

Librería Matplotlib

Dra. Yessenia Yari Ramos



Universidad Católica
San Pablo

Departamento de Ciencia
de la Computación

Contenido



Matplotlib



Tipos de Gráficos



Gráfico de Línea



Gráfico de Dispersión



Gráfico de Barras



Histogramas



Subplots

Qué es Matplotlib?



Qué es Matplotlib?

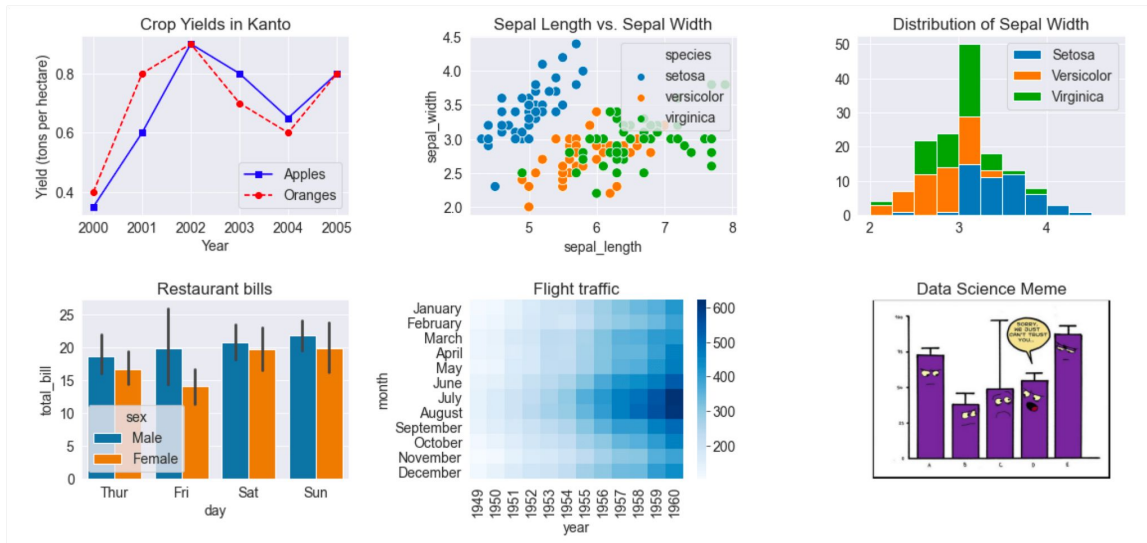
- Matplotlib es una biblioteca de ploteo para el lenguaje de programación Python y su extensión matemática numérica NumPy.
- Proporciona una API orientada a objetos para incrustar gráficos en aplicaciones que utilizan kits de herramientas de GUI de uso general como Tkinter, wxPython, Qt o GTK +.



Qué es Matplotlib?

❖ Los tipos de gráficos más conocidos son:

- Gráfico de Línea
- Gráfico de Dispersión
- Gráfico de Barras
- Histogramas



Qué es Matplotlib?

- Para crear gráficos debemos importar la biblioteca matplotlib

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

Gráficos

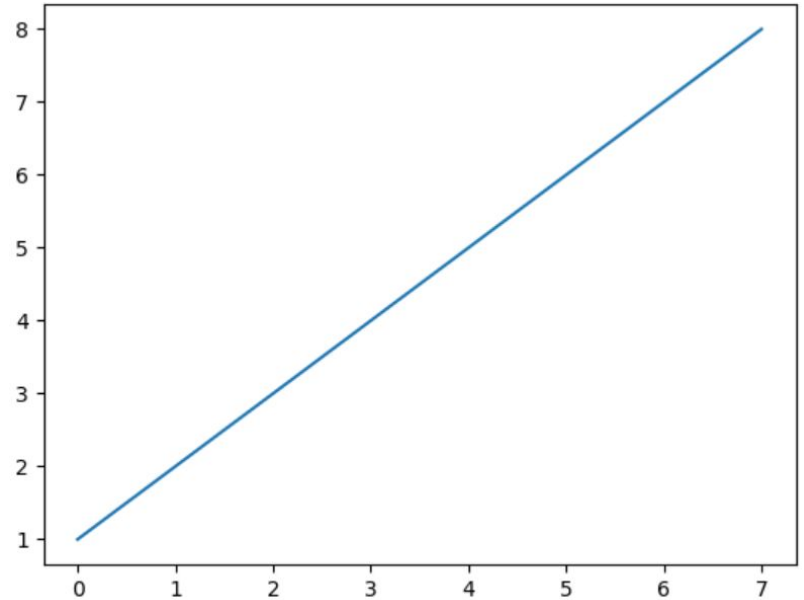


Gráfico de Líneas

- También conocidos como Line Plots
- Un gráfico de líneas o gráfico de líneas o gráfico de líneas o gráfico de curvas es un tipo de gráfico que muestra información como una serie de puntos de datos llamados 'marcadores' conectados por segmentos de línea recta.

Gráficos de Líneas

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 lista_num = np.array([1,2,3,4,5,6,7,8])
5 plt.plot(lista_num)
```

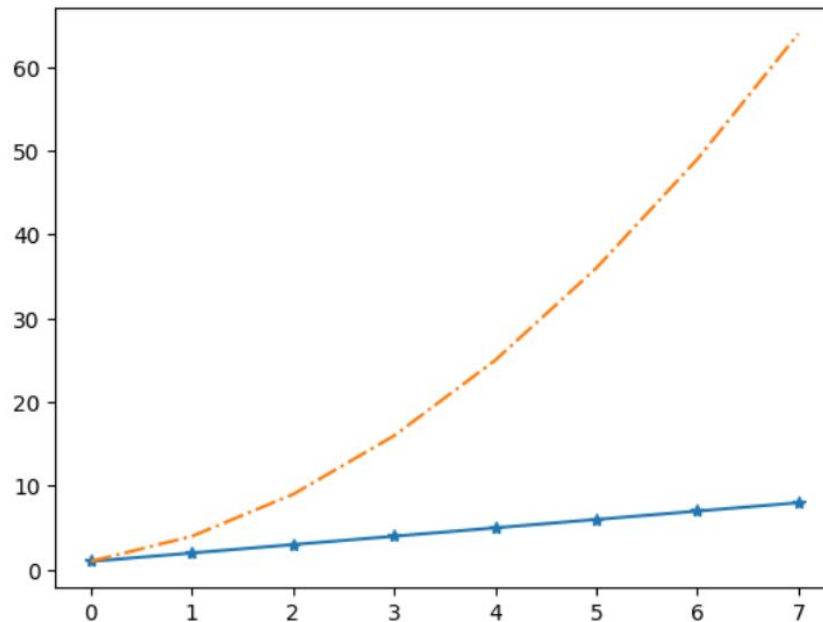


Gráficos de Líneas

- Formato de la línea
- Dos o mas gráficas

```
1 lista_num = np.array([1,2,3,4,5,6,7,8])
2 cuadrados = lista_num **2
3 plt.figure()
4 plt.plot(lista_num, '-*', cuadrados, '-.')
```

```
<matplotlib.lines.Line2D at 0x7d72e4119c00>,  
<matplotlib.lines.Line2D at 0x7d72e4119cc0>]
```

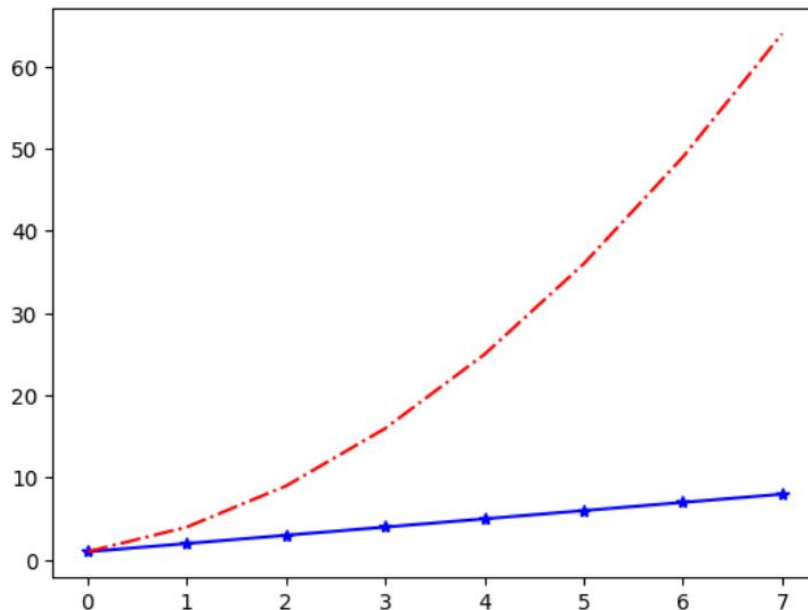


Gráficos de Líneas

- Formato de la línea
- Dos o mas gráficas
- Se puede cambiar de color

```
1 lista_num = np.array([1,2,3,4,5,6,7,8])
2 cuadrados = lista_num **2
3 plt.figure()
4 plt.plot(lista_num, '-*b', cuadrados, '-.r')
5
```

```
<matplotlib.lines.Line2D at 0x7d72e41b1300>,  
<matplotlib.lines.Line2D at 0x7d72e41b1390>]
```

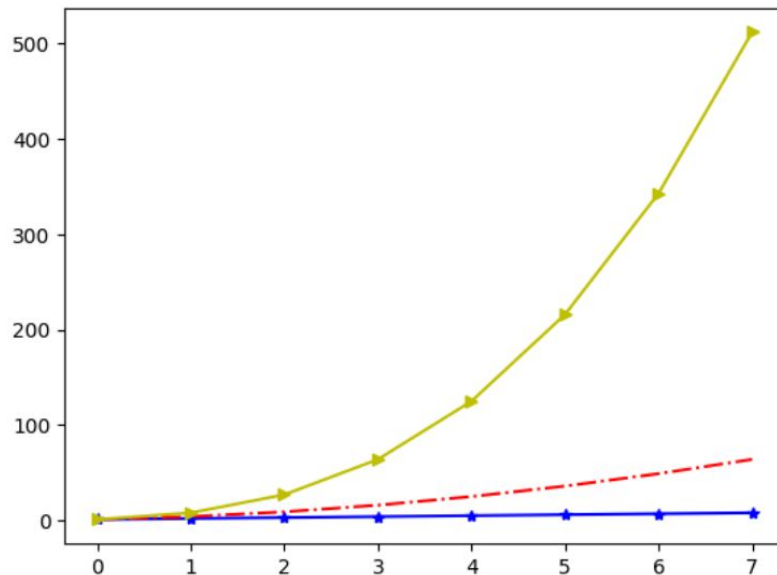


Gráficos de Líneas

- Formato de la línea
- Dos o mas gráficas
- Se puede cambiar de color

```
1 lista_num = np.array([1,2,3,4,5,6,7,8])
2 cuadrados = lista_num **2
3 cubos = lista_num**3
4 plt.figure()
5 plt.plot(lista_num, '-*b', cuadrados, '-.r', cubos,'->y')
```

