uc3m | Universidad Carlos III de Madrid

Grado Estadística y Empresa

2021/22

Distancias Estadisticas: Teoria y Aplicaciones

Fabio Scielzo Ortiz

ÍNDICE GENERAL

1. DISTANCIAS ESTADISTICAS:

El concepto de distancia entre elementos de un conjunto ε permite interpretar geometricamente muchas técnicas clásicas del análisis multivariante .

Esta interpretación es posible tanto con variables cuantitativas como categoricas, o incluso cuando no se dispone de variables, siempre que tenga sentido obtener una medida de proximidad entre los elementos de ε

1.1. Definición de distancia:

Dado un conjunto de elementos ε

Casi-Métrica:

Se denomina **casi-metrica** o disimilaridad a toda aplicación $\delta : \varepsilon x \varepsilon \rightarrow \mathbb{R}$ que cumpla las siguientes propiedades:

1) $\delta(i, j) \geq 0 \ , \forall i, j$

2) $\delta(i, i) = 0 \ , \forall i$

3) $\delta(i, j) = \delta(j, i) \ , \forall i, j$

Semi-Métrica:

Se denomica **semi-metrica** a toda disimilaridad que cumpla la desigualdad triangular:

4) $\delta(i, j) \leq \delta(i, k) + \delta(k, i) \ , \forall i, j, k$

Métrica:

Se denomina **metrica** a toda semi-metrica que cumple:

5) $\delta(i, j) = 0 \Leftrightarrow i = j$

Distancia:

Una **distancia** es una **métrica** o una **semi-métrica**

1.2. Matriz de distancias:

Cuando ε sea un conjunto finito , tendremos una matriz de distancias:

Matriz de distancias:

$$D = \begin{pmatrix} 0 & \delta_{12} & ... & \delta_{1n} \\ \delta_{21} & 0 & ... & \delta_{2n} \\ ... & ... & ... & ... \\ \delta_{n1} & \delta_{n2} & ... & 0 \end{pmatrix}$$

con $\delta_{ij} = \delta_{ji}$

También usaremos la matriz de cuadrados de distancias:

Matriz de distancias al cuadrado:

$$D^{(2)} = \begin{pmatrix} 0 & \delta_{12}^2 & ... & \delta_{1n}^2 \\ \delta_{21}^2 & 0 & ... & \delta_{2n}^2 \\ ... & ... & ... & ... \\ \delta_{n1}^2 & \delta_{n2}^2 & ... & 0 \end{pmatrix}$$

No debe confundirse con $D^2 = D \cdot D$

2. DISTANCIAS CON VARIABLES CUANTITATIVAS:

Sean $X_1, ..., X_p$ variables cuantitativas,

Sean $x_i = (x_{i1}, ..., x_{ip})^t$ y $x_j = (x_{i1}, ..., x_{ip})^t$ los valores (observaciones) de las variables $X_1, ..., X_p$ para los elementos o individuos i y j de la muestra.

2.1. Distancia Euclidea:

Distancia Euclidea:

La distancia euclidea entre los elementos / individuos i y j respecto de las variables cuantitativas $X_1, ..., X_p$ se define como:

$$\delta^2(i, j)_{Euclidea} = \sum_{k=1}^{p} (x_{ik} - x_{jk})^2 = (x_i - x_j)^t \cdot (x_i - x_j)$$

$$\delta(i, j)_{Euclidea} = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2} = \sqrt{(x_i - x_j)^t \cdot (x_i - x_j)}$$

2.1.1. Inconvenientes:

Pese a que es una de las distancias mas conocidas no es adecuada en muchos casos por las siguientes razones:

1) **Presupone** que las **variables** son **incorreladas** y con **varianza unidad**.

2) **No** es **invariante frente a cambios de escala** (cambios de unidades de medida) de las variables.

Veamos que significa esto ultimo con mas detalle:

Si se aplica un cambio de escala a las variables $a \cdot X_j + b$, con $a \neq 1$ y $b \neq 0$

Ahora las observaciones para los elementos i y j son $a \cdot x_i + b$ y $a \cdot x_j + b$

Entonces la distancia euclidea entre los elementos i y j respecto de las variables escaladas $a \cdot X_j + b$ es:

$$\delta^2(i, j)_{Euclidea} = a^2 \cdot (x_i - x_j)^t \cdot (x_i - x_j)$$

2.1.2. Aplicación en R: Data-set

Data-set de trabajo, tendra 4 variables cuantitativas, 3 binarias y 3 categoricas multiples:

```r
#Cuantitativas
X1 <- rnorm(50, mean=10 , sd=15)
X2 <- rnorm(50, mean=10 , sd=15)
X3 <- rnorm(50, mean=10 , sd=15)
X4 <- rnorm(50, mean=10 , sd=15)

#Binarias
X5<- round(runif(50))
X6<- round(runif(50))
X7<- round(runif(50))

#Categoricas multiples
X8<-round(runif(50, min=0, max=4)) #categorias: 0,1,2,3,4
X9<-round(runif(50, min=0, max=3))  #categorias: 0,1,2,3
X10<-round(runif(50, min=0, max=5))  #categorias: 0,1,2,3,4,5
```

```r
library(tidyverse)

Datos_Mixtos<-tibble(X1,X2,X3,X4,X5,X6,X7,X8,X9,X10)
```

Observación: al no haber fijado semilla aleatoria, los resultados numericos que en este trabajo se obtengan no se obtendran si se reproduce el codigo, debido a la aleatoriedad del data-set.

```r
Datos_Cuantitativos <- Datos_Mixtos%>%select(1:4)
```

X1 <dbl>	X2 <dbl>	X3 <dbl>	X4 <dbl>
-8.1332426	4.8784024	-17.0687714	-4.4326816
-14.2068498	13.0333758	-21.8233349	-0.6493458
12.2132856	-8.7883466	36.6426908	-5.5453174
22.7573377	10.3320840	2.9471420	-14.3266445
6.3147590	25.0873836	0.3277923	18.2358284
-14.8599494	24.0703325	1.0882005	10.4715531
9.3842225	-14.5996736	1.7041652	35.7188878
33.6985681	5.4460857	22.9162756	21.0916602
-0.7572772	15.4888918	13.2406387	-0.5323519
0.9396013	19.2296040	0.7513538	1.7749528
26.5205561	14.3328379	19.7931778	10.5736266
9.9669342	0.3729050	17.9946197	-8.0098499
1.4609010	-4.6219132	9.1251820	-6.1557409
-3.4954412	7.7376606	31.2834967	12.5695518
-5.4234701	10.1284803	37.3394583	18.8686520

1-15 of 50 rows Previous 1 2 3 4 Next

2.1.3. Aplicación en R: Distancia euclidea

Programamos la distancia Euclidea:

```r
Dist_Euclidea <- function(i,j, Matriz_Datos_Cuantitativos){

Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_
   Cuantitativos)

Dist_Euclidea = (Matriz_Datos_Cuantitativos[i,] - Matriz_Datos
   _Cuantitativos[j,])%*%(Matriz_Datos_Cuantitativos[i,] -
   Matriz_Datos_Cuantitativos[j,])

Dist_Euclidea<-sqrt(Dist_Euclidea)

return(Dist_Euclidea)
}
```

```r
Dist_Euclidea(1,2, Datos_Cuantitativos)
```

```
         [,1]
[1,] 11.84533
```

Programamos la matriz de distancias Euclideas:

```
Matriz_Dist_Euclideas <- function( Matriz_Datos_Cuantitativos
    ){

  Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_
      Cuantitativos)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Cuantitativos)[1] ,
      nrow=dim(Matriz_Datos_Cuantitativos)[1] )

  for(i in 1:dim(Matriz_Datos_Cuantitativos)[1] ){
    for(j in 1:dim(Matriz_Datos_Cuantitativos)[1]){

  M[i,j]=Dist_Euclidea(i,j, Matriz_Datos_Cuantitativos)

   }
  }
  return(M)
}
```

```
Matriz_Dist_Euclideas(Datos_Cuantitativos)
```

Las primeras 20 filas y 11 columnas de la matriz de distancias obtenida son:

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]
[1,]	0.00000	11.84533	59.05015	38.50322	37.86356	31.07074	51.48620	63.24965	33.17921	25.38435	53.612169
[2,]	11.84533	0.00000	67.94452	46.62932	37.59997	27.76425	56.53597	69.47518	37.63531	27.98753	59.315255
[3,]	59.05015	67.94452	0.00000	41.10073	55.37754	57.73487	54.45374	39.52442	36.47473	47.47500	35.819179
[4,]	38.50322	46.62932	41.10073	0.00000	39.43669	47.14028	57.50246	42.38875	29.59338	28.62276	30.561091
[5,]	37.86356	37.59997	55.37754	39.43669	0.00000	22.58904	43.49754	40.67000	25.71247	18.28512	31.008790
[6,]	31.07074	27.76425	57.73487	47.14028	22.58904	0.00000	52.16279	57.39376	23.26573	18.67626	46.444070
[7,]	51.48620	56.53597	54.45374	57.50246	43.49754	52.16279	0.00000	40.70540	49.55214	48.67063	45.719150
[8,]	63.24965	69.47518	39.52442	42.38875	40.67000	57.39376	40.70540	0.00000	43.00324	46.12542	15.839223
[9,]	33.17921	37.63531	36.47473	29.59338	25.71247	23.26573	49.55214	43.00324	0.00000	13.34834	30.194290
[10,]	25.38435	27.98753	47.47500	28.62276	18.28512	18.67626	48.67063	46.12542	13.34834	0.00000	33.442059
[11,]	53.61217	59.31525	35.81918	30.56109	31.00879	46.44407	45.71915	15.83922	30.19429	33.44206	0.000000
[12,]	39.87673	48.82935	21.04279	23.00235	40.31248	42.48919	49.01121	38.21058	20.54301	28.81216	28.591651
[13,]	29.51941	39.31054	29.84223	27.96591	39.73091	37.82427	44.39472	45.53292	21.39917	26.49875	37.161092
[14,]	51.54321	56.01655	29.60984	47.14121	37.25071	36.22245	45.55953	39.13160	23.76518	34.64779	32.870467
[15,]	59.48205	62.98233	35.57287	55.48836	41.61513	40.84223	48.83189	42.01700	31.74365	41.88321	37.613519
[16,]	59.65716	64.93189	61.34823	41.01327	42.51726	62.89295	46.68495	30.71192	53.67968	48.99879	32.790144
[17,]	62.98522	68.20483	59.86417	39.11678	67.34998	66.98204	94.06408	73.84047	51.13414	53.65182	59.855308
[18,]	52.38393	56.01120	44.85783	34.19970	23.09088	41.23240	46.88377	22.27365	29.90698	30.17712	11.007831
[19,]	46.10638	51.28469	37.46877	20.67733	30.74481	39.81449	57.30302	34.90198	21.05604	24.59085	19.257623
[20,]	62.44051	72.63348	26.63085	41.17312	59.41630	68.61267	46.79356	32.63146	49.28544	54.98109	36.457404

2.2. Distancia Minkowski:

Distancia Minkowski:

La distancia de Minkowski con parametro $q = 1, 2, ...$ entre los individuos i y j respecto de las variables cuantitativas $X_1, ..., X_k$ es:

$$\delta_q(i,j)_{Minkowski} = \left(\sum_{k=1}^{p} \mid x_{ik} - x_{jk} \mid ^{q} \right)^{(1/q)}$$

2.2.1. Inconvenientes:

1) **Presupone** que las **variables** son **incorreladas** y con **varianza unidad**.

2) **No** es **invariante frente a cambios de escala** (cambios de unidades de medida) de las variables.

3) Es dificilmente euclidianizable (veremos mas adelante que significa esto).

2.2.2. Casos particulares de la distancia de Minkowski:

Distancia Euclidea:

$$\delta_2(i,j)_{Minkowski} = \delta(i,j)_{Euclidea} \qquad (q = 2)$$

Distancia Manhattan:

Distancia Manhattan:

$$\delta_1(i,j)_{Minkowski} = \sum_{k=1}^{p} \mid x_{ik} - x_{jk} \mid \qquad (q = 1)$$

Distancia Dominante:

Distancia Dominante:

$$\delta_{\infty}(i, j)_{Minkowski} = max\{ \mid x_{i1} - x_{j1} \mid , ..., \mid x_{ip} - x_{jp} \mid \} \qquad (q \rightarrow \infty)$$

2.3. Aplicación en R: Distancia de Minkowski:

Programamos la distancia de Minkowski:

```
Dist_Minkowski <- function(i,j, q , Matriz_Datos_Cuantitativos ){

Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_Cuantitativos)

Dist_Minkowski = ( sum( ( abs(Matriz_Datos_Cuantitativos[i,] - Matriz_Datos_Cuantitativos[j,]) )^q ) )^(1/q)

return(Dist_Minkowski)
}
```

Programamos la matriz de distancias de Minkowski:

```
Matriz_Dist_Minkowski <- function(q , Matriz_Datos_Cuantitativos ){

  Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_Cuantitativos)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Cuantitativos)[1] , nrow=dim(Matriz_Datos_Cuantitativos)[1] )

  for(i in 1:dim(Matriz_Datos_Cuantitativos)[1] ){
    for(j in 1:dim(Matriz_Datos_Cuantitativos)[1]){

  M[i,j]=Dist_Minkowski(i,j, q , Matriz_Datos_Cuantitativos)

   }
  }
 return(M)
}
```

Distancia euclidea: (q=2)

```r
Datos_Cuantitativos<- as.matrix(Datos_Cuantitativos)

Dist_Minkowski(1,2, q=2, Datos_Cuantitativos)
```

```
[1] 11.84533
```

```r
Matriz_Dist_Minkowski(q=2 , Datos_Cuantitativos)
```

```
          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]    [,10]     [,11]
 [1,]  0.00000 11.84533 59.05015 38.50322 37.86356 31.07074 51.48620 63.24965 33.17921 25.38435 53.612169
 [2,] 11.84533  0.00000 67.94452 46.62932 37.59997 27.76425 56.53597 69.47518 37.63531 27.98753 59.315255
 [3,] 59.05015 67.94452  0.00000 41.10073 55.37754 57.73487 54.45374 39.52442 36.47473 47.47500 35.819179
 [4,] 38.50322 46.62932 41.10073  0.00000 39.43669 47.14028 57.50246 42.38875 29.59338 28.62276 30.561091
 [5,] 37.86356 37.59997 55.37754 39.43669  0.00000 22.58904 43.49754 40.67000 25.71247 18.28512 31.008790
 [6,] 31.07074 27.76425 57.73487 47.14028 22.58904  0.00000 52.16279 57.39376 23.26573 18.67626 46.444070
 [7,] 51.48620 56.53597 54.45374 57.50246 43.49754 52.16279  0.00000 40.70540 49.55214 48.67063 45.719150
 [8,] 63.24965 69.47518 39.52442 42.38875 40.67000 57.39376 40.70540  0.00000 43.00324 46.12542 15.839223
 [9,] 33.17921 37.63531 36.47473 29.59338 25.71247 23.26573 49.55214 43.00324  0.00000 13.34834 30.194290
[10,] 25.38435 27.98753 47.47500 28.62276 18.28512 18.67626 48.67063 46.12542 13.34834  0.00000 33.442059
[11,] 53.61217 59.31525 35.81918 30.56109 31.00879 46.44407 45.71915 15.83922 30.19429 33.44206  0.000000
[12,] 39.87673 48.82935 21.04279 23.00235 40.31248 42.48919 49.01121 38.21058 20.54301 28.81216 28.591651
[13,] 29.51941 39.31054 29.84223 27.96591 39.73091 37.82427 44.39472 45.53292 21.39917 26.49875 37.161092
[14,] 51.54321 56.01655 29.60984 47.14121 37.25071 36.22245 45.55953 39.13160 23.76518 34.64779 32.870467
[15,] 59.48205 62.98233 35.57287 55.48836 41.61513 40.84223 48.83189 42.01700 31.74365 41.88321 37.613519
[16,] 59.65716 64.93189 61.34823 41.01327 42.51726 62.89295 46.68495 30.71192 53.67968 48.99879 32.790144
[17,] 62.98522 68.20483 59.86417 39.11678 67.34998 66.98204 94.06408 73.84047 51.13414 53.65182 59.855308
[18,] 52.38393 56.01120 44.85783 34.19970 23.09088 41.23240 46.88377 22.27365 29.90698 30.17712 11.007831
[19,] 46.10638 51.28469 37.46877 20.67733 30.74481 39.81449 57.30302 34.90198 21.05604 24.59085 19.257623
[20,] 62.44051 72.63348 26.63085 41.17312 59.41630 68.61267 46.79356 32.63146 49.28544 54.98109 36.457404
```

Distancia Manhattan: (q=1)

```r
Dist_Minkowski(1,2, q=1, Datos_Cuantitativos)
```

```
[1] 22.76648
```

```r
Matriz_Dist_Minkowski(q=1 , Datos_Cuantitativos)
```

```
           [,1]      [,2]      [,3]     [,4]      [,5]      [,6]      [,7]      [,8]     [,9]    [,10]    [,11]
 [1,]   0.00000  22.76648  88.83738 66.25414  74.72206  58.97984  95.92005 107.90888 52.19619 47.45181 95.97649
 [2,]  22.76648   0.00000 111.60386 78.11326  73.61192  45.72249 111.11986 121.97332 51.08606 46.34167 94.86635
 [3,]  88.83738 111.60386   0.00000 72.14136  99.87030 111.50327  84.84312  76.08311 65.66282 82.50324 70.39691
 [4,]  66.25414  78.11326  72.14136  0.00000  66.37970  78.01267  89.59338  71.21467 52.75921 49.01264 49.51028
 [5,]  74.72206  73.61192  99.87030 66.37970   0.00000  30.71644  61.61595  72.46942 48.35155 28.11737 58.08793
 [6,]  58.97984  45.72249 111.50327 78.01267  30.71644   0.00000  88.77748  99.63095 45.84046 29.67373 69.92505
 [7,]  95.92005 111.11986  84.84312 89.59338  61.61595  88.77748   0.00000  80.19944 88.01778 77.17065 89.30312
 [8,] 107.90888 121.97332  76.08311 71.21467  72.46942  99.63095  80.19944   0.00000 75.79830 88.02411 29.70590
 [9,]  52.19619  51.08606  65.66282 52.75921  48.35155  45.84046  88.01778  75.79830  0.00000 20.23418 46.09240
[10,]  47.45181  46.34167  82.50324 49.01264  28.11737  29.67373  77.17065  88.02411 20.23418  0.00000 58.31822
[11,]  95.97649  94.86635  70.39691 49.51028  58.08793  69.92505  89.30312  29.70590 46.09240 58.31822  0.00000
[12,]  61.24623  84.01271  32.52021 44.11385  72.27916  83.91213  75.57448  62.82798 38.07168 54.91210 50.89559
[13,]  47.01147  69.77795  43.04675 50.59938  67.75211  69.67737  67.19673  83.34416 32.06783 40.67734 71.41177
[14,]  72.85156  82.33285  55.70880 84.07975  63.78190  59.99048  87.94567  56.37491 41.63416 57.25373 50.09742
[15,]  85.66941  90.36907  61.66432 95.97202  64.34162  68.02669  92.02138  60.45062 53.52643 69.14600 61.98969
[16,]  92.83920  99.67147 117.25872 65.21984  61.23589  89.91824  83.27879  45.50357 90.39047 77.63771 57.40314
[17,] 122.53213 128.98867 105.42092 67.32743  93.14710 106.81417 145.87138 115.54366 78.13660 87.49560 85.83776
[18,] 101.31849 100.20835  85.08087 54.85228  37.57468  58.51911  85.53781  41.11184 49.12229 53.86668 20.51325
[19,]  77.65102  84.10755  58.22622 38.95832  54.28464  65.91762 100.99026  64.19583 33.25548 42.61448 34.48993
[20,] 111.30490 126.50471  50.30227 75.74606 110.90497 122.53794  84.25513  48.92239 80.32969 93.53791 52.81704
```

2.3.1. Aplicación en R: Distancia Dominante

Programamos la distancia Dominante:

```r
Dist_Dominante <- function(i,j,   Matriz_Datos_Cuantitativos){

Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_Cuantitativos)

Dist_Dominante =  max( abs(Matriz_Datos_Cuantitativos[i,] - Matriz_Datos_Cuantitativos[j,]) )

return(Dist_Dominante)
}
```

```r
Dist_Dominante(1,2, Datos_Cuantitativos)
```

```
[1] 8.154973
```

Programamos la matriz de distancias dominantes:

```r
Matriz_Dist_Dominante <- function( Matriz_Datos_Cuantitativos ){

  Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_Cuantitativos)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Cuantitativos)[1] , nrow=dim(Matriz_Datos_Cuantitativos)[1] )

  for(i in 1:dim(Matriz_Datos_Cuantitativos)[1] ){
    for(j in 1:dim(Matriz_Datos_Cuantitativos)[1]){

  M[i,j]=Dist_Dominante(i,j,  Matriz_Datos_Cuantitativos)

    }
  }
 return(M)
}
```

```r
Matriz_Dist_Dominante(Datos_Cuantitativos)
```

```
           [,1]      [,2]     [,3]     [,4]      [,5]     [,6]     [,7]     [,8]     [,9]     [,10]     [,11]
 [1,]  0.000000  8.154973 53.71146 30.89058 22.668510 19.19193 40.15157 41.83181 30.30941 17.820125 36.861949
 [2,]  8.154973  0.000000 58.46603 36.96419 22.151127 22.91154 36.36823 47.90542 35.06397 22.574689 41.616513
 [3,] 53.711462 58.466026  0.00000 33.69555 36.314898 35.55449 41.26421 26.63698 24.27724 35.891337 23.121184
 [4,] 30.890580 36.964187 33.69555  0.00000 32.562473 37.61729 50.04553 35.41830 23.51461 21.817736 24.900271
 [5,] 22.668510 22.151127 36.31490 32.56247  0.000000 21.17471 39.68706 27.38381 18.76818 16.460876 20.205797
 [6,] 19.191930 22.911535 35.55449 37.61729 21.174708  0.00000 38.67001 48.55852 14.10267 15.799551 41.380505
 [7,] 40.151569 36.368234 41.26421 50.04553 39.687057 38.67001  0.00000 24.31435 36.25124 33.943935 28.932511
 [8,] 41.831811 47.905418 26.63698 35.41830 27.383809 48.55852 24.31435  0.00000 34.45585 32.758967 10.518034
 [9,] 30.309410 35.063974 24.27724 23.51461 18.768180 14.10267 36.25124 34.45585  0.00000 12.489285 27.277833
[10,] 17.820125 22.574689 35.89134 21.81774 16.460876 15.79955 33.94393 32.75897 12.48928  0.000000 25.580955
[11,] 36.861949 41.616513 23.12118 24.90027 20.205797 41.38051 28.93251 10.51803 27.27783 25.580955  0.000000
[12,] 35.063391 39.817955 18.64807 15.04748 26.245678 24.82688 43.72874 29.10151 15.11599 18.856699 18.583476
[13,] 26.193953 30.948517 27.51751 21.29644 29.709297 28.69225 41.87463 32.23767 20.11081 23.851517 25.059655
[14,] 48.352268 53.106832 18.11487 28.33635 30.955704 30.19530 29.57933 37.19401 18.04286 30.532143 30.015997
[15,] 54.408230 59.162793 24.41397 34.39232 37.011666 36.25126 35.63529 39.12204 24.09882 36.588105 31.944026
[16,] 53.432666 59.506273 41.84906 33.25433 38.984664 60.15937 35.91520 28.12265 46.05670 44.359822 24.999550
[17,] 42.192297 45.975633 41.07966 32.29833 64.860807 57.09653 82.34387 67.71664 46.09263 48.399932 57.198605
[18,] 32.190968 37.812760 31.49515 29.45393 17.291152 38.46586 37.30648 17.26072 24.36319 22.666309  8.373969
[19,] 33.447547 38.202111 30.86640 13.43163 23.005524 33.39351 40.48858 25.86136 19.29084 17.593964 15.343322
[20,] 40.906728 45.661292 19.09769 30.69121 45.446507 46.17093 34.43514 25.80521 35.84801 39.588727 34.691961
```

2.4. Distancia de Canberra:

Distancia Canberra:

La distancia de Canberra entre los elementos i y j respecto de las variables cuantitativas $X_1, ..., X_p$ es:

$$\delta(i,j)_{Canberra} = \sum_{k=1}^{p} \frac{\mid x_{ik} - x_{jk} \mid}{\mid x_{ik} \mid + \mid x_{jk} \mid}$$

2.4.1. Inconvenientes:

1) **Presupone** que las **variables** son **incorreladas** y con **varianza unidad**.

Aunque **si** es **invariante frente a cambios de escala** (cambios de unidades de medida) de las variables.

2.4.2. Aplicación en R: Distancia Canberra

Programamos la distancia de Canberra:

```r
Dist_Canberra <- function(i,j,   Matriz_Datos_Cuantitativos){

Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_Cuantitativos)

Dist_Canberra =   sum( abs(Matriz_Datos_Cuantitativos[i,] - Matriz_Datos_Cuantitativos[j,])/(abs(Matriz_Datos_Cuantitativos[i,])+ abs(Matriz_Datos_Cuantitativos[j,])) )

return(Dist_Canberra)
}
```

```r
Dist_Canberra(1,2, Datos_Cuantitativos)
```

```
[1] 1.59386
```

Programamos la matriz de distancias de Canberra:

```r
Matriz_Dist_Canberra <- function( Matriz_Datos_Cuantitativos ) {

  Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_Cuantitativos)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Cuantitativos)[1] , nrow=dim(Matriz_Datos_Cuantitativos)[1] )

  for(i in 1:dim(Matriz_Datos_Cuantitativos)[1] ){
    for(j in 1:dim(Matriz_Datos_Cuantitativos)[1]){

  M[i,j]=Dist_Canberra(i,j,  Matriz_Datos_Cuantitativos)

    }
  }
 return(M)
}
```

```
Matriz_Dist_Canberra(Datos_Cuantitativos)
```

```
          [,1]     [,2]      [,3]     [,4]     [,5]     [,6]     [,7]      [,8]     [,9]    [,10]     [,11]
 [1,] 0.000000 1.593860 3.1115089 2.885963 3.674402 2.955515 4.000000 3.0549842 3.136161 3.595288 3.4921304
 [2,] 1.593860 0.000000 3.7903532 3.028892 3.316206 2.319931 4.000000 3.4105796 2.083884 3.192054 3.0474842
 [3,] 3.111509 3.790353 0.0000000 2.594523 3.300624 3.942318 2.290583 2.6984357 3.293953 3.816940 2.6679343
 [4,] 2.885963 3.028892 2.5945232 0.000000 2.781984 2.860005 2.683301 2.2755692 2.763941 2.815378 1.9793718
 [5,] 3.674402 3.316206 3.3006240 2.781984 0.000000 1.828166 2.196915 2.3720506 3.188237 2.088232 2.1215650
 [6,] 2.955515 2.319931 3.9423178 2.860005 1.828166 0.000000 2.767181 2.8767838 2.968057 2.005038 2.1541830
 [7,] 4.000000 4.000000 2.2905832 2.683301 2.196915 2.767181 0.000000 2.6834021 3.771939 3.111323 2.8619071
 [8,] 3.054984 3.410580 2.6984357 2.275569 2.372051 2.876784 2.683402 0.0000000 2.747315 3.285598 0.9737895
 [9,] 3.136161 2.083884 3.2939528 2.763941 3.188237 2.968057 3.771939 2.7473155 0.000000 3.000346 2.2371240
[10,] 3.595288 3.192054 3.8169405 2.815378 2.088232 2.005038 3.111323 3.2855984 3.000346 0.000000 2.7168476
[11,] 3.492130 3.047484 2.6679343 1.979372 2.121565 2.154183 2.861907 0.9737895 2.237124 2.716848 0.0000000
[12,] 3.145472 3.794390 1.6243988 2.322525 3.159238 3.855438 2.857090 2.5356206 2.980539 3.709491 2.4505601
[13,] 3.162731 3.809159 1.7504261 2.790031 3.554885 3.786907 2.934949 3.3473126 3.024813 3.065013 3.2644812
[14,] 2.625461 2.860043 3.0788973 2.971385 2.691750 2.156437 3.376076 1.5813686 2.382823 3.131761 1.6100279
[15,] 2.549729 2.572857 3.0094180 2.863642 2.424428 2.102463 3.221387 1.5956402 2.440654 3.098592 1.7607312
[16,] 2.832654 2.794495 3.5752839 2.396703 2.243191 2.740605 2.964051 1.4501666 3.261769 3.147265 1.7699373
[17,] 3.532624 3.342450 2.3923378 1.794794 2.480112 2.941317 3.083831 2.2275800 2.314137 2.989182 1.6966413
[18,] 3.646303 3.270660 2.7337946 2.066649 1.678449 2.076762 2.633509 1.1588054 2.255311 2.701663 0.5951859
[19,] 2.674676 3.017946 1.6629353 1.660176 2.516298 2.918570 3.139241 2.0609765 2.080537 2.884732 1.4843861
[20,] 4.000000 4.000000 2.0474725 2.938142 3.505675 3.694274 2.500727 1.9416935 3.285807 3.041210 1.9590059
```

2.5. Distancia de Karl Pearson:

Distancia de Pearson:

La distancia de Karl Pearson entre los elementos i y j respecto de las variables cuantitativas $X_1, ..., X_p$ es:

$$\delta^2(i,j)_{Pearson} = \sum_{k=1}^{p} \frac{(x_{ik} - x_{jk})^2}{s_k^2} = (x_i - x_j)^t \cdot S_0^{-1} \cdot (x_i - x_j)$$

$$\delta(i,j)_{Pearson} = \sqrt{\sum_{k=1}^{p} \frac{(x_{ik} - x_{jk})^2}{s_k^2}} = \sqrt{(x_i - x_j)^t \cdot S_0^{-1} \cdot (x_i - x_j)}$$

Donde:
$S_0 = diag(s_1^2, ..., s_p^2)$
s_k^2 es la varianza de X_k

Observación:

Con la distancia de Karl Pearson el peso que se atribuye a la diferencia entre individuos respecto de una variable es mayor cuanto menor sea la dispersion de dicha variable, y viceversa.

2.5.1. Inconvenientes

1) **Presupone** que las **variables** son **incorreladas** y con **varianza unidad**.

Aunque **si** es **invariante frente a cambios de escala** (cambios de unidades de medida) de las variables.

2.5.2. Aplicación en R: Distancia de Pearson

Programamos la distancia de Pearson:

```r
Dist_Pearson <- function(i,j,   Matriz_Datos_Cuantitativos){

Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_Cuantitativos)

Dist_Pearson  =   sum( ((Matriz_Datos_Cuantitativos[i,] - Matriz_Datos_Cuantitativos[j,])^2)/diag(cov(Matriz_Datos_Cuantitativos)) )

return(sqrt(Dist_Pearson) )
}
```

```r
Dist_Pearson(1,2, Datos_Cuantitativos)
```

```
[1] 0.7665315
```

```r
Matriz_Dist_Pearson <- function( Matriz_Datos_Cuantitativos ){

  Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_Cuantitativos)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Cuantitativos)[1] , nrow=dim(Matriz_Datos_Cuantitativos)[1] )

  for(i in 1:dim(Matriz_Datos_Cuantitativos)[1] ){
    for(j in 1:dim(Matriz_Datos_Cuantitativos)[1]){

  M[i,j]=Dist_Pearson(i,j,  Matriz_Datos_Cuantitativos)

    }
  }
 return(M)
}
```

```
Matriz_Dist_Pearson(Datos_Cuantitativos)
```

```
           [,1]      [,2]      [,3]     [,4]      [,5]     [,6]     [,7]     [,8]      [,9]     [,10]     [,11]
 [1,] 0.0000000 0.7665315 4.1340453 2.587439 2.4759983 2.046861 3.356715 4.289427 2.3171951 1.7050413 3.6557673
 [2,] 0.7665315 0.0000000 4.7135132 3.138238 2.5194692 1.905826 3.691722 4.720045 2.6496115 1.9346491 4.0612662
 [3,] 4.1340453 4.7135132 0.0000000 2.816488 3.6794116 3.841170 3.662471 2.585239 2.4183247 3.2052753 2.3328535
 [4,] 2.5874392 3.1382383 2.8164877 0.000000 2.5399572 3.074064 3.684258 2.800882 1.9578628 1.8641488 2.0336230
 [5,] 2.4759983 2.5194692 3.6794116 2.539957 0.0000000 1.485892 2.706140 2.709763 1.6959644 1.1766895 2.0915832
 [6,] 2.0468613 1.9058262 3.8411704 3.074064 1.4858920 0.000000 3.302696 3.801788 1.5477077 1.2207653 3.0946160
 [7,] 3.3567152 3.6917223 3.6624705 3.684258 2.7061398 3.302696 0.000000 2.695445 3.1680046 3.0770959 2.9580385
 [8,] 4.2894271 4.7200452 2.5852385 2.800882 2.7097627 3.801788 2.695445 0.000000 2.8224770 3.0701196 1.0171158
 [9,] 2.3171951 2.6496115 2.4183247 1.957863 1.6959644 1.547708 3.168005 2.822477 0.0000000 0.9359033 1.9928768
[10,] 1.7050413 1.9346491 3.2052753 1.864149 1.1766895 1.220765 3.077096 3.070120 0.9359033 0.0000000 2.2563973
[11,] 3.6557673 4.0612662 2.3328535 2.033623 2.0915832 3.094616 2.958038 1.017116 1.9928768 2.2563973 0.0000000
[12,] 2.7867500 3.3744198 1.4572629 1.548948 2.6129697 2.772253 3.186149 2.488662 1.3105649 1.8998578 1.8405082
[13,] 2.0548678 2.6869638 2.0965568 1.815286 2.5162452 2.402426 2.868040 2.992665 1.3339621 1.6686987 2.4165236
[14,] 3.6245343 3.9470909 1.9037220 3.176325 2.5572597 2.490394 3.044949 2.587081 1.6186860 2.4046784 2.1835779
[15,] 4.1652453 4.4309943 2.2800319 3.743496 2.8936608 2.838937 3.298389 2.795013 2.1688127 2.9102550 2.5193636
[16,] 3.9326942 4.2954881 4.1610051 2.674489 2.7830991 4.134922 3.026202 2.156325 3.5606477 3.2181157 2.2554527
[17,] 4.1615643 4.5420991 3.8562159 2.525047 4.3666763 4.360422 6.026722 4.755582 3.2989014 3.4876801 3.8572228
[18,] 3.5240518 3.8086887 2.9088266 2.223643 1.5709455 2.743698 2.975889 1.425848 1.9540596 2.0163476 0.7075314
[19,] 3.1419491 3.5237481 2.4249575 1.376430 2.0462608 2.648189 3.660258 2.247800 1.3814848 1.6700737 1.2427113
[20,] 4.2169611 4.8793074 1.7659945 2.669354 3.8176460 4.445071 3.102504 2.049031 3.1543881 3.5604494 2.2642817
```

2.6. Distancia de Mahalanobis:

Distancia de Mahalanobis:

La distancia de Mahalanobis entre los elementos i y j respecto de las variables cuantitativas $X_1, ..., X_p$ es:

$$\delta^2(i, j)_{Maha} = (x_i - x_j)^t \cdot S^{-1} \cdot (x_i - x_j)$$

$$\delta(i, j)_{Maha} = \sqrt{(x_i - x_j)^t \cdot S^{-1} \cdot (x_i - x_j)}$$

Donde:

S es la matriz de covarianzas de la matriz de datos $X = (X_1, ..., X_p)$

2.6.1. Ventajas:

La distancia de Mahalanobis es adecuada como medida de descrepancia entre datos por las siguientes razones:

1) Es **invariante frente a transformaciones lineales** de las variables

2) **Tiene en cuenta las correlaciones entre las variables**. Por ejemplo, no aumenta por el hecho de aumentar el numero de variables observadas, sino que solo aumentará cuando las nuevas variables no sean redundantes respecto de la información aportada por las anteriores.

Observación:

1) La distancia euclidea coincide con la de Mahalanobis cuando $S = I$

2) La distancia de Karl Pearson coincide con la de Mahalanobis cuando $S = diag(s_1^2, ..., s_p^2)$

2.6.2. Aplicación en R: Distancia de Mahalanobis

Programamos la distancia de Mahalanobis:

```
Dist_Mahalanobis <- function(i,j,   Matriz_Datos_Cuantitativos){

Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_Cuantitativos)

Dist_Mahalanobis  =    t( Matriz_Datos_Cuantitativos[i,] - Matriz_Datos_Cuantitativos[j,] )  %*% solve(cov(Matriz_Datos_Cuantitativos)) %*% ( Matriz_Datos_Cuantitativos[i,] - Matriz_Datos_Cuantitativos[j,] )

Dist_Mahalanobis <- sqrt(Dist_Mahalanobis)

return( Dist_Mahalanobis )
}
```

```
Dist_Mahalanobis(1,2, Datos_Cuantitativos)
```

```
          [,1]
[1,] 0.7384377
```

Programamos la matriz de distancias de Mahalanobis:

```
Matriz_Dist_Mahalanobis <- function( Matriz_Datos_Cuantitativos ){

  Matriz_Datos_Cuantitativos=as.matrix(Matriz_Datos_Cuantitativos)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Cuantitativos)[1] , nrow=dim(Matriz_Datos_Cuantitativos)[1] )

  for(i in 1:dim(Matriz_Datos_Cuantitativos)[1] ){
    for(j in 1:dim(Matriz_Datos_Cuantitativos)[1]){

  M[i,j]=Dist_Mahalanobis(i,j,  Matriz_Datos_Cuantitativos)

    }
  }
 return(M)
}
```

```r
Matriz_Dist_Mahalanobis(Datos_Cuantitativos)
```

```
           [,1]      [,2]      [,3]     [,4]      [,5]     [,6]      [,7]      [,8]      [,9]     [,10]     [,11]
 [1,] 0.0000000 0.7384377 3.9788574 2.444996 2.5692943 2.205804 3.2640580 4.1150559 2.3138245 1.7166677 3.5237623
 [2,] 0.7384377 0.0000000 4.4920546 2.941129 2.5206119 2.015150 3.5259949 4.4561551 2.5609024 1.8548342 3.8386520
 [3,] 3.9788574 4.4920546 0.0000000 3.007805 3.6555102 3.642408 3.4906740 2.6830491 2.3215174 3.1234319 2.4672874
 [4,] 2.4449958 2.9411288 3.0078051 0.000000 2.6208769 3.143340 3.6512263 2.8323810 2.1325193 1.8961685 2.1342843
 [5,] 2.5692943 2.5206119 3.6555102 2.620877 0.0000000 1.498364 2.6181668 2.5103723 1.7286650 1.2089587 1.9445687
 [6,] 2.2058040 2.0151496 3.6424076 3.143340 1.4983639 0.000000 3.0950975 3.5511155 1.4654074 1.2585501 2.9109903
 [7,] 3.2640580 3.5259949 3.4906740 3.651226 2.6181668 3.095097 0.0000000 2.5718018 2.9796743 2.9201241 2.8268707
 [8,] 4.1150559 4.4561551 2.6830491 2.832381 2.5103723 3.551116 2.5718018 0.0000000 2.6836014 2.8760331 0.9569153
 [9,] 2.3138245 2.5609024 2.3215174 2.132519 1.7286650 1.465407 2.9796743 2.6836014 0.0000000 0.9597218 1.9323146
[10,] 1.7166677 1.8548342 3.1234319 1.896168 1.2089587 1.258550 2.9201241 2.8760331 0.9597218 0.0000000 2.1141826
[11,] 3.5237623 3.8386520 2.4672874 2.134284 1.9445687 2.910990 2.8268707 0.9569153 1.9323146 2.1141826 0.0000000
[12,] 2.6356384 3.1644191 1.4644753 1.708877 2.6334132 2.670792 3.0387803 2.4877365 1.2904842 1.8428050 1.9102473
[13,] 1.9781281 2.5529312 2.0205466 1.952399 2.6254593 2.398479 2.7758344 2.9811526 1.3772200 1.7054755 2.4654917
[14,] 3.7361456 3.9659210 1.9160149 3.532411 2.6866535 2.409838 2.9593610 2.6666633 1.7154982 2.5380309 2.3668902
[15,] 4.3431665 4.5164398 2.3771749 4.158321 3.0739307 2.810234 3.2793781 2.9540955 2.3263366 3.1032584 2.7745376
[16,] 3.8613532 4.1401385 4.4025231 2.645047 2.8093890 4.158190 3.2208554 2.3209227 3.6569616 3.2017312 2.3673278
[17,] 3.9236804 4.2783884 3.9117227 2.386881 4.3080899 4.333959 5.8130906 4.6433198 3.3290392 3.4035894 3.8067194
[18,] 3.4799704 3.6658304 3.0410859 2.367370 1.4574909 2.627328 2.8895462 1.3420813 1.9777208 1.9545562 0.7011910
[19,] 3.0106374 3.3130806 2.5215480 1.483720 1.9527302 2.537933 3.4881144 2.1307603 1.3975836 1.5596548 1.1873742
[20,] 3.9171206 4.5302385 1.8009551 2.603486 3.7072797 4.213938 2.9733733 2.1546965 2.9888824 3.3538678 2.3021805
```

3. DISTANCIAS CON VARIABLES CATEGORICAS:

3.1. Similaridad:

Se trata de un concepto dual al de distancia que expresa la proximidad o semejanza entre dos elementos.

Similaridad:

Dado un conjunto de elementos ε , se denomina similaridad a toda aplicacion
$s : \varepsilon x \varepsilon \rightarrow \mathbb{R}$ tal que:
1) $0 \leq s_{ij} \leq 1$
2) $s_{ii} = 1$
3) $s_{ij} = s_{ji}$

3.2. Matriz de Similaridades:

Matriz de Similaridad:

$$\mathcal{S} = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ \dots & \dots & \dots & \dots \\ s_{n1} & \delta_{n2} & \dots & s_{nn} \end{pmatrix}$$

Con $s_{ii} = 1$ y $s_{ij} = s_{ji}$

3.3. Pasar de una similaridad a una distancia:

Pasar de una similaridad a una distancia:

Las siguientes transofrmaciones permiten pasar de una medida de similaridad a una distancia:

1)

$$\delta_{ij} = 1 - s_{ij}$$

2)

$$\delta_{ij} = \sqrt{1 - s_{ij}}$$

3) Transformación de Gower:

$$\delta_{ij}^2 = s_{ii} + s_{jj} - 2 \cdot s_{ij}$$

3.4. Similaridades con variables categoricas binarias:

Sean $X_1, ..., X_p$ variables categoricas **binarias** ($X_j \in \{0, 1\}$)

Los principales coeficientes de similaridad entre dos individuos/elementos respecto de variables binarias se suelen calcular a partir de los siguientes parametros:

$a_{ij} =$ nº de variables con respuesta 1 en ambos elementos i y j

$b_{ij} =$ nº de variables con respuesta 0 en el elemento i y respuesta 1 en el j

$c_{ij} =$ nº de variables con respuesta 1 en el elemento i y respuesta 0 en el j

$d_{ij} =$ nº de variables con respuesta 0 en ambos elementos i y j

Observación:

$a_{ij} + b_{ij} + c_{ij} + d_{ij} = p$

3.4.1. Matrices con los parametros a, b, c y d:

Dada una matriz de datos $X = (X_1, ..., X_p)$ de tamaño nxp de variables categoricas binarias, entonces:

Matrices con los parametros :

$$a = X \cdot X^t$$

$$b = (\overrightarrow{1}_{nxp} - X) \cdot X^t$$

$$c = b^t$$

$$d = (\overrightarrow{1}_{nxp} - X) \cdot (\overrightarrow{1}_{nxp} - X)^t$$

Son las matrices que contienen a los parametros a_{ij} , b_{ij} , c_{ij} y d_{ij} , respectivamente.

3.4.2. Aplicación en R:

```
Datos_Binarios <- Datos_Mixtos%>% select(5:7)
```

A tibble: 50 x 3

X5 <dbl>	X6 <dbl>	X7 <dbl>
0	1	0
1	1	1
1	0	0
1	1	1
1	1	0
1	1	1
1	0	0
0	0	0
0	0	0
1	0	0
0	1	0
1	0	1
0	1	0
1	1	1
0	1	1

1-15 of 50 rows Previous 1 2 Next

Obtencion de las matrices con los parametros a, b, c y d:

```
Datos_Binarios <- as.matrix(Datos_Binarios)

a = Datos_Binarios %*% t(Datos_Binarios)

unos<- rep(1, dim(Datos_Binarios)[2])

Matriz_Unos<- matrix( rep(unos, dim(Datos_Binarios)[1]), ncol=
   dim(Datos_Binarios)[2]) #Matriz de unos de tamano nxp

b = (Matriz_Unos-Datos_Binarios)%*%t(Datos_Binarios)

c = t(b)

d = (Matriz_Unos - Datos_Binarios)%*%t(Matriz_Unos - Datos_
   Binarios)
```

3.4.3. Coeficiente de Sokal: (Simple matching coefficient)

Coeficiente de Sokal:

El coeficiente de similaridad de Sokal entre los elementos/individuos i y j respecto de las variables binarias $X_1, ..., X_p$ es:

$$S(i, j)_{Sokal} = \frac{a_{ij} + d_{ij}}{a_{ij} + b_{ij} + c_{ij} + d_{ij}} = \frac{a_{ij} + d_{ij}}{p}$$

Distancia de Sokal:

Distancia de Sokal:

Obtenemos la distancia de Sokal como:

$$\delta(i, j)_{Sokal} = \sqrt{S(i, i)_{Sokal} + S(j, j)_{Sokal} - 2 \cdot S(i, j)_{Sokal}}$$

Aplicación en R: Coeficiente de Similaridad de Sokal:

Programamos el coeficiente de similaridad de Sokal:

```r
Similaridad_Sokal <- function(i,j,   Matriz_Datos_Binarios){

  Matriz_Datos_Binarios=as.matrix(Matriz_Datos_Binarios)


  a= Matriz_Datos_Binarios %*% t(Matriz_Datos_Binarios)

  unos<- rep(1, dim(Matriz_Datos_Binarios)[2])

  Matriz_Unos<- matrix( rep(unos, dim(Matriz_Datos_Binarios)[1]), ncol=dim(Matriz_Datos_Binarios)[2])
  #Matriz de unos de tamano nxp

  b=(Matriz_Unos-Matriz_Datos_Binarios)%*%t(Matriz_Datos_Binarios)

  c= t(b)

  d= (Matriz_Unos - Matriz_Datos_Binarios)%*%t(Matriz_Unos -
      Matriz_Datos_Binarios)


Similaridad_Sokal  = (a[i,j] + d[i,j]) /  dim(Matriz_Datos_Binarios)[2]

return(Similaridad_Sokal)
}
```

```r
Similaridad_Sokal(1, 2, Datos_Binarios)
```

```
[1] 0.3333333
```

```r
Similaridad_Sokal(7,8, Datos_Binarios)
```

```
[1] 0.6666667
```

Programamos la matriz de similaridades de Sokal:

```r
Matriz_Similaridad_Sokal <- function( Matriz_Datos_
    Cuantitativos ){

  Matriz_Datos_Binarios=as.matrix(Matriz_Datos_Binarios)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Cuantitativos)[1] ,
      nrow=dim(Matriz_Datos_Cuantitativos)[1] )

  for(i in 1:dim(Matriz_Datos_Cuantitativos)[1] ){
    for(j in 1:dim(Matriz_Datos_Cuantitativos)[1]){

  M[i,j]=Similaridad_Sokal(i,j,  Matriz_Datos_Cuantitativos)

    }
  }
 return(M)
}
```

```r
Matriz_Similaridad_Sokal(Datos_Binarios)
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]
[1,]	1.0000000	0.3333333	0.3333333	0.3333333	0.6666667	0.3333333	0.3333333	0.6666667	0.6666667	0.3333333	1.0000000
[2,]	0.3333333	1.0000000	0.3333333	1.0000000	0.6666667	1.0000000	0.3333333	0.0000000	0.0000000	0.3333333	0.3333333
[3,]	0.3333333	0.3333333	1.0000000	0.3333333	0.6666667	0.3333333	1.0000000	0.6666667	0.6666667	1.0000000	0.3333333
[4,]	0.3333333	1.0000000	0.3333333	1.0000000	0.6666667	1.0000000	0.3333333	0.0000000	0.0000000	0.3333333	0.3333333
[5,]	0.6666667	0.6666667	0.6666667	0.6666667	1.0000000	0.6666667	0.6666667	0.3333333	0.3333333	0.6666667	0.6666667
[6,]	0.3333333	1.0000000	0.3333333	1.0000000	0.6666667	1.0000000	0.3333333	0.0000000	0.0000000	0.3333333	0.3333333
[7,]	0.3333333	0.3333333	1.0000000	0.3333333	0.6666667	0.3333333	1.0000000	0.6666667	0.6666667	1.0000000	0.3333333
[8,]	0.6666667	0.0000000	0.6666667	0.0000000	0.3333333	0.0000000	0.6666667	1.0000000	1.0000000	0.6666667	0.6666667
[9,]	0.6666667	0.0000000	0.6666667	0.0000000	0.3333333	0.0000000	0.6666667	1.0000000	1.0000000	0.6666667	0.6666667
[10,]	0.3333333	0.3333333	1.0000000	0.3333333	0.6666667	0.3333333	1.0000000	0.6666667	0.6666667	1.0000000	0.3333333
[11,]	1.0000000	0.3333333	0.3333333	0.3333333	0.6666667	0.3333333	0.3333333	0.6666667	0.6666667	0.3333333	1.0000000
[12,]	0.0000000	0.6666667	0.6666667	0.6666667	0.3333333	0.6666667	0.6666667	0.3333333	0.3333333	0.6666667	0.0000000
[13,]	1.0000000	0.3333333	0.3333333	0.3333333	0.6666667	0.3333333	0.3333333	0.6666667	0.6666667	0.3333333	1.0000000
[14,]	0.3333333	1.0000000	0.3333333	1.0000000	0.6666667	1.0000000	0.3333333	0.0000000	0.0000000	0.3333333	0.3333333
[15,]	0.6666667	0.6666667	0.0000000	0.6666667	0.3333333	0.6666667	0.0000000	0.3333333	0.3333333	0.0000000	0.6666667
[16,]	1.0000000	0.3333333	0.3333333	0.3333333	0.6666667	0.3333333	0.3333333	0.6666667	0.6666667	0.3333333	1.0000000
[17,]	1.0000000	0.3333333	0.3333333	0.3333333	0.6666667	0.3333333	0.3333333	0.6666667	0.6666667	0.3333333	1.0000000
[18,]	0.3333333	1.0000000	0.3333333	1.0000000	0.6666667	1.0000000	0.3333333	0.0000000	0.0000000	0.3333333	0.3333333
[19,]	0.6666667	0.0000000	0.6666667	0.0000000	0.3333333	0.0000000	0.6666667	1.0000000	1.0000000	0.6666667	0.6666667
[20,]	0.3333333	0.3333333	0.3333333	0.3333333	0.0000000	0.3333333	0.3333333	0.6666667	0.6666667	0.3333333	0.3333333

En este caso pasamos de similaridad a distancia usando la transformacion:

$$\delta(i,j)_{Sokal} = \sqrt{S(i,i)_{Sokal} + S(j,j)_{Sokal} - 2 \cdot S(i,j)_{Sokal}}$$

```
Dist_Sokal <- function(i,j,   Matriz_Datos_Binarios){

Matriz_Datos_Binarios=as.matrix(Matriz_Datos_Binarios)

Dist_Sokal  = sqrt( Similaridad_Sokal(i,i,   Matriz_Datos_
   Binarios) + Similaridad_Sokal(j,j,   Matriz_Datos_Binarios)
    - 2*Similaridad_Sokal(i,j,   Matriz_Datos_Binarios)  )

return( Dist_Sokal )
}
```

```
Dist_Sokal(1, 2, Matriz_Datos_Binarios = Datos_Binarios)
```

```
[1] 1.154701
```

```
Matriz_Distancias_Sokal <- function( Matriz_Datos_Binarios ){

  Matriz_Datos_Binarios=as.matrix(Matriz_Datos_Binarios)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Binarios)[1] , nrow=dim
     (Matriz_Datos_Binarios)[1] )

  for(i in 1:dim(Matriz_Datos_Binarios)[1] ){
    for(j in 1:dim(Matriz_Datos_Binarios)[1]){

  M[i,j]=Dist_Sokal(i,j,  Matriz_Datos_Binarios)

    }
  }
 return(M)
}
```

```r
Matriz_Distancias_Sokal(Datos_Binarios)
```

```
            [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]     [,10]     [,11]
 [1,] 0.0000000 1.1547005 1.1547005 1.1547005 0.8164966 1.1547005 1.1547005 0.8164966 0.8164966 1.1547005 0.0000000
 [2,] 1.1547005 0.0000000 1.1547005 0.0000000 0.8164966 0.0000000 1.1547005 1.4142136 1.4142136 1.1547005 1.1547005
 [3,] 1.1547005 1.1547005 0.0000000 1.1547005 0.8164966 1.1547005 0.0000000 0.8164966 0.8164966 0.0000000 1.1547005
 [4,] 1.1547005 0.0000000 1.1547005 0.0000000 0.8164966 0.0000000 1.1547005 1.4142136 1.4142136 1.1547005 1.1547005
 [5,] 0.8164966 0.8164966 0.8164966 0.8164966 0.0000000 0.8164966 0.8164966 1.1547005 1.1547005 0.8164966 0.8164966
 [6,] 1.1547005 0.0000000 1.1547005 0.0000000 0.8164966 0.0000000 1.1547005 1.4142136 1.4142136 1.1547005 1.1547005
 [7,] 1.1547005 1.1547005 0.0000000 1.1547005 0.8164966 1.1547005 0.0000000 0.8164966 0.8164966 0.0000000 1.1547005
 [8,] 0.8164966 1.4142136 0.8164966 1.4142136 1.1547005 1.4142136 0.8164966 0.0000000 0.0000000 0.8164966 0.8164966
 [9,] 0.8164966 1.4142136 0.8164966 1.4142136 1.1547005 1.4142136 0.8164966 0.0000000 0.0000000 0.8164966 0.8164966
[10,] 1.1547005 1.1547005 0.0000000 1.1547005 0.8164966 1.1547005 0.0000000 0.8164966 0.8164966 0.0000000 1.1547005
[11,] 0.0000000 1.1547005 1.1547005 1.1547005 0.8164966 1.1547005 1.1547005 0.8164966 0.8164966 1.1547005 0.0000000
[12,] 1.4142136 0.8164966 0.8164966 0.8164966 1.1547005 0.8164966 0.8164966 1.1547005 1.1547005 0.8164966 1.4142136
[13,] 0.0000000 1.1547005 1.1547005 1.1547005 0.8164966 1.1547005 1.1547005 0.8164966 0.8164966 1.1547005 0.0000000
[14,] 1.1547005 0.0000000 1.1547005 0.0000000 0.8164966 0.0000000 1.1547005 1.4142136 1.4142136 1.1547005 1.1547005
[15,] 0.8164966 0.8164966 1.4142136 0.8164966 1.1547005 0.8164966 1.4142136 1.1547005 1.1547005 1.4142136 0.8164966
[16,] 0.0000000 1.1547005 1.1547005 1.1547005 0.8164966 1.1547005 1.1547005 0.8164966 0.8164966 1.1547005 0.0000000
[17,] 0.0000000 1.1547005 1.1547005 1.1547005 0.8164966 1.1547005 1.1547005 0.8164966 0.8164966 1.1547005 0.0000000
[18,] 1.1547005 0.0000000 1.1547005 0.0000000 0.8164966 0.0000000 1.1547005 1.4142136 1.4142136 1.1547005 1.1547005
[19,] 0.8164966 1.4142136 0.8164966 1.4142136 1.1547005 1.4142136 0.8164966 0.0000000 0.0000000 0.8164966 0.8164966
[20,] 1.1547005 1.1547005 1.1547005 1.1547005 1.4142136 1.1547005 1.1547005 0.8164966 0.8164966 1.1547005 1.1547005
```

3.4.4. Coeficiente de Jaccard:

Coeficiente de Jaccard:

El coeficiente de similaridad de Jaccard entre los elementos/individuos i y j respecto de las variables binarias $X_1, ..., X_p$ es:

$$S(i,j)_{Jaccard} = \frac{a_{ij}}{a_{ij} + b_{ij} + c_{ij}}$$

Distancia de Jaccard:

Distancia de Jaccard:

Obtenemos la distancia de Jaccard como:

$$\delta(i,j)_{Jaccard} = \sqrt{S(i,i)_{Jaccard} + S(j,j)_{Jaccard} - 2 \cdot S(i,j)_{Jaccard}}$$

3.4.5. Aplicación en R: Coeficiente de Similaridad de Jaccard:

Programamos el coeficiente de similaridad de Jaccard:

```r
Similaridad_Jaccard<- function(i,j,   Matriz_Datos_Binarios){

  Matriz_Datos_Binarios=as.matrix(Matriz_Datos_Binarios)

  a= Matriz_Datos_Binarios %*% t(Matriz_Datos_Binarios)

  unos<- rep(1, dim(Matriz_Datos_Binarios)[2])

  Matriz_Unos<- matrix( rep(unos, dim(Matriz_Datos_Binarios)[1]), ncol=dim(Matriz_Datos_Binarios)[2])
  #Matriz de unos de tamano nxp

  b=(Matriz_Unos-Matriz_Datos_Binarios)%*%t(Matriz_Datos_Binarios)

  c= t(b)

  d= (Matriz_Unos - Matriz_Datos_Binarios)%*%t(Matriz_Unos -
      Matriz_Datos_Binarios)


Similaridad_Jaccard  = a[i,j] /  (a[i,j] + b[i,j] + c[i,j])

return(Similaridad_Jaccard)
}
```

```r
Similaridad_Jaccard(1,2, Datos_Binarios)
```

```
[1] 0.3333333
```

```r
Similaridad_Jaccard(9,8, Datos_Binarios)
```

```
[1] 0
```

Programamos la matriz de similaridades de Jaccard:

```r
Matriz_Similaridad_Jaccard <- function( Matriz_Datos_
    Cuantitativos ){

 Matriz_Datos_Binarios=as.matrix(Matriz_Datos_Binarios)


  M<-matrix(NA, ncol =dim(Matriz_Datos_Cuantitativos)[1] ,
      nrow=dim(Matriz_Datos_Cuantitativos)[1] )

  for(i in 1:dim(Matriz_Datos_Cuantitativos)[1] ){
    for(j in 1:dim(Matriz_Datos_Cuantitativos)[1]){

  M[i,j]=Similaridad_Jaccard(i,j,  Matriz_Datos_Cuantitativos)

    }
  }
 return(M)
}
```

```r
Matriz_Similaridad_Jaccard(Datos_Binarios)
```

```
           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7] [,8] [,9]     [,10]     [,11]
 [1,] 1.0000000 0.3333333 0.0000000 0.3333333 0.5000000 0.3333333 0.0000000    0    0 0.0000000 1.0000000
 [2,] 0.3333333 1.0000000 0.3333333 1.0000000 0.6666667 1.0000000 0.3333333    0    0 0.3333333 0.3333333
 [3,] 0.0000000 0.3333333 1.0000000 0.3333333 0.5000000 0.3333333 1.0000000    0    0 1.0000000 0.0000000
 [4,] 0.3333333 1.0000000 0.3333333 1.0000000 0.6666667 1.0000000 0.3333333    0    0 0.3333333 0.3333333
 [5,] 0.5000000 0.6666667 0.5000000 0.6666667 1.0000000 0.6666667 0.5000000    0    0 0.5000000 0.5000000
 [6,] 0.3333333 1.0000000 0.3333333 1.0000000 0.6666667 1.0000000 0.3333333    0    0 0.3333333 0.3333333
 [7,] 0.0000000 0.3333333 1.0000000 0.3333333 0.5000000 0.3333333 1.0000000    0    0 1.0000000 0.0000000
 [8,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  NaN  NaN 0.0000000 0.0000000
 [9,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  NaN  NaN 0.0000000 0.0000000
[10,] 0.0000000 0.3333333 1.0000000 0.3333333 0.5000000 0.3333333 1.0000000    0    0 1.0000000 0.0000000
[11,] 1.0000000 0.3333333 0.0000000 0.3333333 0.5000000 0.3333333 0.0000000    0    0 0.0000000 1.0000000
[12,] 0.0000000 0.6666667 0.5000000 0.6666667 0.3333333 0.6666667 0.5000000    0    0 0.5000000 0.0000000
[13,] 1.0000000 0.3333333 0.0000000 0.3333333 0.5000000 0.3333333 0.0000000    0    0 0.0000000 1.0000000
[14,] 0.3333333 1.0000000 0.3333333 1.0000000 0.6666667 1.0000000 0.3333333    0    0 0.3333333 0.3333333
[15,] 0.5000000 0.6666667 0.0000000 0.6666667 0.3333333 0.6666667 0.0000000    0    0 0.0000000 0.5000000
[16,] 1.0000000 0.3333333 0.0000000 0.3333333 0.5000000 0.3333333 0.0000000    0    0 0.0000000 1.0000000
[17,] 1.0000000 0.3333333 0.0000000 0.3333333 0.5000000 0.3333333 0.0000000    0    0 0.0000000 1.0000000
[18,] 0.3333333 1.0000000 0.3333333 1.0000000 0.6666667 1.0000000 0.3333333    0    0 0.3333333 0.3333333
[19,] 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000 0.0000000  NaN  NaN 0.0000000 0.0000000
[20,] 0.0000000 0.3333333 0.0000000 0.3333333 0.0000000 0.3333333 0.0000000    0    0 0.0000000 0.0000000
```

En este caso pasamos de similaridad a distancia usando la transformacion:

$$\delta(i,j)_{Jaccard} = \sqrt{S(i,i)_{Jaccard} + S(j,j)_{Jaccard} - 2 \cdot S(i,j)_{Jaccard}}$$

```r
Dist_Jaccard <- function(i,j,   Matriz_Datos_Binarios){

Matriz_Datos_Binarios=as.matrix(Matriz_Datos_Binarios)

Dist_Jaccard  = sqrt( Similaridad_Jaccard(i,i,   Matriz_Datos_Binarios) + Similaridad_Jaccard(j,j,   Matriz_Datos_Binarios) - 2*Similaridad_Jaccard(i,j,   Matriz_Datos_Binarios)  )

return( Dist_Jaccard )
}
```

```r
Dist_Jaccard(1,2, Datos_Binarios)
```

```
[1] 1.154701
```

```r
Matriz_Distancias_Jaccard <- function( Matriz_Datos_Binarios ){

 Matriz_Datos_Binarios=as.matrix(Matriz_Datos_Binarios)


  M<-matrix(NA, ncol =dim(Matriz_Datos_Binarios)[1] , nrow=dim(Matriz_Datos_Binarios)[1] )

  for(i in 1:dim(Matriz_Datos_Binarios)[1] ){
    for(j in 1:dim(Matriz_Datos_Binarios)[1]){

  M[i,j]=Dist_Jaccard(i,j,  Matriz_Datos_Binarios)

   }
  }
 return(M)
}
```

```
Matriz_Distancias_Jaccard(Datos_Binarios)
```

```
         [,1]      [,2]     [,3]      [,4]      [,5]      [,6]     [,7] [,8] [,9]    [,10]    [,11]     [,12]
 [1,] 0.000000 1.1547005 1.414214 1.1547005 1.0000000 1.1547005 1.414214  NaN  NaN 1.414214 0.000000 1.4142136
 [2,] 1.154701 0.0000000 1.154701 0.0000000 0.8164966 0.0000000 1.154701  NaN  NaN 1.154701 1.154701 0.8164966
 [3,] 1.414214 1.1547005 0.000000 1.1547005 1.0000000 1.1547005 0.000000  NaN  NaN 0.000000 1.414214 1.0000000
 [4,] 1.154701 0.0000000 1.154701 0.0000000 0.8164966 0.0000000 1.154701  NaN  NaN 1.154701 1.154701 0.8164966
 [5,] 1.000000 0.8164966 1.000000 0.8164966 0.0000000 0.8164966 1.000000  NaN  NaN 1.000000 1.000000 1.1547005
 [6,] 1.154701 0.0000000 1.154701 0.0000000 0.8164966 0.0000000 1.154701  NaN  NaN 1.154701 1.154701 0.8164966
 [7,] 1.414214 1.1547005 0.000000 1.1547005 1.0000000 1.1547005 0.000000  NaN  NaN 0.000000 1.414214 1.0000000
 [8,]      NaN       NaN      NaN       NaN       NaN       NaN      NaN  NaN  NaN      NaN      NaN       NaN
 [9,]      NaN       NaN      NaN       NaN       NaN       NaN      NaN  NaN  NaN      NaN      NaN       NaN
[10,] 1.414214 1.1547005 0.000000 1.1547005 1.0000000 1.1547005 0.000000  NaN  NaN 0.000000 1.414214 1.0000000
[11,] 0.000000 1.1547005 1.414214 1.1547005 1.0000000 1.1547005 1.414214  NaN  NaN 1.414214 0.000000 1.4142136
[12,] 1.414214 0.8164966 1.000000 0.8164966 1.1547005 0.8164966 1.000000  NaN  NaN 1.000000 1.414214 0.0000000
[13,] 0.000000 1.1547005 1.414214 1.1547005 1.0000000 1.1547005 1.414214  NaN  NaN 1.414214 0.000000 1.4142136
[14,] 1.154701 0.0000000 1.154701 0.0000000 0.8164966 0.0000000 1.154701  NaN  NaN 1.154701 1.154701 0.8164966
[15,] 1.000000 0.8164966 1.414214 0.8164966 1.1547005 0.8164966 1.414214  NaN  NaN 1.414214 1.000000 1.1547005
[16,] 0.000000 1.1547005 1.414214 1.1547005 1.0000000 1.1547005 1.414214  NaN  NaN 1.414214 0.000000 1.4142136
[17,] 0.000000 1.1547005 1.414214 1.1547005 1.0000000 1.1547005 1.414214  NaN  NaN 1.414214 0.000000 1.4142136
[18,] 1.154701 0.0000000 1.154701 0.0000000 0.8164966 0.0000000 1.154701  NaN  NaN 1.154701 1.154701 0.8164966
[19,]      NaN       NaN      NaN       NaN       NaN       NaN      NaN  NaN  NaN      NaN      NaN       NaN
[20,] 1.414214 1.1547005 1.414214 1.1547005 1.4142136 1.1547005 1.414214  NaN  NaN 1.414214 1.414214 1.0000000
```

3.4.6. Más coeficientes de similaridad:

Coeficiente		Valores	Prop. euclídea
S_1	$\frac{a}{b+c}$	$(0,\infty)$	
S_2	$\frac{a}{a+b+c+d}$	$(0,1)$	sí
S_3	$\frac{a}{a+b+c}$	$(0,1)$	sí
S_4	$\frac{a+d}{a+b+c+d}$	$(0,1)$	sí
S_5	$\frac{a}{a+2(b+c)}$	$(0,1)$	sí
S_6	$\frac{a+d}{a+2(b+c)+d}$	$(0,1)$	sí
S_7	$\frac{a}{a+0.5(b+c)}$	$(0,1)$	sí
S_8	$\frac{a+d}{a+0.5(b+c)+d}$	$(0,1)$	no
$\vdots$			

Coeficiente		Valores	Prop. euclídea
$\vdots$			
S_9	$\frac{a+d-(b+c)}{a+b+c+d}$	$(-1,1)$	sí
S_{10}	$\frac{1}{2}\left(\frac{a}{a+b}+\frac{a}{a+c}\right)$	$(0,1)$	no
S_{11}	$\frac{1}{2}\left(\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{c+d}+\frac{d}{b+d}\right)$	$(0,1)$	no
S_{12}	$\frac{a}{\sqrt{(a+b)(a+c)}}$	$(0,1)$	sí
S_{13}	$\frac{ad}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	$(0,1)$	sí
S_{14}	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	$(-1,1)$	sí
S_{15}	$\frac{ad-bc}{ad+bc}$	$(-1,1)$	no

3.5. Similaridades con variables categoricas múltiples

Sean $X_1, ..., X_p$ variables **categoricas multiples** (**no binarias**) con posible distinto numero de categorias.

Los parámetros usados para construir los coeficientes de similaridad con variables categoricas multiples son:
α_{ij} = nº de coincidencias entre las p variables para ambos elementos i y j
$p - \alpha_{ij}$ = nº de no coincidencias entre las p variables para ambos elementos i y j

3.5.1. Coeficiente de Coincidencias:

La medida de similaridad mas habitual en estos casos es el coeficiente de coincidencias:

Coeficiente de Coincidencias:

El coeficiente de coincidencias entre los elementos/individuos i y j respecto de las variables categoricas múlttiples (no binarias) $X_1, ..., X_p$ es:

$$S(i, j)_{Coincidencias} = \frac{\alpha_{ij}}{p}$$

Observación:

Cuando las variables son binarias el coeficiente de coincidencias coincide con el de Sokal, puesto que $\alpha_{ij} = a_{ij} + b_{ij}$

Distancia de Coincidencias

Distancia de Coincidencias:

Obtenemos la distancia de coincidencias:

$$\delta(i, j)_{Coincidencias} = \sqrt{S(i, i)_{Coincidencias} + S(j, j)_{Coincidencias} - 2 \cdot S(i, j)_{Coincidencias}}$$

3.5.2. Aplicación en R:

```r
Datos_Categoricos_Multiples <- Datos_Mixtos%>% select(8:10)
```

A tibble: 50 x 3

X8 <dbl>	X9 <dbl>	X10 <dbl>
3	2	4
4	3	4
3	2	0
2	0	2
2	2	4
3	0	1
1	0	1
2	3	2
2	2	1
2	2	4
1	0	2
1	1	2
0	3	3
3	1	1
4	0	2

1-15 of 50 rows Previous 1 2 Next

3.5.3. Coeficiente de similaridad de Coincidencias:

Primero programamos una funcion que nos permita obtener $\alpha_{i,j}$

```
alpha<- function(i,j, Matriz_Datos_Categoricos_Multiples){

  X=as.matrix(Matriz_Datos_Categoricos_Multiples)

  alpha=ifelse( X[i,]==X[j,] , yes = 1 , 0)

  # Otra forma de hacer lo mismo, pero menos eficiente:

  # alpha=rep(0, dim(Matriz_Datos_Categoricos_Multiples)[2])

  # for(k in 1:dim(X)[2]){
  # if( X[i,k]==X[j,k] ){ alpha[k]=1 } else { alpha[k]=0 }
  # }

  alpha=sum(alpha)

  return(alpha)
}
```

```
alpha(1,3 , Datos_Categoricos_Multiples)
```

```
[1] 2
```

Ahora programamos el coeficiente de coincidencias:

```
Similaridad_Coincidencias <- function(i,j,   Matriz_Datos_
    Categoricos_Multiples){

Matriz_Datos_Categoricos_Multiples=as.matrix(Matriz_Datos_
    Categoricos_Multiples)

Similaridad_Coincidencias  =  alpha(i,j, Matriz_Datos_
    Categoricos_Multiples) / dim(Matriz_Datos_Categoricos_
    Multiples)[2]

return(Similaridad_Coincidencias )
}
```

```
Similaridad_Coincidencias(1,3, Datos_Categoricos_Multiples)
```

```
[1] 0.6666667
```

En este caso pasamos de similaridad a distancia usando la transformacion:

$$\delta(i,j)_{Coincidencias} = \sqrt{S(i,i)_{Coincidencias} + S(j,j)_{Coincidencias} - 2 \cdot S(i,j)_{Coincidencias}}$$

```
Dist_Coincidencias <- function(i,j,   Matriz_Datos_Categoricos_Multiples){

Matriz_Datos_Categoricos_Multiples=as.matrix(Matriz_Datos_Categoricos_Multiples)

Dist_Coincidencias  = sqrt( Similaridad_Coincidencias(i,i, Matriz_Datos_Categoricos_Multiples) + Similaridad_Coincidencias(j,j,   Matriz_Datos_Categoricos_Multiples) - 2*Similaridad_Coincidencias(i,j,   Matriz_Datos_Categoricos_Multiples)  )

return(Dist_Coincidencias )
}
```

```
Dist_Coincidencias(1,3, Datos_Categoricos_Multiples)
```

```
[1] 0.8164966
```

```
Matriz_Similaridad_Coincidencias <- function( Matriz_Datos_Categoricos_Multiples ){

  Matriz_Datos_Categoricos_Multiples=as.matrix(Matriz_Datos_Categoricos_Multiples)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Categoricos_Multiples)[1] , nrow=dim(Matriz_Datos_Categoricos_Multiples)[1] )

  for(i in 1:dim(Matriz_Datos_Categoricos_Multiples)[1] ){
    for(j in 1:dim(Matriz_Datos_Categoricos_Multiples)[1]){

  M[i,j]=Similaridad_Coincidencias(i,j,  Matriz_Datos_Categoricos_Multiples)

```

```
    }
   }
  return(M)
}
```

```
Matriz_Similaridad_Coincidencias(Datos_Categoricos_Multiples)
```

3.5.4. Mas coeficentes de similaridad:

Coeficiente		Valores	Cuando se aplica sobre binarias ...
SC_1	$\frac{\alpha}{p}$	$(0,1)$	S_4
SC_2	$\frac{\alpha}{p-\alpha}$	$(0,\infty)$	
SC_3	$\frac{\alpha-(p-\alpha)}{p}$	$(-1,1)$	S_9
SC_4	$\frac{\alpha}{\alpha+2(p-\alpha)}$	$(0,1)$	S_6

4. DISTANCIAS CON VARIABLES DE TIPO MIXTO:

Sean $X = (X_1, ..., X_p)$ una matriz de datos de tipo mixto tal que:

$X_1, ..., X_{p_1}$ son variables cuantitativas

$X_{p_1+1}, ..., X_{p_1+p_2}$ son variables categoricas binarias

$X_{p_1+p_2+1}, ..., X_{p_1+p_2+p_3}$ son variables categoricas multiples (no binarias).

Donde: $p = p_1 + p_2 + p_3$

4.1. Coeficiente de similaridad de Gower:

Coeficiente de similaridad de Gower:

El coeficiente de similaridad de Gower entre los elementos i y j respecto de las variables $X_1, ..., X_p$ es:

$$S(i, j)_{Gower} = \frac{\sum_{k=1}^{p_1}\left(1 - \frac{|\, x_{ik} - x_{jk} \,|}{G_k}\right) + a_{ij} + \alpha_{ij}}{p_1 + (p_2 - d_{ij}) + p_3}$$

Donde:

p_1 es el numero de variables cuantitativas

p_2 es el numero de variables categoricas binarias

p_3 es el numero de variables categoricas multiples (no binarias)

G_k es el rango de la k-esima variable cuantitativa ($G_k = max(X_k) - min(X_k)$)

a_{ij} es el numero de variables binarias (hay p_2) para las que las respuesta es 1 en ambos individuos i y j

d_{ij} es el numero de variables binarias (hay p_2) para las que las respuesta es 0 en ambos individuos i y j

α_{ij} es el numero de coincidencias entre las variables categoricas multiples no binarias (hay p_3) para los individuos i y j

4.1.1. Distancia de Gower:

Distancia de Gower:

La distancia de Gower se obtiene como:

$$\delta(i, j)_{Gower} = \sqrt{1 - S(i, j)_{Gower}}$$

4.1.2. Propiedades:

El coeficiente de similaridad de Gower es la suma de diferentes coeficientes apropiados para cada tipo de variables.

Si solo tenemos variables cuantitativas, la distancia que se obtiene es:

$$\frac{1}{p}\sum_{k=1}^{p}\left(1-\frac{\mid x_{ik}-x_{jk}\mid}{G_k}\right)$$

Si solo tenemos variables categoricas binarias, el coeficiente de similaridad de Gower coincide con el de Jaccard.

Si solo tenemos variables categoricas múltiples no binarias, el coeficiente d Gower coincide con el coeficiente de coincidencias.

Con esta idea pueden construirse otros coeficientes de similaridad para datos de tipo mixto. Algunas recomendaciones para ello son las siguientes:

Si se quiere que el coeficiente resultante tega la propiedad euclidea, todos los coeficientes que se combinen deben tenerla.

Para variables cuantitativas, deben usarse coeficientes que dividan cada comparacion por un factor de normalizacion antes de sumar.

Para variables binarias y cualitativas serán preferibles aquellos coeficientes que tomen valores en [0,1] para evitar rescalar las similaridades antes de sumar

4.1.3. Aplicación en R: Coeficiente de Gower

Programamos el coeficiente de Gower:

```
Similaridad_Gower <- function(i,j,   Matriz_Datos_Mixtos, p1, p2, p3){

  X=as.matrix(Matriz_Datos_Mixtos) #tienen que estar las variables ordenadas del siguiente modo: las p1 primeras son cuantitativas, las p2 siguientes son binarias, las p3 siguientes son categoricas multiples (no binarias). De modo que p=p1+p2+p3

#####################
  G<- function(k, X){

  G  =max(X[,k])-min(X[,k])

  return(G)
  }
#####################

  G_vector<-rep(0, p1)

  for(r in 1:p1){
  G_vector[r]=G(r, X)
  }

  Matriz_Datos_Binarios = X[ , (p1+1):(p2+p1)]

  a= Matriz_Datos_Binarios %*% t(Matriz_Datos_Binarios)

  unos<- rep(1, dim(Matriz_Datos_Binarios)[2])

  Matriz_Unos<- matrix( rep(unos, dim(Matriz_Datos_Binarios)[1]),
              ncol=dim(Matriz_Datos_Binarios)[2])

  d= (Matriz_Unos - Matriz_Datos_Binarios)%*%t(Matriz_Unos -
     Matriz_Datos_Binarios)

  Matriz_Datos_Categoricos_Multiples = X[ , (p1+p2+1):(p1+p2+p3)]

  Matriz_Datos_Cuantitativos = X[ , 1:p1]
```

```
Similaridad_Gower = (  sum( 1 - abs(Matriz_Datos_Cuantitativos[i,] - Matriz_Datos_Cuantitativos[j,])/G_vector ) + a[i,j] + alpha(i,j,Matriz_Datos_Categoricos_Multiples)  ) / (p1+p2 - d[i,j] + p3)

return(Similaridad_Gower)
}
```

```
Similaridad_Gower(1,3, Datos_Mixtos, p1=4, p2=3, p3=3)
```

```
[1] 0.5165602
```

```
Matriz_Similaridad_Gower <- function( Matriz_Datos_Mixtos, p1, p2, p3 ){

  Matriz_Datos_Mixtos=as.matrix(Matriz_Datos_Mixtos)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Mixtos)[1] , nrow=dim(Matriz_Datos_Mixtos)[1] )

  for(i in 1:dim(Matriz_Datos_Mixtos)[1] ){
    for(j in 1:dim(Matriz_Datos_Mixtos)[1]){

  M[i,j]=Similaridad_Gower(i,j,  Matriz_Datos_Mixtos, p1, p2, p3)

    }
  }
 return(M)
}
```

```r
Matriz_Similaridad_Gower(Datos_Mixtos, p1=4, p2=3, p3=3)
```

```
           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]     [,10]     [,11]
 [1,] 1.0000000 0.5654928 0.5165602 0.4022478 0.6565382 0.5125703 0.2935260 0.3063523 0.5264566 0.5868549 0.4481669
 [2,] 0.5654928 1.0000000 0.3303969 0.5859647 0.5928171 0.6328963 0.3388553 0.3216607 0.3230980 0.5301022 0.3604663
 [3,] 0.5165602 0.3303969 1.0000000 0.3911722 0.5015897 0.4316305 0.4797372 0.3635812 0.4985116 0.5921263 0.3277794
 [4,] 0.4022478 0.5859647 0.3911722 1.0000000 0.6069985 0.6871983 0.4754642 0.5024760 0.4236752 0.5292688 0.6322549
 [5,] 0.6565382 0.5928171 0.5015897 0.6069985 1.0000000 0.5556749 0.4508424 0.4317373 0.5902273 0.8460574 0.4590014
 [6,] 0.5125703 0.6328963 0.4316305 0.6871983 0.5556749 1.0000000 0.5670997 0.2499051 0.4328212 0.4569248 0.4930530
 [7,] 0.2935260 0.3388553 0.4797372 0.4754642 0.4508424 0.5670997 1.0000000 0.3503660 0.4664281 0.4850596 0.5202198
 [8,] 0.3063523 0.3216607 0.3635812 0.5024760 0.4317373 0.2499051 0.3503660 1.0000000 0.5576532 0.4635663 0.5710651
 [9,] 0.5264566 0.3230980 0.4985116 0.4236752 0.5902273 0.4328212 0.4664281 0.5576532 1.0000000 0.7121911 0.4168814
[10,] 0.5868549 0.5301022 0.5921263 0.5292688 0.8460574 0.4569248 0.4850596 0.4635663 0.7121911 1.0000000 0.3488899
[11,] 0.4481669 0.3604663 0.3277794 0.6322549 0.4590014 0.4930530 0.5202198 0.5710651 0.4168814 0.3488899 1.0000000
[12,] 0.3084297 0.4739225 0.5005427 0.6338809 0.3949563 0.4751561 0.5518909 0.4587867 0.3803721 0.4635851 0.5263597
[13,] 0.5355538 0.4939358 0.3721983 0.4236313 0.4447447 0.3951694 0.3437900 0.4758841 0.4375915 0.3747110 0.4926987
[14,] 0.4954432 0.5796797 0.5186107 0.5809913 0.5031109 0.8079241 0.4709507 0.3170245 0.4400840 0.4154410 0.4239709
[15,] 0.4196521 0.5699125 0.3110001 0.6653853 0.4012674 0.5981570 0.3626549 0.4541431 0.3605311 0.2998350 0.6759610
[16,] 0.5825527 0.4547046 0.2532723 0.6112008 0.6723344 0.4641745 0.4170948 0.5397802 0.4604751 0.5401412 0.6441637
[17,] 0.5301965 0.3169008 0.3859776 0.6051784 0.6424607 0.3550158 0.2193040 0.5524408 0.6168983 0.5343449 0.6063757
[18,] 0.4503915 0.5523242 0.4729275 0.6241128 0.5440019 0.8128936 0.4711159 0.3370115 0.4292262 0.4222220 0.4690993
[19,] 0.4757416 0.2733369 0.5096197 0.3423419 0.4717718 0.3047944 0.4422621 0.4401317 0.6427924 0.5466831 0.5640501
[20,] 0.2557322 0.4048409 0.4715115 0.3856562 0.2305699 0.3107696 0.3137202 0.4097992 0.3419218 0.2809335 0.3527430
```

En este caso pasamos de similaridad a distancia usando la transformacion:

$$\delta(i, j)_{Gower} = \sqrt{1 - S(i, j)_{Gower}}$$

```r
Dist_Gower <- function(i, j, Matriz_Datos_Mixtos , p1, p2, p3)
    {

Dist_Gower <- sqrt( 1 - Similaridad_Gower(i, j, Matriz_Datos_
   Mixtos , p1, p2, p3) )

return(Dist_Gower)
}
```

```r
Dist_Gower(1,3, Datos_Mixtos, p1=4, p2=3, p3=3)
```

```
[1] 0.6952984
```

```r
Matriz_Dist_Gower <- function( Matriz_Datos_Mixtos , p1 , p2 , p3
    ){

  Matriz_Datos_Mixtos=as.matrix(Matriz_Datos_Mixtos)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Mixtos)[1] , nrow=dim(
      Matriz_Datos_Mixtos)[1] )

  for(i in 1:dim(Matriz_Datos_Mixtos)[1] ){
    for(j in 1:dim(Matriz_Datos_Mixtos)[1]){

  M[i,j]=Dist_Gower(i,j,  Matriz_Datos_Mixtos, p1, p2, p3)

    }
  }
 return(M)
}
```

```r
Matriz_Dist_Gower(Datos_Mixtos, p1=4, p2=3, p3=3)
```

```
           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]     [,10]     [,11]
 [1,] 0.0000000 0.6591716 0.6952984 0.7731444 0.5860561 0.6981616 0.8405201 0.8328551 0.6881449 0.6427636 0.7428547
 [2,] 0.6591716 0.0000000 0.8182928 0.6434557 0.6381088 0.6058908 0.8131080 0.8236136 0.8227405 0.6854909 0.7997085
 [3,] 0.6952984 0.8182928 0.0000000 0.7802742 0.7059818 0.7539029 0.7212925 0.7977586 0.7081584 0.6386499 0.8198906
 [4,] 0.7731444 0.6434557 0.7802742 0.0000000 0.6268983 0.5592868 0.7242484 0.7053538 0.7591606 0.6860985 0.6064199
 [5,] 0.5860561 0.6381088 0.7059818 0.6268983 0.0000000 0.6665771 0.7410517 0.7538320 0.6401349 0.3923553 0.7355261
 [6,] 0.6981616 0.6058908 0.7539029 0.5592868 0.6665771 0.0000000 0.6579516 0.8660802 0.7531128 0.7369364 0.7120021
 [7,] 0.8405201 0.8131080 0.7212925 0.7242484 0.7410517 0.6579516 0.0000000 0.8059988 0.7304600 0.7175935 0.6926617
 [8,] 0.8328551 0.8236136 0.7977586 0.7053538 0.7538320 0.8660802 0.8059988 0.0000000 0.6650916 0.7324163 0.6549312
 [9,] 0.6881449 0.8227405 0.7081584 0.7591606 0.6401349 0.7531128 0.7304600 0.6650916 0.0000000 0.5364782 0.7636220
[10,] 0.6427636 0.6854909 0.6386499 0.6860985 0.3923553 0.7369364 0.7175935 0.7324163 0.5364782 0.0000000 0.8069139
[11,] 0.7428547 0.7997085 0.8198906 0.6064199 0.7355261 0.7120021 0.6926617 0.6549312 0.7636220 0.8069139 0.0000000
[12,] 0.8316070 0.7253120 0.7067229 0.6050777 0.7778455 0.7244611 0.6694095 0.7356720 0.7871645 0.7324035 0.6882153
[13,] 0.6815029 0.7113819 0.7923394 0.7591895 0.7451546 0.7777086 0.8100679 0.7239585 0.7499390 0.7907522 0.7122509
[14,] 0.7103216 0.6483212 0.6938223 0.6473088 0.7049036 0.4382646 0.7273578 0.8264233 0.7482754 0.7645646 0.7589658
[15,] 0.7618057 0.6558105 0.8300602 0.5784589 0.7737781 0.6339109 0.7983389 0.7388213 0.7996680 0.8367586 0.5692443
[16,] 0.6461016 0.7384412 0.8641341 0.6235376 0.5724208 0.7320010 0.7634823 0.6783950 0.7345236 0.6781289 0.5965202
[17,] 0.6854221 0.8264981 0.7835958 0.6283483 0.5979459 0.8031091 0.8835700 0.6689987 0.6189521 0.6823892 0.6273949
[18,] 0.7413559 0.6690858 0.7259976 0.6130964 0.6752763 0.4325579 0.7272442 0.8142411 0.7554957 0.7601171 0.7286293
[19,] 0.7240569 0.8524454 0.7002716 0.8109612 0.7267931 0.8337899 0.7468185 0.7482435 0.5976684 0.6732881 0.6602650
[20,] 0.8627095 0.7714656 0.7269722 0.7838009 0.8771717 0.8301990 0.8284201 0.7682453 0.8112202 0.8479779 0.8045229
```

4.2. Coeficiente de similaridad de Gower-Mahalanobis:

Coeficiente de similaridad de Gower-Mahalanobis:

El coeficiente de similaridad de Gower-Mahalanobis entre los elementos i y j respecto de las variables $X_1, ..., X_p$ es:

$$S(i,j)_{Gower-Maha} = \frac{\left(1 - \frac{\delta(i,j)_{Maha}}{max(D_{Maha})}\right) + a_{ij} + \alpha_{ij}}{(p_2 - d_{ij}) + p_3}$$

Donde:

p_2 es el numero de variables categoricas binarias

p_3 es el numero de variables categoricas multiples (no binarias)

$\delta(i,j)_{Maha}$ es la distancia de Mahalanobis entre los individuos respecto de las p_1 variables cuantitativas

$max(D_{Maha})$ es el maximo valor de la matriz de distancias de Mahalanobis $\delta(i,j)_{Maha}$ entre los individuos respecto de las p_1 variables cuantitativas.

a_{ij} es el numero de variables binarias (hay p_2) para las que las respuesta es 1 en ambos individuos i y j

d_{ij} es el numero de variables binarias (hay p_2) para las que las respuesta es 0 en ambos individuos i y j

α_{ij} es el numero de coincidencias entre las variables categoricas multiples no binarias (hay p_3) para los individuos i y j

4.2.1. Distancia de Gower-Mahalanobis:

Distancia de Gower-Mahalanobis:

La distancia de Gower-Mahalanobis:

$$\delta(i,j)_{Gower-Maha} = \sqrt{S(i,i)_{Gower-Maha} + S(j,j)_{Gower-Maha} - 2 \cdot S(i,j)_{Gower-Maha}}$$

4.2.2. Aplicación en R: Coeficiente de similaridad de Gower-Mahalanobis

```
Similaridad_Gower_Mahalanobis <- function(i,j,   Matriz_Datos_Mixtos, p1, p2, p3){

  X=as.matrix(Matriz_Datos_Mixtos)
#tienen que estar las variables ordenadas del siguiente modo: las p1 primeras son cuantitativas, las p2 siguientes son binarias, las p3 siguientes son categoricas multiples (no binarias). De modo que p=p1+p2+p3

######################
  G<- function(k, X){

  G=max(X[,k])-min(X[,k])

  return(G)
  }
######################

  G_vector<-rep(0, p1)

  for(r in 1:p1){
  G_vector[r]=G(r, X)
  }

  Matriz_Datos_Binarios = X[ , (p1+1):(p2+p1)]

  a= Matriz_Datos_Binarios %*% t(Matriz_Datos_Binarios)

  unos<- rep(1, dim(Matriz_Datos_Binarios)[2])

  Matriz_Unos<- matrix( rep(unos, dim(Matriz_Datos_Binarios)[1]),
              ncol=dim(Matriz_Datos_Binarios)[2])

  d= (Matriz_Unos - Matriz_Datos_Binarios)%*%t(Matriz_Unos -
      Matriz_Datos_Binarios)

  Matriz_Datos_Categoricos_Multiples = X[ , (p1+p2+1):(p1+p2+p3)]

  Matriz_Datos_Cuantitativos = X[ , 1:p1]

```

```
  max_maha=max(Matriz_Datos_Cuantitativos)

Similaridad_Gower_Mahalanobis = (  1 - Dist_Mahalanobis(i,j, Matriz_Datos_Cuantitativos)/max_maha  + a[i,j] + alpha(i,j, Matriz_Datos_Categoricos_Multiples)  ) / ( p2- d[i,j] + p3 )

return(Similaridad_Gower_Mahalanobis)
}
```

```
Similaridad_Gower_Mahalanobis(1,2, Datos_Mixtos, p1=4, p2=3, p3=3)
```

```
Matriz_Similaridad_Gower_Mahalanobis <- function( Matriz_Datos_Mixtos, p1, p2, p3 ){

  Matriz_Datos_Mixtos=as.matrix(Matriz_Datos_Mixtos)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Mixtos)[1] , nrow=dim(Matriz_Datos_Mixtos)[1] )

  for(i in 1:dim(Matriz_Datos_Mixtos)[1] ){
    for(j in 1:dim(Matriz_Datos_Mixtos)[1]){

  M[i,j]=Similaridad_Gower_Mahalanobis(i,j,  Matriz_Datos_Mixtos, p1, p2, p3)

   }
  }
 return(M)
}
```

```
Matriz_Similaridad_Gower_Mahalanobis(Datos_Mixtos, p1=4, p2=3, p3=3)
```

En este caso pasamos de similaridad a distancia usando la transformacion:

$$\delta(i,j)_{Gower-Maha} = \sqrt{S(i,i)_{Gower-Maha} + S(j,j)_{Gower-Maha} - 2 \cdot S(i,j)_{Gower-Maha}}$$

```
Dist_Gower_Mahalanobis <- function(i, j, Matriz_Datos_Mixtos , p1, p2, p3) {

Dist_Gower_Mahalanobis <- sqrt( Similaridad_Gower_Mahalanobis( i, i, Matriz_Datos_Mixtos , p1, p2, p3) + Similaridad_Gower_Mahalanobis(j, j, Matriz_Datos_Mixtos , p1, p2, p3) -2*Similaridad_Gower_Mahalanobis(i, j, Matriz_Datos_Mixtos , p1, p2, p3))

return(Dist_Gower_Mahalanobis)
}
```

```
Dist_Gower_Mahalanobis(1,2, Datos_Mixtos, p1=4, p2=3, p3=3)
```

```
Matriz_Dist_Gower_Mahalanobis <- function( Matriz_Datos_Mixtos , p1, p2, p3 ){

  Matriz_Datos_Mixtos=as.matrix(Matriz_Datos_Mixtos)

  M<-matrix(NA, ncol =dim(Matriz_Datos_Mixtos)[1] , nrow=dim(Matriz_Datos_Mixtos)[1] )

  for(i in 1:dim(Matriz_Datos_Mixtos)[1] ){
    for(j in 1:dim(Matriz_Datos_Mixtos)[1]){

  M[i,j]=Dist_Gower_Mahalanobis(i,j,  Matriz_Datos_Mixtos, p1, p2, p3)

    }
  }
 return(M)
}
```

```
Matriz_Dist_Gower_Mahalanobis(Datos_Mixtos, p1=4, p2=3, p3=3)
```