
Programación para la Computación Científica - IA



Python for Programmers

Data Visualization with matplotlib

Universidad Sergio Arboleda
Prof. John Corredor

Installing Matplotlib

Para hacer inferencias estadísticas necesarias, se hace necesario visualizar datos y Matplotlib es una de esas soluciones para los usuarios de Python. Es una biblioteca de trazado muy poderosa útil para aquellos que trabajan con Python y NumPy. El módulo más utilizado de Matplotlib es Pyplot, que proporciona una interfaz como MATLAB pero, en cambio, usa Python y es de código abierto.

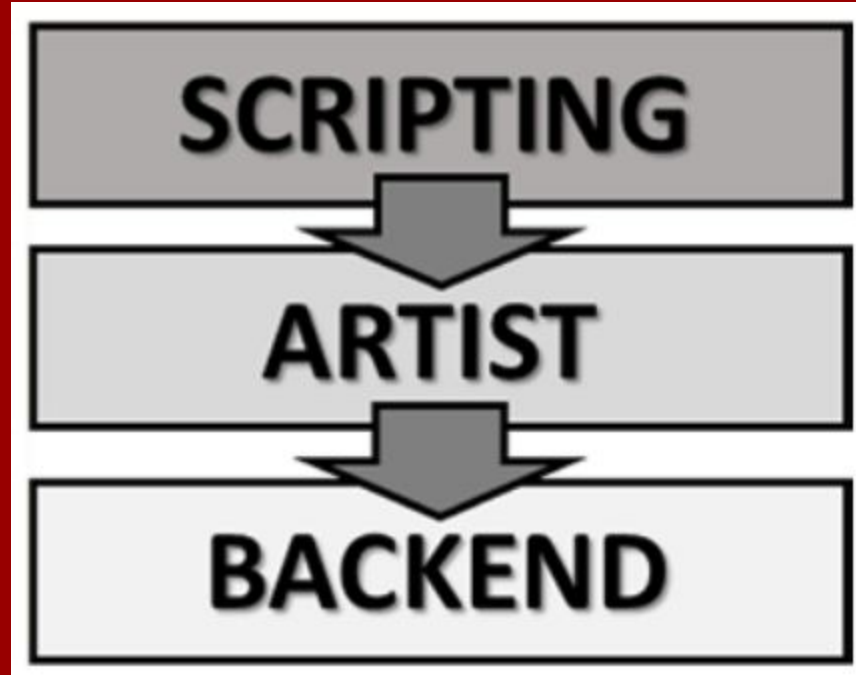
```
python -m pip install -U pip  
python -m pip install -U matplotlib
```

Matplotlib es una biblioteca de Python especializada en el desarrollo de gráficos bidimensionales (incluidos gráficos en 3D); En los últimos años, se ha generalizado en los círculos científicos y de ingeniería. (<http://matplotlib.org>).

- Extrema simplicidad en su uso
- Desarrollo gradual y visualización interactiva de datos
- Expresiones y textos en LaTeX
- Mayor control sobre los elementos gráficos
- Exportar en muchos formatos, como PNG, PDF, SVG y EPS.

Una **figura de Matplotlib** se caracteriza:

- **Figura:** es una figura completa que puede contener uno o más de un eje (parcelas). Puede pensar en una figura como un lienzo que contiene tramas.
- **Ejes:** es lo que generalmente pensamos como una trama. Una figura puede contener muchos ejes. Contiene dos o tres objetos (en el caso de 3D) del Eje. Cada eje tiene un título, una etiqueta x y una etiqueta y.
- **Eje:** son la línea numérica como objetos y se encargan de generar los límites del gráfico.
- **Artista:** Todo lo que se puede ver en la figura es un artista como objetos de texto, objetos Line2D, objetos de colección. La mayoría de los artistas están vinculados a los



En el diagrama de la arquitectura matplotlib, la capa que funciona en el nivel más bajo es el Backend. En esta capa están las API de matplotlib, un conjunto de clases que desempeñan el papel de implementación de los elementos gráficos en un nivel bajo.

- FigureCanvas es el objeto que encarna el concepto de área de dibujo.
- Renderer es el objeto que se basa en FigureCanvas.
- Evento es el objeto que maneja las entradas del usuario (eventos de teclado y ratón).

Arquitectura matplotlib: Artist Layer

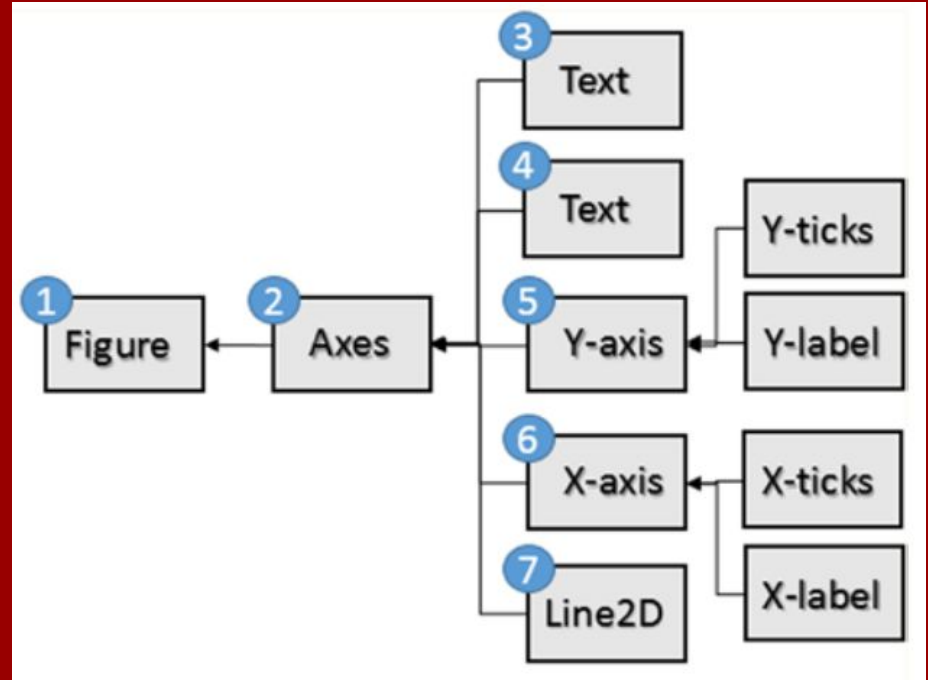
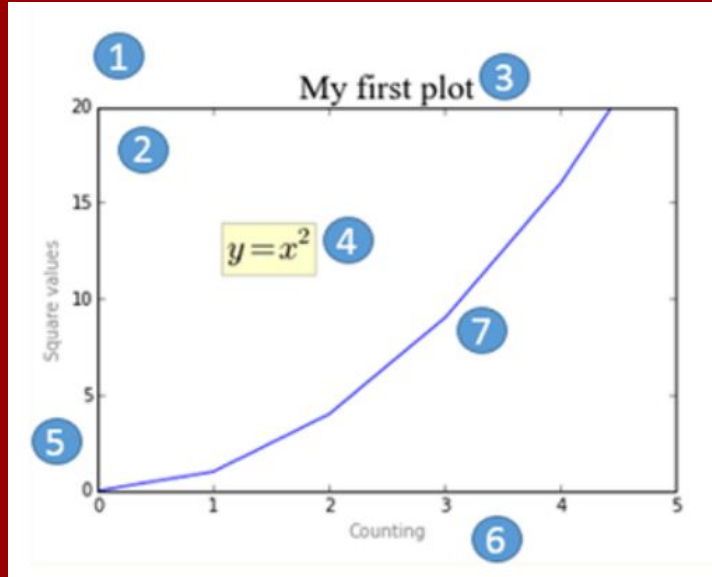


Figure: Each element of a chart corresponds to an instance of Artist structured in a hierarchy

Arquitectura matplotlib: Artist Layer

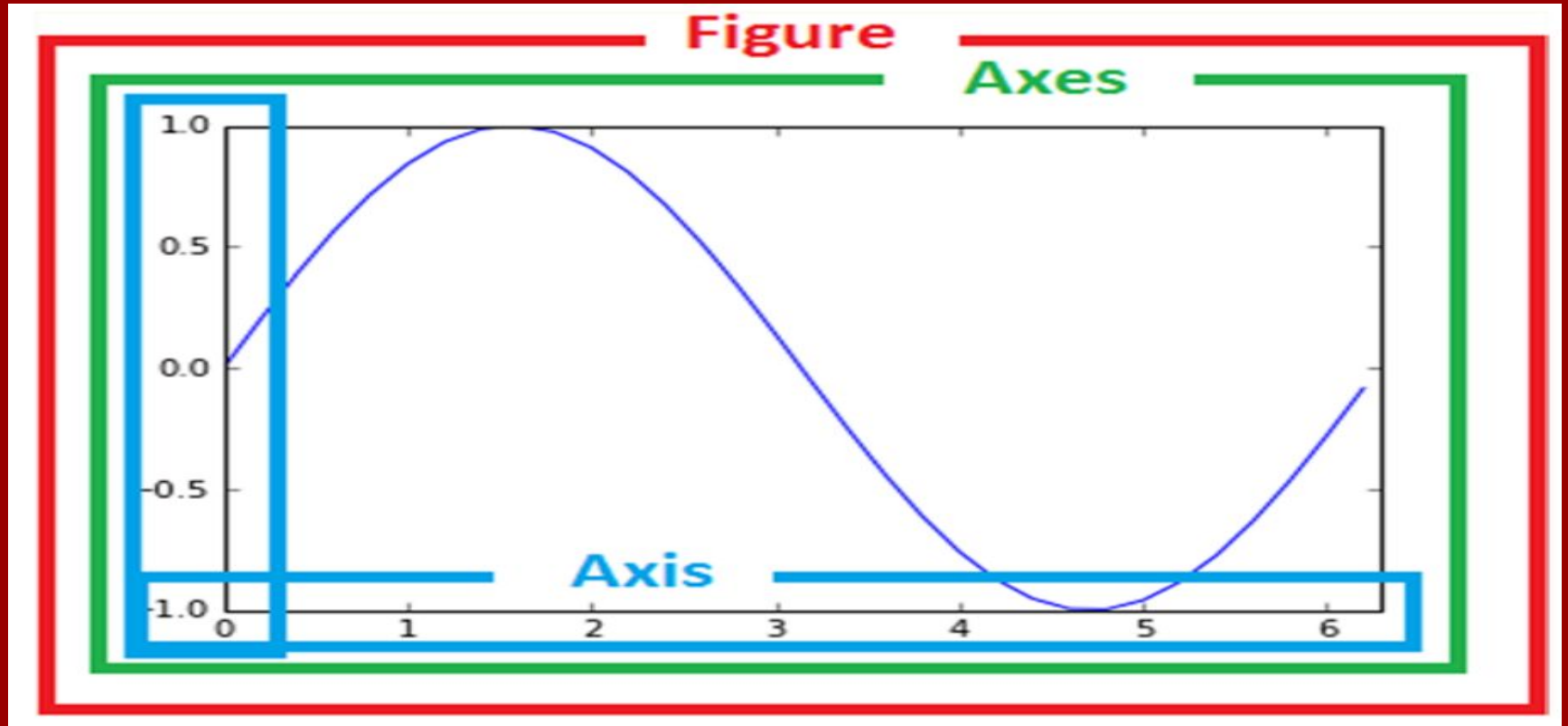


Figure: Los tres objetos principales de artista en la jerarquía de la capa del artista


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

```
plt
```

```
import matplotlib.pyplot as plt
plt.axis([0,5,0,20])
plt.title('My first plot',fontsize=20,fontname='DejaVu Sans')
plt.xlabel('Counting',color='red')
plt.ylabel('Square values',color='blue')
plt.text(1,1.5,'First')
plt.text(2,4.5,'Second')
plt.text(3,9.5,'Third')
plt.text(4,16.5,'Fourth')
plt.text(1.1,12,r'$y = x^2$',fontsize=20,bbox={'facecolor':'yellow','alpha':0.2})
plt.grid(True)
plt.plot([1,2,3,4],[1,4,9,16],'ro')
plt.legend(['First series'])
```

Location Code	Location String
0	best
1	upper-right
2	upper-left
3	lower-right
4	lower-left
5	right
6	center-left
7	center-right
8	lower-center
9	upper-center
10	center

```
plt.axis([0,5,0,20])
plt.title('My first plot',fontsize=20,fontname='DejaVu Sans')
plt.xlabel('Counting',color='red')
plt.ylabel('Square values',color='blue')
plt.text(1,1.5,'First')
plt.text(2,4.5,'Second')
plt.text(3,9.5,'Third')
plt.text(4,16.5,'Fourth')
plt.text(1.1,12,r'$y = x^2$',fontsize=20,bbox={'facecolor':'yellow','alpha':0.2})
plt.plot([1,2,3,4],[1,4,9,16],'ro')
plt.plot([1,2,3,4],[0.8,3.5,8,15],'g^')
plt.plot([1,2,3,4],[0.5,2.5,4,12],'b*')
plt.legend(['First series','Second series','Third series'],loc=?)
```

```
import datetime
import numpy as np
import matplotlib.pyplot as plt
events = [datetime.date(2019,1,23),
          datetime.date(2019,1,28),
          datetime.date(2019,2,3),
          datetime.date(2019,2,21),
          datetime.date(2019,3,15),
          datetime.date(2019,3,24),
          datetime.date(2019,4,8),
          datetime.date(2019,4,24)]
readings = [12,22,25,20,18,15,17,14]
plt.plot(events,readings)
```

>Grafico de Linea (Line Chart)

A line chart is a sequence of data points connected by a line. Each data point consists of a pair of values (x,y), which will be reported in the chart according to the scale of values of the two axes (x and y).

$$y = \sin(3 * x) / x$$

```
import matplotlib.pyplot as plt
import numpy as np
in x = np.arange(-2*np.pi,2*np.pi,0.01)
in y = np.sin(3*x)/x
x y2 = np.sin(2*x)/x
y y3 = np.sin(4*x)/x
p plt.plot(x,y)
plt.plot(x,y2)
plt.plot(x,y3)
```

Code	Color
b	blue
g	green
r	red
c	cyan
m	magenta
y	yellow
k	black
w	white

de Linea (Line Chart)

```
plt.plot(x,y,color='b')
plt.plot(x,y2,color='r')
plt.plot(x,y3,color='g')
plt.xticks([-2*np.pi, -np.pi, 0, np.pi, 2*np.pi],
           [r'$-2\pi$',r'$-\pi$',r'$0$',r'$+\pi$',r'$+2\pi$'])
plt.yticks([-1,0,+1,+2,+3],
           [r'$-1$',r'$0$',r'$+1$',r'$+2$',r'$+3$'])
```

Code	Color
b	blue
g	green
r	red

```
plt.yticks([-1,0,+1,+2,+3],
           [r'$-1$',r'$0$',r'$+1$',r'$+2$',r'$+3$'])
ax = plt.gca()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data',0))
ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data',0))
```

```
plt.plot(x,y,color='b')
plt.plot(x,y2,color='r')
plt.plot(x,y3,color='g')
plt.xticks([-2*np.pi, -np.pi, 0, np.pi, 2*np.pi],
            [r'$-2\pi$',r'$-\pi$',r'$0$',r'$+\pi$',r'$+2\pi$'])
plt.yticks([-1,0,+1,+2,+3],
            [r'$-1$',r'$0$',r'$+1$',r'$+2$',r'$+3$'])
```

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
data = {'series1':[1,3,4,3,5],
        'series2':[2,4,5,2,4],
        'series3':[3,2,3,1,3]}
df = pd.DataFrame(data)
x = np.arange(5)
plt.axis([0,5,0,7])
plt.plot(x,df)
plt.legend(data, loc=
```

```
df = pd.DataFrame(np.random.rand(6, 4),
                  index=['one', 'two', 'three', 'four', 'five', 'six'],
                  columns=pd.Index(['A', 'B', 'C', 'D'], name='Genus'))
```

```
df
```

```
df.plot.bar()
```

--->> vertical

Un histograma consiste en rectángulos adyacentes erigidos en el eje x, divididos en intervalos discretos llamados contenedores, y con un área proporcional a la frecuencia de las ocurrencias para ese contenedor. Este tipo de visualización se usa comúnmente en estudios estadísticos sobre la distribución de muestras.

```
import matplotlib.pyplot as plt
import numpy as np
pop = np.random.randint(0,100,100)
plt.hist(pop,bins=10)
```

Es muy similar a un histograma pero en este caso el eje x no se usa para hacer referencia a valores numéricos sino a categorías.

```
from matplotlib import pylab as plt
import numpy as np
sorted_ratings = [8.3, 8.2, 8.2, 8.3, 8.5, 8.4, 8.2, 8.5, 8.2, 8.2]
Years = ['1921', '1925', '1926', '1927', '1931', '1931', '1934', '1936', '1939', '1939']
x_pos = np.arange(len(Years))
plt.bar(x_pos, sorted_ratings)
plt.suptitle('Ratings based on years', fontsize=14)
plt.xticks(x_pos, Years)
plt.ylabel('Rating', fontsize=12)
plt.xlabel('Year', fontsize=12)
plt.show()
```

```
import matplotlib.pyplot as plt
import numpy as np
index = np.arange(5)
values1 = [5,7,3,4,6]
std1 = [0.8,1,0.4,0.9,1.3]
plt.title('A Horizontal Bar Chart')
plt.barh(index,values1,xerr=std1,error_kw={'ecolor':'0.1','capsize':6},alpha=0.7,
label='First')
plt.yticks(index+0.4,['A','B','C','D','E'])
plt.legend(loc=5)
```

> Barras: Pandas

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
data = {'series1':[1,3,4,3,5],
        'series2':[2,4,5,2,4],
        'series3':[3,2,3,1,3]}
df = pd.DataFrame(data)
df.plot(kind='bar')
```

```
import matplotlib.pyplot as plt
import numpy as np
series1 = np.array([3,4,5,3])
series2 = np.array([1,2,2,5])
series3 = np.array([2,3,3,4])
index = np.arange(4)
plt.axis([0,4,0,15])
plt.bar(index,series1,color='r')
plt.bar(index,series2,color='b',bottom=series1)
plt.bar(index,series3,color='g',bottom=(series2+series1))
plt.xticks(index+0.4,['Jan15','Feb15','Mar15','Apr15'])
```

```
import matplotlib.pyplot as plt
import numpy as np
series1 = np.array([3,4,5,3])
series2 = np.array([1,2,2,5])
series3 = np.array([2,3,3,4])
index = np.arange(4)
plt.axis([0,15,0,4])
plt.barh(index,series1,color='r')
plt.barh(index,series2,color='b',left=series1)
plt.barh(index,series3,color='g',left=(series2+series1))
plt.yticks(index+0.4,['Jan15','Feb15','Mar15','Apr15'])
plt.title('A Multiseries Horizontal Stacked Bar Chart')
```

```
import matplotlib.pyplot as plt
import pandas as pd
data = {'series1':[1,3,4,3,5],
        'series2':[2,4,5,2,4],
        'series3':[3,2,3,1,3]}
df = pd.DataFrame(data)
df.plot(kind='bar', stacked=True)
```

> Gráficos Tarta (PIE)

```
import matplotlib.pyplot as plt
labels = ['Nokia','Samsung','Apple','Lumia']
values = [10,30,45,15]
colors = ['yellow','green','red','blue']
plt.pie(values,labels=labels,colors=colors)
plt.axis('equal')
```

```
import matplotlib.pyplot as plt
labels = ['Nokia','Samsung','Apple','Lumia']
values = [10,30,45,15]
colors = ['yellow','green','red','blue']
explode = [0.3,0,0,0]
plt.title('A Pie Chart')
plt.pie(values,labels=labels,colors=colors,explode=explode,startangle=
plt.axis('equal')
```

```
import matplotlib.pyplot as plt
labels = ['Nokia','Samsung','Apple','Lumia']
values = [10,30,45,15]
colors = ['yellow','green','red','blue']
explode = [0.3,0,0,0]
plt.title('A Pie Chart')
plt.pie(values,labels=labels,colors=colors,explode=explode,
shadow=True,autopct='%1.1f%%',startangle=180)
plt.axis('equal')
```

```
import matplotlib.pyplot as plt
import pandas as pd
data = {'series1':[1,3,4,3,5], 'series2':[2,4,5,2,4], 'series3':[3,2,3,1,3]}
df = pd.DataFrame(data)
df['series1'].plot(kind='pie', figsize=(6,6))
```

```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
fig = plt.figure()
ax = Axes3D(fig)
X = np.arange(-2,2,0.1)
Y = np.arange(-2,2,0.1)
X,Y = np.meshgrid(X,Y)
def f(x,y):
    return (1 - y**5 + x**5)*np.exp(-x**2-y**2)
ax.plot_surface(X,Y,f(X,Y), rstride=1, cstride=1, cmap=plt.cm.hot)
ax.view_init(elev=30,azim=125)
```



```
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
xs = np.random.randint(30,40,100)
ys = np.random.randint(20,30,100)
zs = np.random.randint(10,20,100)
xs2 = np.random.randint(50,60,100)
ys2 = np.random.randint(30,40,100)
zs2 = np.random.randint(50,70,100)
xs3 = np.random.randint(10,30,100)
ys3 = np.random.randint(40,50,100)
zs3 = np.random.randint(40,50,100)
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(xs,ys,zs)
ax.scatter(xs2,ys2,zs2,c='r',marker='^')
ax.scatter(xs3,ys3,zs3,c='g',marker='*')
ax.set_xlabel('X Label') ax.set_ylabel('Y Label') ax.set_zlabel('Z Label')
```

```
import matplotlib.pyplot as plt
import numpy as np
fig = plt.figure()
ax = fig.add_axes([0.1,0.1,0.8,0.8])
inner_ax = fig.add_axes([0.6,0.6,0.25,0.25])
x1 = np.arange(10)
y1 = np.array([1,2,7,1,5,2,4,2,3,1])
x2 = np.arange(10)
y2 = np.array([1,3,4,5,4,5,2,6,4,3])
ax.plot(x1,y1)
inner_ax.plot(x2,y2)
```

> MultiPaneles

```
import matplotlib.pyplot as plt
import numpy as np
fig = plt.figure()
ax1 = fig.add_subplot(2, 2, 1)
ax2 = fig.add_subplot(2, 2, 2)
ax3 = fig.add_subplot(2, 2, 3)
```

```
import matplotlib.pyplot as plt
import numpy as np
fig = plt.figure()
ax1 = fig.add_subplot(2, 2, 1)
_ = ax1.hist(np.random.randn(100), bins=20, color='k', alpha=0.3)
ax2 = fig.add_subplot(2, 2, 2)
plt.plot(np.random.randn(50).cumsum(), 'k--')
ax3 = fig.add_subplot(2, 2, 3)
ax3.scatter(np.arange(30), np.arange(30) + 3 * np.random.randn(30))
```

> MultiPaneles

```
import matplotlib.pyplot as plt
import numpy as np
gs = plt.GridSpec(3,3)
fig = plt.figure(figsize=(6,6))
x1 = np.array([1,3,2,5])
y1 = np.array([4,3,7,2])
x2 = np.arange(5)
y2 = np.array([3,2,4,6,4])
s1 = fig.add_subplot(gs[1,:2])
s1.plot(x1,y1,'g')
s2 = fig.add_subplot(gs[0,:2])
s2.bar(x2,y2)
s3 = fig.add_subplot(gs[2,0])
s3.barh(x2,y2,color='g')
s4 = fig.add_subplot(gs[:2,2])
s4.plot(x2,y2,'k')
s5 = fig.add_subplot(gs[2,1:])
s5.plot(x1,y1,'b^',x2,y2,'yo')
```

Quiz

Considere un subconjunto de datos de estudiantes que tiene una mezcla inusual de información sobre (género, sueño, televisión, ejercicio, computadora, promedio) y (altura). Las unidades para computadora, televisión, dormir y hacer ejercicio son horas, la altura está en cm y el promedio se mide en una escala de 5.0.

Genero	TV	Computador	Dormir	AlturaE	AlturaMama	AlturaPapa	EjercicioE	Promedio
Female	13.0	10.0	3.50	66.0	66.0	71.0	10.0	4.000
Male	20.0	7.0	9.00	72.0	64.0	65.0	2.0	2.300
Male	15.0	15.0	6.00	68.0	62.0	74.0	3.0	2.600
Male	8.0	20.0	6.00	68.0	59.0	70.0	6.0	2.800

Crear un programa en Python que genere el .csv

References

- ★ Kernighan, Brian W., and Dennis M. Ritchie. The C Programming Language. Vol. 2. Englewood Cliffs: prentice-Hall, 1988.
- ★ Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. Operating System Concepts. Vol. 8. Wiley, 2013.
- ★ <https://planningtank.com/computer-applications/data-processing-cycle>
- ★ <https://www.talend.com/resources/what-is-data-processing/>
- ★ <https://peda.net/kenya/ass/subjects2/computer-studies/form-3/data-processing/dpc2>
- ★ https://www.academia.edu/38210518/What_is_Data_Processing
- ★ <https://www.studocu.com/en/document/polytechnic-university-of-the-philippines/information-and-communication-technology/lecture-notes/data-processing-lectures-in-data-processing/3167716/view>
- ★ <http://download.nos.org/srsec330/330L2.pdf>