Chapter 9: Transformation

Stats 500, Fall 2015

Brian Thelen, University of Michigan
443 West Hall, bjthelen@umich.edu

Transformation

- Transforming the response
- Transforming the predictors

Why?

- Nonlinearity
- Heteroscedasticity
- May improve fit

Box-Cox Method

Transformation on the response: $y \to g_\lambda(y)$.
A family of transformations **indexed by λ** when $y > 0$:

$$g_\lambda(y) = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \ln y & \lambda = 0 \end{cases}$$

Box-Cox Method Continued

- Can compute **likelihood** of the data using the normal assumption for any given λ
- Choose λ to **maximize** :

$$L(\lambda) = -\frac{n}{2}\ln\left(RSS_\lambda/n\right) + (\lambda - 1)\sum_i \ln y_i$$

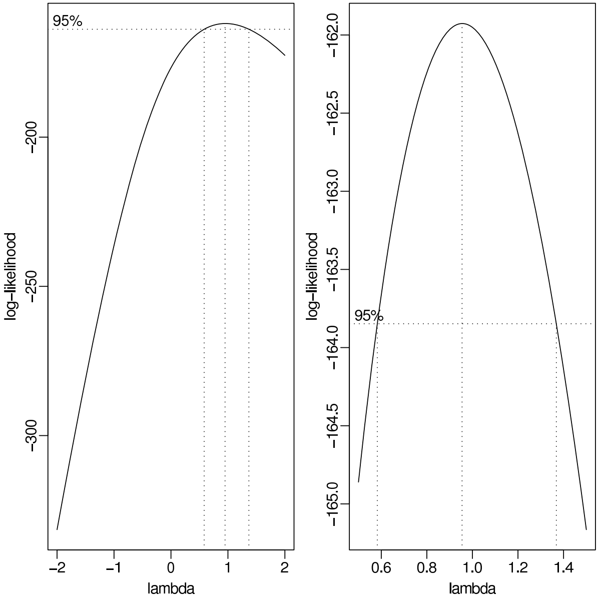
- Compute confidence intervals for λ using asymptotic distribution of the likelihood

Savings & Gala Examples

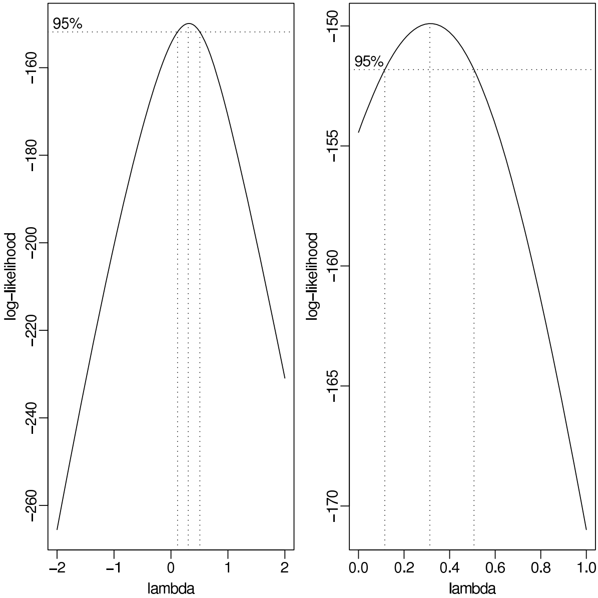
Recall from Chapter 6 & 8

```
> library(MASS)
## Box-Cox method for Savings data
> g = lm(sr ~ pop15 + pop75 + dpi + ddpi, savings)
> boxcox(g, plotit=T)
> boxcox(g, plotit=T, lambda=seq(0.5, 1.5, by=0.1))
## Box-Cox method for Gala data
> g = lm(Species ~ Area + Elevation + Nearest
    + Scruz + Adjacent, gala)
> boxcox(g, plotit=T)
> boxcox(g, plotit=T, lambda=seq(0, 1, by=0.05))
```

Savings Example



Gala Example



Remarks on Box-Cox Method

- May not choose the λ that exactly maximizes $L(\lambda)$, but instead choose one that is **easily interpreted** .
- Sensitive to **outliers** . E.g., $\hat{\lambda} = 5$ – ask why?
- If some $y_i \le 0$, can add a constant.
- Transformations of proportions, counts – generalized linear models (later)

$$\ln\left(\frac{y}{1-y}\right)$$

Transforming the Predictors

Before:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

Now:

$$y = \beta_0 + \beta_1 f_1(x) + \cdots + \beta_q f_q(x) + \epsilon$$

$f_j(x)$ are called the **basis functions** . Examples:

- Polynomials
- Regression splines

Polynomials (One Predictor Case)

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_d x_1^d + \epsilon$$

How to choose d:

1. Keep **adding** terms until the new term is not statistically significant

OR

2. Start with a large d – keep **eliminating** the non-significant highest order term

Warning: Do not eliminate lower order terms even if they are not statistically significant.

Savings Example

```
# tired of typing data = savings?
> attach(savings)

## Polynomials
## 1st degree
> summary(lm(sr ~ ddpi))
Coefficients:
            Estimate Std.Error t value Pr(>|t|)
(Intercept)   7.8830    1.0110   7.797 4.46e-10
ddpi        0.4758    0.2146   2.217   0.0314
```

```
## 2nd degree
> summary(lm(sr ~ ddpi + I(ddpi^2)))
             Estimate Std.Error t value Pr(>|t|)
(Intercept)  5.13038   1.43472   3.576 0.000821
ddpi        1.75752   0.53772   3.268 0.002026
I(ddpi^2) -0.09299   0.03612  -2.574 0.013262
```

```
## 3rd degree
> summary(lm(sr ~ ddpi + I(ddpi^2) + I(ddpi^3)))
            Estimate Std.Error t value Pr(>|t|)
Intercept  5.145e+00 2.199e+00   2.340   0.0237
ddpi      1.746e+00 1.380e+00   1.265   0.2123
ddpi^2  -9.097e-02 2.256e-01  -0.403   0.6886
ddpi^3  -8.497e-05 9.374e-03  -0.009   0.9928
```

```
## Be careful with elimination
> mddpi = ddpi - 10
> summary(lm(sr ~ mddpi + I(mddpi^2)))
Coefficients:
           Estimate Std.Error t value Pr(>|t|)
Intercept  13.40705   1.42401   9.415 2.16e-12
mddpi     -0.10219   0.30274  -0.338   0.7372
mddpi^2   -0.09299   0.03612  -2.574   0.0133
```

Orthogonal Polynomials

For numerical stability:

$$\begin{aligned} z_1 &= a_1 + b_1 x \\ z_2 &= a_2 + b_2 x + c_2 x^2 \\ z_3 &= a_3 + b_3 x + c_3 x^2 + d_3 x^3 \\ \vdots &= \vdots \end{aligned}$$

$a, b, c \ldots$ are chosen so that $z_j^T z_{j'} = 0$ when $j \neq j'$.

Savings Example

```
## Orthogonal polynomials
> summary(lm(sr ~ poly(ddpi, 4)))
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      9.67100    0.58460  16.543   <2e-16 ***
poly(ddpi, 4)1   9.55899    4.13376   2.312   0.0254 *
poly(ddpi, 4)2 -10.49988    4.13376  -2.540   0.0146 *
poly(ddpi, 4)3  -0.03737    4.13376  -0.009   0.9928
poly(ddpi, 4)4   3.61197    4.13376   0.874   0.3869
Residual standard error:  4.134 on 45 degrees of freedom
Multiple R-Squared: 0.2182    Adjusted R-squared: 0.1488
F-statistic: 3.141 on 4 and 45 DF       p-value: 0.02321
```

Polynomials in several predictors

Define polynomials in more than one variable. E.g.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \epsilon$$

R command:

```
> g = lm(sr ~ polym(pop15, ddpi, degree=2))
```

Regression Splines

Disadvantage of polynomials: each data point affects the fit **globally** . Remedy: B-**spline** .

Cubic B-spline basis functions on interval (a, b) with pre-specified knots $t_1, \ldots, t_k$:

- Non-zero on interval defined by four successive knots and zero elsewhere. **Local** influence property.
- Cubic polynomial fit to each four successive knots.
- **Smoothness** .
- Integrates to one.

Simulation Example

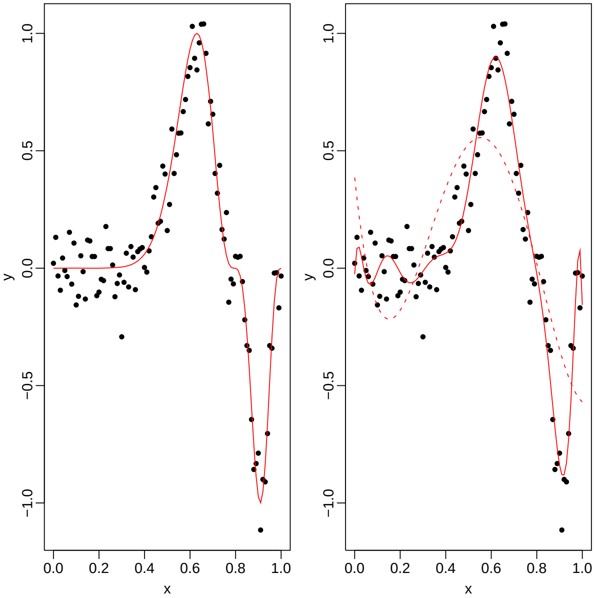
$$y = \sin^3\left(2\pi x^3\right) + \epsilon, \quad \epsilon \sim N(0, 0.1^2)$$

- Not a polynomial, not a cubic spline...
- But smooth and has many inflection points

Simulation Example

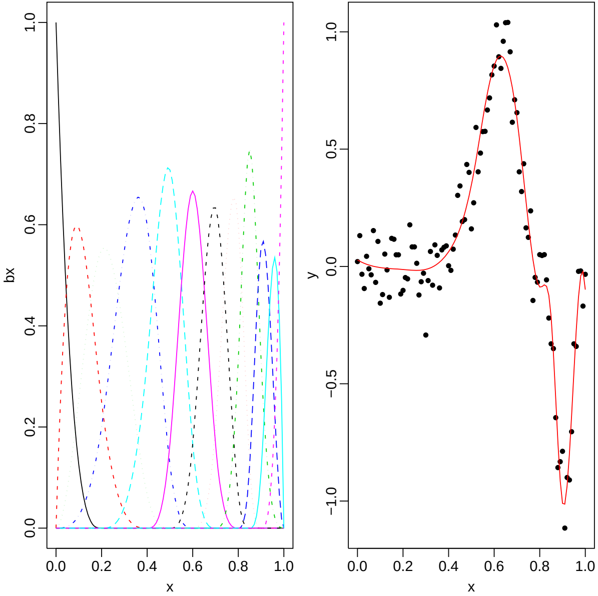
```
## Data generation
> myf = function(x) sin(2*pi*x^3)^3
> x = seq(0, 1, by=0.01)
> y = myf(x) + 0.1*rnorm(101)
> matplot(x, cbind(y, myf(x)), type="pl")

## Polynomials
> g4 = lm(y ~ poly(x, 4))
> g12 = lm(y ~ poly(x, 12))
> matplot(x, cbind(y, g4$fit, g12$fit),  type="pll")
```

Polynomial results



```
## Regression splines
> library(splines)
> knots = c(0, 0, 0, 0, 0.2, 0.4, 0.5, 0.6,
    0.7, 0.8, 0.85, 0.9, 1, 1, 1, 1)
> bx = splineDesign(knots, x)
> gs = lm(y ~ bx)
> matplot(x, bx, type="l")
> matplot(x, cbind(y, gs$fit), type="pl")
```

Spline results



Other Transformations

- Smoothing splines
- Generalized additive models
- CART, MARS, MART, neural networks

Rule of thumb:
– for large data sets, complex models are better (**with appropriate control of the number of parameters**);
– for small data sets or high noise levels (e.g., social sciences), standard regression is more appropriate.