



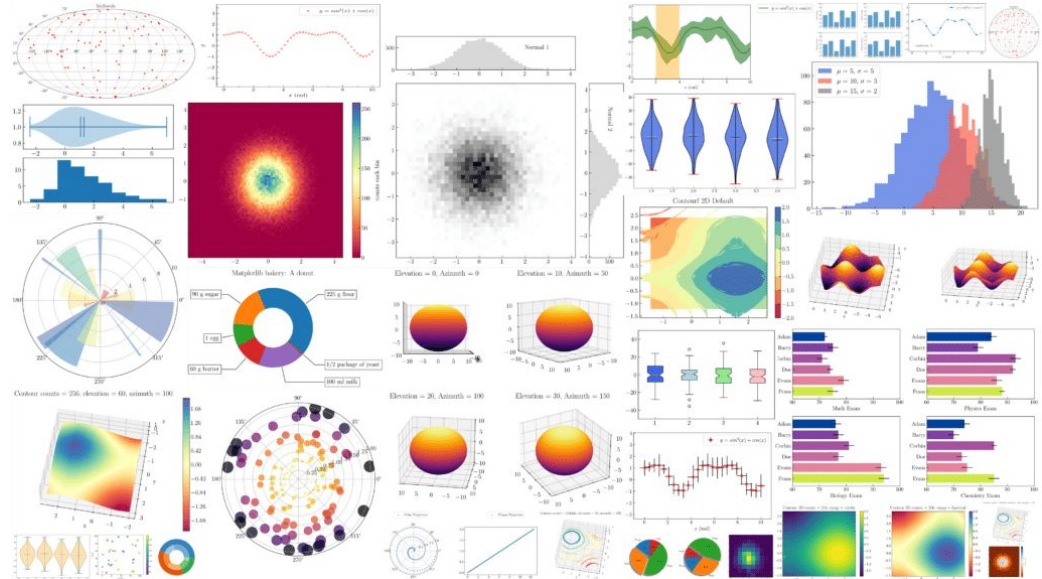
Equipo “Electroadictos” Grupo: 4CM11

¿Qué es Pyplot?

Pyplot

Pyplot es un módulo de Matplotlib que contiene muchas funciones para añadir elementos como:

- Líneas
- Imágenes o Textos



Instalación

1. `sudo apt install python3-pip` (En el caso de que no se tenga pip instalado)
2. `pip install matplotlib`

```
sistemas@arancel:~$ pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.6.2-cp38-cp38-manylinux_2_12_x86_64.manylinux2010_x86_64.whl (9.4 MB)
    |████████████████████████████████████████| 9.4 MB 820 kB/s
Collecting packaging>=20.0
  Downloading packaging-21.3-py3-none-any.whl (40 kB)
    |████████████████████████████████████████| 40 kB 491 kB/s
Requirement already satisfied: python-dateutil>=2.7 in /usr/lib/python3/dist-packages (from matplotlib) (2.7.3)
Collecting fonttools>=4.22.0
  Downloading fonttools-4.38.0-py3-none-any.whl (965 kB)
    |████████████████████████████████████████| 965 kB 1.1 MB/s
Collecting cyclor>=0.10
  Downloading cyclor-0.11.0-py3-none-any.whl (6.4 kB)
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.4.4-cp38-cp38-manylinux_2_5_x86_64.manylinux1_x86_64.whl (1.2 MB)
    |████████████████████████████████████████| 1.2 MB 780 kB/s
Collecting contourpy>=1.0.1
  Downloading contourpy-1.0.6-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (295 kB)
    |████████████████████████████████████████| 295 kB 1.4 MB/s
Requirement already satisfied: pillow>=6.2.0 in /usr/lib/python3/dist-packages (
```

Se importan con el comando `import matplotlib.pyplot as plt`, así como se muestra en el siguiente ejemplo:

```
1  import matplotlib.pyplot as plt
2  plt.plot([1, 2, 3, 4])
3  plt.ylabel('some numbers')
4  plt.show()
5
```

Funciones básicas

scatter(x, y): Dibuja un diagrama de puntos con las coordenadas de la lista x en el eje X y las coordenadas de la lista y en el eje Y.

plot(x, y): Dibuja un polígono con los vértices dados por las coordenadas de la lista x en el eje X y las coordenadas de la lista y en el eje Y.

hist(x, bins): Dibuja un histograma con las frecuencias resultantes de agrupar los datos de la lista x en las clases definidas por la lista bins.

show(): Mostrar el gráfico

savefig('imag.png'): Guardar el gráfico en formato png

Numpy. linspace (start , stop , num = 50 , endpoint = True)

Devuelve números espaciados uniformemente en un intervalo específico.

