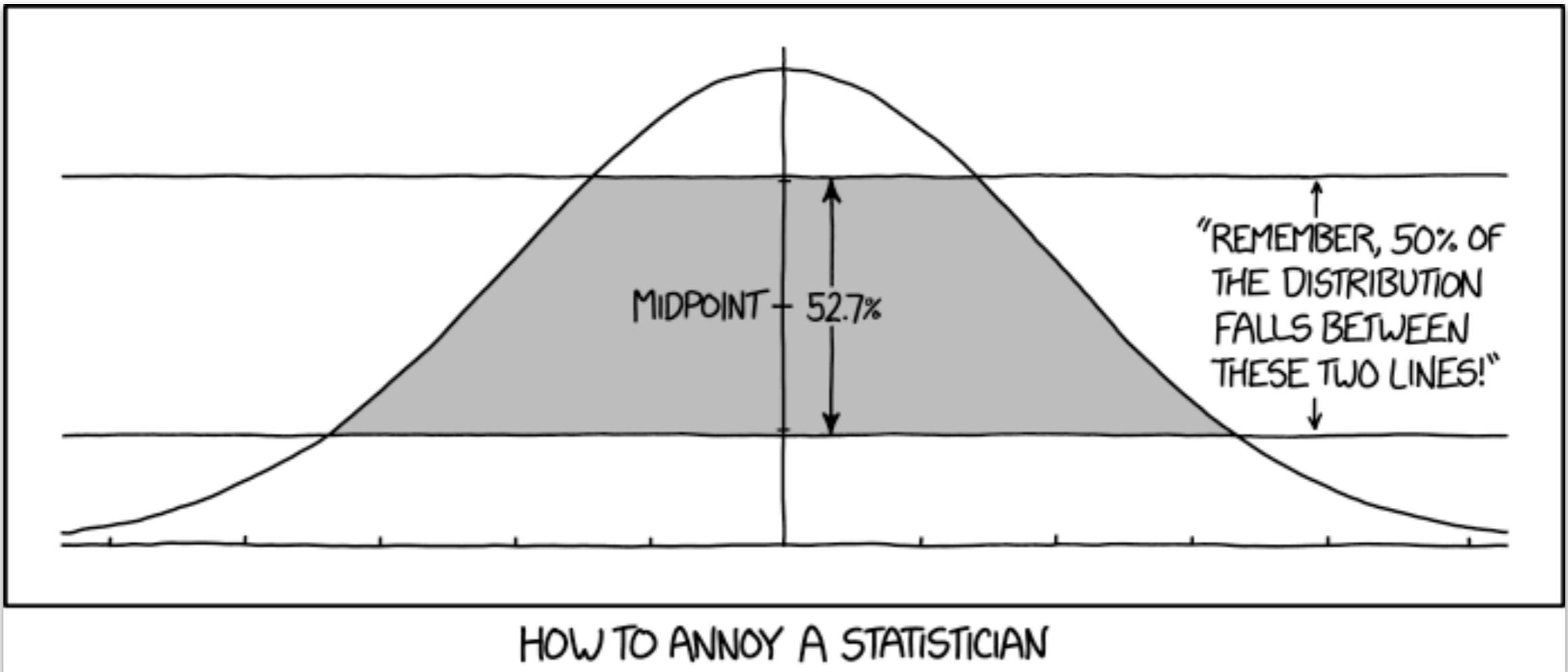


Bayesian data analysis 2



We're listening to "Ngui Yi" by "Blick Bassy"

03/02/2022

Logistics

Final presentation

Questions Responses 18 Settings

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 survey interface includes a top navigation bar with 'Questions' (red underline), 'Responses' (18), and 'Settings'. On the right side, there is a vertical toolbar with icons for adding a question (+), saving (disk), translating (Tr), inserting a file (cloud), inserting a video (play), and inserting a table (grid).

Plan for today

- Quick recap
- Doing Bayesian data analysis with *greta*
- Doing Bayesian data analysis **with BRMS**
 - Recipe for Bayesian data analysis
 - Poker data
 - Sleep data
 - Titanic data

Quick recap

Quick recap: Bayesian data analysis philosophy

Goal of data analysis: Inference about the world

Frequentist statistics

- generate a sampling distribution of the test statistic assuming H_0
- compare observed value of the test statistic with the sampling distribution
- reject the H_0 if probability of observed value (or more extreme values) is less than α

Bayesian statistics

- directly test hypotheses of interest
- define prior over hypotheses $p(H)$
- compute likelihood of the data for each hypothesis $p(D|H)$
- use Bayes' rule to infer the posterior over hypotheses given the data $p(H|D)$

21

Bayesian Recipe

- Hypotheses
- Prior over hypotheses
- Data
- Likelihood of the data given each hypothesis
- Posterior over hypotheses given the data

+ a healthy dose
of Bayes' rule

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

37

Clue guide to probability

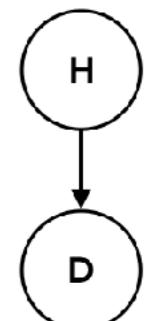
$$p(B|A) = \frac{p(A|B) \cdot p(B)}{p(A)}$$

we derived this using the definition of conditional probability

$$p(H|D) = \frac{\text{likelihood} \cdot \text{prior}}{p(D)} \quad \begin{matrix} \text{subjective probability} \\ \text{interpretation} \end{matrix}$$

$H = \text{Hypothesis}$

$D = \text{Data}$



formal framework for learning from data

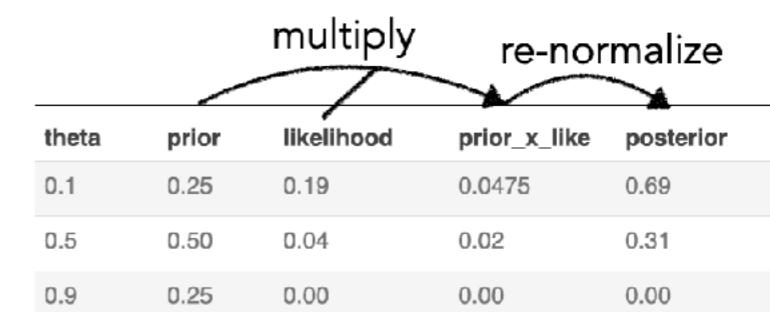
updating our prior belief $p(H)$, to a posterior belief $p(H|D)$ given some data

27

Coin flip example

```

1 # data
2 data = rep(0:1, c(8, 2)) ← 8 tails and 2 heads
3
4 # parameters
5 theta = c(0.1, 0.5, 0.9) ← hypotheses
6
7 # prior
8 prior = c(0.25, 0.5, 0.25) ← prior over the hypotheses
9
10 # likelihood
11 likelihood = dbinom(sum(data == 1), size = length(data), prob = theta)
12 ← binomial distribution
13 # posterior
14 posterior = likelihood * prior / sum(likelihood * prior)
  
```



39

6

Quick recap: What affects the posterior?

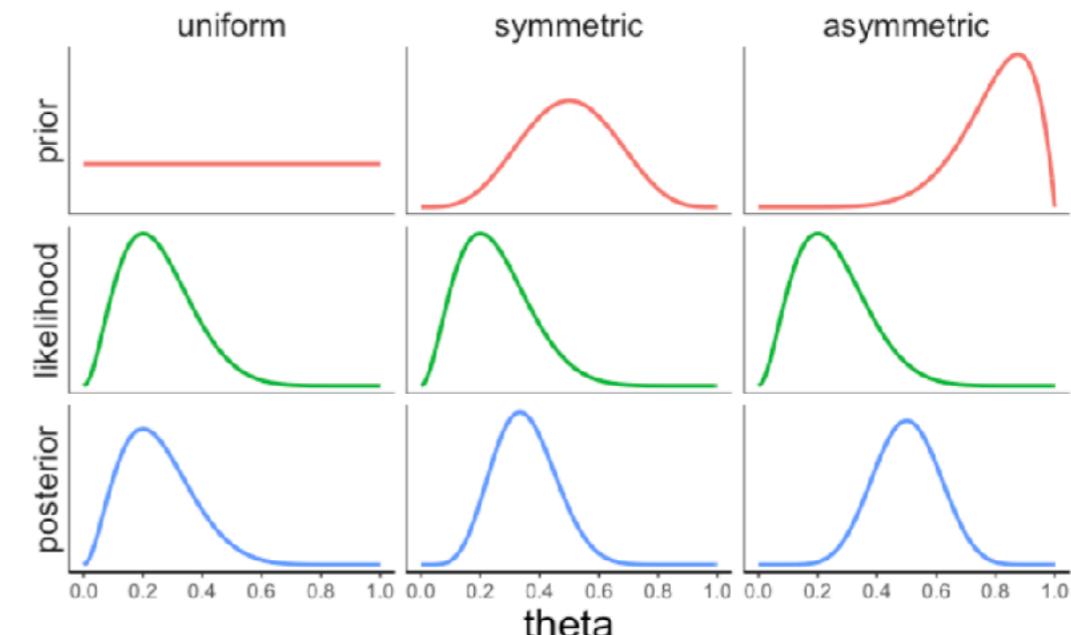
What affects the posterior?

1. the prior over hypotheses
2. the likelihood of the data given each hypothesis

$$\text{Posterior} = \frac{\text{Likelihood} \cdot \text{Prior}}{\text{Normalizing constant}}$$
$$p(H|D) = \frac{p(D|H) \cdot p(H)}{p(D)}$$

The effect of the **prior**

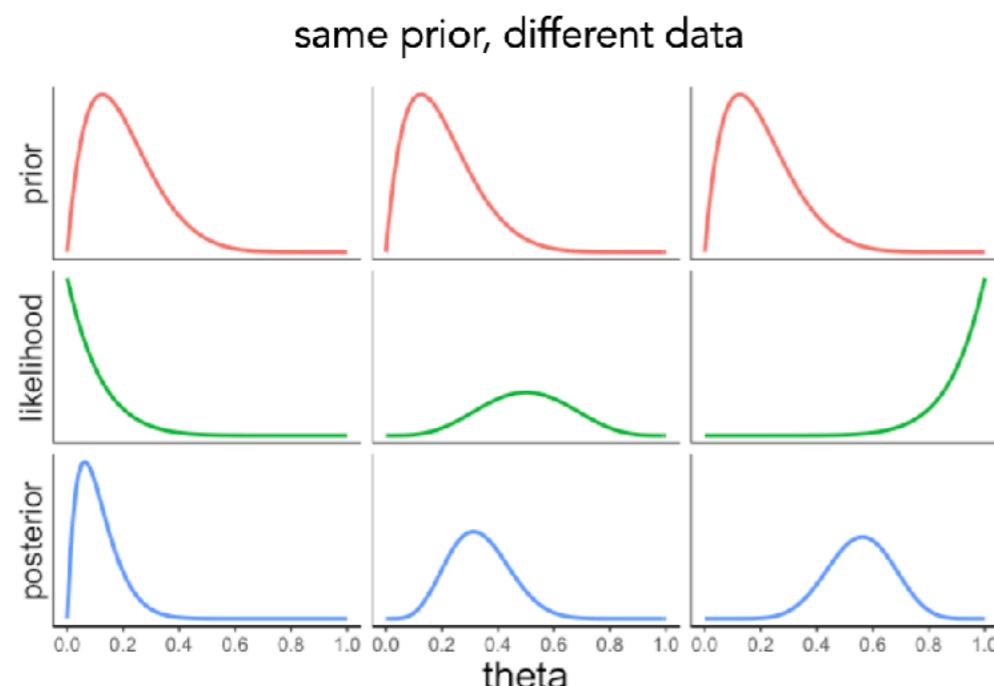
same data, different priors



45

46

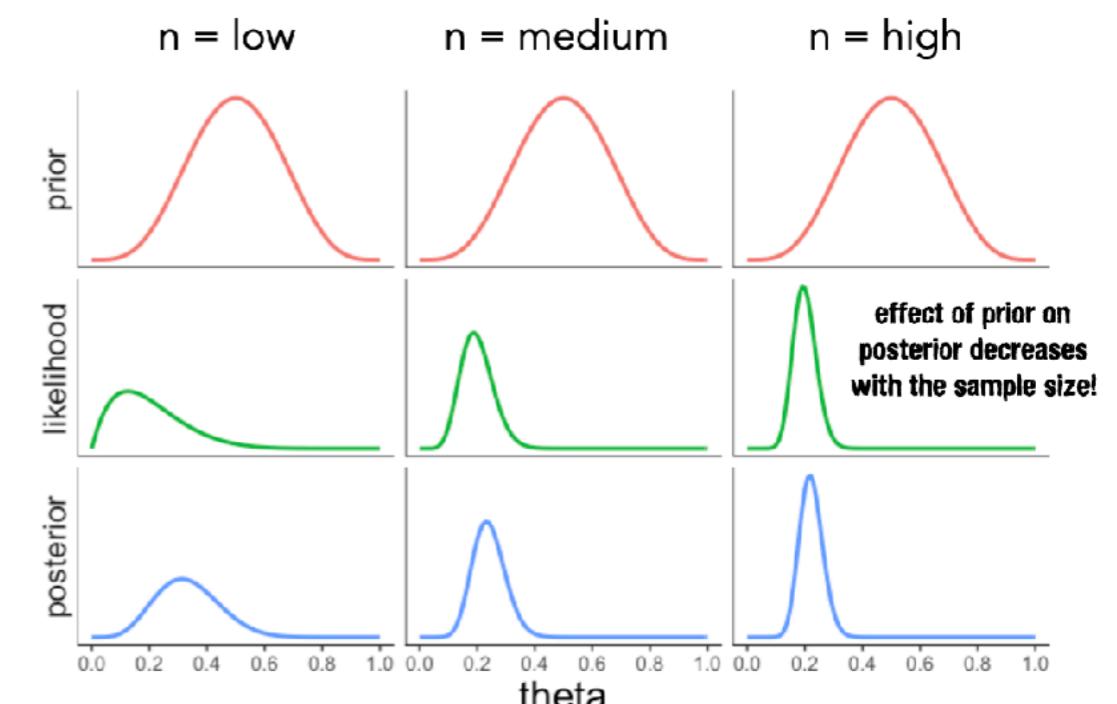
The effect of the **likelihood**



45

46

The effect of **sample size**



47

48

7

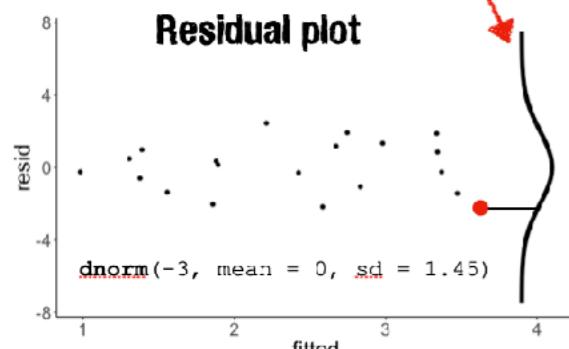
Quick recap: likelihood, prior, inference

Likelihood

Gaussian distribution

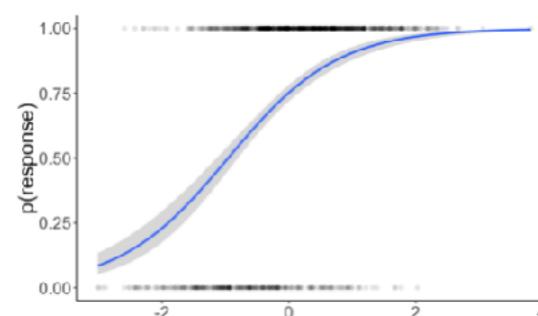
$$Y_i = b_0 + b_1 \cdot x_i + e_i$$

$$e_i \sim \mathcal{N}(0, \sigma)$$



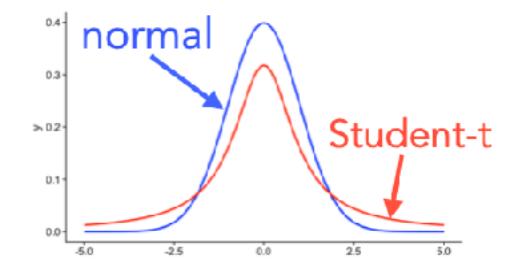
Binomial distribution

```
1 fit.glm = glm(formula = survived ~ 1 + fare,  
2 family = "binomial",  
3 data = df.titanic)
```



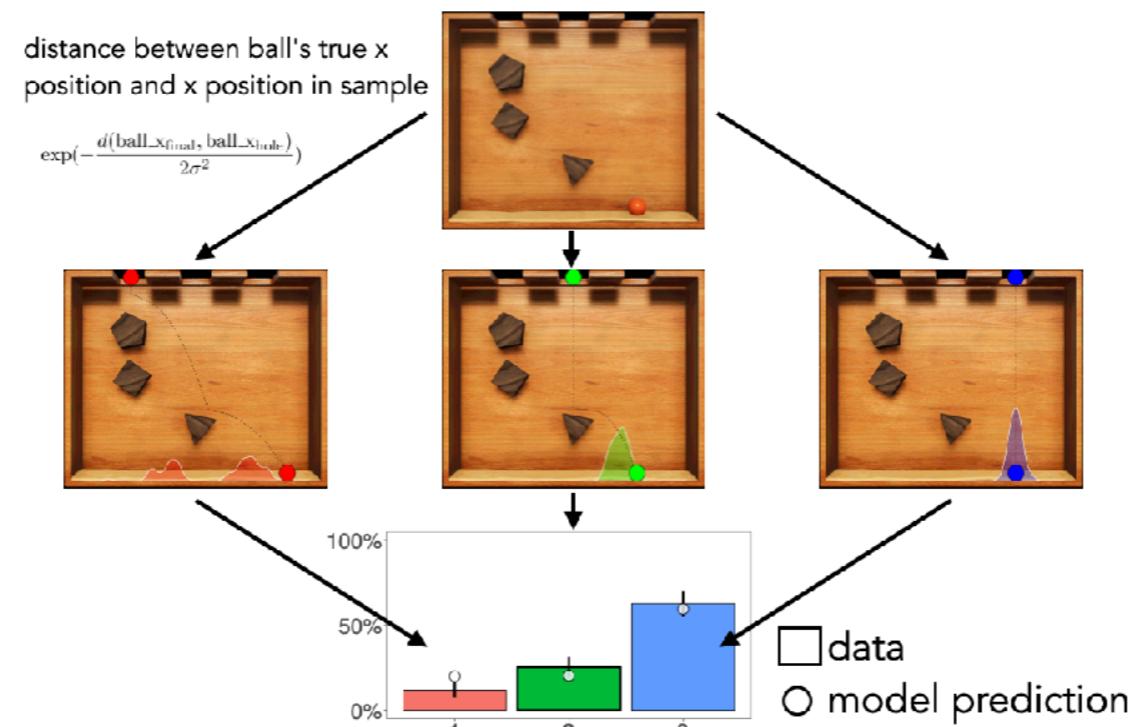
Likelihood

- **Bernoulli:**
 - binary data
 - a single trial
- **Poisson:** count of discrete events
- **Beta-binomial:** like binomial but probability of success may change across trials
- **Student-t:**
 - same as Normal
 - handles greater variability in the data (distribution has **fat tails**)
- ...



53

54

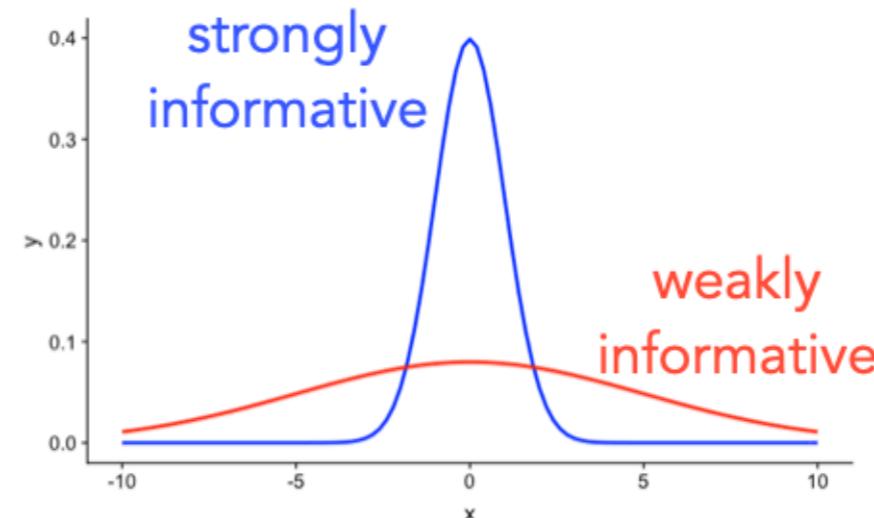


Quick recap: likelihood, prior, inference

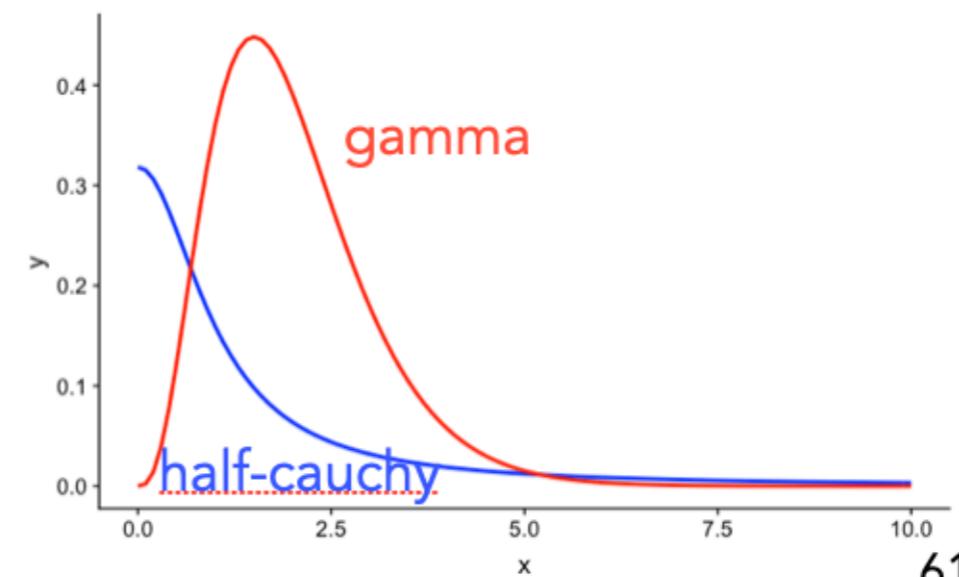
Prior

- **uniform:**
 - continuous or discrete
 - bounded between minimum and maximum
- **gaussian:**
 - sd determines how informative the prior is
- **gamma, half-cauchy:**
 - for parameters we know are positive

for **beta coefficients** in a regression



for **standard deviation** of the Gaussian



Quick recap: likelihood, prior, inference

$$\text{Posterior } p(H | D) = \frac{\text{Likelihood} \cdot \text{Prior}}{p(D)}$$

Normalizing constant

the devil is in the denominator ...

Doing Bayesian inference

Discrete hypothesis space

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

sum over all possibilities

Continuous hypothesis space

$$p(H|D) = \frac{p(D|H) \cdot p(H)}{\int_{-\infty}^{\infty} p(D|H_i) \cdot p(H_i) dH_i}$$

integral over all possibilities

Discretizing the parameters

```
1 # grid  
2 theta = seq(0, 1, 0.01) ← 100 discrete values  
3
```

```
4 # data  
5 data = rep(0:1, c(8, 2))  
6  
7 # calculate posterior
```

```
8 df.prior = tibble(theta = theta,  
9 prior_uniform = dbeta(theta, shape1 = 1, shape2 = 1),  
10 prior_normal = dbeta(theta, shape1 = 5, shape2 = 5),  
11 prior_biased = dbeta(theta, shape1 = 8, shape2 = 2)) %>%
```

```
12 pivot_longer(cols = -theta,  
names_to = "prior_index",  
values_to = "prior") %>%
```

```
13 mutate(likelihood = dbinom(sum(data == 1),  
size = length(data),  
prob = theta)) %>%
```

```
16 group_by(prior_index) %>%
```

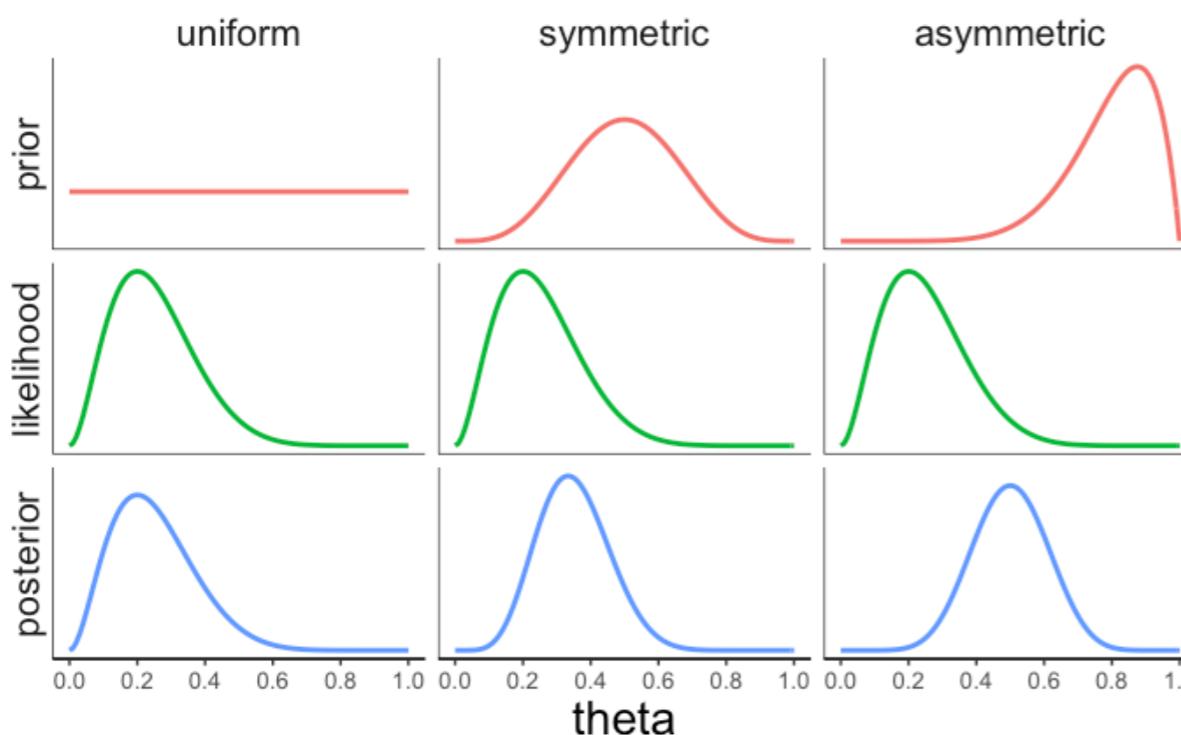
```
17 mutate(posterior = likelihood * prior / sum(likelihood * prior)) %>%
```

```
18 ungroup() %>%
```

```
19 pivot_longer(cols = -c(theta, prior_index),  
names_to = "index",  
values_to = "value")
```

for 3 variables, we would already
need 1 Mio combinations

The CURSE of
dimensionality



Inference via sampling

- we cannot directly calculate the probability of the posterior (because it might have a pretty weird shape)
- **but:** we can draw random samples from the posterior
- we can then use our data wrangling and visualization skills to make inferences based on these samples

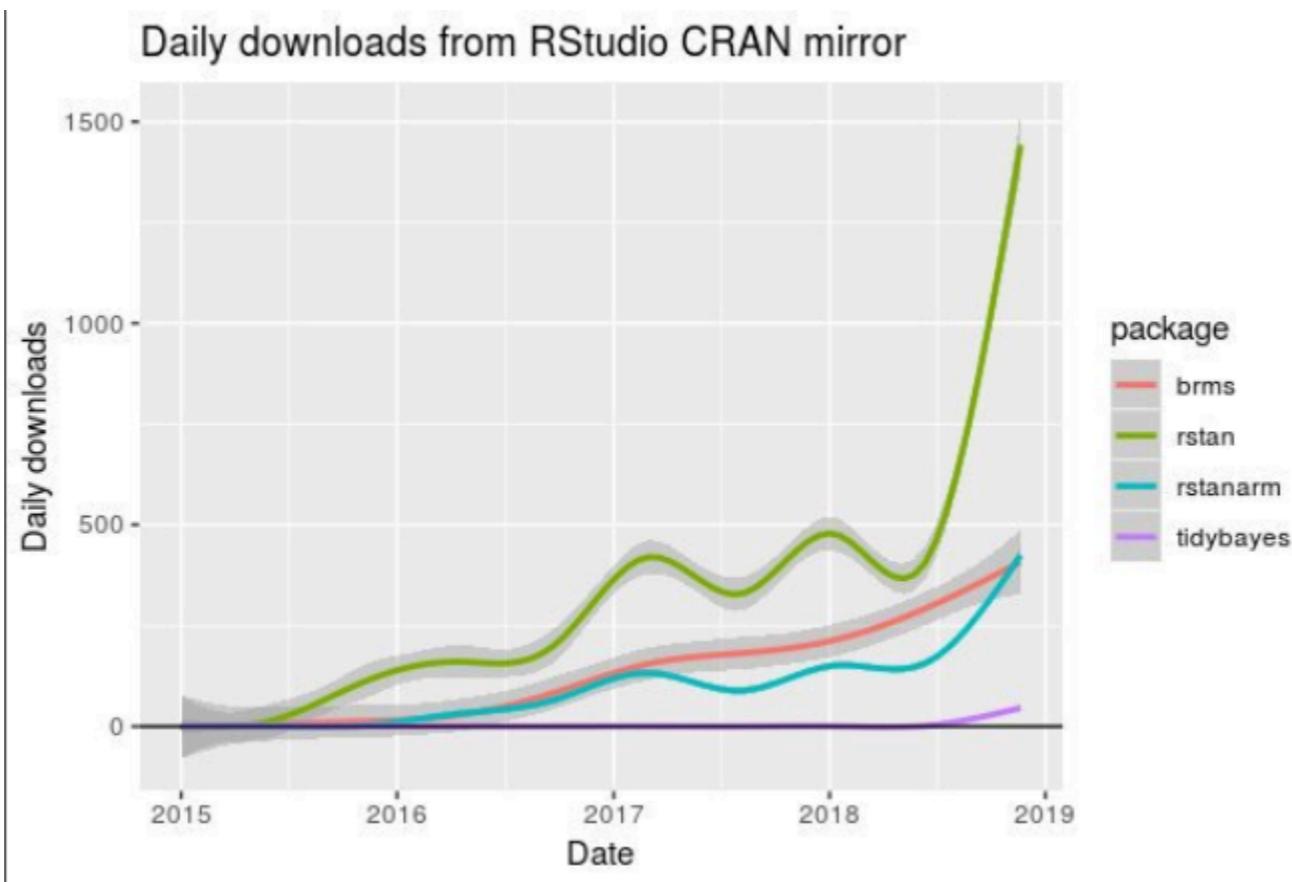
It's as if ...

we don't have **pnorm()**

but we do have **rnorm()**

Inference via sampling

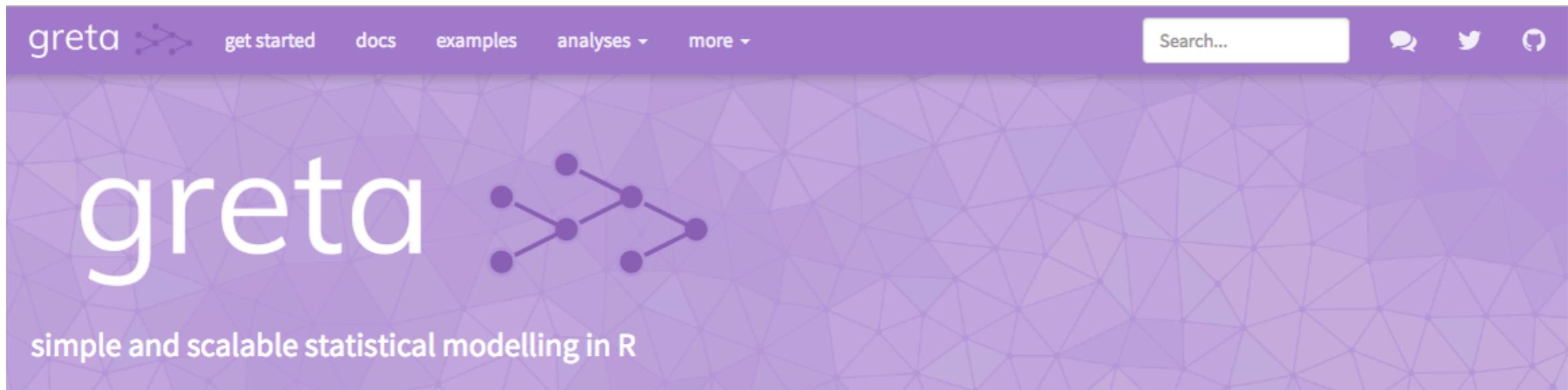
- Bayesian data analysis is becoming more popular because:
 - computers are getting more powerful
 - inference techniques are getting better
 - software packages become easier to use



Doing Bayesian data analysis

Software packages

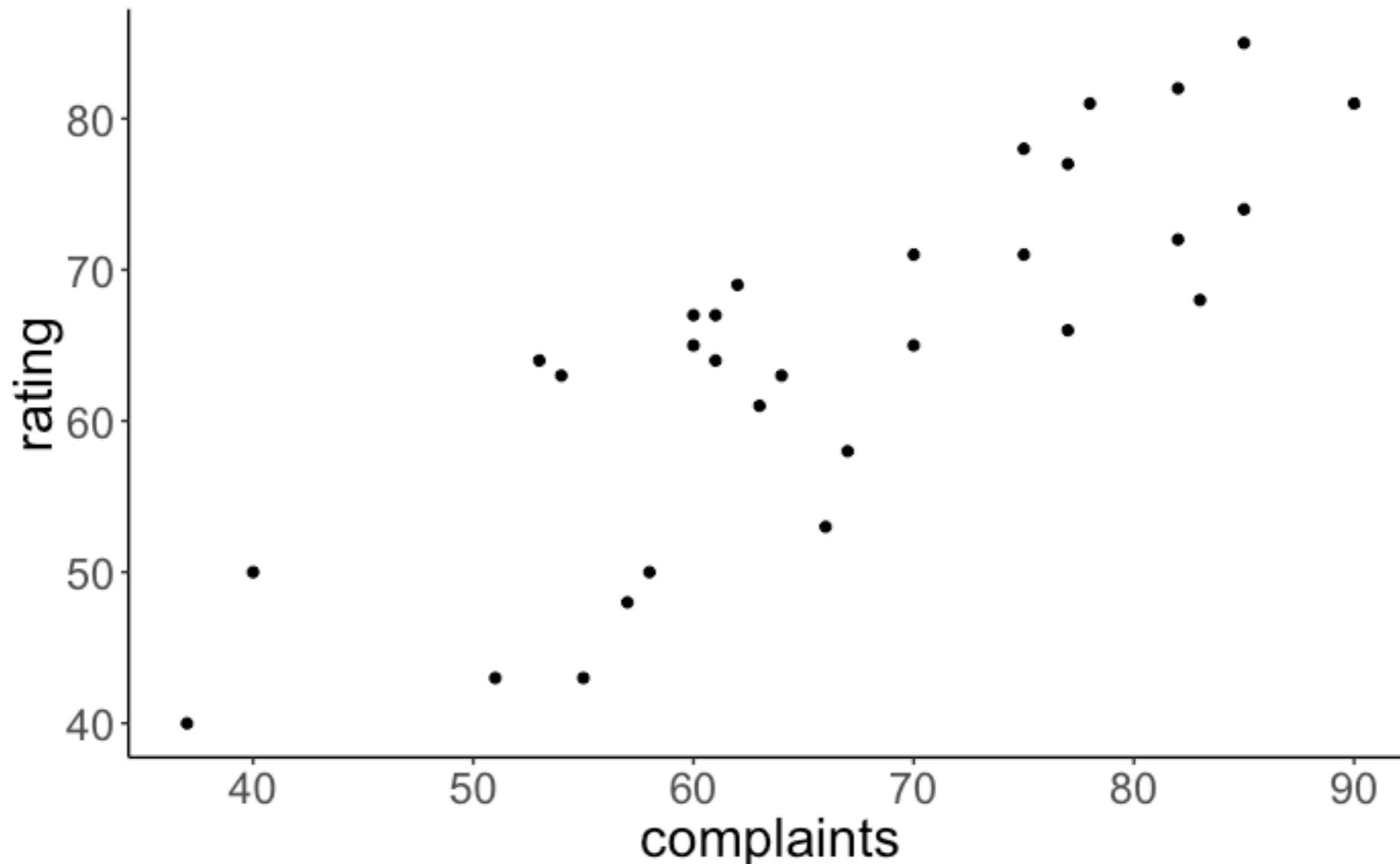
```
library("greta")
```



- 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 ...)

Attitude data set

What's the relationship between how well an employee handles complaints and their overall rating?



Frequentist analysis

Frequentist analysis

```
1 # fit model
2 fit = lm(formula = rating ~ 1 + complaints,
3           data = df.attitude)
4
5 # print summary
6 fit %>% summary()
```

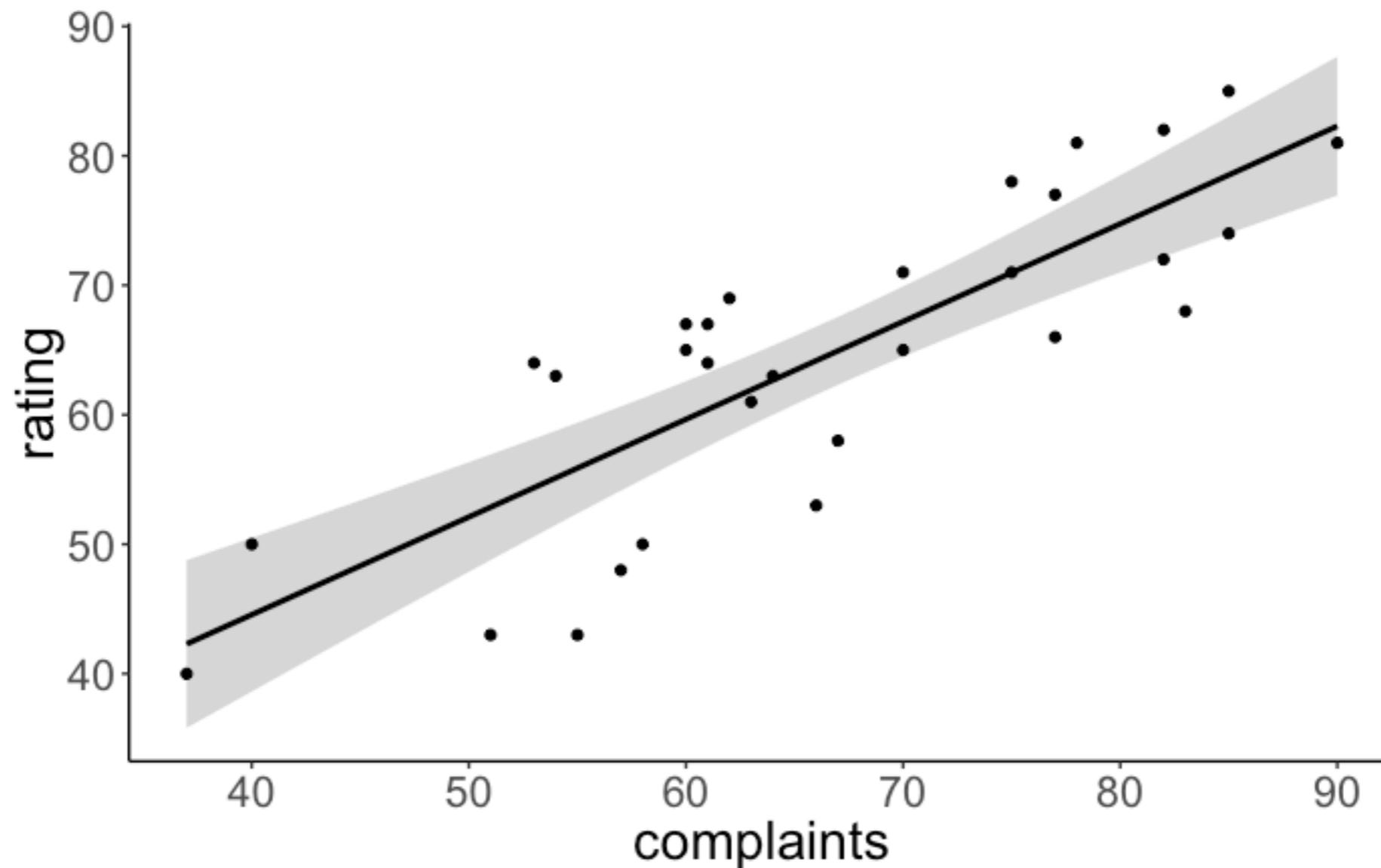
```
Call:
lm(formula = rating ~ 1 + complaints, data = df.attitude)

Residuals:
    Min      1Q  Median      3Q     Max 
-12.8799 -5.9905  0.1783  6.2978  9.6294 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 14.37632   6.61999   2.172   0.0385 *  
complaints   0.75461   0.09753   7.737 1.99e-08 *** 
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.993 on 28 degrees of freedom
Multiple R-squared:  0.6813, Adjusted R-squared:  0.6699 
F-statistic: 59.86 on 1 and 28 DF,  p-value: 1.988e-08
```

Visualize model predictions



Best-fitting regression line with confidence interval

Bayesian analysis

Model specification

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

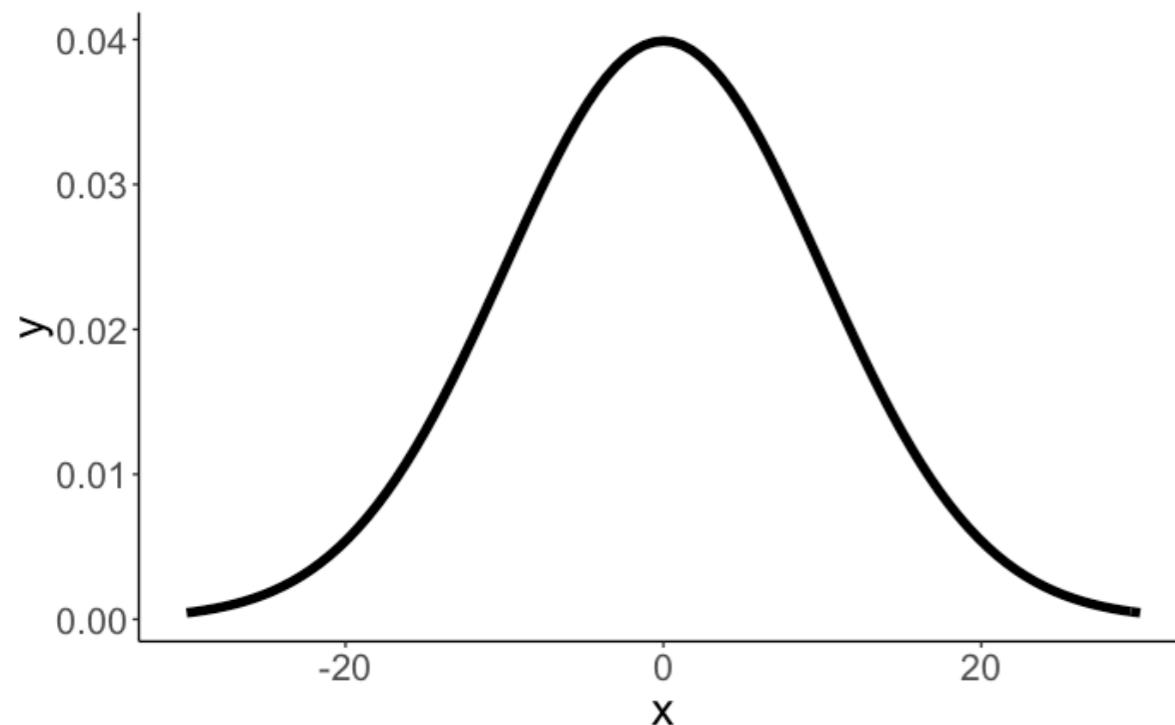
← **build the model**

← **Gaussian likelihood**

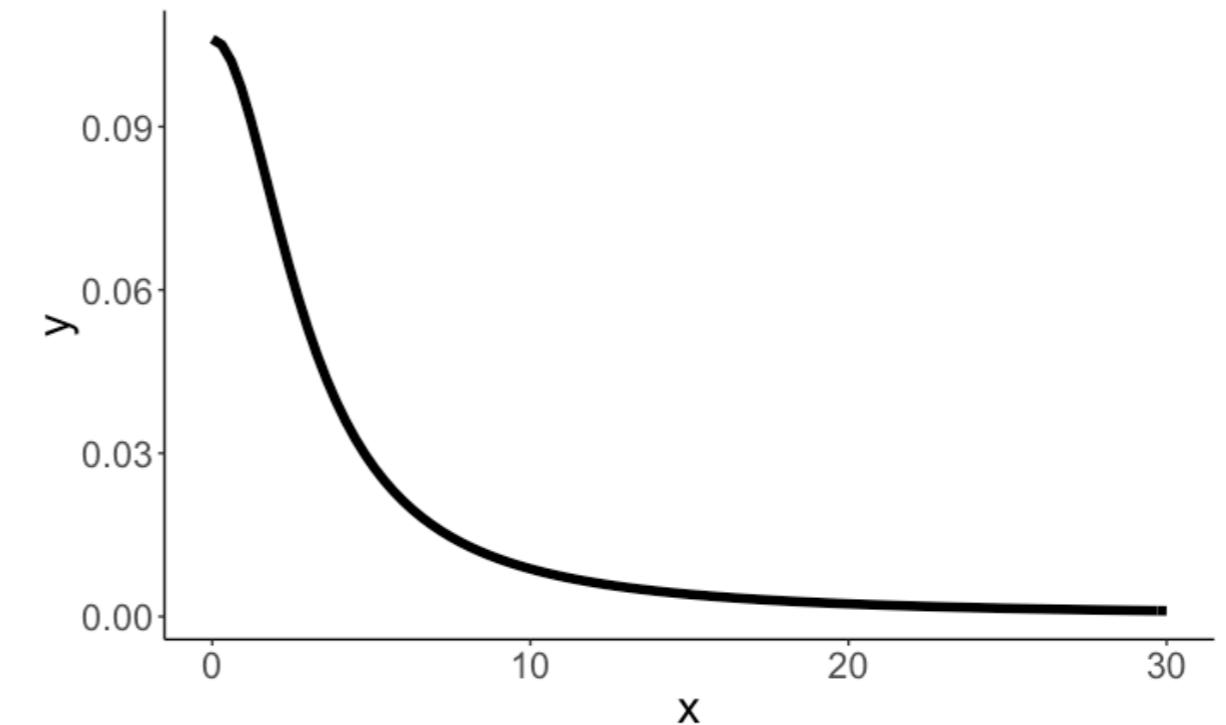
← **linear combination**

← **priors**

Priors



**Gaussian prior on
intercept and coefficient**

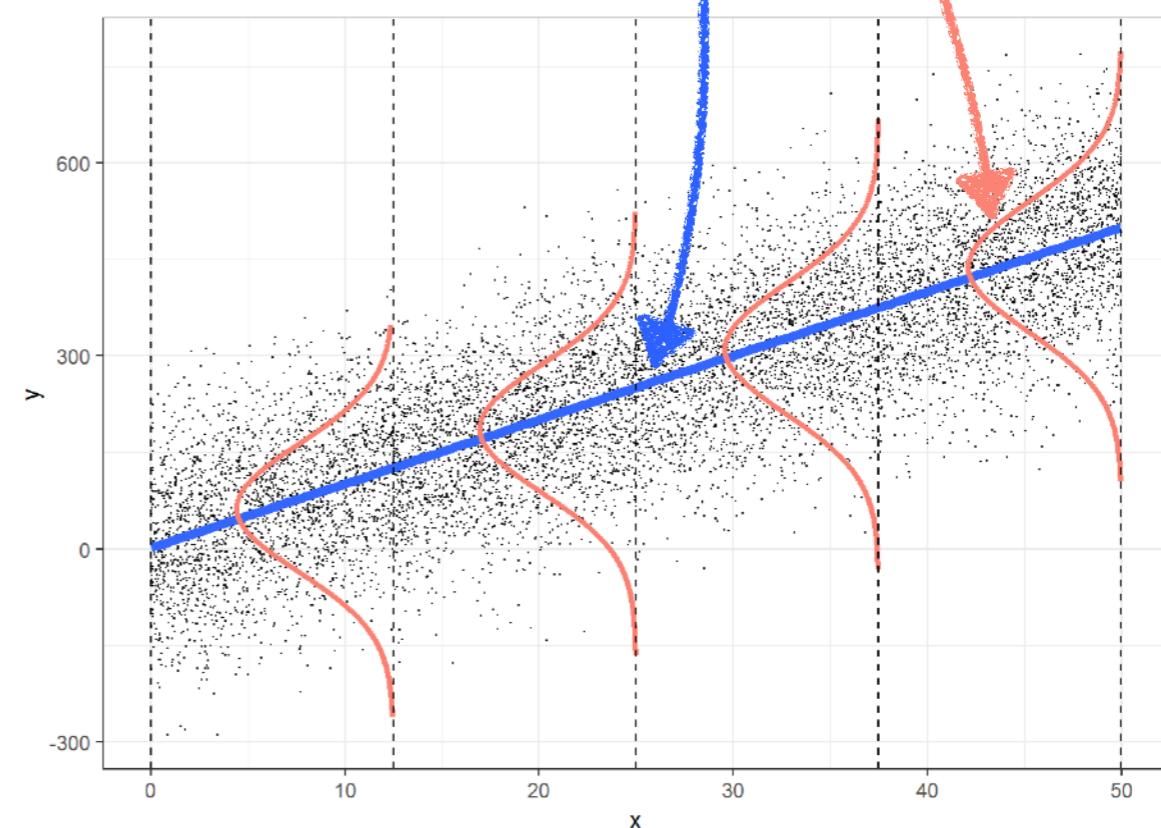
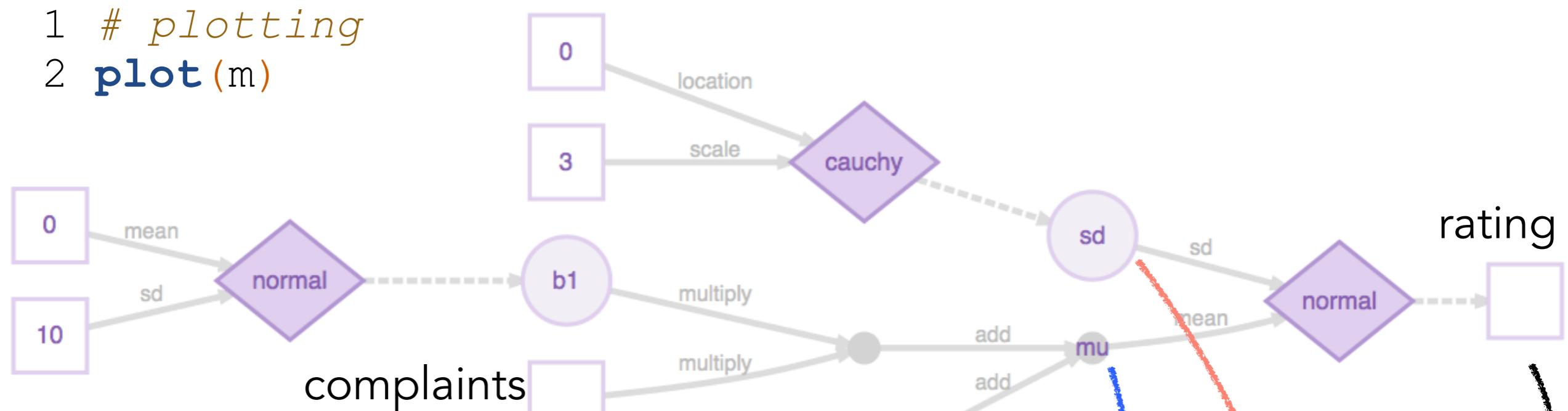


**Truncated Cauchy prior on
the standard deviation**

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

Graphical representation of the model

```
1 # plotting  
2 plot(m)
```



Inference via sampling

Markov Chain
Monte Carlo
inference

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

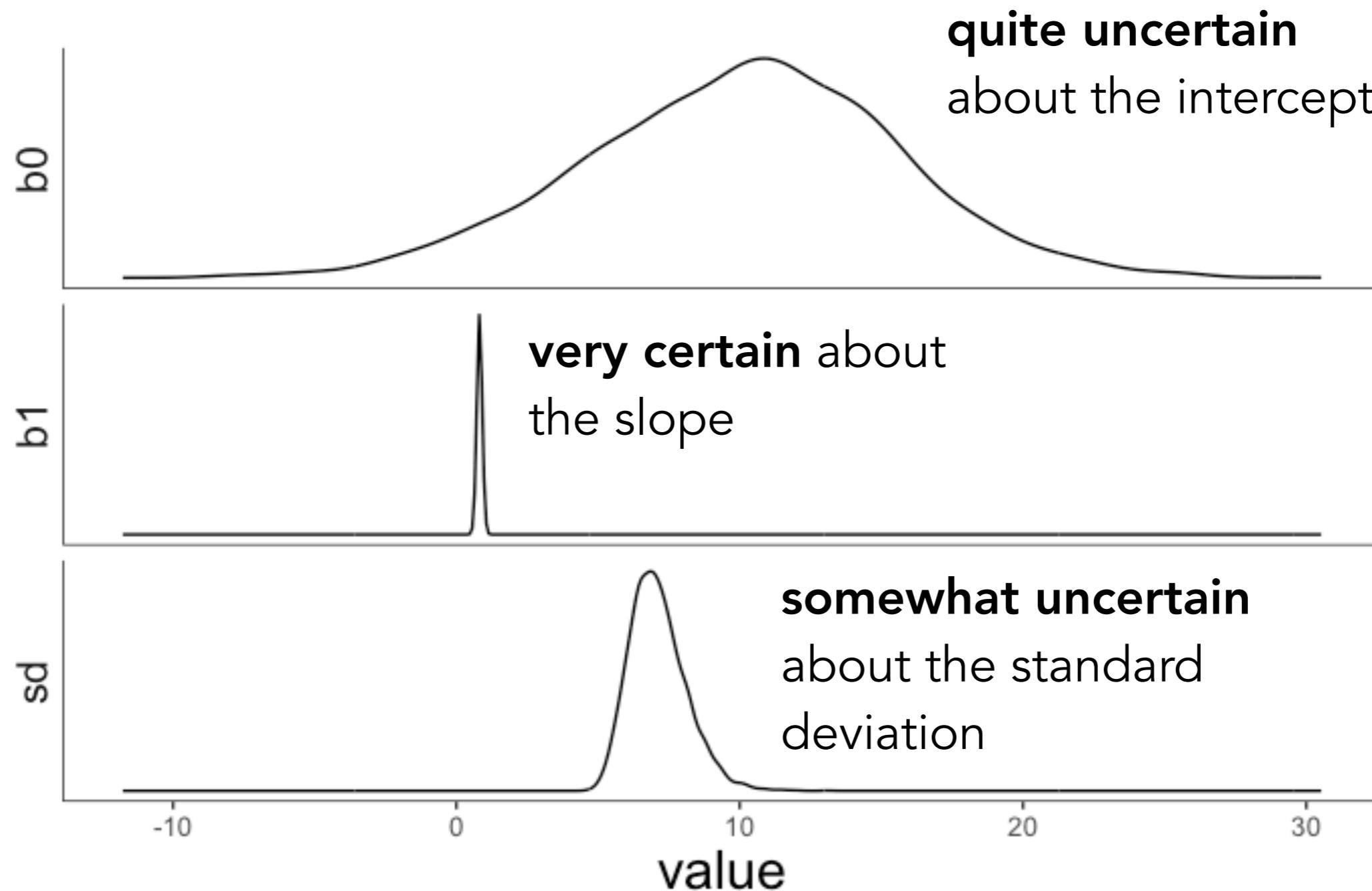
chain	iteration	draw	b0	b1	sd
1	1	1	6.08	0.87	7.60
1	2	2	1.12	0.95	7.66
1	3	3	-1.83	0.99	7.01
1	4	4	-4.23	1.02	6.64
1	5	5	3.26	0.87	7.96
1	6	6	-1.04	0.98	7.67
1	7	7	-0.83	0.97	10.12
1	8	8	-1.41	0.97	8.02
1	9	9	9.46	0.81	6.30
1	10	10	10.02	0.84	6.57

each of these is a solution
for explaining the data

nice visualization of MCMC
samplers

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

Visualize the posterior



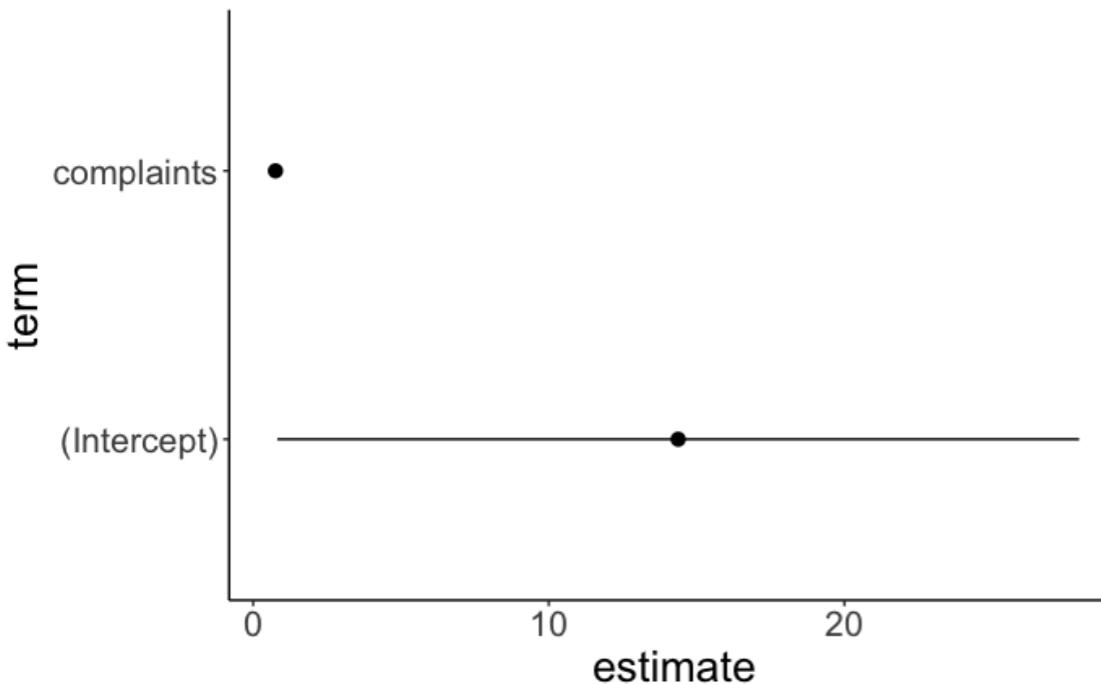
This is the solution of a Bayesian analysis.

A posterior distribution over each parameter in our model.

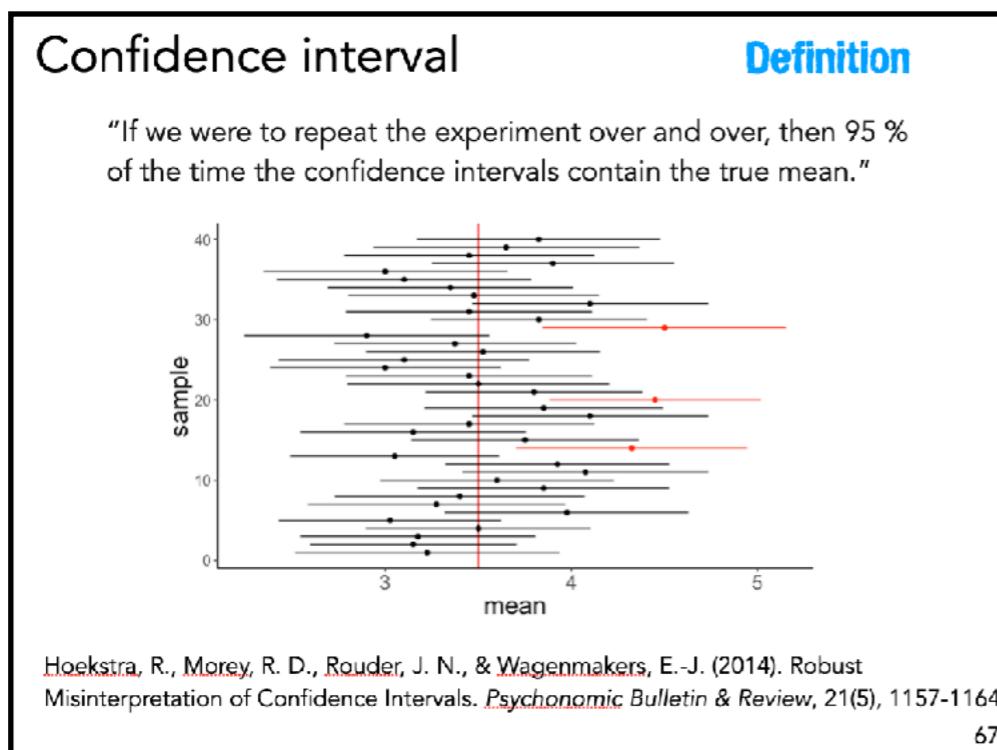
We can use this to visualize model predictions, and to test hypotheses. 26

Confidence interval

```
1 fit.lm = lm(formula = rating ~ 1 + complaints,  
2                      data = df.attitude)
```

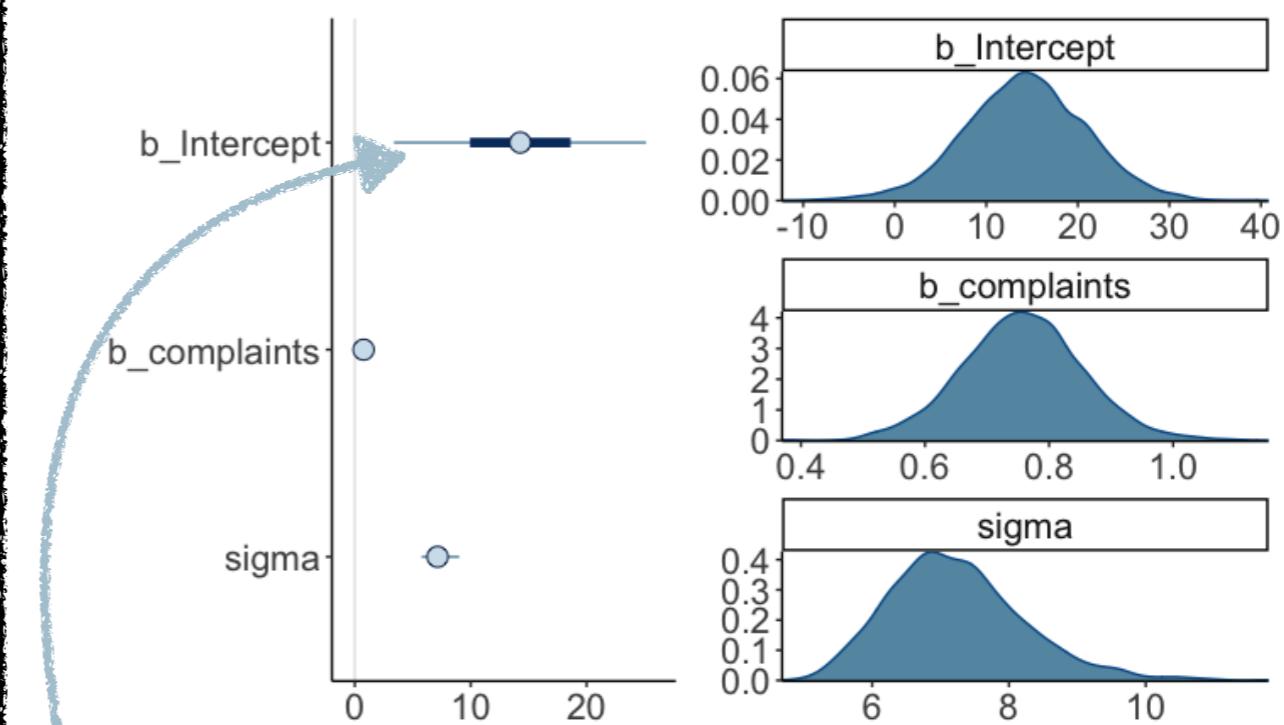


point estimate + confidence interval



Credible interval

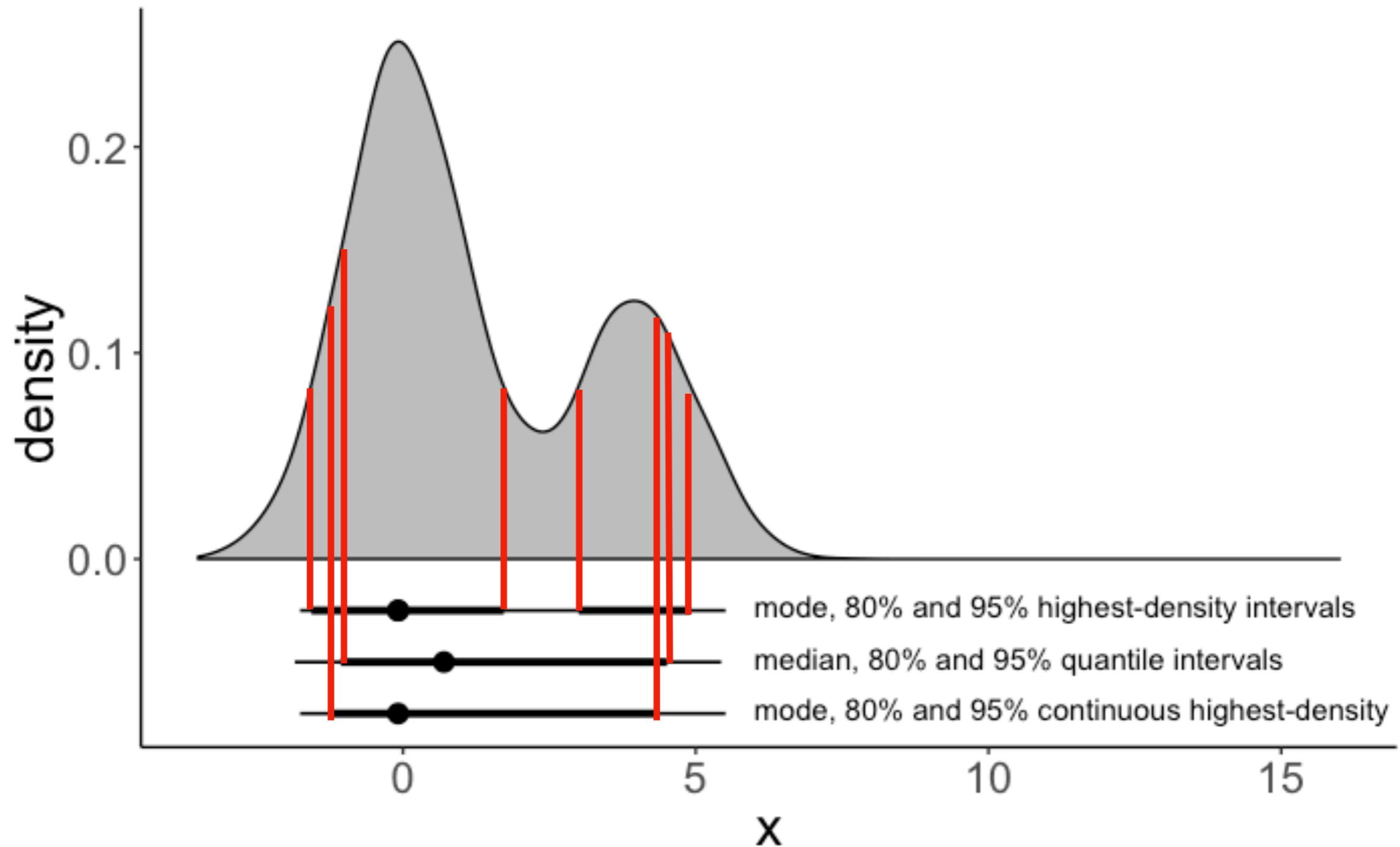
```
1 fit.brm = brm(formula = rating ~ 1 + complaints,  
2                      data = df.attitude)
```



full posterior distribution over each parameter

with 90% the true parameter lies within this range

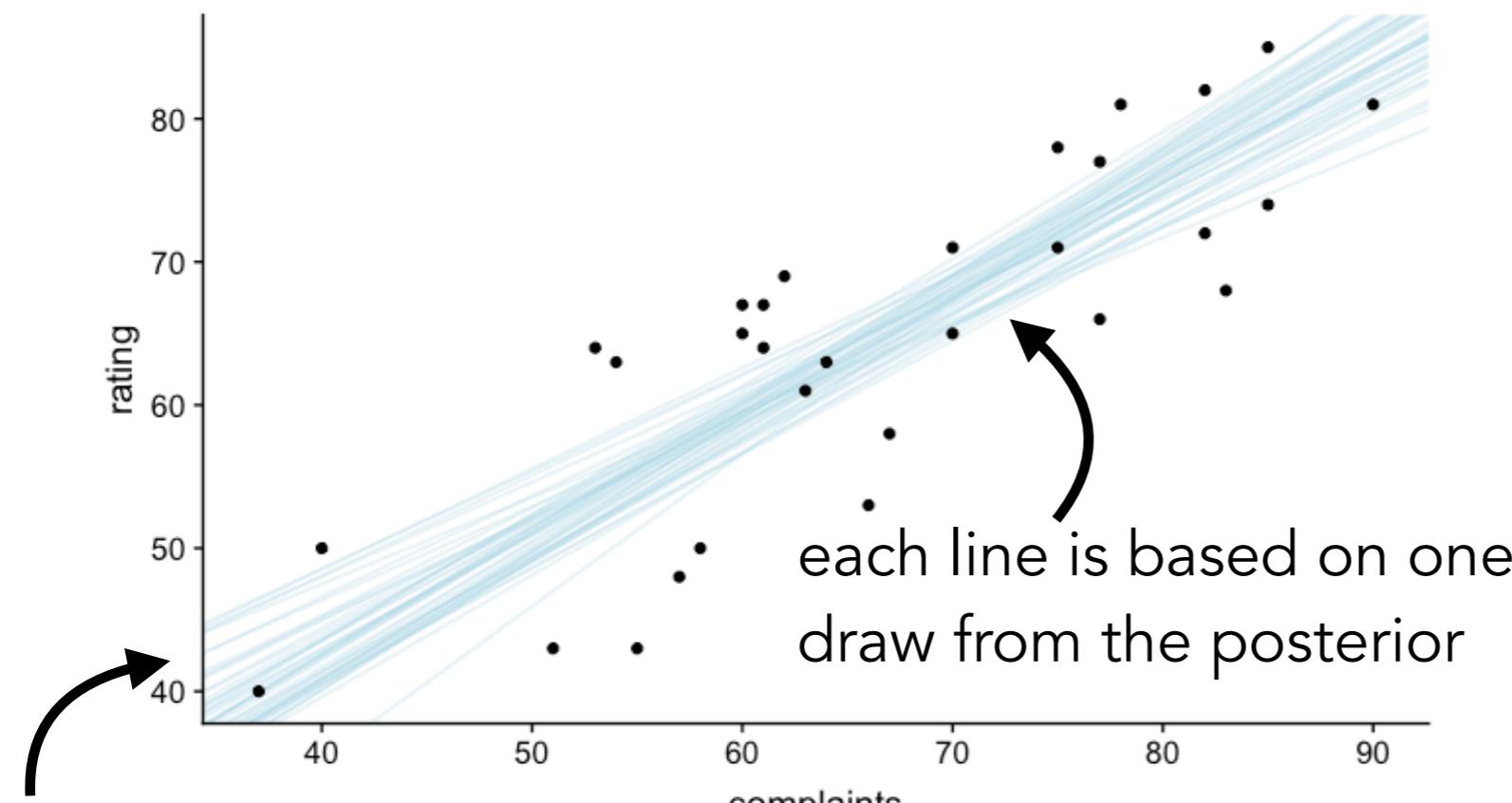
Different kinds of credible intervals



ways of summarizing the posterior distribution

Visualize the model predictions

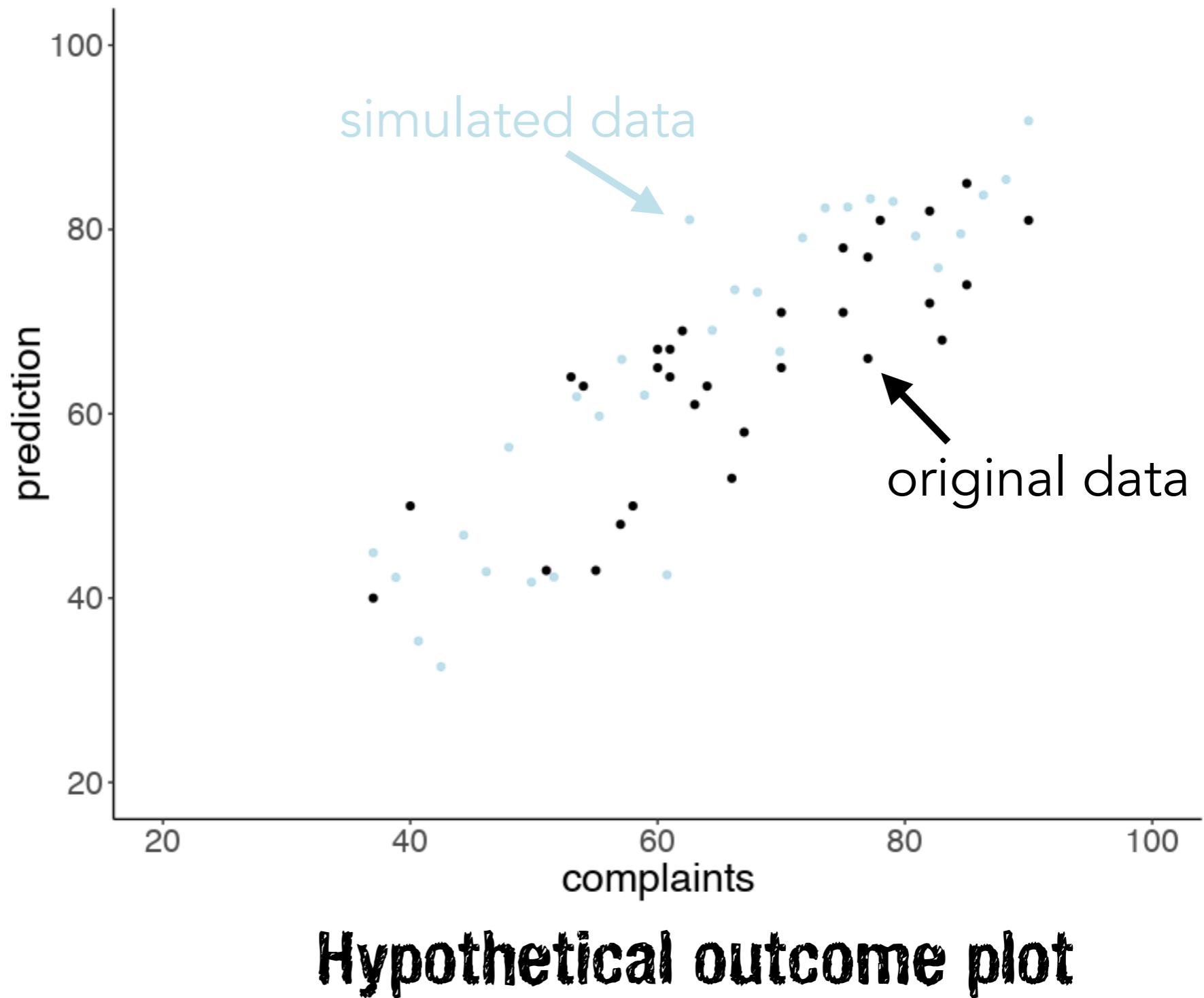
```
1 ggplot(data = df.attitude,
2         mapping = aes(x = complaints,
3                         y = rating)) +
4   geom_abline(data = df.draws %>%
5               sample_n(size = 50),
6               aes(intercept = b0,
7                   slope = b1),
8               alpha = 0.3,
9              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

Posterior predictive check

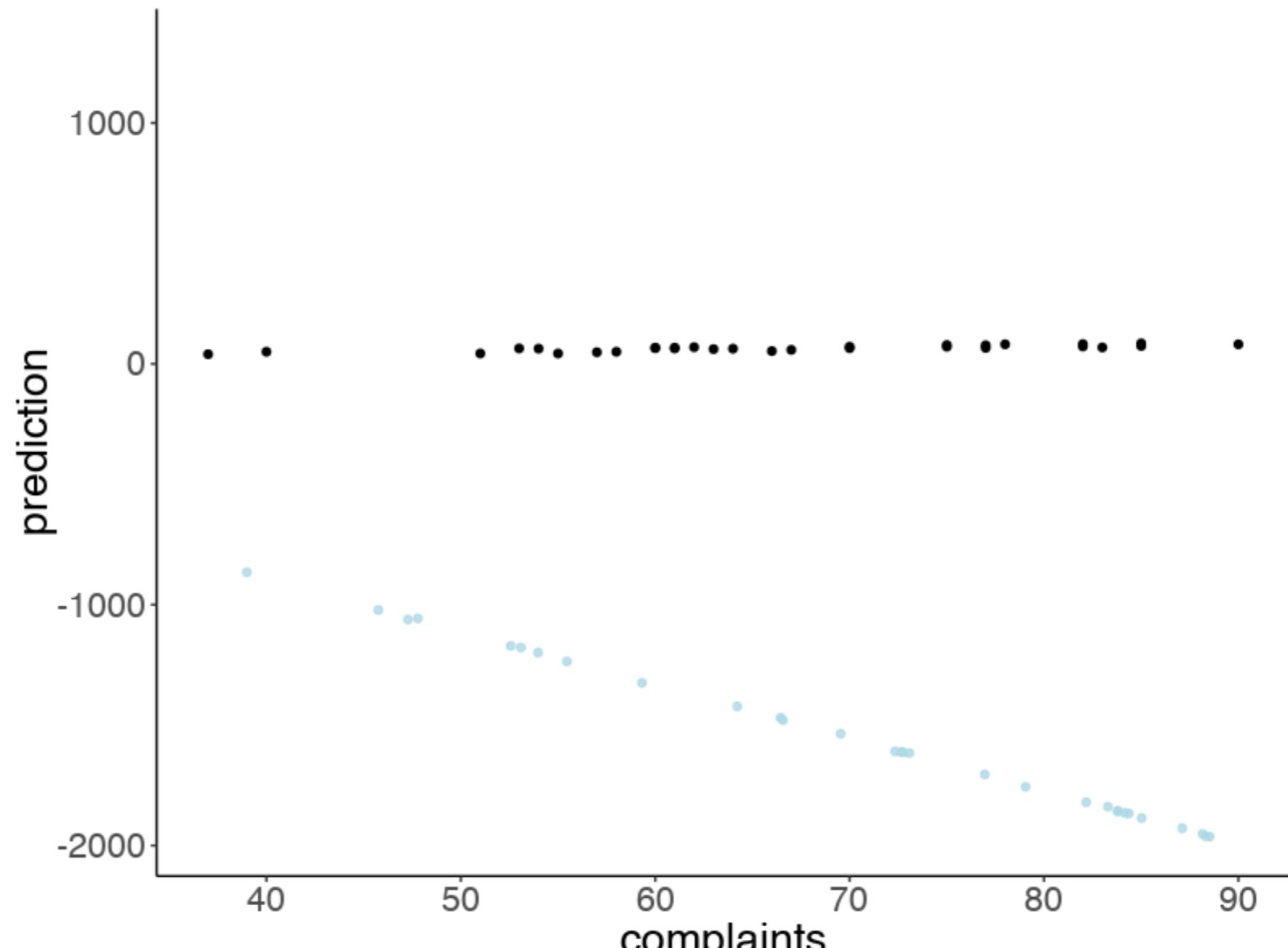
1. sample parameters from the **posterior distribution**
2. generate data using these parameters (using the likelihood function)



Hypothetical outcome plot

Prior predictive check

1. sample parameters from the **prior distribution**
2. generate data using these parameters (using the likelihood function)



Hypothetical outcome plot



Paul Bürkner

Doing Bayesian data analysis with BRMS

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.

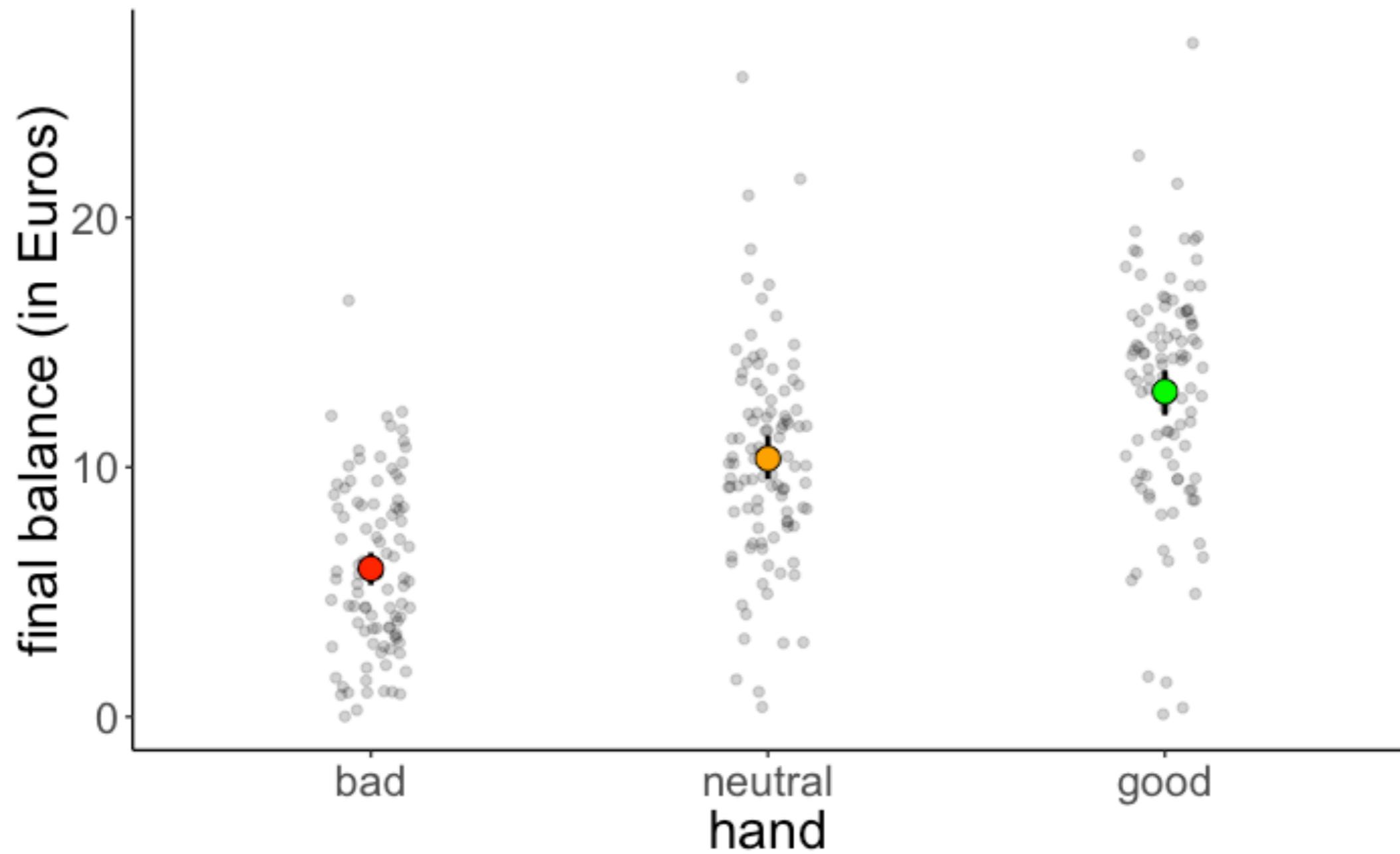
Stan

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

Poker data



Using lm()

```
1 fit.lm_poker = lm(formula = balance ~ 1 + hand,  
2                      data = df.poker)  
3  
4 fit.lm_poker %>% 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.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	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	1-95% 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	1-95% 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

**almost identical
results!**

What about the priors?

```
1 fit.brm_poker = 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.brm_poker_full = 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")  
10 ), ← priors  
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   ), ← initialization  
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
```

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

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

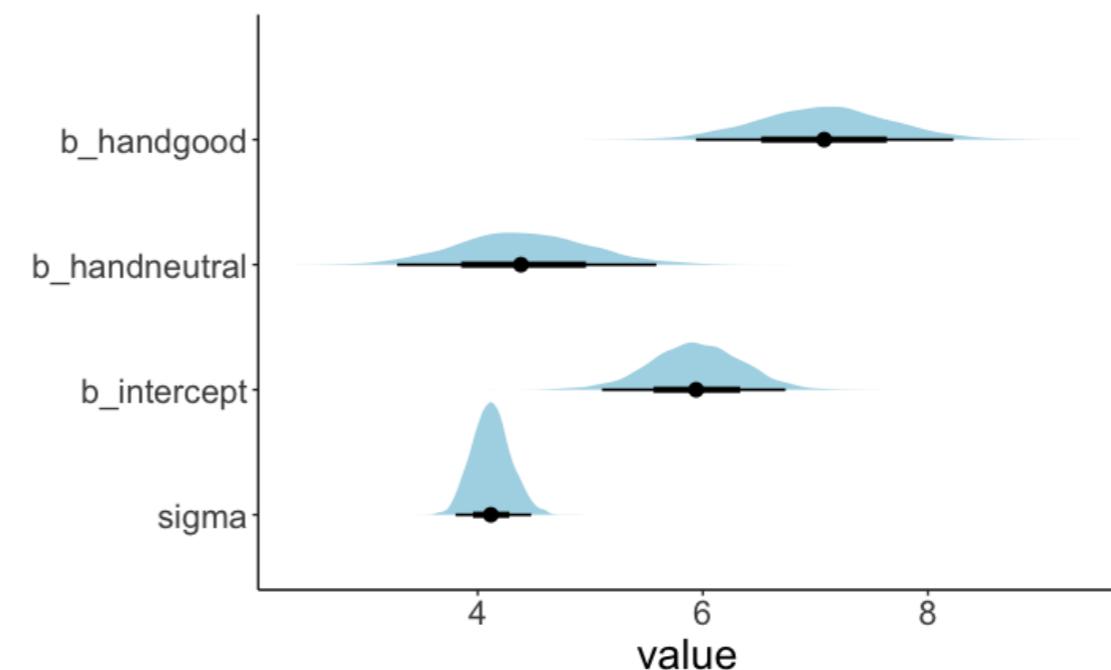


summary of posterior

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



Testing hypotheses

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/>

Nice paper ...

Bayesian regression modeling (for factorial designs): A tutorial

Michael Franke & Timo Roettger

Generalized linear mixed models are handy tools for statistical inference, and Bayesian approaches to applying these become increasingly popular. This tutorial provides an accessible, non-technical introduction to the use and feel of Bayesian mixed effects regression models. The focus is on data from a factorial-design experiment.

This tutorial should take you about 1.5 hours.

Motivation & intended audience

We would like to thank Oliver Bott, Joseph Cassilas, Artur Czeszumski, Fabian Dablander, Judith Degen, Elisa Kreiss, and Bodo Winter for their invaluable comments and suggestions on an earlier draft.

on canvas

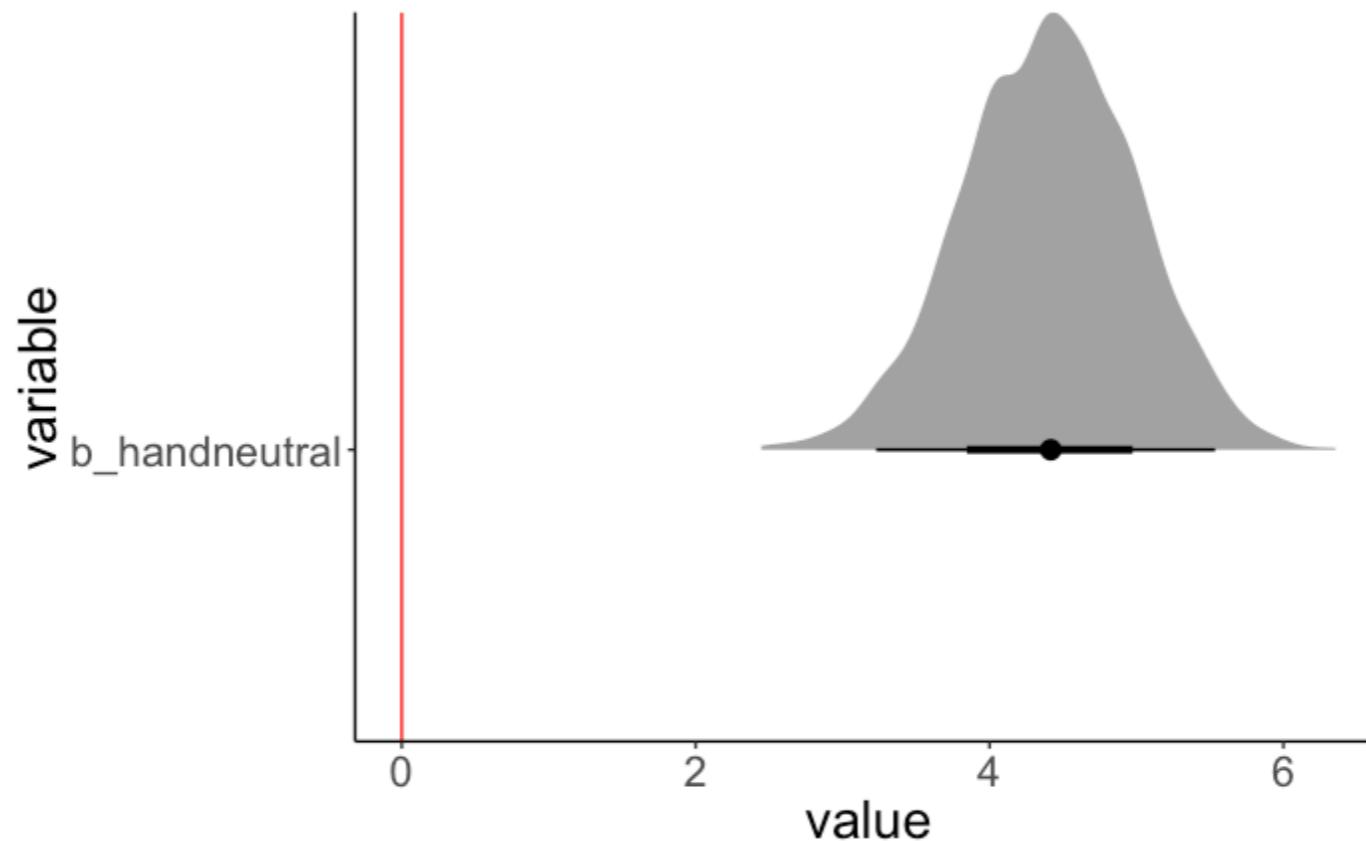
Franke and Roettger - Bayesian regression modeling (for factorial design.pdf

see also

<https://michael-franke.github.io/intro-data-analysis/Chap-04-03-predictors-multiple-predictors.html>

Asking questions based on the posterior

Do neutral hands earn more money than bad hands?



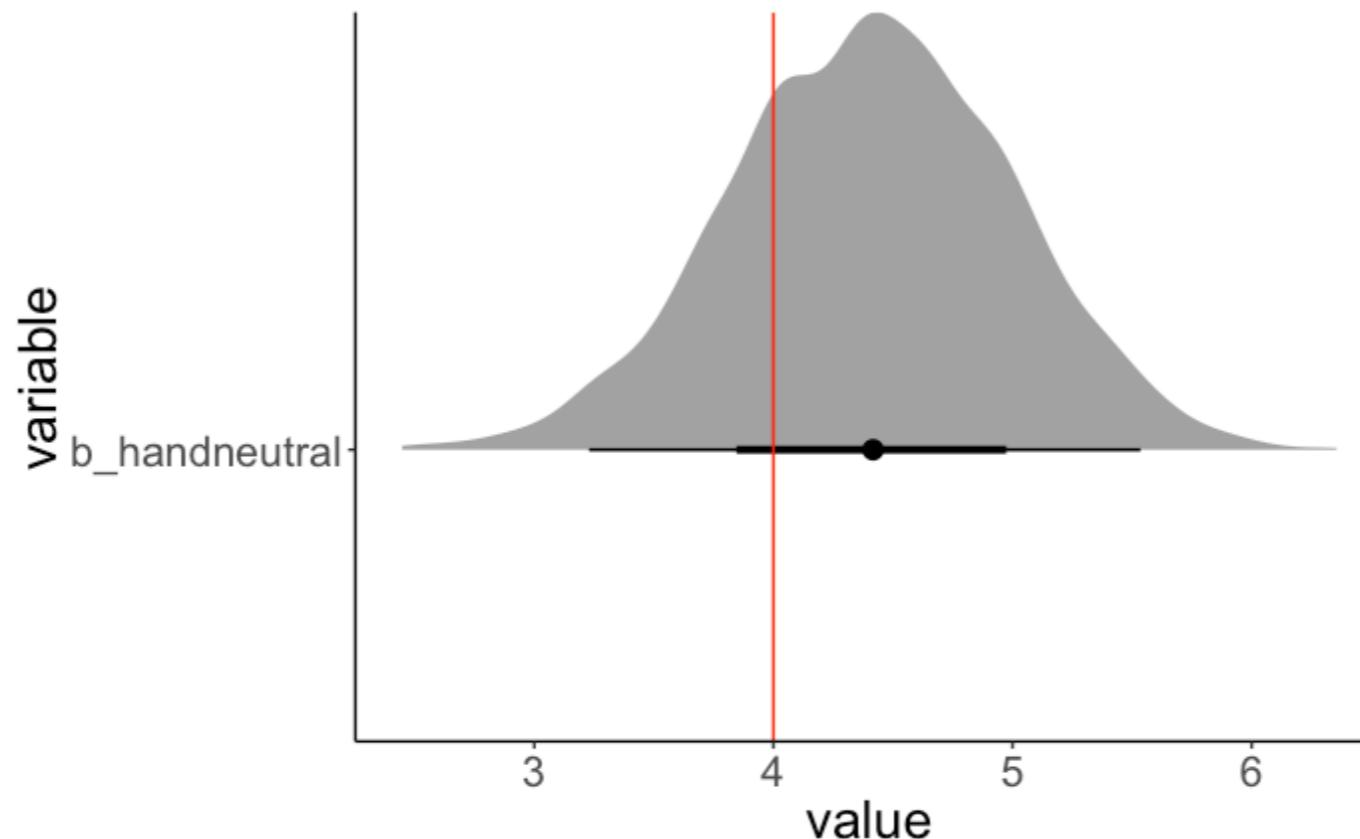
What's the probability that `handneutral` is less than 0?

```
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?



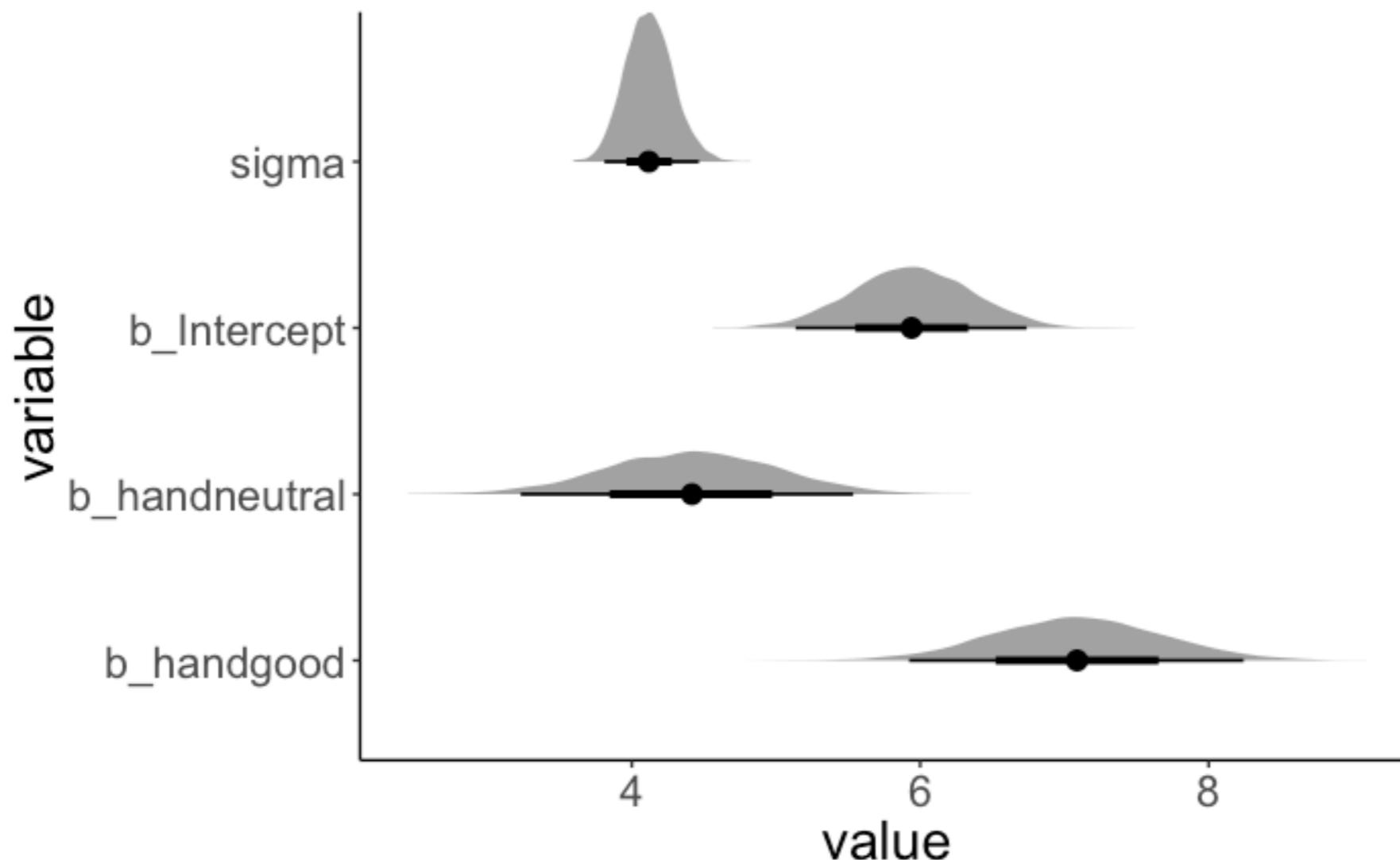
What's the probability that `handneutral` is **more than 4**?

```
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?

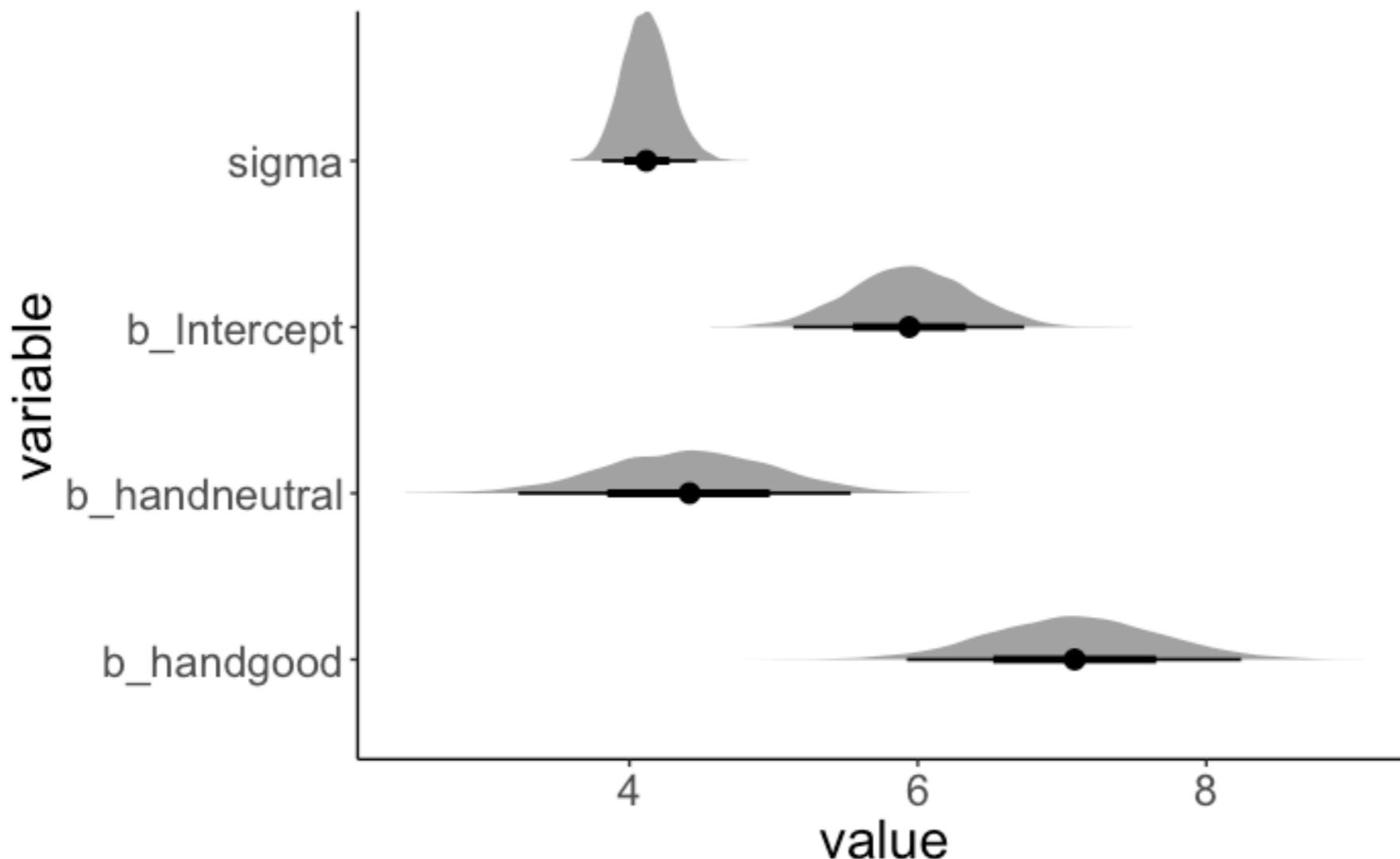


```
1 hypothesis(fit.brm,  
2           hypothesis = "handgood + Intercept > 2 * Intercept")
```

$p = 0.89$

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
```

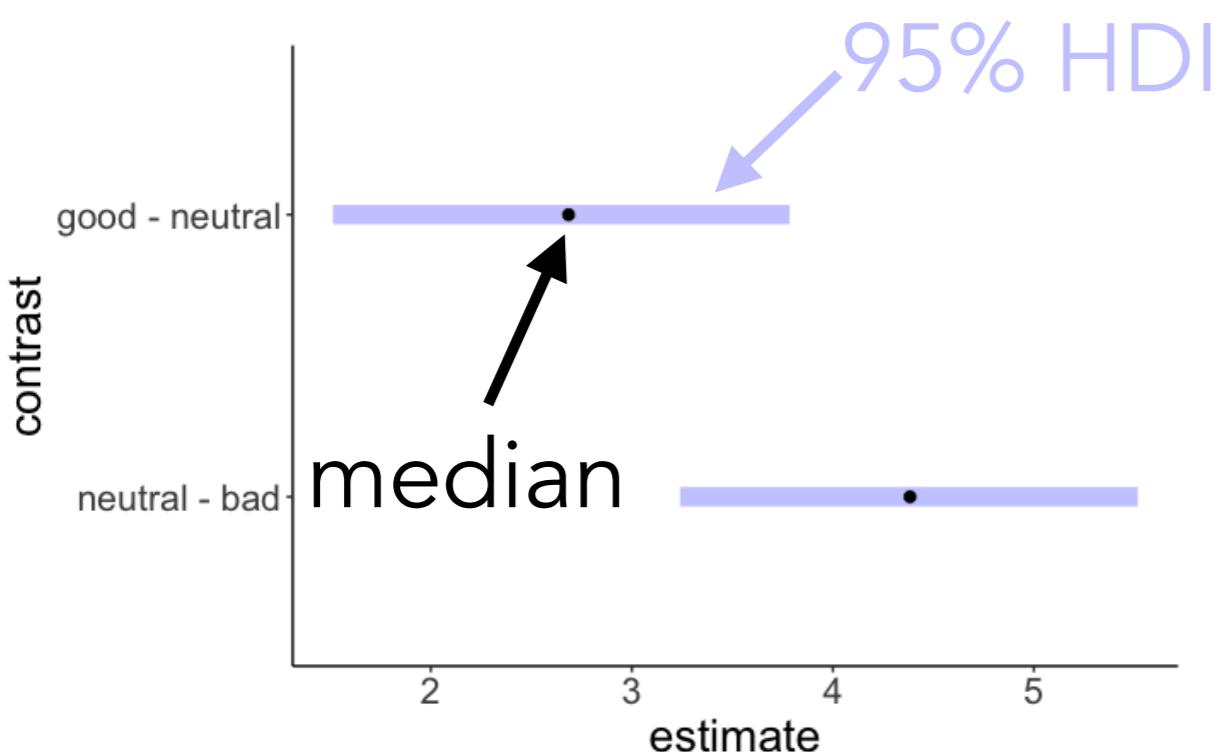
Point estimate displayed: median
HPD interval probability: 0.95

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

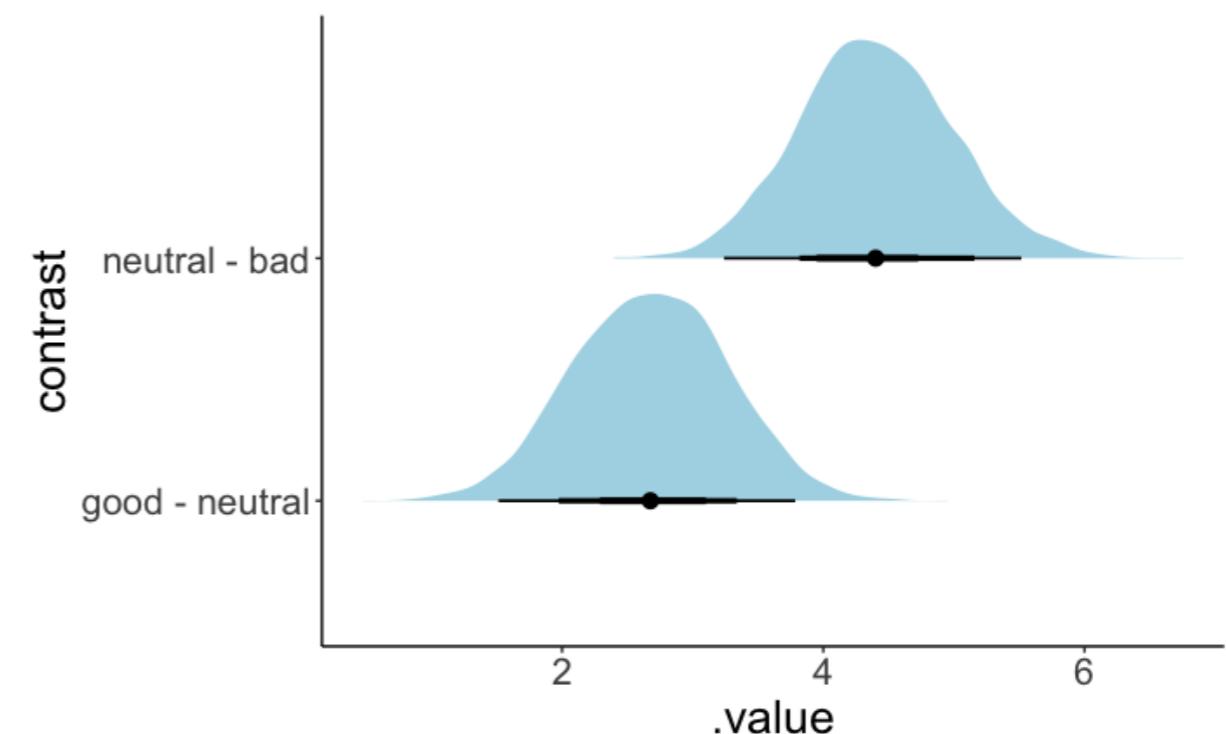
Point estimate displayed: median
HPD interval probability: 0.95

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))
```

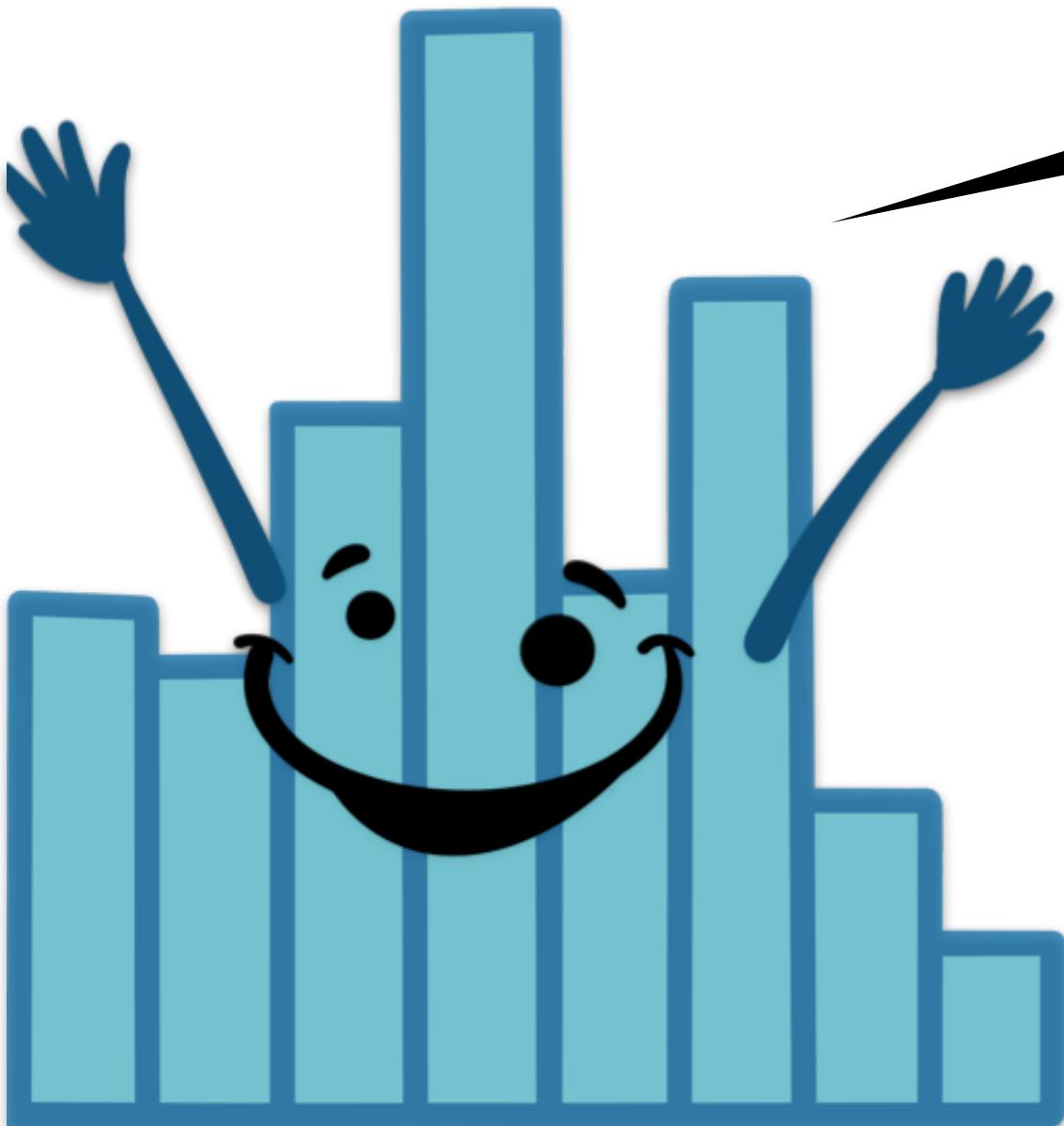


mean, 50% HDI, 75% HDI, 95% HDI

We're listening to
"Lantern (Radio Edit)"
by "Les Gordon"

02:00

stretch break!

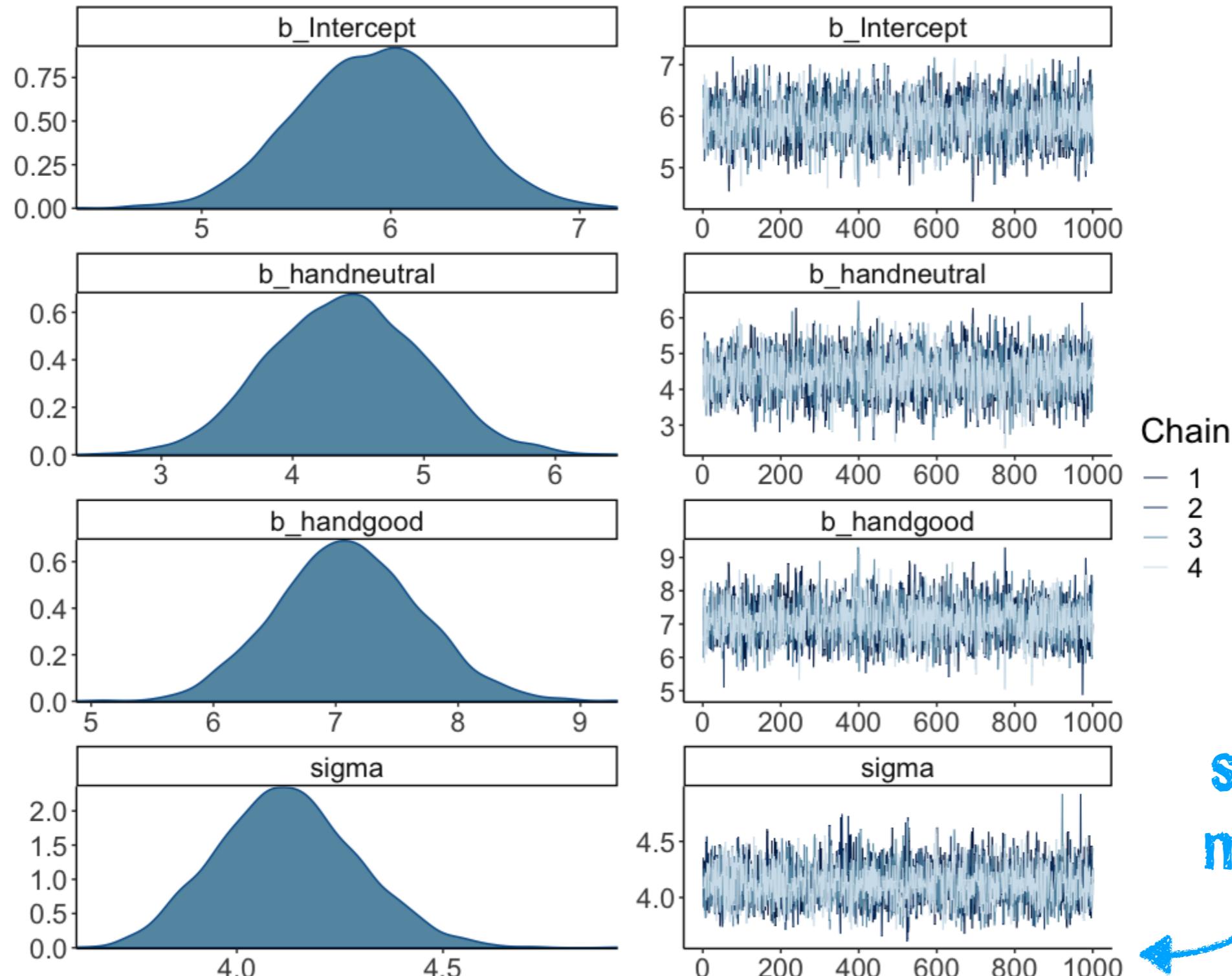


Model evaluation

1. Check whether inference worked

Can we trust the model results?

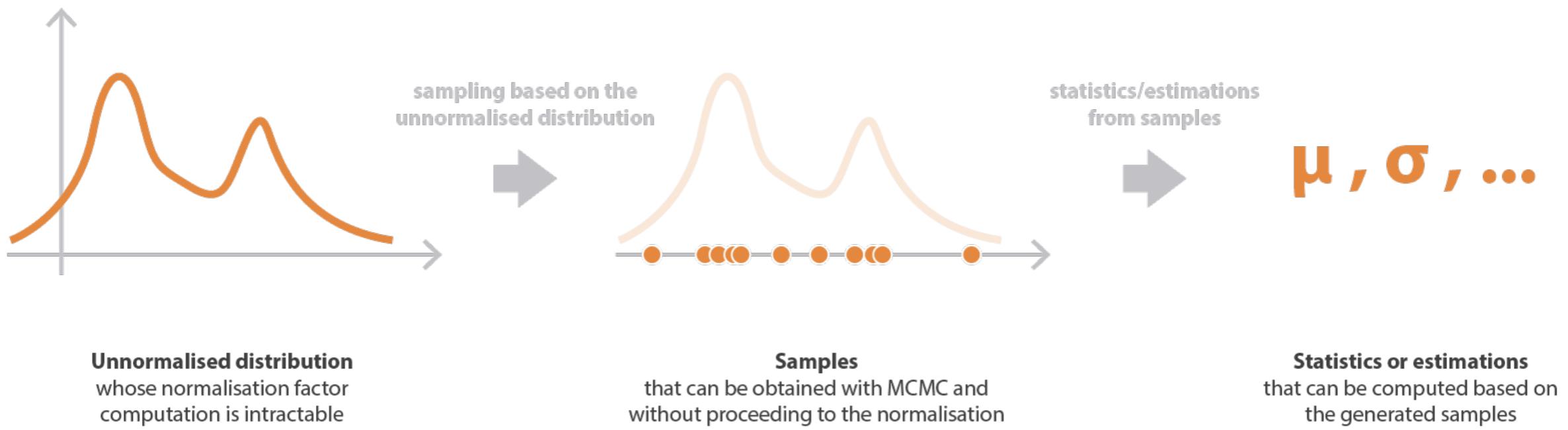
`plot(fit.brm_poker)`



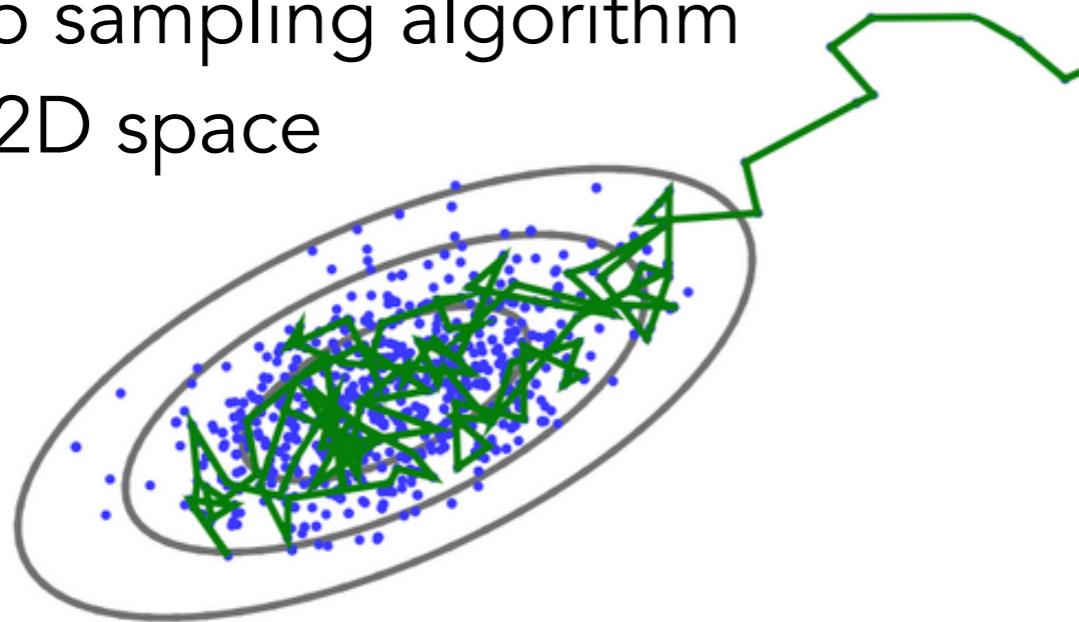
sample
number

Can we trust the model results?

Inference via Markov Chain Monte Carlo (MCMC)



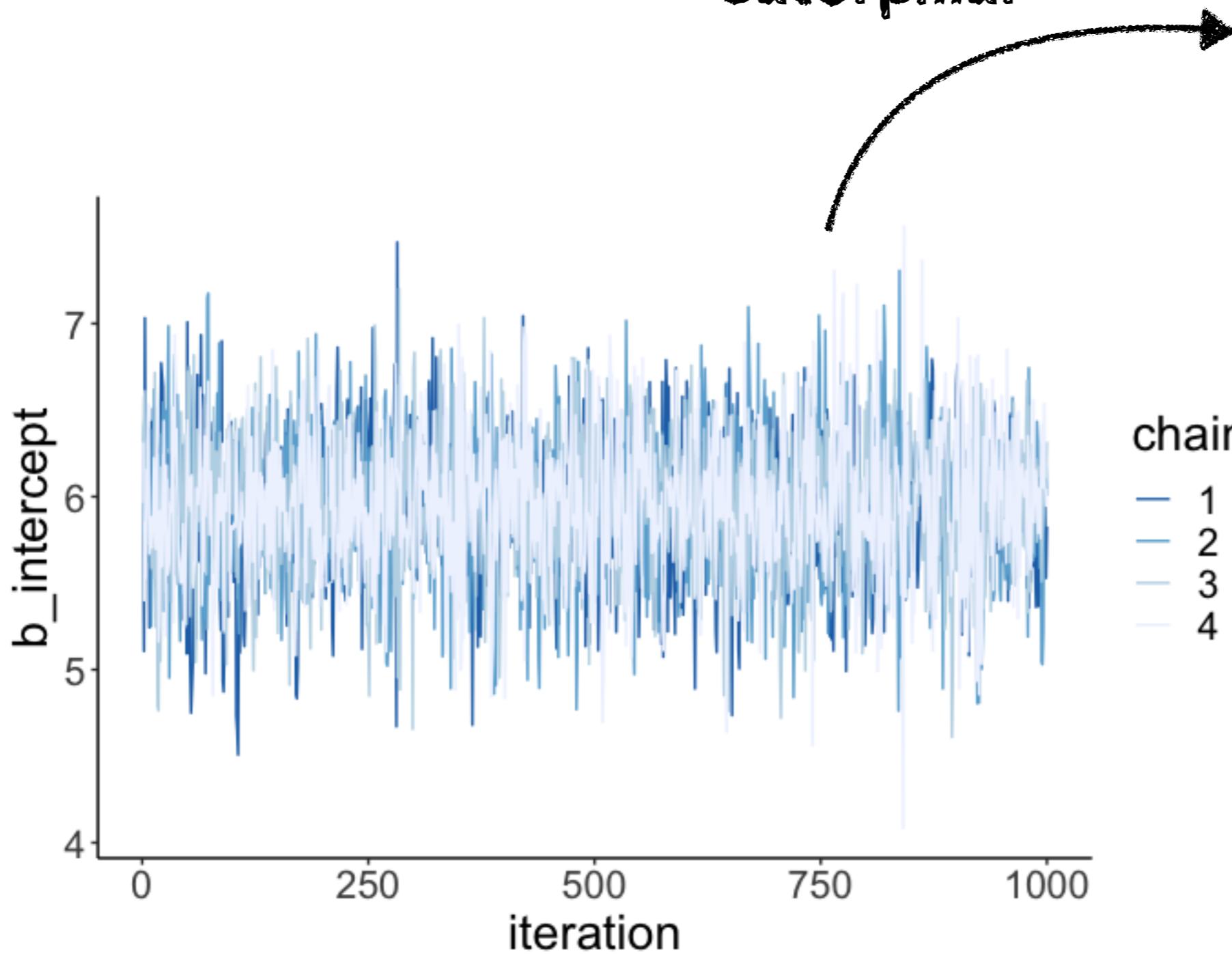
Markov Chain Monte
Carlo sampling algorithm
in a 2D space



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

Can we trust the model results?

looks like a fuzzy caterpillar



Stats twitter chimes in ...



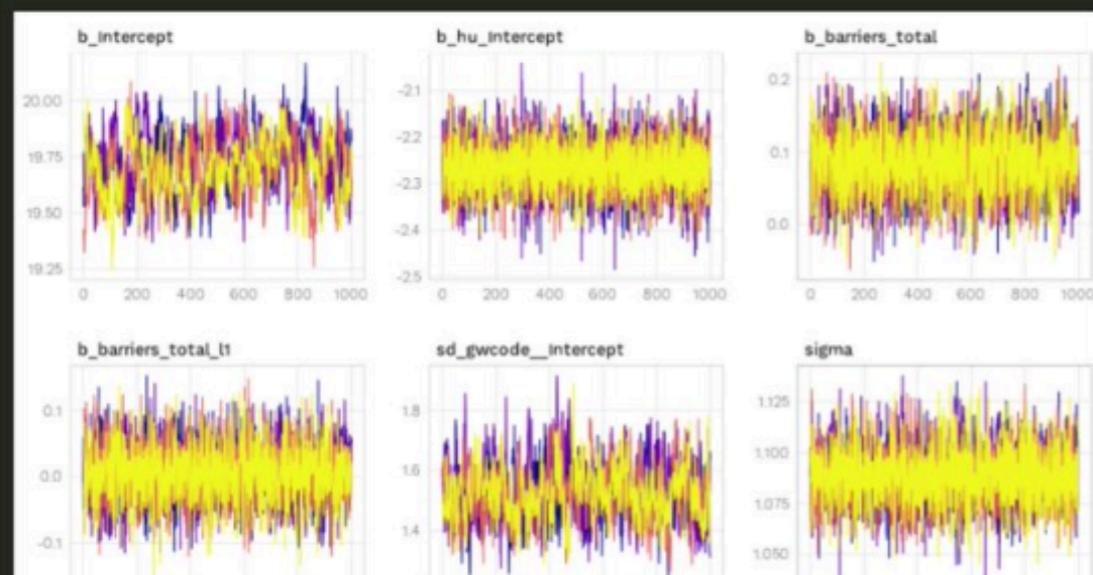
Andrew Heiss
@andrewheiss

...

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()
```
...
```



...

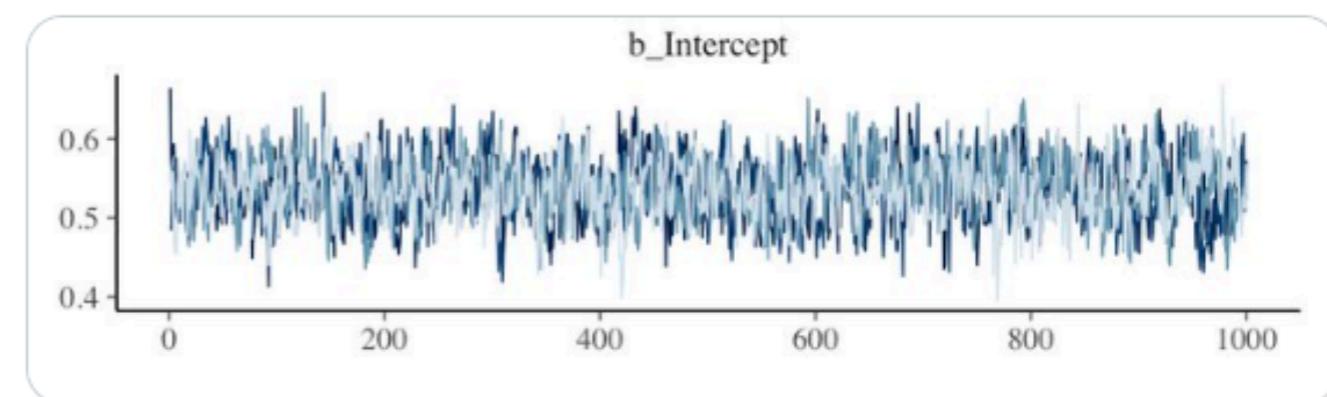


Chelsea Parlett-Pelleriti
@ChelseaParlett

...

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.brm_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

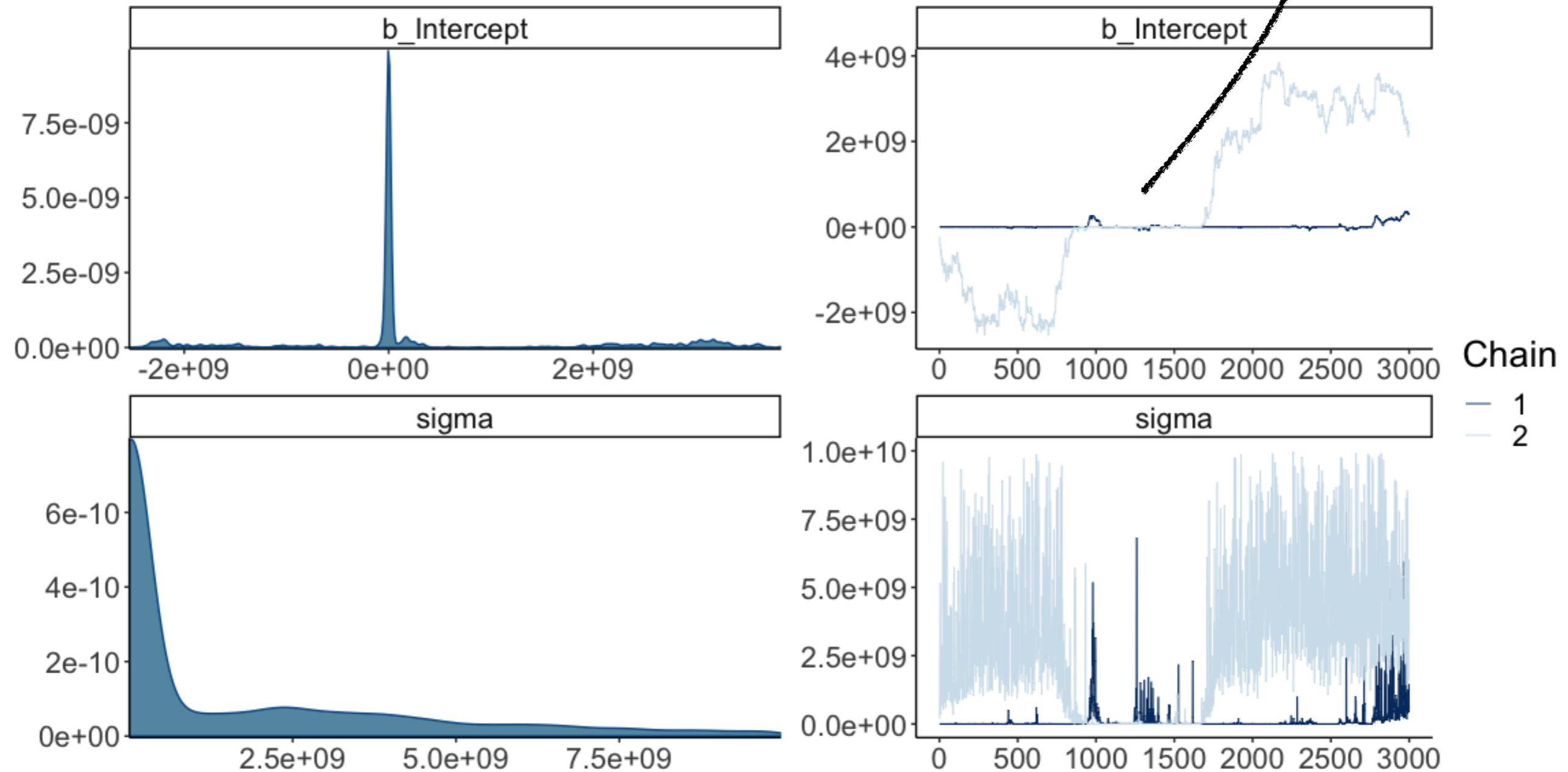
Population-Level Effects:
  Estimate   Est.Error    1-95% CI    u-95% CI Rhat Bulk_ESS Tail_ESS
Intercept 357550121.58 1416057299.71 -2244033111.47 3333594132.43 1.78      3      24

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:

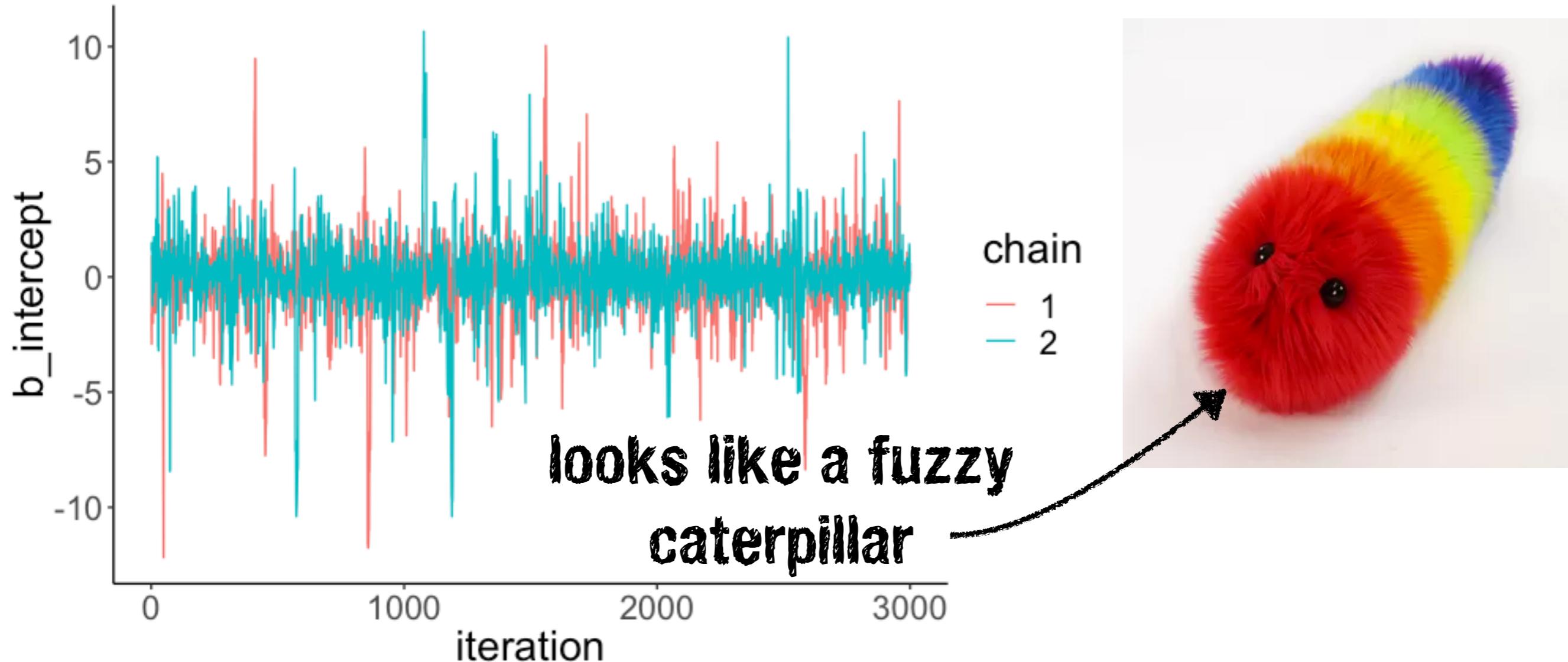
| | Estimate | Est.Error | l-95% | CI | u-95% | CI | Eff.Sample | Rhat |
|-----------|----------|-----------|-------|------|-------|----|------------|------|
| Intercept | -0.06 | 1.72 | -3.78 | 3.27 | | | 1033 | 1.00 |

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



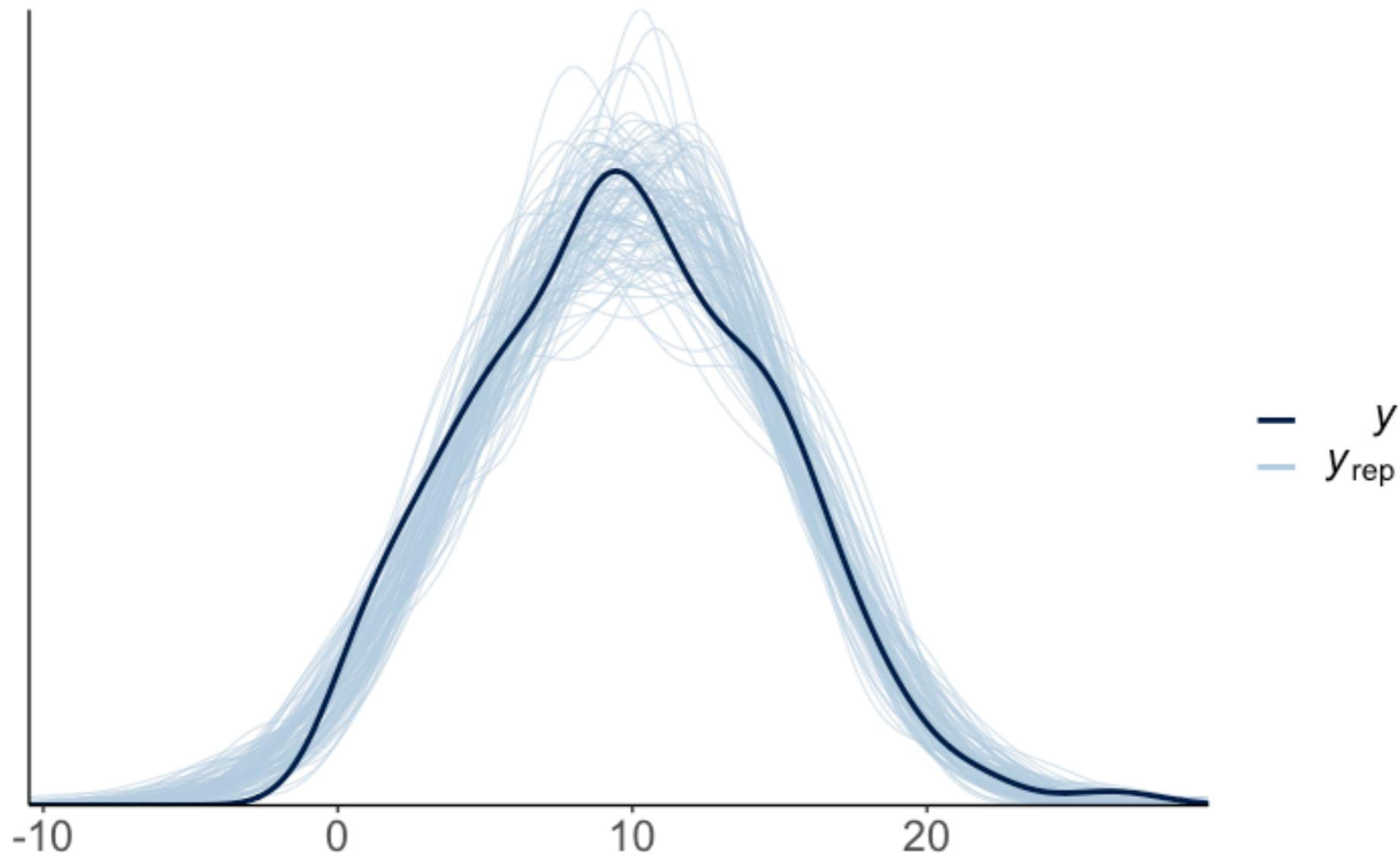
if things go wrong:

- 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

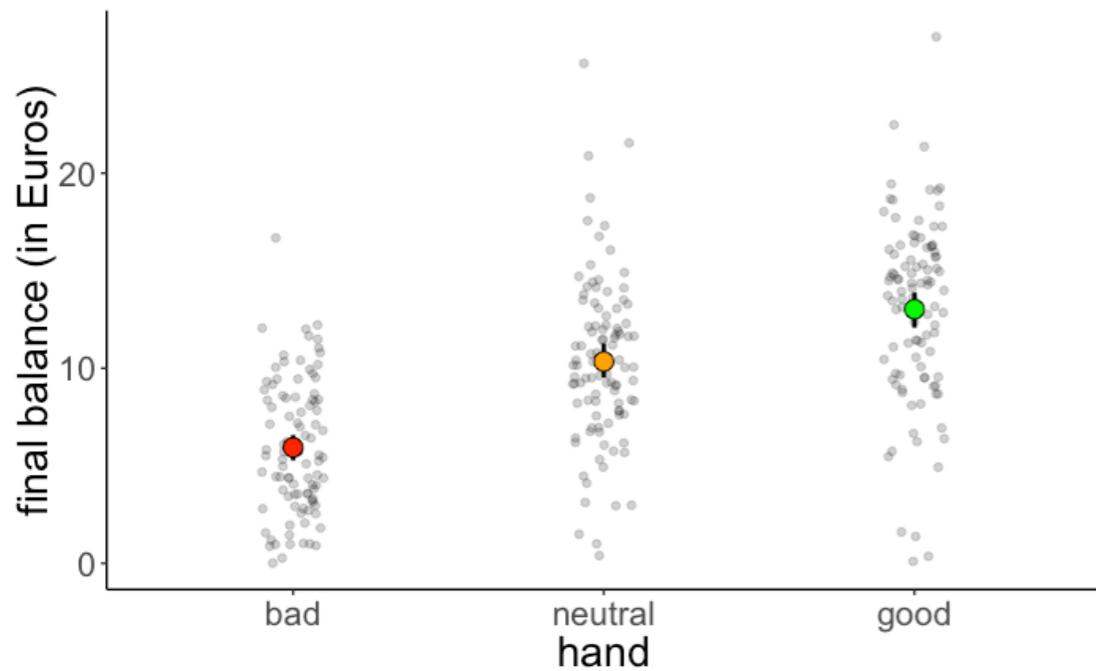
```
pp_check(fit.brm, nsamples = 100)
```



The model accurately captures the distribution of the response variable

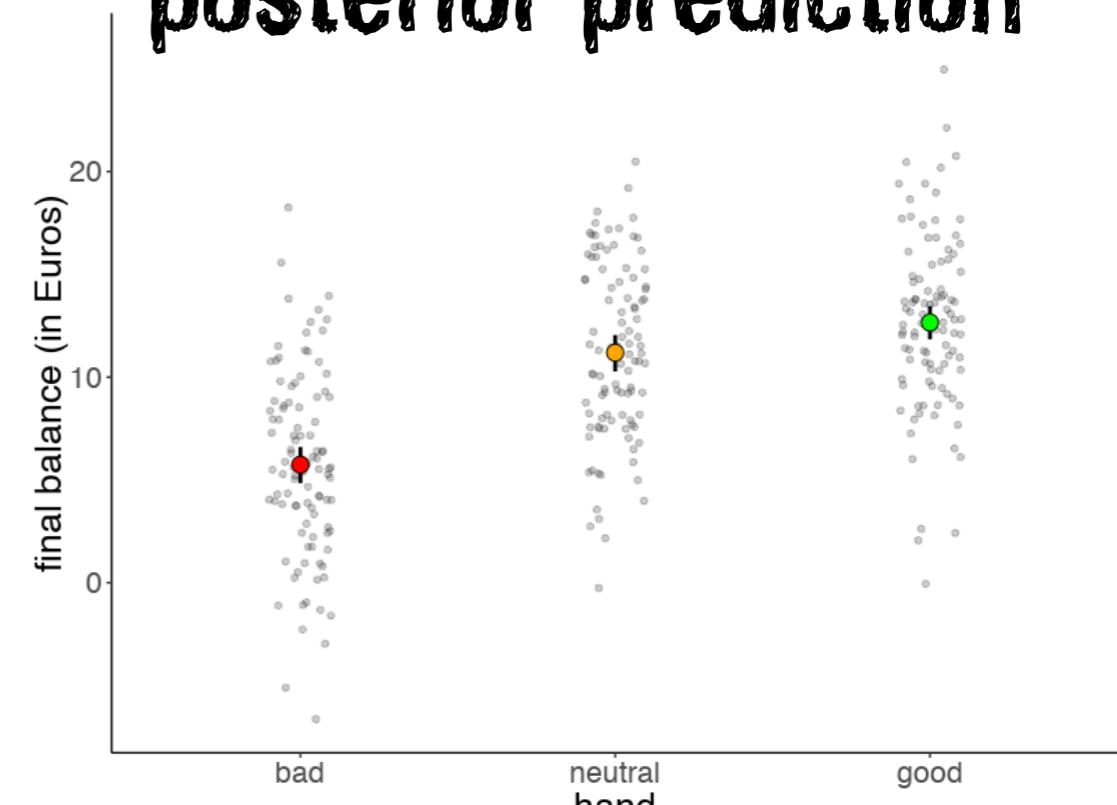
Posterior predictive check

original data

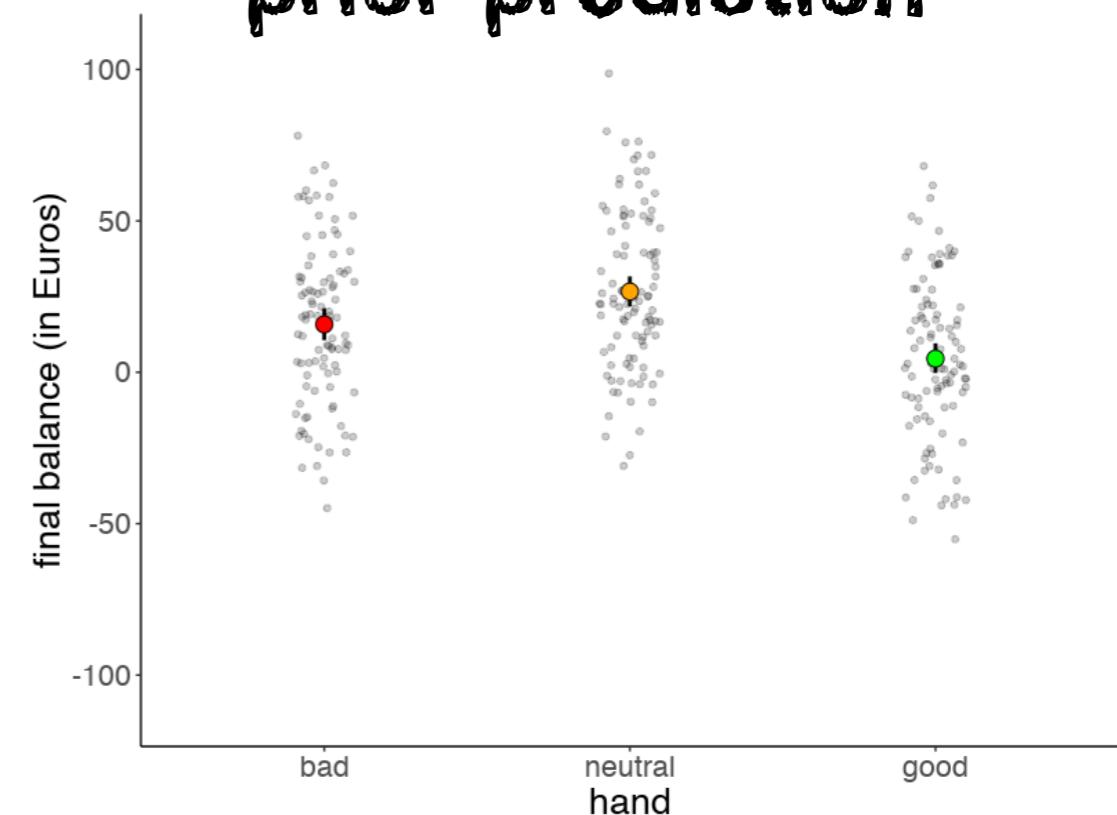


take a look at course notes
for how to make these

posterior prediction



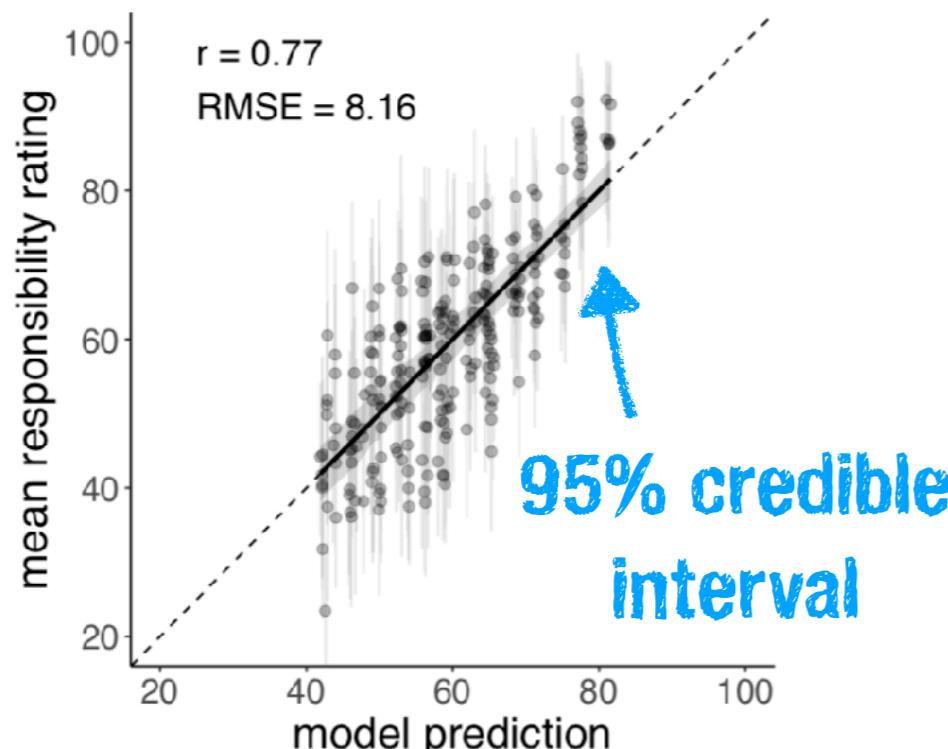
prior prediction



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

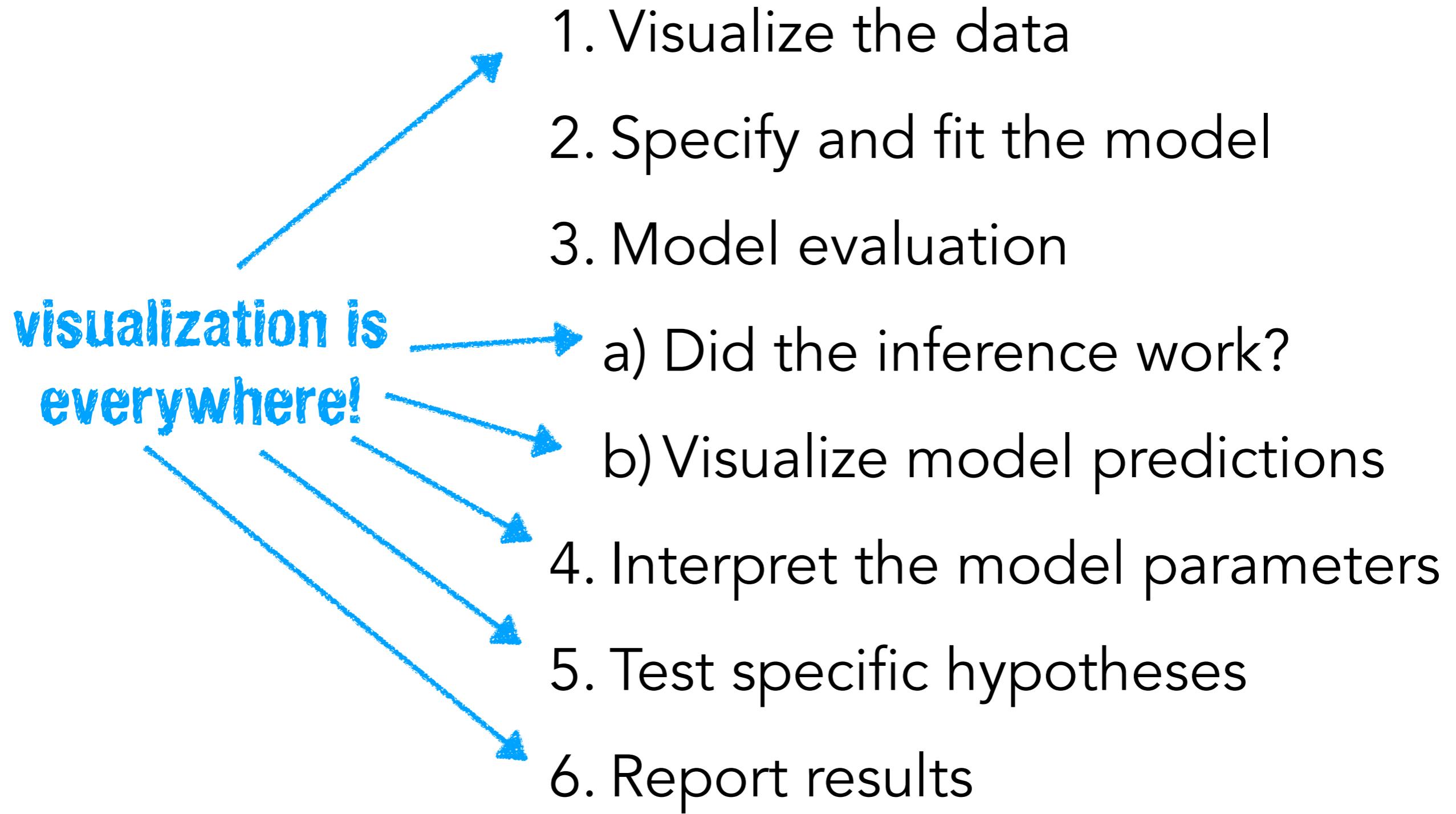
parameter
estimates

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 0.¹

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).

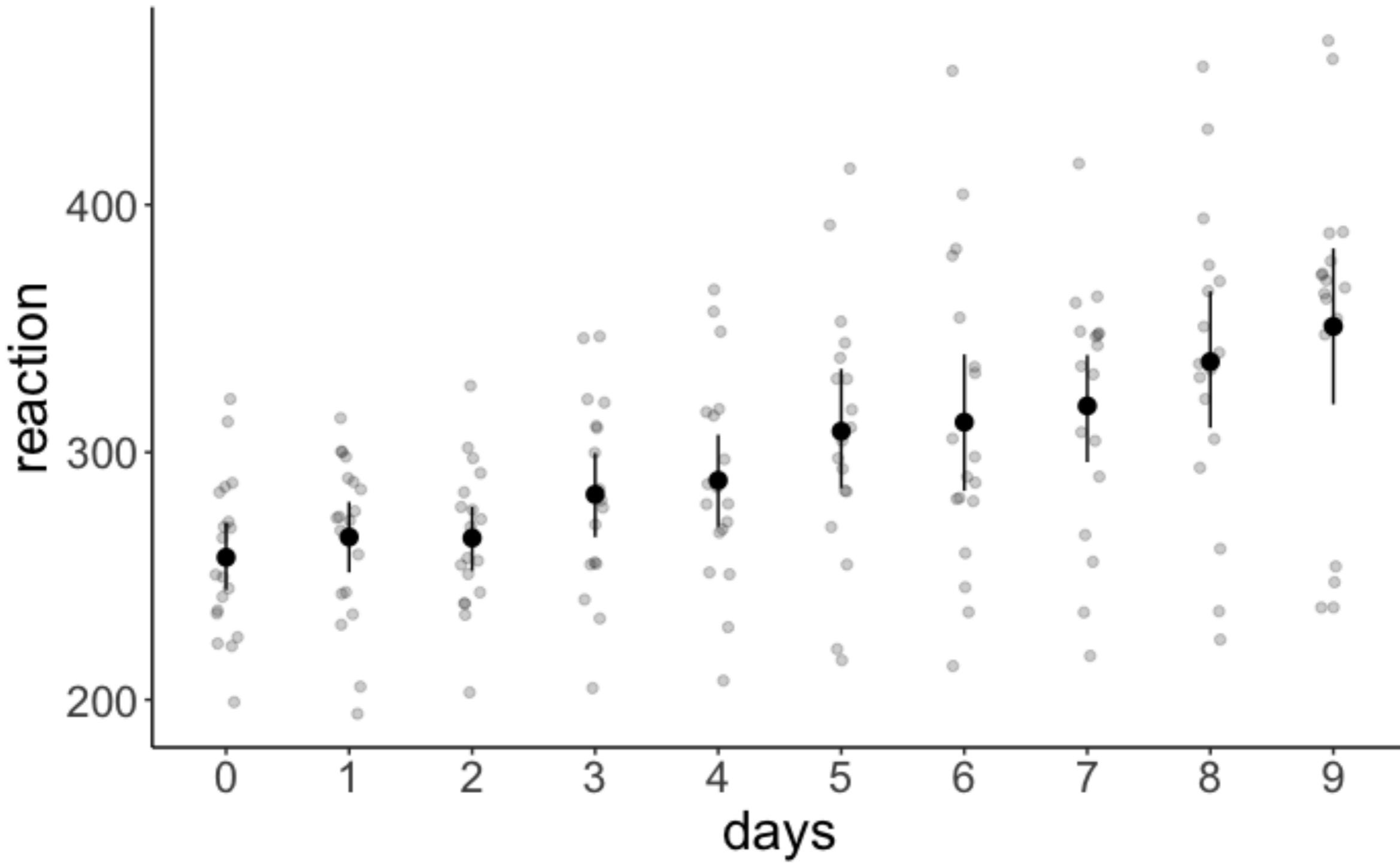
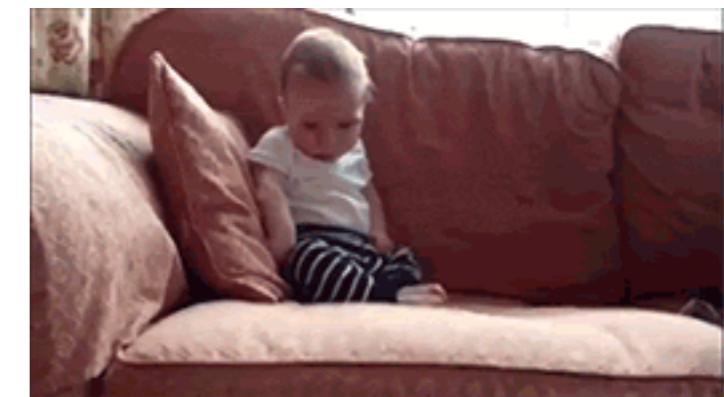
Recipe for Bayesian analysis with brms



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

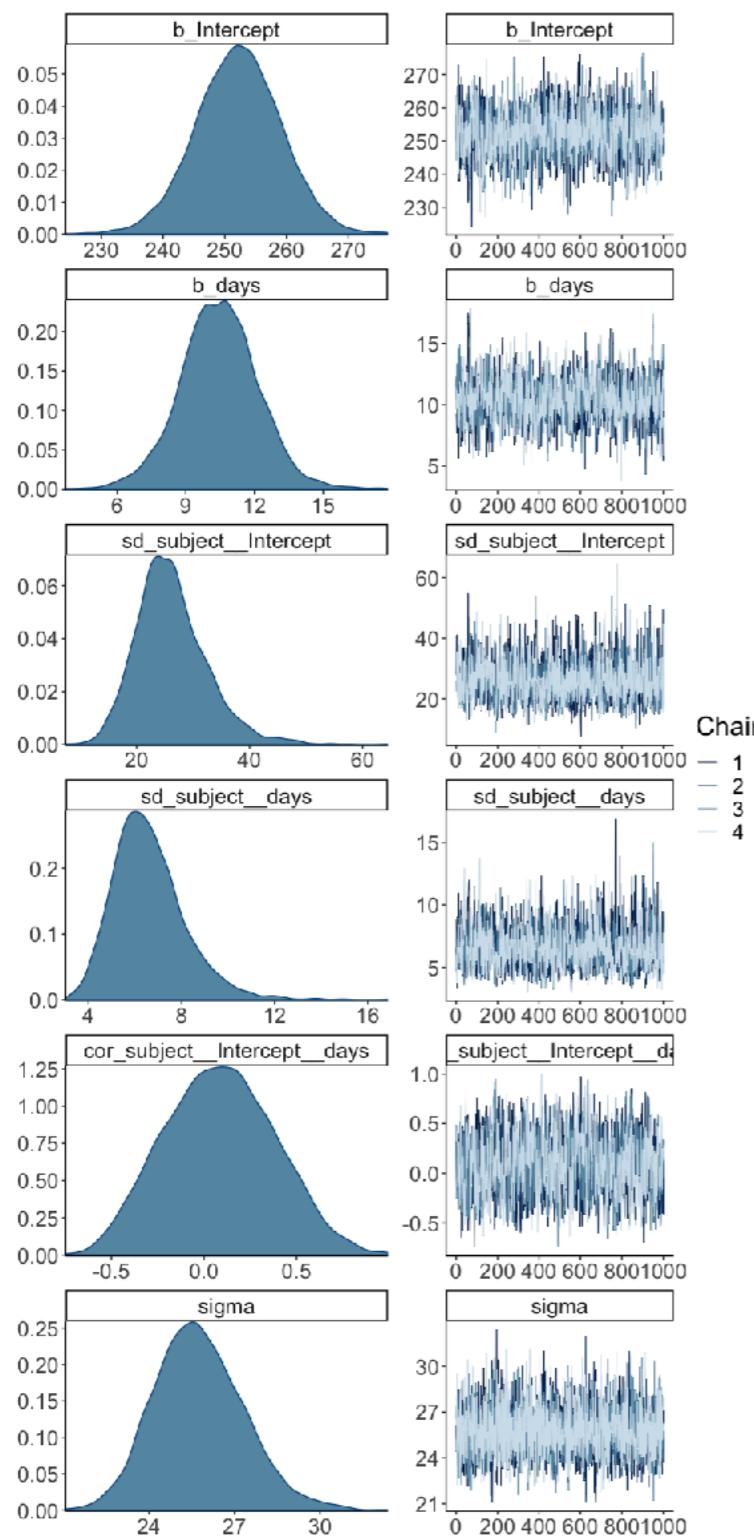
1.00 is
good!

