**Visualization using matplotlib:**

**Import**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from matplotlib import rcParams

import seaborn as sb

setting configs plot size 5\*4 inches and seaborn style as whitegrid

**Syntax :**

rcParams['figure.figsize']=5,4

# rcParams means parameters to configure, here value 5,4 is standard

**Creating a linechart :**

Default plotting is line chart

**Scatter Plot:**

Mostly used data reprersntation

**Seaborn:**

seaborn is core of matplot

**DataFrame:**

pandas.core.frame.DataFrame

**Syntax:**

type(cars)

**Plot Several Variables:**

**Syntax**

df = cars.loc[:,['cyl','mpg','wt']]

Here [: denotes all rows , ['selected col 1','selected col 2','selected col 3'] ]

**Print:**

**Syntax :**

plt.show()

just for print mechanism

**Default chart:**

**Syntax:**

cars.mpg.plot(kind='barh')

default chart is line,bar = normal vertical,barh= horizontal

**pie chart:**

**Syntax:**

plt.pie(sales,[0,0,0.1,0])

If we need to highlight or pull something out. 0.1 and 0.2 are usual standards to showcase the target data

**Highlight the data with respect to our cars:**

**Syntax:**

cars.mpg[3:9].plot(kind='pie')

**Saving a Plot:**

**Syntax:**

plt.savefig('pie\_21mar.png',format='png')

**Object Oriented Plotting:**

# Steps for object oriented plotting

# 1.Create a blank fig obj or canvas

# 2.Add axes

# 3.Generate a plot within object

# 4.Specify plotting and layout parameters

**# Creates canvas**

**Syntax:**

fig = plt.figure()

**# Add the axes**

Syntax:

ax =fig.add\_axes([0,0,0.5,0.5])

**# Generate a plot**

Syntax:

ax.plot(x,y)

plt.show()

**Setting axes,limits and ticks:**

**Syntax:**

x = [1,2,3,4,5,6,7,8,9,10]

y = [1,2,7,4,0,4,3,2,9,5]

**# Create canvas**

fig = plt.figure()

**# Add axes**

ax = fig.add\_axes([0,0,1,1])

ax.set\_xlim([2,7]) # set xlimit

ax.set\_ylim([0,5]) # set ylimit

ax.set\_xticks([1,2,3,4,5,7,10])

set xticksv(1,2,3,4,5,7,10) here we are specifying the x\_ticks data to reflect in the x axis

ax.set\_yticks([0,1,2,3,4,5,8.5, max(y)])

set yticks(0,1,2,3,4,5,8.5) here we are specifying the x\_ticks data to reflect in the y axis

**# note here in yticks one value is float so all values typecasted into float data type**

**# Plotting on the axes**

ax.plot(x,y)

plt.show()

**Generating multiple plots:**

fig,ax=plt.subplots(2,3)

# 2,3 means 2 rows and 3 columns

**plot fomatting:**

plt.bar(x,y,width=[0.5,0.5,0.5,0.1,0.5],color=['red','blue','yellow','green','orange'])

**# width can be used to show 3rd dimension**

plt.plot(x,y,lw=2,ls='-.',color='orange',marker='p',mew=25,markerfacecolor='blue')

# lw stands for line width 2 is preferable,line style is .,\_etc,marker is define the shape,mew is marker width,markerfacecolor is the inside the shape of the color

**labels and Annotation:**

plt.xlabel('year',fontsize=18,color='red') # xlabel is to define the x-axis

plt.ylabel('sales in the USD',fontsize=18,color='green') # ylabel is to define the y-axis

**legend:**

plt.legend(label\_names,loc='lower right') # legend is to show one single side with the respective colors,loc represents is up,down,right,left

**ticks and labels:**

ax.set\_xticks(range(32))

**# contains the excel table range**

ax.set\_xticklabels(cars.car\_model,rotation=45,fontsize=10)

**# In this type rotation is based on labels**

ax.set\_title('Miles per galon of cars in mtcars')

**# this the graph title**

ax.set\_xlabel('car names')

ax.set\_ylabel('Miles')

ax.set\_ylim([0,45])

ax.annotate('ToyotaCorolla',fontsize=10,xy=(19,33.9),xytext=(20,40),arrowprops=dict

(facecoor='green'))

#Annotate it is used for highlight purpose,#19 is the row,33.9 is the mileage