In [1]:
```python
# import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
# read_csv() # use to load or import csv file
data=pd.read_csv("C:\\Users\\Niranjani N\\Desktop\iris.csv")
print(data)
```

```
      Id  SepalLengthCm  SepalWidthCm  PetalLengthCm  PetalWidthCm  \
0      1            5.1           3.5            1.4           0.2
1      2            4.9           3.0            1.4           0.2
2      3            4.7           3.2            1.3           0.2
3      4            4.6           3.1            1.5           0.2
4      5            5.0           3.6            1.4           0.2
..   ...            ...           ...            ...           ...
145  146            6.7           3.0            5.2           2.3
146  147            6.3           2.5            5.0           1.9
147  148            6.5           3.0            5.2           2.0
148  149            6.2           3.4            5.4           2.3
149  150            5.9           3.0            5.1           1.8

            Species
0       Iris-setosa
1       Iris-setosa
2       Iris-setosa
3       Iris-setosa
4       Iris-setosa
..              ...
145  Iris-virginica
146  Iris-virginica
147  Iris-virginica
148  Iris-virginica
149  Iris-virginica

[150 rows x 6 columns]
```

In [3]:
```python
data.head() # head is used to show the first five values of the dataset
```

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

In [4]:
```python
data.tail() # tail used to show the last five values of the dataset
```

Out[4]:

	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

In [5]:
```python
data.shape # it is used to find the no.of rows and columns
```

Out[5]: (150, 6)

In [6]:
```python
data.info() # total information about dataset
```

```
<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 [7]:
```python
data.isna().sum() # checking the null values
```

Out[7]:
```
Id               0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

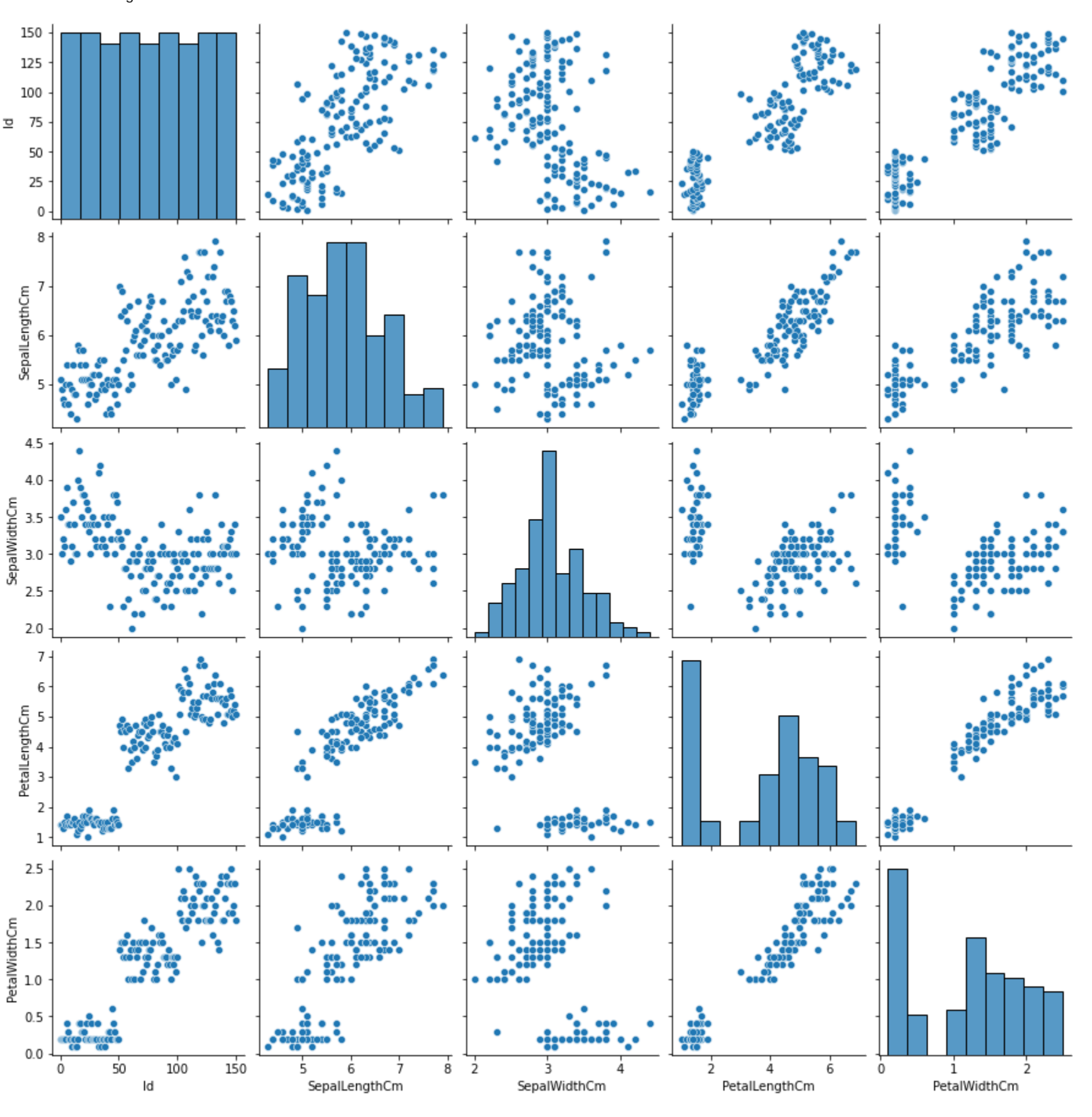
In [8]:
```python
data.describe() # description about the dataset
```

Out[8]:

	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

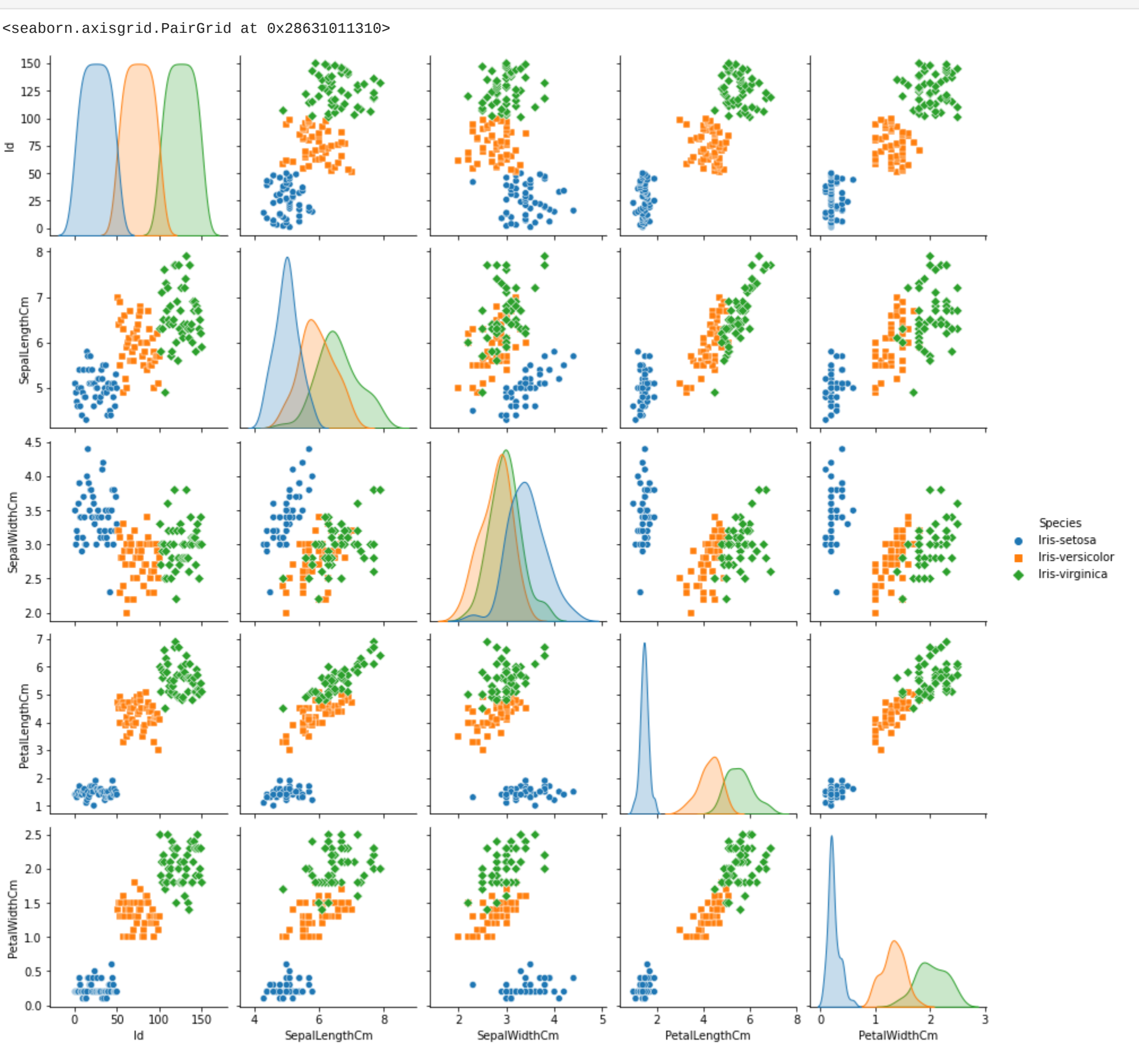
In [9]:
```python
sns.pairplot(data)
```

Out[9]: <seaborn.axisgrid.PairGrid at 0x286302076d0>



In [10]:
```python
sns.pairplot(data,hue='Species',markers=['o','s','D'])
```

Out[10]: <seaborn.axisgrid.PairGrid at 0x28631011310>

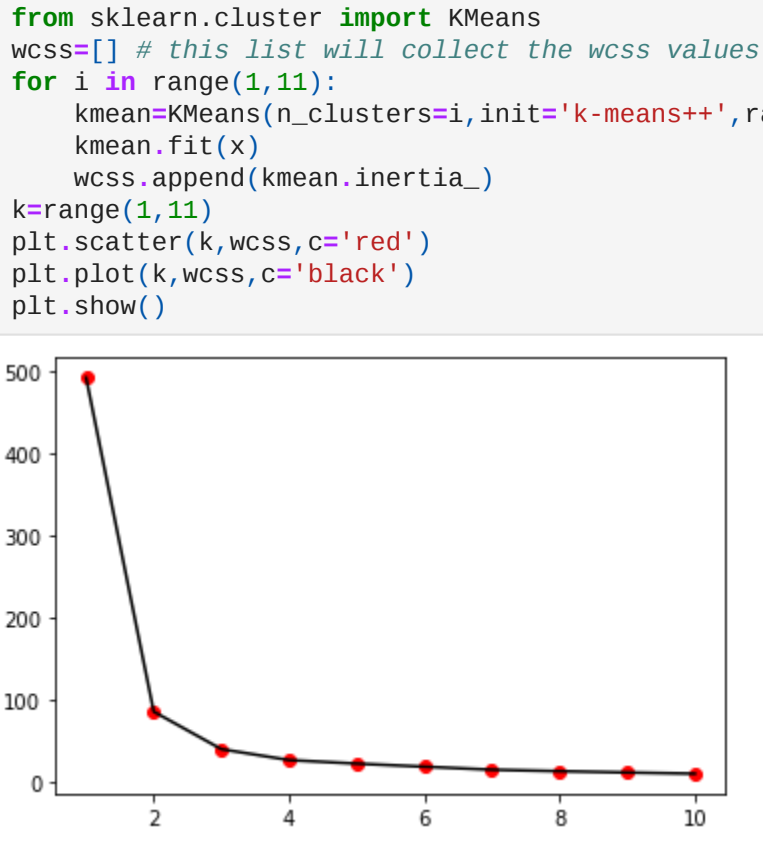


In [11]:
```python
x=data.iloc[:,2:4].values
```

In [12]:
```python
x
```

Out[12]:
```
array([[3.5, 1.4],
       [3. , 1.4],
       [3.2, 1.3],
       [3.1, 1.5],
       [3.6, 1.4],
       [3.9, 1.7],
       [3.4, 1.4],
       [3.4, 1.5],
       [2.9, 1.4],
       [3.1, 1.5],
       [3.7, 1.5],
       [3.4, 1.6],
       [3. , 1.4],
       [3. , 1.1],
       [4. , 1.2],
       [4.4, 1.5],
       [3.9, 1.3],
       [3.5, 1.4],
       [3.8, 1.7],
       [3.8, 1.5],
       [3.4, 1.7],
       [3.7, 1.5],
       [3.6, 1. ],
       [3.3, 1.7],
       [3.4, 1.9],
       [3. , 1.6],
       [3.4, 1.6],
       [3.5, 1.5],
       [3.4, 1.4],
       [3.2, 1.6],
       [3.1, 1.6],
       [3.4, 1.5],
       [4.1, 1.5],
       [4.2, 1.4],
       [3.1, 1.5],
       [3.2, 1.2],
       [3.5, 1.3],
       [3.1, 1.5],
       [3. , 1.3],
       [3.4, 1.5],
       [3.5, 1.3],
       [2.3, 1.3],
       [3.2, 1.3],
       [3.5, 1.6],
       [3.8, 1.9],
       [3. , 1.4],
       [3.8, 1.6],
       [3.2, 1.4],
       [3.7, 1.5],
       [3.3, 1.4],
       [3.2, 4.7],
       [3.2, 4.5],
       [3.1, 4.9],
       [2.3, 4. ],
       [2.8, 4.6],
       [2.8, 4.5],
       [3.3, 4.7],
       [2.4, 3.3],
       [2.9, 4.6],
       [2.7, 3.9],
       [2. , 3.5],
       [3. , 4.2],
       [2.2, 4. ],
       [2.9, 4.7],
       [2.9, 3.6],
       [3.1, 4.4],
       [3. , 4.5],
       [2.7, 4.1],
       [2.2, 4.5],
       [2.5, 3.9],
       [3.2, 4.8],
       [2.8, 4. ],
       [2.5, 4.9],
       [2.8, 4.7],
       [2.9, 4.3],
       [3. , 4.4],
       [2.8, 4.8],
       [3. , 5. ],
       [2.9, 4.5],
       [2.6, 3.5],
       [2.4, 3.8],
       [2.4, 3.7],
       [2.7, 3.9],
       [2.7, 5.1],
       [3. , 4.5],
       [3.4, 4.5],
       [3.1, 4.7],
       [2.3, 4.4],
       [3. , 4.1],
       [2.5, 4. ],
       [2.6, 4.4],
       [3. , 4.6],
       [2.6, 4. ],
       [2.3, 3.3],
       [2.7, 4.2],
       [3. , 4.2],
       [2.9, 4.2],
       [2.9, 4.3],
       [2.5, 3. ],
       [2.8, 4.1],
       [3.3, 6. ],
       [2.7, 5.1],
       [3. , 5.9],
       [2.9, 5.6],
       [3. , 5.8],
       [3. , 6.6],
       [2.5, 4.5],
       [2.9, 6.3],
       [2.5, 5.8],
       [3.6, 6.1],
       [3.2, 5.1],
       [2.7, 5.3],
       [3. , 5.5],
       [2.5, 5. ],
       [2.8, 5.1],
       [3.2, 5.3],
       [3. , 5.5],
       [3.8, 6.7],
       [2.6, 6.9],
       [2.2, 5. ],
       [3.2, 5.7],
       [2.8, 4.9],
       [2.8, 6.7],
       [2.7, 4.9],
       [3.3, 5.7],
       [3.2, 6. ],
       [2.8, 4.8],
       [3. , 4.9],
       [2.8, 5.6],
       [3. , 5.8],
       [2.8, 6.1],
       [3.8, 6.4],
       [2.8, 5.6],
       [2.8, 5.1],
       [2.6, 5.6],
       [3. , 6.1],
       [3.4, 5.6],
       [3.1, 5.5],
       [3. , 4.8],
       [3.1, 5.4],
       [3.1, 5.6],
       [3.1, 5.1],
       [2.7, 5.1],
       [3.2, 5.9],
       [3.3, 5.7],
       [3. , 5.2],
       [2.5, 5. ],
       [3. , 5.2],
       [3.4, 5.4],
       [3. , 5.1]])
```

In [13]:
```python
from sklearn.cluster import KMeans
wcss=[] # this list will collect the wcss values for the different clusters
for i in range(1,11):
    kmean=KMeans(n_clusters=i,init='k-means++',random_state=42)
    kmean.fit(x)
    wcss.append(kmean.inertia_)
k=range(1,11)
plt.scatter(k,wcss,c='red')
plt.plot(k,wcss,c='black')
plt.show()
```



In [14]:
```python
kmean=KMeans(n_clusters=4,init='k-means++',random_state=42)
y_pred=kmean.fit_predict(x)
```

In [15]:
```python
y_pred
```

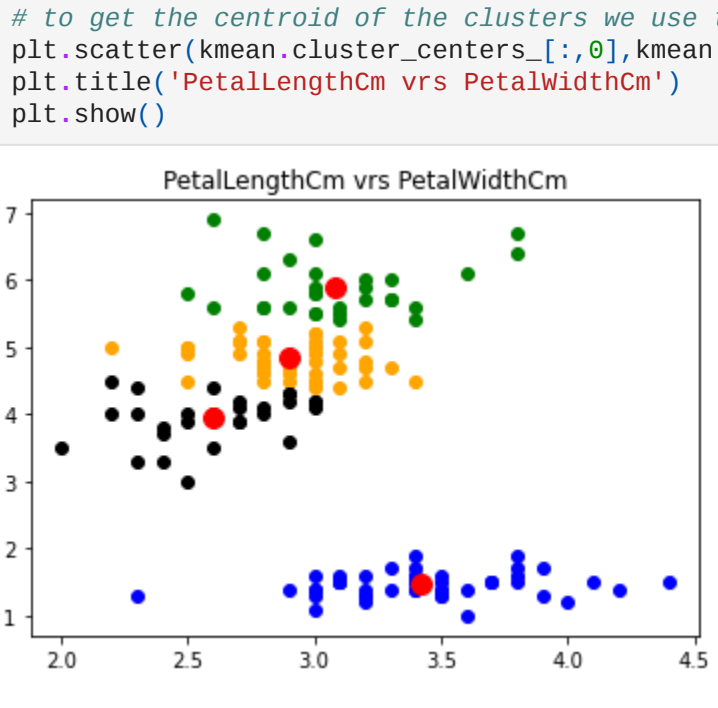
Out[15]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 0, 0, 0, 3, 0, 0, 0, 3, 0, 3, 3, 3, 3, 0, 3, 0,
       0, 3, 3, 3, 0, 3, 0, 0, 3, 0, 0, 0, 0, 3, 3, 3, 3, 0, 0, 0, 0, 3,
       3, 3, 3, 0, 3, 3, 3, 3, 3, 3, 3, 3, 2, 0, 2, 2, 2, 2, 0, 2, 2, 2,
       0, 0, 2, 0, 0, 0, 2, 2, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 2, 2, 2, 2,
       2, 0, 2, 2, 2, 2, 0, 2, 2, 0, 0, 2, 2, 0, 0, 0, 2, 0])
```

In [16]:
```python
# To visualize the clusters we are going to use the y_pred values row index
# as a reference to extract the corresponding rows from x
text=x[:,0]
text=text.reshape(150,1)
```

In [17]:
```python
compare=np.append(arr=y_pred.reshape(150,1),values=text,axis=1)
print(compare)
```

```
[[1.  3.5]
 [1.  3. ]
 [1.  3.2]
 [1.  3.1]
 [1.  3.6]
 [1.  3.9]
 [1.  3.4]
 [1.  3.4]
 [1.  2.9]
 [1.  3.1]
 [1.  3.7]
 [1.  3.4]
 [1.  3. ]
 [1.  3. ]
 [1.  4. ]
 [1.  4.4]
 [1.  3.9]
 [1.  3.5]
 [1.  3.8]
 [1.  3.8]
 [1.  3.4]
 [1.  3.7]
 [1.  3.6]
 [1.  3.3]
 [1.  3.4]
 [1.  3. ]
 [1.  3.4]
 [1.  3.5]
 [1.  3.4]
 [1.  3.2]
 [1.  3.1]
 [1.  3.4]
 [1.  4.1]
 [1.  4.2]
 [1.  3.1]
 [1.  3.2]
 [1.  3.5]
 [1.  3.1]
 [1.  3. ]
 [1.  3.4]
 [1.  3.5]
 [1.  2.3]
 [1.  3.2]
 [1.  3.5]
 [1.  3.8]
 [1.  3. ]
 [1.  3.8]
 [1.  3.2]
 [1.  3.7]
 [1.  3.3]
 [0.  3.2]
 [0.  3.2]
 [0.  3.1]
 [3.  2.3]
 [0.  2.8]
 [0.  2.8]
 [0.  3.3]
 [3.  2.4]
 [0.  2.9]
 [3.  2.7]
 [3.  2. ]
 [3.  3. ]
 [3.  2.2]
 [0.  2.9]
 [3.  2.9]
 [0.  3.1]
 [0.  3. ]
 [3.  2.7]
 [3.  2.2]
 [3.  2.5]
 [0.  3.2]
 [3.  2.8]
 [0.  2.5]
 [0.  2.8]
 [3.  2.9]
 [0.  3. ]
 [0.  2.8]
 [0.  3. ]
 [0.  2.9]
 [3.  2.6]
 [3.  2.4]
 [3.  2.4]
 [3.  2.7]
 [0.  2.7]
 [0.  3. ]
 [0.  3.4]
 [0.  3.1]
 [3.  2.3]
 [3.  3. ]
 [3.  2.5]
 [3.  2.6]
 [0.  3. ]
 [3.  2.6]
 [3.  2.3]
 [3.  2.7]
 [3.  3. ]
 [3.  2.9]
 [3.  2.9]
 [3.  2.5]
 [3.  2.8]
 [2.  3.3]
 [0.  2.7]
 [2.  3. ]
 [2.  2.9]
 [2.  3. ]
 [2.  3. ]
 [0.  2.5]
 [2.  2.9]
 [2.  2.5]
 [2.  3.6]
 [0.  3.2]
 [0.  2.7]
 [2.  3. ]
 [0.  2.5]
 [0.  2.8]
 [0.  3.2]
 [2.  3. ]
 [2.  3.8]
 [2.  2.6]
 [0.  2.2]
 [2.  3.2]
 [0.  2.8]
 [2.  2.8]
 [0.  2.7]
 [2.  3.3]
 [2.  3.2]
 [0.  2.8]
 [0.  3. ]
 [2.  2.8]
 [2.  3. ]
 [2.  2.8]
 [2.  3.8]
 [2.  2.8]
 [0.  2.8]
 [2.  2.6]
 [2.  3. ]
 [2.  3.4]
 [2.  3.1]
 [0.  3. ]
 [2.  3.1]
 [2.  3.1]
 [0.  3.1]
 [0.  2.7]
 [2.  3.2]
 [2.  3.3]
 [0.  3. ]
 [0.  2.5]
 [0.  3. ]
 [2.  3.4]
 [0.  3. ]]
```

In [18]:
```python
plt.scatter(x[y_pred == 0,0],x[y_pred == 0,1],c='orange',label = 'cluster1')
plt.scatter(x[y_pred == 1,0],x[y_pred == 1,1],c='blue',label = 'cluster2')
plt.scatter(x[y_pred == 2,0],x[y_pred == 2,1],c='green',label = 'cluster3')
plt.scatter(x[y_pred == 3,0],x[y_pred == 3,1],c='black',label = 'cluster4')
# to get the centroid of the clusters we use the inbuilt clusters_centers function
plt.scatter(kmean.cluster_centers_[:,0],kmean.cluster_centers_[:,1],s=100,c='red')
plt.title('PetalLengthCm vrs PetalWidthCm')
plt.show()
```



In [ ]: