

分布の検定と独立性の検定 (χ^2 検定)

青木繁伸

2020年3月17日

1 目的

いわゆる χ^2 検定 (分布の検定, 独立性の検定) を行う。

独立性の検定の場合には, 残差分析の結果 (残差および標準化残差) を返す。

2 使用法

```
import sys
sys.path.append("statlib")
from xtest import chisq_test
chisq_test(x, correct=True, p=None, verbose=True)
```

2.1 引数

<code>x</code>	独立性の検定の場合は 2 次元配列, 分布の検定の場合はベクトル
<code>correct</code>	独立性の検定で 2×2 分割表の場合にイエーツの補正を行う (デフォルトは <code>True</code>)。
<code>p</code>	分布の検定の場合に各カテゴリーの理論比。デフォルトは <code>None</code> で, その場合には各カテゴリーの理論比は等しいものと計算され, 一様性の検定を行う。
<code>verbose</code>	必要最小限のプリント出力をする (デフォルトは <code>True</code>)。

2.2 戻り値の名前

<code>"chisq"</code>	検定統計量 (χ^2 分布にしたがう)
<code>"df"</code>	自由度
<code>"pvalue"</code>	p 値
<code>"E"</code>	各セルの期待値
<code>"residuals"</code>	残差
<code>"stdres"</code>	標準化残差
<code>"method"</code>	検定手法名

3 使用例

3.1 独立性の検定

```
import sys
sys.path.append("statlib")
from xtest import chisq_test

x = [[22, 43, 54], [35, 21, 32]]
a = chisq_test(x)
```

```
Pearson's Chi-squared test
data:
[[22 43 54]
 [35 21 32]]
chisq = 11.777, df = 2, p value = 0.00277
```

3.2 残差分析

```
x = [[4,5,2,0], [0,7,6,1], [1,0,3,1]]
a = chisq_test(x)
```

```
Pearson's Chi-squared test
data:
[[4 5 2 0]
 [0 7 6 1]
 [1 0 3 1]]
chisq = 11.344, df = 6, p value = 0.07830
Chi-squared approximation may be incorrect
```

```
print(a["residuals"])
```

```
[[ 1.60018938  0.28603878 -1.01245685 -0.85634884]
 [-1.52752523  0.59160798  0.38251843  0.06900656]
 [ 0.18257419 -1.41421356  0.86164044  1.15470054]]
```

```
print(a["stdres"])
```

```
[[ 2.20265173  0.46401617 -1.59861608 -1.11382292]
 [-2.29128785  1.04582503  0.65816809  0.0978076 ]
 [ 0.21908902 -2.          1.18604324  1.30930734]]
```

3.3 2×2 分割表

3.3.1 イエーツの補正

2×2 分割表で、イエーツの補正を行う。

```
x = [[2, 4], [5, 2]]
a = chisq_test(x)
```

Pearson's Chi-squared test with Yates' continuity correction

data:

```
[[2 4]
```

```
[5 2]]
```

```
chisq = 0.66511, df = 1, p value = 0.41476
```

Chi-squared approximation may be incorrect

3.3.2 イエーツの補正なし

2×2 分割表で、イエーツの補正を行わない。

```
x = [[2, 4], [5, 2]]
a = chisq_test(x, correct=False)
```

Pearson's Chi-squared test

data:

```
[[2 4]
```

```
[5 2]]
```

```
chisq = 1.8866, df = 1, p value = 0.16958
```

Chi-squared approximation may be incorrect

3.4 分布の検定

3.4.1 一様性の検定

一様性の検定（理論比が等しい）を行う。

```
x = [1, 3, 4]
a = chisq_test(x)
```

Chi-squared test for given probabilities

data: [1 3 4]

```
chisq = 1.75, df = 2, p value = 0.41686
```

Chi-squared approximation may be incorrect

3.4.2 理論比を与える場合

理論比を与えて分布の検定を行う。

```
x = [1, 3, 4]
a = chisq_test(x, p=[0.25, 0.25, 0.50])
```

```
Chi-squared test for given probabilities
data: [1 3 4]
chisq = 1, df = 2, p value = 0.60653
Chi-squared approximation may be incorrect
```

4 既存の Python 関数との比較

4.1 分布の検定 `scipy.stats.chisquare()`

4.1.1 理論比を与えない場合

理論比を与えない場合は、各カテゴリーの度数が等しいかどうかの検定（一様性の検定）になる。

```
f = [29, 11, 15, 13, 21, 11]
from scipy.stats import chisquare
chisquare(f)
```

```
Power_divergenceResult(statistic=15.08, pvalue=0.01002592409077502)
```

```
a = chisq_test(f)
```

```
Chi-squared test for given probabilities
data: [29 11 15 13 21 11]
chisq = 15.08, df = 5, p value = 0.01003
```

```
print("chisq =", a["chisq"])
```

```
chisq = 15.080000000000005
```

```
print("p value =", a["pvalue"])
```

```
p value = 0.010025924090774997
```

4.1.2 理論比を与える場合 `scipy.stats.chisquare()`

`scipy.stats.chisquare()` は理論比ではなく、期待値を与える。

```
f = [114, 42, 35, 9]
from scipy.stats import chisquare
import numpy as np

chisquare(f, f_exp=200*np.array([9, 3, 3, 1])/16)
```

```
Power_divergenceResult(statistic=1.7066666666666666, pvalue=0.6354526615164937)
```

```
a = chisq_test(f, p=[9, 3, 3, 1])
```

```
Chi-squared test for given probabilities
data: [114 42 35 9]
chisq = 1.7067, df = 3, p value = 0.63545
```

```
print("chisq =", a["chisq"])
```

```
chisq = 1.7066666666666666
```

```
print("p value =", a["pvalue"])
```

```
p value = 0.6354526615164937
```

4.2 独立性の検定 `scipy.stats.chi2_contingency()`

4.2.1 通常の場合

```
x = [[2, 5, 7, 10], [8, 4, 6, 1]]
from scipy.stats import chi2_contingency
```

```
chi2_contingency(x)
```

```
(10.715152878310773,
 0.013370216505194306,
 3,
 array([[5.58139535, 5.02325581, 7.25581395, 6.13953488],
        [4.41860465, 3.97674419, 5.74418605, 4.86046512]]))
```

```
a = chi2_test(x)
```

```
Pearson's Chi-squared test
```

```
data:
```

```
[[ 2  5  7 10]
```

```
 [ 8  4  6  1]]
```

```
chisq = 10.715, df = 3, p value = 0.01337
```

```
Chi-squared approximation may be incorrect
```

```
print("chisq =", a["chisq"])
```

```
chisq = 10.715152878310773
```

```
print("df =", a["df"])
```

```
df = 3
```

```
print("p value =", a["pvalue"])
```

```
p value = 0.013370216505194306
```

```
print("expectation\n", a["E"], sep="")
```

```
expectation
```

```
[[5.58139535 5.02325581 7.25581395 6.13953488]
```

```
 [4.41860465 3.97674419 5.74418605 4.86046512]]
```

4.2.2 対数尤度比検定 (G^2 検定)

引数で `lambda="log-likelihood"` を指定すると、対数尤度比に基づいた、いわゆる G^2 検定を行うことができる。

対応する関数は `statlib.xtest.G_squared.test()` である。

```
g, p, dof, expctd = chi2_contingency(x, lambda_="log-likelihood")
print("g =", g)
```

```
g = 12.007766316932514
```

```
print("p =", p)
```

```
p = 0.0073566036791189605
```

```
from xtest import G_squared_test
a = G_squared_test(x)
```

```
G-squared test
```

```
chisq = 12.008, df = 3, p value = 0.00736
```

```
print("G2 =", a["G2"])
```

```
G2 = 12.007766316932532
```

```
print("p value =", a["pvalue"])
```

```
p value = 0.00735660367911889
```