

shadow-fox

March 16, 2025

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

```
[ ]: from google.colab import files
uploaded = files.upload()
```

<IPython.core.display.HTML object>

Saving archive.zip to archive (1).zip

```
[ ]: import zipfile
import os

# Extract the ZIP file
with zipfile.ZipFile('archive.zip', 'r') as zip_ref:
    zip_ref.extractall('extracted_files') # Extract to 'extracted_files' folder

# List the extracted files
os.listdir('extracted_files')
```

```
[ ]: ['iris.data.csv']
```

```
[ ]: data = sns.load_dataset('iris')
data.head()
```

```
[ ]:      sepal_length  sepal_width  petal_length  petal_width  species
0           5.1           3.5           1.4           0.2  setosa
1           4.9           3.0           1.4           0.2  setosa
2           4.7           3.2           1.3           0.2  setosa
3           4.6           3.1           1.5           0.2  setosa
4           5.0           3.6           1.4           0.2  setosa
```

Matplotlib Overview:-Supports a wide range of plot types, including line, bar, scatter, and histogram. Allows full control over plot elements like axes, labels, and legends, making it ideal for publication-quality visuals

- Low-level plotting library

- High customization
- Suitable for static visualizations

```
[ ]: import pandas as pd
import matplotlib.pyplot as plt

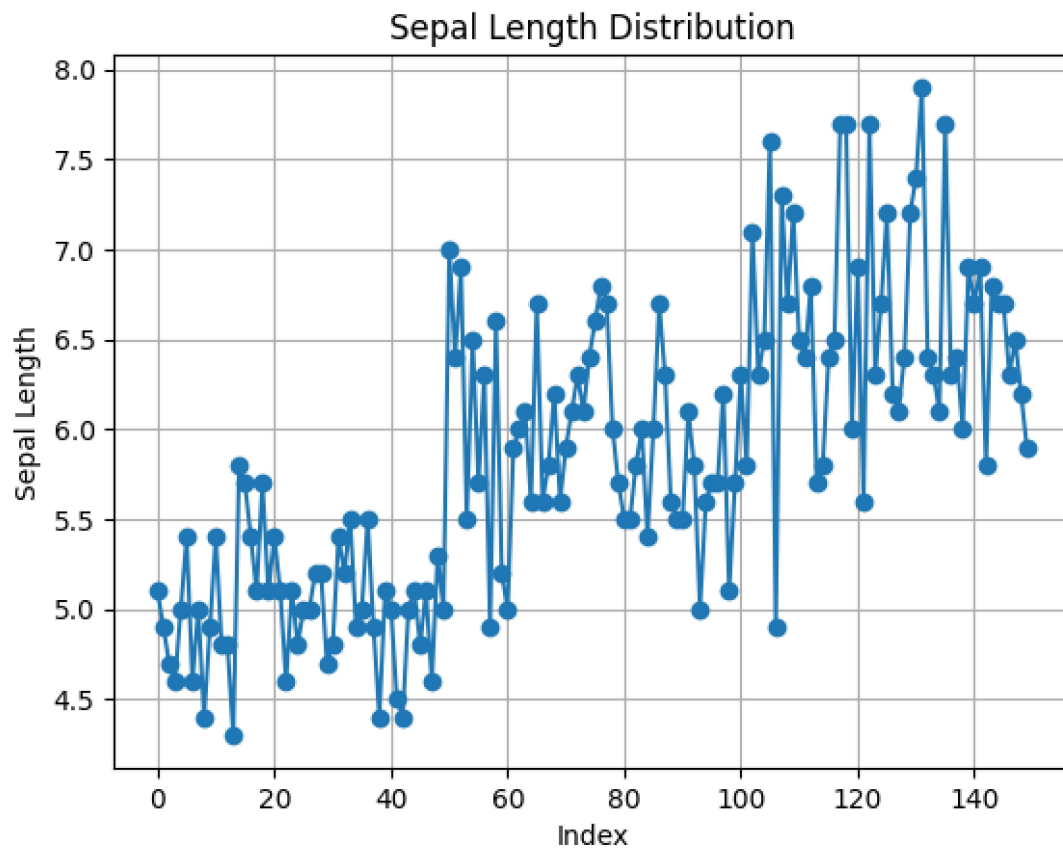
df = pd.DataFrame(data)

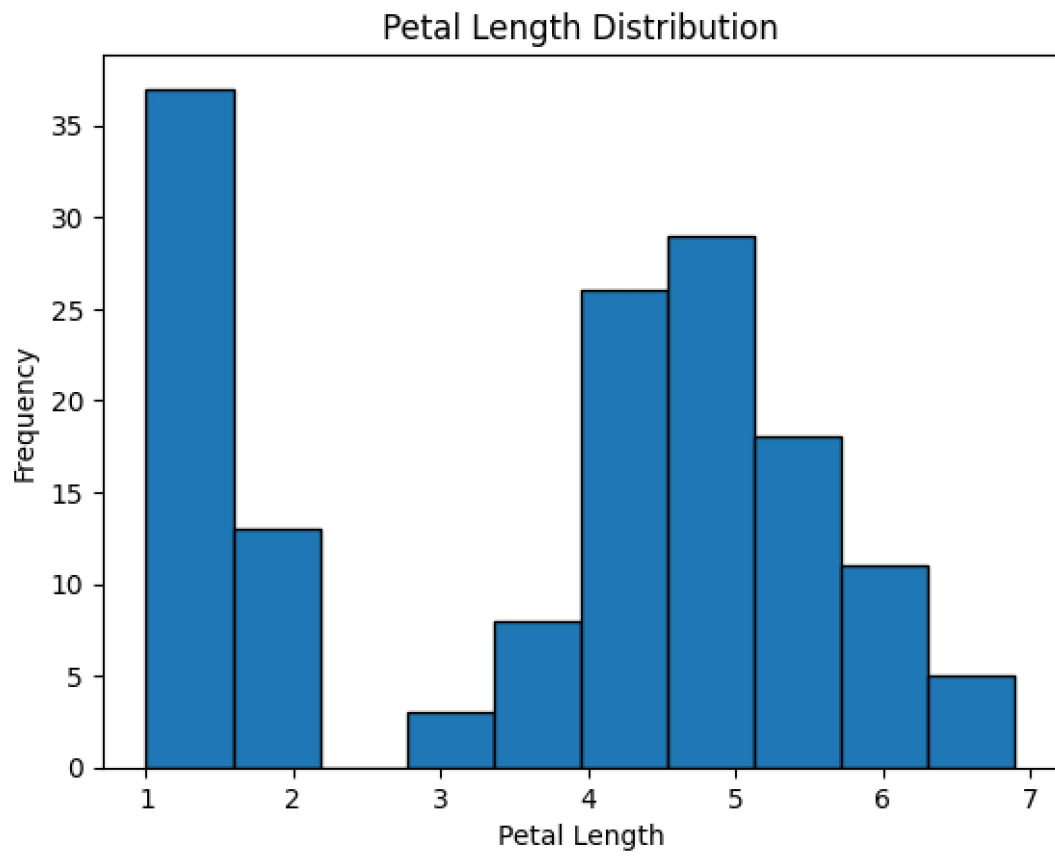
# Line Plot: Sepal Length vs Index
plt.plot(df['sepal_length'], marker='o')
plt.title('Sepal Length Distribution')
plt.xlabel('Index')
plt.ylabel('Sepal Length')
plt.grid(True)
plt.show()

# Histogram: Petal Length Distribution
plt.hist(df['petal_length'], bins=10, edgecolor='black')
plt.title('Petal Length Distribution')
plt.xlabel('Petal Length')
plt.ylabel('Frequency')
plt.show()

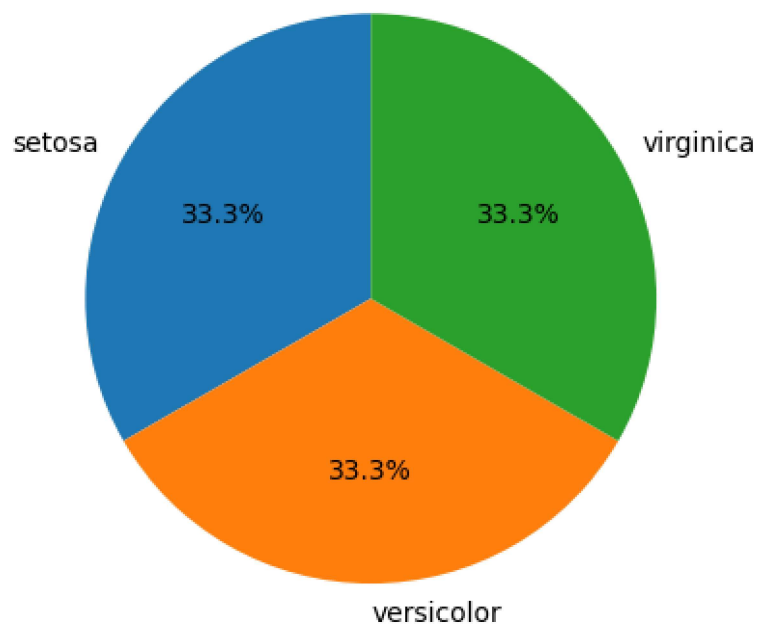
# Pie Chart: Species Count
species_counts = df['species'].value_counts()
plt.pie(species_counts, labels=species_counts.index, autopct='%1.1f%%',
        ↪startangle=90)
plt.title('Species Distribution')
plt.show()

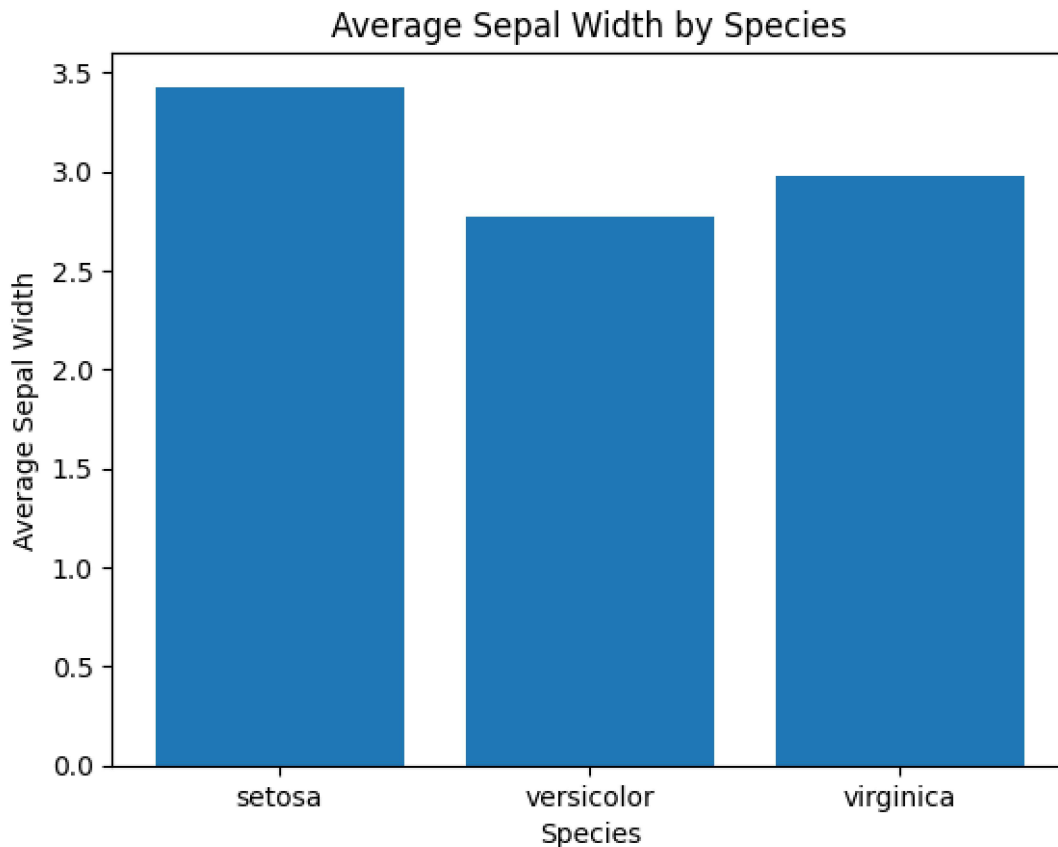
# Bar Plot: Average Sepal Width by Species
avg_sepal_width = df.groupby('species')['sepal_width'].mean()
plt.bar(avg_sepal_width.index, avg_sepal_width.values)
plt.title('Average Sepal Width by Species')
plt.xlabel('Species')
plt.ylabel('Average Sepal Width')
plt.show()
```





Species Distribution





Seaborn Overview: Integrates well with Pandas DataFrames for easy data manipulation and visualization. Provides built-in themes and color palettes for creating visually appealing plots with minimal effort.

- Built on top of Matplotlib
- High-level interface for statistical plots
- Excellent for visualizing relationships between data

```
[ ]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Sample DataFrame (Iris dataset)

df = pd.DataFrame(data)

# Bar Plot: Average Sepal Length by Species
sns.barplot(x='species', y='sepal_length', data=df)
plt.title('Average Sepal Length by Species')
plt.show()
plt.savefig('chart_name.png')
```

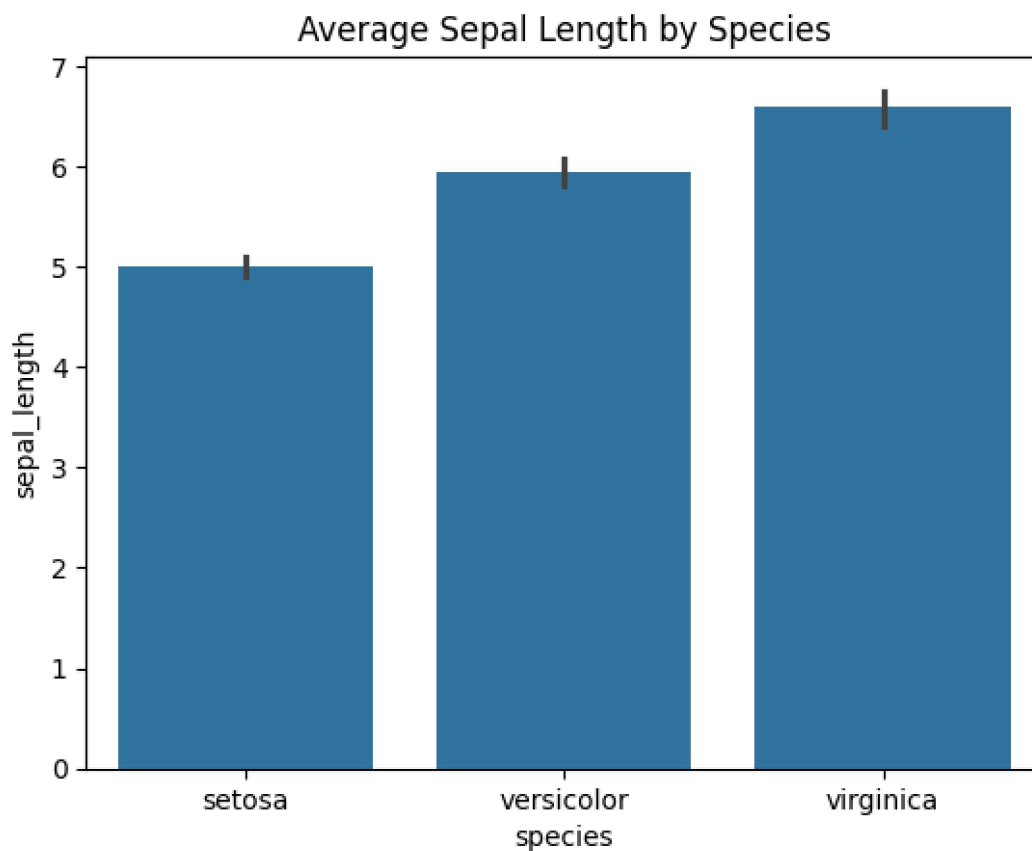
```

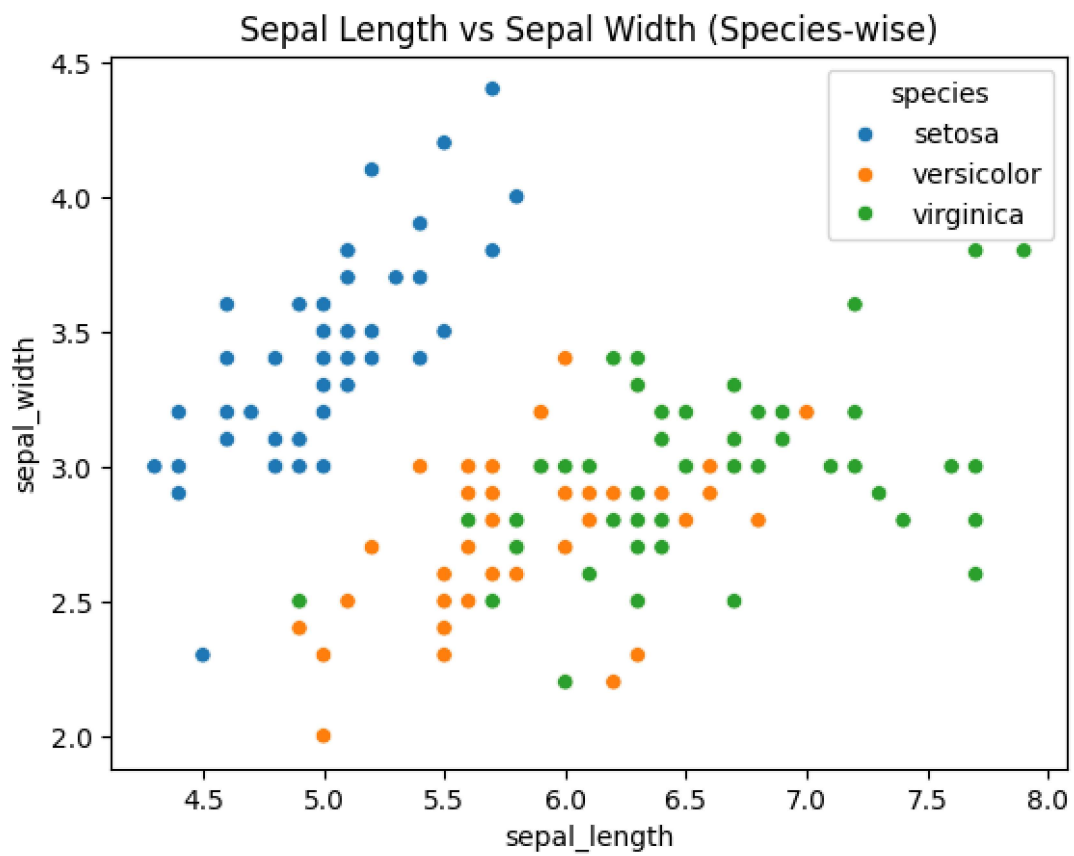
# Scatter Plot: Sepal Length vs Sepal Width
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species', data=df)
plt.title('Sepal Length vs Sepal Width (Species-wise)')
plt.show()
plt.savefig('chart_name.png')

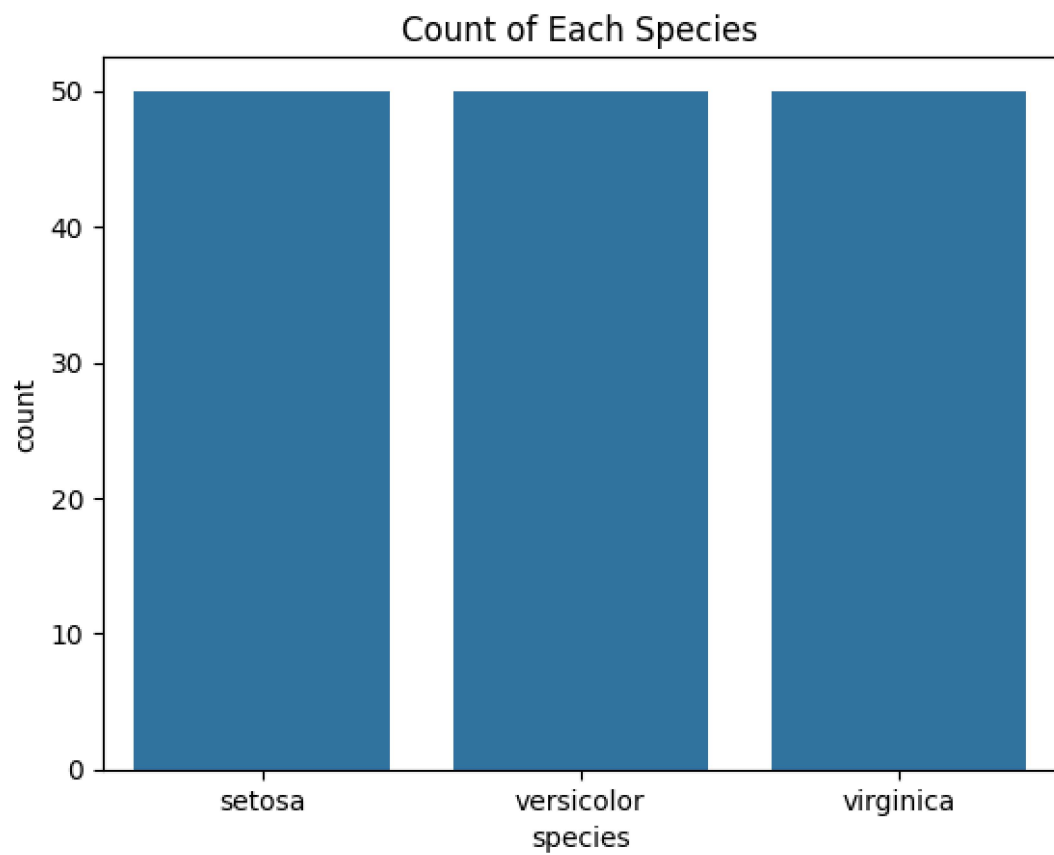
# Count Plot: Count of Each Species
sns.countplot(x='species', data=df)
plt.title('Count of Each Species')
plt.show()
plt.savefig('chart_name.png')

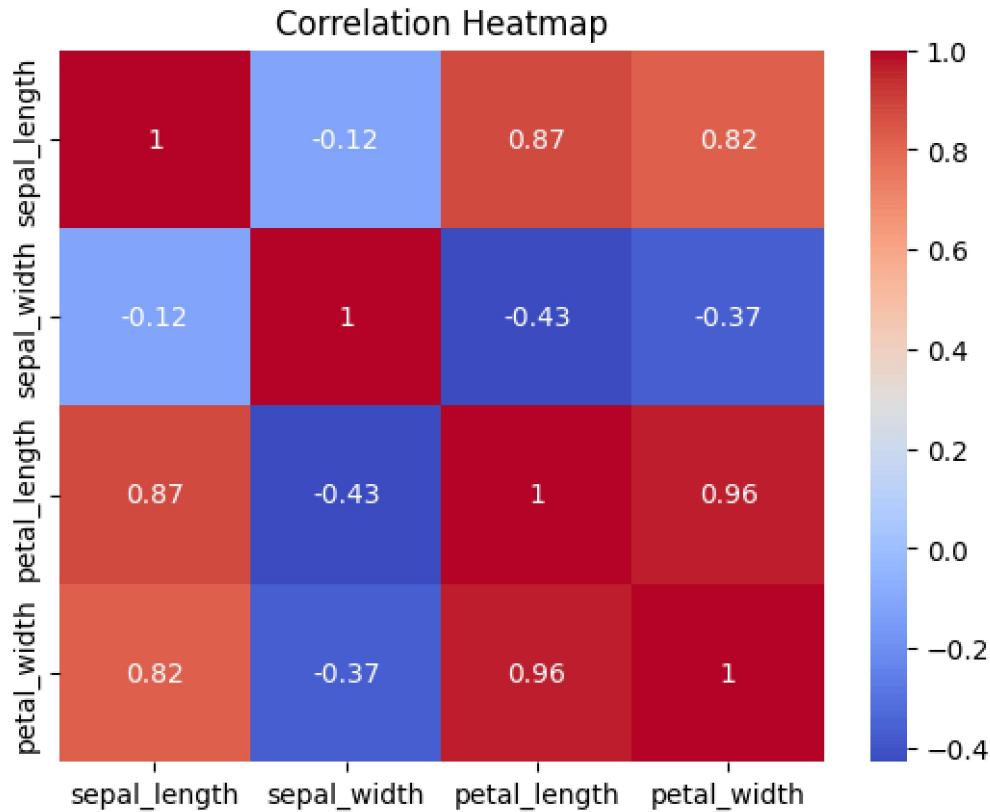
# Heatmap: Correlation Heatmap
correlation = df.drop('species', axis=1).corr()
sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
plt.savefig('chart_name.png')

```









<Figure size 640x480 with 0 Axes>

3# Comparison Between Matplotlib and Seaborn:- * Ease of Use and Customization: Matplotlib provides complete control over plot elements but requires more coding effort. Seaborn, on the other hand, simplifies complex statistical plots with minimal code and offers visually appealing default styles. * Interactivity and Performance: Matplotlib is primarily static with limited interactivity, while Seaborn can be integrated with interactive libraries like Plotly. Both perform well with moderate datasets but struggle with extremely large datasets. * Use Cases: Matplotlib is ideal for detailed, fully customizable visualizations, while Seaborn excels in quick, aesthetically pleasing statistical plots.

1 4.Resources

Matplotlib: https://matplotlib.org/stable/users/explain/quick_start.html#quick-start

Seaborn <https://seaborn.pydata.org/tutorial/introduction.html>