STAT 447: Homework 7

Kevin Liu (94200474)

March 9, 2025

Question 1

```
fit = stan(
  seed = 123,
  file = "beta_binomial.stan",
  data = list(n=3, k=3),
  iter = 1000
)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on
## ’C:\Users\kevin\OneDrive\Desktop\STAT 447\Exercises\Exercise
## 7\beta_binomial.stan’
```

```
##
## SAMPLING FOR MODEL ’anon_model’ NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 6e-06 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.06 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 1: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 1: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 1: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 1: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 1: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 1: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 1: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 1: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 1: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 1: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 1: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.002 seconds (Warm-up)
## Chain 1:                0.003 seconds (Sampling)
## Chain 1:                0.005 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL ’anon_model’ NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 2e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.02 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 2: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 2: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 2: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 2: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 2: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 2: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 2: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 2: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 2: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 2: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 2: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.002 seconds (Warm-up)
## Chain 2:                0.001 seconds (Sampling)
## Chain 2:                0.003 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL ’anon_model’ NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 1e-06 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.01 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 3: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 3: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 3: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 3: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 3: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 3: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 3: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 3: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 3: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 3: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 3: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 0.002 seconds (Warm-up)
## Chain 3:                0.002 seconds (Sampling)
## Chain 3:                0.004 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL ’anon_model’ NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 2e-06 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.02 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:   1 / 1000 [  0%]  (Warmup)
## Chain 4: Iteration: 100 / 1000 [ 10%]  (Warmup)
## Chain 4: Iteration: 200 / 1000 [ 20%]  (Warmup)
## Chain 4: Iteration: 300 / 1000 [ 30%]  (Warmup)
## Chain 4: Iteration: 400 / 1000 [ 40%]  (Warmup)
## Chain 4: Iteration: 500 / 1000 [ 50%]  (Warmup)
## Chain 4: Iteration: 501 / 1000 [ 50%]  (Sampling)
## Chain 4: Iteration: 600 / 1000 [ 60%]  (Sampling)
## Chain 4: Iteration: 700 / 1000 [ 70%]  (Sampling)
## Chain 4: Iteration: 800 / 1000 [ 80%]  (Sampling)
## Chain 4: Iteration: 900 / 1000 [ 90%]  (Sampling)
## Chain 4: Iteration: 1000 / 1000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 0.002 seconds (Warm-up)
## Chain 4:                0.001 seconds (Sampling)
## Chain 4:                0.003 seconds (Total)
## Chain 4:
```

```r
print(fit)
```

```
## Inference for Stan model: anon_model.
## 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
## p     0.81    0.01 0.16  0.40  0.73  0.85  0.93  0.99   765    1
## lp__ -3.07    0.03 0.76 -5.39 -3.28 -2.77 -2.56 -2.50   606    1
##
## Samples were drawn using NUTS(diag_e) at Sun Mar  9 23:11:59 2025.
## 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 IQR of the posterior is [-2.77, 0.50].

Question 2

```r
suppressPackageStartupMessages(require(ggplot2))
suppressPackageStartupMessages(require(dplyr))
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```r
df = read.csv(
  "https://raw.githubusercontent.com/UBC-Stat-ML/web447/1e345149a5b698ccdf0a7e9b0aeabec2463c50ca/data/su
  sep = ";", header=FALSE) %>%
  mutate(count = ceiling(V4)) %>%
  rename(year = V3) %>%
  filter(year > 2005)
```

```r
count = as.integer(df$count)
time = as.numeric(df$year)

data = list(
  n = length(count),
  counts = count,
  time = time
)

init_fun <- function() {
  list(theta1 = 40, theta2 = 1.0, theta3 = 0.25)
}

fit = stan(
  seed = 123,
  file = "poisson.stan",
  data = data,
  init = init_fun,
  iter = 2500
)
```

```
## Warning in readLines(file, warn = TRUE): incomplete final line found on
## ’C:\Users\kevin\OneDrive\Desktop\STAT 447\Exercises\Exercise 7\poisson.stan’
```

```
##
## SAMPLING FOR MODEL ’anon_model’ NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 6.6e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.66 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2500 [  0%]  (Warmup)
## Chain 1: Iteration:  250 / 2500 [ 10%]  (Warmup)
## Chain 1: Iteration:  500 / 2500 [ 20%]  (Warmup)
## Chain 1: Iteration:  750 / 2500 [ 30%]  (Warmup)
## Chain 1: Iteration: 1000 / 2500 [ 40%]  (Warmup)
## Chain 1: Iteration: 1250 / 2500 [ 50%]  (Warmup)
## Chain 1: Iteration: 1251 / 2500 [ 50%]  (Sampling)
## Chain 1: Iteration: 1500 / 2500 [ 60%]  (Sampling)
## Chain 1: Iteration: 1750 / 2500 [ 70%]  (Sampling)
## Chain 1: Iteration: 2000 / 2500 [ 80%]  (Sampling)
## Chain 1: Iteration: 2250 / 2500 [ 90%]  (Sampling)
## Chain 1: Iteration: 2500 / 2500 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 25.38 seconds (Warm-up)
## Chain 1:                15.869 seconds (Sampling)
## Chain 1:                41.249 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL ’anon_model’ NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 3.3e-05 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.33 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2500 [  0%]  (Warmup)
## Chain 2: Iteration:  250 / 2500 [ 10%]  (Warmup)
## Chain 2: Iteration:  500 / 2500 [ 20%]  (Warmup)
## Chain 2: Iteration:  750 / 2500 [ 30%]  (Warmup)
## Chain 2: Iteration: 1000 / 2500 [ 40%]  (Warmup)
## Chain 2: Iteration: 1250 / 2500 [ 50%]  (Warmup)
## Chain 2: Iteration: 1251 / 2500 [ 50%]  (Sampling)
## Chain 2: Iteration: 1500 / 2500 [ 60%]  (Sampling)
## Chain 2: Iteration: 1750 / 2500 [ 70%]  (Sampling)
## Chain 2: Iteration: 2000 / 2500 [ 80%]  (Sampling)
## Chain 2: Iteration: 2250 / 2500 [ 90%]  (Sampling)
## Chain 2: Iteration: 2500 / 2500 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 24.744 seconds (Warm-up)
## Chain 2:                15.694 seconds (Sampling)
## Chain 2:                40.438 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL ’anon_model’ NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 3.1e-05 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.31 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2500 [  0%]  (Warmup)
## Chain 3: Iteration:  250 / 2500 [ 10%]  (Warmup)
## Chain 3: Iteration:  500 / 2500 [ 20%]  (Warmup)
## Chain 3: Iteration:  750 / 2500 [ 30%]  (Warmup)
## Chain 3: Iteration: 1000 / 2500 [ 40%]  (Warmup)
## Chain 3: Iteration: 1250 / 2500 [ 50%]  (Warmup)
## Chain 3: Iteration: 1251 / 2500 [ 50%]  (Sampling)
## Chain 3: Iteration: 1500 / 2500 [ 60%]  (Sampling)
## Chain 3: Iteration: 1750 / 2500 [ 70%]  (Sampling)
## Chain 3: Iteration: 2000 / 2500 [ 80%]  (Sampling)
## Chain 3: Iteration: 2250 / 2500 [ 90%]  (Sampling)
## Chain 3: Iteration: 2500 / 2500 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 20.347 seconds (Warm-up)
## Chain 3:                10.549 seconds (Sampling)
## Chain 3:                30.896 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL ’anon_model’ NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 3.1e-05 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.31 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
```

```
## Chain 4: Iteration:    1 / 2500 [  0%]  (Warmup)
## Chain 4: Iteration:  250 / 2500 [ 10%]  (Warmup)
## Chain 4: Iteration:  500 / 2500 [ 20%]  (Warmup)
## Chain 4: Iteration:  750 / 2500 [ 30%]  (Warmup)
## Chain 4: Iteration: 1000 / 2500 [ 40%]  (Warmup)
## Chain 4: Iteration: 1250 / 2500 [ 50%]  (Warmup)
## Chain 4: Iteration: 1251 / 2500 [ 50%]  (Sampling)
## Chain 4: Iteration: 1500 / 2500 [ 60%]  (Sampling)
## Chain 4: Iteration: 1750 / 2500 [ 70%]  (Sampling)
## Chain 4: Iteration: 2000 / 2500 [ 80%]  (Sampling)
## Chain 4: Iteration: 2250 / 2500 [ 90%]  (Sampling)
## Chain 4: Iteration: 2500 / 2500 [100%]  (Sampling)
## Chain 4: 
## Chain 4:  Elapsed Time: 23.669 seconds (Warm-up)
## Chain 4:                13.526 seconds (Sampling)
## Chain 4:                37.195 seconds (Total)
## Chain 4:
```

```
## Warning: There were 112 transitions after warmup that exceeded the maximum treedepth. Increase max_t
## https://mc-stan.org/misc/warnings.html#maximum-treedepth-exceeded
```

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

```r
print(fit)
```

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=2500; warmup=1250; thin=1; 
## post-warmup draws per chain=1250, total post-warmup draws=5000.
## 
##            mean se_mean   sd     2.5%      25%      50%      75%    97.5% n_eff
## theta1    42.44    0.01 0.41    41.63    42.16    42.43    42.72    43.23  1642
## theta2     1.00    0.00 0.00     1.00     1.00     1.00     1.00     1.00  1591
## theta3     0.27    0.01 0.26     0.01     0.08     0.19     0.37     0.96  1582
## lp__   28335.38    0.04 1.28 28332.21 28334.78 28335.72 28336.32 28336.85  1328
##        Rhat
## theta1    1
## theta2    1
## theta3    1
## lp__      1
## 
## Samples were drawn using NUTS(diag_e) at Sun Mar  9 23:15:04 2025.
## 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).
```

```r
library(bayesplot)
```

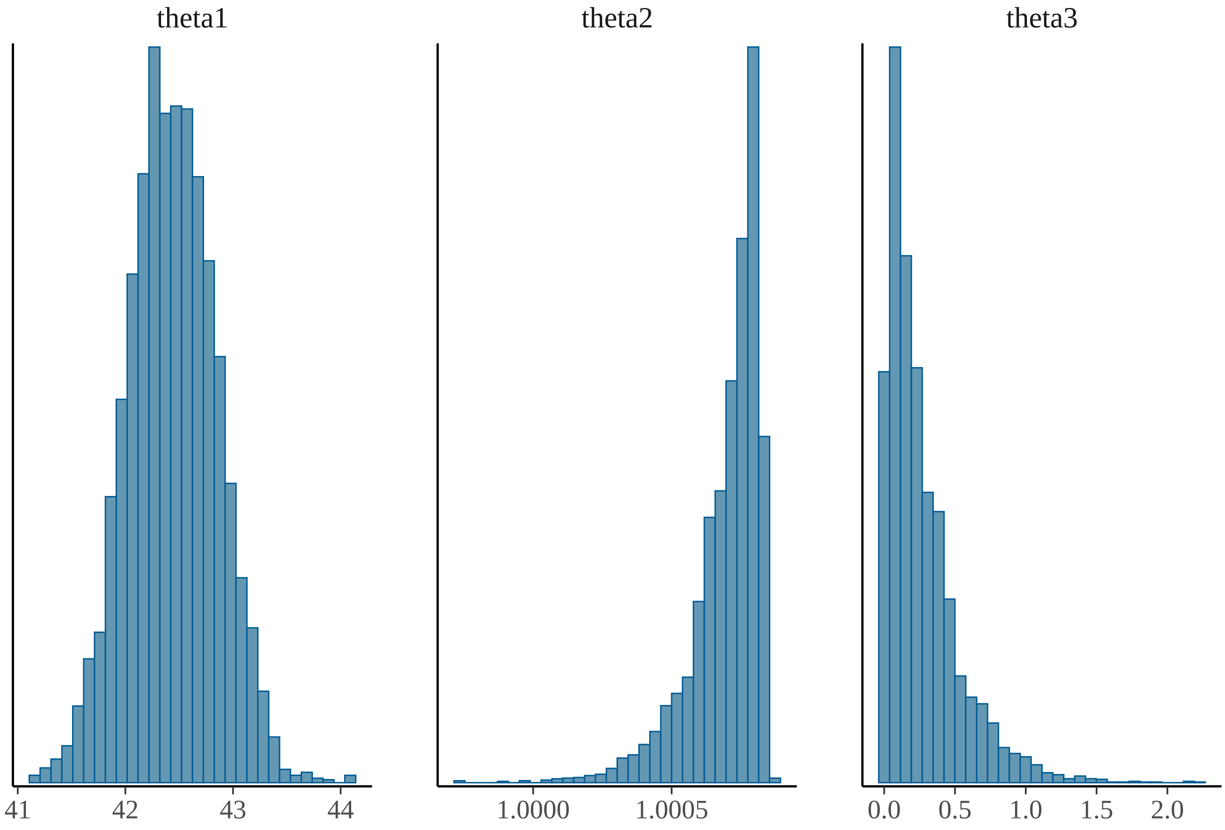
```
## Warning: package 'bayesplot' was built under R version 4.4.3
```

```
## This is bayesplot version 1.11.1
```

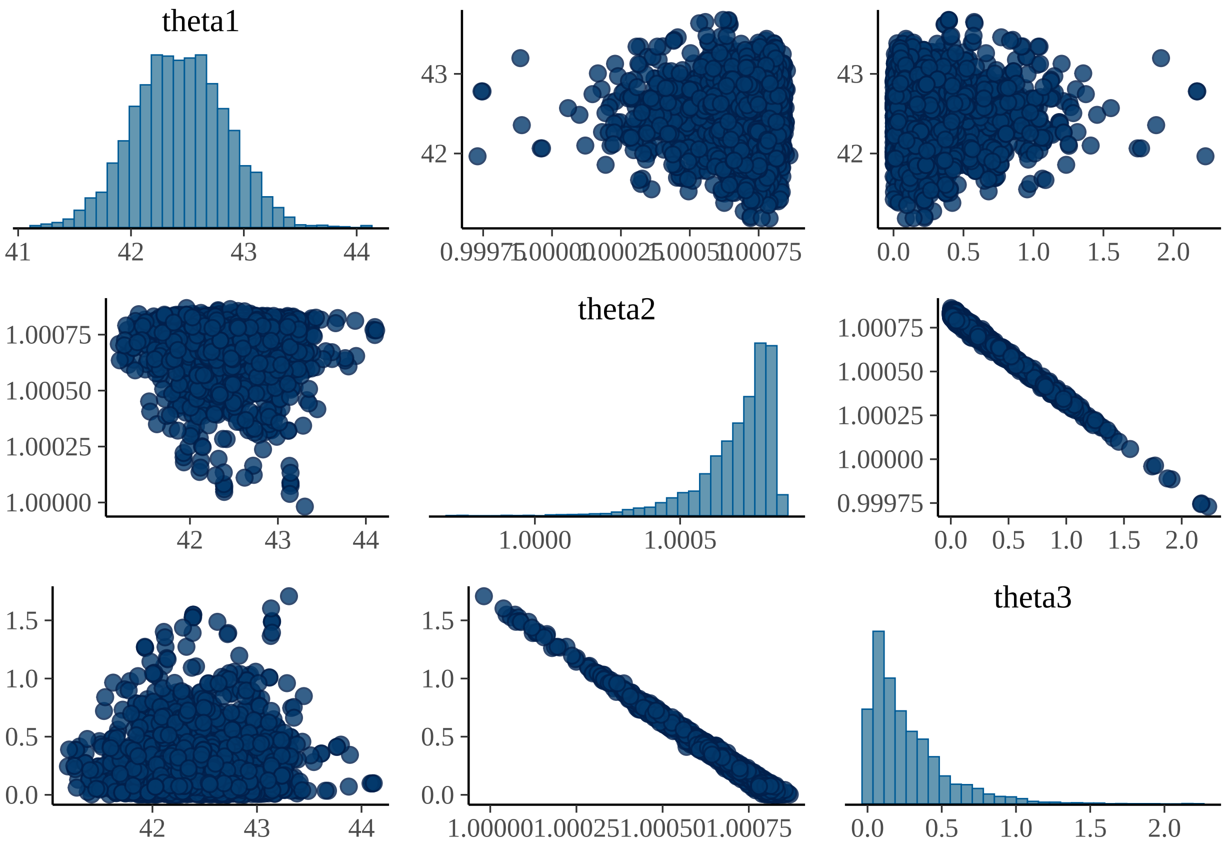
```
## - Online documentation and vignettes at mc-stan.org/bayesplot
## - bayesplot theme set to bayesplot::theme_default()

##    * Does _not_ affect other ggplot2 plots

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

```
mcmc_hist(as.array(fit), pars = c("theta1", "theta2", "theta3"))
```

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



```
mcmc_pairs(as.array(fit), pars = c("theta1", "theta2", "theta3"))
```



Question 3

```
# prior: Beta(alpha, beta)
alpha = 1
beta = 2

# observations: binomial draws
n_successes = 3
n_trials = 3

gamma_beta_binomial = function(p) {
  if (p < 0 || p > 1) return(0.0)
  dbeta(p, alpha, beta) * dbinom(x = n_successes, size = n_trials, prob = p)
}

# simple Metropolis-Hastings algorithm (normal proposal)
simple_mh = function(gamma, initial_point, n_iters) {
  samples = numeric(n_iters)
  dim = length(initial_point)

  # TODO
  for (i in 2:n_iters) {
    proposal = rnorm(1, mean = samples[i - 1], sd = 0.1)
    if (proposal < 0 || proposal > 1) {
      samples[i] = samples[i - 1]
      next
    }

    accept = gamma(proposal) / gamma(samples[i - 1])

    if (runif(1) < accept) {
      samples[i] = proposal
    } else {
      samples[i] = samples[i - 1]
    }
  }

  return(samples)
}

set.seed(123)

samples = simple_mh(gamma_beta_binomial, 0.5, 1500)

mean = mean(samples)
median = median(samples)

print(mean)
```

```
## [1] 0.6527977
```

```
print(median)
```

```
## [1] 0.6644872
```