Álvaro López Pérez: alvaro.lopez.perez@udc.es

Luca Grygar Casas: luca.grygarc@udc.es

Cargamos los datos

```
load("Spain.RData")
datos <- Spain
attach(datos)
```

Cargamos el fichero de los datos e incorporamos las columnas del Data Frame al entorno de variables.

1) Ejercicio

Estudio univariante y multivariante

```
head(datos)
```

```
##            prop.inm   prop.em prop.act prop.emp prop.paro     ipc      pib
## Andalucía  5.119569  5.750006    59.35    39.03     34.23 100.736 16521.48
## Aragón     8.923154  9.400895    58.68    47.73     18.65 100.981 24646.14
## Asturias   6.237551  7.617976    51.63    40.90     20.78 100.756 19505.32
## Baleares  15.192307 11.927219    63.23    51.29     18.88 100.241 23438.19
## Canarias   7.975230  6.965184    61.55    42.42     31.08 101.041 18758.61
## Cantabria  9.136823  9.981571    56.16    45.82     18.42 101.226 20362.67
```

Vemos las primeras 6 filas, nos es útil para un primer contacto con las variables.

```
str(datos)
```

```
## 'data.frame':    19 obs. of  7 variables:
##  $ prop.inm : num  5.12 8.92 6.24 15.19 7.98 ...
##  $ prop.em  : num  5.75 9.4 7.62 11.93 6.97 ...
##  $ prop.act : num  59.4 58.7 51.6 63.2 61.5 ...
##  $ prop.emp : num  39 47.7 40.9 51.3 42.4 ...
##  $ prop.paro: num  34.2 18.6 20.8 18.9 31.1 ...
##  $ ipc      : num  101 101 101 100 101 ...
##  $ pib      : num  16521 24646 19505 23438 18759 ...
```

Tenemos un conjunto de datos numéricos que evalúa distintas características de las comunidades autónomas españolas. Estas son: empleo,paro,emigración,inmigración,actividad en cuanto a la proporción de la población de esa región. Además, el PIB y el IPC como valor económico concreto.

Podemos ver que hay una diferencia enorme entre comunidades. Vamos a considerar el PIB como la mejor medida para comparar la riqueza entre comunidades.

Rango del PIB:

```
cat("PIB: ",range(pib), "\n")
```

```
## PIB:  15223.92 30637.71
```

Ordenamos las comunidades por PIB:

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
ord <- datos %>% arrange(desc(pib))
```

Este sería el top 5:

```
h.datos <- head(ord,n=5)
h.datos
```

```
##             prop.inm  prop.em prop.act prop.emp prop.paro     ipc      pib
## Madrid     11.455740 9.002980    64.82    53.15     18.00 100.733 30637.71
## País Vasco  7.530449 5.961146    57.76    48.17     16.60 100.253 29514.39
## Navarra     9.699008 8.629264    59.55    50.66     14.92 100.622 28039.59
## Cataluña    5.451834 5.084221    62.61    50.16     19.88 100.113 26584.51
## Aragón      8.923154 9.400895    58.68    47.73     18.65 100.981 24646.14
```

Este sería el top 5 colista:

```
t.datos <- tail(ord,n=5)
t.datos
```

```
##                     prop.inm   prop.em prop.act prop.emp prop.paro     ipc
## Murcia              8.901316  8.715984    61.08    44.43     27.26 100.829
## Castilla-La Mancha 12.401142 16.596133    58.73    41.99     28.50 101.492
## Melilla            27.724255 26.379523    55.65    39.23     29.52 100.623
## Andalucía           5.119569  5.750006    59.35    39.03     34.23 100.736
## Extremadura         7.566919  9.788992    55.50    38.87     29.96 101.115
##                        pib
## Murcia             18155.32
## Castilla-La Mancha 17265.45
## Melilla            16669.70
## Andalucía          16521.48
## Extremadura        15223.92
```

Intentemos ahora comparar intuitivamente como afectan las demás variables a la riqueza de la comunidad autónoma.

