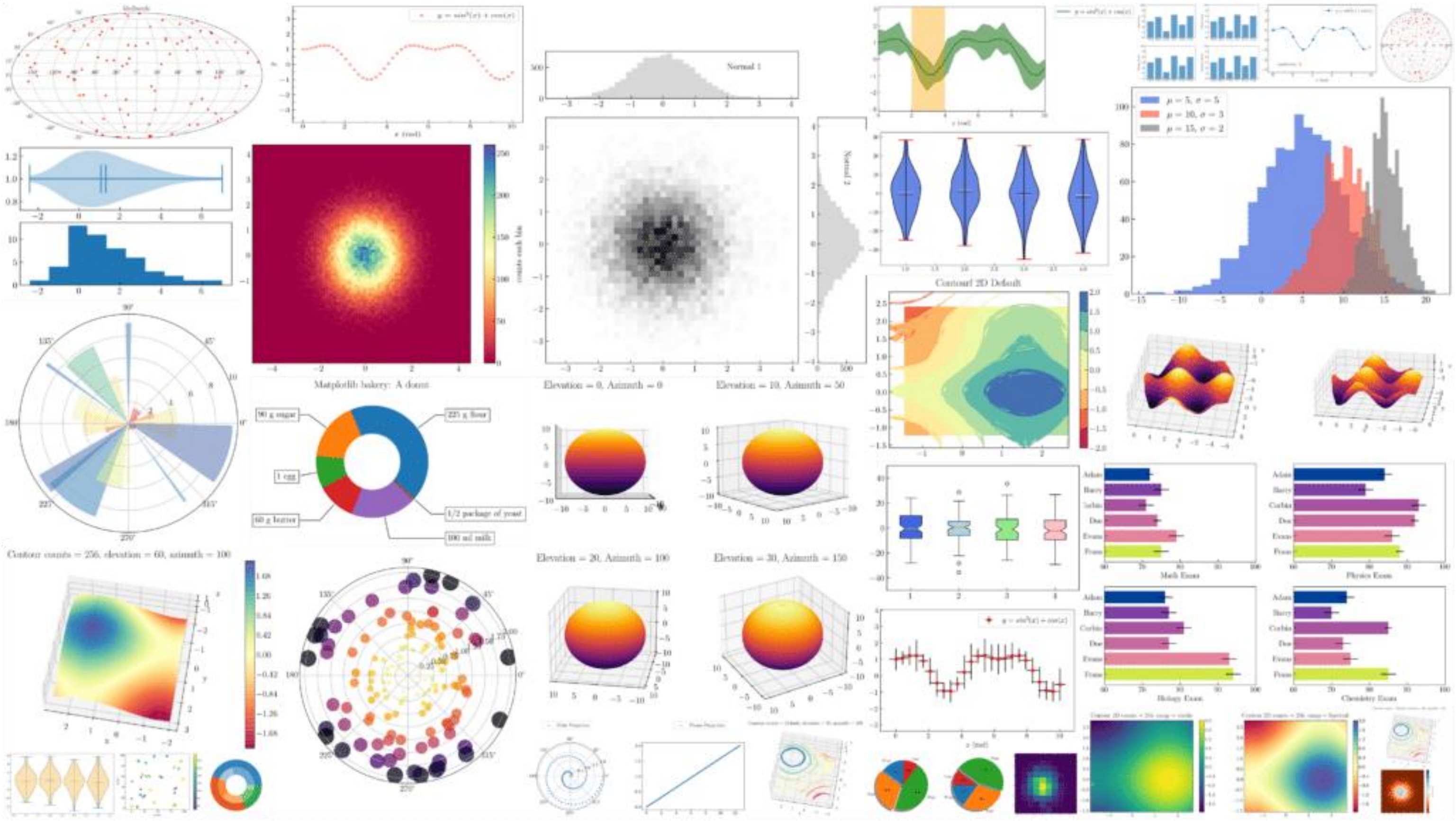
Programación para Analítica de Datos MODULO EN PYTHON – Graficas

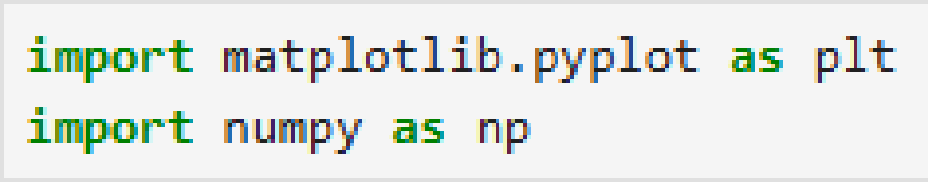
POR: Guillermo Andres De Mendoza Corrales

Libreria

Matplot: https://matplotlib.org/contents.html

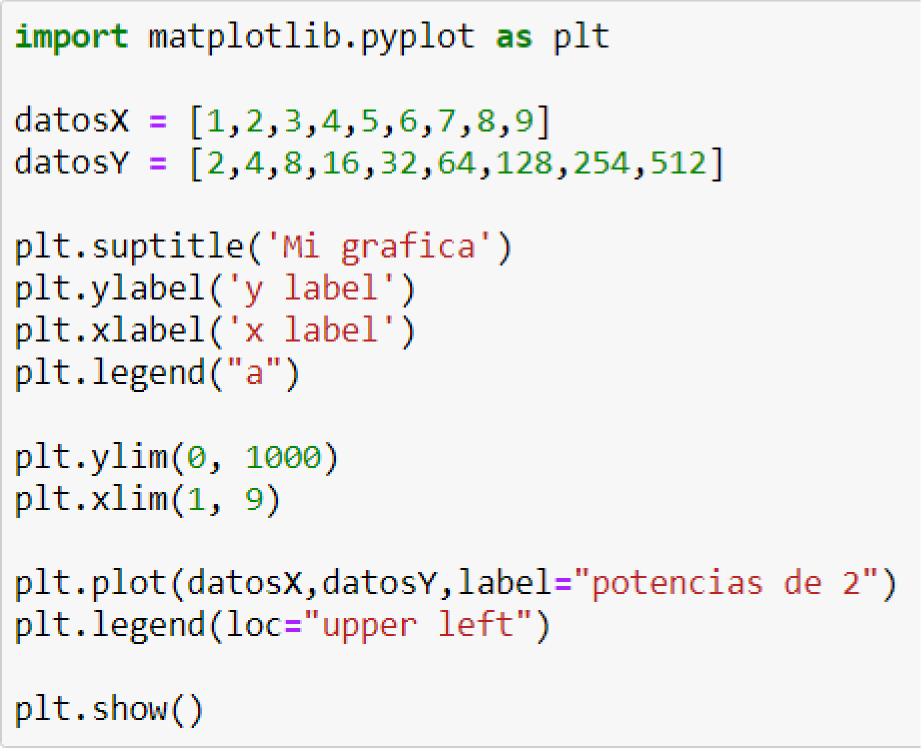


imports

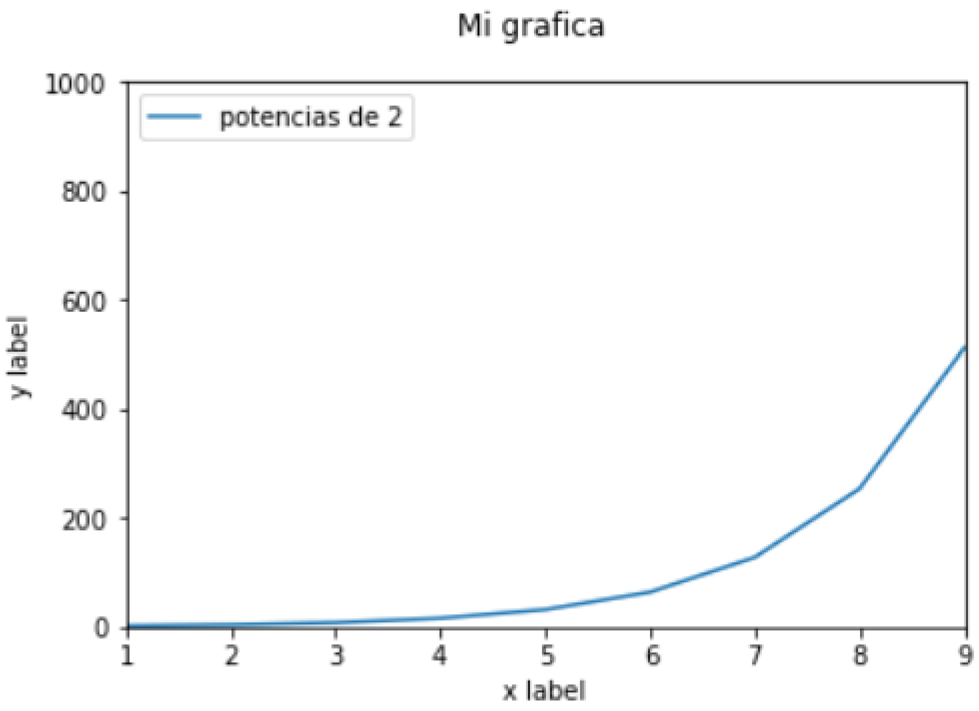


```python
import matplotlib.pyplot as plt
import numpy as np
```

