

Bayesian regression modeling: Theory & practice

Part 2: Priors, prior and posterior predictives, and categorical predictors

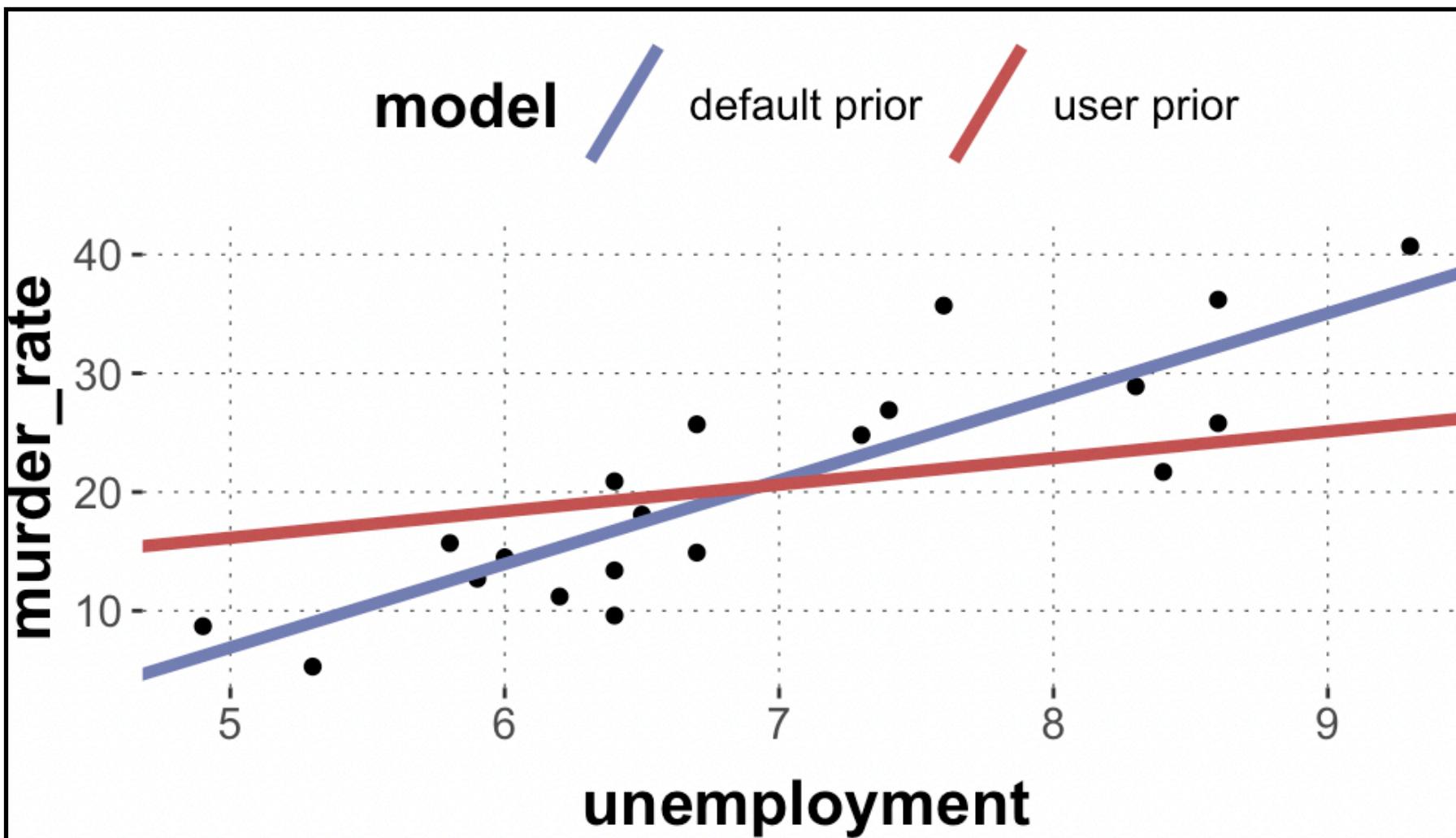
Michael Franke



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

```
> 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);  
}
```



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 predictions

Monte Carlo sampling

- ▶ fix a model with $P(D | \theta)$ and $P(\theta)$
 - latter can be prior or posterior
- ▶ a sample from the predictive distribution is obtained 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^* :
$$D^* \sim P(D | \theta^*)$$
- ▶ 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 samples “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

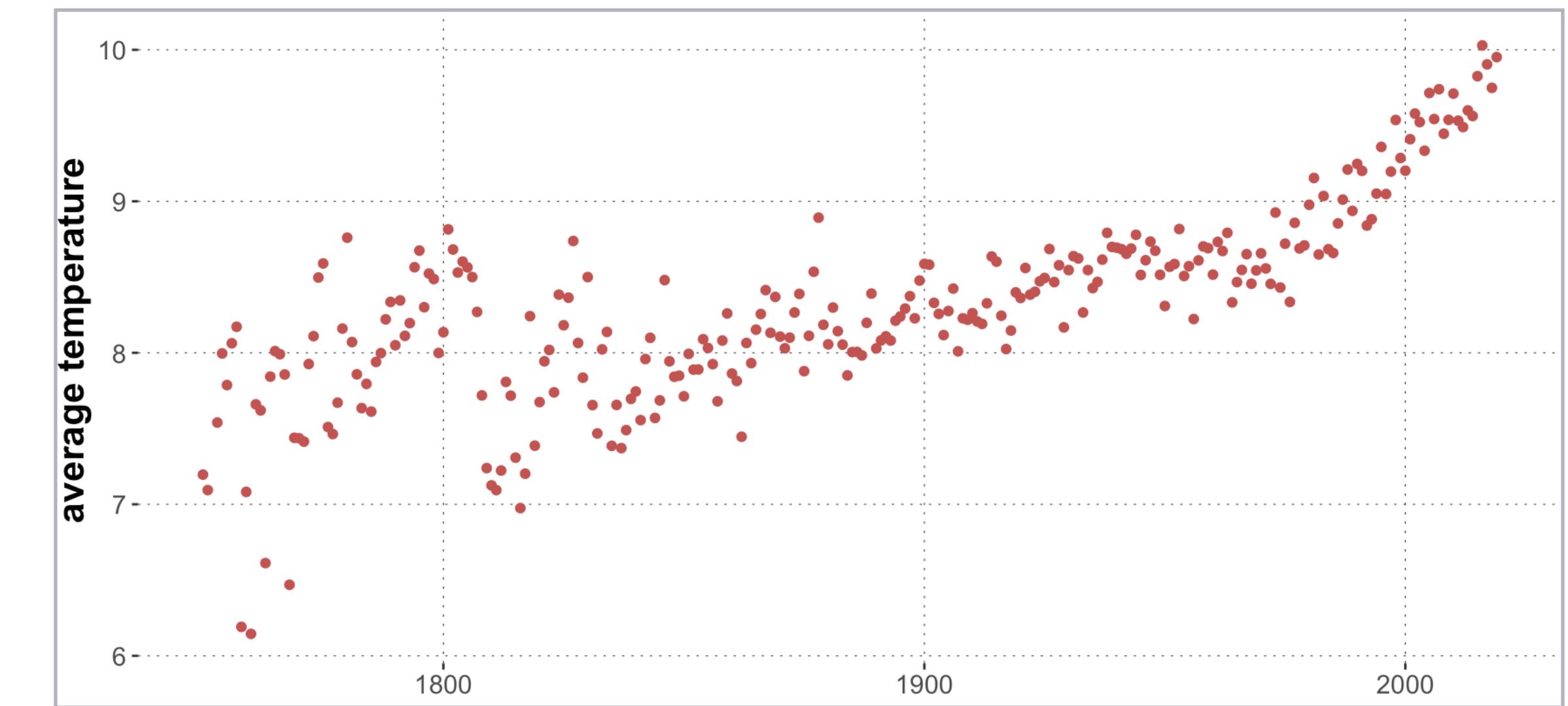
data, model and posterior predictions

- ▶ data:
 - average world temperature 1750-2019

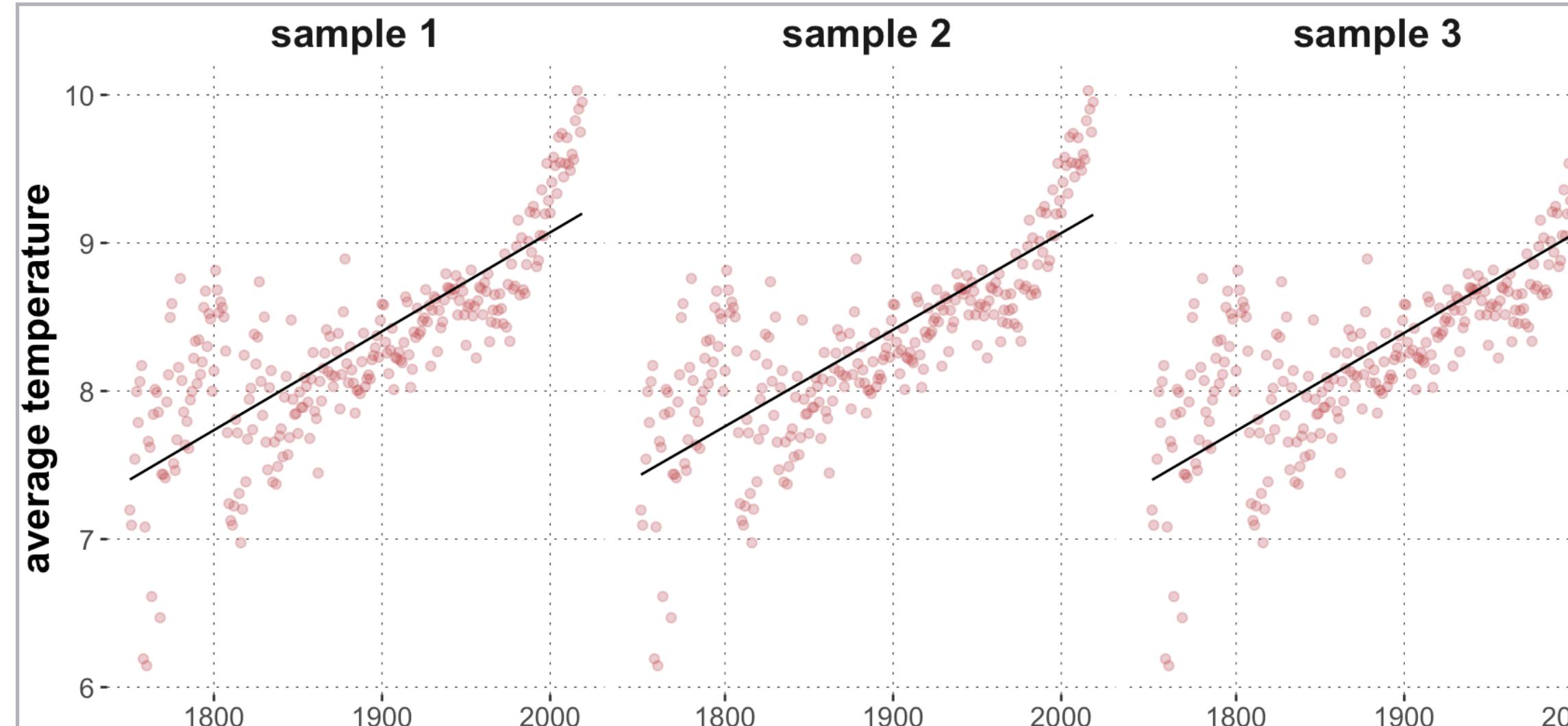
- ▶ model:

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

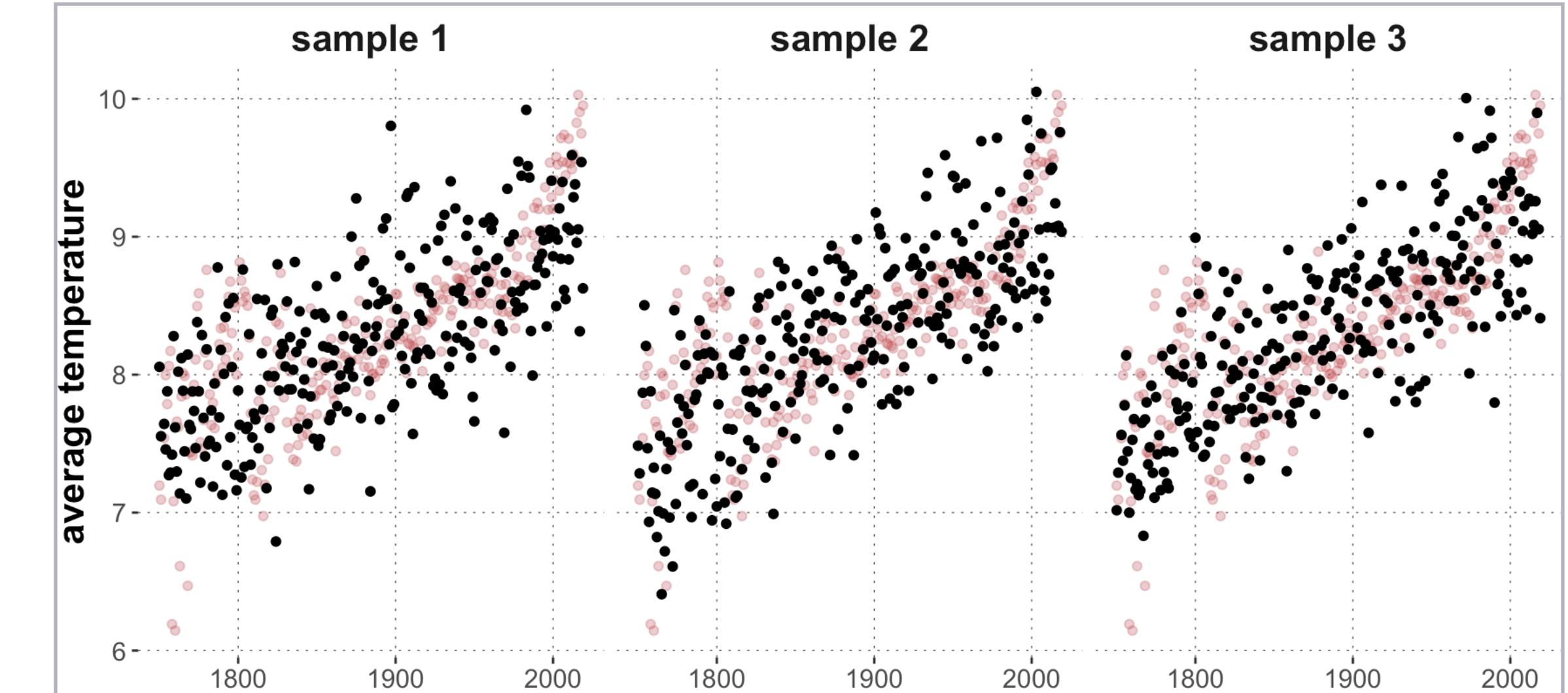
Data



Samples from the posterior predictive



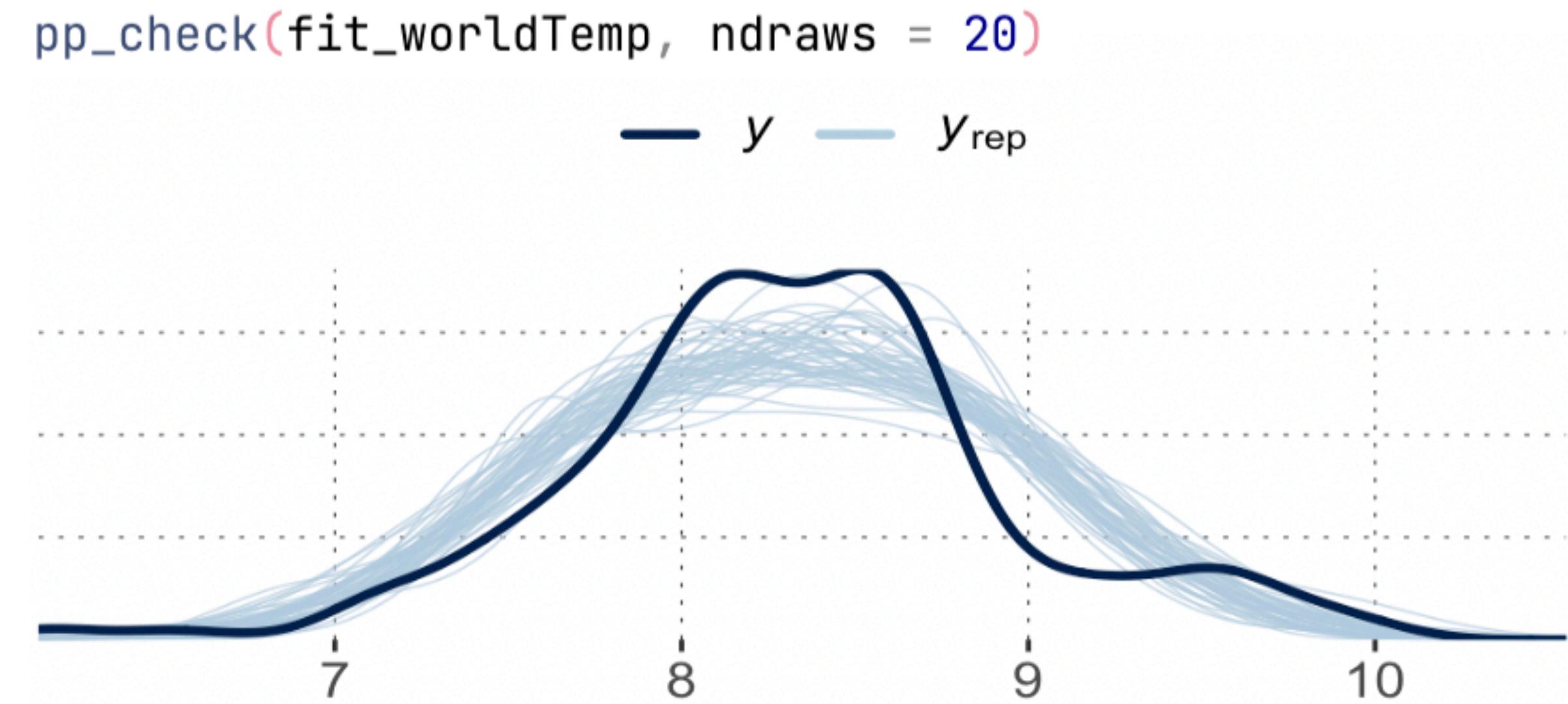
Samples from the posterior predictive



Visual posterior predictive checks

for world-temperature data

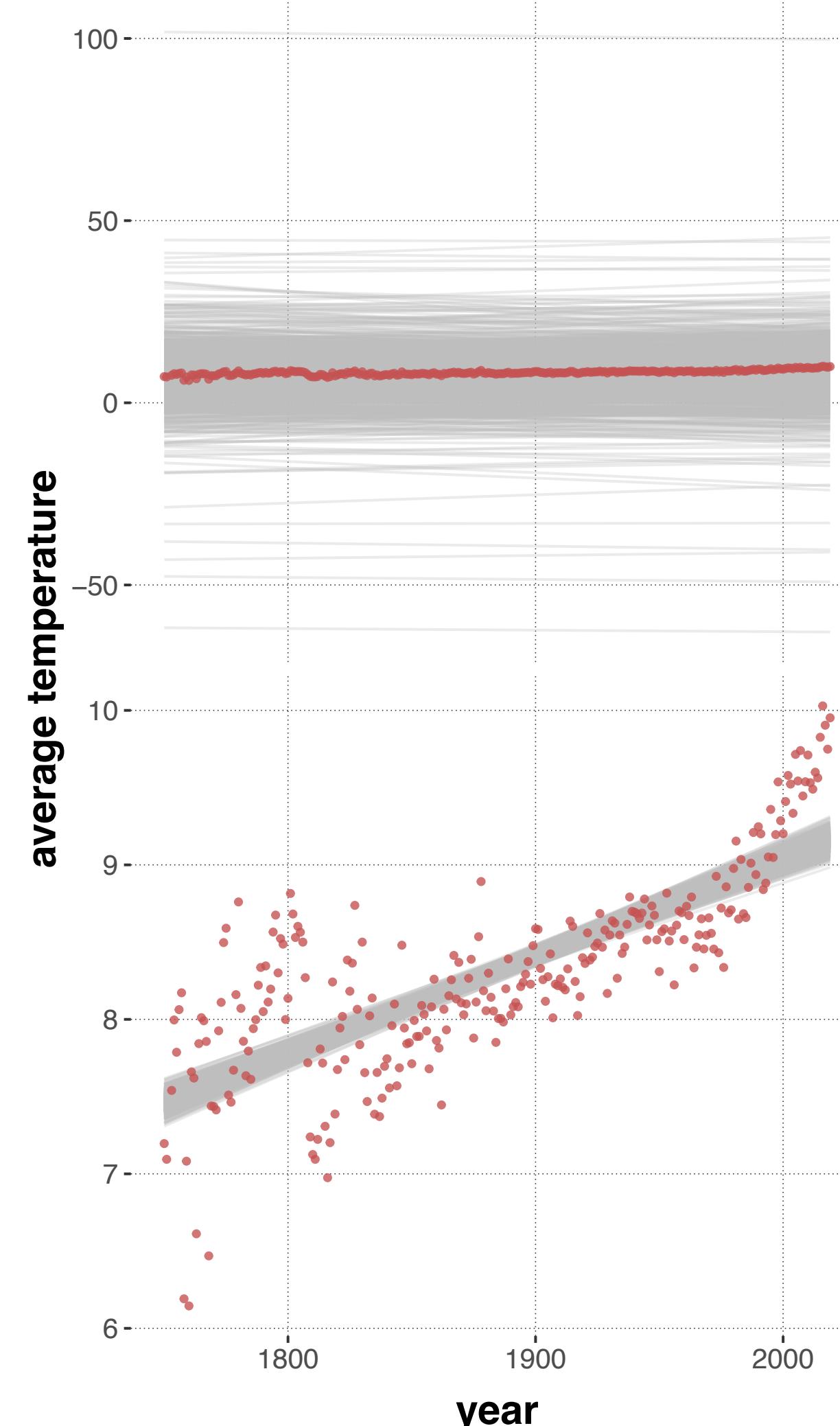
- ▶ black line:
 - distribution of observed temperature
- ▶ each of the 50 blue lines:
 - distribution of temperatures predicted for same years given a sample from the posterior



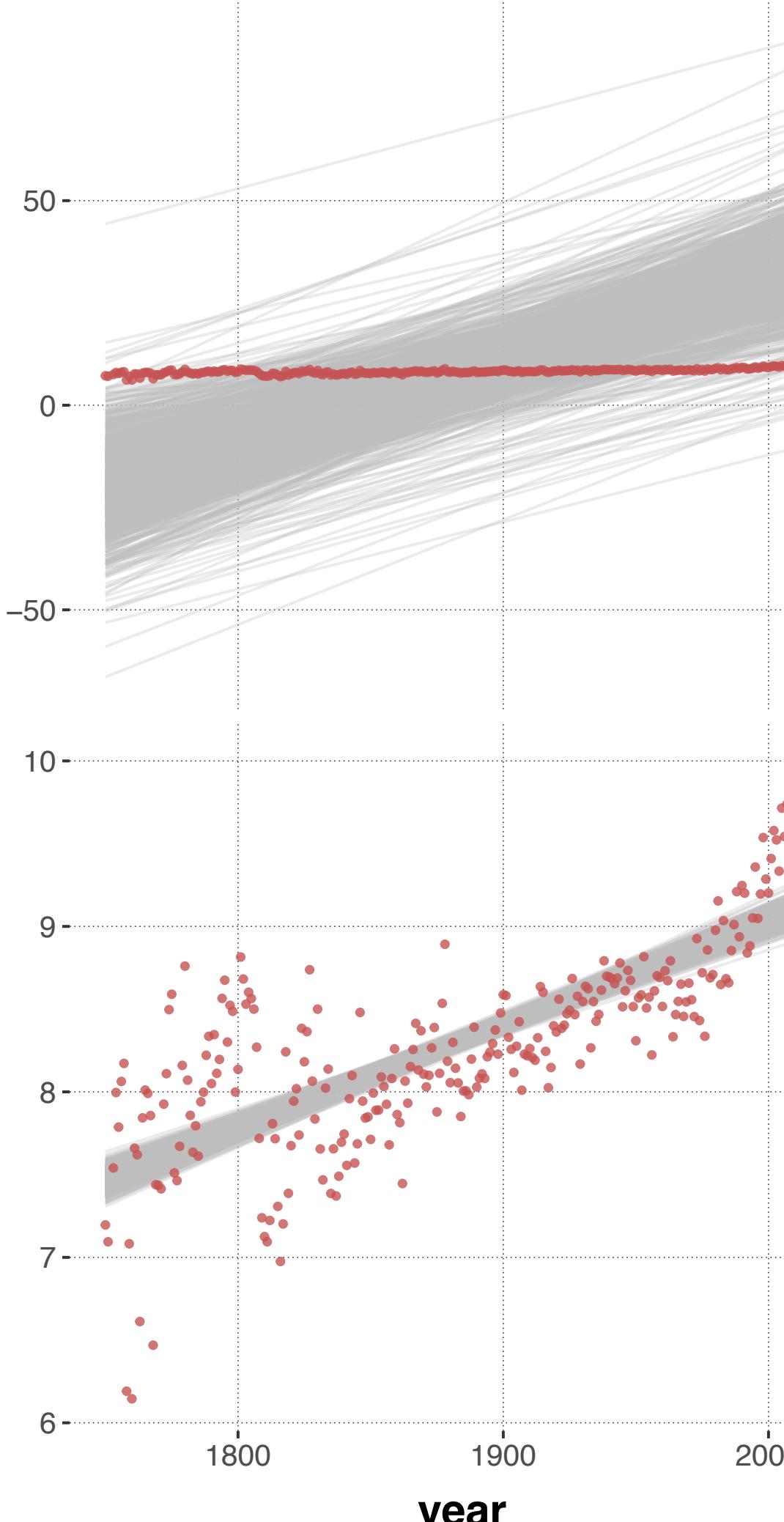
Exploring prior and posterior prediction

Prior & posterior samples of linear predictor value (μ)

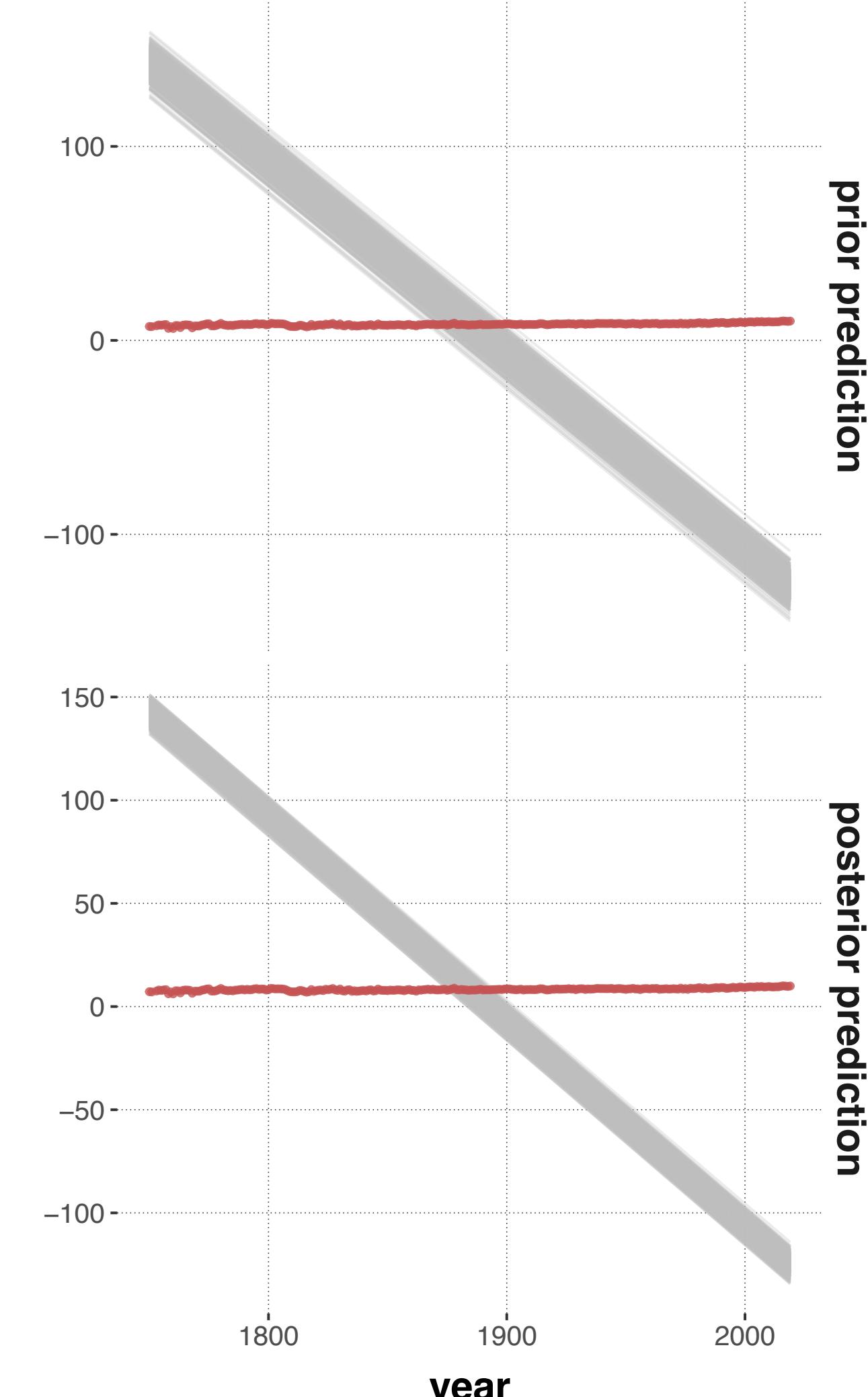
$\beta_{\text{year}} \sim \text{Normal}(0, 0.02)$



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



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



Bayesian predictive p -values

a generalization

- ▶ fix a model with $P(D | \theta)$ and $P(\theta)$
 - latter can be prior or posterior
 - gives prior / posterior predictive p -values
- ▶ $P_M(D)$ is the predictive distribution for model M
- ▶ Bayesian predictive p -value for observed data d_{obs} :

$$p(d_{\text{obs}}) = P_M(D \in \{d | P_M(d) \leq P_M(d_{\text{obs}})\})$$

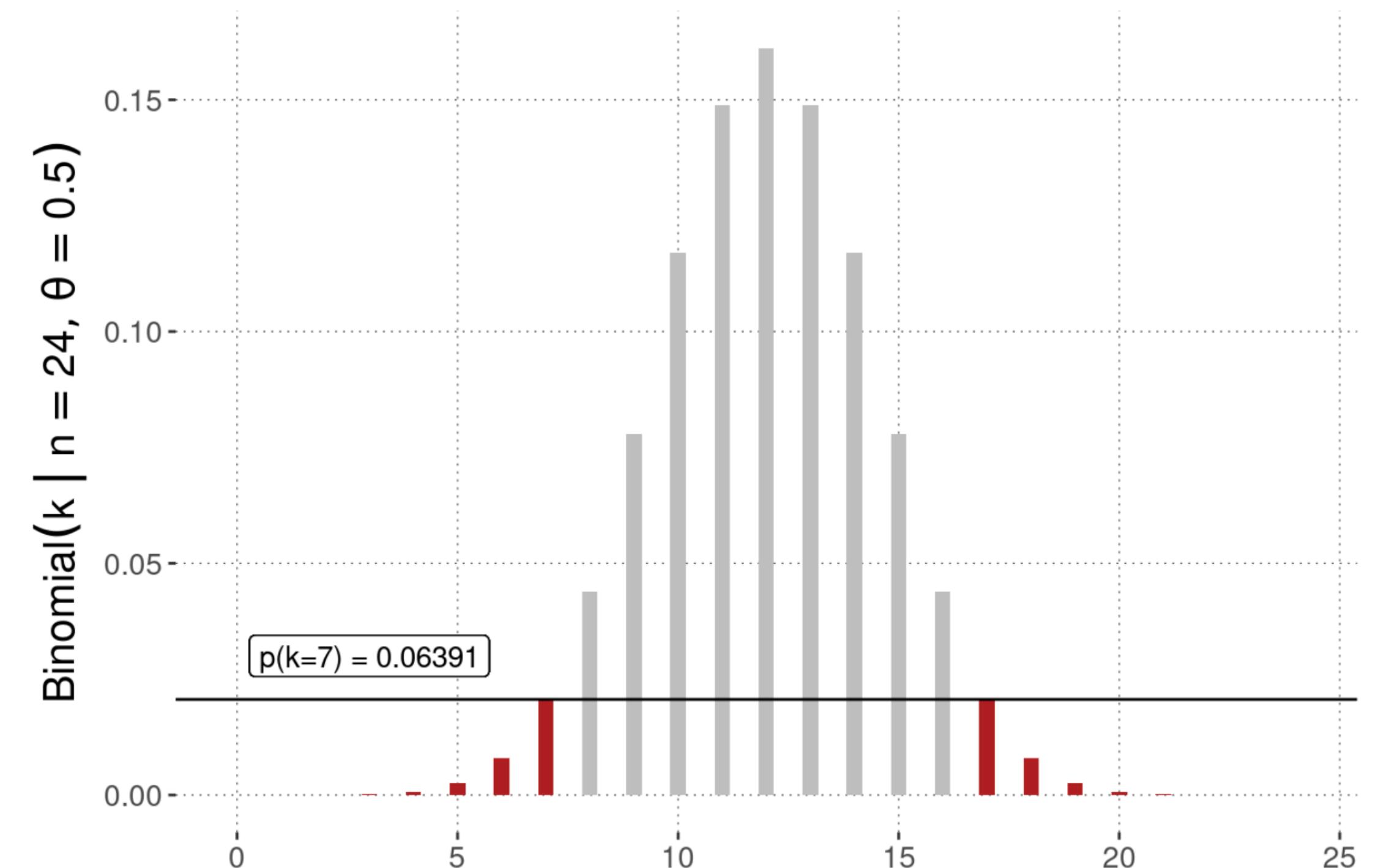
- ▶ approximated by sampling:

$$p(d_{\text{obs}}) \approx \frac{1}{n} \sum_{i=1}^n [P_M(d_i) \leq P_M(d_{\text{obs}})]$$

where $d_i \sim P_M(D)$ is a sample from the predictive distribution

Recap: frequentist p -values

$$p(D_{\text{obs}}) = P\left(T|H_0 \sum_{H_0,a} t(D_{\text{obs}})\right)$$



read more [here](#)