

Comandos útiles

Atributos de cada tipo

Atributo	Tipos
mode	Todos
Storage.mode	Todos los datos de modo numérico
length	Todos
names	Vectores y listas
dim	Matrices y arrays
dimnames	Matrices y arrays
tsp	Series temporales
levels	Factores
class	Cualquier clase

Operadores lógicos

Operador	Operación
!x	Negación de x. Los T los convierte en F y viceversa.
x&y	Intersección, operador lógico y: T y T da T, otra comparación da F.
x y	Unión, operador lógico o: F y F da F, otra comparación da T.
xor(x,y)	Exclusivo OR, xor (T, F)==T, otra comparación da F.
all	Para una secuencia de argumentos lógicos all devuelve el valor lógico que indica si todos los elementos son TRUE.
any	Para una secuencia de argumentos lógicos any devuelve el valor lógico que indica si algún elemento es TRUE.

Coerción de tipos

Tipo	Comprobación	Coerción
array	is.array()	as.array()
character	is.character()	as.character()
complex	is.complex()	as.complex()

double	is.double()	as.double()
factor	is.factor()	as.factor()
integer	is.integer()	as.integer()
list	is.list()	as.list()
logical	is.logical()	as.logical()
matrix	is.matrix()	as.matrix()
NA	is.na()	-
NaN	is.nan()	-
NULL	is.null()	as.null()
numeric	is.numeric()	as.numeric()
ts	is.ts()	as.ts()
vector	is.vector()	as.vector()

Funciones útiles para trabajar con matrices

Función	Utilidad
ncol(x)	Número de columnas de x.
nrow(x)	Número de filas de x.
t(x)	Transpuesta de x
cbind(...)	Combina secuencias de vectores/matrices por col's.
rbind(...)	Combina secuencias de vectores/matrices por filas.
diag(x)	Extrae diagonal de matriz o crea matriz diagonal.
col(x)	Crea una matriz con elemento ij igual al valor j
row(x)	Crea una matriz con elemento ij igual al valor i
apply(x,margin,FUN,)	Aplica la función FUN a la dimensión especificada en margin 1 indica filas, 2 indica columnas. NB.
outer(x,y,fun="*") otra forma x %o %y	Para dos vectores x e y , crea una matriz $A[i,j]=FUN(x[i],y[j])$. Por defecto crea el producto externo.

Algunas operaciones con matrices

Función	Utilidad
x %* %y	Multiplicación de matrices
crossprod(x,y=x)	Idem que t(x) %* %y, pero más rápida
cov(x,y=x,use="all.obs")	Matriz de varianzas-covarianzas
cor(x,y=x,use="all.obs")	Matriz de correlaciones
scale(x,center=,scale=)	Resta a las columnas la media si center=TRUE, Si center es un vector, resta dichos valores. Idem para scale ,luego de centrar, pero divide por la desviación típica si scale=TRUE o los valores asignados si es un vector.
chol(x)	Descomposición de Choleski.
solve(a,b,tol=1e-7)	Resolución de la ecuación $a \%* \%x=b$. tol=tolerancia para detectar dependencias lineales en las columnas de a

<code>eigen(x)</code>	Cálculo de valores y vectores propios.
<code>sdv(x)</code>	Descomposición en valores singulares.

Tipos de gráficos

<code>fourfoldplot(x)</code>	utilizando cuartos de círculos, visualiza la asociación entre dos variables dicotómicas para diferentes poblaciones (x debe ser un arreglo de $\text{dim} = c(2, 2, k)$, o una matriz de $\text{dim} = c(2, 2)$ si $k = 1$)
<code>assocplot(x)</code>	Gráfica 'amigable' de Cohen mostrando desviaciones de independencia de filas y columnas en una tabla de contingencia de dos dimensiones
<code>mosaicplot(x)</code>	gráfico 'mosaico' de los residuales de una regresión log-lineal de una tabla de contingencia
<code>pairs(x)</code>	si x es una matriz o un marco de datos, dibuja todas las posibles gráficas bivariadas entre las columnas de x
<code>plot.ts(x)</code>	si x es un objeto de clase "ts", grafica x con respecto al tiempo. x puede ser multivariada pero las series deben tener la misma frecuencia y fechas
<code>ts.plot(x)</code>	igual a la anterior pero si x es multivariado, las series pueden tener diferentes fechas pero la misma frecuencia
<code>hist(x)</code>	histograma de las frecuencias de x
<code>barplot(x)</code>	histograma de los valores de x
<code>qqnorm(x)</code>	cuartiles de x con respecto a lo esperado bajo una distribución normal
<code>qqplot(x, y)</code>	cuartiles de y con respecto a los cuartiles de x
<code>contour(x, y, z)</code>	gráfico de contornos (los datos son interpolados para dibujar las curvas), x y y deben ser vectores, z debe ser una matriz tal que $\text{dim}(z) = c(\text{length}(x), \text{length}(y))$ (x y y pueden ser omitidos)
<code>filled.contour(x, y, z)</code>	igual al anterior, pero las áreas entre contornos están coloreadas, y se dibuja una leyenda de colores
<code>image(x, y, z)</code>	igual al anterior pero con colores (se grafican los datos crudos)
<code>persp(x, y, z)</code>	igual al anterior pero en perspectiva (se grafican los datos crudos)
<code>stars(x)</code>	si x es una matriz o un marco de datos, dibuja una gráfica con segmentos o una estrella, donde cada fila de x es representada por una estrella, y las columnas son las longitudes de los segmentos
<code>symbols(x, y, ...)</code>	dibuja, en las coordenadas dadas por x y y , símbolos (círculos, cuadrados, rectángulos, estrellas, termómetros o cajas) cuyos tamaños, colores ... son especificados con argumentos adicionales
<code>termplot(mod.obj)</code>	gráfico de los efectos (parciales) de un modelo de regresión (<code>mod.obj</code>)

Atributos gráficos

adj	controla la justificación del texto (0 justificado a la izquierda, 0.5 centrado, 1 justificado a la derecha)
bg	especifica el color del fondo (ej. : <code>bg=#ed</code> , <code>bg=blue</code> , ... La lista de los 657 colores disponibles se puede ver con <code>colors()</code>)
bty	controla el tipo de caja que se dibuja alrededor del gráfico: <code>o</code> , <code>"l"</code> , <code>"7"</code> , <code>ä</code> , <code>ü</code> o <code>]"</code> (la caja se parece a su respectivo caracter); si <code>bty="n"</code> no se dibuja la caja
cex	un valor que controla el tamaño del texto y símbolos con respecto al valor por defecto; los siguientes parámetros tienen el mismo control para números en los ejes, <code>cex.axis</code> , títulos en los ejes, <code>cex.lab</code> , el título principal, <code>cex.main</code> , y el subtítulo, <code>cex.sub</code>
col	controla el color de los símbolos; como en <code>cex</code> estos son: <code>col.axis</code> , <code>col.lab</code> , <code>col.main</code> y <code>col.sub</code>
font	un entero que controla el estilo del texto (1: normal, 2: cursiva, 3: negrilla, 4: negrilla cursiva); como en <code>cex</code> existen: <code>font.axis</code> , <code>font.lab</code> , <code>font.main</code> y <code>font.sub</code>
las	un entero que controla la orientación de los caracteres en los ejes (0: paralelo a los ejes, 1: horizontal, 2: perpendicular a los ejes, 3: vertical)
lty	un entero o caracter que controla el tipo de las líneas; (1: sólida, 2: quebrada, 3: punteada, 4: punto-línea, 5: línea larga-corta, 6: dos líneas cortas), o una secuencia de hasta 8 caracteres (entre <code>"0"</code> y <code>"9"</code>) que especifica alternativamente la longitud en puntos o pixeles, de los elementos dibujados y los blancos; por ejemplo <code>lty="44"</code> tendrá el mismo efecto que <code>lty=2</code>
lwd	un número que controla la anchura de las líneas
mar	un vector con 4 valores numéricos que controla el espacio entre los ejes y el borde de la gráfica en la forma <code>c(inferior, izquierda, superior, derecha)</code> ; los valores por defecto son <code>c(5.1, 4.1, 4.1, 2.1)</code>
mfcol	un vector del tipo <code>c(nr,nc)</code> que divide la ventana gráfica como una matriz con <code>nr</code> filas y <code>nc</code> columnas; las gráficas se dibujan sucesivamente en las columnas (véase la sección 4.1.2)
mfrow	igual al anterior, pero las gráficas se dibujan en las filas (ver sección 4.1.2)
pch	controla el tipo de símbolo, ya sea un entero entre 1 y 25, o un caracter entre <code>"</code> (Fig. 2)
ps	un entero que controla el tamaño (en puntos) de textos y símbolos
pty	un caracter que especifica el tipo de región a graficar, <code>"s"</code> : cuadrada, <code>"m"</code> : máxima
tck	un valor que especifica la longitud de los marcadores de eje como una fracción de la altura o anchura máxima del gráfico; si <code>tck=1</code> se dibuja una rejilla
tcl	un valor que especifica la longitud de los marcadores de eje como una fracción de la altura de una línea de texto (por defecto <code>tcl=-0.5</code>)
xaxt	si <code>xaxt="n"</code> el eje <code>x</code> se coloca pero no se muestra (util en conjunción con <code>axis(side=1, ...)</code>)
yaxt	if <code>yaxt="n"</code> el eje <code>y</code> se coloca pero no se muestra (util en conjunción con <code>axis(side=2, ...)</code>)

<code>points(x, y)</code>	agrega puntos (se puede usar la opción <code>type=</code>)
<code>lines(x, y)</code>	igual a la anterior pero con líneas
<code>text(x, y, labels, ...)</code>	agrega texto dado por <code>labels</code> en las coordenadas (x,y) ; un uso típico: <code>plot(x, y, type="n"); text(x, y, names)</code>
<code>mtext(text, side=3, line=0, ...)</code>	agrega texto dado por <code>text</code> en el margen especificado por <code>side</code> (ver <code>axis()</code> más abajo); <code>line</code> especifica la línea del área de graficado
<code>segments(x0, y0, x1, y1)</code>	dibuja una línea desde el punto (x_0,y_0) hasta el punto (x_1,y_1)
<code>arrows(x0, y0, x1, y1, angle=30, code=2)</code>	igual al anterior pero con flechas desde (x_0,y_0) si <code>code=2</code> , al punto (x_1,y_1) si <code>code=1</code> , o en ambos si <code>code=3</code> ; <code>angle</code> controla el ángulo desde la base de la flecha hasta la punta de la misma
<code>abline(a,b)</code>	dibuja una línea con pendiente b e intercepto a
<code>abline(h=y)</code>	dibuja una línea horizontal en la ordenada y
<code>abline(v=x)</code>	dibuja una línea vertical en la abscisa x
<code>abline(lm.obj)</code>	dibuja la línea de regresión dada por <code>lm.obj</code> (ver sección 5)
<code>rect(x1, y1, x2, y2)</code>	dibuja un rectángulo donde las esquinas izquierda, derecha, superior e inferior están dadas por x_1, x_2, y_1, y_2 , respectivamente
<code>polygon(x, y)</code>	dibuja un polígono uniendo los puntos dados por x y y
<code>legend(x, y, legend)</code>	agrega la leyenda en el punto (x,y) con símbolos dados por <code>legend</code>
<code>title()</code>	agrega un título y opcionalmente un sub-título
<code>axis(side, vect)</code>	agrega un eje en la parte inferior (<code>side=1</code>), izquierda (2), superior (3), o derecha (4); <code>vect</code> (opcional) da la abscisa (u ordenada) donde se deben dibujar los marcadores ('tick marks') del eje
<code>rug(x)</code>	dibuja los datos x en el eje x como pequeñas líneas verticales
<code>locator(n, type="n", ...)</code>	devuelve las coordenadas (x,y) después que el usuario a hecho click n veces en el gráfico con el ratón; también dibuja símbolos (<code>type="p"</code>) o líneas (<code>type="l"</code>) con respecto a parámetros gráficos opcionales (...); por defecto no se dibuja nada (<code>type="n"</code>)
<code>identify(x, ...)</code>	similar a <code>locator()</code> con la diferencia que imprime en la gráfica el valor de x (u opcionalmente de una leyenda especificada en la opción <code>labels=</code>) más cercano al punto donde se hizo click. Util para identificar puntos en la gráfica que están asociados con nombres.

