Taller 6

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 27-mar-2020 11:59 PM

[Andrés Ramírez Vela]

[andrese.ramirez@urosario.edu.co]

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.)

Resuelva la parte 1 de este documento (http://www.math.pitt.edu/~sussmanm/3040Summer14/exercisesII.pdf).

In [1]:
```
import numpy as np
import matplotlib.pylab as plt
```

In [2]:
```
#1#

a = 2

#2#

np.power(a,2)
```

Out[2]: 4

In [3]:
```
np.power(a,3)
```

Out[3]: 8

In [4]:
```
#3#
theta = np.deg2rad(90)
#el angulo está medido en radianes, si se desea usar grados se debe conv
ertir los valores a esta medida#
```

In [5]:
```
#4#
np.sin(theta)
```

Out[5]: 1.0

In [6]:
```
np.round(np.cos(theta))
```

Out[6]: 0.0

In [7]:
```
#5
meshpoints = np.linspace(-1,1,500)
meshpoints
```

Out[7]:
```
array([-1.        , -0.99599198, -0.99198397, -0.98797595, -0.98396794,
       -0.97995992, -0.9759519 , -0.97194389, -0.96793587, -0.96392786,
       -0.95991984, -0.95591182, -0.95190381, -0.94789579, -0.94388778,
       -0.93987976, -0.93587174, -0.93186373, -0.92785571, -0.9238477 ,
       -0.91983968, -0.91583166, -0.91182365, -0.90781563, -0.90380762,
       -0.8997996 , -0.89579158, -0.89178357, -0.88777555, -0.88376754,
       -0.87975952, -0.8757515 , -0.87174349, -0.86773547, -0.86372745,
       -0.85971944, -0.85571142, -0.85170341, -0.84769539, -0.84368737,
       -0.83967936, -0.83567134, -0.83166333, -0.82765531, -0.82364729,
       -0.81963928, -0.81563126, -0.81162325, -0.80761523, -0.80360721,
       -0.7995992 , -0.79559118, -0.79158317, -0.78757515, -0.78356713,
       -0.77955912, -0.7755511 , -0.77154309, -0.76753507, -0.76352705,
       -0.75951904, -0.75551102, -0.75150301, -0.74749499, -0.74348697,
       -0.73947896, -0.73547094, -0.73146293, -0.72745491, -0.72344689,
       -0.71943888, -0.71543086, -0.71142285, -0.70741483, -0.70340681,
       -0.6993988 , -0.69539078, -0.69138277, -0.68737475, -0.68336673,
       -0.67935872, -0.6753507 , -0.67134269, -0.66733467, -0.66332665,
       -0.65931864, -0.65531062, -0.65130261, -0.64729459, -0.64328657,
       -0.63927856, -0.63527054, -0.63126253, -0.62725451, -0.62324649,
       -0.61923848, -0.61523046, -0.61122244, -0.60721443, -0.60320641,
       -0.5991984 , -0.59519038, -0.59118236, -0.58717435, -0.58316633,
       -0.57915832, -0.5751503 , -0.57114228, -0.56713427, -0.56312625,
       -0.55911824, -0.55511022, -0.5511022 , -0.54709419, -0.54308617,
       -0.53907816, -0.53507014, -0.53106212, -0.52705411, -0.52304609,
       -0.51903808, -0.51503006, -0.51102204, -0.50701403, -0.50300601,
       -0.498998  , -0.49498998, -0.49098196, -0.48697395, -0.48296593,
       -0.47895792, -0.4749499 , -0.47094188, -0.46693387, -0.46292585,
       -0.45891784, -0.45490982, -0.4509018 , -0.44689379, -0.44288577,
       -0.43887776, -0.43486974, -0.43086172, -0.42685371, -0.42284569,
       -0.41883768, -0.41482966, -0.41082164, -0.40681363, -0.40280561,
       -0.3987976 , -0.39478958, -0.39078156, -0.38677355, -0.38276553,
       -0.37875752, -0.3747495 , -0.37074148, -0.36673347, -0.36272545,
       -0.35871743, -0.35470942, -0.3507014 , -0.34669339, -0.34268537,
       -0.33867735, -0.33466934, -0.33066132, -0.32665331, -0.32264529,
       -0.31863727, -0.31462926, -0.31062124, -0.30661323, -0.30260521,
       -0.29859719, -0.29458918, -0.29058116, -0.28657315, -0.28256513,
       -0.27855711, -0.2745491 , -0.27054108, -0.26653307, -0.26252505,
       -0.25851703, -0.25450902, -0.250501  , -0.24649299, -0.24248497,
       -0.23847695, -0.23446894, -0.23046092, -0.22645291, -0.22244489,
       -0.21843687, -0.21442886, -0.21042084, -0.20641283, -0.20240481,
       -0.19839679, -0.19438878, -0.19038076, -0.18637275, -0.18236473,
       -0.17835671, -0.1743487 , -0.17034068, -0.16633267, -0.16232465,
       -0.15831663, -0.15430862, -0.1503006 , -0.14629259, -0.14228457,
       -0.13827655, -0.13426854, -0.13026052, -0.12625251, -0.12224449,
       -0.11823647, -0.11422846, -0.11022044, -0.10621242, -0.10220441,
       -0.09819639, -0.09418838, -0.09018036, -0.08617234, -0.08216433,
       -0.07815631, -0.0741483 , -0.07014028, -0.06613226, -0.06212425,
       -0.05811623, -0.05410822, -0.0501002 , -0.04609218, -0.04208417,
       -0.03807615, -0.03406814, -0.03006012, -0.0260521 , -0.02204409,
       -0.01803607, -0.01402806, -0.01002004, -0.00601202, -0.00200401,
        0.00200401,  0.00601202,  0.01002004,  0.01402806,  0.01803607,
        0.02204409,  0.0260521 ,  0.03006012,  0.03406814,  0.03807615,
        0.04208417,  0.04609218,  0.0501002 ,  0.05410822,  0.05811623,
        0.06212425,  0.06613226,  0.07014028,  0.0741483 ,  0.07815631,
        0.08216433,  0.08617234,  0.09018036,  0.09418838,  0.09819639,
        0.10220441,  0.10621242,  0.11022044,  0.11422846,  0.11823647,
        0.12224449,  0.12625251,  0.13026052,  0.13426854,  0.13827655,
```

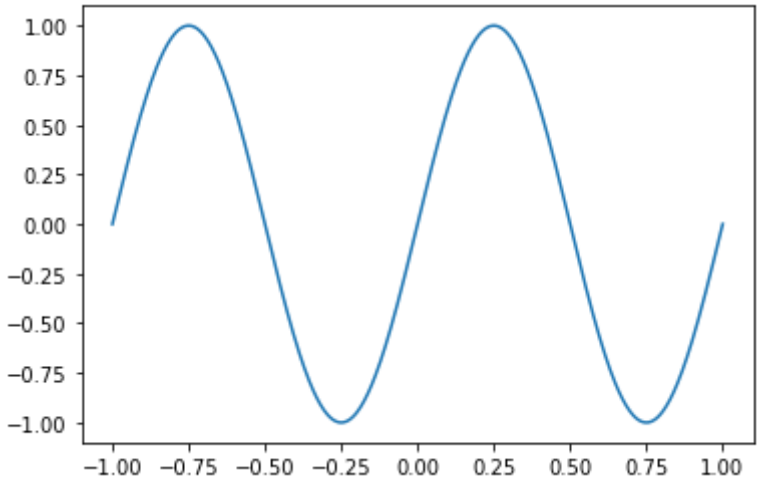
```
        0.14228457,  0.14629259,  0.1503006 ,  0.15430862,  0.15831663,
        0.16232465,  0.16633267,  0.17034068,  0.1743487 ,  0.17835671,
        0.18236473,  0.18637275,  0.19038076,  0.19438878,  0.19839679,
        0.20240481,  0.20641283,  0.21042084,  0.21442886,  0.21843687,
        0.22244489,  0.22645291,  0.23046092,  0.23446894,  0.23847695,
        0.24248497,  0.24649299,  0.250501  ,  0.25450902,  0.25851703,
        0.26252505,  0.26653307,  0.27054108,  0.2745491 ,  0.27855711,
        0.28256513,  0.28657315,  0.29058116,  0.29458918,  0.29859719,
        0.30260521,  0.30661323,  0.31062124,  0.31462926,  0.31863727,
        0.32264529,  0.32665331,  0.33066132,  0.33466934,  0.33867735,
        0.34268537,  0.34669339,  0.3507014 ,  0.35470942,  0.35871743,
        0.36272545,  0.36673347,  0.37074148,  0.3747495 ,  0.37875752,
        0.38276553,  0.38677355,  0.39078156,  0.39478958,  0.3987976 ,
        0.40280561,  0.40681363,  0.41082164,  0.41482966,  0.41883768,
        0.42284569,  0.42685371,  0.43086172,  0.43486974,  0.43887776,
        0.44288577,  0.44689379,  0.4509018 ,  0.45490982,  0.45891784,
        0.46292585,  0.46693387,  0.47094188,  0.4749499 ,  0.47895792,
        0.48296593,  0.48697395,  0.49098196,  0.49498998,  0.498998  ,
        0.50300601,  0.50701403,  0.51102204,  0.51503006,  0.51903808,
        0.52304609,  0.52705411,  0.53106212,  0.53507014,  0.53907816,
        0.54308617,  0.54709419,  0.5511022 ,  0.55511022,  0.55911824,
        0.56312625,  0.56713427,  0.57114228,  0.5751503 ,  0.57915832,
        0.58316633,  0.58717435,  0.59118236,  0.59519038,  0.5991984 ,
        0.60320641,  0.60721443,  0.61122244,  0.61523046,  0.61923848,
        0.62324649,  0.62725451,  0.63126253,  0.63527054,  0.63927856,
        0.64328657,  0.64729459,  0.65130261,  0.65531062,  0.65931864,
        0.66332665,  0.66733467,  0.67134269,  0.6753507 ,  0.67935872,
        0.68336673,  0.68737475,  0.69138277,  0.69539078,  0.6993988 ,
        0.70340681,  0.70741483,  0.71142285,  0.71543086,  0.71943888,
        0.72344689,  0.72745491,  0.73146293,  0.73547094,  0.73947896,
        0.74348697,  0.74749499,  0.75150301,  0.75551102,  0.75951904,
        0.76352705,  0.76753507,  0.77154309,  0.7755511 ,  0.77955912,
        0.78356713,  0.78757515,  0.79158317,  0.79559118,  0.7995992 ,
        0.80360721,  0.80761523,  0.81162325,  0.81563126,  0.81963928,
        0.82364729,  0.82765531,  0.83166333,  0.83567134,  0.83967936,
        0.84368737,  0.84769539,  0.85170341,  0.85571142,  0.85971944,
        0.86372745,  0.86773547,  0.87174349,  0.8757515 ,  0.87975952,
        0.88376754,  0.88777555,  0.89178357,  0.89579158,  0.8997996 ,
        0.90380762,  0.90781563,  0.91182365,  0.91583166,  0.91983968,
        0.9238477 ,  0.92785571,  0.93186373,  0.93587174,  0.93987976,
        0.94388778,  0.94789579,  0.95190381,  0.95591182,  0.95991984,
        0.96392786,  0.96793587,  0.97194389,  0.9759519 ,  0.97995992,
        0.98396794,  0.98797595,  0.99198397,  0.99599198,  1.
])
```

