

Documentación librería Matplotlib

Gráfica funcional

```
1. x = np.linspace(0, 5, 10)
2. y = x**2
3.
4. plt.plot(x, y)
5. plt.title("Ejemplo de gráfica funcional")
6. plt.xlabel("Eje X")
7. plt.ylabel("Eje Y")
8.
```

Se crea una gráfica simple de una función cuadrática $y=x^2$, con etiquetas y título incluidos.

Varios plots (subgráficas en una sola figura)

```
1. plt.subplot(1,2,1)
2. plt.plot(x, y, 'r')
3. plt.subplot(1,2,2)
4. plt.plot(x, y, 'b')
5.
```

Se muestran dos gráficos en la misma figura usando subplot, uno en rojo y otro en azul.

Figuras y ejes personalizados

```
1. figura_1 = plt.figure("Figura 1", figsize=(5,4), dpi=100)
2. eje_1 = figura_1.add_axes([0.1, 0.1, 0.9, 0.9])
3. eje_1.plot(x, y, label='x / x^2')
4. eje_1.plot(y, x, label='x^2 / x')
5. eje_1.legend(loc=0)
6.
```

Se define una figura manualmente con dos sistemas de ejes (add_axes). Se muestran dos curvas con leyenda.

Subplots organizados automáticamente

```
1. fig_1, eje_1 = plt.subplots(figsize=(8,4), nrows=1, ncols=3)
2. plt.tight_layout()
3.
```

Crea una figura con 3 gráficos en una fila, distribuidos automáticamente.

Apariencia del gráfico

```
1. ejes.plot(x, y, color='navy', alpha=0.75, lw=2, ls='_.', marker='o', markersize=7,
markerfacecolor='y')
2.
```

Se personalizan estilo de línea, color, grosor, tipo de marcador, transparencia y límites del gráfico.

Guardar una figura

```
1. figura.savefig("figura_ejemplo.png")
2.
```

Guarda la figura generada como imagen PNG.

Pandas + Matplotlib

```
1. dataframe_helados = pd.read_csv('icecreamsales.csv')
2. plt.plot(x, y)
3. plt.bar(x, y)
4.
```

Se carga un CSV, se organizan los datos y se representan en línea y barras. También se usa `annotate` y `text` para anotar el gráfico.

Histogramas

```
1. plt.hist(lista_3, bins=11, density=True, stacked=True)
2.
```

Se representa un histograma con la suma de dos listas de datos aleatorios. Muestra la distribución de frecuencias.

Gráficos de barras

```
1. plt.bar(x, y, color='navy', yerr=var)
2.
```

Se representan barras verticales con errores (`yerr`). También se construye un gráfico de barras apiladas (hombres y mujeres).

Gráficos de pastel

```
1. plt.pie(personas, explode=sobresaltado, labels=tipos, colors=colores, shadow=True,
startangle=140)
2.
```

Gráfico de pastel con resaltado (`explode`) en el primer sector, colores aleatorios y leyenda añadida.

Series temporales

```
1. plt.plot(array_fechas, goog_five)
2.
```

Se carga un CSV con datos de Google, se convierten a fechas laborales (`bdate_range`) y se grafica la evolución temporal del precio.

Tablas

```
1. tabla = plt.table(cellText=datos_mostrar.to_numpy(), loc='center',
collabels=encabezados)
2.
```

Se muestra una tabla visual en una figura con los últimos 10 datos del DataFrame de Google, formateados con 2 decimales.

Diagramas de dispersión

```
1. plt.scatter(pos_x, pos_y, area, c=colores, alpha=0.5)
2.
```

Se crea un gráfico de dispersión (scatter plot) con tamaño y color aleatorio, ideal para visualizar la distribución y densidad de puntos.

Mapas de calor

```
1. mapa = ejes.imshow(datos_aleatorios, cmap='hot')  
2.
```

Se construye un mapa de calor con valores aleatorios y etiquetas personalizadas. Se añaden los valores numéricos directamente sobre el gráfico.