Using AD Model Builder and R together: getting started with the `R2admb` package

Ben Bolker

November 25, 2009

1 Installation

(Provide hints on where to find AD Model Builder ...)

- `http://admb-project.org/`
- `http://admb-project.org/downloads`
- `http://code.google.com/p/admb-project/`
- `http://code.google.com/p/admb-project/downloads/list`

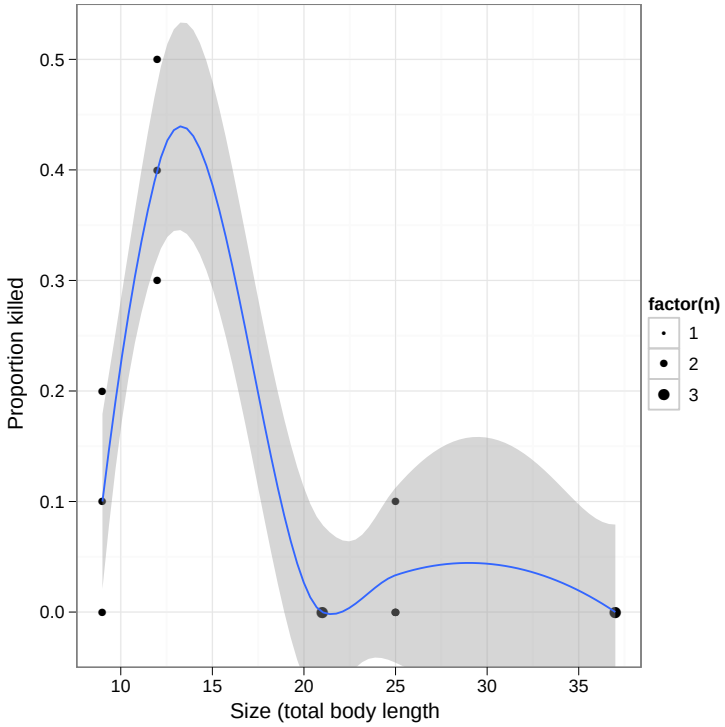
2 Basics

```
> library(R2admb)
> library(ggplot2)

> ReedfrogSizepred <-
+   data.frame(TBL = rep(c(9,12,21,25,37),each=3),
+              Kill = c(0,2,1,3,4,5,0,0,0,0,1,0,0,0,0L))
```

Initial view of the data (proportion killed):

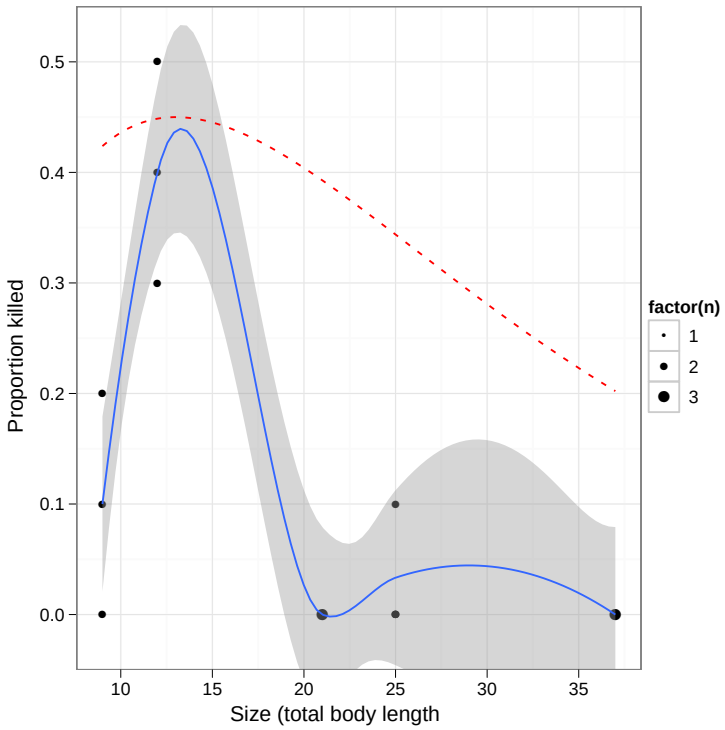
```
> g1  <- ggplot(ReedfrogSizepred,
+              aes(x=TBL,y=Kill/10))+
+   geom_point()+stat_sum(aes(size=factor(..n..)))+
+   geom_smooth()+
+   theme_bw()+
+   labs(size="n",x="Size (total body length",
```

```
+         y="Proportion killed")+
+    coord_cartesian(ylim=c(-0.05,0.55))
> print(g1)
```



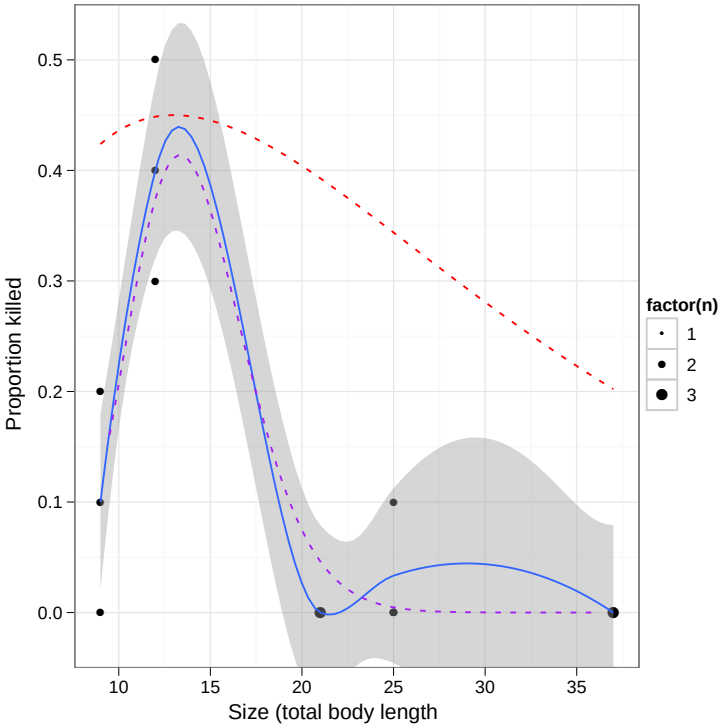
So if $p(\text{kill}) = c((S/d)\exp(1-(S/d)))^g$ (peak occurs at $S = d$, peak height=c) then a reasonable set of first estimates would be $c = 0.45$, $d = 13$.

```
> startest <- stat_function(fun = function(x) { 0.45*((x/13)*exp(1-x/13)) },
+                           lty=2,colour="red")
> print(g1+startest)
```



(Not great, but perhaps adequate as a starting point.)

```
> library(bbmle)
> m0 <- mle2(Kill~dbinom(c*((TBL/d)*exp(1-TBL/d))^g,size=10),
+            start=list(c=0.45,d=13,g=1),data=ReedfrogSizepred,
+            method="L-BFGS-B",
+            lower=c(c=0.003,d=10,g=0),
+            upper=c(c=0.8,d=20,g=20),
+            control=list(parscale=c(c=0.5,d=10,g=1)))

> TBLvec = seq(9.5,36,length=100)
> predfr <-
+   data.frame(TBL=TBLvec,
+              Kill=predict(m0,newdata=data.frame(TBL=TBLvec)))
> print(g1+startest+
+       geom_line(data=predfr,colour="purple",lty=2))
```



Here's the TPL (AD Model Builder definition) file:

```
DATA_SECTION
  init_int nobs                  // # of observations
  init_vector nexposed(1,nobs) // # exposed per trial
  init_vector TBL(1,nobs)      // total body length
  init_vector Kill(1,nobs)     // # killed per trial
PARAMETER_SECTION
  init_bounded_number c(0,1) // baseline mort prob
  init_bounded_number d(0,50) // size scaling factor
  init_bounded_number g(-1,25) // size scaling power
  vector prob(1,nobs)    // per capita mort prob
  objective_function_value f
  sdreport_number rc;    // this & following for MCMC
  sdreport_number rd;
  sdreport_number rg;
PROCEDURE_SECTION
  rc = c; rd = d; rg = g;     // set MCMC reporting
  dvariable fpen=0.0;         // penalty variable
```

```
  // power-Ricker
  prob = c*pow(elem_prod(TBL/d,exp(1-TBL/d)),g);
  // penalties: constrain 0.001 <= prob <= 0.999
  prob = posfun(prob,0.001,fpen);
  f += 1000*fpen;
  prob = 1-posfun(1-prob,0.001,fpen);
  f += 1000*fpen;
  // binomial negative log-likelihood
  f -= sum( log_comb(nexposed,Kill)+
            elem_prod(Kill,log(prob))+
            elem_prod(nexposed-Kill,log(1-prob)));
```

```
> setup_admb()

            ADMB_HOME
"/usr/local/src/admb"

> m1 <- do_admb("ReedfrogSizepred",
+               input=c(list(nobs=nrow(ReedfrogSizepred),
+                 nexposed=rep(10,nrow(ReedfrogSizepred))),
+                 ReedfrogSizepred),
+               param_list=list(c=0.45,d=13,g=1),
+               clean=TRUE,verbose=TRUE)

compiling with args: '  -s ' ...
compile output:
  *** tpl2cpp -bounds  ReedfrogSizepred  *** adcomp   -s ReedfrogSizepred g++ -c -g -
compile log:

writing data and input files ...
running compiled executable with args: '  '...
Run output:


Initial statistics: 3 variables; iteration 0; function evaluation 0
Function value   4.8749328e+01; maximum gradient component mag  -1.4757e+02
Var   Value    Gradient   |Var   Value    Gradient   |Var   Value    Gradient
  1 -0.06377  1.05426e+02 |  2 -0.31873  8.92838e+01 |  3 -0.64218 -1.47567e+02

Intermediate statistics: 3 variables; iteration 10; function evaluation 20
```

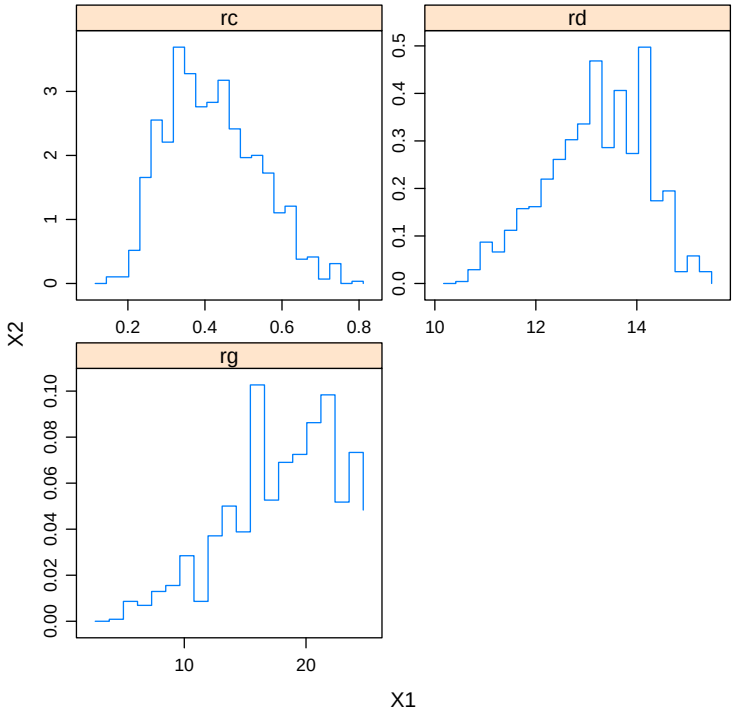
```
Function value   1.7909366e+01; maximum gradient component mag  -6.5352e+00
Var   Value    Gradient   |Var   Value    Gradient   |Var   Value    Gradient
  1 -0.25277 -3.76720e-01 |  2 -0.57193 -6.53522e+00 |  3 -0.62412  9.63776e-01
3 variables; iteration 20; function evaluation 33
Function value   1.4833004e+01; maximum gradient component mag   1.4407e+01
Var   Value    Gradient   |Var   Value    Gradient   |Var   Value    Gradient
  1 -0.21561  9.00119e+00 |  2 -0.33001  1.44067e+01 |  3 -0.17298 -1.21473e+01
3 variables; iteration 30; function evaluation 44
Function value   1.2893758e+01; maximum gradient component mag   3.3696e-04
Var   Value    Gradient   |Var   Value    Gradient   |Var   Value    Gradient
  1 -0.11026 -8.26206e-05 |  2 -0.30859  3.36962e-04 |  3  0.31917 -7.23330e-06

 - final statistics:
3 variables; iteration 31; function evaluation 45
Function value   1.2894e+01; maximum gradient component mag   6.1615e-06
Exit code = 1;  converg criter   1.0000e-04
Var   Value    Gradient   |Var   Value    Gradient   |Var   Value    Gradient
  1 -0.11026 -2.51372e-07 |  2 -0.30859  6.16151e-06 |  3  0.31917 -8.97411e-07
Estimating row 1 out of 3 for hessian
Estimating row 2 out of 3 for hessian
Estimating row 3 out of 3 for hessian




reading output ...

> unlink("reedfrogsizepred.tpl")
```

Exercise the extractor methods:

```
> m1

Model file: reedfrogsizepred
Negative log-likelihood: 12.8938
Coefficients:
         c          d          g
 0.4138331 13.3508215 18.2479066

> coef(m1)
```

```
         c          d          g
 0.4138331 13.3508215 18.2479066

> summary(m1)

Model file: reedfrogsizepred
Negative log-likelihood: 12.8938
Coefficients:
  Estimate Std. Error z value Pr(>|z|)
c   0.4138     0.1257   3.292 0.000996 ***
d  13.3508     0.8111  16.461  < 2e-16 ***
g  18.2479     6.0331   3.025 0.002489 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> coef(summary(m1))

    Estimate Std. Error   z value     Pr(>|z|)
c  0.4138331    0.12572  3.291705 9.958205e-04
d 13.3508215    0.81107 16.460751 7.022349e-61
g 18.2479066    6.03310  3.024632 2.489359e-03

> vcov(m1)

           c         d          g
c 0.01580552 0.0578055  0.5043901
d 0.05780550 0.6578345  2.2464986
g 0.50439009 2.2464986 36.3982956

> logLik(m1)

[1] -12.8938

> deviance(m1)

[1] 25.7876

> AIC(m1)

[1] 31.7876
```

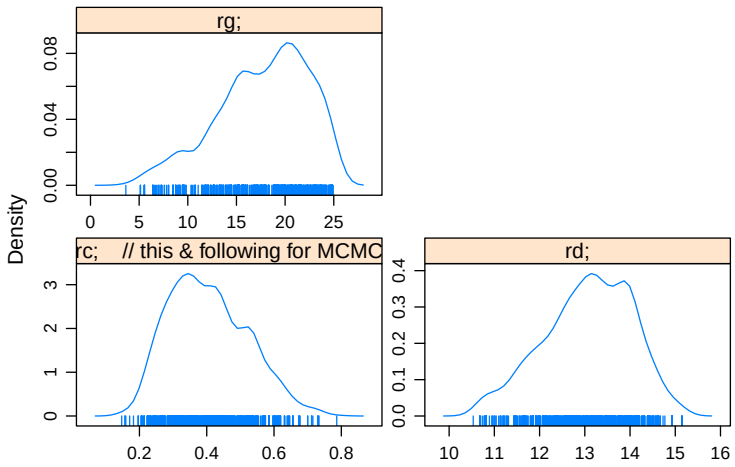
```
> m1MC <- do_admb("ReedfrogSizepred",
+                 input=c(list(nobs=nrow(ReedfrogSizepred),
+                   nexposed=rep(10,nrow(ReedfrogSizepred))),
+                   ReedfrogSizepred),
+                   mcmc=TRUE,
+                   param_list=list(c=0.45,d=13,g=1),
+                   clean=TRUE)
> unlink("reedfrogsizepred.tpl")

> print(plot(m1MC$hist))
```



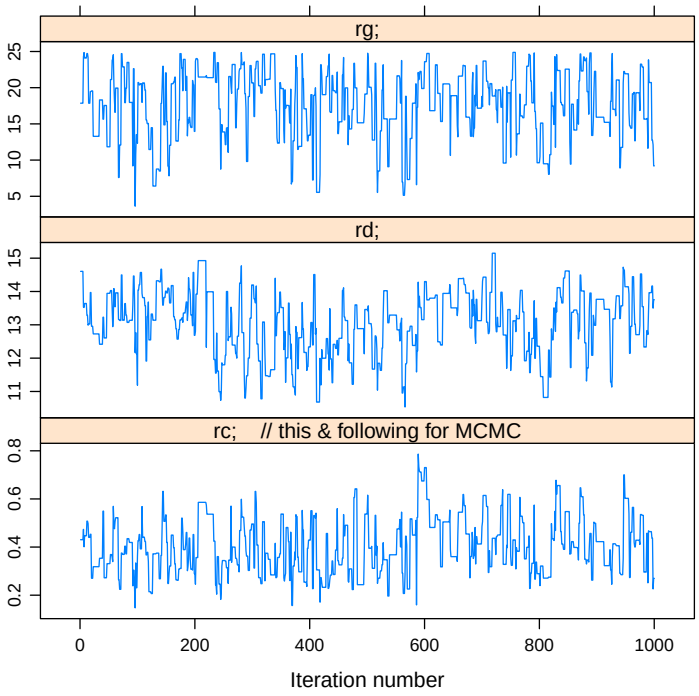

```
> library(coda)
> HPDinterval(as.mcmc(m1MC$mcmc))

                                        lower      upper
rc;    // this & following for MCMC  0.2071513  0.6190827
rd;                                 10.9495253 14.6161983
rg;                                  8.7644415 24.8812550
```

```
attr(,"Probability")
[1] 0.95
```

```
> print(densityplot(as.mcmc(m1MC$mcmc)))
```




```
> print(xyplot(as.mcmc(m1MC$mcmc)))
```


rg;
rd;
rc; // this & following for MCMC
Iteration number