

Python para el análisis de datos- Lectura 7

Ing. Pedro Rotta

Universidad de Piura - Vida Universitaria

Enero-2022

Matplotlib

Es una librería fundamental en python. Fue elaborada por Jhon Hunter mientras hacía su doctorado en el 2002 para poder desarrollar gráficas parecidas al Matlab en python.

Su sintaxis de instalación es parecida a las que ya hemos visto.

```
pip install matplotlib
```

En colab sería:

```
!pip install matplotlib
```

Para importar la librería lo escribimos de esta manera:

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

figure y plot

El objeto principal de matplotlib es figure. Esto crea un molde o un objeto sobre el que se puede definir trazas. La sintaxis es:

```
fig1 = plt.figure(figsize = (float,float))
```

Este objeto no es obligatorio, pero una vez definido, puedes elegir el tamaño de la figura que va a contener las trazas, cuántos subplots se necesitan, entre otras cosas.

Luego de definir la figura se puede implementar la traza con plot con la siguiente sintaxis:

```
fig1 = plt.plot(x,y)
```

Donde x e y son las variables a graficar como se ve en el problema 1.

Etiquetas y subplots

La sintaxis para crear etiquetas a las axis va de la sgte forma :

```
fig1 = plt.xlabel('Label x')
```

```
fig1 = plt.ylabel('Label y')
```

Otra forma de sintaxis es mediante subplots:

```
fig1,ax = subplots()
```

```
ax.plot(x,y)
```

```
ax.set_xlabel('x label')
```

```
ax.set_ylabel('y label')
```

Ver problema 2

Subplots

También se puede colocar varios trazos en una misma figura, con una etiqueta respectiva:

```
fig,ax = subplots()  
ax.plot(x,y1)  
ax.plot(x,y2)  
ax.legend(('one','two'),loc = 'best')
```

Ver problema 3

Subplots

También se puede colocar varios trazos en una misma figura, con una etiqueta respectiva:

```
fig,ax = subplots()  
ax.plot(x,y1)  
ax.plot(x,y2)  
ax.legend(('one','two'),loc = 'best')
```

Ver problema 3

Para hacer 2 trazas en una misma figura, pero en diferente traza, se tiene que colocar la siguiente sintaxis:

```
fig,ax = subplots(m,n)  
ax[0].plot(x,y,'r-o')  
ax[0].plot(x,y*3,'g--s')
```

Ver problema 4.

Spines

Dentro de todo rectángulo que contiene una traza existen 4 spines que definen el comportamiento de las axisas. Cuando queremos cambiar el estilo de las axisas, podemos hacerlo mediante los spines.

```
ax.spines['left'].set_position('center')
ax.spines['right'].set_color('none')
ax.spines['bottom'].set_position('center')
ax.spines['top'].set_color('none')
```

Ver problema 5.

Compartir axisa

Para compartir una axisa en el gráfico de distintas trazas, usamos la siguiente sintaxis:

```
fig,axs = subplot(n,1,sharex = True, sharey = True)
```

Ver problema 6.

Patches

Puedes dibujar círculos, polígonos, etc. Usando el módulo patches de matplotlib.

Su sintaxis es:

```
from matplotlib.patches import Circle, Rectangle,  
Ellipse  
ax.add_patch(Circle((0,0),1,color = 'g'))  
ax.axis('equal')  
ax.grid(True)
```

Ver problema 7.

Patches

Puedes dibujar círculos, polígonos, etc. Usando el módulo patches de matplotlib.

Su sintaxis es:

```
from matplotlib.patches import Circle, Rectangle,
Ellipse
ax.add_patch(Circle((0,0),1,color = 'g'))
ax.axis('equal')
ax.grid(True)
```

Ver problema 7.

Para dibujar un rectángulo

```
ax.add_patch(Rectangle((-1,1),width = 1,height = 2,
color = 'r',alpha = 0.5))
```

Ver problema 8