Bayesian Analysis
A brief introduction

Michael L. Thompson, Ph.D.

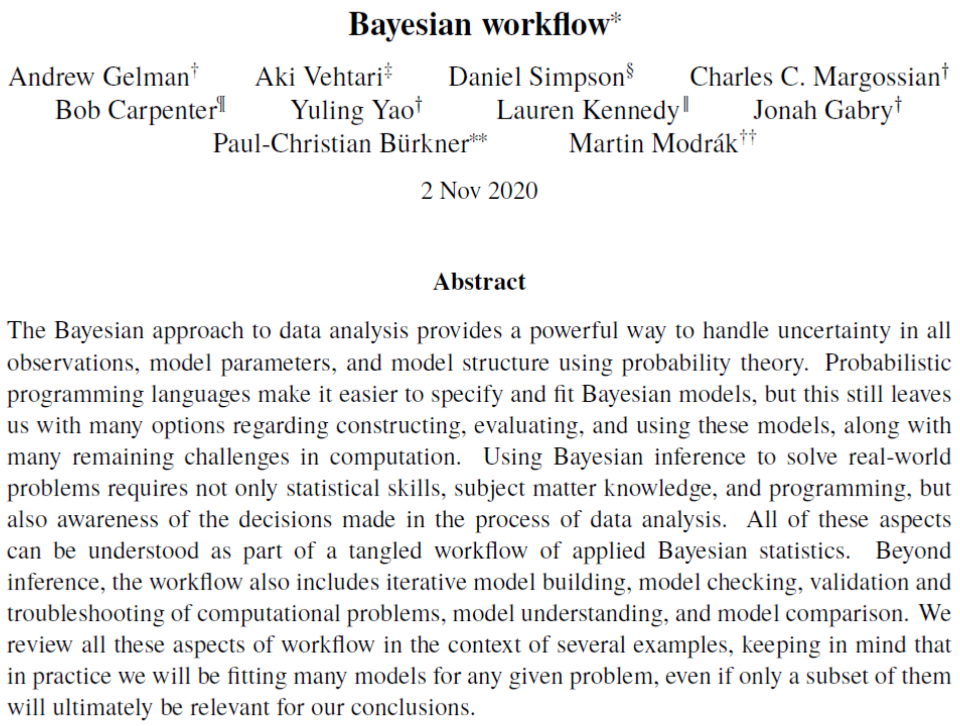
Nov. 17, 2020

(**EXCERPTS**)

Bayesian Workflow

- "Bayesian Workflow", Prof. Andrew Gelman, *et al.*
    1. **Specify Model**
        - Motivated by Exploratory Analysis leveraging visualizations
        - Weakly informative Priors
    2. **Perform Prior Predictive Checking**
        - Facilitated by visualizations
    3. **Estimate Posterior (Fit Model)**
        - Performed by Markov Chain Monte Carlo simulation
    4. **Validate Computation**
        - Diagnostic checking
    5. **Evaluate Model**
        - Leverage Posterior Predictive Checking
    6. **Compare Models**
    7. **Apply Model**
        - Decision Making, etc.

..but not a linear process



Bayesian workflow[*]

Andrew Gelman[†] Aki Vehtari[‡] Daniel Simpson[§] Charles C. Margossian[†]
Bob Carpenter[¶] Yuling Yao[†] Lauren Kennedy[‖] Jonah Gabry[†]
Paul-Christian Bürkner[**] Martin Modrák[††]

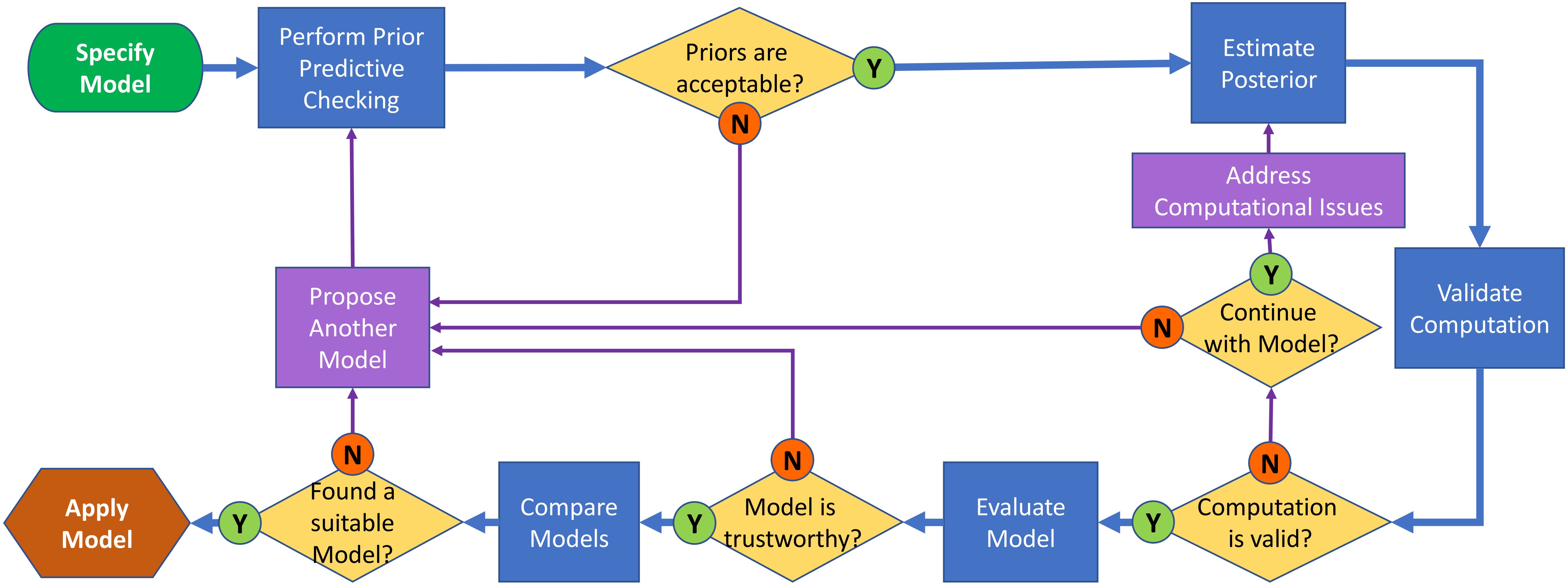
2 Nov 2020

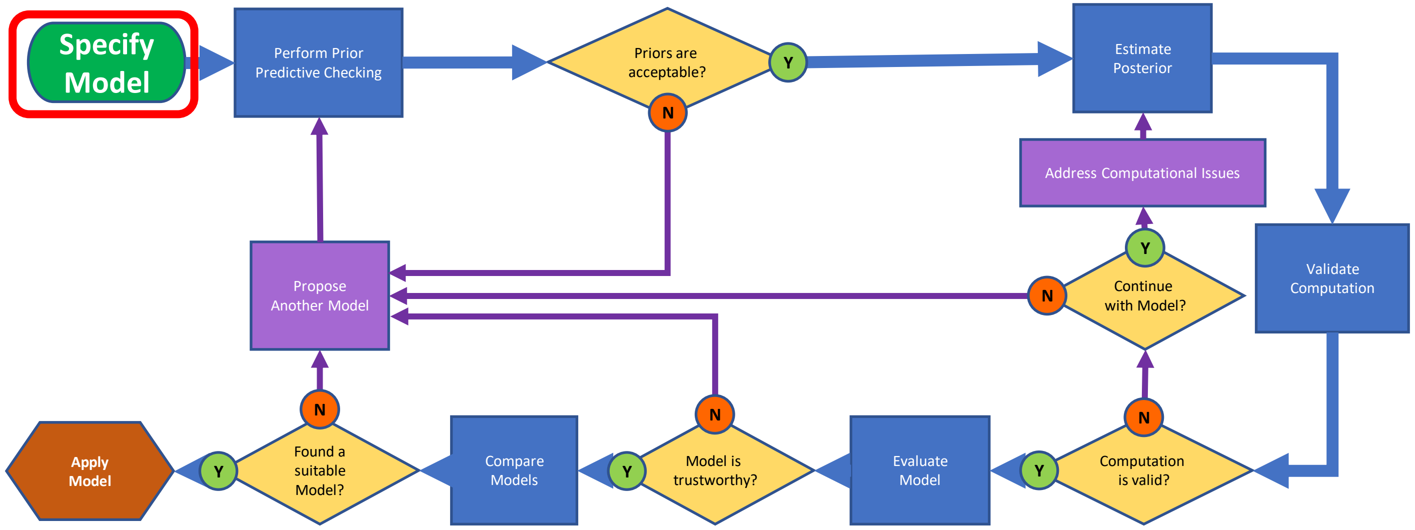
Abstract

The Bayesian approach to data analysis provides a powerful way to handle uncertainty in all observations, model parameters, and model structure using probability theory. Probabilistic programming languages make it easier to specify and fit Bayesian models, but this still leaves us with many options regarding constructing, evaluating, and using these models, along with many remaining challenges in computation. Using Bayesian inference to solve real-world problems requires not only statistical skills, subject matter knowledge, and programming, but also awareness of the decisions made in the process of data analysis. All of these aspects can be understood as part of a tangled workflow of applied Bayesian statistics. Beyond inference, the workflow also includes iterative model building, model checking, validation and troubleshooting of computational problems, model understanding, and model comparison. We review all these aspects of workflow in the context of several examples, keeping in mind that in practice we will be fitting many models for any given problem, even if only a subset of them will ultimately be relevant for our conclusions.

Bayesian Workflow

(adapted from Fig. 1, Gelman *et al*.)





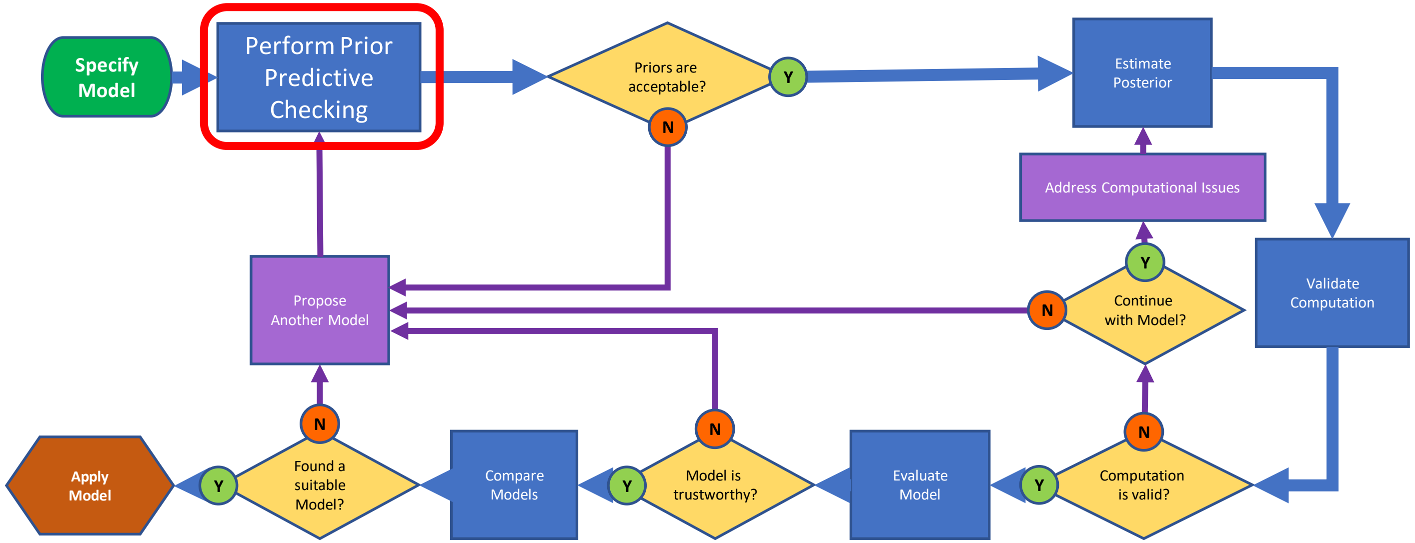
Bayesian Workflow

1. Specify Model

- Problem Statement*
  - We wish to estimate the success rate π of our promotional ad campaign amongst our target population. We have data from a small test amongst N=16 customers, in which we succeeded with n=6.
- Generative Model
  - Base Model: $n \sim \mathrm{Binomial}(N,\pi)$; prior $\mathrm{p}(\pi)$
- Weakly Informative Priors
  - Previous experience tells us that typically $\pi \sim 0.30$ and almost always our campaigns have $\pi < 0.80$; we'll use $\pi \sim \mathrm{Beta}(\alpha,\beta)$ and set the hyperparams assuming μ=0.30 and let 97.5%-ile = 0.80; so use α=0.96, β=2.24

Just as with the simple Mosaic plot examples, this gives us the joint distribution, but as a simulator rather than a 2D plot. And applying Bayes Rule is, again, as simple as selecting the generated hypotheses (parameter values) that coincide with the same generated success rates as the evidence, in this case n.

*Adapted from Rasmus Baath: "Bayesian Analysis Tutorial, Part 1: What"; YouTube (2017).

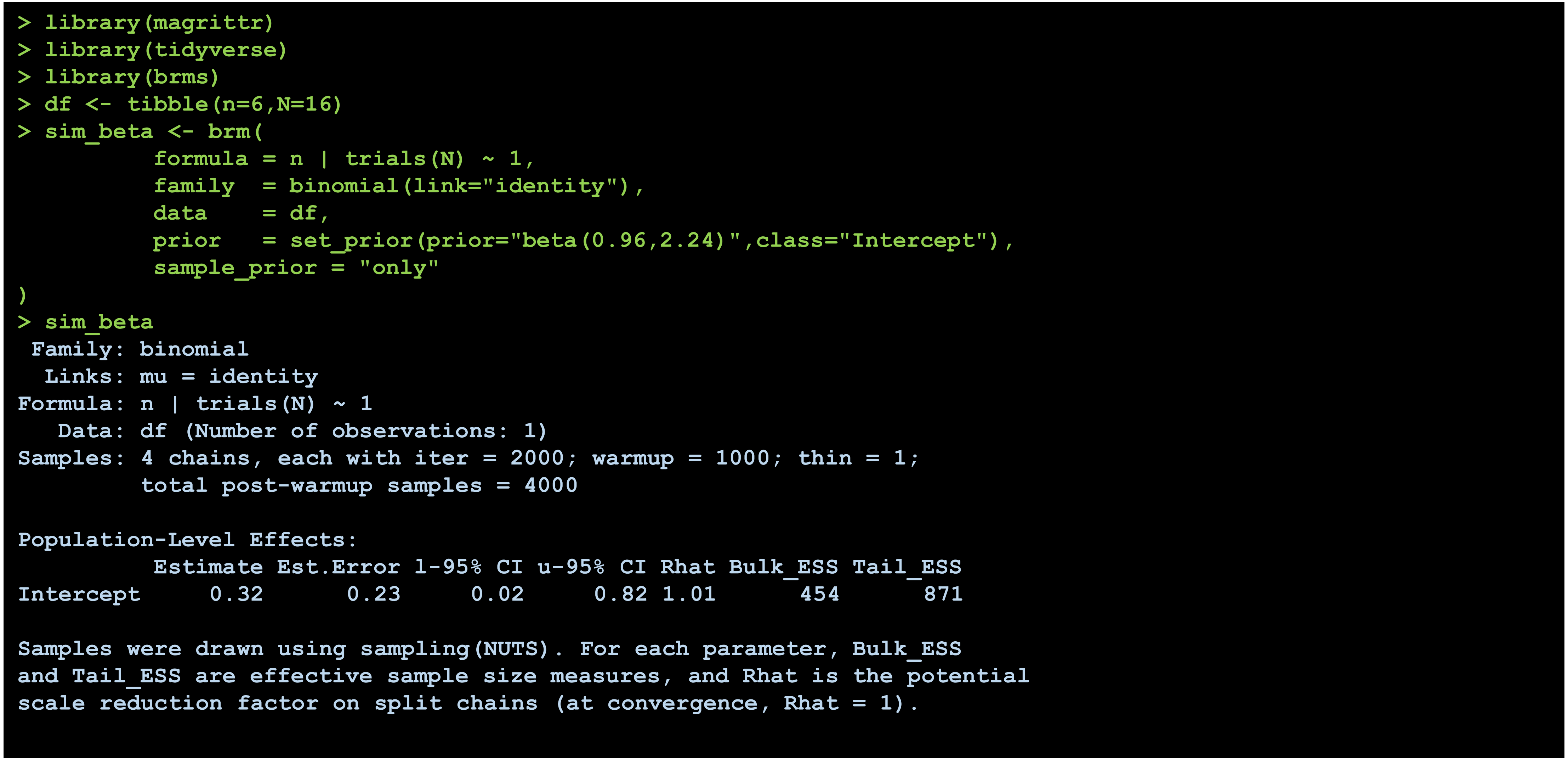


Bayesian Workflow
2. Perform Prior Predictive Checking

- Simulate Response Data & Assess How Plausible They Are
  - I follow the advice of Gelman & colleagues* in picking “weakly informative” priors. These are priors that when used in generating “fake” data (i.e., simulating from the joint distn.) yield predictions of the responses that are at least plausible, even if not entirely likely.
- In Practice
  - I do Bayesian Analysis in R using package `brms` (Bayesian Regression Modeling with Stan).

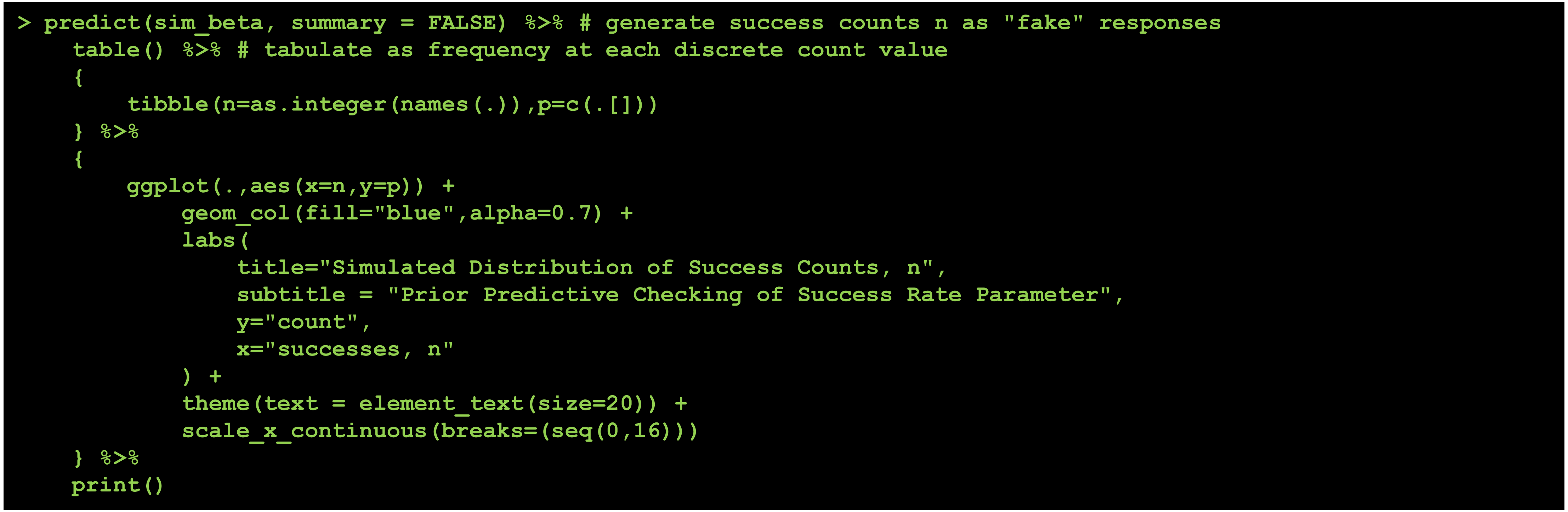
*See the papers at these 2 links:
- Gelman et al. “The Prior Can Often Only Be Understood in the Context of the Likelihood”
- Gabry et al. “Visualization in Bayesian Workflow”

Prior Predictive Checking

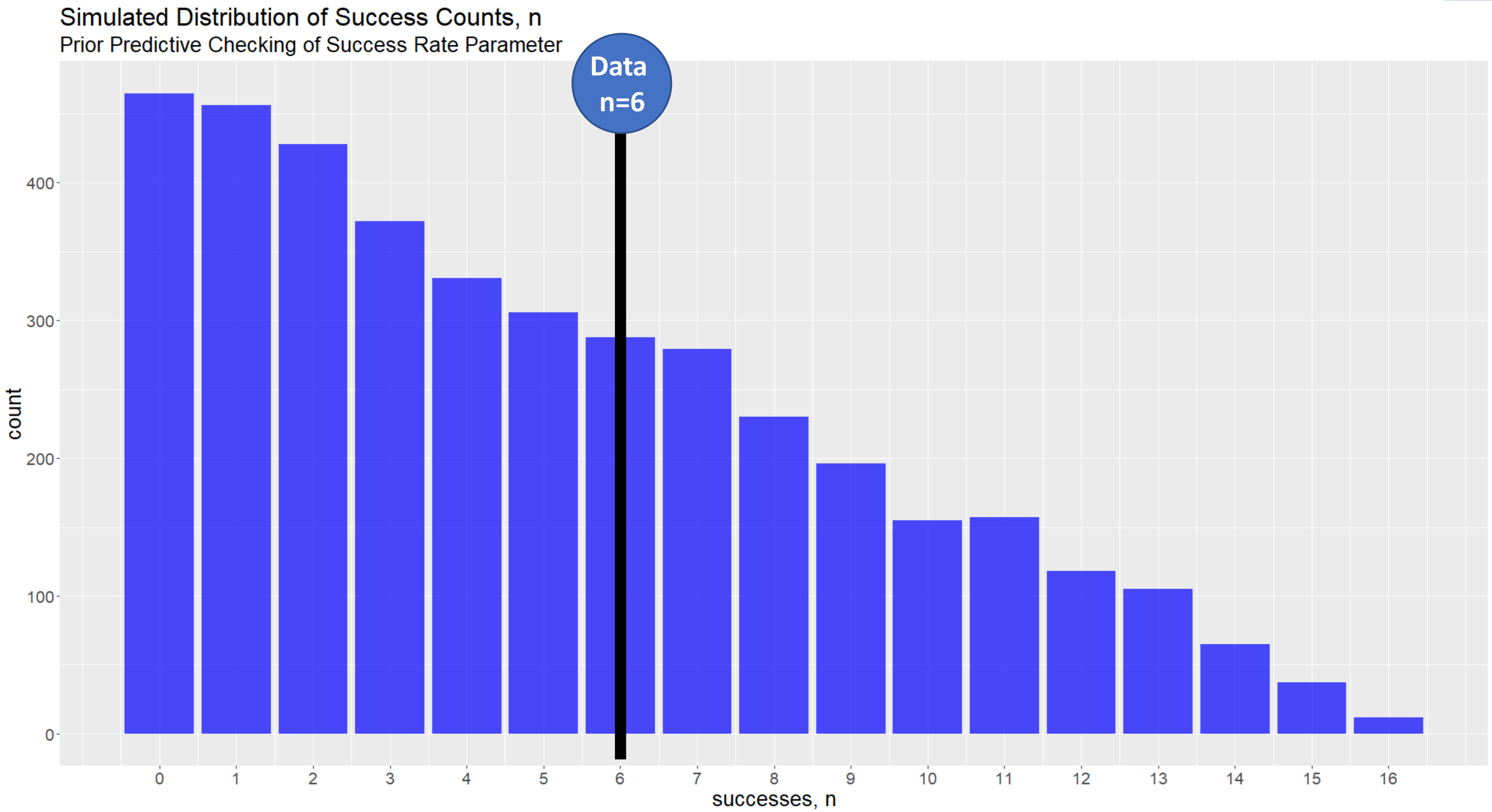


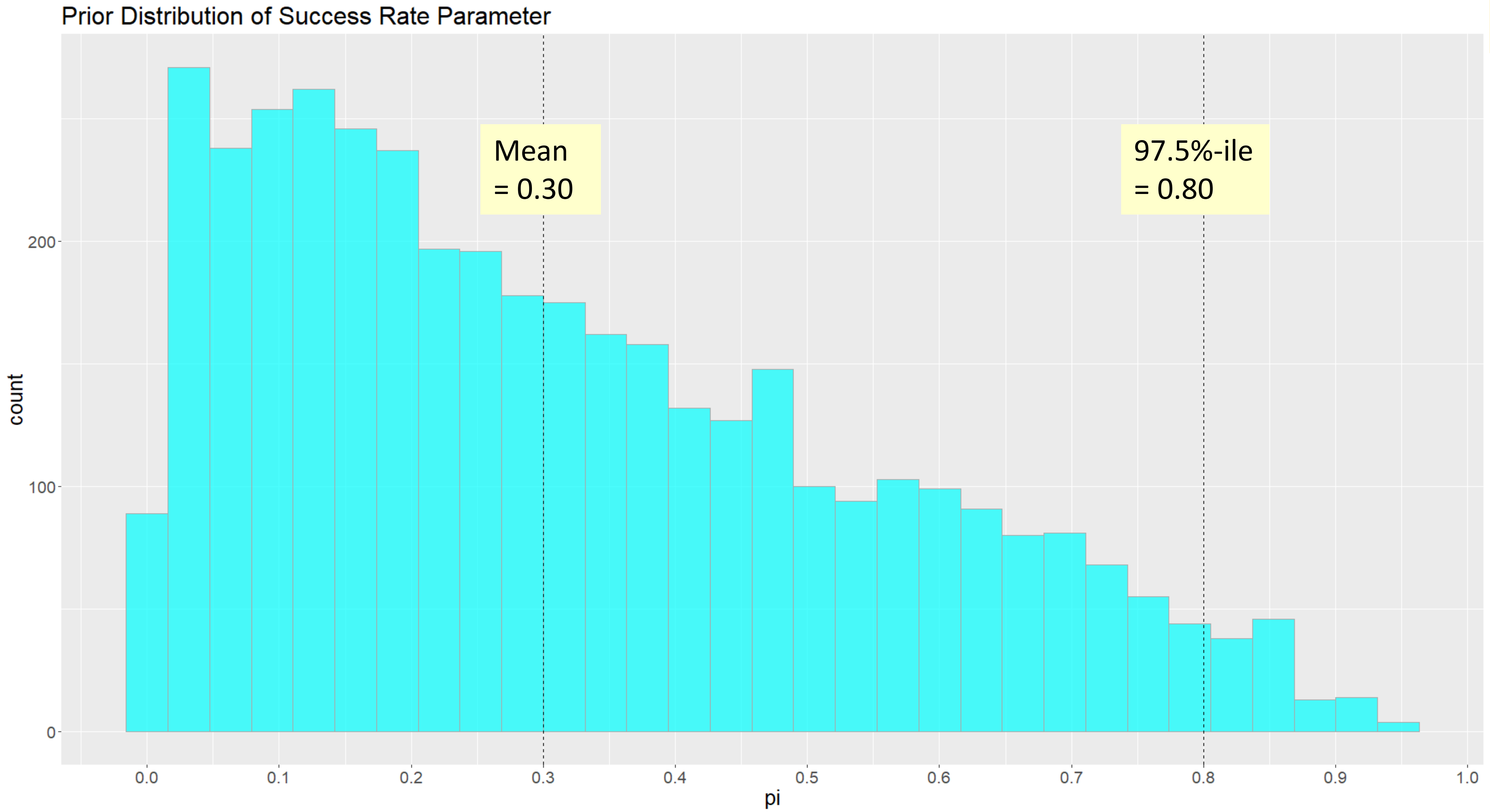
```
> library(magrittr)
> library(tidyverse)
> library(brms)
> df <- tibble(n=6,N=16)
> sim_beta <- brm(
          formula = n | trials(N) ~ 1,
          family  = binomial(link="identity"),
          data    = df,
          prior   = set_prior(prior="beta(0.96,2.24)",class="Intercept"),
          sample_prior = "only"
)
> sim_beta
 Family: binomial
  Links: mu = identity
Formula: n | trials(N) ~ 1
   Data: df (Number of observations: 1)
Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup samples = 4000

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.32      0.23     0.02     0.82 1.01      454      871

Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
```

Prior Predictive Checking



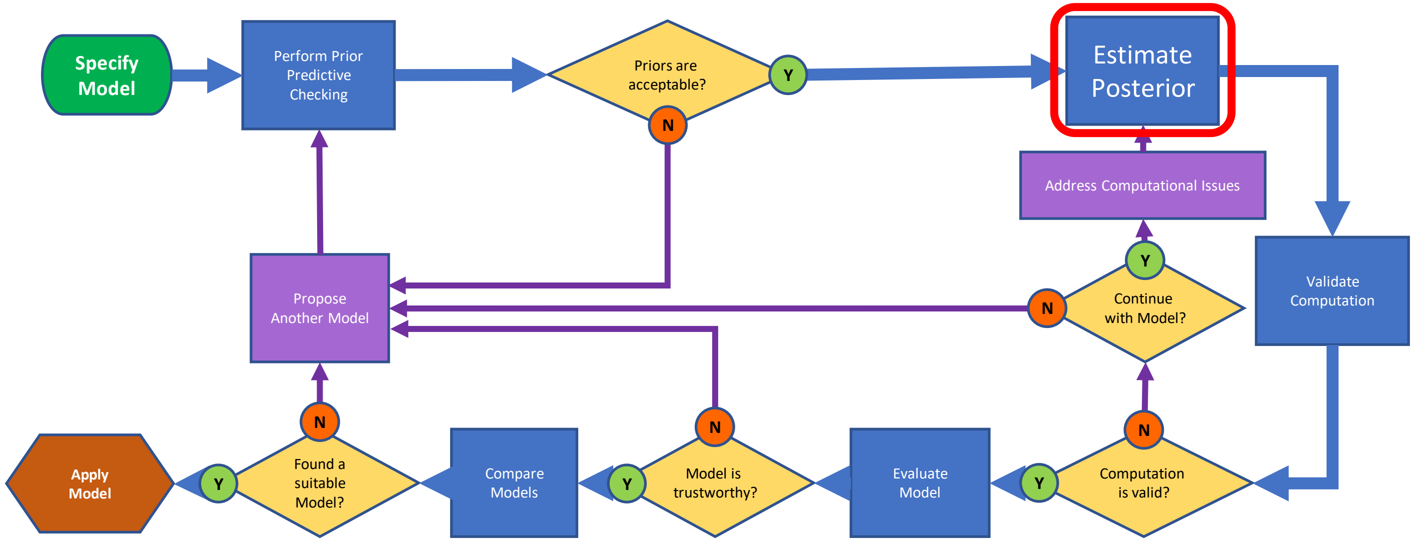
```
> predict(sim_beta, summary = FALSE) %>% # generate success counts n as "fake" responses
    table() %>% # tabulate as frequency at each discrete count value
    {
        tibble(n=as.integer(names(.)),p=c(.[]))
    } %>%
    {
        ggplot(.,aes(x=n,y=p)) +
            geom_col(fill="blue",alpha=0.7) +
            labs(
                title="Simulated Distribution of Success Counts, n",
                subtitle = "Prior Predictive Checking of Success Rate Parameter",
                y="count",
                x="successes, n"
            ) +
            theme(text = element_text(size=20)) +
            scale_x_continuous(breaks=(seq(0,16)))
    } %>%
    print()
```


Simulated Distribution of Success Counts, n
Prior Predictive Checking of Success Rate Parameter
Data
n=6
count
0
100
200
300
400
successes, n
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16



Mean
= 0.30

97.5%-ile
= 0.80

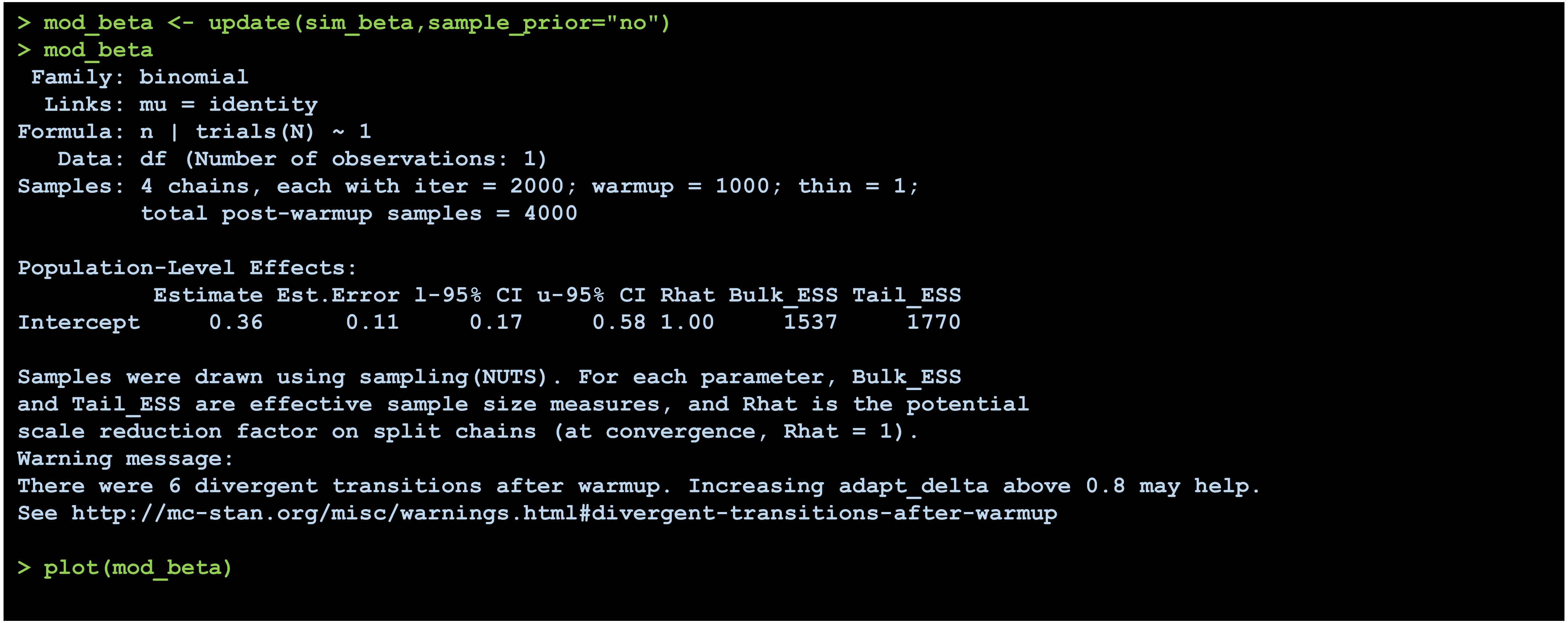


Bayesian Workflow

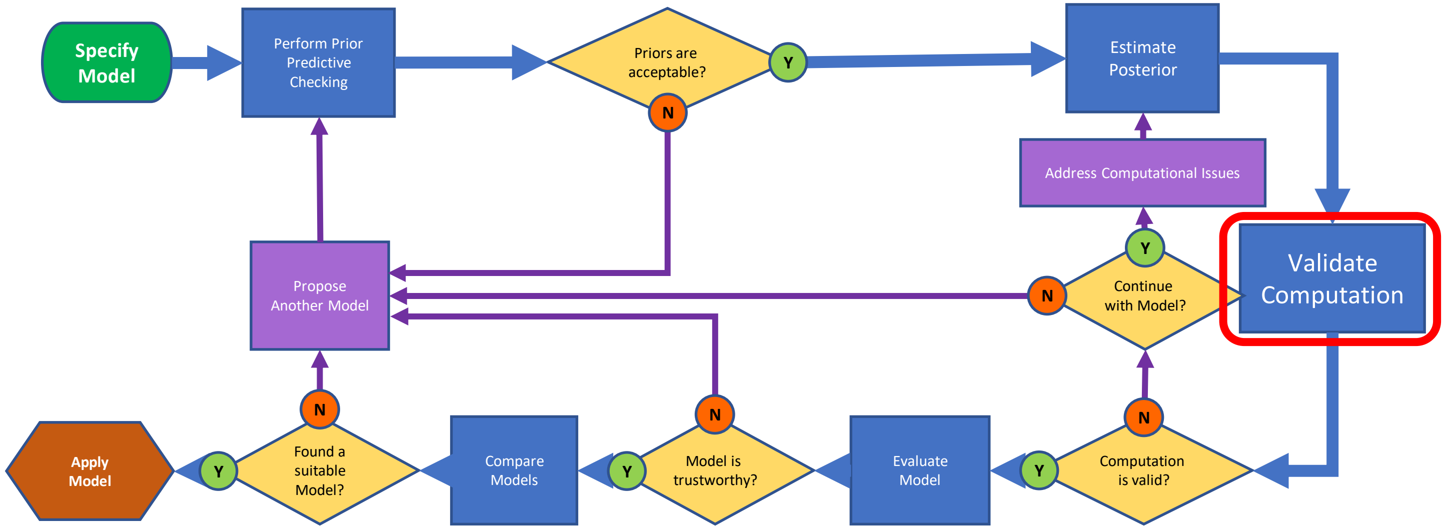
3. Estimate Posterior

- Fitting a Model
  - In Bayesian Analysis, fitting the model comes down to estimating the posterior distribution of the parameters given the data. This is conceptually like simulating responses from the joint and filtering out those simulated results that don’t match the observed data for the responses and retaining the parameters of the model that generated the retained predicted responses. However, doing it that way (as rejection sampling) is only practical on simple problems. We use more advanced algorithms, like MCMC below, for modern real-world problems.
- Markov Chain Monte Carlo (MCMC) Simulation
  - The *de facto* standard in sampling from the posterior distribution is Hamiltonian Monte Carlo with the “No-U-Turn” Sampler (HMC-NUTS) as implemented in the Stan probabilistic programming language.

Estimating Posterior



```
> mod_beta <- update(sim_beta,sample_prior="no")
> mod_beta
 Family: binomial
  Links: mu = identity
Formula: n | trials(N) ~ 1
   Data: df (Number of observations: 1)
Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup samples = 4000

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.36      0.11     0.17     0.58 1.00     1537     1770

Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
Warning message:
There were 6 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
See http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

> plot(mod_beta)
```

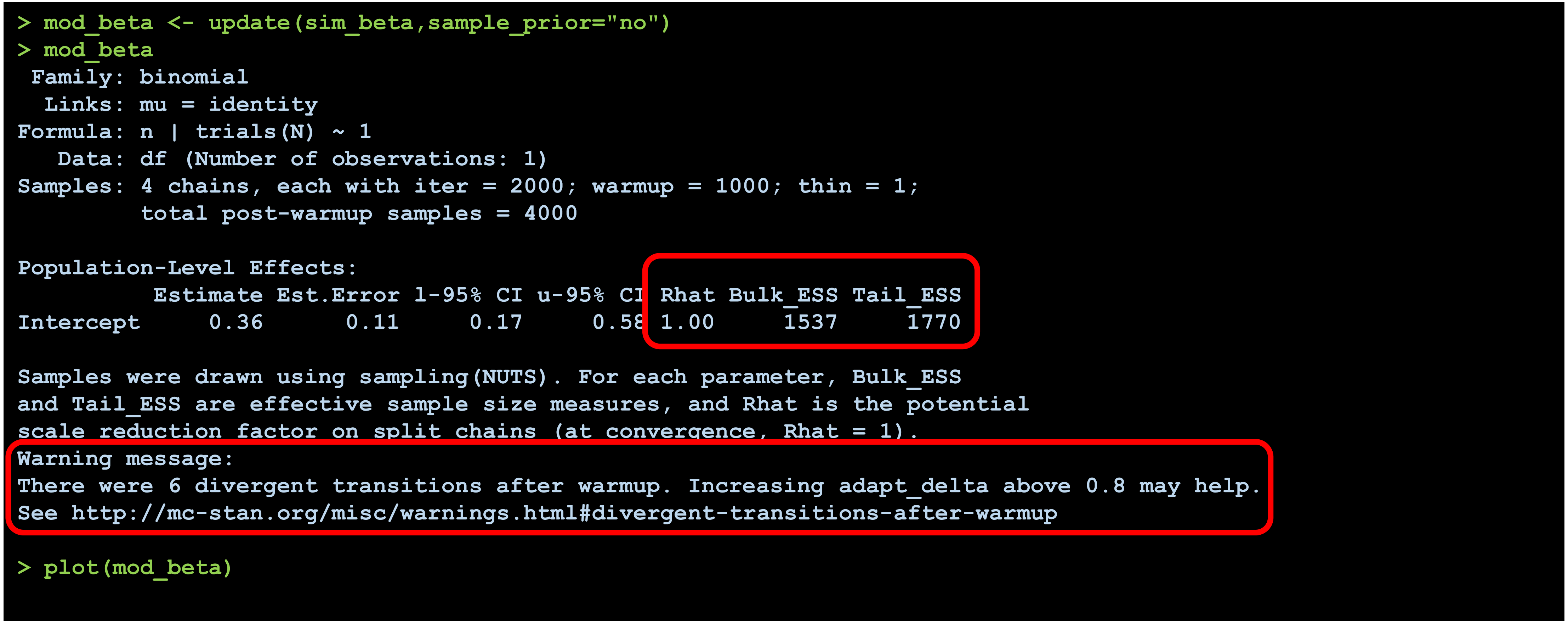


Bayesian Workflow

4. Validate Computation

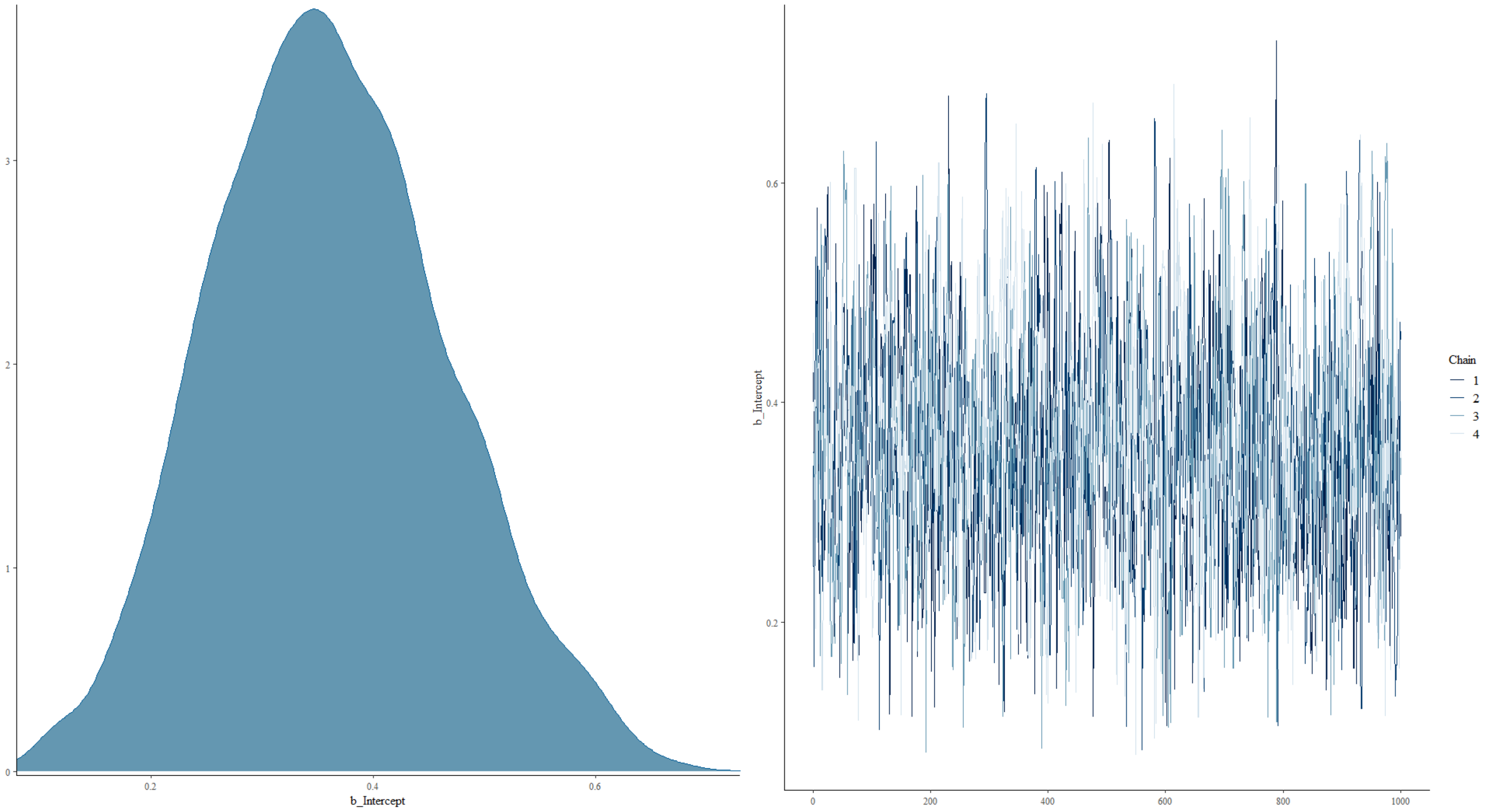
- Assessing Convergence
  - Before we evaluate the model for its predictive or explanatory ability, we must be sure we have achieved valid computation, i.e. the simulation converged to the actual posterior probability distribution.
- Diagnostics
  - Stan and the R packages based upon it provide us with a wealth of diagnostics, which we use to validate the computation. See the explanations & tips here: “Brief Guide to Stan’s Warnings: Runtime Warnings”.

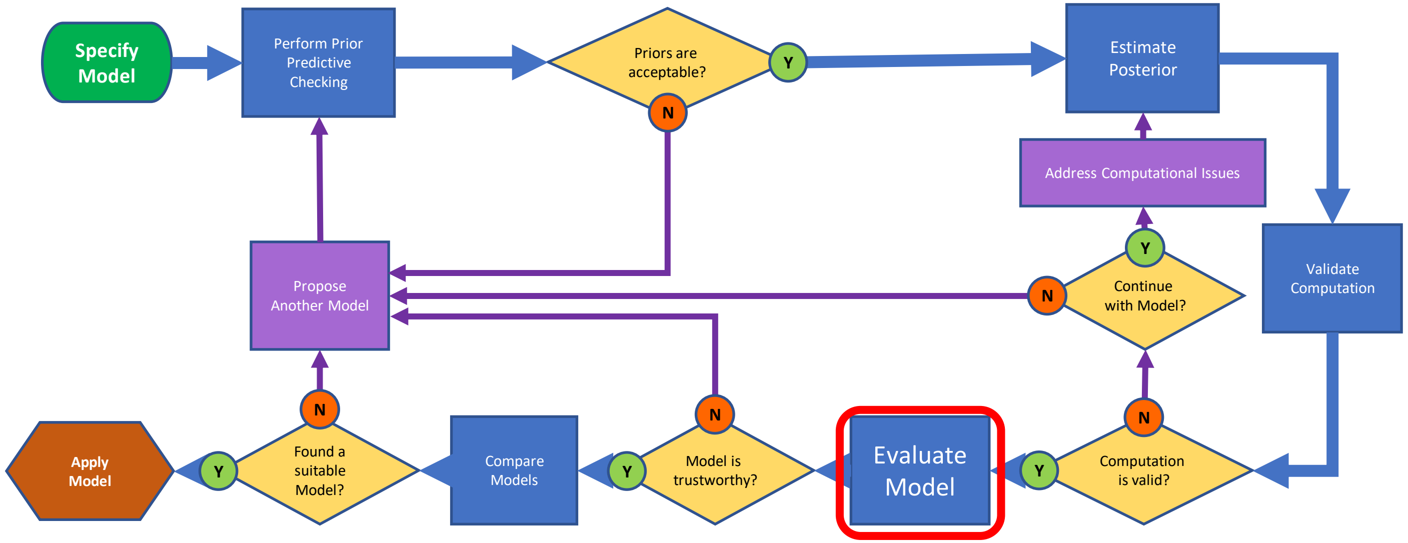
Checking Diagnostics & Warnings



```
> mod_beta <- update(sim_beta,sample_prior="no")
> mod_beta
 Family: binomial
  Links: mu = identity
Formula: n | trials(N) ~ 1
   Data: df (Number of observations: 1)
Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup samples = 4000

Population-Level Effects:
          Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     0.36      0.11     0.17     0.58 1.00     1537     1770

Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
and Tail_ESS are effective sample size measures, and Rhat is the potential
scale reduction factor on split chains (at convergence, Rhat = 1).
Warning message:
There were 6 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
See http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

> plot(mod_beta)
```

Posterior Density & Monte Carlo Trace Plot for success rate parameter π

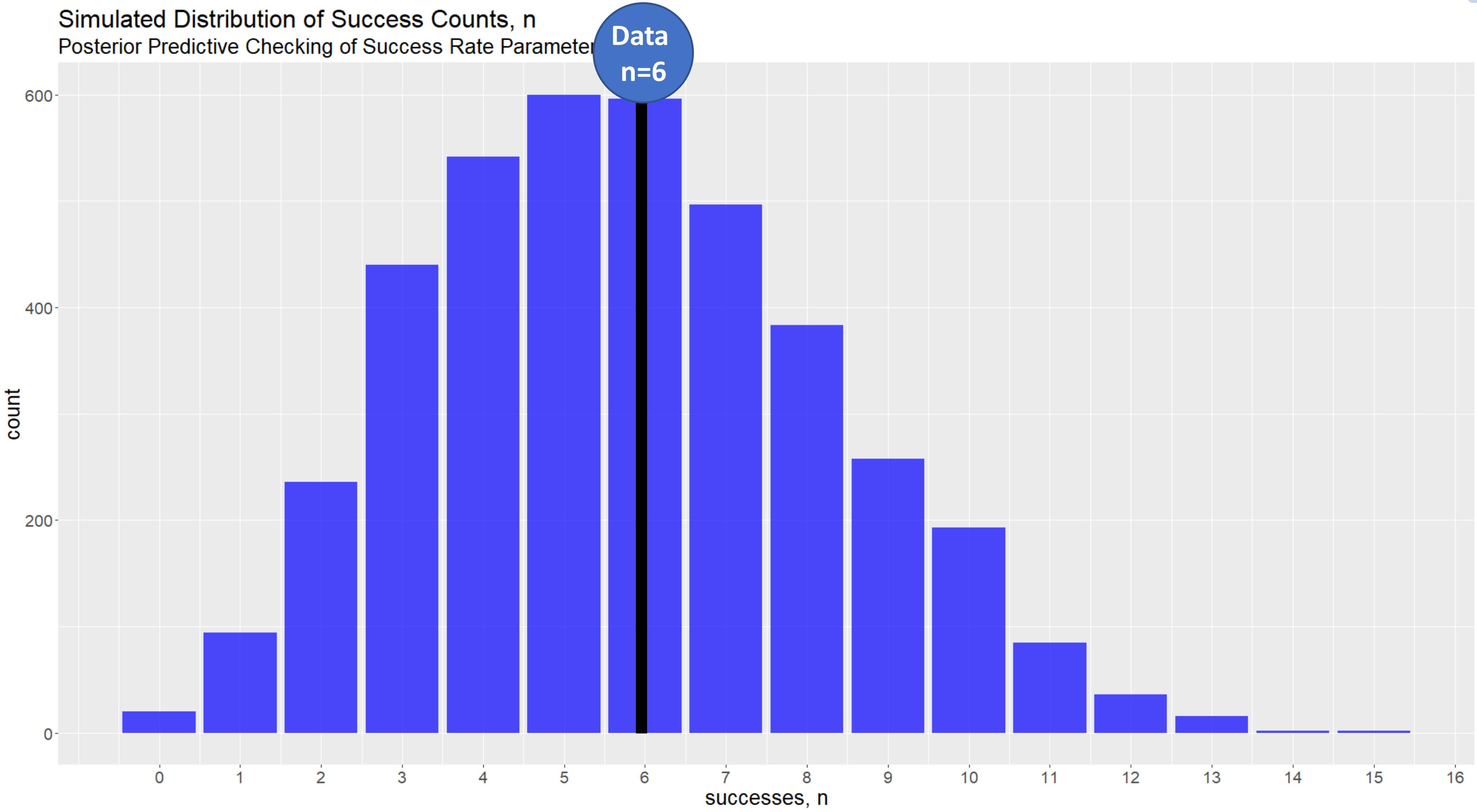


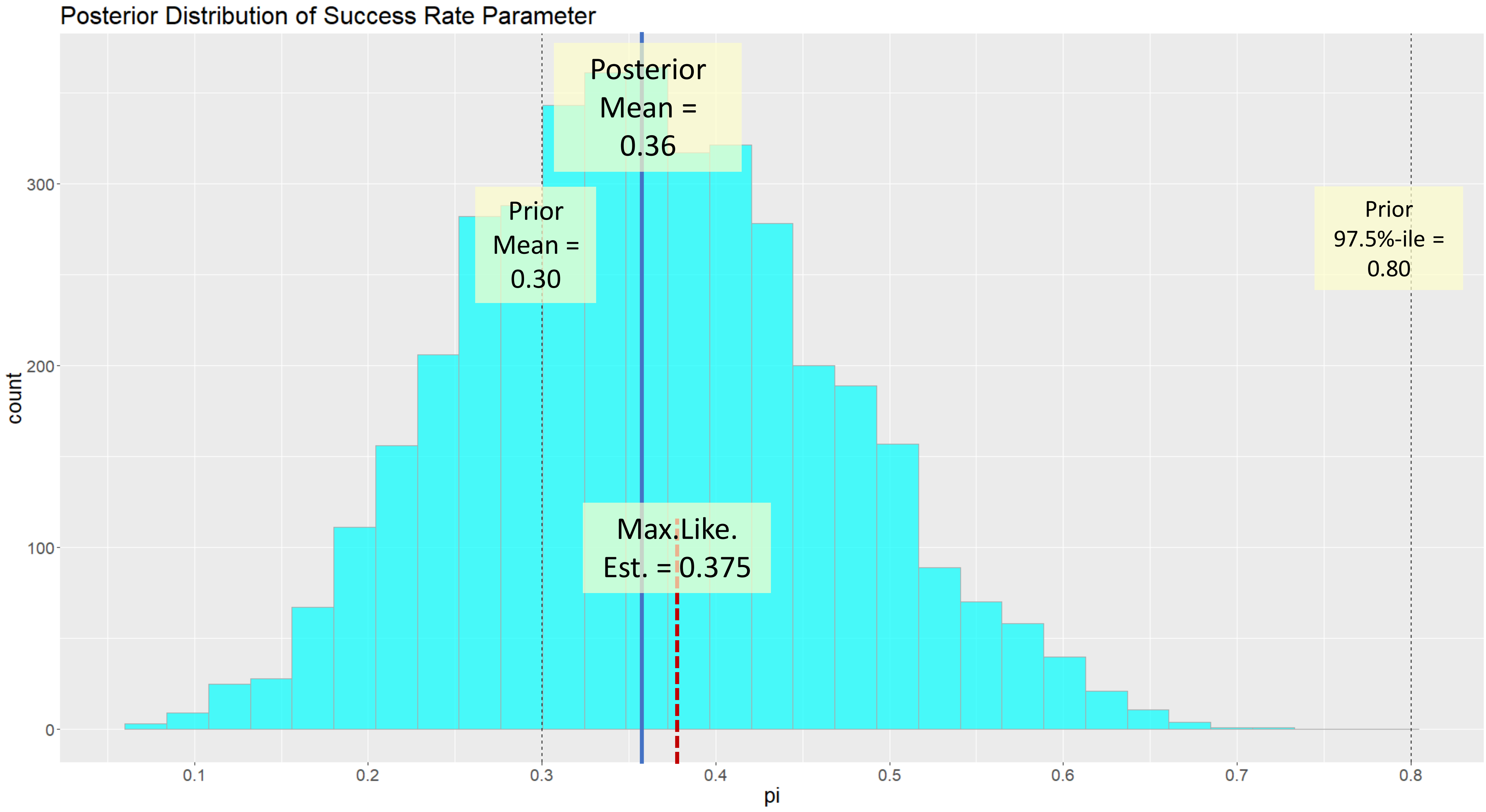


Bayesian Workflow

5. Evaluate Model

- Posterior Predictive Checking
  - Analogous to assessing the plausibility of simulated data from the prior distributions during Prior Predictive Checking, we do a similar check, now explicitly using the data to assess the quality of the posterior predicted responses.
- Leave-One-Out (LOO) Validation
  - Much recent work has been done on assessing the out-of-sample predictive ability of models using the posterior distribution. The "loo" calculation of the expected value of the log-posterior (elpd) is an efficient way of evaluating models.


Simulated Distribution of Success Counts, n
Posterior Predictive Checking of Success Rate Parameter
Data
n=6
count
600
400
200
0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
successes, n

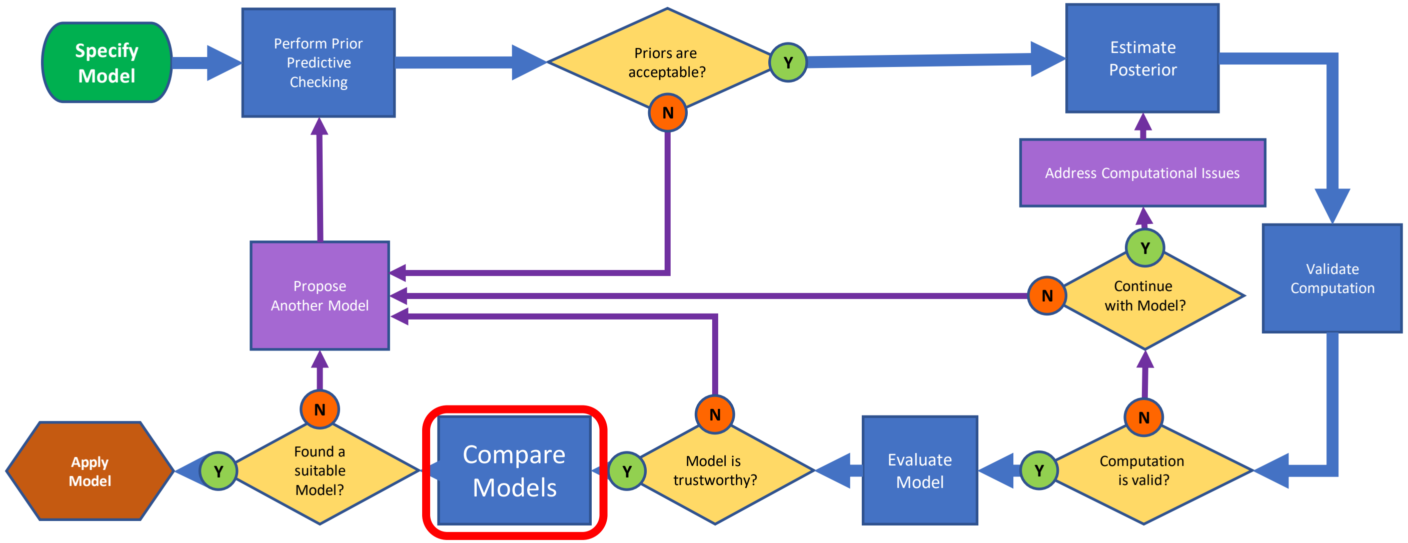


Posterior Mean = 0.36

Prior Mean = 0.30

Prior 97.5%-ile = 0.80

Max.Like. Est. = 0.375



Bayesian Workflow
6. Compare Models

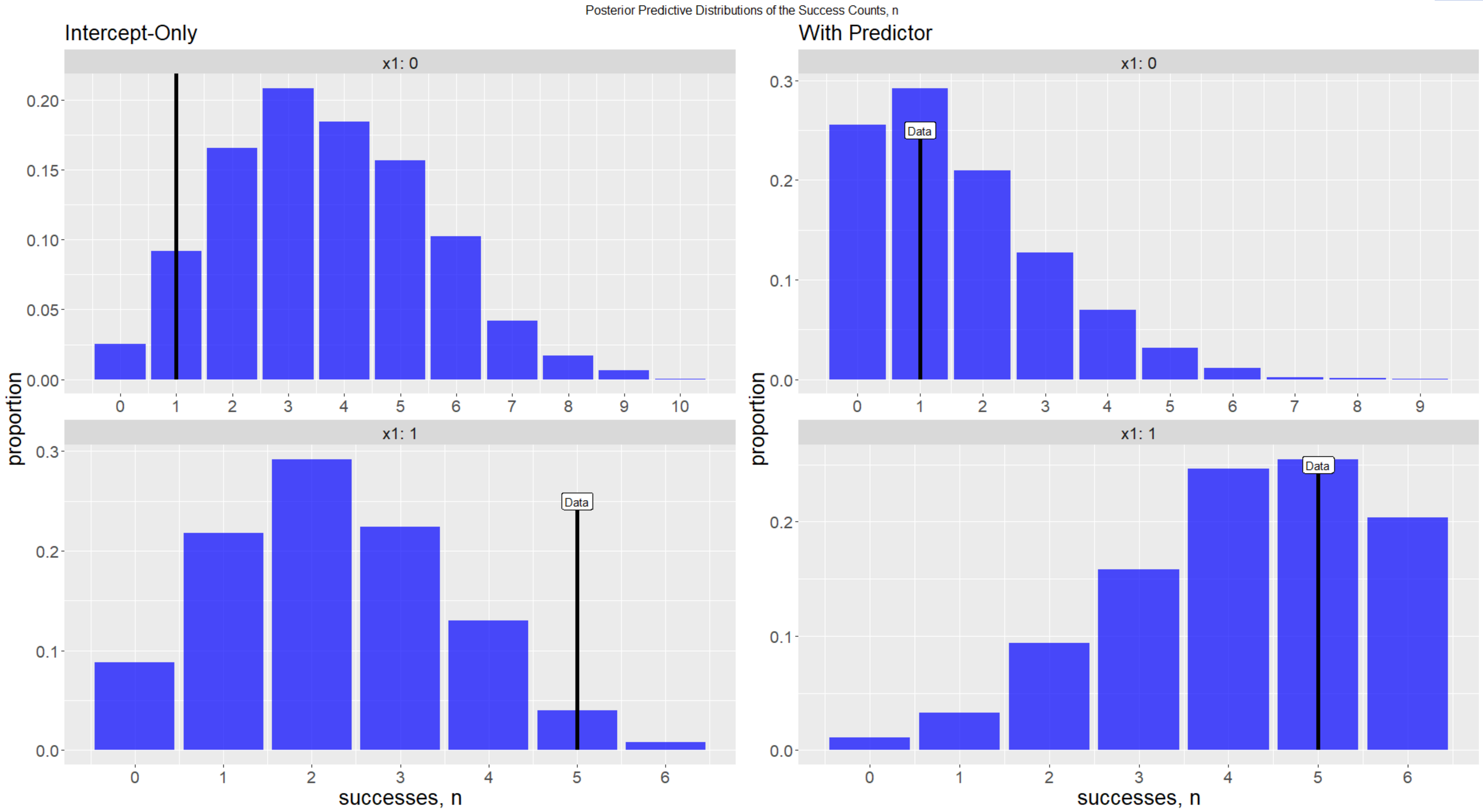
- Loo and Pointwise Comparison
  - The model with a significantly better loo-computed elpd is typically preferred. For models with comparable elpd, we usually pick the simpler of the model or the model that agrees most closely with our domain knowledge – i.e. simplicity and expertise break the ties.

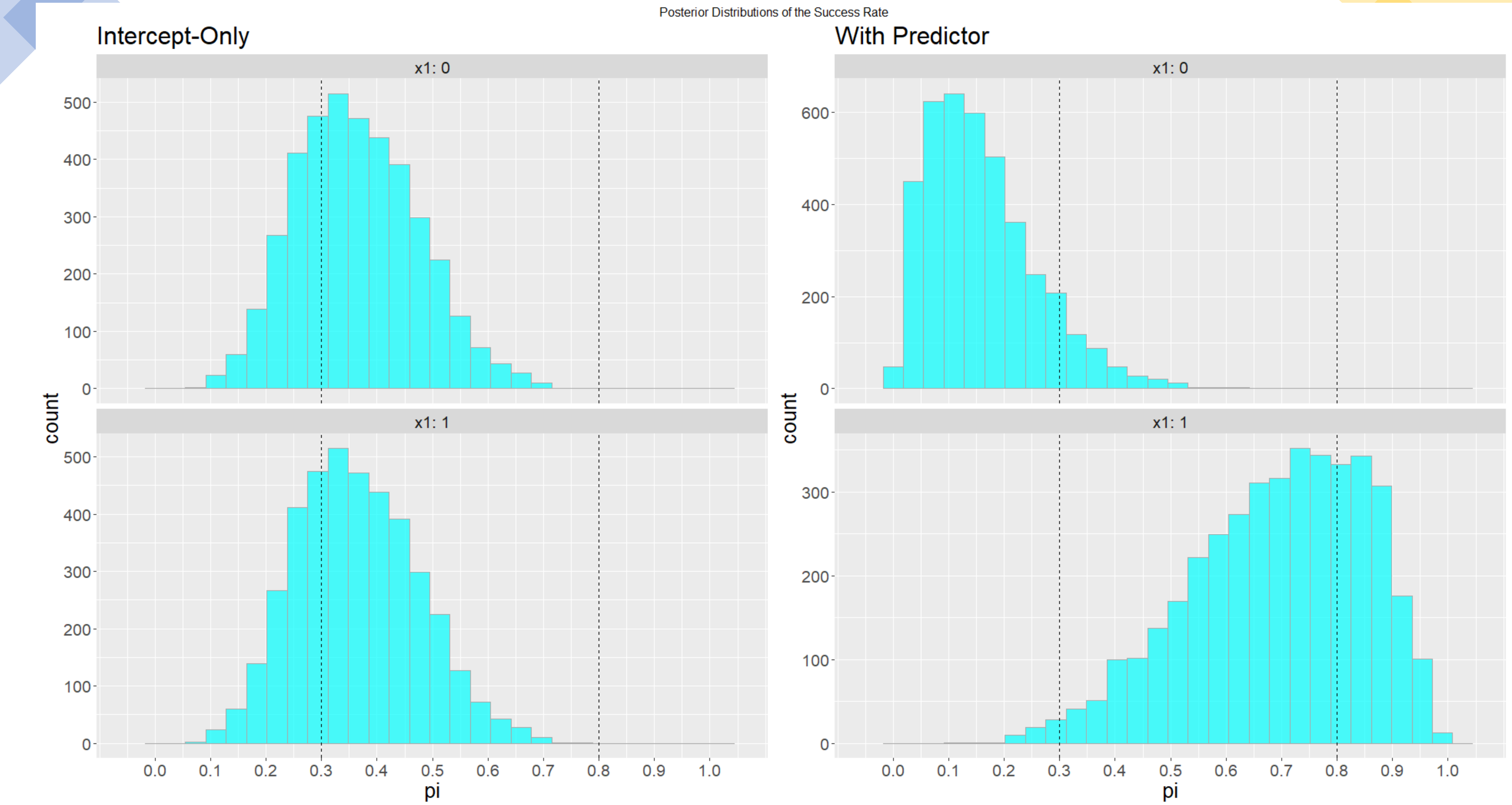
  - *This tiny example can’t show the value of loo & pointwise metrics. So, illustrate this below with data including a covariate x1 and building two models, one without the covariate as predictor (i.e., intercept only) and the other with it.*

Model Comparison

```
> # Data w/covariate & two binomial-logit models.
> df2 <- tibble(n=c(5,1),N=c(6,10),x1=c(1,0))
> sim2_int <- brm(
    formula = n | trials(N) ~ 1,
    family = binomial(link="logit"),
    data = df2,
    prior = set_prior( prior = "normal(-1,1.5)",
           class="Intercept"),
    sample_prior = "only"
)
> sim2_wx1 <- brm(
    formula = n | trials(N) ~ 1 + x1,
    family=binomial(link="logit"),
    data = df2,
    prior = c(
        set_prior( prior = "normal(-1,1.5)",
           class="Intercept"),
        set_prior( prior = "normal(0,2)", class = "b")
    ),
    sample_prior = "only"
)
> mod2_int <- update(
    sim2_int,
    sample_prior = “no”,
    save_pars = save_pars(all=TRUE)
)
> mod2_wx1 <- update(
    sim2_wx1,
    sample_prior = “no”,
    save_pars = save_pars(all=TRUE)
)
```

```
> # Add the LOO criterion to both of them,
  #  then compare the models
> mod2_int <- add_criterion(
    mod2_int,
    criterion = "loo",
    moment_match = TRUE
)
> mod2_wx1  <- add_criterion(
    mod2_wx1,
    criterion = "loo",
    moment_match = TRUE
)
> loo_compare(
    mod2_int,
    mod2_wx1,
    model_names = c("intercept_only","with_predictor")
)
               elpd_diff se_diff
with_predictor  0.0       0.0
intercept_only -5.8       0.2
```


Posterior Predictive Distributions of the Success Counts, n
Intercept-Only
With Predictor
x1: 0
x1: 1
x1: 0
x1: 1
proportion
proportion
0.20
0.15
0.10
0.05
0.00
0 1 2 3 4 5 6 7 8 9 10
0.3
0.2
0.1
0.0
0 1 2 3 4 5 6
0.3
0.2
0.1
0.0
0 1 2 3 4 5 6 7 8 9
0.2
0.1
0.0
0 1 2 3 4 5 6
Data
Data
Data
successes, n
successes, n


Posterior Distributions of the Success Rate
Intercept-Only
With Predictor
x1: 0
x1: 1
x1: 0
x1: 1
count
count
0
100
200
300
400
500
0
200
400
600
0
100
200
300
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
pi
pi

Question for You:

Given the complexity of Bayesian Analysis, when is it a good value, i.e. when does it provide results so valuable that they outweigh the additional complexity?

Use of Bayesian Analysis is Compelled by…

… These Issues

- **Uncertainty Quantification (UQ)**
  - Sparsity & Missing Data
  - Risk & Decision Analysis; Post-Inference App: Explanations; Confirmations
- **Domain Knowledge (DK) Inclusion**
  - Expertise; Causality/Structural/State Space/Mechanism Representation
- **Heterogeneity**
  - Multi-level Behavior (context-specific; individual-specific)
- **Latent Structure**
  - Factor Analysis; Conceptual Embeddings; Underlying Cognitive Constructs
- **Multiple Responses**
  - Information Fusion;
    Multimodal Data Streams
- **Adaptation**
  - Autonomous & Active Learning Systems; Adaptive Questionnaires; Adaptive Experiments

Example: Predicting Success Rates

Base Model: $n \sim \mathrm{Binomial}(N,\pi)$; prior $\mathrm{p}(\pi)$

- **UQ:** posterior captures uncertainty
  $\mathrm{p}(\pi \mid \{N,n\}_{\mathrm{Data}})$; prior $\mathrm{p}(x)$, e.g. “error-in-variables”
- **DK Inclusion**: leverage expertise; interventions on x;
  $u(x) = \mathrm{f}(x;\theta)$; captures mechanism (possibly nonlinear)
  $\pi = g^{-1}(u(x))$ (logit link function, $g(.)$)
- **Heterogeneity:** hyper-personalization
  context- and individual-specific parameters
  $u(x_{ij}) = \mathrm{f}(x_{ij};\theta_{ij})$; $\theta_{ij} \sim \mathrm{Normal}(\theta_j)$; $\theta_j \sim \mathrm{Normal}(\bar{\theta})$
- **Latent Structure:** cognitive constructs manifest as observed behavior; e.g. customer preferences
  $u(x,z) = \mathrm{f}(x,z;\theta)$; $\mathrm{z} = h(x)$ captures theory of behavior
- **Multiple Responses:** universal latent constructs z driving behaviors measured by different modalities – e.g. text reviews, videos, photos, survey responses. Augment trial-success data $y=\{N,n\}_{\mathrm{Data}}$ with other data & add measurement models (likelihood modules)
- **Adaptation:** leverage UQ to compute learning objectives to generate new x; e.g. optimal product design; adaptive recommenders

More on Bayesian Analysis….

- Rasmus Baath "Bayesian Analysis (What, Why, How)" Tutorial
  - YouTube Videos (2017) (each ~30 min.)

- Michael Thompson's Flipboard e-zine mashup: "Bayesian"

- Prof. Andrew Gelman's blog: "Statistical Modeling, Causal Inference, and Social Science"
  - **Note:** *Your obligations as a professional include being aware of the big issues faced by your profession. Read key Influencers.*

- BANA 8090 – Special Topics: Bayesian Analysis, instructor: M.L. Thompson
  - Spring 2021 (1st half)

Readings

- Statistics
    - Gelman et al. “Bayesian Workflow”
    - Gabry et al. “Visualization in Bayesian Workflow”
    - Gelman et al. “The Prior Can Often Only Be Understood in the Context of the Likelihood”
    - Richard McElreath “Statistical Rethinking”

- AI/ML
    - Chris Bishop (Microsoft Research) “Model-Based Machine Learning”
    - Daphne Koller & Nir Friedman “Probabilistic Graphical Models” (excerpt)
    - Norman Fenton & Martin Neil “Risk Assessment & Decision Analysis with Bayesian Networks” (sample chapters)

About the Presenter

www.linkedin.com/in/mlthomps

Dr. Michael L. Thompson is retired from the Procter & Gamble Company, where he led Bayesian Analysis R&D in consumer & market modeling. His degrees are in Chemical Engineering: B.S., Northwestern University, ’82; M.S., MIT, ’84; and Ph.D., MIT, ’96, with minor in Statistics and Artificial Intelligence. Michael has extensive experience in the process industry, having worked for Dow, Alcoa, Amoco, and Mitsubishi Chemical (Japan). At P&G for 21 years, Michael applied his expertise in Bayesian Analysis, especially Bayesian belief networks (BBN), to deliver results in the consumer-packaged goods (CPG) industry. His contributions spanned business functions including R&D, Engineering, Manufacturing, Marketing, and Business Analytics. He has authored journal articles ranging from fluidized bed reactors to hybrid probabilistic and first-principles biochemical models to optimal consumer product design. Currently, Michael is a Term Adjunct in the Lindner College of Business at the University of Cincinnati, where he teaches Bayesian Analysis to candidates for the Master of Science in Business Analytics. He also serves on the Advisory Board for the Retail AI Lab of the Northwestern University Retail Analytics Council.