Bayesian Workflow

Andrew Johnson

Aalto University

Nordic Probabilistic AI School 2023

Outline

- Bayesian Inference
- Bayesian Workflow
- MCMC Sampling
- HMC/NUTS & Stan
- Step-by-Step Workflow: Epilepsy RCT
- Conclusions

Why Bayesian Inference?

We have knowledge and assumptions beyond just the current observations

We have some uncertainty in our predictions/beliefs

Bayesian Inference

- Let's make a bet on what a new friend does for a living

- Meet Avery

- Is Avery a:
  a) Farmer
  b) Librarian

Bayesian Inference

- What observations and what assumptions do we have at this stage?

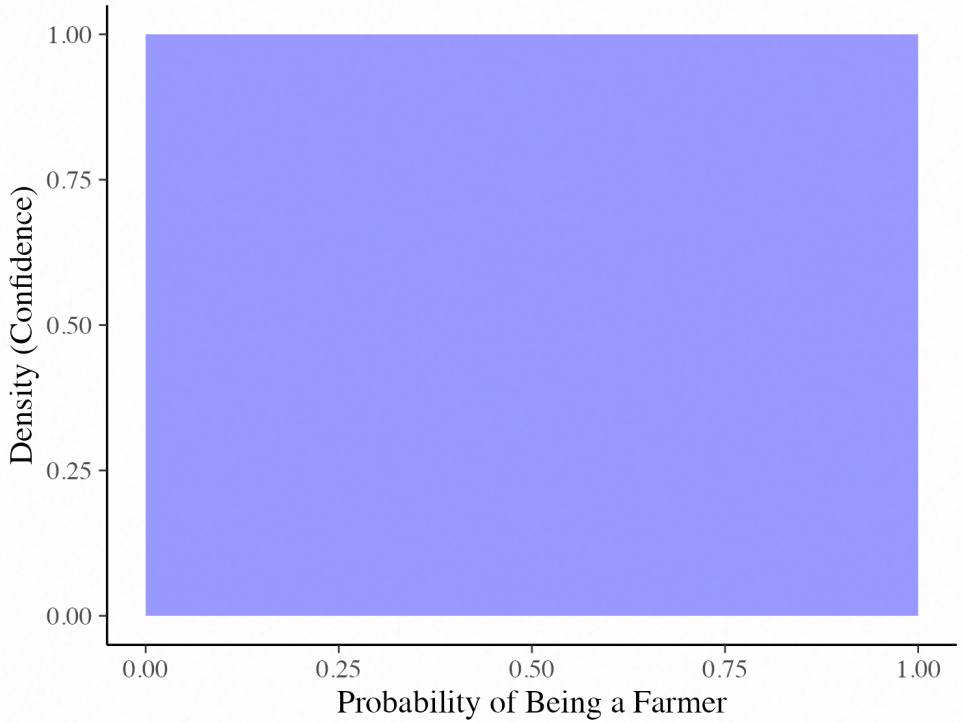
- Observations:
  - None

- Assumptions:
  - None

- Prediction:
  - Avery is equally likely to be a farmer or a librarian, and we have no certainty in our prediction

Bayesian Inference

- Every probability of Avery being a farmer is possible
- We consider every probability to be equally likely



Bayesian Inference

- What if Avery told you a bit more about themselves?

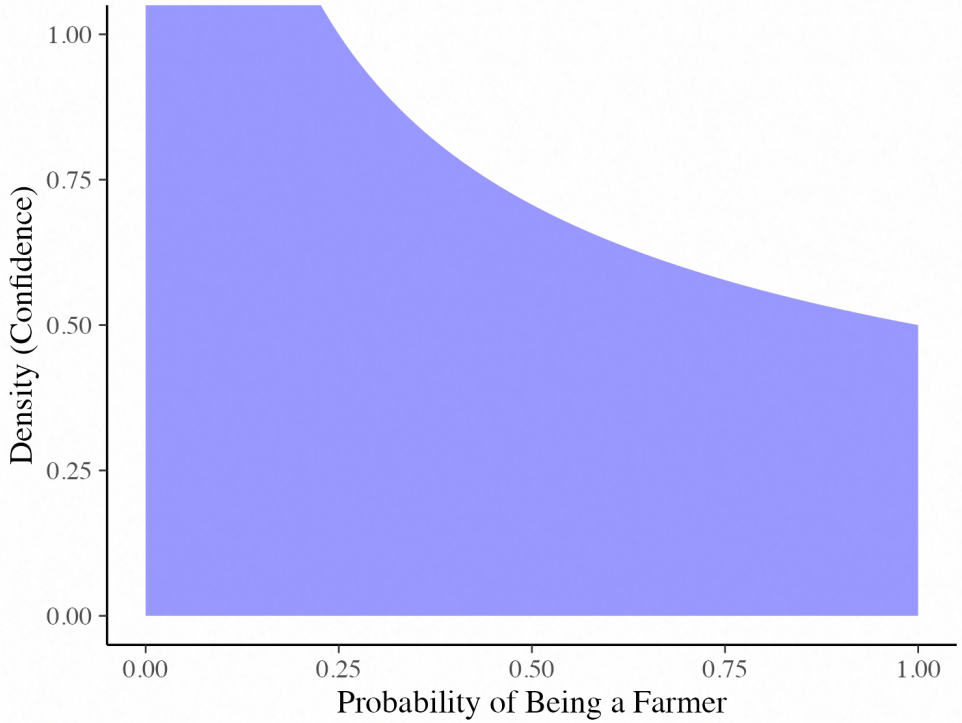
- Observations:
  - Avery prefers structure and organisation
  - They are generally described as ‘quiet’ and ‘introverted’

- Assumptions:
  - None

- Prediction:
  - We predict that Avery is more likely to be a librarian than a farmer
  - How confident should we be in this prediction? Why?

Bayesian Inference

- We believe that the probability of being a farmer is small, but we might still be wrong
- We have more confidence in lower probabilities than higher probabilities
- What if we had some idea about the probability of any given person being a farmer?



Bayesian Inference

- What if you knew some worldwide occupation statistics?

- Observations:
  - Avery prefers structure and organisation
  - They are generally described as ‘quiet’ and ‘introverted’

Bayesian Inference

- What if you knew some worldwide occupation statistics?

- Observations:
  - Avery prefers structure and organisation
  - They are generally described as ‘quiet’ and ‘introverted’

- Assumptions:
  a) Farmer: 874 Million worldwide (2020)[1]
  b) Librarian 1.6 Million worldwide (2020)[2]

1. https://librarymap.ifla.org/map/Metric/Full-Time-Staff/LibraryType/National-Libraries,Academic-Libraries,Public-Libraries,Community-Libraries,School-Libraries,Other-Libraries/Weight/Totals-by-Country
2. https://reliefweb.int/report/world/fao-statistical-yearbook-2021-world-food-and-agriculture

Bayesian Inference

- What if you knew some worldwide occupation statistics?

- Observations:
  - Avery prefers structure and organisation
  - They are generally described as ‘quiet’ and ‘introverted’

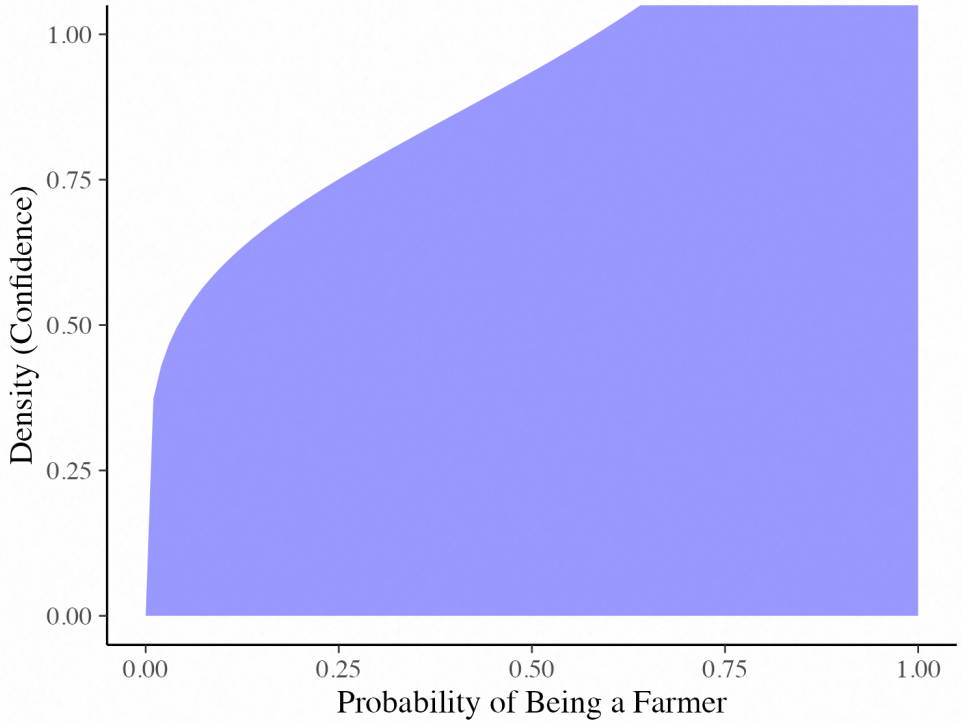
- Assumptions:
  a) Farmer: 874 Million worldwide (2020)[1]
  b) Librarian 1.6 Million worldwide (2020)[2]

- Prediction:
  - We predict that Avery is much more likely to be a farmer than a librarian
  - How much additional certainty/influence should worldwide statistics add?

1. https://librarymap.ifla.org/map/Metric/Full-Time-Staff/LibraryType/National-Libraries,Academic-Libraries,Public-Libraries,Community-Libraries,School-Libraries,Other-Libraries/Weight/Totals-by-Country
2. https://reliefweb.int/report/world/fao-statistical-yearbook-2021-world-food-and-agriculture

Bayesian Inference

- Considering the global probability of being a farmer, we predict Avery more likely to be a farmer
- Without knowing where Avery lives, we don't have strong confidence in the information provided by the global probability
- What if we knew more about Avery? How does our confidence in our prior change the influence of that information?



Bayesian Inference

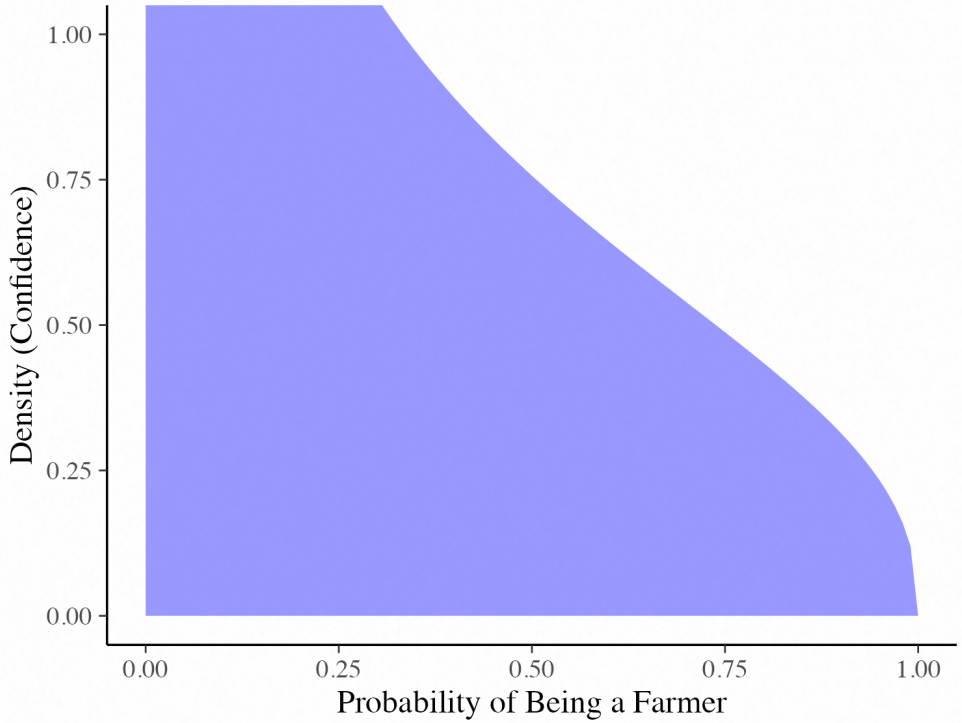
- Let's learn some more about Avery:

- Observations:
  - Avery prefers structure and organisation
  - They are generally described as 'quiet' and 'introverted'
  - **Avery loves books and spends most afternoons at the library**

- Assumptions:
  a) Farmer: 874 Million worldwide (2020)[1]
  b) Librarian 1.6 Million worldwide (2020)[2]

Bayesian Inference

- Because we don't have strong confidence in our prior, we give more weight to our observations
- We predict that Avery is more likely to be a librarian, and we have a strong level of confidence in that prediction
- But what would happen if we had very strong prior information?



Bayesian Inference

- What if you knew that Avery was from:

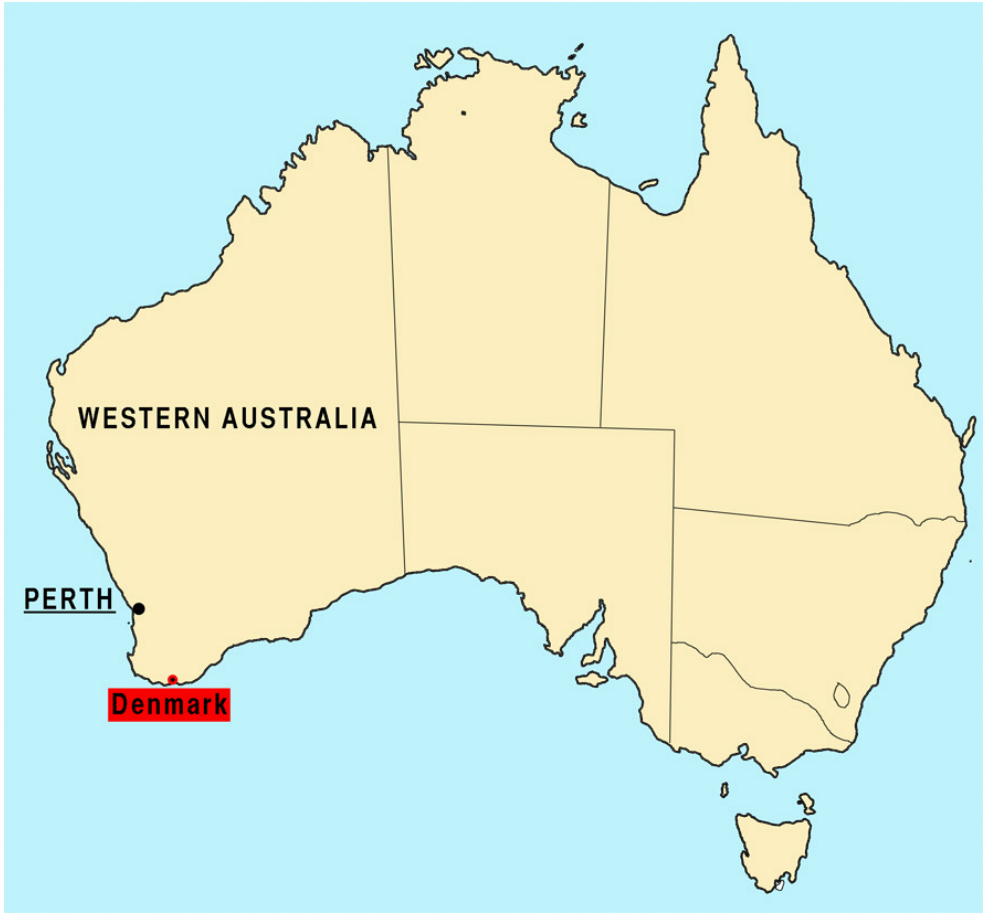
- Denmark

Bayesian Inference

- What if you knew that Avery was from:

- Denmark, Australia
  - Population 6,422
  - Primary industry is agriculture
  - Only 2 full-time librarians

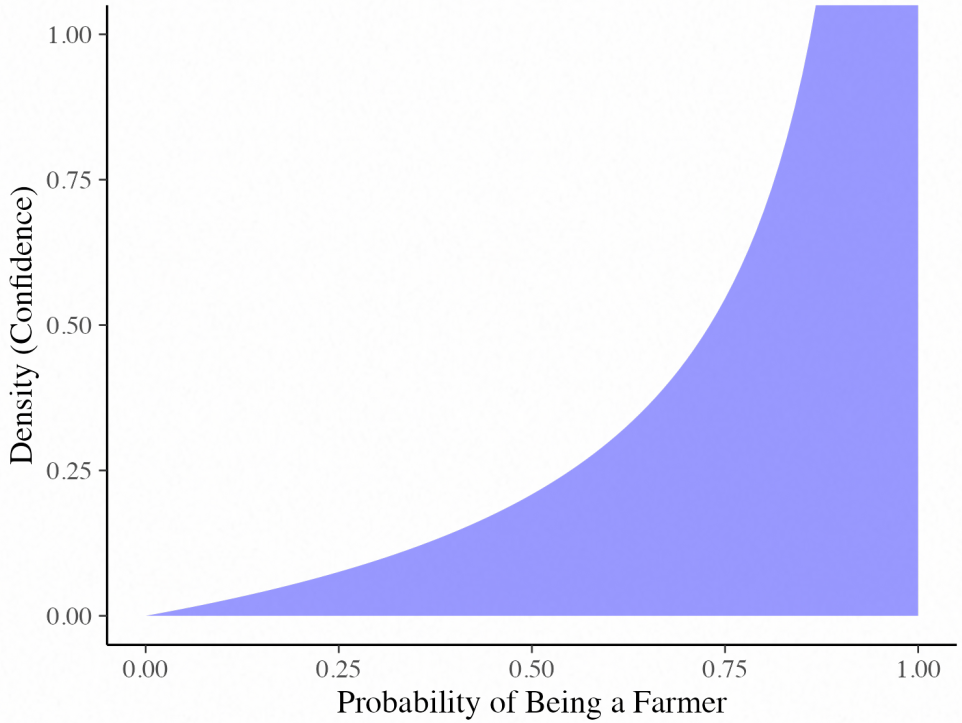
- What does this mean for our predictions about Avery?



Bayesian Inference

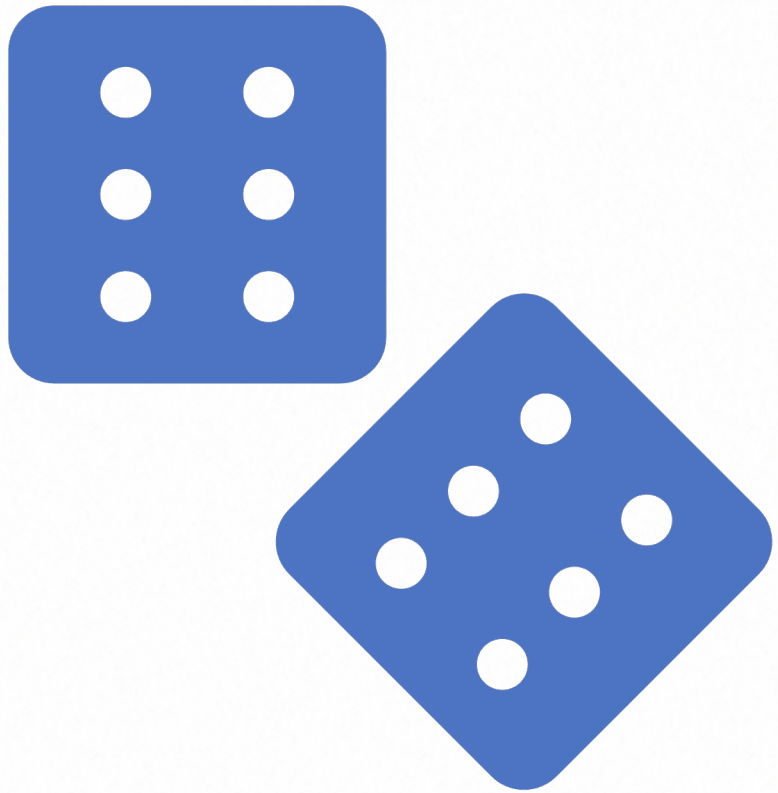
- Our prior information tells us that Avery is very highly likely to be a farmer, regardless of their personality

- The strength of this prior information means that we need more and more observations about Avery before we would change our prediction



Bayesian Inference – Key Takeaways

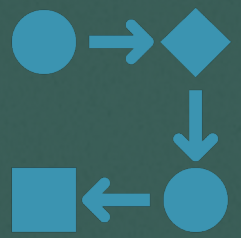
- We consider both our observations and external/prior information when making predictions
- Our strength of confidence/certainty in these also influence our confidence/certainty in our predictions
- If one of them is “weak” then the other will drive our predictions and certainty



Components of Bayesian Inference

$$p(\theta|y) \propto p(y|\theta)p(\theta)$$

- $p(\theta)$: Prior Knowledge
  - What is the probability of a person being a farmer?
- $p(y|\theta)$: Likelihood
  - How does Avery's personality contribute to their probability of being a farmer?
- $p(\theta|y)$: Posterior
  - What is the probability of Avery being a farmer?



How do we take some observations and prior information and turn it into a model?

How can we be confident we've made good modelling choices?

Bayesian Workflow, of course

Bayesian Workflow

- What is it and why would I need it?

- Bayesian inference is powerful, but model development can be a complex process
  - Multiple options for prior & likelihood choice
  - Multiple ways to specify the same prior & likelihood

- Can lead to series of ad-hoc decisions and discarded models

- We need a systematic, but flexible, process to follow

Bayesian Workflow

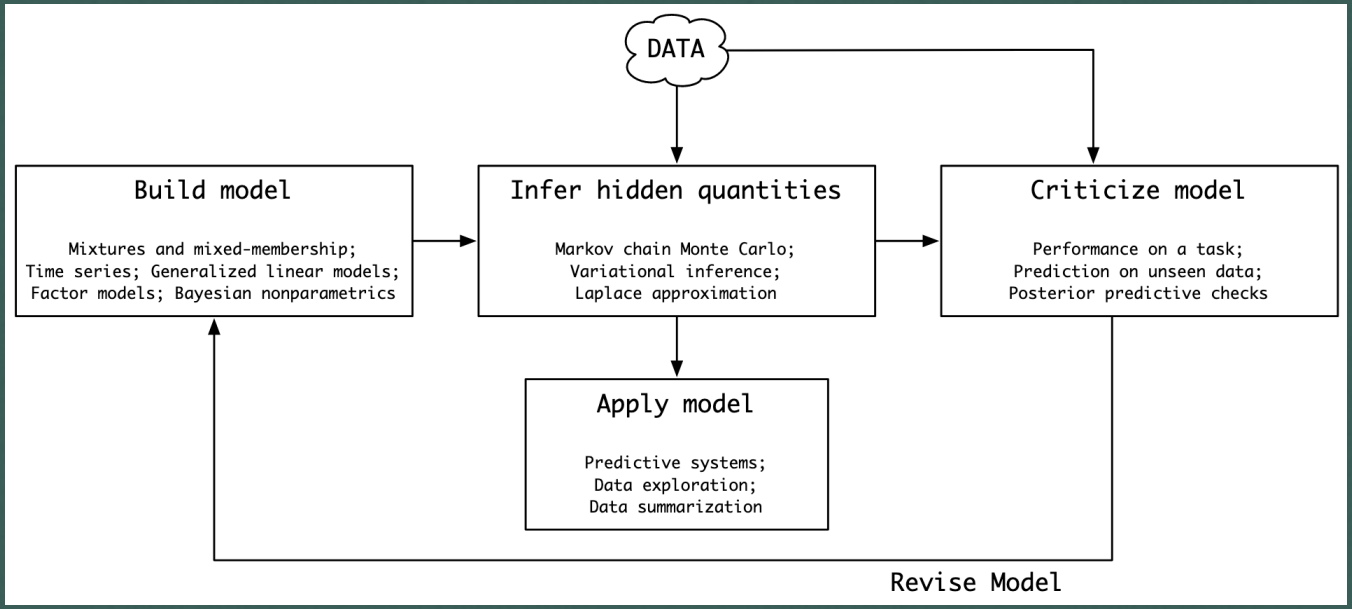
- Three broad stages of Bayesian modelling:
  - Model Building
  - Inference
  - Model Checking/Improvement
- Each stage naturally covers multiple issues/tasks
- The results from one stage can require returning to earlier stages to make changes
- Iterative process

Principles of Bayesian Workflow

- Systematic
  - Replace ad-hoc decision-making with recommended, evidence-based, guidance
- Flexible
  - Every analysis is different
  - Rigid set of steps to follow will result in poorer results for some
- Practical
  - Time and resources are not infinite
  - Want a robust process while reducing unnecessary effort and time
    - Fail early, fail fast

Early Workflow

- Early modelling workflow proposed by Box, popularised by David Blei (2014) is Box's Loop:

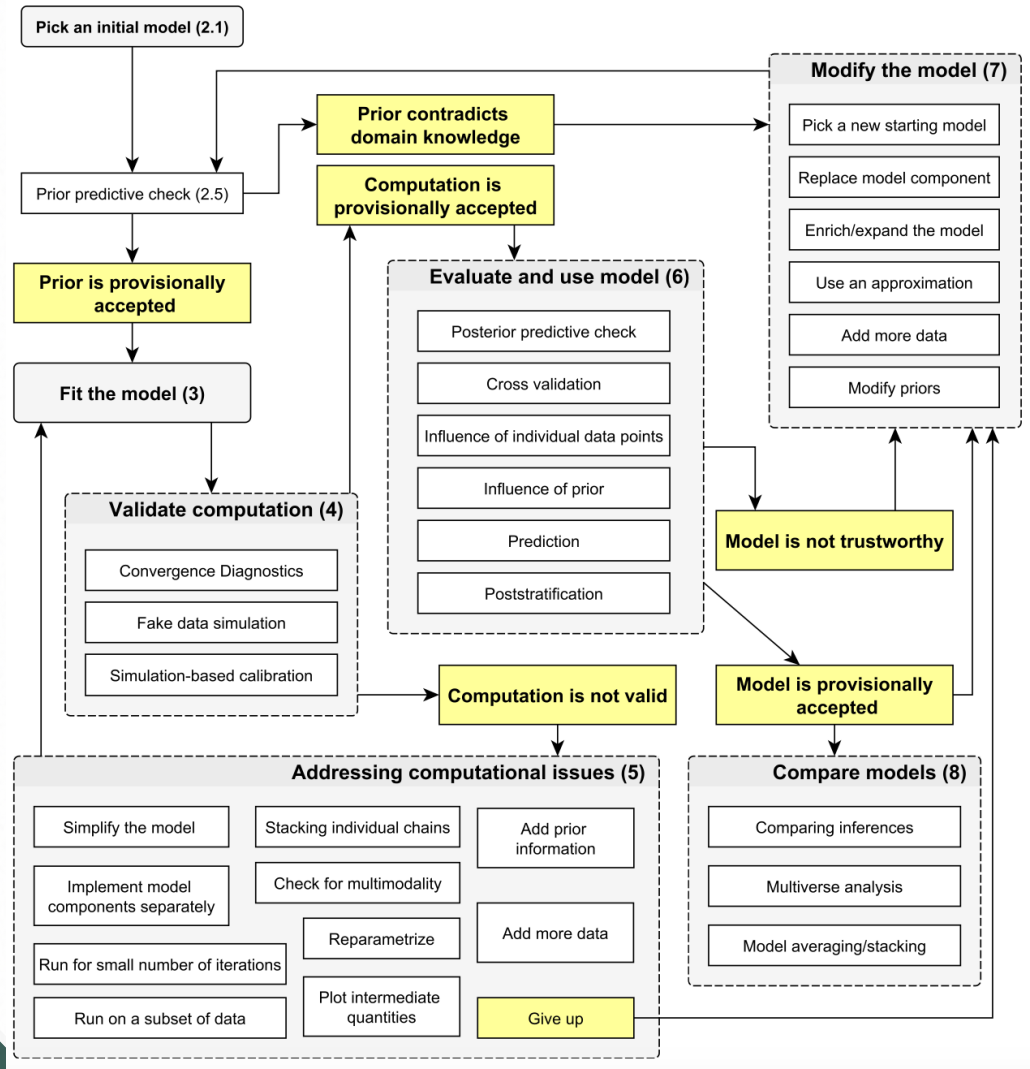


Gelman et al. (2020)

- Proposed series of steps to follow
  - Application & interpretation of diagnostics
  - Early identification of model-misspecification
  - Assessing computational & model trustworthiness before inference

Gelman et al. (2020)

- Proposed series of steps to follow
  - Application & interpretation of diagnostics
  - Early identification of model-misspecification
  - Assessing computational & model trustworthiness before inference



Computation in Bayesian Inference

- We want to make inferences using our posterior distribution:

$$p(\theta|y) \propto p(y|\theta)p(\theta)$$

- The posterior distribution is not able to be specified in closed-form for majority of models

- We need a numerical approximation: Markov Chain Monte Carlo

MCMC Sampling

- Markov Chain Monte Carlo is a powerful tool for numerically computing expectations over complex/difficult distributions

- Key component is the Markov chain
  - Sequential set of samples from a given target distribution
  - Each sample informs where the next sample should be drawn from (Markov transition)
  - Given enough time (and sampling), these will quantify the target distribution

- Markov Chain Monte Carlo estimate is then the average of the target function over these samples

MCMC Sampling

- Given target distribution (posterior): $p(\theta|y)$

MCMC Sampling

- Given target distribution (posterior):
$$p(\theta|y)$$

- We estimate a Markov Chain for N iterations (N sequential samples):
$$\{s_1, \dots, s_N\}$$

MCMC Sampling

- Given target distribution (posterior):
$$p(\theta|y)$$

- We estimate a Markov Chain for N iterations (N sequential samples):
$$\{s_1, \dots, s_N\}$$

- Apply our target function to the Markov Chain samples:
$$\hat{f}_N = \frac{1}{N}\sum_{n=0}^{N} f(s_n)$$

MCMC Sampling

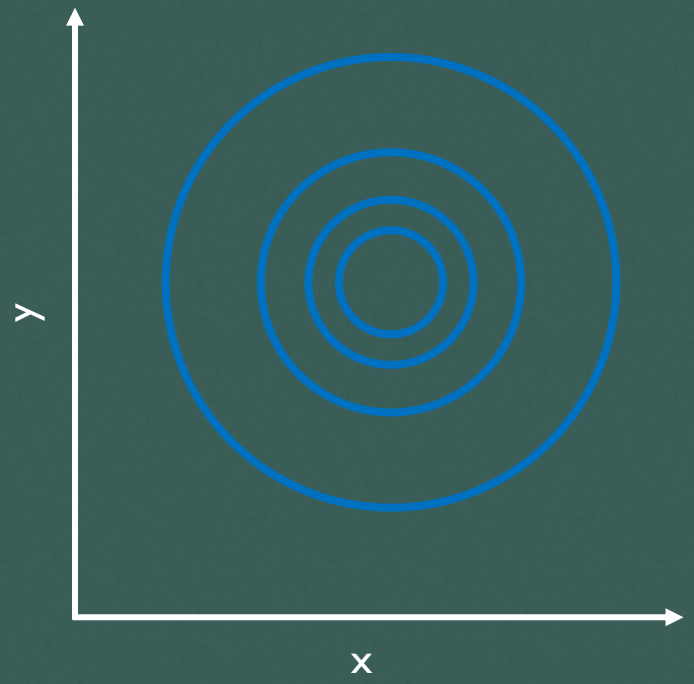
- Given target distribution (posterior):
$$p(\theta|y)$$

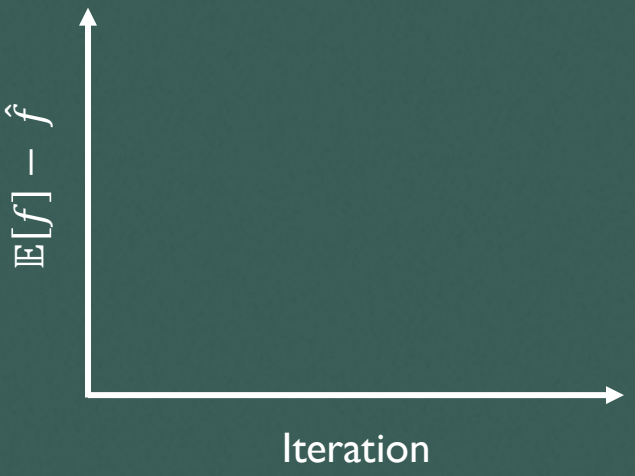
- We estimate a Markov Chain for N iterations (N sequential samples):
$$\{s_1, \dots, s_N\}$$

- Apply our target function to the Markov Chain samples:
$$\hat{f}_N = \frac{1}{N}\sum_{n=0}^{N} f(s_n)$$

- This estimate converges to the true expectation:
$$\lim_{N\to\infty} \hat{f}_N = \mathbb{E}_{p(\theta|y)}[f]$$

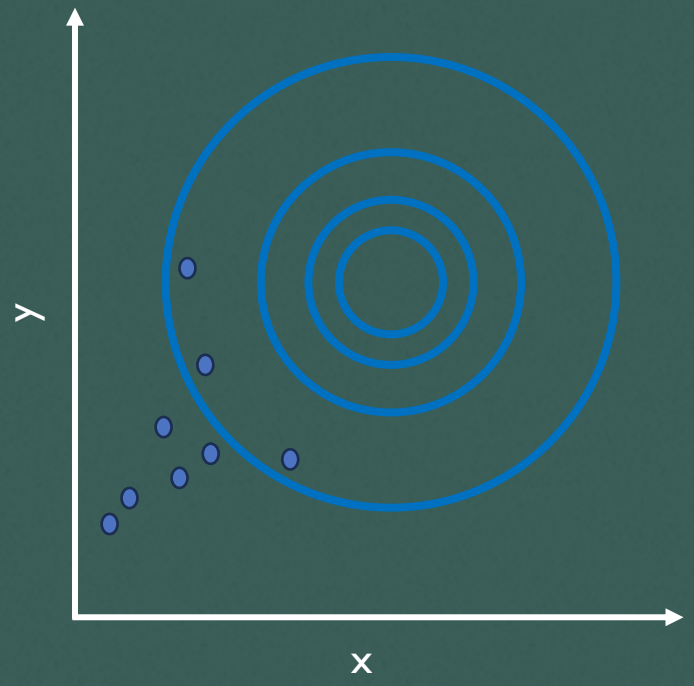
MCMC Sampling

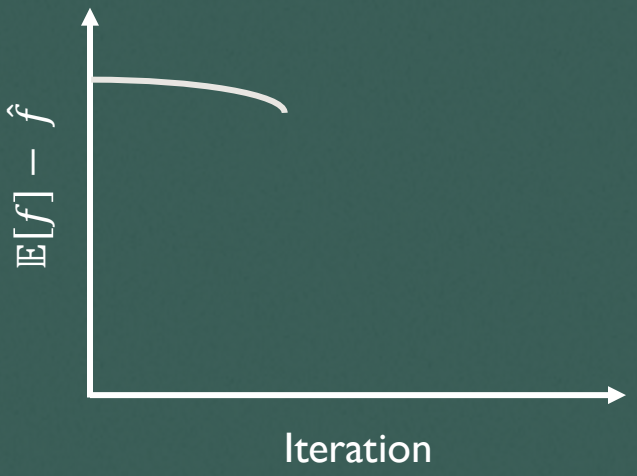




MCMC Sampling

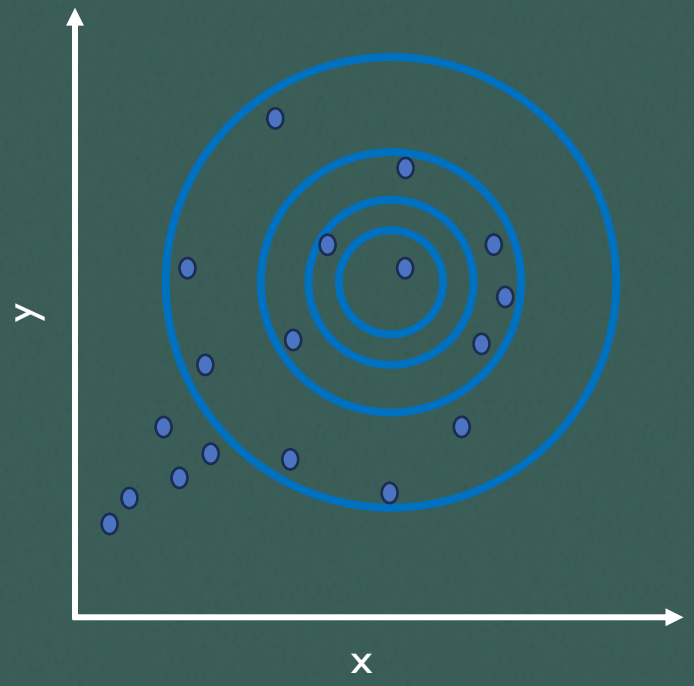
- Most bias present earliest

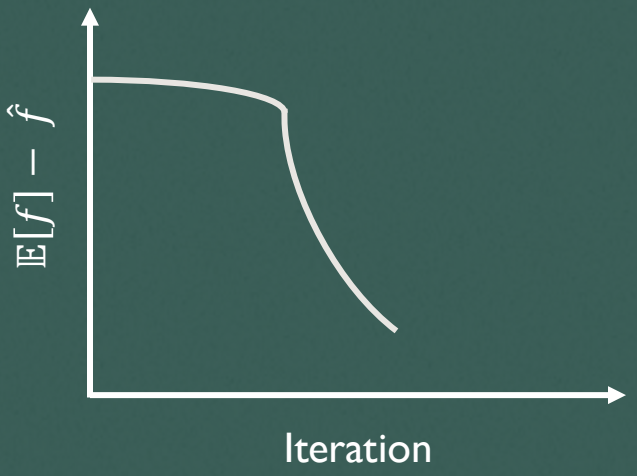




MCMC Sampling

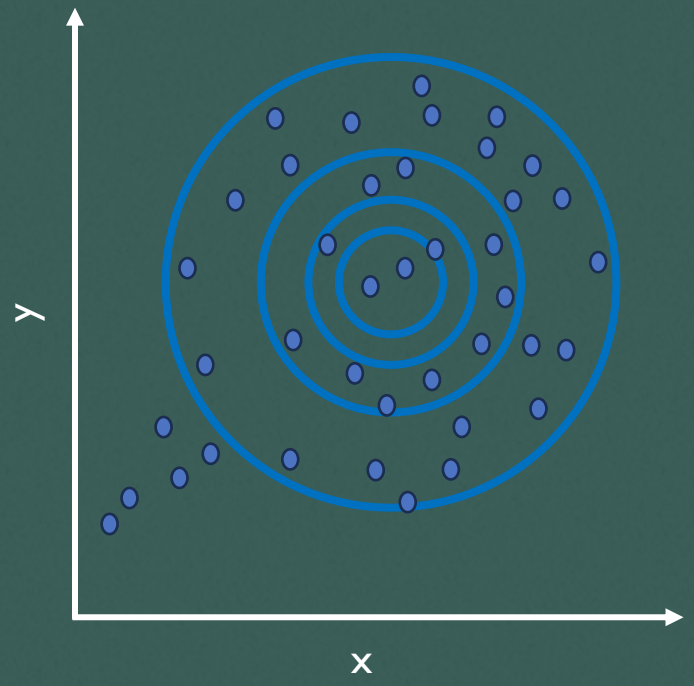
- Reduces once probability mass is found and initially explored

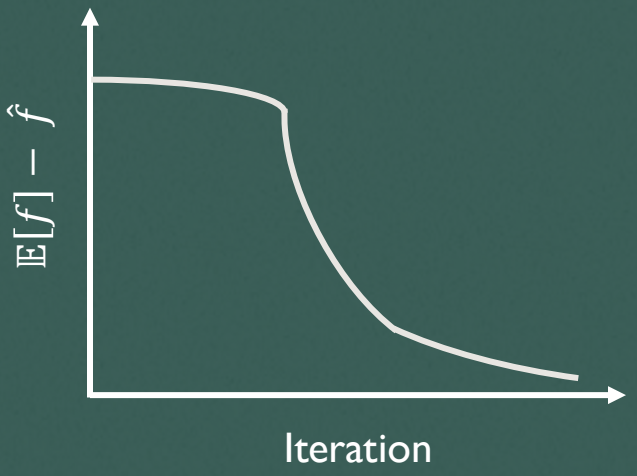




MCMC Sampling

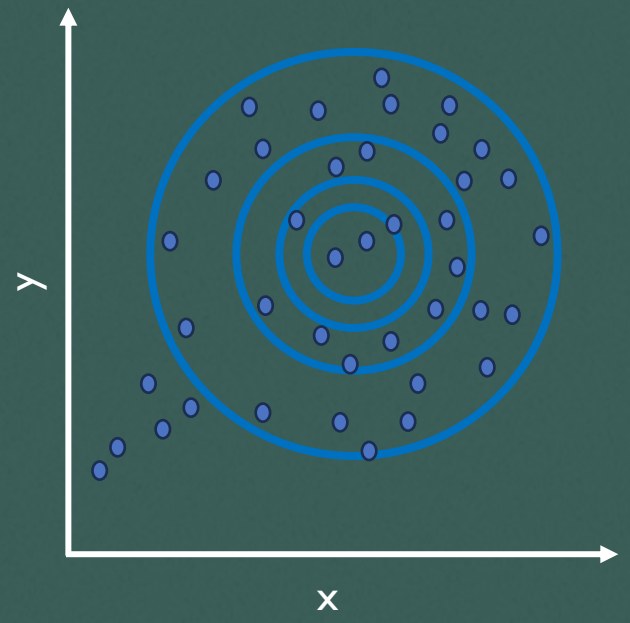
- Asymptotically tends to zero bias as exploration continues

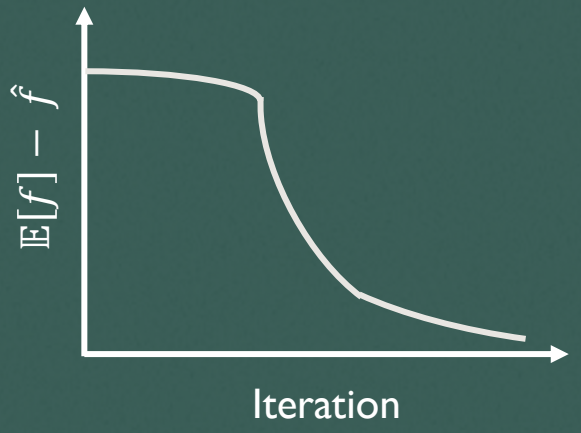




MCMC Sampling

- Asymptotically tends to zero bias as exploration continues





Note two sources of bias:

- Finding probability mass
- Fully exploring probability mass

HMC: Scaling Bayesian Inference

Traditional MCMC methods like MH & Gibbs (often, not always) struggle with increasing model complexity

Parameter space to be explored becomes increasingly sparse as dimensionality increases

Hamiltonian Monte Carlo uses the gradient of the target probability density function to "guide" sampling

- Requires the specification of "tuning" parameters
- Unique to the posterior at hand (in most cases)

A self-tuning version of HMC – the No U-Turn Sampler (NUTS) since been popularised by Stan

Stan

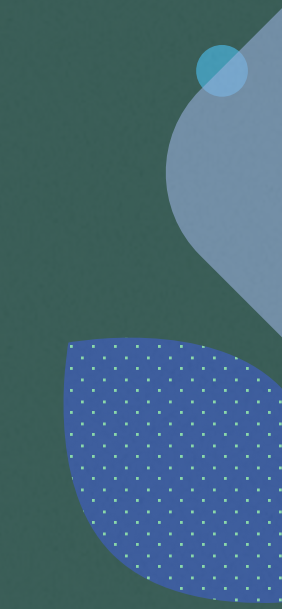
- PPL providing estimation via:
  - MCMC Sampling
    - NUTS
    - HMC
  - Optimisation
    - Newton
    - BFGS
    - L-BFGS
  - Automatic Differentiation Variational Inference
  - Laplace Approximation

Structure of a Stan Program

```
data {}

parameters {}

model {}
```



Structure of a Stan Program

```
data {}

parameters {}

model {}
```

- Linear Model, unknown Mean & SD:

$$y_i \sim N(\mu, \sigma); \;\; i = 1, \ldots, N$$

$$\mu \sim N(0, 5)$$

$$\sigma \sim N^{+}(0, 5)$$

Structure of a Stan Program

```
data {
  int N;
  vector[N] y;
}
parameters {}
model {}
```

- Linear Model, unknown Mean & SD:

$$y_i \sim N(\mu, \sigma); \;\; i = 1, \ldots, N$$

$$\mu \sim N(0, 5)$$

$$\sigma \sim N^{+}(0, 5)$$

- Data we have:
  - Number of observations (N)
  - Observations ($y_{1:N}$)

Structure of a Stan Program

```
data {
  int N;
  vector[N] y;
}
parameters {
  real mu;
  real<lower=0> sigma;
}
model {}
```

- Linear Model, unknown Mean & SD:

$$y_i \sim N(\mu, \sigma); \;\; i = 1, \ldots, N$$

$$\mu \sim N(0, 5)$$

$$\sigma \sim N^{+}(0, 5)$$

- Parameters we want to estimate:
  - Location parameter (μ)
  - Positive scale parameter (σ)

Structure of a Stan Program

```
data {
  int N;
  vector[N] y;
}
parameters {
  real mu;
  real<lower=0> sigma;
}
model {
  mu ~ normal(0, 5);
  sigma ~ normal(0, 5);
}
```

- Linear Model, unknown mean & SD:

$$y_i \sim N(\mu, \sigma); \;\; i = 1, \ldots, N$$

$$\mu \sim N(0, 5)$$

$$\sigma \sim N^{+}(0, 5)$$

- Priors:
  - μ: Normal
  - σ: Half-Normal

Structure of a Stan Program

```
data {
  int N;
  vector[N] y;
}
parameters {
  real mu;
  real<lower=0> sigma;
}
model {
  mu ~ normal(0, 5);
  sigma ~ normal(0, 5)
  y ~ normal(mu, sigma);
}
```

- Linear Model, unknown mean & SD:

$$y_i \sim N(\mu, \sigma); \;\; i = 1, \ldots, N$$

$$\mu \sim N(0, 5)$$

$$\sigma \sim N^{+}(0, 5)$$

- Priors:
  - μ: Normal
  - σ: Half-Normal
- Likelihood:
  - y: Independent Normals

Stan-centricities

- Strong static typing
  - Every variable must have a defined type that cannot change

Stan-centricities

- Strong static typing
  - Every variable must have a defined type that cannot change

- Easy Vectorisation
  - Functions & distributions take either/both scalars and containers
  - Allows for re-using expensive computation and efficient handling of large inputs

Stan-centricities

- Strong static typing
  - Every variable must have a defined type that cannot change
- Easy Vectorisation
  - Functions & distributions take either/both scalars and containers
  - Allows for re-using expensive computation and efficient handling of large inputs
- Compiled vs. Interpreted Language
  - PPL's like Pyro or JAGS/BUGS directly execute ("interpret") the commands given
  - Stan models are translated to C++ and compiled to machine code for execution

Stan-centricities

- Compiled Language Benefits
  - Reduces overhead – only the code needed for the model is executed
  - Compilers are *very* good at optimising code for efficient execution
  - Machine-specific optimisations (e.g., SIMD)

- Compiled Language Limitations
  - Additional time & resource usage for compilation
  - Requires working C++ toolchain for compilation
  - Reduces accessibility for new users

What have we done so far?

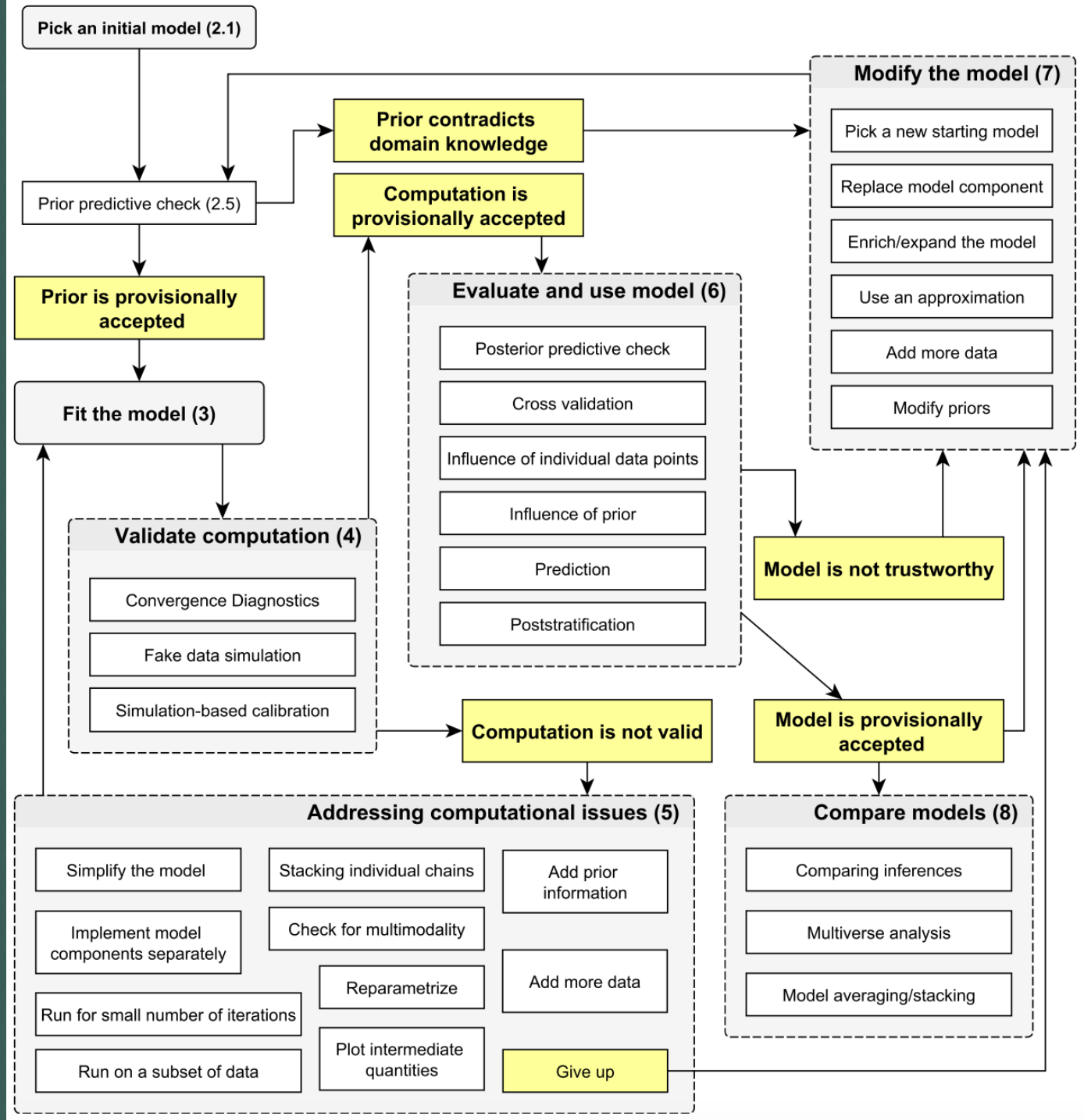
1. Established *why* we might want to apply a Bayesian model
2. Introduced the concept of a Bayesian Workflow
3. Introduced the statistical/computational methods we will need for estimating a Bayesian model
4. Introduced the PPL we will use for specifying a Bayesian model

How do we put this into practice?

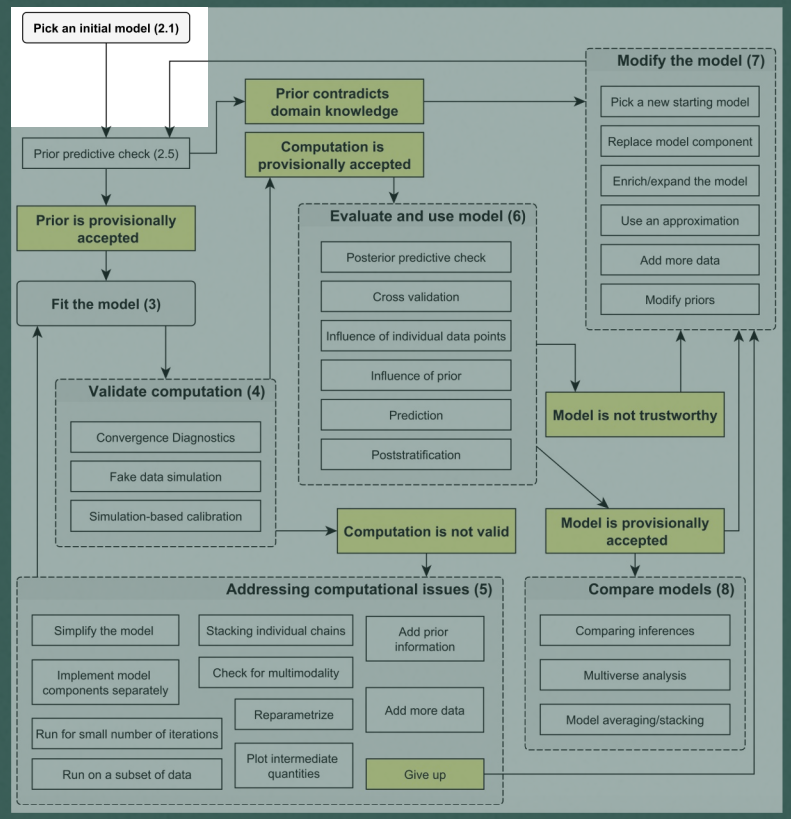
The Problem

- Randomised-controlled trial for a new epilepsy drug:
  - 59 participants
    - Control: 28
    - Treatment: 31
  - 4 assessments (plus baseline)
  - Predictors:
    - Age of participant
    - Baseline number of seizures
  - Outcome: number of seizures since last visit

The Workflow



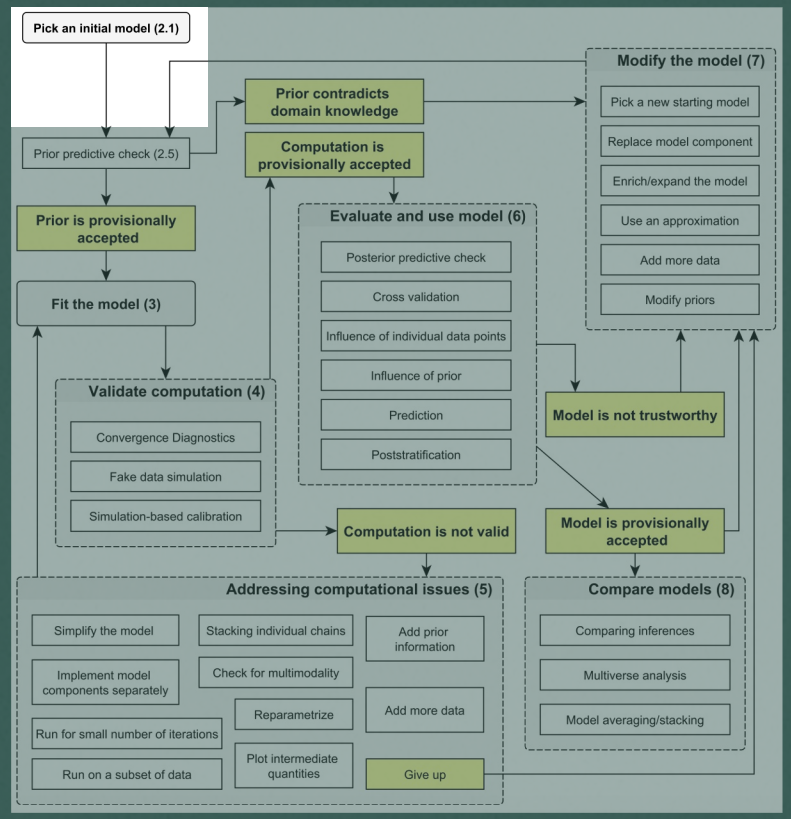
Initial Model - Hypothesis



- What do we hypothesise are the relationships between our predictors and our outcome?

- Age:
  - We don't expect age to affect the efficacy of the drug
- Baseline Seizures:
  - We expect participants with more baseline seizures will show a greater response

Initial Model - Likelihood



- Outcome is a count, so Poisson is a good start:

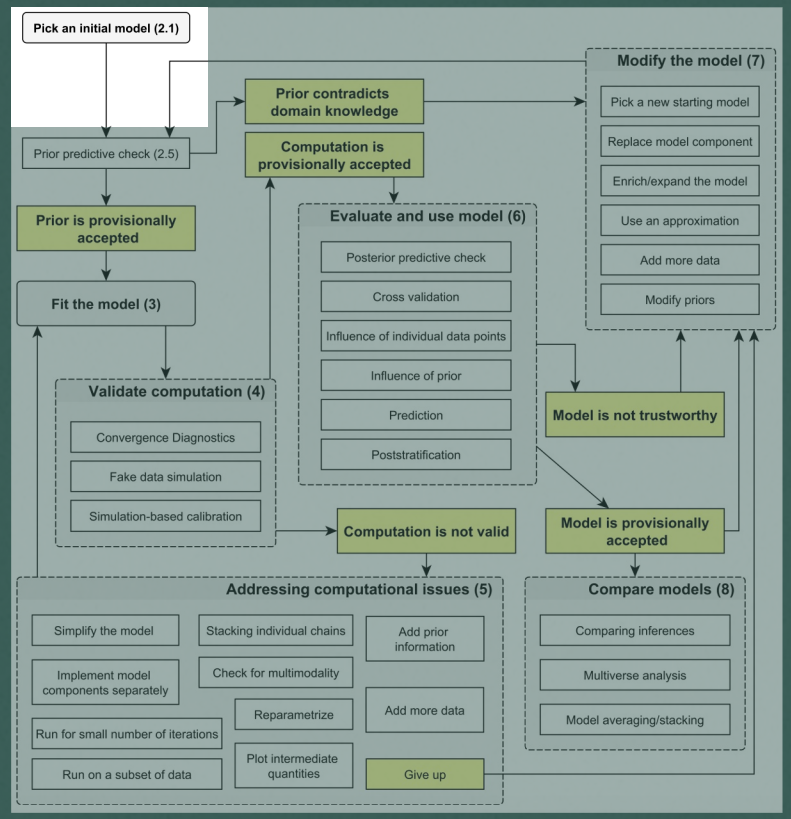
$$y_i \sim \text{Poisson}(\lambda_i)$$

- For a Poisson Generalised Linear Model (GLM) we commonly use a log link:

$$\lambda_i = \exp(\alpha + x_i^T \beta)$$

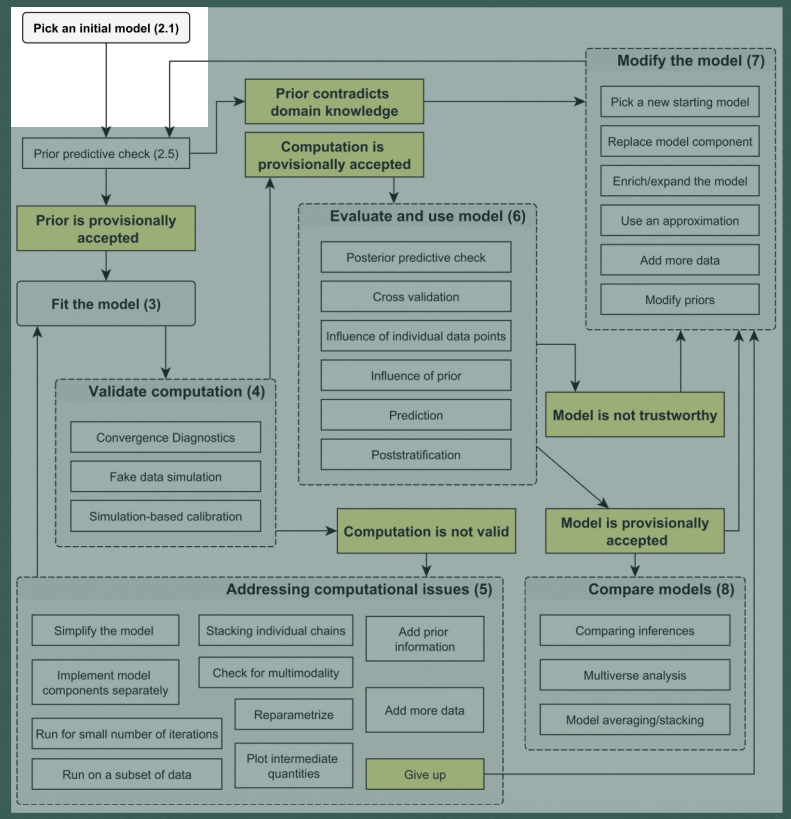
- Predictors:
  - Treatment
  - Age
  - Baseline Seizures
  - Baseline Seizures * Treatment

Initial Model - Priors



- Previous literature and pilot studies are not available to help with specifying informative priors
- We'll instead use *weakly*-informative priors
  - Regularise our estimates to feasible/possible values
- Coefficients are on the log-scale, so are *multiplicative* effects
  - $\beta = 0.1;\ \exp(0.1) = 1.105$
    - Implies a 10.5% increase
  - We'll use a $\text{Normal}(0,1)$ prior

Initial Model - Full



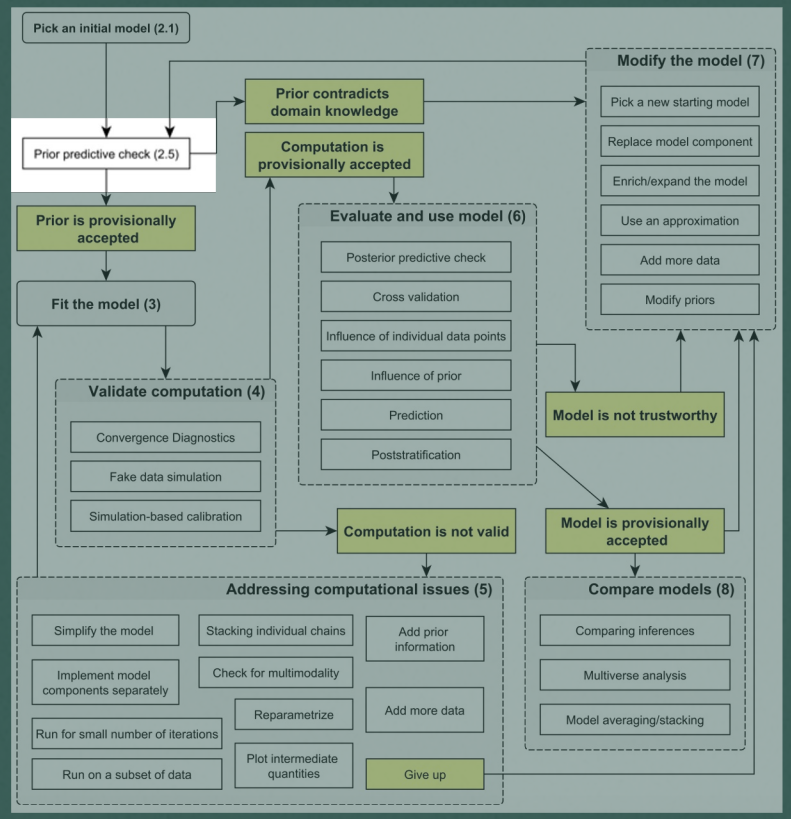
$$y_i \sim \text{Poisson}(\lambda_i)$$

$$\lambda_i = \exp(\alpha + x_i^T \beta)$$

$$\alpha \sim \text{N}(0, 5)$$

$$\beta_{1:4} \sim N(0,1)$$

Prior Predictive Check



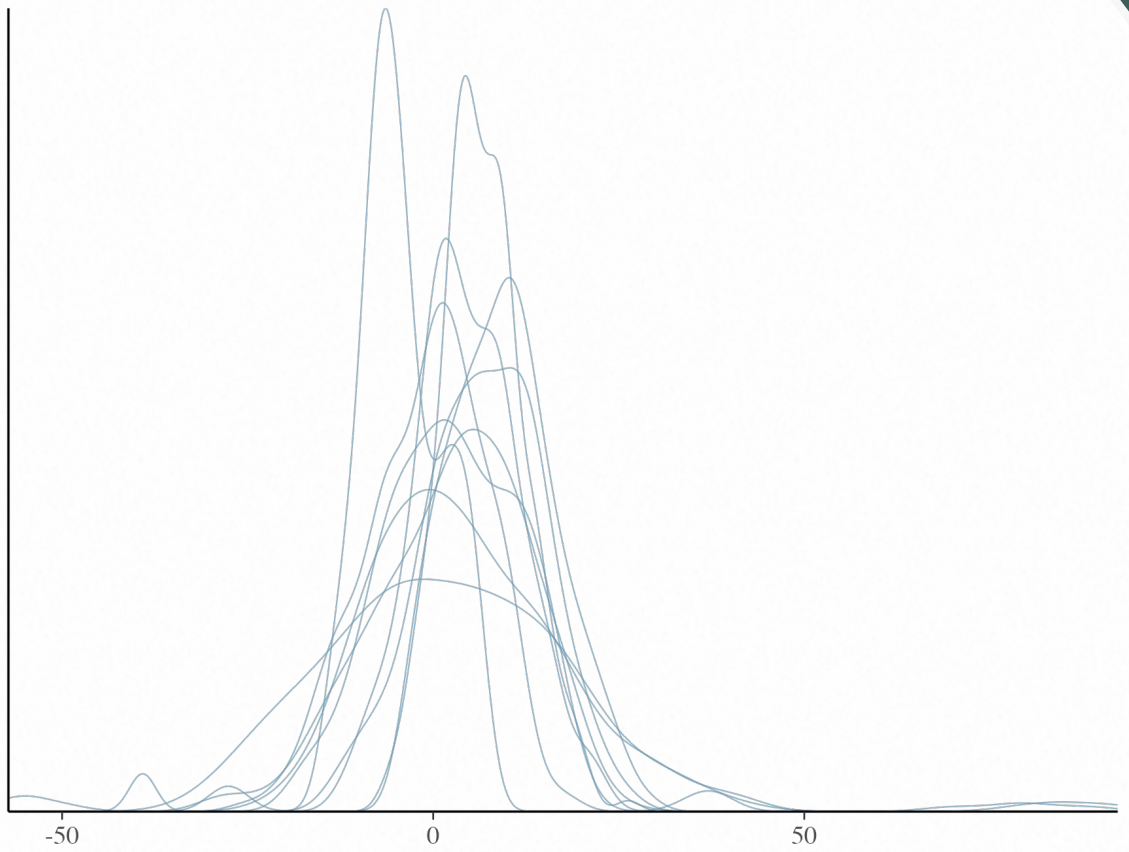
- What kind of observed data does this prior imply?
- Process:
  - Simulate data from our prior
  - Compare to our observations

- Allows us to identify whether model-misspecification is likely, before fitting the final model
  - *Fail early!*

Prior Predictive Check

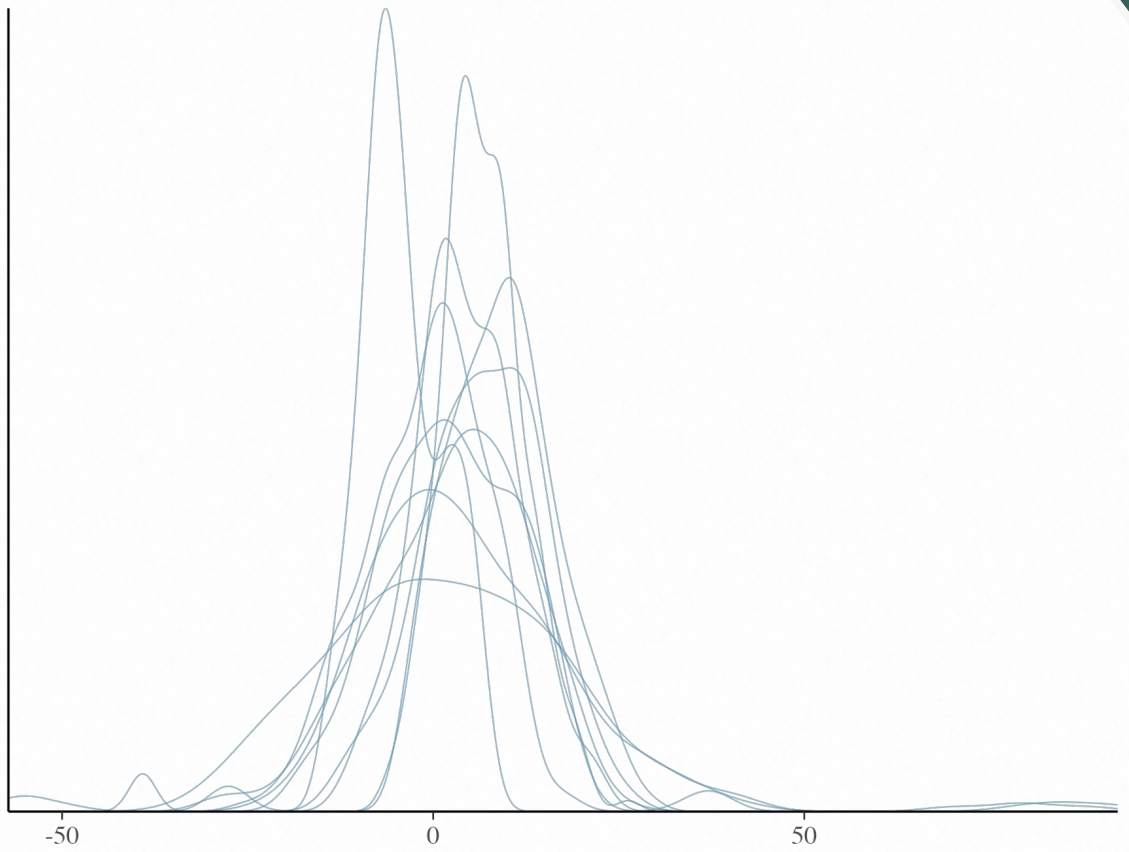
- What would happen if we chose a Gaussian/Normal likelihood instead of a Poisson?

- Why would we reject our model here?

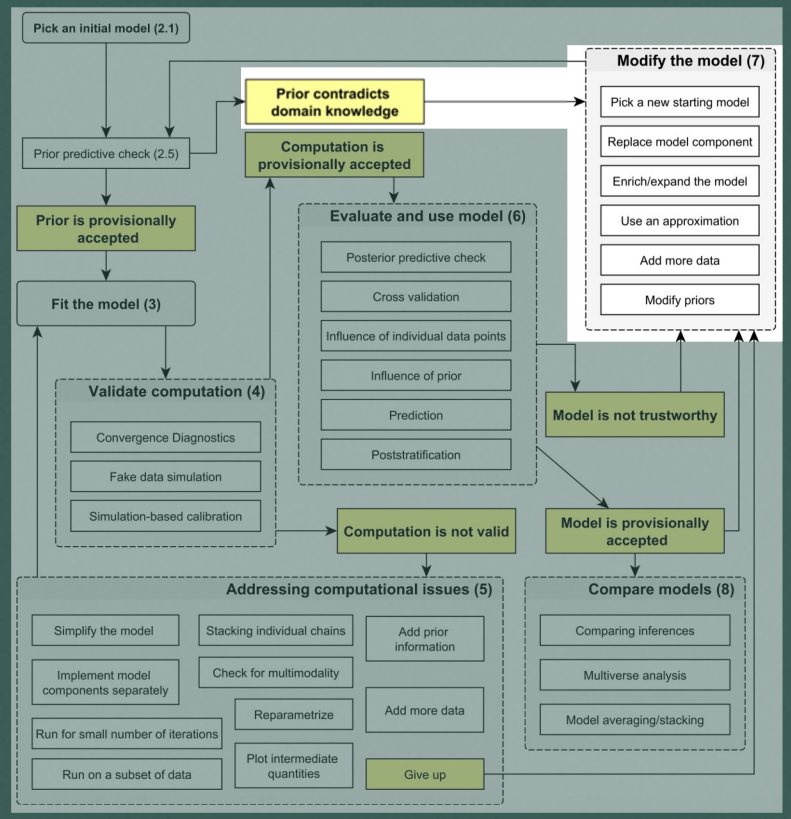


Prior Predictive Check

- What would happen if we chose a Gaussian/Normal likelihood instead of a Poisson?
- Why would we reject our model here?
- Model implies impossible values
  - Negative seizures

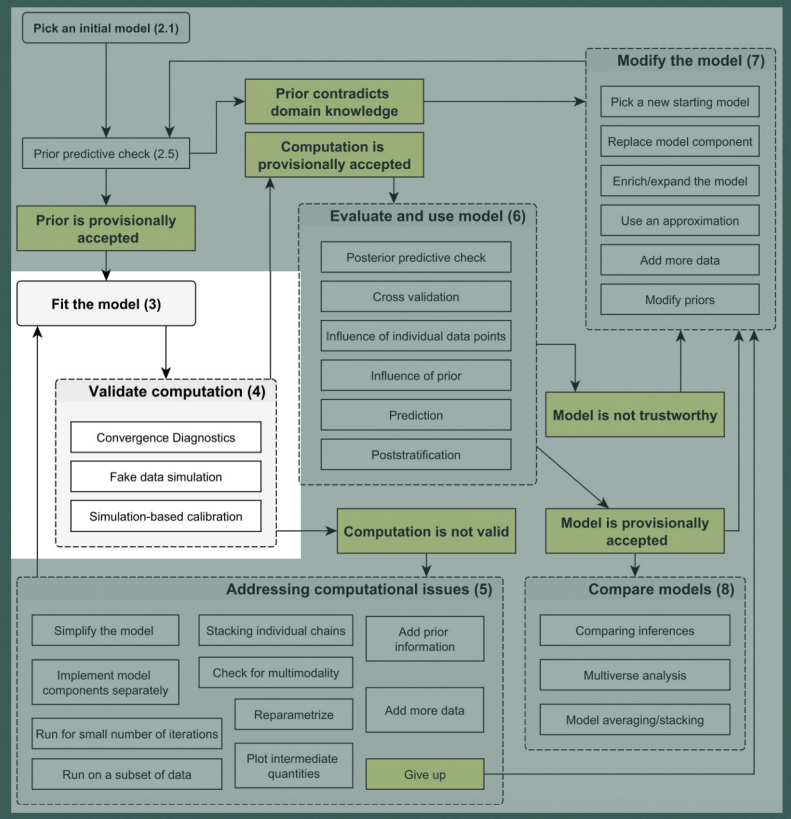


Prior Predictive Check - Failed



- If the model or priors do not appear appropriate, what should you do?
  - Re-consider the data-generating process (likelihood) you've chosen – what does it imply for the actual phenomenon?
    - E.g., negative seizure counts?
  - Check your prior choices
    - Do they imply impossible values?

Fit the Model & Validate



- Once we've run the model, how do we know we can trust the results?
  - Recall the sources of bias from MCMC
    - Finding the probability mass
    - Fully exploring the probability mass

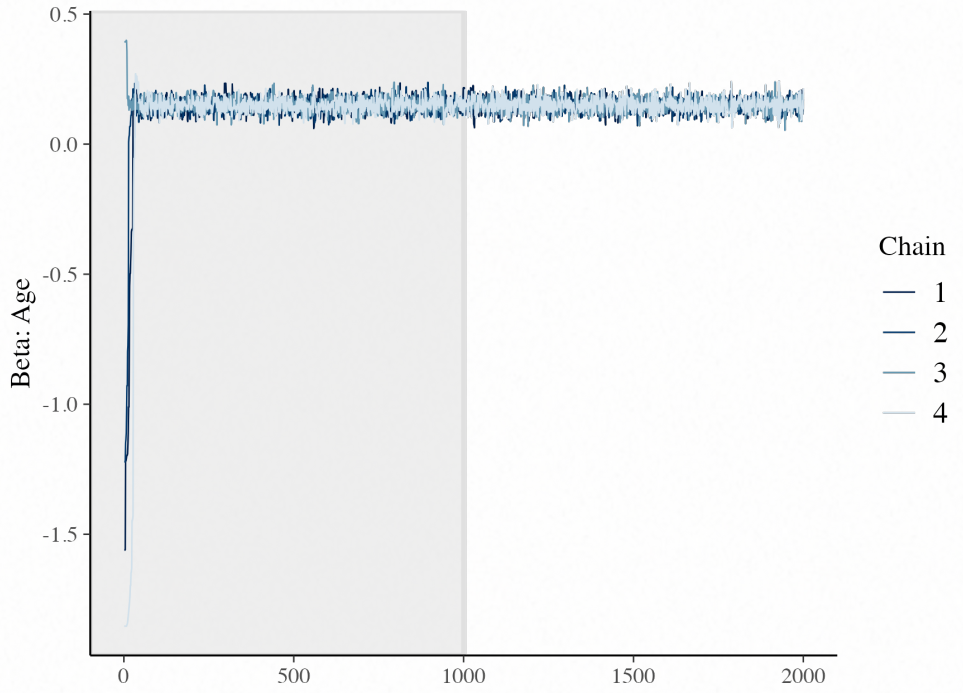
- Convergence diagnostics
  - Traceplots and $\hat{R}$-statistic

- Effective Sample Size

Traceplots

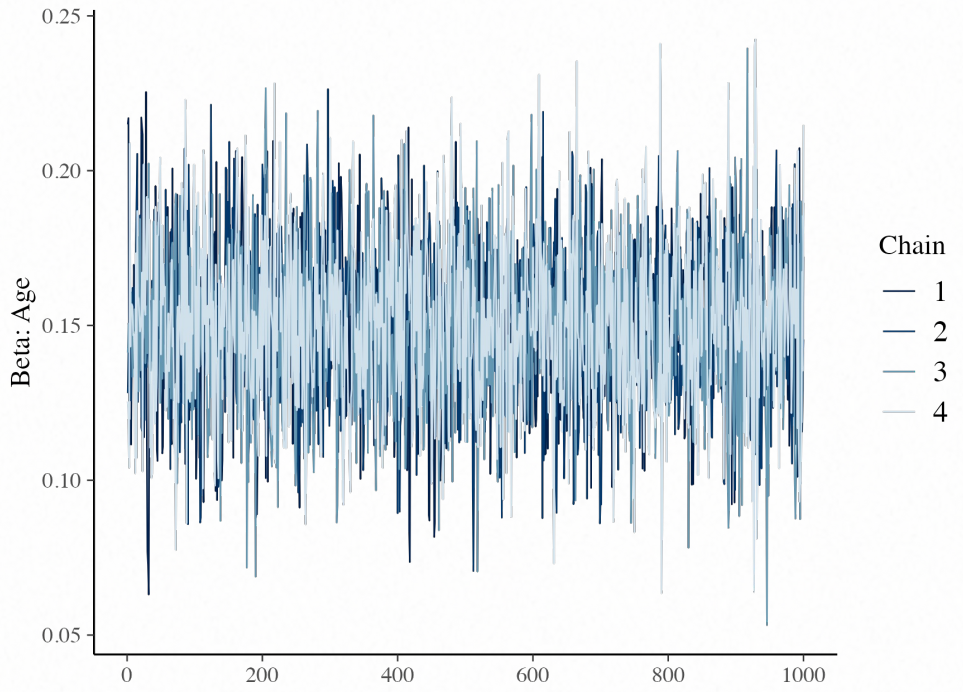
- Graphically show the progression of chains during sampling
- We want to see that all chains have found the same range of values
  - Also want to see that the full range of values has been sampled

- Notice the strange points as the start?
  - Finding the probability mass
  - Discarded as “warm-up”



Traceplots

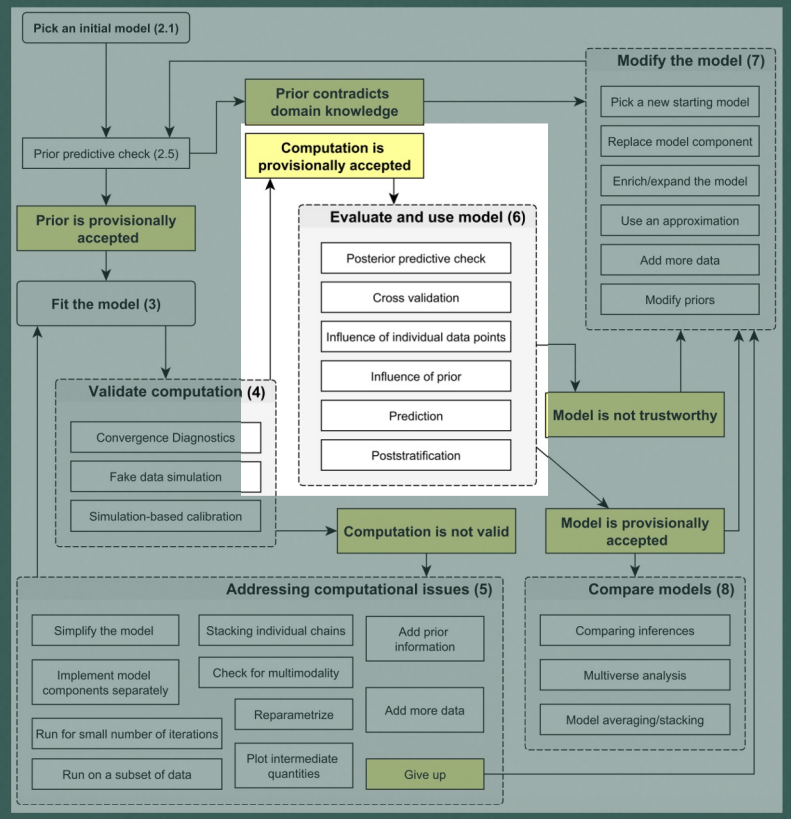
- Convergence
  - If all chains have found the same range of values, and have explored them fully
- An indication that the probability mass has been fully sampled
  - Start the chains from the diverse values to make sure
- Supplement graphical inspection with $\hat{R}$-statistic:
  - Summarises variance within and between chains
  - We want values near 1



Effective Sample Size

- MCMC estimator assumes *independent* samples
- The greater the dependence, the greater the uncertainty in our estimates
- Effective Sample Size (ESS) estimates the number of independent samples needed for the same uncertainty as our estimates
- We inspect in two forms:
  - Bulk ESS: ESS in the ‘body’ of our posterior
  - Tail ESS: ESS in the tails

Evaluate Model

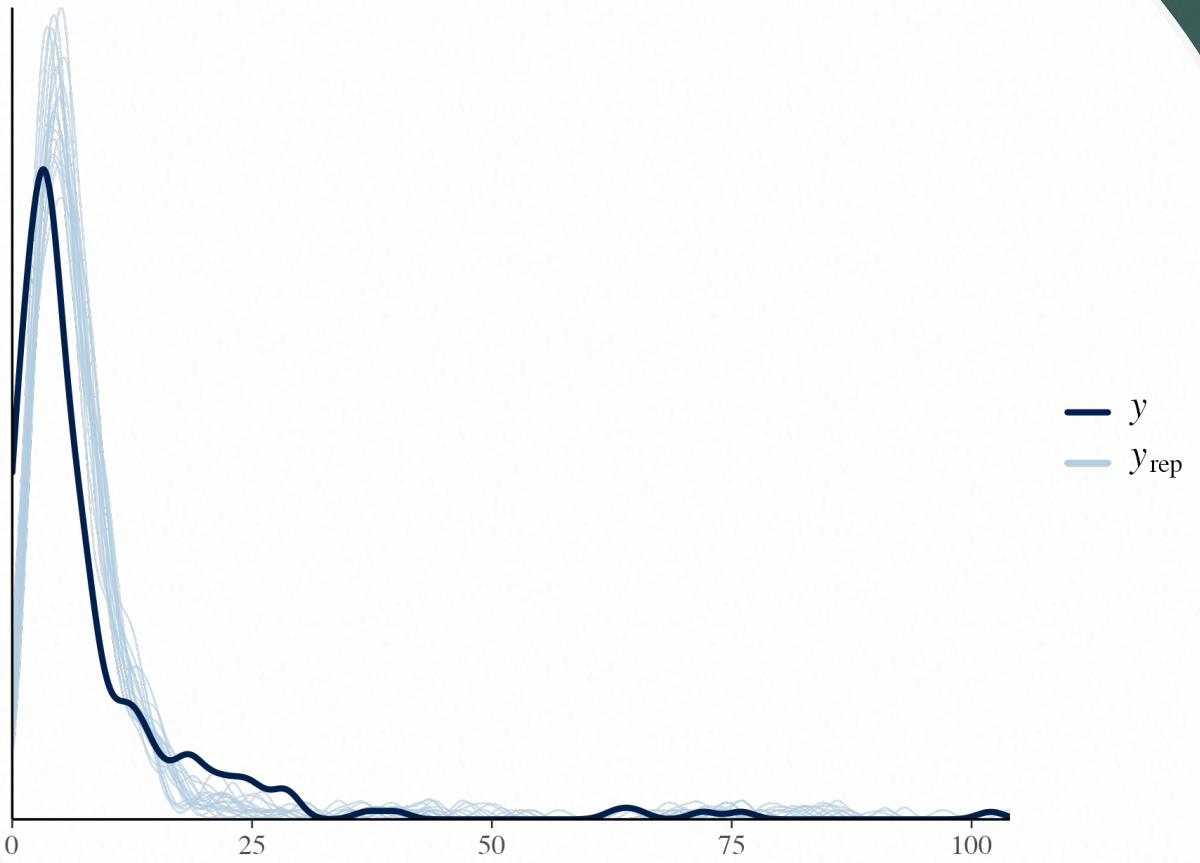


- Once satisfied that there are no computational issues, we can assess whether the model is appropriate for our data

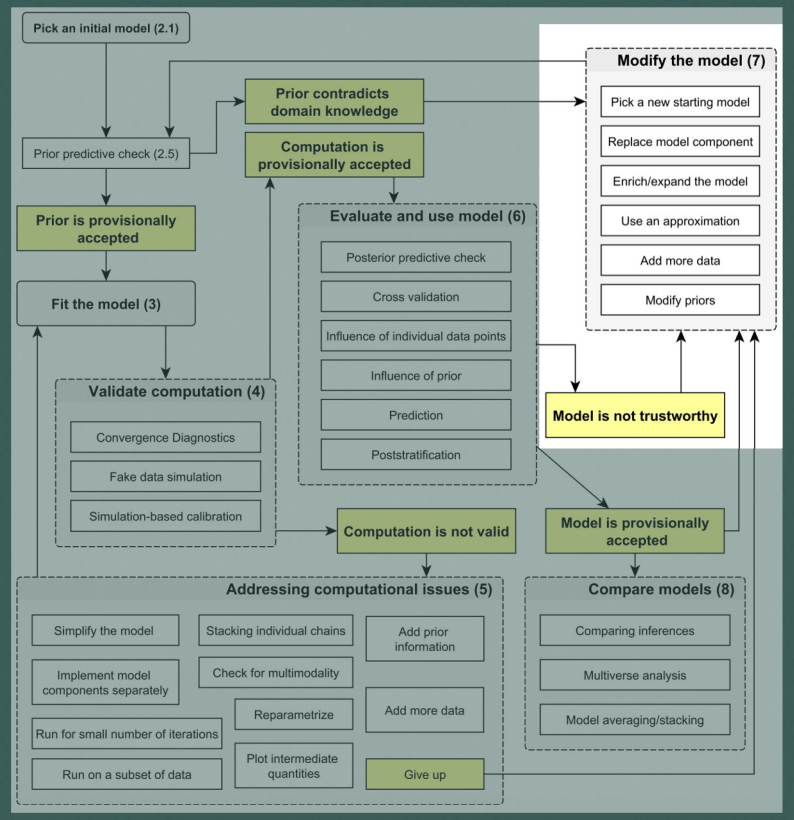
- Good first step is Posterior-Predictive Checking
  - Simulate data from the posterior and assess difference from our observations

Posterior Predictive Check

- Simulate data from the posterior and assess difference from our observations
- Model appears to predict more smaller values than we have observed
- How could we improve this?

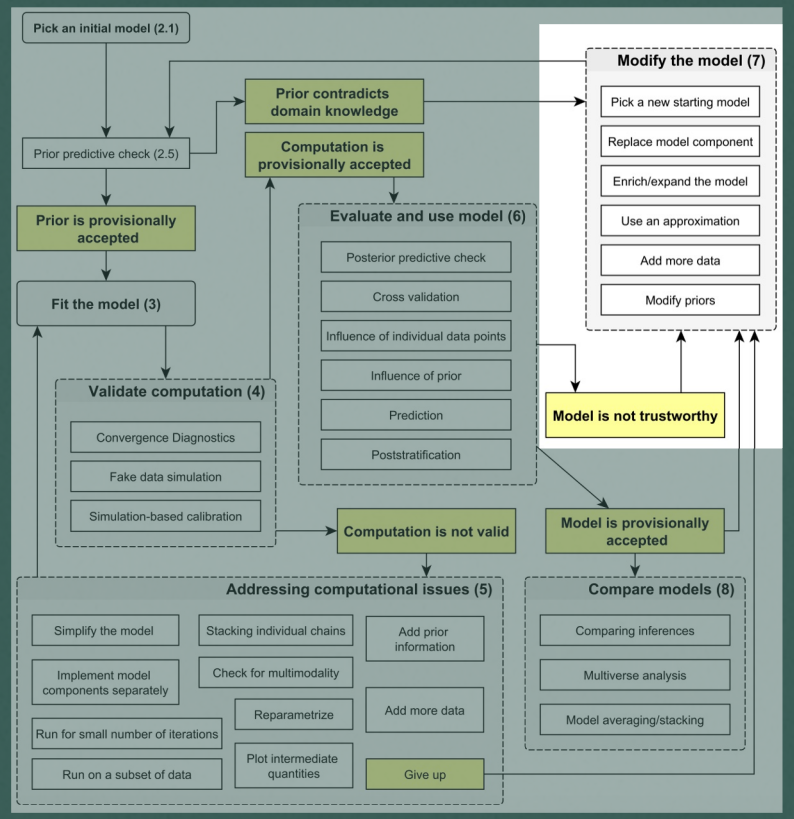


Reject Model



- What other model could we apply here?
- Consider the assumptions made by our model

Reject Model



- What other model could we apply here?
- Consider the assumptions made by our model
- Poisson assumes equal mean and variance
  - Let's add some flexibility by including a random effect
  - Allows for greater variance than mean
- New additions:

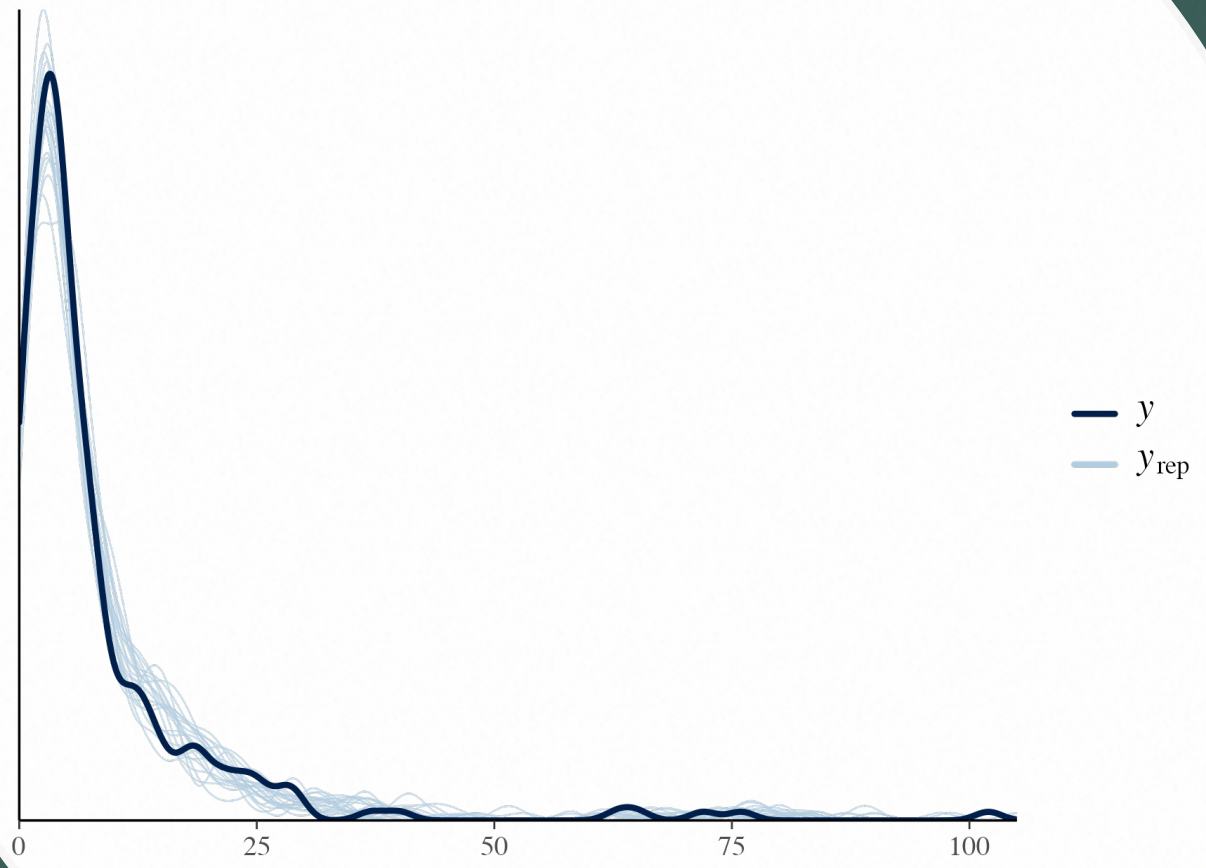
$$u_i \sim N(0, \sigma)$$

$$\sigma \sim Cauchy^{+}(0, 5)$$

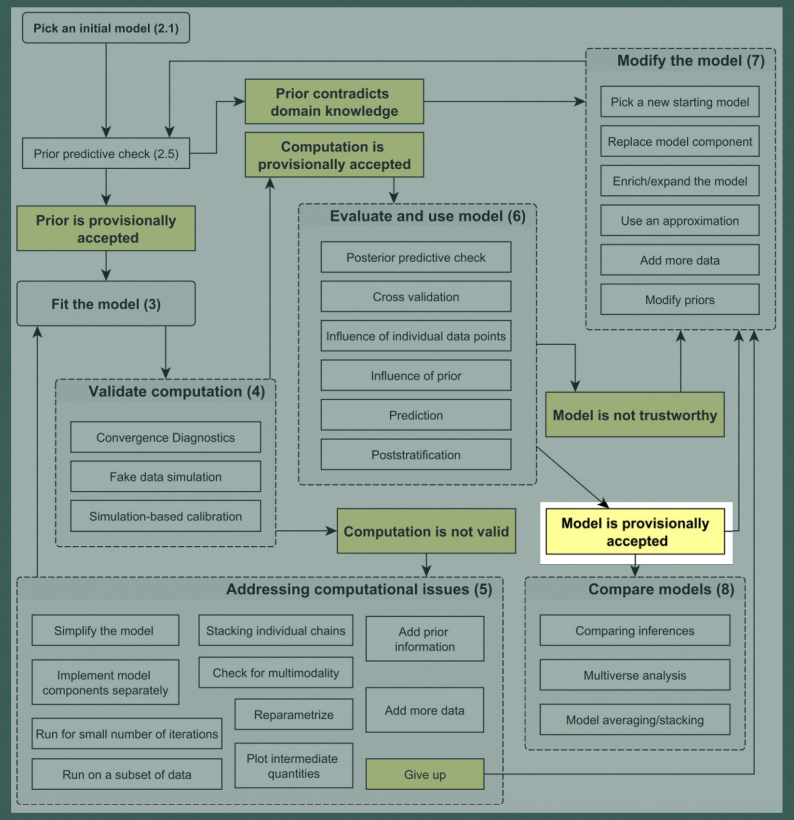
New Model

- The posterior-predictive checks show much better fit across the full range of values

- Improves the confidence we have when making inferences



Accept Model



- Now that we have a model and priors that we are happy with, what's next?
  - We can stop and evaluate our results – was there a difference between the treatment groups?
  - Is there an alternative model that we want to evaluate?
    - Our model now requires estimating a parameter per participant, can we avoid this?

Alternative Models

Our current model is:

$$y_i \sim \text{Poisson}(\lambda_i)$$

$$\lambda_i = \exp(\alpha + x_i^T \beta + u_i)$$

$$\alpha \sim \text{N}(0, 5)$$

$$\beta_{1:4} \sim N(0,1)$$

$$u_i \sim N(0, \sigma)$$

$$\sigma \sim Cauchy^+(0, 5)$$

What if we used a different distribution for our random effect u_i?

Alternative Models

- Let's try a Gamma distribution instead:

$$y_i \sim \text{Poisson}(\lambda_i \theta_i)$$

$$\lambda_i = \exp(\alpha + x_i^T \beta)$$

$$\alpha \sim \text{N}(0, 5)$$

$$\beta_{1:4} \sim N(0,1)$$

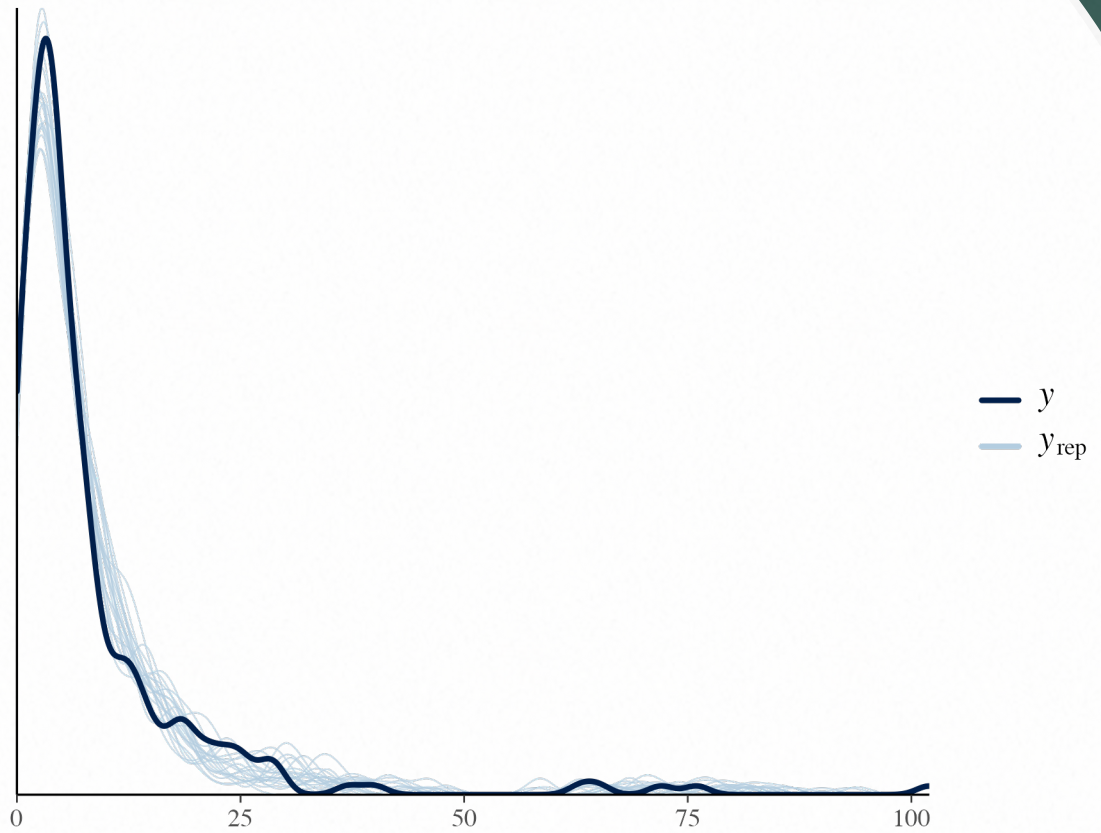
$$\theta_i \sim Gamma(\phi, \phi)$$

$$\phi \sim Cauchy^{+}(0, 5)$$

Alternative Models

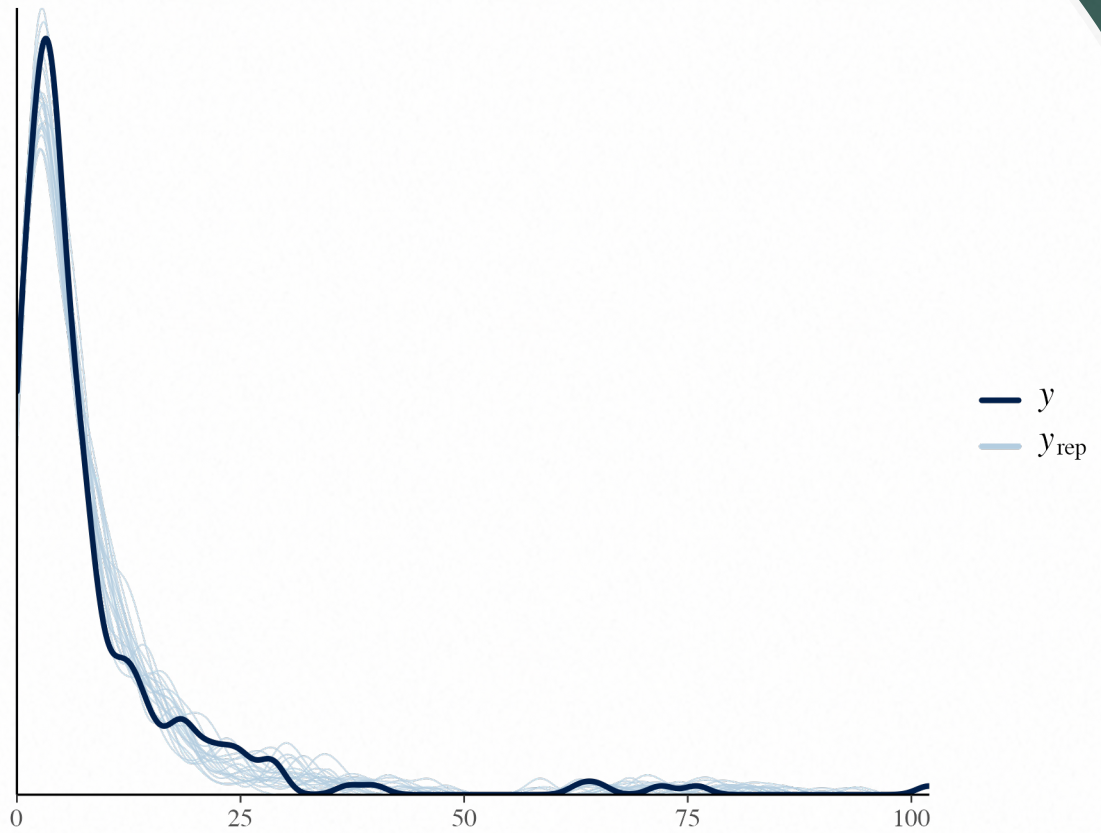
- Posterior-Predictive Check indicates that it also provides a good fit to the data

- But we're still estimating the same number of parameters, so what's the point?



Alternative Models

- Posterior-Predictive Check indicates that it also provides a good fit to the data
- But we're still estimating the same number of parameters, so what's the point?
- Marginalisation



Alternative Models - Marginalisation

- A useful tool for reducing computational burden/complexity

- Constructing the same posterior density while needing to estimate fewer parameters

Alternative Models - Marginalisation

- A useful tool for reducing computational burden/complexity
- Constructing the same posterior density while needing to estimate fewer parameters
- In this case, we can represent the Poisson with a Gamma-distributed random effect as a Negative-Binomial parameterised by its mean and dispersion:

$$\int Poisson(y \mid \lambda\theta) \cdot Gamma(\theta|\phi,\phi)d\theta = NB(y|\lambda,\phi)$$

- Estimate the same model with N fewer parameters!

Alternative Models - Marginalisation

- Let's try the Negative-Binomial Distribution:

$$y_i \sim \mathrm{NB}(\lambda_i, \phi)$$

$$\lambda_i = \exp(\alpha + x_i^T \beta)$$

$$\alpha \sim \mathrm{N}(0, 5)$$

$$\beta_{1:4} \sim N(0,1)$$

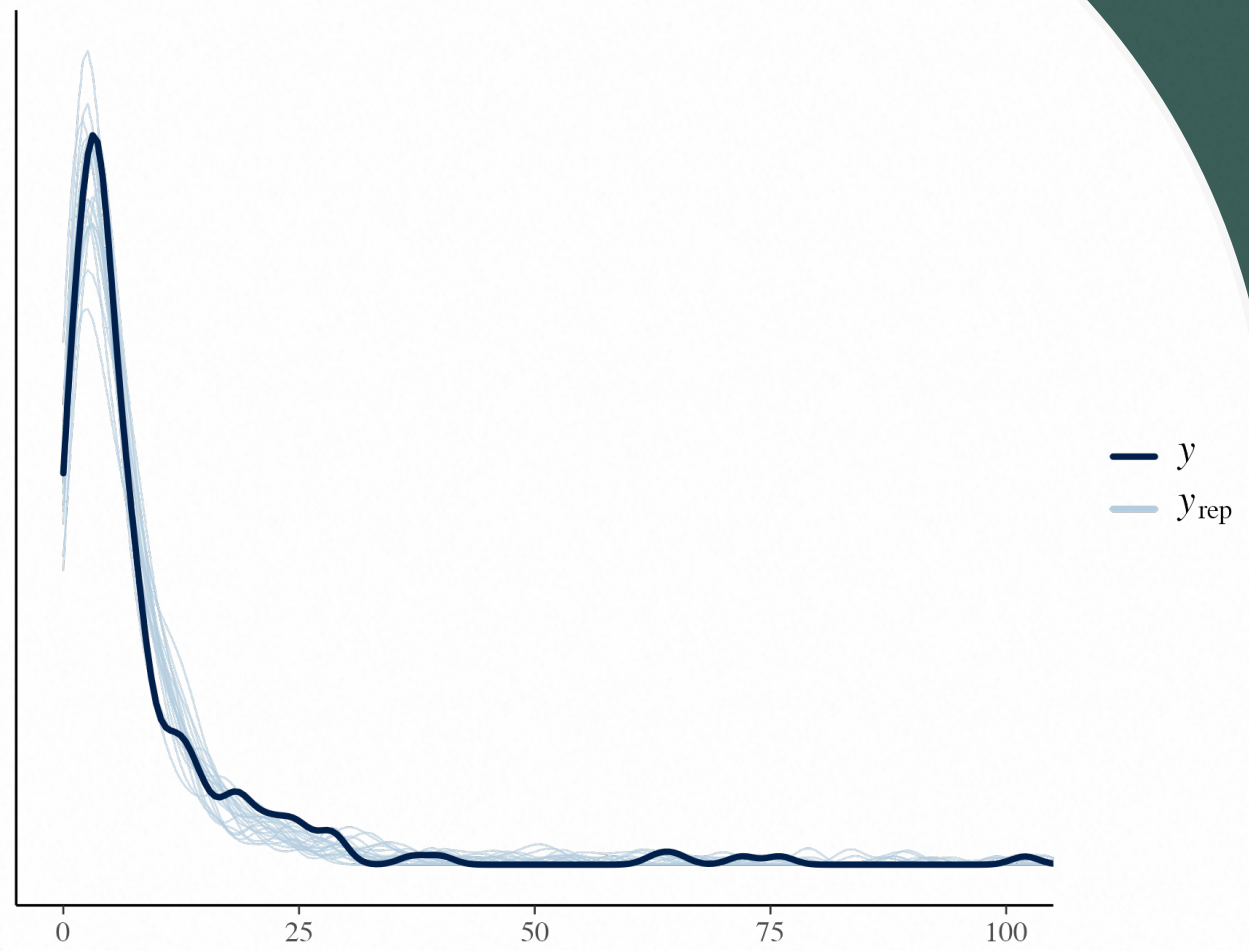
$$\phi \sim Cauchy^{+}(0, 5)$$

Alternative Models - Marginalisation

- Posterior-Predictive Check shows similarly good fit to the observed data

- Significantly reduced runtime!
  - Normal: 9.66 seconds
  - Gamma: 14.48 seconds
  - NB: 2.8 seconds

- We can do better!



Computational Efficiency

- Note the construction of the linear predictor requires multiplying data (x_i^T) with parameters (β):

$$\lambda_i = \exp(\alpha + x_i^T \beta)$$

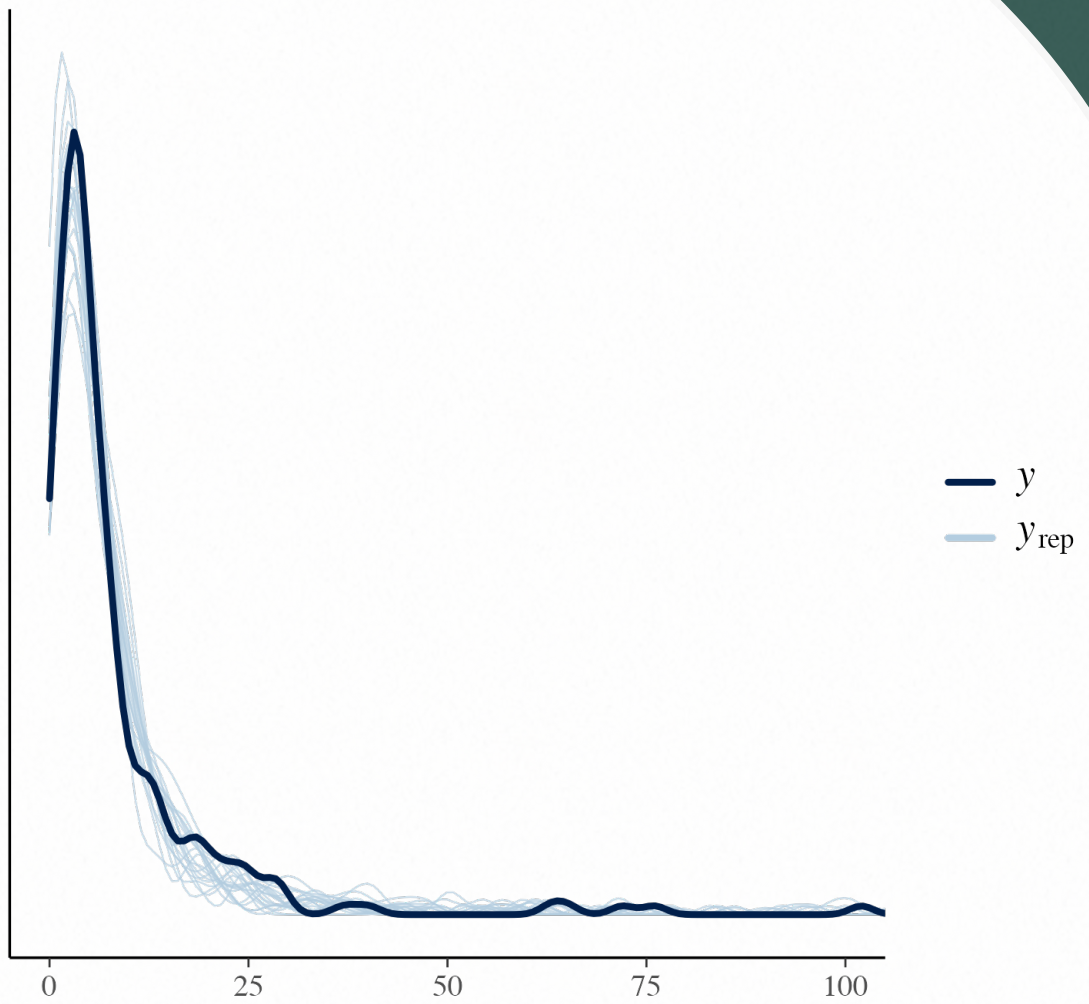
- This means we have to calculate and propagate gradients for each result

- We can instead take advantage of Stan's specialised GLM distributions to identify which components require gradient calculations

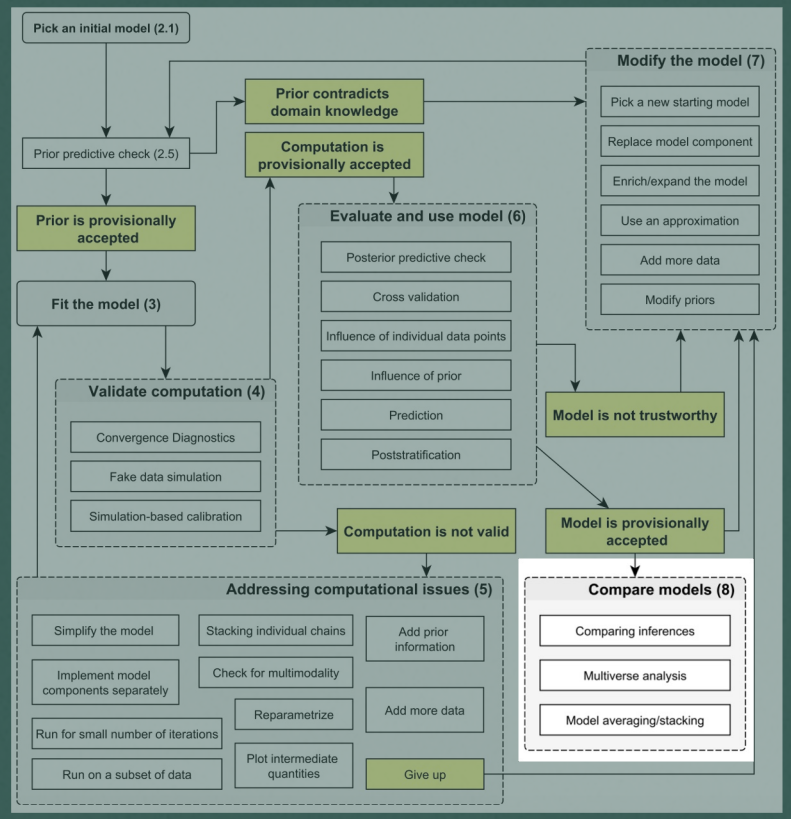
Alternative Models - GLM

- Posterior-Predictive Check is near-identical, as the likelihood is the same

- Significantly reduced runtime again!
  - Normal: 9.66 seconds
  - Gamma: 14.48 seconds
  - NB: 2.8 seconds
  - NB + GLM: 1.26 seconds

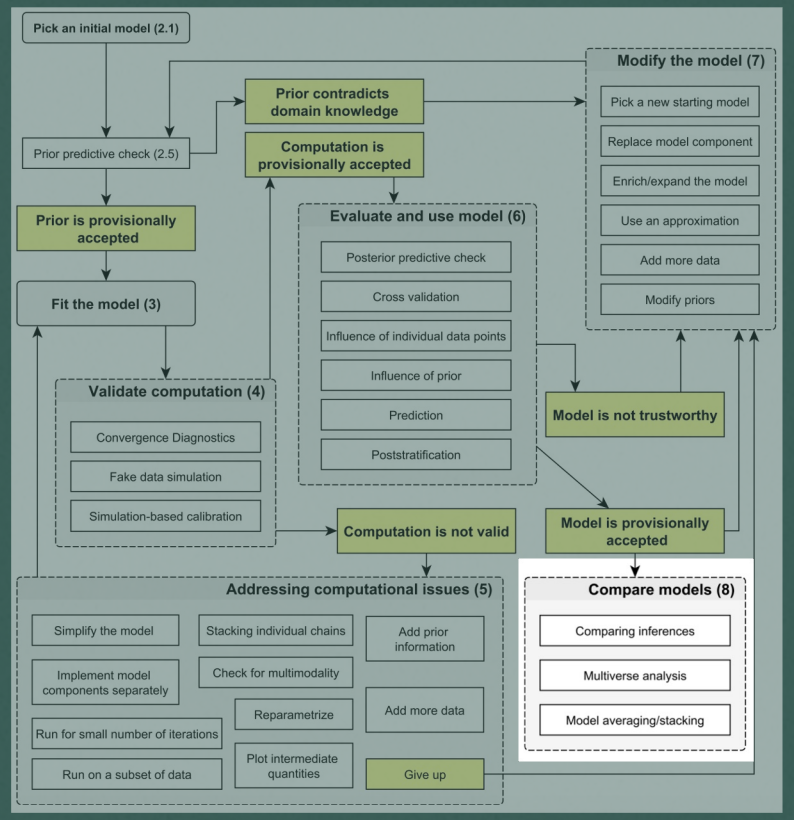


Compare Models



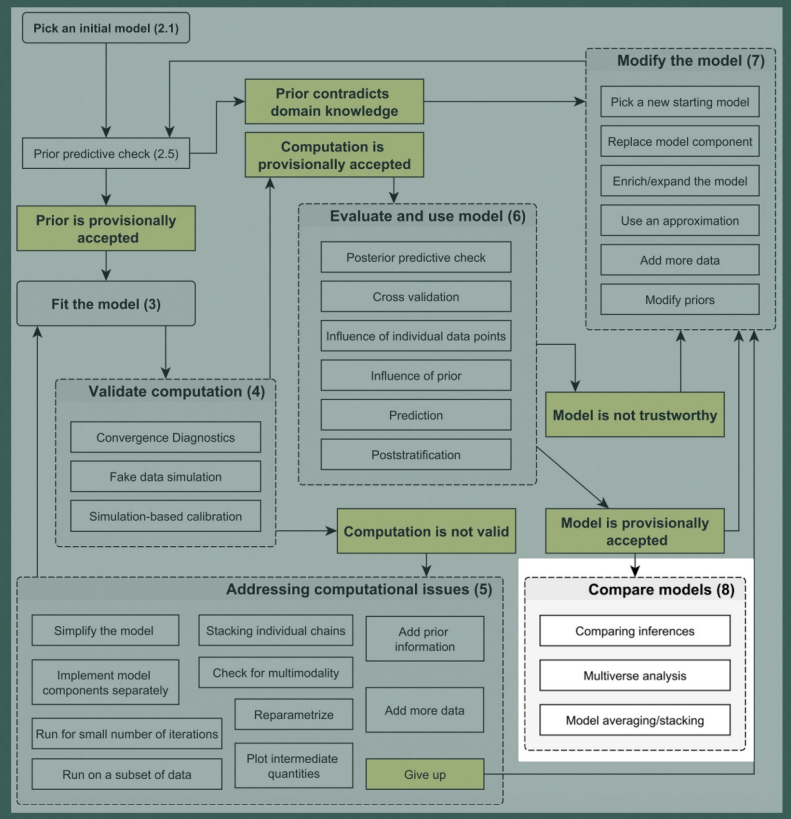
- What were our inferences about the treatment effect between the Poisson-Normal and the Negative-Binomial (GLM) models?

Compare Models



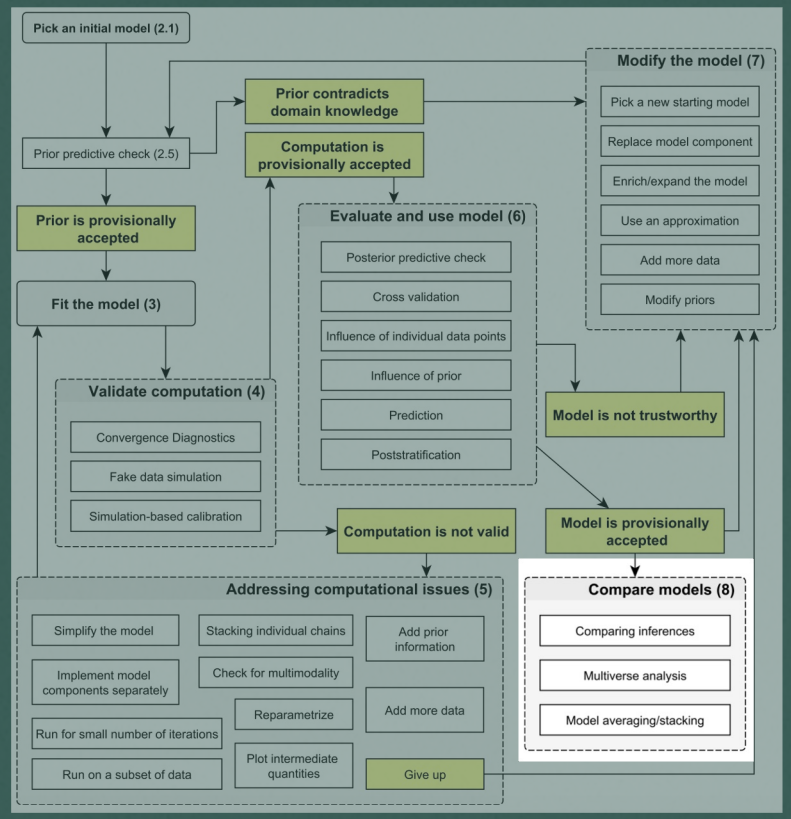
- What were our inferences about the treatment effect between the Poisson-Normal and the Negative-Binomial (GLM) models?
- Poisson:
  - $\beta_{treat} = -0.27\ [-0.54, 0.01]$
    - $ESS_{bulk} = 722$
    - $ESS_{tail} = 1199$

Compare Models



- What were our inferences about the treatment effect between the Poisson-Normal and the Negative-Binomial (GLM) models?
- Poisson:
  - $\beta_{treat} = -0.27\ [-0.54, 0.01]$
    - ESS_{bulk} = 722
    - ESS_{tail} = 1199
- Negative-Binomial:
  - $\beta_{treat} = -0.18\ [-0.36, -0.02]$
    - ESS_{bulk} = 2097
    - ESS_{tail} = 2075

Compare Models



- What were our inferences about the treatment effect between the Poisson-Normal and the Negative-Binomial (GLM) models?

- The Negative-Binomial model was both faster and resulted in higher quality estimates

Conclusions

- Bayesian inference provides powerful addition to our modelling toolkit
- Multiple pathways we can take in developing a Bayesian model
- Following a structured workflow can help reduce unnecessary work and ad-hoc decision-making