Taller 6

Métodos Computacionales para Políticas Públicas - URosario

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

Juan Sebastián Gómez

Juansebastian.gomezm@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).

Numpy Practice

In [1]:

```
import numpy as np
import scipy.linalg as la
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
import math
import datetime as dt

# Punto 1
x = 8

# Punto 2
print(np.square(x))
print(np.power(x,3))

# Punto 3

theta = 90

# Punto 4

seno = np.sin(theta)
print("seno: ", seno)
coseno = np.cos(theta)
print("coseno: ", coseno)

#La librería trabaja con grados, pero la operación se puede pasar a radianes. Para
 este caso, la librería está usando grados.
```

```
64
512
seno:  0.8939966636005579
coseno:  -0.4480736161291701
```

Punto 5

In [2]:

```
meshPoints = np.linspace(-1,1,500)
print(meshPoints)
```

```
[-1.         -0.99599198 -0.99198397 -0.98797595 -0.98396794 -0.9799599
2
 -0.9759519  -0.97194389 -0.96793587 -0.96392786 -0.95991984 -0.9559118
2
 -0.95190381 -0.94789579 -0.94388778 -0.93987976 -0.93587174 -0.9318637
3
 -0.92785571 -0.9238477  -0.91983968 -0.91583166 -0.91182365 -0.9078156
3
 -0.90380762 -0.8997996  -0.89579158 -0.89178357 -0.88777555 -0.8837675
4
 -0.87975952 -0.8757515  -0.87174349 -0.86773547 -0.86372745 -0.8597194
4
 -0.85571142 -0.85170341 -0.84769539 -0.84368737 -0.83967936 -0.8356713
4
 -0.83166333 -0.82765531 -0.82364729 -0.81963928 -0.81563126 -0.8116232
5
 -0.80761523 -0.80360721 -0.7995992  -0.79559118 -0.79158317 -0.7875751
5
 -0.78356713 -0.77955912 -0.7755511  -0.77154309 -0.76753507 -0.7635270
5
 -0.75951904 -0.75551102 -0.75150301 -0.74749499 -0.74348697 -0.7394789
6
 -0.73547094 -0.73146293 -0.72745491 -0.72344689 -0.71943888 -0.7154308
6
 -0.71142285 -0.70741483 -0.70340681 -0.6993988  -0.69539078 -0.6913827
7
 -0.68737475 -0.68336673 -0.67935872 -0.6753507  -0.67134269 -0.6673346
7
 -0.66332665 -0.65931864 -0.65531062 -0.65130261 -0.64729459 -0.6432865
7
 -0.63927856 -0.63527054 -0.63126253 -0.62725451 -0.62324649 -0.6192384
8
 -0.61523046 -0.61122244 -0.60721443 -0.60320641 -0.5991984  -0.5951903
8
 -0.59118236 -0.58717435 -0.58316633 -0.57915832 -0.5751503  -0.5711422
8
 -0.56713427 -0.56312625 -0.55911824 -0.55511022 -0.5511022  -0.5470941
9
 -0.54308617 -0.53907816 -0.53507014 -0.53106212 -0.52705411 -0.5230460
9
 -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 -0.4268537
1
 -0.42284569 -0.41883768 -0.41482966 -0.41082164 -0.40681363 -0.4028056
1
 -0.3987976  -0.39478958 -0.39078156 -0.38677355 -0.38276553 -0.3787575
2
 -0.3747495  -0.37074148 -0.36673347 -0.36272545 -0.35871743 -0.3547094
2
 -0.3507014  -0.34669339 -0.34268537 -0.33867735 -0.33466934 -0.3306613
2
 -0.32665331 -0.32264529 -0.31863727 -0.31462926 -0.31062124 -0.3066132
3
 -0.30260521 -0.29859719 -0.29458918 -0.29058116 -0.28657315 -0.2825651
3
```

```
 -0.27855711 -0.2745491  -0.27054108 -0.26653307 -0.26252505 -0.2585170
3
 -0.25450902 -0.250501   -0.24649299 -0.24248497 -0.23847695 -0.2344689
4
 -0.23046092 -0.22645291 -0.22244489 -0.21843687 -0.21442886 -0.2104208
4
 -0.20641283 -0.20240481 -0.19839679 -0.19438878 -0.19038076 -0.1863727
5
 -0.18236473 -0.17835671 -0.1743487  -0.17034068 -0.16633267 -0.1623246
5
 -0.15831663 -0.15430862 -0.1503006  -0.14629259 -0.14228457 -0.1382765
5
 -0.13426854 -0.13026052 -0.12625251 -0.12224449 -0.11823647 -0.1142284
6
 -0.11022044 -0.10621242 -0.10220441 -0.09819639 -0.09418838 -0.0901803
6
 -0.08617234 -0.08216433 -0.07815631 -0.0741483  -0.07014028 -0.0661322
6
 -0.06212425 -0.05811623 -0.05410822 -0.0501002  -0.04609218 -0.0420841
7
 -0.03807615 -0.03406814 -0.03006012 -0.0260521  -0.02204409 -0.0180360
7
 -0.01402806 -0.01002004 -0.00601202 -0.00200401  0.00200401  0.0060120
2
  0.01002004  0.01402806  0.01803607  0.02204409  0.0260521   0.0300601
2
  0.03406814  0.03807615  0.04208417  0.04609218  0.0501002   0.0541082
2
  0.05811623  0.06212425  0.06613226  0.07014028  0.0741483   0.0781563
1
  0.08216433  0.08617234  0.09018036  0.09418838  0.09819639  0.1022044
1
  0.10621242  0.11022044  0.11422846  0.11823647  0.12224449  0.1262525
1
  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  0.1983967
9
  0.20240481  0.20641283  0.21042084  0.21442886  0.21843687  0.2224448
9
  0.22645291  0.23046092  0.23446894  0.23847695  0.24248497  0.2464929
9
  0.250501    0.25450902  0.25851703  0.26252505  0.26653307  0.2705410
8
  0.2745491   0.27855711  0.28256513  0.28657315  0.29058116  0.2945891
8
  0.29859719  0.30260521  0.30661323  0.31062124  0.31462926  0.3186372
7
  0.32264529  0.32665331  0.33066132  0.33466934  0.33867735  0.3426853
7
  0.34669339  0.3507014   0.35470942  0.35871743  0.36272545  0.3667334
7
  0.37074148  0.3747495   0.37875752  0.38276553  0.38677355  0.3907815
6
  0.39478958  0.3987976   0.40280561  0.40681363  0.41082164  0.4148296
6
  0.41883768  0.42284569  0.42685371  0.43086172  0.43486974  0.4388777
```

```
6
  0.44288577  0.44689379  0.4509018   0.45490982  0.45891784  0.4629258
5
  0.46693387  0.47094188  0.4749499   0.47895792  0.48296593  0.4869739
5
  0.49098196  0.49498998  0.498998    0.50300601  0.50701403  0.5110220
4
  0.51503006  0.51903808  0.52304609  0.52705411  0.53106212  0.5350701
4
  0.53907816  0.54308617  0.54709419  0.5511022   0.55511022  0.5591182
4
  0.56312625  0.56713427  0.57114228  0.5751503   0.57915832  0.5831663
3
  0.58717435  0.59118236  0.59519038  0.5991984   0.60320641  0.6072144
3
  0.61122244  0.61523046  0.61923848  0.62324649  0.62725451  0.6312625
3
  0.63527054  0.63927856  0.64328657  0.64729459  0.65130261  0.6553106
2
  0.65931864  0.66332665  0.66733467  0.67134269  0.6753507   0.6793587
2
  0.68336673  0.68737475  0.69138277  0.69539078  0.6993988   0.7034068
1
  0.70741483  0.71142285  0.71543086  0.71943888  0.72344689  0.7274549
1
  0.73146293  0.73547094  0.73947896  0.74348697  0.74749499  0.7515030
1
  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  0.8236472
9
  0.82765531  0.83166333  0.83567134  0.83967936  0.84368737  0.8476953
9
  0.85170341  0.85571142  0.85971944  0.86372745  0.86773547  0.8717434
9
  0.8757515   0.87975952  0.88376754  0.88777555  0.89178357  0.8957915
8
  0.8997996   0.90380762  0.90781563  0.91182365  0.91583166  0.9198396
8
  0.9238477   0.92785571  0.93186373  0.93587174  0.93987976  0.9438877
8
  0.94789579  0.95190381  0.95591182  0.95991984  0.96392786  0.9679358
7
  0.97194389  0.9759519   0.97995992  0.98396794  0.98797595  0.9919839
7
  0.99599198  1.        ]
```

Punto 6

In [3]:

```
print(meshPoints[52]) #Esta es la expresión que arrojará el valor. Lo que imprima es el valor
```

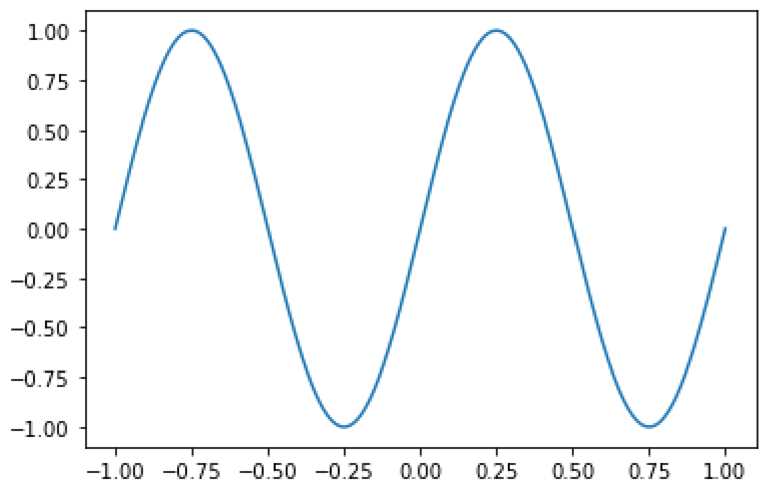
```
-0.7915831663326653
```

Punto 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.

In [4]:

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

plt.savefig("punto7-taller6.jpg")
```




```
<Figure size 432x288 with 0 Axes>
```

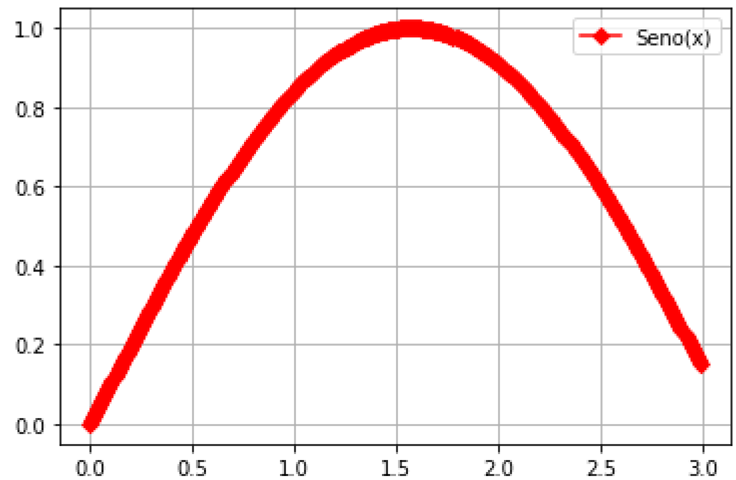
Resuelva los ejercicios de las secciones 4.1, 5.1, 6.1, 7.4 y 8.5 de este documento (http://www.python-academy.com/download/pycon2012/matplotlib_handout.pdf).

Punto 4.1

1. Plot a simple graph of a sinus function in the range 0 to 3 with a step size of 0.01.
2. Make the line red. Add diamond-shaped markers with size of 5.
3. Add a legend and a grid to the plot.

In [5]:

```
import numpy as np
import scipy.linalg as la
import matplotlib.pyplot as plt

x = np.arange(0.0,3.0,0.01)
plt.plot(x, np.sin(x), marker = 'D', color ='red', markersize = 5, label = 'Seno
(x)')
plt.legend()
plt.grid(True)
```

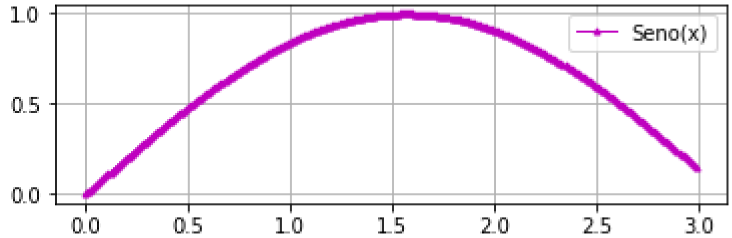


Punto 5.1

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.

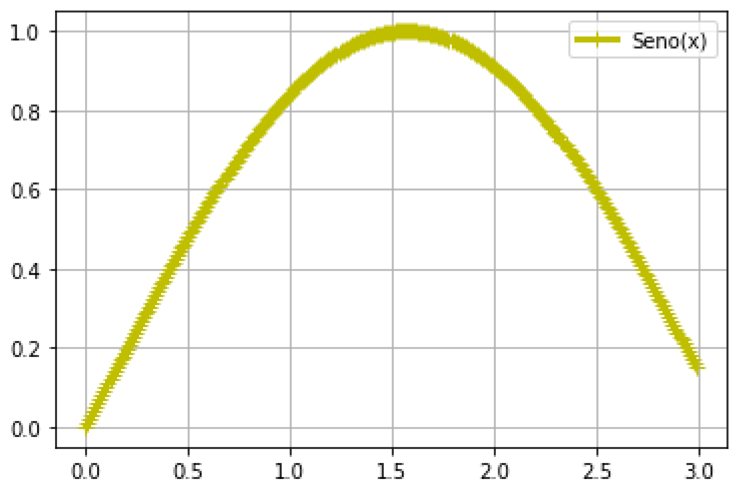
In [6]:

```
x = np.arange(0.0,3.0,0.01)
y = np.sin(x)

plt.subplot(2,1,2)
plt.plot(x, y, marker = '^', color ='m', markersize = 3, linewidth = 1 , label = 'S
eno(x)')
plt.legend()
plt.grid(True)
```



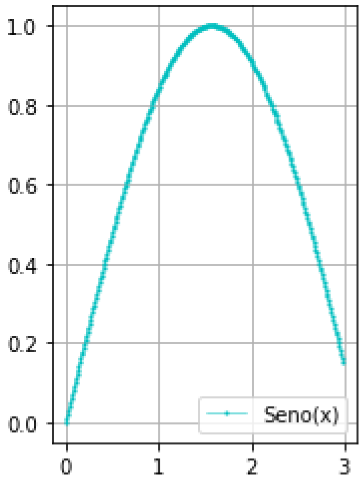
In [7]:

```
plt.plot(x, y, marker = '+', color ='y', markersize = 8,  linewidth = 3 , label =
'Seno(x)')
plt.legend()
plt.grid(True)
```



In [8]:

```
plt.subplot(1,2,1)
plt.plot(x, y, marker = 's', color ='c', markersize = 1, linewidth = 0.5, label =
'Seno(x)')
plt.legend()
plt.grid(True)
```

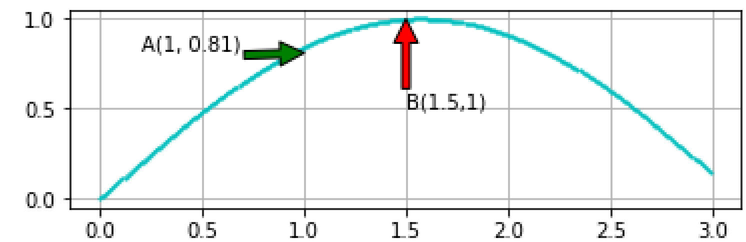


Punto 6.1

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

In [9]:

```
x = np.arange(0.0,3.0,0.01)
y = np.sin(x)

plt.subplot(2,1,2)
plt.plot(x, y, marker = 's', color ='c', markersize = 1, linewidth = 0.5)
plt.annotate('A(1, 0.81)', xy=(1,0.81), xytext=(0.2, 0.82), arrowprops=dict(facecol
or='green'))
plt.annotate('B(1.5,1)', xy=(1.5,1), xytext=(1.5, 0.5),arrowprops=dict(facecolor='r
ed'))
plt.grid(True)
```



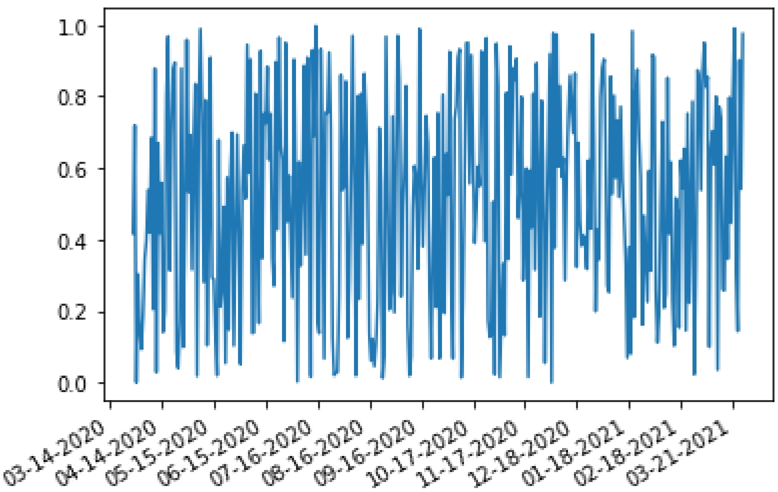
Punto 7.4

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

Nota: Puse un intervalo de 30 días(1 mes), para que pudiesen observarse las fechas. de lo contrario si dejaba el intervalo en 1, era imposible leer algo en el eje x.

In [10]:

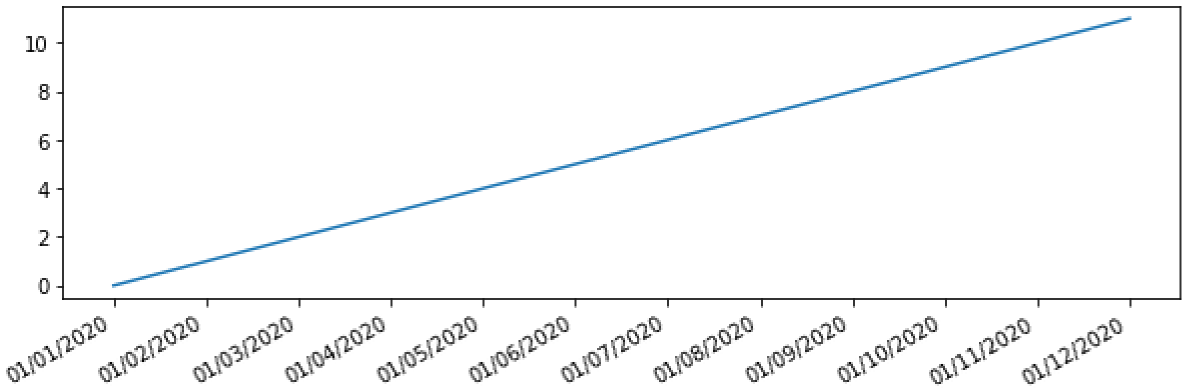
```
np.random.seed(1)

N = 365
y = np.random.rand(N)

hoy = dt.datetime.now()
mañana = hoy + dt.timedelta(days=365)
days = mdates.drange(hoy,mañana,dt.timedelta(days=1))

plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m-%d-%Y'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=31))
plt.plot(days,y)
plt.gcf().autofmt_xdate()
plt.rcParams['figure.figsize'] = [10.0, 3.0]
plt.show()
```



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

In [11]:

```
dates = ['01/01/2020','01/02/2020','01/03/2020','01/04/2020','01/05/2020','01/06/20
20','01/07/2020','01/08/2020','01/09/2020','01/10/2020','01/11/2020','01/12/2020']
x = [dt.datetime.strptime(d,'%m/%d/%Y').date() for d in dates]
z = range(len(x))
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m/%d/%Y'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator())
plt.plot(x,z)
plt.gcf().autofmt_xdate()
```



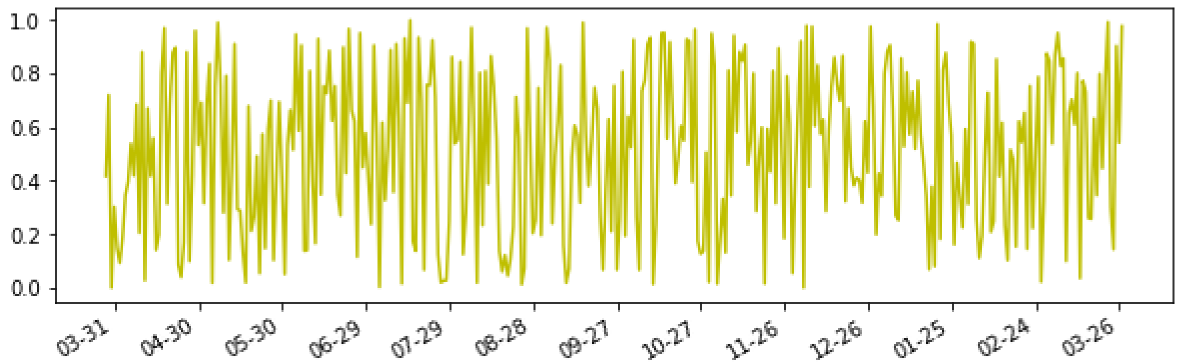
1. Display the dates with and without the year. Show the month as number and as first three letters of the month name.

En el primer gráfico del punto 7.4 se mostrararon las fechas con año.

En este gráfico se muestran las fechas sin año

In [12]:

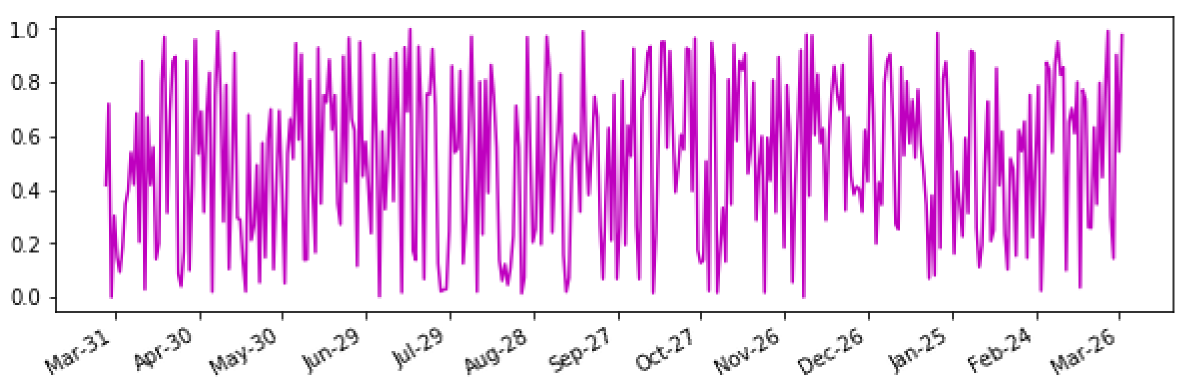
```
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%m-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=30))
plt.plot(days,y, color = 'y')
plt.gcf().autofmt_xdate()
plt.rcParams['figure.figsize'] = [10.0, 3.0]
plt.show()
```



En este gráfico se muestran las fechas con las primeras 3 letras del nombre del mes, porque en el gráfico anterior se mostraron los meses en números

In [13]:

```
plt.gca().xaxis.set_major_formatter(mdates.DateFormatter('%b-%d'))
plt.gca().xaxis.set_major_locator(mdates.DayLocator(interval=30))
plt.plot(days,y, color = 'm')
plt.gcf().autofmt_xdate()
plt.rcParams['figure.figsize'] = [10.0, 3.0]
plt.show()
```



Punto 8.5

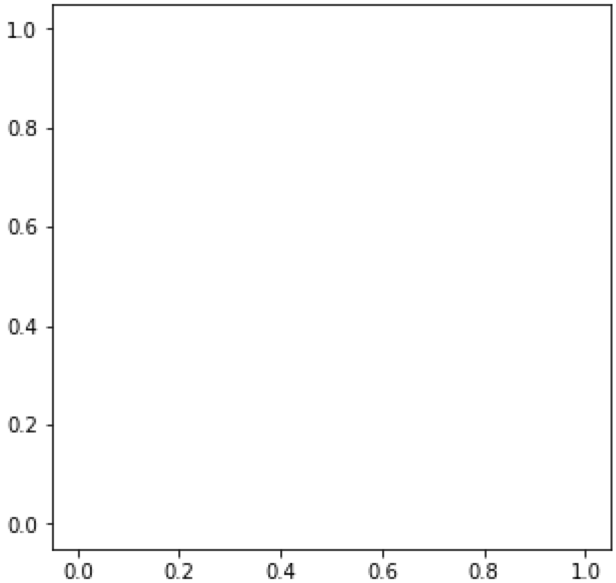
1. Dibuja 2 figuras. Una de 5x5 y otra de 10x10

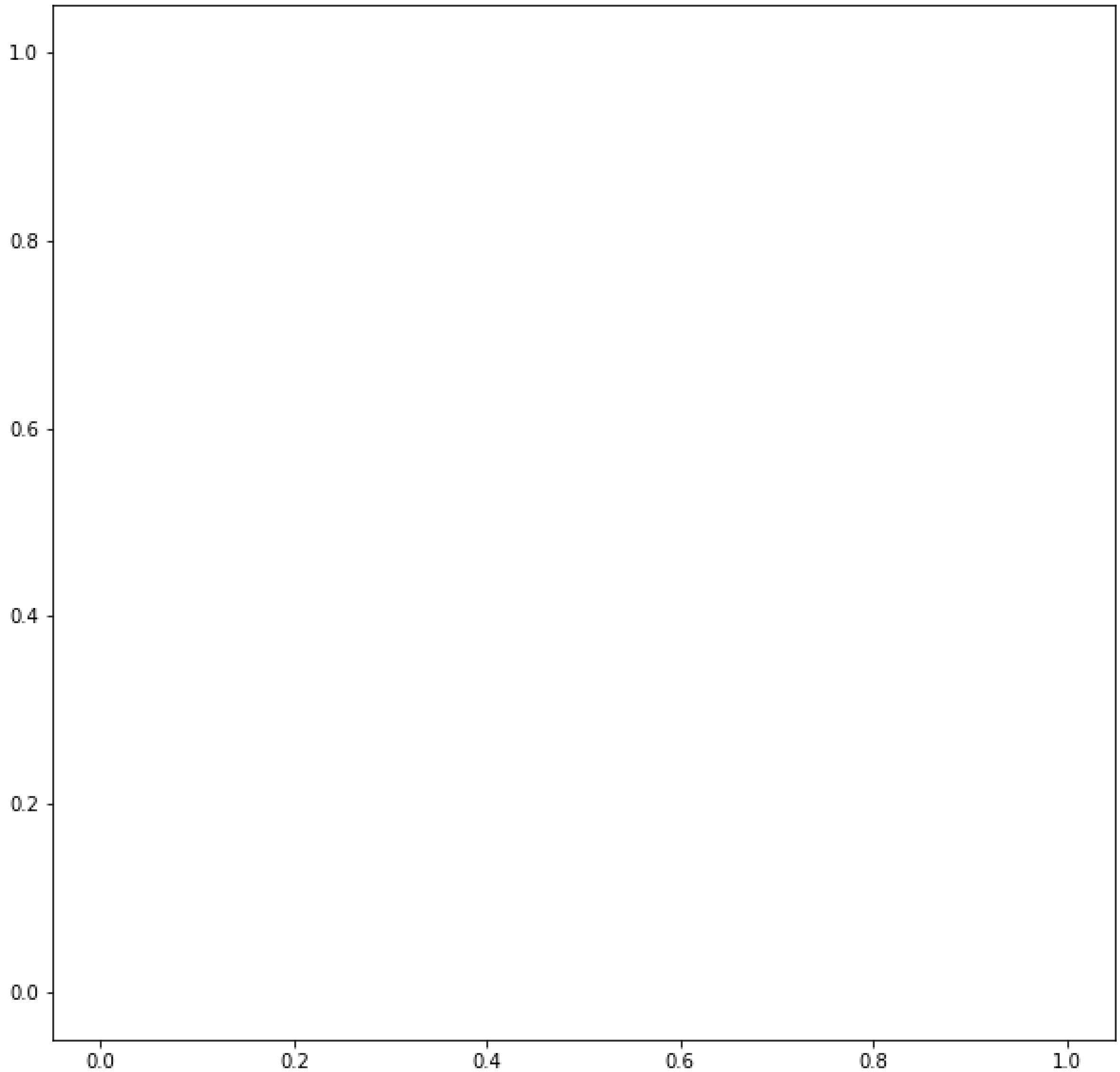
In [14]:

```
plt.figure(figsize=(5,5))
plt.plot()
plt.figure(figsize=(10,10))
plt.plot()
```

Out[14]:

```
[]
```

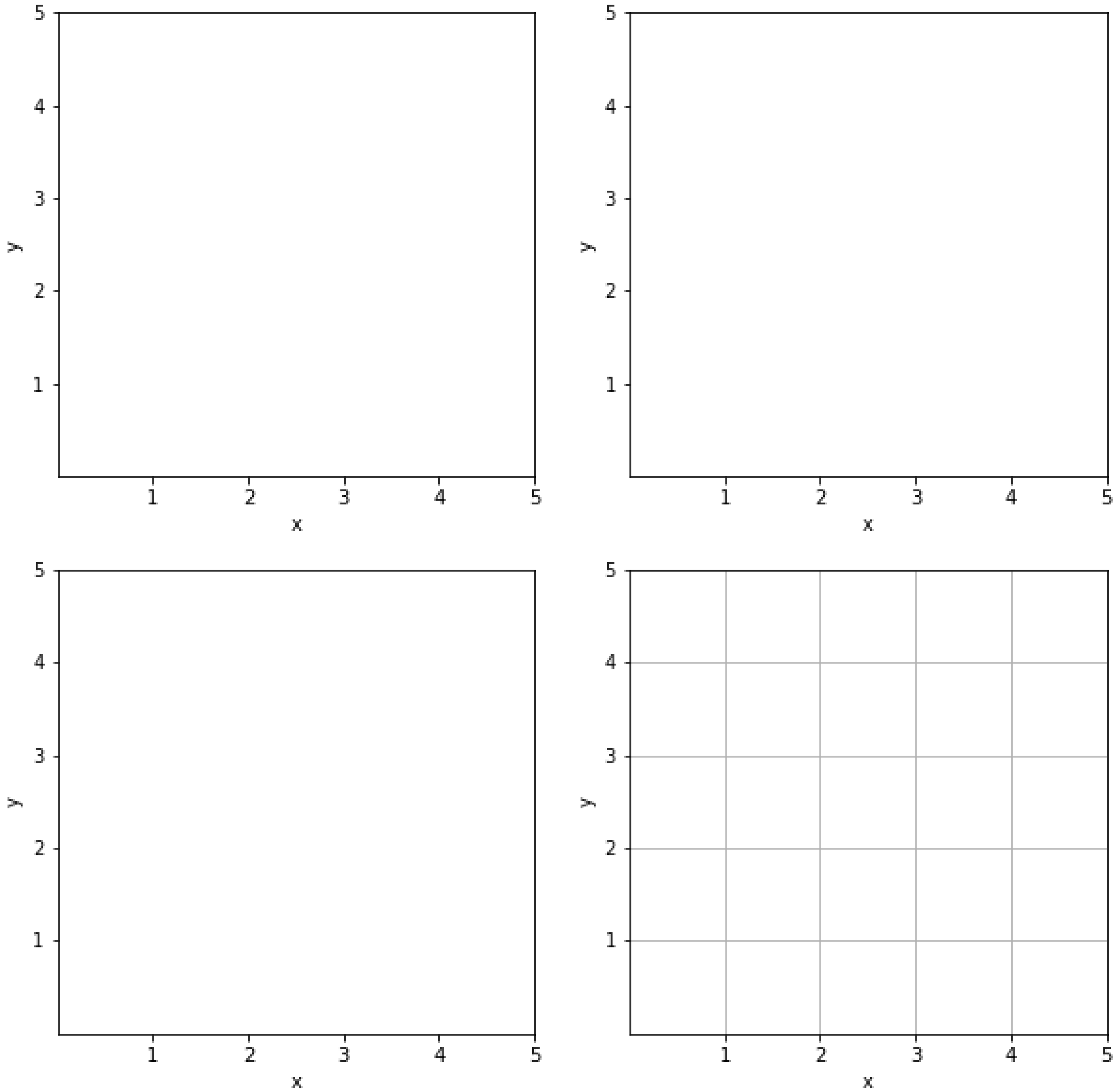



1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

1. Agrega 4 subplots a una figura.

In [15]:

```
plt.figure(figsize=(10,10))
plt.subplot(221)
plt.xlabel('x')
plt.ylabel('y')
plt.xticks([1,2,3,4,5])
plt.yticks([1,2,3,4,5])
plt.subplot(222)
plt.xlabel('x')
plt.ylabel('y')
plt.xticks([1,2,3,4,5])
plt.yticks([1,2,3,4,5])
plt.subplot(223)
plt.xlabel('x')
plt.ylabel('y')
plt.xticks([1,2,3,4,5])
plt.yticks([1,2,3,4,5])
plt.subplot(224)
plt.xlabel('x')
plt.ylabel('y')
plt.xticks([1,2,3,4,5])
plt.yticks([1,2,3,4,5])
plt.grid(True)
```



y
x
1
2
3
4
5

1. Ubica un gráfico pequeño dentro de uno más grande

In [16]:

```
figura = plt.figure()
a = figura.add_axes([1, 1, 1, 1])
b = figura.add_axes([1.5, 1.5, 0.4, 0.4])
```