

抵抗直線 (resistant line)

青木繁伸

2020年3月17日

1 目的

抵抗直線 (resistant line) を描く。

2 使用法

```
import sys
sys.path.append("statlib")
from multi import resistant_line
resistant_line(x, y, no_iteration=False, bootstrap=1, sig=0.9, verbose=True)
```

2.1 引数

x	変数ベクトル
y	変数ベクトル
no_iteration	収束するまで繰り返し推定する
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 = range(1961, 1994)
```

```
y = [139.130, 140.293, 143.137, 147.350, 150.686, 144.317,
     151.207, 152.882, 156.867, 155.749, 157.735, 162.962, 159.036,
     158.589, 149.213, 148.725, 161.331, 161.363, 158.899, 142.862,
     139.085, 162.026, 162.117, 163.621, 152.982, 170.722, 162.175,
     144.809, 167.581, 185.984, 176.457, 134.477, 134.477]
```

```
import sys
sys.path.append("statlib")
from multi import resistant_line

a = resistant_line(x, y)
```

```
Intercept: -1189.8441520815638
Slope: 0.6818749926271668
```

```
import numpy as np
np.random.seed(123) # 乱数の種 (通常は不要)
b = resistant_line(x, y, bootstrap=1000)
```

	Estimate	Lower C.L.	Upper C.L.
Intercept	-1189.844152	-1937.840964	822.115012
Slope	0.681875	-0.338769	1.060364

```
import numpy as np

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 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="resitant line", 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()
```

