

Plots - VI

subplot()

This notebook demonstrates yet another way of combining multiple separate plots into one single figure using the `subplot()` function. Each subplot will have its own set of axes. This is useful when related data are being plotted. The `subplot()` function creates a grid of plots automatically and eliminates the need to manually set the axes like we did in the case of the `axes()` function.

To use the subplot function

1. Declare the data for the x-axis and the y-axis as usual
2. Use the subplot() command for the first subplot
3. Call the plot() function and also set the title and label for the first plot
4. Use the subplot() command for the second subplot
5. Set the title, and label and call the plot function
6. Call the show() function

The subplot function has the following arguments:

```
subplot(nrows, ncols, subplot)
    -nrows: number of rows
    -ncols: number of columns
    -subplot: number of the subplot that is to be activated
```

The subplots are numbered row-wise starting from the top left. Note that the index value starts at 1 and not at 0. The `tight_layout()` function ensures adequate spacing between the subplots.

In [1]:

```
import matplotlib.pyplot as plt

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

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

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

#Call subplot() for the first plot. One row, 2 columns and subplot number = 1
plt.subplot(1,2,1)

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

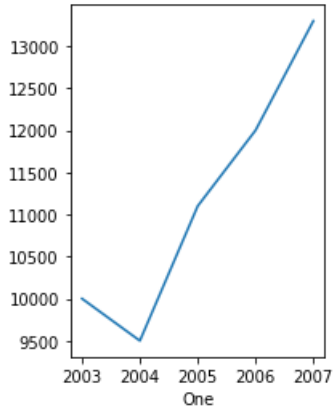
# Call the plot() function for the first plot
plt.plot(years, full_time_students)

#Call subplot() for the second plot. One row, 2 columns and subplot number = 2
plt.subplot(1,2,2)

# Set the title and label for the second plot and call the plot function for the second time
plt.xlabel('Two')
plt.title('Title Two')
plt.plot(years, part_time_students)

# tight_layout() provides spacing between the subplot()
plt.tight_layout()
#Last step: show the plot
plt.show()
```

Title One



Title Two

