**WEEK-5**

**AIM:**

Write a program to perform Exploratory Data Analysis on real time datasets using the following approaches:

a) Univariate Analysis

b) Multivariate Analysis

c) Visualization using correlation matrix

**Description:**

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.

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.

Multivariate analysis: is required when more than two variables have to be analyzed simultaneously. It is a tremendously hard task for the human brain to visualize a relationship among 4 variables in a graph and thus multivariate analysis is used to study more complex sets of data. Types of Multivariate Analysis include Cluster Analysis, Factor Analysis, Multiple Regression Analysis, Principal Component Analysis, etc. More than 20 different ways to perform multivariate analysis exist and which one to choose depends upon the type of data and the end goal to achieve.

**Code**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

ds=pd.read\_csv('/content/drive/MyDrive/datasets/Iris.csv')

ds.head()

Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm Species

0 1 5.1 3.5 1.4 0.2 Iris-setosa

1 2 4.9 3.0 1.4 0.2 Iris-setosa

2 3 4.7 3.2 1.3 0.2 Iris-setosa

3 4 4.6 3.1 1.5 0.2 Iris-setosa

4 5 5.0 3.6 1.4 0.2 Iris-setosa

print(ds.describe())

Id SepalLengthCm SepalWidthCm PetalLengthCm PetalWidthCm

count 150.000000 150.000000 150.000000 150.000000 150.000000

mean 75.500000 5.843333 3.054000 3.758667 1.198667

std 43.445368 0.828066 0.433594 1.764420 0.763161

min 1.000000 4.300000 2.000000 1.000000 0.100000

25% 38.250000 5.100000 2.800000 1.600000 0.300000

50% 75.500000 5.800000 3.000000 4.350000 1.300000

75% 112.750000 6.400000 3.300000 5.100000 1.800000

max 150.000000 7.900000 4.400000 6.900000 2.500000

ds.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 150 entries, 0 to 149

Data columns (total 6 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Id 150 non-null int64

1 SepalLengthCm 150 non-null float64

2 SepalWidthCm 150 non-null float64

3 PetalLengthCm 150 non-null float64

4 PetalWidthCm 150 non-null float64

5 Species 150 non-null object

dtypes: float64(4), int64(1), object(1)

memory usage: 7.2+ KB

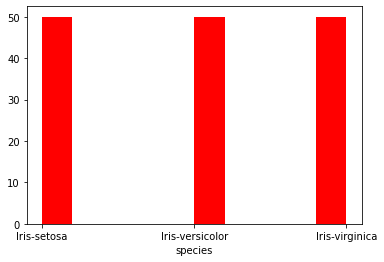
#Numerical Data

#Histogram

plt.hist(ds['Species'],color="red")

plt.xlabel('species')

ext(0.5, 0, 'species')

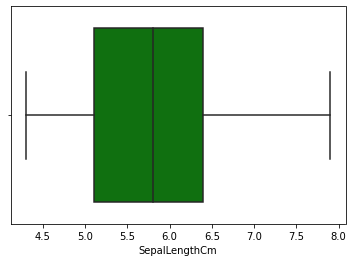


sns.boxplot(ds['SepalLengthCm'], color="green")

/usr/local/lib/python3.7/dist-packages/seaborn/\_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f95128f4d90>

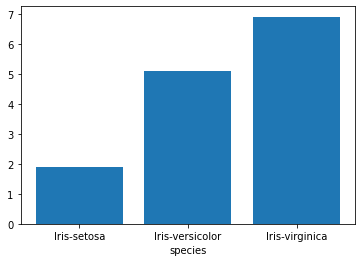


#barplot

plt.bar(ds['Species'],ds['PetalLengthCm'])

plt.xlabel('species')

Text(0.5, 0, 'species')



ds['Species'].value\_counts().plot.bar()

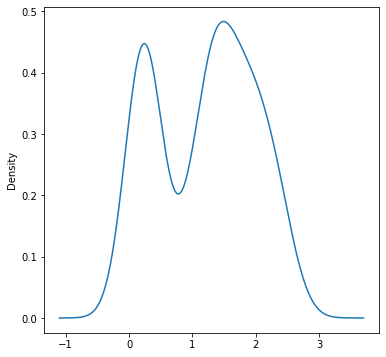
<matplotlib.axes.\_subplots.AxesSubplot at 0x7fb286d85110>



plt.figure(figsize=(6,6))

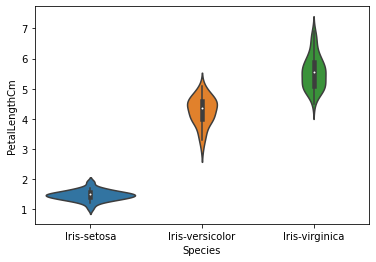
ds['PetalWidthCm'].plot(kind="density")

<matplotlib.axes.\_subplots.AxesSubplot at 0x7fb286af0750>



sns.violinplot(x='Species',y='PetalLengthCm',size=7,data=ds)

<matplotlib.axes.\_subplots.AxesSubplot at 0x7f951268b110>



n=ds["Species"].unique()

print(n)

['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']

v=ds['Species'].value\_counts()

x=(np.array(v)).tolist()

print(x)

[50, 50, 50]

plt.pie(x,labels=n,autopct='%.0f%%')

([<matplotlib.patches.Wedge at 0x7f95120c3590>,

<matplotlib.patches.Wedge at 0x7f95120c3c90>,

<matplotlib.patches.Wedge at 0x7f951204e550>],

[Text(0.5499999702695115, 0.9526279613277875, 'Iris-setosa'),

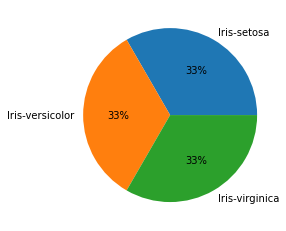
Text(-1.0999999999999954, -1.0298943258065002e-07, 'Iris-versicolor'),

Text(0.5500001486524352, -0.9526278583383436, 'Iris-virginica')],

[Text(0.2999999837833699, 0.5196152516333385, '33%'),

Text(-0.5999999999999974, -5.6176054134900006e-08, '33%'),

Text(0.30000008108314646, -0.5196151954572783, '33%')])



# **Bivariate analysis And Multivariate Analysis**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

df=pd.read\_csv('/content/drive/MyDrive/datasets/Linear Regression/titanic.csv

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 891 entries, 0 to 890

Data columns (total 12 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 PassengerId 891 non-null int64

1 Survived 891 non-null int64

2 Pclass 891 non-null int64

3 Name 891 non-null object

4 Sex 891 non-null object

5 Age 714 non-null float64

6 SibSp 891 non-null int64

7 Parch 891 non-null int64

8 Ticket 891 non-null object

9 Fare 891 non-null float64

10 Cabin 204 non-null object

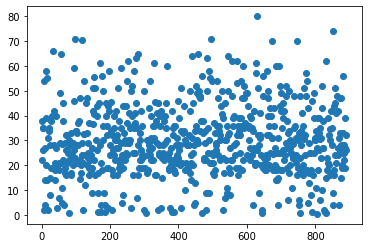
11 Embarked 889 non-null object

dtypes: float64(2), int64(5), object(5)

memory usage: 83.7+ KB

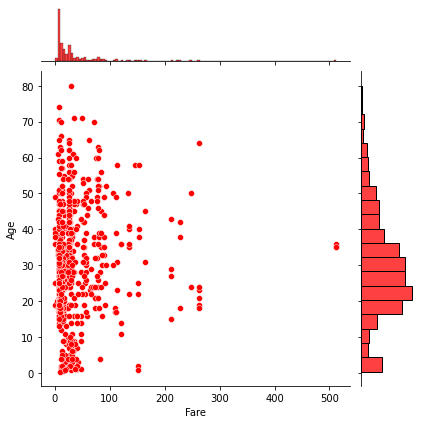
plt.scatter(df['PassengerId'],df['Age'])

<matplotlib.collections.PathCollection at 0x7f5d83108a90>

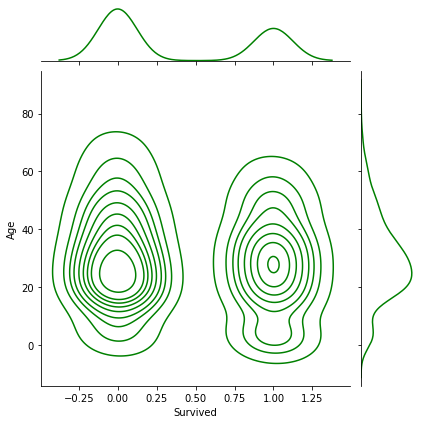


sns.jointplot(x='Fare',y='Age',data=df,color='r')

<seaborn.axisgrid.JointGrid at 0x7f5d8308d3d0>

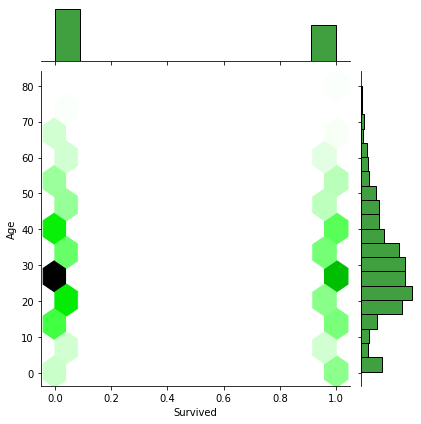


sns.jointplot(x='Survived',y='Age',data=df,color='g',kind="kde")

<seaborn.axisgrid.JointGrid at 0x7f5d80149450>

sns.jointplot(x='Survived',y='Age',data=df,color='g',kind="hex")

<seaborn.axisgrid.JointGrid at 0x7f5d8014b410>



sns.pairplot(df,hue='Survived')

<seaborn.axisgrid.PairGrid at 0x7f5d7d6a1750>

