Part 1: Using 'pyplot.plot'

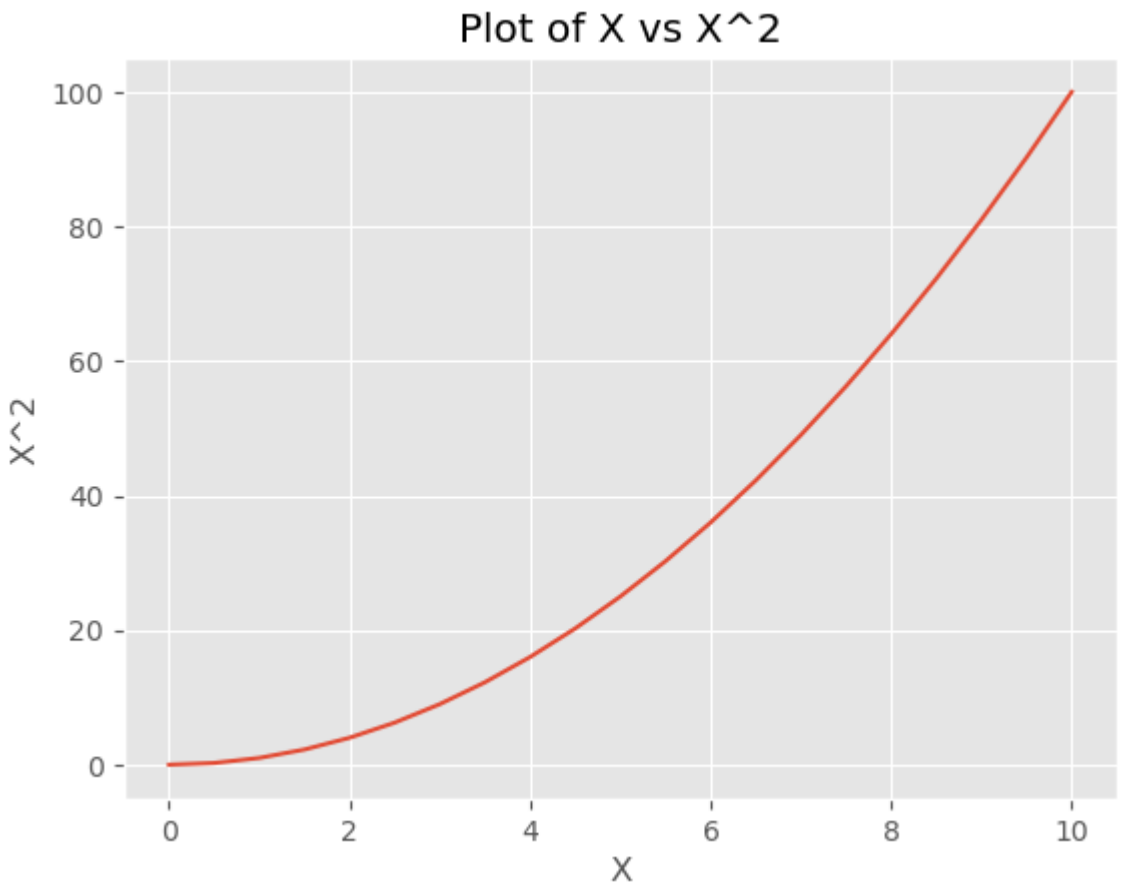
1. Basic Line Plot:

In [191…

```
#a. Generate a list of numbers from 0 to 10, incremented by 0.5.
import numpy as np
x=np.arange(0,10.5,.5)
x

#b. Plot the numbers against their square values using an available style of y
import matplotlib as mpl
import matplotlib.pyplot as plt

y1=x**2 #create y-values
plt.style.use('ggplot')
plt.plot(x,y1) #plt.plot creates a basic axes


#c. Label the x-axis as 'X', the y-axis as 'X^2', and give the plot a title "P
plt.xlabel('X')
plt.ylabel('X^2')
plt.title('Plot of X vs X^2')
```

Out[191]:

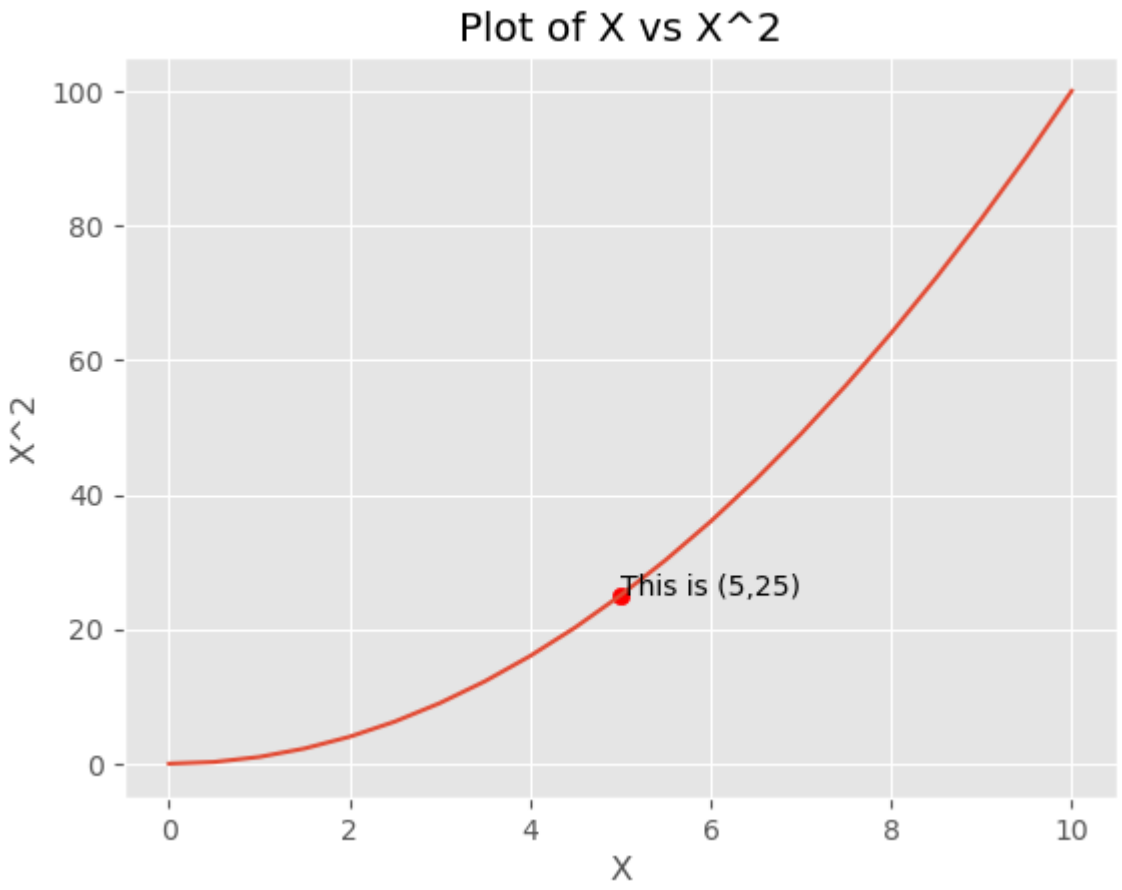
```
Text(0.5, 1.0, 'Plot of X vs X^2')
```



2. Plot Annotations:

In [136…
```
#b. Annotate the point (5, 25) with the text "This is (5,25)".

plt.style.use('ggplot')
plot=plt.plot(x,y1) #plt.plot creates a basic axes
plt.xlabel('X')
plt.ylabel('X^2')
plt.title('Plot of X vs X^2')
plt.scatter(5,25, c='r')
plt.annotate('This is (5,25)', (5,25))
```

