In [1]:

```
#Loading Packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
```

In [2]:

```
#importing dataset
data = pd.read_csv("E:\\Datasets\\DataSet\\Family Income.csv")
```

In [3]:

```
#checking null values
data.isnull().sum()

#drop null values (or) replacing values with mean/most_frequent
data = data.drop(['Household Head Occupation','Household Head Class of Worker'],axis=1)
```

In [4]:

```
#converting categorical into binary
from sklearn.preprocessing import LabelEncoder
lencoder = LabelEncoder()
data.iloc[:, 25:26]= lencoder.fit_transform(data.iloc[:,25:26])
data.iloc[:, 29:30]= lencoder.fit_transform(data.iloc[:,29:30])
```

```
C:\Users\Personal\Anaconda\lib\site-packages\sklearn\preprocessing\label.p
y:235: DataConversionWarning: A column-vector y was passed when a 1d array
was expected. Please change the shape of y to (n_samples, ), for example u
sing ravel().
  y = column_or_1d(y, warn=True)
```

In [5]:

```
#slicing the data
x= data.iloc[:,[2,14,15,18,20,24]]
y= data.iloc[:,0]
x.columns
```

Out[5]:

```
Index(['Total Food Expenditure',
       'Clothing, Footwear and Other Wear Expenditure',
       'Housing and water Expenditure', 'Transportation Expenditure',
       'Education Expenditure',
       'Total Income from Entrepreneurial Acitivites'],
      dtype='object')
```

In [6]:

```
#Model selection  and splitting
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2,random_state=0)
```

In [7]:

```
#model implimentation
from sklearn.linear_model import LinearRegression
regressor  = LinearRegression()
regressor.fit(x_train,y_train)
```

Out[7]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=F
alse)
```

In [8]:

```
#model prediction
regressor.predict(x_test)
```

Out[8]:

```
array([252505.44058364, 130752.47454519, 184074.83346863, ...,
       101847.78849308, 489179.09669978, 141086.24765936])
```

In [10]:

```
#model metrics
a=regressor.score(x_train,y_train)
b=regressor.score(x_test,y_test)
```

In [14]:

```
#checking Variance influence factor
vif = 1/(1-a)
print(a)
print(b)
print(vif)
```

```
0.7867521635748046
0.7129189582039273
4.689379347352896
```