# PRACTICAS CON MATLAB
# (elementos necesarios para la construcción de un Biplot)

## P1. INSTRUCCIONES ELEMENTALES

### *P1.1 Definición de matrices*

El primer paso es el de como introducir los datos de una matriz de trabajo con un nombre que pueda ser utilizado posteriormente para realizar operaciones. Los elementos de la matriz irán entre corchetes, los elementos de la misma fila se separan con comas (,) o espacios y las distintas filas se separan con punto y coma (;) o con RETURN. La siguiente instrucción asigna la matriz definida a la derecha de la igualdad a la variable a que después se utilizará en los cálculos.

```
”x=[1,2,3;2,3,4
1 3 2]
x =
     1     2     3
     2     3     4
     1     3     2
```

Tendremos así una matriz denominada a, lista para cálculos posteriores.

Un punto y coma (;) al final del corchete que cierra la matriz significa que la repuesta no se muestra al definir una matriz, esto es especialmente útil cuando se utilizan varias instrucciones y no se quiere mostrar el cálculo intermedio. por ejemplo, la siguiente instrucción no mostraría el valor de la matriz a ya construida.

```
”x=[1,2,3;2,3,4
1 3 2];
```

Los elementos de la matriz son directamente utilizables en cualquier fórmula; se accede a ellos con el nombre de la variable y los índices entre paréntesis separados por una coma. La siguiente instrucción asigna el elemento de la fila 1 y de la columna 2 de la matriz a una variable denominada y.

```
”y=x(1,2)

y =
     2
```

Es posible también seleccionar submatrices, por ejemplo, si en lugar del índice de la fila escribimos dos puntos (:) significa que tomamos todas las filas, si escribimos dos números separados por dos puntos significa que tomaremos todos los valores del índice entre esos dos valores. Por ejemplo, la instrucción siguiente asigna, en la matriz denominada col12, todas las filas de x y las columnas desde la 1 hasta la 2.

```
”col12=x(:,1:2)

col12 =
     1     2
     2     3
     1     3
```

Las matrices pueden ser también matrices de caracteres. Por ejemplo, supongamos que queremos introducir una matriz con caracteres que después servirá para etiquetar las filas de la matriz x en un gráfico. Cada una de las etiquetas ocupará una fila de la matriz, las columnas serán los caracteres de la etiqueta, los caracteres se introducen entre comillas simples.

```
”l=['aa';'bb';'cc']
l =
aa
bb
cc
```

Esta forma de introducir los datos implica que las etiquetas han de tener todas el mismo número de caracteres.

En cualquier momento de una sesión de trabajo es posible comprobar cuales son las variables que se han definido hasta el momento utilizando el comando who.

```
Your variables are:
x         col12     y
```

En este momento hemos definido la matriz `x`, la variable `y` que contiene el elemento (1,2) de `x` y la matriz `col12` que contiene las dos primeras columnas de `x`. Es posible obtener información adicional utilizando el comando `whos`.

```
      Name        Size            Elements     Bytes    Density   Complex

         x        3 by 3                 9        72       Full        No
     col12        3 by 2                 6        48       Full        No
         y        1 by 1                 1         8       Full        No

Grand total is 16 elements using 128 bytes

leaving 7266896 bytes of memory free.
```

El comando nos proporciona, además, el tamaño de las matrices definidas, el número de elementos, el espacio que ocupan en memoria, la densidad de elementos distintos de cero (full significa completa) y si la matriz tiene elementos complejos o no. Proporciona también el espacio de memoria ocupado hasta el momento y el espacio de trabajo restante para operaciones futuras.

El comando `clear` borra el espacio de memoria. Seguido del nombre de varias variables separadas por comas, borra solamente las variables especificadas, si no se especifica ninguna se limpia todo el espacio de trabajo.

La instrucción help seguida de cualquier función o comando proporciona información sobre el mismo.

### *P1.2 Algunas funciones que construyen matrices especiales útiles*

El programa permite obtener algunas matrices útiles para los cálculos. Para ello utiliza funciones predefinidas con varios parámetros que definen el orden de las matrices a obtener.

La función `ones` construye una matriz de unos. Con un solo parámetro `ones(n)` construye una matriz cuadrada de orden n; con dos parámetros `ones(n,m)` construye una matriz con `n` filas y `m` columnas.

```
”b=ones(3)
b =
     1     1     1
     1     1     1
     1     1     1

”c=ones(3,1)
c =
     1
     1
     1
```

La función `zeros` construye una matriz de ceros con el mismo criterio que la función `ones`.
La función `eye(n)` construye la matriz identidad del orden especificado en el parámetro `n`.

```
”d=eye(3)
d =
     1     0     0
     0     1     0
     0     0     1
```

La función `diag(a)` obtiene, en un vector columna, la diagonal principal de la matriz `x`.

```
”e=diag(x)
e =
     1
     3
     2
```

Si se aplica a un vector (fila o columna) construye una matriz diagonal cuyos elementos diagonales

```
”f=diag(e)
f =
     1     0     0
     0     3     0
     0     0     2
```

Por ejemplo `diag(diag(x))` construiría una matriz diagonal cuyos elementos diagonales son los de la diagonal de `x`.

```
”diag(diag(x))
ans =
     1     0     0
     0     3     0
     0     0     2
```

Hará observado que la matriz aparece en una variable denominada `ans` (answer). Este es el nombre utilizado por defecto cuando el resultado de una función o un cálculo no se asigna a ninguna variable concreta. La matriz `ans` estará disponible para operar hasta que se realice otra operación sin asignar.

La función `sum(var)` produce como resultado un vector con la suma de las columnas de `var`. La función `cumsum(var)` produce como resultado una matriz con las sumas acumuladas de las columnas de `var`.

Si queremos obtener el tamaño de una matriz introducida anteriormente podemos utilizar la función `size(x)` que nos devolverá un vector con dos componentes conteniendo el número de filas y columnas de `x`.

```
»size(x)
ans =
     3     3
```

Si queremos guardar las dimensiones en las variables `n` y `p`, usaremos la instrucción `[n,p]=size(x).`

```
»[n,p]=size(x)
n =
     3
p =
     3
```

### *P1.3 Operadores y caracteres especiales*

Una vez definidas las matrices, están dispuestas para la realización de operaciones con las mismas. Varios son los operadores disponibles, algunos realizan las operaciones matriciales estándar mientras que otros tienen algunas particularidades que expondremos a continuación. La tabla siguiente contiene los operadores habituales.

| Operador | Name | HELP topic |
|---|---|---|
| + | Mas | arith |
| - | Menos | arith |
| * | Multiplicación de matrices | arith |
| ^ | Matrix power | arith |
| \ | Backslash or left division | slash |
| / | Slash or right division | slash |

Los operadores realizan las operaciones habituales con matrices, que pueden ser asignadas a una nueva matriz, por ejemplo, `c=a*b` calcularía el producto de `a` por `b` y lo pondría en `c`. Los operadores de división por la izquierda y por la derecha calculan primero la inversa de la matriz a la izquierda o a la derecha y realizan después la multiplicación. Por ejemplo `x/y` es lo mismo que `x * inv(y)` donde `inv` es una función que calcula la inversa de la matriz `y`.

Si al operador le precede un punto, la operación correspondiente se realiza elemento a elemento. Por ejemplo, para la operación `z = x./y` tendríamos que `c(i,j)=x(i,j)/y(i,j).` Lo mismo ocurriría para los productos y las potencias.

Los siguientes operadores lógicos proporcionan resultados 1 ó 0 dependiendo de si la comparación es verdadera o falsa.

```
   ==     Equality                    relop
   < >    Relational operators        relop
   &      Logical AND                 relop
   |      Logical OR                  relop
   ~      Logical NOT                 relop
   xor    Logical EXCLUSIVE OR        xor
```

Obsérvese que el operador de igualdad son dos signos iguales para distinguirlo del operador de asignación utilizado antes.

### *P1.4 Los M-files*

Lo que llamaremos M-files son archivos de texto que contienen instrucciones ejecutables de Matlab. Este tipo de archivos permite la realización de programas completos para aplicar en diversas situaciones. Se seleccionan en el menú File opción New M-file, de forma que aparece un sencillo editor de textos que nos permite escribir y manipular las instrucciones.

Al guardar los archivos de este tipo, han de terminar con la extensión .m para que el programa los reconozca como archivos de instrucciones. Los nombres no contendrán caracteres especiales o caracteres que puedan ser confundidos por el programa con operadores, etc. Para ejecutar las instrucciones seleccionar la opción Save and Execute del menú File.

El m-file puede incluir cualquier tipo de instrucción de la misma forma en que se utilizan en la ventana de comandos del programa.

Si la línea de instrucciones comienza con un % el programa interpretará que se trata de un comentario y no la ejecuta. Esta opción es útil para escribir comentarios sobre la construcción del programa o para suspender

temporalmente la ejecución de alguna línea del mismo en las pruebas de funcionamiento.

Si antes de comenzar las instrucciones ejecutables de un programa se escriben varias líneas de comentarios, están podrán ser utilizadas como ayuda en la ventana de comandos.

### *P1.5 Funciones de entrada/salida de datos dentro de un programa*

Algunas funciones relacionadas con la entrada y con la salida de los datos y resultados son particularmente útiles.

Es posible introducir algunos datos necesarios en el desarrollo del programa desde el teclado mediante la instrucción `input('cadena de caracteres')` espera a que el usuario introduzca un número desde el teclado y presione la tecla return. Por ejemplo, si en un programa para biplot deseamos pedir al usuario que introduzca el numero de ejes que desea para la representación, podríamos hacerlo mediante la instrucción

```
input('Deme el numero de ejes para la representacion')
```

Si el valor que queremos introducir por el teclado es una cadena de caracteres la instrucción es `input('cadena de caracteres', 's')`.