```
[<matplotlib.lines.Line2D at 0x7d72c1468130>,
<matplotlib.lines.Line2D at 0x7d72c14681c0>,
<matplotlib.lines.Line2D at 0x7d72c1468310>]
```

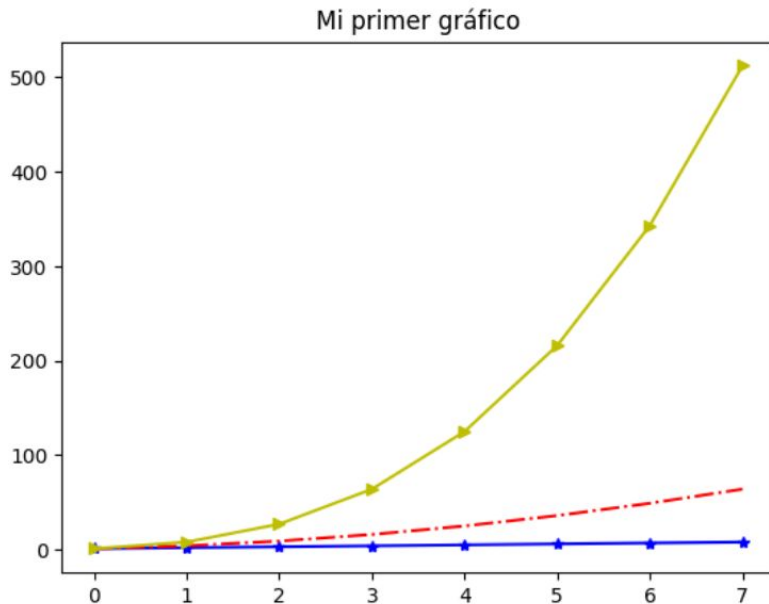


Gráficos de Líneas

- Título al gráfico

```
1 lista_num = np.array([1,2,3,4,5,6,7,8])
2 cuadrados = lista_num **2
3 cubos = lista_num**3
4 plt.figure()
5 plt.plot(lista_num, '-*b', cuadrados, '-.r', cubos,'->y')
6 plt.title("Mi primer gráfico")
```

Text(0.5, 1.0, 'Mi primer gráfico')

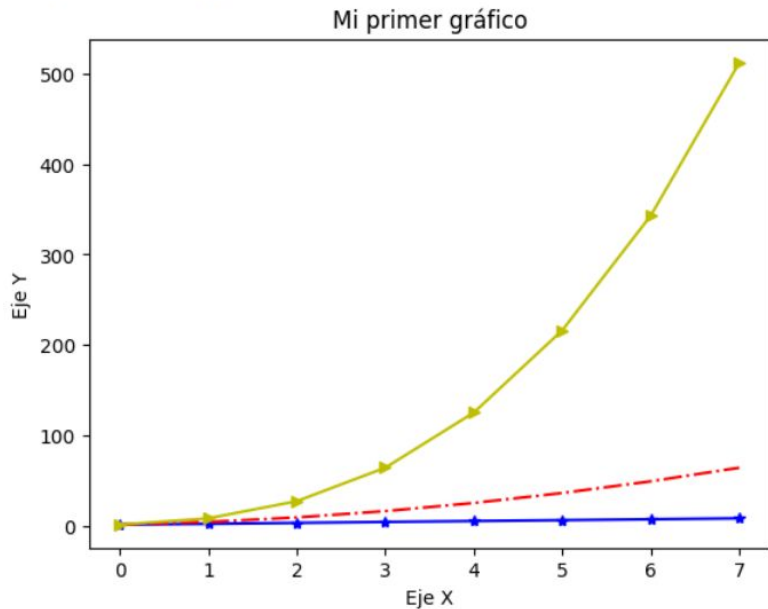


Gráficos de Líneas

- Título al gráfico
- Nombres en los ejes

```
1 lista_num = np.array([1,2,3,4,5,6,7,8])
2 cuadrados = lista_num **2
3 cubos = lista_num**3
4 plt.figure()
5 plt.plot(lista_num, '-*b', cuadrados, '-.r', cubos,'->y')
6 plt.title("Mi primer gráfico")
7 plt.xlabel("Eje X")
8 plt.ylabel("Eje Y")
```

Text(0, 0.5, 'Eje Y')

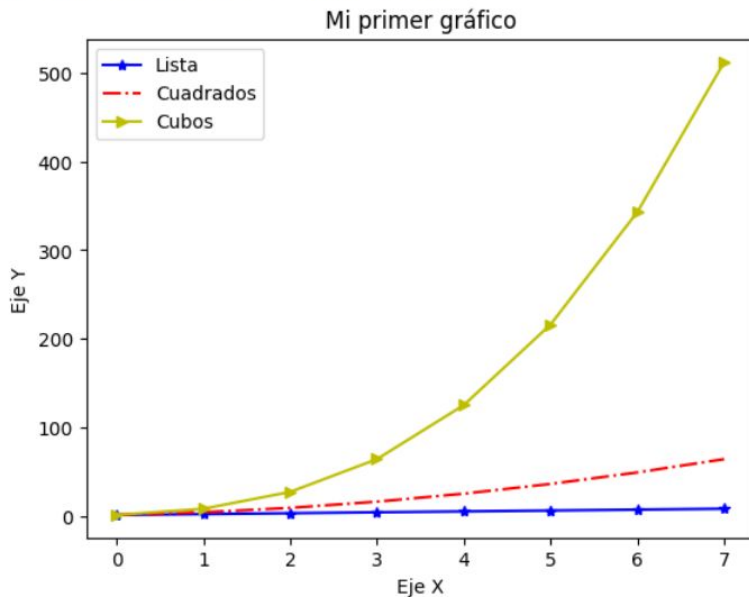


Gráficos de Líneas

- Título al gráfico
- Nombres en los ejes
- Leyenda

```
1 lista_num = np.array([1,2,3,4,5,6,7,8])
2 cuadrados = lista_num **2
3 cubos = lista_num**3
4 plt.figure()
5 plt.plot(lista_num, '-*b', cuadrados, '-.r', cubos,'->y')
6 plt.title("Mi primer gráfico")
7 plt.xlabel("Eje X")
8 plt.ylabel("Eje Y")
9 plt.legend(['Lista', 'Cuadrados', 'Cubos'])
```

<matplotlib.legend.Legend at 0x7d72e42bcf70>



Gráficos de Líneas

- Título al gráfico
- Nombres en los ejes
- Leyenda
- Rotación de rótulos

```
1 lista_num = np.array([1,2,3,4,5,6,7,8])
2 cuadrados = lista_num **2
3 cubos = lista_num**3
4 plt.figure()
5 plt.plot(lista_num, '-*b', cuadrados, '-.r', cubos,'->y')
6 plt.title("Mi primer gráfico")
7 plt.xlabel("Eje X")
8 plt.ylabel("Eje Y")
9 plt.legend(['Lista', 'Cuadrados', 'Cubos'])
10 x = plt.gca().xaxis
11 for item in x.get_ticklabels():
12     item.set_rotation(45)
13
```

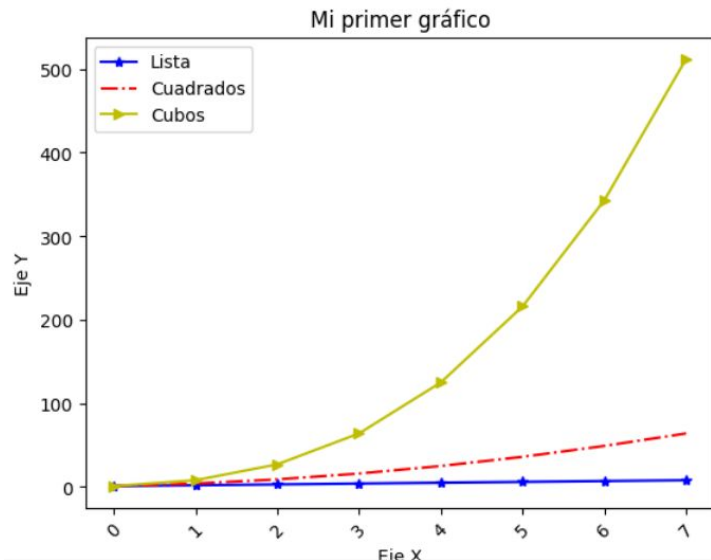


Gráfico de Dispersión

- También conocido como scatter plot
- Utiliza puntos para representar valores para dos variables numéricas diferentes.
- La posición de cada punto en el eje horizontal y vertical indica valores para un punto de datos individual.
- Los gráficos de dispersión se utilizan para observar las relaciones entre las variables.

```
1 x = np.array([1,2,3,4,5])  
2 y = x  
3 plt.figure()  
4 plt.scatter(x, y)  
5
```

<matplotlib.collections.PathCollection at 0x7d72c0c08940>

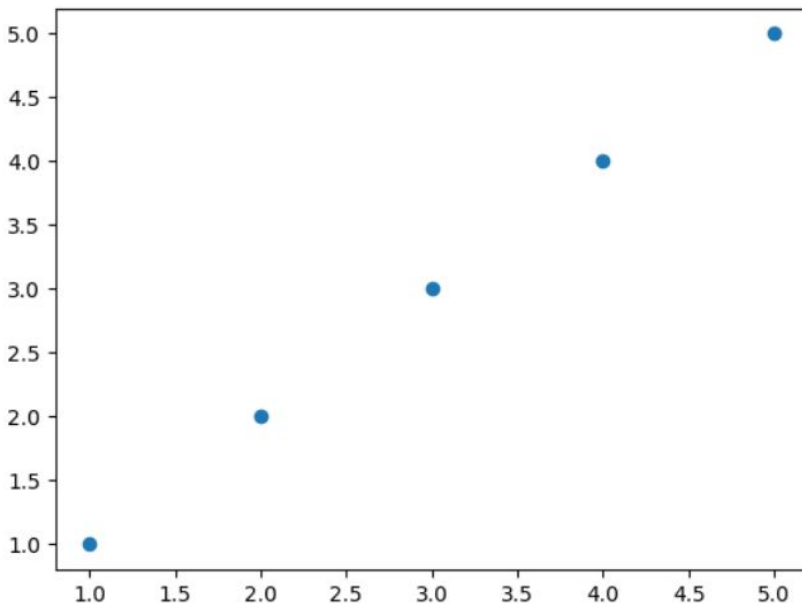
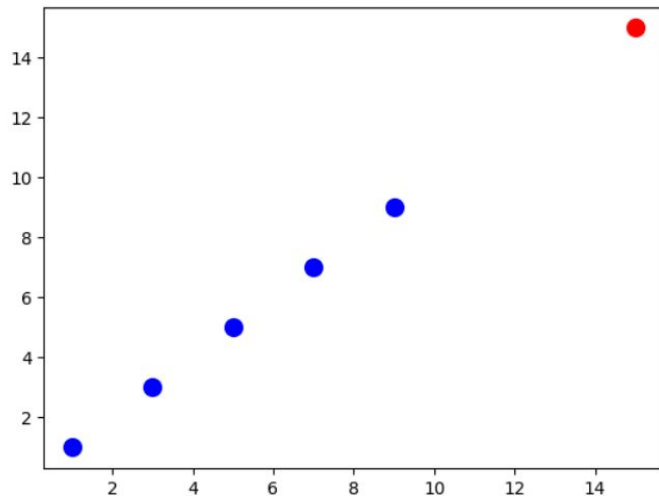


Gráfico de Dispersión

```
1 x = np.array([1,3,5,7,9,15])
2 y = x
3 plt.scatter(x[5], y[5], s=100, c='red', label='Punto Rojo')
4 plt.scatter(x[:5], y[:5], s=100, c='blue', label='Punto Azul')
5
```

<matplotlib.collections.PathCollection at 0x7d72c0a622f0>



