

Escalado Multidimensional

Ana Karen Martínez Marín

2022-06-02

Práctica del Escalado Multidimensional por coordenadas principales

Matriz de datos

En esta ocasión se trabajó con la matriz denominada **eurodist** precargada en R.

- Cargamos la matriz de datos eurodist.

```
data.dist<-eurodist
```

- Exploración de la matriz de datos

```
dim(data.dist)
```

```
## NULL
```

No arroja NULL debido a que los datos no están transformados, ya que están escalonados.

- Transformación de los datos en matriz

```
data.dist<-as.matrix(data.dist)
```

- Extracción de las filas de la matriz
- Número de ciudades

```
n<-nrow(data.dist)
```

Escalado Multidimensional Clásico

- 1.- Cálculo de autovalores.

```
mds.cities<-cmdscale(data.dist, eig = TRUE)
```

```
mds.cities
```

```
## $points
```

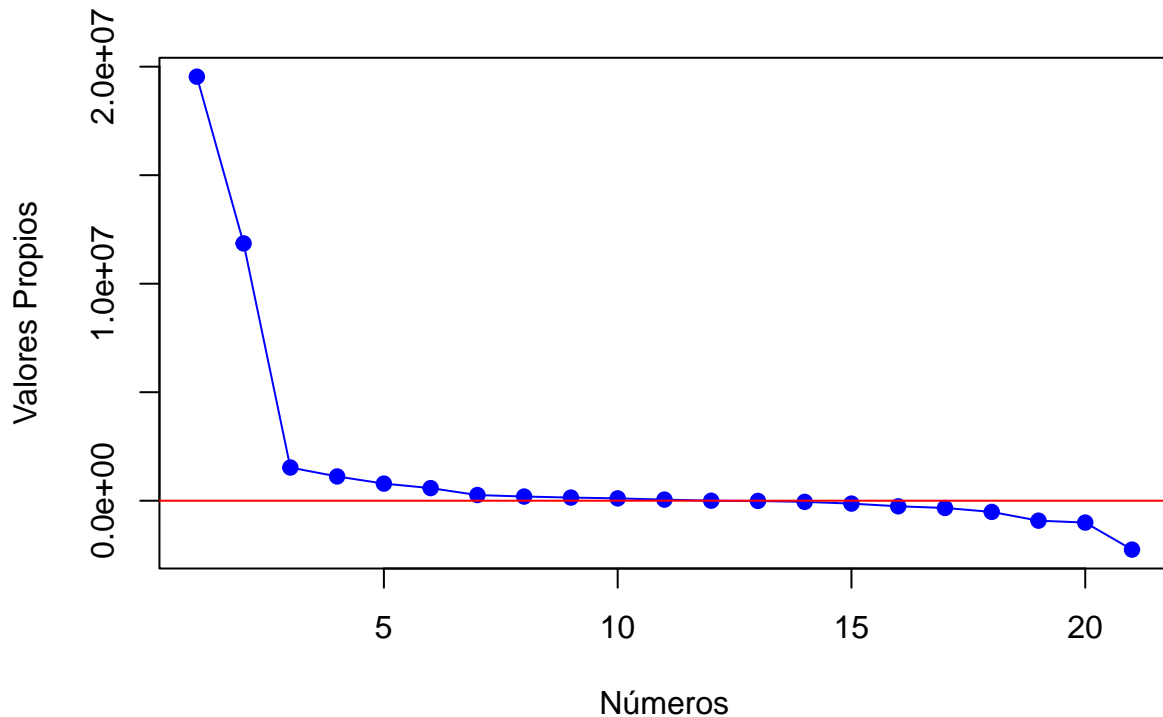
```
##           [,1]      [,2]
## Athens    2290.274680 1798.80293
## Barcelona -825.382790  546.81148
## Brussels   59.183341 -367.08135
## Calais     -82.845973 -429.91466
## Cherbourg -352.499435 -290.90843
## Cologne   293.689633 -405.31194
## Copenhagen 681.931545 -1108.64478
## Geneva     -9.423364  240.40600
## Gibraltar -2048.449113  642.45854
## Hamburg    561.108970 -773.36929
```

```
## Hook of Holland    164.921799  -549.36704
## Lisbon             -1935.040811   49.12514
## Lyons              -226.423236  187.08779
## Madrid             -1423.353697  305.87513
## Marseilles         -299.498710  388.80726
## Milan              260.878046  416.67381
## Munich             587.675679   81.18224
## Paris              -156.836257 -211.13911
## Rome               709.413282 1109.36665
## Stockholm          839.445911 -1836.79055
## Vienna             911.230500  205.93020
##
## $eig
## [1] 1.953838e+07 1.185656e+07 1.528844e+06 1.118742e+06 7.893472e+05
## [6] 5.816552e+05 2.623192e+05 1.925976e+05 1.450845e+05 1.079673e+05
## [11] 5.139484e+04 -3.259629e-09 -9.496124e+03 -5.305820e+04 -1.322166e+05
## [16] -2.573360e+05 -3.326719e+05 -5.162523e+05 -9.191491e+05 -1.006504e+06
## [21] -2.251844e+06
##
## $x
## NULL
##
## $ac
## [1] 0
##
## $GOF
## [1] 0.7537543 0.8679134
```

Dentro del objeto `mds.cities` se encuentran almacenados los valores propios (eigenvalues) en `mds.cities$eig`.

- **2.- Generación del gráfico.**

```
plot(mds.cities$eig, pch=19, col="blue",
     xlab="Números", ylab="Valores Propios",
     type="o")
abline(a=0, b=0, col="red")
```



Se identifican autovalores negativos. pero Se considera como solución el seleccionar $r=2$ coordenadas principales.

- **3.- Medidas de precisión.**

```
m<-sum(abs(mds.cities$eig[1:2]))/sum(abs(mds.cities$eig))
m
```

```
## [1] 0.7537543
```

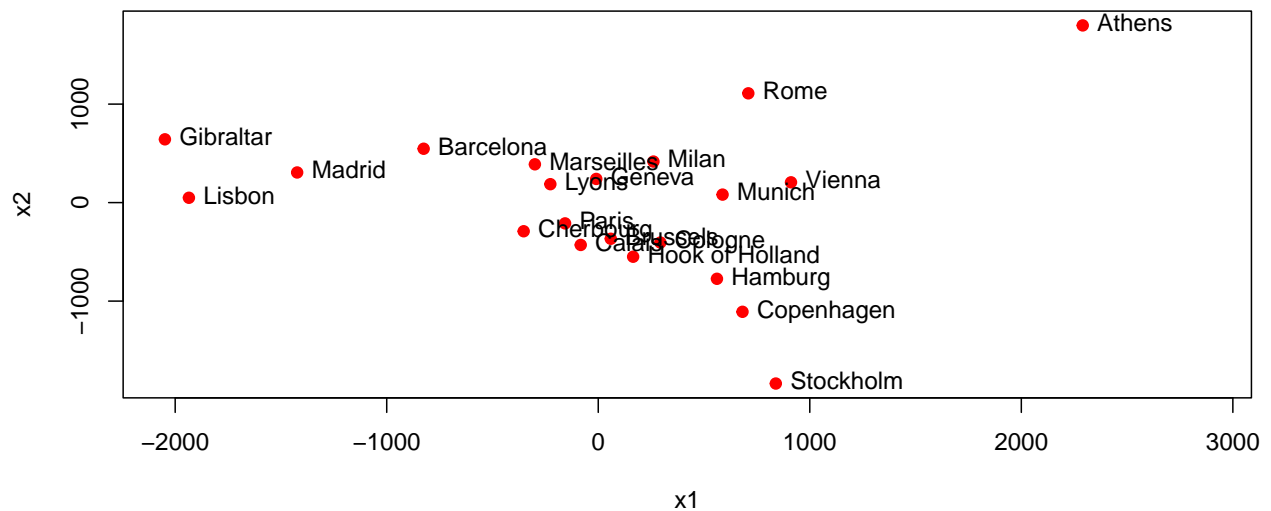
- **4.- Obtención de coordenadas principales.** Fijando $k=2$ y se realice con los dos primeros autovalores.

```
mds.cities<-cmdscale(data.dist, eig=TRUE, k=2)

x1<-mds.cities$points[,1]
x2<-mds.cities$points[,2]
```

- **5.- Generación del gráfico** En dos dimensiones de los datos con las coordenadas obtenidas.

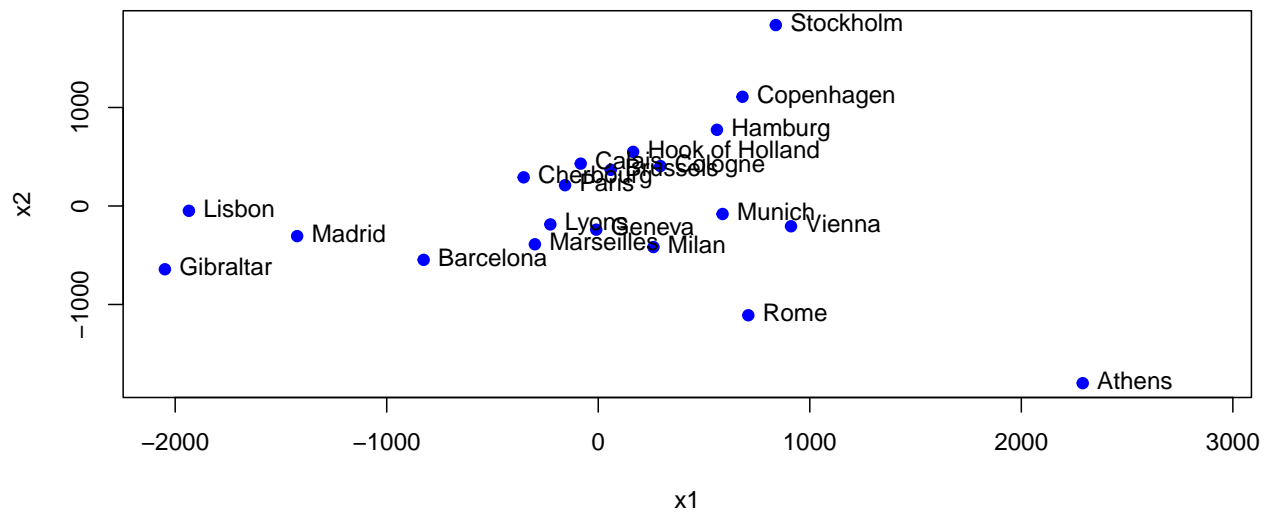
```
plot(x1,x2,pch=19, col="red",
      xlim = range(x1)+c(0,600))
text(x1,x2, pos=4, labels = rownames(data.dist),
      col="black")
```



- Se invierten los ejes del plot.

```
x2<--x2

plot2<-plot(x1,x2,pch=19, col="blue",
  xlim = range(x1)+c(0,600))
text(x1,x2, pos=4, labels = rownames(data.dist),
  col="black")
```



En el gráfico se puede notar con mayor claridad la cercanía o lejanía que hay de una ciudad a otra.