

## matplotlib

### Cambio de aspecto de los gráficos

Los gráficos creados con Matplotlib son personalizables y puede cambiarse el aspecto de casi todos sus elementos. Los elementos que suelen modificarse más a menudo son:

- Colores
- Marcadores de puntos
- Estilo de líneas
- Títulos
- Ejes
- Leyenda
- Rejilla

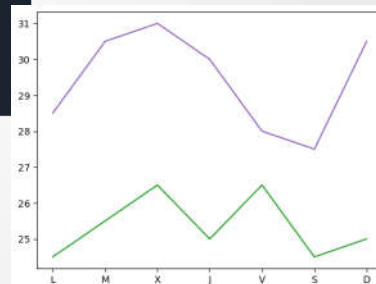
Mgr. Jose Luis Vera

## matplotlib

### Cambio de aspecto de los gráficos (colores)

Para cambiar el color de los objetos se utiliza el parámetro `color = nombre-color`, donde `nombre-color` es una cadena con el nombre del color de entre los [colores disponibles](#).

```
1 import matplotlib.pyplot as plt
2 fig, ax = plt.subplots()
3 dias = ['L', 'M', 'X', 'J', 'V', 'S', 'D']
4 temperaturas = {'Cochabamba': [28.5, 30.5, 31, 30, 28, 27.5, 30.5],
5                 'Santa Cruz': [24.5, 25.5, 26.5, 25, 26.5, 24.5, 25]}
6 ax.plot(dias, temperaturas['Cochabamba'], color = 'tab:purple')
7 ax.plot(dias, temperaturas['Santa Cruz'], color = 'tab:green')
8 plt.show()
```



Mgr. Jose Luis Vera

## matplotlib

### Cambio de aspecto de los gráficos (marcadores)

Para cambiar la forma de los puntos marcadores se utiliza el parámetro **marker = nombre-marcador** donde **nombre-marcador** es una cadena con el nombre del marcador de entre los [marcadores disponibles](#)

```
1 import matplotlib.pyplot as plt
2 fig, ax = plt.subplots()
3 dias = ['L', 'M', 'X', 'J', 'V', 'S', 'D']
4 temperaturas = {'Cochabamba':[28.5, 30.5, 31, 30, 28, 27.5, 30.5],
5                 'Santa Cruz':[24.5, 25.5, 26.5, 25, 26.5, 24.5, 25]}
6 # cambio de color
7 #ax.plot(dias, temperaturas['Cochabamba'], color = 'tab:purple')
8 #ax.plot(dias, temperaturas['Santa Cruz'], color = 'tab:green')
9 #cambio de marcadores
10 #ax.plot(dias, temperaturas['Cochabamba'], marker='^')
11 #ax.plot(dias, temperaturas['Santa Cruz'], marker='o')
12
13 #combinación de marcadores y colores
14 ax.plot(dias, temperaturas['Cochabamba'], color = 'tab:purple', marker='^')
15 ax.plot(dias, temperaturas['Santa Cruz'], color = 'tab:green', marker='o')
16 plt.show()
```

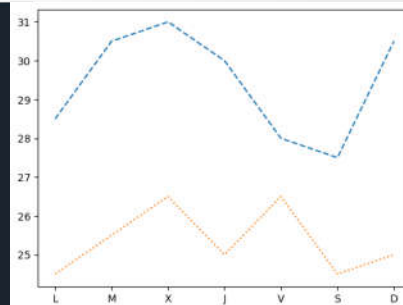
Mgr. Luis Vera

## matplotlib

### Cambio de aspecto de los gráficos (líneas)

Para cambiar el estilo de las líneas se utiliza el parámetro **linestyle = nombre-estilo** donde **nombre-estilo** es una cadena con el nombre del estilo de entre los [estilos disponibles](#)

```
1 import matplotlib.pyplot as plt
2 fig, ax = plt.subplots()
3 dias = ['L', 'M', 'X', 'J', 'V', 'S', 'D']
4 temperaturas = {'Cochabamba':[28.5, 30.5, 31, 30, 28, 27.5, 30.5],
5                 'Santa Cruz':[24.5, 25.5, 26.5, 25, 26.5, 24.5, 25]}
6
7 # cambia el estilo de líneas
8 ax.plot(dias, temperaturas['Cochabamba'], linestyle= 'dashed')
9 ax.plot(dias, temperaturas['Santa Cruz'], linestyle= 'dotted')
10
11 #combinación de marcadores y colores
12 """
13 ax.plot(dias, temperaturas['Cochabamba'],
14         color = 'tab:purple', marker='^',linestyle= 'dashed')
15 ax.plot(dias, temperaturas['Santa Cruz'],
16         color = 'tab:green', marker='o', linestyle= 'dotted')
17 """
18 plt.show()
```



