Name: Neeraj Rijhwani Rollno.64 Div :D15C

Exp 4

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

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

```
[1] import pandas as pd
    import scipy.stats as stats
```

```
df = pd.read_csv('/content/Employee.csv')
```

```
df.head()
```

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	Bachelors	2017	Bangalore	3	34	Male	No	0	0
1	Bachelors	2013	Pune	1	28	Female	No	3	1
2	Bachelors	2014	New Delhi	3	38	Female	No	2	0
3	Masters	2016	Bangalore	3	27	Male	No	5	1
4	Masters	2017	Pune	3	24	Male	Yes	2	1

Next steps: Generate code with df View recommended plots New interactive sheet

Name: Neeraj Rijhwani Rollno.64 Div :D15C

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_corr, pearson_p = stats.pearsonr(df['Age'], df['ExperienceInCurrentDomain'])

print(f"Pearson's Correlation Coefficient: {pearson_corr}")
print(f"P-value: {pearson_p}")
```

```
Pearson's Correlation Coefficient: -0.13464285083693067
P-value: 2.8637816441811323e-20
```

3. 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:

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

```
[13] spearman_corr, spearman_p = stats.spearmanr(df['Age'], df['ExperienceInCurrentDomain'])

     print(f"Spearman's Rank Correlation Coefficient: {spearman_corr}")
     print(f"P-value: {spearman_p}")
```

```
Spearman's Rank Correlation Coefficient: -0.14172932292026683
P-value: 2.6218815420869774e-22
```

Name: Neeraj Rijhwani Rollno.64 Div :D15C

4. 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:

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

```
[14] kendall_corr, kendall_p = stats.kendalltau(df['Age'], df['ExperienceInCurrentDomain'])

     print(f"Kendall’s Rank Correlation Coefficient: {kendall_corr}")
     print(f"P-value: {kendall_p}")
```

```
Kendall’s Rank Correlation Coefficient: -0.05223701755751474
P-value: 2.2249017210277004e-06
```

Name: Neeraj Rijhwani Rollno.64 Div :D15C

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:

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

```python
df['Experience_Category'] = pd.cut(df['ExperienceInCurrentDomain'], bins=[0, 5, 10, 20, 30], labels=['0-5', '6-10', '11-20', '21-30'])
df['Performance_Category'] = pd.cut(df['Age'], bins=[20, 30, 40, 50, 60], labels=['20-30', '30-40', '40-50', '50-60'])

contingency_table = pd.crosstab(df['Experience_Category'], df['Performance_Category'])

chi2_stat, p_val, dof, expected = stats.chi2_contingency(contingency_table)

print(f"Chi-Squared Statistic: {chi2_stat}")
print(f"P-value: {p_val}")
print(f"Degrees of Freedom: {dof}")
print("Expected Frequencies Table:")
print(expected)
```

```
Chi-Squared Statistic: 43.97421499426579
P-value: 2.8256641457475885e-10
Degrees of Freedom: 2
Expected Frequencies Table:
[[3.08375430e+03 1.12254235e+03 7.47033504e+01]
 [1.22456957e+01 4.45765472e+00 2.96649604e-01]]
```

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.