INSIGHTS FROM MULTIVARIATE DATA ANALYSIS: A CASE STUDY USING THE IRIS DATASET

SUBMITTED BY :

YUKTHA K IYER

II MSC DSA (A) – 24MSRDS065

TABLE OF CONTENTS

1. INTRODUCTION

Multivariate Data Analysis looks at more than two variables at the same time to understand the relationships, patterns, and structures in a dataset. Unlike univariate and bivariate analyses, multivariate methods offer greater insights by examining the combined effects of several variables. In this activity, we will use the Iris dataset to show different multivariate analysis techniques. These techniques include descriptive statistics, covariance and correlation analysis, canonical correlation analysis, outlier detection, assumption testing, and advanced data visualizations.

2. DATASET DESCRIPTION

The Iris dataset is a well-known multivariate dataset containing 150 observations of iris flowers. Each observation has four numerical attributes and one categorical attribute. Variables Sepal Length Sepal Width Petal Length Petal Width Species (Setosa, Versicolor, Virginica) This dataset is frequently used to demonstrate statistical and machine learning methods since it clearly separates classes.

3. MEAN VECTOR

The mean vector shows the average value for each variable in the dataset. It indicates the central tendency in multivariate data and serves as a reference for distance-based evaluations. The calculated mean vector shows that petal characteristics vary more than sepal characteristics. This suggests that petal features are essential for telling species apart.

4. COVARIANCE MATRIX

The covariance matrix measures how pairs of variables change together.

Interpretation:

- ✓ Positive covariance means the variables increase together.
- ✓ Larger covariance values suggest stronger linear relationships.
- ✓ Petal Length and Petal Width show strong covariance, indicating a high degree of linear dependence.

5. CORRELATION MATRIX

The correlation matrix converts covariance values to a range from -1 to +1.

Important Insights:

- ✓ There is a significant positive correlation between Petal Length and Petal Width.
- ✓ A moderate correlation exists between Sepal Length and Petal Length.
- ✓ The correlations involving Sepal Width are weak. Correlation heatmaps effectively illustrate these relationships.

6. GRAPHICAL VISUALIZATIONS

6.1 Histograms

- ✓ Histograms were used to show the distribution of each variable.
- ✓ Petal features have clear multi-modal distributions that indicate species separation.

6.2 Boxplots

- ✓ Boxplots help find outliers and compare distributions.
- ✓ Petal measurements show distinct medians for different species.

6.3 Pair Plot

- ✓ Pairwise scatter plots with class labels show clear clustering among species, especially for petal variables.

7. CANONICAL CORRELATION ANALYSIS (CCA)

A Canonical Correlation Analysis was performed involving:

- ✓ Group 1: Sepal Length and Sepal Width.
- ✓ Group 2: Petal Length and Petal Width.

The results indicated a notable canonical correlation coefficient, highlighting a robust connection between sepal and petal measurements. CCA aids in comprehending the relationship between one set of variables and another within a multivariate context.

8. HANDLING MISSING DATA

The missing values created for demonstration purposes showed how to handle absent data. The techniques used included mean imputation, which replaces missing values with the column's average, and row deletion, which removes rows with missing entries. Mean imputation preserves the overall size of the dataset. Row deletion keeps the integrity of the data but reduces the sample size.

9. MULTIVARIATE OUTLIER DETECTION

The Mahalanobis Distance identified multivariate outliers.

Approach:

- ✓ The calculation of the distance was performed using the mean vector alongside the covariance matrix.
- ✓ An outlier detection threshold was established based on Chi-square values.

Findings: Only a small number of entries surpassed the threshold, suggesting that the dataset contains very few multivariate outliers.

10. ASSUMPTION TESTING

Shapiro-Wilk Test : This test assesses the normality of individual variables.

Note: Certain variables exhibit departures from normality, which is typical in actual datasets and supports the use of analysis based on visualization.

11. PRINCIPAL COMPONENT ANALYSIS (PCA)

PCA was utilized to lower the dimensionality from four variables to two primary components. Results:

- ✓ The initial two components account for the majority of the variance.
- ✓ PCA scatter plots indicate distinct separation between the species.
- ✓ PCA verifies that petal measurements are the main factors contributing to variance.

12. ADVANCED MULTIVARIATE VISUALIZATIONS

To improve understanding and visual appeal, various sophisticated visualizations were employed:

- ✓ Neon Scatter Plot – Emphasizes the separation of classes.
- ✓ KDE Density Plot – Displays a smooth probability distribution.
- ✓ Violin Plot – Merges distribution and density information.
- ✓ Ridge Plot – Facilitates visual comparisons of feature distributions.
- ✓ 3D Scatter Plot – Offers a visualization of multiple dimensions.
- ✓ Radar Chart – Compares averages across different variables.
- ✓ Hexbin and Contour Plots – Provide insights based on density.

These visualizations deliver both analytical richness and visual clarity.

13. RESULTS AND DISCUSSION

Petal characteristics provide more valuable information compared to sepal characteristics. Significant multivariate correlations are present among petal variables. Using multivariate methods, the species are distinctly separable. Sophisticated visualizations significantly improve the ability to recognize patterns. Integrating statistical metrics with visual analysis results in a strong interpretation.

14. CONCLUSION

This exercise effectively showcased the use of multivariate data analysis methods with the Iris dataset. Various statistical metrics, dimensionality reduction methods, correlation assessments, and sophisticated visualizations were employed to thoroughly investigate and clarify the dataset. Multivariate analysis is a highly effective strategy for comprehending intricate datasets and plays a crucial role in data science applications.

15. TOOLS AND LIBRARIES USED

- ✓ Python
- ✓ NumPy
- ✓ Pandas
- ✓ Matplotlib
- ✓ Seaborn
- ✓ Scikit-learn

16. REFERENCES

1. Johnson, R. A., & Wichern, D. W. (2007). *Applied Multivariate Statistical Analysis*.
2. Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems.
3. Scikit-learn Documentation
4. Seaborn Visualization Library

17. SOURCE CODE

```
# ========================================

# Activity 1: Multivariate Data Analysis

# Complete Python Code

# ========================================


# Import required libraries
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.datasets import load_iris
from sklearn.cross_decomposition import CCA
from scipy.stats import chi2, shapiro

# ----------------------------------------
# 1. Import Multivariate Dataset
# ----------------------------------------

iris = load_iris()
df = pd.DataFrame(iris.data, columns=[
    'Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width'
])
df['Species'] = iris.target

print("\nDataset Head:")
print(df.head())

# ----------------------------------------
# 2. Mean Vector
# ----------------------------------------

mean_vector = df.iloc[:, :4].mean()
print("\nMean Vector:")
print(mean_vector)
```

```
# ----------------------------------------
# 3. Covariance Matrix
# ----------------------------------------

cov_matrix = df.iloc[:, :4].cov()
print("\nCovariance Matrix:")
print(cov_matrix)

# ----------------------------------------
# 4. Correlation Matrix
# ----------------------------------------

corr_matrix = df.iloc[:, :4].corr()
print("\nCorrelation Matrix:")
print(corr_matrix)

# ----------------------------------------
# 5. Graphical Visualizations
# ----------------------------------------

# Histograms
df.iloc[:, :4].hist(figsize=(10, 6))
plt.suptitle("Histograms of Variables")
plt.show()

# Boxplots
df.iloc[:, :4].boxplot(figsize=(10, 6))
plt.title("Boxplots for Outlier Detection")
plt.show()
```

```
# ----------------------------------------
# 6. Matrix Scatter Plot
# ----------------------------------------

sns.pairplot(df, hue="Species")
plt.show()

# ----------------------------------------
# 7. Canonical Correlation Analysis (CCA)
# ----------------------------------------

# Define two variable sets
X = df[['Sepal_Length', 'Sepal_Width']]
Y = df[['Petal_Length', 'Petal_Width']]

cca = CCA(n_components=1)
cca.fit(X, Y)

X_c, Y_c = cca.transform(X, Y)

corr_cca = np.corrcoef(X_c.T, Y_c.T)[0, 1]
print("\nCanonical Correlation Coefficient:", corr_cca)

# ----------------------------------------
# 8. Handling Missing Data
# ----------------------------------------

# Introduce missing values artificially
df_missing = df.copy()
```

```
df_missing.iloc[0:10, 0] = np.nan

print("\nMissing Values Count:")
print(df_missing.isnull().sum())

# Method 1: Mean Imputation
df_mean_imputed = df_missing.fillna(df_missing.mean(numeric_only=True))

# Method 2: Row Deletion
df_row_deleted = df_missing.dropna()

# ----------------------------------------
# 9. Multivariate Outlier Detection
# ----------------------------------------

X_values = df.iloc[:, :4]
mean_vec = X_values.mean()
cov_mat = X_values.cov()
inv_cov_mat = np.linalg.inv(cov_mat)

mahalanobis_dist = []

for i in range(len(X_values)):
    diff = X_values.iloc[i] - mean_vec
    md = np.sqrt(diff.T @ inv_cov_mat @ diff)
    mahalanobis_dist.append(md)
```

```
df['Mahalanobis_Distance'] = mahalanobis_dist

# Threshold using Chi-square distribution
threshold = chi2.ppf(0.975, df=4)

outliers = df[df['Mahalanobis_Distance'] > threshold]

print("\nNumber of Multivariate Outliers:", len(outliers))

# ----------------------------------------
# 10. Assumption Testing
# ----------------------------------------

# Univariate Normality Test (Shapiro-Wilk)
print("\nShapiro-Wilk Normality Test Results:")
for col in df.columns[:4]:
    stat, p = shapiro(df[col])
    print(f"{col}: p-value = {p}")

# ----------------------------------------
# End of Code
# ----------------------------------------




import matplotlib.pyplot as plt
import seaborn as sns
```

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.cross_decomposition import CCA
from scipy.stats import chi2

# Re-initialize variables from the first cell to ensure they are available
iris = load_iris()
df = pd.DataFrame(iris.data, columns=[
    'Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width'
])
df['Species'] = iris.target

corr_matrix = df.iloc[:, :4].corr()

# For CCA plot
X = df[['Sepal_Length', 'Sepal_Width']]
Y = df[['Petal_Length', 'Petal_Width']]
cca = CCA(n_components=1)
ccs = cca.fit(X, Y)
X_c, Y_c = cca.transform(X, Y)

# For Mahalanobis Distance plot
X_values = df.iloc[:, :4]
mean_vec = X_values.mean()
cov_mat = X_values.cov()
inv_cov_mat = np.linalg.inv(cov_mat)

mahalanobis_dist = []
```

```
for i in range(len(X_values)):
    diff = X_values.iloc[i] - mean_vec
    md = np.sqrt(diff.T @ inv_cov_mat @ diff)
    mahalanobis_dist.append(md)

df['Mahalanobis_Distance'] = mahalanobis_dist

threshold = chi2.ppf(0.975, df=4)

# ----------------------------------------
# 11. Additional Multivariate Visualizations
# ----------------------------------------

# 1. Correlation Heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Correlation Heatmap of Iris Variables")
plt.show()

# 2. Species-wise Mean Comparison (Bar Plot)
species_means = df.groupby('Species').mean()

species_means.plot(kind='bar', figsize=(10, 6))
plt.title("Species-wise Mean Comparison")
plt.xlabel("Species")
plt.ylabel("Mean Value")
plt.xticks(rotation=0)
plt.legend(title="Variables")
plt.show()
```

```
# 3. Sepal Length vs Petal Length Scatter Plot
plt.figure(figsize=(8, 6))
sns.scatterplot(
    data=df,
    x="Sepal_Length",
    y="Petal_Length",
    hue="Species",
    style="Species",
    s=70
)
plt.title("Sepal Length vs Petal Length by Species")
plt.show()

# 4. Mahalanobis Distance Plot
plt.figure(figsize=(8, 5))
plt.plot(df['Mahalanobis_Distance'], marker='o', linestyle='')
plt.axhline(y=threshold, color='r', linestyle='--', label='Chi-square Threshold')
plt.title("Mahalanobis Distance Plot")
plt.xlabel("Observation Index")
plt.ylabel("Mahalanobis Distance")
plt.legend()
plt.show()

# 5. Canonical Correlation Scatter Plot
plt.figure(figsize=(7, 5))
plt.scatter(X_c, Y_c)
plt.xlabel("Canonical Variable 1 (X)")
plt.ylabel("Canonical Variable 1 (Y)")
plt.title("Canonical Correlation Analysis Scatter Plot")
```

```
plt.show()
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.decomposition import PCA
from pandas.plotting import parallel_coordinates, andrews_curves

# Re-initialize variables to ensure they are available
iris = load_iris()
df = pd.DataFrame(iris.data, columns=[
    'Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width'
])
df['Species'] = iris.target

# ----------------------------------------
# 12. More Advanced Multivariate Visualizations
# ----------------------------------------

# 1. PCA Scatter Plot (Dimensionality Reduction)
pca = PCA(n_components=2)
pca_components = pca.fit_transform(df.iloc[:, :4])

pca_df = pd.DataFrame(
    pca_components,
    columns=['PC1', 'PC2']
)
pca_df['Species'] = df['Species']
```

```
plt.figure(figsize=(8, 6))
sns.scatterplot(
    data=pca_df,
    x='PC1',
    y='PC2',
    hue='Species',
    style='Species',
    s=70
)
plt.title("PCA Scatter Plot (First Two Principal Components)")
plt.show()

# 2. Violin Plot (Distribution + Density)
plt.figure(figsize=(10, 6))
sns.violinplot(
    data=df.melt(id_vars='Species'),
    x='variable',
    y='value',
    hue='Species',
    split=True
)
plt.title("Violin Plot of Features by Species")
plt.xlabel("Variables")
plt.ylabel("Value")
plt.show()

# 3. Kernel Density Estimation (KDE) Plot
plt.figure(figsize=(8, 6))
for species in df['Species'].unique():
```

```
    sns.kdeplot(
        df[df['Species'] == species]['Petal_Length'],
        label=f"Species {species}",
        fill=True
    )
plt.title("KDE Plot of Petal Length by Species")
plt.xlabel("Petal Length")
plt.legend()
plt.show()

# 4. Parallel Coordinates Plot
plt.figure(figsize=(10, 6))
parallel_coordinates(
    df[['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width', 'Species']],
    class_column='Species',
    colormap='viridis'
)
plt.title("Parallel Coordinates Plot")
plt.show()

# 5. Andrews Curves
plt.figure(figsize=(10, 6))
andrews_curves(
    df[['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width', 'Species']],
    class_column='Species'
)
plt.title("Andrews Curves for Multivariate Visualization")
plt.show()
# ----------------------------------------
```

```
# 13. Aesthetic & Colorful Visualizations
# ----------------------------------------

sns.set(style="whitegrid", palette="Set2")

# 1. Pairplot with KDE Diagonal (More Elegant)
sns.pairplot(
    df,
    hue="Species",
    diag_kind="kde",
    corner=True,
    palette="Set2"
)
plt.suptitle("Enhanced Pairplot with KDE", y=1.02)
plt.show()

# 2. Joint Plot (Petal Length vs Petal Width)
sns.jointplot(
    data=df,
    x="Petal_Length",
    y="Petal_Width",
    hue="Species",
    palette="bright",
    height=7
)
plt.show()

# 3. 3D Scatter Plot (Very Attractive)
from mpl_toolkits.mplot3d import Axes3D
```

```
fig = plt.figure(figsize=(9, 7))
ax = fig.add_subplot(111, projection='3d')

scatter = ax.scatter(
    df['Sepal_Length'],
    df['Petal_Length'],
    df['Petal_Width'],
    c=df['Species'],
    cmap='viridis',
    s=60
)

ax.set_xlabel("Sepal Length")
ax.set_ylabel("Petal Length")
ax.set_zlabel("Petal Width")
ax.set_title("3D Scatter Plot of Iris Dataset")
plt.colorbar(scatter, label="Species")
plt.show()

# 4. Radar Chart (Spider Plot) – Class Profile Visualization
from math import pi

features = ['Sepal_Length', 'Sepal_Width', 'Petal_Length', 'Petal_Width']
angles = [n / float(len(features)) * 2 * pi for n in range(len(features))]
angles += angles[:1]

plt.figure(figsize=(8, 8))
ax = plt.subplot(111, polar=True)
```

```
for species in df['Species'].unique():
    values = df[df['Species'] == species][features].mean().tolist()
    values += values[:1]
    ax.plot(angles, values, label=f"Species {species}", linewidth=2)
    ax.fill(angles, values, alpha=0.25)

ax.set_thetagrids(np.degrees(angles[:-1]), features)
ax.set_title("Radar Chart of Feature Means by Species", y=1.1)
ax.legend(loc='upper right', bbox_to_anchor=(1.3, 1.1))
plt.show()

# 5. Bubble Plot (Size + Color Encoding)
plt.figure(figsize=(8, 6))
plt.scatter(
    df['Sepal_Length'],
    df['Sepal_Width'],
    s=df['Petal_Length'] * 25,
    c=df['Species'],
    cmap='plasma',
    alpha=0.6
)
plt.xlabel("Sepal Length")
plt.ylabel("Sepal Width")
plt.title("Bubble Plot (Bubble Size = Petal Length)")
plt.colorbar(label="Species")
plt.show()
# ----------------------------------------
# 14. Something Different – Creative & Rare Visuals
# ----------------------------------------
```

```
plt.style.use('seaborn-v0_8-whitegrid')

# 1. Hexbin Density Plot (Very Different Look)
plt.figure(figsize=(8, 6))
plt.hexbin(
    df['Sepal_Length'],
    df['Petal_Length'],
    gridsize=25,
    cmap='magma'
)
plt.colorbar(label="Density")
plt.xlabel("Sepal Length")
plt.ylabel("Petal Length")
plt.title("Hexbin Density Plot (Sepal vs Petal)")
plt.show()

# 2. Contour Density Plot (Smooth & Premium)
plt.figure(figsize=(8, 6))
sns.kdeplot(
    data=df,
    x="Sepal_Length",
    y="Petal_Length",
    hue="Species",
    fill=True,
    alpha=0.5,
    palette="Spectral"
)
plt.title("Contour Density Plot by Species")
```

```
plt.show()

# 3. Swarm + Box Combination (Stylish & Informative) – FIXED
plt.figure(figsize=(9, 6))

sns.boxplot(
    data=df,
    x="Species",
    y="Petal_Length",
    hue="Species",      # ✅ added
    palette="pastel",
    legend=False        # ✅ avoids duplicate legend
)

sns.swarmplot(
    data=df,
    x="Species",
    y="Petal_Length",
    color="black",
    alpha=0.6
)

plt.title("Petal Length Distribution (Box + Swarm)")
plt.show()

# 4. Lollipop Plot (Minimal & Elegant)
means = df.groupby("Species")["Petal_Width"].mean()

plt.figure(figsize=(8, 5))
```

```
plt.stem(
    means.index,
    means.values,
    basefmt=" ",
    linefmt="C2-",
    markerfmt="C2o"
)
plt.xlabel("Species")
plt.ylabel("Mean Petal Width")
plt.title("Lollipop Plot of Mean Petal Width")
plt.show()

# 5. Gradient Line Plot (Very Aesthetic)
plt.figure(figsize=(9, 6))

for species in df['Species'].unique():
    sorted_df = df[df['Species'] == species].sort_values("Sepal_Length")
    plt.plot(
        sorted_df["Sepal_Length"],
        sorted_df["Petal_Length"],
        linewidth=2,
        alpha=0.8,
        label=f"Species {species}"
    )

plt.xlabel("Sepal Length")
plt.ylabel("Petal Length")
plt.title("Gradient-style Trend Lines by Species")
plt.legend()
```

```
plt.show()
```

18. OUTPUT

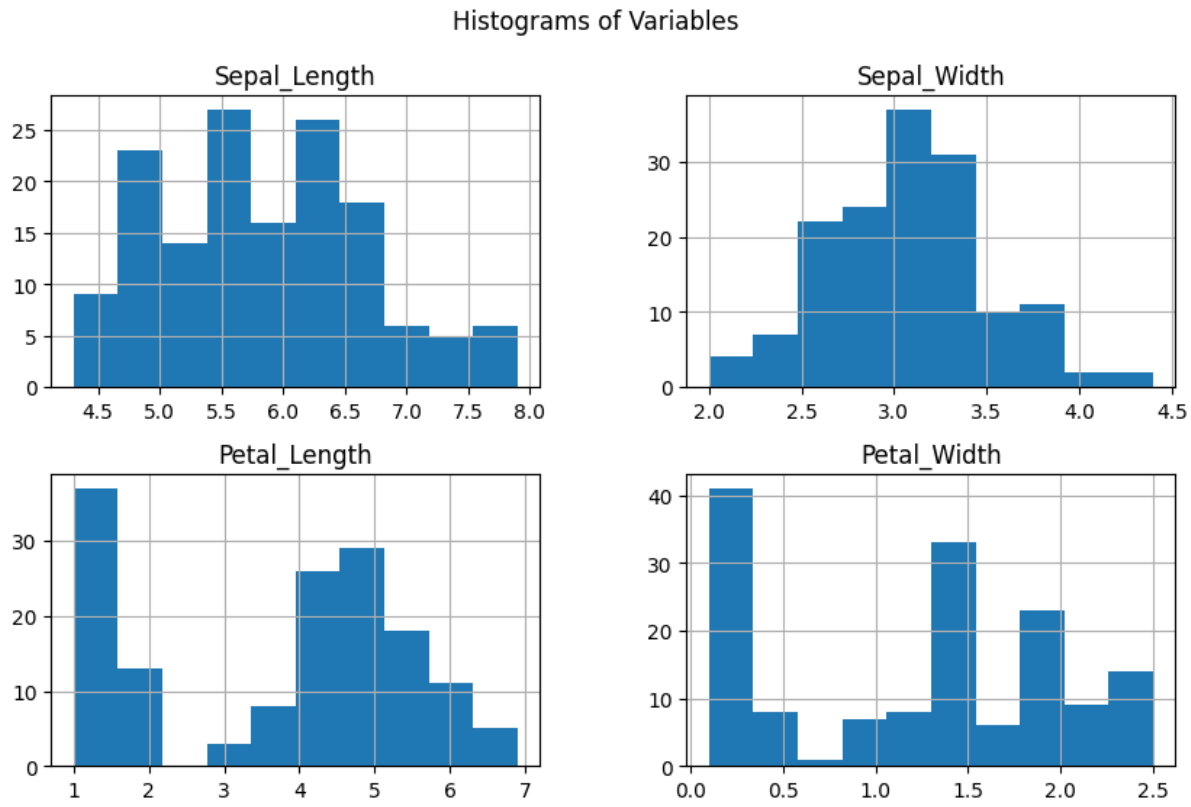

FIG : 1 – HISTOGRAM OF VARIABLES

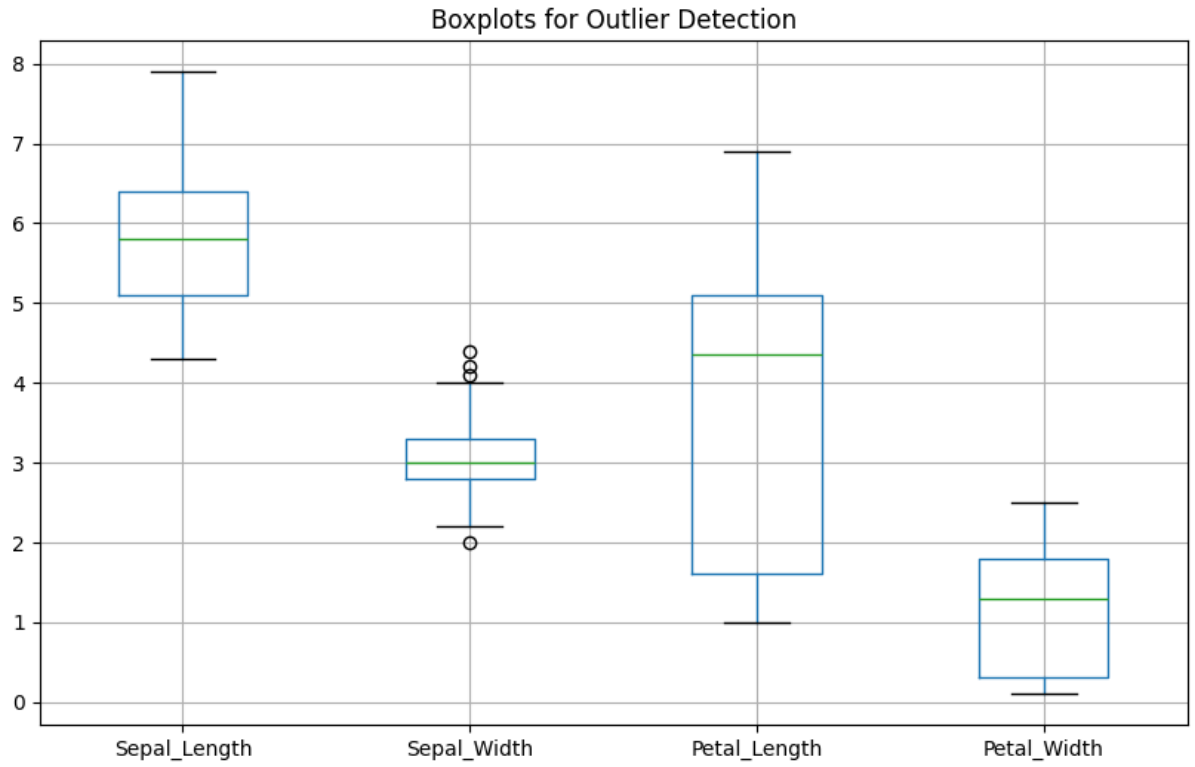

FIG : 2 – BOXPLOTS FOR OUTLIER DETECTION

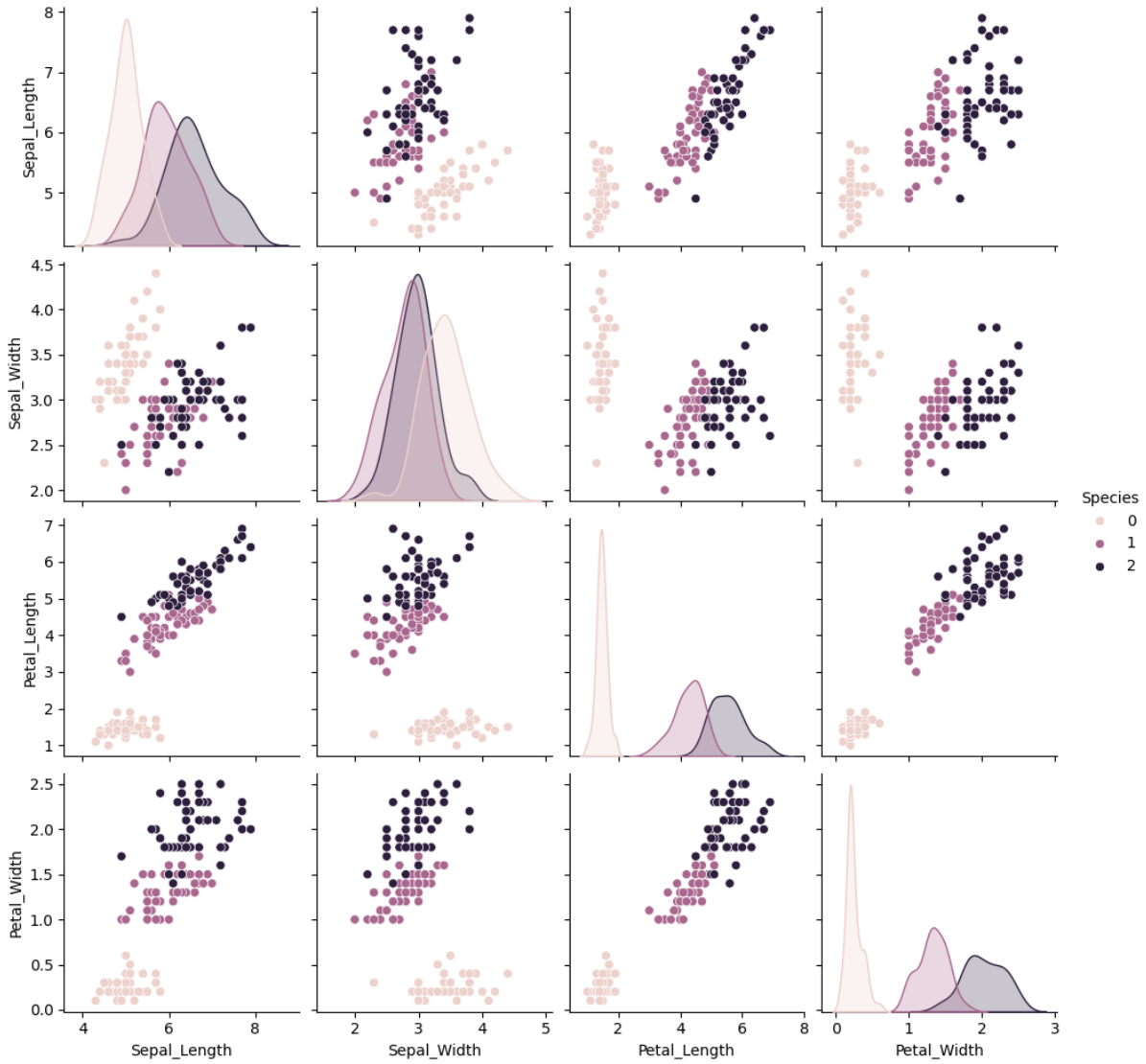

FIG : 3 - PAIR PLOT (SCATTERPLOT MATRIX) OF IRIS DATASET FEATURES BY SPECIES

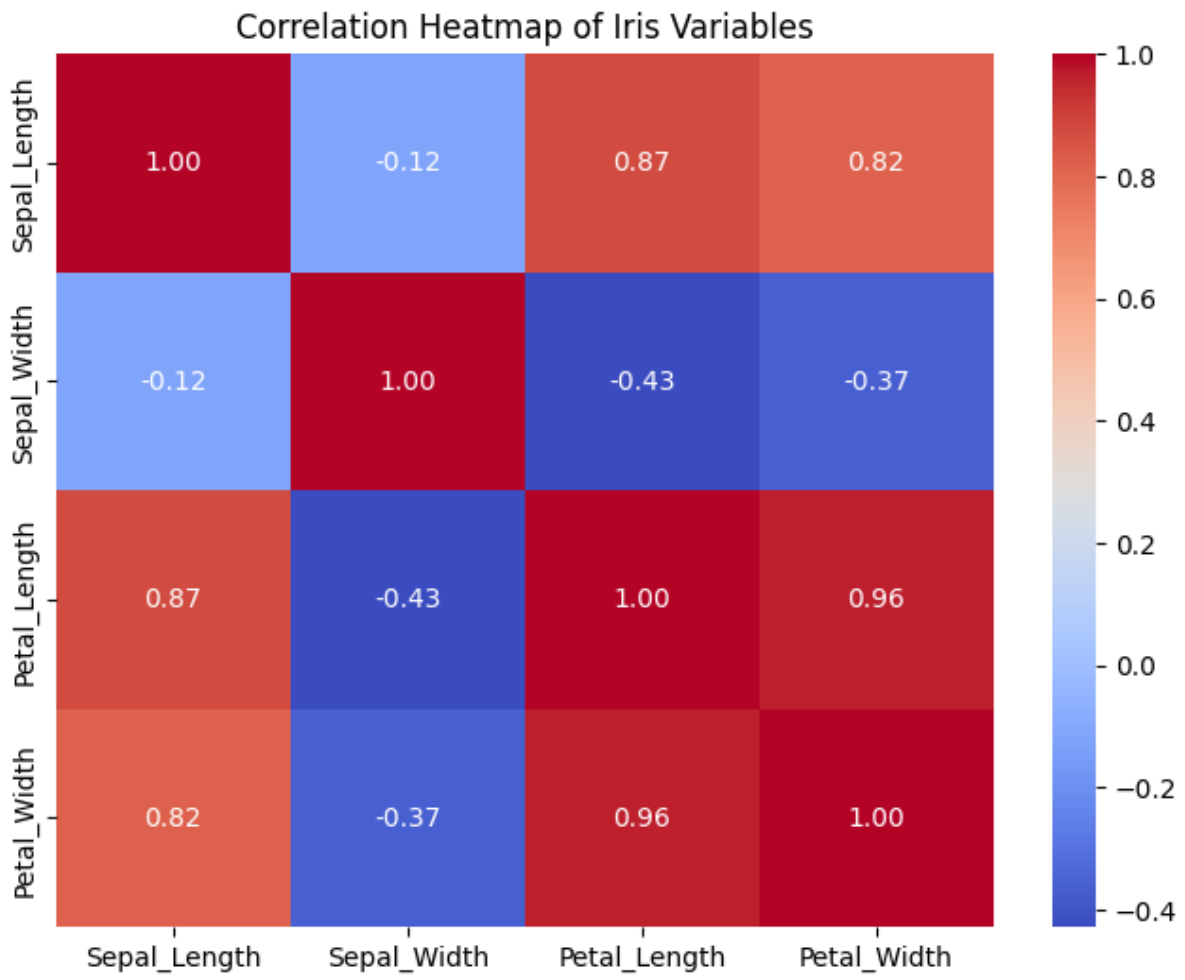

FIG : 4 – CORRELATION HEATMAP OF IRIS VARIABLES

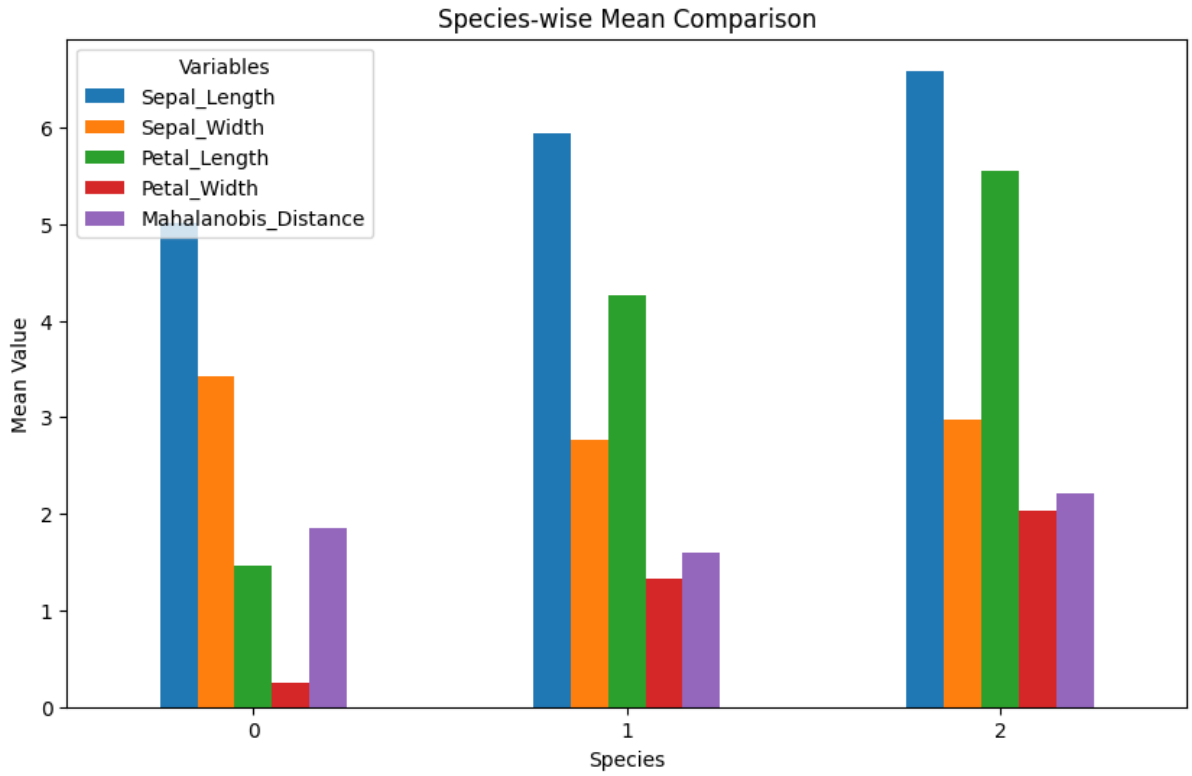

FIG : 5 – SPECIES – WISE MEAN COMPARISON

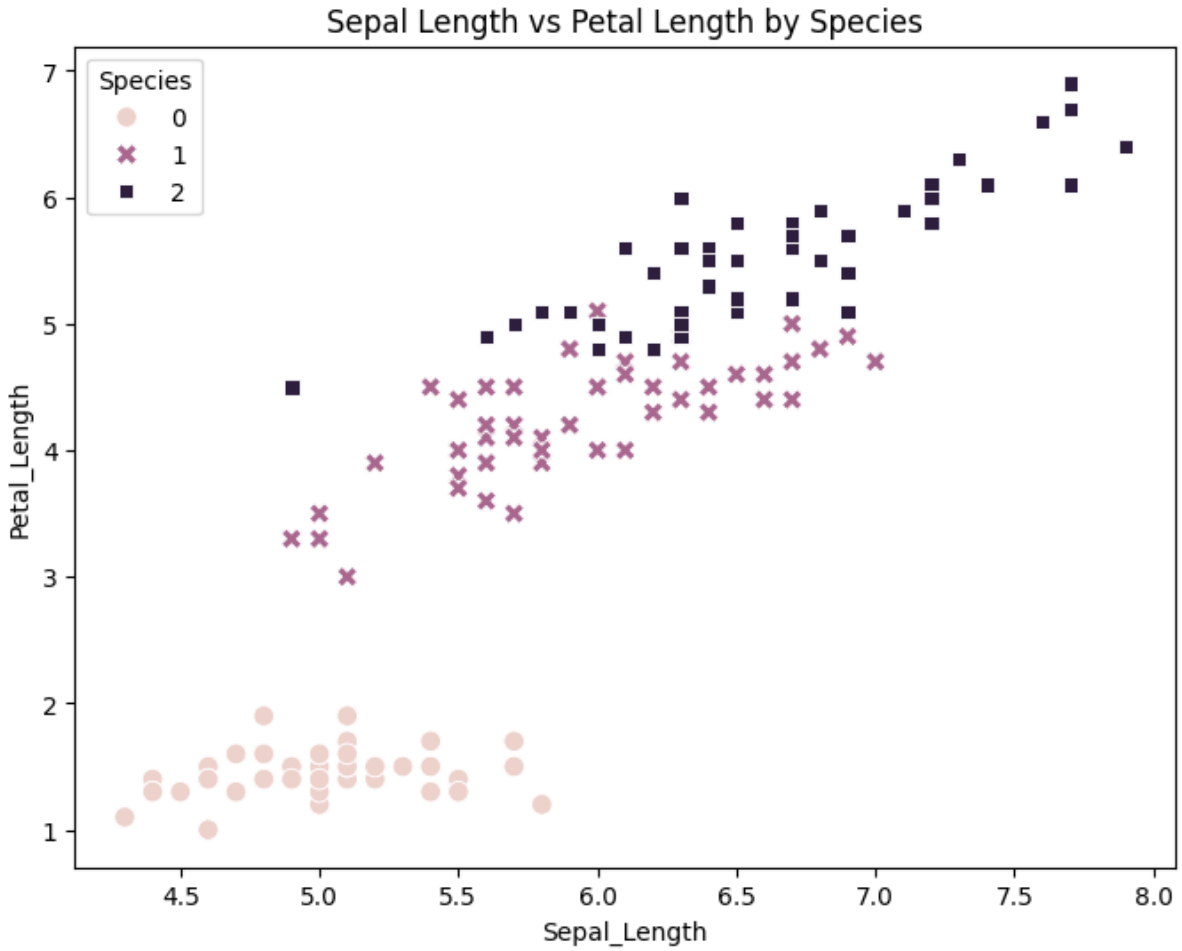

FIG : 6 – SEPAL LENGTH VS PETAL LENGTH BY SPECIES

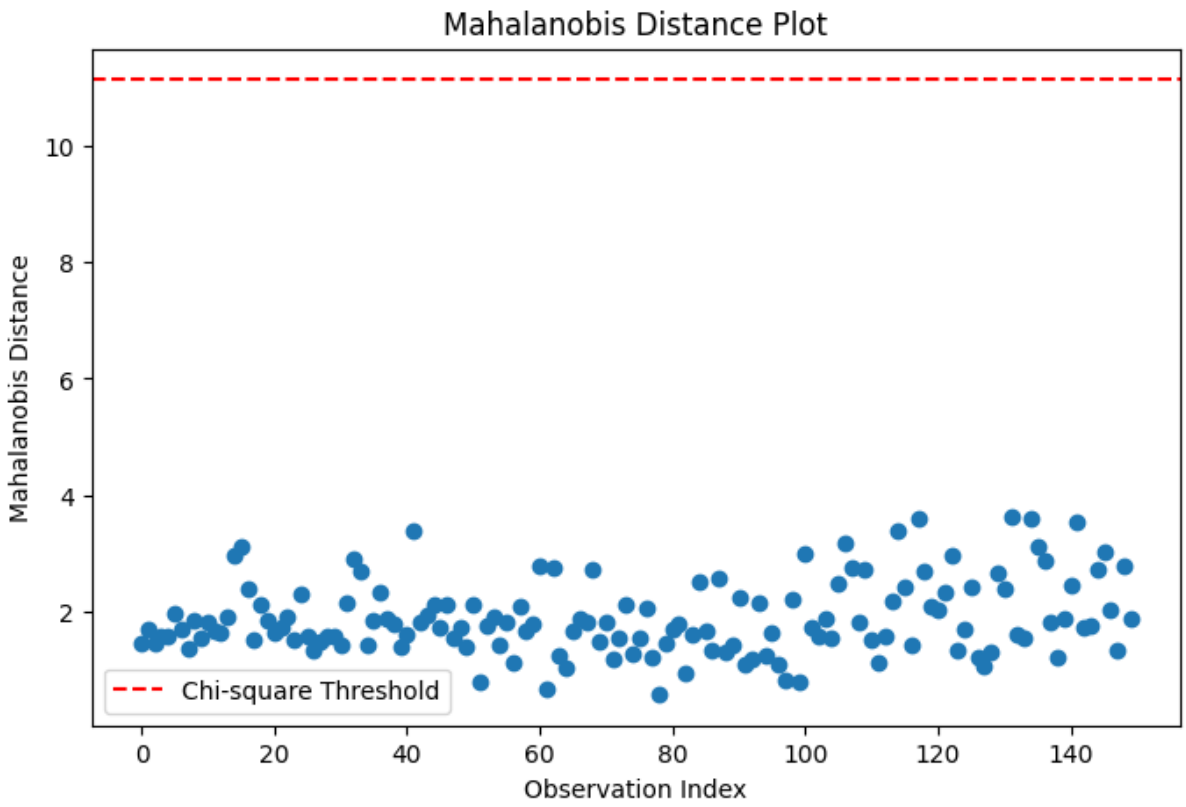

FIG : 7 – MAHALANOBIS DISTANCE PLOT

