three-component models

Kate Zhang

16/02/2021

## Computation time for three-component models

require(rjags)

## Loading required package: rjags

## Loading required package: coda

## Linked to JAGS 4.3.0

## Loaded modules: basemod,bugs

require(coda)  
require(here)

## Loading required package: here

## here() starts at /Users/wenzhang/Documents/Github/Efficiency-of-marginalising-over-discrete-latent-parameters

require(rstan)

## Loading required package: rstan

## Loading required package: StanHeaders

## Loading required package: ggplot2

## rstan (Version 2.21.1, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling  
## options(mc.cores = parallel::detectCores()).  
## To avoid recompilation of unchanged Stan programs, we recommend calling  
## rstan\_options(auto\_write = TRUE)

##   
## Attaching package: 'rstan'

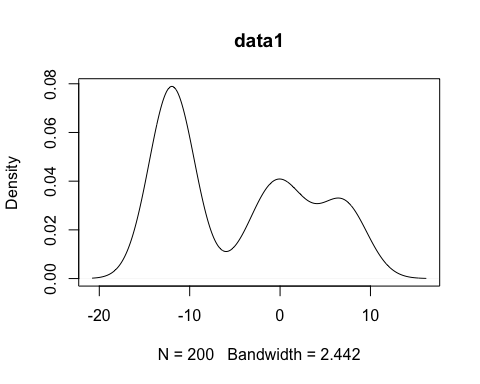
## The following object is masked from 'package:coda':  
##   
## traceplot

load.module("mix")

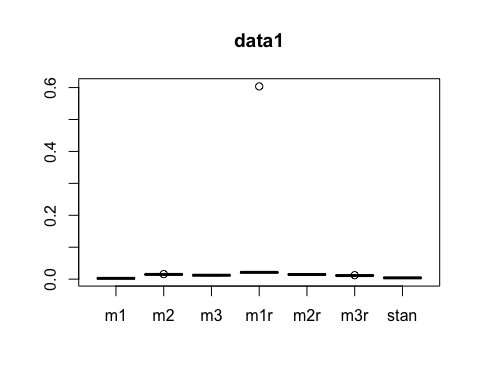
## module mix loaded

NT <- 6 # number of trials

# With data1:

N = 200 mixing\_p = c(0.20, 0.50, 0.3) mu = c(7, -12, 0) prec = c(1, 4, 1/4) sigma = sqrt(1/prec) 

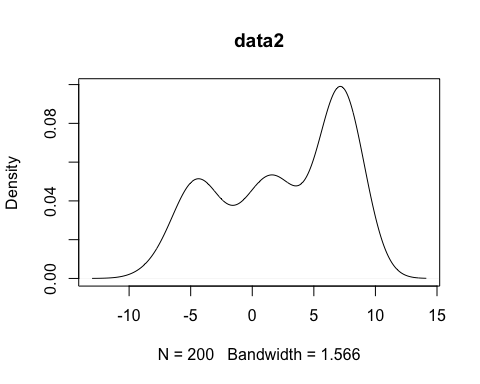
data1 <- c(rowMeans(r1$result[,4,]), rowMeans(r1$resultr[,4,]),rowMeans(r1$resultstan)[3])  
boxplot(t(rbind(r1$result[,4,],r1$resultr[,4,],r1$resultstan[3:3,,drop=FALSE])),main="data1")



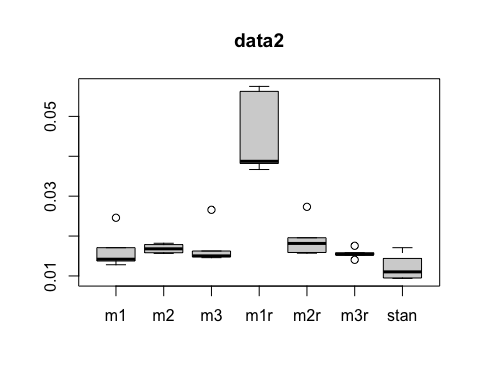
# With data2:

N = 200 mixing\_p = c(0.20, 0.50, 0.3) mu = c(7, -12, 0) prec = c(1, 4, 1/4) sigma = sqrt(1/prec)

x <- as.numeric(unlist(read.table(here("Data", "tdata2.txt"))))  
plot(density(x), main="data2")



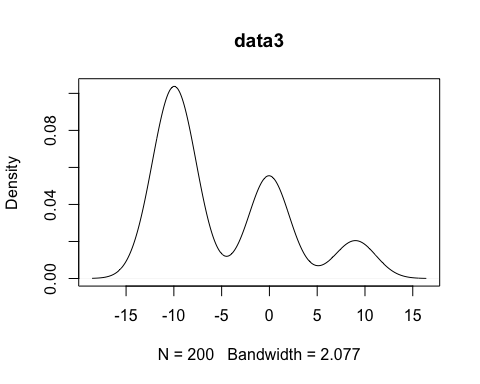
data2 <- c(rowMeans(r2$result[,4,]), rowMeans(r2$resultr[,4,]),rowMeans(r2$resultstan)[3])  
boxplot(t(rbind(r2$result[,4,],r2$resultr[,4,],r2$resultstan[3:3,,drop=FALSE])),main="data2")



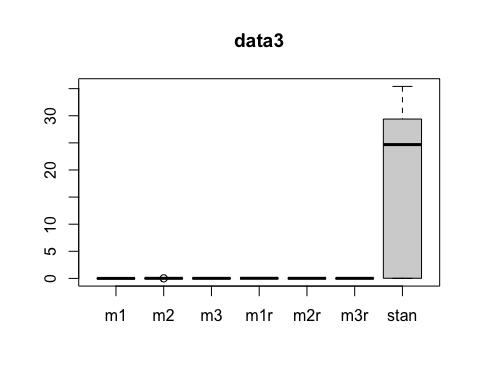
# With data3:

N = 200 mixing\_p = c(0.10, 0.60, 0.3) mu = c(9, -10, 0) prec = c(4, 1, 6) sigma = sqrt(1/prec)

x <- as.numeric(unlist(read.table(here("Data", "tdata3.txt"))))  
plot(density(x), main="data3")



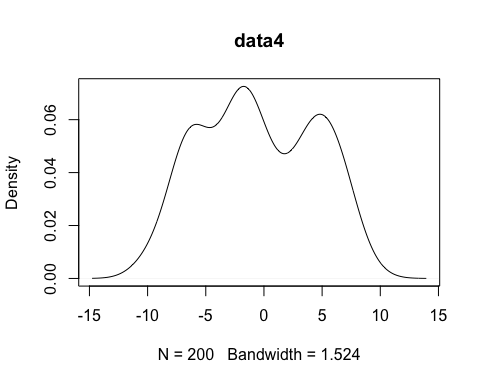
data3 <- c(rowMeans(r3$result[,4,]), rowMeans(r3$resultr[,4,]),rowMeans(r3$resultstan)[3])  
boxplot(t(rbind(r3$result[,4,],r3$resultr[,4,],r3$resultstan[3:3,,drop=FALSE])),main="data3")



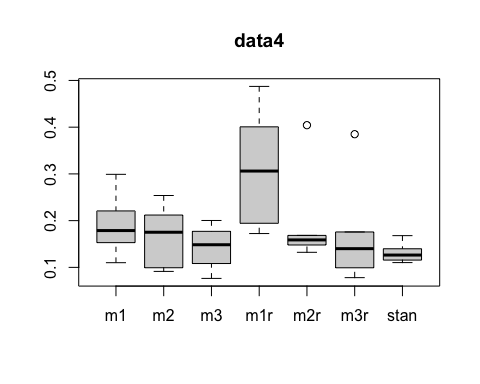
# With data4:

N = 200 mixing\_p = c(0.4, 0.30, 0.4) mu = c(-1, -6, 4) prec = c(1/3, 1/3, 1/4) sigma = sqrt(1/prec)

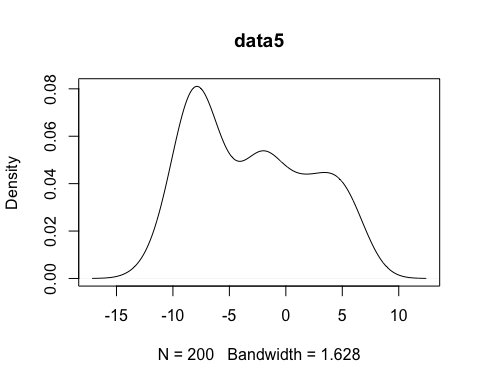
x <- as.numeric(unlist(read.table(here("Data", "tdata4.txt"))))  
plot(density(x), main="data4")



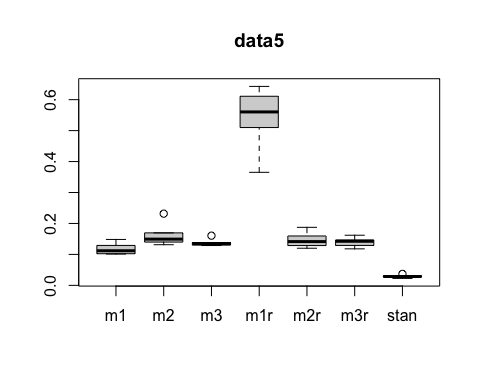
data4 <- c(rowMeans(r4$result[,4,]), rowMeans(r4$resultr[,4,]),rowMeans(r4$resultstan)[3])  
boxplot(t(rbind(r4$result[,4,],r4$resultr[,4,],r4$resultstan[3:3,,drop=FALSE])),main="data4")

 # With data5: N = 200 mixing\_p = c(0.3, 0.50, 0.2) mu = c(-2, -8, 4) prec = c(1/3, 1/3, 1/4) sigma = sqrt(1/prec)

x <- as.numeric(unlist(read.table(here("Data", "tdata5.txt"))))  
plot(density(x), main="data5")

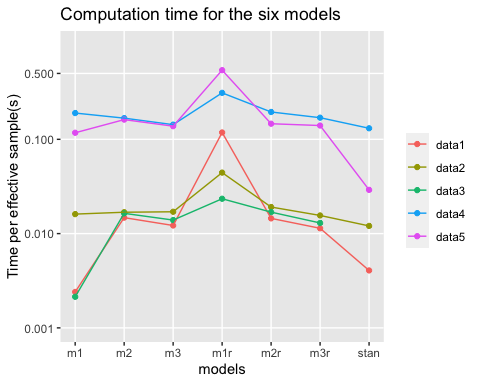


data5 <- c(rowMeans(r5$result[,4,]), rowMeans(r5$resultr[,4,]),rowMeans(r5$resultstan)[3])  
boxplot(t(rbind(r5$result[,4,],r5$resultr[,4,],r5$resultstan[3:3,,drop=FALSE])),main="data5")



## Warning: Removed 1 rows containing missing values (geom\_point).

## Warning: Removed 1 row(s) containing missing values (geom\_path).



## Warning: Removed 1 rows containing missing values (geom\_point).

## Warning: Removed 1 row(s) containing missing values (geom\_path).

