Passing-Bablok 法による回帰直線

青木繁伸

1　目的

Passing-Bablok 法による回帰直線を求める。

もともとこの方法は，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

```
from Passing_Bablok_regression import Passing_Bablok_regression Passing_Bablok_regression(x, y, n=1,
```

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
"intercept CL"	bootstlap を 2 以上にしたときに，切片の信頼限界
"slope CL"	bootstlap 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 Passing_Bablok_regression import Passing_Bablok_regression

```

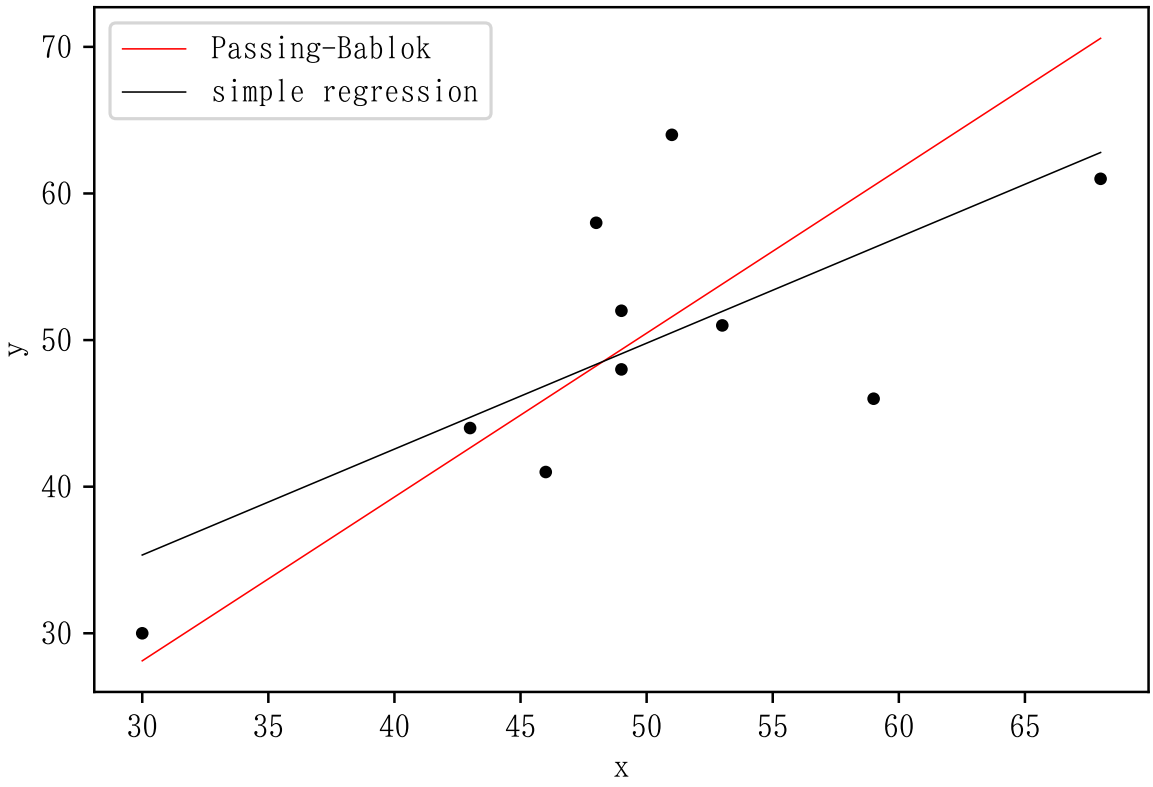
```
a = Passing_Bablok_regression(x, y, bootstrap=1000)
```

	Estimate	Lower C.L.	Upper C.L.
Intercept	-5.400810	-170.6	125.2
Slope	1.117409	-1.4	4.6

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

```
import scipy as sp
import matplotlib.pyplot as plt

intercept = a["intercept"]
slope = a["slope"]
x0 = sp.amin(x)
x1 = sp.amax(x)
x2 = sp.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()
```


Passing-Bablok
simple regression
70
60
50
40
30
y
30
35
40
45
50
55
60
65
x