Tarea 14

Parte 1: Hacer un resumen del uso de matplotlib que se usa en Python para construir representaciones gráficas.

Nombre: German Jordi Arreortúa Reyes.

Materia: Programación Avanzada.

Fecha de entrega: 21 de junio de 2022

Para realizar gráficos en Python se trabaja con la biblioteca matplotlib. Al construir graficos con matplolib tengamos en cuenta dichas gráficos son en realidad una jerarquía de objetos de Python anidados, refiriéndonos con jerarquía a una estructura en forma de árbol de objetos matplotlib subyacentes a cada gráfico.

Un objeto Figure es el contenedor más externo para un gráfico matplotlib, que puede contener varios objetos Axes (diagrama o gráfico individual). Podemos pensar en el objeto Figure como un contenedor en forma de caja que contiene uno o más Axes, debajo de los Axes en la jerarquía hay objetos más pequeños, como marcas, líneas individuales, leyendas y cuadros de texto, a continuación se muestra una representación de dicha jerarquía.

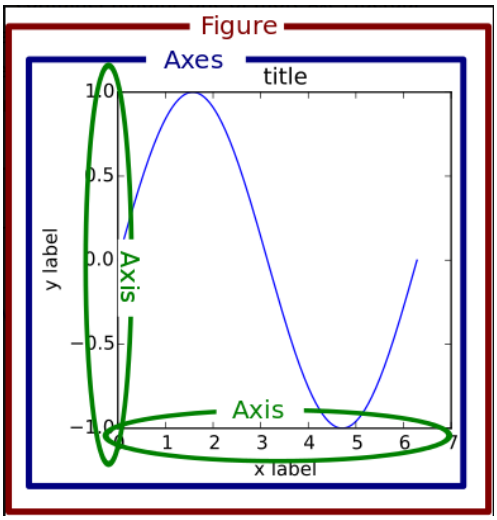


Figura 1

Ahora veamos cómo se crea una figura con un solo Axes

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

rng = np.arange(50)
rnd = np.random.randint(0, 10, size=(3, rng.size))
yrs = 1950 + rng

fig, ax = plt.subplots(figsize=(5, 3))
ax.stackplot(yrs, rng + rnd, labels=['Eastasia', 'Eurasia', 'Oceania'])
ax.set_title('Combined debt growth over time')
ax.legend(loc='upper left')
ax.set_ylabel('Total debt')
ax.set_xlim(xmin=yrs[0], xmax=yrs[-1])
```

```
fig.tight_layout()
plt.show()
```

Primeramente importamos matplotlib.pyplot como plt, y para este ejemplo también se importa la biblioteca numpy.

rng es un objeto ‘numpy.ndarray’ con los valores de 0 a 49, rnd es un objeto ‘numpy.ndarray’ con forma matricial de tamaño 3x50 en el cual se amacena números enteros aleatorios de 0 a 9, e yrs es un ‘numpy.ndarray’ donde a cada entero de rng se le sumo 1950.

Ahora pasemos a los pasos que involucran a la biblioteca matplotlib.

- fig, ax = plt.subplots(figsize=(5, 3))
  Se crea el objeto figura conteniendo un solo Axes, figsize=(5, 3) indica el tamaño del gráfico.

- ax.stackplot(yrs, rng + rnd, labels=['Eastasia', 'Eurasia', 'Oceania'])

  Esto lo que realiza en la representación gráfica es que le estamos indicando básicamente el dominio y el rango, es decir va tomar los elementos de yrs (que forman el eje x) y los de rng + rnd (eje y) y los ponemos en correspondencia, en este caso a cada elemento de yrs le correponden 3 elementos de rng + rnd, pues recordemos que este tiene forma matricial, es como si le pidiéramos que realice tres líneas apiladas rellenadas de diferente color entre líneas y se generen las etiquetas para el gráfico (ver figura 2).

- ax.set_title('Combined debt growth over time')
  Se agrega un título.

- ax.legend(loc='upper left')
  Se indica en qué posición se pone el ‘legend’ el cúal es el lugar donde van las etiquetas 'Eastasia', 'Eurasia', 'Oceania'. Si en lugar de loc='upper left' ponemos loc='upper rigth' entonces se coloca en la parte superior derecha como se muestra en la figura 3.

- ax.set_ylabel('Total debt')
  Se coloca una etiqueta al eje Y.

- ax.set_xlim(xmin=yrs[0], xmax=yrs[-1])
  Se indica el valor minino y máximo en el eje X.

- fig.tight_layout()
  Se aplica al objeto Figure como un todo para limpiar el relleno de espacios en blanco.

- plt.show()
  Es para que nos muestra la representación gráfica la cual se muestra a continuación

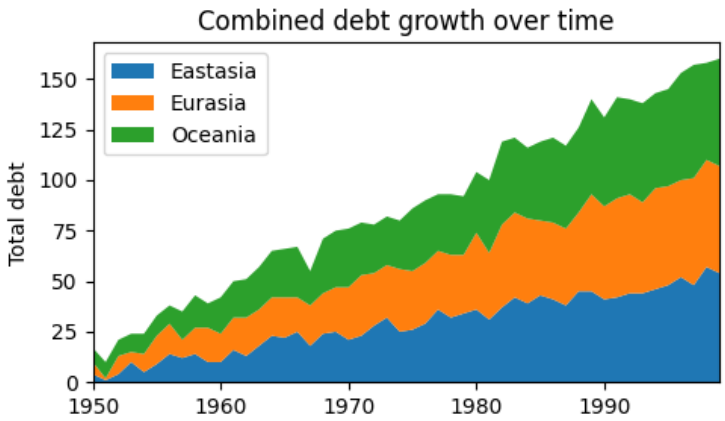


Figura 2

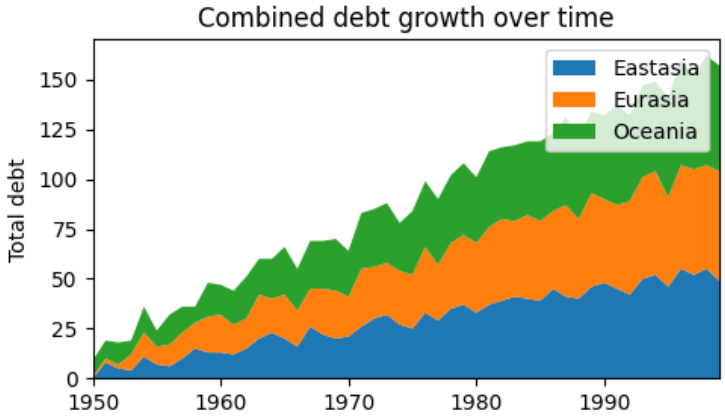


Figura 3

A continuación se muestra un ejemplo para realizar un gráfico con dos Axes.

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

x = np.random.randint(low=1, high=11, size=50)
y = x + np.random.randint(1, 5, size=x.size)
data = np.column_stack((x, y))

fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(8, 4))

ax1.scatter(x=x, y=y, marker='o', c='r',
            edgecolor='b')
ax1.set_title('Scatter: $x$ versus $y$')
ax1.set_xlabel('$x$')
ax1.set_ylabel('$y$')

ax2.hist(data, bins=np.arange(data.min(), data.max()),
         label=('x', 'y'))
ax2.legend(loc=(0.65, 0.8))
ax2.set_title('Frequencies of $x$ and $y$')
```

```
ax2.yaxis.tick_right()
plt.show()
```

Primeramente se crearon x e y que son objetos ‘numpy.ndarray’ el primero con 50 valores aleatorios del 1 al 10 y el segundo con 50 valores que se obtiene al sumar cada valor de x con los valores obtenidos aleatoriamente de 1 a 4 y data es un ‘numpy.ndarray’ con forma matricial siendo las columnas x e y.

A continuaciones véalos los pasos en la construcción del gráfico.

Primeramente se crea el objeto figura conteniendo dos Axes ax1 y ax2. Puesto que nrows=1, ncols=2 entonces se está creando un figura “1x2” (una fila dos columnas) el resultado devuelto de plt.subplots(1, 2) ahora es un objeto Figura y una matriz NumPy de objetos Axes. En términos de la representación gráfica, se está indicando que la figura tenga dos subgráficos que se muestren en dos columnas, como en la figura 4 de tamaño (8,4).

Con ax1.scatter(x=x, y=y, marker='o', c='r', edgecolor='b') nos está indicando que el Axes ax1 sea un diagrama de dispersión tal que los valores en el eje ’x’ sean los de x (objeto ‘numpy.ndarray’) y los valores en el eje ‘y’ sean los de y (objeto ‘numpy.ndarray’). marker='o' es el estilo de marcador para el diagrama de dispersión en este caso es ‘o’ el cual es un circulo pero existen varios por ejemplo “.” Lo marcara con un punto, ‘v’ será un triángulo hacia abajo, ets. En [3] encontramos los marcadores. c='r' es el color del marcador, en este caso es rojo y finalmente edgecolor='b' es el color del borde del marcador, en este caso es azul. En la figura 5 se muestran el diagrama de dispersión obtenido para marker='v', c='g', edgecolor='r'.

Con ax2.hist(data, bins=np.arange(data.min(), data.max()),label=('x', 'y')), nos indica que se construirá el histograma de ‘data’ con el número de bins= np.arange(data.min(), data.max()) y etiquetas ‘x’ e ‘y’.
Y por último x2.yaxis.tick_right()mueva los ticks y las etiquetas de ticks (si están presentes) a la derecha de los ejes.

A continuación se muestra el grafico obtenido del ejemplo anterior.

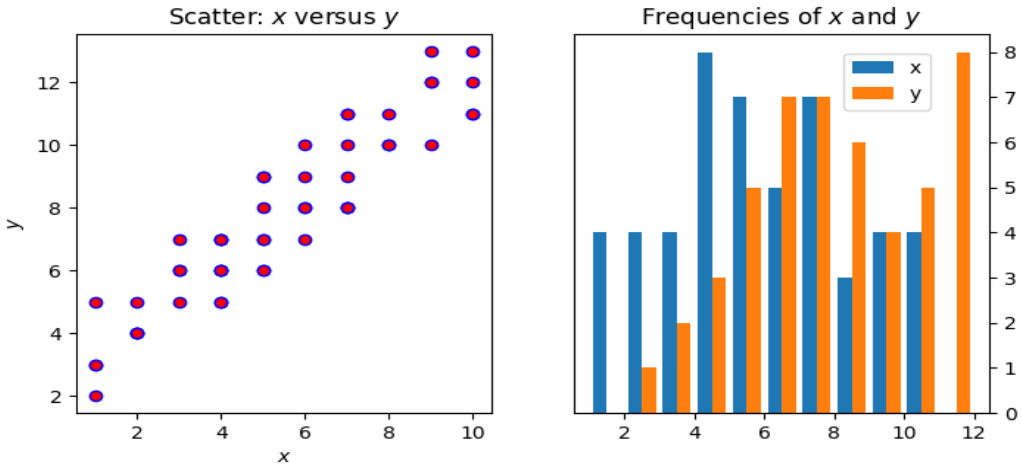


Figura 4

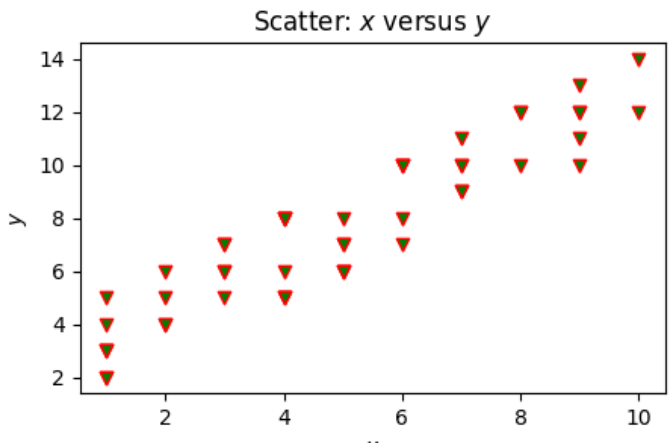


Figura 5

En [1] podemos encontrar más ejemplos de cómo se realizan representaciones graficas usando matplotlib.

Fuentes:

[1] https://realpython.com/python-matplotlib-guide/

[2] https://matplotlib.org/stable/index.html

[3] https://matplotlib.org/stable/api/markers_api.html#module-matplotlib.markers

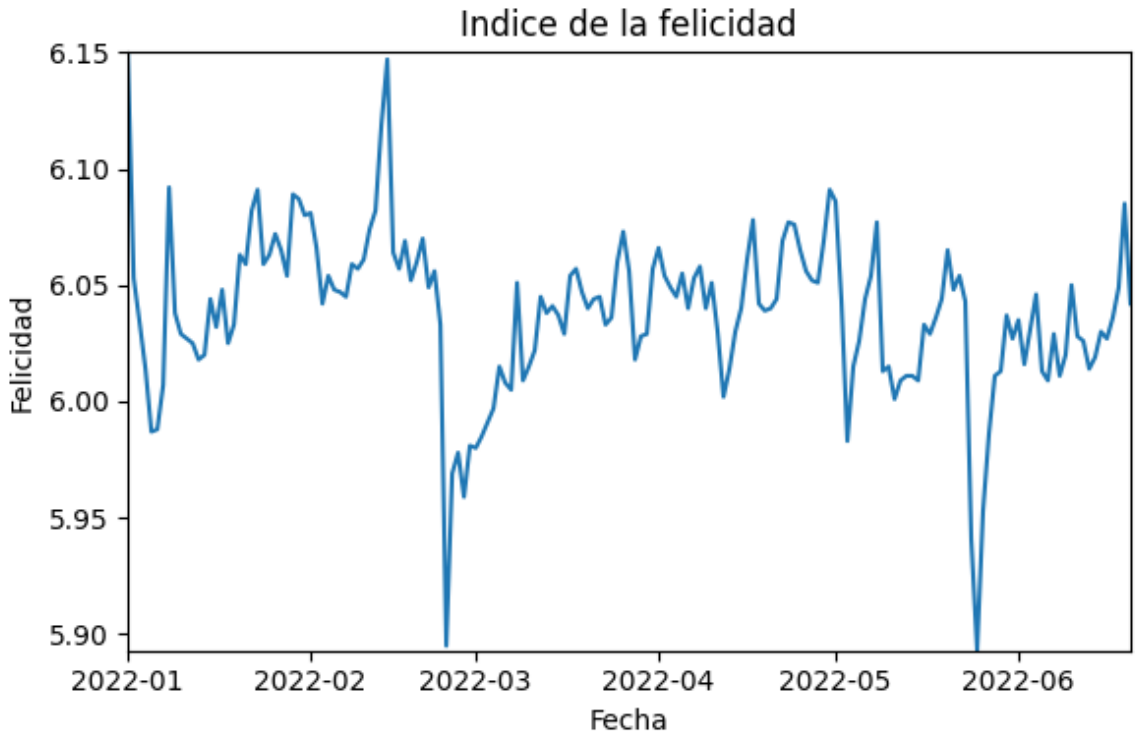
Tarea 14 parte 2

En el programa del índice de la felicidad escribir un método para dibujar la serie de tiempo del índice utilizando matplotlib. El programa debe dibujar toda la serie que hayan podido descargar. En el eje horizontal deben aparecer fechas como etiquetas y en el vertical el valor del índice. Incluya textos explicativos de lo que se presenta en cada eje y un título.

A continuación se muestra el procedimiento para graficar el índice de la felicidad.

```
def grafica(datos):
    fechas = []
    happiness = []
    for dato in datos:
        fechas.append(dato[0])
        happiness.append(dato[2])
    fig, ax = plt.subplots(figsize=(6, 4))
    ax.plot(fechas, happiness)
    ax.set_title('Indice de la felicidad')
    ax.set_xlabel('Fecha')
    ax.set_ylabel('Felicidad')
    ax.set_xlim(xmin=fechas[0], xmax=fechas[-1])
    ax.set_ylim(ymin=min(happiness), ymax=max(happiness))
    fig.tight_layout()
    plt.show()
```

Se muestra el grafico obtenido



Si se desea ponerle relleno a la gráfica en lugar de ax.plot(fechas, happiness) se cambia por ax.stackplot(fechas, happiness) y el resultado es el siguiente.