ASSIGNMENT-1

Roshan Thomas

RMCA-B

Roll no: 15

Date: 13/12/2021

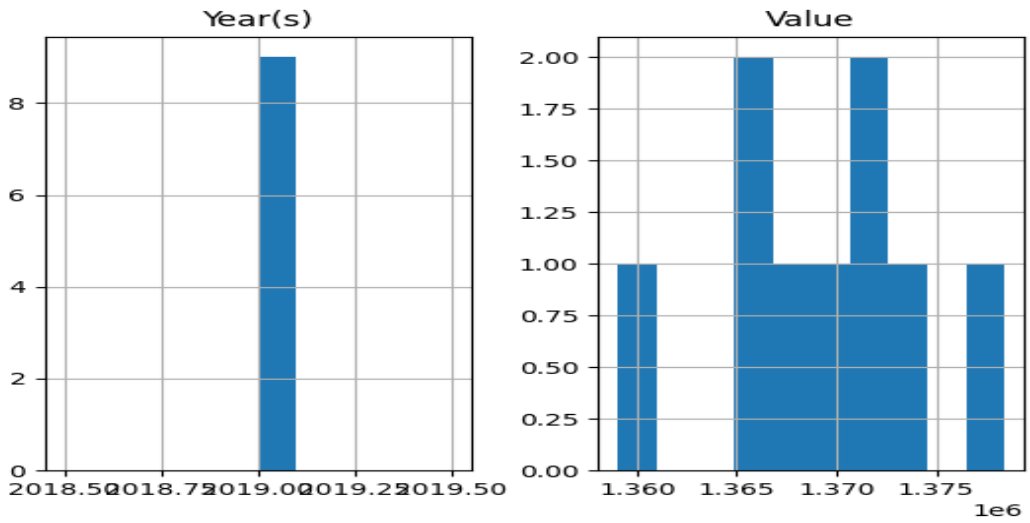
1. HISTOGRAM

The histogram represents the frequency of occurrence of specific phenomena which lie within a specific range of values and arranged in consecutive and fixed intervals.

PROGRAM

```
import pandas as pd
import matplotlib.pyplot as plt
data = [
['India', 2019, 'Medium', 1368737.513],
['India', 2019, 'High', 1378419.072],
['India', 2019, 'Low', 1359043.965],
['India', 2019, 'Constant fertility', 1373707.838],
['India', 2019,'Instant replacement', 1366687.871],
['India', 2019, 'Zero migration', 1370868.782],
['India', 2019,'Constant mortality', 1366282.778],
['India', 2019, 'No change', 1371221.64],
['India', 2019, 'Momentum', 1367400.614],]
df = pd.DataFrame(data, columns = ([ 'Country or Area', 'Year(s)', 'Variant', 'Value']))
df.hist()
plt.show()
```

OUTPUT

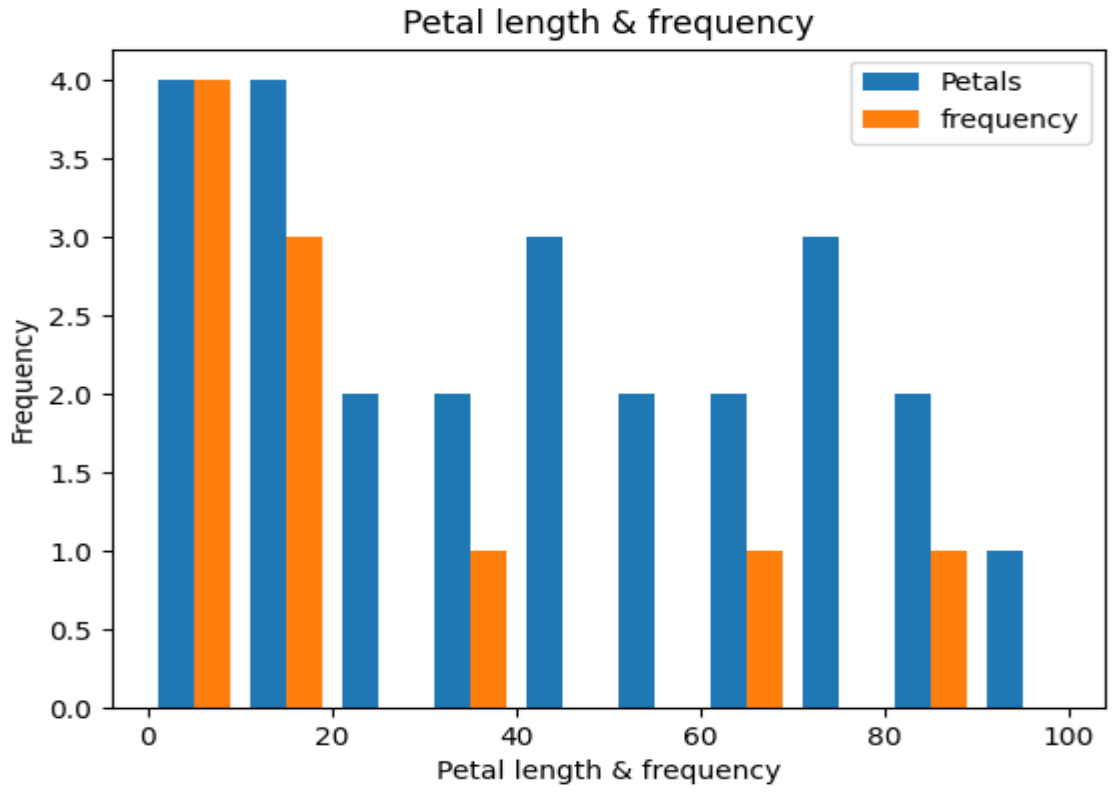


2. CLASS STRATIFIED HISTOGRAM

PROGRAM

```
from matplotlib import pyplot as plt
import numpy as np
a = np.array([2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,
        71,73,79,83,89,97]) # primes
b=np.array([2,3,5,7,13,17,19,31,61,89]) # exponents
bins = [0,10,20,30,40,50,60,70,80,90,100]
plt.hist([a,b],bins,label=['Petals','frequency'])
plt.legend(loc='upper right')
plt.title("Petal length & frequency")
plt.xlabel("Petal length & frequency")
plt.ylabel("Frequency")
plt.show()
```

OUTPUT

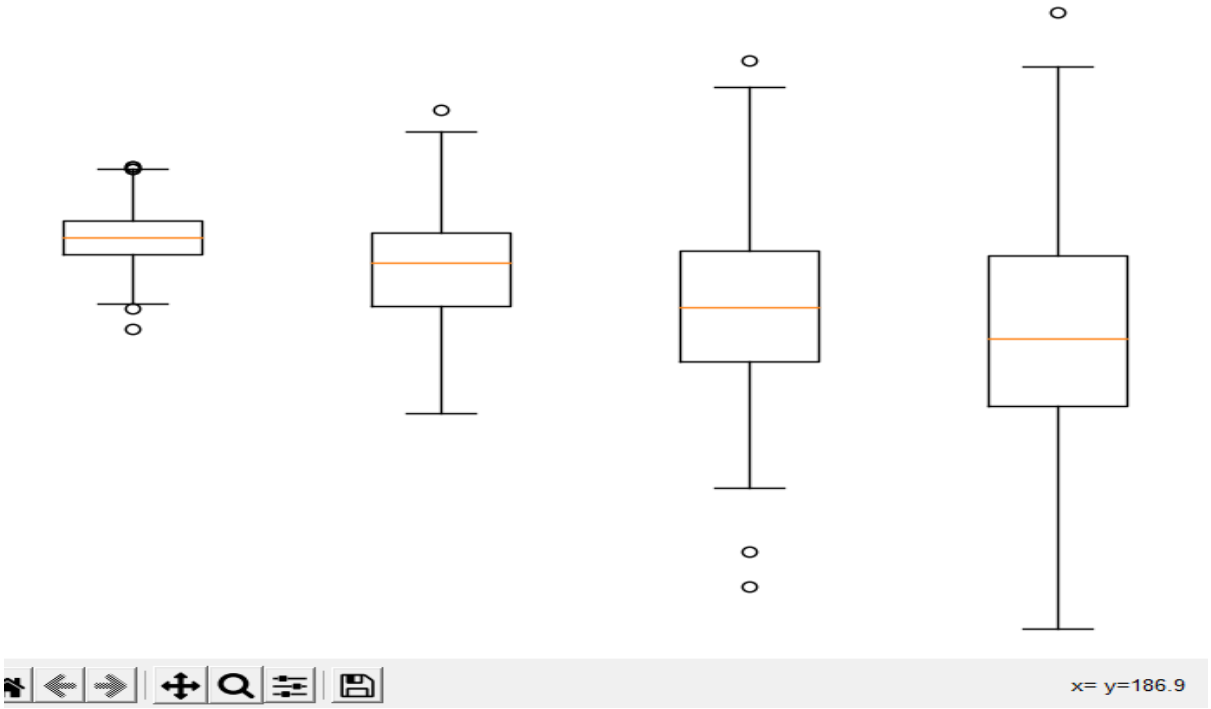


3. BOX WHISKER PLOT(QUANTILE PLOT)

PROGRAM

```
# Import libraries
import matplotlib.pyplot as plt
import numpy as np

# Creating dataset
np.random.seed(10)

data_1 = np.random.normal(100, 10, 200)
data_2 = np.random.normal(90, 20, 200)
data_3 = np.random.normal(80, 30, 200)
data_4 = np.random.normal(70, 40, 200)
data = [data_1, data_2, data_3, data_4]

fig = plt.figure(figsize=(10, 7))

# Creating axes instance
ax = fig.add_axes([0, 0, 1, 1])

# Creating plot
bp = ax.boxplot(data)

# show plot
plt.show()
```

OUTPUT

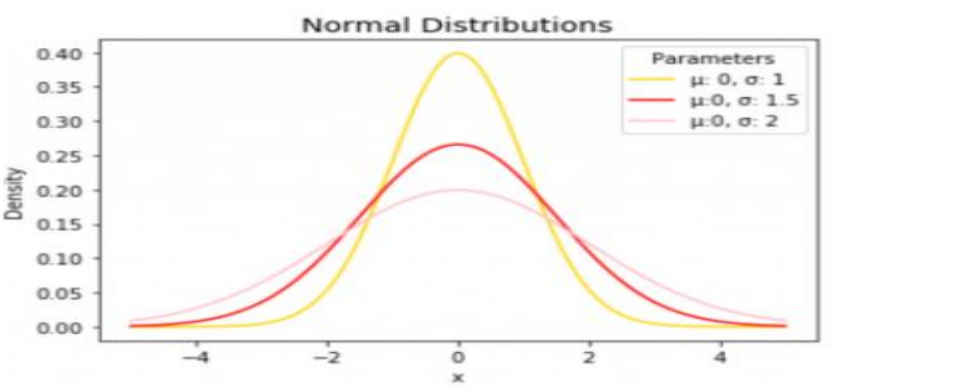


4. DISTRIBUTION PLOT

PROGAM

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

#x-axis ranges from -5 and 5 with .001 steps
x = np.arange(-5, 5, 0.001)

#define multiple normal distributions
plt.plot(x, norm.pdf(x, 0, 1), label='μ: 0, σ: 1', color='gold')
plt.plot(x, norm.pdf(x, 0, 1.5), label='μ:0, σ: 1.5', color='red')
plt.plot(x, norm.pdf(x, 0, 2), label='μ:0, σ: 2', color='pink')

#add legend to plot
plt.legend(title='Parameters')

