Python102

Python for Data Science Bootcamp

(5.1) Python for Data Visualization

Matplotlib

AIAT Academy

Python for Data Visualization Outline

- **Matplotlib**
- Seaborn

Seaborn

matplotlib

AiAT ACADEMY

Matplotlib

Introduction to Matplotlib

- The most popular plotting library for Python
- Control over every aspect of a figure
- Designed to have similar feel to MATLAB's graphical plotting
- https://matplotlib.org

Matplotlib Installation

- To install Matplotlib, just going to your terminal or command prompt and typing

```
conda install matplotlib
```

or

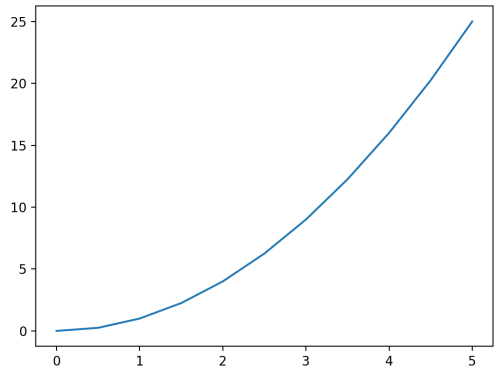
```
pip install matplotlib
```

Matplotlib Usages (Simple Plotting)

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

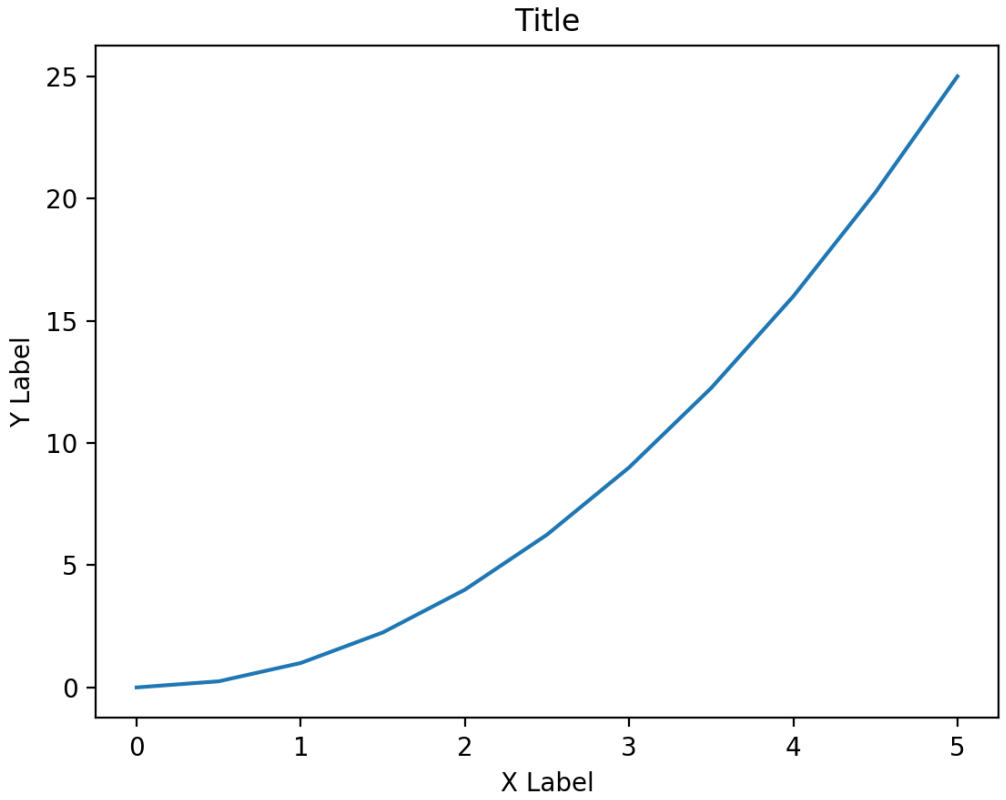
Allow a graph plotted by matplotlib to show in colab/jupyther

```
x = np.linspace(0,5,11)  # generate linear space 0 to 5
# array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
y = x ** 2
# array([ 0. ,  0.25,  1. ,  2.25,  4. ,  6.25,  9. , 12.25, 16. , 20.25, 25. ])

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



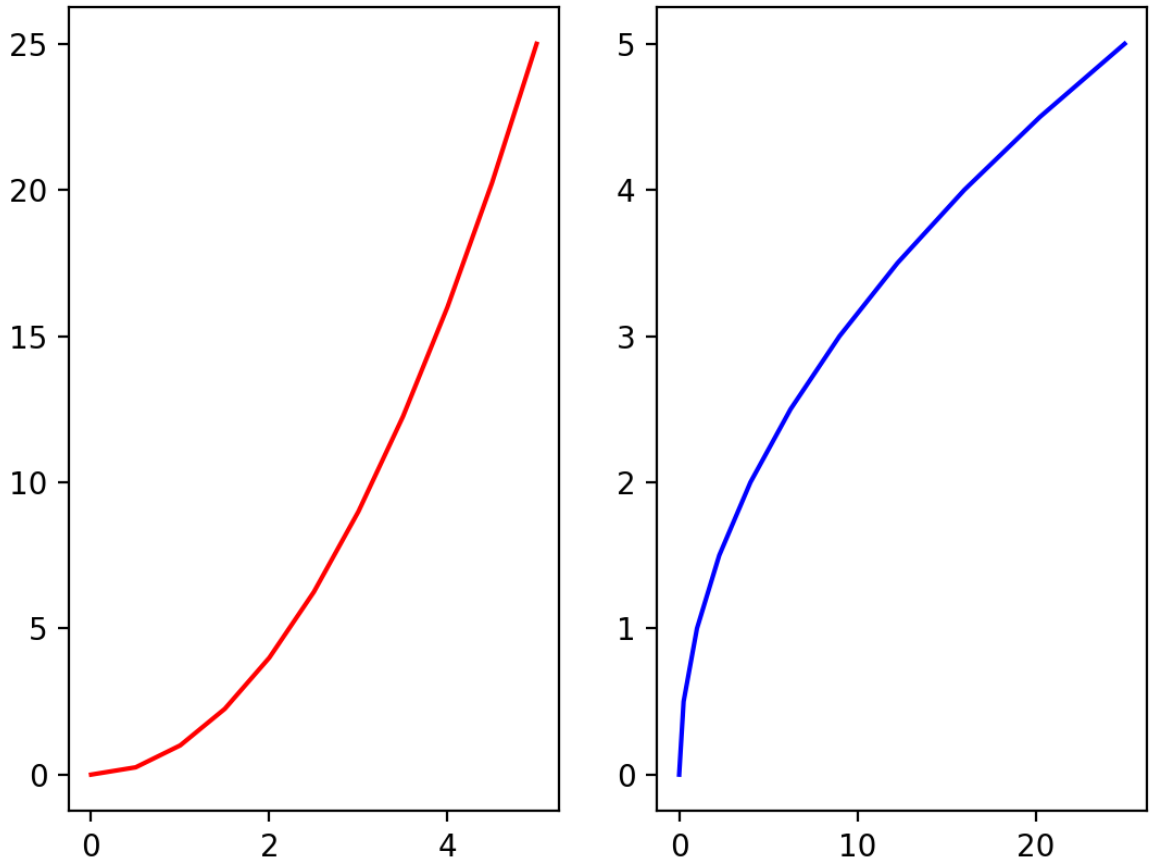
Matplotlib Usages (Simple Plotting)

```
x = np.linspace(0,5,11)  # generate linear space 0 to 5
# array([0. , 0.5, 1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. ])
y = x ** 2
# array([ 0. ,  0.25,  1. ,  2.25,  4. ,  6.25,  9. , 12.25, 16. , 20.25, 25. ])
plt.plot(x, y)
plt.xlabel('X Label')
plt.ylabel('Y Label')
plt.title('Title')
plt.show()
```



Matplotlib Usages (Multiple Plotting)

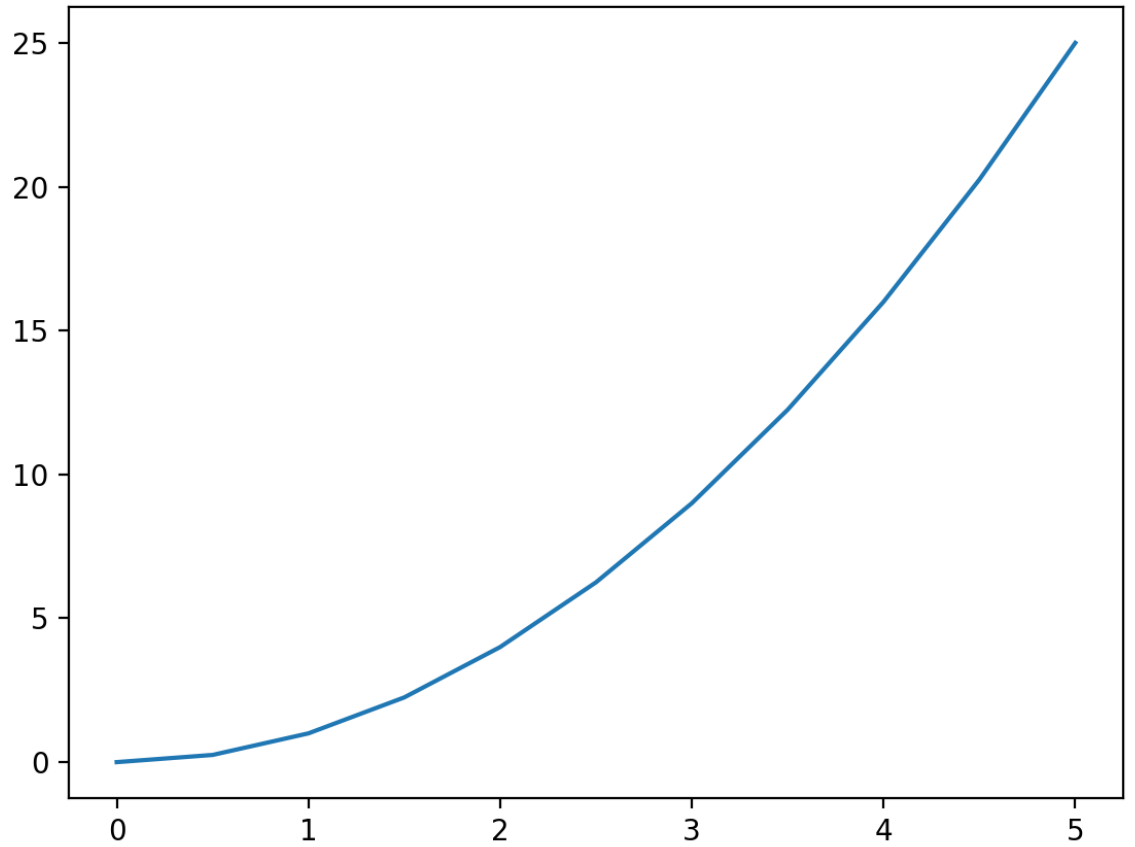
```
x = np.linspace(0,5,11)  # generate linear space 0 to 5
y = x ** 2
plt.subplot(1,2,1)
plt.plot(x, y, 'r')

plt.subplot(1,2,2)
plt.plot(y, x, 'b')
plt.show()
```




```
# plt.subplot(nrow, ncolumn, plotnumber)
```

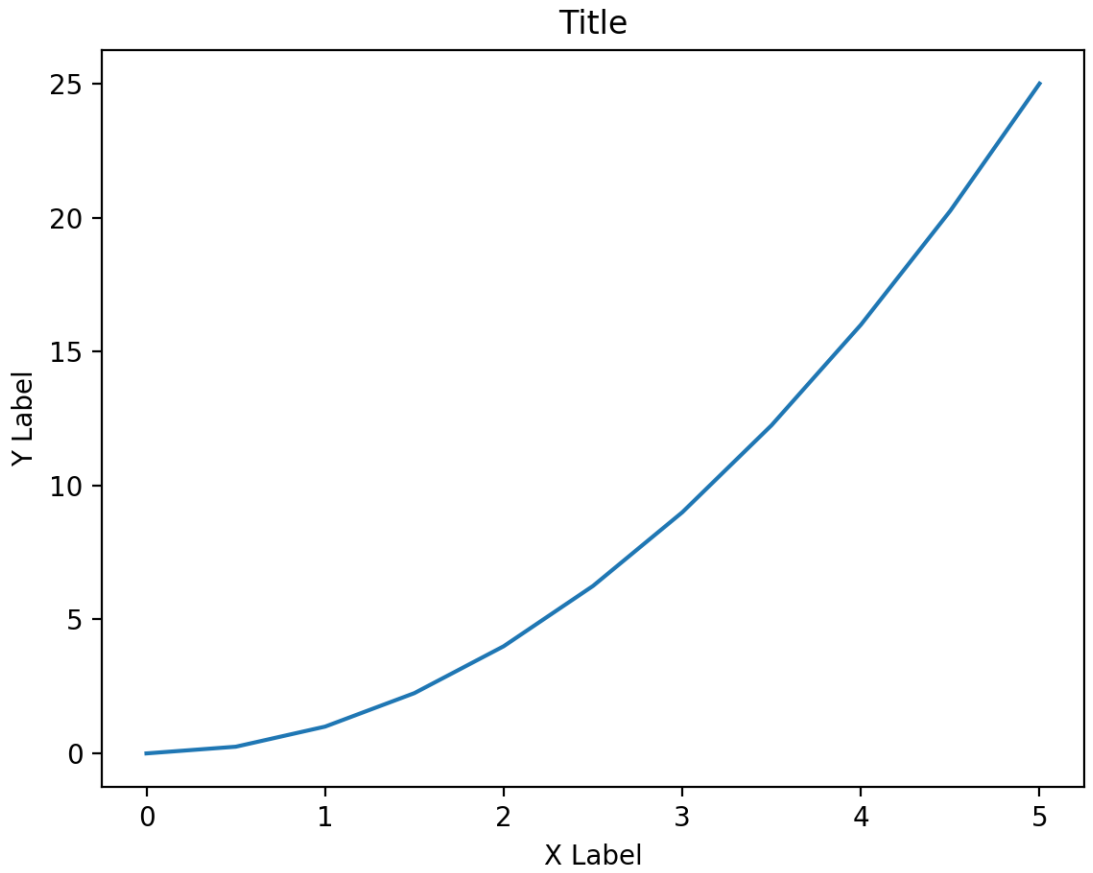
Matplotlib Usages (Plotting Object)

```
fig = plt.figure() # generate a figure object (variable)
axes = fig.add_axes([0.1,0.1,0.8,0.8])
# fig.axes(left_canvas,bottom_canvas,width_figure,height_figure)
axes.plot(x,y)
plt.show()
```



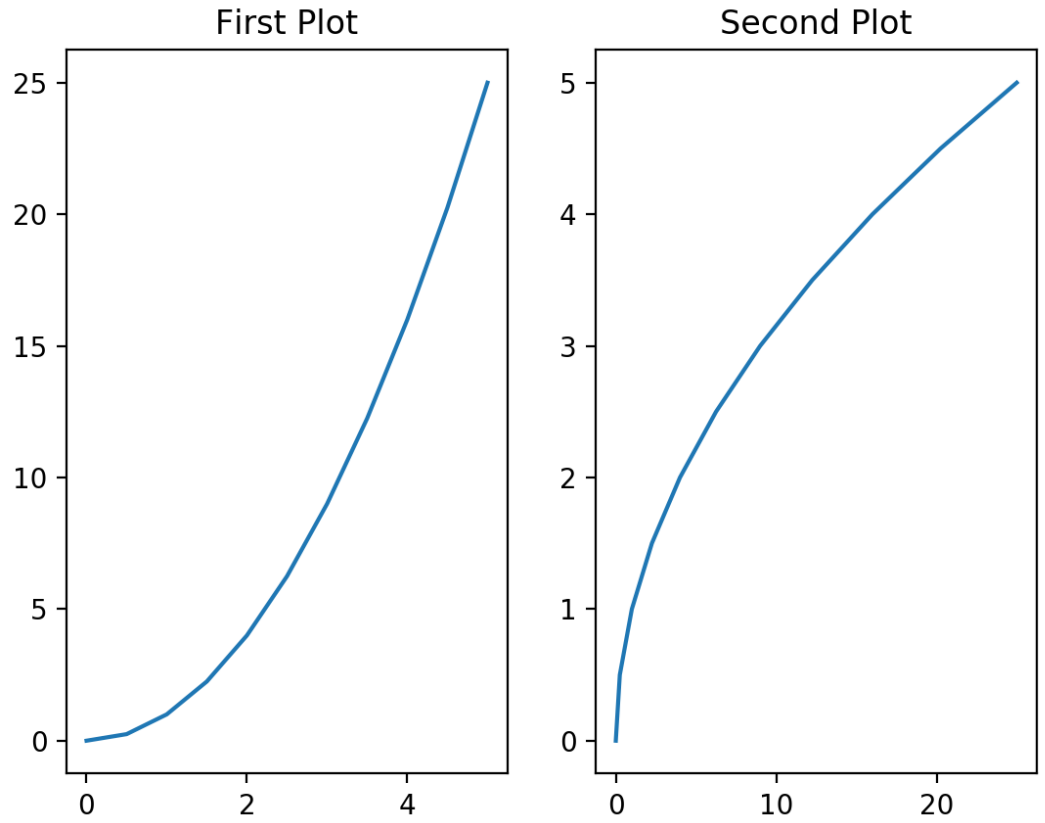
Matplotlib Usages (Plotting Object)

```
fig = plt.figure() # generate a figure object (variable)
axes = fig.add_axes([0.1,0.1,0.8,0.8])
# fig.axes(left_canvas,bottom_canvas,width_figure,height_figure)
axes.plot(x,y)
axes.set_xlabel('X Label')
axes.set_ylabel('Y Label')
axes.set_title('Title')
plt.show()
```



Matplotlib Usages (Plotting Multiple Object)

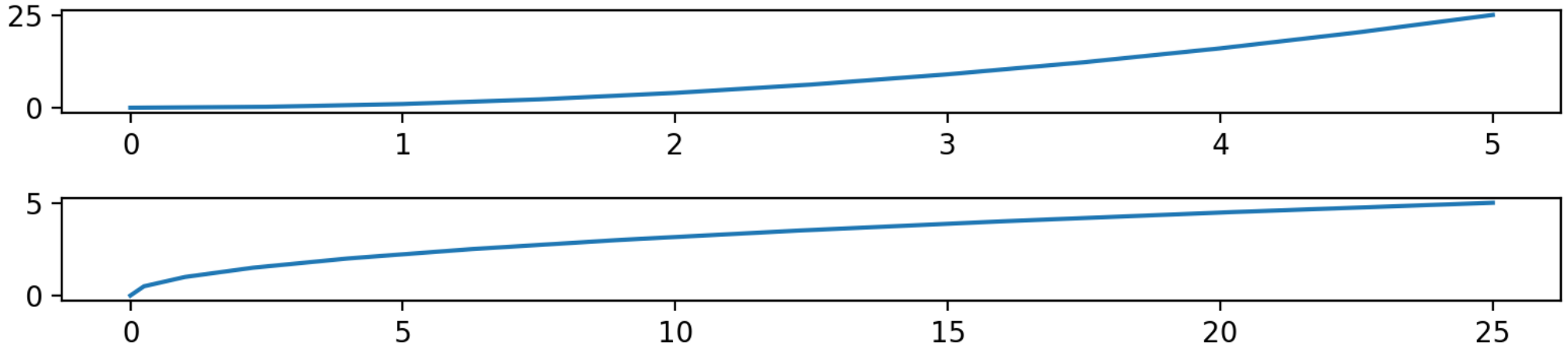
```
fig, axes = plt.subplots(nrows=1, ncols=2)
axes[0].plot(x,y)
axes[0].set_title('First Plot')
axes[1].plot(y,x)
axes[1].set_title('Second Plot')
plt.show()
```



Matplotlib Usages (Plotting Multiple Object)

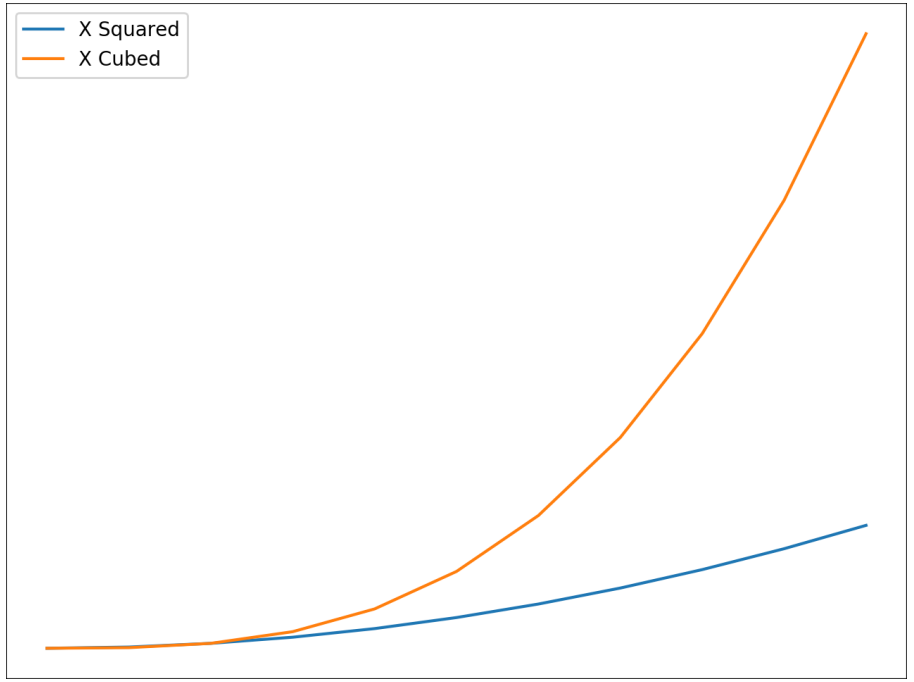
AiAT ACADEMY

```
fig,axes = plt.subplots(nrows=2,ncols=1,figsize=(8,2))
axes[0].plot(x,y)
axes[1].plot(y,x)
plt.tight_layout()
# tighten plotting
plt.show()
fig.savefig('my_picture.png')
```



Matplotlib Usages (Plotting with Legend)

```
fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
ax.plot(x,x**2, label='X Squared')
ax.plot(x,x**3, label='X Cubed')
ax.legend(loc='best')   # legend location see more in documentary
plt.show()
```



Matplotlib Usages (More Details)

```
fig = plt.figure()
ax = fig.add_axes([0.1,0.1,0.8,0.8])
ax.plot(x,y,linewidth=3,alpha=0.5,linestyle='--',marker='o's)
# alpha for line transparent
plt.show()
```