2021-I_mcpp_taller_6_Valentina_Cuenca

April 9, 2021

1 Taller 6

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 9-abr-2021 11:59 PM

Valentina Cuenca

norma.cuenca@urosario.edu.co

1.1 Instrucciones:

- Guarde una copia de este *Jupyter Notebook* en su computador, idealmente en una carpeta destinada al material del curso.
- Modifique el nombre del archivo del *notebook*, agregando al final un guión inferior y su nombre y apellido, separados estos últimos por otro guión inferior. Por ejemplo, mi *notebook* se llamaría: mcpp_taller6_santiago_matallana
- Marque el *notebook* con su nombre y e-mail en el bloque verde arriba. Reemplace el texto "[Su nombre acá]" con su nombre y apellido. Similar para su e-mail.
- Desarrolle la totalidad del taller sobre este *notebook*, insertando las celdas que sea necesario debajo de cada pregunta. Haga buen uso de las celdas para código y de las celdas tipo *markdown* según el caso.
- Recuerde salvar periódicamente sus avances.
- Cuando termine el taller:
  1. Descárguelo en PDF. Si tiene algún problema con la conversión, descárguelo en HTML.
  2. Suba todos los archivos a su repositorio en GitHub, en una carpeta destinada exclusivamente para este taller, antes de la fecha y hora límites.

(Todos los ejercicios tienen el mismo valor.)

1.1.1 Resuelva la parte 1 de este documento.

[20]:
```
import numpy as np
import scipy.linalg as la
import matplotlib.pyplot as plt
import math as m
```

1. Choose a value and set the variable x to that value

[3]:
```
x= 20
```

2. What is command to compute the square of x? Its cube?

```
[7]: np.square(x)
```

```
[7]: 400
```

```
[8]: np.power(x, 3)
```

```
[8]: 8000
```

3. Choose an angle   and set the variable theta to its value (a number)

```
[13]: theta= 30
```

4. What is sin ? cos ? Angles can be measured in degrees or radians. Which of these are being used used?

Esta función espera que le des el valor en radiones

```
[9]: np.sin(theta)
```

```
[9]: -0.9880316240928618
```

```
[10]: np.cos(theta)
```

```
[10]: 0.15425144988758405
```

5. Use the np.linspace function to create a row vector called meshPoints containing exactly 500 values with values evenly spaced between -1 and 1.

```
[15]: meshPoints= np.linspace(-1,1,200)
      meshPoints
```

```
[15]: array([-1.        , -0.98994975, -0.9798995 , -0.96984925, -0.95979899,
             -0.94974874, -0.93969849, -0.92964824, -0.91959799, -0.90954774,
             -0.89949749, -0.88944724, -0.87939698, -0.86934673, -0.85929648,
             -0.84924623, -0.83919598, -0.82914573, -0.81909548, -0.80904523,
             -0.79899497, -0.78894472, -0.77889447, -0.76884422, -0.75879397,
             -0.74874372, -0.73869347, -0.72864322, -0.71859296, -0.70854271,
             -0.69849246, -0.68844221, -0.67839196, -0.66834171, -0.65829146,
             -0.64824121, -0.63819095, -0.6281407 , -0.61809045, -0.6080402 ,
             -0.59798995, -0.5879397 , -0.57788945, -0.5678392 , -0.55778894,
             -0.54773869, -0.53768844, -0.52763819, -0.51758794, -0.50753769,
             -0.49748744, -0.48743719, -0.47738693, -0.46733668, -0.45728643,
             -0.44723618, -0.43718593, -0.42713568, -0.41708543, -0.40703518,
             -0.39698492, -0.38693467, -0.37688442, -0.36683417, -0.35678392,
             -0.34673367, -0.33668342, -0.32663317, -0.31658291, -0.30653266,
             -0.29648241, -0.28643216, -0.27638191, -0.26633166, -0.25628141,
             -0.24623116, -0.2361809 , -0.22613065, -0.2160804 , -0.20603015,
             -0.1959799 , -0.18592965, -0.1758794 , -0.16582915, -0.15577889,
       -0.14572864, -0.13567839, -0.12562814, -0.11557789, -0.10552764,
       -0.09547739, -0.08542714, -0.07537688, -0.06532663, -0.05527638,
       -0.04522613, -0.03517588, -0.02512563, -0.01507538, -0.00502513,
        0.00502513,  0.01507538,  0.02512563,  0.03517588,  0.04522613,
        0.05527638,  0.06532663,  0.07537688,  0.08542714,  0.09547739,
        0.10552764,  0.11557789,  0.12562814,  0.13567839,  0.14572864,
        0.15577889,  0.16582915,  0.1758794 ,  0.18592965,  0.1959799 ,
        0.20603015,  0.2160804 ,  0.22613065,  0.2361809 ,  0.24623116,
        0.25628141,  0.26633166,  0.27638191,  0.28643216,  0.29648241,
        0.30653266,  0.31658291,  0.32663317,  0.33668342,  0.34673367,
        0.35678392,  0.36683417,  0.37688442,  0.38693467,  0.39698492,
        0.40703518,  0.41708543,  0.42713568,  0.43718593,  0.44723618,
        0.45728643,  0.46733668,  0.47738693,  0.48743719,  0.49748744,
        0.50753769,  0.51758794,  0.52763819,  0.53768844,  0.54773869,
        0.55778894,  0.5678392 ,  0.57788945,  0.5879397 ,  0.59798995,
        0.6080402 ,  0.61809045,  0.6281407 ,  0.63819095,  0.64824121,
        0.65829146,  0.66834171,  0.67839196,  0.68844221,  0.69849246,
        0.70854271,  0.71859296,  0.72864322,  0.73869347,  0.74874372,
        0.75879397,  0.76884422,  0.77889447,  0.78894472,  0.79899497,
        0.80904523,  0.81909548,  0.82914573,  0.83919598,  0.84924623,
        0.85929648,  0.86934673,  0.87939698,  0.88944724,  0.89949749,
        0.90954774,  0.91959799,  0.92964824,  0.93969849,  0.94974874,
        0.95979899,  0.96984925,  0.9798995 ,  0.98994975,  1.        ])
```

6. What expression will yield the value of the 53th element of meshPoints? What is this value?

Creo que producirá el valor de -0.46733668.

```
[16]: meshPoints[53]
```

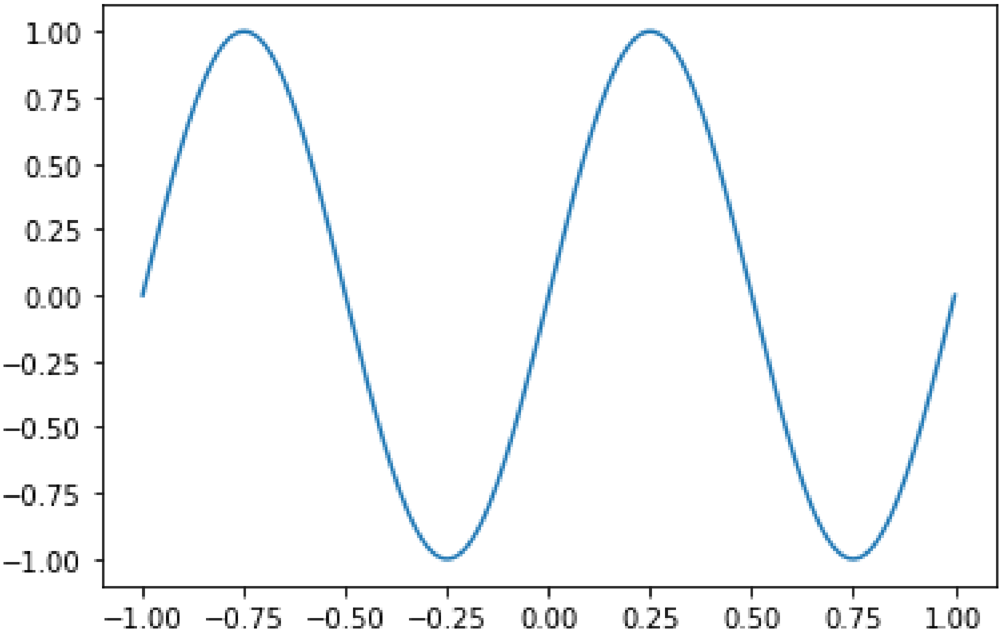
```
[16]: -0.46733668341708545
```

7. Produce a plot of a sinusoid on the interval [−1, 1] using the command

plt.plot(meshPoints,np.sin(2*pi*meshPoints))

Please save this plot as a jpeg (.jpg) file and send it along with your work.

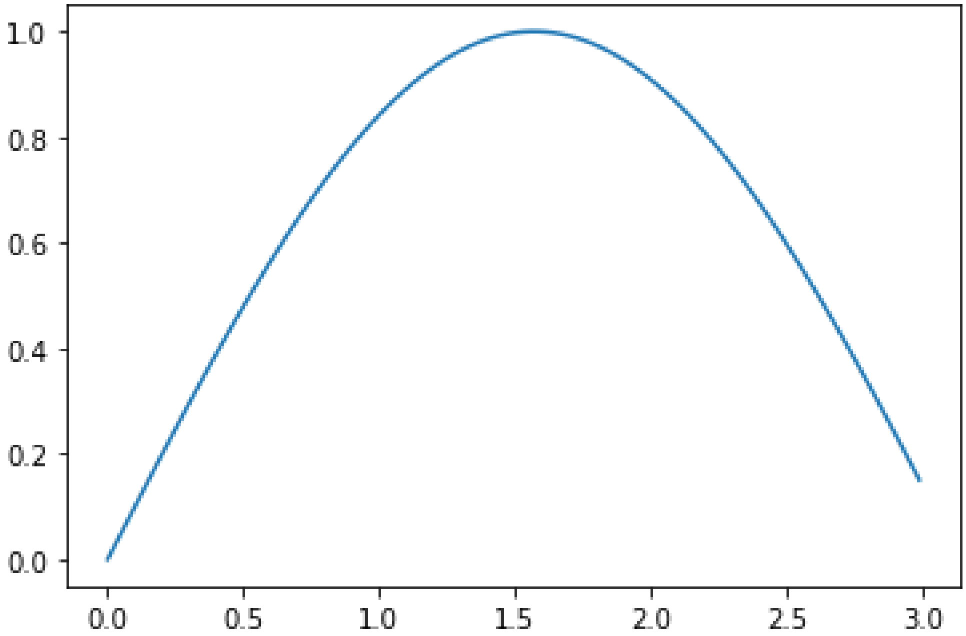
```
[24]: plt.plot(meshPoints,np.sin(2*m.pi*meshPoints));
      plt.savefig("Plot generated using Matplotlib.jpg")
```



