

RMA(Reduced Major Axis regression)

青木繁伸

1 目的

Reduced Major Axis regression を計算する。

2 使用法

```
from Reduced_Major_Axis_regression import Reduced_Major_Axis_regression
Reduced_Major_Axis_regression(x, y)
```

2.1 引数

x	データベクトル (リストでもよい)
y	データベクトル (リストでもよい)
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

引数はリスト型で与えてもよい。

```
y = [61, 37, 65, 69, 54, 93, 87, 89, 100, 90, 97]
x = [14, 17, 24, 25, 27, 33, 34, 37, 40, 41, 42]

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

a = Reduced_Major_Axis_regression(x, y)
```

	Estimate	S.E.	2.5%	97.5%
Intercept	12.193785	10.549750	-11.671407	36.058977

Slope 2.119366 0.332496 1.367208 2.871524

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

a = Reduced_Major_Axis_regression(x, y)
```

	Estimate	S.E.	2.5%	97.5%
Intercept	-0.449658	1.754993	-6.034828	5.135513
Slope	1.216553	0.529150	-0.467440	2.900545

```
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="RMA regression", 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()
```