In [8]:
```
#6
print(meshpoints[52])
```

```
-0.7915831663326653
```

In [9]:
```
#7
plt.plot(meshpoints,np.sin(2*np.pi*meshpoints))
```

Out[9]:
```
[<matplotlib.lines.Line2D at 0x1113ab890>]
```



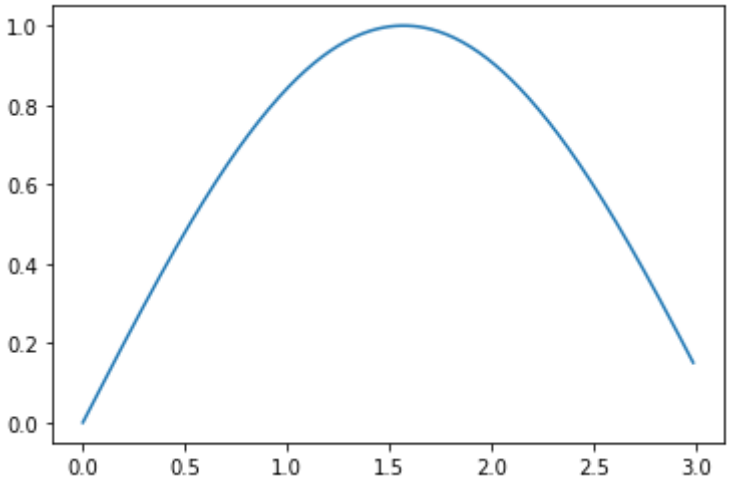
Resuelva los ejercicios de las secciones 4.1, 5.1, 6.1, 7.4 y 8.5 de este documento (http://www.python-academy.com/download/pycon2012/matplotlib_handout.pdf).

In [10]:
```
#4.1.
x = np.arange(0, 3, 0.01)
print(x)
```

```
[0.   0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1  0.11 0.12 0.13
 0.14 0.15 0.16 0.17 0.18 0.19 0.2  0.21 0.22 0.23 0.24 0.25 0.26 0.27
 0.28 0.29 0.3  0.31 0.32 0.33 0.34 0.35 0.36 0.37 0.38 0.39 0.4  0.41
 0.42 0.43 0.44 0.45 0.46 0.47 0.48 0.49 0.5  0.51 0.52 0.53 0.54 0.55
 0.56 0.57 0.58 0.59 0.6  0.61 0.62 0.63 0.64 0.65 0.66 0.67 0.68 0.69
 0.7  0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.8  0.81 0.82 0.83
 0.84 0.85 0.86 0.87 0.88 0.89 0.9  0.91 0.92 0.93 0.94 0.95 0.96 0.97
 0.98 0.99 1.   1.01 1.02 1.03 1.04 1.05 1.06 1.07 1.08 1.09 1.1  1.11
 1.12 1.13 1.14 1.15 1.16 1.17 1.18 1.19 1.2  1.21 1.22 1.23 1.24 1.25
 1.26 1.27 1.28 1.29 1.3  1.31 1.32 1.33 1.34 1.35 1.36 1.37 1.38 1.39
 1.4  1.41 1.42 1.43 1.44 1.45 1.46 1.47 1.48 1.49 1.5  1.51 1.52 1.53
 1.54 1.55 1.56 1.57 1.58 1.59 1.6  1.61 1.62 1.63 1.64 1.65 1.66 1.67
 1.68 1.69 1.7  1.71 1.72 1.73 1.74 1.75 1.76 1.77 1.78 1.79 1.8  1.81
 1.82 1.83 1.84 1.85 1.86 1.87 1.88 1.89 1.9  1.91 1.92 1.93 1.94 1.95
 1.96 1.97 1.98 1.99 2.   2.01 2.02 2.03 2.04 2.05 2.06 2.07 2.08 2.09
 2.1  2.11 2.12 2.13 2.14 2.15 2.16 2.17 2.18 2.19 2.2  2.21 2.22 2.23
 2.24 2.25 2.26 2.27 2.28 2.29 2.3  2.31 2.32 2.33 2.34 2.35 2.36 2.37
 2.38 2.39 2.4  2.41 2.42 2.43 2.44 2.45 2.46 2.47 2.48 2.49 2.5  2.51
 2.52 2.53 2.54 2.55 2.56 2.57 2.58 2.59 2.6  2.61 2.62 2.63 2.64 2.65
 2.66 2.67 2.68 2.69 2.7  2.71 2.72 2.73 2.74 2.75 2.76 2.77 2.78 2.79
 2.8  2.81 2.82 2.83 2.84 2.85 2.86 2.87 2.88 2.89 2.9  2.91 2.92 2.93
 2.94 2.95 2.96 2.97 2.98 2.99]
```

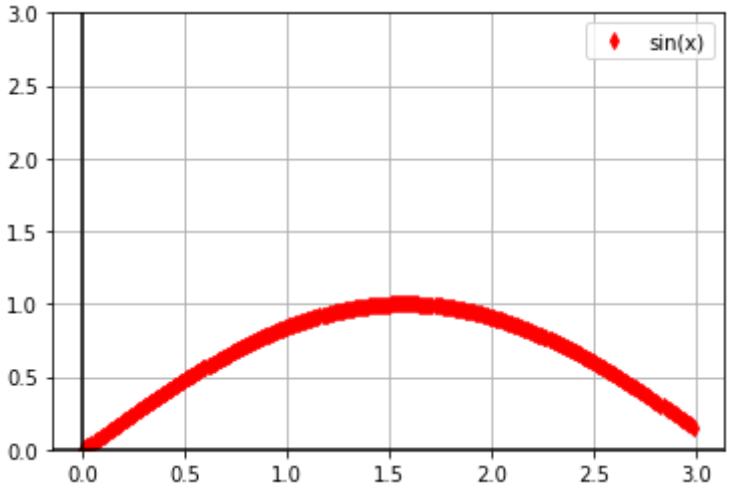
In [11]:
```
#4.1.1
plt.plot(x, np.sin(x))
```

Out[11]:
```
[<matplotlib.lines.Line2D at 0x1114cd450>]
```



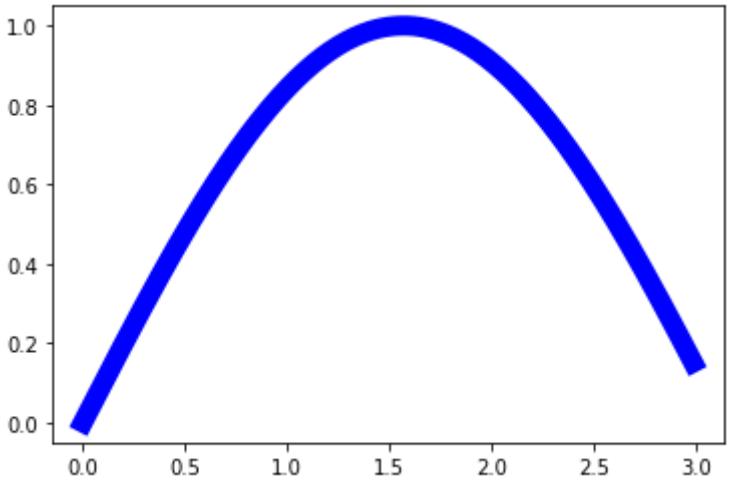
In [12]:
```
#4.1.2 Y 4.1.3
plt.plot(x, np.sin(x), "d", markersize= 5, color="red", label="sin(x)")
plt.ylim(0, 3)
plt.axhline(0, color="black")
plt.axvline(0, color="black")
plt.grid(True)
plt.legend()
```

Out[12]:
```
<matplotlib.legend.Legend at 0x111543e50>
```



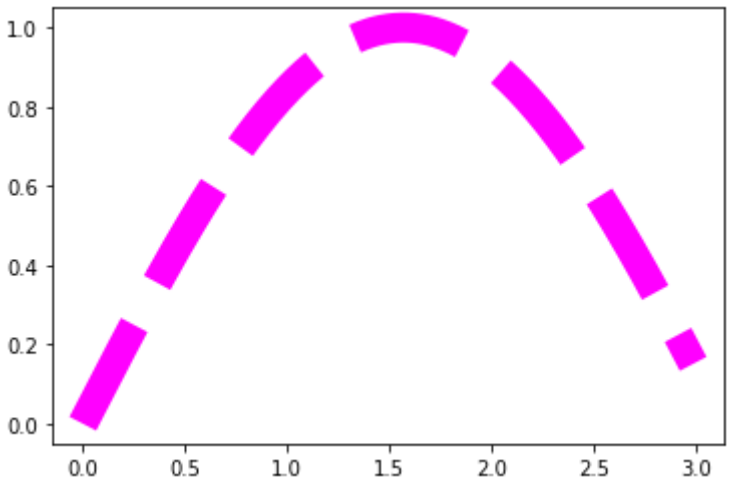
In [13]:
```
#5.1
plt.plot(x, np.sin(x), "-", linewidth= 10, color="blue", label="sin(x)")
```

Out[13]:
```
[<matplotlib.lines.Line2D at 0x111684390>]
```



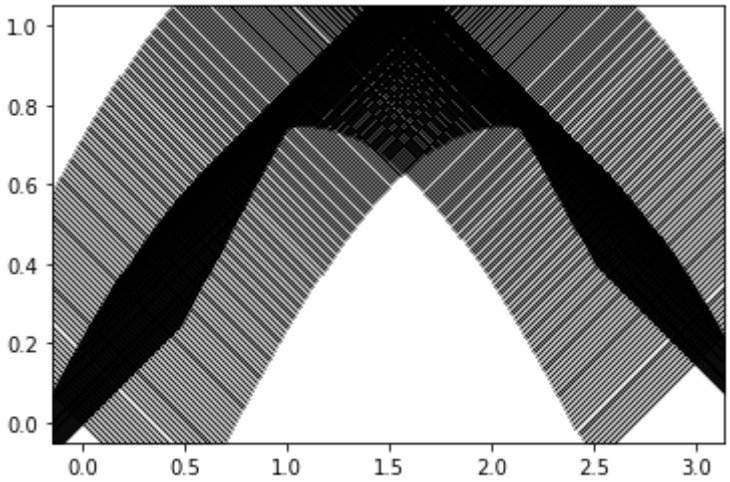
In [14]:
```
plt.plot(x, np.sin(x), "--", linewidth= 15, color="magenta", label="sin
(x)")
```

Out[14]:
```
[<matplotlib.lines.Line2D at 0x1116f4210>]
```



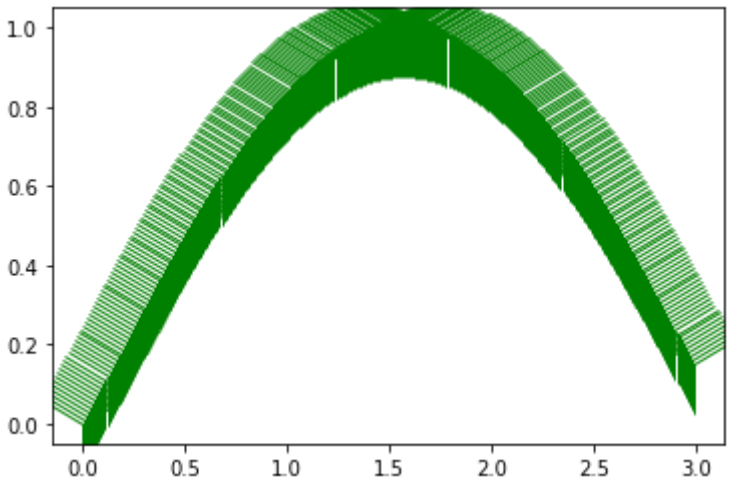
In [15]:
```
plt.plot(x, np.sin(x), "x", markersize= 100, color="black", label="sin
(x)")
```

Out[15]:
```
[<matplotlib.lines.Line2D at 0x11181a090>]
```



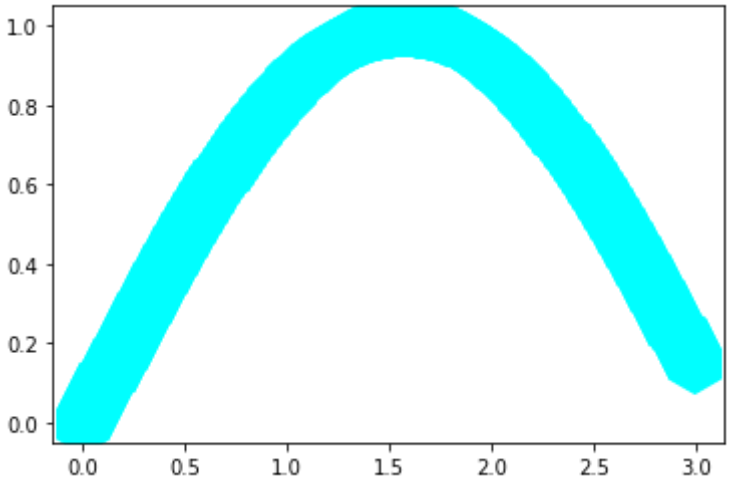
In [16]:
```
plt.plot(x, np.sin(x), "1", markersize= 50, color="green", label="sin
(x)")
```

Out[16]:
```
[<matplotlib.lines.Line2D at 0x1118d5ed0>]
```

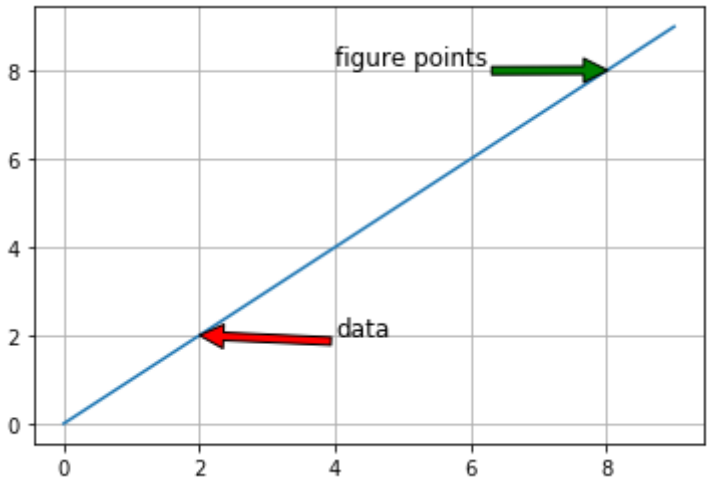


In [17]:
```
plt.plot(x, np.sin(x), "h", markersize= 30, color="cyan", label="sin(x)"
)
```

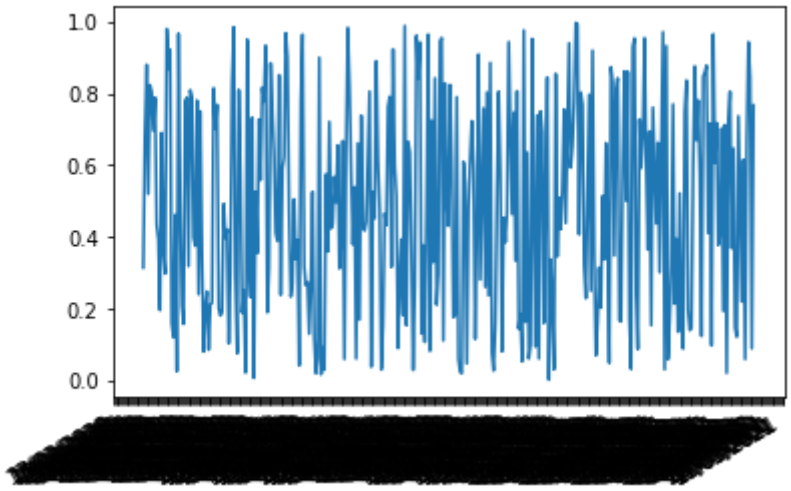
Out[17]:
```
[<matplotlib.lines.Line2D at 0x111944cd0>]
```



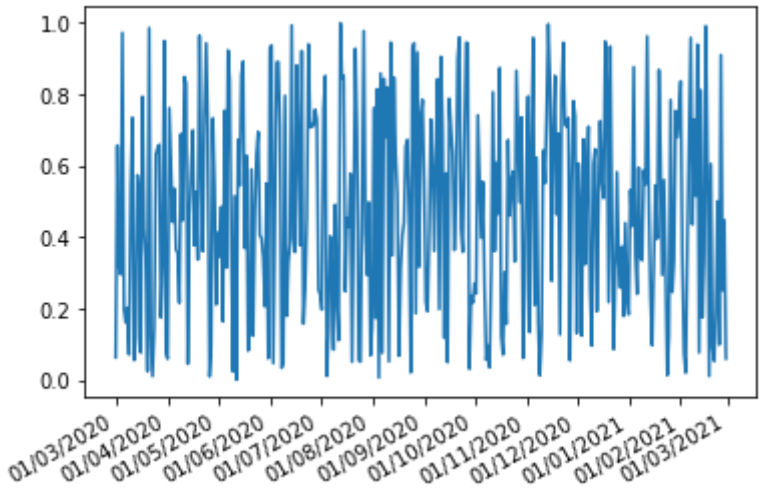
In [18]:
```
#6.1
plt.plot(np.arange(10))
plt.annotate('figure points', xy=(307,210), xycoords='figure points', xy
text= (4,8),
             textcoords= "data" ,arrowprops=dict(facecolor='green'),
            horizontalalignment='left', verticalalignment='bottom', font
size=12)
plt.annotate('data', xy=(2,2), xycoords='data', xytext= (4,2), textcoord
s= "data",
             arrowprops=dict(facecolor='red'), fontsize=12)
plt.grid(True)
```



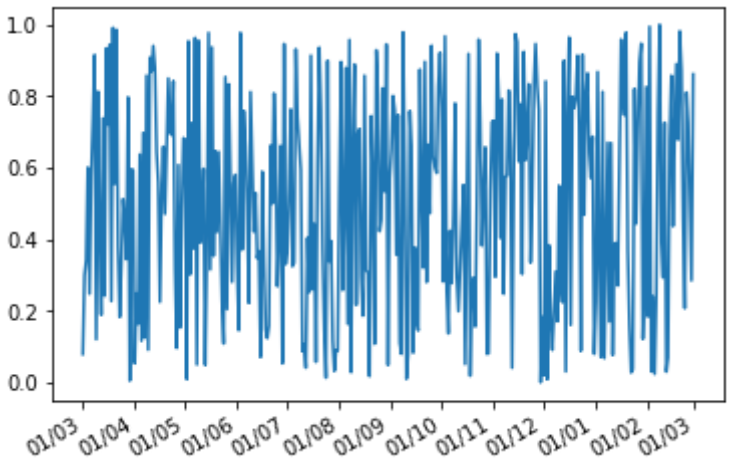
In [19]:
```
#7.4.1. En este ejercicio no entendi muy bien si se tocaba mostrar los 3
65 días,
#por ello no se notan los valores en x, sin embargo, el punto siguiente
 (7.4.1),
#muestra lo que considero unifica los punto 7.4.1. y 7.4.2
import matplotlib.dates as mdates
import datetime as dt
base = dt.datetime(2020,3,1)
dates = np.array([base + dt.timedelta(days=i) for i in range(365)])
y = np.random.rand(len(dates))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d/%m/%Y'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator())
plt.plot(dates,y)
plt.gcf().autofmt_xdate()
```



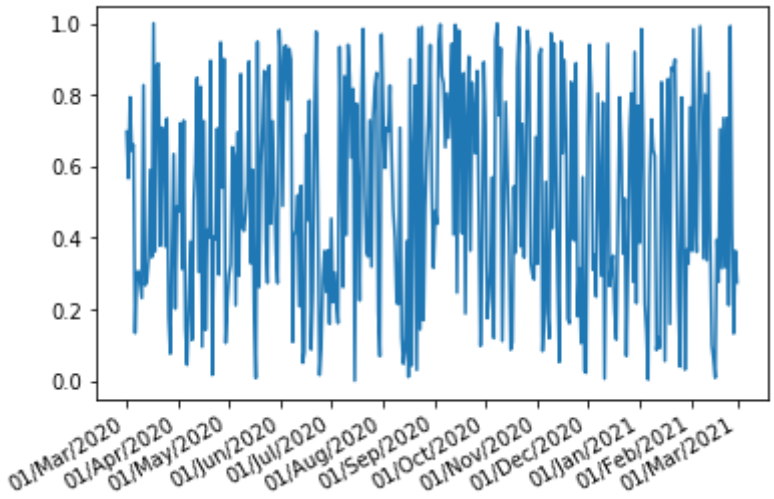
In [20]:
```
#7.4.2
import matplotlib.dates as mdates
import datetime as dt
base = dt.datetime(2020,3,1)
dates = np.array([base + dt.timedelta(days=i) for i in range(365)])
y = np.random.rand(len(dates))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d/%m/%Y'))
plt.gca().xaxis.set_major_locator(mdates.MonthLocator())
plt.plot(dates,y)
plt.gcf().autofmt_xdate()
```



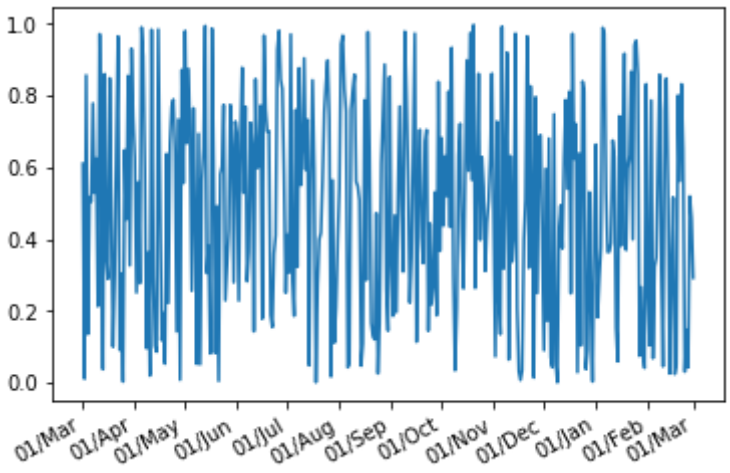
In [21]:
```
#7.3
base = dt.datetime(2020,3,1)
dates = np.array([base + dt.timedelta(days=i) for i in range(365)])
y = np.random.rand(len(dates))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d/%m'))
plt.gca().xaxis.set_major_locator(mdates.MonthLocator())
plt.plot(dates,y)
plt.gcf().autofmt_xdate()
```



In [22]:
```
#7.3
base = dt.datetime(2020,3,1)
dates = np.array([base + dt.timedelta(days=i) for i in range(365)])
y = np.random.rand(len(dates))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d/%b/%Y'))
plt.gca().xaxis.set_major_locator(mdates.MonthLocator())
plt.plot(dates,y)
plt.gcf().autofmt_xdate()
```



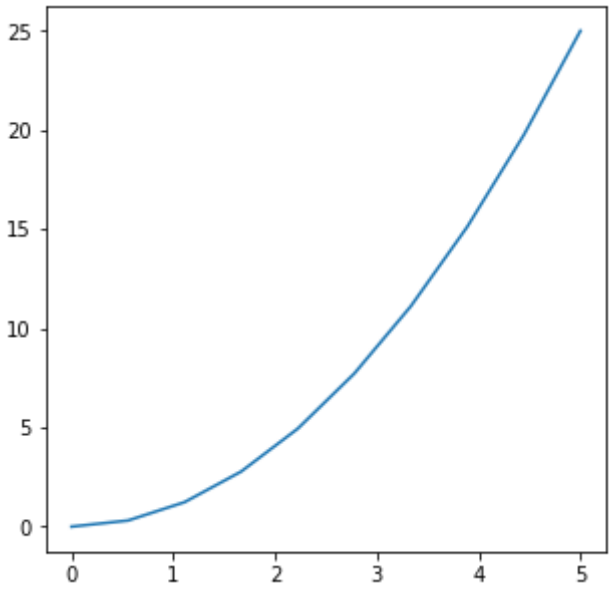
In [23]:
```
#7.3
base = dt.datetime(2020,3,1)
dates = np.array([base + dt.timedelta(days=i) for i in range(365)])
y = np.random.rand(len(dates))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%d/%b'))
plt.gca().xaxis.set_major_locator(mdates.MonthLocator())
plt.plot(dates,y)
plt.gcf().autofmt_xdate()
```

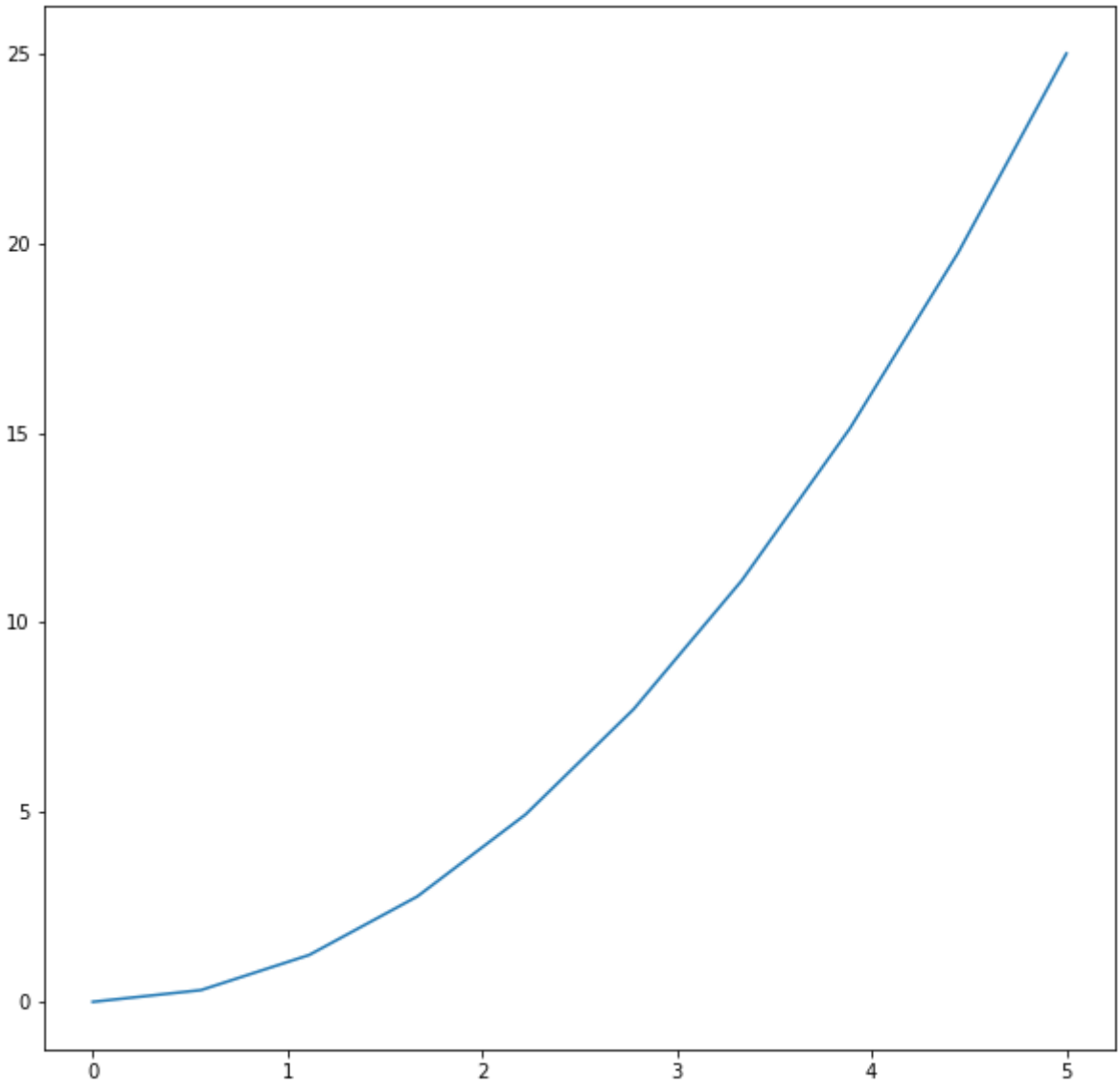


In [24]:
```
#8.5.1
x = np.linspace(0, 5, 10)
y = x ** 2
plt.figure(figsize=(5, 5))
plt.plot(x,y)

z = np.linspace(0, 5, 10)
v = x ** 2
plt.figure(figsize=(10, 10))
plt.plot(z,v)
#el punto no señalaba que se hiciera mediante un subplot por eso lo hice
de esta manera
```

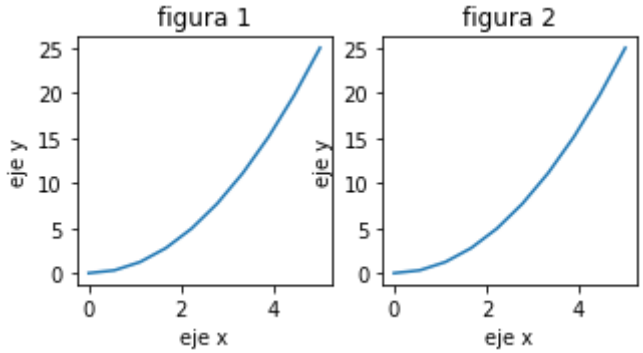
Out[24]: [<matplotlib.lines.Line2D at 0x112ba91d0>]





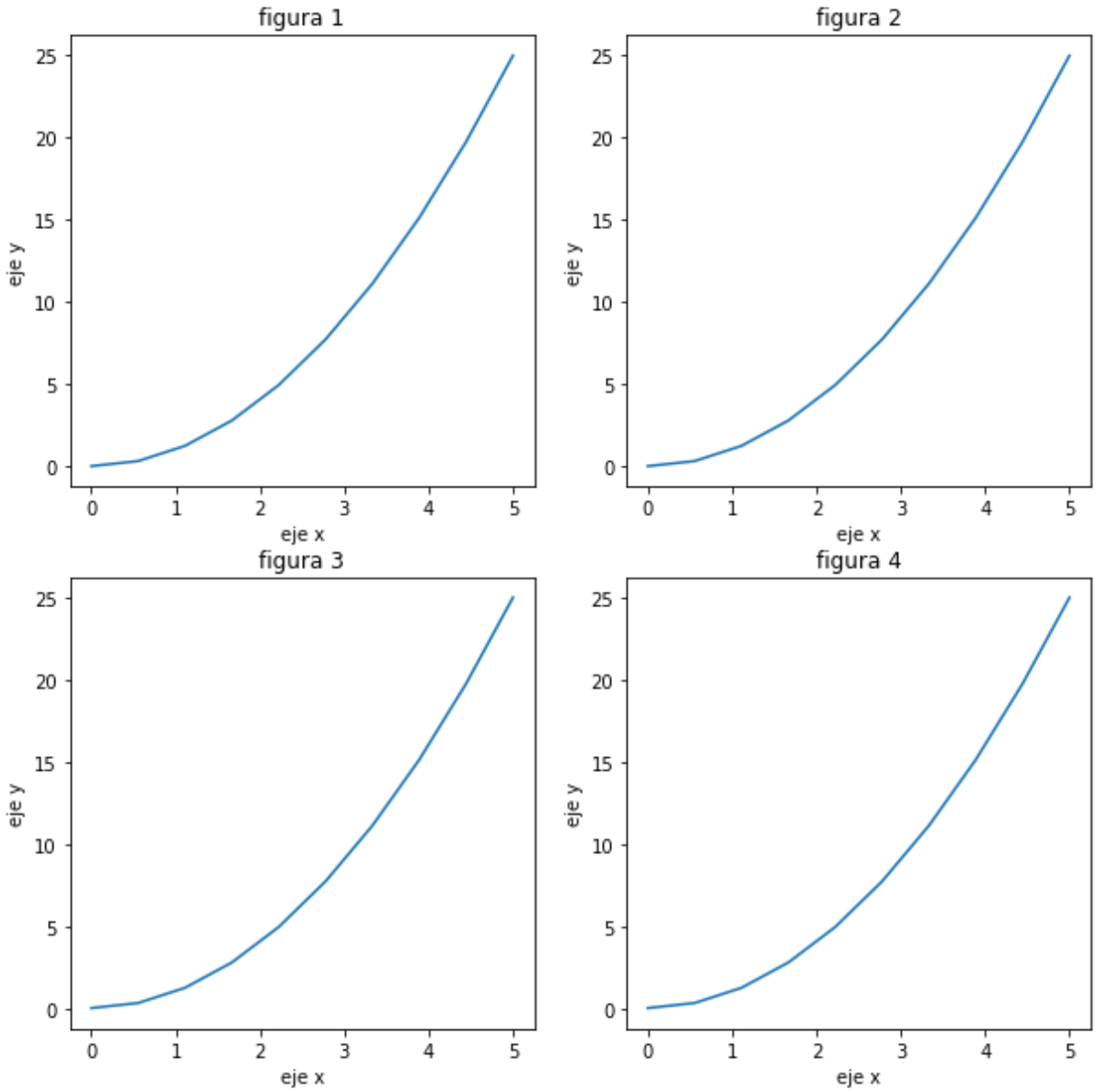
In [27]:
```
# si era mediante un sublpot se podía de la siguiente manera
plt.figure(figsize=(5,5))
f= plt.subplot(2,2,1)
f.set_title("figura 1")
f.set_xlabel("eje x")
f.set_ylabel("eje y")
f.plot(x,y)
g= plt.subplot(2,2,2)
g.set_title("figura 2")
g.set_xlabel("eje x")
g.set_ylabel("eje y")
g.plot(x,y)
```

Out[27]:
```
[<matplotlib.lines.Line2D at 0x111e4c990>]
```



In [25]:
```
#8.5.2
plt.figure(figsize=(10, 10))
f1= plt.subplot(2,2,1)
f1.set_title("figura 1")
f1.set_xlabel("eje x")
f1.set_ylabel("eje y")
f1.plot(x,y)
f2= plt.subplot(2,2,2)
f2.set_title("figura 2")
f2.set_xlabel("eje x")
f2.set_ylabel("eje y")
f2.plot(x,y)
f3= plt.subplot(2,2,3)
f3.set_title("figura 3")
f3.set_xlabel("eje x")
f3.set_ylabel("eje y")
f3.plot(x,y)
f4= plt.subplot(2,2,4)
f4.set_title("figura 4")
f4.set_xlabel("eje x")
f4.set_ylabel("eje y")
f4.plot(x,y)
```

Out[25]: [<matplotlib.lines.Line2D at 0x112ff0810>]



In [26]:
```
#8.5.3
fig = plt.figure()
ax1 = fig.add_axes([0.25, 0.3, 0.9, 0.9])
ax1.plot(x,y)
ax2 = fig.add_axes([0.8, 0.4, 0.2, 0.2])
ax2.plot(x,y)
```

Out[26]:
```
[<matplotlib.lines.Line2D at 0x11287b810>]
```