

Wilcoxon 検定

青木繁伸

2020年3月17日

1 目的

ウィルコクソン検定を行う。

2 使用法

```
import sys
sys.path.append("statlib")
from xtest import Wilcoxon_test
Wilcoxon_test(x, y=None, alternative="two_sided", mu=0, paired=False,
              correct=True, verbose=True)
```

独立2標本の場合のみ、二次データ（分割表）を対象として検定を行うこともできる。

```
import sys
sys.path.append("statlib")
from xtest import Wilcoxon_test2
Wilcoxon_test2(d, alternative="two_sided", mu=0, correct=True, verbose=True)
```

2.1 引数

<code>x</code>	データベクトル
<code>y</code>	データベクトル
<code>d</code>	$2 \times m$ 行列
<code>alternative</code>	対立仮説: <code>two_sided</code> (デフォルト), <code>less</code> , <code>greater</code>
<code>mu</code>	対応のある検定の場合の母平均 (デフォルトは 0)
<code>paired</code>	対応のあるデータの場合に <code>True</code> を指定する (デフォルトは <code>False</code>)
<code>var_equal</code>	等分散性を仮定するかどうか (デフォルトは <code>False</code>)
<code>confllevel</code>	信頼区間を求めるときの信頼率 (デフォルトは 0.95)
<code>verbose</code>	必要最小限のプリント出力をする (デフォルトは <code>True</code>)

2.2 戻り値の名前

"W"	検定統計量
"E"	検定統計量の平均値
"V"	検定統計量の分散
"Z"	検定統計量の標準化得点 Z
"pvalue"	p 値
"alternative"	対立仮説の種別
"method"	検定手法名

3 使用例

3.1 独立 2 標本の場合

```
import sys
sys.path.append("statlib")
from xtest import Wilcoxon_test

x = [3, 2, 1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1]
y = [3, 2, 4, 5, 6, 7, 4, 5, 3, 4, 5, 6, 8]

a = Wilcoxon_test(x, y)
```

```
Wilcoxon rank sum test with continuity correction
W = 41.5, E(W) = 84.5, V(W) = 370.5, Z = -2.208, p value = 0.02725
alternative hypothesis: true location shift is not equal to 0
```

上記の x と y は、以下のようにして分割表 d として表現できる。

```
import pandas as pd
import numpy as np

df = pd.DataFrame({"g": np.repeat([1,2], len(x)), "xy": x + y})
d = pd.crosstab(df["g"], df["xy"])
print(d)
```

```
xy  1  2  3  4  5  6  7  8
g
1   2  3  3  2  2  1  0  0
2   0  1  2  3  3  2  1  1
```

元のデータがなくても、集計表があれば検定を行うことができ、元データでの検定と同じ結果になる。

```
d = [[2, 3, 3, 2, 2, 1, 0, 0],
      [0, 1, 2, 3, 3, 2, 1, 1]]
```

```
from xtest import Wilcoxon_test2
```

```
b = Wilcoxon_test2(d)
```

Wilcoxon rank sum test with continuity correction

W = 41.5, E(W) = 84.5, V(W) = 370.5, Z = -2.208, p value = 0.02725

alternative hypothesis: true location shift is not equal to 0

3.2 関連のある 2 標本の場合

```
v = [0.8, 0.7, 0.8, 1.1, 0.8, 1.0, 0.4, 0.6, 1.0, 0.9, 0.7, 0.7, 0.7,  
     0.6, 0.9, 1.0, 0.9, 0.4, 1.1, 1.0]
```

```
w = [0.8, 0.6, 0.9, 1.0, 1.0, 1.1, 0.5, 0.5, 0.9, 1.0, 0.5, 0.6, 0.8,  
     0.6, 0.8, 0.9, 0.8, 0.7, 1.1, 1.1]
```

```
c = Wilcoxon_test(v, w, paired = True)
```

Wilcoxon signed rank test with continuity correction

W = 75.5, E(W) = 76.5, V(W) = 389.25, Z = -0.025343, p value = 0.97978

alternative hypothesis: true location shift is not equal to 0

3.3 1 標本の場合

関連のある 2 標本の元データがあるとき、それぞれのデータの差を取る意味があるならば、差を取ったデータを検定対象として同じ結果が得られる。

元のデータがない場合とか、差しか観察できないような場合に適用する手法である。

```
import numpy as np
```

```
diff = np.array(v) - np.array(w)  
print(diff)
```

```
[ 0.   0.1 -0.1  0.1 -0.2 -0.1 -0.1  0.1  0.1 -0.1  0.2  0.1 -0.1  0.  
  0.1  0.1  0.1 -0.3  0.  -0.1]
```

```
c = Wilcoxon_test(diff)
```

Wilcoxon signed rank test with continuity correction

W = 66.5, E(W) = 76.5, V(W) = 424.25, Z = -0.46122, p value = 0.64464

alternative hypothesis: true location is not equal to 0

ところが、得られた結果が異なっている。この原因は、コンピュータ内で保持されるデータの精度の問題に起因する。

元のデータを（この場合は）10 倍して整数値として記述・分析し直してみる。

```
v10 = [8, 7, 8, 11, 8, 10, 4, 6, 10, 9, 7, 7, 7, 6, 9, 10, 9, 4, 11,
      10]
w10 = [8, 6, 9, 10, 10, 11, 5, 5, 9, 10, 5, 6, 8, 6, 8, 9, 8, 7, 11,
      11]
diff10 = np.array(v10) - np.array(w10)
c10 = Wilcoxon_test(v10, w10, paired = True)
```

Wilcoxon signed rank test with continuity correction

W = 75.5, E(W) = 76.5, V(W) = 389.25, Z = -0.025343, p value = 0.97978
alternative hypothesis: true location shift is not equal to 0

関連のある2標本として検定したときと同じ結果になる。

また、コンピュータ上で差を取って、そのまま `Wilcoxon.test2()` にあたえるのではなく、上記 `print(diff)` で表示された値を再入力すれば、同じ結果になる。

```
diff2 = [ 0.0,  0.1, -0.1,  0.1, -0.2, -0.1, -0.1,  0.1,  0.1, -0.1,
         0.2,
         0.1, -0.1,  0.0,  0.1,  0.1,  0.1, -0.3,  0.0, -0.1]
c2 = Wilcoxon_test(diff2)
```

Wilcoxon signed rank test with continuity correction

W = 75.5, E(W) = 76.5, V(W) = 389.25, Z = -0.025343, p value = 0.97978
alternative hypothesis: true location is not equal to 0

`diff` と `diff2` は見た目は全く同じであるが、コンピュータ内部では異なるというのが検定結果に違いが出る原因である。

実は、関連のある2標本の場合として `Wilcoxon.test()` を使用しても、与えられた2引数 `x`, `y` を整数化した後で差を取ってから検定を行っている^{*1}。

4 既存の Python 関数との比較 `scipy.stats.wilcoxon()`

4.1 `scipy.stats.wilcoxon()`

`scipy.stats.wilcoxon()` は対応のある2標本の検定である。`statlib.xtest.Wilcoxon.test()` では、`paired=True` を指定した場合に相当する。

ただし、`statlib.xtest.Wilcoxon.test()` では `correction=False` がデフォルトであり、差が0になるペアの扱いとしてデフォルトの `zero.method = "wilcox"` の他に2方法がある。

```
from scipy.stats import wilcoxon
x = [3,2,1,2,3,4,6,3,4,5,6,7,5,6,8]
y = [2,4,5,2,4,5,6,5,4,3,6,2,4,6,1]

from scipy.stats import wilcoxon
wilcoxon(x, y, correction=True)
```

^{*1} R の `wilcox.test()` ではこのような事前処理をしていないので、誤った検定結果が表示される。

```
WilcoxonResult(statistic=25.0, pvalue=0.8369957300480487)
```

```
wilcoxon(x, y, zero_method="wilcox", correction=True)
```

```
WilcoxonResult(statistic=25.0, pvalue=0.8369957300480487)
```

```
wilcoxon(x, y, zero_method="pratt", correction=True)
```

```
WilcoxonResult(statistic=50.0, pvalue=0.9072219640301218)
```

```
wilcoxon(x, y, zero_method="zsplit", correction=True)
```

```
WilcoxonResult(statistic=57.5, pvalue=0.9089372626959147)
```

```
a = Wilcoxon_test(x, y, paired=True)
```

Wilcoxon signed rank test with continuity correction

$\bar{W} = 30$, $E(W) = 27.5$, $V(W) = 94.5$, $Z = 0.20574$, p value = 0.83700

alternative hypothesis: true location shift is not equal to 0

```
print("W =", a["W"])
```

```
W = 30.0
```

```
print("p value =", a["pvalue"])
```

```
p value = 0.8369957300480487
```

4.2 scipy.stats.mannwhitneyu()

`scipy.stats.mannwhitneyu()` は対応のない 2 標本の検定である。両側検定を行うときにも、明示的に `alternative="two-sided"` と指定しなければならない。

```
u = [2, 1, 2, 2, 3, 4, 5, 4, 4, 3, 5, 6, 7, 4, 3, 5]
```

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

```
from scipy.stats import mannwhitneyu  
mannwhitneyu(u, v, alternative="two-sided")
```

```
MannwhitneyUResult(statistic=183.0, pvalue=0.03421908780745614)
```

```
a = Wilcoxon_test(u, v)
```

Wilcoxon rank sum test with continuity correction

$W = 183$, $E(W) = 128$, $V(W) = 662.45$, $Z = 2.1175$, p value = 0.03422

alternative hypothesis: true location shift is not equal to 0

```
print(a["W"], a["pvalue"])
```

```
183.0 0.03421908780745614
```

4.3 `scipy.stats.ranksums()`

`scipy.stats.ranksums()` は `scipy.stats.mannwhitneyu()` よりもっとやっかいな存在である。`scipy.stats.mannwhitneyu()` は同順位補正を行わないし、連続性の補正もしない。マニュアルで `For tie-handling and an optional continuity correction see `scipy.stats.mannwhitneyu`.` とあるが、ひどいものである。

上と同じデータを分析してみる。

```
from scipy.stats import ranksums
ranksums(u, v)
```

```
RanksumsResult(statistic=2.072890493972125, pvalue=0.038182471913683506)
```

マニュアルに `This test should be used to compare two samples from continuous distributions.` とあるのは、`It does not handle ties between measurements in x and y.` という居直りを正当化するためであろう。