Revista de Matemática: Teoría y Aplicaciones 1998 **5**(2) : 133–147
cimpa – ucr – ccss issn: 1409-2433

ACCPLOT: gráficos del acp con matemática

Carlos L. Arce S.*

Recibido: 19 marzo 1998

Resumen

`ACPPlot` es un comando para construir los gráficos propios del Análisis en Componentes Principales (ACP), planos principales y círculos de correlación; en ambos casos, agregando opciones para unir los puntos en trayectorias, agrupar conjuntos de puntos, etiquetar y en general pulir la presentación de los gráficos. Su desarrollo utiliza el lenguaje *Mathematica* y representa un ejemplo de aplicación del gran potencial de *Mathematica* en la investigación de los métodos matemáticos del análisis de datos. Paralelamente se muestran también, las facilidades que da *Mathematica* para hacer los cálculos del ACP y del Análisis Factorial de Correspondencias.

Palabras clave: análisis en componentes principales, gráficos, planos principales, software, Mathematica.

Abstract

`ACPPlot` is a command for creating graphics for Principal Component Analysis (PCA), principal planes and correlations circles; in both cases, adding options for joining points with trajectories, clustering points, labeling and for improving the general presentation of graphics. It uses *Mathematica* language and it represents an example of application of the capacities of *Mathematica* in the investigation of mathamatical methods for data analysis. There are also shown the facilities given by *Mathematica* for computations in PCA and Correspondence Analysis.

Keywords: principal component analysis, graphics, principal planes, software, Mathematica.

AMS Subject Classification: 62-07, 62H25, 68U05

*PIMAD, Escuela de Matemática, Universidad de Costa Rica, 2060 San José Costa Rica; Tel. +(506) 207 4534, Fax: +(506) 207 4397; E-mail: carce@cariari.ucr.ac.cr

Introducción

A continuación se presenta una descripción de la sintaxis de `ACPPlot`, incluyendo las opciones que lo diferencian de otros procedimientos de *Mathematica* que construyen gráficos.

1 Sintaxis del procedimiento `ACPPlot`

La especificación de las coordenadas de los puntos a graficar se puede hacer en dos formas:
a) como una sola lista de puntos, {{x1,y1},{x2,y2},...}:

```
ACPPlot[{{x1,y1},{x2,y2},...}, opciones]
```

o, b) como una lista que contiene sublistas de puntos, definiendo una estructura de grupos:

```
ACPPlot[{{{x1,y1},{x2,y2},...},
         {{u1,w1},{u2,w2},...},...}, opciones]
```

La palabra `opciones`, supone ninguna, una o muchas especificaciones para cambiar características del gráfico, en el estilo usual en que lo hacen los comandos de graficación en *Mathematica*, es decir, como reglas de sustitución. Por ejemplo `Axes->False` o `PlotStyle->{Rojo,Azul}`. Adelante se explican en detalle.

2 Ejemplo 1: ACP

Para ilustrar las posibilidades que ofrece `ACPPlot`, se elaborarán gráficos que provienen de un ACP sobre mediciones de algunos indicadores de la calidad del agua en varios ríos que surten el agua a la Planta Hidroeléctrica La Garita y puntos en el mismo embalse de La Garita.

Las siguientes órdenes para *Mathematica* definen la tabla de datos, sobre las mediciones de las variables en estudio y establecen la etiquetas para las variables y los puntos de muestreo:

```
dat = {{55.0,15.0,7.15,172.0,0.54},
       {58.0,13.7,7.20,164.0,0.52},
       {58.0,30.6,7.30,180.0,0.68},
       {59.0,10.0,7.15,157.0,0.55},
       {58.0,10.5,7.20,165.0,0.66},
       {67.0,16.2,8.30,165.0,0.40},
       {70.0,6.1,8.35,187.0,0.62},
       {77.0,1.5,8.35,192.0,0.64},
       {62.0,15.0,8.10,295.0,0.92}};
   EtiIndic = {"Calidad","DBO","PH","Sólidos","Fosfatos"};
EtiRios = {"Centro","Orilla","Represa","Salida","Desfogue",
      "Alajuela","Ciruelas","Tizate","Virilla"};
```

A continuación se presentan las órdenes para calcular el ACP, con métricas $M = D_{1/\sigma^2}$, inversas de varianzas y $Dp = D_{1/n}$:

```
   <<varcor.ma
      (* Definición de métricas *)
```

```
M = DiagonalMatrix[1/Varianza[dat]];
Dp = DiagonalMatrix[Table[1/9,{9}]];
      (* Centraje de datos y definición de V y VM *)
X = Centra[dat];
V = Transpose[X].Dp.X;
VM = V.M;
      (* Cálculo de valores y vectores propios de VM *)
{landas,Ut} = Eigensystem[VM];
      (* Ejes principales u1,...,up M-ortononormalizados *)
U = Transpose[Normaliza[Ut,M]];
Y = X.M.U; (* Componentes principales *)
      (* Matriz de correlaciones entre variables y comp.  principales *)
R = Sqrt[M].U.DiagonalMatrix[Sqrt[landas]];
```

Se ha evitado mostrar los resultados parciales de cada una de las órdenes anteriores, con el propósito de resumir el procedimiento, pero estos se presentan en las dos siguientes órdenes, que definen las coordenadas de los individuos `Rios`, en el primer plano principal del ACP y las coordenadas para las variables `Indic`, en el respectivo plano principal del espacio de las variables:

```
Rios = Columna[Y,{1,2}]
```

$$\{\{-1.56388, 0.0112841\}, \{-1.31758, -0.414592\}, \{-1.57427, 1.38505\},\\ \{-1.11687, -0.592589\}, \{-0.877952, 0.101833\}, \{0.2519, -1.51597\},\\ \{1.68507, -0.811724\}, \{2.56748, -1.19951\}, \{1.9461, 3.03622\}\}$$

```
Indic = Columna[R,{1,2}]
```

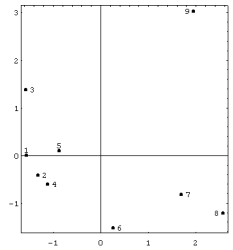
$$\{\{0.864503, -0.423917\}, \{-0.584565, 0.504383\}, \{0.908118, -0.166334\},\\ \{0.596772, 0.771403\}, \{0.412486, 0.853414\}\}$$

La información en la variables `Rios` e `Indic` son los datos mínimos requeridos para construir un gráfico con `ACCPPlot`:

```
ACPPlot.ma
```

Esta orden activa el procedimiento `ACPPlot` que se supone definido en el archivo `ACPPlot.ma` presente en el directorio `C:\wnmath22`.

```
ACPPlot[Rios];
```



3 Opción Etiquetas

Con la opción `Etiquetas`, se aporta la lista de etiquetas asociadas a los puntos que se grafican. `ACPPlot` trata de colocarlas alrededor de los respectivos puntos sin que se produzcan sobreposiciones. Los posibles valores para esta opción son los siguientes:

`Etiquetas->Ninguna:` No hay etiquetas para los puntos.

`Etiquetas->Automatic:` Se generan etiquetas de la forma `"1", "2", "3", ...` para cada grupo de puntos y los coloca automáticamente, tratando de evitar, en lo posible, la sobreposición de etiquetas. Es el valor establecido por omisión para `Etiquetas`.

`Etiquetas->{"Eti1","Eti2",...}`

`Etiquetas->{{"EtiA1","EtiA2",...}, {"EtiB1","EtiB2",...},...}`
Cada etiqueta debe ser del tipo `String`, (encerradas entre comillas dobles) y su número debe coincidir con el número de puntos a graficar. En este caso, la posición de cada etiqueta alrededor del punto se determina automáticamente, tratando de evitar sobreposiciones, hasta donde sea posible. Pueden aparecer agrupadas en sublistas o no.

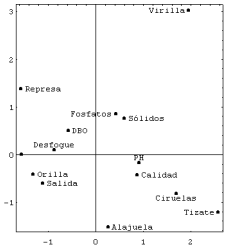
`Etiquetas->{{"Eti1",pos1},{"Eti2",pos2},...}:` Se dan las etiquetas y su posición relativa alrededor del punto donde se deben colocar, utilizando las palabras `Derecha, Izquierda`, `Arriba` o `Abajo`. Por lo tanto, no hay colocación automática de etiquetas.

`Etiquetas->{{"EtiTray1",nodo1,pos1},{"EtiTray2",nodo2,pos2},...}:` Esta es la forma de etiquetar trayectorias, cuando los puntos se dan agrupados, y se utiliza conjuntamente con la opción `PlotJoined->True`. Supone una especificación de etiqueta por cada grupo de puntos o trayectoria, la cual tiene la forma de un triplete con: a) la etiqueta, b) el punto de la trayectoria donde se colocará la etiqueta, (indicado como un entero, comenzando con 1, para identificar el primer punto) y c) la posición alrededor del punto donde aparecerá la etiqueta (`Derecha, Izquierda`, `Arriba` o `Abajo`.) El número de etiquetas debe ser igual al nómero de grupos o trayectorias. En la página **??** se presenta un ejemplo de esta forma de dar etiquetas.

3.1 Ejemplo

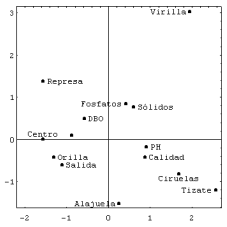
Las variables `Rios`, `Indic`, `EtiRios` y `EtiIndic` se definieron en el ejemplo 1 (sección **??**) en la página **??**.

```
ACPPlot[{Rios,Indic},Etiquetas->{EtiRios,EtiIndic}];
```



Observe que en el anterior gráfico, no aparece la etiqueta del punto $\{-1.5638, 0.0112\}$, `Centro`. El algoritmo de colocación automática de `ACPPlot`, no encontró espacio para colocarla. Sin embargo, con la opción `PlotRange` se puede redefinir el intervalo del eje x, y con esto agregar más espacio horizontal a la izquierda del gráfico, a fin de que la etiqueta `Centro` pueda ser colocada:

```
ACPPlot[{Rios,Indic},Etiquetas->{EtiRios,EtiIndic},
      PlotRange->{{-2.2,2.7},Automatic}];
```



Si no se está conforme con el resultado de la distribución automática de etiquetas, se puede especificar la ubicación deseada. En la página **??** se presenta un ejemplo donde se hace esto.

