

Matplotlib

January 23, 2018

1 Usando Matplotlib

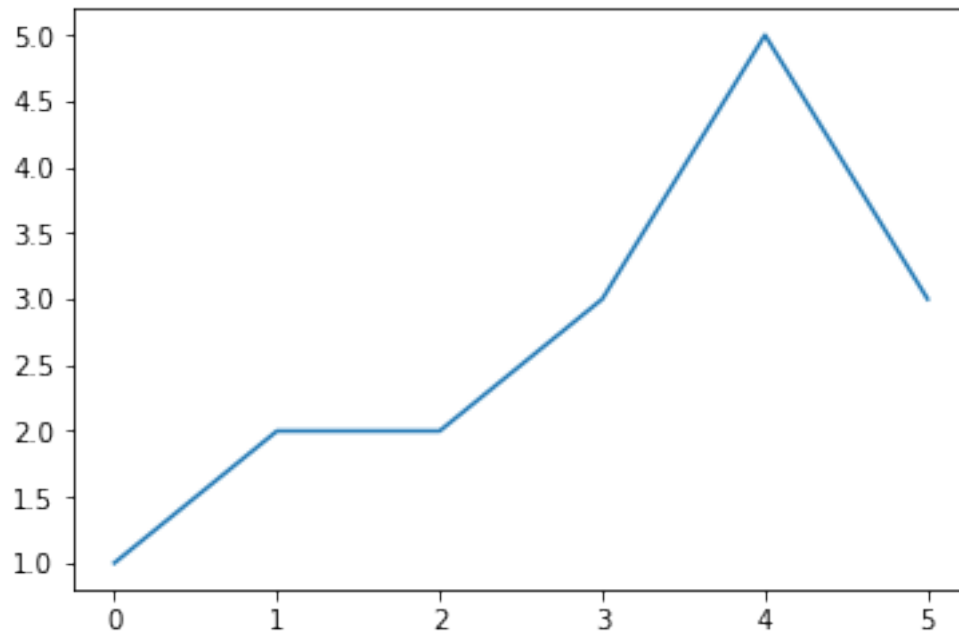
Matplotlib es una librería que produce figuras con calidad de publicación de forma muy sencilla.

```
In [4]: %matplotlib inline
```

1.1 plot y show

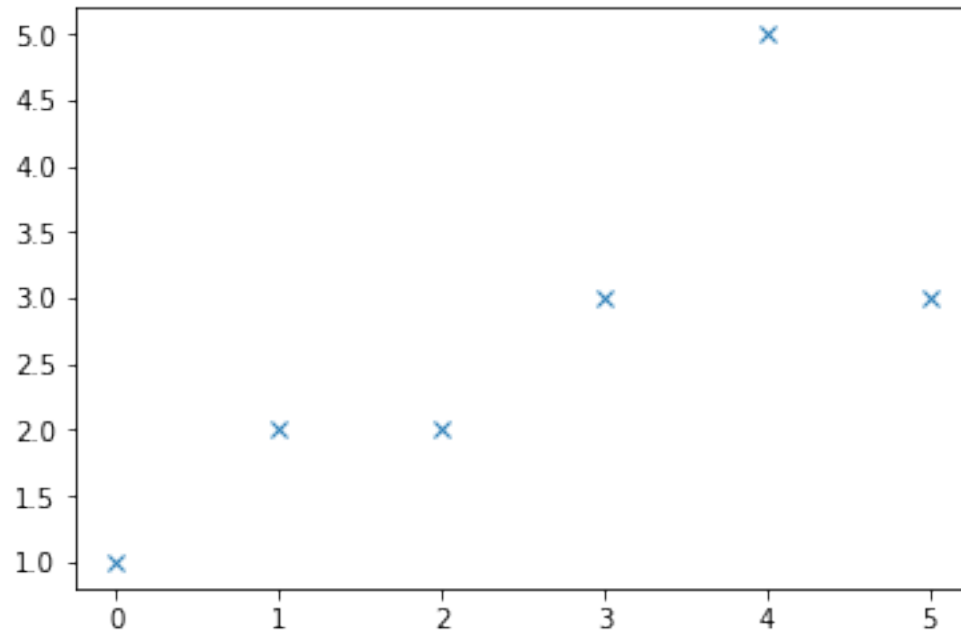
```
In [81]: from matplotlib.pyplot import *  
         #import matplotlib.pyplot as plt
```

```
In [29]: plot([1,2,2,3,5,3])  
         show()  
         #plt.plot([1,2,2,3,5,3])  
         #plt.show()
```

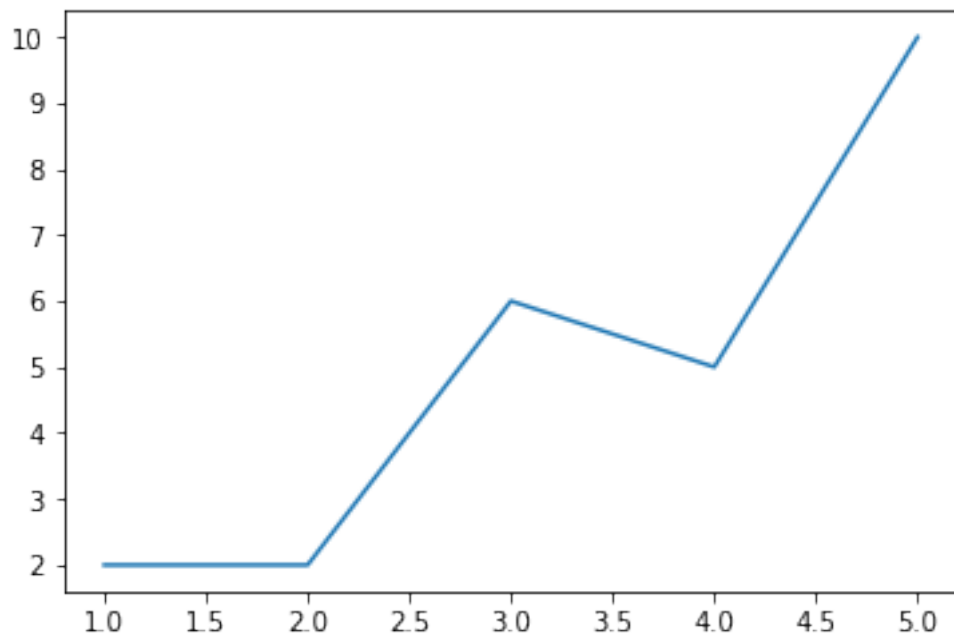


1.1.1 Graficar puntos

```
In [36]: plot([1,2,2,3,5,3], 'x')  
show()
```



```
In [46]: plot([1,2,3,4,5],[2,2,6,5,10])  
show()
```

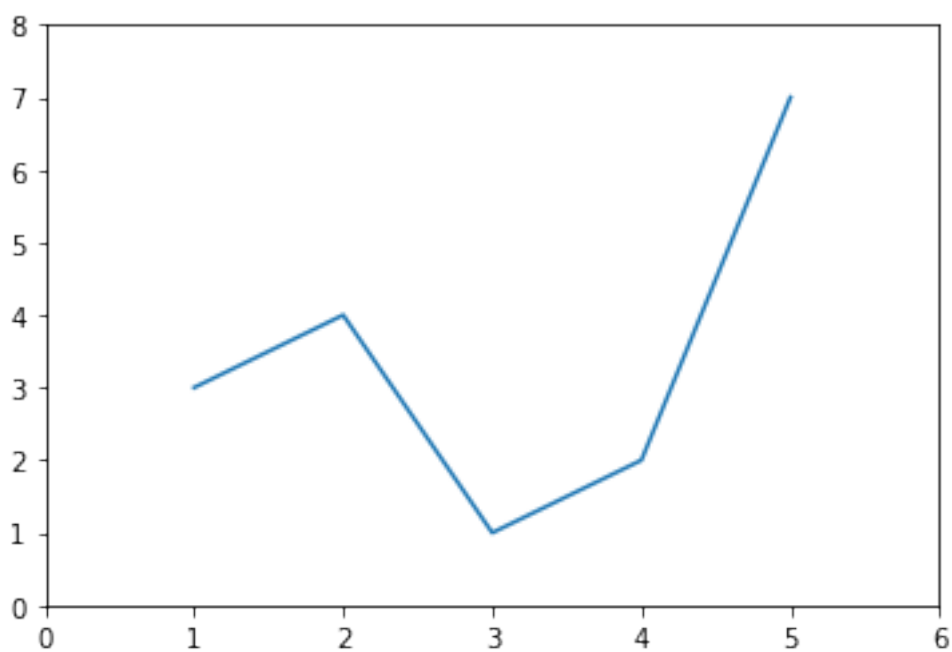


1.1.2 Límites

xlim(lim inferior x, lim superior x)

ylim(lim inferior y, lim superior y)

```
In [58]: plot([1,2,3,4,5],[3,4,1,2,7])  
        xlim(0,6)  
        ylim(0,8)  
        show()
```



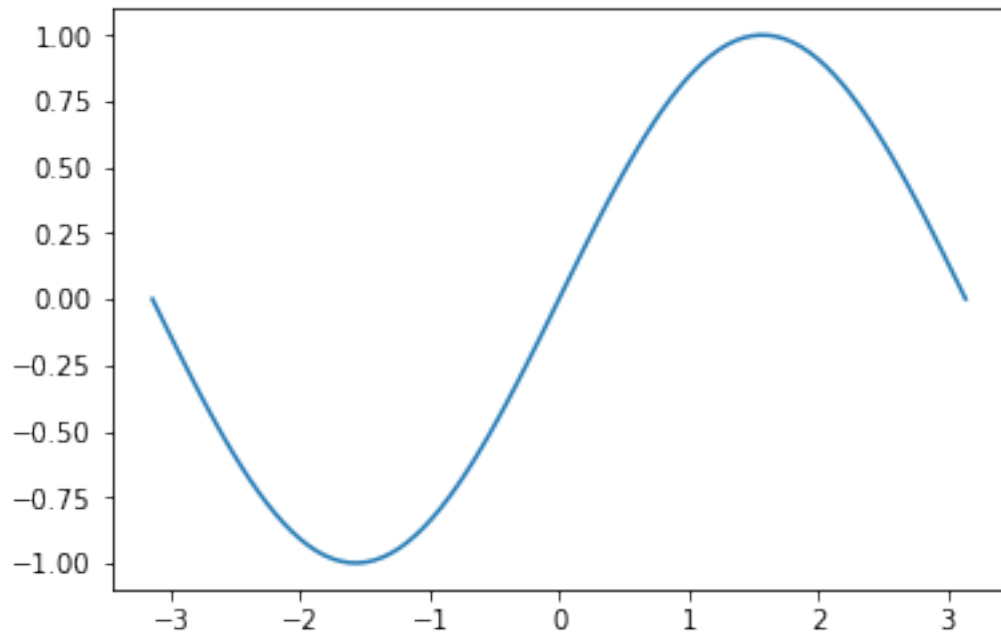
1.1.3 linspace

linspace(lim inferior, lim superior, cantidad)

```
In [80]: from numpy import *
```

```
In [78]: x = linspace(-pi, pi, 100)  
        y = sin(x)
```

```
        plot(x,y)  
        show()
```



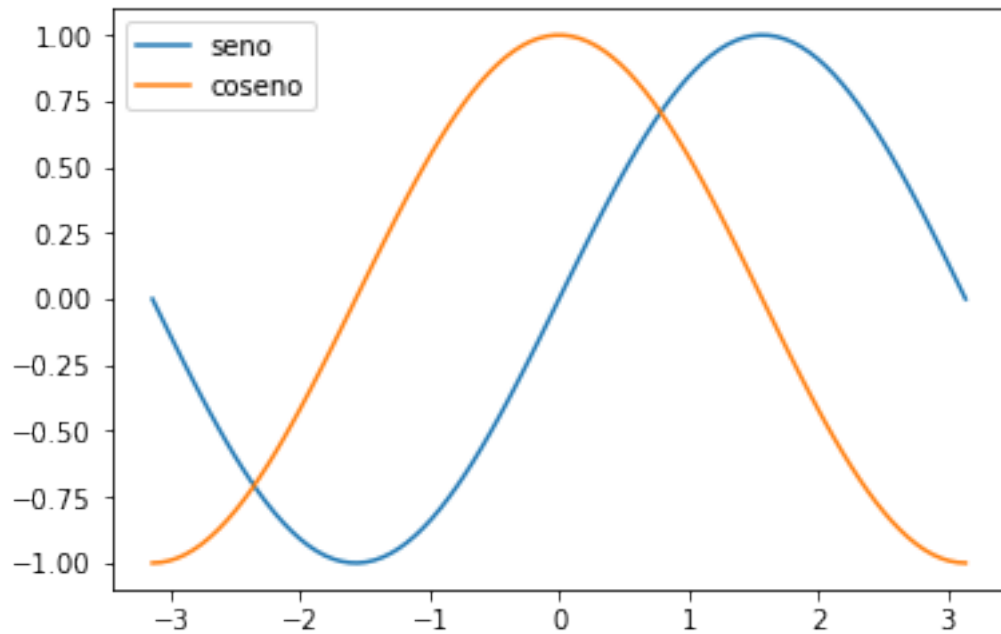
1.1.4 label y legend

```
In [77]: x = linspace(-pi, pi, 100)
         y = sin(x)
         z = cos(x)

         plot(x,y, label='seno')
         plot(x,z, label='coseno')

         legend(loc = 'upper left')

         show()
```



1.1.5 Figuras múltiples y color

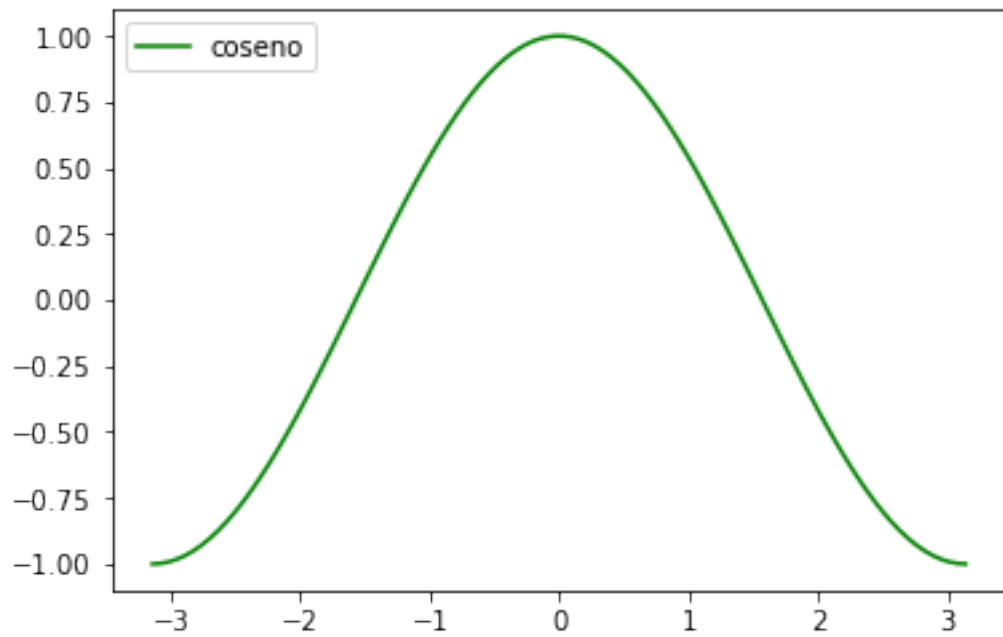
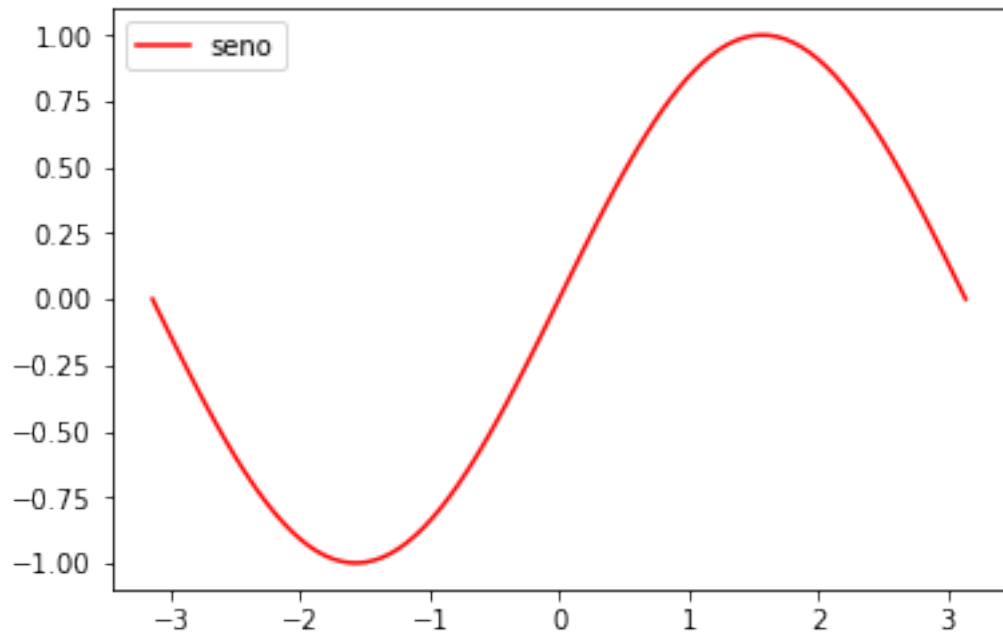
```
In [12]: x = linspace(-pi, pi, 100)
         y = sin(x)
         z = cos(x)

         a = figure(1)

         plot(x,y, label='seno', color='r')
         legend(loc = 'upper left')

         b = figure(2)
         plot(x,z, label='coseno', color='g')
         legend(loc = 'upper left')

         show()
```



