Course Title: Artificial Intelligence

Waqas Mehmood

Lab # 4

14th October, 2025

What You'll Learn Today

- Introduction to Data Visualization
- Setting Up Jupyter Notebook and Loading Datasets
- Matplotlib Basics: *Plots and Customization*
- Seaborn Basics: *Statistical Visualizations*
- Advanced Visualizations and Best Practices
- Interactive Lab Exercises and Q&A

Why Visualize Data?

a) Helps uncover patterns, trends, and insights.
b) Communicates complex information effectively.
c) Essential for exploratory data analysis (EDA).

Key Libraries:

- **Matplotlib**: Flexible, low-level plotting library for creating static, animated, and interactive visualizations.
- **Seaborn**: Built on Matplotlib; simplifies creating informative statistical graphics with high-level interfaces.

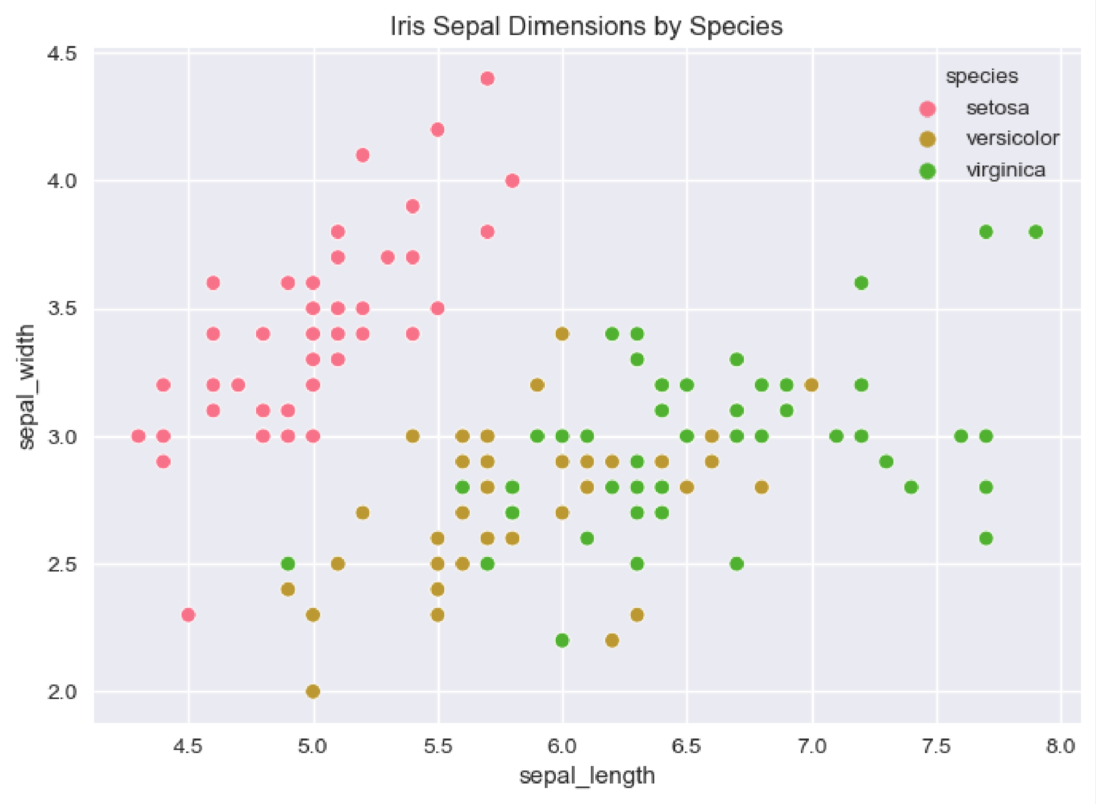
Practice Material

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

# Load built-in dataset from Seaborn
df = sns.load_dataset('iris')
```

Plot Types – Scatter Plot

Function

```
sns.scatterplot(data=df, x='sepal_length', y='sepal_width', hue='species')

plt.title('Iris Sepal Dimensions by Species')

plt.show()
```

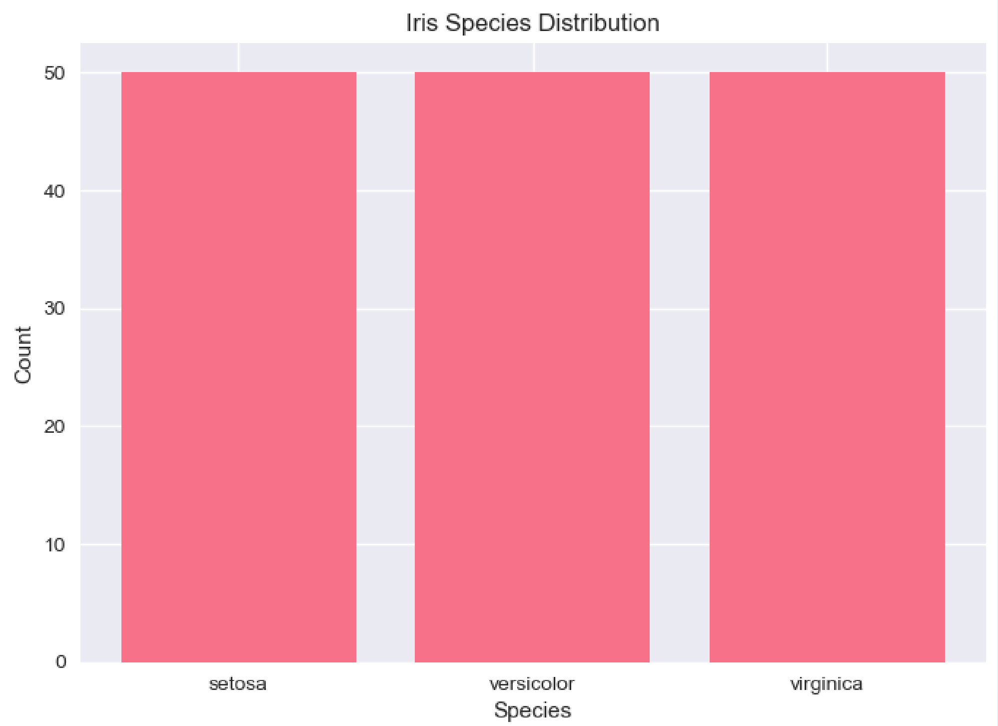


Description: This would show a scatter of points clustering by species dimensions.

Plot Types – Bar Chart

Function

```
species_count = df['species'].value_counts()

plt.bar(species_count.index, species_count.values)

plt.title('Iris Species Distribution')

plt.xlabel('Species')

plt.ylabel('Count')

plt.show()
```

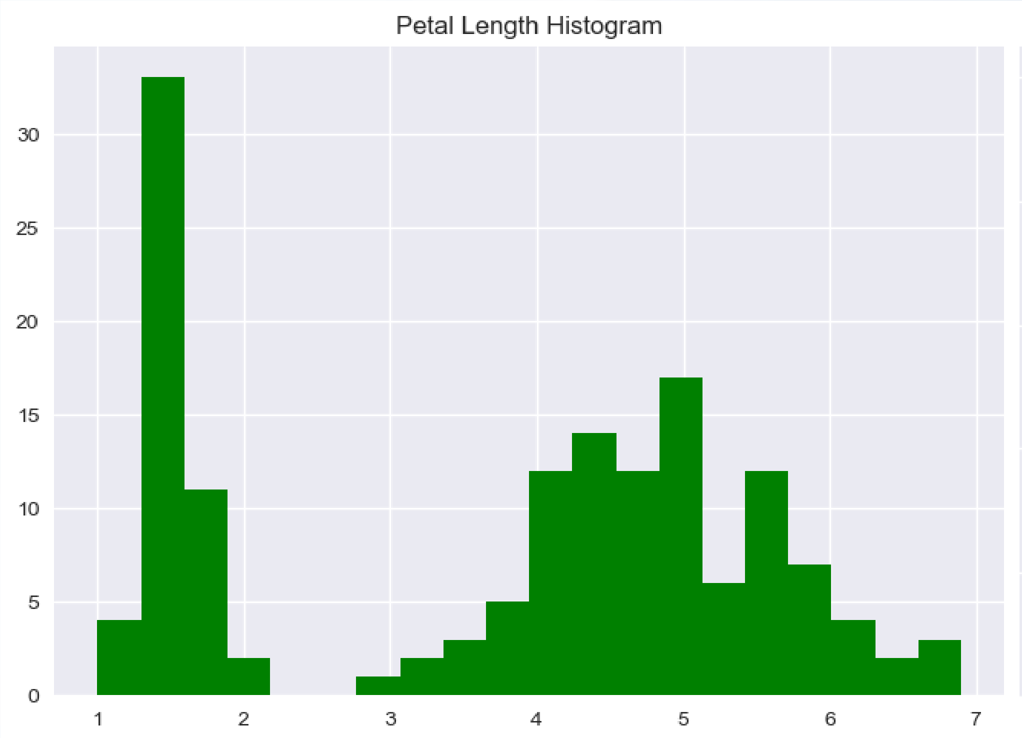


Description: Bars showing equal counts for setosa, versicolor, virginica.

Plot Types – Histogram

Function

```
plt.hist(df['petal_length']
, bins=20, color='green')

plt.title('Petal Length
Histogram')

plt.show()
```



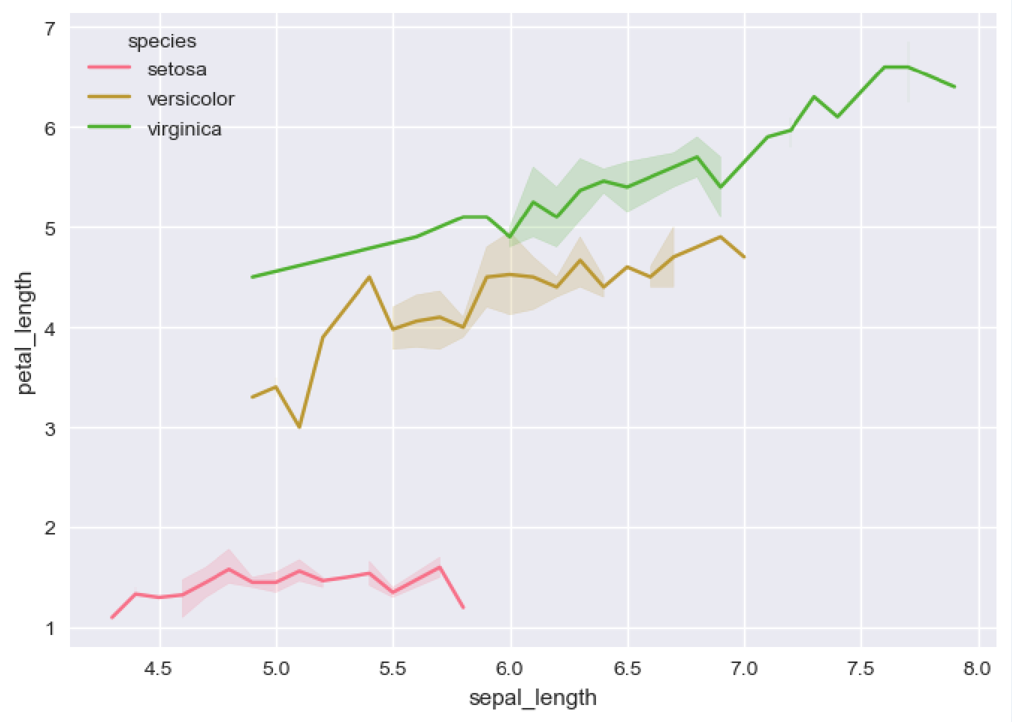
Description: Bimodal distribution highlighting species differences.

Plot Types – Lineplot

Function

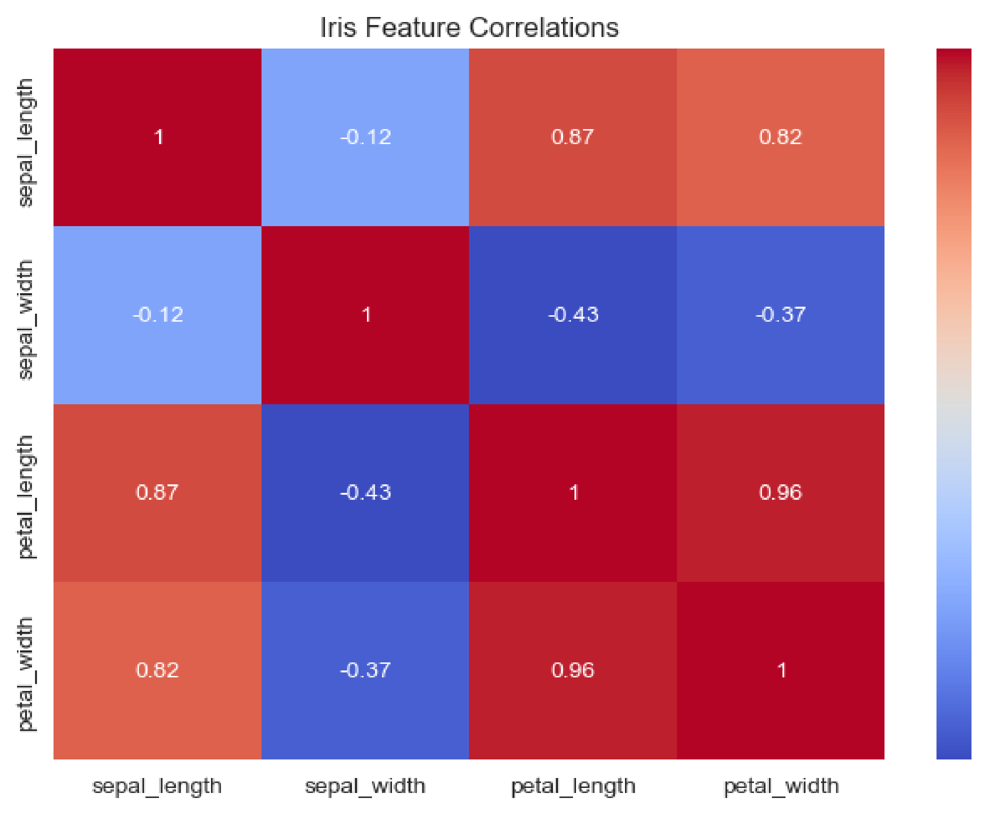
```
sns.lineplot(data=df, x='sepal_length',
y='petal_length', hue='species')
```

Description: Lines showing positive correlation per species.



Plot Types – Heatmap

```
corr = df.drop('species',
axis=1).corr()

sns.heatmap(corr,
annot=True, cmap='coolwarm')

plt.title('Iris Feature
Correlations')

plt.show()
```



Description: Color grid showing strong positive correlations like petal_length and petal_width.

What is a Correlation Heatmap?

A heatmap shows the correlation coefficients between pairs of variables, ranging from -1 to 1.

A value close to 1 (red) indicates a strong positive correlation (as one variable increases, the other does too).

A value close to -1 (blue) indicates a strong negative correlation (as one variable increases, the other decreases).

A value near 0 (light colors) suggests little to no correlation.

Key Observations:

Sepal Length vs. Petal Length (0.87): There's a strong positive correlation, meaning longer sepals tend to be associated with longer petals.

Sepal Width vs. Petal Length (-0.43): There's a moderate negative correlation, suggesting that wider sepals are somewhat linked to shorter petals.

Petal Length vs. Petal Width (0.96): This is a very strong positive correlation, indicating that longer petals are highly likely to be wider as well.

Sepal Length vs. Sepal Width (-0.12): The correlation is weak and slightly negative, showing little relationship between these two measurements.

Heatmap - Insights

This heatmap helps identify which features are most related, which can be useful for tasks like feature selection in machine learning or understanding the dataset’s structure.

For example, since petal length and petal width are highly correlated (0.96), they might provide redundant information in some analyses.

<u>The weak correlations</u> involving sepal width suggest it may vary more independently of the other features.

Subplots

```
fig, axs = plt.subplots(1, 2, figsize=(12, 5))

axs[0].scatter(df['sepal_length'], df['sepal_width'])

axs[1].hist(df['petal_length'])

plt.show()
```