- NAME:RAIYAN RIYAZ CHANDLE
- ROLL NO:B15
- PRN:2324000573
- EXPERIMENT NO:4
- TITLE:CORRELATION AND ITS COEFFICIENT

```
import numpy as np
```

```
def Pearson_Correllation(X, Y):
    meanX = np.mean(X)
    meanY = np.mean(Y)

    Numerator = np.sum((X-meanX)*(Y-meanY))

    denominatorX = np.sqrt(np.sum((X-meanX)**2))
    denominatorY = np.sqrt(np.sum((Y-meanY)**2))

    Pearson_Correllation_Coefficient=Numerator/(denominatorX*denominatorY)
    return Pearson_Correllation_Coefficient
```

```
A=np.array([1,2,3,4,5])
B=np.array([5,6,8,11,13])

Pearson_Correllation_Coefficient=Pearson_Correllation(A,B)
print(f"Pearson Correlation Coefficient = {Pearson_Correllation_Coefficient}")
```

```
Pearson Correlation Coefficient = 0.9877569118027771
```

```
from scipy.stats import pearsonr
```

```
scipy_Pearson= pearsonr(A,B)
print(f"Pearson Correlation Coefficient using scipy: {scipy_Pearson}")
```

```
Pearson Correlation Coefficient using scipy: PearsonRResult(statistic=0.9877569118027771, pvalue=0.0016231966955949143)
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
import pandas as pd
```

```
df = pd.read_csv('/content/drive/MyDrive/DAVL LAB PROGRAMS/LAB_4_14_FEB/Data.csv')
```

```
df.head(10)
```

	mpg	weight
0	18.0	3504
1	15.0	3693
2	18.0	3436
3	16.0	3433
4	17.0	3449
5	15.0	4341
6	14.0	4354
7	14.0	4312
8	14.0	4425
9	15.0	3850

```
variableX=df['mpg']
variableY=df['weight']
```

```
Data_scipy_Pearson=pearsonr(variableX,variableY)
print(f"Pearson Correlation Coefficient for data user library function: {Data_scipy_Pearson}")
```

```
Pearson Correlation Coefficient for data user library function: PearsonRResult(statistic=-0.8782814536196801, pvalue=1.693957670381
```

```
Data_Manual_Pearson=Pearson_Correllation(variableX,variableY)
print(f"Pearson Correlation Coefficient for data using manual function: {Data_Manual_Pearson}")
```

```
Pearson Correlation Coefficient for data using manual function: -0.8782814536196802
```

```
def Spearman_Rank_Correlation(X, Y):
    rankX=np.argsort(np.argsort(X)) + 1
    rankY=np.argsort(np.argsort(Y)) + 1
    d_square=np.sum((rankX-rankY)**2)
    n=len(X)
    spearman=1-(6 *d_square)/(n*(n**2-1))
    return spearman



X = np.array([1, 2, 3, 4, 5])
Y = np.array([7,8,11,13,12])

spearman=Spearman_Rank_Correlation(X, Y)
print(f"Spearman Rank Correlation Coefficient = {spearman}")
```

```
Spearman Rank Correlation Coefficient = 0.9
```

```
from scipy.stats import spearmanr
```

```
Spearman_Correlation=spearmanr(X, Y)
print(f"Spearman Correlation Coefficient = {Spearman_Correlation}")
```

```
Spearman Correlation Coefficient = SignificanceResult(statistic=0.8999999999999998, pvalue=0.03738607346849874)
```

```
Spearman_for_data_using_scipy= spearmanr(variableX,variableY)
print(f"Spearman Correlation Coefficient using scipy={Spearman_for_data_using_scipy}")
```

```
Spearman Correlation Coefficient using scipy=SignificanceResult(statistic=-0.9140707511066734, pvalue=7.979113456201118e-119)
```

```
Spearman_for_data_using_manualFunc= Spearman_Rank_Correlation(variableX,variableY)
print(f"Spearman Correlation Coefficient using scipy={Spearman_for_data_using_manualFunc}")
```

```
Spearman Correlation Coefficient using scipy=-0.9143608262314025
```

Start coding or generate with AI.