In [1]:
```
!pip install seaborn pandas matplotlib
```

```
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: pandas in /home/sargam/.conda/envs/myenv/li
b/python3.11/site-packages (2.2.3)
Collecting matplotlib
  Downloading matplotlib-3.10.1-cp311-cp311-manylinux_2_17_x86_64.manylinu
x2014_x86_64.whl.metadata (11 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /home/sargam/.cond
a/envs/myenv/lib/python3.11/site-packages (from seaborn) (2.0.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/sargam/.con
da/envs/myenv/lib/python3.11/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /home/sargam/.conda/envs/my
env/lib/python3.11/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /home/sargam/.conda/envs/
myenv/lib/python3.11/site-packages (from pandas) (2025.2)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2
014_x86_64.whl.metadata (5.5 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.57.0-cp311-cp311-manylinux_2_17_x86_64.manylinux
2014_x86_64.whl.metadata (102 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Downloading kiwisolver-1.4.8-cp311-cp311-manylinux_2_17_x86_64.manylinux
2014_x86_64.whl.metadata (6.2 kB)
Requirement already satisfied: packaging>=20.0 in /home/sargam/.conda/env
s/myenv/lib/python3.11/site-packages (from matplotlib) (24.2)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-11.2.1-cp311-cp311-manylinux_2_28_x86_64.whl.metadata
(8.9 kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.2.3-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: six>=1.5 in /home/sargam/.conda/envs/myenv/
lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas) (1.17.
0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
Downloading matplotlib-3.10.1-cp311-cp311-manylinux_2_17_x86_64.manylinux2
014_x86_64.whl (8.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.6/8.6 MB 7.1 MB/s eta 0:00:0
0[31m6.7 MB/s eta 0:00:01m
Downloading contourpy-1.3.2-cp311-cp311-manylinux_2_17_x86_64.manylinux201
4_x86_64.whl (326 kB)
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.57.0-cp311-cp311-manylinux_2_17_x86_64.manylinux20
14_x86_64.whl (4.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.9/4.9 MB 5.4 MB/s eta 0:00:0
0 MB/s eta 0:00:01m
Downloading kiwisolver-1.4.8-cp311-cp311-manylinux_2_17_x86_64.manylinux20
14_x86_64.whl (1.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.4/1.4 MB 10.7 MB/s eta 0:00:
00
Downloading pillow-11.2.1-cp311-cp311-manylinux_2_28_x86_64.whl (4.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.6/4.6 MB 10.8 MB/s eta 0:00:
0031m11.4 MB/s eta 0:00:01
Downloading pyparsing-3.2.3-py3-none-any.whl (111 kB)
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, c
ycler, contourpy, matplotlib, seaborn
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8/8 [seaborn]━━ 7/8 [seaborn]
ib]
```

```
Successfully installed contourpy-1.3.2 cycler-0.12.1 fonttools-4.57.0 kiwi
solver-1.4.8 matplotlib-3.10.1 pillow-11.2.1 pyparsing-3.2.3 seaborn-0.13.
2
```

In [3]:
```python
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:
```python
iris = sns.load_dataset('iris')
```

In [5]:
```python
# Statistical summary of numeric features
print(iris.describe())

# Information about the dataset: column types and null values
print(iris.info())
```

```
       sepal_length  sepal_width  petal_length  petal_width
count    150.000000   150.000000    150.000000   150.000000
mean       5.843333     3.057333      3.758000     1.199333
std        0.828066     0.435866      1.765298     0.762238
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 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
None
```

In [7]:
```python
iris.shape
```

Out[7]:
```
(150, 5)
```

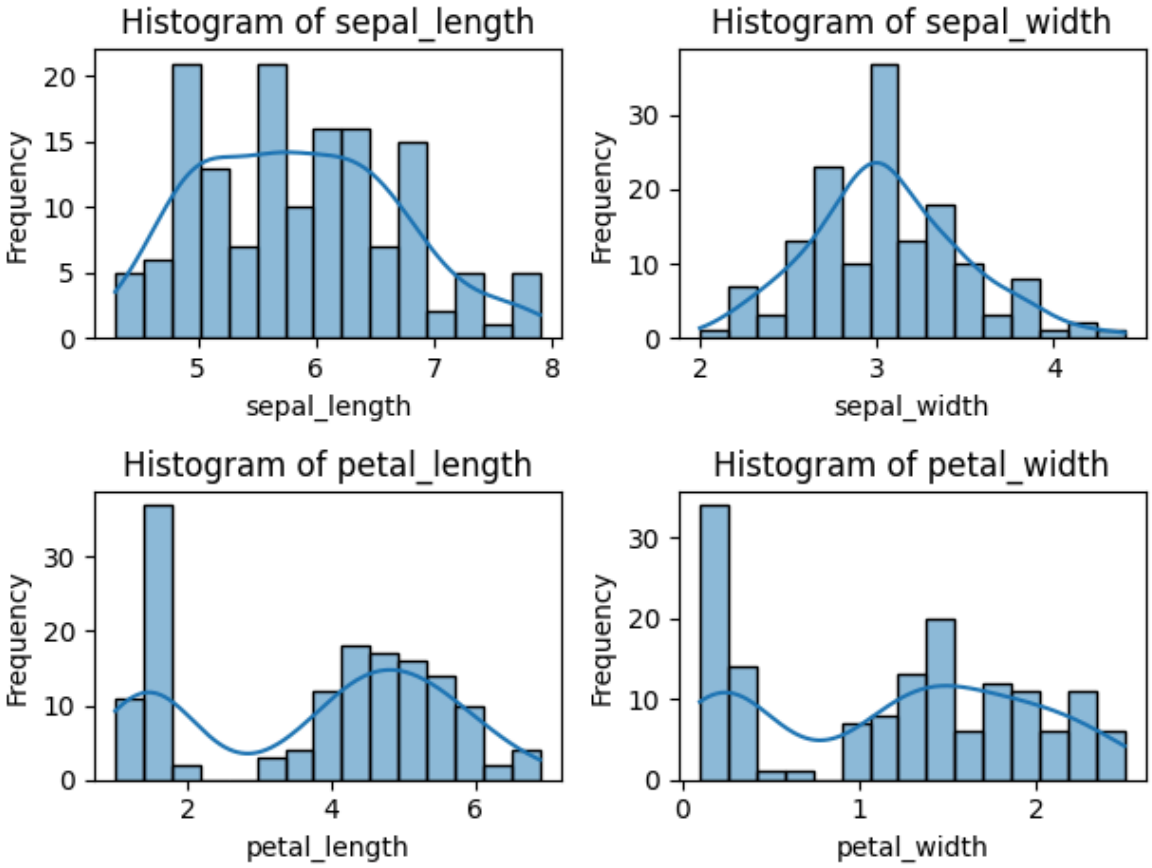
In [10]:
```python
# Automatically Detecting Feature Types
for col in iris.columns:
    if iris[col].dtype == 'object':
        print(f"{col}: nominal, categorical")
    else:
        print(f"{col}: numeric, continuous")
```

```
sepal_length: numeric, continuous
sepal_width: numeric, continuous
petal_length: numeric, continuous
petal_width: numeric, continuous
species: nominal, categorical
```

In [11]:
```python
# Create histograms with KDE for each feature
for i, feature in enumerate(iris.columns[:-1], 1):  # Exclude the last co
    plt.subplot(2, 2, i)
    sns.histplot(iris[feature], kde=True, bins=15)
```

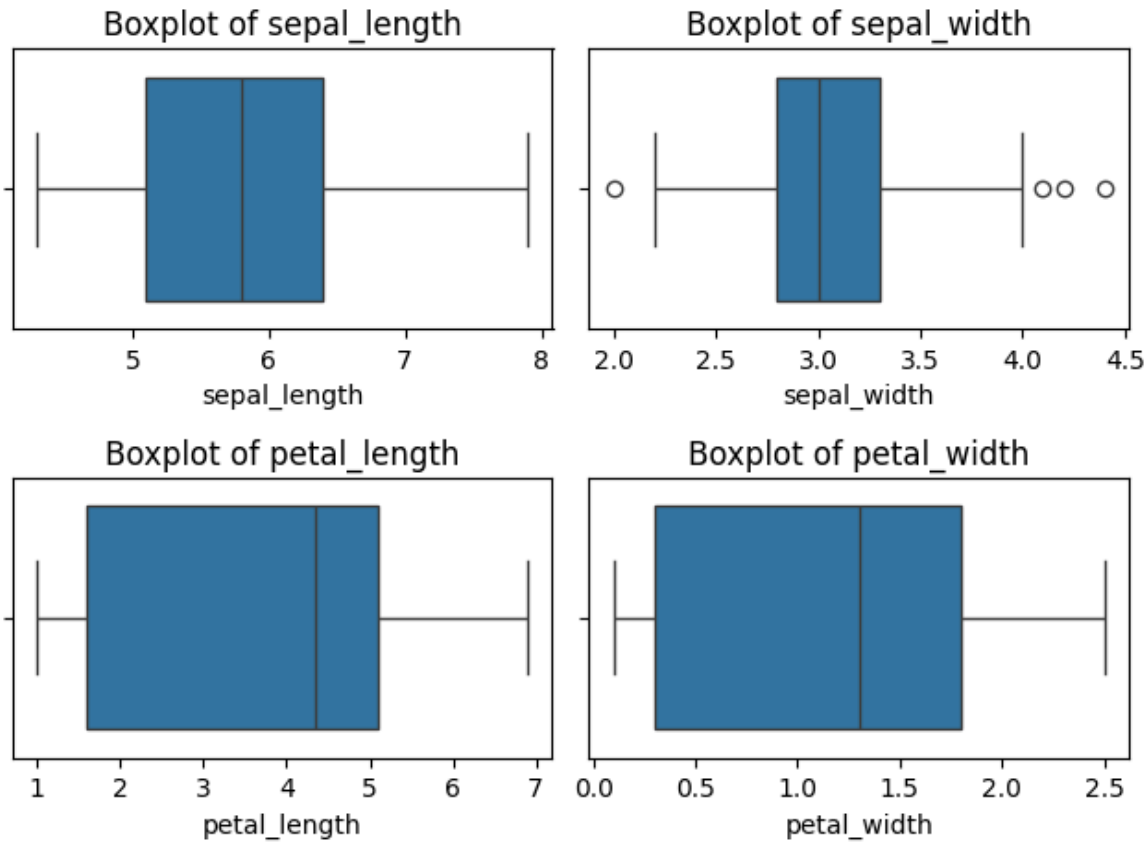
```
    plt.title(f'Histogram of {feature}')
    plt.xlabel(feature)
    plt.ylabel('Frequency')
    plt.tight_layout()

plt.show()
```



In [12]:
```
# Create boxplots for each feature (excluding 'species' column)
for i, feature in enumerate(iris.columns[:-1], 1):  # Exclude the 'specie
    plt.subplot(2, 2, i)  # 2x2 grid of subplots
    sns.boxplot(x=iris[feature])
    plt.title(f'Boxplot of {feature}')
    plt.xlabel(feature)

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



In [17]:
```
print("\nOutliers in the Iris dataset (if any can be identified fromthe p
print("1. Outliers in Sepal Length: Observed as points outside the whiske
print("2. Outliers in Petal Length: Observed as points outside the whiske
```

```
Outliers in the Iris dataset (if any can be identified fromthe plots):
1. Outliers in Sepal Length: Observed as points outside the whiskers in th
e boxplot.
2. Outliers in Petal Length: Observed as points outside the whiskers in th
e boxplot.
```

In [23]:
```
from scipy.stats import zscore

# Step 2: Identify outliers using Z-scores
numeric_features = iris.select_dtypes(include=['float64'])

# Calculate Z-scores
z_scores = numeric_features.apply(zscore)

# Identify outliers (Z-score > 3 or Z-score < -3)
outliers = (z_scores.abs() > 3).sum()

# Print the outliers count for each feature
print("Number of outliers for each feature:")
print(outliers)
```

```
Number of outliers for each feature:
sepal_length    0
sepal_width     1
petal_length    0
petal_width     0
dtype: int64
```

In [ ]: