

Análisis Exploratorio y Programación Estadística

La Librería Matplotlib

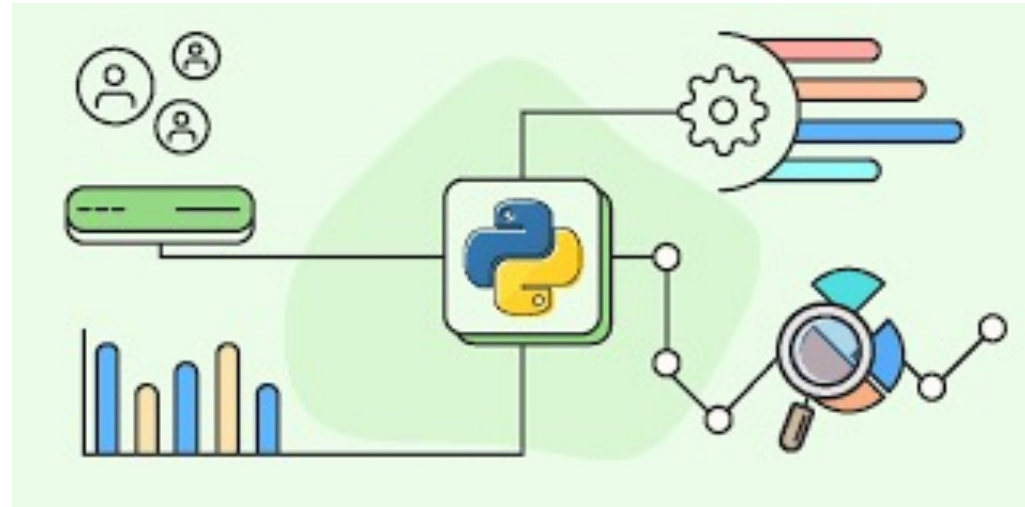
Especialización en Ciencia de Datos

2021



Objetivos

- Aprender sobre visualización de datos
- Aprender a usar la librería matplotlib



Contenido:

1. Visualización de datos
2. La Librería Matplotlib
3. Comandos Básicos
4. Múltiples gráficos



1. Visualización de datos



Por qué visualizar los datos?



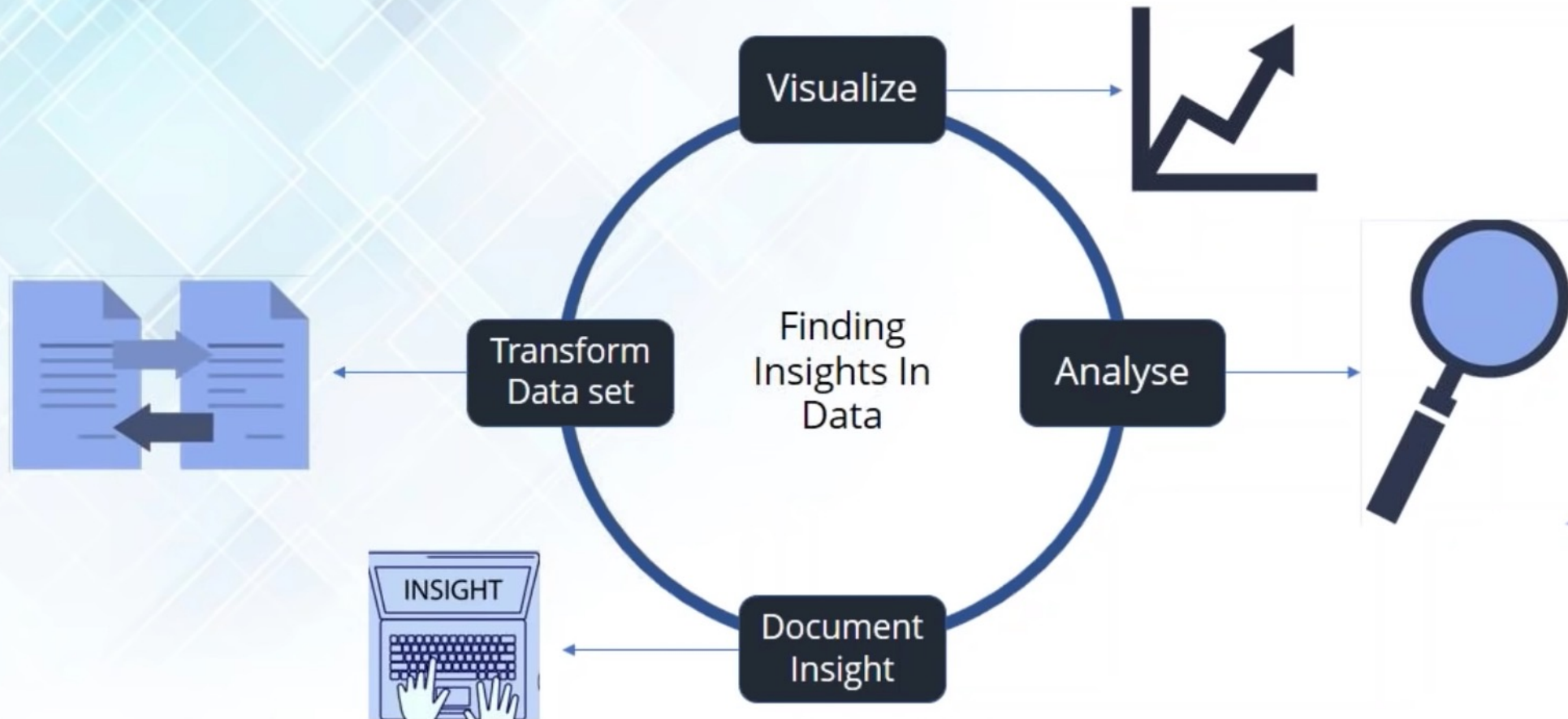
Por qué visualizar los datos?



Nos permite visualizar rápidamente interpretar los datos y su comportamiento. Nos permite ajustar las diferentes variables

Por qué visualizar los datos?

Data visualization is the presentation of data in a pictorial or graphical format.



Librerías de visualización

Python cuenta con varias librerías para visualización las principales son:

- [matplotlib](#) para gráficas sencillas: bars, pies, lines, scatter plots, etc.
- [Seaborn](#) para visualización estadística: Para crear mapas de calor o de alguna manera resumiendo los datos y aún desea mostrar la distribución de los datos.
- [Plotly](#) y [Bokeh](#) para visualización interactiva: Si los datos son tan complejos (o no puede ver la información de sus datos), utilice plotly y Bokeh para crear visualizaciones interactivas que permitan a los usuarios explorar los datos mismos.



2. La librería Matplotlib

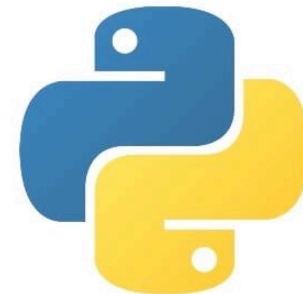


Python Matplotlib

- **matplotlib.pyplot** es una librería para graficar usada para generar gráficos en 2D en python.
- Se puede utilizar en scripts de Python, shell, servidores de aplicaciones web y otros kits de herramientas de interfaz gráfica de usuario.
- Matploitlib es una biblioteca de Python que se utiliza para trazar, esta biblioteca de Python proporciona API orientadas a objetos para integrar trazados en aplicaciones.

<https://matplotlib.org/>

matplotlib

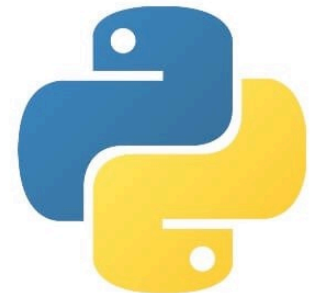


Características Matplotlib

Algunos de los principales Pros de Matplotlib son:

- Generalmente es fácil comenzar por graficas simples
- Soporte para etiquetas personalizadas y textos
- Gran control de cada elemento en una figura
- Salida de alta calidad en muchos formatos
- Muy personalizable en general

Matplotlib le permite crear figuras reproducibles mediante programación.



Instalación Matplotlib

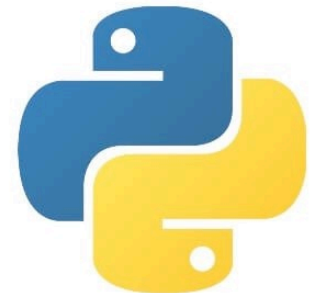
Se debe instalar Matplotlib, pero si instalo Anaconda ya viene instalado, en caso de que no lo tenga se puede instalar así:

`pip install matplotlib` o `conda install matplotlib`

en Jupyter notebook `!pip install matplotlib` Diagramas de barras

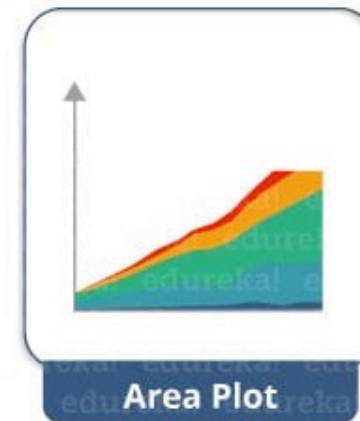
Importar el modulo `matplotlib.pyplot` con el nombre de `plt` (esto es un estandar en la comunidad):

```
import matplotlib.pyplot as plt
```

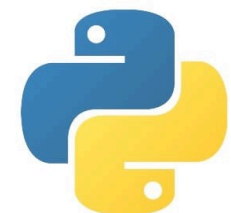


Python Matplotlib: Tipos de gráficos

Hay varios gráficos que se pueden crear usando python matplotlib. Algunos de ellos se enumeran a continuación:



matplotlib



3. Comandos Básicos



Comandos Básicos

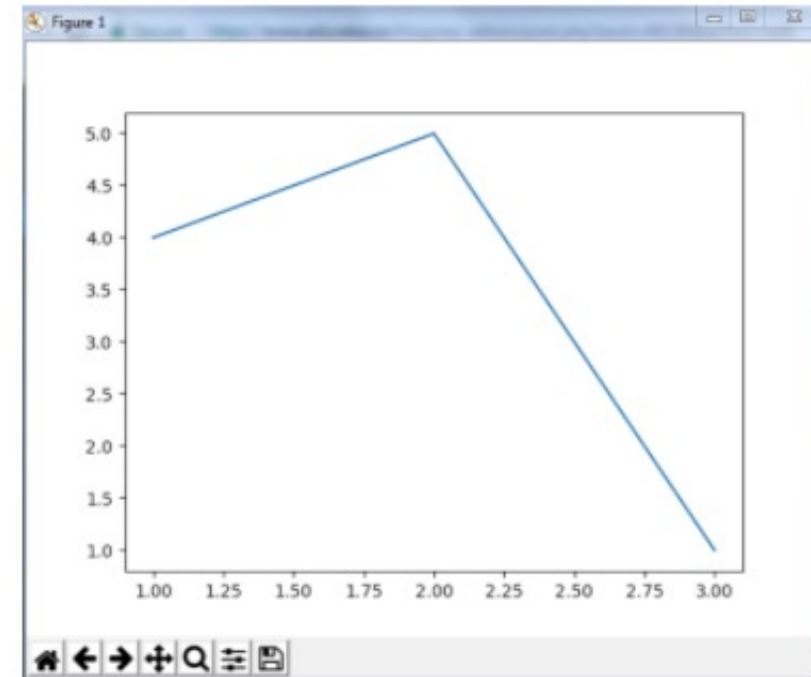
Déjame mostrarte códigos muy básicos en python matplotlib para generar un gráfico simple.

