

```
In [1]: 1 import pandas as pd
        2
        3 # Reading the CSV file
        4 df = pd.read_csv("Iris.csv")
        5
        6 # Printing top 5 rows
        7 df.head()
```

Out[1]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
In [2]: 1 df.shape
```

Out[2]: (150, 6)

```
In [3]: 1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Id               150 non-null    int64
1   SepalLengthCm    150 non-null    float64
2   SepalWidthCm     150 non-null    float64
3   PetalLengthCm    150 non-null    float64
4   PetalWidthCm     150 non-null    float64
5   Species          150 non-null    object
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

In [4]: 1 df.describe()

Out[4]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

In [5]: 1 df.isnull().sum()

Out[5]: Id 0
 SepalLengthCm 0
 SepalWidthCm 0
 PetalLengthCm 0
 PetalWidthCm 0
 Species 0
 dtype: int64

In [6]: 1 data = df.drop_duplicates(subset ="Species",)
 2 data

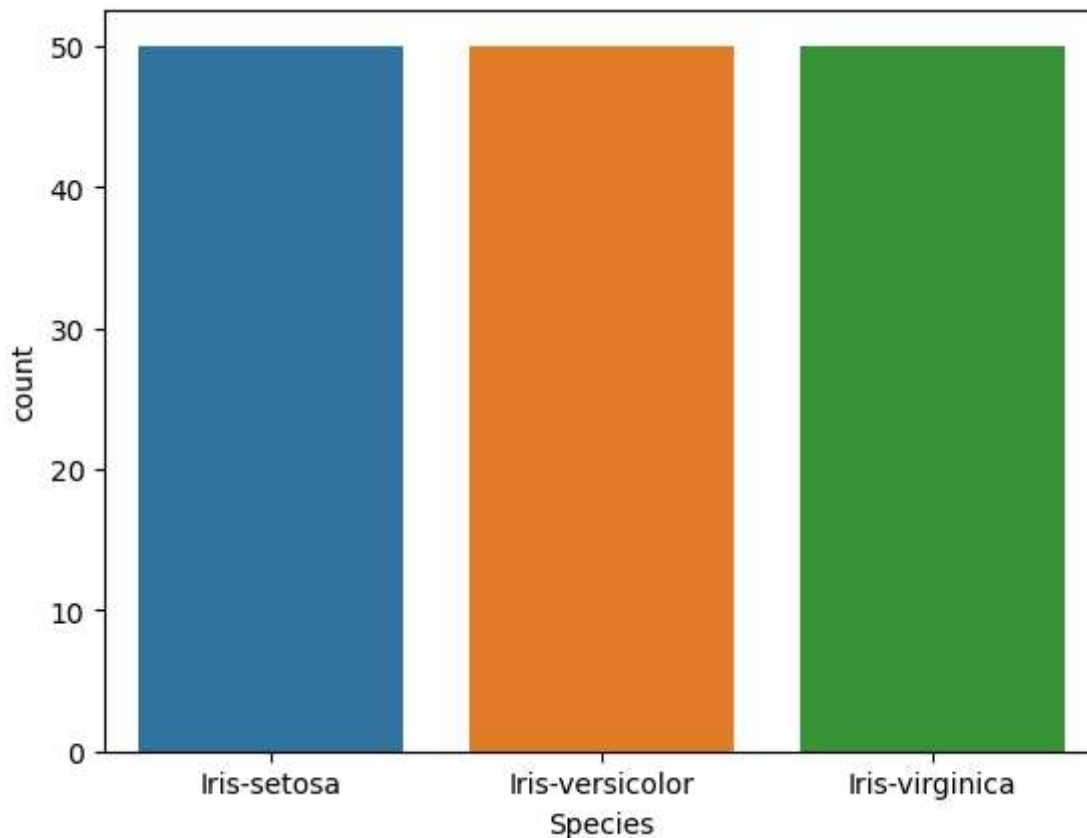
Out[6]:

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
50	51	7.0	3.2	4.7	1.4	Iris-versicolor
100	101	6.3	3.3	6.0	2.5	Iris-virginica

In [7]: 1 df.value_counts("Species")

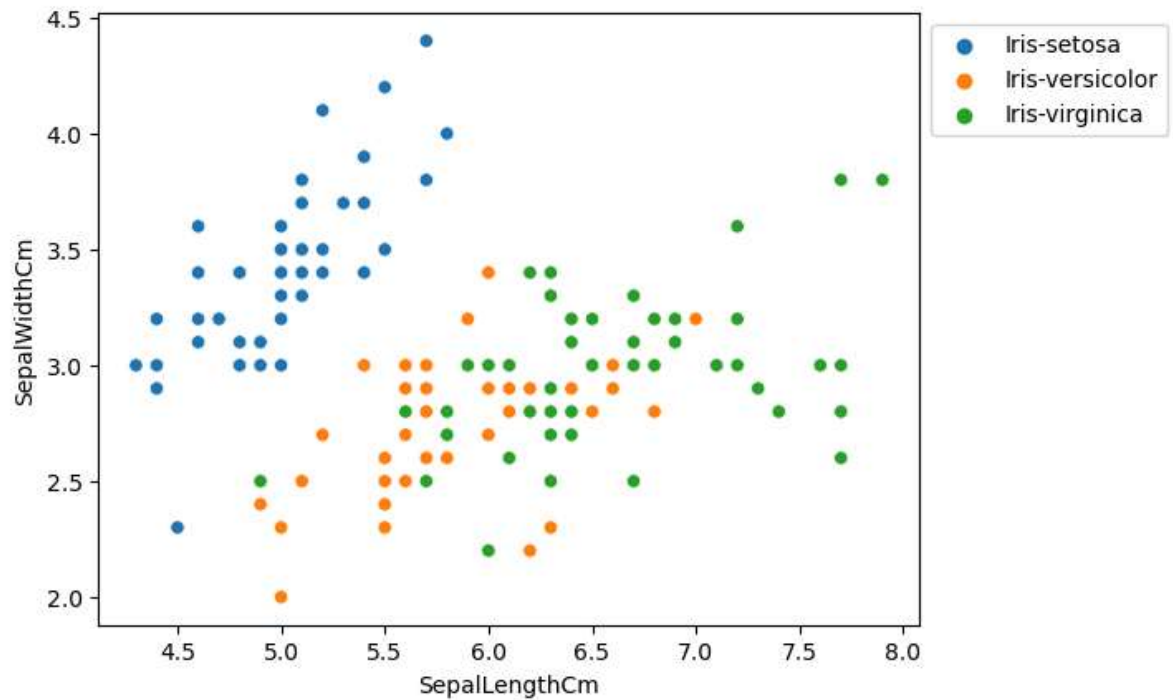
Out[7]: Species
 Iris-setosa 50
 Iris-versicolor 50
 Iris-virginica 50
 dtype: int64

```
In [8]: 1 # importing packages
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4
5
6 sns.countplot(x='Species', data=df, )
7 plt.show()
```



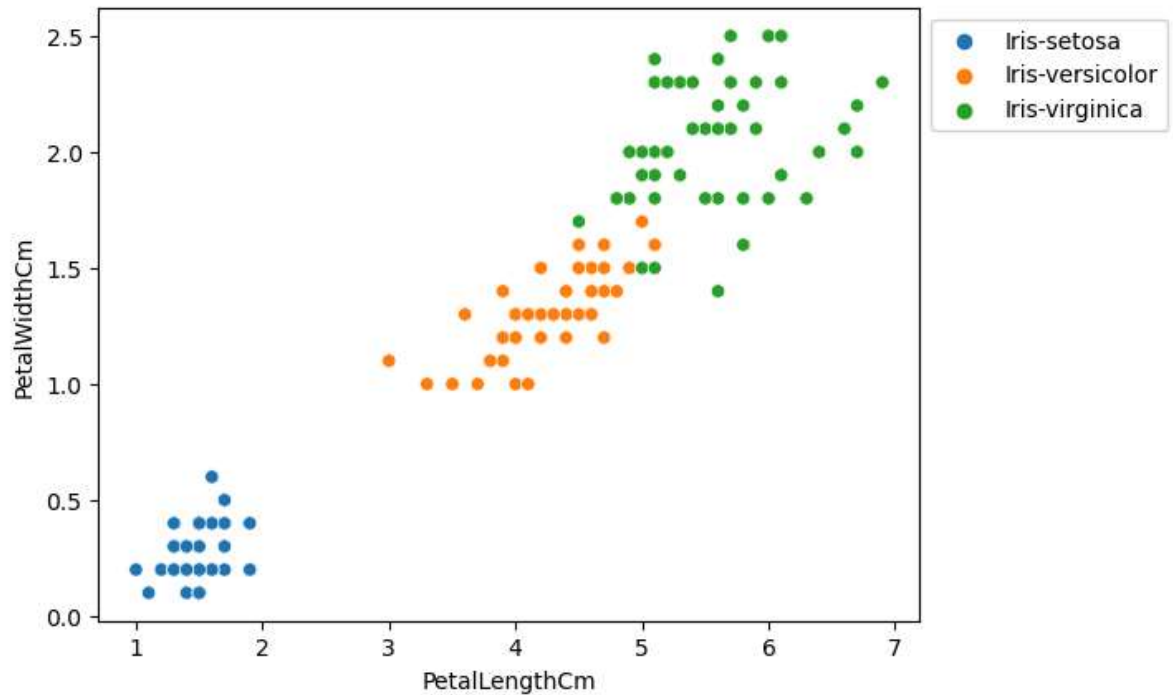
Comparing Sepal Length and Sepal Width

```
In [9]: 1 # importing packages
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4
5
6 sns.scatterplot(x='SepalLengthCm', y='SepalWidthCm',
7               hue='Species', data=df, )
8
9 # Placing Legend outside the Figure
10 plt.legend(bbox_to_anchor=(1, 1), loc=2)
11
12 plt.show()
```



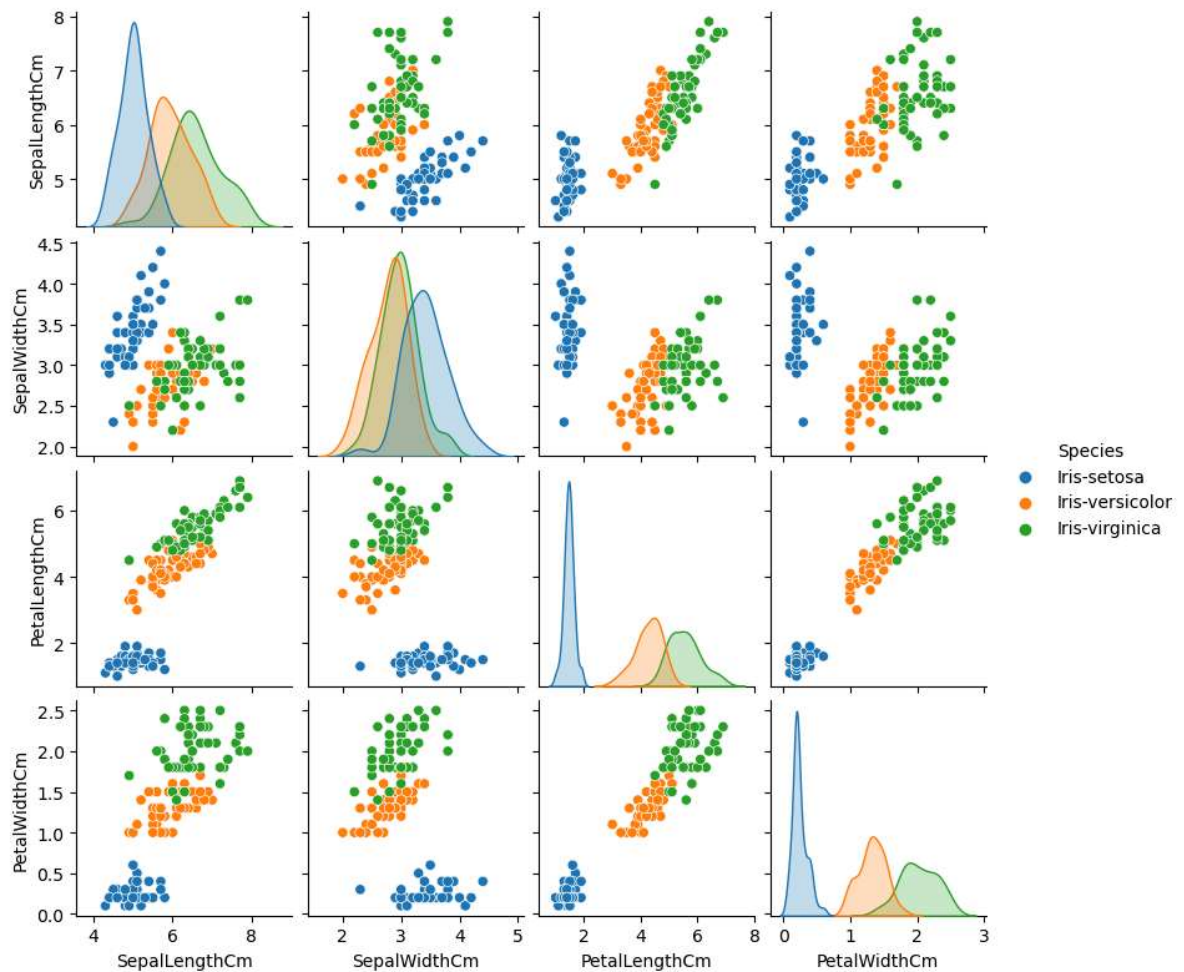
Comparing Petal Length and Petal Width

```
In [10]: 1 # importing packages
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4
5
6 sns.scatterplot(x='PetalLengthCm', y='PetalWidthCm',
7               hue='Species', data=df, )
8
9 # Placing Legend outside the Figure
10 plt.legend(bbox_to_anchor=(1, 1), loc=2)
11
12 plt.show()
```



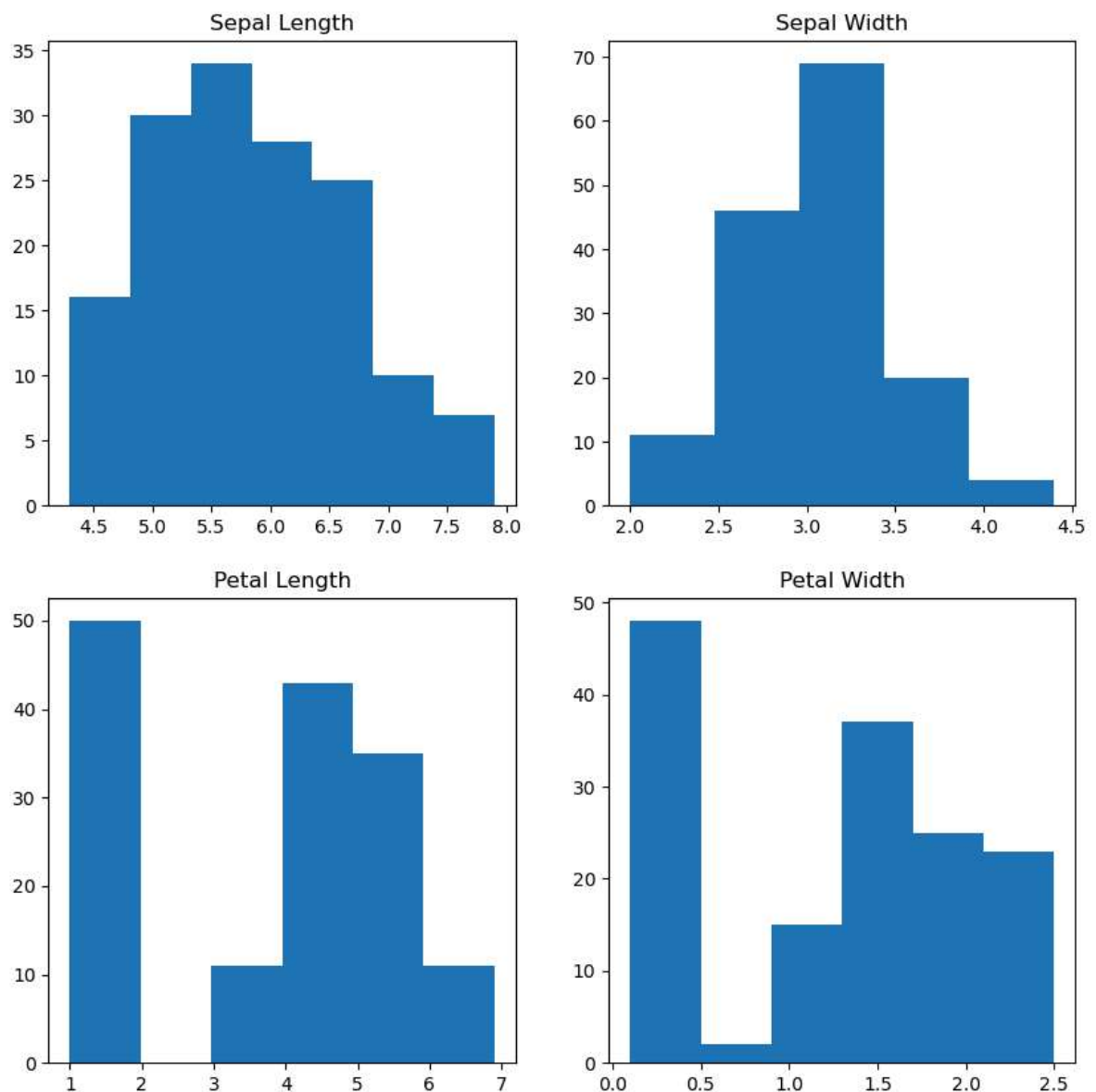
```
In [11]: 1 # importing packages
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4
5
6 sns.pairplot(df.drop(['Id'], axis = 1),
7             hue='Species', height=2)
```

Out[11]: <seaborn.axisgrid.PairGrid at 0x1cf044d7110>



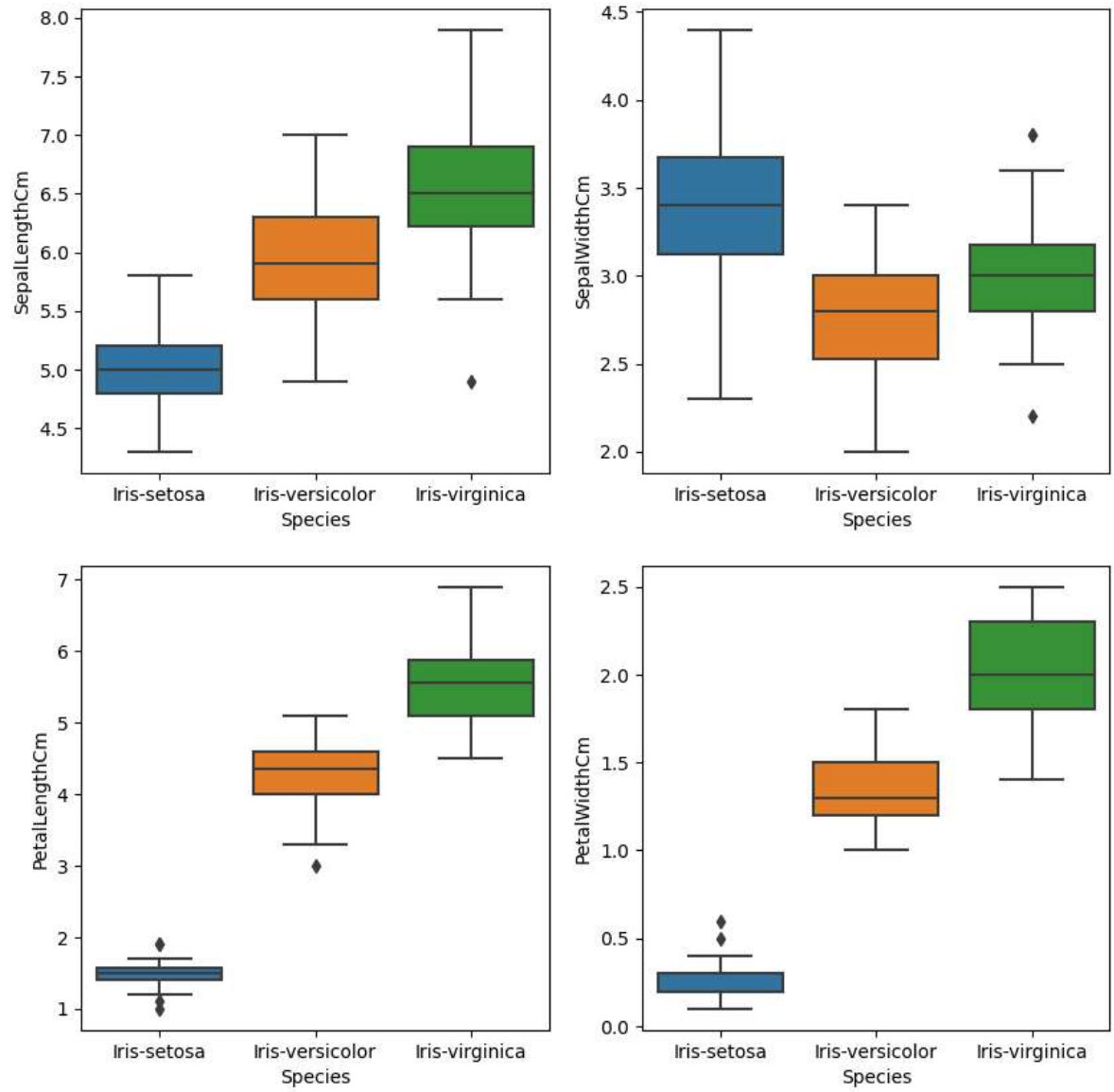
Histograms

```
In [12]: 1 # importing packages
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4
5
6 fig, axes = plt.subplots(2, 2, figsize=(10,10))
7
8 axes[0,0].set_title("Sepal Length")
9 axes[0,0].hist(df['SepalLengthCm'], bins=7)
10
11 axes[0,1].set_title("Sepal Width")
12 axes[0,1].hist(df['SepalWidthCm'], bins=5);
13
14 axes[1,0].set_title("Petal Length")
15 axes[1,0].hist(df['PetalLengthCm'], bins=6);
16
17 axes[1,1].set_title("Petal Width")
18 axes[1,1].hist(df['PetalWidthCm'], bins=6);
```



Box Plots


```
In [13]: 1 # importing packages
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4
5 def graph(y):
6     sns.boxplot(x="Species", y=y, data=df)
7
8 plt.figure(figsize=(10,10))
9
10 # Adding the subplot at the specified
11 # grid position
12 plt.subplot(221)
13 graph('SepalLengthCm')
14
15 plt.subplot(222)
16 graph('SepalWidthCm')
17
18 plt.subplot(223)
19 graph('PetalLengthCm')
20
21 plt.subplot(224)
22 graph('PetalWidthCm')
23
24 plt.show()
```



In []:

1