8schools

Chris Parrish

July 9, 2016

Contents

8schools

> *The code in the first section of these notes is from the installation instructions for Stan. It was the first Stan model I ever ran. The code in the second section, using ggplot2 to display some of the features of the fit, is from a Stan case study by Danial C. Furr. The 8 schools study is discussed in Appendix C.2 of BDA3*

references: - Gelman, et al., BDA3, Appendix C.2 - Stan case study by Daniel C. Furr

```
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
library(ggplot2)
```

data

```
schools_dat <- list(J = 8,
                    y = c(28,  8, -3,  7, -1,  1, 18, 12),
                    sigma = c(15, 10, 16, 11,  9, 11, 10, 18))
```

model

8schools.stan

```
data {
  int<lower=0> J;         // number of schools
  real y[J];              // estimated treatment effects
  real<lower=0> sigma[J]; // s.e. of effect estimates
}
parameters {
  real mu;                // population mean
  real<lower=0> tau;      // population sd
  real eta[J];            // school-level errors
}
transformed parameters {
  real theta[J];          // school effects
  for (j in 1:J)
    theta[j] = mu + tau * eta[j];
}
model {
  eta ~ normal(0, 1);
  y ~ normal(theta, sigma);
}
```

fit

```
fit <- stan(file = '8schools.stan', data = schools_dat,
            iter = 1000, chains = 4)
```

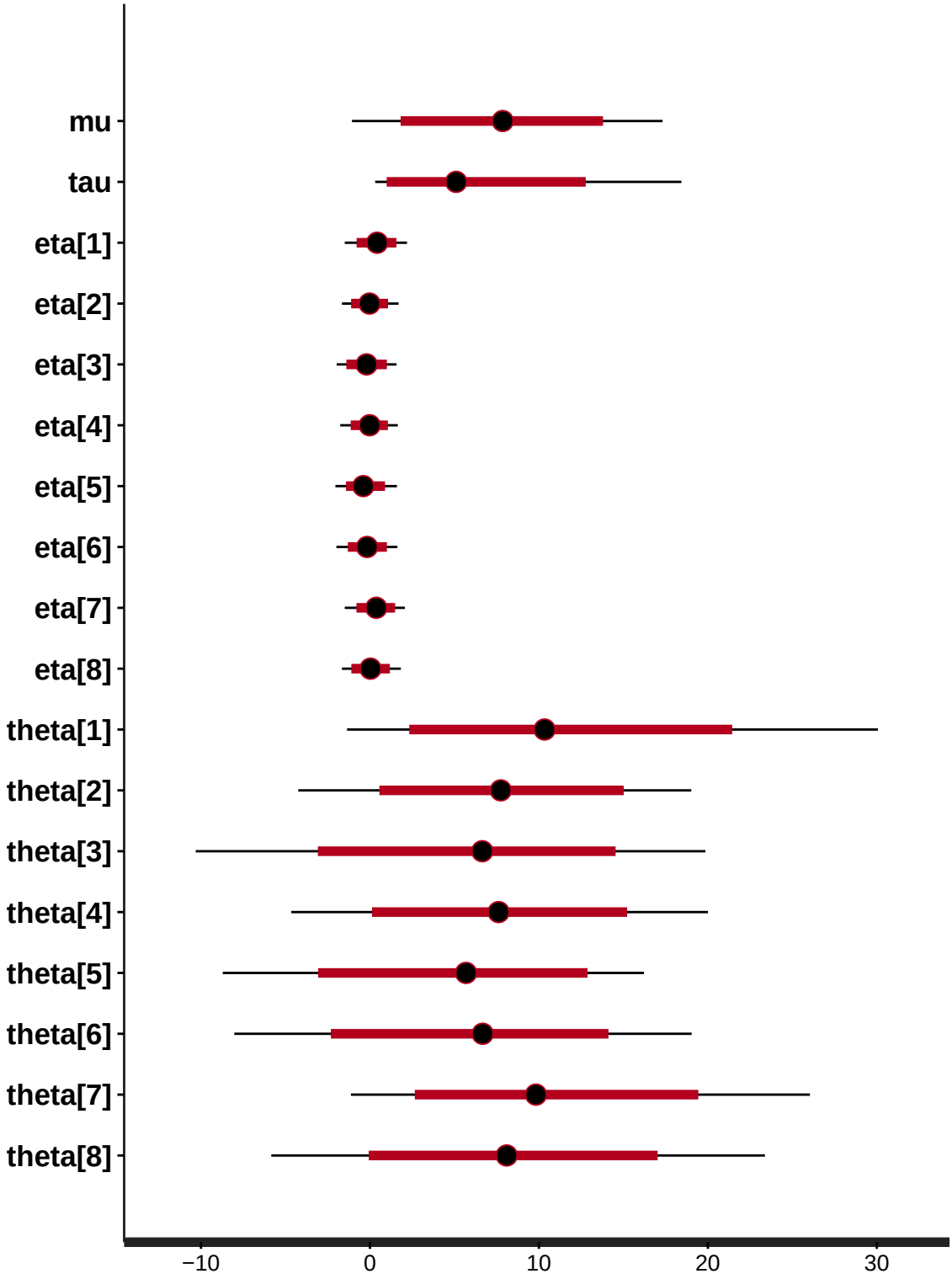
```
## Warning: There were 2 divergent transitions after warmup. Increasing
## adapt_delta above 0.8 may help.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
plot(fit, pars = c("mu", "tau", "eta", "theta"))
```

```
## ci_level: 0.8 (80% intervals)
```

```
## outer_level: 0.95 (95% intervals)
```




```
# pairs(fit)
print(fit)
```

```
## Inference for Stan model: 8schools.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
```

```
##
##           mean se_mean   sd   2.5%   25%   50%   75% 97.5% n_eff Rhat
## mu        7.94    0.14 4.67  -1.06  4.92  7.86 11.11 17.31  1171    1
## tau       6.12    0.17 4.85   0.31  2.43  5.10  8.71 18.41   820    1
## eta[1]    0.41    0.02 0.93  -1.49 -0.20  0.43  1.05  2.19  1503    1
## eta[2]   -0.01    0.02 0.85  -1.66 -0.58 -0.03  0.54  1.69  1551    1
## eta[3]   -0.20    0.02 0.92  -1.97 -0.80 -0.20  0.41  1.58  1641    1
## eta[4]   -0.03    0.02 0.86  -1.76 -0.60 -0.02  0.54  1.65  1500    1
## eta[5]   -0.32    0.02 0.91  -2.04 -0.94 -0.39  0.29  1.59  1367    1
## eta[6]   -0.16    0.02 0.91  -1.98 -0.76 -0.17  0.43  1.63  2000    1
## eta[7]    0.35    0.02 0.90  -1.49 -0.25  0.38  0.96  2.07  1535    1
## eta[8]    0.05    0.02 0.88  -1.65 -0.55  0.03  0.63  1.83  2000    1
## theta[1] 11.38    0.22 8.01  -1.35  5.89 10.35 15.72 30.06  1335    1
## theta[2]  7.73    0.13 5.88  -4.24  4.19  7.75 11.48 19.01  2000    1
## theta[3]  6.24    0.16 7.28 -10.31  2.29  6.65 10.72 19.85  1957    1
## theta[4]  7.68    0.14 6.16  -4.64  3.88  7.62 11.56 20.00  2000    1
## theta[5]  5.27    0.14 6.33  -8.70  1.74  5.72  9.39 16.25  2000    1
## theta[6]  6.26    0.15 6.70  -8.03  2.40  6.68 10.53 19.04  2000    1
## theta[7] 10.59    0.15 6.80  -1.12  5.99  9.83 14.49 26.03  2000    1
## theta[8]  8.34    0.19 7.18  -5.83  4.12  8.10 12.51 23.36  1455    1
## lp__     -4.79    0.10 2.57 -10.48 -6.31 -4.48 -3.00 -0.55   661    1
##
## Samples were drawn using NUTS(diag_e) at Sat Jul  9 23:59:27 2016.
## 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).
##  The estimated Bayesian Fraction of Missing Information is a measure of
##  the efficiency of the sampler with values close to 1 being ideal.
##  For each chain, these estimates are
##  0.8 0.9 0.9 0.7
```

alternate method

```
#  alternate method
schools_code <- paste(readLines('8schools.stan'), collapse = '\n')
fit1 <- stan(model_code = schools_code, data = schools_dat,
             iter = 1000, chains = 4)
```

```
## Warning: There were 2 divergent transitions after warmup. Increasing
## adapt_delta above 0.8 may help.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

do more

```
#  do more
fit2 <- stan(fit = fit1, data = schools_dat, iter = 10000, chains = 4)
```

```
## Warning: There were 91 divergent transitions after warmup. Increasing
## adapt_delta above 0.8 may help.
```

```
## Warning: Examine the pairs() plot to diagnose sampling problems
```

results

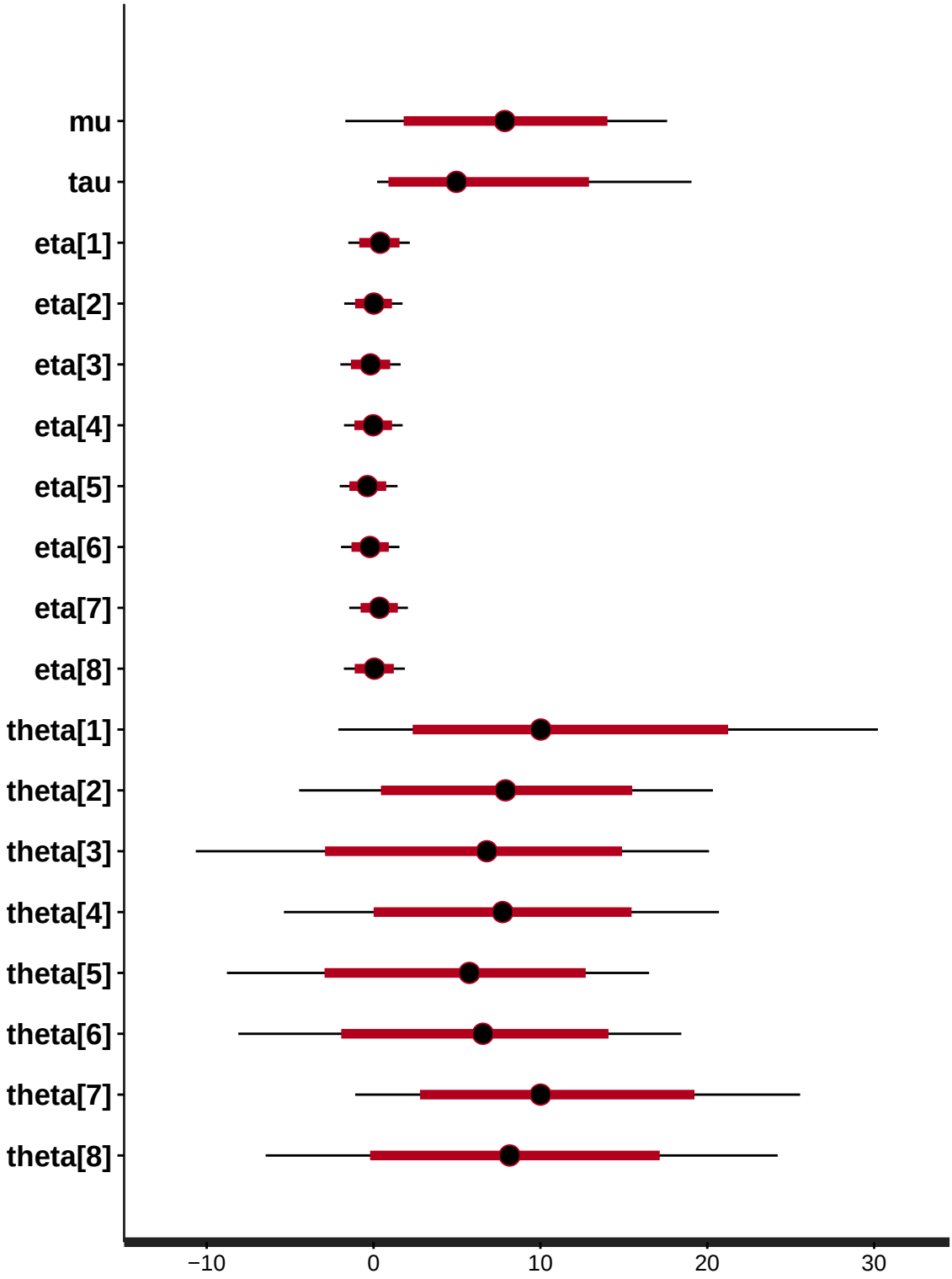
```
#  results
print(fit2)
```

```
## Inference for Stan model: 5f0bdf0403318871b8fe126f239ca318.
## 4 chains, each with iter=10000; warmup=5000; thin=1;
## post-warmup draws per chain=5000, total post-warmup draws=20000.
##
##           mean se_mean   sd   2.5%   25%   50%   75% 97.5% n_eff Rhat
## mu        7.89    0.06 4.83  -1.69  4.72  7.87 11.01 17.59  7562    1
## tau       6.17    0.06 5.13   0.21  2.32  4.97  8.64 19.06  7002    1
## eta[1]    0.37    0.01 0.94  -1.51 -0.24  0.40  1.00  2.17 13587    1
## eta[2]    0.01    0.01 0.88  -1.76 -0.57  0.01  0.58  1.74 13713    1
## eta[3]   -0.18    0.01 0.92  -1.99 -0.81 -0.19  0.44  1.63 14523    1
## eta[4]   -0.02    0.01 0.88  -1.77 -0.60 -0.02  0.55  1.75 13264    1
## eta[5]   -0.35    0.01 0.87  -2.03 -0.93 -0.37  0.20  1.44 12909    1
## eta[6]   -0.21    0.01 0.88  -1.95 -0.79 -0.22  0.37  1.55 13734    1
## eta[7]    0.35    0.01 0.89  -1.46 -0.21  0.36  0.93  2.06 12893    1
## eta[8]    0.05    0.01 0.92  -1.78 -0.56  0.05  0.66  1.88 15293    1
## theta[1] 11.06    0.08 7.99  -2.11  5.89 10.02 15.06 30.22 10778    1
## theta[2]  7.94    0.05 6.15  -4.46  4.08  7.91 11.70 20.34 16547    1
## theta[3]  6.29    0.07 7.51 -10.66  2.31  6.78 10.91 20.11 13287    1
## theta[4]  7.74    0.05 6.41  -5.37  3.90  7.74 11.65 20.70 14469    1
## theta[5]  5.26    0.05 6.30  -8.79  1.57  5.74  9.51 16.51 14270    1
## theta[6]  6.27    0.05 6.58  -8.10  2.44  6.55 10.49 18.44 15138    1
## theta[7] 10.58    0.06 6.66  -1.10  6.10 10.01 14.44 25.56 12969    1
## theta[8]  8.32    0.06 7.40  -6.47  4.05  8.16 12.36 24.22 13284    1
## lp__     -4.89    0.03 2.61 -10.81 -6.48 -4.63 -3.04 -0.52  6252    1
##
## Samples were drawn using NUTS(diag_e) at Sun Jul 10 00:00:05 2016.
## 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).
##  The estimated Bayesian Fraction of Missing Information is a measure of
##  the efficiency of the sampler with values close to 1 being ideal.
##  For each chain, these estimates are
##  0.9 0.9 0.9 0.8
```

```
plot(fit2, pars = c("mu", "tau", "eta", "theta"))
```

```
## ci_level: 0.8 (80% intervals)
```

```
## outer_level: 0.95 (95% intervals)
```




```
la <- extract(fit2, permuted = TRUE) # return a list of arrays
mu <- la$mu
```

return an array of three dimensions: iterations, chains, parameters

```
### return an array of three dimensions: iterations, chains, parameters
a <- extract(fit2, permuted = FALSE)
```

use S3 functions as.array (or as.matrix) on stanfit objects

```
### use S3 functions as.array (or as.matrix) on stanfit objects
a2 <- as.array(fit2)
m <- as.matrix(fit2)
```

print fit

```
print(fit, digits = 1)
```

```
## Inference for Stan model: 8schools.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##          mean se_mean  sd  2.5%  25%  50%  75% 97.5% n_eff Rhat
## mu        7.9     0.1 4.7  -1.1  4.9  7.9 11.1  17.3  1171    1
## tau       6.1     0.2 4.8   0.3  2.4  5.1  8.7  18.4   820    1
## eta[1]    0.4     0.0 0.9  -1.5 -0.2  0.4  1.0   2.2  1503    1
## eta[2]    0.0     0.0 0.9  -1.7 -0.6  0.0  0.5   1.7  1551    1
## eta[3]   -0.2     0.0 0.9  -2.0 -0.8 -0.2  0.4   1.6  1641    1
## eta[4]    0.0     0.0 0.9  -1.8 -0.6  0.0  0.5   1.6  1500    1
## eta[5]   -0.3     0.0 0.9  -2.0 -0.9 -0.4  0.3   1.6  1367    1
## eta[6]   -0.2     0.0 0.9  -2.0 -0.8 -0.2  0.4   1.6  2000    1
## eta[7]    0.3     0.0 0.9  -1.5 -0.2  0.4  1.0   2.1  1535    1
## eta[8]    0.1     0.0 0.9  -1.7 -0.5  0.0  0.6   1.8  2000    1
## theta[1] 11.4     0.2 8.0  -1.4  5.9 10.3 15.7  30.1  1335    1
## theta[2]  7.7     0.1 5.9  -4.2  4.2  7.7 11.5  19.0  2000    1
## theta[3]  6.2     0.2 7.3 -10.3  2.3  6.7 10.7  19.9  1957    1
## theta[4]  7.7     0.1 6.2  -4.6  3.9  7.6 11.6  20.0  2000    1
## theta[5]  5.3     0.1 6.3  -8.7  1.7  5.7  9.4  16.2  2000    1
## theta[6]  6.3     0.1 6.7  -8.0  2.4  6.7 10.5  19.0  2000    1
## theta[7] 10.6     0.2 6.8  -1.1  6.0  9.8 14.5  26.0  2000    1
## theta[8]  8.3     0.2 7.2  -5.8  4.1  8.1 12.5  23.4  1455    1
## lp__     -4.8     0.1 2.6 -10.5 -6.3 -4.5 -3.0  -0.5   661    1
##
## Samples were drawn using NUTS(diag_e) at Sat Jul  9 23:59:27 2016.
## 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).
##  The estimated Bayesian Fraction of Missing Information is a measure of
##  the efficiency of the sampler with values close to 1 being ideal.
##  For each chain, these estimates are
##  0.8 0.9 0.9 0.7
```

illustrate the fit

summary of fit

This code is from a Stan case study by Daniel C. Furr (see the link at the top of this page).

```
# summary(fit)
# summary(fit)$summary   # same as summary(fit)[[1]]
fit.summary <- as.data.frame(summary(fit)[[1]])
fit.summary$parameter <- factor(rownames(fit.summary), rev(rownames(fit.summary)))
str(fit.summary)
```

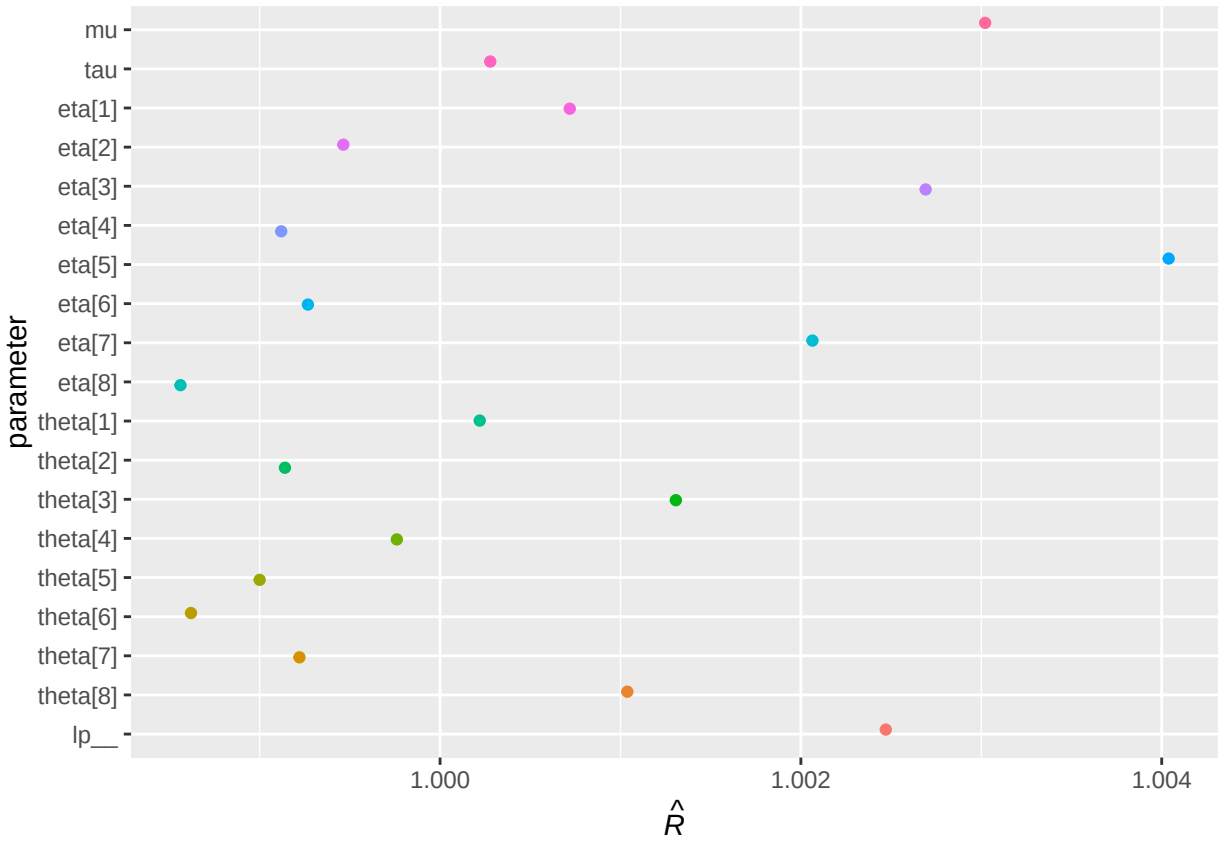
```
## 'data.frame':    19 obs. of  11 variables:
##  $ mean     : num  7.937 6.119 0.408 -0.012 -0.197 ...
##  $ se_mean  : num  0.1364 0.1693 0.0241 0.0216 0.0226 ...
##  $ sd       : num  4.668 4.848 0.934 0.851 0.916 ...
##  $ 2.5%     : num  -1.06 0.31 -1.49 -1.66 -1.97 ...
##  $ 25%      : num  4.921 2.43 -0.198 -0.576 -0.795 ...
##  $ 50%      : num  7.8619 5.1039 0.4279 -0.0257 -0.1958 ...
##  $ 75%      : num  11.106 8.714 1.047 0.536 0.414 ...
##  $ 97.5%    : num  17.31 18.41 2.19 1.69 1.58 ...
##  $ n_eff    : num  1171 820 1503 1551 1641 ...
##  $ Rhat     : num  1.003 1 1.001 0.999 1.003 ...
##  $ parameter: Factor w/ 19 levels "lp__","theta[8]",..: 19 18 17 16 15 14 13 12 11 10 ...
```

```
head(fit.summary)
```

```
##               mean    se_mean        sd       2.5%        25%         50%
## mu      7.93717900 0.13644498 4.6684528 -1.0552365  4.9207580  7.86192183
## tau     6.11946147 0.16927022 4.8483487  0.3098117  2.4304779  5.10390333
## eta[1]  0.40770536 0.02407817 0.9336227 -1.4885327 -0.1980571  0.42786440
## eta[2] -0.01195838 0.02160835 0.8509208 -1.6586925 -0.5761485 -0.02572574
## eta[3] -0.19666033 0.02262244 0.9163340 -1.9698459 -0.7950578 -0.19579059
## eta[4] -0.02791043 0.02221010 0.8602684 -1.7583749 -0.5993322 -0.01861026
##               75%     97.5%     n_eff      Rhat parameter
## mu     11.1060799 17.314609 1170.6601 1.0030213        mu
## tau     8.7141075 18.407946  820.4017 1.0002784       tau
## eta[1]  1.0471336  2.194272 1503.4742 1.0007191    eta[1]
## eta[2]  0.5362687  1.693887 1550.7267 0.9994640    eta[2]
## eta[3]  0.4143969  1.575057 1640.6982 1.0026916    eta[3]
## eta[4]  0.5444139  1.649505 1500.2609 0.9991196    eta[4]
```

show $\hat{R}$

```
ggplot(fit.summary) +
  aes(x = parameter, y = Rhat, color = parameter) +
  geom_jitter(height = 0, width = 0.5, show.legend = FALSE) +
  ylab(expression(hat(italic(R)))) +
  coord_flip()
```



posterior parameters

This code is from a Stan case study by Daniel C. Furr (see the link at the top of this page).

```
wanted_pars <- c("mu", "tau",
                 paste0("eta[", 1:8, "]"),
                 paste0("theta[", 1:8, "]"))
estimated_values <- fit.summary[wanted_pars, c("mean", "2.5%", "97.5%")]
```

data frame

```
# Assemble a data frame to pass to ggplot()
sim_df <- data.frame(parameter = factor(wanted_pars, rev(wanted_pars)),
                     row.names = NULL)
sim_df$middle <- estimated_values[, "mean"]
sim_df$lower <- estimated_values[, "2.5%"]
sim_df$upper <- estimated_values[, "97.5%"]
str(sim_df)
```

```
## 'data.frame':    18 obs. of  4 variables:
##  $ parameter: Factor w/ 18 levels "theta[8]","theta[7]",..: 18 17 16 15 14 13 12 11 10 9 ...
##  $ middle   : num  7.937 6.119 0.408 -0.012 -0.197 ...
##  $ lower    : num  -1.06 0.31 -1.49 -1.66 -1.97 ...
##  $ upper    : num  17.31 18.41 2.19 1.69 1.58 ...
```

show posterior parameters

```
# Plot
ggplot(sim_df) +
  aes(x = parameter, y = middle, ymin = lower, ymax = upper) +
  scale_x_discrete() +
  labs(y = "95% confidence interval", x = NULL) +
  geom_abline(intercept = 0, slope = 0, color = "white") +
  geom_linerange(color = "steelblue") +
  geom_point(color = "darkred") +
  theme_gray() +
  coord_flip()
```