

Name: Kshitij Hundre  
Div:D15C  
Roll No:18

## Exp 4 : Statistical Hypothesis Testing Using SciPy and Scikit-Learn

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

**Problem Statement:** Perform the following Tests:Correlation Tests:

- a) Pearson's Correlation Coefficient
- b) Spearman's Rank Correlation
- c) Kendall's Rank Correlation
- d) Chi-Squared Test

### Introduction to Hypothesis Testing

Hypothesis testing is a statistical method used to make inferences about a population based on sample data. It helps in determining whether the observed results are due to chance or if there is a statistically significant relationship between variables.

In this experiment, we will conduct **correlation tests and a chi-squared test** using Python's `scipy.stats` library.

### Theory and Output:

#### 1.Loading dataset:

Data loading is the first step in data analysis. The dataset is stored in a CSV file and read using `pandas.read_csv()`.

The first few rows are displayed to understand the dataset structure

```
import pandas as pd

# Load the dataset
df = pd.read_csv('sc.csv') # suomermarket sales data

# Display first few rows
print(df.head())

# Display column names and data types
print(df.info())

# Summary statistics
print(df.describe())
```

	Invoice ID	Branch	City	Customer type	Gender	\
0	750-67-8428	A	Yangon	Member	Female	
1	226-31-3081	C	Naypyitaw	Normal	Female	
2	631-41-3108	A	Yangon	Normal	Male	
3	123-19-1176	A	Yangon	Member	Male	
4	373-73-7910	A	Yangon	Normal	Male	

  

	Product line	Unit price	Quantity	Tax 5%	Total	Date	\
0	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	
1	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	
2	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	
3	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	
4	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	

  

	Time	Payment	cogs	gross margin	percentage	gross income	Rating
0	13:08	Ewallet	522.83	4.761905		26.1415	9.1
1	10:29	Cash	76.40	4.761905		3.8200	9.6
2	13:23	Credit card	324.31	4.761905		16.2155	7.4
3	20:33	Ewallet	465.76	4.761905		23.2880	8.4
4	10:37	Ewallet	604.17	4.761905		30.2085	5.3

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Invoice ID           1000 non-null  object
1   Branch              1000 non-null  object
2   City                1000 non-null  object
```

## 2. Pearson's Correlation Coefficient:

Pearson's Correlation Coefficient (denoted as  $r$ ) measures the **linear** relationship between two continuous variables.

Values range from **-1 to +1**:

- **+1**: Perfect positive correlation
- **0**: No correlation
- **-1**: Perfect negative correlation

The formula for Pearson's Correlation Coefficient is:

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \sum (Y_i - \bar{Y})^2}}$$

```
#pearson correlation between quantity and total
from scipy.stats import pearsonr

corr, p_value = pearsonr(df['Total'], df['Quantity'])
print(f"Pearson Correlation Coefficient: {corr:.4f}")
print(f"P-value: {p_value:.4f}")
# this correlation is significant..as p < 0.005.....for pearson correlation strong +ve = 1...strong -ve = -1.....no correlation = 0
```

➔ Pearson Correlation Coefficient: 0.7055  
P-value: 0.0000

## 3. Spearman's Rank Correlation

- Spearman's Rank Correlation (denoted as  $\rho$ , rho) measures the monotonic relationship between two variables.
- It does not require normally distributed data.
- If ranks of two variables are related, it indicates correlation.
- The formula is:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

```
from scipy.stats import spearmanr

corr, p_value = spearmanr(df['Customer type'], df['Rating'])
print(f"Spearman Correlation Coefficient: {corr:.4f}")
print(f"P-value: {p_value:.4f}")
```

➔ Spearman Correlation Coefficient: 0.0187  
P-value: 0.5552

#### 4. Kendall's Rank Correlation

##### Theory:

- Kendall's Tau ( $\tau$ ) measures the **ordinal association** between two variables.
- It counts **concordant** and **discordant** pairs:
  - **Concordant pairs**: If one variable increases, the other also increases.
  - **Discordant pairs**: One increases while the other decreases.
- The formula is:

$$\tau = \frac{(C - D)}{\frac{1}{2}n(n - 1)}$$

```
from scipy.stats import kendalltau

corr, p_value = kendalltau(df['Gender'], df['Payment'])
print(f"Kendall's Rank Correlation Coefficient: {corr:.4f}")
print(f"P-value: {p_value:.4f}")
```

→ Kendall's Rank Correlation Coefficient: 0.0420  
P-value: 0.1587

#### 5. Chi-Squared Test

- The **Chi-Squared Test** is used for **categorical data** to check if two variables are independent.
- It compares **observed** and **expected** frequencies.
- The formula is:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

```
from scipy.stats import chi2_contingency

# Create a contingency table
contingency_table = pd.crosstab(df['Gender'], df['Product line'])

# Perform Chi-Squared test
chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)
print(f"Chi-Squared Statistic: {chi2_stat:.4f}")
print(f"P-value: {p_value:.4f}")
print(f"Degrees of Freedom: {dof}")

#p-value ≥ 0.05 → No significant relationship.
```

Chi-Squared Statistic: 5.7445  
P-value: 0.3319  
Degrees of Freedom: 5

## Conclusion

1. **Pearson's Correlation:** Measures **linear relationship** between numerical variables. If  $p < 0.05$ , the correlation is significant.
2. **Spearman's Correlation:** Checks for **monotonic relationship**. If  $p < 0.05$ , variables move together in a ranked order.
3. **Kendall's Correlation:** Identifies **ordinal association**. A small **p-value** means a strong relationship.
4. **Chi-Square Test:** Determines **independence of categorical variables**. If  $p < 0.05$ , variables are dependent; otherwise, they are independent.

## Final Summary:

- If  $p < 0.05$ , the test indicates a significant relationship.
- If  $p > 0.05$ , no strong relationship exists.

These tests help understand **associations** in the dataset for data-driven decisions.