Nonlinear Regression Models

Why nonlinear? Non-linear regression deals with the effect of one or more predictor variables on a response variable when the effect of any predictor variable on the response variable is not linear.

Nonlinear models can be:

1. Intrinsically linear:
   - Linear with respect to the parameters
   - Can be transformed to a linear model, for example: $\mathbf{y} = \theta_0 \mathbf{x_1}^{\theta_1} \mathbf{x}_2^{\theta_2} \boldsymbol{\varepsilon}$

     transformed to:

$$\ln \mathbf{y} = \ln \theta_0 + \theta_1 \ln \mathbf{x_1} + \theta_2 \ln \mathbf{x}_2 + \ln \boldsymbol{\varepsilon}$$

Note: Some texts use Θ for the coefficients instead of β, to distinguish from linear models.

- This linear version of the model can be fit using linear least squares, with unbiased estimates of all coefficients of the nonlinear, EXCEPT that the estimate of the first coefficient β_0 will have a slight bias.
- Also, this can only be transformed if we consider the errors to be "multiplicative" rather than "additive" in the original linear model

2. Intrinsically nonlinear models
   - Not linear with respect to the parameters, for example:

$$\mathbf{y} = \theta_0 + \theta_1 \mathbf{x_1}^{\theta_2} \mathbf{x}_2^{\theta_3} \boldsymbol{\varepsilon}$$

In general, for any nonlinear function, we will use:

$$\mathbf{y} = f(\mathbf{x}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}$$

$$\mathbf{OR}$$

$$\mathbf{y} = f(\mathbf{x}, \boldsymbol{\beta}) + \boldsymbol{\varepsilon}$$

To estimate model parameters, intrinsically nonlinear models with additive error term are fit using nonlinear regression.

There are many forms of nonlinear models. Details of model forms can be found in standard text books. In this note we omit statistical theories of nonlinear regression models which are well covered by Bates and Watts (2007) and others.

Methods to Find Estimates of the Parameters

Estimates of coefficients and variances can be found using:

1. Nonlinear least squares: The objective is to find a set of coefficients that minimizes the sum of squared error (SSE, same as for OLS on linear models). The variances are then estimated separately.

2. Maximum likelihood: The objective is to find the set of parameters (coefficients and variances) that maximizes the likelihood that you would get the sample data.

Nonlinear Least Squares

Using the least squared error criterion to solve for the coefficients, the objective function is:

$$S(\hat{\boldsymbol{\theta}}) = \sum_{i=1}^{n} e_i^{\,2} = \sum_{i=1}^{n} \left[y_i - f(\mathbf{x}_i, \theta) \right]^2$$

Which is the sum of squared error (SSE) given particular values for the coefficients. To obtain the estimated coefficients that give the smallest SSE (minimum of this function), we could

1. get the partial first derivatives of this *S* function, with respect to each parameter,
2. set these equal to zero, and
3. solve, as with linear least squares.

However, a set of nonlinear equations (the partial first derivatives) will be obtained if the original nonlinear equation was intrinsically nonlinear. **We cannot obtain a single unique solution (no closed form).**

For example:

$$y_i = \beta_1 + \beta_2 e^{\beta_3 x_i} + \varepsilon_i$$

And then:

$$S(\hat{\boldsymbol{\beta}}) = \sum_{i=1}^{n} e_i^{\,2} = \sum_{i=1}^{n} (y_i - (\hat{\beta}_1 + \hat{\beta}_2 e^{\hat{\beta}_3 x_i}))^2$$

Then the partial derivatives with respect to each estimated coefficient are:

$$\frac{\delta S(\hat{\boldsymbol{\beta}})}{\delta \hat{\beta}_1} = -2 \sum_{i=1}^{n} (y_i - (\hat{\beta}_1 + \hat{\beta}_2 e^{\hat{\beta}_3 x_i}))$$

$$\frac{\delta S(\hat{\boldsymbol{\beta}})}{\delta \hat{\beta}_2} = -2 \sum_{i=1}^{n} (y_i - (\hat{\beta}_1 + \hat{\beta}_2 e^{\hat{\beta}_3 x_i})) \times e^{\hat{\beta}_3 x_i}$$

$$\frac{\delta S(\hat{\boldsymbol{\beta}})}{\delta \hat{\beta}_3} = -2 \sum_{i=1}^{n} (y_i - (\hat{\beta}_1 + \hat{\beta}_2 e^{\hat{\beta}_3 x_i})) \times \hat{\beta}_2 x_i e^{\hat{\beta}_3 x_i}$$

Which, in general, is -2 (the squared term comes down, and we are subtracting the function so this becomes -2) times the error, times the first derivative of function with respect to that coefficient.

