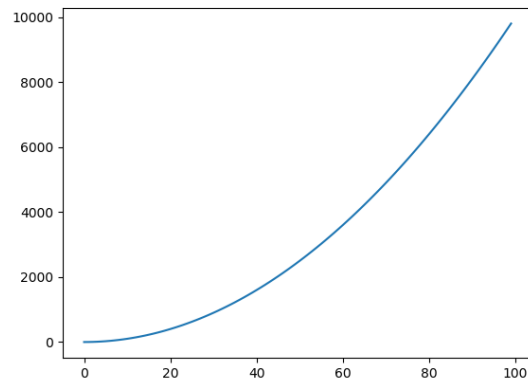


Figure 1: A simple xy line chart



1 Matplotlib: simple xy line charts, and simple save to file

Matplotlib is a powerful library for plotting data. Data may be in the form of lists, or data from NumPy and Pandas may be used directly. Charts are very highly configurable, but here we'll focus on the key basics first.

A useful resource for matplotlib is a gallery of charts with associated code: <https://matplotlib.org/gallery.html>

1.1 Plotting a single line

The following code plots a simple line plot and saves the file in png format. Other formats may be used (e.g. jpg), but png offers an open source high quality format which is ideal for charts.

```
import matplotlib.pyplot as plt

# The following line is only needed to display chart in Jupyter notebooks
%matplotlib inline

X=range(100)
Y=[value ** 2 for value in X] # (A list comprehension)

plt.plot(X,Y)
plt.show()
plt.savefig('plot_01.png') # optional save as png file
```

1.2 Plotting two lines

To plot two lines we simply add another plot before generating the chart with `plt.show()`

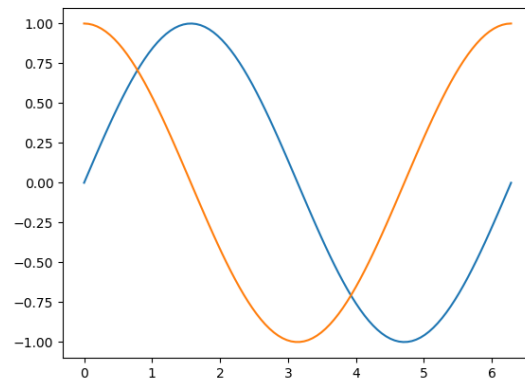
```
import numpy as np

import matplotlib.pyplot as plt

# The following line is only needed to display chart in Jupyter notebooks
%matplotlib inline

# np.linspace creates an array of equally spaced numbers
X=np.linspace(0,2*np.pi,100) # 100 points between 0 and 2*pi
```

Figure 2: A simple xy line chart with two lines



```
Ya=np.sin(X)
Yb=np.cos(X)

plt.plot(X,Ya)
plt.plot(X,Yb)
plt.show()
```

1.3 Saving figures

To save a figure use `plt.savefig('my_filename.png')` before `plt.show()` to save as png format (best for figures, but you can also use jpg or tif).