Output -

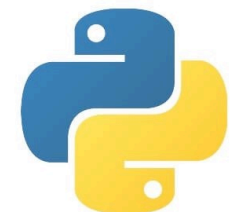
```
1 from matplotlib import pyplot as plt
2
3 #Plotting to our canvas
4 plt.plot([1,2,3],[4,5,1])
5
6 #Showing what we plotted
7
8 plt.show()
9
```

```
%matplotlib inline
```

Con tres líneas de código, puede generar un gráfico básico usando python matplotlib. Simple, ¿no es así?



matplotlib

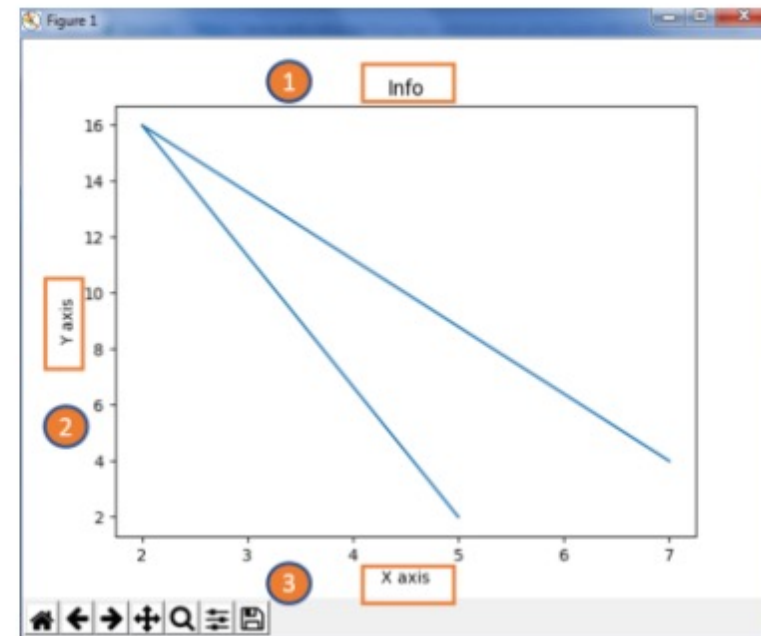


Comandos Básicos

Veamos cómo podemos agregar título, etiquetas a nuestro gráfico creado por la biblioteca matplotlib de Python para darle más significado.

```
1 from matplotlib import pyplot as plt
2
3 x = [5,2,7]
4 y = [2,16,4]
5 plt.plot(x,y)
6 plt.title('Info')
7 plt.ylabel('Y axis')
8 plt.xlabel('X axis')
9 plt.show()
```

Output -



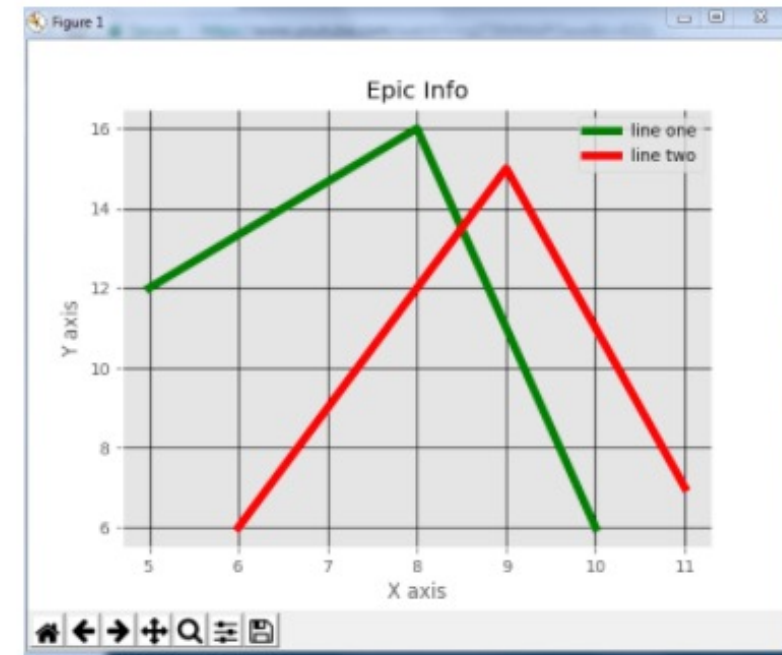
Incluso puede probar muchas técnicas de estilo para crear un mejor gráfico. ¿Qué sucede si desea cambiar el ancho o el color de una línea en particular o qué sucede si desea tener algunas líneas de cuadrícula, allí necesita un estilo?

