Debugging in Stan

- · Yesterday
 - RStan usage
 - Stan: data, parameters, log joint model
- · Next hour
 - Compiler vs run-time errors
 - Example Model posterior predictive check, marginalization, truncation
 - How to get help

Compiler vs Run-time error

- · Compiler error
 - malformed Stan program
 - won't compile
 - quick to detect
- · Run-time error
 - Stan program syntactically correct
 - won't run
 - need to run to debut error
 - sometimes a data problem

Compiler errors

- · Start R, load RStan
- Use stanc("filename") to compile and fail quickly
- Follow along with compiler_error_1.stan
- Drill: compiler_error_2.stan compiler_error_3.stan

Run-time errors

- · Syntactically correct, but error at runtime.
- Data errors for runtime_error_1.stan:
 - data <- list(N = 4, y = c(0, 1, 1, 2))
 - data <- list(N = 4, y = c(0, 1))
 - Create a data set that will work.

Run-time errors

- Constraints: runtime_error_2.stan (no data)
- · How to fix this?

- Uninitialized: runtime_error_3.stan (no data)
- print()

Recap

- · Compiler errors: fail immediately.
- · Run-time errors: need to run to debug.
- · Easier to debug when program is simple.

Example model

- Read count (int) data:data <- read_rdump("morning.data.R")
- · Look at data.

Example model: fake data

· Drill: Poisson regression, posterior predictive check

$$\lambda \sim \dots$$

 $y_n \sim \text{Poisson}(\lambda)$

or

$$p(\lambda, y, N) = p(\lambda) * \prod_{n=1}^{N} \frac{\lambda^{y_n}}{y_n!} e^{-\lambda}$$
$$= p(\lambda) * \prod_{n=1}^{N} Poisson(y_n, \lambda)$$

· Generate int y_rep[N] in generated quantities

Example model: marginalization

· Zero inflation: additional zeros in data. Build model

$$\begin{split} p(\lambda,\theta,y,N) = & p(\lambda) \times p(\theta) \\ & \times \prod_{n=1}^{N} (\mathbb{1}_{y_n=0} \times \theta + (1-\theta) \times \operatorname{Poisson}(y_n,\lambda)) \\ = & p(\lambda) \times p(\theta) \\ & \times \prod_{n=1}^{N} (\mathbb{1}_{y_n=0} (\theta + (1-\theta) \times \operatorname{Poisson}(0,\lambda)) \\ & + \mathbb{1}_{y_n \neq 0} ((1-\theta) \times \operatorname{Poisson}(y[n],\lambda))) \end{split}$$

Update y_rep (this might be easier to reason about)

Example model: generate data

```
set.seed(223)
lambda <- 5
theta <-0.2
U <- 10
N < -200
y \leftarrow rep(NA, N)
for (n in 1:N) {
  if (rbinom(1, 1, theta) == 1) {
    v[n] <- 0
  } else {
    x <- rpois(1, lambda)</pre>
    while (x >= U)
      x <- rpois(1, lambda)
    y[n] < -x
stan_rdump(c("N", "y"), "morning.data.R")
```

Example model: truncation

· Truncate at upper bound, U

$$\begin{aligned} p(\lambda, \theta, y, N) &= p(\lambda) \times p(\theta) \\ &\times \prod_{n=1}^{N} (\mathbb{1}_{y_n = 0} \times \theta \\ &+ (1 - \theta) \times \frac{\text{Poisson}(y_n, \lambda)}{\text{Poisson_cdf}(U, \lambda)} \end{aligned}$$

Update y_rep

Help?!

- The Manual
- Users list
 http://mc-stan.org/groups.html
 https://groups.google.com/forum/#!forum/stan-users
 - Request for help
 - The full Stan program. Snipets don't help
 - Compiler issue? The full output.
 - Data or fake data.