Output from CRAN bcrm 0.4.7

Harry Southworth

15th December 2018

Contents

1 Introduction

This document contains the output of running the examples in the bcrm help file, as installed from CRAN on 2018-12-08. The reason is to be able to compare future versions with the current CRAN version.

Some of the plots are slightly different (titles and point shapes) due to changes to github.com/mikesweeting/bcrm that have not been published to CRAN.

Note that one of the examples (with the 3+3 design) failed to run through knitr and rmarkdown. Also, there is a final example which (on some runs) reveals a bug: there is a systematic change in the simulated trials after the 100th.

1.1 Example 1

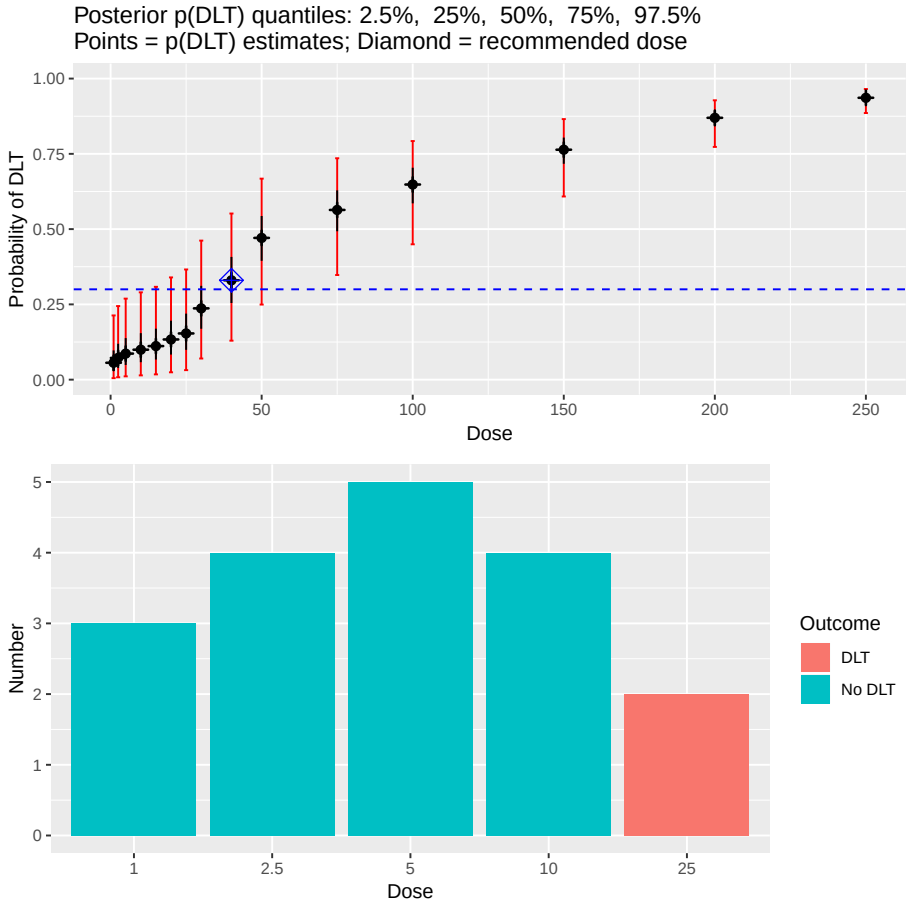
```
startTime <- Sys.time()

seed  <-  20181208
set.seed(seed)

library(bcrm)
library(ggplot2)

## Dose-escalation cancer trial example as described in Neuenschwander et al 2008.
## Pre-defined doses
dose <- c(1, 2.5, 5, 10, 15, 20, 25, 30, 40, 50, 75,
          100, 150, 200, 250)
## Pre-specified probabilities of toxicity
## [dose levels 11-15 not specified in the paper,  and are for illustration only]
p.tox0 <- c(0.010, 0.015, 0.020, 0.025, 0.030, 0.040,
            0.050, 0.100, 0.170, 0.300, 0.400, 0.500,
            0.650, 0.800, 0.900)
## Data from the first 5 cohorts of 18 patients
data <- data.frame(patient=1:18,
                   dose=rep(c(1:4, 7), c(3, 4, 5, 4, 2)),
                   tox=rep(0:1, c(16, 2)))
## Target toxicity level
target.tox <- 0.30

## A 1-parameter power model is used,  with standardised doses calculated using
## the plug-in prior median
## Prior for alpha is lognormal with mean 0 (on log scale)
## and standard deviation 1.34 (on log scale)
## The recommended dose for the next cohort if posterior mean is used
Power.LN.bcrm <- bcrm(stop=list(nmax=18), data=data,
                      p.tox0=p.tox0, dose=dose, ff="power",
                      prior.alpha=list(3, 0, 1.34^2),
                      target.tox=target.tox, constrain=FALSE,
                      sdose.calculate="median", pointest="mean")
```

```
##
##  Stopping:  Reached maximum sample size
```

```
print(Power.LN.bcrm)
```

```
##
##  Estimation method:  exact
##  Target toxicity level:  0.3
##  Model:  1-parameter power
##  Prior:  Lognormal( Mean:0,  Variance:1.7956)
##
##  Standardised doses (skeleton):
##     1   2.5     5    10    15    20    25    30    40    50    75   100
## 0.010 0.015 0.020 0.025 0.030 0.040 0.050 0.100 0.170 0.300 0.400 0.500
##   150   200   250
## 0.650 0.800 0.900
##
##  Unmodified (unconstrained) CRM used
##
##  Posterior mean estimate of probability of toxicity used to select next dose
##
##  Toxicities observed:
##             Doses
##              1 2.5 5 10 15 20 25 30 40 50 75 100 150 200 250
##   n          3   4 5  4  0  0  2  0  0  0  0   0   0   0   0
##   Toxicities 0   0 0  0  0  0  2  0  0  0  0   0   0   0   0
##
##  Posterior estimates of toxicity:
##         Doses
##               1    2.5      5     10     15     20     25    30    40
##   Mean   0.0702 0.0866 0.1010 0.1130 0.1250 0.1460 0.1650 0.244 0.333
##   SD     0.0558 0.0630 0.0686 0.0731 0.0769 0.0831 0.0879 0.102 0.109
##   Median 0.0561 0.0723 0.0865 0.0995 0.1110 0.1330 0.1530 0.237 0.330
##         Doses
##             50     75   100   150    200    250
##   Mean   0.467 0.5580 0.641 0.757 0.8650 0.9330
##   SD     0.108 0.0996 0.088 0.066 0.0398 0.0205
##   Median 0.471 0.5640 0.648 0.764 0.8700 0.9360
##          Doses
## Quantiles       1     2.5      5     10     15     20     25     30    40
##     2.5%  0.00493 0.00787 0.0110 0.0142 0.0175 0.0244 0.0316 0.0702 0.130
##     25%   0.02860 0.03910 0.0488 0.0579 0.0667 0.0833 0.0990 0.1690 0.255
##     50%   0.05610 0.07230 0.0865 0.0995 0.1110 0.1330 0.1530 0.2370 0.330
##     75%   0.09710 0.11900 0.1380 0.1540 0.1690 0.1960 0.2190 0.3120 0.408
##     97.5% 0.21300 0.24400 0.2690 0.2900 0.3080 0.3400 0.3660 0.4620 0.552
##          Doses
## Quantiles    50    75   100   150   200   250
##     2.5%  0.249 0.347 0.450 0.608 0.773 0.886
##     25%   0.395 0.493 0.586 0.717 0.842 0.922
##     50%   0.471 0.564 0.648 0.764 0.870 0.936
```

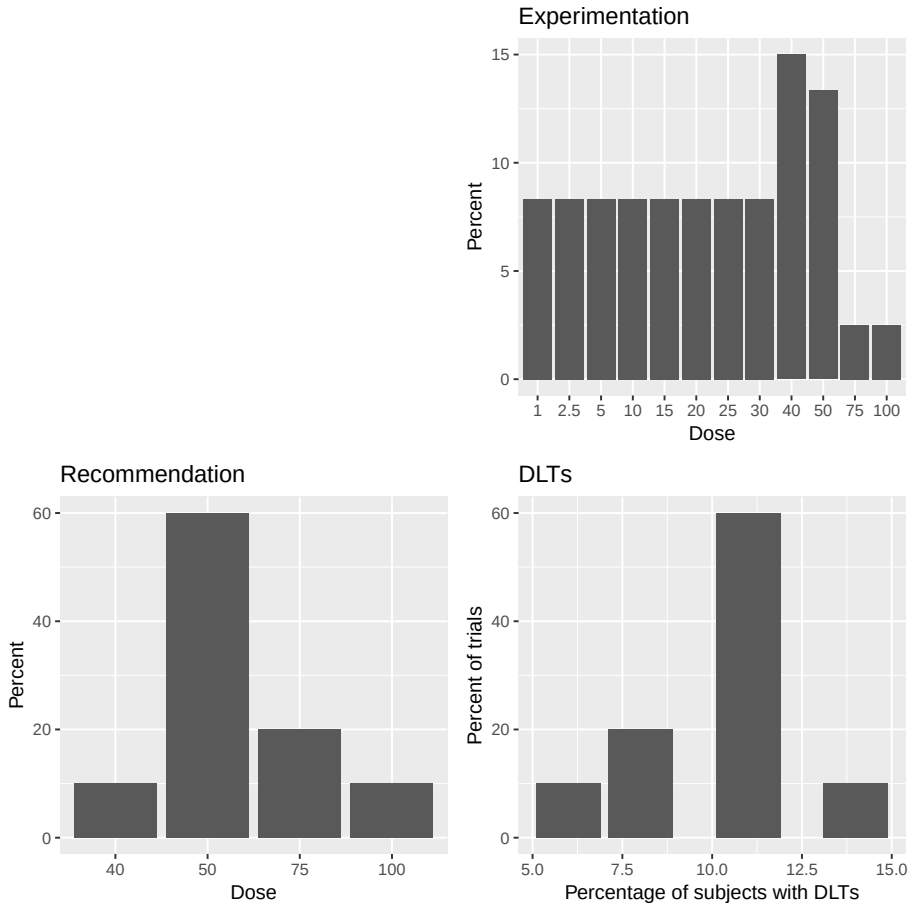
```
##      75%   0.544 0.629 0.704 0.804 0.893 0.948
##      97.5% 0.668 0.735 0.792 0.865 0.928 0.965
##
##  Next recommended dose:  40

plot(Power.LN.bcrm)
```



1.2 Example 2

```
## Simulate 10 replicate trials of size 36 (cohort size 3) using this design
## with constraint (i.e. no dose-skipping) and starting at lowest dose
## True probabilities of toxicity are set to pre-specified probabilities (p.tox0)
Power.LN.bcrm.sim <- bcrm(stop=list(nmax=36), p.tox0=p.tox0,
                          dose=dose, ff="power",
                          prior.alpha=list(3, 0, 1.34^2),
                          target.tox=target.tox, constrain=TRUE,
                          sdose.calculate="median", pointest="mean",
                          start=1, simulate=TRUE, nsims=10,
                          truep=p.tox0)
```

```
## Simulated trial:  10

print(Power.LN.bcrm.sim)

## Operating characteristics based on  10  simulations:
##
##
## Sample size 36
##
##                              Doses
##                               No dose      1    2.5      5     10     15
##   Experimentation proportion      NA 0.0833 0.0833 0.0833 0.0833 0.0833
##   Recommendation proportion        0 0.0000 0.0000 0.0000 0.0000 0.0000
##                              Doses
##                                   20     25     30   40    50    75   100
##   Experimentation proportion 0.0833 0.0833 0.0833 0.15 0.133 0.025 0.025
##   Recommendation proportion  0.0000 0.0000 0.0000 0.10 0.600 0.200 0.100
##                              Doses
##                               150 200 250
##   Experimentation proportion   0   0   0
##   Recommendation proportion    0   0   0
##
##                              Probability of DLT
##                               [0,0.2] (0.2,0.4] (0.4,0.6] (0.6,0.8] (0.8,1]
##   Experimentation proportion   0.817     0.158     0.025         0       0
##   Recommendation proportion    0.100     0.800     0.100         0       0

plot(Power.LN.bcrm.sim)
```



1.3 Example 3

Note that in the following example, threep3=TRUE but it won't work in the CRAN version because it prompts for a go-ahead and knitr and rmarkdown can't cope with that.

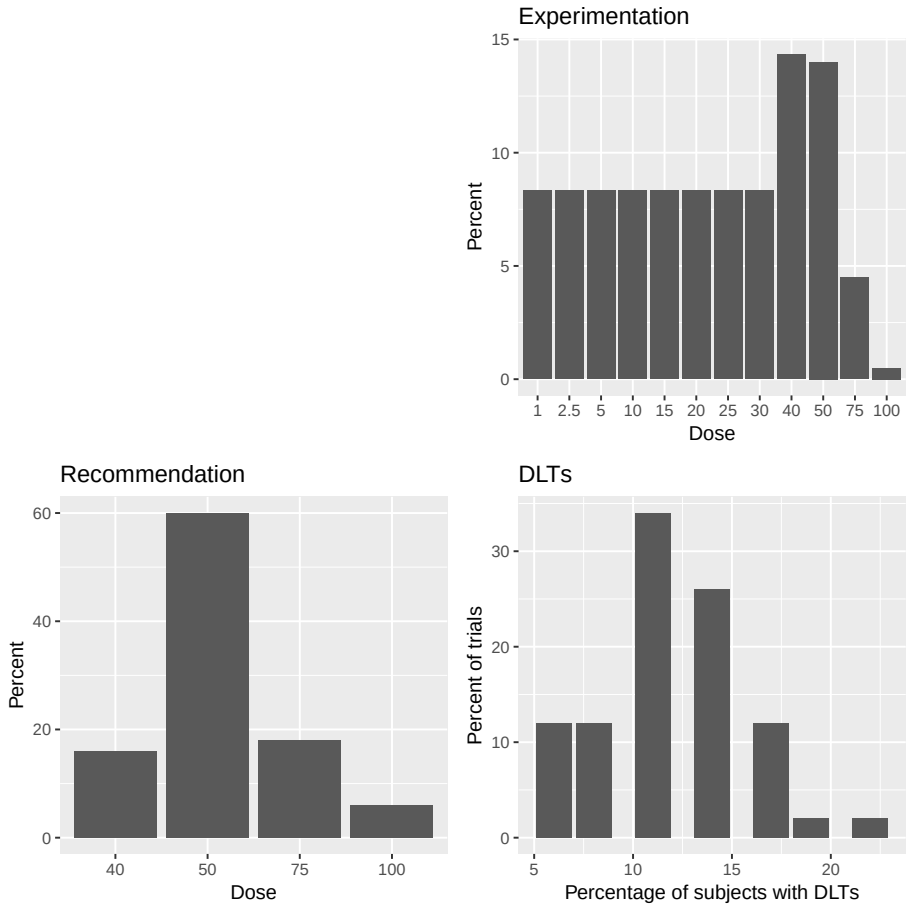
```
## Comparing this CRM design with the standard 3+3 design
## (only considering the first 12 dose levels)
Power.LN.bcrm.compare.sim <- bcrm(stop=list(nmax=36),
                                  p.tox0=p.tox0[1:12],
                                  dose=dose[1:12], ff="power",
                                  prior.alpha=list(3, 0, 1.34^2),
                                  target.tox=target.tox,
                                  constrain=TRUE,
                                  sdose.calculate="median",
                                  pointest="mean", start=1,
                                  simulate=TRUE, nsims=50,
                                  truep=p.tox0[1:12], threep3=TRUE,
                                  quietly=TRUE)
```

```
##  Calculating operating characteristics of a standard 3+3 trial for
comparison...
## 12% complete
## 16% complete
## 20% complete
## 24% complete
## 28% complete
## 32% complete
## 36% complete
## 40% complete
## 44% complete
## 48% complete
## 52% complete
## 56% complete
## 60% complete
## 64% complete
## 68% complete
## 72% complete
## 76% complete
## 80% complete
## 84% complete
## 88% complete
## 92% complete
## 96% complete
## 100% complete
```

```
print(Power.LN.bcrm.compare.sim)
```

```
## Operating characteristics based on  50  simulations:
##
##
## Sample size 36
##
##                              Doses
##                               No dose      1    2.5      5     10     15
##   Experimentation proportion      NA 0.0833 0.0833 0.0833 0.0833 0.0833
##   Recommendation proportion        0 0.0000 0.0000 0.0000 0.0000 0.0000
##                              Doses
##                                   20     25     30    40   50    75   100
##   Experimentation proportion 0.0833 0.0833 0.0833 0.143 0.14 0.045 0.005
##   Recommendation proportion  0.0000 0.0000 0.0000 0.160 0.60 0.180 0.060
##
##                              Probability of DLT
##                               [0,0.2] (0.2,0.4] (0.4,0.6] (0.6,0.8] (0.8,1]
##   Experimentation proportion    0.81     0.185     0.005         0       0
##   Recommendation proportion     0.16     0.780     0.060         0       0
```

```
plot(Power.LN.bcrm.compare.sim)
```



1.4 Example 4

```
## A 2-parameter model,  using priors as specified in Neuenschwander et al 2008.
## Posterior mean used to choose the next dose
## Standardised doses using reference dose,  250mg
set.seed(20181214)

sdose <- log(dose/250)
## Bivariate lognormal prior for two parameters
mu <- c(2.15, 0.52)
Sigma <- rbind(c(0.84^2, 0.134), c(0.134, 0.80^2))
## Using rjags (requires JAGS to be installed)
TwoPLogistic.mean.bcrm <- bcrm(stop=list(nmax=18), data=data,
                               sdose=sdose, dose=dose,
                               ff="logit2",
                               prior.alpha=list(4, mu, Sigma),
                               target.tox=target.tox,
                               constrain=FALSE,
                               pointest="mean", method="rjags")
```

```
##
##  Stopping:  Reached maximum sample size
```

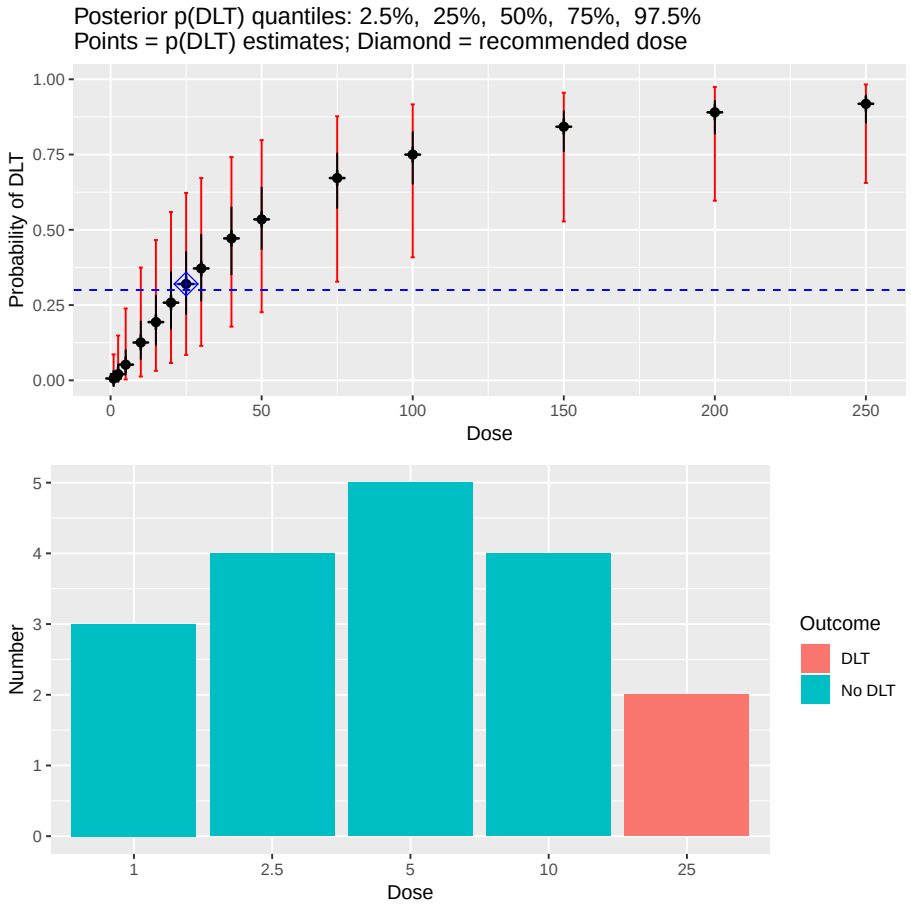
```
print(TwoPLogistic.mean.bcrm)
```

```
##
##  Estimation method:  rjags
##  Target toxicity level:  0.3
##  Model:  Two-parameter logistic
##  Prior:  Log Multivariate Normal
```

```
## Mean Vector:
```

```
## [1] 2.15 0.52
##
## Variance-Covariance Matrix:
##        [,1]  [,2]
## [1,] 0.7056 0.134
## [2,] 0.1340 0.640
##
##  Standardised doses (skeleton):
##          1        2.5          5         10         15         20
## -5.5214609 -4.6051702 -3.9120230 -3.2188758 -2.8134107 -2.5257286
##         25         30         40         50         75        100
## -2.3025851 -2.1202635 -1.8325815 -1.6094379 -1.2039728 -0.9162907
##        150        200        250
## -0.5108256 -0.2231436  0.0000000
##
##  Unmodified (unconstrained) CRM used
##
##  Posterior mean estimate of probability of toxicity used to select next dose
##
##  Toxicities observed:
##             Doses
##              1 2.5 5 10 15 20 25 30 40 50 75 100 150 200 250
##   n          3   4 5  4  0  0  2  0  0  0  0   0   0   0   0
##   Toxicities 0   0 0  0  0  0  2  0  0  0  0   0   0   0   0
##
##  Posterior estimates of toxicity:
##         Doses
##               1    2.5      5     10    15    20    25    30    40    50
##   Mean   0.0155 0.0365 0.0719 0.1420 0.209 0.270 0.326 0.377 0.463 0.533
##   SD     0.0235 0.0412 0.0637 0.0958 0.117 0.130 0.139 0.143 0.147 0.146
##   Median 0.0060 0.0205 0.0520 0.1260 0.193 0.258 0.320 0.372 0.471 0.535
##         Doses
##            75   100   150   200    250
##   Mean   0.655 0.731 0.816 0.861 0.8880
##   SD     0.137 0.127 0.112 0.101 0.0934
##   Median 0.672 0.750 0.842 0.890 0.9180
##          Doses
## Quantiles        1      2.5       5     10     15     20     25    30
```

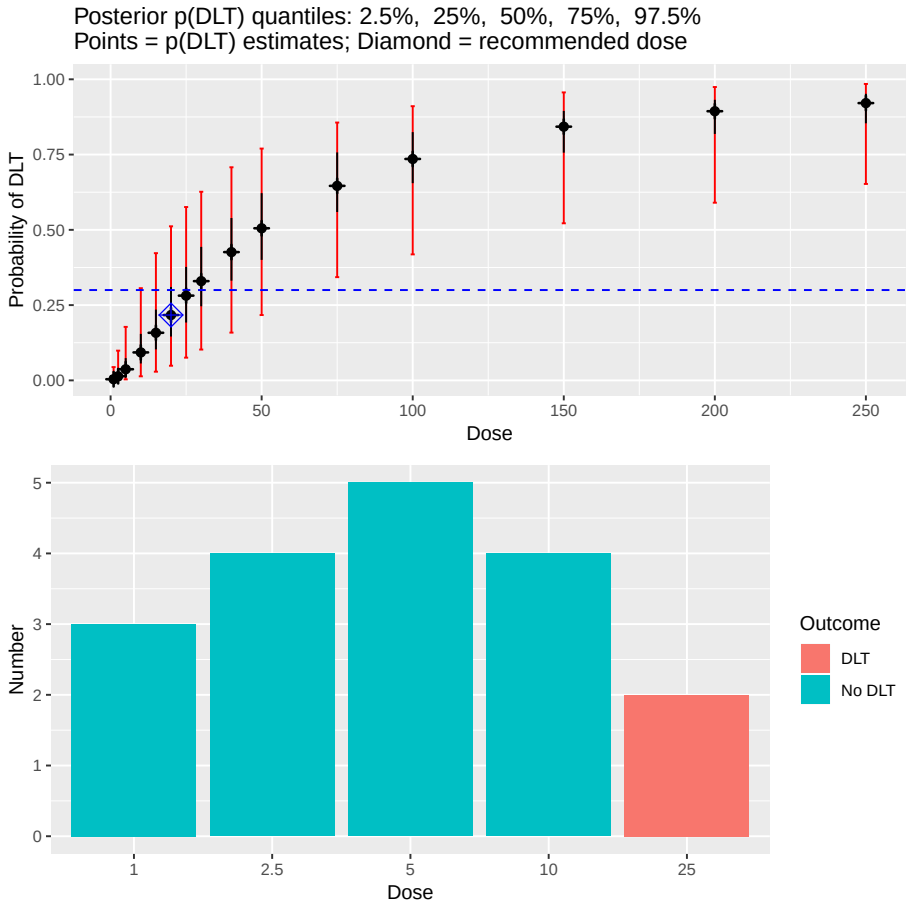
```
##     2.5%  6.53e-05 0.000602 0.00285 0.0127 0.0314 0.0575 0.0843 0.114
##     25%   2.02e-03 0.008370 0.02500 0.0680 0.1150 0.1690 0.2180 0.262
##     50%   6.00e-03 0.020500 0.05200 0.1260 0.1930 0.2580 0.3200 0.372
##     75%   1.91e-02 0.053000 0.10300 0.1980 0.2830 0.3620 0.4280 0.486
##     97.5% 8.61e-02 0.149000 0.23900 0.3750 0.4660 0.5590 0.6230 0.672
##          Doses
## Quantiles    40    50    75   100   150   200   250
##     2.5%  0.178 0.226 0.328 0.409 0.528 0.597 0.656
##     25%   0.350 0.434 0.570 0.651 0.759 0.817 0.854
##     50%   0.471 0.535 0.672 0.750 0.842 0.890 0.918
##     75%   0.577 0.643 0.756 0.827 0.896 0.931 0.949
##     97.5% 0.742 0.798 0.877 0.917 0.955 0.975 0.983
##
##  Next recommended dose:  25

plot(TwoPLogistic.mean.bcrm)
```



1.5 Example 5

```
## A 2-parameter model,  using an EWOC design with feasibility
## bound (MTD quantile)
## of 0.25 to choose the next dose
## Using rjags (requires JAGS to be installed)
TwoPLogistic.EWOC0.25.bcrm <- bcrm(stop=list(nmax=18), data=data,
                                   sdose=sdose, dose=dose,
                                   ff="logit2",
                                   prior.alpha=list(4, mu, Sigma),
                                   target.tox=target.tox,
                                   constrain=FALSE,
                                   pointest=0.25, method="rjags")

##
##  Stopping:  Reached maximum sample size

print(TwoPLogistic.EWOC0.25.bcrm)

##
##  Estimation method:  rjags
##  Target toxicity level:  0.3
##  Model:  Two-parameter logistic
##  Prior:  Log Multivariate Normal

## Mean Vector:

## [1] 2.15 0.52
##
## Variance-Covariance Matrix:
##        [,1]  [,2]
## [1,] 0.7056 0.134
## [2,] 0.1340 0.640
##
##  Standardised doses (skeleton):
##          1        2.5          5         10         15         20
## -5.5214609 -4.6051702 -3.9120230 -3.2188758 -2.8134107 -2.5257286
##         25         30         40         50         75        100
## -2.3025851 -2.1202635 -1.8325815 -1.6094379 -1.2039728 -0.9162907
##        150        200        250
## -0.5108256 -0.2231436  0.0000000
##
##  Unmodified (unconstrained) CRM used
##
##  25 percentile of (standardised) MTD distribution used to select next dose
##  25 percentile is: -2.562308
##
##  Toxicities observed:
##             Doses
##              1 2.5 5 10 15 20 25 30 40 50 75 100 150 200 250
##   n          3   4 5  4  0  0  2  0  0  0  0   0   0   0   0
##   Toxicities 0   0 0  0  0  0  2  0  0  0  0   0   0   0   0
##
```

```
##  Posterior estimates of toxicity:
##         Doses
##                1    2.5      5     10    15    20    25    30    40    50
##   Mean   0.00911 0.0248 0.0535 0.1150 0.177 0.236 0.292 0.344 0.434 0.508
##   SD     0.01360 0.0288 0.0504 0.0831 0.106 0.121 0.131 0.138 0.145 0.146
##   Median 0.00387 0.0139 0.0369 0.0929 0.158 0.217 0.281 0.330 0.426 0.505
##         Doses
##             75   100   150    200    250
##   Mean   0.641 0.723 0.816 0.8650 0.8940
##   SD     0.138 0.127 0.107 0.0928 0.0817
##   Median 0.646 0.735 0.843 0.8940 0.9210
##          Doses
## Quantiles        1      2.5       5     10     15     20     25    30
##     2.5%  0.000085 0.000623 0.00331 0.0135 0.0289 0.0486 0.0755 0.102
##     25%   0.001070 0.005480 0.01970 0.0562 0.1030 0.1460 0.1920 0.246
##     50%   0.003870 0.013900 0.03690 0.0929 0.1580 0.2170 0.2810 0.330
##     75%   0.010600 0.032500 0.07380 0.1540 0.2350 0.3100 0.3770 0.443
##     97.5% 0.044100 0.098700 0.17800 0.3060 0.4230 0.5120 0.5760 0.626
##          Doses
## Quantiles    40    50    75   100   150   200   250
##     2.5%  0.159 0.217 0.343 0.418 0.522 0.590 0.652
##     25%   0.331 0.400 0.559 0.655 0.756 0.819 0.854
##     50%   0.426 0.505 0.646 0.735 0.843 0.894 0.921
##     75%   0.539 0.622 0.757 0.825 0.895 0.932 0.952
##     97.5% 0.708 0.770 0.856 0.911 0.957 0.974 0.985
##
##  Next recommended dose:  20

plot(TwoPLogistic.EWOC0.25.bcrm)
```



1.6 Example 6

```
## A 2-parameter model,  using a loss function based on
## intervals of toxicity to choose the next dose
## Using rjags (requires JAGS to be installed)
## Toxicity cut-points
tox.cutpoints <- c(0.2, 0.35, 0.6)
## Losses associated with toxicity intervals
## [0, 0.2]=1,  (0.2, 0.35]=0,  (0.35, 0.6]=1,  (0.6, 1]=2
loss <- c(1, 0, 1, 2)
TwoPLogistic.tox.intervals.bcrm <- bcrm(stop=list(nmax=18),
                                        data=data, sdose=sdose,
                                        dose=dose, ff="logit2",
                                        prior.alpha=list(4, mu, Sigma),
                                        target.tox=target.tox,
                                        constrain=FALSE,
                                        tox.cutpoints=tox.cutpoints,
                                        loss=loss, method="rjags")
```

```
##
##  Stopping:  Reached maximum sample size
```

```
print(TwoPLogistic.tox.intervals.bcrm)
```

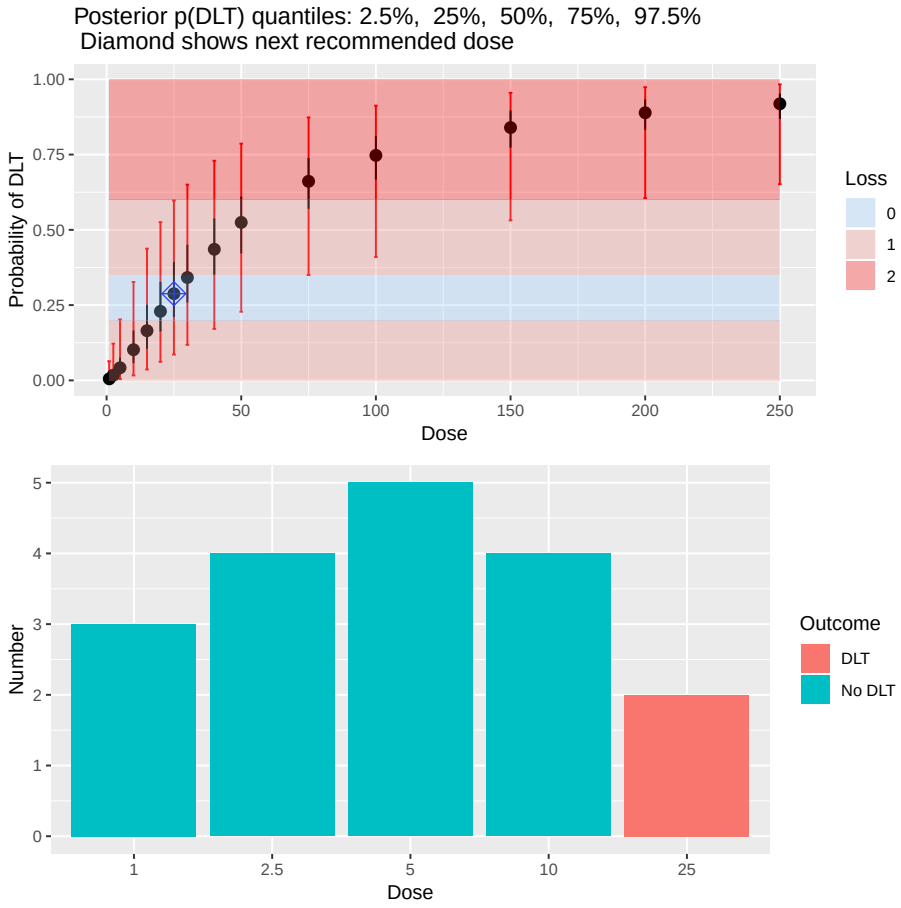
```
##
##  Estimation method:  rjags
##  Target toxicity level:  0.3
##  Model:  Two-parameter logistic
##  Prior:  Log Multivariate Normal
```

```
## Mean Vector:
```

```
## [1] 2.15 0.52
##
## Variance-Covariance Matrix:
##        [,1]  [,2]
## [1,] 0.7056 0.134
## [2,] 0.1340 0.640
##
##  Standardised doses (skeleton):
##          1        2.5          5         10         15         20
## -5.5214609 -4.6051702 -3.9120230 -3.2188758 -2.8134107 -2.5257286
##         25         30         40         50         75        100
## -2.3025851 -2.1202635 -1.8325815 -1.6094379 -1.2039728 -0.9162907
##        150        200        250
## -0.5108256 -0.2231436  0.0000000
##
##  Unmodified (unconstrained) CRM used
##
##  Loss function given intervals of toxicity used to select next dose.
##  Loss function:
##    (0,0.2] (0.2,0.35] (0.35,0.6]    (0.6,1]
##          1          0          1          2
##
##  Toxicities observed:
##             Doses
##              1 2.5 5 10 15 20 25 30 40 50 75 100 150 200 250
##   n          3   4 5  4  0  0  2  0  0  0  0   0   0   0   0
##   Toxicities 0   0 0  0  0  0  2  0  0  0  0   0   0   0   0
##
##  Posterior estimates of toxicity:
##         Doses
##                 1    2.5      5     10    15    20    25    30    40    50
##   Mean   0.01150 0.0293 0.0608 0.1260 0.190 0.250 0.306 0.358 0.447 0.520
##   SD     0.01790 0.0345 0.0570 0.0899 0.112 0.126 0.134 0.139 0.143 0.142
##   Median 0.00527 0.0175 0.0419 0.1020 0.165 0.229 0.288 0.342 0.435 0.525
##         Doses
##             75   100   150    200    250
##   Mean   0.648 0.728 0.817 0.8640 0.8920
##   SD     0.134 0.125 0.108 0.0959 0.0865
```

```
##   Median 0.661 0.747 0.840 0.8890 0.9190
##          Doses
## Quantiles        1     2.5       5     10     15     20     25    30    40
##     2.5%  0.000133 0.00101 0.00445 0.0165 0.0362 0.0617 0.0859 0.118 0.171
##     25%   0.001310 0.00612 0.01930 0.0574 0.1040 0.1620 0.2100 0.259 0.351
##     50%   0.005270 0.01750 0.04190 0.1020 0.1650 0.2290 0.2880 0.342 0.435
##     75%   0.011900 0.03220 0.07540 0.1660 0.2510 0.3270 0.3930 0.451 0.538
##     97.5% 0.063500 0.12200 0.20200 0.3270 0.4370 0.5250 0.5970 0.650 0.729
##          Doses
## Quantiles    50    75   100   150   200   250
##     2.5%  0.228 0.350 0.410 0.532 0.605 0.652
##     25%   0.422 0.570 0.667 0.773 0.832 0.868
##     50%   0.525 0.661 0.747 0.840 0.889 0.919
##     75%   0.611 0.739 0.812 0.897 0.933 0.952
##     97.5% 0.786 0.873 0.912 0.955 0.974 0.983
##
##  Posterior expected loss at each dose:
##           1   2.5      5    10    15     20     25     30     40     50
## [1,] 0.9995 0.998 0.9745 0.832 0.707 0.6515 0.5645 0.6555 0.9575 1.1765
##          75   100   150   200   250
## [1,] 1.6695 1.838 1.939 1.974 1.983
##
##  Posterior probability of dose being in each toxicity interval                    Doses
## Toxicity intervals      1   2.5      5    10    15     20     25     30
##         (0,0.2]    0.9995 0.998 0.9740 0.809 0.601 0.4380 0.2230 0.1095
##         (0.2,0.35] 0.0005 0.002 0.0255 0.168 0.293 0.3540 0.4585 0.4035
##         (0.35,0.6] 0.0000 0.000 0.0005 0.023 0.106 0.2025 0.2955 0.4280
##         (0.6,1]    0.0000 0.000 0.0000 0.000 0.000 0.0055 0.0230 0.0590
##                   Doses
## Toxicity intervals     40     50     75    100    150    200    250
##         (0,0.2]    0.0440 0.0155 0.0015 0.0015 0.0000 0.0000 0.0000
##         (0.2,0.35] 0.2010 0.1060 0.0240 0.0075 0.0025 0.0015 0.0015
##         (0.35,0.6] 0.5965 0.5960 0.2810 0.1455 0.0560 0.0230 0.0140
##         (0.6,1]    0.1585 0.2825 0.6935 0.8455 0.9415 0.9755 0.9845
##
##  Next recommended dose:  25
```

```
plot(TwoPLogistic.tox.intervals.bcrm)
```

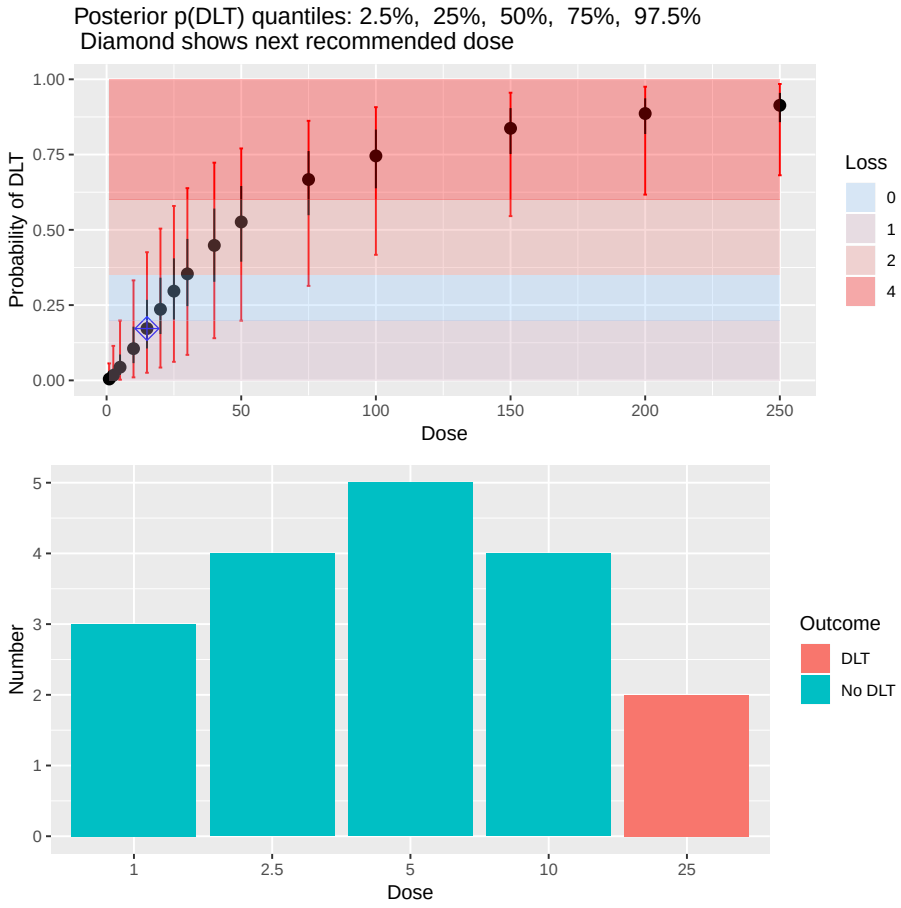


1.7 Example 7

```
## Greater loss associated with overdosing and unacceptable toxicity
## [0, 0.2]=1,  (0.2, 0.35]=0,  (0.35, 0.6]=2,  (0.6, 1]=4
loss2 <- c(1, 0, 2, 4)
TwoPLogistic.tox.intervals.2.bcrm <- bcrm(stop=list(nmax=18),
                                          data=data, sdose=sdose,
                                          dose=dose, ff="logit2",
                                          prior.alpha=list(4, mu, Sigma),
                                          target.tox=target.tox,
                                          constrain=FALSE,
                                          tox.cutpoints=tox.cutpoints,
                                          loss=loss2, method="rjags")

##
##  Stopping:  Reached maximum sample size

print(TwoPLogistic.tox.intervals.2.bcrm)

##
```

```
##  Estimation method:  rjags
##  Target toxicity level:  0.3
##  Model:  Two-parameter logistic
##  Prior:  Log Multivariate Normal

## Mean Vector:

## [1] 2.15 0.52
##
## Variance-Covariance Matrix:
##        [,1]  [,2]
## [1,] 0.7056 0.134
## [2,] 0.1340 0.640
##
##  Standardised doses (skeleton):
##          1        2.5          5         10         15         20
## -5.5214609 -4.6051702 -3.9120230 -3.2188758 -2.8134107 -2.5257286
##         25         30         40         50         75        100
## -2.3025851 -2.1202635 -1.8325815 -1.6094379 -1.2039728 -0.9162907
##        150        200        250
## -0.5108256 -0.2231436  0.0000000
##
##  Unmodified (unconstrained) CRM used
##
##  Loss function given intervals of toxicity used to select next dose.
##  Loss function:
##    (0,0.2] (0.2,0.35] (0.35,0.6]    (0.6,1]
##          1          0          2          4
##
##  Toxicities observed:
##             Doses
##              1 2.5 5 10 15 20 25 30 40 50 75 100 150 200 250
##   n          3   4 5  4  0  0  2  0  0  0  0   0   0   0   0
##   Toxicities 0   0 0  0  0  0  2  0  0  0  0   0   0   0   0
##
##  Posterior estimates of toxicity:
##         Doses
##               1    2.5      5     10    15    20    25    30    40    50
##   Mean   0.0105 0.0276 0.0586 0.1240 0.188 0.249 0.306 0.357 0.445 0.517
##   SD     0.0156 0.0314 0.0528 0.0859 0.110 0.128 0.140 0.148 0.157 0.158
##   Median 0.0047 0.0167 0.0437 0.1060 0.172 0.236 0.296 0.354 0.448 0.526
##         Doses
##             75   100   150    200    250
##   Mean   0.644 0.724 0.815 0.8630 0.8920
##   SD     0.149 0.135 0.111 0.0944 0.0827
##   Median 0.667 0.746 0.837 0.8860 0.9140
##          Doses
## Quantiles        1      2.5       5     10     15     20     25     30
##     2.5%  6.25e-05 0.000492 0.00234 0.0100 0.0253 0.0426 0.0616 0.0847
##     25%   1.34e-03 0.006050 0.01790 0.0572 0.1060 0.1540 0.2020 0.2470
```

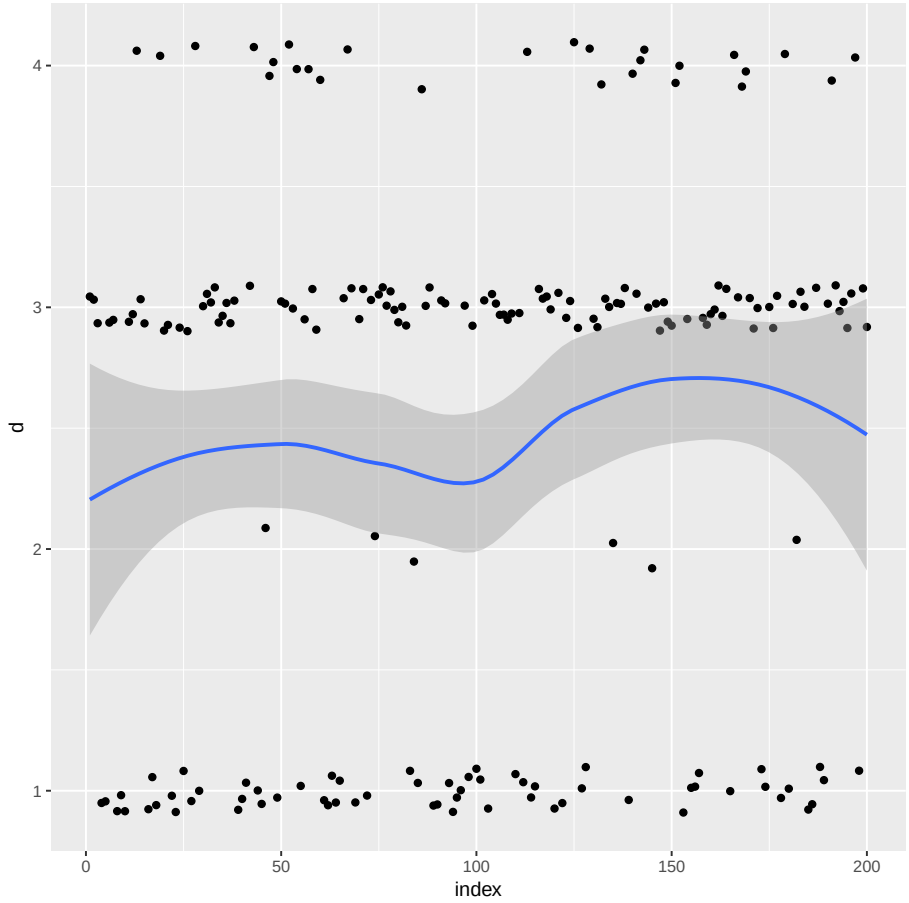
```
##     50%   4.70e-03 0.016700 0.04370 0.1060 0.1720 0.2360 0.2960 0.3540
##     75%   1.26e-02 0.039100 0.08590 0.1780 0.2680 0.3410 0.4050 0.4700
##     97.5% 5.63e-02 0.115000 0.19900 0.3320 0.4260 0.5040 0.5790 0.6380
##          Doses
## Quantiles    40    50    75   100   150   200   250
##     2.5%  0.140 0.198 0.314 0.417 0.546 0.617 0.681
##     25%   0.327 0.394 0.549 0.638 0.752 0.818 0.858
##     50%   0.448 0.526 0.667 0.746 0.837 0.886 0.914
##     75%   0.571 0.646 0.762 0.833 0.904 0.937 0.955
##     97.5% 0.723 0.770 0.862 0.907 0.955 0.975 0.985
##
##  Posterior expected loss at each dose:
##      1   2.5      5   10     15     20   25     30    40    50    75   100
## [1,] 1 0.998 0.9765 0.85 0.7525 0.8475 1.04 1.2825 1.863 2.436 3.179 3.612
##        150   200   250
## [1,] 3.889 3.965 3.984
##
##  Posterior probability of dose being in each toxicity interval                  Doses
## Toxicity intervals 1   2.5      5    10     15     20     25     30     40
##         (0,0.2]    1 0.998 0.9765 0.820 0.5825 0.3755 0.2490 0.1735 0.0630
##         (0.2,0.35] 0 0.002 0.0235 0.165 0.3325 0.3960 0.3710 0.3220 0.2165
##         (0.35,0.6] 0 0.000 0.0000 0.015 0.0850 0.2210 0.3645 0.4545 0.5410
##         (0.6,1]    0 0.000 0.0000 0.000 0.0000 0.0075 0.0155 0.0500 0.1795
##                   Doses
## Toxicity intervals     50     75    100    150    200   250
##         (0,0.2]    0.0280 0.0020 0.0000 0.0000 0.0000 0.000
##         (0.2,0.35] 0.1255 0.0315 0.0115 0.0005 0.0005 0.000
##         (0.35,0.6] 0.4890 0.3445 0.1710 0.0545 0.0165 0.008
##         (0.6,1]    0.3575 0.6220 0.8175 0.9450 0.9830 0.992
##
##  Next recommended dose:  15

plot(TwoPLogistic.tox.intervals.2.bcrm)
```



1.8 Example 8

```
# And a final example, not from the help file, that reveals a bug
# on some runs
dose <- 1:12
p.tox0 <- seq(.04, .5, length.out = length(dose))
target.tox <- 0.33
sdose <- log(dose/25)
mu <- c(2.15,0.52)
Sigma <- rbind(c(0.84^2,0.134),c(0.134,0.80^2))

m <- bcrm::bcrm(stop=list(nmax=18), sdose=sdose,
                dose=dose, ff="logit2",
                prior.alpha=list(4,mu,Sigma),
                target.tox=target.tox,
                constrain=FALSE, quietly=TRUE,
                pointest=0.25, method="rjags",
                truep=p.tox0 * 1.5, simulate=TRUE, nsim=200)
```

```
## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

d <- sapply(m, function(X){
  max(diff(X$data$dose))
})
d <- data.frame(d, index = 1:200)

ggplot(d, aes(index, d)) +
  geom_point(position=position_jitter(height=.1, width=0)) +
  geom_smooth()

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



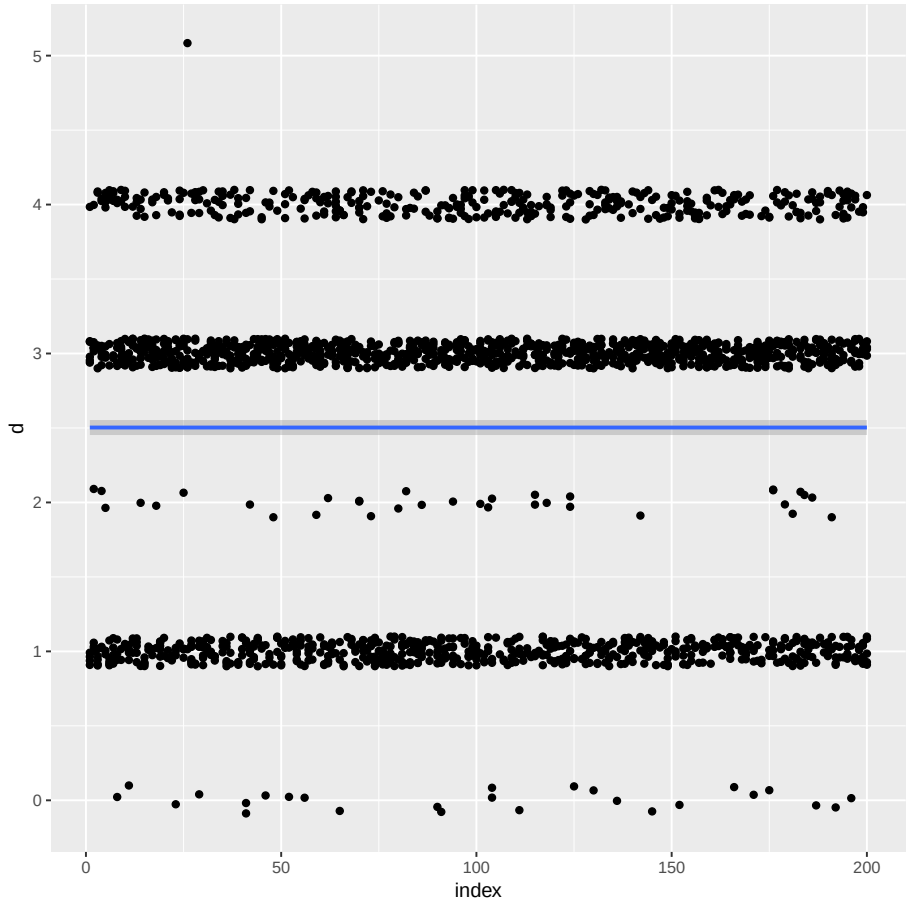
1.9 Example 9

This example is the same as the last one but with more simulated trials. It is to compare the times of a minimally fixed version of 4.7 (4.7.999) with 4.9.X.

```
# And a final example, not from the help file, that reveals a bug
# on some runs
dose <- 1:12
p.tox0 <- seq(.04, .5, length.out = length(dose))
target.tox <- 0.33
sdose <- log(dose/25)
mu <- c(2.15,0.52)
Sigma <- rbind(c(0.84^2,0.134),c(0.134,0.80^2))

m <- bcrm::bcrm(stop=list(nmax=18), sdose=sdose,
                dose=dose, ff="logit2",
                prior.alpha=list(4,mu,Sigma),
                target.tox=target.tox,
                constrain=FALSE, quietly=TRUE,
                pointest=0.25, method="rjags",
                truep=p.tox0 * 1.5, simulate=TRUE,
                nsim=2000)

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete
```

```
## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation
```

```
## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

## Warning in rjags::jags.model(path.model, data = mydata, n.chains
= 2, quiet = TRUE, :  Adaptation incomplete

## NOTE: Stopping adaptation

d <- sapply(m, function(X){
  max(diff(X$data$dose))
})
d <- data.frame(d, index = 1:200)

ggplot(d, aes(index, d)) +
  geom_point(position=position_jitter(height=.1, width=0)) +
  geom_smooth()

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs =
"cs")'
```




```
Sys.time() - startTime

## Time difference of 6.789291 mins
```

2 Information on the R session

Information on the R session, in the interests of reproducibility.

- R version 3.4.4 (2018-03-15), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_GB.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_GB.UTF-8`, `LC_COLLATE=en_GB.UTF-8`, `LC_MONETARY=en_GB.UTF-8`, `LC_MESSAGES=en_GB.UTF-8`, `LC_PAPER=en_GB.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_GB.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 18.04.1 LTS`
- Matrix products: default
- BLAS: `/usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1`
- LAPACK: `/usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: bcrm 0.4.9.2, ggplot2 3.1.0
- Loaded via a namespace (and not attached): assertthat 0.2.0, bindr 0.1.1, bindrcpp 0.2.2, boot 1.3-20, coda 0.19-2, colorspace 1.3-2, compiler 3.4.4, crayon 1.3.4, digest 0.6.18, dplyr 0.7.8, evaluate 0.12, glue 1.3.0, grid 3.4.4, gtable 0.2.0, highr 0.7, knitr 1.20, labeling 0.3, lattice 0.20-38, lazyeval 0.2.1, magrittr 1.5, Matrix 1.2-15, mgcv 1.8-26, munsell 0.5.0, mvtnorm 1.0-8, nlme 3.1-137, pillar 1.3.0, pkgconfig 2.0.2, plyr 1.8.4, purrr 0.2.5, R2WinBUGS 2.1-21, R6 2.3.0, Rcpp 1.0.0, rjags 4-8, rlang 0.3.0.1, scales 1.0.0, stringi 1.2.4, stringr 1.3.1, tibble 1.4.2, tidyselect 0.2.5, tools 3.4.4, withr 2.1.2, yaml 2.2.0