Question 2

Importing Libraries

In [39]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statistics as st
import seaborn as sns
```

Reading the csv Files

In [2]:

```python
df = pd.read_csv('Iris.csv')
```

In [3]:

```python
df
```

Out[3]:

	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

In [4]:

```python
df.drop('Id', axis = 1, inplace = True) # Removing Id column from the dataframe
```

In [5]:

```python
df
```

Out[5]:

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

150 rows × 5 columns

In [6]:

```
categories = [i for i in df['Species'].unique()]
categories
```

Out[6]:

```
['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
```

In [19]:

```
features = [feat for feat in df.columns if df[feat].dtype != 'O']
features
```

Out[19]:

```
['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']
```

In [20]:

```
df.isnull().sum()  # Checking NULL values
```

Out[20]:

```
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [21]:

```
species_group = df.groupby('Species')
```

Calculating mean, standard deviation and variance

In [22]:

```
species_group.mean()
```

Out[22]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Species				
Iris-setosa	5.006	3.418	1.464	0.244
Iris-versicolor	5.936	2.770	4.260	1.326
Iris-virginica	6.588	2.974	5.552	2.026

```
In [23]: species_group.std()
```

Out[23]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Species				
Iris-setosa	0.352490	0.381024	0.173511	0.107210
Iris-versicolor	0.516171	0.313798	0.469911	0.197753
Iris-virginica	0.635880	0.322497	0.551895	0.274650

```
In [24]: species_group.var()
```

Out[24]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
Species				
Iris-setosa	0.124249	0.145180	0.030106	0.011494
Iris-versicolor	0.266433	0.098469	0.220816	0.039106
Iris-virginica	0.404343	0.104004	0.304588	0.075433

```
In [25]: pd.options.display.max_columns = 100 # Setting max column size to 100 so the out
```

Statistical Details of the Species

```
In [26]: species_group.describe()
```

Out[26]:

	SepalLengthCm										
	count	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**	**count**	**mean**	**st**
Species											
Iris-setosa	50.0	5.006	0.352490	4.3	4.800	5.0	5.2	5.8	50.0	3.418	0.38102
Iris-versicolor	50.0	5.936	0.516171	4.9	5.600	5.9	6.3	7.0	50.0	2.770	0.31379
Iris-virginica	50.0	6.588	0.635880	4.9	6.225	6.5	6.9	7.9	50.0	2.974	0.32249

Calculation Covariance without using pandas library

In [27]:

```
def covariance(x, y):
    # Finding the mean of the series x and y
    mean_x = sum(x)/len(x)
    mean_y = sum(y)/len(y)
    # Subtracting mean from the individual elements
    sub_x = [i - mean_x for i in x]
    sub_y = [i - mean_y for i in y]
    numerator = sum([sub_x[i]*sub_y[i] for i in range(len(sub_x))])
    denominator = len(x)-1
    cov = numerator/denominator
    return cov
```

In [31]:

```
for i in [0,1,2,3]:
    for j in [0,1,2,3]:
        if (i < j and i != j):
            val = covariance(df[features[i]],df[features[j]])
            print('Covariance for {} and {} : {}'.format(features[i],features[j]
```

```
Covariance for SepalLengthCm and SepalWidthCm : -0.03926845637583892
Covariance for SepalLengthCm and PetalLengthCm : 1.2736823266219242
Covariance for SepalLengthCm and PetalWidthCm : 0.5169038031319912
Covariance for SepalWidthCm and PetalLengthCm : -0.32171275167785246
Covariance for SepalWidthCm and PetalWidthCm : -0.11798120805369122
Covariance for PetalLengthCm and PetalWidthCm : 1.2963874720357946
```

Calculating Covariance using pandas Library

In [33]:

```
for i in [0,1,2,3]:
    for j in [0,1,2,3]:
        if (i < j and i != j):
            val = df[features[i]].cov(df[features[j]])
            print('Covariance for {} and {} : {}'.format(features[i],features[j]
```

```
Covariance for SepalLengthCm and SepalWidthCm : -0.03926845637583891
Covariance for SepalLengthCm and PetalLengthCm : 1.2736823266219242
Covariance for SepalLengthCm and PetalWidthCm : 0.5169038031319911
Covariance for SepalWidthCm and PetalLengthCm : -0.32171275167785235
Covariance for SepalWidthCm and PetalWidthCm : -0.11798120805369125
Covariance for PetalLengthCm and PetalWidthCm : 1.296387472035794
```

