Data visualization of categorical data; categorial data can be: nominal, qualitative , ordinal

Libraries Used:

1. Seaborn: Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python.
2. Pandas: Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring and manipulating data.
3. Numpy: is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.
4. Sklearn: It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

```
import seaborn as sns
df = sns.load_dataset('iris')
df
```

	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

Next steps: View recommended plots

```
#list down there features and tere types available in dataset
df.columns
```

```
    Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
           'species'],
          dtype='object')
```

```
df.info()
```

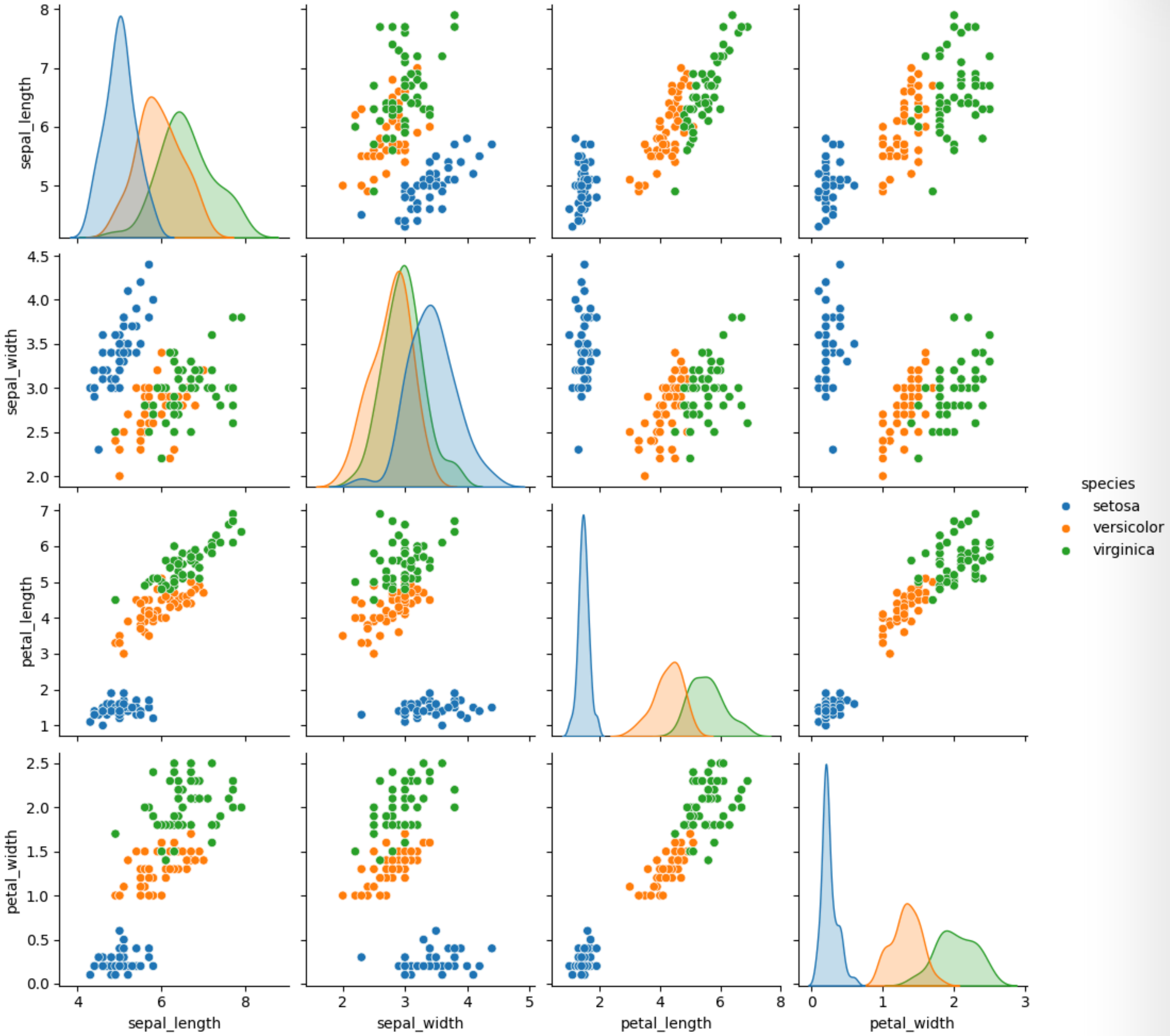
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 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
```

