DATA VISUALISATION

USING PYTHON

VISUALISATION: is the representation of an object, situation, or set of information as a chart or other image (the formation of a mental image of something.)

Visualization is the easiest way to analyze and easily understand information.

Python offers many graphics libraries that include lots of different features. Out of these libraries we are going to learn MATPLOTLIB.

MATPLOTLIB: Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

Why Matplotlib

✓ Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits.

✓ Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code.

✓ It supports interactive and non-interactive plotting and can save images in several output formats (PNG, PS & Others)

✓ Pyplot module of matplotlib library supports a very wide variety of graphs and plots like histograms, barcharts power spectra, error charts etc. It is used along with NumPy to provide an environment for MatLab.

How to install Matplotlib

• Python shell must be installed in your System
  ✓ Open command prompt as an Administrator
  ✓ Go back to root directory by typing cd\
  ✓ Now type pip install matplotlib

After successful installation, we are ready to use matplotlib and its modules
Using pyplot we can plot following type of charts:

<table>
<thead>
<tr>
<th>S. No</th>
<th>Plot/Chart Name</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Line Chart</td>
<td><img src="image" alt="Line Chart Example" /></td>
<td><img src="image" alt="Line Chart Example" /></td>
</tr>
<tr>
<td>2</td>
<td>Bar Chart (Vertical)</td>
<td><img src="image" alt="Bar Chart Example" /></td>
<td><img src="image" alt="Bar Chart Example" /></td>
</tr>
</tbody>
</table>

Let's understand terms (Title, Labels, x label, y label etc) used for different sections in plots/chart through this diagram.
Some important Functions which are required to draw different Plots are:

<table>
<thead>
<tr>
<th>S.No</th>
<th>Function Name</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>plot()</td>
<td>If you provide a single list or array to the <code>plot()</code> command, matplotlib assumes it is a sequence of y values, and automatically generates the x values for you. Since python ranges start with 0, the default x vector has the same length as y but starts with 0. Hence the x data are [0,1,2,3] .</td>
</tr>
<tr>
<td>2</td>
<td>xlabel()</td>
<td>Used to mention Xlabel with chart</td>
</tr>
<tr>
<td>3</td>
<td>ylabel()</td>
<td>Used to mention Ylabel with chart</td>
</tr>
<tr>
<td>4</td>
<td>title()</td>
<td>To specify Title of the chart</td>
</tr>
<tr>
<td>5</td>
<td>bar()</td>
<td>To draw bar chart</td>
</tr>
<tr>
<td>6</td>
<td>barh()</td>
<td>To draw horizontal bar diagram</td>
</tr>
<tr>
<td>7</td>
<td>Pie()</td>
<td>To draw pie chart</td>
</tr>
<tr>
<td>8</td>
<td>Figure()</td>
<td>To specify size of plot area</td>
</tr>
</tbody>
</table>
LINE GRAPH

1. A line Chart/Plot/Graph is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments.

Let's take an example to draw Line Graph

The table shows passenger car fuel rates in miles per gallon for several years. Make a LINE GRAPH of the data. During which 2-year period did the fuel rate decrease?

RATE:   21.0  20.7  21.2  21.6

Solution

```
import matplotlib.pyplot as p
p.plot(Yr,rate)
p.show()
```

Output:

In the above Figure it is not clear What is 2000,2002 etc, or 20.6,21.2............So We should put Labels on it. To put Label we will do following changes in the Code

```
import matplotlib.pyplot as p
p.plot(Yr,rate,color='red')
p.xlabel('Year') # To Put Label At X Axis
p.ylabel('Rate') # To put Label At Y Axis
p.title('Fuel Rates in every Two Year') # To Write Title of the Line Chart
p.show()
```

You can change Line Style by specifying pattern with plot function.
FOR EXAMPLE
```
p.plot( Y, '—') # To draw dashed line
p.plot( Y, '-.') # To draw dash-dot line
```
BAR GRAPH:

A bar graph / bar chart / bar diagram is a visual tool that uses bars (Vertical or Horizontal) to represent/compare categorical data. Let's take an example to make it clear

1. The number of bed-sheets manufactured by a factory during five consecutive weeks is given below.

<table>
<thead>
<tr>
<th>Week</th>
<th>First</th>
<th>Second</th>
<th>Third</th>
<th>Fourth</th>
<th>Fifth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Bed-sheets</td>
<td>600</td>
<td>850</td>
<td>700</td>
<td>300</td>
<td>900</td>
</tr>
</tbody>
</table>

Draw the bar graph representing the above data.

SOLUTION: PYTHON CODE

```python
import matplotlib.pyplot as p

x=['First', 'Second', 'Third', 'Fourth', 'Fifth']  # Assigning Data for X axis
y=[600, 850, 700, 300, 900]                         # Assigning Data for Y axis

p.title('Production By Factory')
p.xlabel('Week')
p.ylabel('No. of Bed Sheets')
p.bar(x, y, color='Blue', width=.5)  # To draw bar graph, with each bar of .50 width
p.show()
```

OUTPUT: fig 1
We can draw grids also with the chart by adding `plt.grid(True)` with the above coding. Output will be as follows.

If we want to plot Horizontal bar graph for the same question then we use `barh()` function and we have to interchange x and y labels as shown below.

```python
import matplotlib.pyplot as plt

x=['First', 'Second', 'Third', 'Fourth', 'Fifth']  # Assigning Data for X axis
y=[600, 850, 700, 300, 900]  # Assigning Data for Y axis

plt.title('Production By Factory')
plt.ylabel('Week')
plt.xlabel('NO. of Bed Sheets')
plt.barh(x, y, color='Blue')  # To draw bar graph, with each bar of 0.50 width
plt.show()
```

And Output will be as: fig 1.2
2. The number of students in 7 different classes is given below. Represent this data on the bar graph.

<table>
<thead>
<tr>
<th>Class</th>
<th>6th</th>
<th>7th</th>
<th>8th</th>
<th>9th</th>
<th>10th</th>
<th>11th</th>
<th>12th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Students</td>
<td>130</td>
<td>120</td>
<td>135</td>
<td>130</td>
<td>150</td>
<td>80</td>
<td>75</td>
</tr>
</tbody>
</table>

**SOLUTION : PYTHON CODE**

```python
import matplotlib.pyplot as p

x=[6,7,8,9,10,11,12]  # Assigning Data for X axis
y=[130,120,135,130,150,80,75]  # Assigning Data for Y axis

p.title('Class Strength')
p.xlabel('CLASS')
p.ylabel('NO. of Students')
p.bar(x,y,color='green',width=.40)  # To draw bar graph, with each bar of .40 width
p.show()
```

**OUTPUT : fig 2**
3. The number of students and Pass percentage in 7 different classes is given below. Represent this data on the bar graph.

<table>
<thead>
<tr>
<th>Class</th>
<th>6&lt;sup&gt;th&lt;/sup&gt;</th>
<th>7&lt;sup&gt;th&lt;/sup&gt;</th>
<th>8&lt;sup&gt;th&lt;/sup&gt;</th>
<th>9&lt;sup&gt;th&lt;/sup&gt;</th>
<th>10&lt;sup&gt;th&lt;/sup&gt;</th>
<th>11&lt;sup&gt;th&lt;/sup&gt;</th>
<th>12&lt;sup&gt;th&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Students</td>
<td>130</td>
<td>120</td>
<td>135</td>
<td>130</td>
<td>150</td>
<td>80</td>
<td>75</td>
</tr>
<tr>
<td>Pass Percentage</td>
<td>70</td>
<td>80</td>
<td>76</td>
<td>89</td>
<td>90</td>
<td>95</td>
<td>100</td>
</tr>
</tbody>
</table>

**SOLUTION : PYTHON CODE**

```python
import matplotlib.pyplot as p
x=[6,7,8,9,10,11,12]  # Assigning Data for X axis
y1=[130,120,135,130,150,80,75]  # NO. of Students for Y axis
x1=[5.5,6.5,7.5,8.5,9.5,10.5,11.5]
y2=[70,80,76,89,90,95,100]  # Pass Percentage for Y axis
p.title('Class Strength & Pass Percentage')
p.bar(x,y1,color='green',width=0.4,label='No.of Students')
p.bar(x1,y2,color='r',width=0.4,label='Pass Percentage')
p.xlabel('Class')
p.legend()
p.show()
```
PIE CHART

A pie chart is a circular chart divided into sectors which is proportional to the whole quantity it represents.

A pie chart displays data, information, and statistics in an easy-to-read 'pie-slice' format. Different sizes of slice indicates how much of one data element exists.

4. The number of students and Pass percentage in 7 different classes is given below. Represent this data on the Pie Chart

<table>
<thead>
<tr>
<th>Class</th>
<th>9th</th>
<th>10th</th>
<th>11th</th>
<th>12th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Students</td>
<td>130</td>
<td>150</td>
<td>80</td>
<td>75</td>
</tr>
</tbody>
</table>

SOLUTION: PYTHON CODE
OUTPUT : fig 4

Note : If we also want to show how much percent each slice is representing then we have to specify value for `autopct='%.2f%%'` parameter with the `pie()` function as below:

```
import matplotlib.pyplot as plt
l=[9,10,11,12] # Defining labels
stud=[130,150,80,75] # Assigning Data for pie
plt.title ('Class Strength')
plt.pie (stud, labels=l, colors=['gold', 'red', 'pink', 'lightcoral'], autopct='% .2f%%')
plt.show()
```

Now out will be as
Use `figure(figsize=[width,height])` function to specify size of Plot.

For example:

```python
import matplotlib.pyplot as p
l=[9,10,11,12]   # Defining labels
stud=[130,150,80,75]  # Assigning Data for pie
p.figure (figsize=[10,5])
p.title ('Class Strength')
p.pie (stud,labels=l,colors=['gold','red','pink','lightcoral'],autopct='%.2f%')
p.show ()
```

---

### Lets take another example to understand pie chart more clear

5. An analysis has been done in the school to identify hobby of Students as given below.

<table>
<thead>
<tr>
<th>Hobby</th>
<th>Music</th>
<th>Dance</th>
<th>Games</th>
<th>Reading</th>
<th>Drawing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Students</td>
<td>130</td>
<td>150</td>
<td>180</td>
<td>75</td>
<td>160</td>
</tr>
</tbody>
</table>

Represent this data on the Pie Chart. Slice colour [pink,green,blue,gold and light sky blue]

```python
import matplotlib.pyplot as p
L=['Drawing','Music','Dance','Games','Reading']   # Defining labels
nos=[130,150,80,75,160]  # Assigning Data for pie
c=['pink','green','blue','gold','lightskyblue']
eexplode=(0.1,0,0,0,0)
p.figure (figsize=[8,5])
p.title ('Hobbies Analysis')
p.pie (nos=explode,labels=L,colors=c,autopct='%2f% ',shadow=True,startangle=170)
p.show ()
```

---

- The `startangle` parameter rotates the pie chart by the specified number of degrees. The rotation is counter clock wise and performed on X Axis of the pie chart.
- Shadow effect can be provided using the `shadow` parameter of the `pie()` function. Passing `True` will make a shadow appear below the rim of the pie chart. By default value of `shadow` is `False` and there will be no shadow of the pie chart.
After applying various (explode, colors, shadow etc) parameters with pie function (EXAMPLE 5) out will look like this.

MULTIPLE PLOTS

To draw Multiple lines in one chart we can simply call plot() function multiple times.
MULTIPLE VIEWS

We can plot Multiple Graphs in the same window using subplot() function

```python
import matplotlib.pyplot as p

p.subplot ( 2,2,1)
p.plot ( Yr,rate,color="red")
p.title ( 'Multiple View Example', loc='center')
p.xlabel ( 'Year')

rate1=[80,90,97,98]
p.subplot ( 2,2,2)
p.plot ( Yr1,rate1,color="green")
p.title ( 'Multiple View Example', loc='center')
p.xlabel ( 'Year')

rate2=[10,15,20,25]
p.subplot ( 2,2,3)
p.plot ( Yr2,rate2,color="blue")
p.title ( 'Multiple View Example', loc='center')
p.xlabel ( 'Year')

p.show ()
```

OUTPUT:

We can also plot different types of graph AS BELOW