

Model Selection, Diagnostics, Transformation & Nonlinear Models

Objectives

- Traditional & Criterion-Based Model Selection
- Statistics & Plots for Model Diagnostics
- Box-Cox Transformation
- Fitting Nonlinear Models

Traditional Testing-Based Procedures

Backward Elimination

- Begins with all variables in the model.
 1. Remove variable with the smallest insignificant partial F value.
 2. Remove variable with the smallest insignificant partial F value in the presence of those variables in the model.
 3. Continue
- The only technique among F, B & S that guarantees consideration of the full k -variable model.
- Inflexible. Once out, cannot come back in.

Forward Selection

- Begins with no variable in the model.
 1. Add variable that has the largest significant F value.
 2. Add variable that has the largest significant partial F value given that x_1 is in the model.
 3. Continue.
- “Best” variable is added first.
- Inflexible. Once in, cannot come out.

Stepwise Regression

- Modification of F, adds variables just like F.
 1. At each stage, all variables in the model must submit to a partial F -test.
 2. Drop the variable with the smallest insignificant partial F value (if any).
 3. Continue adding/dropping (this is the difference from forward selection).
- More flexible than F.

Critic of F, B & S (product of the 1950's)

1. Leaves many models not computed and therefore is not guaranteed to find the “best” model.
2. Results depend on the significance levels.

3. Partial F -test may not be valid a test. $F = \frac{RSS(x_j | \text{all other variables})}{s^2}$, where s^2 may be an estimate of σ^2 from a very incomplete “underfit” model (i.e., large s^2). As a result, F is too small, and it too often finds “deserving” variables insignificant. To counter this, we set a liberal entry significance level (for example, 0.25~0.30 than usual 0.05).

Ex 1. Backward elimination.

```
> library(faraway)
> statedata <- data.frame(state.x77,row.names=state.abb)
> dim(statedata)
[1] 50 8
> statedata <- data.frame(state.x77,row.names=state.abb)
> dim(statedata)
[1] 50 8
> head(statedata)
  Population Income Illiteracy Life.Exp Murder HS.Grad Frost Area
AL       3615   3624         2.1   69.05   15.1   41.3    20 50708
AK        365   6315         1.5   69.31   11.3   66.7   152 566432
AZ       2212   4530         1.8   70.55    7.8   58.1    15 113417
AR        2110   3378         1.9   70.66   10.1   39.9    65 51945
CA      21198   5114         1.1   71.71   10.3   62.6    20 156361
CO       2541   4884         0.7   72.06    6.8   63.9   166 103766
> model1 <- lm(Life.Exp ~., data=statedata)      #Begin with all of the x-variables.
> summary(model1)
```

Coefficients:

	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

```
> model1 <- update(model1, ~.-Area)
> summary(model1)
```

Coefficients:

	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

```
> model1 <- update(model1, ~.-Illiteracy)
> summary(model1)

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
```

```
> model1 <- update(model1, ~.-Income)
> summary(model1)

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
```

```
> model1 <- update(model1, ~.-Population)
> summary(model1)

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
```

Process stops here because all remaining terms are significant. There is nothing more to remove.

Criterion-Based Procedures

Below, p = number of x -variables+1. For example, in a model of y vs. seven x -variables, $p=8$.

- $AIC = -2 \cdot \max \log\text{-likelihood} + 2(p + 1)$ $\{p + 1$ because variance is also estimated. $\}$

Use this AIC when variance is estimated by the maximum likelihood method.

- $AIC = n \cdot \log\left(\frac{RSS}{n}\right) + (2p)$

Use this AIC when the variance is estimated by the least-squares method.

- $AIC_c = AIC + \left\{ \frac{2(p+1)(p+2)}{n - (p+1) - 1} \right\}$

Use this second-order AIC when the the sample size is small in comparison to the number of estimated parameters [Burnham & Anderson, 2002], recommend its use when $\frac{n}{p+1} < 40$.

- $BIC = -2 \cdot \max \log\text{-likelihood} + \{\log(n)\} \cdot (p+1)$

Ex 2. AIC, AICc & BIC

```
library(faraway)
data(state)
statedata <- data.frame(state.x77, row.names=state.abb)
dim(statedata)
[1] 50 8
model1 <- lm(Life.Exp ~., data=statedata)
step(model1) #AIC for the "least squares" case is calculated here.
#AIC = nxlog(RSS/n)+(2xp)
Start: AIC=-22.18
```

	Df	Sum of Sq	RSS	AIC
- Area	1	0.0011	23.298	-24.182
- Income	1	0.0044	23.302	-24.175
- Illiteracy	1	0.0047	23.302	-24.174
<none			23.297	-22.185
- Population	1	1.7472	25.044	-20.569
- Frost	1	1.8466	25.144	-20.371
- HS.Grad	1	2.4413	25.738	-19.202
- Murder	1	23.1411	46.438	10.305

#AIC = 50xlog(23.297/50)+(2x8)=-22.185
#23.297 is the RSS of model1

```
Step: AIC=-24.18
Life.Exp ~ Population + Income + Illiteracy + Murder + HS.Grad + Frost
```

	Df	Sum of Sq	RSS	AIC
- Illiteracy	1	0.0038	23.302	-26.174
- Income	1	0.0059	23.304	-26.170
<none			23.298	-24.182
- Population	1	1.7599	25.058	-22.541
- Frost	1	2.0488	25.347	-21.968
- HS.Grad	1	2.9804	26.279	-20.163
- Murder	1	26.2721	49.570	11.569

```
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	17.456

```
Step: AIC=-28.16
```

```

Life.Exp ~ Population + Murder + HS.Grad + Frost
      Df Sum of Sq  RSS   AIC
<none                23.308 -28.161      #AIC =50xlog(23.308/50)+(2x5)=-28.161
- Population    1      2.064 25.372 -25.920
- Frost         1      3.122 26.430 -23.877
- HS.Grad       1      5.112 28.420 -20.246
- Murder        1     34.816 58.124  15.528

Coefficients:
(Intercept)  Population      Murder      HS.Grad      Frost 
7.103e+01    5.014e-05   -3.001e-01    4.658e-02   -5.943e-03

> model1 <- lm(Life.Exp~Population+Income+Illiteracy+Murder+HS.Grad+Frost+Area)
> model2 <- lm(Life.Exp~Population+Income+Illiteracy+Murder+HS.Grad+Frost)
> model3 <- lm(Life.Exp~Population+Income+Murder+HS.Grad+Frost)
> model4 <- lm(Life.Exp~Population+Murder+HS.Grad+Frost)
> AIC(model1, model2, model3, model4)  #AIC for the "maximum likelihood" case is calculated
                                     #AIC = (-2xlogLik)+2x(p+1)

      df      AIC
model1  9 121.7092      #AIC = (-2x-51.85461)+(2x9) = 121.7092
model2  8 119.7116      #AIC = (-2x-51.85578)+(2x8) = 119.7116
model3  7 117.7196      #AIC = (-2x-51.85891)+(2x7) = 117.7196
model4  6 115.7326      #AIC = (-2x-51.86631)+(2x6) = 115.7326
> logLik(model1)
'log Lik.' -51.85461 (df=9)
> logLik(model2)
'log Lik.' -51.85578 (df=8)
> logLik(model3)
'log Lik.' -51.85981 (df=7)
> logLik(model4)
'log Lik.' -51.86631 (df=6)

> library(AICcmodavg)
> AICc(model1)      #AICc = AIC + {(2x9x10)/(50-9-1)} = 121.7092 + 4.5 = 126.2092
[1] 126.2092
> AICc(model2)
[1] 123.2238
> AICc(model3)
[1] 120.3863
> AICc(model4)
[1] 117.6861

> BIC(model1,model2,model3,model4)
      df      BIC
model1  9 138.9174      #BIC = (-2x-51.85461)+(log(50)x9) = 138.9174
model2  8 135.0077
model3  7 131.1038
model4  4 132.7372

```

More Criteria for Model Selection

- Coefficient of determination $R^2 = 1 - \frac{RSS}{TSS}$
 R^2 shows the percentage of total variability explained by the model.
- Adjusted R^2 , i.e., $R_a^2 = 1 - \frac{RSS/df_E}{TSS/df_{total}} = 1 - \frac{RSS/(n-p)}{TSS/(n-1)}$

- Mallows' $C_p = \frac{RSS_p}{\hat{\sigma}_{full\ model}^2} + 2p - n$

In an ideal case, $\frac{RSS_p}{\hat{\sigma}_{full\ model}^2} = n - p$, which means $C_p = p$.

Ex 3. R_a^2 and C_p , Use of all possible subsets to find the best model.

```
> library(leaps)
> model1 <- regsubsets(Life.Exp~.,data=statedata) #nvmax=5, e.g., specifies how many models to check
> (rs <- summary(model1))
Subset selection object
Call: regsubsets.formula(Life.Exp ~ ., data = statedata)
7 Variables (and intercept)
      Forced in Forced out
Population    FALSE      FALSE
Income        FALSE      FALSE
Illiteracy    FALSE      FALSE
Murder         FALSE      FALSE
HS.Grad        FALSE      FALSE
Frost          FALSE      FALSE
Area           FALSE      FALSE
1 subsets of each size up to 7
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 ) "*" "*" "*" "*" "*" "*" "*" "*"
> names(rs)
[1] "which" "rsq" "rss" "adjr2" "cp" "bic" "outmat" "obj"
> par(mfrow=c(1,2))
> plot(2:8,rs$adjr2, pch=16,col=2,xlab="# of Parameters",ylab="Adjusted R^2")
> text(2:8,rs$adjr2,labels=round(rs$adjr2,4),pos=1,offset=0.3,cex=0.7)
> plot(2:8,rs$cp,pch=16,col=2,xlab="# of Parameters",ylab="Cp"); abline(0,1)
> text(2:8,rs$cp,labels=round(rs$cp,2),pos=1,offset=0.3,cex=0.7)
> #pos=1 places texts below symbols, offset=0.3 is about how far below,
> #cex=0.7 means 70% of the usual size.
```

(See Fig. 1 & 2 below.) Adjusted R^2 shows a model with $p = 5$ is the best, and that's the model with Murder, HS.Grad, Frost and Population. C_p shows that $p = 4$ is the best, and that's the model with Murder, HS.Grad and Frost. The last value of $C_p = 8$ is meaningless because C_p is always p for the "last" model. R offers some other additional plots for visual model selection. Recall that a model with $p = 5$ was the best using AIC , so we will be using this model as the best and move on to the diagnostics.

```
> library(car)
> plot(model1,scale="adjr2")
> subsets(model1,statistics="adjr2") #default statistics is BIC.
```

Model Diagnostics

Ex 4. Default Model Diagnostics in R

```
> library(faraway)
> data(state)
> statedata <- data.frame(state.x77,row.names=state.abb)
> attach(statedata)
> model4 <- lm(Life.Exp~Population+Murder+HS.Grad+Frost)
> summary(model4)
```

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

```
> par(mfrow=c(2,2))
> plot(model4,pch=16,col=2)
> library(car)
> influenceIndexPlot(model4,id.n=5,id.cex=0.7,col=2)
> influencePlot(model4,col=2)
```

Four Default Diagnostic Plots:

- **Residuals vs. Fitted values:** *random scatter around zero* is what we're looking for. Any pattern is BAD. There are specific ways of dealing with particular patterns. For example, if residuals show “curvature,” we need to include higher order x -terms in the model. In case of “megaphone” shape of residuals (i.e., increasing variabilities of residuals), we need to transform y 's to stabilize the unequal variances. Box-Cox transformation is widely used to find an optimal transformation for such a case.
- **Normal QQ plot of Residuals:** *Straight line* means residuals are considered normal (= good news).
- **Scale-Location plot:** Scattered around *horizontal line* means good news. If there is a significant slope, we need to transform y 's.
- **Residuals vs. Leverage plot:** Because R plots the “standardized” residuals, anything greater than 2 or less than -2 are “outliers.” It also identifies any observations that have particularly high *leverages*.

Othe Plots & Statistics to Identify Outliers:

- **“hat” diagonal values** indicate *how far* a data point is away from the *data center*. Data points are known to have *high leverages* when they have **“hat” diagonal values** $> \frac{2p}{n}$. Recall $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the “Hat” matrix in linear models, where \mathbf{X} is the matrix of independent variables with the first-column of 1’s. Since $Var(\hat{\epsilon}) = \sigma^2(1 - h_{ii})$, high h_{ii} values mean small variance of residuals, i.e., model fit will be “forced” close to the data point. This is called *high leverage*.
- **Cook’s D statistic** measures “overall” influence of a data point, i.e., it’s used to identify observations with high *influences*. A data point is considered having a very **high influence** when $D > \frac{4}{n - p}$. It’s defined as:

$$D_i = \frac{(\hat{y} - \hat{y}_{-i})^T (\hat{y} - \hat{y}_{-i})}{p\hat{\sigma}^2}$$

It measures the *change* in the fit if a single case (i.e., the i th case) is removed from the dataset, thus a high D value means such a data point can cause a big change in the model fit.

- **R studentized residual** shows whether a point goes with the “trend” or not. (i.e., it detects *outliers*). A data point is a “positive” outlier if > 2 , and a “negative” outlier if < -2 .

Statistics to Identify Multicollinearity:

- **VIF** (variance inflation factor) indicates whether certain x -variable is fraught with collinearity with other x -variables. A variable is said to have unbearably high collinearity when its **VIF value** > 10 . Recall VIF of the i th x -variable is defined as $VIF_i = \frac{1}{1 - R_i^2}$, where R_i^2 is the R^2 value when you fit a regression model of x_i using all other x -variables. As a result, high VIF value means that particular x_i variable is highly correlated with other x -variables (i.e., x_i has a high multicollinearity).

```
> library(faraway)
> vif(model4)
Population      Murder      HS.Grad      Frost
1.189835      1.727844      1.356791      1.498077
```

Here, no variable has VIF greater than 2, the model does NOT have any multicollinearity problem.

Box-Cox Transformation to Stabilize the Variance (of y)

Sometimes a good transformation of the response variable (y) would stabilize the unequal variance. The BoxCox transformation offers a simple solution. The idea is to find an optimal power transformation of y , so that the *transformed* y maximizes the (log) likelihood. A specified set of explanatory variables will be fitted to $\frac{y^\lambda - 1}{\lambda}$. The Box-Cox function in R checks λ values between -2 and 2 by default, and you can specify different values of λ as well. Shown below are how y 's are transformed for various values of λ . In case of $\lambda = -1$, the dependent variable becomes $\frac{y^{-1} - 1}{-1} = 1 - \frac{1}{y}$.

- $\lambda = -1$ means (negative) inverse transformation, i.e., use $-y^{-1}$ i.e., $-\frac{1}{y}$
- $\lambda = 0$ means log transformation, i.e., use $\log(y)$
- $\lambda = 0.5$ means square-root transformation, i.e., use \sqrt{y}
- $\lambda = 2$ means 2nd power transformation, i.e., use y^2

Ex 5. Box-Cox transformation of y to stabilize unequal variance of y .

```
> library(faraway)
> dim(gala)
[1] 30 7
> head(gala)
```

	Species	Endemics	Area	Elevation	Nearest	Scruz	Adjacent
Baltra	58	23	25.09	346	0.6	0.6	1.84
Bartolome	31	21	1.24	109	0.6	26.3	572.33
Caldwell	3	3	0.21	114	2.8	58.7	0.78
Champion	25	9	0.10	46	1.9	47.4	0.18
Coamano	2	1	0.05	77	1.9	1.9	903.82
Daphne.Major	18	11	0.34	119	8.0	8.0	1.84

```
> attach(gala)
> model1 <- lm(Species~Area+Elevation+Nearest+Scruz+Adjacent)
> par(mfrow=c(2,2))
> plot(model1)      #See Fig. 7
> library(MASS)
> boxcox(model1)    #See Fig. 6
> model2 <- lm(Species^(1/3)~Area+Elevation+Nearest+Scruz+Adjacent)
> par(mfrow=c(2,2))
> plot(model2)      #See Fig. 7
> boxcox(model2)    #See Fig. 6
> par(mfrow=c(1,2))
> boxcox(model1); boxcox(model2) #See Fig. 6
```

Popular Non-Linear Models

Some useful non-linear models are shown below.

Name	Equation
Asymptotic functions	
• Michaelis-Menten	$y = \frac{ax}{1 + bx}$
• 2-parameter asymptotic exponential	$y = a(1 - e^{-bx})$
• 3-parameter asymptotic exponential	$y = a - be^{-bx}$
S-shaped functions	
• 2-parameter logistic	$y = \frac{e^{a+bx}}{1 + e^{a+bx}} = \frac{1}{1 + e^{-(a+bx)}}$
• 3-parameter logistic	$y = \frac{a}{1 + be^{-cx}}$
• 4-parameter logistic	$y = a + \frac{b-a}{1 + e^{(c-x)/d}}$
• Weibull	$y = a - be^{-cx^d}$
• Gompertz	$y = ae^{-be^{-cx}}$
Humped curves	
• Ricker curve	$y = axe^{-bx}$
• 1st-order compartment	$y = ke^{-e^ax} - e^{-e^bx}$
• Bell-shaped	$y = ae^{(- bx ^2)}$
• Biexponential	$y = ae^{bx} - ce^{-dx}$

Ex 1. For the data shown in Fig. 1, let's fit a nonlinear model of

$$y = a - be^{-cx}$$

The slightly tedious thing is that you have to specify initial guesses for the values of the parameters a, b and c (note, however, that some common non-linear models have “self-starting” versions in R which bypass this step; see Ex 2 below). Let's plot the data to work out sensible starting values. It always helps in cases like this to work out the equation's “behaviour at the limits” – that is to say, to find the values of y when $x \rightarrow 0$ and when $x \rightarrow \infty$. For $x \rightarrow \infty$, we have $y \rightarrow a$, and for $x \rightarrow 0$, we have $y \rightarrow a - b$. That is to say, we can take the initial values for $a \approx 150$, and $b \approx 150$.

Our guess at the value of c is slightly harder. Where the curve is rising most steeply, **jaw** length is about 40 where **age** is 5. Rearranging the equation gives

$$c = -\frac{\log\{(a-y)/b\}}{x} \approx -\frac{\log\{(150-40)/150\}}{5} = 0.06$$

Now that we have the three parameter estimates, we have R work out the rest:

```
> data1 <- read.table("U:\\STAT510\\jaws.txt",header=T)
> dim(data1)
[1] 54 2
> head(data1)
      age      bone
```

```

1  0.000000  0.00000
2  5.112000 20.22000
3  1.320000 11.11130
4 35.240000 140.65000
5  1.632931 26.15218
6  2.297635 10.00100
> attach(data1)
> plot(age,bone,pch=21,col="red",bg="green")
> model1 <- nls(bone~a-b*exp(-c*age), start=list(a=150,b=150,c=0.06))
> summary(model1)

```

Formula: bone ~ a - b * exp(-c * age)

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
a	115.2528	2.9139	39.55	< 2e-16 ***
b	118.6874	7.8925	15.04	< 2e-16 ***
c	0.1235	0.0171	7.22	2.44e-09 ***

Residual standard error: 13.21 on 51 degrees of freedom

Number of iterations to convergence: 5

Achieved convergence tolerance: 4.758e-06

```

> x <- seq(0,50,0.01)
> y <- predict(model1,list(age=x))
> lines(x,y,col="blue",lwd=2)

> SST = sum((bone-mean(bone))^2)
> SSE = sum((summary(model1)$resid)^2)
> 1-(SSE/SST)
[1] 0.8492187

```

The model $y = a - be^{-cx}$ explained 84.9% of the total variation in bone length.

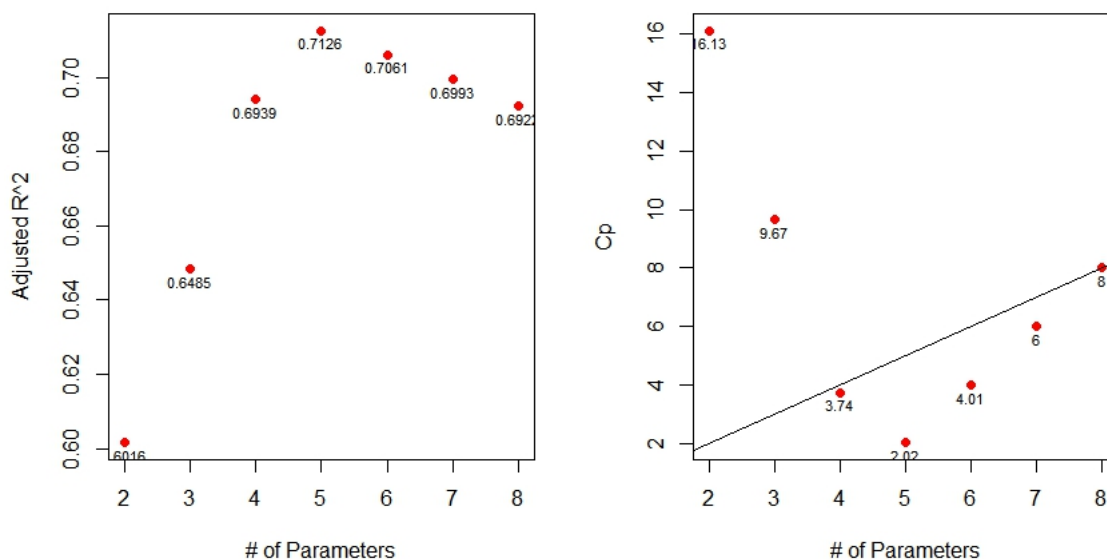


Figure 1: Finding the best model with Adjusted R^2 and C_p

