Taller 6

Métodos Computacionales para Políticas Públicas - URosario

Entrega: viernes 16-oct-2020 11:59 PM

[Ivonne Paola Ubaque Galán]

[ivonne.ubaque@urosario.edu.co (mailto:ivonne.ubaque@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: Descárguelo en PDF. Si tiene algún problema con la conversión, descárguelo en HTML. 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.)

Parte I

Exercise 7: Numpy practice

In [16]:
```
import math
import numpy as np
import scipy.linalg as la
import matplotlib.pyplot as plt
```

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

In [23]:
```
m1 = "-" * 8
x = 19
```

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

In [24]:

```
print("Square of {x} is {square}, Cube is {cube}".format(x=x, square=x ** 2, cube
print(m1)
```

```
Square of 19 is 361, Cube is 6859
--------
```

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

In [25]:

```
teta = 8 *(math.pi /150)
```

What is sin θ? cos θ? Angles can be measured in degrees or radians.

Which of these are being used used?

In [26]:

```
print("θ = {} radian, Sin of θ = {}, Cos of θ = {}".format(teta, math.sin(teta),
# print("ϑ = {} radian, Sin of ϑ = {}, Cos of ϑ = {}".format(teta, np.sin(teta),
print(m1)
```

```
θ = 0.16755160819145562 radian, Sin of θ = 0.16676874671610226, Cos of θ = 0.98
5996037070505
--------
```

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.

In [34]:
```
meshpoints = np.linspace(-1, 1, 500)
print("500 values with values evenly spaced between -1 and 1\n", meshpoints)
print(m1)
```

```
500 values with values evenly spaced between -1 and 1
 [-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.        ]
--------
```

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

What is this value?

```
In [36]: print("El puesto 53 de la lista de valores es {}".format(meshpoints[52]))
         print(m1)
```

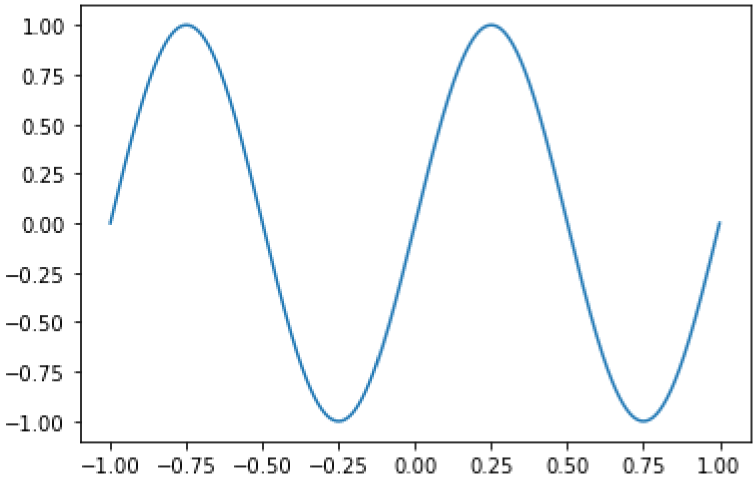
```
El puesto 53 de la lista de valores es -0.7915831663326653
--------
```

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 [38]:

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



7.1

In [39]:

```
plt.savefig("sin.png")
```

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

Parte II

Sección 4.1

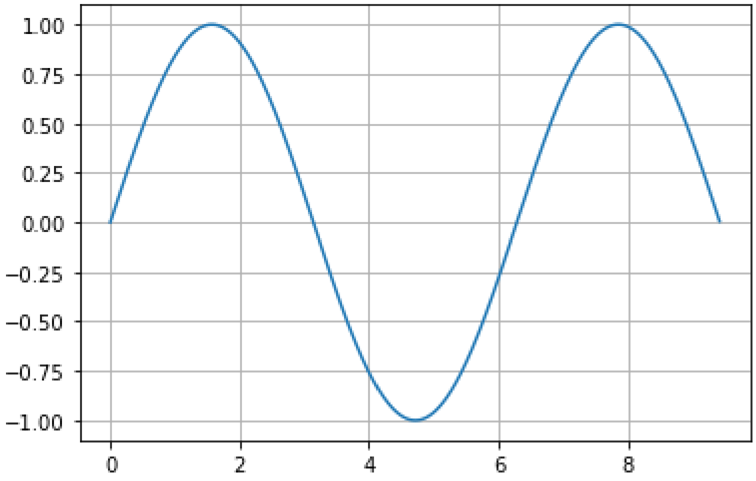
1. Plot a simple graph of a sinus function in the range 0 to 3 with a step size of 0.01.

In [49]:

```
import matplotlib.pyplot as plt
from numpy import arange,sin,pi
```

In [78]:

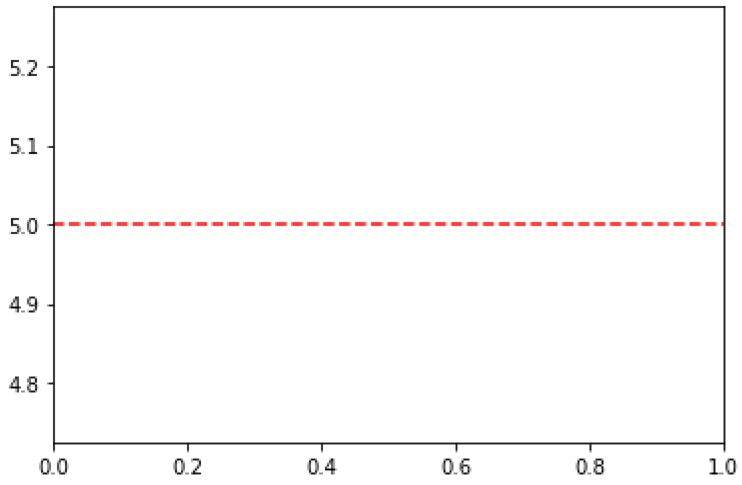
```
x=arange(0.0,3*pi,0.01)
y=sin(x)

plt.plot(x,y)
plt.grid (axis = 'both')
plt.show()
```



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

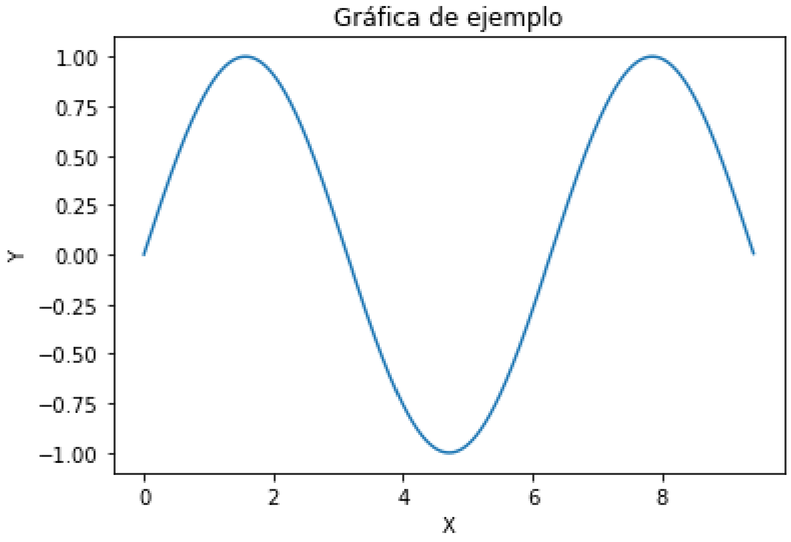
```
In [76]: plt.axhline(5, color="red", linestyle="--")

         ## Profesor, no pude dar con la convención de diamante, lo que "googlee" indicado
         ## y no logré detectar cual fue mi falla. Propongo línea punteada :)
```

```
Out[76]: <matplotlib.lines.Line2D at 0x1eae7fd1ac0>
```