1.1.2 Resuelva los ejercicios de las secciones 4.1, 5.1, 6.1, 7.4 y 8.5 de este documento.

1. Plot a simple graph of a sinus function in the range 0 to 3 with a step size of 0.01

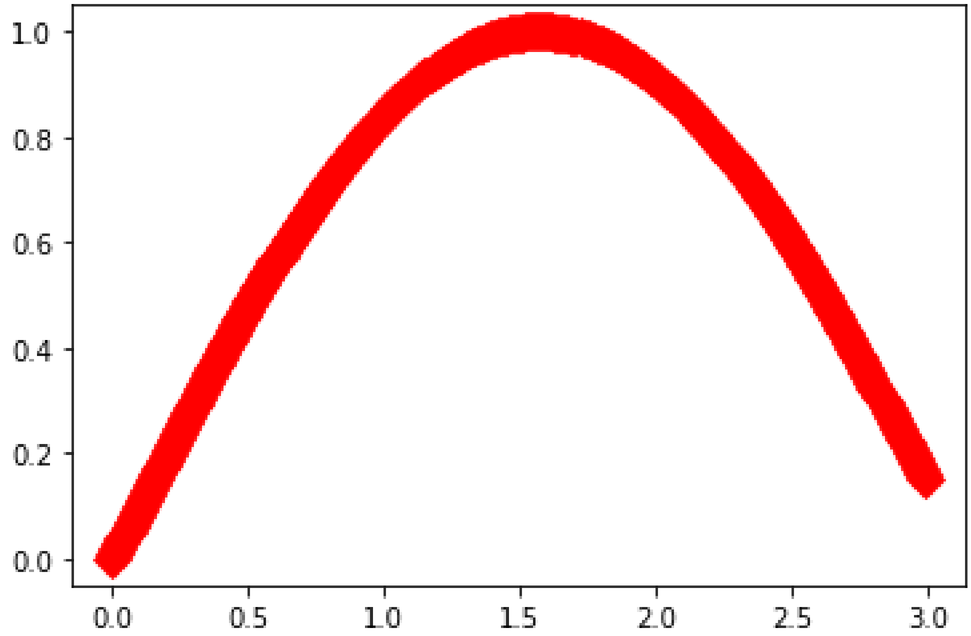
[29]:
```
x=np.arange(0,3,0.01)
plt.plot(x,np.sin(x));
```



2. Make the line red. Add diamond-shaped markers with size of 5

[38]:

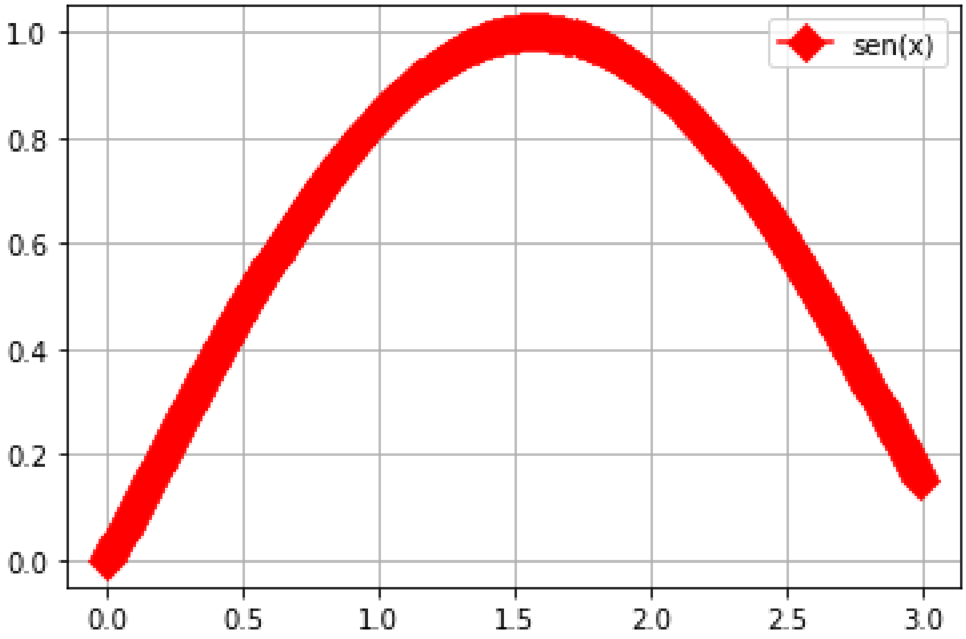
```
x=np.arange(0,3,0.01)
plt.plot(x,np.sin(x), linestyle="-",marker="D",color="red", mew=5 );
```



3. . Add a legend and a grid to the plot.

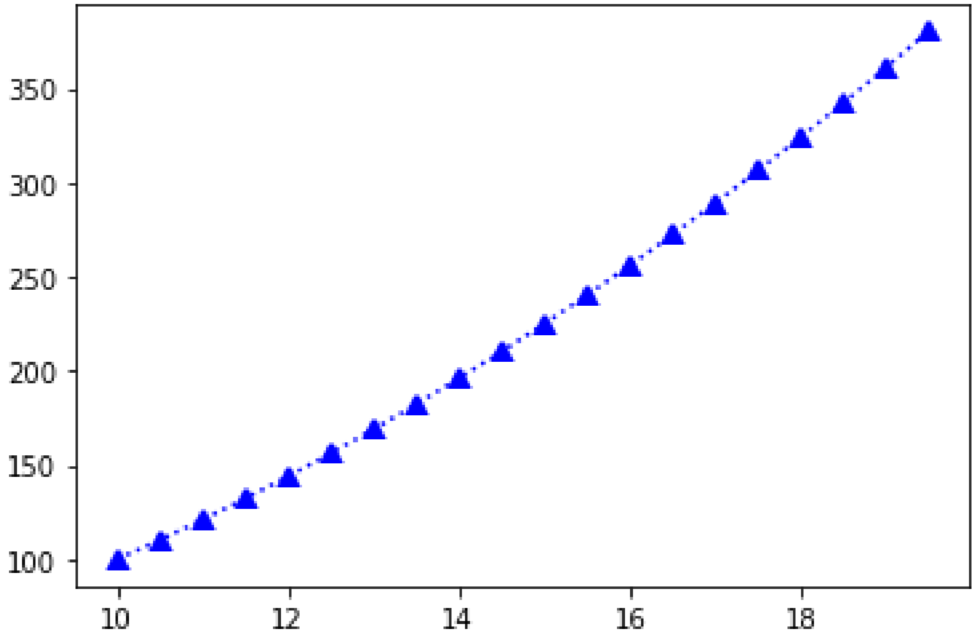
[41]:

```
x=np.arange(0,3,0.01)
plt.grid()
plt.plot(x,np.sin(x), linestyle="-",marker="D",color="red",␣
 ↪mew=5,label="sen(x)")
plt.legend();
```

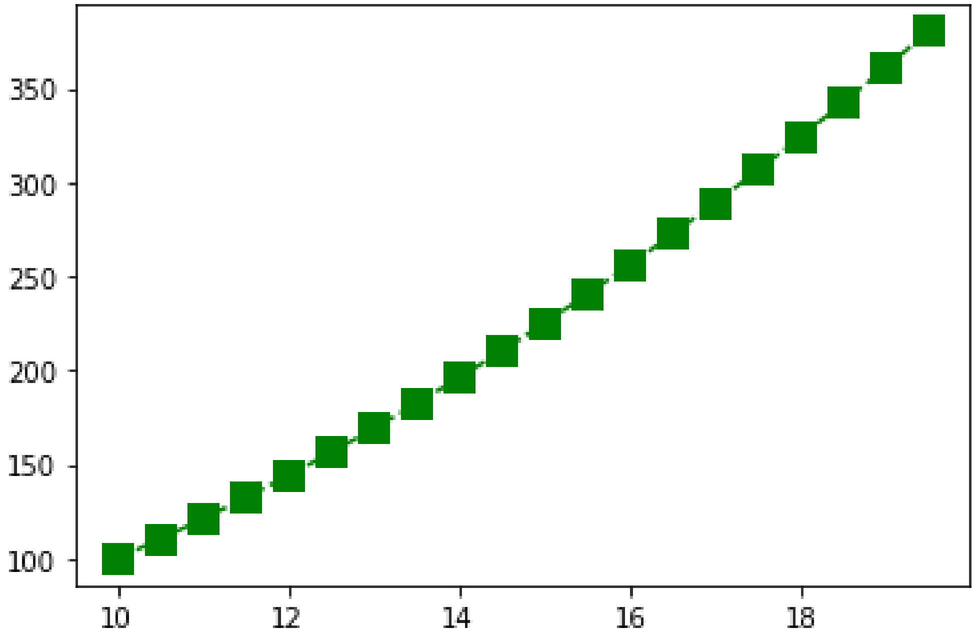


1. Apply different line styles to a plot. Change line color and thickness as well as the size and the kind of the marker. Experiment with different styles.

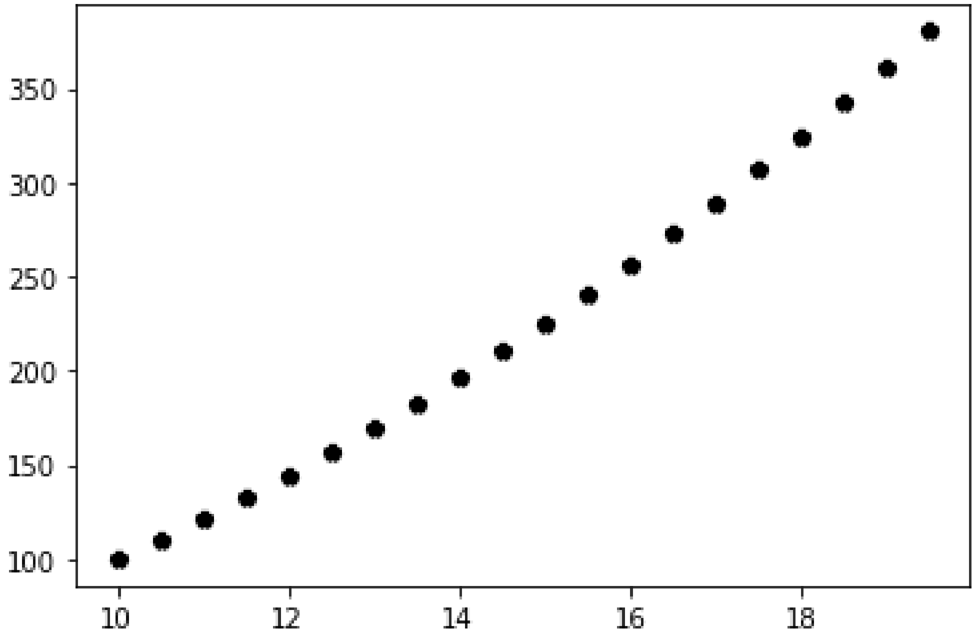
[44]:
```
y=np.arange(10,20,0.5)
plt.plot(y,y**2, linestyle=":",marker="^",color="b", mew=2);
```



```
[48]: plt.plot(y,y**2, linestyle="--",marker="s",color="g", mew=6);
```



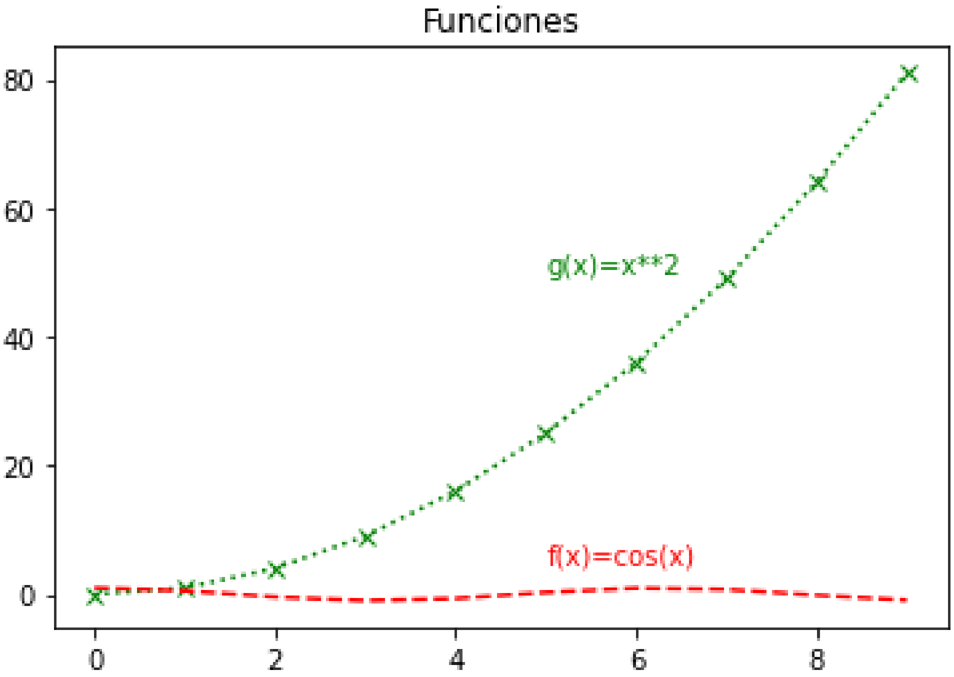
```
[52]: plt.plot(y,y**2, linestyle="None",marker="*",color="k", mew=3);
```



1. Annotate a line at two places with text. Use green and red arrows and align it according to figure points and data.

[88]:

```
lista1 = np.arange(0,10,1)
plt.title("Funciones")
plt.plot(lista1,np.square(lista1) ,marker='x', linestyle=':', color='g')

plt.plot(lista1,np.cos(lista1) , linestyle='--', color='r')
t1 = plt.text(5, 5, 'f(x)=cos(x)',color="r")
t2=plt.text(5,50,"g(x)=x**2",color="g")
```

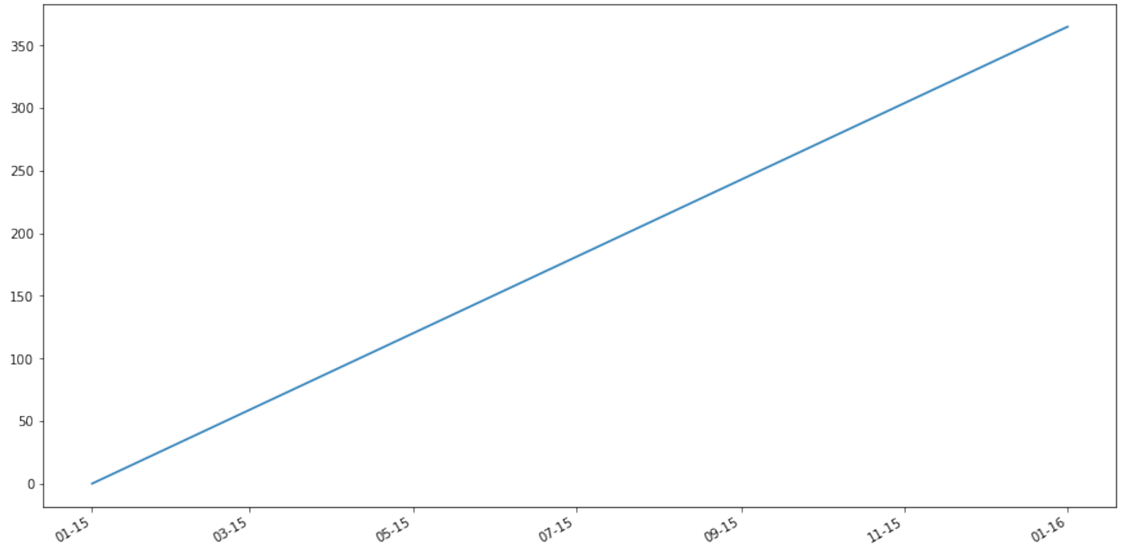


1. Plot a graph with dates for one year with daily values at the x axis using the built-in module datetime.

[135]:

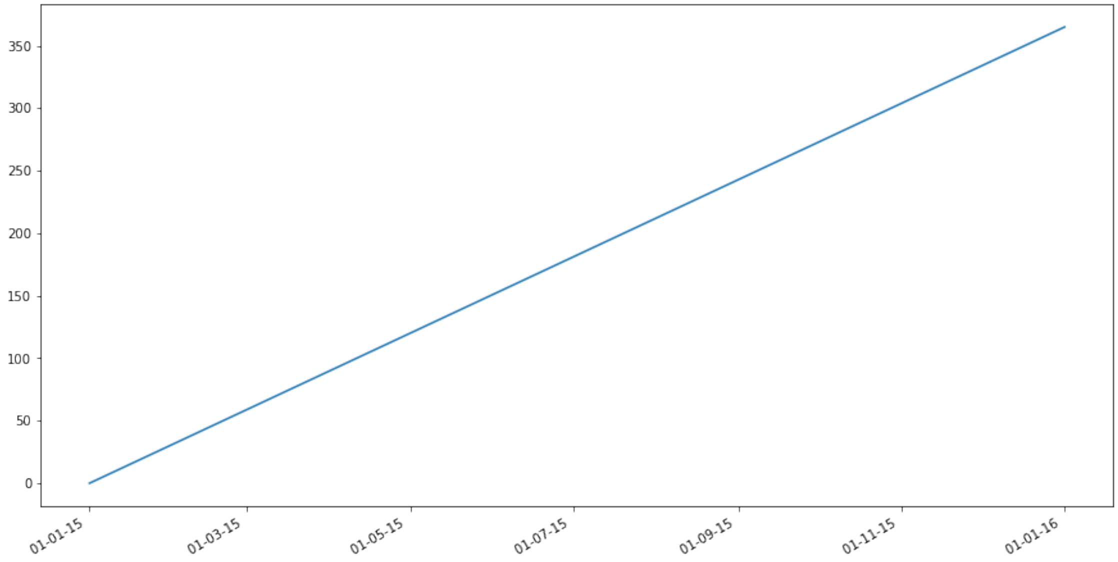
```
import pandas as pd
import matplotlib.dates as mdates

times = pd.date_range("2015-01-01", periods=366, freq="1D")

fig, ax = plt.subplots(1)
fig.autofmt_xdate()
plt.plot(times, range(times.size))

xfmt = mdates.DateFormatter("%m-%y")
```

```
ax.xaxis.set_major_formatter(xfmt)

plt.show()
```



2. Format the dates in such a way that only the first day of the month is shown.

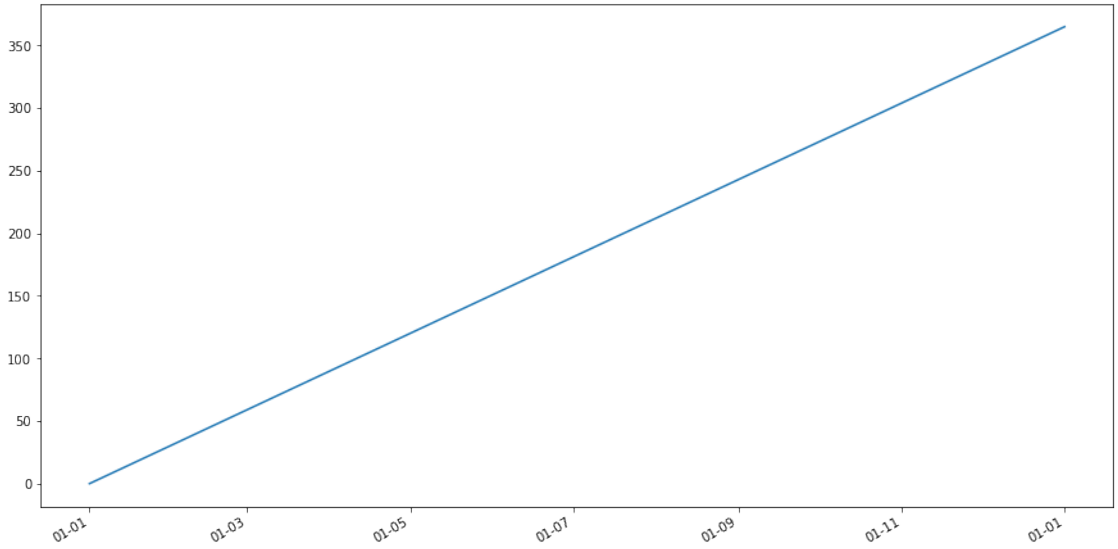
[129]:
```
times = pd.date_range("2015-01-01", periods=366, freq="1D")

fig, ax = plt.subplots(1)
fig.autofmt_xdate()
plt.plot(times, range(times.size))

xfmt = mdates.DateFormatter("%d-%m-%y")
ax.xaxis.set_major_formatter(xfmt)

plt.show()
```



3. Display the dates with and without the year. Show the month as number and as first three letters of the month name.

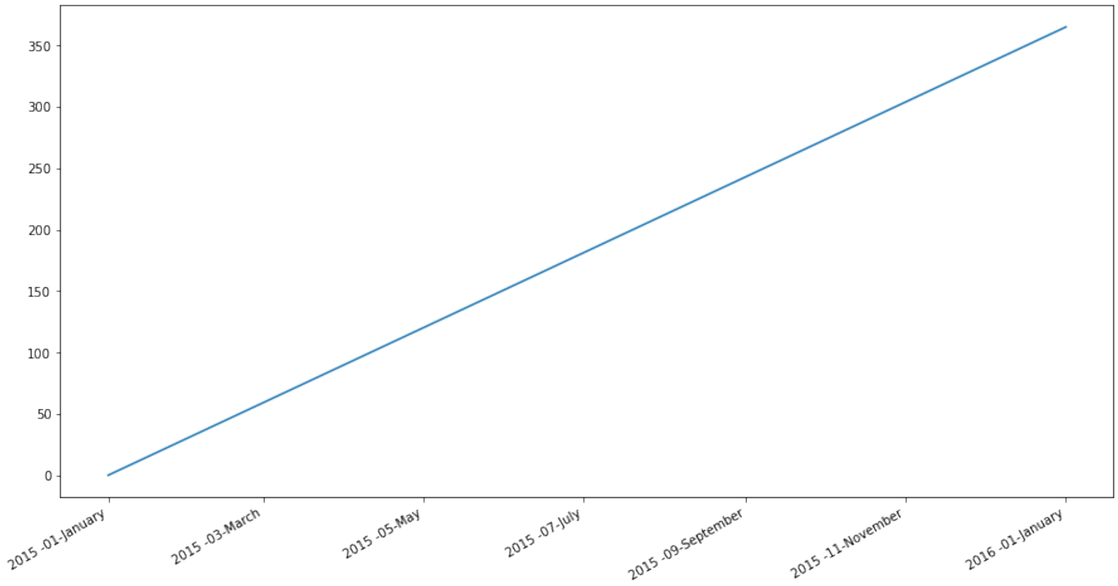
[136]:

```
fig, ax = plt.subplots(1)
fig.autofmt_xdate()
plt.plot(times, range(times.size))

xfmt = mdates.DateFormatter("%d-%m")
ax.xaxis.set_major_formatter(xfmt)

plt.show()
```



[139]:

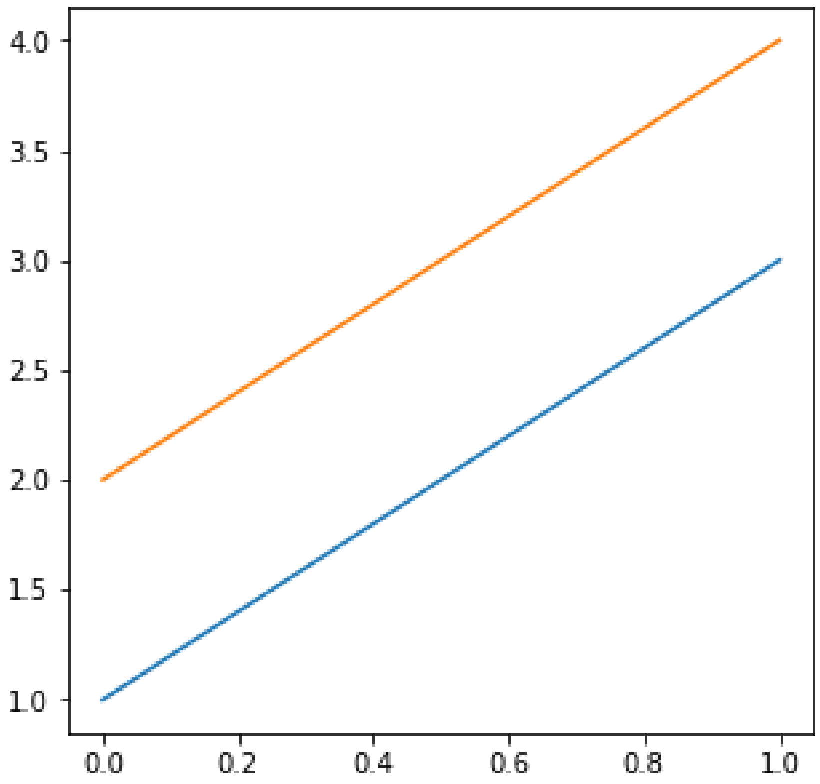
```
fig, ax = plt.subplots(1)
fig.autofmt_xdate()
plt.plot(times, range(times.size))

xfmt = mdates.DateFormatter("%Y -%m-%B")
ax.xaxis.set_major_formatter(xfmt)

plt.show()
```

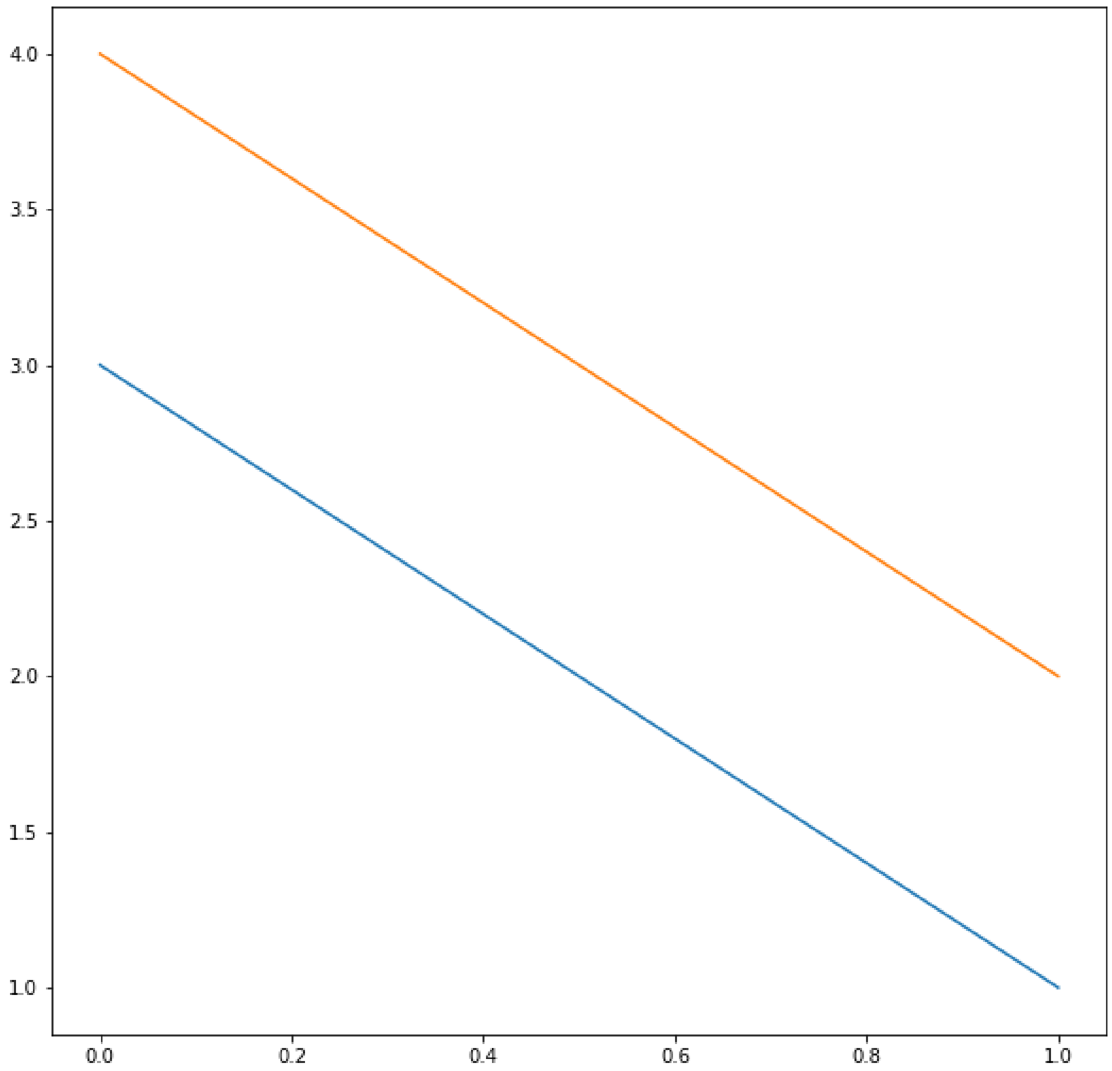


1. Draw two figures, one 5 by 5, one 10 by 10 inches.

[163]:

```
plt.figure(1,figsize=(5,5))

plt.plot([[1,2], [3, 4]])

plt.show()
plt.figure(2,figsize=(10,10))
plt.plot([[3,4], [1, 2]])
plt.show()
```

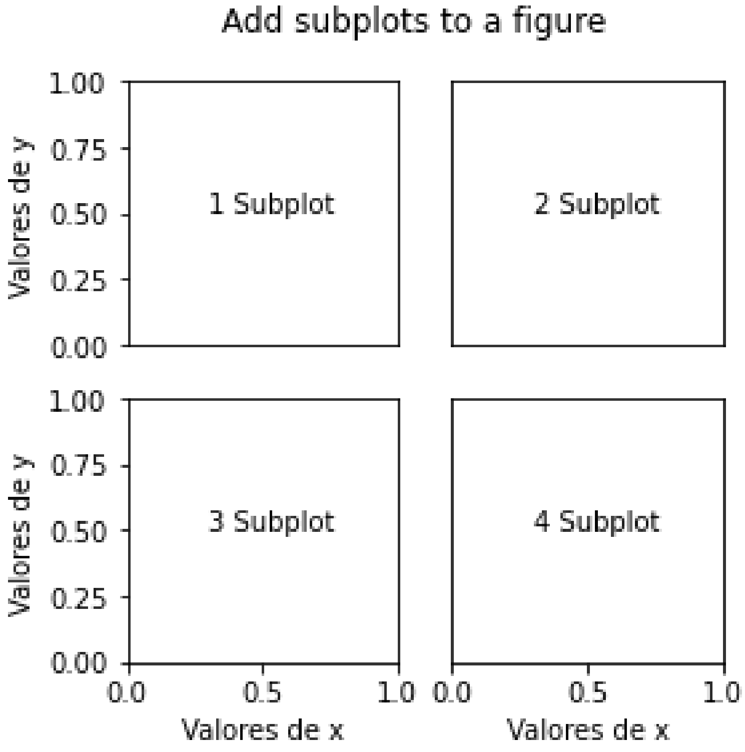

4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0



[155]:
```
plt.rcParams["figure.figsize"] = (4, 4)

plt.show()
```

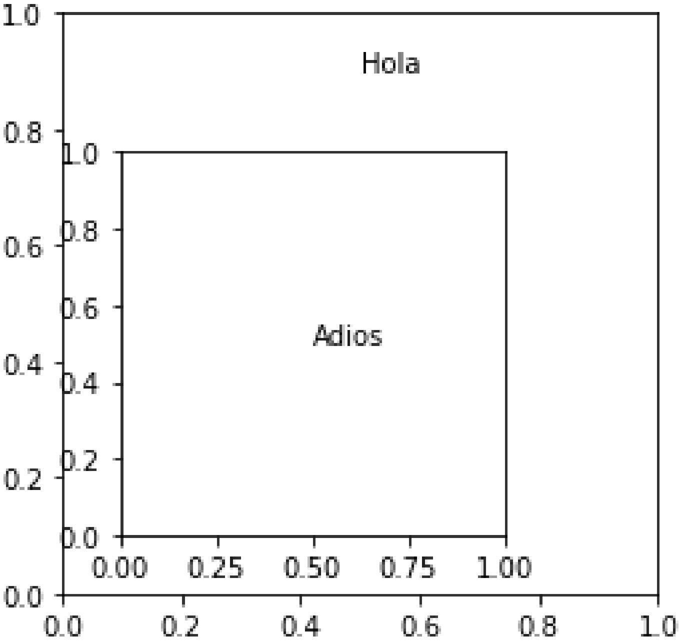
[179]:
```
fig=plt.figure()

subplot1=fig.add_subplot(2,2,1)
subplot1.text(0.3, 0.5, '1 Subplot')
plt.ylabel("Valores de y")
subplot1.get_xaxis().set_visible(False)

subplot2=fig.add_subplot(2,2,2)
subplot2.text(0.3, 0.5, '2 Subplot')
subplot2.get_xaxis().set_visible(False)
```

```
subplot2.get_yaxis().set_visible(False)

subplot3=fig.add_subplot(2,2,3)
subplot3.text(0.3, 0.5, '3 Subplot')
plt.ylabel("Valores de y")
plt.xlabel("Valores de x")


subplot4=fig.add_subplot(2,2,4)
subplot4.text(0.3, 0.5, '4 Subplot')
plt.xlabel("Valores de x")
subplot4.get_yaxis().set_visible(False)
fig.suptitle("Add subplots to a figure")
plt.show()
```



3. Place a small plot in one bigger plot.

[190]:
```
fig=plt.figure()
x=plt.axes()
x.text(0.5, 0.9, 'Hola')
y=plt.axes([0.2,0.2,0.5,0.5])
y.text(0.5, 0.5, 'Adios')
plt.show()
```


Hola
Adios
0.0
0.2
0.4
0.6
0.8
1.0
0.00
0.25
0.50
0.75
1.00

[ ]: