In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [21]:
```
data = pd.read_csv('Org_profit.csv')
```

In [22]:
```
data.head()
```

Out[22]:

	Research Spend	Daily usages	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 [23]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 5 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Research Spend   50 non-null     float64
 1   Daily usages     50 non-null     float64
 2   Marketing Spend  50 non-null     float64
 3   State            50 non-null     object 
 4   Profit           50 non-null     float64
dtypes: float64(4), object(1)
memory usage: 2.1+ KB
```

In [24]:
```
# Independent Variable and Dependent Variable
x = data.iloc[:,:-2].values
y = data.iloc[:, -1].values
```

In [26]:
```
# 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 [27]:
```
# # Fitting Multiple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train, y_train)
```

Out[27]:
```
LinearRegression()
```

In [28]:
```
# Predicting the Test set results
y_pred = regressor.predict(x_test)
```

In [29]:
```
y_test-y_pred
```

Out[29]:
```
array([  -619.5169696 ,  11496.34006874,  12554.04629956,   4887.04023264,
        11422.46432776, -10158.33864795,  14115.4830943 ,   -671.24686776,
        -4403.86555221,  -2876.07408795])
```

In [30]:
```
# accuracy metrics
from sklearn.metrics import mean_squared_error,r2_score
rmse = (np.sqrt(mean_squared_error(y_test, y_pred)))
r2score = r2_score(y_test, y_pred)
```

In [31]:
```
print(rmse)
print(r2score)
```

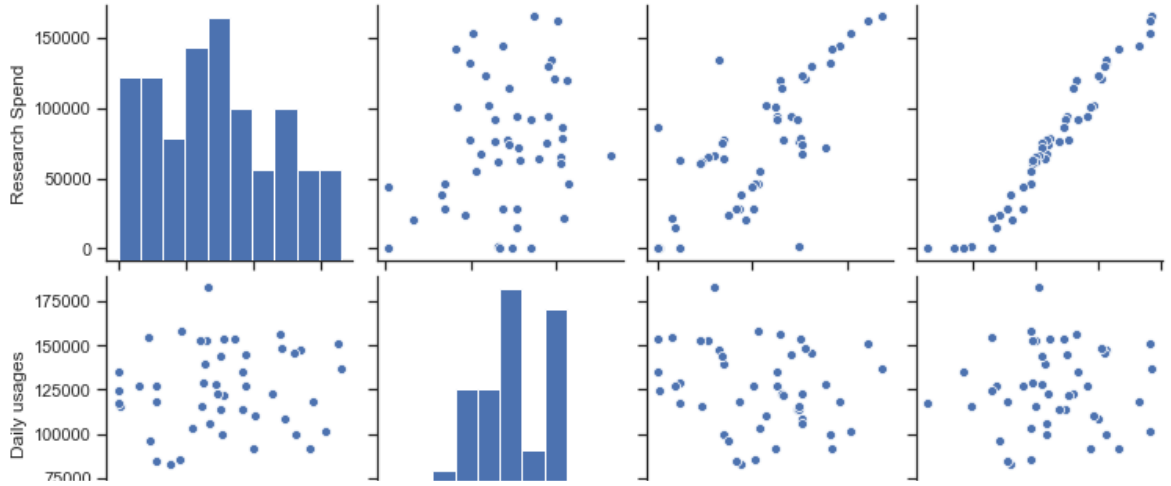
```
8803.775790469343
0.9393955917820571
```

In [19]:
```
regressor.score(x_test,y_test)
```

Out[19]:
```
0.9393955917820571
```

In [151]:
```
sns.set(style="ticks")
sns.pairplot(data)
```

Out[151]:
```
<seaborn.axisgrid.PairGrid at 0x274e3843a48>
```



In [152]:
```
data.corr()
```

Out[152]:

	Research Spend	Daily usages	Marketing Spend	Profit
Research Spend	1.000000	0.241955	0.724248	0.972900
Daily usages	0.241955	1.000000	-0.032154	0.200717
Marketing Spend	0.724248	-0.032154	1.000000	0.747766
Profit	0.972900	0.200717	0.747766	1.000000

In [32]:
```
x1 = data.iloc[:,[0,2]].values
y1 = data.iloc[:,-1].values
```

In [33]:
```
x1
```

Out[33]:
```
array([[165349.2 , 471784.1 ],
       [162597.7 , 443898.53],
       [153441.51, 407934.54],
       [144372.41, 383199.62],
       [142107.34, 366168.42],
       [131876.9 , 362861.36],
       [134615.46, 127716.82],
       [130298.13, 323876.68],
       [120542.52, 311613.29],
       [123334.88, 304981.62],
       [101913.08, 229160.95],
       [100671.96, 249744.55],
       [ 93863.75, 249839.44],
       [ 91992.39, 252664.93],
       [119943.24, 256512.92],
       [114523.61, 261776.23],
       [ 78013.11, 264346.06],
       [ 94657.16, 282574.31],
       [ 91749.16, 294919.57],
       [ 86419.7 ,      0.  ],
       [ 76253.86, 298664.47],
       [ 78389.47, 299737.29],
       [ 73994.56, 303319.26],
       [ 67532.53, 304768.73],
       [ 77044.01, 140574.81],
       [ 64664.71, 137962.62],
       [ 75328.87, 134050.07],
       [ 72107.6 , 353183.81],
       [ 66051.52, 118148.2 ],
       [ 65605.48, 107138.38],
       [ 61994.48,  91131.24],
       [ 61136.38,  88218.23],
       [ 63408.86,  46085.25],
       [ 55493.95, 214634.81],
       [ 46426.07, 210797.67],
       [ 46014.02, 205517.64],
       [ 28663.76, 201126.82],
       [ 44069.95, 197029.42],
       [ 20229.59, 185265.1 ],
       [ 38558.51, 174999.3 ],
       [ 28754.33, 172795.67],
       [ 27892.92, 164470.71],
       [ 23640.93, 148001.11],
       [ 15505.73,  35534.17],
       [ 22177.74,  28334.72],
       [  1000.23,   1903.93],
       [  1315.46, 297114.46],
       [     0.  ,      0.  ],
       [   542.05,      0.  ],
       [     0.  ,  45173.06]])
```

In [34]:
```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
x1_train, x1_test, y1_train, y1_test = train_test_split(x1, y1, test_size = 0.2, random_state = 0)
```

In [35]:
```
# # Fitting Multiple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x1_train, y1_train)
```

Out[35]:
```
LinearRegression()
```

In [36]:
```
# Predicting the Test set results
y1_pred = regressor.predict(x1_test)
```

In [37]:
```
y1_pred
```

Out[37]:
```
array([102284.64605183, 133873.92383812, 134182.1495165 ,  73701.1069363 ,
       180642.25299736, 114717.24903894,  68335.07575312,  97433.45922275,
       114580.92136452, 170343.31979498])
```

In [39]:
```
# accuracy metrics
from sklearn.metrics import mean_squared_error,r2_score
rmse = (np.sqrt(mean_squared_error(y1_test, y1_pred)))
r2score = r2_score(y1_test, y1_pred)
```

In [40]:
```
print(rmse)
print(r2score)
```

```
8198.797190788484
0.9474386447268489
```

In [ ]:
```

```