Chapter 8: Transformation

Gunwoong Park

Lecture Note

University of Seoul

Outline

- Transforming the response
    - The Box-Cox method
    - Logit
    - Fisher transformation
- Transforming the predictors
    - Polynomials
    - Regression splines

Reasons to try transformations

- Nonlinearity
- Non-constant error variance
- Correlated errors
- May improve fit
- Prior information: Incorporate a physical law or some other known relationship

Examples

$$
\begin{aligned}
Y &= \exp(\beta_0 + \beta_1 X) \cdot \exp(\epsilon) \\
\ln(Y) &= \beta_0 + \beta_1 X + \epsilon
\end{aligned}
$$

How to interpret $\hat{\beta}_1$?

$\Rightarrow$ An increase of one in X_1 would multiply the predicted response by $e^{\hat{\beta}_1}$

Box-Cox Method

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

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

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

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

- R tries a lot of λs

Remarks

RSS: Residual Sum of Square

$$\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

A good model has a small RSS

Box-Cox Method Continued

- In practice, use y^λ.
- In terms of prediction, use maximizer λ.
- In terms of interpretation, use interpretable value near the maximizer λ.

Box-Cox Method Continued

If $\hat{\lambda} = 0.46$, it would be hard to explain what this new response $y^{0.46}$ means.

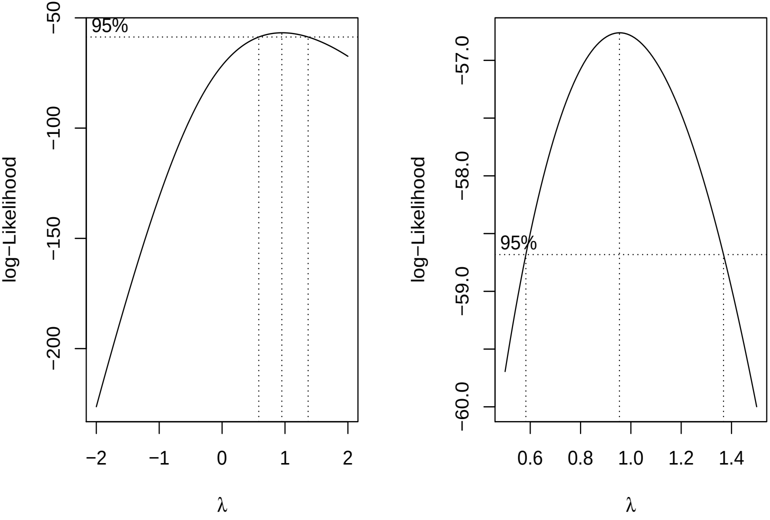
If 95% CI for λ contains 0.5, $\sqrt{y}$ would be preferred because it is easier to interpret.

Savings & Galapagos Tortoise Examples

Recall from Chapter 4 & 6

```
> 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))
```

Savings Example

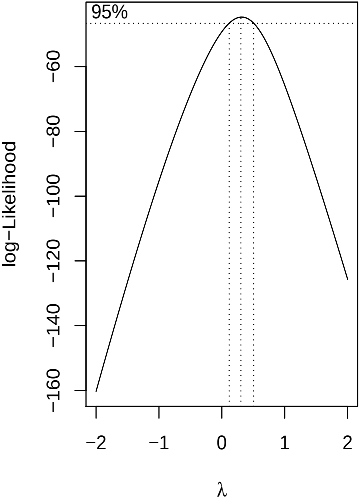


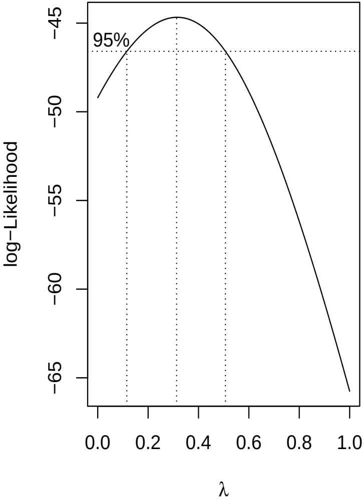
Savings & Galapagos Tortoise Examples

Recall from Chapter 4 & 6

```
> library(MASS)
## Box-Cox method for the Tortoise 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))
```

Galapagos Tortoise Example





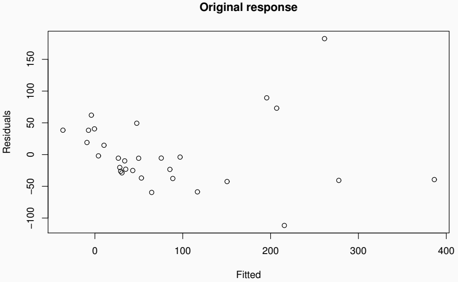
Transformation in the Tortoise example

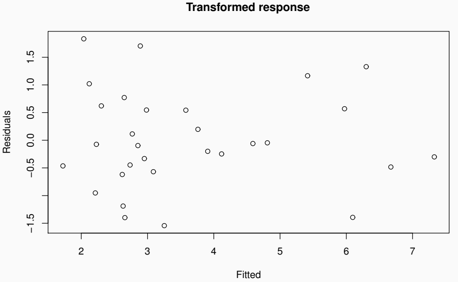
```
> summary(lm(Species ~ Area + Elevation + Nearest +
+         Scruz + Adjacent, data=gala))
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.068221  19.154198   0.369 0.715351
Area        -0.023938   0.022422  -1.068 0.296318
Elevation    0.319465   0.053663   5.953 3.82e-06 ***
Nearest      0.009144   1.054136   0.009 0.993151
Scruz       -0.240524   0.215402  -1.117 0.275208
Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
---
Residual standard error: 60.98 on 24 degrees of freedom
Multiple R-squared: 0.7658,     Adjusted R-squared: 0.7171
F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

Transformation in the Tortoise example

```
> summary(lm(Species^(1/3) ~ Area + Elevation + Nearest +
+         Scruz + Adjacent, data=gala))
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.2479224  0.3052013   7.365 1.32e-07 ***
Area        -0.0007349  0.0003573  -2.057  0.05070 .
Elevation    0.0054510  0.0008551   6.375 1.37e-06 ***
Nearest      0.0118152  0.0167965   0.703  0.48855
Scruz       -0.0045951  0.0034322  -1.339  0.19317
Adjacent    -0.0010597  0.0002820  -3.757  0.00097 ***
---
Residual standard error: 0.9716 on 24 degrees of freedom
Multiple R-squared: 0.7543,     Adjusted R-squared: 0.7032
F-statistic: 14.74 on 5 and 24 DF,  p-value: 1.192e-06
```

Diagnostic plots





Remarks on the 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 $y_i \leq 0$, can add a constant.
- Transformations of proportions, counts – generalized linear models (later in the course)
- A "quick fix": if y_i's are proportions (range from 0 to 1), consider

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

2. Logit Method

A special transformations for binomial data (count).

- Response y_i: number of successes out of n_i independent trials with probability of success p_i

$$\text{Logit}(p) = \log\left(\frac{p}{1-p}\right)$$

Logit Method: Binomial Data

- $x = (x_1, x_2, \ldots, x_p)$: predictors (quantitative, factors, or both)
- Goal: model the relationship between y and $x_1, \ldots, x_p$ via modeling the relationship between p_i and $x_1, \ldots, x_p$.

Review: The Binomial Distribution

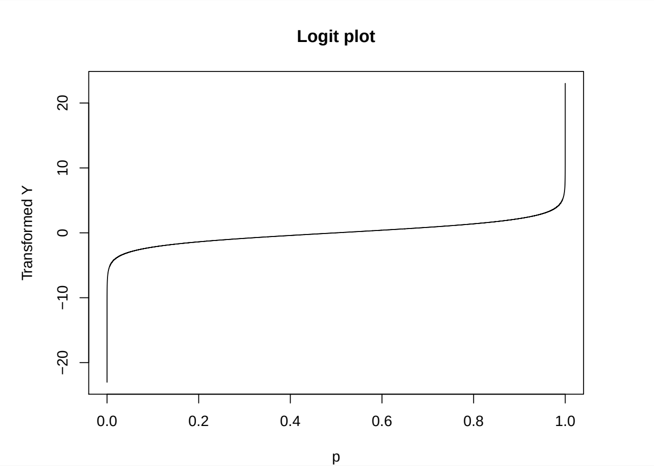
- n independent trials $Z_1, \ldots, Z_n$
- $P(Z_i = 1) = p$ ("success")
  $P(Z_i = 0) = 1 - p$ ("failure")
- The binomial variable $Y = \sum_{i=1}^{n} Z_i$ is the total number of successes out of n iid trials

Review: The Binomial Distribution

- $E(Y) = np$
- $Var(Y) = np(1-p)$
- Sample proportion (estimate of p)

$$\hat{p} = \frac{Y}{n}$$

Logit Method

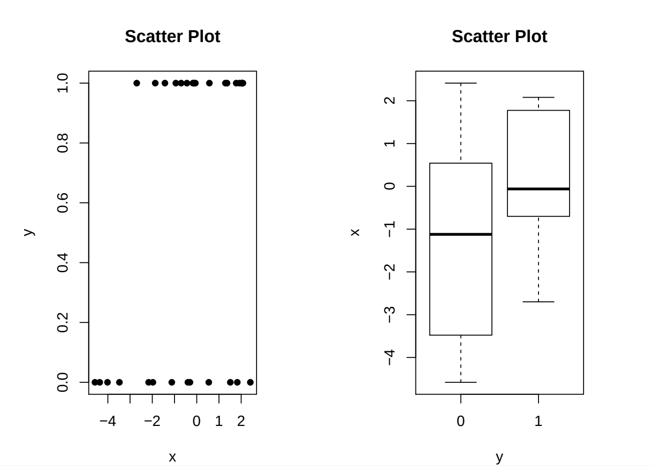


Simulation Study

- Y: Response, $\{0, 1\}$ (e.g. disease ,gender)
- X: Predictor
- Model:

$$\text{Logit}(Y) = \beta_0 + \beta_1 X + \epsilon$$

Simulation Study



Simulation Study