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

```
In [80]: plt.title("Gráfica de ejemplo")
         plt.ylabel('Y')
         plt.xlabel('X')
         plt.plot(x, y);
```

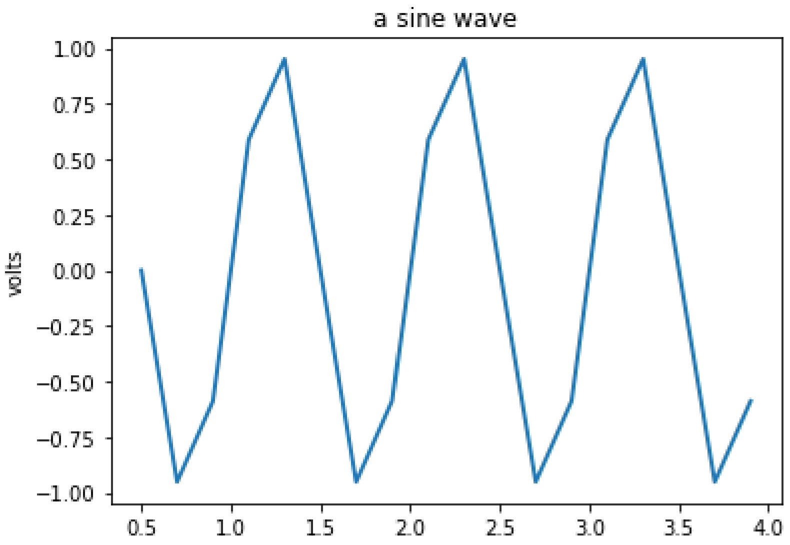


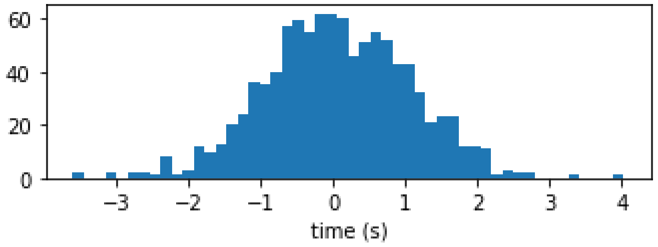
5.1 Exercise

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 [92]:

```
import numpy as np
import matplotlib.pyplot as plt

grafica = plt.figure()
grafica.subplots_adjust(top=1.9)
ax1 = grafica.add_subplot(211)
ax1.set_ylabel('volts')
ax1.set_title('a sine wave')

t = np.arange(0.5, 4.0, 0.2)
s = np.sin(2 * np.pi * t)
line, = ax1.plot(t, s, lw=2)

np.random.seed(19680801)

ax2 = grafica.add_axes([0.15, 0.1, 0.7, 0.3])
n, bins, patches = ax2.hist(np.random.randn(1000), 50)
ax2.set_xlabel('time (s)')

plt.show()
```



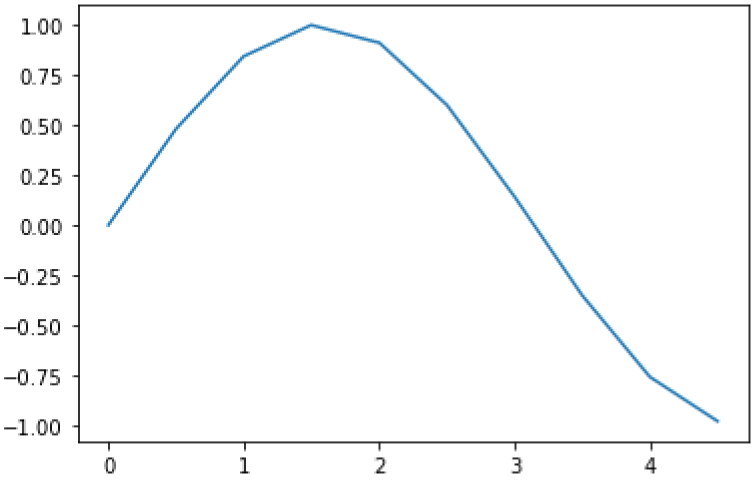


In [93]:

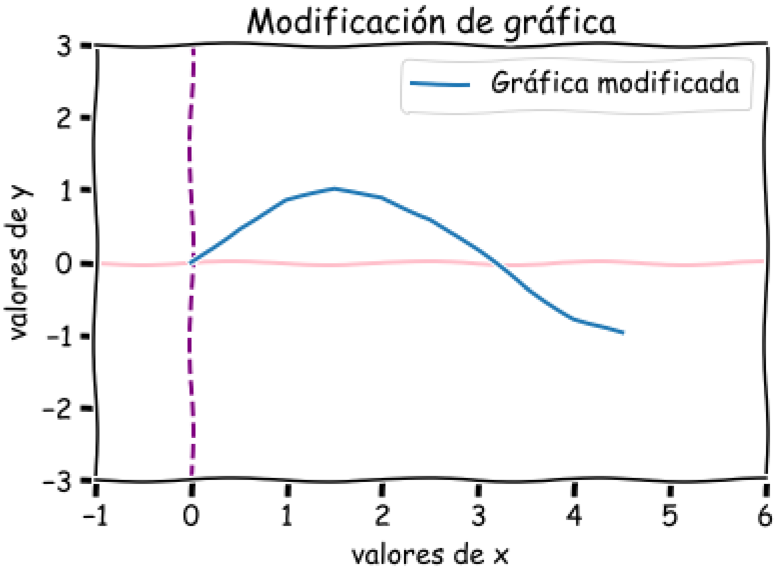
```
import math
import numpy as np
from matplotlib import pyplot as plt

x = np.array(range(10))*0.5
y = np.zeros (len(x))
for i in range (len(x)):
    y[i] = math.sin(x[i])


plt.ion()
plt.plot(x,y)
```

Out[93]:

```
[<matplotlib.lines.Line2D at 0x1eae94c7850>]
```

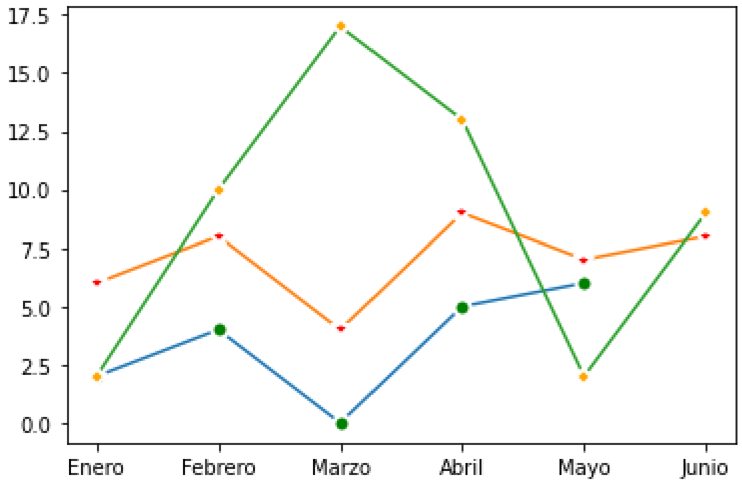


```
In [103]: with plt.xkcd():
              plt.title("Modificación de gráfica")
              plt.xlabel("valores de x")
              plt.ylabel("valores de y")
              plt.axhline(0, color="pink", linestyle="-")
              plt.axvline(0, color="purple", linestyle="--")
              plt.xlim(-1, 6)
              plt.ylim(-3, 3)
              plt.plot(x,y, label = "Gráfica modificada")
              plt.legend();
```



In [111]:

```
meses = ['Enero', 'Febrero', 'Marzo', 'Abril', 'Mayo', 'Junio']
mapeado = range(len(meses))

plt.plot(np.random.randint(9, size=[5]),
         marker="o", markersize="8", markeredgewidth="2",
         markerfacecolor="green", markeredgecolor="white")
plt.plot(np.random.randint(12, size=[6]),
         marker="*", markersize="10", markeredgewidth="2",
         markerfacecolor="red", markeredgecolor="white")
plt.plot(np.random.randint(18, size=[6]),
         marker="D", markersize="5", markeredgewidth="2",
         markerfacecolor="orange", markeredgecolor="white")
plt.xticks(mapeado, meses)
plt.show()
```

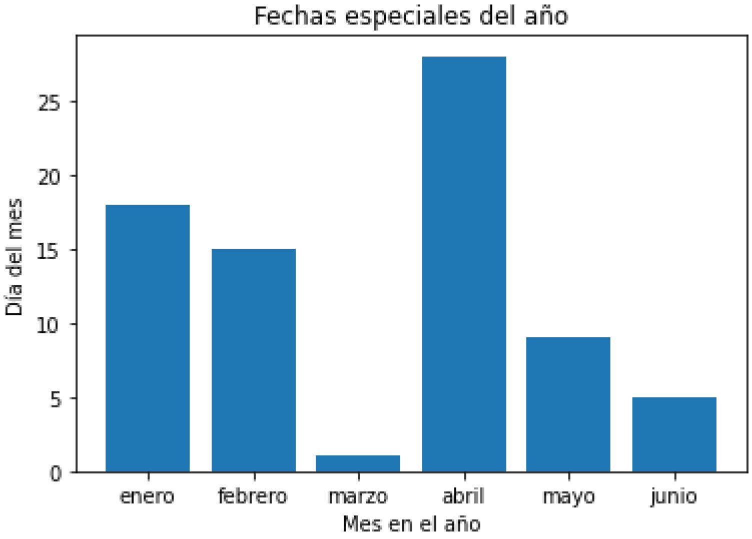


7.4 Exercises

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

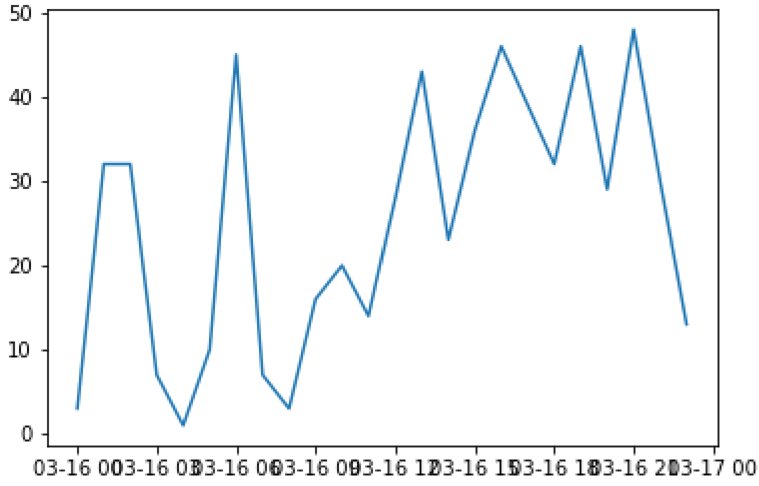
In [120]:

```
## modo 1
labels = ["enero", "febrero", "marzo", "abril", "mayo", "junio"]
values = [18, 15, 1, 28, 9, 5]

plt.title("Fechas especiales del año")
plt.ylabel("Día del mes")
plt.xlabel("Mes en el año")
xs = np.arange(len(labels))

xtick_locs = xs
plt.xticks(xtick_locs, labels)

plt.bar(xs, values);
```



In [127]:

```
## modo 2
import matplotlib.pyplot as plt
import datetime
import numpy as np

x = np.array([datetime.datetime(2014, 3, 16, i, 0) for i in range(24)])
y = np.random.randint(50, size=x.shape)

plt.plot(x,y)
plt.show()
```



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

In [ ]:

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

In [ ]: