


Member-only story

STATISTICS

Gentle Introduction to Chi-Square Test for Independence


Beginners guide to Chi-square using Jupyter Notebook





Shinichi Okada · Follow

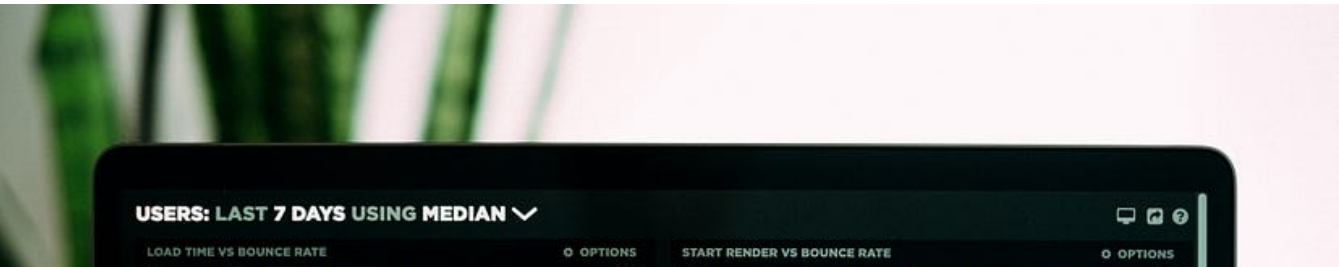
Published in Towards Data Science

9 min read · Jan 20, 2020

 Listen

 Share

 More



Open in app



Photo by [Luke Chesser](#) on [Unsplash](#)

Table of contents

Introduction

1. Prerequisite
2. SciPy package
3. Setup
4. Python indexing
5. chi2 contingency
6. Expected values
7. χ^2 value
8. Side note about Latex
9. p-value
10. Degree of freedom
11. Importing data
12. `Pandas.DataFrame.transpose()`
13. Critical values
14. The null and alternative hypotheses

Conclusion

Introduction

The Chi-square test for independence is also called Pearson's chi-square test. Chi-square test for independence is used in science, economics, marketing, or other various fields. There are three ways to use the Chi-square. The Chi-square test for independence shows how two sets of data are independent of each other. Chi-square of the Goodness-of-fit test shows how different your data to the expected value. The test for homogeneity determines if two or more populations have the same distribution of a single categorical variable.

In this article, we are going to explore the Chi-square test for independence with the Jupyter Notebook. Oh by the way we pronounce "Chi" as "kai" like "kite", NOT "chi" in "Chili". χ is a Greek letter for "Chi", so χ^2 reads Chi-square.

Prerequisite

Even though this article is aimed at beginners who have little experience with coding, reading "[Beginner's Guide to Jupyter Notebook From](#)" will help you how to get started.

Beginner's Guide to Jupyter Notebook

From the setup to the descriptive statistics

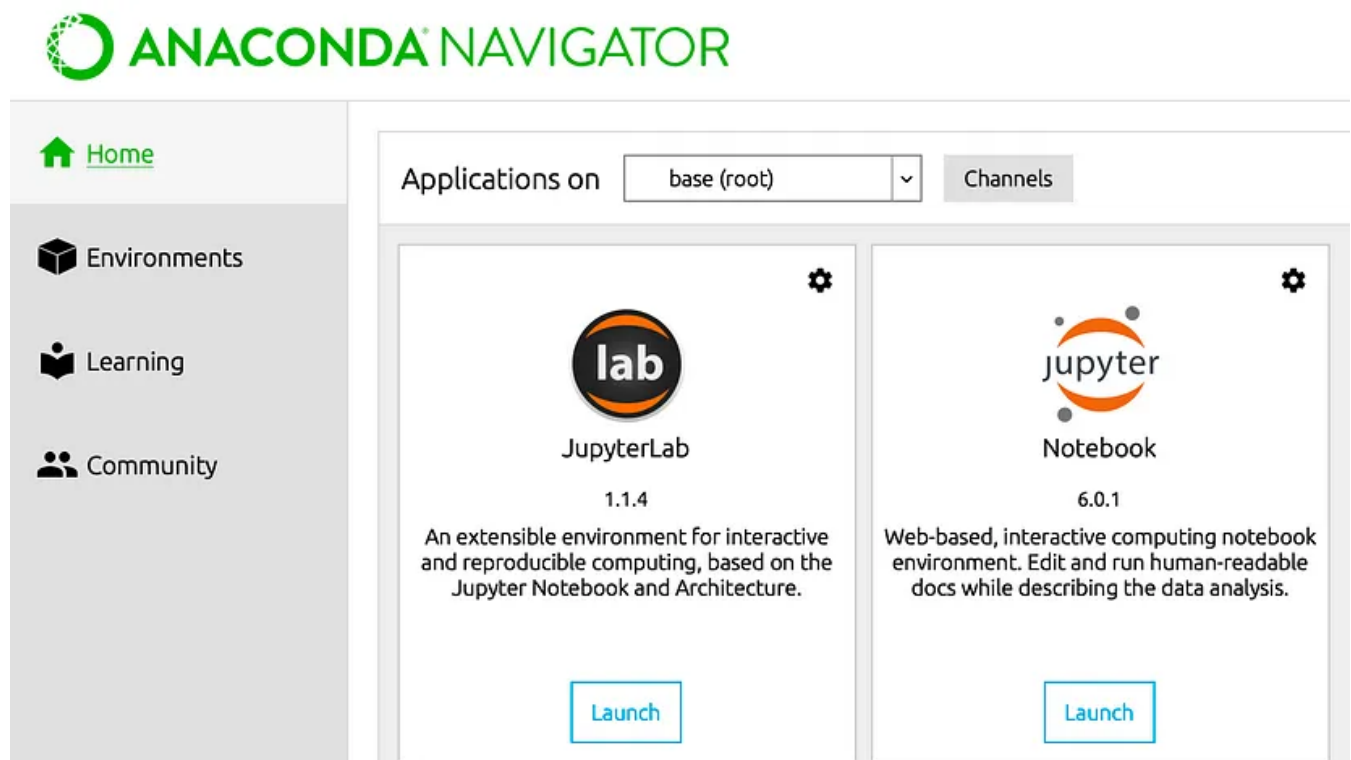
towardsdatascience.com

SciPy package

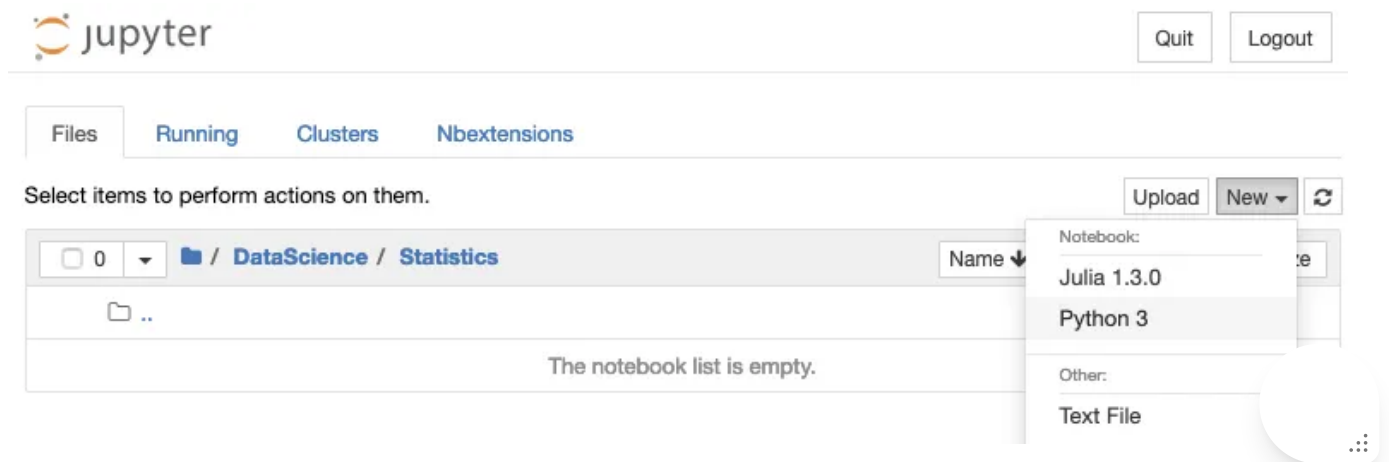
In order to find Chi-square, we are going to use the SciPy package. SciPy is a Python-based open-source software for mathematics, science, and engineering. `scipy.stats.chi2_contingency` is a useful tool for the Chi-square test for independence. There is another one called `scipy.stats.chisquare` which is used for Chi-square of Goodness of fit test.

