Multiple Linear Regression :-

```
import pandas as pd

import numpy as np


path = 'D://Programing/ML/SK Learn/Machine Learning A-Z Template Folder/Part 2 -
Regression/P14-Part2-Regression/Section 7 - Multiple Linear
Regression/Python/50_Startups.csv'

dataset = pd.read_csv(path)


#turning data to matrix

col = dataset.shape[1]

X = dataset.iloc[ : , :col-1].values

y = dataset.iloc[ : , col-1:col].values


#encodering for dataset

from sklearn.preprocessing import LabelEncoder , OneHotEncoder

labelencoder = LabelEncoder()

X[ : , 3] = labelencoder.fit_transform(X[ : , 3])

onehotencoder = OneHotEncoder(categorical_features = [3])

X = onehotencoder.fit_transform(X).toarray()


#Avoiding the dummy variable trap

X = X[ : , 1 : ] #we don't need to do that her becouse the python liberary do that automaticly

            #but we some time need to do ite once manually


#spliting the dataset to training set and test set

from sklearn.model_selection import train_test_split

X_train , X_test , y_train , y_test = train_test_split(X , y , test_size = 1/5 , random_state = 0)
```

```
#fiting the training set

from sklearn.linear_model import LinearRegression

Mlinearregression = LinearRegression()

Mlinearregression.fit(X_train , y_train)


#predicting the X test

y_pred = Mlinearregression.predict(X_test)


#building the optimal model using backward elimination

import statsmodels.formula.api as sm

X = np.append(arr = np.ones((50,1)) , values = X , axis = 1)

X_opt = X[ : , [0,1,2,3,4,5,6]]

Mlinearregression_OLS = sm.OLS(endog = y , exog = X_opt).fit()

Mlinearregression_OLS.summary()

X_opt = X[ : , [0,1,2,3,4,6]]

Mlinearregression_OLS = sm.OLS(endog = y , exog = X_opt).fit()

Mlinearregression_OLS.summary()

#that without excude Avoiding the dummy variable trap code ..with using it in the tuturial

#we do the three lines ones then check P value , if we find P value boger than 0.05 we delete
that line

#then we try it agein and agein until all P values be smaller than 0.05
```

if you are also interested in some automatic implementations of Backward Elimination in Python, please find two of them below:

Backward Elimination with p-values only:

```
import statsmodels.formula.api as sm

def backwardElimination(x, sl):
    numVars = len(x[0])
    for i in range(0, numVars):
        regressor_OLS = sm.OLS(y, x).fit()
        maxVar = max(regressor_OLS.pvalues).astype(float)
        if maxVar > sl:
            for j in range(0, numVars - i):
                if (regressor_OLS.pvalues[j].astype(float) == maxVar):
                    x = np.delete(x, j, 1)
    regressor_OLS.summary()
    return x

SL = 0.05
X_opt = X[:, [0, 1, 2, 3, 4, 5]]
X_Modeled = backwardElimination(X_opt, SL)
```