Assignment 01: Evaluate the Ad Budget Dataset of XYZ Firm

The comments/sections provided are your cues to perform the assignment. You don't need to limit yourself to the number of rows/cells provided. You can add additional rows in each section to add more lines of code.

If at any point in time you need help on solving this assignment, view our demo video to understand the different steps of the code.

Happy coding!

1: Import the dataset

In [1]:

```
#Import the required libraries
import pandas as pd
```

In [30]:

```
#Import the advertising dataset
df=pd.read_csv("Advertising Budget and Sales.csv", index_col=0)
df
```

Out[30]:

	TV Ad Budget ($)	Radio Ad Budget ($)	Newspaper Ad Budget ($)	Sales ($)
1	230.1	37.8	69.2	22.1
2	44.5	39.3	45.1	10.4
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9
...	...	...	...	...
196	38.2	3.7	13.8	7.6
197	94.2	4.9	8.1	9.7
198	177.0	9.3	6.4	12.8
199	283.6	42.0	66.2	25.5
200	232.1	8.6	8.7	13.4

200 rows × 4 columns

2: Analyze the dataset

In [16]:

```
#View the initial few records of the dataset
df.head()
```

Out[16]:

	TV Ad Budget ($)	Radio Ad Budget ($)	Newspaper Ad Budget ($)	Sales ($)
1	230.1	37.8	69.2	22.1

2	TV Ad Budget ($)	Radio Ad Budget ($)	Newspaper Ad Budget ($)	Sales ($)
3	17.2	45.9	69.3	9.3
4	151.5	41.3	58.5	18.5
5	180.8	10.8	58.4	12.9

In [17]:

```
#Check the total number of elements in the dataset
df.size
```

Out[17]:

```
800
```

3: Find the features or media channels used by the firm

In [18]:

```
#Check the number of observations (rows) and attributes (columns) in the dataset
df.shape
```

Out[18]:

```
(200, 4)
```

In [19]:

```
#View the names of each of the attributes
df.columns
```

Out[19]:

```
Index(['TV Ad Budget ($)', 'Radio Ad Budget ($)', 'Newspaper Ad Budget ($)',
       'Sales ($)'],
      dtype='object')
```

4: Create objects to train and test the model; find the sales figures for each channel

In [32]:

```
#Create a feature object from the columns
x_feature=df[["TV Ad Budget ($)","Radio Ad Budget ($)","Newspaper Ad Budget ($)"]]
```

In [34]:

```
#View the feature object
x_feature.head()
```

Out[34]:

	TV Ad Budget ($)	Radio Ad Budget ($)	Newspaper Ad Budget ($)
1	230.1	37.8	69.2
2	44.5	39.3	45.1
3	17.2	45.9	69.3
4	151.5	41.3	58.5
5	180.8	10.8	58.4

In [35]:

```
#Create a target object (Hint: use the sales column as it is the response of the dataset)
y_target=df[["Sales ($)"]]
```

In [37]:

```
#View the target object
y_target.head()
```

Out[37]:

	Sales ($)
1	22.1
2	10.4
3	9.3
4	18.5
5	12.9

In [41]:

```
#Verify if all the observations have been captured in the feature object
x_feature.shape
```

Out[41]:

```
(200, 3)
```

In [42]:

```
#Verify if all the observations have been captured in the target object
y_target.shape
```

Out[42]:

```
(200, 1)
```

5: Split the original dataset into training and testing datasets for the model

In [47]:

```
#Split the dataset (by default, 75% is the training data and 25% is the testing data)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x_feature,y_target,random_state=1)
```

In [49]:

```
#Verify if the training and testing datasets are split correctly (Hint: use the shape() m
ethod)
print (x_train.shape)
print (x_test.shape)
print (y_train.shape)
print (y_test.shape)
```

```
(150, 3)
(50, 3)
(150, 1)
(50, 1)
```

6: Create a model to predict the sales outcome

In [50]:

```
#Create a linear regression model
from sklearn.linear_model import LinearRegression
linreg=LinearRegression()
linreg.fit(x_train,y_train)
```

Out[50]:

```
LinearRegression()
```

In [52]:

```
#Print the intercept and coefficients
print(linreg.intercept_)
print(linreg.coef_)
```

```
[2.87696662]
[[0.04656457 0.17915812 0.00345046]]
```

In [54]:

```
#Predict the outcome for the testing dataset
y_pred=linreg.predict(x_test)
y_pred
```

Out[54]:

```
array([[21.70910292],
       [16.41055243],
       [ 7.60955058],
       [17.80769552],
       [18.6146359 ],
       [23.83573998],
       [16.32488681],
       [13.43225536],
       [ 9.17173403],
       [17.333853  ],
       [14.44479482],
       [ 9.83511973],
       [17.18797614],
       [16.73086831],
       [15.05529391],
       [15.61434433],
       [12.42541574],
       [17.17716376],
       [11.08827566],
       [18.00537501],
       [ 9.28438889],
       [12.98458458],
       [ 8.79950614],
       [10.42382499],
       [11.3846456 ],
       [14.98082512],
       [ 9.78853268],
       [19.39643187],
       [18.18099936],
       [17.12807566],
       [21.54670213],
       [14.69809481],
       [16.24641438],
       [12.32114579],
       [19.92422501],
       [15.32498602],
       [13.88726522],
       [10.03162255],
       [20.93105915],
       [ 7.44936831],
       [ 3.64695761],
       [ 7.22020178],
       [ 5.9962782 ],
       [18.43381853],
       [ 8.39408045],
       [14.08371047],
       [15.02195699],
       [20.35836418],
       [20.57036347],
       [19.60636679]])
```

7: Calculate the Mean Square Error (MSE)

In [55]:

```
#Import required libraries for calculating MSE (mean square error)
```

```
from sklearn import metrics
import numpy as np
```

In [58]:

```
#Calculate the MSE
print (np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

```
1.4046514230328953
```

In [1]:

```
print("Project successfully completed on Analysing Ad Budjet for Different Media Channels")
```

```
Project successfully completed on Analysing Ad Budjet for Different Media Channels
```

In [3]:

```
print("Thank You Simplilearn")
```

```
Thank You Simplilearn
```

In [ ]: