# Matplotlib Workbook

May 10, 2022

## 1 Matplotlib Workbook

Welcome to the Matplotlib Workbook. Let's start with importing the NumPy and Matplotlib libraries.

[1]:

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

We need the following line of code to make interactive plots when using Jupyter Notebook.

[9]:

```
%matplotlib inline
```

### 1.0.1 Line plots

[3]:

```
x = np.linspace(0, 2, 100)
```

[7]:

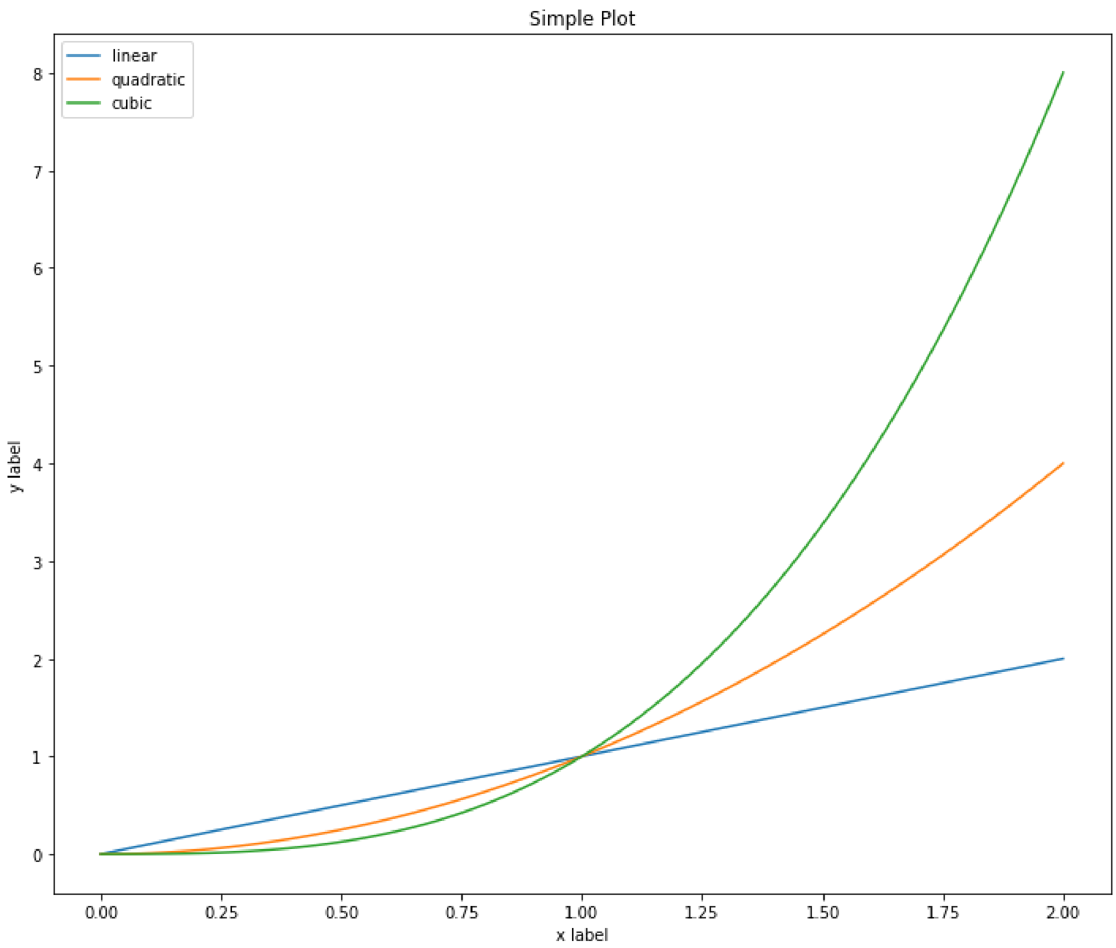
```
plt.figure?
```

[10]:

```
# Note that even in the OO-style, we use `.pyplot.figure` to create the figure.
fig = plt.figure(figsize=(12,10))
ax = plt.subplot(111)  # Create a figure and an axes.
ax.plot(x, x, label='linear')  # Plot some data on the axes.
ax.plot(x, x**2, label='quadratic')  # Plot more data on the axes...
ax.plot(x, x**3, label='cubic')  # ... and some more.
ax.set_xlabel('x label')  # Add an x-label to the axes.
ax.set_ylabel('y label')  # Add a y-label to the axes.
ax.set_title("Simple Plot")  # Add a title to the axes.
ax.legend()  # Add a legend.
```

[10]:

```
<matplotlib.legend.Legend at 0x127bff4f0>
```



[11]:

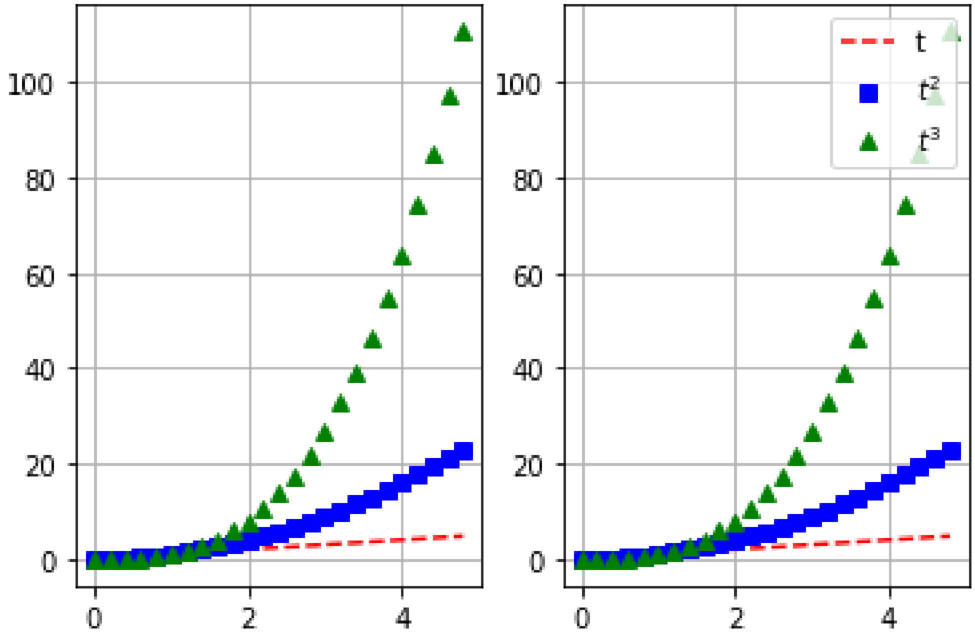
```
# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

fig = figsize=(10,8)
# red dashes, blue squares and green triangles

ax = plt.subplot(121)
ax.plot(t, t, 'r--', t, t**2, 'bs', t, t**3, 'g^')
ax.grid()

ay = plt.subplot(122)
ay.plot(t, t, 'r--', label='t')
ay.plot(t, t**2, 'bs', label='$t^{2}$')
ay.plot(t, t**3, 'g^', label='$t^{3}$')
ay.grid()
ay.legend(loc=1)
```

```
[11]: <matplotlib.legend.Legend at 0x127c98c10>
```



### 1.0.2 Scatter Plots

```
[12]: np.random.randint?
```

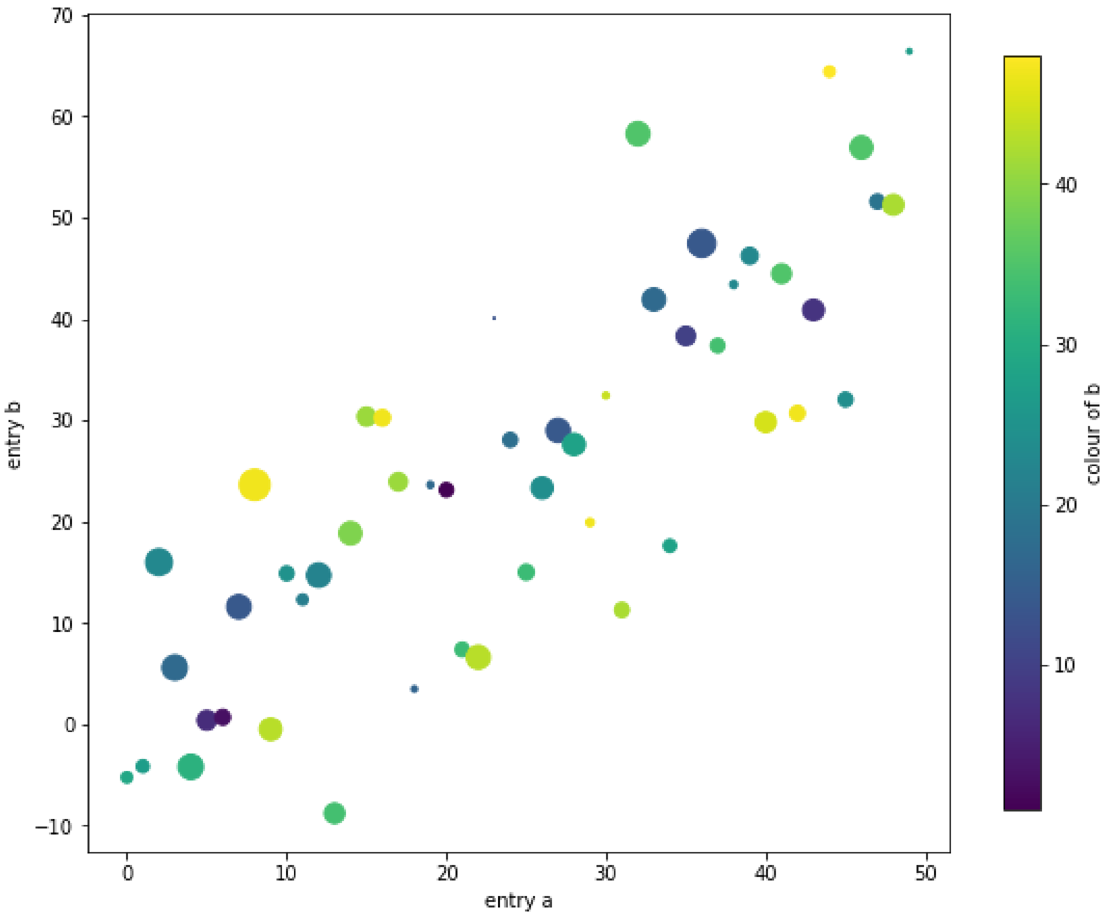
```
[13]: data = {'a': np.arange(50),
              'c': np.random.randint(0, 50, 50),
              'd': np.random.randn(50)}
      data['b'] = data['a'] + 10 * np.random.randn(50)
      data['d'] = np.abs(data['d']) * 100
```

```
[14]: data
```

```
[14]: {'a': array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
              17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
              34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]),
       'c': array([29, 27, 23, 17, 31,  7,  3, 14, 47, 43, 25, 21, 22, 34, 39, 41, 47,
              41, 16, 17,  1, 33, 43, 13, 18, 33, 24, 14, 28, 47, 44, 42, 35, 17,
              28, 10, 14, 34, 23, 23, 45, 35, 47,  8, 48, 24, 35, 19, 42, 27]),
       'd': array([ 33.13847645,  43.90203165, 189.41883073, 168.83947593,
              168.42302308, 100.63537241,  63.98568884, 157.0240793 ,
              256.21347667, 134.83073148,  56.72739845,  33.00473045,
              156.93332248, 108.66417225, 140.1825163 ,  94.03613738,
               71.14806202,  87.60817155,  10.35818864,  12.34320653,
               53.7150167 ,  55.02165224, 153.19112992,   1.05295764,
               57.86409711,  66.11869438, 129.52077621, 154.08874659,
              131.53019956,  18.90485044,  11.34840737,  60.23888727,
              151.76043822, 141.79040114,  45.91025509,  99.10729028,
              206.99692318,  55.0857902 ,  16.16690134,  74.80109463,
              115.3296595 , 104.33731958,  62.67967573, 121.39445992,
               34.12502145,  57.15570798, 140.65169944,  58.35663001,
              115.96065487,   7.89538821]),
       'b': array([-5.23556273, -4.15628858, 15.9914454 ,  5.58183079, -4.20139509,
               0.39487062,  0.68355592, 11.59807904, 23.62873203, -0.49006754,
              14.89220799, 12.30616941, 14.7076258 , -8.78493072, 18.85260537,
              30.37128989, 30.25588553, 23.93645818,  3.48754746, 23.62332997,
              23.12345246,  7.37018704,  6.6053359 , 40.0961254 , 28.06814543,
              15.00607547, 23.32707286, 28.99259581, 27.61152352, 19.92444832,
              32.4485113 , 11.27552871, 58.28147304, 41.92133866, 17.61510948,
              38.32072108, 47.44976713, 37.3690511 , 43.40170491, 46.24162922,
              29.82385341, 44.46911167, 30.70186955, 40.88618862, 64.41945724,
              32.04765839, 56.92737117, 51.5931456 , 51.26543428, 66.42533935])}
```

[16]:
```python
fig = plt.figure(figsize=(10,8))
ax = plt.subplot(111)
cplot = ax.scatter('a', 'b', c='c', s='d', data=data)
cbar = plt.colorbar(cplot, ax=ax, shrink=0.9)
cbar.ax.set_ylabel('colour of b')
plt.xlabel('entry a')
plt.ylabel('entry b')

plt.show()
```



### 1.0.3 Contour Plots

[17]:

```
x = np.linspace(-10,10,21)
y = np.linspace(0,20,21)

X, Y = np.meshgrid(x,y)

Z = np.random.rand(21,21)*100.
```

[19]:

```
print(np.shape(X))
```

```
(21, 21)
```

[21]:

```
c_levels = np.linspace(Z.min(), Z.max(), 10)
print(c_levels)
```

```
[2.96755651e-02 1.11359065e+01 2.22421375e+01 3.33483684e+01
 4.44545993e+01 5.55608303e+01 6.66670612e+01 7.77732922e+01
 8.88795231e+01 9.99857541e+01]
```

[22]:

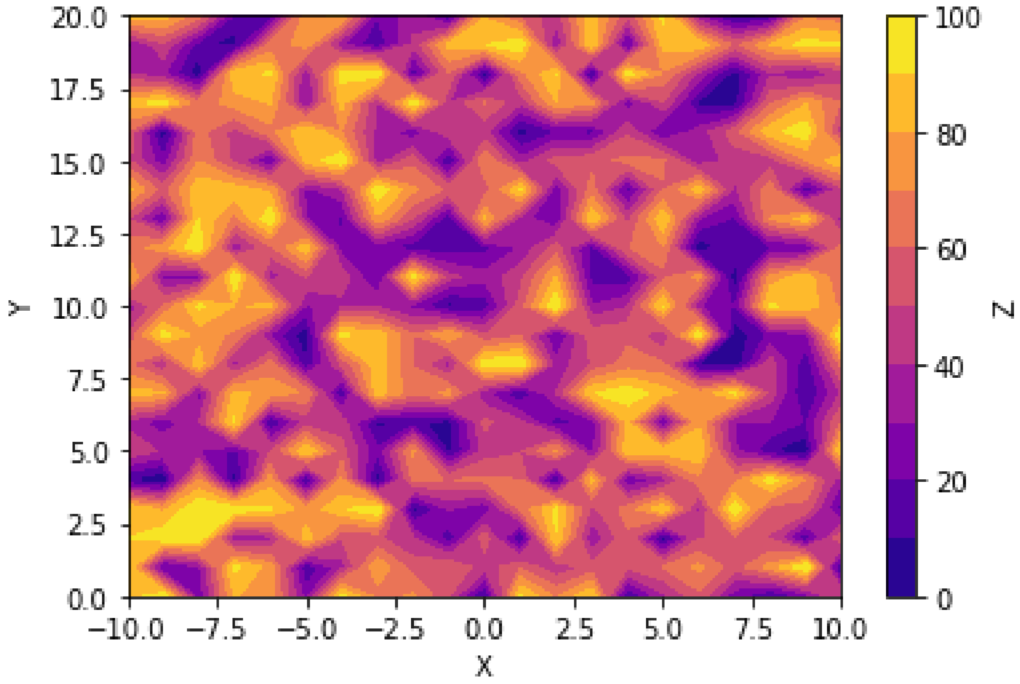
```
c_levels = np.linspace(0, 100, 11)
print(c_levels)
```

```
[  0.  10.  20.  30.  40.  50.  60.  70.  80.  90. 100.]
```

[25]:

```
fig = figsize=(10,8)
ax = plt.subplot(111)
cplot = ax.contourf(X, Y, Z, levels=c_levels, cmap='plasma')
cbar = plt.colorbar(cplot, ax=ax)
cbar.ax.set_ylabel('Z')
plt.xlabel('X')
plt.ylabel('Y')

plt.show()
```



### 1.0.4 Polar Plots

[27]:

```
rad = np.linspace(0,10,11)
theta = np.linspace(0, 2*np.pi, 91)

R, T = np.meshgrid(rad,theta)

Z = np.zeros(np.shape(R))
```

[29]:

```
Z
```

[29]:

```
array([[0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       …,
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.]])
```

[30]:

```
Z[R<2] = 20.
Z[R>6] = 60.
```

[31]:

```
Z
```

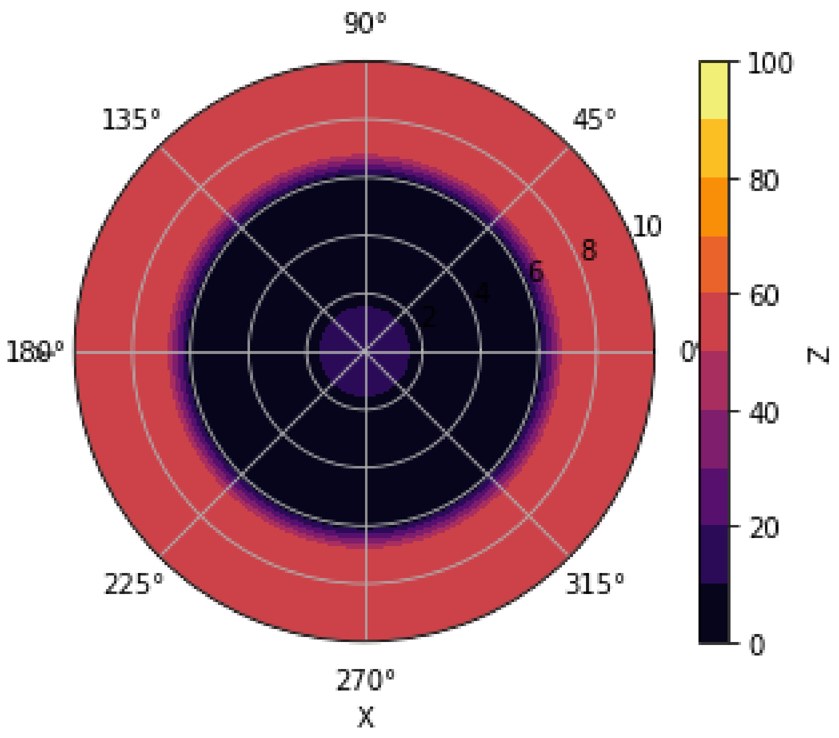
[31]:

```
array([[20., 20.,  0., …, 60., 60., 60.],
       [20., 20.,  0., …, 60., 60., 60.],
       [20., 20.,  0., …, 60., 60., 60.],
       …,
       [20., 20.,  0., …, 60., 60., 60.],
       [20., 20.,  0., …, 60., 60., 60.],
       [20., 20.,  0., …, 60., 60., 60.]])
```

[32]:

```
fig = figsize=(10,8)
ax = plt.subplot(111, polar=True)
cplot = ax.contourf(T, R, Z, levels=c_levels, cmap='inferno')
cbar = plt.colorbar(cplot, ax=ax)
cbar.ax.set_ylabel('Z')
plt.xlabel('X')
plt.ylabel('Y')

plt.show()
```

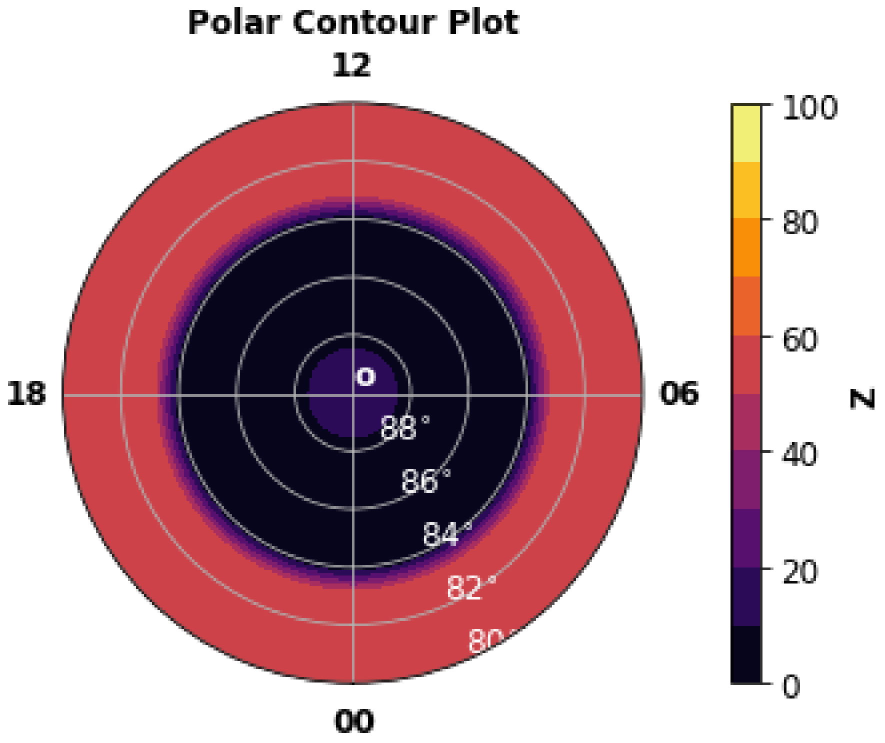


[33]:

```
fig = figsize=(10,8)
ax = plt.subplot(111, polar=True)
cplot = ax.contourf(T, R, Z, levels=c_levels, cmap='inferno')
cbar = plt.colorbar(cplot, ax=ax, pad=0.1)
cbar.ax.set_ylabel('Z', fontsize=12, weight='bold')
cbar.ax.tick_params(labelsize=12)
ax.set_theta_zero_location('S')
ax.set_xticks(np.deg2rad([0,90,180,270]))
ax.set_xticklabels(['00','06','12','18'], fontsize=12, weight='bold')
ax.set_yticks([0,2,4,6,8,10])
ax.set_yticklabels(['o', '$88^{\circ}$', '$86^{\circ}$', '$84^{\circ}$',␣
 ↪'$82^{\circ}$', '$80^{\circ}$'], fontsize=12, weight='bold',color='white')
ax.set_title('Polar Contour Plot', fontsize=12, weight='bold')
```

[33]:

```
Text(0.5, 1.0, 'Polar Contour Plot')
```



```
[34]: ax.set_rlabel_position?
```

Congratulations! You have completed the Matplotlib Workbook. Copy this workbook with a new name and try changing the exercises to explore the functions further.