```
1 x = np.array([1,3,5,7,9,15])
2 y = x
3 plt.scatter(x[5], y[5], s=100, c='red', label='Punto Rojo')
4 plt.scatter(x[:5], y[:5], s=100, c='blue', label='Punto Azul')
5 plt.title("Gráfico 2")
6 plt.ylabel("Eje Y")
7 plt.xlabel("Eje X")
8 plt.legend()
9
10
```

<matplotlib.legend.Legend at 0x7d72c0b57970>

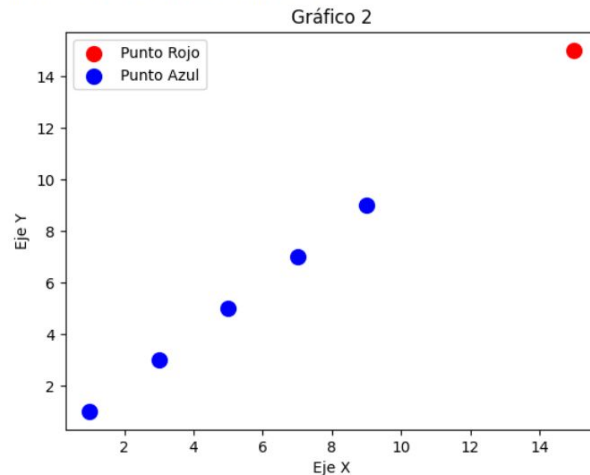
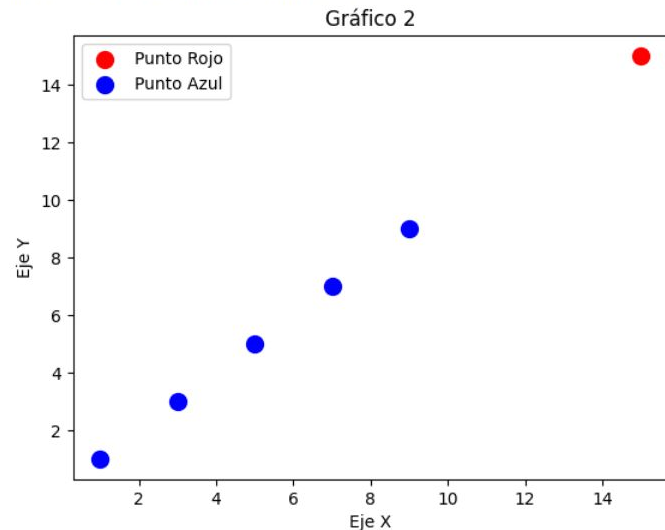


Gráfico de Disper

```
4 plt.scatter(x[:5], y[:5], s=100, c='blue', label='Punto Azul')
5 plt.title("Gráfico 2")
6 plt.ylabel("Eje Y")
7 plt.xlabel("Eje X")
8 plt.legend(loc = 4, frameon=False, title="Leyeda")
9
```

```
1 x = np.array([1,3,5,7,9,15])
2 y = x
3 plt.scatter(x[5], y[5], s=100, c='red', label='Punto Rojo')
4 plt.scatter(x[:5], y[:5], s=100, c='blue', label='Punto Azul')
5 plt.title("Gráfico 2")
6 plt.ylabel("Eje Y")
7 plt.xlabel("Eje X")
8 plt.legend()
9
10
```

<matplotlib.legend.Legend at 0x7d72c0b57970>



<matplotlib.legend.Legend at 0x7d72c090f9a0>

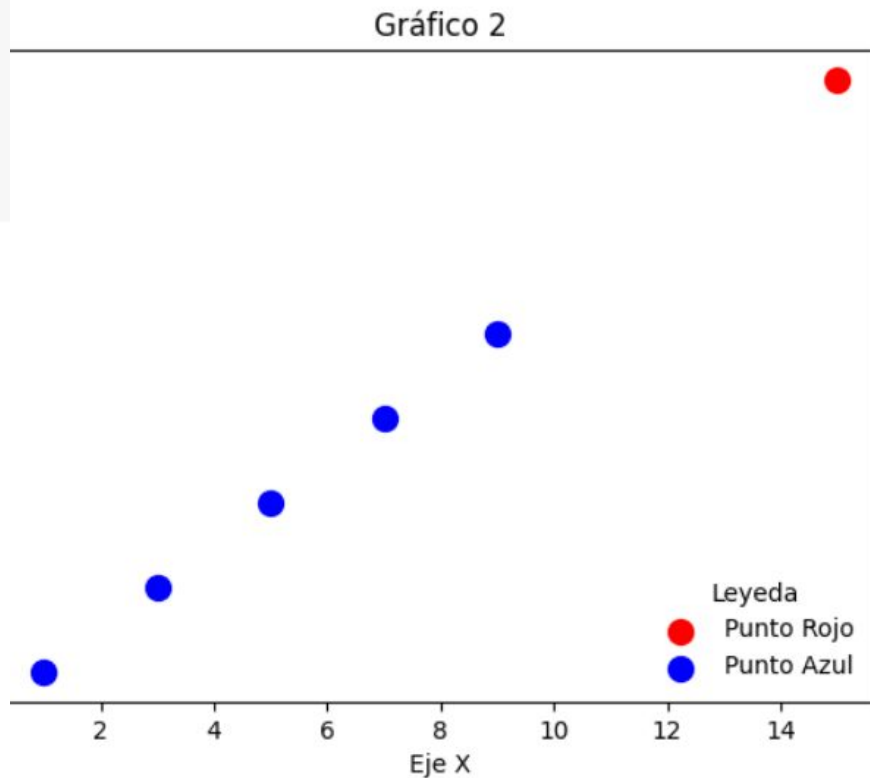


Gráfico de Barras

- También conocido como Bar chart.
- Presenta datos categóricos con barras rectangulares con alturas o longitudes proporcionales a los valores que representan.
- Las barras se pueden trazar vertical u horizontalmente.
- Algunos gráficos de barras presentan barras agrupadas en grupos de más de una, mostrando los valores de más de una variable medida.

