

# Box-Cox 変換

青木繁伸

2020年3月17日

## 1 目的

Box-Cox 変換の、最適な  $\lambda$  求める。

得られた  $\lambda$  をそのまま採用するのではなく、意味の明らかな近傍の値を採用するべきである。

$\lambda = 0$  対数変換

$\lambda = 0.5$  平方根変換

$\lambda = -1$  逆数変換

$\lambda = 1$  単なる線形変換（もとのデータから 1 引くだけ）なので、分布型は変わらない。

## 2 使用法

### 2.1 図を描いて求める

```
import sys
sys.path.append("statlib")
from univ import Box_Cox_transformation
Box_Cox_transformation(x, l=-3, r=3, delta=0.1, plot=True, verbose=True)
```

#### 2.1.1 引数

<code>x</code>	データベクトル
<code>l</code>	$\lambda$ の探索範囲起点（デフォルトは -3）
<code>r</code>	$\lambda$ の探索範囲終点（デフォルトは 3）
<code>delta</code>	探索ステップ（デフォルトは 0.1）
<code>plot</code>	図を描くなら <code>True</code>
<code>verbose</code>	必要最小限のプリント出力をする

#### 2.1.2 戻り値の名前

<code>"min"</code>	最適な $\lambda$
<code>"result"</code>	詳細な結果

## 2.2 シンプレックス法により求める

```
import sys
sys.path.append("statlib")
from univ import Box_Cox_transformation2
Box_Cox_transformation2(x, loop=500, epsilon=1e-15, Alpha=2,
                        Beta=0.5, Gamma=2, verbose=True)
```

### 2.2.1 引数

x	データベクトル
loop	収束計算の許容ループ数 (デフォルトは 500)
epsilon	収束判定値 (デフォルトは 1e-15)
Alpha	シンプレックス法の $\alpha$ (デフォルトは 2)
Beta	シンプレックス法の $\beta$ (デフォルトは 0.5)
Gamma	シンプレックス法の $\gamma$ (デフォルトは 2)
verbose	必要最小限のプリント出力をする (デフォルトは True)

### 2.2.2 戻り値の名前

"lambda"	最適な $\lambda$
"converged"	収束していれば True

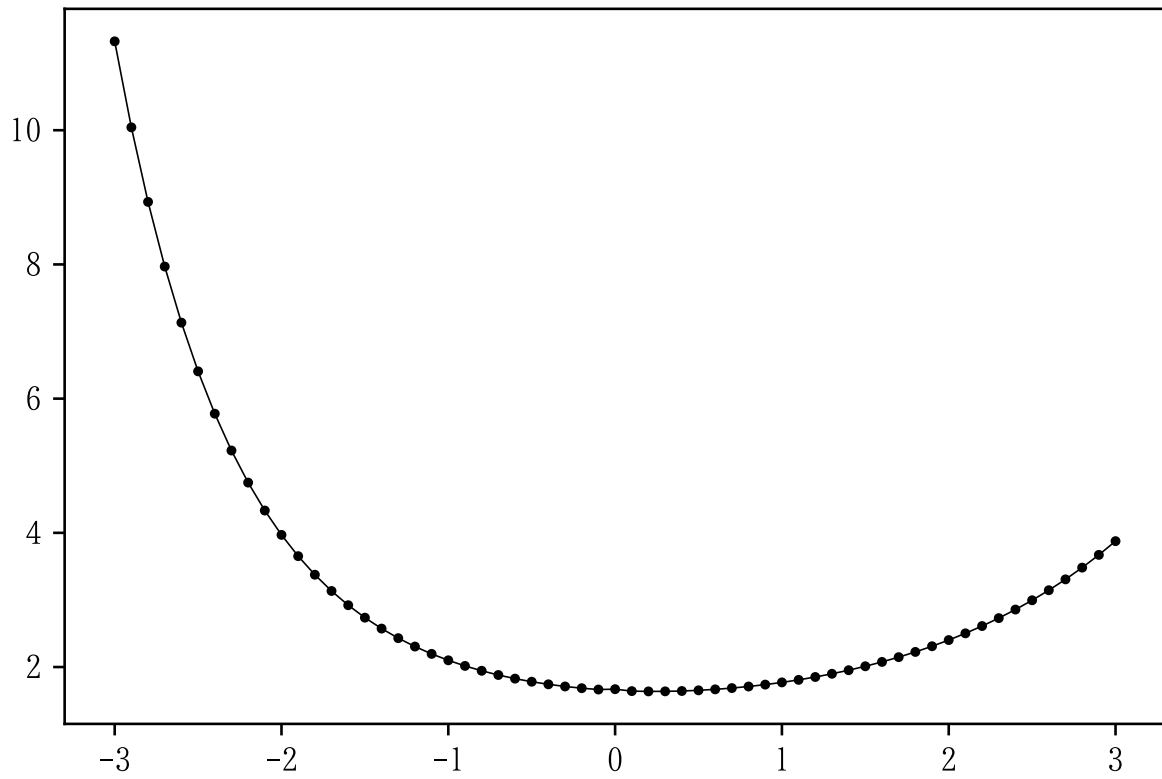
## 3 使用例

```
x = [5, 5, 3.3, 4.3, 4, 5.5, 4, 6, 5, 5, 4, 4.3, 5.3, 5, 6, 6.7,
6.5, 6, 6, 5.3, 7, 5, 6.3, 5.3, 4.5, 6, 7, 2, 2.5, 1.5, 1.7,
1, 1, 2, 1, 1.7, 2, 2, 1, 1.3, 2.5, 1, 2, 2, 1, 3, 1.3, 1, 2,
2, 1.7, 1, 1, 1, 4, 5.3, 3, 3.7, 2.7, 3.3, 5, 2.7, 4.7, 3.7,
3.7, 4, 4.7, 4.3, 5.7, 4.7, 5.3, 5.5, 4.3, 7, 5, 5.3, 5.7, 5.3,
4.3, 5, 5, 5, 4.3, 4.7, 1.3, 1.3, 1.3, 1.7, 2, 1, 1.3, 1.3, 1.7,
1.7, 1.3, 1.3, 1.3, 2.3, 2, 2.7, 2, 1.7, 2.3, 1, 2, 2, 2, 1.7,
2, 1.3, 1.3, 2, 1.3, 1.7, 5, 5, 3.7, 3.3, 3, 3.5, 3.3, 3, 3,
4.3, 3.7, 4, 4, 3.3, 5, 6, 4, 4, 5.7, 5.5, 5.5, 5.5, 5.3, 4.7,
5, 4.3, 4.3, 3, 6, 2.3, 1.7, 2, 1, 1.5, 1, 1, 2, 1, 2, 1.3, 1.7,
1.7, 2, 2, 2, 1.7, 2, 1.7, 0.5, 1.7, 1.3, 1.7, 2.3, 2, 1, 1,
1.3, 2, 2, 5.5, 3.5, 5.3, 5, 6, 8, 6, 4, 2, 3, 2.5, 2, 1, 2,
1, 3, 2]
```

```
import sys
sys.path.append("statlib")
from univ import Box_Cox_transformation
```

```
a = Box_Cox_transformation(x)
```

Lambda = 0.2



```
import sys
sys.path.append("statlib")
from univ import Box_Cox_transformation2

a = Box_Cox_transformation2(x, loop=100)
```

lambda = 0.23686