

Deming 法による回帰直線

青木繁伸

1 目的

Deming 法による回帰直線のパラメータを求める。

2 使用法

```
from Deming_regression import Deming_regression
Deming_regression(x, y, bootstrap=1, a=1, sig=0.95, verbose=True)
```

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）
a	分散比（デフォルトは1） 大きくすると普通の回帰直線に近づく。小さくすると本法の特徴が強く表れる
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"	bootstrapを2以上にしたときに、切片の信頼限界
"slope CL"	bootstrapを2以上にしたときに、傾きの信頼限界

3 使用例

```
x = [2, 1, 3, 2, 4, 3, 5, 6, 5, 4, 1, 4, 5, 6]
y = [2, 3, 2, 4, 5, 4, 6, 6, 7, 5, 6, 6, 5, 6]

import sys
sys.path.append("statlib")
from Deming_regression import Deming_regression

a = Deming_regression(x, y)
```

Intercept: 1.5411333283518531

Slope: 0.8906692824132167

```
b = Deming_regression(x, y, bootstrap=1000)
```

	Estimate	Lower C.L.	Upper C.L.
Intercept	1.541133	-1.691876	4.327238
Slope	0.890669	0.273779	1.672156

```
import scipy as sp

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 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="Deming", 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()
```

