NAME KAUSHAL S GALAV D15C 11

Experiment - 4

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

DataSet Link : [Diabetes]

Theory :

Correlation and association tests are used in statistics to measure relationships between variables. Pearson's Correlation Coefficient quantifies the linear relationship between two continuous variables, ranging from -1 (strong negative correlation) to +1 (strong positive correlation), with 0 indicating no correlation. It assumes normally distributed data and is sensitive to outliers. In contrast, Spearman's Rank Correlation is a non-parametric test that evaluates the monotonic relationship between two variables by ranking data points. It is useful when the relationship is non-linear and is less affected by extreme values, making it suitable for ordinal or skewed data.

Another non-parametric alternative is Kendall's Rank Correlation, which measures the strength of association based on the concordance of data pairs. It is more robust for small sample sizes and ties in data, providing a more reliable assessment of rank-based dependencies. Unlike Pearson's method, both Spearman and Kendall's tests do not assume normality and work well for ordinal data. These correlation tests help understand variable dependencies in various fields like finance, medicine, and social sciences.

The Chi-Squared Test is used for testing relationships between categorical variables. It evaluates whether observed frequencies significantly differ from expected frequencies under the assumption of independence. This test is widely used in feature selection, independence testing, and market research to determine associations in categorical data. Unlike correlation tests that measure strength and direction, the Chi-Square test assesses whether variables are statistically dependent without indicating the strength of association. These statistical methods collectively play a vital role in data-driven decision-making and hypothesis testing in research.

Output:

1.Importing Required Libraries :

Importing required libraries ensures that all necessary tools for data handling, analysis, visualization, and modeling are available. It enables efficient execution of tasks like data manipulation, statistical analysis, and machine learning.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.feature_selection import chi2
```

2. Loading dataset:

The purpose of loading a dataset is to import data into a Python environment for analysis, preprocessing, and visualization. It serves as the first step in data processing to enable further exploration and model building.

```
file_path = "/content/sample_data/Diabetes.xlsx"

# Load Excel file
df = pd.read_excel(file_path)

# Display the first 5 rows
df.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

3. Exploratory Data Analysis (EDA):

This process helps understand the dataset's structure, identify missing values, and detect data inconsistencies. It provides summary statistics to analyze distributions, central tendencies, and variability before further processing.

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

# Check for missing values
print("\nMissing Values:\n", df.isnull().sum())

# Summary statistics
df.describe()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

1. 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

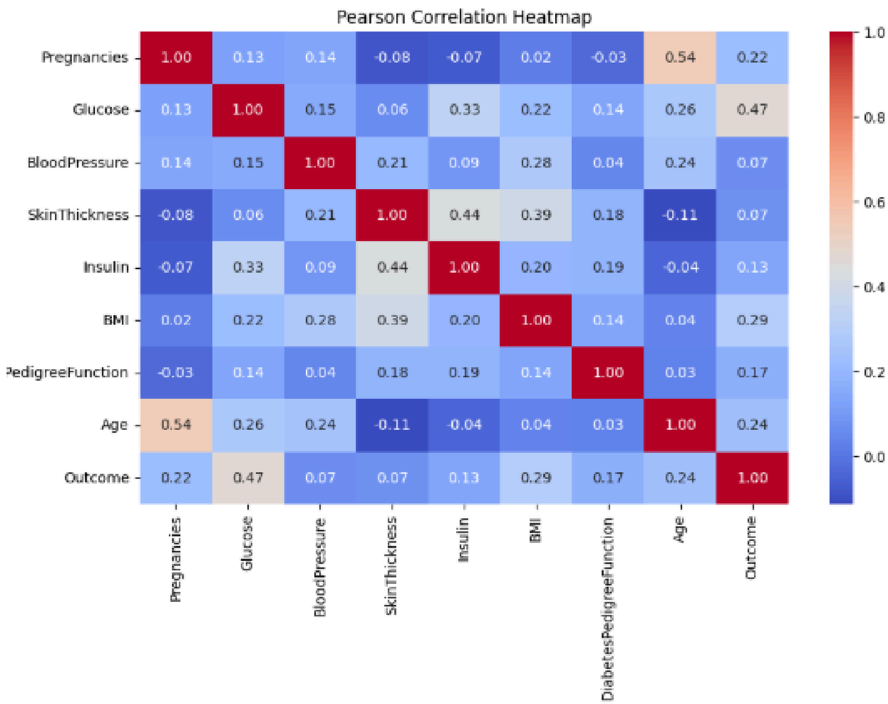
The formula for Pearson's Correlation Coefficient is:

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

```python
#Pearson's Correlation Coefficient
pearson_corr = df.corr(method='pearson')
print("\nPearson Correlation Coefficient:\n", pearson_corr)

# Heatmap of Pearson correlation
plt.figure(figsize=(10, 6))
sns.heatmap(pearson_corr, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Pearson Correlation Heatmap")
plt.show()
```

```
Pearson Correlation Coefficient:
                           Pregnancies   Glucose  BloodPressure  SkinThickness  \
Pregnancies                  1.000000  0.129459       0.141282      -0.081672
Glucose                      0.129459  1.000000       0.152590       0.057328
BloodPressure                0.141282  0.152590       1.000000       0.207371
SkinThickness               -0.081672  0.057328       0.207371       1.000000
Insulin                     -0.073535  0.331357       0.088933       0.436783
BMI                          0.017683  0.221071       0.281805       0.392573
DiabetesPedigreeFunction    -0.033523  0.137337       0.041265       0.183928
Age                          0.544341  0.263514       0.239528      -0.113970
Outcome                      0.221898  0.466581       0.065068       0.074752

                           Insulin       BMI  DiabetesPedigreeFunction  \
Pregnancies              -0.073535  0.017683                 -0.033523
Glucose                   0.331357  0.221071                  0.137337
BloodPressure             0.088933  0.281805                  0.041265
SkinThickness             0.436783  0.392573                  0.183928
Insulin                   1.000000  0.197859                  0.185071
BMI                       0.197859  1.000000                  0.140647
DiabetesPedigreeFunction  0.185071  0.140647                  1.000000
Age                      -0.042163  0.036242                  0.033561
Outcome                   0.130548  0.292695                  0.173844

                               Age   Outcome
Pregnancies               0.544341  0.221898
Glucose                   0.263514  0.466581
BloodPressure             0.239528  0.065068
SkinThickness            -0.113970  0.074752
Insulin                  -0.042163  0.130548
BMI                       0.036242  0.292695
DiabetesPedigreeFunction  0.033561  0.173844
Age                       1.000000  0.238356
Outcome                   0.238356  1.000000
```

Heatmap of Pearson correlation :



2.Spearman's Rank Correlation

- Spearman's Rank Correlation (denoted as ρ, 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:

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

```python
# Spearman's Rank Correlation
spearman_corr = df.corr(method='spearman')
print("\nSpearman's Rank Correlation:\n", spearman_corr)
```

```
Spearman's Rank Correlation:
                          Pregnancies   Glucose  BloodPressure  SkinThickness  \
Pregnancies                  1.000000  0.130734       0.185127      -0.085222
Glucose                      0.130734  1.000000       0.235191       0.060022
BloodPressure                0.185127  0.235191       1.000000       0.126486
SkinThickness               -0.085222  0.060022       0.126486       1.000000
Insulin                     -0.126723  0.213206      -0.006771       0.541000
BMI                          0.000132  0.231141       0.292870       0.443615
DiabetesPedigreeFunction    -0.043242  0.091293       0.030046       0.180390
Age                          0.607216  0.285045       0.350895      -0.066795
Outcome                      0.198689  0.475776       0.142921       0.089728

                           Insulin       BMI  DiabetesPedigreeFunction  \
Pregnancies              -0.126723  0.000132                 -0.043242
Glucose                   0.213206  0.231141                  0.091293
BloodPressure            -0.006771  0.292870                  0.030046
SkinThickness             0.541000  0.443615                  0.180390
Insulin                   1.000000  0.192726                  0.221150
BMI                       0.192726  1.000000                  0.141192
DiabetesPedigreeFunction  0.221150  0.141192                  1.000000
Age                      -0.114213  0.131186                  0.042909
Outcome                   0.066472  0.309707                  0.175353

                               Age   Outcome
Pregnancies               0.607216  0.198689
Glucose                   0.285045  0.475776
BloodPressure             0.350895  0.142921
SkinThickness            -0.066795  0.089728
Insulin                  -0.114213  0.066472
BMI                       0.131186  0.309707
DiabetesPedigreeFunction  0.042909  0.175353
Age                       1.000000  0.309040
Outcome                   0.309040  1.000000
```

3.Kendall’s Rank Correlation

Theory:

- Kendall’s Tau (τ) 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:

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

```
[7] #Kendall’s Rank Correlation
    kendall_corr = df.corr(method='kendall')
    print("\nKendall’s Rank Correlation:\n", kendall_corr)
```

```
Kendall’s Rank Correlation:
                           Pregnancies   Glucose  BloodPressure  SkinThickness  \
Pregnancies                  1.000000  0.091323       0.135440      -0.064401
Glucose                      0.091323  1.000000       0.159961       0.039046
BloodPressure                0.135440  0.159961       1.000000       0.094868
SkinThickness               -0.064401  0.039046       0.094868       1.000000
Insulin                     -0.096417  0.163645      -0.003682       0.420066
BMI                          0.004183  0.155862       0.205222       0.331532
DiabetesPedigreeFunction    -0.029959  0.061871       0.019448       0.126457
Age                          0.458272  0.196510       0.246056      -0.044754
Outcome                      0.170370  0.390565       0.119206       0.076297

                           Insulin       BMI  DiabetesPedigreeFunction  \
Pregnancies              -0.096417  0.004183                 -0.029959
Glucose                   0.163645  0.155862                  0.061871
BloodPressure            -0.003682  0.205222                  0.019448
SkinThickness             0.420066  0.331532                  0.126457
Insulin                   1.000000  0.141587                  0.161652
BMI                       0.141587  1.000000                  0.094644
DiabetesPedigreeFunction  0.161652  0.094644                  1.000000
Age                      -0.080176  0.088678                  0.028042
Outcome                   0.058531  0.253676                  0.143359

                               Age   Outcome
Pregnancies               0.458272  0.170370
Glucose                   0.196510  0.390565
BloodPressure             0.246056  0.119206
SkinThickness            -0.044754  0.076297
Insulin                  -0.080176  0.058531
BMI                       0.088678  0.253676
DiabetesPedigreeFunction  0.028042  0.143359
Age                       1.000000  0.257363
Outcome                   0.257363  1.000000
```

4. 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:

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

```python
# Extract categorical features and target variable
X = df_encoded[categorical_features]
y = df_encoded[target_column]

# Compute Chi-Square test
chi2_stat, p_val = chi2(X, y)

# Display results
for i in range(len(categorical_features)):
    print(f"Feature: {categorical_features[i]}, Chi-Square Stat: {chi2_stat[i]}, p-value: {p_val[i]}")
```

```
Feature: Glucose_Bin, Chi-Square Stat: 22.943251366038417, p-value: 1.6685495815767347e-06
Feature: BMI_Bin, Chi-Square Stat: 3.722269132425522, p-value: 0.05369135831749405
Feature: Age_Bin, Chi-Square Stat: 15.402185620122738, p-value: 8.68877387715916e-05
```

Conclusion

- **Pearson's Correlation:** Measures the **linear** relationship between two numerical variables. A **p-value < 0.05** indicates a statistically significant correlation.
- **Spearman's Correlation:** Evaluates the **monotonic** relationship between variables, considering ranks instead of exact values. A **p-value < 0.05** suggests a significant ranked association.
- **Kendall's Correlation:** Identifies the **ordinal association** between variables. A **small p-value** implies a strong dependency in rank ordering.
- **Chi-Square Test:** Assesses whether **categorical variables** are independent. If **p < 0.05**, they are dependent; otherwise, they are independent.

Final Summary:

If **p < 0.05**, the test suggests a statistically significant relationship between variables.
If **p > 0.05**, no strong relationship exists.
These statistical tests help uncover associations in the dataset, guiding data-driven decision-making.

This refined version keeps the essence of your conclusion while making it more precise and readable.