

# Mood 検定

青木繁伸

2019 年 3 月 7 日

## 1 目的

独立二標本の散布度に差があるか、Mood 検定を行う。

## 2 使用法

```
import sys
sys.path.append("statlib")
from xttest import Mood_test
Mood_test(x, y, alternative="two_sided", verbose=True)
```

### 2.1 引数

x	データベクトル (リストでもよい)
y	データベクトル (リストでもよい)
alternative	対立仮説 (デフォルトは "two_sided"。その他に "less", "greater" を指定することができる)
verbose	必要最小限のプリント出力をする

### 2.2 戻り値の名前

"z"	検定統計量 (標準正規分布にしたがう)
"pvalue"	$p$ 値
"alternative"	対立仮説
"method"	検定手法名

## 3 使用例

```
x = [111, 107, 100, 99, 102, 106, 109, 108, 104, 99,
     101, 96, 97, 102, 107, 113, 116, 113, 110, 98]
y = [107, 108, 106, 98, 105, 103, 110, 105, 104,
     100, 96, 108, 103, 104, 114, 114, 113, 108, 106, 99]

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

```
from xtest import Mood_test

a = Mood_test(x, y)
```

```
Mood two-sample test of scale
Z = 1.0371, p value = 0.29968
alternative hypothesis: two_sided
```

```
b = Mood_test(x, y, alternative="less")
```

```
Mood two-sample test of scale
Z = 1.0371, p value = 0.85016
alternative hypothesis: less
```

```
c = Mood_test(x, y, alternative="greater")
```

```
Mood two-sample test of scale
Z = 1.0371, p value = 0.14984
alternative hypothesis: greater
```

```
d = Mood_test([1, 3, 5, 7, 9], [2, 4, 6, 8])
```

```
Mood two-sample test of scale
Z = 0.72075, p value = 0.47106
alternative hypothesis: two_sided
```

## 4 既存の Python 関数との比較 scipy.stats.mood()

Python にも Mood 検定を行う mood() がある。しかし、データに同順位がある場合を考慮していないので、同順位のある場合には適切とはいえない。また、片側検定にも対応していない。

上の使用例では、x, y は同順位を含むので、Mood\_test() の結果とは一致しない。

```
from scipy.stats import mood
mood(x, y)
```

```
(0.9919203990624894, 0.32123635731324385)
```

以下のデータ例においては同順位を含まないので、Mood\_test() の結果と一致する。

```
mood([1, 3, 5, 7, 9], [2, 4, 6, 8])
```

```
(0.7207499701564469, 0.47106336173321905)
```