4 Distinga

Con esta opción se controla la formación de grupos en la lista de los puntos, cuando estos se han dado como una sola lista, a fin de distinguirlos con colores o construir trayectorias uniéndolos con líneas mediante la opción `PlotJoined->True`.

`Distinga->{n1,n2,...,nk}`: Si los datos forman una única lista con n puntos, los distingue particionando la lista en k grupos de tamaño `n1, n2,..., nk` respectivamente y respetando el orden de los puntos en la lista, esto siempre que $n1 + n2 + \cdots + nk = n$. Cuando $n1 + n2 + \cdots + nk < n$, construye $k + 1$ grupos, formando el grupo $k + 1$ con los últimos $n - (n1 + n2 + \cdots + nk)$ puntos de la lista.

`Distinga->Automatic`: Es el valor por omisión de `Distinga` y respeta la agrupación explícita en la lista de puntos, si esta existe. Cuando hay una sola lista de puntos y la opción `PlotStyle` especifica varios estilos (para puntos o líneas), entonces se genera una partición de puntos en grupos del mismo tamaño (salvo tal vez el último), que se determina como la cantidad de puntos divida entre el número de estilos dados.

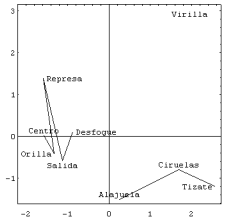
4.1 Ejemplo

La siguiente orden, con la opción `Distinga->{5,3}`, agrupa los 9 puntos contenidos en la variable `Rios` del ejemplo 1 en 3 grupos, como lo muestra el gráfico.

```
ACPPlot[Rios,Distinga->{5,3},Etiquetas->EtiRios2,
      PlotRange->{{-2.2,2.7},Automatic}];
```

Como la distinción de los grupos se hace con colores y en estas notas no es posible reproducirlos, entonces se ha omitido el gráfico que de todas formas tendría una apariencia similar al de la página **??**. En su lugar, se agrega la opción `PlotJoined->Si` —que se veráá más adelante— para unir los puntos de cada grupo con una línea recta, y hacer evidente la agrupación obtenida con `Distinga->{5,3}`.

```
ACPPlot[Rios,Distinga->{5,3},Etiquetas->EtiRios3,
  PlotRange->{{-2.2,2.7},Automatic},PlotJoined->Si];
```



Este ejemplo sólo ilustra un recurso de `ACPPlot` y no pretende sugerir que las trayectorias anteriores tengan alguna interpretación interesante. Por otra parte, las etiquetas fueron reubicadas, redefiniendo `EtiRios3`.

5 Círculo y ColorCir

Mediante la opción `Circulo` se agregan al gráfico círculos de radios dados, centrados en el origen. El valor por omisión es `Circulo->0`, lo que significa que no traza círculos.

`Circulo->r` : En este caso se agrega un círculo de radio `r`.

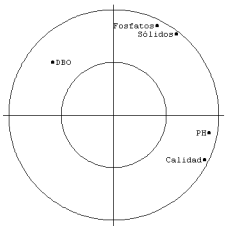
`Circulo->{r1,r2,...}` : Agrega un círculo de radio `ri`, por cada número `ri` dado.

Con `ColorCir` se determina el color de los círculos, el valor por omisión es `ColorCir->Hue[0.55]`, que corresponde a un celeste.

5.1 Ejemplo

El siguiente ejemplo muestra el primer plano principal del espacio de variables en el ejemplo 1 (sección **??**), con círculos de radio 0.5 y 1:

```
ACPPlot[Indic,Etiquetas->EtiIndic,Circulo->{0.5,1}, Ticks->None];
```



La opción `Ticks->None` elimina las marcas y los respectivos números en los ejes .

6 Renglones y Columnas

La colocación de etiquetas a los puntos de un gráfico con `ACPPlot`, es un proceso que se ejecuta automáticamente, mediante un algoritmo que trata de evitar la sobreposición de etiquetas. Este algoritmo divide el área del gráfico en n renglones y k columnas, formando una especie de cuadrícula, y para su funcionamiento óptimo se supone que en cada cuadro sólo cabe un caracter alfabético o numérico. De esta manera, una vez que se

ha determinado el tamao del gráfico, y el "font" a utilizar en el texto del gráfico, se debe tener que:

$$n = \frac{\text{altura-del-área-del-gráfico}}{\text{altura-máxima-de-los-caracteres}}$$

y

$$k = \frac{\text{ancho-del-área-del-gráfico}}{\text{ancho-promedio-de-los-caracteres}}$$

Aunque estos valores, dependen del tamao final del área del gráfico, el que se determina opcionalmente al "arrastrar" uno de sus vértices con el "mouse", el algoritmo trabaja apropiadamente, con los valores fijados por omisión, al menos en los casos en que no hay configuraciones de puntos "complejas" de etiquetar.

7 Ejemplo 2: AFC

Antes de describir otras opciones se presentará un segundo ejemplo de cálculos con *Mathematica*, esta vez de un AFC, a fin de elaborar las ilustraciones de las siguientes opciones de `ACPPlot`.

