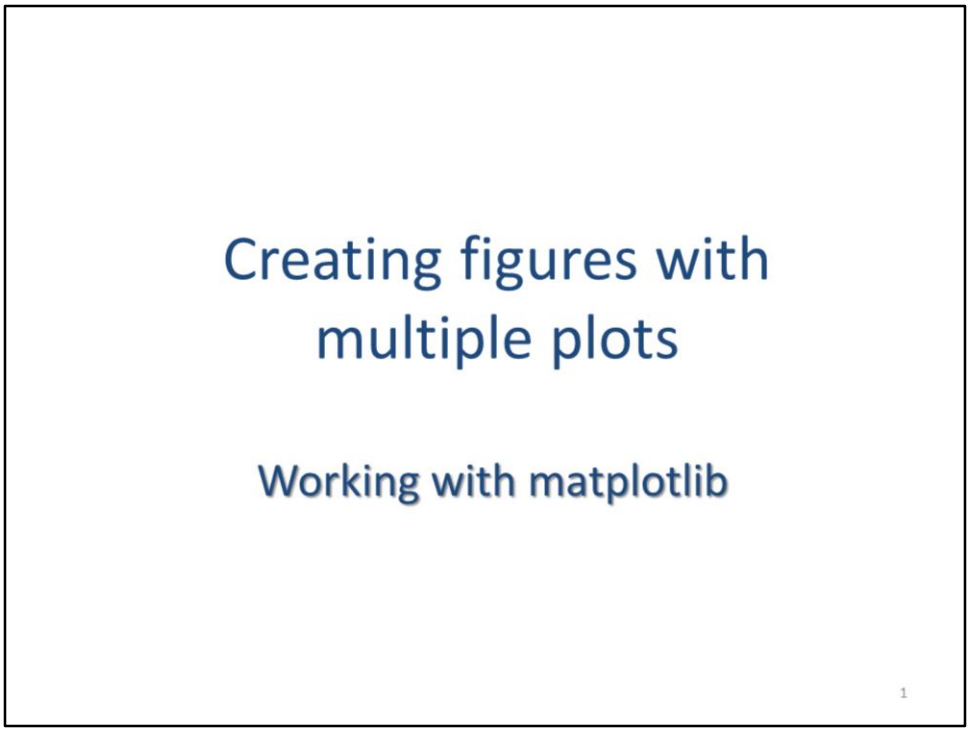


This video will discuss how to create figures that contain multiple plots.

Creating multiple subplots per figure

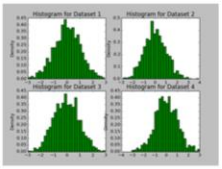
- Create subplot...

number of rows in figure

```
figure.subplot(nrows, ncols, plot_number)
```

number of columns in figure

active subplot position (see table below)



- Subplot will remain active until a new subplot is created.
  - plots, titles, etc. will be added to active subplot

plot_number

1	2
3	4

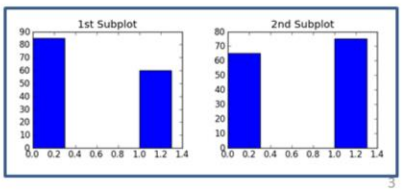
The subplot method allows multiple separate plots to be included in a single figure.

The nrows and ncols parameters specify the number of rows and columns in the figure. The plot_number parameter activates the corresponding plot – the table indicates how plots are numbered in a figure.

The subplot will remain active until another subplot statement activates a different plot number. Plots, titles, legends, and labels will be added to the subplot that is currently active.

Example – creating subplots

```
figure.subplot(1, 2, 1)   ← create and activate subplot 1
figure.bar (left = [0,1], height = [85, 60], width = 0.3)
figure.title("1st Subplot")
figure.subplot(1, 2, 2)   ← create and activate subplot 2
figure.bar (left = [0,1], height = [65, 75], width = 0.3)
figure.title("2nd Subplot")
figure.show()
```



This slide will show an example script that creates a figure with subplots.

The subplot statement segments the figure into 1 row with 2 columns and also activates the 1st subplot.

These statements create a bar plot and add a title to subplot 1.

The subplot statement is used again to activate the 2nd subplot.

A bar plot and title are now added to subplot 2.

Subplots size and position in figure

- Format size and position of subplots…

figure.subplots_adjust(left, bottom, right, top, wspace, hspace)

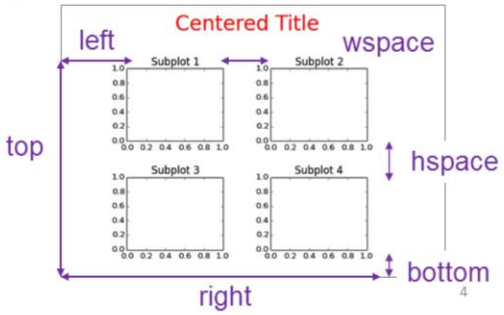
coordinates of subplot locations (see below), range from 0 to 1

- Title of set of subplots…

figure.suptitle('Centered Title', color = 'r')

- Refer to help page for more info for options:

http://matplotlib.org/api/pyplot_api.html#matplotlib.pyplot.subplot



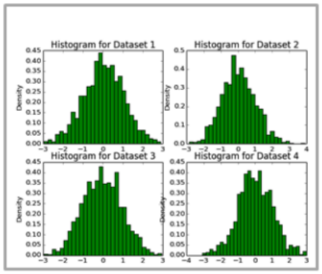
The **subplots_adjust** method can set the size and positions of the subplots within the figure. Note that subplot sizes are set indirectly by specifying the spacing between subplots and the figure margin sizes. The coordinates of the subplot locations range from 0 to 1. The illustration indicates how the parameters are defined. Setting the size and positions of subplots usually requires some trial and error.

A title can be added to the figure itself by using the **suptitle** method**.**

More information can be found on subplots in the pyplot documentation.

Example: creating subplots

```
import matplotlib.pyplot as figure
import numpy
for plotNum in [1,2,3,4]:
    data = numpy.random.randn(1000)
    figure.subplot(2, 2, plotNum)
    figure.hist(data, bins = 30, normed = True, color = "g")
    figure.ylabel("Density", size = 12)
    figure.title("Histogram for Dataset %s" % plotNum)
```



5

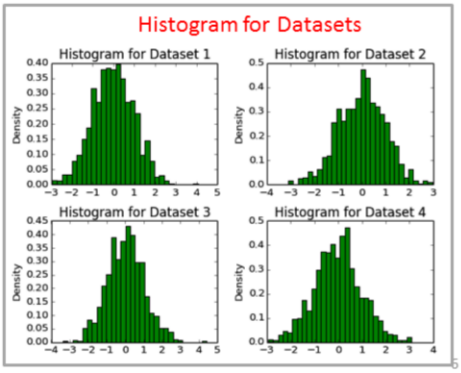
The next two slides will show an example that adjusts the sizes and positions of the subplots in a figure.

A for loop is used to iterate through each of the subplots – note that the 1st subplot is plot number 1. Inside the for loop…

- Example data for the subplot is created using numpy.
- The current plot number is activated. Note that the figure is divided into 2 rows and 2 columns.
- A histogram is created in the current subplot.
- The y-axis title and the plot titles are set.

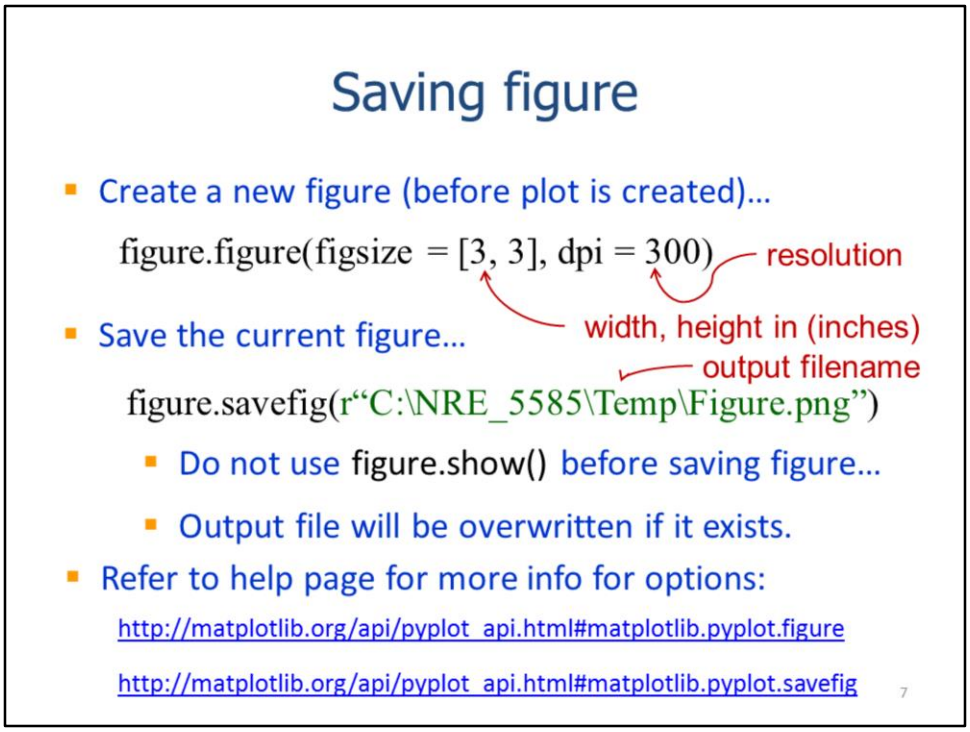
Example continued: creating subplots

```
figure.subplots_adjust(bottom=0.1, top=0.85,
        wspace=0.3,hspace=0.3)
figure.suptitle("Histogram for Datasets", size=20,
        color="r")
figure.show()
```



After the for loop finishes plotting the data within each subplot, the **subplots_adjust** method is used to set the positions and spacing between the subplots.

The **suptitle** method is used to set the "super" title for the figure.



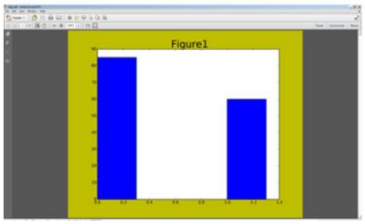
When you intend to save a figure, the **figure** method should be used to specify the dimensions and resolution of the figure. Note that these figure parameters should be set before any plots are created.

After the plots, titles, and other elements are added to the figure, the figure can be saved using the **savefig** method. The desired file extension should be included in the output file name. The **show** method should not be used before saving the figure. If a figure already exists, then it will be overwritten without any request for confirmation.

Additional information on creating figures can be found in the pyplot documentation.

Example: creating a figure

```
import matplotlib.pyplot as figure
figure.figure(figsize = [10,8])
dataLst=[85, 60]
figure.bar([0,1], dataLst, width = 0.3)
figure.title("Figure1", size = 30)
figure.savefig(r"C:\NRE5585\Fig1.pdf", facecolor="y",
    edgecolor="k")
```



Refer to help page for more info

8

This slide will show an example script that creates and saves a figure.

The figure size is set to 10 inches wide x 8 inches tall.

A bar plot is created within the figure.

The figure is saved as a .pdf file. Note that information on **facecolor** and **edgecolor** can be found in the pyplot documentation.