

Argentina Programa

Muestreo de datos Matplotlib



Matplotlib

Matplotlib es una biblioteca popular de Python utilizada para la visualización de datos en 2D y 3D. Esta potente herramienta permite crear una amplia variedad de gráficos, desde simples diagramas de líneas hasta sofisticadas visualizaciones tridimensionales. En este tutorial, te presentaré una introducción a Matplotlib en Jupyter Notebook y te mostraré algunas de las funciones más utilizadas para crear gráficos y realizar análisis de datos.

Gráfico de Línea

El gráfico de líneas es uno de los tipos de gráficos más utilizados para representar datos continuos, como series de tiempo o tendencias. Para crear un gráfico de líneas básico, podemos usar la función `plot()` de Matplotlib.

```
# Datos de ejemplo
x = np.linspace(0, 10, 100)
y = np.linspace(0, 10, 100)
# Crear el gráfico de líneas
plt.plot(x, y)
plt.title('Gráfico de Líneas')
plt.xlabel('Eje X')
plt.ylabel('Eje Y')
plt.grid(True)
plt.show()
```

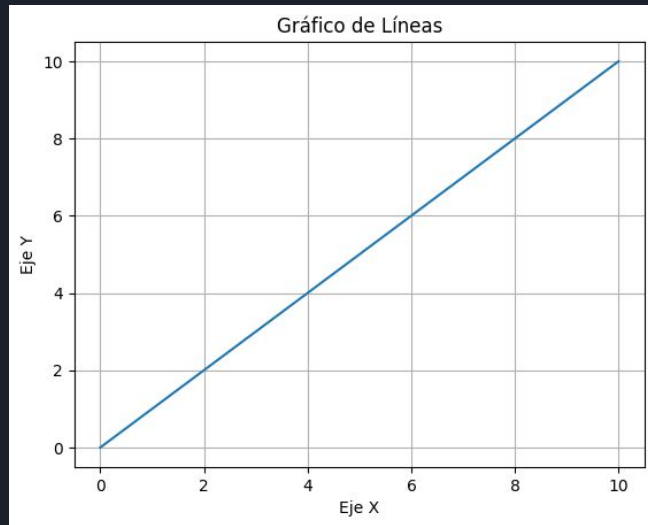


Gráfico de Barras

Los gráficos de barras son útiles para comparar diferentes categorías o valores discretos.

```
# Datos de ejemplo
categorias = ['A', 'B', 'C', 'D', 'E']
valores = [25, 18, 10, 32, 15]
# Crear el gráfico de barras
plt.bar(categorias, valores)
plt.title('Gráfico de Barras')
plt.xlabel('Categorías')
plt.ylabel('Valores')
plt.show()
```

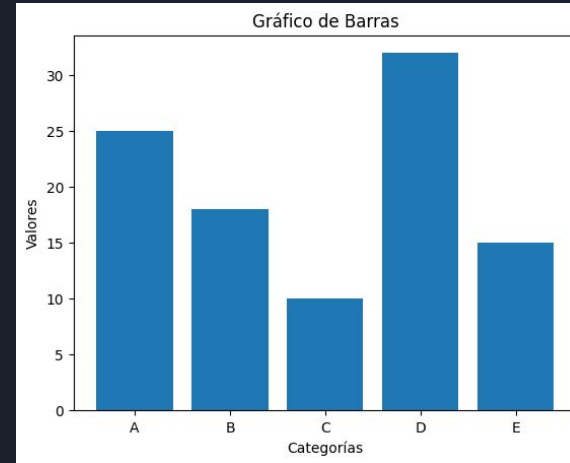


Gráfico de torta

El gráfico de torta muestra las proporciones de diferentes categorías en un todo.

```
#Datos
```

```
categorias = ['A', 'B', 'C', 'D', 'E']
```

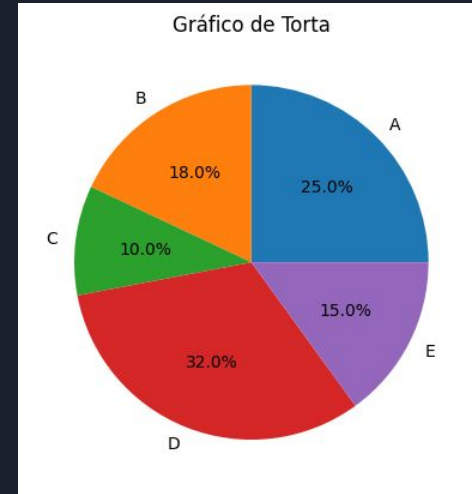
```
valores = [25, 18, 10, 32, 15]
```

```
# Crear el gráfico de pastel
```

```
plt.pie(valores, labels=categorias, autopct='%1.1f%%')
```

```
plt.title('Gráfico de Torta')
```

```
plt.show()
```



Gráficos Para Análisis de Datos

Histograma

Los histogramas son útiles para visualizar la distribución de un conjunto de datos.

```
# Datos
datos = np.random.randn(1000)
# Crear el histograma
plt.hist(datos, bins=20)
plt.title('Histograma')
plt.xlabel('Valores')
plt.ylabel('Frecuencia')
plt.show()
```

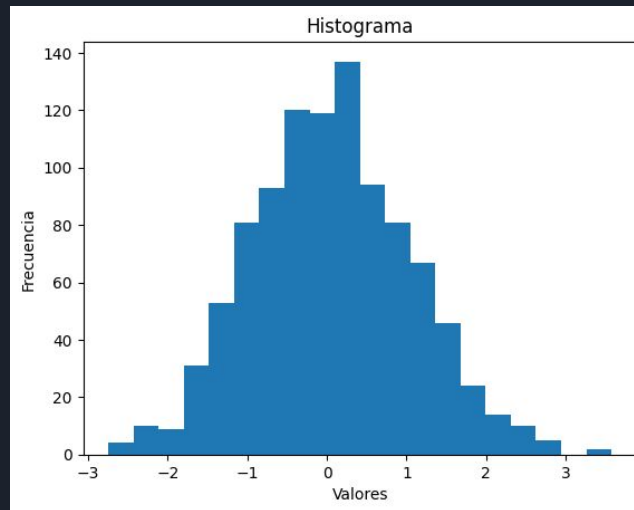
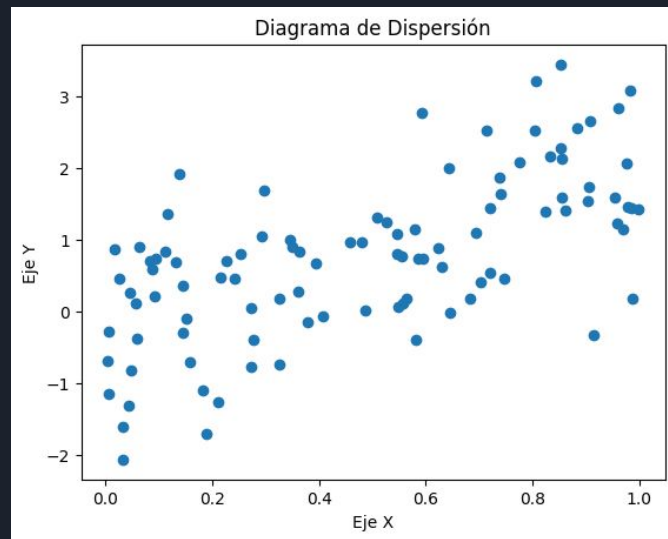


Gráfico para Análisis de Datos

Dispersión

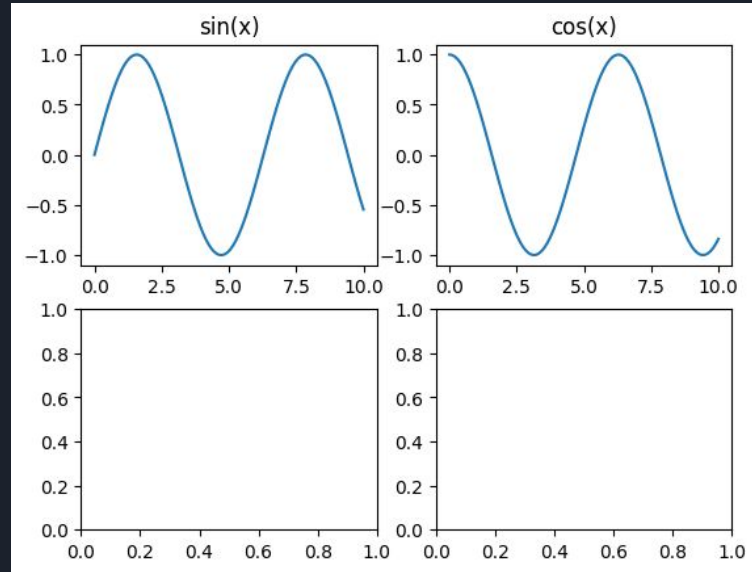
El diagrama de dispersión es útil para visualizar la relación entre dos conjuntos de datos.

```
#Datos
x = np.random.rand(100)
y = 2 * x + np.random.randn(100)
# Crear el diagrama de dispersión
plt.scatter(x, y)
plt.title('Diagrama de Dispersión')
plt.xlabel('Eje X')
plt.ylabel('Eje Y')
plt.show()
```



Adicional Gráfica Múltiple

```
# Crear Subtramas
fig, axes = plt.subplots(2, 2)
x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)
axes[0, 0].plot(x, y1)
axes[0, 0].set_title('sin(x)')
axes[0, 1].plot(x, y2)
axes[0, 1].set_title('cos(x)')
plt.show()
```





Para Más Funciones

<https://colab.research.google.com/drive/1OzIUUhZOBO0AxBmSdEFUmkOrWeD3vvU#scrollTo=NNIGTWT0GO2v>