In [141…

```
# importing the libraries
import numpy as np
import pandas as pd
```

In [143…

```
# loading the csv file
dataset=pd.read_csv(r"C:\Users\user\Documents\Salary_Data.csv")
```

In [145…

```
dataset.head()
```

Out[145…

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

Mean

In [148…

```
dataset.mean() # this gives the entire dataframe mean
```

Out[148…

```
YearsExperience        5.313333
Salary             76003.000000
dtype: float64
```

In [150…

```
dataset['Salary'].mean() # this gives that particular column mean
```

Out[150…

```
76003.0
```

Meadian

In [153…

```
dataset.median() # this gives the entire dataframe median
```

Out[153…

```
YearsExperience        4.7
Salary             65237.0
dtype: float64
```

In [155…

```
dataset['Salary'].median() # this gives that particular column median
```

Out[155…

```
65237.0
```

Mode

In [158…

```
dataset.mode() # this gives the mode of that entire dataframe
```

Out[158…

	YearsExperience	Salary
0	3.2	37731
1	4.0	39343
2	NaN	39891
3	NaN	43525
4	NaN	46205
5	NaN	54445
6	NaN	55794
7	NaN	56642
8	NaN	56957
9	NaN	57081
10	NaN	57189
11	NaN	60150
12	NaN	61111
13	NaN	63218
14	NaN	64445
15	NaN	66029
16	NaN	67938
17	NaN	81363
18	NaN	83088
19	NaN	91738
20	NaN	93940
21	NaN	98273
22	NaN	101302
23	NaN	105582
24	NaN	109431
25	NaN	112635
26	NaN	113812
27	NaN	116969
28	NaN	121872
29	NaN	122391

In [160…

```
dataset['Salary'].mode() # gives mode of that particular column
```

```
Out[160…  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 [163…  dataset.var()
```

```
Out[163…  YearsExperience    8.053609e+00
          Salary             7.515510e+08
          dtype: float64
```

```
In [165…  dataset['Salary'].var()
```

```
Out[165…  751550960.4137931
```

Standard Deviation

```
In [168…  dataset.std()
```

```
Out[168…  YearsExperience        2.837888
          Salary             27414.429785
          dtype: float64
```

```
In [170…  dataset['Salary'].std()
```

```
Out[170…  27414.4297845823
```

Coefficient of Variance

In [173…

```
# for calculating coefficient of variance we need to import the library first
from scipy.stats import variation
variation(dataset.values)
```

Out[173…

```
array([0.5251297 , 0.35463929])
```

In [175…

```
variation(dataset['Salary']) # gives coefficient of variance for that particular
```

Out[175…

```
0.3546392938275572
```

Correlation

In [178…

```
dataset.corr() # gives correlation of entire dataframe
```

Out[178…

	YearsExperience	Salary
YearsExperience	1.000000	0.978242
Salary	0.978242	1.000000

In [180…

```
dataset['Salary'].corr(dataset['YearsExperience']) # gives correlation of the pa
```

Out[180…

```
0.9782416184887598
```

Skewness

In [183…

```
dataset.skew()
```

Out[183…

```
YearsExperience    0.37956
Salary             0.35412
dtype: float64
```

In [185…

```
dataset['Salary'].skew()
```

Out[185…

```
0.35411967922959153
```

Standard error

In [188…

```
dataset.sem() # gives standard error of the entire dataset
```

Out[188…

```
YearsExperience       0.518125
Salary             5005.167198
dtype: float64
```

In [190…

```
dataset['Salary'].sem() # gives the standard error of the particular column
```

Out[190…

```
5005.167198052405
```

Zscore

In [193…

```
# for calculating the zscore we need to import the library
import scipy.stats as stats
```

```
dataset.apply(stats.zscore)
```

Out[193…

	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 [195…

```
stats.zscore(dataset['Salary'])
```

Out[195…

```
0    -1.360113
1    -1.105527
2    -1.419919
3    -1.204957
4    -1.339781
5    -0.718307
6    -0.588158
7    -0.799817
8    -0.428810
9    -0.698013
10   -0.474333
11   -0.749769
12   -0.706620
13   -0.702020
14   -0.552504
15   -0.299217
16   -0.370043
17    0.262859
18    0.198860
19    0.665476
20    0.583780
21    0.826233
22    0.938611
23    1.402741
24    1.240203
25    1.097402
26    1.519868
27    1.359074
28    1.721028
29    1.701773
Name: Salary, dtype: float64
```

Degree of Freedom

In [198…

```
a=dataset.shape[0]
b=dataset.shape[1]

degree_of_freedom=a-b
print(degree_of_freedom)
```

```
28
```

Sum Of Squares Of Regression(SSR)

In [206…

```
# first we have to seperate the data into dependent and independent variables
x=dataset.iloc[:,:-1].values # independent variable
y=dataset.iloc[:,1].values # dependen variables
y_mean=np.mean(y)
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) # predicting the model
y=y[0:6]
SSR=np.sum((y_predict-y_mean)**2)
print(SSR)
```

```
6263152884.284127
```

Sum Of Squares Of Error(SSE)

In [220…

```
# first we have to seperate the data into dependent and independent variables
x=dataset.iloc[:,:-1].values # independent variable
y=dataset.iloc[:,1].values # dependen variables
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) # predicting the model
y=y[0:6]
SSE=np.sum((y-y_predict)**2)
print(SSE)
```

```
15274062883.9432
```

Sum Of Squares Total

In [225…

```
mean_total=np.mean(dataset.values)
SST=np.sum((dataset.values-mean_total)**2)
print(SST)
```

```
108429703765.82735
```

R-Square

In [228…

```
r_square=SSR/SST
r_square
```

Out[228…

```
0.05776233510524465
```

In [ ]: