

Bayesian data analysis 3

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



We're listening to "All I Need" by
"Quinn DeVeaux"

03/04/2022

Logistics

Homework 7 (the last one, yay)

My name goes here

The names of the people I have worked w

2022-03-03 22:20:30

1 Instructions

This homework is due by **Thursday, March 10th, 8:00pm**.

As per usual, please upload your rendered pdf on Canvas.

Note:

- Some code chunks contain some skeleton code. The code chunk option so that knitting the RMarkdown document doesn't throw any errors if eval=T when you knit your homework, so that your calculations are interpretable, and describe these results from the model.
- Make sure to show the results of your calculations in the knitted pdf function at the end of a code chunk.
- Some questions ask for a short written response as indicated by the question.

Good luck with the homework! If you have any questions, make sure to Thursday and/or post your questions on EdStem.

1.1 Load data

For the logistic regression question, we will use a data set which has info transplant patients.

```
data(heart_transplant, package = "openintro")
df.heart = heart_transplant %>%
  mutate(survived = ifelse(survived == "dead", 0, 1))
```

Here is a description of the data set:

The Stanford University Heart Transplant Study was conducted experimental heart transplant program increased lifespan. Each patient designated officially a heart transplant candidate, meaning that would most likely benefit from a new heart. Then the actual heart few weeks to several months depending on the availability of a donor this waiting period show improvement and get deselected as a heart the purposes of this experiment those patients were kept in the da

1.2 Part 1: Logistic regression (3 points)

Question 1.1: (1 point)

Fit a logistic regression where you predict whether or not a person survived the model summary.

1

YOUR CODE HERE

#####

Question 1.2: (1 point)

Use inverse logit to transform the log odds of the model coefficients into a probability of survival. Make sure the result is interpretable, and describe these results from the model.

YOUR CODE HERE

#####

Your answer:

Question 1.3: (1 point)

Visualize the results of the logistic regression model that you've fitted in Question 2. This question asks you to plot the relationship between age and the probability of survival.

YOUR CODE HERE

#####

Your answer:

1.3 Part 2: Bayesian inference “by hand” (8 points)

This homework is from “PSYCH 10: Introduction to Statistical Methods” which is taught You may find taking a look at this online chapter useful: [Doing Bayesian Estimation](#)

Question 2.1: (2 points)

Let's say that we are interested in the probability that a new drug called Bayesium will cure a patient of frequentitis. For our purposes we are happy to estimate this variable (which we will call theta) using a binomial distribution. We know that the probability of success is the nearest of 0.1.

Create a data frame called bayesium that includes the following variables:

- theta: a vector containing values of theta ranging from 0.0 to 1.0 in steps of 0.1
- flat_prior: a vector representing a flat, uniform prior across all possible values of theta. This should be a probability distribution sum to one.
- bayes_prior: a vector containing a prior based on a binomial distribution with a success probability of 0.1. This reflects the prior of a Bayesian who strongly believes that the treatment will have a positive effect. This can be obtained using the command: `dbinom(seq(0,10), 10, p = 0.9)`
- freq_prior: a vector containing a prior based on a binomial distribution with a success probability of 0.1. This reflects the prior of a frequentist who strongly believes that the treatment will have a positive effect. This can be obtained using the command: `dbinom(seq(0,10), 10, p = 0.1)`
- dogmatic_prior: a vector containing a prior with all of its density on theta = 0.1. This reflects the prior of an very dogmatic Bayesian with very strong beliefs about the success rate of the drug.

YOUR CODE HERE

#####

Question 2.2: (2 points)

Let's say that we perform a clinical trial of the new drug in 20 people and we find that 10 are saved by the drug. Create a variable within the bayesium data frame called likelihood that contains the binomial likelihood for these data given each value of theta. (Hint: use an R function

2

function of the binomial distribution to calculate the number of ‘successes’ or saved lives in a certain number of trials or participants).

Then compute the posterior probabilities for each different prior and add them to the bayesium data frame within the following variables:

- posterior_flat: posterior given flat_prior
- posterior_bayes: posterior given bayes_prior
- posterior_freq: posterior given freq_prior
- posterior_dogmatic: posterior given dogmatic_prior

Each of these should be normalized so that they are probabilities (i.e. they sum to one).

```
df.bayesium = df.bayesium %>%
  ### YOUR CODE HERE ###
  mutate(likelihood = ) %>%
  # compute posterior probabilities for each type of prior, and add to df
  #####
  
```

Question 2.3: (4 points)

For each of the four priors, make a separate figure in which you plot the prior across all values of theta, and include the following: the prior with a black dotted line, the likelihood as a red dashed line, and the posterior with a solid line in blue. Add a title to the plot that includes the name of the prior. You can combine the figures into one using the “patchwork” or “cowplot” library (take a look at notes from the class 03_visualization2). Alternatively, you can also just create one figure and make figure panels using the `facet_wrap()` or `facet_grid()` function.

YOUR CODE HERE

plot priors, likelihood, and posteriors

#####

Then, in your own words, describe how the different priors affect the respective maximum posterior estimates.

Your answer:

1.4 Part 3: Bayesian data analysis (4 points)

Question 3.1: (1 point)

Build a bayesian model from the df.heart data similar to the frequentist model in Question 1.1. Call this model fit.brn

```
### YOUR CODE HERE ###
fit.brn = brm(formula =
  family =
  data =
  file = "cache/brn",
  seed = 1)
#####

  
```

Question 3.2: (1 point)

Print the summary of this model and interpret its coefficients. Compare these to the coefficients from the frequentist model you built in Question 1.1.

3

Final presentation survey

goal: 28

Responses 23

Final presentation

Thanks for filling out this survey to help us with planning!

How are you planning to present? *

- In class (preferred option if possible)
- Remotely (live)
- I will record the presentation and submit a video before March 16th.
- Other...

What's your name (e.g. Tobias Gerstenberg)? *

Short answer text

What's the name of your team's github repository (e.g. final-project-tobi)? *

Short answer text

How many people are in your team (e.g. 1, 2, or 3)?

Short answer text

The screenshot shows a Google Forms survey titled "Final presentation". At the top, there are tabs for "Questions", "Responses" (which is circled in blue), and "Settings". A blue arrow points from the "Responses" tab to the word "goal: 28" at the top right. The survey contains four questions: 1) "How are you planning to present?" with four radio button options. 2) "What's your name (e.g. Tobias Gerstenberg)? *" with a "Short answer text" input field. 3) "What's the name of your team's github repository (e.g. final-project-tobi)? *" with a "Short answer text" input field. 4) "How many people are in your team (e.g. 1, 2, or 3)? " with a "Short answer text" input field. On the right side of the form, there is a vertical toolbar with icons for adding questions, attachments, and other settings.

<https://forms.gle/DdVaig7W63JgBcPC9>

Course evaluations

[Home](#) / Results / Project Response Rates / Node Response Rates / Course Response Rates

Course Response Rates Winter 2021 Course Feedback

Q Search

Course Code	Course Title	Course Unique ID
<input type="text"/>	<input type="text"/>	<input type="text"/>

Search

Courses

Code	Instructor	Enrollments	Responded	Response Rate	Opted-Out	% of Enrollments Opted Out	Responded With Opted-Out Removed	Response Rate With Opted-Out Removed	View
W21-PSYCH-252-01	Tobias Gerstenberg	30	7	23.33%	0	0%	7	23.33%	

Total 1 Records per page

<http://evaluationkit.stanford.edu>

Plan for today

- Quick recap
- Doing Bayesian data analysis **with BRMS**
 - Recipe for Bayesian data analysis
 - Testing hypotheses
 - Model evaluation
 - Reporting results
 - Examples:
 - Sleep data
 - Titanic data
- Going beyond

Quick recap

Quick recap: Doing Bayesian data analysis

Software packages



- let's us write Bayesian models directly in R with a simple syntax
- uses Tensorflow to implement Hamiltonian Monte Carlo sampling (a fast inference algorithm ...)

```
library("greta")
```

Model specification

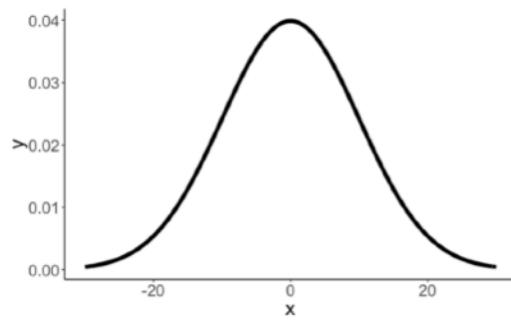
```
1 library("greta")
2 library("tidybayes")
3
4 # variables & priors
5 b0 = normal(0, 10)
6 b1 = normal(0, 10)
7 sd = cauchy(0, 3, truncation = c(0, Inf))
8
9 # linear predictor
10 mu = b0 + b1 * attitude$complaints
11
12 # observation model (likelihood)
13 distribution(attitude$rating) = normal(mu, sd)
14
15 # define the model
16 m = model(b0, b1, sd)
```

Annotations:

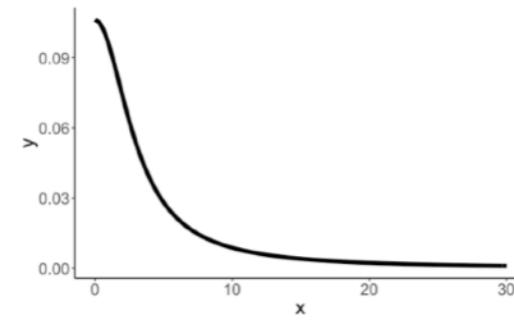
- priors: points to line 5-7
- linear combination: points to line 10
- Gaussian likelihood: points to line 13
- build the model: points to line 16

16

Priors



Gaussian prior on intercept and coefficient

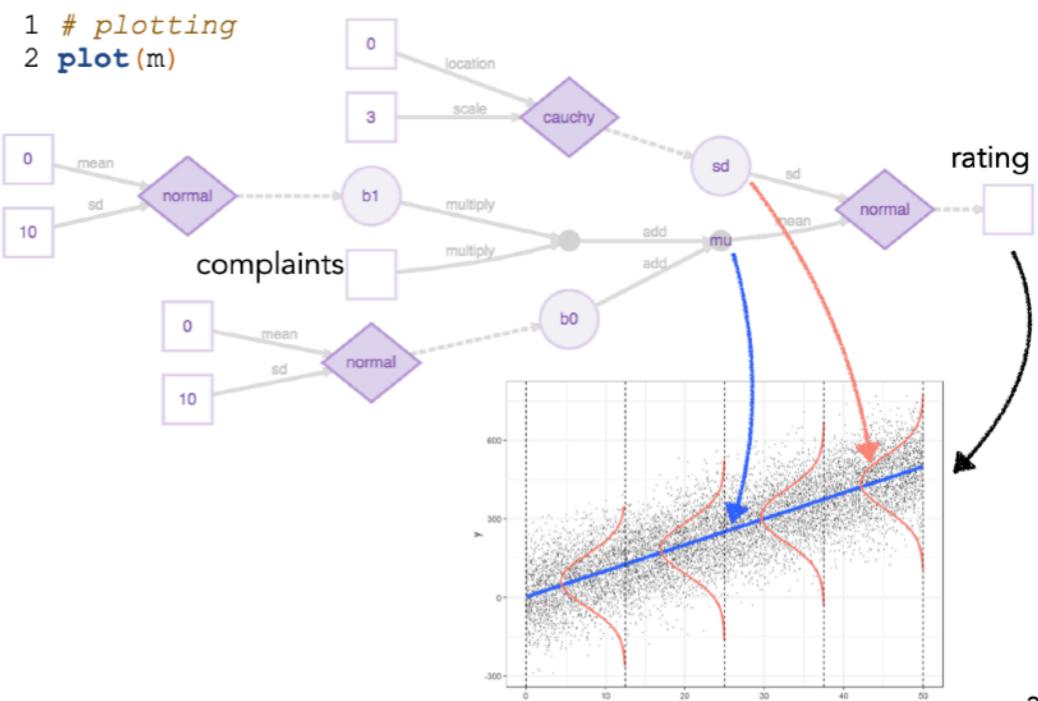


Truncated Cauchy prior on the standard deviation

weakly informative priors (allow for a wide range of possible values)

16

Graphical representation of the model



22

23

8

Quick recap: Doing Bayesian data analysis

Inference via sampling

```
1 # sampling
2 draws = mcmc(m, n_samples = 1000)
3
4 # tidy up the draws
5 df.draws = tidy_draws(draws) %>%
6   clean_names()
```

chain	iteration	draw	b0	b1	sd
1	1	1	6.08	0.87	7.60
1	2	2	1.12	0.95	7.66
1	3	3	-1.83	0.99	7.01
1	4	4	-4.23	1.02	6.64
1	5	5	3.26	0.87	7.96
1	6	6	-1.04	0.98	7.67
1	7	7	-0.83	0.97	10.12
1	8	8	-1.41	0.97	8.02
1	9	9	9.46	0.81	6.30
1	10	10	10.02	0.84	6.57

Markov Chain
Monte Carlo
inference

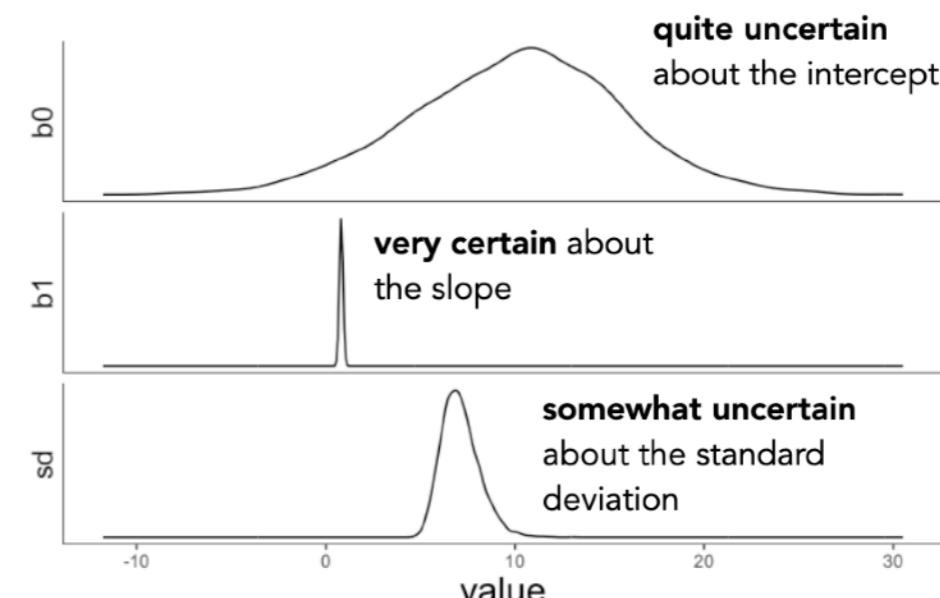
each of these is a solution
for explaining the data

nice visualization of MCMC
samplers

<https://github.com/chi-feng/mcmc-demo>

25

Visualize the posterior

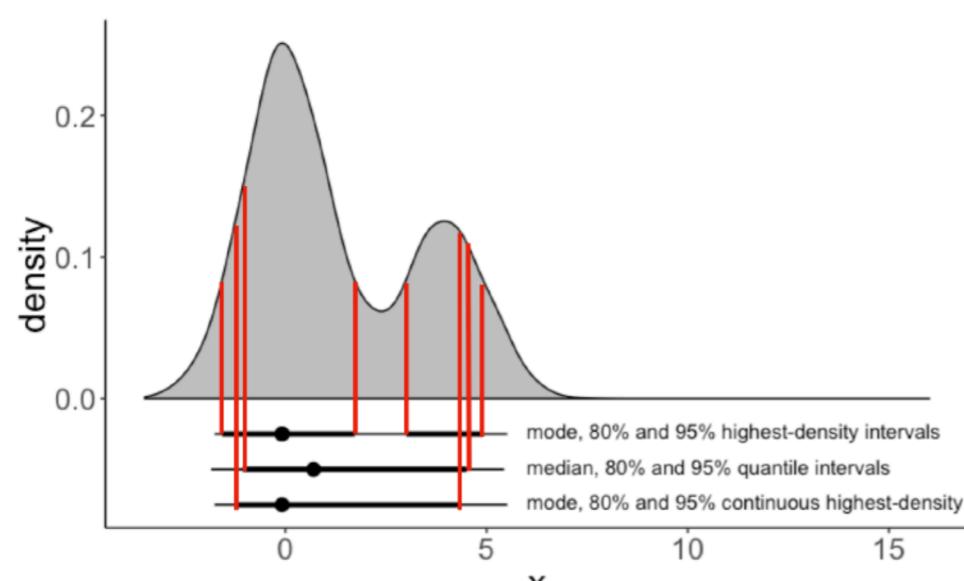


This is the solution of a Bayesian analysis.

A posterior distribution over each parameter in our model.

We can use this to visualize model predictions, and to test hypotheses. 26

Different kinds of credible intervals



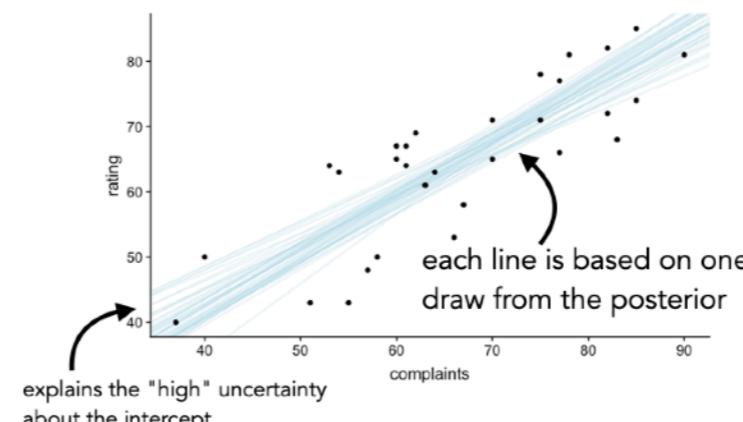
ways of summarizing the posterior distribution

28

Visualize the model predictions

```
1 ggplot(data = df.attitude,
2         mapping = aes(x = complaints,
3                         y = rating)) +
4   geom_abline(data = df.draws %>%
5     sample_n(size = 50),
6     aes(intercept = b0,
7           slope = b1),
8     alpha = 0.3,
9     color = "lightblue") +
10  geom_point()
```

chain	iteration	draw	b0	b1	sd
1	1	1	6.08	0.87	7.60
1	2	2	1.12	0.95	7.66
1	3	3	-1.83	0.99	7.01
1	4	4	-4.23	1.02	6.64
1	5	5	3.26	0.87	7.96
1	6	6	-1.04	0.98	7.67
1	7	7	-0.83	0.97	10.12
1	8	8	-1.41	0.97	8.02
1	9	9	9.46	0.81	6.30
1	10	10	10.02	0.84	6.57



9

29

Quick recap: Doing Bayesian data analysis

Software packages



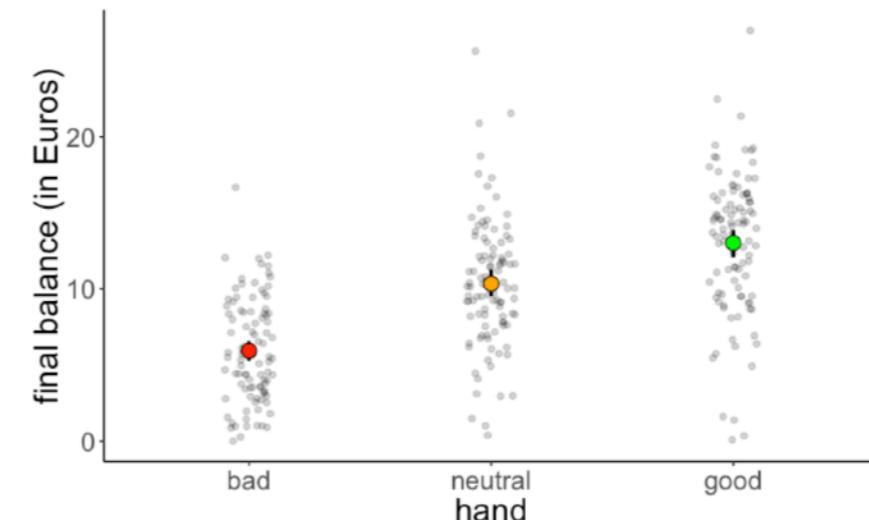
Bayesian regression modeling with Stan

`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

34

Poker data



37

Using `brm()`

cool!

```
1 fit.brm_poker = brm(formula = balance ~ 1 + hand,
2                         data = df.poker)
3
4 fit.brm_poker %>% 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 1-95% CI 2.5%-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 1-95% CI 2.5%-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).
```

39

"Full" specification of the model

```
1 fit.brm_poker_full = brm(
2   formula = balance ~ 1 + hand,
3   family = "gaussian",
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")
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/brm_poker_full", ← save the model result
21   seed = 1               ← make reproducible
22 )
```

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

42

10

Testing hypotheses

Results

posterior samples

b_Intercept	b_handneutral	b_handgood	sigma
5.97	4.27	7.48	3.94
5.11	5.25	7.40	3.91
7.03	3.78	5.80	4.48
5.72	4.18	7.25	4.00
6.01	4.44	6.15	4.57
5.94	4.69	6.72	4.36
6.39	3.84	6.40	3.92
5.24	5.15	7.69	4.16
6.12	4.51	7.20	4.14
6.43	3.71	6.37	4.13
5.85	5.01	7.32	4.00
6.51	3.58	6.62	3.95
5.85	4.45	7.62	4.17
5.80	5.45	6.36	4.10
5.48	5.51	7.22	3.99

:

maximum
a posteriori

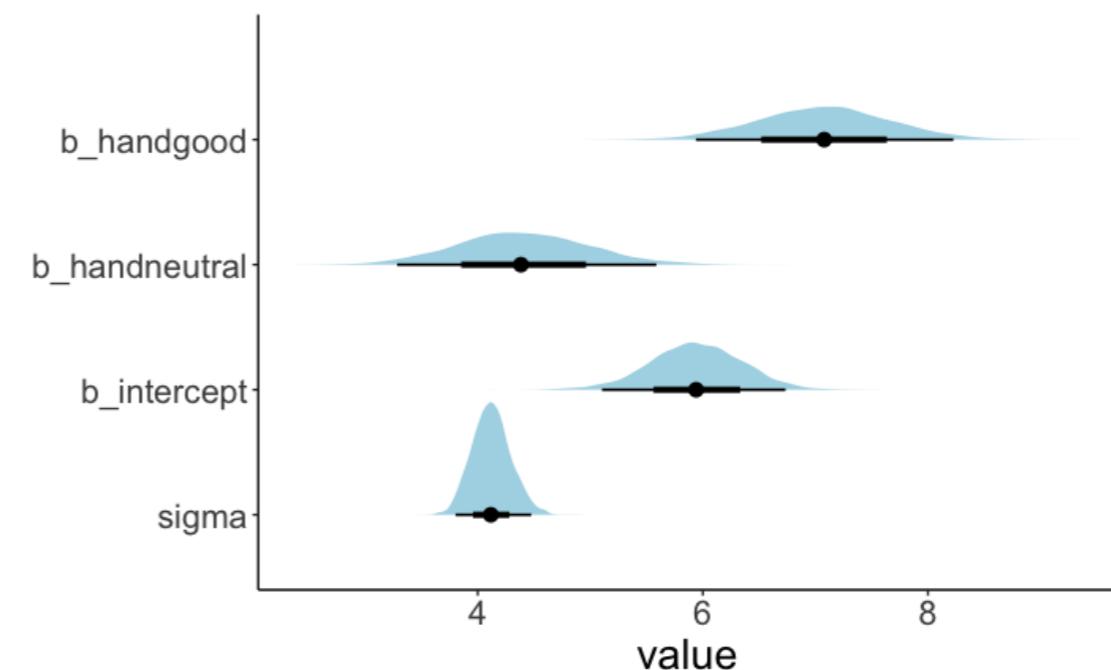


summary of posterior

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

visualization



Nice blog post ...

How to calculate contrasts from a fitted brms model

STATISTICS TUTORIAL R BRMS

Model fitting can take a long time, so it's useful to know how to calculate contrasts from the fitted model object.

AUTHOR

Matti Vuorre 

PUBLISHED

Feb. 5, 2020

AFFILIATION

University of Oxford

CITATION

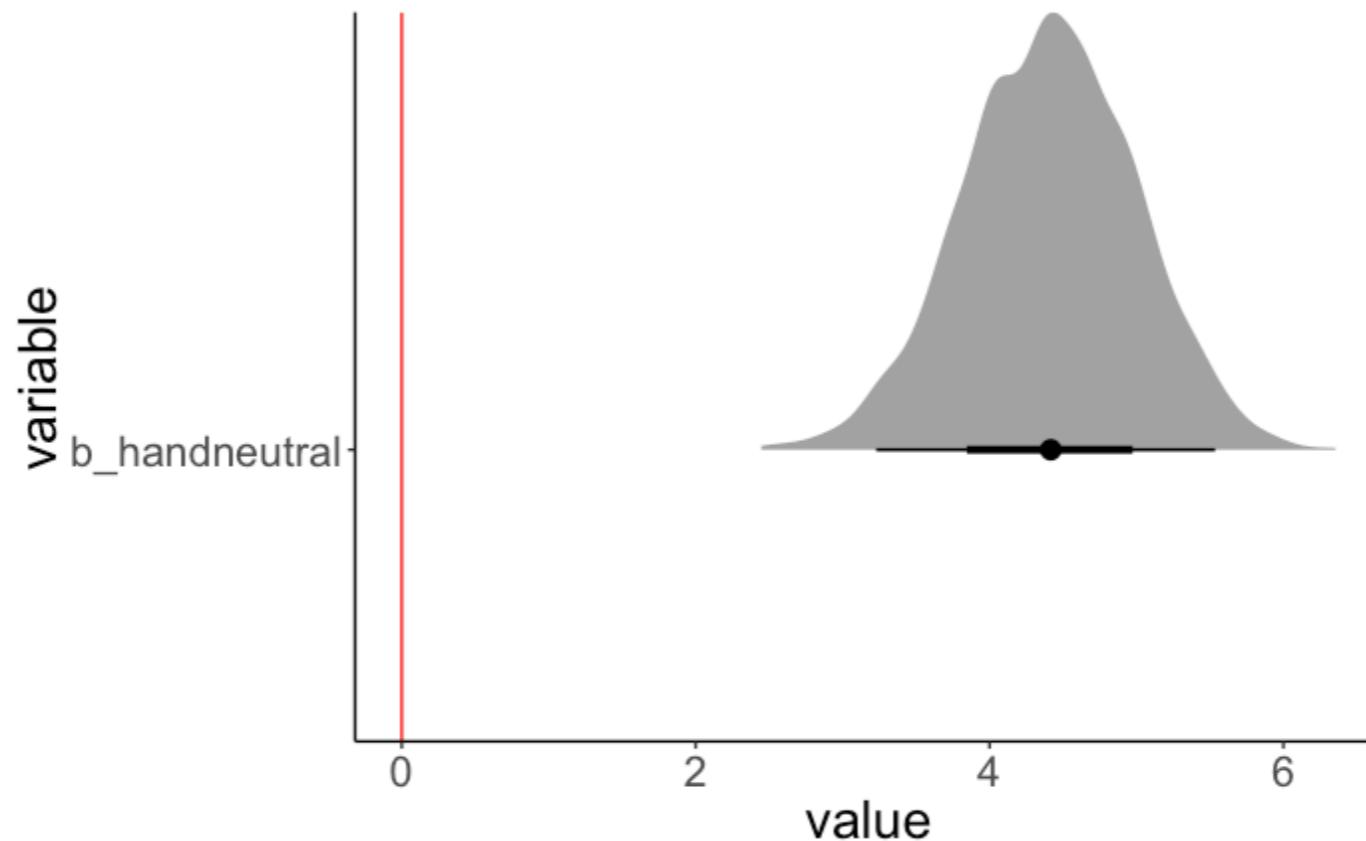
Vuorre, 2020

brms (Bayesian Regression Models using Stan) is an R package that allows fitting complex (multilevel, multivariate, mixture, ...) statistical models with straightforward R modeling syntax, while using Stan for bayesian inference under the hood. You will find many uses of that package on this blog. I am particularly fond of brms' helper functions for post-processing (visualizing, summarizing, etc) the fitted models. In this post, I will show how to calculate and visualize arbitrary contrasts (aka "(general linear) hypothesis tests") with brms, with full uncertainty estimates.

<https://mvuorre.github.io/blog/posts/2020-02-06-how-to-calculate-contrasts-from-a-fitted-brms-model/>

Asking questions based on the posterior

Do neutral hands earn more money than bad hands?



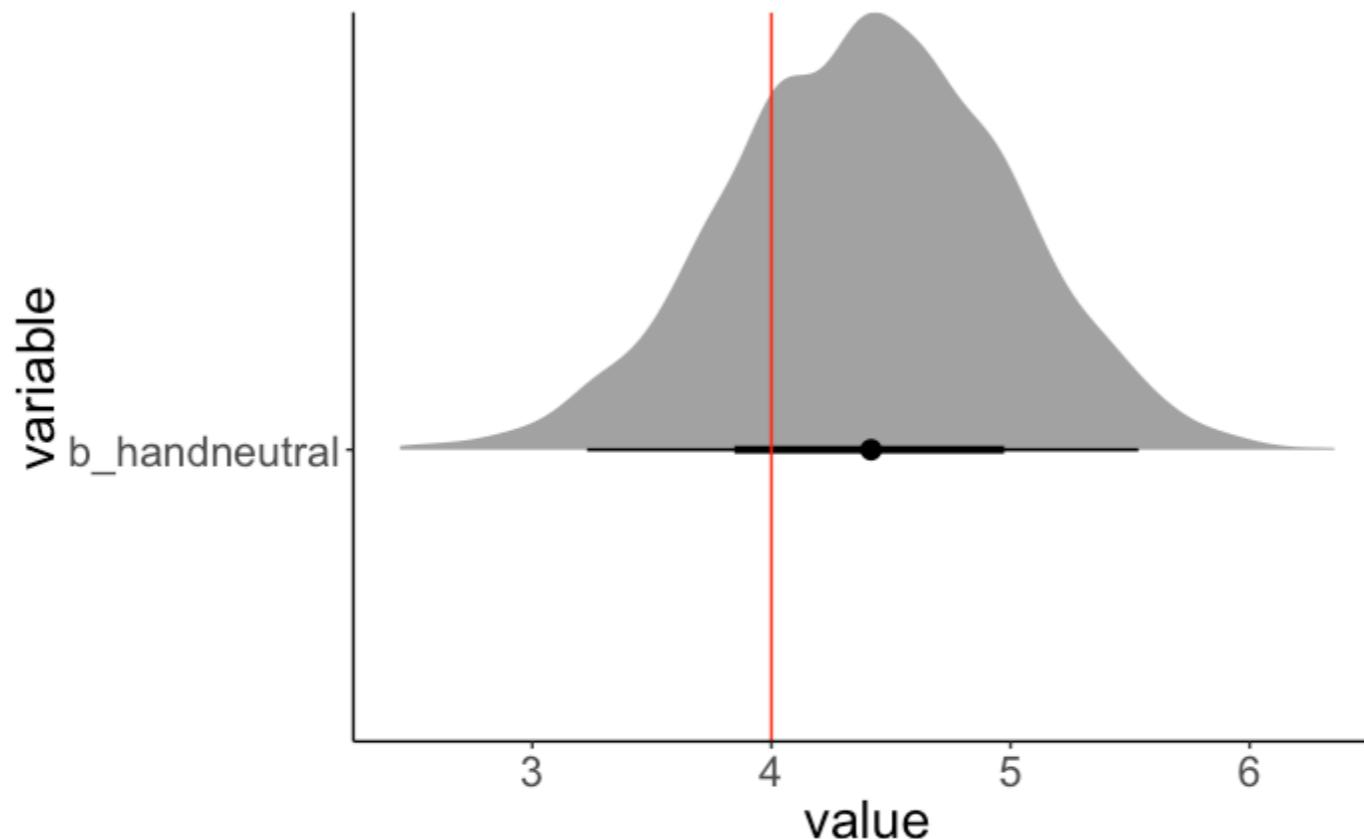
What's the probability that `handneutral` is less than 0?

```
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?



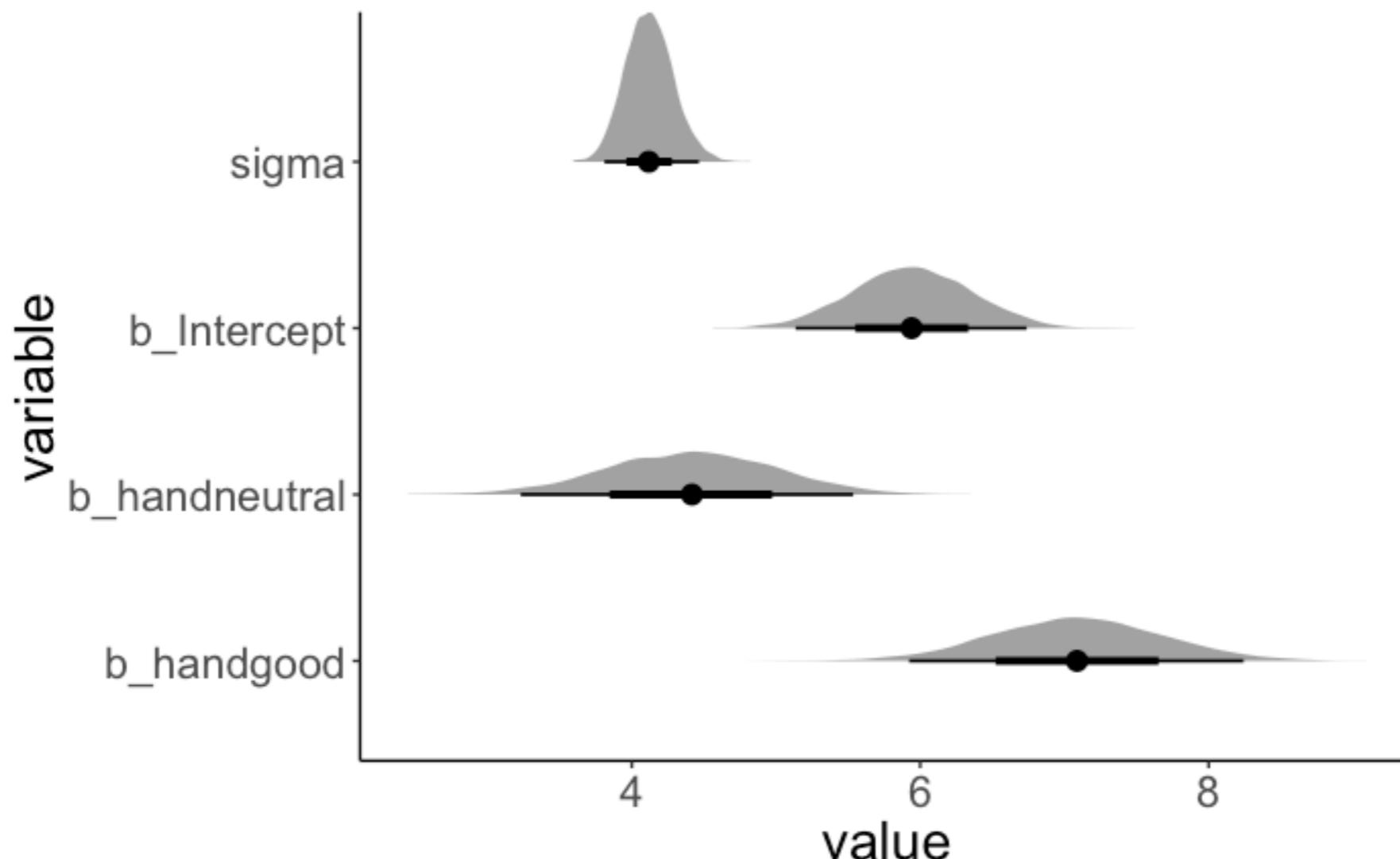
What's the probability that `handneutral` is **more than 4**?

```
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?

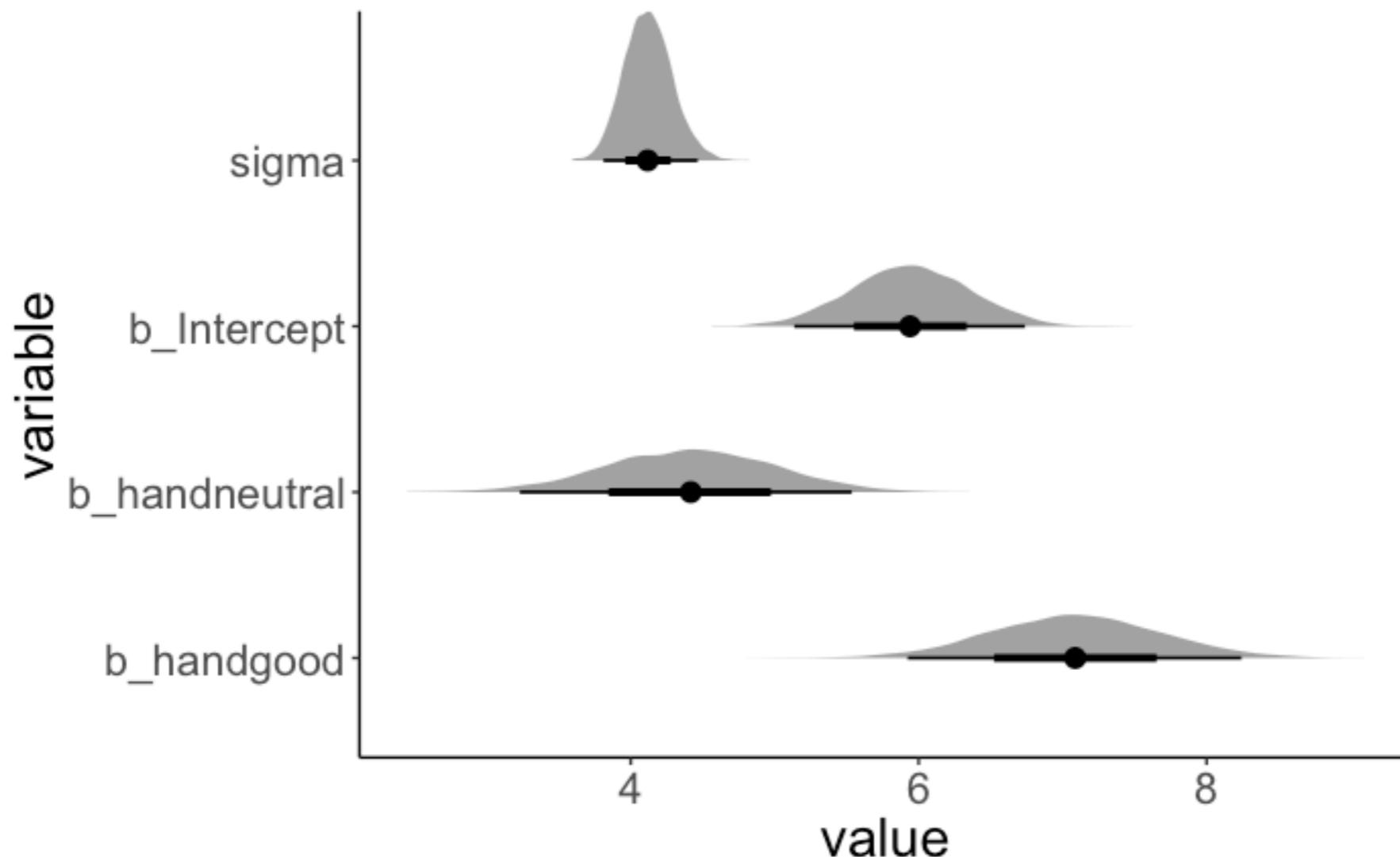


```
1 hypothesis(fit.brm,  
2   hypothesis = "handgood + Intercept > 2 * Intercept")
```

$p = 0.89$

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!

The "emmeans" package is your friend!

```
1 fit.brm_poker %>%
2   emmeans(specs = consec ~ hand)
```

estimated
mean for
each group

contrasts →

\$emmeans				
hand	emmean	lower.HPD	upper.HPD	
bad	5.94	5.16	6.78	
neutral	10.34	9.55	11.15	
good	13.02	12.22	13.82	

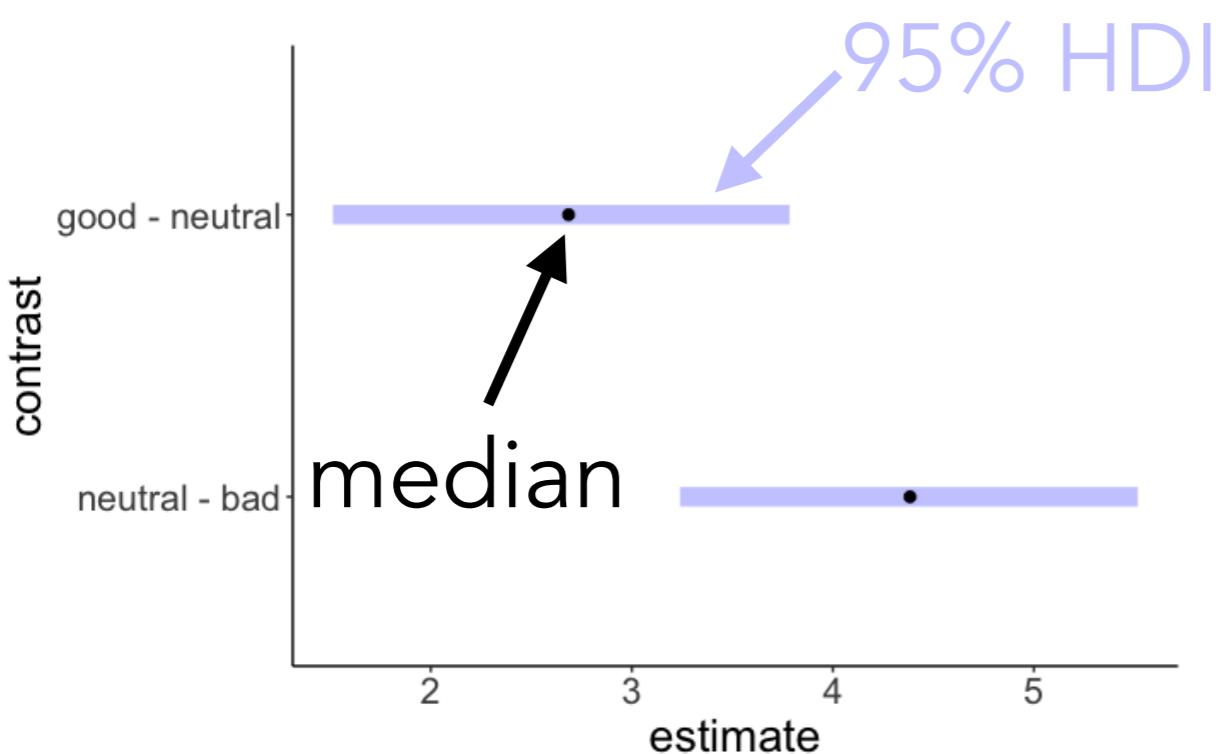
Point estimate displayed: median
HPD interval probability: 0.95

\$contrasts				
contrast	estimate	lower.HPD	upper.HPD	
neutral - bad	4.38	3.24	5.52	
good - neutral	2.69	1.51	3.78	

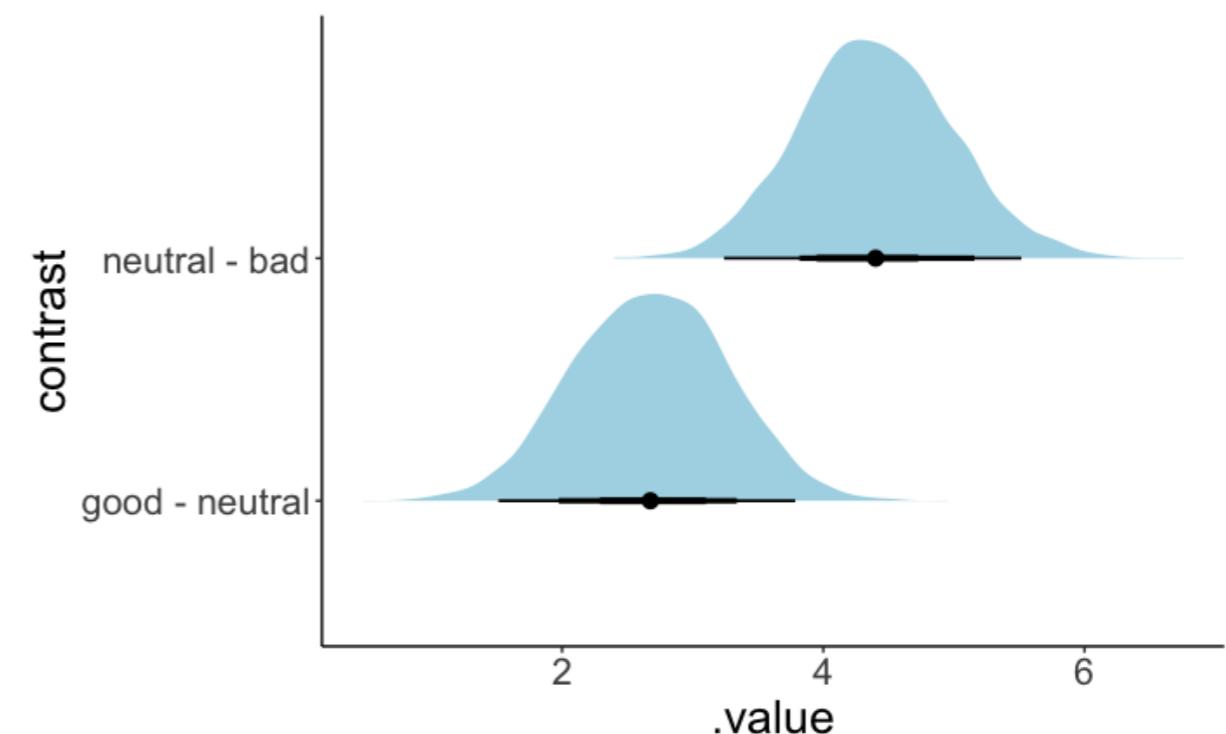
Point estimate displayed: median
HPD interval probability: 0.95

Visualizing the contrasts

```
1 fit.brm_poker %>%
2   emmeans(specs = consec ~ hand) %>%
3   pluck("contrasts") %>%
4   plot()
```



```
1 fit.brm_poker %>%
2   emmeans(specs = consec ~ hand) %>%
3   pluck("contrasts") %>%
4   gather_emmeans_draws() %>%
5   ggplot(mapping = aes(y = contrast,
6                         x = .value)) +
7   stat_halfeye(fill = "lightblue",
8               point_interval = mean_hdi,
9               .width = c(0.5, 0.75, 0.95))
```



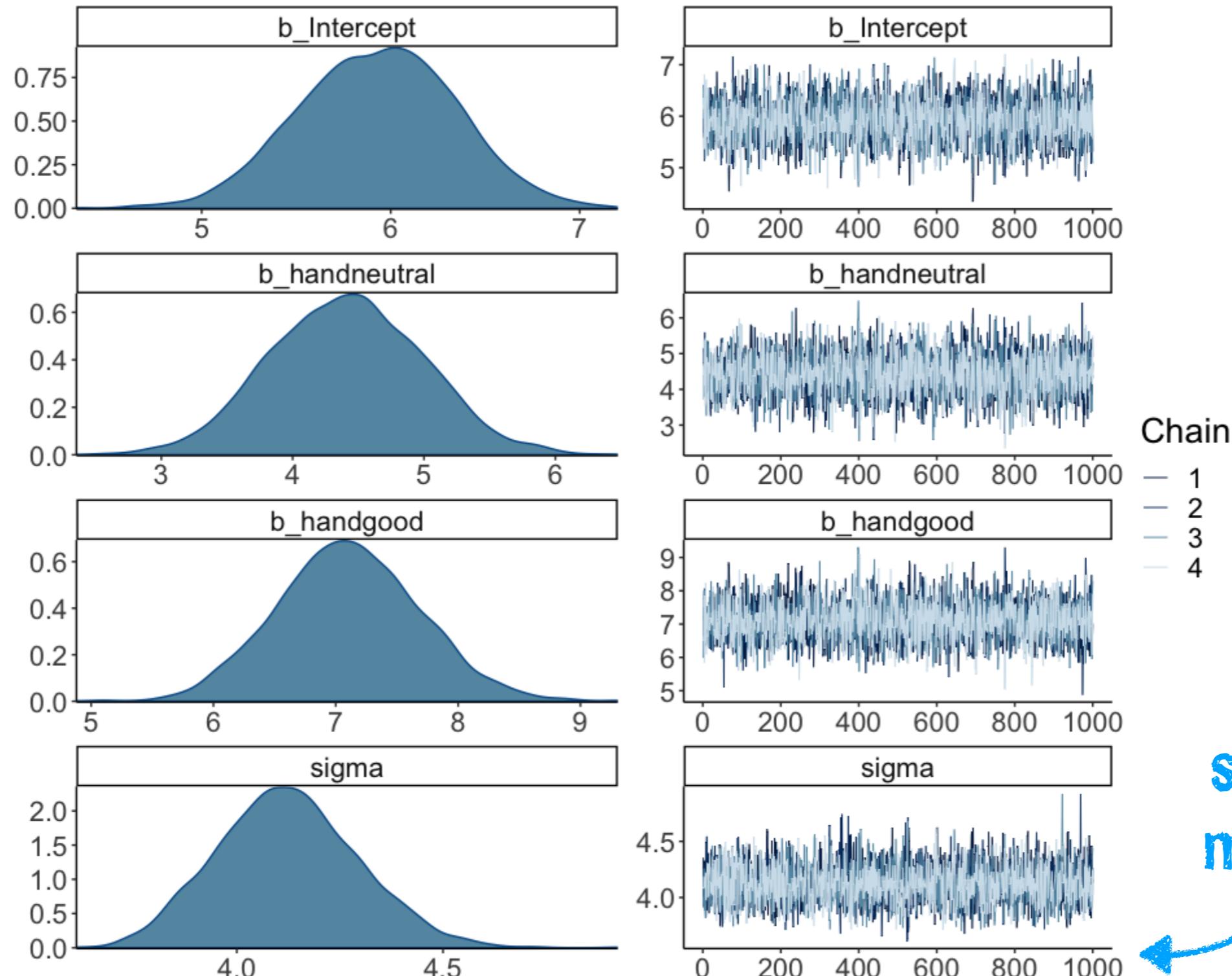
mean, 50% HDI, 75% HDI, 95% HDI

Model evaluation

1. Check whether inference worked

Can we trust the model results?

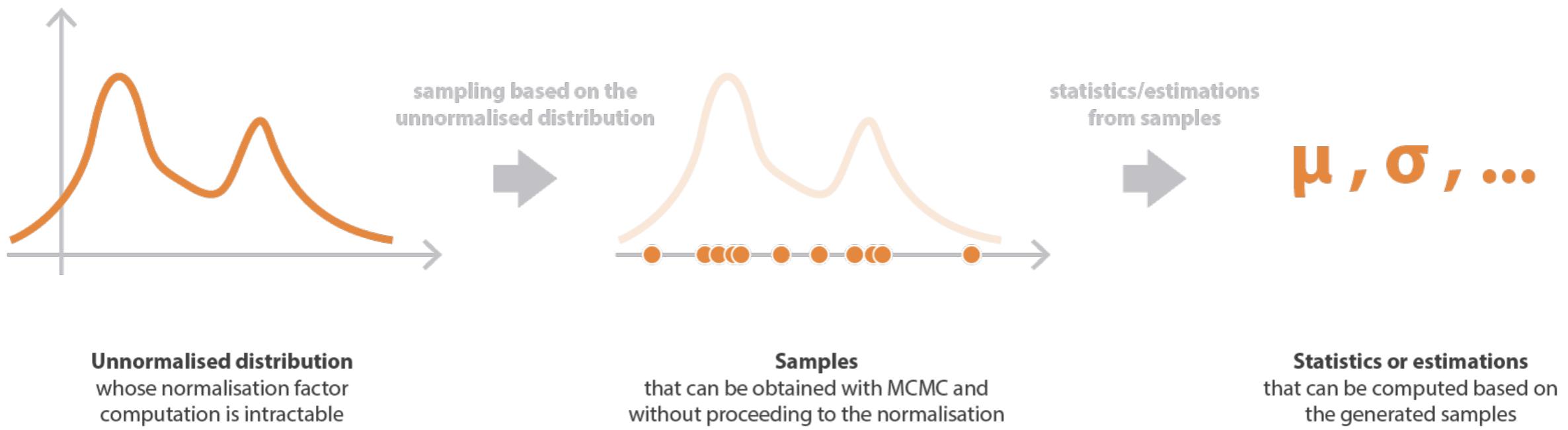
`plot(fit.brm_poker)`



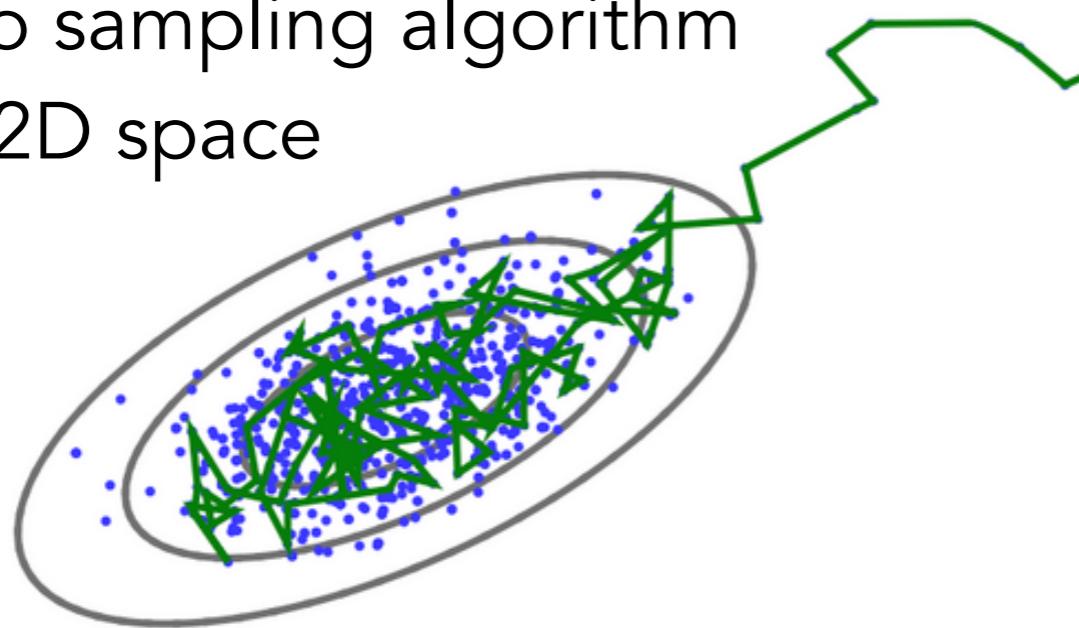
sample
number

Can we trust the model results?

Inference via Markov Chain Monte Carlo (MCMC)



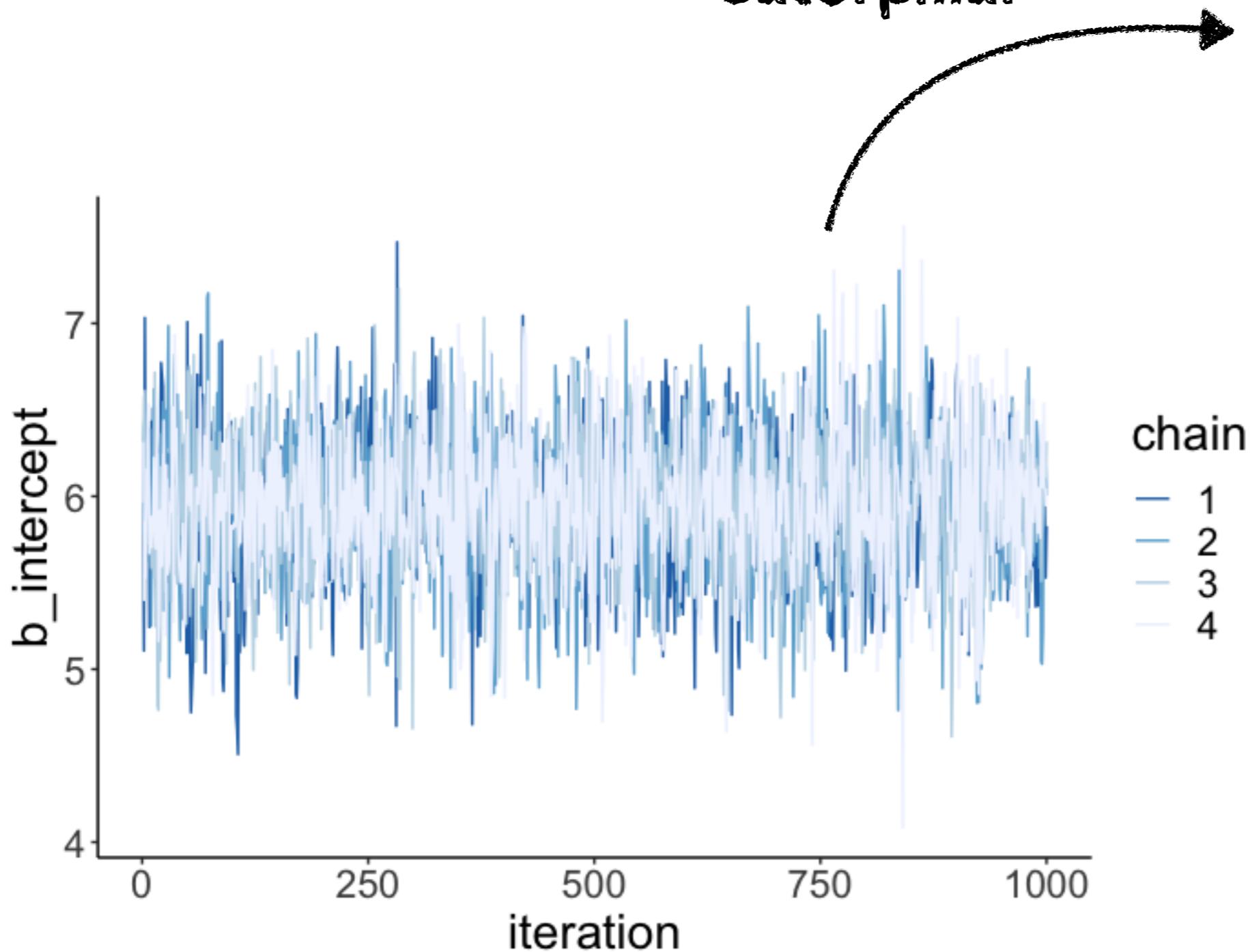
Markov Chain Monte
Carlo sampling algorithm
in a 2D space



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

Can we trust the model results?

looks like a fuzzy caterpillar



Stats twitter chimes in ...



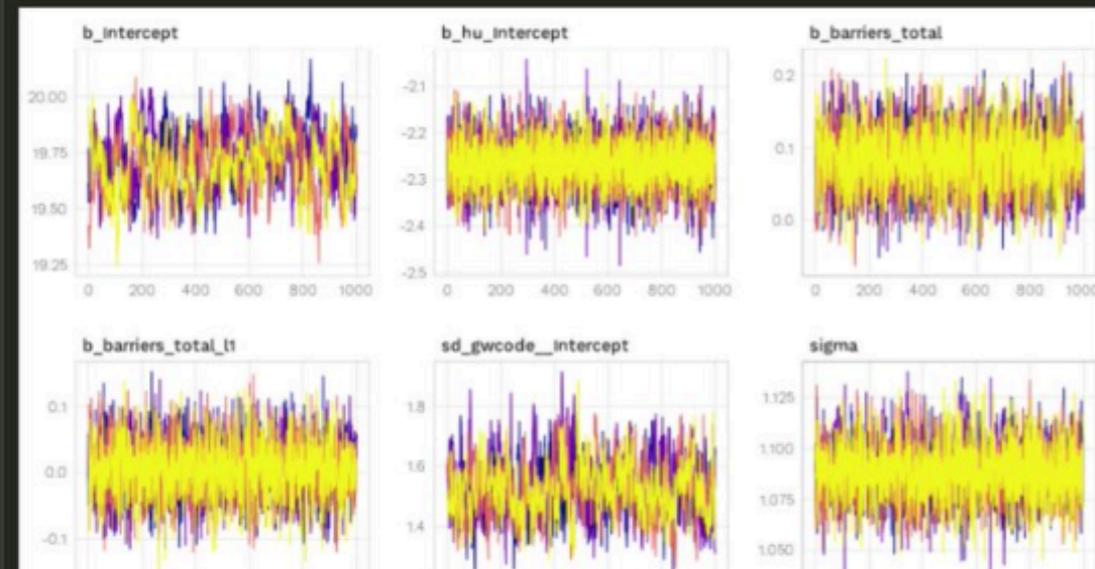
Andrew Heiss
@andrewheiss

...

love that "looking for fuzzy caterpillars" is like a legitimate analytical strategy

Check for fuzzy caterpillars:

```
```{r}
aid_hu_fit %>%
 posterior_samples(add_chain = TRUE) %>%
 select(-starts_with("r_gwcode"), -lp__, -iter) %>%
 mcmc_trace() +
 theme_donors()
```



...

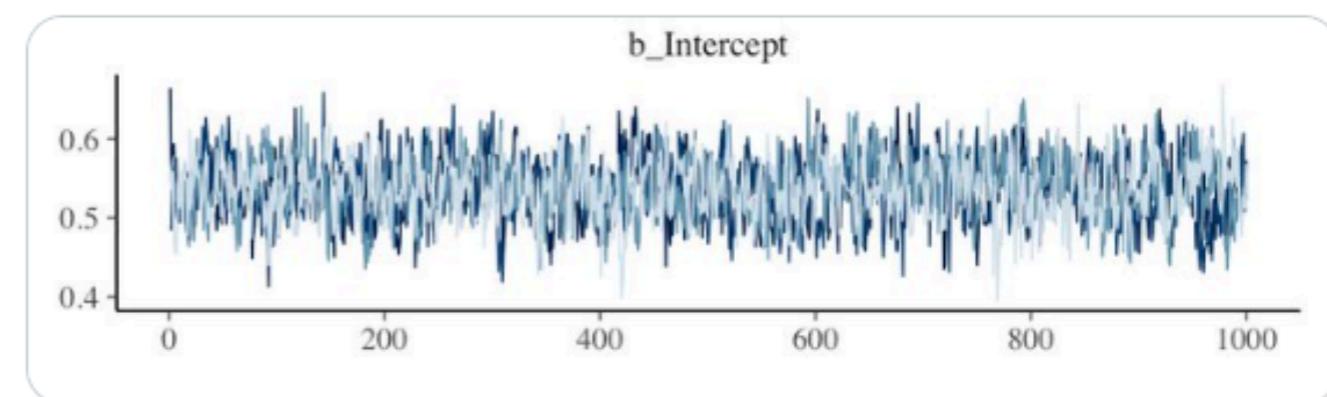


Chelsea Parlett-Pelleriti  
@ChelseaParlett

...

Do your chains just flow?  
Do they sample to and fro?  
Do they mix together well?  
Is your R-hat small, or no?

Are your trace plots looking killer,  
like a fuzzy caterpillar?  
Do your chains just flow?



# When things don't work out

```
1 df.data = tibble(y = c(-1, 1))
2
3 fit.brm_wrong = 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/brm_wrong")
```

**only two data points!**

**incredibly wide uniform priors**

**10000000000**

# When things don't work out

**summary(fit.brn\_wrong)**

```
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

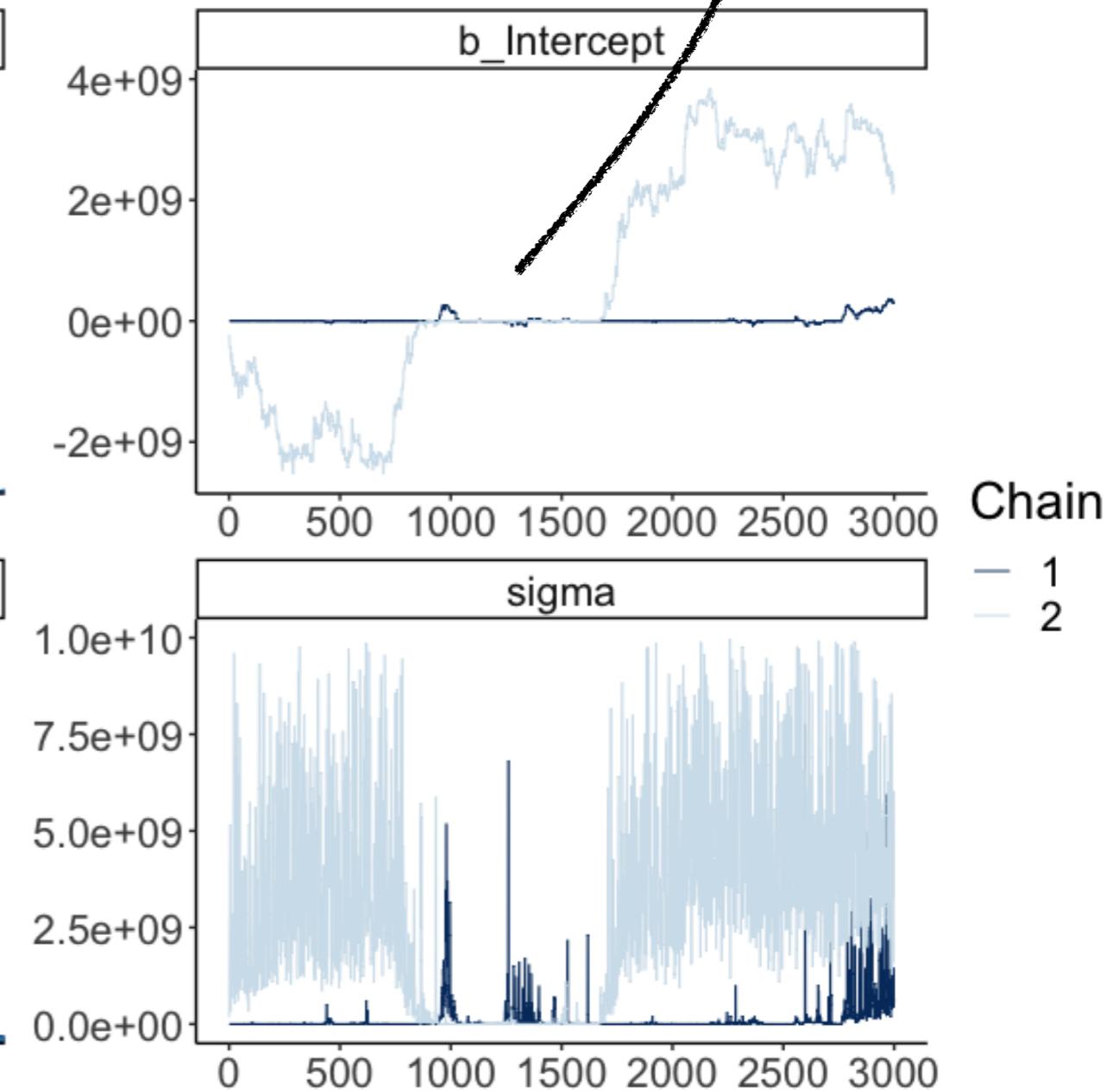
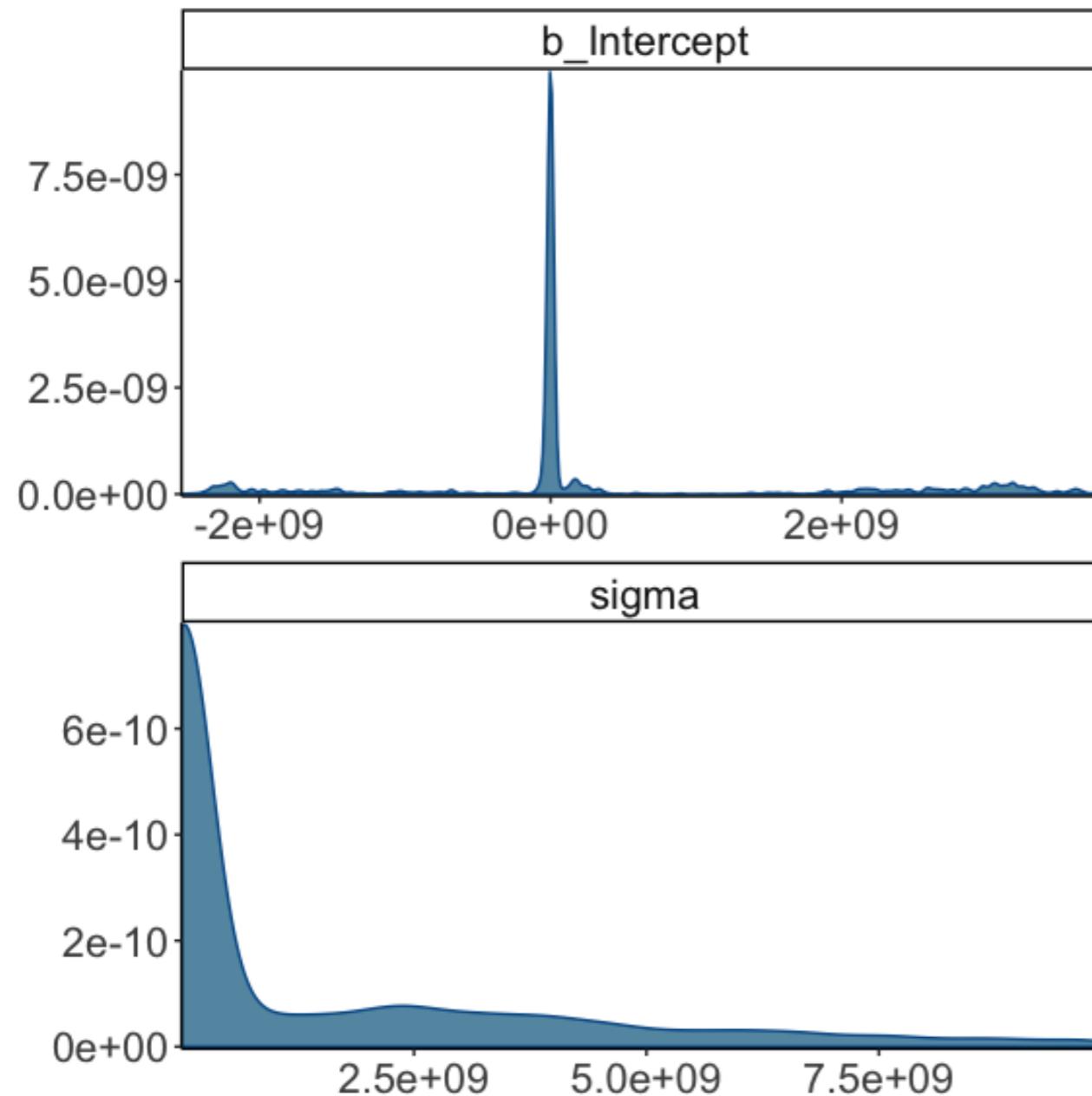
Population-Level Effects:
Estimate Est.Error 1-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept 357550121.58 1416057299.71 -2244033111.47 3333594132.43 1.78 3 24

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



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:

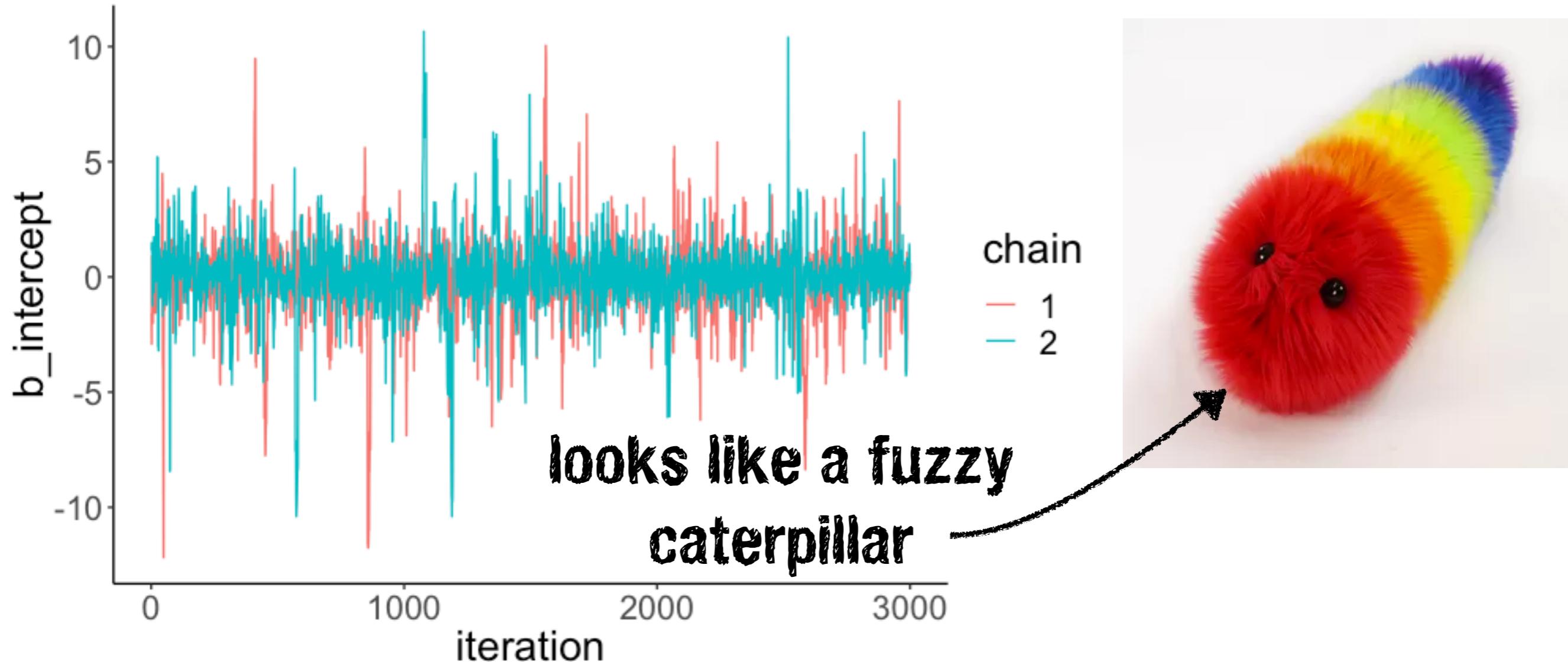
	Estimate	Est.Error	l-95%	CI	u-95%	CI	Eff.Sample	Rhat
Intercept	-0.06	1.72	-3.78	3.27			1033	1.00

#### Family Specific Parameters:

	Estimate	Est.Error	l-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



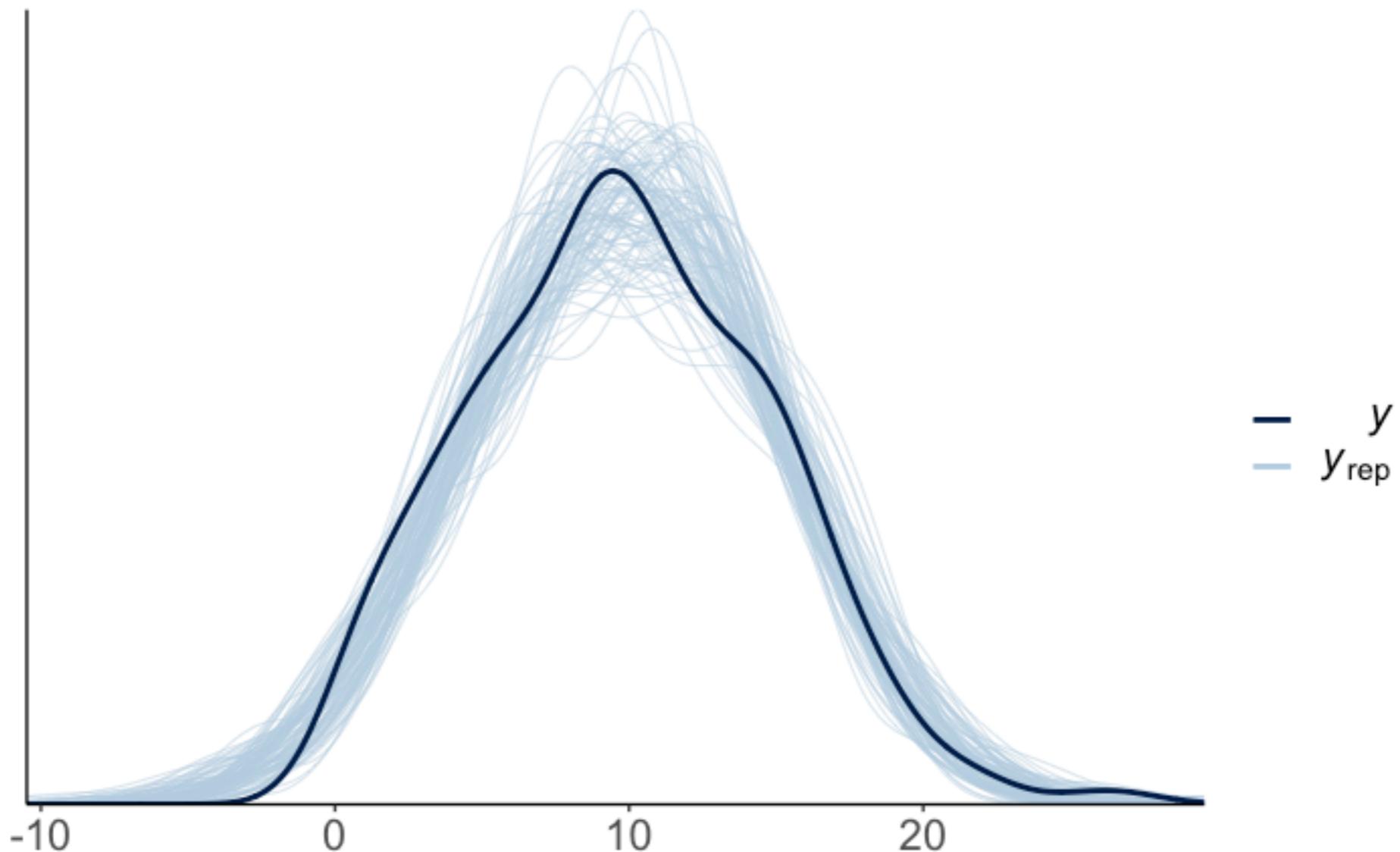
**if things go wrong:**

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

## **2. Visualize model predictions**

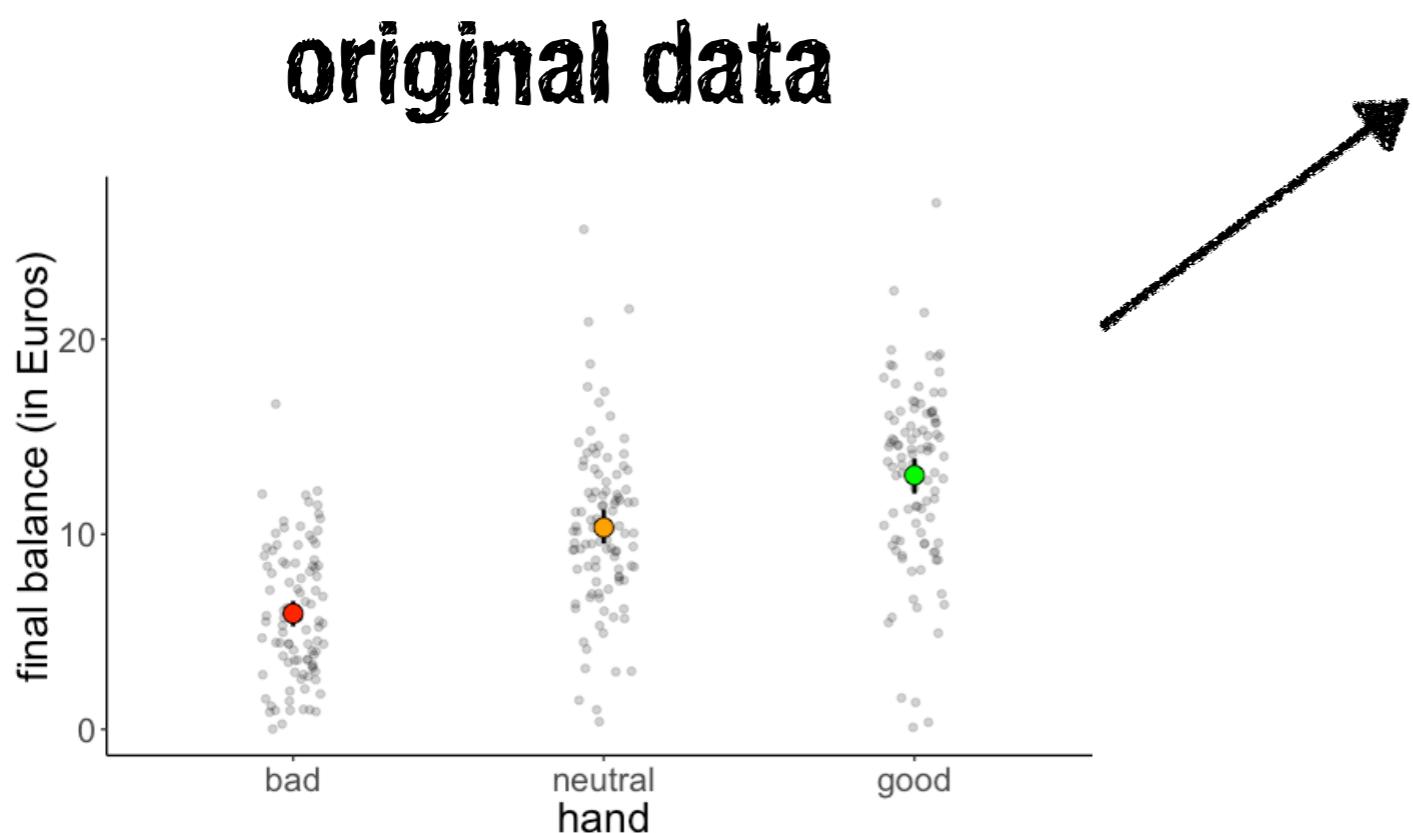
# Posterior predictive check

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

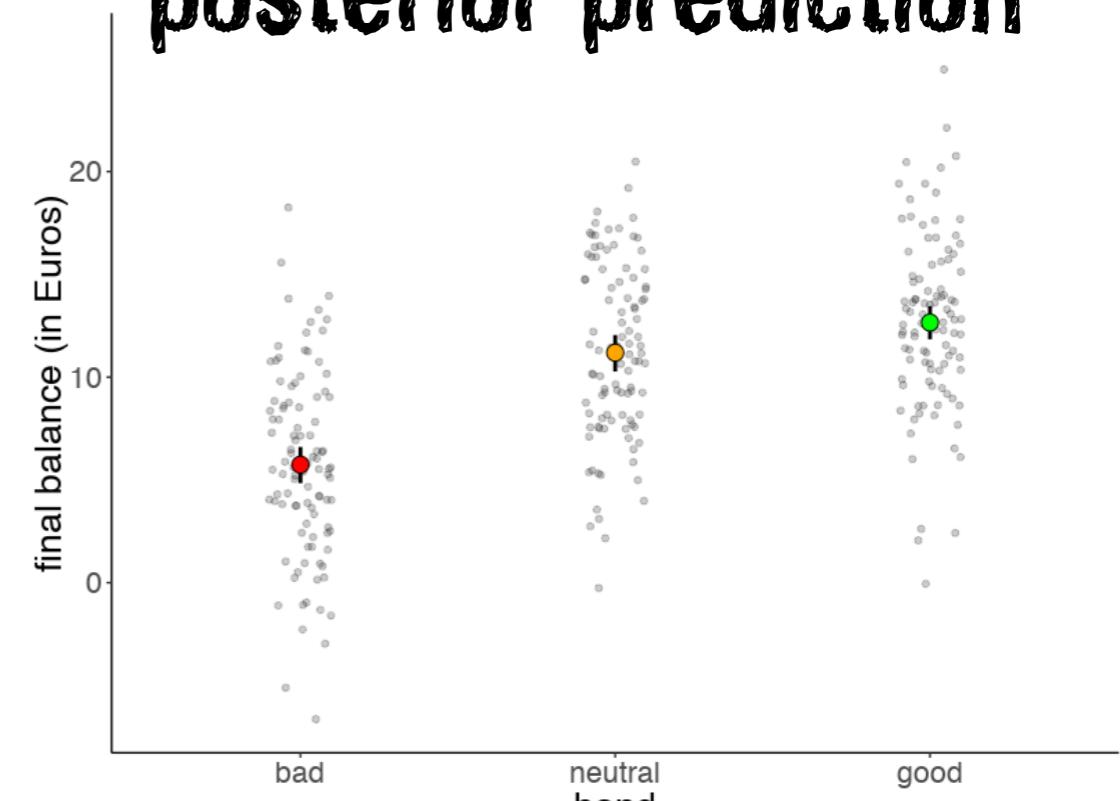


The model accurately captures the distribution of the response variable

# Posterior predictive check



# posterior prediction

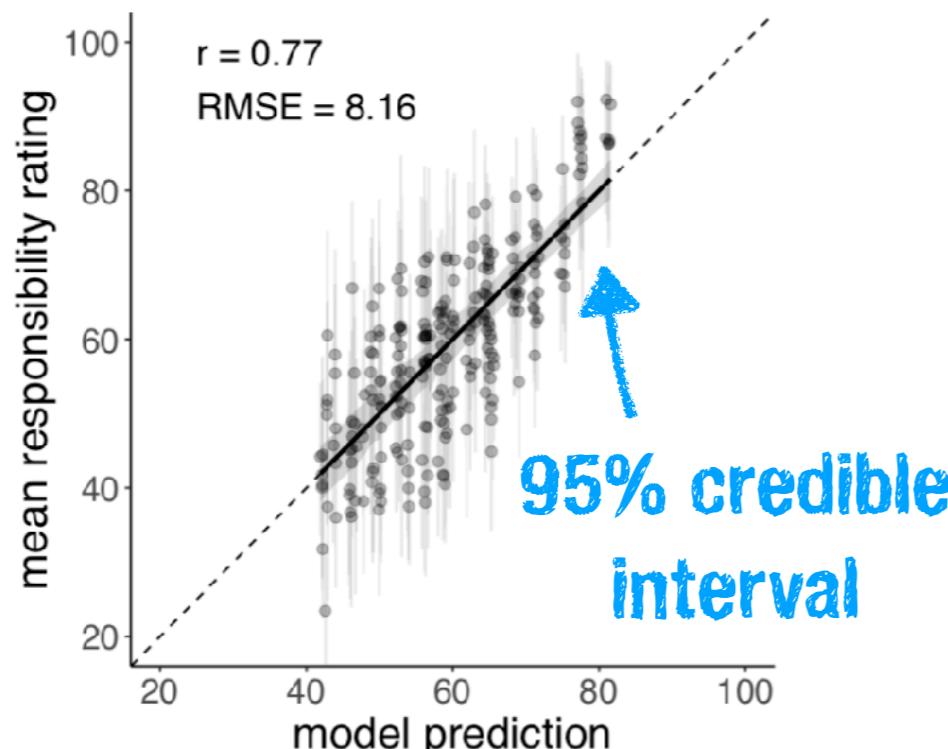


take a look at course notes  
for how to make these

# **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.*

$\text{responsibility} \sim 1 + \text{surprise} + \text{pivotality} + \text{n\_causes} + (1 + \text{surprise} + \text{pivotality} + \text{n\_causes} | \text{participant})$

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

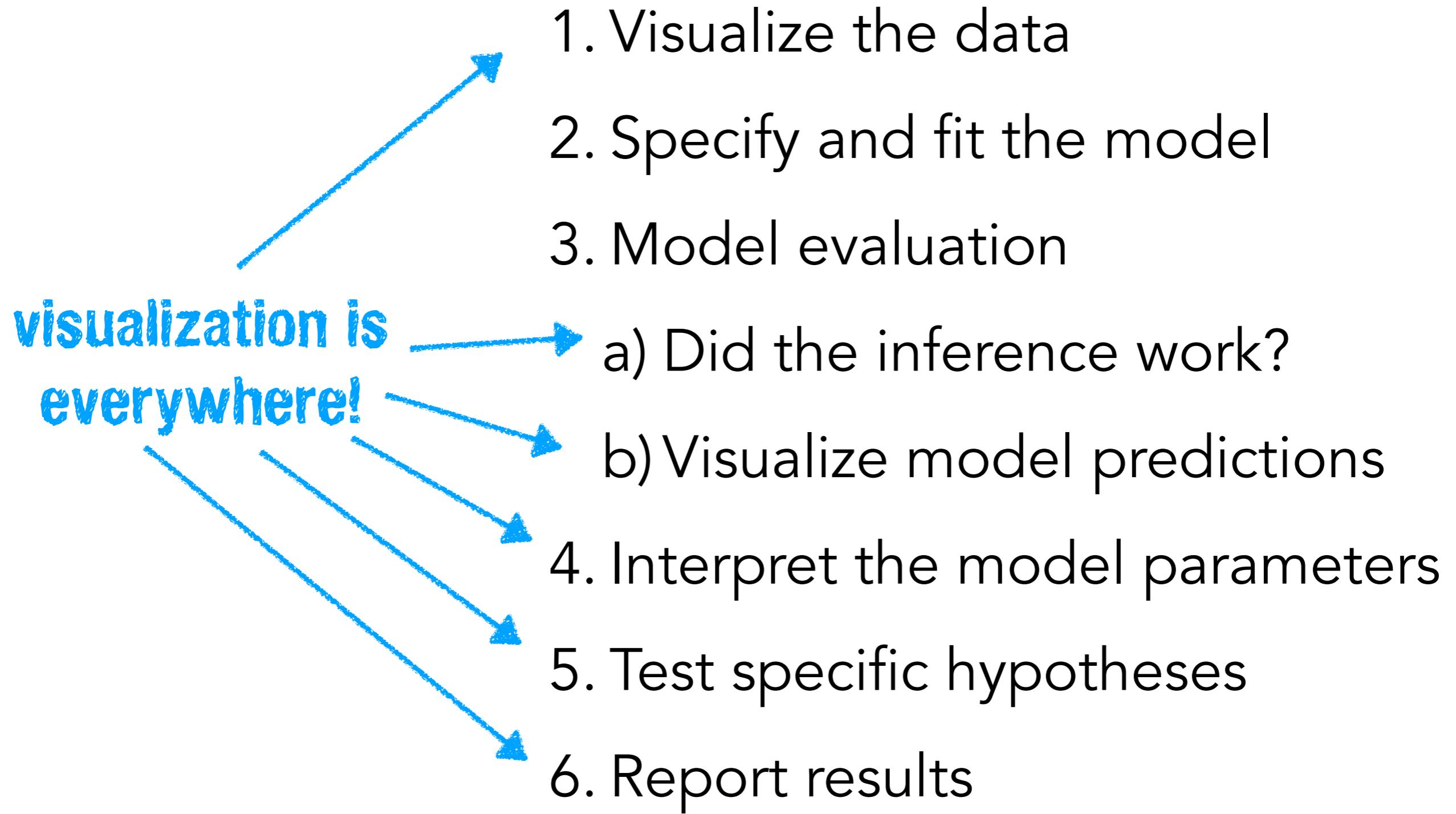
parameter  
estimates

## 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.<sup>1</sup>

<sup>1</sup>For 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).

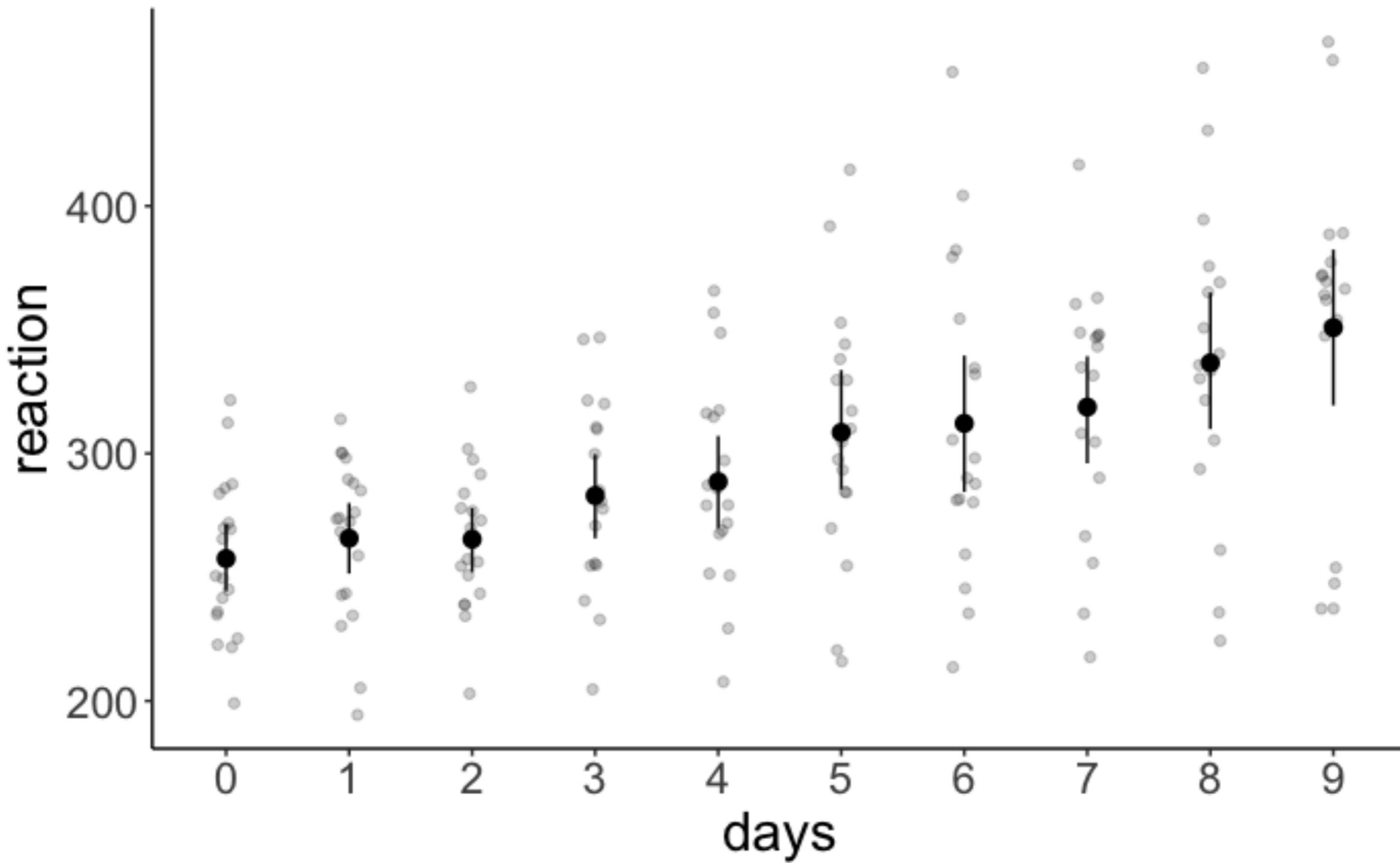
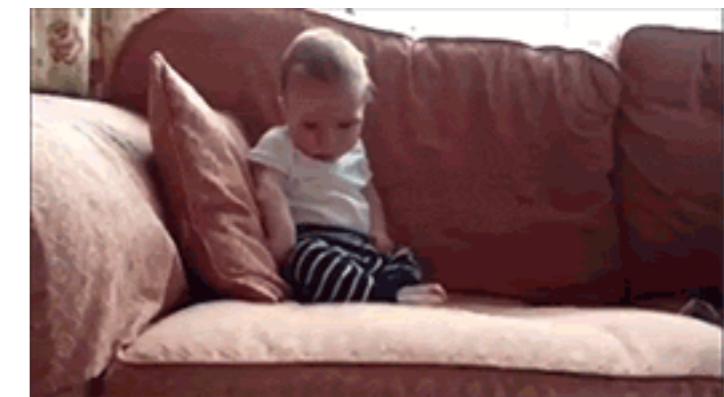
# Recipe for Bayesian analysis with brms



# Sleep data

# **1. Visualize the data**

# Feeling sleepy?



## **2. Specify and fit the model**

# 1. Specify and fit the model

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



# **3. Model evaluation**

# a) Did the inference work?

```
1 fit.brm_sleep %>%
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	1-95% 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	1-95% 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	1-95% 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

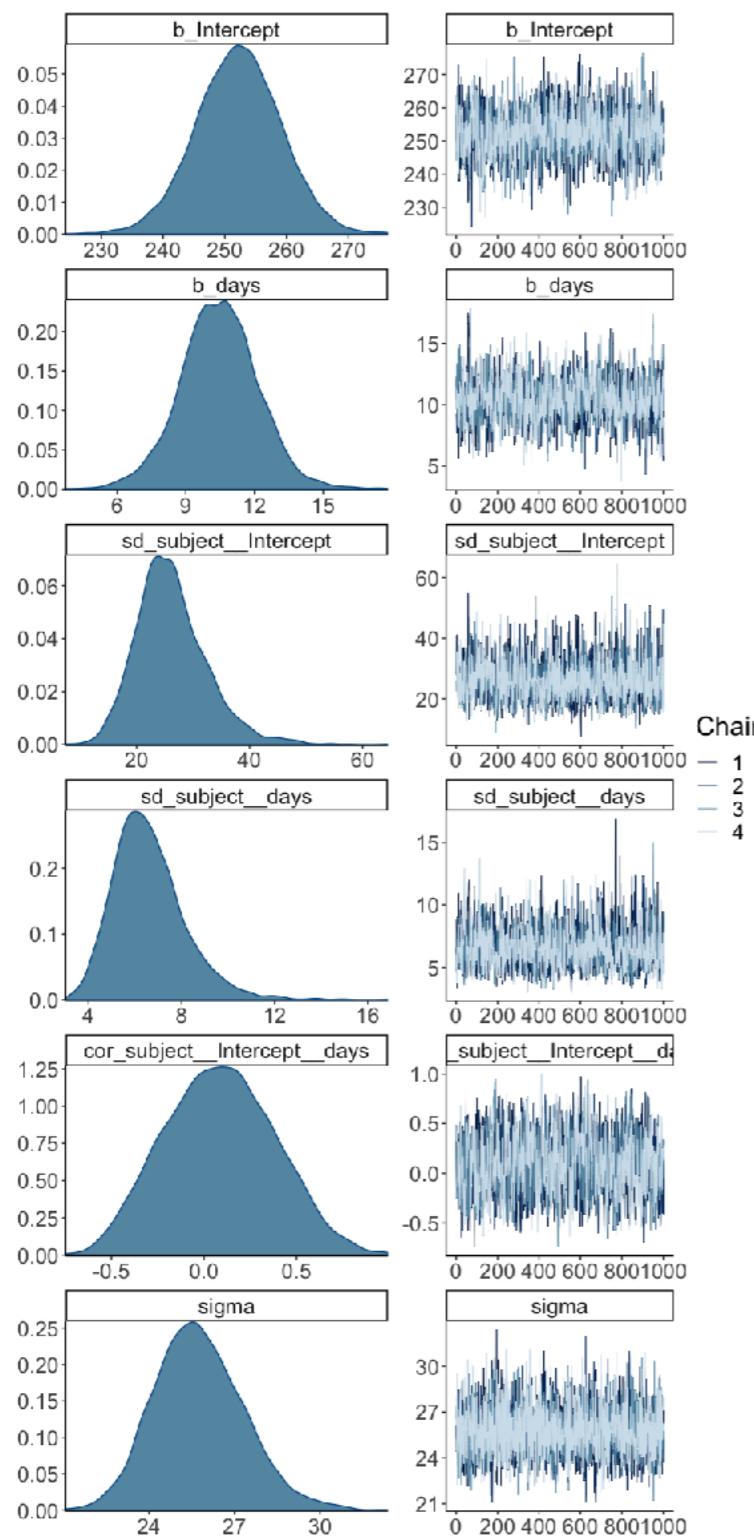
1.00 is  
good!

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>

# a) Did the inference work?

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

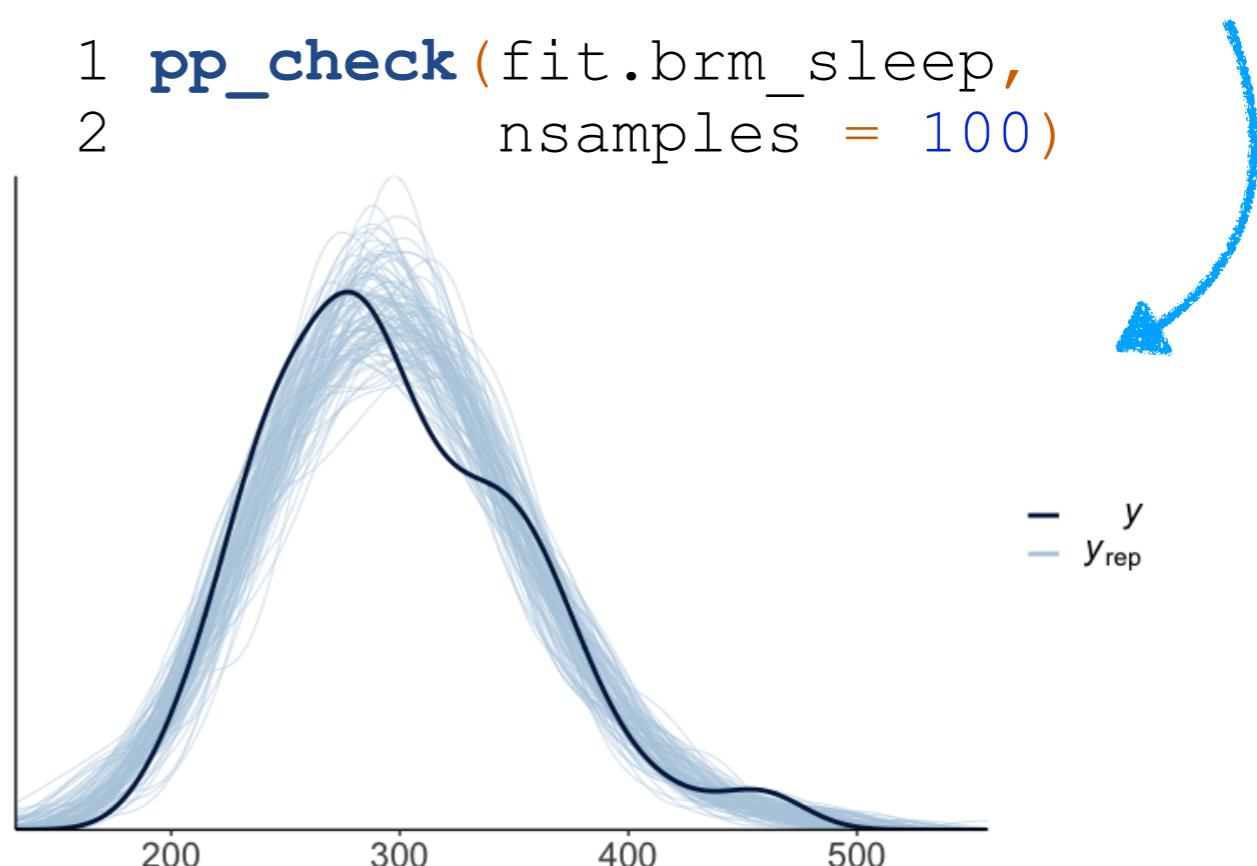


these look good!

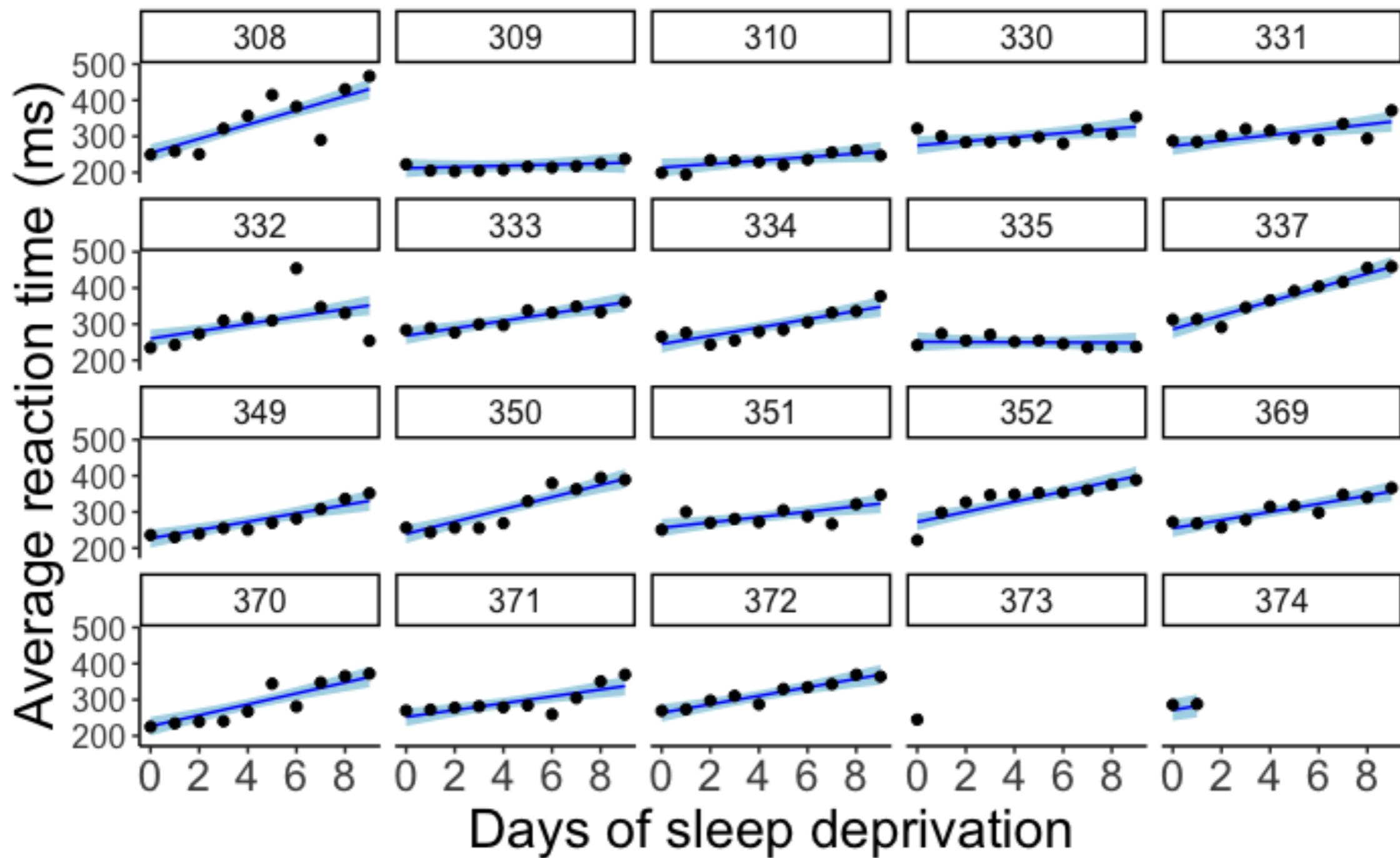


also looks good!

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

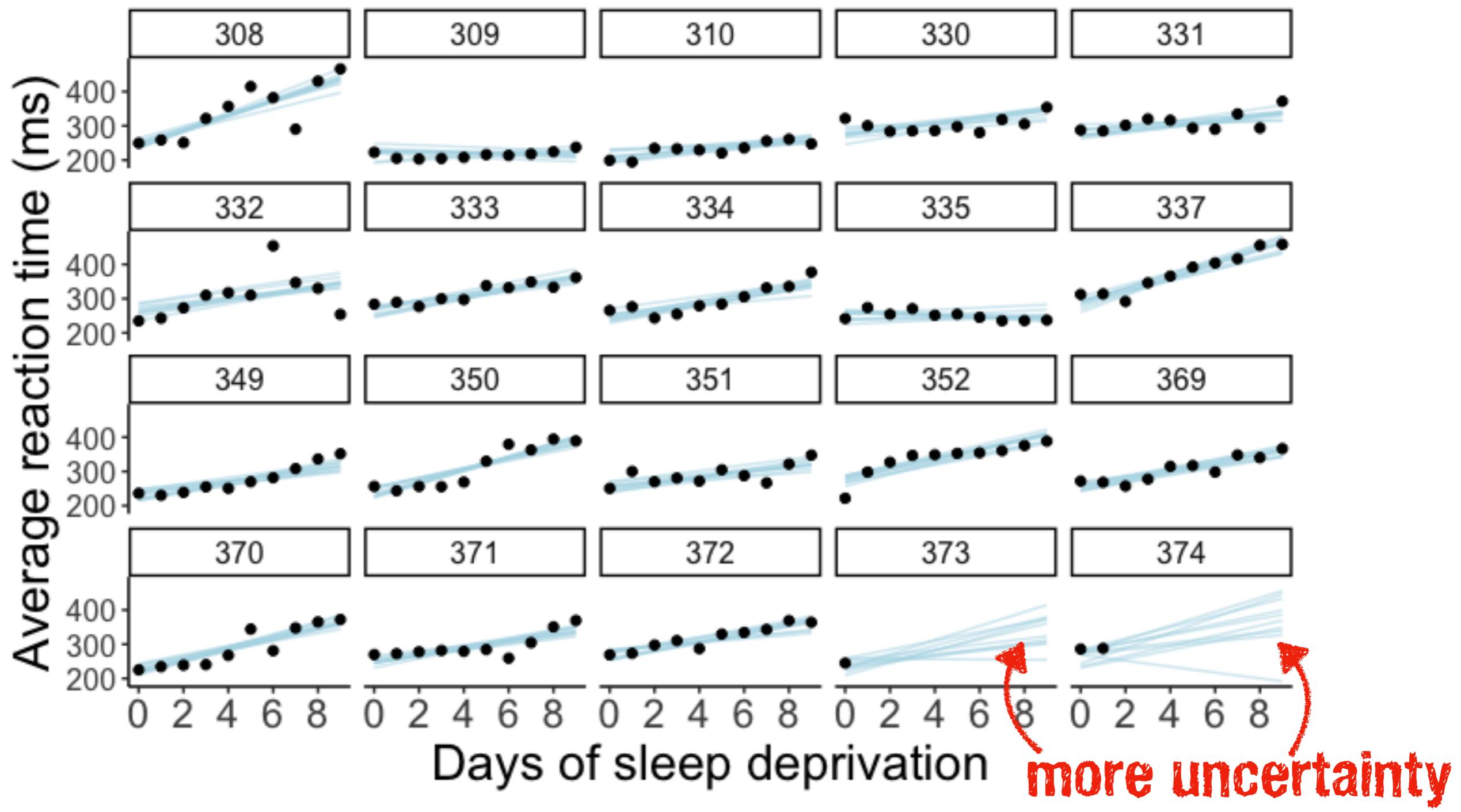


## b) Visualize the model predictions



**regression lines with 95% highest density intervals**

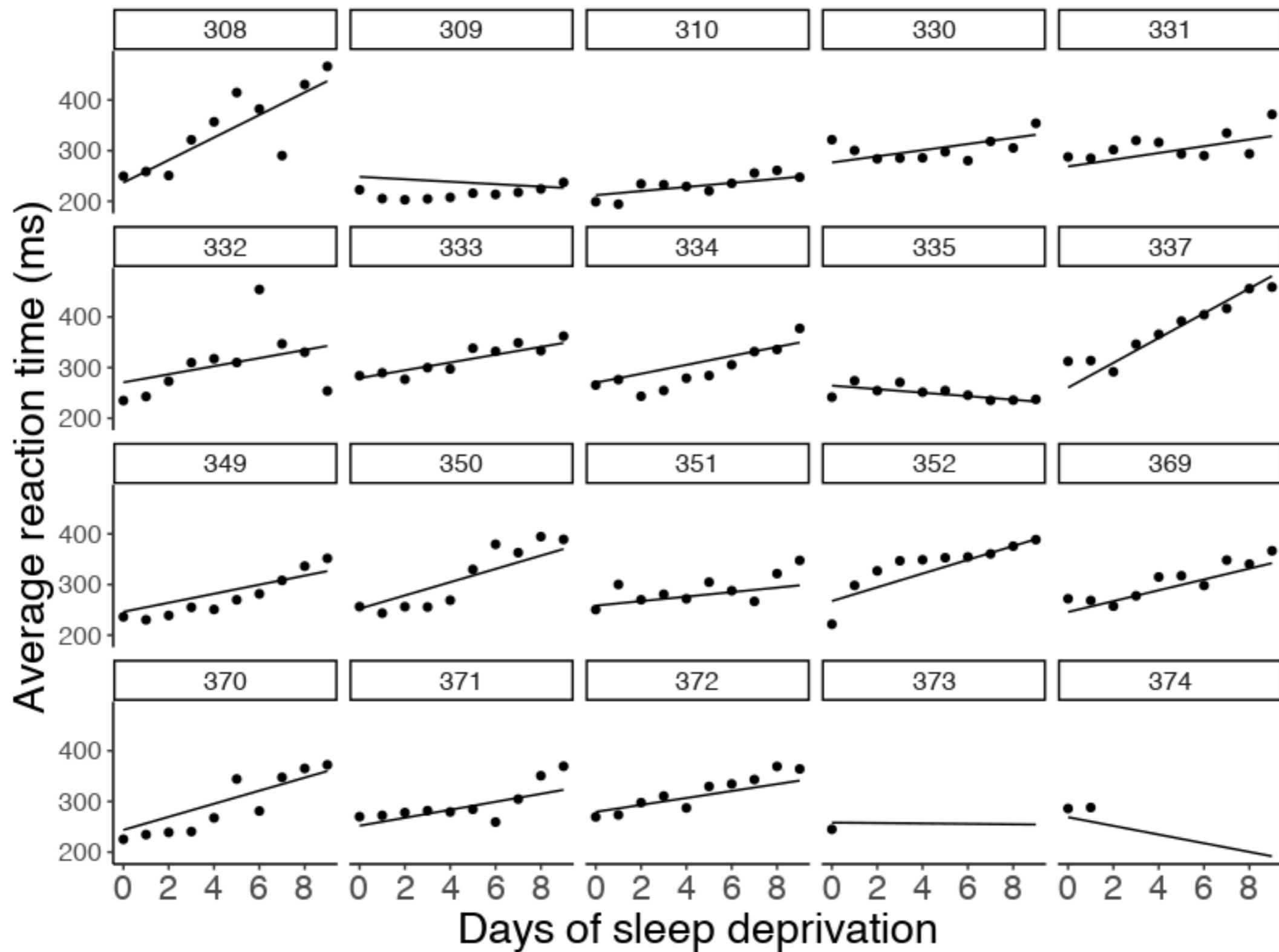
## b) Visualize the model predictions



10 random samples from the posterior distribution

# b) Visualize the model predictions

if you're feeling fancy

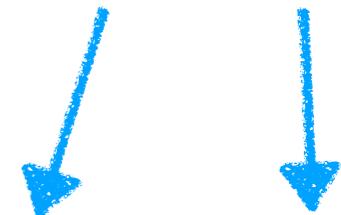


## **4. Interpret the model parameters**

# 4. Interpret the model parameters

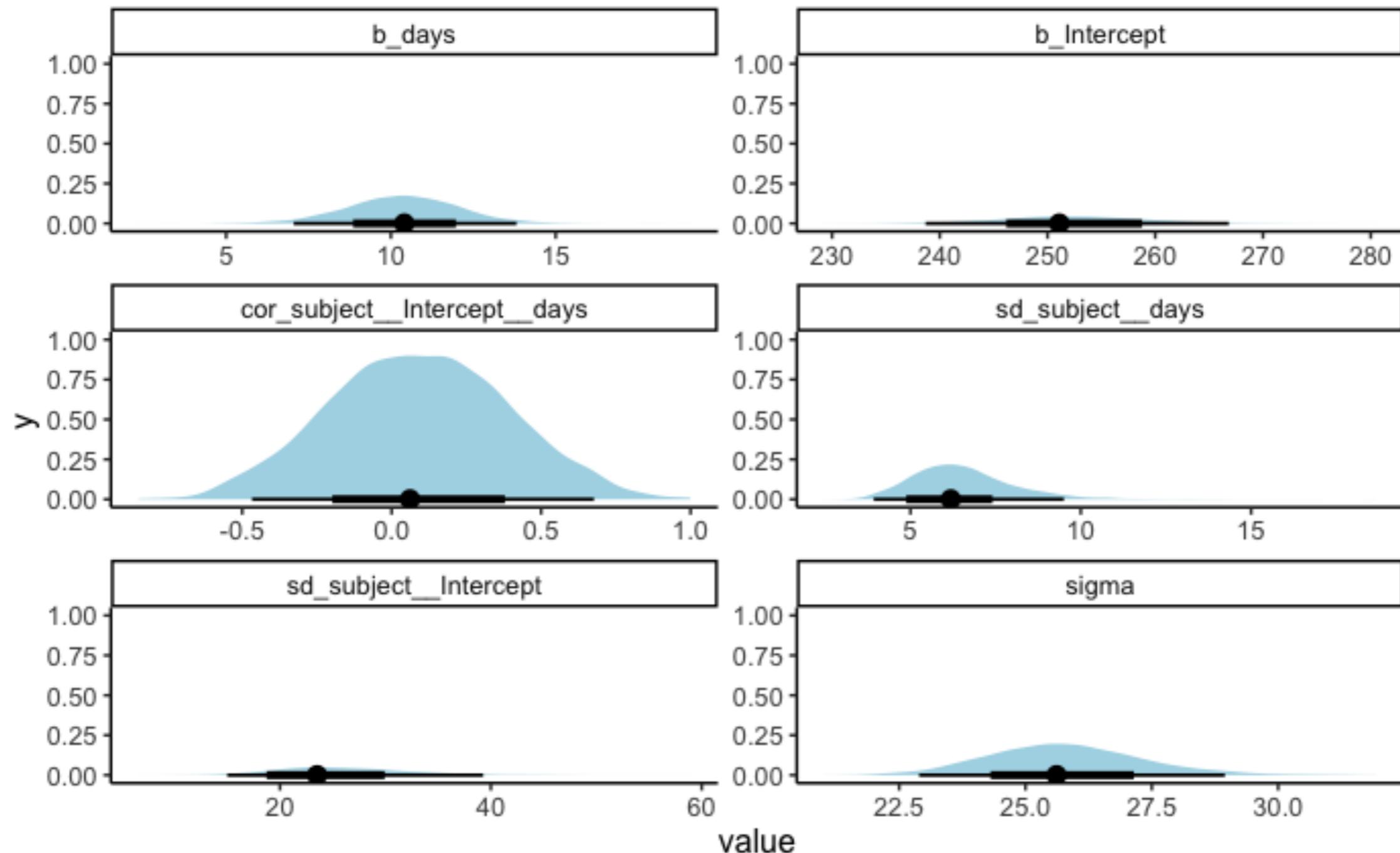
```
1 fit.brm_sleep %>%
2 tidy(conf.method = "HPDinterval")
```

95% highest  
density interval



effect	component	group	term	estimate	std.error	conf.low	conf.high
fixed	cond	NA	(Intercept)	252.39	7.00	238.69	266.82
fixed	cond	NA	days	10.34	1.72	7.05	13.81
ran_pars	cond	subject	sd__(Intercept)	26.14	6.37	15.00	39.27
ran_pars	cond	subject	sd__days	6.55	1.54	3.92	9.50
ran_pars	cond	subject	cor__(Intercept).days	0.09	0.30	-0.47	0.68
ran_pars	cond	Residual	sd__Observation	25.80	1.54	22.90	28.95

# 4. Interpret the model parameters



Posterior distribution for most parameters

# **5. Test specific hypotheses**

# 5. Test specific hypotheses

**Did reaction times increase with the number of days of sleep deprivation?**

```
1 fit.brm_sleep %>%
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

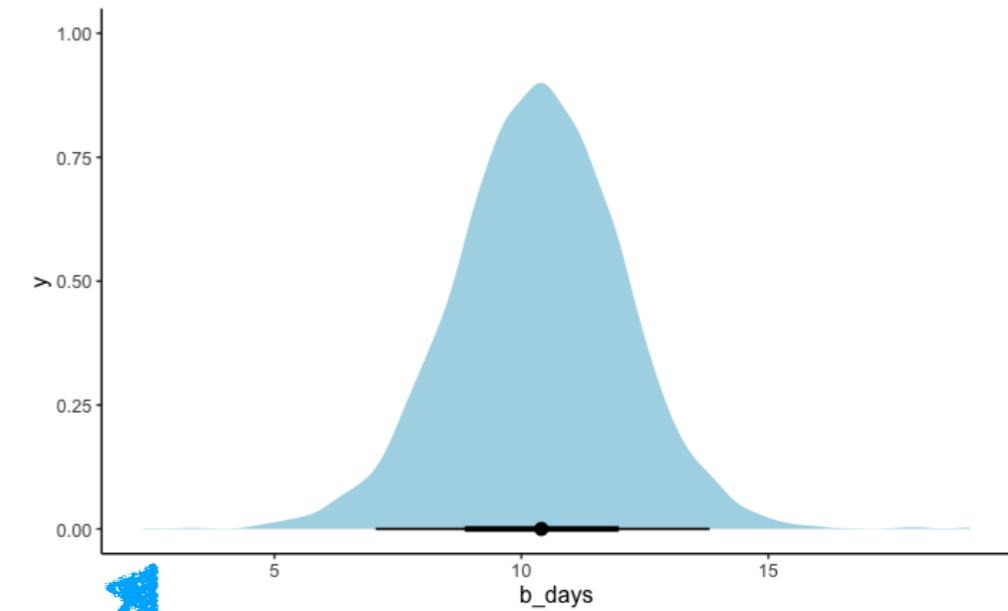
Population-Level Effects:

	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

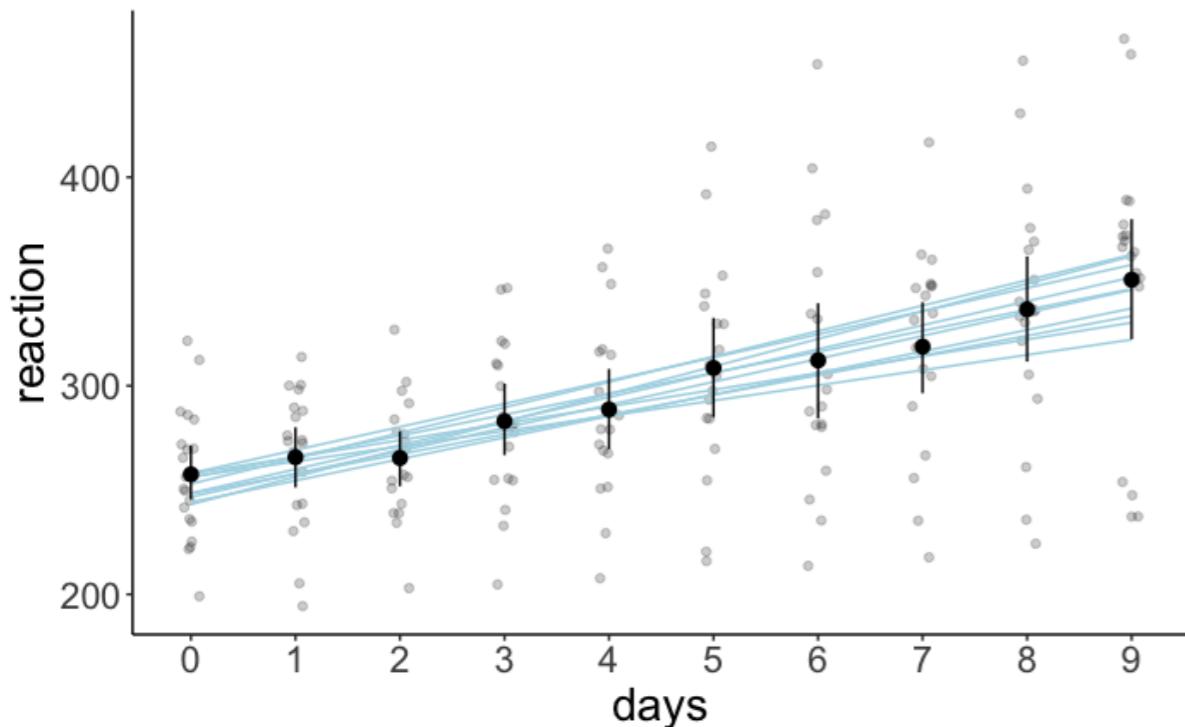
Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).



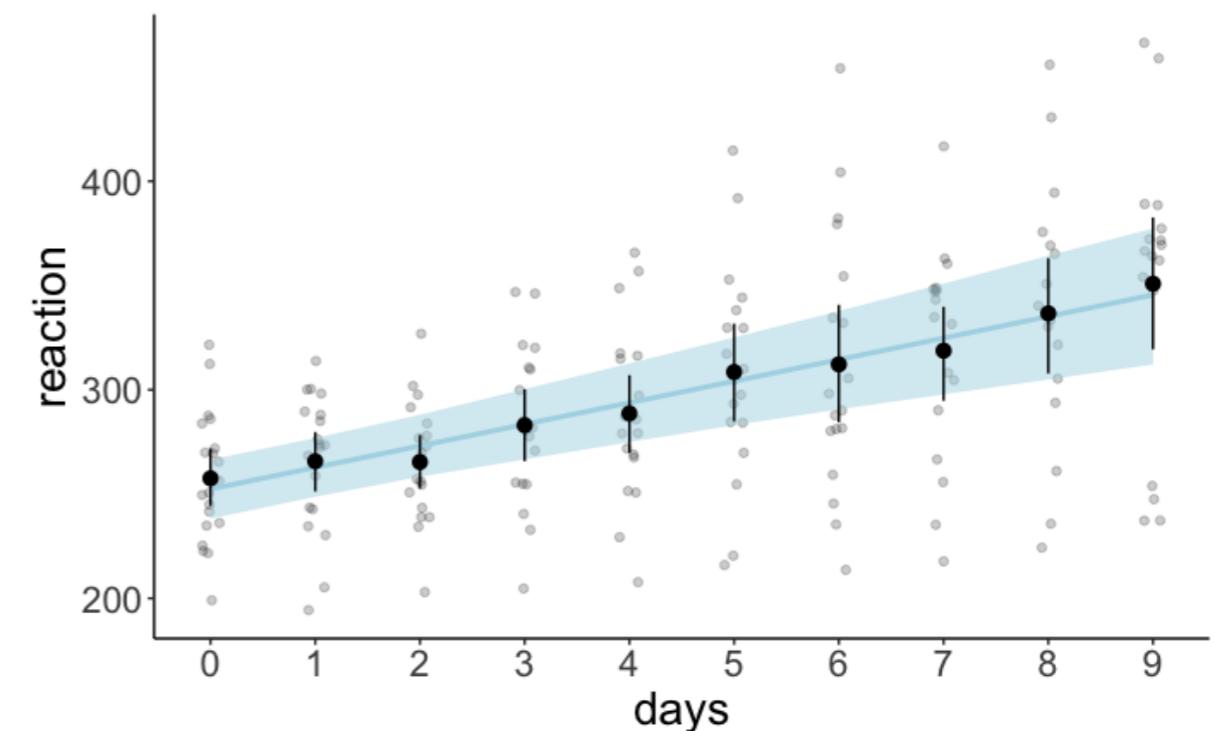
# **6. Report results**

# 6. Report results

10 draws from the posterior



credible intervals



With each day of sleep deprivation, the reaction time increased by 10.5ms (95% HDI: 7.13, 13.78).

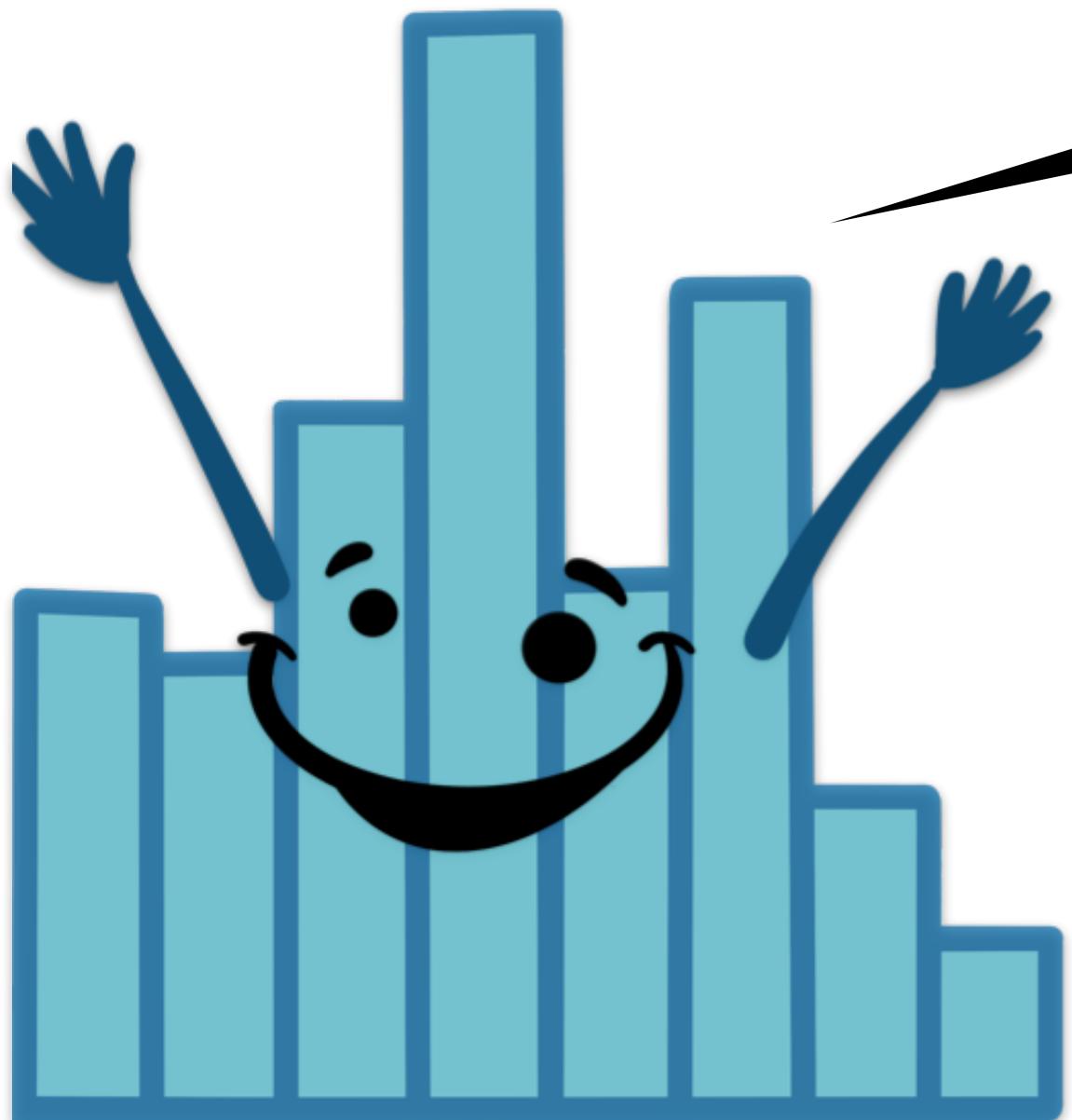
# Recipe for Bayesian analysis with brms

1. Visualize the data
2. Specify and fit the model
3. Model evaluation
  - a) Did the inference work?
  - b) Visualize model predictions
4. Interpret the model parameters
5. Test specific hypotheses
6. Report results

We're listening to "King James -  
Instrumental" by "Anderson.Paak"  
submitted by Peter

02:00

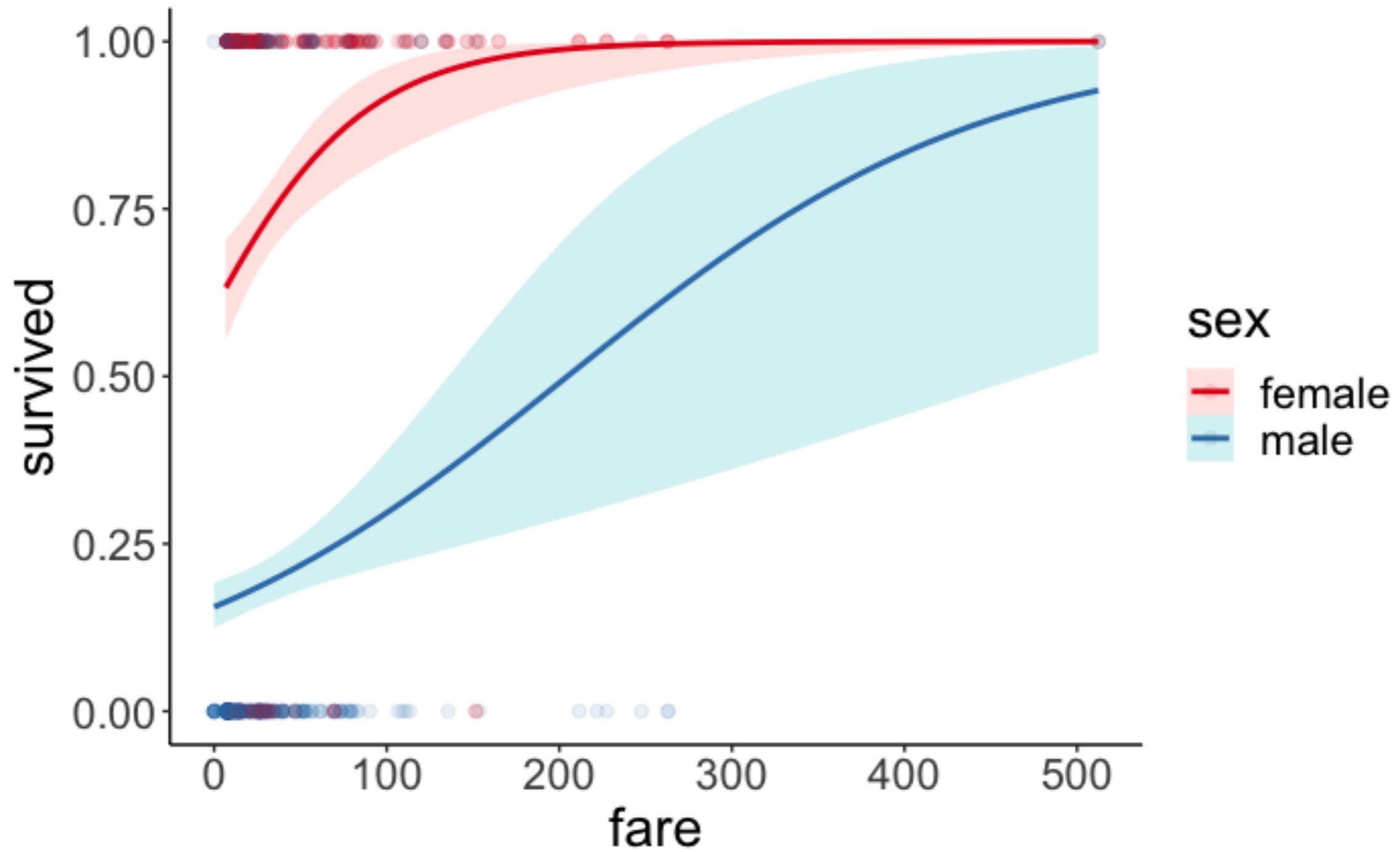
stretch break!



# Titanic data

# **1. Visualize the data**

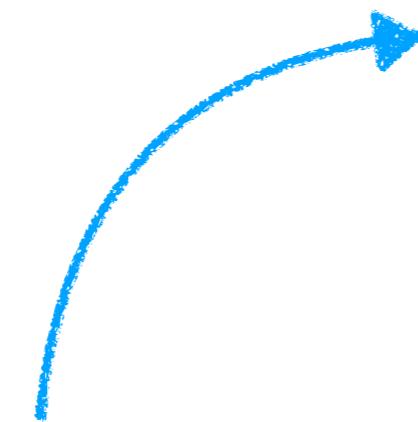
# Feeling cold?



## **2. Specify and fit the model**

# 1. Specify and fit the model

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



just need to  
change the family

# **3. Model evaluation**

# a) Did the inference work?

```
1 fit.brm_titanic %>%
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	0.39	0.19	0.03	0.76	1.00	2010	2625
fare	0.02	0.01	0.01	0.03	1.00	1545	2124
sexmale	-2.09	0.23	-2.54	-1.65	1.00	1754	1984
fare:sexmale	-0.01	0.01	-0.02	-0.00	1.00	1479	2041

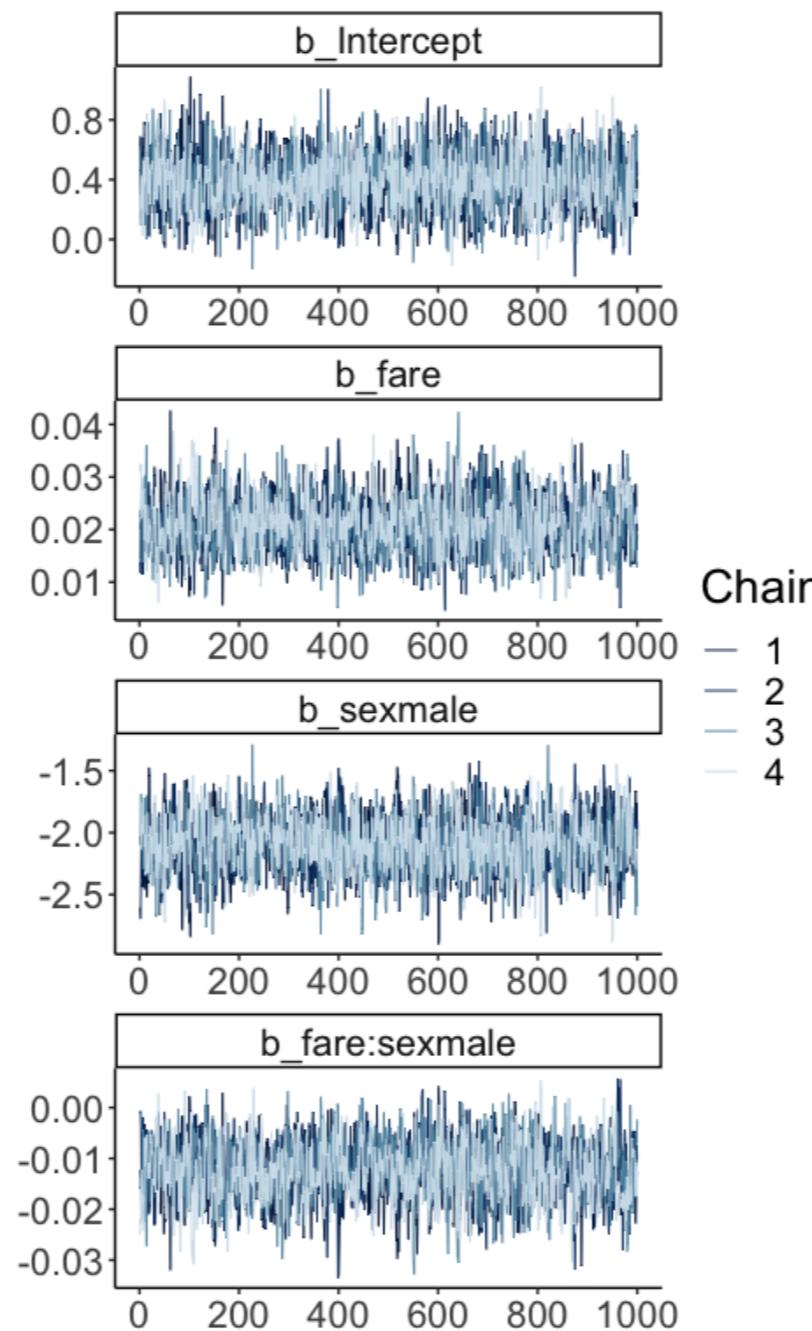
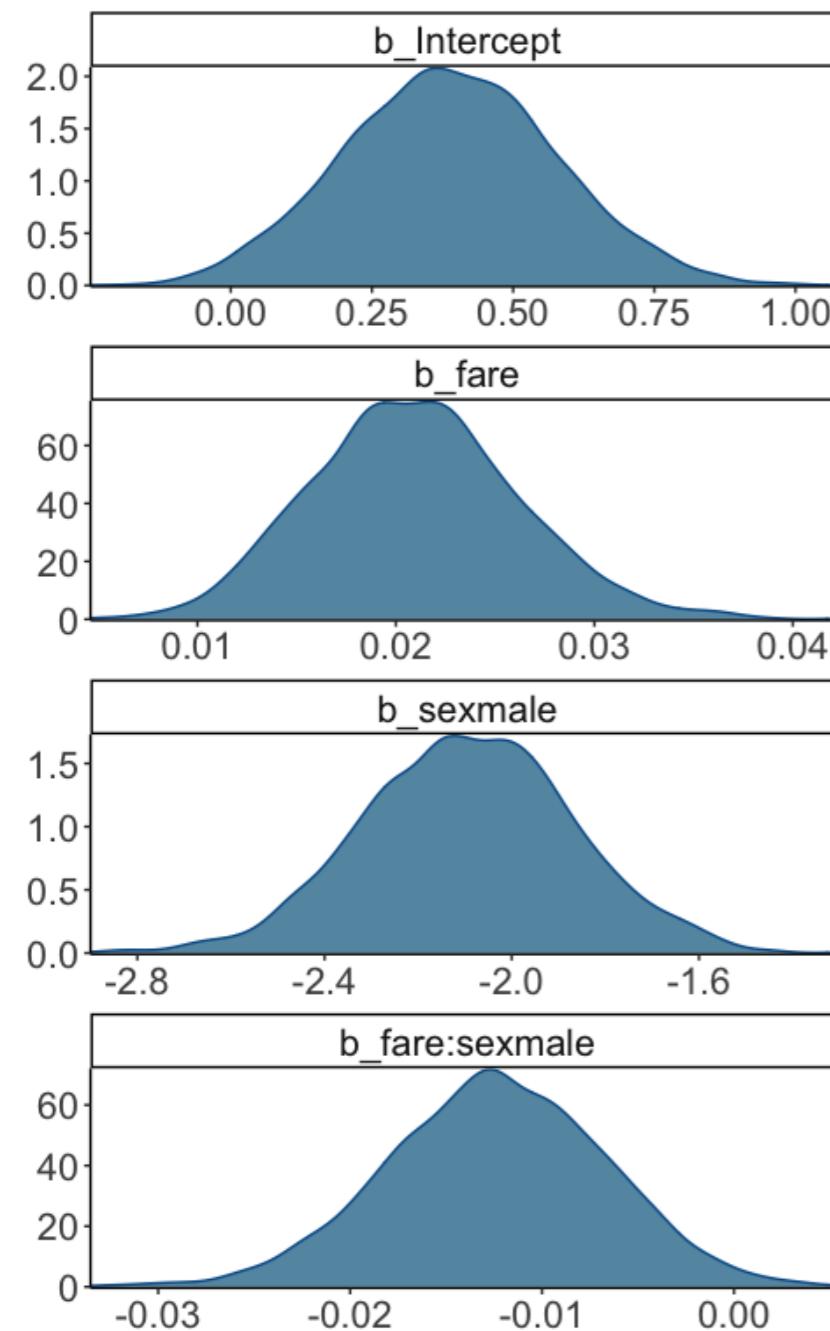
Samples were drawn using sampling(NUTS). For each parameter, Bulk\_ESS and Tail\_ESS are effective sample size measures, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).

looks good

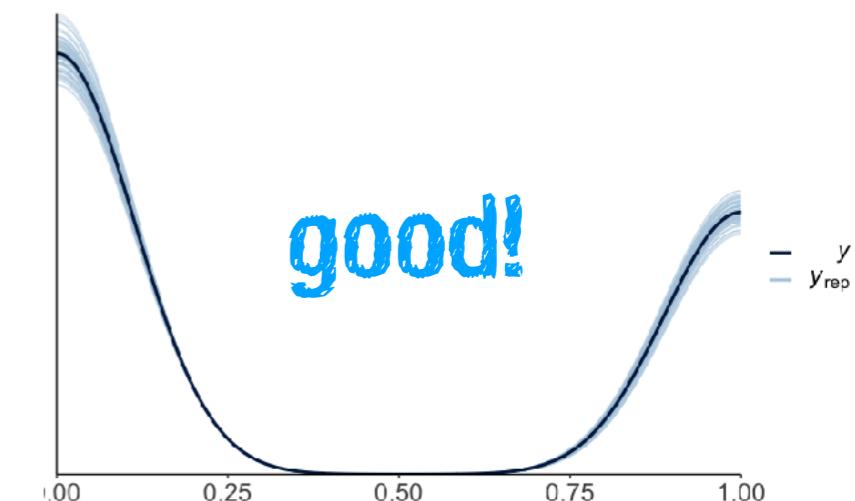
<https://mc-stan.org/misc/warnings.html>

# a) Did the inference work?

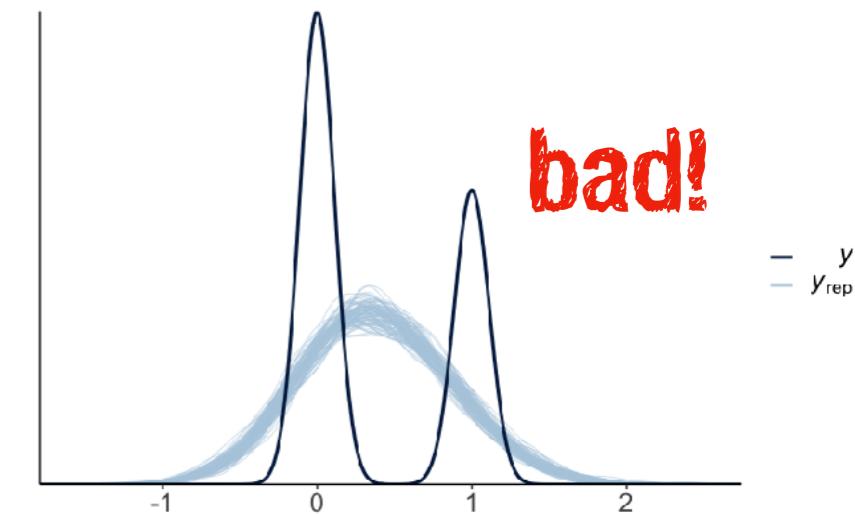
```
1 fit.brm_titanic %>%
2 plot()
```



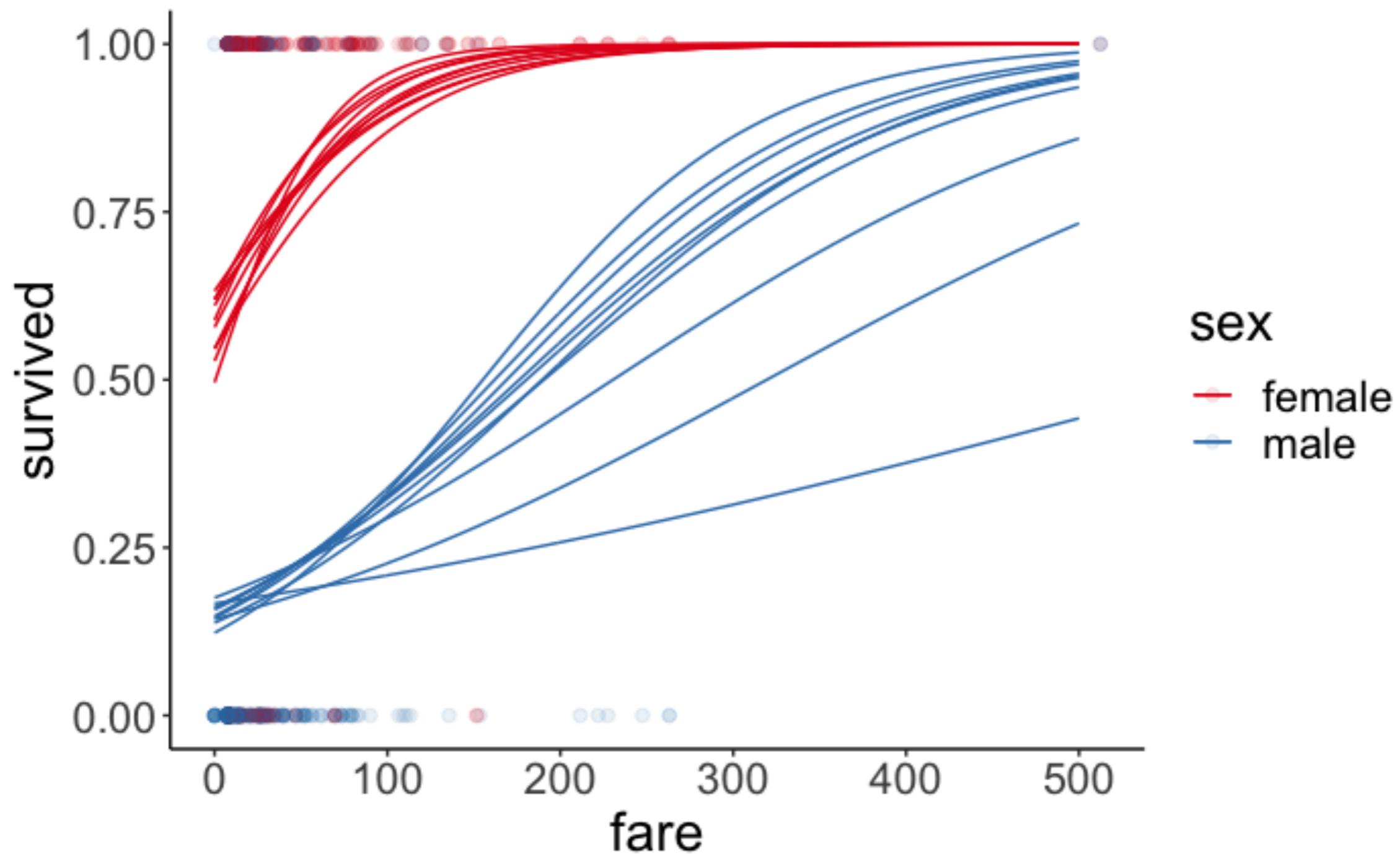
```
1 pp_check(fit.brm_titanic,
2 nsamples = 100)
```



**model with Gaussian family**



## b) Visualize the model predictions



## **4. Interpret the model parameters**

# 4. Interpret the model parameters

```
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	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	0.39	0.19	0.03	0.76	1.00	2010	2625
fare	0.02	0.01	0.01	0.03	1.00	1545	2124
sexmale	-2.09	0.23	-2.54	-1.65	1.00	1754	1984
fare:sexmale	-0.01	0.01	-0.02	-0.00	1.00	1479	2041

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

log odds

# 4. Interpret the model parameters



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

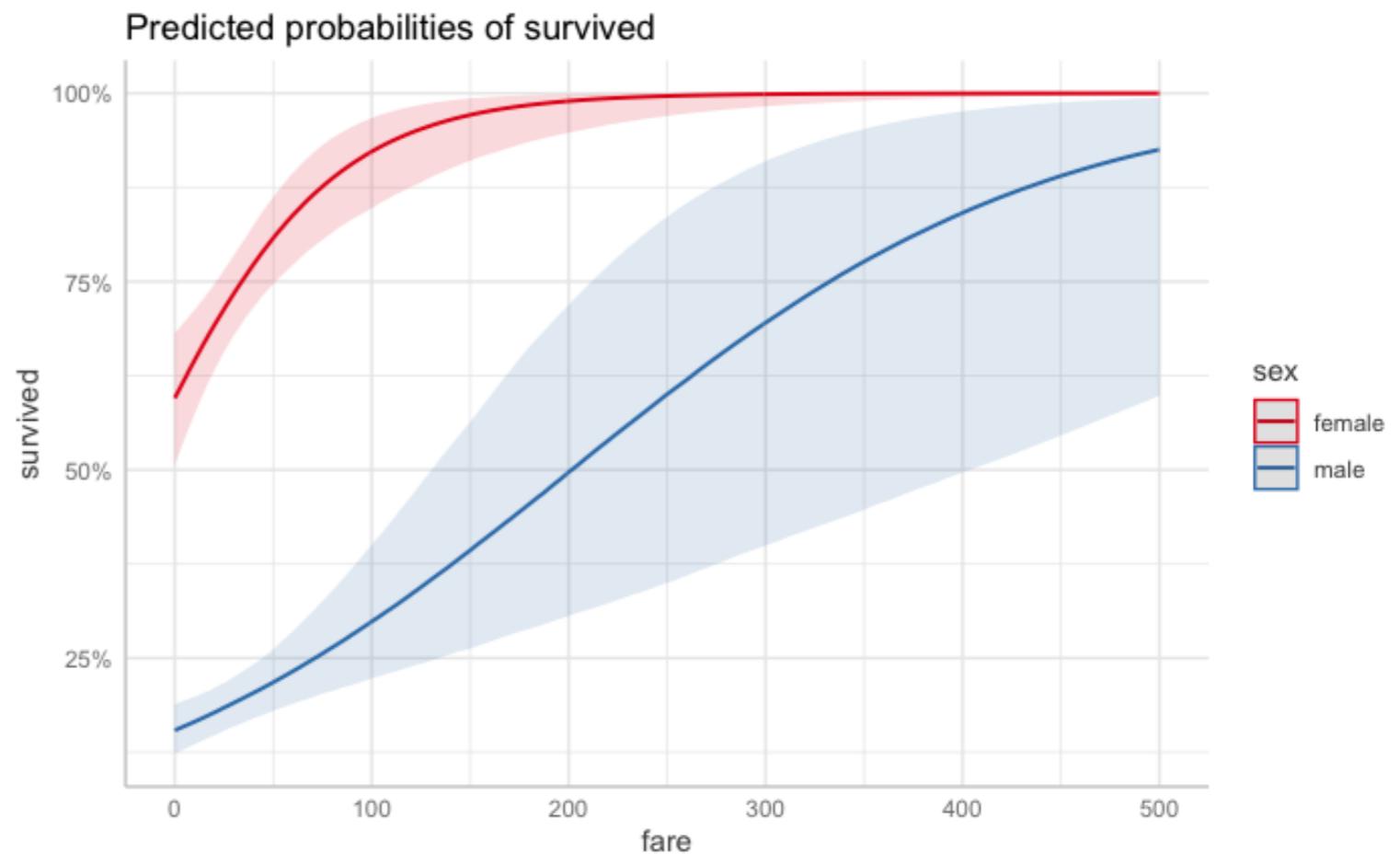
```
Predicted probabilities of survived
x = fare

sex = female

 x | Predicted | 95% CI
-----+
 0 | 0.60 | [0.51, 0.68]
 83 | 0.89 | [0.82, 0.95]
 167 | 0.98 | [0.93, 1.00]
 250 | 1.00 | [0.97, 1.00]
 333 | 1.00 | [0.99, 1.00]
 500 | 1.00 | [1.00, 1.00]

sex = male

 x | Predicted | 95% CI
-----+
 0 | 0.15 | [0.12, 0.19]
 83 | 0.27 | [0.21, 0.35]
 167 | 0.43 | [0.28, 0.62]
 250 | 0.60 | [0.35, 0.84]
 333 | 0.75 | [0.43, 0.94]
 500 | 0.93 | [0.60, 0.99]
```



# **5. Test specific hypotheses**

# 5. Test specific hypotheses

## Were women more likely to survive than men?

```
1 fit.brm_titanic %>%
2 emmeans(specs = pairwise ~ sex,
3 type = "response")
```

NOTE: Results may be misleading due to involvement in interactions

\$emmeans

sex	response	lower.HPD	upper.HPD
female	0.743	0.69	0.795
male	0.194	0.16	0.225

Point estimate displayed: median

Results are back-transformed from the logit scale

HPD interval probability: 0.95

\$contrasts

contrast	odds.ratio	lower.HPD	upper.HPD
female / male	12.1	8.39	16.6

Point estimate displayed: median

Results are back-transformed from the log odds ratio scale

HPD interval probability: 0.95

$$\frac{\left(\frac{p_f}{1 - p_f}\right)}{\left(\frac{p_m}{1 - p_m}\right)}$$

# 5. Test specific hypotheses

Was the effect of fare on survival different for men vs women?

```
1 fit.brn_titanic %>%
2 emtrends(specs = pairwise ~ sex,
3 var = "fare")
```

```
$emtrends
 sex fare.trend lower.HPD upper.HPD
 female 0.02083 0.01129 0.0316
 male 0.00845 0.00385 0.0135
```

Point estimate displayed: median  
HPD interval probability: 0.95

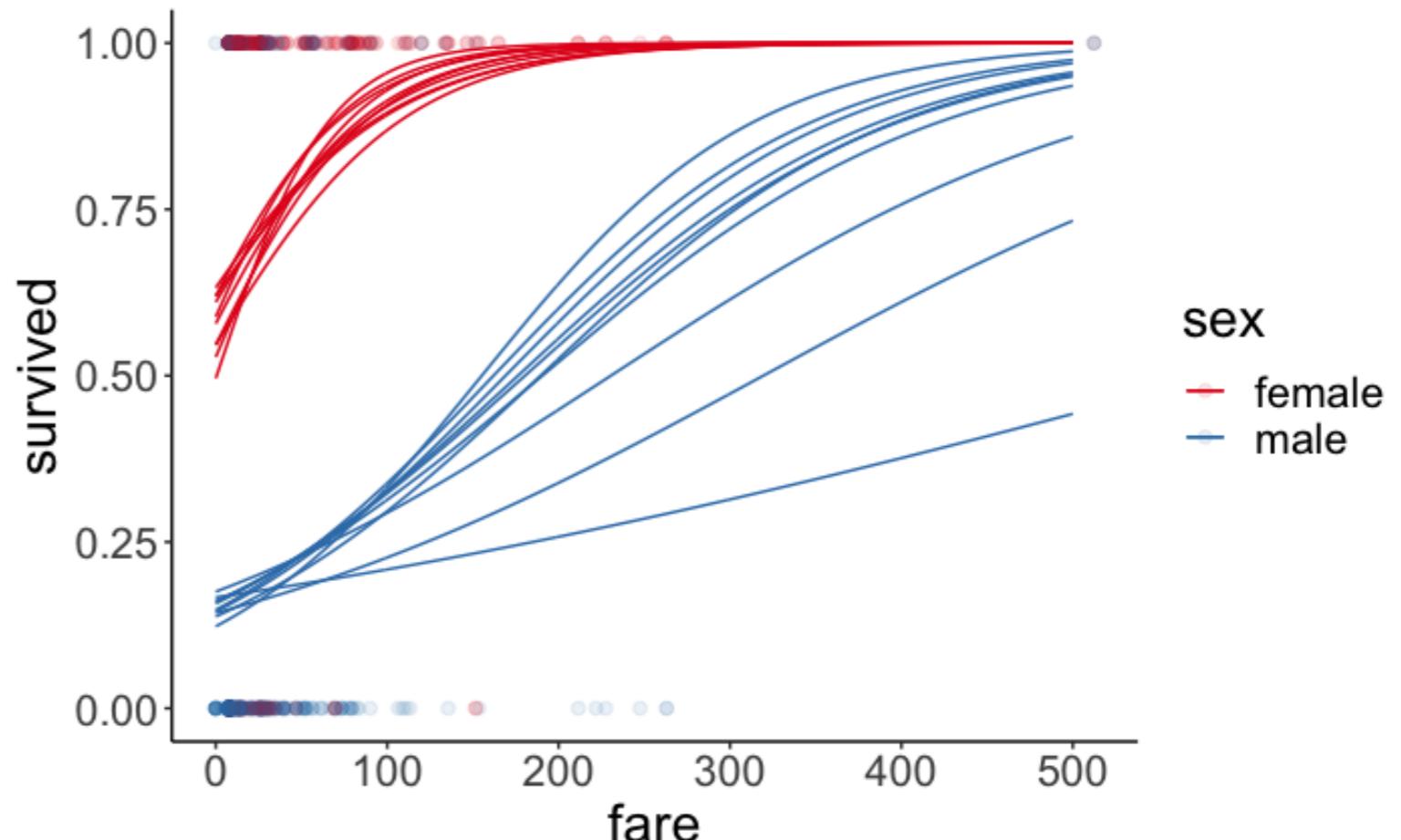
```
$contrasts
 contrast estimate lower.HPD upper.HPD
 female - male 0.0124 0.000884 0.0232
```

Point estimate displayed: median  
HPD interval probability: 0.95

the chance of survival  
increased more with fare  
for female than male  
passengers

# **6. Report results**

# 6. Report results



Female passengers were more likely to survive (74.3%) than male passengers (19.4%). The estimated odds ratio of survival for female vs. male passengers was 12.1 [8.4, 16.6].

The chance of survival increased more with fare for female compared to male passengers. The difference in slopes on the log odds scale was 0.01 [0, 0.02].

# Going beyond

# **Evidence for the null hypothesis**

# Evidence for the null hypothesis



[Front Psychol. 2014; 5: 781.](#)

Published online 2014 Jul 29. doi: [10.3389/fpsyg.2014.00781](https://doi.org/10.3389/fpsyg.2014.00781)

PMCID: PMC4114196

PMID: [25120503](#)

## Using Bayes to get the most out of non-significant results

[Zoltan Dienes\\*](#)

► Author information ► Article notes ► Copyright and License information [Disclaimer](#)

### [HTML] Using Bayes to get the most out of non-significant results

[Z Dienes - Frontiers in psychology, 2014 - frontiersin.org](#)

No scientific conclusion follows automatically from a statistically non-significant result, yet people routinely use non-significant results to guide conclusions about the status of theories (or the effectiveness of practices). To know whether a non-significant result counts against a theory, or if it just indicates data insensitivity, researchers must use one of: power, intervals (such as confidence or credibility intervals), or else an indicator of the relative evidence for one theory over another, such as a Bayes factor. I argue Bayes factors allow theory to be ...

☆ 99 Cited by 966 Related articles All 14 versions Web of Science: 583 Import into BibTeX »

1551 now

- There is nothing special about  $H_0$  compared to  $H_1$  in Bayesian inference
- We can get evidence of  $H_0$  over  $H_1$  (e.g. using the Bayes factor approach)

# Bayes factor

check this out

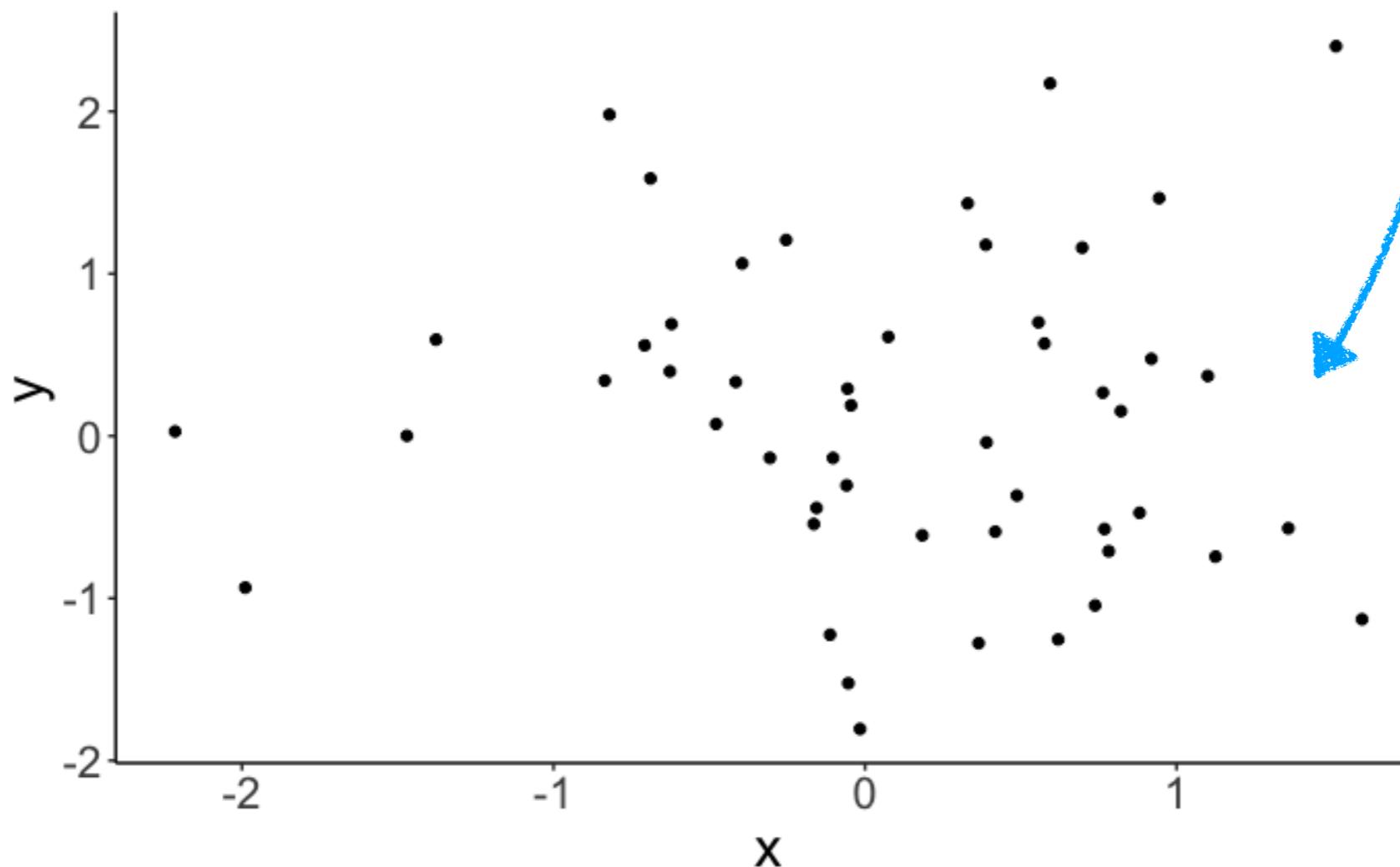


<https://vuorre.netlify.com/post/2017/03/21/bayes-factors-with-brms/>

# Approximate LOO

# Evidence for the null hypothesis

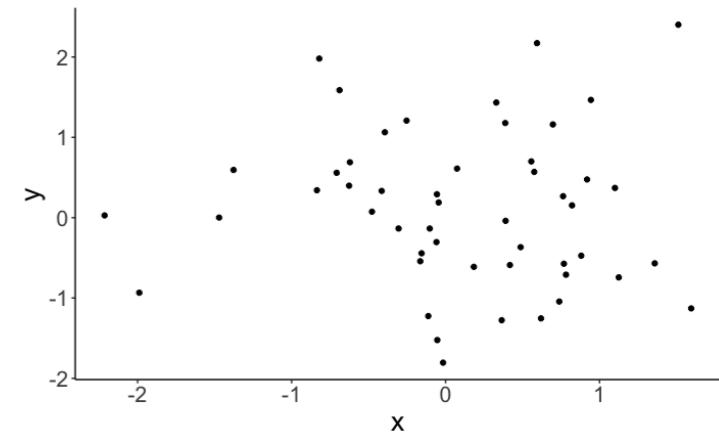
```
1 set.seed(1)
2 df.loo = tibble(x = rnorm(n = 50),
3 y = rnorm(n = 50))
4
5 ggplot(data = df.loo,
6 mapping = aes(x = x,
7 y = y)) +
8 geom_point()
```



no relationship  
between x and y

# Evidence for the null hypothesis

```
1 fit.lm_loo = lm(formula = y ~ 1 + x,
2 data = df.loo)
3
4 fit.lm_loo %>%
5 summary()
```



```
Call:
lm(formula = y ~ 1 + x, data = df.loo)
```

Residuals:

Min	1Q	Median	3Q	Max
-4.2185	-0.6735	0.0018	0.6734	4.2428

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	0.0006437	0.0031639	0.203	0.839
x	-0.0019184	0.0031541	-0.608	0.543

```
Residual standard error: 1.001 on 99998 degrees of freedom
Multiple R-squared: 3.7e-06, Adjusted R-squared: -6.301e-06
F-statistic: 0.37 on 1 and 99998 DF, p-value: 0.543
```

cannot reject the  $H_0$  that the reduction in error due to  $x$  is what one would have expected by chance

# Evidence for the null hypothesis

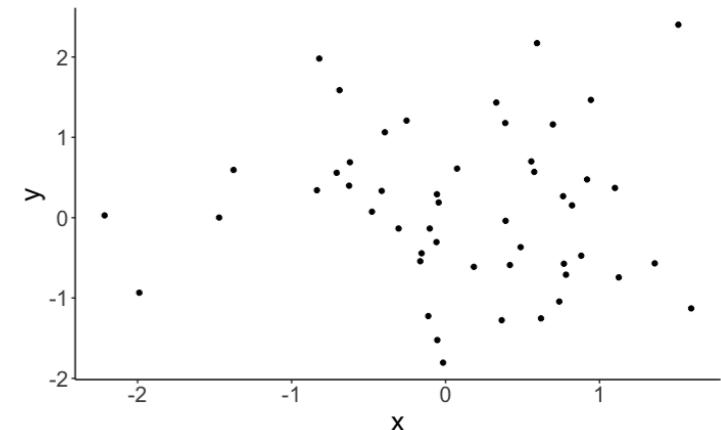
```
1 fit.brm_loo1 = brm(formula = y ~ 1, data = df.loo)
2
3 fit.brm_loo2 = brm(formula = y ~ 1 + x, data = df.loo)
4
5 fit.brm_loo1 = add_criterion(fit.brm_loo1, criterion = "loo")
6
7 fit.brm_loo2 = add_criterion(fit.brm_loo2, criterion = "loo")
```

**loo\_compare(fit.brm\_loo1, fit.brm\_loo2)**

	elpd_diff	se_diff
fit.brm_loo1	0.0	0.0
fit.brm_loo2	-1.1	0.5

**model\_weights(fit.brm\_loo1, fit.brm\_loo2)**

fit.brm_loo1	fit.brm_loo2
99.99999	0.00001



approximate  
leave-one-out  
cross-validation

**I want only positive coefficients!**

# I only want positive coefficients!

```
1 brm(formula = how_much_i_love_stats ~ 1 + tobi + andrew + catherine + jon + dan,
2 data = df.stats_love)
```

coefficients in the model

```
1 # priors
2 priors = c(set_prior("normal(0,10)", class = "b", lb = 0))
```

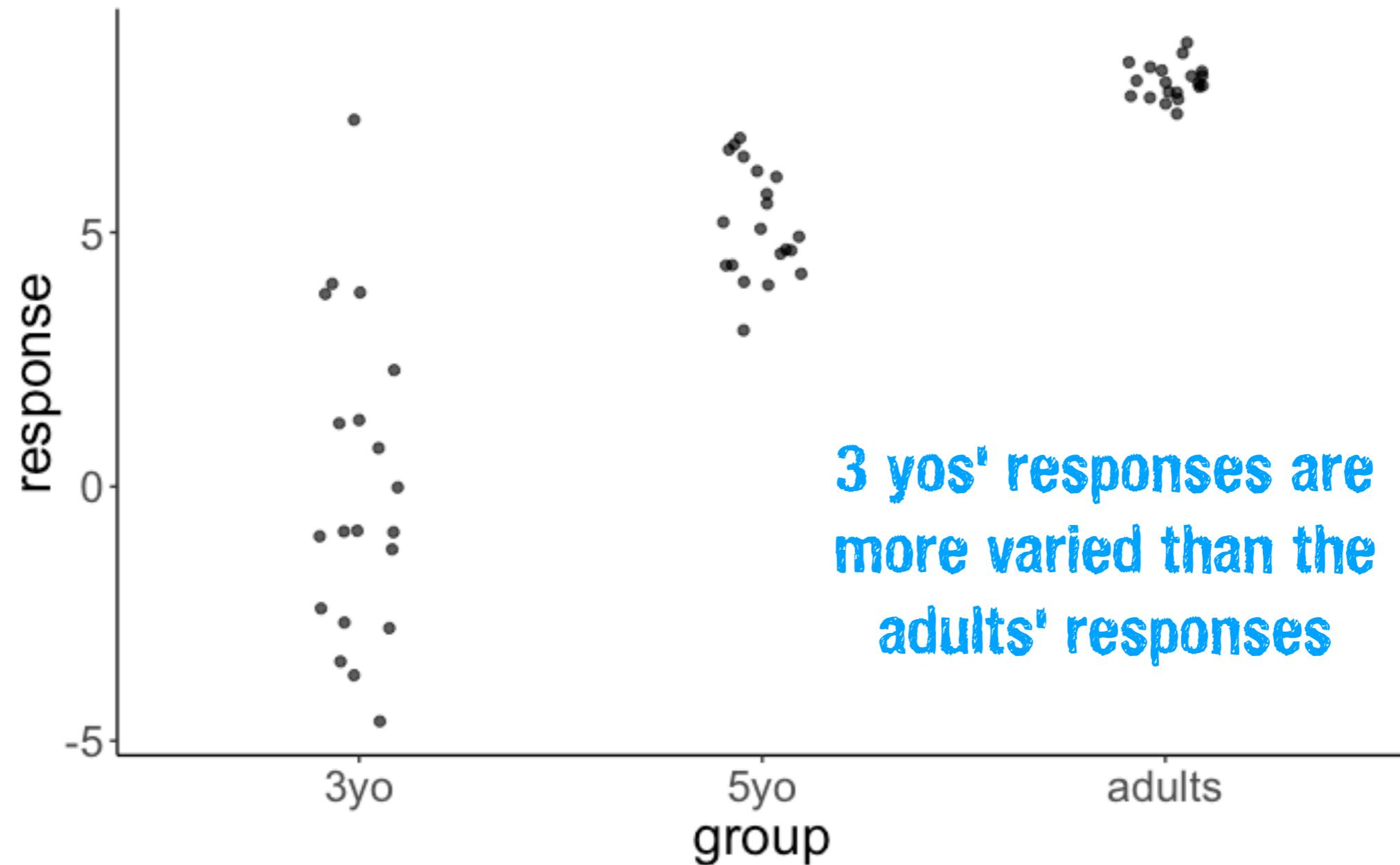
lower bound = 0

```
1 brm(formula = how_much_i_love_stats ~ 1 + tobi + andrew + catherine + jon + dan,
2 prior = priors,
3 data = df.stats_love)
```

# **Dealing with unequal variance**

# Unequal variance aka heteroscedasticity

```
1 df.variance = tibble(group = rep(c("3yo", "5yo", "adults"), each = 20),
2 response = rnorm(n = 60,
3 mean = rep(c(0, 5, 8), each = 20),
4 sd = rep(c(3, 1.5, 0.3), each = 20)))
```



# Unequal variance aka heteroscedasticity

```
1 fit.lm1 = lm(formula = response ~ 1 + group,
2 data = df.variance)
3
4 fit.lm1 %>%
5 summary()
```

```
Call:
lm(formula = response ~ 1 + group, data = df.variance)

Residuals:
 Min 1Q Median 3Q Max
-4.6145 -0.8288 -0.0879 0.6315 7.2193

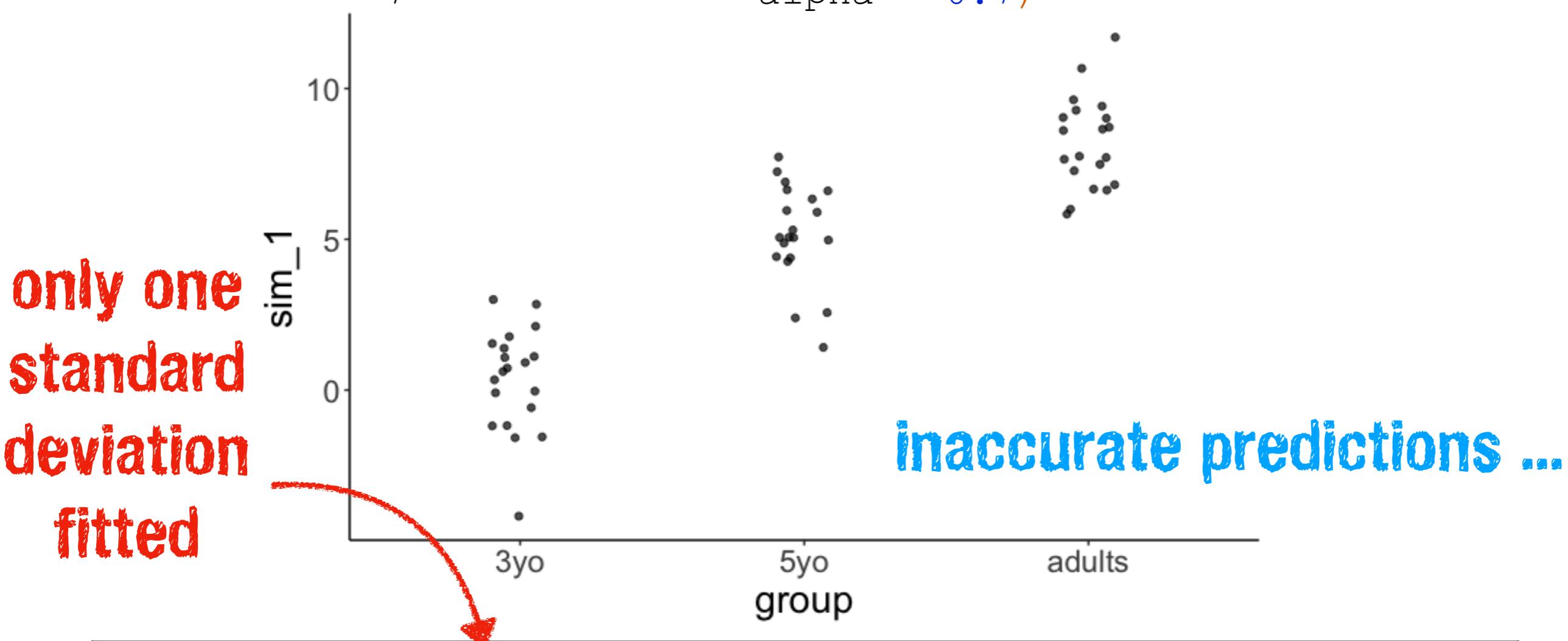
Coefficients:
 Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.005336 0.421618 -0.013 0.99
group5yo 5.172810 0.596258 8.675 5.25e-12 ***
groupadults 7.970655 0.596258 13.368 < 2e-16 ***

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

Residual standard error: 1.886 on 57 degrees of freedom
Multiple R-squared: 0.7635, Adjusted R-squared: 0.7552
F-statistic: 91.99 on 2 and 57 DF, p-value: < 2.2e-16
```

# Unequal variance aka heteroscedasticity

```
1 fit.lm1 %>%
2 simulate() %>%
3 bind_cols(df.variance) %>%
4 ggplot(aes(x = group, y = sim_1)) +
5 geom_jitter(height = 0,
6 width = 0.1,
7 alpha = 0.7)
```



r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual
0.76	0.76	1.89	91.99	0	3	-121.65	251.3	259.68	202.65	57

# Unequal variance aka heteroscedasticity

```
1 fit.brml = brm(formula = bf(response ~ group,
2 sigma ~ group),
3 data = df.variance,
4 file = "cache/brml",
5 seed = 1)
```

modeling both the  
means and variances

```
Family: gaussian
Links: mu = identity; sigma = log
Formula: response ~ group
 sigma ~ group
Data: df.variance (Number of observations: 60)
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	-0.01	0.73	-1.41	1.51	1.01	1107	1072
sigma_Intercept	1.15	0.17	0.85	1.51	1.00	1991	1922
group5yo	5.18	0.77	3.60	6.65	1.00	1252	1327
groupadults	7.98	0.74	6.47	9.37	1.01	1110	1079
sigma_group5yo	-1.05	0.24	-1.51	-0.57	1.00	2249	2420
sigma_groupadults	-2.19	0.24	-2.66	-1.74	1.00	2171	2427

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

# Unequal variance aka heteroscedasticity

```
Family: gaussian
Links: mu = identity; sigma = log ← on a log scale!
Formula: response ~ group
 sigma ~ group
Data: df.variance (Number of observations: 60)
Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
 total post-warmup samples = 4000
```

Population-Level Effects:

	Estimate	Est.Error	1-95% CI	u-95% CI	Rhat	Bulk_ESS	Tail_ESS
Intercept	-0.01	0.73	-1.41	1.51	1.01	1107	1072
sigma_Intercept	1.15	0.17	0.85	1.51	1.00	1991	1922
group5yo	5.18	0.77	3.60	6.65	1.00	1252	1327
groupadults	7.98	0.74	6.47	9.37	1.01	1110	1079
sigma_group5yo	-1.05	0.24	-1.51	-0.57	1.00	2249	2420
sigma_groupadults	-2.19	0.24	-2.66	-1.74	1.00	2171	2427

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

mean = **c(0, 5, 8)**

sd = **c(3, 1.5, 0.3)**

**3 year olds**  $e^{1.15} = 3.16$

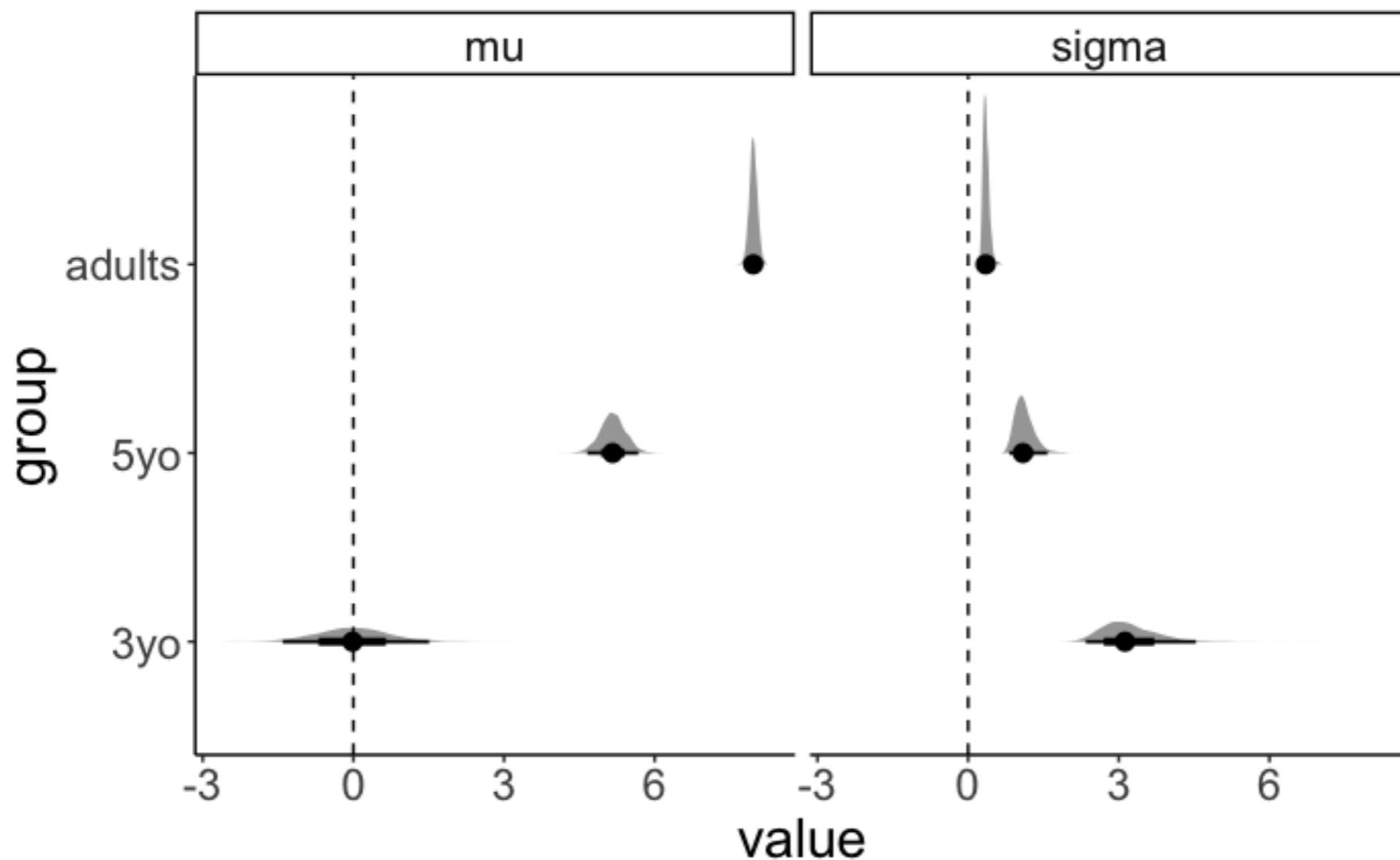


**5 year olds**  $e^{1.15+(-1.05)} = 1.10$

**adults**  $e^{1.15+(-2.19)} = 0.35$

# Unequal variance aka heteroscedasticity

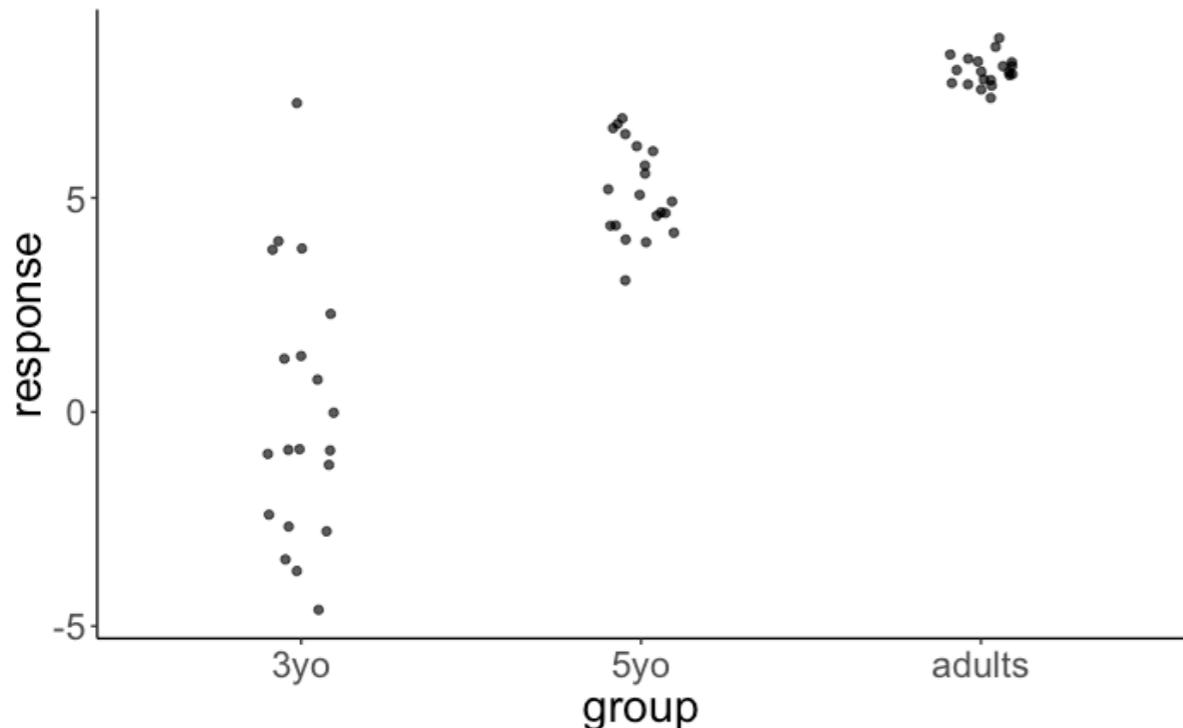
```
1 df.variance %>%
2 expand(group) %>%
3 add_fitted_draws(fit.brml, dpar = TRUE) %>%
4 select(group, .row, .draw, posterior = .value, mu, sigma) %>%
5 pivot_longer(cols = c(mu, sigma),
6 names_to = "index",
7 values_to = "value") %>%
8 ggplot(aes(x = value, y = group)) +
9 geom_halfeyeh() +
10 geom_vline(xintercept = 0, linetype = "dashed") +
11 facet_grid(cols = vars(index))
```



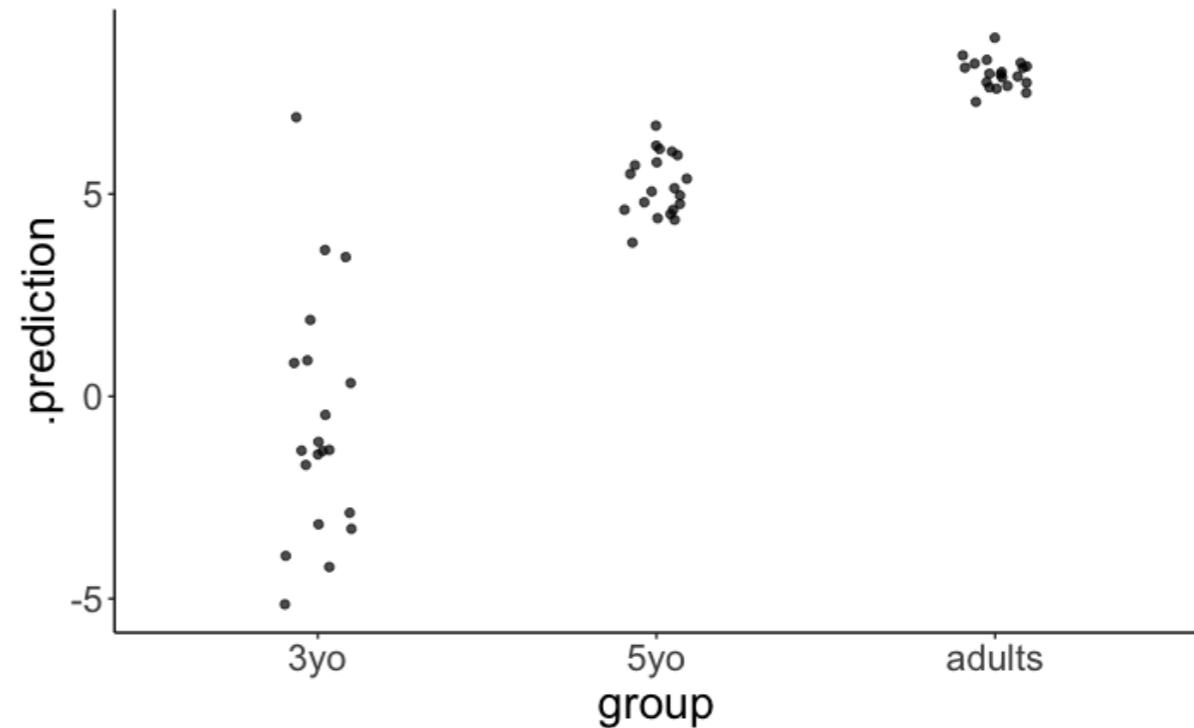
# Unequal variance aka heteroscedasticity

```
1 df.variance %>%
2 add_predicted_draws(model = fit.brml,
3 n = 1) %>%
4 ggplot(aes(x = group, y = .prediction)) +
5 geom_jitter(height = 0,
6 width = 0.1,
7 alpha = 0.7)
```

original data



predicted data



these predictions look good!

Thanks