```
df.dtypes
```

```
sepal_length    float64
sepal_width     float64
petal_length    float64
petal_width     float64
species          object
dtype: object
```

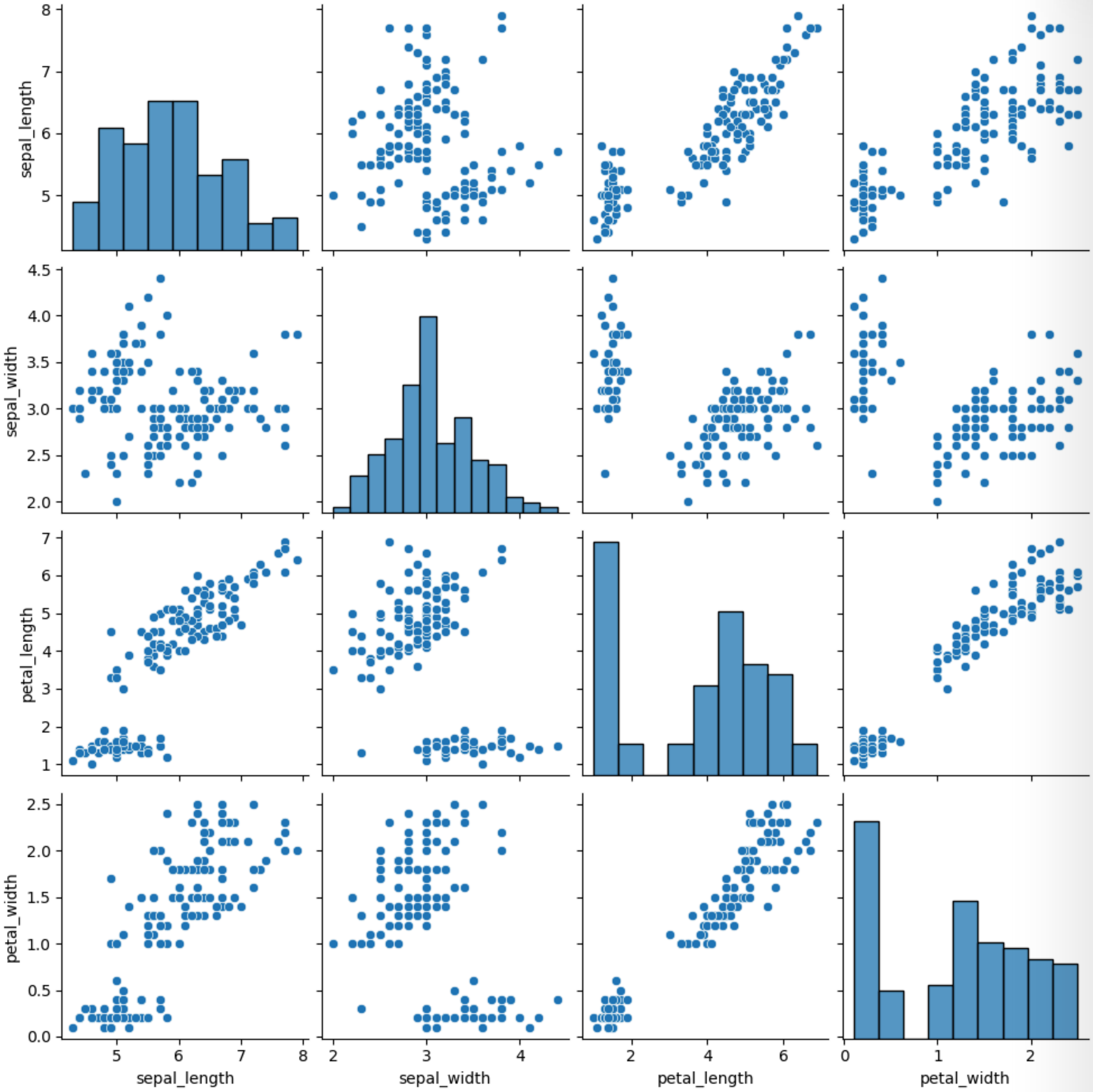
```
sns.pairplot(df,hue='species')
```

