**CYCLE 3**

1. **Sarah bought a new car in 2001 for $24,000. The dollar value of her car changed each year as shown in the table below.**

**Value of Sarah’s Car**

**Year Value**

**2001 $24,000**

**2002 $22,500**

**2003 $19,700**

**2004 $17,500**

**2005 $14,500**

**2006 $10,000**

**2007 $ 5,800**

**Represent the following information using a line graph with following style properties**

**X- axis - Year**

**Y –axis - Car Value**

**title –Value Depreciation (left Aligned)**

**Line Style dash dot and Line-color should be red**

**point using \* symbol with green color and size 20**

**Code:**

import matplotlib.pyplot as plt

import numpy as np

# Data for x and y axis

year = np.array([2001, 2002, 2003, 2004, 2005, 2006, 2007])

value = np.array([24000, 22500, 19700, 17500, 14500, 10000, 5800])

#define graph characteristics

plt.plot(year, value, marker = "\*", linestyle='-.', color='red', markersize=20, markerfacecolor='green')

#name axis

plt.title("Value Depreciation", loc='left')

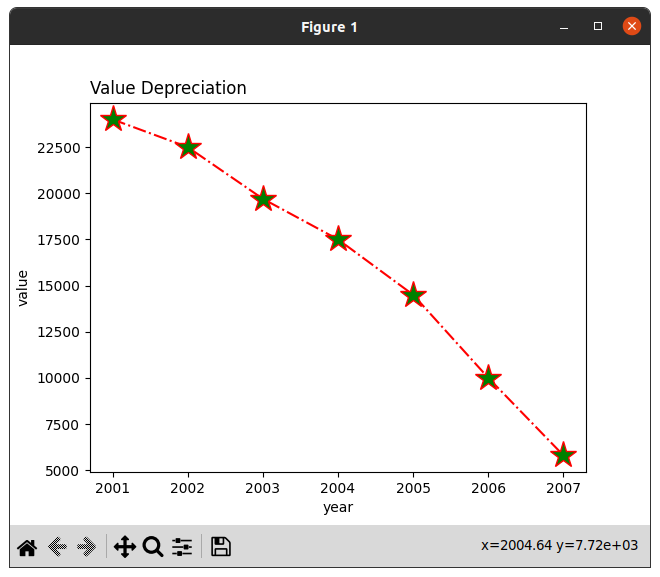
plt.xlabel("year")

plt.ylabel("value")

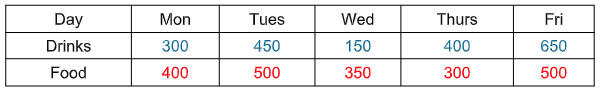
#show graph

plt.show()

**Output:**



1. **Following table gives the daily sales of the following items in a shop**



**Use subplot function to draw the line graphs with grids(color as blue and line style dotted) for the above information as 2 separate graphs in two rows**

**a) Properties for the Graph 1:**

**X label- Days of week**

**Y label-Sale of Drinks**

**Title-Sales Data1 (right aligned)**

**Line –dotted with cyan color**

**Points- hexagon shape with color magenta and outline black**

**b) Properties for the Graph 2:**

**X label- Days of Week**

**Y label-Sale of Food**

**Title-Sales Data2 ( center aligned)**

**Line –dashed with yellow color**

**Points- diamond shape with color green and outline red**

**Code:**

import matplotlib.pyplot as plt

import numpy as np

day = ["Mon", "Tue", "Wed", "Thurs", "Fri"]

drinks = np.array([300, 450, 150, 400, 650])

food = np.array([400, 500, 350, 300, 500])

fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(8, 8))

#plot1

ax1.plot(day, drinks, linestyle='--', color='cyan', marker='h', markersize=10, markeredgecolor='black')

ax1.set\_title("sales data1", loc='right')

ax1.set\_xlabel("days of week")

ax1.set\_ylabel("sale of drinks")

#plot2

ax2.plot(day, food, linestyle='--', color='yellow', marker='d', markersize=10, markeredgecolor='red')

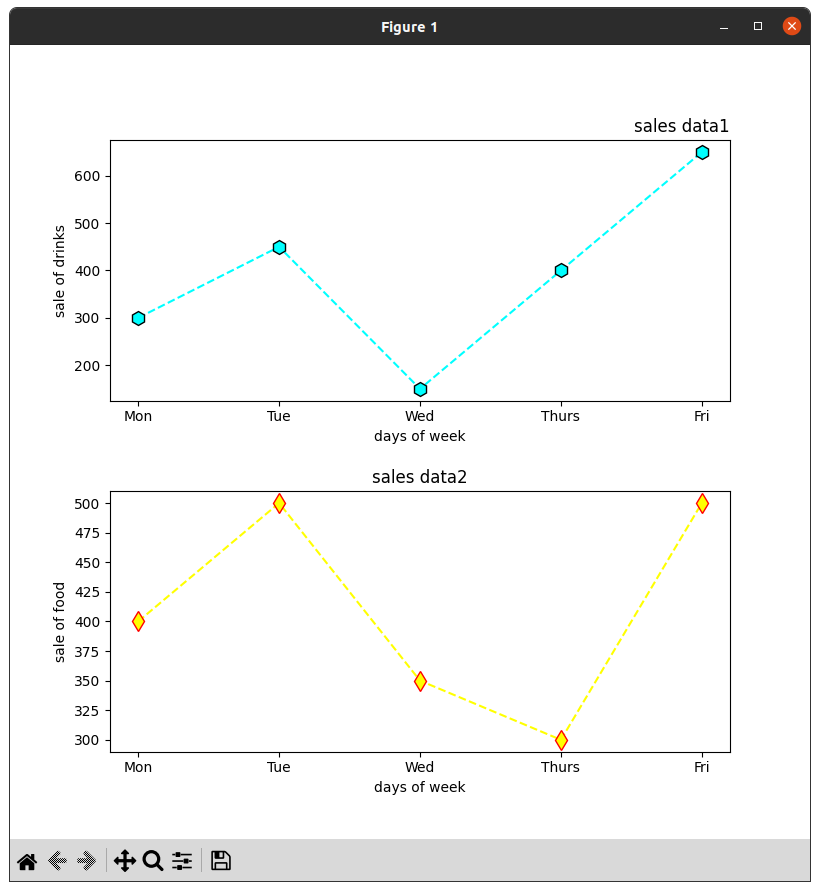
ax2.set\_title("sales data2", loc='center')

ax2.set\_xlabel("days of week")

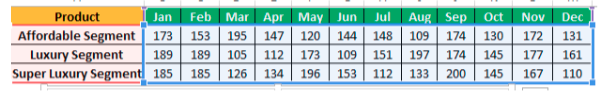
ax2.set\_ylabel("sale of food")

plt.show()

**Output:**



1. **Create scatter plot for the below data:(use Scatter function)**



**Create scatter plot for each Segment with following properties within one graph**

**X Label- Months of Year with font size 18**

**Y-Label- Sales of Segments**

**Title –Sales Data**

**Color for Affordable segment- pink**

**Color for Luxury Segment- Yellow**

**Color for Super luxury segment-blue**

**Code:**

import matplotlib.pyplot as plt

import numpy as np

#data

Months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

affordable\_segment = [173, 153, 195, 147, 120, 144, 148, 109, 174, 130, 172, 131]

luxury\_segment = [189, 189, 105, 112, 173, 109, 151, 197, 174, 145, 177, 161]

super\_luxury\_segment = [185, 185, 126, 134, 196, 153, 112, 133, 200, 145, 167, 110]

plt.title("Sales Data")

plt.xlabel("months of year", fontsize=18)

plt.ylabel("Sales of Segments")

#plots for different segments

plt.scatter(Months, luxury\_segment, color='yellow', label='Luxury', s=100)

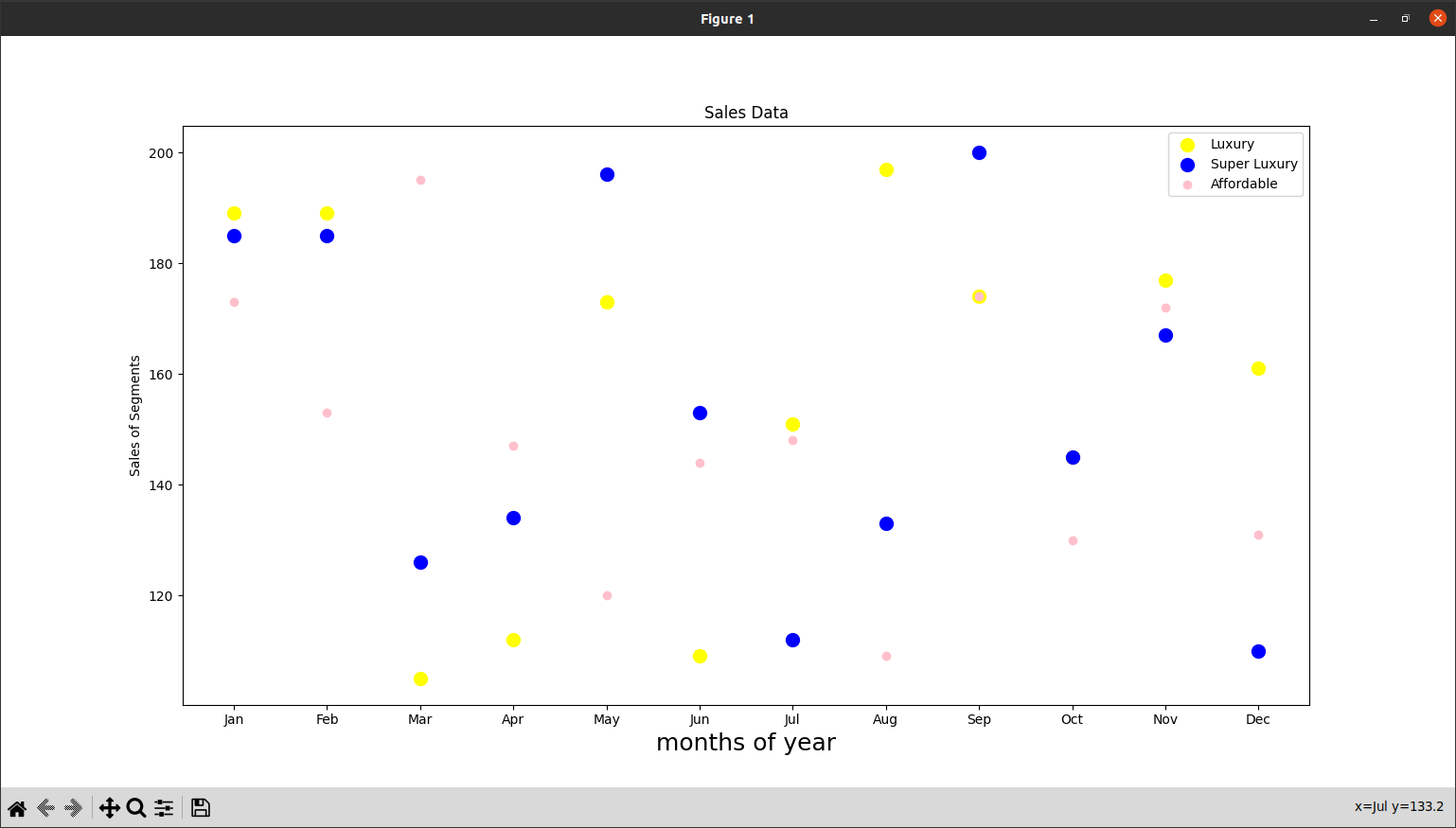
plt.scatter(Months, super\_luxury\_segment, color='blue', label='Super Luxury', s=100)

plt.scatter(Months, affordable\_segment, color='pink', label='Affordable')

plt.legend()

plt.show()

**Output:**



1. **Display the above data using multiline plot( 3 different lines in same graph)**

**Display the description of the graph in upper right corner(use legend())**

**Use different colors and line styles for 3 different lines**

**Code:**

import matplotlib.pyplot as plt

import numpy as np

#data

Months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']

affordable\_segment = [173, 153, 195, 147, 120, 144, 148, 109, 174, 130, 172, 131]

luxury\_segment = [189, 189, 105, 112, 173, 109, 151, 197, 174, 145, 177, 161]

super\_luxury\_segment = [185, 185, 126, 134, 196, 153, 112, 133, 200, 145, 167, 110]

plt.title("Sales Data")

plt.xlabel("months of year", fontsize=18)

plt.ylabel("Sales of Segments")

plt.plot(Months, luxury\_segment, color='blue', label='Luxury', linestyle='-', marker='o')

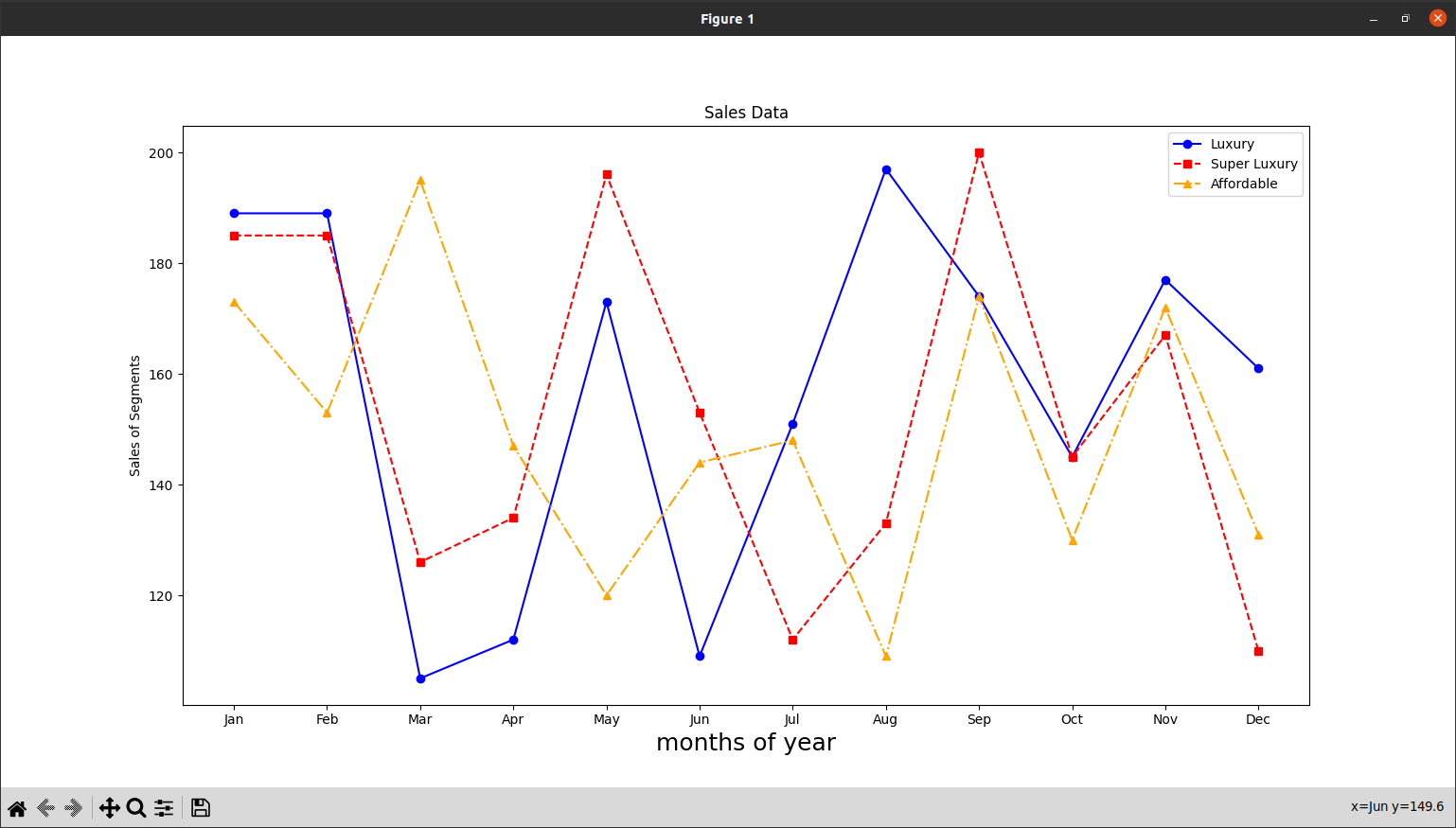
plt.plot(Months, super\_luxury\_segment, color='red', label='Super Luxury', linestyle='--', marker='s')

plt.plot(Months, affordable\_segment, color='orange', label='Affordable', linestyle='-.', marker='^')

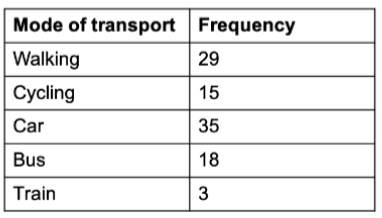
plt.legend(loc='upper right')

plt.show()

**Output:**



1. **100 students were asked what their primary mode of transport for getting to school was. The results of this survey are recorded in the table below. Construct a bar graph representing this information.**



**Create a bar graph with:**

**X axis -mode of Transport and Y axis ‘frequency’**

**Provide appropriate labels and title**

**Width .1, color green**

**Code:**

import matplotlib.pyplot as plt

#data

modes\_of\_transport = ["Walking", "Cycling", "Car", "Bus", "Train"]

frequency = [29, 15, 35, 18, 3]

plt.xlabel('Mode of Transport')

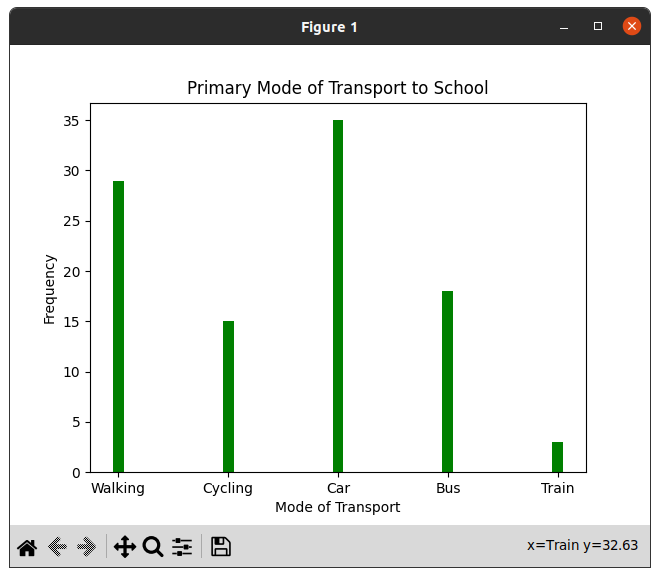
plt.ylabel('Frequency')

plt.title('Primary Mode of Transport to School')

plt.bar(modes\_of\_transport, frequency, width=0.1, color='green')

plt.show()

**Output:**



1. **We are provided with the height of 30 cherry trees. The height of the trees (in inches): 61, 63, 64, 66, 68, 69, 71, 71.5, 72, 72.5, 73, 73.5, 74, 74.5, 76, 76.2, 76.5, 77, 77.5, 78, 78.5, 79, 79.2, 80, 81, 82, 83, 84, 85, 87.Create a histogram with a bin size of 5**

**Code:**

import matplotlib.pyplot as plt

#data

heights = [61, 63, 64, 66, 68, 69, 71, 71.5, 72, 72.5, 73, 73.5, 74, 74.5, 76, 76.2,

76.5, 77, 77.5, 78, 78.5, 79, 79.2, 80, 81, 82, 83, 84, 85, 87]

#histogram defining

plt.hist(heights, bins=5, edgecolor='black', color='skyblue')

#labels

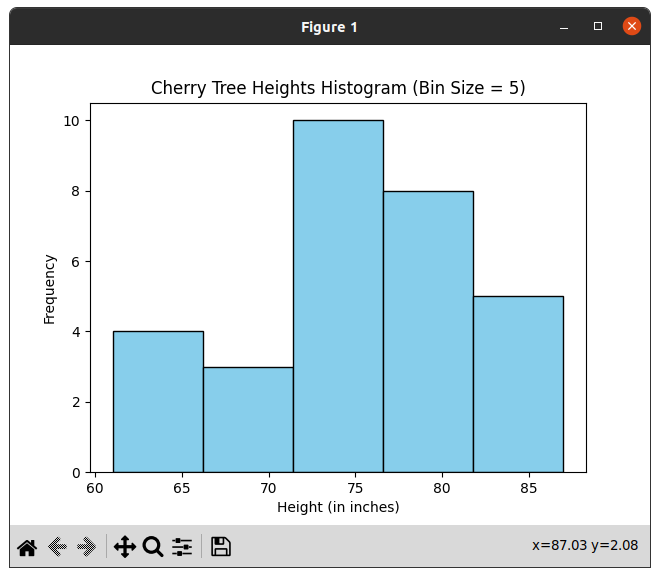
plt.xlabel('Height (in inches)')

plt.ylabel('Frequency')

plt.title('Cherry Tree Heights Histogram (Bin Size = 5)')

plt.show()

**Output:**



1. **Using the pandas function read\_csv(), read the given ‘iris’ data set.**

**i) Display Shape of the data set.**

**ii) First 5 and last five rows of data set(head and tail).**

**iii) Size of dataset.**

**iv) No. of samples available for each variety.**

**v) Description of the data set( use describe ).**

**Code:**

import pandas as pd

print("----------------------\n Reg.No: SJC22MCA-2022\n Name: Denzel Sunny\n Batch: S3 MCA\n ----------------------\n")

df = pd.read\_csv('iris.csv')

#shape of data

print("Shape of data: ", df.shape)

#first 5 rows of the data

print("\nfirst 5: \n", df.head())

#last 5 rows of the data

print("\nlast 5: \n", df.tail())

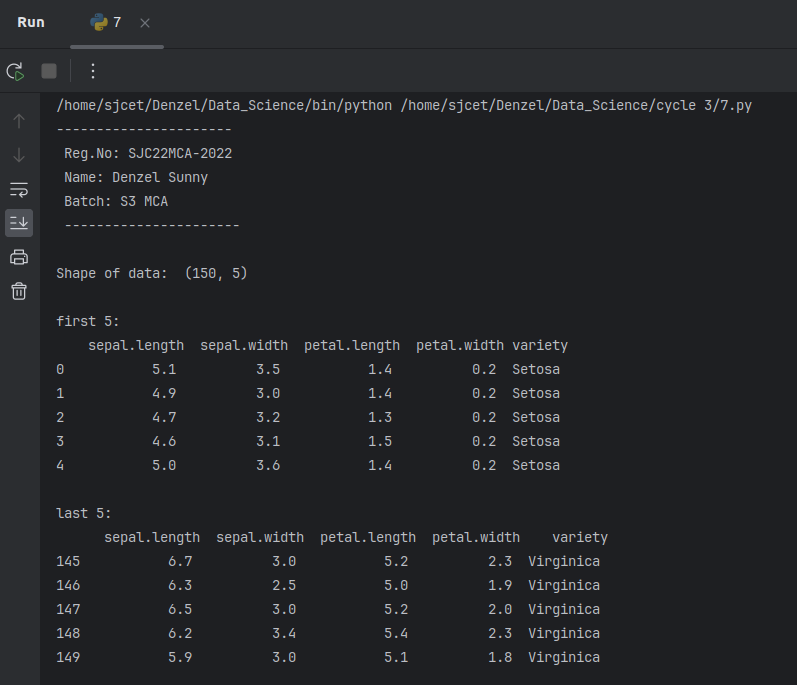
#description

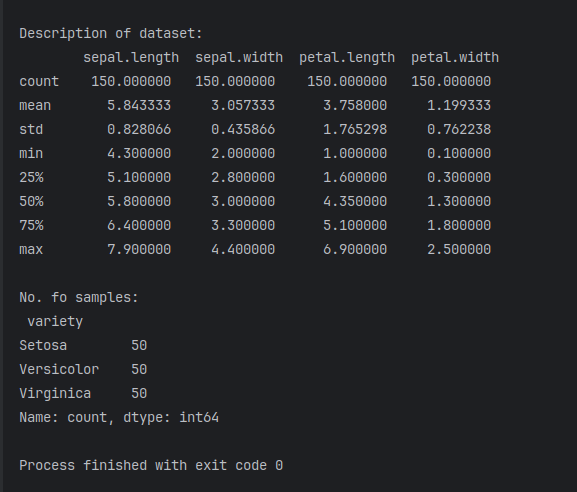
print("\nDescription of dataset: \n", df.describe())

#count no of sample for each variety

print("\nNo. fo samples: \n", df['variety'].value\_counts())

**Output:**

****

****

1. **Use pairplot() function in seaborn to display pairwise relationships between attributes. Try different kind of plots {‘scatter’, ‘kde’, ‘hist’, ‘reg’} and different kind of markers.**

**Code:**

import seaborn as sns

import matplotlib as mpl

import matplotlib.pyplot as plt

import pandas as pd

iris = pd.read\_csv("iris.csv")

pairplot = sns.pairplot(iris, kind="scatter", markers=["o"])

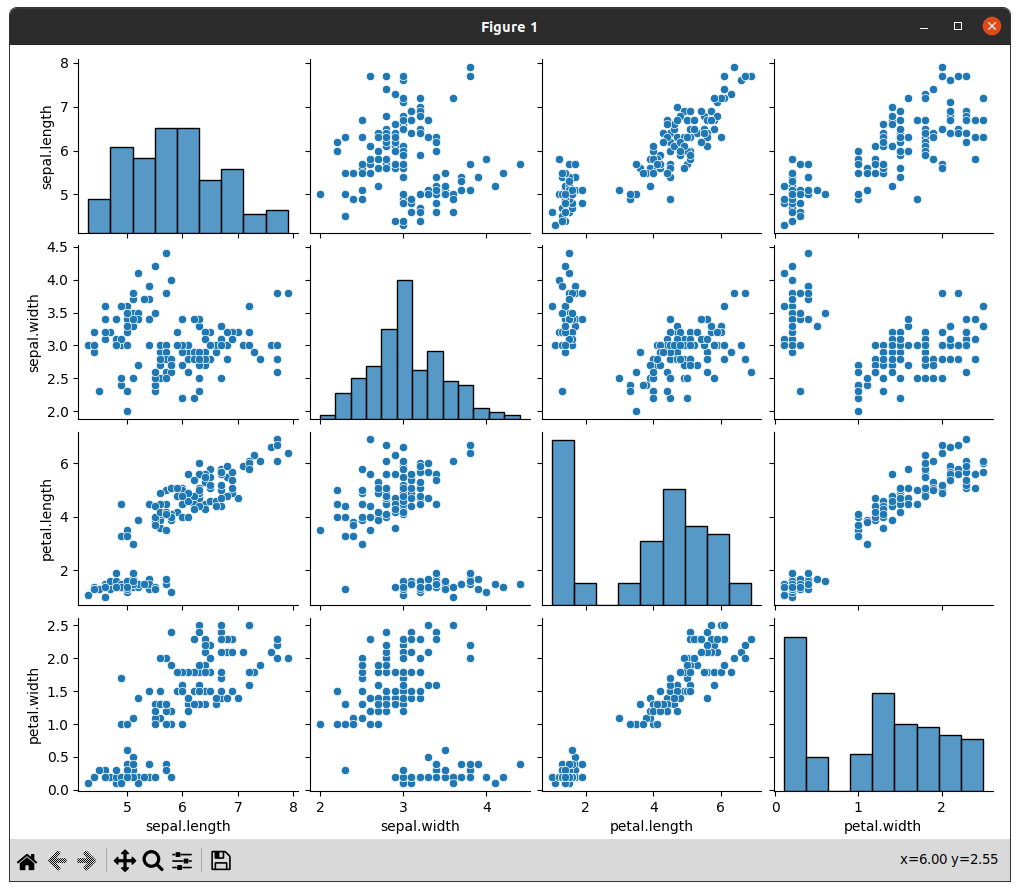
#pairplot = sns.pairplot(iris, kind="kde", markers=["s"])

#pairplot = sns.pairplot(iris, kind="reg", markers=["D"])

#pairplot = sns.pairplot(iris, kind="hist")

plt.show()

**Output:**



1. **Using the iris data set,get familiarize with functions:**

**1) displot()**

**2) histplot()**

**3) relplot()**

**Code:**

import seaborn as sns

import matplotlib.pyplot as plt

iris = sns.load\_dataset("iris")

#sns.displot(data=iris, x="sepal\_length", kde=True)

#sns.relplot(data=iris, x="sepal\_length", y="petal\_length", hue="species")

sns.histplot(data=iris, x="petal\_length", bins=10, kde=True)

plt.show()

**Output:**