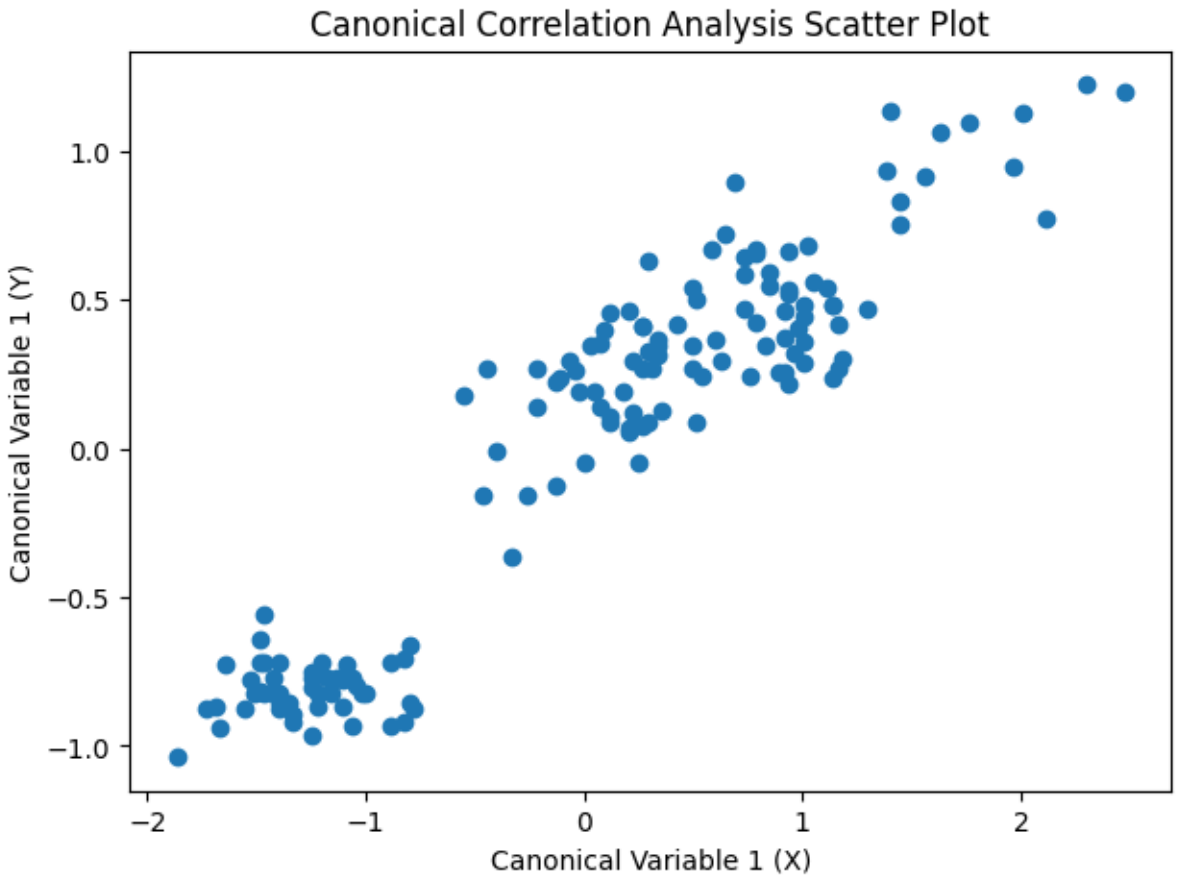

FIG : 8 – CANNONICAL CORRELATION ANALYSIS SCATTER PLOT

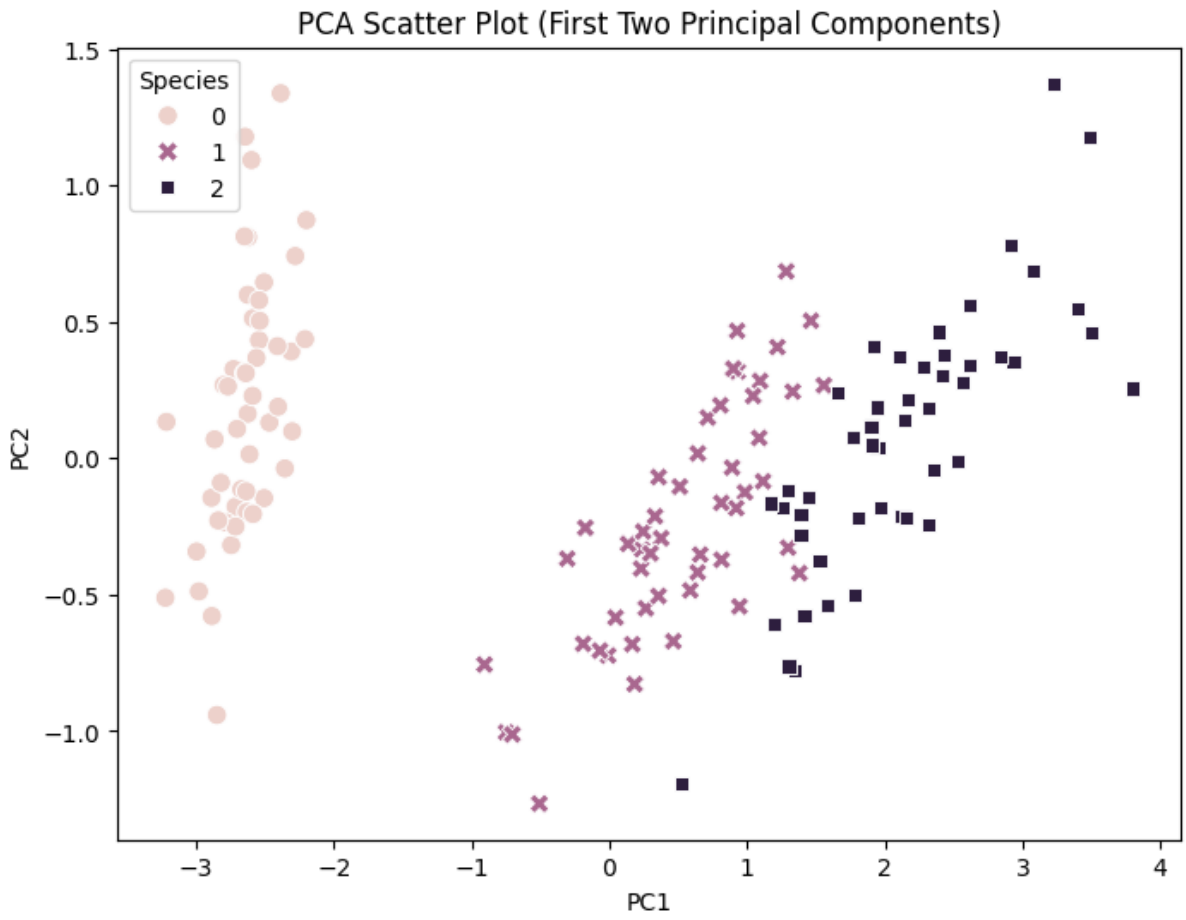

FIG : 9 – PCA SCATTER PLOT

FIG : 10 – VIOLIN PLOT OF FEATURES BY SPECIES

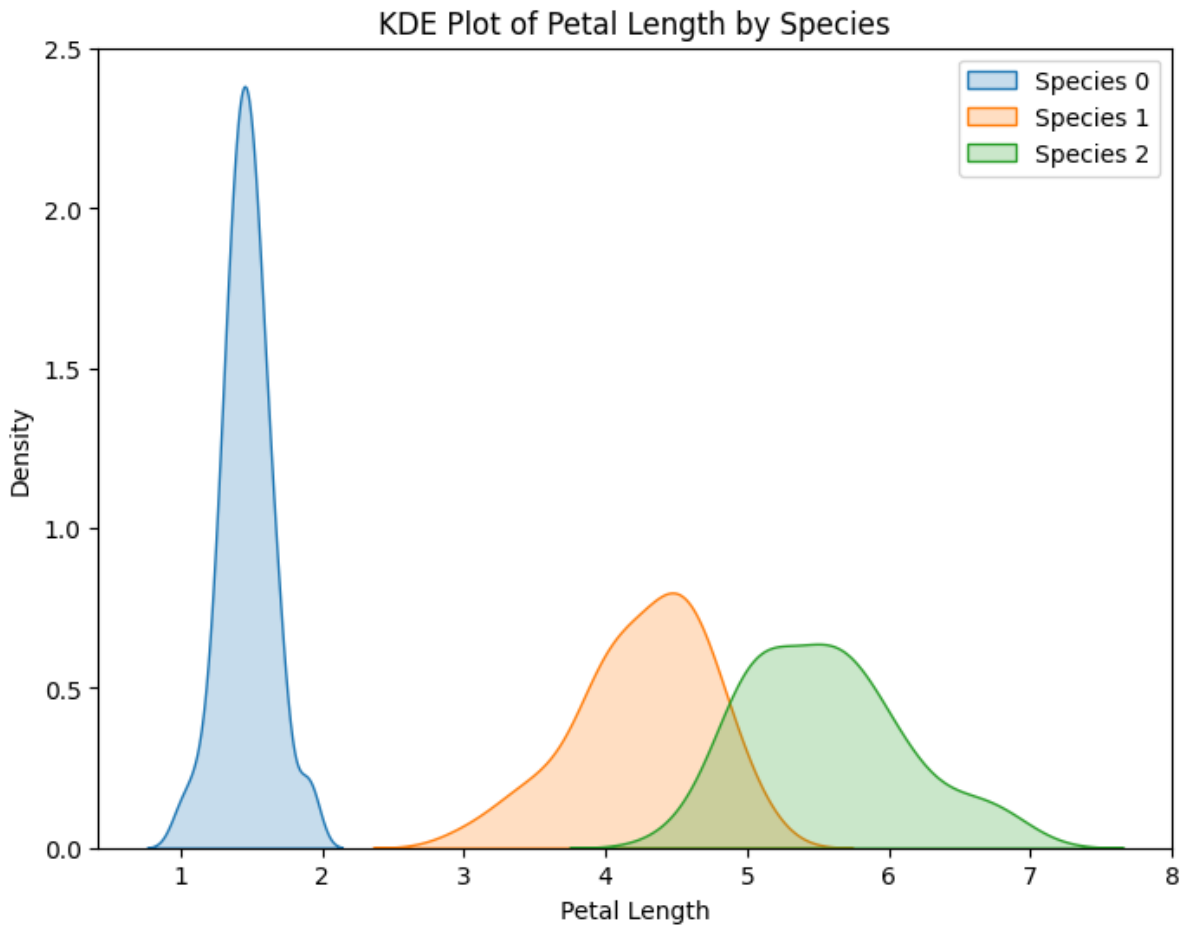

FIG : 12 – KDE PLOT OF PETAL LENGTH BY SPECIES

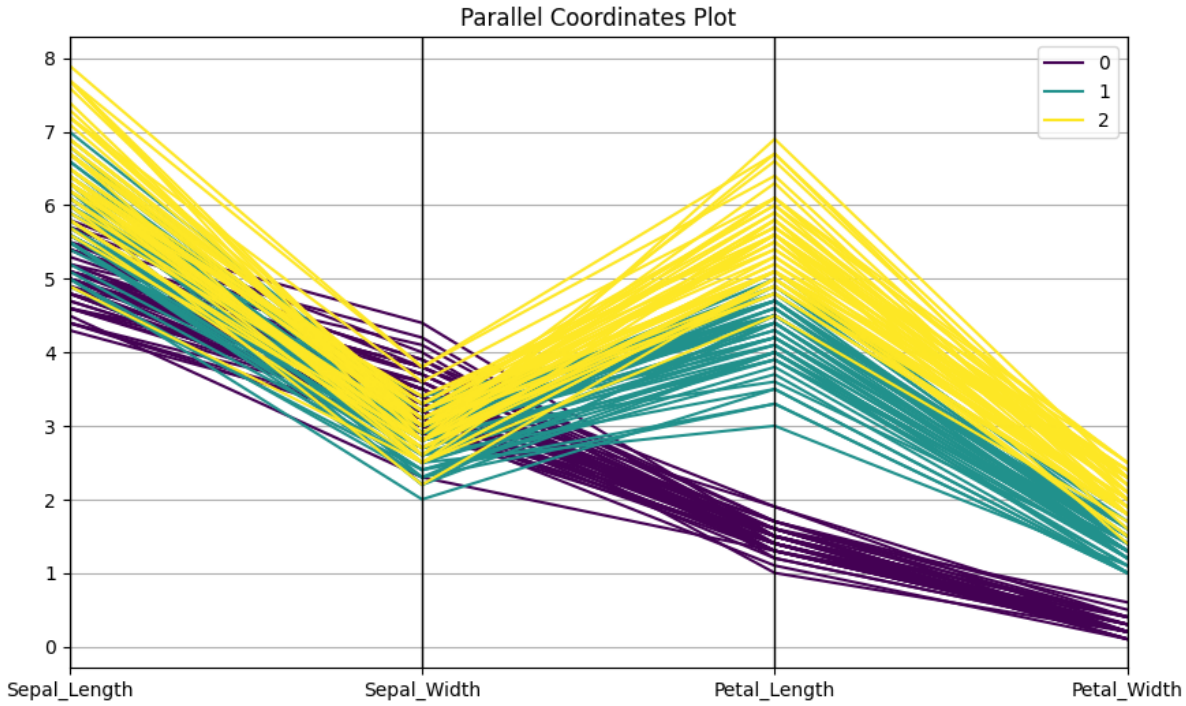

FIG : 13 – PARALLEL COORDINATES PLOT

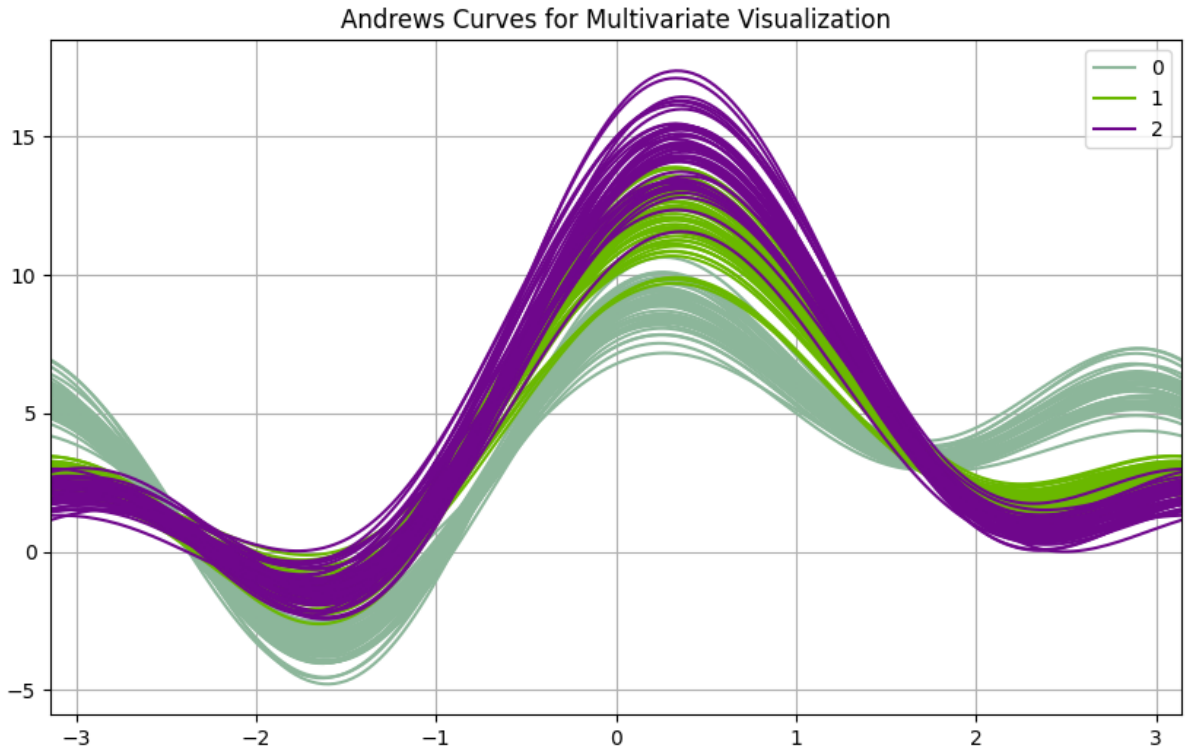

FIG : 14 – ANDREWS CURVES FOR MULTIVARIATE VISUALIZATION


Enhanced Pairplot with KDE
Sepal_Width
Petal_Length
Petal_Width
Mahalanobis_Distance
Sepal_Length
Sepal_Width
Petal_Length
Petal_Width
Mahalanobis_Distance
Species
0
1
2

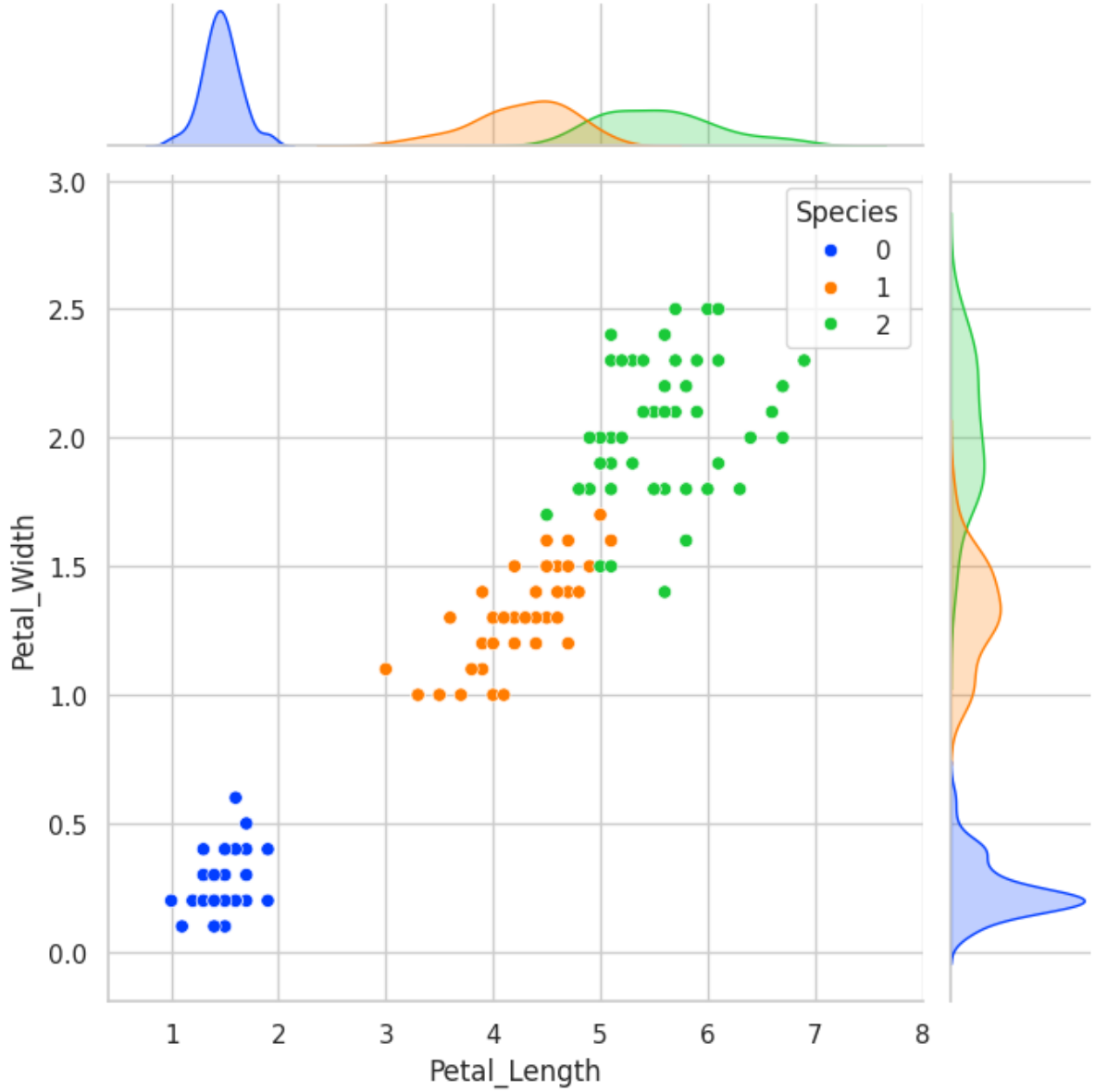

FIG : 15 – ENHANCED PAIRPLOT WITH KDE

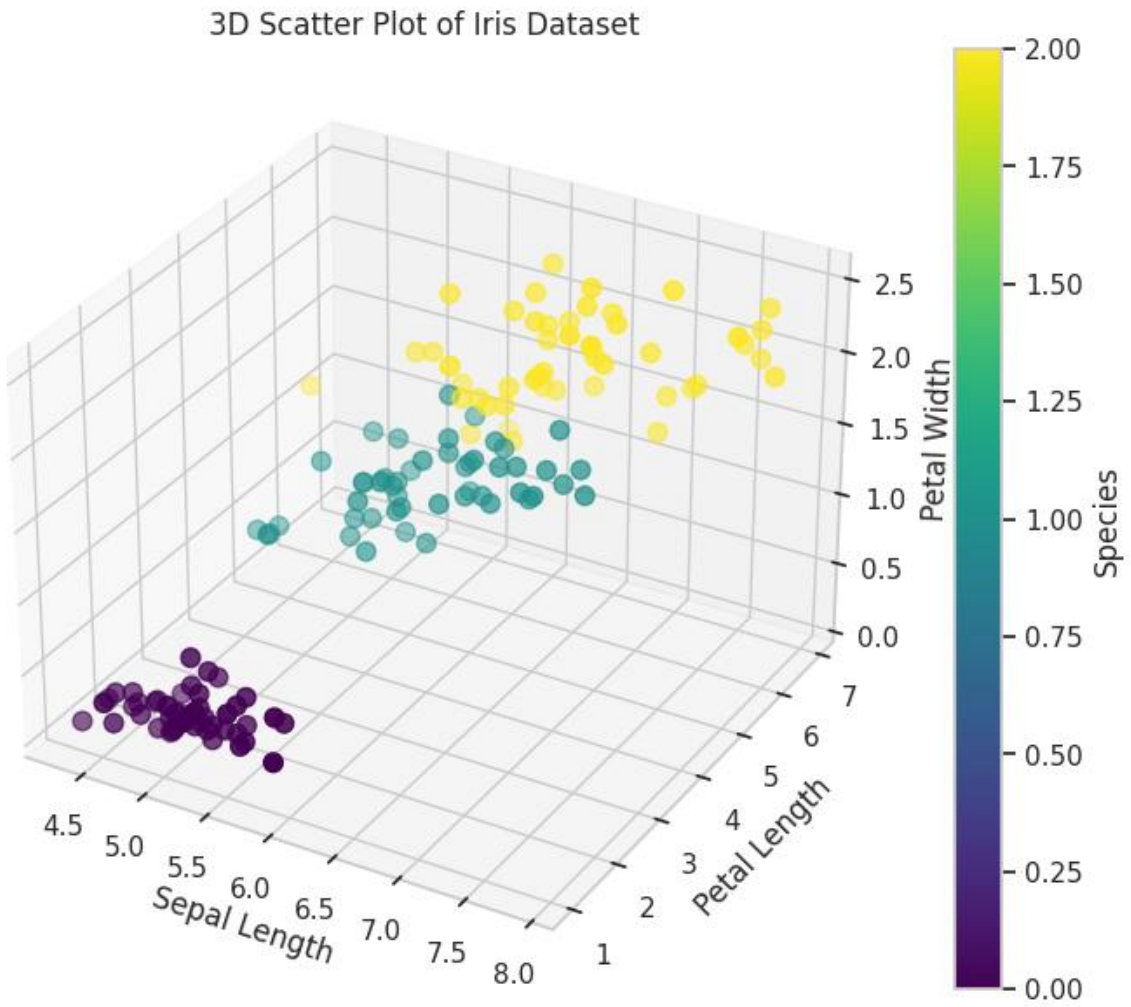

FIG : 16 – 3D SCATTER PLOT OF IRIS DATASET

FIG : 17 – RADAR CHART OF FEATURE MEANS BY SPECIES

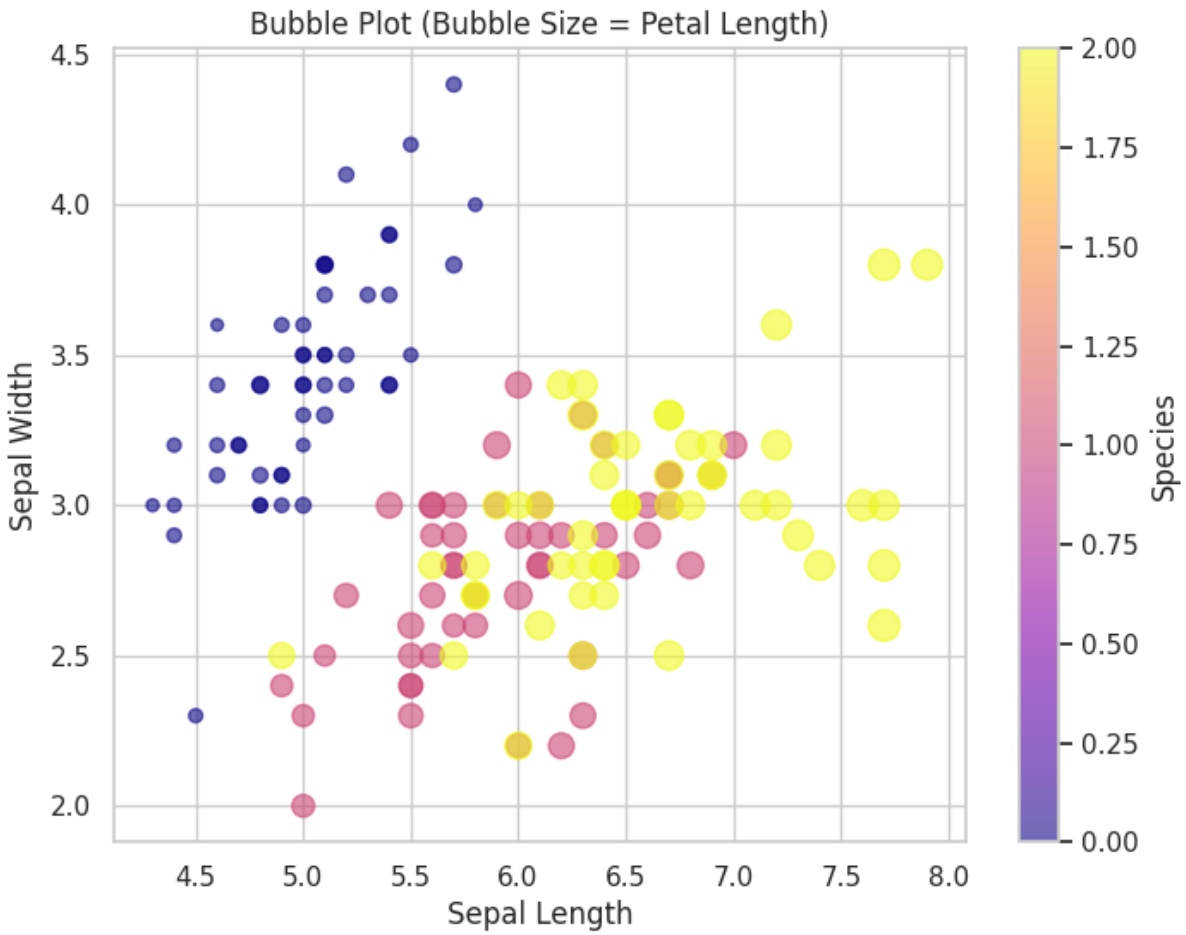

FIG : 18 – BUBBLE PLOT

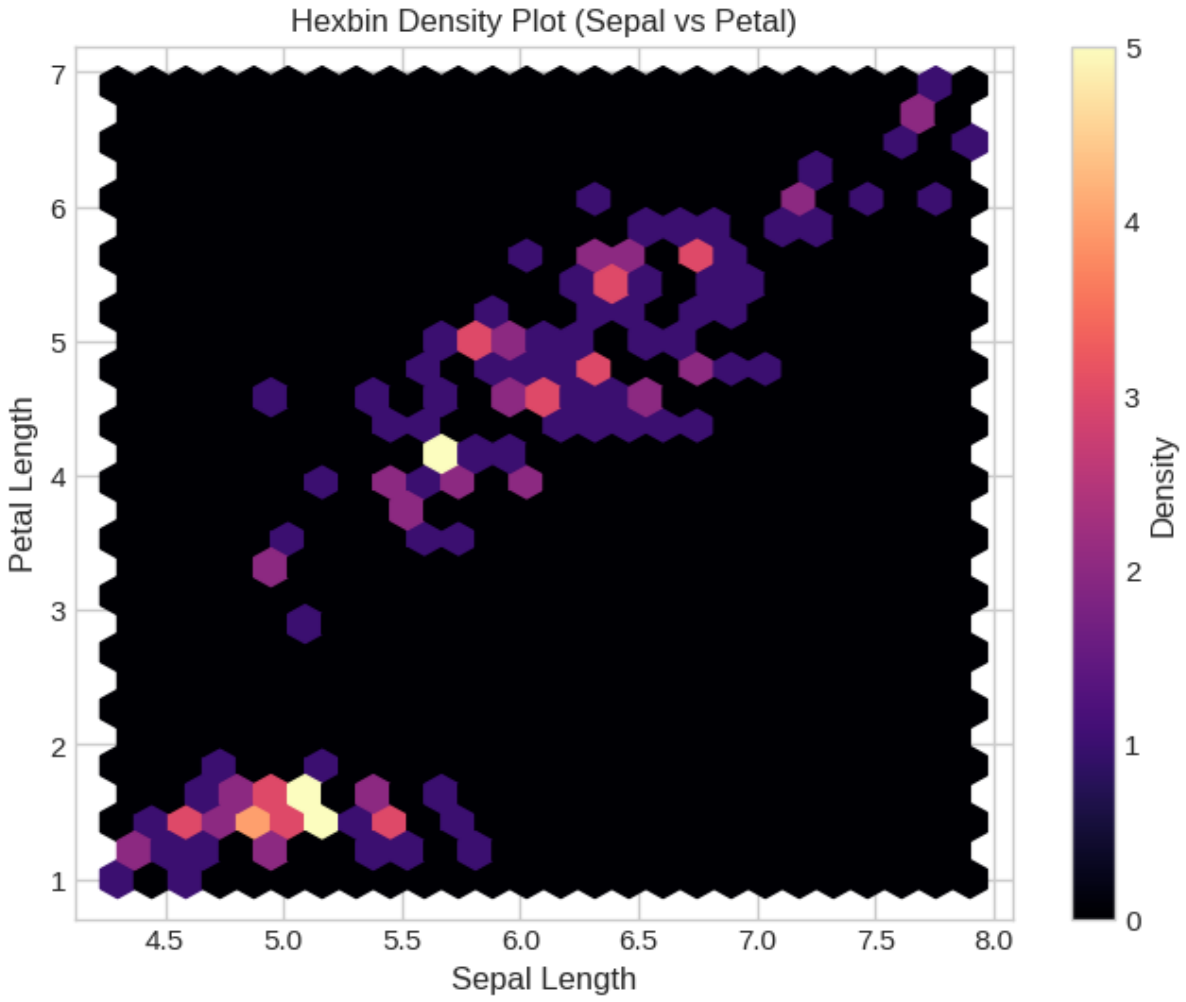

FIG : 19 – HEXBIN DENSITY PLOT

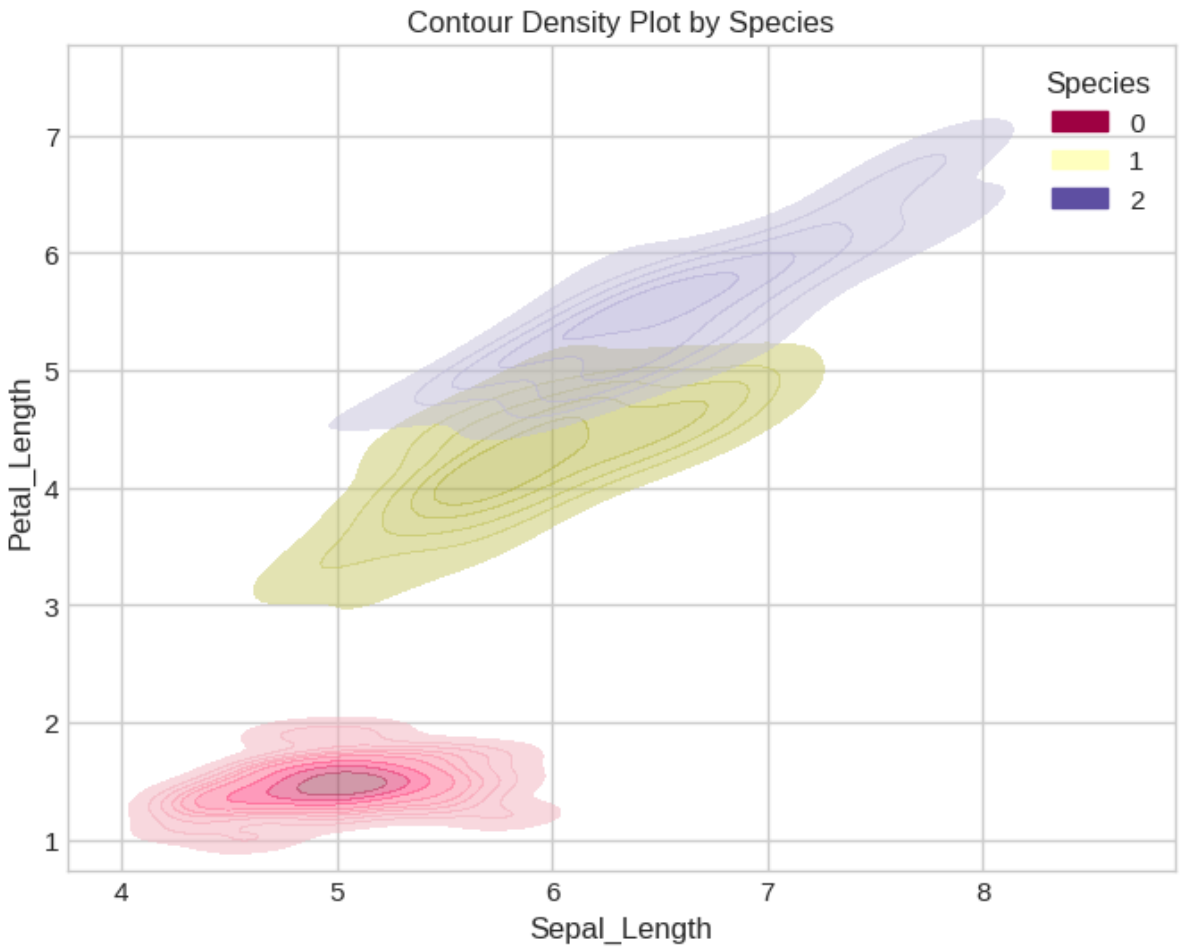

FIG : 20 – CONTOUR DENSITY PLOT BY SPECIES

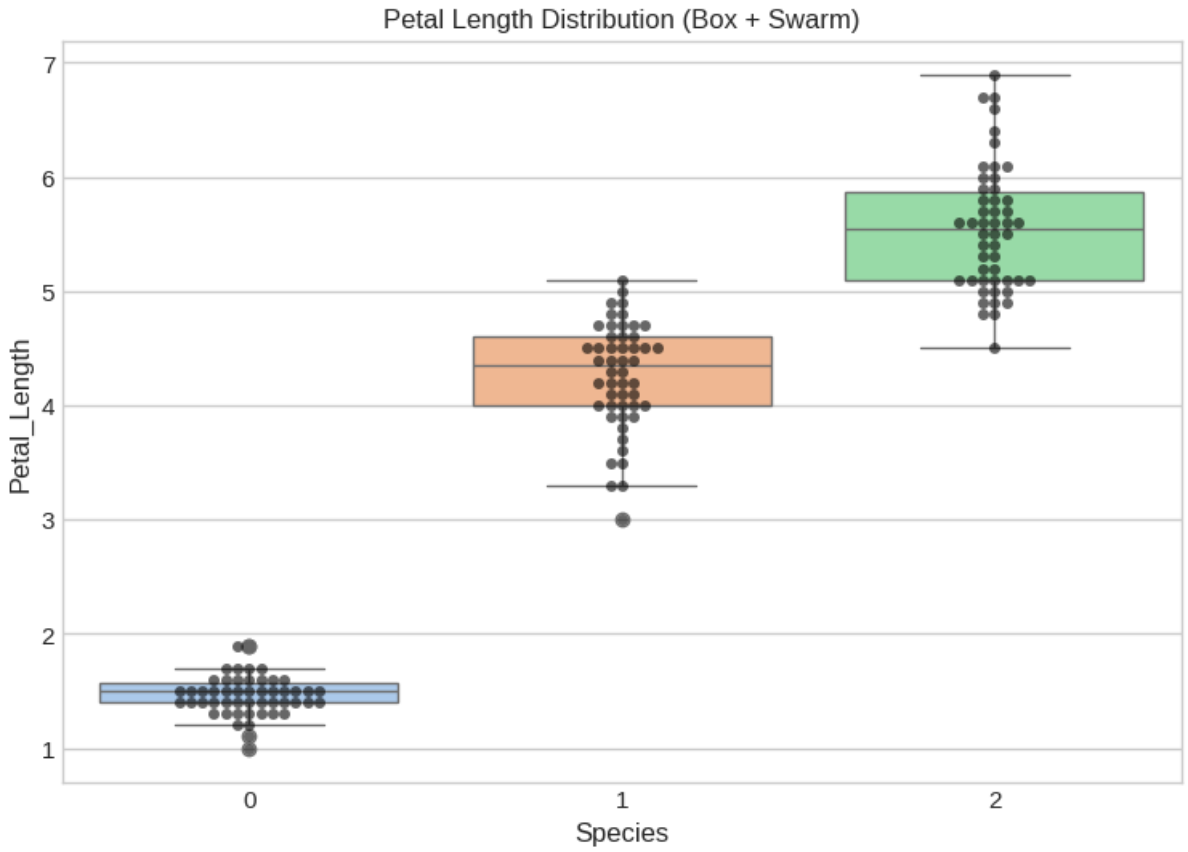

FIG : 21 – PETAL LENGTH DISTRIBUTION

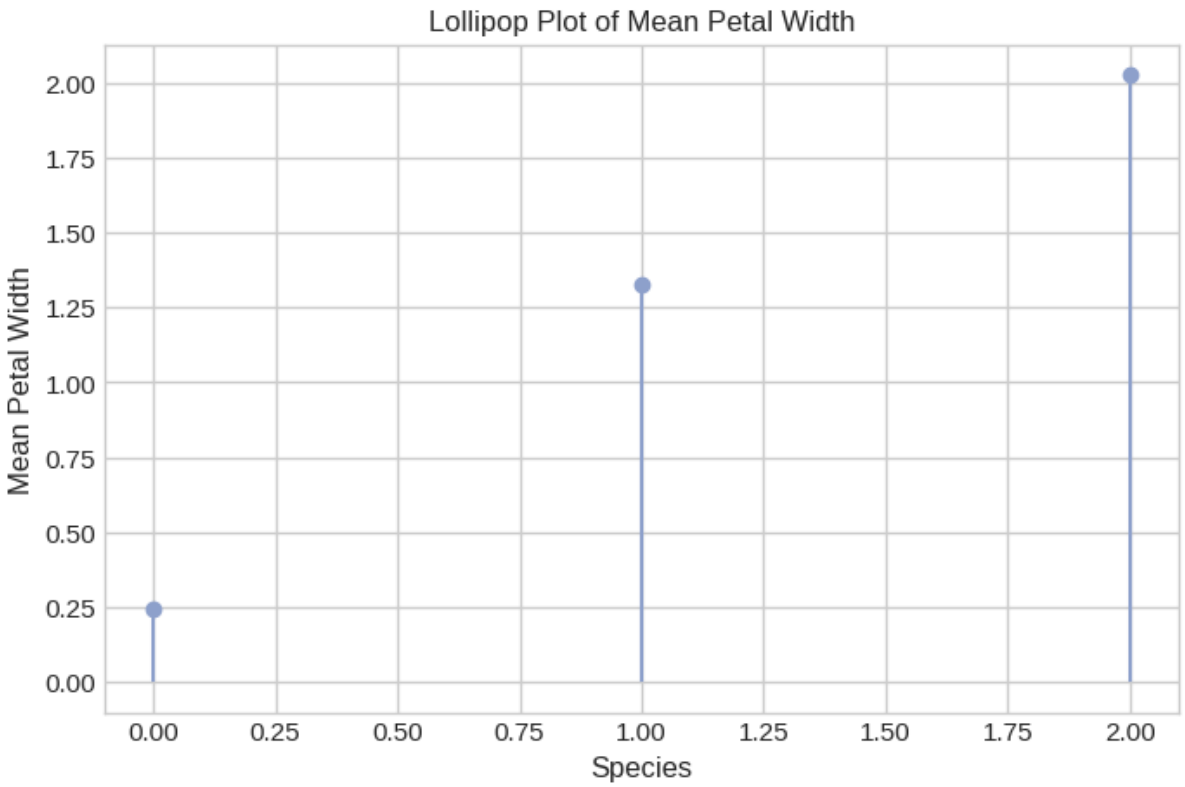

FIG : 22 – LOLLIPOP PLOT OF MEAN PETAL WIDTH

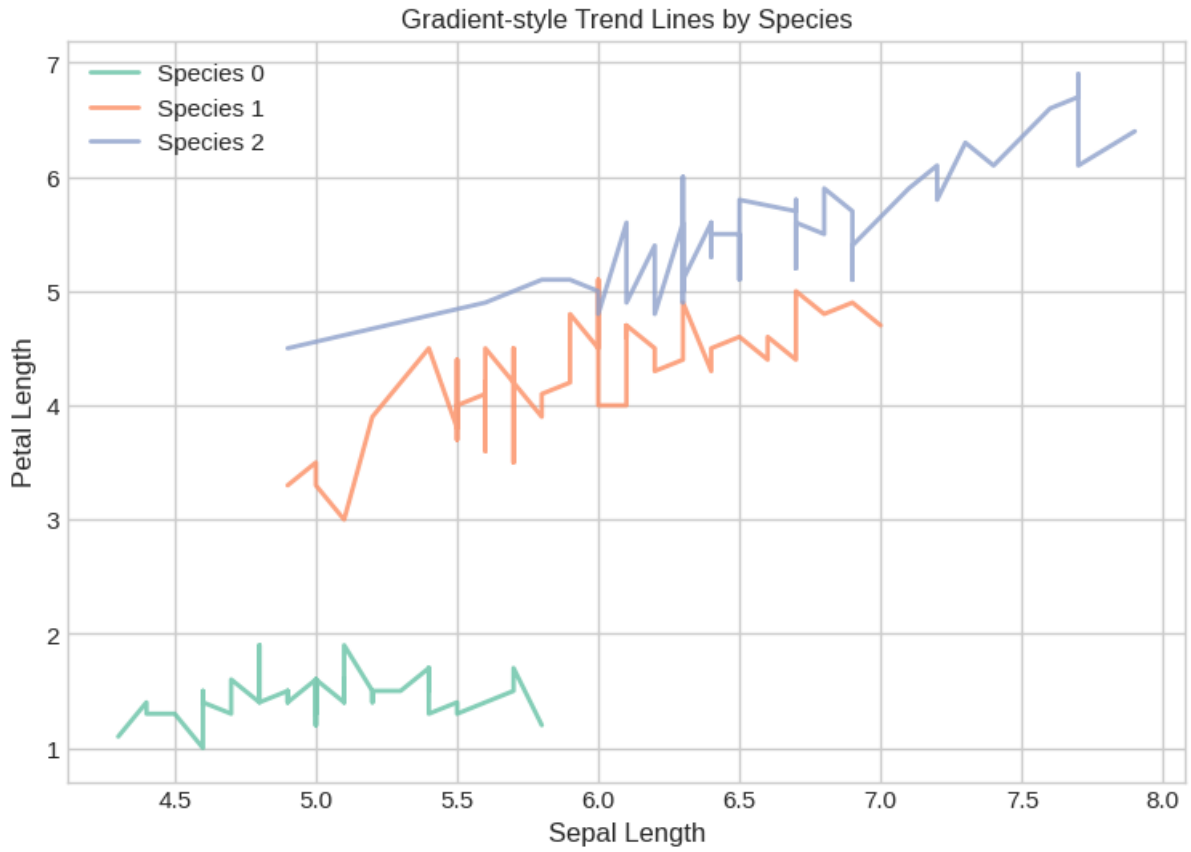

FIG : 23 – GRADIENT – STYLE TREND LINES BY SPECIES