Chapter 9: Transformation

Outline

1. Transforming the response
   - The Box-Cox method
2. Transforming the predictors
   - Polynomials
   - Regression splines

Reasons to try transformations

- Nonlinearity
- Heteroscedasticity
- May improve fit
- Incorporate a physical law or some other known relationship

Box-Cox Method

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

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

Box-Cox Method Continued

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

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

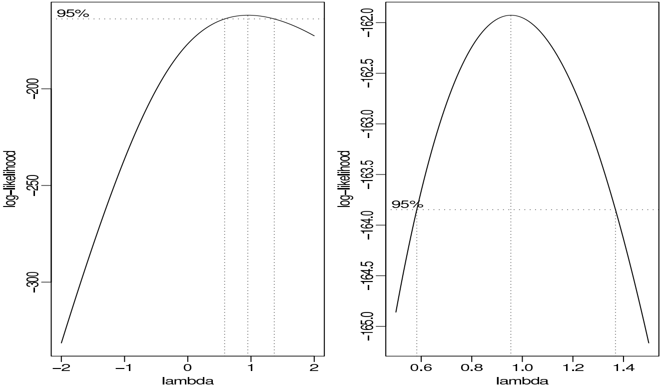
- Compute confidence intervals for λ using asymptotic distribution of the likelihood

Savings & Galapagos Tortoise Examples

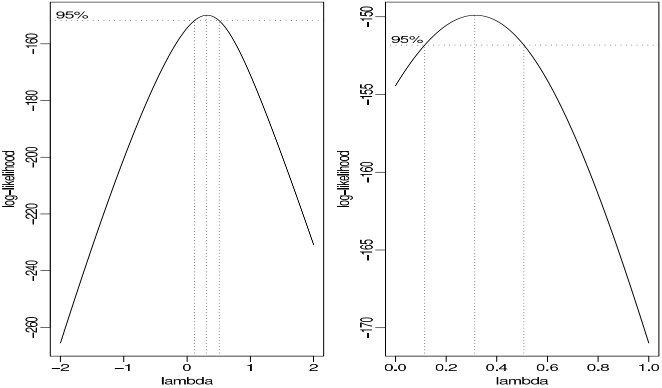
Recall from Chapter 4 & 6

```
> 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))
## 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))
```

Savings Example



Galapagos Tortoise Example



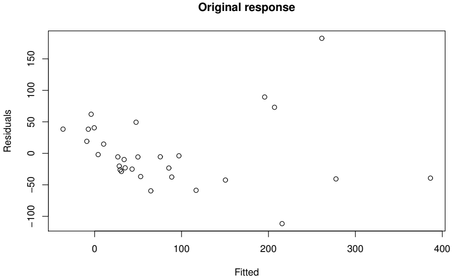
Transformation in the Tortoise example

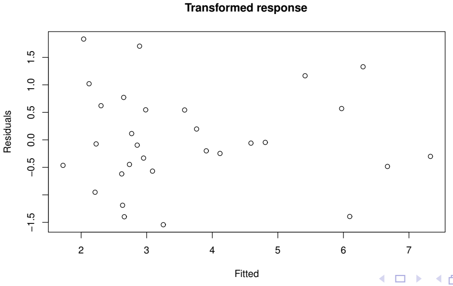
```
> summary(lm(Species ~ Area + Elevation + Nearest +
+          Scruz + Adjacent, data=gala))
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.068221  19.154198   0.369 0.715351
Area        -0.023938   0.022422  -1.068 0.296318
Elevation    0.319465   0.053663   5.953 3.82e-06 ***
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

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

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:

- 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

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 $\Rightarrow$ local influence property
- Cubic polynomial fit to each four successive knots
- Smooth
- Integrates to one

Simulation Example

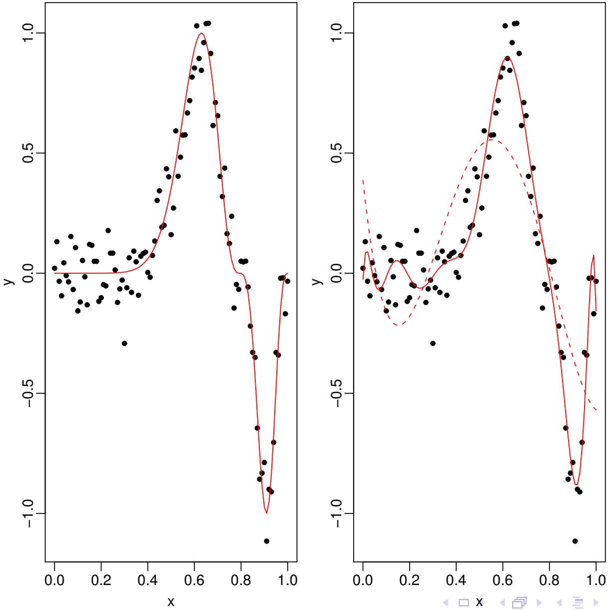
$$y = \sin^3\left(2\pi x^3\right) + \epsilon, \quad \epsilon \sim N(0, 0.1^2)$$

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

Simulation Example

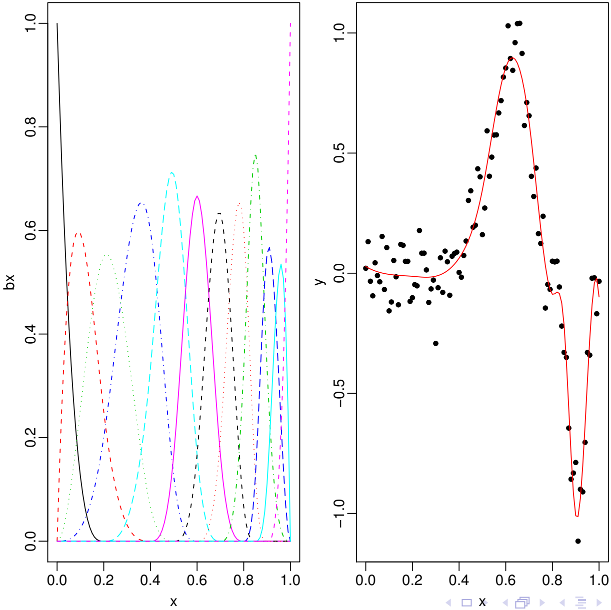
```
## Data generation
> myf = function(x) sin(2*pi*x^3)^3
> x = seq(0, 1, by=0.01)
> y = myf(x) + 0.1*rnorm(101)
> matplot(x, cbind(y, myf(x)), type="pl")

## 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.

Chapter 10: Variable Selection

Variable Selection

1. Testing-based approaches
   - Backward elimination
   - Forward selection
   - Stepwise regression
2. Criterion-based approaches
   - AIC and BIC
   - Adjusted R^2
   - Mallows' C_p

Testing-based approaches

- General idea: **test significance** of predictors and eliminate in some principled fashion
- Based on individual p-values
- Multiple testing is not accounted for, but ranking is more important than the absolute size of p-values
- Different methods use different **rules to add/delete predictors**

Backward Elimination

1. Start with all the predictors in the model
2. **Remove** the predictor with the highest p-value greater than α
3. Refit the model and go to step 2
4. Stop when all p-values are less than α

$\alpha > 0.05$ may be better if prediction is the goal.

Forward Selection

1. Start with no predictor variables
2. For all predictors not in the model, check the p-value **if** they are added to the model
3. **Add** the one with the **smallest p-value** less than α
4. Refit the model and go to step 2
5. Stop when no new predictors can be added

Stepwise regression is a combination of backward elimination and forward selection (allows to add variables back after they have been removed).

Life Expectancy Example

- Census data from 50 states
- Response: life expectancy in years (1969-71)
- Predictors:

```
'Population': population estimate as of July 1, 1975
'Income': per capita income (1974)
'Illiteracy': illiteracy (1970, percent of population)
'Murder': murder and non-negligent manslaughter rate
   per 100,000 population (1976)
'HS Grad': percent high-school graduates (1970)
'Frost': mean number of days with minimum temperature
   below freezing (1931-1960) in capital or large city
'Area': land area in square miles
```

Life Expectancy Example Continued

```
> data(state)
# reassemble the data (add row names)
> statedata = data.frame(state.x77, row.names=state.abb)
> g = lm(Life.Exp ~ ., data=statedata)
```

```
> summary(g)
             Estimate Std.Error t value Pr(>|t|)
Intercept   7.094e+01 1.748e+00  40.586  < 2e-16
Population  5.180e-05 2.919e-05   1.775   0.0832
Income     -2.180e-05 2.444e-04  -0.089   0.9293
Illiteracy  3.382e-02 3.663e-01   0.092   0.9269
Murder     -3.011e-01 4.662e-02  -6.459 8.68e-08
HS.Grad     4.893e-02 2.332e-02   2.098   0.0420
Frost      -5.735e-03 3.143e-03  -1.825   0.0752
Area       -7.383e-08 1.668e-06  -0.044   0.9649
---
Residual standard error:   0.7448 on 42 degrees of freedom
Multiple R-Squared: 0.7362     Adjusted R-squared: 0.6922
F-statistic: 16.74 on 7 and 42 DF      p-value: 2.534e-10
```

```
## Backward elimination - drop largest p-value
> g = update(g, . ~ . - Area)
> summary(g)
             Estimate Std.Error t value Pr(>|t|)
Intercept   7.099e+01 1.387e+00  51.165  < 2e-16
Population  5.188e-05 2.879e-05   1.802   0.0785
Income     -2.444e-05 2.343e-04  -0.104   0.9174
Illiteracy  2.846e-02 3.416e-01   0.083   0.9340
Murder     -3.018e-01 4.334e-02  -6.963 1.45e-08
HS.Grad     4.847e-02 2.067e-02   2.345   0.0237
Frost      -5.776e-03 2.970e-03  -1.945   0.0584
Residual standard error:   0.7361 on 43 degrees of freedom
Multiple R-Squared: 0.7361      Adjusted R-squared: 0.6993
F-statistic: 19.99 on 6 and 43 DF      p-value: 5.362e-11
```

```
## Continue dropping
> g = update(g, . ~ . - Illiteracy)
> summary(g)
Coefficients:
            Estimate Std.Error t value Pr(>|t|)
Intercept  7.107e+01 1.029e+00  69.067  < 2e-16
Population 5.115e-05 2.709e-05   1.888   0.0657
Income    -2.477e-05 2.316e-04  -0.107   0.9153
Murder    -3.000e-01 3.704e-02  -8.099 2.91e-10
HS.Grad    4.776e-02 1.859e-02   2.569   0.0137
Frost     -5.910e-03 2.468e-03  -2.395   0.0210
Residual standard error:   0.7277 on 44 degrees of freedom
Multiple R-Squared: 0.7361      Adjusted R-squared: 0.7061
F-statistic: 24.55 on 5 and 44 DF       p-value: 1.019e-11
```

```
## Continue dropping
> g = update(g, . ~ . - Income)
> summary(g)
Coefficients:
            Estimate Std.Error t value Pr(>|t|)
Intercept  7.103e+01 9.529e-01  74.542  < 2e-16
Population 5.014e-05 2.512e-05   1.996  0.05201
Murder    -3.001e-01 3.661e-02  -8.199 1.77e-10
HS.Grad    4.658e-02 1.483e-02   3.142  0.00297
Frost     -5.943e-03 2.421e-03  -2.455  0.01802
Residual standard error:   0.7197 on 45 degrees of freedom
Multiple R-Squared: 0.736       Adjusted R-squared: 0.7126
F-statistic: 31.37 on 4 and 45 DF       p-value: 1.696e-12
```

```
## Borderline case... would keep for prediction,
## but try dropping
> g = update(g, . ~ . - Population)
> summary(g)
Coefficients:
           Estimate Std.Error t value Pr(>|t|)
Intercept 71.036379  0.983262  72.246  < 2e-16
Murder    -0.283065  0.036731  -7.706 8.04e-10
HS.Grad    0.049949  0.015201   3.286  0.00195
Frost     -0.006912  0.002447  -2.824  0.00699
Residual standard error:   0.7427 on 46 degrees of freedom
Multiple R-Squared: 0.7127      Adjusted R-squared: 0.6939
F-statistic: 38.03 on 3 and 46 DF       p-value: 1.634e-12
```

```
## Cannot conclude other predictors have no effect
## on response: e.g., Illiteracy
> summary(lm(Life.Exp ~ Illiteracy + Murder
     + Frost, statedata))
Coefficients:
           Estimate Std.Error t value Pr(>|t|)
Intercept 74.556717  0.584251 127.611  < 2e-16
Illiteracy-0.601761  0.298927  -2.013  0.04998
Murder    -0.280047  0.043394  -6.454 6.03e-08
Frost     -0.008691  0.002959  -2.937  0.00517
Residual standard error:   0.7911 on 46 degrees of freedom
Multiple R-Squared: 0.6739      Adjusted R-squared: 0.6527
F-statistic: 31.69 on 3 and 46 DF       p-value: 2.915e-11
```

Remarks on Testing-based approaches

- Greedy. May miss the optimal model.
- Remember not to take *p*-values at face value (multiple testing).
- Variables not selected can still be correlated with the response, but they do not improve the fit enough to be included.
- Tend to pick smaller models than desirable for prediction purposes.

Criterion-based Model Selection

- General idea: choose the model that optimizes a criterion which balances goodness-of-fit and model size.
- No p-values involved
- Some theoretical guarantees
- Different methods use different goodness-of-fit measures and different penalties for model size

AIC and BIC

- Akaike information criterion (AIC)

$$\text{AIC} = n \ln(\text{RSS}/n) + 2(p+1)$$

  R function: `step(...,k=2)` (default)

- Bayes information criterion (BIC)

$$\text{BIC} = n \ln(\text{RSS}/n) + (p+1) \ln n$$

  R function: `step(..., k=log(n))`

Pick a model that minimizes AIC or BIC

Life Expectancy Example

```
> ## AIC
> g = lm(Life.Exp ~ ., data=statedata)
> step(g)
Start:  AIC= -22.18
 Life.Exp ~ Population + Income + Illiteracy +
   Murder + HS.Grad + Frost + Area
             Df Sum of Sq     RSS     AIC
- Area        1     0.001  23.298 -24.182
- Income      1     0.004  23.302 -24.175
- Illiteracy  1     0.005  23.302 -24.174
<none>                     23.297 -22.185
- Population  1     1.747  25.044 -20.569
- Frost       1     1.847  25.144 -20.371
- HS.Grad     1     2.441  25.738 -19.202
- Murder      1    23.141  46.438  10.305
```

```
Step:  AIC= -24.18
 Life.Exp ~ Population + Income + Illiteracy +
   Murder + HS.Grad + Frost
             Df Sum of Sq    RSS     AIC
- Illiteracy  1     0.004 23.302 -26.174
- Income      1     0.006 23.304 -26.170
<none>                    23.298 -24.182
- Population  1     1.760 25.058 -22.541
- Frost       1     2.049 25.347 -21.968
- HS.Grad     1     2.980 26.279 -20.163
- Murder      1    26.272 49.570  11.568
```

```
Step:  AIC= -26.17
 Life.Exp ~ Population + Income + Murder +
    HS.Grad + Frost

             Df Sum of Sq    RSS     AIC
- Income      1     0.006 23.308 -28.161
<none>                    23.302 -26.174
- Population  1     1.887 25.189 -24.280
- Frost       1     3.037 26.339 -22.048
- HS.Grad     1     3.495 26.797 -21.187
- Murder      1    34.739 58.041  17.457
```

```
Step:  AIC= -28.16
 Life.Exp ~ Population + Murder + HS.Grad +
    Frost

             Df Sum of Sq    RSS     AIC
<none>                    23.308 -28.161
- Population  1     2.064 25.372 -25.920
- Frost       1     3.122 26.430 -23.876
- HS.Grad     1     5.112 28.420 -20.246
- Murder      1    34.816 58.124  15.528

Coefficients:
(Intercept    Population    Murder     HS.Grad       Frost
  71.03       5.014e-05    -0.3001    4.658e-02  -5.943e-03
```

