In [1]:
```
# Aim: To perform Simple Linear Regression and Find out Coefficient of it.
```

In [2]:
```
# Name : Shriya Mechineni
# class : 3rd year
# Section : A
# Roll No. : 49
```

In [3]:
```
import numpy as np
import pandas as pd
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
from sklearn.linear_model import LinearRegression
```

In [4]:
```
import os
```

In [5]:
```
os.getcwd()
```

Out[5]:
```
'C:\\Users\\admin'
```

In [6]:
```
os.chdir("C:\\Users\\admin\\Desktop")
```

In [7]:
```
df=pd.read_csv("iris.csv")
```

In [8]:
```
df.head()
```

Out[8]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

In [9]:
```
df.head(10)
```

Out[9]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 5 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sepal_length  150 non-null    float64
 1   sepal_width   150 non-null    float64
 2   petal_length  150 non-null    float64
 3   petal_width   150 non-null    float64
 4   species       150 non-null    object 
dtypes: float64(4), object(1)
memory usage: 6.0+ KB
```

In [11]:
```
df.tail()
```

Out[11]:

	sepal_length	sepal_width	petal_length	petal_width	species
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

In [12]:
```
df.describe()
```

Out[12]:

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

In [13]:
```
df.shape
```

Out[13]:
```
(150, 5)
```

In [14]:
```
df.size
```

Out[14]:
```
750
```

In [15]:
```
df.ndim
```

Out[15]:
```
2
```

In [16]:
```
df.isnull()
```

Out[16]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
...	...	...	...	...	...
145	False	False	False	False	False
146	False	False	False	False	False
147	False	False	False	False	False
148	False	False	False	False	False
149	False	False	False	False	False

150 rows × 5 columns

In [17]:
```
df.isnull
```

Out[17]:
```
<bound method DataFrame.isnull of      sepal_length  sepal_width  petal_length  petal_wi
dth    species
0             5.1          3.5           1.4          0.2     setosa
1             4.9          3.0           1.4          0.2     setosa
2             4.7          3.2           1.3          0.2     setosa
3             4.6          3.1           1.5          0.2     setosa
4             5.0          3.6           1.4          0.2     setosa
..            ...          ...           ...          ...        ...
145           6.7          3.0           5.2          2.3  virginica
146           6.3          2.5           5.0          1.9  virginica
147           6.5          3.0           5.2          2.0  virginica
148           6.2          3.4           5.4          2.3  virginica
149           5.9          3.0           5.1          1.8  virginica

[150 rows x 5 columns]>
```

In [18]:
```
df.isnull().sum()
```

Out[18]:
```
sepal_length    0
sepal_width     0
petal_length    0
petal_width     0
species         0
dtype: int64
```

In [19]:
```
x = np.arange(1,25).reshape(12,2)
y = np.array([0,1,1,0,1,0,0,1,1,0,1,0])
```

In [20]:
```
x
```

Out[20]:
```
array([[ 1,  2],
       [ 3,  4],
       [ 5,  6],
       [ 7,  8],
       [ 9, 10],
       [11, 12],
       [13, 14],
       [15, 16],
       [17, 18],
       [19, 20],
       [21, 22],
       [23, 24]])
```

In [21]:
```
y
```

Out[21]:
```
array([0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0])
```

In [22]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y) #test_size=.3, random_state=42
```

In [23]:
```
y_train
```

Out[23]:
```
array([1, 1, 0, 1, 0, 1, 0, 0, 0])
```

In [24]:
```
y_test
```

Out[24]:
```
array([1, 1, 0])
```

In [25]:
```
x_train
```

Out[25]:
```
array([[21, 22],
       [ 5,  6],
       [ 7,  8],
       [17, 18],
       [19, 20],
       [ 3,  4],
       [13, 14],
       [23, 24],
       [ 1,  2]])
```

In [26]:
```
x_test
```

Out[26]:
```
array([[ 9, 10],
       [15, 16],
       [11, 12]])
```

In [27]:
```
from sklearn.linear_model import LinearRegression
model = LinearRegression().fit(x_train,y_train)
model.score(x_train,y_train)
```

Out[27]:
```
0.0048633440514470605
```

In [ ]:
```

```