BDA - Assignment 6

Anonymous

```r
library(tidyr)
library(rstan)
```

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

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

```
## rstan (Version 2.19.2, 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:tidyr':
##
##     extract
```

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

```
## This is loo version 2.1.0.
## **NOTE: As of version 2.0.0 loo defaults to 1 core but we recommend using as many as possible. Use t
```

```
##
## Attaching package: 'loo'
```

```
## The following object is masked from 'package:rstan':
##
##     loo
```

```r
library(ggplot2)
library(gridExtra)
library(bayesplot)
```

```
## This is bayesplot version 1.7.0
```

```
## - 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
```

```
theme_set(bayesplot::theme_default(base_family = "sans"))
library(shinystan)
```

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

```
##
## This is shinystan version 2.5.0
```

```
source(’stan_utility.R’)
library(aaltobda)
SEED <- 48927 # set random seed for reproducability
```

Problem 1:

The following lines will read the data. Data is summarized into a vectors of the number of animals and the number of deaths in each experiment.

```
data("bioassay")
x = bioassay$x
y = bioassay$y
n = bioassay$n
k = length(n)

mu_alpha <- 0
s_alpha <- 2
mu_beta <- 10
s_beta <- 10
rho <- 0.5
sigma <-  matrix(c(s_alpha^2, rho*s_alpha*s_beta, rho*s_alpha*s_beta, s_beta^2 ), ncol=2)
mu = c(mu_alpha, mu_beta)

d_bin <- list(k = k,
              n = n,
              x = x,
              y = y,
              sigma_0 = sigma,
              mu_0 = mu)
```

The model for the bioassay data in Stan syntax can be read from the Assignment6.stan file which is as follows:

```
writeLines(readLines("Assignment6.stan"))
```

```
## // bioassay model
## data {
##   int<lower=0> k;
##   int n[k];
##   row_vector[k] x;
##   int y[k];
##   vector[2] mu_0;
##   matrix[2, 2] sigma_0;
## }
##
## parameters {
##   row_vector[2] theta;
## }
##
## model {
##   theta ~ multi_normal(mu_0, sigma_0);
##   for (i in 1:k) {
##     y[i] ~ binomial_logit(n[i],theta[1]+theta[2]*x[i]);
##   }
## }
```

Then, in order to fit the model on the data, we use the following command:

```
fit <- stan(file="Assignment6.stan", data = d_bin, seed = SEED)
```

Problem 2: Estimarion of $\hat{R}$

For the $\hat{R}$ convergance analysis, I am using function $\hat{R}$ from package rstan. In the following analysis, theta[1] is α and theta[2] is β.

```
monitor(fit, probs = c(0.1, 0.5, 0.9))
```

```
## 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  0.9  2.4  1.0 0.9     1     1266     1589
## theta[2]  4.1  9.8 19.2 10.5 4.6     1     1186     1597
## lp__     -9.1 -6.8 -6.1 -7.1 1.0     1     1502     1712
##
## 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).
```

```
print(fit)
```

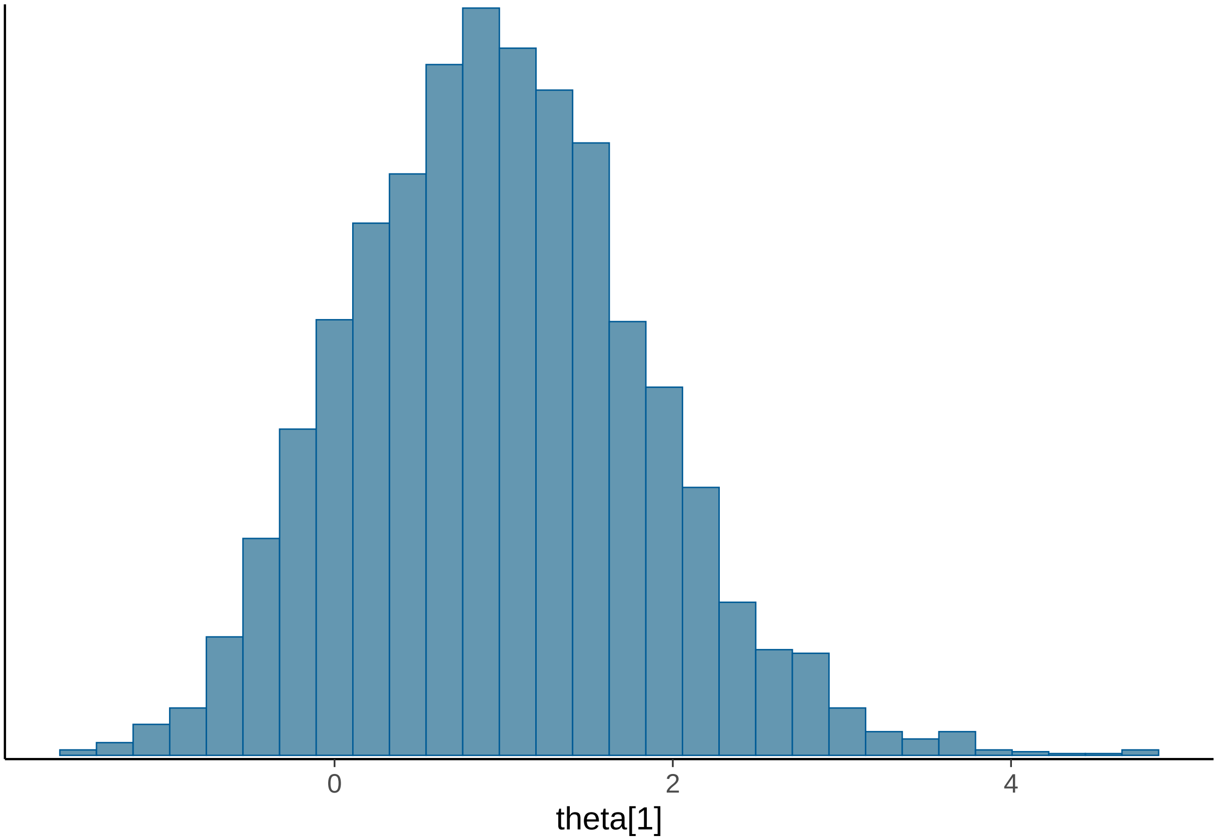
```
## Inference for Stan model: Assignment6.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##           mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## theta[1]  0.98    0.02 0.87 -0.59  0.36  0.95  1.52  2.81  1249    1
## theta[2] 10.49    0.14 4.60  3.41  7.06  9.83 13.21 21.11  1102    1
## lp__     -7.07    0.03 0.99 -9.80 -7.43 -6.77 -6.39 -6.11  1208    1
##
## Samples were drawn using NUTS(diag_e) at Fri Oct 25 20:54:11 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

$\hat{R}$ compares the whithin and between variances of the chains. The within variance tells how much each chain has explored and on the other hand between variance is the total variance when we combine all chains. If we run long enough, the average of within variances and total variance will be close together and the $\hat{R}$ would be close to 1.

As it can be seen in the above analysis the $\hat{R}$ for both α and β is 1, therefore, the chains have probably converged and estimates are reliable.

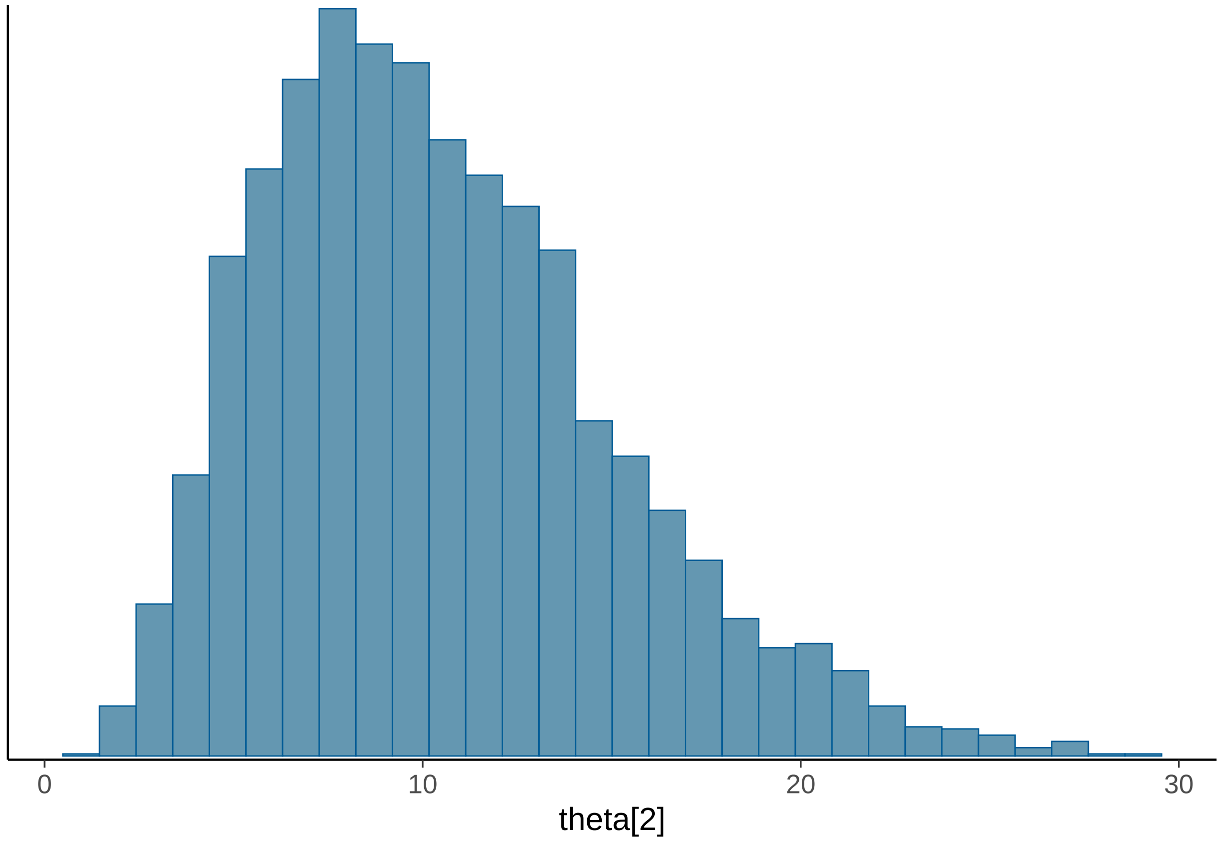
```
draws <- as.data.frame(fit)
mcmc_hist(draws, pars = 'theta[1]')
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```



```
mcmc_hist(draws, pars = 'theta[2]')
```

```
## 'stat_bin()' using 'bins = 30'. Pick better value with 'binwidth'.
```

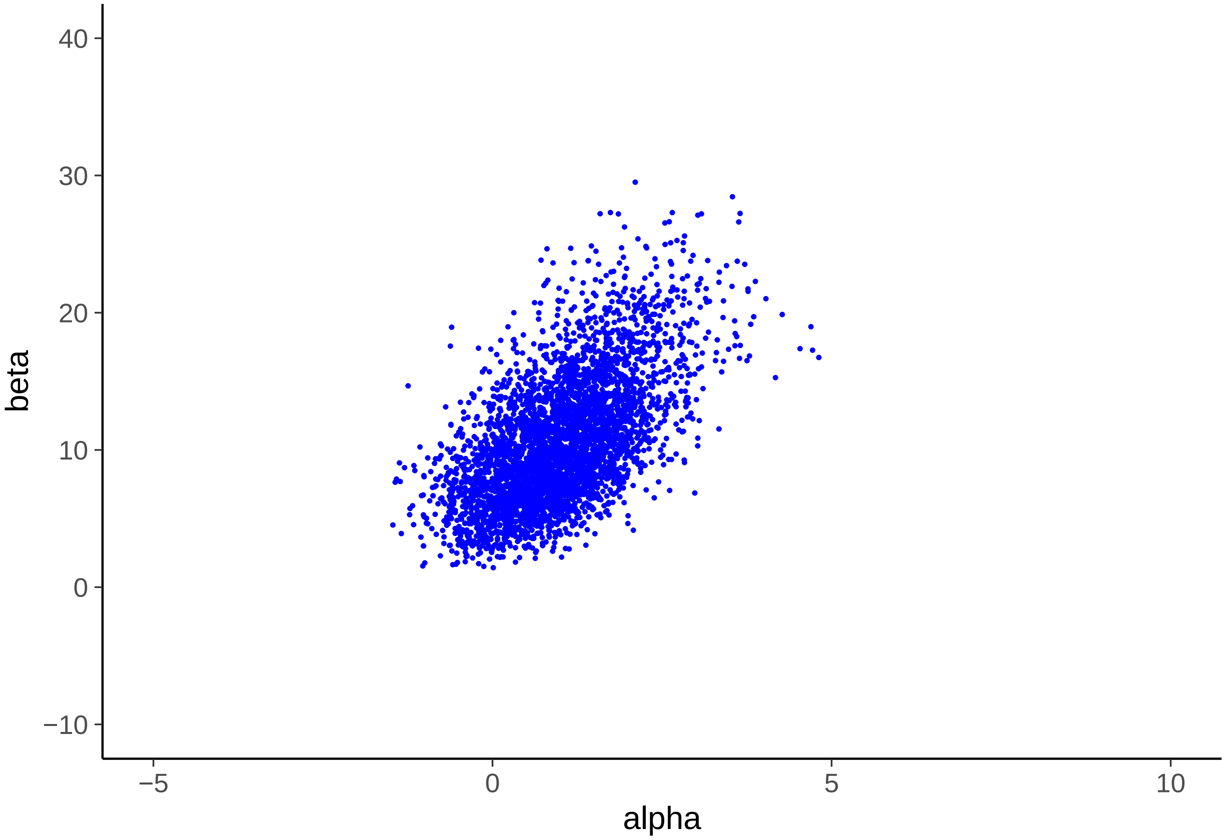


Problem 3: Scatter plot of α and β

In the following figure you can see the scatter plot of draws of α and β for draws implemented above.

```
xl <- c(-5, 10)
yl <- c(-10, 40)

ggplot(data = data.frame(draws$‘theta[1]‘, draws$‘theta[2]‘)) +
  geom_point(aes(draws$‘theta[1]‘, draws$‘theta[2]‘), color = ’blue’, size = 0.3) +
  coord_cartesian(xlim = xl, ylim = yl) +
  labs(x = ’alpha’, y = ’beta’)
```


40
30
20
10
0
−10
beta
−5
0
5
10
alpha