In [149]:

```
#importing the libraries for predicting profit for companies
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
%matplotlib inline
```

In [150]:

```
#importing dataset
companies = pd.read_csv(r"C:\Users\G.SAI KRISHNA\Desktop\ML_Projects\profit_prediction\1
000_Companies.csv")
#Independent var's
X = companies.iloc[:,:-1].values #fetch rows until second last column
#Dependent Var i.e. Profit
Y = companies.iloc[:,4].values

companies.head()
```

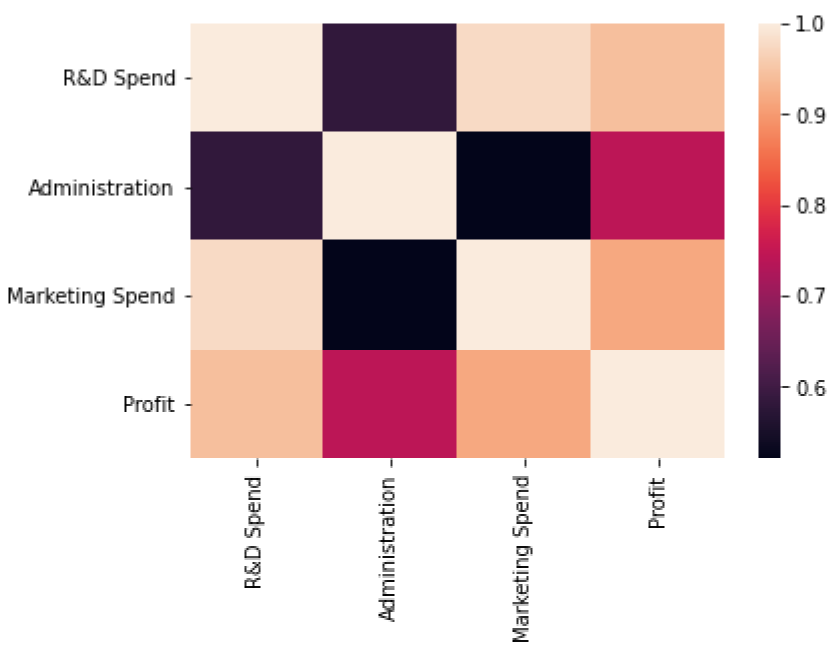
Out[150]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94

In [151]:

```
#Data visualization through Correlation Matrix
sns.heatmap(companies.corr())
```

Out[151]: <matplotlib.axes._subplots.AxesSubplot at 0x1e755e6c070>



In [152]:
```
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.compose import ColumnTransformer
#Encode Country Column
ct = ColumnTransformer([("State", OneHotEncoder(), [3])], remainder = 'passthrough')
X = ct.fit_transform(X)

X
```

Out[152]:
```
array([[0.0, 0.0, 1.0, 165349.2, 136897.8, 471784.1],
       [1.0, 0.0, 0.0, 162597.7, 151377.59, 443898.53],
       [0.0, 1.0, 0.0, 153441.51, 101145.55, 407934.54],
       ...,
       [1.0, 0.0, 0.0, 100275.47, 241926.31, 227142.82],
       [1.0, 0.0, 0.0, 128456.23, 321652.14, 281692.32],
       [0.0, 0.0, 1.0, 161181.72, 270939.86, 295442.17]], dtype=object)
```

In [153]:
```
X=X[:,1:]
print(X)
```

```
[[0.0 1.0 165349.2 136897.8 471784.1]
 [0.0 0.0 162597.7 151377.59 443898.53]
 [1.0 0.0 153441.51 101145.55 407934.54]
 ...
 [0.0 0.0 100275.47 241926.31 227142.82]
 [0.0 0.0 128456.23 321652.14 281692.32]
 [0.0 1.0 161181.72 270939.86 295442.17]]
```

In [154]:
```
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 [155]:
```
#fitting multiple linear regression model
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train,Y_train)
```

Out[155]:
```
LinearRegression()
```

In [156]:
```
Y_pred = regressor.predict(X_test)
print(Y_pred)
```

```
[ 89790.61532915  88427.07187361  94894.67836972 175680.86725611
  83411.73042088 110571.90200074 132145.22936439  91473.37719686
 164597.05380606  53222.82667401  66950.19050989 150566.43987005
 126915.20858596  59337.8597105  177513.91053062  75316.28143051
 118248.14406603 164574.40699902 170937.2898107  182069.11645084
 118845.03252689  85669.95112229 180992.59396144  84145.08220145
 105005.83769214 101233.56772747  53831.07669091  56881.41475224
  68896.39346905 210040.00765883 120778.72270894 111724.87157654
 101487.90541518 137959.02649624  63969.95996743 108857.91214126
 186014.72531988 171442.64130747 174644.26529205 117671.49128195
  96731.37857433 165452.25779409 107724.34331255  50194.54176913
 116513.89532179  58632.4898682  158416.4682761   78541.48521609
 159727.66671743 131137.87699644 184880.70924516 174609.0826688
  93745.66352059  78341.13383418 180745.9043908   84461.61490552
 142900.90602903 170618.44098397  84365.09530839 105307.3716218
 141660.07290787  52527.34340442 141842.9626416  139176.27973195
  98294.52669666 113586.86790969 126754.21895489 152135.51985562
  58864.51658955 174285.5736113  124624.04380784 169065.77658978
  91279.33198209 156170.37268962  84307.26579366  77877.75223097
 120414.02421346  93380.44273241 139020.62514121 143604.67103573
 171148.30815368 140082.97050131 106369.71689747 155641.43851388
 140030.10330037 110172.87893525  69672.98677565  88148.52068041
 140133.59925093 148479.09537887 157916.63505257  58532.94863141
  93707.3842239  112646.37475705  56556.1894366  107414.89996181
 147352.80227752 152144.10104034 167808.11701784 118750.25230713
 120763.27666701 139029.95295663 157527.90934119 121962.0621496
  87091.32399736 104792.91384333  95335.22679185 178389.52287436
 181942.63776381 109831.34945506 165254.03344096 167806.06491902
 158002.44642543 174782.86900956 170196.77102699  52302.18161612
 176938.1159579  104751.83583865  82710.31528805 138890.52767844
 144274.74675425 161679.0183644  169662.05445895 120450.9231013
 158880.70799547 110213.73252824 169674.51532366  60760.61300841
 159036.99629068 158169.44286048 174511.70494475 156294.79927784
 103714.37583212  85635.96237574 141603.54878757 165917.6915698
 121182.03641977 170751.87883894 100505.77549411  82097.51033128
 178643.18879843 101790.48384578  70507.40958622  90250.04230088
  61247.4996268   68912.1753452   72775.81613475 176914.08873124
  89704.69244931 129209.43730015  92672.90938383  88133.59175044
 172836.33021619  60893.62070014 169015.8944601  166450.24453204
 165425.54476415 102170.5169499  181594.57928216  73702.57942561
  91267.42979668 135791.54160195  64922.802573    71775.70235726
  60603.91401515 184288.61041916 176286.69585945 158907.75687039
 141359.32216439 154611.17928321  58549.58863233  90618.58407899
 152885.51163925 168398.05223805  72485.36274539 116064.24350667
  80087.80697208 149828.90896188 116806.9595737  130191.48845161
 174534.42670329 293584.45948283 146270.83174789 150646.69178014
  86107.47782247  69967.20842246  70096.78368773  69033.69170769
 120666.75708063  89677.68014063 166824.27091662 125514.76626409
  67209.67687466 140930.69427702 118544.30490695 165897.61905906
 168655.48652552 147009.66805049 141396.22104146 109086.50634849]
```

In [157]:
```
print(regressor.coef_)
```

```
[-8.80536598e+02 -6.98169073e+02  5.25845857e-01  8.44390881e-01
  1.07574255e-01]
```

In [158]:
```
print(regressor.intercept_)
```

```
-51035.22972402591
```

In [159]:
```
from sklearn.metrics import r2_score
r2_score(Y_test,Y_pred)
```

Out[159]:
```
0.9112695892268863
```