```
h.mean <- colMeans(h.datos)[c(-6,-7)]
t.mean <- colMeans(t.datos)[c(-6,-7)]

h.mean
```

```
##  prop.inm   prop.em  prop.act  prop.emp prop.paro
##  8.612037  7.615701 60.684000 49.974000 17.610000
```

```
t.mean
```

```
##  prop.inm   prop.em  prop.act  prop.emp prop.paro
##  12.34264  13.44613  58.06200  40.71000  29.89400
```

A ojo podemos ver como hay diferencias considerables entre las diferentes proporciones. La tasa de inmigración y emigración es menor en las comunidades con mayor pib, además de la tasa de empleo, por ejemplo.

Estudio de las correlaciones

```
datos.cor <- cor(datos)
cat("Correlaciones a destacar: ","\n")
```

```
## Correlaciones a destacar:
```

```
cat("Emigración/Inmigración: ",datos.cor[2,1],"\n")
```

```
## Emigración/Inmigración:  0.9327393
```

```
cat("Paro/empleo: ", datos.cor[5,2], "\n")
```

```
## Paro/empleo:  0.385752
```

```
cat("IPC/PIB: ", datos.cor[7,6], "\n")
```

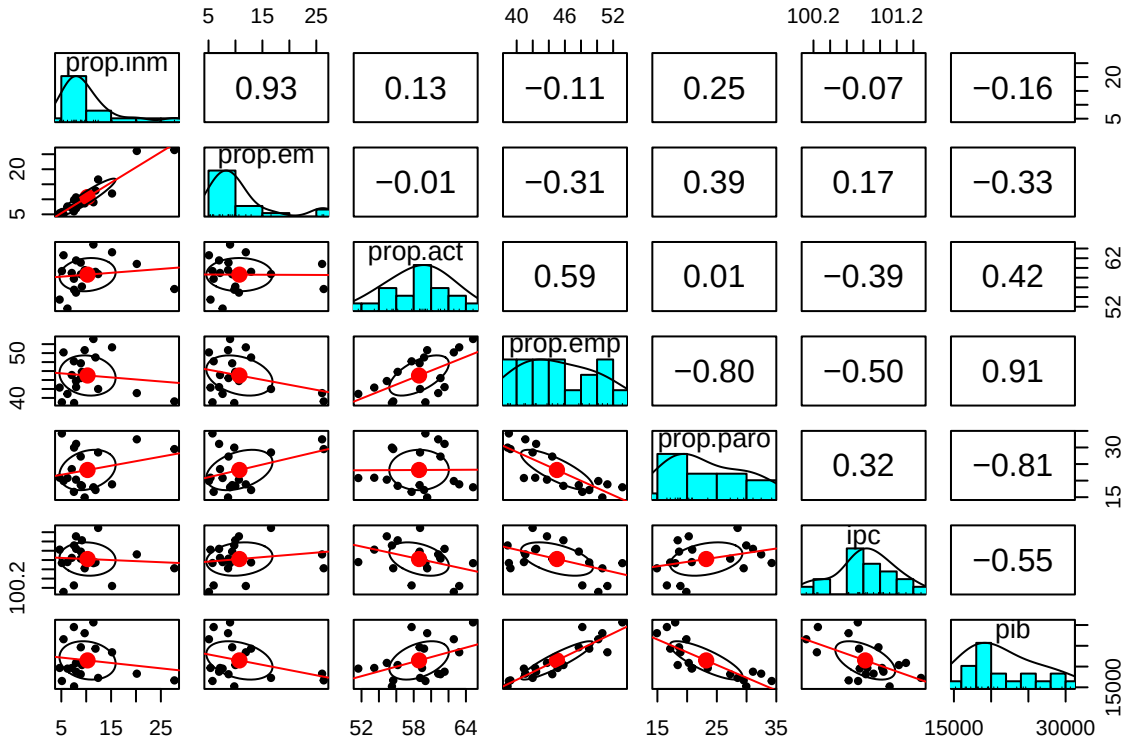
```
## IPC/PIB:  -0.5476956
```

Algunas correlaciones a destacar pueden ser, la relación entre el ipc y el pib, cuánto mayor sea una menor es la otra , entre el paro y el empleo es negativa igual que en las variables emigración e inmigración. En la siguiente gráfica podemos ver como todas están representadas a través de gráficas con nubes de puntos, además de boxplots en la diagonal que nos ayudan a ver como se distribuyen las variables.

```
library(psych)
```

```
## Warning: package 'psych' was built under R version 4.3.3
```

```
pairs.panels(datos, smooth = TRUE, density=TRUE, digits = 2, ellipses=TRUE, method="pearson", pch = 20,
```



2/3/4

Nosotros para resolver el problema utilizaremos la distancia euclídea

```
datos.scale <- scale(datos,center=TRUE,scale=TRUE)
```

datos escalados (pib con un rango enorme) Los datos escalados son más "estables".

```
d.scale<- dist(datos.scale)
```

datos sin escalar Si utilizaramos los datos sin escalar, las distancias dependerían de las unidades ,por ejemplo, si nos diese por medir la tasa de inmigración por proporción de inmigrantes por 10.000 personas tendría menos relevancia y si fuera entre 100 más, ya que esa variable tendría mayor valor numérico

```
 d<- dist(datos)
```

```
library(StatMatch)
```

distancia de mahalanobis:

```
## Warning: package 'StatMatch' was built under R version 4.3.3
## Loading required package: proxy
## Warning: package 'proxy' was built under R version 4.3.2
```

```
##
## Attaching package: 'proxy'
## The following objects are masked from 'package:stats':
##
##     as.dist, dist
## The following object is masked from 'package:base':
##
##     as.matrix
## Loading required package: survey
## Warning: package 'survey' was built under R version 4.3.3
## Loading required package: grid
## Loading required package: Matrix
## Warning: package 'Matrix' was built under R version 4.3.2
## Loading required package: survival
##
## Attaching package: 'survey'
## The following object is masked from 'package:graphics':
##
##     dotchart
## Loading required package: lpSolve
## Warning: package 'lpSolve' was built under R version 4.3.3
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 4.3.3
##
## Attaching package: 'ggplot2'
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
d.mahalanobis <- mahalanobis.dist(datos)
```

5/6/7/8/9/10

MDS en R (manual)

Matriz A:

La martiz A, es la matriz de distancias de los datos elevada al cuadrado y multiplicada por -1/2.

```
library(philentropy)
```

```
## Warning: package 'philentropy' was built under R version 4.3.3
##
## Attaching package: 'philentropy'
```

```
## The following objects are masked from 'package:psych':
##
##     distance, manhattan, minkowski
```

```
dist.dat <- as.matrix(dist(datos.scale))
n <- dim(dist.dat)[1]
A <- (-1/2)*dist.dat^2
cat("Matriz A (4 primeras columnas): \n"); A[, 1:4]
```

```
## Matriz A (4 primeras columnas):
```

```
##                     Andalucía      Aragón    Asturias   Baleares
## Andalucía            0.000000  -7.3455606  -5.3352144 -11.706441
## Aragón              -7.345561   0.0000000  -4.3035443  -4.046367
## Asturias            -5.335214  -4.3035443   0.0000000 -11.345738
## Baleares           -11.706441  -4.0463674 -11.3457384   0.000000
## Canarias            -1.248668  -4.0482409  -6.0544500  -8.203940
## Cantabria           -6.705807  -1.0276810  -2.6254114  -7.466594
## Castilla-León       -6.180302  -1.8181966  -2.0664383  -9.796188
## Castilla-La Mancha  -5.283818  -5.2814262  -6.8491773 -11.600481
## Cataluña           -10.175193  -4.2572910 -10.1346151  -2.464133
## ComValenciana       -2.844889  -1.2592462  -2.8725124  -5.091754
## Extremadura         -1.780911  -6.2781196  -2.8910765 -13.526520
## Galicia             -4.698619  -3.0378161  -0.5769325 -11.117759
## Madrid             -15.200203  -3.5194834 -14.5877750  -2.683103
## Murcia              -1.942001  -2.5945348  -4.8665067  -5.055774
## Navarra            -11.971996  -1.2170209  -7.4713665  -2.470498
## País Vasco         -11.307666  -2.8852890  -6.5294028  -3.841056
## Rioja               -9.213392  -0.5933702  -5.6563423  -2.056987
## Ceuta               -9.462881 -10.4842326 -13.1261268 -10.767581
## Melilla            -14.577497 -15.1167588 -13.9407641 -14.442096
```

Matriz de centrado

La matriz de centrado o H, es una auxiliar utilizada para centrar los datos, dado que multiplicar una fila de la matriz por un vector columna, daria como resultado un vector columna en que cada componente seria igual a los componente del vector original menos su media.

```
H <- as.matrix(diag(rep((n-1)/n,length.out=n))) # En la diagonal 6/7
H[which(H==0)] <- -1/n # En las otras posiciones, -1/7
cat("Matriz H (4 primeras columnas): \n"); H[, 1:4]
```

```
## Matriz H (4 primeras columnas):
```

```
##              [,1]        [,2]        [,3]        [,4]
##  [1,]  0.94736842 -0.05263158 -0.05263158 -0.05263158
##  [2,] -0.05263158  0.94736842 -0.05263158 -0.05263158
##  [3,] -0.05263158 -0.05263158  0.94736842 -0.05263158
##  [4,] -0.05263158 -0.05263158 -0.05263158  0.94736842
##  [5,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
##  [6,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
##  [7,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
##  [8,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
##  [9,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
## [10,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
## [11,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
## [12,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
```

```
## [13,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
## [14,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
## [15,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
## [16,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
## [17,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
## [18,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
## [19,] -0.05263158 -0.05263158 -0.05263158 -0.05263158
```

Matriz B = HAH

La matriz B es el resultado de aplicar una transformación sobre A, para dar una matriz semidefinida positiva.

```
cat("Matriz B (4 primeras columnas): \n")
```

```
## Matriz B (4 primeras columnas):
```

```
((B <- H %*% A %*% H)[, 1:4])
```

```
##             [,1]         [,2]       [,3]        [,4]
##  [1,]  7.7874797 -2.603705992  1.6234410 -3.88201621
##  [2,] -2.6037060  1.696229430 -0.3905140  0.73243204
##  [3,]  1.6234410 -0.390514046  6.1298310 -4.35013818
##  [4,] -3.8820162  0.732432035 -4.3501382  7.86136950
##  [5,]  4.6420663 -1.203131398 -0.9925397 -2.27626080
##  [6,] -1.5431035  1.089396957  1.7084673 -2.26694654
##  [7,] -0.5321324  0.784347899  2.7529070 -4.11107310
##  [8,]  1.8900598 -1.153173811 -0.5041242 -4.38965848
##  [9,] -1.4582782  1.413998687 -2.2465246  6.28972702
## [10,]  1.4634942  0.003511374  0.6070460 -0.74642647
## [11,]  5.4481009 -2.094732537  3.5091114 -6.26056252
## [12,]  1.6828467  0.298023995  4.9757084 -4.69934863
## [13,] -5.6019851  3.033109181 -5.8183816  6.95205928
## [14,]  2.7819526 -0.916206494 -0.9713776 -0.29487565
## [15,] -5.0988421  2.610507426 -1.4270374  4.43960025
## [16,] -3.3690909  2.007660747  0.5803478  4.13446395
## [17,] -4.1022083  1.472187910 -1.3739834  3.09114162
## [18,]  1.1041214 -2.962855180 -3.3879486 -0.16363355
## [19,] -0.2321997 -3.817086182 -0.4242907 -0.05985354
```

```
cat("\n Autovalores de B: \n")
```

```
##
##  Autovalores de B:
```

```
(lambda <- eigen(B)$values)
```

```
##  [1]  6.203655e+01  3.389752e+01  1.651071e+01  1.111802e+01  1.843932e+00
##  [6]  5.866170e-01  6.658247e-03  1.335688e-14  5.511543e-15  1.977561e-15
## [11]  1.109855e-15  3.530319e-16  2.262438e-16 -1.813958e-16 -4.507361e-16
## [16] -9.355795e-16 -1.489930e-15 -3.119176e-15 -3.311719e-15
```

```
autovec <- eigen(B)$vectors
vec12 <- autovec[,1:2]
lambda_12 <- diag(lambda[1:2])
cat("\n Puntos elegidos: \n")
```

```
##
##  Puntos elegidos:
```

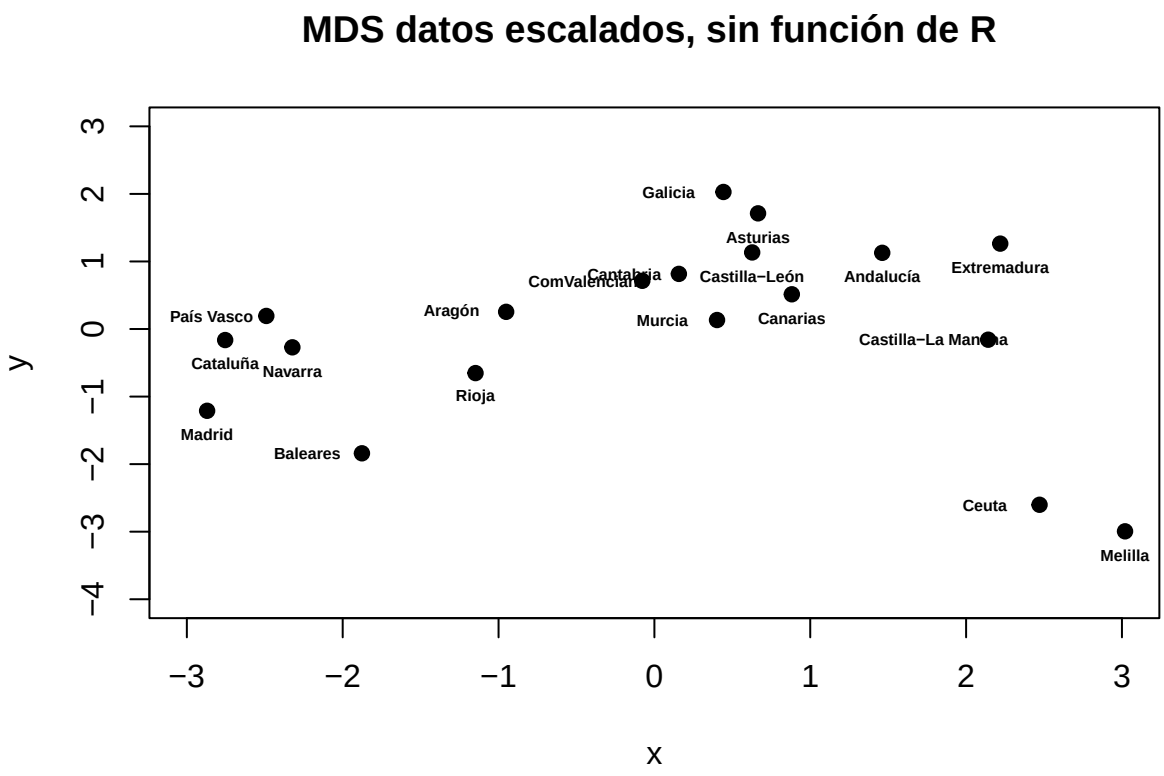
```r
(result <- vec12 %*% sqrt(lambda_12))
```

```
##              [,1]       [,2]
##  [1,]  1.46190590  1.1278025
##  [2,] -0.95115545  0.2541695
##  [3,]  0.66503246  1.7121386
##  [4,] -1.87646463 -1.8397326
##  [5,]  0.88182634  0.5141468
##  [6,]  0.15710676  0.8162380
##  [7,]  0.62750055  1.1332744
##  [8,]  2.14165309 -0.1580988
##  [9,] -2.75404238 -0.1630953
## [10,] -0.07769691  0.7138671
## [11,]  2.21897303  1.2640123
## [12,]  0.44296802  2.0286697
## [13,] -2.86986150 -1.2104842
## [14,]  0.40197557  0.1329778
## [15,] -2.32284323 -0.2709345
## [16,] -2.49010149  0.1937078
## [17,] -1.14782537 -0.6524291
## [18,]  2.47120389 -2.6016840
## [19,]  3.01984536 -2.9945460
```

```r
d.names <- row.names(Spain)
```

Representación Gráfica

```r
plot(result, pch=19, main="MDS datos escalados, sin función de R ", xlab="x",ylab="y", xlim=c(-3,3),yli
text(result-c(0, 0.35), labels=d.names,cex=0.5, font=2)
```



No vemos que sea necesario realizar ninguna rotación.

MDS función R

Podemos separar los datos en dos tipos, demografía y economía de la comunidad por lo tanto tendría sentido representar las distancias en dos dimensiones.

Datos escalados

En este caso en vez de todo el proceso anterior usaremos la función cmdscale para realizar el proceso.

```
datos.scale <- scale(datos,center=TRUE,scale=TRUE)
d <- as.matrix(dist(datos.scale))
cat("Puntos elegido:; \n")
```

```
## Puntos elegido:;
```

```
(fit <- cmdscale(d, eig=TRUE, k=2))$points
```

```
##                         [,1]       [,2]
## Andalucía           1.46190590  1.1278025
## Aragón             -0.95115545  0.2541695
## Asturias            0.66503246  1.7121386
## Baleares           -1.87646463 -1.8397326
## Canarias            0.88182634  0.5141468
## Cantabria           0.15710676  0.8162380
## Castilla-León       0.62750055  1.1332744
```

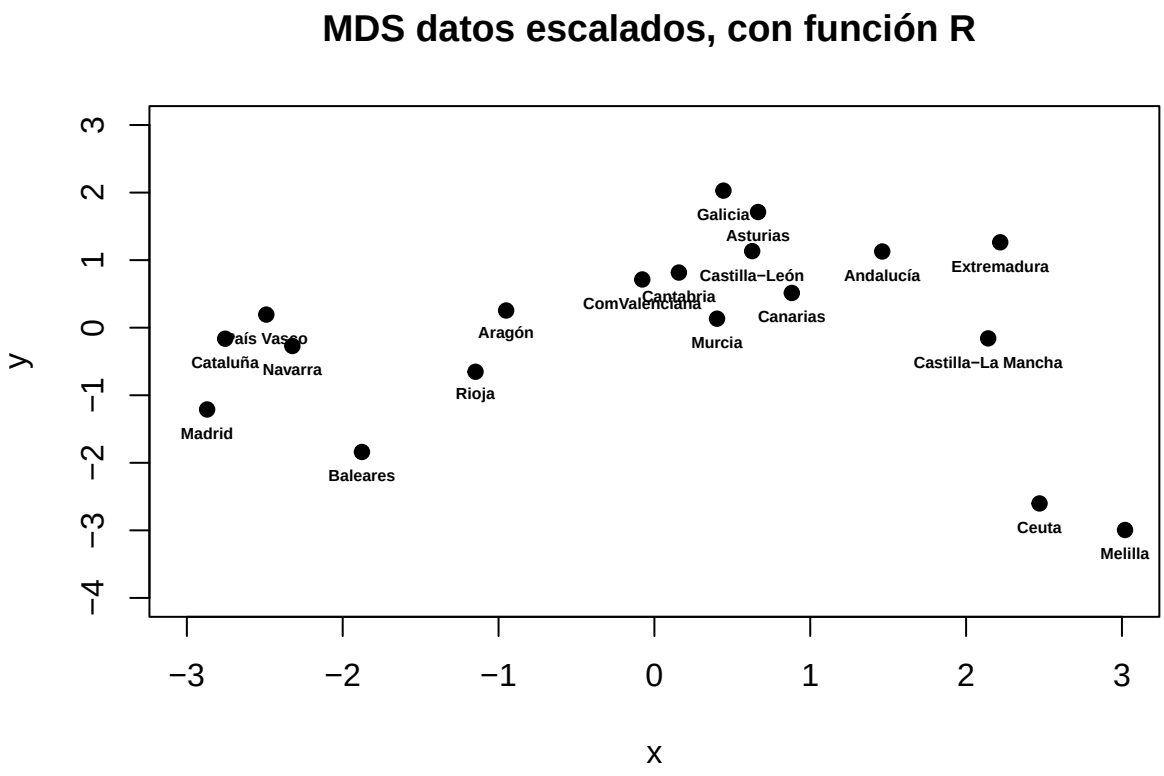
```
## Castilla-La Mancha  2.14165309 -0.1580988
## Cataluña           -2.75404238 -0.1630953
## ComValenciana      -0.07769691  0.7138671
## Extremadura         2.21897303  1.2640123
## Galicia             0.44296802  2.0286697
## Madrid             -2.86986150 -1.2104842
## Murcia              0.40197557  0.1329778
## Navarra            -2.32284323 -0.2709345
## País Vasco         -2.49010149  0.1937078
## Rioja              -1.14782537 -0.6524291
## Ceuta               2.47120389 -2.6016840
## Melilla             3.01984536 -2.9945460
```

```
# k es el número de dimensiones
```

Representamos el nuevo MDS:

Representación Gráfica

```
x <- fit$points[,1]; y <- fit$points[,2]
plot(x,y, pch=19, main="MDS datos escalados, con función R ", xlab="x",ylab="y", xlim=c(-3,3),ylim=c(-4
text(x,y-0.35, labels=d.names,cex=0.5, font=2)
```



Datos sin escalar

Creamos la matriz de ditancia con los datos sin escalar, compararemos los resultados en parte para sobreexplicar la pregunta lanzada en anteriores ejercicios. ¿Cómo afectan las unidades de las variables a las distancias? Lo

veremos con el MDS.