Devuelve un número de muestras espaciadas uniformemente, calculadas sobre el intervalo [start , stop].

matplotlib.pyplot. axvline (x = 0 , ymin = 0 , ymax = 1 , ** kwargs)

Agrega una línea vertical a través de los ejes.

Parámetros :

x flotante, por defecto: 0

posición x en las coordenadas de datos de la línea vertical.

flotante ymin , por defecto: 0

Debe estar entre 0 y 1, siendo 0 la parte inferior del gráfico y 1 la parte superior del gráfico.

flotante ymax , predeterminado: 1

Debe estar entre 0 y 1, siendo 0 la parte inferior del gráfico y 1 la parte superior del gráfico.

matplotlib.pyplot. axhline (y = 0 , xmin = 0 , xmax = 1 , ** kwargs)

Agregue una línea horizontal a través de los ejes.

Parámetros :

y flotante, por defecto: 0

posición y en las coordenadas de datos de la línea horizontal.

flotador xmin , por defecto: 0

Debe estar entre 0 y 1, siendo 0 el extremo izquierdo del gráfico y 1 el extremo derecho del gráfico.

flotador xmax , predeterminado: 1

Debe estar entre 0 y 1, siendo 0 el extremo izquierdo del gráfico y 1 el extremo derecho del gráfico.

matplotlib.pyplot. axline (xy1 , xy2 = Ninguno , * , pendiente = Ninguno , ** kwargs)

Agrega una línea recta infinitamente larga.

La línea se puede definir por dos puntos xy1 y xy2 , o por un punto xy1 y una pendiente .

Parámetros :

xy1, xy2 (flotante, flotante)

Puntos por los que pasa la línea. Se debe dar xy2 o la pendiente .

flotador de pendiente , opcional

La pendiente de la línea. Se debe dar xy2 o la pendiente .

Numpy. arange([inicio ,] parada , [paso ,] dtype=Ninguno , * , like=Ninguno)

Devuelve valores espaciados uniformemente dentro de un intervalo dado.

- **arange(parada):** Los valores se generan dentro del intervalo semiabierto [0, stop) (en otras palabras, el intervalo que incluye el inicio pero excluye la parada).
- **arange(inicio, parada):** Los valores se generan dentro del intervalo semiabierto [start, stop].
- **arange(inicio, parada, paso)** Los valores se generan dentro del intervalo semiabierto [start, stop), con el espacio entre valores dado por paso.

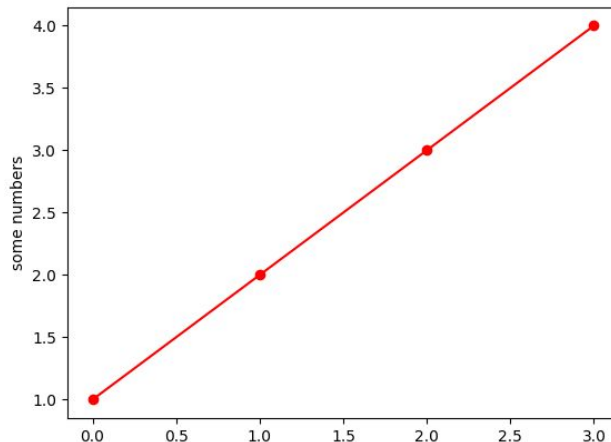
Tipos principales que ofrece Pyplot

Simple

Se traza una lista de números contra su índice. Crea una línea recta debido a que la tasa de cambio es 1 para los ejes X e Y. Utilice una cadena de formato (aquí, 'o-r') para establecer los marcadores (círculos), el estilo de línea (línea continua) y el color (rojo).

```
import matplotlib.pyplot as plt

plt.plot([1, 2, 3, 4], 'o-r')
plt.ylabel('some numbers')
plt.show()
```



Líneas Infinitas

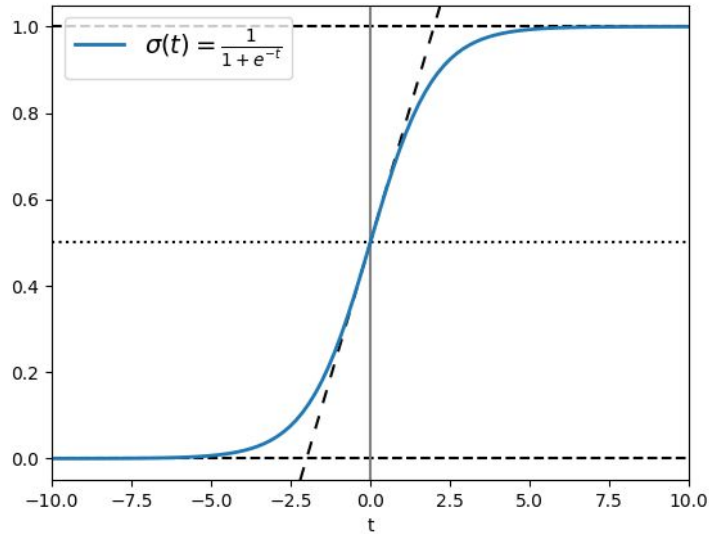
Axvline y **axhline** dibujan líneas infinitas verticales/horizontales, en posiciones x / y dadas . Por lo general, se utilizan para marcar valores de datos especiales, en este ejemplo el centro y los valores límite de la función sigmoide.

Axline dibuja infinitas líneas rectas en direcciones arbitrarias.

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

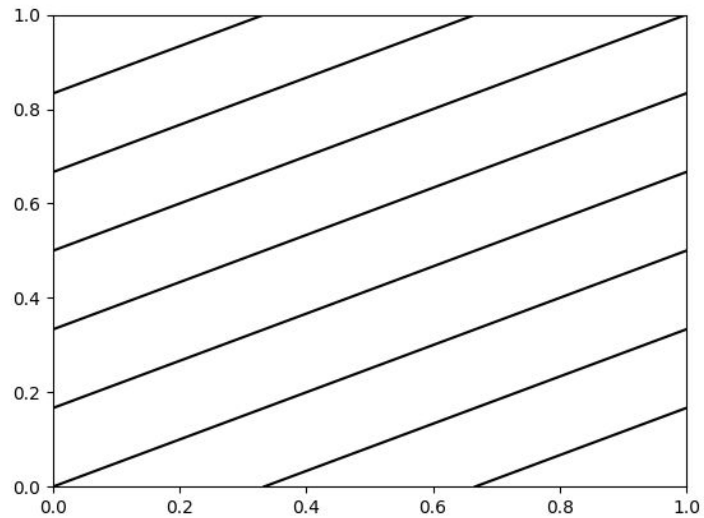
t = np.linspace(-10, 10, 100)
sig = 1 / (1 + np.exp(-t))

plt.axhline(y=0, color="black", linestyle="--")
plt.axhline(y=0.5, color="black", linestyle=":")
plt.axhline(y=1.0, color="black", linestyle="--")
plt.axvline(color="grey")
plt.axline((0, 0.5), slope=0.25, color="black", linestyle=(0, (5, 5)))
plt.plot(t, sig, linewidth=2, label=r"$\sigma(t) = \frac{1}{1 + e^{-t}}$")
plt.xlim(-10, 10)
plt.xlabel("t")
plt.legend(fontsize=14)
plt.show()
```



Axline también se puede usar con un parámetro de “transform”, que se aplica al punto, pero no a la pendiente. Esto puede ser útil para dibujar líneas de cuadrícula diagonales con una pendiente fija, que permanecen en su lugar cuando se mueven los límites de la parcela.

