

Passing-Bablok 法による回帰直線

青木繁伸

2020年3月17日

1 目的

Passing-Bablok 法による回帰直線を求める。

もともこの方法は、2つの手法により得られる測定値が同等であるかどうかを検定するためのものである。したがって、正の相関を持つ2変数に適用されるべきものである。

ブートストラップ法により得られる信頼限界に基づき、傾きおよび切片がそれぞれの信頼区間内にある場合に、2つの手法は同等と見なされる。

2 使用法

```
import sys
sys.path.append("statlib")
from multi import Passing_Bablok_regression
Passing_Bablok_regression(x, y, n=1, sig=0.95)
```

2.1 引数

x	変数ベクトル
y	変数ベクトル
bootstrap	2以上の値を指定することにより、ブートストラップ法によるパラメータ（切片と傾き）の信頼限界を求める（少なくとも1000以上が望ましい）
sig	信頼率（デフォルトは0.95）
verbose	必要最小限のプリント出力をする

2.2 戻り値の名前

"intercept"	切片
"slope"	傾き
"interceptCL"	bootstrapを2以上にしたときに、切片の信頼限界
"slopeCL"	bootstrapを2以上にしたときに、傾きの信頼限界

3 使用例

```
x = [51, 49, 59, 49, 46, 68, 53, 30, 48, 43]
y = [64, 52, 46, 48, 41, 61, 51, 30, 58, 44]

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

import numpy as np
np.random.seed(123) # 乱数の種 (通常は不要)

a = Passing_Bablok_regression(x, y, bootstrap=1000)
```

	Estimate	Lower C.L.	Upper C.L.
Intercept	-5.400810	-170.670000	127.296667
Slope	1.117409	-1.407083	4.600000

```
def simple_reg(x, y):
    mx = np.mean(x)
    my = np.mean(y)
    slope = sum((x-mx)*(y-my))/sum((x-mx)**2)
    intercept = my-slope*mx
    return intercept, slope
```

```
import numpy as np
import matplotlib.pyplot as plt

intercept = a["intercept"]
slope = a["slope"]
x0 = np.amin(x)
x1 = np.amax(x)
x2 = np.array([x0, x1])
y2 = intercept + slope * x2
intercept_reg, slope_reg = simple_reg(x, y)
y3 = intercept_reg + slope_reg * x2
plt.scatter(x, y, c="black", s=9)
plt.plot(x2, y2, label="Passing-Bablok", linewidth=0.5, color="red")
plt.plot(x2, y3, label="simple regression", linewidth=0.5, color="
    black")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.show()
```

