Fitting a von Bertalanffy Growth Function

Preliminaries

```
> library(FSAdata)            # for TroutBR data
> library(FSA)                # for filterD(), headtail(), col2rgbt(), vbFuns(), vbStart()
> library(nlstools)           # for nlsBoot()
```

Loading the Data and Some Preparations

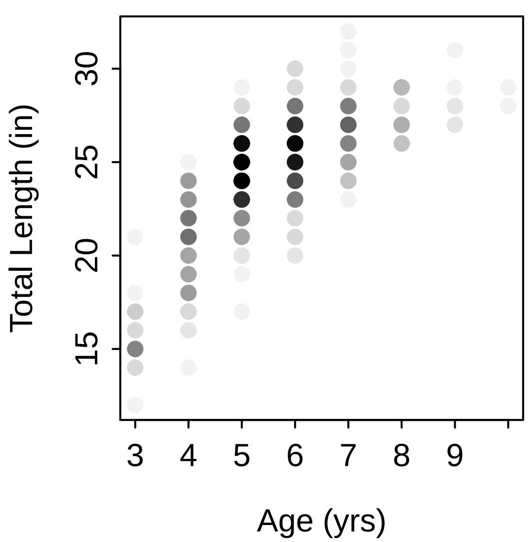
```
> data(TroutBR)
> str(TroutBR)
'data.frame':   851 obs. of  3 variables:
 $ tl     : int  16 16 17 17 17 17 17 17 17 17 ...
 $ age    : int  4 4 2 3 3 3 3 3 3 4 ...
 $ species: Factor w/ 2 levels "Brown","Rainbow": 1 1 1 1 1 1 1 1 1 1 ...
```

```
> rbt <- filterD(TroutBR,species=="Rainbow")
> headtail(rbt)
    tl age species
1   12   3 Rainbow
2   14   3 Rainbow
3   14   3 Rainbow
625 31   7 Rainbow
626 31   9 Rainbow
627 32   7 Rainbow
```

```
> xlbl <- "Age (yrs)"
> ylbl <- "Total Length (in)"
> clr <- col2rgbt("black",1/20)
```

Examine Plot of Data

```
> plot(tl~age,data=rbt,pch=19,col=clr,xlab=xlbl,ylab=ylbl)
```



Fit Typical VBGF

Declare a Function

```
> vb <- vbFuns("Typical",msg=TRUE)
You have chosen the 'Typical', 'Traditional', or 'BevertonHolt' parameterization.

  E[L|t] = Linf*(1-exp(-K*(t-t0)))

  where Linf = asymptotic mean length
           K = exponential rate of approach to Linf
          t0 = the theoretical age when length = 0 (a modeling artifact)
```

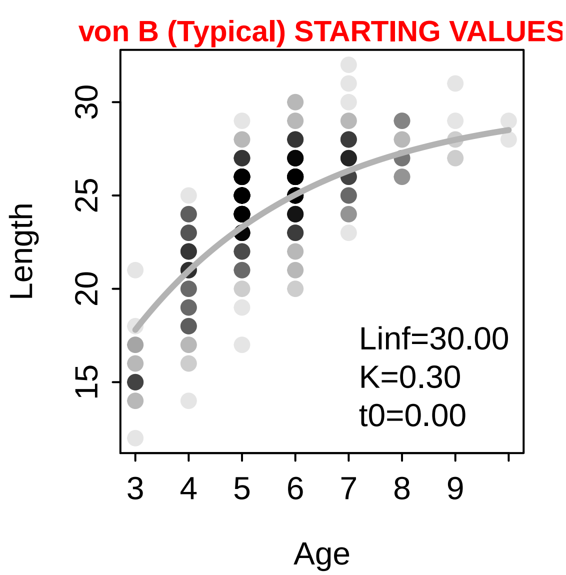
```
> vb
function (t, Linf, K = NULL, t0 = NULL)
{
    if (length(Linf) == 3) {
        K <- Linf[[2]]
        t0 <- Linf[[3]]
        Linf <- Linf[[1]]
    }
    Linf * (1 - exp(-K * (t - t0)))
}
<environment: 0x05f66c88>
```

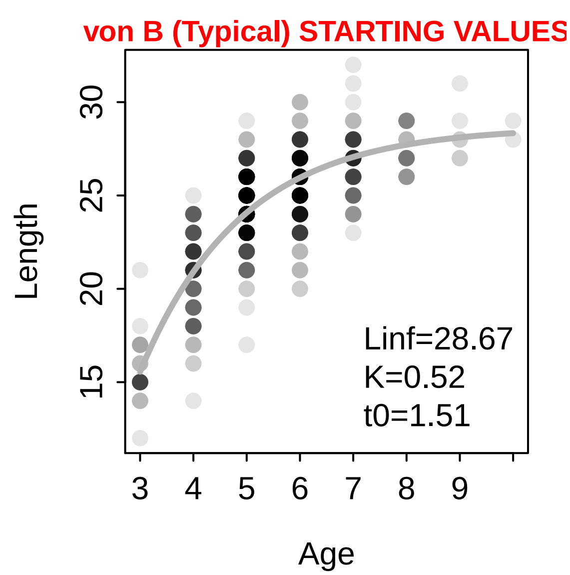
```
> ## Next two simply demonstrate how this function works
> vb(8,Linf=300,K=0.3,t0=-1)
[1] 279.8383
```

```
> vb(1:8,c(300,0.3,-1))
[1] 135.3565 178.0291 209.6417 233.0610 250.4103 263.2631 272.7846 279.8383
```

Find Starting Values

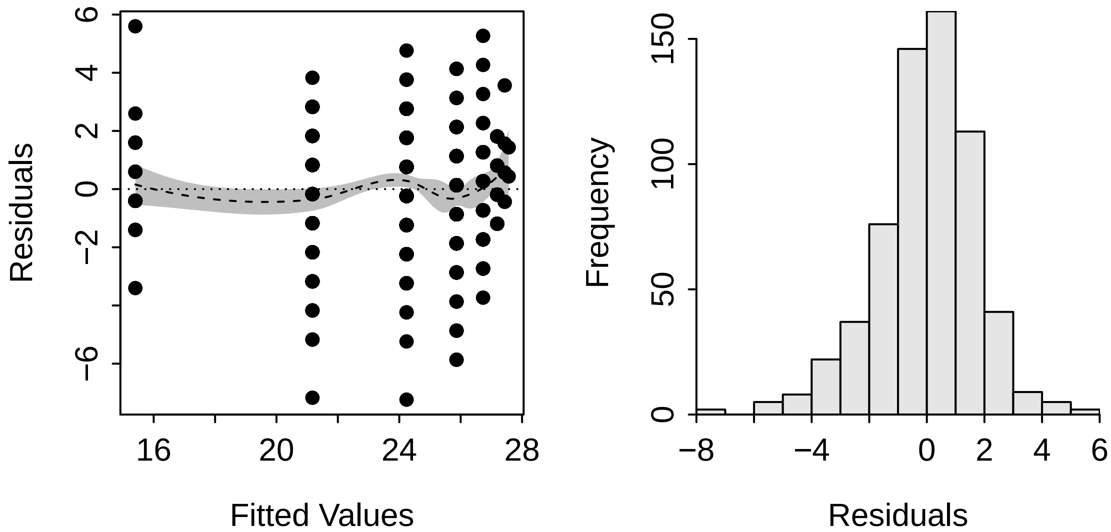
```
> # Demos manual generation with plot ... LEFT plot
> svb <- vbStarts(tl~age,data=rbt,type="Typical",plot=TRUE,
                  fixed=list(Linf=30,K=0.3,t0=0))
> # Demos automatic generation ... RIGHT plot
> svb <- vbStarts(tl~age,data=rbt,type="Typical",plot=TRUE)
```





Check Assumptions

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



Summarize the Model Fit

```
> summary(fit1,correlation=TRUE)

Formula: tl ~ vb(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: 9.57e-06
```

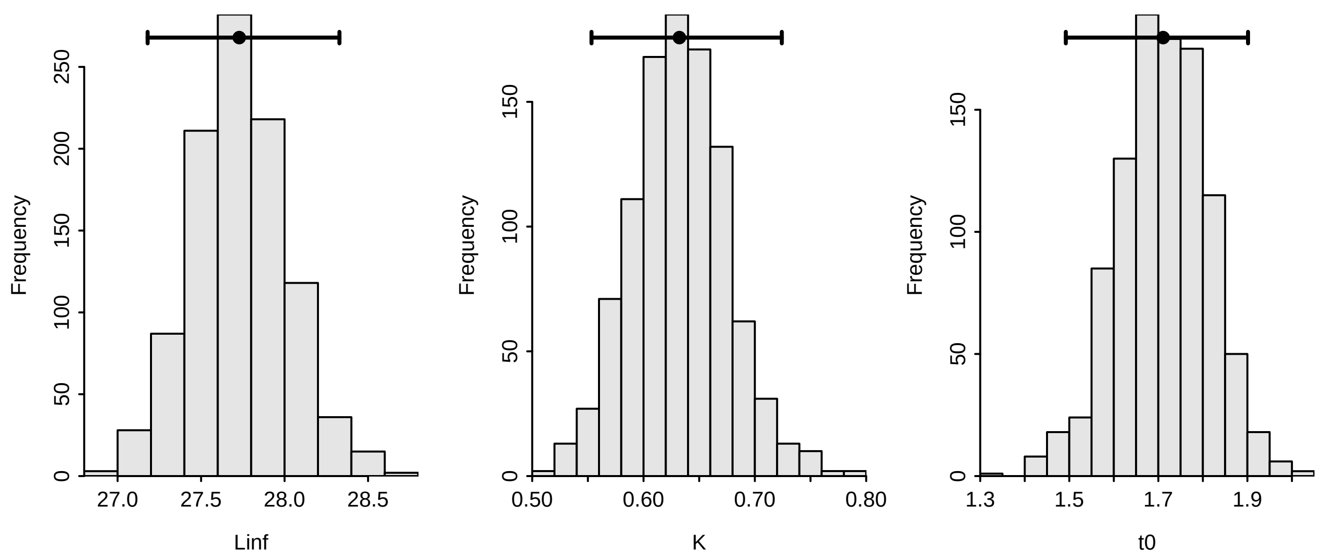
```
> ( cf <- coef(fit1) )
      Linf          K         t0
27.7119083  0.6324231  1.7168636
```

```
> confint(fit1)
           2.5%      97.5%
Linf 27.1916077 28.3279785
K     0.5499956  0.7192266
t0    1.4930214  1.8999245
```

```
> boot1 <- nlsBoot(fit1,niter=1000)
> str(boot1)
List of 4
 $ coefboot: num [1:1000, 1:3] 27.7 27.8 27.7 28.2 27.7 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : chr [1:3] "Linf" "K" "t0"
 $ rse     : num [1:1000] 1.7 1.8 1.88 1.73 1.78 ...
 $ bootCI  : num [1:3, 1:3] 27.712 0.631 1.71 27.18 0.553 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:3] "Linf" "K" "t0"
  .. ..$ : chr [1:3] "Median" "2.5%" "97.5%"
 $ estiboot: num [1:3, 1:2] 27.7285 0.6322 1.7108 0.2877 0.0428 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:3] "Linf" "K" "t0"
  .. ..$ : chr [1:2] "Estimate" "Std. error"
 - attr(*, "class")= chr "nlsBoot"
```

```
> headtail(boot1$coefboot)
            Linf         K       t0
[1,]     27.70937 0.6220906 1.690328
[2,]     27.78841 0.6192417 1.726996
[3,]     27.66438 0.6280694 1.709791
 [998,] 27.61946 0.6498783 1.807286
 [999,] 27.54571 0.6303501 1.717794
[1000,] 27.68641 0.6512426 1.731938
```

```
> confint(boot1,plot=TRUE,rows=1,cols=3)
        95% LCI    95% UCI
Linf 27.1798943 28.3284789
K     0.5532742  0.7242048
t0    1.4922470  1.9014481
```



Make Predictions

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

```
> pv <- apply(boot1$coefboot,MARGIN=1,FUN=vb,t=ageX)
> quantile(pv,c(0.025,0.975))
    2.5%    97.5%
26.83587 27.54593
```

Visualize the Fit

```
> plot(tl~age,data=rbt,xlab=xlbl,ylab=ylbl,pch=19,col=clr)
> curve(vb(x,cf),from=3,to=10,n=500,lwd=2,col="red",add=TRUE)
```