Monte Carlo Methods

Professor Laura Baracaldo

Posterior inference for arbitrary functions

- Assume that $Y \sim \mathrm{Bin}(n, \theta)$ but that you are interested in the log odds:

$$\gamma = \log \mathrm{odds}(\theta) = \log \frac{\theta}{1-\theta}$$

- We use a Beta prior, so that the posterior distribution for θ is also a Beta distribution.

How do we estimate the posterior distribution for the log odds?

Posterior inference for arbitrary functions

Method of transformations

1. Find the inverse, $\theta = g^{-1}(\gamma) = \frac{e^\gamma}{1+e^\gamma}$

2. Compute $\frac{dg^{-1}(\gamma)}{d\gamma}$

3. Find $p_\gamma(\gamma \mid y_1, \ldots y_n) = \left|\frac{dg^{-1}(\gamma)}{d\gamma}\right| \times p_\theta(g^{-1}(\gamma) \mid y_1, \ldots y_n)$

Posterior inference for arbitrary functions

Method of transformations

1. Find the inverse, $\theta = g^{-1}(\gamma) = \frac{e^\gamma}{1+e^\gamma}$

2. Compute $\frac{dg^{-1}(\gamma)}{d\gamma}$

3. Find $p_\gamma(\gamma \mid y_1, \dots y_n) = \left|\frac{dg^{-1}(\gamma)}{d\gamma}\right| \times p_\theta(g^{-1}(\gamma) \mid y_1, \dots y_n)$

Don't bother! If we're computing expected values, don't need the method of transformations.

Posterior inference for arbitrary functions

For any $\gamma = g(\theta)$ we have

$$E(g(\theta)|y) = \int g(\theta)p(\theta|y)d\theta$$

Posterior inference for arbitrary functions

For any $\gamma = g(\theta)$ we have

$$E(g(\theta)|y) = \int g(\theta)p(\theta|y)d\theta$$

- $E[\gamma \mid y] = \int \log(\frac{\theta}{1-\theta})p(\theta \mid y)d\theta$
- $Pr(\theta \in R \mid y) = E(I\left[\theta \in R\right]|y)$
- $Var(\theta \mid y) = E\left[(\theta - E[\theta \mid y])^2|y\right]$

$$\int g(\theta)p_\theta(\theta \mid y_1, \ldots y_n)d\theta = \int \gamma p_\gamma(\gamma \mid y_1, \ldots y_n)d\gamma$$

Law of the Unconscious Statstician

Monte Carlo Method for Computing Integrals

Law of the Unconscious Statstician:

$$\int g(\theta)p_\theta(\theta \mid y_1, \ldots y_n)d\theta = \int \gamma p_\gamma(\gamma \mid y_1, \ldots y_n)d\gamma$$

We can approximate this integral through simulation!

Monte Carlo Method for Computing Integrals

- $\bar{\theta} = \sum_{s=1}^{S} \theta^{(s)}/S \rightarrow \mathrm{E}[\theta|y_1, \ldots, y_n]$
- $\sum_{s=1}^{S} \left(\theta^{(s)} - \bar{\theta}\right)^2/(S-1) \rightarrow \mathrm{Var}[\theta|y_1, \ldots, y_n]$
- $\# \left(\theta^{(s)} \leq c\right)/S \rightarrow \Pr(\theta \leq c|y_1, \ldots, y_n)$
- the α-percentile of $\left\{\theta^{(1)}, \ldots, \theta^{(S)}\right\} \rightarrow \theta_\alpha$

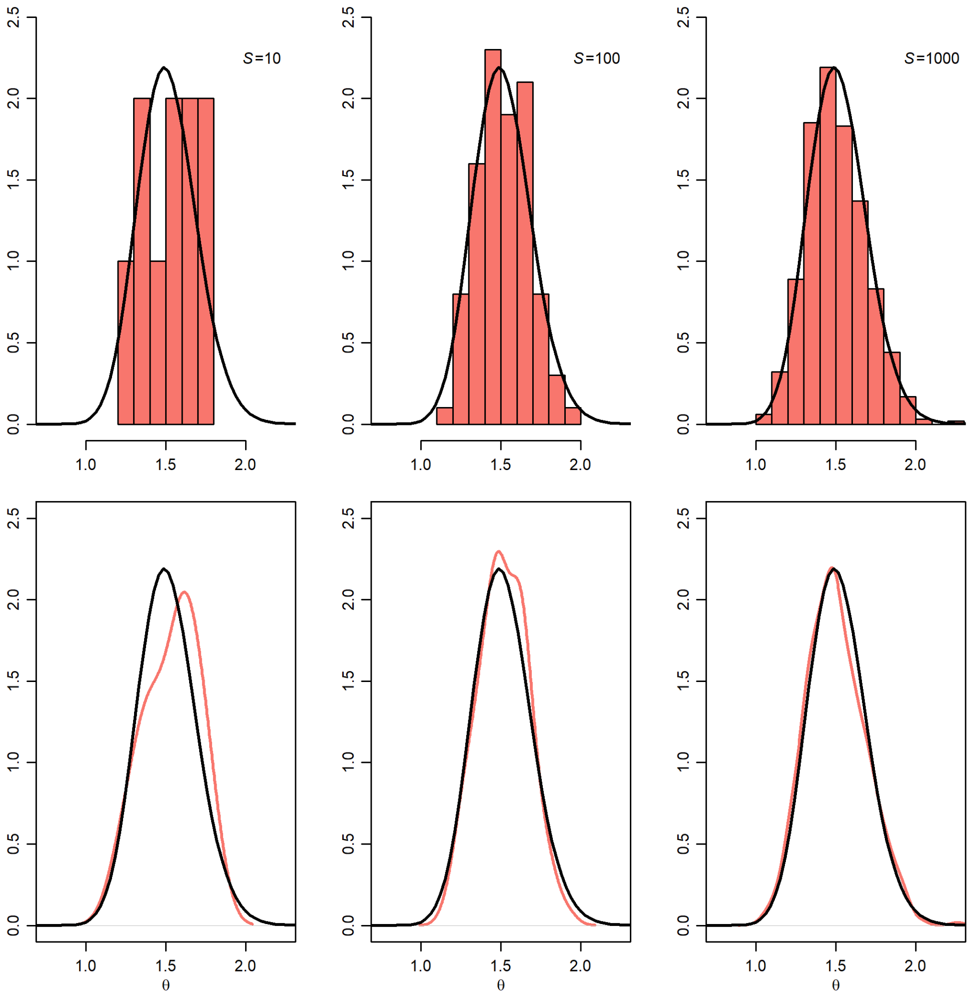
Monte Carlo Error

- Reminder: $\bar{\theta} = \sum_{s=1}^{S} \theta^{(s)}/S$ and S is the number of samples.
- If posterior samples are independent then:

$$\mathrm{Var}(\bar{\theta}) = \frac{1}{S^2}\sum_{s=1}^{S}\mathrm{Var}(\theta^{(s)}) = \frac{\mathrm{Var}(\theta \mid y_1, \ldots y_n)}{S}$$

- In general, the Monte Carlo error decreases with $\frac{1}{S}$
- Monte Carlo integration can be very powerful *if* you we can sample from the posterior!
  - This is a *big* "if"

Monte Carlo approximations of a distribution



Posterior inference for arbitrary functions

Assume we want to estimate the posterior mean $E[\gamma \mid y_1, \ldots y_n]$. For example, assume $\gamma = \log \text{odds}(\theta) = \log \frac{\theta}{1-\theta}$. Then:

1. sample $\theta^{(1)} \sim p(\theta | y_1, \ldots, y_n)$, compute $\gamma^{(1)} = \log(\frac{\theta^{(1)}}{1-\theta^{(1)}})$

Posterior inference for arbitrary functions

Assume we want to estimate the posterior mean $E[\gamma \mid y_1, \ldots y_n]$. For example, assume $\gamma = \log \text{odds}(\theta) = \log \frac{\theta}{1-\theta}$. Then:

1. sample $\theta^{(1)} \sim p(\theta|y_1, \ldots, y_n)$, compute $\gamma^{(1)} = \log(\frac{\theta^{(1)}}{1-\theta^{(1)}})$
2. sample $\theta^{(2)} \sim p(\theta|y_1, \ldots, y_n)$, compute $\gamma^{(2)} = \log(\frac{\theta^{(2)}}{1-\theta^{(2)}})$

Posterior inference for arbitrary functions

Assume we want to estimate the posterior mean $E[\gamma \mid y_1, \ldots y_n]$. For example, assume $\gamma = \log \text{odds}(\theta) = \log \frac{\theta}{1-\theta}$. Then:

1. sample $\theta^{(1)} \sim p(\theta|y_1, \ldots, y_n)$, compute $\gamma^{(1)} = \log(\frac{\theta^{(1)}}{1-\theta^{(1)}})$
2. sample $\theta^{(2)} \sim p(\theta|y_1, \ldots, y_n)$, compute $\gamma^{(2)} = \log(\frac{\theta^{(2)}}{1-\theta^{(2)}})$
3. sample $\theta^{(3)} \sim p(\theta|y_1, \ldots, y_n)$, compute $\gamma^{(3)} = \log(\frac{\theta^{(3)}}{1-\theta^{(3)}})$

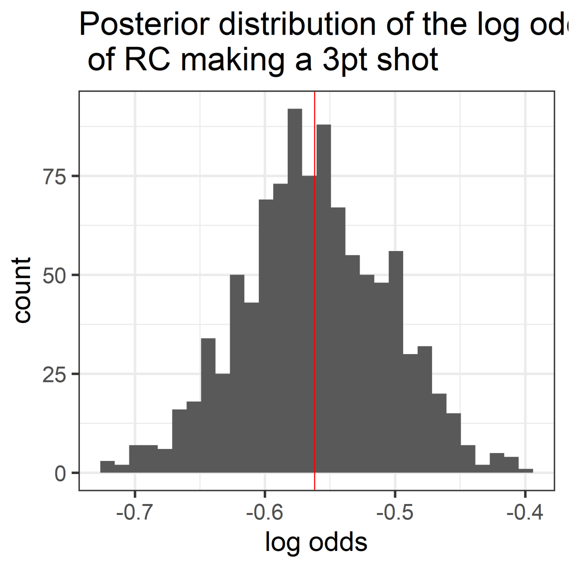
... etc until we have S samples.

Compute $\frac{1}{S}\sum_i^S \gamma^{(i)}$ where S is the number of Monte Carlo samples.

Log-odds of Robert Covington Made 3

Assume $P(\theta \mid y) = \text{Beta}(527, 924)$, the posterior distribution for γ

```
theta_samples <- rbeta(1000, 527, 924)
gamma_samples <- log(theta_samples / (1-theta_samples))
```



Posterior Predictive Checks

Posterior predictive model checking

- Let y_{obs} represent the observe data $y_1, \ldots y_n$
- Let $\tilde{y}$ represent n replicated (e.g fake) observations generated from the model
- $p(\tilde{y} \mid y_{\text{obs}}) = \int p(\tilde{y} \mid \theta) p(\theta \mid y_{\text{obs}}) d\theta$
- Generate test quantity from $t(\tilde{y})$
- Check if the simulated test quantities are similar to the observed test quantity, $t(y_{\text{obs}})$

Posterior predictive model checking

- If the model fits the data, then fake data generated under the model should look similar to the observed data
- Discrepancies can be due to model misfit or chance (or both!)
- Monte Carlo approach:

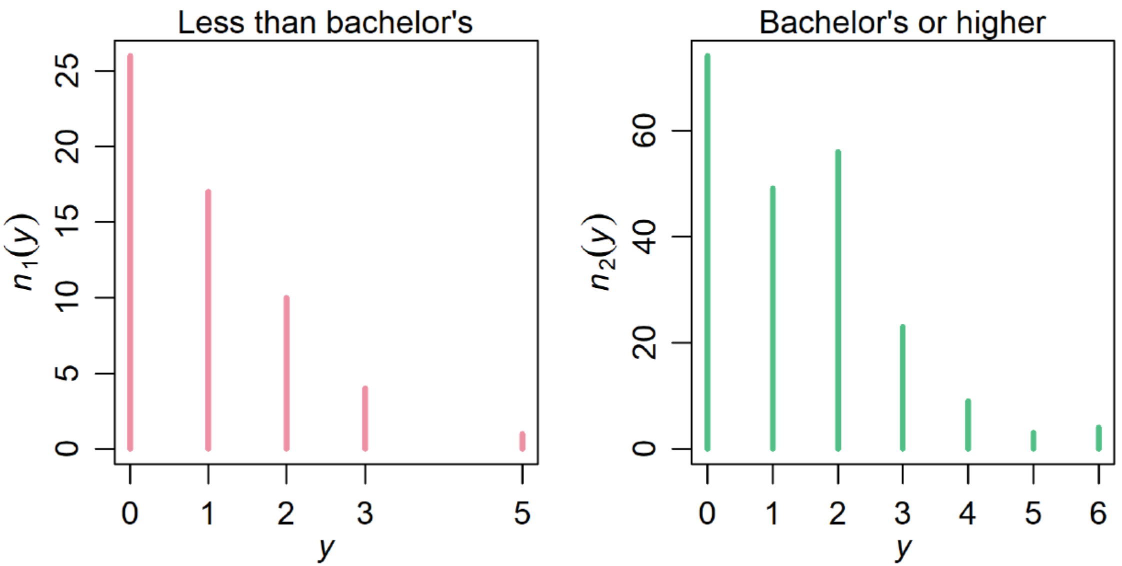
  1. sample $\theta^{(s)} \sim p(\theta|\boldsymbol{Y} = \boldsymbol{y}_{\text{obs}})$
  2. sample $\tilde{\boldsymbol{y}}^{(s)} = \left(\tilde{y}_1^{(s)}, \ldots, \tilde{y}_n^{(s)}\right) \sim \text{i.i.d. } p(y|\theta^{(s)})$
     - $\tilde{y}$ has same number of observations as y_{obs}
  3. compute $t^{(s)} = t\left(\tilde{\boldsymbol{y}}^{(s)}\right)$

Predictive Checks: an example

- In the 1990's there was a survey of 276 men, in their 30s
- Recorded number of children and educational attainment
  - Bachelor's degree or higher ($n_1 = 58$)
  - Less than bachelor's degree ($n_2 = 218$)

$$Y_{1,1} \ldots, Y_{n_1,1} | \theta_1 \sim \text{ i.i.d. Poisson } (\theta_1)$$
$$Y_{1,2} \ldots, Y_{n_2,2} | \theta_2 \sim \text{ i.i.d. Poisson}(\theta_2)$$

PPCs example



Number of children x Number of men

A Bayesian Modeling Process (overview)

1. Propose a sampling model or DGP, here $Y \mid \theta \sim \text{Pois}(\theta)$
2. Propose a prior distribution, here $\theta \sim \text{Gamma}(a, b)$
3. Compute the posterior distribution, here $p(\theta \mid Y = y) \sim \text{Gamma}(a + y, \beta + \nu)$

A Bayesian Modeling Process (overview)

1. Propose a sampling model or DGP, here $Y \mid \theta \sim \text{Pois}(\theta)$

2. Propose a prior distribution, here $\theta \sim \text{Gamma}(a, b)$

3. Compute the posterior distribution, here $p(\theta \mid Y = y) \sim \text{Gamma}(a + y, \beta + \nu)$

4. Simulate test statistics, $T(\tilde{y})^{(s)}$ from the posterior predictive distribution

    - for s in 1:num_fake_data
    - Sample $\theta^{(s)} \sim Gamma(a + y, b + \nu)$
    - Sample $\tilde{y}^{(s)} \sim$ i.i.d $\text{Pois}(\theta^{(s)})$ (same sample size as y_{obs})
    - Compute $T(\tilde{y}^{(s)})$

A Bayesian Modeling Process (overview)

1. Propose a sampling model or DGP, here $Y \mid \theta \sim \text{Pois}(\theta)$

2. Propose a prior distribution, here $\theta \sim \text{Gamma}(a, b)$

3. Compute the posterior distribution, here
   $p(\theta \mid Y = y) \sim \text{Gamma}(a + y, \beta + \nu)$

4. Simulate test statistics, $T(\tilde{y})^{(s)}$ from the posterior predictive distribution

   - for s in 1:num_fake_data
   - Sample $\theta^{(s)} \sim Gamma(a + y, b + \nu)$
   - Sample $\tilde{y}^{(s)} \sim$ i.i.d Pois$(\theta^{(s)})$ (same sample size as y_{obs})
   - Compute $T(\tilde{y}^{(s)})$

5. Compare the samples $T(\tilde{y}^{(s)})$ to $T(y_{\text{obs}})$

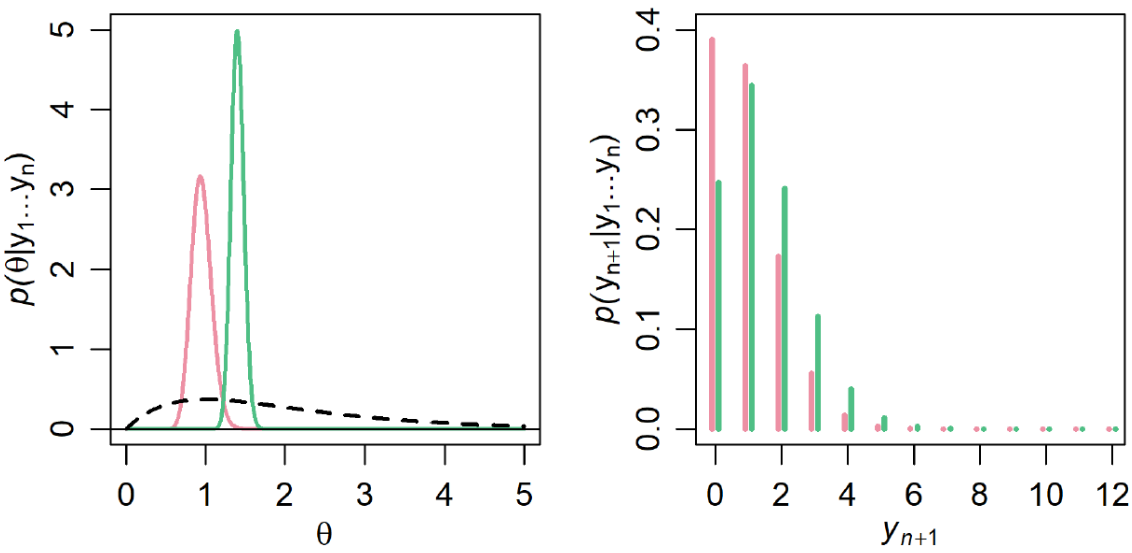
6. Identify any model misfit, go back to step 1 and repeat.

PPCs example

- In the 1990's there was a survey of 276 men, in their 30s
- Recorded number of children and educational attainment
  - Bachelor's degree or higher ($n_1 = 58$)
  - Less than bachelor's degree ($n_2 = 218$)

$$
\begin{aligned}
Y_{1,1} \ldots, Y_{n_1,1} | \theta_1 &\sim \text{ i.d. . Poisson } (\theta_1) \\
Y_{1,2} \ldots, Y_{n_2,2} | \theta_2 &\sim \text{ i.d. Poisson}(\theta_2)
\end{aligned}
$$

PPCs example



PPCs example

- Let's check the model fit for the "without Bachelor's" group first
- Do S times:
  - sample $n_2 = 218$ observations $\tilde{y}$ from the posterior predictive distribution
- Let $T(\tilde{y})$ be the fraction of men with no children

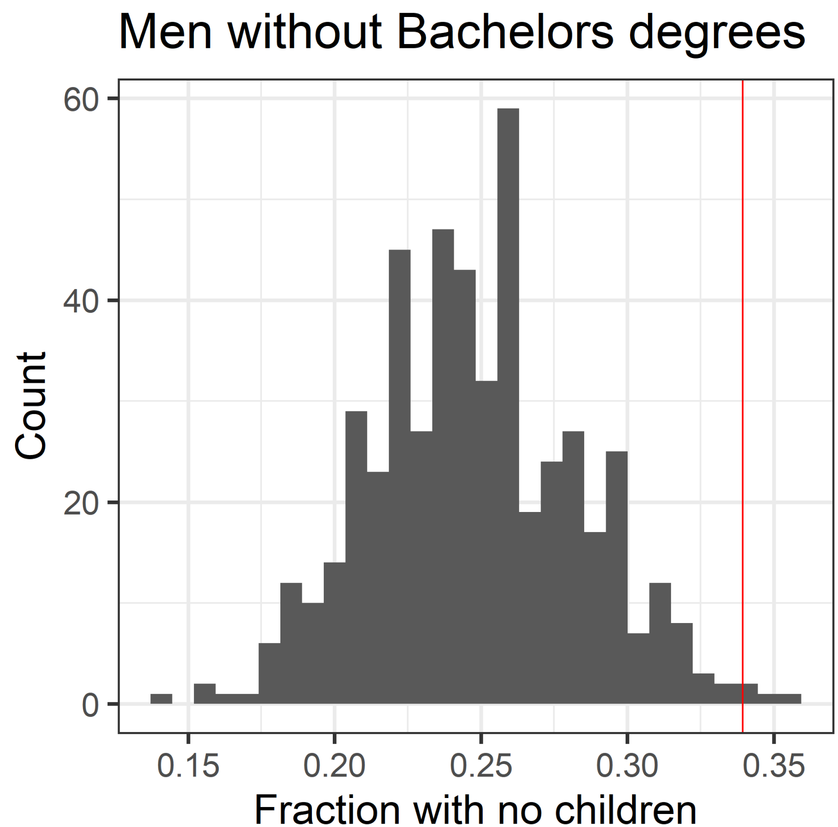
PPCs example

- Let's check the model fit for the "without Bachelor's" group first
- Do S times:
  - sample $n_2 = 218$ observations $\tilde{y}$ from the posterior predictive distribution
- Let $T(\tilde{y})$ be the fraction of men with no children

```r
S <- 1000
t_s <- numeric(S)
for(s in 1:S){
  theta_s <- rgamma(1, a, b) # whatever a and b are for my posterior
  ytilde_s <- rpois(n=218, theta = theta_s)
  t_s[s] <- mean(ytilde_s == 0) # compute test stat
}

## then visualize histogram of t_s
```

PPCs example

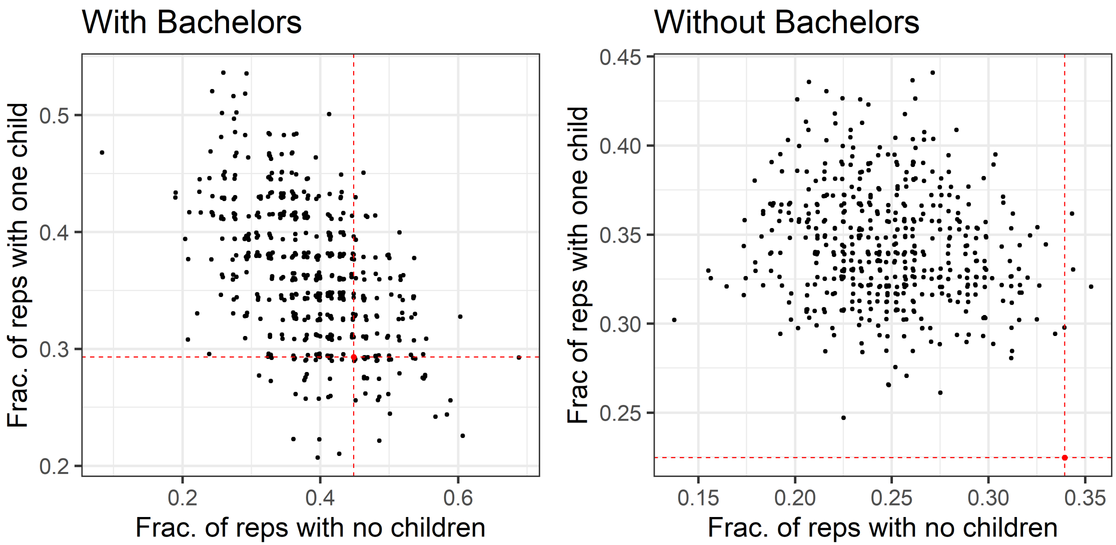


$$\Pr(T^{\text{rep}} > T^{\text{obs}}) = 0.006$$

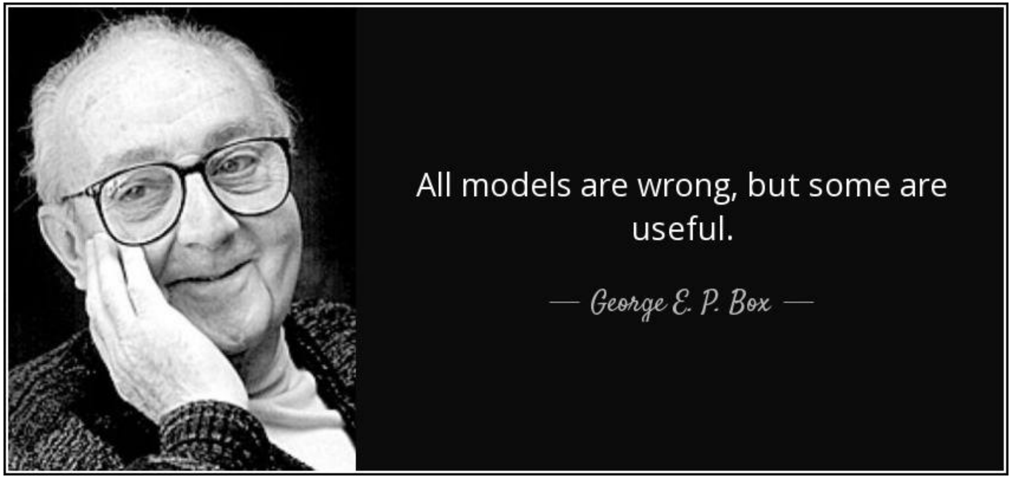
PPCs example

- Model checking both groups
- Look at fit for two different test statistics:
  - Fraction with no children
  - Fraction with one child

Poisson example



All models are wrong



If the model is "wrong", how can we improve it?

PPCs and Model Refinement

- How might we refine the model?
- What might be a better data generating process?
- How do we choose test statistics to investigate? What other statistics might be worth checking?

Sampling strategies

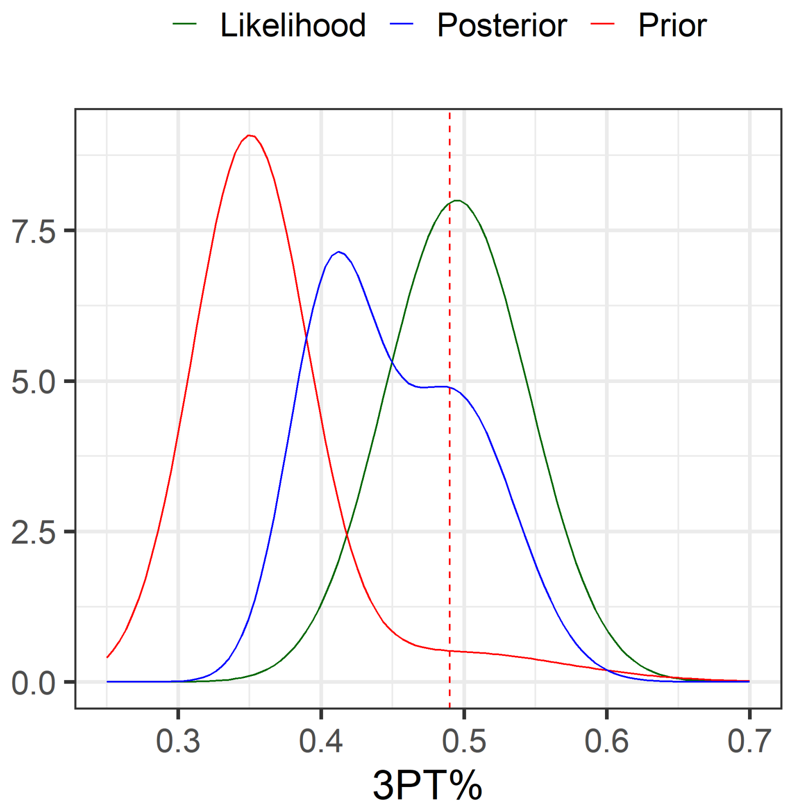
Example: non-conjugate Prior Distributions

- Conjugate prior distributions make the math / concepts easy but no reason they should reflect our true prior belief
- In theory, want to build the best model possible, not one that is convenient
- If we choose a non-conjugate prior distribution, then the posterior distribution may have a "complicated" density. Need Monte Carlo to estimate posterior summaries.

Estimating Robert Covington's skill

- Binomial likelihood is $p(y \mid \theta) \propto \theta^y(1-\theta)^{n-y}$
- Assume I use a mixture normal prior is $p(\theta) = 0.9f_1(\theta) + 0.1f_2(\theta)$
  - f_1 is $N(\mu = 0.35, \sigma = 0.04)$ and f_2 is $N(\mu = 0.5, \sigma = 0.08)$

Example: estimating shooting skill in basketball



How can we compute the posterior mean and probability interval?

Sampling strategies

- Monte Carlo methods assume that we have a method for easily generating a pseudo-random number!
- If the `R` includes the appropriate random number generating function, e.g. `rnorm` then Monte Carlo is easy
- If not, we need to be more clever about how we generate samples.
    - Inversion Sampling (works for univariate)
    - Grid sampling (works for low dimensional problems)
    - Rejection sampling (can be good for low dimensional problems)
    - Importance sampling (useful in some cases, hard in general)
    - Markov Chain Monte Carlo

Sampling strategies

- Reminder: why sampling? We want to approximate difficult integrals.
  - We can represent expected values, probabilities, quantiles etc all as integrals
- In Bayesian stats we usually know how to write down the (proportional) posterior density: $L(\theta)P(\theta)$
- Knowing the pdf does not mean by default we know to sample from that distribution!
- If we can devise a way to sample

Probability Integral Transform

- Suppose that a random variable, Y has a continuous distribution for with CDF is F_Y.
- Then the random variable $U = F_Y(Y)$ has a uniform distribution
    - This is known as the "probability integral transform PIT"
- By taking the inverse of F_Y we have $F^{-1}(U) = Y$

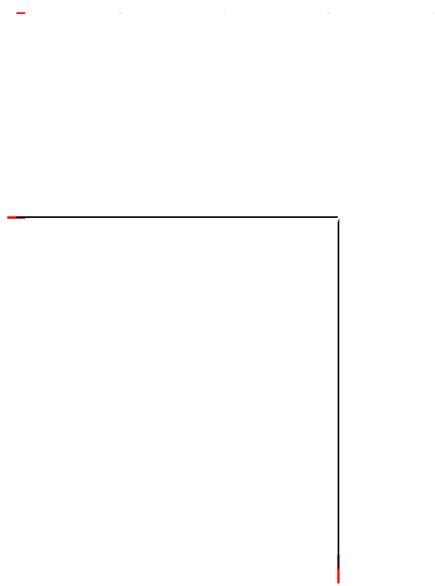
Inversion Sampling

The inverse transform sampling method works as follows:

1. Generate a random number u from $\text{Unif}[0, 1]$
2. Find the inverse of the desired CDF, e.g. $F_Y^{-1}(u)$.
3. Compute $y = F_Y^{-1}(u)$. y is now a sample from the desired distribution.

Inversion Sampling

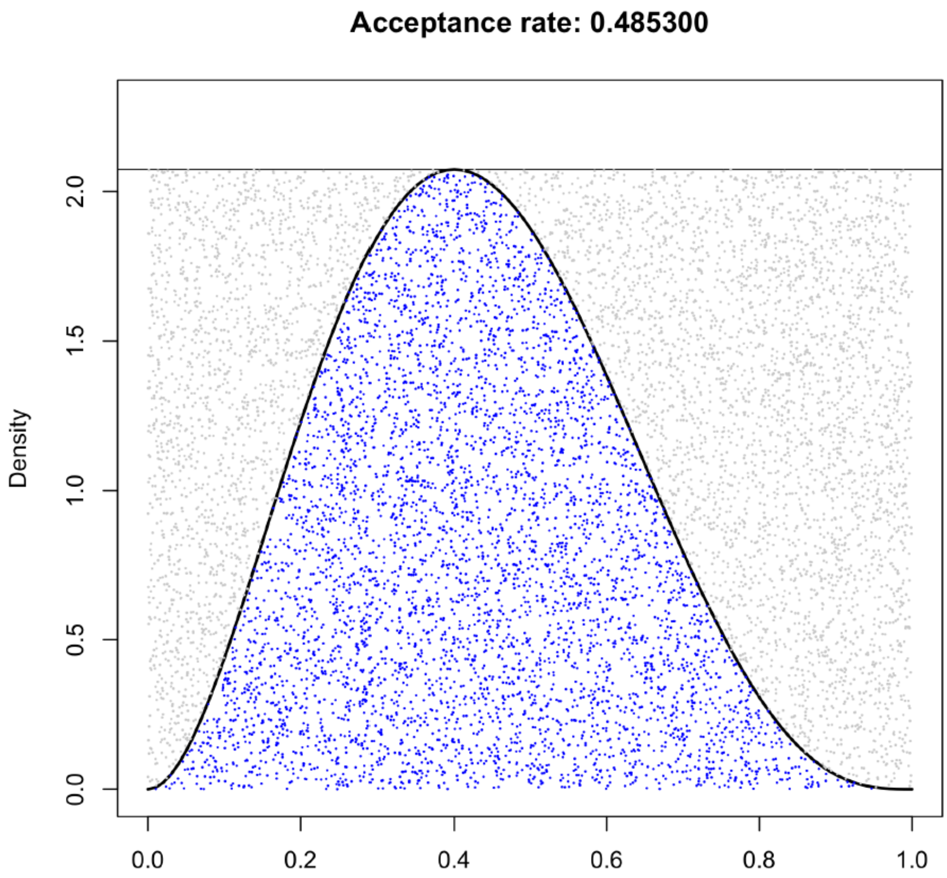
Animation Demo



Inversion Sampling

- Inversion sampling can be a fast and simple way to sample from a distribution
- Only effective if we know the inverse-CDF and can easily compute it
- This is a big challenge in practice. For example, even the normal distribution has a CDF, Φ, which cannot be expressed analytically.
    - Shifts from one hard problem (sampling) to another (computing an integral)
    - Need alternatives!

Rejection Sampling



Rejection Sampling

1. Choose a proposal density, $q(\theta)$ that we can easily sample from (e.g. uniform or normal) such that:

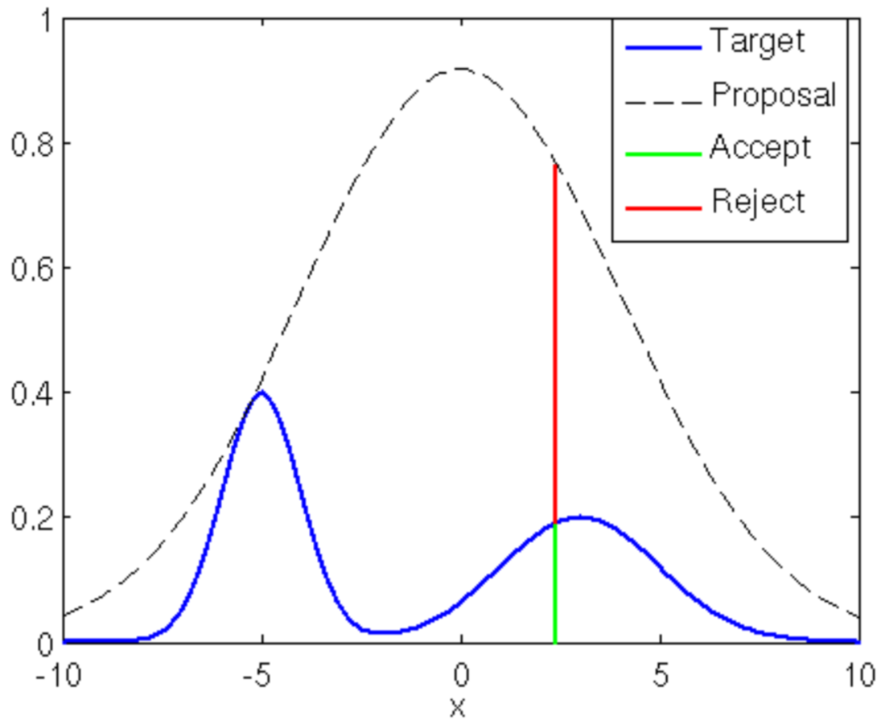
2. Find $M = \max \frac{p(\theta|y)}{q(\theta)}$

    - If $M = \infty$ then q cannot be used as a proposal distribution
    - If M is finite, $Mq(\theta)$ "envelopes" $p(\theta\ y)$

3. Draw a sample, $\theta^{(s)}$ from $q(\theta)$

4. Accept $\theta^{(s)}$ as a draw from $p(\theta \mid y)$ with probability $\frac{p(\theta^{(s)}|y)}{Mq(\theta^{(s)})}$

Rejection Sampling



Markov Chain Monte Carlo

- Markov Chain Monte Carlo (MCMC)
- More effective approach to sampling from multi-parameter distributions
- Samples in MCMC are **not** independent samples

MCMC Sampling with Stan

Stan

"A state-of-the-art platform for statistical modeling and high-performance statistical computation."

Monte Carlo Sampling with Stan

- Stan is a "probabilistic programming language" with its own simple syntax
- R interface to Stan via `cmdstanr` package
- Define parameters θ and datatypes y
- Provide sampling model $p(y \mid \theta)$ and prior $p(\theta)$
- Stan translates model syntax into the log posterior density $\log(p(\theta \mid y))$ and automatically generates samples from this density

MCMC with Stan

```
data {
  int<lower=0> makes;
  int<lower=makes> attempts;
} parameters {
  real<lower=0, upper=1> theta;
}
model {
  makes ~ binomial(attempts, theta);
  theta ~ uniform(0, 1);
}
```

MCMC with Stan

```
# BEGIN SOLUTION
stan_fit <- binomial_mod$sample(data=list(makes = 50, attempts=101
                                show_messages = FALSE, chains=1, r
```

```
## Running MCMC with 1 chain...
##
## Chain 1 finished in 0.0 seconds.
```

```
samples <- stan_fit$draws(format="df")
# END SOLUTION

post_mean <- mean(samples$theta) # SOLUTION
print(post_mean)
```

```
## [1] 0.4969143
```

Summary

- Monte Carlo methods for computing integrals
    - posterior means and variances
    - posterior probabilities
- Posterior predictive checking
    - Asks: is data simulated from the model consistent with what I observe?
    - If not, revise and refine the model
- Getting random samples is not always easy
    - Inversion sampling useful in limited circumstances
    - More sophisticated techniques needed, in general