```
#Import some libraries to perform some calculations, visualization, plotting, remove warnings and other usage of functions

import pandas as pd
import matplotlib.pyplot as plt
import seaborn  as sns
from scipy import stats
import numpy as np
import warnings
warnings.filterwarnings("ignore")
```

```
#Load the dataset of Iris Species and stored in variable called iris:

iris = pd.read_csv("/content/dataset.csv")
iris
```

	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
...	...	...	...	...	...	...
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

150 rows × 6 columns

Next steps: Generate code with iris | View recommended plots | New interactive sheet

Basic Pandas

```
#This command gives the information of given dataset:

iris.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
```

```
#This command gives the static information of given dataset:

iris.describe()
```

	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

```
#This command shows the columns of given dataset:

iris.columns
```

```
Index(['Id', 'SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm',
       'Species'],
      dtype='object')
```

```
#This command shows the order pair of given dataset:

iris.shape
```

```
(150, 6)
```

```
#This command shows the first 5 rows of given dataset:

iris.head()
```

	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

Next steps: Generate code with iris | View recommended plots | New interactive sheet

```
#This command shows the last 5 rows of given dataset:

iris.tail()
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
145	146	6.7	3.0	5.2	2.3	Iris-virginica
146	147	6.3	2.5	5.0	1.9	Iris-virginica
147	148	6.5	3.0	5.2	2.0	Iris-virginica
148	149	6.2	3.4	5.4	2.3	Iris-virginica
149	150	5.9	3.0	5.1	1.8	Iris-virginica

```
#This command gives the missing values of given dataset:

iris.isnull().sum()
```

	0
Id	0
SepalLengthCm	0
SepalWidthCm	0
PetalLengthCm	0
PetalWidthCm	0
Species	0

dtype: int64

```
#This command counts the value of every columns individually:

iris.count()
```

	0
Id	150
SepalLengthCm	150
SepalWidthCm	150
PetalLengthCm	150
PetalWidthCm	150
Species	150

dtype: int64

```
#This command counts the value of target column:

iris.Species.value_counts()
```

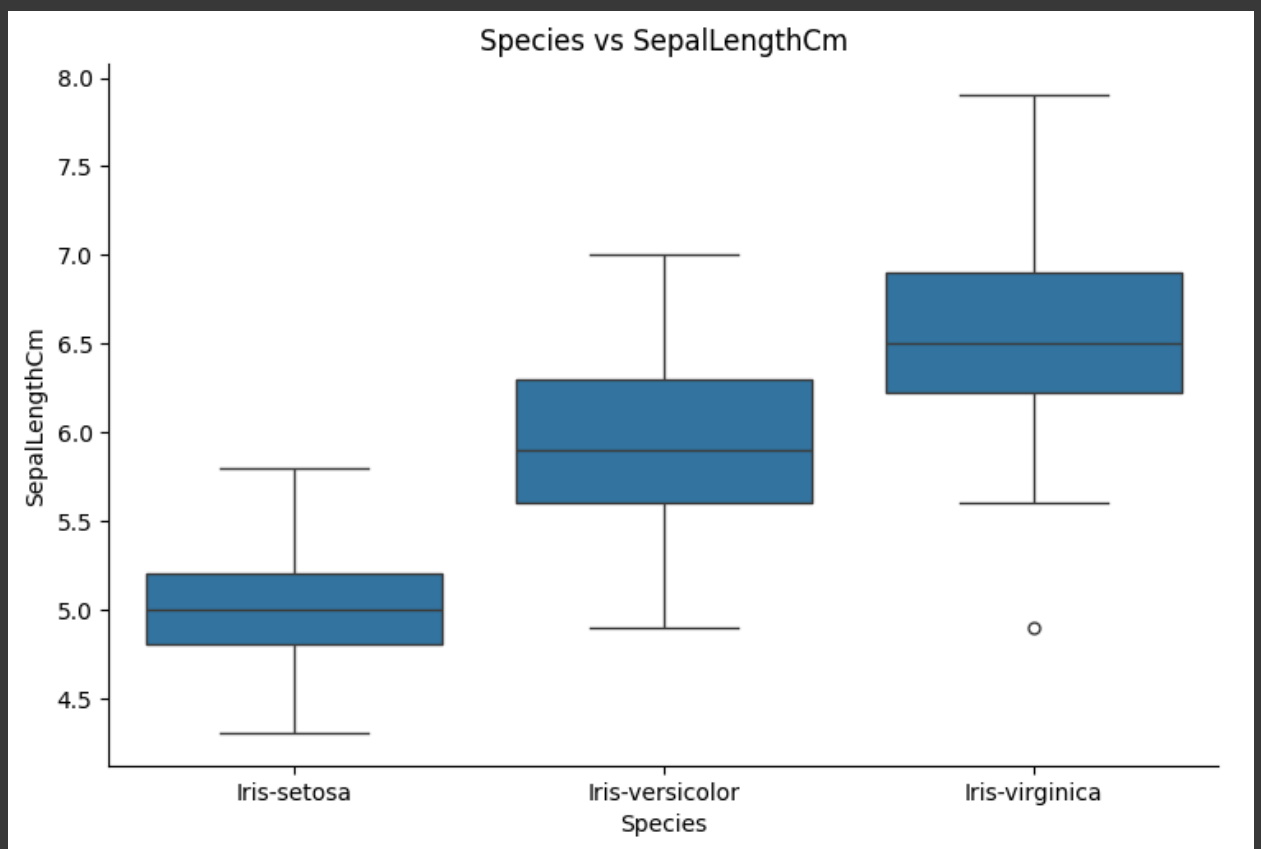
Species	count
Iris-setosa	50
Iris-versicolor	50
Iris-virginica	50

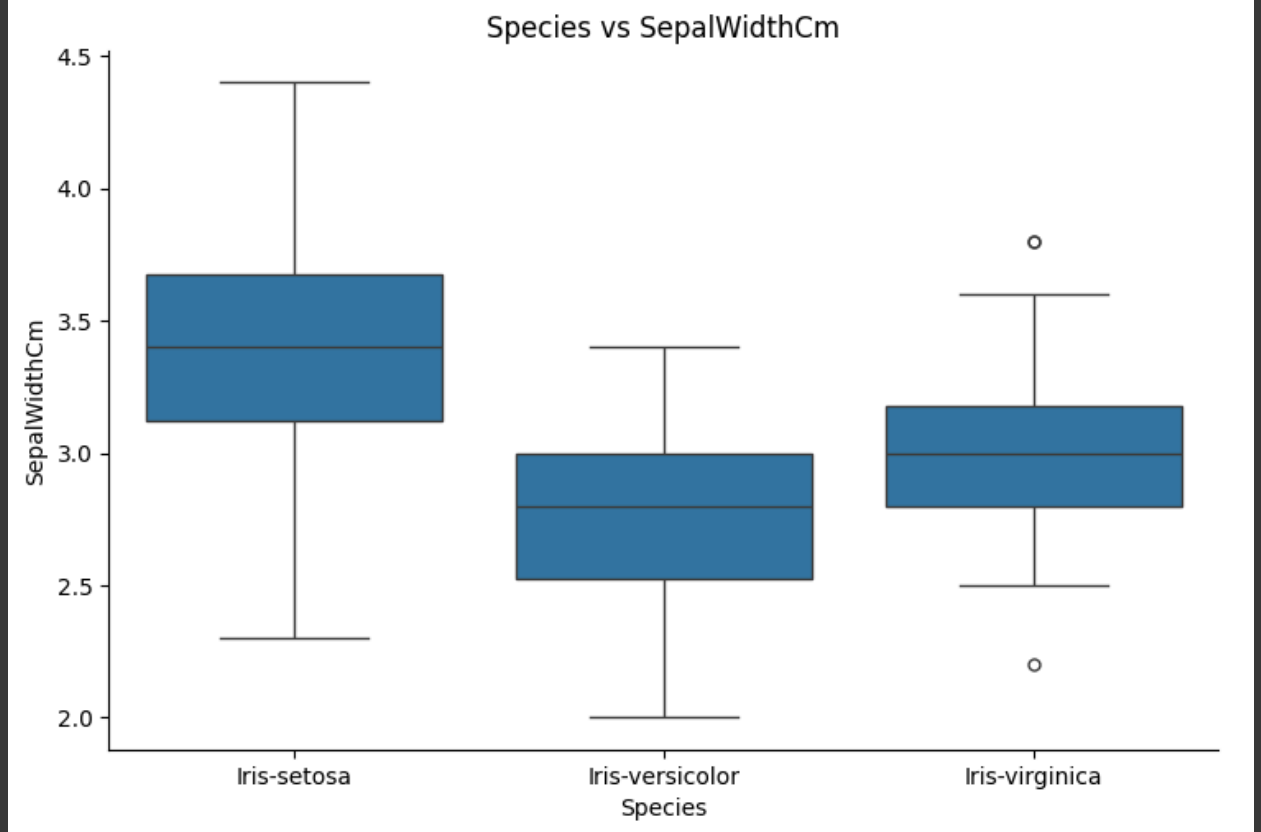
dtype: int64

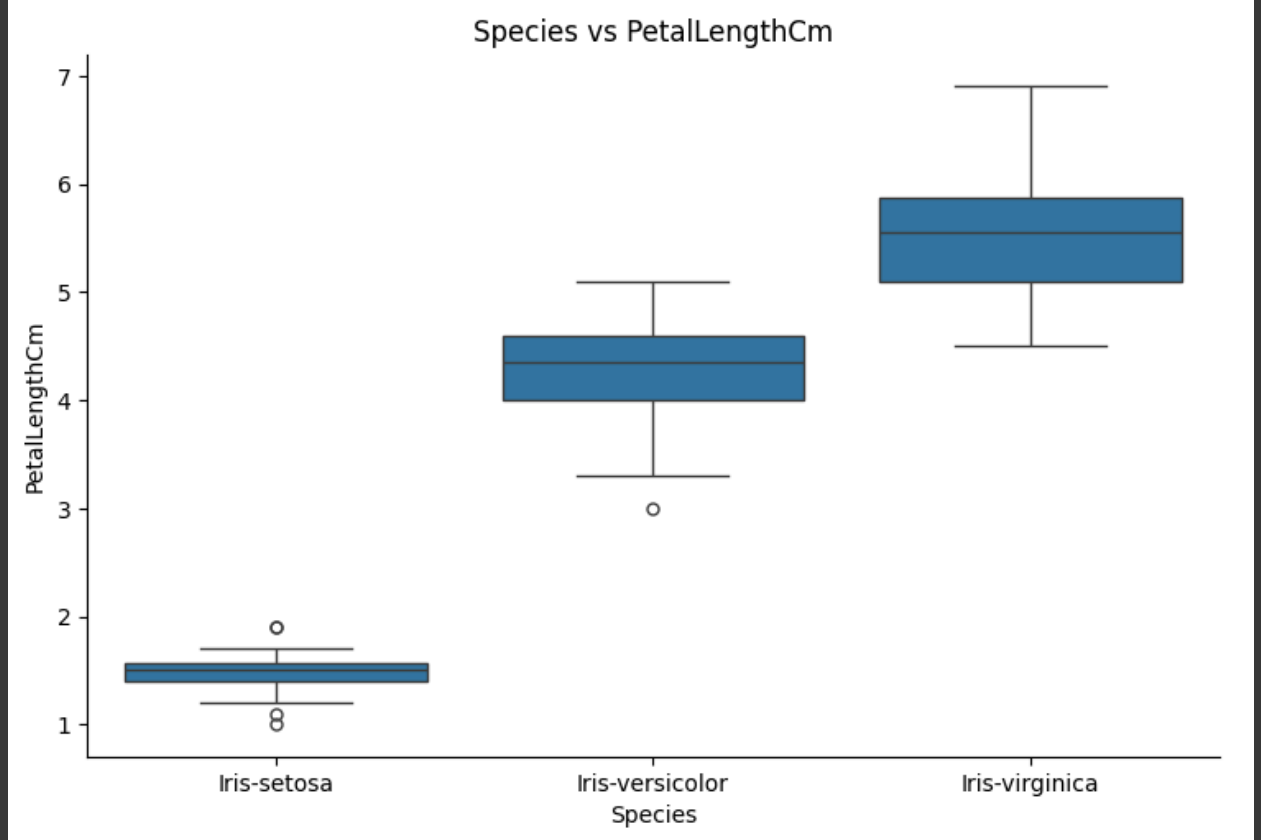
Visualization:

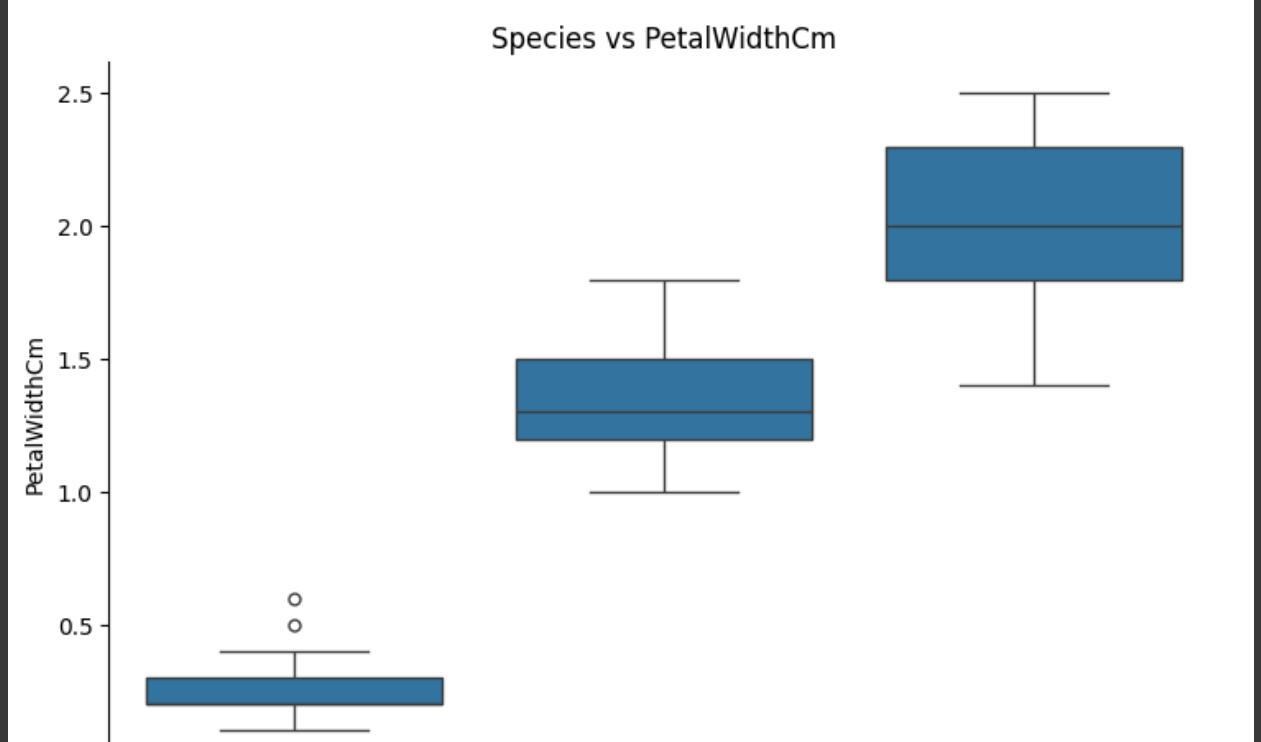
```
# This command shows the relationship between target and other columns in the form of catplot:

