

To perform simple linear regression and find out the coefficient of it.

```
In [ ]: #Name: Siddhi N. Sakharkar
        #Roll no.: 51
        #Sec:B
```

```
In [ ]: 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 [28]: os.getcwd()
```

```
Out[28]: 'C:\\Users\\lenovo\\Desktop'
```

```
In [29]: os.chdir('C:\\Users\\lenovo\\Desktop')
```

```
In [30]: df=pd.read_csv('Iris.csv')
```

```
In [31]: df.head()
```

```
Out[31]:
```

| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|---|----|---------------|--------------|---------------|--------------|-------------|
| 0 | 1 | 5.1 | 3.5 | 1.4 | 0.2 | Iris-setosa |
| 1 | 2 | 4.9 | 3.0 | 1.4 | 0.2 | Iris-setosa |
| 2 | 3 | 4.7 | 3.2 | 1.3 | 0.2 | Iris-setosa |
| 3 | 4 | 4.6 | 3.1 | 1.5 | 0.2 | Iris-setosa |
| 4 | 5 | 5.0 | 3.6 | 1.4 | 0.2 | Iris-setosa |

```
In [32]: df.tail()
```

```
Out[32]:
```

| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm | Species |
|-----|-----|---------------|--------------|---------------|--------------|----------------|
| 145 | 146 | 6.7 | 3.0 | 5.2 | 2.3 | Iris-virginica |
| 146 | 147 | 6.3 | 2.5 | 5.0 | 1.9 | Iris-virginica |
| 147 | 148 | 6.5 | 3.0 | 5.2 | 2.0 | Iris-virginica |
| 148 | 149 | 6.2 | 3.4 | 5.4 | 2.3 | Iris-virginica |
| 149 | 150 | 5.9 | 3.0 | 5.1 | 1.8 | Iris-virginica |

```
In [33]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype  
---  -
 0   Id              150 non-null   int64  
 1   SepalLengthCm   150 non-null   float64
 2   SepalWidthCm    150 non-null   float64
 3   PetalLengthCm   150 non-null   float64
 4   PetalWidthCm    150 non-null   float64
 5   Species         150 non-null   object  
dtypes: float64(4), int64(1), object(1)
memory usage: 7.2+ KB
```

```
In [34]: df.describe()
```

Out[34]:

| | Id | SepalLengthCm | SepalWidthCm | PetalLengthCm | PetalWidthCm |
|-------|------------|---------------|--------------|---------------|--------------|
| count | 150.000000 | 150.000000 | 150.000000 | 150.000000 | 150.000000 |
| mean | 75.500000 | 5.843333 | 3.054000 | 3.758667 | 1.198667 |
| std | 43.445368 | 0.828066 | 0.433594 | 1.764420 | 0.763161 |
| min | 1.000000 | 4.300000 | 2.000000 | 1.000000 | 0.100000 |
| 25% | 38.250000 | 5.100000 | 2.800000 | 1.600000 | 0.300000 |
| 50% | 75.500000 | 5.800000 | 3.000000 | 4.350000 | 1.300000 |
| 75% | 112.750000 | 6.400000 | 3.300000 | 5.100000 | 1.800000 |
| max | 150.000000 | 7.900000 | 4.400000 | 6.900000 | 2.500000 |

In [35]:

```
df.isna().sum()
```

Out[35]:

```
Id                0
SepalLengthCm    0
SepalWidthCm     0
PetalLengthCm    0
PetalWidthCm     0
Species          0
dtype: int64
```

In [43]:

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

In [44]:

```
x
```

Out[44]:

```
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 [45]:

```
y
```

Out[45]:

```
array([0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0])
```

In [46]:

```
x_train, x_test, y_train, y_test = train_test_split(x,y)
```

In [47]:

```
y_train
```

Out[47]:

```
array([1, 0, 0, 1, 0, 1, 0, 1, 1])
```

In [48]:

```
y_test
```

Out[48]:

```
array([0, 1, 0])
```

In [49]:

```
x_train
```

Out[49]:

```
array([[17, 18],
       [ 7,  8],
       [13, 14],
       [ 9, 10],
       [ 1,  2],
       [ 5,  6],
       [19, 20],
```

```
[21, 22],  
[15, 16]])
```

```
In [50]: x_test
```

```
Out[50]: array([[11, 12],  
               [ 3,  4],  
               [23, 24]])
```

```
In [51]: from sklearn.linear_model import LinearRegression  
model = LinearRegression().fit(x_train,y_train)  
model.score(x_train, y_train)
```

```
Out[51]: 0.06963855421686738
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js