Mysql root

**MATPLOTLIB**

**What is?**

* 2D and 3D plotting python library
* Produce high quality graph
* Support animation

**Matplotlib Graphs**

* Histogram
* Bar chart
* Scatterplots
* Area plot
* Pie chart
* Error chart
* Power spectra

**Installation**

* Pip install Matplotlib

**Importing Matplotlib**

From Matplotlib import pyplot as plt

or

import Matplotlib.pyplot asplt

**LINE PLOT**

* To plot a line we use **plt**.**plot() method,** and give **x** and **y** value
* To show plot use **plt.show()** method
* **plt.title(“x”,”y”)** method is used to give title to plot
* **plt.xlabel(“x”)** is for labeling the line plot
* **plt.ylabel(“y”)** is for labeling the line plot
* **plt.axis([axis([xmin, xmax, ymin, ymax])** to give startint point to line plot
* **color = “r”** change the color of graph

The supported color abbreviations are the single letter codes

character color

``'b'`` blue

``'g'`` green

``'r'`` red

``'c'`` cyan

``'m'`` magenta

``'y'`` yellow

``'k'`` black

``'w'`` white

* **marker = “o”** toshow graph point

\*\*Markers\*\*

character description

``'.'`` point marker

``','`` pixel marker

``'o'`` circle marker

``'v'`` triangle\_down marker

``'^'`` triangle\_up marker

``'<'`` triangle\_left marker

``'>'`` triangle\_right marker

``'1'`` tri\_down marker

``'2'`` tri\_up marker

``'3'`` tri\_left marker

``'4'`` tri\_right marker

``'8'`` octagon marker

``'s'`` square marker

``'p'`` pentagon marker

``'P'`` plus (filled) marker

``'\*'`` star marker

``'h'`` hexagon1 marker

``'H'`` hexagon2 marker

``'+'`` plus marker

``'x'`` x marker

``'X'`` x (filled) marker

``'D'`` diamond marker

``'d'`` thin\_diamond marker

``'|'`` vline marker

``'\_'`` hline marker

* **linestyle=”--“** to change the style of line in graph

\*\*Line Styles\*\*

character description

============= ===============================

``'-'`` solid line style

``'--'`` dashed line style

``'-.'`` dash-dot line style

``':'`` dotted line style

* **linewidth=4** to change the width of line
* **markersize=10** to change the size of dot in graph

eg. plt.plot(days, temp, color="g", marker ='o', linestyle="--", linewidth=4,

markersize=10)

**godmode to give color, marker and linestyle**

eg. plt.plot(days, temp, "go--", linewidth=4, markersize=10)

* **fontsize=20** to change the size of title

eg. plt.title("Varanasi Temp", fontsize=20)

* **plt.legend(["Temperature Line"])** to give legend to graph line
* **plt.legend(["Temperature Line"], loc=4)** position the legend(between 0,1,2,3,4)
* **style.use("ggplot")** to show grid in background of the graph, to use this style import style from matlplotlib **from matplotlib import style**
* **plt.grid(color='r', linestyle='-', linewidth=2)** to change the color of grid linestylr and linewidth
* **Use plot method multiple time to plot may graph**

**Eg.** plt.plot(days, temp, "go--", linewidth=4, markersize=10, label="Vns Temperature")

plt.plot(days, mumbai\_temp, "ko-", linewidth=4, markersize=10, label="mumbai Temperature")

x=days

y=temp

import matplotlib.pyplot as plt

days = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]

temp = [36.6,25.5,28,51,51,25,65.0,45.2,44,69,49.5,25.3,48.2,44.6,48.5]

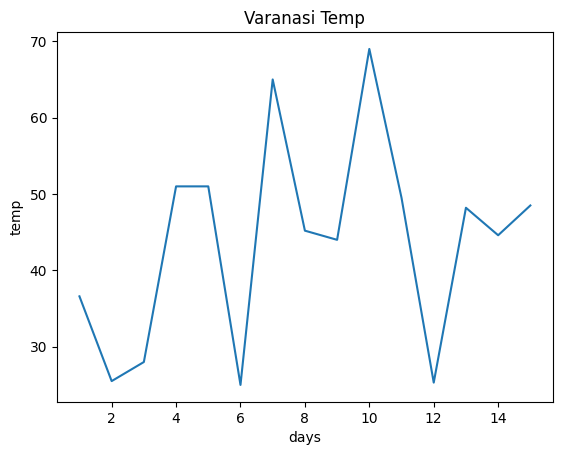
plt.plot(days, temp)

plt.title("Varanasi Temp")

plt.xlabel("days")

plt.ylabel("temp")

plt.show()



**HISTOGRAM PLOT**

To show information in bar in called histogram.

* **plt.hist(ml\_student\_age)**, hist method is use to plot histogram
* **bins is use to define range**

bins=[15,20,25,30,35,40,45]

plt.hist(ml\_student\_age, bins)

* **r.width=0.8** provide space between each bars

**BAR CHART**

* **plt.bar(classes, class1\_students)**, plt.bar method is use to plot a bar chart

Syntax: plt.bar(x,y), where x is classes and y is class1\_studennts.

* **plt.barh(classes, class1\_students),** barh method is use to plot a bar in horizontal.

**Parameter in bar chart:**

plt.bar(

x,

height,

width=0.8,

bottom=None,

\*,

align='center',

data=None,

\*\*kwargs,

)

alpha: scalar or None

angle: unknown

animated: bool

antialiased or aa: bool or None

bounds: (left, bottom, width, height)

capstyle: `.CapStyle` or {'butt', 'projecting', 'round'}

clip\_box: `.Bbox`

clip\_on: bool

clip\_path: Patch or (Path, Transform) or None

color: color

edgecolor or ec: color or None

facecolor or fc: color or None

figure: `.Figure`

fill: bool

gid: str

hatch: {'/', '\\', '|', '-', '+', 'x', 'o', 'O', '.', '\*'}

height: unknown

in\_layout: bool

joinstyle: `.JoinStyle` or {'miter', 'round', 'bevel'}

label: object

linestyle or ls: {'-', '--', '-.', ':', '', (offset, on-off-seq), ...}

linewidth or lw: float or None

mouseover: bool

path\_effects: `.AbstractPathEffect`

picker: None or bool or float or callable

rasterized: bool

sketch\_params: (scale: float, length: float, randomness: float)

snap: bool or None

transform: `.Transform`

url: str

visible: bool

width: unknown

x: unknown

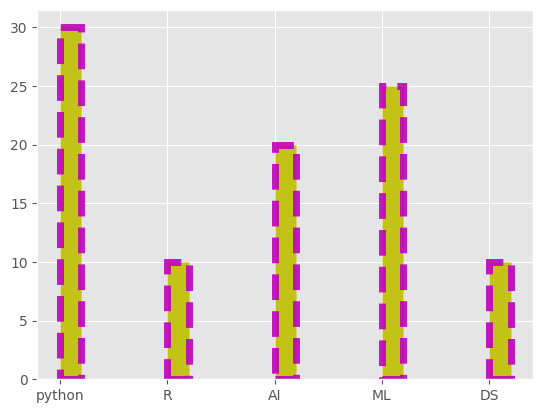
xy: (float, float)

y: unknown

zorder: float

**eg.:** plt.bar(classes, class1\_students, width=0.2, align="edge", color="y",edgecolor="m", linewidth=5, alpha =0.9, linestyle= "--", label ="Class 1 Students", visible=False)

**OUTPUT:**

****

* **plt.figure(figsize=(16,9)),**  use to change the size of bar figure

**SCATTER PLOT**

* **df\_covid = pd.read\_csv("E:\\programms\matplotlib\_tutorial\owid-covid-data.csv")**, to read the csv file from system
* **df\_covid.shape**, to know to rows and columns
* **plt.scatter(x,y)**, to plot the scatter plot

**Parameters in scatter plot**

plt.scatter(

x,

y,

s=None,

c=None,

marker=None,

cmap=None,

norm=None,

vmin=None,

vmax=None,

alpha=None,

linewidths=None,

\*,

edgecolors=None,

plotnonfinite=False,

data=None,

\*\*kwargs,

)

x, y : float or array-like, shape (n, )

The data positions.

s : float or array-like, shape (n, ), optional

The marker size in points\*\*2.

Default is ``rcParams['lines.markersize'] \*\* 2``.

c : array-like or list of colors or color, optional

The marker colors. Possible values:

- A scalar or sequence of n numbers to be mapped to colors using

\*cmap\* and \*norm\*.

- A 2D array in which the rows are RGB or RGBA.

- A sequence of colors of length n.

- A single color format string.

**PIE CHART**

classes = ['python', 'R', 'AI', 'ML', 'DS']

class1\_students=[30,10,20,25,10]

explode = [0.3,0,0,0.2,0]

colors =["c","b","r","m","g"]

textprops ={"fontsize": 15}

plt.pie(class1\_students, labels = classes, explode= explode, colors=colors, autopct="%0.1f%%", shadow=True, radius=1.4,startangle = 270, textprops=textprops)

plt.show()

* **plt.pie(),** is use to plot pie chart

**Parameter in pie chart:**

plt**.**pie**(**

x**,**

explode**=None,**

labels**=None,**

colors**=None,**

autopct**=None,**

pctdistance**=0.6,**

shadow**=False,**

labeldistance**=1.1,**

startangle**=0,**

radius**=1,**

counterclock**=True,**

wedgeprops**=None,**

textprops**=None,**

center**=(0,** **0),**

frame**=False,**

rotatelabels**=False,**

**\*,**

normalize**=True,**

data**=None,**

**)**

**Subplot**

Subplot is use when we want to plot many plot.

* **Plt.subplot(x,y,index),** is use to plot subplot. Where x is row y is columns and index.

**Eg.:**

plt.subplot(2,2,1)

plt.subplot(2,2,2)

plt.subplot(2,2,3)

plt.subplot(2,2,4)

**This upper code will make 4 section of subplot.**

**SAVE FIGURE**

How to save figure (plot and chart)?

To save ant chart first create that chart then use this method to save the file in working directory.

* **Plt.savefig(),** is use to save figure
* Writ this code above show function.

**Parameter of savefig**

savefig(fname, \*, dpi='figure', format=None, metadata=None,

bbox\_inches=None, pad\_inches=0.1,

facecolor='auto', edgecolor='auto',

backend=None, \*\*kwargs

)

**Image show and Color bar using Matplotlib**

* **import matplotlib.image as mpimg** import this module to read image from drive
* **img = mpimg.imread("pie\_chart.png")**, to read the image
* **type(img)**, to check the type of image
* **img.shape**
* **plt.imshow(img),** to show the image
* **plt.axis("off"),** to hide the axis
* **plt.colorbar()**, display colorbar aside

Eg.:-

single\_channel\_img2 =img2[:,:,1]

img2 = mpimg.imread("H:\\Wallpapers\\wrath.jpg")

plt.imshow(img2)

plt.axis("off")

plt.imshow(single\_channel\_img2,cmap = "hot")

plt.colorbar()

plt.show()