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

In [2]:
```
df=pd.read_csv(r"C:\Users\admin\Downloads\22nd, 23rd- slr (1)\22nd, 23rd- slr\SIMPL
```

Mean

In [3]:
```
df.mean()
```

Out[3]:
```
YearsExperience        5.313333
Salary             76003.000000
dtype: float64
```

In [4]:
```
df['Salary'].mean()
```

Out[4]:
```
np.float64(76003.0)
```

Median

In [5]:
```
df.median()
```

Out[5]:
```
YearsExperience        4.7
Salary             65237.0
dtype: float64
```

In [6]:
```
df['Salary'].median()
```

Out[6]:
```
np.float64(65237.0)
```

Mode

In [7]:
```
df['Salary'].mode()
```

Out[7]:
```
0      37731
1      39343
2      39891
3      43525
4      46205
5      54445
6      55794
7      56642
8      56957
9      57081
10     57189
11     60150
12     61111
13     63218
14     64445
15     66029
16     67938
17     81363
18     83088
19     91738
20     93940
21     98273
22    101302
23    105582
24    109431
25    112635
26    113812
27    116969
28    121872
29    122391
Name: Salary, dtype: int64
```

Variance

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

Out[8]:
```
YearsExperience    8.053609e+00
Salary             7.515510e+08
dtype: float64
```

In [9]:
```
df['Salary'].var()
```

Out[9]:
```
np.float64(751550960.4137931)
```

Standard Deviation

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

Out[10]:
```
YearsExperience        2.837888
Salary             27414.429785
dtype: float64
```

In [11]:
```
df['Salary'].std()
```

Out[11]:
```
np.float64(27414.4297845823)
```

Coefficient of Variation(CV)

In [12]:
```
from scipy.stats import variation

variation(df.values)
```

Out[12]:
```
array([0.5251297 , 0.35463929])
```

In [13]:
```
variation(df['Salary'])
```

Out[13]:
```
np.float64(0.3546392938275572)
```

Correlation

In [14]:
```
df.corr()
```

Out[14]:

	YearsExperience	Salary
YearsExperience	1.000000	0.978242
Salary	0.978242	1.000000

Skewness

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

Out[16]:
```
YearsExperience    0.37956
Salary             0.35412
dtype: float64
```

In [17]:
```
df['Salary'].skew()
```

Out[17]:
```
np.float64(0.35411967922959153)
```

Standard Error

In [18]:
```
df.sem()
```

Out[18]:
```
YearsExperience       0.518125
Salary             5005.167198
dtype: float64
```

In [19]:
```
 df['Salary'].sem()
```

Out[19]: np.float64(5005.167198052405)

Z-Score

In [20]:
```
import scipy.stats as stats
df.apply(stats.zscore)
```

Out[20]:

	YearsExperience	Salary
0	-1.510053	-1.360113
1	-1.438373	-1.105527
2	-1.366693	-1.419919
3	-1.187494	-1.204957
4	-1.115814	-1.339781
5	-0.864935	-0.718307
6	-0.829096	-0.588158
7	-0.757416	-0.799817
8	-0.757416	-0.428810
9	-0.578216	-0.698013
10	-0.506537	-0.474333
11	-0.470697	-0.749769
12	-0.470697	-0.706620
13	-0.434857	-0.702020
14	-0.291498	-0.552504
15	-0.148138	-0.299217
16	-0.076458	-0.370043
17	-0.004779	0.262859
18	0.210261	0.198860
19	0.246100	0.665476
20	0.532819	0.583780
21	0.640339	0.826233
22	0.927058	0.938611
23	1.034577	1.402741
24	1.213777	1.240203
25	1.321296	1.097402
26	1.500496	1.519868
27	1.536336	1.359074
28	1.787215	1.721028
29	1.858894	1.701773

In [21]:
```
stats.zscore(df['Salary'])
```

Out[21]:
```
array([-1.36011263, -1.10552744, -1.419919  , -1.20495739, -1.33978143,
       -0.71830716, -0.58815781, -0.79981746, -0.42881019, -0.69801306,
       -0.47433279, -0.74976858, -0.70662043, -0.70201994, -0.55250402,
       -0.29921736, -0.37004264,  0.26285865,  0.19885989,  0.66547573,
        0.58377993,  0.82623317,  0.93861127,  1.40274136,  1.24020308,
        1.09740238,  1.51986835,  1.3590738 ,  1.72102849,  1.70177321])
```

Degree of Freedom

In [22]:
```
a = df.shape[0] # this will gives us no.of rows
b = df.shape[1] # this will give us no.of columns
degree_of_freedom = a-b
print(degree_of_freedom)
```

```
28
```

Sum of Squares Regression (SSR)

In [23]:
```
#First we have to separate dependent and independent variables
X=df.iloc[:,:-1].values #independent variable
y=df.iloc[:,1].values
# dependent variable
y_mean = np.mean(y) # this will calculate mean of dependent variable
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.20,random_state=0)

from sklearn.linear_model import LinearRegression
reg =  LinearRegression()
reg.fit(X_train,y_train)

y_predict = reg.predict(X_test) # before doing this we have to train,test and split
SSR = np.sum((y_predict-y_mean)**2)
print(SSR)
```

```
6263152884.28413
```

Sum of Squares Error (SSE)

In [24]:
```
#First we have to separate dependent and independent variables
X=df.iloc[:,:-1].values #independent variable
y=df.iloc[:,1].values
# dependent variable

from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.20,random_state=0)
from sklearn.linear_model import LinearRegression
reg =  LinearRegression()
reg.fit(X_train,y_train)
y_predict = reg.predict(X_test) # before doing this we have to train,test and split
y = y[0:6]
```

```
SSE = np.sum((y-y_predict)**2)
print(SSE)
```

```
15274062883.943203
```

In [ ]:
```
## Sum of Squares Total (SST)
```

In [ ]:
```

```

In [ ]:
```

```