```
d <- as.matrix(dist(datos))
```

Usamos la función de R para representar los datos

```
lab <- row.names(datos)
cat("Puntos elegidos: \n")
```

```
## Puntos elegidos:
```

```
(fit <- cmdscale(d, eig=TRUE, k=2))$points # k es el número de dimensiones
```

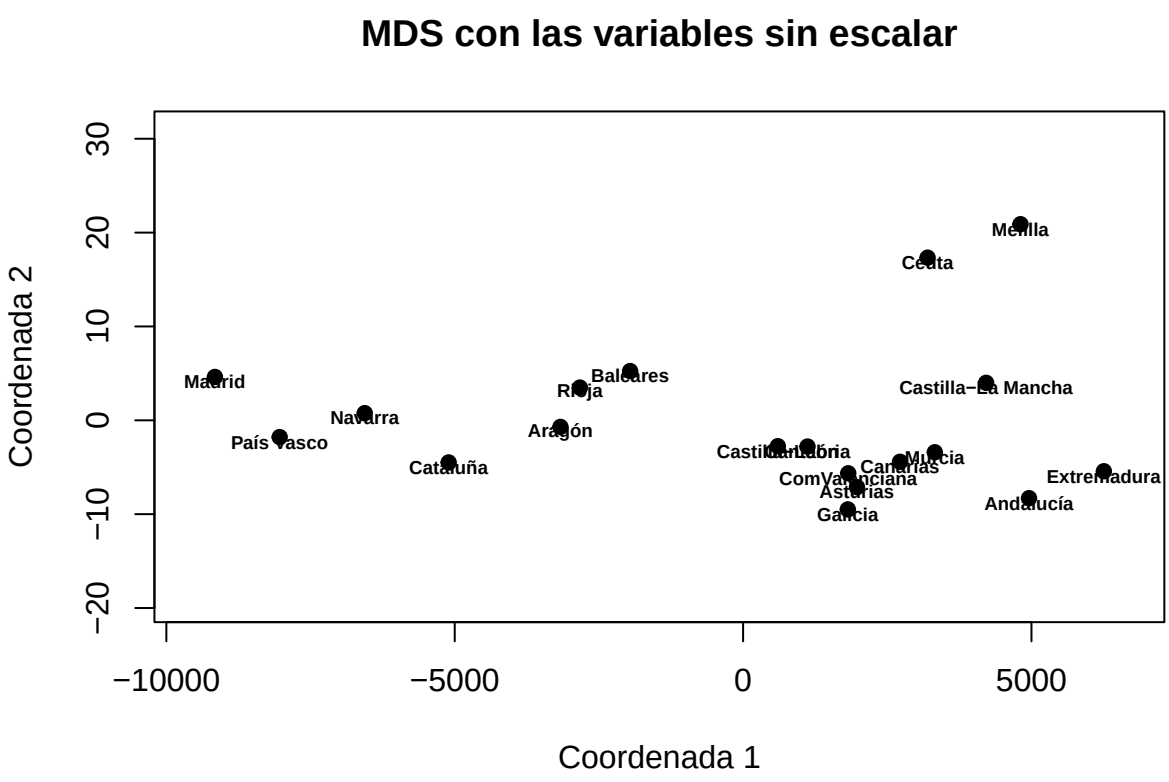
```
##                         [,1]       [,2]
## Andalucía           4957.986 -8.2982059
## Aragón             -3166.687 -0.7018420
## Asturias            1974.141 -7.1196982
## Baleares           -1958.737  5.2478556
## Canarias            2720.861 -4.4189234
## Cantabria           1116.781 -2.7957937
## Castilla-León        602.594 -2.7558750
## Castilla-La Mancha  4214.017  3.9952011
## Cataluña           -5105.052 -4.4901003
## ComValenciana       1823.124 -5.6439811
## Extremadura         6255.549 -5.4272709
## Galicia             1815.797 -9.4912415
## Madrid             -9158.258  4.6203316
## Murcia              3324.142 -3.4060526
## Navarra            -6560.133  0.7639462
## País Vasco         -8034.931 -1.7875199
## Rioja              -2829.621  3.4860347
## Ceuta               3198.652 17.3284246
## Melilla             4809.776 20.8947106
```

```
x <- fit$points[,1]; y <- fit$points[,2]
x.max <- max(x); x.min <- min(x)
y.max <- max(y); y.min <- min(y)
```

Representación Gráfica

Creamos el gráfico reescalándolo basándonos en los máximos y mínimos de las coordenadas y ajustándolos a ojo.

```
plot(x, y, xlab="Coordenada 1", ylab="Coordenada 2", main="MDS con las variables sin escalar", pch=19,
text(x, y-0.5, labels = d.names, cex=0.6, font=2)
```



Podemos percibir como este mds, "le hace más caso" a la variable del PIB que en este caso es la que más valor tiene al no ser una proporción como las demás.

11

MDS con las variables

```
dist.var <- as.matrix(prcomp(scale(Spain, center = TRUE, scale = TRUE))$rotation)
v.names <- colnames(Spain)
n <- dim(dist.var)[1]
cat("Puntos elegidos: \n")
```

```
## Puntos elegidos:
```

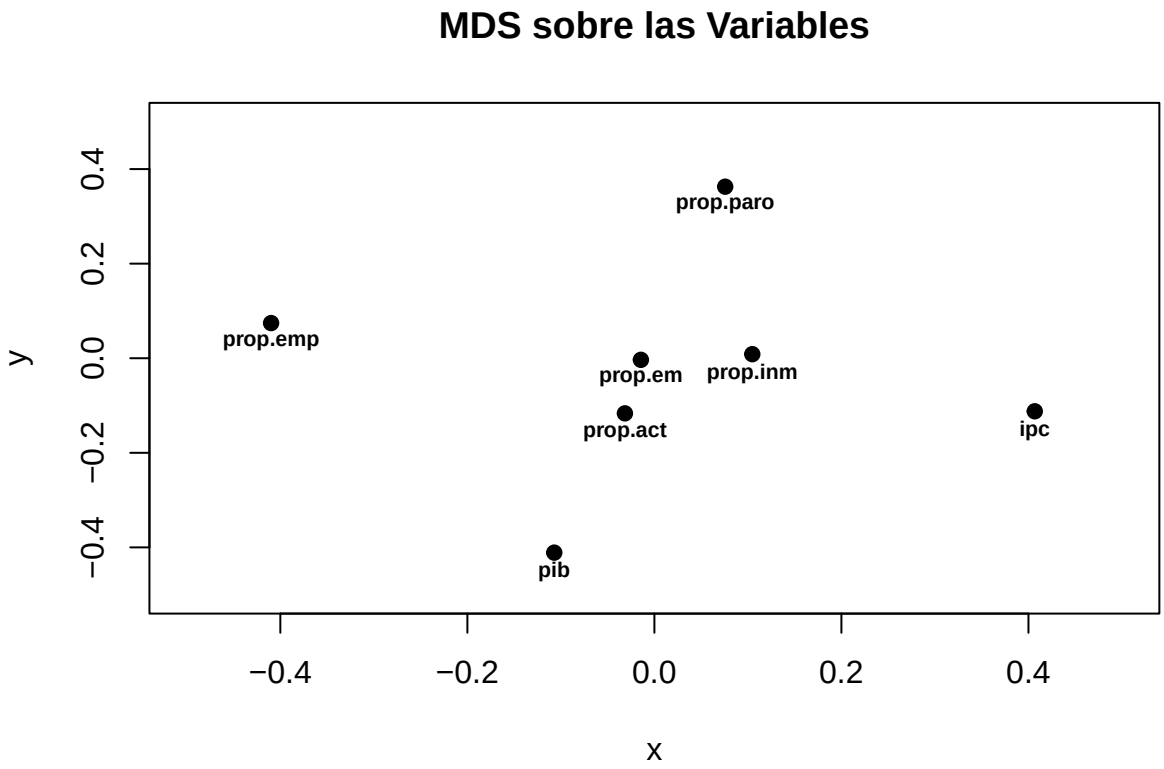
```
fit <- cmdscale(dist.var, eig = TRUE, k = 2); fit$points
```

```
##                 [,1]         [,2]
## prop.inm   0.10469084  0.008575635
## prop.em   -0.01443457 -0.003375902
## prop.act  -0.03152434 -0.116583982
## prop.emp  -0.40990577  0.074313427
## prop.paro  0.07569926  0.362743923
## ipc        0.40675912 -0.112202141
## pib       -0.10708420 -0.411074152
```

```
x <- fit$points[,1]; y <- fit$points[,2]
```

Representación gráfica

```
plot(x,y, pch=19, main="MDS sobre las Variables", xlab="x",ylab="y", xlim=c(-0.5,0.5),ylim=c(-0.5,0.5))
text(x,y-0.04, labels=v.names,cex = 0.7, font=2)
```



Conclusión

El método de reducción de la dimensión que hemos aplicado nos ha resultado útil para ver qué comunidades son más similares en cuánto a las variables ofrecidas. Además hemos podido ver que las más potentes económicamente, como descubrimos en el estudio previo, quedaban cerca en el **"mapa"**. Por inercia se ven grupos; Madrid,Cataluña,País Vasco o por otra parte Ceuta y Melilla. Graficando fácilmente una visión intuitiva de lo que ocurre en la realidad.