

正準相関分析

青木繁伸

2020年3月17日

1 目的

正準相関分析を行う。

Rにも `cancor` 関数があるが、出力されるのは標準化されていない係数だけであり、しかもその係数は「データの個数から1引いたものの平方根」で割り算されているものである。

ここでは、係数および標準化された係数の両方を表示する関数を提示する。

2 使用法

```
import sys
sys.path.append("statlib")
from multi import cancor
cancor(x, gr1, gr2, verbose=True)
```

2.1 引数

| | |
|----------------------|-----------------|
| <code>x</code> | データフレーム |
| <code>gr1</code> | 第1変数グループのリスト |
| <code>gr2</code> | 第2変数グループのリスト |
| <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

p , q は第1変数グループ、第2変数グループに含まれる変数の個数、 k は p , q の小さい方の数、 n はサンプルサイズとして、以下のものが返される。

| | |
|-------------------------|---|
| <code>"ccc"</code> | k 個の正準相関係数 canonical correlation coefficients |
| <code>"std.coef"</code> | 標準化係数 standardized coefficients $p \times k$ と $q \times k$ の2つの行列 |
| <code>"coef"</code> | 標準化されていない係数 coefficients $p \times k$ と $q \times k$ の2つの行列 |
| <code>"cs"</code> | 正準得点 canonical score 第1, 第2変数グループそれぞれに対して $n \times k$ 行列1つずつ |

3 使用法

```
import pandas as pd

x = pd.DataFrame({
    "a": [2, 1, 0, -1, -2],
    "b": [1, 2, 0, -2, -1],
    "c": [2, -1, 0, -2, 1],
    "d": [2, -1, 0, 1, -2]})

import sys
sys.path.append("statlib")
from multi import cancor

a = cancor(x, ["a", "b"], ["c", "d"])
```

Canonical score of Group 1

| | Axis-1 | Axis-2 |
|---|-----------|-----------|
| 0 | -1.264911 | -0.632456 |
| 1 | 0.632456 | -1.264911 |
| 2 | 0.000000 | 0.000000 |
| 3 | -0.632456 | 1.264911 |
| 4 | 1.264911 | 0.632456 |

Canonical score of Group 2

| | Axis-1 | Axis-2 |
|---|---------------|-----------|
| 0 | 1.264911e+00 | -1.144155 |
| 1 | -6.324555e-01 | 0.572078 |
| 2 | -4.440892e-17 | 0.000000 |
| 3 | 6.324555e-01 | 1.334848 |
| 4 | -1.264911e+00 | -0.762770 |

正準相関係数

```
import numpy as np

print("canonical_correlation_coefficients")
np.around(a["ccc"], 7)
```

```
canonical_correlation_coefficients
array([1.          , 0.3015113])
```

標準化された係数

```
print("standardized coefficients")
np.around(a["stdcoef"][0], 7)
```

```
standardized coefficients
array([[ -1.6666667, -0.        ],
       [ 1.3333333, -1.        ]])
```

```
np.around(a["stdcoef"][1], 7)
```

```
array([[ -0.        , -1.0050378],
       [ 1.        ,  0.1005038]])
```

係数

```
print("coefficients")
np.around(a["coef"][0], 7)
```

```
coefficients
array([[ -1.0540926, -0.        ],
       [ 0.843274 , -0.6324555]])
```

```
np.around(a["coef"][1], 7)
```

```
array([[ -0.        , -0.6356417],
       [ 0.6324555,  0.0635642]])
```

正準得点

```
print("canonical scores")
np.around(a["cs"][0], 7)
```

```
canonical scores
array([[ -1.2649111, -0.6324555],
       [ 0.6324555, -1.2649111],
       [ 0.        ,  0.        ],
       [-0.6324555,  1.2649111],
       [ 1.2649111,  0.6324555]])
```

```
np.around(a["cs"][1], 7)
```

```
array([[ 1.2649111, -1.1441551],
       [-0.6324555,  0.5720776],
       [-0.        ,  0.        ],
       [ 0.6324555,  1.3348476],
       [-1.2649111, -0.7627701]])
```

4 使用例 2

```
x = pd.DataFrame({
    "a": [2, 1, 0, -1, -2, 3],
    "b": [1, 2, 0, -2, -1, 4],
    "c": [2, -1, 0, -2, 1, 2],
    "d": [2, -1, 0, 1, -2, 4],
    "e": [3, -3, 2, 3, -1, 1]})
a = cancor(x, ["a", "b"], ["c", "d", "e"])
```

Canonical score of Group 1

```
      Axis-1   Axis-2
0  0.262552  1.446307
1 -0.767001 -0.423280
2  0.300900 -0.070565
3  1.368802  0.282150
4  0.339249 -1.587437
5 -1.504502  0.352825
```

Canonical score of Group 2

```
      Axis-1   Axis-2
0  0.120758  0.756109
1 -0.758523 -1.191970
2  0.816415  0.017124
3  1.189761  0.627642
4  0.149976 -1.242753
5 -1.518388  1.033849
```

正準相関係数

```
print("canonical correlation coefficients")
np.around(a["ccc"], 7)
```

```
canonical correlation coefficients
array([0.9645994, 0.8223086])
```

標準化された係数

```
print("standardized coefficients")
```

```
standardized coefficients
```

```
np.around(a["stdcoef"][0], 7)
```

```
array([[ 0.6181246,  2.1118284],
       [-1.5103402, -1.6002405]])
```

```
np.around(a["stdcoef"][1], 7)
```

```
array([[ -0.2762663, -0.0804173],  
       [ -0.8474466,  0.7030491],  
       [ 1.026069 ,  0.4480469]])
```

係数

```
print("coefficients")  
np.around(a["coef"][0], 7)
```

```
coefficients  
array([[ 0.3304015,  1.1288198],  
       [-0.6991516, -0.7407674]])
```

```
np.around(a["coef"][1], 7)
```

```
array([[ -0.1691778, -0.0492453],  
       [-0.3922915,  0.3254485],  
       [ 0.4272815,  0.1865783]])
```

正準得点

```
print("canonical scores")  
np.around(a["cs"][0], 7)
```

```
canonical scores  
array([[ 0.2625517,  1.4463072],  
       [-0.7670014, -0.42328  ],  
       [ 0.3009003, -0.0705649],  
       [ 1.3688021,  0.2821502],  
       [ 0.3392489, -1.587437 ],  
       [-1.5045017,  0.3528246]])
```

```
np.around(a["cs"][1], 7)
```

```
array([[ 0.1207582,  0.7561087],  
       [-0.7585229, -1.1919704],  
       [ 0.8164154,  0.0171241],  
       [ 1.1897611,  0.6276416],  
       [ 0.149976 , -1.242753 ],  
       [-1.5183879,  1.0338491]])
```