El Análisis Factorial de Correspondencias (AFC), que se hará corresponde a las variables: Nivel de Instrucción e Ingresos Totales, medidas sobre los jefes de familia en la Encuesta Nacional de Hogares, del ao 1994, para la región metropolitana. Con esta encuesta se construyó la siguiente tabla de contingencias `EdxIng`, Nivel de Instrucción contra Ingresos Totales, utilizando la biblioteca de procedimientos `Conteos.ma`.

```
(*Lectura de la tabla EdxIng, registrada en el archivo MiTabla.txt *)
MiTabla.txt
```

$$\{\{20,59,42,8,1,3,0,0,0,0,0\},$$
$$\{42,134,116,49,12,5,1,5,1,0,4\},$$
$$\{53,113,281,170,70,28,21,4,3,4,5\},$$
$$\{13,13,96,110,60,31,8,6,6,2,11\},$$
$$\{0,0,11,12,5,9,6,2,0,0,2\},$$
$$\{5,0,8,34,24,25,22,20,12,7,19\}\}$$

La orden siguiente define las etiquetas para las modalidades de las dos variables que involucra la tabla. Las columnas corresponden a la variable ingreso total y sus etiquetas muestran el límite superior (en miles de colones) de cada clase de ingreso.

```
EtiFil = {"Ning.Esc","Esc.Inc","Esc.Comp","Colegio","Ed.Tec","Ed.Sup"};
EtiCol = {"0","20","40","60","80","100","125","150","175","200","250"};
```

Cálculo de algunos datos requeridos para el ACP que requiere el AFC. No se muestran los resultados parciales:

```
      (* Totales de filas, columnas y global *)
tc = Apply[Plus,EdxIng];
tf = Apply[Plus,EdxIng,{1}];
T = Apply[Plus,tf];
      (* Centros de Gravedad filas y columnas *)
Gf = tc/T//N;
Gc = tf/T//N;
```

```
      (* Tablas de perfiles fila y columna*)
Pf = (1/tf) EdxIng //N;
Pc = (1/tc) Transpose[EdxIng] //N;
      (* Definición de métricas *)
InvDq = DiagonalMatrix[T/tc]//N;
Dp = DiagonalMatrix[tf/T]//N;
InvDp = DiagonalMatrix[T/tf]//N;
```

Cálculo del ACP sobre la tabla `Pf` obtenida, con `M = InvDq` y `Dp` como se define arriba. La primera orden activa los procedimientos de la biblioteca `varcor.ma`, en este caso, sólo para utilizar el procedimiento `Normaliza`. El paquete `varcor.ma` fue desarrollado por el autor y está disponible solicitándolo al mismo.

`varcor.ma`

```
Vf = Transpose[Pf].Dp.Pf;
      (* Vectores y valores propios de VM *)
{Landas,Ut} = Eigensystem[Vf.InvDq];
      (* M-Normalización de vectores que generan los ejes principales *)
U = Transpose[Normaliza[Ut,InvDq]];
      (* Inercia explicada *)
inerciaTotal = Apply[Plus,Landas]-1;
IE = Rest[100 Landas/inerciaTotal];
      (* Componentes Principales:
        coordenadas de perfiles fila sobre ejes principales *)
Y = Pf.InvDq.U;
      (* Coordenadas de perfiles columna sobre ejes principales *)
Z = InvDq.U.DiagonalMatrix[Sqrt[Landas]];
```

Definición de los puntos correspondientes a perfiles fila y columna en el plano de ejes 2 y 3:

```
PtsPFil = Columna[Y,{2,3}]
```

$$\{\{-0.7735, 0.427935\}, \{-0.54232, 0.279352\}, \{-0.117024, -0.180424\},$$
$$\{0.356034, -0.333867\}, \{0.745172, -0.123647\}, \{1.3024, 0.570174\}\}$$

```
PtsPCol = Columna[Z,{2,3}]
```

$$\{\{-0.44984, 0.224675\}, \{-0.714536, 0.384778\}, \{-0.244733, -0.170292\},$$
$$\{0.18633, -0.27345\}, \{0.426822, -0.297379\}, \{0.746825, 0.0134594\}, \{1.02109, 0.313454\},$$
$$\{1.28619, 0.859049\}, \{1.37746, 0.672833\}, \{1.29334, 0.647263\}, \{1.20027, 0.562319\}\}$$

8 PlotJoined

Si en el conjunto de puntos se define alguna agrupación, `PlotJoined->Si` forma una trayectoria por cada grupo de puntos, uniendo lo puntos según el orden en que aparecen. Con la opción `PlotStyle` se puede elegir un estilo de línea para todas la trayectorias o un estilo para cada trayectoria.

8.1 Ejemplo

Los siguientes comandos redefinen las variables `EtiFil` y `EtiCol` para detallar la posición que deben ocupar, alrededor del punto, dado que la colocación automática de etiquetas no produce resultados satisfactorios.

```
EtiFil2 = {{"Ning.Esc",Arriba},{"Esc.Inc",Derecha},{"Esc.Comp",Derecha},
      {"Colegio",Abajo},{"Ed.Tec",Derecha},{"Ed.Sup",Derecha}};
EtiCol2 = {{"0",Derecha},{"20",Izquierda}, {"40",Izquierda},
      {"60",Izquierda},{"80",Derecha},{"100",Izquierda},{"125",Izquierda},
      {"150",Arriba},{"175",Derecha},{"200",Izquierda},{"250",Izquierda}};
ACPPlot[{PtsPFil,PtsPCol}, Etiquetas->{EtiFil2,EtiCol2},
      PlotRange->{{-1.1,1.8},{-0.5,1}},PlotJoined->Si];
```

9 PlotStyle

Se utiliza para especificar el estilo de los puntos o líneas en el gráfico. Aunque `ListPlot` tiene una opción con este nombre, en `ACPPlot` adquiere características muy particulares, como se verá. Un estilo para un punto, es la definición del tamao y su color, y para las líneas, es la definición de grosor, color y el tipo del trazo (continuo o punteado).

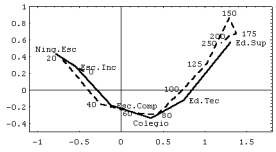
`PlotStyle->Automatic`**:** Genera automáticamente estilos diferentes de puntos o líneas para ser aplicados a cada grupo de puntos, definido explícitamente o utilizando `Distinga`.

`PlotSyle->{estilo1,estilo2, ...}`**:** Aplica cada estilo dado a cada grupo de puntos. El número de estilos y el número de grupos debe ser el mismo, excepto cuando no hay agrupaciones definidas en la lista de puntos o con `Distinga`, en cuyo caso genera una agrupación de los puntos en tantos grupos como estilos se hayan dado.

9.1 Ejemplo

La orden dada elige un estilo de línea punteado para una de las trayectorias, utilizando la función `Dashing`.

```
ACPPlot[{PtsPFil,PtsPCol},Etiquetas->{EtiFil,EtiCol},
      PlotRange->{{-1.1,1.8},{-0.5,1}},PlotJoined->Si,
      PlotStyle->{Rojo,{Dashing[{0.02,0.02}],Azul}}];
```



10 Otras opciones

Todas aquellas opciones que se utilizan con `ListPlot` también se pueden aplicar en `ACPPlot`: `Axes`, `Frame`, `AspectRatio`, `PlotRange`, `PlotLabel`, `AxesLabel`, `FrameLabel`, `AxesStyle`, `FrameStyle`, `DefaultFont`, `Background`, `Ticks`, `FrameTicks`, `GridLines`, `DisplayFunction`.

El siguiente ejemplo, ilustra la utilización de `PlotLabel` y `FrameLabel`:

```
ACPPlot[{PtsPFil,PtsPCol}, PlotJoined->Si,
  Etiquetas->{{"Niv.Inst",5,Derecha}, {"Ingresos",8,Izquierda}},
  PlotLabel->FontForm["Educación vs Ingresos", {"Times-Bold",14}],
  FrameLabel->{"Eje 2:  72.5%","Eje 3:  22.26%"}];
```

11 Conclusiones

Actualmente se prepara una nueva implementación para `ACPPlot` que permita representar los puntos, de diversas formas: triángulos, cuadrados, etc., objetos que deben ser magnificables. De esta manera, se puede mejorar la presentación de muchos gráficos, que se imprimen utilizando un solo color de tinta. También se trabaja en la creación del procedimiento `ACPPlot3D`, para la construcción de estos gráficos en 3 dimensiones.

12 ACPPlot: definición del procedimiento

```
BeginPackage["ACPPlot`"];
ACPPlot::usage =
      "ACPPlot[Puntos,Opciones]:  construye un gráfico al estilo de ListPlot, pero agregando
      una serie de opciones.  También acepta la sintaxis ACPPlot[{Pts1,Pts2,...},Opciones]."
(* ACPlot reconoce todas las opciones para ListPlot y además se agregan las siguientes:  *)
Etiquetas::usage = "La opción Etiquetas->{Eti1,Eti2,...}, especifica una lista de etiquetas
      para los puntos a graficar y las coloca automáticamente tratando de evitar sobreposiciones.
      Al igual que en la lista de puntos, las etiquetas pueden estar agrupadas.  Con el valor por
      omisión Etiquetas->Automatic, se genera una lista de etiquetas de la forma {1,2,...} para
      cada grupo de puntos, si hay más de uno."
Circulo::usage = "Esta opción permite trazar círculos de radio centrados en el origen, de radios
      dados.  El valor por omisión es:  Circulo->0, con lo cual no traza círculos."
ColorCir::usage = "Define el color para el círculo"
Renglones::usage = "No.  de renglones en que se particiona horizontalmente el área del gráfico,
      para efectos de colocar las etiquetas.  Valor por omisión:  Renglones->16"
Columnas::usage = "No de columnas en que se particiona verticalmente el área del gráfico,
      para efectos de colocar las etiquetas.  Valor por omisión Columnas->24"
Distinga::usage = "La opción Distinga->{n1,n2,...,nk} partiona y distingue el conjunto de puntos
      a graficar en grupos de tamaos n1,n2, ..., nk.  Si la suma de los enteros dados es menor
      que el número de puntos, crea un grupo con los restantes.  Con el valor por omisión
      Distinga->Automatic, respeta la agrupación explícita con que puedan venir los puntos o si
      se han especificado varios estilos para puntos o líneas en PlotStyle genera una agrupación
      uniforme según este número de estilos."
Si::usage = "Posible valor para opciones como Circulo, Axes, etc."
No::usage = "Posible valor para opciones como Circulo, Axes, etc."
Auto::usage = "Igual que Automatic, es un posible valor de algunas opciones"
Ninguna::usag = "Valor para opción Etiquetas, para omitirlas."
Derecha::usage = "Posible valor para opción Etiquetas:
      Etiquetas->{ListaEti,Derecha}:  Alinea todas las etiquetas a la derecha del punto.
      Etiquetas->{Auto,Derecha}:  Genera automáticamente las etiquetas y las alinea todas
      a la derecha del punto."
Izquierda::usage = "Como Derecha es un posible valor para Etiquetas"
Arriba::usage = "Como Derecha es un posible valor para Etiquetas"
Abajo::usage = "Como Derecha es un posible valor para Etiquetas"
Negro::usage = "Posible valor para usar en estilos"
Rojo::usage = "Posible valor para usar en estilos"
Azul::usage = "Posible valor para usar en estilos"
Verde::usage = "Posible valor para usar en estilos"
Amarillo::usage = "Posible valor para usar en estilos"
Rosado::usage = "Posible valor para usar en estilos"
Celeste::usage = "Posible valor para usar en estilos"
Gris::usage = "Posible valor para usar en estilos"
Begin["`privado`"];
CatStls = {RGBColor->Hue,GrayLevel->Hue,AbsoluteThickness->Thickness,AbsoluteDashing->Dashing};
StyloQ[_RGBColor|_Hue|_GrayLevel|_AbsolutePointSize|_PointSize] := True;
StyloQ[_Thickness|_AbsoluteThickness|_Dashing|_AbsoluteDashing] := True;
StyloQ[ls:{__?StyloQ}] := Length[ls] == Length[Union[Map[Head,ls]/.CatStls]];
StyloQ[_] := False;
Resto[x_List,k_Integer] := Take[x,k-Length[x]];
Parta[{},{}] := {};
Parta[x_List,{}] := {x};
Parta[x_List,p_List] := Prepend[Parta[Resto[x,First[p]],Rest[p]],Take[x,First[p]]];
aPtsSty[{pts:{{_?NumberQ,_?NumberQ}..}},Automatic,Automatic] := {{pts,Estilos[1]}};
aPtsSty[{pts:{{_?NumberQ,_?NumberQ}..}},Automatic,stils_?StyloQ] := {{pts,stils}};
aPtsSty[{pts:{{_?NumberQ,_?NumberQ}..}},parti:{__Integer},stils:{__?StyloQ}] :=
      Transpose[{Parta[pts,parti],stils}];
aPtsSty[{pts:{{_?NumberQ,_?NumberQ}..}},parti:{__Integer},Automatic] :=
      With[{vrspts = Parta[pts,parti]},Transpose[{vrspts,Estilos[Length[vrspts]]}]];
aPtsSty[{pts:{{_?NumberQ,_?NumberQ}..}},Automatic,stils:{__?StyloQ}] :=
      aPtsSty[{pts},Table[Floor[N[Length[pts]/Length[stils]]],{Length[stils]-1}],stils];
```

```
aPtsSty[pts:{{{_?NumberQ,_?NumberQ}..}..},Automatic,stils:{__?StyloQ}] := Transpose[{pts,stils}];
aPtsSty[pts:{{{_?NumberQ,_?NumberQ}..}..},Automatic,Automatic] :=
      aPtsSty[pts,Automatic,Estilos[Length[pts]]];
Grupos[{npts_Integer},Automatic,Automatic|_?StyloQ] := {npts};
Grupos[{npts_Integer},parti:{__Integer},_] :=
      With[{k = Apply[Plus,parti]},If[npts == k,parti,Append[parti,npts-k]]];
Grupos[{npts_Integer},_,Stls:{__?StyloQ}] :=
      Module[{k,nst},
            nst = Length[Stls];
            k = Floor[N[npts/nst]];
            If[npts == k nst,Table[k,{nst}],Append[Table[k,{nst-1}],npts - (k (nst-1))]]];
Grupos[npts:{_Integer,__Integer},_,_] := npts;
ColocaEt[Ets_,z1_,z2_,nr_,nc_,{{xmin_,xmax_},{ymin_,ymax_}}] :=
  Module[{aRgl,aCol,dE,OEtis,Busca,dst,Dstn,Coloca,ActdE,ColocaPts},
        aRgl[y_] := Round[N[((nr-1)/(ymax-ymin))(y-ymin) + 1]];
        aCol[x_] := Round[N[((nc-1)/(xmax-xmin))(x-xmin) + 1]];
        dE = Table[{{0,0},{nc+1,nc+1}},{nr}];
        OEtis = {};
        Busca[ci_,rglj_] := Module[{Bus},
          Bus[i_,{{nc+1,nc+1}}] := 0;
          Bus[i_,dEj_] :=
            If[Interval[dEj[[1]]] < Interval[ci] < Interval[dEj[[2]]],i,Bus[i+1,Rest[dEj]]];
          Bus[1,rglj]];
        dst[{_,b_},{c_,_}] := c - b;
        Dstn[ci_,rg_] := With[{p = Busca[ci,dE[[rg]]]},
          If[p == 0,{0,0},{dst[dE[[rg,p]],ci],dst[ci,dE[[rg,p+1]]]}]];
        Coloca[{x_,y_},len_] := Module[{ds,mx,le2},
          le2 = Floor[N[len/2]];
          ds = If[y > nr || y < 1, {-1,-1,-1,-1},
              {Last[Dstn[{x+1,x+len+1},y]],
              First[Dstn[{x-len-1,x-1},y]],
              If[y == nr,0,Apply[Min,Dstn[{x-le2,x+le2},y+1]]],
              If[y == 1,0,Apply[Min,Dstn[{x-le2,x+le2},y-1]]]}];
          mx = Max[ds];
          Which[mx == -1,AppendTo[OEtis,Fuera],
              mx == 0,AppendTo[OEtis,NoCabe],
              ds[[1]] > 3,ActdE[{x+1,x+len+1},y,Derecha],
              ds[[2]] > 3,ActdE[{x-len-1,x-1},y,Izquierda],
              mx == ds[[1]],ActdE[{x+1,x+len+1},y,Derecha],
              mx == ds[[2]],ActdE[{x-len-1,x-1},y,Izquierda],
              mx == ds[[3]],ActdE[{x-le2,x+le2},y+1,Arriba],
              mx == ds[[4]],ActdE[{x-le2,x+le2},y-1,Abajo]]];
        ActdE[ci_,y_,Ori_] := With[{p = Busca[ci,dE[[y]]]},
          dE[[y]] = Insert[dE[[y]],ci,p+1];
          AppendTo[OEtis,Ori]];
        ColocaPts[{x_,y_}] := Module[{p},
          If[y > nr || y < 1, Ok, p = Busca[{x,x},dE[[y]]];
          If[p > 0,dE[[y]] = Insert[dE[[y]],{x,x},p+1],Ok]]];
        dE[[aRgl[0]]] = Insert[dE[[aRgl[0]]],{1,nr},2];
        Do[ColocaPts[{aCol[z1[[i]]],aRgl[z2[[i]]]}],{i,1,Length[z1]}];
        Do[Coloca[{aCol[z1[[i]]],aRgl[z2[[i]]]},StringLength[Ets[[i]]]],{i,1,Length[z1]}];
        OEtis];
Rgls = {Si->True,No->False,Auto->Automatic,Derecha->{-1,0},Izquierda->{1,0},Arriba->{0,-1},
      Abajo->{0,1},NoCabe->{},Fuera->{},Ninguna->None,Negro->RGBColor[0,0,0],Rojo->RGBColor[1,0,0],
      Azul->RGBColor[0,0,1],Verde->RGBColor[0,1,0],Amarillo->RGBColor[1,1,0],
      Rosado->RGBColor[1,0,1],Celeste->RGBColor[0,1,1],Gris->GrayLevel[0.5]};
Colores = {RGBColor[0,0,0],RGBColor[1,0,0],RGBColor[0,0,1],RGBColor[0,1,0],RGBColor[1,1,0],
      RGBColor[1,0,1],RGBColor[0,1,1],GrayLevel[0.5]};
Estilos[0] := {};
Estilos[1] := {AbsolutePointSize[2],RGBColor[0,0,0]};
Estilos[k_] := Transpose[{Table[AbsolutePointSize[2],{k}],Take[Colores,k]}]/; k < 9;
Estilos[k_] := With[{t = Quotient[k,8]},
```

```
  Join[Apply[Join,Table[Estilos[8],{t}]],Estilos[k - 8 t]]]/;k > 8;
Options[ACPPlot] := {Etiquetas->Automatic,Circulo->0,Renglones->16,Columnas->24,
      DisplayFunction->$DisplayFunction,AspectRatio->Automatic,PlotRange->Automatic,
      PlotStyle->Automatic,Distinga->Automatic,PlotJoined->False,AxesStyle->{Hue[0.55]},
      ColorCir->Hue[0.55]};
EstasNo[opcion:_Rule | _RuleDelayed] := FreeQ[{Etiquetas,Circulo,AspectRatio,
      DisplayFunction,PlotRange,Frame,PlotStyle,PlotJoined,AxesStyle,ColorCir,Distinga,
      Renglones,Columnas},opcion[[1]]];
ACPPlot[Pts:{{{_?NumberQ,_?NumberQ}..}..}, Opc:(_Rule|_RuleDelayed)...]  :=
  Module[{g1,g2,g3,y1,y2,Prte, Ops,AspR,DisF,Etis,Cir,Prng,coo,Coor,dx,GeneraEt,Cuad,PltS,
  EjeS,PlJn,MiPlot,MiListPlot,AutoEtis,MisEtis,Puntos,grps,EtisOK},
  {y1,y2} = Transpose[Flatten[Pts,1]];
  MisEtis[n_] := Table[ToString[i],{i,1,n}];
  AutoEtis[ns:{__Integer}] := Apply[Join,Map[MisEtis,ns]];
  GeneraEt[Automatic,n_] := {AutoEtis[n],ColocaEt[AutoEtis[n],y1,
      y2,nrgl,ncol,FullOptions[g1,PlotRange]]};
  GeneraEt[{E_List,or_Symbol},n_] := {E,Table[or,{Apply[Plus,n]}]};
  GeneraEt[{Automatic,or_Symbol},n_] := {AutoEtis[n],Table[or,{Apply[Plus,n]}]};
  GeneraEt[Et:{{_String,_Symbol}..},n_] := Transpose[Et];
  GeneraEt[Et:{{{_String,_Symbol}..}..},n_] := Transpose[Apply[Join,Et]];
  GeneraEt[Et:{__String},n_] := {Et,ColocaEt[Et,y1,y2,nrgl,ncol,FullOptions[g1,PlotRange]]};
  GeneraEt[Et:{{__String}..},n_] := {Apply[Join,Et],
      ColocaEt[Apply[Join,Et],y1,y2,nrgl,ncol,FullOptions[g1,PlotRange]]};
  Coor[{x_,y_},ori_] := Which[ori === Derecha,{x+dx,y},ori === Izquierda,{x-dx,y},True,{x,y}];
  AspR = AspectRatio/.{Opc}/.Options[ACPPlot]/.Rgls;
  DisF = DisplayFunction/.{Opc}/.Options[ACPPlot]/.Rgls;
  Prng = PlotRange/.{Opc}/.Options[ACPPlot]/.Rgls;
  PltS = PlotStyle/.{Opc}/.Options[ACPPlot]/.Rgls;
  Prte = Distinga/.{Opc}/.Options[ACPPlot]/.Rgls;
  EjeS = AxesStyle/.{Opc}/.Options[ACPPlot]/.Rgls;
  PlJn = PlotJoined/.{Opc}/.Options[ACPPlot]/.Rgls;
  Cir = Circulo/.{Opc}/.Options[ACPPlot]/.Rgls;
  Cuad = Frame/.{Opc}/.{Frame->(0 === Cir)};
  ColC = ColorCir/.{Opc}/.Options[ACPPlot]/.Rgls;
  nrgl = Renglones/.{Opc}/.Options[ACPPlot]/.Rgls;
  ncol = Columnas/.{Opc}/.Options[ACPPlot]/.Rgls;
  Ops = Join[{AspectRatio->AspR,DisplayFunction->DisF,PlotRange->Prng,Frame->Cuad},
      Select[{Opc},EstasNo]]/.Rgls;
  MiPlot[{puntos_,stilo_}] := ListPlot[puntos,PlotStyle->stilo,Prolog->{AbsolutePointSize[2]},
      PlotRange->Prng,AxesStyle->EjeS,PlotJoined->PlJn,DisplayFunction->Identity];
  MiListPlot[pts_,dstng_,stls_] := Show[Map[MiPlot,aPtsSty[pts,dstng,stls]]];
  g1 = MiListPlot[Pts,Prte,PltS];
  dx = (Last[#]-First[#])&[First[FullOptions[g1,PlotRange]]]/ncol/2;
  grps = Grupos[Map[Length,Pts],Prte,PltS];
  Puntos = Transpose[{y1,y2}];
  EtisOK[{},{},_] := {};
  EtisOK[Ets_,prt_,j_] := With[{S = First[Ets],k = First[prt]},
      Append[EtisOK[Rest[Ets],Rest[prt],j+k],
        ReplacePart[S,Coor[Puntos[[j+S[[2]]]],S[[3]]],2]]];
  g2 = With[{Etqts = Etiquetas/.{Opc}/.Options[ACPPlot]/.Ninguna->None},
      Switch[Etqts,
        None, {},
        {{_String,_Integer,_Symbol}..},
          Graphics[Apply[Text,EtisOK[Etqts,grps,0]/.Rgls,{1}]],
        _,Etis = GeneraEt[Etqts,grps];
          coo = Thread[Coor[Puntos,Etis[[2]]]];
          Etis = Etis/.Rgls;
          Etis = Transpose[{Etis[[1]],coo,Etis[[2]]}];
          Etis = DeleteCases[Etis,{_,_,{}}];
          Graphics[Apply[Text,Etis,{1}]]]];
  g3 = Switch[Cir, 0, {},
      _Real|_Integer|_Rational,Graphics[{ColC,Circle[{0,0},Cir]}],
```

```
      _List,Graphics[Join[{ColC},Map[Circle[{0,0},#]&,Cir]]]];
  Apply[Show,Join[{g1,g2,g3},Ops]]];
ACPPlot[Pts:{{_?NumberQ,_?NumberQ}..},opciones:(_Rule | _RuleDelayed)...]  :=
  ACPPlot[{Pts},opciones];
End[];
SetAttributes[ACPPlot, ReadProtected];

EndPackage[]
```

Referencias

[1] Diday, E.; Lemaire, J.; Pouget, J.; Testu, F. (1982) *léments d'Analyse des Données*. Dunod, París.

[2] Gray, J. (1994) *Mastering Mathematica: Programming Methods and Applications*. A. P. Professional, Massachusetts.

[3] Wolfram, S. (1988) *Mathematica, a System for Doing Mathematics by Computer*. Addison-Wesley, Massachusetts.