In [28]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt


from warnings import filterwarnings
filterwarnings(action='ignore')

iris = pd.read_csv("IRIS.csv")
```

In [3]:
```python
print(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
```

In [4]:
```python
print(iris.describe())
```

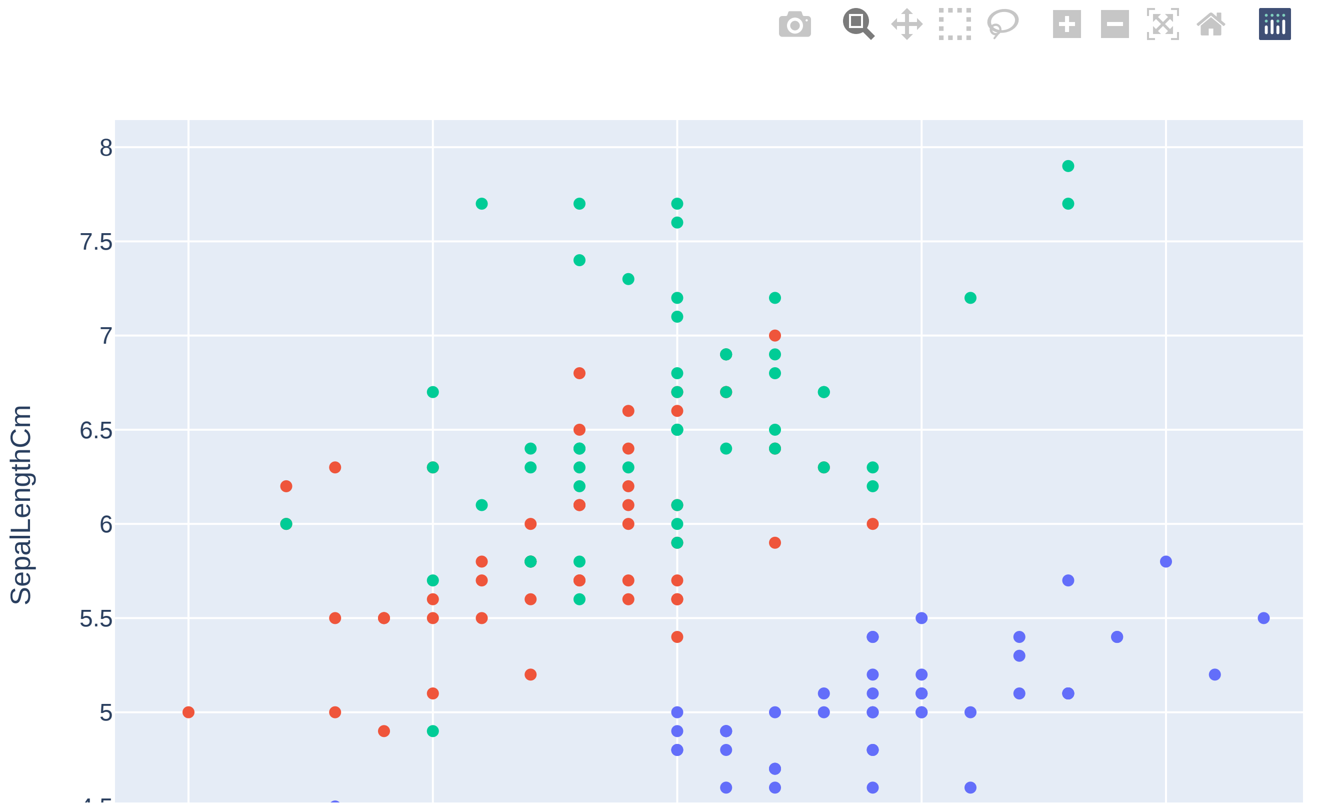
```
               Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm
count  150.000000     150.000000    150.000000     150.000000    150.000000
mean    75.500000       5.843333      3.054000       3.758667      1.198667
std     43.445368       0.828066      0.433594       1.764420      0.763161
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
```

In [6]:
```python
print("Target Labels", iris["Species"].unique())
```

```
Target Labels ['Iris-setosa' 'Iris-versicolor' 'Iris-virginica']
```

In [7]:
```python
import plotly.express as px
fig = px.scatter(iris, x="SepalWidthCm", y="SepalLengthCm", color="Species")
fig.show()
```

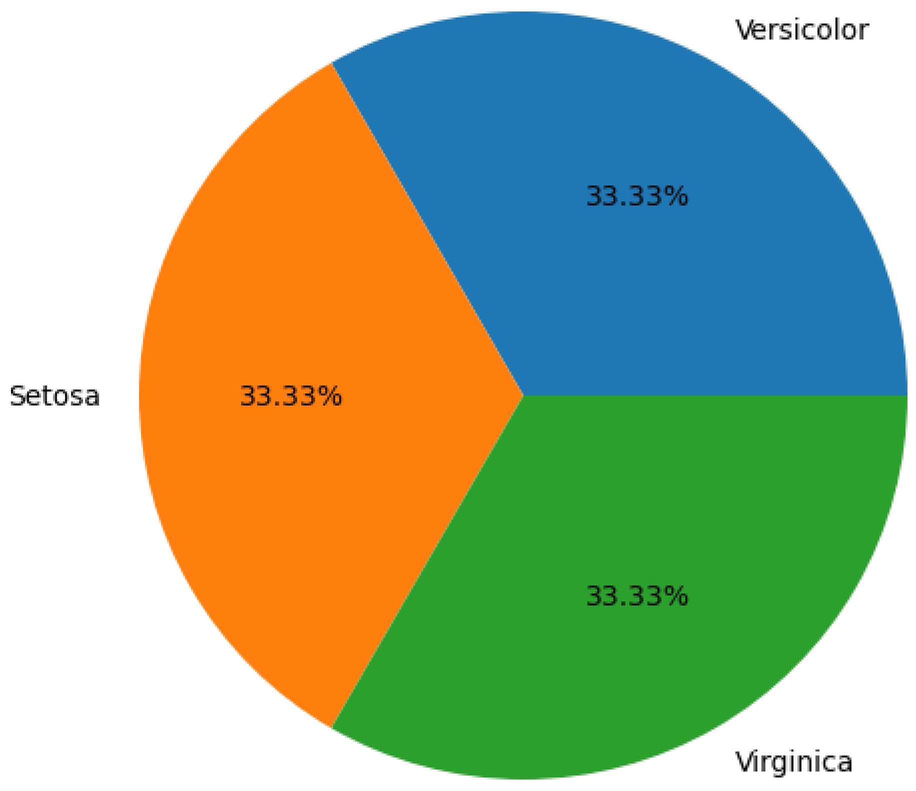
Loading [MathJax]/extensions/Safe.js



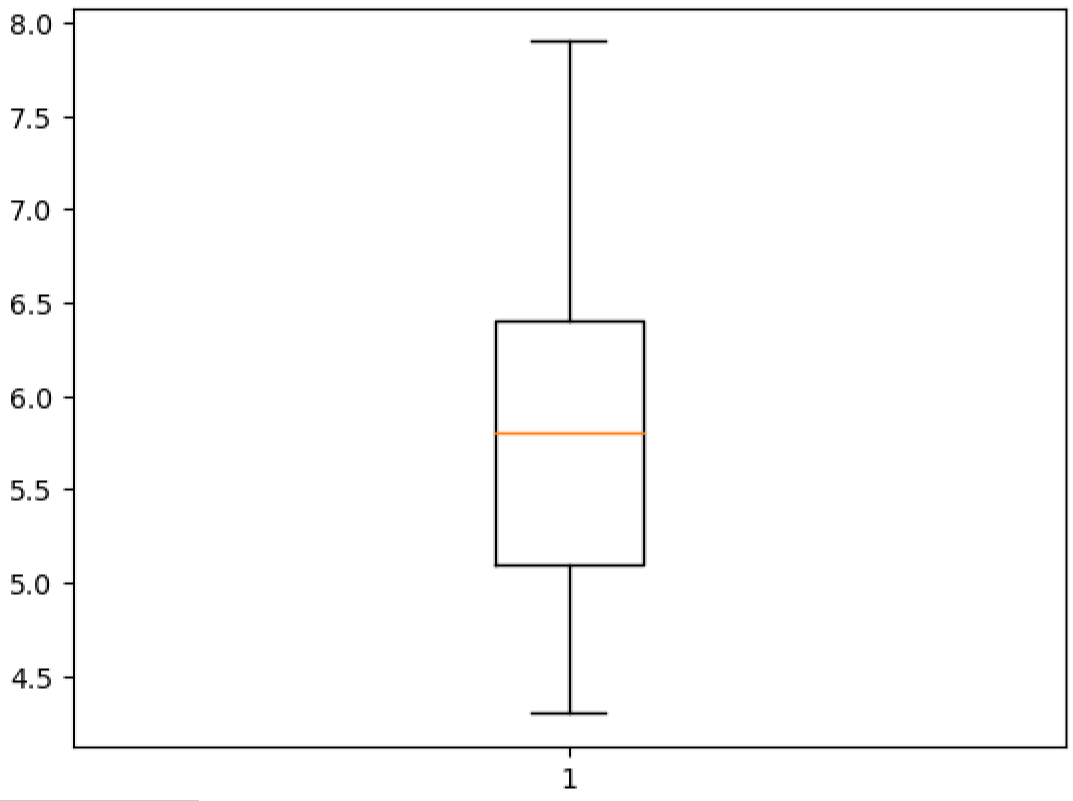
```
In [13]: n = len(iris[iris['Species'] == 'versicolor'])
         print("No of Versicolor in Dataset:",n)

         n1 = len(iris[iris['Species'] == 'virginica'])
         print("No of Virginica in Dataset:",n1)

         n2 = len(iris[iris['Species'] == 'setosa'])
         print("No of Setosa in Dataset:",n2)

         fig = plt.figure()
         ax = fig.add_axes([0,0,1,1])
         ax.axis('equal')
         l = ['Versicolor', 'Setosa', 'Virginica']
         s = [50,50,50]
         ax.pie(s, labels = l,autopct='%1.2f%%')
         plt.show()
```

```
No of Versicolor in Dataset: 0
No of Virginica in Dataset: 0
No of Setosa in Dataset: 0
```

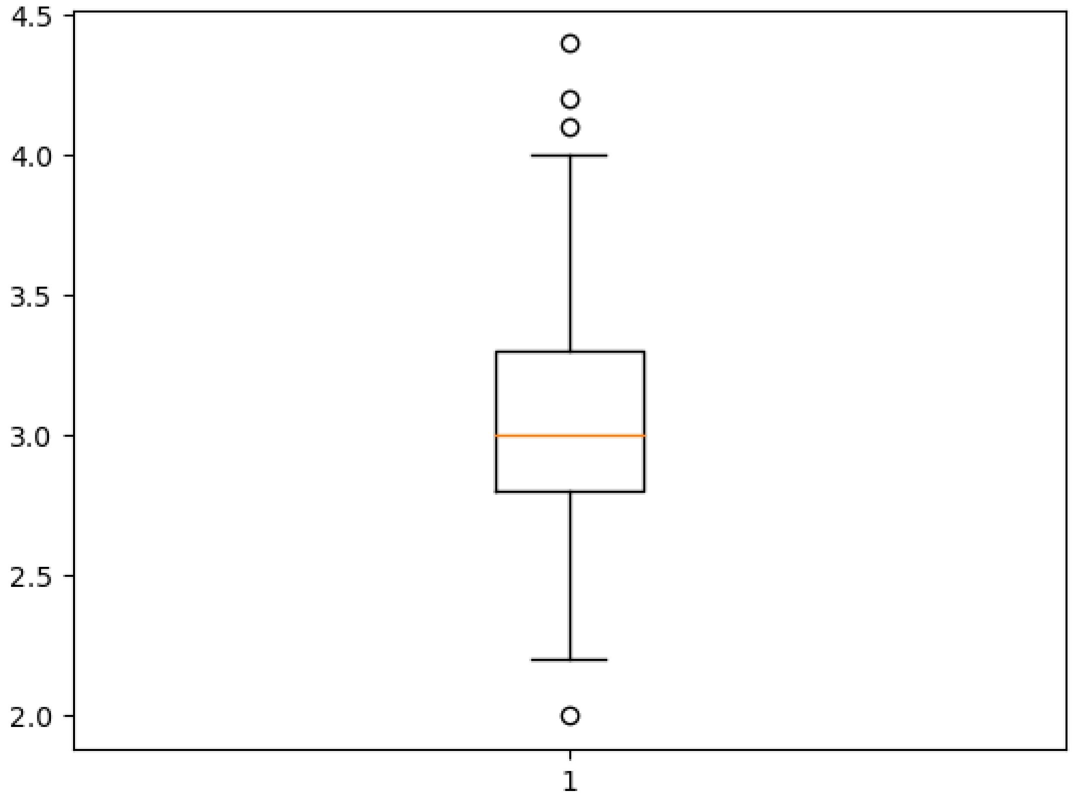
Loading [MathJax]/extensions/Safe.js



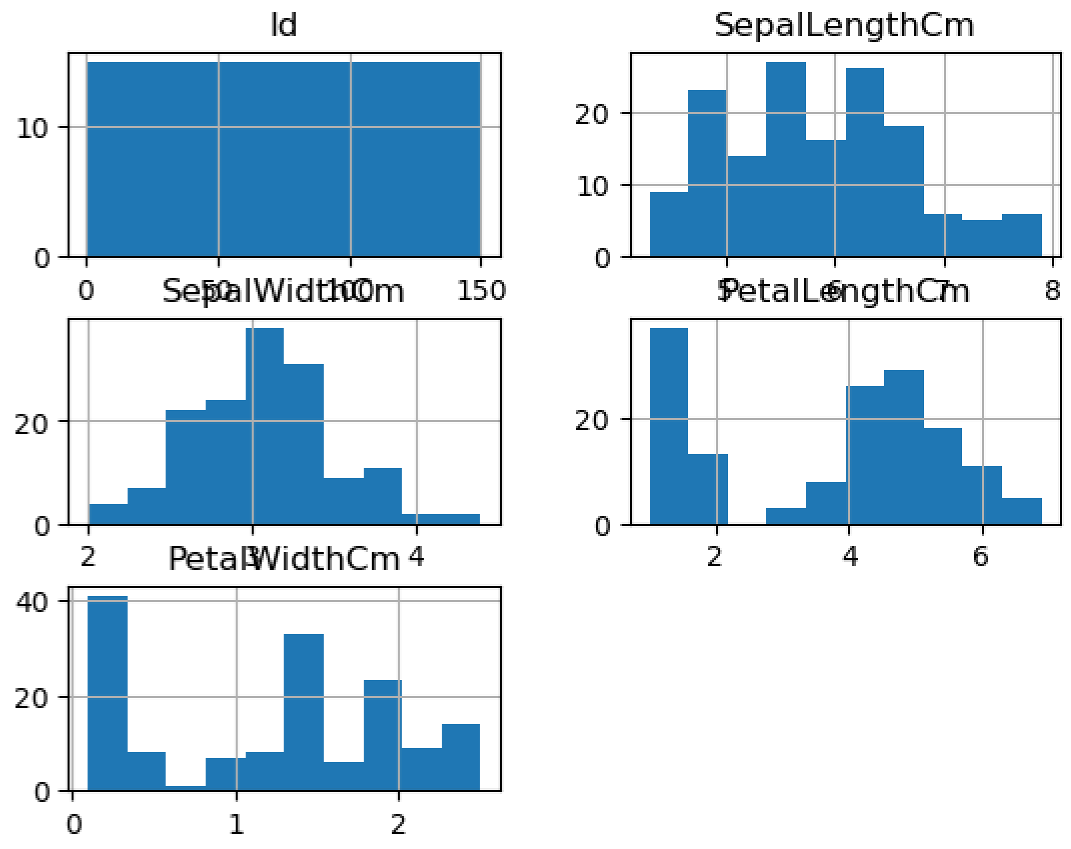
```
In [15]: import matplotlib.pyplot as plt
         plt.figure(1)
         plt.boxplot([iris['SepalLengthCm']])
         plt.figure(2)
         plt.boxplot([iris['SepalWidthCm']])
         plt.show()
```



Loading [MathJax]/extensions/Safe.js



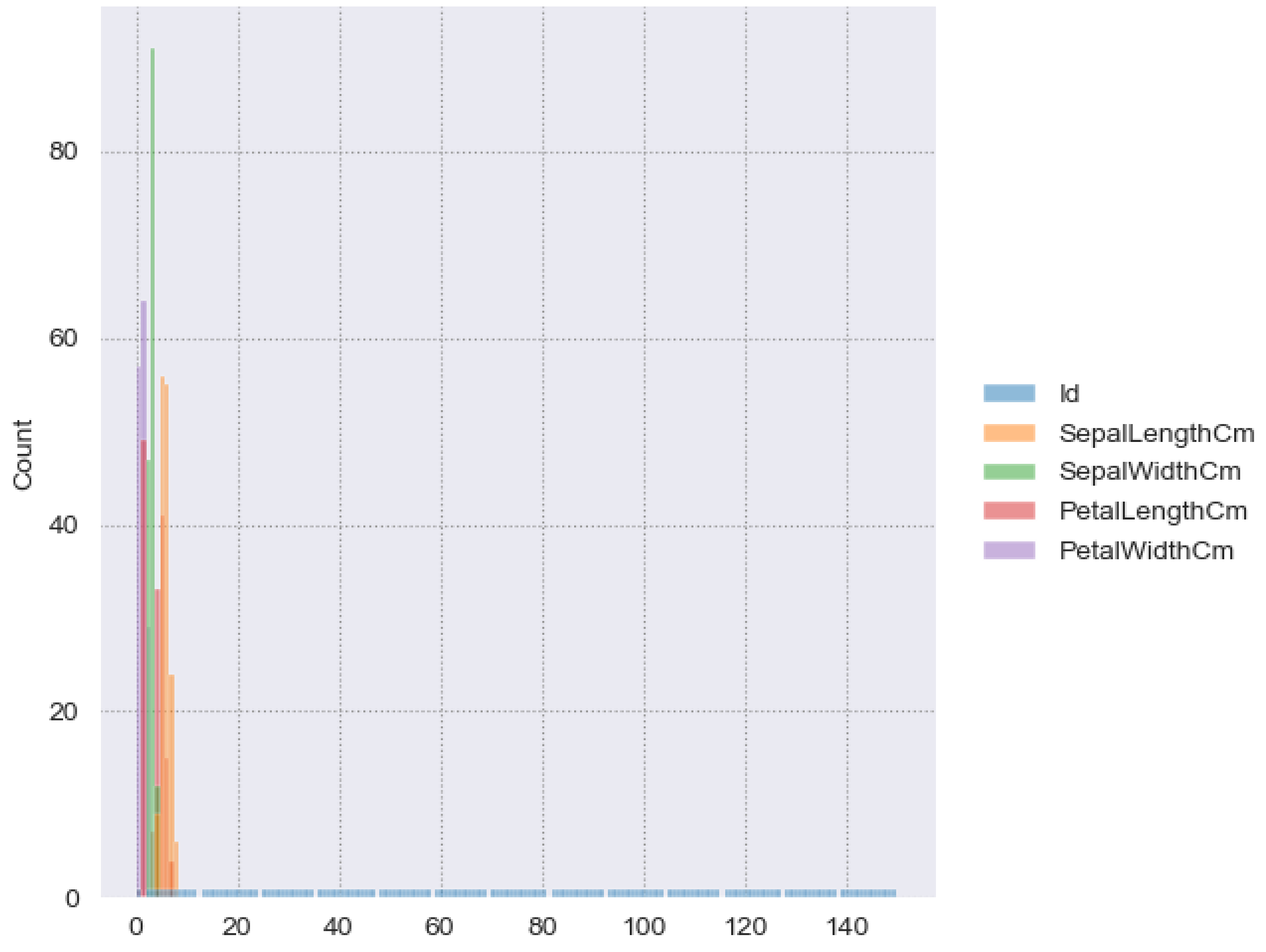
```
In [16]: iris.hist()
         plt.show()
```



```
In [20]: sns.set_style("darkgrid", {"grid.color": ".6", "grid.linestyle": ":"})
         plt.figure(figsize=(15, 10))
         sns.displot(iris)
         plt.show()
```

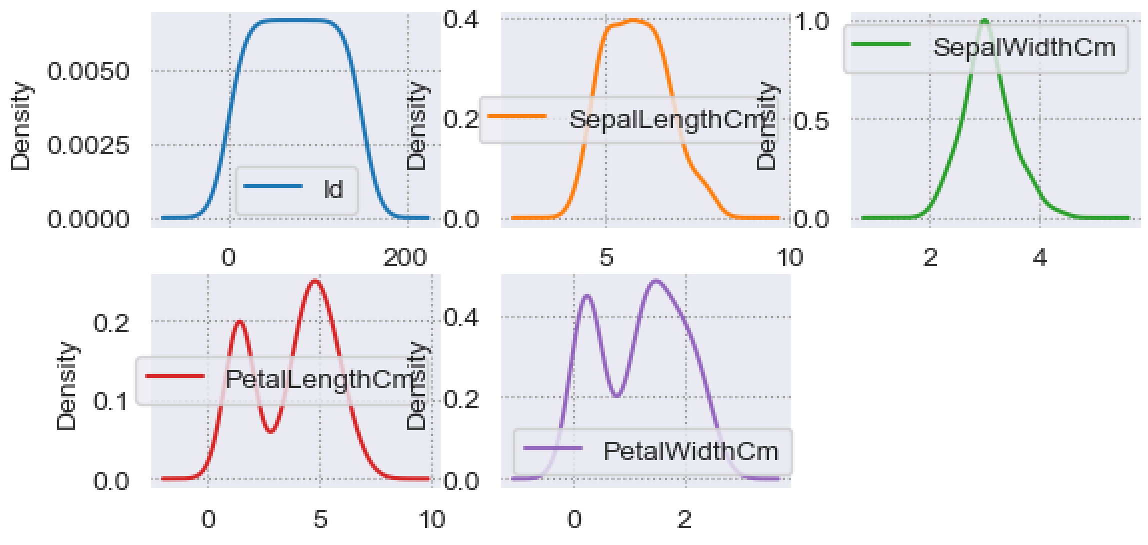
```
<Figure size 1500x1000 with 0 Axes>
```

Loading [MathJax]/extensions/Safe.js



```
In [21]: iris.plot(kind ='density',subplots = True, layout =(3,3),sharex = False)
```

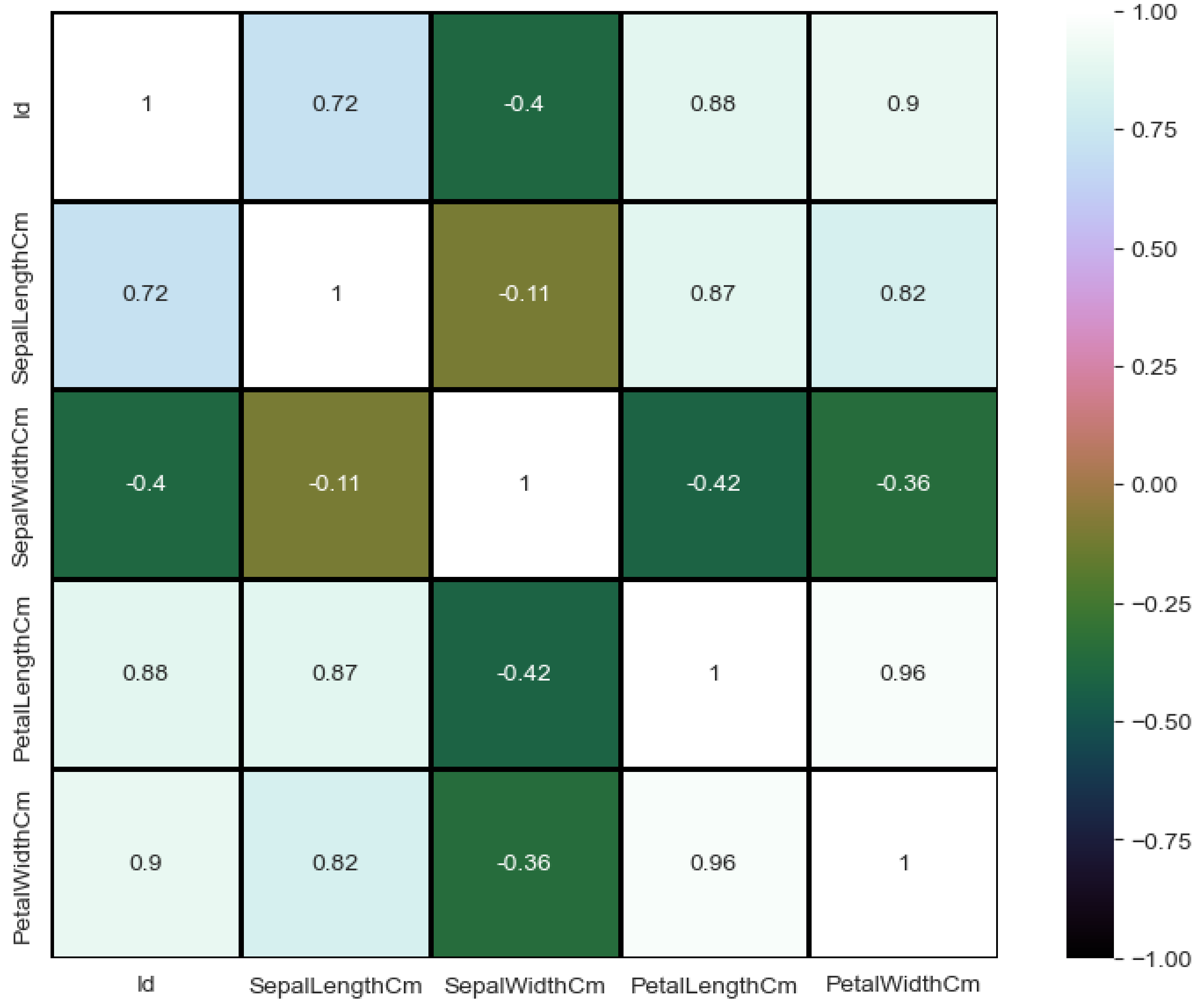
```
Out[21]: array([[<Axes: ylabel='Density'>, <Axes: ylabel='Density'>,
                 <Axes: ylabel='Density'>],
                [<Axes: ylabel='Density'>, <Axes: ylabel='Density'>,
                 <Axes: ylabel='Density'>],
                [<Axes: ylabel='Density'>, <Axes: ylabel='Density'>,
                 <Axes: ylabel='Density'>]], dtype=object)
```



```
In [24]: #Heat Maps
         fig=plt.gcf()
         fig.set_size_inches(10,7)
         fig=sns.heatmap(iris.corr(),annot=True,cmap="cubehelix",linewidths=1,linecolor='k',squar
```

Loading [MathJax]/extensions/Safe.js

```
C:\Users\User\AppData\Local\Temp\ipykernel_8128\3486906420.py:4: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version,
it will default to False. Select only valid columns or specify the value of numeric_only
to silence this warning.
```



```
In [25]: X = iris['SepalLengthCm'].values.reshape(-1,1)
         print(X)
```

Loading [MathJax]/extensions/Safe.js

```
[[5.1]
 [4.9]
 [4.7]
 [4.6]
 [5. ]
 [5.4]
 [4.6]
 [5. ]
 [4.4]
 [4.9]
 [5.4]
 [4.8]
 [4.8]
 [4.3]
 [5.8]
 [5.7]
 [5.4]
 [5.1]
 [5.7]
 [5.1]
 [5.4]
 [5.1]
 [4.6]
 [5.1]
 [4.8]
 [5. ]
 [5. ]
 [5.2]
 [5.2]
 [4.7]
 [4.8]
 [5.4]
 [5.2]
 [5.5]
 [4.9]
 [5. ]
 [5.5]
 [4.9]
 [4.4]
 [5.1]
 [5. ]
 [4.5]
 [4.4]
 [5. ]
 [5.1]
 [4.8]
 [5.1]
 [4.6]
 [5.3]
 [5. ]
 [7. ]
 [6.4]
 [6.9]
 [5.5]
 [6.5]
 [5.7]
 [6.3]
 [4.9]
 [6.6]
 [5.2]
 [5. ]
 [5.9]
 [6. ]
 [6.1]
```

Loading [MathJax]/extensions/Safe.js

```
[5.6]
[6.7]
[5.6]
[5.8]
[6.2]
[5.6]
[5.9]
[6.1]
[6.3]
[6.1]
[6.4]
[6.6]
[6.8]
[6.7]
[6. ]
[5.7]
[5.5]
[5.5]
[5.8]
[6. ]
[5.4]
[6. ]
[6.7]
[6.3]
[5.6]
[5.5]
[5.5]
[6.1]
[5.8]
[5. ]
[5.6]
[5.7]
[5.7]
[6.2]
[5.1]
[5.7]
[6.3]
[5.8]
[7.1]
[6.3]
[6.5]
[7.6]
[4.9]
[7.3]
[6.7]
[7.2]
[6.5]
[6.4]
[6.8]
[5.7]
[5.8]
[6.4]
[6.5]
[7.7]
[7.7]
[6. ]
[6.9]
[5.6]
[7.7]
[6.3]
[6.7]
[7.2]
[6.2]
[6.1]
```
Loading [MathJax]/extensions/Safe.js

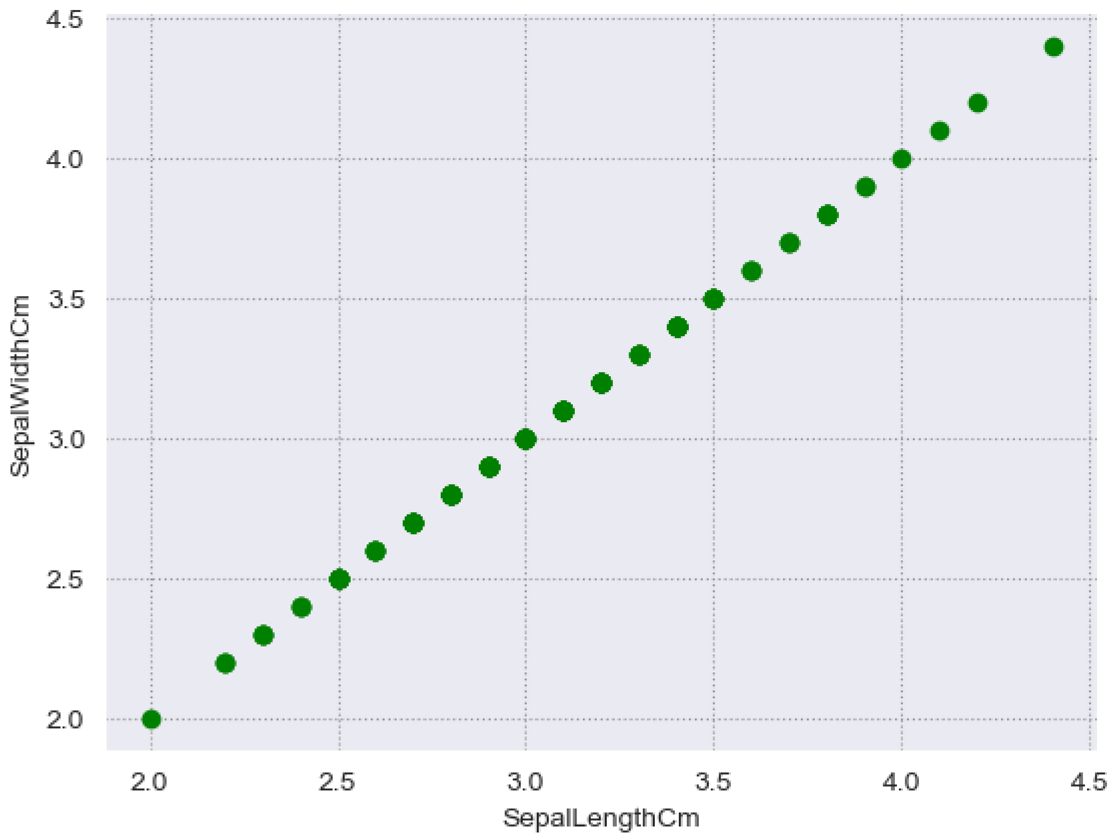
```
 [6.4]
 [7.2]
 [7.4]
 [7.9]
 [6.4]
 [6.3]
 [6.1]
 [7.7]
 [6.3]
 [6.4]
 [6. ]
 [6.9]
 [6.7]
 [6.9]
 [5.8]
 [6.8]
 [6.7]
 [6.7]
 [6.3]
 [6.5]
 [6.2]
 [5.9]]
```

In [29]:
```
Y= iris['SepalWidthCm'].values.reshape(-1,1)
print(Y)
```

```
[[3.5]
 [3. ]
 [3.2]
 [3.1]
 [3.6]
 [3.9]
 [3.4]
 [3.4]
 [2.9]
 [3.1]
 [3.7]
 [3.4]
 [3. ]
 [3. ]
 [4. ]
 [4.4]
 [3.9]
 [3.5]
 [3.8]
 [3.8]
 [3.4]
 [3.7]
 [3.6]
 [3.3]
 [3.4]
 [3. ]
 [3.4]
 [3.5]
 [3.4]
 [3.2]
 [3.1]
 [3.4]
 [4.1]
 [4.2]
 [3.1]
 [3.2]
 [3.5]
 [3.1]
 [3. ]
 [3.4]
 [3.5]
 [2.3]
 [3.2]
 [3.5]
 [3.8]
 [3. ]
 [3.8]
 [3.2]
 [3.7]
 [3.3]
 [3.2]
 [3.2]
 [3.1]
 [2.3]
 [2.8]
 [2.8]
 [3.3]
 [2.4]
 [2.9]
 [2.7]
 [2. ]
 [3. ]
 [2.2]
 [2.9]
 [2.9]
 [3.1]
 [3. ]
 [2.7]
 [2.2]
 [2.5]
 [3.2]
 [2.8]
 [2.5]
 [2.8]
 [2.9]
 [3. ]
 [2.8]
 [3. ]
 [2.9]
 [2.6]
 [2.4]
 [2.4]
 [2.7]
 [2.7]
 [3. ]
 [3.4]
 [3.1]
 [2.3]
 [3. ]
 [2.5]
 [2.6]
 [3. ]
 [2.6]
 [2.3]
 [2.7]
 [3. ]
 [2.9]
 [2.9]
 [2.5]
 [2.8]
 [3.3]
 [2.7]
 [3. ]
 [2.9]
 [3. ]
 [3. ]
 [2.5]
 [2.9]
 [2.5]
 [3.6]
 [3.2]
 [2.7]
 [3. ]
 [2.5]
 [2.8]
 [3.2]
 [3. ]
 [3.8]
 [2.6]
 [2.2]
 [3.2]
 [2.8]
 [2.8]
 [2.7]
 [3.3]
 [3.2]
 [2.8]
 [3. ]
```
Loading [MathJax]/extensions/Safe.js

```
 [2.8]
 [3. ]
 [2.8]
 [3.8]
 [2.8]
 [2.8]
 [2.6]
 [3. ]
 [3.4]
 [3.1]
 [3. ]
 [3.1]
 [3.1]
 [3.1]
 [2.7]
 [3.2]
 [3.3]
 [3. ]
 [2.5]
 [3. ]
 [3.4]
 [3. ]]
```

In [30]:
```python
plt.xlabel("SepalLengthCm")
plt.ylabel("SepalWidthCm")
plt.scatter(X,Y,color='g')
plt.show()
```



In [ ]:

In [9]:
```python
x = iris.drop("Species", axis=1)
y = iris["Species"]
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y,
                                                    test_size=0.2,
                                                    random_state=0)
```

Loading [MathJax]/extensions/Safe.js

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(x_train, y_train)
```

Out[9]:

```
▼ KNeighborsClassifier
KNeighborsClassifier(n_neighbors=1)
```

In [ ]:

In [11]:

```
x_new = np.array([[5.1, 3.9, 1, 0.2, 1]])
prediction = knn.predict(x_new)
print("Prediction: {}".format(prediction))
```

```
Prediction: ['Iris-setosa']
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning:

X does not have valid feature names, but KNeighborsClassifier was fitted with feature na
mes
```

In [ ]:

Loading [MathJax]/extensions/Safe.js