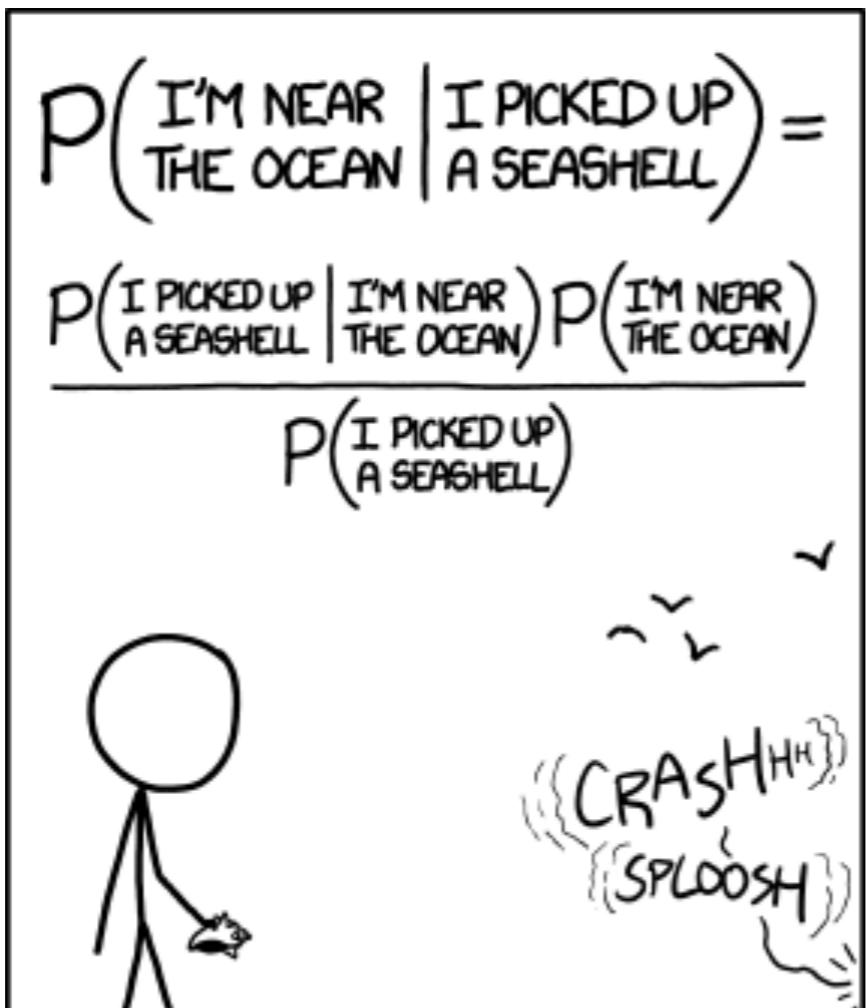


Bayesian data analysis 3



STATISTICALLY SPEAKING, IF YOU PICK UP A SEASHELL AND DON'T HOLD IT TO YOUR EAR, YOU CAN PROBABLY HEAR THE OCEAN.

MODIFIED BAYES' THEOREM:

$$P(H|x) = P(H) \times \left(1 + P(C) \times \left(\frac{P(x|H)}{P(x)} - 1 \right) \right)$$

H: HYPOTHESIS

x: OBSERVATION

P(H): PRIOR PROBABILITY THAT H IS TRUE

P(x): PRIOR PROBABILITY OF OBSERVING x

P(C): PROBABILITY THAT YOU'RE USING BAYESIAN STATISTICS CORRECTLY

Logistics

Application section

Thursday, March 5th, 4:30pm - 5:20pm in 160-322

Applied Bayesian Data Analysis



Final presentations

24 responses

Accepting responses

[Summary](#) [Question](#) [Individual](#)

When/how will you present?

24 responses

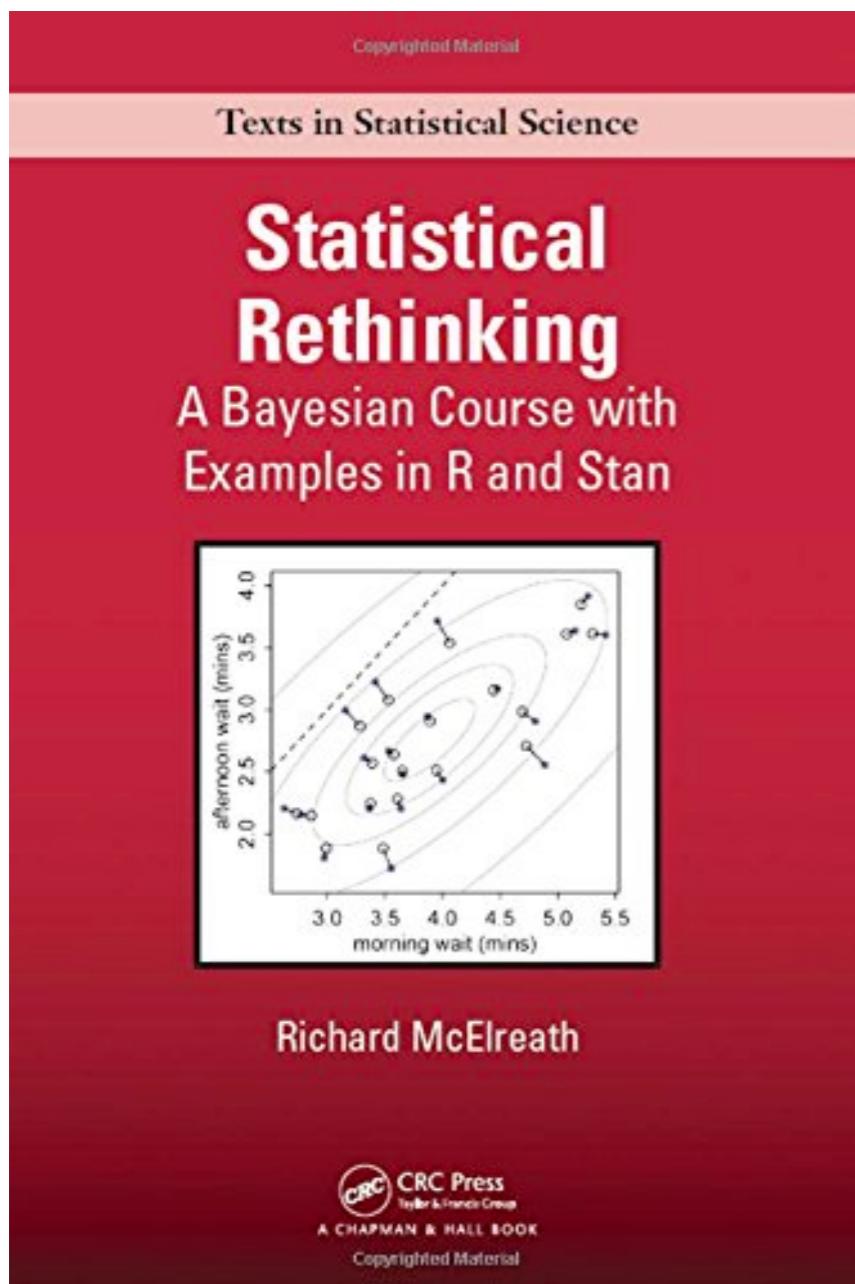
Method	Percentage
On March 18th (Final presentations day)	95.8%
On March 13th (Final class)	2.5%
On March 16th at 3pm (Stats instructors meeting)	1.7%
I will record the presentation and submit a video.	0.0%

On March 18th (Final presentations day)
On March 13th (Final class)
On March 16th at 3pm (Stats instructors meeting)
I will record the presentation and submit a video.

<https://tinyurl.com/psych252presentation>

Things that came up

Statistical rethinking



Pinned Tweet

Solomon Kurz
@SolomonKurz

The 1.1.0 update to my project translating Statistical Rethinking into [#brms](#) and the [#tidyverse](#) is up!

Here's the new link: bookdown.org/content/3890

Here's the new doi: doi.org/10.5281/zenodo...

1/7

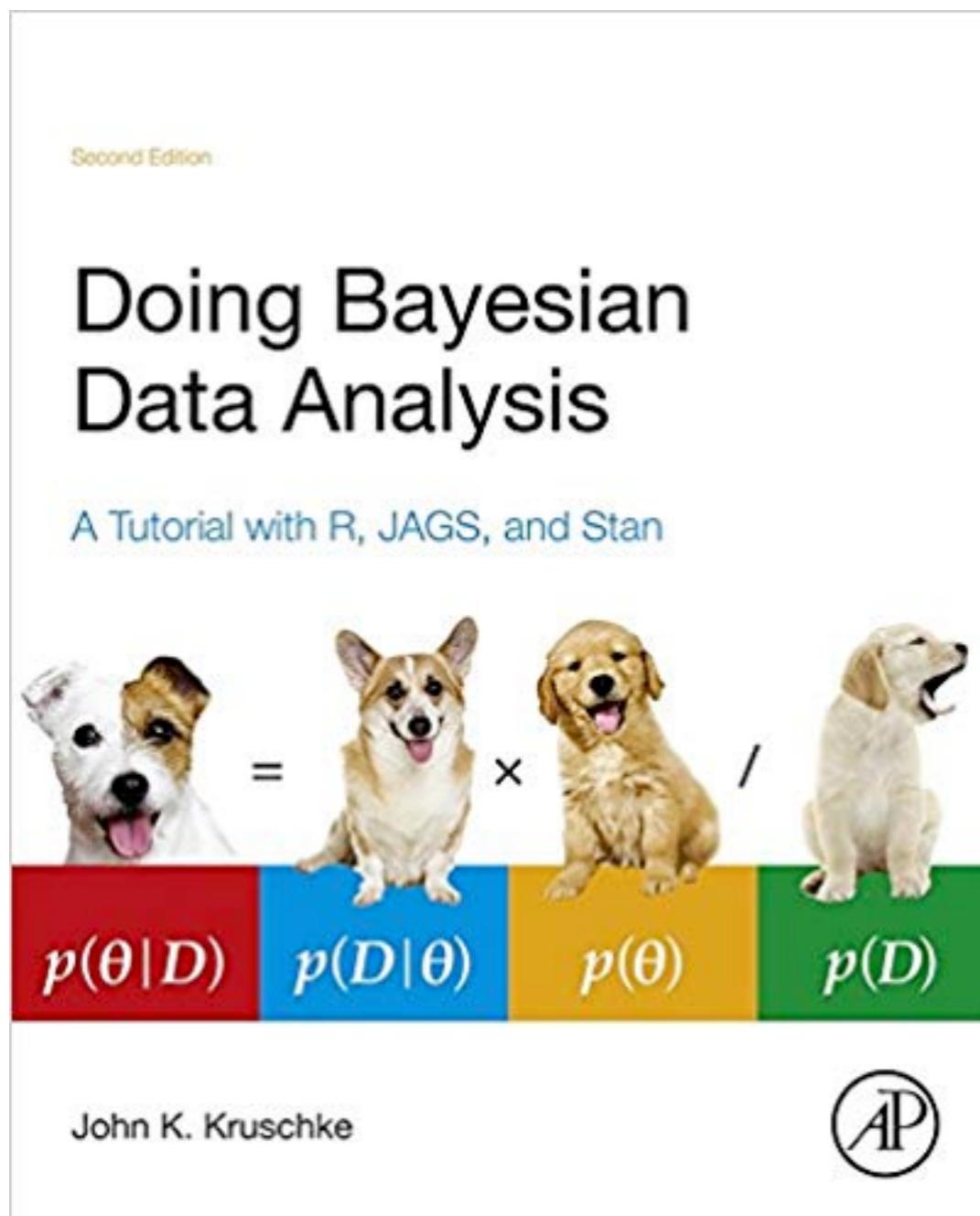


7:37 AM · Mar 2, 2020 [Twitter Web App](#)

38 Retweets 182 Likes

<https://bookdown.org/content/3890/>

Doing Bayesian Data Analysis



Doing Bayesian Data Analysis in brms and the
tidyverse

version 0.0.5

A Solomon Kurz

2019-12-19



https://bookdown.org/ajkurz/DBDA_recoded/

Plan for today

- Building Bayesian models with `brms`
- Model evaluation
- Testing hypotheses
- Model comparison
- Inference evaluation: Did things work out?
- Reporting results
- Steps for fitting a Bayesian model in `brms`
 - Sleep data
 - Titanic data

Plan for today

- **Building Bayesian models with `brms`**
- Model evaluation
- Testing hypotheses
- Model comparison
- Inference evaluation: Did things work out?
- Reporting results
- Steps for fitting a Bayesian model in `brms`
 - Sleep data
 - Titanic data

Main theme today



Software packages



Bayesian regression
modeling with Stan

```
library("brms")
```

- very powerful package that makes it easy to run Bayesian regression models
- we specify models using the same syntax we've already learned based on **lm()**, **glm()**, and **lmer()**
- brms turns this into Stan code and fits the model
- we can then use tidybayes to investigate the posterior

Software packages



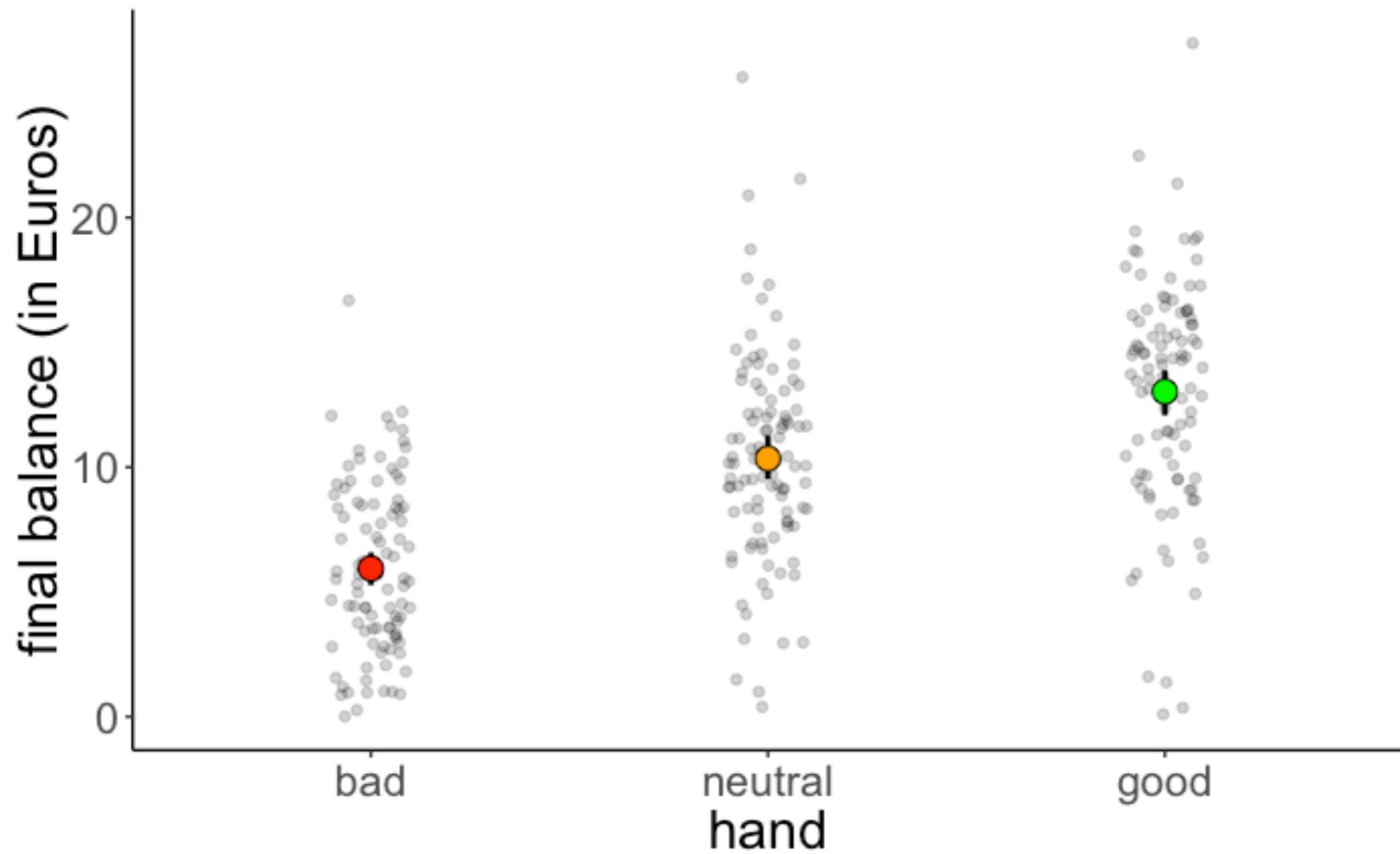
The logo consists of a large red circle containing a white 'S' shape. From behind the circle, several thin, light-colored lines radiate outwards in various directions, resembling energy or data flow.

Stan

Stan® is a state-of-the-art platform for statistical modeling and high-performance statistical computation. Thousands of users rely on Stan for statistical modeling, data analysis, and prediction in the social, biological, and physical sciences, engineering, and business.

<https://mc-stan.org/>

Poker data



Using lm()

```
1 fit.lm = lm(formula = balance ~ 1 + hand,  
2               data = df.poker)  
3  
4 fit.lm %>% summary()
```

```
Call:  
lm(formula = balance ~ 1 + hand, data = df.poker)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-12.9264 -2.5902 -0.0115  2.6573 15.2834  
  
Coefficients:  
              Estimate Std. Error t value Pr(>|t|)  
(Intercept) 5.9415    0.4111 14.451 < 2e-16 ***  
handneutral 4.4051    0.5815  7.576 4.55e-13 ***  
handgood    7.0849    0.5815 12.185 < 2e-16 ***  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 4.111 on 297 degrees of freedom  
Multiple R-squared:  0.3377, Adjusted R-squared:  0.3332  
F-statistic: 75.7 on 2 and 297 DF,  p-value: < 2.2e-16
```

Using brm()

COOL!

```
1 fit.brn = brm(formula = balance ~ 1 + hand,  
2                  data = df.poker)  
3  
4 fit.brn %>% summary()
```

```
Family: gaussian  
Links: mu = identity; sigma = identity  
Formula: balance ~ 1 + hand  
Data: df.poker (Number of observations: 300)  
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	5.93	0.41	5.12	6.72	1.00	2986	2744		
handneutral	4.41	0.58	3.30	5.55	1.00	3497	2903		
handgood	7.10	0.58	5.99	8.29	1.00	3545	2932		

Family Specific Parameters:

	Estimate	Est.Error	l-95%	CI	u-95%	CI	Rhat	Bulk_ESS	Tail_ESS
sigma	4.12	0.17	3.81	4.46	1.00	3650	2921		

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).

Comparison between lm() and brm()

lm()

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.9415	0.4111	14.451	< 2e-16 ***
handneutral	4.4051	0.5815	7.576	4.55e-13 ***
handgood	7.0849	0.5815	12.185	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

brm()

Population-Level Effects:

	Estimate	Est.Error	l-95%	CI	u-95%	CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	5.93	0.41	5.12		6.72	1.00	1	2986	2744
handneutral	4.41	0.58	3.30		5.55	1.00	1	3497	2903
handgood	7.10	0.58	5.99		8.29	1.00	1	3545	2932

**almost identical
results!**

What about the priors?

```
1 fit.brm = brm(formula = balance ~ 1 + hand,  
2                  data = df.poker)
```

By default, brms uses weakly informative priors for the model parameters.

There are quite a few other defaults, let's take a look under the hood ...

"Full" specification of the model

```
1 fit.brm2 = brm(  
2   formula = balance ~ 1 + hand,  
3   family = "gaussian", ← likelihood  
4   data = df.poker,  
5   prior = c(  
6     prior(normal(0, 10), class = "b", coef = "handgood"),  
7     prior(normal(0, 10), class = "b", coef = "handneutral"),  
8     prior(student_t(3, 3, 10), class = "Intercept"),  
9     prior(student_t(3, 0, 10), class = "sigma") ← priors  
10 ),  
11   inits = list(  
12     list(Intercept = 0, sigma = 1, handgood = 5, handneutral = 5),  
13     list(Intercept = -5, sigma = 3, handgood = 2, handneutral = 2),  
14     list(Intercept = 2, sigma = 1, handgood = -1, handneutral = 1),  
15     list(Intercept = 1, sigma = 2, handgood = 2, handneutral = -2)  
16   ),  
17   iter = 4000, ← how many runs in the inference chain  
18   warmup = 1000, ← how long for the warmup  
19   chains = 4, ← how many chains  
20   file = "cache/brm2", ← save the model result  
21   seed = 1 )  
22 )
```

make reproducible

likelihood

priors

initialization

how many runs in the inference chain

how long for the warmup

how many chains

fitting Bayesian models takes some time, so storing results is key

Turned into Stan code

```
// generated with brms 2.7.0
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, K - 1] Xc; // centered version of X
  vector[K - 1] 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 temp_Intercept; // temporary intercept
  real<lower=0> sigma; // residual SD
}
transformed parameters {
}
model {
  vector[N] mu = temp_Intercept + Xc * b;
  // priors including all constants
  target += normal_lpdf(b[1] | 0, 10);
  target += normal_lpdf(b[2] | 0, 10);
  target += student_t_lpdf(temp_Intercept | 3, 3, 10);
  target += student_t_lpdf(sigma | 3, 0, 10)
    - 1 * student_t_lccdf(0 | 3, 0, 10);
  // likelihood including all constants
  if (!prior_only) {
    target += normal_lpdf(Y | mu, sigma);
  }
}
generated quantities {
  // actual population-level intercept
  real b_Intercept = temp_Intercept - dot_product(means_X, b);
}
```

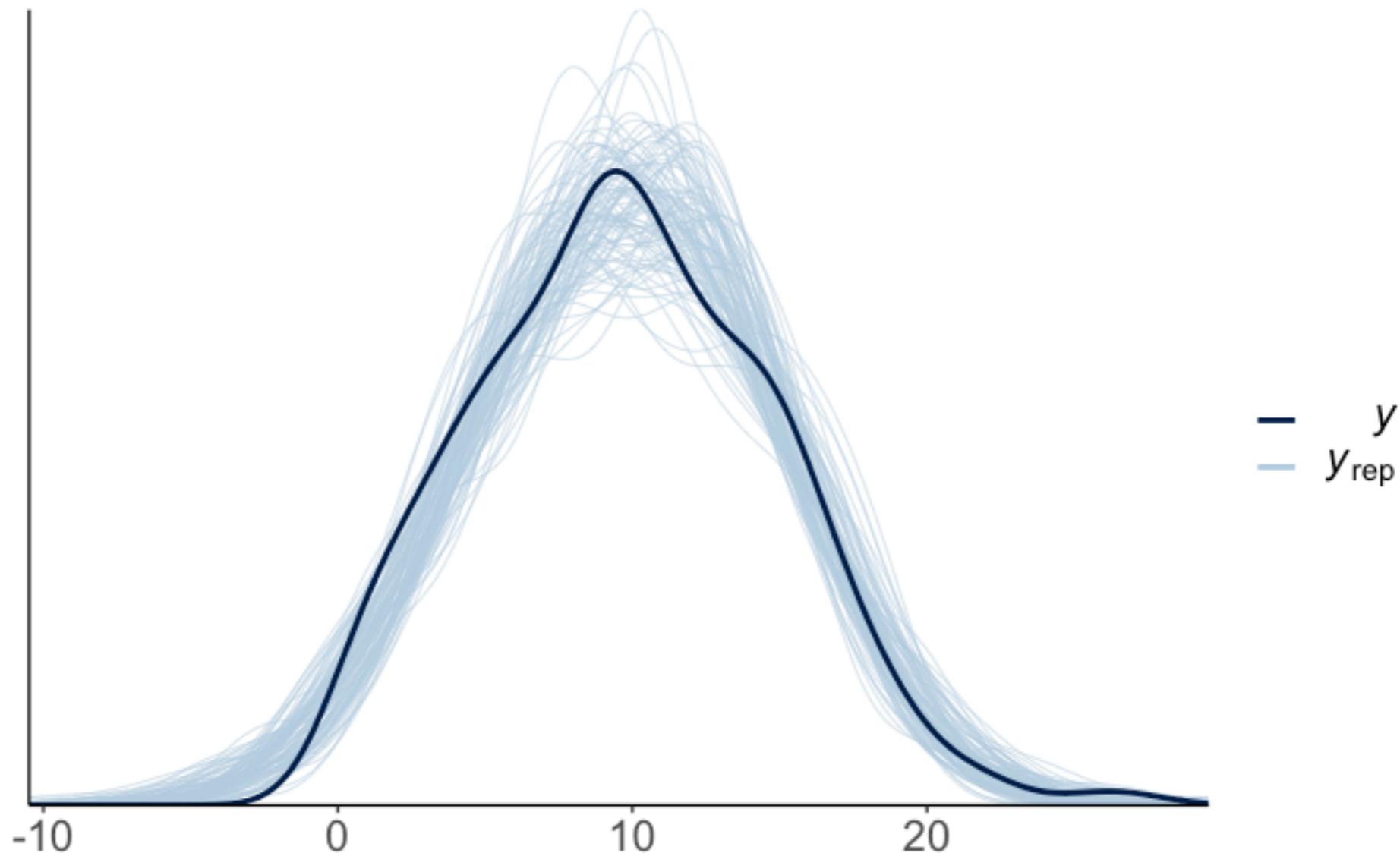
- probabilistic programming language
- flexible construction of Bayesian models
- ports have been written for R, Python, Julia, ...
- implements a fast inference algorithm

Plan for today

- Building Bayesian models with `brms`
- **Model evaluation**
- Testing hypotheses
- Model comparison
- Inference evaluation: Did things work out?
- Reporting results
- Steps for fitting a Bayesian model in `brms`
 - Sleep data
 - Titanic data

Posterior predictive check

```
pp_check(fit.brm, nsamples = 100)
```



The model accurately captures the distribution of the response variable

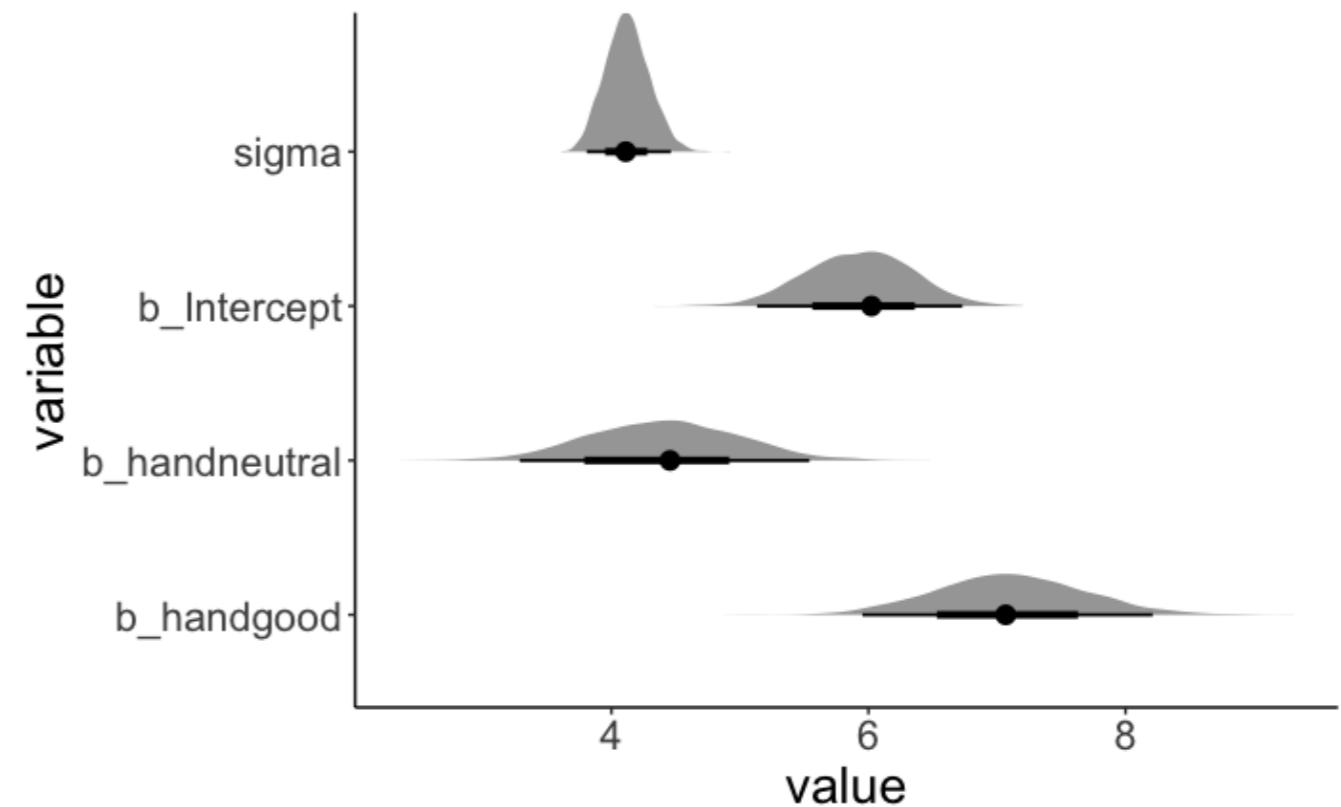
Summary of the posterior

```
1 fit.brml %>%
2   posterior_samples() %>%
3   select(-lp_) %>%
4   pivot_longer(cols = everything(),
5                 names_to = "variable",
6                 values_to = "value") %>%
7   ggplot(data = .,
8         mapping = aes(y = variable,
9                         x = value)) +
10  geom_halfeyeh(fun.data = mode_hdih)
```

from the
"tidybayes"
package

b_Intercept	b_handneutral	b_handgood	sigma
5.97	4.77	6.65	4.36
5.69	4.73	6.92	3.92
5.59	5.15	7.49	4.08
6.00	4.29	6.47	4.32
6.03	4.10	6.42	4.33

:



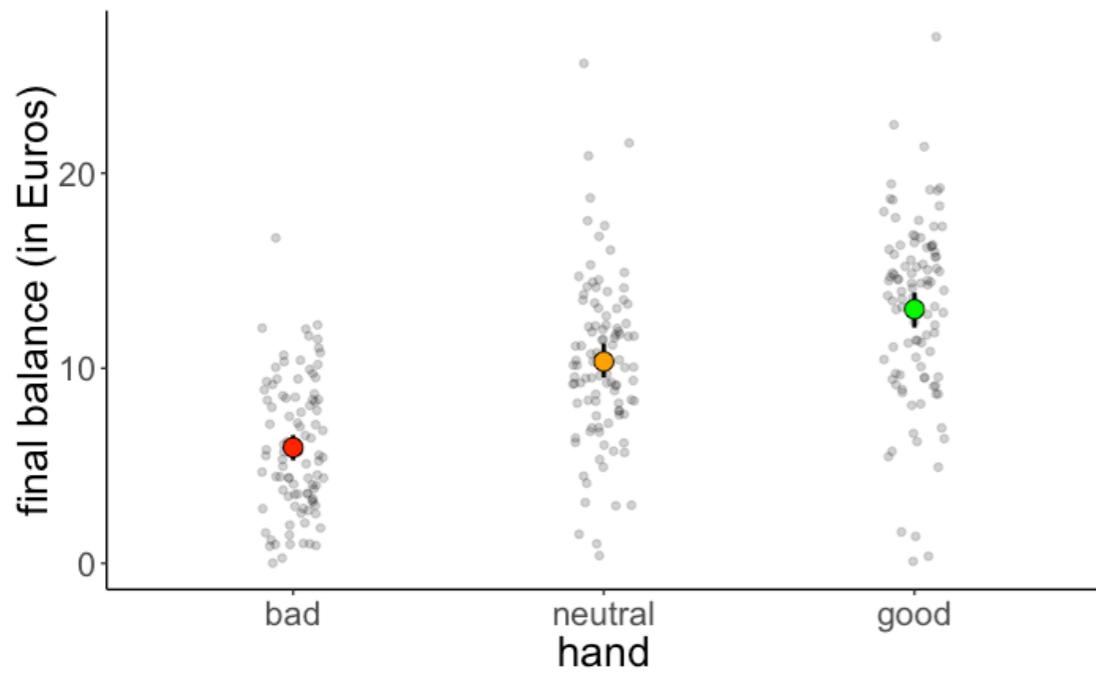
maximum
a posteriori

parameter	lower	mode	upper
b_handgood	5.97	7.07	8.27
b_handneutral	3.21	4.43	5.51
b_intercept	5.17	5.95	6.77
sigma	3.81	4.12	4.47

MAP estimate and 95%
highest density interval

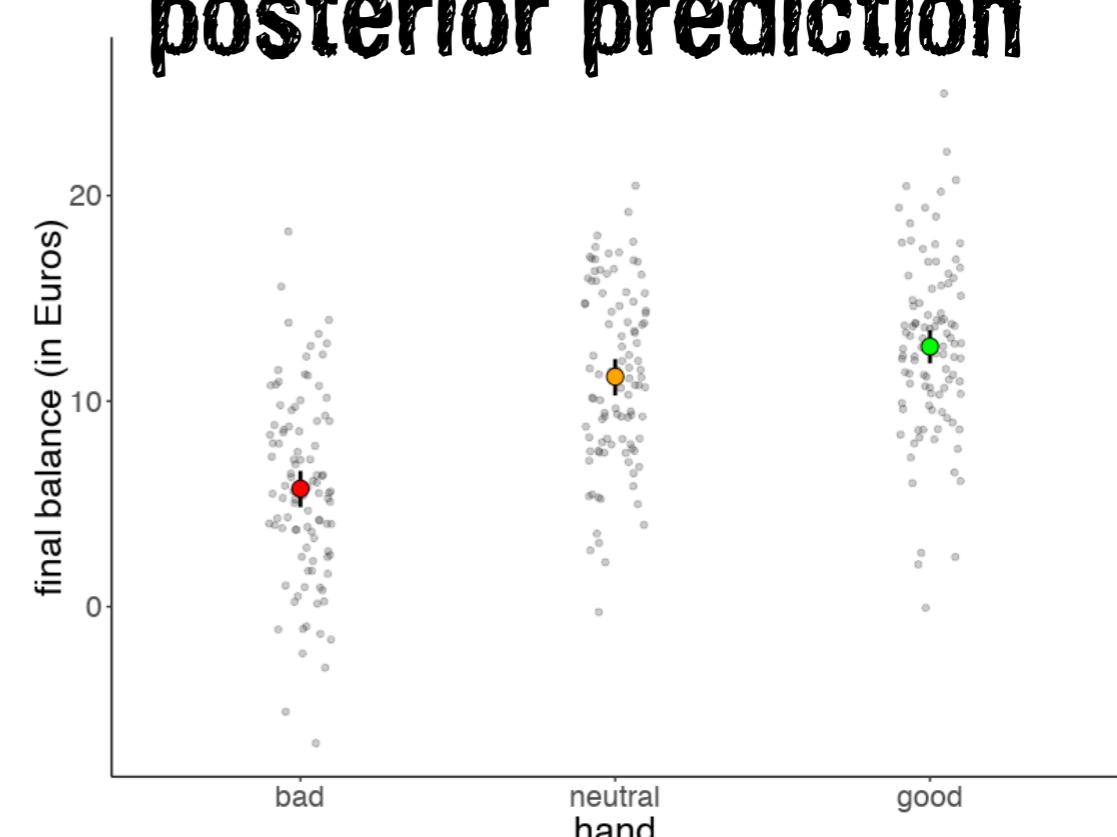
Posterior predictive check

original data

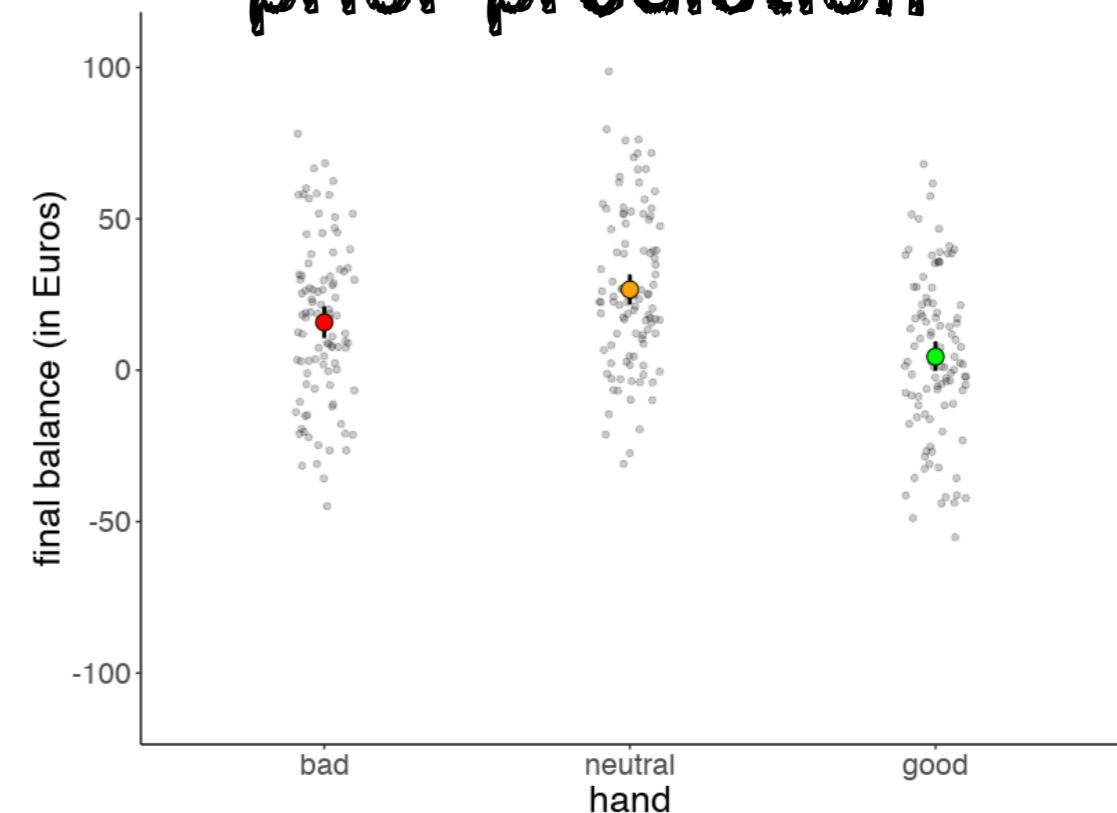


take a look at course notes
for how to make these

posterior prediction



prior prediction

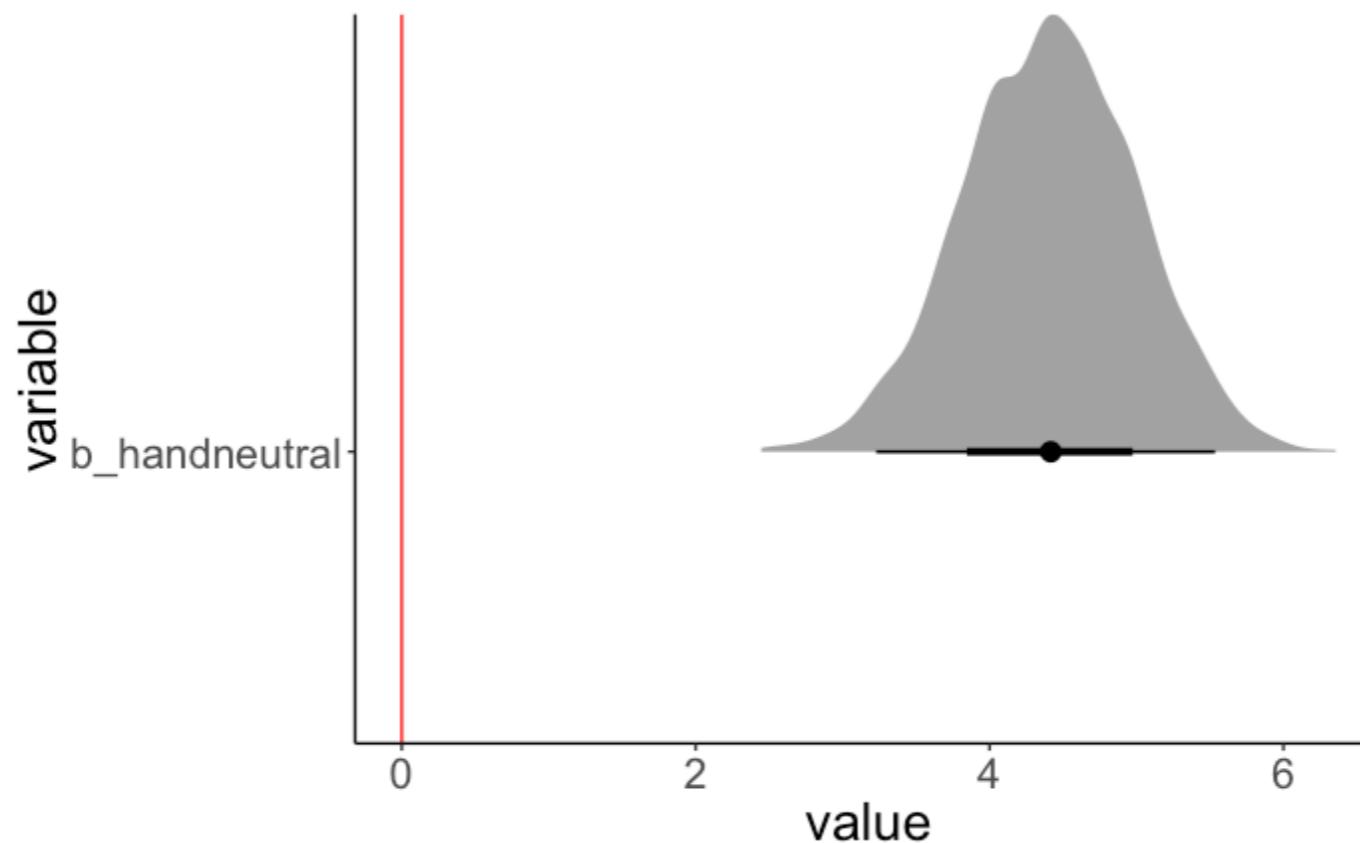


Plan for today

- Building Bayesian models with `brms`
- Model evaluation
- **Testing hypotheses**
- Model comparison
- Inference evaluation: Did things work out?
- Reporting results
- Steps for fitting a Bayesian model in `brms`
 - Sleep data
 - Titanic data

Asking questions based on the posterior

Do neutral hands earn more money than bad hands?



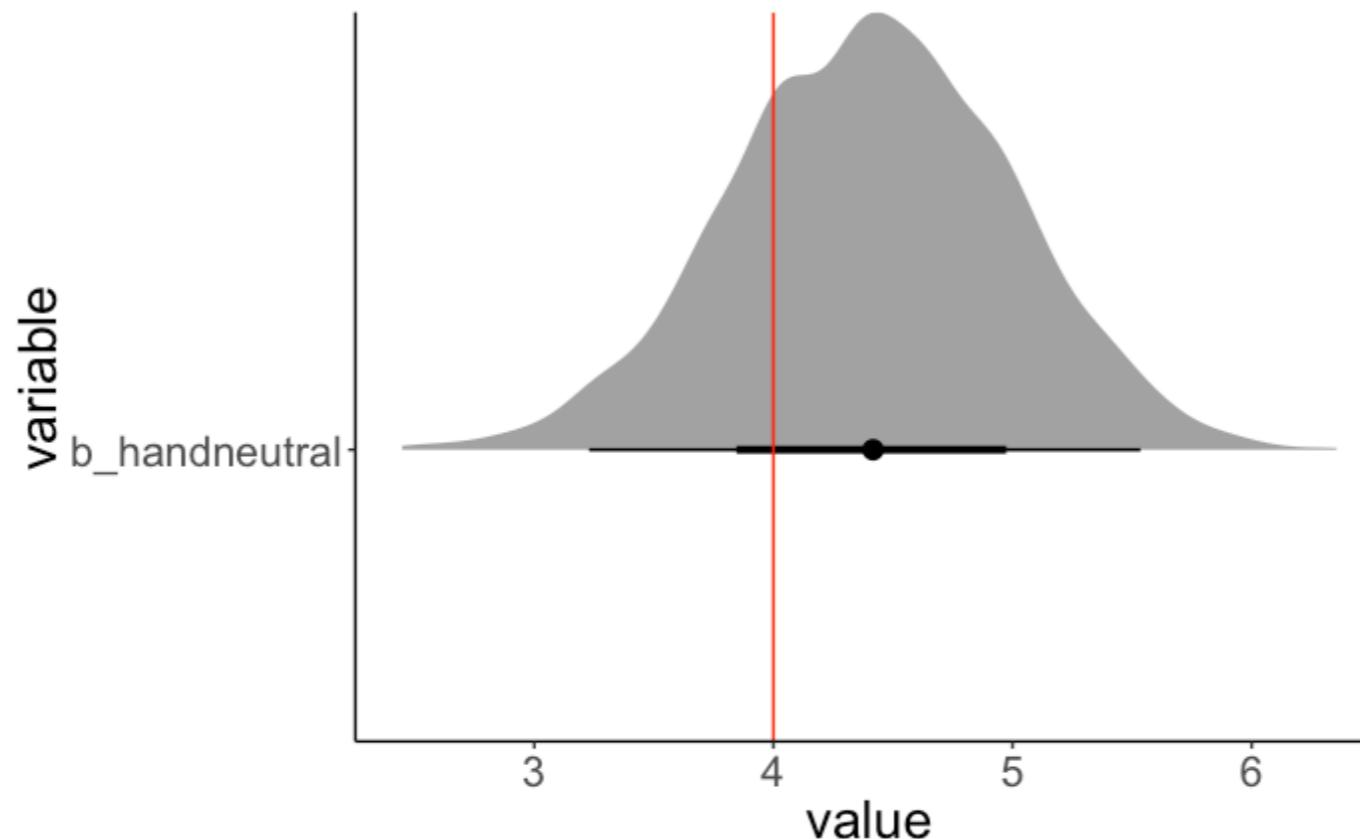
What's the probability that `handneutral` is less than 0?

```
1 hypothesis(fit.brn,  
2             hypothesis = "handneutral < 0")
```

$$p = 0$$

Asking questions based on the posterior

Do neutral hands earn much more money than bad hands?



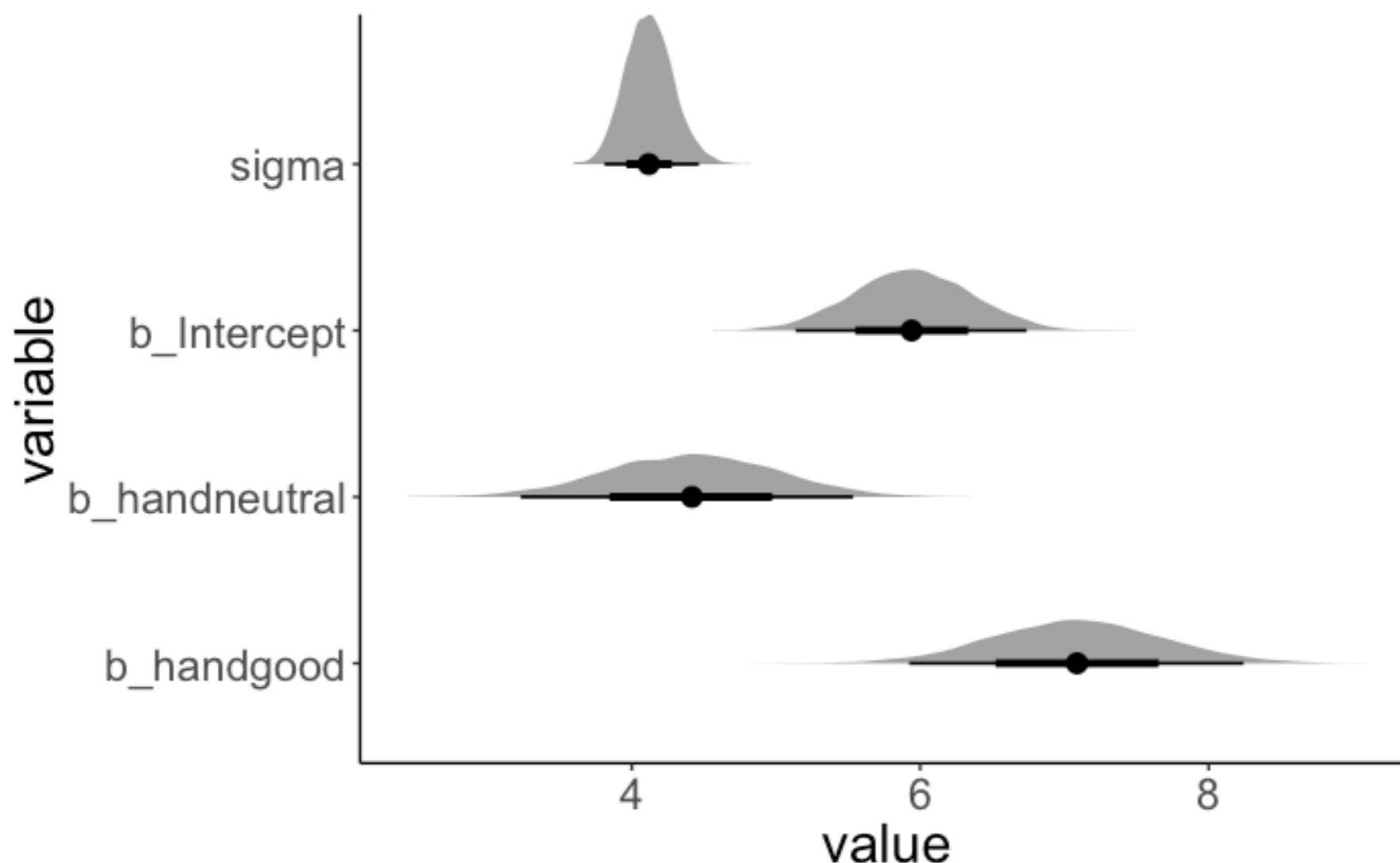
What's the probability that `handneutral` is **more than 4**?

```
1 hypothesis(fit.brm,  
2             hypothesis = "handneutral > 4")
```

$$p = 0.75$$

Asking questions based on the posterior

Do good hands make twice as much as bad hands?

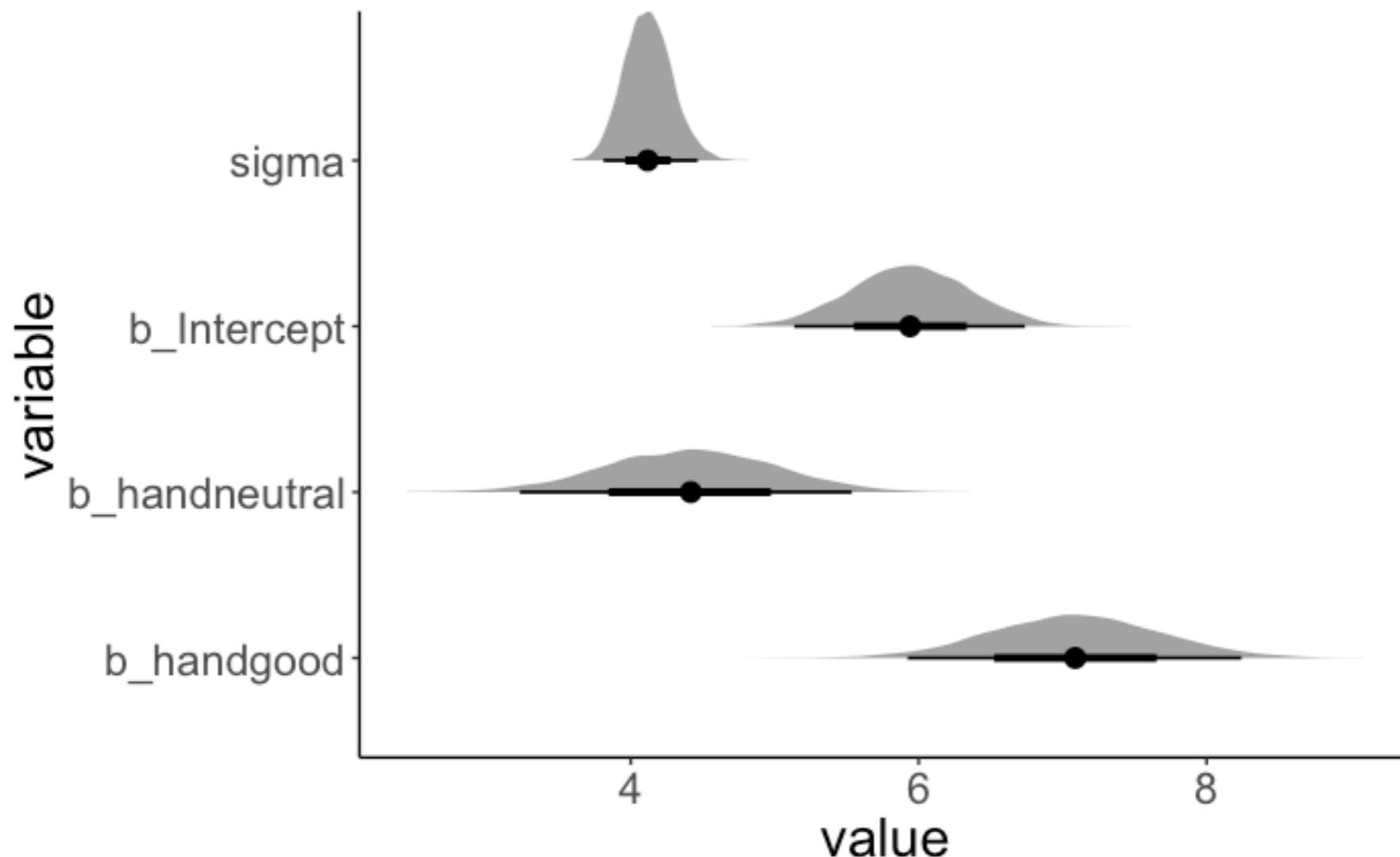


```
1 hypothesis(fit.brm,  
2   hypothesis = "handgood + Intercept > 2 * Intercept")
```

$p = 0.89$

Asking questions based on the posterior

Are neutral hands worse than bad and good hands combined?



```
1 hypothesis(fit.brm,  
2 hypothesis = "Intercept + handneutral < (Intercept + Intercept + handgood) / 2")
```

$p = 0.04$

Testing hypothesis

```
1 df.hypothesis = fit.brm %>%
2   posterior_samples() %>%
3   clean_names() %>%
4   select(starts_with("b_")) %>%
5   mutate(neutral = b_intercept + b_handneutral,
6         bad_good_average = (b_intercept + b_intercept + b_handgood)/2,
7         hypothesis = neutral < bad_good_average)
```

samples
from the
posterior



b_intercept	b_handneutral	b_handgood	neutral	bad_good_average	hypothesis
6.07	4.10	7.20	10.17	9.67	FALSE
6.06	4.44	6.95	10.49	9.53	FALSE
5.88	5.00	6.73	10.87	9.24	FALSE
5.85	4.78	6.18	10.63	8.94	FALSE
5.86	4.46	7.68	10.32	9.70	FALSE

```
1 df.hypothesis %>%
2   summarize(p = sum(hypothesis) / n())
```

$$p = 0.04$$

Testing hypotheses

Having a posterior distribution allows us to ask questions about the data in a very flexible way!

Plan for today

- Building Bayesian models with `brms`
- Model evaluation
- Testing hypotheses
- **Model comparison**
- Inference evaluation: Did things work out?
- Reporting results
- Steps for fitting a Bayesian model in `brms`
 - Sleep data
 - Titanic data

Bayes factor

$$p(H|D) = \frac{p(D|H) \cdot p(H)}{\sum_{i=1}^n p(D|H_i) \cdot p(H_i)}$$

Two hypotheses

$$p(H_1|D) = \frac{p(D|H_1) \cdot p(H_1)}{p(D|H_1) \cdot p(H_1) + p(D|H_2) \cdot p(H_2)}$$

Odds form = Bayes factor

If priors are equal

$$\frac{p(H_1|D)}{p(H_2|D)} = \frac{p(D|H_1)}{p(D|H_2)} \cdot \frac{p(H_1)}{p(H_2)}$$

$$BF = \frac{p(D|H_1)}{p(D|H_2)}$$

Fit two models and compare

```
1 fit.brm2 = brm(formula = balance ~ 1 + hand,  
2                   data = df.poker,  
3                   save_all_pars = T, ← needs to be set  
4                   file = "cache/brm2")  
5  
6 fit.brm3 = brm(formula = balance ~ 1 + hand + skill,  
7                   data = df.poker,  
8                   save_all_pars = T,  
9                   file = "cache/brm3")
```

bayes_factor(fit.brm3, fit.brm2)

$$BF = \frac{p(D | H_1)}{p(D | H_0)} = 3.81$$

relative evidence for one model over the other

Bayes factor BF_{10}			Interpretation
	>	100	Decisive evidence for H_1
30	-	100	Very Strong evidence for H_1
10	-	30	Strong evidence for H_1
3	-	10	Substantial evidence for H_1
1	-	3	Anecdotal evidence for H_1
	1		No evidence
1/3	-	1	Anecdotal evidence for H_0
1/10	-	1/3	Substantial evidence for H_0
1/30	-	1/10	Strong evidence for H_0
1/100	-	1/30	Very Strong evidence for H_0
	<	1/100	Decisive evidence for H_0

Approximate cross-validation

```
1 fit.brm2 = add_criterion(fit.brm2,  
2                           criterion = c("loo", "waic"),  
3                           reloo = T)  
4  
5 fit.brm3 = add_criterion(fit.brm3,  
6                           criterion = c("loo", "waic"),  
7                           reloo = T)  
8  
9 loo_compare(fit.brm2, fit.brm3)
```

	elpd_diff	se_diff
fit.brm3	0.0	0.0
fit.brm2	-0.3	1.5

not much of a difference ...

expected log pointwise predictive density for a new dataset

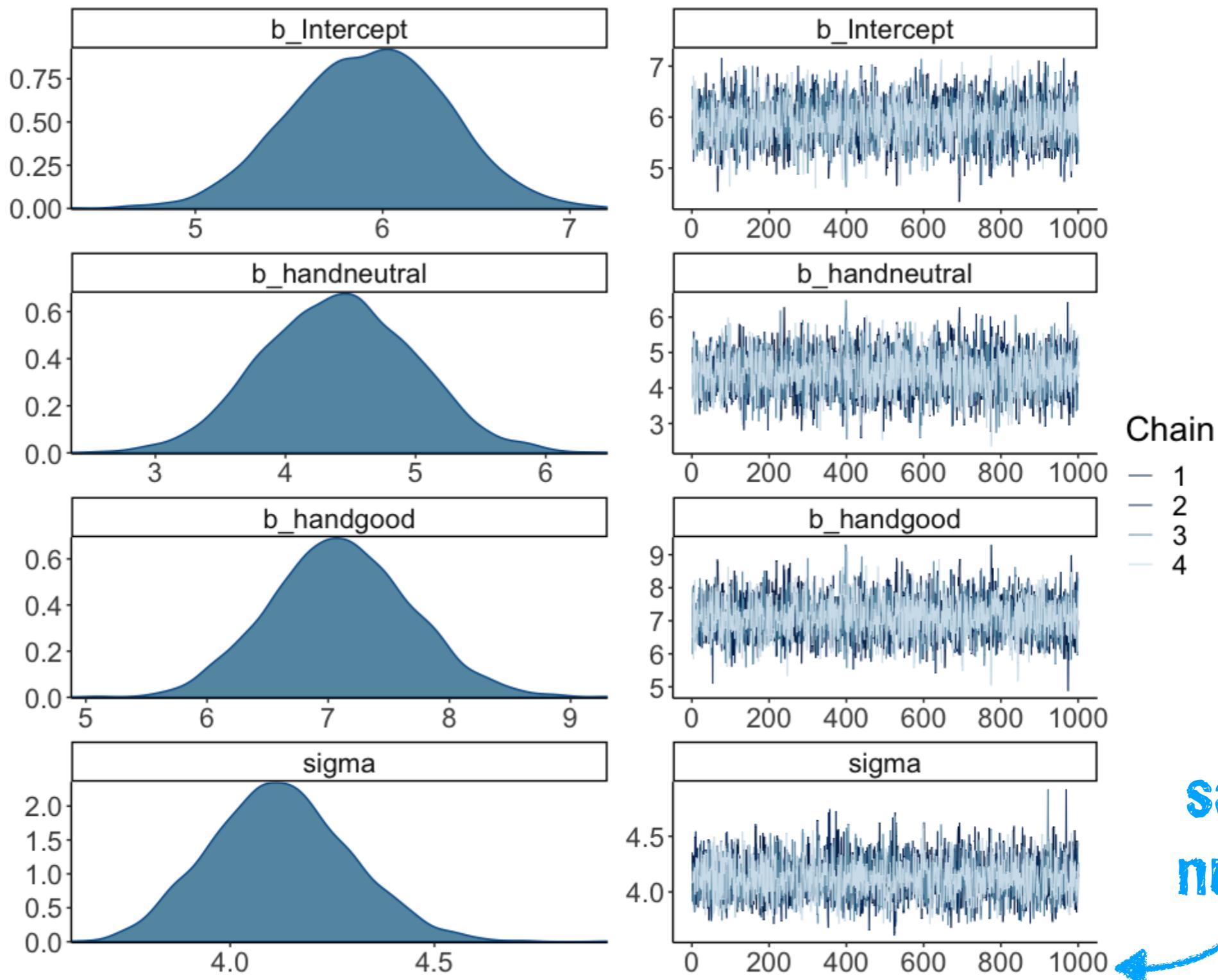
rule of thumb: elpd_diff should be at least twice the standard error (se_diff) to conclude that one model is better than the other

Plan for today

- Building Bayesian models with `brms`
- Model evaluation
- Testing hypotheses
- Model comparison
- **Inference evaluation: Did things work out?**
- Reporting results
- Steps for fitting a Bayesian model in `brms`
 - Sleep data
 - Titanic data

Can we trust the model results?

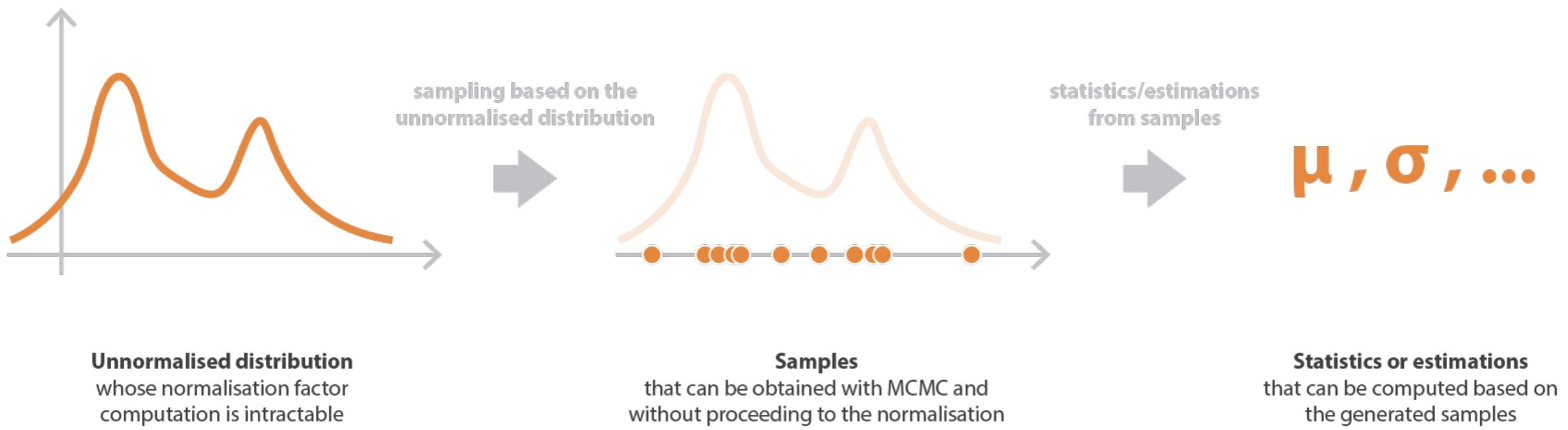
`plot(fit.brm)`



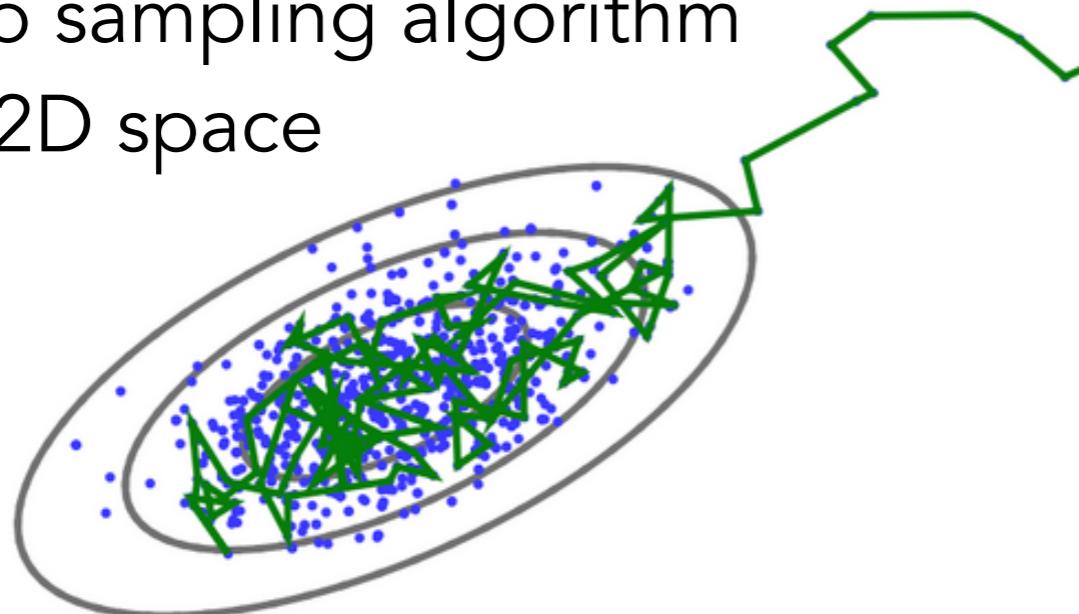
sample
number

Can we trust the model results?

Inference via Markov Chain Monte Carlo (MCMC)

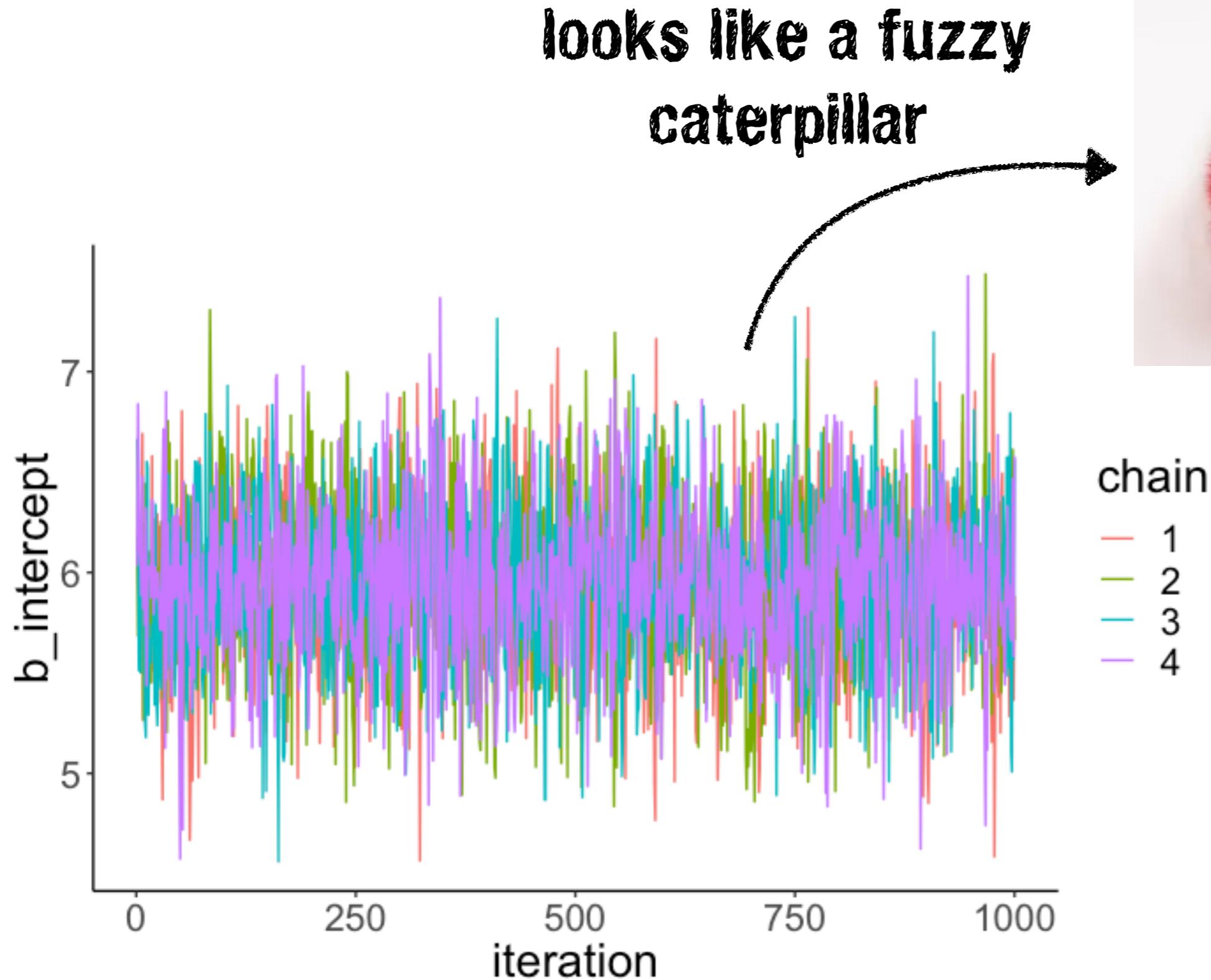


Markov Chain Monte Carlo sampling algorithm in a 2D space



goal: draw **independent** samples from the posterior distribution

Can we trust the model results?



When things don't work out

```
1 df.data = tibble(y = c(-1, 1)) ← only two data points!
2
3 fit.brm5 = brm(data = df.data,
4                  family = gaussian,
5                  formula = y ~ 1,
6                  prior = c(prior(uniform(-1e10, 1e10), class = Intercept),
7                            prior(uniform(0, 1e10), class = sigma)),
8                  inits = list(list(Intercept = 0, sigma = 1),
9                            list(Intercept = 0, sigma = 1)),
10                 iter = 4000,
11                 warmup = 1000,
12                 chains = 2,
13                 file = "cache/brm5")
```

incredibly wide uniform priors
10000000000

When things don't work out

summary (fit.brml)

```
The model has not converged (some Rhats are > 1.1). Do not analyse the results!
We recommend running more iterations and/or setting stronger priors. There were 1203
divergent transitions after warmup. Increasing adapt_delta above 0.8 may help.
See http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup Family:
gaussian
```

```
Links: mu = identity; sigma = identity
Formula: y ~ 1
Data: df.data (Number of observations: 2)
Samples: 2 chains, each with iter = 4000; warmup = 1000; thin = 1;
         total post-warmup samples = 6000
```

Population-Level Effects:

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	357550121.58	1416057299.71	-2244033111.47	3333594132.43	1.78	3	24

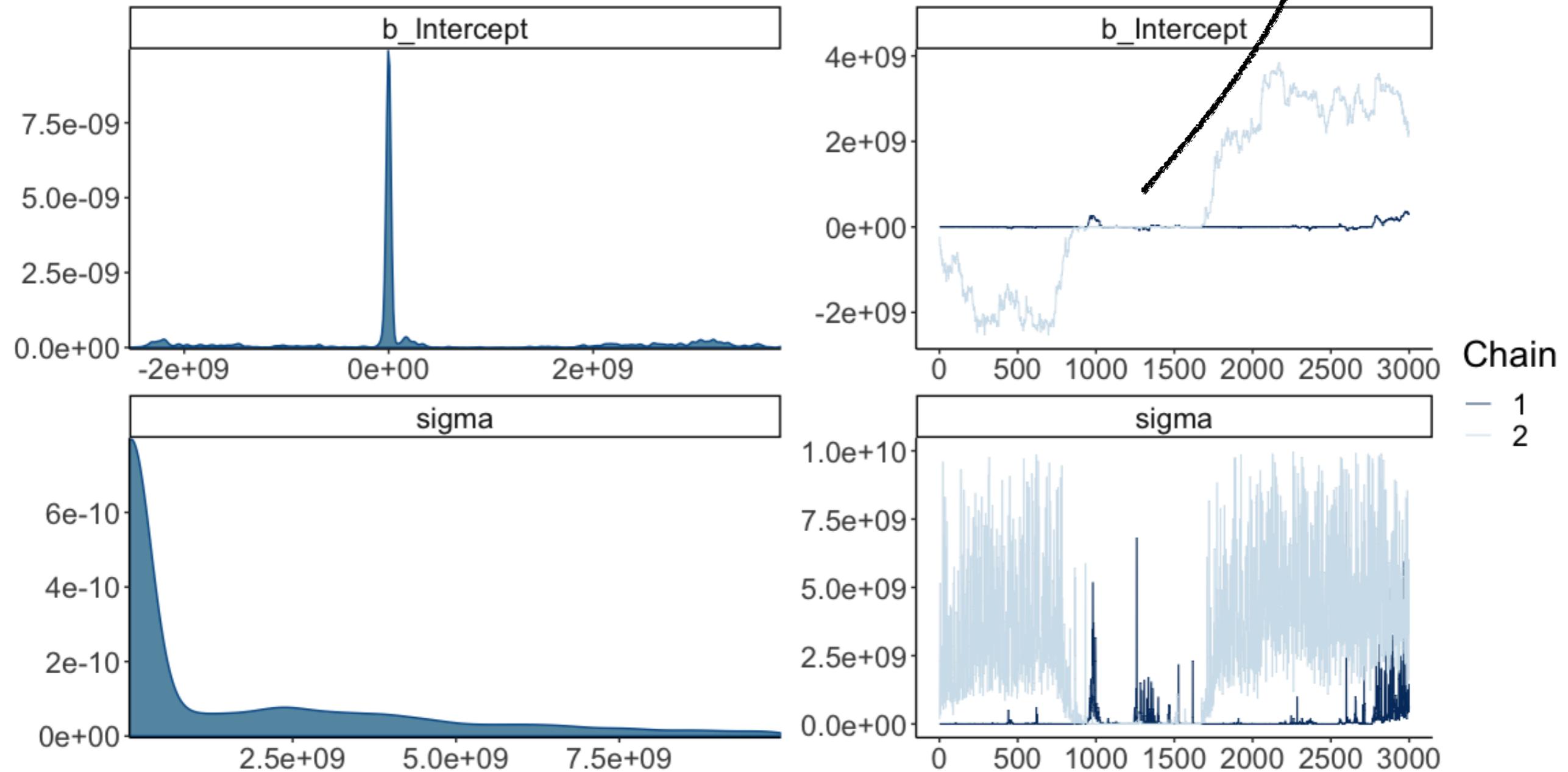
Family Specific Parameters:

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
sigma	1524412740.64	2392424321.98	21668.93	8317582240.06	1.40	4	41

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).

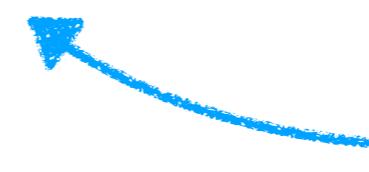
When things don't work out

doesn't look like a
fuzzy caterpillar



Having somewhat informative priors fixes things

```
1 fit.brm6 = brm(data = df.data,
2                   family = gaussian,
3                   formula = y ~ 1,
4                   prior = c(prior(normal(0, 10), class = Intercept),
5                             prior(cauchy(0, 1), class = sigma)),
6                   iter = 4000,
7                   warmup = 1000,
8                   chains = 2,
9                   seed = 1,
10                  file = "cache/brm6")
```



more reasonable priors

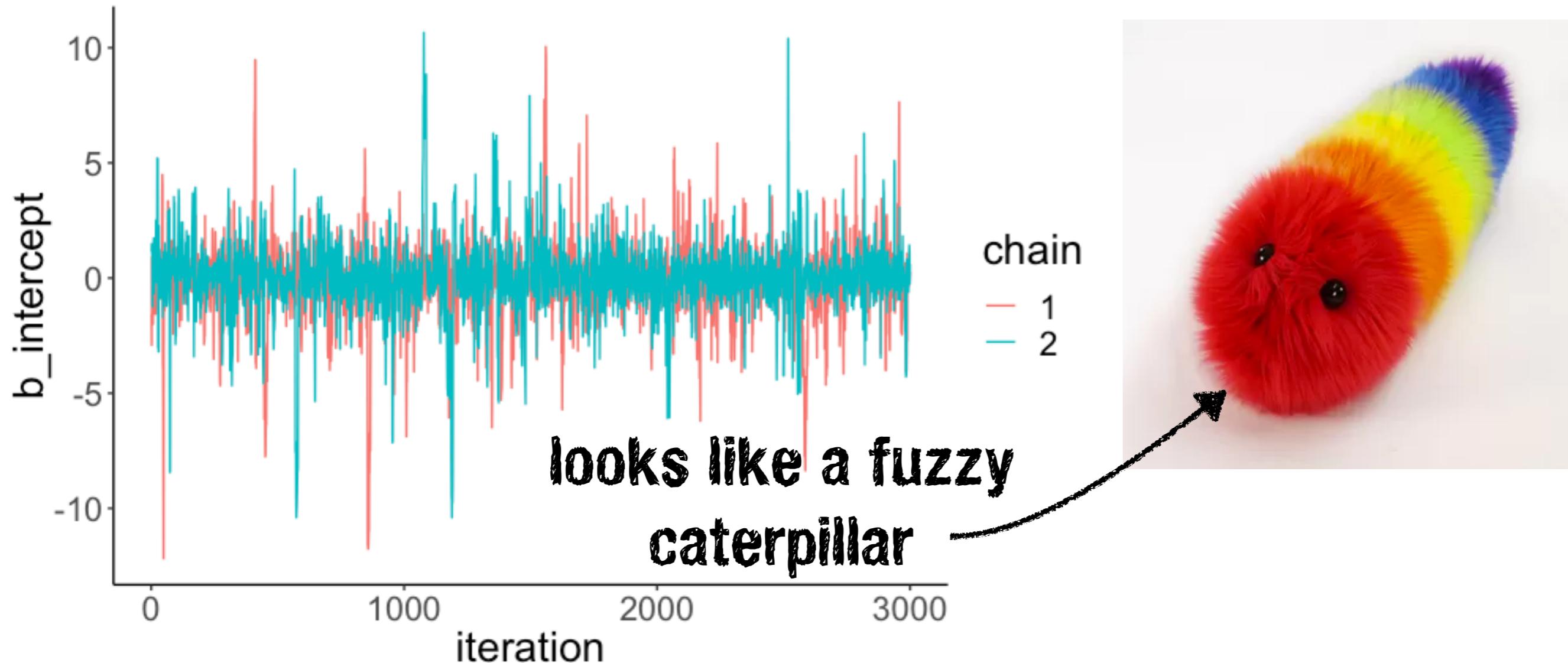
```
Family: gaussian
Links: mu = identity; sigma = identity
Formula: y ~ 1
Data: list(y = c(-1, 1)) (Number of observations: 2)
Samples: 2 chains, each with iter = 4000; warmup = 1000; thin = 1;
         total post-warmup samples = 6000
```

```
Population-Level Effects:
Estimate Est.Error 1-95% CI u-95% CI Eff.Sample Rhat
Intercept -0.06     1.72    -3.78     3.27      1033 1.00
```

```
Family Specific Parameters:
Estimate Est.Error 1-95% CI u-95% CI Eff.Sample Rhat
sigma     2.21      6.99     0.61     6.92      1006 1.00
```

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).

Having somewhat informative priors fixes things



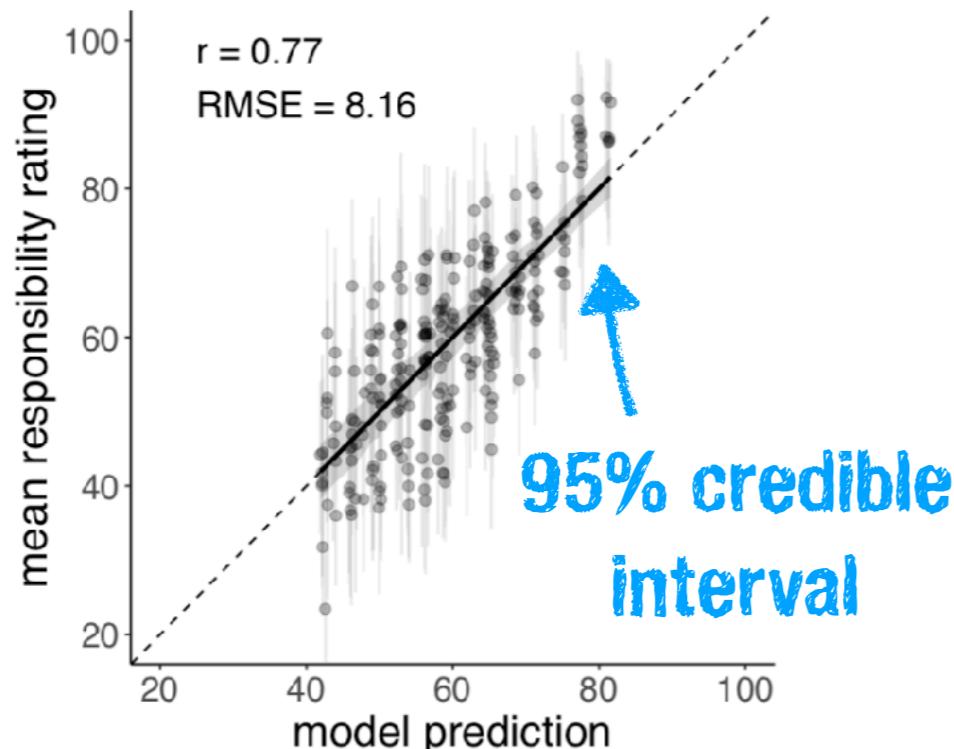
if things go wrong:

- set more informative priors
- run more warm-up samples
- adjust the sampling algorithm as suggested via the control argument

Reporting results

Reporting results

Plots



Tables

Table 1
Estimates of the mean, standard error, and 95% HDIs of the different predictors in the Bayesian mixed effects model. Note: n causes = number of causes.

responsibility ~ 1 + surprise + pivotality + n causes + (1 + surprise + pivotality + n causes | participant)

term	estimate	std.error	lower 95% HDI	upper 95% HDI
intercept	59.94	3.25	54.70	65.22
surprise	21.68	4.57	14.17	29.23
pivotality	13.52	1.82	10.47	16.53
n causes	-5.72	0.50	-6.55	-4.90

model formula

parameter estimates

Text

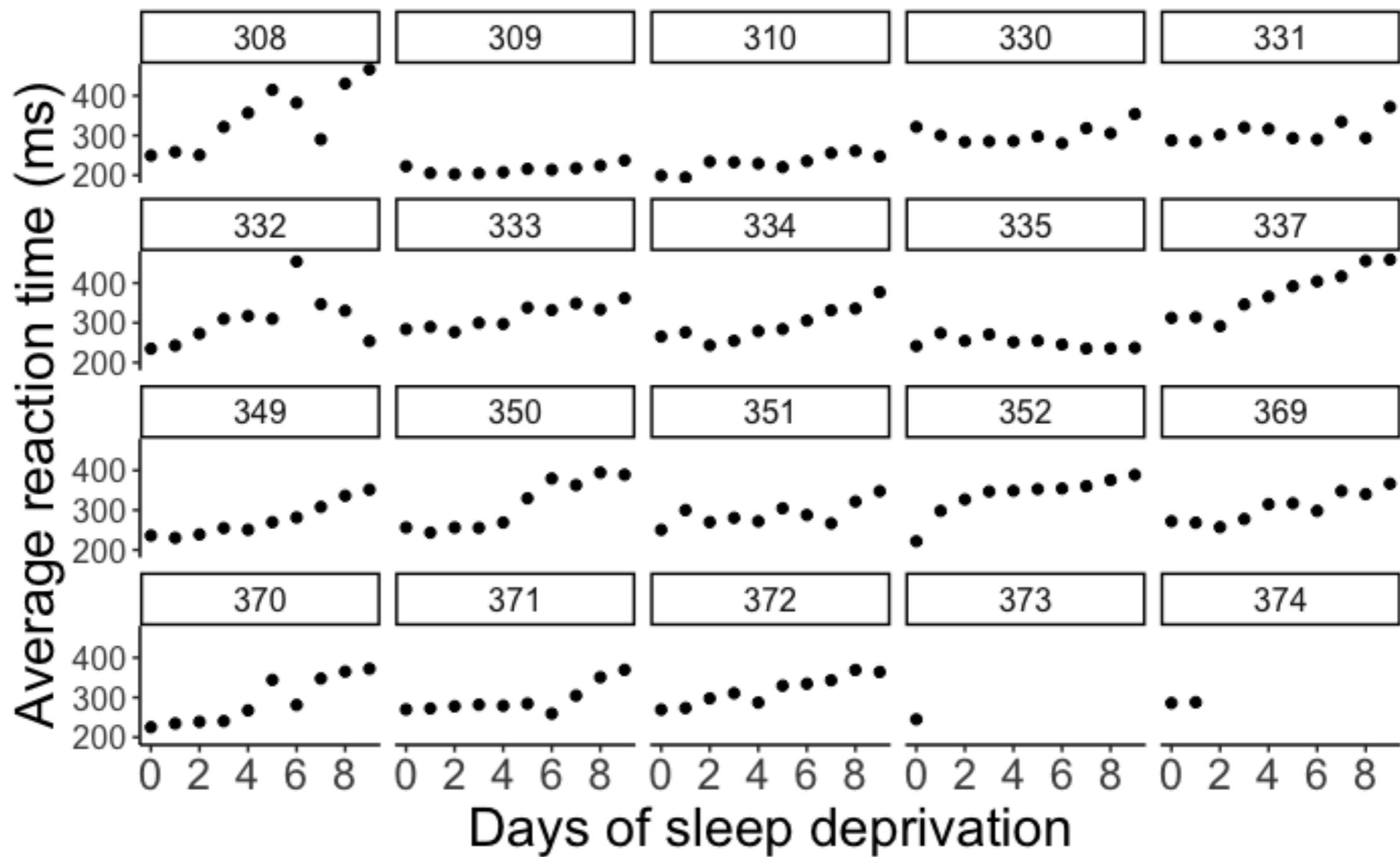
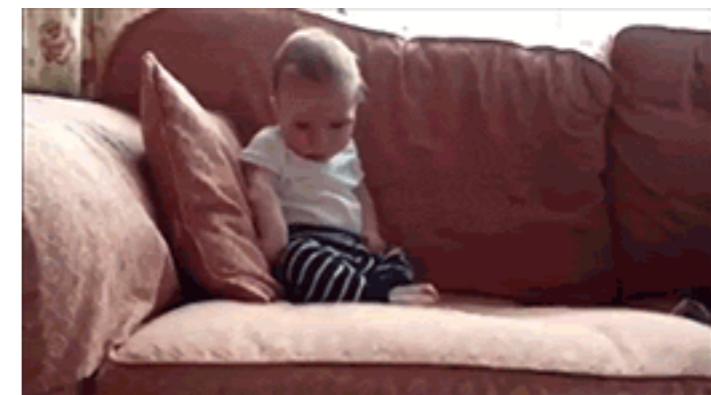
We computed a Bayesian mixed effects model with random intercepts and slopes to predict participants' responsibility judgments (see Table 1). Figure 6b shows a scatter plot of the model predictions and participants' responsibility judgments for the full set of 170 scenarios (with 250 judgments). Overall, the model predicts participants' responsibility judgments well with $r = .77$ and RMSE = 8.16. Table 1 shows the estimates of the different predictors. As can be seen, none of the predictors' 95% HDIs overlap with 0.¹

¹For any statistical claim, we report the mean of the posterior distribution together with the 95% highest-density interval (HDI). All Bayesian models were written in Stan (Carpenter et al., 2017) and accessed with the brms package (Bürkner, 2017) in R (R Core Team, 2019).

Steps for fitting a Bayesian model in brms

1. Specify and fit the model
2. Check the model diagnostics
3. Validate the model
 - posterior predictive check
4. Interpret the model parameters
5. Visualize the results
 - posterior summaries
 - show data and model together

Feeling sleepy?



1. Specify and fit the model

```
1 fit.brm7 = brm(formula = reaction ~ 1 + days + (1 + days | subject),  
2                   data = df.sleep,  
3                   file = "cache/brm7")
```



2. Check the model diagnostics

```
1 fit.brm7 %>%
2   summary()
```

```
Family: gaussian
Links: mu = identity; sigma = identity
Formula: reaction ~ 1 + days + (1 + days | subject)
Data: df.sleep (Number of observations: 183)
Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup samples = 4000

Group-Level Effects:
~subject (Number of levels: 20)
  Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sd(Intercept)    26.18     6.25    15.65    40.54 1.00    1879    2463
sd(days)          6.59     1.53     4.14    10.13 1.00    1145    1625
cor(Intercept,days) 0.09     0.29    -0.46     0.67 1.00     993    1526

Population-Level Effects:
  Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept     252.18      6.86   238.47   265.42 1.00    1826    2766
days          10.46      1.69     7.13    13.78 1.00    1203    1782

Family Specific Parameters:
  Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
sigma       25.77      1.57   22.93    29.14 1.00    3864    2773

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).
```

Rhat of

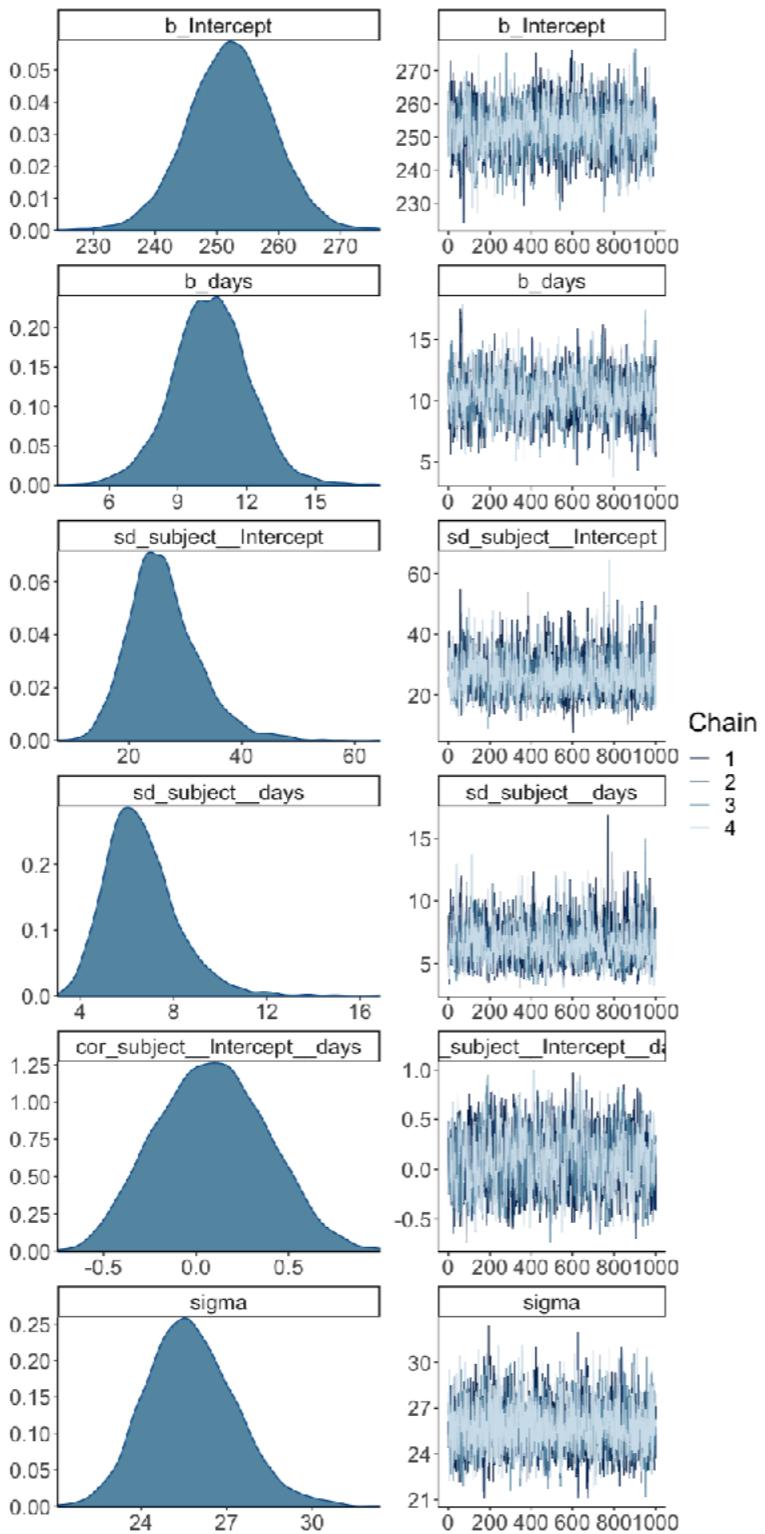
1.00 is
good!

Roughly speaking, the effective sample size (**ESS**) of a quantity of interest captures how many independent draws contain the same amount of information as the dependent sample obtained by the MCMC algorithm.

<https://mc-stan.org/misc/warnings.html>

2. Check the model diagnostics

```
1 fit.brm7 %>%
2   plot(N = 6)
```

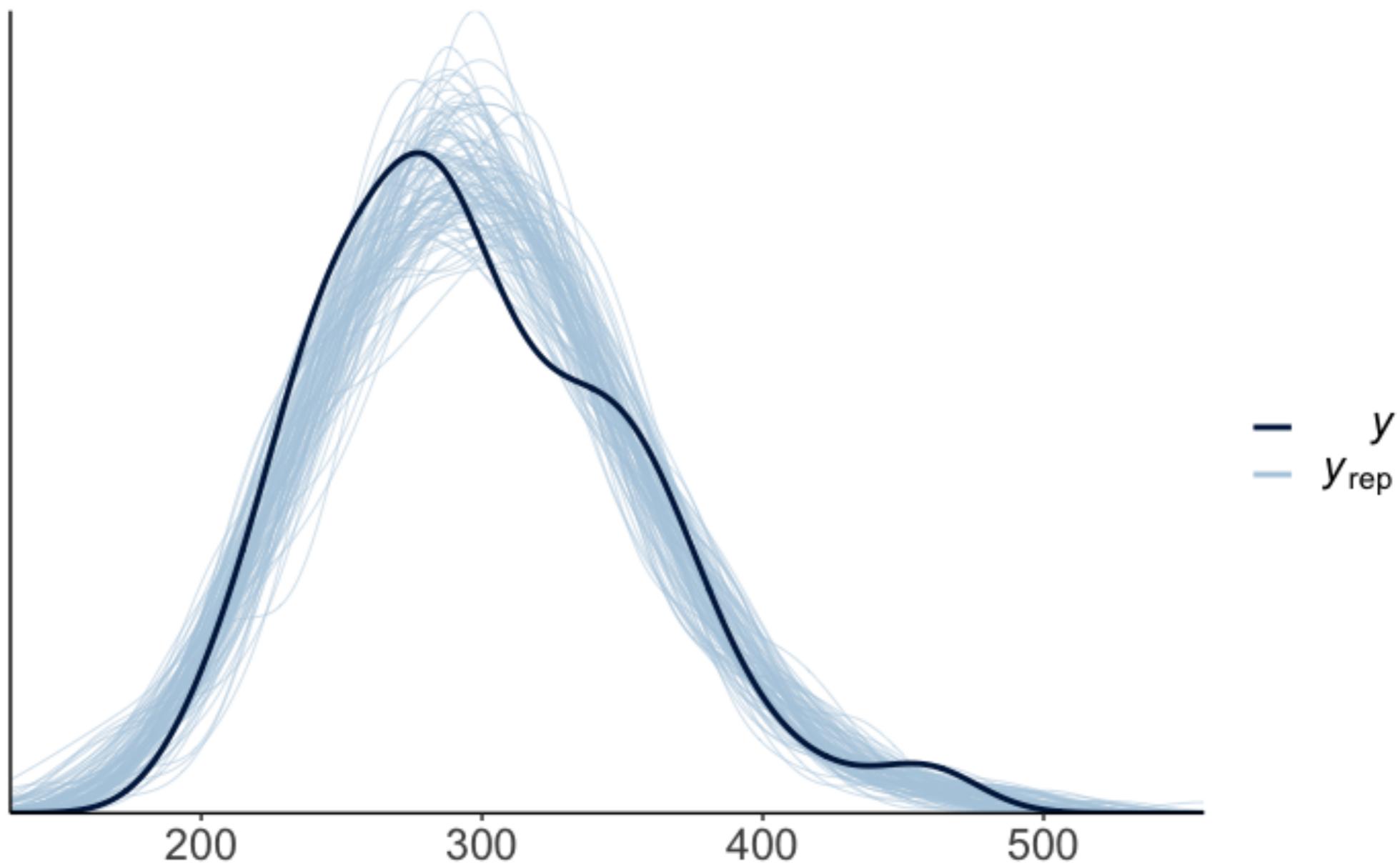


these look good!



3. Validate the model

```
1 pp_check(fit.brm7,  
2           nsamples = 100)
```



posterior predictive distribution looks good

4. Interpret the model parameters

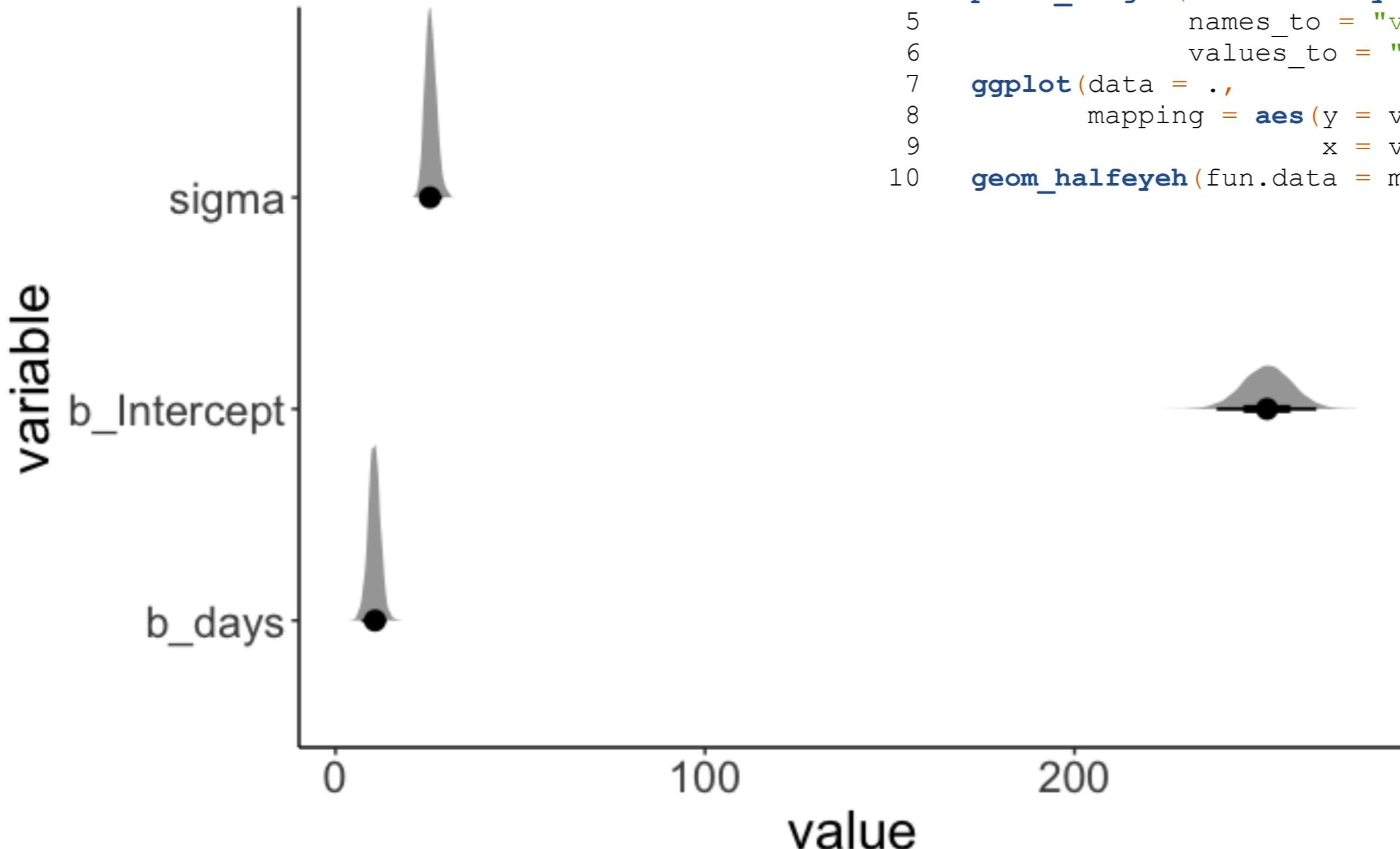
95% highest density interval

```
1 fit.brm7 %>%
  2   tidy()
```

effect	component	group	term	estimate	std.error	conf.low	conf.high
fixed	cond	NA	(Intercept)	252.18	6.86	238.47	265.42
fixed	cond	NA	days	10.46	1.69	7.13	13.78
ran_pars	cond	subject	sd__(Intercept)	26.18	6.25	15.65	40.54
ran_pars	cond	subject	sd__days	6.59	1.53	4.14	10.13
ran_pars	cond	subject	cor__(Intercept).days	0.09	0.29	-0.46	0.67
ran_pars	cond	Residual	sd__Observation	25.77	1.57	22.93	29.14

- interpretation of parameter estimates is the same as in frequentist analysis
- highest density intervals have a more intuitive interpretation than confidence intervals

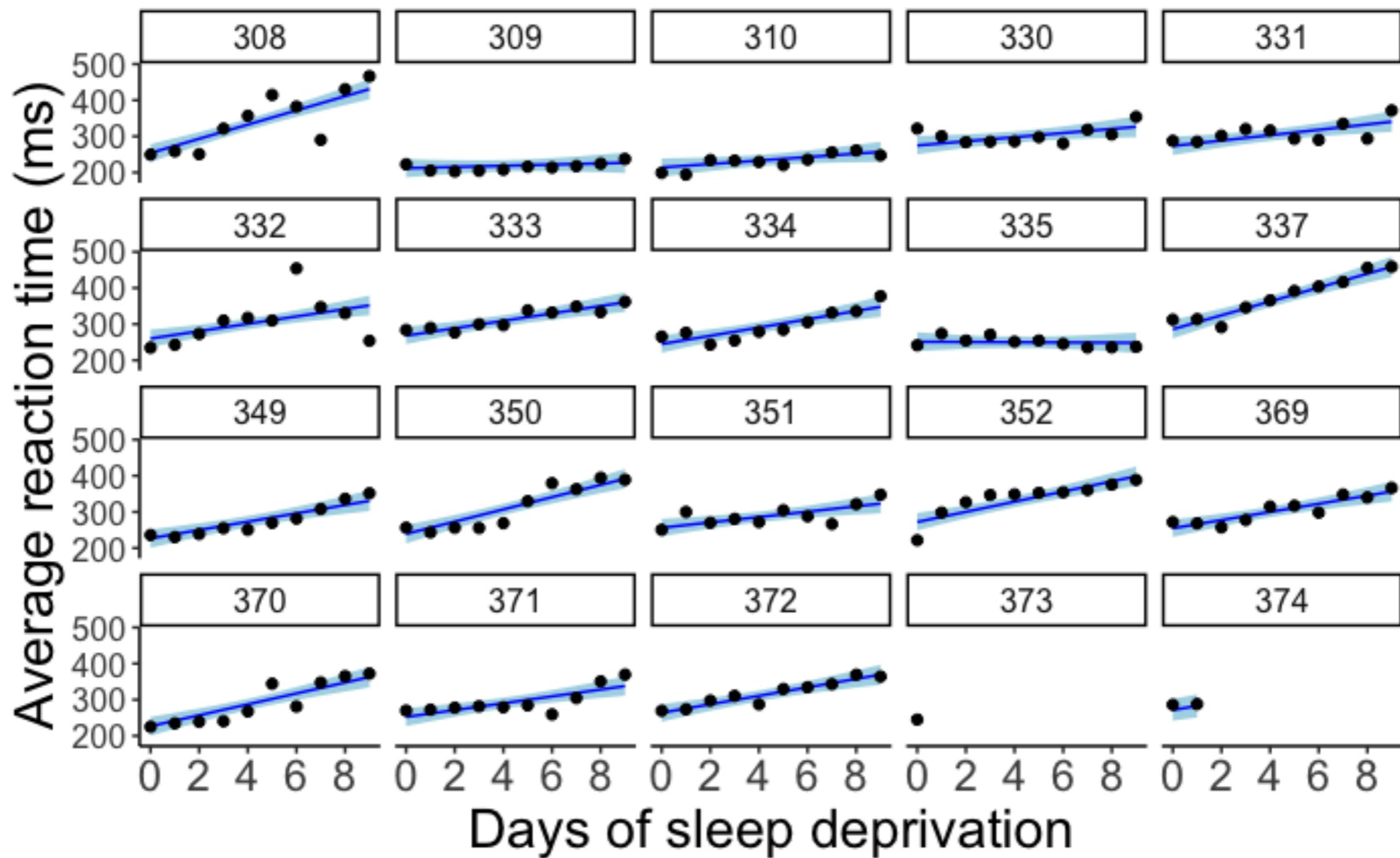
5. Visualize the results



```
1 fit.brm7 %>%
2   posterior_samples() %>%
3   select(-c(lp__, contains("subject")))) %>%
4   pivot_longer(cols = everything(),
5                 names_to = "variable",
6                 values_to = "value") %>%
7   ggplot(data = .,
8         mapping = aes(y = variable,
9                        x = value)) +
10  geom_halfeyeh(fun.data = mode_hdih)
```

posterior distributions of the different model parameters

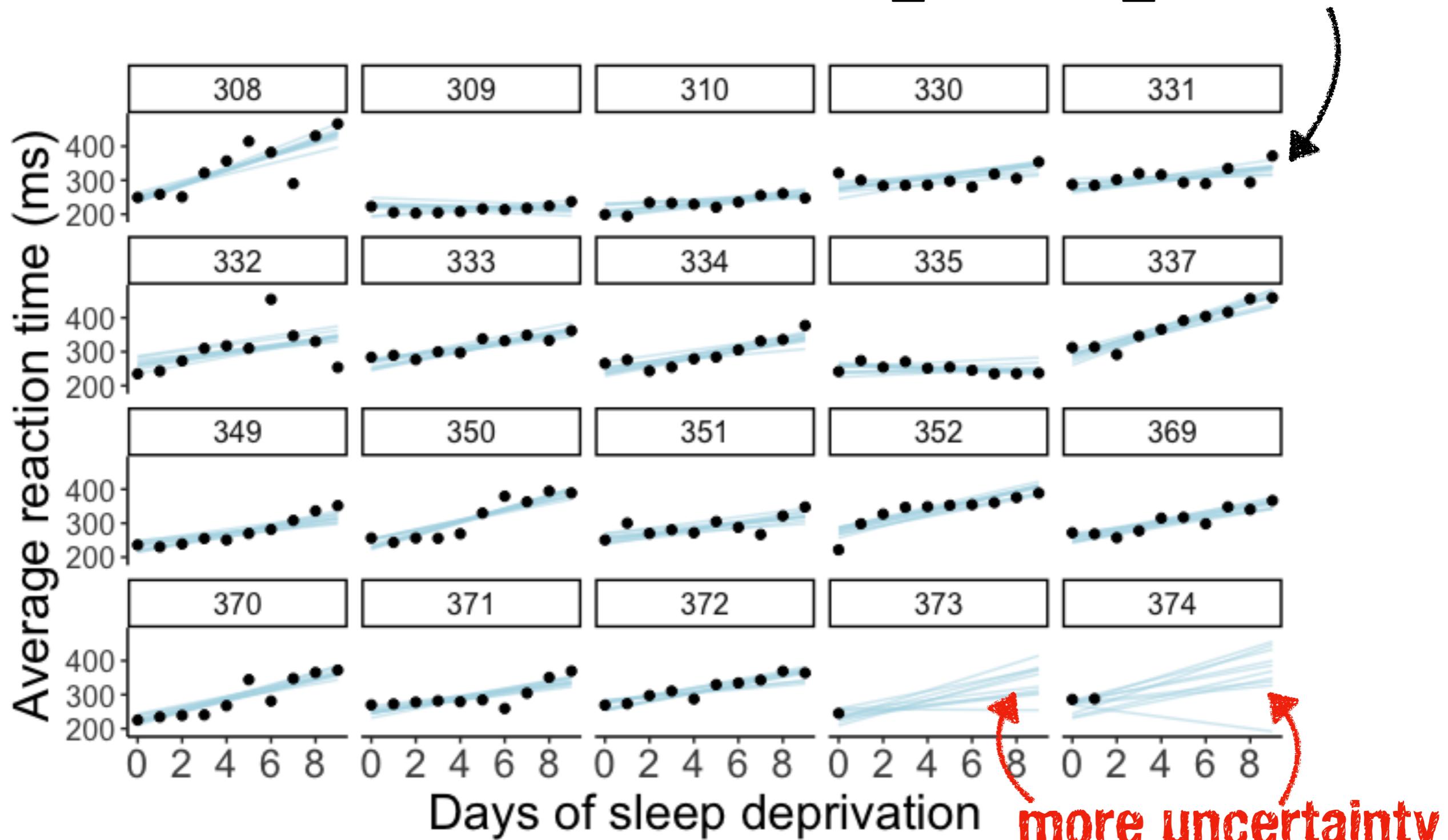
5. Visualize the results



regression lines with 95% highest density intervals

5. Visualize the results

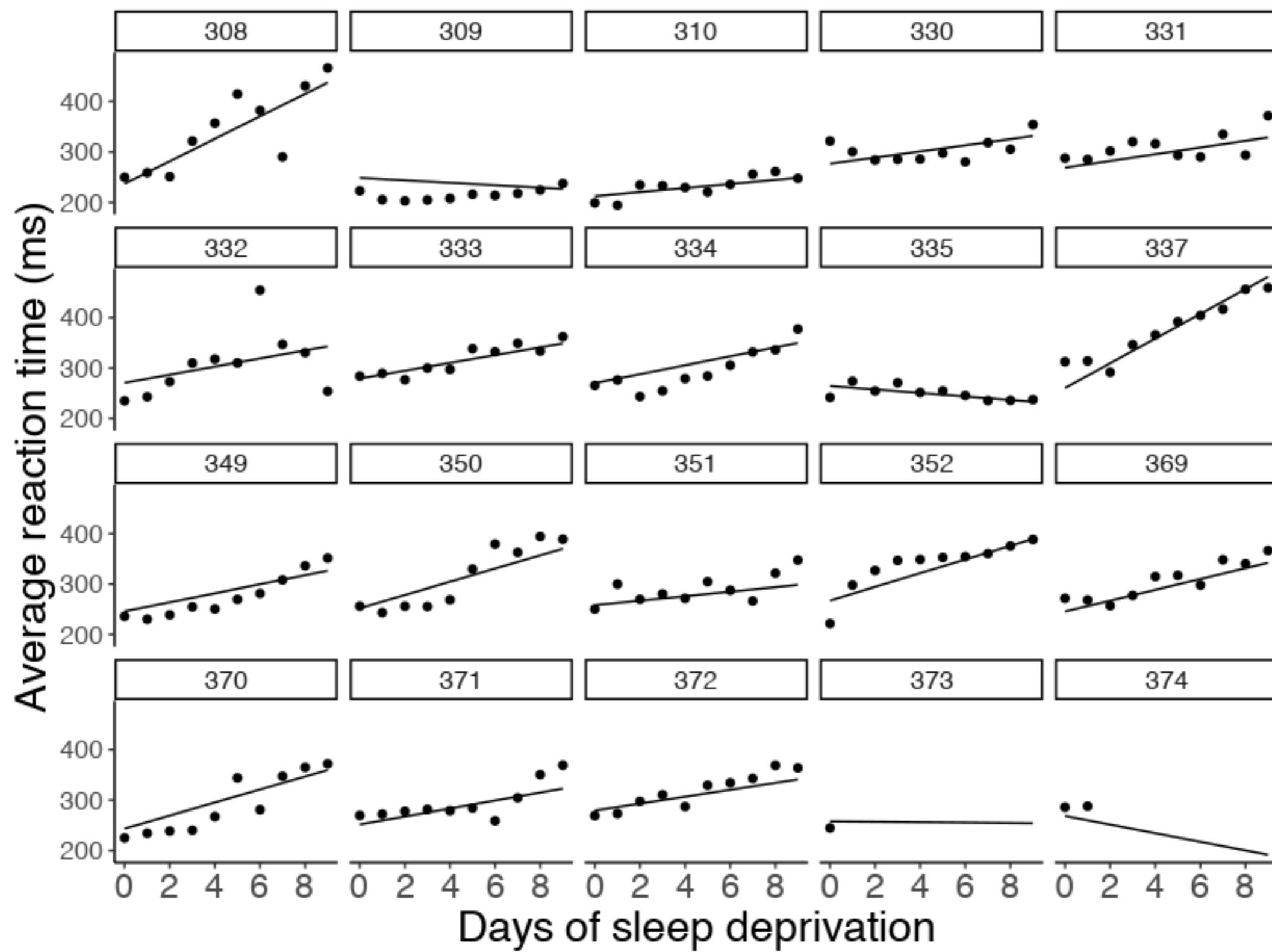
`add_fitted_draws()`



10 random samples from the posterior distribution

5. Visualize the results

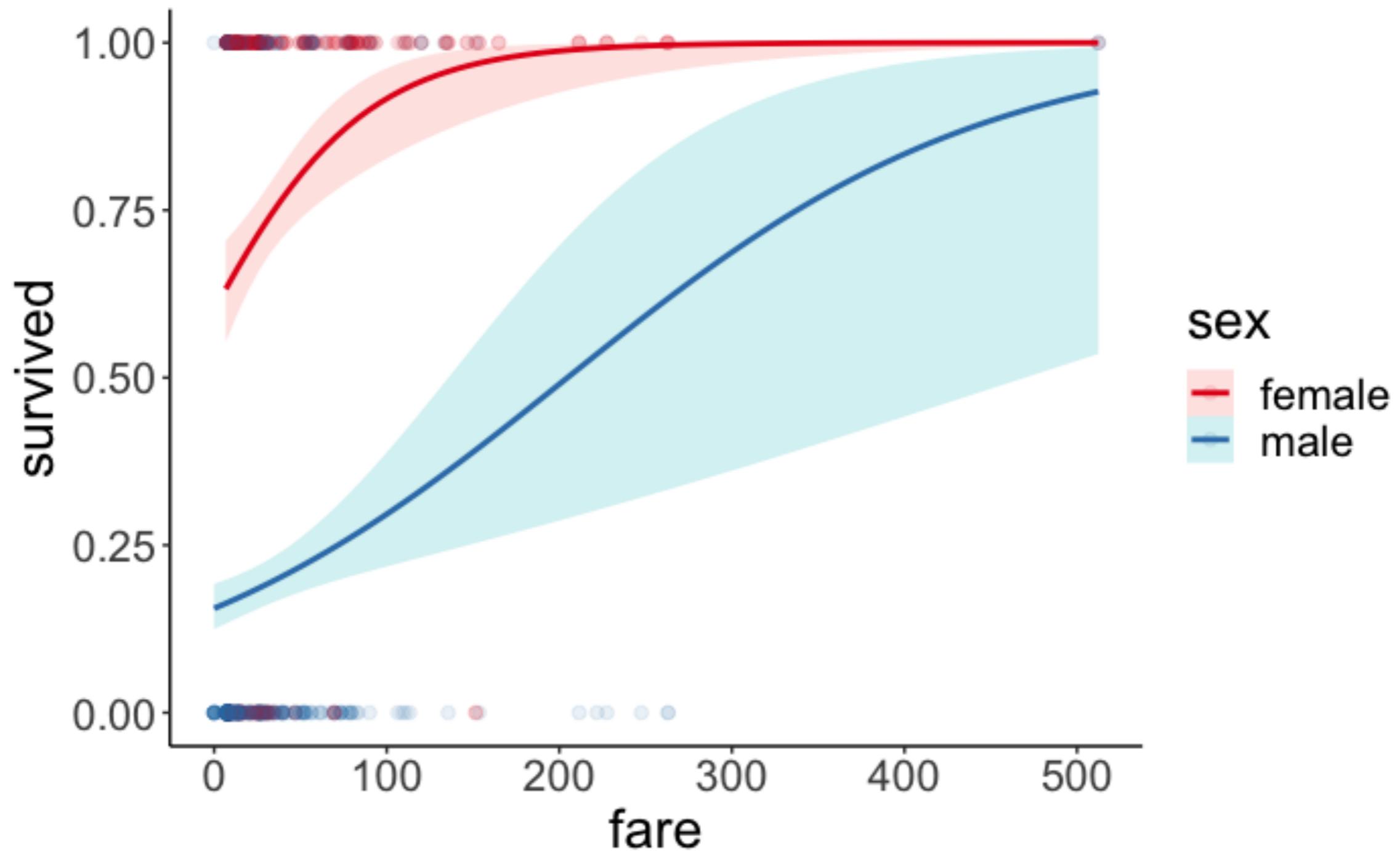
if you're feeling fancy



Steps for fitting a Bayesian model in brms

1. Specify and fit the model
2. Check the model diagnostics
3. Validate the model
 - posterior predictive check
4. Interpret the model parameters
5. Visualize the results
 - posterior summaries
 - show data and model together

Feeling cold?



1. Specify and fit the model

```
1 fit.brm8 = brm(formula = survived ~ 1 + fare + sex,  
2                   family = "bernoulli",  
3                   data = df.titanic,  
4                   file = "cache/brm8",  
5                   seed = 1)
```

2. Check the model diagnostics

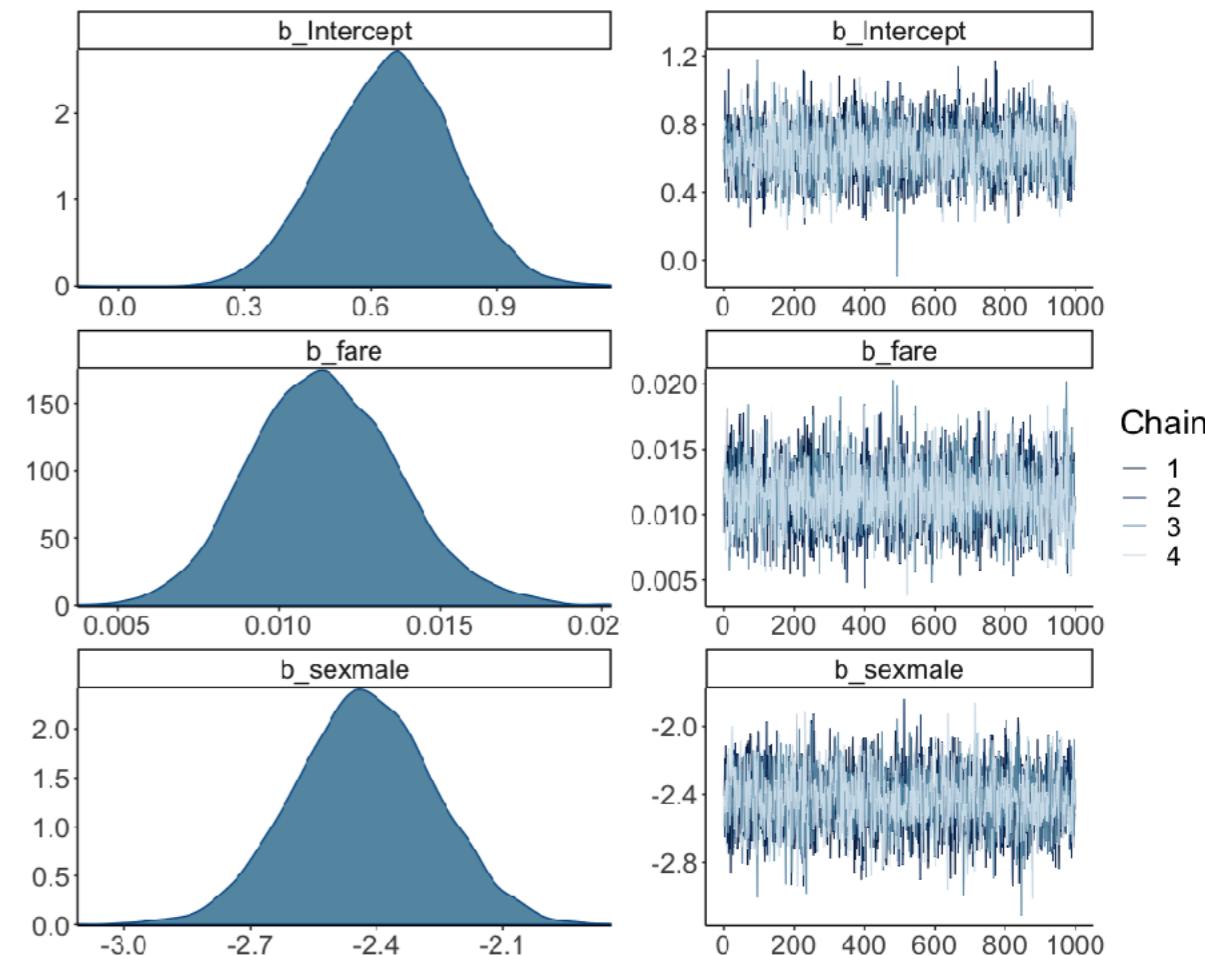
```
1 fit.brms %>%
2   summary()
```

```
Family: bernoulli
Links: mu = logit
Formula: survived ~ 1 + fare + sex
Data: df.titanic (Number of observations: 891)
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    0.64     0.15    0.35    0.93 1.00   3446   3198
fare          0.01     0.00    0.01    0.02 1.00   5268   2420
sexmale      -2.43     0.17   -2.76   -2.10 1.00   2959   2502

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).
```

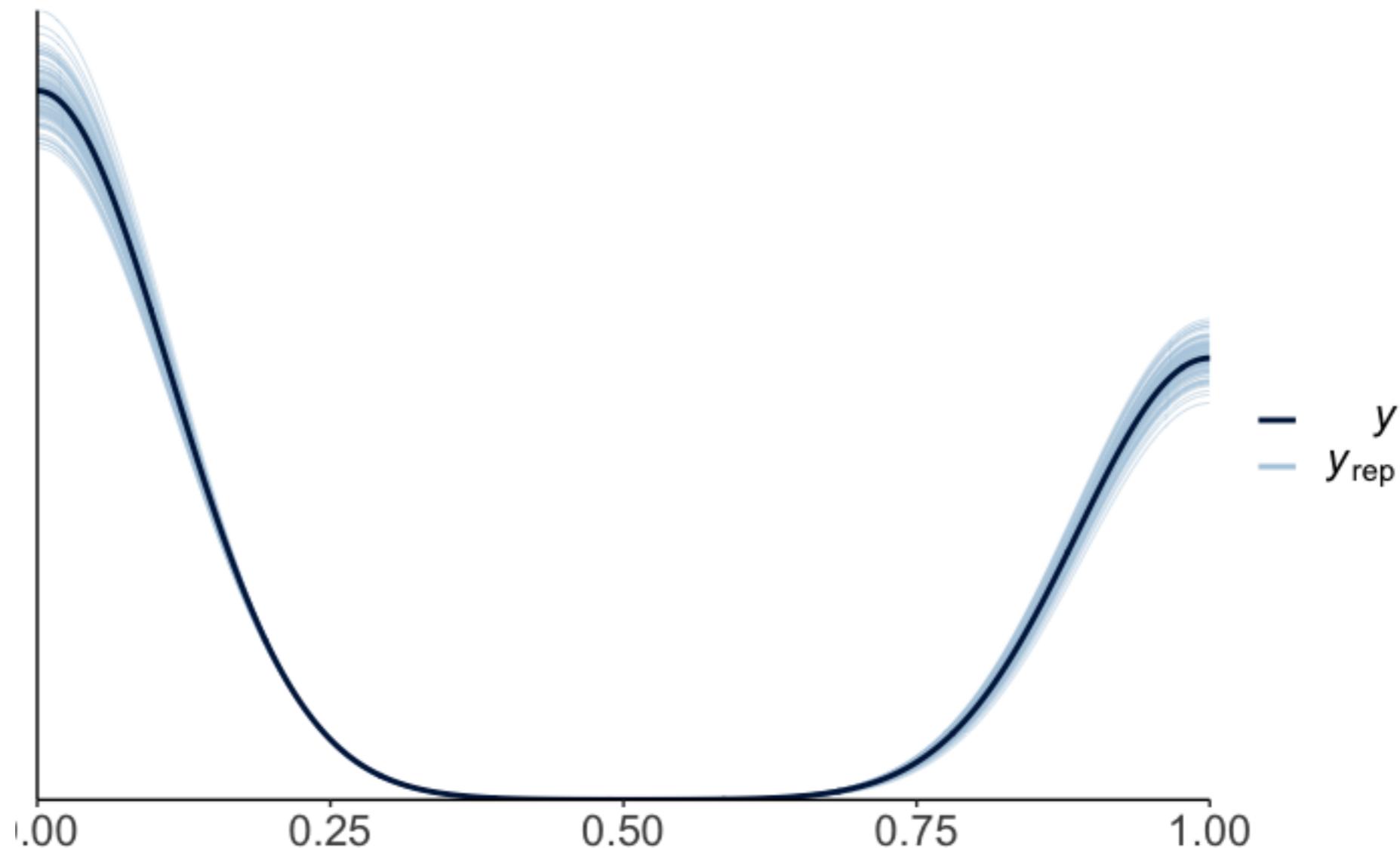
```
1 fit.brms %>%
2   plot()
```



all looking good!

3. Validate the model

```
1 pp_check(fit.brms, nsamples = 100)
```



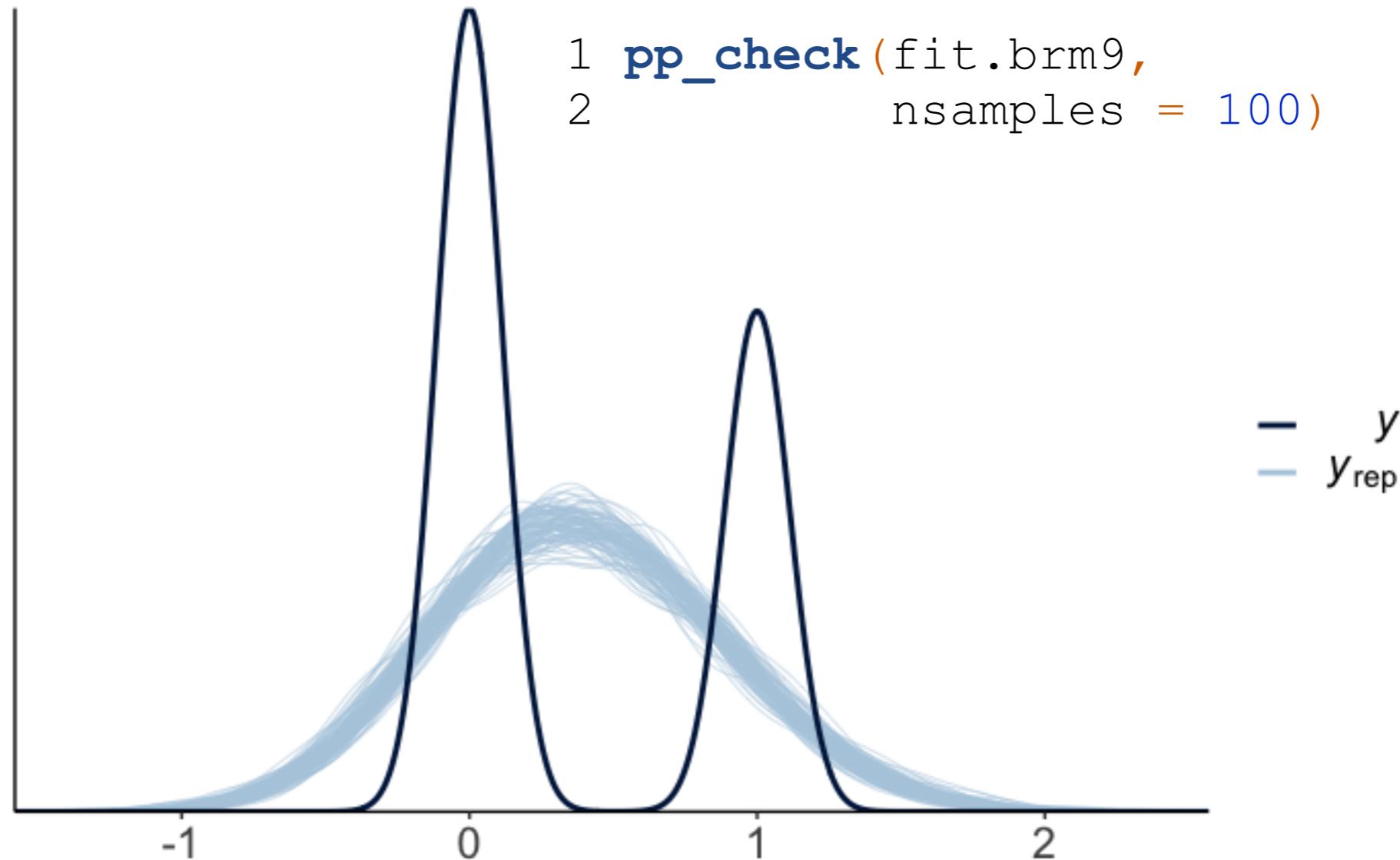
posterior predictive distribution looks good

3. Validate the model

```
1 fit.brm9 = brm(formula = survived ~ 1 + fare + sex,  
2                   data = df.titanic,  
3                   file = "cache/brm9",  
4                   seed = 1)
```

assumes a Gaussian likelihood function!

```
1 pp_check(fit.brm9,  
2            nsamples = 100)
```



posterior predictive distribution it **not** looking good!!

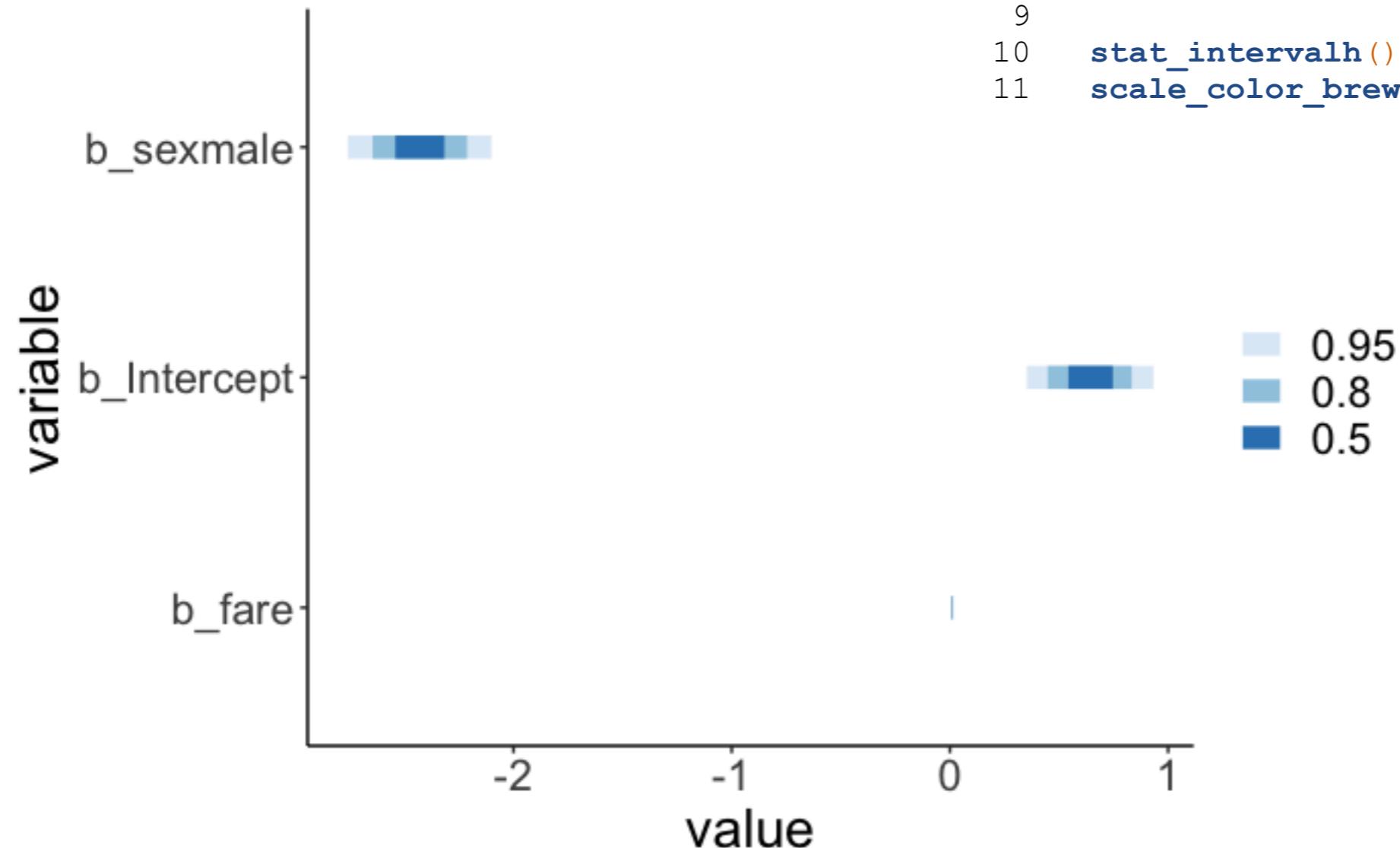
4. Interpret the model parameters

```
1 fit.brms %>%  
2   tidy()
```

term	estimate	std.error	conf.low	conf.high
intercept	0.64	0.15	0.35	0.93
fare	0.01	0.00	0.01	0.02
sex	-2.43	0.17	-2.76	-2.10

$$\ln\left(\frac{\widehat{p(\text{survived})}_i}{1 - \widehat{p(\text{survived})}_i}\right) = b_0 + b_1 \cdot \text{fare}_i + b_2 \cdot \text{sex}_i$$

5. Visualize the results

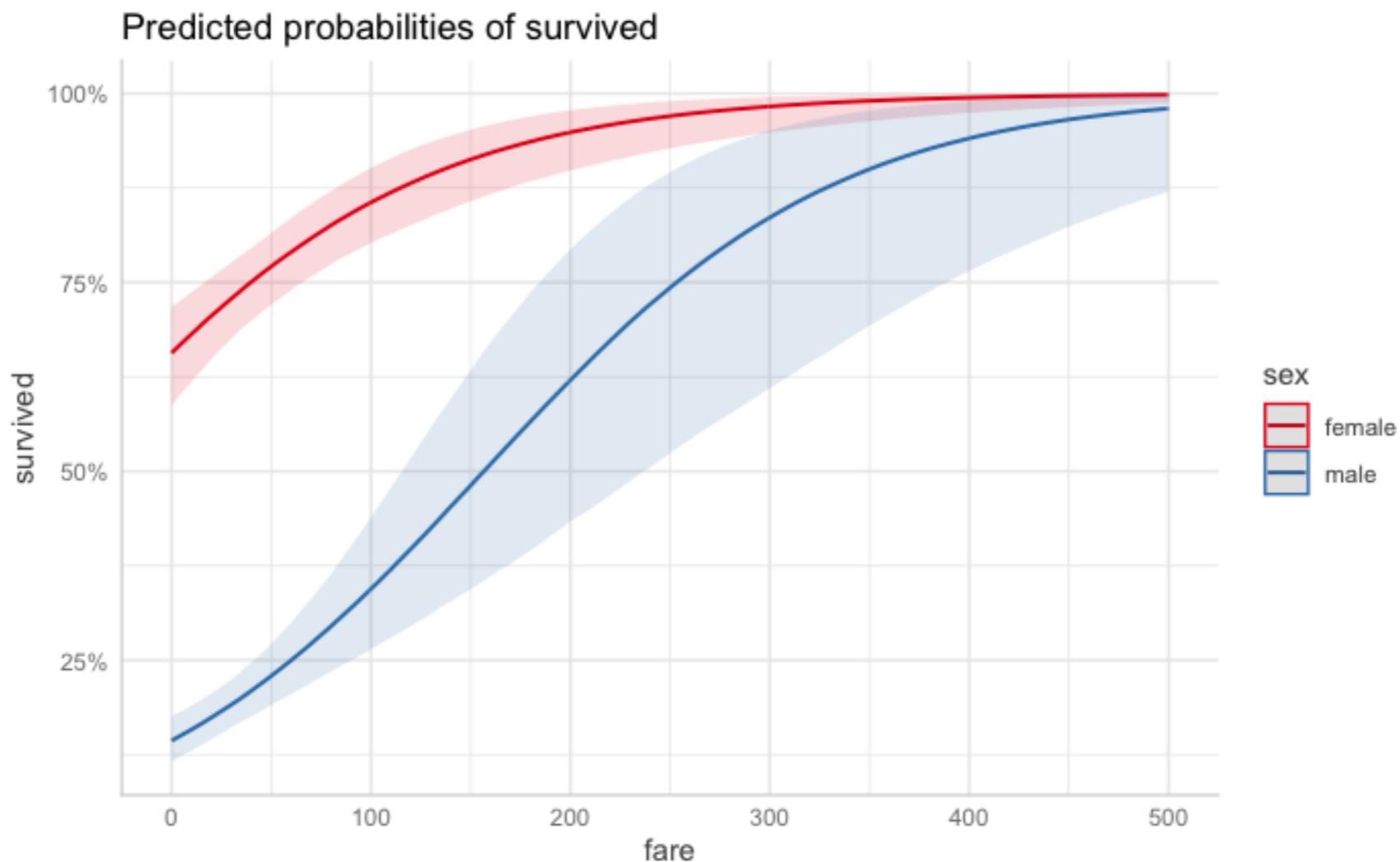


```
1 fit.brm8 %>%
2   posterior_samples() %>%
3   select(-lp__) %>%
4   pivot_longer(cols = everything(),
5                 names_to = "variable",
6                 values_to = "value") %>%
7   ggplot(data = .,
8           mapping = aes(y = variable,
9                             x = value)) +
10  stat_intervalh() +
11  scale_color_brewer()
```

posterior distributions of the different model parameters

5. Visualize the results

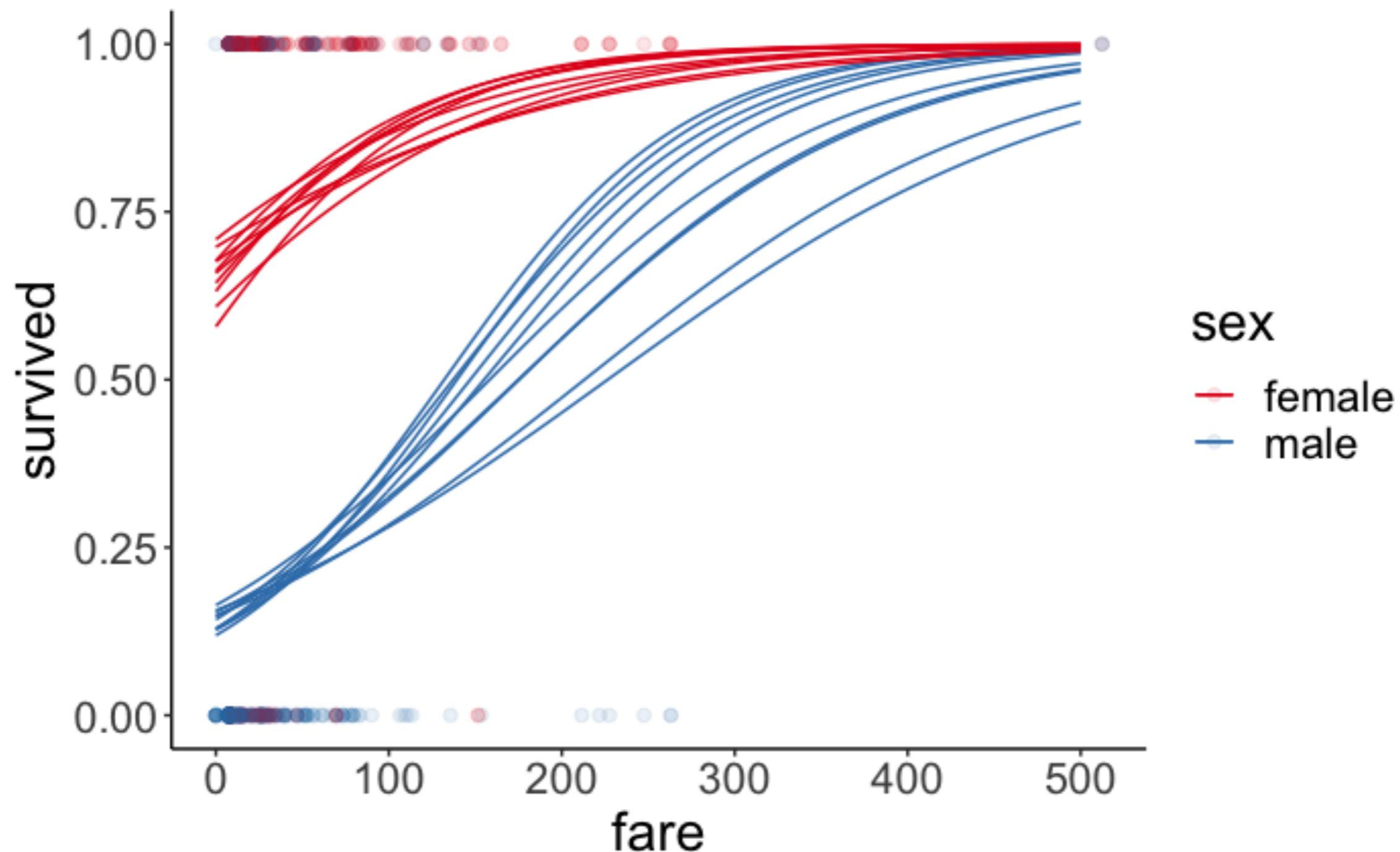
```
1 fit.brms %>%
2   ggpredict(terms = c("fare [0:500]", "sex")) %>%
3   plot()
```



the ggeffects package works with brms!

5. Visualize the results

`add_fitted_draws()`



10 random samples from the posterior distribution

Summary

- Building Bayesian models with `brms`
- Model evaluation
- Testing hypotheses
- Model comparison
- Inference evaluation: Did things work out?
- Reporting results
- Steps for fitting a Bayesian model in `brms`
 - Sleep data
 - Titanic data

Feedback

How was the pace of today's class?

much a little just a little much
too too right too too
slow slow

How happy were you with today's class overall?



What did you like about today's class? What could be improved next time?

Thank you!