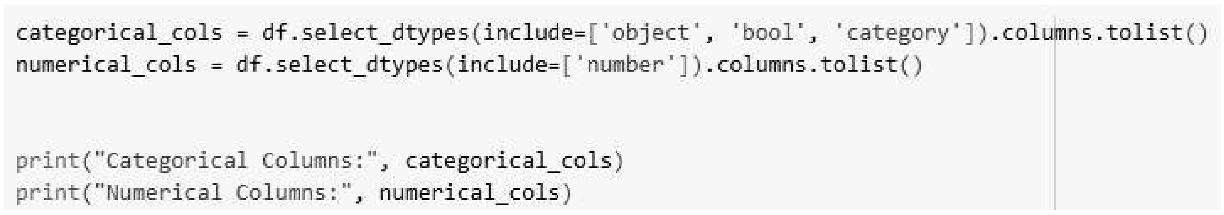
AIDS_Exp4 **Avan Shetty** **D15C-52**

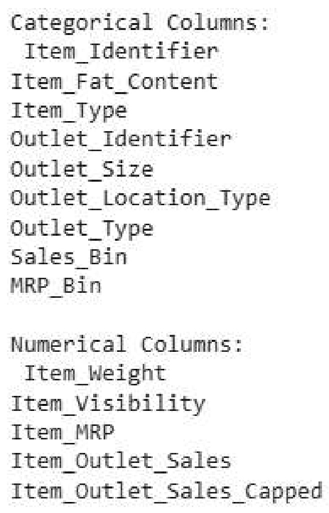
Aim: Implementation of Statistical Hypothesis Test using Scipy and Sci-kit learn.

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

Theory and Output:
1. Data Overview:
The dataset consists of various attributes related to retail products, including product details, pricing, and sales figures. It contains categorical columns such as `**Item_Type**`, `**Outlet_Identifier**`, and `O**utlet_Location_Type**`, which describe product categories and store locations. Numerical attributes include `**Item_Weight**`, `**Item_MRP**`, and `**Item_Outlet_Sales**`, which represent product weight, pricing, and sales revenue. The dataset is useful for analyzing relationships between pricing, sales performance, and product characteristics. Missing values in certain columns may require handling for accurate analysis.



```
categorical_cols = df.select_dtypes(include=['object', 'bool', 'category']).columns.tolist()
numerical_cols = df.select_dtypes(include=['number']).columns.tolist()


print("Categorical Columns:", categorical_cols)
print("Numerical Columns:", numerical_cols)
```



```
Categorical Columns:
 Item_Identifier
Item_Fat_Content
Item_Type
Outlet_Identifier
Outlet_Size
Outlet_Location_Type
Outlet_Type
Sales_Bin
MRP_Bin

Numerical Columns:
 Item_Weight
Item_Visibility
Item_MRP
Item_Outlet_Sales
Item_Outlet_Sales_Capped
```

2. Understanding what is correlation tests:
Correlation tests measure the strength and direction of the relationship between two variables. They help determine how changes in one variable are associated with changes in another.

3. **Pearson Correlation Coefficient**

Pearson's correlation coefficient, also known as Pearson's (r) or simply correlation coefficient, measures the linear relationship between two continuous variables.

The coefficient assumes that the variables are normally distributed and have a linear relationship.

Formula :

$$r = \frac{\sum_{i=1}^{n}\left(\left(x_i - \overline{x}\right)\left(y_i - \overline{y}\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2 \sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2}}$$

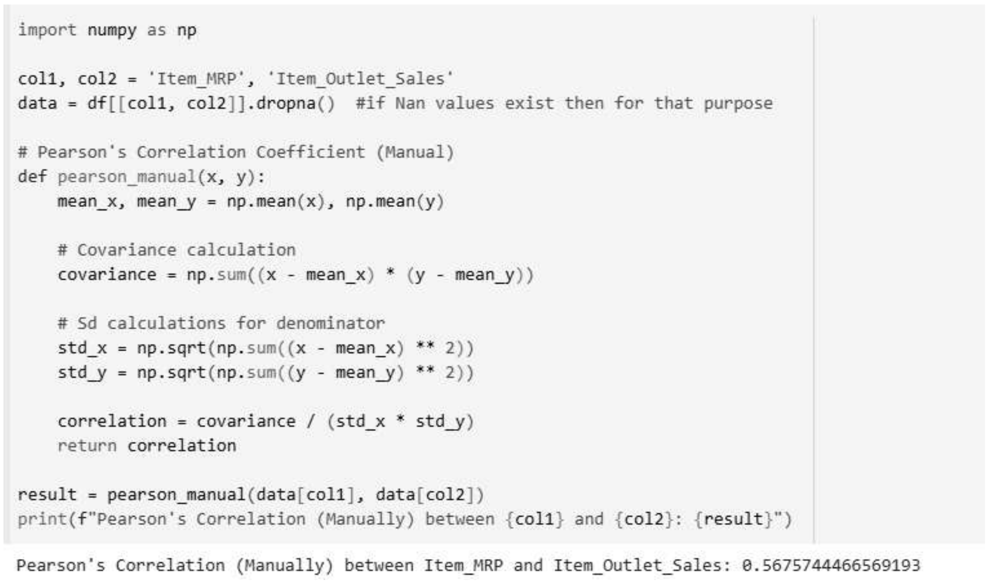
Indications:

1. Numerator (Covariance Term):
   - This calculates the sum of the product of deviations of x and y from their respective means.
   - This represents the covariance between x and y showing how they vary together.
2. Denominator (Standard Deviation Term):
   - The first square root term represents the standard deviation of x
   - The second square root term represents the standard deviation of y
   - This ensures normalization, making r independent of the scale of the variables.

- $r > 0$ → Positive correlation
- $r < 0$ → Negative correlation
- $r = 0$ → No linear correlation.

3. The **p-value** in Pearson's correlation indicates statistical significance:
   - p < 0.05 → Significant correlation, unlikely due to chance.
   - p > 0.05 → Not significant, insufficient evidence of correlation.
   A **p-value of 0** suggests a very strong relationship, often due to a large dataset, confirming the correlation is highly unlikely to be random

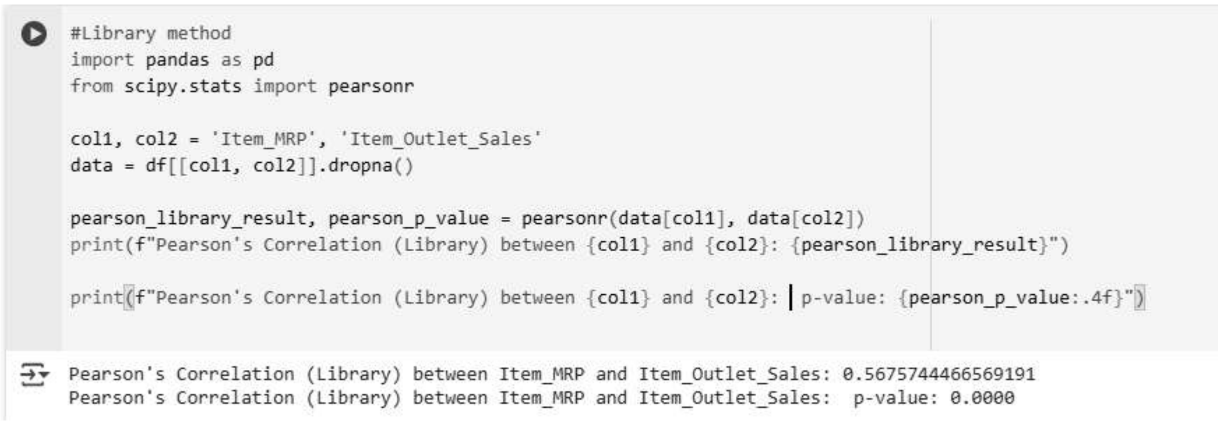
Manual Method:



```
import numpy as np

col1, col2 = 'Item_MRP', 'Item_Outlet_Sales'
data = df[[col1, col2]].dropna()  #if Nan values exist then for that purpose

# Pearson's Correlation Coefficient (Manual)
def pearson_manual(x, y):
    mean_x, mean_y = np.mean(x), np.mean(y)

    # Covariance calculation
    covariance = np.sum((x - mean_x) * (y - mean_y))

    # Sd calculations for denominator
    std_x = np.sqrt(np.sum((x - mean_x) ** 2))
    std_y = np.sqrt(np.sum((y - mean_y) ** 2))

    correlation = covariance / (std_x * std_y)
    return correlation

result = pearson_manual(data[col1], data[col2])
print(f"Pearson's Correlation (Manually) between {col1} and {col2}: {result}")
```

```
Pearson's Correlation (Manually) between Item_MRP and Item_Outlet_Sales: 0.5675744466569193
```

Library Function:



```
#Library method
import pandas as pd
from scipy.stats import pearsonr

col1, col2 = 'Item_MRP', 'Item_Outlet_Sales'
data = df[[col1, col2]].dropna()

pearson_library_result, pearson_p_value = pearsonr(data[col1], data[col2])
print(f"Pearson's Correlation (Library) between {col1} and {col2}: {pearson_library_result}")

print(f"Pearson's Correlation (Library) between {col1} and {col2}: | p-value: {pearson_p_value:.4f}")
```

```
Pearson's Correlation (Library) between Item_MRP and Item_Outlet_Sales: 0.5675744466569191
Pearson's Correlation (Library) between Item_MRP and Item_Outlet_Sales:  p-value: 0.0000
```

Analysis:

The Pearson correlation coefficient was calculated using both a manual approach and scipy.stats.pearsonr(). The manual method involved computing the means, covariance, and standard deviations, yielding a correlation of **0.5676**. The library-based approach, after handling missing values, returned the same result.
Since both methods produced identical values, this confirms the correctness of the manual calculation. The **moderate positive correlation** (0.5676) indicates that **higher** Item MRP is generally associated with higher Item Outlet Sales, though other factors may also influence sales.

The **p-value being 0** suggests that the correlation is statistically significant at any common significance level (e.g., 0.05 or 0.01). This means there is very **strong evidence** that the relationship between Item MRP and Item Outlet Sales is not due to random chance

4. **Spearman Rank Correlation :**

Spearman’s rank correlation coefficient measures the monotonic relationship between two variables. Unlike Pearson’s correlation, it does not assume a linear relationship or normality, making it useful for ordinal and non-normally distributed data.

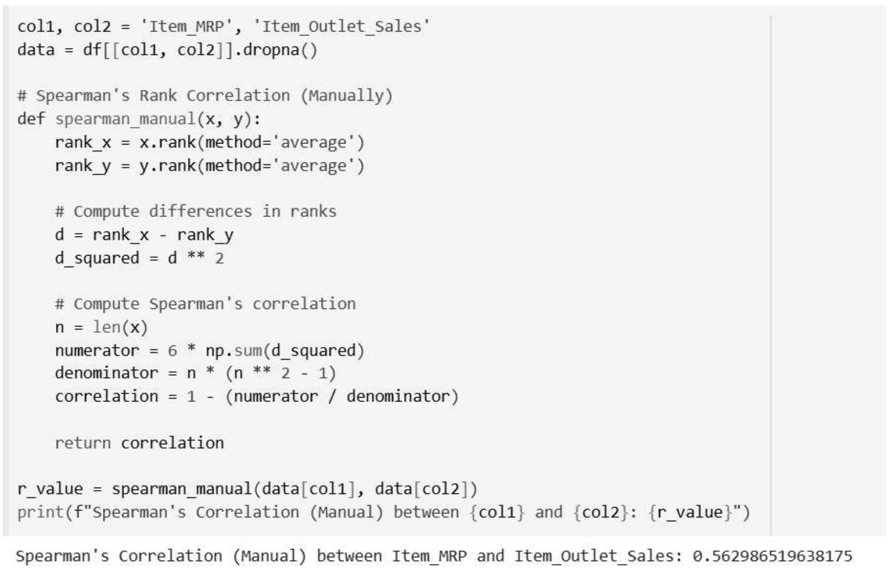
Formula:
Spearman's correlation is computed using the ranked values of the variables. It is given by:

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

Indications:
1: Convert each dataset into ranks, assigning the average rank in case of ties.
2: Compute (di) the difference between ranks for each pair.
3: Square each (di) and sum all values.
4: Apply the formula to calculate r
5. Interpret the results
- $r > 0$ → Positive monotonic relationship
- $r < 0$ → Negative monotonic relationship
- $r = 0$ → No monotonic relationship

6. The p-value tests the null hypothesis that there is no association between the variables:
- $p < 0.05$ → Significant correlation, unlikely due to chance.
- $p > 0.05$ → Not significant, insufficient evidence of correlation.
- A p-value of 0 suggests a highly significant relationship, often due to a large dataset.

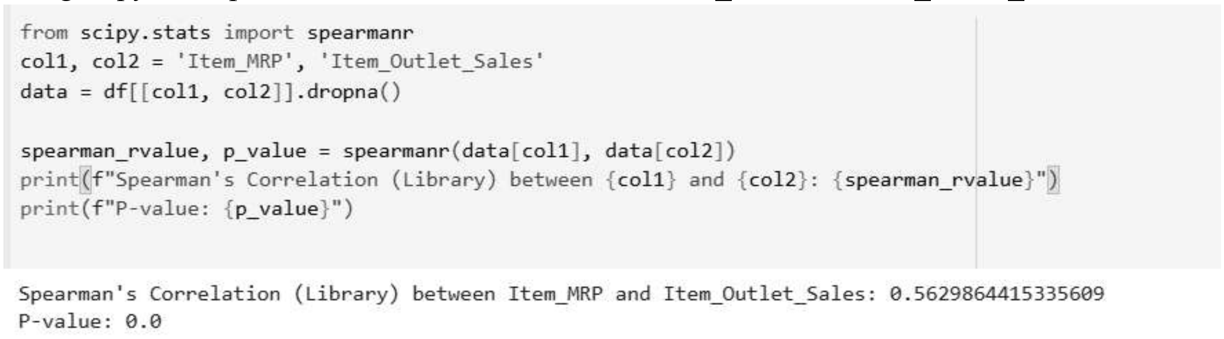
Manual Method:



```
col1, col2 = 'Item_MRP', 'Item_Outlet_Sales'
data = df[[col1, col2]].dropna()

# Spearman's Rank Correlation (Manually)
def spearman_manual(x, y):
    rank_x = x.rank(method='average')
    rank_y = y.rank(method='average')

    # Compute differences in ranks
    d = rank_x - rank_y
    d_squared = d ** 2

    # Compute Spearman's correlation
    n = len(x)
    numerator = 6 * np.sum(d_squared)
    denominator = n * (n ** 2 - 1)
    correlation = 1 - (numerator / denominator)

    return correlation

r_value = spearman_manual(data[col1], data[col2])
print(f"Spearman's Correlation (Manual) between {col1} and {col2}: {r_value}")
```

```
Spearman's Correlation (Manual) between Item_MRP and Item_Outlet_Sales: 0.562986519638175
```

Library Function:
Using scipy.stats.spearmanr to calculate the rank for Item_MRP and Item_Oullet_Sales



```
from scipy.stats import spearmanr
col1, col2 = 'Item_MRP', 'Item_Outlet_Sales'
data = df[[col1, col2]].dropna()

spearman_rvalue, p_value = spearmanr(data[col1], data[col2])
print(f"Spearman's Correlation (Library) between {col1} and {col2}: {spearman_rvalue}")
print(f"P-value: {p_value}")
```

```
Spearman's Correlation (Library) between Item_MRP and Item_Outlet_Sales: 0.5629864415335609
P-value: 0.0
```

Analysis:
From code output:

- Manually calculated value: ρ=0.562986519638175
- Library calculated result: ρ=0.5629864415335609

As it was computed manually and using scipy.stats.spearmanr() to analyze the relationship between Item_MRP and Item_Outlet_Sales, the manual method involved ranking values, computing rank differences, and applying the Spearman formula. yielding a correlation of **0.56299**. The library-based method produced the same result, confirming the correctness of the manually calculated result.
The **p-value was 0.0**, indicating that the correlation is statistically significant at any common significance level. This suggests a **strong relationship** between Item MRP and Item Outlet Sales, though other factors may also influence sales

5. **Kendall's Rank Correlation**

Kendall's tau measures the strength and direction of the association between two variables. It is non-parametric and does not assume normality or a linear relationship, making it ideal for **ordinal data** or **nonlinear monotonic relationships**.

Formula :

Kendall's correlation is computed based on the number of concordant and discordant pairs:

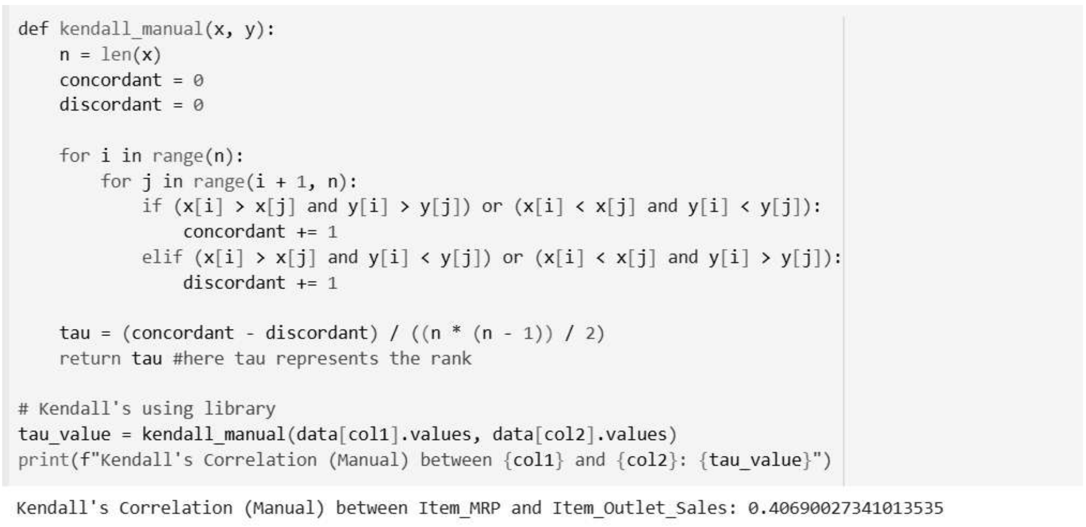
$$\tau = \frac{\text{Number of concordant pairs} - \text{Number of discordant pairs}}{n(n-1)/2}$$

Concordant: When the order of values in both variables matches (i.e., if one value increases, the other also increases).
Discordant: When the order of values in one variable is reversed in the other (i.e., if one value increases while the other decreases).

n(n−1)/2 is the total number of unique pairs

<u>Steps for Calculation:</u>
1. Compare all pairs: Count concordant (C) and discordant (D) pairs.
2. Compute tau: Use the formula to calculate Kendall's correlation coefficient.
3. Interpret results:
    - $\tau > 0 \rightarrow$ Positive monotonic relationship
    - $\tau < 0 \rightarrow$ Negative monotonic relationship
    - $\tau = 0 \rightarrow$ No monotonic relationship

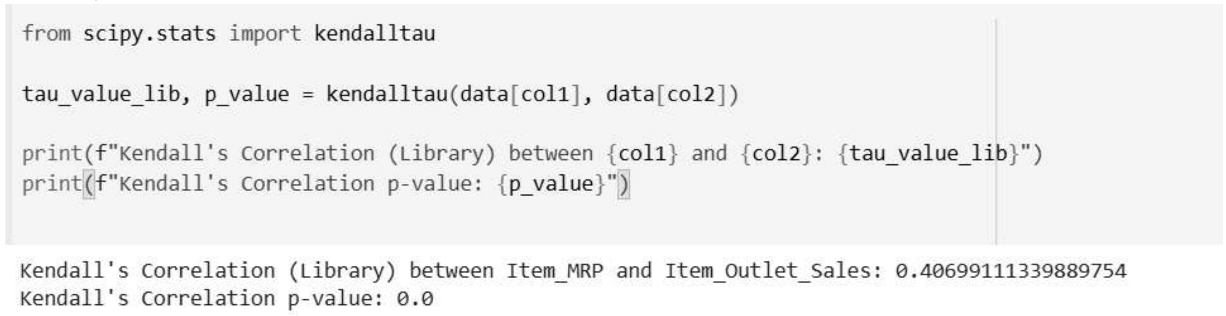
Manual Method:



```
def kendall_manual(x, y):
    n = len(x)
    concordant = 0
    discordant = 0

    for i in range(n):
        for j in range(i + 1, n):
            if (x[i] > x[j] and y[i] > y[j]) or (x[i] < x[j] and y[i] < y[j]):
                concordant += 1
            elif (x[i] > x[j] and y[i] < y[j]) or (x[i] < x[j] and y[i] > y[j]):
                discordant += 1

    tau = (concordant - discordant) / ((n * (n - 1)) / 2)
    return tau #here tau represents the rank

# Kendall's using library
tau_value = kendall_manual(data[col1].values, data[col2].values)
print(f"Kendall's Correlation (Manual) between {col1} and {col2}: {tau_value}")
```

```
Kendall's Correlation (Manual) between Item_MRP and Item_Outlet_Sales: 0.40690027341013535
```

Library Function:



```
from scipy.stats import kendalltau

tau_value_lib, p_value = kendalltau(data[col1], data[col2])

print(f"Kendall's Correlation (Library) between {col1} and {col2}: {tau_value_lib}")
print(f"Kendall's Correlation p-value: {p_value}")
```

```
Kendall's Correlation (Library) between Item_MRP and Item_Outlet_Sales: 0.40699111339889754
Kendall's Correlation p-value: 0.0
```

Analysis:
- Manually calculated Kendall's correlation: **0.4069**
- Library-calculated Kendall's correlation: **0.40699**

The Kendall's Tau value of **0.4069** indicates a positive correlation between Item MRP and Item Outlet Sales. This suggests that as Item MRP increases, Item Outlet Sales tend to increase as well. The p-value of 0.0 confirms that the correlation is statistically significant, indicating a reliable relationship, though other factors could also influence sales.

6. **Chi-Squared Test**

The Chi-Squared Test of Independence is used to determine if there is a significant association between two categorical variables. It compares the observed frequency of occurrences in each category with the expected frequency, under the assumption that the variables are independent

Formula:
The Chi-Squared statistic is calculated using the formula:

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

Where:
1. Oi = observed frequency in the i-th category
2. Ei = expected frequency in the i-th category
3. The sum is taken over all categories

Steps for Calculation:
1. Create a Contingency Table:
Organize the observed data into a contingency table (cross-tabulation) showing the frequency of occurrences for each combination of the two categorical variables.
2. Calculate Expected Frequencies:
For each cell in the contingency table, calculate the expected frequency under the assumption that the variables are independent:
3. Compute Chi-Squared Statistic:
Calculate the Chi-Squared statistic using the formula

4. Degrees of Freedom (df):
The degrees of freedom are calculated as: $df=(r-1)\times(c-1)$: [r = rows, c = cols]
5. Compare with Critical Value:
The Chi-Squared statistic is compared with a critical value from the Chi-Squared distribution table at the chosen significance level (usually $\alpha=0.05$) and degrees of freedom. If the statistic is greater than the critical value, we reject the null hypothesis.

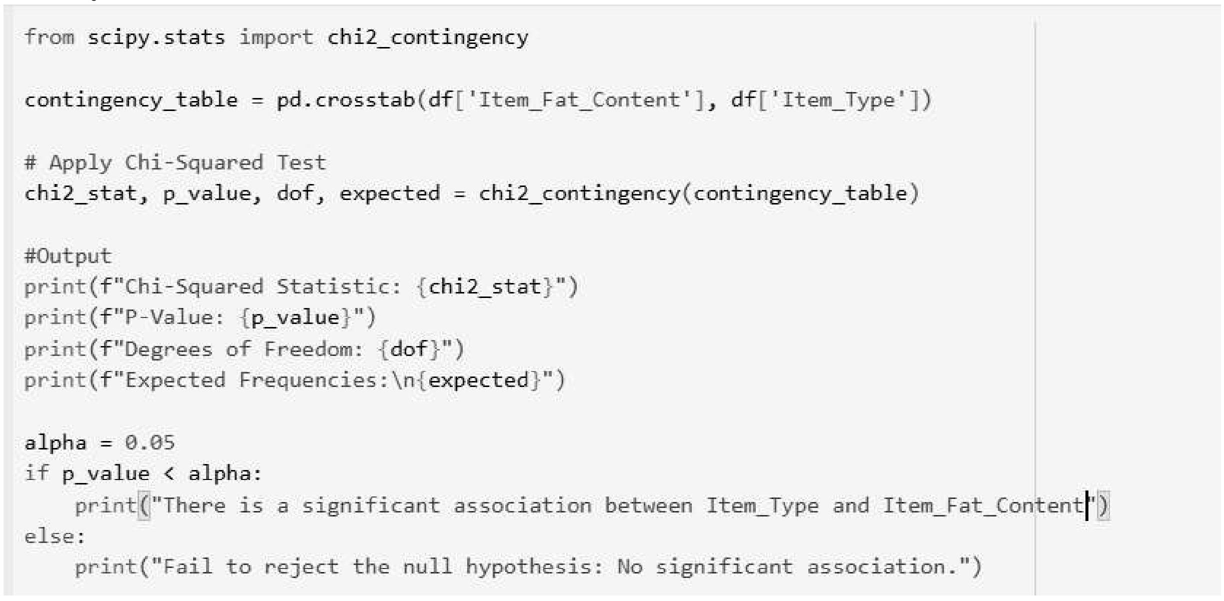
Interpret Results:
- $p < 0.05$: Significant relationship, the variables are likely associated.
- $p > 0.05$: Fail to reject the null hypothesis, no significant relationship, the variables are likely independent.

Manual Method:

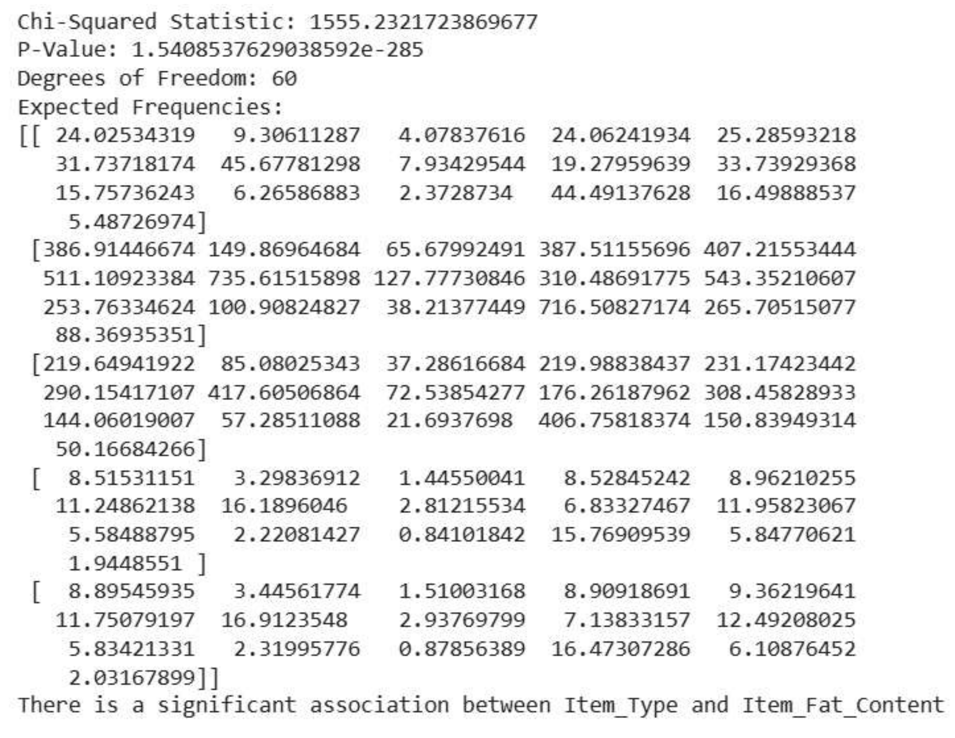
```python
from scipy.stats import chi2
contingency_table = pd.crosstab(df['Item_Type'], df['Item_Fat_Content'])

#Expected frequencies (Ei)
total = contingency_table.sum().sum()
rows = contingency_table.sum(axis=1)
cols = contingency_table.sum(axis=0)

expected = np.outer(rows, cols) / total

#Calculate the Chi-Squared statistic
observed = contingency_table.values  #(Oi)
chi_squared_stat = ((observed - expected) ** 2 / expected).sum()

#(dof) = (rows - 1) * (cols - 1)
df = (contingency_table.shape[0] - 1) * (contingency_table.shape[1] - 1)

alpha = 0.05
critical_value = chi2.ppf(1 - alpha, df)

if chi_squared_stat > critical_value:
    result = 'There is a significant relationship. between Item_Type and Item_Fat_Content'
else:
    result = 'Fail to reject null hypothesis: No significant relationship.'

#final output
print(f"Chi-Squared Statistic: {chi_squared_stat}")
print(f"Critical Value: {critical_value}")
print(result)
```

```
Chi-Squared Statistic: 1555.2321723869677
Critical Value: 79.08194448784874
There is a significant relationship. between Item_Type and Item_Fat_Content
```

Library Function:



```
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(df['Item_Fat_Content'], df['Item_Type'])

# Apply Chi-Squared Test
chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)

#Output
print(f"Chi-Squared Statistic: {chi2_stat}")
print(f"P-Value: {p_value}")
print(f"Degrees of Freedom: {dof}")
print(f"Expected Frequencies:\n{expected}")

alpha = 0.05
if p_value < alpha:
    print("There is a significant association between Item_Type and Item_Fat_Content")
else:
    print("Fail to reject the null hypothesis: No significant association.")
```



```
Chi-Squared Statistic: 1555.2321723869677
P-Value: 1.5408537629038592e-285
Degrees of Freedom: 60
Expected Frequencies:
[[ 24.02534319   9.30611287   4.07837616  24.06241934  25.28593218
   31.73718174  45.67781298   7.93429544  19.27959639  33.73929368
   15.75736243   6.26586883   2.3728734   44.49137628  16.49888537
    5.48726974]
 [386.91446674 149.86964684  65.67992491 387.51155696 407.21553444
  511.10923384 735.61515898 127.77730846 310.48691775 543.35210607
  253.76334624 100.90824827  38.21377449 716.50827174 265.70515077
   88.36935351]
 [219.64941922  85.08025343  37.28616684 219.98838437 231.17423442
  290.15417107 417.60506864  72.53854277 176.26187962 308.45828933
  144.06019007  57.28511088  21.6937698  406.75818374 150.83949314
   50.16684266]
 [  8.51531151   3.29836912   1.44550041   8.52845242   8.96210255
   11.24862138  16.1896046    2.81215534   6.83327467  11.95823067
    5.58488795   2.22081427   0.84101842  15.76909539   5.84770621
    1.9448551 ]
 [  8.89545935   3.44561774   1.51003168   8.90918691   9.36219641
   11.75079197  16.9123548    2.93769799   7.13833157  12.49208025
    5.83421331   2.31995776   0.87856389  16.47307286   6.10876452
    2.03167899]]
There is a significant association between Item_Type and Item_Fat_Content
```

Analysis:

The Chi-Squared Statistic calculated manually (1555.23) and using the library function (1555.23) match, confirming the correctness of the calculation. The manual method compared the Chi-Squared Statistic with the critical value of 79.08 (at α = 0.05 and 60 degrees of freedom). Since the statistic is much larger than the critical value, we reject the null hypothesis, indicating a significant association between the two categorical variables.

Additionally, the library function provides a p-value of 1.54e-285, which is effectively zero. Since this p-value is far below the significance level of 0.05, it reinforces the rejection of the null hypothesis. Both methods confirm that there is a significant association between the two variables, meaning that '**Item_Fat_Content**' and '**Item_Type**' are not independent, and their distribution is significantly related.

Final Observations :

Test	Relationship between variables
Pearson's Correlation	Moderate positive linear relationship (r = 0.5676)
Spearman's Rank Correlation	Moderate positive monotonic relationship (ρ = 0.56299)
Kendall's Rank Correlation	Moderate positive monotonic relationship (τ = 0.4069)
Chi-Squared Test	Significant association (Chi-Squared Statistic = 1555.23, p-value ≈ 0)

Conclusion:
We applied various statistical hypothesis tests Pearson's correlation, Spearman's rank correlation, Kendall's Tau, and the Chi-Squared test to analyze relationships in a retail dataset. The results revealed a positive association between `**Item_MRP**` and `**Item_Outlet_Sales**`, indicating that higher pricing tends to correlate with higher sales. Additionally, the Chi-Squared test showed a significant relationship between categorical variables like `**Item_Fat_Content**` and `**Item_Type**`, suggesting that these factors are not independent. Rather than depending on the library methods we calculated using the formula manually to understand in depth the consistency this dataset holds.