In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
x = np.array([95,85,80,70,60])
y = np.array([85,95,70,65,70])
```

In [3]:
```
model = np.polyfit(x,y,1)
```

In [4]:
```
model
```

Out[4]:
```
array([ 0.64383562, 26.78082192])
```

In [5]:
```
predict = np.poly1d(model)
predict(65)
```

Out[5]:
```
68.63013698630137
```

In [6]:
```
y_pred = predict(x)
y_pred
```

Out[6]:
```
array([87.94520548, 81.50684932, 78.28767123, 71.84931507, 65.4109589 ])
```

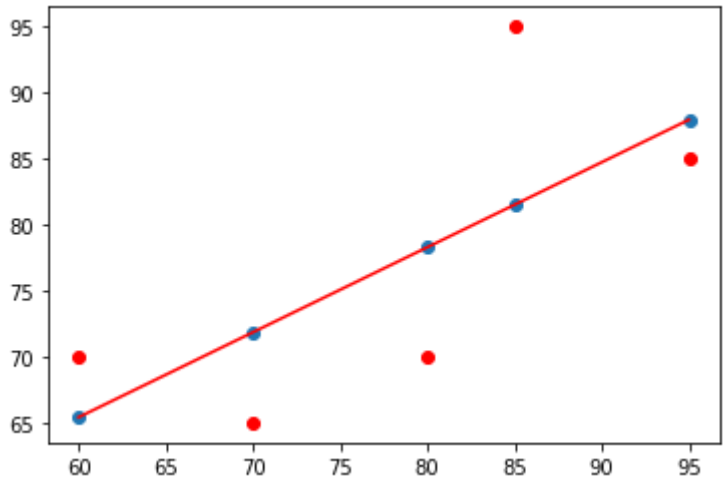
using sklearn

In [7]:
```
from sklearn.metrics import r2_score
r2_score(y, y_pred)
```

Out[7]:
```
0.4803218090889326
```

In [8]:
```
y_line = model[1] + model[0]* x
plt.plot(x, y_line, c = 'r')
plt.scatter(x, y_pred)
plt.scatter(x,y,c='r')
```

Out[8]:
```
<matplotlib.collections.PathCollection at 0x27af3958c10>
```



In [9]:
```
pip install sklearn
```

```
Requirement already satisfied: sklearn in c:\users\hp\anaconda3\lib\site-packages
(0.0)
Requirement already satisfied: scikit-learn in c:\users\hp\anaconda3\lib\site-packag
es (from sklearn) (0.24.2)
Requirement already satisfied: joblib>=0.11 in c:\users\hp\anaconda3\lib\site-packag
es (from scikit-learn->sklearn) (1.1.0)
Requirement already satisfied: scipy>=0.19.1 in c:\users\hp\anaconda3\lib\site-packa
ges (from scikit-learn->sklearn) (1.7.1)
Requirement already satisfied: numpy>=1.13.3 in c:\users\hp\anaconda3\lib\site-packa
ges (from scikit-learn->sklearn) (1.20.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\hp\anaconda3\lib\sit
e-packages (from scikit-learn->sklearn) (2.2.0)
Note: you may need to restart the kernel to use updated packages.
```

Boston Dataset

In [10]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [11]:
```
pip install sklearn
```

```
Requirement already satisfied: sklearn in c:\users\hp\anaconda3\lib\site-packages
(0.0)
Requirement already satisfied: scikit-learn in c:\users\hp\anaconda3\lib\site-packag
es (from sklearn) (0.24.2)
Requirement already satisfied: scipy>=0.19.1 in c:\users\hp\anaconda3\lib\site-packa
ges (from scikit-learn->sklearn) (1.7.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\hp\anaconda3\lib\sit
e-packages (from scikit-learn->sklearn) (2.2.0)
Requirement already satisfied: joblib>=0.11 in c:\users\hp\anaconda3\lib\site-packag
es (from scikit-learn->sklearn) (1.1.0)
Requirement already satisfied: numpy>=1.13.3 in c:\users\hp\anaconda3\lib\site-packa
ges (from scikit-learn->sklearn) (1.20.3)
Note: you may need to restart the kernel to use updated packages.
```

In [12]:
```
df = pd.read_csv("housing.csv")
df
```

Out[12]:

	0.00632 18.00 2.310 0 0.5380 6.5750 65.20 4.0900 1 296.0 15.30 396.90 4.98 24.00

	0.00632 18.00 2.310 0 0.5380 6.5750 65.20 4.0900 1 296.0 15.30 396.90 4.98 24.00
0	0.02731 0.00 7.070 0 0.4690 6.4210 78...
1	0.02729 0.00 7.070 0 0.4690 7.1850 61...
2	0.03237 0.00 2.180 0 0.4580 6.9980 45...
3	0.06905 0.00 2.180 0 0.4580 7.1470 54...
4	0.02985 0.00 2.180 0 0.4580 6.4300 58...
...	...
500	0.06263 0.00 11.930 0 0.5730 6.5930 69...
501	0.04527 0.00 11.930 0 0.5730 6.1200 76...
502	0.06076 0.00 11.930 0 0.5730 6.9760 91...
503	0.10959 0.00 11.930 0 0.5730 6.7940 89...
504	0.04741 0.00 11.930 0 0.5730 6.0300 80...

505 rows × 1 columns

In [13]:
```
from sklearn.datasets import load_boston
boston = load_boston()
data = pd.DataFrame(boston.data)
data.columns = boston.feature_names
data.head()
```

Out[13]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

In [14]:
```
data['PRICE'] = boston.target
data.isnull().sum()
x = data.drop(['PRICE'], axis = 1)
y = data['PRICE']
```

In [15]:
```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size =0.2,random_state =
```

In [16]:
```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
model = lm.fit(xtrain, ytrain)
```

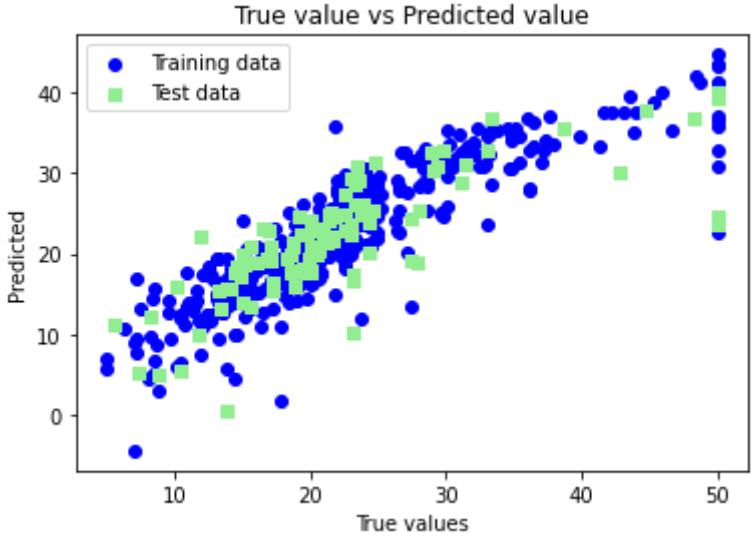
In [17]:
```
ytrain_pred = lm.predict(xtrain)
ytest_pred = lm.predict(xtest)
```

In [18]:
```
df=pd.DataFrame(ytrain_pred,ytrain)
df=pd.DataFrame(ytest_pred,ytest)
```

In [19]:
```
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(ytest, ytest_pred)
print(mse)
mse = mean_squared_error(ytrain_pred,ytrain)
print(mse)
```

```
33.448979997676524
19.326470203585725
```

In [20]:
```
plt.scatter(ytrain ,ytrain_pred,c='blue',marker='o',label='Training data')
plt.scatter(ytest,ytest_pred ,c='lightgreen',marker='s',label='Test data')
plt.xlabel('True values')
plt.ylabel('Predicted')
plt.title("True value vs Predicted value")
plt.legend(loc= 'upper left') #plt.hlines(y=0,xmin=0,xmax=50)
plt.plot()
plt.show()
```



Conclusion:

In [21]:
```
# In  this  way  we have done   data  analysis  using linear regression
# for Boston Dataset and predict the price of houses using the
# features of the Boston Dataset.
```

In [ ]:
```
```