Experiment No 7

7.1 Aim/Purpose of the Experiment

To familiarize the students with data visualization using one feature variables.

7.2 Prerequisites

Basic knowledge of programming, python syntax, matplotlib, seaborn, different libraries.

7.3 Materials/Equipment/Apparatus / Devices/Software required

Jupyter Notebook.

7.4 Introduction and Theory

Univariate analysis is a statistical method used to describe and understand the distribution, central tendency, and variability of a single variable. In Python, you can perform univariate analysis using libraries such as NumPy, Pandas, and Matplotlib/Seaborn for data manipulation, analysis, and visualization. Here's a brief outline of the process:

- Data Preparation: Load your dataset into a Pandas DataFrame and clean/preprocess the data if necessary. Ensure that the variable of interest is properly formatted and ready for analysis.
- Descriptive Statistics: Compute descriptive statistics for the variable of interest. This includes measures such as mean, median, mode, standard deviation, variance, minimum, maximum, and quartiles. These statistics provide an initial understanding of the distribution and characteristics of the variable.

Code:

```
import pandas as pd

import numpy as np

df=pd.read_csv('iris.csv')

df

#Finding the data types of variables in the DataFrame

df.dtypes

#Importing libraries essential for data visualization

#MATPLOTLIB

import matplotlib.pyplot as plt

%matplotlib inline

#SEABORN

import seaborn as sns

#Plots for continuous variables' analysis

#ENUMERATIVE PLOTS

#UNIVARIATE SCATTER PLOT

plt.scatter(df.index,df['sepal.width'])

plt.show()

sns.scatterplot(x=df.index,y=df['sepal.width'],hue=df['variety'])

#LINE PLOT WITH MARKERS
```

```
#Setting title, figure size, labels and font size in matplotlib
plt.figure(figsize=(6,6))
plt.title('Line plot of petal length')
plt.xlabel('index',fontsize=20)
plt.ylabel('petal length',fontsize=20)
plt.plot(df.index,df['petal.length'],markevery=1,marker='d')
for name, group in df.groupby('variety'):
plt.plot(group.index, group['petal.length'], label=name,markevery=1,marker='d')
plt.legend()
plt.show()
#Setting title, figure size,labels and font size in seaborn
sns.set(rc={'figure.figsize':(7,7)})
sns.set(font_scale=1.5)
fig=sns.lineplot(x=df.index,y=df['petal.length'],markevery=1,marker='d',data=df,hue=df['variety'])
fig.set(xlabel='index')
#STRIP PLOT
sns.stripplot(y=df['sepal.width'])
# Strip-plot(category wise)
sns.stripplot(x=df['variety'],y=df['sepal.width'])
#SWARM PLOT
#Setting figure size
sns.set(rc={'figure.figsize':(5,5)})
#Swarm-plot
sns.swarmplot(x=df['sepal.width'])
#Swarm-plot category wise
sns.swarmplot(x=df['variety'],y=df['sepal.width'])
#SUMMARY PLOTS
#HISTOGRAM
plt.hist(df['petal.width'])
sns.distplot(df['petal.width'],kde=False,color='black',bins=10)
#DENSITY PLOT
plt.figure(figsize=(5,5))
```

```
df['petal.length'].plot(kind='density')
sns.set(rc={'figure.figsize':(5,5)})
sns.kdeplot(df['petal.length'],shade=True)
#RUG PLOT
fig, ax = plt.subplots()
sns.rugplot(df['sepal.length'])
ax.set_xlim(3,9)
plt.show()
from scipy import stats
import numpy as np
kdf=df['sepal.length'].to_numpy()
rdf=np.hstack(kdf)
density = stats.kde.gaussian_kde(rdf)
x = np.arange(3,9,0.1)
plt.plot(x, density(x))
plt.plot(rdf,[0.01]*len(rdf), '|')
sns.distplot(df['sepal.length'],rug=True,hist=False)
#BOX PLOT
plt.boxplot(df['sepal.width'])
#Removing the column with categorical variables
dfM=df.drop('variety',axis=1)
plt.figure(figsize=(9,9))
#Set Title
plt.title('Box plots of the 4 variables')
plt.boxplot(dfM.values,labels=['SepalLength','SepalWidth','PetalLength','PetalWidth'])
sns.boxplot(df['sepal.width'])
sns.set(rc={'figure.figsize':(9,9)})
sns.boxplot(x="variable", y="value", data=pd.melt(dfM))
#distplot()
sns.set(rc={'figure.figsize':(6,6)})
sns.distplot(df['petal.length'],color='black',rug=True)
#VIOLIN PLOT
```

```
plt.figure(figsize=(7,7))
plt.violinplot(dfM.values,showmedians=True)
sns.set(rc={'figure.figsize':(5,5)})
sns.violinplot(df['sepal.width'],orient='vertical')

sns.set(rc={'figure.figsize':(9,9)})
sns.violinplot(x=df['variety'], y=df['petal.width'],data=df)
#Plots for categorical variables' analysis
#BAR PLOT
df['variety'].value_counts().plot.bar()
sns.countplot(df['variety'])
#PIE CHART
plt.pie(df['variety'].value_counts(),labels=['SETOSA','VERSICOLOR','VIRGINICA'],shadow=True)
df1=df.sample(frac=0.35)
plt.figure(figsize=(5,5))
plt.pie(df1['variety'].value_counts(),startangle=90,autopct='%.3f',labels=['SETOSA','VERSICOLOR','VIRGINICA'],shadow=True)
```

CONCLUSION :

The theoretical and practical aspects are same for this experiment.