

lab7

2023-09-11

Modelling with Stan

- Stan is its own language used in r, python etc and need a C++ compiler.
- We input our model and it calculates the log posterior density and generates samples from it.
- It can be confusing to learn at first and you will frequently encounter errors and bugs so don't worry too much.

A stan file needs 3 things:

1. Data: sample data and its size
2. Parameters: What we wish to sample
3. Model: Our likelihood and priors

We need to specify data types such as integers or real numbers and we are also able to impose bounds using “<lower=0>”. Its also important to remember to end lines of code with a semi-colon ;. We use the “check” button on our Stan file and want it to return “file.stan is syntactically correct.” otherwise we have a problem in our code.

Example: Fly Wing Length

We wish to generate samples of the length of a species of fly's wings. We are given that

$$y_i \sim N(\mu, \sigma^2)$$

and we assume the following priors on μ and σ based on a previous study where we find that the average wing length is 1.9mm:

$$\sigma \sim \text{Cauchy}(0, 1) \text{ and } \mu \sim N(\mu_0, \frac{\sigma^2}{\kappa_0}) = N(1.9, \frac{\sigma^2}{1})$$

. We have the following data

```
y = c(1.64, 1.7, 1.72, 1.74, 1.82, 1.82, 1.9, 2.08)
n = length(y)
```

Now, we want to sample values of μ and σ^2 :

```
## This is cmdstanr version 0.6.1
```

```
## - CmdStanR documentation and vignettes: mc-stan.org/cmdstanr
```

```
## - CmdStan path: C:/Users/lnbar/.cmdstan/cmdstan-2.33.0

## - CmdStan version: 2.33.0

## Warning: package 'bayesplot' was built under R version 4.2.3

## This is bayesplot version 1.10.0

## - 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
```

```
stan_model <- cmdstan_model("example_stan.stan")

stan_fit <-
  stan_model$sample(
    data = list(N = n, y = y, k0=1),
    refresh = 0, show_messages=FALSE)
```

We can inspect our model output in a number of ways. The first is Posterior summary statistics:

```
stan_fit$summary()
```

```
## # A tibble: 3 x 10
##   variable    mean median      sd      mad    q5    q95  rhat ess_bulk ess_tail
##   <chr>      <num> <num> <num> <num> <num> <num> <num>    <num>    <num>
## 1 lp__      11.4   11.8  1.26  0.889  8.94  12.6   1.00     898.     843.
## 2 mu         1.88   1.88  0.0386 0.0282 1.81   1.93   1.00    1168.     845.
## 3 sigma      0.188  0.177 0.0574 0.0468 0.118  0.291  1.00     865.    1195.
```

We will also want to extract draws from our posterior:

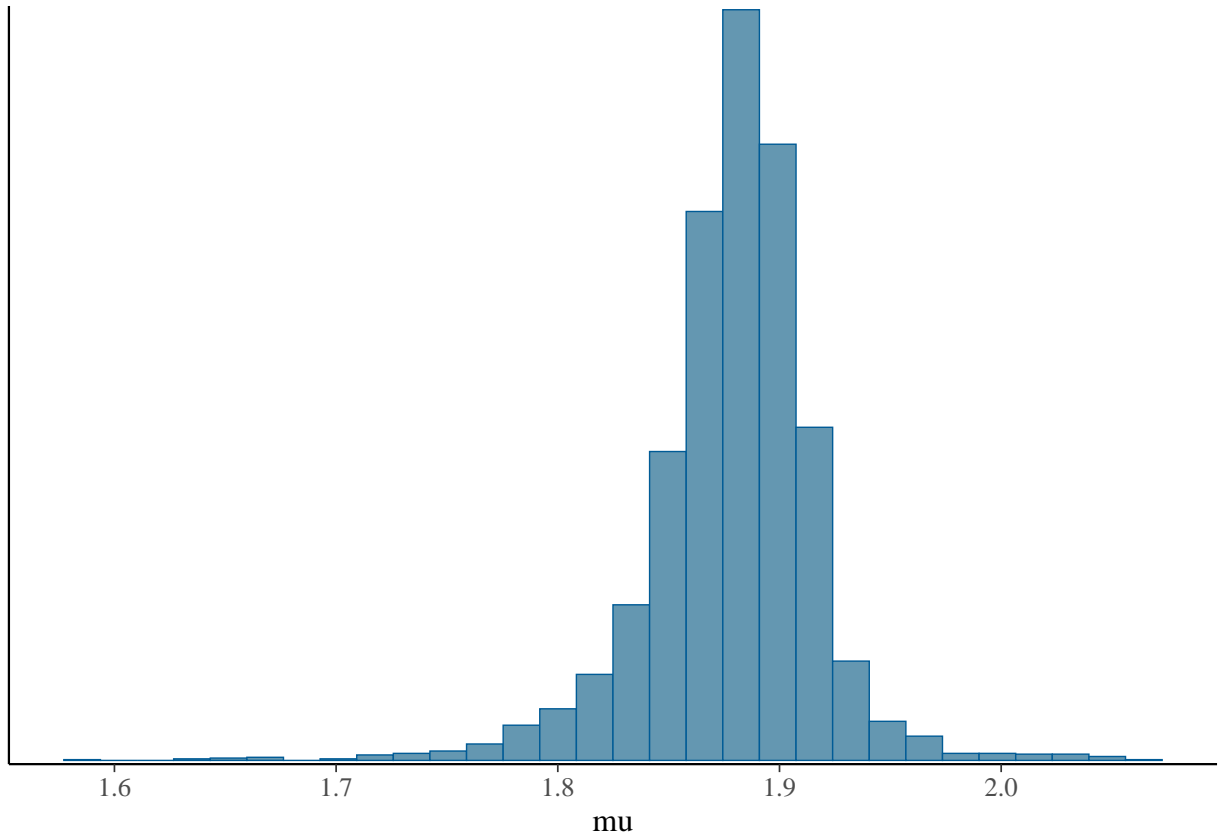
```
draws_df <- stan_fit$draws(format = "df")
print(draws_df)
```

```
## # A draws_df: 1000 iterations, 4 chains, and 3 variables
##   lp__ mu sigma
## 1    12 1.9  0.19
## 2    11 1.9  0.15
## 3    11 1.8  0.16
## 4    12 1.9  0.20
## 5    12 1.9  0.18
## 6    12 1.9  0.17
## 7    12 1.9  0.19
## 8    12 1.9  0.12
## 9    11 1.9  0.11
## 10   10 1.9  0.11
## # ... with 3990 more draws
## # ... hidden reserved variables {'.chain', '.iteration', '.draw'}
```

We can plot these draws as follows:

```
mcmc_hist(stan_fit$draws("mu"))
```

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



```
mcmc_hist(stan_fit$draws("sigma"))
```

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

