Plots - V

axes() function

In this notebook, we discuss additional customization of graphs. Sometimes it is desirable to compare two plots side by side rather in one single graph. This is particularly true when the scales of the two plots are very different. We use the `axes()` function to do this.

Follow the same process we have used so far to generate the plots. The only difference is as follows:

After defining the data for the two axes, call the axes() function by providing the following values:

1. xlo = the x coordinate of the lower bottom point where the graph begins
2. ylo = the y coordinate of the lower bottom point where the graph begins
3. width = the width of the grap h area
4. height = the height of the graph area Note that all of the above values must be in figure units (numbers between 0 and 1 scaled to the dimensions of the figure) After the above step, set the label and title and call the plot function for the first plot.

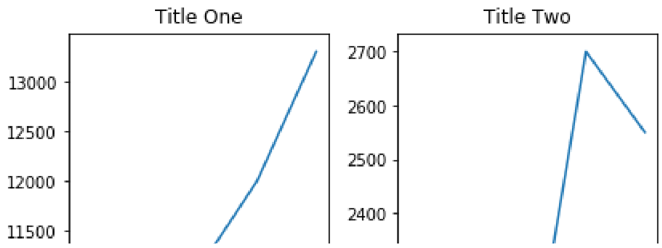
Call the `axes()` function again for the second plot changing the arguments as appropriate. If the plots are to display side by side, the only difference is likely to be in the value of xlo. The ylo value is the only likely change if the plots are displayed one on top of the other

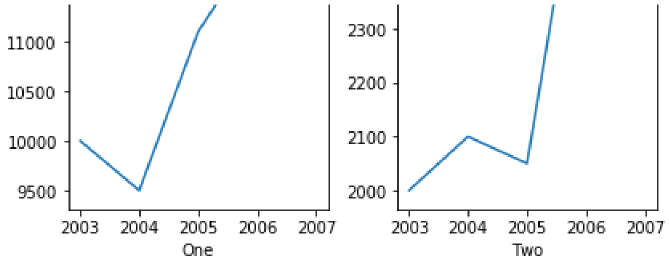
After calling `axes()` the second time, set the label and title for the second plot and call the plot function. Finally show the plot.

In this example, the graphs are plotted side by side.

In [1]:

```
'''
In this example, the graphs are plotted side by side.
'''
import matplotlib.pyplot as plt

# x-axis data
years = ['2003', '2004', '2005', '2006', '2007']

# line 1 data
full_time_students = [10000, 9500, 11100, 12000, 13300]

# line 2 data
part_time_students = [2000, 2100, 2050, 2700, 2550]

#Call axes() for the first plot
plt.axes([0.05, 0.05, 0.375, 0.9])

# Set the axes, title and label for the first plot and call the plot function for the first time
plt.xlabel('One')
plt.title('Title One')
plt.plot(years, full_time_students)

# Set the axes, title and label for the second plot and call the plot function for the second time
plt.axes([0.525, 0.05, 0.375, 0.9])
plt.xlabel('Two')
plt.title('Title Two')
plt.plot(years, part_time_students)

#Last step:  show the plot
plt.show()
```

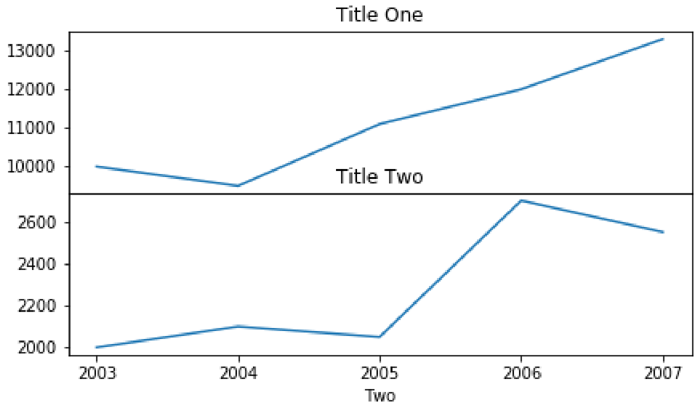




In this example, the graphs are plotted one on top of the other.

In [2]:

```
'''
In this example, the graphs are plotted one on top one another.
'''
import matplotlib.pyplot as plt

# x-axis data
years = ['2003', '2004', '2005', '2006', '2007']

# line 1 data
full_time_students = [10000, 9500, 11100, 12000, 13300]

# line 2 data
part_time_students = [2000, 2100, 2050, 2700, 2550]


#Call axes() for the first plot
plt.axes([0.05, 0.4, 0.9, 0.35])

#Set the label and title for the first plot
plt.xlabel('One')
plt.title('Title One')

# Call the plot() function for the first plot
plt.plot(years, full_time_students)

# Set the axes, title and label for the second plot and call the plot function for the second time
plt.axes([0.05, 0.05, 0.9, 0.35])
plt.xlabel('Two')
plt.title('Title Two')
plt.plot(years, part_time_students)

#Last step:  show the plot
plt.show()
```


Title One
Title Two
Two