Gráficas con Matplotlib, Polinomios e Interpolación

MAT Y TEC II

Prof. Robert Muñoz

Escuela de Matemática,
Facultad de Ciencias,
UASD

2025

Tabla de Contenido

Matplotlib

- Matplotlib es una librería completa para crear visualizaciones estáticas, animadas e interactivas en Python.

Matplotlib

- Matplotlib es una librería completa para crear visualizaciones estáticas, animadas e interactivas en Python.
- Visualización de datos en 2D y 3D.

Matplotlib

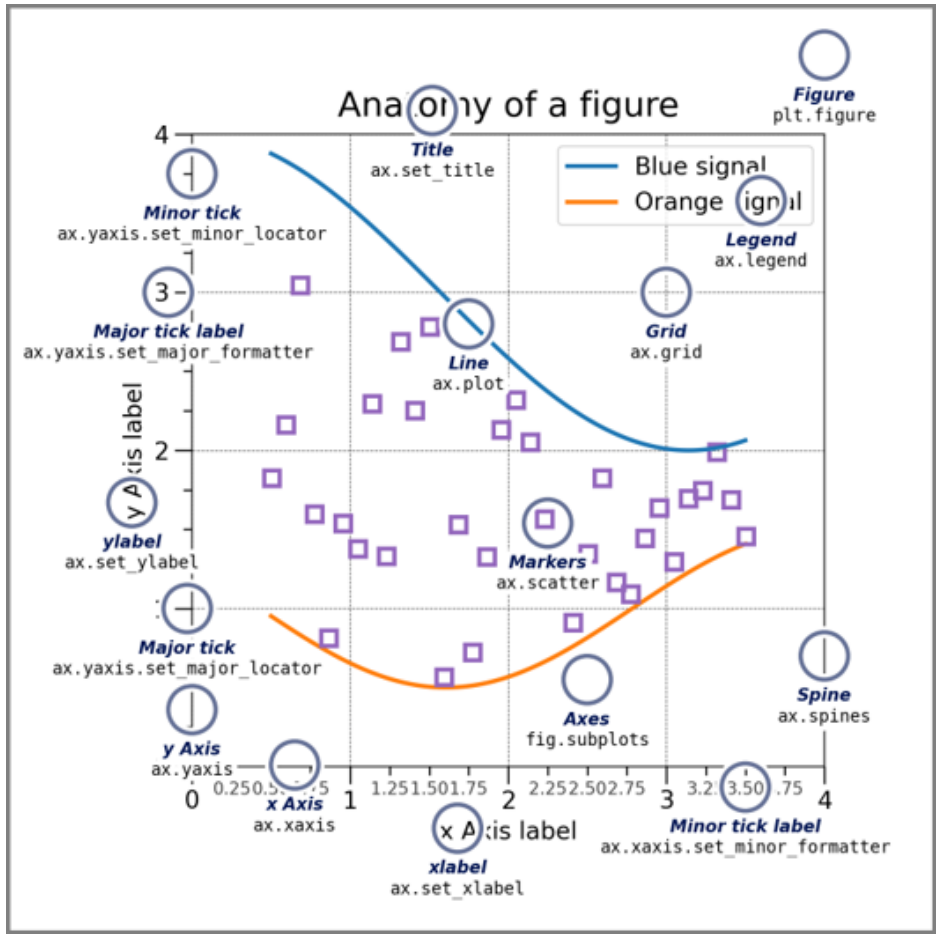
- Matplotlib es una librería completa para crear visualizaciones estáticas, animadas e interactivas en Python.
- Visualización de datos en 2D y 3D.
- Animaciones y procesamiento de imágenes.

Matplotlib

- Matplotlib es una librería completa para crear visualizaciones estáticas, animadas e interactivas en Python.
- Visualización de datos en 2D y 3D.
- Animaciones y procesamiento de imágenes.
- Aplicaciones comunes en ciencia de datos, ingeniería, finanzas, matemática, física, biología y más.

Matplotlib

- Matplotlib es una librería completa para crear visualizaciones estáticas, animadas e interactivas en Python.
- Visualización de datos en 2D y 3D.
- Animaciones y procesamiento de imágenes.
- Aplicaciones comunes en ciencia de datos, ingeniería, finanzas, matemática, física, biología y más.



Instalación e Importación

Instalación

!pip install matplotlib (en el notebook).

pip install matplotlib (en terminal de anaconda).

Instalación e Importación

Instalación

!pip install matplotlib (en el notebook).
pip install matplotlib (en terminal de anaconda).

Importación

import matplotlib.pyplot as plt

Instalación e Importación

Instalación

!pip install matplotlib (en el notebook).
pip install matplotlib (en terminal de anaconda).

Importación

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

Info: Pyplot es un submódulo de matplotlib que proporciona una interfaz fácil de usar para crear gráficos.

Gráfico de Línea

El gráfico de línea es una de las formas más comunes de visualizar datos.
Para crear un gráfico de línea básico, podemos utilizar los siguientes comandos:

```
1 x=[1,3,4,6] #dominio
2 y=[10,8,5,3] #rango
3 fig, ax=plt.subplots() #comando de matplotlib
4 ax.plot(x,y)
```

Gráfico de Línea con arrays

También podemos usar arrays...

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 x=np.arange(-3,4) #dominio
4 y=x**2 #rango
5 fig, ax=plt.subplots() #comando de matplotlib
6 ax.plot(x,y)
```

Gráfico de Dispersión

Utilizando los mismos datos del ejemplo anterior ....

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 x=np.arange(-3,4) #dominio
4 y=x**2 #rango
5 fig, ax=plt.subplots() #comando de matplotlib
6 ax.scatter(x,y)
```

Gráfico de barras

Utilizando los mismos datos del ejemplo anterior ....

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 x=np.arange(-3,4) #dominio
4 y=x**2 #rango
5 fig, ax=plt.subplots() #comando de matplotlib
6 ax.bar(x,y)
```

Guardar un gráfico

Utilizando los mismos datos del ejemplo anterior ....

```
1  import matplotlib.pyplot as plt
2  # Datos
3  x = [1, 2, 3, 4, 5]
4  y = [2, 3, 5, 7, 11]
5  # Crear figura y ejes
6  fig, ax = plt.subplots()
7  ax.plot(x, y)
8  # A\~nadir t\'itulo y etiquetas
9  ax.set_title("Gr\'afico de ejemplo")
10 ax.set_xlabel("Eje X")
11 ax.set_ylabel("Eje Y")
12 # Guardar gr\'afico
13 fig.savefig("grafico_ejemplo.png")
```

Personalización de los gráficos

Algunas de las opciones de personalización incluyen:

- Títulos y etiquetas de ejes
- Puntos personalizados
- Colores y estilos de línea
- Rangos de ejes personalizados.
- ...

Personalización de los gráficos

Algunas de las opciones de personalización incluyen:

- Títulos y etiquetas de ejes
- Puntos personalizados
- Colores y estilos de línea
- Rangos de ejes personalizados.
- ...

VER NOTEBOOK de la clase...

Otros gráficos

- Gráfico de barras horizontal y vertical
- Gráfico de pastel
- Histogramas
- Gráfico de cajas
- 3D
- Gráficas en coordenadas polares

Para seguir explorando: https://matplotlib.org/

Definición de un Polinomio

Supongamos que queremos definir el polinomio $p(x) = 2x^3 + 3x^2 - 5x + 7$. Podemos usar poly1d de numpy de la siguiente manera:

Definición de un Polinomio

Supongamos que queremos definir el polinomio $p(x) = 2x^3 + 3x^2 - 5x + 7$. Podemos usar poly1d de numpy de la siguiente manera:

```
1 import numpy as np
2
3 # Definir el polinomio
4 coeficientes = [2, 3, -5, 7]
5 polinomio = np.poly1d(coeficientes)
6
7 print(polinomio)
```

Comandos aplicables a un polinomio

- Evaluación
- Raíces
- Integración
- Derivación
- Operaciones $+, -, x, /$
- ...

Ver notebook de la clase...

Interpolación Polinomial

Se pretende encontrar el polinomio que mejor se ajuste a estos datos. Para ello, podemos utilizar la función numpy.polyfit(), que realiza una regresión polinómica sobre los datos de entrada. El primer argumento de esta función es el conjunto de datos en el eje x, el segundo es el conjunto de datos en el eje y y el tercer argumento es el grado del polinomio que queremos ajustar.

Ejemplo de Interpolación

```
1  # Datos datos
2  x = np.array([0, 1, 2, 3, 4, 5])
3  y = np.array([0, 1, 4, 9, 16, 25])
4
5  # Ajustar grado 2
6  grado = 2
7  coeficientes = np.polyfit(x, y, grado)
8
9  # Creaci\'on polinomio
10 polinomio = np.poly1d(coeficientes)
11
12 # Imprimir el polinomio
13
14 print(polinomio)
```