Taller 6

Métodos Computacionales para Políticas Públicas - URosario

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

[Juan Diego Castro Rodríguez]

[juand.castro@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.

In [1]:
```
import numpy as np
import scipy.linalg as la
import matplotlib.pyplot as plt
```

In [2]:
```
X=5
```

In [7]:
```
s=np.square(X)
print(s)
```

```
25
```

In [8]:
```
theta=25
```

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

```
-0.13235175009777303
```

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

```
0.9912028118634736
```

In [18]:
```
meshPoints=np.linspace(1,-1, num=501)
```

In [21]:
```
print(meshPoints[52])
```

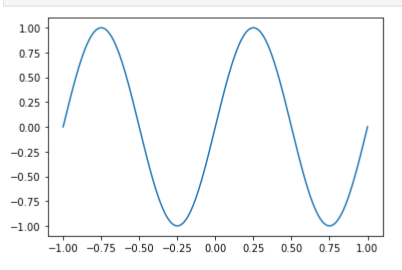
```
0.792
```

In [22]:
```
print(meshPoints)
```

```
[ 1.     0.996  0.992  0.988  0.984  0.98   0.976  0.972  0.968  0.964
  0.96   0.956  0.952  0.948  0.944  0.94   0.936  0.932  0.928  0.924
  0.92   0.916  0.912  0.908  0.904  0.9    0.896  0.892  0.888  0.884
  0.88   0.876  0.872  0.868  0.864  0.86   0.856  0.852  0.848  0.844
  0.84   0.836  0.832  0.828  0.824  0.82   0.816  0.812  0.808  0.804
  0.8    0.796  0.792  0.788  0.784  0.78   0.776  0.772  0.768  0.764
  0.76   0.756  0.752  0.748  0.744  0.74   0.736  0.732  0.728  0.724
  0.72   0.716  0.712  0.708  0.704  0.7    0.696  0.692  0.688  0.684
  0.68   0.676  0.672  0.668  0.664  0.66   0.656  0.652  0.648  0.644
  0.64   0.636  0.632  0.628  0.624  0.62   0.616  0.612  0.608  0.604
  0.6    0.596  0.592  0.588  0.584  0.58   0.576  0.572  0.568  0.564
  0.56   0.556  0.552  0.548  0.544  0.54   0.536  0.532  0.528  0.524
  0.52   0.516  0.512  0.508  0.504  0.5    0.496  0.492  0.488  0.484
  0.48   0.476  0.472  0.468  0.464  0.46   0.456  0.452  0.448  0.444
  0.44   0.436  0.432  0.428  0.424  0.42   0.416  0.412  0.408  0.404
  0.4    0.396  0.392  0.388  0.384  0.38   0.376  0.372  0.368  0.364
  0.36   0.356  0.352  0.348  0.344  0.34   0.336  0.332  0.328  0.324
  0.32   0.316  0.312  0.308  0.304  0.3    0.296  0.292  0.288  0.284
  0.28   0.276  0.272  0.268  0.264  0.26   0.256  0.252  0.248  0.244
  0.24   0.236  0.232  0.228  0.224  0.22   0.216  0.212  0.208  0.204
