**Experiment Number: 01**

**Problem Statement:** Write Python code to find the statistical properties like- Mean, Median, Mode, Standard Deviation, Variance, Percentile, Data Distribution, Histogram, Big Data Distribution, Normal Data Distribution, Scatter Plot, Random Data Distributions etc.

**NAME:** Harshwardhan Patil  **ROLLNO:** 50

**CLASS:** TY-IT-B **BATCH:** B1

**DATE OF PERFORMANCE: 06/01/2025**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

PROGARM:  
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

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

mean = iris['sepal\_length'].mean()

median = iris['sepal\_length'].median()

mode = iris['sepal\_length'].mode()[0]

std\_dev = iris['sepal\_length'].std()

variance = iris['sepal\_length'].var()

percentile\_25 = np.percentile(iris['sepal\_length'], 25)

percentile\_75 = np.percentile(iris['sepal\_length'], 75)

print("Statistical Properties of 'sepal\_length':")

print(f"Mean: {mean}")

print(f"Median: {median}")

print(f"Mode: {mode}")

print(f"Standard Deviation: {std\_dev}")

print(f"Variance: {variance}")

print(f"25th Percentile: {percentile\_25}")

print(f"75th Percentile: {percentile\_75}\n")

fig, axes = plt.subplots(2, 3, figsize=(12, 8))

fig.suptitle("Statistical Analysis and Data Visualization (Iris Dataset)", fontsize=16)

sns.kdeplot(iris['sepal\_length'], shade=True, ax=axes[0, 0])

axes[0, 0].set\_title("Data Distribution of Sepal Length")

axes[0, 0].set\_xlabel("Sepal Length")

axes[0, 0].set\_ylabel("Density")

axes[0, 1].hist(iris['sepal\_width'], bins=20, edgecolor='k')

axes[0, 1].set\_title("Histogram of Sepal Width")

axes[0, 1].set\_xlabel("Sepal Width")

axes[0, 1].set\_ylabel("Frequency")

sns.histplot(iris['sepal\_length'], bins=30, kde=True, ax=axes[0, 2])

axes[0, 2].set\_title("Big Data Distribution: Sepal Length")

axes[0, 2].set\_xlabel("Sepal Length")

axes[0, 2].set\_ylabel("Frequency")

sns.histplot(iris['petal\_width'], bins=20, kde=True, ax=axes[1, 0])

axes[1, 0].set\_title("Normal Data Distribution: Petal Width")

axes[1, 0].set\_xlabel("Petal Width")

axes[1, 0].set\_ylabel("Frequency")

sns.scatterplot(x='sepal\_length', y='sepal\_width', data=iris, hue='species', ax=axes[1, 1])

axes[1, 1].set\_title("Scatter Plot: Sepal Length vs Sepal Width")

axes[1, 1].set\_xlabel("Sepal Length")

axes[1, 1].set\_ylabel("Sepal Width")

axes[1, 1].legend(title='Species')

axes[1, 2].hist(iris['petal\_length'], bins=10, edgecolor='k')

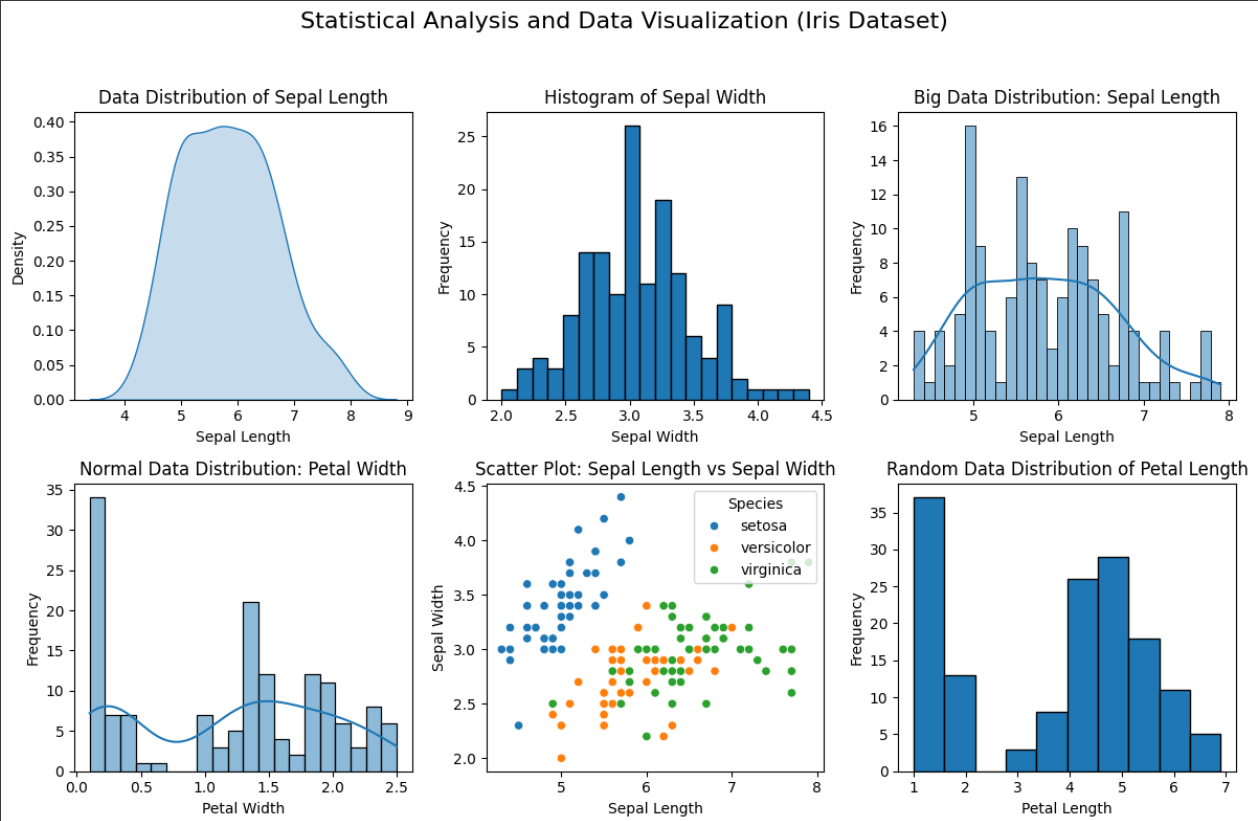
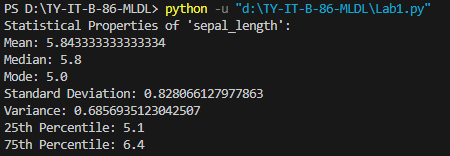
axes[1, 2].set\_title("Random Data Distribution of Petal Length")

axes[1, 2].set\_xlabel("Petal Length")

axes[1, 2].set\_ylabel("Frequency")

plt.tight\_layout(rect=[0, 0, 1, 0.95])

plt.show()

**OUTPUT:** **Part 2:**import numpy as np

import matplotlib.pyplot as plt

from scipy import stats

import seaborn as sns

# Sample sentences

sentences = [

"The quick brown fox jumps over the lazy dog.",

"A journey of a thousand miles begins with a single step.",

"To be or not to be, that is the question.",

"All that glitters is not gold.",

"I think, therefore I am.",

"In the middle of difficulty lies opportunity.",

"Success is not final; failure is not fatal: It is the courage to continue that counts."

]

# Function to calculate sentence statistics

def get\_sentence\_statistics(sentences):

sentence\_lengths = [len(sentence.split()) for sentence in sentences] # Number of words per sentence

mean\_length = np.mean(sentence\_lengths)

median\_length = np.median(sentence\_lengths)

# Check if mode is a scalar or an array

mode\_result = stats.mode(sentence\_lengths)

mode\_length = mode\_result.mode[0] if isinstance(mode\_result.mode, np.ndarray) else mode\_result.mode # If it's a scalar, assign it directly; otherwise, access the first element

std\_dev = np.std(sentence\_lengths)

variance = np.var(sentence\_lengths)

percentile\_25 = np.percentile(sentence\_lengths, 25)

percentile\_75 = np.percentile(sentence\_lengths, 75)

print(f"Mean sentence length: {mean\_length}")

print(f"Median sentence length: {median\_length}")

print(f"Mode sentence length: {mode\_length}")

print(f"Standard Deviation: {std\_dev}")

print(f"Variance: {variance}")

print(f"25th Percentile: {percentile\_25}")

print(f"75th Percentile: {percentile\_75}")

return sentence\_lengths

# Get statistics

sentence\_lengths = get\_sentence\_statistics(sentences)

# Data Distribution Plot

plt.figure(figsize=(10, 5))

sns.histplot(sentence\_lengths, kde=True, color='skyblue', bins=6)

plt.title("Histogram and Data Distribution of Sentence Lengths")

plt.xlabel("Number of Words")

plt.ylabel("Frequency")

plt.show()

# Big Data Distribution

large\_sentence\_lengths = np.random.choice(sentence\_lengths, 1000, replace=True) # Simulate big data

plt.figure(figsize=(10, 5))

sns.histplot(large\_sentence\_lengths, kde=True, color='orange')

plt.title("Big Data Distribution (Simulated)")

plt.xlabel("Number of Words")

plt.ylabel("Frequency")

plt.show()

# Normal Data Distribution

normal\_dist = np.random.normal(loc=np.mean(sentence\_lengths), scale=np.std(sentence\_lengths), size=1000)

plt.figure(figsize=(10, 5))

sns.histplot(normal\_dist, kde=True, color='green')

plt.title("Normal Data Distribution")

plt.xlabel("Number of Words")

plt.ylabel("Frequency")

plt.show()

# Scatter Plot

plt.figure(figsize=(10, 5))

plt.scatter(range(len(sentence\_lengths)), sentence\_lengths, color='red')

plt.title("Scatter Plot of Sentence Lengths")

plt.xlabel("Sentence Index")

plt.ylabel("Number of Words")

plt.show()

# Random Data Distributions

random\_data = np.random.uniform(low=min(sentence\_lengths), high=max(sentence\_lengths), size=1000)

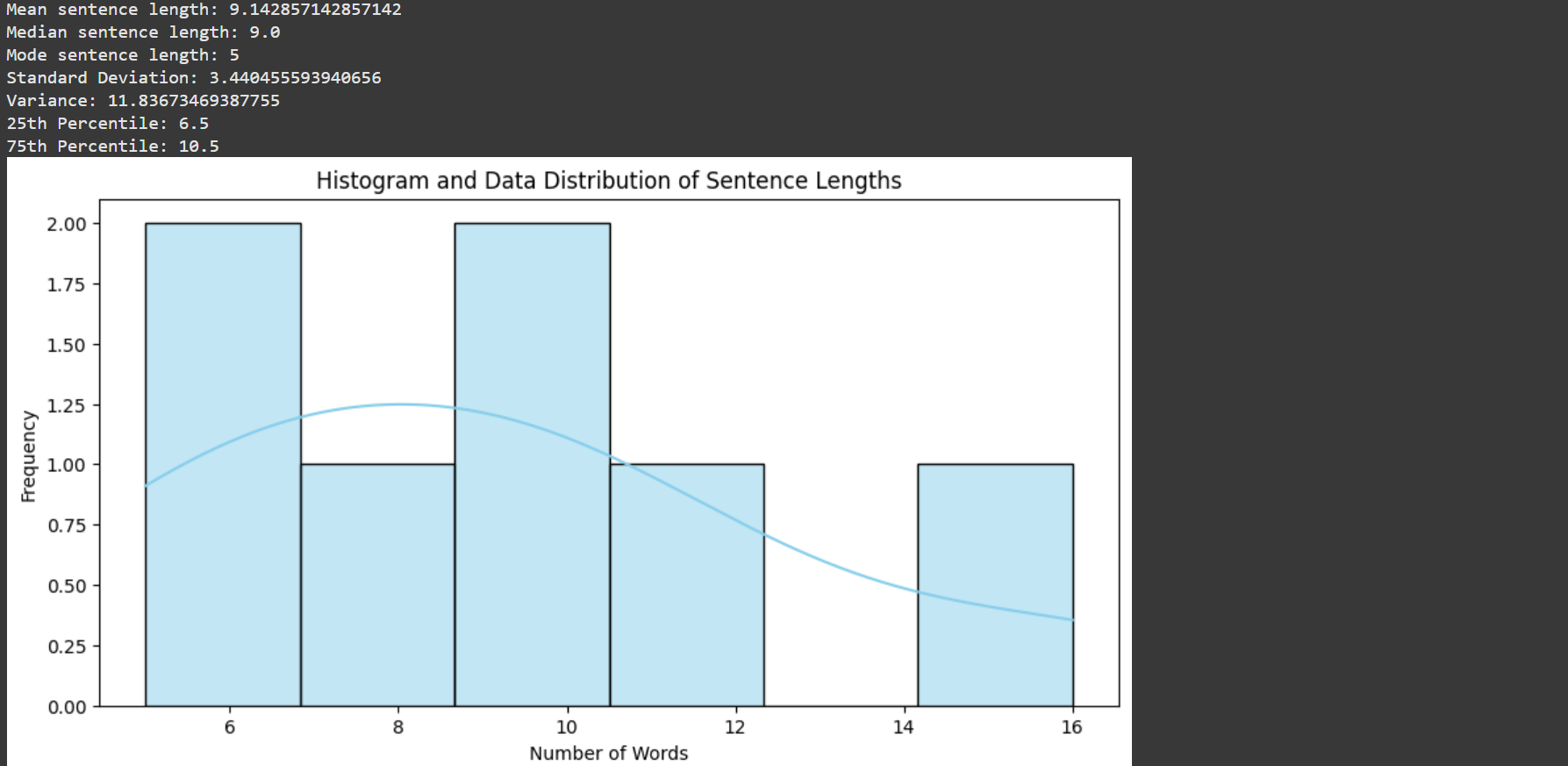
plt.figure(figsize=(10, 5))

sns.histplot(random\_data, kde=True, color='purple')

plt.title("Random Data Distribution")

plt.xlabel("Number of Words")

plt.ylabel("Frequency")

plt.show()  
  
  
A graph with orange lines and numbers

AI-generated content may be incorrect.  
A green and black graph

AI-generated content may be incorrect.  
A graph with red dots

AI-generated content may be incorrect.  
A graph with purple lines

AI-generated content may be incorrect.