How can I perform McNemar’s Test using Python?

Authored by

stats writer

April 16, 2024

McNemar's Test is a statistical test used to compare the proportions of two paired samples. In order to perform McNemar's Test using Python, one can use the statsmodels library which offers a function called mcnemar(). This function takes in the two paired samples as input and returns the p-value and test statistic for the test. The p-value can then be compared to a pre-determined significance level to determine if there is a significant difference between the two samples. Additionally, packages like scipy and scikit-learn also offer functions for performing McNemar's Test. Overall, using Python to perform McNemar's Test provides a quick and efficient way to analyze paired sample data and make statistical inferences.

Perform McNemar's Test in Python

McNemar's Test is used to determine if there is a statistically significant difference in proportions between paired data.

This tutorial explains how to perform McNemar's Test in Python.

Example: McNemar's Test in Python

Suppose researchers want to know if a certain marketing video can change people's opinion of a particular law. They survey 100 people to find out if they do or do not support the law. Then, they show all 100 people the marketing video and survey them again once the video is over.

The following table shows the total number of people who supported the law both before and after viewing

the video:

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

To determine if there was a statistically significant difference in the proportion of people who supported the law before and after viewing the video, we can perform McNemar's Test.

Step 1: Create the data.

First, we will create a table to hold our data:

```
data = ,
]
```

Step 2: Perform McNemar's Test

Next, we can use the from the statsmodels Python library, which uses the following syntax:

```
mcnemar(table, exact=True, correction=True)
```

where:

table: A square contingency tableexact: If exact is true, then the binomial distribution will be used. If exact is false, then the Chi-Square distribution will be usedcorrection: If true, a continuity correction is used. As a rule of thumb, this correction is typically applied when any of the cell counts in the table are less than 5.

The following code shows how to use this function in our specific example:

```
from statsmodels.stats.contingency_tables import mcnemar

#McNemar's Test with no continuity correction
print(mcnemar(data, exact=False))
```

pvalue 0.000181
statistic 14.019

```
#McNemar's Test with continuity correction
print(mcnemar(data, exact=False, correction=False))
```

pvalue 0.000103
statistic 15.077

This means in both cases we would reject the null

hypothesis and conclude that the proportion of people who supported the law before and after watching the marketing video was statistically significant different.