1.1.6 title, xlabel y ylabel

```
In [88]: x = linspace(-pi, pi, 100)
         y = sin(x)
```

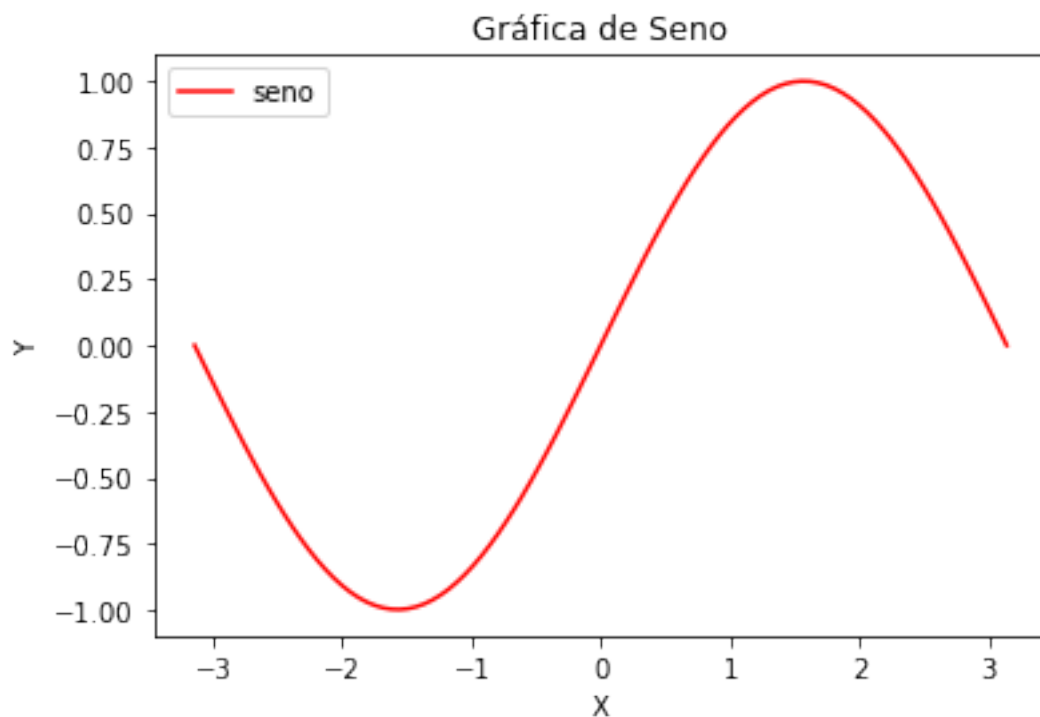
```

plot(x,y, label='seno', color='r')
xlabel('X')
ylabel('Y')
title("Gráfica de Seno")

legend(loc = 'upper left')

show()

```



1.1.7 Subgráficas

subplot(renglones, columnas, numero de gráfica)

```

In [96]: x = linspace(-pi, pi, 100)
         y = sin(x)
         z = cos(x)

         subplot(2,1,1)
         plot(x,y, label='seno', color='r')
         legend(loc = 'upper left')

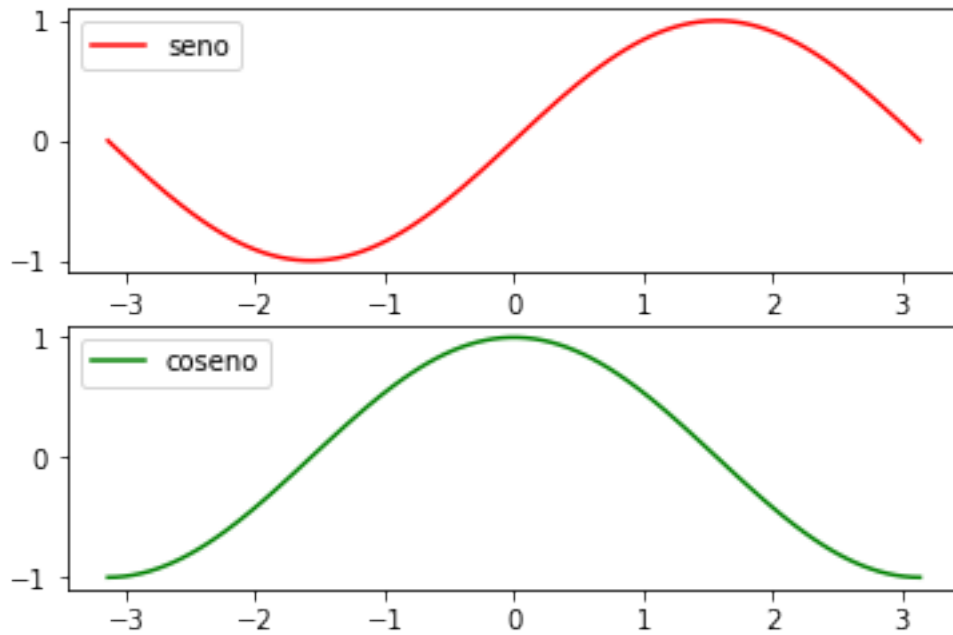
```

```

subplot(2,1,2)
plot(x,z, label='coseno', color='g')
legend(loc = 'upper left')

show()

```

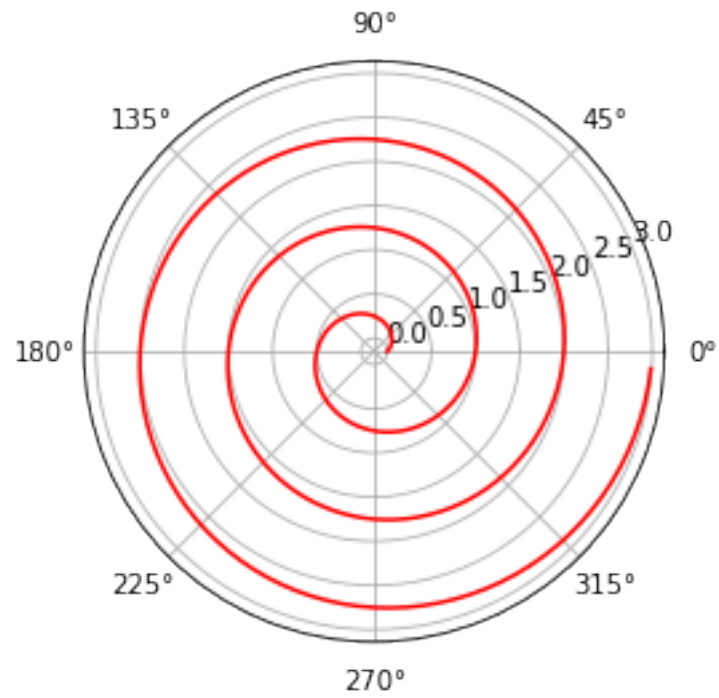


1.1.8 Gráfica polar

```

In [99]: angulo = arange(0, 3, 0.01) #Se generan valores de angulo
r = 2*pi*angulo # Se calcula r para cada valor de angulo
subplot(111, polar = True) # Se crea la subgráfica polar
plot(r, angulo, color = 'r')
show()

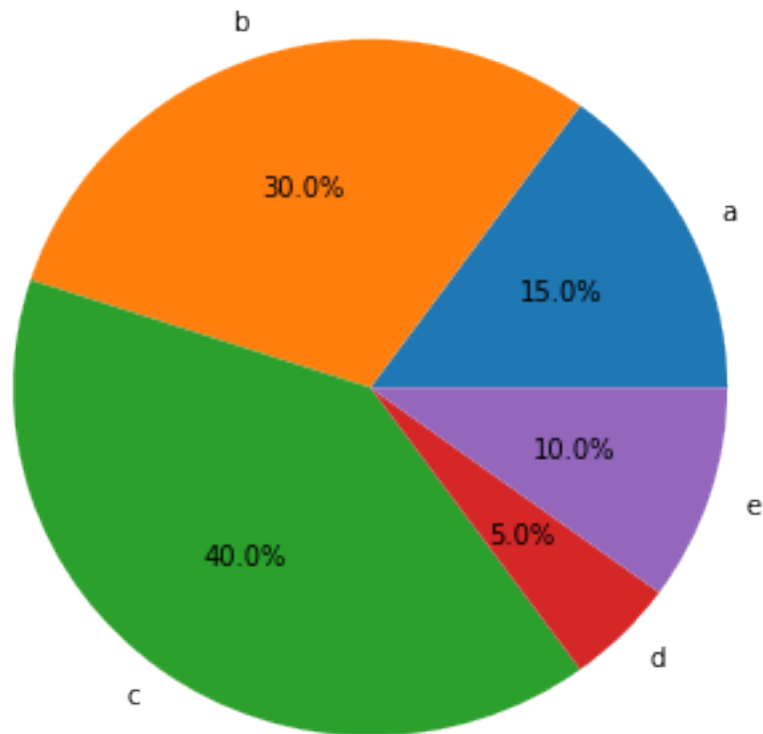
```

1.1.9 Gráfica de pastel

```
In [105]: figure(1, figsize = (6,6))
          fracs = [15, 30, 40, 5, 10]
          pie(frac, labels = ["a","b","c","d","e"], autopct='%1.1f%%')
          title('Gráfica de pastel')
          show()
```

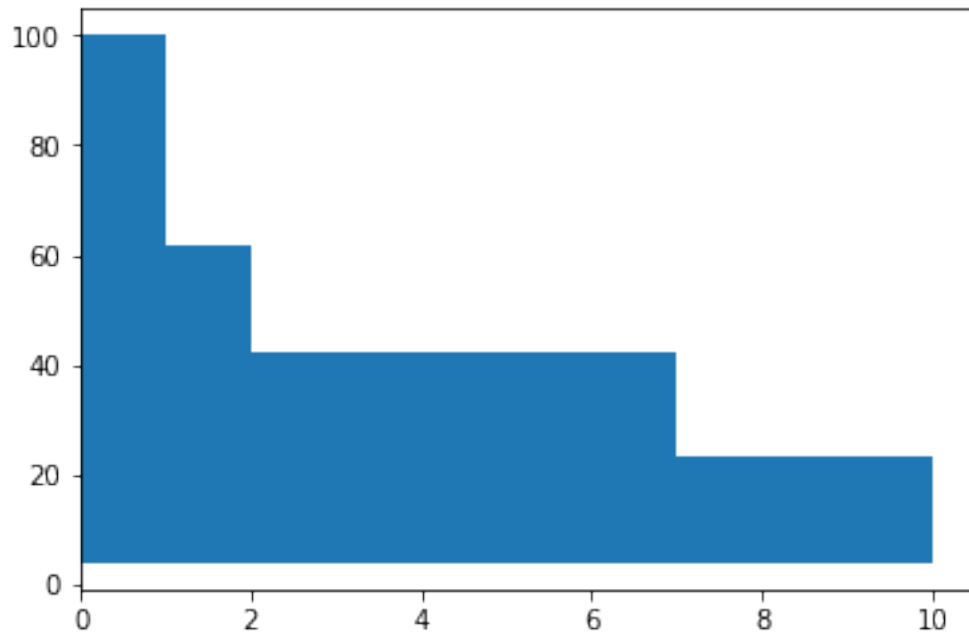
Gráfica de pastel



1.1.10 Gráfica de histograma

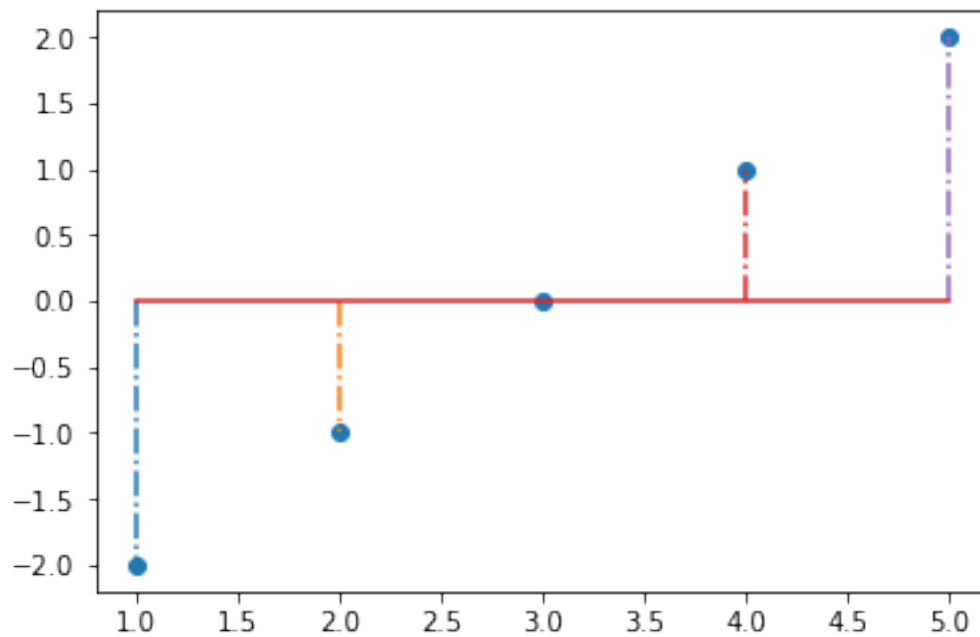
```
In [79]: import matplotlib.mlab as mlab
```

```
x = [21,22,23,4,5,6,77,8,9,10,31,32,33,34,35,36,37,18,49,50,100]  
numero = 5  
plt.hist(x, numero, orientation="horizontal")  
plt.show()
```



1.1.11 Gráfica de stem o de puntos

In [52]: `stem([1,2,3,4,5], [-2,-1,0,1,2], '-.')`
`show()`



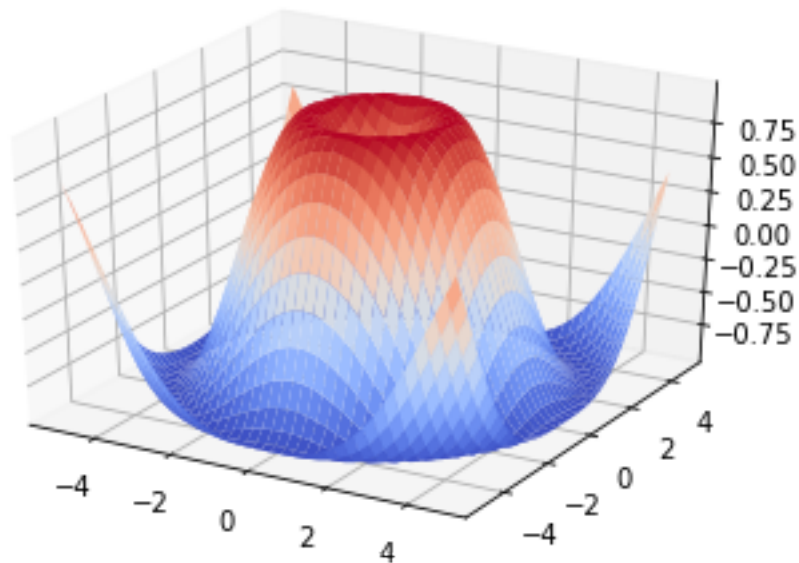
```
In [102]: from mpl_toolkits.mplot3d import Axes3D
          from matplotlib import *
          from matplotlib.ticker import *
          from matplotlib.pyplot import *

          figura = figure()
          ejes = figura.gca(projection='3d')

          # Datos
          X = arange(-5, 5, 0.25)
          Y = arange(-5, 5, 0.25)
          X, Y = meshgrid(X, Y)
          R = sqrt(X**2 + Y**2)
          Z = sin(R)

          # imprimimos superficie
          surf = ejes.plot_surface(X, Y, Z, cmap=cm.coolwarm)

          show()
```



Links: <https://matplotlib.org/tutorials/index.html#introductory>
<https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks>