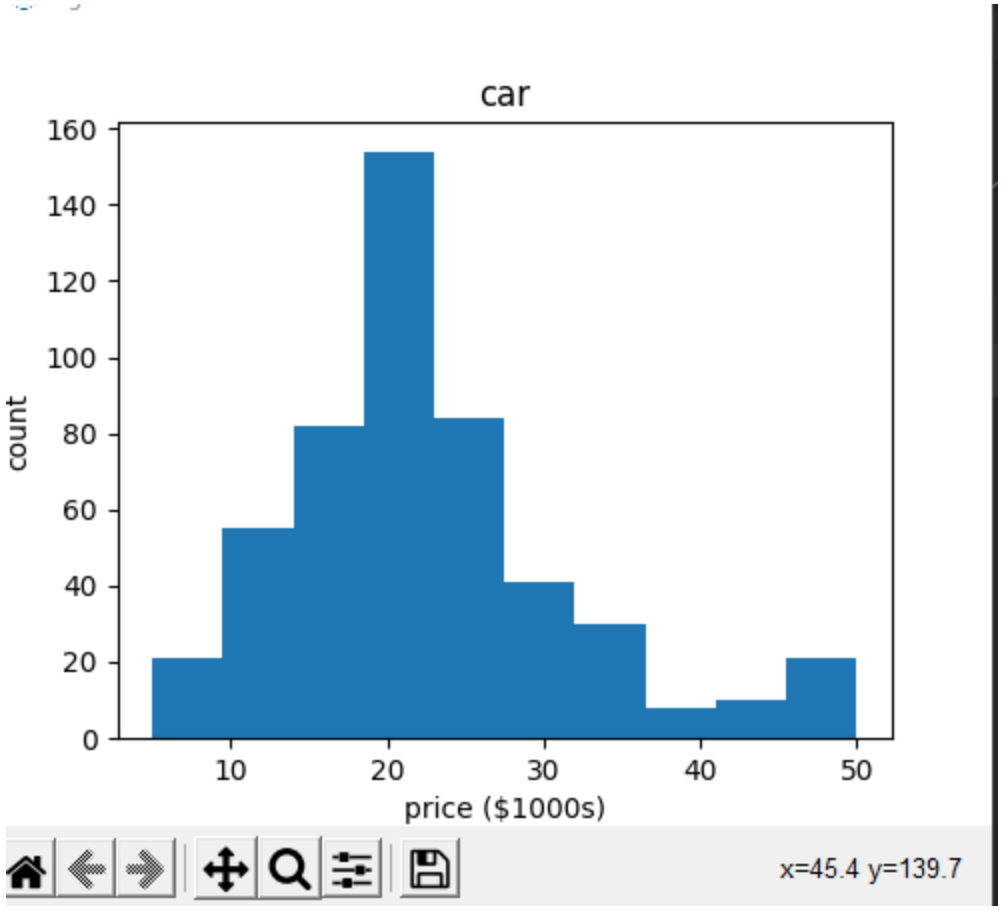
**AIM:**

Program to implement linear and multiple regression techniques using any standards datasets available in the public domain and evaluate its performance.

**PROGRAM**

```
import pandas

from sklearn.datasets import load_boston
import pandas as pd
import matplotlib.pyplot as plt

df = pandas.read_csv("cars.csv")

X = df[['Weight', 'Volume']]
y = df['CO2']
from sklearn import linear_model

regr = linear_model.LinearRegression()
regr.fit(X, y)

#predict the CO2 emission of a car where the weight is 2300kg, and the volume is 1300cm3:
predictedCO2 = regr.predict([[2300, 1300]])

print(predictedCO2)

boston = load_boston()
plt.figure(figsize=(5, 4))
plt.hist(boston.target)
plt.title('car')
plt.xlabel('price ($1000s)' )
plt.ylabel('count')
plt.show()
```

**OUTPUT**




**PROGRAM**

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model, metrics

# load the boston dataset
boston = datasets.load_boston(return_X_y=False)

# defining feature matrix(X) and response vector(y)
X = boston.data
y = boston.target

# splitting X and y into training and testing sets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
                                                    random_state=1)
```

```
# create linear regression object
reg = linear_model.LinearRegression()

# train the model using the training sets
reg.fit(X_train, y_train)

# regression coefficients
print('Coefficients: ', reg.coef_)

# variance score: 1 means perfect prediction
print('Variance score: {}'.format(reg.score(X_test, y_test)))
```

**OUTPUT**

```
Coefficients:  [-8.95714048e-02  6.73132853e-02  5.04649248e-02  2.18579583e+00
 -1.72053975e+01  3.63606995e+00  2.05579939e-03 -1.36602886e+00
  2.89576718e-01 -1.22700072e-02 -8.34881849e-01  9.40360790e-03
 -5.04008320e-01]
Variance score: 0.7209056672661767
```