

Fitting a von Bertalanffy Growth Function

Derek H. Ogle, Northland College

16-Aug-2015

Preliminaries

```
> library(FSA)                # for headtail(), filterD(), vbModels(), vbStart(), vbFuns()
> library(FSAdata)            # for TroutBR data
> library(nlstools)           # for nlsBoot()
```

Loading the Data and Some Preparations

```
> data(TroutBR)
> rbt <- filterD(TroutBR,species=="Rainbow")
> str(rbt)
'data.frame':   627 obs. of  3 variables:
 $ t1      : int   12 14 14 14 14 15 15 15 15 15 ...
 $ age     : int    3 3 3 3 4 3 3 3 3 3 ...
 $ species: Factor w/ 1 level "Rainbow": 1 1 1 1 1 1 1 1 1 1 ...

> xlbl <- "Age (yrs)"
> ylbl <- "Total Length (in)"
> clr <- rgb(0,0,0,0.05)
```

What Parameterizations are Available in FSA?

```
> vbModels()
```

FSA von Bertalanffy Parameterizations

Typical: $E(L_t) = L_\infty \left(1 - e^{-K(t-t_0)}\right)$

Original: $E(L_t) = L_\infty - (L_\infty - L_0) e^{-Kt}$

GQ: $E(L_t) = \frac{\omega}{K} \left(1 - e^{-K(t-t_0)}\right)$

Mooij: $E(L_t) = L_\infty - (L_\infty - L_0) e^{-\frac{\omega}{L_\infty} t}$

Weisberg: $E(L_t) = L_\infty \left(1 - e^{-\frac{\log(2)}{(t_{50}-t_0)}(t-t_0)}\right)$

Schnute: $E(L_t) = L_1 + (L_3 - L_1) \frac{1 - e^{-K(t-t_1)}}{1 - e^{-K(t_3-t_1)}}$

Francis: $E(L_t) = L_1 + (L_3 - L_1) \frac{1 - r^{\frac{t-t_1}{t_3-t_1}}}{1 - r^2}$

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

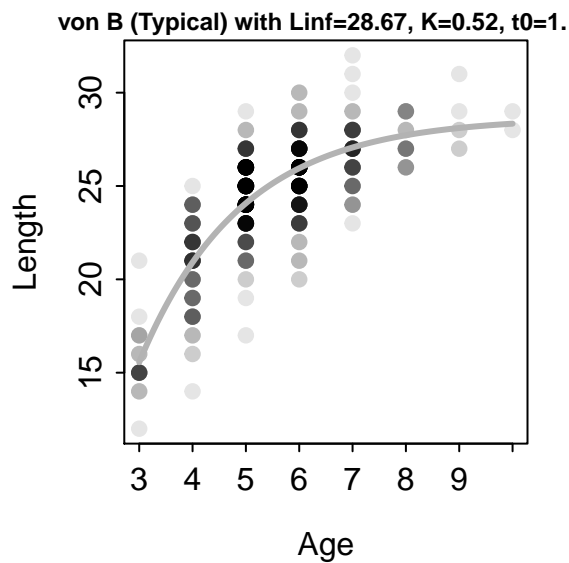
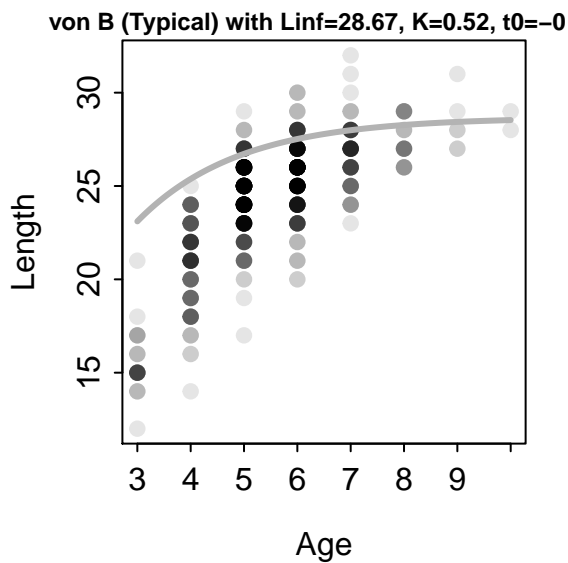
Fit Typical VBGF

Declare a Function

```
> vb <- vbFuns("Typical")
> vb
function(t,Linf,K=NULL,t0=NULL) {
  if (length(Linf)==3) { K <- Linf[[2]]
                        t0 <- Linf[[3]]
                        Linf <- Linf[[1]] }
  Linf*(1-exp(-K*(t-t0)))
}
<environment: 0x03ba9d64>
```

Find Starting Values

```
> svb.bad <- vbStarts(tl~age,data=rbt,type="Typical",plot=TRUE) # Left
> svb <- vbStarts(tl~age,data=rbt,type="Typical",meth0="yngAge",plot=TRUE) # Right
> unlist(svb) # unlist() only to save space
      Linf      K      t0
28.6732137 0.5242049 1.5075626
```



```
> # Dynamically approximately fit the function -- Can't be shown in a handout
> vbStarts(tl~age,data=rbt,type="typical",dynamicPlot=TRUE)
> svb2 <- list(Linf=28.7,K=0.52,t0=1.62)
```

Fit the Model

```
> fit1 <- nls(tl~vb(age,Linf,K,t0),data=rbt,start=svb)
> summary(fit1,correlation=TRUE)
```

Formula: $tl \sim vb(\text{age}, Linf, K, t0)$

Parameters:

	Estimate	Std. Error	t value	Pr(> t)
Linf	27.71191	0.28383	97.64	<2e-16
K	0.63242	0.04248	14.89	<2e-16
t0	1.71686	0.10159	16.90	<2e-16

Residual standard error: 1.775 on 624 degrees of freedom

Correlation of Parameter Estimates:

	Linf	K
K	-0.91	
t0	-0.71	0.92

Number of iterations to convergence: 3

Achieved convergence tolerance: 9.636e-06

```
> ( cf <- coef(fit1) )
```

	Linf	K	t0
	27.7119085	0.6324231	1.7168636

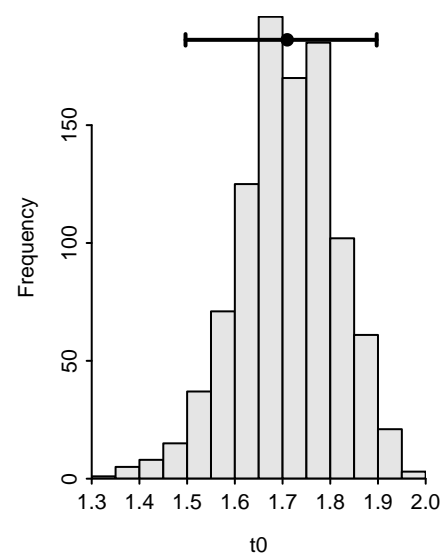
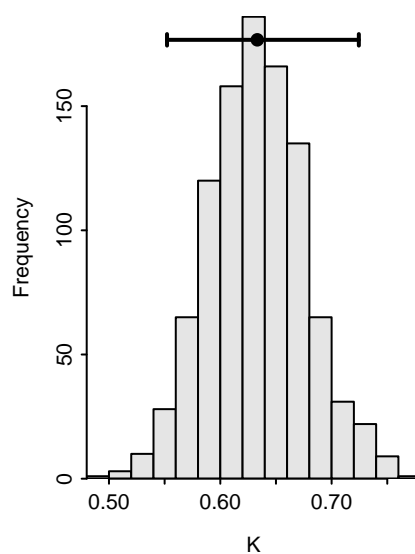
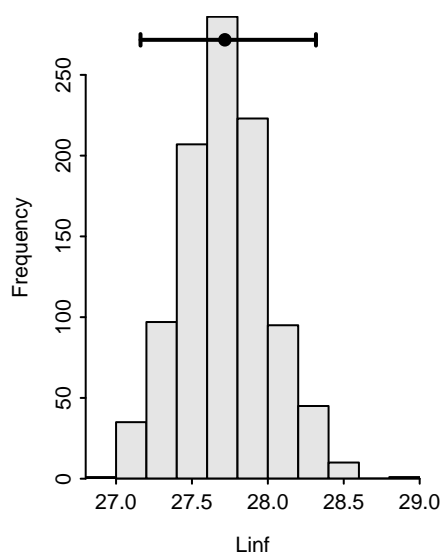
```
> confint(fit1)
```

Waiting for profiling to be done...

	2.5%	97.5%
Linf	27.1916077	28.3279785
K	0.5499956	0.7192266
t0	1.4930214	1.8999245

```
> boot1 <- nlsBoot(fit1,niter=1000)
```

```
> confint(boot1,plot=TRUE,rows=1,cols=3)
```



	95% LCI	95% UCI
Linf	27.1610340	28.3168119
K	0.5521347	0.7244517
t0	1.4966400	1.8973160

Make Predictions

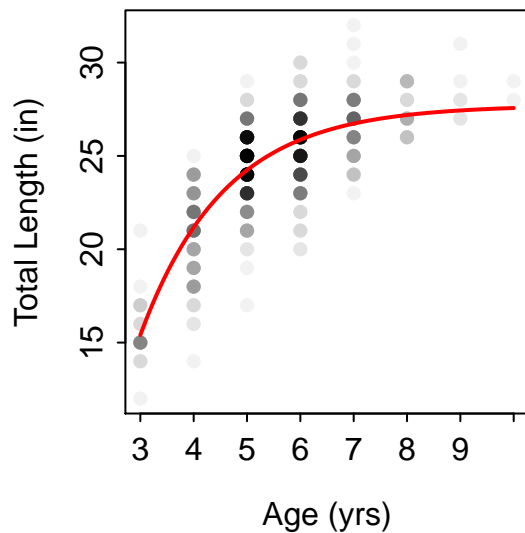
```
> ageX <- 8
> predict(fit1,data.frame(age=ageX))
[1] 27.19077

> headtail(boot1$coefboot)
      Linf      K      t0
[1,] 27.99618 0.5967413 1.590801
[2,] 27.49018 0.6299656 1.658670
[3,] 27.76405 0.6240239 1.682046
[998,] 27.43896 0.6521031 1.746466
[999,] 27.76504 0.6158193 1.662552
[1000,] 28.07417 0.6026139 1.684777

> pv <- apply(boot1$coefboot,MARGIN=1,FUN=vb,t=ageX)
> quantile(pv,c(0.025,0.975))
      2.5%      97.5%
26.82817 27.54216
```

Visualize the Fit

```
> plot(tl~age,data=rbt,xlab=xlbl,ylab=ylbl,pch=16,col=clr)
> curve(vb(x,cf),from=3,to=10,n=500,lwd=2,col="red",add=TRUE)
```



```
> residPlot(fit1)
```