- BIC picked the same model.

Adjusted R^2

Recall

$$R^2 = 1 - \frac{RSS}{TSS}$$

Definition of adjusted R^2:

$$\begin{aligned} R_a^2 &= 1 - \frac{RSS/(n-(p+1))}{TSS/(n-1)} \\ &= 1 - \left(\frac{n-1}{n-(p+1)}\right)(1-R^2) \end{aligned}$$

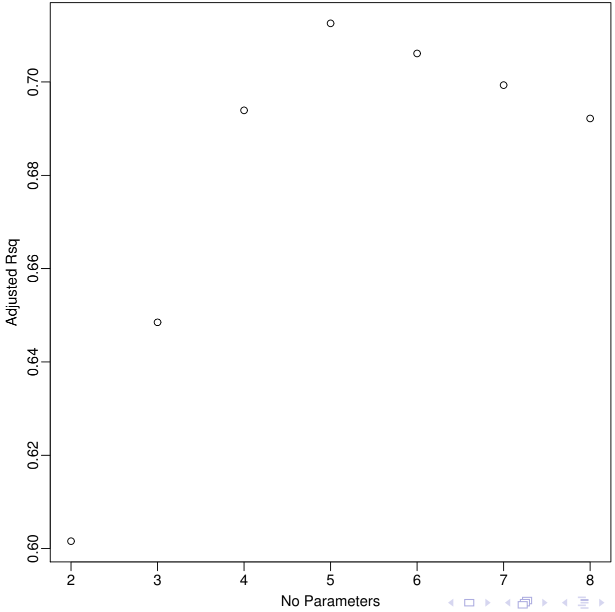
- Adding a predictor will not necessarily increase R_a^2
- Maximizing R_a^2 is equivalent to minimizing RSE $\hat{\sigma}$.

Life Expectancy Example

```
> ## Adjusted R^2
> library(leaps)
> b = regsubsets(Life.Exp ~ ., data=statedata)
> summary(b)
Selection Algorithm: exhaustive
         Population Income Illiteracy Murder HS.Grad Frost Area
1  ( 1 ) " "        " "    " "        "*"    " "     " "   " "
2  ( 1 ) " "        " "    " "        "*"    "*"     " "   " "
3  ( 1 ) " "        " "    " "        "*"    "*"     "*"   " "
4  ( 1 ) "*"        " "    " "        "*"    "*"     "*"   " "
5  ( 1 ) "*"        "*"    " "        "*"    "*"     "*"   " "
6  ( 1 ) "*"        "*"    "*"        "*"    "*"     "*"   " "
7  ( 1 ) "*"        "*"    "*"        "*"    "*"     "*"   "*"
```

```
# plot adjusted R2 against p+1
> rs = summary(b)
> plot(2:8, rs$adjr2, xlab="No. of Parameters",
  ylab="Adjusted Rsq")

# select model with largest adjusted R2
> which.max(rs$adjr2)
[1] 4
```

Adjusted R^2 for the Life Expectancy Data



Mallows' C_p

Definition:

$$C_p = \frac{RSS_p}{\hat{\sigma}^2} + 2(p+1) - n$$

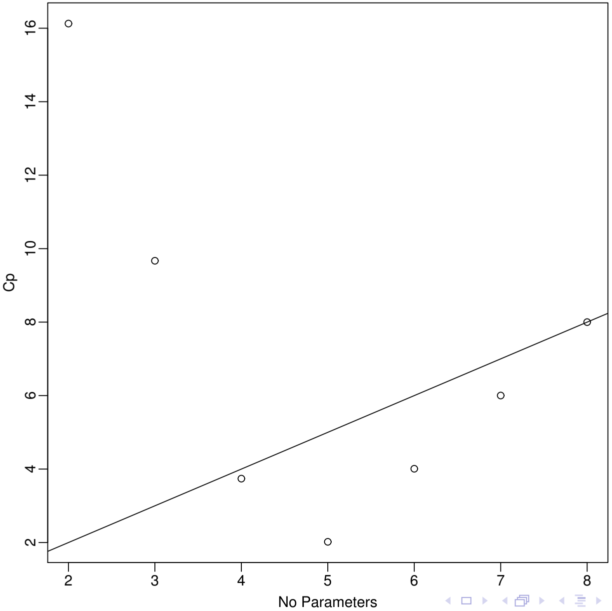
- $\hat{\sigma}^2$ is estimated from the model with all predictors
- RSS_p is from the model with p predictors
- Goal: minimize C_p.
- C_p around or less than $p+1$ indicates good fit.
- C_p estimates the mean squared error (MSE)

$$\frac{1}{\sigma^2} \sum_i E(\hat{y}_i - Ey_i)^2$$

Life Expectancy Example

```
> ## Mallows Cp
> library(leaps)
> b = regsubsets(Life.Exp ~ ., data=statedata)
> rs = summary(b)

> which.min(rs$cp)
[1] 4

> plot(2:8, rs$cp, xlab="No. Parameters",
      ylab="Cp")
> abline(0, 1)
```

C_p Plot for the Life Expectancy Data



Variable Selection Summary

- Variable selection methods are sensitive to outliers
- Generally, criterion-based methods are preferred
- It may happen that several models provide very similar fit
- If models with similar fit lead to very different conclusions, the data are ambiguous
- If conclusions are similar, choose a simpler model and/or predictors that are easier to measure

Chapter 11: Shrinkage Methods

Outline

- Ridge regression
- Lasso
- (skip PLS and PCR)

Ridge Regression

Penalizing the square of the coefficients

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- The coefficients $\hat{\boldsymbol{\beta}}^{\text{ridge}}$ are shrunken towards zero.
- $\lambda \geq 0$ is a tuning parameter.
- λ controls the amount of shrinkage.
- What happens if $\lambda \to 0$?
- What happens if $\lambda \to \infty$?

Equivalent Formulation

$$\min_{\boldsymbol{\beta}} \quad \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$
$$\text{subject to} \quad \sum_{j=1}^{p}\beta_j^2 \le s$$

- Explicitly constraint the size of the coefficients.

When there are many highly correlated variables

- $\hat{\beta}^{\text{ols}}$ may have a large coefficient on one variable and a similarly large negative coefficient on its correlated variable (Unstable).
- In ridge regression, the size constraint tries to avoid this phenomenon.

NOTE: The ridge estimate is not equivariant under scaling of the predictors.
Often standardize the predictors first.

Solution for Ridge Regression

- The solution is

$$\hat{\beta}^{\text{ridge}} = (\boldsymbol{X}^{\top}\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}^{\top}\boldsymbol{y}$$

- $\hat{\beta}$ is linear in $\boldsymbol{y}$.
- $\hat{\beta}$ is biased.
- Even if $\boldsymbol{X}$ is not full-rank, $(\boldsymbol{X}^{\top}\boldsymbol{X} + \lambda \boldsymbol{I})$ is invertible.
- $\hat{\beta}^{\text{ridge}}$ has smaller variance than the OLS, thus may have smaller mean square error (MSE).

Shrinkage in Ridge

Suppose orthonormal design ($\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{I}$). Then $\hat{\beta}^{\text{ols}} = \boldsymbol{X}^\top \boldsymbol{y}$, and

$$(\boldsymbol{y} - \boldsymbol{X}\beta)^\top(\boldsymbol{y} - \boldsymbol{X}\beta) = \text{constant} + \sum_{j=1}^{p}(\beta_j - \hat{\beta}_j^{\text{ols}})^2.$$

Then ridge regression minimizes

$$\sum_{j=1}^{p}(\beta_j - \hat{\beta}_j^{\text{ols}})^2 + \lambda \sum_{j=1}^{p} \beta_j^2.$$

Equivalent to the component-wise minimization

$$\min_{\beta_j}(\beta_j - \hat{\beta}_j^{\text{ols}})^2 + \lambda\beta_j^2 \Longrightarrow \hat{\beta}_j^{\text{ridge}} = \frac{1}{1+\lambda}\hat{\beta}_j^{\text{ols}}.$$

Shrinkage in Ridge

- Shrink the estimate towards zero by a positive constant less than 1
- $\mathrm{Var}(\hat{\beta}_j^{\mathrm{ridge}}) = \frac{1}{(1+\lambda)^2}\mathrm{Var}(\hat{\beta}_j^{\mathrm{ols}})$.
- $\lambda \uparrow$, shrinkage $\uparrow$, bias $\uparrow$, variance $\downarrow$
- $\lambda \downarrow$, shrinkage $\downarrow$, bias $\downarrow$, variance $\uparrow$.

Model Assessment

Objectives:

1. Choose a value of a tuning parameter for a technique.
2. Estimate the prediction performance of a given model.

- For both of these purposes, the best approach is to run the procedure on an independent test set, if one is available.
- If possible one should use different test data for (1) and (2) above: a validation set for (1) and a test set for (2).

Cross-Validation

- Often there is insufficient data to create a separate validation or test set; setting some data aside for validation is possible, but affects the accuracy of training estimates
- In this instance, K-fold cross-validation is useful.

1. Divide the data into K disjoint subsets.
2. Use subsets $2, \ldots, K$ as training data and subset 1 as validation data. Compute the PE on subset 1.
3. Repeat for each subset.
4. Average the result.

LASSO

Least absolute shrinkage and selection operator (Chen, Donoho and Saunders 1996; Tibshirani 1996)

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- Shrinkage
- Sparsity: some fitted coefficients are exactly zero

Continuous variable selection

Equivalent Formulation

$$\min_{\boldsymbol{\beta}} \quad \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2$$

$$\text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq s$$

Soft Thresholding

When $\boldsymbol{X}$ is orthonormal, we can minimize over $\boldsymbol{\beta}$ componentwise

$$\min_{\beta_j} \; (\beta_j - \hat{\beta}_j^{\text{ols}})^2 + \lambda|\beta_j|.$$

The solution is

$$\hat{\beta}_j^{\text{lasso}} = \begin{cases} \hat{\beta}_j^{\text{ols}} - \frac{\lambda}{2} & \text{if } \hat{\beta}_j^{\text{ols}} > \frac{\lambda}{2} \\ 0 & \text{if } |\hat{\beta}_j^{\text{ols}}| \le \frac{\lambda}{2} \\ \hat{\beta}_j^{\text{ols}} + \frac{\lambda}{2} & \text{if } \hat{\beta}_j^{\text{ols}} < -\frac{\lambda}{2} \end{cases}$$

$$= \text{sign}(\hat{\beta}_j^{\text{ols}}) \cdot \left(|\hat{\beta}_j^{\text{ols}}| - \frac{\lambda}{2} \right)_+$$

- Lasso shrinks large coefficients by a constant.
- Lasso truncates small coefficients to zero.

Ridge vs Lasso