In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

At first i have loaded the required libraries.

In [2]:

```
file = "../input/iris/Iris.csv"
data = pd.read_csv(file)
```

Using the pandas library, i have read the dataframe.

In [3]:

```
data.head()
```

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]:

```
data.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]:

```
print(data.shape)
missing_val = data.isnull().sum()
print(missing_val[missing_val>0])
```

```
(150, 6)
Series([], dtype: int64)
```

Here we can see that there are no missing values in the data. So the data need not be cleaned.

In [6]:

```
data.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
```

We can see that all the values are non-null and datatype of each feature.

In [7]:

```
data.nunique()
```

Out[7]:

```
Id               150
SepalLengthCm     35
SepalWidthCm      23
PetalLengthCm     43
PetalWidthCm      22
Species            3
dtype: int64
```

In the data there are 3 unique species.

In [8]:

```
data.Species.unique()
```

Out[8]:

```
array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

Those three unique Iris species are setosa, versicolor and virginica.

In [9]:

```
dfs = data.loc[data["Species"]=="Iris-setosa"]
dfve = data.loc[data["Species"]=="Iris-versicolor"]
dfvi = data.loc[data["Species"]=="Iris-virginica"]
sp=[dfs,dfve,dfvi]
```

Here we have allocated seperate dataframes for each unique species. So that we can analyze each of them easily.

DATA VISUALISATION

Data Distribution

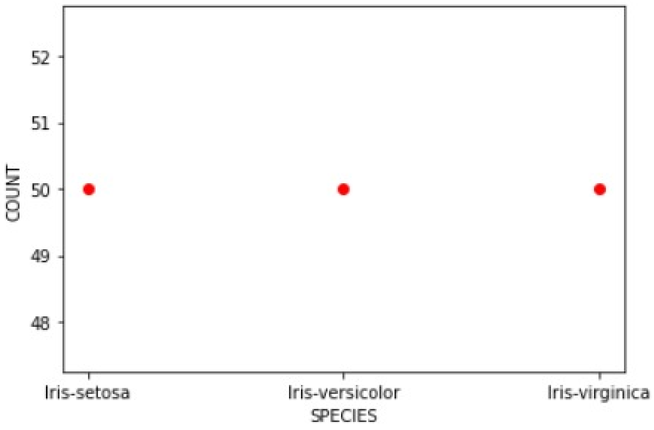
First let us see how the data is distributed among the three species.

In [10]:

```
x=[i.Species.unique()[0] for i in sp]
y=[i.shape[0] for i in sp]
plt.plot(x,y,"ro")
plt.xlabel("SPECIES")
plt.ylabel("COUNT")
```

Out[10]:

```
Text(0, 0.5, 'COUNT')
```



As we can see in the graph the data is distributed equally among the three species i.e each species have 50 rows of data.

Mean values of the features i.e Sepal length and width, Petal length and width.

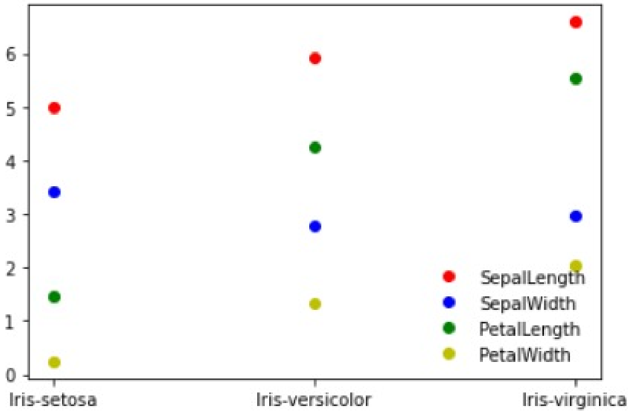
Mean values of the features i.e Sepal length and width, Petal length and width.

In [11]:

```
x=[i.Species.unique()[0] for i in sp]
y1=[i.SepalLengthCm.mean() for i in sp]
y2=[i.SepalWidthCm.mean() for i in sp]
y3=[i.PetalLengthCm.mean() for i in sp]
y4=[i.PetalWidthCm.mean() for i in sp]
fig,ax=plt.subplots()
ax.plot(x,y1,"ro",label="SepalLength")
ax.plot(x,y2,"bo",label="SepalWidth")
ax.plot(x,y3,"go",label="PetalLength")
ax.plot(x,y4,"yo",label="PetalWidth")
ax.legend(loc="lower right", frameon = False)
```

Out[11]:

```
<matplotlib.legend.Legend at 0x7f246444ced0>
```

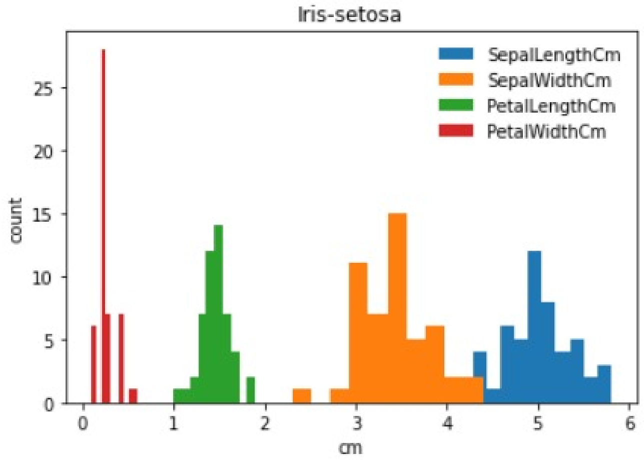


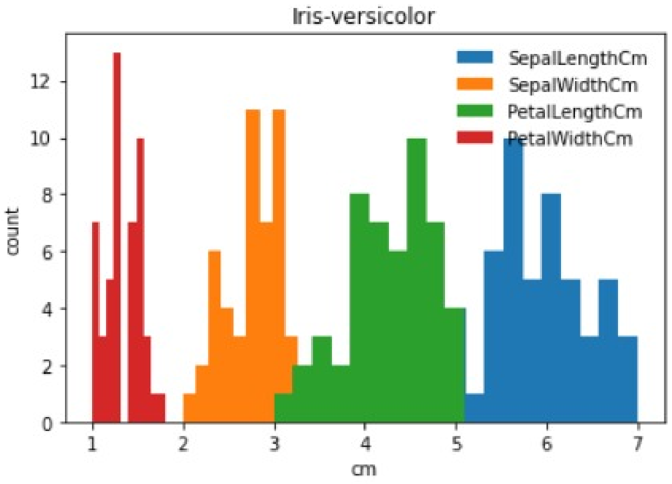
From the above graph we can see that average sepal length, petal length and petal width of virginica is greater than versicolor which is greater than setosa. Where as the average Sepal width of setosa is greater than that of versicolor and virginica.

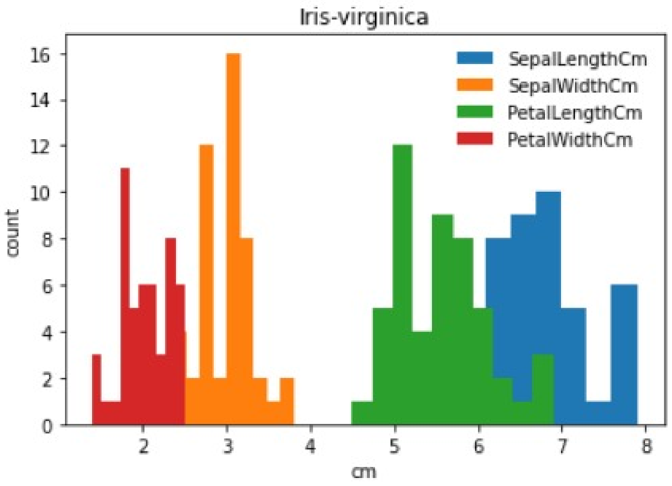
Now i have plotted histograms of sepal length and width, petal length and width of each species

In [12]:

```
feat = ['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']
for i in sp:
    fig,ax=plt.subplots()
    plt.title(i.Species.unique()[0])
    plt.ylabel("count")
    plt.xlabel("cm")
    for j in range(4):
        ax.hist(i[feat[j]],label=feat[j])
        ax.legend(frameon = False)
```




Iris-versicolor
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
count
0
2
4
6
8
10
12
1
2
3
4
5
6
7
cm


Iris-virginica
SepalLengthCm
SepalWidthCm
PetalLengthCm
PetalWidthCm
count
0
2
4
6
8
10
12
14
16
2
3
4
5
6
7
8
cm