

Learning McNemar's Test: A Python Tutorial for Paired Data Analysis

Authored by
Mohammed loot

November 8, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning McNemar's Test: A Python Tutorial for Paired Data Analysis*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=12703>

In the realm of statistical analysis, dealing with data where observations are linked--known as **paired data** or repeated measures--requires specialized tools. Among these, [McNemar's Test](#) stands out as a powerful non-parametric statistical technique designed specifically for assessing differences in proportions between two dependent samples. This test is indispensable when analyzing scenarios where subjects are measured before and after a critical event, intervention, or treatment, and the outcome variable is **dichotomous** (e.g., success/failure, yes/no, agree/disagree).

The central purpose of McNemar's Test is to evaluate whether the observed changes in classification are sufficiently large to be considered a [statistically significant difference](#), rather than merely random fluctuation. Unlike tests for independent samples (like the standard Chi-Squared test), McNemar's Test leverages the inherent relationship within the paired observations, making it highly sensitive to true shifts in population characteristics.

This expert tutorial offers a thorough examination of the underlying methodology of McNemar's Test. We will delve into its requirements, explore a practical case study, and provide step-by-step instructions on how to implement this crucial statistical procedure using the robust statistical modeling library in **Python**.

The Core Methodology: Marginal Homogeneity

McNemar's Test is founded on the principle of assessing **marginal homogeneity**. This refers to the statistical equality of the marginal totals (row and column totals) derived from a specific 2x2 contingency table constructed from the paired data. When we utilize a [repeated measures design](#), we collect two sets of measurements (Time 1 and Time 2) from the exact same cohort of subjects. The results are summarized in a matrix where the cells represent the counts of subjects transitioning between the two categorical states.

The test is specifically designed to evaluate the impact of an intervention by formalizing the [null hypothesis](#) (H_0). The [null hypothesis](#) states that the probability of a subject transitioning from State A at Time 1 to State B at Time 2 is equal to the probability of transitioning from State B at Time 1 to State A at Time 2. In simpler terms, if the intervention has no effect, we expect an equal number of people changing their classification in both directions.

The power of McNemar's method stems from its ability to ignore the **concordant pairs**--the subjects who maintained the same status both before and after the intervention. These subjects, while important for context, provide no statistical information regarding the effectiveness or influence of the intervention itself. By excluding them, the test achieves a focused, high-resolution analysis of true change.

Focusing on Discordant Pairs in the 2x2 Table

To perform McNemar's Test, the data must first be structured into a standard 2x2 table. Let's denote the two outcomes as 'Success' (S) and 'Failure' (F). The table structure organizes the counts based on the paired observations, where the rows usually represent the post-intervention status and the columns represent the pre-intervention status.

Cell A: Subjects classified as S at Time 1 and S at Time 2 (Concordant).

Cell D: Subjects classified as F at Time 1 and F at Time 2 (Concordant).

Cell B: Subjects classified as S at Time 1 and F at Time 2 (Discordant Change 1).

Cell C: Subjects classified as F at Time 1 and S at Time 2 (Discordant Change 2).

The critical comparison hinges entirely on cells B and C, the **discordant pairs**. Cell B counts the subjects who switched from the initial status to the secondary status, while Cell C counts those who switched in the opposite direction. If the intervention is successful, we expect a significant imbalance between the counts B and C.

If the test yields a [statistically significant difference](#) (i.e., we reject the [null hypothesis](#)), it directly implies that the probability of changing status in one direction is not equal to the probability of changing in the other. This deviation from marginal homogeneity provides compelling evidence that the intervention caused a measurable shift in the paired proportions, confirming its effectiveness.

Practical Application: A Market Research Opinion Study

To solidify our understanding, let us examine a typical application of McNemar's Test within market research. Imagine an organization launching a controversial yet important marketing campaign via a video designed to sway public sentiment regarding a proposed piece of legislation. A cohort of 100 participants is recruited, and their opinions (Support or Do Not Support) are recorded at two distinct stages: before viewing the video (Pre-Intervention) and immediately after viewing the video (Post-Intervention).

Since the fundamental unit of observation remains the individual participant measured across two time points, the data structure is inherently **paired**. Using an inappropriate test for independent samples would violate statistical assumptions and lead to inflated Type I error rates. McNemar's Test correctly accounts for the dependency within the data, providing a robust framework for drawing conclusions about the video's causal influence on opinion shift.

The resulting data is meticulously summarized in the 2x2 contingency table below. Note how the table maps the exact trajectory of each participant's opinion status from the initial measurement to the final measurement, providing a complete picture of the transitions observed across the sample:

	Before Marketing Video	
After Marketing Video	Support	Do not support
Support	30	40
Do not Support	12	18

Analyzing this table reveals several key insights. The **concordant pairs** are 30 (Supported both times) and 18 (Did not support both times). The crucial **discordant pairs** are the 40 individuals who moved from 'Do not support' (Before) to 'Support' (After), and the 12 individuals who moved in the opposite, undesired direction--from 'Support' (Before) to 'Do not Support' (After). Our statistical challenge is to determine if the difference between these two transition counts (40 vs. 12) is truly reflective of a successful intervention or merely due to random chance.

Implementing McNemar's Test Using Python's statsmodels Library

Python provides excellent resources for advanced statistical computation, notably through the [statsmodels](#) library. This package is specifically engineered to handle complex statistical modeling and inference, including procedures tailored for dependent data like McNemar's Test. The implementation requires careful preparation of the data input and precise specification of the testing parameters.

Step 1: Structuring the Contingency Data Matrix

Before executing the test, the 2x2 contingency table must be accurately structured into a format readable by Python, typically a nested list or a **NumPy array**. It is absolutely crucial that the data matrix mirrors the convention required by the statistical function, which generally places the "After" measurements along the rows and the "Before" measurements along the columns. This ensures that the off-diagonal elements (the discordant pairs) are correctly identified by the function for calculation.

```
data = ,
]
```

In this specific structure, the count 40 represents the transition from 'Before Do Not Support' to 'After Support' (cell C), and 12 represents the transition from 'Before Support' to 'After Do Not Support' (cell B). This precise alignment is the foundation upon which the statistical test is built.

Step 2: Selecting the Appropriate Statistical Approach

The `mcnemar`` function, imported from the `statsmodels.stats.contingency_tables`` module, is the

workhorse for this analysis. It offers flexibility through its parameters, allowing the user to select between the exact test (based on the [binomial distribution](#)) and the asymptotic test (based on the [Chi-Square distribution](#)). The appropriate choice depends primarily on the total frequency of the discordant pairs ($B + C$).

The primary parameters controlling the test execution are:

table: The required input, specifying the 2x2 matrix of paired frequencies.

exact: If set to **True**, the function computes the P-value using the **exact binomial test**, which is highly recommended when the sum of discordant pairs ($40 + 12 = 52$ in our case) is small, typically less than 20. When **False**, the function defaults to the asymptotic approach, which relies on the large-sample approximation of the [Chi-Square distribution](#). Given our large sample of 52 discordant pairs, the asymptotic method is statistically valid and often preferred for computational efficiency.

correction: This parameter applies specifically when using the asymptotic test (`exact=False`). If **True**, a **continuity correction** (often referred to as Yates' correction) is applied. This correction adjusts the Chi-Square statistic downward, leading to a more conservative P-value. While historically common, modern statisticians sometimes omit this correction unless cell sizes are extremely small.

Step 3: Executing the McNemar Test and Reviewing Output

The following Python script demonstrates how to load the data and execute the asymptotic McNemar's Test, first without applying Yates' continuity correction, and then applying it for comparison. This showcases the minor, yet sometimes important, impact of the correction on the final statistical outcome.

```
from statsmodels.stats.contingency_tables import mcnemar
import numpy as np
```

```
data = np.array([, ])
```

```
# Test 1: Asymptotic Chi-Square approach (Standard)
print("Result without Continuity Correction:")
print(mcnemar(data, exact=False, correction=False))
```

```
# Test 2: Asymptotic Chi-Square approach (With Yates' Correction)
print("\nResult with Continuity Correction:")
print(mcnemar(data, exact=False, correction=True))
```

```
# Output Example:
```

Result without Continuity Correction:

pvalue 0.000103

statistic 15.077

Result with Continuity Correction:

pvalue 0.000181

statistic 14.019

Interpreting the Results and Drawing Conclusions

The output provided by [statsmodels](#) delivers two critical statistical values: the **test statistic** (which, in the asymptotic case, is a Chi-Square value, χ^2) and the corresponding **P-value**. The test statistic measures the magnitude of the observed discrepancy between the two discordant counts (40 vs. 12). A larger statistic indicates a greater deviation from the expectation set by the [null hypothesis](#).

The P-value represents the probability of observing a difference in transition rates as extreme as, or more extreme than, the one calculated, assuming the [null hypothesis](#) of no effect (marginal homogeneity) is true. Comparing the P-values from both tests--0.000103 (uncorrected) and 0.000181 (corrected)--we see that both values are extremely small, falling far below the standard threshold for significance, $\alpha = 0.05$.

Because $P < 0.05$, we possess overwhelming evidence to **reject the null hypothesis**. This statistically robust conclusion indicates that the proportion of participants supporting the law before the marketing video intervention is [statistically significantly different](#) from the proportion supporting it afterward. Furthermore, by observing that the number of positive shifts (40) is dramatically higher than the number of negative shifts (12), we confirm the direction of the effect: the marketing video was successful in promoting support for the law among the surveyed population.