La instrucción `disp(var)` muestra en la ventana de comandos la matriz especificada en `var`. Sirve para mostrar, por ejemplo, los resultados de cálculos intermedios en un programa.

Al final tiene un m-file completo que construye una representación biplot.

### *P1.6 Las funciones definidas por el usuario*

Las funciones definidas por el usuario son en realidad M-files con una estructura especial. La diferencia es que permiten la utilización de parámetros en la llamada desde el programa. Comienzan siempre con la palabra `function` seguida de los nombres de la matrices que contienen los resultados separados por comas y entre corchetes, el signo igual, el nombre de la función y los parámetros separados por comas y entre paréntesis. Por ejemplo, la función siguiente

```
function [a,b,inercia]=biplot(x,trans,alfa,neje,label1,label2,titulo)
```

es una función denominada biplot que depende de 7 parámetros (x, trans, alfa, neje, label1, label2, titulo) y que produce como resultados la matrices a, b e inercia.

### *P1.7 Las instrucciones de condición*

En muchos programas es necesario tomar decisiones entre distintas alternativas dependiendo de si se verifica o no una condición. Por ejemplo, cual es la transformación inicial que aplicaremos a los datos en la construcción de un biplot. Esto lo haremos mediante la sentencia if

```
IF condición
  grupo de instrucciones1;
ELSE
  grupo de instrucciones2;
END
```

Si la condición se verifica se ejecuta el grupo de instrucciones 1 y si no se ejecuta el grupo 2. la instrucción ha de terminar siempre en un end para que el programa sepa cual es el conjunto de instrucciones a ejecutar. El grupo 2 puede

omitirse y eliminar el ELSE de forma que si la condición no se verifica, no se toma ninguna acción.
Tras el ELSE puede definirse un nuevo IF con lo que se tendría un conjunto de condiciones encadenadas.
La condición es generalmente el resultado de expresión operador expresión donde los operadores son == (igual), < (menor), > (mayor), <= (menor o igual), >= (mayor o igual), o ~= (distinto).

### *P1.8 Las instrucciones de repetición*

En algunos casos es necesario repetir un grupo de instrucciones un cierto número de veces, por ejemplo, para todos los elementos de una matriz de datos y hacer operaciones con cada uno de ellos que no pueden realizarse con operaciones matriciales.

La instrucción para repetir un número predefinido de veces es la instrucción `for`. Veámoslo con un ejemplo, la instrucción siguiente repetiría `a(i) = 1` desde que `i` vale 1 hasta que valga `n`, incrementando en una unidad el valor de `i` en cada caso. Es decir crearía un vector de n componentes todas iguales a 1. Tiene el mismo efecto que `ones(n)`.

```
for i = 1:n,
   a(i) = 1;
end
```

Es posible anidar varios bucles `for` para recorrer, por ejemplo, todos los elementos de una matriz. El siguiente ejemplo construye una matriz de `n` filas y `p` columnas con todos los elementos iguales a 1, es decir tiene el mismo efecto que `ones(n,p)`.

```
for i = 1:n,
    for j = 1:n,
        a(i,j) = 1;
    end
end
```

### *P1.9 La descomposición en valores singulares*

Es posible calcular la descomposición en valores singulares mediante la función `[u,d,v] = svd(x)` que produce los vectores singulares por la izquierda en **U**, los valores singulares en a diagonal principal de **D** y los vectores singulares por la derecha en la matriz **V**.
Por si mismo `svd(x)` devuelve solamente los valores singulares. Mediante la forma `[u,d,v] = svd(x,0)` se obtiene la descomposición mínima, es decir si **X** es nxp con n>p, entonces se calculan solo las p primeras columnas de **U** y **D** es nxn.

Otra posible descomposición es la siguiente: Si el rango de la matriz es r, calcular solo las primeras r columnas de **U** y **V**, de forma que **D** es rxr. Mediante instrucciones de MATLAB el cálculo podría realizarse de la siguiente forma.

```
[u,d,v]=svd(x,0);
r=rank(x);
d=d(1:r, 1:r);
u=u(:,1:r);
v=v(:,1:r);
```

Como ya habrá adivinado, `rank(x)` es una función que calcula el rango de X.

### *P1.10 Diagramas de dispersión en Matlab*

Los diagramas de dispersión en Matlab se llevan a cabo mediante el comando `plot`.
`PLOT(X,Y)` dibuja el vector X frente al vector Y. Pueden obtenerse diferentes símbolos, colores y tipos de líneas añadiendo un argumento con 1, 2 ó 3

caracteres entre comillas que indican el color, el carácter y el tipo de línea. `PLOT(X,Y,S)`.

```
y     yellow      .     point
m     magenta     o     circle
c     cyan        x     x-mark
r     red         +     plus
g     green       -     solid
b     blue        *     star
w     white       :     dotted
k     black       -.    dashdot
                  --    dashed
```

Por ejemplo, `PLOT(X,Y,'c+')` dibuja los puntos de color cyan con un signo +.

El comando `text(X,Y,label)` dibuja los caracteres contenidos en label en las posiciones de los puntos definidas por X e Y. Ha de utilizarse después del comando plot ya que por si mismo no es capaz de dibujar los ejes de la representación.

Los comandos `xlabel(cadena), ylabel(cadena) y title(cadena)` permiten poner leyendas a los ejes y un título general para el gráfico.

El comando `axis` permite el cambio del aspecto de los ejes. Nos limitamos aquí a una de sus opciones que permite el dibujar la misma escala en los dos ejes de forma que las distancias sean realmente interpretables. La forma es la siguiente:

```
 AXIS Plot axis scaling and appearance.
  AXIS([XMIN XMAX YMIN YMAX]) coloca la escala de los ejes x e y en el
gráfico actual.

  AXIS('auto') pone el escalado de los ejes por defecto al modo
automático, donde xmin = min(x), xmax = max(x), etc.

  V = AXIS devuelve un vector fila que contiene las escalas del
gráfico actual.

  AXIS('equal') coloca los ejes de forma que los incrementos en el eje
X sean iguales a los incrementos en el eje Y. Esto implica que son
intepretables ángulos y relaciones de perpendiculariadad, que se
mantienen sobre la impresión en papel.
```

```
  AXIS('square') hace que los ejes sean cuadrados.

  AXIS('normal') reestablece las características de los ejes y elimina
las restricciones. Deshace los efectos de AXIS('square') and
AXIS('equal').
```

## P2. PROGRAMA PARA LA CONSTRUCCION DE UNA REPRESENTACION BIPLOT

Presentamos a continuación un programa para la construcción de una representación Biplot. El programa exige que previamente a su utilización hayamos construido un m-file que contenga la matriz de datos con el nombre `x`, los vectores de etiquetas para filas y columnas denominados `label1` y `label2` respectivamente y una cadena de caracteres denominada `titulo` que se utilizara como identificación del estudio. Se recuerda que las etiquetas han de tener todas la misma longitud. Conviene comenzar el M-file de los datos con la orden `clear` para borrar el espacio de memoria de trabajo y evitar cruces con análisis anteriores.

La matriz de datos puede pegarse directamente de programas como el Statview, pero teniendo en cuenta que el Matlab solo acepta puntos como separador decimal. El programa tiene la opción de reemplazar en el archivo de texto que permite pasar de unos a otros.

Se trata de una primera versión simple del programa, que puede complementarse con otra utilidades. Obsérvese que la mayor parte de las instrucciones están dirigidas a las transformaciones y al aspecto de las salidas, ya que el cálculo propiamente dicho se realiza de forma muy simple.