The derivatives are also nonlinear models, and there is no explicit solution if we set this equal to zero. Therefore, there is no closed form to solve for the coefficients.

Definition: A nonlinear function is one for which first derivatives for least squares estimation of the parameters are nonlinear functions of the parameters. Nonlinearity refers to the method required to find estimates of the parameters rather than the shape of the function. For example, a parabola, where y is a linear function of x and x squared is a curvilinear model, but the derivatives will be linear functions of the coefficients so this is NOT a nonlinear function.

Finding Parameter Estimates

1. Choose a set of estimated coefficients (**b**(0)=vector of starting values) to start your search (iteration=0) for the least squares solution to your nonlinear model, called the "starting parameters".

Where should you get this starting set?

i. Previous fit using the same model on a similar dataset. This could be from published papers, or based on previous work you have done.

ii. Guess a logical set of starting parameters using physical or biological rules that have meaning for your model.

Example: volume of a cone is found as area at the base, times height of the cone, divided by 3 (based on volume of a solid of revolution). For tree volume then:

$$volume = \theta_0 dbh^{\theta_1} height^{\theta_2} \boldsymbol{\varepsilon}$$

Given volume in m^3, dbh in cm, and height in m:

$$\text{area at base in m}^2 = \pi(\text{radius at base in m})^2$$

$$\cong \pi\left(\frac{dbh}{100}\Big/2\right)^2 = \frac{\pi}{40{,}000} dbh^2$$

Therefore, use:

$$\hat{\theta}_{0(0)} = \frac{\pi}{40{,}000}; \quad \hat{\theta}_{1(0)} = 2; \quad \hat{\theta}_{2(0)} = 1$$

As the starting set. Since trees are not perfectly conical, the coefficients will then deviate from these starting parameters.

iii. Fit a linear model that is very nearly the same as the nonlinear model, and use these coefficients as your starting set of parameters for the nonlinear search.

For example, for the model:

$$\mathbf{y} = \theta_0 + \theta_1 \mathbf{x_1}^{\theta_2} \mathbf{x}_2^{\theta_3} \boldsymbol{\varepsilon}$$

Fit the linear model:

$$\ln \mathbf{y} = \ln \theta_1 + \theta_2 \ln \mathbf{x_1} + \theta_3 \ln \mathbf{x}_2 + \ln \boldsymbol{\varepsilon}$$

And then use:

$$\hat{\theta}_{0(0)} = 0 \qquad \hat{\theta}_{1(0)} = \exp(\hat{\ln\theta}_1)$$

$$\hat{\theta}_{2(0)} = \hat{\theta}_2 \qquad \hat{\theta}_{3(0)} = \hat{\theta}_3$$

From this linear fit, as the starting parameters for your nonlinear model. CAUTION: Starting parameters obtained this third way are often "too tight" and you will need to round them off or alter them a little – otherwise the nonlinear search may not proceed.

iv. Use a grid search:

a. Choose several sets of possible coefficients
b. Calculate the SSE for each set of coefficients.
c. Select one of these sets that give the lowest SSE to use as starting parameters in your search.

2. Once you have selected a starting set of parameters, calculate the SSE ("evaluate the objective function for these parameters", and also calculate the Jacobian matrix for this set of coefficients and for each observation in the data.
3. Based on the SSE and the Jacobian matrix, adjust the coefficients to a new set (iteration 1).

$$\mathbf{b(1) = b(0) + correction(1)}$$

The corrections will depend on the SSE at iteration=0, and the procedure used to search. Procedures used will vary the "*stepsize*" and the "*direction*". Generally, a procedure that has too large a stepsize will miss the set of coefficients that gives the smallest SSE. However, a procedure that has very small stepsize will result in many iterations to find the set of coefficients that gives the smallest SSE.

4. Calculate the SSE for iteration=1 coefficients, and the Jacobian matrix.
5. Adjust the coefficients to get **b**(2).
6. Repeat until an optimum (smallest SSE) is reached (the solution *converges*). Can occur when: i) SSE no longer changes; ii) coefficients no longer change (corrections are zero). Most packages have defaults for convergence criteria that you can alter.

Problems:

1) Local optimum rather than a global optimum is obtained.
2) The default number of iterations is reached, before you have a global optimum.

Hints:

1. Try several sets of starting parameters to see if the same results occur, to assure a global rather than local optimum.
2. May wish to use a search algorithm with large stepsizes, followed by an algorithm with small stepsizes to reduce the number of iterations needed and to obtain a global minimum.
3. Different algorithms may work better for different datasets and models. Try different algorithms and compare solutions. Should all achieve the same set of coefficients with the same minimum SSE. (e.g., Gauss-Newton, Marquardt's, etc).

Assumptions of Nonlinear Least Squares

1. Error terms are normally distributed, and iid (as with OLS, independent, identically distributed (iid) means independent and have the same variances).
2. Assume first, second, third, etc. derivatives exist (i.e. the function is continuous).

NOTE: There are a few other conditions to obtain some of the large sample properties as given in Judge et al. 1985 (pp 198-199).

The assumptions that the error terms are normal and iid are NOT REQUIRED to obtain a nonlinear least squares solution (also true for OLS). However, these assumptions are needed in order to have good statistical properties and to calculate confidence intervals, test hypotheses, etc.

The assumptions that the error terms are normal (or some other known distribution) and iid ARE required in order to OBTAIN a maximum likelihood solution, since the probability distributions are used directly in finding the estimated coefficients.

Properties of Nonlinear Least Squares Estimators

Under the assumptions we obtain:

1) Consistent estimates of the coefficients; converges almost surely to the true parameters (strong consistency).
2) Asymptotically normally distributed coefficients (use z rather than t for confidence intervals).
3) Estimates of coefficients will be biased, but this bias decreases if the nonlinear function is closer to being linear.

Covariance Matrix for Coefficients

An asymptotic estimator of the variances of the coefficients is:

$$\mathbf{\hat{Var}(\hat{\beta})} = \hat{\sigma}_{\varepsilon}^{\,2}[\mathbf{Z(b)'Z(b)}]^{-1}$$

Where

$$\hat{\sigma}_{\varepsilon}^{\,2} = \frac{SSE}{n-p}$$

Confidence Intervals for Coefficients

For large sample sizes, asymptotic confidence intervals can be calculated as:

$$b_j \pm z_{\alpha/2} \times s_{b_j}$$

For small sample sizes, we can use Chebyshev's inequality

$$\mathrm{Prob}\left(\left| b_j - E[b_j] \right| \le \frac{\sigma_{b_j}}{\sqrt{\alpha}} \right) \ge 1-\alpha$$

to obtain a conservative estimate:

$$b_j \pm s_{b_j} \Big/ \sqrt{\alpha}$$

Confidence Intervals for Mean of y's or Prediction Intervals for New y's Given Particular Values for x's.

- Calculated using the same way as for OLS, but using the $\widehat{\mathbf{Var}}(\hat{\boldsymbol{\beta}})$ from the nonlinear fit instead of the OLS fit.

Testing Hypotheses

To test an hypothesis about a single coefficient for large sample sizes:

HO: $\beta_j = \text{constant}$ H1: $\beta_j \neq \text{constant}$

$$z = \frac{b_j - c}{s_{b_j}} \qquad z_{critical} = z_{\alpha/2}$$

Reject HO if $z > z_{critical.}$

For one or more restrictions to the model, can use a partial F-test:

- Only appropriate for large sample sizes
- Restrictions are often more complex with nonlinear models (e.g., see nonlinear restrictions on handout regarding restrictions)

HO: Restriction is justified

H1: Restriction is NOT justified

$$partial\ F = \frac{\left(SSE(restricted) - SSE(unrestricted)\right)/r}{SSE(unrestricted)/(n-p)(unrestricted)}$$

r is the number of restrictions to the nonlinear model (df error(restricted) - df error(unrestricted))

- Under H0, this follows an F distribution for a 1-α percentile with *r* and *n-p* degrees of freedom. Therefore, Fcritical=$F_{r,n-p,\ 1-\alpha}$.
- If F>Fcritical, we reject H0 (not likely true). The restrictions ARE not justified, and result in a poorer model fit.

Measures of Goodness of Fit

- Compare SSE (and Root MSE) for different models with the same data
- Calculate an I^2 (Pseudo R^2) value: Careful!! Use corrected SSY on the denominator – often not given on nonlinear least squares outputs.

Examples Using R to Fit Nonlinear Regression

To get a first impression of the data, we make a plot. As we have two species, we choose a conditional scatter plot using the function xyplot() in the package lattice. The resulting plot is shown in Figure 1.

xyplot(ht ~ dbh | spp, data=swodata, ylab="Height (m)", xlab= "dbh (cm)")

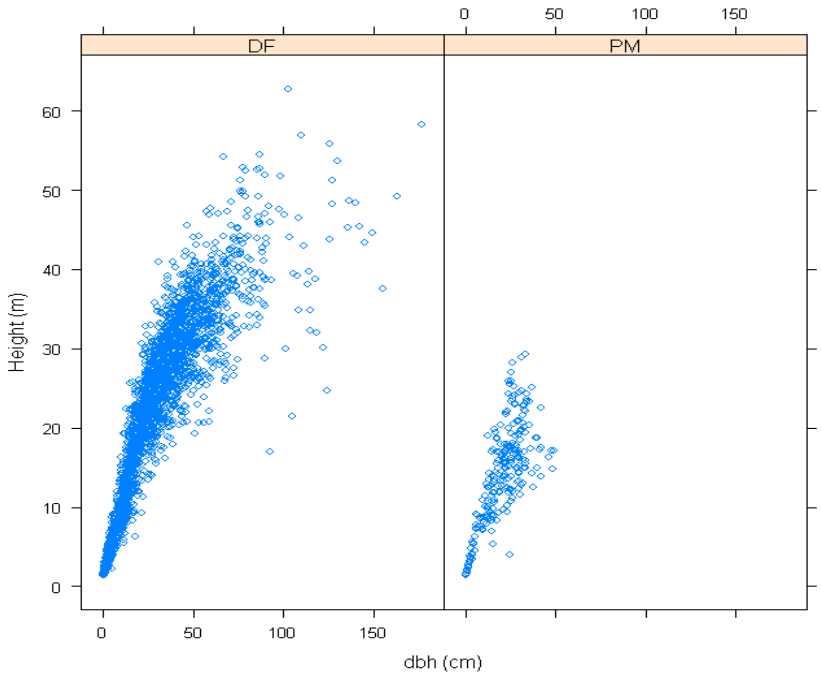


Figure 1. Height over diameter for Douglas-fir (DF) and Pacific Madrone (PM)

In R nonlinear models are fitted using nls() function (Bates and Chambers 1992). The nls() function comes with the standard installation of R. Through examples, we discuss how to obtain parameter estimates, predictions, and summary measures, and how to make plots of the fitted regression curve.

Data: swodata will be used (More information about the data is provided on Page 13.

Model fitting:

The height-diameter data are often described by the Chapman-Richard (CR) model using three parameters.

$$\hat{ht}_i = 1.3 + \beta_0\left[1 - e^{\beta_1 dbh}\right]^{\beta_2} + \varepsilon_i;\ \varepsilon_i \sim N\left(0, \sigma^2\right) \qquad [1]$$

Where: β_0 is fixed, unknown asymptotic height, β_1 is fixed, unknown steepness parameter, which is always negative, and β_2 is fixed, unknown curvature parameter that determines the rate of increase. Equation [1] is concave downward without an inflection point. β_0, β_1, and β_2 are species dependent coefficients, $\beta_0 > 0$, $\beta_1 < 0$, e is the Naperian constant (*i.e.*, 2.718).

Fitting the C-R model to data frame swodata can be done using nls().

> fit.CR<-nls(ht2~b0((1-exp(b1*dbh))^b2), data=DF, start=list(b0=64, b1=-.01, b2=0.92),*
+ trace=TRUE, nls.control(maxiter = 500))

```
77251 :  64.00    -0.01  0.92
72953 :  58.0378 -0.0117  0.9278
68740 :  49.9849 -0.0152  0.9494
59593 :  42.5647 -0.0216  1.0076
39097 :  40.6295 -0.0309  1.1315
37637 :  41.6171 -0.0323  1.1737
37637 :  41.5749 -0.0324  1.1756
37637 :  41.5706 -0.0324  1.1759
37637 :  41.5701 -0.0324  1.1759
```

The first argument, ht2~ b0*((1-exp(b1*dbh))^b2 is the model formula, where ~ is used to relate the response, ht2 (height-1.3), to the mean function, b0*((1-exp(b1*dbh))^b2, which is explicitly formulated on the right-hand side. Note that the predictor and the three parameters have to be specified explicitly, unlike the linear regression specification in lm().

The second argument (data) specifies the data frame containing the response and predictors. In order to initiate the estimation algorithm, we need to supply starting values for the parameters. The argument *start* supplies the starting values for the parameters.

