**Funciones**

*Definición*



Esta es la estructura básica de una función. Los argumentos son las distintas variables de las que depende la función. Es importante la colocación de las llaves y que se coloque la final que muchas veces se nos olvida poner, porque sino no se cierra la función. A su vez también es importante que todas las operaciones referentes a la función se realicen entre las llaves.

Como recomendación personal y también del profesor, es aconsejable definir las funciones al principio del script.

Para obtener valores de esa función, simplemente hay que poner el mismo número de valores que variables tiene la función.



*Representación de gráficos*

Una vez definida la función y dados unos valores, pudiendo haberlo hecho con comandos como runif o rnorm. Importante no poner un = después de un plot, es un error muy común. Para representar la función la sintaxis es:



Una vez hecho esto, se pueden añadir más variables para que la gráfica tome forma. Se definen dentro del paréntesis.

**Título**

Le añadimos un título a la gráfica con el objetivo de saber a primera visita de lo que trata. Es importante que se coloquen las comillas, puesto que no es un vector sino una asignación.



**Nombre de los ejes**

Es importante etiquetar a los ejes de abscisas y de ordenadas, es decir el eje x y el eje y, respectivamente, con el fin de saber qué representa cada eje. Como el título, también tiene que ir entre comillas.



**Tamaño y estilo del texto**

Es posible variar el estilo y la fuente del texto dentro del gráfico, es decir, de los nombres de los ejes y el título. Ambas deben ir entre comillas y se designa cex al tamaño y font a la fuente:



En cuanto al tamaño, cuanto más alto sea el número, más grande será la letra.

La fuente varía del 1 al 4 ya que hay cuatro tipos de fuentes posibles:



**Color**

Para que sea más llamativa la gráfica le podemos definir un color, pero hay que escribirlo en inglés y entre comillas. Para que el color sea más oscuro podemos añadir la palabra ‘dark’ delante del color.



**Límites**

Para que las líneas de los ejes no sean infinitas hay que establecer unos límites y así cercar la función. Sobre todo, cuando se representa más de una función para poder visualizarlas bien. Es importante que se defina como un vector escribiendo c().



**Símbolos**

Se utiliza el comando pch para cambiar los símbolos de los puntos de la gráfica. Su estructura es:



Cada valor se corresponde con un símbolo, la tabla es la siguiente y se puede encontrar en internet:



**Tipo**

Como bien sabemos, una gráfica puede tener diferentes formas, por ejemplo, puede ser de dispersión, un histograma…

Se designa con el comando type y la letra correspondiente al tipo entre comillas.



**Orientación**

Determina la orientación de los caracteres en los ejes:

 

**Superposición**

Cuando se quiere representar más de una función en un mismo gráfico. La instrucción par sirve para cambiar de manera permanente parámetros gráficos. El argumento “true” indica que la nueva curva se agrega al gráfico anterior.



Se pueden eliminar los ejes con el siguiente comando:



Asimismo, también se puede añadir una leyenda con la simbología de la gráfica. Antes de definir la leyenda hay que asignar un nombre a los componentes del vector.



Después hay que crear una matriz que tenga por filas o columnas los vectores.



Una vez hecho esto, tenemos que decidir dónde colocar la leyenda, las posiciones pueden ser: top, bottom, left, right, top right, top left, bottom right, bottom left. Después de esto, se escribe rownames(A) o colnames(A) en función de si se han designado los vectores en filas o columnas, respectivamente. Finalmente,  escribimos los colores de los que queremos rellenar el cuadradito que sale al lado de los nombres en la leyenda.



Por lo tanto, la sintaxis sería:



**Gráfico circular**

Los gráficos no tienen por qué ser todos del mismo tipo en el que se representan los valores según el eje X y el eje Y; podemos crear otro tipo de gráficos como son los gráficos de sectores circulares. Su definición es:



Donde clockwise puede ser TRUE o FALSE según se quiera en sentido horario o antihorario, respectivamente.

**Gráfico de barras**

Otro tipo de gráfico es el de barras, en el que los valores se representan mediante rectángulos, cuya sintaxis es:



Si queremos superponer las barras para comparar los valores que toman dos funciones distintas en los mismos puntos o intervalos de puntos, hay que construir una matriz que tenga por filas los vectores.



*Creación de varios gráficos*

Si queremos representar diferentes gráficos en la misma pantalla, habrá que distribuirlos de alguna manera. Se ha de utilizar la instrucción par seguida de:

* mfcol: un vector del tipo c(nr,nc), que divide la ventana gráfica como una matriz con nr filas y nc columnas; las gráficas se dibujan sucesivamente en las columnas
* mfrow: igual al anterior, pero las gráficas se dibujan siguiendo el orden de las filas.

Si añadimos también el comando mar (este indica un vector con 4 valores numéricos), podemos controlar el espacio entre los ejes y el borde de la gráfica de la siguiente forma:

