

Herramientas Ejercicio 3

Modelado, Optimización y Simulación

Profesor
Germán Montoya

Oficina ML648

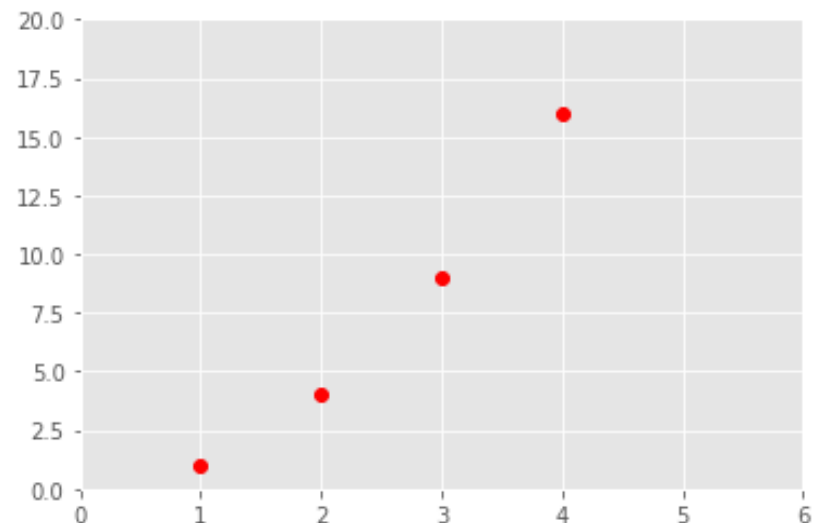
Graficas

Cargamos la librería

```
import matplotlib.pyplot as plt  
plt.style.use('ggplot')
```

Pintamos una señal donde la primera lista es el eje del tiempo y las segunda lista son los valores de la función.

```
plt.plot([1,2,3,4], [1,4,9,16], 'ro')  
plt.axis([0, 6, 0, 20])  
plt.show()
```



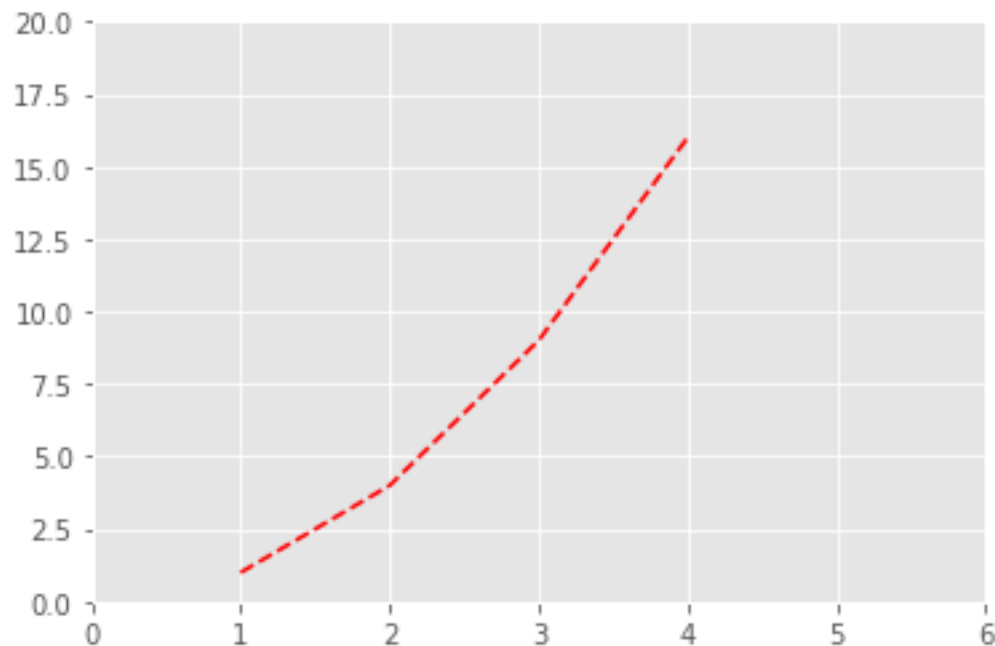
Graficas

Cuando queremos que las líneas sean interlineadas.

```
plt.plot([1,2,3,4], [1,4,9,16], 'r--')
```

```
plt.axis([0, 6, 0, 20])
```

```
plt.show()
```



Graficas

Cuando queremos graficar una señal senoidal en el eje del tiempo.

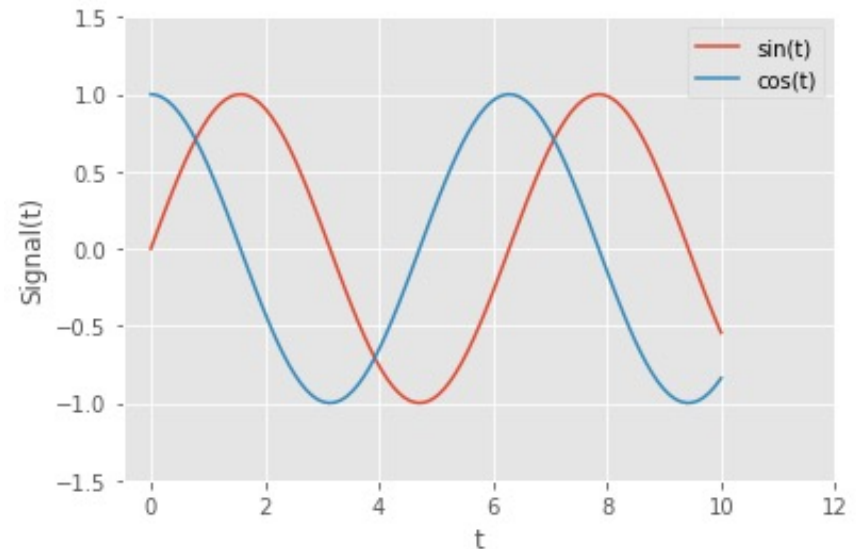
```
t = np.linspace(0, 10.0, 100)
plt.plot(t, np.sin(t))
plt.xlabel('Time')
plt.ylabel('Signal')
plt.title('My Plot  $\frac{a}{b}$ ');
plt.show()
```



Graficas

Varias curvas en la misma figura.

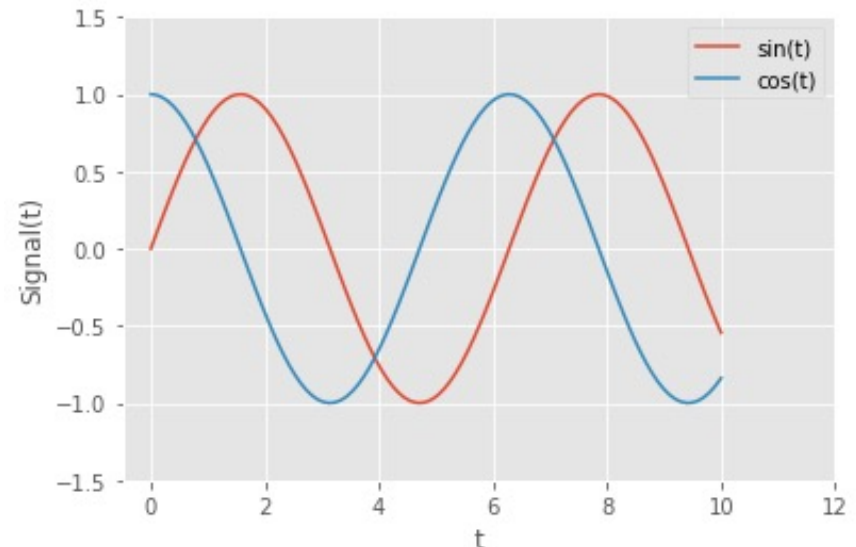
```
t = np.linspace(0, 10.0, 100)
plt.figure(1)
plt.plot(t, np.sin(t), label='sin(t)')
plt.plot(t, np.cos(t), label='cos(t)')
plt.xlabel('t')
plt.ylabel('Signal(t)')
plt.ylim(-1.5, 1.5)
plt.xlim(right=12.0)
plt.legend()
plt.show()
```



Graficas

Varias curvas en la misma figura.

```
t = np.linspace(0, 10.0, 100)
plt.figure(1)
plt.plot(t, np.sin(t), label='sin(t)')
plt.plot(t, np.cos(t), label='cos(t)')
plt.xlabel('t')
plt.ylabel('Signal(t)')
plt.ylim(-1.5, 1.5)
plt.xlim(right=12.0)
plt.legend()
plt.show()
```



Graficas

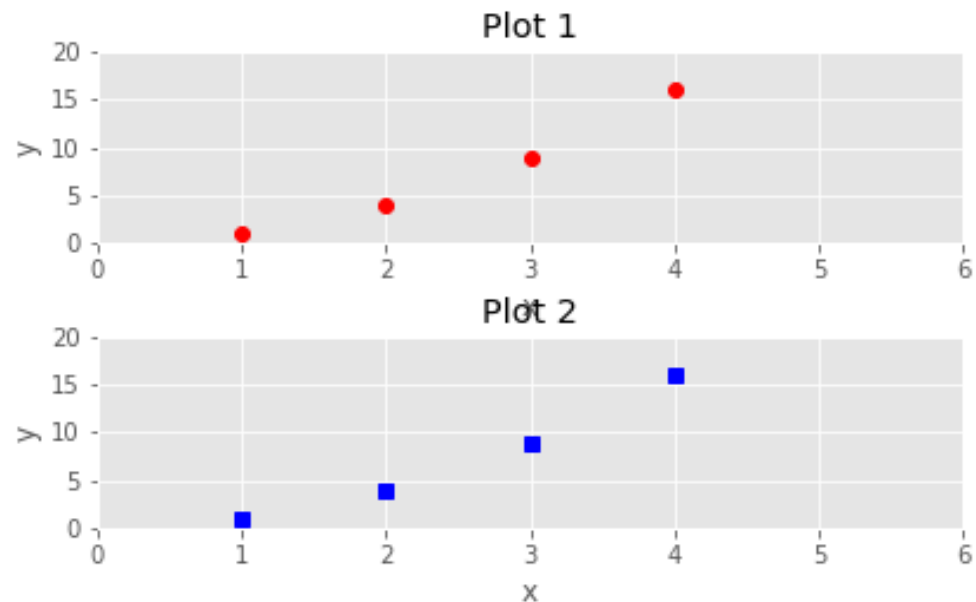
Cuando queremos manejar subplots (varias curvas en la misma figura pero separadas).

```
plt.figure(1)
plt.subplot(211)
plt.plot([1,2,3,4], [1,4,9,16], 'ro')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Plot 1')
plt.axis([0, 6, 0, 20])
plt.grid(True)
```

```
plt.subplot(212)
plt.plot([1,2,3,4], [1,4,9,16], 'bs')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Plot 2')
plt.axis([0, 6, 0, 20])
plt.grid(True)
```

```
plt.subplots_adjust(top=0.9, bottom=0.2, left=0.10, right=0.95, hspace=0.5, wspace=1)
```

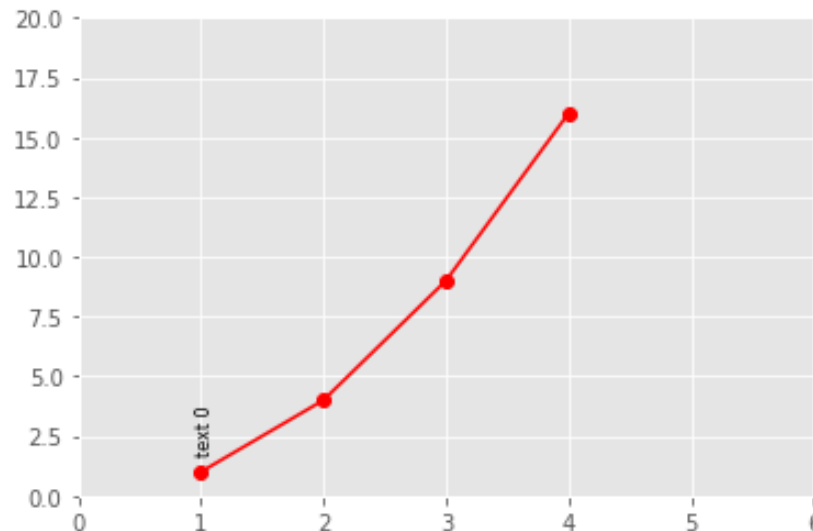
```
plt.show()
```



Graficas

Poner un texto en la curva.

```
plt.figure(1)  
plt.plot([1,2,3,4], [1,4,9,16], 'ro-')  
plt.axis([0, 6, 0, 20])  
plt.text(0.95, 3.2, 'text 0', rotation=90, size=8)  
plt.show()
```



Graficas

Graficar una única coordenada.

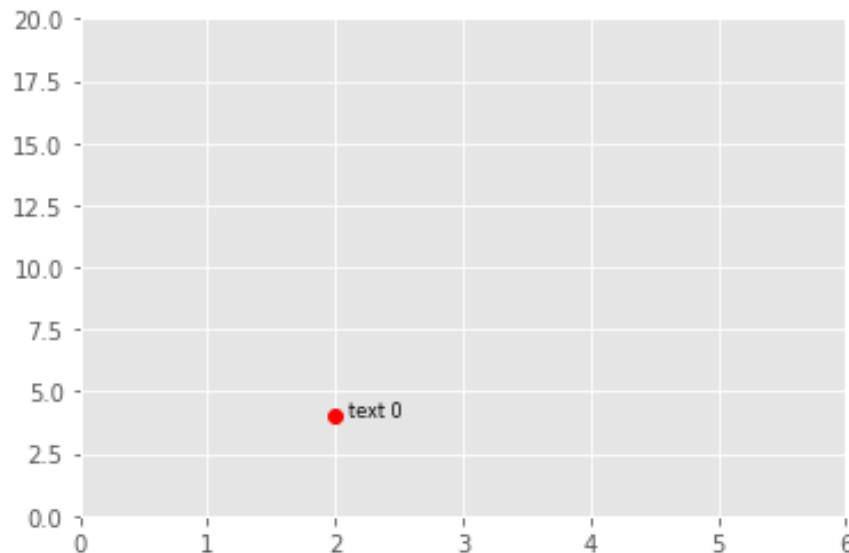
```
plt.figure(1)
```

```
plt.plot([2], [4], 'ro-')
```

```
plt.axis([0, 6, 0, 20])
```

```
plt.text(2.1, 4, 'text 0', size=8)
```

```
plt.show()
```



Graficas

Graficar una línea entre dos puntos.

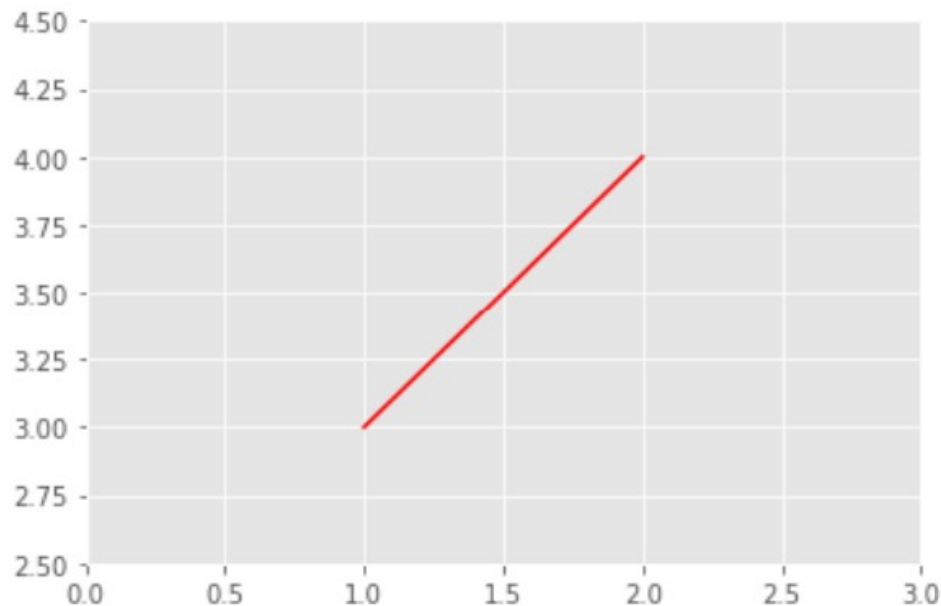
```
plt.figure(1)
```

```
#plt.plot([x1,x2], [y1,y2], 'r-')
```

```
plt.plot([1,2], [3,4], 'r-')
```

```
plt.axis([0, 3, 2.5, 4.5])
```

```
plt.show()
```



Resultados en Pyomo

Chequear cuanto vale una variable de decisión en Pyomo.

```
if Model.x[i,j].value==1:
```

sentencia 1

sentencia 2

.

.

sentencia n

Algunas operaciones adicionales

Raíz cuadrada.

```
import math
```

```
A=math.sqrt(4)
```

Elevar al cuadrado

```
A=3**2
```