En el programa no se han detallado todos los procedimientos, solamente se trata de un listado con algunos comentarios sobre el desarrollo de las instrucciones.

```
% Biplot    biplot Graphical display
% Disponga la matriz de datos en x (minuscula) y dos matrices
```

```
% label1 y label2 con las etiquetas de filas y columnas
% respectivamente. Una cadena de caracteres denominada
% titulo permite titular todos los graficos

%-------------------------(Calculo de las dimensiones de la matriz)-----------------------
[n,p]=size(x);

%-------------------- (Vectores con números de oden de filas y columnas) ------------------
nfilas=(1:n)'; ncols=(1:p)';

%----------------(Calculo  de las transformaciones previas en los datos) --------------------

disp('     ');disp('     ');disp('     ');
disp('1- Restar media global      2- Doble centrado');
disp('3- Restar media de columnas 4-Estandarizar por columnas');
disp('5- Restar media de filas    6-Estandarizar por filas');
disp('7- Datos brutos');

trans = input('Seleccione una opcción y presione ENTER :');

if trans ==1,
  media=sum(sum(x))/(n*p); x=x-media;
end;

if trans ==2,
  x=(eye(n)-ones(n)/n)*x*(eye(p)-ones(p)/p);
end;

if trans ==3,
  x=(eye(n)-ones(n)/n)*x;
end;

if trans ==4,
  x=(eye(n)-ones(n)/n)*x; S=diag((x'*x)/n); x=x*inv(sqrt(diag(S)));
end;

if trans ==5, x=x*(eye(p)-ones(p)/p); end;

if trans ==6,
  x=x*(eye(p)-ones(p)/p); S=diag((x*x')/p); x=inv(sqrt(diag(S)))*x;
end;

%--------------------(Constante alfa para el cálculo del biplot)-------------------------
disp('     ');disp('     ');disp('     ');
disp('Deme el valor de alfa (0=GH, 1=JK, 0.5=sqrt, >1=HJ)');
disp('Los   marcadores   son   reescalados   manteniendo   el   producto
escalar');
disp('para   clarificar   la   representación   si   selecciona   escalado
óptimo');

alfa = input('Escriba el valor y presione ENTER :');
```

```
%-----------------------(Número de ejes de la representación)-----------------------
neje = input('Deme el numero de ejes de la represntacion :');

%-------------------------(Cálculo de la DVS para el biplot)--------------------------
[u,d,v]=svd(x,0);
r=rank(x);
d=d(1:r, 1:r);
u=u(:,1:r);
v=v(:,1:r);

%---------------------(Cálculo de las contribuciones para el biplot)-----------------
a=u*diag(diag((d)));
b=v*diag(diag((d)));
sf=sum((a.^2)')';
cf=(inv(diag(sf))*(a.^2))*1000;
cf=cf(:,1:neje);

sc=sum((b.^2)')';
cc=(inv(diag(sc))*(b.^2))*1000;
cc=cc(:,1:neje);

%-----------------------------(Impresión de la inercia explicada)-------------------
disp(titulo)
valprop=diag(d.^2);
iner=(diag(d.^2)/sum(diag(d.^2)))*100;
inercia=cumsum(iner);
disp('Inercia absorbida'); disp(iner);
disp('Inercia acumulada'); disp(inercia);

%----------------------------(Impresión de los resultados- opcional)----------------
resul=input('desea todos los resultados (s/n) : ','s');

if resul=='s',

tit='contribuciones relativas del factor al elemento (filas)';
disp(tit); disp([nfilas,cf]);
tit='contribuciones relativas del factor al elemento (columnas)';
disp(tit); disp([ncols,cc]);

tit='contribuciones relativas acumuladas del factor al elemento
(filas)';
disp(tit);
cfa=cumsum(cf')';
disp([nfilas,cfa]);

tit='contribuciones relativas acumuladas del factor al elemento
(columnas)';
disp(tit);
cca=cumsum(cc')';
disp([ncols,cca]);
end;
```

```
%--------------------(Cálculo de las coordenadas para el biplot )-------------------------
if alfa<=1,
a=u*diag(diag((d.^alfa)));
b=v*diag(diag((d.^(1-alfa))));
end;

a=a(:,1:neje);
b=b(:,1:neje);

%-------------------------(Reescalado para mejorar el aspecto )---------------------------
escala=input('Desea reescalado optimo de las coordenadas (s/n) :
','s');

if escala == 's',
sca = 0;
scb=0;

sca=sum(sum(a.^2));
scb=sum(sum(b.^2));

sca=sca/n;
scb=scb/p;
scf=sqrt(sqrt(scb/sca))
a=a*scf;
b=b/scf;

end;

%--------------------(Cálculo de las medias para situar en el biplot )-----------------------
mf=mean(a)
mc=mean(b)

%-----------------------------(Representaciones gráficas )---------------------------------
for i=1:neje-1
  for j=i+1:neje
figure;
plot(a(:,i),a(:,j),'+',b(:,i),b(:,j),'*',mf(i),mf(j),'o',mc(i),mc(j),'
o')
text(a(:,i),a(:,j),label1);
text(b(:,i),b(:,j),label2);
str=['ejes ' num2str(i) ' y ' num2str(j)];
xlabel(str);
axis('equal');
title(titulo)
   end;
end;

gfit=(1-sum(sum(((x-a*b').^2)))/sum(sum(((x).^2))))*100
```

La característica más importante del programa es que calcula las contribuciones siempre de la misma manera, independientemente del tipo de biplot seleccionado. Esto es así porque las calidades de representación, en un HJ-Biplot, coinciden con la bondad de ajuste separada por filas o columnas en los biplots CMP y RMP. La diferencia en la interpretación es que en la forma HJ interpretamos la proyección sobre los ejes y en las otras la proyección sobre las variables originales.

El programa podría completarse con características adicionales para mejorar la presentación, por ejemplo, colores para distintas filas o columnas, crear etiquetas cuando el usuario no las proporciona por defecto, seleccionar puntos cercanos para clarificar el gráfico, etc...

## P3. EJEMPLO

Consideremos el siguiente ejemplo, adaptado de GABRIEL y ODOROFF (1990) en el que consideramos tres variables, el número de nidos de cigüeña, la tasa de nacimientos y el consumo de electricidad per capita.

| Año | Nidos de Cigüeñas | Tasa de Nacimientos | Consumo de Electricidad |
|---|---|---|---|
| 53 | 177 | 17,9 | 701 |
| 54 | 210 | 17,3 | 779 |
| 55 | 217 | 17,3 | 849 |
| 56 | 214 | 17,2 | 867 |
| 57 | 199 | 16,8 | 909 |
| 58 | 186 | 16,5 | 962 |
| 59 | 165 | 16,3 | 1042 |
| 60 | 145 | 16,6 | 1153 |
| 61 | 135 | 16,6 | 1270 |
| 62 | 154 | 16,7 | 1435 |
| 63 | 121 | 17,6 | 1584 |
| 64 | 111 | 17,7 | 1725 |
| 65 | 106 | 18,0 | 1916 |
| 66 | 101 | 18,4 | 2069 |
| 67 | 87 | 16,8 | 2227 |
| 68 | 73 | 15,3 | 2438 |
| 69 | 65 | 14,6 | 2709 |
| 70 | 60 | 14,4 | 3197 |
| 71 | 54 | 15,2 | 3358 |
| 72 | 51 | 15,2 | 3667 |
| 73 | 45 | 14,3 | 3763 |
| 74 | 40 | 14,1 | 3701 |
| 75 | 39 | 14,2 | 3871 |
| 76 | 37 | 12,9 | 4289 |
| 77 | 35 | 12,2 | 4533 |

Tabla 1: Datos tomados de GABRIEL y ODOROFF (1990)

El primer paso consiste en preparar el archivo de datos de forma que se pueda entender en Matlab. Comenzamos con la palabra `clear` para limpiar todas las variables usadas anteriormente. Añadimos una variable denominada `titulo` que contiene una cadena de caracteres con información sobre el trabajo. Pueden añadirse todas las líneas de comentario que se deseen siempre que vayan precedidas del carácter %.

Los datos se colocan en una matriz x. Es posible pegarlos directamente si estaban en otro programa o leerlos de un archivo "solo texto" creado por

cualquier otro programa. Recuérdese que los datos de la misma fila van separado por espacios en blanco (pueden ser tabuladores si los datos estaban en un archivo de texto separado por tabuladores, o se copiaron directamente de una hoja de cálculo de cualquier paquete estadístico), y las distintas filas van separadas por un ; o una vuelta de carro.

```
clear

titulo='Nidos, Nacimientos y Consumo de Electricidad en Dinamarca'

x=[177      17.9  701
210   17.3  779
217   17.3  849
214   17.2  867
199   16.8  909
186   16.5  962
165   16.3  1042
145   16.6  1153
135   16.6  1270
154   16.7  1435
121   17.6  1584
111   17.7  1725
106   18.0  1916
101   18.4  2069
87    16.8  2227
73    15.3  2438
65    14.6  2709
60    14.4  3197
54    15.2  3358
51    15.2  3667
45    14.3  3763
40    14.1  3701
39    14.2  3871
37    12.9  4289
35    12.2  4533];

label1=['53'; '54';'55';'56';'57';'58';'59';'60';'61';'62';'63';'64'
'65';'66';'67';'68';'69';'70';'71';'72';'73';'74';'75';'76';'77'];
label2=['nidos'; 'Nacim'; 'Elect'];
```

Las etiquetas para filas y columnas se colocan en vectores denominados y respectivamente. Todas las etiquetas tendrán el mismo numero de caracteres e irán colocadas entre comillas simples.

Una vez que el archivo de datos está listo (en el m-file correspondiente), se ejecuta el m-file y después el que contiene las instrucciones del biplot.

La secuencia de resultados es la siguiente. Se ha seleccionado como transformación el estandarizado de los datos por columnas, debido a que las escalas de medida son muy diferentes.

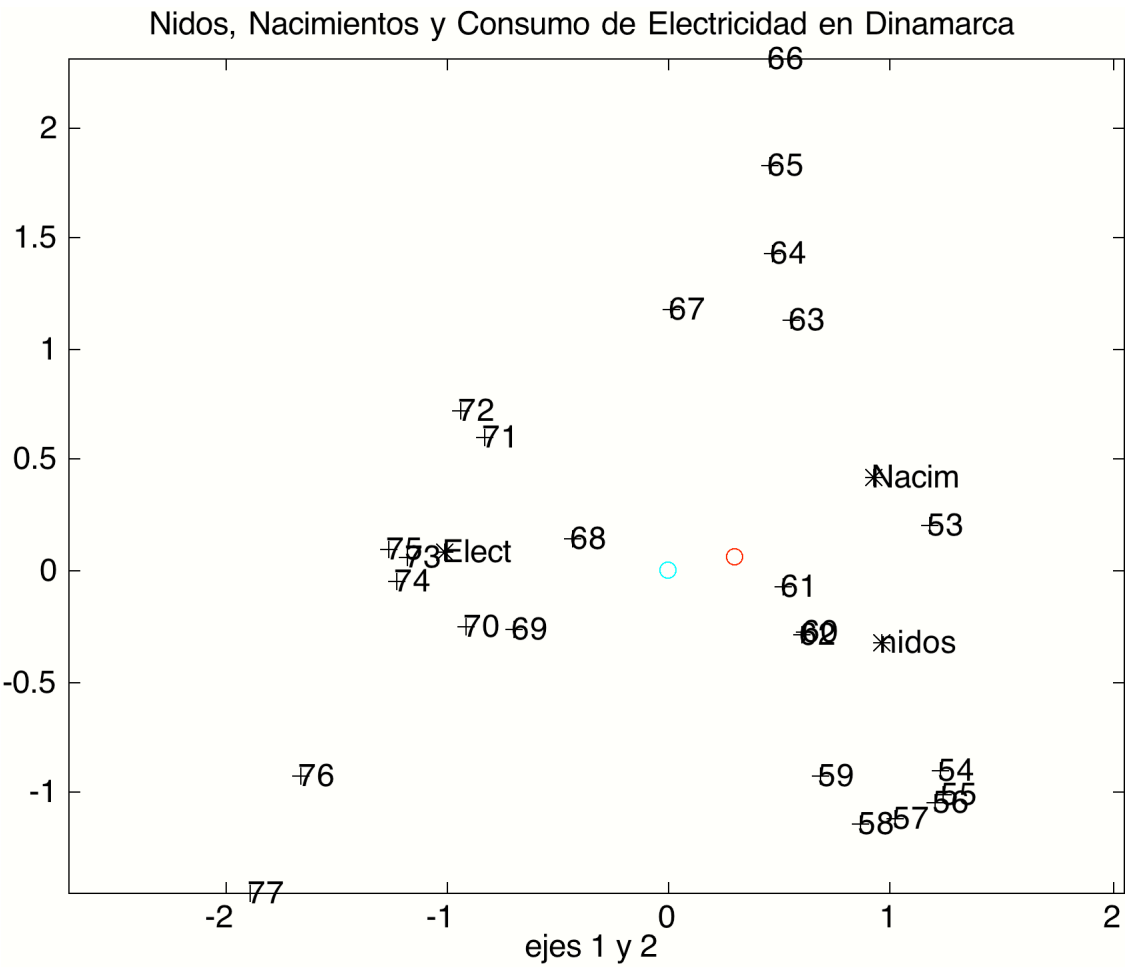
```
titulo =

Nidos, Nacimientos y Consumo de Electricidad en Dinamarca



1- Restar media global      2- Doble centrado
3- Restar media de columnas 4-Estandarizar por columnas
5- Restar media de filas    6-Estandarizar por filas
7- Datos brutos

Seleccione una opcción y presione ENTER :4


Deme el valor de alfa (0=GH, 1=JK, 0.5=sqrt, >1=HJ)
Los marcadores son reescalados manteniendo el producto escalar
para clarificar la representación si selecciona escalado óptimo

Escriba el valor y presione ENTER :0

Deme el numero de ejes de la represntacion :2
Nidos, Nacimientos y Consumo de Electricidad en Dinamarca
Inercia absorbida
   89.4377
    9.4466
    1.1157

Inercia acumulada
   89.4377
   98.8843
  100.0000


desea todos los resultados (s/n) : s
contribuciones relativas del factor al elemento (filas)
    1.0000  995.9666    3.0847
    2.0000  938.0427   54.4489
    3.0000  916.9957   65.0164
    4.0000  911.3582   73.5305
    5.0000  884.8379  113.1372
    6.0000  845.1714  153.7294
    7.0000  814.4270  154.8191
    8.0000  910.0944   19.8407
    9.0000  893.0701    1.7420
```

```
   10.0000  976.0134   23.9765
   11.0000  687.3319  310.5983
   12.0000  506.0078  493.1123
   13.0000  360.8157  633.3628
   14.0000  265.4700  713.2816
   15.0000    0.5043  983.7042
   16.0000  823.1649   10.0458
   17.0000  907.0892   14.0257
   18.0000  986.8357    7.7265
   19.0000  945.0290   52.4233
   20.0000  920.1633   57.8520
   21.0000  997.3862    0.2385
   22.0000  999.7474    0.1385
   23.0000  997.3704    0.6436
   24.0000  966.7441   31.4078
   25.0000  939.4602   58.6916

contribuciones relativas del factor al elemento (columnas)
    1.0000  886.9499  102.7829
    2.0000  822.8873  174.5187
    3.0000  973.2949    6.0957

contribuciones relativas acumuladas del factor al elemento (filas)
    1.0000  995.9666  999.0513
    2.0000  938.0427  992.4916
    3.0000  916.9957  982.0121
    4.0000  911.3582  984.8887
    5.0000  884.8379  997.9750
    6.0000  845.1714  998.9007
    7.0000  814.4270  969.2461
    8.0000  910.0944  929.9351
    9.0000  893.0701  894.8121
   10.0000  976.0134  999.9899
   11.0000  687.3319  997.9302
   12.0000  506.0078  999.1201
   13.0000  360.8157  994.1784
   14.0000  265.4700  978.7517
   15.0000    0.5043  984.2085
   16.0000  823.1649  833.2107
   17.0000  907.0892  921.1148
   18.0000  986.8357  994.5622
   19.0000  945.0290  997.4522
   20.0000  920.1633  978.0153
   21.0000  997.3862  997.6246
   22.0000  999.7474  999.8859
   23.0000  997.3704  998.0141
   24.0000  966.7441  998.1520
   25.0000  939.4602  998.1518

contribuciones relativas acumuladas del factor al elemento (columnas)
    1.0000  886.9499  989.7327
    2.0000  822.8873  997.4060
    3.0000  973.2949  979.3906


Desea reescalado optimo de las coordenadas (s/n) : n
```

```
mf = 0 0

mc =    0.2934    0.0596
```

El 98.88% de a variabilidad de los datos estandarizados es absorbido por los dos primeros ejes de la representación, lo que indica que la aproximación es buena y puede hacerse una interpretación adecuada en sólo dos dimensiones.

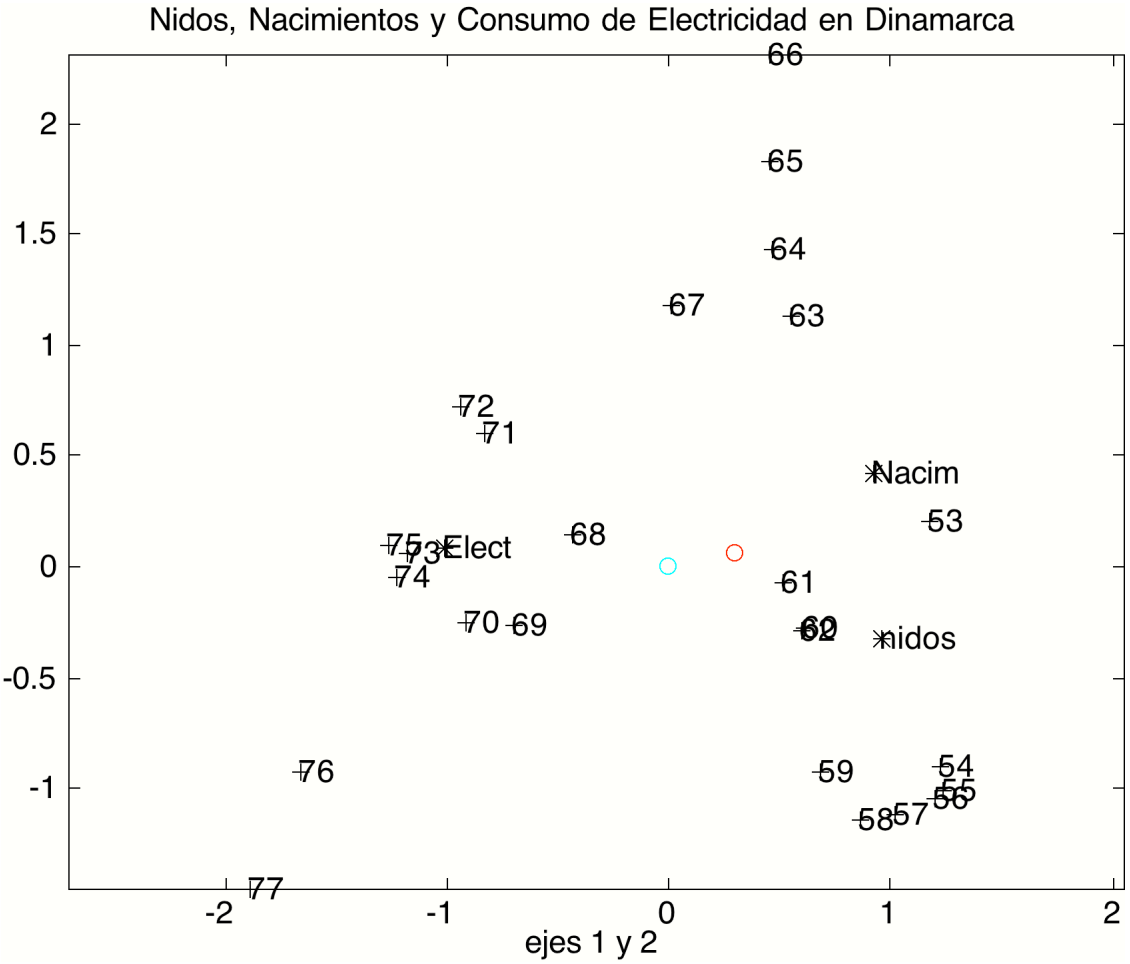
El gráfico obtenido, en dos dimensiones, para la representación HJ-Biplot aparece en la figura siguiente.




Generalmente el gráfico necesita algunos retoques en algún programa de dibujo, que permita separar las etiquetas y clarificar el gráfico. Esto es lo que se hizo en la introducción a la teoría donde, además puede consultarse la interpretación.

# PRACTICAS CON MATLAB

## (elementos necesarios para la construcción de un Biplot)




José Luis Vicente Villardón
Departamento de Estadística
Universidad de Salamanca

# PRACTICA CON MATLAB