```
<seaborn.axisgrid.PairGrid at 0x781c30da7730>
```



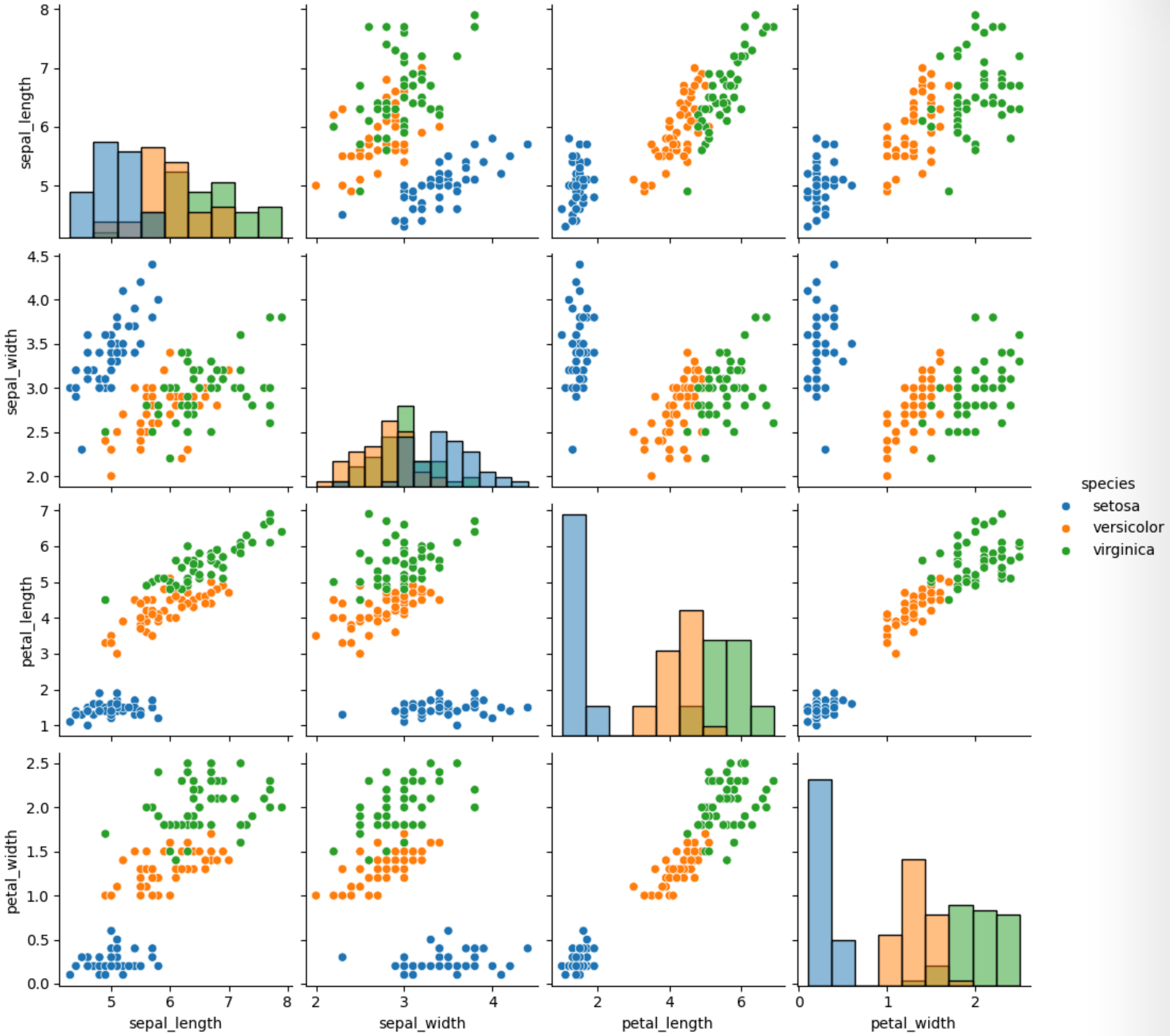
```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x781c2b6b6680>
```



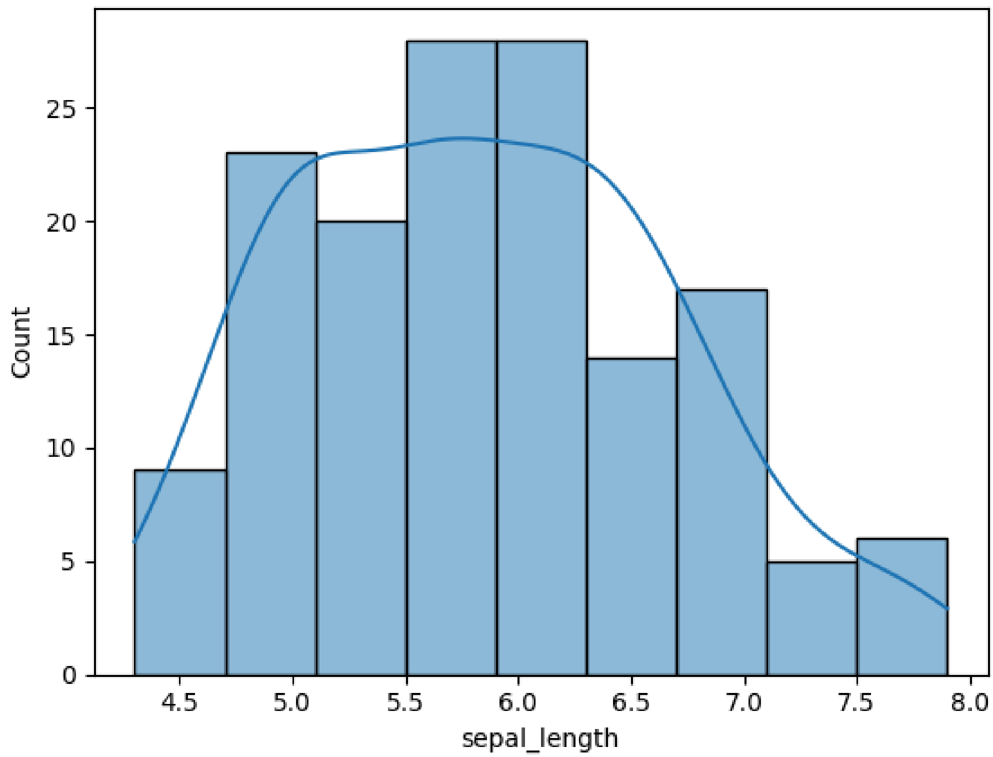
```
sns.pairplot(df,hue='species',diag_kind='hist')
```

```
<seaborn.axisgrid.PairGrid at 0x781c2aa9b040>
```



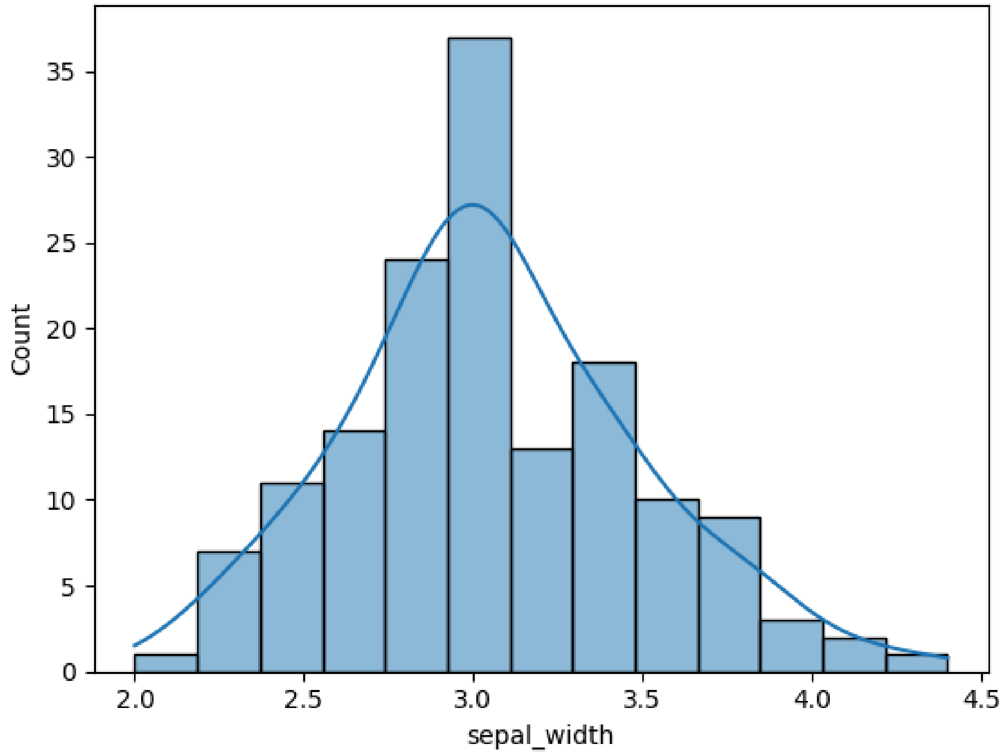
```
sns.histplot(df['sepal_length'],kde=True)
```

```
<Axes: xlabel='sepal_length', ylabel='Count'>
```



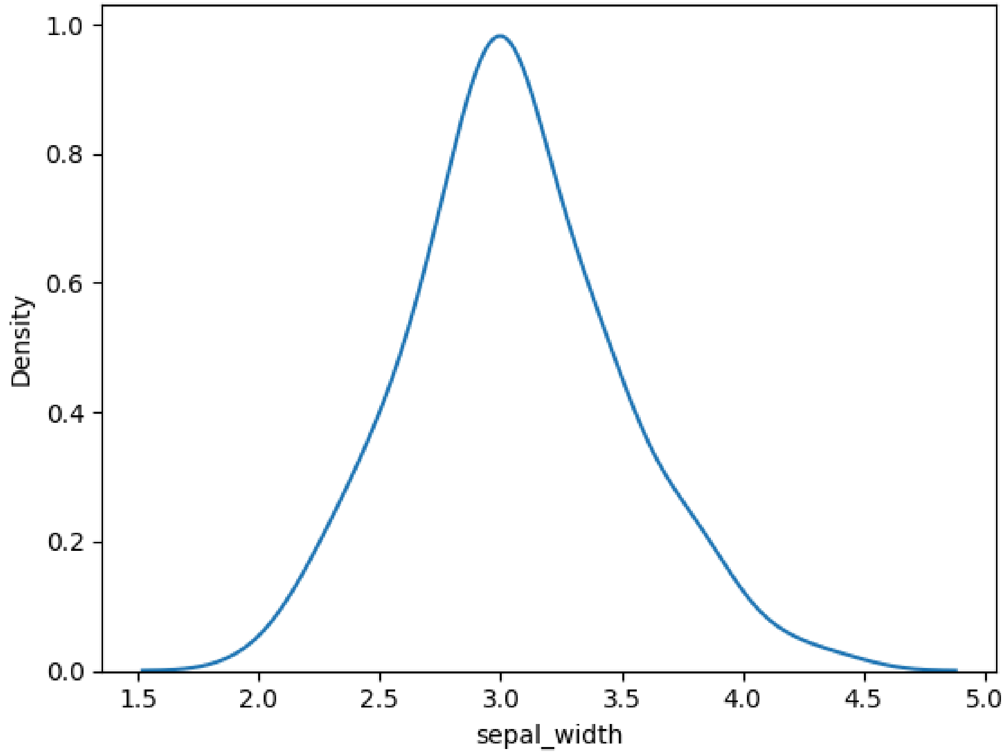
```
sns.histplot(df['sepal_width'],kde=True)
```

```
<Axes: xlabel='sepal_width', ylabel='Count'>
```



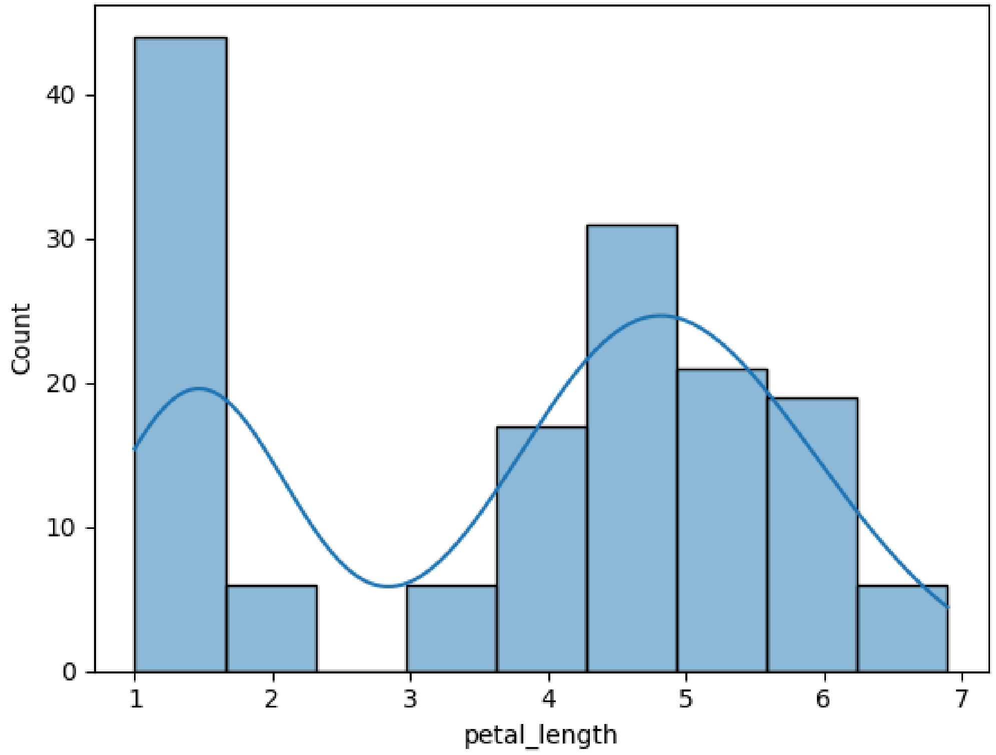
```
sns.kdeplot(df['sepal_width'])
```

```
<Axes: xlabel='sepal_width', ylabel='Density'>
```



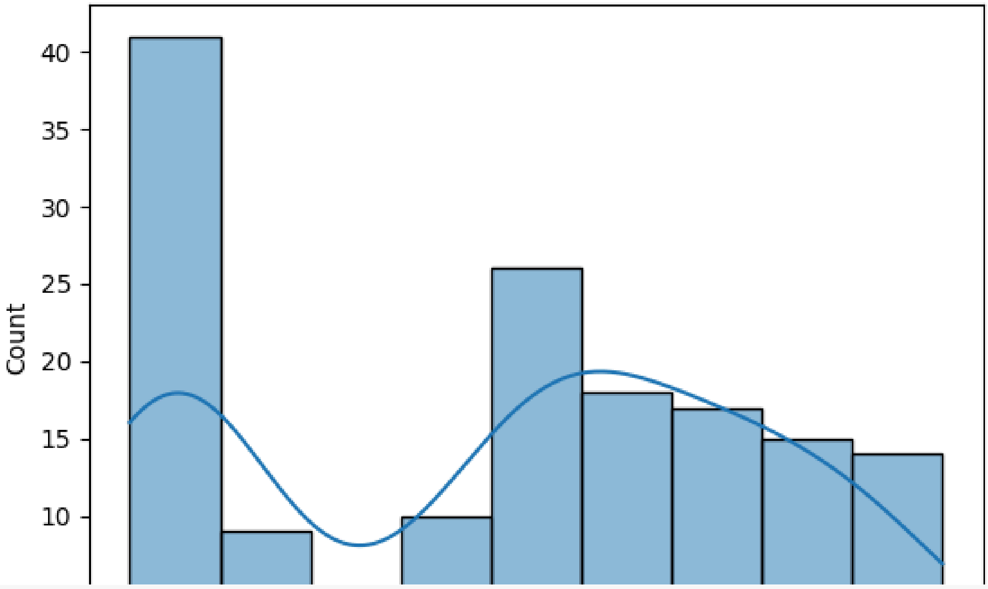
```
sns.histplot(df['petal_length'],kde=True)
```

```
<Axes: xlabel='petal_length', ylabel='Count'>
```



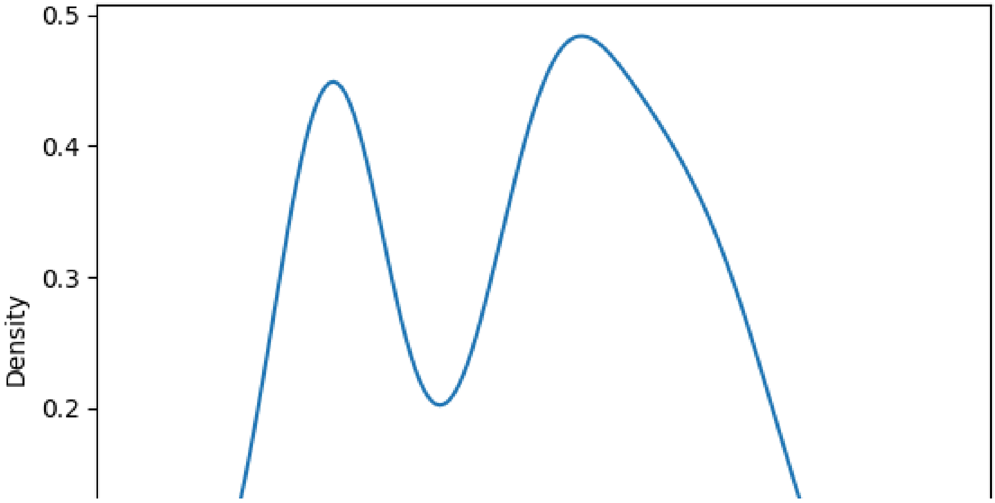
```
sns.histplot(df['petal_width'],kde=True)
```

```
<Axes: xlabel='petal_width', ylabel='Count'>
```



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

```
<Axes: xlabel='petal_width', ylabel='Density'>
```


Density