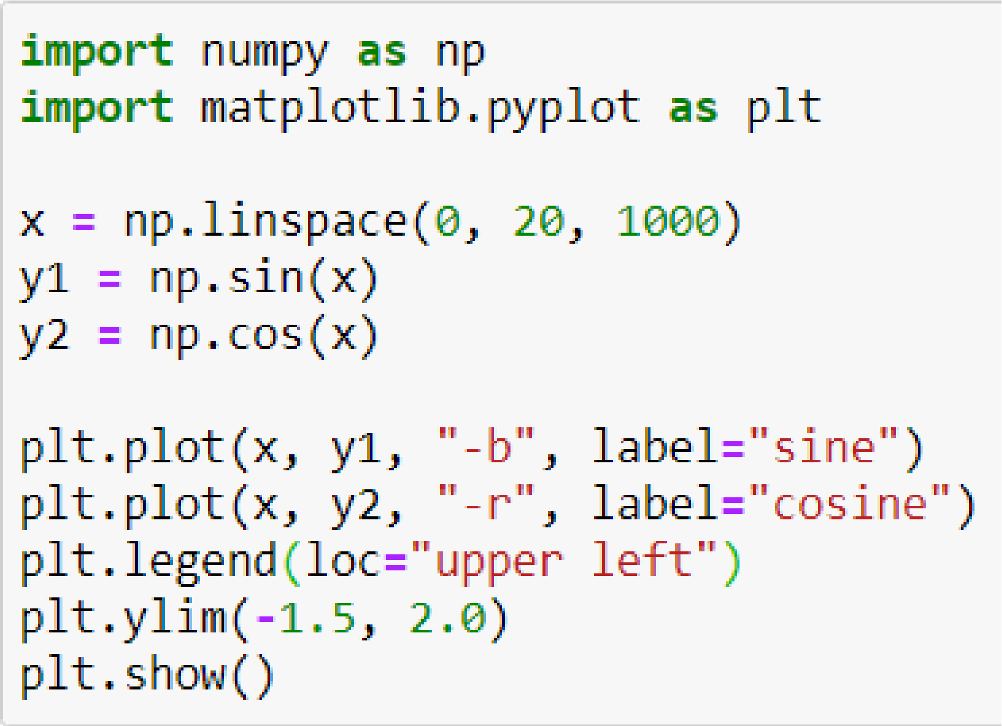
Graficas líneas



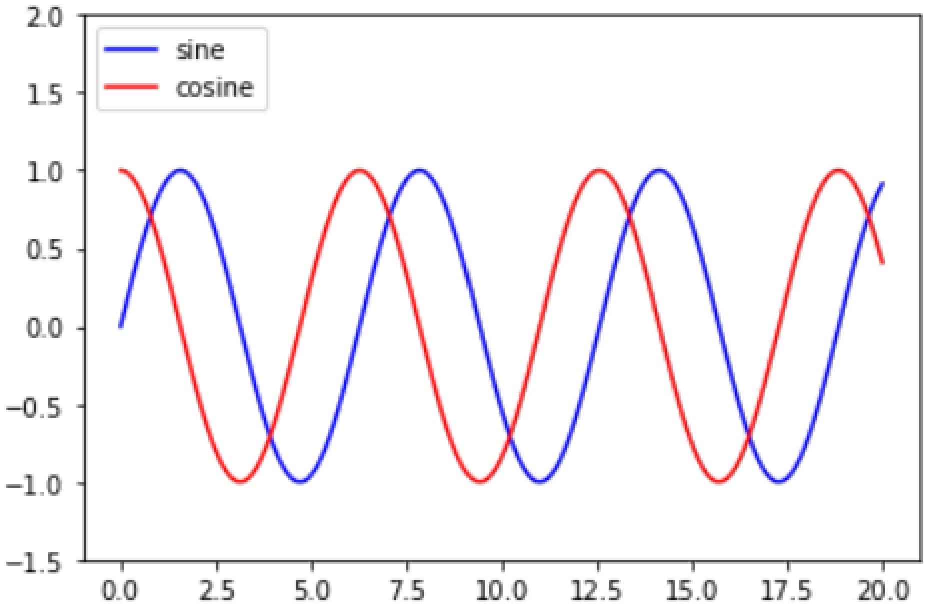
```
import matplotlib.pyplot as plt

datosX = [1,2,3,4,5,6,7,8,9]
datosY = [2,4,8,16,32,64,128,254,512]

plt.suptitle('Mi grafica')
plt.ylabel('y label')
plt.xlabel('x label')
plt.legend("a")

plt.ylim(0, 1000)
plt.xlim(1, 9)

plt.plot(datosX,datosY,label="potencias de 2")
plt.legend(loc="upper left")

plt.show()
```



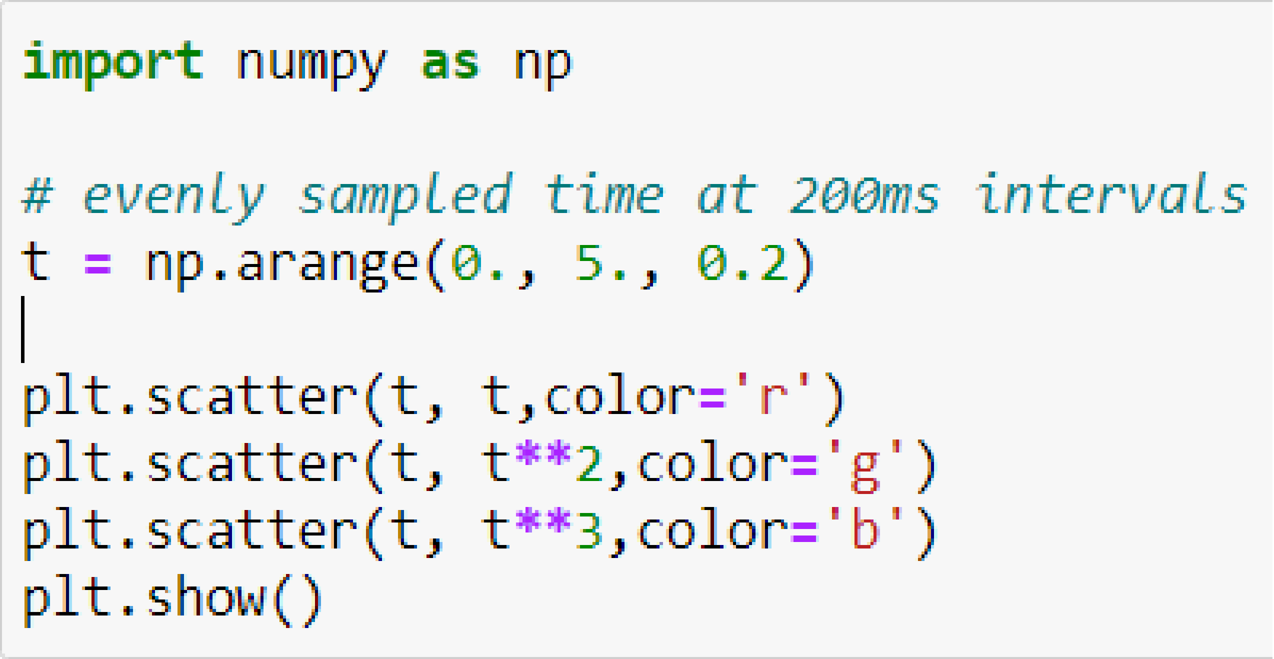
Graficas líneas multiples



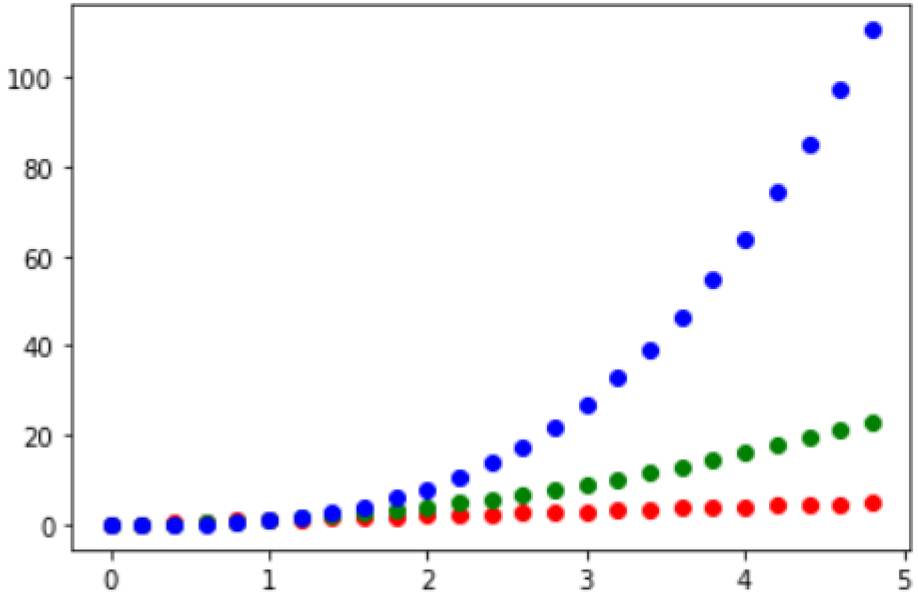
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 20, 1000)
y1 = np.sin(x)
y2 = np.cos(x)

plt.plot(x, y1, "-b", label="sine")
plt.plot(x, y2, "-r", label="cosine")
plt.legend(loc="upper left")
plt.ylim(-1.5, 2.0)
plt.show()
```



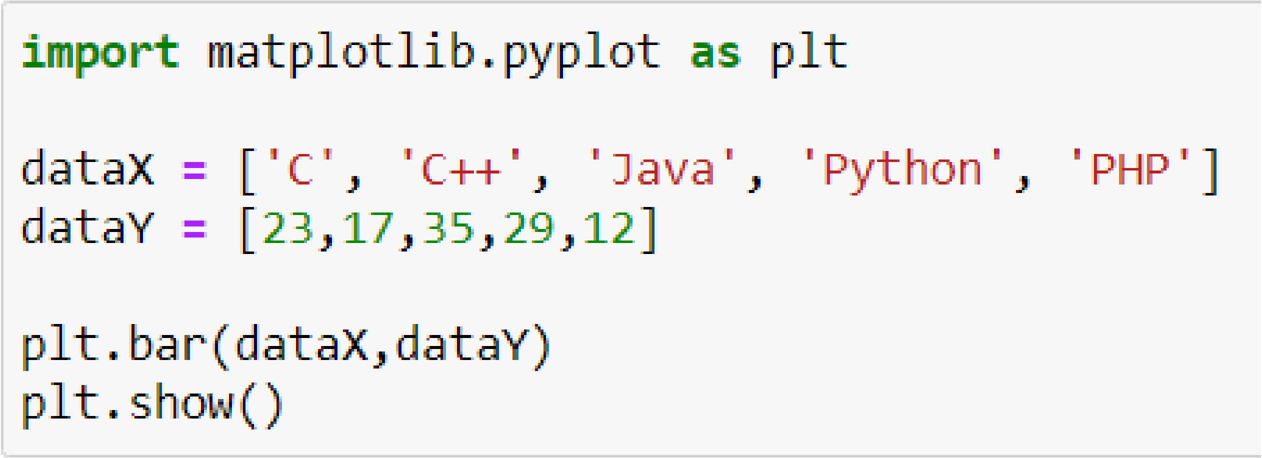
Graficas puntos



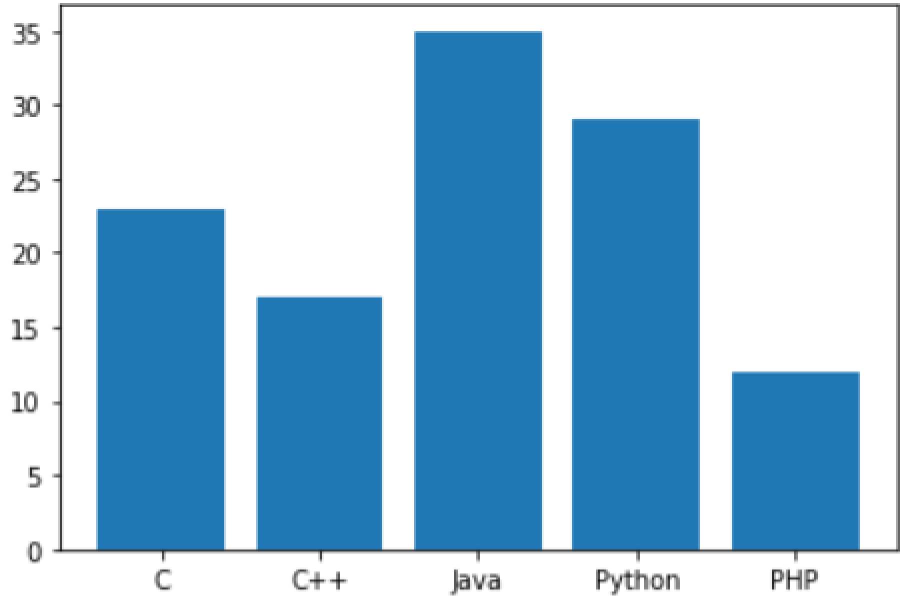
```python
import numpy as np

# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

plt.scatter(t, t,color='r')
plt.scatter(t, t**2,color='g')
plt.scatter(t, t**3,color='b')
plt.show()
```



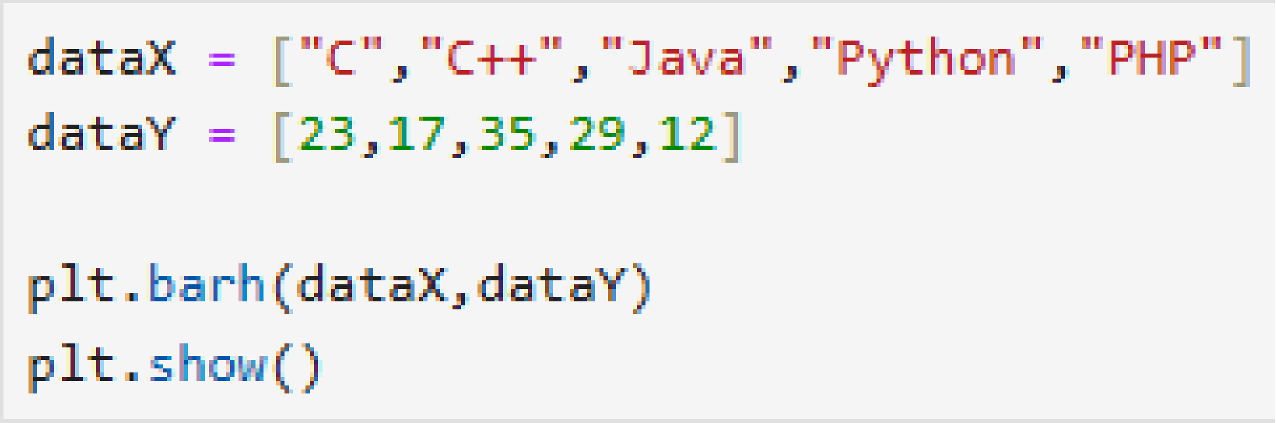
Barras verticales



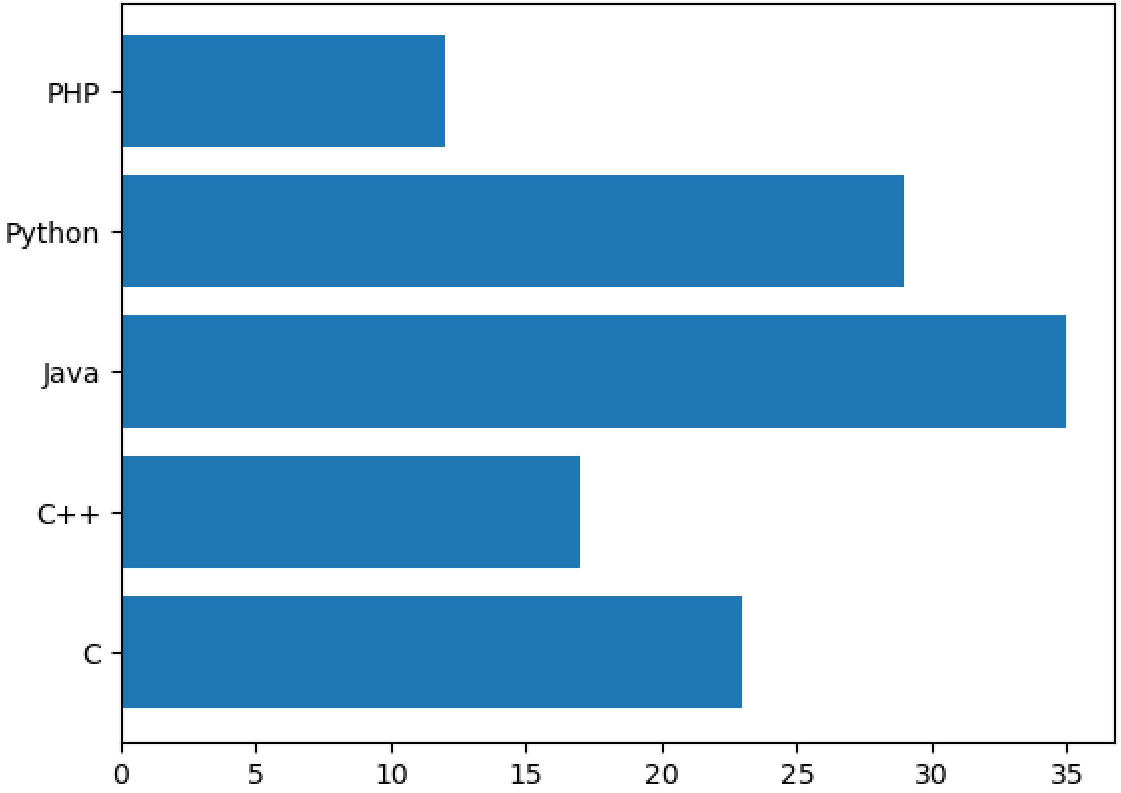
```python
import matplotlib.pyplot as plt

dataX = ['C', 'C++', 'Java', 'Python', 'PHP']
dataY = [23,17,35,29,12]

plt.bar(dataX,dataY)
plt.show()
```



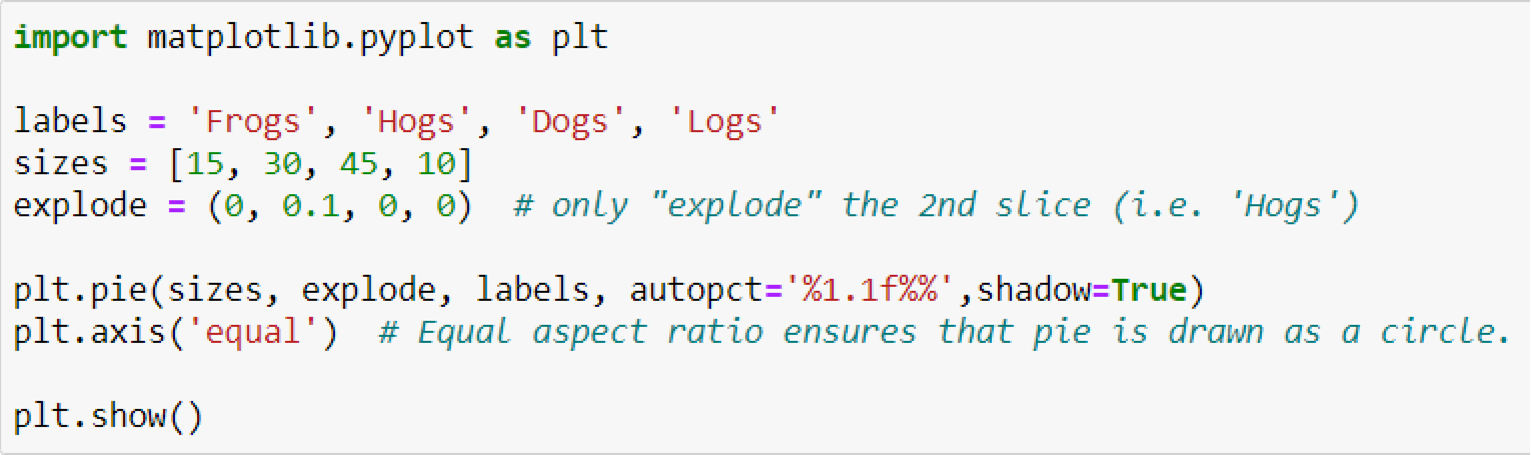
Barras horizontales



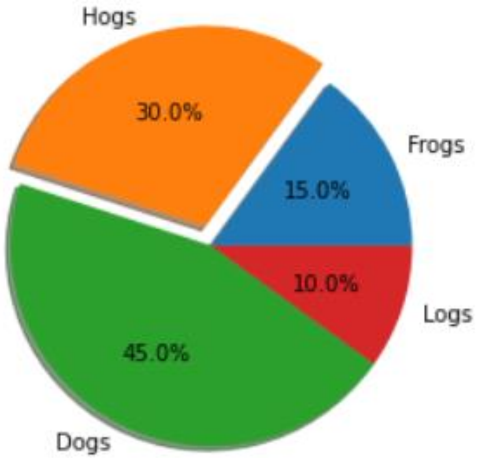
```
dataX = ["C","C++","Java","Python","PHP"]
dataY = [23,17,35,29,12]

plt.barh(dataX,dataY)
plt.show()
```

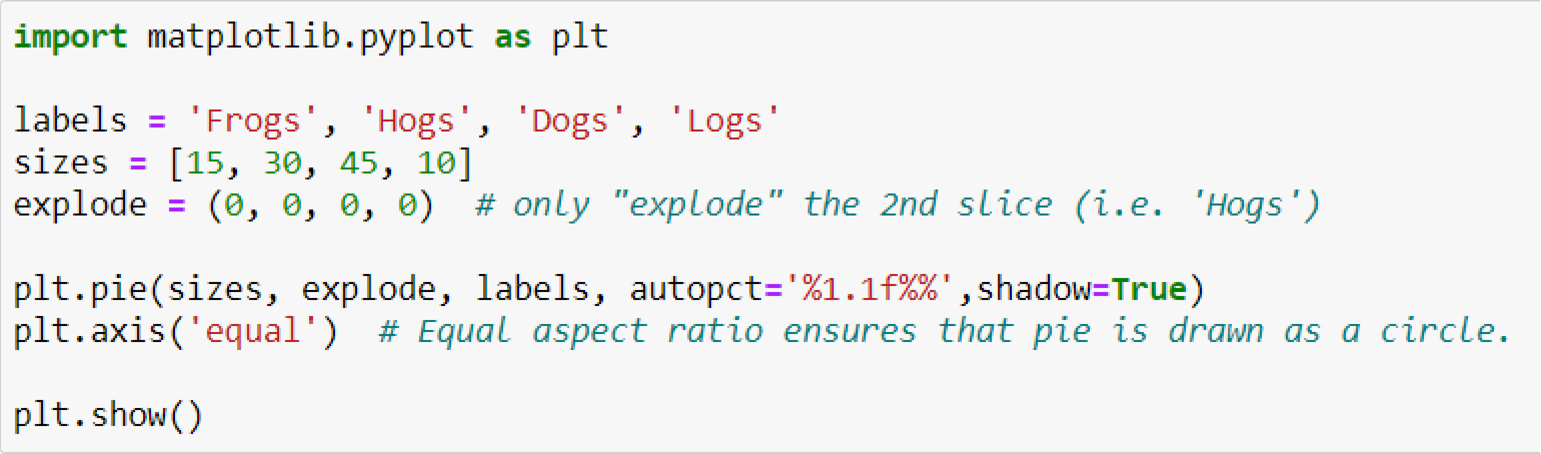


Graficas pastel

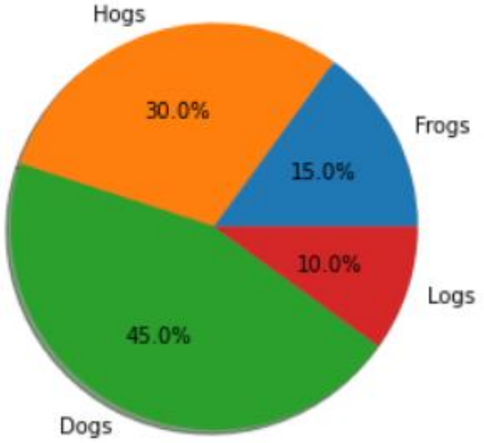


```
import matplotlib.pyplot as plt

labels = 'Frogs', 'Hogs', 'Dogs', 'Logs'
sizes = [15, 30, 45, 10]
explode = (0, 0.1, 0, 0)  # only "explode" the 2nd slice (i.e. 'Hogs')

plt.pie(sizes, explode, labels, autopct='%1.1f%%',shadow=True)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```

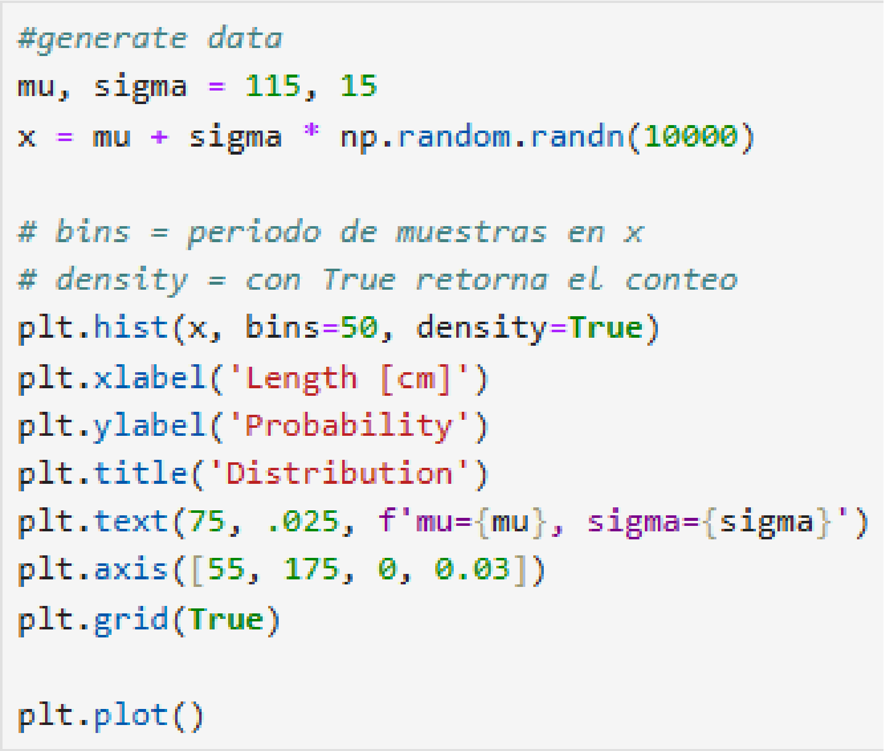




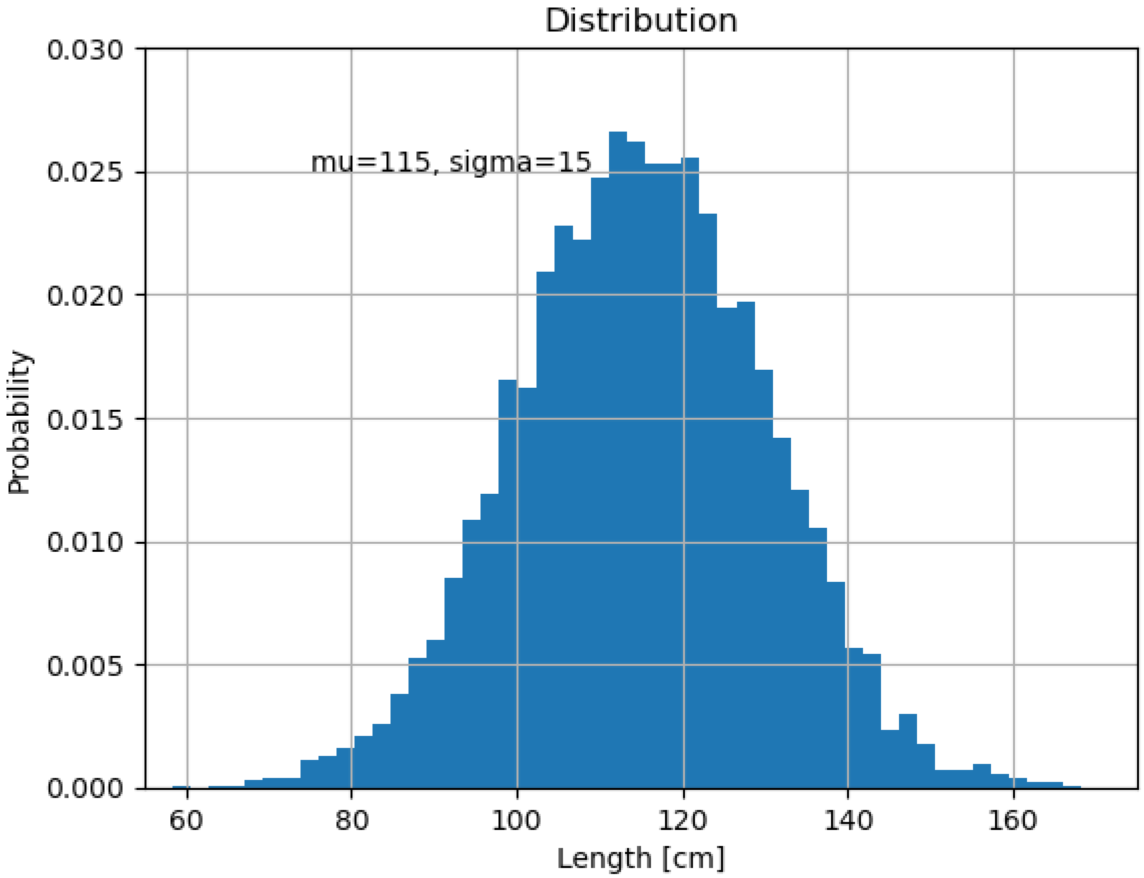
```
import matplotlib.pyplot as plt

labels = 'Frogs', 'Hogs', 'Dogs', 'Logs'
sizes = [15, 30, 45, 10]
explode = (0, 0, 0, 0)  # only "explode" the 2nd slice (i.e. 'Hogs')

plt.pie(sizes, explode, labels, autopct='%1.1f%%',shadow=True)
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```



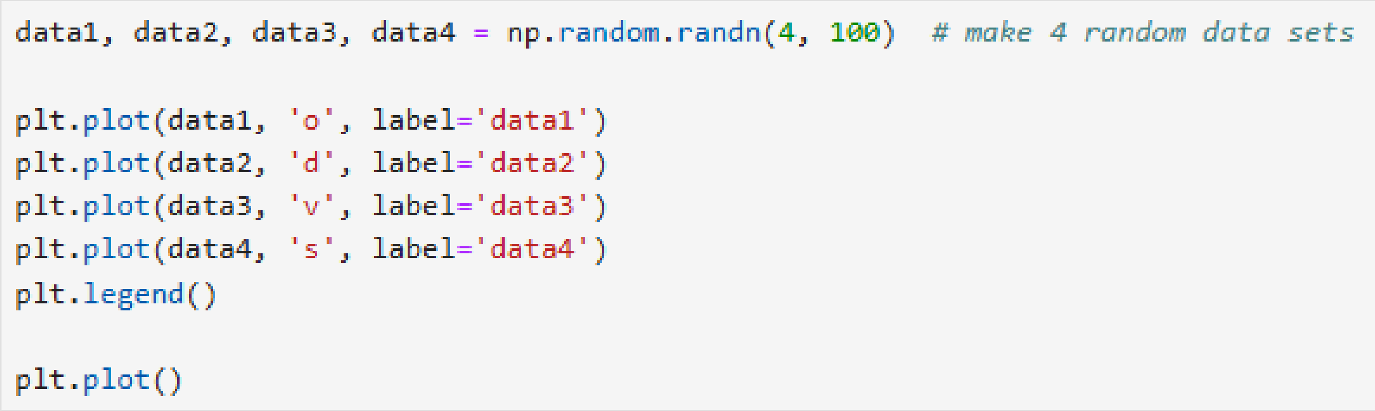
Histograma



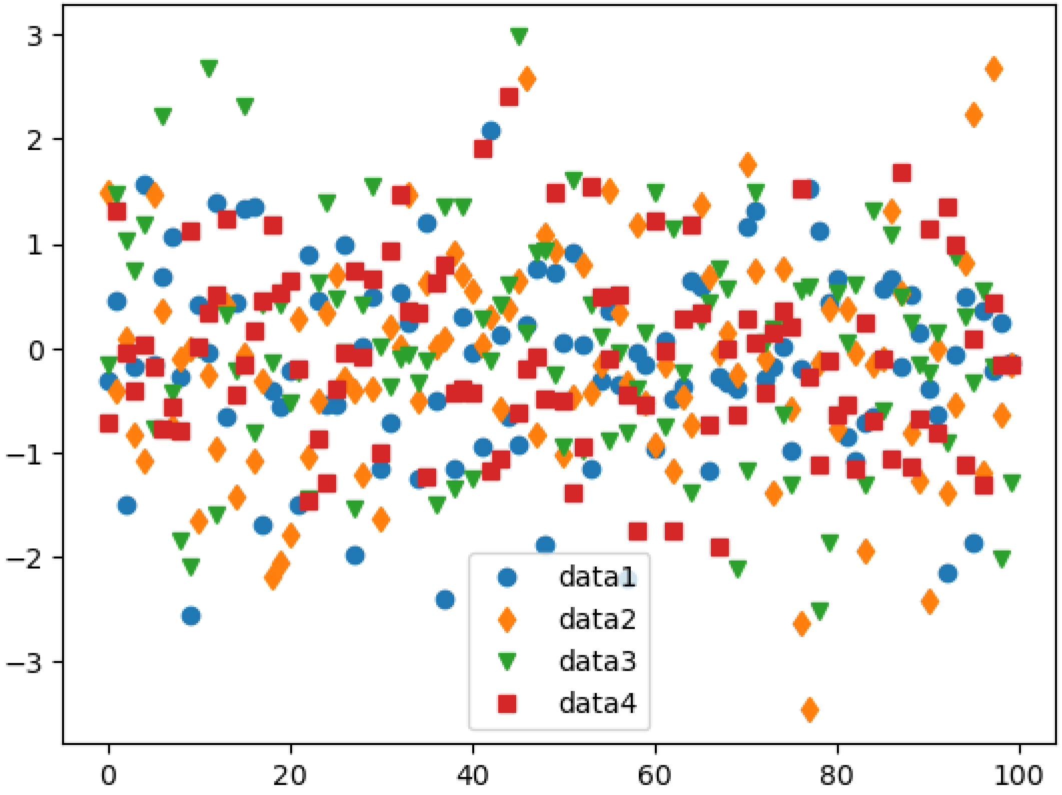
```
#generate data
mu, sigma = 115, 15
x = mu + sigma * np.random.randn(10000)

# bins = periodo de muestras en x
# density = con True retorna el conteo
plt.hist(x, bins=50, density=True)
plt.xlabel('Length [cm]')
plt.ylabel('Probability')
plt.title('Distribution')
plt.text(75, .025, f'mu={mu}, sigma={sigma}')
plt.axis([55, 175, 0, 0.03])
plt.grid(True)

plt.plot()
```



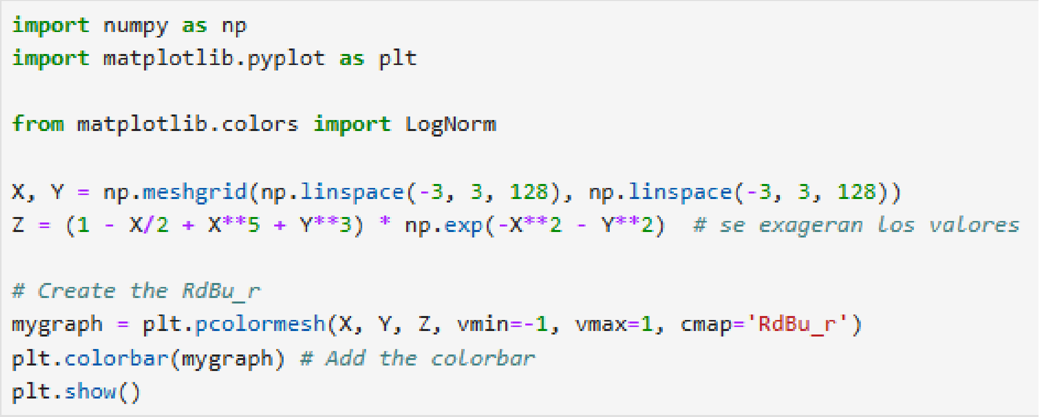
Simbolos



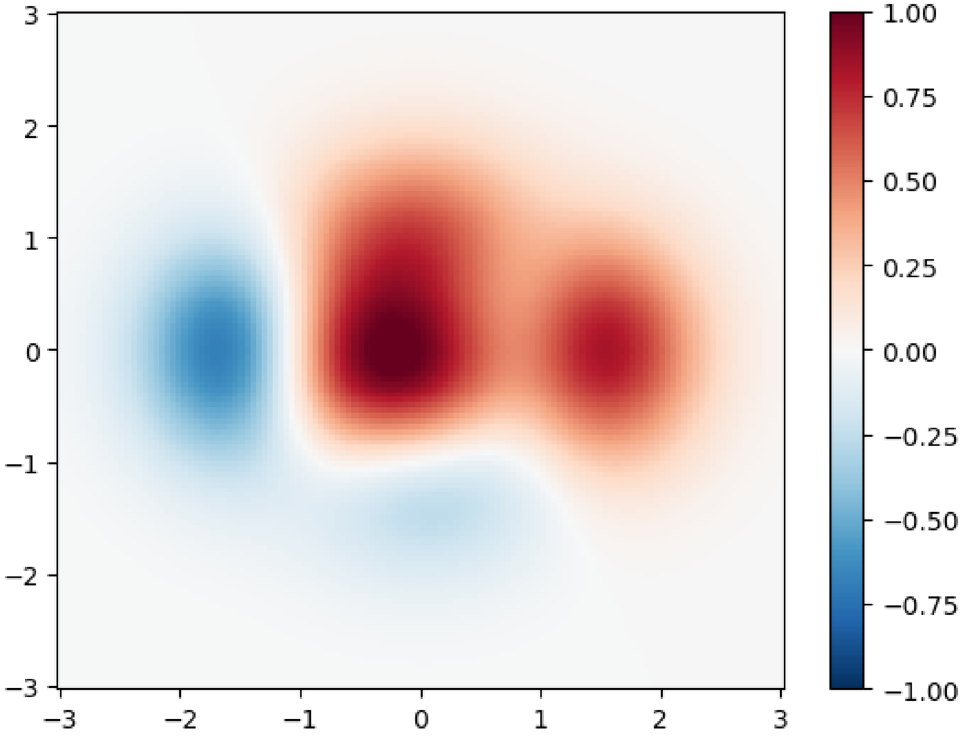
```
data1, data2, data3, data4 = np.random.randn(4, 100)  # make 4 random data sets

plt.plot(data1, 'o', label='data1')
plt.plot(data2, 'd', label='data2')
plt.plot(data3, 'v', label='data3')
plt.plot(data4, 's', label='data4')
plt.legend()

plt.plot()
```



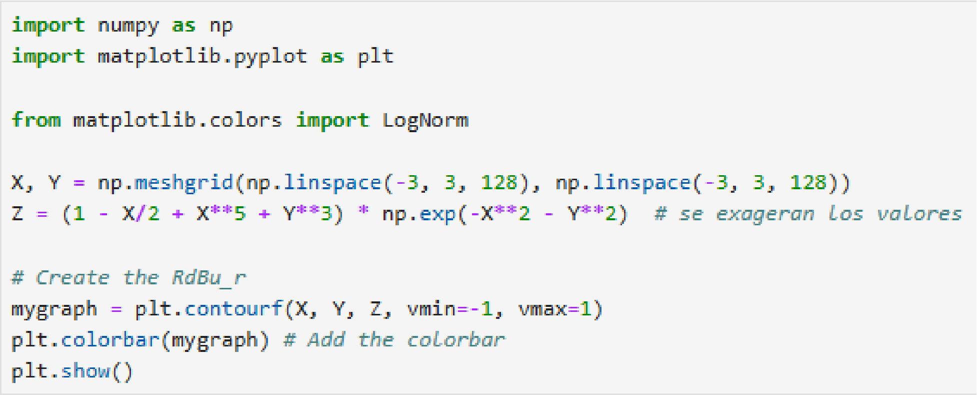
Grid: pcolormesh



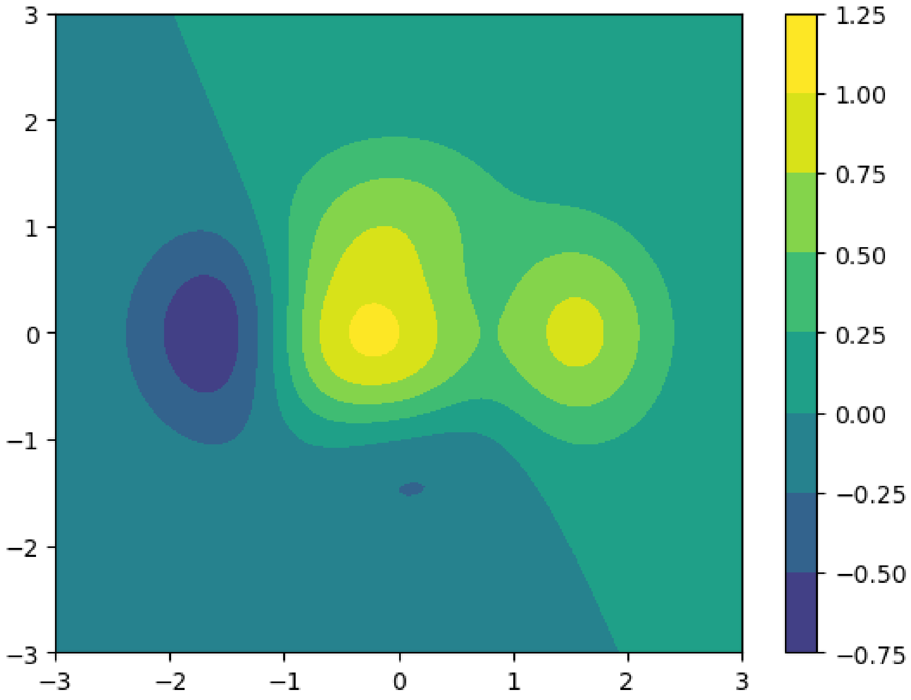
```python
import numpy as np
import matplotlib.pyplot as plt

from matplotlib.colors import LogNorm

X, Y = np.meshgrid(np.linspace(-3, 3, 128), np.linspace(-3, 3, 128))
Z = (1 - X/2 + X**5 + Y**3) * np.exp(-X**2 - Y**2)  # se exageran los valores

# Create the RdBu_r
mygraph = plt.pcolormesh(X, Y, Z, vmin=-1, vmax=1, cmap='RdBu_r')
plt.colorbar(mygraph) # Add the colorbar
plt.show()
```



Grid: pcolormesh



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

from matplotlib.colors import LogNorm

X, Y = np.meshgrid(np.linspace(-3, 3, 128), np.linspace(-3, 3, 128))
Z = (1 - X/2 + X**5 + Y**3) * np.exp(-X**2 - Y**2)  # se exageran los valores

# Create the RdBu_r
mygraph = plt.contourf(X, Y, Z, vmin=-1, vmax=1)
plt.colorbar(mygraph) # Add the colorbar
plt.show()
```



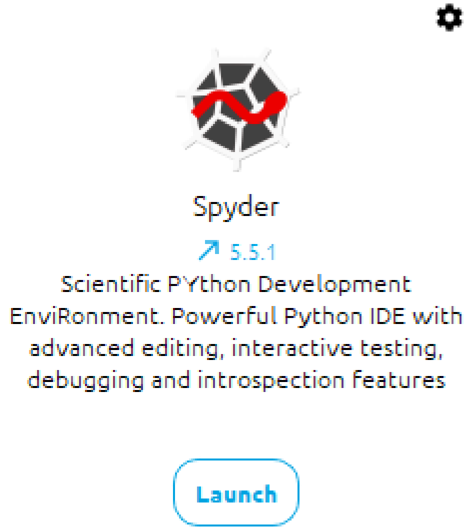
3D in Spider

Spider

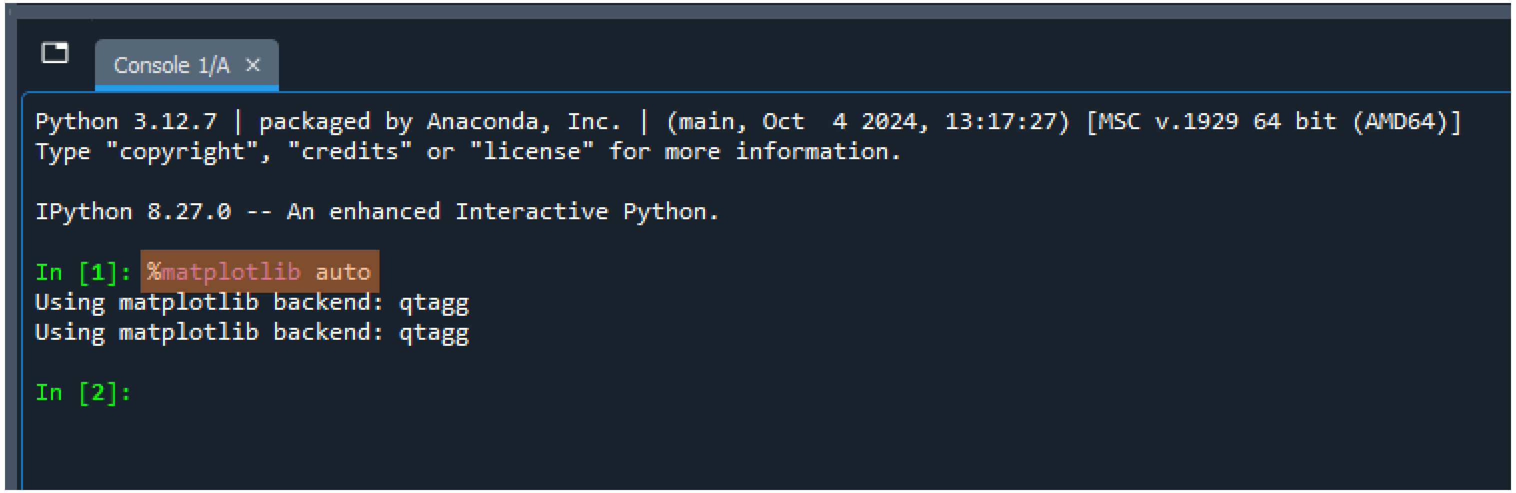
Permite a los plots ser interactivos

SPIDER COMMAND:

%matplotlib auto



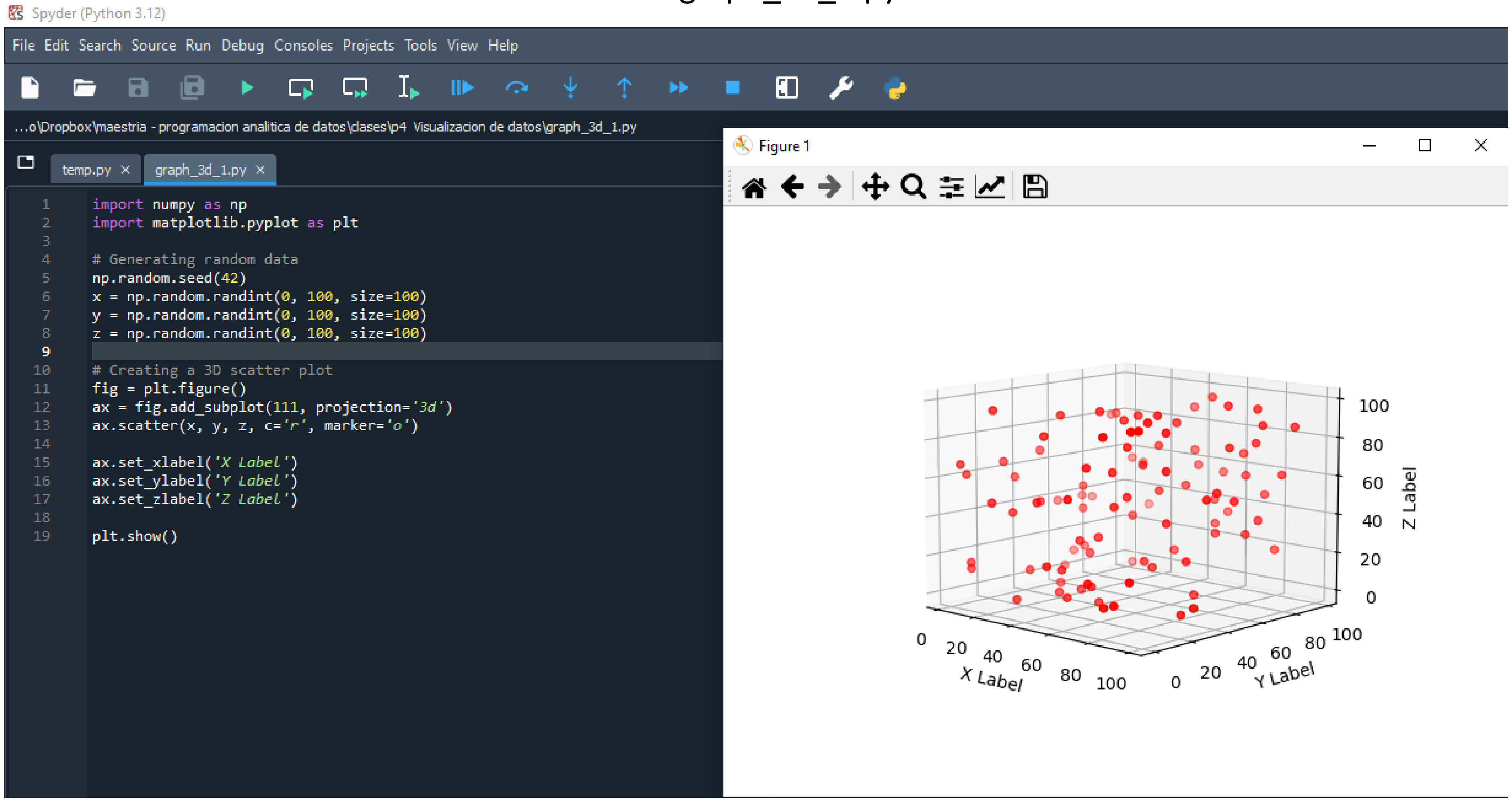
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features



```
Python 3.12.7 | packaged by Anaconda, Inc. | (main, Oct  4 2024, 13:17:27) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.27.0 -- An enhanced Interactive Python.

In [1]: %matplotlib auto
Using matplotlib backend: qtagg
Using matplotlib backend: qtagg

In [2]:
```

3D Graphs

File: graph_3d_1.py

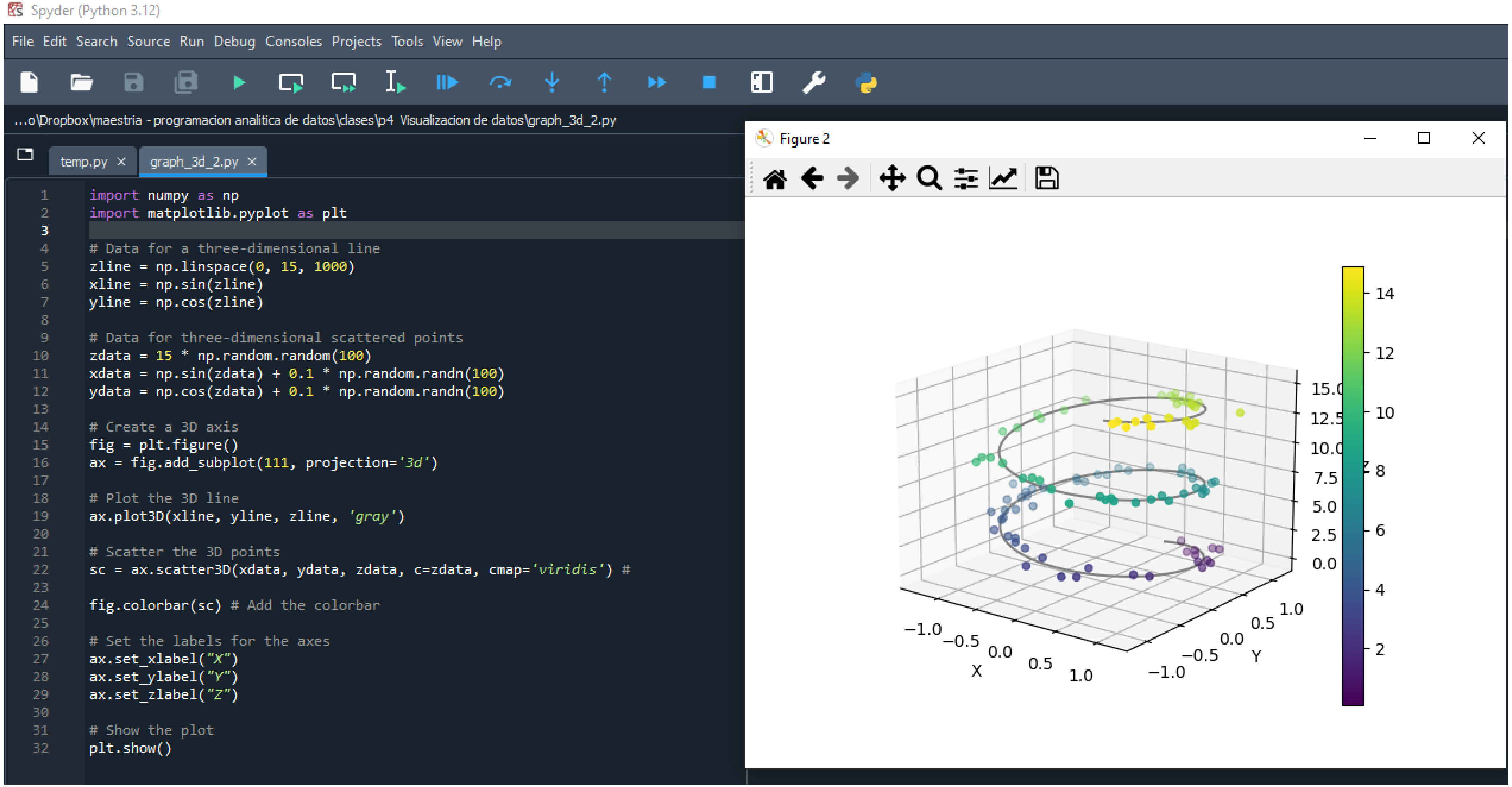


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...o\Dropbox\maestria - programacion analitica de datos\clases\p4 Visualizacion de datos\graph_3d_1.py
temp.py
graph_3d_1.py
Figure 1
0
20
40
60
80
100
X Label
0
20
40
60
80
100
Y Label
0
20
40
60
80
100
Z Label

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

# Generating random data
np.random.seed(42)
x = np.random.randint(0, 100, size=100)
y = np.random.randint(0, 100, size=100)
z = np.random.randint(0, 100, size=100)

# Creating a 3D scatter plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(x, y, z, c='r', marker='o')

ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')

plt.show()
```

3D Graphs

File: graph_3d_2.py



Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
...o\Dropbox\maestria - programacion analitica de datos\clases\p4 Visualizacion de datos\graph_3d_2.py
temp.py
graph_3d_2.py
Figure 2
14
12
10
8
6
4
2
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Z
−1.0
−0.5
0.0
0.5
1.0
X
1.0
0.5
0.0
−0.5
−1.0
Y

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

# Data for a three-dimensional line
zline = np.linspace(0, 15, 1000)
xline = np.sin(zline)
yline = np.cos(zline)

# Data for three-dimensional scattered points
zdata = 15 * np.random.random(100)
xdata = np.sin(zdata) + 0.1 * np.random.randn(100)
ydata = np.cos(zdata) + 0.1 * np.random.randn(100)

# Create a 3D axis
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Plot the 3D line
ax.plot3D(xline, yline, zline, 'gray')

# Scatter the 3D points
sc = ax.scatter3D(xdata, ydata, zdata, c=zdata, cmap='viridis') #

fig.colorbar(sc) # Add the colorbar

# Set the labels for the axes
ax.set_xlabel(“X”)
ax.set_ylabel(“Y”)
ax.set_zlabel(“Z”)

# Show the plot
plt.show()
```