```
for pos in np.linspace(-2, 1, 10):  
    plt.axline((pos, 0), slope=0.5, color='k', transform=plt.gca().transAxes)  
  
plt.ylim([0, 1])  
plt.xlim([0, 1])  
plt.show()
```



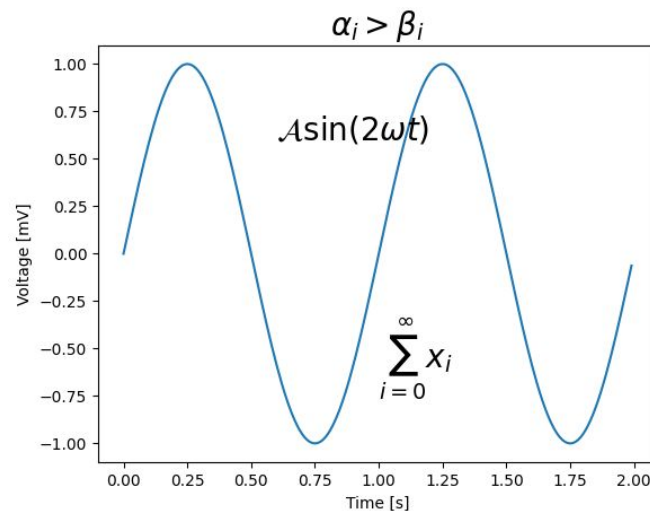
Matemático

Utiliza las expresiones matemáticas

NumPy es una biblioteca para el lenguaje de programación Python que da soporte para crear vectores y matrices grandes.

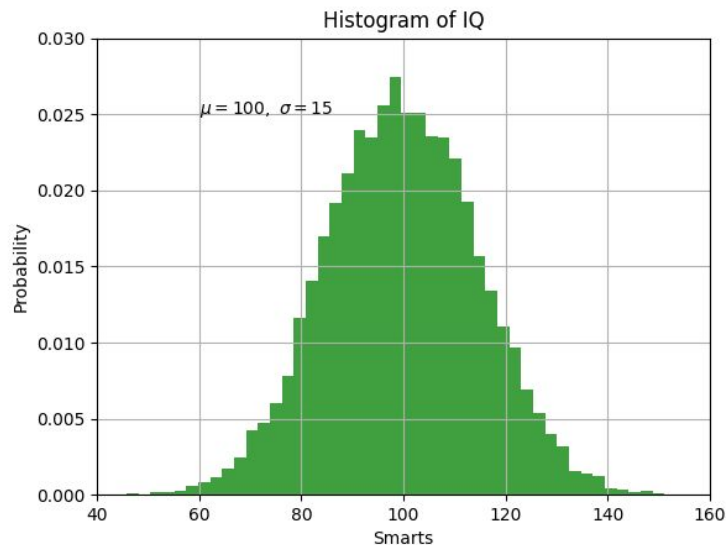
```
import numpy as np
import matplotlib.pyplot as plt
t = np.arange(0.0, 2.0, 0.01)
s = np.sin(2*np.pi*t)

plt.plot(t, s)
plt.title(r'$\alpha_i > \beta_i$', fontsize=20)
plt.text(1, -0.6, r'$\sum_{i=0}^{\infty} x_i$', fontsize=20)
plt.text(0.6, 0.6, r'$\mathcal{A}\mathrm{sin}(2 \ \omega \ t)$',
         fontsize=20)
plt.xlabel('Time [s]')
plt.ylabel('Voltage [mV]')
plt.show()
```



Histogramas

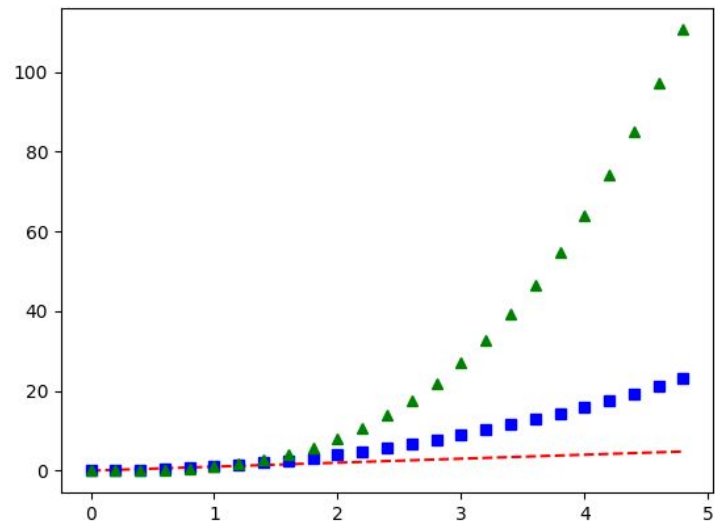
```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Fixing random state for reproducibility
5 np.random.seed(19680801)
6
7 mu, sigma = 100, 15
8 x = mu + sigma * np.random.randn(10000)
9
10 # the histogram of the data
11 n, bins, patches = plt.hist(x, 50, density=True, facecolor='g', alpha=0.75)
12
13
14 plt.xlabel('Smarts')
15 plt.ylabel('Probability')
16 plt.title('Histogram of IQ')
17 plt.text(60, .025, r'$\mu=100,\ \sigma=15$')
18 plt.xlim(40, 160)
19 plt.ylim(0, 0.03)
20 plt.grid(True)
21 plt.show()
```



Tres

Traza tres diagramas de líneas en una sola llamada a plot.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # evenly sampled time at 200ms intervals
5 t = np.arange(0., 5., 0.2)
6
7 # red dashes, blue squares and green triangles
8 plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
9 plt.show()
```



Dos subtramas o más

Ejemplo, se crea una figura con dos subtramas

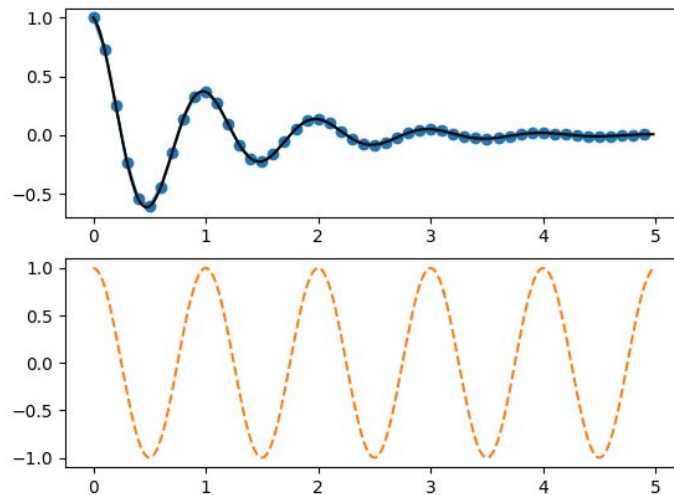
```
import numpy as np
import matplotlib.pyplot as plt

def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)

plt.figure()
plt.subplot(211)
plt.plot(t1, f(t1), color='tab:blue', marker='o')
plt.plot(t2, f(t2), color='black')

plt.subplot(212)
plt.plot(t2, np.cos(2*np.pi*t2), color='tab:orange', linestyle='--')
plt.show()
```



Diferencias Pyplot y Pylab

Recordar matplotlib es el paquete completo, como se mencionó antes pyplot es un módulo de este, pero también se cuenta con otro módulo llamado Pylab.

- Pyplot proporciona la interfaz estado-máquina a la biblioteca de trazado subyacente en matplotlib. Esto significa que las figuras y los ejes se crean implícita y automáticamente para lograr el trazado deseado. Por ejemplo, si se llama a plot desde pyplot, se crearán automáticamente las figuras y los ejes necesarios para conseguir el gráfico deseado. Si se establece un título, éste se asignará automáticamente al objeto de ejes actual.
- Pylab combina la funcionalidad de pyplot (para trazar) con la de numpy (para las matemáticas y para trabajar con arrays) en un único espacio de nombres, haciendo que ese espacio de nombres (o entorno) sea aún más parecido a MATLAB. Por ejemplo, uno puede llamar a las funciones sin y cos igual que lo haría en MATLAB, además de tener todas las características de pyplot.

Anexo

Colores base



Tableau Palette



Colores CSS

