5.4. OTROS GRÁFICOS DE matplotlib.pyplot

Este módulo proporciona clases y funciones con las que podemos obtener gran variedad de gráficos matemáticos y estadísticos, de los que introduciré algunos. En todos los ejemplos usaré el alias plt para el módulo, y np para numpy. En algunos ejemplos usaré el módulo time para usar el instante actual como semilla en la generación de números pseudoaleatorios proporcionados por el módulo np.random.

```
import matplotlib.pyplot as plt
import numpy as np
```

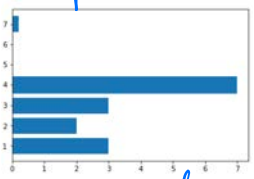
```
import time
np.random.seed(int(time.time()))
```

DIAGRAMA DE BARRAS

Las funciones bar y barh dibujan un diagrama de barras vertical y horizontal, respectivamente. Su prototipo es

bar(datos, frecuencias [, ancho] [, inicio] [, **kwargs])

```
plt.barh((1,2,3,4,7),(3,1,3,7,.2))
```

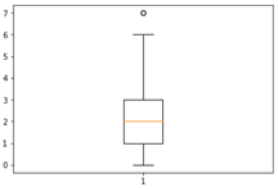


Con la función broken_barh se representan diagramas de "barras rotas", útiles para representar diagramas de Gant.

DIAGRAMA DE CAJA Y BIGOTES

Este diagrama es útil para visualizar la forma de un conjunto de datos respecto a sus medidas de posición central. Sólo es necesario indicar los datos a representar, aceptando además múltiples opciones.

```
b = np.random.binomial(7,.3,5000)
plt.boxplot(b)
```

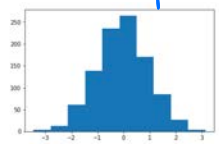


HISTOGRAMA

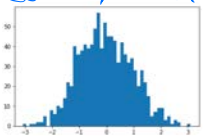
Usaré una muestra de 1000 valores pseudoaleatorios para ilustrar varios ejemplos de la función hist. Si sólo indicamos la serie a representar obtendremos un histograma en el que el rango de los valores representados se divide en 10 intervalos. Si queremos decidir en cuántos intervalos se ha de dividir el rango de los valores a representar lo podemos indicar como segundo parámetro.

```
z = np.random.randn(1000)
```

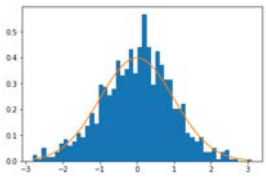
```
plt.hist(z)
```



```
plt.hist(z,50)
```



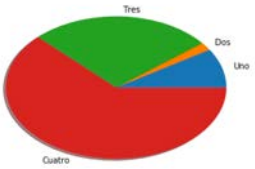
Si queremos componer el histograma con alguna distribución conocida podemos aprovechar que la función hist devuelve una tupla con 3 objetos: 1) la frecuencia de cada intervalo del histograma, 2) los puntos que dividen el rango de los datos en intervalos, y 3) los rectángulos que se han dibujado para formar el histograma. Usaremos además la opción density para que se estandaricen las frecuencias del histograma y sean comparables el histograma y la función representada con plot.



```
f, x, r = plt.hist(z, 50, density=True)
y = 1 / np.sqrt(2*np.pi) * np.exp(-0.5 * x**2)
plt.plot(x,y)
```

Gráfico de torta

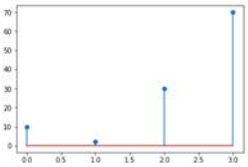
Este gráfico muestra un círculo dividido en tantos sectores como datos le proporcionemos, de área proporcional a cada uno de los datos.




```
plt.pie((10, 2, 30, 70), labels=('Uno', 'Dos', 'Tres', 'Cuatro'), shadow=True)
```

Diagrama de tallos

Sólo necesitamos indicar la longitud de los tallos.



```
plt.stem((10, 2, 30, 70))
```