AI Python for Beginners Project 1

1. Setting Up the Environment

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style='whitegrid')
%matplotlib inline
```

2. Loading the Dataset

```
Data=pd.read_csv("/content/Iris.csv")
Data.head()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

3. Data Structure Information

```
Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   SepalLengthCm  150 non-null    float64
 1   SepalWidthCm   150 non-null    float64
 2   PetalLengthCm  150 non-null    float64
 3   PetalWidthCm   150 non-null    float64
 4   Species        150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

4. Check for Missing Values

```
Data.isnull().sum()
```

	0
SepalLengthCm	0
SepalWidthCm	0
PetalLengthCm	0
PetalWidthCm	0
Species	0

dtype: int64

5. Statistical Summary

```
Data.describe()
```

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

Start coding or generate with AI.

6. Data Cleaning

Handle Missing Values

```
Data.dropna(inplace=True)
```

7. Data Types Verification

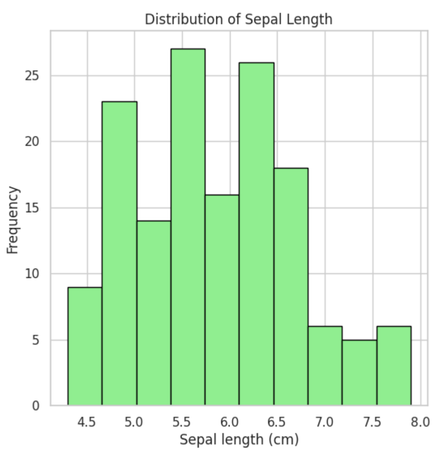
```
Data.dtypes
```

	0
SepalLengthCm	float64
SepalWidthCm	float64
PetalLengthCm	float64
PetalWidthCm	float64
Species	object

dtype: object

8. Performing Data Analysis

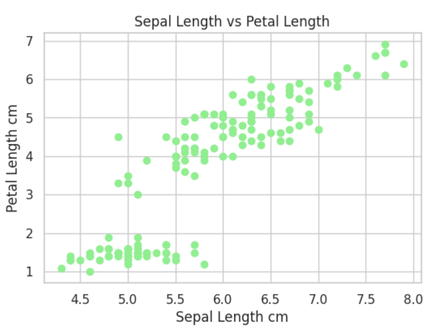
```
# Distribytion of Sepal length
plt.figure(figsize=(6,6))
plt.hist(Data['SepalLengthCm'],bins=10,color="lightgreen",edgecolor="black")
plt.title("Distribution of Sepal Length")
plt.xlabel("Sepal length (cm)")
plt.ylabel("Frequency")
plt.show()
```



Analyzing Relationships Between Variables

Sepal Length vs. Petal Length

```
#Box plot
plt.figure(figsize=(6,4))
plt.scatter(Data["SepalLengthCm"],Data["PetalLengthCm"],c="lightgreen")
plt.title("Sepal Length vs Petal Length")
plt.xlabel("Sepal Length cm")
plt.ylabel("Petal Length cm")
plt.show()
```



Grouping and Aggregation

Mean Measurements by Species

```
Data.groupby("Species").mean()
```

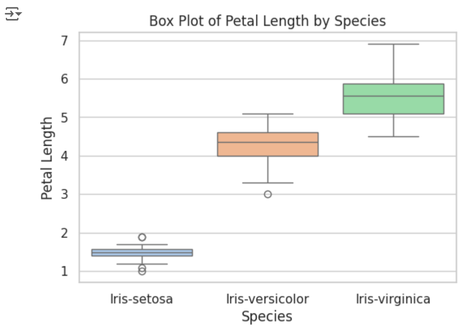
Species	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Iris-setosa	5.006	3.418	1.464	0.244
Iris-versicolor	5.936	2.770	4.260	1.326
Iris-virginica	6.588	2.974	5.552	2.026

Start coding or generate with AI.

```
#Bar plot
plt.figure(figsize=(6,4))
Data.groupby("Species")['SepalLengthCm'].mean().plot(kind='bar',color=['skyblue','lightgreen','lightpink'],edgecolor='black')
plt.title('Mean Sepal Length by Species')
plt.xlabel('Species')
plt.xticks(rotation=45)
plt.ylabel('Mean Sepal Length')
plt.show()
```

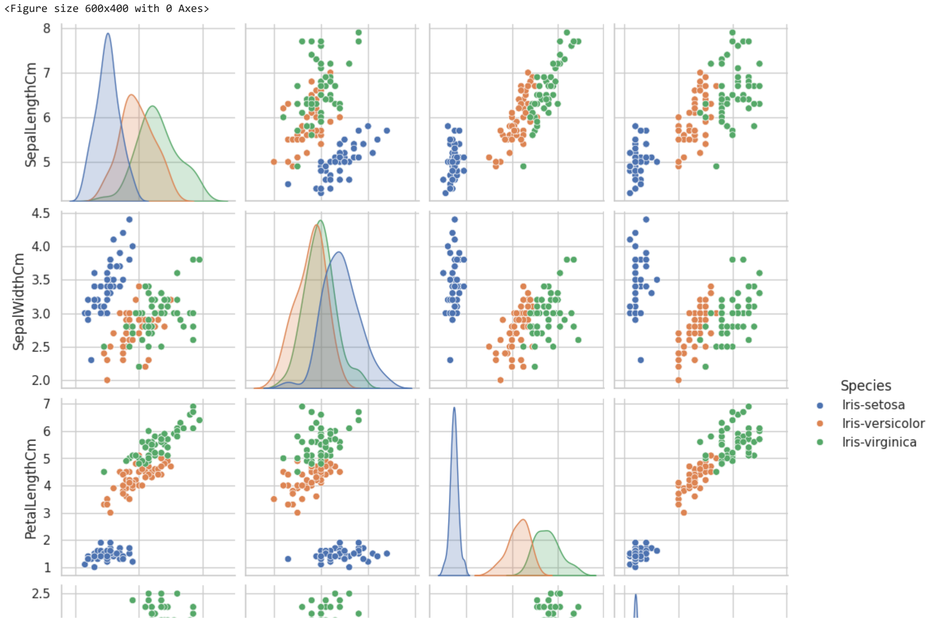

```
# Box plot for petal length by species
plt.figure(figsize=(6,4))
sns.boxplot(x='Species', y='PetalLengthCm',hue='Species',palette='pastel',data=Data)
plt.title('Box Plot of Petal Length by Species')
plt.xlabel('Species')
plt.ylabel('Petal Length')

plt.show()
```

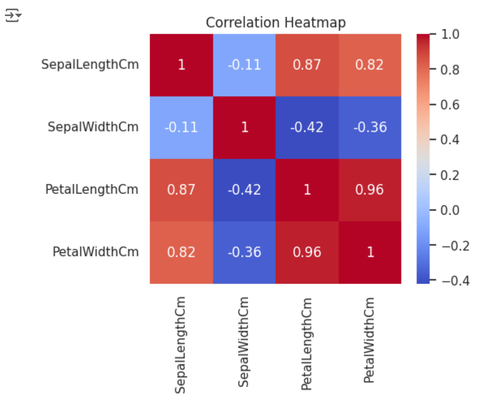



```
# Pair plot of all variables colored by species
plt.figure(figsize=(6,4))
sns.pairplot(Data, hue='Species')
plt.show()
```

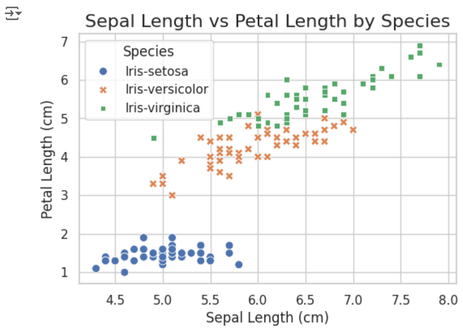
```
<Figure size 600x400 with 0 Axes>
```



```
#Corelation Heatmap
plt.figure(figsize=(5,4))
Numeric_df=Data.select_dtypes(include='number')
corr=Numeric_df.corr()
sns.heatmap(corr,annot=True,cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```




```
# Scatter Plot
plt.figure(figsize=(6, 4))
sns.scatterplot(x='SepalLengthCm', y='PetalLengthCm', style= 'Species',hue='Species', data=Data, s=50)
plt.title('Sepal Length vs Petal Length by Species', fontsize=16)
plt.xlabel('Sepal Length (cm)', fontsize=12)
plt.ylabel('Petal Length (cm)', fontsize=12)
plt.legend(title='Species')
plt.show()
```



Conclusion

The objective of this analysis was to explore the iris dataset using python to understand the relationship between features and classify the different species of iris. Our analysis revealed several key insights:

- A positive correlation was observed between Sepal Length and Petal Length, with both features increasing together across the dataset.
- Iris-Virginica had the highest mean Sepal Length, followed by Iris-Versicolor, and Iris-Setosa had the smallest mean Sepal Length.
- The box plot analysis of Petal Length showed varying outliers:

  Iris-Virginica had no outliers. Iris-Versicolor had one outlier below the minimum. Iris-Setosa had three outliers below the minimum and one outlier above the maximum.
- The correlation analysis found a weak negative relationship between Sepal Width and Sepal Length, a moderate negative relationship between Sepal Length and Petal Length, and strong positive relationships between Petal Length and Sepal Length, as well as between Petal Width and Sepal Length. Additionally, Sepal Width and Petal Width had a moderate negative correlation, while Petal Length and Petal Width showed almost no correlation.
- Overall, this project demonstrates how data analysis and visualization techniques can be used to gain valuable insights from the Iris dataset.