdots, y_m)$ and $\mathbf{z} = (z_1, \cdots, z_m)$ 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

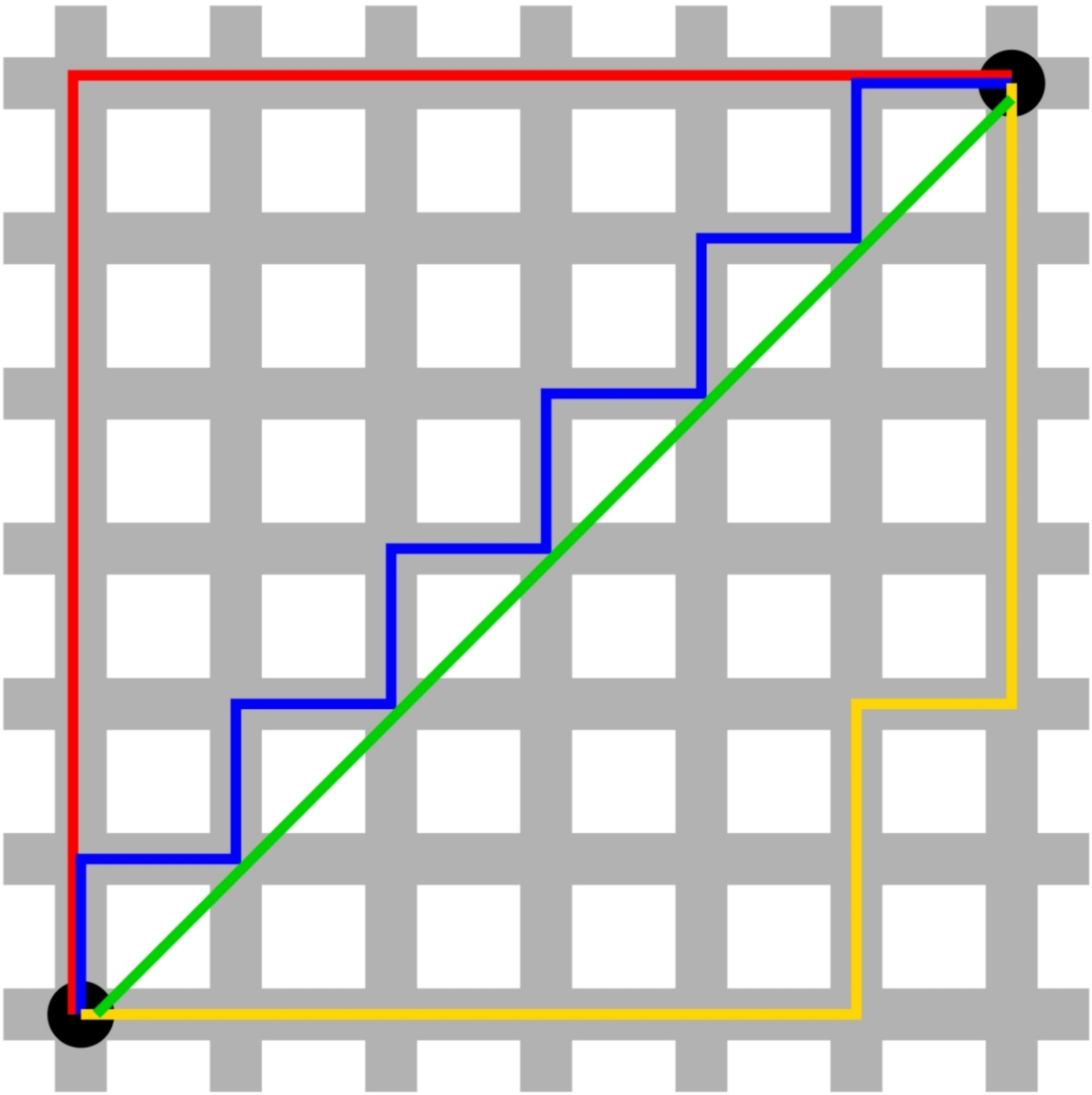
- Is the squared Euclidian distance a true distance?

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 0.000000 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:

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

- 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.

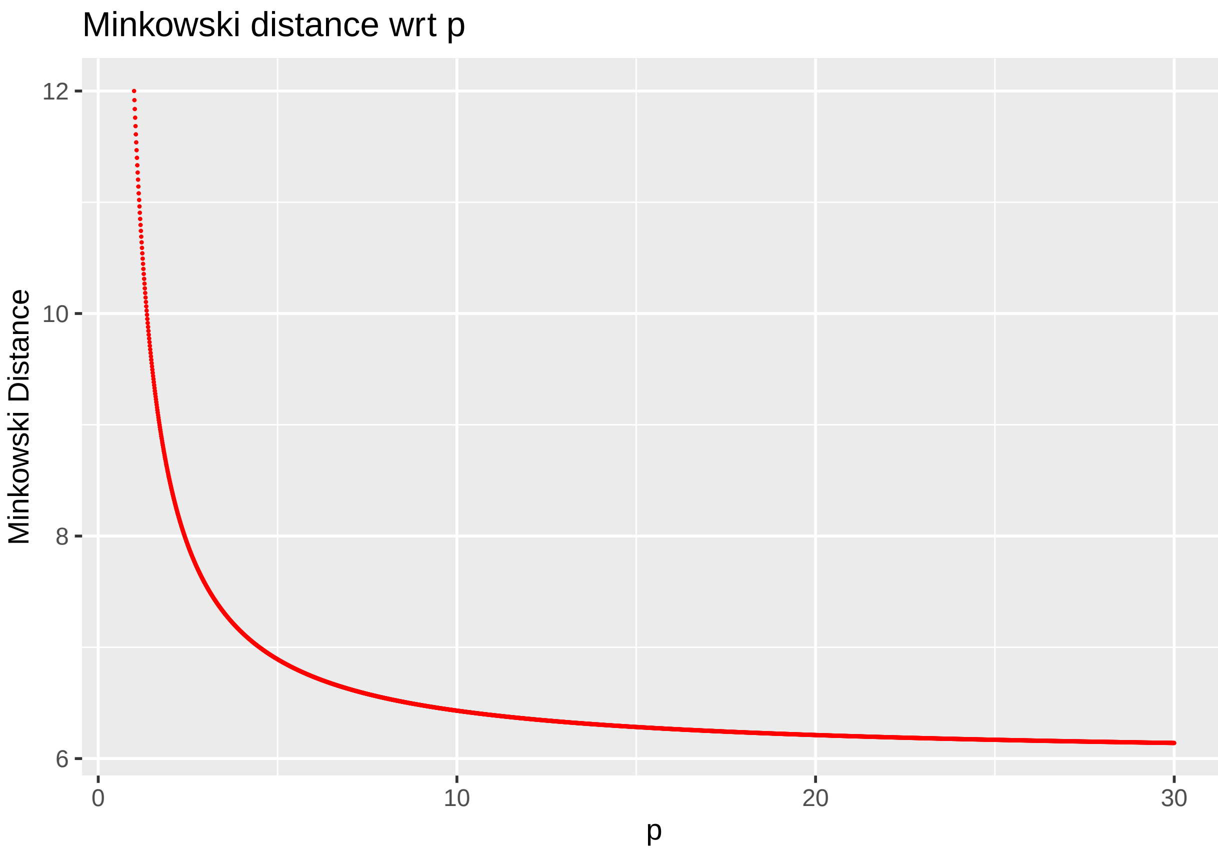
Minkowski distance

- Both the Euclidian and the Manat、tan distances are special cases of the Minkowski distance which is defined, for $p \geq 1$, by:
$$d(\mathbf{x}, \mathbf{y}) = \left[\sum_{j=1} |x_j - y_j|^p\right]^{1/p}.$$
- 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 $\mathbb{R}^m$.

```r
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:

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

- The corresponding norm is:

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

Minkowski inequality

- The proof of the triangular inequality A3 is based on the Minkowski inequality:

- For any nonnegative real numbers $a_1, \cdots, a_m$; $b_1, \cdots, b_m$, 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 $b_j = |y_j - z_j|$, $j = 1, \cdots, n$, 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 $a_1, \cdots, a_m$; $b_1, \cdots, b_m$, 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 occuring 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 od thes above inequality is greater then $|\sum_{j=1}^{m} x_j y_j|$, 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:

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

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

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

- 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 fullfilled.

Cosine correlation distance

- The cosine of the angle θ 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 seperately each of the vectors to ranked data values:

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

- 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 correspnding 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 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 $d_j = x_j^r - y_j^r,\ j=1,\cdots,n$.
- 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:

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

  and to be *discordant* if:

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

  where 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 τ 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 uniteless variables.
- A well known method is the z-score transformation.
- Let $\mathbf{v} \equiv (v_1, \cdots, v_n)$ a vector of measurements recrded for n individuals or objects.

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

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

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

- 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 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 2×2 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

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

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

- 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,
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", "
```

- 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.
- 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.

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 measues embedding d 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 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.

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 as $0, 1$.
- 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 .
- Le M be the number of states and code the outcomes as $1, \cdots, M$.
- We may choose $1 =$ blue, $2 =$ brown, $3 =$ green, and $4 =$ 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.

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

variable		END 1	END 0	similarity
WAR	1	5	4	36
	0	1	8	
FLY	1	1	3	6
	0	5	9	
VER	1	6	7	30
	0	0	5	
HAI	1	2	5	6
	0	4	7	

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

Figure 1: KAUFMAN and ROUSSEEUW, p. 296

REVISED DATA
****

	WAR	FLY	VER	END	GRO	HAI
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

Figure 2: KAUFMAN and ROUSSEEUW p. 297

- If $\mathbf{x}, \mathbf{y}$, are both n 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:

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

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

$$N_{b+c} = n - N_{a+d}.$$

- Let N_d be the number of attributes on which the two records match in a "not applicable" category:

$$N_d = \sum_{j=1}^{m} \delta(x_j, y_j).$$

- The distance corresponding to the simple matching approach is:

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

- Therefore:

$$d(\mathbf{x}, \mathbf{y}) = \frac{N_{a+d}}{N_{a+d} + N_{b+c}}.$$

- Note that simple matching has exactly the same meaning as in the preceding section.

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 wheight 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 jth measure or variable to the dissimilarity measure.
- If the jth 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 jth 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 i over the available data.
- Consider the following data set:

	variable							
object	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.

Data

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 (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)
```

```
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:kableExtra':
##
##     group_rows
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
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")
```

```
      )
```

```
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)
```

```
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
```

```
daisy(data[,1:8],metric = "Gower")
```

```
## Warning in daisy(data[, 1:8], metric = "Gower"): with mixed variables, metric
## "gower" is used automatically
```

```
## Dissimilarities :
##            1         2         3         4         5         6         7
## 2  0.8875408
## 3  0.5272467 0.5147059
## 4  0.3517974 0.5504493 0.5651552
## 5  0.4115605 0.6226307 0.3726307 0.6383578
## 6  0.2269199 0.6606209 0.3003268 0.4189951 0.3443627
## 7  0.2876225 0.5999183 0.4896242 0.3435866 0.4197712 0.1892974
## 8  0.4234069 0.4641340 0.6038399 0.2960376 0.4673203 0.5714869 0.4107843
## 9  0.5808824 0.4316585 0.4463644 0.8076797 0.3306781 0.5136846 0.5890931
## 10 0.6094363 0.4531046 0.4678105 0.5570670 0.3812908 0.4119281 0.5865196
## 11 0.3278595 0.7096814 0.5993873 0.6518791 0.3864788 0.4828840 0.5652369
## 12 0.4267565 0.5857843 0.6004902 0.5132761 0.5000817 0.5248366 0.6391340
## 13 0.5196487 0.5248366 0.5395425 0.7464461 0.2919118 0.4524510 0.5278595
## 14 0.2926062 0.5949346 0.6096405 0.3680147 0.5203431 0.3656863 0.5049837
## 15 0.6221814 0.3903595 0.5300654 0.5531454 0.4602124 0.5091503 0.3345588
## 16 0.6935866 0.3575163 0.6222222 0.3417892 0.7301471 0.5107843 0.4353758
## 17 0.7765114 0.1904412 0.5801471 0.4247141 0.6880719 0.5937092 0.5183007
## 18 0.4610294 0.4515114 0.7162173 0.4378268 0.4755310 0.6438317 0.4692402
##            8         9        10        11        12        13        14
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9  0.6366422
## 10 0.6639706 0.4256127
## 11 0.4955474 0.4308007 0.3948121
## 12 0.4216503 0.4194036 0.3812092 0.2636029
## 13 0.5754085 0.2181781 0.3643791 0.3445670 0.2331699
## 14 0.4558007 0.4396650 0.3609477 0.2838644 0.1591503 0.3784314
## 15 0.4512255 0.2545343 0.4210784 0.4806781 0.4295752 0.3183007 0.4351307
## 16 0.6378268 0.6494690 0.3488562 0.7436683 0.6050654 0.5882353 0.4598039
## 17 0.4707516 0.6073938 0.3067810 0.7015931 0.5629902 0.5461601 0.5427288
## 18 0.1417892 0.5198529 0.8057598 0.5359477 0.5495507 0.5733252 0.5698121
##           15        16        17
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16 0.3949346
## 17 0.3528595 0.1670752
```

```
## 18 0.5096814 0.7796160 0.6125408
##
## Metric :  mixed ;  Types = N, N, N, N, O, O, I, I
## Number of objects : 18
```

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().

```
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