Harshada Mhaske (73)

Download the iris flower dataset or any other dataset into a dataframe. Scan the dataset and given the interference as

1. List down the feature and types
2. Create histogram for each feature in the dataset
3. Create a boxplot for each seature in the dataset
4. Compare distributions and identify

Import seaborn library

```
import seaborn as sns
```

In [ ]:

Load the Iris dataset using seaborn

```
df = sns.load_dataset('iris')
```

In [ ]:

Display the dataframe

```
df
```

In [ ]:

Out [3]:

	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
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

```
In [ ]:  df.columns
```

Out [4]: Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
                'species'],
               dtype='object')

Display information about the dataframe, including data types and missing values

```
In [ ]:  df.info()
```

```
<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
```

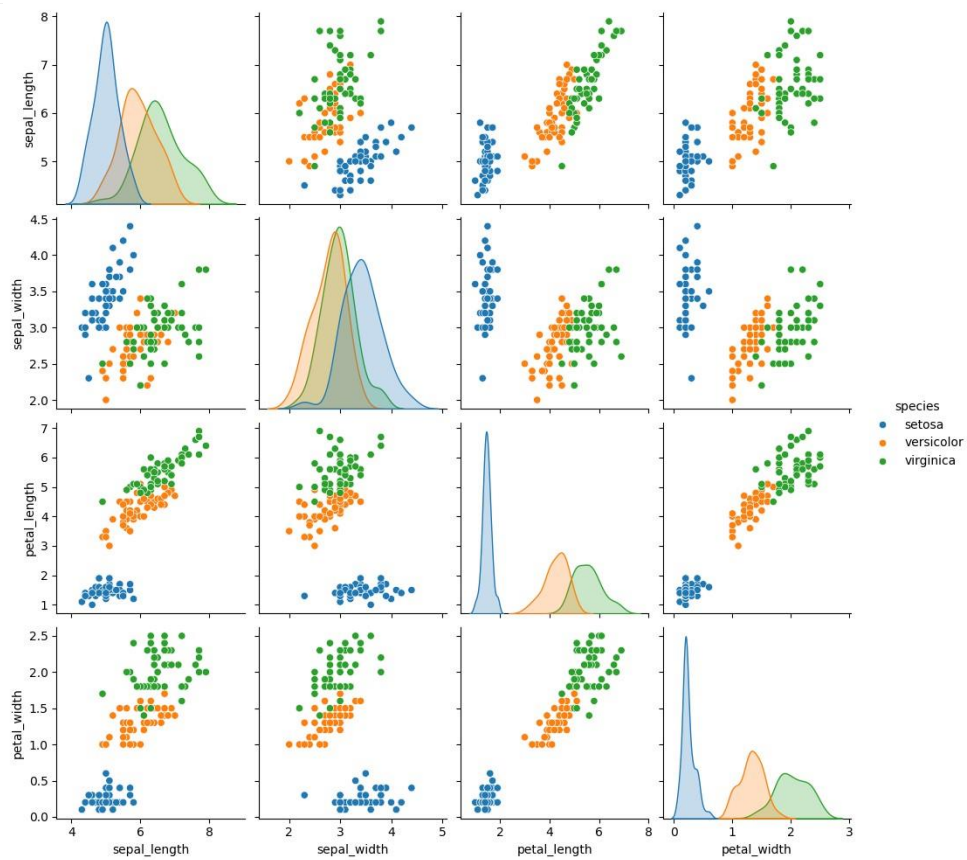Display the data types of columns in the dataframe

```
In [ ]:  df.dtypes
```

Out [6]: sepal_length   float64
         sepal_width   float64
         petal_length   float64
         petal_width   float64
         species   object
         dtype: object

Create a pair plot to visualize pairwise relationships between different variables with hue based on species
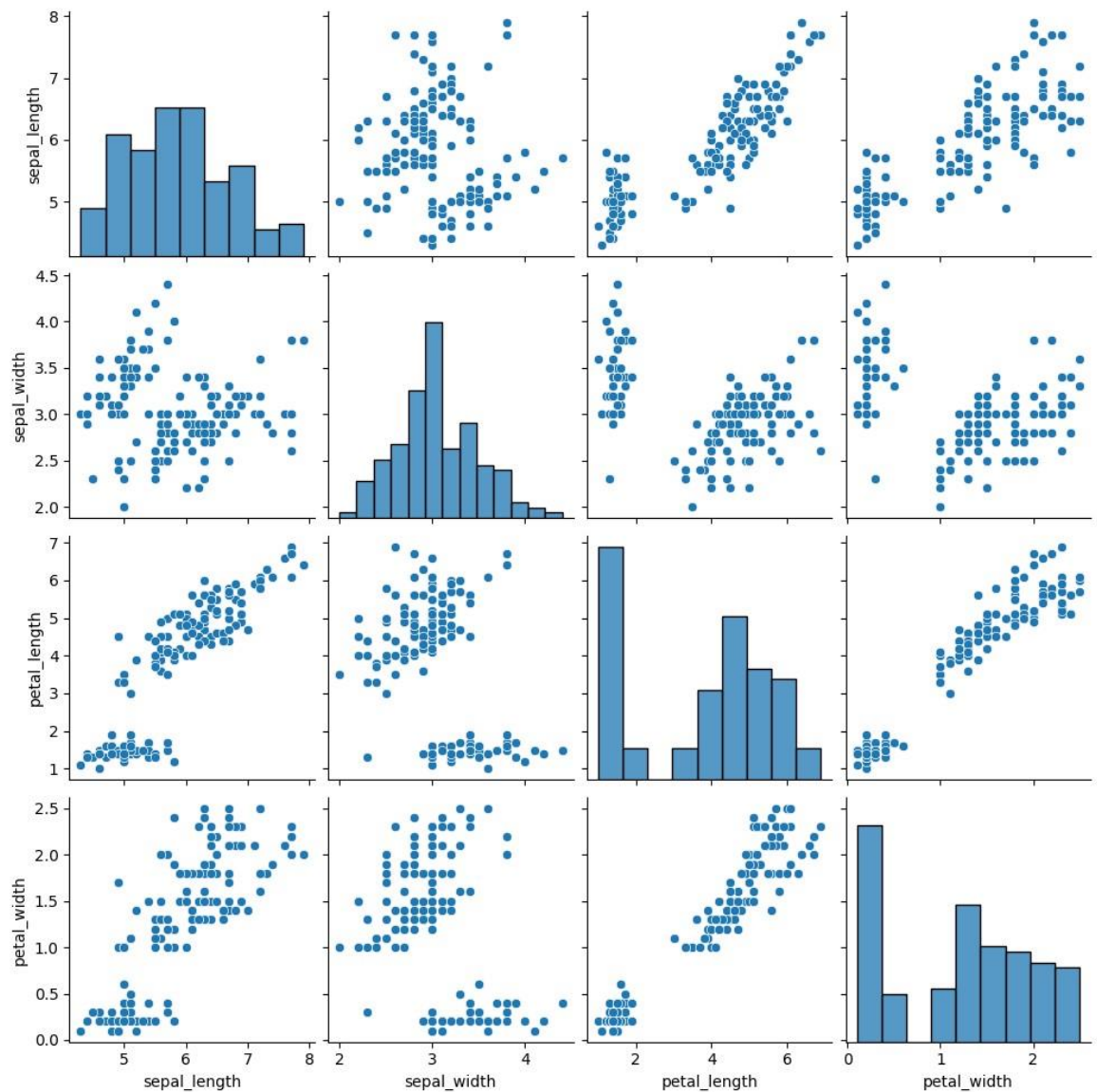
```
In [ ]:  sns.pairplot(df,hue = 'species')
```

Out [7]: <seaborn.axisgrid.PairGrid at 0x7a48a6ae370>

Create a pair plot to visualize pairwise relationships between different variables

```
In [ ]: sns.pairplot(df)
```
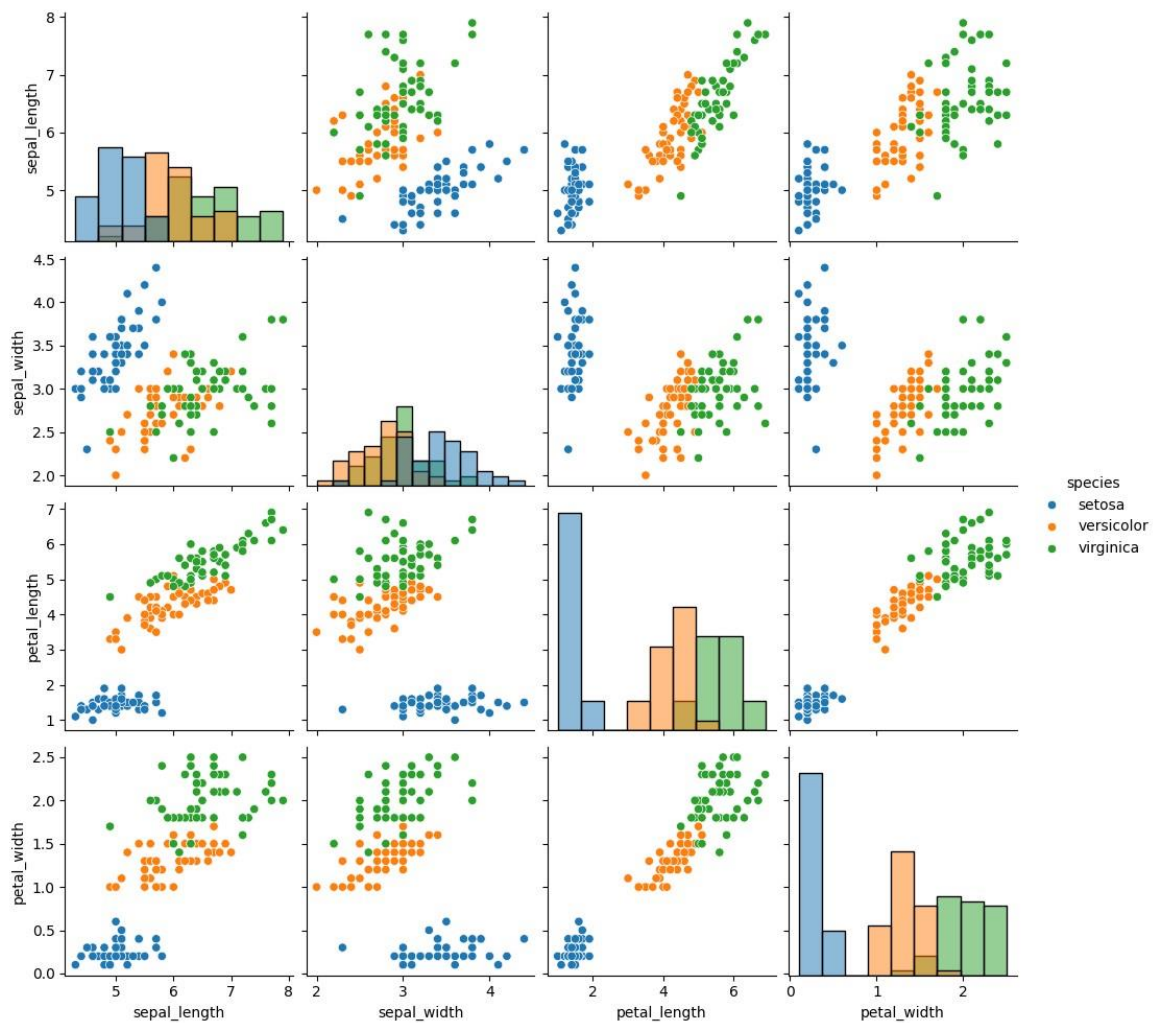
Out [8]: <seaborn.axisgrid.PairGrid at 0x106c700>

Create a pair plot to visualize pairwise relationships between different variables with hue based on species and histogram on the diagonal

```
In [ ]:  sns.pairplot(df,hue='species',diag_kind='hist')
```
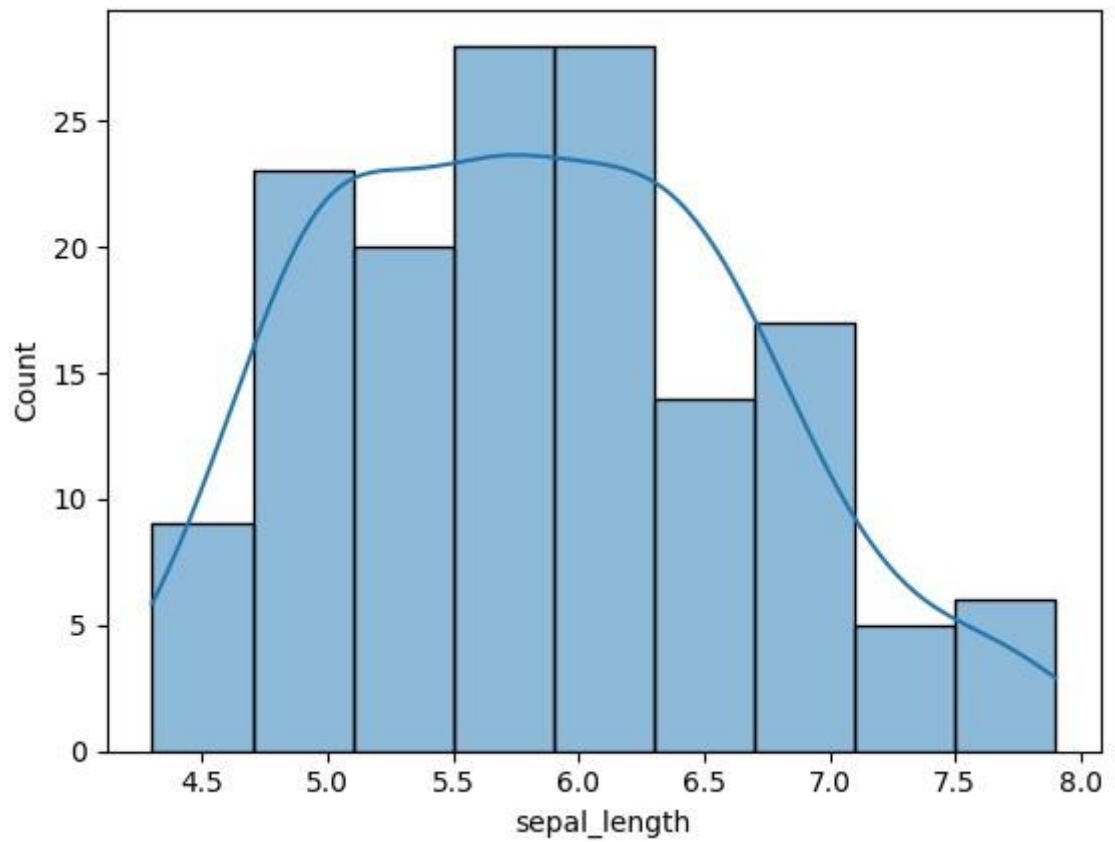
Out [9]: <seaborn.axisgrid.PairGrid at 0x7a48a1e3b400>

Create a histogram to visualize the distribution of sepal length with a kernel density estimate

```
In [ ]:  sns.histplot(df['sepal_length'],kde = True)
```
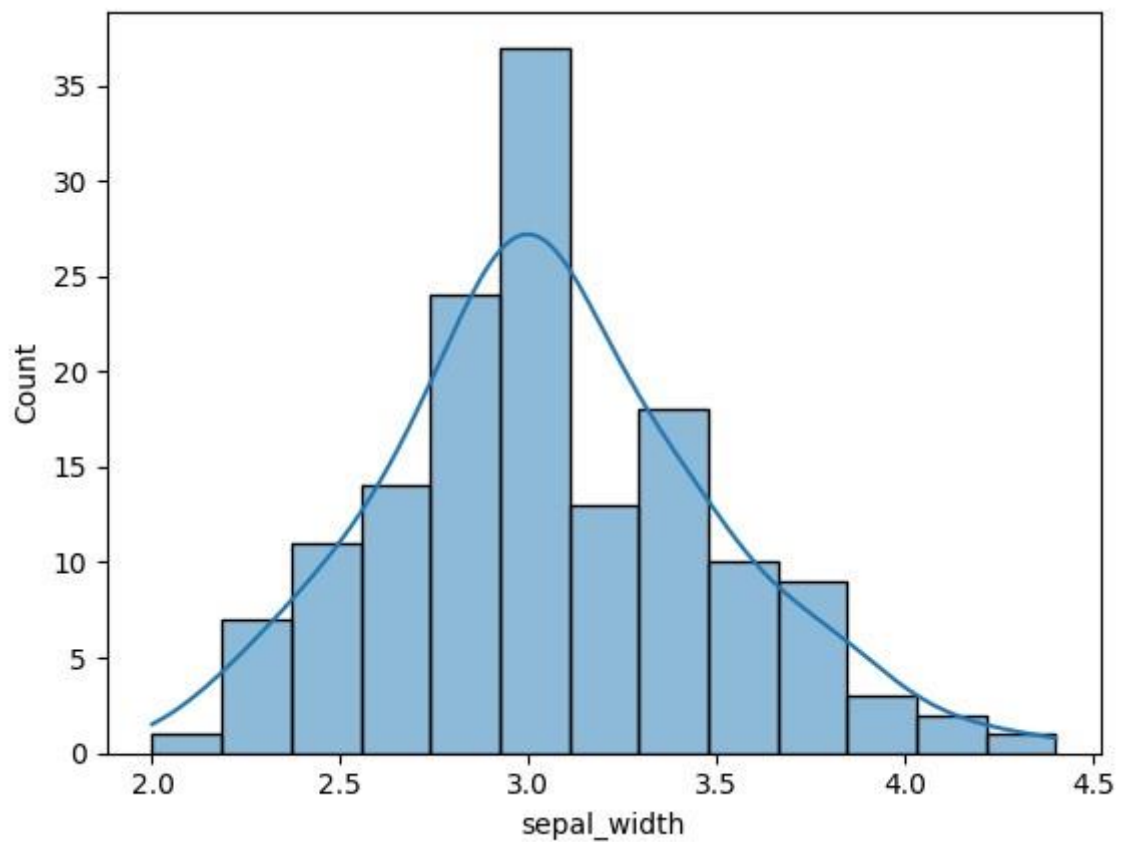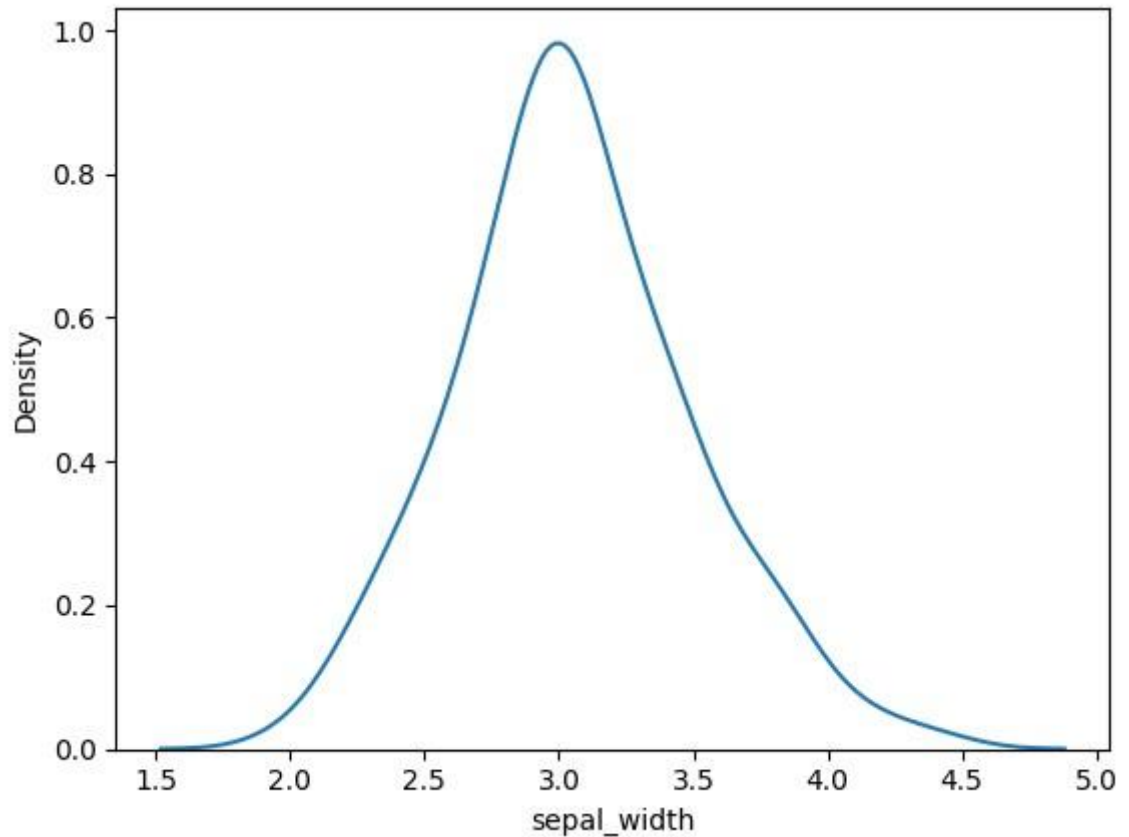
Out [10]: <Axes: xlabel='sepal_length', ylabel='Count'>

Create a histogram to visualize the distribution of sepal width with a kernel density estimate

In [ ]:
```
sns.histplot(df['sepal_width'],kde = True)
```

Out [11]: <Axes: xlabel='sepal width', ylabel='Count'>

Create a kernel density estimate plot to visualize the distribution of sepal width
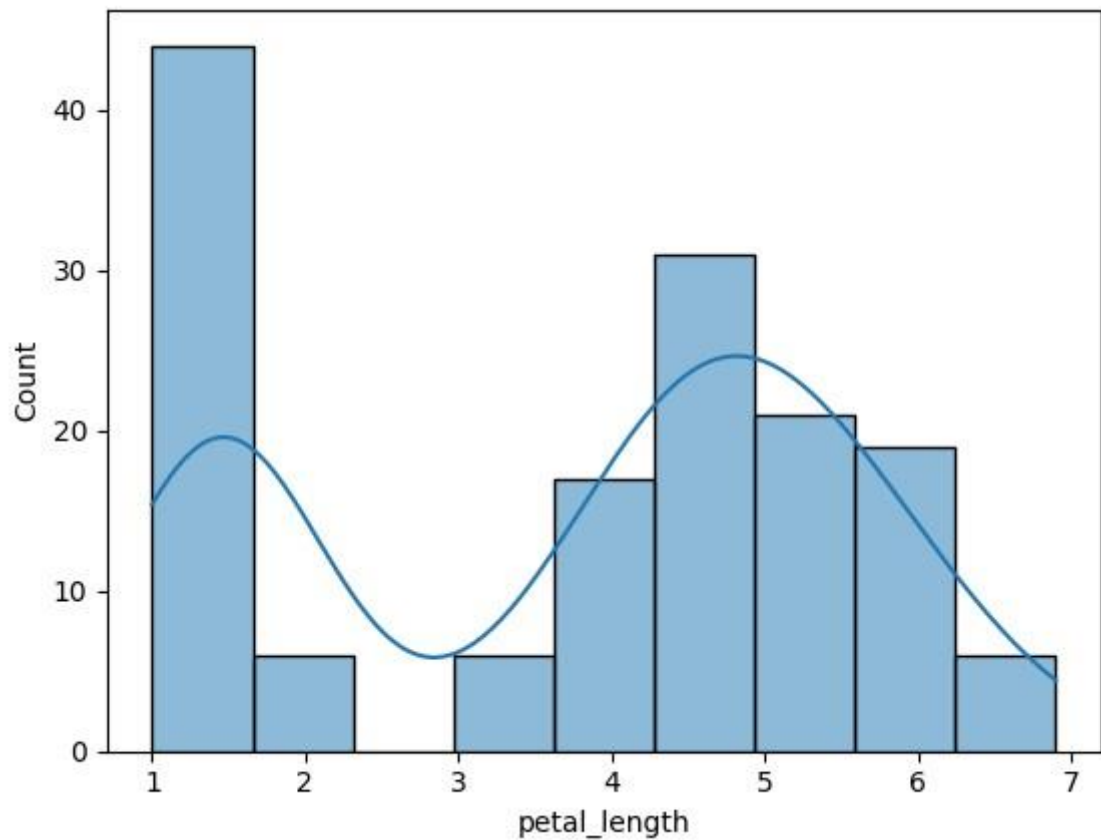
```
sns.kdeplot(df['sepal_width'])
```

Out [12]: <Axes: xlabel='sepal width', ylabel='Density'>

Create a histogram to visualize the distribution of petal length with a kernel density estimate

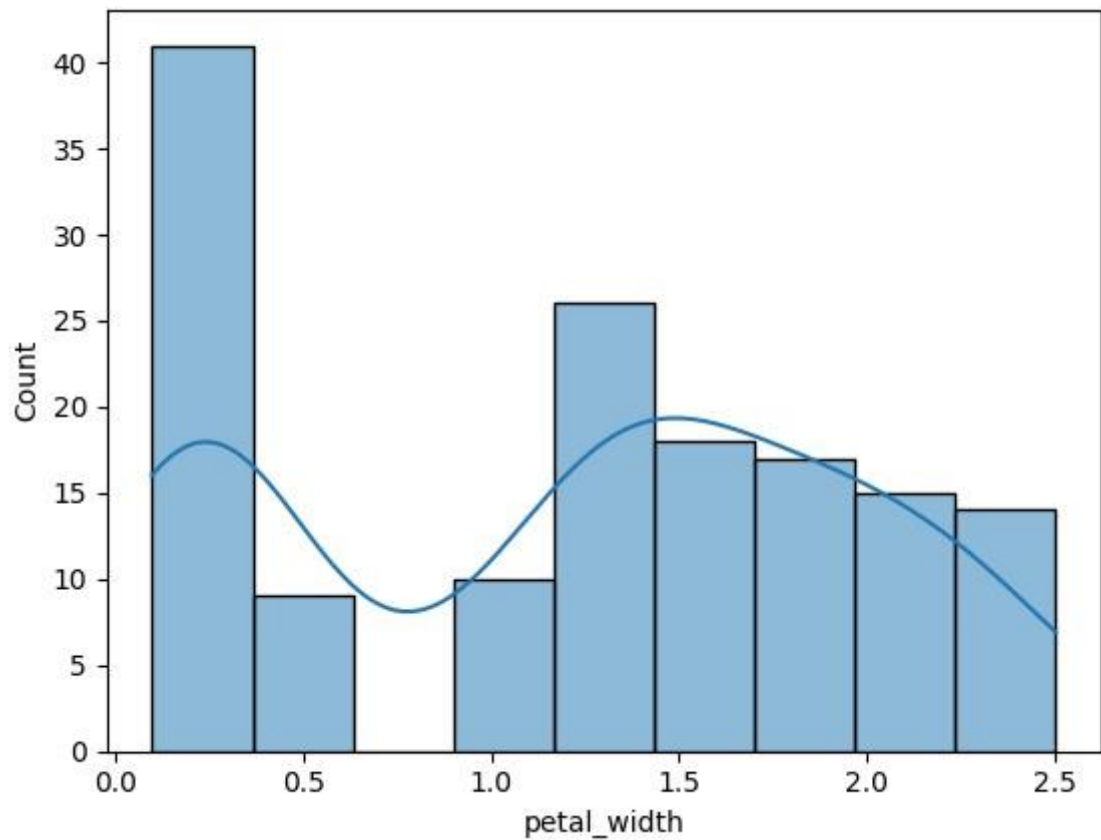In [ ]:
```
sns.histplot(df['petal_length'],kde = True)
```

Out [13]: <Axes: xlabel='petal_length', ylabel='Count'>

Create a histogram to visualize the distribution of petal width with a kernel density estimate

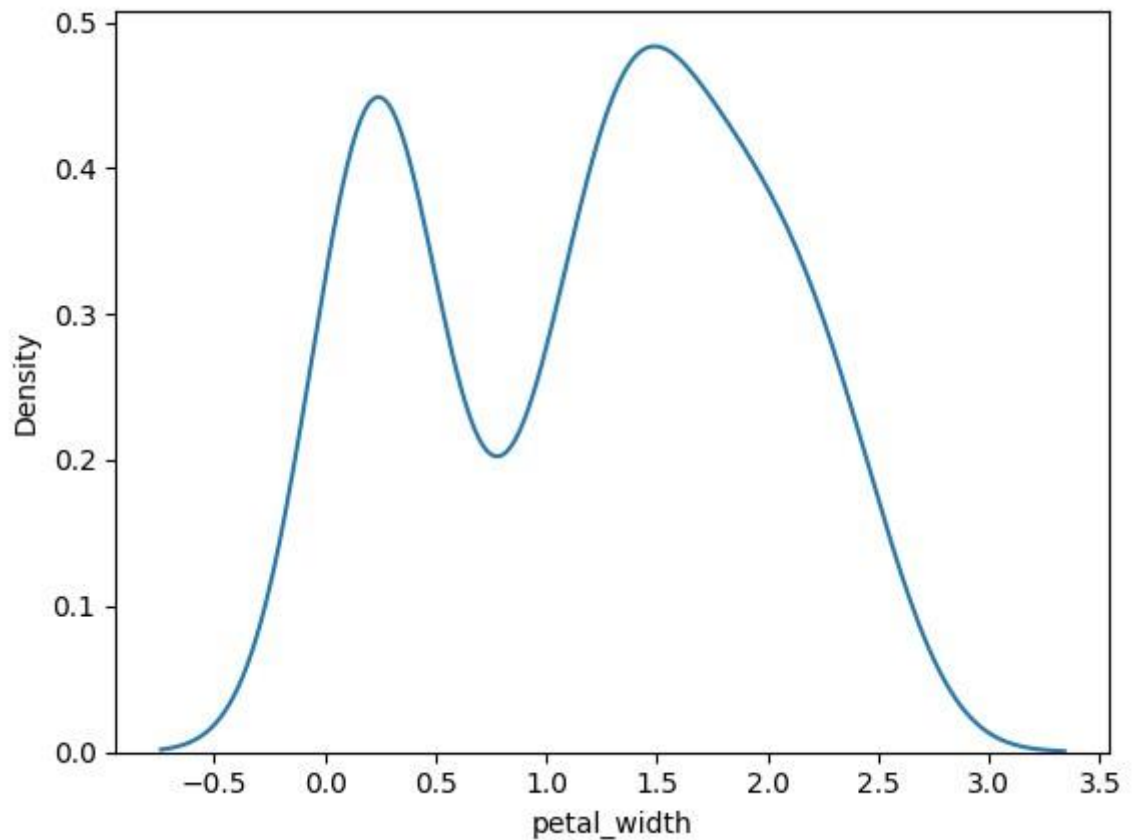```
In [ ]: sns.histplot(df['petal_width'],kde = True)
```

Out [14]: <Axes: xlabel='petal width', ylabel='Count'>

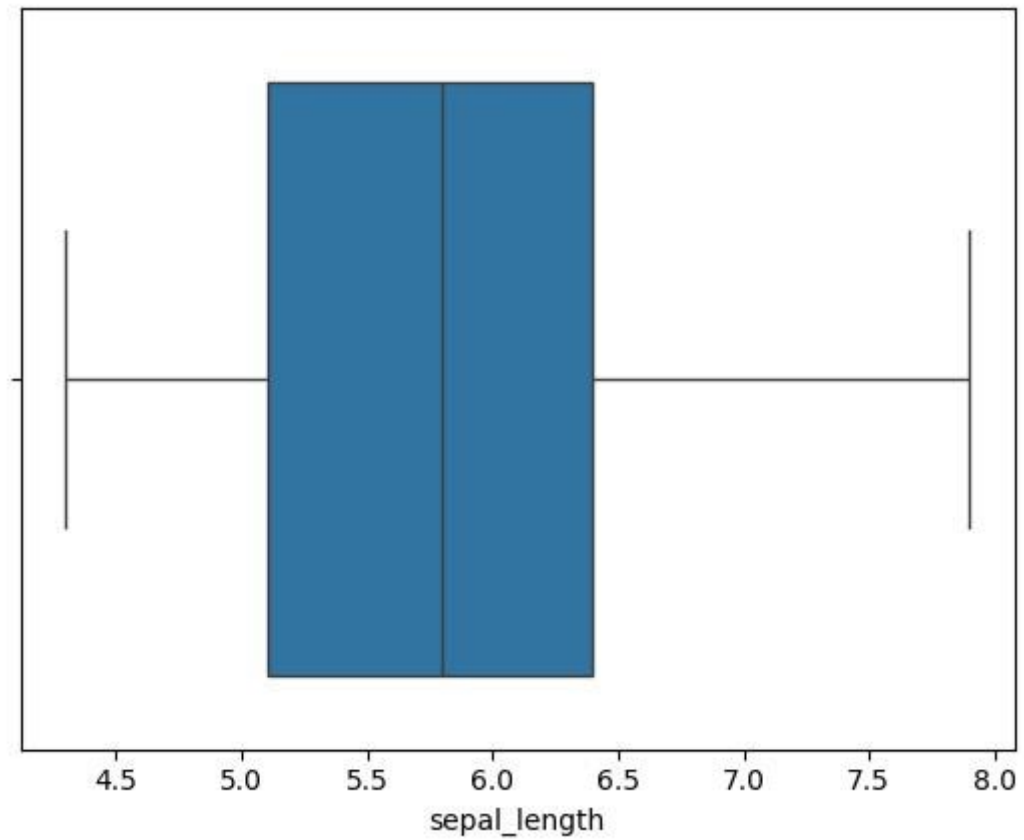Create a kernel density estimate plot to visualize the distribution of petal width

In [ ]:
```
sns.kdeplot(df['petal_width'])
```

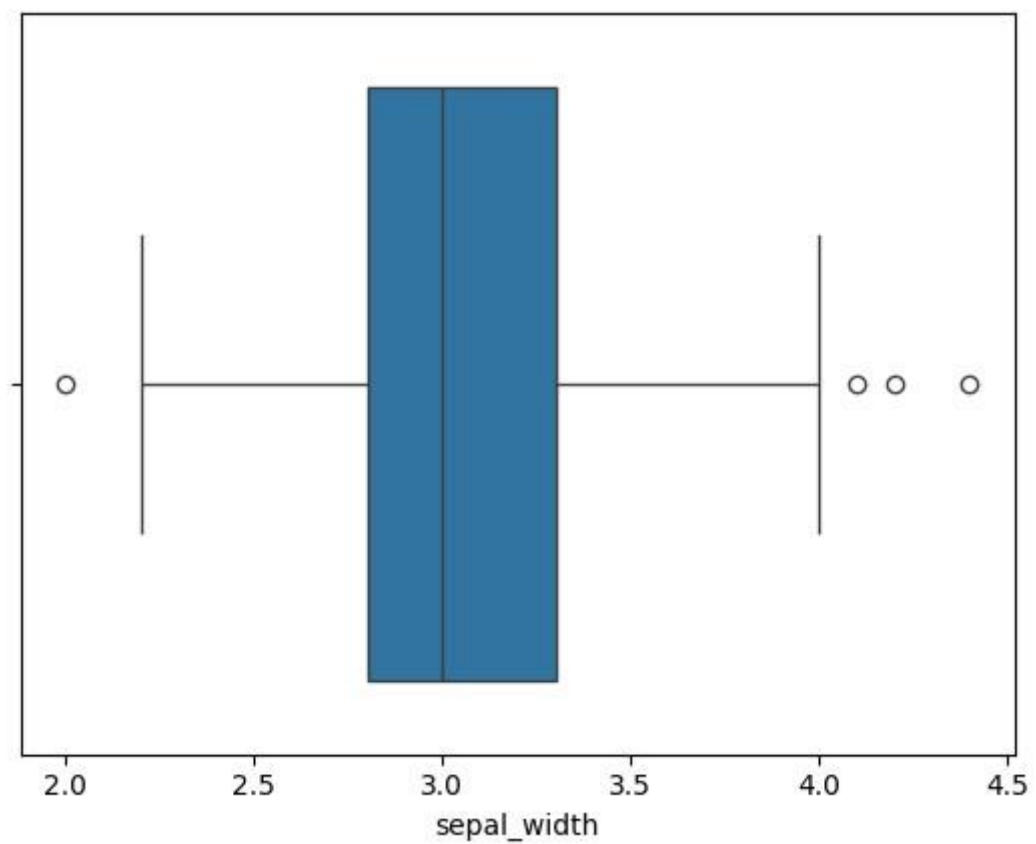Out [15]: <Axes: xlabel='petal width', ylabel='Density'>



Create a box plot to visualize the distribution of sepal length
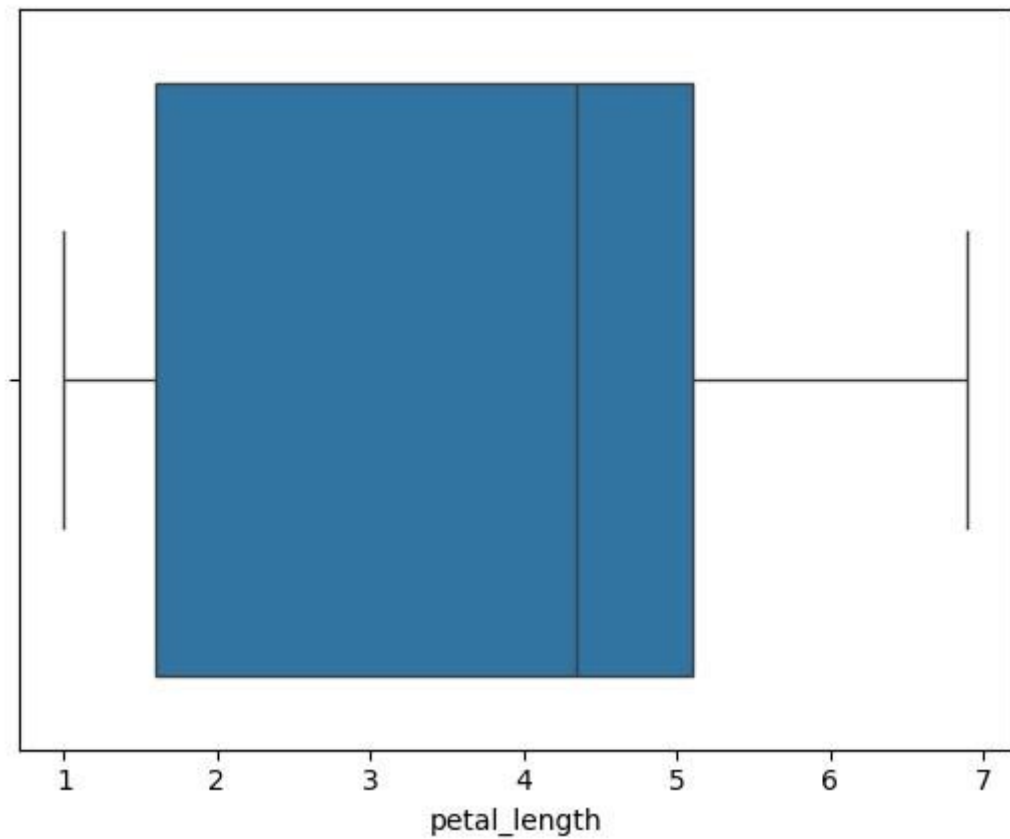
In [ ]:
```
sns.boxplot(x=df['sepal_length']);
```

Create a box plot to visualize the distribution of sepal width

In [ ]: 
```python
sns.boxplot(x=df['sepal_width']);
```
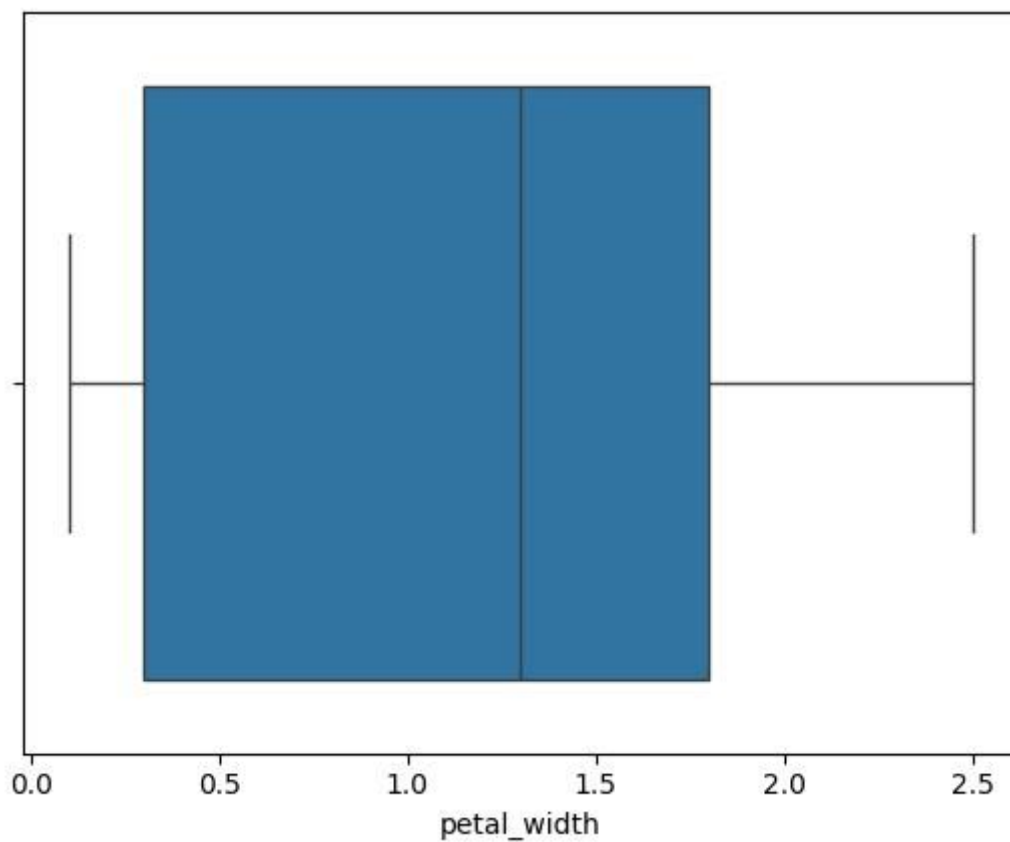
Create a box plot to visualize the distribution of petal length

In [ ]:
```python
sns.boxplot(x=df['petal_length']);
```


petal_length

In [ ]:
```python
sns.boxplot(x=df['petal_width']);
```


petal_width

In [ ]: