In [1]:
```
# importing packages
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
# Loading the data
df = pd.read_csv("IRIS.csv")
df.head()
```

Out[2]:

	sepal_length	sepal_width	petal_length	petal_width	species
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 [3]:
```
df.shape
```

Out[3]:
```
(150, 5)
```

In [4]:
```
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
```

Only one column has categorical data and all the other columns are of the numeric type with non-Null entries.

In [5]:
```
df.describe()
```

Out[5]:

	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

In [6]:
```
# We know there are no missing values in dataset so lets check for duplictae values
data = df.drop_duplicates(subset ="species",)
data
```

Out[6]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	Iris-setosa
50	7.0	3.2	4.7	1.4	Iris-versicolor
100	6.3	3.3	6.0	2.5	Iris-virginica

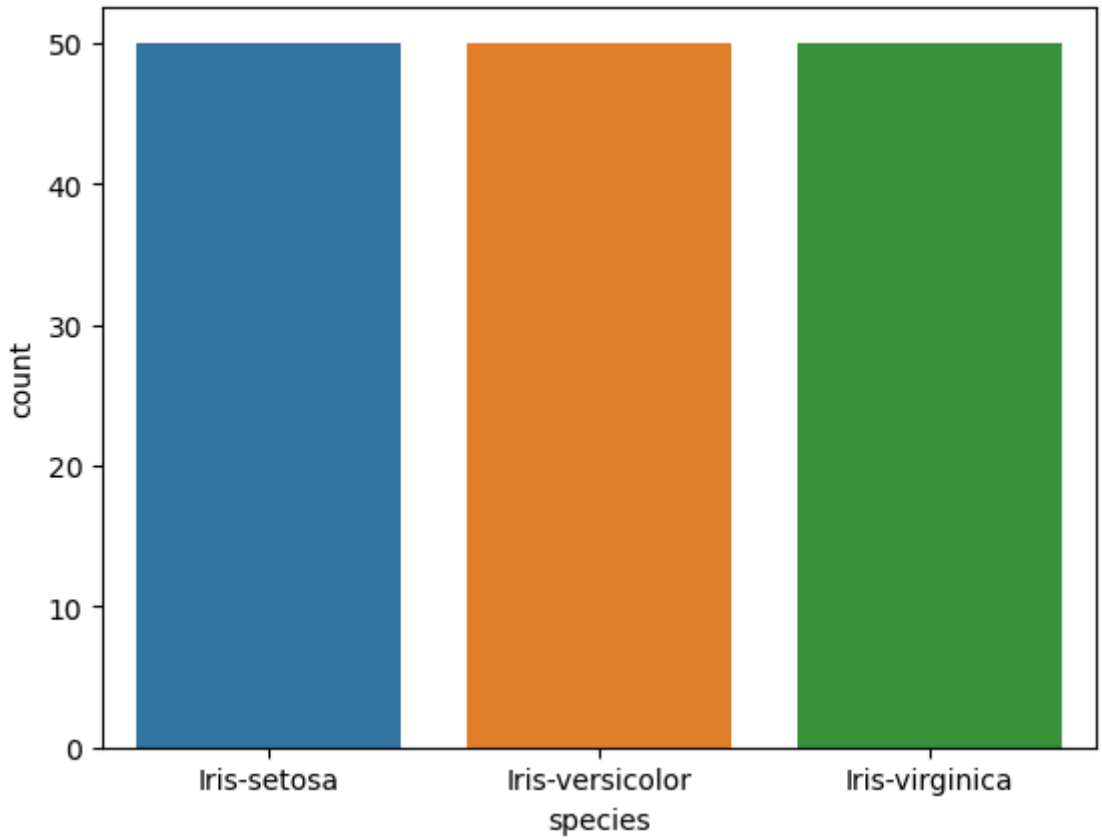
There are only three unique species.

In [7]:
```
df.value_counts("species")
```

Out[7]:
```
species
Iris-setosa        50
Iris-versicolor    50
Iris-virginica     50
dtype: int64
```

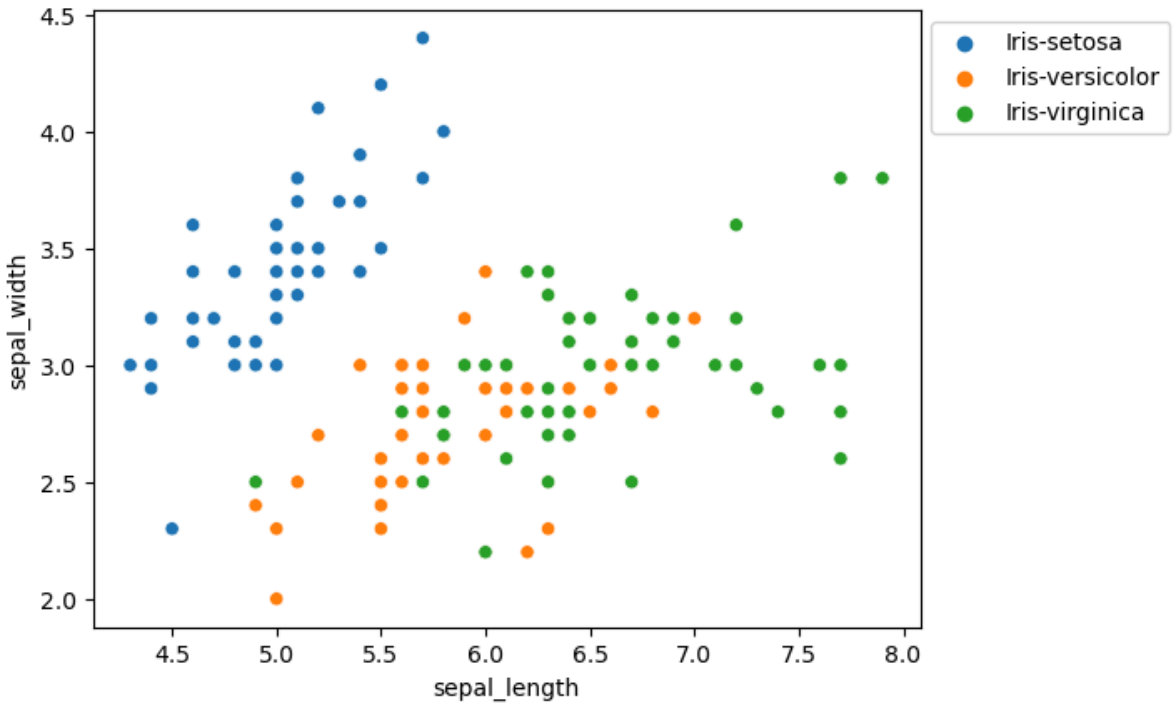
All the species contain an equal amount of rows, so we should not delete any entries.

In [8]:
```
sns.countplot(x='species', data=df )
plt.show()
```



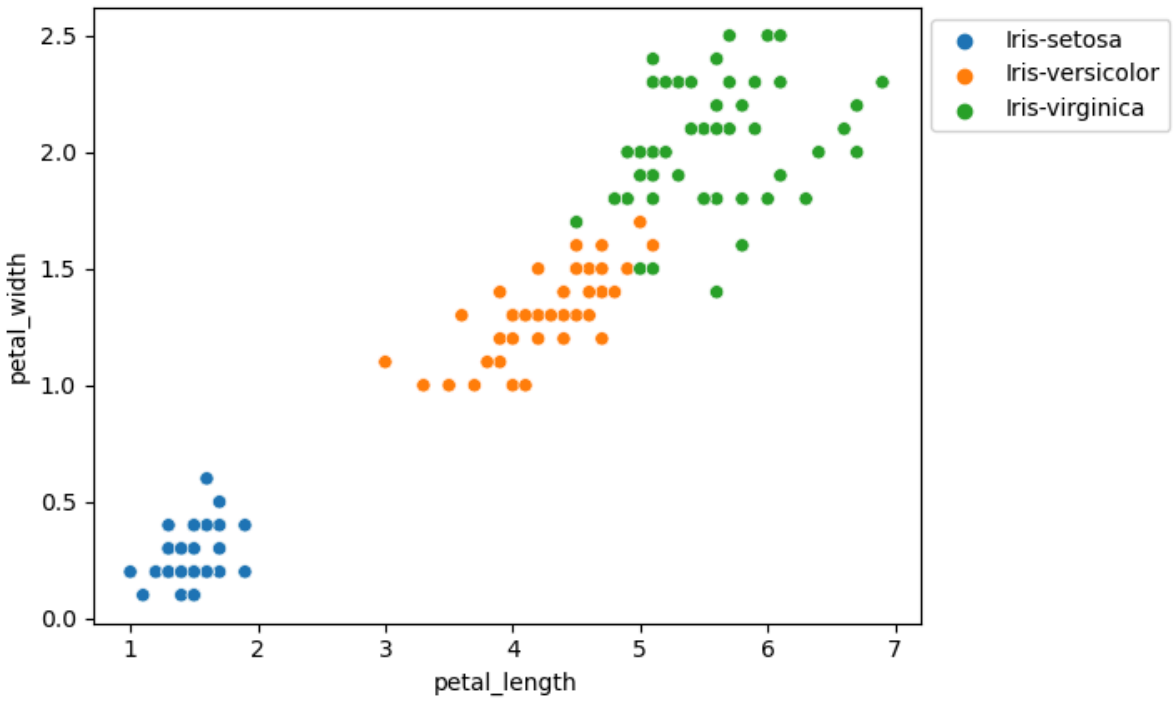
In [9]:
```
# Comparing Sepal Length and Sepal Width
sns.scatterplot(x='sepal_length', y='sepal_width',
                hue='species', data=df, )

# Placing Legend outside the Figure
plt.legend(bbox_to_anchor=(1, 1), loc=2)
```

```
plt.show()
```



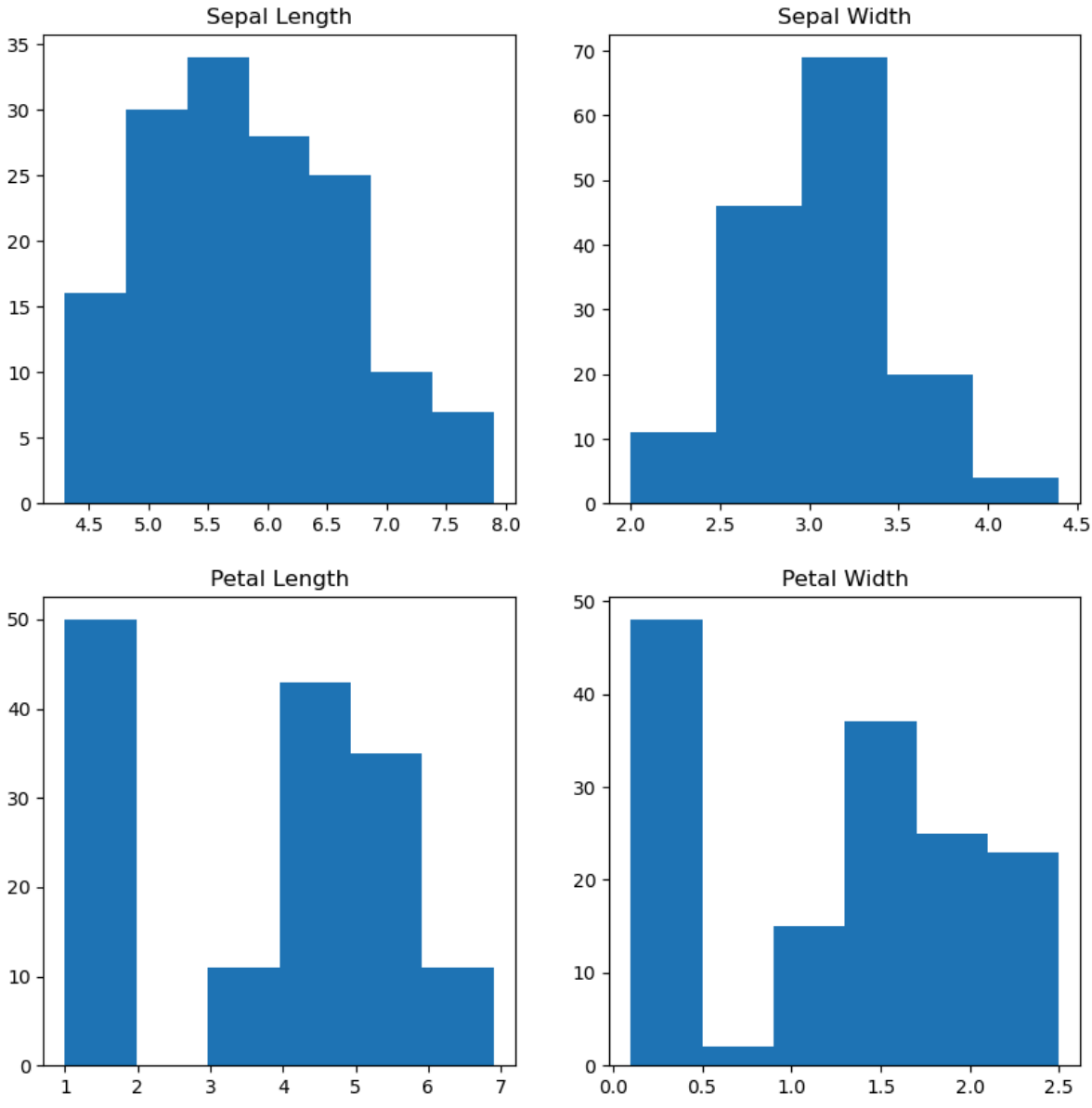
1. Species Setosa has smaller sepal lengths but larger sepal widths.
2. Versicolor Species lies in the middle of the other two species in terms of sepal length and width
3. Species Virginica has larger sepal lengths but smaller sepal widths.

In [10]:
```
# Comparing Petal Length and Petal Width
sns.scatterplot(x='petal_length', y='petal_width',
                hue='species', data=df, )

plt.legend(bbox_to_anchor=(1, 1), loc=2)

plt.show()
```



1. Species Setosa has smaller petal lengths and widths.
2. Versicolor Species lies in the middle of the other two species in terms of petal length and width
3. Species Virginica has the largest of petal lengths and widths.

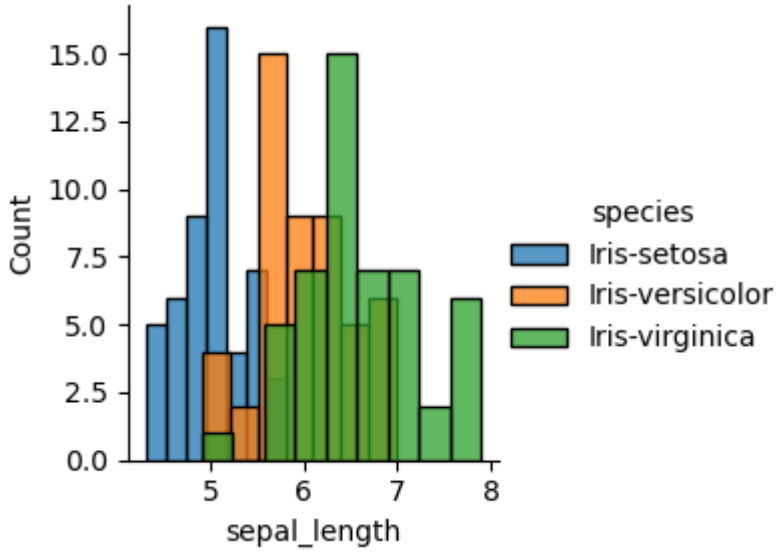
In [11]:
```
fig, axes = plt.subplots(2, 2, figsize=(10,10))

axes[0,0].set_title("Sepal Length")
axes[0,0].hist(df['sepal_length'], bins=7)

axes[0,1].set_title("Sepal Width")
axes[0,1].hist(df['sepal_width'], bins=5);

axes[1,0].set_title("Petal Length")
axes[1,0].hist(df['petal_length'], bins=6);

axes[1,1].set_title("Petal Width")
axes[1,1].hist(df['petal_width'], bins=6);
```

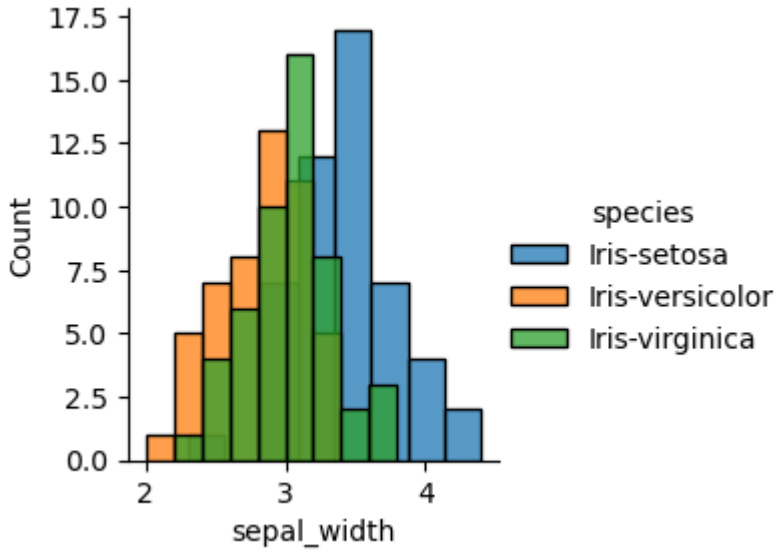


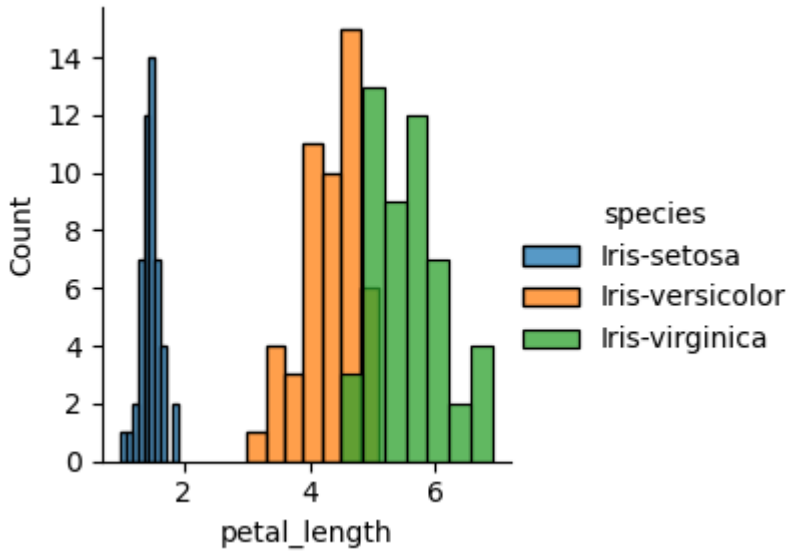
1. The highest frequency of the sepal length is between 30 and 35 which is between 5.5 and 6

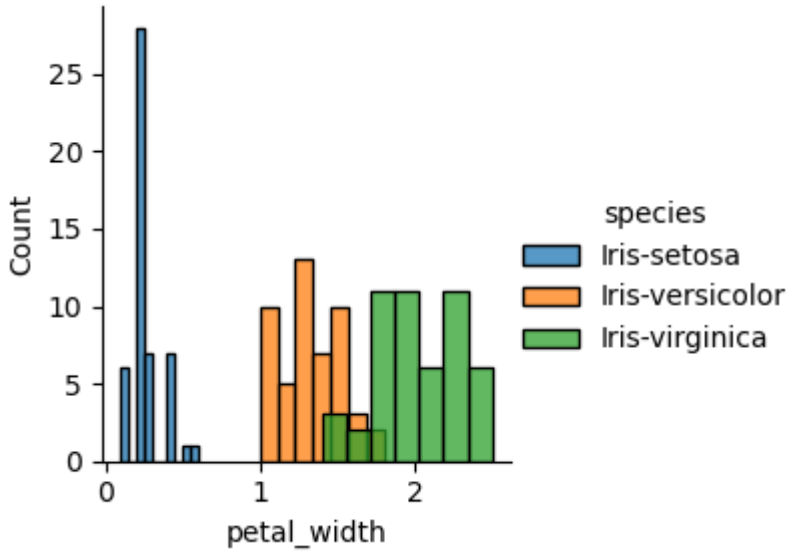
2. The highest frequency of the sepal Width is around 70 which is between 3.0 and 3.5
3. The highest frequency of the petal length is around 50 which is between 1 and 2
4. The highest frequency of the petal width is between 40 and 50 which is between 0.0 and 0.5

In [12]:
```
plot = sns.FacetGrid(df, hue="species")
plot.map(sns.histplot, "sepal_length").add_legend()

plot = sns.FacetGrid(df, hue="species")
plot.map(sns.histplot, "sepal_width").add_legend()

plot = sns.FacetGrid(df, hue="species")
plot.map(sns.histplot, "petal_length").add_legend()

plot = sns.FacetGrid(df, hue="species")
plot.map(sns.histplot, "petal_width").add_legend()

plt.show()
```









1. In the case of Sepal Length, there is a huge amount of overlapping.
2. In the case of Sepal Width also, there is a huge amount of overlapping.
3. In the case of Petal Length, there is a very little amount of overlapping.
4. In the case of Petal Width also, there is a very little amount of overlapping.

So we can use Petal Length and Petal Width as the classification feature.