Problem Statement:3

In [1]:
```
import pandas as pd
```

In [4]:
```
Advertising=pd.read_excel("E:\Data science training\R AND PYTHON KPMG\stat and ml\ASSIGNMENT\Advertising Budget a
Advertising.head()
```

Out[4]:

	Unnamed: 0	TV Ad Budget ($)	Radio Ad Budget ($)	Newspaper Ad Budget ($)	Sales ($)
0	1	230.1	37.8	69.2	22.1
1	2	44.5	39.3	45.1	10.4
2	3	17.2	45.9	69.3	9.3
3	4	151.5	41.3	58.5	18.5
4	5	180.8	10.8	58.4	12.9

In [6]:
```
# Adding the Columns Which are relavant for Analysis.
Advertising1=Advertising[['TV Ad Budget ($)','Radio Ad Budget ($)','Newspaper Ad Budget ($)','Sales ($)']]
Advertising1.head()
```

Out[6]:

	TV Ad Budget ($)	Radio Ad Budget ($)	Newspaper Ad Budget ($)	Sales ($)
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	9.3
3	151.5	41.3	58.5	18.5
4	180.8	10.8	58.4	12.9

In [8]:
```
Advertising1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   TV Ad Budget ($)        200 non-null    float64
 1   Radio Ad Budget ($)     200 non-null    float64
 2   Newspaper Ad Budget ($)  200 non-null    float64
 3   Sales ($)               200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
```

In [9]:
```
#checking For Null value present in DataSet.
Advertising1.isnull().sum()
```

Out[9]:
```
TV Ad Budget ($)           0
Radio Ad Budget ($)        0
Newspaper Ad Budget ($)    0
Sales ($)                  0
dtype: int64
```

In [11]:
```
## since there is 3 input variables i.e. TV Ad Budget ,Radio Ad Budget ,Newspaper Ad Budget
  # we using Multiple Linear Regression(MLR) hrer.

### Defining the X and Y
Y = Advertising1[['Sales ($)']]

X = Advertising1.drop(columns=['Sales ($)'])
```

In [12]:
```
X.head()
```

Out[12]:

	TV Ad Budget ($)	Radio Ad Budget ($)	Newspaper Ad Budget ($)
0	230.1	37.8	69.2

1	44.5	39.3	45.1
2	17.2	45.9	69.3
3	151.5	41.3	58.5
4	180.8	10.8	58.4

In [13]:
```
#For building the model i keep train_data  to 80%
#### and for testing i should keep 20% of datapoints

from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X,Y, train_size = 0.8, random_state = 1234)

len(X_train), len(X_test), len(Y_train), len(Y_test)
```

Out[13]:
```
(160, 40, 160, 40)
```

In [14]:
```
## Define our model object

from sklearn.linear_model import LinearRegression

mlr = LinearRegression()

### Fit this model object on our training dataset

model = mlr.fit(X_train, Y_train)

model
```

Out[14]:
```
LinearRegression()
```

In [16]:
```
#Prepare a multiple linear regression model Y = m1X1 + m2X2 +..... + C alongwith the R-sq value.

#Y -> Sales

#X -> TV Ad Budget ,Radio Ad Budget ,Newspaper Ad Budget

#  Find the values of slope, intercept

## slope is m, Intercept is C, X is gpa, Y is admit

print(model.coef_) # this will give the m value
print(model.intercept_) # this will give me the Constant/Intercept value
```

```
[[0.04560079 0.18927341 0.00237545]]
[2.8496683]
```

In [17]:
```
# Find the R-sq value of my model .

r_sq = model.score(X_train,Y_train)
r_sq
```

Out[17]:
```
0.8938358233693336
```

In [18]:
```
## 2) Find the RMSE value of the Model
Y_test['Pred_Sales'] = model.predict(X_test)
```

In [20]:
```
Y_test.head()
```

Out[20]:

	Sales ($)	Pred_Sales
197	12.8	12.696454
157	10.1	9.984446
31	11.9	11.383047
48	14.8	16.319223
63	14.0	13.155316

In [22]:
```
##### Error i.e. (Sales ($) - Pred_Sales)
Y_test['Error'] = Y_test['Sales ($)'] - Y_test['Pred_Sales']
```

In [23]:
```
##### Square of Error
Y_test['Sq_Error'] = (Y_test['Error']) **2
```

In [24]:
```
Y_test.head()
```

Out[24]:

	Sales ($)	Pred_Sales	Error	Sq_Error
197	12.8	12.696454	0.103546	0.010722
157	10.1	9.984446	0.115554	0.013353
31	11.9	11.383047	0.516953	0.267240
48	14.8	16.319223	-1.519223	2.308039
63	14.0	13.155316	0.844684	0.713490

In [25]:
```
## Mean of Sq_Error

Error_mean = Y_test['Sq_Error'].mean()
```

In [26]:
```
## Find the Square root value of Error_mean

import math

RMSE = math.sqrt(Error_mean)
RMSE
```

Out[26]:
```
1.704667427572053
```

SO RMSE VALUE IS 1.704667427572053\

In [27]:
```
## C) What would be the Sales if an organization decides to allocate 150 as TV budget,
 ##           50 as Radio Budget and 60 as Newspaper Buddget
```

In [32]:
```
## Create a validation data
val_data = pd.DataFrame({"TV Ad Budget ($)" : [150],"Radio Ad Budget ($)" : [50],"Newspaper Ad Budget ($)" : [60]

#### predict the Sales offered
Predict_sales = model.predict(val_data)
Predict_sales
```

Out[32]:
```
array([[19.29598443]])
```

In [34]:
```
# Upper Range of Prediction

print('Upper Range of Sales is ', Predict_sales + RMSE)
```

```
Upper Range of Sales is  [[21.00065185]]
```

In [35]:
```
# Lower Range of Prediction

print('Lower Range of Sales is ', Predict_sales - RMSE)
```

```
Lower Range of Sales is  [[17.591317]]
```

In [ ]: