Statstics Code Sheet With example

In [168…

```
import pandas as pd
import numpy as np
```

In [170…

```
salary=pd.read_csv(r"E:\Data Science & AI\Dataset files\Salary_Data.csv")
```

In [172…

```
salary.head()
```

Out[172…

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

Mean

In [175…

```
salary.mean()
```

Out[175…

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

In [177…

```
salary['Salary'].mean() # this will give us mean of that particular column
```

Out[177…

```
76003.0
```

Median

In [191…

```
salary.median()
```

Out[191…

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

In [193…

```
salary['Salary'].median() # this will give us median of that particular column
```

Out[193…

```
65237.0
```

Mode

In [196…

```
salary.mode()
```

Out[196…

	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 [198…

```
salary['Salary'].mode() # this will give us mode of that particular column
```

Out[198…

```
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 [201…

```
salary.var()
```

Out[201…

```
YearsExperience    8.053609e+00
Salary             7.515510e+08
dtype: float64
```

In [203…

```
salary['Salary'].var() # this will give us variance of that particular column4
```

Out[203…

```
751550960.4137931
```

Standard Deviation

In [206…

```
salary.std()
```

Out[206…

```
YearsExperience        2.837888
Salary             27414.429785
dtype: float64
```

In [208…

```
salary['Salary'].std() # this will give us standard deviation of that particular
```

Out[208…

```
27414.4297845823
```

Coefficient of variation(cv)

In [218…

```
# for calculating cv we have to import a library first
from scipy.stats import variation
variation(salary.values) # this will give cv of entire dataframe
```

Out[218…

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

In [220…

```
variation(salary['Salary']) # this will give us CV of that particular column
```

Out[220…

```
0.3546392938275572
```

Correlation

In [223…

```
salary.corr()
```

Out[223…

	YearsExperience	Salary
YearsExperience	1.000000	0.978242
Salary	0.978242	1.000000

In [231…

```
salary['Salary'].corr(salary['YearsExperience']) # this will give us correlation
```

Out[231…

```
0.9782416184887598
```

Skewness

In [234…

```
salary.skew() # this will give skewness of entire dataframe
```

Out[234…

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

In [236…

```
salary['Salary'].skew() # this will give us skewness of that particular column
```

Out[236…

```
0.35411967922959153
```

Standard Error

In [254…

```
salary.sem()
```

Out[254…

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

In [256…

```
salary['Salary'].sem() # this will give us standard error of that particular col
```

Out[256…

```
5005.167198052405
```

Z-score

In [259…

```
# for calculating Z-score we have to import a library first
import scipy.stats as stats
```

```
salary.apply(stats.zscore) # this will give Z-score of entire dataframe
```

Out[259…

	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 [261…

```
stats.zscore(salary['Salary']) # this will give us Z-score of that particular co
```

```
Out[261…   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 [264…

```
a = salary.shape[0] # this will gives us no.of rows
b = salary.shape[1] # this will give us no.of columns
degree_of_freedom = a-b
print(degree_of_freedom) # this will give us degree of freedom for entire datase
```

```
28
```

Sum of Squares Regression (SSR)

In [267…

```
#First we have to separate dependent and independent variables
x=salary.iloc[:,:-1].values #independent variable
y=salary.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
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 sp
SSR = np.sum((y_predict-y_mean)**2)
print(SSR)
```

```
6263152884.28413
```

Sum of Squares Error (SSE)

In [272…

```
#First we have to separate dependent and independent variables
x=salary.iloc[:,:-1].values #independent variable
y=salary.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
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 sp
y = y[0:6]
SSE = np.sum((y-y_predict)**2)
print(SSE)
```

```
15274062883.943203
```

Sum of Squares Total (SST)

In [275…

```
mean_total = np.mean(salary.values) # here df.to_numpy()will convert pandas Data
SST = np.sum((salary.values-mean_total)**2)
print(SST)
```

```
108429703765.82735
```

R-Square

In [278…

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

Out[278…

```
0.05776233510524468
```

In [ ]: