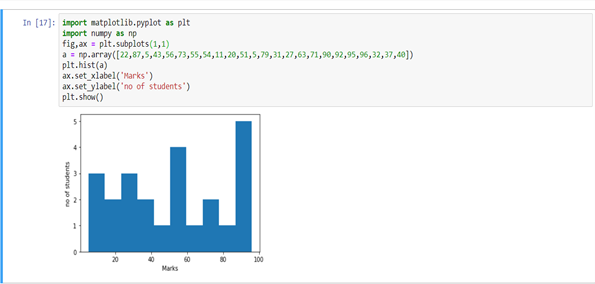
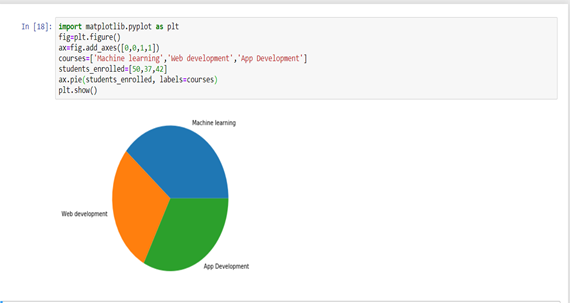
## Univariate Analysis

Uni means one and variate means variable, so in univariate analysis, there is only one dependable variable. The objective of univariate analysis is to derive the data, define and summarize it, and analyze the pattern present in it. In a dataset, it explores each variable separately. It is possible for two kinds of variables- Categorical and Numerical.

1.Histogram

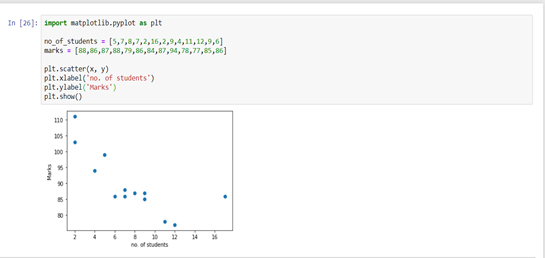


2.Pie Chart



## 2.Bivariate Analysis

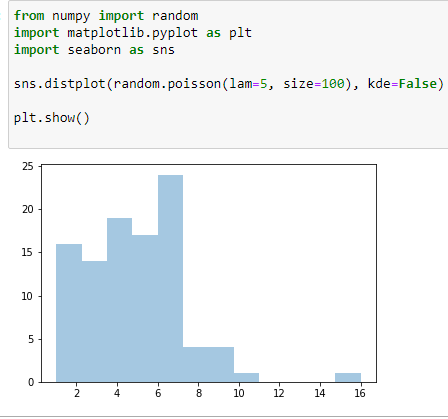
Bi means two and variate means variable, so here there are two variables. The analysis is related to cause and the relationship between the two variables.



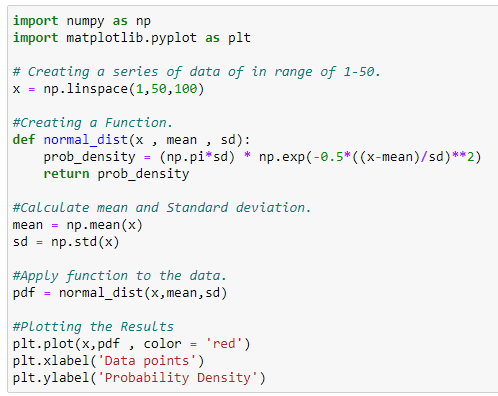
Probability Distributions

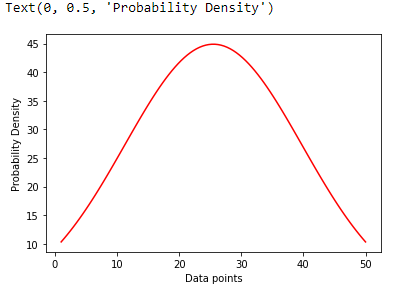
1.Discrete Probability Distributions

**1.Poisson Distribution**

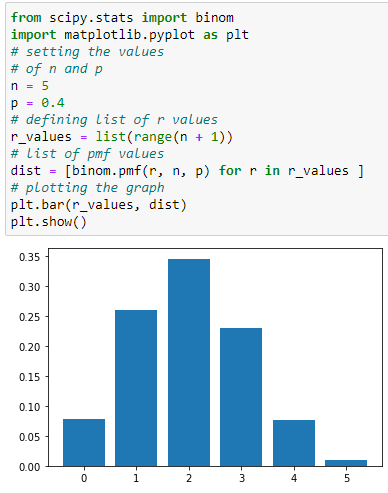


2.Normal Distribution

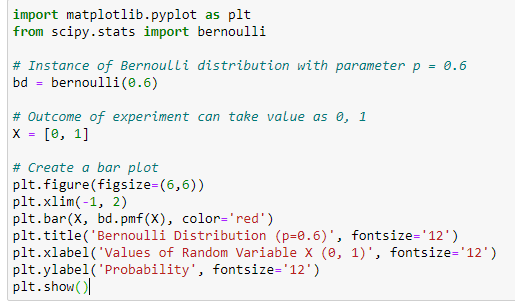




#### 3.Binomial Distribution

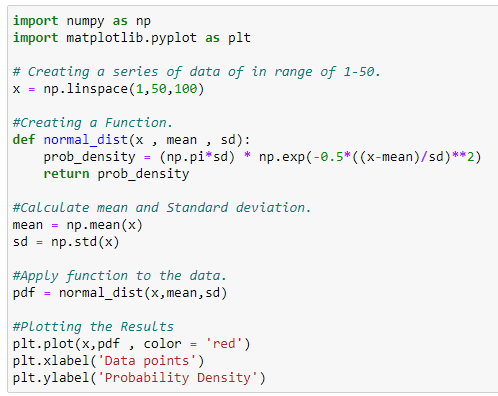


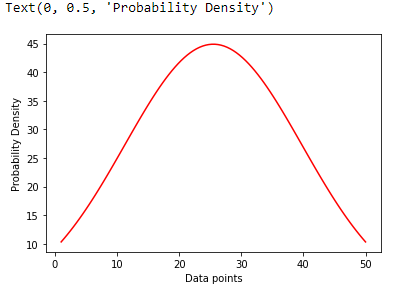
4.Bernoulli's Distribution

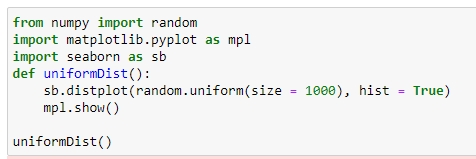


### 2.Continuous Probability Distributions

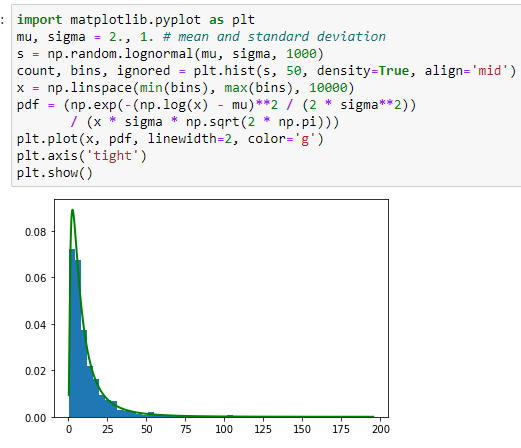
#### 1.Normal Distribution



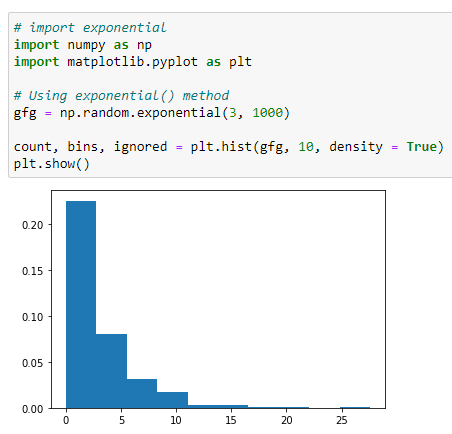


2.Continuous Uniform Distribution

3.Log-Normal Distribution



4.Exponential Distribution



**Handling missing values**

**1. using isnull()and notnull()**

import pandas as pd

 importnumpy as np

 dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, 45, 56, np.nan],

        'Third Score':[np.nan, 40, 80, 98]}

 df = pd.DataFrame(dict)

 df.isnull()

import pandas as pd

data = pd.read\_csv("employees.csv")

 bool\_series = pd.isnull(data["Gender"])

 data[bool\_series]

import pandas as pd

importnumpy as np

 dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, 45, 56, np.nan],

        'Third Score':[np.nan, 40, 80, 98]}

df = pd.DataFrame(dict)

 df.notnull()

import pandas as pd

data = pd.read\_csv("employees.csv")

bool\_series = pd.notnull(data["Gender"])

data[bool\_series]

2.filling a missing value with 0

import pandas as pd

 importnumpy as np

 dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, 45, 56, np.nan],

        'Third Score':[np.nan, 40, 80, 98]}

 df = pd.DataFrame(dict)

df.fillna(0)

3. filling a missing value with previous ones/ffill and bfill using fillna() methods

import pandas as pd

importnumpy as np

dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, 45, 56, np.nan],

        'Third Score':[np.nan, 40, 80, 98]}

df = pd.DataFrame(dict)

df.fillna(method ='pad')

import pandas as pd

 importnumpy as np

dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, 45, 56, np.nan],

        'Third Score':[np.nan, 40, 80, 98]}

 df = pd.DataFrame(dict)

 df.fillna(method ='bfill')

4.Printing the first 10 to 24 rows of the data frame for visualization

import pandas as pd

data = pd.read\_csv("employees.csv")

 data[10:25]

import pandas as pd

data = pd.read\_csv("employees.csv")

data["Gender"].fillna("No Gender", inplace = True)

data

5.replace  Nan value in dataframe with value -99

import pandas as pd

data = pd.read\_csv("employees.csv")

data.replace(to\_replace = np.nan, value = -99)

6.interpolate the missing values

import pandas as pd

df = pd.DataFrame({"A":[12, 4, 5, None, 1],

                   "B":[None, 2, 54, 3, None],

                   "C":[20, 16, None, 3, 8],

                   "D":[14, 3, None, None, 6]})

 df

df.interpolate(method ='linear', limit\_direction ='forward')

7.using dropna() function

import pandas as pd

 importnumpy as np

 dict = {'First Score':[100, 90, np.nan, 95],

        'Second Score': [30, np.nan, 45, 56],

        'Third Score':[52, 40, 80, 98],

        'Fourth Score':[np.nan, np.nan, np.nan, 65]}

 df = pd.DataFrame(dict)

 df.dropna()

import pandas as pd

 importnumpy as np

 dict = {'First Score':[100, np.nan, np.nan, 95],

        'Second Score': [30, np.nan, 45, 56],

        'Third Score':[52, np.nan, 80, 98],

        'Fourth Score':[np.nan, np.nan, np.nan, 65]}

 df = pd.DataFrame(dict)

 df.dropna(how = 'all')

import pandas as pd

importnumpy as np

dict = {'First Score':[100, np.nan, np.nan, 95],

        'Second Score': [30, np.nan, 45, 56],

        'Third Score':[52, np.nan, 80, 98],

        'Fourth Score':[60, 67, 68, 65]}

 df = pd.DataFrame(dict)

df.dropna(axis = 1)

8.making new data frame with dropped NA values

import pandas as pd

data = pd.read\_csv("employees.csv")

new\_data = data.dropna(axis = 0, how ='any')

new\_data

print("Old data frame length:", len(data))

print("New data frame length:", len(new\_data))

print("Number of rows with at least 1 NA value: ", (len(data)-len(new\_data)))

9.use isna() function to detect the missing values in a dataframe.

|  |
| --- |
| Import pandas as pd  sr =pd.Series([12, 5, None, 5, None, 11])  sr  sr.isna() |
|  |

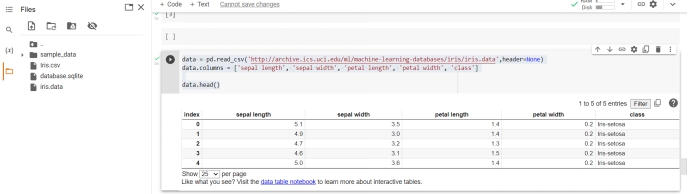
**1.view the first 5 rows of the DataFrame**

import pandas aspd

data = pd.read\_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data',header=None)

data.columns = ['sepal length', 'sepal width', 'petal length', 'petal width', 'class']

data.head()



**2.Compute and print the mean and the standard deviation for each of the 4 measurement columns (i.e. sepal length and width, petal length and width).**

From pandas.api.types import is\_numeric\_dtype

for col in data.columns:

if is\_numeric\_dtype(data[col]):

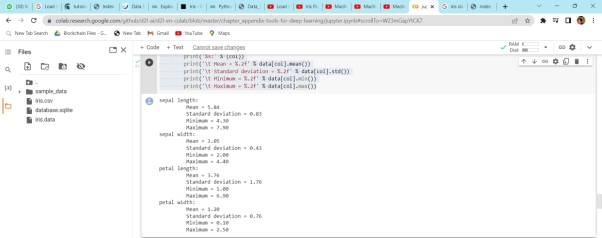
print('%s:' % (col))

print('\t Mean = %.2f' % data[col].mean())

print('\t Standard deviation = %.2f' % data[col].std())

print('\t Minimum = %.2f' % data[col].min())

print('\t Maximum = %.2f' % data[col].max())

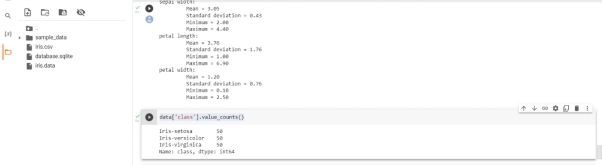


**3.count the no of values in class cloumn**

import pandas aspd

data = pd.read\_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data',header=None)

data['class'].value\_counts()

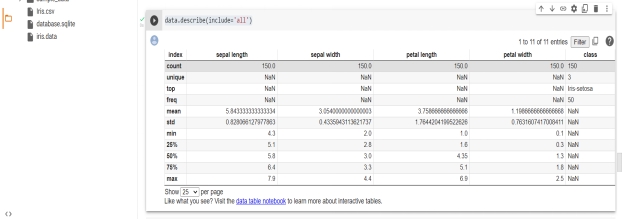


4.Display all function on iris data

import pandas aspd

data = pd.read\_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data',header=None)

data.describe(include='all')



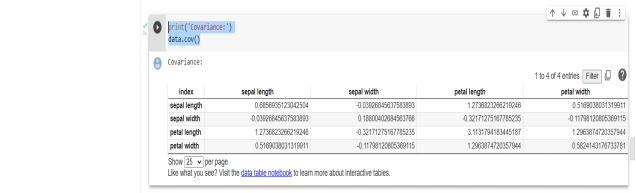
**5.Finding relationship in data - Covariance method**

import pandas aspd

data = pd.read\_csv('http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data',header=None)

print('Covariance:')

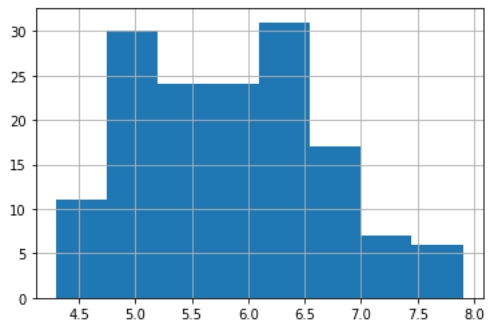
data.cov()



6.Plot a histogram graph of particular coloumn

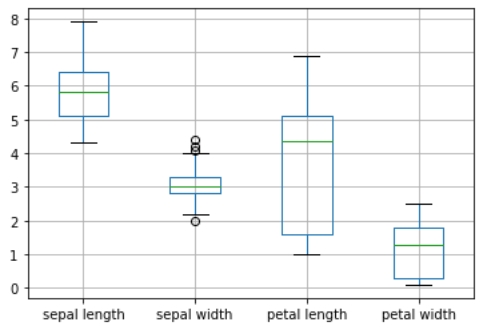
%matplotlibinline

data['sepal length'].hist(bins=8)



7.plot boxpolt

data.boxplot()



8.Scatter plot

Import matplotlib.pyplot as plt

fig, axes = plt.subplots(3, 2, figsize=(12,12))

index = 0

foriinrange(3):

for j inrange(i+1,4):

ax1 = int(index/2)

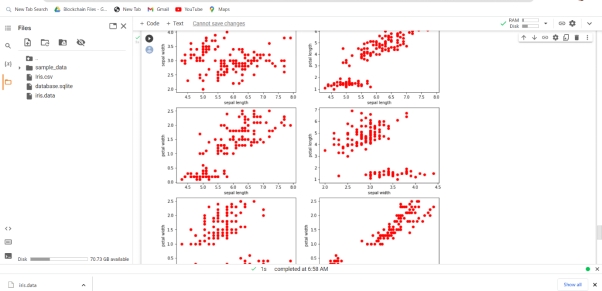
ax2 = index % 2

axes[ax1][ax2].scatter(data[data.columns[i]], data[data.columns[j]], color='red')

axes[ax1][ax2].set\_xlabel(data.columns[i])

axes[ax1][ax2].set\_ylabel(data.columns[j])

index = index + 1



## Parallel Coordinates Plot

From pandas.plotting import parallel\_coordinates

%matplotlibinline

parallel\_coordinates(data, 'class')

