

The `mle2` package

Ben Bolker

September 18, 2009

Contents

1 Introduction	1
2 Example	1

1 Introduction

There's not much here yet ... but hopefully I will be adding to this in the future.

To do:

- use `predict`, `simulate` etc. to demonstrate different parametric bootstrap approaches to confidence and prediction intervals
 - use `predict` to get means and standard deviations, use delta method?
 - use `vcov`, assuming quadratic profiles, with `predict(..., newparams=...)`
 - prediction intervals assuming no parameter uncertainty with `simulate`
 - both together ...

2 Example

```
> frogdat <- data.frame(size = rep(c(9, 12, 21, 25, 37), each = 3),
+   killed = c(0, 2, 1, 3, 4, 5, rep(0, 4), 1, rep(0, 4)))
> frogdat$initial <- rep(10, nrow(frogdat))

> library(bbmle)
> library(ggplot2)
> print(gg1 <- ggplot(frogdat, aes(x = size, y = killed)) + geom_point() +
+   stat_sum(aes(size = factor(..n..))) + labs(size = "#") +
+   scale_x_continuous(limits = c(0, 40)))
> m1 <- mle2(killed ~ dbinom(prob = c * (size/d)^g * exp(1 - size/d),
+   size = initial), data = frogdat, start = list(c = 0.5, d = 5,
```

```

+     g = 1))
> pdat <- data.frame(size = 1:40, initial = rep(10, 40))
> pdat1 <- data.frame(pdat, killed = predict(m1, newdata = pdat))
> m2 <- mle2(killed ~ dbinom(prob = c * ((size/d) * exp(1 - size/d))^g,
+     size = initial), data = frogdat, start = list(c = 0.5, d = 5,
+     g = 1))
> pdat2 <- data.frame(pdat, killed = predict(m2, newdata = pdat))
> gg1 + geom_line(data = pdat1, colour = "red") + geom_line(data = pdat2,
+     colour = "blue")

> coef(m2)

           c           d           g
0.4138847 13.3517574 18.2511264

> prof2 <- profile(m2)

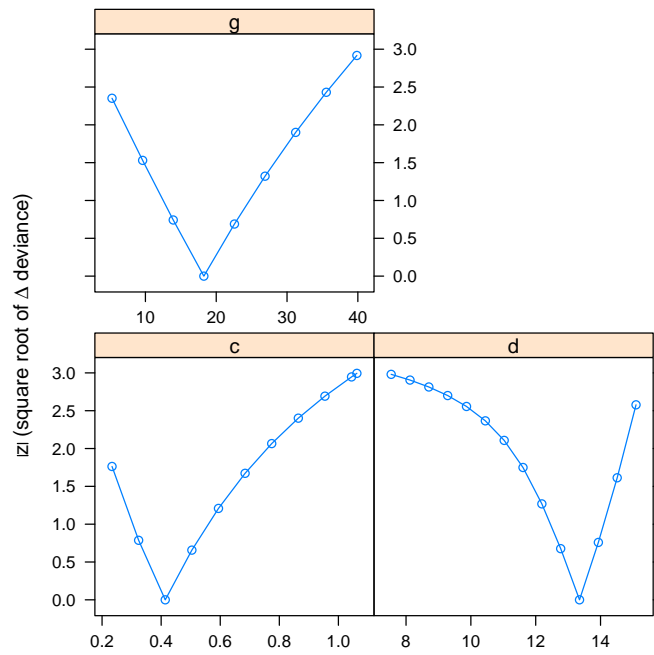
Three different ways to draw the profile:

> print(plot(prof2))

NULL

> prof2_df <- as.data.frame(prof2)
> library(lattice)
> print(xyplot(abs(z) ~ focal | param, data = prof2_df, subset = abs(z) <
+     3, type = "b", xlab = "", ylab = expression(paste(abs(z),
+     " (square root of ", Delta, " deviance)")), scale = list(x = list(relation = "free"))))

```



```
> print(ggplot(subset(prof2_df, abs(z) < 3), aes(x = focal, y = abs(z))) +
+       geom_line() + facet_wrap(~param, scale = "free_x"))
```

