

Bayesian data analysis: Theory & practice

Part 2: Parameter estimation, MCMC & simple linear regression in BRMS

Michael Franke

Practical goal for today

master this!

```
Family: gaussian  
Links: mu = identity; sigma = identity  
Formula: murder_rate ~ unemployment  
Data: murder_data (Number of observations: 20)  
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	-28.48	7.32	-42.05	-13.79	1.00	3014	2362
unemployment	7.07	1.04	4.97	9.04	1.00	2978	2451

Family Specific Parameters:

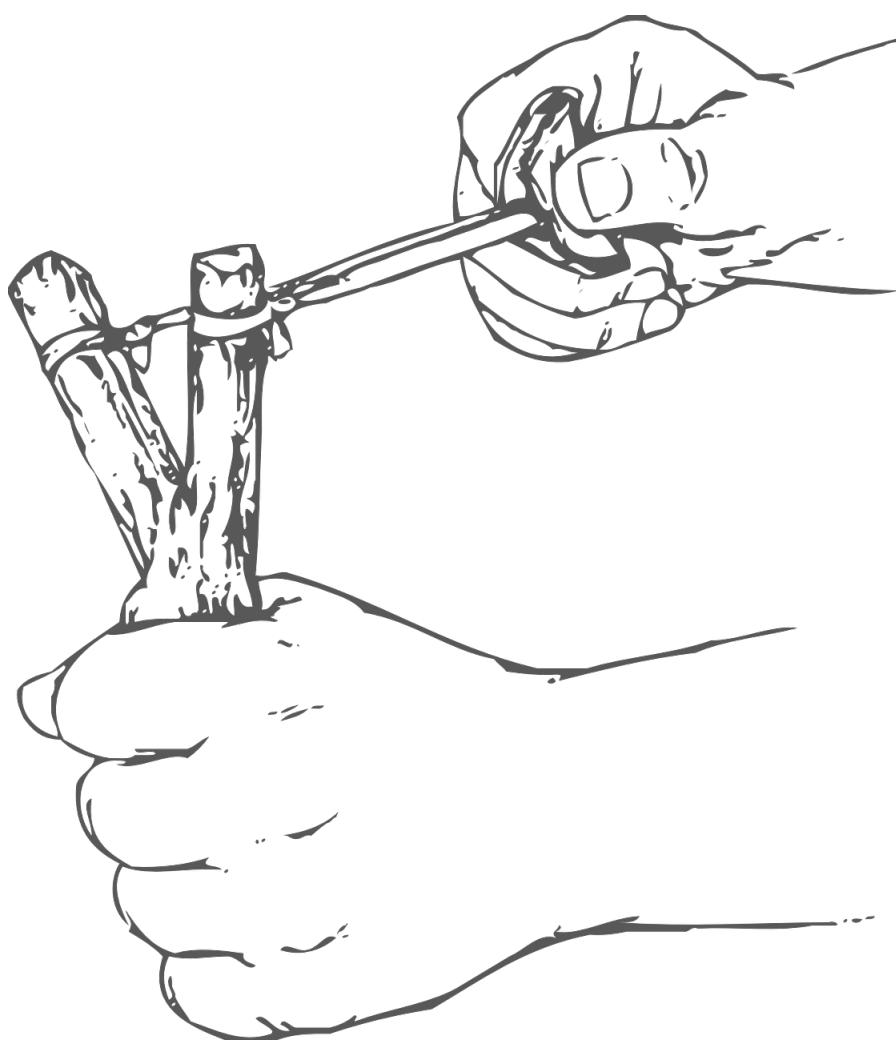
	Estimate	Est.Error	l-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	5.42	0.96	3.88	7.63	1.00	2664	2196

Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

Main learning goals

for this part

1. Bayesian parameter estimation
 - a. sampling-based approaches
 - b. Bayesian summary statistics
2. Markov Chain Monte Carlo sampling
 - a. Metropolis Hastings
 - b. Hamiltonian Monte Carlo
 - c. diagnostics and tuning-parameters
3. Bayesian regression with the BRMS package
 - a. extracting samples
 - b. priors & predictive functions
4. multiple regression w/ categorical predictors
 - a. contrast coding
 - b. derived variables in Bayesian analysis





parameter estimation

Computing posterior distributions

problem of computational complexity

$$P(\theta | D) = \frac{P(D | \theta) \pi(\theta)}{\int P(D | \theta) \pi(\theta) d\theta}$$

The diagram illustrates the components of the posterior distribution formula:

- The numerator $P(D | \theta)$ is annotated with two green checkmarks and the text "fast & easy".
- The denominator $\int P(D | \theta) \pi(\theta) d\theta$ is annotated with a red X and the text "possibly intractable" followed by a skull icon.

Excursion: Posteriors from conjugacy

closed-form posteriors from clever choice of priors

- ▶ prior $P(\theta)$ is a **conjugate prior** for likelihood $P(D | \theta)$ iff prior $P(\theta)$ and posterior $P(\theta | D)$ are the same kind of probability distribution, e.g.:
 - prior: $\theta \sim \text{Beta}(1,1)$
 - posterior: $\theta | D \sim \text{Beta}(8,18)$
- ▶ **claim:** the beta distribution is a conjugate prior for the binomial likelihood function
 - proof:

$$P(\theta | k, N) \propto \text{Binomial}(k; N, \theta) \text{ Beta}(\theta | a, b)$$

$$P(\theta | k, N) \propto \theta^k (1 - \theta)^{N-k} \theta^{a-1} (1 - \theta)^{b-1}$$

$$P(\theta | k, N) \propto \theta^{k+a-1} (1 - \theta)^{N-k+b-1}$$

$$P(\theta | k, N) = \text{Beta}(\theta | k + a, N - k + b)$$

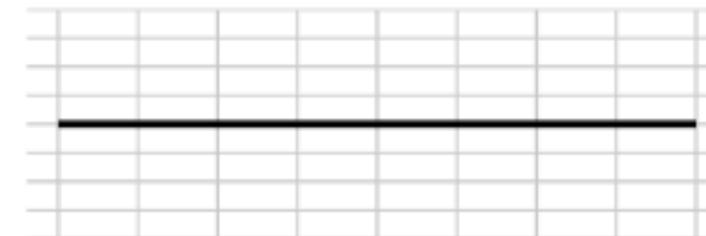


Excursion: Sequential updating

for the beta-binomial model

- ▶ sequence of updating does not matter
 - any order of single-observation updates
 - any ‘chunking’: whole data set, different subsets in whatever sequence (as long as disjoined)
- ▶ “today’s posterior is tomorrow’s prior”

$a=1$ $b=1$



$a=1$ $b=2$



$a=1$ $b=3$



$a=2$ $b=1$



$a=2$ $b=2$



$a=2$ $b=3$



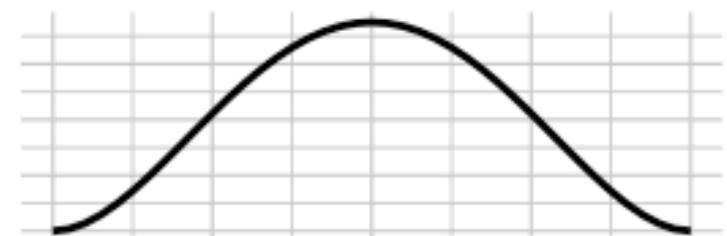
$a=3$ $b=1$



$a=3$ $b=2$



$a=3$ $b=3$

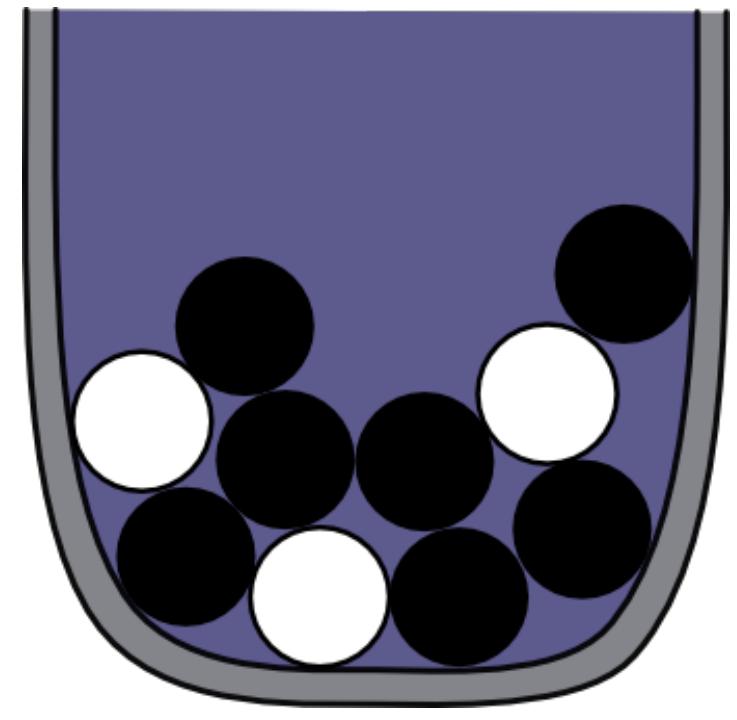
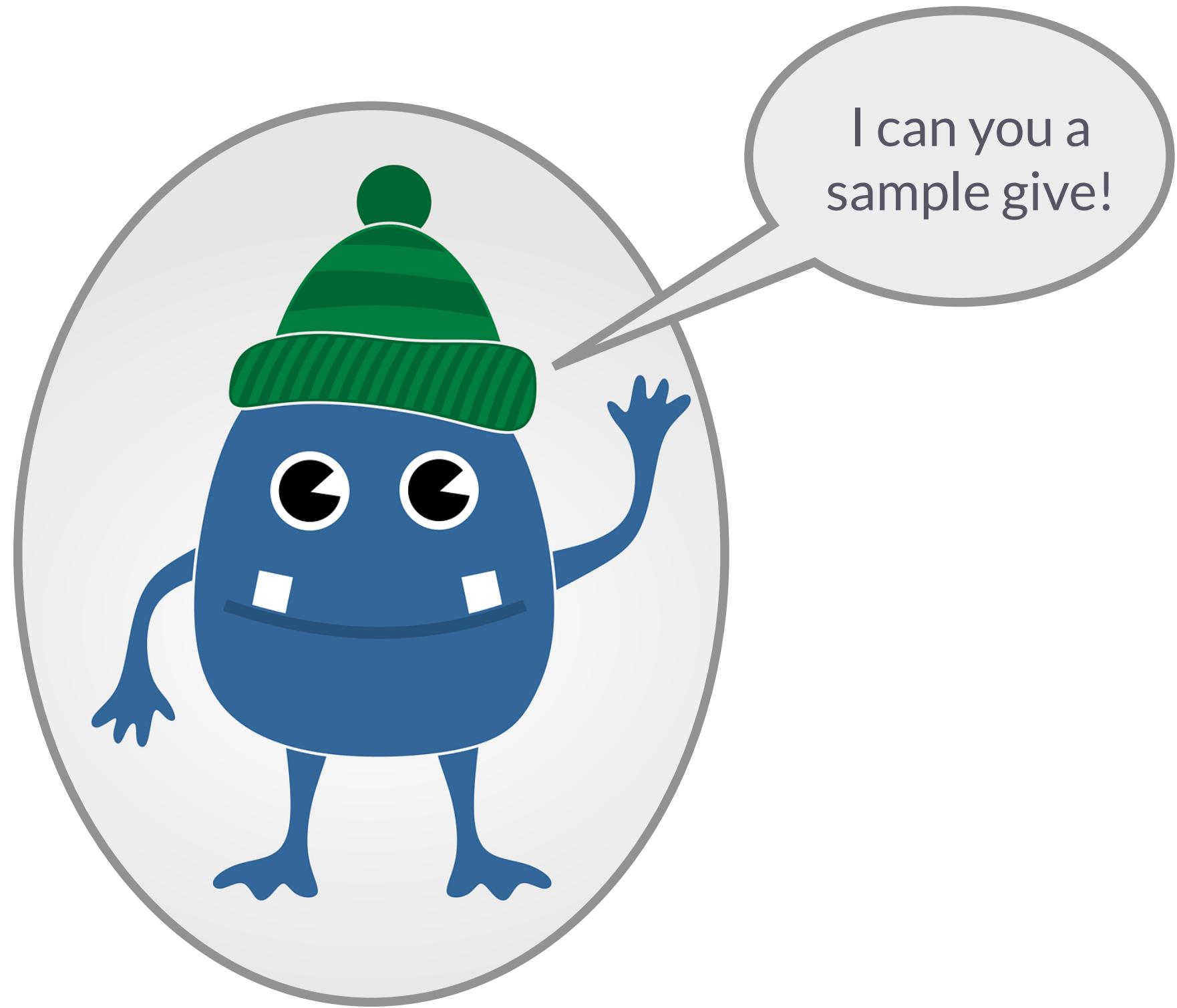


read more [here](#)

Approximating distributions via sampling

our go-to solution for approximating posterior distributions beyond conjugacy

- ▶ we can approximate many summary statistics of a probability distribution by either:
 - a large set of representative samples; or
 - an oracle that returns a sample if needed.



Temporal development of the proportion of draws from an urn





summary statistics

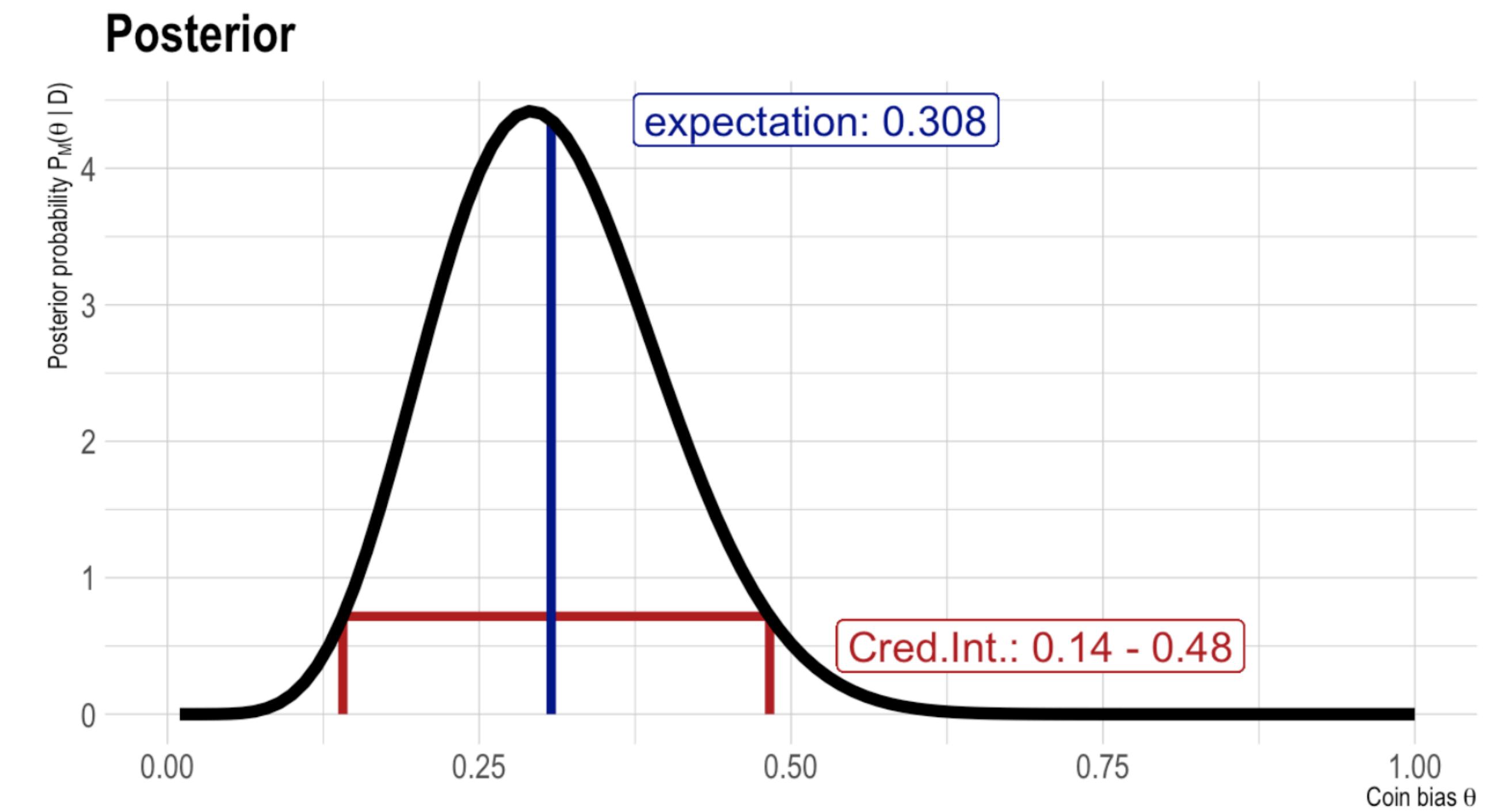
Parameter estimation

point- and interval-valued estimates

- ▶ Bayes' rule for parameter estimation:

$$P(\theta | D) = \frac{P(D | \theta) P(\theta)}{\int P(D | \theta) P(\theta) d\theta}$$

- ▶ common point estimates (“best” values):
 - maximum likelihood estimate (MLE)
 - maximum a posteriori (MAP)
 - **posterior mean / expected value**
- ▶ common interval estimates (range of “good” values):
 - confidence intervals
 - inner quantile ranges
 - **credible intervals**



read more [here](#)

Point-valued estimates

MLE, MAP and (posterior) expected value

► MLE:

$$\arg \max_{\theta} P(D | \theta)$$

- doesn't take prior into account (not Bayesian)
- not necessarily unique

► MAP:

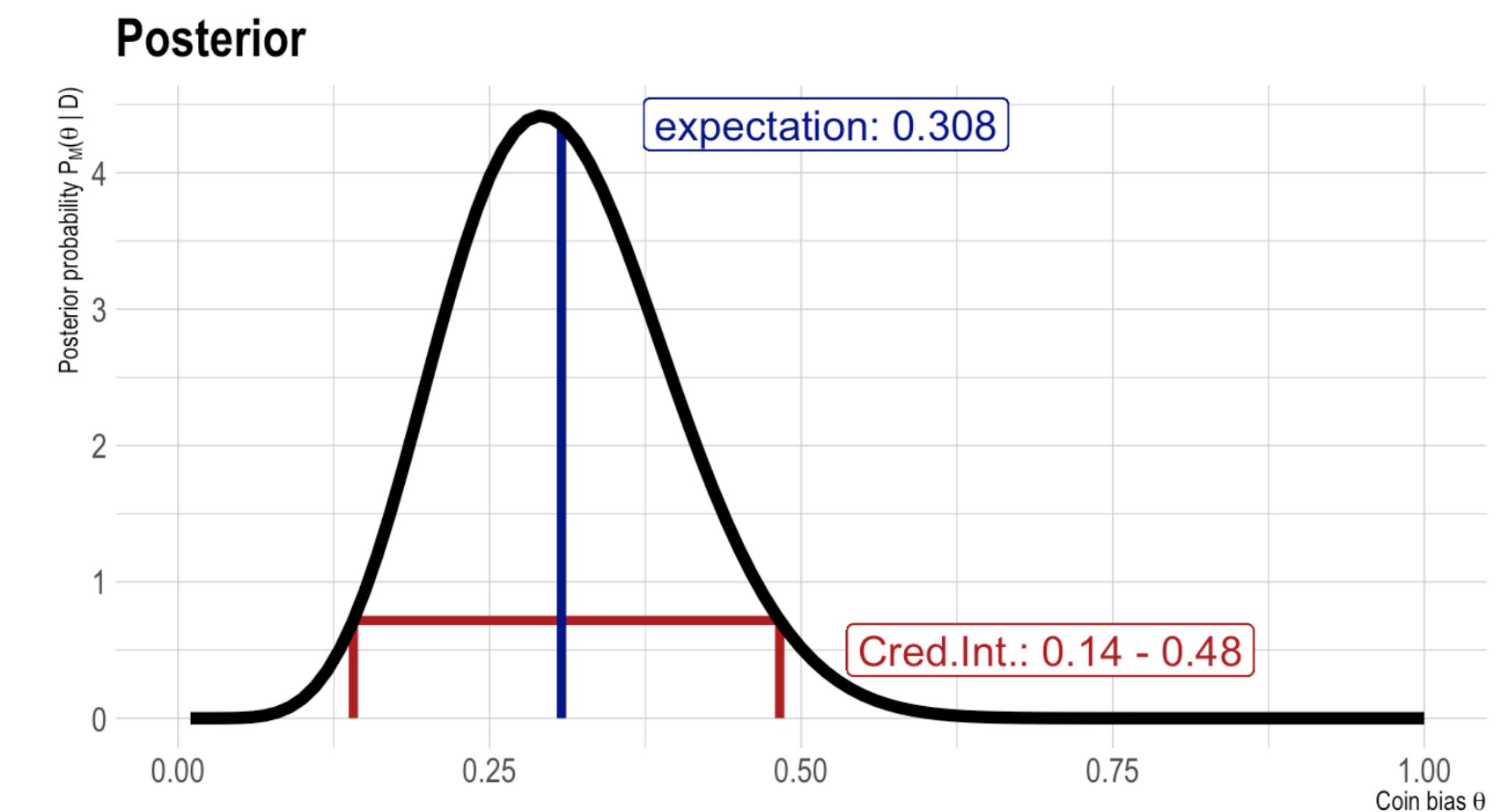
$$\arg \max_{\theta} P(\theta | D)$$

- local / does not consider full distribution (not fully Bayesian)
- increasingly uninformative in larger parameter spaces
- not necessarily unique

► posterior mean / expected valued

$$\mathbb{E}_{P(\theta|D)} = \int \theta P(\theta | D) d\theta$$

- holistic / depends on full distribution ("genuinely Bayesian")
- always unique (for proper priors/posteriors)



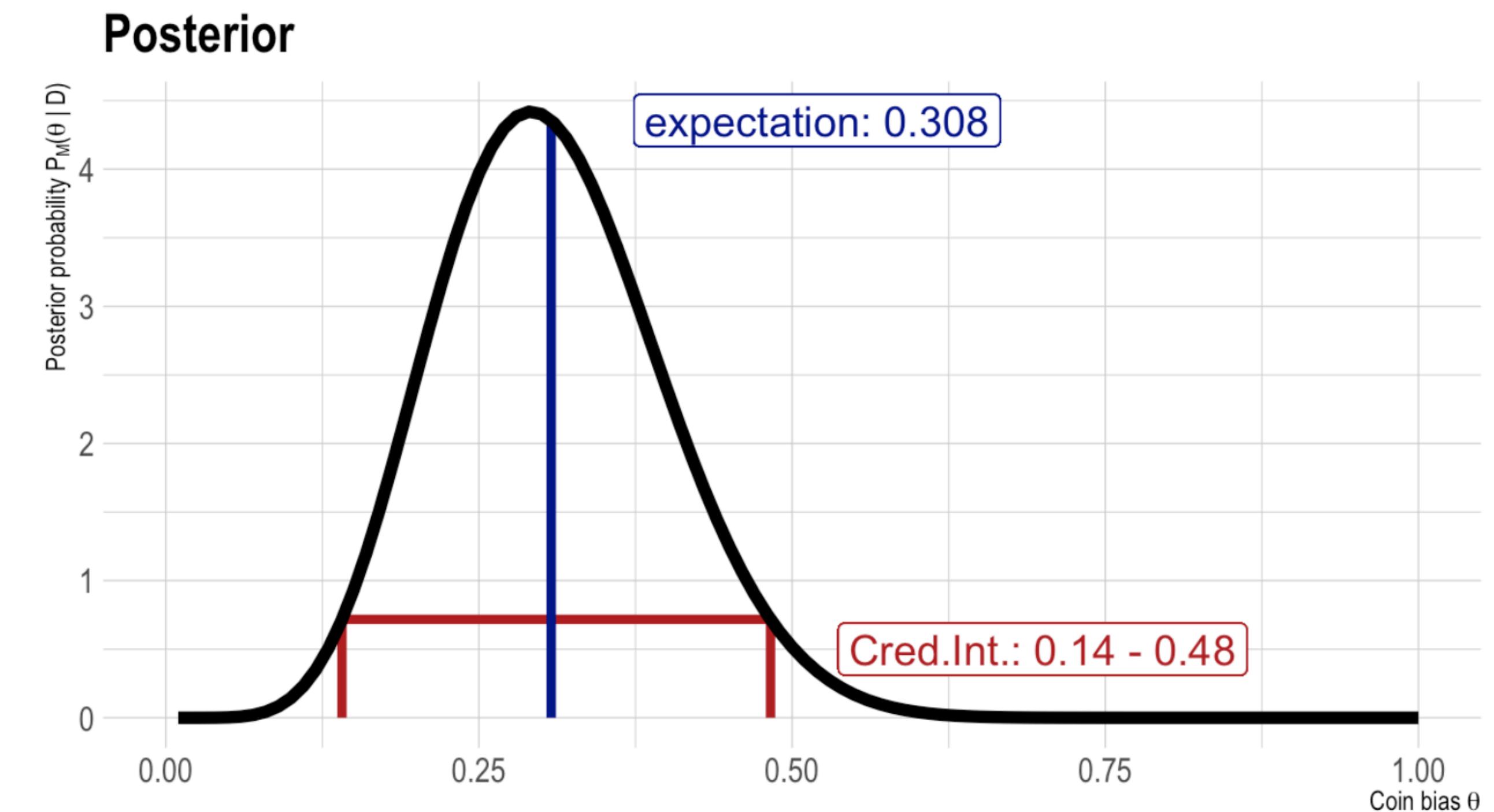
read more [here](#)

Bayesian credible intervals

a.k.a.: highest density interval ...

An interval I is a $\gamma\%$ credible interval, if:

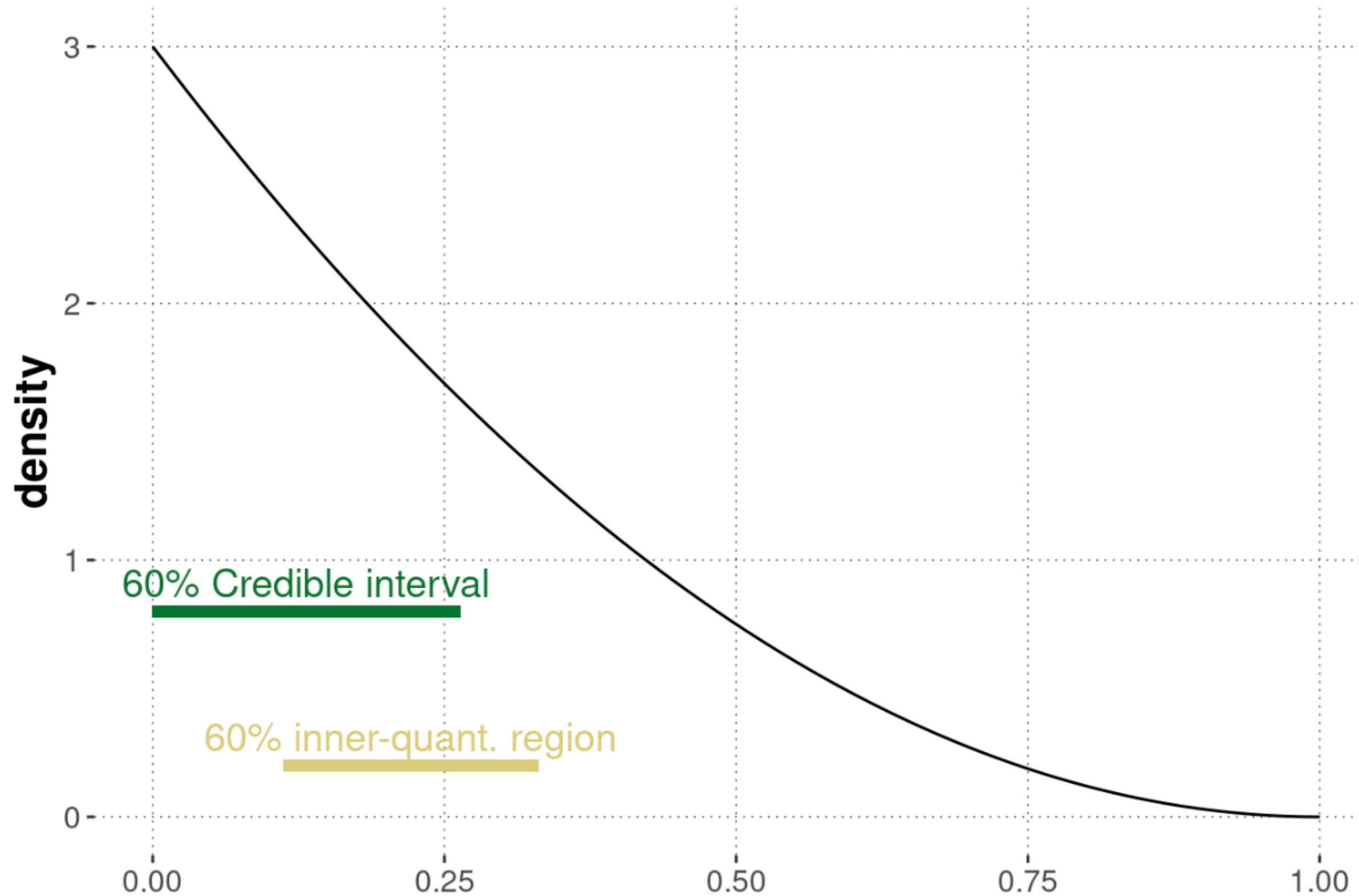
1. the probability that $\theta \in I$ is $\frac{\gamma}{100}$, and
2. no value outside of I is more likely than any point inside of I .



read more [here](#)

Inner quantile regions

!!!are not credible intervals!!!



read more [here](#)

Post-everything choice of bounds

