

TEMA 2: Matplotlib

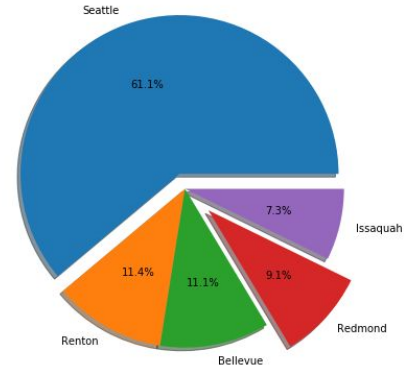
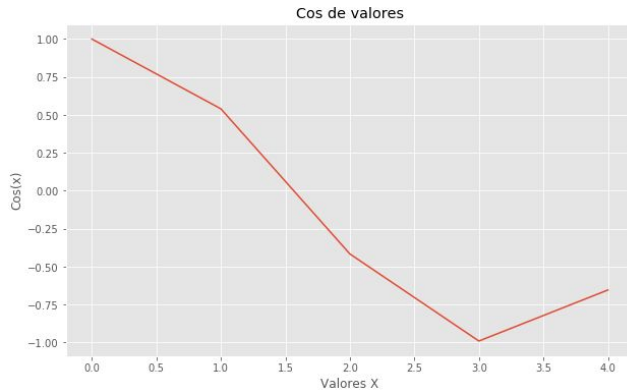
NumPy es una librería que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas de forma eficiente.

Matplotlib

Matplotlib es una biblioteca para la generación de gráficos en dos dimensiones, a partir de datos contenidos en listas o arrays en el lenguaje de programación Python.

<https://matplotlib.org/>

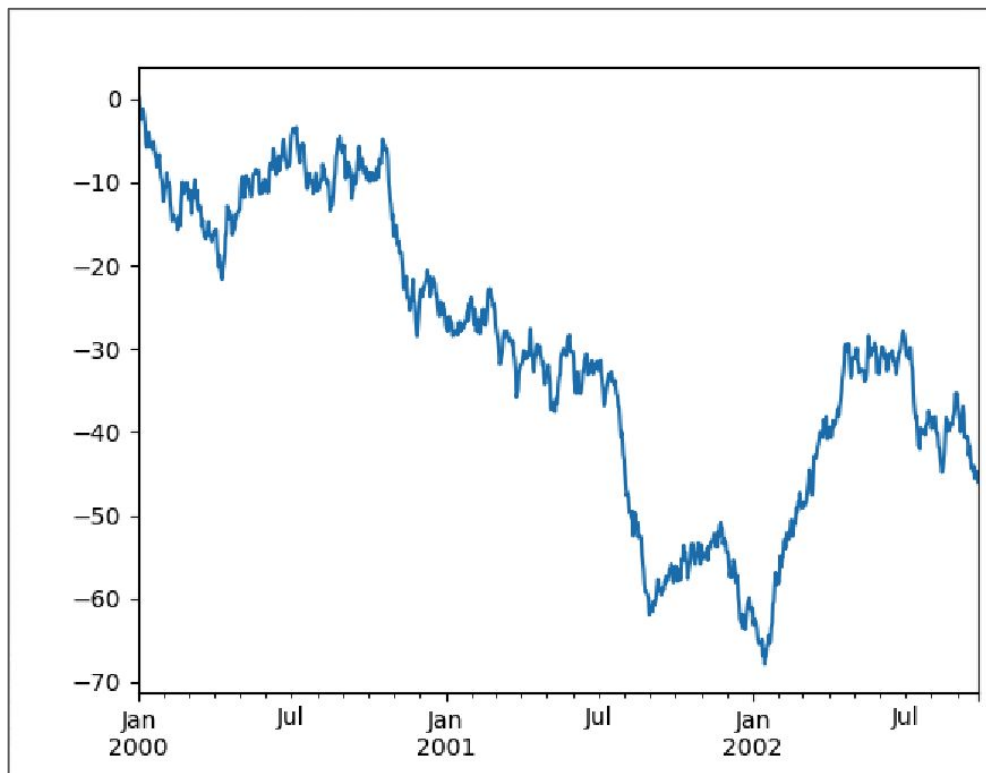
Acceso a un cheat sheet (resumen): <https://matplotlib.org/cheatsheets/>



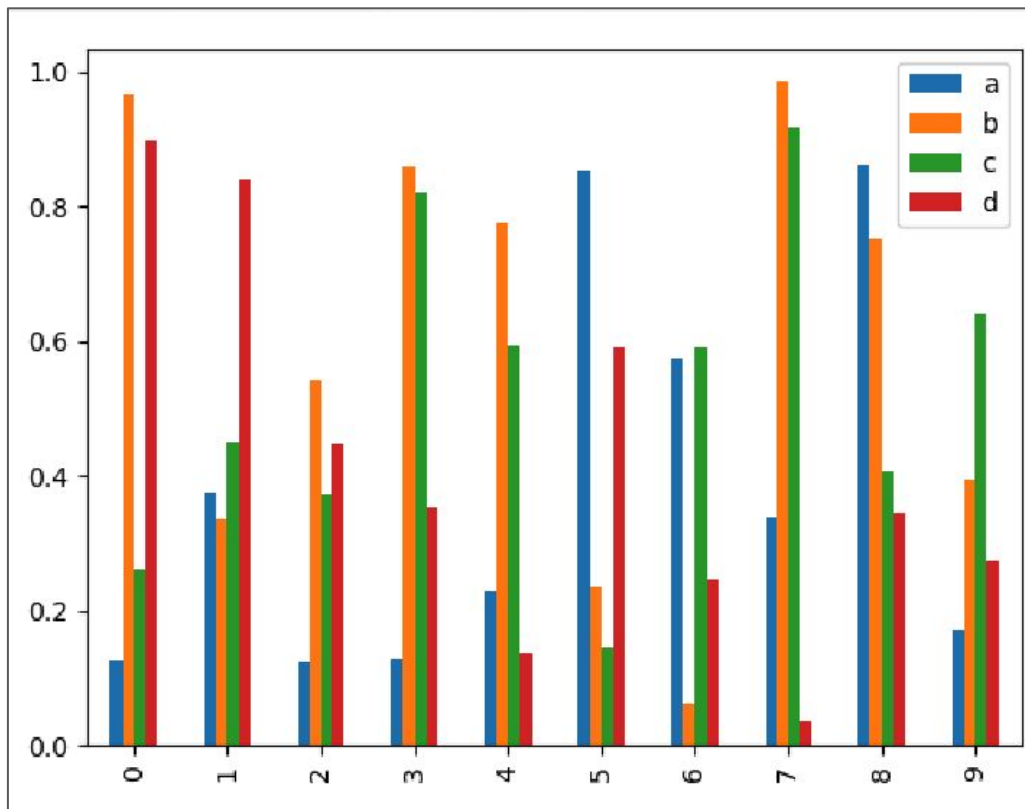
Otras librerías de visualización

- Seaborn: <https://seaborn.pydata.org/>
- Bokeh: <http://bokeh.org/>
- Altair: <https://altair-viz.github.io/>
- Plotly: <https://plotly.com/>

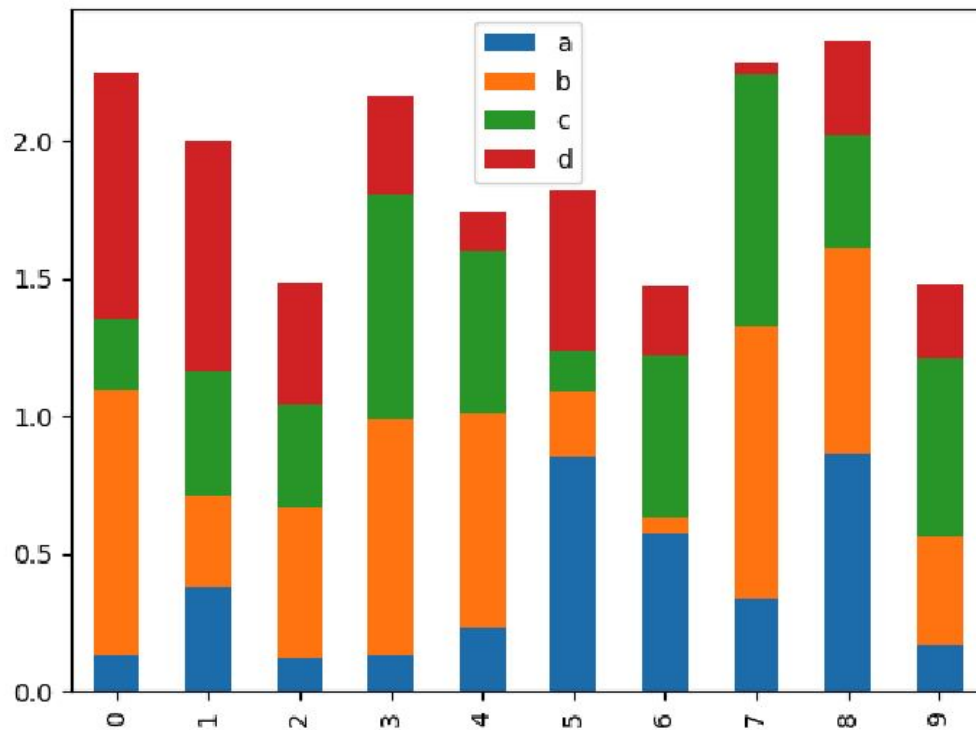
Tipos de gráficos: gráficos de líneas



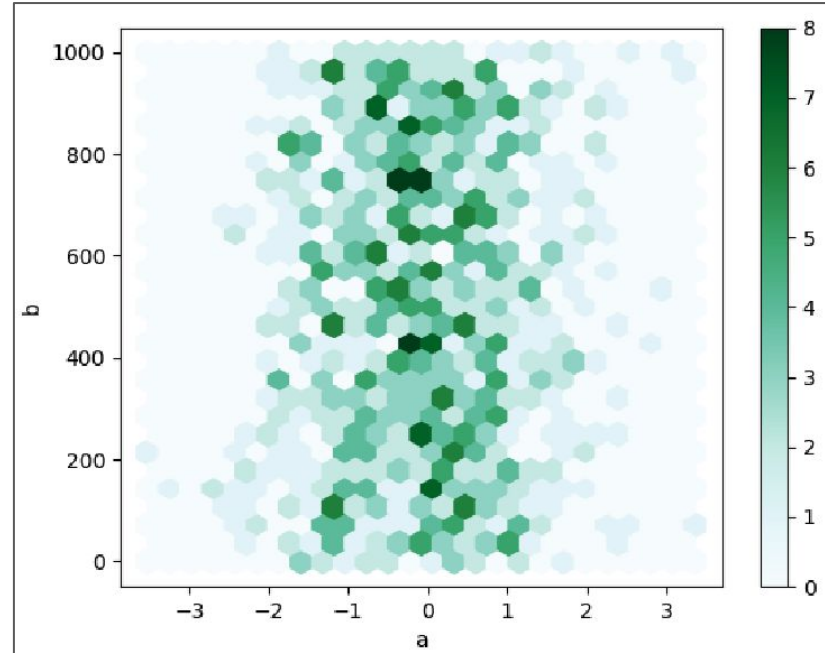
Tipos de gráficos: gráficos de barras



Tipos de gráficos: gráficos de barras apiladas

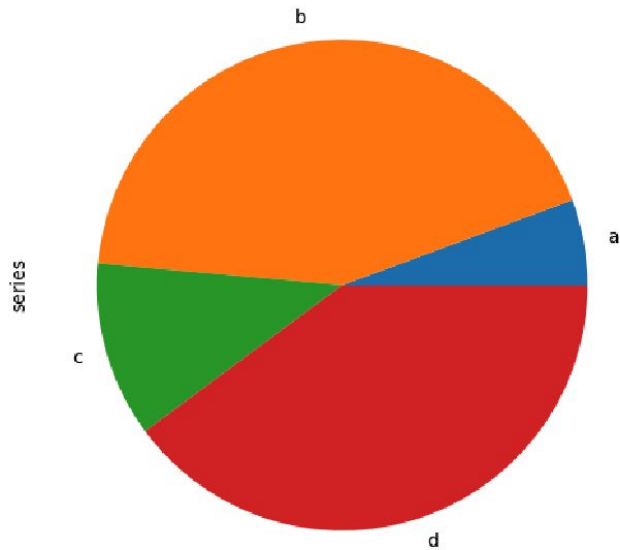


Tipos de gráficos: nube de hexágonos



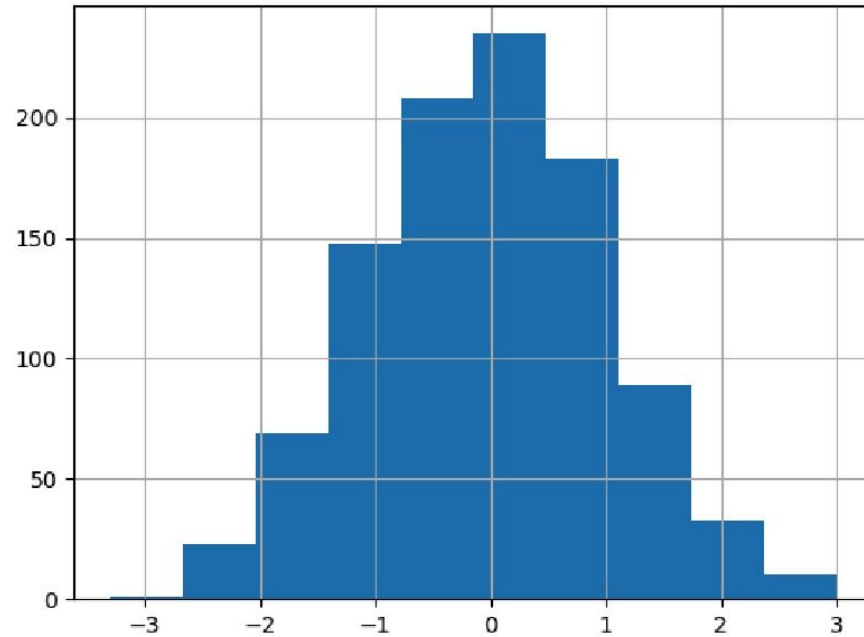
Suelen usarse en lugar de la nube de puntos si tenemos demasiados datos, lo que generaría un gráfico demasiado denso.

Tipos de gráficos: gráfico de tarta



Tipos de gráficos: histograma

La superficie de cada barra es proporcional a la frecuencia de los valores representados.



Matplotlib

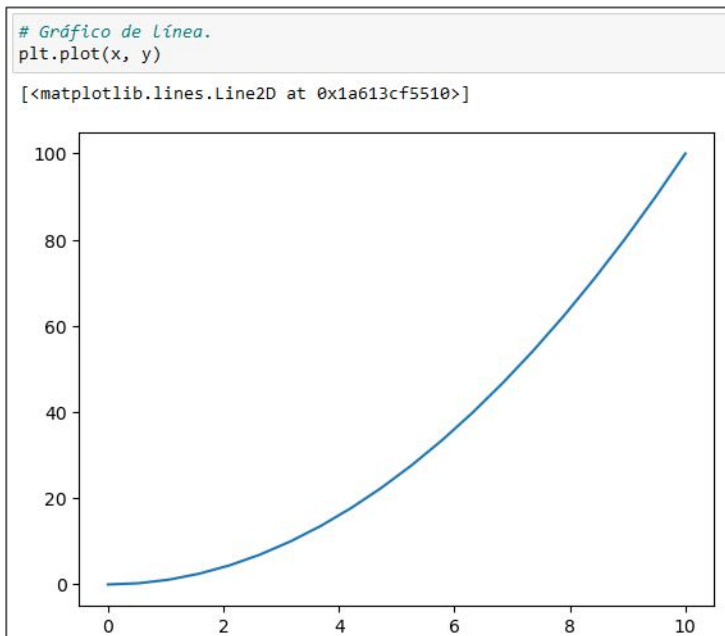
Trabajaremos con Jupyter Notebooks, ya que tiene buena integración con la visualización de gráficos.

Importar la librería:

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

Gráfico de línea o plot

La forma más rápida de generar un gráfico es mediante la invocación de las funciones de matplotlib.pyplot, aunque más adelante veremos otro método. Un gráfico de línea representa la relación entre dos variables, x e y. Cada punto representado en el plano lo une mediante una línea. x e y pueden ser ndarrays de Numpy.



Gráficos de línea

Requieren que los datos estén ordenados de forma creciente en eje x para ver algo lógico.

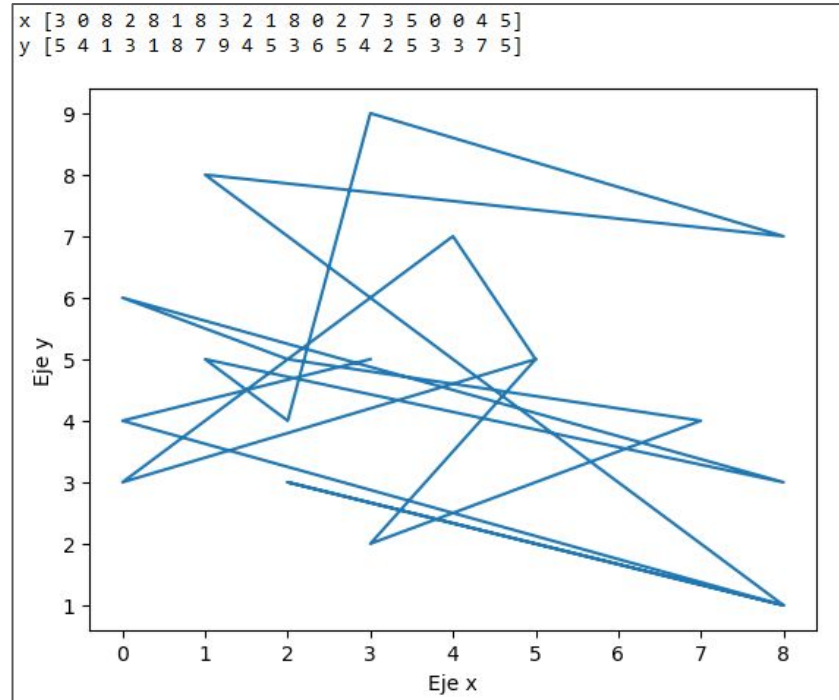


Gráfico de línea o plot.

