指数回帰 $b^y = a * x$

青木繁伸

2020 年 3 月 17 日

1　目的

指数回帰曲線 $b^y = a * x$ への当てはめを行う。

両辺の対数をとると $y \log b = \log a + \log x$ で，さらに両辺を $\log b$ で割ると，$y = \dfrac{\log a}{\log b} + \dfrac{1}{\log b} \log x$ のようになるので，従属変数 を独立変数の対数を取ったものを $\log x$ で予測する単回帰式 $y = \text{intercept} + \text{slope} \cdot \log x$ を解けばよい。

$\text{slope} = \dfrac{1}{\log b}$ なので，$\log b = \dfrac{1}{\text{slope}}$ となり，$b = \exp \dfrac{1}{\text{slope}}$

$\text{intercept} = \dfrac{\log a}{\log b}$ なので，$\log a = \text{intercept} \cdot \log b$ となり，$a = \exp \dfrac{\text{intercept}}{\text{slope}}$

2　使用法

```
import sys
sys.path.append("statlib")
from multi import exp3
```

2.1　引数

`x`	独立変数ベクトル（リストでもよい）
`y`	従属変数ベクトル（リストでもよい）
`verbose`	必要最小限のプリント出力をする（デフォルトは `True`）。

2.2　戻り値の名前

`"a"`	a
`"b"`	b
`"intercept"`	intercept
`"slope"`	slope
`"predicted"`	予測値
`"resid"`	残差（観察値 – 予測値）
`"x"`	独立変数 x
`"y"`	従属変数 y

"method"　　分析手法名

3 使用例

```
import sys
sys.path.append("statlib")
from multi import exp3

import matplotlib.pyplot as plt

def graph(x, y, intercept, slope):
    x0 = np.min(x)
    x1 = np.max(x)
    delta = (x1-x0)*0.05
    x2 = np.arange(max(x0-delta, 0), x1+delta, (x1-x0+2*delta)/500)
    y2 = intercept + slope * np.log(x2)
    plt.scatter(x, y, c="black", s=9)
    plt.plot(x2, y2, linewidth=0.5, color="red")
    plt.xlabel("x")
    plt.ylabel("y")
    plt.show()
```

3.1 使用例 1

```
import numpy as np

x = np.arange(1, 11)
y = np.array([0.631, 1.262, 1.631, 1.893, 2.096, 2.262, 2.402, 2.524,
    2.631, 2.727])
ans = exp3(x, y)
```

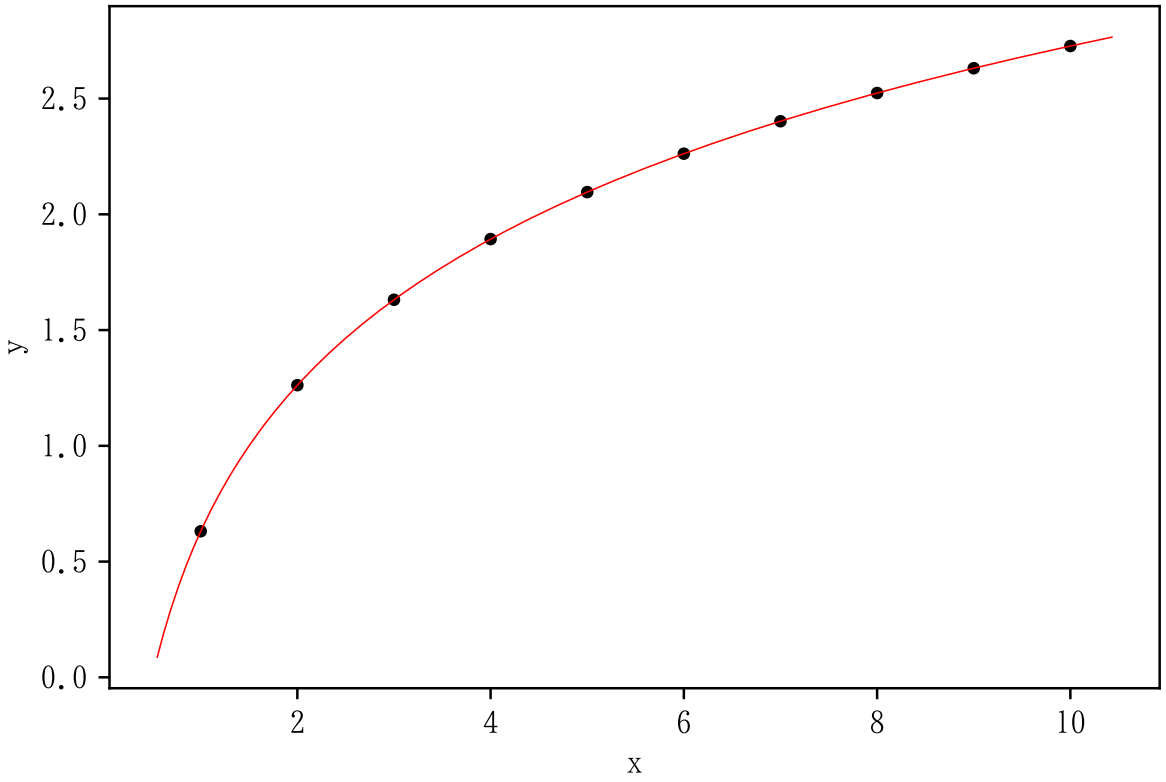
```
b^y = a * x
a = 2.0002, b = 2.99997

    x      y     pred.    resid.
0   1  0.631  0.631025 -0.000025
1   2  1.262  1.261960  0.000040
2   3  1.631  1.631034 -0.000034
3   4  1.893  1.892896  0.000104
4   5  2.096  2.096011 -0.000011
5   6  2.262  2.261969  0.000031
6   7  2.402  2.402284 -0.000284
7   8  2.524  2.523831  0.000169
```

```
8   9  2.631  2.631043 -0.000043
9  10  2.727  2.726947  0.000053
```

```
print(ans)
```

```
{'a': 2.000197950638152, 'b': 2.999971611554844, 'intercept': 0.6310252756461274, 'slope': 0.91
       2.26196908, 2.40228428, 2.52383084, 2.6310425 , 2.7269466 ]), 'resid': array([-2.5275646
       -1.14149727e-05,  3.09227031e-05, -2.84281487e-04,  1.69160023e-04,
       -4.25027269e-05,  5.33969170e-05]), 'x': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
       1.79175947, 1.94591015, 2.07944154, 2.19722458, 2.30258509]), 'y': array([0.631, 1.262,
       2.727]), 'model': 'bˆy = a * x'}
```

```
graph(x, y, ans["intercept"], ans["slope"])
```



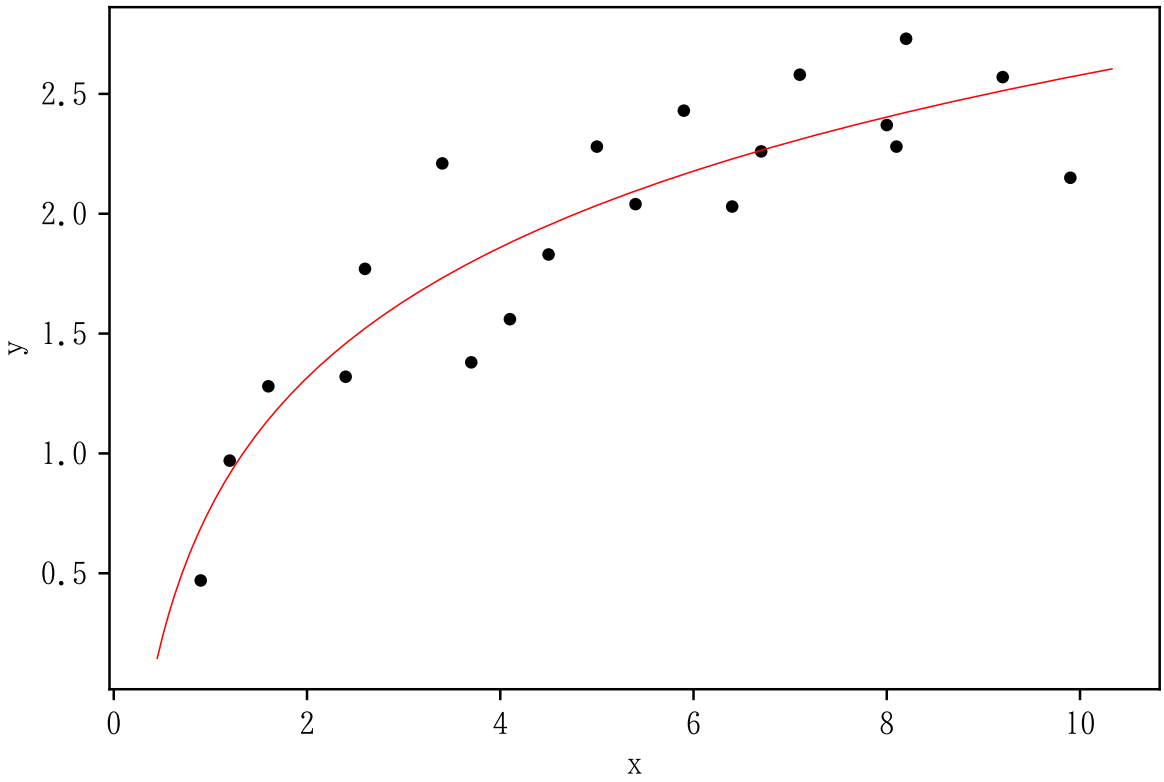
3.2 使用例 2

```
x2 = [0.9, 1.2, 1.6, 2.4, 2.6, 3.4, 4.1, 4.5, 5, 5.4, 5.9, 6.7, 7.1,
8, 8.2, 9.2, 9.9, 8.1, 6.4, 3.7]
y2 = [0.47, 0.97, 1.28, 1.32, 1.77, 2.21, 1.56, 1.83, 2.28, 2.04,
2.43, 2.26, 2.58, 2.37, 2.73, 2.57, 2.15, 2.28, 2.03, 1.38]
ans2 = exp3(x2, y2)
graph(x2, y2, ans2["intercept"], ans2["slope"])
```

```
bˆy = a * x
```

a = 2.67526, b = 3.57705

	x	y	pred.	resid.
0	0.9	0.47	0.689414	-0.219414
1	1.2	0.97	0.915129	0.054871
2	1.6	1.28	1.140843	0.139157
3	2.4	1.32	1.458970	-0.138970
4	2.6	1.77	1.521771	0.248229
5	3.4	2.21	1.732251	0.477749
6	4.1	1.56	1.879136	-0.319136
7	4.5	1.83	1.952175	-0.122175
8	5.0	2.28	2.034840	0.245160
9	5.4	2.04	2.095224	-0.055224
10	5.9	2.43	2.164703	0.265297
11	6.7	2.26	2.264468	-0.004468
12	7.1	2.58	2.309965	0.270035
13	8.0	2.37	2.403604	-0.033604
14	8.2	2.73	2.422978	0.307022
15	9.2	2.57	2.513261	0.056739
16	9.9	2.15	2.570796	-0.420796
17	8.1	2.28	2.413351	-0.133351
18	6.4	2.03	2.228526	-0.198526
19	3.7	1.38	1.798594	-0.418594


2.5
2.0
1.5
1.0
0.5
y
0
2
4
6
8
10
x