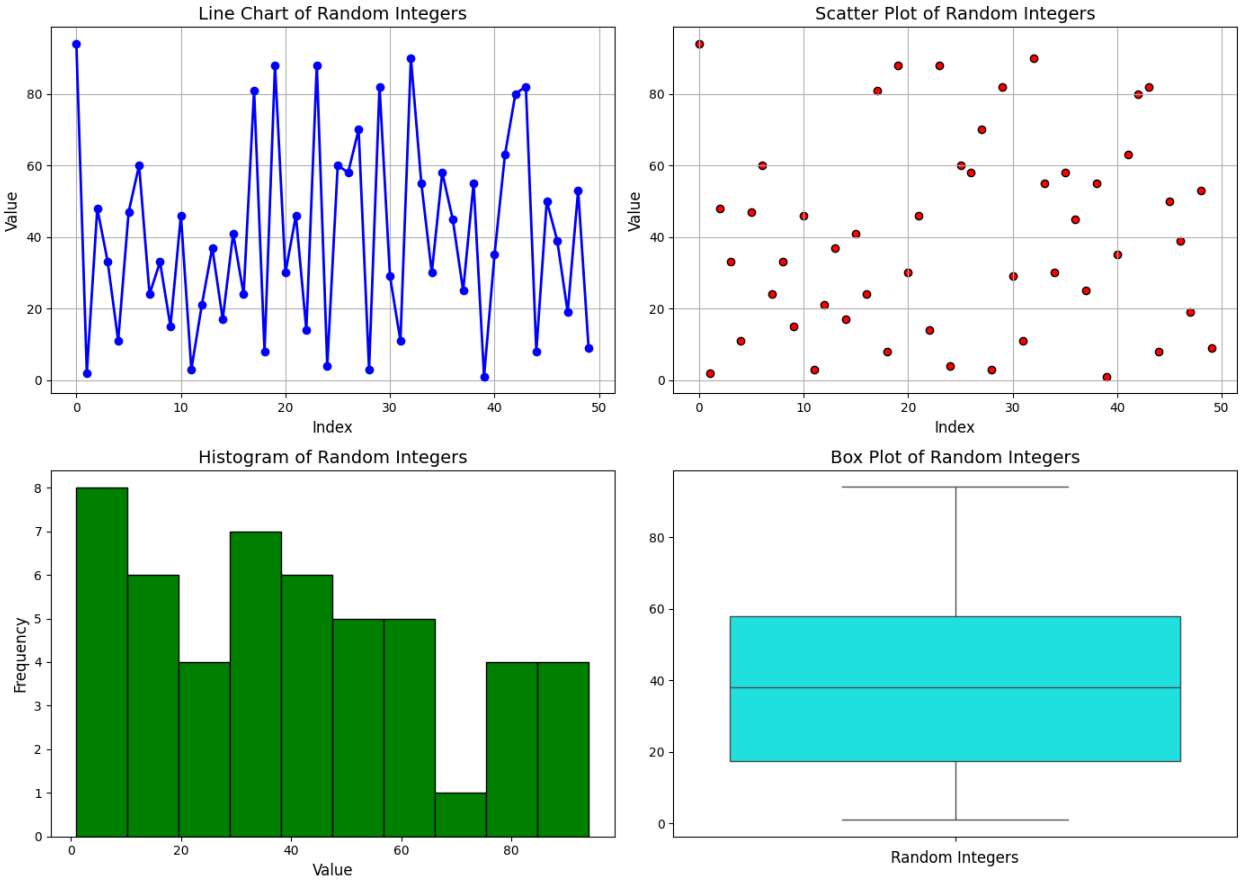
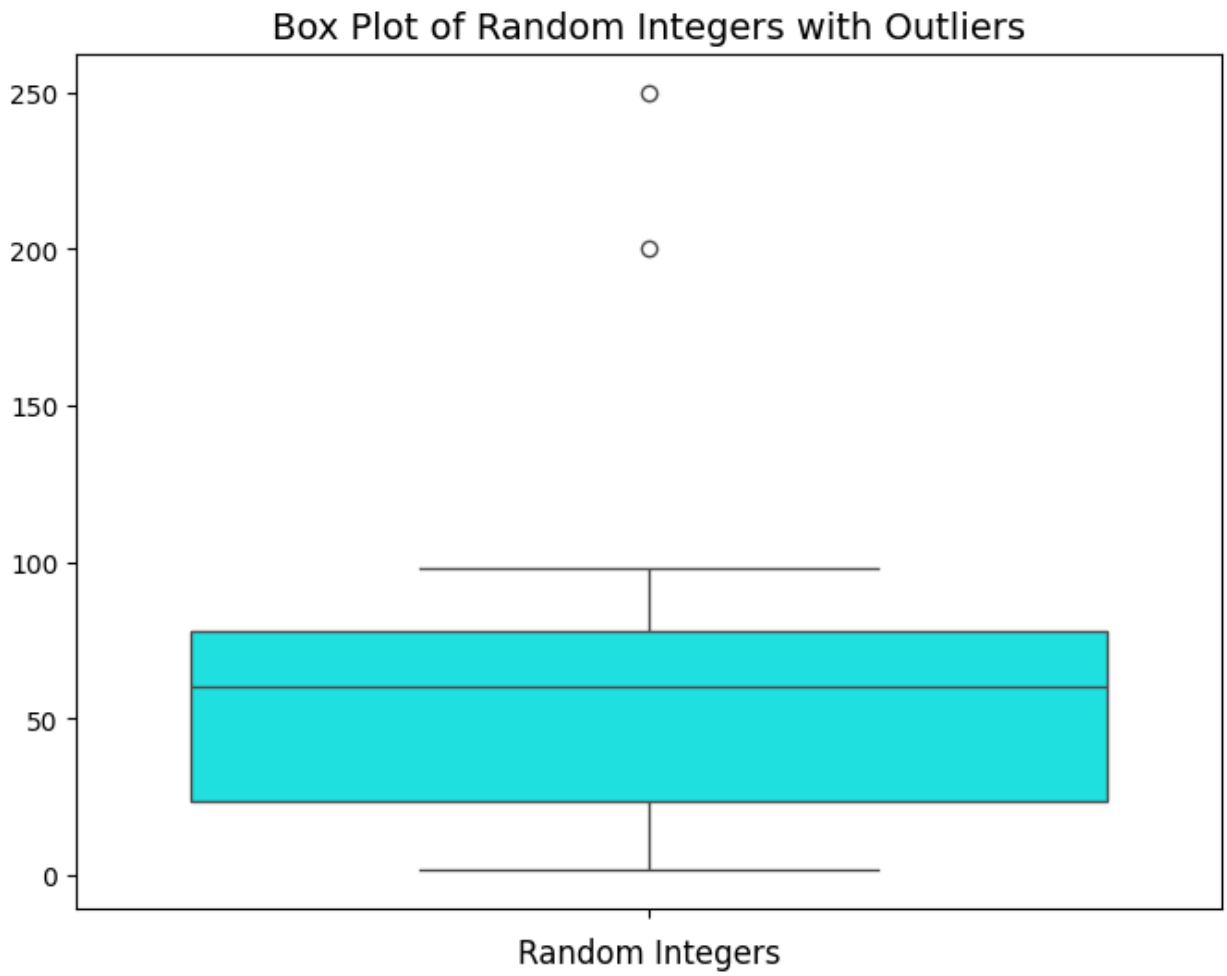
```python
#ASSIGNMENT NUMBER : 4

#SET :A 1)

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Generate a random array of 50 integers between 1 and 100
data = np.random.randint(1, 100, size=50)

# Set up the subplots (2 rows, 2 columns)
fig, axs = plt.subplots(2, 2, figsize=(14, 10))

# Line chart
axs[0, 0].plot(data, color='blue', marker='o', linestyle='-',
linewidth=2, markersize=6)
axs[0, 0].set_title('Line Chart of Random Integers', fontsize=14)
axs[0, 0].set_xlabel('Index', fontsize=12)
axs[0, 0].set_ylabel('Value', fontsize=12)
axs[0, 0].grid(True)

# Scatter plot
axs[0, 1].scatter(range(len(data)), data, color='red',
edgecolor='black')
axs[0, 1].set_title('Scatter Plot of Random Integers', fontsize=14)
axs[0, 1].set_xlabel('Index', fontsize=12)
axs[0, 1].set_ylabel('Value', fontsize=12)
axs[0, 1].grid(True)

# Histogram
axs[1, 0].hist(data, bins=10, color='green', edgecolor='black')
axs[1, 0].set_title('Histogram of Random Integers', fontsize=14)
axs[1, 0].set_xlabel('Value', fontsize=12)
axs[1, 0].set_ylabel('Frequency', fontsize=12)

# Box plot
sns.boxplot(data=data, ax=axs[1, 1], color='cyan')
axs[1, 1].set_title('Box Plot of Random Integers', fontsize=14)
axs[1, 1].set_xlabel('Random Integers', fontsize=12)

# Adjust the layout to prevent overlapping
plt.tight_layout()

# Show the plots
plt.show()
```

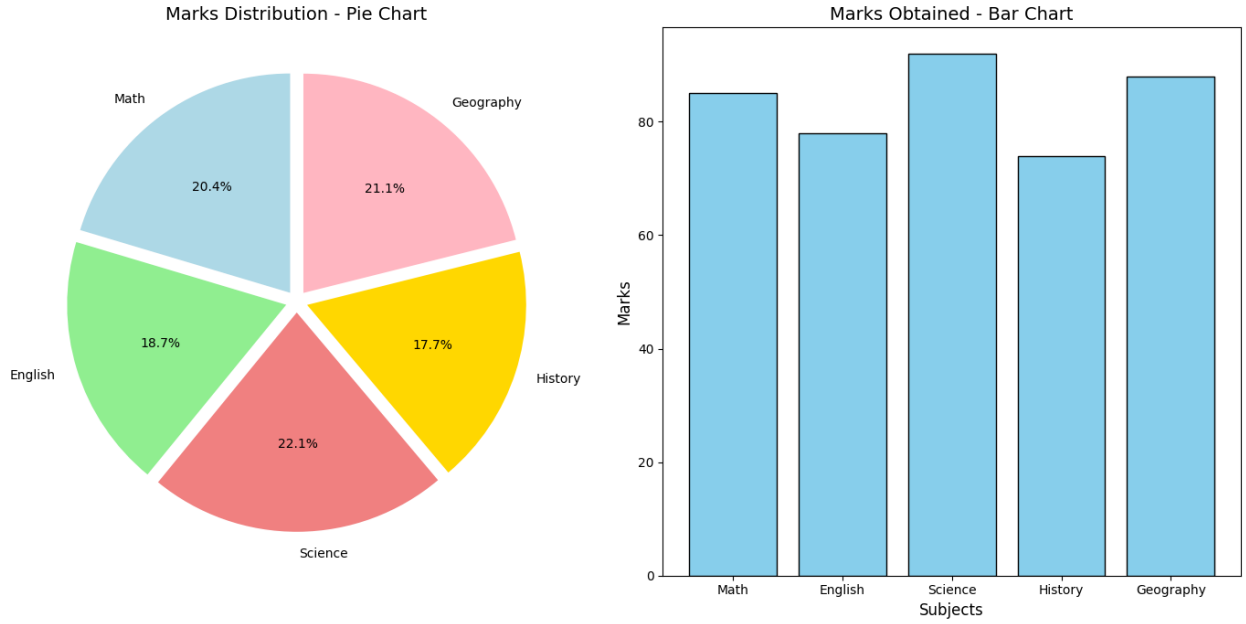



```
2)

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Generate a random array of 50 integers between 1 and 100
data = np.random.randint(1, 100, size=50)

# Add two outliers
data = np.append(data, [200, 250])

# Set up the plot
plt.figure(figsize=(8, 6))

# Box plot
sns.boxplot(data=data, color='cyan')
plt.title('Box Plot of Random Integers with Outliers', fontsize=14)
plt.xlabel('Random Integers', fontsize=12)

# Display the plot
plt.show()
```




```
3)

import matplotlib.pyplot as plt

# List of subject names
subjects = ['Math', 'English', 'Science', 'History', 'Geography']

# List of marks obtained in each subject
marks = [85, 78, 92, 74, 88]

# Set up the figure with 1 row and 2 columns for pie chart and bar
chart
fig, axs = plt.subplots(1, 2, figsize=(14, 7))

# Pie chart
axs[0].pie(marks, labels=subjects, autopct='%1.1f%%',
colors=['lightblue', 'lightgreen', 'lightcoral', 'gold', 'lightpink'],
startangle=90, explode=[0.05]*len(subjects))
axs[0].set_title('Marks Distribution - Pie Chart', fontsize=14)
```

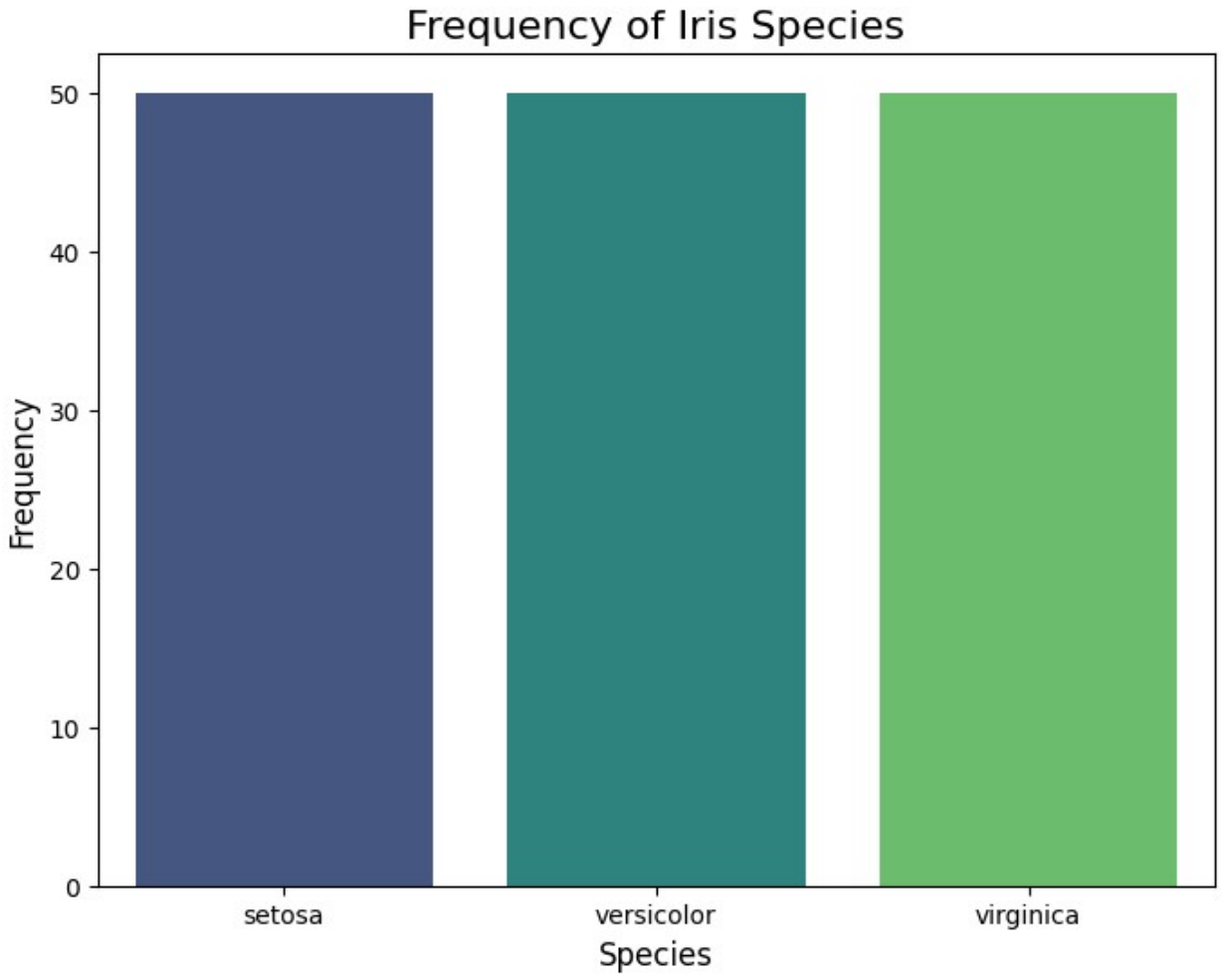
```
# Bar chart
axs[1].bar(subjects, marks, color='skyblue', edgecolor='black')
axs[1].set_title('Marks Obtained - Bar Chart', fontsize=14)
axs[1].set_xlabel('Subjects', fontsize=12)
axs[1].set_ylabel('Marks', fontsize=12)

# Adjust layout to prevent overlapping
plt.tight_layout()

# Show the charts
plt.show()
```




```
4)

import seaborn as sns
import matplotlib.pyplot as plt

# Load the Iris dataset from seaborn
iris = sns.load_dataset('iris')

# Count the occurrences of each species
species_counts = iris['species'].value_counts()

# Create a bar plot to display the frequency of each species
plt.figure(figsize=(8, 6))
sns.barplot(x=species_counts.index, y=species_counts.values,
palette='viridis')

# Adding labels and title
plt.title('Frequency of Iris Species', fontsize=16)
```

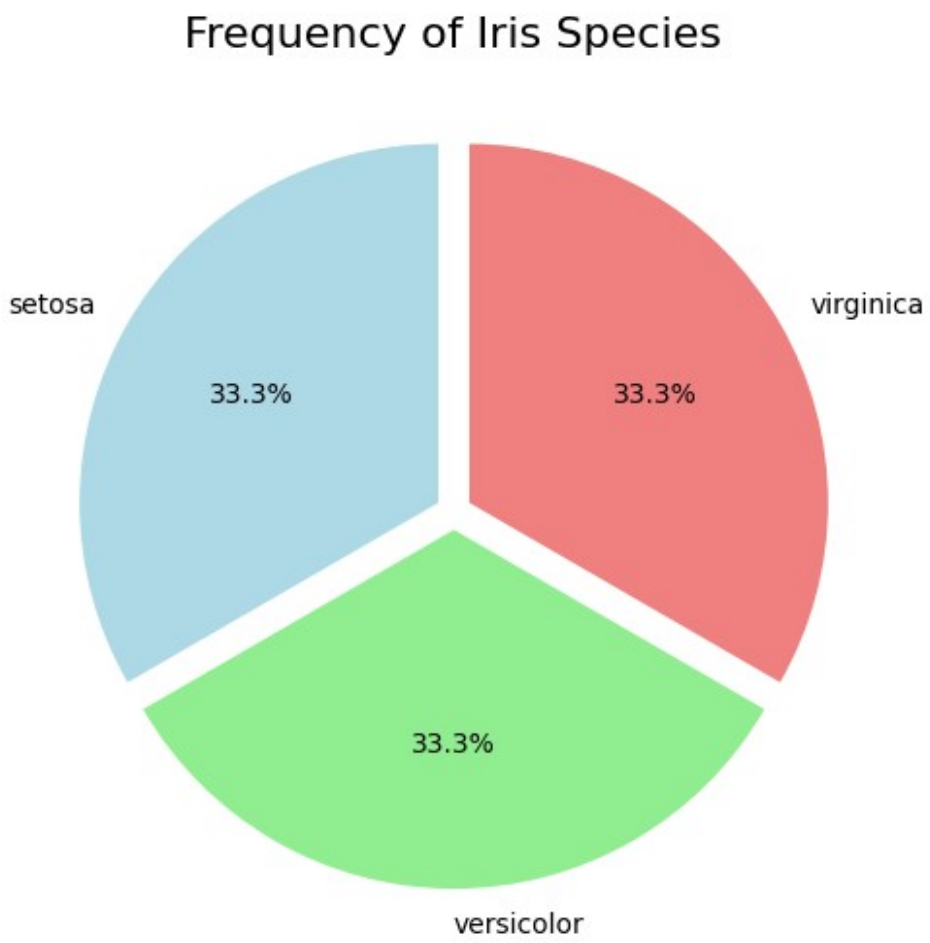
```
plt.xlabel('Species', fontsize=12)
plt.ylabel('Frequency', fontsize=12)

# Show the plot
plt.show()
```

```
C:\Users\ecs\AppData\Local\Temp\ipykernel_7856\2228815281.py:12:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(x=species_counts.index, y=species_counts.values,
palette='viridis')
```

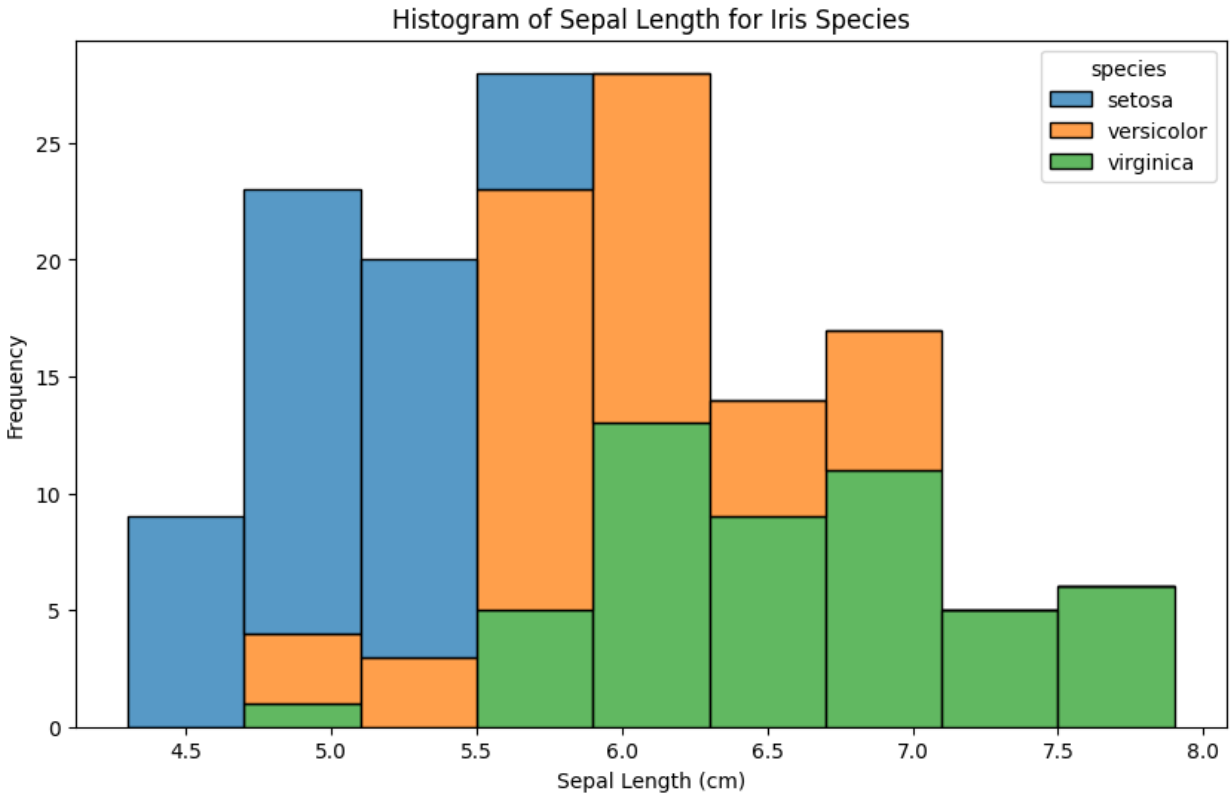



```
5)
```

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

# Load the Iris dataset from seaborn
iris = sns.load_dataset('iris')

# Count the occurrences of each species
species_counts = iris['species'].value_counts()

# Create a pie chart to display the frequency of each species
plt.figure(figsize=(8, 6))
plt.pie(species_counts, labels=species_counts.index, autopct='%1.1f%
%', colors=['lightblue', 'lightgreen', 'lightcoral'], startangle=90,
explode=[0.05, 0.05, 0.05])

# Adding title
plt.title('Frequency of Iris Species', fontsize=16)

# Show the plot
plt.show()
```

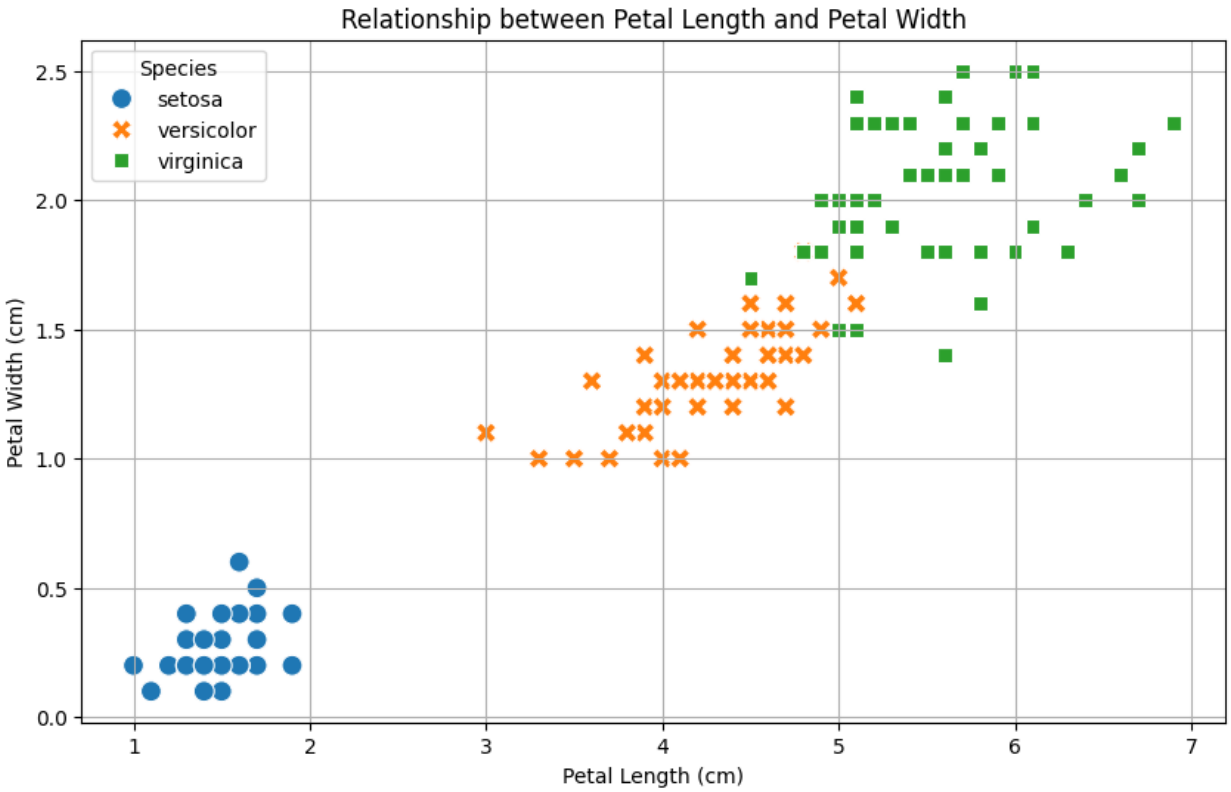


```
6)

# Importing necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Load the iris dataset
iris = sns.load_dataset('iris')

# Plotting histograms for the three species
plt.figure(figsize=(10, 6))

# Create histograms for each species' sepal length
sns.histplot(data=iris, x='sepal_length', hue='species',
multiple='stack')

# Add labels and title
plt.title('Histogram of Sepal Length for Iris Species')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Frequency')

# Show the plot
plt.show()
```



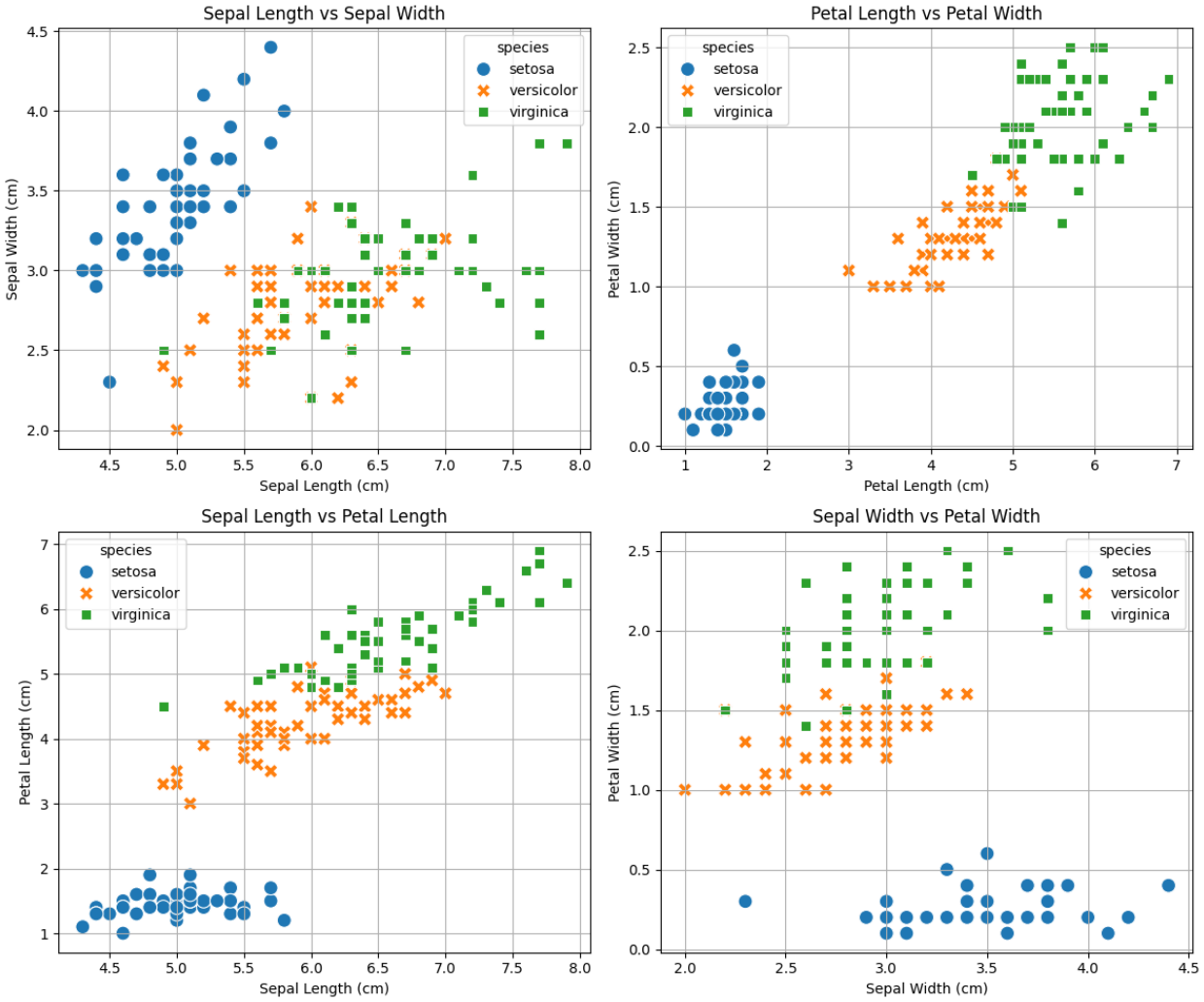

```
#SET B: 1)
```

```
# Importing necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Load the iris dataset
iris = sns.load_dataset('iris')

# Create a scatter plot to show the relationship between petal length
and petal width
plt.figure(figsize=(10, 6))
sns.scatterplot(data=iris, x='petal_length', y='petal_width',
hue='species', style='species', s=100)

# Add labels and title
plt.title('Relationship between Petal Length and Petal Width')
plt.xlabel('Petal Length (cm)')
plt.ylabel('Petal Width (cm)')

# Show the plot
plt.legend(title='Species')
plt.grid(True)
plt.show()
```




```
#2)
```

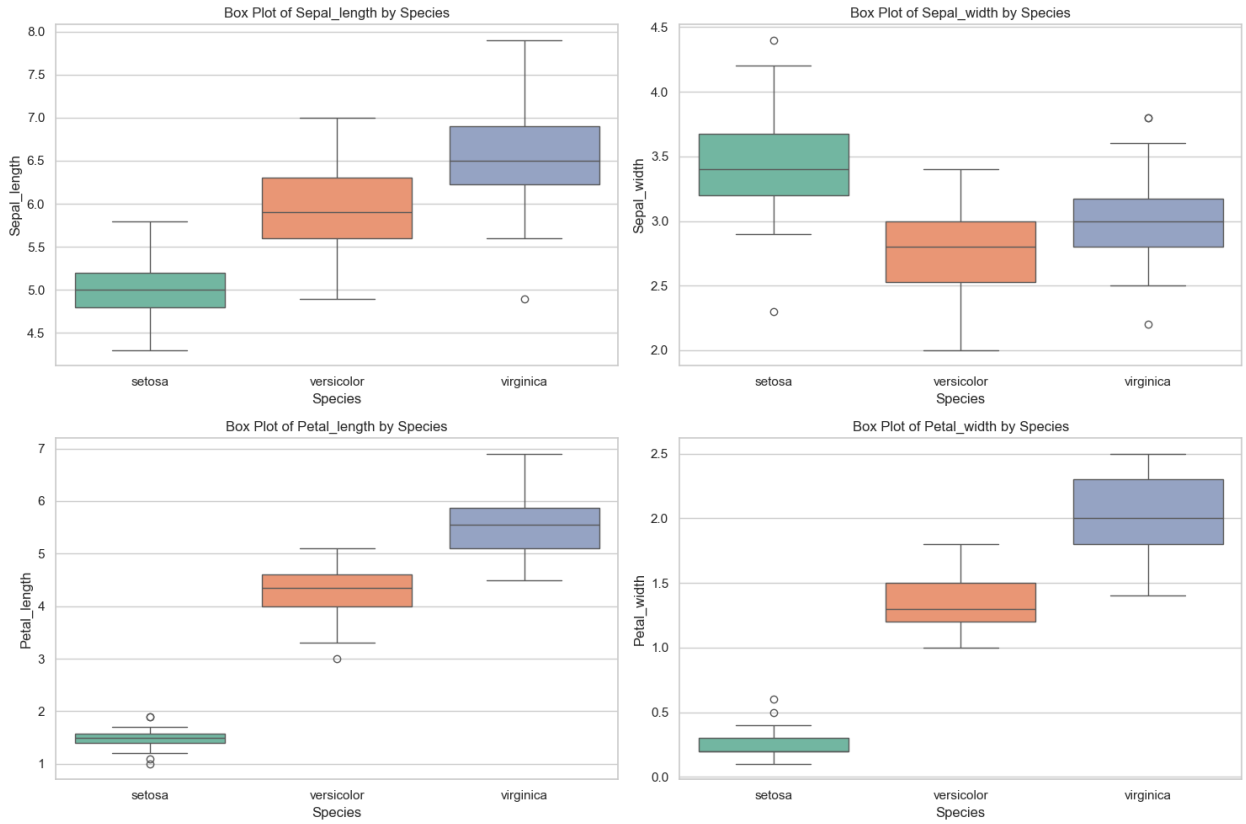
```python
# Importing necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Load the iris dataset
iris = sns.load_dataset('iris')

# Create a scatter plot for Sepal Length vs Sepal Width
plt.figure(figsize=(12, 10))

plt.subplot(2, 2, 1)
sns.scatterplot(data=iris, x='sepal_length', y='sepal_width',
hue='species', style='species', s=100)
plt.title('Sepal Length vs Sepal Width')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Sepal Width (cm)')
plt.grid(True)

# Create a scatter plot for Petal Length vs Petal Width
plt.subplot(2, 2, 2)
sns.scatterplot(data=iris, x='petal_length', y='petal_width',
hue='species', style='species', s=100)
plt.title('Petal Length vs Petal Width')
plt.xlabel('Petal Length (cm)')
plt.ylabel('Petal Width (cm)')
plt.grid(True)

# Create a scatter plot for Sepal Length vs Petal Length
plt.subplot(2, 2, 3)
sns.scatterplot(data=iris, x='sepal_length', y='petal_length',
hue='species', style='species', s=100)
plt.title('Sepal Length vs Petal Length')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Petal Length (cm)')
plt.grid(True)

# Create a scatter plot for Sepal Width vs Petal Width
plt.subplot(2, 2, 4)
sns.scatterplot(data=iris, x='sepal_width', y='petal_width',
hue='species', style='species', s=100)
plt.title('Sepal Width vs Petal Width')
plt.xlabel('Sepal Width (cm)')
plt.ylabel('Petal Width (cm)')
plt.grid(True)

# Adjust layout and show the plot
plt.tight_layout()
plt.show()
```




```
3)

# Importing necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Load the iris dataset
iris = sns.load_dataset('iris')

# Set the style of the visualization
sns.set(style="whitegrid")

# Create box plots for each feature
features = ['sepal_length', 'sepal_width', 'petal_length',
'petal_width']

plt.figure(figsize=(15, 10))

# Iterate over features to create box plots
```

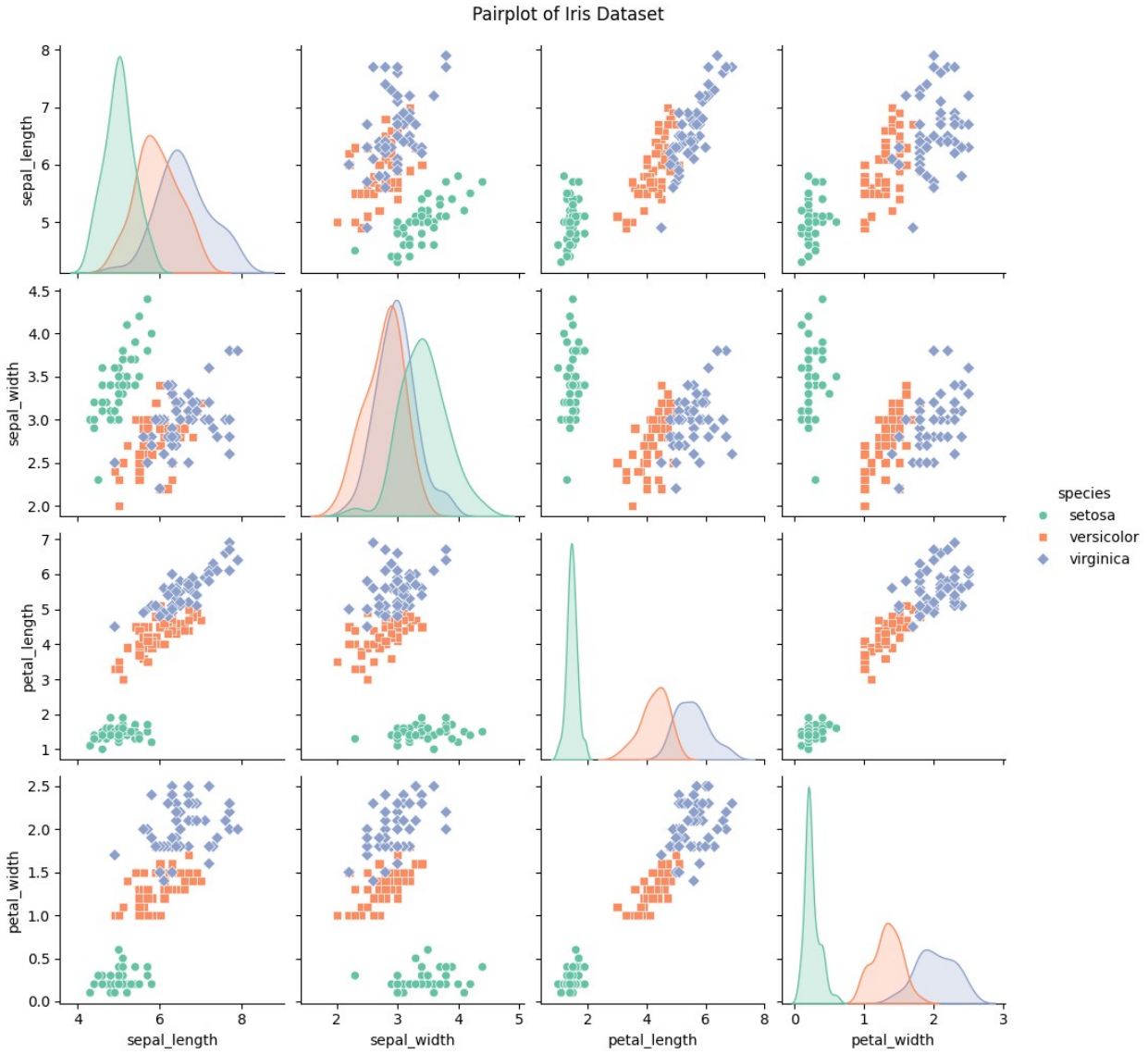
```python
for i, feature in enumerate(features):
    plt.subplot(2, 2, i + 1)  # Create a grid of 2x2 subplots
    sns.boxplot(data=iris, x='species', y=feature, palette='Set2')
    plt.title(f'Box Plot of {feature.capitalize()} by Species')
    plt.xlabel('Species')
    plt.ylabel(feature.capitalize())

# Adjust layout and show the plots
plt.tight_layout()
plt.show()
```

```
C:\Users\ecs\AppData\Local\Temp\ipykernel_7856\2019540408.py:19:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=iris, x='species', y=feature, palette='Set2')
C:\Users\ecs\AppData\Local\Temp\ipykernel_7856\2019540408.py:19:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=iris, x='species', y=feature, palette='Set2')
C:\Users\ecs\AppData\Local\Temp\ipykernel_7856\2019540408.py:19:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=iris, x='species', y=feature, palette='Set2')
C:\Users\ecs\AppData\Local\Temp\ipykernel_7856\2019540408.py:19:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(data=iris, x='species', y=feature, palette='Set2')
```




```
#SET C: 1)

# Importing necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Load the iris dataset
iris = sns.load_dataset('iris')

# Create a pairplot
plt.figure(figsize=(12, 12))
pair_plot = sns.pairplot(iris, hue='species', markers=["o", "s", "D"],
palette='Set2')

# Add a title
plt.suptitle('Pairplot of Iris Dataset', y=1.02)  # Adjust y for
better title placement

# Show the plot
plt.show()
```

```
<Figure size 1200x1200 with 0 Axes>
```


Pairplot of Iris Dataset
sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica