二元配置分散分析

青木繁伸

1　目的

二元配置分散分析を行う。

2　使用法

twoway_anova(x, a, b, verbose=True)

2.1　引数

x	測定値
a	要因 A を表す変数
b	要因 B を表す変数
verbose	必要最小限のプリント出力をする

2.2　戻り値の名前

"result"	結果をデータフレームにして返す

3　使用例

3.1　繰り返し数が 1 の場合

```
import pandas as pd

a = [1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3]
b = [1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]
x = [9, 17, 12, 16, 1, 21, 16, 11, 7, 19, 6, 9]
df = pd.DataFrame({"x": x, "a": a, "b": b})

import sys
sys.path.append("statlib")
from twoway_anova import twoway_anova

ans = twoway_anova(df)
```

```
          SS  d.f.         MS   F value   p value
a   21.500000     2  10.750000  0.659284  0.551030
b  268.666667     3  89.555556  5.492334  0.037192
```

```
e   97.833333     6  16.305556       NaN       NaN
T  388.000000    11  35.272727       NaN       NaN
```

3.2 繰り返し数が周辺度数に比例する場合

```
import scipy as sp

# case 2-1
x = [24.8, 23.9, 24.1, 28.8, 22.6, 28.0, 26.4, 27.4, 29.4, 30.0, 28.7,
        29.2, 25.0, 26.6, 27.9, 28.5, 27.1, 25.2, 27.9, 29.2, 26.7,
           25.0,
        29.9, 29.4, 27.5, 32.5, 29.5, 26.3, 28.2, 31.8, 30.2, 31.7,
           29.2,
        30.3, 30.9, 28.4, 29.8, 26.7, 30.7, 28.5, 26.5, 26.7, 31.7,
           27.2,
        25.5, 29.9, 27.3, 28.8, 28.5, 25.7, 28.7, 31.3, 29.4, 29.8,
           30.3,
        29.6, 31.7, 33.6, 32.0, 34.3]
a = sp.ravel([sp.repeat(range(4), 3) for i in range(5)])
b = sp.repeat(range(5), 12)
df = pd.DataFrame({"x": x, "a": a, "b": b})

ans = twoway_anova(df)
```

母数モデル

```
             SS  d.f.         MS   F value   p value
a     51.397333     3  17.132444  4.937067  0.005204
b    106.287333     4  26.571833  7.657221  0.000111
a:b   52.852667    12   4.404389  1.269215  0.273893
e    138.806667    40   3.470167       NaN       NaN
T    349.344000    59   5.921085       NaN       NaN
```

変量モデル

```
             SS  d.f.         MS   F value   p value
a     51.397333     3  17.132444  3.889857  0.037391
b    106.287333     4  26.571833  6.033035  0.006719
a:b   52.852667    12   4.404389  1.269215  0.273893
e    138.806667    40   3.470167       NaN       NaN
T    349.344000    59   5.921085       NaN       NaN
```

```
# case 2-2
d = sp.array([  1, 1, 7.0, 1, 1, 5.7,
        1, 2, 4.6, 1, 2, 5.6,
        1, 3, 7.0, 1, 3, 7.4,
        1, 3, 6.6, 1, 4, 7.3,
        1, 4, 6.3, 1, 4, 7.8,
        1, 5, 5.3, 1, 5, 4.7,
        2, 1, 5.8, 2, 1, 8.9,
        2, 2, 10.0, 2, 2, 7.1,
```

```
        2, 3, 6.5, 2, 3, 9.8,
        2, 3, 5.0, 2, 4, 9.7,
        2, 4, 6.4, 2, 4, 9.9,
        2, 5, 7.8, 2, 5, 5.4,
        3, 1, 8.1, 3, 1, 6.1, 3, 1, 10.2, 3, 1, 6.9,
        3, 2, 5.2, 3, 2, 5.7, 3, 2, 11.8, 3, 2, 6.1,
        3, 3, 10.4, 3, 3, 5.6, 3, 3, 9.4, 3, 3, 5.9, 3, 3, 12.1, 3, 3,
            5.7,
        3, 4, 5.3, 3, 4, 9.7, 3, 4, 5.5, 3, 4, 10.2, 3, 4, 13.8, 3, 4,
            7.0,
        3, 5, 11.1, 3, 5, 4.2, 3, 5, 7.1, 3, 5, 5])
d = d.reshape((48, 3))
df = pd.DataFrame(d[:,[2,0,1]], columns=["x", "a", "b"])

ans = twoway_anova(df)
```

母数モデル

	SS	d.f.	MS	F value	p value
a	20.796875	2	10.398437	1.758991	0.187983
b	19.536458	4	4.884115	0.826193	0.518037
a:b	7.848958	8	0.981120	0.165965	0.994003
e	195.082500	33	5.911591	NaN	NaN
T	243.264792	47	5.175847	NaN	NaN

変量モデル

	SS	d.f.	MS	F value	p value
a	20.796875	2	10.398437	10.598540	0.005636
b	19.536458	4	4.884115	4.978102	0.026004
a:b	7.848958	8	0.981120	0.165965	0.994003
e	195.082500	33	5.911591	NaN	NaN
T	243.264792	47	5.175847	NaN	NaN

3.3 一般の場合

前述のどれでもない場合。

```
# case 3
a = [1, 1, 1, 1, 2, 2, 2, 2, 2, 2]
b = [1, 1, 2, 2, 1, 1, 2, 2, 2, 2]
x = [17, 16, 25, 22, 18, 26, 34, 30, 34, 30]
df = pd.DataFrame({"x": x, "a": a, "b": b})

ans = twoway_anova(df)
```

	SS	d.f.	MS	F value	p value
a	121.440476	1	121.440476	13.747978	0.009995
b	177.190476	1	177.190476	20.059299	0.004198
a:b	5.142857	1	5.142857	0.582210	0.474369
e	53.000000	6	8.833333	NaN	NaN
T	415.600000	9	46.177778	NaN	NaN

このプログラムでは Type II の平方和を使っているので，（結果表示中の順番は異なっても）要因の指定順序にによって結果が異なることはない。

```
df = pd.DataFrame({"x": x, "b": b, "a": a})

ans = twoway_anova(df)
```

```
             SS  d.f.          MS    F value   p value
b    177.190476     1  177.190476  20.059299  0.004198
a    121.440476     1  121.440476  13.747978  0.009995
b:a    5.142857     1    5.142857   0.582210  0.474369
e     53.000000     6    8.833333        NaN       NaN
T    415.600000     9   46.177778        NaN       NaN
```