

3.6 Creating Histograms with PyPlot

A histogram is a summarisation tool for discrete or continuous data. A histogram provides a visual interpretation of numerical data by showing the number of data points that fall within a specified range of values (called *bins*). It is similar to a vertical bar graph. However, a histogram, unlike a vertical bar graph, shows no gaps between the bars (see Fig. 3.3)

Histograms are a great way to show results of *continuous data*, such as : weight, height, how much time, and so forth. But when the data is in categories (such as *Country* or *Subject* etc.), one should use a bar chart.

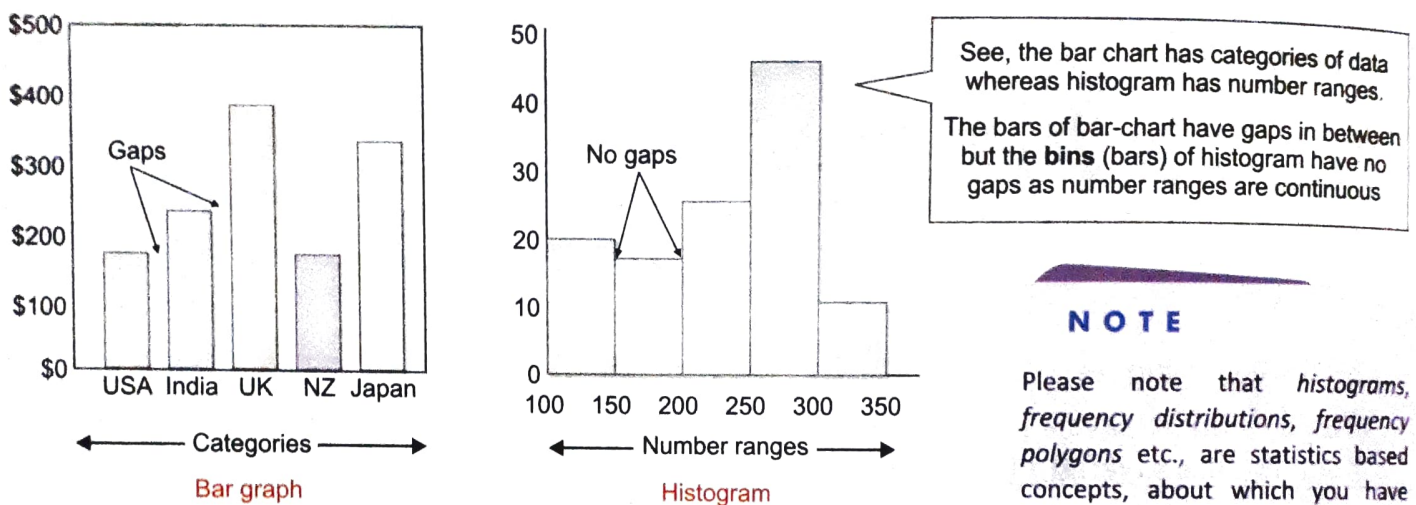


Figure 3.3 Bar graph vs histograms.

Let us now talk about how you can create histograms in Python.

Histogram using `hist()` Function

The `hist()` of PyPlot module lets you create and plot histogram from a give sequence(s) of numbers. The syntax for using `hist()` function of pyplot is :

```
matplotlib.pyplot.hist(x, bins = None, cumulative = False, histtype = 'bar',
                        align = 'mid', orientation = 'vertical', )
```

Parameters :

- x** (n,) array or sequence of (n,) arrays to be plotted on histogram.
- bins** int³, optional. If an integer is given, bins + 1 bin-edges are calculated and returned. Default value is automatically provided internally⁴.
- cumulative** bool, optional; If *True*, then a histogram is computed where each bin gives the counts in that bin plus all bins for smaller values. The last bin gives the total number of datapoints. Default is *False*.
- histtype** ('bar', 'barstacked', 'step', 'stepfilled'), optional ; the type of histogram to draw. 'bar' is a traditional bar-type histogram. If multiple data are given, the bars are arranged side by side.

3. bins can take a sequence too. For complete syntax and working refer to pyplot documentation.

4. taken from `rcParams` dictionary of `matplotlib` that contains default values for various parameters.

NOTE

Please note that *histograms, frequency distributions, frequency polygons* etc., are statistics based concepts, about which you have read in your statistics component of Economics. Therefore, we are not discussing the statistical explanations of these concepts here, instead we are just focussing on how these can be plotted using PyPlot.

'**barstacked**' is a bar-type histogram where multiple data are stacked on top of each other.

'**step**' generates a *lineplot* that is by default unfilled.

'**stepfilled**' generates a *lineplot* that is by default filled.

Default is 'bar' type of histtype

orientation {'horizontal', 'vertical'}, optional ; If 'horizontal', barh will be used for bar-type histograms.

Have a look at following examples that plot histograms from two *ndarrays* *x* and *y* each having randomly generated numbers (more than 100 number ; both arrays shown below).

```
x = array([-0.04773042, -0.54508323, 0.85572137, ..., 0.44027371, -0.26309049, -0.87732363])
y = array([-0.02914181, -0.33280057, 0.52246068, ..., 0.26880911, -0.1606299, -0.53564994])
```

Before you plot, make sure to import the required library/modules, *e.g.*, we have given following import statements in the beginning :

```
import matplotlib.pyplot as pl
import numpy as np
```

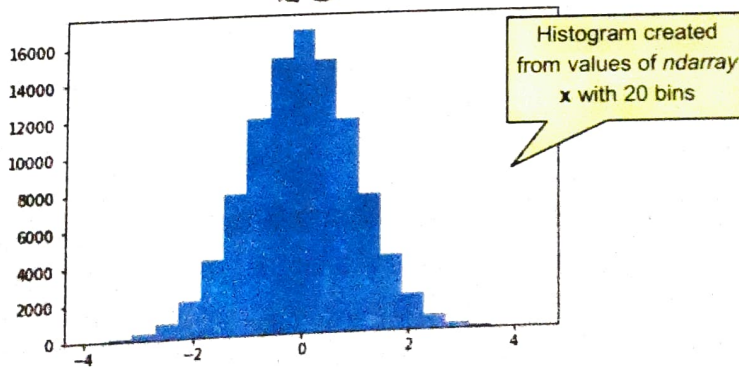
NOTE

The **pyplot** is a module in the **matplotlib** data visualization package of Python. That is why you always import it as **matplotlib.pyplot**.

1. Plot a histogram from an ndarray *x* with 20 bins

To specify bins, the **bins** argument of **hist()** is used.

```
pl.hist(x, bins = 20)
```



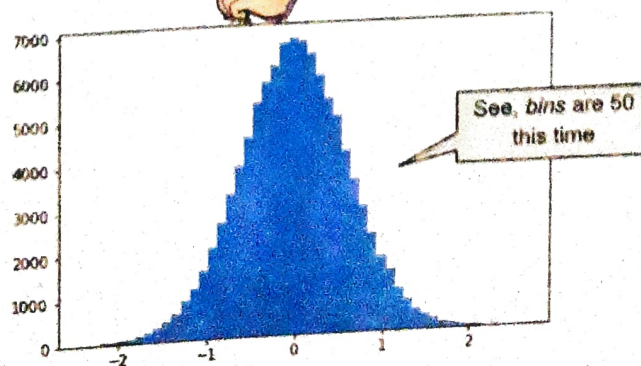
To see
Histogram plot
in action



Scan
QR Code

2. Plot a histogram from an ndarray *y* with 50 bins

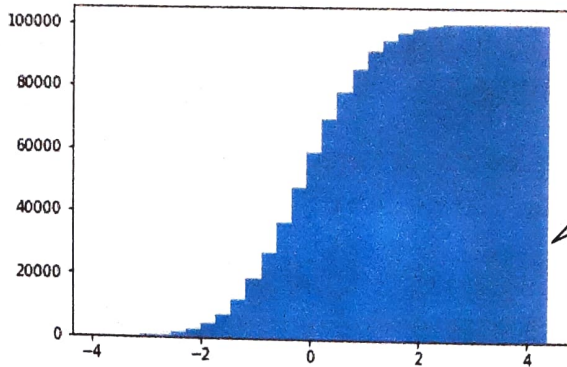
```
pl.hist(y, bins = 50)
```



3. Plot a cumulative histogram of ndarray x with 30 bins

For cumulative histogram, argument **cumulative** of `hist()` function is used.

```
pl.hist(x, bins = 30, cumulative = True)
```

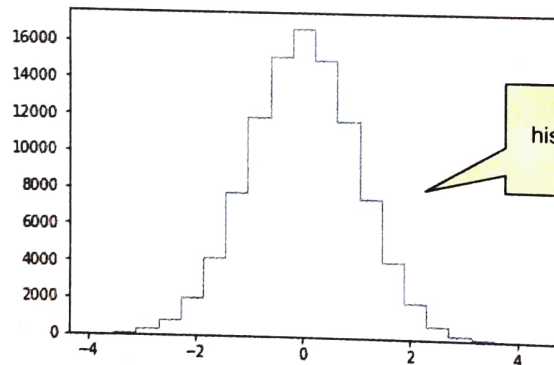


In cumulative histogram, counts for a bin are computed as : counts in that bin + all bins for smaller values

4. Plot ndarray x 's histogram as 'step' type histogram with 20 bins

To specify histogram type, argument **histtype** is to be used

```
pl.hist(x, bins=20, histtype = 'step')
```

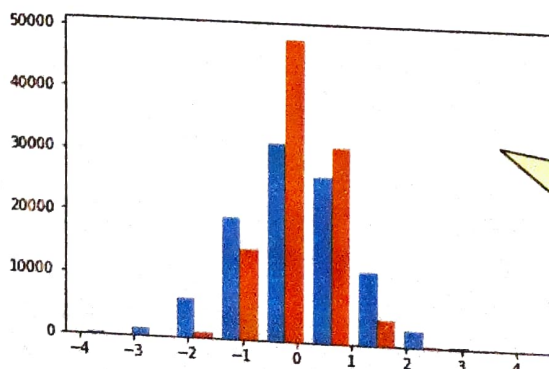


Step histtype is a histogram plotted as an unfilled *lineplot*

5. Plot both ndarray x and y in same histogram with

```
pl.hist([x, y])
```

To plot multiple arrays, provide them in form a sequence e.g., we have given arrays x and y in a list as $[x, y]$



Since we have given no bins argument, the number of bins are automatically decided through matplotlib default values dictionary

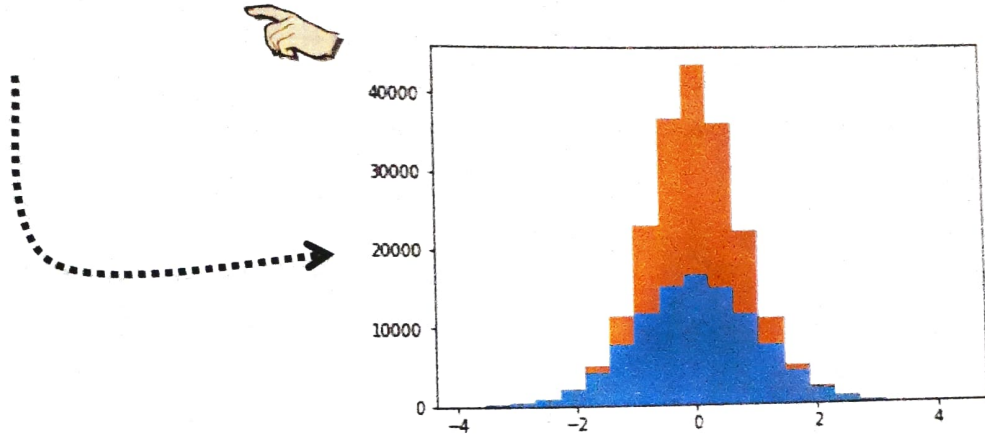
NOTE

A histogram is a summarisation tool for discrete or continuous data. A histogram provides a visual interpretation of numerical data by showing the number of data points that fall within a specified range of values (called **bins**).

6. Plot a *stacked bar* type histogram from both ndarray x and y

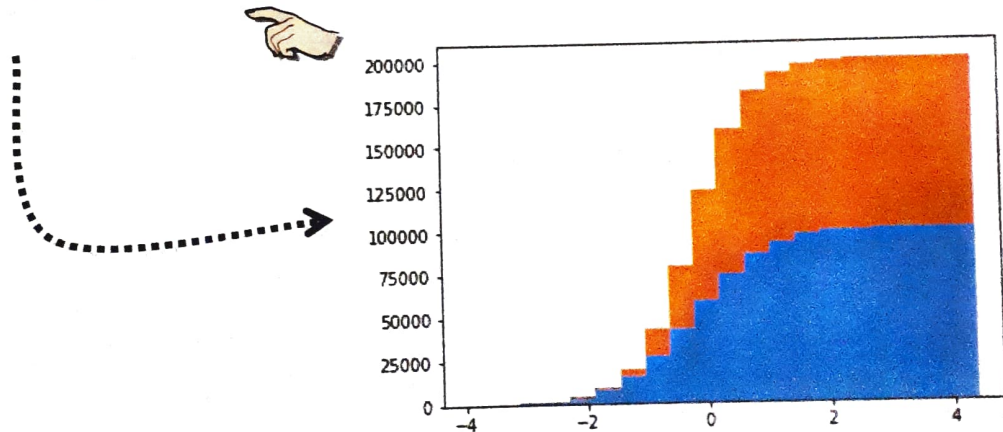
(a) *regular histogram*

```
pl.hist([x,y], histtype = 'barstacked')
```



(b) *cumulative histogram*

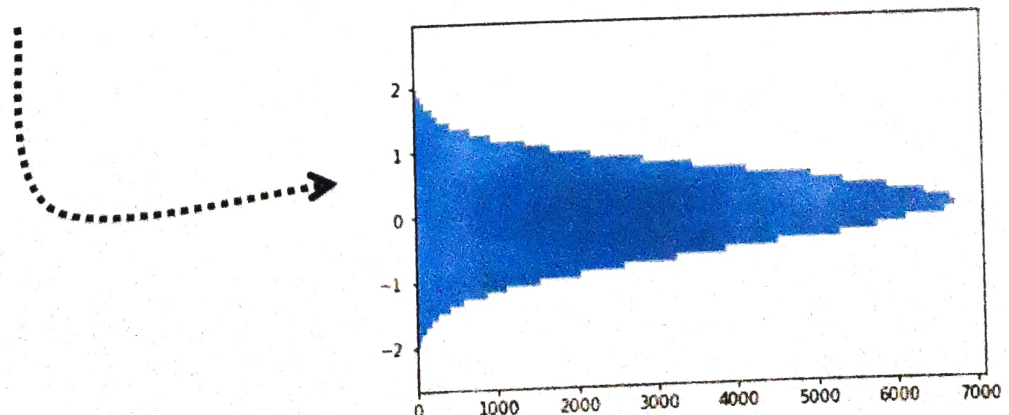
```
pl.hist([x,y], histtype = 'barstacked', cumulative = True)
```



7. Plot a *horizontal histogram* from ndarray y with 50 bins

To change the orientation of the histogram, we can use **orientation** argument.

```
pl.hist(y, bins = 50, orientation = 'horizontal')
```



EXAMPLE 27 A survey gathers height and weight of 100 participants and recorded the participants' ages as :

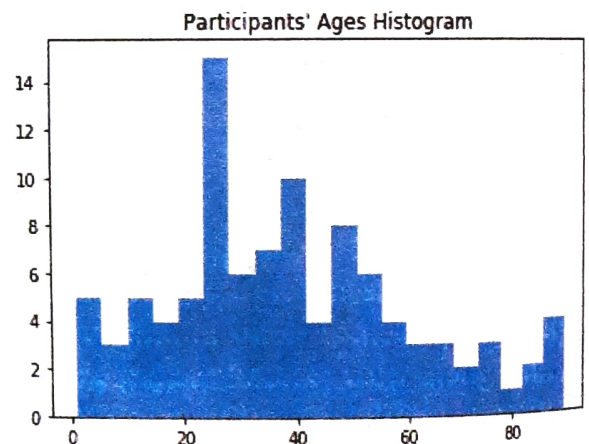
```
ages = [ 1, 1,2,3,5,7,8,9,10, 10,11,13,13,15,16,17,18, 19,20, 21,
        21,23,24,24,24, 25,25,25,25,26,26,26,27,27,27,27,27, 29,30,
        30,30,30,31,33,34,34,34,35,36,36,37,37,37,38,38,39, 40,40, 41,
        41,42, 43,45,45,46,46, 46, 47,48,48,49,50,51,51, 52, 52, 53, 54,
        55,56,57,58,60, 61,63,65,66,68,70,72,74, 75,77,81,83,84,87,89,90,91]
```

Write a program to plot a histogram from above data with 20 bins.

SOLUTION

```
import matplotlib.pyplot as plt.
ages = [ 1, 1,2,3,5,7,8,9,10, 10,11,13,13,15,16,17,18, 19,20, 21,
        21,23,24,24,24, 25,25,25,25,26,26,26,27,27,27,27,27, 29,30,
        30,30,30,31,33,34,34,34,35,36,36,37,37,37,38,38,39, 40,40,41,
        41,42, 43,45,45,46,46, 46, 47,48,48,49,50,51,51, 52, 52, 53, 54,
        55,56,57,58,60, 61,63,65,66,68,70,72,74, 75,77,81,83,84,87,89,90,91]
plt.hist(ages, bins = 20)
plt.title ("Participants' Ages Histogram")
plt.show()
```

Output



EXAMPLE 28 Prof Awasthi is doing some research in the field of Environment. For some plotting purposes, he has generated some data as :

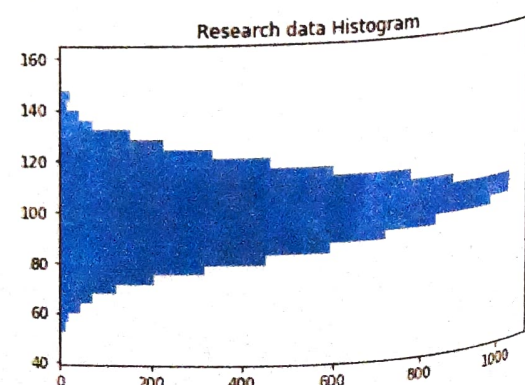
```
mu = 100
sigma = 15
x = mu + sigma * numpy.random.randn(10000)
```

Write a program to plot this data on a horizontal histogram with this data.

SOLUTION

```
import numpy as np
import matplotlib.pyplot as plt
mu = 100
sigma = 15
x = mu + sigma * np.random.randn(10000)
plt.hist(x, bins = 30, orientation = 'horizontal')
plt.title('Research data Histogram')
plt.show()
```

Output



EXAMPLE 29 Prof Awasthi is doing some research in the field of Environment. For some plotting purposes, he has generated some data as:

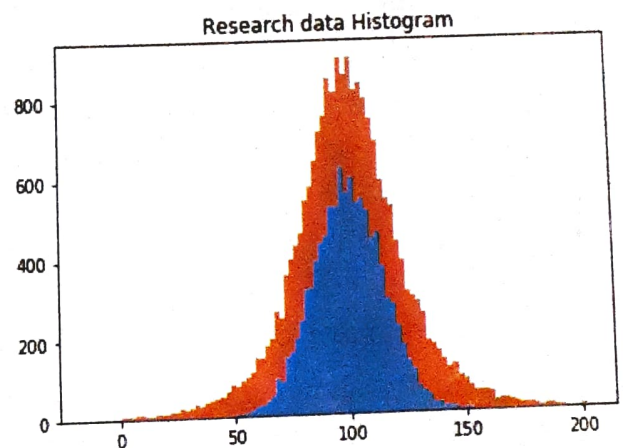
```
mu = 100
sigma = 15
x = mu + sigma * numpy.random.randn(10000)
y = mu + 30 * np.random.randn(10000)
```

Write a program to plot this data on a bar-stacked horizontal histogram with both x and y axes.

SOLUTION

```
import numpy as np
import matplotlib.pyplot as plt
mu = 100
sigma = 15
x = mu + sigma * np.random.randn(10000)
y = mu + 30 * np.random.randn(10000)
plt.hist([x,y], bins = 100, histtype = 'barstacked')
plt.title('Research data Histogram')
plt.show()
```

Output



3.7 Creating Frequency Polygons

A frequency polygon is a type of frequency distribution graph. In a frequency polygon, the number of observations is marked with a single point at the midpoint of an interval. A straight line then connects each set of points. Frequency polygons make it easy to compare two or more distributions on the same set of axes⁵.

Let's look at an example of a frequency polygon.

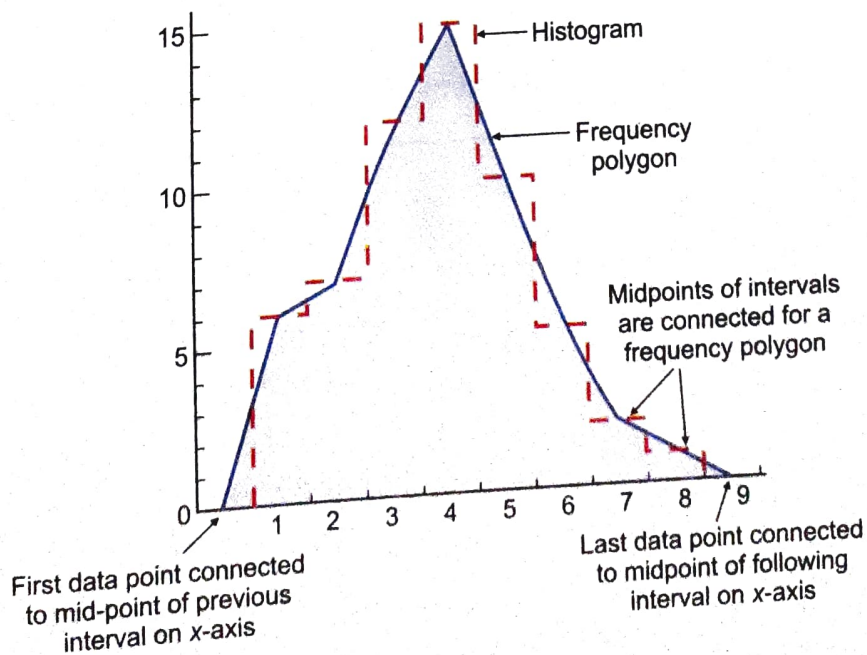


Figure 3.4 Frequency polygon

3.9 Plotting Data from a DataFrame

Till now you have plotted data from either linear lists, or 1D arrays. You can also plot data from a DataFrame using its columns selectively.

You can do it in *two* ways :

- (i) Using PyPlot's graph functions
- (ii) Using DataFrame's `plot()` function. It is available from version 0.17.0 onwards.

3.9.1 Plotting a DataFrame's Data using PyPlot's Graph Functions

To plot a DataFrame's data, just pass its column name to the Pyplot's graph functions (`plot()`, `bar()`, `barh()`, `scatter()`, `boxplot()`, `hist()`). It will treat the passed column's data as a Series and plot it, e.g., if we have a DataFrame `df2` as shown below :

DataFrame df2

	Age	Projects
0	30	13
1	27	17
2	32	16
3	40	20
4	28	21
5	32	14

DataFrame df

	Name	Sex	Position	City	Age	Projects	Budget
0	Rabia	F	Manager	Bangalore	30	13	8
1	Evan	M	Programmer	New Delhi	27	17	13
2	Jia	F	Manager	Chennai	32	16	32
3	Lalit	M	Manager	Mumbai	40	20	21
4	Jaspreet	M	Programmer	Chennai	28	21	17
5	Suji	F	Programmer	Bangalore	32	14	10

DataFrame fdf

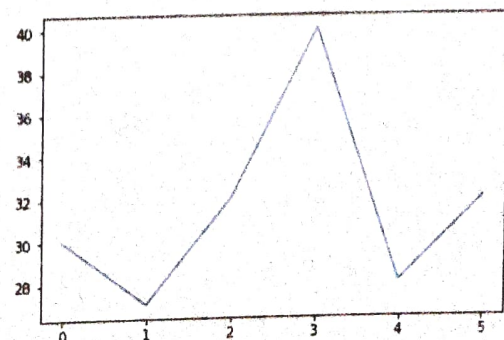
	Day1	Day2	Day3	Day4	Day5
0	74.25	56.03	59.30	69.00	89.65
1	76.06	68.71	72.07	78.47	79.65
2	69.50	62.89	77.65	65.53	80.75
3	72.55	56.42	66.46	76.85	85.08

Reference 3.2

And if you write the code as :

```
import pandas as pd
import matplotlib.pyplot as plt
: #df2 created or loaded
plt.plot(df2.Age)
```

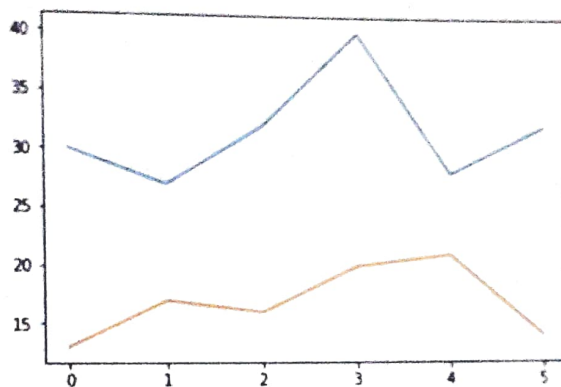
It will give you a plot as shown here.



To avoid a situation like above (plotting of non-numeric columns), you can explicitly specify the column names to be plotted, e.g., to plot only the *Age* and *Projects* columns against the index values of the above DataFrame *df*, you may write :

```
import pandas as pd
import matplotlib.pyplot as plt
: #df2 created or loaded
plt.plot(df.index, df.Age, df.Projects)
```

It will give you a plot as shown below, where it has plotted only *Age* and *Projects* columns against *index*.



3.9.2 Plotting a DataFrame's Data using DataFrame's plot()

Pandas provides a function *plot()* which you can use with DataFrame as :

```
<DF>.plot()
```

And which will plot from the data of the DataFrame automatically. The DataFrame's *plot()* is a versatile function, which can plot all types of chart by just specifying *kind* argument.

Various arguments that *<DF>.plot()* can take are⁷ :

dataSeries or *DataFrame* : The object for which the method is called.

- kind** : type of the plot, can take values as
 - 'line' : line plot (default)
 - 'bar' : vertical bar plot
 - 'barh' : horizontal bar plot
 - 'hist' : histogram
 - 'box' : boxplot
 - 'pie' : pie plot
 - 'scatter' : scatter plot
- And some other plot types

NOTES

- ❖ With scatter, you have to explicitly specify the x and y arguments.
- ❖ If you do not give kind argument at all, line chart will be plotted.

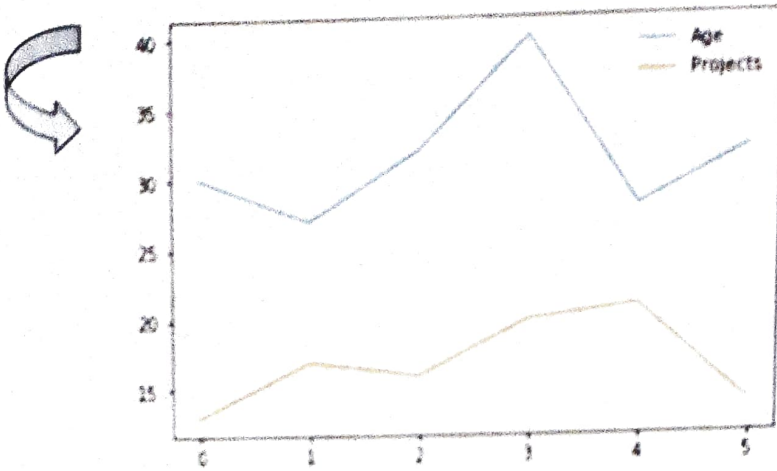
Before we show you the use of *<DF>.plot()*, let us talk about the advantages of it :

- ❖ It plots only the numeric columns unlike *plot()* of PyPlot when used with a DataFrame.
- ❖ It automatically adds legends for the plotted data.

7. Please note, we are not giving full syntax of this function here as it is beyond the scope of this book.

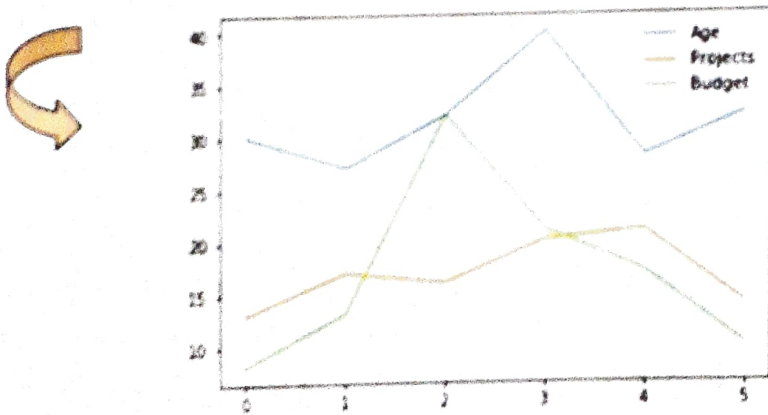
Consider the `df` and `df2` DataFrame you have used above and plot these using pandas `plot()`:

```
import pandas as pd
import matplotlib.pyplot as plt
: #df2 created or loaded
df2.plot() # df2 contains numeric columns only
```



See, `<DF>.plot()` has added legends automatically. The `kind` argument is missing, hence line chart is plotted.

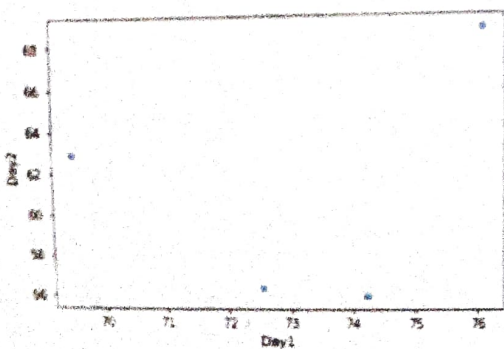
```
df.plot() # df contains numeric as well as non-numeric columns
# above code line is equivalent to df.plot(kind = 'line')
```



See, `<DF>.plot()` has:
 (i) Plotted only the numeric columns of DataFrame `df`
 (ii) added legends automatically.

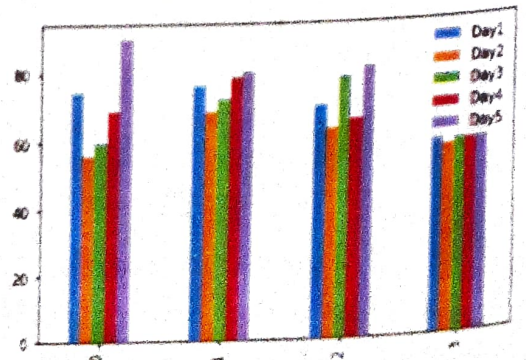
Just play with the `kind` argument and create different chart types.

◆ **Creating scatter charts with `DF.plot()`** (see left figure below)
`fdf.plot(x = 'Day1', y = 'Day2', kind = 'scatter')`

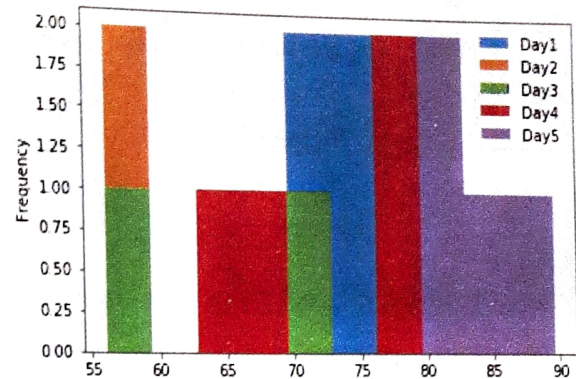
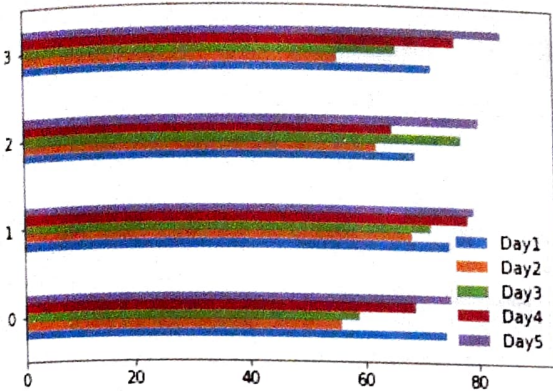