The attached R (NLIN_SWO.r) code provides some example of graphical exploration and grid search methods to select starting values. R also provides self-starting functions for commonly used nonlinear models. Details of such models are listed in (Ritz and Streibig 2008) and Pinheiro and Bates (2000).

The argument trace controls whether or not the parameter values at each step in the iterative estimation procedure should be displayed. We see that the parameter values rapidly stabilize, indicating that the initial values were good guesses.

To compare different model forms and their fits to the same dataset, one can obtain a summary of measure of the model fit as follow.

To obtain the SSE, we use the deviance method.
deviance(fit.CR)
```
[1] 37637
```

To obtain the logarithm of the likelihood function, we use the logLik method
logLik(fit.CR)

```
'log Lik.' -6600 (df=4)
```

In addition to the value of the log likelihood function, the number of parameter (p+1) is shown.
We can use the *coef* method to list the parameter estimates.

```
b0        b1      b2
41.5701 -0.0324  1.1759
```

For a detailed summary of the model fit, can use the summary method.
> summary(fit.CR)

Formula: ht2 ~ b0 * ((1 - exp(b1 * dbh))^b2)

Formula: ht2 ~ b0 * ((1 - exp(b1 * dbh))^b2)

```
Parameters:
   Estimate Std. Error t value Pr(>|t|)
b0 41.57009    0.55144    75.4   <2e-16
b1 -0.03239    0.00141   -23.0   <2e-16
b2  1.17591    0.03591    32.8   <2e-16

Residual standard error: 4 on 2350 degrees of freedom

Number of iterations to convergence: 8
Achieved convergence tolerance: 2.61e-06
```

…
The fitted C-R equation is given by the following expression.

$$\widehat{ht}_i = 1.3 + 41.5701\left[1 - e^{-0.0324 \times dbh_i}\right]^{1.1759}$$

Use equation to predict height to new observations (dbhs)

dVal <- c(10,50,100,150)
predict(fit.CR, data.frame(dbh=dVal))

```
[1]  9.17 32.07 39.66 41.19
```

For a given dbh, the above equation can be used for predicting tree height values. The predicted values play a major role in model diagnostics such as residual plots. The fitted values from the model fit.CR fit are obtained using a data frame of hypothetical (dbh) values and the predict function.

Model diagnostics:
Assumptions:

A. Checking model form (mean structure) via plot of the fitted regression curve
Create a plot of the original data superimposed with the C-R fitted curve. The plot of the fitted regression curve is used to examine the mean structure.

```
plot(ht2~dbh, data=DF, ylim=c(0,90), ylab="Height(m)", xlim=c(0,180), xlab="dbh (cm)")
dVal <-with(DF, seq(min(dbh), max(dbh), length.out=100))
lines(dVal, predict(fit.CR, newdata=data.frame(dbh=dVal)),col="red")
abline(h=1.3+coef(fit.CR)[1], lty=2, col="green")
```

B. Test for normality using the using the Shapiro-Wilk test:
```
> shapiro.test(resid)
```

C. Check assumptions of equal variance and normality of residuals (observed value - predicted value)
```
plot(fitted(fit.CR), residuals(fit.CR), xlab="Fitted Values", ylab="Residuals")
abline(a=0, b=0)
```

Please note heteroscedsacity - we will fix this problem towards the end of this or in the nonlinear mixed effect regression (NLME) session.

C. Examine independence using lag plot, which is a plot of each residual versus the lag-one residual
```
plot(residuals(fit.CR), c(residuals(fit.CR)[-1], NA), xlab="Residuals", ylab="Lagged Residuals")
```

Model 2: Ratkowsky's model

[2] Ratkowsky (1990) $ht_i = 1.3 + e^{\left[\beta_0 + \frac{\beta_1}{dbh_i + \beta_2}\right]} + \varepsilon_{2i}\,;\ \varepsilon_{2i} \sim N\left(0, \sigma_2{}^2\right)$

```
fit.rat<-nls(ht2~exp(b0+(b1/(dbh+b2))),data=DF,
        start=list(b0=4.3, b1=-49, b2=13), trace=TRUE, nls.control(maxiter = 500))
```

Print out summary of fit
```
> summary(fit.rat)
```

Compare model 1 and 2 – see the difference in asymptotic height.
We can use AIC to compare models; the smaller AIC the better. If more than 10 select one model over the other (Burnham and Anderson 2002).

```
> AIC(fit.CR)
[1] 13209

> AIC(fit.rat)
[1] 13175
```

> anova(fit.CR,fit.rat)
Analysis of Variance Table

*Model 1: ht2 ~ b0 * ((1 - exp(b1 * dbh))^b2)*
Model 2: ht2 ~ exp(b0 + (b1/(dbh + b2)))

	Res.Df	*Res.Sum Sq*	*Df*	*Sum Sq*	*F value*	*Pr(>F)*
1	2350	37637				
2	2350	37109	0	0		

Examine assumptions for the Ratkowsky's equation.

Additional stand and positional variables:
Would other stand variables improve the fit of the model? Will a nonlinear model with stand and positional variables improve the fit of the model?

```
fit.CR2<-nls(ht2~(a0+a1*ccfl+a2*ba)*((1-exp(b1*dbh))^b2),data=DF,
                        start=list(a0=47, a1=0.065, a2=0.13, b1=-.017, b2=0.97),
    control=nls.control(maxiter=9000, tol=0.01, minFactor=1/1000))
```

Model comparison:
> anova(fit.CR,fit.rat,fit.CR2)
Analysis of Variance Table

Model 1: ht2 ~ b0 * ((1 - exp(b1 * dbh))^b2)
Model 2: ht2 ~ exp(b0 + (b1/(dbh + b2)))
Model 3: ht2 ~ (a0 + a1 * ccfl + a2 * ba) * ((1 - exp(b1 * dbh))^b2)

	Res.Df	Res.Sum Sq	Df	Sum Sq	F value	Pr(>F)
1	2350	37637				
2	2350	37109	0	0		
3	2348	34095	2	3014	104	<2e-16

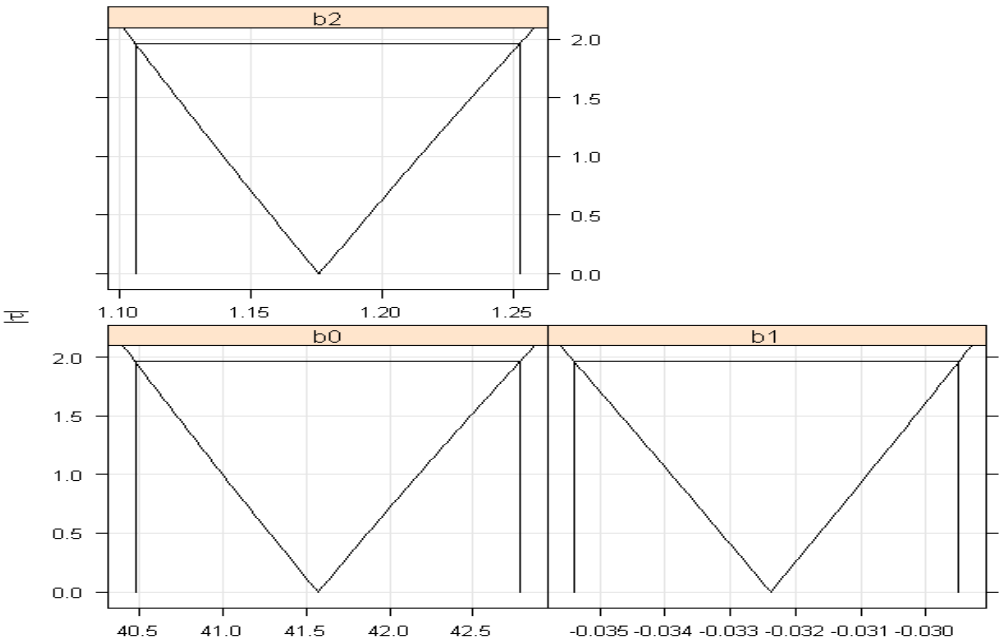
Profile likelihood: indicates if the linear approximation is perfect for each parameter or not.

For each parameter estimate, the profile t function is defined as (Venables and Ripley 2002 p. ??)

$$\tau(\beta_j) = sign(\beta_j - \hat{\beta}_j)\frac{\sqrt{RSS(\beta_j) - RSS(\hat{\beta}_j)}}{s}$$

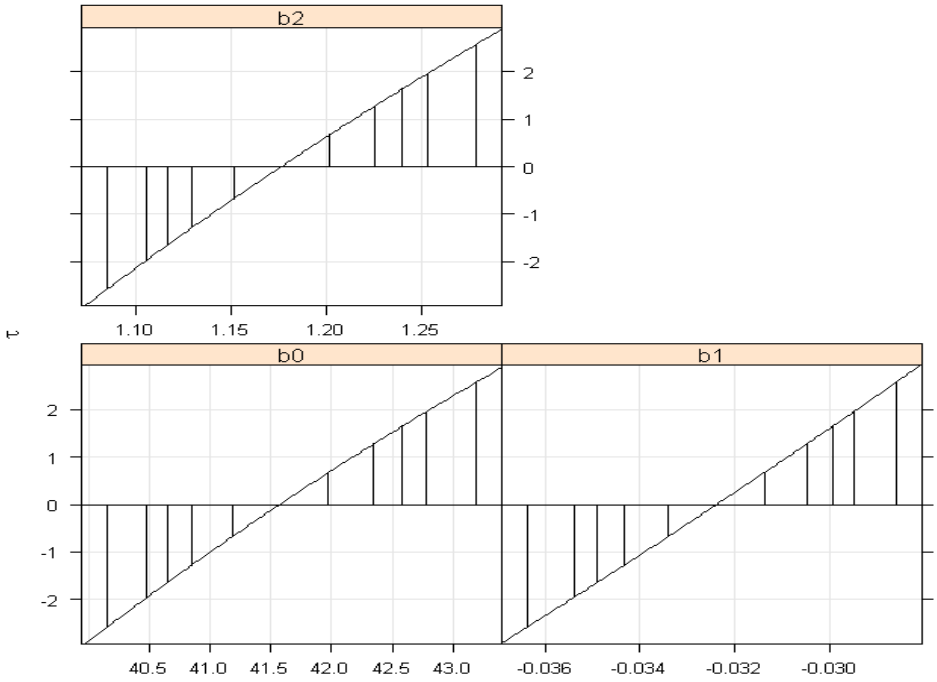
where s=residual standard error, RSS=,

Plot of the profile t functions for the three parameters in the C-R function
> plot(profile(fit. CR), conf=0.95)



Obtain the original tau values instead of absolute values and examine if curvature exist or the linear approximation is perfect. The graph indicates that the linear approximation is not perfect in b0 component. The linear approximation appears acceptable for b1 and b2, as there was not curvature for these parameter estimates.

opar1 <- par(mfrow=c(3,3), mar=c(4,4,2,1), las=1)
plot(profile(fit.CR), absVal=FALSE)
par(opar1)



Confidence intervals for the tau; extract the profiled versions

> confint(fit.CR)

2.5% 97.5%

b0 40.4746 42.7817

b1 -0.0354 -0.0295

b2 1.1057 1.2526

Remedies for model violations: [More might be needed here]

In R, the gnls() function in the package nlme allows generalized least squares estimation of the power-of-the mean and other variance models. Default is weight=varPower(form=fitted(.)) is the variance is a power of the fitted values. Different variance models are provided by Pinheiro and Bates (2000)

A. Weighted least squares – weighted by a selected variable, etc.

B. Variance Modeling - explicitly models the variance; assumes errors are additive and normally distributed. For example, weights=varPower() specifies the variance model.

C. Transformations - uses Box-Cox regression

Nonlinear Regression Models

Exercise 1: Fitting Nonlinear Height-Diameter Models

Background:
Modeling stand development over time relies on accurate estimates of tree height (ht) and diameter (dbh). Accurate ht measures are required to describe vertical stand structure, and estimate stand volume and site quality. Because measuring ht is costly, height-diameter functions (H-D) are commonly used to predict tree height.

Objective: To develop nonlinear H-D equations for south west Oregon (SWO) and conduct model diagnostics.

Data: Data were collected in SWO. Elevation of the sample plots ranged from 250 to 1600 meters. Stand structures found in the sample area ranged from even-aged stands of one or two stories to uneven-aged stands. Each stand was sampled with 4 to 25 sample points spaced 45.73-meters apart. ht and dbh were measured on all undamaged trees. The data were analyzed in Temesgen et al. (2007). For this assignment, a subset of the data was extracted to fit, compare, and select height-diameter functions.

You will analyze data in a file called swodata.csv. Variables (and their definition) in order of appearance on the record are described as follows:

1. trno Tree number identifier	8. tph Number of trees per hectare
2. dbh Tree diameter at breast height (cm)	9. spp Tree species identifier
3. cl Tree crown length (m)	10. ccfl Crown competition factor in larger trees
4. dead Status identifier (1=Live, 0=dead)	11. standid Stand number identifier
5. ht Tree height (m)	12. plotid Plot number identifier
6. bal Basal area in larger trees (m^2/ha)	13. nsnag Number of snags/ha:
7. ba Basal area/ha (m^2/ha)	

Two commonly used height-diameter equations for estimating tree height in Oregon's forests are:

[1] Chapman-Richards function (Richards 1959): $ht_i = 1.3 + \beta_0 \left[1 - e^{\beta_1 dbh_i}\right]^{\beta_2} + \varepsilon_{1i}$; $\varepsilon_{1i} \sim N\left(0, \sigma_1^2\right)$

Where: β_0 is fixed, unknown asymptote; β_1 is fixed, unknown steepness parameter; and β_2 is fixed, unknown curvature parameter that determines the rate of increase; β_0, β_1, and β_2 are species dependent coefficients, $\beta_0 > 0$, $\beta_1 < 0$, e is the Naperian constant (*i.e.*, 2.718).

[2] Ratkowsky (1990) $ht_i = 1.3 + e^{\left[\beta_0 + \frac{\beta_1}{dbh_i + \beta_2}\right]} + \varepsilon_{2i}$; $\varepsilon_{2i} \sim N\left(0, \sigma_2^2\right)$

Equation (1) is used by Garman et al. (1995), while Equation (2) is used by Flewelling and de Jong (1994).

Read in the data using the R script found in 'NLIN_SWO.r'. Check to make sure the data have 50 stands, 2699 tree records.

In this exercise, three runs will be obtained and results will be compared.

1. Use the nls() function and fit nonlinear regression Equations 1 and 2 with ht as the response variable and dbh as predictor variable to the data set *swodata (use only live DF)*, using the following starting values parameter for Equations 1 (β_0=64, β_1=-.01, β_2=0.92); and Equations 1 (β_0=4.3, β_1=-49, β_2=13).

2. The starting values in (1) are obtained from literature. Initial approximations for each parameter were obtained from linear transformation of the equations, where possible. Vary the starting value of each parameter in order to find a global minimum. Does changing the starting values change SSE, AIC, etc?

3. Plot the fitted regression curve with the data.

4. Check whether the model meets the assumptions of nonlinear regression
   a) Normality
   b) Check for non constant variance. Are there equal variances over the full range of predicted values at 0.05 α-level? Is weighting necessary (e.g., by 1/DBH) to account for constant variance?
   c) Use graphical check for independence (i.e., correlated errors). What do you conclude?

5. Each equation's parameters should be evaluated to determine if they were significantly different from zero using the asymptotic t-test. Parameters not significantly different from 0 at p = 0.05 should be set to zero and the remaining parameter re-estimated.

6. Compare the fit statistics (e.g., AIC and SSE) for Equations (1) and (2) and select one equation for future height prediction in SWO.

7. Height-diameter functions are expected to vary by tree species and model form. The predictive ability of different model forms and associated estimated parameters also varies from species to species. Did the parameter estimates and model performance vary for DF and BM?

8. Would the inclusion of competition measures such as CCFL and BA improve the performance of the Chapman Richards' function equation?

References:

Bates, D.M. and J.M. Chambers. 1988. Nonlinear regression analysis and its applications. John Wiley and Sons, New York.

Bates, D.M. and J.M. Chambers 1992. Statistical models in S, chapter 10 (Nonlinear models). Chapman and Hall, Boca Raton, FL.

Burnham, K.P. and D.R. Anderson. 2002. Model selection and multi-model inference. Springer, New York, 2nd edition.

Flewelling, J.W. and R.D. de Jong. 1994. Considerations in simultaneous curve fitting for repeated height-diameter measurements. Can. J. For. Res. 24: 1408-1414.

Garman, S.L., S.A Acker, J.L. Ohmann, and T.A. Spies. 1995. Asymptotic height-diameter equations for twenty-four tree species in Western Oregon. Res. Cont. 10, For. Res. Lab. Oregon State Univ., Corvallis, OR 22 p.

Judge et al. 1985. The theory and practice of econometrics, 2nd ed., New York.

Pinheiro, J.C. and D.M. Bates 2000. Mixed-effects models in S and S-plus. Springer-Verlag, New York.

Ratkowsky, D.A. 1990. Handbook of nonlinear regression. Marcel and Dekker, Inc. New York.

Richards, F.J. 1959. A flexible growth function for empirical use. Jour. Expt. Botany 10: 290-300.

Ritz, C. and Streibig, J.C. 2008. Nonlinear regression with R. Springer, New York, 148p.

Temesgen, H., D.W. Hann, and V.J. Monleon. 2007. Regional height-diameter equations for major tree species of southwest Oregon. West. J. Appl. For. 22(3): 213-219.