Introduction to Matplotlib

Created By

Basic plots for data visualization

Data visualization is highly useful to explore the insights of data by Visualizing it on various types of plots. The few types of most used plot types are:

1. Line plot (most basic)
2. Bar plot (used for categorical variables)
3. Histogram (used for numeric variables)
4. Scatter plot (used to find correlation between variables)
5. Area plot / Stack plot (plot various variable that contribute to some information)
6. Pie plot (gives rough idea about data demographics)

Tool Used

These notes are based on the Matplotlib library of python. I am using python version 3.9.7 on Spyder IDE from anaconda.

Line plots

This is the most basic type of plot, that draws the connections between two data points via a solid line. These graphs are used to see trends in data and how one variable is changing w.r.t another. It plots a line between two variables (generally called as X and Y).

```
# plotting a line map
import matplotlib.pyplot as plt
X = [1,2,3,4]
```

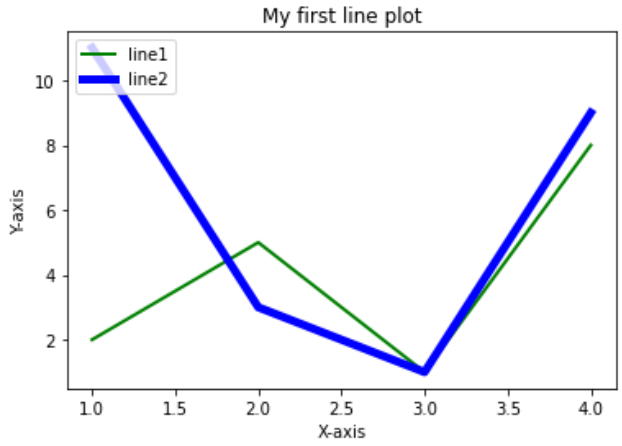
```
Y=[2,5,1,8]

# plot the line with green color and width=2
plt.plot(X,Y, 'g', label='line1', linewidth=2)
plt.title('My first line plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

The output of this set of python code is:

Now, let's add another variable Z and plot this on the same graph figure. The python code to do this is:

```
X = [1,2,3,4]
Y=[2,5,1,8]
Z=[11,3,1,9]
# line 1 with green color
plt.plot(X,Y, 'g', label='line1', linewidth=2)
# line 2 with blue color
plt.plot(X,Z,'b',label='line2',linewidth=5)
plt.legend(['line1', 'line2'],loc='upper left')
plt.title('My first line plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.show()
```

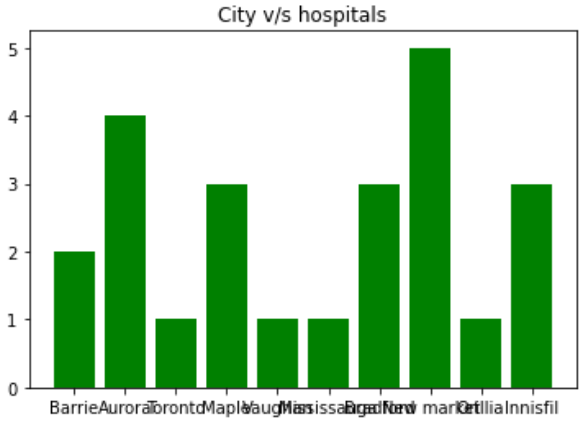
The graph looks like as shown below:



Bar Graphs

The bar graphs are the graphs commonly used for categorical data visualization. For e.g. if we want to see what is number of hospitals in various 10 cities of a province, we may plot city names on x-axis and number of hospitals on y-axis.

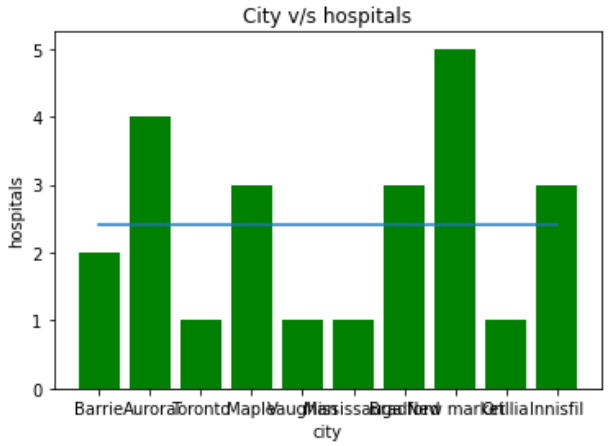
```
city = ['Barrie','Aurora','Toronto','Maple','Vaughan','Mississauga','Bradford','New market','Orillia','Innisfil']
hospitals = [2,4,1,3,1,1,3,5,1,3]

plt.bar(city, hospitals, label='bar 1', color='g')
plt.title('City v/s hospitals')
plt.xlable('city')
plt.ylable('hospitals')
plt.show()
```



The bar graphs gives us insights about the data. For example this bar graphs tells us that new market has highest number of hospitals and four cities Toronto, maple, Mississauga, and Orillia has only one hospital. From bar graphs, we can also get some statistical parameters such as average, mode, and median. For example, if we want to plot a line showing average number of hospitals, we may use Numpy or script.stats to get the mean value. The python code to show the mean line is copied below:

```
city = ['Barrie','Aurora','Toronto','Maple','Vaughan','Mississauga','Bradford','New market','Orillia','Innisfil']
hospitals = [2,4,1,3,1,1,3,5,1,3]
import numpy as np
average = np.sum(hospitals)/len(hospitals)
average = [average]*10
plt.bar(city, hospitals, label='bar 1', color='g')
plt.plot(city, average)
plt.title('City v/s hospitals')
plt.xlabel('city')
plt.ylabel('hospitals')
plt.show()
```

The resultant bar graph is:



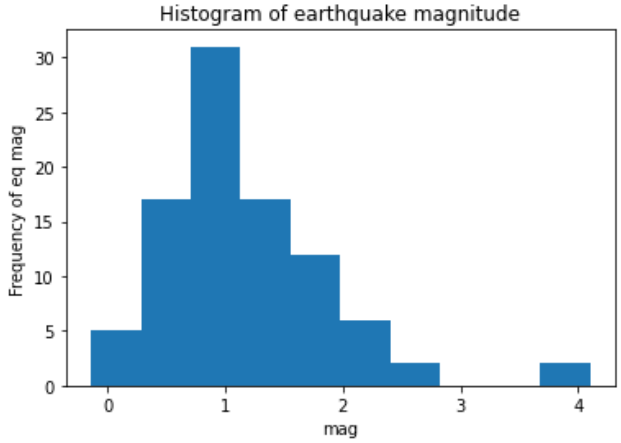
We may add more statistical parameters to the bar graph such as mode and median value.

Histogram plots

The histogram plots are mainly used to plot frequency in a particular category. For example, we have a data showing earthquake that happened in the USA in a day. The data has columns such as latitude, longitude, mag, magType, mag error, depth, depthError, and more. We want to plot the histogram of magnitude of earthquakes and another histogram shows the magType.

```
# import pandas to read csv file
import pandas as pd
data = pd.read_csv('all_day.csv')
data.info() # check the info of each column
data = data.dropna() # drop rows with missing enteries.

plt.hist(data['mag'])
plt.title('Histogram of earthquake magnitude')
plt.xlabel('mag')
plt.ylabel('Frequency of eq mag')
plt.show()
```

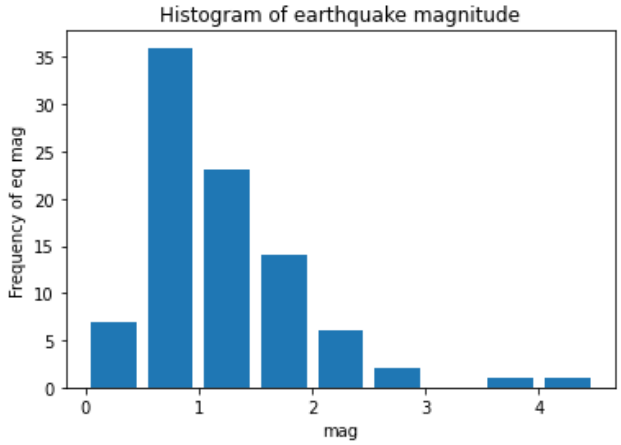
The histogram showing earthquake magnitudes are shown below. From this histogram, it is clear that maximum number of earthquake (more than 30) recorded in a day in the USA has an intensity of 1 and only a few (less than 5).



The information provided by the above histogram is not precise. We have a few tricks to improve the visual and information content from the histogram.

For plotting a histogram we need to have bins on X-axis and count on Y-axis. We can reduce the bin size to see the more details about the histogram. The improved histogram is shown below:

```
bins = [0,0.5,1,1.5,2,2.5,3,3.5,4,4.5]
plt.hist(data['mag'], bins, histtype='bar', rwidth=0.8)
plt.title('Histogram of earthquake magnitude')
plt.xlabel('mag')
plt.ylabel('Frequency of eq mag')
plt.show()
```

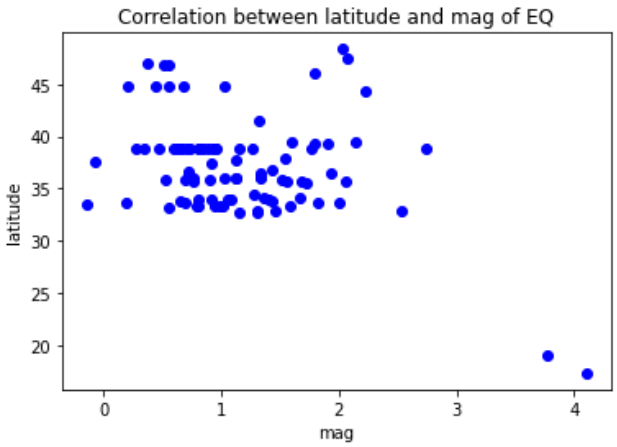


The histogram shows that the number of EQs having an intensity less than 0.5 is 6, the number of EQs having an intensity between 0.5 and 1 is 35, and so on and so forth we can get more deeper information from the histogram.

Scatter plots

These plots are used to see the correlation between two variables (when plotted in 2D) or three variables (when plotted in 3D). This is quite a useful way to find the outliers as well. For the EQ data, let's plot the scatter plot between the mag of EQ and the latitude of the place where EQ happened.

```
X=data['mag']
Y= data['latitude']
plt.scatter(X, Y, color='b')

plt.title('Correlation between latitude and mag of EQ')
plt.xlabel('mag')
plt.ylabel('latitude')
plt.show()
```



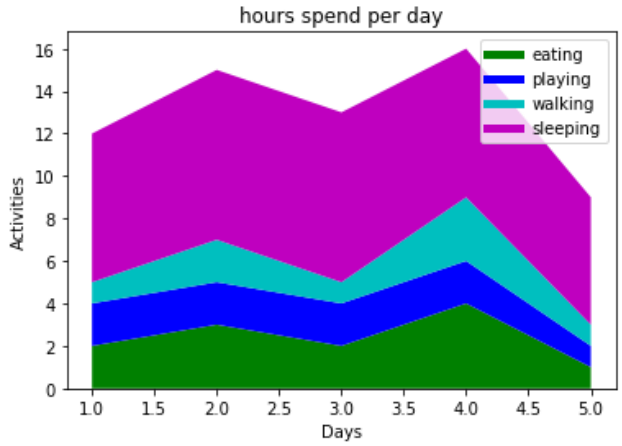
The scatter plots show that the max number of EQ has an intensity between 0.5 to 2.5, and happened in the places where the latitude is in the range of 32 to 45. Only two instances of EQ happened to have intensity around 4 and latitude less than 20. These two scatter points are showing the outliers in the data.

Area plot/Stack plot

The area plot is used to track changes in the dependent variable w.r.t independent variable. The number of variables might be more than one which is contributing towards one big goal. It is just like line plots but area under the line is filled with colours. For example, for certain number of days, we have the number of hours spend on various activities such as eating, playing, sleeping and walking. We may use area plot or stack plot to understand the data insights.

```
days=[1,2,3,4,5]

eating = [2,3,2,4,1]
playing=[2,2,2,2,1]
walking = [1,2,1,3,1]
sleeping = [7,8,8,7,6]

plt.plot([],[],'g', label='eating', linewidth=5) # empty lines with color green
plt.plot([],[],'b', label='playing', linewidth=5)
```

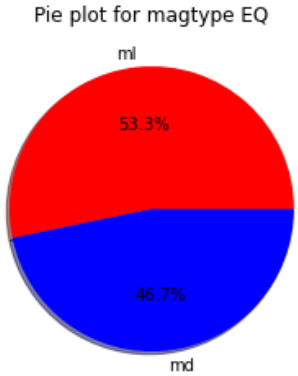
```
plt.plot([],[],'c', label='walking', linewidth=5)
plt.plot([],[],'m', label='sleeping', linewidth=5)

plt.stackplot(days, eating, playing, walking, sleeping, colors=['g','b','c','m'])

plt.xlabel('Days')
plt.ylabel('Activities')
plt.legend(['eating','playing','walking','sleeping'], loc='upper right')
plt.title('hours spend per day')
```



Pie chart

Pie plot is line an area plot, but on a circular disk, where each slice of the disk is shown by the percentage of the data it represents that means the higher contributing category in the data is shown by a bigger slice and so on and so forth. Let's take our EQ data example and see the contributions of magType in EQ data.

```
from collections import Counter

C  = Counter(data['magType'])
data_pie = [C['ml'], C['md']]

plt.pie(data_pie, colors=['r','b'], labels = ['ml','md'], shadow=(True), autopct='%1.1f%%')
plt.title('Pie plot for magtype EQ')
plt.show()
```



We can see from the magType pie chart that 53.3% of the EQ are of type ‘ml’, and rest 46.7% of the EQ are of type ‘md’.