

# 平均値の検定 ( $t$ 検定)

青木繁伸

2020年3月17日

## 1 目的

1 標本の場合に母平均の検定, 対応のある 2 標本の平均値の差の検定, 独立 2 標本の場合の平均値の差の検定を行う。

2 群の平均値の差の検定では, R と同じくデフォルトで, 2 群の等分散を仮定しない Welch の方法による検定を行う。

## 2 使用法

原データを用いる場合

```
import sys
sys.path.append("statlib")
from xtest import t_test
t_test(x, y=None, alternative="two_sided", mu=0.0, paired=False, var_equal=False,
       conflevel=0.95, verbose=True)}
```

二次データを用いる場合

```
import sys
sys.path.append("statlib")
from xtest2 import t_test2
t_test2(nx, mx, ux, ny, my, uy, var_equal = False, 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。1 標本検定 (母平均の検定) の場合には指定しない (デフォルトは <code>None</code> )。
<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>conflevel</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>nx</code>	第一群のデータ個数
<code>mx</code>	第一群の平均値
<code>ux</code>	第一群の不偏分散
<code>ny</code>	第二群のデータ個数 (一標本の場合は <code>None</code> )
<code>my</code>	第二群の平均値 (一標本の場合は <code>None</code> )
<code>uy</code>	第二群の不偏分散 (一標本の場合は <code>None</code> ) <code>True</code> )

## 2.2 戻り値の名前

<code>"t"</code>	検定統計量 ( $t$ 分布にしたがう)
<code>"df"</code>	自由度
<code>"pvalue"</code>	$p$ 値
<code>"confint"</code>	信頼区間
<code>"confllevel"</code>	信頼率
<code>"estimate"</code>	標本平均
<code>"mu"</code>	母平均
<code>"alternative"</code>	両側検定・片側検定の種別
<code>"method"</code>	検定手法名

## 3 使用例

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

### 3.1 独立 2 標本, 等分散を仮定しない

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

a = t_test(x, y)
```

```
Welch Two Sample t-test
t = 0.19389, df = 9.9949, p value = 0.85014
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: [-1.74874, 2.08208]
sample estimates: mean of x = 2.83333333, mean of y = 2.66666667
```

二次データを使っても同じ結果になることを確認する。二次データの有効数字が少ないと  $p$  値に影響が出る。

```
from xtest import t_test2
import numpy as np

a = t_test2(len(x), np.mean(x), np.var(x, ddof=1), len(y), np.mean(y),
            np.var(y, ddof=1))
```

Welch Two Sample t-test

```
t = 0.19389, df = 9.9949, p value = 0.85014
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: [-1.74874, 2.08208]
sample estimates: mean of x = 2.83333333, mean of y = 2.66666667
```

### 3.2 独立 2 標本, 等分散を仮定する

```
a = t_test(x, y, var_equal=True)
```

Two Sample t-test

```
t = 0.19389, df = 10, p value = 0.85014
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: [-1.74861, 2.08194]
sample estimates: mean of x = 2.83333333, mean of y = 2.66666667
```

### 3.3 独立 2 標本, 等分散を仮定しない, 片側検定

```
a = t_test(x, y, alternative="less")
```

Welch Two Sample t-test

```
t = 0.19389, df = 9.9949, p value = 0.57493
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval: [-inf, 1.72471]
sample estimates: mean of x = 2.83333333, mean of y = 2.66666667
```

```
a = t_test(x, y, alternative="greater")
```

Welch Two Sample t-test

```
t = 0.19389, df = 9.9949, p value = 0.42507
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval: [-1.39138, inf]
sample estimates: mean of x = 2.83333333, mean of y = 2.66666667
```

### 3.4 対応のある標本

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

Paired t-test

```
t = 0.21035, df = 5, p value = 0.84170
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: [-1.87007, 2.2034]
sample estimate: mean of the differences = 0.166667
```

```
import numpy as np
diff = np.ravel(x)-np.ravel(y)
a = t_test(diff)
```

One Sample t-test

```
t = 0.21035, df = 5, p value = 0.84170
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: [-1.87007, 2.2034]
sample estimate: mean of the differences = 0.166667
```

```
a = t_test2(len(diff), np.mean(diff), np.var(diff, ddof=1))
```

One Sample t-test

```
t = 0.21035, df = 5, p value = 0.84170
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: [-1.87007, 2.2034]
sample estimate: mean of the differences = 0.166667
```

### 3.5 対応のある標本, 片側検定

```
a = t_test(x, y, paired=True, alternative="less")
```

Paired t-test

```
t = 0.21035, df = 5, p value = 0.57915
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval: [-inf, 1.76324]
sample estimate: mean of the differences = 0.166667
```

```
a = t_test(x, y, paired=True, alternative="greater")
```

Paired t-test

```
t = 0.21035, df = 5, p value = 0.42085
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval: [-1.42991, inf]
sample estimate: mean of the differences = 0.166667
```

```
print(a)
```

```
{'t': 0.21035158095583562, 'df': 5, 'pvalue': 0.4208483515552009,
'confint': (-1.4299051010319612, inf), 'confllevel': 0.95, 'estimate':
0.16666666666666666, 'mu': 0, 'alternative': 'greater', 'method':
'Paired t-test'}
```

## 4 Python の既存関数との比較

### 4.1 1 標本 (母平均) の検定 `scipy.stats.ttest_1samp()`

`scipy.stats.ttest_1samp()` は両側検定である。

```
from scipy.stats import ttest_1samp
ttest_1samp([2,1,3,2,4,5,3,4,6,8], 3.4)
```

```
Ttest_1sampResult(statistic=0.6030226891555271, pvalue=0.5613798502873494)
```

```
a = t_test([2,1,3,2,4,5,3,4,6,8], mu=3.4)
```

One Sample t-test

t = 0.60302, df = 9, p value = 0.56138

alternative hypothesis: true difference in means is not equal to 3.4

95 percent confidence interval: [2.29945, 5.30055]

sample estimate: mean of the differences = 3.8

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

```
t = 0.6030226891555271
```

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

```
p value = 0.5613798502873494
```

### 4.2 対応のある 2 標本の検定 `scipy.stats.ttest_rel()`

まあもって、差をとってから `scipy.stats.ttest_1samp()` を使ってもよい (母平均は 0 を指定する)。

`scipy.stats.ttest_rel()` は 0 以外の母平均を指定することができないので、母平均が 0 以外の場合には、`scipy.stats.ttest_1samp()` を使用する。

```
x = [2,1,3,4,2,5,4,3,6,5,4]
y = [3,2,3,4,1,2,3,5,4,3,2]
from scipy.stats import ttest_rel
ttest_rel(x, y)
```

```
Ttest_relResult(statistic=1.347150628109127, pvalue=0.20766366882929266)
```

```
ttest_1samp(np.array(x) - np.array(y), 0)
```

```
Ttest_1sampResult(statistic=1.347150628109127, pvalue=0.20766366882929266)
```

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

Paired t-test

```
t = 1.3472, df = 10, p value = 0.20766
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: [-0.416159, 1.68889]
sample estimate: mean of the differences = 0.636364
```

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

```
t = 1.347150628109127
```

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

```
p value = 0.20766366882929266
```

### 4.3 独立 2 標本の検定 `scipy.stats.ttest_ind()`

`scipy.stats.ttest_ind()` はデフォルトでは 2 群が等分散であることを仮定している。`statlib/t_test()` はデフォルトで等分散を仮定していない。

#### 4.3.1 等分散を仮定しない場合

```
x = [2, 1, 3, 4, 2, 5, 4, 3, 6, 5, 4, 2, 1, 4, 5]
y = [3, 2, 3, 4, 1, 2, 3, 5, 4, 3, 2]
from scipy.stats import ttest_ind
ttest_ind(x, y, equal_var=False)
```

```
Ttest_indResult(statistic=0.9321435176756289, pvalue=0.3605470860756169)
```

```
a = t_test(x, y)
```

Welch Two Sample t-test

```
t = 0.93214, df = 23.996, p value = 0.36055
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: [-0.596043, 1.57786]
sample estimates: mean of x = 3.4, mean of y = 2.90909091
```

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

```
t = 0.9321435176756289
```

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

```
p value = 0.3605470860756167
```

#### 4.3.2 等分散を仮定する場合

```
ttest_ind(x, y)
```

```
Ttest_indResult(statistic=0.8883683312313673, pvalue=0.3831600163024228)
```

```
a = t_test(x, y, var_equal=True)
```

Two Sample t-test

t = 0.88837, df = 24, p value = 0.38316

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval: [-0.649594, 1.63141]

sample estimates: mean of x = 3.4, mean of y = 2.90909091

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

t = 0.8883683312313673

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

p value = 0.3831600163024228