Out[136]:
```
Text(5, 25, 'This is (5,25)')
```



Part 2: Using Object-Oriented 'ax' methods

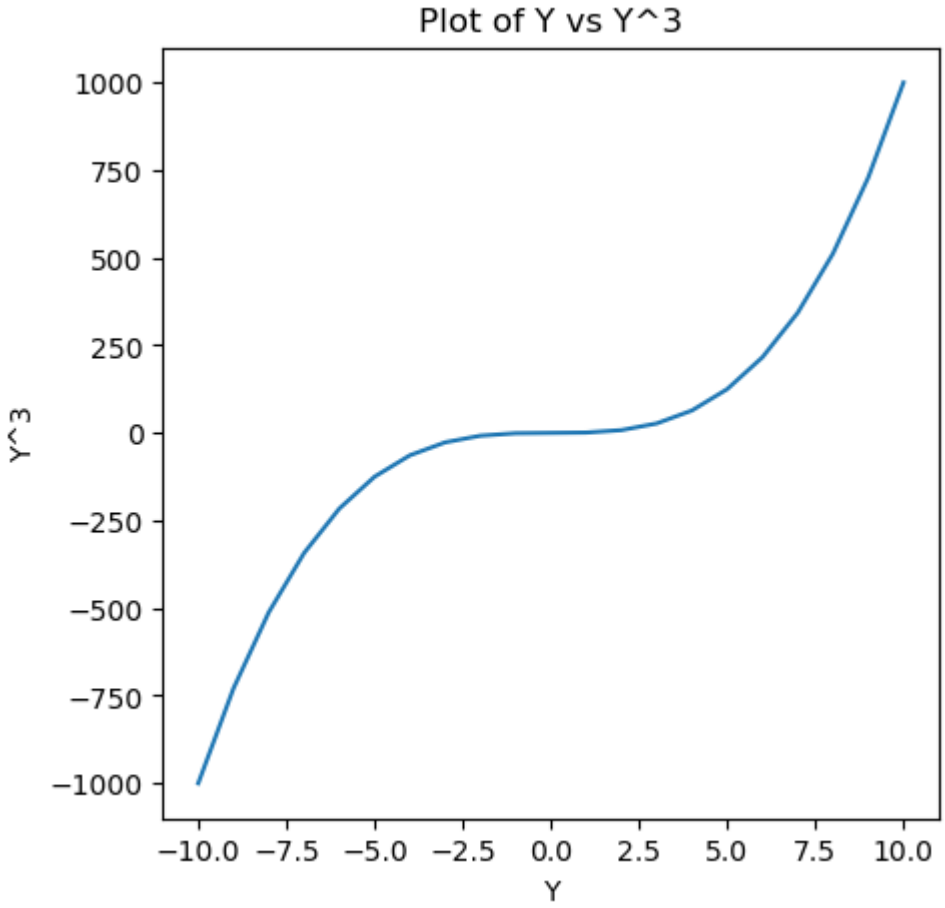
3. Basic Line Plot:

In [192…
```
#a. Generate a list of numbers from -10 to 10, incremented by 1.
y= np.arange(-10,11,1)
```

In [193…
```
#b. Create a subplot object and plot the numbers against their cube values.
ytothe3rd=y**3 #create y values

plt.style.use('default') #set style
fig,axs= plt.subplots(ncols=1,nrows=1,figsize=(5,5)) #OO method of plotting
axs.plot(y,ytothe3rd)

#c. Label the x-axis as 'Y', the y-axis as 'Y^3', and give the plot a title "P
axs.set_xlabel("Y")
```

```
axs.set_title("Plot of Y vs Y^3", loc='center')
axs.set_ylabel("Y^3")
```

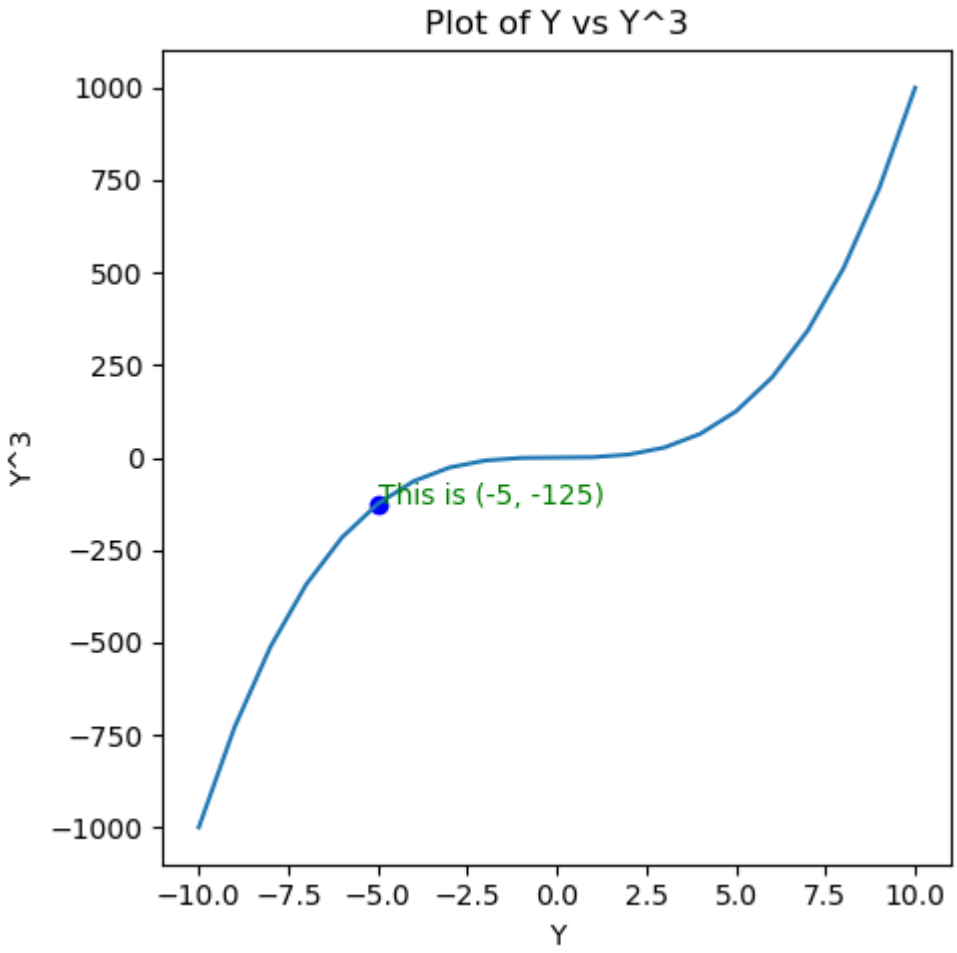
Out[193]: Text(0, 0.5, 'Y^3')



In [138…

```
fig,axs= plt.subplots(ncols=1,nrows=1,figsize=(5,5)) #OO method of plotting
axs.plot(y,ytothe3rd)

#c. Label the x-axis as 'Y', the y-axis as 'Y^3', and give the plot a title "P
plt.style.use('default')
axs.set_xlabel("Y")
axs.set_title("Plot of Y vs Y^3", loc='center')
axs.set_ylabel("Y^3")

plt.tight_layout()

axs.scatter(-5,-125, c='b')
axs.annotate('This is (-5, -125)', (-5,-125), color='g')
```

Out[138]: Text(-5, -125, 'This is (-5, -125)')

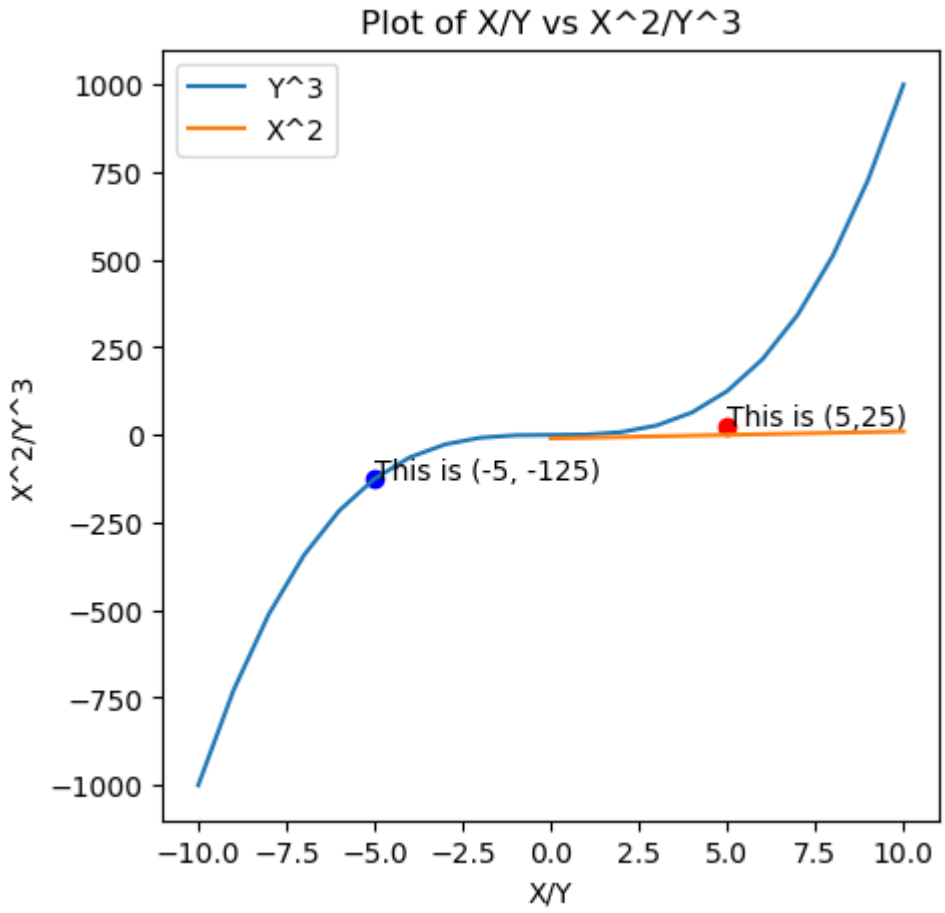


5. Combining Plots:

In [134…

```
#a. Plot both functions (X^2 and Y^3) on the same figure, using different colo
plt.style.use('default') #set style
fig,axs= plt.subplots(ncols=1,nrows=1,figsize=(5,5)) #OO method of plotting
axs.plot(y,ytothe3rd, label='Y^3')
axs.plot(x,y,label='X^2')

#c. Label the x-axis as 'Y', the y-axis as 'Y^3', and give the plot a title "P
axs.set_xlabel("X/Y")
axs.set_title("Plot of X/Y vs X^2/Y^3", loc='center')
axs.set_ylabel("X^2/Y^3")


axs.scatter(-5,-125, c='b')
axs.annotate('This is (-5, -125)', (-5,-125))

plt.scatter(5,25, c='r')
plt.annotate('This is (5,25)', (5,25))

plt.legend(loc ='upper left')
```

Out[134]: <matplotlib.legend.Legend at 0x15ff33510>



6. Functional Plotting:

In [183…

```
#a. Write a function to plot its two input arguments, y and x, against one ano
#It should label the y-axis "activity" and the x-axis "time", and have a good

def pltplotting(x,y,z = '-'):
    '''
    For solid line, enter '-' into the last argument.
    For circle markers, enter 'o'.
    For both, enter both.
    The default is a solid line.
    '''

    #create the graph
    plt.style.use('default')
    plt.plot(x,y,z)
    plt.xlabel('time')
    plt.ylabel('activity')
```

In [184…

```
help(pltplotting)
```

```
Help on function pltplotting in module __main__:

pltplotting(x, y, z='-')
    For solid line, enter '-' into the last argument.
    For circle markers, enter 'o'.
    For both, enter both.
    The default is a solid line.
```

In [196…

```
pltplotting([1,2,3,6,45],[4,56,87,31,32],'-o')
```