Calculation Correlation without using pandas library

In [34]:

```
# Writing the function for Correlation Coefficient
def correlation(x, y):
    # Finding the mean of the series x and y
    mean_x = sum(x)/float(len(x))
    mean_y = sum(y)/float(len(y))
    # Subtracting mean from the individual elements
    sub_x = [i-mean_x for i in x]
    sub_y = [i-mean_y for i in y]
    # covariance for x and y
    numerator = sum([sub_x[i]*sub_y[i] for i in range(len(sub_x))])
    # Standard Deviation of x and y
    std_deviation_x = sum([sub_x[i]**2.0 for i in range(len(sub_x))])
    std_deviation_y = sum([sub_y[i]**2.0 for i in range(len(sub_y))])
    # squaring by 0.5 to find the square root
    denominator = (std_deviation_x*std_deviation_y)**0.5 # short but equivalent
    cor = numerator/denominator
    return cor
```

In [35]:

```
for i in [0,1,2,3]:
    for j in [0,1,2,3]:
        if (i < j and i != j):
            val = correlation(df[features[i]],df[features[j]])
            print('Correlation coefficient for {} and {} : {}'.format(features[i
```

```
Correlation coefficient for SepalLengthCm and SepalWidthCm : -0.10936924995064935
Correlation coefficient for SepalLengthCm and PetalLengthCm : 0.8717541573048719
Correlation coefficient for SepalLengthCm and PetalWidthCm : 0.8179536333691635
Correlation coefficient for SepalWidthCm and PetalLengthCm : -0.42051609640115484
Correlation coefficient for SepalWidthCm and PetalWidthCm : -0.3565440896138055
Correlation coefficient for PetalLengthCm and PetalWidthCm : 0.9627570970509667
```

Calculation Correlation using pandas library

In [38]:

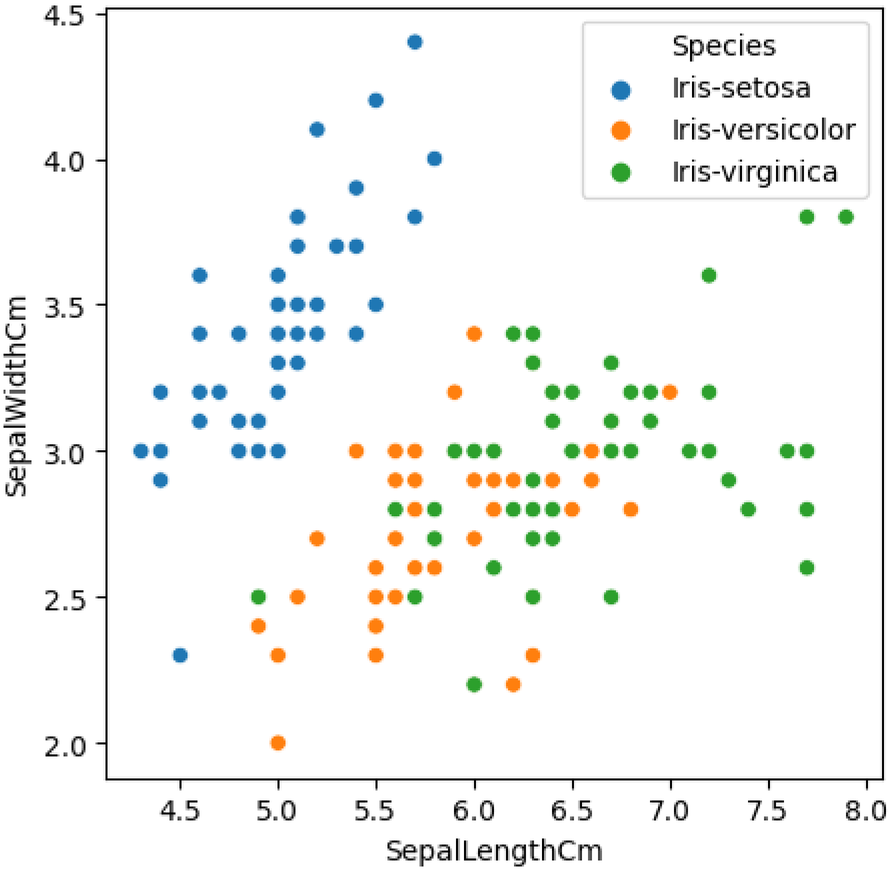
```
for i in [0,1,2,3]:
    for j in [0,1,2,3]:
        if (i < j and i != j):
            val = df[features[i]].corr(df[features[j]])
            print('Correlation coefficient for {} and {} : {}'.format(features[i
```