Mgr. Jose Luis Vera

## matplotlib

### Cambio de aspecto de los gráficos

Para añadir un título principal al gráfico se utiliza el siguiente método:

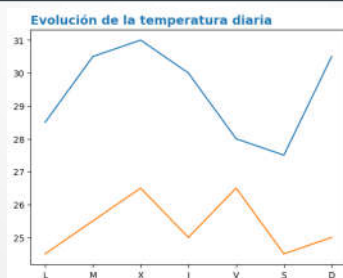
`ax.set_title(titulo, loc=alineacion, fontdict=fuente)`: Añade un título con el contenido de la cadena `titulo` a los ejes `ax`. El parámetro `loc` indica la alineación del título, que puede ser `'left'` (izquierda), `'center'` (centro) o `'right'` (derecha), y el parámetro `fontdict` indica mediante un diccionario las características de la fuente (la el tamaño `fontsize`, el grosor `fontweight` o el color `color`)

Mgr. Jose Luis Vera

## matplotlib

### Cambio de aspecto de los gráficos (Títulos)

```
1 import matplotlib.pyplot as plt
2 fig, ax = plt.subplots()
3 dias = ['L', 'M', 'X', 'J', 'V', 'S', 'D']
4 temperaturas = {'Cochabamba':[28.5, 30.5, 31, 30, 28, 27.5, 30.5],
5                 'Santa Cruz':[24.5, 25.5, 26.5, 25, 26.5, 24.5, 25]}
6 ax.plot(dias, temperaturas['Cochabamba'])
7 ax.plot(dias, temperaturas['Santa Cruz'])
8 ax.set_title('Evolución de la temperatura diaria',
9             loc = "left",
10             fontdict = {'fontsize':14, 'fontweight':'bold', 'color':'tab:blue'})
11 plt.show()
```



Mgr. Jose Luis Vera

## matplotlib

### Cambio de aspecto de los gráficos, manejo de ejes ,I

- `ax.set_xlabel(titulo)` : Añade un título con el contenido de la cadena `titulo` al eje x de `ax`. Se puede personalizar la alineación y la fuente con los mismos parámetros que para el título principal.
- `ax.set_ylabel(titulo)` : Añade un título con el contenido de la cadena `titulo` al eje y de `ax`. Se puede personalizar la alineación y la fuente con los mismos parámetros que para el título principal.
- `ax.set_xlim([limite-inferior, limite-superior])` : Establece los límites que se muestran en el eje x de `ax`.
- `ax.set_ylim([limite-inferior, limite-superior])` : Establece los límites que se muestran en el eje y de `ax`.

Mgr. Jose Luis Vera

## matplotlib

### Cambio de aspecto de los gráficos, manejo de ejes, II

- `ax.set_xticks(marcas)` : Dibuja marcas en el eje x de `ax` en las posiciones indicadas en la lista `marcas`.
- `ax.set_yticks(marcas)` : Dibuja marcas en el eje y de `ax` en las posiciones indicadas en la lista `marcas`.
- `ax.set_xscale(escala)` : Establece la escala del eje x de `ax`, donde el parámetro `escala` puede ser `'linear'` (lineal) o `'log'` (logarítmica).
- `ax.set_yscale(escala)` : Establece la escala del eje y de `ax`, donde el parámetro `escala` puede ser `'linear'` (lineal) o `'log'` (logarítmica).

Mgr. Jose Luis Vera

## matplotlib

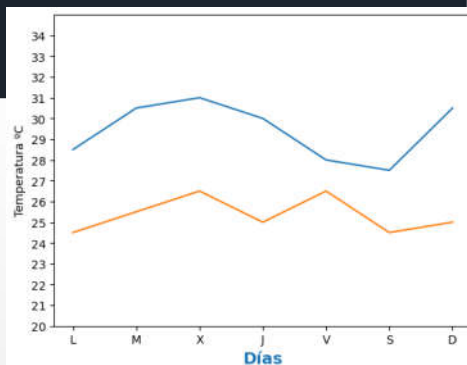
## Cambio de aspecto de los gráficos, manejo de ejes, III

Modo

```

1 import matplotlib.pyplot as plt
2 fig, ax = plt.subplots()
3 dias = ['L', 'M', 'X', 'J', 'V', 'S', 'D']
4 temperaturas = {'Cochabamba':[28.5, 30.5, 31, 30, 28, 27.5, 30.5],
5                 'Santa Cruz':[24.5, 25.5, 26.5, 25, 26.5, 24.5, 25]}
6 ax.plot(dias, temperaturas['Cochabamba'])
7 ax.plot(dias, temperaturas['Santa Cruz'])
8 ax.set_xlabel("Días", fontdict = {'fontsize':14, 'fontweight':'bold', 'color':'tab:blue'})
9 ax.set_ylabel("Temperatura °C")
10 ax.set_ylim([20,35])
11 ax.set_yticks(range(20, 35))
12 plt.show()
13

```



Mgr. Jose Luis Vera

## matplotlib

## Cambio de aspecto de los gráficos, leyenda

Modalidad Virtual

Para añadir una leyenda a un gráfico se utiliza el siguiente método:

`ax.legend(leyendas, loc = posición)`

Dibuja un leyenda en los ejes **ax** con los nombres indicados en la lista **leyendas**. El parámetro **loc** indica la posición en la que se dibuja la leyenda y puede ser 'upper left' (arriba izquierda), 'upper center' (arriba centro), 'upper right' (arriba derecha), 'center left' (centro izquierda), 'center' (centro), 'center right' (centro derecha), 'lower left' (abajo izquierda), 'lower center' (abajo centro), 'lower right' (abajo derecha).

Se puede omitir la lista **leyendas** si se indica la leyenda de cada serie en la función que la dibuja mediante el parámetro **label**

Mgr. Jose Luis Vera

## matplotlib

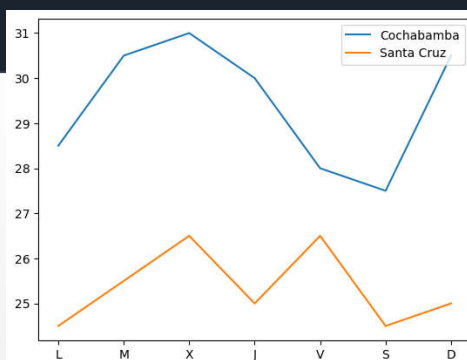
### Cambio de aspecto de los gráficos, leyenda

Modalidad Virtual

```

1 import matplotlib.pyplot as plt
2 fig, ax = plt.subplots()
3 dias = ['L', 'M', 'X', 'J', 'V', 'S', 'D']
4 temperaturas = {'Cochabamba': [28.5, 30.5, 31, 30, 28, 27.5, 30.5],
5                 'Santa Cruz': [24.5, 25.5, 26.5, 25, 26.5, 24.5, 25]}
6 ax.plot(dias, temperaturas['Cochabamba'], label='Cochabamba')
7 ax.plot(dias, temperaturas['Santa Cruz'], label='Santa Cruz')
8
9 ax.legend(loc = 'upper right')
10 plt.show()

```



Mgr. Jose Luis Vera

## matplotlib

### Cambio de aspecto de los gráficos, rejilla

Modalidad Virtual

```
ax.grid(axis=ejes, color=color, linestyle=estilo)
```

Dibuja una rejilla en los ejes de **ax**.

El parámetro **axis** indica los ejes sobre los que se dibuja la rejilla y puede ser **'x'** (eje x), **'y'** (eje y) o **'both'** (ambos).

Los parámetros **color** y **linestyle** establecen el color y el estilo de las líneas de la rejilla, y pueden tomar los mismos valores vistos en los apartados de colores y líneas

Mgr. Jose Luis Vera

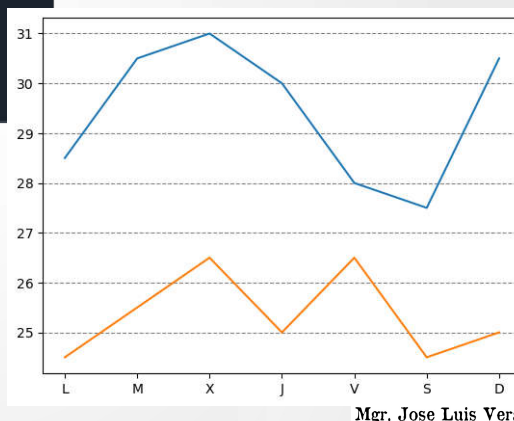
## matplotlib

### Cambio de aspecto de los gráficos, rejilla

```

1 import matplotlib.pyplot as plt
2 fig, ax = plt.subplots()
3 dias = ['L', 'M', 'X', 'J', 'V', 'S', 'D']
4 temperaturas = {'Cochabamba': [28.5, 30.5, 31, 30, 28, 27.5, 30.5],
5                 'Santa Cruz': [24.5, 25.5, 26.5, 25, 26.5, 24.5, 25]}
6 ax.plot(dias, temperaturas['Cochabamba'], label='Cochabamba')
7 ax.plot(dias, temperaturas['Santa Cruz'], label='Santa Cruz')
8
9 ax.grid(axis='y', color='gray', linestyle='dashed')
10 plt.show()
11

```



## matplotlib

### Cambio de aspecto de los gráficos , múltiples gráficos

Es posible dibujar varios gráficos en distintos ejes en una misma figura organizados en forma de tabla.

Para ello, cuando se inicializa la figura y los ejes, hay que pasarle a la función **subplots** el número de filas y columnas de la tabla que contendrá los gráficos.

Con esto los distintos ejes se organizan en un array y se puede acceder a cada uno de ellos a través de sus índices.

Si se quiere que los distintos ejes compartan los mismos límites para los ejes se pueden pasar los parámetros **sharex = True** para el eje x o **sharey = True** para el eje y.

Mgr. Jose Luis Vera

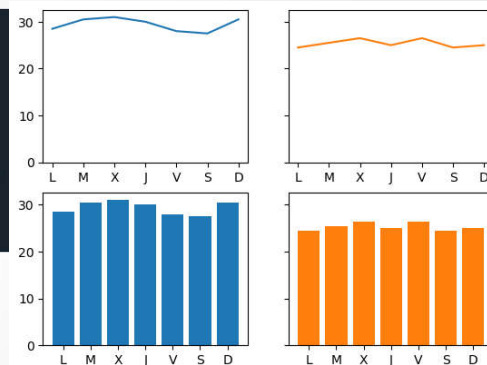
## matplotlib

### Cambio de aspecto de los gráficos, múltiples gráficos

```

1 # multiples gráficos
2 import matplotlib.pyplot as plt
3 fig, ax = plt.subplots(2,2,sharey=True)
4 dias = ['L', 'M', 'X', 'J', 'V', 'S', 'D']
5 temperaturas = {'Cochabamba':[28.5, 30.5, 31, 30, 28, 27.5, 30.5],
6                 'Santa Cruz':[24.5, 25.5, 26.5, 25, 26.5, 24.5, 25]}
7 ax[0, 0].plot(dias, temperaturas['Cochabamba'])
8 ax[0, 1].plot(dias, temperaturas['Santa Cruz'], color = 'tab:orange')
9 ax[1, 0].bar(dias, temperaturas['Cochabamba'])
10 ax[1, 1].bar(dias, temperaturas['Santa Cruz'], color = 'tab:orange')
11 plt.show()

```



Mgr. Jose Luis Vera

## matplotlib

### Cambio de aspecto de los gráficos, 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'** (lineas), **'scatter'** (puntos), **'bar'** (barras verticales), **'barh'** (barras horizontales), **'hist'** (histograma), **'box'** (cajas), **'density'** (densidad), **'area'** (area) o **'pie'** (sectores).

Es posible pasar otros parámetros para indicar el color, el marcador o el estilo de línea

Mgr. Jose Luis Vera

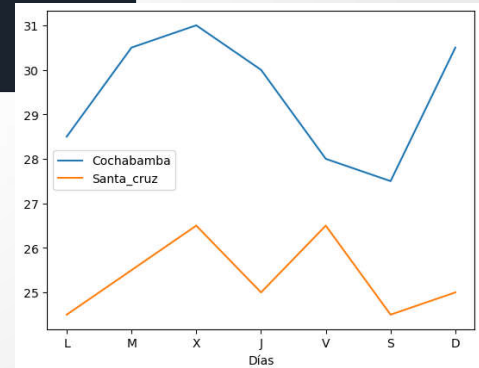
## matplotlib

### Cambio de aspecto de los gráficos, integración con pandas

```

3 import pandas as pd
4 import matplotlib.pyplot as plt
5 df = pd.DataFrame({'Días': ['L', 'M', 'X', 'J', 'V', 'S', 'D'],
6                   'Cochabamba': [28.5, 30.5, 31, 30, 28, 27.5, 30.5],
7                   'Santa_cruz': [24.5, 25.5, 26.5, 25, 26.5, 24.5, 25]})
8 fig, ax = plt.subplots()
9 df.plot(x = 'Días', y = 'Cochabamba', ax = ax)
10 df.plot(x = 'Días', y = 'Santa_cruz', ax = ax)
11 plt.show()

```



Mgr. Jose Luis Vera

## Tipos de datos/Métodos

# FIN

Mgr. Jose Luis Vera