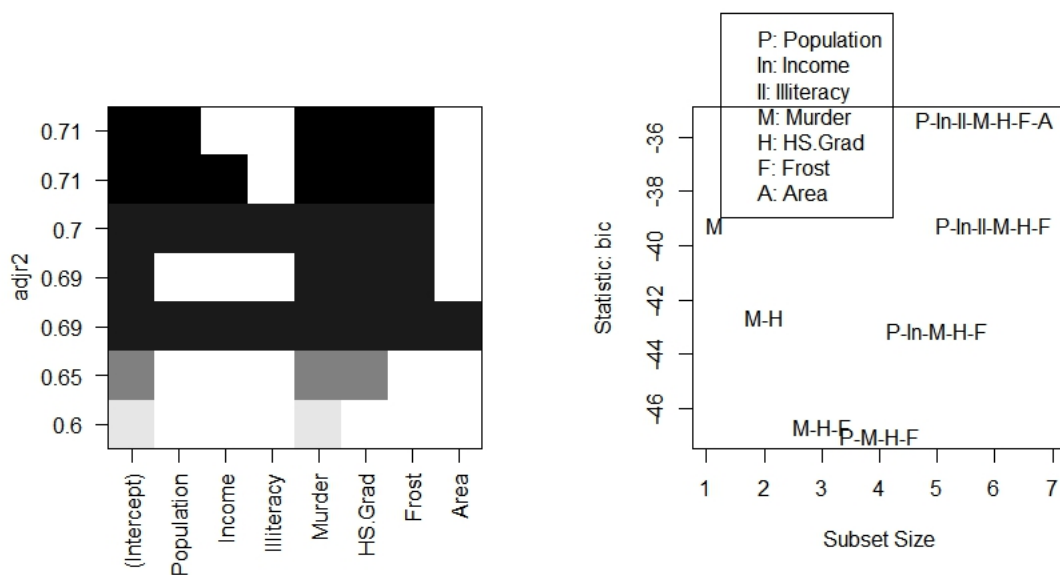


Figure 2: Another plot using Adjusted R^2 and BIC

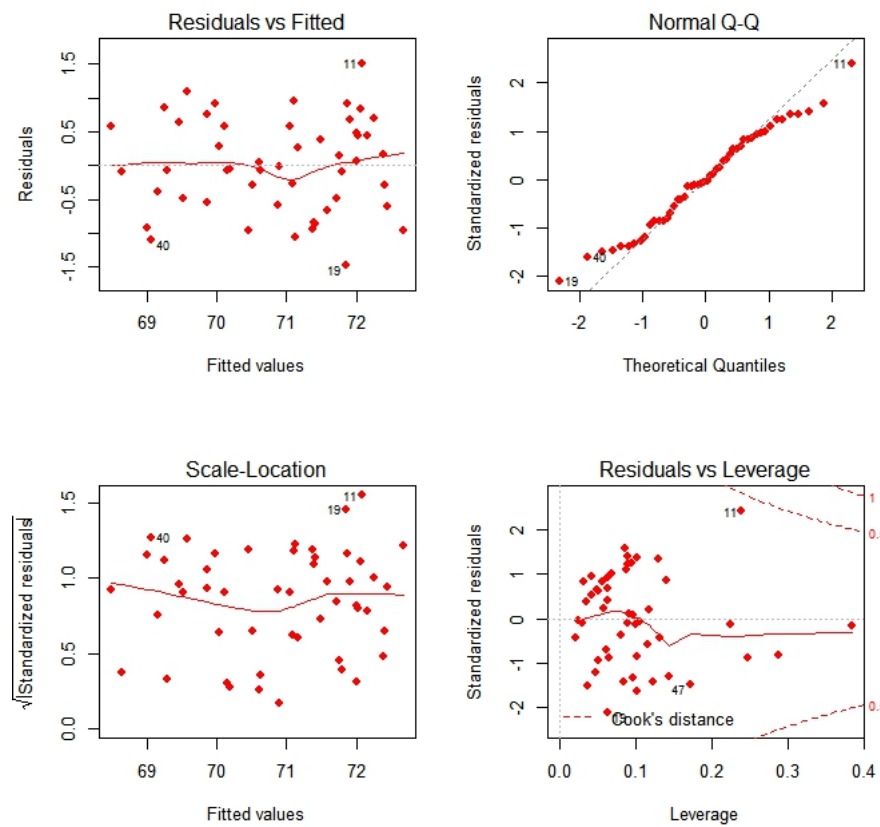


Figure 3: Default diagnostic plots in R by `> plot(model14, pch=16, col=2)`

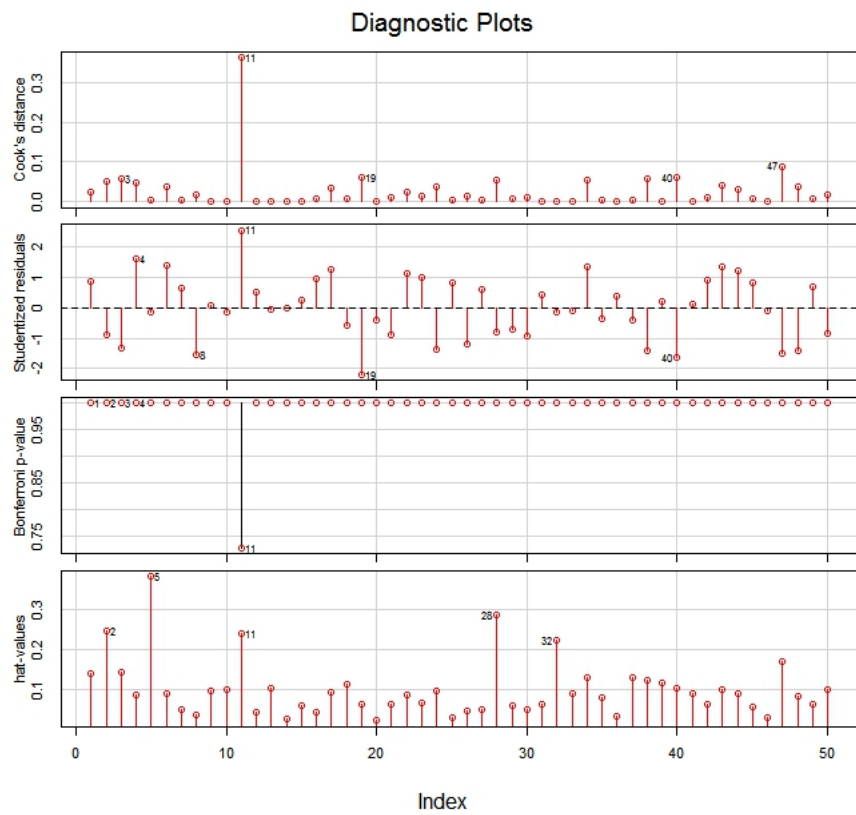


Figure 4: `> influenceIndexPlot(model4,id.n=5,id.cex=0.7,col=2)`

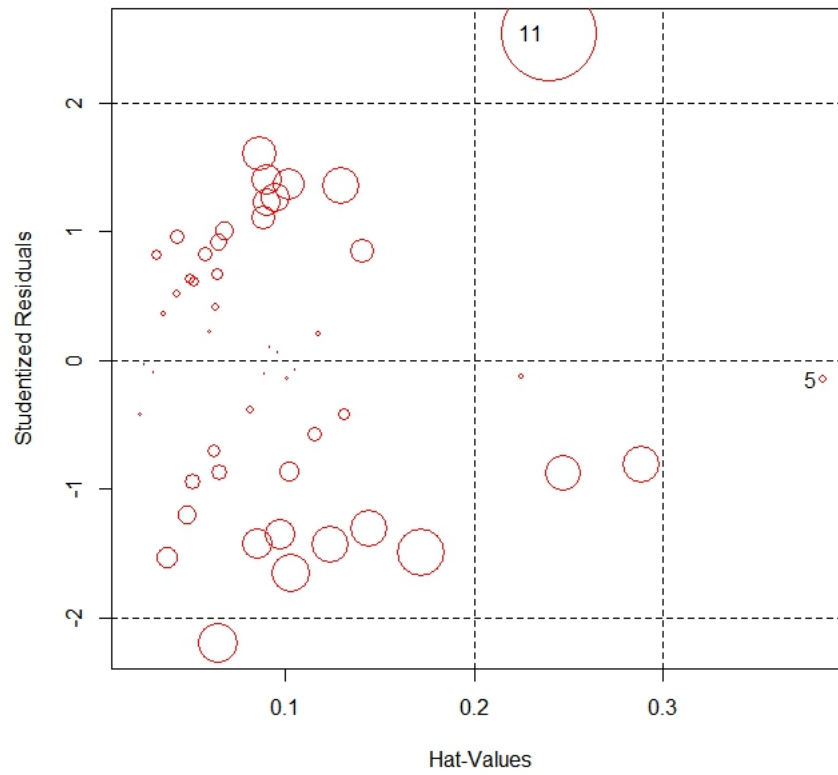


Figure 5: `> influencePlot(model4,col=2)`

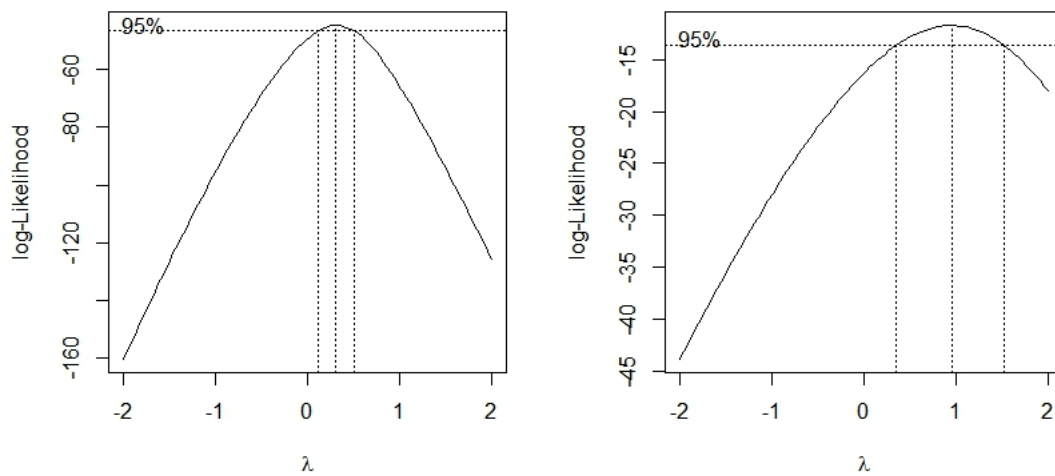


Figure 6: The first plot shows $\lambda \approx 0.33$, thus, we use $\sqrt[3]{y}$. Second plot shows $\lambda \approx 1$, thus, no need for any more transformation of y .

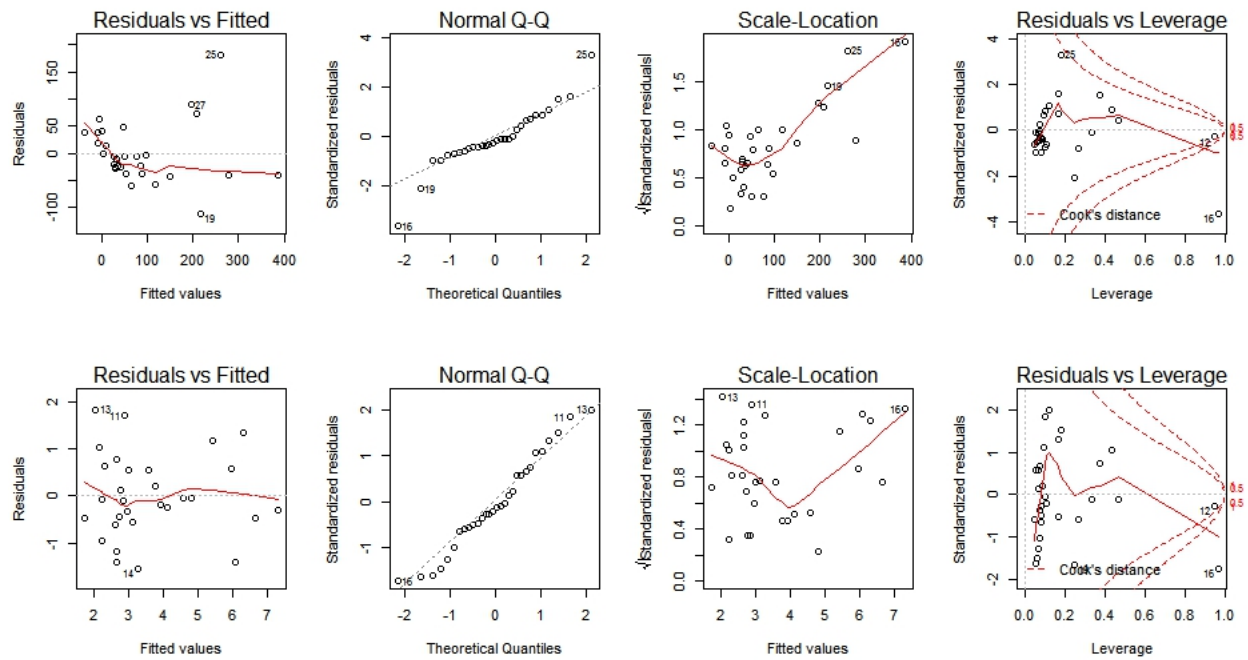


Figure 7: First set of four diagnostic plots show “unequal” variance of residuals (from `model1`). Second set of four diagnostic plots show more “equal” variance of residuals (from `model2`).

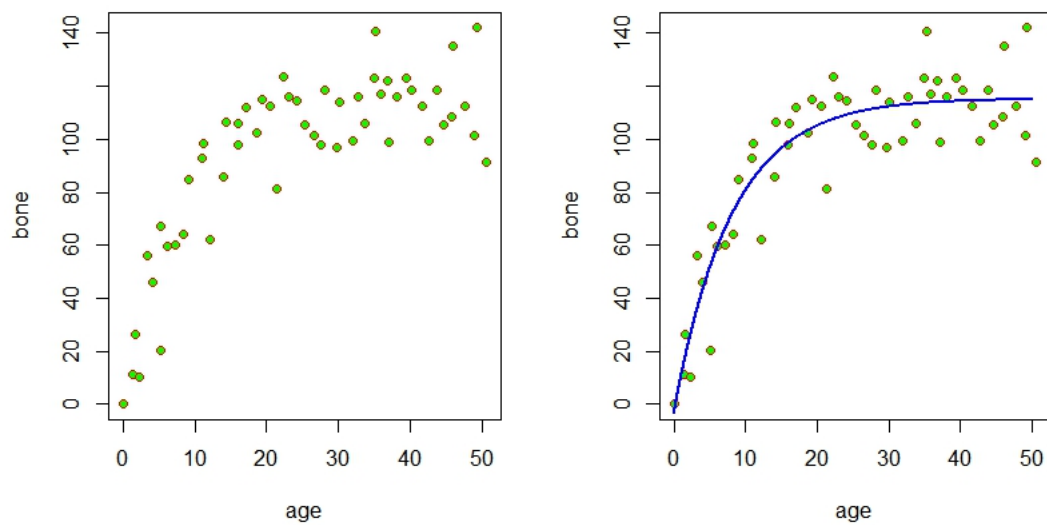


Figure 8: Plot of x vs. y and the same plot overlaid with the estimated nonlinear model