

## Experiment 4

**Aim:** Implementation of Statistical Hypothesis Test using Scipy and Sci-kit learn.

Perform the following correlation tests:

- Pearson's Correlation Coefficient
- Spearman's Rank Correlation
- Kendall's Rank Correlation
- Chi-Squared Test

### Performance:

- Prerequisite: We import necessary libraries such as pandas for data manipulation, numpy for numerical operations, scipy.stats for statistical calculations, and seaborn and matplotlib.pyplot for visualization and load data into Pandas. To understand the dataset structure, we print its basic information using `df.info()` to check column types (numerical or categorical) and `df.head()` to preview the first few rows:

Command: import pandas as pd

import numpy as np

import scipy.stats as stats

import seaborn as sns

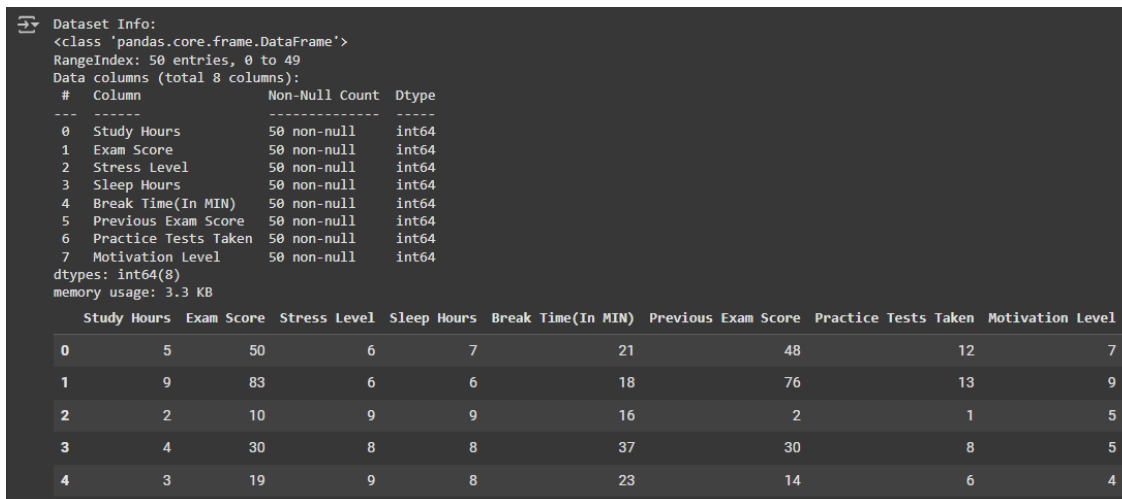
import matplotlib.pyplot as plt

df = pd.read\_csv('Student Performance Analysis.csv')

print("Dataset Info:")

df.info()

df.head()



Dataset Info:

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 50 entries, 0 to 49  
Data columns (total 8 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   Study Hours            50 non-null    int64  
1   Exam Score             50 non-null    int64  
2   Stress Level           50 non-null    int64  
3   Sleep Hours            50 non-null    int64  
4   Break Time(In MIN)     50 non-null    int64  
5   Previous Exam Score    50 non-null    int64  
6   Practice Tests Taken   50 non-null    int64  
7   Motivation Level       50 non-null    int64  
dtypes: int64(8)  
memory usage: 3.3 KB
```

	Study Hours	Exam Score	Stress Level	Sleep Hours	Break Time(In MIN)	Previous Exam Score	Practice Tests Taken	Motivation Level
0	5	50	6	7	21	48	12	7
1	9	83	6	6	18	76	13	9
2	2	10	9	9	16	2	1	5
3	4	30	8	8	37	30	8	5
4	3	19	9	8	23	14	6	4

- To perform correlation tests on specific features, we manually select two columns (col1 and col2), which should be numerical for Pearson, Spearman, and Kendall correlation. It is then checked whether the selected column names exist in the dataset to prevent errors. This step ensures we are working with the correct variables for our analysis.

```
Command: col1 = 'Study Hours'
```

```
col2 = 'Exam Score'
```

```
if col1 not in df.columns or col2 not in df.columns:
```

```
    raise ValueError("One or both selected columns do not exist in the dataset!")
```

```
print(f"Selected Columns: {col1}, {col2}")
```

```
➡ Selected Columns: Study Hours, Exam Score
```

#### a) Pearson's Correlation Coefficient:

```
Command: pearson_corr, _ = stats.pearsonr(df[col1], df[col2])
```

```
print(f"Pearson Correlation Coefficient between {col1} and {col2}: {pearson_corr:.4f}")
```

```
➡ Pearson Correlation Coefficient between Study Hours and Exam Score: 0.9648
```

Pearson's correlation measures the linear relationship between two continuous variables. We compute it using `stats.pearsonr(df[col1], df[col2])`, which returns a correlation coefficient ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation), with 0 indicating no correlation.

The Pearson correlation of 0.9648 indicates a strong positive linear relationship between Study Hours and Exam Score. This means that as students study more hours, their exam scores tend to increase proportionally, showing a near-perfect linear trend.

#### b) Spearman's Rank Correlation:

```
Command: spearman_corr, _ = stats.spearmanr(df[col1], df[col2])
```

```
print(f"Spearman's Rank Correlation between {col1} and {col2}: {spearman_corr:.4f}")
```

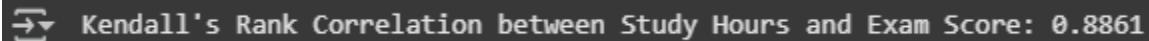
```
➡ Spearman's Rank Correlation between Study Hours and Exam Score: 0.9671
```

Spearman's correlation measures the monotonic relationship between two variables, making it more robust to outliers and non-linear relationships than Pearson's. It is calculated using `stats.spearmanr(df[col1], df[col2])`, which ranks the values before computing the correlation. Like Pearson's, the coefficient ranges from -1 to +1, where higher absolute values indicate stronger relationships. This test is useful when data is not normally distributed.

With a Spearman correlation of 0.9671, there is a strong monotonic relationship between Study Hours and Exam Score. Even if the relationship is not strictly linear, the ranking of students based on study hours aligns closely with their exam performance.

### c) Kendall's Rank Correlation:

```
Command: kendall_corr, _ = stats.kendalltau(df[col1], df[col2])
print(f'Kendall's Rank Correlation between {col1} and {col2}: {kendall_corr:.4f}')
```



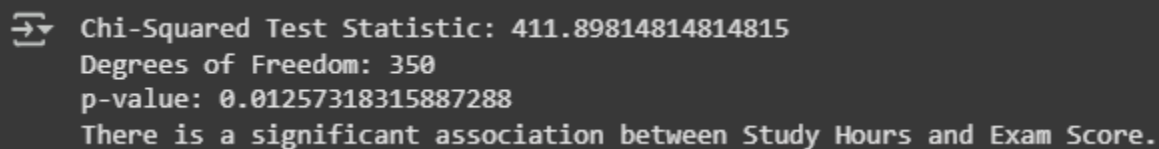
```
→ Kendall's Rank Correlation between Study Hours and Exam Score: 0.8861
```

Kendall's correlation is another non-parametric test that assesses the strength of association between two variables based on the concordance of pairs. It is calculated using `stats.kendalltau(df[col1], df[col2])`, which measures the agreement between ranked data points. Kendall's method is particularly useful for smaller datasets and when working with ordinal data. The correlation coefficient, like the previous tests, ranges between -1 and +1, with values closer to  $\pm 1$  indicating stronger relationships.

The Kendall correlation of 0.8861 suggests a very strong agreement in ranking between Study Hours and Exam Score. Students who study more consistently rank higher in exam scores, reinforcing the predictability of performance based on study time.

### d) Chi-Squared Test:

```
Command: contingency_table = pd.crosstab(df[col1], df[col2])
chi2_stat, p_val, dof, expected = stats.chi2_contingency(contingency_table)
print(f'Chi-Squared Test Statistic: {chi2_stat}')
print(f'Degrees of Freedom: {dof}')
print(f'p-value: {p_val}')
if p_val < 0.05:
    print(f'There is a significant association between {col1} and {col2}.')
else:
    print(f'There is NO significant association between {col1} and {col2}.')
```



```
→ Chi-Squared Test Statistic: 411.89814814814815
Degrees of Freedom: 350
p-value: 0.01257318315887288
There is a significant association between Study Hours and Exam Score.
```

The Chi-Squared test is used to examine the association between two categorical variables by creating a contingency table using `pd.crosstab(df[col1], df[col2])`. The test statistic, degrees of freedom, and p-value are computed using `stats.chi2_contingency(contingency_table)`. If the p-value is less than 0.05, we reject the null hypothesis, indicating a significant association between the two variables. Otherwise, there is no significant relationship. This test is particularly useful in analyzing dependencies between categorical-like numerical data, such as grouped scores or study hour categories.

The Chi-Squared test result shows a statistically significant association ( $p < 0.05$ ) between Study Hours and Exam Score. This means that the two variables are not independent, and study time likely influences exam performance.

## Conclusion:

1. In this experiment, we learned about the implementation of Statistical Hypothesis Test using Scipy and Sci-kit learn.
2. The Pearson correlation coefficient (0.9648) indicates a strong positive linear relationship between Study Hours and Exam Score, meaning that as students study more, their exam scores tend to increase proportionally.
3. The Spearman's rank correlation (0.9671) suggests that the ranking of students based on study hours closely matches their ranking in exam scores, even if the relationship is not perfectly linear.
4. The Kendall's rank correlation (0.8861) confirms a strong agreement in ranking, showing that students who study more are highly likely to rank higher in exam performance.
5. The Chi-Squared test result ( $\chi^2 = 411.90$ ,  $p = 0.0126$ ) indicates a statistically significant association between Study Hours and Exam Score, proving that study time plays an important role in influencing exam performance.
6. Overall, all tests confirm a strong positive relationship between study hours and exam scores, suggesting that increasing study time is likely to improve exam performance