

fishR Vignette - Back-Calculation, Vigliola and Meekan's Formulations

Dr. Derek Ogle, Northland College

December 16, 2013

This vignette attempts to fit the models and apply the back-calculation models that are detailed in Appendix 1 of [Vigliola and Meekan \(2009\)](#). As with the main back-calculation vignette, this vignette requires functions in the FSA package maintained by the author and the `lattice` and `doBy` packages. These packages are loaded into R with

```
> library(FSA)
> library(lattice) # for pairs
> library(doBy)   # for summaryBy
```

All analyses in this document use the West Bearskin Lake smallmouth bass (*Micropterus dolomieu*) data set used in [Weisberg \(1993\)](#). The West Bearskin Lake smallmouth bass data are found in the `SMBassWB` data file of the `FSAdata` package and can be read into R with,

```
> data(SMBassWB)
```

For the purposes of this example the analysis will focus on only those fish captured in 1990 as found from the year-at-capture and age-at-capture with,

```
> SMBassWByearclass <- SMBassWB$yearcap-SMBassWB$agecap
> wb90 <- Subset(SMBassWB,yearcap==1990)
```

1 Fitting the Various Linear Models

```
> wb90$logR <- log(wb90$radcap)
> wb90$logL <- log(wb90$lencap)
```

```
> # 0 forces not fitting intercept
> f1 <- g1 <- lm(lencap~0+radcap,data=wb90)
> f2 <- lm(radcap~lencap,data=wb90)
> f3 <- lm(radcap~lencap+agecap,data=wb90)
> f4 <- lm(logR~logL,data=wb90)
> ## next model does not converge with these data
> f4n <- nls(radcap~B*radcap^C,data=wb90,start=list(B=exp(coef(f4)[1]),C=coef(f4)[2]),
  control=list(maxiter=500,warnOnly=TRUE))
```

Warning: number of iterations exceeded maximum of 500

```
> ##
> f5 <- nls(radcap~((lencap-a)/b)^(1/c),data=wb90,start=list(a=0,b=exp(coef(f4)[1])/
  coef(f4)[2]),c=1/coef(f4)[2])
> f6 <- lm(radcap~poly(lencap,2,row=TRUE),data=wb90)
> # f7 is demonstrated with a cubic polynomial
> f7 <- lm(radcap~poly(lencap,3,row=TRUE),data=wb90)
> f8 <- lm(radcap~logL,data=wb90)
>
> g2 <- lm(lencap~radcap,data=wb90)
```

```

> g3 <- lm(lencap~radcap+agecap,data=wb90)
> g4 <- lm(logL~logR,data=wb90)
> g4n <- nls(lencap~b*radcap^c,data=wb90,start=list(b=exp(coef(g4)[1]),c=coef(g4)[2]))
> g5 <- nls(lencap~a+b*radcap^c,data=wb90,start=list(a=0,b=exp(coef(g4)[1]),c=coef(g4)[2]))
> g6 <- lm(lencap~poly(radcap,2,raw=TRUE),data=wb90)
> # g7 is demonstrated with a cubic polynomial
> g7 <- lm(lencap~poly(radcap,3,raw=TRUE),data=wb90)
> g8lin <- lm(logL~radcap,data=wb90)
> g8 <- nls(lencap~exp(a+b*radcap),data=wb90,start=list(a=coef(g8lin)[1],b=coef(g8lin)[2]))

```

2 Performing the Back-Calculations

Must reshape first ...

```

> wb90r <- gReshape(wb90,in.pre="anu")
> head(wb90r)

```

	species	lake	gear	yearcap	fish	agecap	lencap	radcap	logR	logL	prvAge	anu
1	SMB	WB	E	1990	482	1	75	1.511	0.4126	4.317	1	1.511
2	SMB	WB	E	1990	768	1	75	1.580	0.4574	4.317	1	1.580
3	SMB	WB	E	1990	428	1	71	1.487	0.3969	4.263	1	1.487
4	SMB	WB	E	1990	478	2	116	2.736	1.0065	4.754	1	1.606
5	SMB	WB	E	1990	379	2	117	3.055	1.1168	4.762	1	1.596
6	SMB	WB	E	1990	477	2	111	2.449	0.8957	4.710	1	1.321

```

> ##
> ## Back-calculating with BCM1 (Dahl-Lea)
> ( BCM1 <- bcFuns(1) )
function(Lc,Ri,Rc) { (Ri/Rc)*Lc }
<environment: 0x083105e0>
> wb90r$bcm1 <- with(wb90r,BCM1(lencap,anu,radcap))
>
> ##
> ## Back-calculating with BCM2 (Fraser-Lee)
> ( BCM2 <- bcFuns(2) )
function(Lc,Ri,Rc,a) { a+(Lc-a)*(Ri/Rc) }
<environment: 0x083a3b5c>
> ( a <- coef(g2)[1] )
(Intercept)
41.65
> wb90r$bcm2 <- with(wb90r,BCM2(lencap,anu,radcap,a))
>
> ##
> ## Back-calculating with BCM3
> ( BCM3 <- bcFuns(3) )
function(Lc,Ri,Rc,L0p,R0p) { Lc+(Ri-Rc)*(Lc-L0p)/(Rc-R0p) }
<environment: 0x08478c2c>
> # Set biological intercept values
> L0p <- 20 # just made this up
> R0p <- 0.2 # this too

```

```

> wb90r$bcm3 <- with(wb90r,BCM3(lencap,anu,radcap,L0p,R0p))
>
> ##
> ## Back-calculating with BCM4
> ( BCM4 <- bcFuns(4) )
function(Lc,Ri,Rc,a,b) { (a+b*Ri)*Lc/(a+b*Rc) }
<environment: 0x0857f954>
> ( a <- coef(g2)[1] )
(Intercept)
  41.65
> ( b <- coef(g2)[2] )
radcap
  27.36
> wb90r$bcm4 <- with(wb90r,BCM4(lencap,anu,radcap,a,b))
>
> ##
> ## Back-calculating with BCM6
> ( BCM6 <- bcFuns(6) )
function(Lc,Ri,Rc,A,B) { (Ri/Rc*(A+B*Lc)-A)/B }
<environment: 0x0726eac0>
> ( A <- coef(f2)[1] )
(Intercept)
  -1.304
> ( B <- coef(f2)[2] )
lencap
  0.03537
> wb90r$bcm6 <- with(wb90r,BCM6(lencap,anu,radcap,A,B))
>
> ##
> ## Back-calculating with BCM7
> ( BCM7 <- bcFuns(7) )
function(Lc,Ri,Rc,agei,agec,a,b,c) { -a/b+(Lc+a/b+c/b*agec)*Ri/Rc-c/b*agei }
<environment: 0x07aee5a8>
> ( a <- coef(f3)[1] )
(Intercept)
  -1.331
> ( b <- coef(f3)[2] )
lencap
  0.03596
> ( c <- coef(f3)[3] )
agecap
 -0.01924
> wb90r$bcm7 <- with(wb90r,BCM7(lencap,anu,radcap,prvAge,agecap,a,b,c))
>
> ##
> ## Back-calculating with BCM8
> ( BCM8 <- bcFuns(8) )
function(Lc,Ri,Rc,agei,agec,A,B,C) { (A+B*Ri+C*agei)/(A+B*Rc+C*agec)*Lc }
<environment: 0x07bd4e60>

```

```

> ( A <- coef(g3)[1] )
(Intercept)
  42.56
> ( B <- coef(g3)[2] )
radcap
  20.15
> ( C <- coef(g3)[3] )
agecap
  8.688
> wb90r$bcm8 <- with(wb90r,BCM8(lencap, anu, radcap, prvAge, agecap, A, B, C))
>
> ##
> ## Back-calculating with BCM9
> ( BCM9 <- bcFuns(9) )
function(Lc,Ri,Rc,c) { Lc*((Ri/Rc)^c) }
<environment: 0x0794d558>
> ( c <- coef(g4)[2] )
  logR
  0.7672
> wb90r$bcm9 <- with(wb90r,BCM9(lencap, anu, radcap, c))
>
> ##
> ## Back-calculating with BCM10
> ( BCM10 <- bcFuns(10) )
function(Lc,Ri,Rc,c) { Lc*((Ri/Rc)^c) }
<environment: 0x078ce110>
> ( c <- coef(g4n)[2] )
  c
  0.7745
> wb90r$bcm10 <- with(wb90r,BCM10(lencap, anu, radcap, c))
>
> ##
> ## Back-calculating with BCM11
> ( BCM11 <- bcFuns(11) )
function(Lc,Ri,Rc,C) { Lc*((Ri/Rc)^(1/C)) }
<environment: 0x07594340>
> # would use f4n here but it did not converge for these data
> ( C <- coef(f4)[2] )
  logL
  1.263
> wb90r$bcm11 <- with(wb90r,BCM11(lencap, anu, radcap, C))
>
> ##
> ## Back-calculating with BCM13
> ( BCM13 <- bcFuns(13) )
function(Lc,Ri,Rc,L0,R0,a) { a + exp(log(L0-a) + ((log(Lc-a)-log(L0-a))*(log(Ri)-log(R0)))/(log(Rc)-
<environment: 0x0734c91c>
> ( a <- coef(g5)[1] )
  a
  12.56

```

```

> # arbitrarily choose R0 to be smallest (anu)lus measurement
> ( R0 <- min(wb90r$anu) )
[1] 0.8422
> # L0 is point on g5 line corresponding to R0
> ( L0 <- predict(g5,data.frame(radcap=R0)) )
[1] 50.88
> wb90r$bcm13 <- with(wb90r,BCM13(lencap,anu,radcap,L0,R0,a))
>
> ##
> ## Back-calculating with BCM14
> ( BCM14 <- bcFuns(14) )
function(Lc,Ri,Rc,L0p,R0p,a) { a + exp(log(L0p-a) + ((log(Lc-a)-log(L0p-a))*(log(Ri)-log(R0p)))/(log
<environment: 0x06f6ee3c>
> # Set biological intercept values (both are made up for this example)
> L0p <- 10
> R0p <- 0.2
>
> # Perform VEM g5 and f5 regressions via nls but with biological intercepts
> g5m <- nls(lencap~L0p-b*R0p^c+b*radcap^c,data=wb90,
             start=list(b=exp(coef(g4)[1]),c=coef(g4)[2]))
> f5m <- nls(radcap~((lencap-L0p-b*R0p^c)/b)^(1/c),data=wb90,
             start=list(b=exp(coef(f4)[1]/coef(f4)[2]),c=1/coef(f4)[2]))
>
> # extract parameter values from the L-R and R-L regressions in order to compute a
> b1 <- coef(g5m)[1]; c1 <- coef(g5m)[2]
> a1 <- L0p-b1*R0p^c1
> b2 <- coef(f5m)[1]; c2 <- coef(f5m)[2]
> a2 <- L0p-b2*R0p^c2
> # used as.numeric() to strip attribute which was labeled as b
> ( a <- as.numeric((a1+a2)/2) )
[1] -3.067
>
> wb90r$bcm14 <- with(wb90r,BCM14(lencap,anu,radcap,L0p,R0p,a))
>
> ##
> ## Back-calculating with BCM15
> ( BCM15 <- bcFuns(15) )
function(Lc,Ri,Rc,a,b,c) { (a+b*Ri^c)/(a+b*Rc^c)*Lc }
<environment: 0x07b05670>
> ( a <- coef(g5)[1] )
      a
12.56
> ( b <- coef(g5)[2] )
      b
44.19
> ( c <- coef(g5)[3] )
      c
0.8298
> wb90r$bcm15 <- with(wb90r,BCM15(lencap,anu,radcap,a,b,c))
>
> ##
> ## Back-calculating with BCM16
> ( BCM16 <- bcFuns(16) )

```

```

function(Lc,Ri,Rc,a,b,c) { a+(Lc-a)*((Ri/Rc)^c) }
<environment: 0x07a48f28>
> ( a <- coef(f5)[1] )
      a
19.49
> ( b <- coef(f5)[2] )
      b
38.03
> ( c <- coef(f5)[3] )
      c
0.8931
> wb90r$bcm16 <- with(wb90r,BCM16(lencap,anu,radcap,a,b,c))
>
> ##
> ## Back-calculating with BCM17
> ( BCM17 <- bcFuns(17) )
function(Lc,Ri,Rc,a,b,c) { (a+b*Ri+c*(Ri^2))/(a+b*Rc+c*(Rc^2))*Lc }
<environment: 0x076fed2c>
> ( a <- coef(g6)[1] )
(Intercept)
      27.9
> ( b <- coef(g6)[2] )
poly(radcap, 2, raw = TRUE)1
      32.88
> ( c <- coef(g6)[3] )
poly(radcap, 2, raw = TRUE)2
      -0.4649
> wb90r$bcm17 <- with(wb90r,BCM17(lencap,anu,radcap,a,b,c))
>
> ##
> ## Back-calculating with BCM18
> ( BCM18 <- bcFuns(18) )
function(Lc,Ri,Rc,a,b,c) {
  cf1 <- b
  cf2 <- c
  Li <- numeric(length(Lc))
  for (i in 1:length(Li)) {
    cf0 <- a-((Ri[i]/Rc[i])*(a+b*Lc[i]+c*Lc[i]^2))
    roots <- Re(polyroot(c(cf0,cf1,cf2)))
    Li[i] <- roots[which(sign(roots)==1)]
  }
  Li
}
<environment: 0x07430e8c>
> ( a <- coef(f6)[1] )
(Intercept)
      -0.8778
> ( b <- cf1 <- coef(f6)[2] )
poly(lencap, 2, raw = TRUE)1
      0.03066

```

```

> ( c <- cf2 <- coef(f6)[3] )
poly(lencap, 2, raw = TRUE)2
      1.167e-05
> wb90r$bcm18 <- with(wb90r,BCM18(lencap,anu,radcap,a,b,c))
>
> ##
> ## Back-calculating with BCM19
> ( BCM19 <- bcFuns(19) )
function(Lc,Ri,Rc,a) { # a must be a vector of coefficients from the polynomial regression
  exps <- 0:(length(a)-1)
  Li <- numeric(length(Lc))
  for (i in 1:length(Lc)) {
    num <- sum( a*Ri[i]^exps )
    denom <- sum( a*Rc[i]^exps )
    Li[i] <- num/denom*Lc[i]
  }
  Li
}
<environment: 0x07d73b08>
> ( a <- coef(g7) )
      (Intercept) poly(radcap, 3, raw = TRUE)1
              30.84098                    31.06255
poly(radcap, 3, raw = TRUE)2 poly(radcap, 3, raw = TRUE)3
      -0.13044                    -0.01871
> wb90r$bcm19 <- with(wb90r,BCM19(lencap,anu,radcap,a))
>
> ##
> ## Back-calculating with BCM20
> ( BCM20 <- bcFuns(20) )
function(Lc,Ri,Rc,a) { # a must be a vector of coefficients from the polynomial regression
  exps <- 0:(length(a)-1)
  Li <- numeric(length(Lc))
  for (i in 1:length(Li)) {
    if (Ri[i]==Rc[i]) { Li[i] <- Lc[i] }
    else {
      cf <- a
      cf[1] <- cf[1] - Ri[i]/Rc[i]*sum(a*Lc[i]^exps)
      roots <- Re(polyroot(cf))
      roots <- roots[which(sign(roots)==1)] # find only positive roots
      roots <- roots[which(roots<=Lc[i])] # only find root less than lencap
      ifelse(length(roots)!=1,Li[i] <- NA,Li[i] <- roots)
    }
  }
  Li
}
<environment: 0x077aee7c>
> ( a <- coef(f7) )
      (Intercept) poly(lencap, 3, raw = TRUE)1
              1.790e-02                    1.535e-02
poly(lencap, 3, raw = TRUE)2 poly(lencap, 3, raw = TRUE)3
      9.316e-05                    -1.357e-07
> wb90r$bcm20 <- with(wb90r,BCM20(lencap,anu,radcap,a))
>

```

```

> ##
> ## Back-calculating with BCM21
> ( BCM21 <- bcFuns(21) )

function(Lc,Ri,Rc,a,b) { exp(a+b*Ri)/exp(a+b*Rc)*Lc }
<environment: 0x07037c94>

> ( a <- coef(g8)[1] )

      a
4.485

> ( b <- coef(g8)[2] )

      b
0.1317

> wb90r$bcm21 <- with(wb90r,BCM21(lencap,anu,radcap,a,b))
>
> ##
> ## Back-calculating with BCM22
> ( BCM22 <- bcFuns(22) )

function(Lc,Ri,Rc,a) { exp(a+(log(Lc)-a)*Ri/Rc) }
<environment: 0x06e15c68>

> ( a <- -coef(f8)[1]/coef(f8)[2] )

(Intercept)
      4.353

> wb90r$bcm22 <- with(wb90r,BCM22(lencap,anu,radcap,a))

```

3 Some Comparisons

```

> means <- summaryBy(bcm1+bcm2+bcm3+bcm4+bcm6+bcm7+bcm8+bcm9+bcm10+
                    bcm11+bcm13+bcm14+bcm15+bcm16+bcm17+bcm18+bcm19+bcm20+
                    bcm21+bcm22~prvAge,data=wb90r,FUN=mean,na.rm=TRUE,keep.names=TRUE)
> round(means,0)

```

prvAge	bcm1	bcm2	bcm3	bcm4	bcm6	bcm7	bcm8	bcm9	bcm10	bcm11	bcm13	bcm14	bcm15	bcm16	
1	1	50	79	59	79	75	75	79	66	66	64	69	60	69	70
2	2	97	114	102	114	112	112	114	111	111	110	112	107	112	111
3	3	137	147	140	147	146	146	146	147	147	146	147	144	147	146
4	4	157	173	162	173	171	171	173	175	175	173	174	170	174	172
5	5	190	201	193	201	200	200	202	204	204	203	203	200	203	201
6	6	231	235	232	235	235	235	235	237	237	237	237	236	237	236
7	7	268	269	268	269	269	269	269	269	269	269	269	269	269	269
8	8	278	283	280	283	283	283	284	286	286	286	285	284	285	284
9	9	314	314	314	314	314	314	314	314	314	314	315	314	314	314
bcm17	bcm18	bcm19	bcm20	bcm21	bcm22										
1	72	71	73	64	105	96									
2	112	111	112	110	126	118									
3	147	146	147	146	149	144									
4	174	172	174	172	171	162									
5	203	201	203	201	195	187									
6	237	236	237	235	230	226									
7	269	269	269	269	268	268									
8	286	284	287	282	271	268									
9	314	314	314	314	314	314									


```

> ( ns <- summaryBy(bcm1+bcm2+bcm3+bcm4+bcm6+bcm7+bcm8+bcm9+bcm10+
                    bcm11+bcm13+bcm14+bcm15+bcm16+bcm17+bcm18+bcm19+bcm20+
                    bcm21+bcm22~prvAge,data=wb90r,FUN=length,keep.names=TRUE) )
  prvAge bcm1 bcm2 bcm3 bcm4 bcm6 bcm7 bcm8 bcm9 bcm10 bcm11 bcm13 bcm14 bcm15 bcm16
1      1   181  181  181  181  181  181  181  181  181  181  181  181  181  181
2      2   178  178  178  178  178  178  178  178  178  178  178  178  178  178
3      3   155  155  155  155  155  155  155  155  155  155  155  155  155  155
4      4    71   71   71   71   71   71   71   71   71   71   71   71   71   71
5      5    64   64   64   64   64   64   64   64   64   64   64   64   64   64
6      6    64   64   64   64   64   64   64   64   64   64   64   64   64   64
7      7    50   50   50   50   50   50   50   50   50   50   50   50   50   50
8      8     2    2    2    2    2    2    2    2    2    2    2    2    2    2
9      9     2    2    2    2    2    2    2    2    2    2    2    2    2    2
  bcm17 bcm18 bcm19 bcm20 bcm21 bcm22
1    181  181  181  181  181  181
2    178  178  178  178  178  178
3    155  155  155  155  155  155
4     71   71   71   71   71   71
5     64   64   64   64   64   64
6     64   64   64   64   64   64
7     50   50   50   50   50   50
8      2    2    2    2    2    2
9      2    2    2    2    2    2

```

Suppose we want to compare the mean length-at-age computed with the BCM4, BCM6, BCM7, BCM8, BCM14, BCM19, and BCM20 models

```

  bcm4 bcm6 bcm7 bcm8 bcm14 bcm19 bcm20
1    79  75  75  79    60   73   64
2   114 112 112 114   107  112  110
3   147 146 146 146   144  147  146
4   173 171 171 173   170  174  172
5   201 200 200 202   200  203  201
6   235 235 235 235   236  237  235
7   269 269 269 269   269  269  269
8   283 283 283 284   284  287  282
9   314 314 314 314   314  314  314

```

... against each other with a red line showing perfect agreement ...

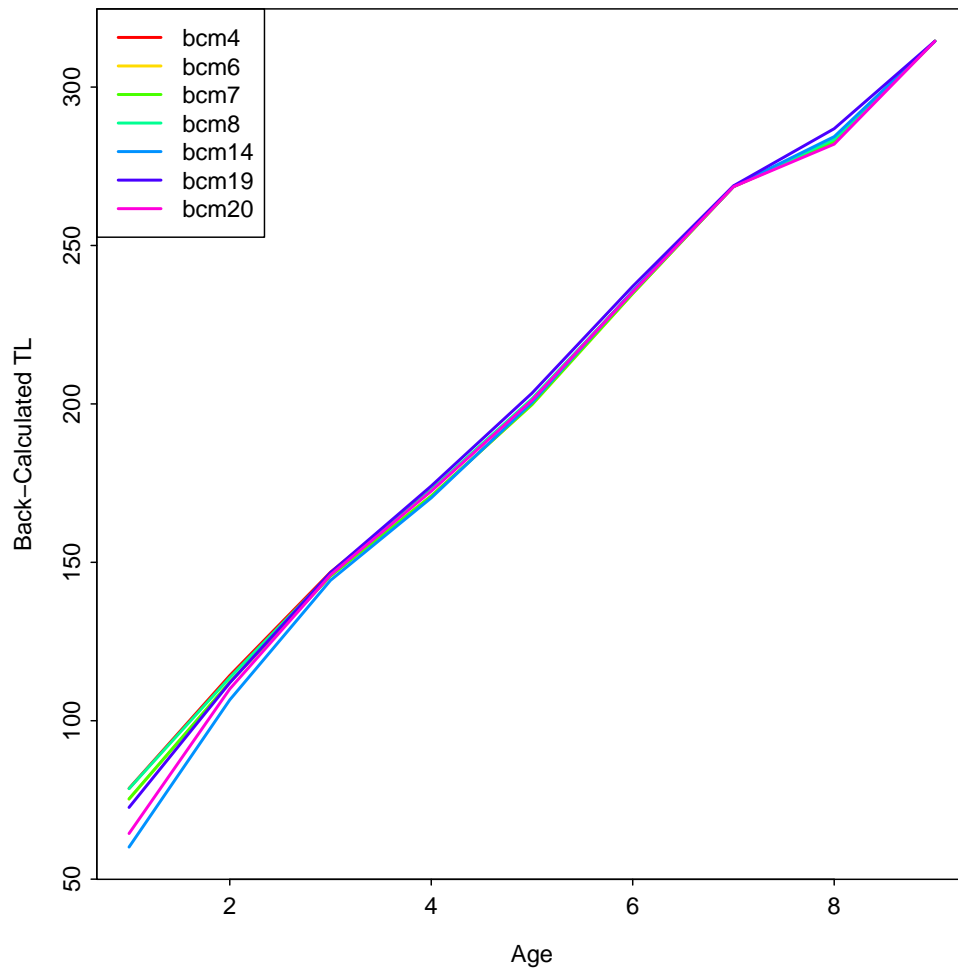


Figure 1. Mean back-calculated lengths-at-age for West Bearskin Lake smallmouth bass for a variety of pairs of models.

References

- Vigliola, L. and M. Meekan. 2009. Tropical Fish Otoliths: Information for Assessment, Management, and Ecology, chapter The back-calculation of fish growth from otoliths, pp. 174–211. Number 11 in Reviews: Methods and Technologies in Fish Biology and Fisheries, Springer. [1](#)
- Weisberg, S. 1993. Using hard-part increment data to estimate age and environmental effects. Canadian Journal of Fisheries and Aquatic Sciences 50(6):1229–1237. [1](#)

Reproducibility Information

Version Information

- **Compiled Date:** Mon Dec 16 2013
- **Compiled Time:** 9:35:28 PM
- **Code Execution Time:** 7.6 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, splines, stats, utils
- **Other Packages:** doBy_4.5-10, FSA_0.4.3, knitr_1.5.15, lattice_0.20-24, MASS_7.3-29, survival_2.37-4
- **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, gdata_2.13.2, gplots_2.12.1, grid_3.0.2, gtools_3.1.1, highr_0.3, Hmisc_3.13-0, KernSmooth_2.23-10, lme4_1.0-5, Matrix_1.1-0, minqa_1.2.1, 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, stringr_0.6.2, tools_3.0.2, zoo_1.7-10
- **Required Packages:** FSA, lattice, doBy and their dependencies (car, gdata, gplots, graphics, grDevices, grid, Hmisc, knitr, lme4, MASS, Matrix, multcomp, nlme, plotrix, quantreg, sciplot, stats, survival, utils)