Comandos Básicos

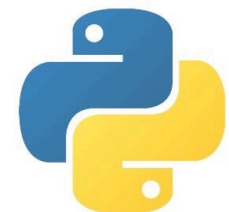
Déjame mostrarte cómo agregar estilo a un gráfico usando Python matplotlib.

```
1 from matplotlib import pyplot as plt
2 from matplotlib import style
3
4 style.use('ggplot')
5 x = [5,8,10]
6 y = [12,16,6]
7 x2 = [6,9,11]
8 y2 = [6,15,7]
9 plt.plot(x,y,'g',label='line one', linewidth=5)
10 plt.plot(x2,y2,'c',label='line two',linewidth=5)
11 plt.title('Epic Info')
12 plt.ylabel('Y axis')
13 plt.xlabel('X axis')
14 plt.legend()
15 plt.grid(True,color='k')
16 plt.show()
```

Output -



matplotlib



Tamaño de la Figura y DPI

Matplotlib permite especificar la relación de aspecto, el DPI y el tamaño de la figura cuando se crea el objeto Figure. Puede usar los argumentos de las palabras clave `figsize` y `dpi`. No es necesario poner las dos.

`figsize` es una tupla del ancho y alto de la figura en pulgadas

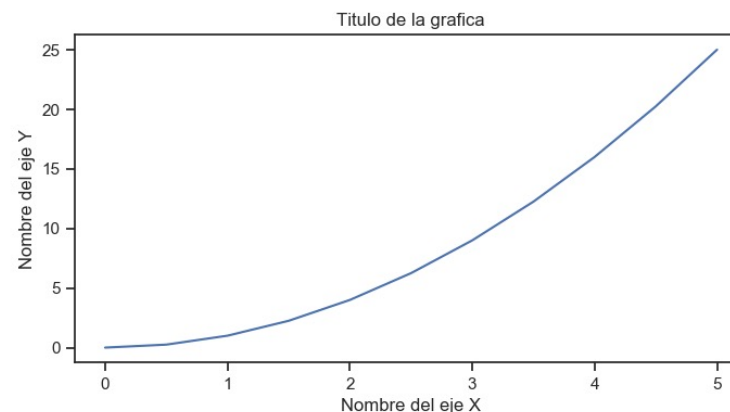
`dpi` es el punto por pulgada (pixel por pulgada).

```
# se cambia el tamaño de la figura y el numero de puntos por pulgada
plt.figure(figsize=(8,4), dpi=100)

plt.plot(x, y) # se grafica una linea de color azul

plt.xlabel('Nombre del eje X') # definir el nombre del eje X
plt.ylabel('Nombre del eje Y') # definir el nombre del eje Y
plt.title('Titulo de la grafica'); # definir el titulo de la grafica

# agrego ; al final del ultimo comando para solo mostrar la grafica
# plt.show() no es necesario en jupyter notebook
```



Escala Logarítmica

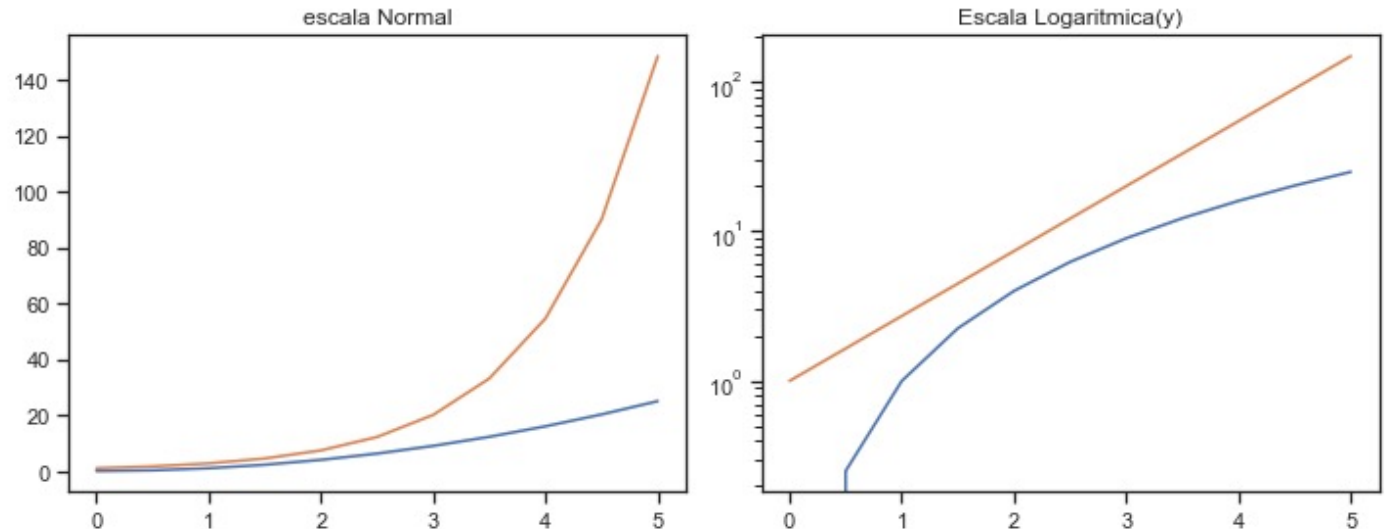
Anotar texto en figuras matplotlib se puede hacer usando la función text. Es compatible con el formato LaTeX al igual que los textos y títulos de la etiqueta del eje:

```
plt.figure(figsize=(10,4))

plt.subplot(1,2,1)
plt.plot(x, x**2, x, np.exp(x))
plt.title("escala Normal")

plt.subplot(1,2,2)
plt.plot(x, x**2, x, np.exp(x))
plt.yscale("log")
plt.title("Escala Logaritmica(y)");

plt.tight_layout() # para que no se superpongan las gra
```



