

fishR Vignette - Von Bertalanffy Growth Models

Dr. Derek Ogle, Northland College

December 16, 2013

Growth of individual organisms within a population is generally characterized by an expression which is representative of individual growth of an “average” animal in the population. Individual growth is treated as an increase in either length or weight with increasing age. Several functions or models have been used to model the mean length or weight of fishes. One model, the von Bertalanffy growth model (VBGM), is the most popular and is the focus of this vignette.

The functions required to perform growth analyses in R are contained in the `FSA` and `FSAdata` packages created by the author and the `nlstools` package. These packages are loaded into R with

```
> library(FSA)
> library(FSAdata)
> library(nlstools)
```

All analyses in this document use the lengths and ages of Atlantic Croaker (*Micropogonias undulatus*; Figure 1) as determined by researchers at Old Dominion University in conjunction with the Virginia Marine Resources Commission (VMRC). The data are stored in the `Croaker2` data frame found in the `FSAdata` package which is loaded along with `FSA`. The first part of this vignette will focus only on the male Atlantic Croaker data as isolated and stored in `crm` with

```
> data(Croaker2)
> crm <- Subset(Croaker2,sex=="M")
```

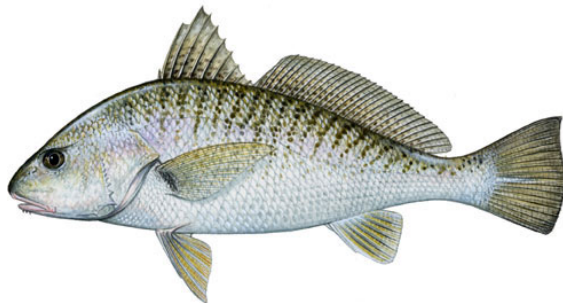


Figure 1. Atlantic croaker drawing (from [Maryland DNR website](#)).

This vignette begins with a brief introduction to the type of data that can be used for modeling growth with the VBGM (Section 1). The general concept of fitting non-linear models is described in Appendix A with these methods first being extended and then applied to the typical VBGM in Section 2. Alternative parameterizations of the VBGM are described, with an example fitting, in Section 3. Methods for comparing VBGM fits between groups are described in Section 5. Finally, Appendix B gives derivations for the different parameterizations of the VBGMs.

1 Size-at-Age Data

The raw data for size-at-age modeling consists of the measurement of size, either length or weight, and the assignment of age to individual fish at the time the fish was captured. This is called *length-at-capture*, *weight-at-capture*, and *age-at-capture* data. An example¹ of this type of data for length and age is shown in Table 1-left and in Figure 2.

¹The remainder of this vignette will focus on length data only

Table 1. Length-at-age for a portion of a sample of male Atlantic Croakers (left) and average length-at-age (right) for all male Atlantic Croakers in 1999.

<i>Indiv</i>	<i>Age</i>	<i>Length</i>	<i>AvgAge</i>	<i>AvgLength</i>
1	1	248	1	248.0
2	2	210	2	248.4
3	2	225	3	281.8
4	2	236	4	298.3
5	2	240	5	328.2
6	2	245	6	345.9
7	2	255	7	332.5
8	2	258	8	344.3
9	2	263	9	370.8
10	2	270	10	327.0
11	2	292		
⋮	⋮	⋮		

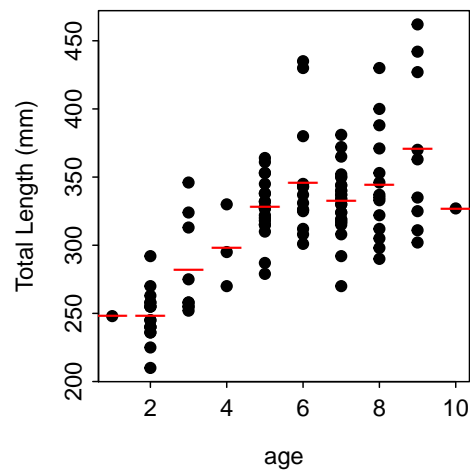


Figure 2. Length-at-age of male Atlantic Croaker. Red lines are the average length-at-age.

Historical methods for estimating the parameters of growth models used the *average* length or weight at each observed age for the sample (Table 1-right) rather than the original data recorded for each individual. The historical methods should no longer be used because high-speed computers can easily perform the calculations or iterations necessary for fitting the growth models to individual at-capture data. In addition, the historical methods lost important information about individual variability. The historical method is carefully discussed in Ricker (1975) and a brief example for fitting the VBGM to the mean data is given in Section 4.2.

It is also common to use “back-calculation” techniques to reconstruct data for growth analyses. This type of data is called *retrospective size-at-age data* (Jones 2000). Growth data generated in this way is generally inappropriate for the simple modeling techniques developed in this vignette because the back-calculated measurements are serially correlated (i.e., the techniques described here require independent observations). The problems created by serially-correlated data and techniques for handling serially-correlated data are described in Jones (2000).

2 “Typical” von Bertalanffy Growth Model

2.1 The Model and Parameters

There are many versions or parameterizations of the VBGM (see Section 3). The most common version is due to Beverton (1954) and Beverton and Holt (1957), who modified the original version introduced by von Bertalanffy (Cailliet *et al.* 2006). This model will be termed the “typical” version of the VBGM throughout this vignette. The typical VBGM is represented by²

$$E[L|t] = L_{\infty} \left(1 - e^{-K(t-t_0)}\right) \quad (1)$$

where

- $E[L|t]$ is the expected or average length at time (or age) t ,
- L_{∞} is the asymptotic average length,
- K is the so-called Brody growth rate coefficient (units are yr^{-1}), and
- t_0 is a modeling artifact that is said to represent the time or age when the average length was zero.

The parameters of (1) have strict meanings that have been misunderstood in past work. The following items address some of these misunderstandings.

- L_{∞} is *not* the maximum length of the animal. Rather L_{∞} is the asymptote for the model of *average* length-at-age (Francis 1988). As with any average, some individuals will be larger than average; thus, some animals will be larger than L_{∞} . This is illustrated in Figure 3.
- Furthermore, Francis (1988) argues that L_{∞} only has meaning in fish populations where “mortality is low enough to allow fish to reach an age at which the mean length (virtually) ceases to increase.”
- K is *not* a growth rate (Ricker 1975, p. 221). The units of K can be seen by algebraically solving (1) for K ,

$$\begin{aligned} 1 - \frac{E[L|t]}{L_{\infty}} &= -e^{-K(t-t_0)} \\ \log\left(\frac{1 - E[L|t]}{L_{\infty}}\right) &= -K(t - t_0) \\ K &= \frac{\log\left(\frac{L_{\infty} - E[L|t]}{L_{\infty}}\right)}{t - t_0} \end{aligned} \quad (2)$$

²A weight specific version essentially raises the length-specific version to the power of b , the allometric relation term from the regression of weight on length (i.e., $W = aL^b$) – i.e., $E[W|t] = (W_{\infty} (1 - e^{-K(t-t_0)}))^b$.

to notice that the units in the numerator disappear. Thus, the units of K are in $time^{-1}$ rather than a change in length for a unit time as required by a growth rate.

- [Schnute and Fournier \(1980\)](#) offer two related interpretations of the meaning of K . First, K “measures the exponential rate of approach to the asymptotic size.” Second, the value e^{-K} is “the fixed fraction by which the annual growth increment is multiplied each year.” This last definition is seen by the relation

$$E[L|t = i + 2] - E[L|t = i + 1] = e^{-K}(E[L|t = i + 1] - E[L|t = i]) \quad (3)$$

where, because $K > 0$, the quantity e^{-K} is a decimal. Thus, each succeeding growth increment is a constant fraction of the current growth increment indicating that growth slows as the fish gets older.

- t_0 , despite its definition above, is a modeling artifact and not a biological parameter ([Schnute and Fournier 1980](#)). The t_0 parameter is included in (1) to adjust or “correct” the model for the initial size of the animal as most of the models do not pass through the origin. This is illustrated in Figure 3. [Beverton \(1954\)](#)[p. 43] said “It must be remembered that the constant t_0 is largely artificial, insofar as it defines the age at which the organism would be of zero length if it grew throughout life with the same pattern of growth as in the post-larval stage.” In addition, [Beverton and Holt \(1957\)](#)[p. 34] stated that “in practice, the constant t_0 must be regarded as quite artificial.”
- The estimation and interpretation of t_0 is further complicated by the fact that the value of t_0 is generally an extrapolation ([Francis 1988](#)) – e.g., Figure 3.
- L_∞ , K , and t_0 are highly related (or correlated). This can be seen in the solution for K above where K depends on the values of L_∞ and t_0 . For example, with all else held constant, K will be larger for smaller values of L_∞ . This dependency among parameters causes problems in model fitting and interpretation ([Ratkowsky 1986](#)). Some of the other VBGM parameterizations (Section 3) have parameters that are much less correlated.

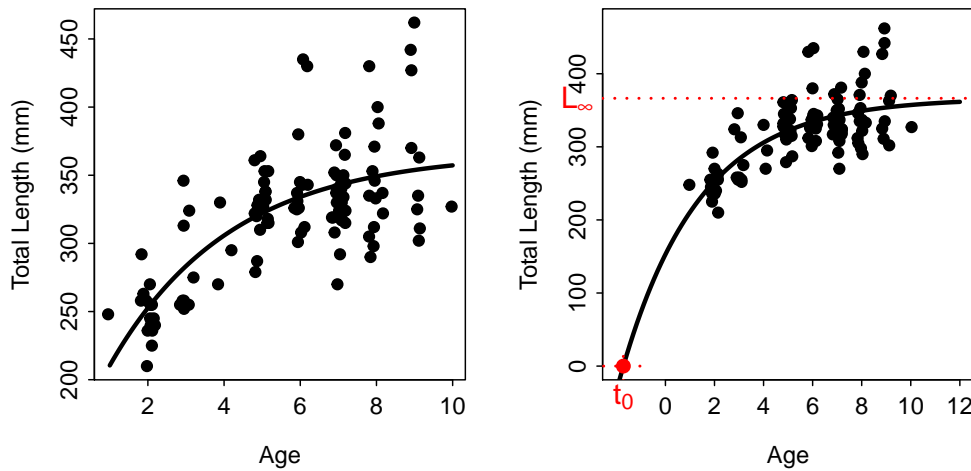


Figure 3. Example of the typical parameterization of the VBGM fit to size-at-age Atlantic Croaker data. The plot on the left is the fitted model expressed over the observed ages and lengths of the data. The plot on the right is the same fitted model but expressed over a range of ages and lengths that allow illustrating the meaning of L_∞ and t_0 .

2.2 Fitting von Bertalanffy Models in R

2.2.1 Starting Values

The VBGM, as well as most other growth models, is a non-linear model that requires non-linear statistical methods to estimate parameter values. Fitting non-linear models³ is a two-step process of first finding starting values for the model parameters and, second, calling `nls()` with the non-linear model and these starting values. Two general methods for finding starting values for the typical VBGM are described below.

Ford (1933) and Walford (1946) introduced the Ford-Walford plot method for estimating L_∞ and K from mean length-at-age data. They showed that a plot of L_{t+1} versus L_t for length data that follows a typical VBGM is linear with a slope equal to e^{-K} . In addition, the point where the linear relationship between L_{t+1} and L_t intercepts the 1:1 line (45°) is an estimate of L_∞ (Figure 4).

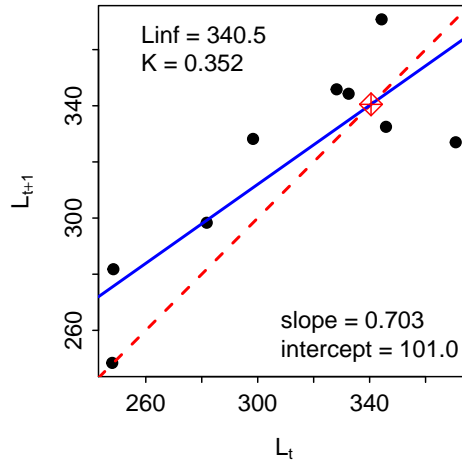


Figure 4. Example of a Ford-Walford plot for the male Atlantic Croaker data. The dashed red line is the 1:1 relationship and the solid blue line is the regression line fit to the observed data. The intersection of these two lines is marked and represents an estimate of L_∞ .

Algebraically manipulating the two observations about the Ford-Walford plot shows that the following equations provide reasonable starting values for L_∞ and K (the “tilde” above each parameter will be used to identify these results as starting values for the non-linear regression),

$$\tilde{K} = -\log(\text{slope})$$

$$\tilde{L}_\infty = \frac{\text{intercept}}{1 - \text{slope}}$$

where *intercept* and *slope* are obtained from fitting the linear regression of L_{t+1} on L_t as in the Ford-Walford plot.

With a starting value for L_∞ and observed values for one age and mean length pair (i.e., (t, \bar{L}_t)), the typical VBGM (i.e., (1)) can be re-arranged to solve for t_0 as follows,

$$\tilde{t}_0 = t * \log\left(\frac{\tilde{L}_\infty - \bar{L}_t}{\tilde{L}_\infty}\right)$$

An alternative method for obtaining \tilde{t}_0 is to fit a second-degree polynomial to the mean length-at-age data,

³Fitting non-linear models is discussed in detail in Appendix A.

find the roots for this polynomial (i.e., where the polynomial function crosses the x- or age axis), and choose the root that is closest to zero⁴.

The methods described above for generating reasonable starting values are implemented in `vbStarts()`. The first argument to `vbStarts()` is a formula of the form `len~age` where `len` and `age` generically represent the variables containing the lengths and ages for individual fish. The `data=` argument must also be set to data frame where the ages and lengths are found. The `vbStarts()` function defaults to using the polynomial regression method to produce a starting value for t_0 . Starting values for t_0 based on the re-arrangement of the typical VBGM can be obtained by using the `meth0="yngAge"` argument. Finally, as an option, a graphic of the VBGM using the default starting values superimposed on the observed data can be constructed by including `plot=TRUE`. The results of `vbStarts()` must be saved to an object for use in subsequent calls to `nls()`. For example, the starting values for the typical VBGM applied to the male Atlantic Croaker data are obtained with

```
> svTypical <- vbStarts(tl~age,data=crm)
> unlist(svTypical)      # unlist used only to save space when viewing the results
      Linf      K      t0
340.5069  0.3517 -1.6732
```

The selection of starting values for the VBGM can also be obtained with `growthModelSim()` from the `FSATeach` package⁵. This function produces a plot of length-at-age with the VBGM superimposed. The superimposed VBGM can be controlled with slider bars tied to the parameters of the VBGM. Thus, the slider bars can be adjusted until a VBGM is produced that “roughly” fits the observed lengths-at-age. The parameters of the VBGM when this rough fit is found are then used as the starting values for the non-linear methods. For this purpose, `growthModelSim()` has three arguments very similar, but in a different order, to what was described for `vbStart()`. The first argument is a string indicating the VBGM to use, the second argument is the model formula, and the third, or `data=`, argument is the data frame in which the observed ages and lengths can be found. The typical VBGM is used with a first argument of `"vbTypical"`. As an example, the starting values for the typical parameterization of the VBGM to be fit to the male Atlantic Croaker length data can be obtained by manipulating the slider bars on the graphic (Figure 5) produced with

```
> growthModelSim("vbTypical",tl~age,data=crm)
```

Following the identification of an approximate fit to the length-at-age data, an object must be created that contains a list of the identified starting values. The starting values are entered into a list, for example, with

```
> svTypical <- list(Linf=376,K=0.3,t0=-1.8)
```

2.2.2 General Fitting of the Models with `nls()`

Parameters for the non-linear VBGM can be estimated with `nls()`. The `nls()` function requires the growth model expression as the first argument, with the appropriate variable names substituted for the generic length and age variables, the data frame from which to draw the variables in the `data=` argument, and the list containing starting parameter values in the `start=` argument. As always, the model fit should be assigned to an object for further analysis. Generally, it is somewhat easier to enter the model function and starting values into objects prior to the `nls()` call. For example, the typical VBGM model is fit to the male Atlantic Croaker data with⁶

⁴This process will be discussed in more detail in Section 3.3.

⁵Note that the `FSATeach` package must be installed as described [here](#) and the loaded with `library(FSATeach)`.

⁶Note that it is good practice to be able to enter the models directly into R. However, `vbFuns()` is a convenience function that can be used to easily create the right-hand-side of VBGM parameterizations used in this vignette. For example, the same typical model could be fit with `fitTypical <- nls(tl ~ vbFuns("typical"),data=crm,start=svTypical)`.

```

> # note that svTypical was created in the previous section
> vbTypical <- tl~Linf*(1-exp(-K*(age-t0)))
> fitTypical <- nls(vbTypical,data=crm,start=svTypical)

```

A quick visual (Figure 6) of the fitted model is constructed with⁷

```

> fitPlot(fitTypical,xlab="Age",ylab="Total Length (mm)",main="")

```

Summary results of the model fit (e.g., coefficient values, measures of variability, parameter correlations) can be obtained with `overview()` from the `nlstools` package as follows,

```

> overview(fitTypical)

-----
Formula: tl ~ Linf * (1 - exp(-K * (age - t0)))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
Linf  366.414    16.754   21.87  <2e-16
K       0.315     0.108    2.92  0.0042
t0     -1.714     1.049   -1.63  0.1049

Residual standard error: 33.4 on 111 degrees of freedom

Number of iterations to convergence: 3
Achieved convergence tolerance: 2.07e-06

-----
Residual sum of squares: 124000

-----
Asymptotic confidence interval:
      2.5%    97.5%
Linf 333.2146 399.6141
K     0.1015  0.5281
t0   -3.7921  0.3637

-----
Correlation matrix:
      Linf      K      t0
Linf  1.0000 -0.9506 -0.8706
K     -0.9506  1.0000  0.9718
t0    -0.8706  0.9718  1.0000

```

These results show that point estimates of the parameters are: $\widehat{L}_\infty = 366.4$, $\widehat{K} = 0.31$, and $\widehat{t}_0 = -1.71$. In addition, as expected, the parameter estimates are very highly correlated.

The results from `overview()` show confidence intervals for the model parameters based on normal distribution theory; however, as discussed in Appendix A, confidence intervals for parameters in non-linear models are best constructed through bootstrap methods. The bootstrap samples are taken by sending the saved `nls()` object to `nlsBoot()` with the `niter=` argument controlling the number of bootstrap samples to take. Confidence intervals, with associated plots of the distribution of parameter estimates from the bootstrap

⁷Note that examples later in this vignette will use `curve()` rather than `fitPlot()` to obtain a visual of the model fit. The `curve()` method is more general and will allow the building of comparative graphs, whereas `fitPlot()` is a specific function and is, thus, limited in what it can display.

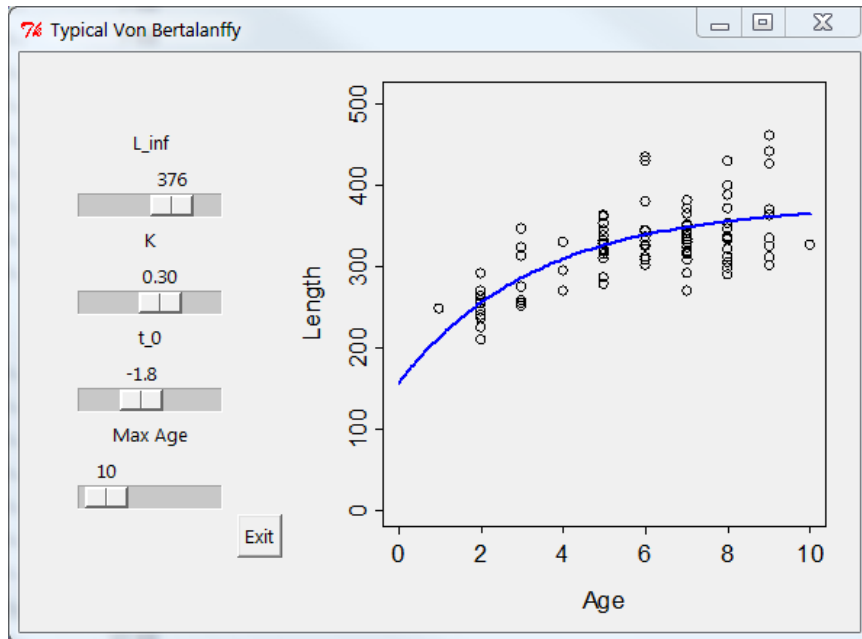


Figure 5. Example of using `growthModelSim()` to find an approximate fit of the typical VBGM parameterization to the male Atlantic Croaker data.

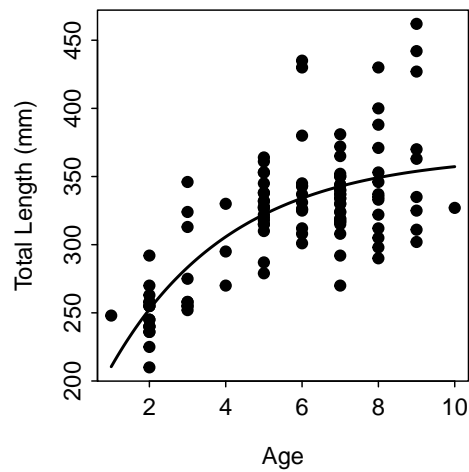


Figure 6. Fitted typical VBGM for the male Atlantic Croaker data.

samples, are constructed by sending the object saved from `nlsBoot()` to `confint()` with the `plot=TRUE` argument. Thus, confidence intervals for the parameters in the typical VBGM fit to the male Atlantic Croaker data are obtained with⁸

```
> bootTypical <- nlsBoot(fitTypical, niter=200) # niter should be nearer 1000
Warning: The fit did not converge 1 times during bootstrapping
> confint(bootTypical, plot=TRUE)
      95% LCI  95% UCI
Linf 344.7057 429.6830
K      0.1346  0.5585
t0    -4.7347 -0.2869
```

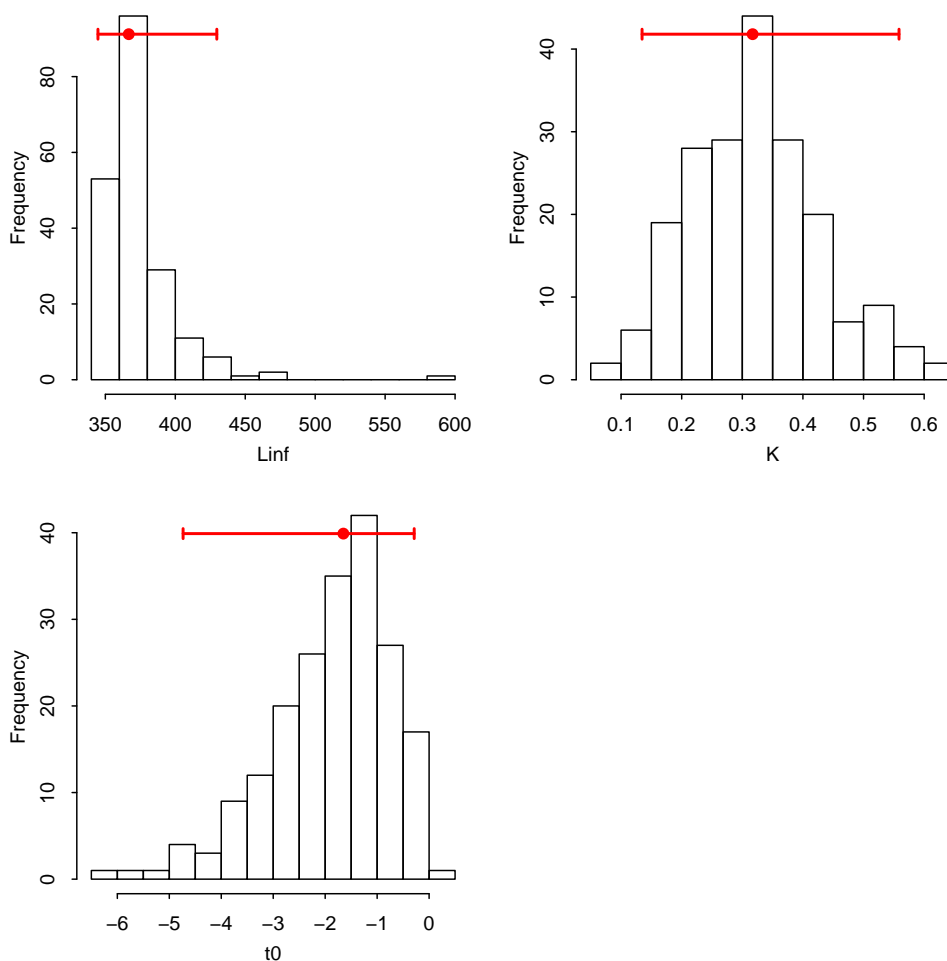


Figure 7. Histogram of the bootstrap results for the typical VBGM for the male Atlantic Croaker data. The red horizontal lines represent the 95% bootstrap confidence intervals.

Thus, for example, one is 95% confident that the Brody growth coefficient (K) is between 0.13 and 0.56. Histograms of the bootstrapped results shows that the distribution of K is approximately normally distributed whereas the distributions of L_∞ and t_0 are very much not normally distributed (Figure 7). Thus, the difference between the asymptotic and bootstrapped confidence intervals are much greater for L_∞ and t_0 than for K .

⁸Bootstrap methods require patience as the algorithm continuously resamples the original data. Furthermore, the bootstrap methods rely on randomization; thus, the results shown below would likely be different upon a different “run” of the functions.

The p-value for testing the hypothesis that $H_0 : K = 0.5$ versus $H_A : K < 0.5$ is obtained with

```
> htest(bootTypical, "K", 0.5, "less")
Ho Value p value
   0.5 0.07538
```

Thus, with a p-value ($p = 0.0754$) slightly larger than a typical α of 0.05, there is not enough evidence to conclude that K is less than 0.5.

Predicted lengths at a given age are obtained with the `predict()` function. However, as discussed in Appendix A, corresponding confidence intervals must be constructed from the bootstrapped results. As an example, the predicted mean length and corresponding confidence interval for the mean length at age-8 can be constructed with

```
> new <- data.frame(age=8)
> predict(fitTypical, new)
[1] 349.2
> ests <- bootTypical$coefboot
> pv <- ests[, "Linf"]*(1-exp(-ests[, "K"]*(8-ests[, "t0"])))
> quantile(pv, c(0.025, 0.975))
 2.5% 97.5%
340.6 358.2
```

Thus, the mean length of all age-8 male Atlantic Croakers is between 341 and 358 mm.

One may be interested in creating a fitted line plot with confidence bounds for the mean length-at-age from the VBGGM. One way to create this plot is to first predict the length-at-age for each age in a range of ages using the parameters from each of the bootstrap samples. One can then plot the 2.5% and 97.5% quantile length-at-age for each age. This process is illustrated with the following code which resulted in Figure ??.

```
> ages2plot <- 0:15
> fitPlot(fitTypical, xlab="Age", ylab="Total Length (mm)", xlim=range(ages2plot), main="")
> LCI <- UCI <- numeric(length(ages2plot))
> for (i in 1:length(ages2plot)) {
  pv <- ests[, "Linf"]*(1-exp(-ests[, "K"]*(ages2plot[i]-ests[, "t0"])))
  LCI[i] <- quantile(pv, 0.025)
  UCI[i] <- quantile(pv, 0.975)
}
> lines(UCI~ages2plot, type="l", col="blue", lwd=2, lty=2)
> lines(LCI~ages2plot, type="l", col="blue", lwd=2, lty=2)
```

One may also want to add approximate prediction bounds to the fitted line plot. One way to do this is to add and subtract the residual standard error from each bootstrap model (these values are found in the `rse` portion of the results saved from the bootstrap sampling – i.e., `bootTypical$rse` in this example.) to the predictions constructed from each model as described for the confidence bounds above⁹. The 2.5% quantile for when the residual standard error was subtracted is an approximate 95% lower prediction bound and the 97.5% quantile for when the residual standard error was added is an approximately 95% upper prediction bound. The code for a plot with both the confidence and prediction bounds is constructed with the code below and is shown in Figure ??.

```
> ages2plot <- 0:15
> fitPlot(fitTypical, xlab="Age", ylab="Total Length (mm)", xlim=range(ages2plot), main="")
```

⁹This assumes that the residuals are homoscedastic or additive. See Section 4.1.

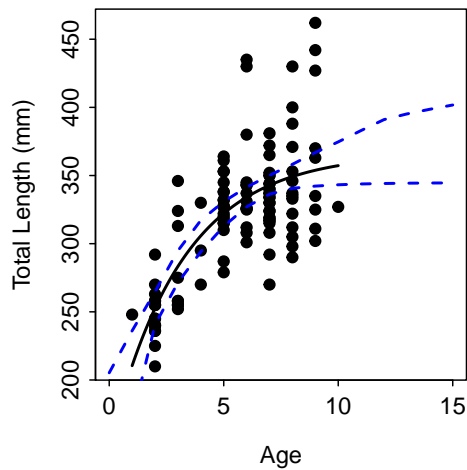


Figure 8. Fitted line plot for the fit of the typical VBGM with approximate 95% bootstrap confidence bounds shown as blue dashed lines.

```

> LCI <- UCI <- LPI <- UPI <- numeric(length(ages2plot))
> for (i in 1:length(ages2plot)) {
  pv <- ests[,"Linf"]*(1-exp(-ests[,"K"]*(ages2plot[i]-ests[,"t0"])))
  LCI[i] <- quantile(pv,0.025)
  UCI[i] <- quantile(pv,0.975)
  LPI[i] <- quantile(pv-bootTypical$rse,0.025)
  UPI[i] <- quantile(pv+bootTypical$rse,0.975)
}
> lines(UCI~ages2plot,type="l",col="blue",lwd=2,lty=2)
> lines(LCI~ages2plot,type="l",col="blue",lwd=2,lty=2)
> lines(UPI~ages2plot,type="l",col="red",lwd=2,lty=2)
> lines(LPI~ages2plot,type="l",col="red",lwd=2,lty=2)

```

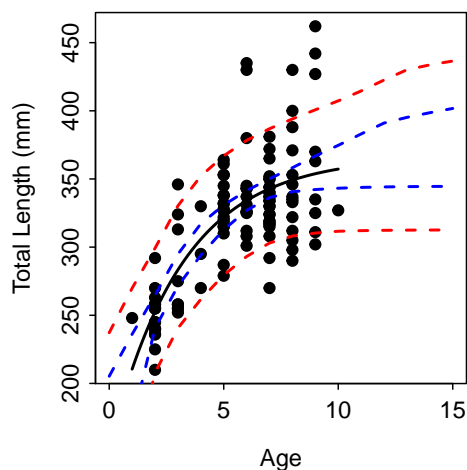


Figure 9. Fitted line plot for the fit of the typical VBGM with approximate 95% bootstrap confidence bounds shown as blue dashed lines and 95% bootstrap prediction bounds shown as red dashed lines.

Finally, submitting the `nlsBoot()` object to `plot()` will produce scatterplots for the bootstrapped values for each pair of parameters (Figure 10). For example,

```
> plot(bootTypical)
```

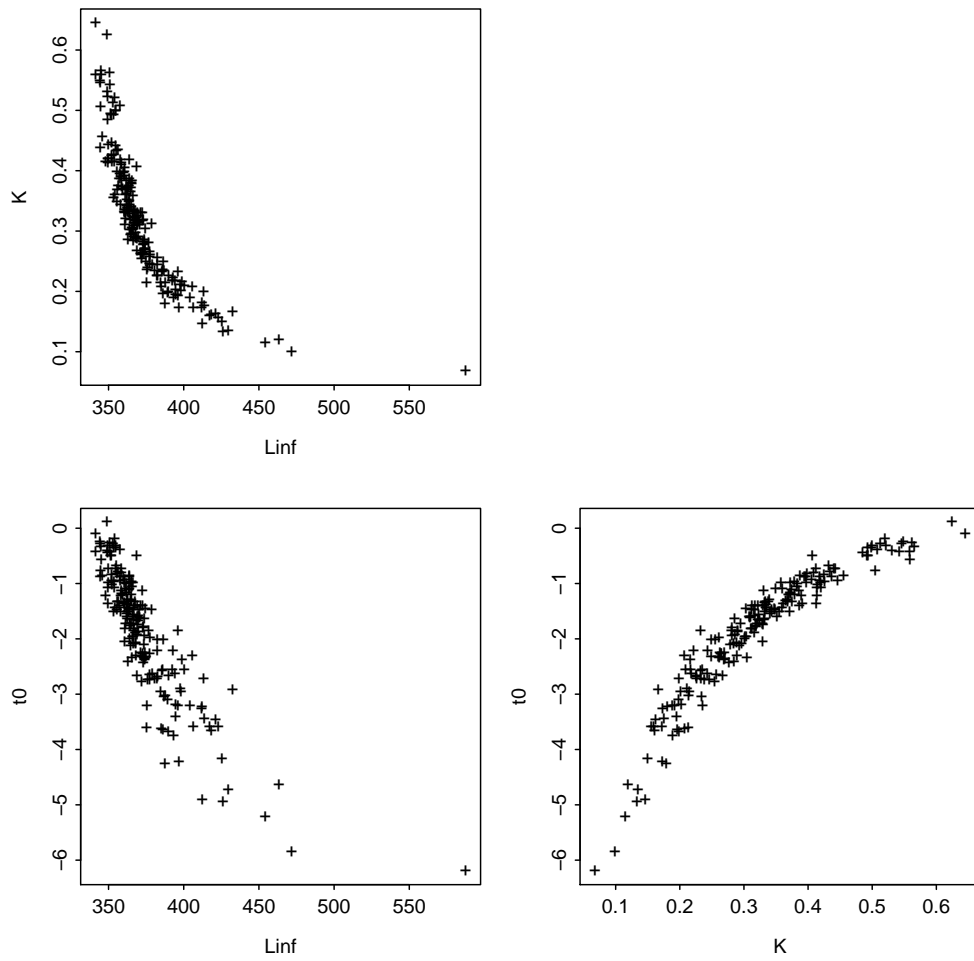


Figure 10. Paired scatterplots for the parameters of the bootstrap results for the typical VBGM for the male Atlantic Croaker data.

These results further illustrate the strong correlation between pairs of the parameters.

2.2.3 Assumption Checking

A non-linear regression, as it has been described here, requires that the variability about the model is constant (i.e., homoscedasticity), the errors are normally distributed, the model adequately fits the data, and there are no influential or outlying points. There are tests to determine if these assumptions have been violated but these tests can be hyper-sensitive (i.e., tend to identify assumption violations) especially with large sample sizes. Thus, the adequacy of meeting these assumptions can be better determined by objectively analyzing two important graphics. The first graphic is a residual plot which plots the model residuals versus the fitted values (Figure 11-Left). In general, the assumptions of the model are fit if NO pattern is observed in the residual plot. Curvature in this plot would suggest that the model does not represent the data very well and a “funneling” from left-to-right would suggest that the variability around the model is not constant (i.e., heteroscedasticity). The second graphic is a histogram of the residuals (Figure 11-Right). In general, the assumption of normality is adequately met if this histogram is symmetric without overly long “tails.”

The residual plot is constructed by submitting the saved `nls()` object to `residPlot()`. The histogram is constructed by submitting the saved `nls()` object to `residuals()` and then submitting this result to `hist()`. These plots for the male Atlantic Crocker data (Figure 11) are constructed with

```
> residPlot(fitTypical)
> hist(residuals(fitTypical),main="")
```

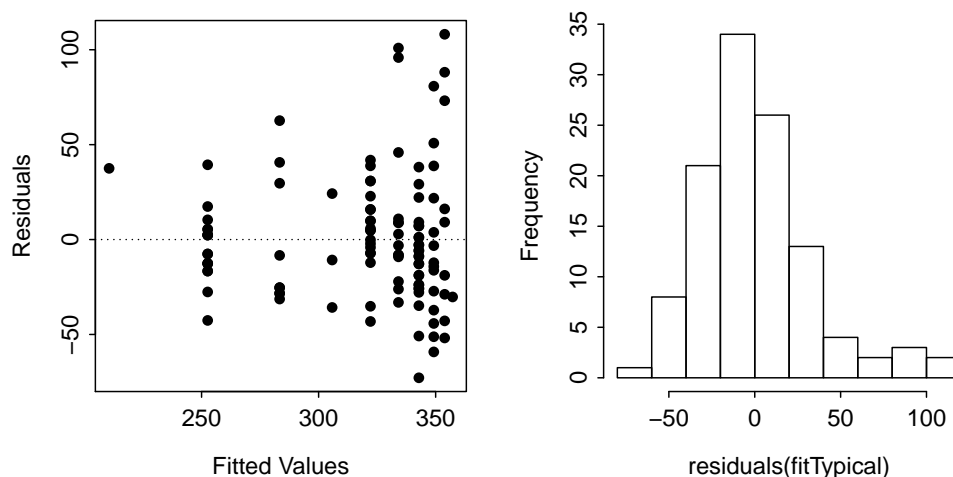


Figure 11. Residual plot (left) and residual histogram (right) from fitting the typical VBGM to the male Atlantic Crocker data.

The residual plot suggests a slight increase in variability with increasing fitted values. The histogram suggests a slight right-skewed distribution of residuals. One method for dealing with a non-constant variance is shown in Section 4.1.

3 Other Parameterizations of the VBGM

3.1 The Models and Their Parameters

A model can often be cast into a different parameterization where the model is functionally the same – i.e., predictions are exactly the same – but it has different parameters. Different parameterizations of models are created for a variety of reasons but two important reasons are that the re-parameterized model (i) has parameters for which the interpretation meets some need and (ii) has parameters that are less correlated.

Each parameterization can ultimately be shown to be equivalent via algebra. The VBGM has been cast in at least five different parameterizations, each of which is discussed in this section.

3.1.1 Original Model Parameterization

The VBGM first proposed by von Bertalanffy (Cailliet *et al.* 2006) is

$$E[L|t] = L_\infty - (L_\infty - L_0) e^{-Kt} \quad (4)$$

where most items are defined as for (1) and L_0 is the mean length at time zero (i.e., birth)¹⁰. Visualizations for the parameters are shown in Figure 12. This parameterization of the VBGM, despite being the model proposed by von Bertalanffy, has been relatively (as compared to the typical VBGM) rarely used in the literature. However, Cailliet *et al.* (2006) recommended its use with chondrichthians.

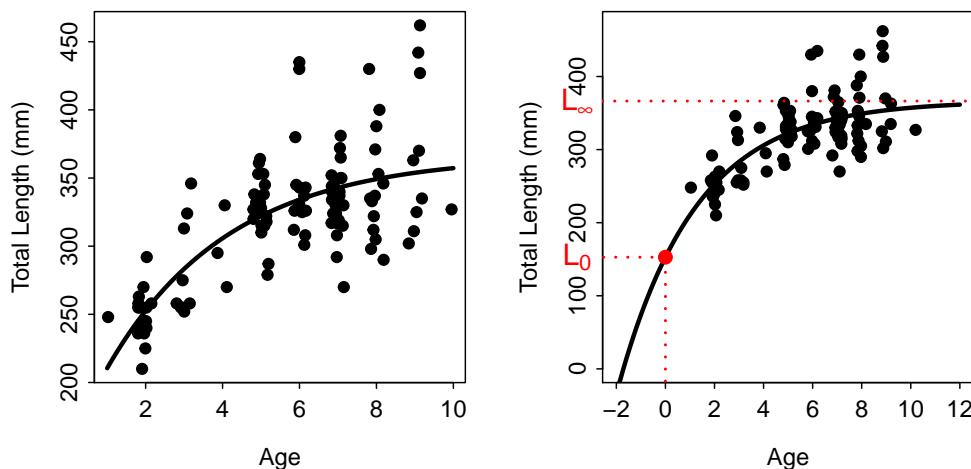


Figure 12. Example of the “original” parameterization of the VBGM fit to size-at-age Atlantic Croaker data. The plot on the left is the fitted model expressed over the observed ages and lengths of the data. The plot on the right is the same fitted model but expressed over a range of ages and lengths that allows for illustrating the meaning of L_∞ and L_0 .

3.1.2 Galucci and Quinn (1979) Parameterization

Galucci and Quinn II (1979) noted that comparisons of “growth” between two groups should involve both K and L_∞ . However, because of the generally high correlation between these two parameters, simultaneous hypothesis tests of these two parameters are compromised and difficult to interpret. To aid comparison between two groups, Galucci and Quinn II (1979) introduced a new parameter, $\omega = KL_\infty$ which, by solving this quantity for L_∞ and substituting into (1), yields yet another reparameterization of the VBGM,

$$E[L|t] = \frac{\omega}{K} \left(1 - e^{-K(t-t_0)} \right) \quad (5)$$

Galucci and Quinn II (1979) state that ω can be thought of as a growth rate because the units are in length-per-time and, in fact, it is representative of the growth rate near t_0 . Furthermore, they claim that ω is the appropriate parameter to use when comparing populations because of its statistical robustness.

¹⁰Cailliet *et al.* (2006) notes that $L_0 = L_\infty(1 - e^{kt_0})$

3.1.3 Mooij *et al.* (1999) Parameterization

Mooij *et al.* (1999) applied the same concept as Galucci and Quinn II (1979) to (4) rather than (1) to yield the following parameterization,

$$E[L|t] = L_\infty - (L_\infty - L_0)e^{-\frac{\omega}{L_\infty}t} \quad (6)$$

The parameters of this model are interpreted as before with the exception that ω is representative of the growth rate near L_0 .

3.1.4 Schnute Parameterization

All previous parameterizations have two major difficulties – (1) highly correlated parameters and (2) some parameters that are largely extrapolations (the “positions” of L_∞ , L_0 , and t_0 are rarely represented in the data). The VBGM parameterizations shown in this, and the next, section are largely based on “expected values” at known ages and, thus, largely alleviate these difficulties (Ratkowsky 1986). The so-called “Schnute” parameterization (from Quinn II and Deriso (1999)) of the VBGM is,

$$E[L|t] = L_1 + (L_2 - L_1) \frac{1 - e^{-K(t-t_1)}}{1 - e^{-K(t_2-t_1)}} \quad (7)$$

where L_1 is the average length at the youngest age, t_1 , and L_2 is the average length at the oldest age, t_2 , *in the sample*.

It is important to note that this parameterization is not more or less parsimonious than any of the previous parameterizations as it still has three parameters – L_1 , L_2 , and K (note that t_1 and t_2 are constants). However, it does have two major advantages over the previous parameterizations. First, the parameters in this parameterization are less correlated than the parameters in other parameterizations (Galucci and Quinn II (1979) and see Section 3.2) and, thus, more stable. Second, this parameterization is directly comparable to the general growth model developed by Schnute (1981). The major drawback of this parameterization is that comparison to results in the literature or from previous studies is difficult, as the typical parameterization is far more prevalent. Schnute and Fournier (1980) did show, however, that the point estimate of L_∞ and t_0 can be obtained from the parameters in the Schnute parameterization as follows,

$$\begin{aligned} L_\infty &= \frac{L_2 - L_1 e^{-K(t_2-t_1)}}{1 - e^{-K(t_2-t_1)}} \\ t_0 &= t_1 + \frac{1}{K} \ln \left(\frac{L_2 - L_1}{L_2 - L_1 e^{-K(t_2-t_1)}} \right) \end{aligned} \quad (8)$$

A common mistake in the interpretation of the Schnute parameterization is to equate L_2 and L_∞ . However, $L_2 \neq L_\infty$ as L_∞ is the average size at the theoretical maximum age and L_2 is the average size at the maximum age *in the sample*. Thus, $L_2 \leq L_\infty$ because the maximum age in the sample is either less than or equal to the theoretical maximum age (Figure 13).

3.1.5 Francis (1988) Parameterization

Francis (1988) provided yet another parameterization of the VBGM in his paper comparing the fit of VBGMs to length-at-age data and to tag-recapture data. His basic argument was that the meanings of L_∞ and K depend on which type of data (and corresponding model) is being fit and that this is an undesirable property. He proposed the following parameterization as a correction to this problem,

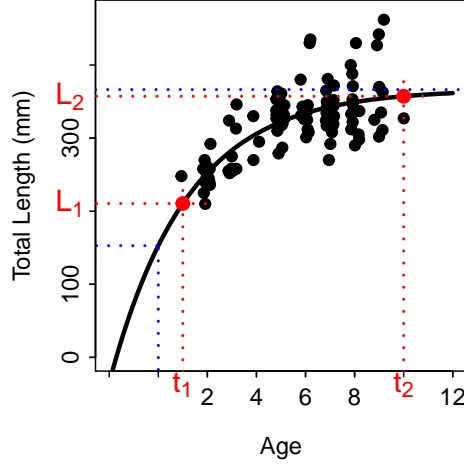


Figure 13. Example of the Schnute parameterization of the VBG fit to size-at-age Atlantic Croaker data expressed over a range of ages and lengths that allow illustrating the meaning of L_1 and L_2 . In addition, the blue lines correspond to L_∞ from Figure 3 and L_0 from Figure 12 for comparative purposes.

$$E[L|t] = L_1 + (L_3 - L_1) \frac{1 - r^{2 \frac{t-t_1}{t_3-t_1}}}{1 - r^2} \quad (9)$$

where

$$r = \frac{L_3 - L_2}{L_2 - L_1}$$

where L_1 , L_2 , and L_3 are the mean lengths at ages t_1 , t_2 , and t_3 , respectively. The t_1 and t_3 are arbitrary reference ages (and t_2 is half-way between each) but are generally a relatively young (i.e., t_1) and old (i.e., t_3) age. Note that r is used to simplify the presentation and is not a new parameter. The resultant parameters from this model are effectively the predicted mean lengths-at-age for the three chosen ages (Figure 14).

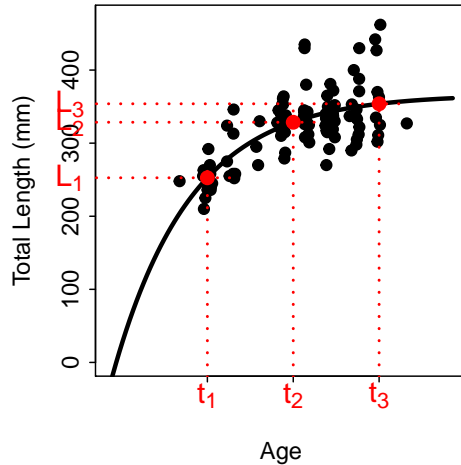


Figure 14. Example of the Francis parameterization of the VBG fit to size-at-age Atlantic Croaker data expressed over a range of ages and lengths that allow illustrating the meaning of L_1 , L_2 , and L_3 . Note that the arbitrary ages chosen were $t_1=2$ and $t_3=9$.

Francis (1988) showed that point estimates of the typical parameters could be derived from these modified parameters as follows,

$$\begin{aligned}
 L_{\infty} &= L_1 + \frac{L_3 - L_1}{1 - r^2} \\
 K &= \frac{-2\log(r)}{t_3 - t_1} \\
 t_0 &= t_1 + \frac{1}{K} \log\left(\frac{L_{\infty} - L_1}{L_{\infty}}\right)
 \end{aligned}
 \tag{10}$$

3.2 Parameterization Comparisons

As noted previously, the different parameterizations of the VBGM are the same model with different parameters (Figure 15). Estimates of parameters that appear in more than one of the parameterizations will be the same among the parameterizations (Table 2). The parameterizations based on expected values (i.e., Schnute and Francis) will generally require fewer iterations to converge when fitting and will result in lower correlations among parameter estimates (Table 2).

Table 2. Parameter estimates and model results from fitting different parameterizations of the VBGM to the male Atlantic Croaker data. Note that L_1 , L_2 , and L_3 are not comparable between parameterizations; L_1 , L_2 , and L_3 correspond to ages 2, 5.5, and 9 in the Francis parameterization; “iters” is the number of iterations to convergence; and r_{max} and r_{mean} are the maximum and average absolute value of correlation coefficients among the three parameters.

Models	L_0	L_{∞}	K	t_0	ω	L_1	L_2	L_3	SE	iters	r_{max}	r_{mean}
Typical	-	366	0.31	-1.71	-	-	-	-	33	4	0.97	0.93
Original	153	366	0.31	-	-	-	-	-	33	4	0.95	0.89
Galucci Quinn	-	-	0.31	-1.71	115.3	-	-	-	33	4	1.00	0.98
Mooij	153	366	-	-	115.3	-	-	-	33	4	0.93	0.89
Schnute	-	-	0.31	-	-	210	357	-	33	4	0.83	0.71
Francis	-	-	-	-	-	253	329	354	33	4	0.08	0.05

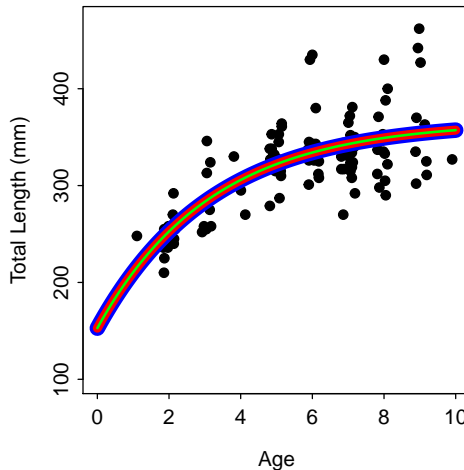


Figure 15. Fits of the typical (blue), Schnute (red), and Francis (green) parameterizations of the VBGM to size-at-age Atlantic Croaker data (auto-generated starting values were used for all parameterizations). Note that the results of all fits are identical and, thus, the fitted lines are directly on top of each other. Different colors and different line widths were used to illustrate this point but may not be readily apparent on the screen or printed page.

3.3 Fitting Alternative VBGMs in R

Methods for finding starting values for L_∞ , K , and t_0 were discussed in Section 2.2.1. Those methods still work for other VBGM parameterizations using these parameters. A starting value for the ω parameter is simply the product of \tilde{L}_∞ and \tilde{K} . A starting value for L_0 can be obtained by algebraically solving (4) for L_0 , i.e.,

$$\tilde{L}_0 = \tilde{L}_\infty + \frac{\bar{L}_i - \tilde{L}_\infty}{e^{\tilde{K}i}}$$

or finding the y-intercept of the second-degree polynomial fit to the mean length-at-age data (Figure 16).

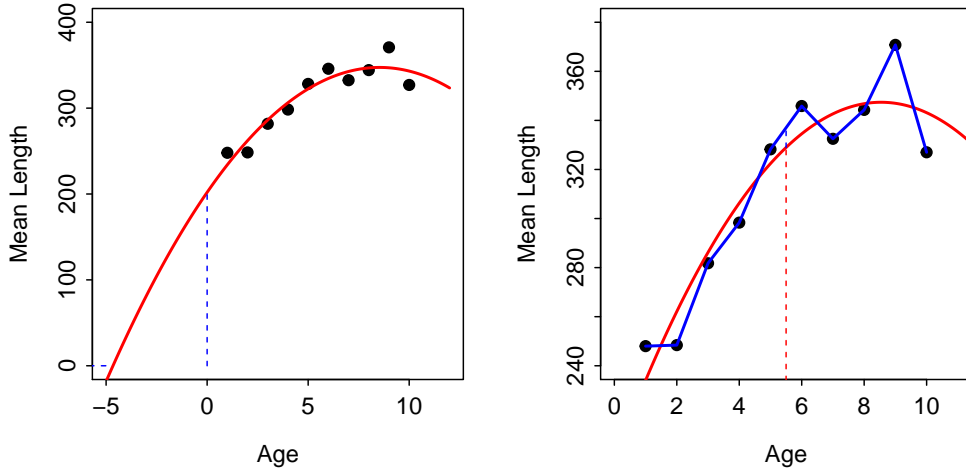


Figure 16. Illustrations for use of the polynomial (solid red) and the linear interpolation (solid blue) functions to estimate starting values. The vertical line in the left plot shows how L_0 is estimated and the horizontal line shows how t_0 is estimated with the polynomial function. The red vertical line in the right plot shows how the starting value for L_2 at $t_2=5.5$ in the Francis parameterization is estimated using the polynomial function and the blue segment shows the difference in estimating L_2 between the polynomial and the interpolating function (i.e., the vertical red plus the vertical blue line illustrate the estimate using the interpolating function).

Starting values for the L_1 , L_2 , and L_3 parameters of the Schnute and Francis models can be obtained with the means lengths at the t_1 , t_2 , and t_3 ages. However, especially in the Schnute method where t_1 and t_2 are the mean lengths at the minimum and maximum ages, respectively, these values may be problematic due to low sample sizes. An alternative to using the sample means is to predict mean lengths from the fits of a second-degree-polynomial to the mean length-at-age data (Figure 16). The polynomial function is useful in that it “averages” across all ages regardless of sample size. The negative side of the polynomial function is that its shape may be erratic at the ends – i.e., near the minimum and maximum age. Another problem occurs in the Francis method if the “middle age” (i.e., t_2) is a fractional age and not specifically represented in the data. This problem can be alleviated by linearly interpolating the mean length at the fractional age from the mean lengths at the two closest ages or by predicting the mean length at the fractional age from the polynomial regression (Figure 16).

The methods described above for generating reasonable starting values are also implemented in `vbStarts()`. The first and second arguments to this function are exactly as described in Section 2.2.1. However, a third, or `type=`, argument is required and is a string indicating which parameterization to use – choices are “original”, “typical” (the default), “GalucciQuinn”, “Mooij”, “Schnute”, and “Francis”. If the Francis method is used then a numeric vector of the three ages corresponding to the three parameters must be entered into the `tFrancis=` argument. The default settings are to use the polynomial regression method to produce starting values for L_0 . Starting values for L_0 based on the re-arrangement of the original parameterization can be obtained by using the `meth0=“yngAge”` argument. The means method is the default method for producing

starting values for L_1 , L_2 , and L_3 . The polynomial function can be used to estimate these parameters by using the `methEV="poly"` argument. For example, the starting values for the original, Galucci and Quinn, and Francis parameterizations of the VBGM for the male Atlantic Croaker data are obtained with

```
> svOriginal <- vbStarts(tl~age,data=crm,type="original")
> unlist(svOriginal)      # unlist used only to save space when viewing the results
      Linf      L0      K
340.5069 201.9713  0.3517
> svGQ <- vbStarts(tl~age,data=crm,type="GalucciQuinn")
> unlist(svGQ)
      omega      K      t0
119.7647  0.3517 -1.6732
> svFrancis <- vbStarts(tl~age,data=crm,type="Francis",tFrancis=c(2,9))
> unlist(svFrancis)
      L1      L2      L3
262.1 329.0 347.0
```

Starting values for these parameterizations can also be obtained with `growthModelSim()` as described in Section 2.2.1, with the exception that the first argument must be one of "vbOriginal", "vbGalucciQuinn", "vbMooij", and "vbSchnute"¹¹.

3.3.1 Example Fitting with Francis Parameterization

The fitting of the Francis (1988) VBGM is similar to the fitting of the typical VBGM but has a different “feel.” First, one should determine the ages present for all fish in the sample and use this to determine the ages to use for t_1 and t_3 in (9). A table of age frequencies is obtained with

```
> table(crm$age)

 1  2  3  4  5  6  7  8  9 10
 1 17  9  3 20 14 26 14  9  1
```

From this, one can see that ages range between 1 and 10 and reasonable choices may be $t_1=2$ and $t_3=9$ ¹² such that $t_2=\frac{2+9}{2}=5.5$. It is easiest to enter the two endpoint values into a vector with

```
> ages <- c(2,9)
```

Reasonable starting values are then obtained as shown previously and stored in `svFrancis`.

This parameterization of the model is a bit cumbersome to write so it is easier to use the `vbFuns()` convenience function. If you did enter the model directly into R it would look like this

```
> vbFrancis <- t1~L1 + (L3-L1)*(1-((L3-L2)/(L2-L1))^(
  2*(age-ts[1])/(ts[3]-ts[1]))) / (1-((L3-L2)/(L2-L1))^2)
```

However, when using `vbFuns()`, the Francis model is declared with

```
> vbFrancis <- vbFuns("Francis")
```

¹¹There is not a method for the Francis (1988) parameterization in `growthModelSim()`.

¹²I chose not to use the youngest and oldest ages because of the very small sample sizes at these ages.

This model is then fit to the male Atlantic Croaker data, and the results are seen, with

```
> fitFrancis <- nls(tl~vbFrancis(age,L1,L2,L3,t1=ages[1],t3=ages[2]),
  data=crm,start=svFrancis)
> overview(fitFrancis)

-----
Formula: tl ~ vbFrancis(age, L1, L2, L3, t1 = ages[1], t3 = ages[2])

Parameters:
  Estimate Std. Error t value Pr(>|t|)
L1   252.60      7.01   36.1 <2e-16
L2   328.60      4.38   75.1 <2e-16
L3   353.85      6.57   53.8 <2e-16

Residual standard error: 33.4 on 111 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 1.35e-06

-----
Residual sum of squares: 124000

-----
Asymptotic confidence interval:
  2.5% 97.5%
L1 238.7 266.5
L2 319.9 337.3
L3 340.8 366.9

-----
Correlation matrix:
      L1      L2      L3
L1  1.00000 -0.02607  0.03304
L2 -0.02607  1.00000 -0.08039
L3  0.03304 -0.08039  1.00000
```

Note the lower correlations among parameters in this parameterization of the model.

The bootstrapped results, obtained with the code below, show fairly symmetric distributions for each parameter (Figure 17) and very weak correlations among each pair of parameters (Figure 18). The fairly symmetric distributions lead to very similar asymptotic and bootstrap confidence intervals.

```
> bootFrancis <- nlsBoot(fitFrancis,niter=200) # B=200 is too low, use approx B=1000
> confint(bootFrancis,plot=TRUE)

  95% LCI 95% UCI
L1   238.4  263.3
L2   320.4  337.5
L3   344.6  367.6

> plot(bootFrancis)
```

The predicted values with bootstrapped confidence intervals are constructed exactly as shown with the typical VBGM except that it is easier to construct the predictions for each bootstrap sample in parts due to the more complicated coding required to implement the Francis parameterization. The predicted length, with associated bootstrapped confidence interval, for a fish of age-8 is constructed with

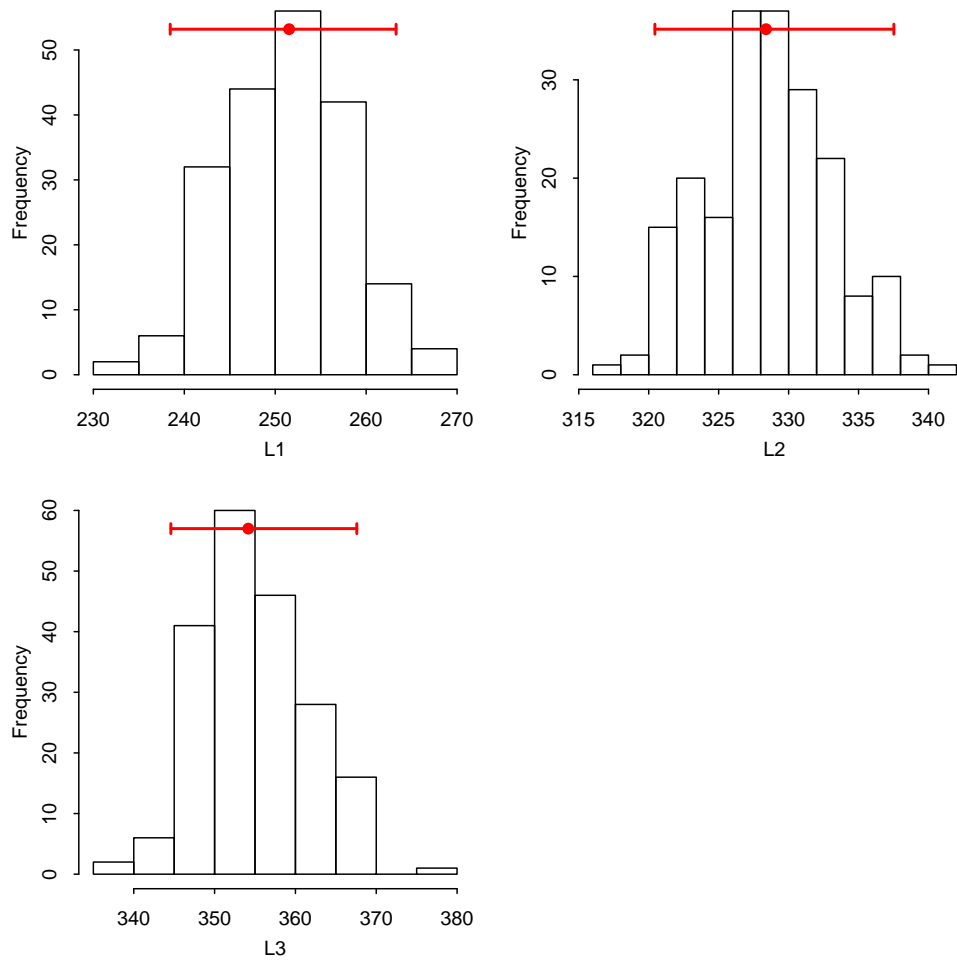


Figure 17. Histogram of the bootstrap results for the Francis VBGM for the male Atlantic Croaker data. Red horizontal lines represent the 95% bootstrap confidence intervals.

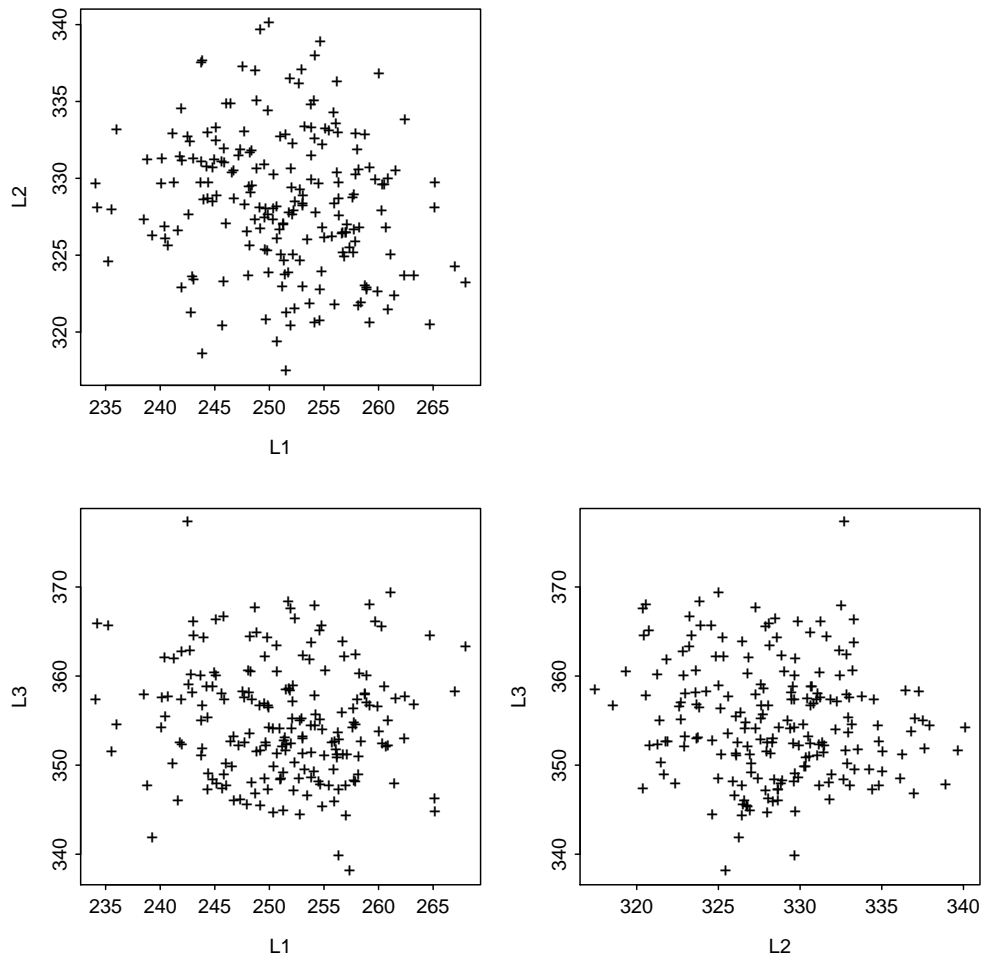


Figure 18. Paired scatterplots for the parameters of the bootstrap results for the Francis VBG for the male Atlantic Croaker data.

```

> new <- data.frame(age=8)
> predict(fitFrancis, new)
[1] 349.2
> ests <- bootFrancis$coefboot
> L1 <- ests[,"L1"]
> L2 <- ests[,"L2"]
> L3 <- ests[,"L3"]
> r <- (L3-L2)/(L2-L1)
> pv <- L1 + (L3-L1)*(1-r^(2*(8-ages[1])/(ages[2]-ages[1]))) / (1-r^2)
> quantile(pv, c(0.025, 0.975))
 2.5% 97.5%
342.0 358.3

```

Thus, the mean length of all age-8 male Atlantic Croakers is between 342 and 358 mm, which is very similar to what was observed when the typical VBGM was used.

4 Other Considerations When Fitting VBGMs

4.1 Multiplicative or Lognormal Error Structures

The model fitting discussed in the previous sections implicitly assumes that the errors around the VBGM are normally distributed and additive. If the errors are denoted by ϵ then, for example, the typical parameterization of the VBGM (i.e., (1)) would be modified to,

$$L_i = L_\infty \left(1 - e^{-K(t_i - t_0)}\right) + \epsilon_i \quad (11)$$

where i is a subscript index for individual fish and $\epsilon \sim N(0, \sigma)$. This model tacitly assumes that the variability in length is constant at all ages. In other words, for instance, the variability in the length of age-1 fish is the same as the variability in the length of age-10 fish. In many instances this assumption will not be true as variability in length tends to increase with increasing length and, thus, age. Smaller fish tend to be more alike than larger fish.

A multiplicative error structure essentially assumes that variability changes, usually increases, in proportion to the expected value of the response variable. Under the assumption of multiplicative errors, the typical parameterization of the VBGM is modified to,

$$L_i = L_\infty \left(1 - e^{-K(t_i - t_0)}\right) e^{\epsilon_i} \quad (12)$$

Taking logarithms of (12) results in

$$\log(L_i) = \log \left(L_\infty \left(1 - e^{-K(t_i - t_0)}\right) \right) + \epsilon_i \quad (13)$$

which shows that the error structure is additive on the log scale. Thus, typical non-linear estimation techniques can be used with the data on the log scale to estimate the model parameters.

A comparison of means and standard deviations of length-at-age often suggests whether the variability increases with increasing length or not. A simple way to construct these summaries is to use `Summarize()` from the FSA package as follows,

```
> Summarize(tl~age,data=crm,numdigs=1)
Warning: To continue, variable(s) on RHS of formula were converted to a factor.
  age  n  mean    sd min  Q1 median  Q3 max percZero
1    1  1 248.0    NA 248 248   248 248 248      0
2    2 17 248.4 18.47 210 240   245 258 292      0
3    3  9 281.8 36.02 252 255   258 313 346      0
4    4  3 298.3 30.14 270 282   295 312 330      0
5    5 20 328.2 21.89 279 317   328 340 364      0
6    6 14 345.9 41.44 301 325   334 344 435      0
7    7 26 332.5 23.53 270 319   334 344 381      0
8    8 14 344.3 40.76 290 314   336 366 430      0
9    9  9 370.8 59.47 302 325   363 427 462      0
10   10  1 327.0    NA 327 327   327 327 327      0
```

While a strong trend between the standard deviations and the means is not evident, there is some suggestion for an increasing standard deviation with an increasing mean length. This was also evident in the residual plots shown in the previous sections.

The typical parameterization of the VBGMM with a multiplicative error structure is fit with,

```
> crm$logTL <- log(crm$t1) # add log TL variable to crm
> svTypicalM <- vbStarts(tl~age,data=crm,type="typical") # get starting values as usual
> vbTypical <- vbFuns(type="typical") # get RHS of typical function
> fitTypicalM <- nls(logTL~log(vbTypical(age,Linf,K,t0)),data=crm,start=svTypicalM)
> overview(fitTypicalM)

-----
Formula: logTL ~ log(vbTypical(age, Linf, K, t0))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
Linf 368.0112    17.9308   20.52 <2e-16
K      0.2904     0.0943    3.08  0.0026
t0    -1.9902     0.9958   -2.00  0.0481

Residual standard error: 0.099 on 111 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 8.91e-06

-----
Residual sum of squares: 1.09

-----
Asymptotic confidence interval:
      2.5%    97.5%
Linf 332.4801 403.54235
K      0.1035  0.47718
t0    -3.9633 -0.01702

-----
Correlation matrix:
      Linf      K      t0
Linf  1.0000 -0.9561 -0.8851
K     -0.9561  1.0000  0.9762
```



```
t0 -0.8851 0.9762 1.0000
```

The residual plots, constructed with the code below, suggest that the results are slightly less heteroscedastic and slightly less skewed with the multiplicative rather than additive error structure (Figure 19).

```
> residPlot(fitTypicalM)
> hist(residuals(fitTypicalM),main="")
```

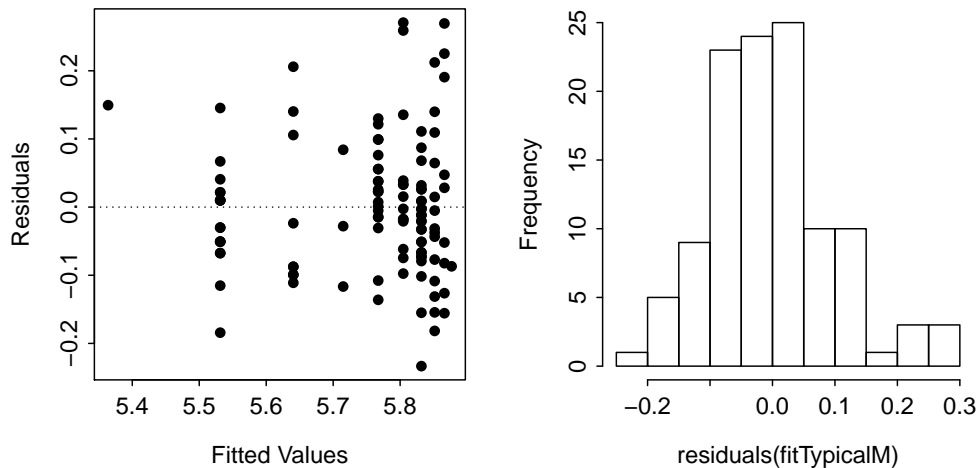


Figure 19. Residual plot (left) and residual histogram (right) from fitting the typical VBGM with multiplicative errors to the male Atlantic Croaker data.

A plot comparing the model fit with additive and multiplicative error structures (constructed with the code below) shows that the different error structures did not result in a major difference in this case. This is likely due to the fact that the residual error was not highly heteroscedastic.

```
> plot(tl~age,data=crm,pch=19)
> curve(vbTypical(x,Linf=coef(fitTypical)),from=1,to=10,col="red",lwd=2,add=TRUE)
> curve(vbTypical(x,Linf=coef(fitTypicalM)),from=1,to=10,col="blue",lwd=2,add=TRUE)
> legend("topleft",legend=c("Additive","Multiplicative"),col=c("red","blue"),lwd=2,
  lty=1,cex=0.75)
```

4.2 Fitting Mean Length-At-Age Data

As mentioned in Section 1, historical methods for fitting the VBGM used mean length-at-age data. In this section, weighted non-linear methods will be used to show how the point estimates, but not the variance estimates, of the parameters can be found from the mean length-at-age data.

Mean length-at-age data can be constructed for the male Atlantic Croaker data with

```
> crm1 <- Summarize(tl~age,data=crm)
Warning: To continue, variable(s) on RHS of formula were converted to a factor.
> str(crm1)
'data.frame': 10 obs. of 10 variables:
 $ age      : Factor w/ 10 levels "1","10","2","3",...: 1 3 4 5 6 7 8 9 10 2
```

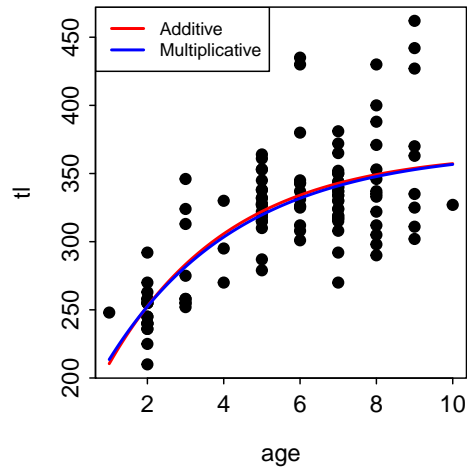


Figure 20. Fitted line plot for the typical VBGm comparing the fits with additive and multiplicative errors to the male Atlantic Croaker data.

```

$ n      : num  1 17 9 3 20 14 26 14 9 1
$ mean   : num  248 248 282 298 328 ...
$ sd     : num  NA 18.5 36 30.1 21.9 ...
$ min    : num  248 210 252 270 279 301 270 290 302 327
$ Q1     : num  248 240 255 282 317 325 319 314 325 327
$ median : num  248 245 258 295 328 334 334 336 363 327
$ Q3     : num  248 258 313 312 340 344 344 366 427 327
$ max    : num  248 292 346 330 364 435 381 430 462 327
$ percZero: num  0 0 0 0 0 0 0 0 0 0

```

However, note that age is a factor variable. The age factor variable can be converted to a numeric variable with `fact2num()` as follows

```

> crm1$age <- fact2num(crm1$age)
> str(crm1)

'data.frame': 10 obs. of 10 variables:
 $ age      : num  1 2 3 4 5 6 7 8 9 10
 $ n       : num  1 17 9 3 20 14 26 14 9 1
 $ mean    : num  248 248 282 298 328 ...
 $ sd     : num  NA 18.5 36 30.1 21.9 ...
 $ min    : num  248 210 252 270 279 301 270 290 302 327
 $ Q1     : num  248 240 255 282 317 325 319 314 325 327
 $ median : num  248 245 258 295 328 334 334 336 363 327
 $ Q3     : num  248 258 313 312 340 344 344 366 427 327
 $ max    : num  248 292 346 330 364 435 381 430 462 327
 $ percZero: num  0 0 0 0 0 0 0 0 0 0

> crm1
  age n mean   sd min  Q1 median  Q3 max percZero
1   1 1 248.0  NA 248 248   248 248 248         0
2   2 17 248.4 18.47 210 240   245 258 292         0
3   3 9 281.8 36.02 252 255   258 313 346         0
4   4 3 298.3 30.14 270 282   295 312 330         0
5   5 20 328.2 21.89 279 317   328 340 364         0
6   6 14 345.9 41.44 301 325   334 344 435         0

```

7	7	26	332.5	23.53	270	319	334	344	381	0
8	8	14	344.3	40.76	290	314	336	366	430	0
9	9	9	370.8	59.47	302	325	363	427	462	0
10	10	1	327.0	NA	327	327	327	327	327	0

A function that performs the “weighting” must then be declared. This function uses a fish’s age and the VBGM to predict a mean length, computes the difference between the observed mean length and this prediction, and multiplies this difference by the square root of the number of fish at that age. The square root of the number of fish at each age is the “weight” that effectively expands the single mean to represent the number of fish that went into the calculation of that mean. This weighting function is declared with

```
> wTypical <- function(meanlen,age,n,Linf,K,t0) {
  pred <- Linf*(1-exp(-K*(age-t0)))
  (meanlen-pred)*sqrt(n)
}
```

The weighted non-linear model is then fit by first finding starting values and then calling `nls()` with a formula that has no left-hand-side and the weighting function as the right-hand-side. This fitting is completed with

```
> svTypicalM <- vbStarts(mean~age,data=crm1)
> wFitTypical <- nls(~wTypical(mean,age,n,Linf,K,t0),data=crm1,start=svTypicalM)
```

The summary of the fit is obtained as usual with

```
> overview(wFitTypical)

-----
Formula: 0 ~ wTypical(mean, age, n, Linf, K, t0)

Parameters:
      Estimate Std. Error t value Pr(>|t|)
Linf  366.415    20.022   18.30  3.6e-07
K       0.315     0.129    2.45  0.044
t0     -1.714     1.253   -1.37  0.214

Residual standard error: 39.9 on 7 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 9.1e-07

-----
Residual sum of squares: 11200

-----
Asymptotic confidence interval:
      2.5%  97.5%
Linf 319.06915 413.760
K     0.01061  0.619
t0   -4.67746  1.249

-----
Correlation matrix:
      Linf      K      t0
```

```

Linf  1.0000 -0.9506 -0.8706
K     -0.9506  1.0000  0.9718
t0    -0.8706  0.9718  1.0000

```

From this it is seen that the point estimates of the parameters are the same as when the model was fit to the individual fish length-at-age data (see output from `overview(fitTypical)` previously). However, the standard errors, residual variance, and confidence intervals are all different. Thus, it is probably prudent to focus only on the parameter estimates with this method, extracted solely with

```

> coef(wFitTypical)
      Linf      K      t0
366.4147  0.3148 -1.7142

```

4.3 Common Model Fitting Problems

The primary problem when fitting a VBGM using `nls()` is the dreaded “failed to converge” message. Sometimes the message can be forced to go away, at the expense of increased computing time, by increasing the maximum number of iterations that `nls()` will use before it quits¹³. However, the problem more often rests with difficulties in the model or difficulties with the data.

Some of the difficulties with the model have been discussed previously. The primary difficulty is that most of the VBGM parameterizations have parameters that are very highly correlated. Parameters that are very highly correlated suggest that there is very little practical difference, in terms of minimizing an error sum-of-squares, between many different combinations of the three parameters. One can think of highly correlated parameters that result in similar error sum-of-squares as a hyper-plane (i.e., think of a “bowl”; see Figure 26) that is very “flat” near the global minimum. The gradient to the global minimum is very shallow and the algorithms move either very slowly or erratically towards that global minimum. The global minimum is, thus, very difficult to find. If appropriate to one’s needs, the user should use the VBGM parameterization that has the least correlated parameters (i.e., the Francis parameterization).

Model convergence can also be aided by simplifying the parameter space over which the error sum-of-squares is being minimized. In other words, there is generally a higher likelihood of convergence if the model has fewer parameters. All parameterizations of the VBGM consist of three parameters, so choice of model does not directly help in this regard. However, the L_0 parameter in the original VBGM parameterization has a very precise biological interpretation – i.e., the average length at age-0 – and has, in some instances (Cailliet *et al.* (2006) provides a list of recent authors from the chondrychthian literature), been set equal to the mean length at hatching determined from other means. The general commands for fitting such a model to estimate the remaining two parameters would be something like

```

> vbOriginal0 <- vbFuns("Original")
> # note no starting value for L0
> svOriginal0 <- list(Linf=300,K=0.3)
> # note constant value (e.g., 5) in L0 position of vb function
> fitOriginal0 <- nls(tl~vbOriginal0(age,Linf,5,K),start=svOriginal0)

```

Cailliet *et al.* (2006) does note that fixing the L_0 value to a constant can greatly impact the VBGM parameter estimates and suggests that if L_0 is fixed that care should be taken to produce the best estimate of L_0 as possible.

A common data difficulty is the lack of age-length data from relatively young fish. Without young fish in the sample, the length-at-age plot may be primarily linear which makes it very difficult to accurately represent the curvature in the “early” part of the VBGM. This problem is illustrated in Figure 21. The model did converge in this example but the much “flatter” VBGM resulted in a dramatically different estimate of K .

¹³The maximum number of iterations can be increased by using `nls.control(maxiter=10000)` before the `nls()` call

Data from younger fish is the primary solution to this problem, though fixing the L_0 value at a constant as described above has also been used to “anchor” the left-side of the VBGM.

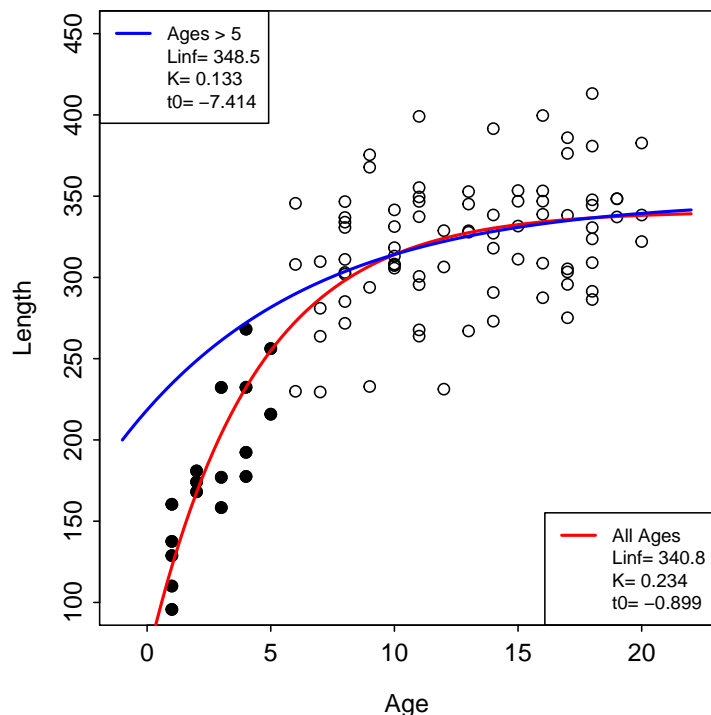


Figure 21. Hypothetical length-at-age data with the fits of the typical VBGM to fish of all ages and only those fish age-6 and older. Data was simulated using `vbGenData(100,300,0.3,-1,minAge=1,maxAge=20,dataType="atCapture")`.

The other data problem occurs at the other end of the distribution – not enough old fish to accurately characterize the asymptote. This problem can be a result of sampling a population where the mortality rate is high enough that fish do not live long enough to approach the asymptote (Francis 1988), from difficulties capturing large old fish (i.e., a size-selective gear is used), or from under-aging older fish (i.e., the larger, older fish would appear younger). In the example in Figure 22, a hypothetical sample of slow-growing long-lived (to age-40) fish was created and it was then assumed that only fish age-15 and younger or fish age-20 and younger were actually sampled. When only age-15 and younger fish were sampled, the length-at-age data looked approximately linear resulting in a very large estimate of L_∞ and, because of the strong correlation with L_∞ , a very low estimate of K . However, when fish age-20 and younger were used, the length-at-age data contained enough of the “curve” in the VBGM to provide parameter estimates similar to when all known fish were used.

The discussion of this section assumes that the VBGM was the proper model to represent fish growth. Other models of fish growth (e.g., Logistic, Gompertz, Richards, Schnute) offer different growth trajectories (e.g., sigmoidal) or more flexibility in identifying a growth trajectory. While the VBGM may be the most prevalent growth model in the literature (Haddon 2001) it is not always the “best” model and some have even argued that it is an inappropriate model (Knight 1968; Roff 1980). For example, Knight (1968) showed that the model fit to data from a left- and right-truncated age structure produced incredibly unrealistic parameter estimates. As another example, Katsanevakis and Maravelias (2008) found that the VBGM, when compared to three other possible models, was the “best” model in only about one-third of the 133 data sets examined. In addition, the Gompertz model is widely used in place of the VBGM when modeling growth during larval or early life stages (Ricker 1979).

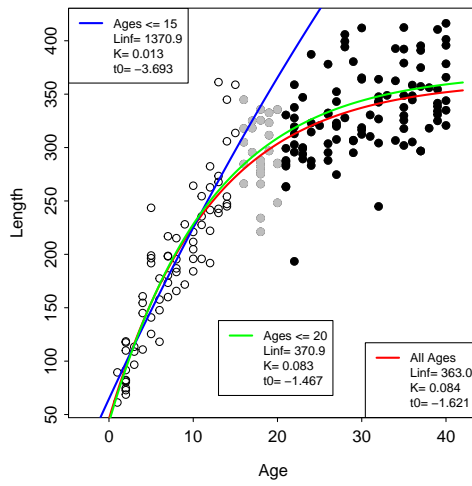


Figure 22. Hypothetical length-at-age data with the fits of the typical VBGM to fish of all ages, only those fish age-15 and younger, and only those fish age-20 and younger. Data was simulated using `vbGenData(200,300,0.12,-1,minAge=1,maxAge=40,dataType="atCapture")`.

5 VBGM Comparison Between Groups

A fairly common analysis in fisheries research is to compare parameters from a growth model between two groups or populations. The statistical methods for making these comparisons requires the use of indicator variables. In this section, the methods for comparing parameters among two groups will be illustrated with the typical VBGM for fish length. This illustration will focus on the comparison of male and female Atlantic Croakers and will, thus, re-run to the full data set in `Croaker2`. A re-examination of the structure of the `Croaker2` data frame shows that `tl` is the “length”, `age` is the “age”, and `sex` is the “group” factor variable.

```
> str(Croaker2)
'data.frame': 318 obs. of 3 variables:
 $ age: int  1 1 1 2 2 2 2 2 2 2 ...
 $ tl : int  243 247 248 330 320 285 280 265 260 248 ...
 $ sex: Factor w/ 2 levels "F","M": 1 1 2 1 1 1 1 1 1 1 ...
```

5.1 Models to be Considered

In order to consider different groups, parameters specific to one of the groups must be defined by adding a subscript or, as we will do here, a number in brackets for the group to the parameter. For example, $L_\infty[1]$ is the asymptotic mean length for individuals in group 1 and $K[2]$ is the Brody growth coefficient for individuals in group 2. In some models below, parameters without the bracketed number are the same or are common to both groups. For example, L_∞ is the asymptotic mean length for all individuals, regardless of group.

The situation where each parameter differs between the two groups is written as follows (following the notation used by [Ritz and Streibig \(2008\)](#)),

$$L(t) = \begin{cases} L_\infty[1] (1 - e^{-K[1](t-t_0[1])}) & \text{if in "group 1"} \\ L_\infty[2] (1 - e^{-K[2](t-t_0[2])}) & \text{if in "group 2"} \end{cases} \quad (14)$$

However, (14) can be written more succinctly as follows,

$$L(t) = L_\infty[group] \left(1 - e^{-K[group](t-t_0[group])}\right) \quad (15)$$

where `group` is 1 if in “group 1” and 2 if in “group 2.”

With the definitions above, several models can be considered that will ultimately help determine which parameters are different among groups. These models are defined as follows,

- GENERAL MODEL: *Separate parameter estimates for individuals in each group.*

$$L(t) = L_\infty[group] \left(1 - e^{-K[group](t-t_0[group])}\right)$$

- One parameter in common between groups.

- COMMON L_∞ MODEL.

$$L(t) = L_\infty \left(1 - e^{-K[group](t-t_0[group])}\right)$$

- COMMON K MODEL.

$$L(t) = L_\infty[group] \left(1 - e^{-K(t-t_0[group])}\right)$$

– COMMON t_0 MODEL.

$$L(t) = L_\infty[\text{group}] \left(1 - e^{-K[\text{group}](t-t_0)} \right)$$

- Two parameters in common between groups.

– COMMON L_∞ AND K MODEL.

$$L(t) = L_\infty \left(1 - e^{-K(t-t_0[\text{group}])} \right)$$

– COMMON L_∞ AND t_0 MODEL.

$$L(t) = L_\infty \left(1 - e^{-K[\text{group}](t-t_0)} \right)$$

– COMMON K AND t_0 MODEL.

$$L(t) = L_\infty[\text{group}] \left(1 - e^{-K(t-t_0)} \right)$$

- COMMON MODEL: Same parameter estimates for both groups. This is the typical parameterization of the VBGM.

$$L(t) = L_\infty \left(1 - e^{-K(t-t_0)} \right)$$

Each “one parameter in common” model is a subset of the GENERAL MODEL, each “two parameters in common” model is a subset of two of the “one parameter in common” models, and the COMMON MODEL is a subset of each “two parameters in common” model. Thus, determining which is the best model is a matter of fitting each model and interpreting several `anova()` results or one `AIC()` result.

5.2 Fitting The Models

5.2.1 Fitting the General Model

The analysis of the multiple non-linear models begins by fitting and checking the assumptions for the GENERAL MODEL. If the assumptions are met with this model, then they should be met for all other subset models. Thus, assumption checking only needs to take place on this most complex GENERAL MODEL.

The GENERAL MODEL requires starting values for six parameters. However, the starting values for the common parameters in the COMMON MODEL can generally be used for both groups in the GENERAL MODEL. Starting values for the COMMON MODEL are found with

```
> ( svCom <- vbStarts(tl~age,data=Croaker2) )
$Linf
[1] 434.7

$K
[1] 0.1837

$t0
[1] -1.635
```

and replicated to serve as starting values for the GENERAL MODEL with

```
> ( svGen <- lapply(svCom,rep,2) )
```



```

$Linf
[1] 434.7 434.7

$K
[1] 0.1837 0.1837

$t0
[1] -1.635 -1.635

```

The general model is then declared and fit with

```

> vbGen <- t1~Linf[sex]*(1-exp(-K[sex]*(age-t0[sex])))
> fitGen <- nls(vbGen,data=Croaker2,start=svGen)

```

Assumption diagnostics are computed with

```

> residPlot(fitGen)
> hist(residuals(fitGen),main="")

```

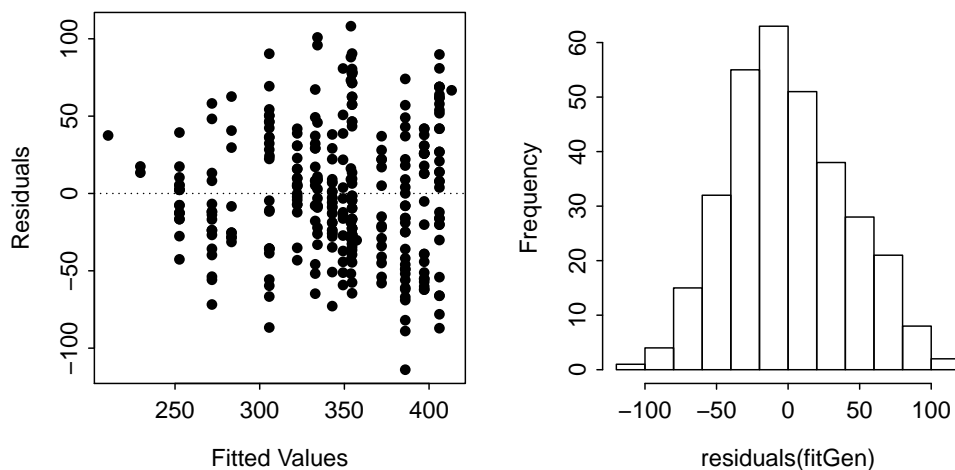


Figure 23. Residual plot (left) and residual histogram (right) from fitting the most general VBGM to the combined male and female Atlantic Croaker data.

The model appears to fit the data appropriately, with slight heteroscedasticity, but approximately normal residuals (Figure 23). Overall, the assumptions are adequately met with this model.

5.2.2 Finding the Best Subset Model

If the assumptions are met for the GENERAL MODEL then each subset model can be fit to the data with the results from each model saved in an appropriately named object. The subset models with corresponding starting values are declared with¹⁴

¹⁴Make sure to note which parameters are separate and which are common between groups and how those appear in the model and starting values list.

```

> vb1KT <- t1~Linf*(1-exp(-K[sex]*(age-t0[sex])))
> sv1KT <- mapply(rep,svCom,c(1,2,2))
> vb1LT <- t1~Linf[sex]*(1-exp(-K*(age-t0[sex])))
> sv1LT <- mapply(rep,svCom,c(2,1,2))
> vb1LK <- t1~Linf[sex]*(1-exp(-K[sex]*(age-t0)))
> sv1LK <- mapply(rep,svCom,c(2,2,1))
> vb2T <- t1~Linf*(1-exp(-K*(age-t0[sex])))
> sv2T <- mapply(rep,svCom,c(1,1,2))
> vb2K <- t1~Linf*(1-exp(-K[sex]*(age-t0)))
> sv2K <- mapply(rep,svCom,c(1,2,1))
> vb2L <- t1~Linf[sex]*(1-exp(-K*(age-t0)))
> sv2L <- mapply(rep,svCom,c(2,1,1))
> vbCom <- t1~Linf*(1-exp(-K*(age-t0)))

```

The starting values for the COMMON K and COMMON K AND T0 models are shown below to illustrate the above declarations¹⁵,

```

> unlist(sv1LT)
  Linf1   Linf2      K    t01    t02
434.6970 434.6970  0.1837 -1.6351 -1.6351
> unlist(sv2L)
  Linf1   Linf2      K    t0
434.6970 434.6970  0.1837 -1.6351

```

All subset models are then fit to the Atlantic Croaker data with

```

> fit1KT <- nls(vb1KT,data=Croaker2,start=sv1KT)
> fit1LT <- nls(vb1LT,data=Croaker2,start=sv1LT)
> fit1LK <- nls(vb1LK,data=Croaker2,start=sv1LK)
> fit2T <- nls(vb2T,data=Croaker2,start=sv2T)
> fit2K <- nls(vb2K,data=Croaker2,start=sv2K)
> fit2L <- nls(vb2L,data=Croaker2,start=sv2L)
> fitCom <- nls(vbCom,data=Croaker2,start=svCom)

```

The model fits are then compared in a hierarchical manner. In other words, each “one parameter in common” model is compared to the GENERAL MODEL. Any “one parameter in common” model that is not statistically different from the GENERAL MODEL is considered “better” than the GENERAL MODEL because it is more parsimonious. If two “one parameter in common” models are better than the GENERAL MODEL, then the one with the smallest RSS is chosen as the “best” “one parameter in common” model. Then, each “two parameters in common” model is compared to the “best” “one parameter in common” model with the same rationale. Finally, if needed, the “best” “two parameters in common” model is compared to the COMMON MODEL. This process stops when a more complex model is significantly different than every simpler subset model. Alternatively, the model with the lowest AIC is the “best” model. These methods are illustrated below.

Each of the “one parameter in common” models is compared to the GENERAL MODEL with

```

> anova(fit1KT,fitGen)
Analysis of Variance Table

Model 1: t1 ~ Linf * (1 - exp(-K[sex] * (age - t0[sex])))
Model 2: t1 ~ Linf[sex] * (1 - exp(-K[sex] * (age - t0[sex])))

```

¹⁵Note that `unlist()` is used here simply to save space.

```

  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     313     546067
2     312     541854  1    4213     2.43  0.12
> anova(fit1LT,fitGen)
Analysis of Variance Table

Model 1: t1 ~ Linf[sex] * (1 - exp(-K * (age - t0[sex])))
Model 2: t1 ~ Linf[sex] * (1 - exp(-K[sex] * (age - t0[sex])))
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     313     542566
2     312     541854  1     712     0.41  0.52
> anova(fit1LK,fitGen)
Analysis of Variance Table

Model 1: t1 ~ Linf[sex] * (1 - exp(-K[sex] * (age - t0)))
Model 2: t1 ~ Linf[sex] * (1 - exp(-K[sex] * (age - t0[sex])))
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     313     542055
2     312     541854  1     200     0.12  0.73

```

These results indicate that each “one parameter in common” models fits the data as well as the more complex GENERAL MODEL. Thus, because each “one parameter in common” model is simpler than the GENERAL MODEL, the GENERAL MODEL will be removed from consideration as the “best model.” Thus, at this point, it is known that separate values for all three parameters are not needed for the two groups.

Of the “one parameter in common” models, the model with t_0 in common is the “best” model because it has the lowest RSS (=542055). Thus, the two “two parameters in common” models that have t_0 in common will be compared to this model with

```

> anova(fit2L,fit1LK)
Analysis of Variance Table

Model 1: t1 ~ Linf[sex] * (1 - exp(-K * (age - t0)))
Model 2: t1 ~ Linf[sex] * (1 - exp(-K[sex] * (age - t0)))
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     314     545239
2     313     542055  1    3184     1.84  0.18
> anova(fit2K,fit1LK)
Analysis of Variance Table

Model 1: t1 ~ Linf * (1 - exp(-K[sex] * (age - t0)))
Model 2: t1 ~ Linf[sex] * (1 - exp(-K[sex] * (age - t0)))
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     314     557513
2     313     542055  1   15458     8.93  0.003

```

The first result suggests that the model with K in common fits the data as well as the model without K in common. The second result indicates that the model withOUT L_∞ in common fits better than the model with L_∞ in common. Thus, the separate L_∞ parameters should “stay” in the model at this point, but a common K parameter can be used.

Finally, the model with K and t_0 in common should be compared to the COMMON MODEL to clarify that L_∞ differs between the groups. This comparison is done with

```

> anova(fitCom,fit2L)
Analysis of Variance Table

Model 1: t1 ~ Linf * (1 - exp(-K * (age - t0)))
Model 2: t1 ~ Linf[sex] * (1 - exp(-K * (age - t0)))
  Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
1     315     647329
2     314     545239  1 102090   58.8 2.2e-13

```

This result shows very strong evidence that the COMMON MODEL does NOT fit the data better than the model with separate L_∞ values. Thus, the model with common K and t_0 parameters, but separate L_∞ parameters, is the model that best fits the data. We conclude that L_∞ , but not K or t_0 , differs between male and female Atlantic Croakers.

Similar results are obtained by computing the AIC for each model with

```

> AIC(fitGen,fit1KT,fit1LT,fit1LK,fit2T,fit2K,fit2L,fitCom)
      df  AIC
fitGen  7 3283
fit1KT  6 3283
fit1LT  6 3281
fit1LK  6 3281
fit2T   5 3295
fit2K   5 3288
fit2L   5 3281
fitCom  4 3333

```

Summary results of the model fit (e.g., coefficient values, measures of variability, parameter correlations) are obtained with

```

> overview(fit2L)

-----
Formula: t1 ~ Linf[sex] * (1 - exp(-K * (age - t0)))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
Linf1 433.1325   19.0672   22.72 < 2e-16
Linf2 388.1380   17.4493   22.24 < 2e-16
K       0.2372    0.0573    4.14 4.5e-05
t0     -2.2583    0.8168   -2.76 0.006

Residual standard error: 41.7 on 314 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 3.55e-06

-----
Residual sum of squares: 545000

-----
Asymptotic confidence interval:
      2.5%   97.5%
Linf1 395.6169 470.6480

```

```

Linf2 353.8056 422.4703
K      0.1244  0.3500
t0     -3.8654 -0.6512

```

```

-----
Correlation matrix:
      Linf1  Linf2      K      t0
Linf1 1.0000  0.9465 -0.9602 -0.8767
Linf2 0.9465  1.0000 -0.9387 -0.8550
K      -0.9602 -0.9387  1.0000  0.9680
t0     -0.8767 -0.8550  0.9680  1.0000

```

From this it is conclude that the females (i.e., “group 1”) has a larger asymptotic mean length then the males.

A fitted line plot can be constructed by first using `plot()` and `points()` to plot different colored points for the female and male fish. A function is then created for the “no difference” common model of the typical VBGM using `vbFuns()`. This function is used within `curve()` to add the fitted lines for both groups, being careful that only the parameters for a particular group are used when adding that group’s fitted line to the plot. For example, the parameter estimates for just the first group are obtained by excluding the parameters for the second group (which consists of only *Linf2* in the second position of the coefficients vector) with

```

> coef(fit2L)[-2]
      Linf1      K      t0
433.1325  0.2372 -2.2583

```

The fitted line plot for the model with only different L_∞ parameters (Figure 24) is constructed with

```

> plot(tl~jitter(age,0.3),data=Croaker2,subset=sex=="F",pch=19,xlab="Age (yrs)",
      ylab="Total Length (mm)")
> points(tl~jitter(age,0.3),data=Croaker2,subset=sex=="M",pch=19,col="gray")
> vbTypical <- vbFuns("typical")
> curve(vbTypical(x,Linf=coef(fit2L)[-2]),from=1,to=10,lwd=2,add=TRUE)
> curve(vbTypical(x,Linf=coef(fit2L)[-1]),from=1,to=10,col="gray",lwd=2,add=TRUE)
> legend("topleft",legend=c("Female","Male"),col=c("black","gray"),lwd=2,lty=1,cex=0.75)

```

As another example, the fitted line plot assuming that the model with different L_∞ parameters and K parameters was chosen (Figure 25) is constructed with

```

> plot(tl~jitter(age,0.3),data=Croaker2,subset=sex=="F",pch=19,xlab="Age (yrs)",
      ylab="Total Length (mm)")
> points(tl~jitter(age,0.3),data=Croaker2,subset=sex=="M",pch=19,col="gray")
> vbTypical <- vbFuns("typical")
> curve(vbTypical(x,Linf=coef(fit1LK)[-c(2,4)]),from=1,to=10,lwd=2,add=TRUE)
> curve(vbTypical(x,Linf=coef(fit1LK)[-c(1,3)]),from=1,to=10,col="gray",lwd=2,add=TRUE)
> legend("topleft",legend=c("Female","Male"),col=c("black","gray"),lwd=2,lty=1,cex=0.75)

```

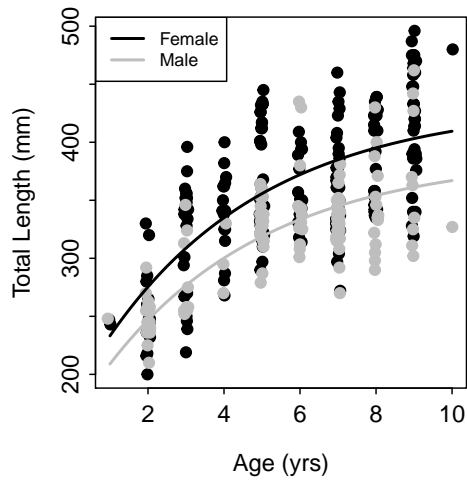


Figure 24. Fitted line plot from fitting the VBGM with separate L_{∞} parameters to male and female Atlantic Croakers.

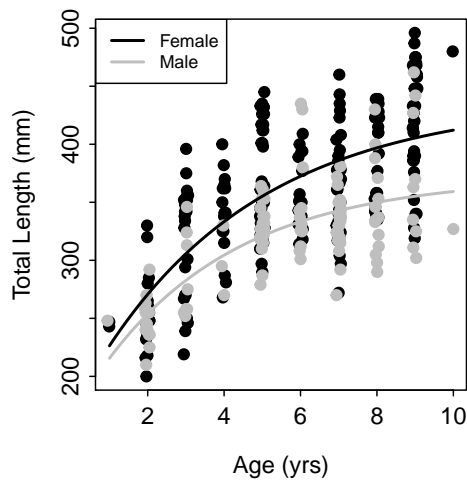


Figure 25. Example of a fitted line plot from fitting the VBGM with separate L_{∞} and K parameters to male and female Atlantic Croakers.

References

- Beverton, R. J. H. 1954. Notes on the use of theoretical models in the study of the dynamics of exploited fish populations. Miscellaneous Contribution 2, United States Fishery Laboratory, Beaufort, North Carolina. [3](#), [4](#)
- Beverton, R. J. H. and S. J. Holt. 1957. On the dynamics of exploited fish populations, *Fisheries Investigations (Series 2)*, volume 19. United Kingdom Ministry of Agriculture and Fisheries, 533 pp. [3](#), [4](#)
- Cailliet, G. M., W. D. Smith, H. F. Mollet, and K. J. Goldman. 2006. Age and growth studies of chondrichthyan fishes: The need for consistency in terminology, verification, validation, and growth function fitting. *Environmental Biology of Fish* 77:211–228. [3](#), [14](#), [28](#)
- Ford, E. 1933. An account of the herring investigations conducted at Plymouth during the years from 1924–1933. *Journal of the Marine Biology Association (U.K.)* 19:305–384. [5](#)
- Fox, J. 2002. *An R and S-PLUS Companion to Applied Regression*. Sage Publications. [41](#)
- Francis, R. 1988. Are growth parameters estimated from tagging and age-length data comparable? *Canadian Journal of Fisheries and Aquatic Sciences* 45:936–942. [3](#), [4](#), [15](#), [16](#), [19](#), [29](#)
- Galucci, V. F. and T. Quinn II. 1979. Reparameterizing, fitting, and testing a simple growth model. *Transactions of the American Fisheries Society* 108:14–25. [14](#), [15](#)
- Haddon, M. J. 2001. *Modelling and Quantitative Methods in Fisheries*. CRC Press. [29](#)
- Jones, C. M. 2000. Fitting growth curves to retrospective size-at-age data. *Fisheries Research* 46:123–129. [3](#)
- Katsanevakis, S. and C. D. Maravelias. 2008. Modelling fish growth: Multi-model inference as a better alternative to *a priori* using von Bertalanffy equation. *Fish and Fisheries* 9:178–187. [29](#)
- Knight, W. 1968. Asymptotic growth: An example of nonsense disguised as mathematics. *Journal of the Fisheries Research Board of Canada* 25:1303–1307. [29](#)
- McKendrick, A. G. and M. K. Pai. 1911. The rate of multiplication of micro-organisms: A mathematical study. *Proceedings of the Royal Society of Edinburgh*. 31:649–655. [43](#)
- Mooij, W. M., J. M. V. Rooij, and S. Wijnhoven. 1999. Analysis and comparison of fish growth from small samples of length-at-age data: Detection of sexual dimorphism in Eurasian perch as an example. *Transactions of the American Fisheries Society* 128:483–490. [15](#)
- Motulsky, H. J. and L. A. Ransnas. 1987. Fitting curves to data using nonlinear regression: A practical and nonmathematical review. *The Federation of American Societies for Experimental Biology Journal* 1:365–374. [41](#), [42](#)
- Quinn II, T. J. and R. B. Deriso. 1999. *Quantitative Fish Dynamics*. Oxford University Press. [15](#)
- Ratkowsky, D. A. 1986. Statistical properties of alternative parameterizations of the von Bertalanffy growth curve. *Canadian Journal of Fisheries and Aquatic Sciences* 43:742–747. [4](#), [15](#)
- Ricker, W. 1975. Computation and interpretation of biological statistics of fish populations. Technical Report Bulletin 191, Bulletin of the Fisheries Research Board of Canada. [3](#)
- Ricker, W. E. 1979. Growth rates and models. *Fish Physiology* 8:677–743. [29](#)
- Ritz, C. and J. C. Streibig. 2008. *Nonlinear Regression with R*. Springer, 144 pp. [31](#)
- Roff, D. A. 1980. A motion for the retirement of the von Bertalanffy function. *Canadian Journal of Fisheries and Aquatic Sciences* 37:127–129. [29](#)
- Schnute, J. 1981. A versatile growth model with statistically stable parameters. *Canadian Journal of Fisheries & Aquatic Sciences* 38:1128–1140. [15](#)
- Schnute, J. and D. Fournier. 1980. A new approach to length-frequency analysis: Growth structure. *Canadian Journal of Fisheries and Aquatic Sciences* 37:1337–1351. [4](#), [15](#), [50](#)

- Walford, L. A. 1946. A new graphic method of describing the growth of animals. *Biological Bulletin* 90:141–147. 5
- Weisberg, S. 2005. *Computing Primer for Applied Linear Regression*, 3rd Edition, using R and SPLUS. Wiley. 43

Appendices

A Fitting Non-Linear Models

Many models in fisheries and ecological research are non-linear. Some of these models can be transformed to a linear scale where traditional linear regression methods can be used. However, some models cannot be “linearized” and linearization may be troublesome with other models. Thus, many types of data are best analyzed by fitting a curve using non-linear regression techniques (Motulsky and Ransnas 1987). The VBGMs are non-linear and cannot be linearized.

A.1 Parameter Estimation

The usual non-linear regression model for individuals is expressed as

$$y_i = f(\vec{\beta}, X) + \epsilon_i \quad (16)$$

where $f(\bullet)$ is a non-linear function that relates the response variable to an explanatory variable (X), $\vec{\beta}$ is a vector of parameters, and $\epsilon_i \sim N(0, \sigma_{Y|X})$ (Fox 2002). The RSS and likelihood functions for these models can be developed in a manner similar to that for linear regression models. However, because the RSS and likelihood functions for non-linear models are non-linear they often require iterative numerical approximation solutions to find the parameter estimates that minimize the RSS or maximize the likelihood function. In other words, calculus cannot be used to solve for parameter estimates as was done with the linear regression models. Instead computers must be used to implement optimization algorithms to estimate the parameters.

Motulsky and Ransnas (1987) provide a nice description of the concept of fitting non-linear models to data. Their description relies on visualizing the RSS surface in a simple example that has two parameters, A and B (Figure 26). In general, the algorithm begins with the user (i.e., you!) supplying a set of starting values for the model parameters. Selection of these starting values is discussed in the next section. The algorithm then finds another set of parameter values that are generally “downhill” from the starting values (a process that involves using the second derivatives of the RSS function). This process is continued such that with each iteration the set of parameter values gets closer and closer to the set of parameter values that are at the minimum RSS value. The algorithm usually continues until the improvement in minimizing RSS from one iteration to the next is considered negligible. The algorithm would proceed similarly if there were more parameters in the model; however, the surface would be more than three-dimensional and, thus, not able to be visualized.

A.2 Choosing Starting Values

The choice of starting values is important to the non-linear regression parameter estimation algorithms. Improper choice of starting values can cause the numerical algorithm to not converge to a minimum RSS value, to behave erratically or not at all (because of problems associated with the second derivatives of the RSS function at the starting values), or to find a local minimum value of RSS rather than the global best-fit minimum (Figure 27).

The process of selecting starting parameter values begins with a thorough understanding of the meaning and impact of each parameter in the non-linear model. Once the parameters are understood, starting values can be obtained in at least three ways:

1. Some parameters can be visually estimated. For example, a parameter that is known to represent the horizontal asymptote of the model can be visually estimated by plotting the data and observing the value of the apparent asymptote.

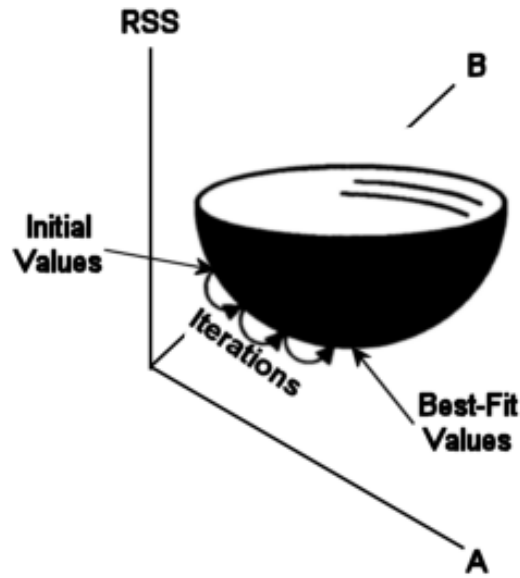


Figure 26. A schematic representation of the RSS as a function of the parameters A and B . Non-linear regression begins at user-defined starting values and iteratively “moves downhill” to find the choices of A and B that minimize the RSS. Graphic was modified from [Motulsky and Ransnas \(1987\)](#).

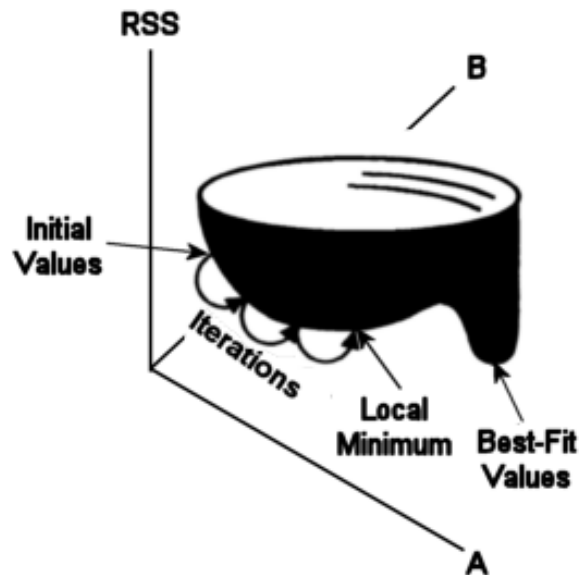


Figure 27. With inappropriate starting values, non-linear regression algorithms might find a local minimum rather than the global best-fit minimum (modified from [Motulsky and Ransnas \(1987\)](#)).

2. Some parameters can be estimated by fixing other parameters or variables at reasonable values (usually from using the previous method) and solving for the parameter.
3. Some parameters can be estimated by superimposing the model on the data and interactively altering the parameters until the model appears to visually fit the data.

The non-linear models used in this vignette will generally (with one exception) use the last strategy. Special purpose functions will be used to visually derive these starting values.

A.3 Non-Linear Models in R & Interpretations

Non-linear models are fit in R with the `nls()` function. The `nls()` function is similar to the `lm()` function in R, but differs in two important ways (Weisberg 2005). First, the user must define *both* the variables and the parameters of the model (in contrast to the `lm()` function where only the variables need to be defined). Second, *starting values* for the parameters must be specified.

The `nls()` function requires three arguments. The first argument is a formula that defines the non-linear model. The formula will typically have the response variable on the right-hand side separated from an algebraic expression of the left-hand side of the model by a `~`. The second argument, named `start=`, is a list of starting values for the parameters of the model. This list also serves to tell R which items in the models are parameters, with the remaining items in the model assumed to be variables. The third argument, named `data=`, is the data frame from which the variables in the model will be drawn. As with the `lm()` function the results from the `nls()` function should be saved to an object.

As an example, consider the logistic growth equation,

$$y_i = \frac{\beta_1}{1 + e^{\beta_2 + \beta_3 x_i}} + \epsilon_i \quad (17)$$

where β_1 is the horizontal asymptote representing the maximum mean of the response variable, β_2 reflects the mean value of the response variable, relative to the asymptote, when the explanatory variable is zero, and β_3 controls the rate of increase or decrease. The fitting of (17) will be illustrated with the number of *Escherchia coli* cells over time from the classic experiment by McKendrick and Pai (1911). The data are read into R and examined as follows,

```
> data(Ecoli)
> str(Ecoli)
'data.frame': 8 obs. of 2 variables:
 $ days : int  1 2 3 4 5 6 7 8
 $ cells: num  0.1 0.2 1.2 3.1 5.1 6 5.9 6
> view(Ecoli)
  days cells
3     3  1.2
4     4  3.1
5     5  5.1
6     6  6.0
7     7  5.9
8     8  6.0
```

The first step in fitting the logistic growth model to these data was to use trial-and-error¹⁶ to identify starting values for the three parameters. The three starting values used in this case were entered into a list object for simplicity,

¹⁶Trial-and-error is not an efficient method to identify starting values. More efficient methods will be described with specific fisheries models. In addition, the starting values for this model could be found by estimating β_1 with the horizontal asymptote in Figure 28 (i.e., $\beta_1 \approx 6$) and then selecting two points from the steepest portion of the curve (i.e., (3,1.2) and (4,3.1)), plugging these into (17) and solving for β_2 and β_3 .

```
> lr.sv <- list(B1=6,B2=7.2,B3=-1.5)
```

The non-linear model in (17) is then fit with `nls()` and the parameter estimates obtained with `summary()` or, as I prefer, `overview()` from the `nlstools` package, as follows,

```
> nl1 <- nls(cells~B1/(1+exp(B2+B3*days)),data=Ecoli,start=lr.sv)
> # summary(nl1) # did not show because redundant with overview()
> overview(nl1)
```

```
-----
Formula: cells ~ B1/(1 + exp(B2 + B3 * days))

Parameters:
      Estimate Std. Error t value Pr(>|t|)
B1    6.0508    0.0731    82.7 4.9e-09
B2    6.2569    0.3464    18.1 9.6e-06
B3   -1.5875    0.0908   -17.5 1.1e-05

Residual standard error: 0.11 on 5 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 6.62e-06

-----
Residual sum of squares: 0.0607

-----
Asymptotic confidence interval:
      2.5% 97.5%
B1  5.863 6.239
B2  5.366 7.147
B3 -1.821 -1.354

-----
Correlation matrix:
      B1      B2      B3
B1  1.0000 -0.4033  0.4909
B2 -0.4033  1.0000 -0.9826
B3  0.4909 -0.9826  1.0000
```

The fit of the non-linear regression model can be constructed with `fitPlot()` using the saved `nls()` object as the first argument. The fitted regression model shown in Figure 28 was constructed with,

```
> fitPlot(nl1,Ecoli,xlab="Day",ylab="Cellsx10^6/ml",cex.main=0.8)
```

A.4 Making Inferences about Model Parameters

Confidence intervals and hypothesis tests can be constructed for the parameters in the non-linear model as they were for the linear model – i.e., using `confint()`. For instance, approximate 95% confidence intervals for the parameters in (17) can be obtained with

```
> confint(nl1)
```

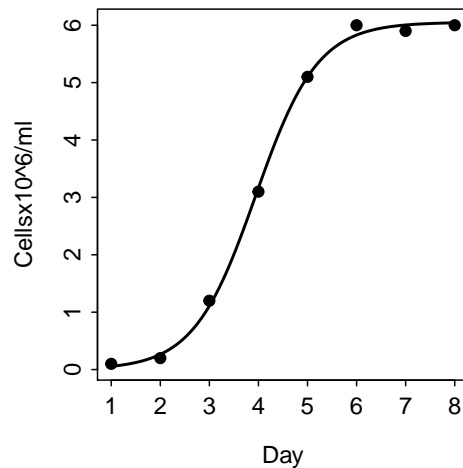


Figure 28. Plot of the number of *Escherichia coli* cells over time with the superimposed fit of the logistic growth equation.

```
Waiting for profiling to be done...
      2.5%  97.5%
B1  5.869  6.240
B2  5.471  7.225
B3 -1.838 -1.382
```

In addition, the t-test and p-value for testing $H_0 : \beta_1 = 0$ versus $H_A : \beta_1 \neq 0$ is shown in the results from `summary()`.

The confidence intervals and hypothesis tests just described, however, may be incorrect in certain situations because the standard errors in non-linear regressions are often poorly estimated. Thus, it is often more appropriate to approximate the confidence interval with a bootstrapping routine. Fortunately, `nlsBoot` will quickly perform the bootstrap method with the results from a non-linear model. The `nlsBoot` function estimates model parameters from a series of `niter` resampled data sets. An approximate 95% CI for a variable can then be computed by finding the values of the bootstrapped parameter estimates with 2.5% of the values smaller and 2.5% of the values greater. This is most easily accomplished by saving the `nlsBoot()` result to an object and submitting that object to `confint()`. For example, a bootstrapped 95% confidence interval for the parameters in the logistic growth model are found with¹⁷,

```
> n11.b <- nlsBoot(n11, niter=200)
> confint(n11.b)
      95% LCI  95% UCI
B1   5.947   6.181
B2   5.825   6.790
B3  -1.733  -1.470
```

In this case, the bootstrapped confidence intervals are substantially different than the confidence intervals based on normal theory. The bootstrapped confidence intervals are likely more appropriate in this case.

¹⁷Note that `B` controls the number of bootstrap samples. The value of `niter=` should be much larger than the 200 shown in this example – e.g., 500 to 5000. A smaller number was used here simply to conserve computing time while producing this document.

A.5 Making Predictions

Predictions of the mean value of the response variable and the response variable for an individual can again be made with `predict()`. The arguments to this function are exactly the same as described for the `lm()` objects – i.e., the first argument is the saved `nls()` object and the second argument is a data frame containing values of the explanatory for which to make predictions. Unfortunately, `predict()` for an `nls` object does not support prediction or confidence intervals as of R version 2.9.1. In other words, `predict()` will only predict the mean value of the response variable at the given value of the explanatory variable. For example, the mean number of *E. coli* cells on day 3 is predicted to be,

```
> predict(nl1, data.frame(days=3))
[1] 1.109
```

A prediction interval can be estimated by using the parameter estimates for each bootstrap sample saved in the `nlsBoot()` object. To make this interval the column corresponding to each parameter must be substituted into the right-hand-side of the non-linear regression model. In addition, the particular value for the explanatory variable must be substituted into the right-hand-side. For example, the 95% prediction interval for the number of cells on day 3 is found with,

```
> pred.day <- 3
> ests <- nl1.b$coefboot
> nl1.bp3 <- ests[, "B1"] / (1 + exp(ests[, "B2"] + ests[, "B3"] * pred.day))
> coli.ci <- quantile(nl1.bp3, c(0.025, 0.975))
> coli.ci
 2.5% 97.5%
0.9878 1.2276
```

Thus, one is 95% confident that the number of cells on any day 3 will be between 0.99 and 1.23.

B Model Derivations

B.1 The von Bertalanffy Model for Length I

One method of deriving the typical VBGM (1) begins by assuming that the rate of growth of an organism declines linearly as the length of the organism increases,

$$\frac{dL(t)}{dt} = K(L_\infty - L(t)) \quad (18)$$

Integration of (18) leads to the intermediate function,

$$\begin{aligned} \frac{dL(t)}{dt} &= K(L_\infty - L(t)) \\ \frac{dL(t)}{L_\infty - L(t)} &= K dt \\ -\log(L_\infty - L(t)) &= kt + c \\ \log(L_\infty - L(t)) &= -kt + c \\ L_\infty - L(t) &= e^c e^{-kt} \\ L_\infty - L(t) &= de^{-kt} \\ L(t) &= L_\infty - de^{-kt} \end{aligned} \quad (19)$$

Note that c and d were both constants of integration in the work above. Now to specifically identify d in (19) consider the *hypothetical* time or age when the average length of the fish is zero, t_0 . More specifically this defines $L(t_0) = 0$ which, when substituted into (19), leads to,

$$\begin{aligned} 0 &= L_\infty - de^{-kt_0} \\ de^{-kt_0} &= L_\infty \\ d &= \frac{L_\infty}{e^{-kt_0}} \\ d &= L_\infty e^{kt_0} \end{aligned} \quad (20)$$

Now substituting (20) into (19) leads to,

$$\begin{aligned} L(t) &= L_\infty - L_\infty e^{kt_0} e^{-kt} \\ L(t) &= L_\infty (1 - e^{kt_0 - kt}) \\ L(t) &= L_\infty (1 - e^{-k(t-t_0)}) \end{aligned}$$

which is (1).

B.2 The von Bertalanffy Model for Length II

A second method for deriving the VBGM is to follow the logic Ludwig von Bertalanffy (1938) who derived the common VBGM (1) from a physiological argument. The weight of an organism at any moment depends upon

the result of two opposing forces: *anabolism*, i.e., the synthesis of tissue, and *catabolism*, i.e., the breakdown of tissue. The rate of anabolism is supposedly proportional to the magnitude of the “resorbing surfaces” of the animal while the rate of catabolism is proportional to the body weight. The size of the “resorbing surfaces” (S) is not directly measurable but it is hypothesized that it is proportional to the square of some linear dimension, say length (L), i.e.,

$$S = pL^2 \quad (21)$$

where p is a proportionality constant. Weight (W) too is taken to be proportional to the cube of some linear dimension, i.e.,

$$W = qL^3 \quad (22)$$

where q is a proportionality constant.

Anabolism and catabolism proceed simultaneously throughout the life of an animal. Thus, the difference between these two forces at any instant defines the rate at which the weight of the animal is changing at that moment. Rates are defined as derivatives, so the instantaneous change in weight over a given time is given by the differential equation

$$\frac{dW(t)}{dt} = aS(t) - cW(t) \quad (23)$$

where a is the coefficient of anabolism and c is the coefficient of catabolism. By substitution of (21) and (22) into (23) we begin von Bertalanffy’s historical development of the model,

$$\begin{aligned} \frac{d[qL(t)^3]}{dt} &= a[pL(t)^2] - c[qL(t)^3] \\ \frac{qd[L(t)^3]}{dt} &= apL(t)^2 - cqL(t)^3 \\ 3qL(t)^2 \frac{d[L(t)]}{dt} &= apL(t)^2 - cqL(t)^3 \\ 3q \frac{d[L(t)]}{dt} &= ap - cqL(t) \\ \frac{d[L(t)]}{dt} &= \frac{ap}{3q} - \frac{cq}{3q}L(t) \end{aligned} \quad (24)$$

The combination of the constants, a , p , and q in the final row of (24) can be replaced with constants, specifically $K = \frac{c}{3}$ and $M = \frac{ap}{cq}$ (Note, thus, that $MK = \frac{ap}{3q}$). Thus, (24) becomes

$$\frac{d[L(t)]}{dt} = MK - KL(t) \quad (25)$$

which is a linear differential equation that we can write in “standard form” as

$$\frac{d[L(t)]}{dt} + KL(t) = MK \quad (26)$$

We can now use the method for solving a linear differential equation (appendix XXX) on (26). In this case, $P'(x) = \frac{d[L(t)]}{dt}$, $f(x) = K$, $P(x) = L(t)$, and $g(x) = MK$. With these definitions $F(x) = kt$ and the integrating factor is e^{Kt} . Therefore, (26) can be rewritten as

$$\begin{aligned}
e^{Kt} \frac{d[L(t)]}{dt} + Ke^{Kt}L(t) &= MKe^{Kt} \\
\frac{d[e^{Kt}L(t)]}{dt} &= MKe^{Kt}
\end{aligned} \tag{27}$$

Following the method for solving linear differential equations we want to integrate both sides of (27) with respect to t evaluated from z_0 (which will correspond to the time when the average length was zero) to some arbitrary z ,

$$\begin{aligned}
\int_{z_0}^z \frac{d[e^{Kt}L(t)]}{dt} dt &= \int_{z_0}^z MKe^{Kt} dt \\
\int_{z_0}^z \frac{d[e^{Kt}L(t)]}{dt} dt &= M \int_{z_0}^z Ke^{Kt} dt \\
e^{Kz}L(z) - e^{Kz_0}L(z_0) &= Me^{Kz} - Me^{Kz_0} \\
e^{Kz}L(z) &= Me^{Kz} - Me^{Kz_0} \\
L(z) &= M - Me^{Kz_0}e^{-Kz} \\
L(z) &= M - Me^{-K(z-z_0)} \\
L(z) &= M(1 - e^{-K(z-z_0)})
\end{aligned} \tag{28}$$

Replacing z with t in (28) and letting $t \rightarrow \infty$ to see that $M = L_\infty$ (as we did in the first model development) it is seen that (28) is indeed equivalent to (1).

B.3 The von Bertalanffy Model for Weight

The development of the VBGM for weight is very simply a substituting of the VBGM for length (1) into the allometric length-weight relationship model,

$$\begin{aligned}
W(t) &= aL(t)^b \\
W(t) &= a \left[L_\infty \left(1 - e^{-K(t-t_0)} \right) \right]^b \\
W(t) &= aL_\infty^b \left(1 - e^{-K(t-t_0)} \right)^b \\
W(t) &= W_\infty \left(1 - e^{-K(t-t_0)} \right)^b
\end{aligned} \tag{29}$$

B.4 Relationship Between K and Growth Increments in the VBGM

From (1), define the expected length for three successive but generic ages,

$$\begin{aligned}
E[L|t = i] &= L_\infty \left(1 - e^{-K(i-t_0)} \right) \\
E[L|t = i + 1] &= L_\infty \left(1 - e^{-K(i+1-t_0)} \right) \\
E[L|t = i + 2] &= L_\infty \left(1 - e^{-K(i+2-t_0)} \right)
\end{aligned} \tag{30}$$

Further define the two annular growth increments between these three successive ages,

$$\begin{aligned}
E[L|t = i + 2] - E[L|t = i + 1] &= L_\infty \left(1 - e^{-K(i+2-t_0)}\right) - L_\infty \left(1 - e^{-K(i+1-t_0)}\right) \\
&= L_\infty \left[e^{-K(i+1-t_0)} - e^{-K(i+2-t_0)}\right] \\
&= L_\infty e^{-K(i+1-t_0)} [1 - e^{-K}] \\
&= e^{-K} L_\infty e^{-K(i-t_0)} [1 - e^{-K}]
\end{aligned} \tag{31}$$

and, with similar algebra,

$$\begin{aligned}
E[L|t = i + 1] - E[L|t = i] &= L_\infty \left(1 - e^{-K(i+1-t_0)}\right) - L_\infty \left(1 - e^{-K(i-t_0)}\right) \\
&= L_\infty e^{-K(i-t_0)} [1 - e^{-K}]
\end{aligned} \tag{32}$$

Now the right-hand-side of (32) can be substituted into the right-hand-side of (31),

$$E[L|t = i + 2] - E[L|t = i + 1] = e^{-K} (E[L|t = i + 1] - E[L|t = i]) \tag{33}$$

This proves the relation shown in (3).

B.5 Schnute Parameterization of the von Bertalanffy Model

The derivation follows that shown by [Schnute and Fournier \(1980\)](#). From (1), define the expected length at the youngest, t_1 , and oldest, t_2 , age in the data as

$$\begin{aligned}
L_1 &= E[L|t = t_1] = L_\infty \left(1 - e^{-K(t_1-t_0)}\right) \\
L_2 &= E[L|t = t_2] = L_\infty \left(1 - e^{-K(t_2-t_0)}\right)
\end{aligned} \tag{34}$$

Also recall from Appendix B.4 that,

$$E[L|t = i + 2] - E[L|t = i + 1] = e^{-K} (E[L|t = i + 1] - E[L|t = i]) \tag{35}$$

The (35) is a second order difference equation that is linear with constant coefficients and boundary conditions defined by (34). Two possible solutions to (35) are,

$$\begin{aligned}
E[L|t] &= \alpha \\
E[L|t] &= \beta e^{-Kt}
\end{aligned} \tag{36}$$

A theory of difference equations states that the solution to (35) must be a combination of these two solutions,

$$E[L|t] = \alpha + \beta e^{-Kt} \tag{37}$$

Substituting (37) into the two boundary conditions of (34) gives,

$$\begin{aligned}
L_1 &= \alpha + \beta e^{-Kt_1} \\
L_2 &= \alpha + \beta e^{-Kt_2}
\end{aligned} \tag{38}$$

which can be solved for α and β ,

$$\begin{aligned}
\alpha &= L_1 - \beta e^{-Kt_1} \\
\beta &= \frac{L_2 - L_1}{e^{-Kt_2} - e^{-Kt_1}}
\end{aligned} \tag{39}$$

Finally, these values are substituted into (37) and simplified,

$$\begin{aligned}
E[L|t] &= \alpha + \beta e^{-Kt} \\
&= L_1 - \beta e^{-Kt_1} + \beta e^{-Kt} \\
&= L_1 + \beta (e^{-Kt} - e^{-Kt_1}) \\
&= L_1 + (L_2 - L_1) \frac{e^{-Kt} - e^{-Kt_1}}{e^{-Kt_2} - e^{-Kt_1}} \\
&= L_1 + (L_2 - L_1) \frac{e^{-Kt} - e^{-Kt_1}}{e^{-Kt_2} - e^{-Kt_1}} \frac{-e^{-Kt_1}}{-e^{-Kt_1}} \\
&= L_1 + (L_2 - L_1) \frac{1 - e^{-K(t-t_1)}}{1 - e^{-K(t_2-t_1)}}
\end{aligned} \tag{40}$$

which is the Schnute parameterization of the VBGM, (7).

Reproducibility Information

Version Information

- **Compiled Date:** Mon Dec 16 2013
- **Compiled Time:** 8:22:23 PM
- **Code Execution Time:** 9.9 s

R Information

- **R Version:** R version 3.0.2 (2013-09-25)
- **System:** Windows, i386-w64-mingw32/i386 (32-bit)
- **Base Packages:** base, datasets, graphics, grDevices, methods, stats, utils
- **Other Packages:** FSA_0.4.3, FSAdata_0.1.4, gdata_2.13.2, knitr_1.5.15, nlstools_0.0-15, xtable_1.7-1
- **Loaded-Only Packages:** bitops_1.0-6, car_2.0-19, caTools_1.16, cluster_1.14.4, evaluate_0.5.1, formatR_0.10, Formula_1.1-1, gplots_2.12.1, grid_3.0.2, gtools_3.1.1, highr_0.3, Hmisc_3.13-0, KernSmooth_2.23-10, lattice_0.20-24, MASS_7.3-29, multcomp_1.3-1, mvtnorm_0.9-9996, nlme_3.1-113, nnet_7.3-7, plotrix_3.5-2, quantreg_5.05, sandwich_2.3-0, sciplot_1.1-0, SparseM_1.03, splines_3.0.2, stringr_0.6.2, survival_2.37-4, tools_3.0.2, zoo_1.7-10
- **Required Packages:** FSA, FSAdata, nlstools and their dependencies (car, gdata, gplots, Hmisc, knitr, multcomp, nlme, plotrix, quantreg, sciplot, stats)