

フリードマン検定

青木繁伸

1 目的

フリードマン検定に引き続いて、多重比較（対比較）を行う。
Rには、`friedman.test`関数が用意されている。

2 使用法

```
from friedman_test import friedman_test
friedman_test(y, groups=None, blocks=None, correction=False,
              multiple_comparison=False, verbose=True)
```

2.1 引数

<code>y</code>	2次元配列, またはデータベクトル
<code>groups</code>	処理 (列)
<code>blocks</code>	被検者 (行)
<code>correction</code>	同順位補正 (デフォルトは補正しない:False)
<code>multiple_comparison</code>	多重比較 (デフォルトは多重比較をしない:False)
<code>verbose</code>	必要最小限のプリント出力をする (デフォルトは True)

2.2 戻り値の名前

<code>"statistic"</code>	検定統計量 (χ^2 分布にしたがう)
<code>"df"</code>	χ^2 分布の自由度
<code>"p value"</code>	p 値
<code>"mc"</code>	多重比較の結果
<code>"y"</code>	2次元配列のデータ
<code>"rank"</code>	2次元配列データの順位
<code>"method"</code>	検定手法名

3 使用例

以下は、Rの`friedman.test()`のexampleに使われているテストデータである。
行は被検者 n , 列は処理 k である。つまり、ひとりの被検者が複数の条件下で測定される。

```
RoundingTimes = [
  [5.40, 5.50, 5.55],
  [5.85, 5.70, 5.75],
```

```
[5.20, 5.60, 5.50],
[5.55, 5.50, 5.40],
[5.90, 5.85, 5.70],
[5.45, 5.55, 5.60],
[5.40, 5.40, 5.35],
[5.45, 5.50, 5.35],
[5.25, 5.15, 5.00],
[5.85, 5.80, 5.70],
[5.25, 5.20, 5.10],
[5.65, 5.55, 5.45],
[5.60, 5.35, 5.45],
[5.05, 5.00, 4.95],
[5.50, 5.50, 5.40],
[5.45, 5.55, 5.50],
[5.55, 5.55, 5.35],
[5.45, 5.50, 5.55],
[5.50, 5.45, 5.25],
[5.65, 5.60, 5.40],
[5.70, 5.65, 5.55],
[6.30, 6.30, 6.25]
]
```

3.1 同順位の補正をしない場合

多くの教科書では、同順位の補正についての記述がない。

```
import sys
sys.path.append("statlib")
from Friedman_test import Friedman_test

a = Friedman_test(RoundingTimes)
```

Friedman rank sum test

Friedman chi-squared = 10.63636, df = 2, p-value = 0.0049

3.2 同順位の補正をする場合

R などでは、同順位の補正を行うのがデフォルトである。

同順位の補正をしない統計量を S 、第 i 行の同順位の数 k_i のベクトルを t_{ij} としたとき、 $T = \sum_{i=1}^n \sum_{j=1}^k t_{ij}^3 - t_{ij}$ を求める。修正項は $C = 1 - T/(n(k^3 - k))$ となり、修正後の統計量は $S' = S/C$ で求められる。同順位が全くない場合は $C = 1$ となるので $S' = S$ である。

```
a = Friedman_test(RoundingTimes, correction=True)
```

Friedman rank sum test

Friedman chi-squared = 11.14286, df = 2, p-value = 0.0038

3.3 データを3つのベクトルで指定する場合

R では、測定値と被検者と条件を別のベクトルで指定することにも対応している。第1引数は測定値のベク

トル, groups が処理, blocks が被検者を表す。

```
x = [44.55555555555556, 28.22222222222222, 24, 28.77777777777778,
      24.55555555555556,
      18.77777777777778]
w = ["A", "B", "A", "B", "A", "B"]
t = ["L", "L", "M", "M", "H", "H"]
a = Friedman_test(x, groups=w, blocks=t)
a["y"]
```

```
Friedman rank sum test
Friedman chi-squared = 0.33333, df = 1, p-value = 0.5637

array([[24.55555556, 18.77777778],
       [44.55555556, 28.22222222],
       [24.          , 28.77777778]])
```

3.4 多重比較をする場合

multiple_comparison=True とすれば, 全ての2種の処理の組み合わせでの検定を行う。

```
x = [[5, 60, 35, 62, 76],
      [24, 44, 74, 63, 76],
      [56, 57, 70, 74, 79],
      [44, 51, 55, 23, 84],
      [ 8, 68, 50, 24, 64],
      [32, 66, 45, 63, 46],
      [25, 38, 70, 58, 77],
      [48, 24, 40, 80, 72]]
a = Friedman_test(x, multiple_comparison=True)
```

```
Friedman rank sum test
Friedman chi-squared = 15.900000, df = 4, p-value = 0.0032

contrast      S  p value
1: 2    3.60000  0.46284
1: 3    4.22500  0.37641
1: 4    5.62500  0.22896
1: 5   15.62500  0.00357
2: 3    0.02500  0.99992
2: 4    0.22500  0.99413
2: 5    4.22500  0.37641
3: 4    0.10000  0.99879
3: 5    3.60000  0.46284
4: 5    2.50000  0.64464
```

3.5 scipy.stats.friedmanchisquare() の使用法

行が被検者, 列が処理である以下のようなデータとする。

```
[[2, 4, 5],
```

```
[3, 3, 5],  
[4, 5, 6],  
[3, 4, 6],  
[6, 7, 7]]
```

`scipy.stats.friedmanchisquare()` では、処理ごとの列ベクトルで指定する。

```
from scipy.stats import friedmanchisquare  
friedmanchisquare([2, 3, 4, 3, 6], [4, 3, 5, 4, 7], [5, 5, 6, 6, 7])
```

```
FriedmanchisquareResult(statistic=9.000000000000009, pvalue=0.011108996538242256)
```

`scipy.stats.friedmanchisquare()` は同順位補正をするので、`friedman.test()` では、`correction=True` を指定すれば、同じ結果になる。

```
import scipy as sp  
  
a = Friedman_test(sp.array([[2, 4, 5], [3, 3, 5], [4, 5, 6], [3, 4,  
6], [6, 7, 7]]), correction=True)
```

```
Friedman rank sum test
```

```
Friedman chi-squared = 9.00000, df = 2, p-value = 0.0111
```