

Fleiss の κ 統計量

青木繁伸

2020年3月17日

1 目的

対象者 n 人に対して 評定者 m 人で、それぞれの評定者は k 個のカテゴリーのいずれかに評定するという状況での一貫性に対する Fleiss の κ 統計量を求める。また、 k 個のカテゴリーごとの κ も求めることができる。

2 使用法

```
import sys
sys.path.append("statlib")
from xtest import kappa_m
kappa_m(r, exact=False, detail=True, verbose=True)
```

2.1 引数

<code>r</code>	関数に与えるのは、 $n \times m$ の二次元配列または二重リストである。
<code>nm</code>	デフォルトは <code>True</code> 。関数に $n \times k$ の二次元配列または二重リストを与えるときは <code>nm=False</code> にすること。
<code>Fleiss</code>	デフォルトでは <code>True</code> で、Fleiss による κ を求める。 Conger による正確な κ を求めるとき <code>False</code> にする。ただしその場合には、正確な κ しか表示されない。
<code>each</code>	<code>True</code> のとき、 <code>verbose=True</code> ならば、カテゴリーごとの κ も表示する。表示しなくても計算され、戻り値としても常に返される。
<code>verbose</code>	必要最小限のプリント出力をする

2.2 戻り値の名前

<code>"n"</code>	対象者の数
<code>"m"</code>	評定者の数
<code>"method"</code>	手法名
<code>"kappa"</code>	κ
<code>"se"</code>	κ の標準誤差
<code>"Z"</code>	検定統計量 (標準正規分布にしたがう)

"pvalue"	p 値
"results"	カテゴリごとの κ
"nk"	元データの $n \times m$ 行列を集計した $n \times k$ 行列
"mk"	元データの $n \times m$ 行列を集計した $m \times k$ 行列

3 使用例

データは数値でなくともよいが、順序は辞書順となるので、意図する順序と異なることがある。

また、一部の解説では $n \times m$ 行列ではなく、それを集計した $n \times k$ 行列から始めているものがあるので混同しないように（戻り値には、 $n \times k$ 行列を含んでいる）。

3.1 $n \times m$ 行列から始める場合

```
r = ["Neu", "Neu", "Neu", "Neu", "Neu", "Neu"],
     ["Per", "Per", "Per", "Oth", "Oth", "Oth"],
     ["Per", "Sch", "Sch", "Sch", "Sch", "Oth"],
     ["Oth", "Oth", "Oth", "Oth", "Oth", "Oth"],
     ["Per", "Per", "Per", "Neu", "Neu", "Neu"],
     ["Dep", "Dep", "Sch", "Sch", "Sch", "Sch"],
     ["Sch", "Sch", "Sch", "Sch", "Oth", "Oth"],
     ["Dep", "Dep", "Sch", "Sch", "Sch", "Neu"],
     ["Dep", "Dep", "Neu", "Neu", "Neu", "Neu"],
     ["Oth", "Oth", "Oth", "Oth", "Oth", "Oth"],
     ["Dep", "Neu", "Neu", "Neu", "Neu", "Neu"],
     ["Dep", "Per", "Neu", "Neu", "Neu", "Neu"],
     ["Per", "Per", "Per", "Sch", "Sch", "Sch"],
     ["Dep", "Neu", "Neu", "Neu", "Neu", "Neu"],
     ["Per", "Per", "Neu", "Neu", "Neu", "Oth"],
     ["Sch", "Sch", "Sch", "Sch", "Sch", "Oth"],
     ["Dep", "Dep", "Dep", "Neu", "Oth", "Oth"],
     ["Dep", "Dep", "Dep", "Dep", "Dep", "Per"],
     ["Per", "Per", "Neu", "Neu", "Neu", "Neu"],
     ["Dep", "Sch", "Sch", "Oth", "Oth", "Oth"],
     ["Oth", "Oth", "Oth", "Oth", "Oth", "Oth"],
     ["Per", "Neu", "Neu", "Neu", "Neu", "Neu"],
     ["Per", "Per", "Neu", "Oth", "Oth", "Oth"],
     ["Dep", "Dep", "Neu", "Neu", "Neu", "Neu"],
     ["Dep", "Neu", "Neu", "Neu", "Neu", "Oth"],
     ["Per", "Per", "Per", "Per", "Per", "Neu"],
     ["Dep", "Dep", "Dep", "Dep", "Oth", "Oth"],
     ["Per", "Per", "Neu", "Neu", "Neu", "Neu"],
     ["Dep", "Sch", "Sch", "Sch", "Sch", "Sch"],
     ["Oth", "Oth", "Oth", "Oth", "Oth", "Oth"]

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

```
from xtest import kappa_m

a = kappa_m(r)
```

```
Fleiss' Kappa: 6 Raters, 30 Subjects, 5 Categories
kappa = 0.43024, SE = 0.024374, Z = 17.652, p value < 0.0001
      Kappa      SE      Z      p value
Dep  0.244755  0.04714   5.192043 < 0.0001
Neu  0.471127  0.04714   9.994119 < 0.0001
Oth  0.566118  0.04714  12.009172 < 0.0001
Per  0.244755  0.04714   5.192043 < 0.0001
Sch  0.520000  0.04714  11.030866 < 0.0001
```

```
b = kappa_m(r, each=False)
```

```
Fleiss' Kappa: 6 Raters, 30 Subjects, 5 Categories
kappa = 0.43024, SE = 0.024374, Z = 17.652, p value < 0.0001
```

```
c = kappa_m(r, Fleiss=False)
```

```
Fleiss' Kappa: 6 Raters, 30 Subjects, 5 Categories (exact value)
kappa = 0.44181
```

3.2 $n \times k$ 行列から始める場合

以下は、対象者 (n) 10 人 (行数)、評定者 (m) 14 人 (行ごとの合計が全て同じ 14 になる)、カテゴリ数 (k) は 5 (列数) の場合のデータを集計した $n \times k$ 行列を与える場合である*¹。

```
nk = [[0,0,0,0,14],
      [0,2,6,4,2],
      [0,0,3,5,6],
      [0,3,9,2,0],
      [2,2,8,1,1],
      [7,7,0,0,0],
      [3,2,6,3,0],
      [2,5,3,2,2],
      [6,5,2,1,0],
      [0,2,2,3,7]]
d = kappa_m(nk, nm=False)
print(d["kappa"])
```

```
Fleiss' Kappa: 14 Raters, 10 Subjects, 5 Categories
kappa = 0.20993, SE = 0.016965, Z = 12.374, p value < 0.0001
      Kappa      SE      Z      p value
1  0.201282  0.03315   6.071916 < 0.0001
2  0.079670  0.03315   2.403352  0.0162
```

*¹ $n \times k$ から、元の $n \times m$ 行列を求めることはできない。

```

3 0.171598 0.03315 5.176450 < 0.0001
4 0.030381 0.03315 0.916491 0.3594
5 0.507657 0.03315 15.314077 < 0.0001
0.20993070442195527

```

3.1 節の使用例により戻される $n \times k$ 行列を使って再分析してみる。

```
e = kappa_m(a["nk"], nm=False)
```

```
Fleiss' Kappa: 6 Raters, 30 Subjects, 5 Categories
```

```
kappa = 0.43024, SE = 0.024374, Z = 17.652, p value < 0.0001
```

	Kappa	SE	Z	p value
1	0.244755	0.04714	5.192043	< 0.0001
2	0.471127	0.04714	9.994119	< 0.0001
3	0.566118	0.04714	12.009172	< 0.0001
4	0.244755	0.04714	5.192043	< 0.0001
5	0.520000	0.04714	11.030866	< 0.0001

同じ結果になる。