Educate!

Annotation is enabled on the page !

Basics Plots – Lines, Bars, Pies And Scatterplots

September 22, 2016 by Admin (http://courselibrary.shaveensingh.com/forum/profile/admin/)

This section will revise and/or practice the basic chart types supported in *matplotlib* and their different parameter settings. We will use some basic or random datasets to demonstrate the various plotting function.

Ensure that you practise creating the following plots as it is a prerequisite task before attending the workshop.

To start with, include the following three lines of code on the top for every visualization you will do using iPython notebook:

```
1 %matplotlib inline #iPython magic to tell the notebook to show all visua
2 import matplotlib.pyplot as plt  #to import matplotibs plotting function
3 import numpy as np  #to import numpy for its utility function
```

Line plots

Purpose: Line plots are one of the most basic chart types out there. It is useful for showing trends in data — usually for time series data with many time points.

matplotlib function: plot(x, y, <other parmeters>)

Parameters:

- x: The x-coordinates of the lines or markers.
- y: The y-coordinates of the lines or markers.

Sample code:

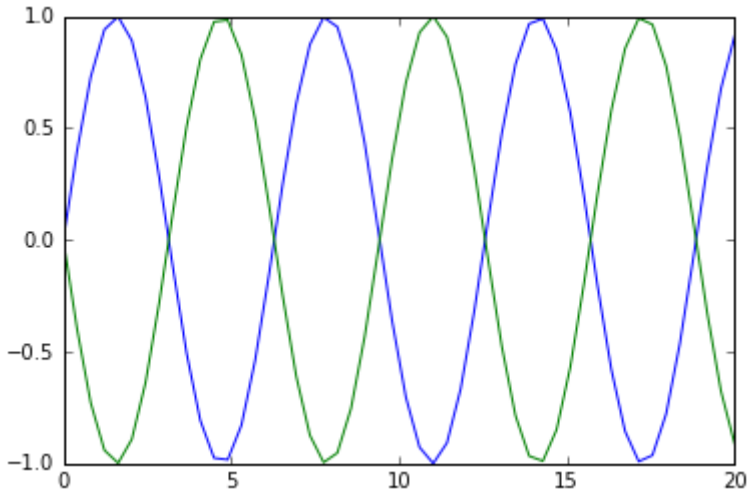
Scroll Depth: 47%

<< View Python Editor

Annotation is enabled on the page !

```
x = np.linspace(0, 20)
y1 = np.sin(x)
y2 = np.sin(x - np.pi)

plt.figure()

plt.plot(x, y1)
plt.plot(x, y2)
```

Output:



(http://educate.shaveensingh.com/wp-content/uploads/2016/09/p1.png) Click to s*ee more examples on line plot:* ↩

Commonly used optional parameters:

- color: Set the color of the line.
- linestyle: Set the line style, e.g., solid, dashed, or none.
- linewidth: Set the line thickness.
- marker: Set the marker style, e.g., circles, triangles, or none.
- markersize: Set the marker size.
- label: Set the label for the line that will show up in the legend.

More example:

```
x = np.linspace(0, 20)
y1 = np.sin(x)
y2 = np.sin(x - np.pi)

plt.figure()

```

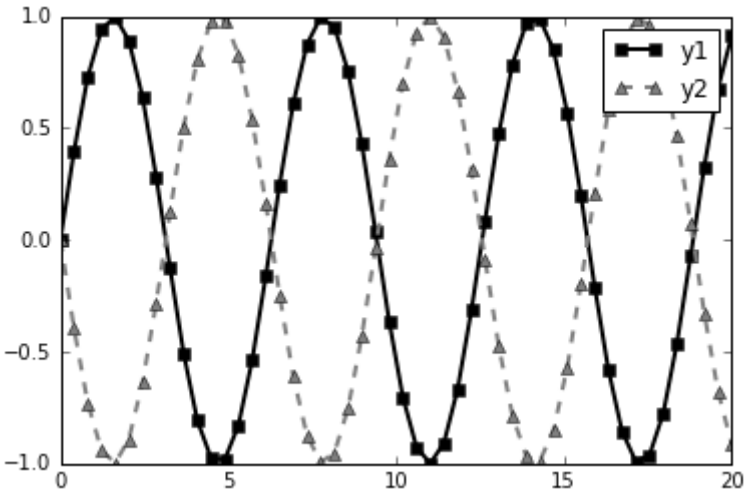
Scroll Depth: 47%

<< View Python Editor

```
plt.plot(x,
         y1,
         color='black',
         linestyle='-',
         linewidth=2,
         marker='s',
         markersize=6,
         label='y1')

plt.plot(x,
         y2,
         color='gray',
         linestyle='--',
         linewidth=2,
         marker='^',
         markersize=6,
         label='y2')

plt.legend()
;
```

Annotation is enabled on the page !

Output:



(http://educate.shaveensingh.com/wp-content/uploads/2016/09/p1-1.png)

Vertical bar charts

Purpose: Comparing categories OR showing temporal trends in data with few (< 4) time points.

Scroll Depth: 47%

<< View Python Editor

matplotlib function: bar(left, height)

matplotlib function: bar(left, height)

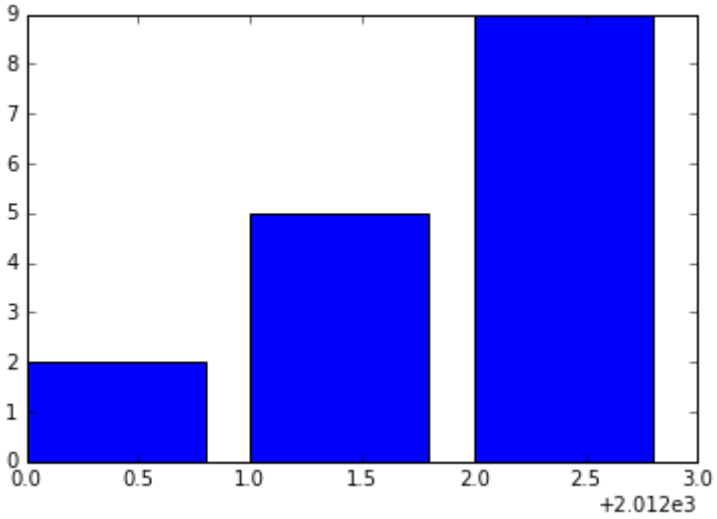
Annotation is enabled on the page !

Parameters:

- left: The x coordinate(s) of the left sides of the bars.
- height: The height(s) of the bars.

```
1 years = np.arange(2012, 2015)
2 values = [2, 5, 9] plt.figure()
3 plt.bar(years, values)
4 ;
```

Output:



(http://educate.shaveensingh.com/wp-content/uploads/2016/09/p1-3.png)

Click to s*ee more examples on vertical bar charts:* ↩

Commonly used parameters:

- color: Set the color of the bars.
- edgecolor: Set the color of the lines on the edges of the bars.
- width: Set the width of the bars.

- align: Set the alignment of the bars, e.g., center them on the x coordinate(s).
- label: Set the label for the bar that will show up in the legend.

Scroll Depth: 47%

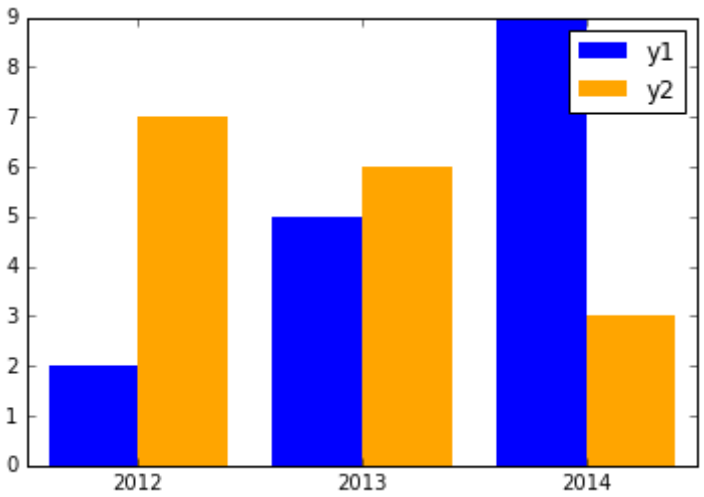
<< View Python Editor

Annotation is enabled on the page !

```
years = np.arange(2012, 2015)
category1_values = [2, 5, 9]
category2_values = [7, 6, 3]

plt.figure()

plt.bar(years - 0.2,
        category1_values,
        color='blue',
        edgecolor='none',
        width=0.4,
        align='center',
        label='y1')

plt.bar(years + 0.2,
        category2_values,
        color='orange',
        edgecolor='none',
        width=0.4,
        align='center',
        label='y2')

plt.xticks(years, [str(year) for year in years])

plt.legend()
;
```

4

Output:



(http://educate.shaveensingh.com/wp-content/uploads/2016/09/p1-2.png)

2

Horizontal bar charts

Scroll Depth: 47%

<< View Python Editor

Annotation is enabled on the page !

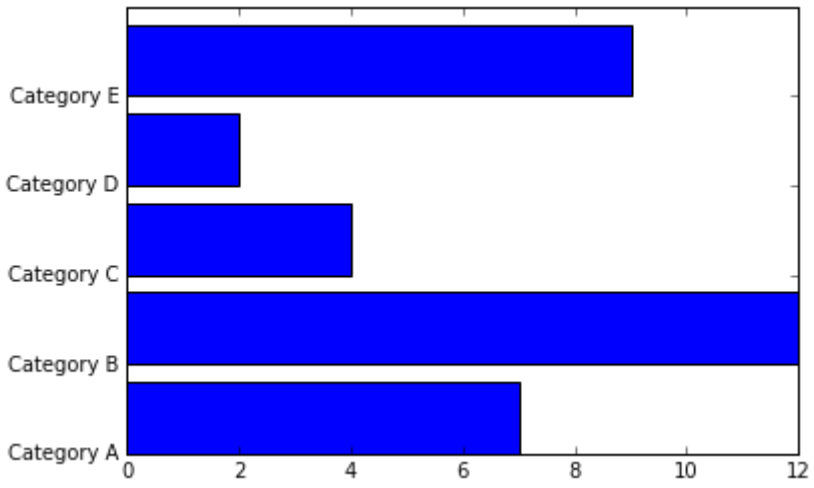
Purpose: Comparing categories.

matplotlib function: barh(bottom, width)

- bottom: The y coordinate(s) of the bars.
- width: The width(s) of the bars.

```
categories = ['A', 'B', 'C', 'D', 'E']
values = [7, 12, 4, 2, 9]

plt.figure()

plt.barh(np.arange(len(categories)), values)

plt.yticks(np.arange(len(categories)),
           ['Category {}'.format(x) for x in categories])
;
```

Output:



Click to see more examples on horizontal bar charts: ↩

Pie charts

Purpose: Displaying a simple proportion.

matplotlib function: pie(sizes)

- sizes: The size of the wedges as either a fraction or number.

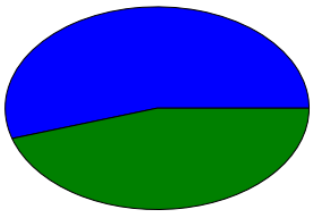
Scroll Depth: 47%

<< View Python Editor

Annotation is enabled on the page !

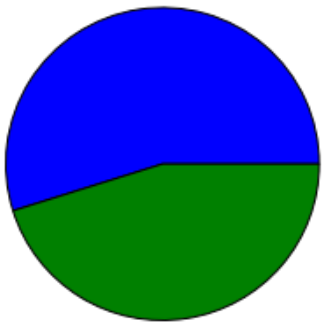
```
counts = [17, 14]

plt.figure()

plt.pie(counts)
;
```

Output:



```
counts = [17, 14]

plt.figure(figsize=(4, 4))

plt.pie(counts)
;
```

Output:



Commonly used parameters:

- colors: Set the colors of the wedges.
- labels: Set the labels of the wedges.
- startangle: Set the angle that the wedges start at.
- autopct: Set the percentage display format of the wedges.

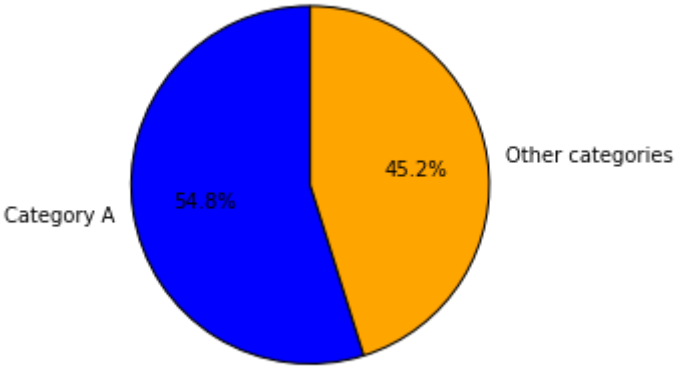
Annotation is enabled on the page !

Click to see more examples on pie charts: ↩

Another example:

```
counts = [17, 14]

plt.figure(figsize=(4, 4))

plt.pie(counts,
    colors=['blue', 'orange'],
    labels=['Category A', 'Other categories'],
    startangle=90,
    autopct='%1.1f%%')
;
```

Output:



Scatter plots

Purpose: Displaying relationships between variables.

matplotlib function: scatter(x, y)

- x, y: The values for the two variables.

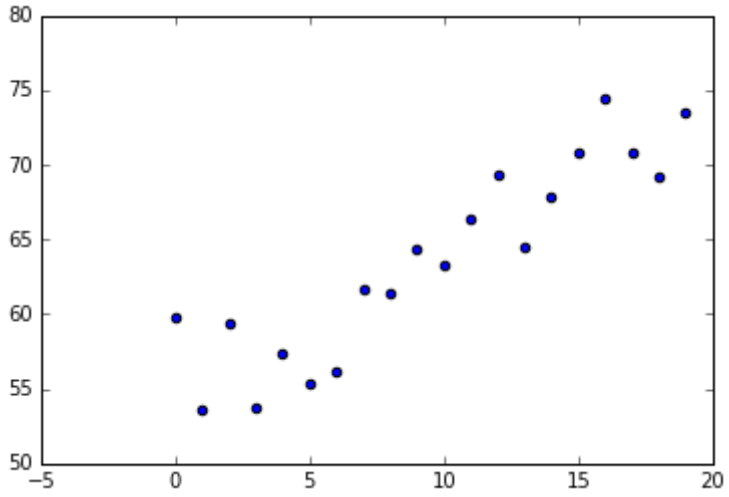
Example:

```
x = range(20)
y = np.arange(50, 70) + (np.random.random(20) * 10.)
```

```
4 plt.figure()
5
6 plt.scatter(x, y)
7 ;
```

Annotation is enabled on the page !

Output:



Commonly used parameters:

- c: Set the color of the markers.
- s: Set the size of the markers.
- marker: Set the marker style, e.g., circles, triangles, or squares.
- edgecolor: Set the color of the lines on the edges of the markers.

Click to see more examples on scatter plots: ↩

You have now completed this section.

« Previous Unit **(http://courselibrary.shaveensingh.com/module-3/tutorial-b-basic-necessities-for-matplotlib/)** Next Unit » **(http://courselibrary.shaveensingh.com/module-3/plotting-distributions-histograms-and-box-plots/)**

Annotation is enabled on the page !

(http://www.themekiller.me/)

Scroll Depth: 47%

<< View Python Editor