

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

Ben Bolker

March 22, 2010

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>

Notes: (1) depending on your operating system etc., you may need to install a C++ compiler (in particular, the MacOS installation instructions will probably ask you to install gcc/g++ from the Xcode package); (2) you will need to have the scripts **admb**, **adcomp**, and **adlink** in the **bin** directory of your ADMB installation.

2 Basics

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

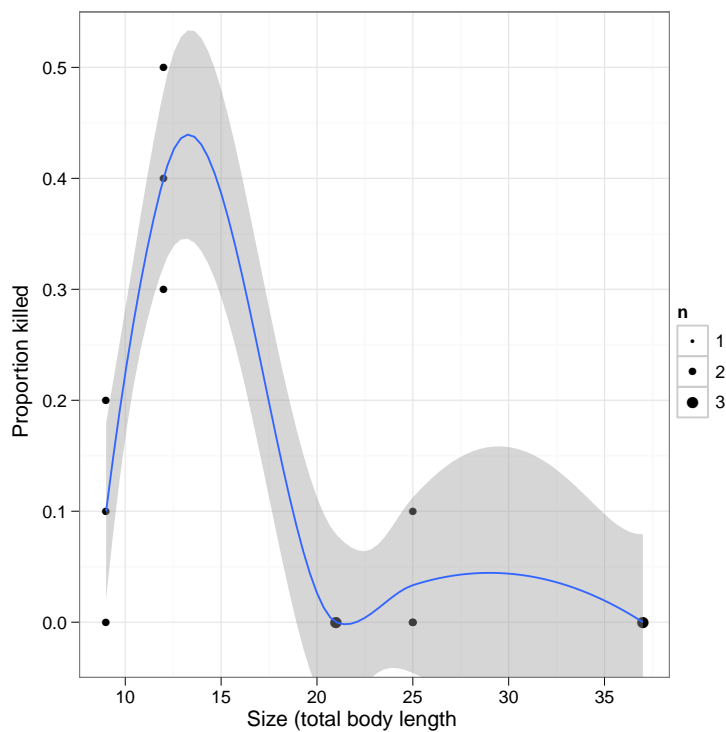
> ReedfrogSizepred <-
+   data.frame(TBL = rep(c(9,12,21,25,37),each=3),
+               Kill = c(0,2,1,3,4,5,0,0,0,0,1,0,0,0,0L))
```

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)

```

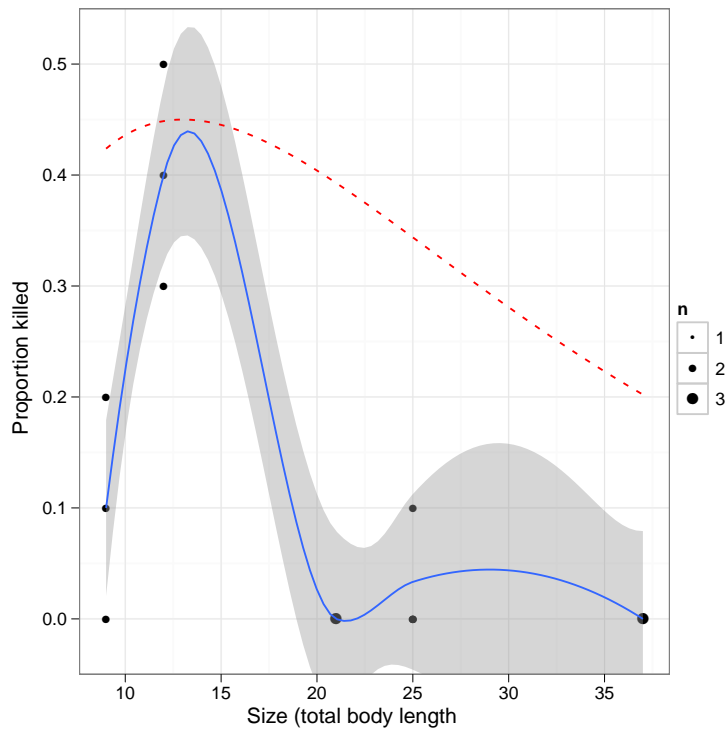


So if $p(\text{kill}) = c((S/d) \exp(1 - (S/d)))^g$ (peak occurs at $S = d$, peak height= c) then a reasonable set of first estimates would be $c = 0.45$, $d = 13$.

```

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

```

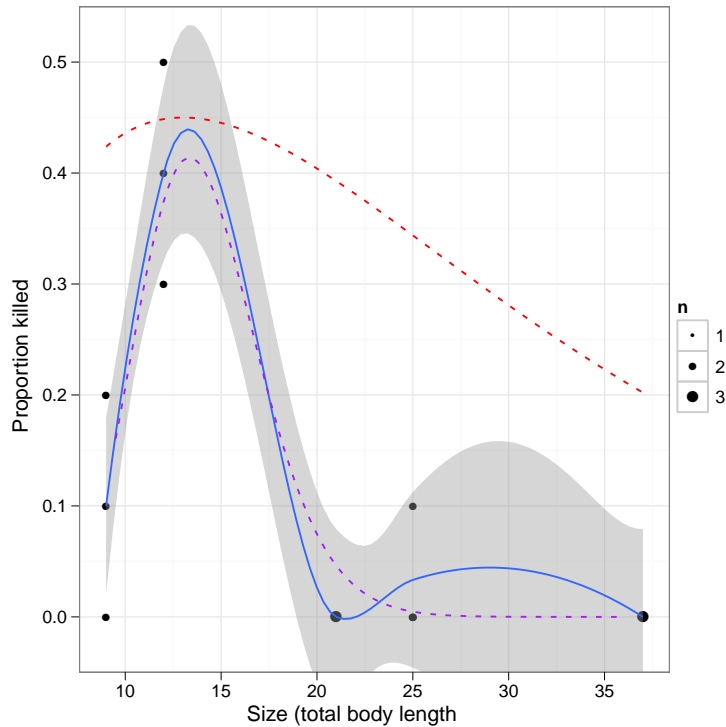


(Not great, but

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:

```

1 DATA_SECTION
2   init_int nobs                // # of observations
3   init_vector nexposed(1,nobs) // # exposed per trial
4   init_vector TBL(1,nobs)      // total body length
5   init_vector Kill(1,nobs)     // # killed per trial
6 PARAMETER_SECTION
7   init_bounded_number c(0,1) // baseline mort prob
8   init_bounded_number d(0,50) // size scaling factor
9   init_bounded_number g(-1,25) // size scaling power
10  vector prob(1,nobs)         // per capita mort prob
11  objective_function_value f
12  sdreport_number rc;         // this & following for MCMC
13  sdreport_number rd;
14  sdreport_number rg;
15 PROCEDURE_SECTION
16  rc = c; rd = d; rg = g;     // set MCMC reporting
17  dvariable fpen=0.0;         // penalty variable

```

```

18 // power-Ricker
19 prob = c*pow(elem_prod(TBL/d,exp(1-TBL/d)),g);
20 // penalties: constrain 0.001 <= prob <= 0.999
21 prob = posfun(prob,0.001,fpen);
22 f += 1000*fpen;
23 prob = 1-posfun(1-prob,0.001,fpen);
24 f += 1000*fpen;
25 // binomial negative log-likelihood
26 f -= sum( log_comb(nexposed,Kill)+
27           elem_prod(Kill,log(prob))+
28           elem_prod(nexposed-Kill,log(1-prob)));

```

```

> setup_admb()
> m1 <- do_admb("ReedfrogSizepred",
+             data=c(list(nobs=nrow(ReedfrogSizepred),
+             nexposed=rep(10,nrow(ReedfrogSizepred))),
+             ReedfrogSizepred),
+             param_list=list(c=0.45,d=13,g=1),
+             clean=TRUE)
> unlink("reedfrogsizedpred.tpl")

```

Exercise the extractor methods:

```

> m1

Model file: reedfrogsizedpred
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: reedfrogsizedpred
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

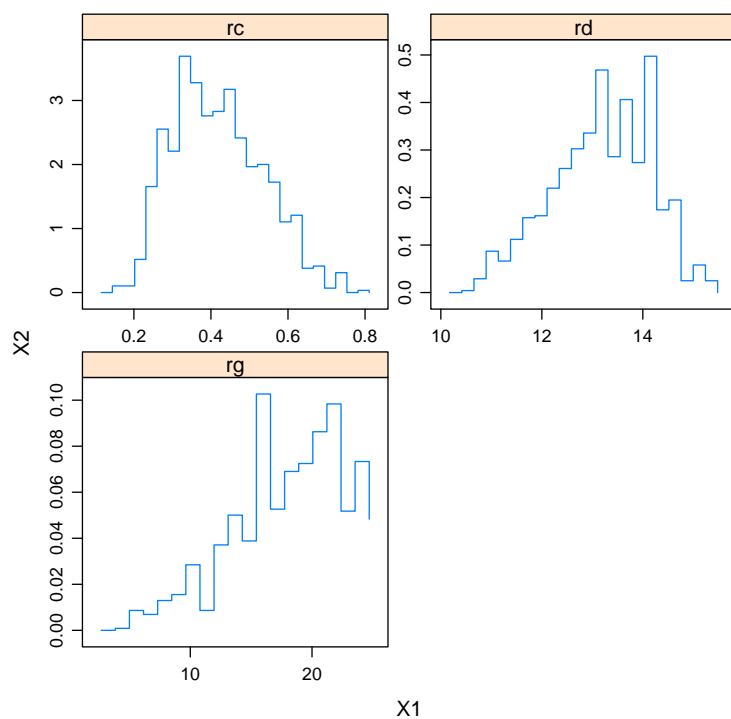
> AIC(m1)

[1] 31.7876

> m1MC <- do_admb("ReedfrogSizepred",
+               data=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("reedfrogsizedpred.tpl")

```

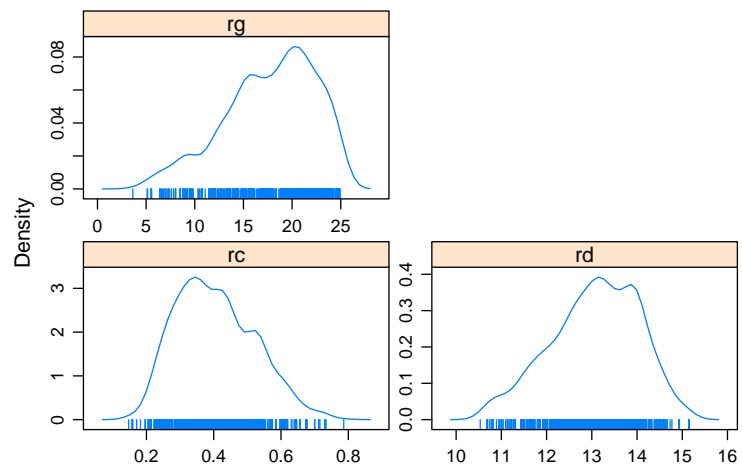
```
> print(plot(m1MC$hist))
```



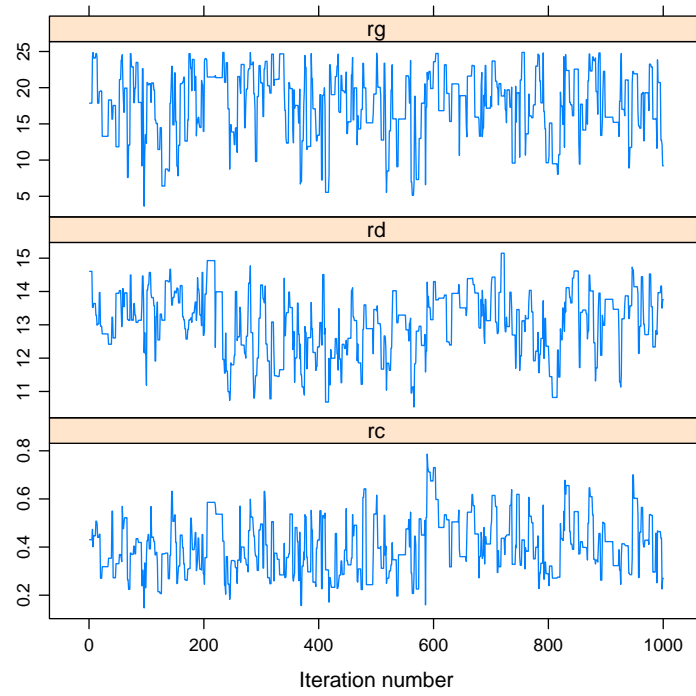
```
> library(coda)
> HPDinterval(as.mcmc(m1MC$mcmc))
```

```
      lower      upper
rc 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)))
```

3 Random effects: gopher tortoise data