

# 一元配置分散分析

青木繁伸

2019年6月24日

## 1 目的

独立  $k$  標本の平均値に差があるかどうか、一元配置分散分析を行う。

一元配置分散分析では各群の分散が等しいことを仮定することが多いが、等分散を仮定しない Welch の方法によることもできる。R では後者がデフォルトである。ここに示す関数も R と同じく、デフォルトでは等分散を仮定しない検定を行う。

また、通常では 3 群以上の平均値の差の検定に使われるが、両側検定のときは、2 群の場合にも使って差し支えない。2 群の場合にはいわゆる  $t$  検定が使われるが、自由度が  $df$  の場合の  $t$  検定統計量  $t_0$  は、第 1 自由度が 1、第 2 自由度が  $df$  の  $F$  検定統計量  $F_0$  で、 $t_0^2 = F_0$  の関係にあるので、 $p$  値はどちらの検定手法によっても同じである。

## 2 使用法

原データを用いる場合

```
import sys
sys.path.append("statlib")
from xttest import oneway_ANOVA
oneway_ANOVA(x, g=None, var_equal=False, verbose=True)
```

二次データを用いる場合

```
import sys
sys.path.append("statlib")
from xttest import oneway_ANOVA2
oneway_ANOVA2(n, m, u, var_equal=False, verbose=True)
```

### 2.1 引数

<code>x</code>	データベクトル (リスト) または群ごとのデータの二重リスト
<code>g</code>	<code>x</code> がデータベクトルの場合はそれぞれのデータの所属群を表すベクトル (リスト) <code>x</code> が二重リストの場合は省略される
<code>n</code>	各群のサンプルサイズを指定するベクトルまたはリスト
<code>m</code>	各群の平均値を指定するベクトルまたはリスト
<code>u</code>	各群の不偏分散を指定するベクトルまたはリスト
<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

"F"	検定統計量 ( $F$ 分布にしたがう)
"df1"	第 1 自由度
"df2"	第 2 自由度
"pvalue"	$p$ 値
"method"	検定手法名

## 3 使用例

### 3.1 使用例 1

データと所属群の 2 つのベクトル (リスト) で与える。

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

a = oneway_ANOVA([3, 2, 4, 2, 3, 5, 4, 3, 3, 1, 3, 4, 5, 4, 5],
                 [1, 1, 2, 1, 3, 3, 1, 3, 1, 2, 2, 2, 1, 2, 2])
```

```
One-way analysis of means(not assuming equal variances)
F = 0.19301, df1 = 2, df2 = 5.9323, p value = 0.82945
```

群の指定は数値でも文字列 (1 文字に限らない) でもよい。

```
b = oneway_ANOVA([3, 2, 4, 2, 3, 5, 4, 3, 3, 1, 3, 4, 5, 4, 5],
                 ["a", "a", "b", "a", "c", "c", "a", "c", "a", "b",
                  "b", "b", "a", "b", "b"])
```

```
One-way analysis of means(not assuming equal variances)
F = 0.19301, df1 = 2, df2 = 5.9323, p value = 0.82945
```

### 3.2 使用例 2

複数の群のデータを二重リストで与える。

使用例 1 と同じデータを群ごとに整理して与える。

```
c = oneway_ANOVA([[3, 2, 2, 4, 3, 5],
                  [4, 1, 3, 4, 4, 5],
                  [3, 5, 3]])
```

```
One-way analysis of means(not assuming equal variances)
F = 0.19301, df1 = 2, df2 = 5.9323, p value = 0.82945
```

### 3.3 使用例 3

`oneway_ANOVA2()` により、サンプルサイズ、平均値、不偏分散しかわからない二次データに基づいて検定を行うことができる。

```
import scipy as sp

w = [[3, 2, 2, 4, 3, 5],
      [4, 1, 3, 4, 4, 5],
      [3, 5, 3]]
n = list(map(len, w))
m = list(map(sp.mean, w))
u = list(map(lambda x: sp.var(x, ddof=1), w))
```

各群のサンプルサイズ (n), 平均値 (m), 不偏分散 (u) がわかっているならば, 元データは分からなくても検定できる。

```
print("n =", n)
```

```
n = [6, 6, 3]
```

```
print("m =", m)
```

```
m = [3.1666666666666665, 3.5, 3.6666666666666665]
```

```
print("u =", u)
```

```
u = [1.3666666666666667, 1.9, 1.3333333333333333]
```

```
from xtest import oneway_ANOVA2
```

```
d = oneway_ANOVA2(n, m, u) # 元データは不要
```

```
One-way analysis of means(not assuming equal variances)
F = 0.19301, df1 = 2, df2 = 5.9323, p value = 0.82945
```

```
e = oneway_ANOVA2(n, m, u, var_equal=True) # 元データは不要
```

```
One-way analysis of means
F = 0.18947, df1 = 2, df2 = 12, p value = 0.82982
```

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

Python には, `scipy.stats.f_oneway()` がある。

引数の与え方が, `statlib.oneway_ANOVA()` では二重リストであるが, `scipy.stats.f_oneway()` では複数のリストであるので注意が必要。

また, `scipy.stats.f_oneway()` は等分散を仮定した検定であり, 等分散を仮定しない検定は存在しない。

```
from scipy.stats import f_oneway
```

```
f_oneway([3, 2, 2, 4, 3, 5],
          [4, 1, 3, 4, 4, 5],
          [3, 5, 3])
```

```
F_onewayResult(statistic=0.18947368421052635, pvalue=0.8298224452842657)
```

```
c = oneway_ANOVA([[3, 2, 2, 4, 3, 5],
                  [4, 1, 3, 4, 4, 5],
```

```
[3, 5, 3]], var_equal=True)
print("F =", c["F"])
print("p value =", c["pvalue"])
```

One-way analysis of means

F = 0.18947, df1 = 2, df2 = 12, p value = 0.82982

F = 0.18947368421052632

p value = 0.8298224452842657

## 5 2群の場合の oneway\_ANOVA() と t\_test()

```
from xtest import t_test
x = [3,2,5,4,3,6,5,8]
y = [4,3,2,6,5]

d = t_test(x, y)
```

Welch Two Sample t-test

t = 0.50918, df = 9.9668, p value = 0.62171

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

95 percent confidence interval: [-1.68898, 2.68898]

sample estimates: mean of x = 4.5, mean of y = 4

```
print("t^2 =", d["t"]**2)
```

t^2 = 0.2592592592592593

```
g = oneway_ANOVA([x, y])
```

One-way analysis of means(not assuming equal variances)

F = 0.25926, df1 = 1, df2 = 9.9668, p value = 0.62171

```
print("F =", e["F"])
```

F = 0.18947368421052668