

Creating bar plots

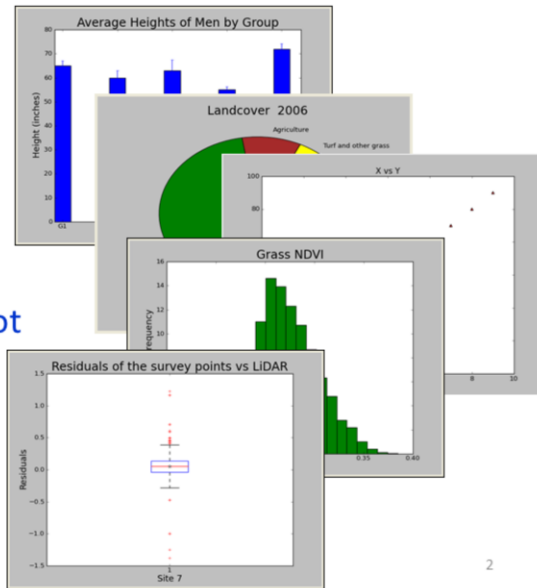
Working with Matplotlib

1

This video will introduce the **matplotlib** module for Python. We will explore matplotlib by creating bar plots.

The matplotlib module

- Qualitative Data
 - Bar plot
 - Pie plot
- Quantitative Data
 - Scatter plot/ Line plot
 - Histogram
 - Boxplot



The matplotlib module gives Python the ability to creating many different types of charts.

The matplotlib module

- Not an out-of-the-box module. Download from...
<http://matplotlib.org/downloads.html>
- Get python 2.7 version for windows (for ArcGIS 10.x)...
 - 1.3.1 — Latest stable version
 - [matplotlib-1.3.1.tar.gz](#)
 - [matplotlib-1.3.1.win-amd64-py2.6.exe](#)
 - [matplotlib-1.3.1.win-amd64-py2.7.exe](#)
- `pyplot` submodule used for graphs...
`import matplotlib.pyplot as figure` ← assigns figure variable to pyplot object
- For documentation on `matplotlib.pyplot`, go to
http://matplotlib.org/api/pyplot_summary.html#

3

Matplotlib is not an out-of-the-box module and must be downloaded. The version downloaded should match the version of Python – the correct version for python 2.7 is shown here.

The **pyplot** submodule of **matplotlib** is used for creating charts. The statement shown here imports only the pyplot sub-module and assigns it a more convenient name.

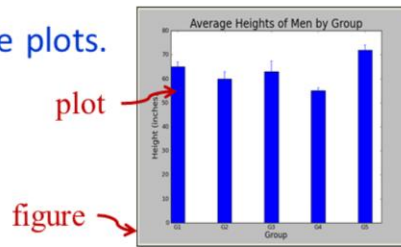
Documentation for pyplot can be found at the link provided.

Terms: figure vs. plots

- A **figure** is the object containing the plots, titles, labels, etc.
 - `pyplot` is the figure object...
- A **plot** is the actual data in the figure (e.g. bars, data points, etc.)
- A figure can contain multiple plots.
- To display a figure...

`figure.show()`

Note: displaying figure clears figure settings



In this lecture, I'll use the term **figure** to refer to the object that contains the plot, title, labels, etc. The `pyplot` sub-module is the figure object to which we'll add the plot.

The **plot** displays that the actual data in the figure in the form of bars, data points, etc. A single figure can contain multiple plots. A figure is displayed by using the **show** method.

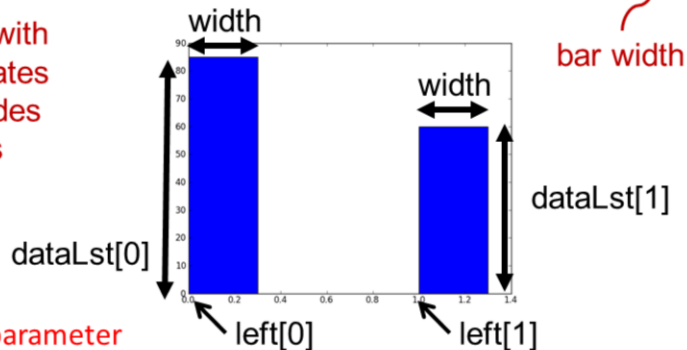
Bar plot – vertical

```
import matplotlib.pyplot as figure
```

```
dataLst = [85, 60]
```

```
barPlot = figure.bar(left = [0,1], height = dataLst, width = 0.3)
```

list/array with
x coordinates
for left sides
of bars



5

This slide will show how to create a vertical bar plot.

The data that will be plotted should be stored in a list or array.

The figure's **bar** method will create the plot.

The **left** parameter value is a list that indicates the position of the bar on the x-axis – the list must provide a bar position for each value in the data list.

The **height** parameter is the dataLst – the values in the list will set the heights for each bar.

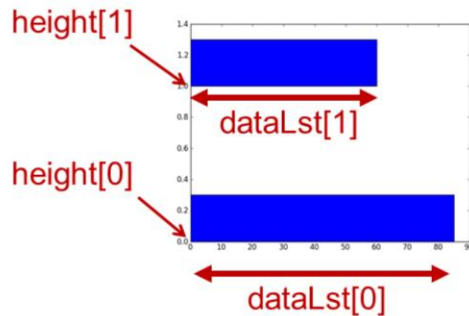
The **width** parameter is a decimal number that specifies the width of the bars.

Note that the **left**, **height**, and **width** parameters are required.

Bar plot –horizontal

```
dataLst = [85, 60]
```

```
barPlot = figure.bar (left = 0, height = [0,1], width = dataLst,  
orientation = 'horizontal')
```



6

This slide shows how to create a horizontal bar plot. The data are specified in a list or an array. The figure's **bar** method creates the plot. For a horizontal bar chart, the **left** parameter will be zero.

The **height** will be the positions of the bars along the y-axis.

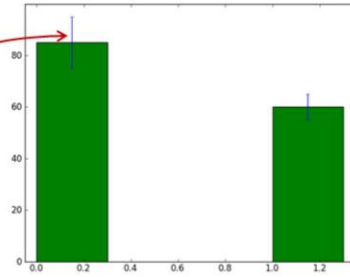
The **width** of the bar will be the data list. The orientation parameter is set to "horizontal" (note that "vertical" is the default).

Bar plot – color, error bars

`y_errorLst = [10, 5]` ← list/array with magnitudes of error (i.e. std. error, std. dev.)

`barPlot = figure.bar (left, height, width, color = 'green')` ← string (single dataset) or list (multiple datasets)

`yerr = y_errorLst` ← use `xerr` for horizontal bar plots



- Refer to help page for more info for optional parameters

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

7

This slide will show how to set error bars and change the bar color for a vertical bar plot.

The magnitudes of the error bars should be contained in a list or array. Standard errors or standard deviations are typically used for error bars.

The color of the bar can be specified through the **color** parameter. This color value may be a single value, if only a single dataset is plot, or a list of values if multiple datasets are plot.

The error bar magnitudes are set through the **yerr** parameter. The value for this parameter is a list or array containing 1 value for each item in the dataLst. If the plot orientation is “horizontal”, then use the **xerr** parameter in place of **yerr**.

Refer to the documentation page indicated by the link to find more options for bar plots.

Bar plot – stacked bar plot

```
heights1 = [85, 60] # dataset 1 list...
```

```
heights2 = [45, 65] # dataset 2 list...
```

```
left = [0, 1] # left side coordinate list/array
```

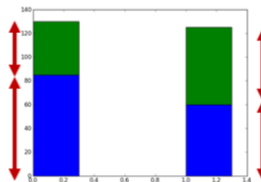
```
barPlot1 = figure.bar (left, heights1, width = 0.3)  
└─┬─┘  
plot for bottom dataset
```

```
└─┬─┘  
plot for top dataset  
barPlot2 = figure.bar (left, heights2, width = 0.3,  
bottom = heights1)
```

y-axis coordinate
for bottom of bar

heights2[0]

heights1[0]



8

Multiple datasets can be included in a bar plot. In this slide, we'll work with a stacked vertical bar plot; the same approach can be used for horizontal bar plots.

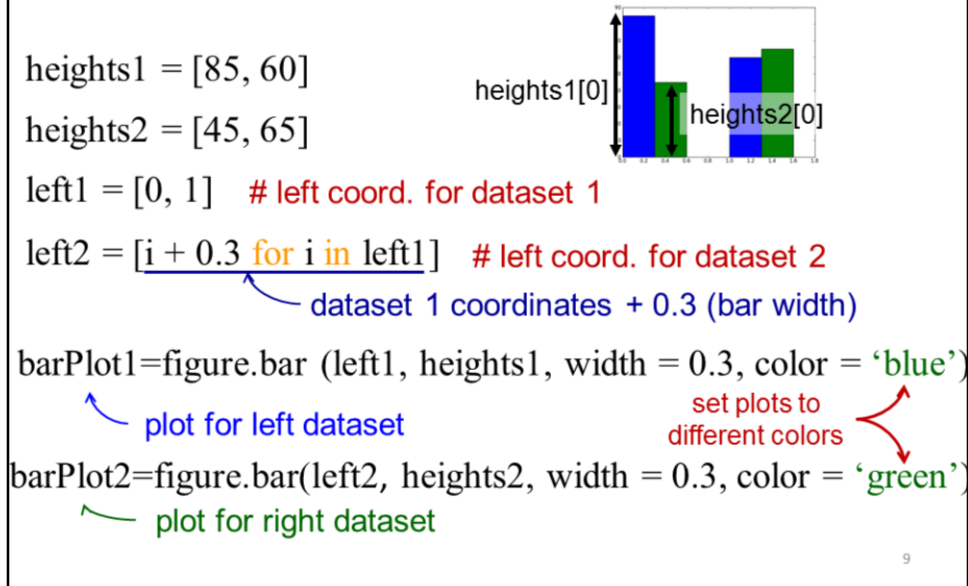
When multiple datasets will be plotted, the data for each dataset should be stored in a list or array. The left positions of the bar plots should also be specified in a list.

The **bar** method will be used to plot the first dataset – the dataset values will be used to set the bar heights.

The **bar** method is used again, on the same figure, to plot the second dataset.

In the second bar plot, the values from the first dataset are used to set the **bottom** parameter – this will place the bars from the 2nd dataset on top of the bars from the 1st dataset. Additional bar statements can be used to plot additional datasets in the figure.

Bar plot – side-by-side bar plot



A side-by-side bar plot can be created when multiple datasets are plotted.

The values for each dataset need to be specified in separate lists. The left positions for the 1st dataset should be specified in a list as usual.

A second lists of left positions needs to be created for the 2nd dataset – the positions should be shifted to the right of the 1st dataset positions by the width of the bar. A list comprehension is an efficient way to add the bar width to each of the left positions for the 1st dataset. Plotting a 3rd dataset would require a left position list that is shifted one bar width to the right of the 2nd left position list.

Each of the datasets are then plotted using the appropriate left position and height lists. A different bar color should be used for each plot.