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)
```

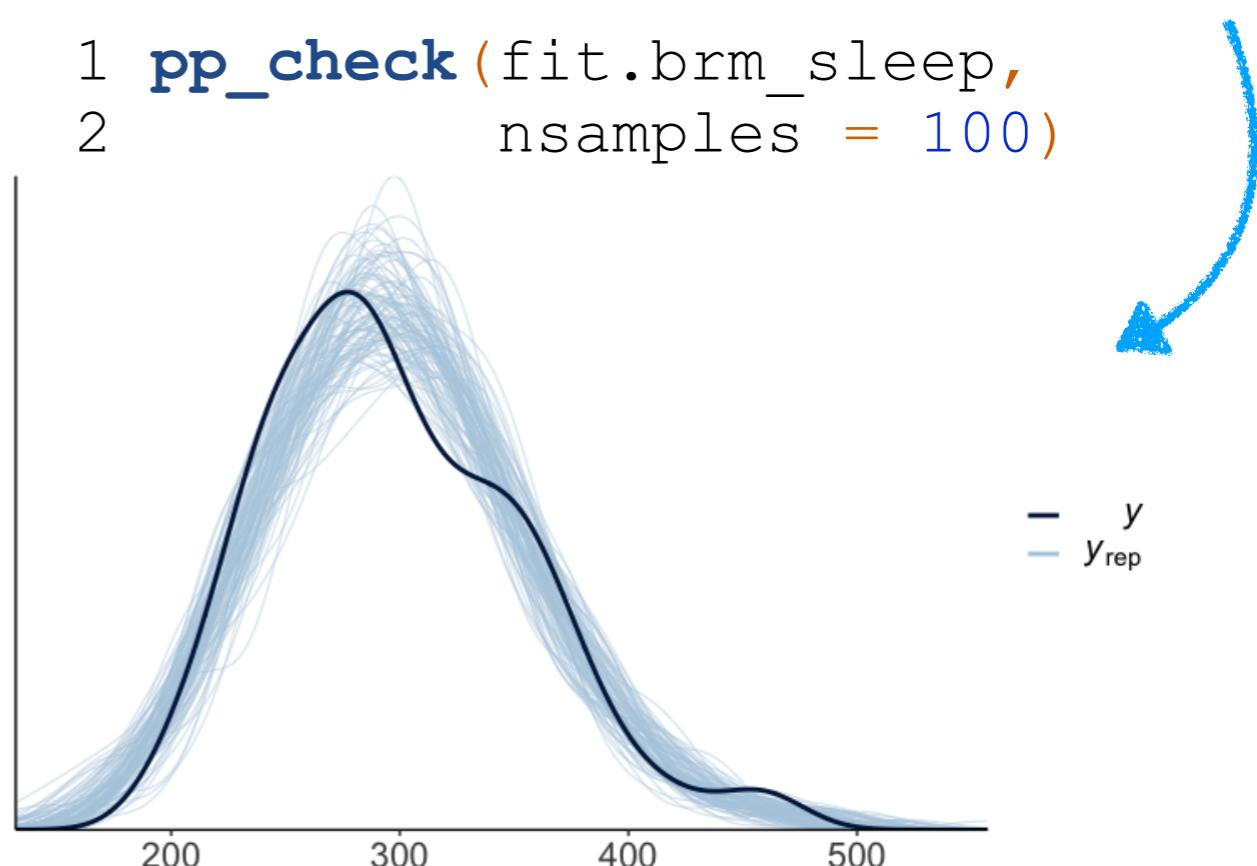


these look good!

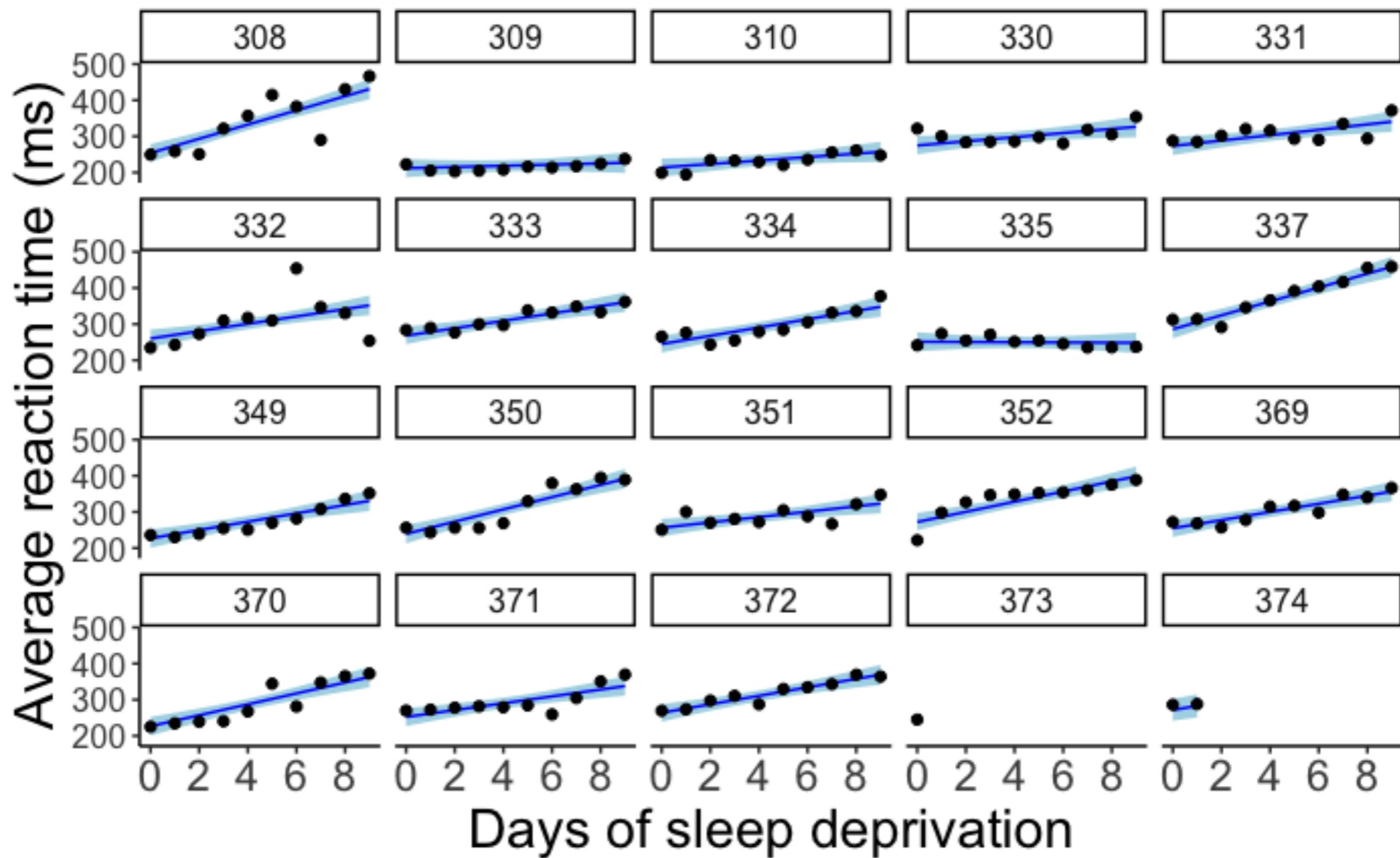


also looks good!

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

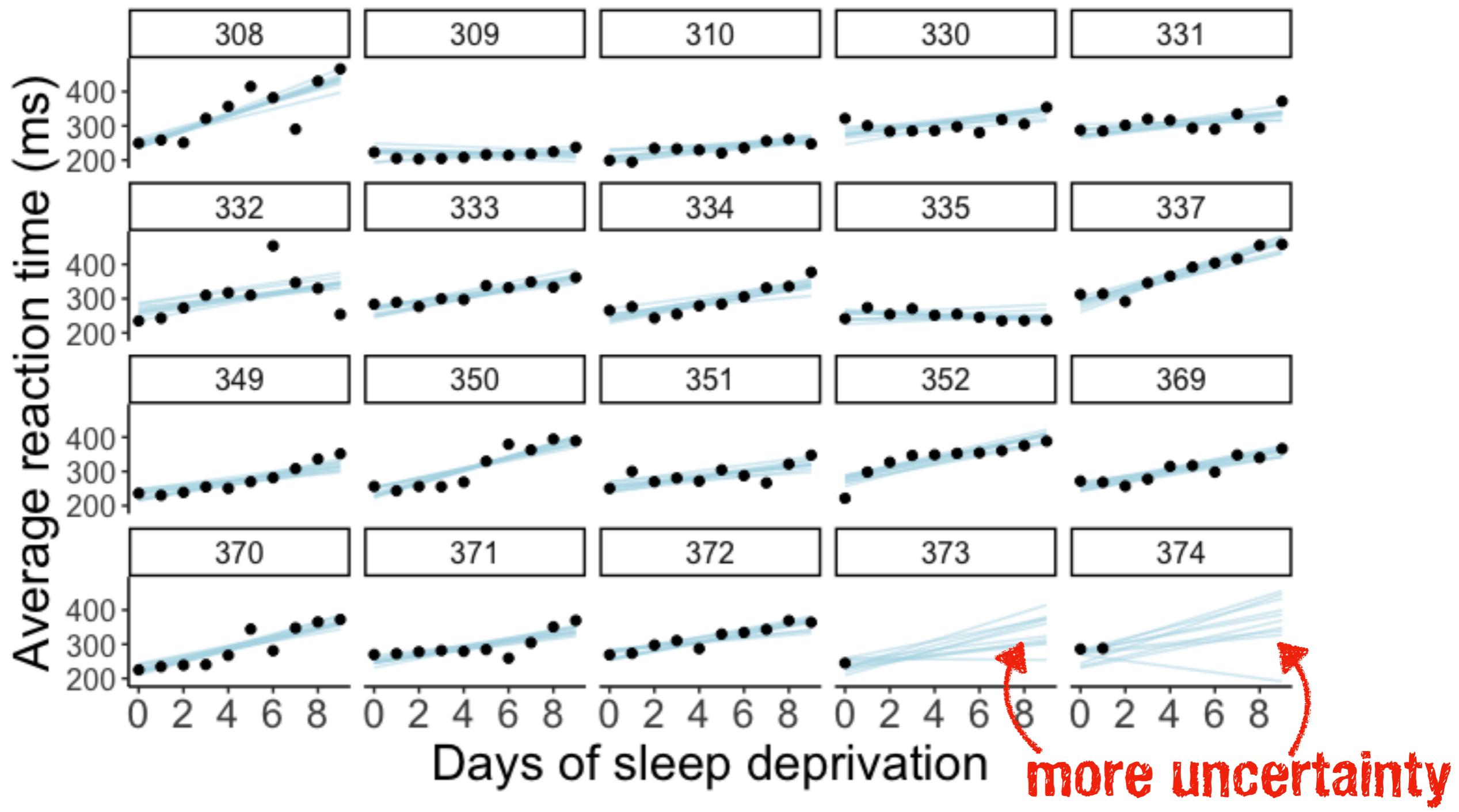


b) Visualize the model predictions



regression lines with 95% highest density intervals

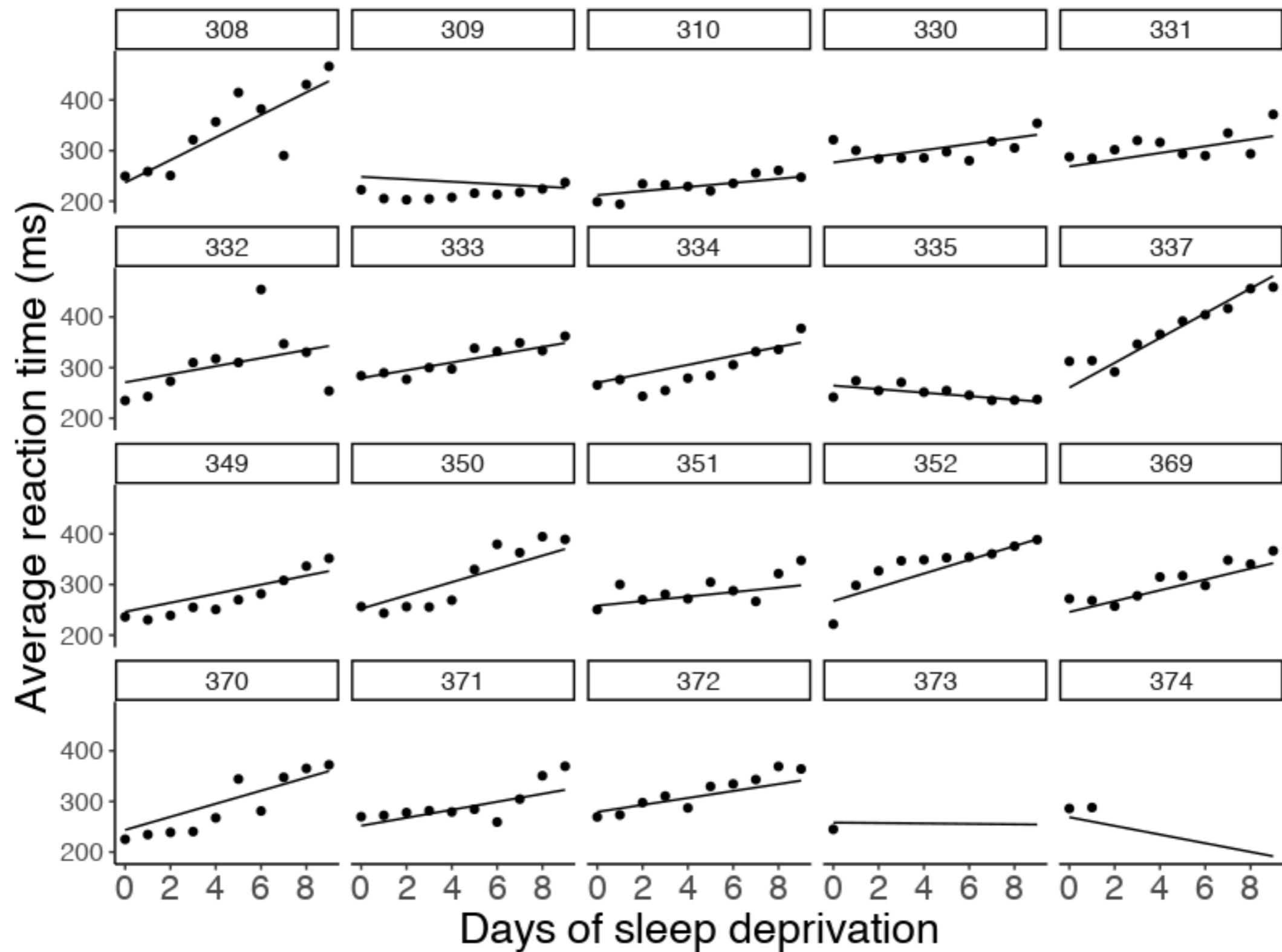
b) Visualize the model predictions



10 random samples from the posterior distribution

b) Visualize the model predictions

if you're feeling fancy



4. Interpret the model parameters

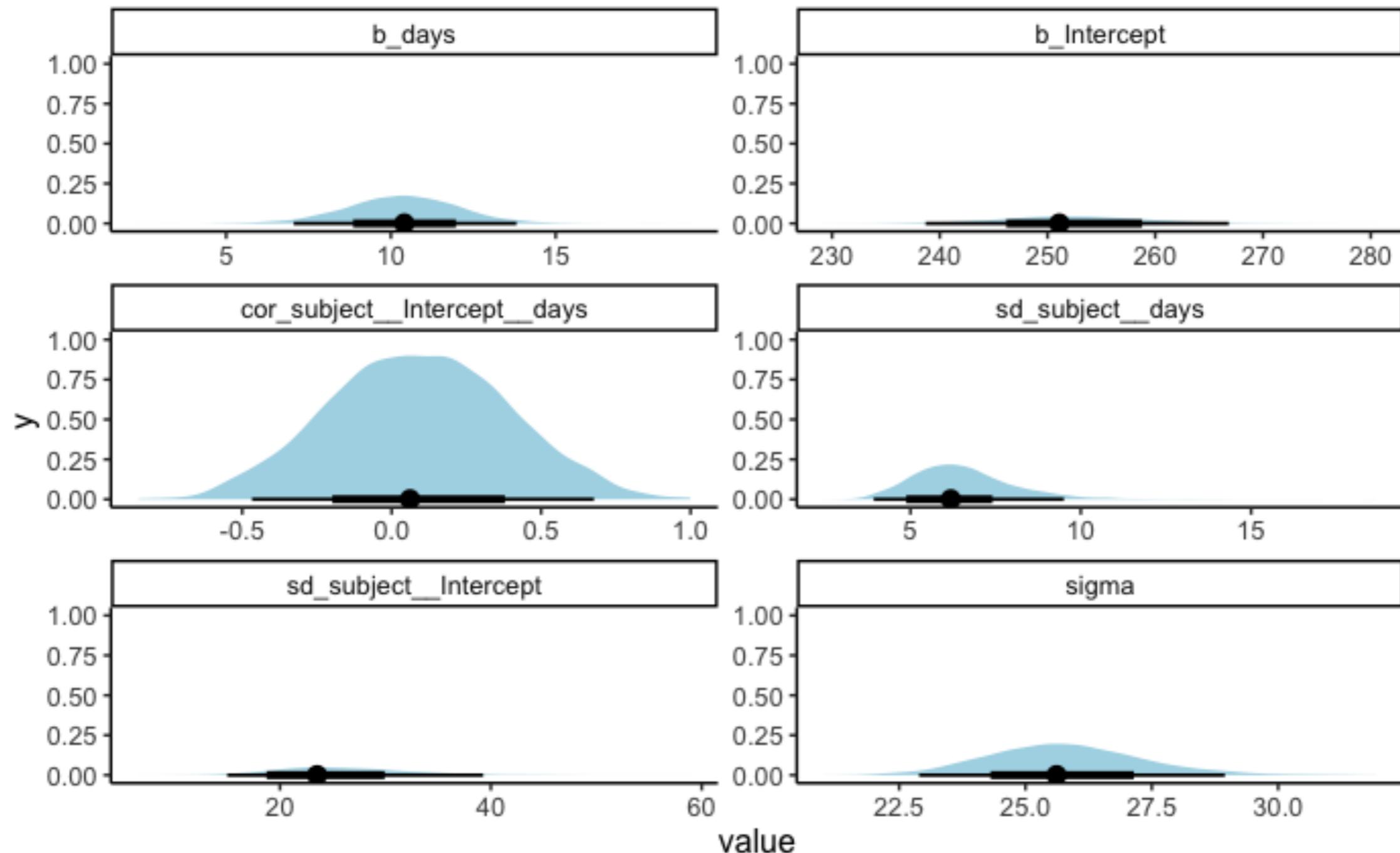
4. Interpret the model parameters

```
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 |

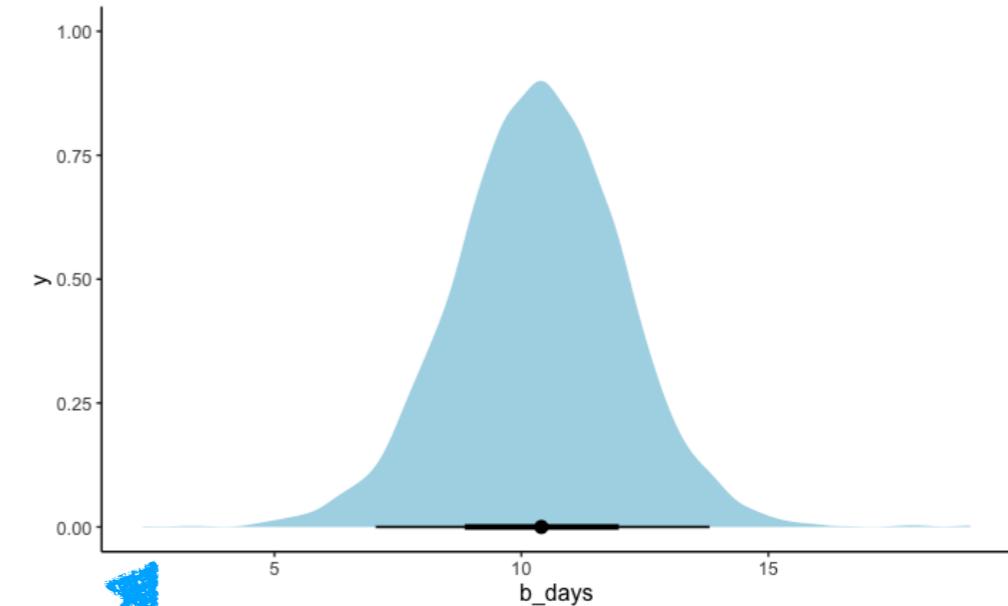
Population-Level Effects:

| | 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 |

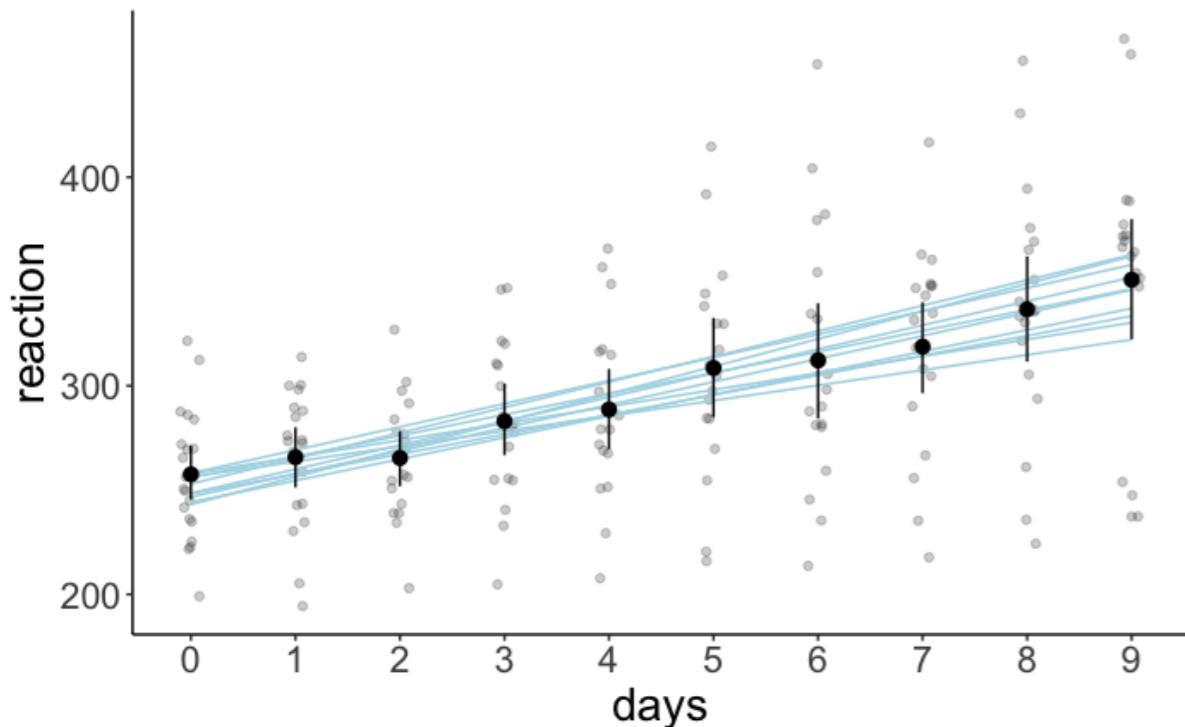
Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample is a crude measure of effective sample size, and Rhat is the potential scale reduction factor on split chains (at convergence, Rhat = 1).



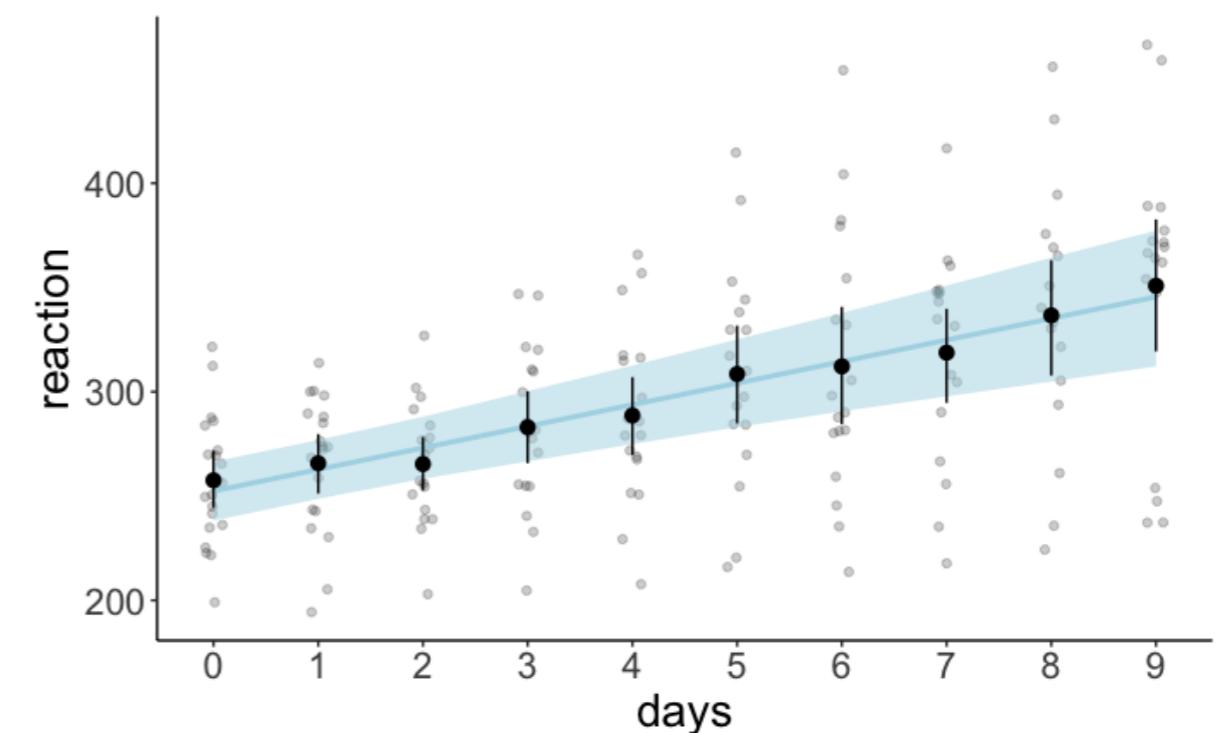
6. Report results

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).

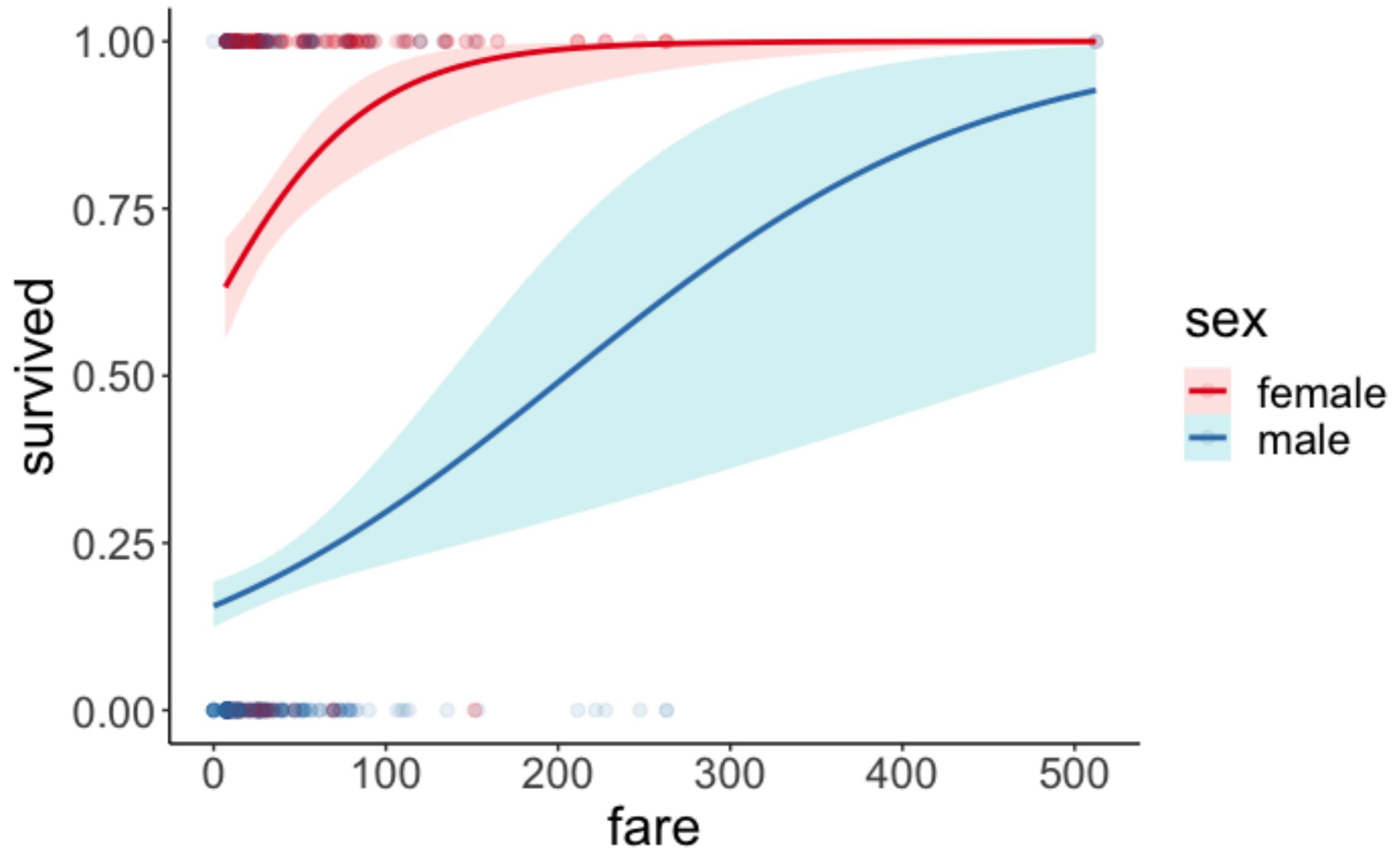
Recipe for Bayesian analysis with brms

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

Titanic data

1. Visualize the data

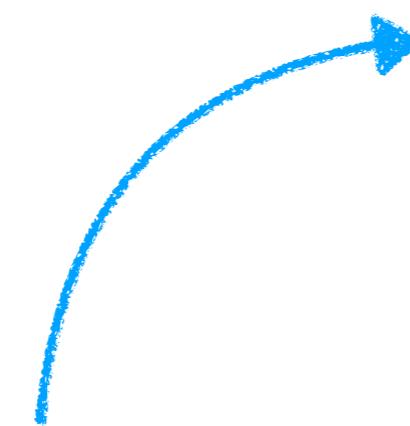
Feeling cold?



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:

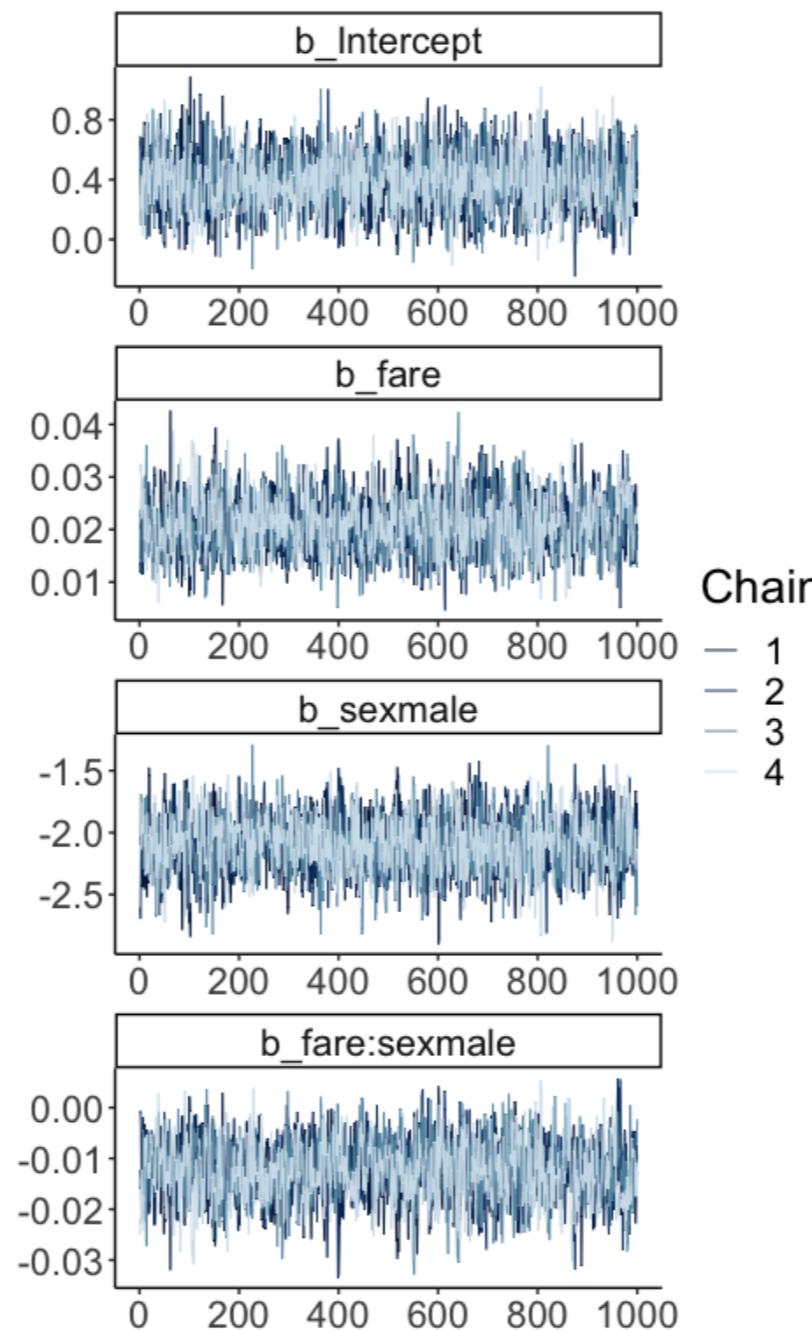
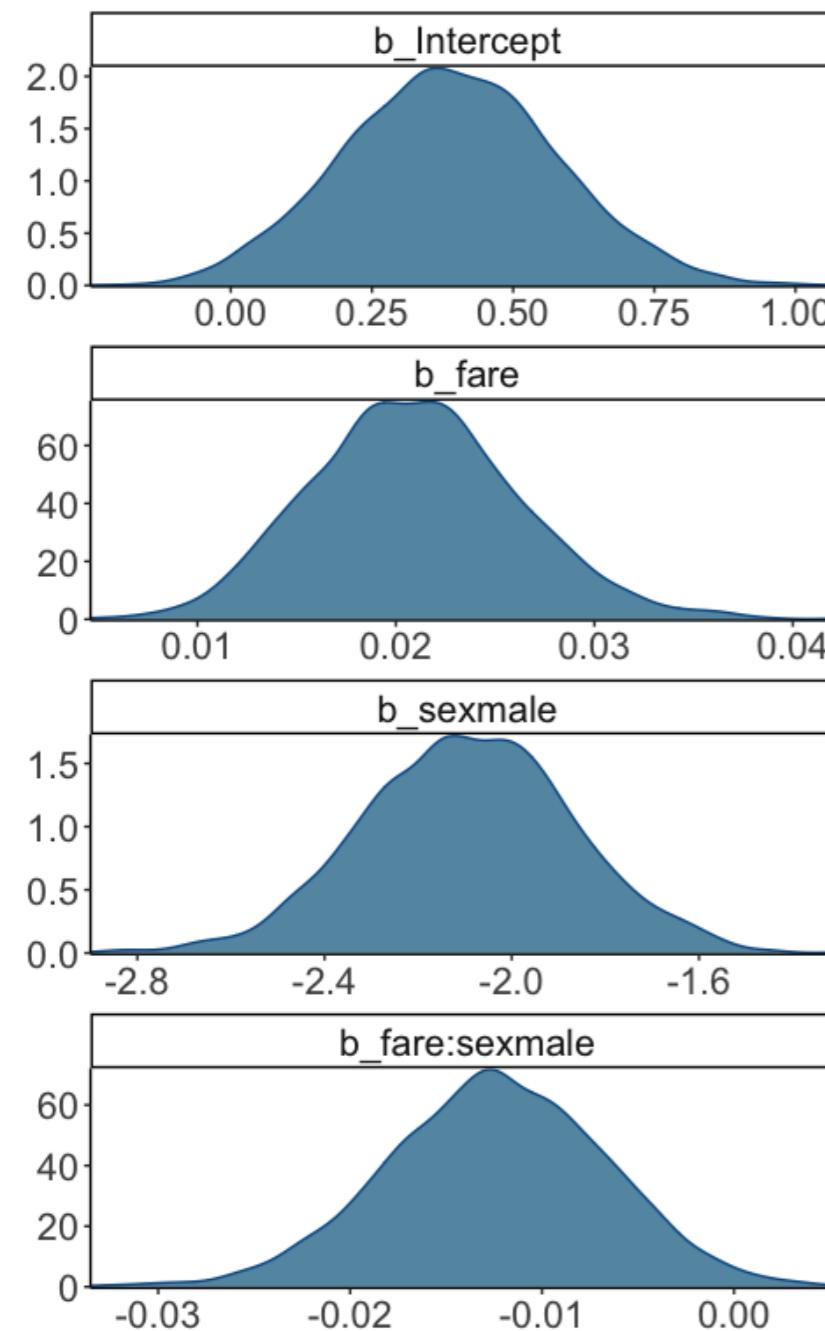
| | Estimate | Est.Error | l-95% CI | u-95% CI | Rhat | Bulk_ESS | Tail_ESS |
|--------------|----------|-----------|----------|----------|------|----------|----------|
| Intercept | 0.39 | 0.19 | 0.03 | 0.76 | 1.00 | 2010 | 2625 |
| fare | 0.02 | 0.01 | 0.01 | 0.03 | 1.00 | 1545 | 2124 |
| sexmale | -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).

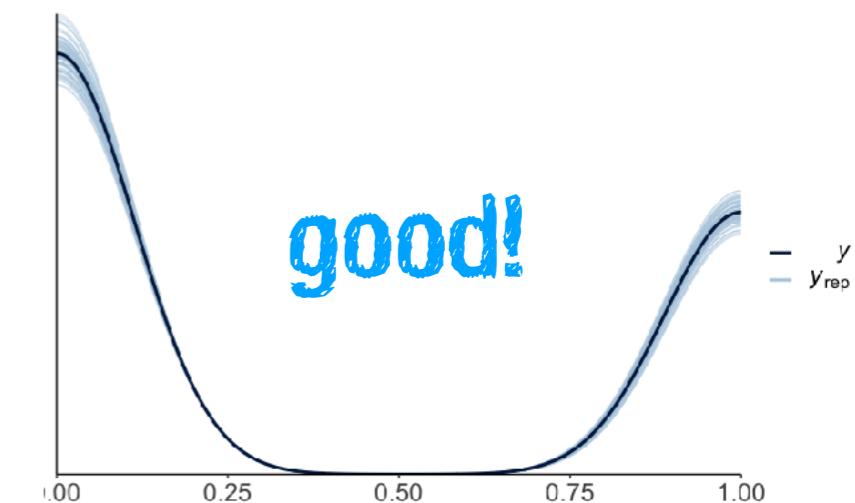
looks good

a) Did the inference work?

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

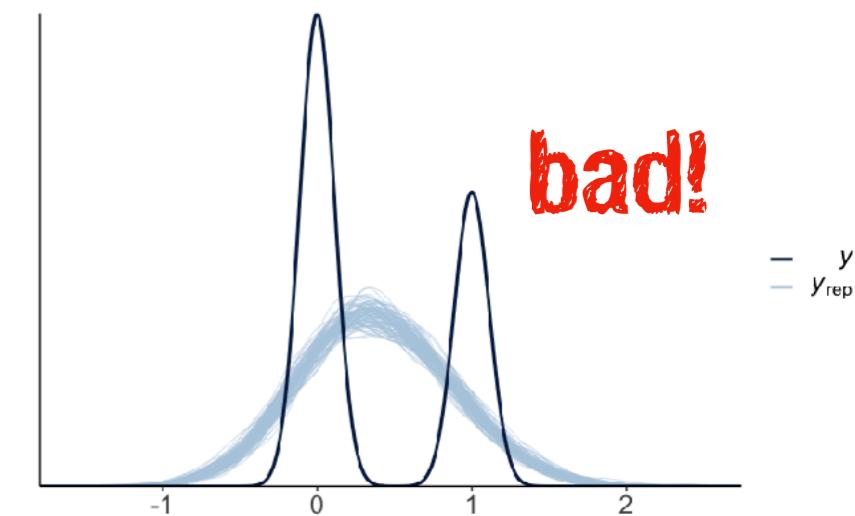


```
1 pp_check(fit.brm_titanic,  
2           nsamples = 100)
```

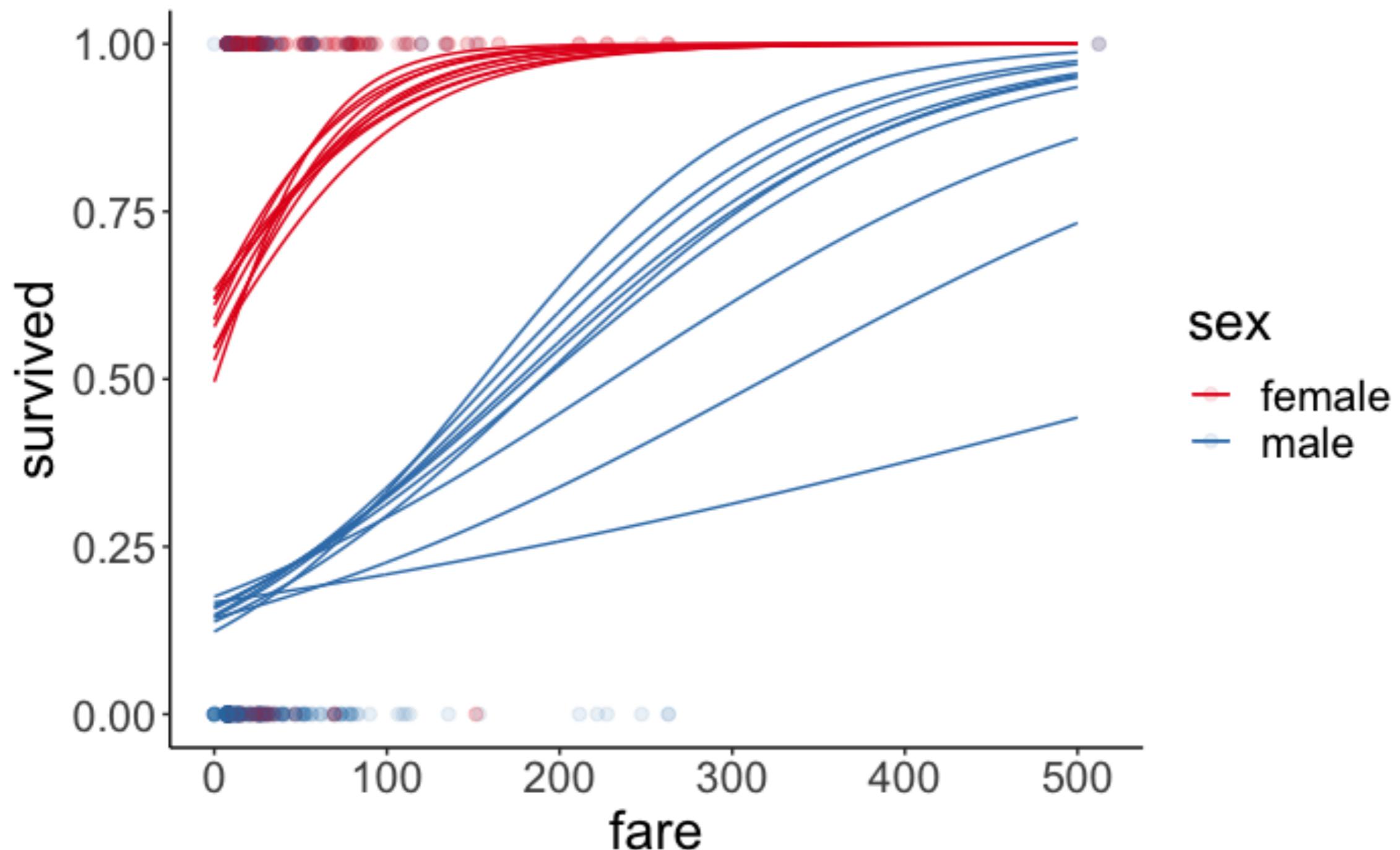


Chain
— 1
— 2
— 3
— 4

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).

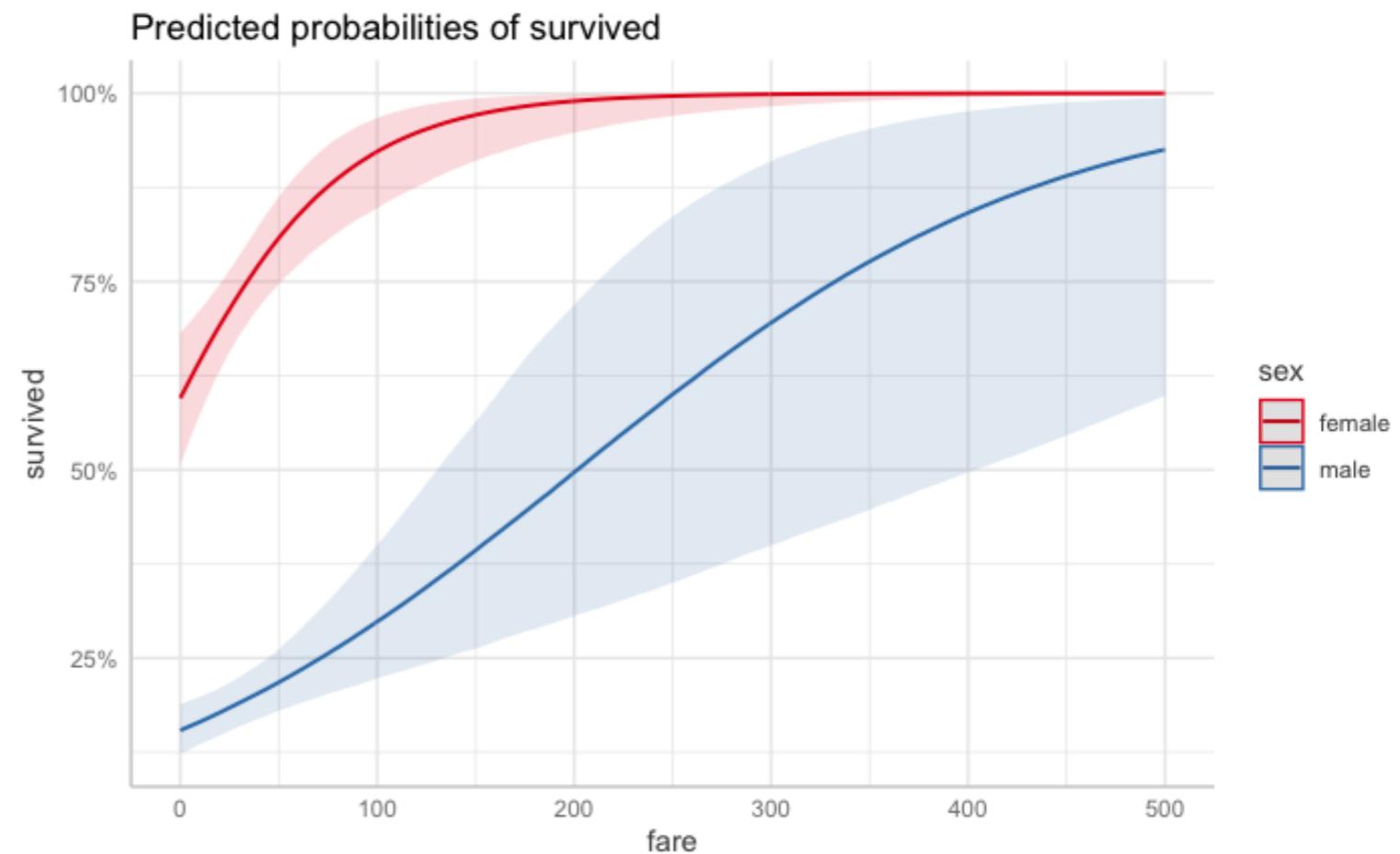
log odds

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 | | | | | | | | | | | | | | | | | | | | | |
| <table><thead><tr><th>x</th><th>Predicted</th><th>95% CI</th></tr></thead><tbody><tr><td>0</td><td>0.60</td><td>[0.51, 0.68]</td></tr><tr><td>83</td><td>0.89</td><td>[0.82, 0.95]</td></tr><tr><td>167</td><td>0.98</td><td>[0.93, 1.00]</td></tr><tr><td>250</td><td>1.00</td><td>[0.97, 1.00]</td></tr><tr><td>333</td><td>1.00</td><td>[0.99, 1.00]</td></tr><tr><td>500</td><td>1.00</td><td>[1.00, 1.00]</td></tr></tbody></table> | 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] |
| 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 | | | | | | | | | | | | | | | | | | | | | |
| <table><thead><tr><th>x</th><th>Predicted</th><th>95% CI</th></tr></thead><tbody><tr><td>0</td><td>0.15</td><td>[0.12, 0.19]</td></tr><tr><td>83</td><td>0.27</td><td>[0.21, 0.35]</td></tr><tr><td>167</td><td>0.43</td><td>[0.28, 0.62]</td></tr><tr><td>250</td><td>0.60</td><td>[0.35, 0.84]</td></tr><tr><td>333</td><td>0.75</td><td>[0.43, 0.94]</td></tr><tr><td>500</td><td>0.93</td><td>[0.60, 0.99]</td></tr></tbody></table> | 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] |
| 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

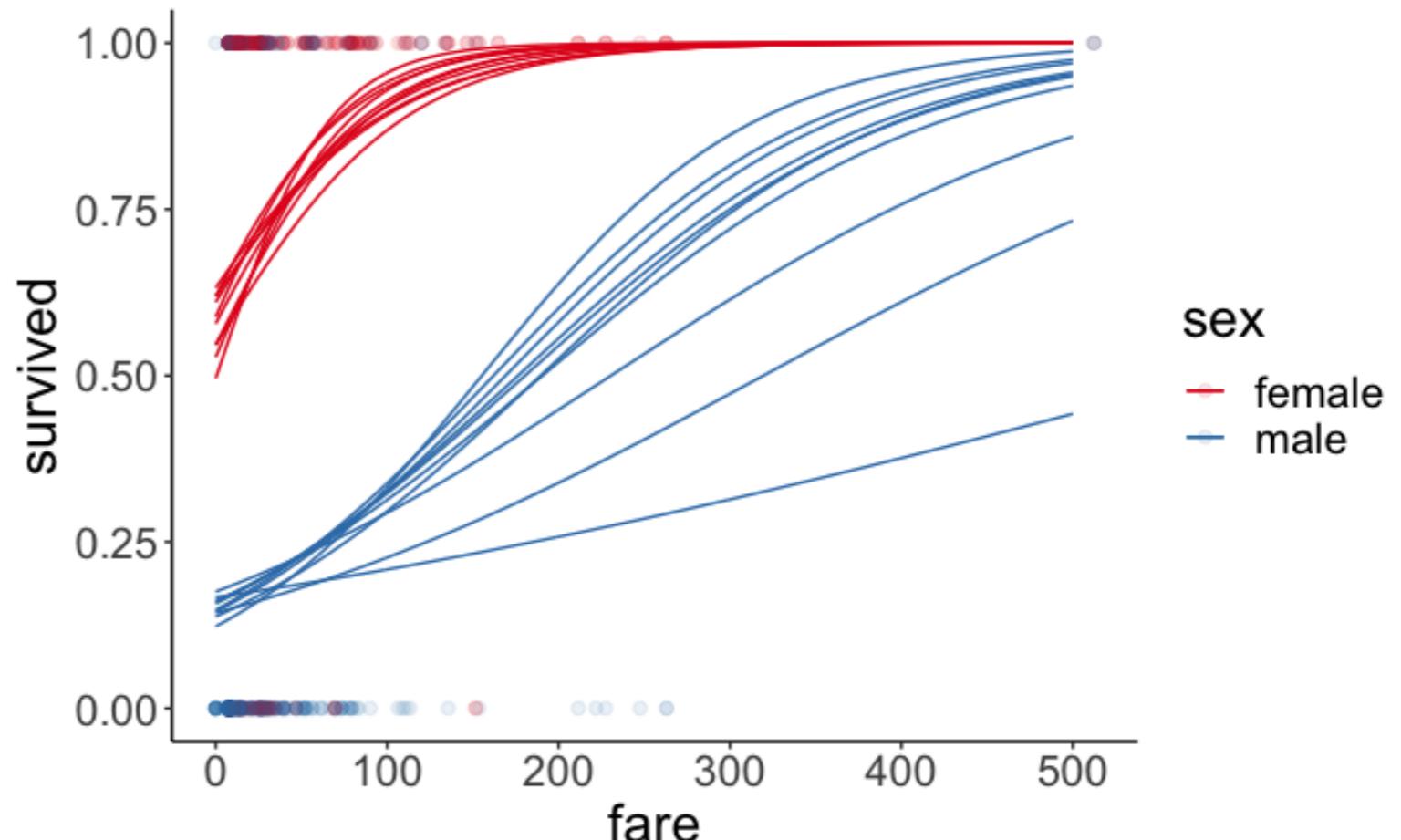
```
$contrasts
  contrast      estimate lower.HPD upper.HPD
  female - male 0.0124   0.000884  0.0232
```

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 [8.39, 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 0.0124 [0.000884, 0.0232].

Summary

- Quick recap
- Doing Bayesian data analysis with `greta`
- Doing Bayesian data analysis **with BRMS**
 - Recipe for Bayesian data analysis
 - Poker data
 - Sleep data
 - Titanic data

Feedback

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

Thank you!