**Hypothesis**

The output **(1.5976240527147705, 0.1101266701438426)** from the **ztest** function typically represents the results of a z-test in statistics. Let's break down what each part of this output means:

1. **First Value (1.5976240527147705)**: This value is the calculated test statistic (z-score) of the z-test. In statistical hypothesis testing, particularly with z-tests, the test statistic measures how far the sample mean is from the population mean in terms of standard errors. A higher absolute value of the z-score indicates stronger evidence against the null hypothesis.
2. **Second Value (0.1101266701438426)**: This value is the p-value associated with the z-test. The p-value is a measure that helps us assess the strength of the evidence against the null hypothesis. It represents the probability of observing the test statistic (or one more extreme) under the assumption that the null hypothesis is true. A lower p-value suggests stronger evidence against the null hypothesis.
3. The **ttest\_1samp** function from **scipy.stats** is used to perform a one-sample t-test in Python. This statistical test is used to determine whether the mean of a single sample differs significantly from a known or hypothesized population mean.

**ttest\_1samp** function returns a tuple **(t-statistic, p-value)**, where:

1. **t-statistic**: This is the calculated t-statistic (t-score) from the t-test, which measures how many standard errors the sample mean is away from the population mean.
2. **p-value**: This is the two-tailed p-value associated with the t-test. It represents the probability of observing the sample data's mean (or one more extreme) if the null hypothesis (that the sample mean equals the population mean) is true.
3. **T-Statistic**: The t-statistic indicates the strength and direction of the difference between the sample mean and the population mean, expressed in terms of standard errors.
4. **P-Value**: The p-value helps assess the statistical significance of the observed difference. A lower p-value (< 0.05) typically indicates stronger evidence against the null hypothesis.
5. The **ttest\_ind** function from **scipy.stats** is used to perform an independent two-sample t-test in Python. This statistical test is used to determine whether the means of two independent samples differ significantly.
6. **T-Statistic**: The t-statistic indicates the strength and direction of the difference between the means of the two samples, relative to the variance within the data.
7. **P-Value**: The p-value helps assess the statistical significance of the observed difference. A lower p-value (< 0.05) typically indicates stronger evidence against the null hypothesis (that the means of the two samples are equal), suggesting that the samples are likely drawn from populations with different means.
8. To perform a chi-squared test using **scipy.stats.chisquare**, you should provide the observed frequencies (**f\_obs**) and, optionally, the expected frequencies (**f\_exp**) for the test.
9. **f\_obs**: This is the array of observed frequencies. It can be a 1-dimensional array representing observed counts in categories
10. **f\_exp**: This is the array of expected frequencies (optional). If provided, it should have the same shape as **f\_obs**.

The **chisquare** function returns a tuple **(chi2 statistic, p-value)**, where:

1. **chi2 statistic**: This is the computed chi-squared statistic based on the observed and expected frequencies.
2. **p-value**: This is the p-value associated with the chi-squared test. It represents the probability of observing the test statistic (or one more extreme) under the null hypothesis (that the observed and expected frequencies are independent).

**Purpose:chisquare**

The chi-square test of independence is used to determine whether there is a significant association between categorical variables in a contingency table. It tests the null hypothesis that the categorical variables are independent.

**Parameters:**

The function chi2\_contingency takes a contingency table as input and returns several outputs that are useful for interpreting the results of the chi-square test:

* **observed**: The observed frequencies in the contingency table.
* **expected**: The expected frequencies under the null hypothesis of independence.
* **chi2**: The test statistic (chi-squared).
* **p**: The p-value of the test.
* **df**: Degrees of freedom.

The code snippet you've provided is using pd.crosstab from the pandas library to create a contingency table based on two columns (Gender and isSmoker) from a DataFrame df. Here’s a step-by-step explanation and example of how this works:

### Step-by-Step Explanation:

1. **Import Pandas**: Make sure you have pandas imported (import pandas as pd).
2. **Create Contingency Table**: Use pd.crosstab to create a contingency table:

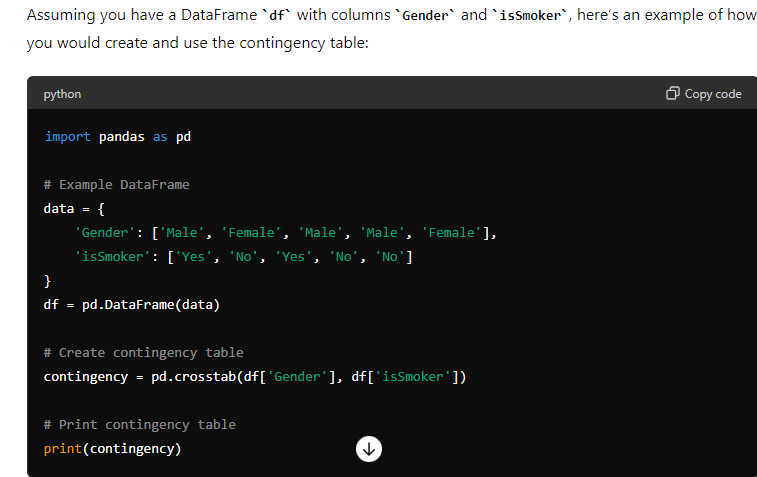
**contingency = pd.crosstab(df['Gender'], df['isSmoker'])**

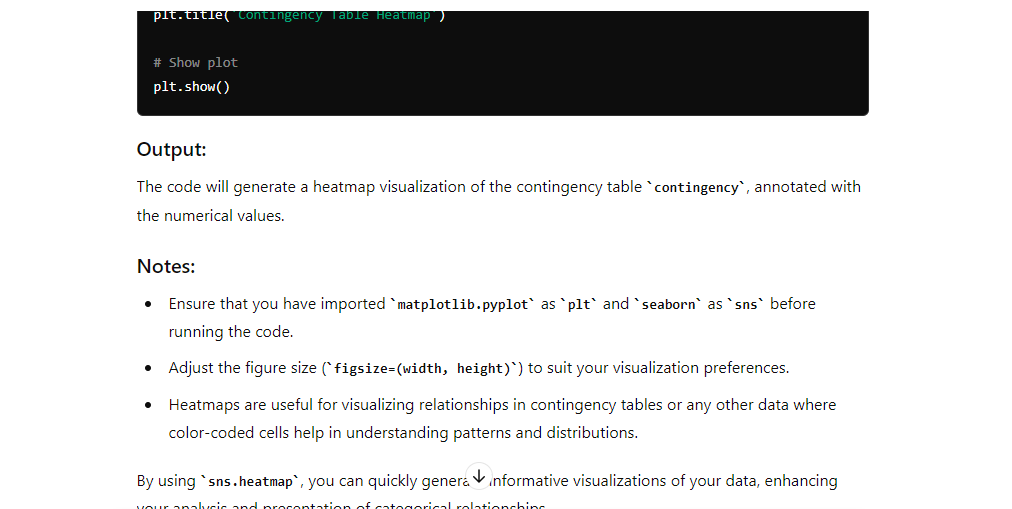
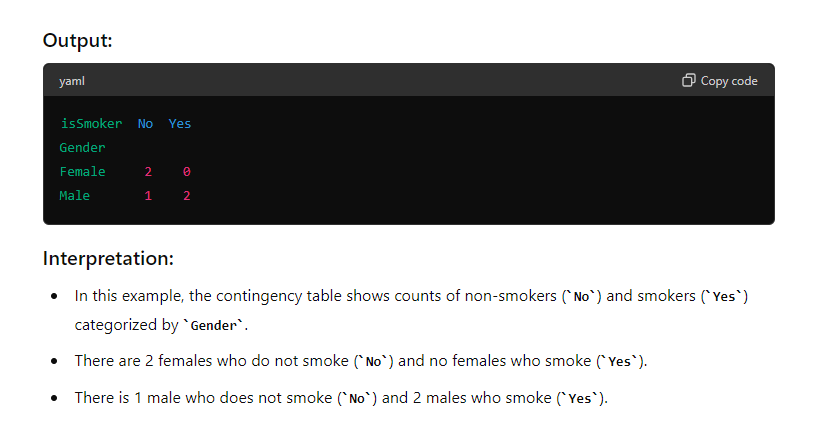
 df['Gender'] and df['isSmoker'] are the columns from the DataFrame df that you want to analyze.

pd.crosstab computes a simple cross-tabulation of two (or more) factors. It calculates a table of frequencies for these factors.

 **Store Results**: The resulting table (contingency) will have rows corresponding to unique values of Gender and columns corresponding to unique values of isSmoker. The cells will contain counts of occurrences where each combination of Gender and isSmoker occurs in the DataFrame.

Eg:-



****