In [1]:

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

# Load the Iris dataset
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
column_names = ['sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'c
iris = pd.read_csv(url, names=column_names)

# Display first few rows
print(iris.head())
```

```
   sepal_length  sepal_width  petal_length  petal_width        class
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
```

In [2]:

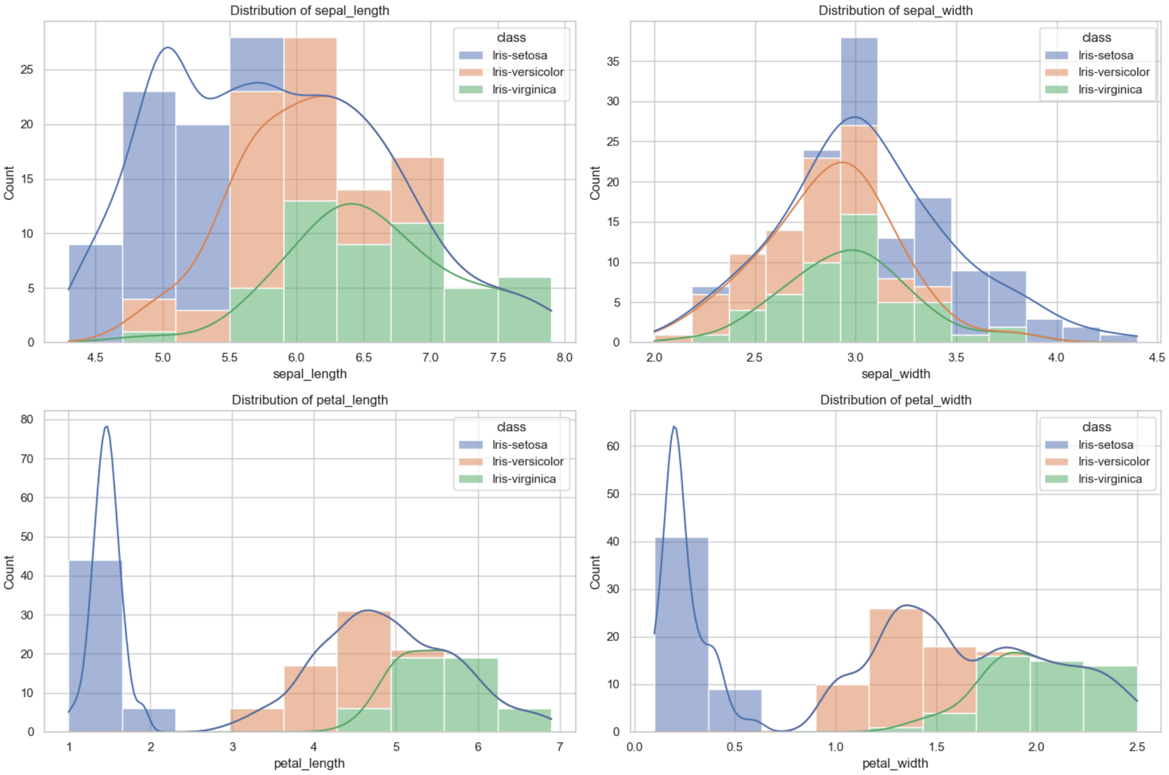
```
print("\nFeature Types:")
print(iris.dtypes)

print("\nFeature Summary:")
print(iris.describe())

print("\nClass Distribution:")
print(iris['class'].value_counts())
```

```
Feature Types:
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
class            object
dtype: object

Feature Summary:
       sepal_length  sepal_width  petal_length  petal_width
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

Class Distribution:
class
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
Name: count, dtype: int64
```

In [5]:

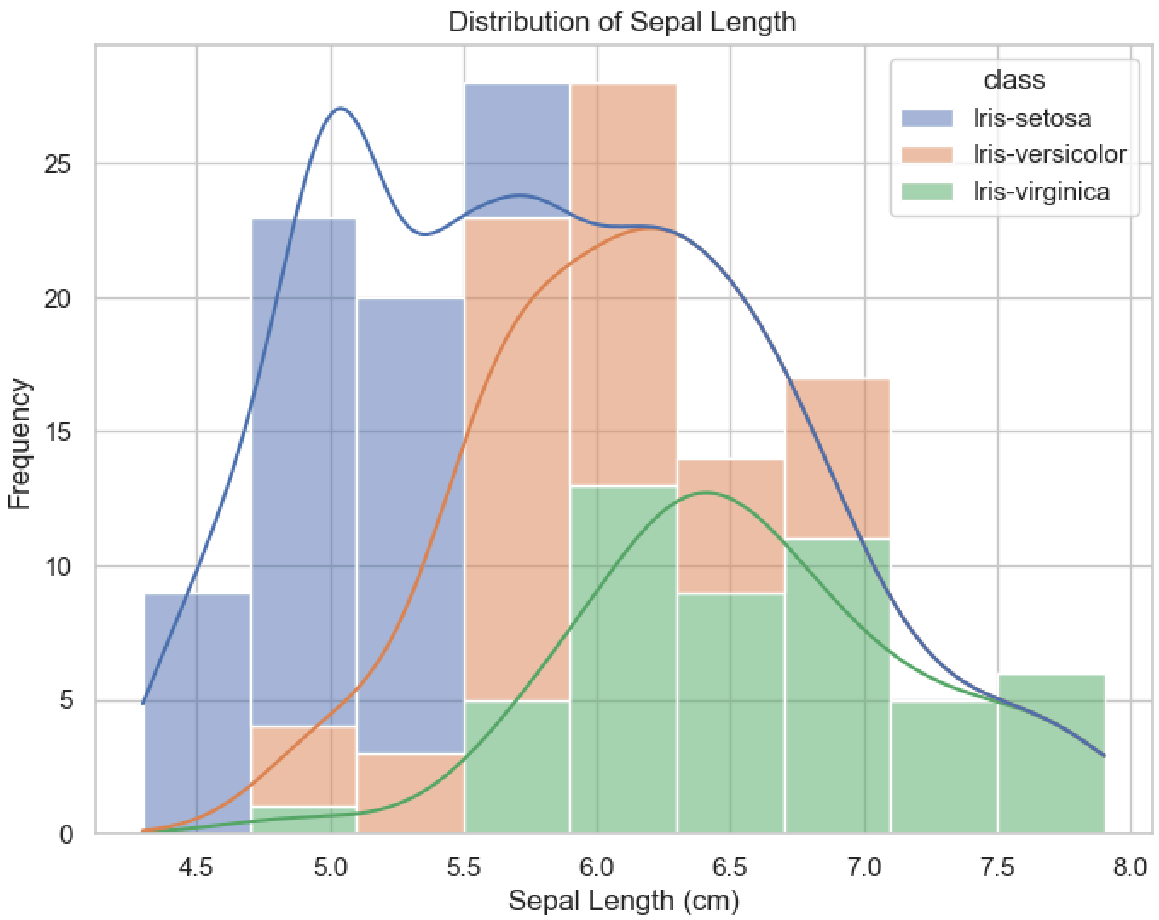
```
# Set style
sns.set(style="whitegrid")

# Create histograms
plt.figure(figsize=(15, 10))
```

```python
for i, feature in enumerate(column_names[:-1]):
    plt.subplot(2, 2, i+1)
    sns.histplot(data=iris, x=feature, kde=True, hue='class', multiple='stack')
    plt.title(f'Distribution of {feature}')
plt.tight_layout()
plt.show()
```

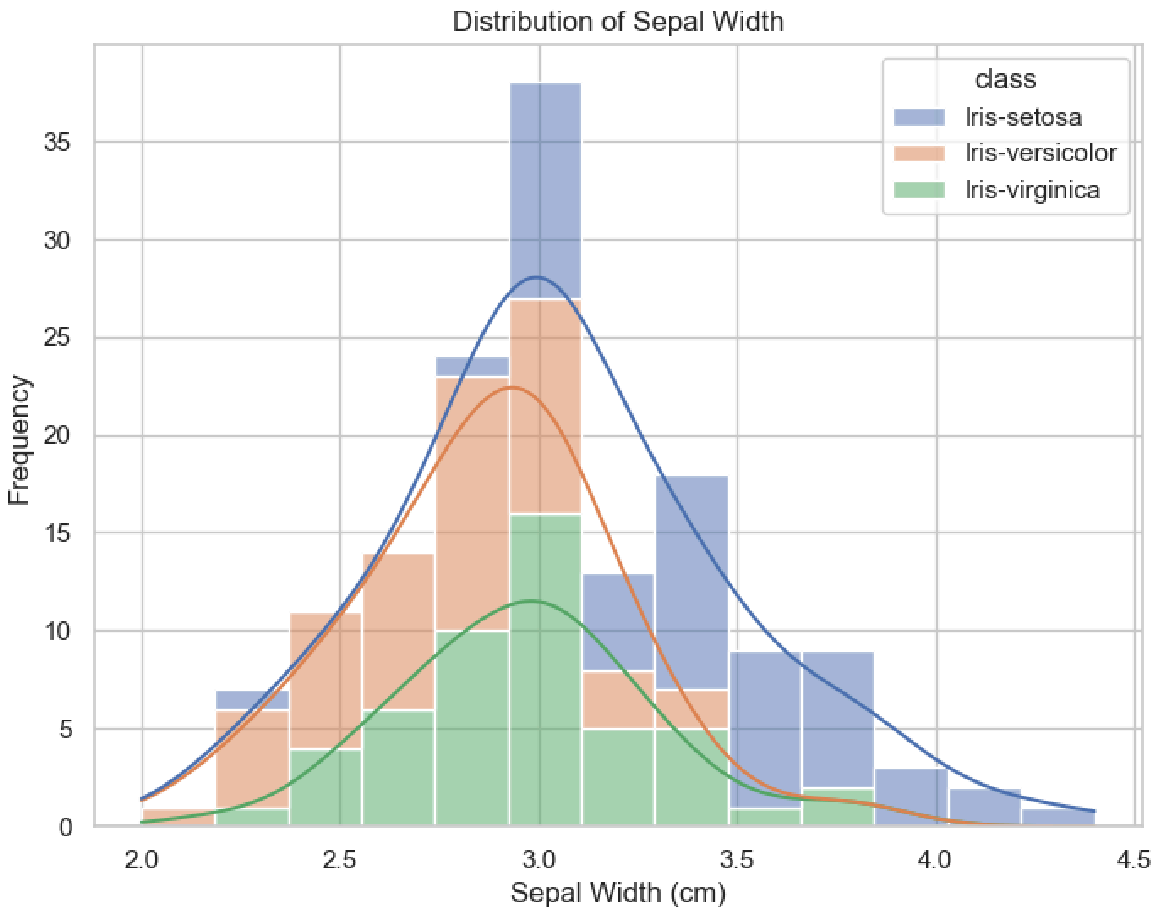


In [6]:

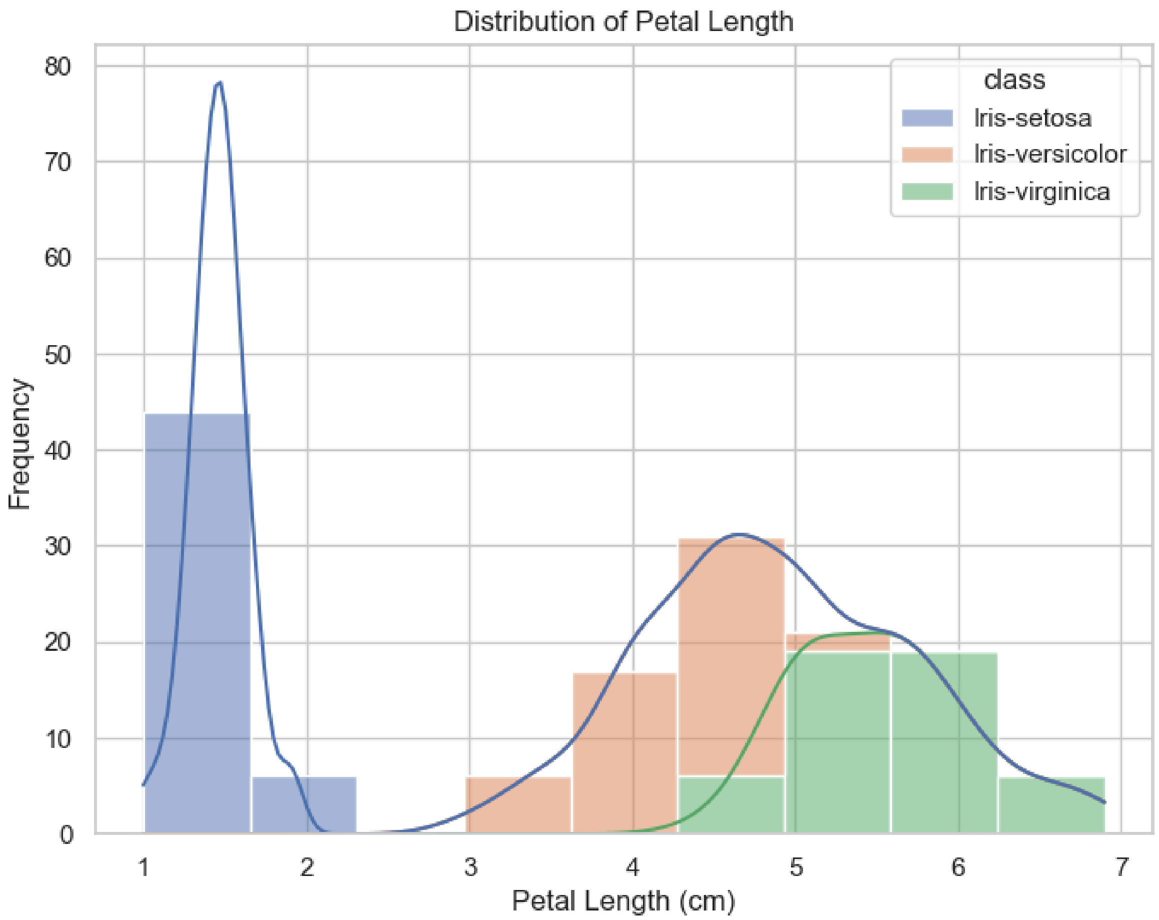
```python
plt.figure(figsize=(8, 6))
sns.histplot(data=iris, x='sepal_length', kde=True, hue='class', multiple='stack
plt.title('Distribution of Sepal Length')
plt.xlabel('Sepal Length (cm)')
plt.ylabel('Frequency')
plt.show()
```



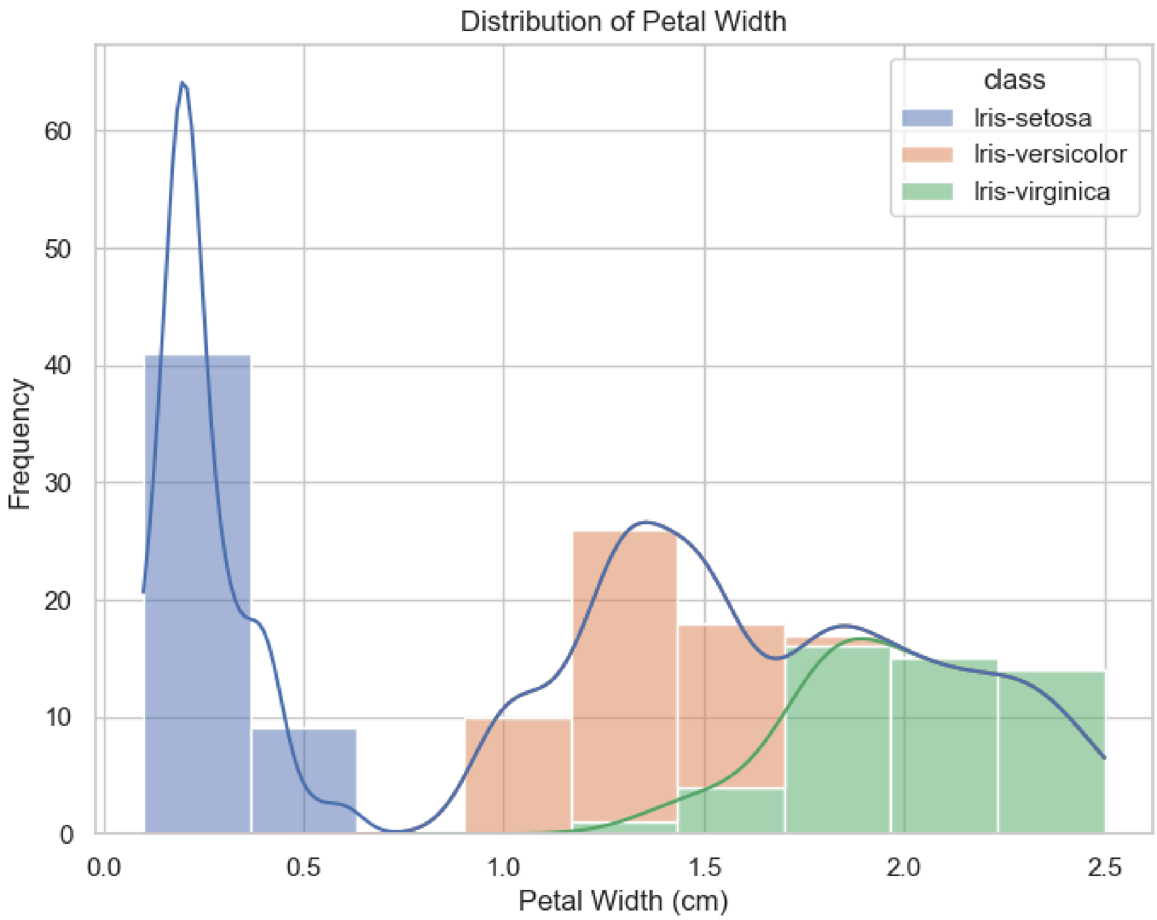
In [7]:
```
plt.figure(figsize=(8, 6))
sns.histplot(data=iris, x='sepal_width', kde=True, hue='class', multiple='stack'
plt.title('Distribution of Sepal Width')
plt.xlabel('Sepal Width (cm)')
plt.ylabel('Frequency')
plt.show()
```



In [8]:
```
plt.figure(figsize=(8, 6))
sns.histplot(data=iris, x='petal_length', kde=True, hue='class', multiple='stack
plt.title('Distribution of Petal Length')
plt.xlabel('Petal Length (cm)')
plt.ylabel('Frequency')
plt.show()
```

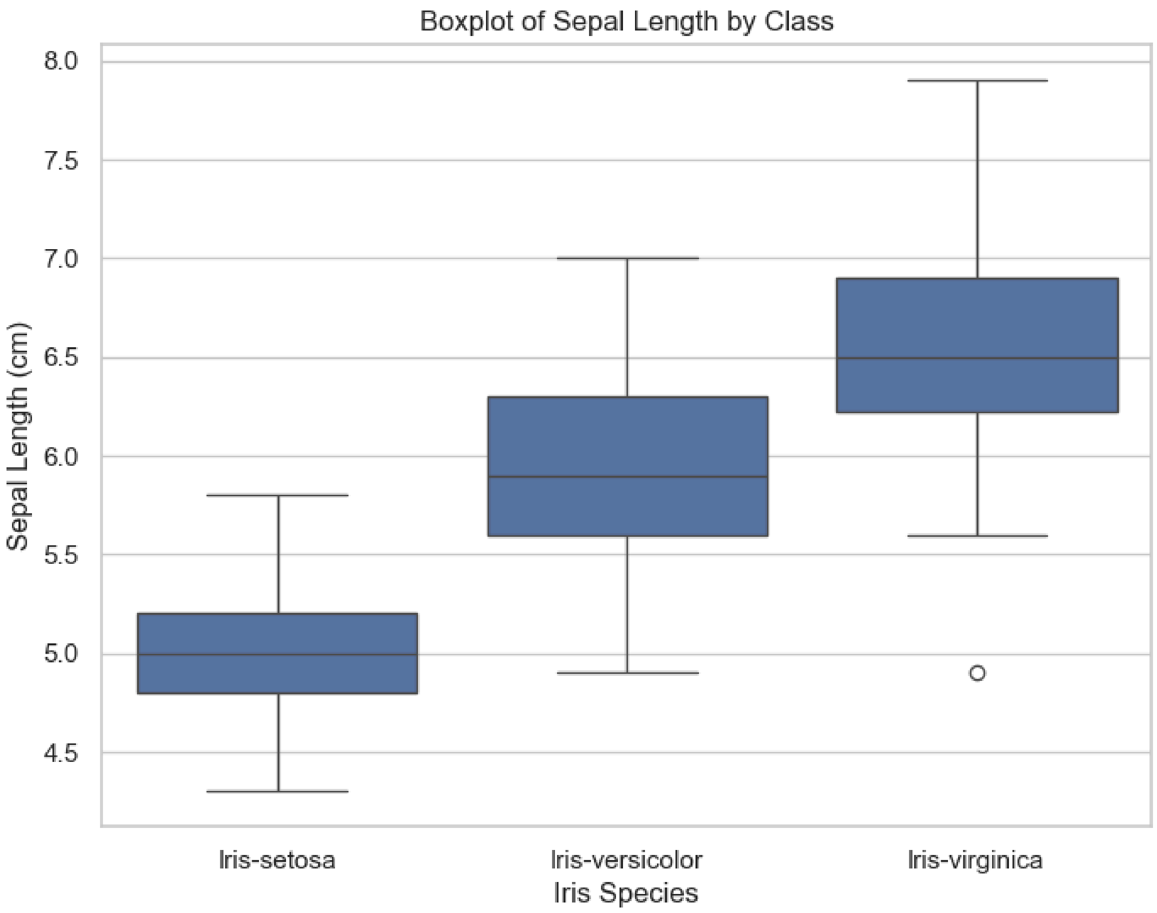


```
In [9]: plt.figure(figsize=(8, 6))
        sns.histplot(data=iris, x='petal_width', kde=True, hue='class', multiple='stack'
        plt.title('Distribution of Petal Width')
        plt.xlabel('Petal Width (cm)')
        plt.ylabel('Frequency')
        plt.show()
```

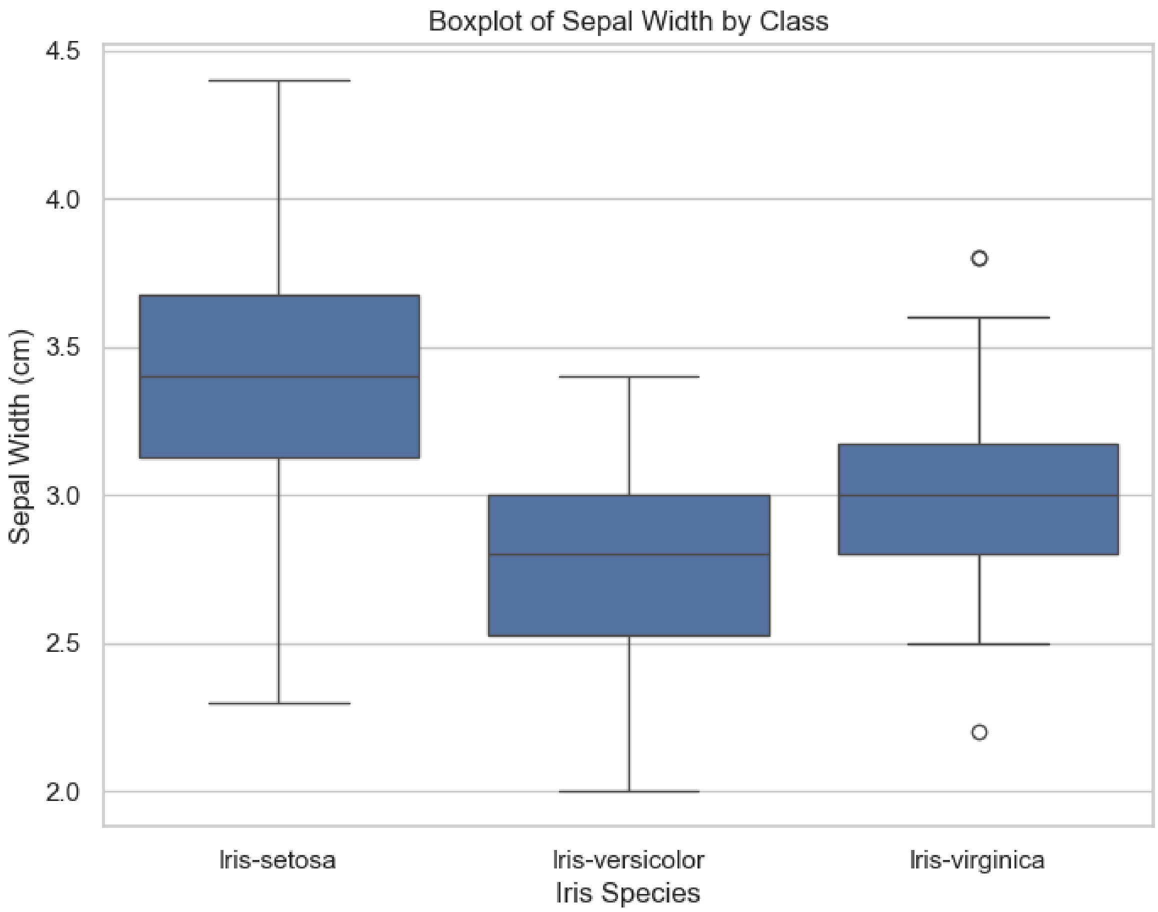


In [10]:

```
plt.figure(figsize=(8, 6))
sns.boxplot(data=iris, x='class', y='sepal_length')
plt.title('Boxplot of Sepal Length by Class')
plt.xlabel('Iris Species')
plt.ylabel('Sepal Length (cm)')
plt.show()
```

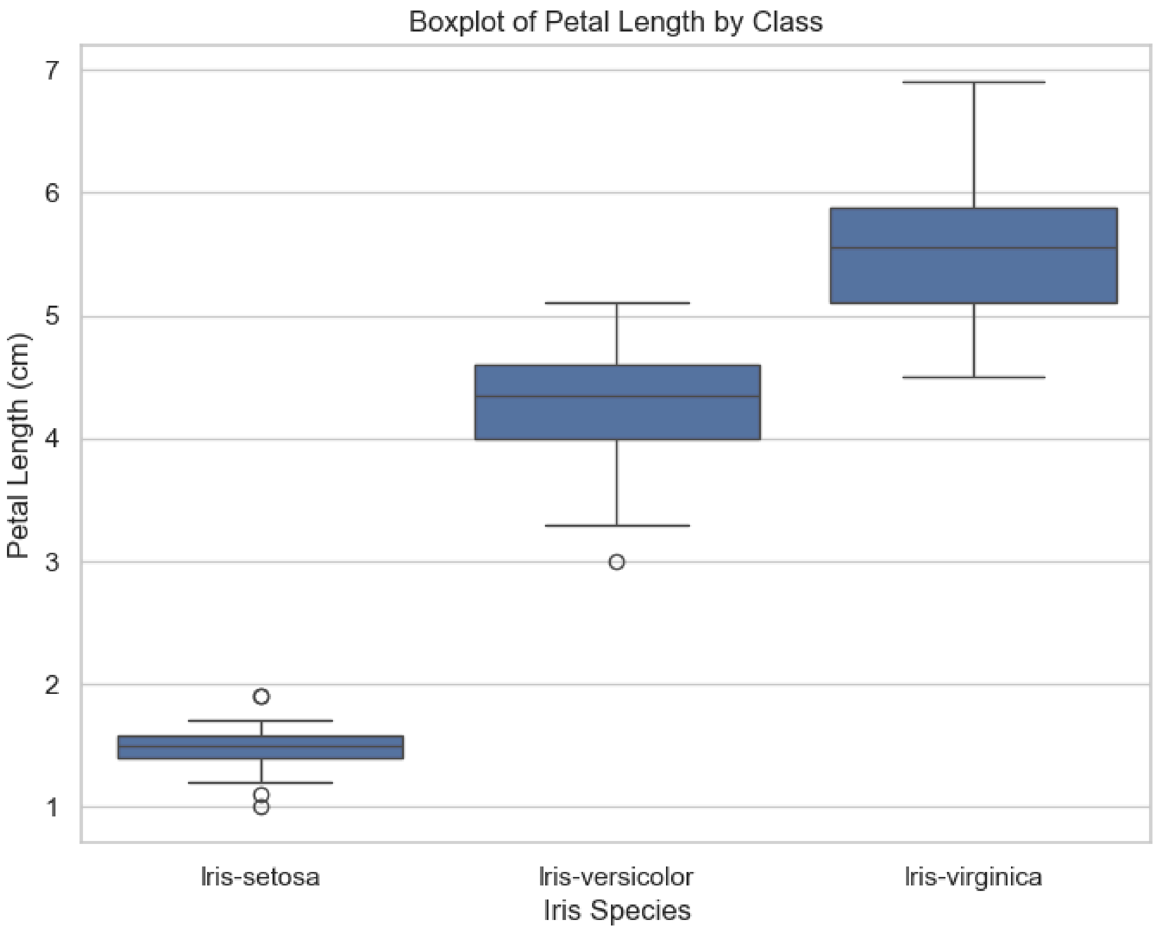


```
In [11]: plt.figure(figsize=(8, 6))
         sns.boxplot(data=iris, x='class', y='sepal_width')
         plt.title('Boxplot of Sepal Width by Class')
         plt.xlabel('Iris Species')
         plt.ylabel('Sepal Width (cm)')
         plt.show()
```

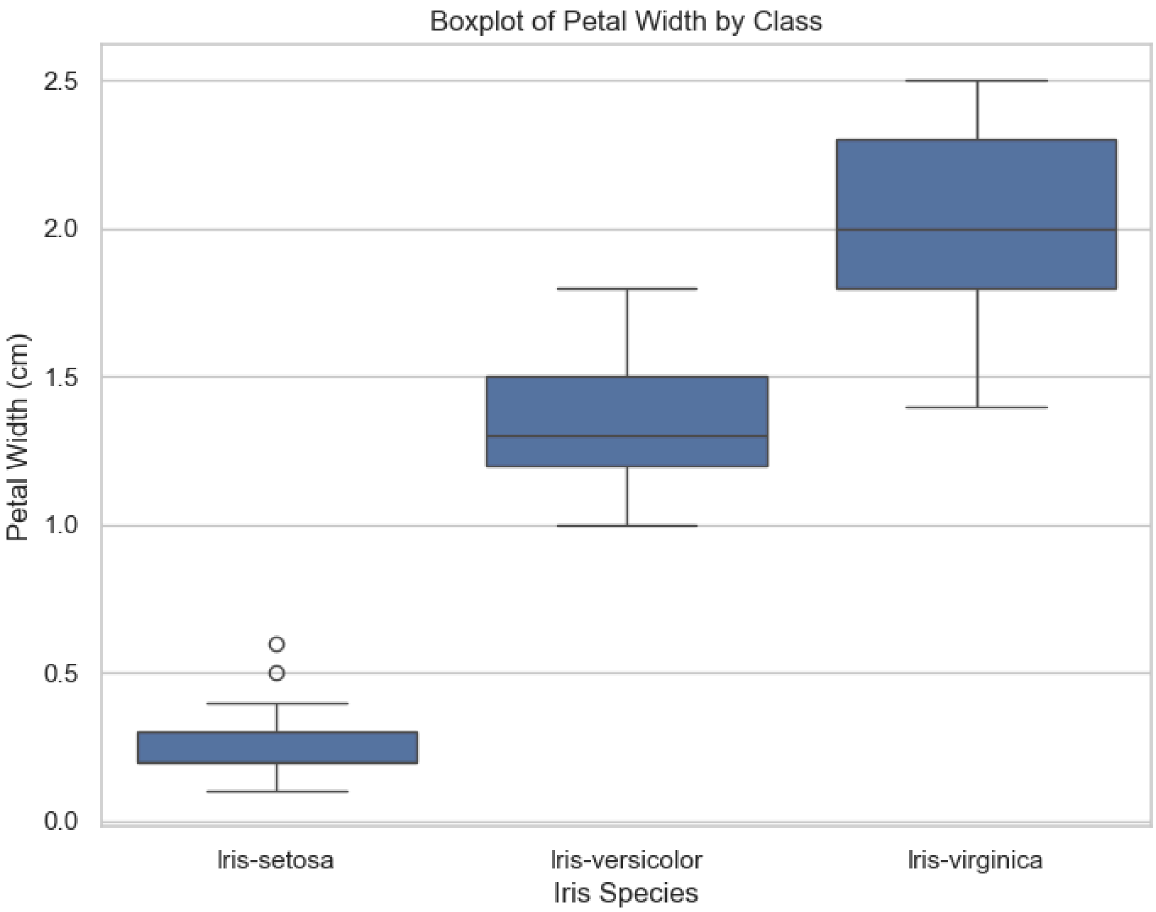


In [12]:

```
plt.figure(figsize=(8, 6))
sns.boxplot(data=iris, x='class', y='petal_length')
plt.title('Boxplot of Petal Length by Class')
plt.xlabel('Iris Species')
plt.ylabel('Petal Length (cm)')
plt.show()
```

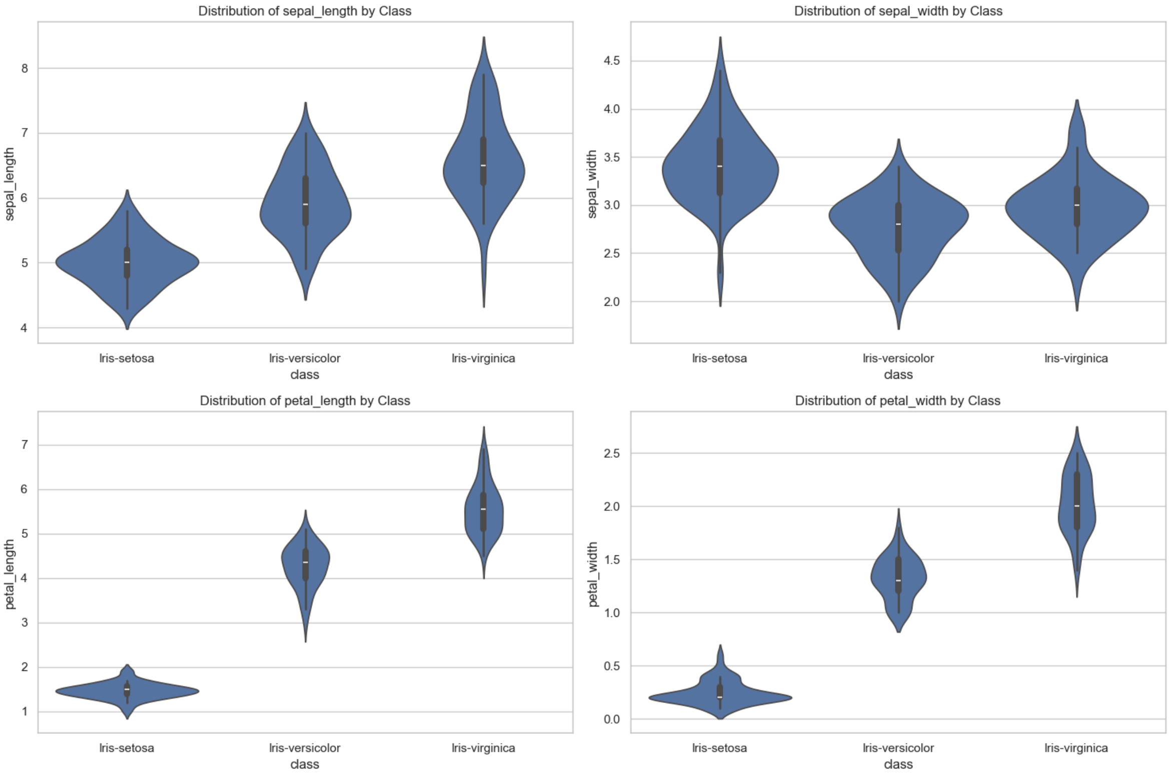


```
In [13]: plt.figure(figsize=(8, 6))
         sns.boxplot(data=iris, x='class', y='petal_width')
         plt.title('Boxplot of Petal Width by Class')
         plt.xlabel('Iris Species')
         plt.ylabel('Petal Width (cm)')
         plt.show()
```



In [14]:

```
plt.figure(figsize=(15, 10))
for i, feature in enumerate(column_names[:-1]):
    plt.subplot(2, 2, i+1)
    sns.violinplot(data=iris, x='class', y=feature)
    plt.title(f'Distribution of {feature} by Class')
plt.tight_layout()
plt.show()
```



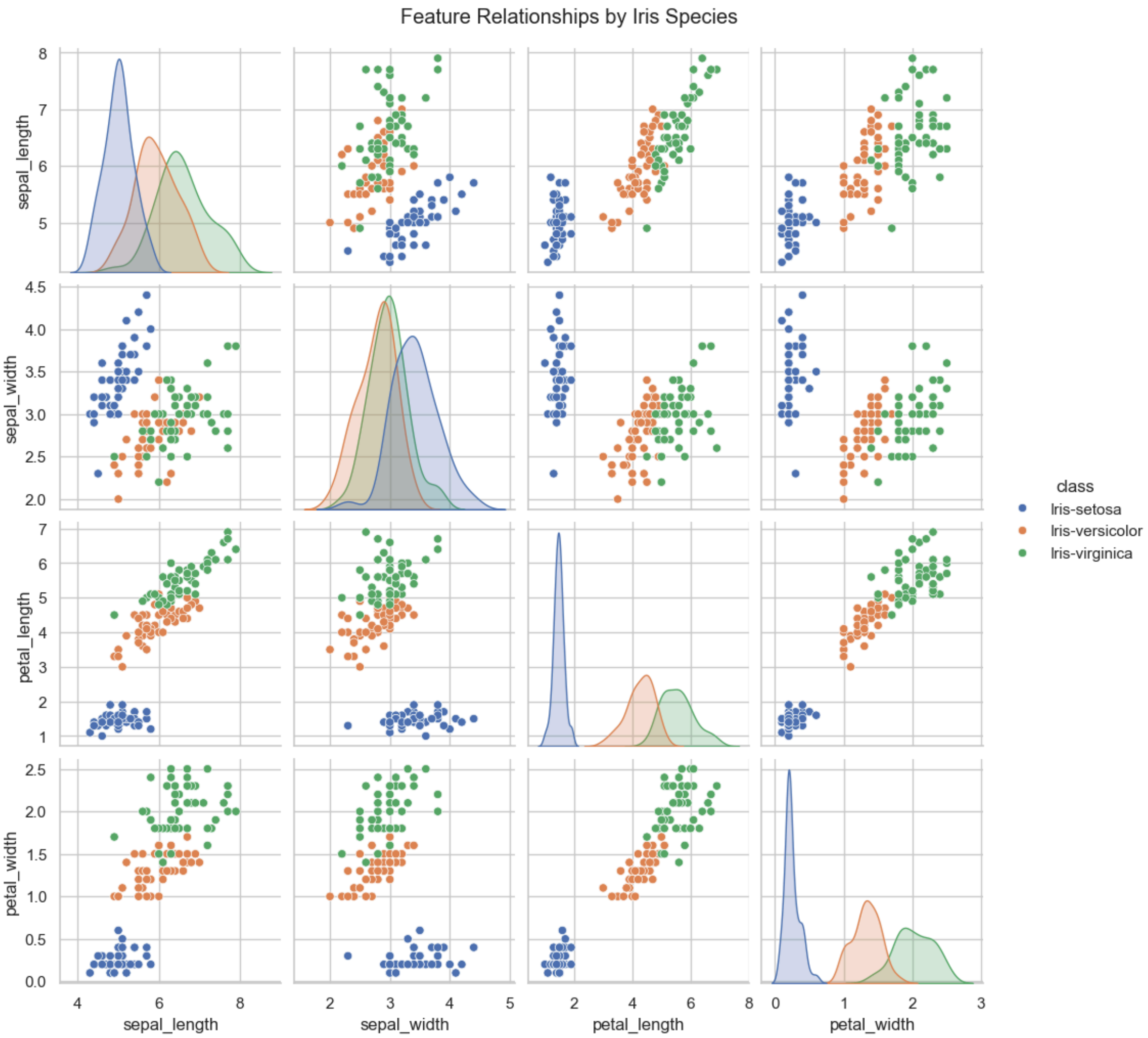
In [15]:

```
# Identify outliers using IQR method
for feature in column_names[:-1]:
    q1 = iris[feature].quantile(0.25)
    q3 = iris[feature].quantile(0.75)
    iqr = q3 - q1
    lower_bound = q1 - (1.5 * iqr)
    upper_bound = q3 + (1.5 * iqr)

    outliers = iris[(iris[feature] < lower_bound) | (iris[feature] > upper_bound
    print(f"\nOutliers in {feature}:")
    print(outliers[['class', feature]])
```

```
Outliers in sepal_length:
Empty DataFrame
Columns: [class, sepal_length]
Index: []

Outliers in sepal_width:
              class  sepal_width
15      Iris-setosa          4.4
32      Iris-setosa          4.1
33      Iris-setosa          4.2
60  Iris-versicolor          2.0

Outliers in petal_length:
Empty DataFrame
Columns: [class, petal_length]
Index: []

Outliers in petal_width:
Empty DataFrame
Columns: [class, petal_width]
Index: []
```

In [16]:

```
sns.pairplot(iris, hue='class', diag_kind='kde')
plt.suptitle('Feature Relationships by Iris Species', y=1.02)
plt.show()
```


Feature Relationships by Iris Species

In [ ]: