In [1]:
```python
import pandas as pd
from sklearn.datasets import load_iris
iris=load_iris()
```

In [2]:
```python
iris.feature_names
```

Out[2]:
```
['sepal length (cm)',
 'sepal width (cm)',
 'petal length (cm)',
 'petal width (cm)']
```

In [3]:
```python
iris.target_names
```

Out[3]:
```
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

In [4]:
```python
df=pd.DataFrame(iris.data,columns=iris.feature_names)
```

In [5]:
```python
df
```

Out[5]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...	...	...	...	...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

In [6]:
```python
df['target']=iris.target
```

In [7]:
```python
df.head()
```

Out[7]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0

In [8]:
```python
df['flowers_name']=df.target.apply(lambda x:iris.target_names[x])
```

In [9]:
```python
df[45:56]
```

Out[9]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	flowers_name
45	4.8	3.0	1.4	0.3	0	setosa
46	5.1	3.8	1.6	0.2	0	setosa
47	4.6	3.2	1.4	0.2	0	setosa
48	5.3	3.7	1.5	0.2	0	setosa
49	5.0	3.3	1.4	0.2	0	setosa
50	7.0	3.2	4.7	1.4	1	versicolor
51	6.4	3.2	4.5	1.5	1	versicolor
52	6.9	3.1	4.9	1.5	1	versicolor
53	5.5	2.3	4.0	1.3	1	versicolor
54	6.5	2.8	4.6	1.5	1	versicolor
55	5.7	2.8	4.5	1.3	1	versicolor

In [10]:
```python
df0=df[:50]
df1=df[50:100]
df2=df[100:150]
```

In [11]:
```python
import matplotlib.pyplot as plt
%matplotlib inline
```
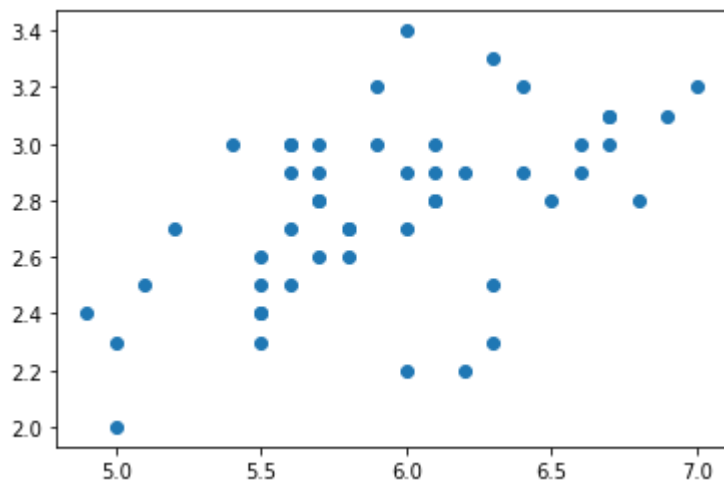
In [12]:
```python
plt.scatter(df0['sepal length (cm)'],df0['sepal width (cm)'],marker='+',color='r'
plt.scatter(df1['sepal length (cm)'],df1['sepal width (cm)'])
```

Out[12]: `<matplotlib.collections.PathCollection at 0x1033d4a3b20>`
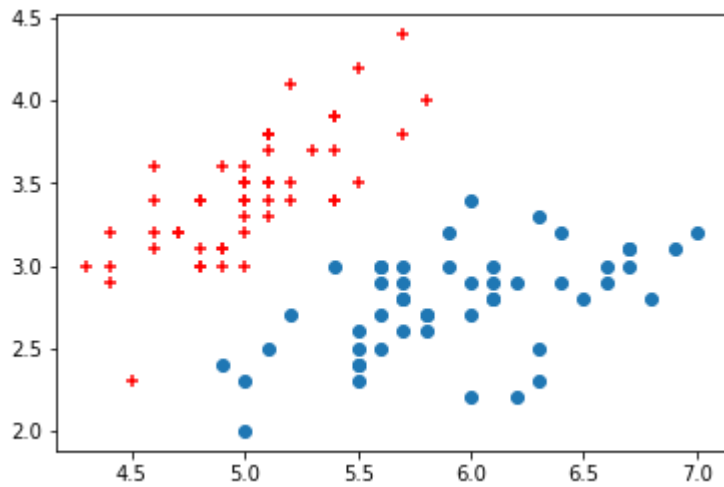
In [13]:
```python
plt.scatter(df1['sepal length (cm)'],df1['sepal width (cm)'])
```

Out[13]: `<matplotlib.collections.PathCollection at 0x10340561d00>`

In [14]:
```python
from sklearn.model_selection import train_test_split
```

In [15]:
```python
x=df.drop(['target','flowers_name'],axis=1)
```

In [16]:
```python
y=df.target
```

In [17]:
```python
x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.2)
```

In [18]:
```python
x_train.head()
```

Out[18]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
61	5.9	3.0	4.2	1.5
138	6.0	3.0	4.8	1.8
3	4.6	3.1	1.5	0.2
126	6.2	2.8	4.8	1.8
135	7.7	3.0	6.1	2.3

In [19]:
```python
from sklearn.svm import SVC
```

In [20]:
```python
model=SVC()
```

In [21]:
```python
model.fit(x_train,y_train)
```

Out[21]: SVC()

In [22]:
```python
model.predict(x_test)
```

Out[22]:
```
array([2, 2, 2, 2, 1, 2, 0, 2, 1, 0, 1, 1, 2, 0, 1, 0, 0, 2, 1, 0, 0, 0,
       0, 2, 1, 2, 1, 2, 1, 1])
```

In [23]:
```python
model.score(x_test,y_test)
```

Out[23]: 0.9666666666666667

In [24]:
```python
model_c=SVC(C=1)
```

In [25]:
```python
model_c.fit(x_train,y_train)
```

Out[25]: SVC(C=1)

In [26]:
```python
model_c.score(x_test,y_test)
```

Out[26]: 0.9666666666666667

In [27]:
```python
model_g=SVC(gamma=1)
```

In [28]:
```python
model_g.fit(x_train,y_train)
```

Out[28]: SVC(gamma=1)

In [29]:
```python
model_g.score(x_test,y_test)
```

Out[29]: 0.9666666666666667

In [30]:
```python
model_kernel=SVC(kernel='rbf')
```

In [31]:
```python
model_kernel.fit(x_train,y_train)
```

Out[31]: SVC()

In [32]:
```python
model_kernel.score(x_test,y_test)
```

Out[32]: 0.9666666666666667

In [ ]: