Pooling and Hierarchical Modeling of Repeated Binary Trial Data with Stan

Stan Development Team (in order of joining):

Andrew Gelman, **Bob Carpenter**, (Matt Hoffman), Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, (Peter Li), Allen Riddell, (Yuanjun Gao), Marco Inacio, Jeffrey Arnold, Mitzi Morris, Rob Trangucci, Rob Goedman, Brian Lau, Jonah Sol Gabry, Alp Kucukelbir, Robert L. Grant, Dustin Tran Alp Kucukelbir, Krzysztof Sakrejda, Aki Vehtari, Rayleigh Lei, Sebastian Weber, Charles Margossian, Thel Seraphim, Vincent Picaud, Imad Ali, Sean Talts

Stan 2.14.0 (March 2017) http://mc-stan.org

Stan

Warmup Exercise I

Central Limit Theorem

Multiple, Repeated Binary Trials

- R Code: `rbinom(10, N, 0.3)`
  - **N = 10** trials (10% to 50% success rate)

    2 2 1 3 3 2 3 2 2 5
  - **N = 100** trials (27% to 34% success rate)

    29 34 27 31 25 31 27 29 32 26
  - **N = 1000** trials (29% to 32% success rate)

    291 297 289 322 305 296 294 297 314 292
  - **N = 10,000** trials (29.5% to 30.7% success rate)

    3014 3031 3017 2886 2995 2944 3067 3069 3051 3068
- Central Limit Thm: uncertainty decreases as $\mathcal{O}(1/\sqrt{N})$

Warmup Exercise II

Modeling Binary Trials

Repeated Binary Trial Model

- **Data**
  - $N \in \{0, 1, \ldots\}$: number of trials (constant)
  - $y_n \in \{0, 1\}$: trial n success (known, modeled data)
- **Parameter**
  - $\theta \in [0, 1]$: chance of success (unknown)
- **Prior**
  - $p(\theta) \;=\; \text{Uniform}(\theta \mid 0, 1) = 1$
- **Likelihood**
  - $p(y \mid \theta) \;=\; \prod_{n=1}^{N} \text{Bernoulli}(y_n \mid \theta) \;=\; \prod_{n=1}^{N} \theta^{y_n} (1 - \theta)^{1 - y_n}$
- **Posterior**
  - $p(\theta \mid y) \propto p(\theta)\, p(y \mid \theta)$

Stan Program

```
data {
  int<lower=0> N;                   // number of trials
  int<lower=0, upper=1> y[N];       // success on trial n
}
parameters {
  real<lower=0, upper=1> theta;     // chance of success
}
model {
  theta ~ uniform(0, 1);            // prior
  for (n in 1:N)
    y[n] ~ bernoulli(theta);        // likelihood
}
```

A Stan Program

- Defines log (posterior) density up to constant, so...
- Equivalent to define log density directly:

```
model {
  target += 0;           // log prior
  for (n in 1:N)         // log likelihood
    target += y[n] * log(theta)
              + (1 - y[n]) * log(1 - theta);
}
```

- Equivalent to drop constant prior and vectorize likelihood:

```
model {
  y ~ bernoulli(theta);
}
```

R: Simulate Data

```
> theta <- 0.30;
> N <- 20;
> y <- rbinom(N, 1, 0.3);

> y

 [1] 1 1 1 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 1

> sum(y) / N

[1] 0.4
```

RStan: Fit

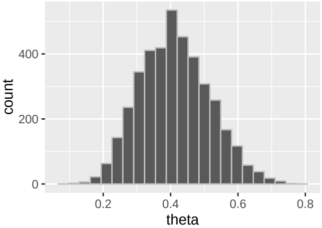
```
> library(rstan);

> fit <- stan("bern.stan",
              data = list(y = y, N = N));

> print(fit, probs=c(0.1, 0.9));

Inference for Stan model: bern.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000,
total post-warmup draws=4000.

        mean se_mean   sd    10%    90% n_eff Rhat
theta   0.41    0.00 0.10   0.28   0.55  1580    1
lp__  -15.40    0.02 0.71 -16.26 -14.89  1557    1

Samples drawn using NUTS(diag_e) at Thu Apr 21 19:38:16 2016.
```

RStan: Posterior Sample

```
> theta_draws <- extract(fit)$theta;
> mean(theta_draws);

[1] 0.4128373

> quantile(theta_draws, probs=c(0.10, 0.90));

      10%       90%
0.2830349 0.5496858
```

Marginal Posterior Histograms

```
theta_draws_df <- data.frame(list(theta = theta_draws));
plot <-
  ggplot(theta_draws_df, aes(x = theta)) +
  geom_histogram(bins=20, color = "gray");
plot;
```



- Displays the full posterior *marginal* distribution $p(\theta \mid y)$

Warmup Exercise III

Birth Rate by Sex

Birth Rate by Sex

- **Laplace**'s data on live births in Paris from 1745–1770:

sex	*live births*
female	241 945
male	251 527

- **Question 1** (Estimation)
  What is the birth rate of boys vs. girls?
- **Question 2** (Event Probability)
  Is a boy more likely to be born than a girl?
- Bayes (1763) set up the “Bayesian” model
- Laplace (1781, 1786) solved for the posterior

Binomial Distribution

- Don't know order of births, only total.
- If $y_1, \ldots, y_N \sim \text{Bernoulli}(\theta)$,

  then $(y_1 + \cdots + y_N) \sim \text{Binomial}(N, \theta)$
- The analytic form is

$$\text{Binomial}(y|N, \theta) \;=\; \binom{N}{y} \theta^y (1 - \theta)^{N-y}$$

  where the binomial coefficient normalizes for permutations (i.e., which subset of n has $y_n = 1$),

$$\binom{N}{y} = \frac{N!}{y!\,(N-y)!}$$

Mathematics vs. Simulation

- Luckily, we don't have to be as good at math as Laplace
- Nowadays, we calculate all these integrals by computer using tools like Stan

> *If you wanted to do foundational research in statistics in the mid-twentieth century, you had to be bit of a mathematician, whether you wanted to or not. . . . if you want to do statistical research at the turn of the twenty-first century, you have to be a computer programmer.*
>
> —*from Andrew's blog*

Calculating Laplace's Answers

```
transformed data {
  int male = 251527;
  int female = 241945;
}
parameters {
  real<lower=0, upper=1> theta;
}
model {
  male ~ binomial(male + female, theta);
}
generated quantities {
  int<lower=0, upper=1> theta_gt_half = (theta > 0.5);
}
```

And the Answer is...

```
> fit <- stan("laplace.stan", iter=100000);
> print(fit, probs=c(0.005, 0.995), digits=3)
```

	mean	0.5%	99.5%
theta	0.51	0.508	0.512
theta_gt_half	1.00	1.000	1.000

- Q1: θ is 99% certain to lie in $(0.508, 0.512)$
- Q2: Laplace "morally certain" boys more prevalent

Posterior Event Probabilities

- Recall that an event A is a collection of outcomes
- So A may be defined by an indicator f on parameters

$$f(\theta) = \begin{cases} 1 & \text{if } \theta \in A \\ 0 & \text{if } \theta \notin A \end{cases}$$

  - $f(\theta) = \mathrm{I}(\theta_1 > \theta_2)$ for $\Pr[\theta_1 > \theta_2 \mid y]$,
  - $f(\theta) = \mathrm{I}(\theta \in (0.50, 0.52)$ for $\Pr[\theta \in (0.50, 0.52) \mid y]$

- Defined by posterior expectation of indicator $f(\theta)$

$$\Pr[A \mid y] \;=\; \mathbb{E}[f(\theta) \mid y] \;=\; \int_\Theta f(\theta)\, p(\theta|y)\, d\theta.$$

Event Probabilities in Stan

- MCMC estimates

$$\Pr[A \mid y] \approx \sum_{m=1}^{M} f(\theta^{(m)})$$

  with posterior draws $\theta^{(1)}, \ldots, \theta^{(M)}$.

- In Stan, only need to define a variable in generated quantities block for the indicator

```
generated quantities {
  int<lower=0, upper=1> theta_gt_half = (theta > 0.5);
}
```

Bayesian Point Estimates

- **Posterior Mean Estimate** (min expected square error)

$$\hat{\theta} \;=\; \mathbb{E}[\theta \mid y] \;=\; \int_{\Theta} \theta \, p(\theta \mid y) \, \mathrm{d}\theta \;\approx\; \frac{1}{M} \sum_{m=1}^{M} \theta^{(m)}.$$

- **Posterior Median Estimate** (min expected absolute error)

$$\bar{\theta} \text{ solves } \Pr[\theta > \bar{\theta}] = 0.5$$

$$\bar{\theta} \;\approx\; \text{median}(\{\theta^{(1)}, \ldots \theta^{(M)}\})$$

  - other quantiles also estimated with posterior draws
  - need a lot of draws for accurate tail estimates

Laplace turns the Crank

- What is probability that a male live birth is more probable?

$$\begin{aligned}
\Pr[\theta > 0.5] &= \int_{\Theta} \mathrm{I}[\theta > 0.5]\, p(\theta|y, N) d\theta \\
&= \int_{0.5}^{1} p(\theta|y, N) d\theta \\
&\approx 1 - 10^{-42}
\end{aligned}$$

- Laplace solved Bayes's integral by
  - determing the posterior was a beta distribution (conjugacy!)
  - and solving the normalization (gamma functions)

Posterior Predictive Distribution

- Predict new data $\tilde{y}$ based on observed data y
- Marginalize out parameters from posterior

$$p(\tilde{y}|y) = \int_{\Theta} p(\tilde{y}|\theta)\, p(\theta|y)\, d\theta.$$

- Average predictions $p(\tilde{y}|\theta)$, weighted by posterior $p(\theta|y)$
  - $\Theta = \{\theta \mid p(\theta|y) > 0\}$ is support of $p(\theta|y)$
- Allows continuous, discrete, or mixed parameters
  - integral notation shorthand for sums and/or integrals

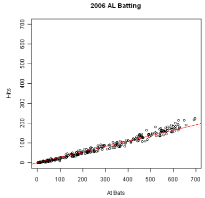
Part III
Hierarchical Models

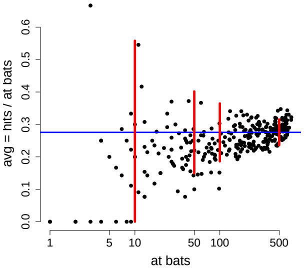
Baseball At-Bats

- For example, consider baseball batting ability.
  - Baseball is sort of like cricket, but with round bats, a one-way field, stationary "bowlers", four bases, short games, and no draws
- Batters have a number of "at-bats" in a season, out of which they get a number of "hits" (hits are a good thing)
- Nobody with higher than 40% success rate since 1950s.
- No player (excluding "bowlers") bats much less than 20%.
- Same approach applies to hospital pediatric surgery complications (a BUGS example), reviews on Yelp, test scores in multiple classrooms, . . .

Baseball Data

- Hits vs. at bats for 2006 AL (no bowlers); line at average
- not much variation
- variation related to number of trials
- success rate increases with number of trials
  - at bats predictively in nuanced models
  - blue is pooled average, red is +/- 2 binomial std devs





Pooling Data

- How do we estimate the ability of a player who we observe getting 6 hits in 10 at-bats? Or 0 hits in 5 at-bats? Estimates of 60% or 0% are absurd!
- Same logic applies to players with 152 hits in 537 at bats.
- *No pooling*: estimate each player separately
- *Complete pooling*: estimate all players together (assume no difference in abilities)
- *Partial pooling*: somewhere in the middle
  - use information about other players (i.e., the population) to estimate a player's ability

Complete Pooling Model in Stan

- Assume players all have same ability
- Assume uniform prior on abilities

```
data {
  int<lower=0> N;                    // items
  int<lower=0> K[N];                 // trials
  int<lower=0> y[N];                 // successes
}
parameters {
  real<lower=0, upper=1> phi;  // chance of success
}
model {
  y ~ binomial(K, phi);              // vectorized likelihood
}
```

No Pooling Model in Stan

- Assume each player has independent ability
- Assume uniform priors on abilities

```
data {
  int<lower=0> N;
  int<lower=0> K[N];
  int<lower=0> y[N];
}
parameters {
  real<lower=0, upper=1> phi[N];
}
model {pp
  y ~ binomial(K, phi);  // now y[n] matches phi[n]
}
```

Hierarchical Models

- Hierarchical models are principled way of determining how much pooling to apply.
- Pull estimates toward the population mean based on amount of variation in population
  - low variance population: more pooling
  - high variance population: less pooling
- In limit
  - as variance goes to 0, get complete pooling
  - as variance goes to ∞, get no pooling

Hierarchical Batting Ability

- Instead of fixed priors, estimate priors along with other parameters
- Still only uses data once for a single model fit
- Data: y_n, K_n: hits, at-bats for player n
- Parameters: ϕ_n: ability for player n
- Hyperparameters: α, β: population mean and variance
- Hyperpriors: fixed priors on α and β (hardcoded)

Hierarchical Batting Model (cont.)

$$\begin{aligned} \theta &\sim \text{Uniform}(0, 1) \\ \kappa &\sim \text{Pareto}(1.5) \\ \phi_n &\sim \text{Beta}(\kappa, \theta,\ \kappa\,(1-\theta)) \\ y_n &\sim \text{Binomial}(K_n, \phi_n) \end{aligned}$$

- Pareto provides power law distro on prior count:

$$\text{Pareto}(u \mid \alpha) \;\propto\; \frac{\alpha}{u^{\alpha+1}}$$

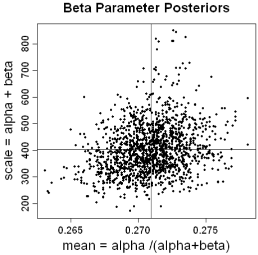
- θ is prior mean; κ is prior count (plus 2).
- Should use more informative prior on θ.

Partial Pooling Model in Stan

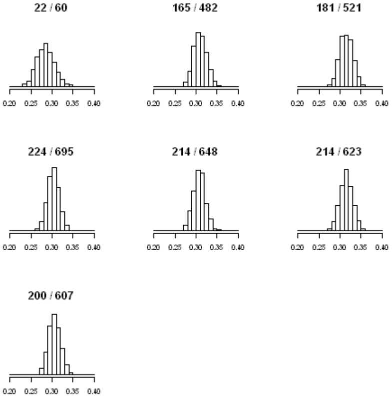
```
data {
  int<lower=0> N;
  int<lower=0> K[N];
  int<lower=0> y[N];
}
parameters {
  real<lower=0, upper=1> theta;
  real<lower=0> kappa;
  vector<lower=0, upper=1>[N] phi;
}
model {
  kappa ~ pareto(1, 1.5);                  // hyperprior
  theta ~ beta(kappa * theta,              // prior
               kappa * (1 - theta));
  y ~ binomial(K, theta);                  // likelihood
}
```

Posterior for Hyperpriors

- Scatterplot of draws
- Crosshairs at mean
- $\kappa = \alpha + \beta$ and $\theta = \frac{\alpha}{\alpha+\beta}$
- Prior mean est: $\hat{\theta} = 0.271$
- Prior count est: $\hat{\kappa} = 400$
- Together yield prior std dev of only 0.022



Posterior Ability (High Avg Players)



Who's the Best?

- Posterior probability that player n has highest ability:

$$\text{Pr}[\phi_n \geq \max(\phi) \mid y]$$

- Code up with indicator variable in Stan

```
generated quantities {
  int<lower=0, upper=1> is_best[N];
  for (n in 1:N)
    is_best[n] <- (phi[n] >= max(phi));
}
```

- Hierarchical model **adjusts for multiple comparisons** by pulling all estimates toward population mean

Results for 2006 AL Season

Player	*Average*	*At-Bats*	Pr[best]
Mauer	.347	521	0.12
Jeter	.343	623	0.11
	.342	482	0.08
	.330	648	0.04
	.330	607	0.04
	.367	60	0.02
	.322	695	0.02

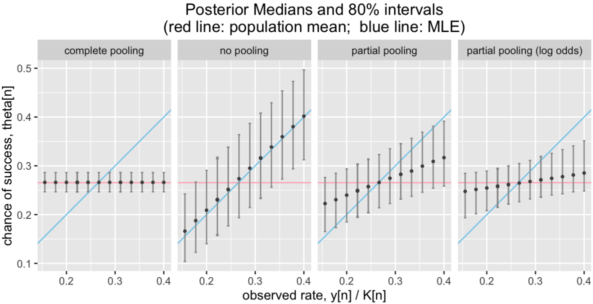
- Posterior probabilities reflect uncertainty in data
- In last game (of 162), Mauer (Minnesota) edged out Jeter (NY)

Efron & Morris (1975) Data

- From their classic analysis for shrinkage/empirical Bayes
- Picked batters with 45 at bats on a given day (artificial!)

	FirstName	LastName	Hits	At.Bats	Rest.At.Bats	Rest.Hits
1	Roberto	Clemente	18	45	367	127
2	Frank	Robinson	17	45	426	127
3	Frank	Howard	16	45	521	144
4	Jay	Johnstone	15	45	275	61
5	Ken	Berry	14	45	418	114
6	Jim	Spencer	14	45	466	126
7	Don	Kessinger	13	45	586	155
8	Luis	Alvarado	12	45	138	29
9	Ron	Santo	11	45	510	137
10	Ron	Swaboda	11	45	200	46
11	Rico	Petrocelli	10	45	538	142

Pooling vs. No-Pooling Estimates

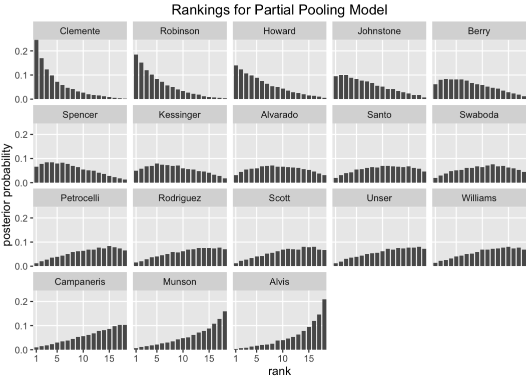


- complete pooling, no pooling, partial pooling, (log odds)

Ranking

```
generated quantities {
  int<lower=1, upper=N> rnk[N];      // rank of player n
  {
    int dsc[N];
    dsc <- sort_indices_desc(theta);
    for (n in 1:N)
      rnk[dsc[n]] <- n;
  }
```

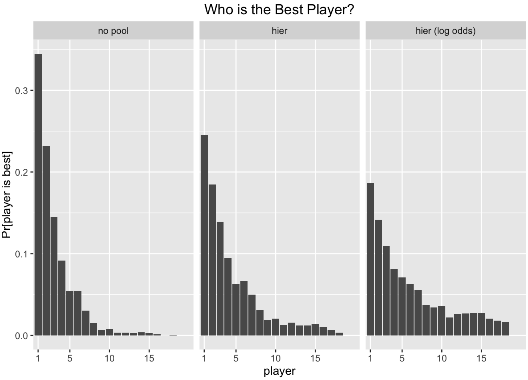
Posterior Ranks



Who is Best? better Stan code

```
generated quantities {
  ...
  int<lower=0, upper=1> is_best[N];
  ...
  for (n in 1:N)
    is_best[n] <- (rnk[n] == 1);  // more efficient
  ...
```

Who is Best? Posterior



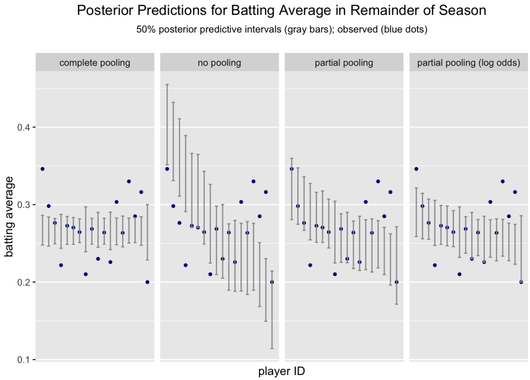
Posterior Predictive Inference

- How do we predict new outcomes (e.g., rest of season)?

```
data {
  int<lower=0> K_new[N];       // new trials
  int<lower=0> y_new[N];       // new successes
  ...
generated quantities {
  int<lower=0> z[N];  // posterior prediction
  for (n in 1:N)
    z[n] <- binomial_rng(K_new[n], theta[n]);
```

- Full Bayes accounts for two sources of uncertainty
  - estimation uncertainty (built into posterior)
  - sampling uncertainty (explicit RNG function)

Posterior Predictions

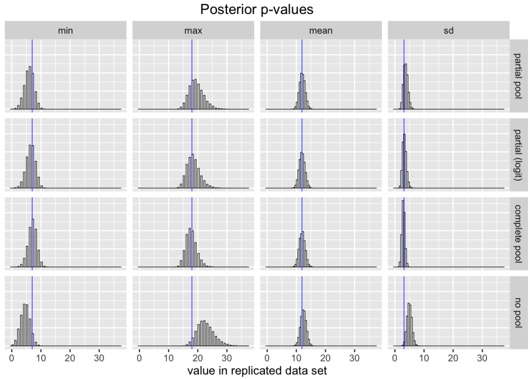


Posterior Predictive Check

- Replicate data from paraemters

```
generated quantities {
  ...
  for (n in 1:N)
    y_rep[n] <- binomial_rng(K[n], theta[n]);
  for (n in 1:N)
    y_pop_rep[n] <- binomial_rng(K[n],
                                 beta_rng(phi * kappa,
                                          (1 - phi) * kappa));
  min_y_rep <- min(y_rep);
  sd_y_rep <- sd(to_vector(y_rep));
  p_min <- (min_y_rep >= min_y);
  p_sd <- (sd_y_rep >= sd_y);
}
```

Posterior p-Values



Calibration and Sharpness

- **Calibration**: A model is calibrated if the 50% intervals contain roughly 50% of the true intervals
  - technically, we expect Binomial$(N, 0.5)$ of N parameters to fall in their 50% intervals
  - we can evaluate with held-out data using cross-validation
- **Sharpness**: One posterior is sharper than another if it concentrates more posterior mass around the true value
  - e.g., central posterior intervals of interest are narrower
  - see: Gneiting, Balabdaoui, and Raftery (2007) Probabilistic forecasts, calibration and sharpness. *JRSS B*.

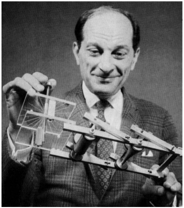
More in the Case Study

- This talk roughly followed my Stan case study:
  - Hierarchical Partial Pooling for Repeated Binary Trials
- Available under case studies at
  - http://mc-stan.org/documentation.
- Contribute case studies in knitr or Jupyter
  - Chris Fonnesbeck (of PyMC3 fame) wrote a great PyStan case study on hierarchical modeling for continuous data as a Python Jupyter notebook (follow above link)
  - Many more case studies, including new ones by Michael Betancourt on core Stan computational issues

Questions?

Stan's Namesake

- Stanislaw Ulam (1909–1984)
- Co-inventor of Monte Carlo method (and hydrogen bomb)



- Ulam holding the Fermiac, Enrico Fermi's physical Monte Carlo simulator for random neutron diffusion