◆ **Creating bar charts with `DF.plot()`**
`fdf.plot(kind = 'bar')`

(see right figure above)

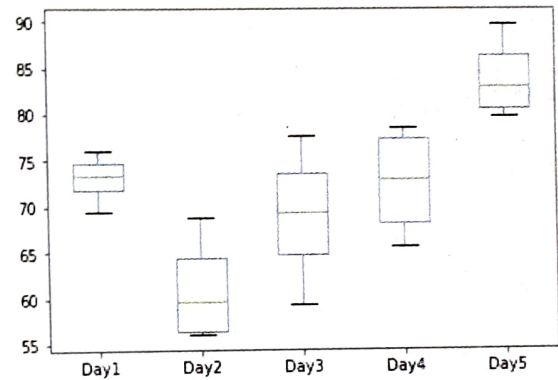


⇒ Creating horizontal bar charts with DF.plot() (see left figure below)
`fdf.plot(kind = 'barh')`



⇒ Creating histogram with DF.plot() (see right figure above)
`fdf.plot(kind = 'hist')`

⇒ Creating boxplots with DF.plot()
`fdf.plot(kind = 'box')`



EXAMPLE 35 Consider the following DataFrame *prodf*:

	Fruits	Pulses	Rice	Wheat
Andhra P.	7830.0	931.0	7452.4	NaN
Gujrat	11950.0	818.0	1930.0	2737.0
Kerala	113.1	1.7	2604.8	NaN
Punjab	7152.0	33.0	11586.2	16440.5
Tripura	44.1	23.2	814.6	0.5
Uttar P.	24169.2	2184.4	13754.0	30056.0

Write a program to plot a scatter chart with the columns *Pulses*.

SOLUTION

Since the given DataFrame *prodf* does not have numeric *index*, we cannot use it for plotting because *x* has the numeric for `scatter()`.

We can create numeric values for *x*-axis as :

```
x = range(0, len(prodf)) # will generate [0, 1, 2, 3, 4, 5]
```

OR

```
x = range(1, len(prodf) + 1) # will generate [1, 2, 3, 4, 5, 6]
```

- ❖ You can create scatter charts using either `plot()` function or `scatter()` function.
- ❖ A Bar Graph/Chart is a graphical display of data using bars of different heights.
- ❖ You can create bar chart using pyplot's `bar()` function.
- ❖ You can change colors of the bars, widths of the bars in `bar()` function.
- ❖ Use `barh()` function to create horizontal bar chart.
- ❖ The `pie()` creates a pie chart.
- ❖ The plot area is known as figure and every other element of chart is contained in it.
- ❖ The axes can be labelled using `xlabel()` and `ylabel()` functions.
- ❖ The limits of axes can be defined using `xlim()` and `ylim()` functions.
- ❖ The tick marks for axes values can be defined using `xticks()` and `yticks()` functions.
- ❖ The `title()` function adds title to the plot.
- ❖ Using `legend()` function, one can add legends to a plot where multiple data ranges have been plotted, but before that the data ranges must have their **label** argument defined in `plot()` or `bar()` function.
- ❖ The **loc** argument of `legend()` provides the location for legend, which by default is 1 or "upper right".
- ❖ A histogram is a summarisation tool for discrete or continuous data.
- ❖ A histogram provides a visual interpretation of numerical data by showing the number of data points that fall within a specified range of values (called bins).
- ❖ Pyplot module's `hist()` lets you create histograms.
- ❖ A frequency polygon is a type of frequency distribution graph.
- ❖ In a frequency polygon, the number of observations is marked with a single point at the midpoint of an interval.
- ❖ The box plot is used to show the range and the middle half of the ranked data.
- ❖ The `boxplot()` of pyplot lets you draw boxplots.
- ❖ A dataframe's data can be plotted in various chart types using `<DF>.plot(kind = <type>)` function.

Objective Type Questions

O T Q s

Multiple Choice Questions

1. PyPlot is an interface of Python's _____ library.
 (a) seaborn (b) plotly (c) ggplot (d) matplotlib
2. For 2D plotting using a Python library, which library interface is often used ?
 (a) seaborn (b) plotly (c) matplotlib (d) matplotlib.pyplot
3. Which of the following is not a valid chart type ?
 (a) histogram (b) statistical (c) pie (d) box
4. Which of the following is not a valid plotting function of pyplot ?
 (a) `plot()` (b) `bar()` (c) `line()` (d) `pie()`
5. Which of the following plotting functions does not plot multiple data series ?
 (a) `plot()` (b) `bar()` (c) `pie()` (d) `barh()`
6. The plot which tells the trend between two graphed variables is the _____ graph/chart.
 (a) line (b) scatter (c) bar (d) pie

7. _____ graph/chart.
- (a) line (b) scatter (c) bar (d) pie
8. A _____ is a summarisation tool for discrete or continuous data.
- (a) quartile (b) histogram (c) mean (d) median
9. A visual representation of the statistical five number summary of a given dataset is known as _____.
- (a) histogram (b) frequency distribution
(c) boxplot (d) frequency polygon
10. Which of the following functions is used to create a line chart ?
- (a) line() (b) plot() (c) chart() (d) plotline()
11. Which of the following function will produce a bar chart ?
- (a) plot() (b) bar() (c) plotbar() (d) barh()
12. Which of the following function will create a vertical bar chart ?
- (a) plot() (b) bar() (c) plotbar() (d) barh()
13. Which of the following function will create a horizontal bar chart ?
- (a) plot() (b) bar() (c) plotbar() (d) barh()
14. To specify the style of line as dashed, which argument of plot() needs to be set ?
- (a) line (b) width (c) style (d) linestyle
15. The datapoints plotted on a graph are called _____.
- (a) points (b) pointers (c) marks (d) markers
16. A _____ graph is a type of chart which displays information as a series of data points connected by straight line segments.
- (a) line (b) bar (c) pie (d) boxplot
17. To create scatter charts using plot(), which argument is skipped ?
- (a) marker (b) linestyle (c) markeredgcolor (d) linewidth
18. In scatter(), which argument is used to specify the size of datapoints ?
- (a) size (b) s (c) marker (d) markersize
19. Which argument of bar() lets you set the thickness of bar ?
- (a) thick (b) thickness (c) width (d) barwidth
20. Which function lets you set the title of the plot ?
- (a) title() (b) plt.title() (c) graphtitle() (d) all of these
21. The command used to give a heading to a graph is _____.
- (a) plt.show() (b) plt.plot() (c) plt.xlabel() (d) plt.title()
22. Which function would you use to set the limits for x-axis of the plot ?
- (a) limits() (b) xlims() (c) xlim() (d) lim()
23. Which function is used to show legends ?
- (a) display() (b) show() (c) legend() (d) legends()
24. Which argument must be set with plotting functions for legend() to display the legends ?
- (a) data (b) label (c) name (d) sequence

25. Which function is used to create a histogram ?
 (a) histo() (b) histogram() (c) hist() (d) histtype
26. Which argument in hist() is used to create a stacked bar type histogram ?
 (a) histt (b) histtype (c) type (d) barstacked
27. Which of the following functions can plot only one data series ?
 (a) plot() (b) bar() (c) boxplot() (d) pie()
28. Which argument must be provided to create wedges out of a pie chart ?
 (a) label (b) autopct (c) explode (d) wedge
29. Which argument should be set to display percentage share of each pie on a pie chart ?
 (a) label (b) autopct (c) explode (d) wedge
30. Which function creates a boxplot ?
 (a) box() (b) plot() (c) boxplot() (d) showbox()
31. Which argument of boxplot() is used to create a filled boxplot ?
 (a) fill (b) box (c) patch_artist (d) patch

Fill in the Blanks

1. A _____ is a plot that shows the underlying frequency distribution of a set of continuous data.
2. Pyplot interface is a collection of methods within _____ library of Python.
3. Pyplot's _____ function is used to create line charts.
4. Pyplot's _____ function is used to create horizontal bar charts.
5. Pyplot's _____ function is used to create scatter charts.
6. Pyplot's _____ function is used to create histogram.
7. The datapoints plotted on a graph are called _____ .
8. The _____ argument of plot() specifies the width for the line.
9. The _____ argument of plot() specifies the style of the line.
10. The _____ argument of bar() specifies the bar width.
11. The _____ function is used to specify ticks for x-axis.
12. To save a plot, _____ function is used.
13. The _____ argument of hist() is set to create a horizontal histogram.
14. The _____ argument shows the arithmetic mean on a boxplot.
15. The _____ argument in a boxplot() creates a notched boxplot.
16. The _____ argument of legend() provides the location of legend.
17. Using Python Matplotlib _____ can be used to count how many values fall into each interval.
 (line plot/bar graph/histogram)

[CBSE Sample Paper 2020-21]

True/False Questions

1. PyPlot is a sub-library of matplotlib library.
2. Statement `import pyplot.matplotlib` is a valid statement for working on pyplot functions.
3. By default, pie chart is printed in elliptical or oval shape.
4. The default shape of pie chart cannot be changed from oval.

5. A line chart can be plotted using pyplot library's `line()` function.
6. A line chart can be plotted using pyplot library's `plot()` function.
7. A bar chart can be plotted using pyplot library's `bar()` function.
8. A bar chart can be plotted using pyplot library's `barh()` function.
9. It is not possible to plot multiple series of values in the same bar graph.
10. A standard marker of representing a non-number data in Python libraries is NaN.
11. If the `linestyle` argument is missing along with `markerstyle-string` in a `plot()`, a scatter type chart get created.
12. The `bar()` function can also create horizontal bar charts.
13. The `pie()` function can plot multiple data series.
14. The plot is always as per the data series being plotted irrespective of the `xlim()`.
15. Frequency polygon is created from histogram.

NOTE : Answers for OTQs are given at the end of the book.

Solved Problems

1. *What is data visualization ? What is its significance ?*

Solution. Data visualization is a general term that describes any effort to help people understand the significance of data by placing it in a visual context. In simple words, Data visualization is the process of displaying data/information in graphical charts, figures and bars.

Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier with data visualization techniques or tools such as line chart, bar chart, pie chart, histogram, scatter chart etc. Thus with data visualization tools, information can be processed in efficient manner and hence better decisions can be made.

2. *What is Python's support for Data visualization ?*

Solution. Python supports data visualizations by providing some useful libraries for visualization. Most commonly used data visualization library is **matplotlib**.

Matplotlib is a Python library, also sometimes known as the plotting library. The matplotlib library offers very extensive range of 2D plot types and output formats. It offers complete 2D support along with limited 3D graphic support. It is useful in producing publication quality figures in interactive environment across platforms. It can also be used for animations as well.

There are many other libraries of Python that can be used for data visualization but matplotlib is very popular for 2D plotting.

3. *What is pyplot ? Is it a Python library ?*

Solution. The *pyplot* is one of the interfaces of *matplotlib* library of Python. This interface offers simple MATLAB style functions that can be used for plotting various types of charts using underlying *matplotlib* library's functionality.

Pyplot is an interface, i.e., a collection of methods for accessing and using underlying functionality of a library, not a library. The *matplotlib* library has many other interfaces too, along with *pyplot* interface.

4. Name some commonly used chart types.

Solution. Some commonly used chart types are : Line chart, Bar chart, Pie chart, Scatter chart etc.

5. Name major components of a chart.

Solution. Major components of a chart are : Figure, Axes, Axis label, Limits, Title, Legends etc.

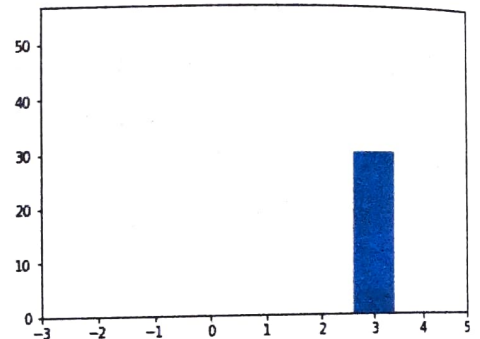
6. Name the functions you will use to create a (i) line chart, (ii) bar chart, (iii) scatter chart.

Solution.

- (i) `matplotlib.pyplot.plot()` (ii) `matplotlib.pyplot.bar()`
 (iii) `matplotlib.pyplot.plot()` and `matplotlib.pyplot.scatter()`

7. Consider the code given below (all required libraries are imported) and the output produced by it. Why is the chart showing one bar only while we are plotting four values on the chart ?

```
:
a = [3, 6, 9, 12]
b = [30, 48, 54, 48]
plt.xlim(-3, 5)
plt.bar(a,b)
plt.show()
```



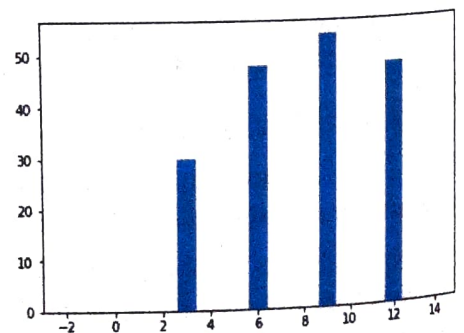
Solution. The given chart is showing a single bar as the limits of x-axis have been set as -3 to 5. On this range, only one value from the data range being plotted falls i.e., only `a[0]` and `b[0]` fall on this range. Thus only a single value `b[0]` i.e., 30 is plotted against `a[0]` i.e., 3.

8. What changes will you make to the code of previous question so that the bars are visible for all four points ? But do keep in mind that the x-axis must begin from the point -3.

Solution.

If we change the limits of x-axis so that all the points being plotted fall in the range of limits, all values will show. Thus, we have changed the limits from -3 to 15, in place of -3 to 5.

```
plt.xlim(-3, 15)
plt.bar(a,b)
plt.show()
```



9. Fill in the blank with the correct statement to plot a bar graph using a matplotlib method, so that Company ABC can see the graphical presentation of its Profit figures for the 2nd quarter of the financial year 2019 (i.e., August, September, October, November).

```
import matplotlib.pyplot as mtp
Months = ['AUG', 'SEP', 'OCT', 'NOV']                      #X Axis
Profits = [125, 220, 230, 175]                              #Y Axis
```

`mtp.show()`

Solution.

```
mtp.bar(Months, Profits)
```


10. Why is following code not producing any result ? Why is it giving errors ?
(Note. All required libraries have been imported and are available)

```
a = range(10, 50, 12)
b = range(90, 200, 20)
matplotlib.pyplot.plot(a, b)
```

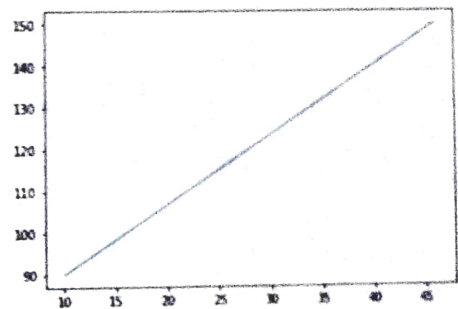
Solution. The above code is producing error because the two sequences being plotted i.e., **a** and **b** do not match in shape. While sequence '**a**' contains 4 elements, sequence '**b**' contains 6 elements. For plotting, it is necessary that the two sequences being plotted must match in their shape.

11. What changes will you recommend to rectify the error in previous question's code ?

Solution. Since both the sequences being plotted must match in their shape, we can achieve this either by adding two elements to sequence **a** so that it has the same shape as sequence **b** (i.e., 6 elements) or by removing two elements from sequence **b** so that it matches the shape of sequence **a** (i.e., 4 elements).

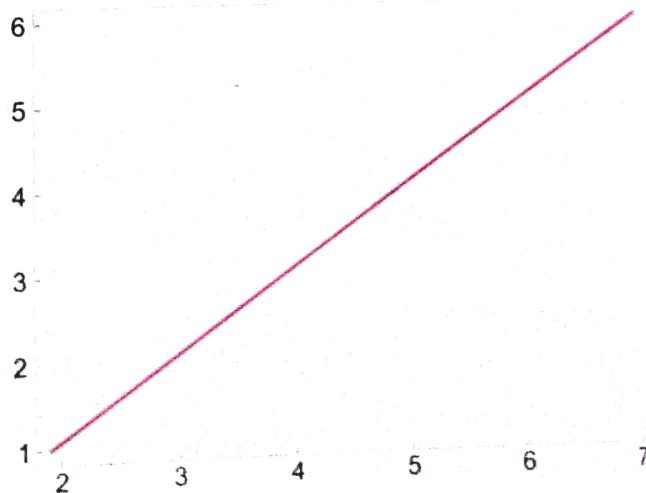
For instance,

```
a = range(10, 50, 12)
b = range(90, 160, 20)
matplotlib.pyplot.plot(a, b)
```



12. Consider the following graph. Write the code to plot it.

[CBSE Sample Paper 2020-21]



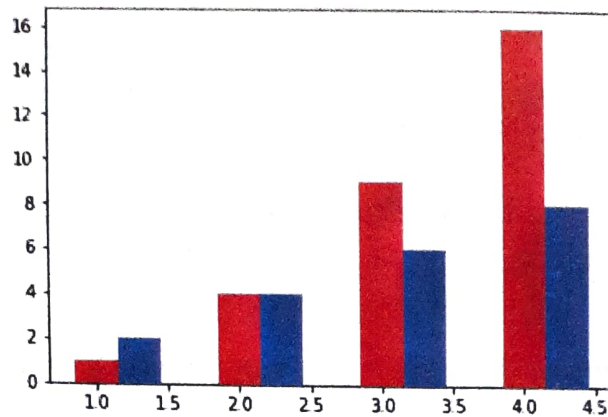
Solution.

```
import matplotlib.pyplot as plt
plt.plot([2,7],[1,6])
plt.show()
```

Alternative answer

```
import matplotlib.pyplot as plt
a = [1, 2, 3, 4, 5, 6]
b = [2, 3, 4, 5, 6, 7]
plt.plot(a,b)
```

13. Given an ndarray p as $([1, 2, 3, 4])$. Write code to plot a bar chart having bars for p and $p**2$ (with red color) and another bar for p vs $p*2$ (with blue color). (assume that libraries have been imported)

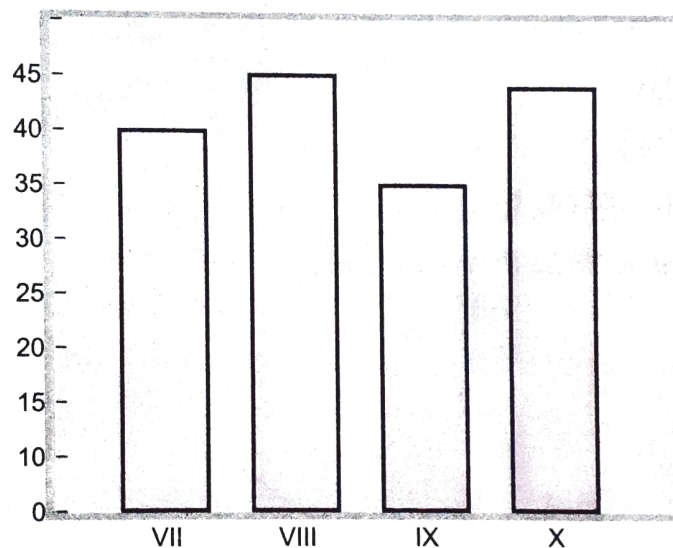


Solution.

```
plt.bar(p, p**2, color = 'r', width = 0.3)
plt.bar(p+0.3, p*2, color = 'b', width = 0.3)
```

14. Draw the following bar graph representing the number of students in each class.

[CBSE SP 2020-21]



Solution.

```
import matplotlib.pyplot as plt
Classes = ['VII', 'VIII', 'IX', 'X']
Students = [40, 45, 35, 44]
plt.bar(classes, students)
plt.show()
```

15. The table below shows the Marks of two students for the four unit tests for academic session 2019-2020. Fill in the blanks to draw a line graph with Test Names on the X-axis and Marks on the Y-axis. [CBSE D 2020C]

Tests	Marks	
	Rohit	Suman
Unit1	85	97
Unit2	88	99
Unit3	89	90
Unit4	87	92


```
import matplotlib.pyplot as plt
Tests = _____ #Assign Test Names
Rohit = _____ #Assign Marks of Rohit
Suman = _____ #Assign Marks of Suman
plt.plot(Tests, Rohit, Suman)
_____ #Label Y axis as Marks
_____ #Add legends "Rohit", "Suman" for the lines
plt.show()
```

Solution.

```
['Unit1', 'Unit2', 'Unit3', 'Unit4']
[85, 88, 89, 87]
[97, 99, 90, 92]
plt.ylabel('Marks')
plt.legend(['Rohit', 'Suman'])
```

16. What is scatter chart? How is it different from line chart ?

Solution. The scatter chart is a graph of plotted points that show the relationship between two sets of data. With a scatter plot, a mark or marker (usually a dot or small circle), represents a single data point. With one mark (point) for every data point a visual distribution of the data can be seen. Depending on how tightly the points cluster together, you may be able to discern a clear trend in the data.

The difference is that with a scatter plot, the decision is made from the data points such that the individual points should not be connected directly together with a line but, instead express a trend.

17. What is histogram ? How is it useful ?

Solution. A histogram is a statistical tool used to summarise discrete or continuous data. It provides a visual interpretation of numerical data by showing the number of data points that fall within a specified range of values (called "bins").

18. Following code is plotting the desired graph but legends are not showing despite giving the legend() of PyPlot. What could be the reason? Suggest a solution for the problem.

```
plt.plot(x, y)
plt.plot(x, z)
plt.legend(loc = "upper left")
```

Solution. The above code won't print the legends because with the plot(), the labels are missing. The legend() will work only when we specify label for data series being plotted in the plot().

The solution for above problem will be :

```
plt.plot(x, y, label = "Y data")
plt.plot(x, z, label = "Z data")
plt.legend(loc = "upper left")
```

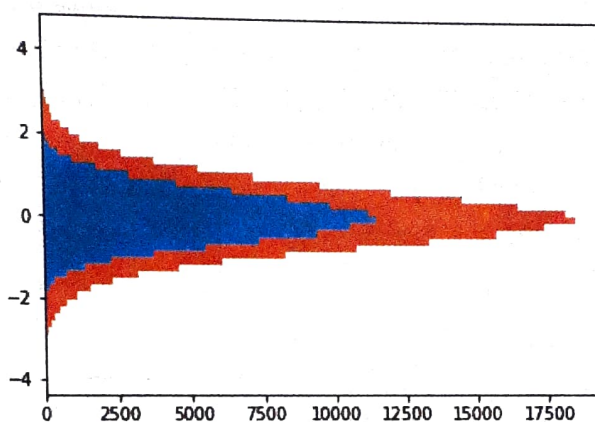
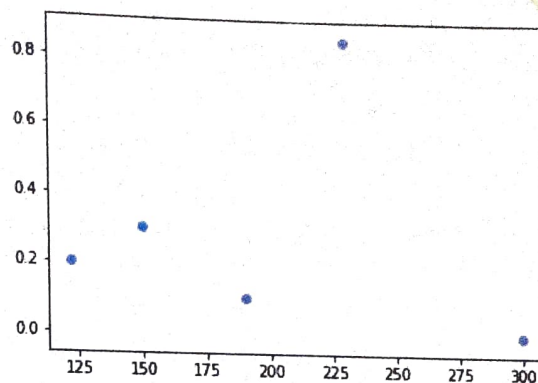
(b)

```
import matplotlib.pyplot as pl
: # df created
pl.scatter(df['meters'], df['weight'])
```

21. Write a program to create a histogram that plots two ndarrays x and y with 48 bins, in stacked horizontal histogram.

Solution.

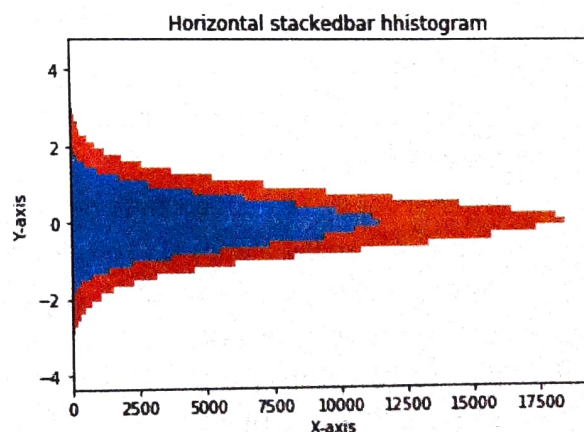
```
import matplotlib.pyplot as pl
: # df created
pl.scatter(df['meters'], df['weight'])
pl.hist([y,x], bins = 48, orientation = 'horizontal', histtype = 'barstacked')
```



22. Write code to add plot title and axes titles to above plot.

Solution.

```
import matplotlib.pyplot as pl
: # df created
pl.scatter(df['meters'], df['weight'])
pl.hist([y,x], bins = 48, orientation = 'horizontal', histtype = 'barstacked')
pl.title("horizontal stackedbar histogram")
pl.xlabel("X-axis")
pl.ylabel("Y-axis")
```

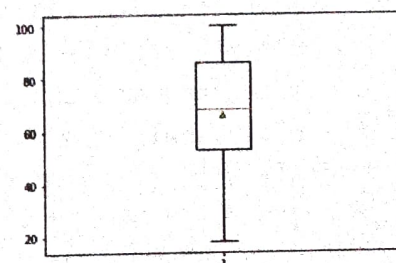


23. Write a program to create a boxplot from the following set of data :

34, 18, 100, 27, 54, 52, 93, 59, 61, 87, 68, 85, 78, 82, 91

Solution.

```
import matplotlib.pyplot as pl
A = [34, 18, 100, 27, 54, 52, 93, 59, 61, 87, 68, 85, 78, 82, 91]
pl.boxplot(A, showmeans = True)
```



Assignment

For
selective assignment
solutions



Scan
QR Code

Type A : Short Answer Questions/Conceptual Questions

1. Name the library of which the PyPlot is an interface.
2. Write the statement to import PyPlot in your script.
3. Name the functions to create the following :

(a) line chart	(b) bar chart	(c) horizontal bar chart	(d) histogram
(e) scatter chart	(f) boxplot	(g) pie chart	
4. What is a line chart ?
5. What is a scatter chart ?
6. What is a pie chart ?
7. What is a bar chart ?
8. What is a histogram ?
9. What is a boxplot ?
10. What is a frequency polygon ?
11. Name the function to label axes.
12. Name the function to give title to a plot.
13. Name the function to set figure size of a plot.
14. Name the function to set limits for the axes.
15. Name the function to show legends on a plot.
16. Name the function to add ticks on axes.

Type B : Application Based Questions

1. What is the significance of data visualization ?
2. How does Python support data visualization ?
3. What is the use of matplotlib and pyplot ?
4. What are the popular ways of plotting data ?
5. Compare bar() and barh() functions.
6. What is the role of legends in a graph/chart ?
7. What will happen if you use legend() without providing any label for the data series being plotted ?
8. What do you understand by xlim and ylim ? How are these linked to data being plotted ?
9. When should you use (i) a line chart, (ii) a bar chart, (iii) a scatter chart, (iv) pie chart, (v) boxplot ?
10. A list namely *temp* contains average temperatures for seven days of last week. You want to see how the temperature changed in last seven days. Which chart type will you plot for the same and why ?
11. What is histogram ? How do you create histograms in Python ?
12. What are various types of histograms that can be created through hist() function ?
13. When should you create histograms and when should you create bar charts to present data visually ?
14. What is cumulative histogram ? How do you create it using PyPlot ?
15. What is frequency polygon ? How do you create it ?
16. What is 5 point summary ?
17. What is Boxplot ? How do you create it in Pyplot ?

Type C : Practical/Knowledge Based Questions

1. Execute the following codes and find out what happens ? (Libraries have been imported already ; plt is the alias name for matplotlib.pyplot)

(a)

```
A = np.arange(2, 20, 2)
B = np.log(A)
plt.plot(A, B)
```

(b)

```
A = np.arange(2, 20, 2)
B = np.log(A)
plt.bar(A, B)
```

(c)

```
X = np.arange(1, 18, 2.655)
B = np.log(X)
plt.scatter(X, Y)
```

Will any code produce error ? Why/Why not ?

2. Write the output from the given python code :

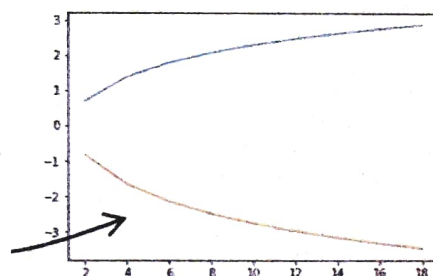
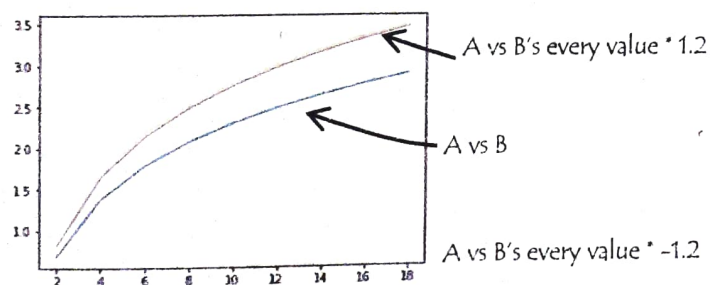
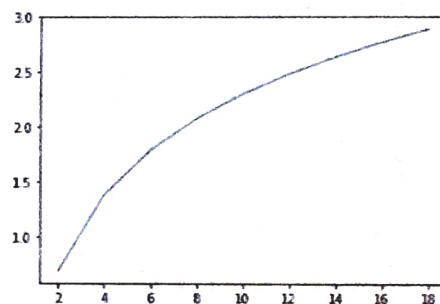
[CBSE D 2020]

```
import matplotlib.pyplot as plt
Months = ['Dec', 'Jan', 'Feb', 'Mar']
Attendance = [70, 90, 75, 95]
plt.bar(Months, Attendance)
plt.show()
```

3. Write a program to add titles for the X-axis, Y-axis and for the whole chart in above codes.

4. `plt.plot(A, B)` produces (A and B are the sequences same as created in question 3) chart as :

Write codes to produce charts as shown below :



5. Given a data frame `df1` as shown below :

	1990	2000	2010
a	52	340	890
b	64	480	560
c	78	688	1102
d	94	766	889

Write code to create :

- A scatter chart from the 1990 and 2010 columns of dataframe `df1`
 - A line chart from the 1990 and 2000 columns of dataframe `df1`
 - Create a bar chart plotting the three columns of dataframe `df1`
6. The score of four teams in 5 IPL matches is available to you. Write a program to plot these in a bar chart.

7. The score of a team in 5 IPL matches is available to you. Write a program to create a pie chart from this data, showing the last match's performance as a wedge.
8. The prices of a stock for 3 months are given. Write a program to show the variations in prices for each month by 3 lines on same line chart. Make sure to add legends and labels. Show grid also.
9. A distribution data stores about 1000 random number. Write a program to create a scatter chart from this data with varying point sizes.
10. Navya has started an online business. A list stores the number of orders in last 6 months. Write a program to plot this data on a horizontal bar chart.
11. Given the following set of data :

Weight measurements for 16 small orders of French-fries (in grams).

78 72 69 81 63 67 65 75
79 74 71 83 71 79 80 69

- (a) Create a simple histogram from the above data
- (b) Create a horizontal histogram from the above data
- (c) Create a step type of histogram from the above data
- (d) Create a cumulative histogram from the above data
12. Create an ndarray containing 16 values and then plot this array along with dataset of previous question in same histogram
 - (a) normal histograms
 - (b) cumulative histograms
 - (c) horizontal histograms
13. Out of above plotted histograms, which ones can be used for creating frequency polygons ? Can you draw frequency polygons from all the above histograms ?
14. Create/draw frequency polygon from the data used in above questions.
15. From the following ordered set of data :

63, 65, 67, 69, 69, 71, 71, 72, 74, 75, 78, 79, 79, 80, 81, 83

- (a) Create a horizontal boxplot
- (b) Create a vertical boxplot
- (c) Show means in the boxplot
- (d) Create boxplot without the box
16. Sina has created ordered set of data from the number of new customers registered on his online service centre in last 20 months.
Write a program to plot this data on a filled boxplot with means shown.