  0.2    0.196  0.192  0.188  0.184  0.18   0.176  0.172  0.168  0.164
  0.16   0.156  0.152  0.148  0.144  0.14   0.136  0.132  0.128  0.124
  0.12   0.116  0.112  0.108  0.104  0.1    0.096  0.092  0.088  0.084
  0.08   0.076  0.072  0.068  0.064  0.06   0.056  0.052  0.048  0.044
  0.04   0.036  0.032  0.028  0.024  0.02   0.016  0.012  0.008  0.004
  0.    -0.004 -0.008 -0.012 -0.016 -0.02  -0.024 -0.028 -0.032 -0.036
 -0.04  -0.044 -0.048 -0.052 -0.056 -0.06  -0.064 -0.068 -0.072 -0.076
 -0.08  -0.084 -0.088 -0.092 -0.096 -0.1   -0.104 -0.108 -0.112 -0.116
 -0.12  -0.124 -0.128 -0.132 -0.136 -0.14  -0.144 -0.148 -0.152 -0.156
 -0.16  -0.164 -0.168 -0.172 -0.176 -0.18  -0.184 -0.188 -0.192 -0.196
 -0.2   -0.204 -0.208 -0.212 -0.216 -0.22  -0.224 -0.228 -0.232 -0.236
 -0.24  -0.244 -0.248 -0.252 -0.256 -0.26  -0.264 -0.268 -0.272 -0.276
 -0.28  -0.284 -0.288 -0.292 -0.296 -0.3   -0.304 -0.308 -0.312 -0.316
 -0.32  -0.324 -0.328 -0.332 -0.336 -0.34  -0.344 -0.348 -0.352 -0.356
 -0.36  -0.364 -0.368 -0.372 -0.376 -0.38  -0.384 -0.388 -0.392 -0.396
 -0.4   -0.404 -0.408 -0.412 -0.416 -0.42  -0.424 -0.428 -0.432 -0.436
 -0.44  -0.444 -0.448 -0.452 -0.456 -0.46  -0.464 -0.468 -0.472 -0.476
 -0.48  -0.484 -0.488 -0.492 -0.496 -0.5   -0.504 -0.508 -0.512 -0.516
 -0.52  -0.524 -0.528 -0.532 -0.536 -0.54  -0.544 -0.548 -0.552 -0.556
 -0.56  -0.564 -0.568 -0.572 -0.576 -0.58  -0.584 -0.588 -0.592 -0.596
 -0.6   -0.604 -0.608 -0.612 -0.616 -0.62  -0.624 -0.628 -0.632 -0.636
 -0.64  -0.644 -0.648 -0.652 -0.656 -0.66  -0.664 -0.668 -0.672 -0.676
 -0.68  -0.684 -0.688 -0.692 -0.696 -0.7   -0.704 -0.708 -0.712 -0.716
 -0.72  -0.724 -0.728 -0.732 -0.736 -0.74  -0.744 -0.748 -0.752 -0.756
 -0.76  -0.764 -0.768 -0.772 -0.776 -0.78  -0.784 -0.788 -0.792 -0.796
 -0.8   -0.804 -0.808 -0.812 -0.816 -0.82  -0.824 -0.828 -0.832 -0.836
 -0.84  -0.844 -0.848 -0.852 -0.856 -0.86  -0.864 -0.868 -0.872 -0.876
 -0.88  -0.884 -0.888 -0.892 -0.896 -0.9   -0.904 -0.908 -0.912 -0.916
 -0.92  -0.924 -0.928 -0.932 -0.936 -0.94  -0.944 -0.948 -0.952 -0.956
 -0.96  -0.964 -0.968 -0.972 -0.976 -0.98  -0.984 -0.988 -0.992 -0.996
 -1.   ]
```

In [25]:
```
import math
pi=math.pi
```

In [27]:
```
plt.plot(meshPoints,np.sin(2*pi*meshPoints))
plt.savefig('sinusoid.png')
```

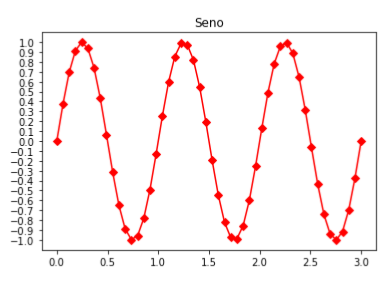


Resuelva los ejercicios de las secciones 4.1, 5.1, 6.1, 7.4 y 8.5 de este documento.

4.1

In [81]:
```
puntos=np.linspace(0, 3, num=50)
amplitud= np.sin(2*pi*time)
plt.plot(puntos, amplitud, "r", marker = 'D', ms=5);
plt.yticks(np.arange(min(amplitud), max(amplitud)+0.1, 0.1))
plt.title('Seno')
```

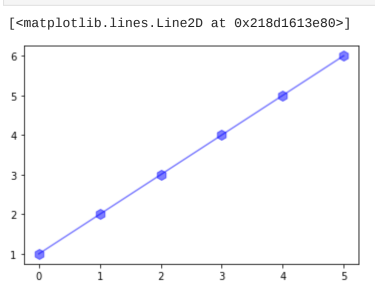
Out[81]:
```
Text(0.5, 1.0, 'Seno')
```



5.1

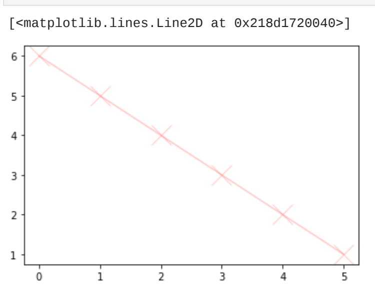
In [89]:
```
px=[1,2,3,4,5,6]
py=[6,5,4,3,2,1]
plt.plot(px, "b", marker='h', ms=10, alpha=0.5)
```

Out[89]:
```
[<matplotlib.lines.Line2D at 0x218d1613e80>]
```



In [93]:
```
plt.plot(py, "r", marker='x', ms=20, alpha=0.2)
```

Out[93]:
```
[<matplotlib.lines.Line2D at 0x218d1720040>]
```

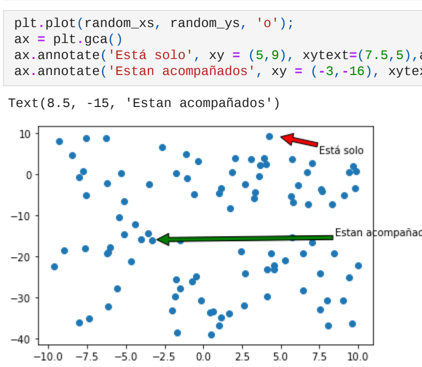


6.1

In [36]:
```
import random
random_xs = [random.uniform(-10,10) for x in range(100)]
random_ys = [random.uniform(-40,10) for x in range(100)]
```

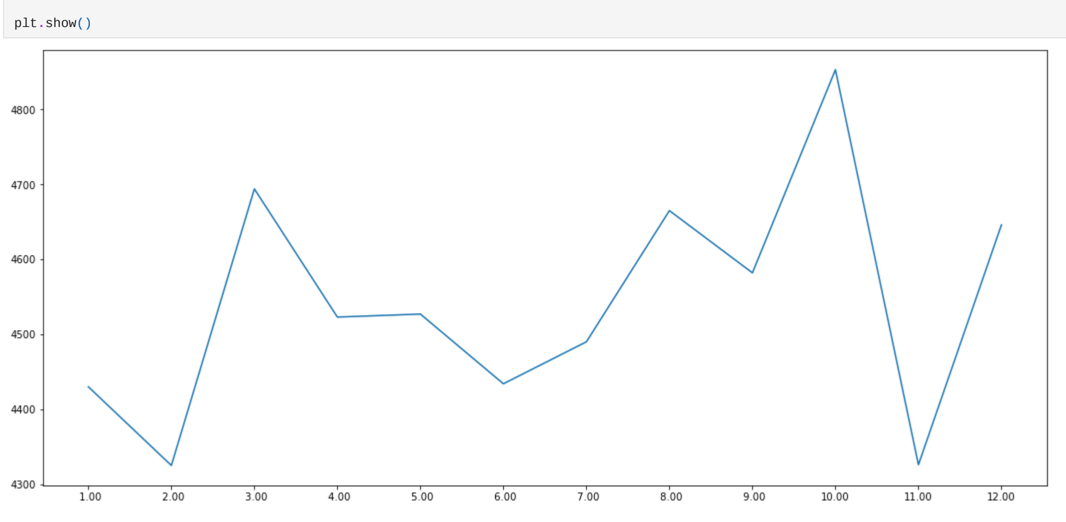
In [61]:
```
plt.plot(random_xs, random_ys, 'o');
ax = plt.gca()
ax.annotate('Está solo', xy = (5,9), xytext=(7.5,5),arrowprops={'facecolor': 'r'})
ax.annotate('Estan acompañados', xy = (-3,-16), xytext=(8.5,-15),arrowprops={'facecolor': 'g'})
```

Out[61]:
```
Text(8.5, -15, 'Estan acompañados')
```

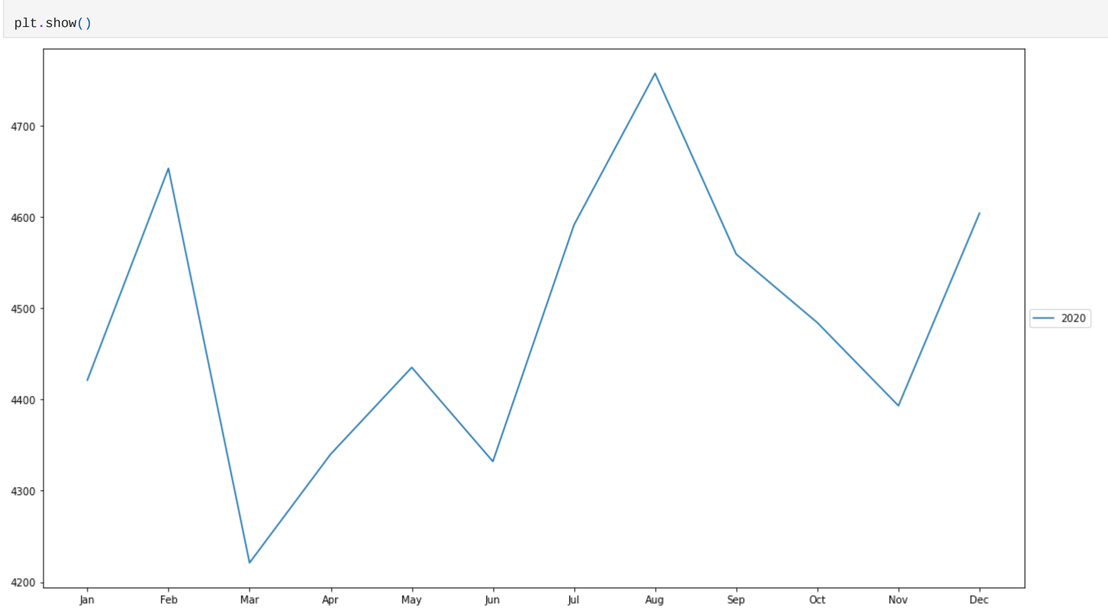


7.4

In [63]:
```
import datetime
import matplotlib.ticker as ticker

dates_d = pd.date_range('2020-01-01', '2020-12-31', freq='D')

ax = plt.figure().add_subplot(111)
ax.plot(pt)

plt.major_locator = plt.MultipleLocator(1)
plt.major_formatter = plt.FormatStrFormatter('%5.2f')
plt.minor_locator = plt.MultipleLocator(1)
ax.xaxis.set_major_locator(plt.major_locator)
ax.xaxis.set_minor_locator(plt.minor_locator)
ax.xaxis.set_major_formatter(plt.major_formatter)

plt.show()
```



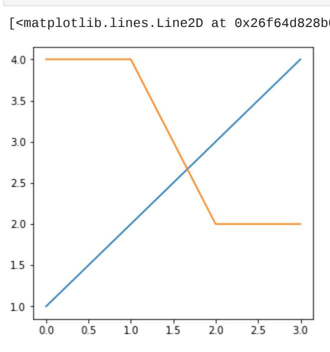
In [67]:
```
dates_d = pd.date_range('2020-01-01', '2020-12-31', freq='D')

ax = plt.figure().add_subplot(111)
ax.plot(pt)

ticklabels = [datetime.date(1900, item, 1).strftime('%b') for item in pt.index]
ax.set_xticks(np.arange(1,13))
ax.set_xticklabels(ticklabels)

ax.legend(pt.columns.tolist(), loc='center left', bbox_to_anchor=(1, .5))
plt.tight_layout(rect=[0, 0, 0.85, 1])

plt.show()
```



8.5

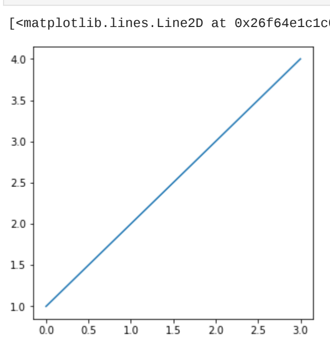
In [91]:
```
from matplotlib.pyplot import figure
datos=[1,2,3,4]
imagen=[4,4,2,2]
plt.plot(datos)
plt.plot(imagen)
```

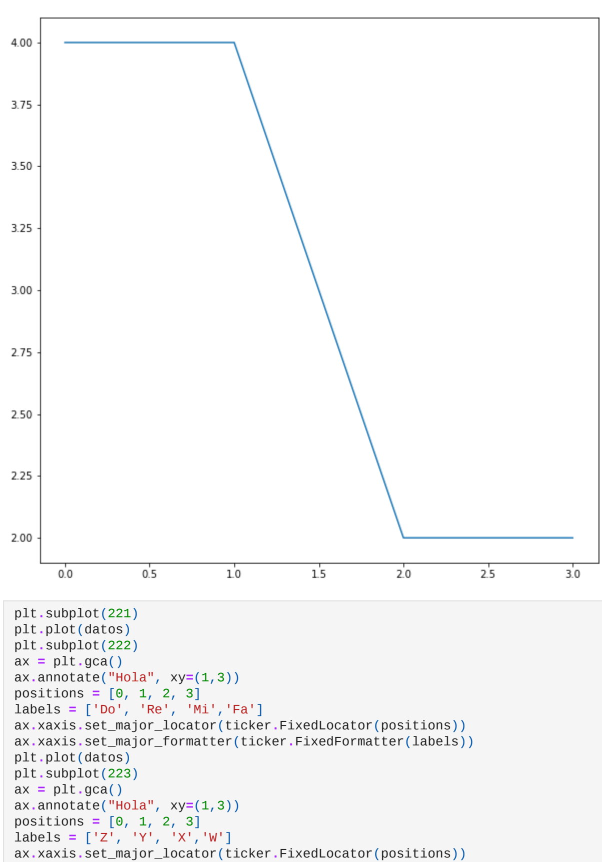
Out[91]:
```
[<matplotlib.lines.Line2D at 0x26f64d828b0>]
```



In [92]:
```
plt.figure(figsize=(5, 5))
plt.plot(datos)

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

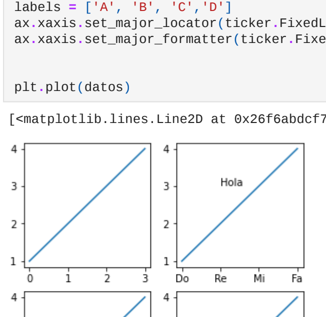
Out[92]:
```
[<matplotlib.lines.Line2D at 0x26f64e1c1c0>]
```





In [150…
```
plt.subplot(221)
plt.plot(datos)
plt.subplot(222)
ax = plt.gca()
ax.annotate("Hola", xy=(1,3))
positions = [0, 1, 2, 3]
labels = ['Do', 'Re', 'Mi','Fa']
ax.xaxis.set_major_locator(ticker.FixedLocator(positions))
ax.xaxis.set_major_formatter(ticker.FixedFormatter(labels))
plt.plot(datos)
plt.subplot(223)
ax = plt.gca()
ax.annotate("Hola", xy=(1,3))
positions = [0, 1, 2, 3]
labels = ['Z', 'Y', 'X','W']
ax.xaxis.set_major_locator(ticker.FixedLocator(positions))
ax.xaxis.set_major_formatter(ticker.FixedFormatter(labels))
plt.plot(datos)
plt.subplot(224)
ax = plt.gca()
ax.annotate("Hola", xy=(1,3))
positions = [0, 1, 2, 3]
labels = ['A', 'B', 'C','D']
ax.xaxis.set_major_locator(ticker.FixedLocator(positions))
ax.xaxis.set_major_formatter(ticker.FixedFormatter(labels))


plt.plot(datos)
```

Out[150…
```
[<matplotlib.lines.Line2D at 0x26f6abdcf70>]
```



In [163…
```
plt.plot(imagen)
a = plt.axes([0.55, 0.51, 0.3, 0.25])
plt.plot(imagen)
a = plt.axes([0.18, 0.25, 0.3, 0.25])
plt.plot(imagen)
```

Out[163…
```
[<matplotlib.lines.Line2D at 0x26f6a9f1460>]
```