

# BDA6

2022-10-16

```
library(aaltobda)
data("bioassay")
library(rstan)
```

```
## Loading required package: StanHeaders
```

```
## Loading required package: ggplot2
```

```
## rstan (Version 2.21.5, 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)
```

```
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
library(shinystan)
```

```
## Warning: package 'shinystan' was built under R version 4.0.5
```

```
## Loading required package: shiny
```

```
##
## This is shinystan version 2.6.0
```

```
library(bayesplot)
```

```
## This is bayesplot version 1.9.0
```

```
## - Online documentation and vignettes at mc-stan.org/bayesplot
```

```
## - bayesplot theme set to bayesplot::theme_default()
```

```
## * Does _not_ affect other ggplot2 plots
```

```
## * See ?bayesplot_theme_set for details on theme setting
```

```
writeLines(readLines("bioassay1.stan"))
```

```
## data {
##   int N[4];
##   vector[4] x;
##   int y[4];
## }
## transformed data{
##   vector[2] mu;
##   matrix[2,2] sigma;
##
##   mu[1] = 0;
##   mu[2] = 10;
##
##
##   sigma[1,1] = 2^2; sigma[1,2] = 12;
##   sigma[2,1] = 12; sigma[2,2] = 10^2;
## }
##
##
## parameters {
##   vector[2] theta;
##
##
## }
##
## transformed parameters{
##   vector[4] a;
##   //a[1] = theta[1] + theta[2]*x[1];
##   //a[2] = theta[1] + theta[2]*x[2];
##   //a[3] = theta[1] + theta[2]*x[3];
##   //a[4] = theta[1] + theta[2]*x[4];
##   a = theta[1] + theta[2]*x;
## }
##
##
## model {
##
##   theta ~ multi_normal(mu,sigma);
##   y ~ binomial_logit(N,a);
##
## }
```

```
bioData <- list(N = bioassay$n, x = bioassay$x, y = bioassay$y)
bioFit <- stan(file="bioassay1.stan",data = bioData)
```

```
## recompiling to avoid crashing R session
```

```
## Trying to compile a simple C file
```

```
## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c
```

```
## clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I
```

```

## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/inc.
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util.
## namespace Eigen {
## ^
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util.
## namespace Eigen {
## ^
## ;
## In file included from <built-in>:1:
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/inc.
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/inclu
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: f
## #include <complex>
## ^~~~~~
## 3 errors generated.
## make: *** [foo.o] Error 1

```

```
monitor(bioFit)
```

```

## Inference for the input samples (4 chains: each with iter = 2000; warmup = 0):
##
##           Q5  Q50  Q95 Mean  SD  Rhat Bulk_ESS Tail_ESS
## theta[1] -0.4  1.0  2.6  1.0 0.9    1    1098    1646
## theta[2]  4.2 10.3 19.5 10.9 4.8    1    1127    1490
## a[1]     -14.9 -7.8 -3.3 -8.4 3.6    1    1324    1550
## a[2]      -4.3 -2.1 -0.7 -2.2 1.1    1    2160    1790
## a[3]      -0.7  0.5  1.8  0.5 0.8    1    1319    1748
## a[4]       3.2  8.5 16.4  9.0 4.1    1    1021    1402
## lp__      -9.4 -6.8 -6.2 -7.2 1.1    1    1209    1614
##
## For each parameter, Bulk_ESS and Tail_ESS are crude measures of
## effective sample size for bulk and tail quantities respectively (an ESS > 100
## per chain is considered good), and Rhat is the potential scale reduction
## factor on rank normalized split chains (at convergence, Rhat <= 1.05).

```

2.

```
s <- summary(bioFit)
s$summary
```

```

##           mean    se_mean      sd      2.5%      25%      50%
## theta[1] 1.025223 0.02792168 0.9140342 -0.5872204 0.35707644 0.9729403
## theta[2] 10.905287 0.14888875 4.7966187  3.5796679  7.36480526 10.2633973
## a[1]     -8.353323 0.10582586 3.6336433 -16.8142016 -10.44585983 -7.8059871
## a[2]     -2.246363 0.02641348 1.1333911 -4.9045769 -2.89415067 -2.1217698
## a[3]       0.479959 0.02189386 0.7891241 -0.9427320 -0.07908573 0.4573979
## a[4]       8.986083 0.13278682 4.1281939  2.5045818  5.88520413  8.4977594
## lp__      -7.179800 0.03306137 1.0746667 -10.2316768 -7.56026204 -6.8355966
##           75%      97.5%    n_eff    Rhat

```

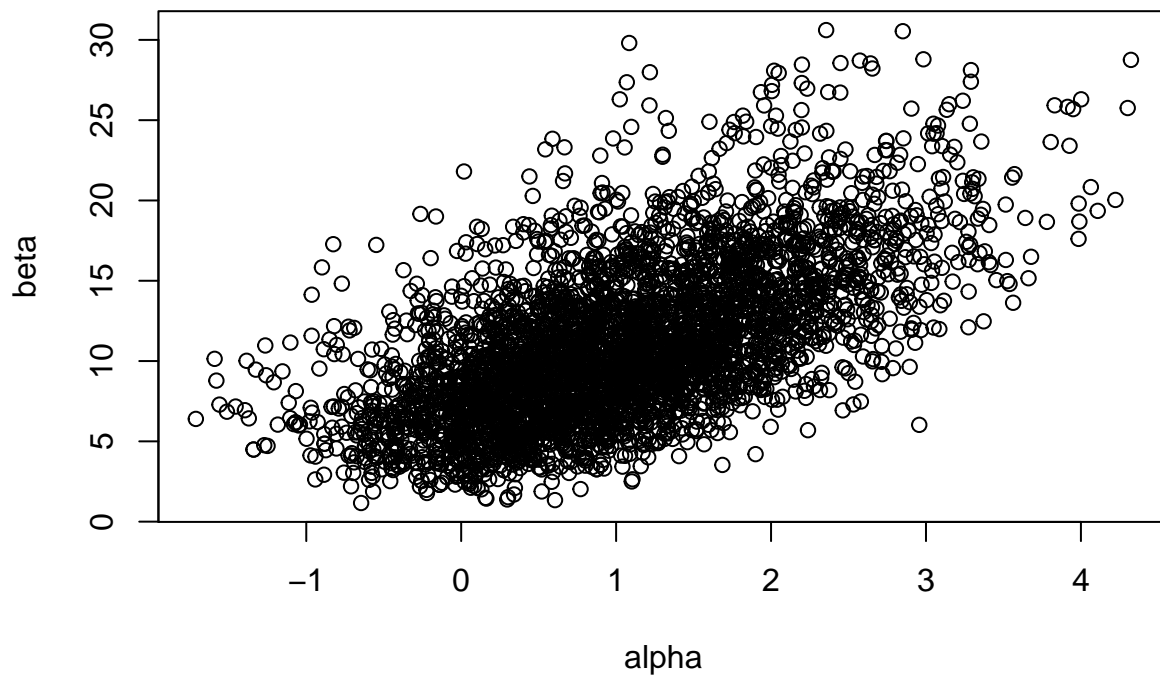
```
## theta[1] 1.6223149 2.9583778 1071.6226 1.002286
## theta[2] 13.6781871 21.8445345 1037.8788 1.003410
## a[1] -5.7049484 -2.7963326 1178.9652 1.002853
## a[2] -1.4375884 -0.3810052 1841.2308 1.000941
## a[3] 0.9993361 2.1166814 1299.1104 1.001306
## a[4] 11.4589419 18.4299457 966.5186 1.003675
## lp_-- -6.4269341 -6.1504544 1056.5884 1.002592
```

Rhat is less than 1.01 for all parameters. In this case  $\theta[1] = \alpha$ , and  $\theta[2] = \beta$ . Values of rhat less than 1.01 can be interpreted as that the chains have converged.

3.

```
frame <- as.data.frame(bioFit)

plot(frame$`theta[1]`, frame$`theta[2]`, xlab = "alpha", ylab = "beta")
```



```
#hist(frame$`theta[1]`, xlab = "alpha")
#hist(frame$`theta[2]`, xlab = "beta")
```

4.

OS: MacOS 12.6

Programming environment: Rstudio

Interface: Rstan

No installation or compilation errors except for some packages for shinystan that could be bypassed by not compiling the newest version and instead using older already compiled version.