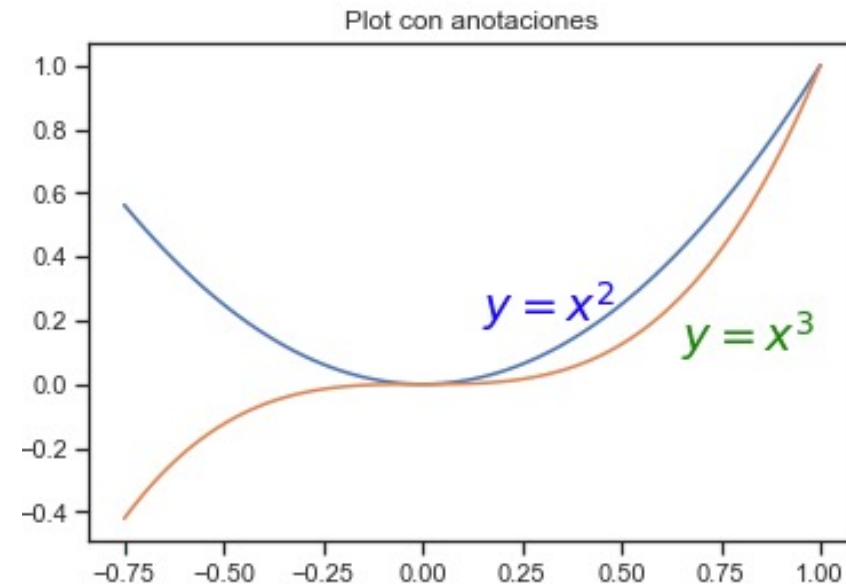
Anotaciones de texto

Anotar texto en figuras matplotlib se puede hacer usando la función `text`. Es compatible con el formato LaTeX al igual que los textos y títulos de la etiqueta del eje:

```
# Datos para graficar
xx = np.linspace(-0.75, 1., 100)

plt.plot(xx, xx**2, xx, xx**3)
plt.title("Plot con anotaciones")

# Anotacion 1
plt.text(0.15, 0.2, r"$y=x^2$", fontsize=20, color="blue")
#Anotacion 2
plt.text(0.65, 0.1, r"$y=x^3$", fontsize=20, color="green");
```

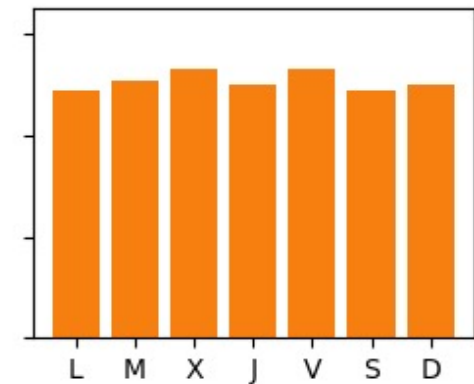
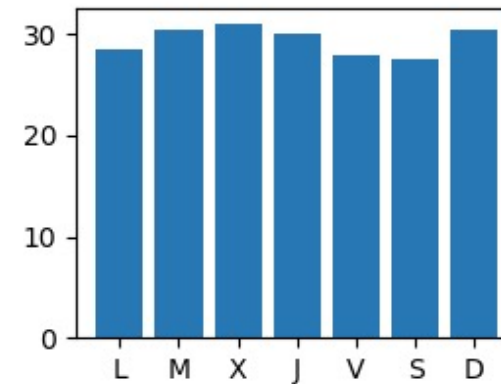
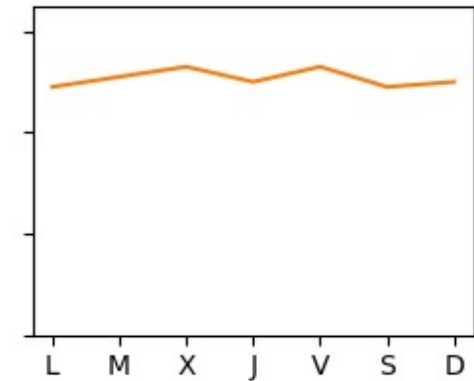
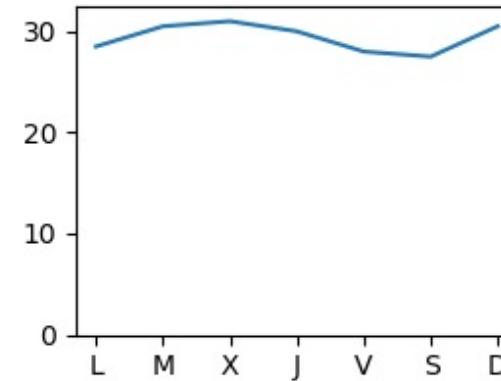


