Name - Gaurang A Raorane Div - D15A Roll no - 49 Batch - C

**Experiment - 4**

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

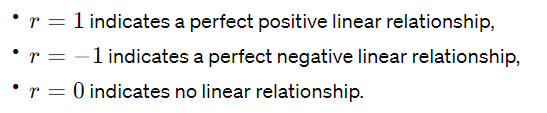
**Theory :-**

Problem statement - Perform the following Tests:

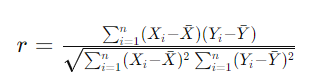
Correlation Tests:

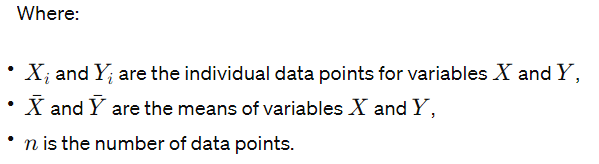
1. Pearson’s Correlation Coefficient:

Pearson's correlation coefficient, denoted as *r*, measures the linear relationship between two continuous variables. It quantifies the strength and direction of the linear association between two variables. The coefficient ranges from -1 to 1, where:

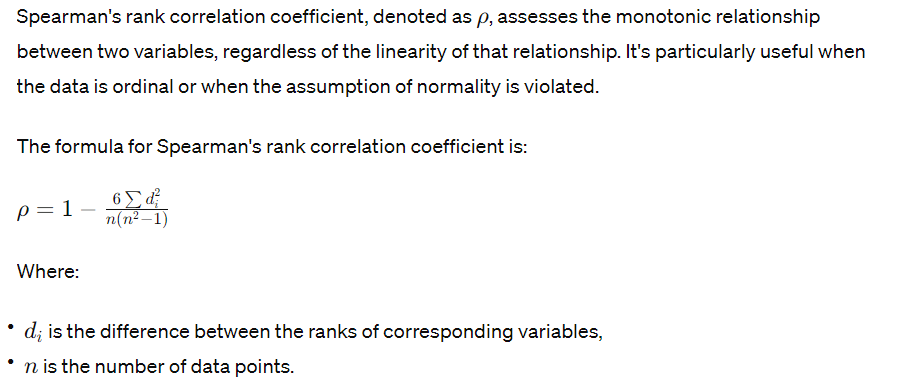


The formula for Pearson's correlation coefficient between variables *X* and *Y* is:

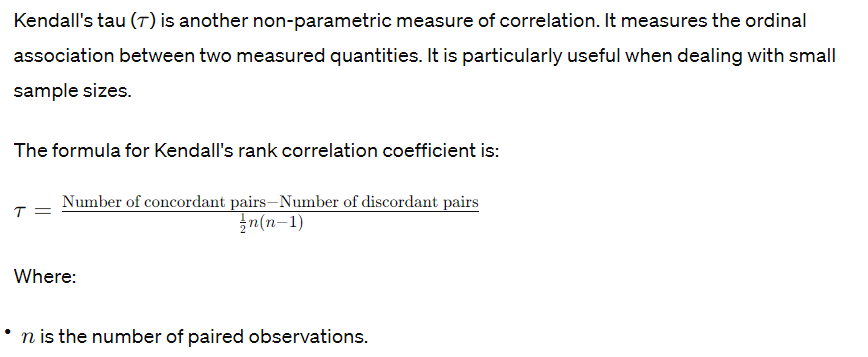




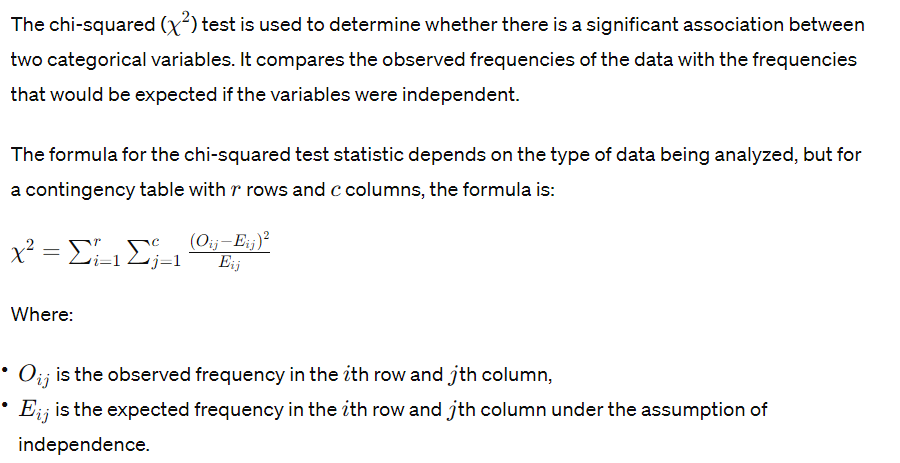
2. Spearman’s Rank Correlation

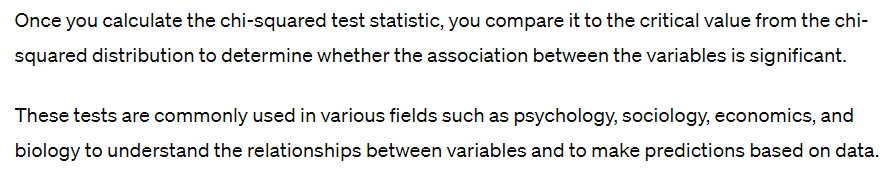


3. Kendall’s Rank Correlation:



4. Chi-Squared Test:





### **1. SciPy:**

Introduction:

SciPy is a Python library that builds upon the NumPy array object. It provides many user-friendly and efficient numerical routines for scientific computing. It includes modules for optimization, integration, interpolation, Fourier transforms, signal processing, linear algebra, and more.

Key Modules and Functionalities:

* Optimization: SciPy offers optimization routines like minimize() for minimizing functions.
* Integration: Integration routines like quad() for numerical integration are available.
* Interpolation: Interpolation functions such as interp1d() are useful for interpolating between data points.
* Linear Algebra: Linear algebra routines include matrix operations, solving linear equations, eigenvalue problems, etc.
* Statistics: Statistical functions such as probability distributions, statistical tests, and descriptive statistics are provided.
* Signal Processing: Functions for signal processing, such as filtering, window functions, spectral analysis, etc., are included.

Usage:

SciPy is extensively used in scientific and engineering applications for tasks ranging from signal processing to optimization. Its comprehensive functionality and integration with other scientific libraries make it a valuable tool for numerical computation.

### **2. scikit-learn:**

Introduction:

scikit-learn is a machine learning library in Python that provides simple and efficient tools for data mining and data analysis. It is built on NumPy, SciPy, and matplotlib. scikit-learn offers various algorithms and tools for classification, regression, clustering, dimensionality reduction, model selection, and preprocessing.

Key Features:

* Consistent Interface: scikit-learn provides a consistent interface across various algorithms, making it easy to experiment with different methods.
* Supervised Learning: It includes algorithms for supervised learning tasks such as classification and regression, including SVMs, decision trees, random forests, etc.
* Unsupervised Learning: For unsupervised tasks like clustering and dimensionality reduction, scikit-learn offers algorithms like K-means clustering, PCA, t-SNE, etc.
* Model Evaluation: It provides tools for model selection and evaluation, including cross-validation, grid search, and performance metrics.
* Data Preprocessing: scikit-learn offers utilities for data preprocessing, such as feature scaling, normalization, and imputation of missing values.

Usage:

scikit-learn is widely used in academia and industry for machine learning tasks, including but not limited to:

* Predictive modeling
* Text mining
* Image recognition
* Bioinformatics
* Recommender systems

**Input :-**

import pandas as pd

import numpy as np

from scipy.stats import pearsonr, spearmanr, kendalltau, chi2\_contingency

from sklearn.datasets import make\_classification

# Generate sample DataFrame

# Replace this with your actual DataFrame or load data from a file

# For demonstration, I'll create a random DataFrame

np.random.seed(0)

data = make\_classification(n\_samples=100, n\_features=2, n\_informative=2, n\_redundant=0, random\_state=0)

df = pd.DataFrame(data[0], columns=['X', 'Y'])

# Display DataFrame info

print(df.info())

# 1. Pearson’s Correlation Coefficient

pearson\_corr, pearson\_p\_value = pearsonr(df['X'], df['Y'])

print("Pearson's correlation coefficient:", pearson\_corr)

print("p-value:", pearson\_p\_value)

# 2. Spearman’s Rank Correlation

spearman\_corr, spearman\_p\_value = spearmanr(df['X'], df['Y'])

print("Spearman's rank correlation coefficient:", spearman\_corr)

print("p-value:", spearman\_p\_value)

# 3. Kendall’s Rank Correlation

kendall\_corr, kendall\_p\_value = kendalltau(df['X'], df['Y'])

print("Kendall's rank correlation coefficient:", kendall\_corr)

print("p-value:", kendall\_p\_value)

# 4. Chi-Squared Test

# For demonstration, let's create some categorical data

df['Category'] = np.random.choice(['A', 'B', 'C'], size=len(df))

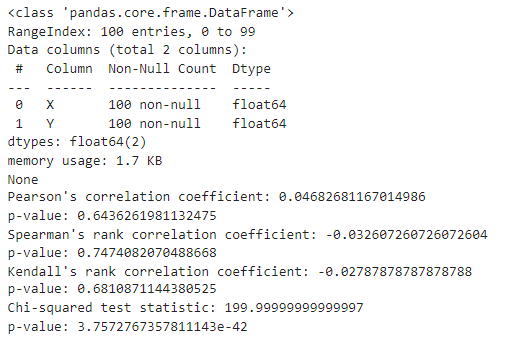
contingency\_table = pd.crosstab(df['Category'], df['Category']) # Replace with actual contingency table

chi2, chi2\_p\_value, \_, \_ = chi2\_contingency(contingency\_table)

print("Chi-squared test statistic:", chi2)

print("p-value:", chi2\_p\_value)

**Output:-**

****

**Inference:-**

* **Pearson's Correlation Coefficient:**
  + Ideal Value: The ideal value depends on the context of your data and the relationship you are examining. In general, a value close to 1 or -1 indicates a strong linear relationship, while a value close to 0 indicates no linear relationship.
  + Interpretation: The correlation coefficient of 0.0468 suggests a very weak positive linear relationship.
  + p-value: The p-value (0.6436) indicates the probability of observing such a correlation coefficient if the true correlation in the population is zero. In this case, since the p-value is greater than 0.05 (common significance level), we fail to reject the null hypothesis of no correlation.
* **Spearman's Rank Correlation:**
  + Ideal Value: The ideal value varies similarly to Pearson's correlation coefficient.
  + Interpretation: The correlation coefficient of -0.0326 suggests a very weak negative monotonic relationship.
  + p-value: The p-value (0.7474) is greater than 0.05, indicating no significant correlation.
* **Kendall's Rank Correlation:**
  + Ideal Value: Similar to Spearman's rank correlation, ideal values vary based on the context.
  + Interpretation: The correlation coefficient of -0.0279 suggests a very weak negative rank correlation.
  + p-value: The p-value (0.6811) is greater than 0.05, indicating no significant correlation.
* **Chi-squared Test:**
  + Ideal Value: For the chi-squared test statistic, a smaller value indicates less discrepancy between the observed and expected frequencies, which would be ideal if there is no association between the categorical variables.
  + Interpretation: The chi-squared test statistic of 200 and a very small p-value (3.76e-42) indicate a significant association between the categorical variables.

In summary, based on the provided results:

* There is no significant linear relationship between the variables for Pearson's correlation coefficient, Spearman's rank correlation, and Kendall's rank correlation.
* However, there is a significant association between the categorical variables for the chi-squared test.

**Conclusion:-**

Through this experiment we understood what different types of correlation test are (pearson’s correlation coefficient, spearman’s rank correlation, kendall’s rank correlation,chi-squared test) and implemented them on a dataset to get the statistical hypothesis test.