

Python para Análisis de Datos

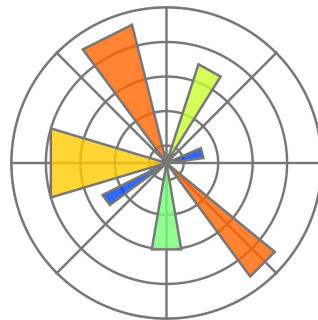
Módulo 06 – Matplotlib 5

Matplotlib

Subplots

En ciertas situaciones podemos querer representar varios gráficos dentro de la misma figura. Podemos crear varios plots dentro de la misma figura con la función `subplot`. Esta función añade un subgráfico a la figura activa y `matplotlib` queda enfocado en ese subplot, de modo que las acciones que hagamos repercuten sobre ese subgráfico. Para pasar a otro subgráfico debemos llamar a `subplot` otra vez.

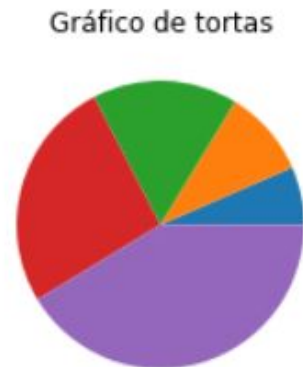
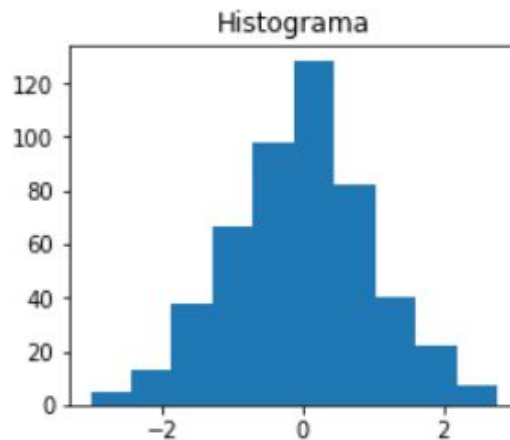
La función `subplot` toma la cantidad de filas y columnas en las que deben acomodarse los subgráficos. Por ejemplo, dos filas y una columna conforman un gráfico sobre otro. Dos filas y dos columnas conforman cuatro subgráficos. El tercer parámetro es el índice del subgráfico sobre el que enfocarse.



Ejemplos

```
plt.figure(figsize=(8,3))  
plt.subplot(1,2,1)  
plt.hist(np.random.randn(500))  
plt.title("Histograma")  
plt.subplot(1,2,2)  
plt.pie([6,9,15,24,38])  
plt.title("Gráfico de tortas")
```

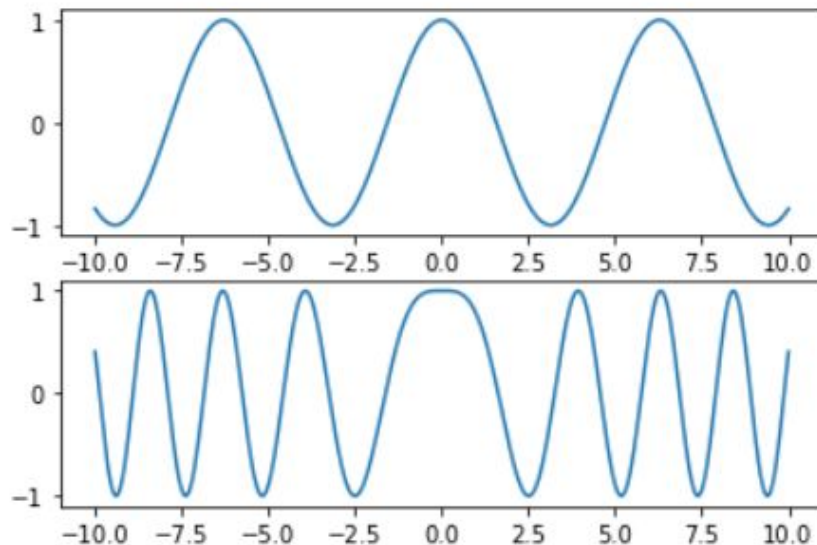
```
Text(0.5, 1.0, 'Gráfico de tortas')
```



Ejemplos

```
x = np.linspace(-10,10,400)
plt.figure(figsize=(6,4))
plt.subplot(2,1,1)
plt.plot(x,np.cos(x))
plt.subplot(2,1,2)
plt.plot(x,np.cos(x*np.log(np.abs(x)+1)))
```

[<matplotlib.lines.Line2D at 0x7f0ef4304550>]



Interfaz orientada a objetos

Hasta ahora vimos que la interfaz pyplot mantiene un registro de cuál es la figura activa y las funciones que ejecutamos tienen efecto sobre esa figura. Para que las funciones tengan efecto en otra parte tenemos que activar una nueva figura o subplot.

La interfaz orientada a objetos permite ponerle nombre a los objetos como figuras y subplots, de modo que podamos referirnos a ellos en cualquier parte y en cualquier orden. Y ahora ya no vamos a trabajar con funciones sino con métodos sobre esos objetos.



plt.subplots

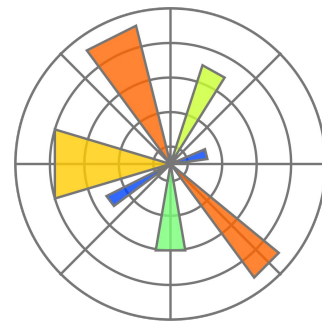
Para crear estos objetos usamos la función `subplots` (en plural) que crea una figura y uno o más subplots. Debemos pasarle la cantidad de filas y columnas que van a diagramar a los subplots (por defecto 1 y 1, es decir, una figura con un subplot).

```
fig, ax = plt.subplots()
```

Si tenemos más de un subplot el segundo elemento es un array que contiene a los objetos de cada subplot.

```
fig, axs = plt.subplots(2,2)  
print(type(axs), axs.shape)
```

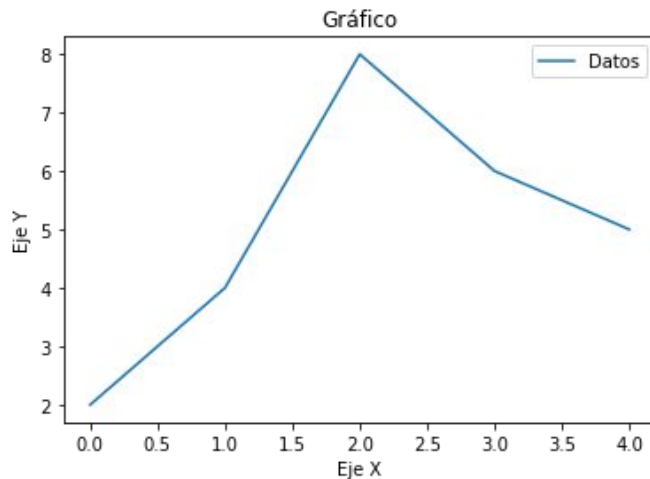
```
<class 'numpy.ndarray'> (2, 2)
```



Axes

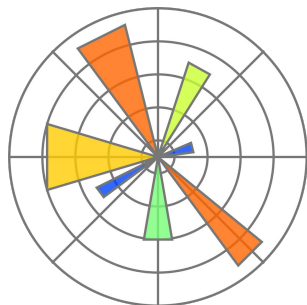
Los objetos de tipo axes tienen numerosos métodos para actuar sobre ellos. Esencialmente, toda función de pyplot tiene un método respectivo para los objetos de tipo axes.

```
fig, ax = plt.subplots()
ax.plot([2,4,8,6,5], label="Datos")
ax.set_title("Gráfico")
ax.set_xlabel("Eje X")
ax.set_ylabel("Eje Y")
ax.legend()
```



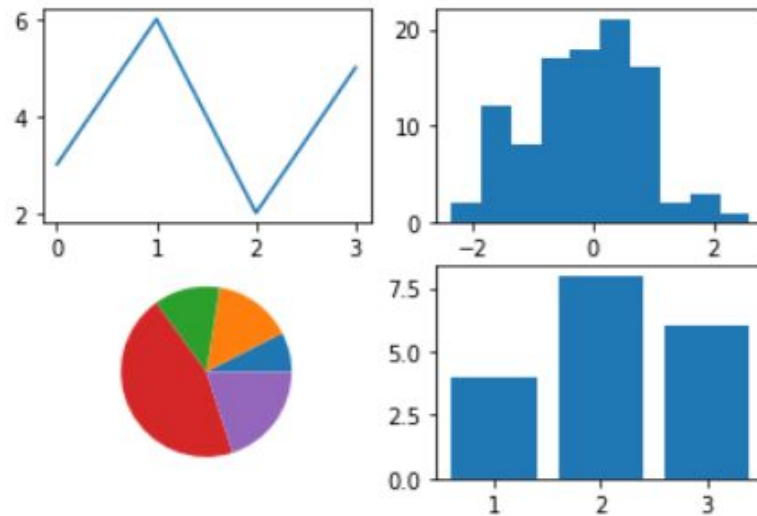
Subplots

Al hacer subgráficos tenemos que acceder a cada objeto del array usando los índices correspondientes.



```
fig, axs = plt.subplots(2,2)
axs[0,0].plot([3,6,2,5])
axs[0,1].hist(np.random.randn(100))
axs[1,0].pie([3,6,5,18,8])
axs[1,1].bar([1,2,3],[4,8,6])
```

<BarContainer object of 3 artists>



¡Muchas gracias!

¡Sigamos trabajando!