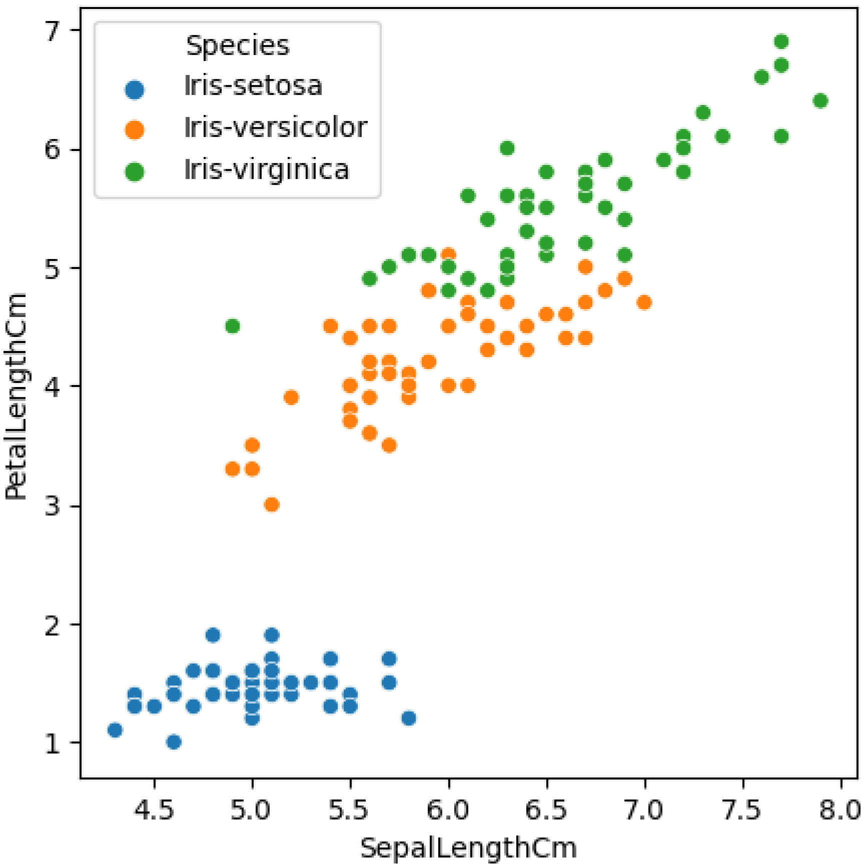
```
Correlation coefficient for SepalLengthCm and SepalWidthCm : -0.1093692499506493
Correlation coefficient for SepalLengthCm and PetalLengthCm : 0.8717541573048712
Correlation coefficient for SepalLengthCm and PetalWidthCm : 0.8179536333691636
Correlation coefficient for SepalWidthCm and PetalLengthCm : -0.42051609640115445
Correlation coefficient for SepalWidthCm and PetalWidthCm : -0.35654408961380574
Correlation coefficient for PetalLengthCm and PetalWidthCm : 0.9627570970509659
```

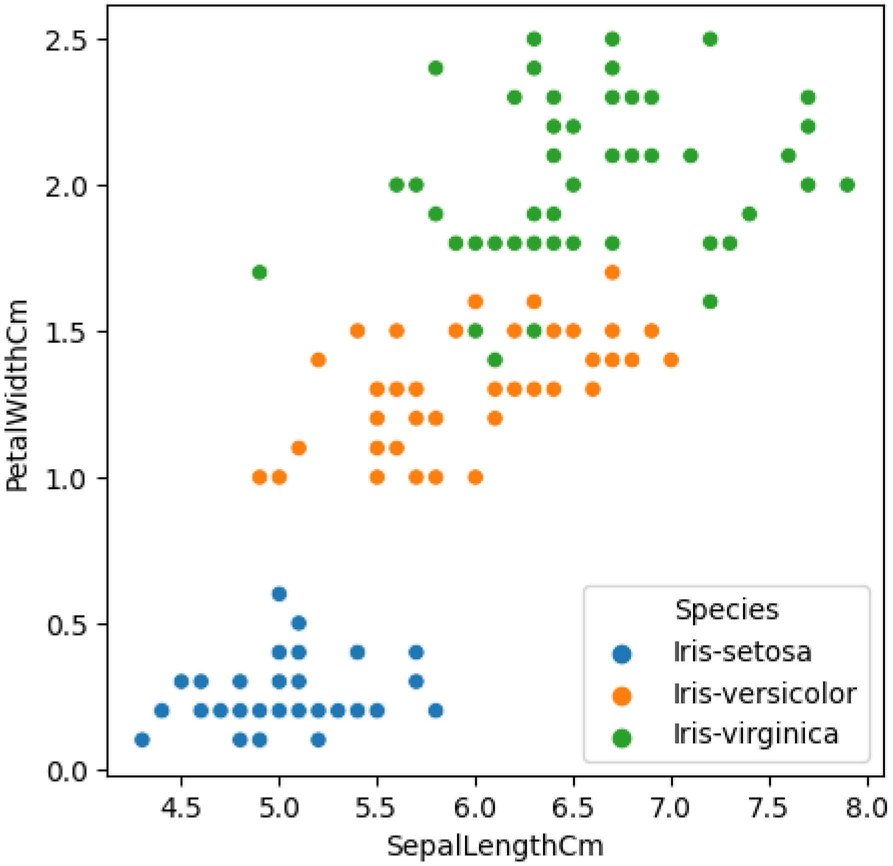
Visualizing the correlation using Seaborn Library