```
> glm(y ~ x, family = "binomial")

Call:  glm(formula = y ~ x, family = "binomial")

Coefficients:
(Intercept)                x
0.4448652914945  0.4014087187817
```

Transforming the Predictors

Before:

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

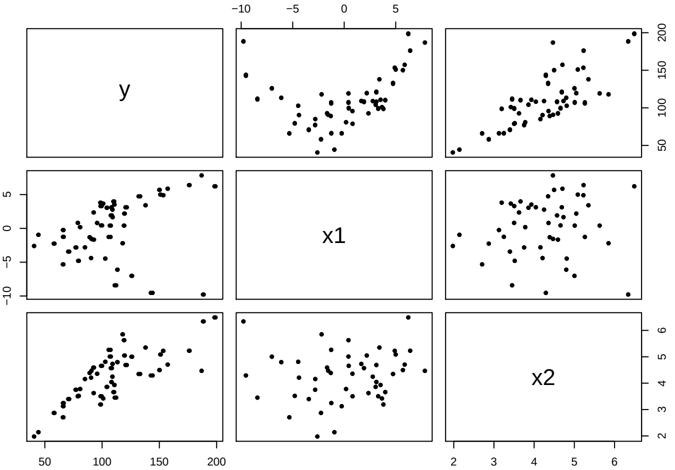
Now:

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

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

1. Broken stick regression (skip)
2. Polynomials
3. Regression splines

Example



Example

$$Y \propto X_1^2$$

$$Y \propto X_2$$

Transforming predictors is better.

2. 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

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

-Focusing only on highest order.

Polynomials (One Predictor Case) Example

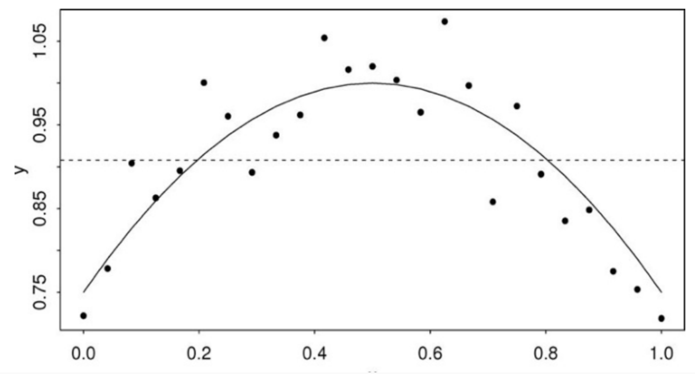


Figure 1: Scatter Plot

Forward: Step 1: 1st degree

```
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)   99.614     15.411   6.464 3.99e-09 ***
x             -3.986      1.498  -2.661  0.00911 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 152.3 on 98 degrees of freedom
Multiple R-squared:  0.06737,Adjusted R-squared:  0.05786
F-statistic:  7.08 on 1 and 98 DF,  p-value: 0.009111
```

Forward: Step 2: 2nd degree

```
Coefficients:
Estimate Std. Error  t value Pr(>|t|)
(Intercept) -0.0288700  0.1169616   -0.247    0.806
x            0.0104287  0.0098592    1.058    0.293
I(x^2)       1.0006549  0.0006423 1557.883   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9679 on 97 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 1.301e+06 on 2 and 97 DF,  p-value: < 2.2e-16
```

Forward: Step 3: 3rd degree

```
Coefficients:
Estimate Std. Error  t value Pr(>|t|)
(Intercept) -7.916e-02  1.212e-01   -0.653    0.515
x           -9.183e-03  1.652e-02   -0.556    0.580
I(x^2)       1.001e+00  7.363e-04 1359.836   <2e-16 ***
I(x^3)       6.495e-05  4.404e-05    1.475    0.144
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9621 on 96 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 8.78e+05 on 3 and 96 DF,  p-value: < 2.2e-16
```

Backward Ellimination

Suppose that the maximum degree $d = 4$.

Backward: Step 1: 4th degree

```
> summary(lm(y ~ x + I(x^2) + I(x^3) + I(x^4) ) )
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.26e-01   1.55e-01    1.46     0.15
x           -1.79e-02   1.95e-02   -0.92     0.36
I(x^2)       9.97e-01   2.35e-03  423.72   <2e-16 ***
I(x^3)       6.23e-05   7.13e-05    0.87     0.38
I(x^4)       8.22e-06   5.53e-06    1.49     0.14
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

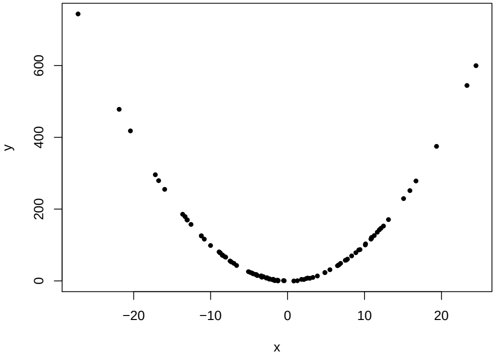
Backward: Step 2: 3rd degree

```
Coefficients:
Estimate Std. Error  t value Pr(>|t|)
(Intercept) -7.916e-02  1.212e-01   -0.653    0.515
x           -9.183e-03  1.652e-02   -0.556    0.580
I(x^2)       1.001e+00  7.363e-04 1359.836   <2e-16 ***
I(x^3)       6.495e-05  4.404e-05    1.475    0.144
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9621 on 96 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 8.78e+05 on 3 and 96 DF,  p-value: < 2.2e-16
```

Backward: Step 3: 2nd degree

```
Coefficients:
Estimate Std. Error  t value Pr(>|t|)
(Intercept) -0.0288700  0.1169616   -0.247    0.806
x            0.0104287  0.0098592    1.058    0.293
I(x^2)       1.0006549  0.0006423 1557.883   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9679 on 97 degrees of freedom
Multiple R-squared:      1,Adjusted R-squared:      1
F-statistic: 1.301e+06 on 2 and 97 DF,  p-value: < 2.2e-16
```

Issue of Forward Selection



Issue of Forward Selection

```
> summary(lm(y ~ x ) )
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)   93.807     13.546    6.93  4.6e-10 ***
x             -0.303      1.398   -0.22     0.83
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Issue of Forward Selection

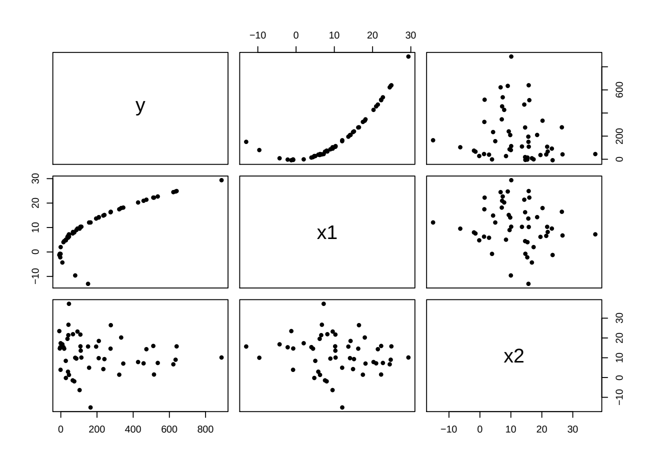
```
> summary(lm(y ~ x + I(x^2)) )
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.072091   0.119886   -0.60     0.55
x           -0.012966   0.010134   -1.28     0.20
I(x^2)       1.000828   0.000733 1365.37   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Polynomials (Two Predictor Case)

- Two Predictors: X_1, X_2
- True Model:

$$Y = 0.5 + X_1 - 0.4X_2 + X_1^2 + \epsilon$$

Polynomials (Two Predictor Case)



Forward: Step 1

```
> summary( lm(y~ x1 + x2, data) )$coef
Estimate Std. Error t value Pr(>|t|)
(Intercept)   -8.150      34.25  -0.238 8.13e-01
x1            20.034       1.81  11.095 1.01e-14
x2            -0.291       1.76  -0.165 8.69e-01
```

Forward: Step 2

```
> summary( lm(y~ x1 + x2 + I(x1*x2) + I(x1^2) + I(x2^2), data) )$coef
Estimate Std. Error  t value  Pr(>|t|)
(Intercept)  0.658052   0.217839    3.021  4.19e-03
x1           0.994548   0.023057   43.135  1.21e-37
x2          -0.414351   0.019318  -21.449  6.28e-25
I(x1 * x2)   0.000708   0.001321    0.536  5.95e-01
I(x1^2)      0.999611   0.000607 1645.597 5.19e-107
I(x2^2)      0.000127   0.000469    0.271  7.88e-01
```

Forward: Step 3

```
> summary( lm(y~ x1 + x2 + I(x1*x2) + (x1^2) + I(x2^2)
+ I(x1^2*x2) + I(x1*x2^2) + I(x1^3) + I(x2^3), data) )$coef
Estimate Std. Error t value Pr(>|t|)
(Intercept)  -67.62499   15.40574   -4.39 7.79e-05
x1            18.97354    1.63939   11.57 1.70e-14
x2            13.40205    2.37122    5.65 1.35e-06
I(x1 * x2)    -1.84729    0.21621   -8.54 1.20e-10
I(x2^2)       -0.59440    0.12344   -4.82 2.03e-05
I(x1^2 * x2)   0.04707    0.00409   11.50 2.07e-14
I(x1 * x2^2)   0.03513    0.00871    4.03 2.34e-04
I(x1^3)        0.01457    0.00202    7.22 8.09e-09
I(x2^3)        0.00671    0.00159    4.22 1.34e-04
```

Forward: Step 4

```
> summary( lm(y~ x1 + x2 + I(x1*x2) + (x1^2) + I(x2^2) + I(x1^2*x2) + I(x1*x2^2) + I(x1^3)
+             + I(x1^3*x2) + I(x1*x2^3)+ I(x1^2*x2^2) + I(x1^4) + I(x2^4)
+             , data) )$coef
Estimate Std. Error t value Pr(>|t|)
(Intercept)    -3.60e+01   6.72e+00   -5.36 5.04e-06
x1              1.23e+01   9.86e-01   12.46 1.29e-14
x2              1.25e+01   1.56e+00    7.98 1.78e-09
I(x1 * x2)     -2.24e+00   1.89e-01  -11.84 5.62e-14
I(x2^2)        -1.05e+00   1.40e-01   -7.53 6.65e-09
I(x1^2 * x2)    1.03e-01   4.27e-03   24.14 7.99e-24
I(x1 * x2^2)    1.17e-01   1.26e-02    9.28 4.43e-11
I(x1^3)         3.46e-02   2.38e-03   14.50 1.32e-16
I(x2^3)         2.90e-02   4.73e-03    6.12 4.80e-07
I(x1^3 * x2)   -1.56e-03   1.26e-04  -12.39 1.53e-14
I(x1 * x2^3)   -1.71e-03   2.83e-04   -6.06 5.87e-07
I(x1^2 * x2^2) -2.30e-03   2.65e-04   -8.70 2.24e-10
I(x1^4)        -3.86e-04   6.04e-05   -6.40 2.05e-07
I(x2^4)        -2.37e-04   4.73e-05   -5.02 1.42e-05
```

Issue

Finding an optimal d may be impossible

Savings Example: Forward Selection

```
# 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
```

Linear Transformation

Linear Transformation of a predictor does not change its p-value.

```
## 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.

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

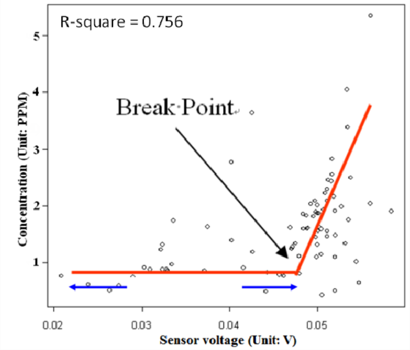
R command:

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

Broken Stick Regression

Sometimes, there are different linear regression models apply in different regions of the data.

- Economic Crisis



Broken Stick Regression: Basis

An important property of broken stick regression is continuity. Hence we define the following basis functions:

$$B_l(x) = \begin{cases} c - x & \text{if } x < c \\ 0 & \text{otherwise} \end{cases} \qquad B_r(x) = \begin{cases} x - c & \text{if } x \geq c \\ 0 & \text{otherwise} \end{cases}$$

where c marks the division between the two groups.

B_l and B_r form a first-order spline basis with a knotpoint at c.

Broken Stick Regression Model

The Form of the model:

$$y = \beta_0 + \beta_1 B_l + \beta_2 B_r + \epsilon.$$

Example:

```
> Ihs < - function (x) ifelse(x < 35, 35!x, 0)
> rhs < - function (x) ifelse(x < 35, 0, x!35)
> gb < - 1m (sr ~ Ihs (pop15) + rhs(pop15), savings)
> x < - seq(20, 48, by=l)
> py < - gb$coef[1]+gb$coef[2]*lhs(x)+gb$coef[3]*rhs(x)
> lines (x, py, lty=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 $\Rightarrow$ local influence property
- Cubic polynomial fit to each four successive knots
- Smooth
- Integrates to one

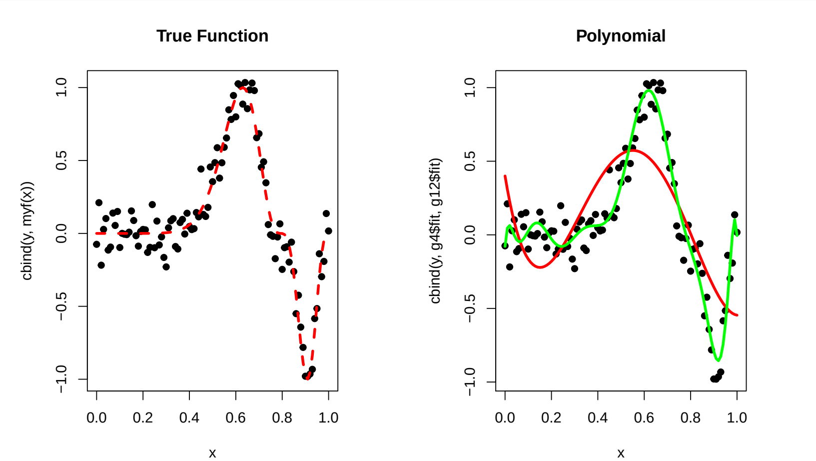
Simulation Example

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

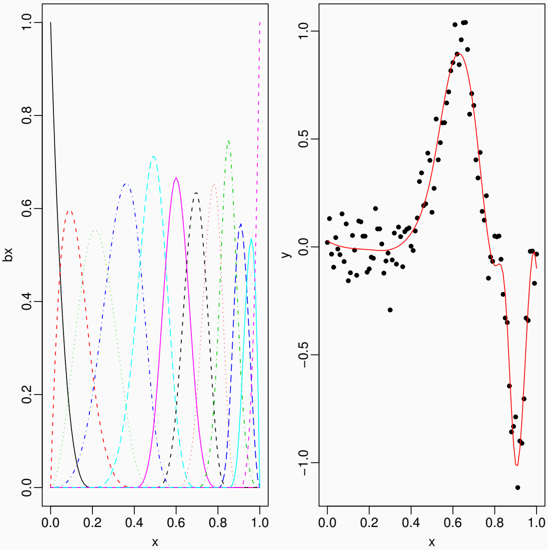
- Not a polynomial, not a cubic spline...
- But smooth and has many inflection points

```
> ## 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", lwd= 3, pch = 20,
main ="True Function")

> ## Polynomials
> g4 = lm(y ~ poly(x, 4))
> g12 = lm(y ~ poly(x, 12))
> matplot(x, cbind(y, g4$fit, g12$fit),  type="plll", pch = 20, lty=1,
lwd= 3, col = c("black","red","green"), main="Polynomial" )
```

Polynomial results



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.

Other Transformations

As we have a better computer,

New Rule of thumb:
Mixtures of complex models are better with appropriate control of the number of parameters;