HENRY

Matplotlib

Data Science

Agenda

- Matplotlib
- Plot()
- Leyenda
- Subplots
- Scatter plots
- Histograma

OBJETIVOS DE LA CLASE

Al finalizar esta lecture estarás en la capacidad de...

- ➜ Identificar los conceptos básicos de Matplotlib y su uso.

Al **finalizar** cada uno de los temas, tendremos un **espacio de consultas**.

Hay un **mentor** asignado para responder el **Q&A**.

¡Pregunta, pregunta, pregunta! :D

Matplotlib

¿Qué es?

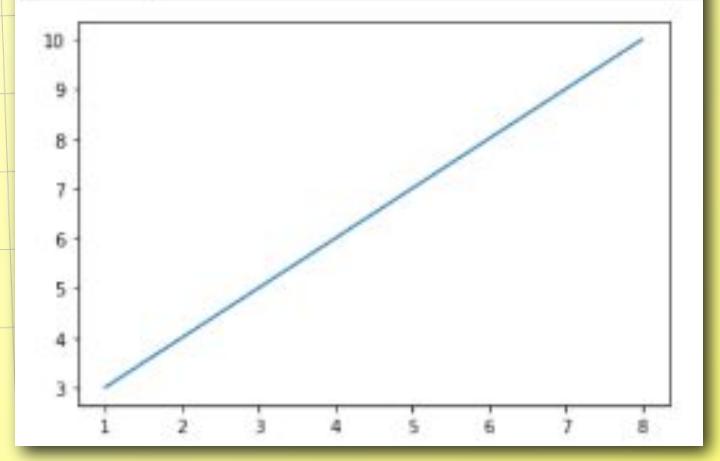
- Matplotlib es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación Python y su extensión matemática NumPy.
- Proporciona una API, pylab, diseñada para recordar a la de MATLAB.

Plot()

¿Qué es?

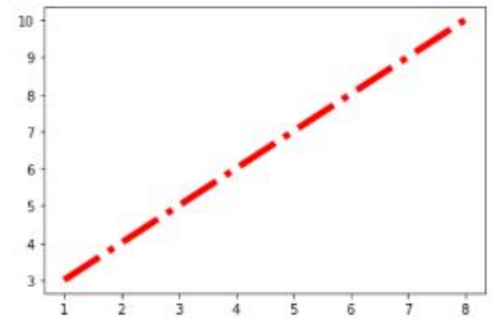
- La función se utiliza para dibujar puntos (marcadores) en un diagrama.
- De forma predeterminada, la función dibuja una línea de un punto a otro.
- El parámetro 1 es una matriz que contiene los puntos del eje x.
- El parámetro 2 es una matriz que contiene los puntos del eje y.

```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

plt.plot(xpoints, ypoints)
plt.show()
```



Distintas figuras

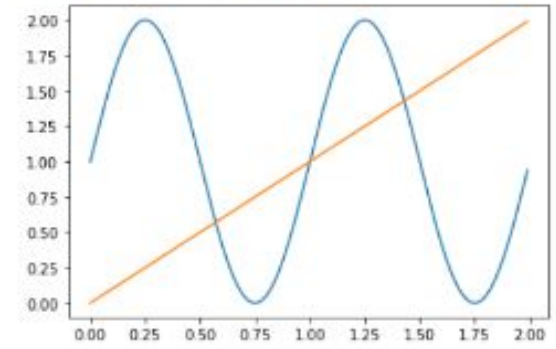
```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

plt.plot(xpoints, ypoints, color='red', linewidth = 5, linestyle = '-.')
plt.show()
```



plot(x, y, color = , linewidth = , linestyle =)

```
import matplotlib.pyplot as plt
import numpy as np

# Generamos las dos curvas
x = np.arange(0.0, 2.0, 0.01)
y = 1 + np.sin(2 * np.pi * x)

x2 = np.arange(0.0, 2.0, 0.01)
y2 = x2

# Generamos la figura y los ejes
fig = plt.figure()
ax = plt.axes()

# Ploteamos las dos lineas
ax.plot(x, y)
ax.plot(x2,y2)
```

```
[<matplotlib.lines.Line2D at 0x1f91633d7c0>]
```



Leyendas

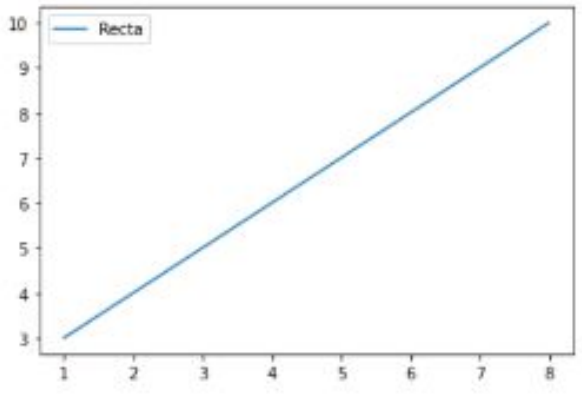
¿Qué son?

Las leyendas son cuadros que se agregan dentro del gráfico que nos brindan información sobre los distintos objetos que estamos graficando. Matplotlib se encarga de manejar muy bien las leyendas. Sólo debemos decirle el nombre de cada objeto que graficamos y avisarle que queremos que agregue la leyenda al gráfico.

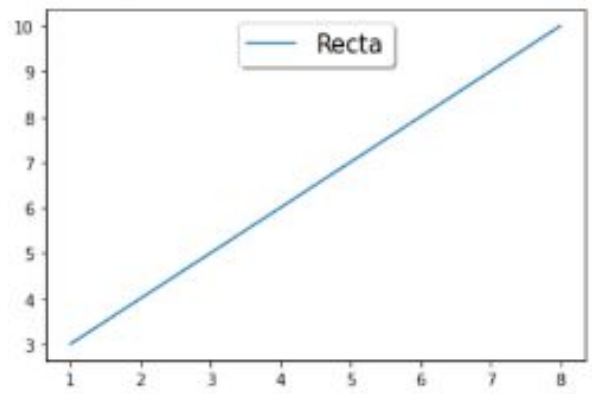
Al igual que los otros objetos de la librería Matplotlib, las propiedades de la leyenda que agregamos al gráfico son muy adaptables a través de parámetros que le podemos pasar al método legend.

Ejemplos

```python
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

ax = plt.axes()
plt.plot(xpoints, ypoints, label = 'Recta')
ax.legend()
plt.show()
```




```python
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

ax = plt.axes()
plt.plot(xpoints, ypoints, label = 'Recta')
ax.legend(loc='upper center', shadow=True, fontsize=15)
plt.show()
```



subplots

¿Qué es?

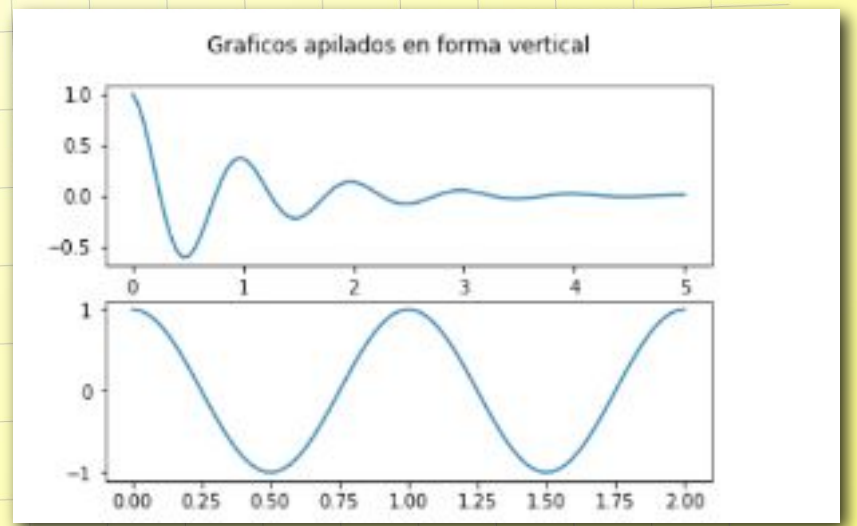
Para visualizar dos gráficos en una misma figura al mismo tiempo y compararlos debemos definir el objeto subplots de matplotlib.

subplots crea un objeto fig que corresponde a la figura (todo el rectángulo donde vamos a graficar) y varios ejes en el objeto axes, los cuales corresponden a los distintos subplots que vamos a hacer dentro de la figura.

Ejemplo

```
# Primero definimos dos curvas (x1,y1) y (x2,y2)
x1 = np.linspace(0.0, 5.0, 100)
x2 = np.linspace(0.0, 2.0, 100)

y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
y2 = np.cos(2 * np.pi * x2)


# Creamos la figura 'fig' y dos ejes en 'axes'.
# Como ponemos (2,1) quiere decir que queremos 2 filas
fig, axes = plt.subplots(2,1)
fig.suptitle('Graficos apilados en forma vertical')
axes[0].plot(x1, y1)
axes[1].plot(x2, y2)
```

```
[<matplotlib.lines.Line2D at 0x1f9165e6af0>]
```

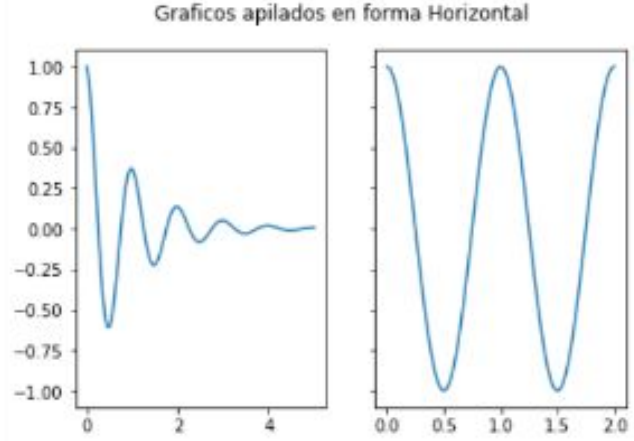


Ejemplo

También se pueden apilar los subplots de manera horizontal en lugar de vertical. Para esto debemos cambiar el argumento que pasamos al crear el objeto subplots.

```
# Si ponemos (1,2) quiere decir que queremos 2 columnas
# si pasamos el argumento 'sharey=True', compartiran los valores del eje y
fig, axes = plt.subplots(1,2,sharey=True)
fig.suptitle('Graficos apilados en forma Horizontal')
axes[0].plot(x1, y1)
axes[1].plot(x2, y2)
```

```
[<matplotlib.lines.Line2D at 0x276c35d2a00>]
```



scatter plots

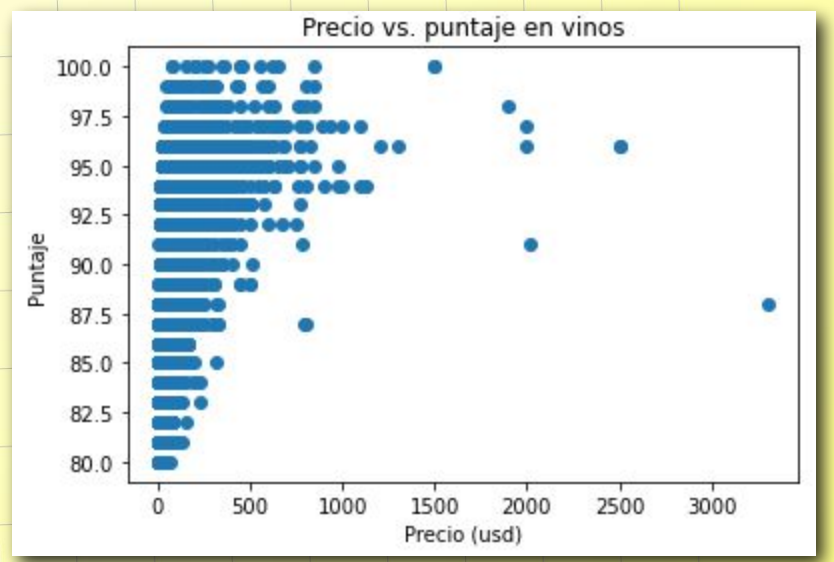
¿Qué es?

- Otro tipo de gráfico que se utiliza mucho es el Scatter Plot.
- El mismo plotea una serie de puntos en un gráfico.
- En general estos puntos tomarán su coordenada horizontal de una lista de valores x y su coordenada vertical de una lista de valores y.

Ejemplo

```
fig = plt.figure()
ax = plt.axes()

ax.scatter(x, y)
ax.set(xlabel='Precio (usd)', ylabel='Puntaje',
       title='Precio vs. puntaje en vinos')
# plt.plot(x, y)
```

```
[Text(0.5, 0, 'Precio (usd)'),
 Text(0, 0.5, 'Puntaje'),
 Text(0.5, 1.0, 'Precio vs. puntaje en vinos')]
```

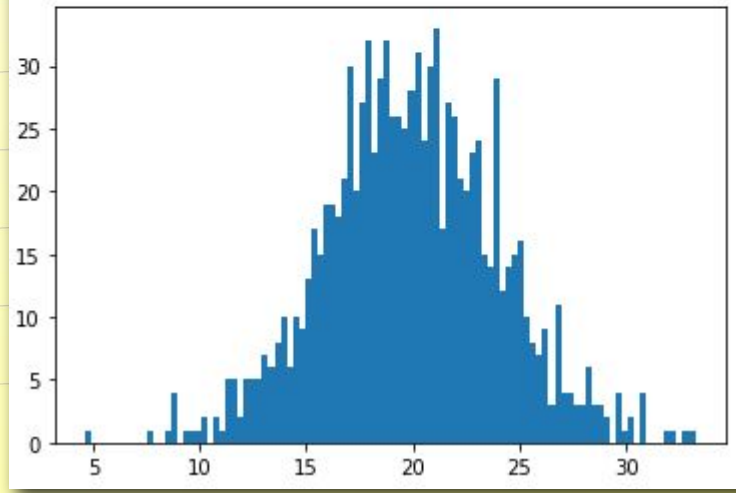


Histograma

¿Qué es?

Matplotlib también permite graficar fácilmente histogramas mediante la función hist.

```
# Aca decidimos la cantidad de bines que queremos tomar
num_bins = 100

# Creamos la figura y los ejes
fig, ax = plt.subplots()

# Ploteamos el histograma
n, bins, _ = ax.hist(valores, bins = num_bins)
```



Por ejemplo, creamos una serie de 1000 valores a partir de una distribución gaussiana en el vector valores...

¿PREGUNTAS?

¿Alguien dijo Homework?

HENRY

Próxima lecture

Seaborn

¡Feedback!

Click on me

Dispones de un formulario en:

- Homeworks
- Guías de clase
- Slack


HENRY