```
1 data = np.array([3, 4, 6, 8, 2, 0, 2, 1, 7])
2 plt.figure()
3 xvals = range(len(data))
4 plt.bar(xvals, data)
5
```

<BarContainer object of 9 artists>

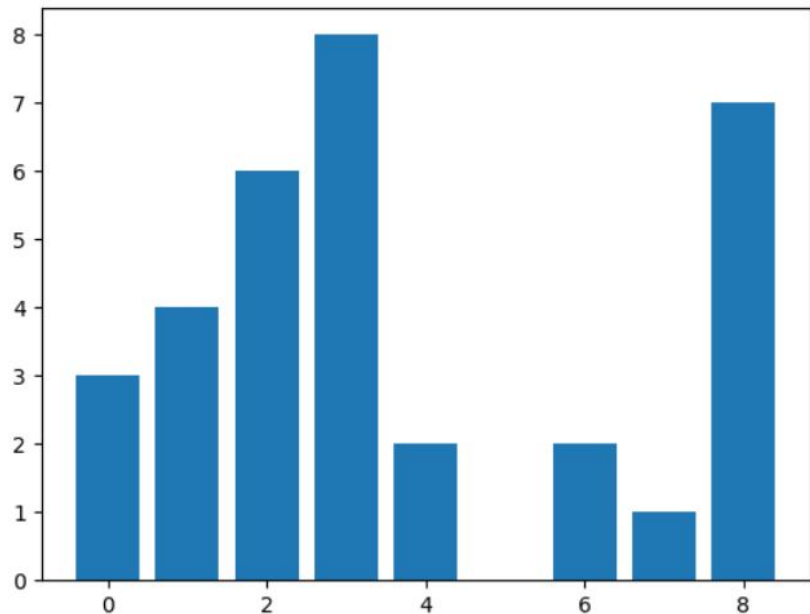
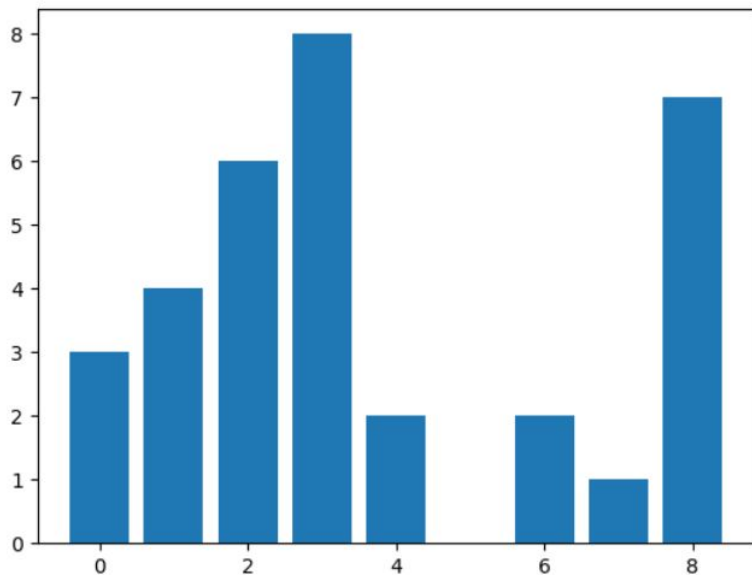


Gráfico de Barras

```
1 data = np.array([3, 4, 6, 8, 2, 0, 2, 1, 7])
2 plt.figure()
3 xvals = range(len(data))
4 plt.bar(xvals, data)
5
```

<BarContainer object of 9 artists>



```
1 data = np.array([3, 4, 6, 8, 2, 0, 2, 1, 7])
2 plt.figure()
3 xvals = range(len(data))
4 #plt.bar(xvals, data)
5 plt.bar(xvals, data, width = 0.5)
```

<BarContainer object of 9 artists>

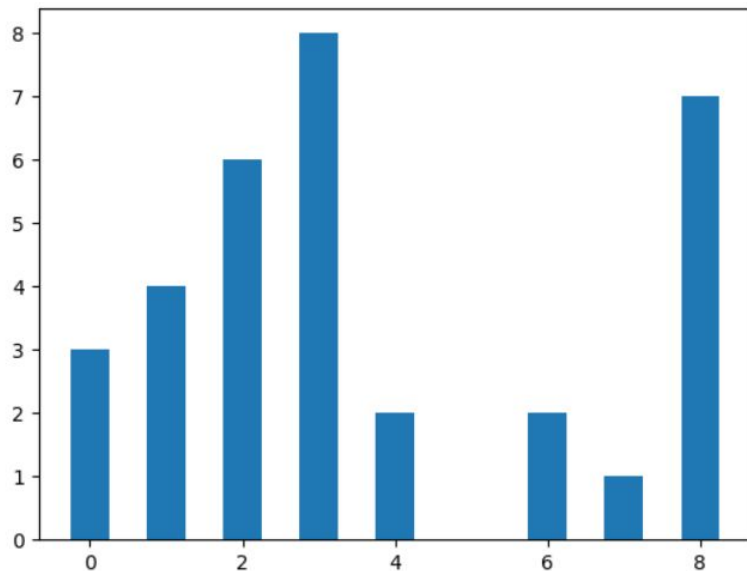
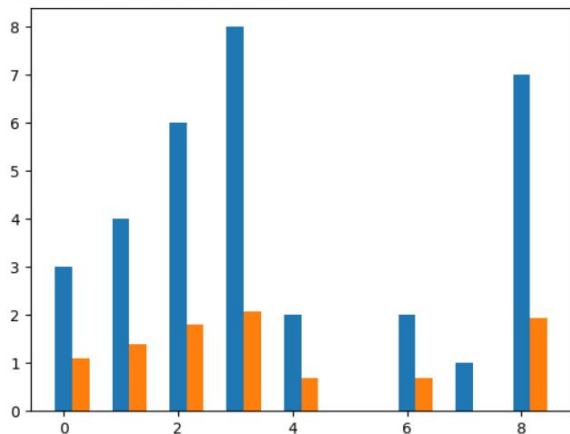


Gráfico de Barras