4. Múltiples gráficos



Múltiples gráficos

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(2, 2, sharey = True)
dias = ['L', 'M', 'X', 'J', 'V', 'S', 'D']
temperaturas = {'Madrid':[28.5, 30.5, 31, 30, 28, 27.5, 30.5], 'Barcelona':[24.5, 25.5, 26, 25, 24, 23.5, 25.5]}
ax[0, 0].plot(dias, temperaturas['Madrid'])
ax[0, 1].plot(dias, temperaturas['Barcelona'], color = 'tab:orange')
ax[1, 0].bar(dias, temperaturas['Madrid'])
ax[1, 1].bar(dias, temperaturas['Barcelona'], color = 'tab:orange')
plt.show()
```



Integración con Pandas

Matplotlib se integra a la perfección con la librería Pandas, permitiendo dibujar gráficos a partir de los datos de las series y DataFrames de Pandas.

`df.plot(kind=tipo, x=columnax, y=columnay, ax=ejes) :`

Dibuja un diagrama del tipo indicado por el parámetro kind en los ejes indicados en el parámetro ax, representando en el eje x la columna del parámetro x y en el eje y la columna del parámetro y.

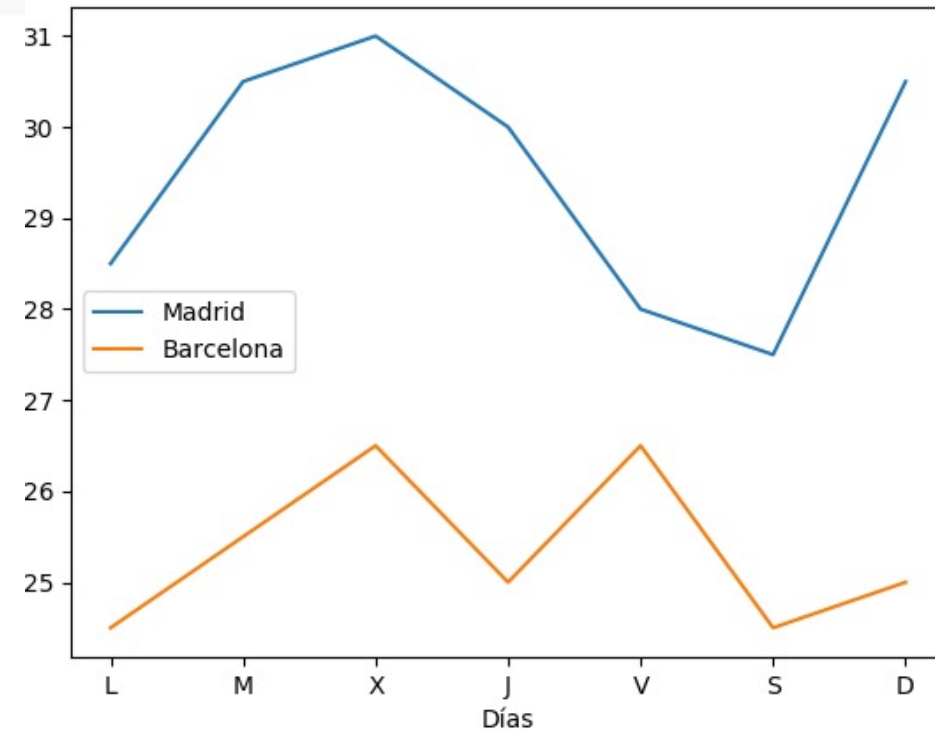
El parámetro kind puede tomar como argumentos:

'line' (líneas), 'scatter' (puntos), 'bar' (barras verticales), 'barh' (barras horizontales), 'hist' (histograma), 'box' (cajas), 'density' (densidad), 'area' (area) o 'pie' (sectores).



Integración con Pandas

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.DataFrame({'Días':['L', 'M', 'X', 'J', 'V', 'S', 'D'],
                  'Madrid':[28.5, 30.5, 31, 30, 28, 27.5, 30.5],
                  'Barcelona':[24.5, 25.5, 26.5, 25, 26.5, 24.5, 25]})
fig, ax = plt.subplots()
df.plot(x = 'Días', y = 'Madrid', ax = ax)
df.plot(x = 'Días', y = 'Barcelona', ax = ax)
plt.show()
```



A Practicar!!!!





 Software Engineering |  IT Staffing |  IT Academy |  IT Consulting

Proyecto apoyado por
CORFO

