STA365 HW2

Ilke Sun

30/03/2021

Task 1

```
exp_data <- read_rds("experimental_data.RDS")
tre_data <- exp_data %>% subset(Z == 1)
pla_data <- exp_data %>% subset(Z == 0)
pop_data <- read_rds("population.RDS")

unique(pop_data$major) # 5 levels
```

```
## [1] 3 5 1 4 2
```

```
unique(pop_data$gender) # 3 levels, 3 x 5 = 15 combinations
```

```
## [1] 2 1 3
```

```
unique(exp_data$major)
```

```
## [1] 2 4 5 3 1
```

```
unique(exp_data$gender) # same levels with pop_data
```

```
## [1] 1 2 3
```

```
range(exp_data$anxiety_before)
```

```
## [1] 10.04295 49.57149
```

```
range(exp_data$anxiety_after)
```

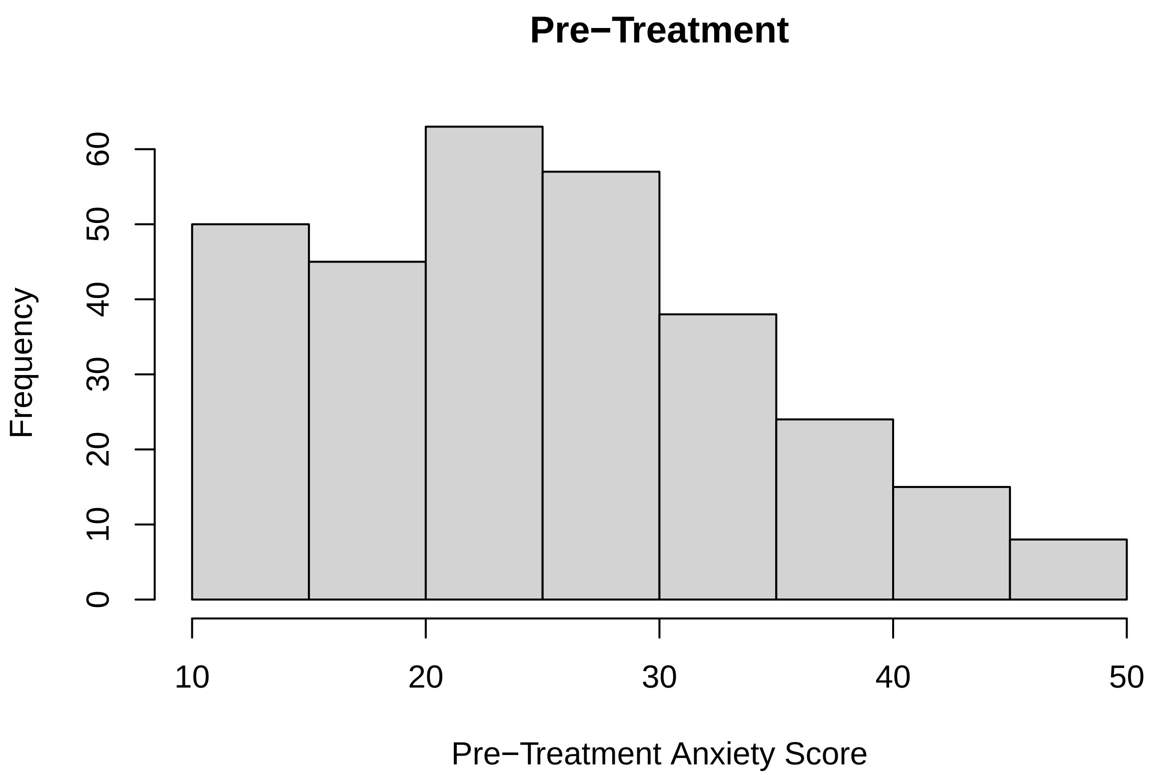
```
## [1] 10.12155 49.10944
```

```
pop_prop <-
  pop_data %>%
  count(major, gender) %>%
  mutate(prop = n/nrow(pop_data))
kable(pop_prop)
```

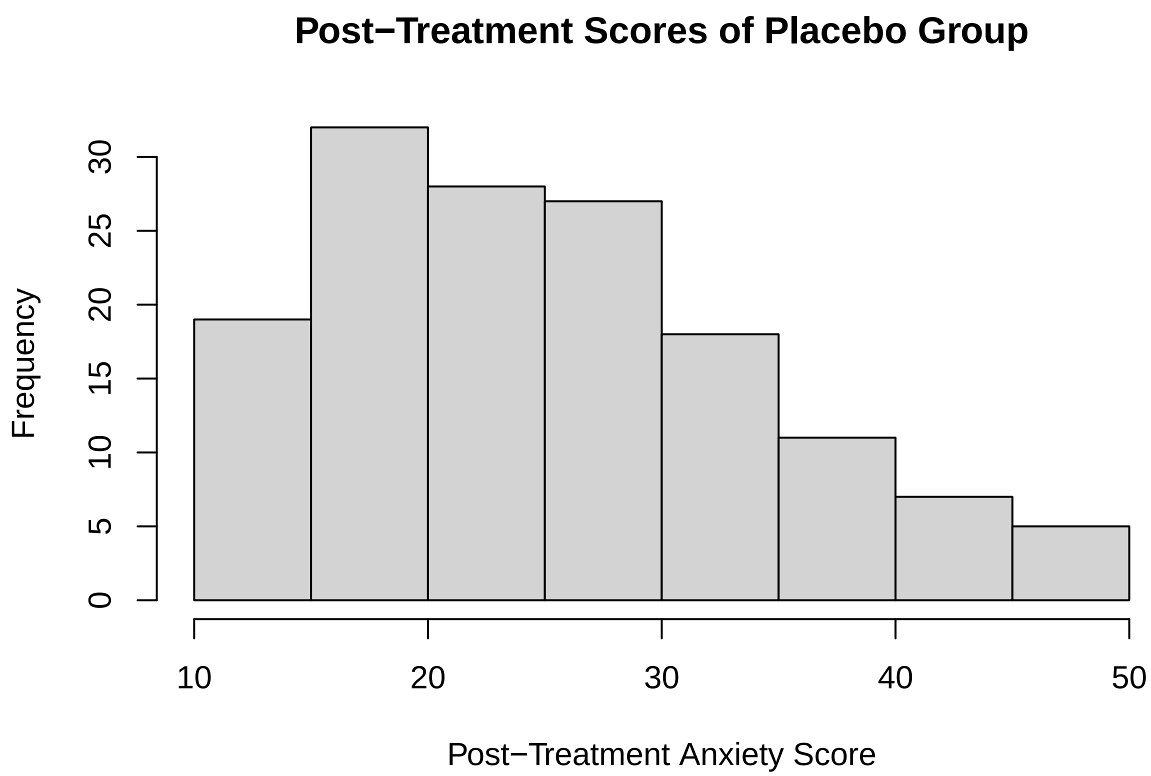
major	gender	n	prop
1	1	265	0.0593239
1	2	225	0.0503694
1	3	14	0.0031341
2	1	268	0.0599955
2	2	235	0.0526080
2	3	12	0.0026864
3	1	428	0.0958137

major	gender	n	prop
3	2	531	0.1188717
3	3	25	0.0055966
4	1	485	0.1085740
4	2	478	0.1070069
4	3	23	0.0051489
5	1	714	0.1598388
5	2	723	0.1618536
5	3	41	0.0091784