```
1 data = np.array([3, 4, 6, 8, 2, 0, 2, 1, 7])
2 plt.figure()
3 xvals = range(len(data))
4 # ploteo de data
5 plt.bar(xvals, data, width=0.3)
6 loga = np.log(data)
7 new_xvals = []
8 for item in xvals:
9     new_xvals.append(item+0.3)
10 # ploteo de logaritmo
11 plt.bar(new_xvals, loga, width=0.3)
```

<ipython-input-27-5d7f99fc7afa>:6: RuntimeWarning: divide by zero encountered in log
loga = np.log(data)
<BarContainer object of 9 artists>



```
1 data = np.array([3, 4, 6, 8, 2, 0, 2, 1, 7])
2 plt.figure()
3 xvals = range(len(data))
4 # ploteo de data
5 plt.bar(xvals, data, width=0.3)
6 loga = np.log(data)
7 new_xvals = []
8 for item in xvals:
9     new_xvals.append(item+0.3)
10 # ploteo de logaritmo
11 plt.bar(new_xvals, loga, width=0.3, color='red')
12
```

<ipython-input-28-869e712bcf2b>:6: RuntimeWarning: divide by zero encountered in log
loga = np.log(data)
<BarContainer object of 9 artists>

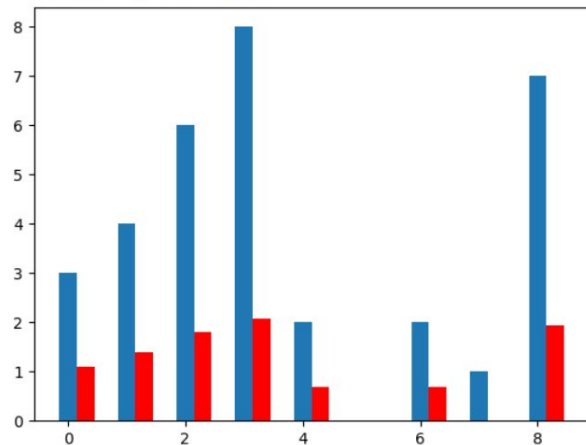


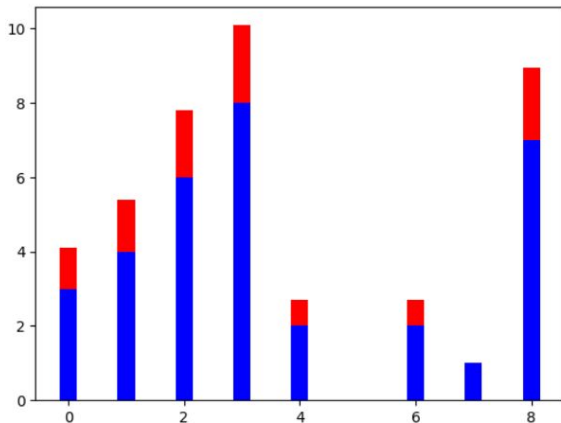
Gráfico de Barras

```
1 data = np.array([3, 4, 6, 8, 2, 0, 2, 1, 7])
2 plt.figure()
3 xvals = range(len(data))
4 # ploteo de data
5 plt.bar(xvals, data, width=0.3)
6 loga = np.log(data)
7 new_xvals = []
8 for item in xvals:
9     new_xvals.append(item+0.3)
10 # ploteo de logaritmo
11 plt.bar(xvals, data, width=0.3, color='b')
12 plt.bar(xvals, loga, width=0.3, color='r', bottom=data)
13
```

<ipython-input-29-19b0f80acce6>:6: RuntimeWarning: divide by zero encountered in log

loga = np.log(data)

<BarContainer object of 9 artists>

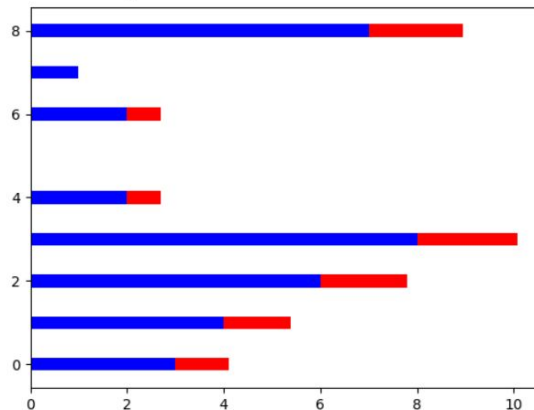


```
1 data = np.array([3, 4, 6, 8, 2, 0, 2, 1, 7])
2 plt.figure()
3 xvals = range(len(data))
4 # ploteo de data
5 plt.bar(xvals, data, width=0.3)
6 loga = np.log(data)
7 new_xvals = []
8 for item in xvals:
9     new_xvals.append(item+0.3)
10 #ploteo de logaritmo
11 #plt.bar(xvals, data, width=0.3, color='b')
12 #plt.bar(xvals, loga, width=0.3, color='r', bottom=data)
13 plt.barh(xvals, data, height=0.3, color='b')
14 plt.barh(xvals, loga, height=0.3, color='r', left=data)
15
```

<ipython-input-7-208848c94c8f>:6: RuntimeWarning: divide by zero encountered in log

loga = np.log(data)

<BarContainer object of 9 artists>

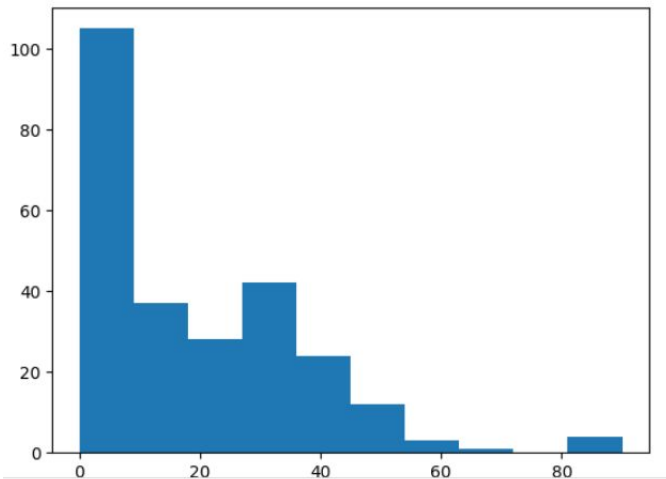


Histogramas

- Un histograma es un gráfico que se utiliza para representar la distribución de frecuencias de algunos puntos de datos de una variable.

```
1 data =imG[0,:]
2 print(data)
3 plt.figure()
4 plt.hist(data)
5
```

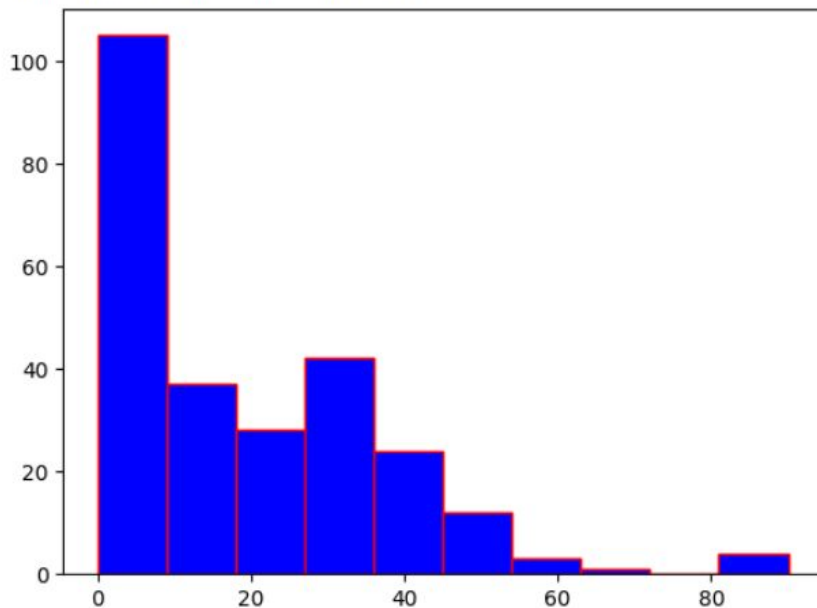
```
22 22 7 22 22 1 29 0 0 22 0 7 7 6 8 0 7 1 45 35 35 52 51 49
52 51 45 22 11 10 34 38 32 46 32 32 20 28 28 44 40 28 40 40 38 40 40 40
44 40 38 40 44 44 34 34 27 27 27 15 5 27 2 3 5 5 3 2 21 12 3 5
2 2 5 3 3 5 12 12 12 21 12 21 21 21 3 12 12 12 12 3 21 21 3 3
21 21 23 23 12 21 21 12 3 14 3 3 2 3 3 3 5 12 21 3 3 12 3 12
12 21 21 23 3 12 5 23 10 3 5 5 12 3 12 5 2 3 5 12 2 3 3 2
3 5 2 15 5 2 3 5 15 33 27 34 37 43 28 24 43 34 3 14 9 13 6 1
1 7 22 0 13 13 6 0 7 6 0 8 7 7 8 7 6 6 8 8 6 7 8 6
7 7 7 8 1 1 0 6 0 50 68 60 58 85 90 90 85 59 52 33 28 24 44 43
41 41 34 27 33 41 15 34 27 5 34 34 27 34 33 27 15 34 2 2 2 15 15 34
27 2 5 2 33 15 2 27 33 34 37 15 35 49 49 2]
array([105., 37., 28., 42., 24., 12., 3., 1., 0., 4.]),
array([ 0., 9., 18., 27., 36., 45., 54., 63., 72., 81., 90.]),
<BarContainer object of 10 artists>
```



Histogramas

```
1 data =img[0,:]
2 #print(data)
3 plt.figure()
4 #plt.hist(data)
5 plt.hist(data, edgecolor='red')
6 plt.hist(data, edgecolor='red', color='blue', bins=10)
```

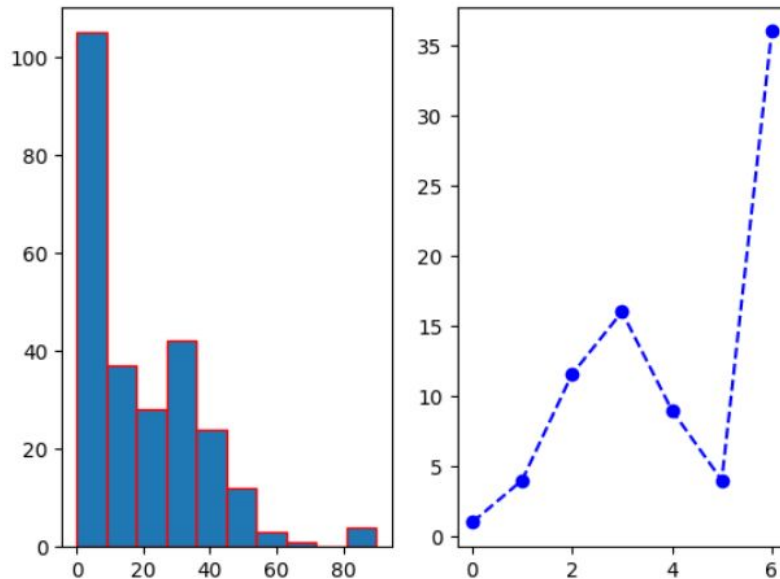
```
(array([105., 37., 28., 42., 24., 12., 3., 1., 0., 4.]),
 array([ 0., 9., 18., 27., 36., 45., 54., 63., 72., 81., 90.]),
 <BarContainer object of 10 artists>)
```



Subplots

```
1 plt.figure()
2 plt.subplot(1,2,1)
3 data1 = imG[0,:]
4 plt.hist(data1, edgecolor='red')
5
6 plt.subplot(1,2,2)
7 datos = np.array([1, 2, 3.4, 4, 3, 2, 6])
8 exp = datos**2
9 plt.plot(exp, '--ob')
10
```

[<matplotlib.lines.Line2D at 0x7bc464be96c0>]





Gracias!