A Specific Example

2022-10-29

```r
# load trial data
load(file = "trial.data.RData")

# new trial data
trialData <- trial.data$trial.data
trialData
```

```
##    id trial1         Y_1 stay trt2         Y_2
## 1   1     1  -6.32137096    0    3 -0.17530621
## 2   2     1  -1.53076212    0    2 -4.79655697
## 3   3     1  -0.17655805    0    3 -6.55720645
## 4   4     1  -1.47344868    0    3 -4.46756408
## 5   5     1   0.45728234    0    3 -0.17751876
## 6   6     1  -6.67444084    0    3  1.23985708
## 7   7     1  -0.71012625    0    2 -4.09188996
## 8   8     1   1.15127372    0    3 -4.12996752
## 9   9     1   0.43557543    0    2 -6.01805874
## 10 10     1  -3.43455942    0    2 -4.26440916
## 11 11     1  -8.07722541    0    3 -0.55262653
## 12 12     1  -6.22277012    0    2 -4.66579721
## 13 13     1  -4.13911275    0    3 -3.36640483
## 14 14     1  -2.47507705    0    2 -3.50755247
## 15 15     1  -0.15998509    0    2 -6.84469154
## 16 16     1  -0.15850461    0    3 -3.87966549
## 17 17     1  -1.23393061    0    2 -4.91798651
## 18 18     1   0.99445623    0    2 -4.91943354
## 19 19     1  -5.87180865    0    2 -4.84386103
## 20 20     1  -6.79102573    0    2  1.44551224
## 21 21     1   0.04555732    0    2 -8.00763546
## 22 22     1  -5.07788865    0    2 -4.34099465
## 23 23     1  -2.36579902    0    3 -3.09988903
## 24 24     1  -1.06005481    0    2 -2.75697202
## 25 25     1 -11.74384297    0    2  0.08454808
## 26 26     1  -4.47592326    0    3  0.26457137
## 27 27     1  -9.93871808    0    3  0.34913910
## 28 28     1  -4.95448516    0    3 -0.43196793
## 29 29     1 -10.60240567    0    2 -1.90269725
## 30 30     1   1.60888134    0    3 -6.35909104
## 31 31     2  -1.99224486    1    2 -2.19704258
## 32 32     2  -2.57900323    1    2 -4.65787796
## 33 33     2  -4.75311060    0    3 -1.35382804
## 34 34     2   0.64956403    1    2 -1.29681747
## 35 35     2  -3.55932597    0    3 -2.80681026
## 36 36     2  -1.67343770    1    2  0.60357409
```

```
## 37 37    2  -0.88900993    1    2 -1.89520305
## 38 38    2  -2.33640906    1    2 -5.24752262
## 39 39    2  -3.02777350    1    2 -2.05314150
## 40 40    2  -2.84731429    1    2 -4.30015486
## 41 41    2  -2.29182315    1    2 -4.84583911
## 42 42    2  -2.62942119    1    2 -1.87721982
## 43 43    2  -2.96530775    1    2 -1.64340373
## 44 44    2  -2.26367807    1    2 -3.06901521
## 45 45    2   4.04110264    1    2  2.60412312
## 46 46    2  -5.32655982    0    3 -1.00692895
## 47 47    2   1.92174110    1    2 -1.23387246
## 48 48    2  -2.47015442    1    2 -5.05406512
## 49 49    2  -2.06384923    1    2  0.37823933
## 50 50    2  -1.36020062    1    2 -0.39537813
## 51 51    2  -7.49220115    0    3 -3.72808751
## 52 52    2  -0.59184296    1    2 -2.37551084
## 53 53    2  -8.00891700    0    3  0.48599988
## 54 54    2  -4.89518564    0    3 -3.39340065
## 55 55    2   3.36832832    1    2  1.73123462
## 56 56    2   1.33834617    1    2 -0.40055701
## 57 57    2   0.52168821    1    2  0.45206607
## 58 58    2 -11.76820966    0    3 -0.40675317
## 59 59    2   0.89522346    1    2  0.14319961
## 60 60    3  -2.21935601    0    2 -4.32914686
## 61 61    3  -4.74370814   NA   NA          NA
## 62 62    3  -2.60339741    1    3 -0.09164069
## 63 63    3  -6.08981753   NA   NA          NA
## 64 64    3  -0.30374579    1    3 -2.67514332
## 65 65    3  -3.50532444   NA   NA          NA
## 66 66    3  -1.19124156    1    3 -3.58328960
## 67 67    3  -3.33182818   NA   NA          NA
## 68 68    3  -7.58214591   NA   NA          NA
## 69 69    3  -5.59332216   NA   NA          NA
## 70 70    3  -4.89883271   NA   NA          NA
## 71 71    3  -7.16360573   NA   NA          NA
## 72 72    3  -1.02663439    1    3 -2.10973054
## 73 73    3  -7.56610309   NA   NA          NA
## 74 74    3   1.37051972    0    2 -4.21206619
## 75 75    3  -1.19871983    0    2 -4.82845389
## 76 76    3   0.30188007    1    3 -0.25460249
## 77 77    3  -5.43994379   NA   NA          NA
## 78 78    3  -7.14399029   NA   NA          NA
## 79 79    3   0.69891535    0    2 -4.95893649
## 80 80    3  -1.19650204    1    3 -4.57571940
## 81 81    3  -1.23134877    0    2 -6.02702643
## 82 82    3  -5.22086921   NA   NA          NA
## 83 83    3  -7.13617474   NA   NA          NA
## 84 84    3  -7.17575323   NA   NA          NA
## 85 85    3  -3.02805073    0    2 -2.62528649
## 86 86    3  -1.84869655    0    2 -2.76441275
## 87 87    3   0.08796896    1    3 -4.69751008
## 88 88    3   1.56281162    1    3 -6.98891603
## 89 89    3   3.33869842    1    3 -2.99274027
## 90 90    3  -2.03407792    1    3 -1.02267110
```

```
## 91 91    3  -4.23027053   NA   NA          NA
## 92 92    3  -1.69119233    0    2 -0.66510545
```

```
# external control (mean and standard deviation)
historical_data <- cbind(trial.data$Y_k, 0.7713)

# MCMC setting
n_MCMC_chain <- 2
n.adapt <- 4000
MCMC_SAMPLE <- 10000

# coverage rate of credible interval
COVERAGE_RATE <- 0.95

# number of treatment arms
NUM_ARMS <- length(unique(trialData$trt1[!is.na(trialData$trt1)]))

# mean and sd of historical data
Y_k <- historical_data[, 1]
s_k <- historical_data[, 2]

# summary level data calculation
Y_1p <- mean(trialData$Y_1[which(trialData$trt1 == 1)])
s_1p <- sd(trialData$Y_1[which(trialData$trt1 == 1)]) /
  sqrt(length(which(trialData$trt1 == 1)))

Y_1l <- mean(trialData$Y_1[which(trialData$trt1 == 2)])
s_1l <- sd(trialData$Y_1[which(trialData$trt1 == 2)]) /
  sqrt(length(which(trialData$trt1 == 2)))
Y_2l <- mean(trialData$Y_2[which(trialData$trt2 == 2)])
s_2l <- sd(trialData$Y_2[which(trialData$trt2 == 2)]) /
  sqrt(length(which(trialData$trt2 == 2)))

Y_1h <- mean(trialData$Y_1[which(trialData$trt1 == 3)])
s_1h <- sd(trialData$Y_1[which(trialData$trt1 == 3)]) /
  sqrt(length(which(trialData$trt1 == 3)))
Y_2h <- mean(trialData$Y_2[which(trialData$trt2 == 3)])
s_2h <- sd(trialData$Y_2[which(trialData$trt2 == 3)]) /
  sqrt(length(which(trialData$trt2 == 3)))

# calculate the mean treatment effect of group (1p, 2l), (1p, 2h),
# (1l, 2l), (1l, 2h), (1h, 2l), (1h, 2h)
cov_data <- na.omit(trialData)
for (i in c(1:3)) {
  for (j in c(1:2)) {
    index <- intersect(which(cov_data$trt1 == i), which(cov_data$trt2 == j + 1))
    assign(paste0("Y_", i, j + 1), c(mean(cov_data$Y_1[index]), mean(cov_data$Y_2[index])))
  }
}
Y_ij <- rbind(Y_12, Y_13, Y_22, Y_23, Y_32, Y_33)

# threshold of responders and nonresponders
threshold_l <- -3.1
```

```
# calculate the bias
bias_h <- mean(trialData$Y_1[intersect(
  which(!is.na(trialData$trt2)),
  which(trialData$trt1 == 3)
)]) - Y_1h
bias_l_low <- Y_1l - mean(trialData$Y_1[intersect(
  which(trialData$trt2 == 3),
  which(trialData$trt1 == 2)
)])
bias_l_high <- mean(trialData$Y_1[intersect(
  which(trialData$trt2 == 2),
  which(trialData$trt1 == 2)
)]) - Y_1l

# robustification. probability p for the mixture model, here we use p = 0.5 for all
p.exch <- rep(0.5, nrow(historical_data) + 1)

# all control info
y <- c(-1.04, Y_1p) # -1.04 is the mean treatment effect obtained from external control data

# priors for mu
mu_guess <- c(-3.5, -3.5, -3.5)

jag <- rjags::jags.model(
  file = "robust_MAC_snSMART.bugs",
  data = list(
    Ntrials = length(Y_k) + 1,
    NUM_ARMS = NUM_ARMS,
    y = y,
    s = c(s_k, s_1p),
    y_new = Y_ij,
    s_new_norm = c(s_1p, s_1l, s_1h, s_2l, s_2h),
    Prior.cov_ij = c(-1, 1), # priors for covariance
    Nmu = 3, # number of \mu
    Ntau = length(y), # number of sources of control data
    bias_l_high = bias_l_high,
    bias_l_low = bias_l_low,
    bias_high = bias_h,
    Prior.bias_sd = c(s_1l, s_1l, s_1h) / 4,
    Prior.tau = matrix(c(rep(0, length(y)), rep(s_1p, length(y))), ncol = 2) / 2,
    Prior.tau_new = matrix(c(0, 0, s_1l, s_1h), ncol = 2) / 2,
    Prior.mu = matrix(c(
      mu_guess[1], mu_guess[2], mu_guess[3],
      sd(trialData$Y_1[which(trialData$trt1 == 1)]),
      sd(trialData$Y_1[which(trialData$trt1 == 2)]),
      sd(trialData$Y_1[which(trialData$trt1 == 3)])
    ),
    ncol = 2
    ),
    p.exch = p.exch,
    Prior.nex = matrix(c(rep(-4, length(y)), rep(10, length(y))), ncol = 2)
  ),
  n.chains = n_MCMC_chain, n.adapt = n.adapt
```

```r
)
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 11
##    Unobserved stochastic nodes: 29
##    Total graph size: 136
##
## Initializing model
```

```r
posterior_sample_RMS <- rjags::coda.samples(
  jag,
  c("mu", "Z", "tau", "tau_new", "theta", "theta_new", "s", "s_new_norm"),
  MCMC_SAMPLE * 2
)

summary(posterior_sample_RMS)
```

```
##
## Iterations = 4001:24000
## Thinning interval = 1
## Number of chains = 2
## Sample size per chain = 20000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                  Mean     SD  Naive SE Time-series SE
## Z[1]           0.5251 0.4994 0.0024969       0.009962
## Z[2]           0.6002 0.4899 0.0024493       0.014202
## mu[1]         -2.5387 1.6673 0.0083364       0.034827
## mu[2]         -3.1689 0.4499 0.0022494       0.008059
## mu[3]         -2.3277 0.3730 0.0018648       0.007824
## s[1]           0.7713 0.0000 0.0000000       0.000000
## s[2]           0.6801 0.0000 0.0000000       0.000000
## s_new_norm[1]  0.6801 0.0000 0.0000000       0.000000
## s_new_norm[2]  0.6186 0.0000 0.0000000       0.000000
## s_new_norm[3]  0.5237 0.0000 0.0000000       0.000000
## s_new_norm[4]  0.3600 0.0000 0.0000000       0.000000
## s_new_norm[5]  0.4041 0.0000 0.0000000       0.000000
## tau[1]         0.2846 0.2128 0.0010639       0.001196
## tau[2]         0.2868 0.2145 0.0010723       0.002152
## tau_new[1]     0.2421 0.1836 0.0009182       0.002228
## tau_new[2]     0.2061 0.1557 0.0007783       0.001815
## theta[1]      -1.4021 0.9080 0.0045402       0.011251
## theta[2]      -3.2328 0.4343 0.0021714       0.004909
## theta_new[1]  -3.1536 0.5176 0.0025882       0.008443
## theta_new[2]  -2.3375 0.4438 0.0022188       0.008582
## theta_new[3]  -3.1884 0.3468 0.0017339       0.004562
## theta_new[4]  -2.3070 0.2835 0.0014173       0.004486
##
```

```
## 2. Quantiles for each variable:
##
##                     2.5%      25%     50%     75%   97.5%
## Z[1]            0.000000  0.00000  1.0000  1.0000  1.0000
## Z[2]            0.000000  0.00000  1.0000  1.0000  1.0000
## mu[1]          -6.197646 -3.34765 -2.7441 -1.6318  0.5567
## mu[2]          -4.059502 -3.45295 -3.1662 -2.8868 -2.2799
## mu[3]          -3.086928 -2.55847 -2.3192 -2.0865 -1.6142
## s[1]            0.771300  0.77130  0.7713  0.7713  0.7713
## s[2]            0.680066  0.68007  0.6801  0.6801  0.6801
## s_new_norm[1]   0.680066  0.68007  0.6801  0.6801  0.6801
## s_new_norm[2]   0.618574  0.61857  0.6186  0.6186  0.6186
## s_new_norm[3]   0.523683  0.52368  0.5237  0.5237  0.5237
## s_new_norm[4]   0.360030  0.36003  0.3600  0.3600  0.3600
## s_new_norm[5]   0.404107  0.40411  0.4041  0.4041  0.4041
## tau[1]          0.011480  0.11503  0.2413  0.4123  0.7870
## tau[2]          0.011391  0.11673  0.2432  0.4146  0.7995
## tau_new[1]      0.009686  0.09618  0.2046  0.3488  0.6856
## tau_new[2]      0.007938  0.08094  0.1748  0.2993  0.5744
## theta[1]       -3.087294 -2.07704 -1.3800 -0.7473  0.3218
## theta[2]       -4.098218 -3.52426 -3.2243 -2.9319 -2.4060
## theta_new[1]   -4.176940 -3.47063 -3.1577 -2.8415 -2.0976
## theta_new[2]   -3.290348 -2.59044 -2.3243 -2.0627 -1.4822
## theta_new[3]   -3.869774 -3.42363 -3.1866 -2.9565 -2.5087
## theta_new[4]   -2.870059 -2.49563 -2.3053 -2.1158 -1.7583
```