In [1]:
```
# Multiple Linear Regression

# Importing the libraries
import os
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [2]:
```
os.chdir("D:/My ML Simulations/My_ML_Work/Part 2 - Regression/Section 5 - Multiple Linear Regression")
```

In [4]:
```
# Importing the dataset
dataset = pd.read_csv('50_Startups.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 4].values
```

In [13]:
```
# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder = LabelEncoder()
X[:, 3] = labelencoder.fit_transform(X[:, 3])
onehotencoder = OneHotEncoder(categories='auto')
X = onehotencoder.fit_transform(X).toarray()
```

In [14]:
```
# Encoding categorical data
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder = LabelEncoder()
X[:, 3] = labelencoder.fit_transform(X[:, 3])
onehotencoder = OneHotEncoder(categories='auto')
X = onehotencoder.fit_transform(X).toarray()
```

In [16]:
```
# Avoiding the Dummy Variable Trap
X = X[:, 1:]
```

In [17]:
```
# Splitting the dataset into the 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 = 0.2, random_state = 0)
```

In [18]:
```
# Feature Scaling
"""from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)
sc_y = StandardScaler()
y_train = sc_y.fit_transform(y_train.reshape(-1,1))"""
```

Out[18]:
```
'from sklearn.preprocessing import StandardScaler\nsc_X = StandardScaler()\nX_train = sc_X.fit_transform(X_train)\nX_test = sc_
X.transform(X_test)\nsc_y = StandardScaler()\ny_train = sc_y.fit_transform(y_train.reshape(-1,1))'
```

In [19]:
```
# Fitting Multiple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[19]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [20]:
```
# Predicting the Test set results
y_pred = regressor.predict(X_test)
```