columns = ['SepalLengthCm' ,  'SepalWidthCm' ,  'PetalLengthCm',  'PetalWidthCm']
for i in range(0,4,1):
  sns.catplot(x= "Species", y= columns[i],  data = iris, kind = "box", aspect = 1.5 )
  plt.title(f"Species vs %s" %columns[i] )
  plt.show()
```

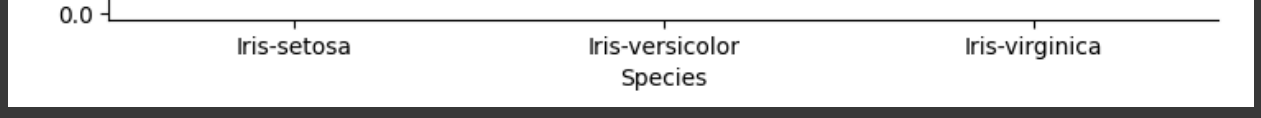
[+]


Species vs SepalLengthCm
SepalLengthCm
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
Iris-setosa
Iris-versicolor
Iris-virginica
Species

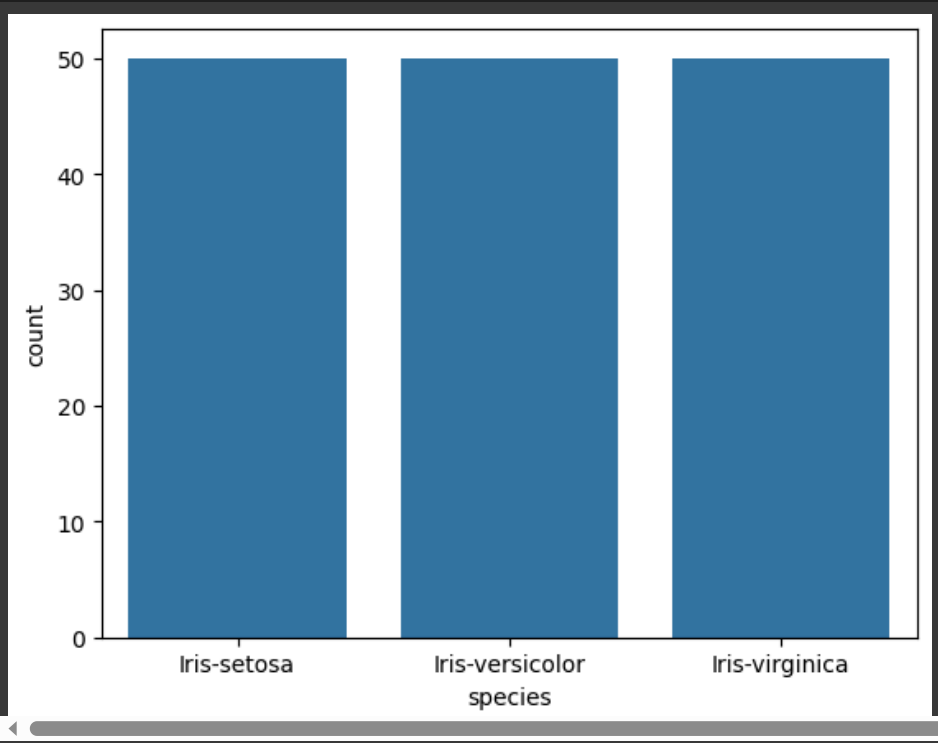

Species vs SepalWidthCm
SepalWidthCm
4.5
4.0
3.5
3.0
2.5
2.0
Iris-setosa
Iris-versicolor
Iris-virginica
Species


Species vs PetalLengthCm
PetalLengthCm
7
6
5
4
3
2
1
Iris-setosa
Iris-versicolor
Iris-virginica
Species


Species vs PetalWidthCm
PetalWidthCm
2.5
2.0
1.5
1.0
0.5




```
# This command shows the value of target column in the form of graph:

sns.countplot(x = 'Species', data = iris)
plt.xlabel('species')
plt.show()
```

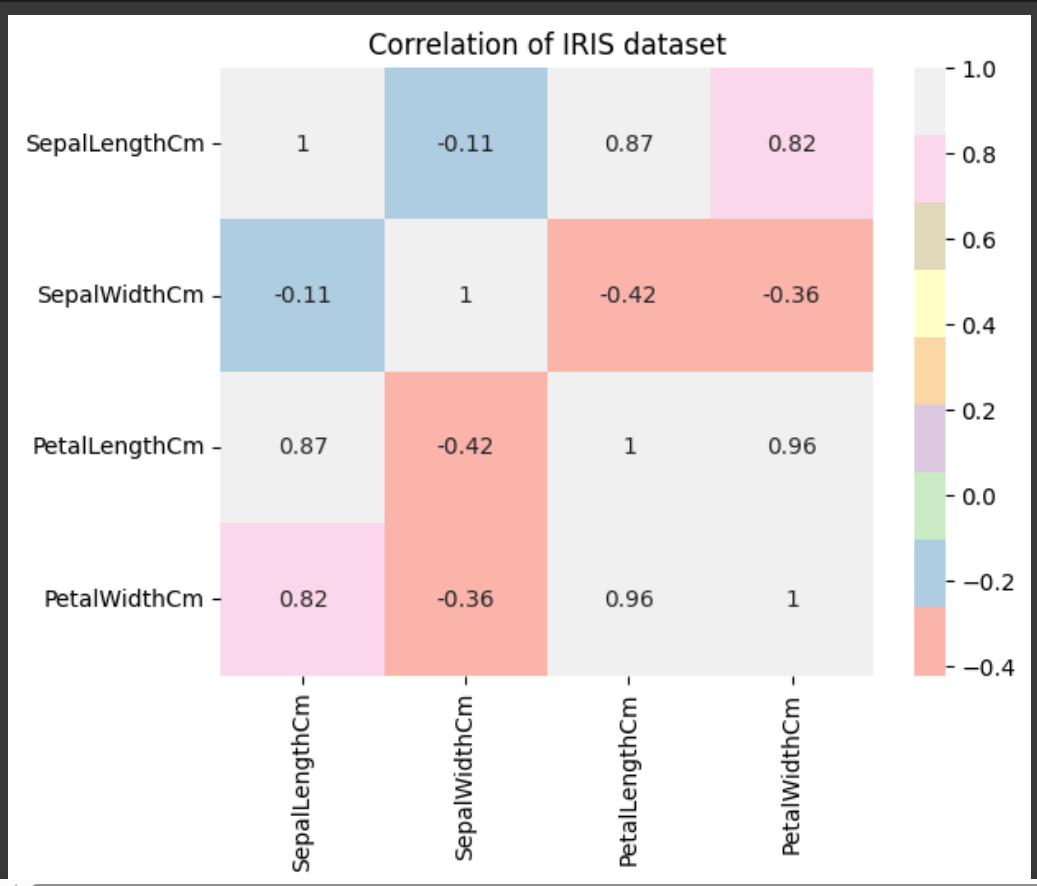



```
#This command is used to identify the co-relation of entire dataset:

# Exclude non-numeric columns ('Id' and 'Species') before calculating correlation
correlation = iris.drop(['Id', 'Species'], axis=1).corr()
display(correlation)
```

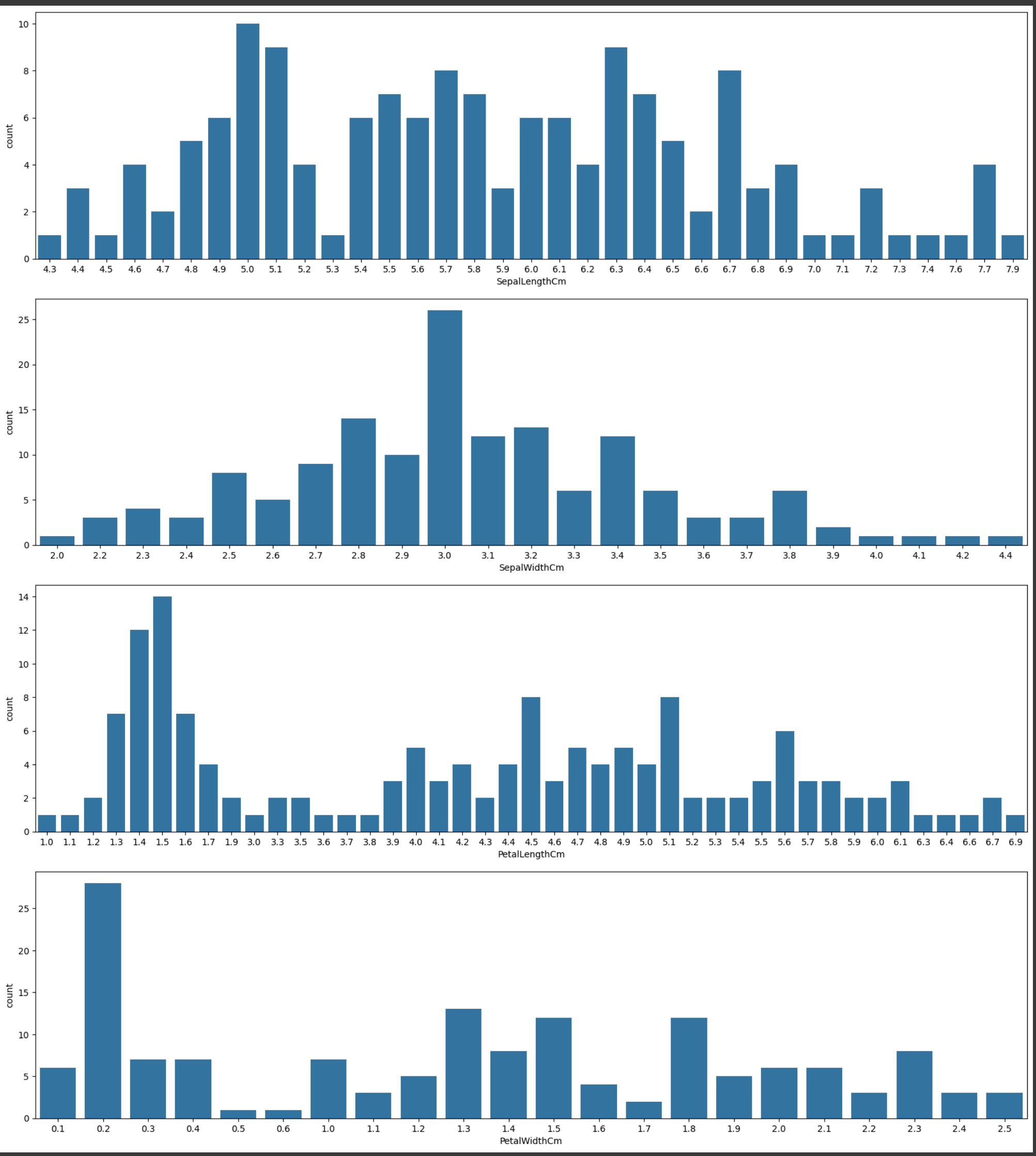
	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.000000	-0.109369	0.871754	0.817954
SepalWidthCm	-0.109369	1.000000	-0.420516	-0.356544
PetalLengthCm	0.871754	-0.420516	1.000000	0.962757
PetalWidthCm	0.817954	-0.356544	0.962757	1.000000

Next steps: Generate code with correlation | View recommended plots | New interactive sheet

```
#This command convert the co-relation into heatmap:
#Graphical view of co-relation from heatmap:

correlation = iris.drop(['Id', 'Species'], axis=1).corr()
sns.heatmap(correlation , annot = True , cmap = 'Pastel1')
plt.title('Correlation of IRIS dataset')
plt.show()
```

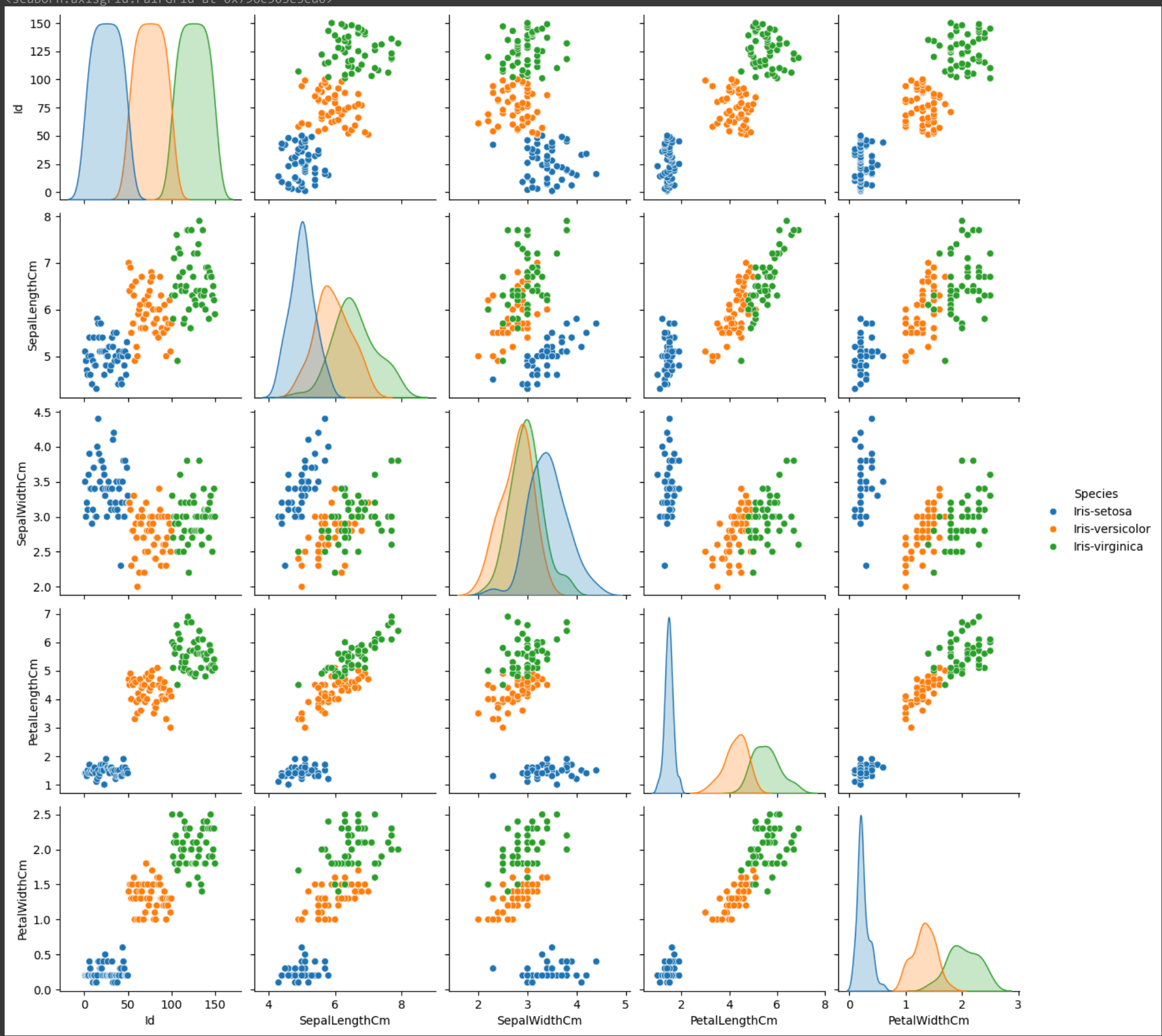



```
#This command draws the countplot in some columns:

arr =  ['SepalLengthCm' , 'SepalWidthCm' , 'PetalLengthCm' , 'PetalWidthCm']
for i in range (0, 4, 1):
  plt.figure ( figsize = (20,5))
  sns.countplot( x = arr[i]  , data = iris)
  plt.show()
```

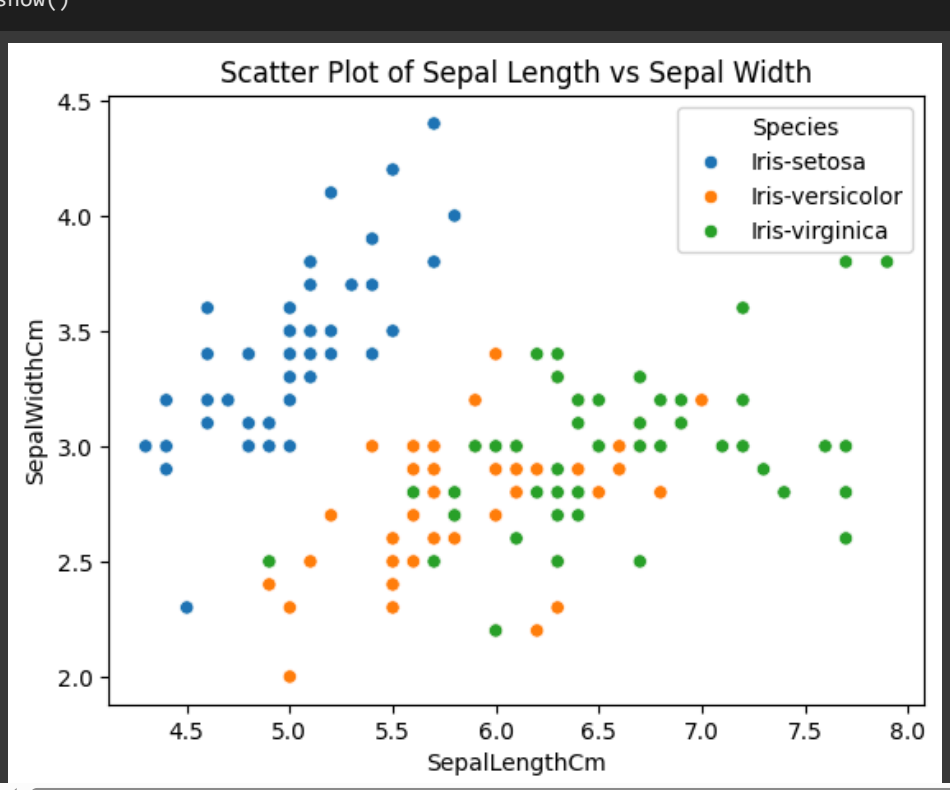



```
#This command draw the pair plot of entire dataset:

sns.pairplot(iris, hue = 'Species', size = 2.5)
```

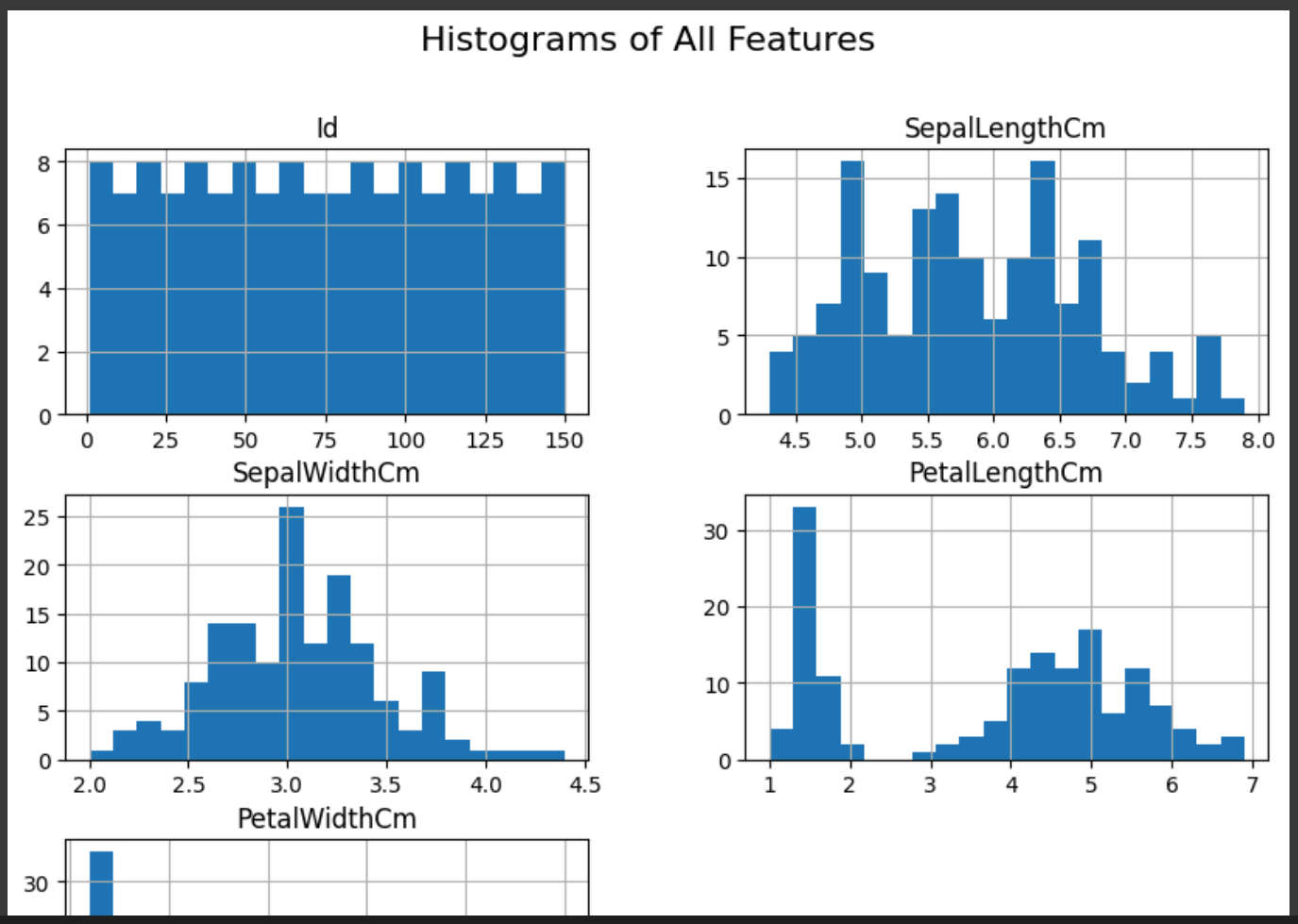
<seaborn.axisgrid.PairGrid at 0x796c503e3ed0>



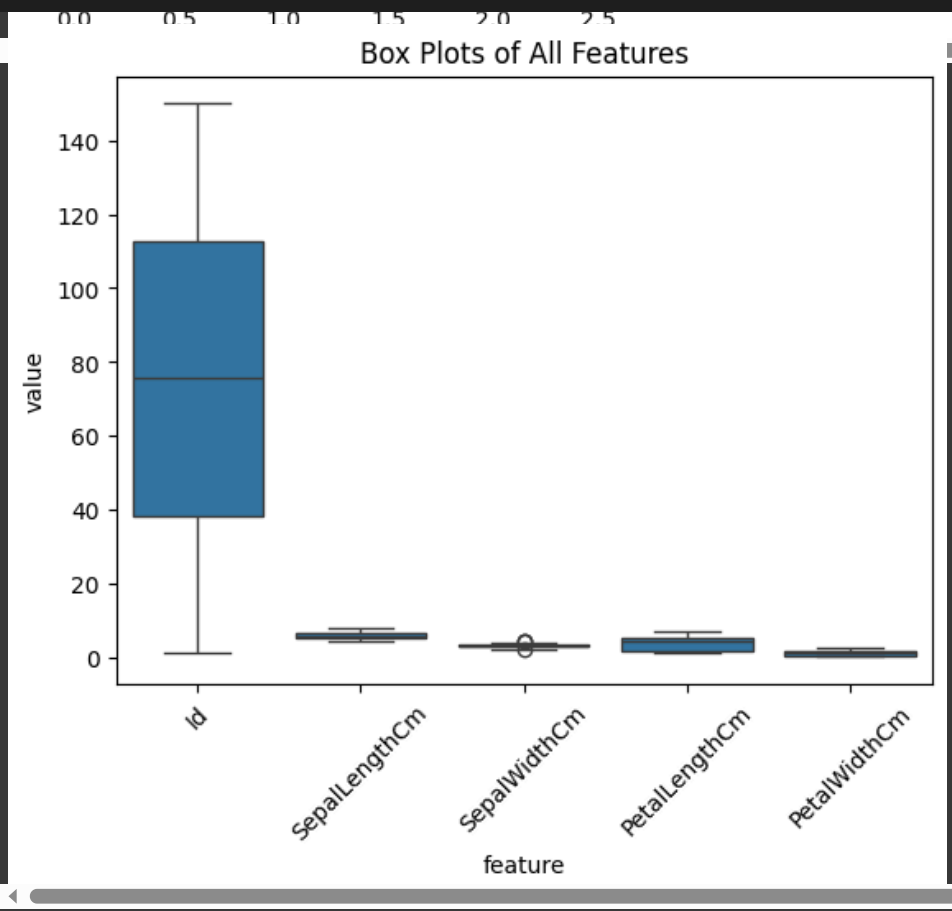

```
sns.scatterplot(data=iris, x='SepalLengthCm', y='SepalWidthCm', hue='Species')
plt.title('Scatter Plot of Sepal Length vs Sepal Width')
plt.show()
```




```
iris.hist(figsize=(10, 8), bins=20)
plt.suptitle('Histograms of All Features', fontsize=16)
plt.show()
```




```
# Multiple features box plot (melt for seaborn)
df_melted = iris.melt(id_vars='Species', var_name='feature', value_name='value')
sns.boxplot(data=df_melted, x='feature', y='value')
plt.xticks(rotation=45)
plt.title('Box Plots of All Features')
plt.show()
```




```
sns.boxplot(x=iris['PetalLengthCm'])
plt.title('Box Plot of Petal Length')
plt.show()
```