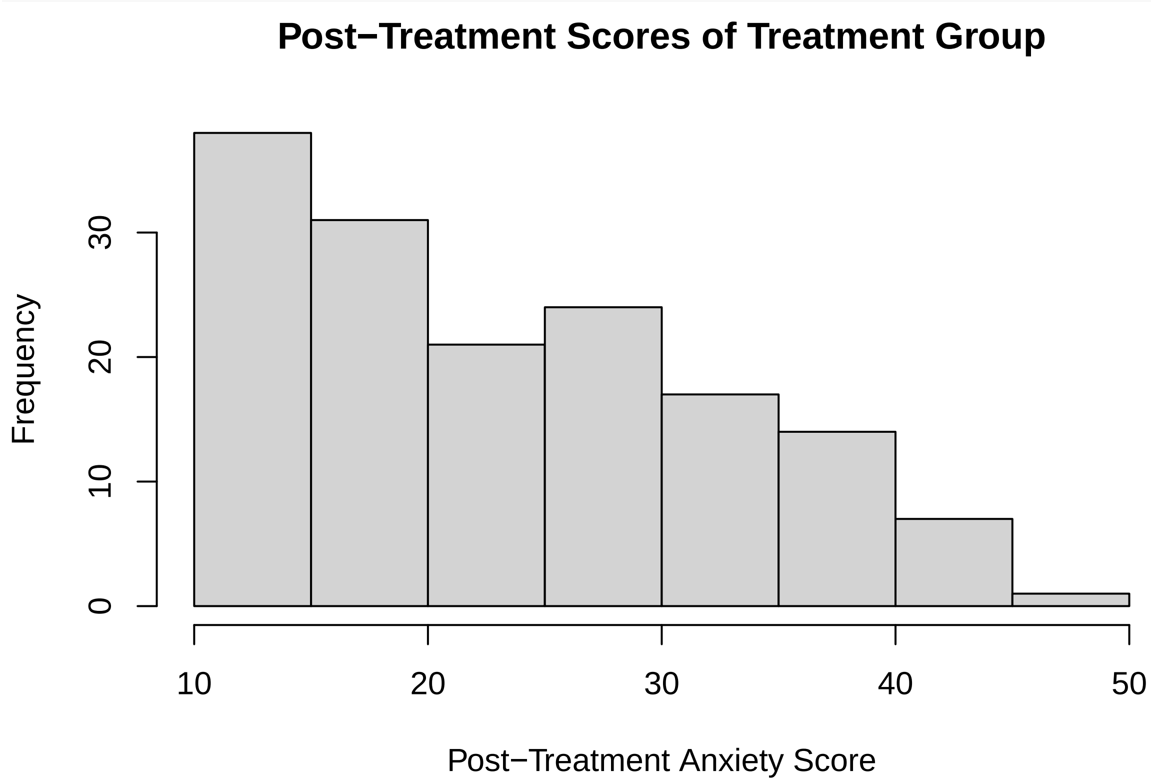
```
hist(exp_data$anxiety_before,
     xlab = "Pre-Treatment Anxiety Score",
     main = "Pre-Treatment")
```




```
hist(pla_data$anxiety_after,
     xlab = "Post-Treatment Anxiety Score",
     main = "Post-Treatment Scores of Placebo Group")
```

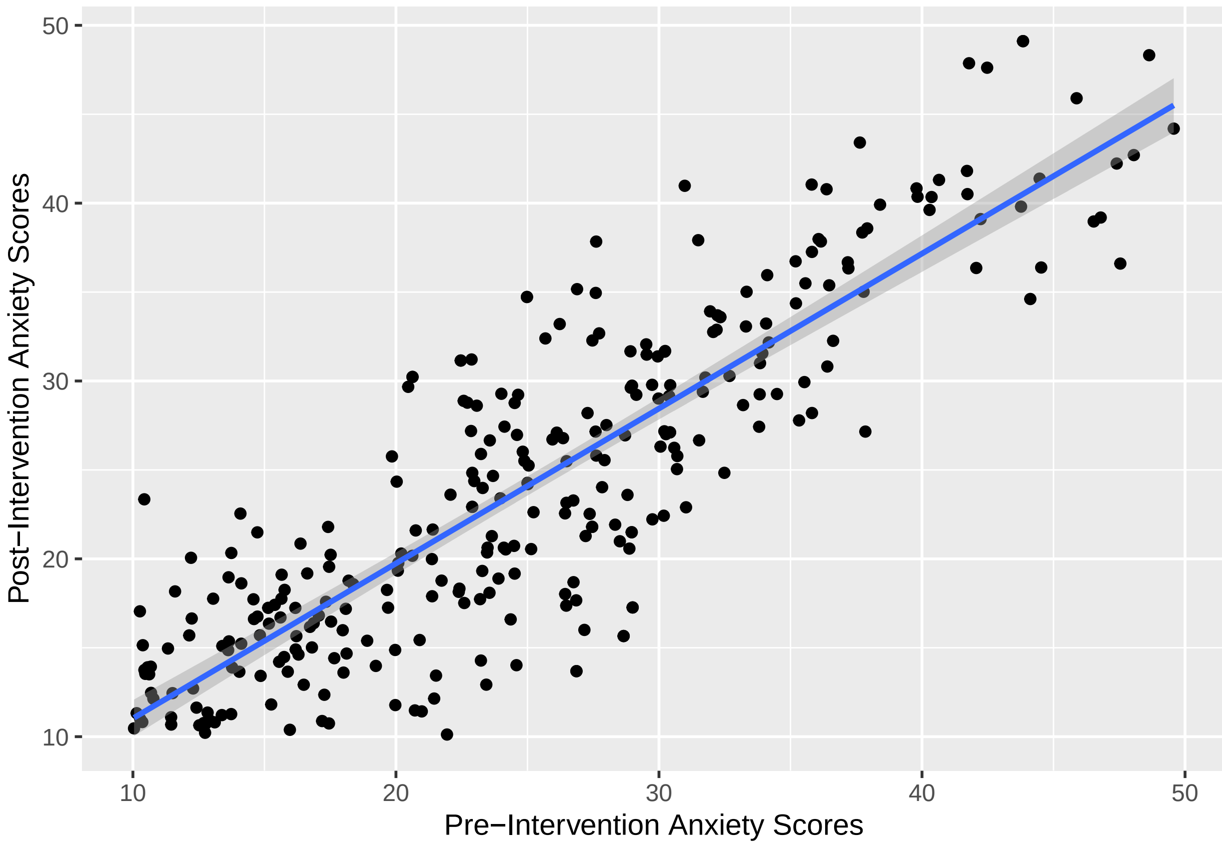



```r
rm(pla_data)
hist(tre_data$anxiety_after,
     xlab = "Post-Treatment Anxiety Score",
     main = "Post-Treatment Scores of Treatment Group")
```



```r
rm(tre_data)
exp_data %>%
  ggplot(aes(x = anxiety_before, y = anxiety_after)) +
  geom_point() + geom_smooth(method = "lm") +
  xlab("Pre-Intervention Anxiety Scores") +
  ylab("Post-Intervention Anxiety Scores")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Pre Anxiety Model

$$
\begin{gathered}
PreAnxietyScore = \mu + u^{\text{gender}}_{\text{gender}[i]} + u^{\text{major}}_{\text{major}[i]} \\
u^{\text{gender}} \sim N(0, \tau^2_{\text{gender}}) \\
u^{\text{major}} \sim N(0, \tau^2_{\text{major}})
\end{gathered}
$$

```r
pre_mod <- cmdstan_model("pre.stan")
```

```
## Model executable is up to date!
```

```r
pre_mod$print()
```

```
## data {
##   int<lower=0> N;
##   real lower_bound;
##   real upper_bound;
##   int<lower = 0> J_major;
```

```
##   int<lower = 0 > J_gender;
##   vector<lower = lower_bound, upper = upper_bound>[N]anxiety_before;
##   int<lower = 1, upper = J_major> major[N];
##   int<lower = 1, upper = J_gender> gender[N];
##   int<lower=0, upper=1> only_prior;
## }
##
## parameters {
##   real mu;
##   real<lower=0> tau_major;
##   vector<multiplier = tau_major>[J_major] u_major;
##   real<lower=0> tau_gender;
##   vector<multiplier = tau_gender>[J_gender] u_gender;
##   real<lower=0> sigma;
## }
##
## transformed parameters {
##   vector [N]eq = mu + u_gender[gender] + u_major[major];
## }
##
## model {
##   mu ~ normal(0, 5);
##   sigma ~ normal(0, 5);
##   u_major ~ normal(0, tau_major);
##   tau_major ~ normal(0, 5);
##   u_gender ~ normal(0, tau_gender);
##   tau_gender ~ normal(0, 5);
##
##   target += normal_lpdf(anxiety_before | eq, sigma) - log_diff_exp(normal_lcdf(
##     upper_bound | eq, sigma), normal_lcdf(lower_bound | eq, sigma));
##
##   if(only_prior == 0) {
##     anxiety_before ~ normal(eq, sigma);
##   }
## }
##
## generated quantities {
##   vector[N] log_lik;
##   vector[N] anxiety_pred;
##   for (i in 1:N) {
##     log_lik[i] = normal_lpdf(anxiety_before[i]| eq[i], sigma);
##     anxiety_pred[i] = normal_rng(eq[i], sigma);
##   }
## }
```

```r
stan_data <- list(N = nrow(exp_data),
                  J_major = length(unique(exp_data$major)),
                  J_gender = length(unique(exp_data$gender)),
                  anxiety_before = exp_data$anxiety_before,
                  major = exp_data$major,
                  gender = exp_data$gender,
                  lower_bound = 10, upper_bound = 50, only_prior = 1)

pre_fit <- pre_mod$sample(stan_data, parallel_chains = 4, refresh = 0,
```

```r
                             show_messages = F, seed = 666)
```

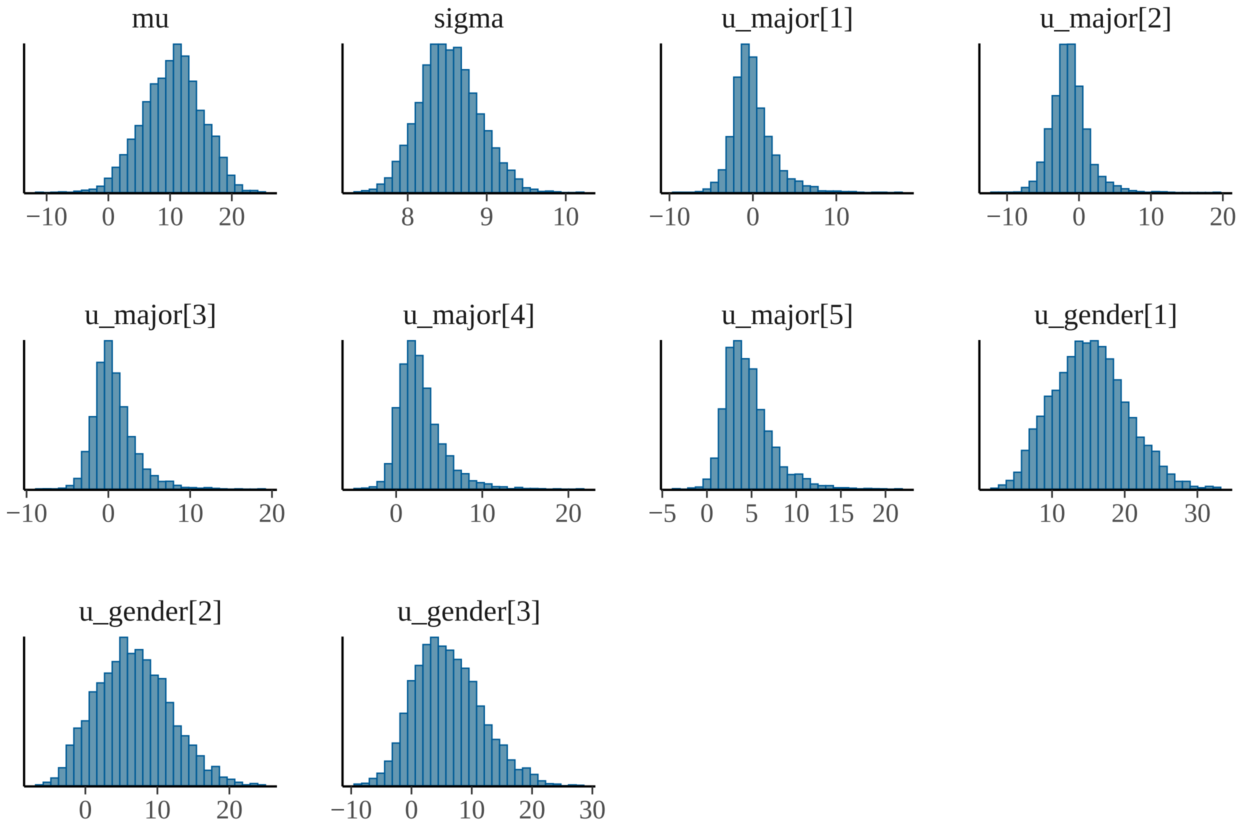
```
## Running MCMC with 4 parallel chains...
##
## Chain 2 finished in 14.9 seconds.
## Chain 1 finished in 15.6 seconds.
## Chain 3 finished in 16.0 seconds.
## Chain 4 finished in 15.9 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 15.6 seconds.
## Total execution time: 16.1 seconds.
```

```r
pre_fit$print(max_rows = 13)
```

```
##     variable     mean   median   sd  mad       q5      q95 rhat ess_bulk
##  lp__        -1070.08 -1069.74 3.04 2.88 -1075.50 -1065.75 1.00     1145
##  mu             10.36    10.59 4.74 4.76     2.32    17.95 1.00     1613
##  tau_major       3.67     3.22 1.90 1.54     1.43     7.32 1.00      984
##  u_major[1]     -0.02    -0.39 2.45 1.89    -3.24     4.52 1.00     1422
##  u_major[2]     -1.36    -1.48 2.52 2.09    -5.06     2.91 1.00     2118
##  u_major[3]      0.67     0.27 2.50 1.99    -2.61     5.36 1.00     1455
##  u_major[4]      2.83     2.39 2.58 2.07    -0.34     7.71 1.00     1292
##  u_major[5]      4.60     4.14 2.62 2.17     1.33     9.70 1.00     1187
##  tau_gender      8.49     8.25 2.96 3.02     4.06    13.66 1.00     1794
##  u_gender[1]    15.24    15.10 5.03 5.06     7.27    24.02 1.00     1267
##  u_gender[2]     6.70     6.54 5.07 5.24    -1.30    15.45 1.00     1186
##  u_gender[3]     6.04     5.59 5.54 5.61    -2.09    15.88 1.00     1581
##  sigma           8.53     8.52 0.37 0.37     7.93     9.17 1.00     3933
##  ess_tail
##      2026
##      2209
##      1469
##      1809
##      2011
##      1872
##      1705
##      1630
##      2013
##      1865
##      1809
##      2367
##      2491
##
##  # showing 13 of 913 rows (change via 'max_rows' argument)
```

```r
mcmc_hist(pre_fit$draws(c("mu", "sigma", "u_major[1]", "u_major[2]",
                          "u_major[3]", "u_major[4]", "u_major[5]",
                          "u_gender[1]", "u_gender[2]", "u_gender[3]")))
```

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




```r
stan_data$only_prior = 0
pre_fit <- pre_mod$sample(stan_data, parallel_chains = 4, refresh = 0,
                          show_messages = F, seed = 666)
```

```
## Running MCMC with 4 parallel chains...
##
## Chain 2 finished in 22.0 seconds.
## Chain 1 finished in 23.1 seconds.
## Chain 3 finished in 23.7 seconds.
## Chain 4 finished in 24.9 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 23.4 seconds.
## Total execution time: 24.9 seconds.
```

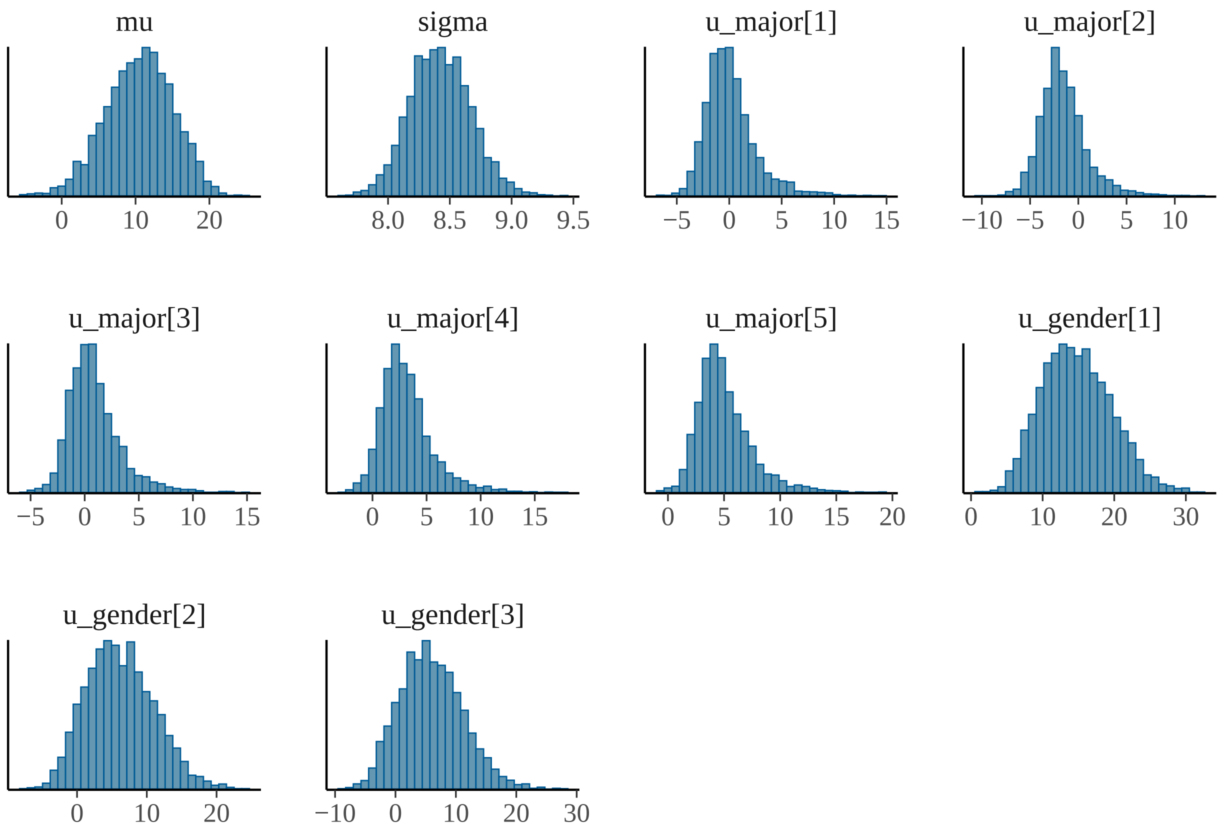
```r
pre_fit$print(max_rows = 13)
```

```
##     variable      mean   median   sd  mad       q5      q95 rhat ess_bulk
##  lp__         -1855.55 -1855.21 2.84 2.74 -1860.76 -1851.51 1.00     1170
##  mu              10.52    10.72 4.53 4.65     2.73    17.71 1.00     1312
##  tau_major        3.90     3.42 1.90 1.46     1.80     7.63 1.00     1094
##  u_major[1]       0.11    -0.24 2.42 1.95    -2.99     4.81 1.00     1456
##  u_major[2]      -1.56    -1.85 2.41 2.02    -4.96     2.95 1.00     1834
##  u_major[3]       0.88     0.50 2.42 1.96    -2.21     5.43 1.00     1381
##  u_major[4]       3.15     2.75 2.42 1.99     0.01     7.80 1.00     1356
##  u_major[5]       5.02     4.60 2.43 1.98     1.94     9.70 1.00     1339
##  tau_gender       8.45     8.24 2.96 3.08     4.06    13.69 1.00     1423
##  u_gender[1]     14.79    14.48 4.83 5.01     7.46    23.11 1.00     1207
##  u_gender[2]      6.22     5.90 4.83 4.98    -1.20    14.52 1.00     1218
```

```
## u_gender[3]      5.95      5.61 5.23 5.28     -2.08     15.02 1.00      1317
## sigma            8.42      8.42 0.25 0.25      8.04      8.84 1.00      3276
## ess_tail
##     1518
##     2058
##     1979
##     1692
##     1808
##     1591
##     1588
##     1629
##     1957
##     1831
##     1831
##     2178
##     2365
##
## # showing 13 of 913 rows (change via 'max_rows' argument)
```

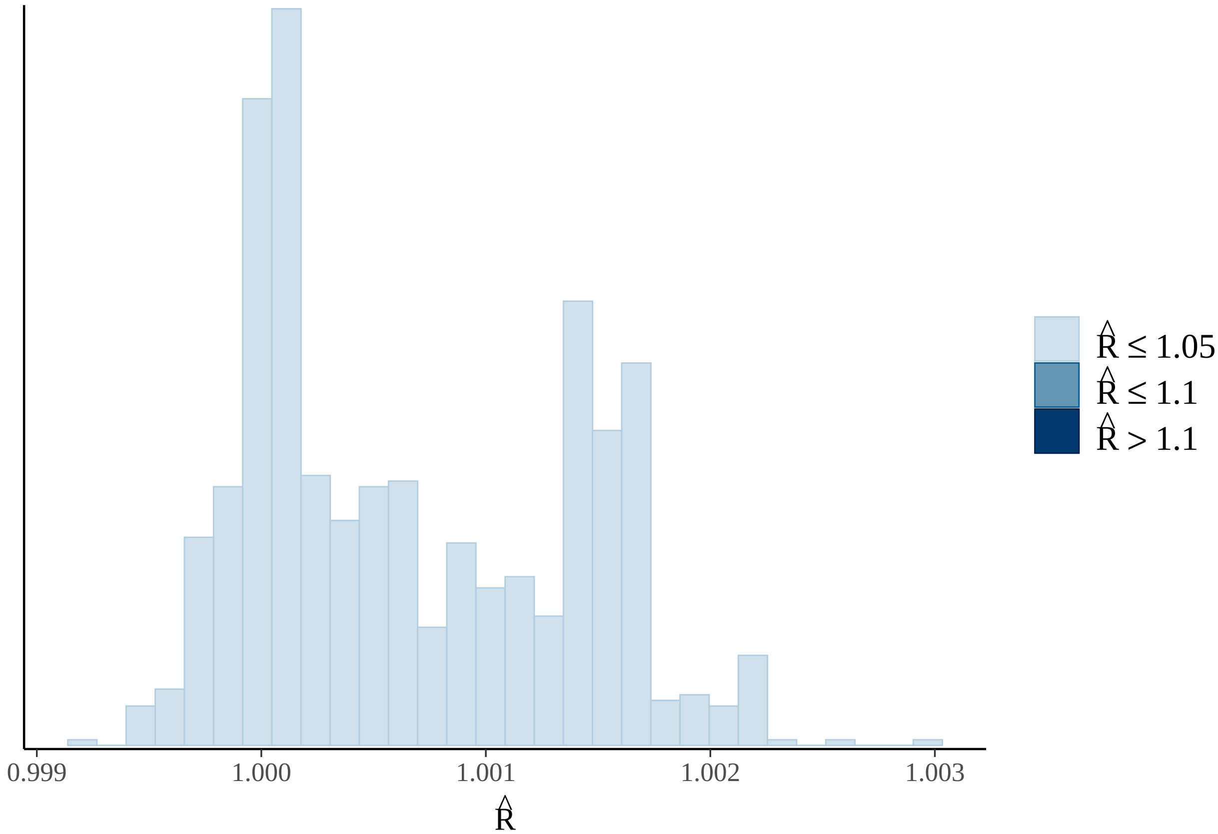
```r
mcmc_hist(pre_fit$draws(c("mu", "sigma", "u_major[1]", "u_major[2]",
                          "u_major[3]", "u_major[4]", "u_major[5]",
                          "u_gender[1]", "u_gender[2]", "u_gender[3]")))
```

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




```r
rhats <- rhat(pre_fit)
mcmc_rhat_hist(rhats)
```

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




```r
rm(stan_data, rhats, pre_mod)

pre_samples <- pre_fit$draws(c("lp__", "mu", "sigma", "tau_major", "tau_gender",
                               "u_major[1]", "u_major[2]", "u_major[3]",
                               "u_major[4]", "u_major[5]", "u_gender[1]",
                               "u_gender[2]", "u_gender[3]")) %>%
  as_draws_df()
```

Post Anxiety Model

$$PostAnxietyScore = \mu + \beta_1 * PreAnxietyScore + \beta_2 * Z + u^{\text{gender}}_{\text{gender}[i]} + u^{\text{major}}_{\text{major}[i]}$$
$$u^{\text{gender}} \sim N(0, \tau^2_{\text{gender}})$$
$$u^{\text{major}} \sim N(0, \tau^2_{\text{major}})$$

```r
post_mod <- cmdstan_model("post.stan")
```

```
## Model executable is up to date!
```

```r
post_mod$print()
```

```
## data {
##   int<lower=0> N;
##   real lower_bound;
##   real upper_bound;
##   vector<lower = lower_bound, upper = upper_bound>[N]anxiety_before;
##   vector<lower=0, upper=1>[N] Z;
```

```
##   int<lower = 0> J_major;
##   int<lower = 0 > J_gender;
##   vector<lower = lower_bound, upper = upper_bound>[N]anxiety_after;
##   int<lower = 1, upper = J_major> major[N];
##   int<lower = 1, upper = J_gender> gender[N];
##   int<lower=0, upper=1> only_prior;
## }
##
## parameters {
##   real mu;
##   real beta1;
##   real beta2;
##   real<lower=0> tau_major;
##   vector<multiplier = tau_major>[J_major] u_major;
##   real<lower=0> tau_gender;
##   vector<multiplier = tau_gender>[J_gender] u_gender;
##   real<lower=0> sigma;
## }
##
## transformed parameters {
##   vector [N]eq = mu + u_gender[gender] + u_major[major] + beta1*anxiety_before
##   + beta2*Z;
## }
##
## model {
##   mu ~ normal(0, 5);
##   sigma ~ normal(0, 5);
##   u_major ~ normal(0, tau_major);
##   tau_major ~ normal(0, 3);
##   u_gender ~ normal(0, tau_gender);
##   tau_gender ~ normal(0, 3);
##
##   target += normal_lpdf(anxiety_after | eq, sigma) -
##   log_diff_exp(normal_lcdf(upper_bound | eq, sigma),
##   normal_lcdf(lower_bound | eq, sigma));
##
##   if(only_prior == 0) {
##     anxiety_after ~ normal(eq, sigma);
##   }
## }
##
## generated quantities {
##   vector[N] log_lik;
##   vector[N] anxiety_pred;
##   for (i in 1:N) {
##     log_lik[i] = normal_lpdf(anxiety_after[i]| eq[i], sigma);
##     anxiety_pred[i] = normal_rng(eq[i], sigma);
##   }
## }
```

```
stan_data <- list(N = nrow(exp_data),
                  J_major = length(unique(exp_data$major)),
                  J_gender = length(unique(exp_data$gender)),
                  anxiety_before = exp_data$anxiety_before,
```

```r
                  major = exp_data$major,
                  gender = exp_data$gender,
                  Z = exp_data$Z,
                  anxiety_after = exp_data$anxiety_after,
                  lower_bound = 10, upper_bound = 50, only_prior = 1)
post_fit <- post_mod$sample(stan_data, parallel_chains = 4, refresh = 0,
                            show_messages = F, seed = 666, adapt_delta = 0.99)
```

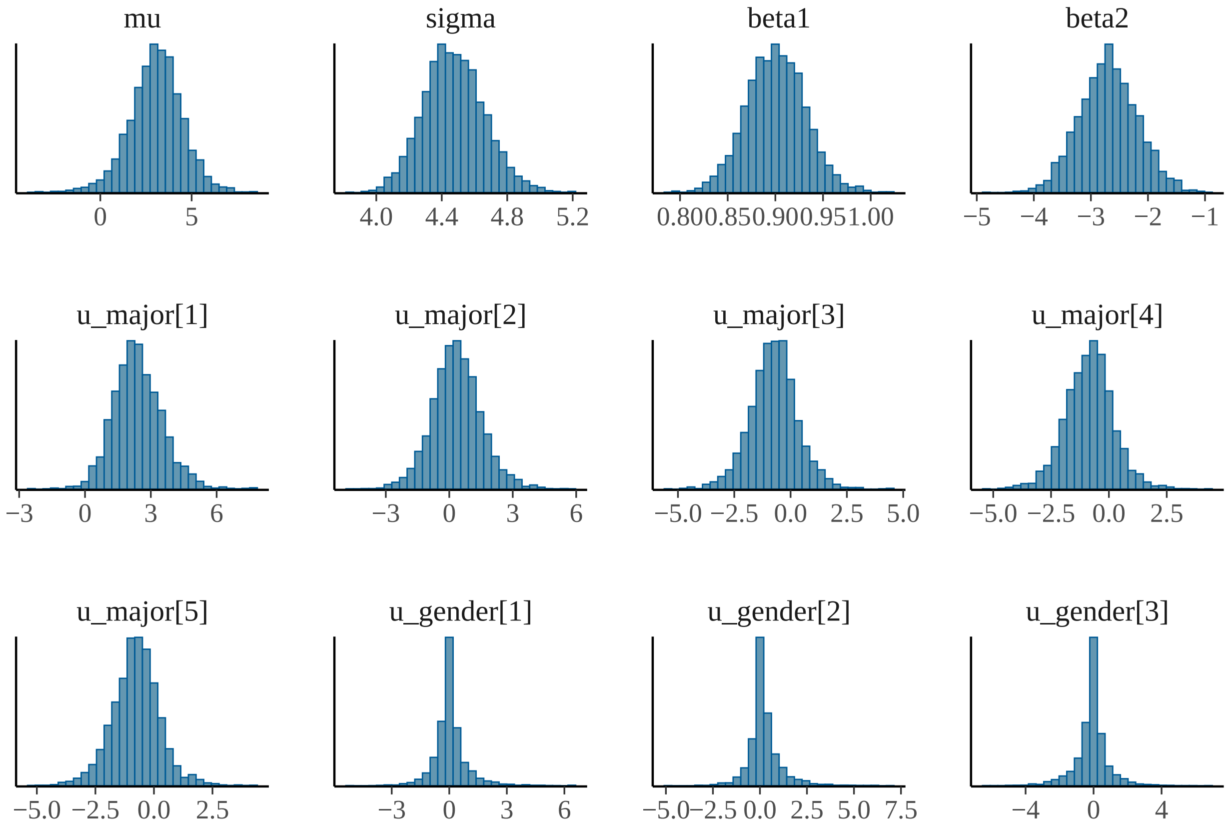
```
## Running MCMC with 4 parallel chains...
##
## Chain 4 finished in 24.4 seconds.
## Chain 1 finished in 27.2 seconds.
## Chain 3 finished in 30.4 seconds.
## Chain 2 finished in 35.2 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 29.3 seconds.
## Total execution time: 35.3 seconds.
```

```r
# 8 divergences when adapt_delta = 0.8
post_fit$print(max_rows = 15)
```

```
##     variable    mean  median   sd  mad      q5     q95 rhat ess_bulk ess_tail
##  lp__        -875.34 -875.00 3.23 3.11 -880.95 -870.71 1.00     1120     1732
##  mu             3.04    3.09 1.46 1.32    0.63    5.35 1.00     1452     1866
##  beta1          0.90    0.90 0.03 0.03    0.85    0.95 1.00     3222     2495
##  beta2         -2.69   -2.69 0.51 0.50   -3.53   -1.85 1.00     3710     2807
##  tau_major      1.96    1.80 0.89 0.75    0.90    3.63 1.00     1252     1956
##  u_major[1]     2.41    2.34 1.13 1.06    0.70    4.37 1.00     1670     2232
##  u_major[2]     0.35    0.31 1.18 1.08   -1.52    2.32 1.00     1977     2469
##  u_major[3]    -0.75   -0.74 1.07 0.96   -2.46    1.01 1.00     1737     1858
##  u_major[4]    -0.85   -0.81 1.05 0.97   -2.55    0.81 1.00     1796     2158
##  u_major[5]    -0.77   -0.76 1.03 0.91   -2.45    0.84 1.00     1708     1807
##  tau_gender     0.97    0.66 0.99 0.67    0.06    2.99 1.00     1631     2158
##  u_gender[1]    0.06    0.00 0.89 0.45   -1.19    1.51 1.00     2151     1953
##  u_gender[2]    0.22    0.08 0.89 0.44   -0.92    1.80 1.00     2064     1752
##  u_gender[3]   -0.11   -0.03 0.99 0.50   -1.82    1.42 1.00     3210     2335
##  sigma          4.48    4.47 0.19 0.18    4.18    4.79 1.00     3889     3022
##
##  # showing 15 of 915 rows (change via 'max_rows' argument)
```

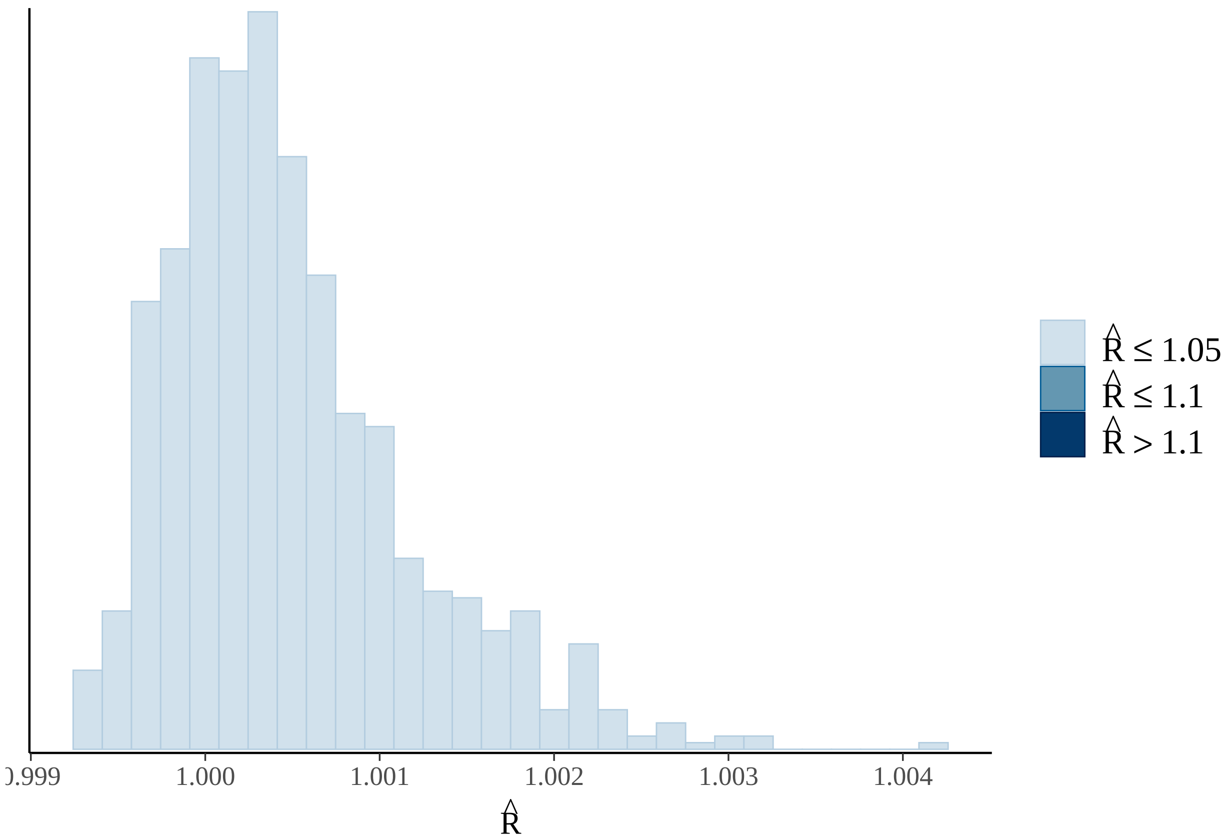
```r
mcmc_hist(post_fit$draws(c("mu", "sigma", "beta1", "beta2", "u_major[1]",
                           "u_major[2]", "u_major[3]", "u_major[4]",
                           "u_major[5]", "u_gender[1]", "u_gender[2]",
                           "u_gender[3]")))
```

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




```
rhats <- rhat(post_fit)
mcmc_rhat_hist(rhats)
```

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




```
stan_data$only_prior = 0
post_fit <- post_mod$sample(stan_data, parallel_chains = 4, refresh = 0,
                             show_messages = F, seed = 666, adapt_delta = 0.99)
```

```
## Running MCMC with 4 parallel chains...
##
## Chain 4 finished in 34.5 seconds.
## Chain 2 finished in 35.8 seconds.
## Chain 3 finished in 36.3 seconds.
## Chain 1 finished in 43.4 seconds.
##
## All 4 chains finished successfully.
## Mean chain execution time: 37.5 seconds.
## Total execution time: 43.4 seconds.
```

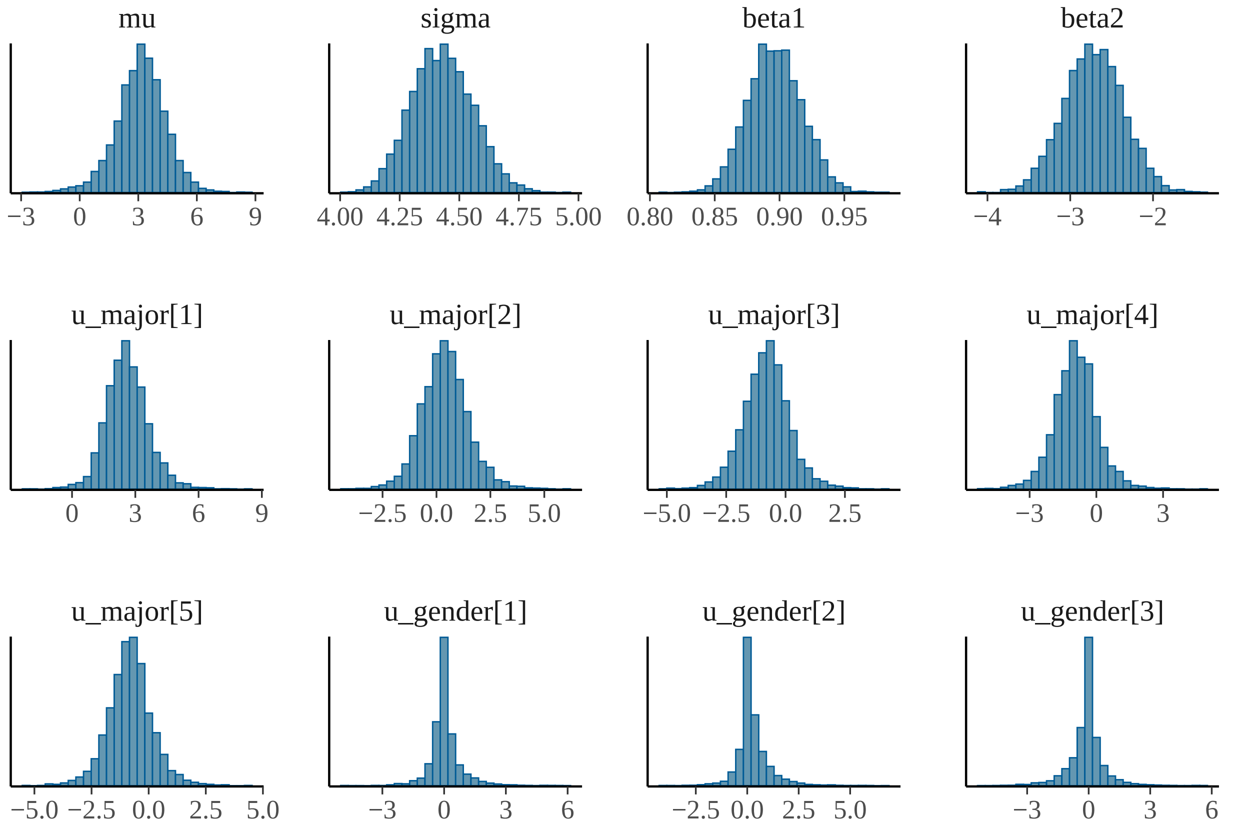
```
# 7 divergences when adapt_delta = 0.8
post_fit$print(max_rows = 15)
```

```
##     variable     mean   median   sd  mad       q5      q95 rhat ess_bulk
##  lp__        -1468.75 -1468.42 3.18 3.13 -1474.48 -1464.09 1.00     1156
##  mu              3.14     3.18 1.28 1.17     0.99     5.16 1.00     1491
##  beta1           0.90     0.90 0.02 0.02     0.86     0.93 1.00     3561
##  beta2          -2.72    -2.72 0.36 0.36    -3.32    -2.13 1.00     3917
##  tau_major       1.93     1.78 0.77 0.66     1.00     3.42 1.00     1566
##  u_major[1]      2.61     2.56 1.01 0.92     1.12     4.34 1.00     1514
##  u_major[2]      0.40     0.36 1.10 1.00    -1.30     2.24 1.00     1899
##  u_major[3]     -0.81    -0.79 0.99 0.89    -2.41     0.81 1.00     1574
##  u_major[4]     -0.91    -0.92 0.99 0.88    -2.49     0.71 1.00     1635
```

```
## u_major[5]      -0.81    -0.81 0.97 0.84    -2.34     0.74 1.00     1607
## tau_gender       0.87     0.55 0.94 0.56     0.05     2.76 1.00     1547
## u_gender[1]      0.06    -0.01 0.78 0.37    -0.95     1.37 1.00     2229
## u_gender[2]      0.26     0.10 0.81 0.38    -0.68     1.68 1.00     2114
## u_gender[3]     -0.10    -0.02 0.87 0.41    -1.50     1.16 1.00     3157
## sigma            4.43     4.43 0.13 0.13     4.22     4.64 1.00     3971
## ess_tail
##     1947
##     1786
##     2745
##     2541
##     2286
##     2054
##     2193
##     2161
##     1877
##     1898
##     1567
##     1957
##     2079
##     2388
##     2729
##
## # showing 15 of 915 rows (change via 'max_rows' argument)
```

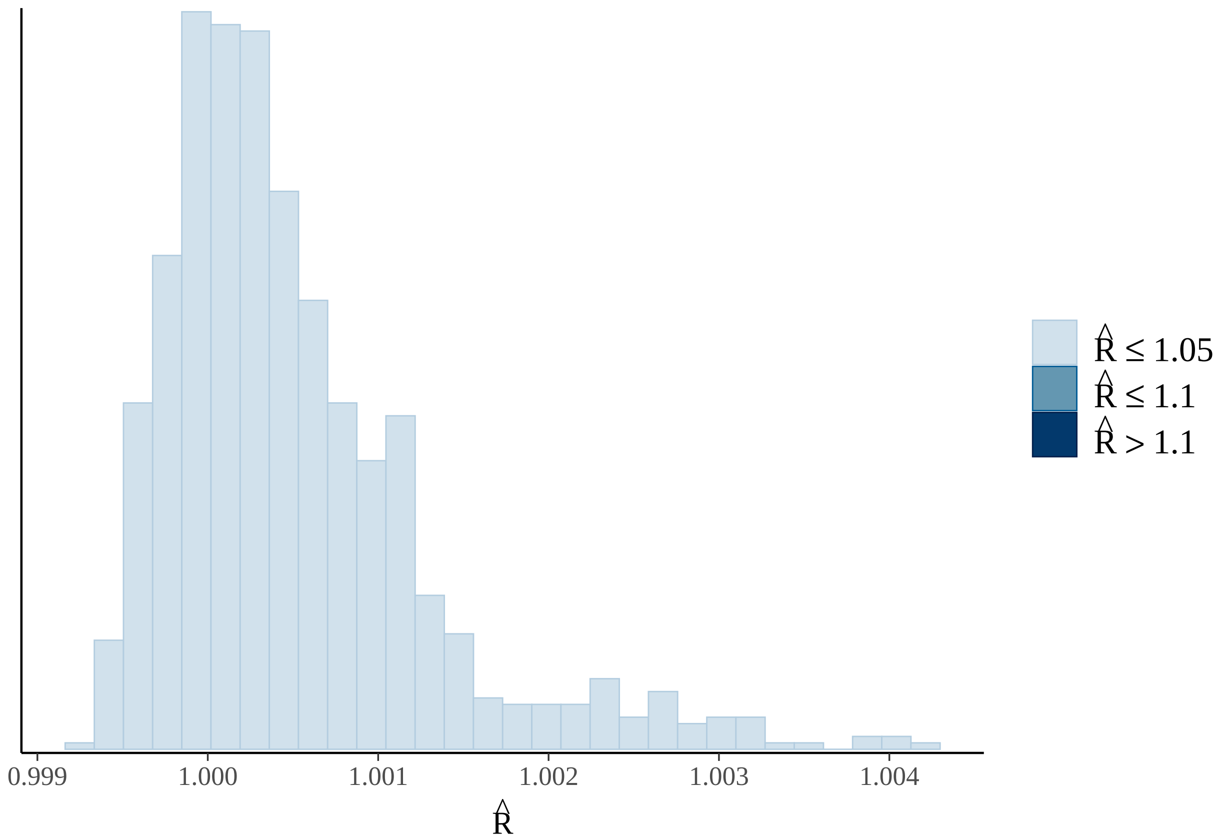
```r
mcmc_hist(post_fit$draws(c("mu", "sigma", "beta1", "beta2", "u_major[1]",
                           "u_major[2]", "u_major[3]", "u_major[4]",
                           "u_major[5]", "u_gender[1]", "u_gender[2]",
                           "u_gender[3]")))
```

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




```r
rhats <- rhat(post_fit)
mcmc_rhat_hist(rhats)
```

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



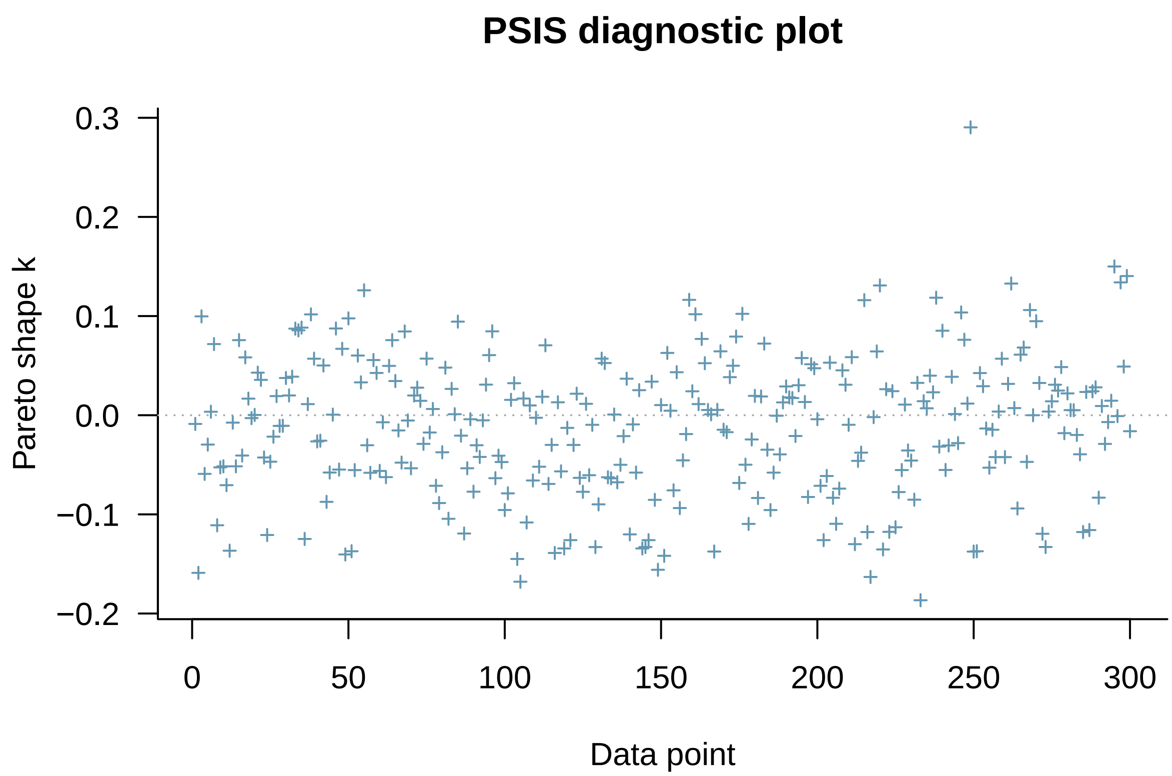

```r
rm(stan_data, rhats, post_mod)

post_samples <- post_fit$draws(c("lp__", "mu", "sigma", "beta1", "beta2",
                                 "tau_major", "tau_gender", "u_major[1]",
                                 "u_major[2]", "u_major[3]", "u_major[4]",
                                 "u_major[5]", "u_gender[1]", "u_gender[2]",
                                 "u_gender[3]")) %>%
  as_draws_df()
```

```r
loo_pre <- pre_fit$loo(save_psis = T)
print(loo_pre)
```

```
##
## Computed from 4000 by 300 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -1065.3 11.4
## p_loo         3.7  0.5
## looic      2130.6 22.8
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

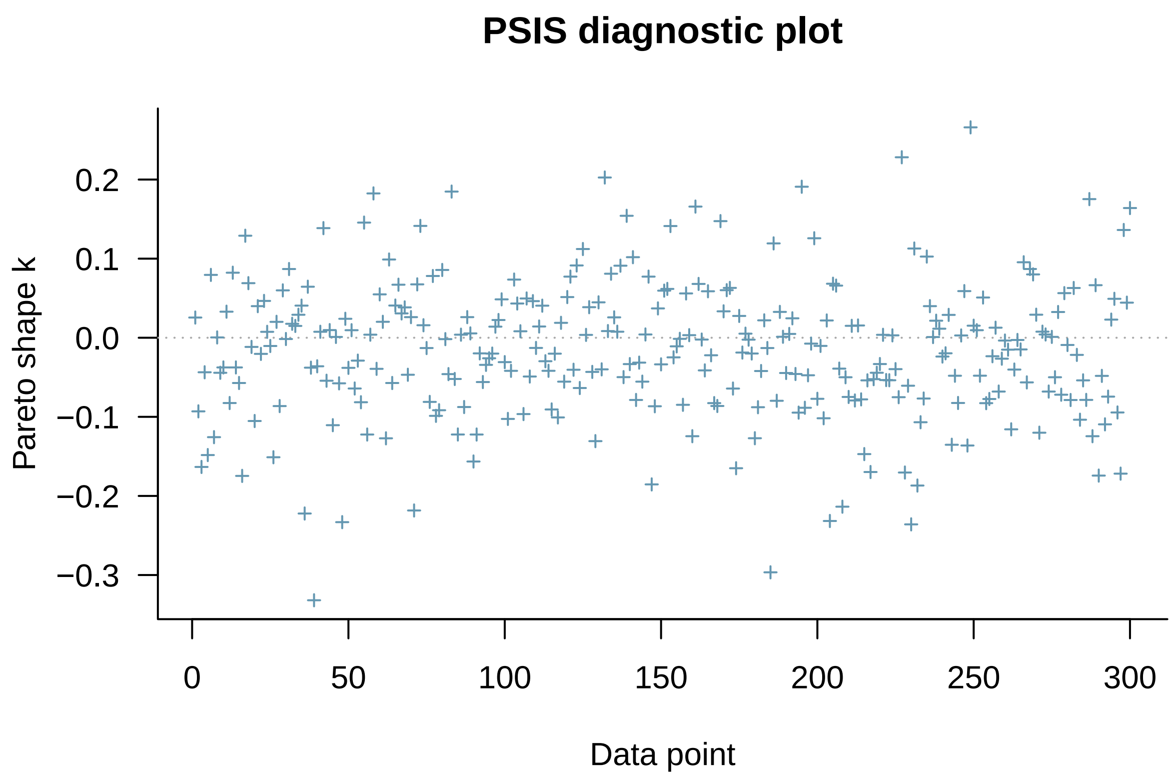
```r
plot(loo_pre)
```




```r
loo_post <- post_fit$loo(save_psis = T)
print(loo_post)
```

```
##
## Computed from 4000 by 300 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -873.3 11.9
## p_loo         4.3  0.5
## looic      1746.6 23.8
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```r
plot(loo_post)
```




```r
loo_compare(loo_pre, loo_post)
```

```
##        elpd_diff se_diff
## model2    0.0       0.0
## model1 -192.0      17.6
```

```r
rm(loo_pre, loo_post, pre_fit, post_fit)
```

In the first task I have fitted two models to the experimental data. The first model predicts the pre-intervention math anxiety scores with the given variables gender and major as random effects. There are three categories in gender and five in major, hence, there are 15 different combinations of these variables. That is the reason why the `pop_prop` has 15 rows, each represents a unique combination of these variables. Moreover, associated number of people in the population and proportion in those cells are also reported in `pop_prop`. The second model predicts the post-intervention math anxiety scores and in this model there are two additional variables compared to the first model which are pre-intervention anxiety scores of subjects and Z, indicator for subject being in placebo (0) or treatment (1) group. Both of these variables are considered as fixed effects in this model.

Observing our posterior checks, we see that the model fits the data sensibly, there are no unexpected centralization, skew or shift in means. Some of the random effect variables have more centralized graphs where most of the values are clustered around the mean. However, this does not create any issues regrading our model or the predictions. Both of these models have appropriate r-hats which are graphed above. Moreover, all models have `ess_bulk` and `ess_tail` well above 500, the lowest value is 984 which corresponds to `ess_bulk` of τ_{major} in the first model. Additionally from the PSIS diagnostic plots above we confirm that k values are in the desired range. Thus, we can conclude that both of these models fit well to data. When these models are compared with each other we observe that model 2 is a slightly better fit when compared with model 1.

Task 2

```r
u_gender_long <- spread_draws(pre_samples, u_gender[gender]) %>%
  select(gender, u_gender, .draw) #4k * 3
u_gender_matrix <- u_gender_long %>%
  pivot_wider(names_from = gender, values_from = u_gender) %>%
  select(-.draw) %>%
  as.matrix
u_major_long <- spread_draws(pre_samples, u_major[major]) %>%
  select(major, u_major, .draw) #4k * 5
u_major_matrix <- u_major_long %>%
  pivot_wider(names_from = major, values_from = u_major) %>%
  select(-.draw) %>%
  as.matrix

pred_anxiety = matrix(pre_samples$mu, nrow = 4000) %*% matrix(1, ncol = 15) +
  u_gender_matrix[,pop_prop$gender] + u_major_matrix[,pop_prop$major]
#3*5 = 15 cols, 4000 samples

mean_pre_anxiety <- colMeans(pred_anxiety)

u_gender_long <- spread_draws(post_samples, u_gender[gender]) %>%
  select(gender, u_gender, .draw)
u_gender_matrix <- u_gender_long %>%
  pivot_wider(names_from = gender, values_from = u_gender) %>%
  select(-.draw) %>%
  as.matrix
u_major_long <- spread_draws(post_samples, u_major[major]) %>%
  select(major, u_major, .draw)
u_major_matrix <- u_major_long %>%
  pivot_wider(names_from = major, values_from = u_major) %>%
  select(-.draw) %>%
  as.matrix

pred_post_anxiety <- matrix(post_samples$mu, nrow = 4000) %*%
  matrix(1, ncol = 15) + u_gender_matrix[,pop_prop$gender] +
  u_major_matrix[,pop_prop$major] + matrix(post_samples$beta1, nrow = 4000) %*%
  matrix(mean_pre_anxiety, ncol = 15)

mean_post_anxiety <- colMeans(pred_post_anxiety)

pred_post_anxiety_Z <- matrix(post_samples$mu, nrow = 4000) %*%
  matrix(1, ncol = 15) + u_gender_matrix[,pop_prop$gender] +
  u_major_matrix[,pop_prop$major] + matrix(post_samples$beta1, nrow = 4000) %*%
  matrix(mean_pre_anxiety, ncol = 15) +
  matrix(post_samples$beta2, nrow = 4000) %*% matrix(1, ncol = 15)

mean_post_anxiety_Z <- colMeans(pred_post_anxiety_Z)

y_Z1 = mean_pre_anxiety - mean_post_anxiety_Z
y_Z0 = mean_pre_anxiety - mean_post_anxiety

E_Z1 = sum(y_Z1 * pop_prop$prop)
```

```
E_Z0 = sum(y_Z0 * pop_prop$prop)

ATE = E_Z1 - E_Z0
ATE
```

```
## [1] 2.718901
```

In the second task, I have used the models I have created in the first task to predict the means for different cells. Then with these means I have calculated the difference between the pre- and post-intervention. Furthermore, I have used the proportions given in `pop_prop` to find the expected results when $Z = 0$ and $Z = 1$. Finally, I have calculated the average treatment effect from $\mathbb{E}(y|Z = 1) - \mathbb{E}(y|Z = 0)$.

The ATE is approximately 2.72 which suggests that the treatment is actually effective. Thus, we can say that this intervention is effective in decreasing math anxiety. Considering that we have poststratified our results to the population, this ATE entails that on average the treatment will decrease the math-anxiety by 2.72 units. Moreover considering the anxiety scores range between 10 and 50 this is a fairly high effect size on decreasing math anxiety.