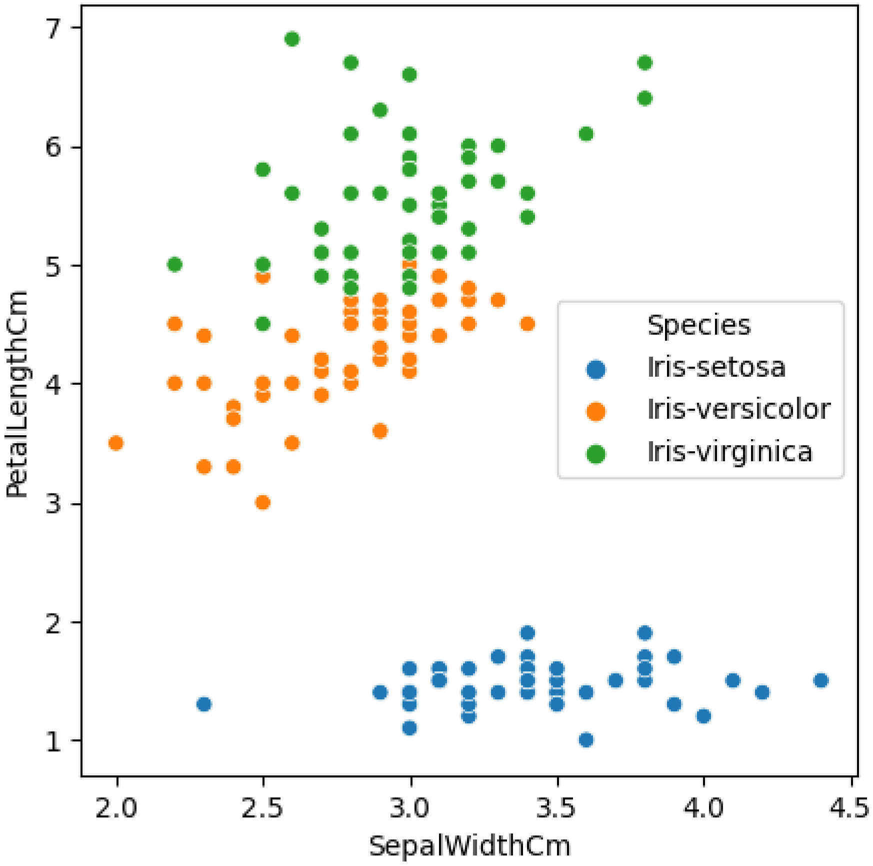
In [42]:

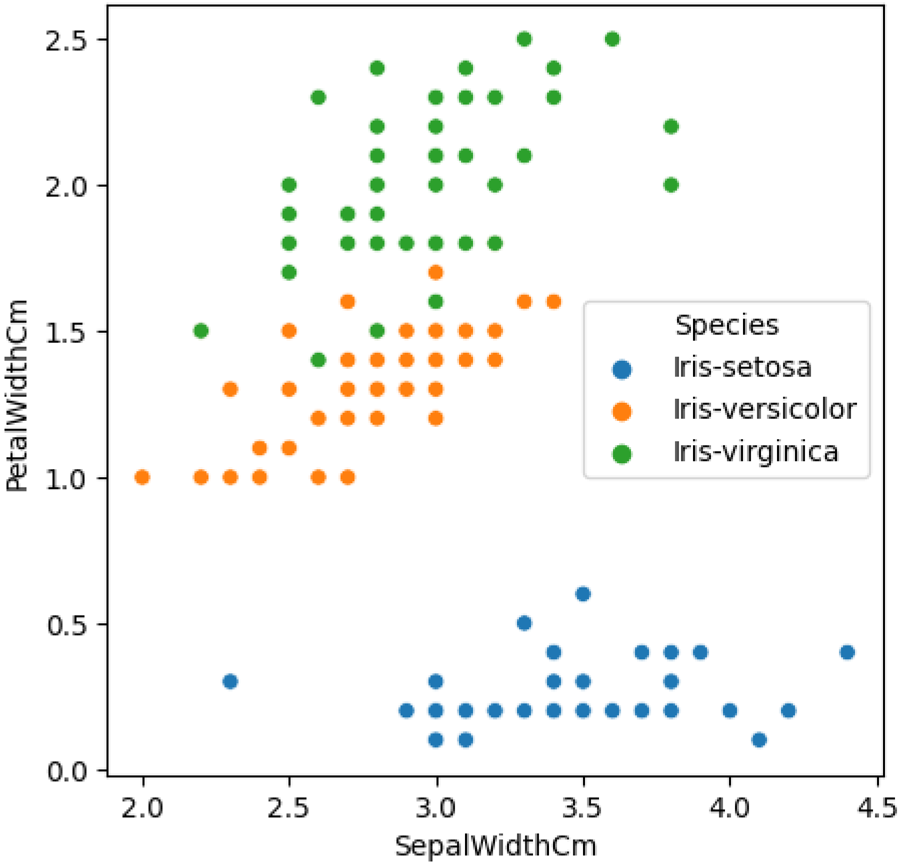
```
for i in [0,1,2,3]:
    for j in [0,1,2,3]:
        if(i<j and i != j ):
            fig = plt.figure()
            fig.set_figheight(5)
            fig.set_figwidth(5)
            ax = sns.scatterplot(x=features[i], y=features[j],data=df, hue='Spec
```

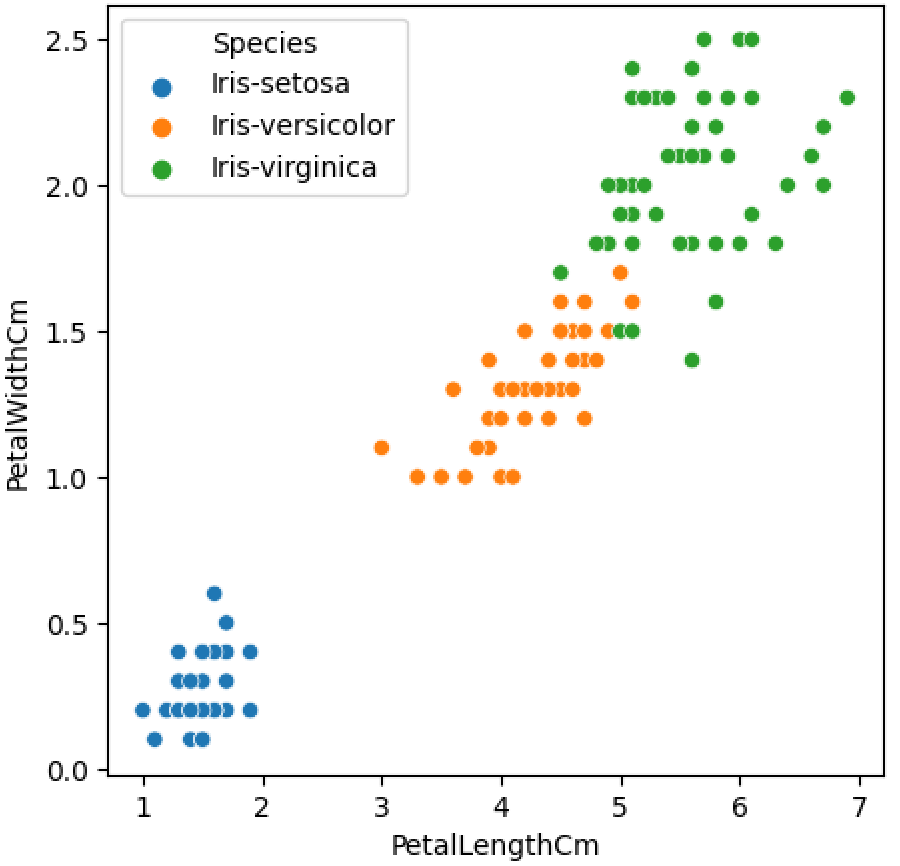

Species
Iris-setosa
Iris-versicolor
Iris-virginica
SepalWidthCm
4.5
4.0
3.5
3.0
2.5
2.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
SepalLengthCm


Species
Iris-setosa
Iris-versicolor
Iris-virginica
PetalLengthCm
7
6
5
4
3
2
1
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
SepalLengthCm


PetalWidthCm
0.0
0.5
1.0
1.5
2.0
2.5
SepalLengthCm
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Species
Iris-setosa
Iris-versicolor
Iris-virginica


PetalLengthCm
1
2
3
4
5
6
7
SepalWidthCm
2.0
2.5
3.0
3.5
4.0
4.5
Species
Iris-setosa
Iris-versicolor
Iris-virginica





Correlation Matrix

In [43]:
```
cormatrix = df.corr(numeric_only=True)
round(cormatrix,4)
```

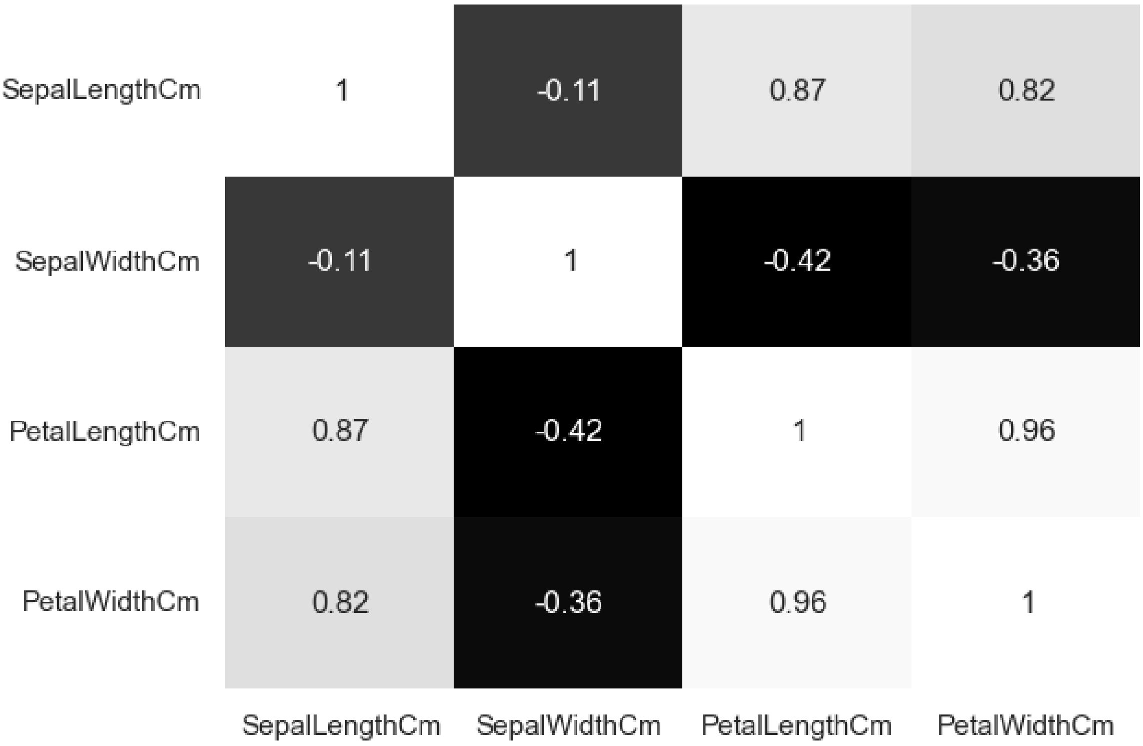
Out[43]:

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
SepalLengthCm	1.0000	-0.1094	0.8718	0.8180
SepalWidthCm	-0.1094	1.0000	-0.4205	-0.3565
PetalLengthCm	0.8718	-0.4205	1.0000	0.9628
PetalWidthCm	0.8180	-0.3565	0.9628	1.0000

Visualizing Correlation matrix using HeatMap

In [54]:

```
sns.set(style="whitegrid")
sns.heatmap(cormatrix, cmap="gray", annot=True, cbar=False,linecolor='blue')
plt.show()
```



In [ ]: