Intro to Stan

What to expect?

- Participate

    - try

    - ask questions

- Skills

    - follow the leader

    - drills

    - exercises

Setup

- Download files from
  http://mc-stan.org/workshops/nyc/1/

- Start R

- Load RStan: library(rstan)

Stan and Bayesian inference

- Stan language:

  - specify $\log p(\theta, y)$
  - imperative probabilistic programming language

- Bernoulli example

  - Assume independent observations of Bernoulli random variable
  - example: bernoulli.stan
  - open and discuss

Stan and Bayesian inference

- RStan:
    - Interface between R and Stan
    - correlated samples from $p(\theta|y)$

- Bernoulli example
    - Generate data, inspect
    - data <- list(N = 1000, y = rbinom(1000, 1, 0.7))
    - Use rstan to fit model
    - bernoulli <- stan("bernoulli.stan", data = data)

Stan and Bayesian inference

- `bernoulli` or `print(bernoulli)`

- `traceplot(bernoulli, "theta")`

- `theta <- extract(bernoulli, "theta")$theta`

- `hist(theta, freq = FALSE)`
  `curve(dbeta(x, sum(data$y)+1, data$N-sum(data$y)+1),`
  `            0, 1, add=TRUE)`

Stan and Bayesian inference

- Posterior analysis in R

    - $\mathbb{E}(\theta|X) \approx \frac{1}{N} \sum_{n=1}^{N} \theta^n$

    - posterior mean?

    - posterior quantiles?

    - theta close to 0.5?

    - posterior probability of generating 7?

Posterior probability

· of generating 7?

· From R:
```
y_rep <- apply(theta, 1,
 function(x) rbinom(data$N, 1, x))
apply(y_rep, 2, sum)
```

· From Stan:
```
bernoulli_generated <- stan("bernoulli_generated.stan",
  data = data)
n_rep <- extract(...)
```

Bernoulli example

· What's the prior on theta?

· empty <- list(N = 0, y = integer(0))
  bernoulli_empty <- stan(fit = bernoulli, data = empty)

· hist(extract(bernoulli_empty, "theta")$theta)

· Change prior

    – model block: theta ~ normal(0.5, 0.2);

    – Not the same model.

What is RStan doing?

1. Read Stan program

2. Translate to C++ (fast)
   cat(get_cppcode(bernoulli@stanmodel))

3. Compile C++ and link to R (slow)

4. Runs NUTS with data and returns values to R

Linear model

· Generate data:

```
data <- list()
data$N <- 50
data$x <- runif(data$N, -100, 100)
data$y <- (10 + 5 * data$x) + rnorm(data$N, 0, 20)
plot(data$x, data$y)
```

· Fit using: linear_model.stan