Atributos

```
# Gráfico de Línea.  
plt.plot(x, y)  
plt.xlabel("Este es mi Eje x")  
plt.ylabel("Este es mi Eje Y")  
plt.title("Título del gráfico");
```

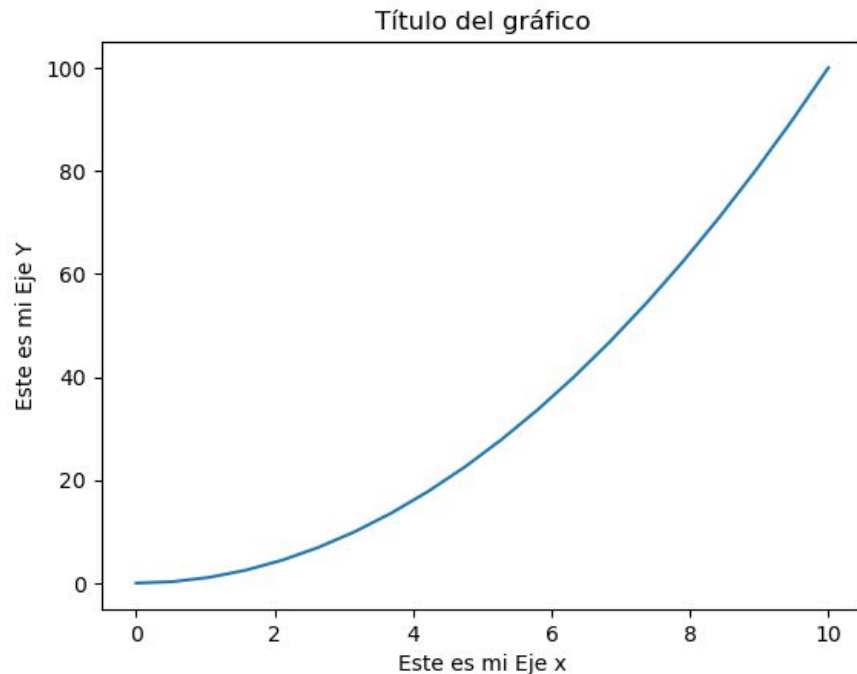


Gráfico de línea o plot.

Atributos

- color: color de línea
- linewidth: ancho línea
- alpha: nivel de transparencia
- linestyle: estilo de línea
- marker: estilo del punto del dato
- markersize: tamaño del punto
- markeredgewidth: grosor línea del dato
- markeredgewidth: color de círculo del punto del dato
- markerfacecolor: color interno del punto del dato.

```
#Atributos de plot
plt.plot(x, y,
         color = "red",
         linewidth = 2,
         alpha = 0.3,
         linestyle = ":",
         marker = 'o',
         markersize = 20,
         markerfacecolor = "blue",
         markeredgewidth = 2,
         markeredgewidth = "green")
```

[<matplotlib.lines.Line2D at 0x1a614084f90>]

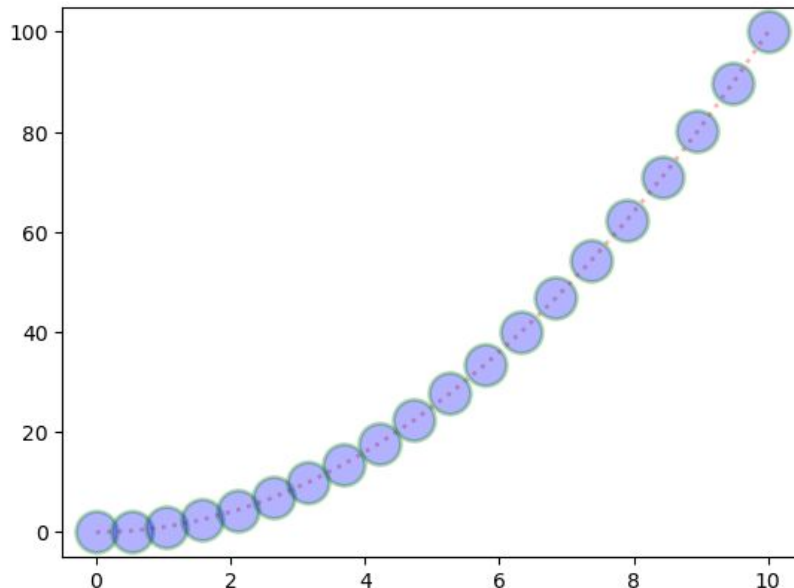


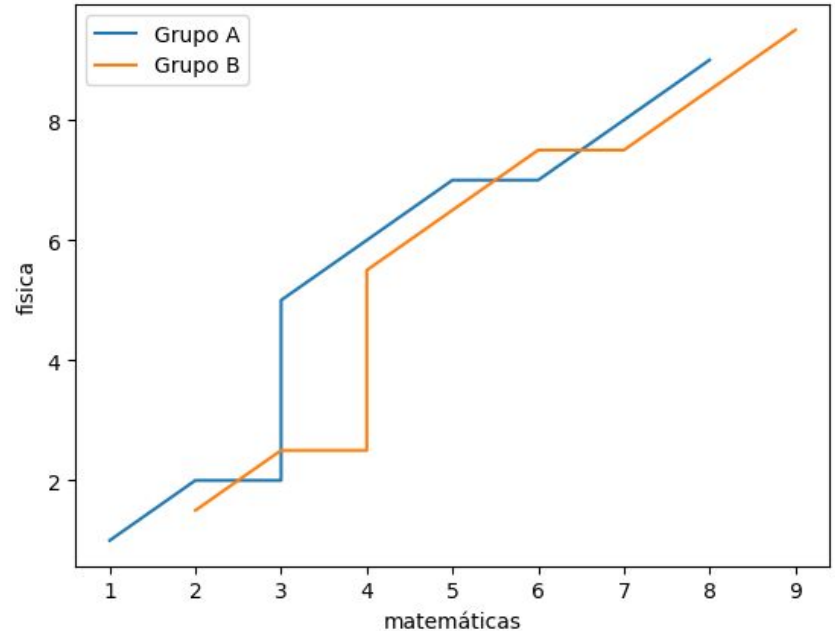
Gráfico de línea o plot.

Podemos pintar más de una línea en el mismo gráfico.

Por ejemplo, imagina que tenemos en variable X nota de matemáticas, en variable Y nota de físicas.

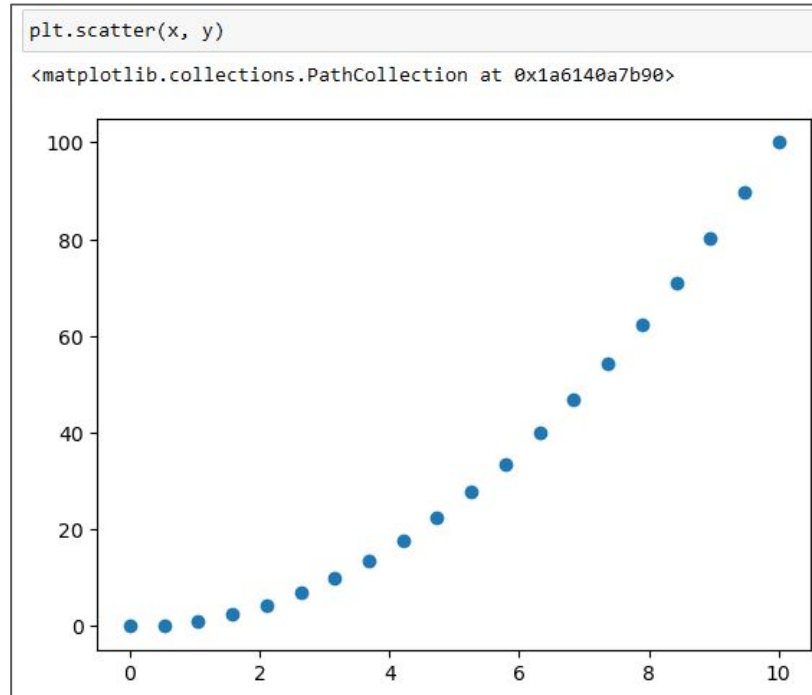
Cada línea representa una clase de alumnos (Grupo A y Grupo B)

```
plt.plot(mates_a, fisica_a, label="Grupo A")  
plt.plot(mates_b, fisica_b, label="Grupo B")  
plt.xlabel("matemáticas")  
plt.ylabel("física")  
plt.legend();
```



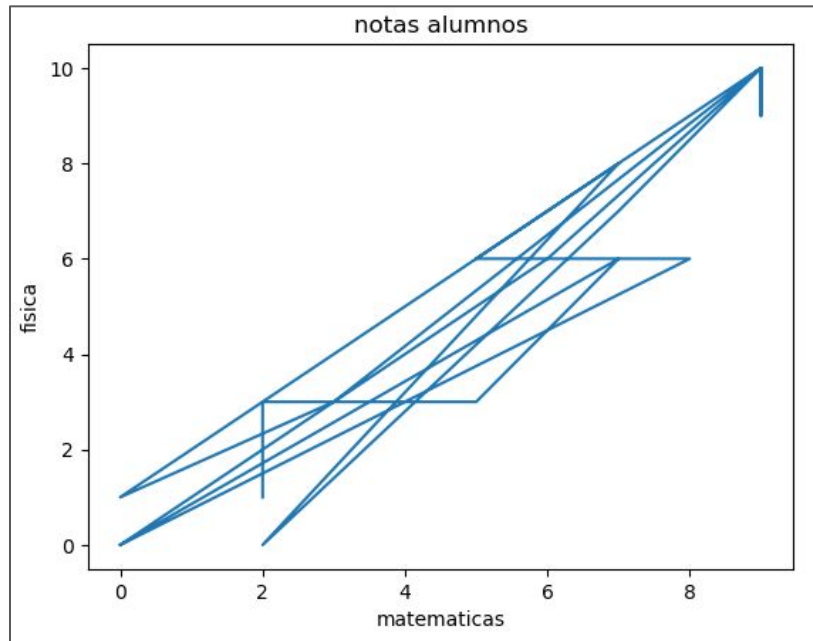
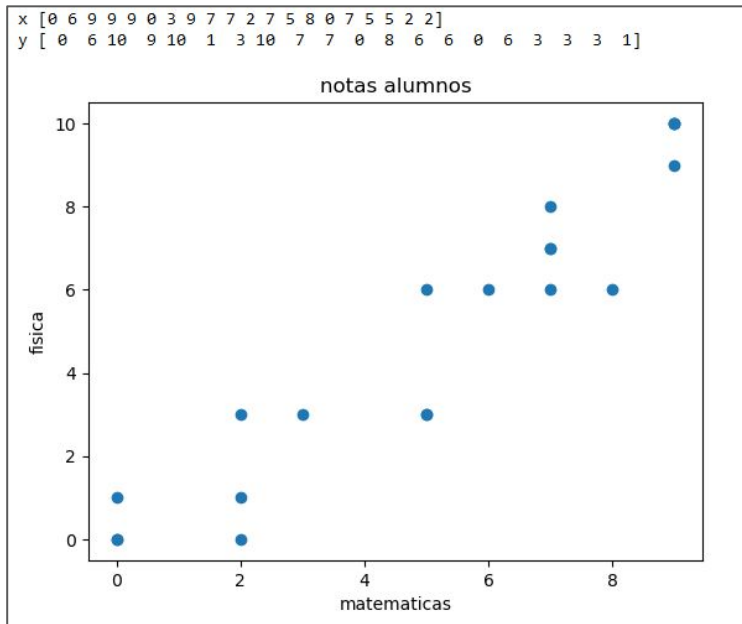
Nube de puntos

La nube de puntos representa los puntos en el plano (x, y) donde cada uno de los puntos representa su valor correspondiente en la variable x e y. Es similar al gráfico de línea, pero sin trazar una línea que una los puntos.



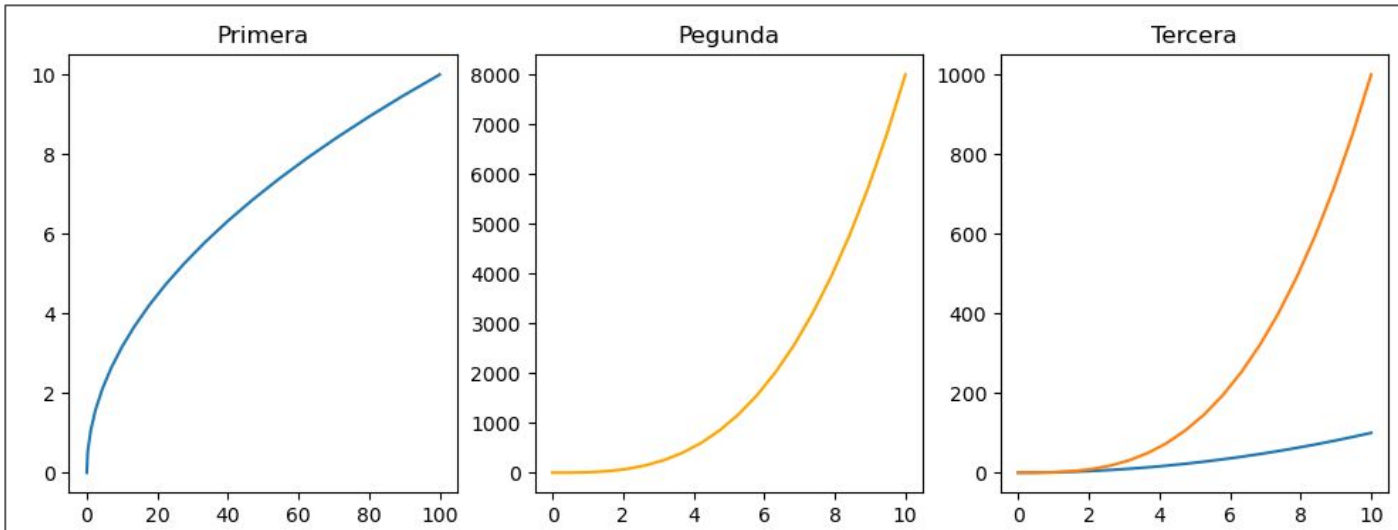
Nube de puntos

Cuando los puntos que representan a los datos dan muchos “saltos” o los datos están desordenados, un gráfico de línea es imposible de visualizar correctamente.



Figuras y subplots

En Matplotlib la forma más correcta de generar gráficas es mediante el objeto “figure”. Un objeto figure puede almacenar 0 o más gráficas, que se llaman subplots o axis.



Figuras y subplots

Creamos un objeto figure indicando el número de subplots en forma de filas y columnas, creando una cuadrícula de gráficas. También indicamos el tamaño total de la figura.

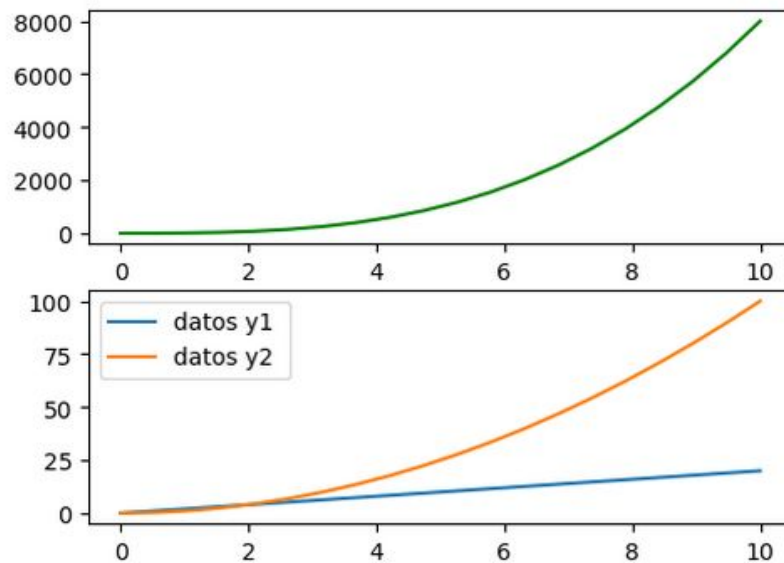
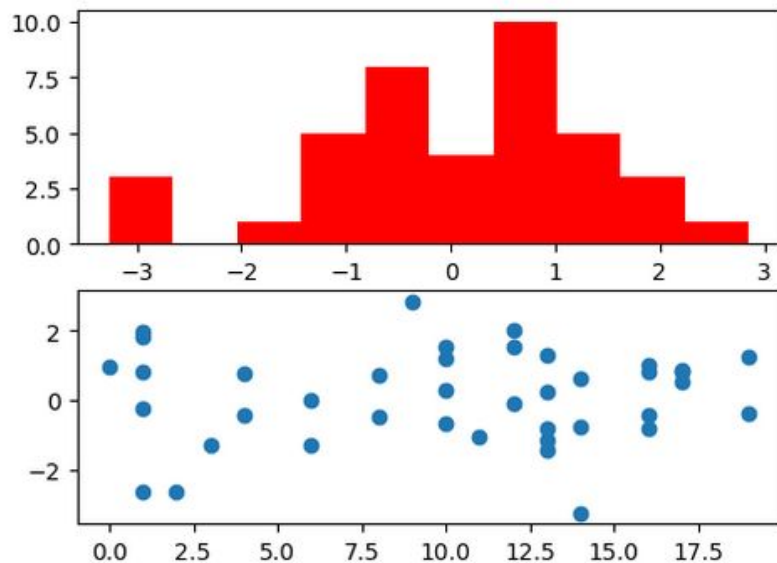
```
fig, ax = plt.subplots(nrows=1, ncols=3, figsize=(12,4))
```

Accedemos a cada subplot mediante la matriz ax y el índice según su posición.

ax[0][0] será la gráfica de la fila 0 y la columna 0, comenzando por la parte superior de la imagen.

```
fig, ax = plt.subplots(nrows=2, ncols=2, figsize=(12,4))
#Gráfica índice [0][0]
ax[0][0].hist(yr, color="red")
#Gráfica 0,1
ax[0][1].plot(x, y**3, color="green")
#Gráfica 1,0
ax[1][0].scatter(xr, yr)
#Gráfica 1,1
ax[1][1].plot(x, y1, label="datos y1 ")
ax[1][1].plot(x, y2, label="datos y2")
ax[1][1].legend()
```

<matplotlib.legend.Legend at 0x1f1213f0b10>



Figures

Se puede almacenar la figura completa en modo archivo de imagen mediante la instrucción `savefig()` indicando el nombre, formato (según la extensión) y la calidad en forma de dpi:

```
fig.savefig("mi_segunda_figura.png", dpi=200)
```

Gráfico de tartas

[Mas info aqui](#)

Debemos indicar las etiquetas (nombres) y las cantidades que corresponden a cada uno.

Con el atributo “autopct” indicamos que muestre el porcentaje y el formato.

```
manzanas = [20,10,25,30]  
nombres = ["Ana","Juan","Diana","Catalina"]  
plt.pie(manzanas, labels=nombres, autopct="%0.1f %%");  
plt.title("Manzanas por persona");
```

