IMPORTING DATASET

In [ ]:

```
import numpy as np #for performing calculations
import pandas as pd # for handling dataset
import matplotlib.pyplot as plt # for visualization
import sklearn # for model development
import seaborn as sns # for visualization
```

In [ ]:

```
dataset=pd.read_csv('50_Startups.csv')
dataset.head()
```

In [ ]:

```
dataset.shape
```

CHECKING DATA FOR ANY ERROR

In [ ]:

```
dataset.info()
```

In [ ]:

```
dataset.duplicated().sum()
```

In [ ]:

```
dataset.isnull().sum()
```

There are no duplicated or null values in dataset. Hence, no need to clean the data.

ANALYSIS OF DATASET

In [ ]:

```
dataset.describe()
```

In [ ]:

```
correlation = dataset.corr()
correlation
```

In [ ]:

```
heat=sns.heatmap(correlation,annot=True,cmap='Blues')
```

In [ ]:

```
grid=sns.pairplot(dataset)
```

From heatmap and pairplotting we found that there is a positive correlation between R&D Spend, Marketing Spend and profit.

In [ ]:

```
outliers = ['Profit']
plt.rcParams['figure.figsize'] = [8,8]
sns.boxplot(data=dataset[outliers], orient="v", palette="Set2" , width=0.7)
plt.title("Outliers Variable Distribution")
plt.ylabel("Profit Range")
plt.xlabel("Continuous Variable")

plt.show()
```

MODEL

spliting Dataset in Dependent & Independent Variables

In [ ]:

```
X = dataset.drop('Profit',axis=1)
y = dataset['Profit']
```

spliting dataset into training and testing data

In [ ]:

```
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test = train_test_split(X,y,train_size=0.7,random_state=0)
```

Training model

In [ ]:

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()
m=model.fit(x_train.values,y_train)
```

In [ ]:

```
y_pred = model.predict(x_test.values)
y_pred
```

In [ ]:

```
df = pd.DataFrame(data={'Actual Value':y_test,'Predicted value':y_pred})
df
```

Model evaluation

In [ ]:

```
from sklearn.metrics import r2_score

Score = r2_score(y_pred, y_test)
print("score :" ,Score*100)
```

The predicted value is close to actual value and model score is also good. Hence, this model can be used for prediction

Example

In [ ]:

```
example=pd.DataFrame({'R&D':199000,'Administration':75000,'Marketing':160000},index=[1])
example
```

In [ ]:

```
predict=m.predict(example.values)
predict
```