Setup

Start Anaconda and launch Jupyter Notebook.



Create a file by clicking New > Python 3.



Rename the file to “Chi-square test for independence”.



In the first cell, we are going to import `chi2_contingency`, `pandas` and `numpy` libraries.

```
from scipy.stats import chi2_contingency
import pandas as pd
import numpy as np
```

When you run code in Jupyter Notebook, you press **SHIFT + RETURN**.

We are going to create sample data. Let's say we collected data on the favorite color of T-shirts for men and women. We want to find out whether color and gender are independent or not. We create a small sample data using the Pandas dataframe and we will store our data in a variable called `tshirts`.

Pandas `index` and `columns` are used to name rows and columns. In order to print what's in our `tshirts` variable, we just write `tshirts` at the end and enter SHIFT + RETURN.

```
tshirts = pd.DataFrame(
    [
        [48,22,33,47],
        [35,36,42,27]
    ],
    index=["Male","Female"],
    columns=["Balck","White","Red","Blue"])
tshirts
```

	Balck	White	Red	Blue
Male	48	22	33	47
Female	35	36	42	27

You can find what the labels in the columns are by using `columns`.

```
tshirts.columns
```

```
Index(['Balck', 'White', 'Red', 'Blue'], dtype='object')
```

Similarly, you can use the `index` to find out what indexes are.

```
Index(['Male', 'Female'], dtype='object')
```

Python indexing

Python uses zero-based indexing. That means, the first element has an index 0, the second has index 1, and so on. If you want to access the fourth value in the `chi2_contingency(tshirts)` you need to use `[3]`.

chi2_contingency

SciPy's `chi2_contingency()` returns four values, χ^2 value, p-value, degree of freedom and expected values.

```
chi2_contingency(tshirts)
```

```
(11.56978992417547,  
 0.00901202511379703,  
 3,  
 array([[42.93103448, 30.0, 38.79310345, 38.27586207],  
        [40.06896552, 28.0, 36.20689655, 35.72413793]]))
```

Expected values

You can find the expected values at the forth in the returned value. It is in an array form. Let's print the expected values in a friendly way. We again use the Pandas dataframe. We are going to add index and column values and round the values to the two decimal places `round(2)`.

```
df=chi2_contingency(tshirts)[3]  
pd.DataFrame(  
    data=df[:,:],  
    index=["Male","Female"],  
    columns=["Black","White","Red","Blue"]  
).round(2)
```

	Black	White	Red	Blue
Male	42.93	30.0	38.79	38.28
Female	40.07	28.0	36.21	35.72

The above table is called a contingency table. You calculate the expected values from the observed data using the following equation.

$$\text{expected value} = \frac{\text{row total} \times \text{column total}}{\text{grand total}}$$

	Black	White	Red	Blue	Total
Male	$\frac{150 \times 83}{290} = 42.93$				150
Female					140
Total	83	58	75	74	290

χ^2 value

You can find the χ^2 value in the first returned value from `chi2_contingency`. But how do you find the χ^2 manually? The formula for the Chi-square is:

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

where O is the actual value and E is the expected value.

Chi-square formula

The χ^2 equation tells us to find the square of the difference between the actual value and expected value and divide it by the expected value. Then add all together to find the χ^2 value.

$$\frac{(48 - 42.93)^2}{42.93} + \frac{(22 - 30)^2}{30} + \frac{(33 - 38.79)^2}{38.79} + \frac{(47 - 38.28)^2}{38.28} + \frac{(35 - 40.07)^2}{40.07} \\ + \frac{(36 - 28)^2}{28} + \frac{(42 - 36.21)^2}{36.21} + \frac{(27 - 35.725)^2}{35.72}$$

Manual calculation of Chi-square

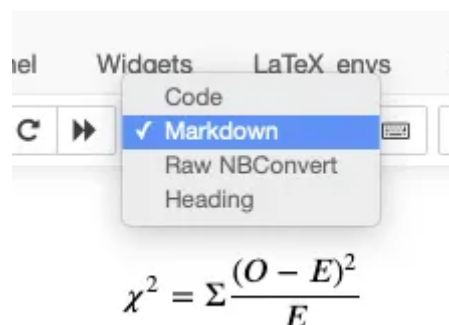
This is what `chi2_contingency` is doing behind the scene. Since Python is 0 based index, in order to print the χ^2 we need to use `[0]` which is the first value.

```
chisquare=chi2_contingency(tshirts)[0]
chisquare
```

11.56978992417547

Side note about Latex

I used Latex, pronounce 'lah-teck' to write the above equation in Jupyter Notebook. The cell you are writing must be Markdown and this is what you need to type in the cell.



```
\begin{equation}
\chi^2=\Sigma\frac{(O-E)^2}{E} \\
\text{where } O \text{ is the actual value and } E \text{ is the expected value.}
\end{equation}
```

p-value

You can find the p-value at the second in the returned value. p-value measures the probability of seeing the effect when the null hypothesis is true. So, when the p-value is low enough, we reject the null hypothesis and conclude the observed effect holds. We will talk about the null hypothesis later in this article.

- <https://www.machinelearningplus.com/statistics/p-value/>
- <https://statisticsbyjim.com/hypothesis-testing/hypothesis-tests-significance-levels-alpha-p-values/>
- <https://www.analyticsvidhya.com/blog/2015/09/hypothesis-testing-explaine>



```
pvalue=chi2_contingency(tshirts)[1]  
pvalue
```

0.00901202511379703

Degree of freedom

You can find the degree of freedom in the third returned value. We are going to use this to find the critical value later. The way you find the degree of freedom (dof) for χ^2 for independence is different from χ^2 Goodness of fit.

For χ^2 for independence:

$$\text{dof} = (\text{the number of rows} - 1) \times (\text{the number of columns} - 1)$$

For example, if your data has 4 rows x 3 columns, then the degree of freedom is:

$$\text{dof} = (4 - 1) \times (3 - 1) = 6$$

For χ^2 Goodness of fit, the categorical data has one dimension. And the degrees of freedom is:

$dof = (n - 1)$ where n is the number of categories that the variable is divided into.

In the returned value from the `chi2_contingency`, the third one is the degree of freedom. We use `[2]` which is the third one. The following will output 3.

```
dof=chi2_contingency(tshirts)[2]  
dof
```



The Subtlety of Spearman's Rank Correlation Coefficient

Untold parts of monotonic relation

towardsdatascience.com

Importing data

Horizontal data


Generally, you want to import data from a file. The first CSV file has data horizontally. By using `pd.read_csv` the data automatically changed to a Pandas dataframe.

The CSV file has the following data.

```
gender,Black,White,Red,Blue  
Male,48,12,33,57  
Female,35,46,42,27
```

Let's store the data to a variable called `tshirtshor`. We add `index_col="gender"` to make the gender column as the index.

```
csvfile = 'https://raw.githubusercontent.com/shinokada/python-for-ib-diploma-mathematics/master/Data/tshirts-horizontal.csv'
tshirtshor = pd.read_csv(csvfile, index_col='gender')
tshirtshor
```



	Black	White	Red	Blue
gender				
Male	48	12	33	57
Female	35	46	42	27

We run `chi2_contingency` on the `tshirtshor`.

```
chi2_contingency(tshirtshor)
```

```
(33.76146477535758, 2.2247293911334693e-07, 3, array([[41.5, 29. , 37.5,
42. ],
[41.5, 29. , 37.5, 42. ]]))
```

Vertical data

We are going to use vertically laid out data. Let's store the data to a variable called `tshirtsver`.

```
csvfile2 = 'https://raw.githubusercontent.com/shinokada/python-for-ib-diploma-mathematics/master/Data/tshirts-vertical.csv'
tshirtsver = pd.read_csv(csvfile2, index_col='Color')
tshirtsver
```

	Male	Female
Color		
Black	48	35
White	12	46
Red	33	42
Blue	57	27

We run `chi2_contingency` on the `tshirtsver`. We get the same values as before except the expected values.

```
chi2_contingency(tshirtsver)
```

```
(33.76146477535758, 2.2247293911334693e-07, 3, array([[41.5, 41.5],  
          [29. , 29. ],  
          [37.5, 37.5],  
          [42. , 42. ]]))
```

Pandas.DataFrame.transpose()

If you prefer horizontal data to vertical data, you can transpose data from vertical to horizontal by using `Pandas.DataFrame.transpose()` or `T` for short.

```
tshirtsver.T
```

Color	Black	White	Red	Blue
Male	48	12	33	57
Female	35	46	42	27

Using `chi2_contingency()`.



```
chi2_contingency(tshirtsver.T)
```

```
(33.76146477535758, 2.2247293911334693e-07, 3, array([[41.5, 29. , 37.5, 42. ],  
          [41.5, 29. , 37.5, 42. ]]))
```

Modeling Functions

From Linear to Logistic regression

towardsdatascience.com

Critical values

The level of significance and degree of freedom can be used to find the critical value. As I mentioned before you can find the degree of freedom from the array. In order to find critical values, you need to import `chi2` from `scipy.state` and define probability from the level of significance, 1%, 5% 10%, etc.

```
from scipy.stats import chi2
significance = 0.01
p = 1 - significance
dof = chi2_contingency(tshirtshor)[2]
critical_value = chi2.ppf(p, dof)
critical_value
```

11.344866730144373

When the degree of freedom is 3 and at the 1% level of significance the critical value is about 11.34. You can confirm with this value using cdf. The following will output 0.99.

```
p = chi2.cdf(critical_value, dof)
p
```

The null and alternative hypotheses

Chi-square test requires to state the null hypothesis, H_0 , and the alternative hypothesis, H_1 . The null hypothesis is the statement that our two variables are independent. The alternative hypothesis is the statement that they are not independent.

H_0 : Two variables are independent. H_1 : Two variables are dependent.

```
subjects = pd.DataFrame(
    [
        [25,46,15],
        [15,44,15],
        [10,10,20]
    ],
    index=['Biology','Chemistry','Physics'],
    columns=['Math SL AA','Math SL AI','Math HL'])
subjects
```

	Math SL AA	Math SL AI	Math HL
Biology	25	46	15
Chemistry	15	44	15
Physics	10	10	20

If the calculated Chi-square is greater than the critical value we reject the null hypothesis.

```
chi, pval, dof, exp = chi2_contingency(subjects)
print('p-value is: ', pval)
significance = 0.05
p = 1 - significance
critical_value = chi2.ppf(p, dof)

print('chi=%.6f, critical value=%.6f\n' % (chi, critical_value))

if chi > critical_value:
    print("""At %.2f level of significance, we reject the null
    hypotheses and accept H1.
    They are not independent.""") % (significance))
else:
    print("""At %.2f level of significance, we accept the null
    hypotheses.
    They are independent.""") % (significance))
```

p-value is: 0.0004176680832291999
chi=20.392835, critical value=9.487729

At 0.05 level of significance, we reject the null hypotheses and accept H
1.
They are not independent.

Alternatively, we can compare the p-value and the level of significance. If $p\text{-value} < \text{the level of significance}$, we reject the null hypothesis.

```
chi, pval, dof, exp = chi2_contingency(subjects)
significance = 0.05
```

```

print('p-value=%.6f, significance=%.2f\n' % (pval, significance))

if pval < significance:
    print("""At %.2f level of significance, we reject the null
    hypotheses and accept H1.
    They are not independent.""" % (significance))
else:
    print("""At %.2f level of significance, we accept the null
    hypotheses.
    They are independent.""" % (significance))

```

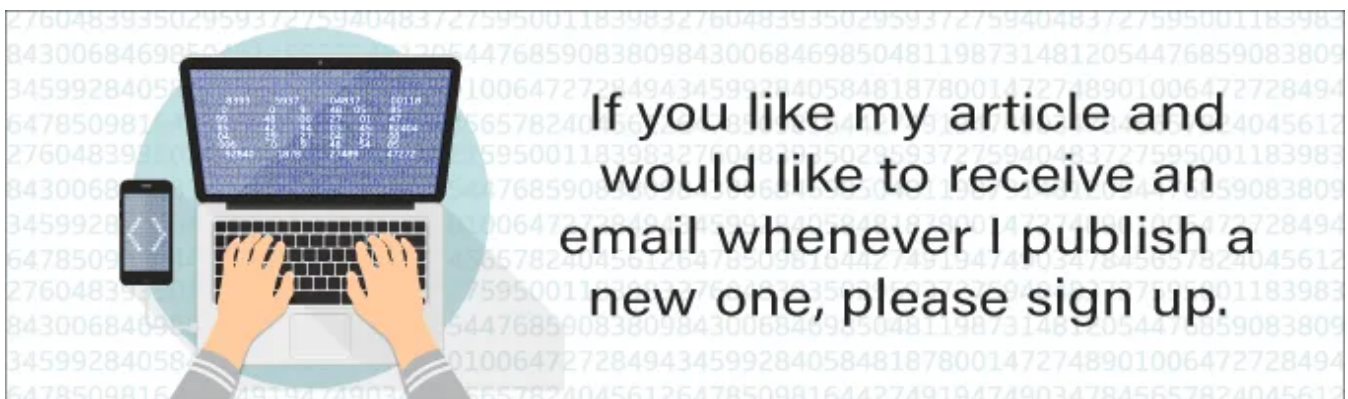
p-value=0.000418, significance=0.05

At 0.05 level of significance, we reject the null hypotheses and accept H1.
They are not independent.

Conclusion

In this article, I explained the basics of the Chi-square test using the Jupyter Notebook. The null and alternative hypotheses, expected values, Chi-square values, p-value, degree of freedom, and critical value are required for the Chi-square test for independence.

Get full access to every story on Medium by [becoming a member](#).



[Please subscribe.](#)

Reference

- <https://stats.stackexchange.com/questions/110718/chi-squared-test-with-scipy-whats-the-difference-between-chi2-contingency-and>
- <https://www.machinelearningplus.com/statistics/p-value/>

Exploring Normal Distribution with Jupyter Notebook

Beginners Guide to Normal Distribution using scipy and matplotlib

towardsdatascience.com

Data Science

Chi Square Test

Expected Value

P Value

Null Hypotheses



Follow

Written by Shinichi Okada

3.3K Followers · Writer for Towards Data Science

A programmer and technology enthusiast with a passion for sharing my knowledge and experience.

More from Shinichi Okada and Towards Data Science