if you want to signal that it's all arbitrary

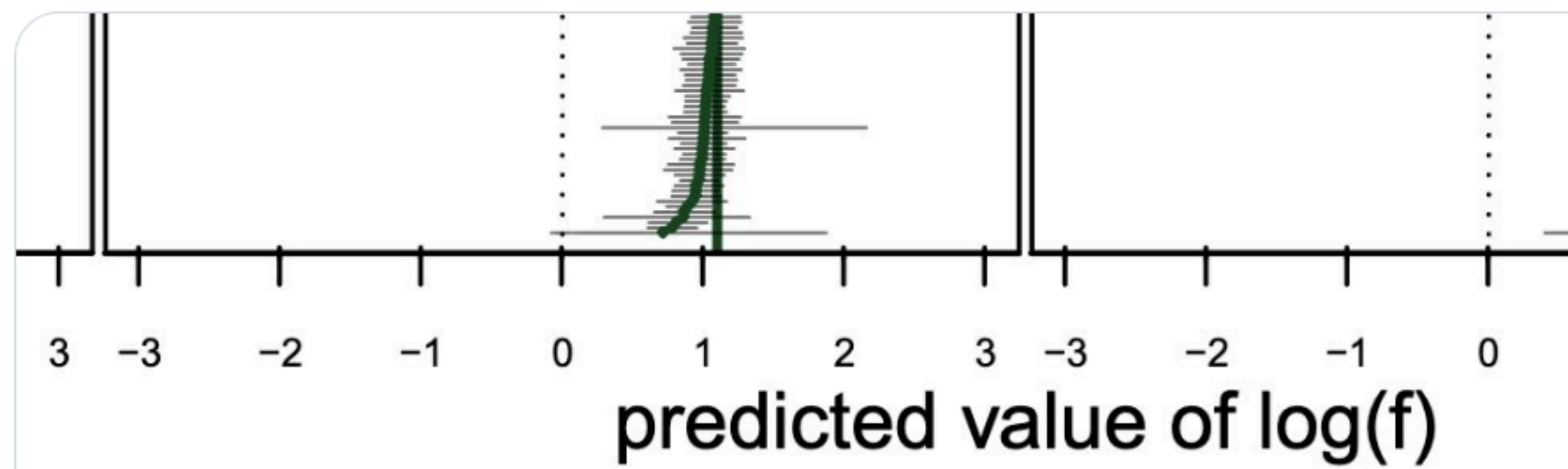


Richard McElreath 🦁

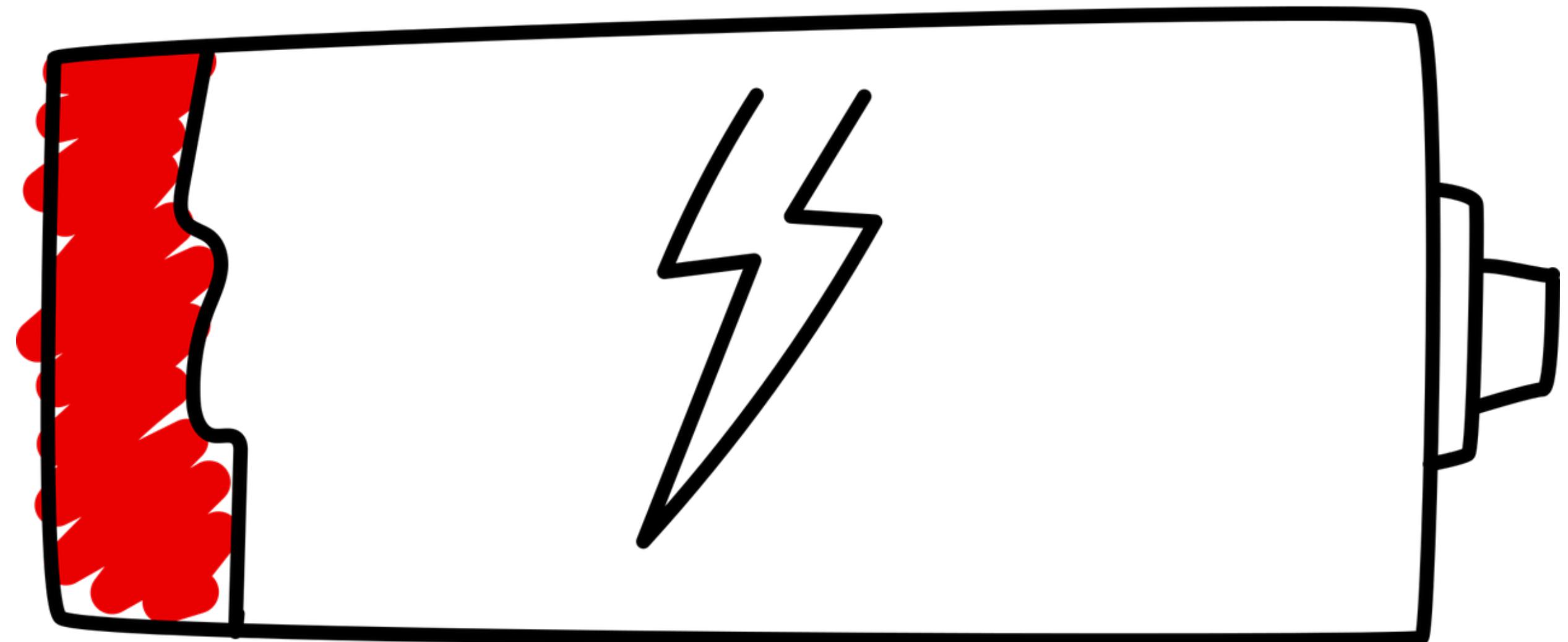
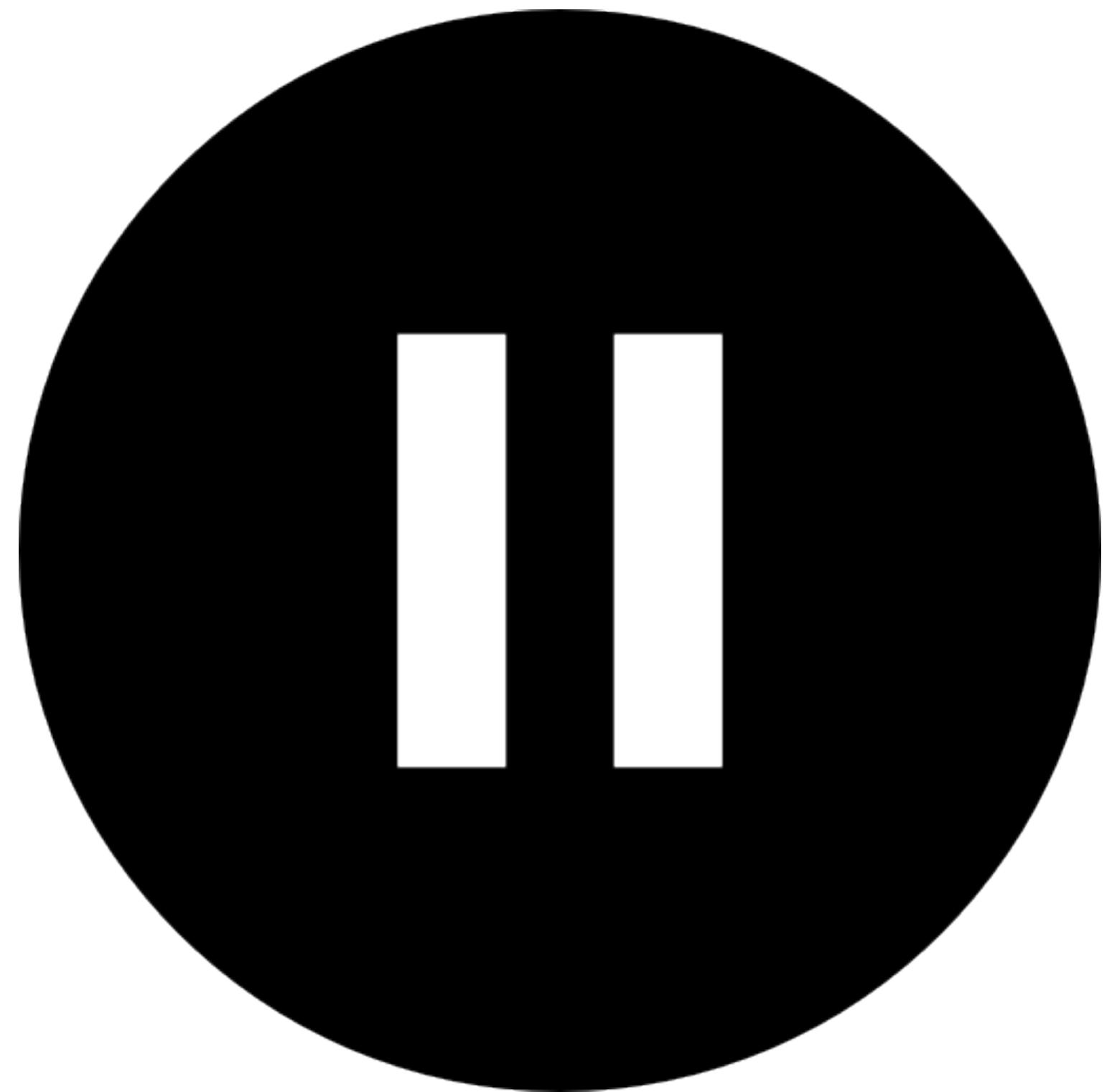
@rlmcelreath

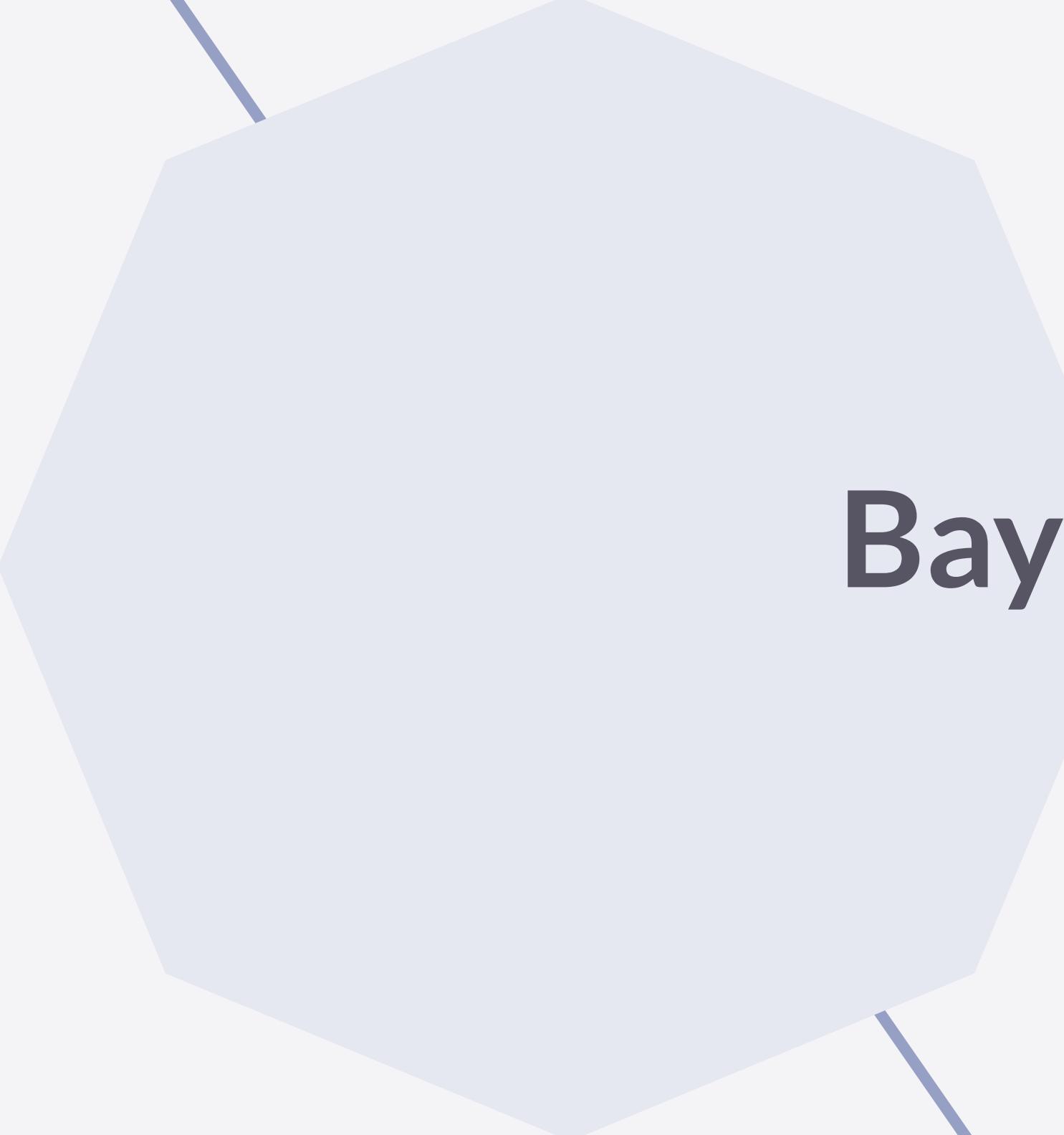
...

This paper sets the new standard for idiosyncratic punk posterior intervals: $30\pi\%$



or predictions of $\log(f)$ for 100 simulations of positive frequency $\alpha = 5, 10, 25, 50, 100, 250$, number of options increases according to the equation and $30\pi\%$ highest posterior density intervals (HPDIs) are used to assess social learning, with negative frequency-dependent learning. The solid coloured vertical line is at known simulated value of $\log(f)$.





Bayesian regression in BRMS

Bayesian linear regression in R

using BRMS and Stan

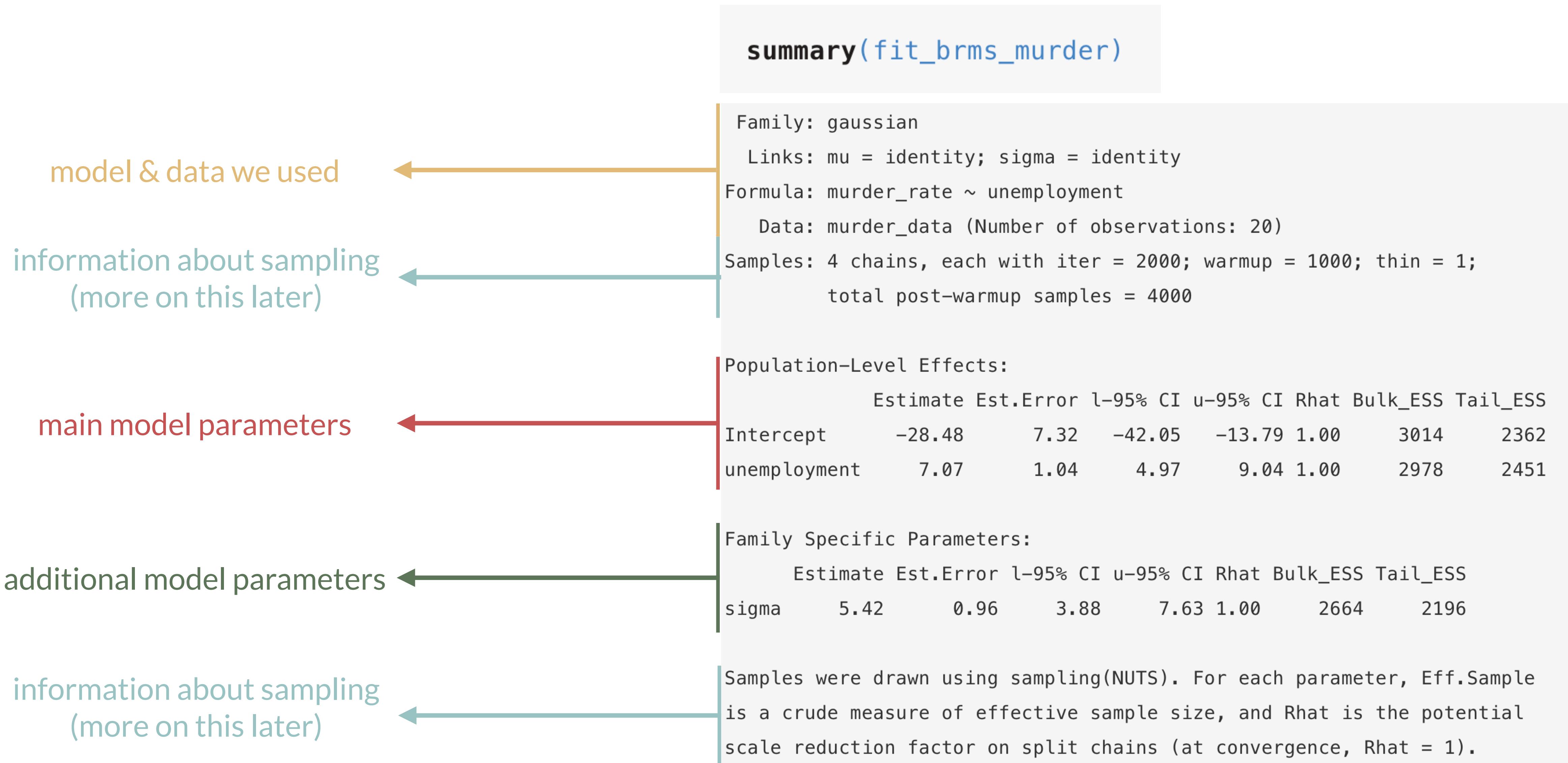
- ▶ R package BRMS provides high-level interface for Bayesian linear regression
- ▶ models are specified with R's formula syntax
- ▶ returns samples from the posterior distribution
 - alternatives: MAPs, variational inference
- ▶ builds on probabilistic programming language Stan
 - powerful, cutting-edge tool for Bayesian computation
 - strong, non-commercial development team
 - many interfaces: stand-alone, R, Python, Julia, ...

```
fit_brms_murder <- brm(  
  # specify what to explain in terms of what  
  # using the formula syntax  
  formula = murder_rate ~ unemployment,  
  # which data to use  
  data = murder_data  
)
```



Stan

Navigating BRMS output



demo



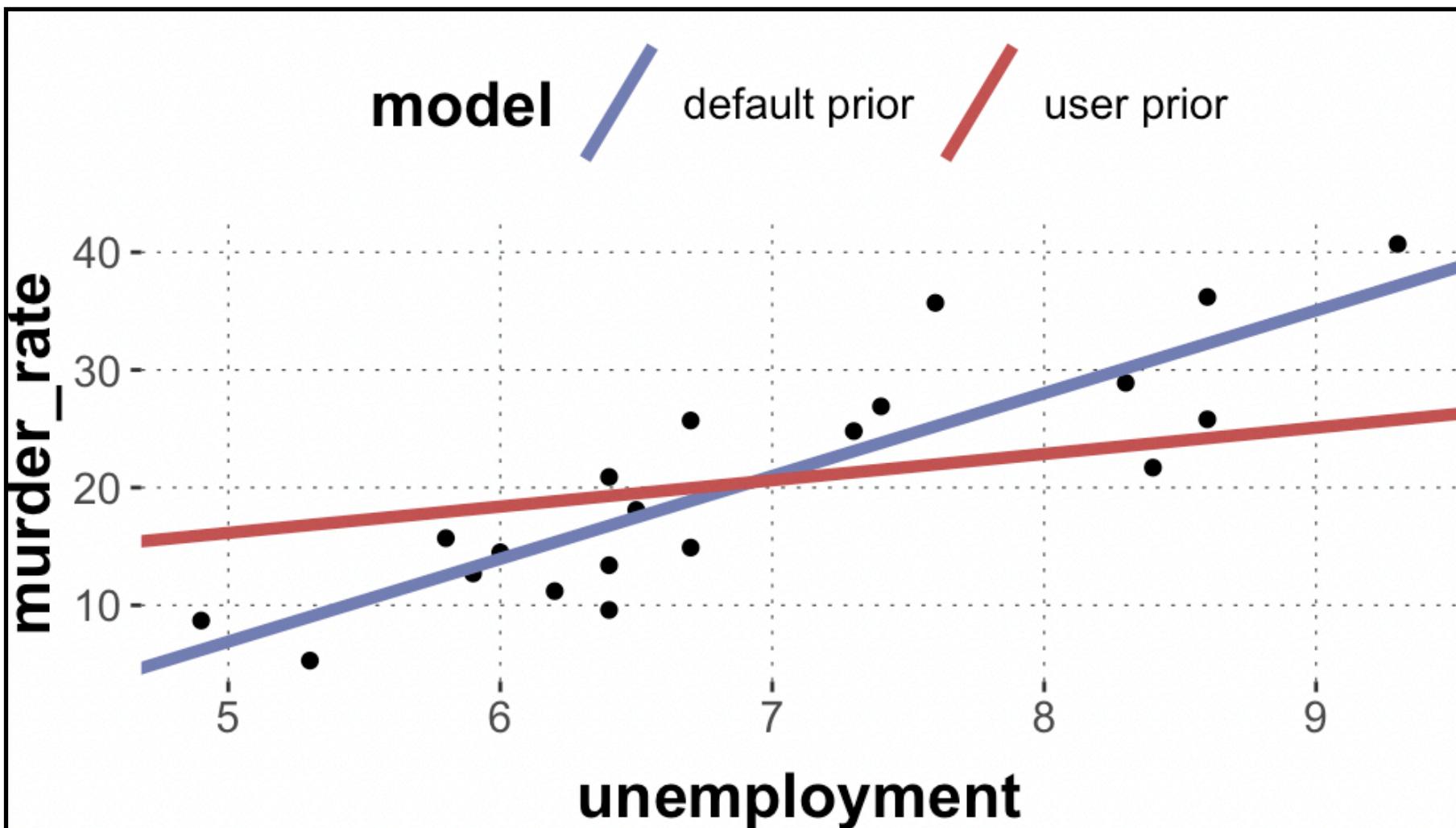
hands-on exploration of a simple linear model in BRMS



Setting priors in brms

Setting priors in BRMS

```
fit_brms_murder <- brm(  
  formula = murder_rate ~ unemployment,  
  data = aida::data_murder  
)  
  
fit_brms_murder_wPrior <- brm(  
  formula = murder_rate ~ unemployment,  
  data = aida::data_murder,  
  prior = prior(normal(0,1), class = "b")  
)
```



```
> brms::prior_summary(fit_brms_murder)  
  prior class     coef group resp dpar nelpar lb ub      source  
  (flat)    b  
  (flat)    b unemployment  
 student_t(3, 19.5, 9.7) Intercept  
 student_t(3, 0, 9.7)   sigma  
 0      default  
  
> brms::prior_summary(fit_brms_murder_wPrior)  
  prior class     coef group resp dpar nelpar lb ub      source  
 normal(0, 1)    b  
 normal(0, 1)    b unemployment  
 student_t(3, 19.5, 9.7) Intercept  
 student_t(3, 0, 9.7)   sigma  
 0      default
```

```
> tidybayes::summarise_draws(fit_brms_murder)  
# A tibble: 5 × 10  
  variable       mean median     sd     mad     q5     q95 rhat ess_bulk ess_tail  
  <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 b_Intercept -28.3  -28.3  7.45  6.99  -40.3  -16.0  1.00  2945.  2304.  
2 b_unemployment 7.04   7.04  1.06  1.01   5.31   8.76  1.00  2929.  2337.  
3 sigma        5.44   5.28  0.994  0.897  4.12   7.26  1.00  2667.  2319.  
4 lprior       -6.07  -6.06  0.0816 0.0655  -6.23  -5.98  1.00  2369.  2121.  
5 lp__         -66.0  -65.6  1.36  1.10  -68.7  -64.5  1.00  1516.  2070.  
  
> tidybayes::summarise_draws(fit_brms_murder_wPrior)  
# A tibble: 5 × 10  
  variable       mean median     sd     mad     q5     q95 rhat ess_bulk ess_tail  
  <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 b_Intercept  4.96   4.84  7.19  7.28  -6.77  17.1  1.00  2299.  2440.  
2 b_unemployment 2.24   2.24  1.01  1.02   0.584  3.89  1.00  2313.  2499.  
3 sigma        8.02   7.81  1.65  1.50   5.67  11.1  1.00  2268.  2218.  
4 lprior       -10.2  -9.72  2.27  2.06  -14.7  -7.56  1.00  2398.  2541.  
5 lp__         -77.8  -77.5  1.28  1.05  -80.2  -76.4  1.00  1881.  2436.
```

Stan code generated by BRMS

- ▶ brms generates Stan code implicitly
- ▶ show it using: `brms::stancode(fitted_model)`
- ▶ prior information increases the log-score
- ▶ priors specified in brms must use Stan-like syntax

```
transformed parameters {  
    real lprior = 0; // prior contributions to the log posterior  
    lprior += normal_lpdf(b | 0, 1);  
    lprior += student_t_lpdf(Intercept | 3, 19.5, 9.7);  
    lprior += student_t_lpdf(sigma | 3, 0, 9.7)  
        - 1 * student_t_lccdf(0 | 3, 0, 9.7);  
}
```

```
// generated with brms 2.18.0  
functions {}  
data {  
    int<lower=1> N; // total number of observations  
    vector[N] Y; // response variable  
    int<lower=1> K; // number of population-level effects  
    matrix[N, K] X; // population-level design matrix  
    int prior_only; // should the likelihood be ignored?  
}  
transformed data {  
    int Kc = K - 1;  
    matrix[N, Kc] Xc; // centered version of X without an intercept  
    vector[Kc] means_X; // column means of X before centering  
    for (i in 2:K) {  
        means_X[i - 1] = mean(X[, i]);  
        Xc[, i - 1] = X[, i] - means_X[i - 1];  
    }  
}  
parameters {  
    vector[Kc] b; // population-level effects  
    real Intercept; // temporary intercept for centered predictors  
    real<lower=0> sigma; // dispersion parameter  
}  
transformed parameters {  
    real lprior = 0; // prior contributions to the log posterior  
    lprior += normal_lpdf(b | 0, 1);  
    lprior += student_t_lpdf(Intercept | 3, 19.5, 9.7);  
    lprior += student_t_lpdf(sigma | 3, 0, 9.7)  
        - 1 * student_t_lccdf(0 | 3, 0, 9.7);  
}  
model {  
    // likelihood including constants  
    if (!prior_only) {  
        target += normal_id_glm_lpdf(Y | Xc, Intercept, b, sigma);  
    }  
    // priors including constants  
    target += lprior;  
}  
generated quantities {  
    // actual population-level intercept  
    real b_Intercept = Intercept - dot_product(means_X, b);  
}
```

demo



assessing and setting priors in BRMS



Prior & posterior predictions

Three pillars of BDA

1. parameter estimation / inference [which parameter values are credible given data and model?]

$$\underbrace{P(\theta | D)}_{\text{posterior}} \propto \underbrace{P(\theta)}_{\text{prior}} \times \underbrace{P(D | \theta)}_{\text{likelihood}}$$

2. predictions [which future data observations are likely given my model?]

a. prior

$$P(D_{\text{pred}}) = \int P(\theta) P(D_{\text{pred}} | \theta) d\theta$$

b. posterior

$$P(D_{\text{pred}} | D_{\text{obs}}) = \int P(\theta | D_{\text{obs}}) P(D_{\text{pred}} | \theta) d\theta$$

3. model comparison [which model of two models is more likely to have generated the data?]

$$\frac{\underbrace{P(M_1 | D)}_{\text{posterior odds}}}{\underbrace{P(M_2 | D)}_{\text{posterior odds}}} = \underbrace{\frac{P(D | M_1)}{P(D | M_2)}}_{\text{Bayes factor}} \frac{\underbrace{P(M_1)}_{\text{prior odds}}}{\underbrace{P(M_2)}_{\text{prior odds}}}$$

Prior and posterior (data) predictions

Monte Carlo sampling

- ▶ fix a model with $P(D | X, \theta)$ and $P(\theta)$
 - latter can be prior or post. conditioned on D'
- ▶ sample from the predictive distribution by:
 - (i) sampling a vector of parameters $\theta^* \sim P(\theta)$
 - (ii) sampling “fake” data D^* from the likelihood function, conditioned on the sampled θ^* (given the relevant predictor values X):
$$D^* \sim P(D | X, \theta^*)$$

- ▶ Monte Carlo sampling:
 - by taking many samples and “aggregating”, we approximate the integrals
 - “aggregating” means that (i) we usually don’t care for just one sample, but the distributional information in a lot of samples, and (ii) we might want to focus on particular aspects of each sampled “fake” data (e.g., a particular summary statistic)

prior predictive

$$P(D_{\text{pred}}) = \int P(\theta) P(D_{\text{pred}} | \theta) d\theta$$

posterior predictive

$$P(D_{\text{pred}} | D_{\text{obs}}) = \int P(\theta | D_{\text{obs}}) P(D_{\text{pred}} | \theta) d\theta$$

predicted linear predictor

- ▶ fix a linear model with $P(\theta)$
 - $P(\theta)$ can be prior or posterior
- ▶ sample a linear predictor $\mu^* \sim P(\mu^* | X, \theta)$
 - X is a matrix of independent variables

Example: World-temperature data

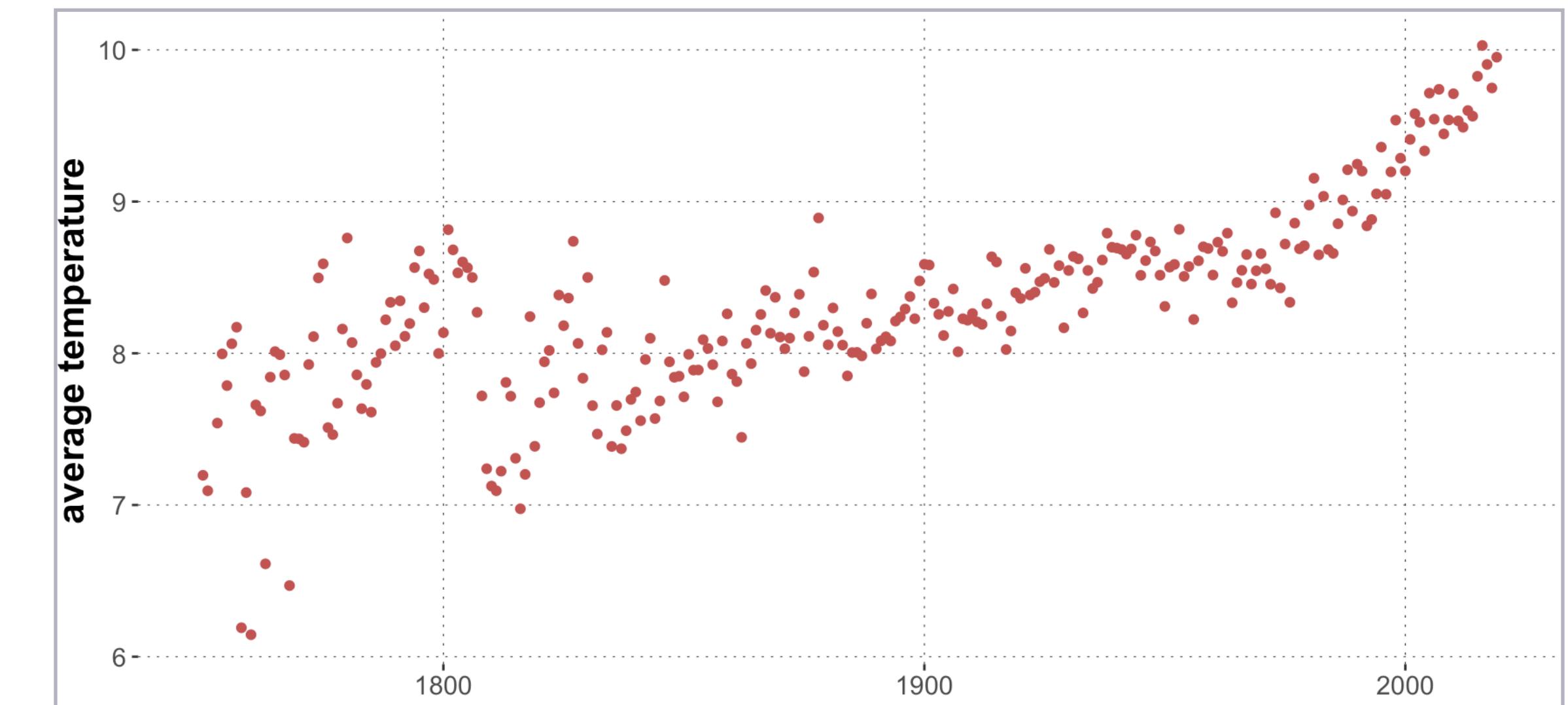
data, model and posterior predictions

- ▶ data:
 - average world temperature 1750-2019

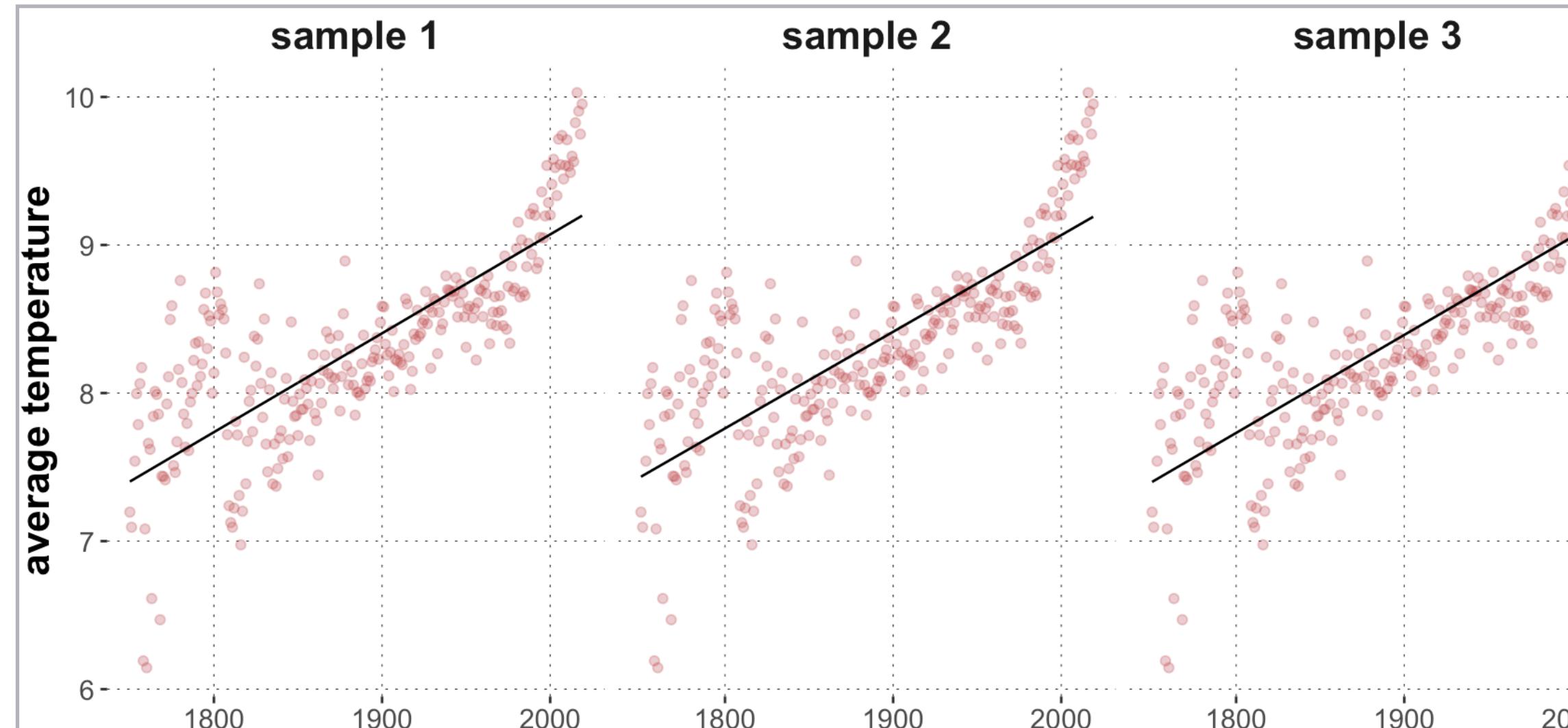
- ▶ model:

```
fit_worldTemp <- brm(  
  avg_temp ~ year,  
  data = aida::data_WorldTemp,  
  prior = prior(student_t(1,0,5), class = "b")  
)
```

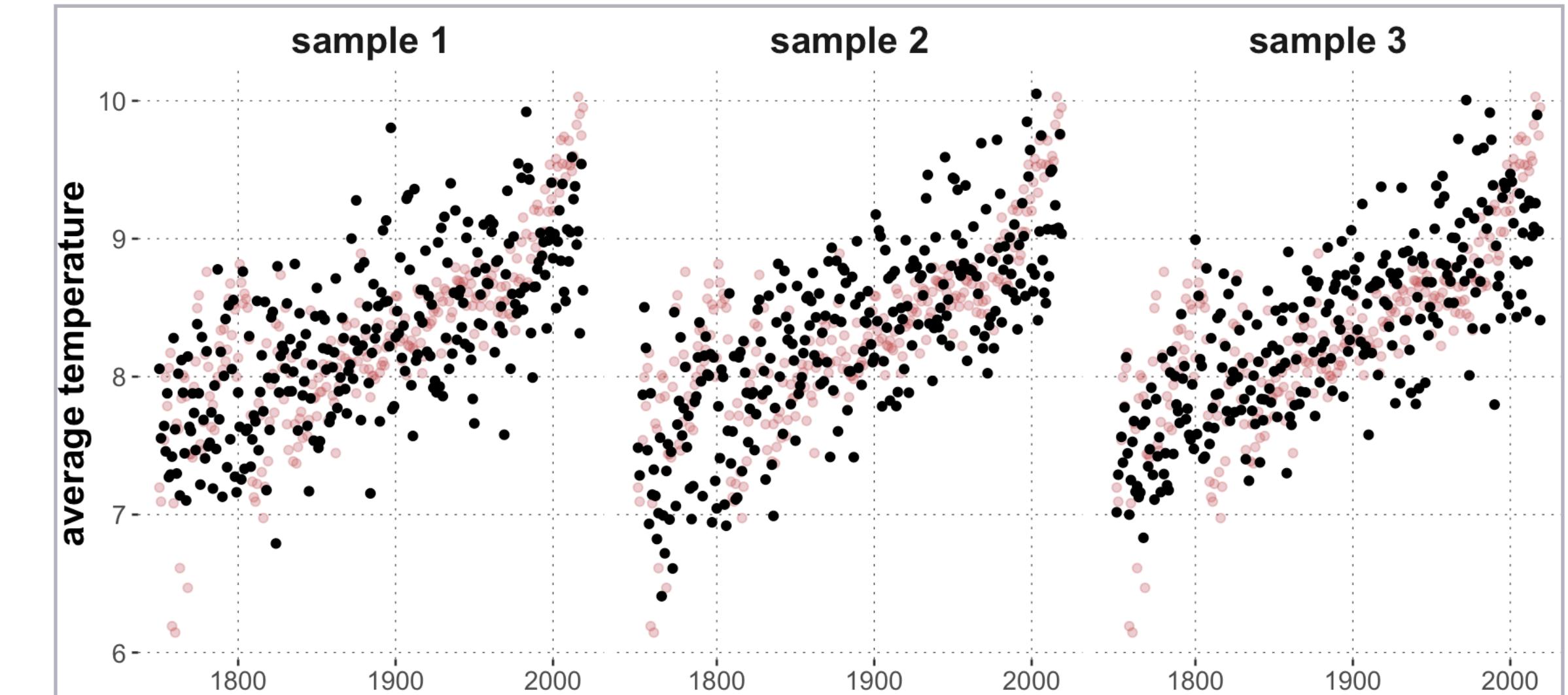
Data



Posterior samples for the central tendency



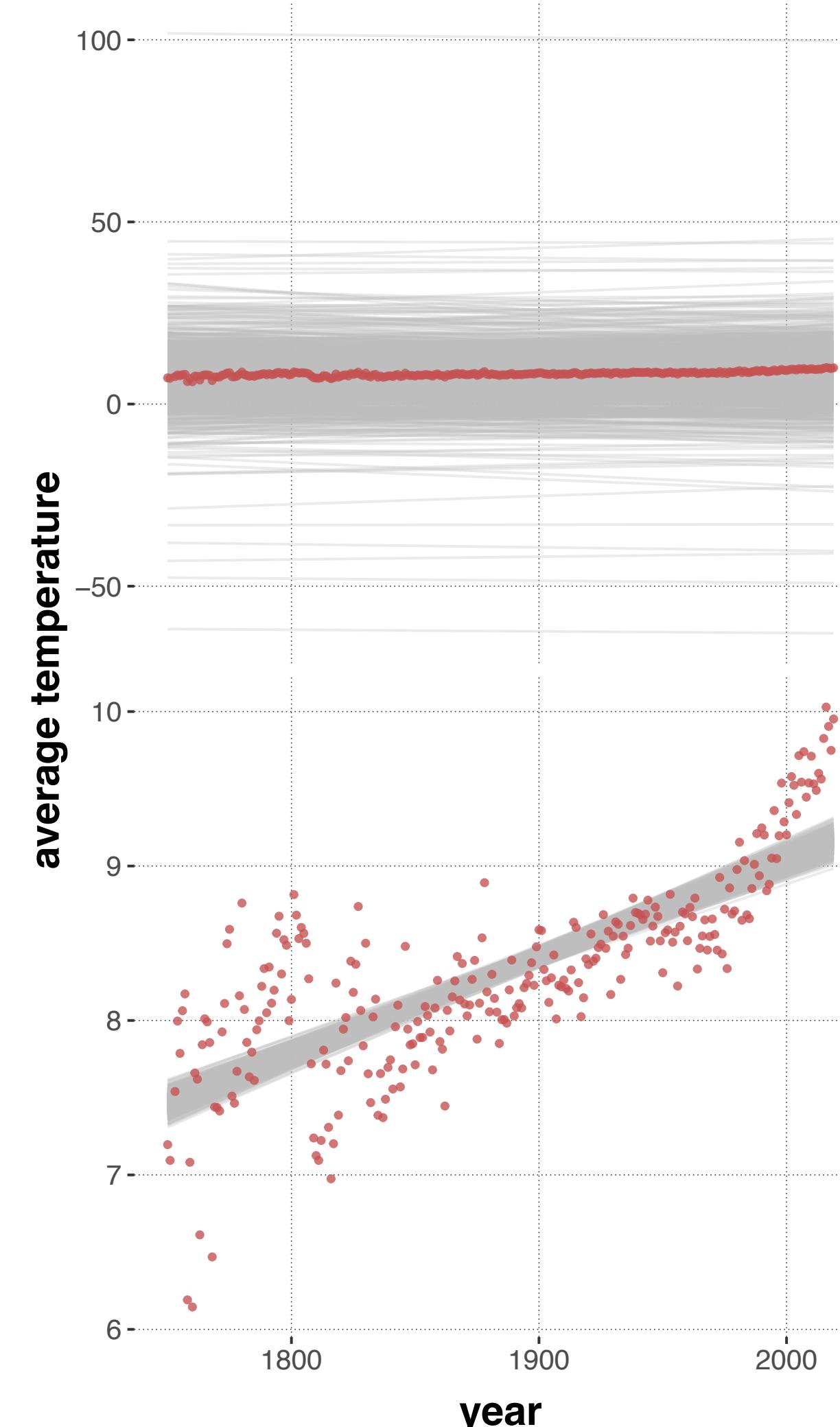
Samples from the posterior predictive (data)



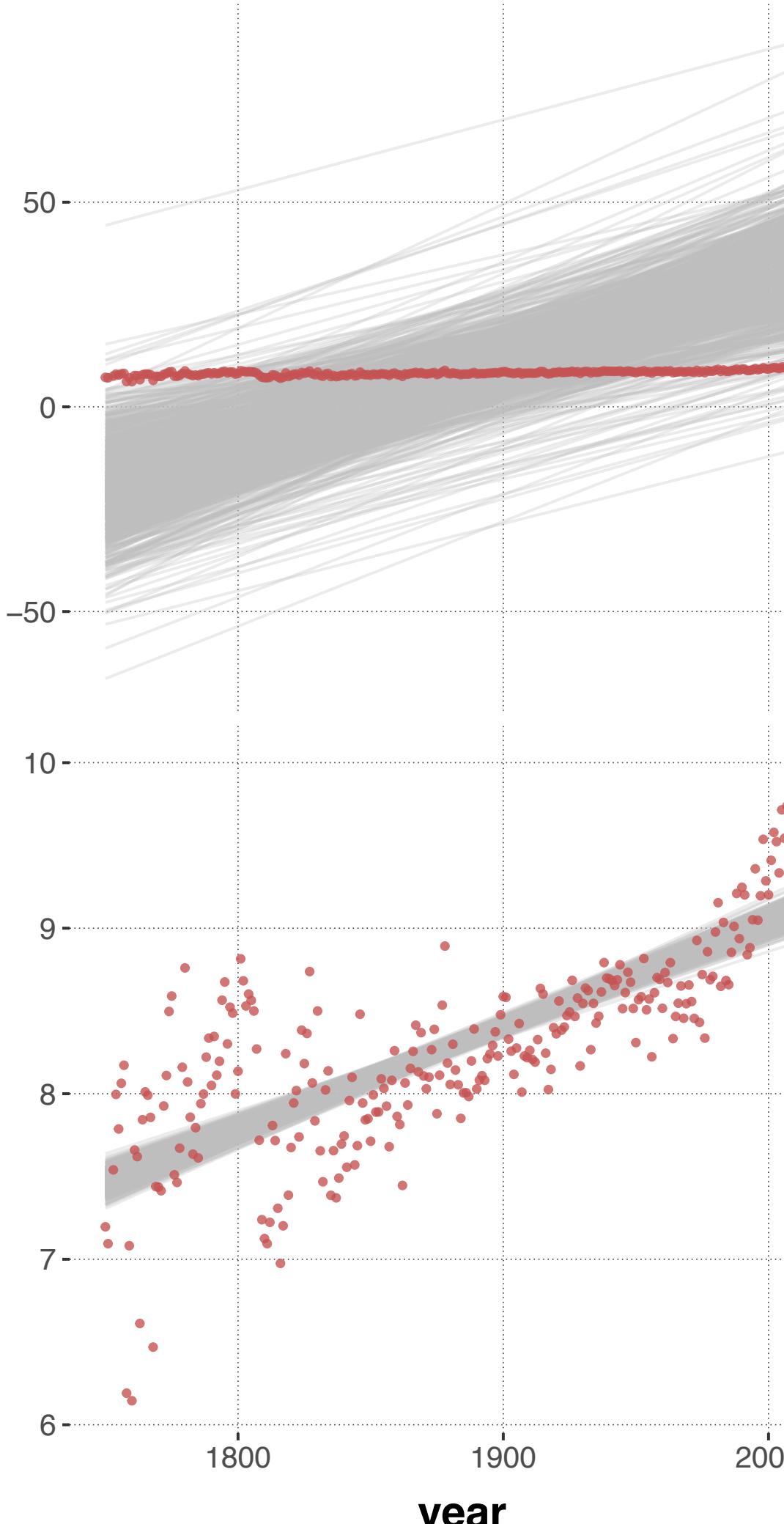
Exploring prior and posterior prediction

Prior & posterior samples of linear predictor value (μ)

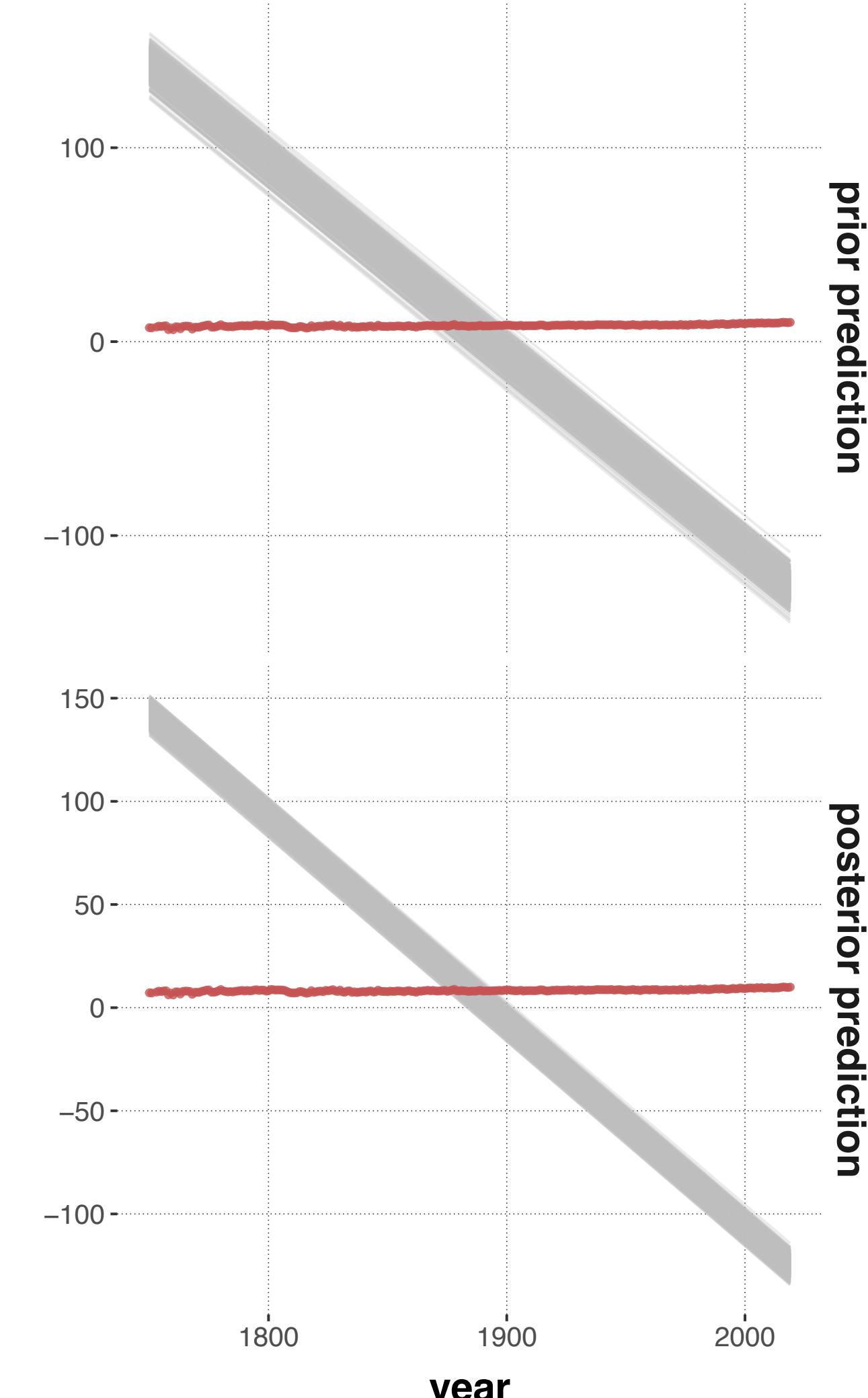
$\beta_{\text{year}} \sim \text{Normal}(0, 0.02)$



$\beta_{\text{year}} \sim \text{Normal}(0.2, 0.05)$



$\beta_{\text{year}} \sim \text{Normal}(-1, 0.005)$



demo



assessing predictive samples in BRMS