How to use NMA: Introduction

Introduction

This document describes how to use the main functions of `NMA` to run a single network meta-analysis.

Example

First load the required packages.

```
library(NMA)
library(dplyr)
library(purrr)
```

Settings

Define the BUGS parameters for MCMC. This is not necessary, but recommended, because there are default values for these.

```
bugs_params <-
  list(
    PROG = "openBugs",  # which version of BUGS to use to run the MCMC
    N.BURNIN = 10,#00,  # number of steps to throw away
    N.SIMS = 150,#0,    # total number of simulations
    N.CHAINS = 2,       # number of chains
    N.THIN = 1,         # thinning rate
    PAUSE = TRUE)
```

Define the scenario we will use for the analysis.

```
RANDOM <- FALSE               # is this a random effects model?
REFTX <- "ERL/GEF"            # reference treatment
is_bin <- TRUE                # include binary data?
is_med <- TRUE                # include median data?
label_name <- "BC_PFS_mFE"
endpoint <- "PFS"             # which end point, PFS, OS, ...?
analysis_type <- "BC"         # main data tag
```

Read in datasets

The trials data consist of up to 3 separate data frames. A main table, `subData`, and optional tables for median event time and binary data, `subDataMed` and `subDataBin` respectively. Lets read in the each data set separately. In another article we will show how to do this in one function call by including a Reference file in the data folder which contains the meta data of how to read in the study data. If there is no binary or median data used in the NMA then the variables `subDataBin` and `subDataMed` are assigned `NA`.

```
file_name <- paste0(here::here("raw_data"), "/survdata_", endpoint, "_")

subData <-
  read.csv(paste0(file_name, analysis_type, ".csv"),
           header = TRUE,
           as.is = TRUE)

subDataBin <-
  if (is_bin) {
    read.csv(paste0(file_name, "bin.csv"),
             header = TRUE,
             as.is = TRUE)
  } else {NA}

subDataMed <-
  if (is_med) {
    read.csv(paste0(file_name, "med.csv"),
             header = TRUE,
             as.is = TRUE) %>%
      mutate(medR = floor(medR))
  } else {NA}
```

Build model

Now we can create the NMA object to use in the modelling. The workflow is to first create this separately to actually doing the fitting. This then means that we can perform modified fits but we don't have to redo any of the preparatory work.

```
nma_model <-
  new_NMA(subData = subData,
          subDataMed = subDataMed,
          subDataBin = subDataBin,
          bugs_params = bugs_params,
          is_random = RANDOM,
          refTx = REFTX ,
          effectParam = "beta",
          modelParams = "totresdev",
          label = label_name,
          endpoint = endpoint)
#> Warning in if (!is.na(subDataBin)) {: the condition has length > 1 and only the first element will b
#> Warning in if (!is.na(subDataMed)) {: the condition has length > 1 and only the first element will b

nma_model
#> $dat
#> $dat$inits
#> function() {
#>       list(
#>         beta = c(NA, rnorm(nTx - 1, 0, 2)),
#>         sd = 0.1,
#>         alpha = rnorm(nStudies)) %>%
#>         .[param_names]
#>     }
#> <bytecode: 0x0000023254a8c190>
#> <environment: 0x0000023254a93f30>
#>
#> $dat$subData
#>     X                                  study                  base            tx       Lmean        Ls
#> 1   6                 ARCHER 1050 (Wu, 2017)               ERL/GEF           DAC -0.47803580 0.113440
#> 2   9                    CONVINCE (Shi 2017)               CIS+PEM           ICO -0.43078292 0.154626
#> 3  26      CTRI/2015/08/006113 (Patil 2017)             CARBO+PEM       ERL/GEF -0.41551544 0.129116
#> 4  15   CTRI/2016/08/007149 (Noronha 2019b)               ERL/GEF GEF+CARBO+PEM -0.67330000 0.134210
#> 5  22                      ENSURE (Wu 2015)               GEM+CIS       ERL/GEF -1.07880966 0.214485
#> 6   5                   FLAURA (Soria 2018)               ERL/GEF           OSI -0.77652879 0.110238
#> 7   4                   GOAL (Campelo 2018)               ERL/GEF       GEF+OLA -0.28768207 0.186450
#> 8   3                    INCREASE (Li 2018)                   ICO ICO high-dose -0.30110509 0.175849
#> 9  17 JO25567 (JapicCTI-111390) (Seto 2014)               ERL/GEF       ERL+BEV -0.61618614 0.200492
#> 10  2             LUX-Lung 3 (Sequist 2013)               CIS+PEM           AFA -0.71334989 0.143742
#> 11 24               LUX-Lung 6 (Wu 2014a)\n               GEM+CIS           AFA -1.34707365 0.163030
#> 12 25                LUX-Lung 7 (Park 2016)               ERL/GEF           AFA -0.24846136 0.123532
#> 13 20              NCT01017874 (Yang, 2014)               ERL/GEF CIS+PEM+GEF_m -0.18632958 0.344369
#> 14  7              NCT01221077 (Leighl 2017               ERL/GEF       LIN+ERL  0.31188676 0.297077
#> 15 18              NCT01469000 (Yang, 2020)               ERL/GEF       GEF+PEM -0.40047757 0.149945
#> 16 16        NCT01532089 (Stinchcombe 2019)               ERL/GEF       ERL+BEV -0.21072103 0.245707
#> 17 21                 NCT01769066 (Yu 2014)               CIS+PEM CIS+PEM+GEF_m -1.60943791 0.690829
#> 18 28                 NCT01864681 (Li 2019)               ERL/GEF       GEF+MET  0.03922071 0.168174
#> 19  8        NCT01897480 (Scagliotti, 2020)               ERL/GEF       EMI+ERL -0.11653382 0.165827
#> 20 12               NCT02148380 (Han, 2017)               ERL/GEF     CARBO+PEM  1.04982212 0.470613
#> 21 10               NCT02148380 (Han, 2017)               ERL/GEF       ERL/GEF  0.00000000 0.215255
#> 22 11               NCT02148380 (Han, 2017)               ERL/GEF GEF+CARBO+PEM -0.73396918 0.453945
#> 23 13      NEJ005/TCOG0902 (Sugawara, 2015) GEF+CARBO+PEM (Alter) GEF+CARBO+PEM -0.34249031 0.267811
#> 24 27  NEJ009 (UMIN000006340) (Hosomi 2019)               ERL/GEF GEF+CARBO+PEM -0.71334989 0.118258
#> 25 19                   NEJ026 (Saito 2019)               ERL/GEF       ERL+BEV -0.50252682 0.189648
#> 26  1                RELAY (Nakagawa, 2019)               ERL/GEF       RAM+ERL -0.52593926 0.127530
#> 27 14            SWOG S1403 (Goldberg 2018)                   AFA       AFA+CET  0.15700375 0.196751
#> 28 23                 TORCH (Gridelli 2012)               GEM+CIS       ERL/GEF -0.51082562 0.353646
#>
#> $dat$subDataBin
#>                     study    base      tx BinR BinN Btx Bbase Bstudy
#> 1 NCT01039948 (Mok, 2016) ERL/GEF ERL/GEF   34   38   1     1     33
#> 2 NCT01039948 (Mok, 2016) ERL/GEF GEF+FIC   27   33  14     1     33
#>
#> $dat$subDataMed
#>                                     study          base            tx median medN medR mediantx medi
#> 1                                 An 2016       ERL/GEF       ERL/GEF   14.0   45   22        1
#> 2                                 An 2016       ERL/GEF       GEF+PEM   18.0   45   22       17
#> 3  CALGB 30406 (NCT00126581) (Janne 2012)       ERL/GEF ERL+PAC+CARBO   14.1   33   16        1
#> 4  CALGB 30406 (NCT00126581) (Janne 2012)       ERL/GEF ERL+PAC+CARBO   17.2   33   16       10
#> 5      GENOA / NCT02319577 (Genova, 2019)       ERL/GEF       ERL/GEF    9.5   21   10        1
#> 6      GENOA / NCT02319577 (Genova, 2019)       ERL/GEF       VIN+GEF    6.2   23   11       24
#> 7        IFCT-1503 ACE-Lung (Cortot 2019)           AFA           AFA   11.1   59   29        2
#> 8        IFCT-1503 ACE-Lung (Cortot 2019)           AFA       AFA+CET   12.8   59   29        3
#> 9                  NCT01502202 (Lee 2016) CIS+PEM+GEF_m       CIS+PEM    7.8   37   18        5
#> 10                 NCT01502202 (Lee 2016) CIS+PEM+GEF_m CIS+PEM+GEF_m   13.3   39   19        6
#> 11         UMIN000013586 (Kitagawa, 2019)       ERL/GEF       ERL/GEF   15.1   10    5        1
#> 12         UMIN000013586 (Kitagawa, 2019)       ERL/GEF       GEF+BEV    5.4    6    3       11
#>
#> $dat$bugsData
#> $dat$bugsData$mu_beta
#> [1] 0
#>
#> $dat$bugsData$prec_beta
#> [1] 1e-06
```

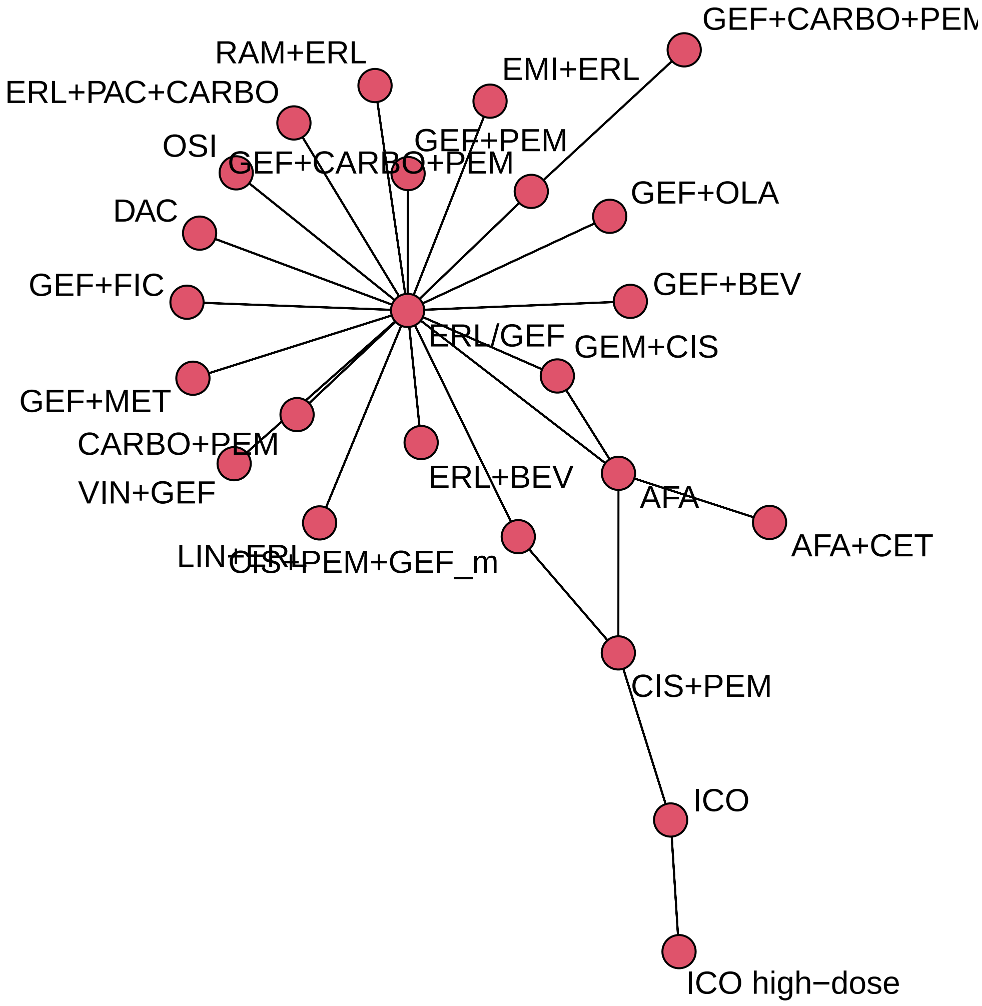
```
#>
#> $dat$bugsData$mu_alpha
#> [1] 0
#>
#> $dat$bugsData$prec_alpha
#> [1] 1e-06
#>
#> $dat$bugsData$Lstudy
#>  [1]  6  9 24 13 20  5  4  3 15  2 22 23 18  7 16 14 19 26  8 10 10 10 11 25 17  1 12 21
#>
#> $dat$bugsData$Ltx
#>  [1]  7 19  1 12  1 22 16 20  9  2  2  2  6 21 17  9  6 15  8  4  1 12 12 12  9 23  3  1
#>
#> $dat$bugsData$Lbase
#>  [1]  1  5  4  1 18  1  1 19  1  5 18  1  1  1  1  1  5  1  1  1  1  1 13  1  1  1  2 18
#>
#> $dat$bugsData$Lmean
#>  [1] -0.47803580 -0.43078292 -0.41551544 -0.67330000 -1.07880966 -0.77652879 -0.28768207 -0.30110509
#> [15] -0.40047757 -0.21072103 -1.60943791  0.03922071 -0.11653382  1.04982212  0.00000000 -0.73396918
#>
#> $dat$bugsData$Lse
#>  [1] 0.1134403 0.1546265 0.1291164 0.1342100 0.2144855 0.1102381 0.1864509 0.1758496 0.2004921 0.143
#> [18] 0.1681749 0.1658275 0.4706135 0.2152550 0.4539455 0.2678118 0.1182584 0.1896482 0.1275307 0.196
#>
#> $dat$bugsData$multi
#>  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
#>
#> $dat$bugsData$LnObs
#> [1] 28
#>
#> $dat$bugsData$nTx
#> [1] 24
#>
#> $dat$bugsData$nStudies
#> [1] 33
#>
#> $dat$bugsData$medianStudy
#>  [1] 27 27 30 30 32 32 29 29 31 31 28 28
#>
#> $dat$bugsData$medianTx
#>  [1]  1 17  1 10  1 24  2  3  5  6  1 11
#>
#> $dat$bugsData$medianBase
#>  [1] 1 1 1 1 1 1 2 2 6 6 1 1
#>
#> $dat$bugsData$Bstudy
#> [1] 33 33
#>
#> $dat$bugsData$Btx
#> [1]  1 14
#>
#> $dat$bugsData$Bbase
#> [1] 1 1
```

```
#>
#> $dat$bugsData$medianN
#>  [1] 45 45 33 33 21 23 59 59 37 39 10  6
#>
#> $dat$bugsData$medianR
#>  [1] 22 22 16 16 10 11 29 29 18 19  5  3
#>
#> $dat$bugsData$median
#>  [1] 14.0 18.0 14.1 17.2  9.5  6.2 11.1 12.8  7.8 13.3 15.1  5.4
#>
#> $dat$bugsData$medianNObs
#> [1] 12
#>
#> $dat$bugsData$Bn
#> [1] 38 33
#>
#> $dat$bugsData$Br
#> [1] 34 27
#>
#> $dat$bugsData$BnObs
#> [1] 2
#>
#>
#> $dat$txList
#>  [1] "ERL/GEF"               "AFA"                   "AFA+CET"               "CARBO+PEM"
#>  [8] "EMI+ERL"               "ERL+BEV"               "ERL+PAC+CARBO"         "GEF+BEV"
#> [15] "GEF+MET"               "GEF+OLA"               "GEF+PEM"               "GEM+CIS"
#> [22] "OSI"                   "RAM+ERL"               "VIN+GEF"
#>
#>
#> $is_med
#> [1] TRUE
#>
#> $is_bin
#> [1] TRUE
#>
#> $bugs_params
#> $bugs_params$PROG
#> [1] "openBugs"
#>
#> $bugs_params$N.BURNIN
#> [1] 10
#>
#> $bugs_params$N.SIMS
#> [1] 150
#>
#> $bugs_params$N.CHAINS
#> [1] 2
#>
#> $bugs_params$N.THIN
#> [1] 1
#>
#> $bugs_params$PAUSE
#> [1] TRUE
#>
#> $bugs_params$run_bugs
#> [1] TRUE
#>
#>
#> $bugs_fn
#> function(...)
#>         R2OpenBUGS::bugs(...)
#> <bytecode: 0x0000023254266d48>
#> <environment: 0x000002325426adb8>
#>
#> $is_random
#> [1] FALSE
#>
#> $refTx
#> [1] "ERL/GEF"
#>
#> $effectParam
#> [1] "beta"
#>
#> $modelParams
#> [1] "totresdev"
#>
#> $label
#> [1] "BC_PFS_mFE"
#>
#> $endpoint
#> [1] "PFS"
#>
#> attr(,"class")
#> [1] "nma"
#> attr(,"CALL")
#> attr(,"CALL")$subData
#> subData
#>
#> attr(,"CALL")$subDataMed
#> subDataMed
#>
#> attr(,"CALL")$subDataBin
#> subDataBin
#>
#> attr(,"CALL")$bugs_params
#> bugs_params
#>
#> attr(,"CALL")$is_random
#> RANDOM
#>
#> attr(,"CALL")$refTx
#> REFTX
#>
#> attr(,"CALL")$effectParam
#> [1] "beta"
```

```
#>
#> attr(,"CALL")$modelParams
#> [1] "totresdev"
#>
#> attr(,"CALL")$label
#> label_name
#>
#> attr(,"CALL")$endpoint
#> endpoint
```

We can view the network graph.

```
library(sna)
plotNetwork(nma_model)
```



Run MCMC

The NMA MCMC function calls the appropriate BUGS model.

```
nma_res <- NMA_run(nma_model)
#> ====== RUNNING BUGS MODEL
#> Warning in dir.create(path = here(folder)): 'C:\Users\Nathan\Documents\R\NMA\output' already exists

nma_res
#> Inference for Bugs model at "C:/Users/Nathan/Documents/R/NMA/inst/FE_med_bin.txt",
#>  2 chains, each with 160 iterations (first 10 discarded)
#>  n.sims = 300 iterations saved
#>            mean    sd  2.5%   25%   50%    75%  97.5% Rhat n.eff
#> beta[2]    -0.3   0.4  -0.8  -0.6  -0.4    0.0    0.8  1.2    22
#> beta[3]    -0.4   0.9  -2.0  -1.7   0.0    0.2    0.5  2.2     3
#> beta[4]     0.5   0.1   0.2   0.4   0.5    0.5    0.7  1.0   300
#> beta[5]     0.0   0.8  -1.1  -0.8   0.1    0.7    1.6  1.3     8
#> beta[6]     1.3   0.9  -0.5   0.4   1.6    2.0    3.0  1.2    12
#> beta[7]    -0.5   0.1  -0.7  -0.6  -0.5   -0.4   -0.2  1.0   300
#> beta[8]    -0.1   0.2  -0.4  -0.2  -0.1    0.0    0.2  1.0   190
#> beta[9]    -0.5   0.1  -0.7  -0.6  -0.5   -0.4   -0.3  1.0   300
#> beta[10]   -0.3   0.5  -1.7  -0.4  -0.3    0.1    0.4  1.4     8
#> beta[11]    0.9   1.0  -1.3   0.4   0.9    1.5    2.7  1.0   300
#> beta[12]   -0.7   0.1  -0.9  -0.7  -0.7   -0.6   -0.5  1.0   300
#> beta[13]   -0.4   0.3  -0.9  -0.5  -0.4   -0.2    0.1  1.0   100
#> beta[14]   -0.3   0.3  -0.9  -0.5  -0.4   -0.1    0.2  1.4     7
#> beta[15]    0.0   0.2  -0.3  -0.1   0.0    0.2    0.3  1.0   160
#> beta[16]   -0.3   0.2  -0.6  -0.4  -0.3   -0.2    0.1  1.0   300
#> beta[17]   -0.7   0.7  -1.8  -1.4  -0.4   -0.3    0.5  2.5     3
#> beta[18]    1.0   0.3   0.6   0.8   1.0    1.2    1.7  1.1    32
#> beta[19]   -0.4   0.8  -1.6  -1.1  -0.4    0.3    1.2  1.3     9
#> beta[20]   -0.7   0.8  -2.0  -1.4  -0.7   -0.1    0.9  1.3    10
#> beta[21]    0.3   0.3  -0.2   0.1   0.3    0.5    0.8  1.0   300
#> beta[22]   -0.8   0.1  -1.0  -0.8  -0.8   -0.7   -0.6  1.0   300
#> beta[23]   -0.5   0.1  -0.8  -0.6  -0.5   -0.4   -0.3  1.1    40
#> beta[24]    0.2   0.4  -0.4   0.0   0.2    0.5    0.9  1.0   300
#> totresdev 765.9 799.5 110.8 141.7 402.7 1347.5 2783.3  1.0    73
#> deviance  781.5 799.3 127.4 156.4 417.2 1362.2 2799.4  1.0    81
#> 
#> For each parameter, n.eff is a crude measure of effective sample size,
#> and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
#> 
#> DIC info (using the rule, pD = Dbar-Dhat)
#> pD = 516.0 and DIC = 1297.0
#> DIC is an estimate of expected predictive error (lower deviance is better).
```

Some useful plots.

```
# # save trace plots to file
# diagnostics(nma_res)

# # save lots of other plots to file
# nma_outputs(nma_res, nma_model)
```

Reconfigure model

It is simple to modify an existing analysis without repeating the previous steps. For example, we can run the NMA for a random effects rather than a fixed effects model version of the same model.

```
nma_model2 <-
  NMA_update(nma_model,
             is_random = TRUE)
#> Warning in if (!is.na(subDataBin)) {: the condition has length > 1 and only the first element will b
#> Warning in if (!is.na(subDataMed)) {: the condition has length > 1 and only the first element will b

nma_res2 <- NMA_run(nma_model2,
                    output_dir = "RE output")
#> ====== RUNNING BUGS MODEL
#> Warning in dir.create(path = here(folder)): 'C:\Users\Nathan\Documents\R\NMA\RE output' already exis

nma_res2
#> Inference for Bugs model at "C:/Users/Nathan/Documents/R/NMA/inst/RE_med_bin.txt",
#>  2 chains, each with 160 iterations (first 10 discarded)
#>  n.sims = 300 iterations saved
#>            mean    sd  2.5%   25%   50%   75%  97.5% Rhat n.eff
#> beta[2]     0.1   4.1  -4.9  -4.4  -0.3   3.9    6.6  4.3     2
#> beta[3]     1.5   1.1  -0.3   0.9   1.2   2.1    3.6  1.6     6
#> beta[4]     0.3   0.5  -0.6  -0.1   0.3   0.7    1.2  1.0    59
#> beta[5]     1.0   5.5  -6.7  -4.4   1.6   6.1    8.7  5.7     2
#> beta[6]    -1.5   1.8  -4.8  -3.1  -0.3  -0.1    0.4  2.9     3
#> beta[7]    -1.8   0.7  -3.1  -2.1  -1.7  -1.3   -0.4  1.2    30
#> beta[8]     0.3   1.1  -1.8  -0.5   0.2   1.0    2.4  2.7     3
#> beta[9]    -0.8   0.6  -2.1  -1.2  -0.8  -0.3    0.1  1.5     6
#> beta[10]   -2.6   1.5  -5.0  -3.8  -2.6  -1.2   -0.3  6.5     2
#> beta[11]    1.0   1.5  -1.8   0.0   1.0   2.1    3.9  1.9     4
#> beta[12]   -0.6   0.9  -2.5  -1.4  -0.5   0.2    0.8  4.1     2
#> beta[13]    0.4   1.6  -2.4  -0.6   0.2   1.1    4.4  2.2     3
#> beta[14]    0.1   0.7  -0.8  -0.4   0.2   0.6    1.6  1.2    12
#> beta[15]    0.0   2.3  -4.5  -0.9  -0.1   1.0    4.3  1.9     4
#> beta[16]    0.8   1.1  -1.4   0.0   0.9   1.5    3.2  2.0     4
#> beta[17]   -0.2   5.9  -7.7  -6.2  -0.3   5.5    8.4  4.8     2
#> beta[18]    1.3   2.6  -1.6  -1.0  -0.2   4.3    5.4  5.0     2
#> beta[19]   -0.5   5.4  -9.3  -4.3   0.1   3.2    7.5  5.0     2
#> beta[20]    0.2   6.7  -9.4  -5.4   0.5   7.0   10.1  5.3     2
#> beta[21]    0.0   1.9  -3.6  -1.6   0.1   1.1    3.9  1.1    26
#> beta[22]   -1.0   1.6  -3.3  -2.6  -1.4   0.7    1.3  5.8     2
#> beta[23]    1.5   1.3  -0.9   0.5   1.4   2.6    3.8  3.2     3
#> beta[24]    1.7   1.2  -0.2   0.7   1.5   2.7    4.0  2.4     3
#> totresdev 643.0 477.0 229.8 310.0 437.7 814.4 1865.6  1.1    24
#> deviance  659.1 478.1 244.9 325.0 453.9 829.1 1889.4  1.1    25
#>
#> For each parameter, n.eff is a crude measure of effective sample size,
#> and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
#>
#> DIC info (using the rule, pD = Dbar-Dhat)
#> pD = 265.3 and DIC = 924.4
#> DIC is an estimate of expected predictive error (lower deviance is better).
```