Fisher の exact test

青木繁伸

1　目的

Fisher の正確確率検定を行う。

R には fisher.test 関数がある（**Python** から R の fisher.test 関数を使用する方法については，第 4 節を参照)。

ここに取り上げる fisher 関数は，フィッシャー流の p 値の決め方 (fisher.test と同じ）の他に，ピアソン流の p 値の決め方も選べる。また，計算量が多量になり実用的な時間内に計算が終了できないような場合に対応するために，モンテカルロ法に基づく近似検定も選択できる（fisher.test の hybrid オプションと同じ)。

フィッシャー流の p 値の決め方とは，周辺和を固定した全ての分割表の生起確率をもとめ，実際に観察された分割表の生起確率より小さいか等しい分割表の生起確率を合計したものが p 値であるとするものである。

ピアソン流の p 値の決め方とは，周辺和を固定した全ての分割表においてピアソンのカイ二乗統計量と生起確率を求め，実際に観察された分割表のピアソンのカイ二乗統計より小さいか等しい分割表の生起確率を合計したものを p 値とするものである。

本来，Fisher の exact test の目的は，独立性の検定にあると思われ，独立性からのずれを評価する統計指標は様々ある。しかるに，フィッシャー流の p 値の決め方は，基盤とする統計量を用いていない。単に生起確率が観察された分割表の生起確率より小さいことのに基づくフィッシャー流の p 値の決め方は不適切であると考えるものである。

2　使用法

```
from Fisher_exact_test import Fisher_exact_test
fisher(x, y=NULL, exact=TRUE, method="Fisher", hybrid=FALSE, loop=10000, verbose=True)
```

2.1　引数

x	分割表（合計を含まない） もしくはベクトル
y	x がベクトルのときは，ベクトル x が分割表のときには無視される
exact	正確な p 値を求める場合，またはシミュレーションにより近似的な p 値を求めるときには True（デフォルト） False の場合にはカイ二乗分布による漸近近似検定のみを行う
method	p 値の決め方。"Fisher" か "Pearson" を指定する。省略した場合には "Fisher" が仮定される
hybrid	True を指定すれば，シミュレーションにより近似的な p 値を計算する
loop	シミュレーションの回数
verbose	必要最小限のプリント出力をする

2.2 戻り値の名前

"chisq"	χ^2 分布を利用する漸近検定統計量
"df"	自由度
"asymP"	漸近検定による p 値
"exactP"	正確な p 値
"simP"	シミュレーションによる p 値

3 使用例

```
x = [[5, 3, 2, 1],
     [4, 3, 5, 2],
     [2, 3, 1, 2]]
```

3.1 フィッシャーの方法による正確な p 値

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

a = Fisher_exact_test(x)
```

```
カイ二乗値 = 3.3963, 自由度 = 6, p 値 = 0.7577
Fisher の方法による, 正確な p 値 = 0.8091
査察した分割表の個数は 24871 個
```

モンテカルロ法による p 値

```
a = Fisher_exact_test(x, hybrid=True)
```

```
カイ二乗値 = 3.3963, 自由度 = 6, p 値 = 0.7577
10000 回のシミュレーション（Fisher の方法）による p 値 = 0.8042
```

3.2 ピアソンの方法による正確な p 値

```
a = Fisher_exact_test(x, method="Pearson")
```

```
カイ二乗値 = 3.3963, 自由度 = 6, p 値 = 0.7577
Pearson の方法による, 正確な p 値 = 0.7878
査察した分割表の個数は 24871 個
```

モンテカルロ法による p 値

```
a = Fisher_exact_test(x, method="Pearson", hybrid=True)
```

```
カイ二乗値 = 3.3963, 自由度 = 6, p 値 = 0.7577
10000 回のシミュレーション（Pearson の方法）による p 値 = 0.7883
```

3.3 2 変数を与える場合

```
x = [1, 2, 3, 2, 1, 2, 3, 2, 1, 2, 3, 2, 2, 3, 1]
y = ["A", "B", "C"]*5
a = Fisher_exact_test(x, y)
```

カイ二乗値 = 1.2857, **自由度** = 4, p **値** = 0.8638

Fisher **の方法による，正確な** p **値** = 1.0

査察した分割表の個数は 180 **個**

4 R の fisher.test() を使う方法

下準備として，rpy2 をインストールしておく。

pip install rpy2

この後，以下のように実行する。

```
import scipy as sp
import rpy2.robjects.numpy2ri
from rpy2.robjects.packages import importr
rpy2.robjects.numpy2ri.activate()

stats = importr('stats')
m = sp.array([[4,4],[4,5],[10,6]]) # 対象とする分割表の定義
res = stats.fisher_test(m)         # fisher.test 起動
print('p-value: {}'.format(res[0][0])) # P 値は
```

p-value: 0.6643639131198411

```
print(res)
```

Fisher's Exact Test for Count Data

data: structure(c(4L, 4L, 10L, 4L, 5L, 6L), .Dim = 3:2)

p-value = 0.6644

alternative hypothesis: two.sided

```
print(type(res))
```

<class 'rpy2.robjects.vectors.ListVector'>

```
print(sp.array(res).shape)
```

(4, 1)

```
print(res[0][0])
```

0.6643639131198411

```
print(res[1][0])
```

two.sided

```
print(res[2][0])
```

```
Fisher's Exact Test for Count Data
```

```
print(res[3][0])
```

```
structure(c(4L, 4L, 10L, 4L, 5L, 6L), .Dim = 3:2)
```