#add axes labels and a title
plt.ylabel('Density')
plt.xlabel('x')
plt.title('Normal Distributions', fontsize=14)
```

OUTPUT

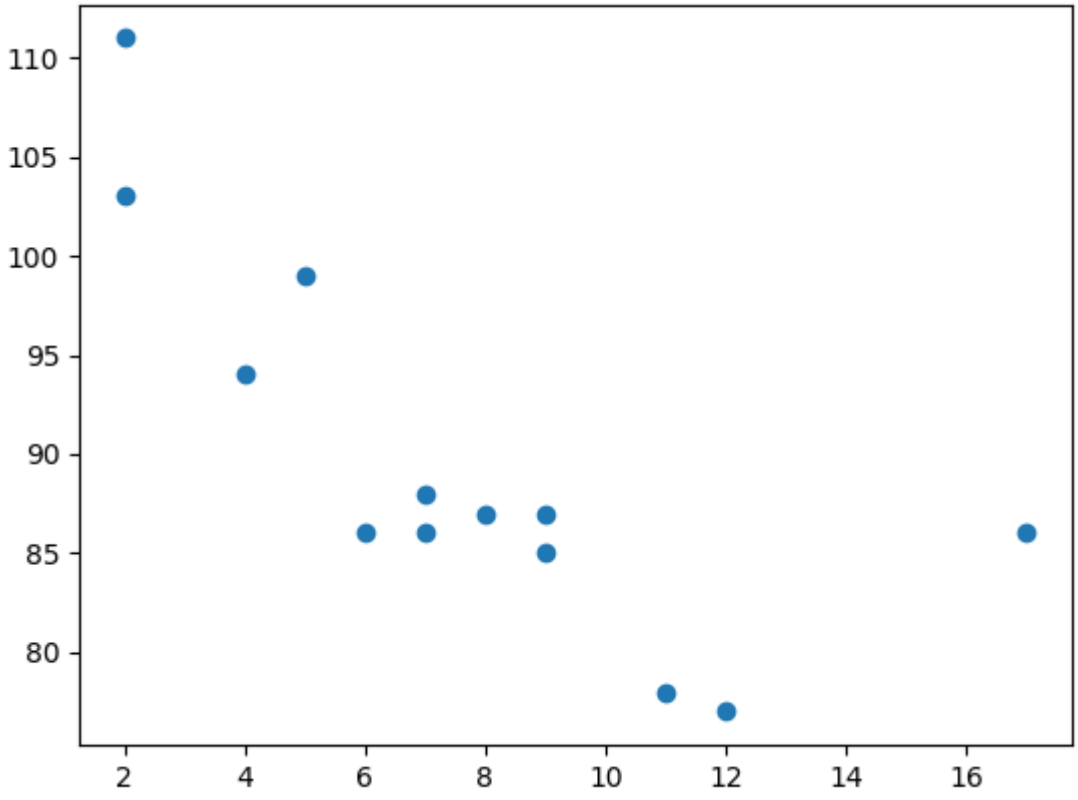


5. SCATTER PLOT

PROGRAM

```
import matplotlib.pyplot as plt

x = [5,7,8,7,2,17,2,9,4,11,12,9,6]
y = [99,86,87,88,111,86,103,87,94,78,77,85,86]

plt.scatter(x, y)
plt.show()
```

OUTPUT

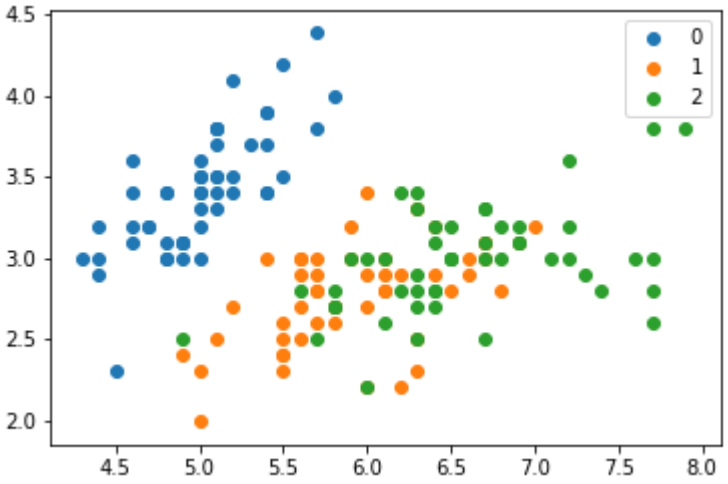


6. MULTIPLE SCATTER

PROGRAM

```
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
feats = load_iris()['data']
target = load_iris()['target']

f, ax = plt.subplots(1)
for i in np.unique(target):
    mask = target == i
    plt.scatter(feats[mask, 0], feats[mask, 1], label=i)
ax.legend()
```

OUTPUT

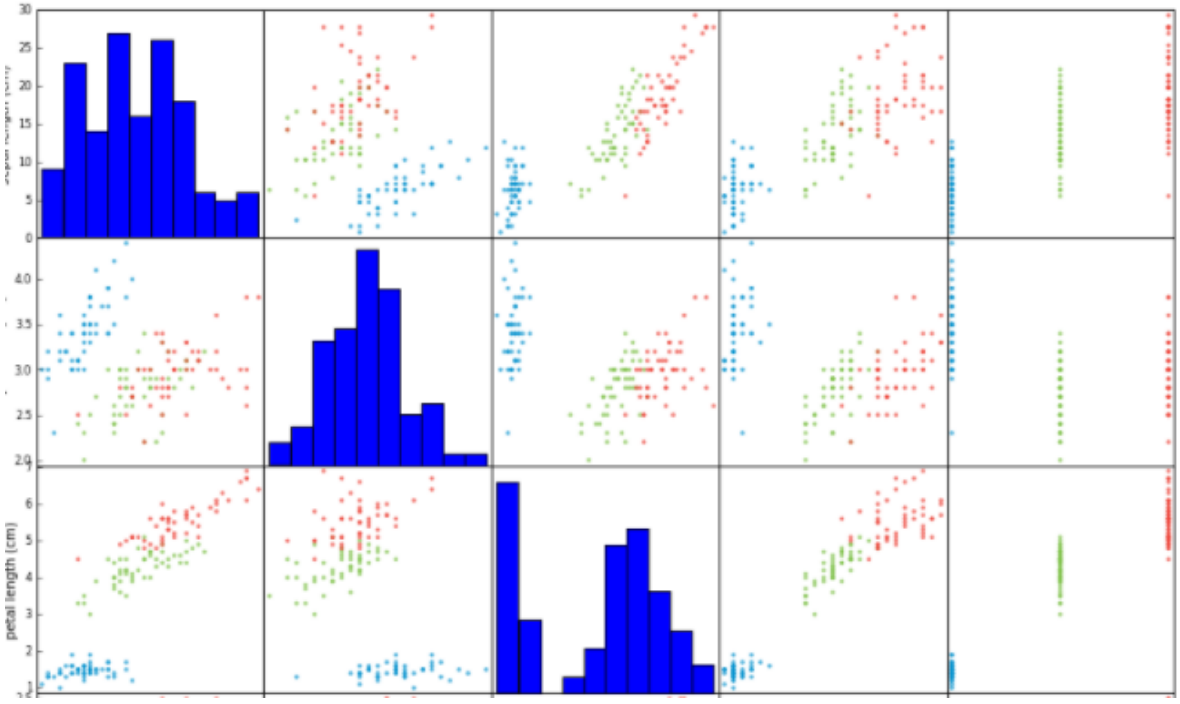


7. MULTIPLE SCATTER MATRIX

PROGRAM

```
from pandas.tools.plotting import scatter_matrix
import pandas as pd
from sklearn import datasets

iris = datasets.load_iris()
iris_data = pd.DataFrame(data=iris['data'],columns=iris['feature_names'])
iris_data["target"] = iris['target']

color_wheel = {1: "#0392cf",
               2: "#7bc043",
               3: "#ee4035"}
colors = iris_data["target"].map(lambda x: color_wheel.get(x + 1))
ax = scatter_matrix(iris_data, color=colors, alpha=0.6, figsize=(15, 15),
diagonal='hist')
```

OUTPUT

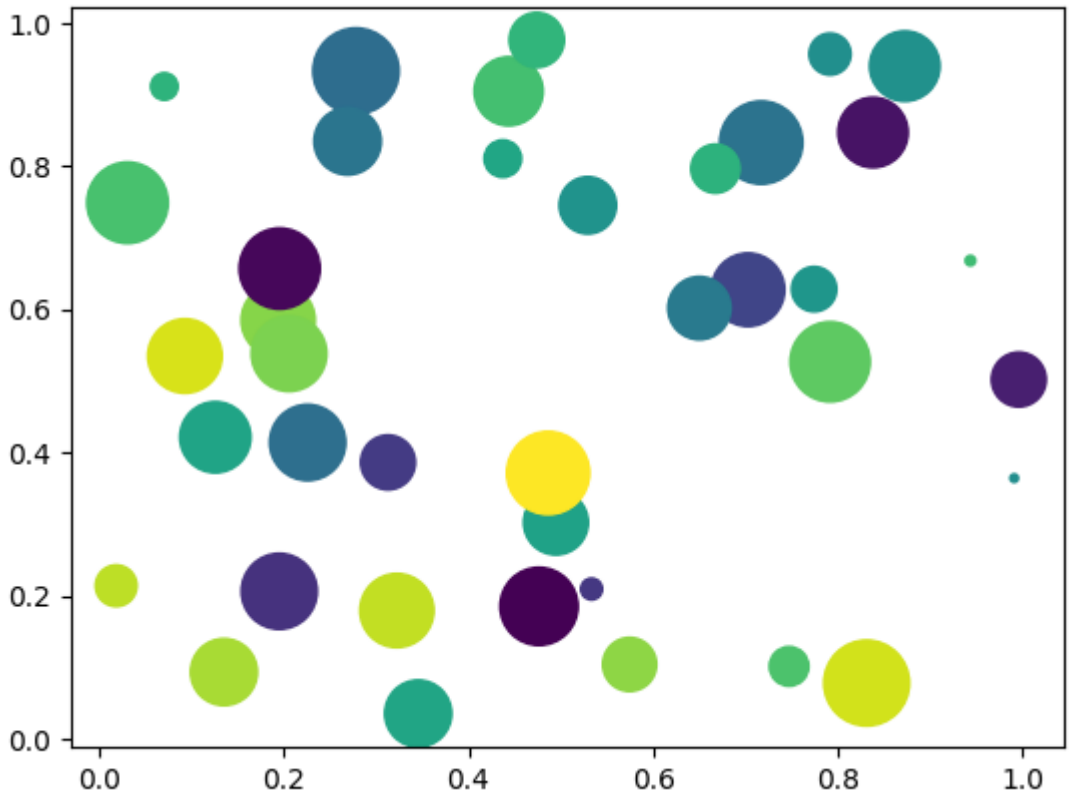


8. BUBBLE PLOT

PROGRAM

```
import matplotlib.pyplot as plt
import numpy as np

# create data
x = np.random.rand(40)
y = np.random.rand(40)
z = np.random.rand(40)
colors = np.random.rand(40)
# use the scatter function
plt.scatter(x, y, s=z * 1000, c=colors)
plt.show()
```

OUTPUT

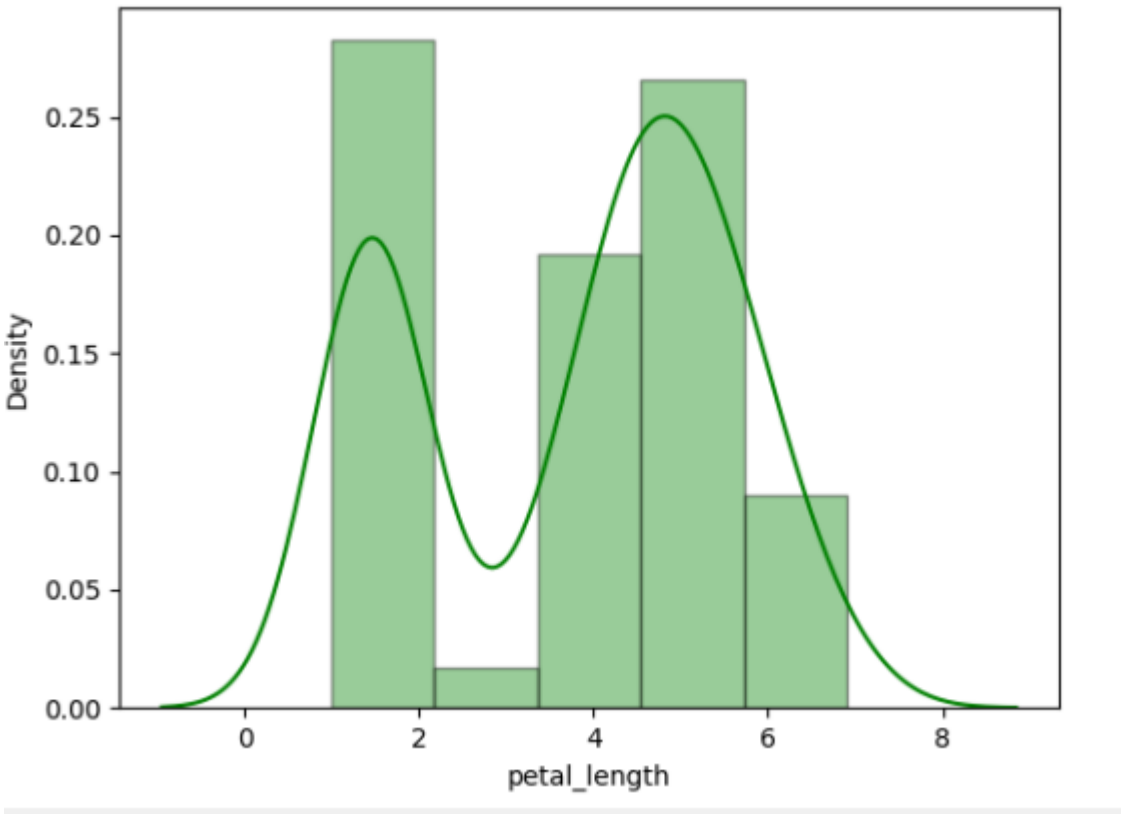


9. DENSITY CHART

PROGRAM

```
#importing various libraries
import seaborn as sns
import matplotlib..pyplot as plt
#importing iris dataset from the library
df2=sns.load dataset('iris')
#plotting histogram and density plot for
#petal length using displot() by setting color
sns.distplot(a=df2.petal_length,color='green',
                hist_kws={"edgecolor":'black'})
#visualizing plot using matplotlib.pyplot library
plt.show()
```

OUTPUT

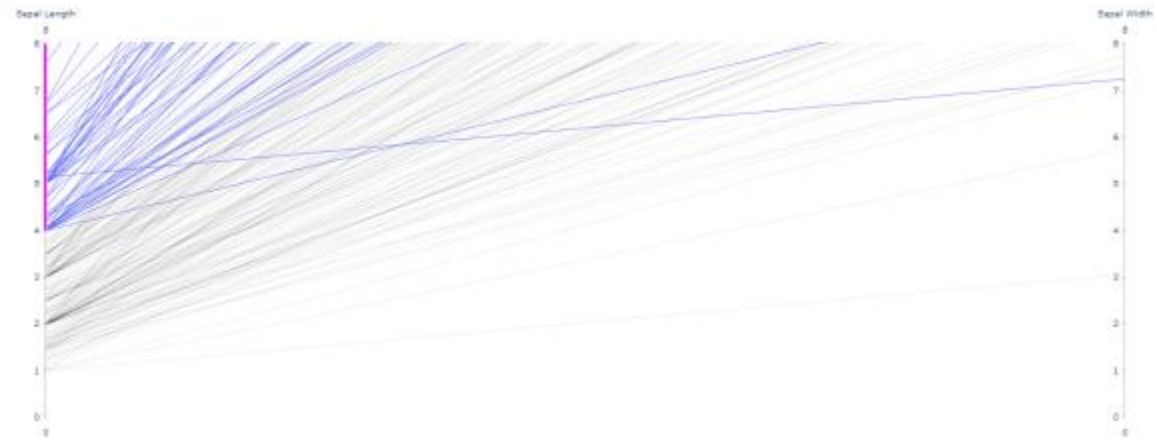


10. PARALLEL PLOT

PROGRAM

```
# importing various package
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# making data frame from csv file
df = pd.read_csv(
    'https://raw.github.com/pandas-dev/'
    'pandas/master/pandas/tests/io/data/csv/iris.csv'
)

# Creating Andrews curves
x = pd.plotting.andrews_curves(df, 'Name')

# ploting the Curve
x.plot()

# Display
plt.show()
```

OUTPUT



11. ANDREWS CURVE

PROGRAM

```
# importing various package
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# making data frame from csv file
df = pd.read_csv(
    'https://raw.github.com/pandas-dev/'
    'pandas/master/pandas/tests/io/data/csv/iris.csv'
)

# Creating Andrews curves
x = pd.plotting.andrews_curves(df, 'Name')

# ploting the Curve
x.plot()

# Display
plt.show()
```

OUTPUT