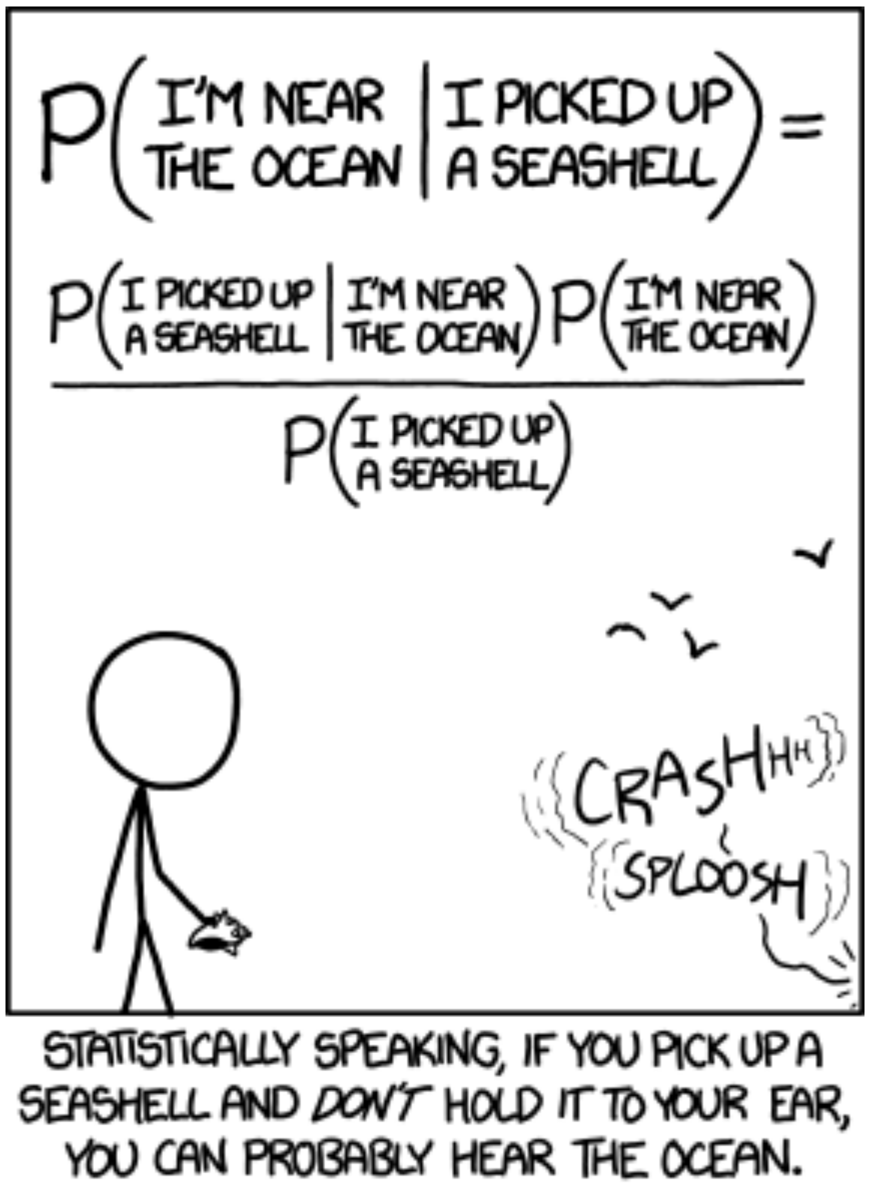
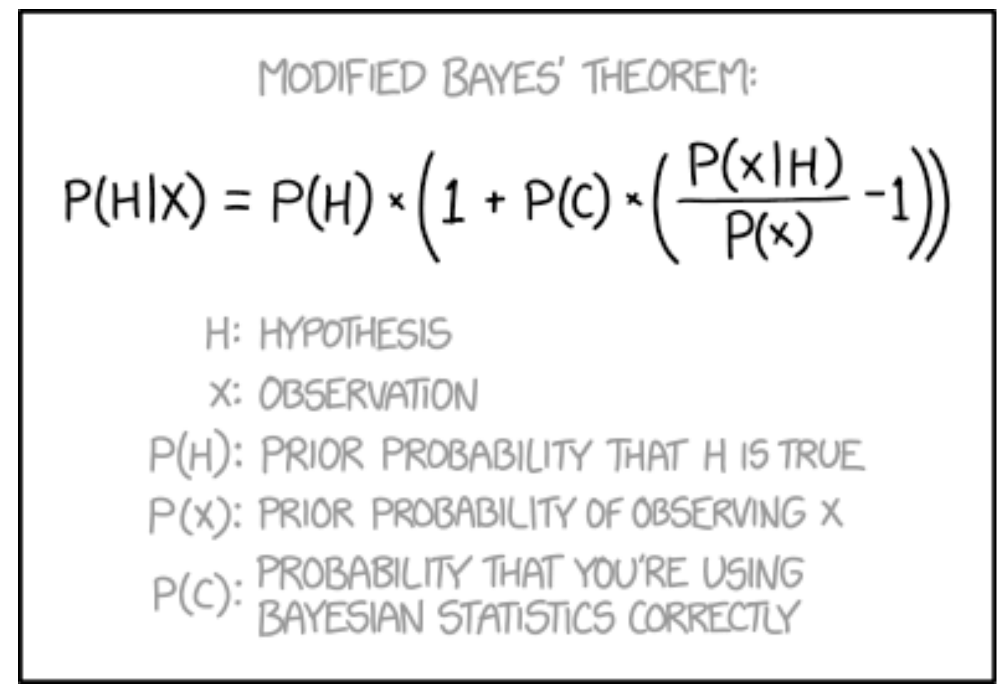
Bayesian data analysis 3





03/08/2019

Logistics

Your feedback

Your feedback

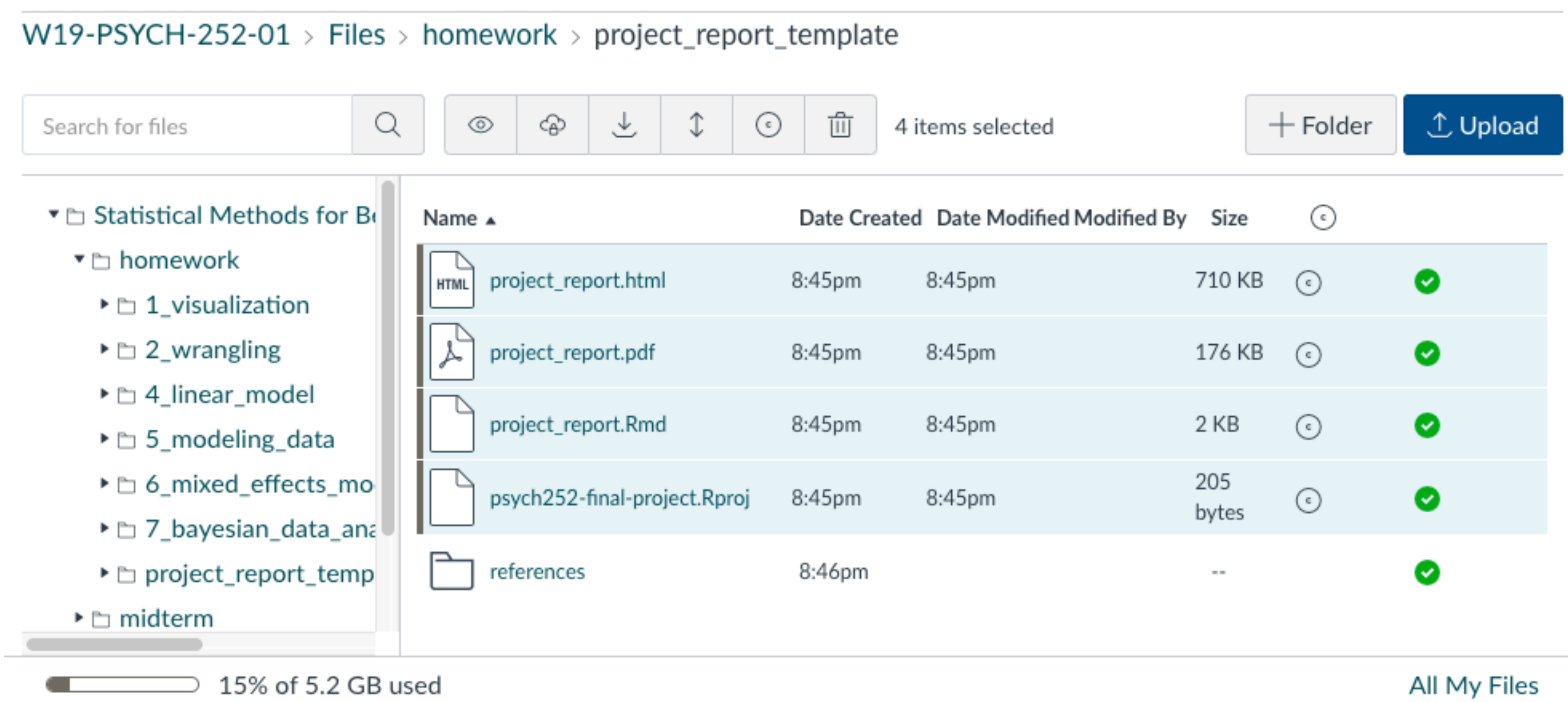
Little hard to follow today. Having one example (possibly more interesting than coin flips?) to follow throughout the lecture would help scaffold the flow. But I'm still excited about Bayes!

Things that came up

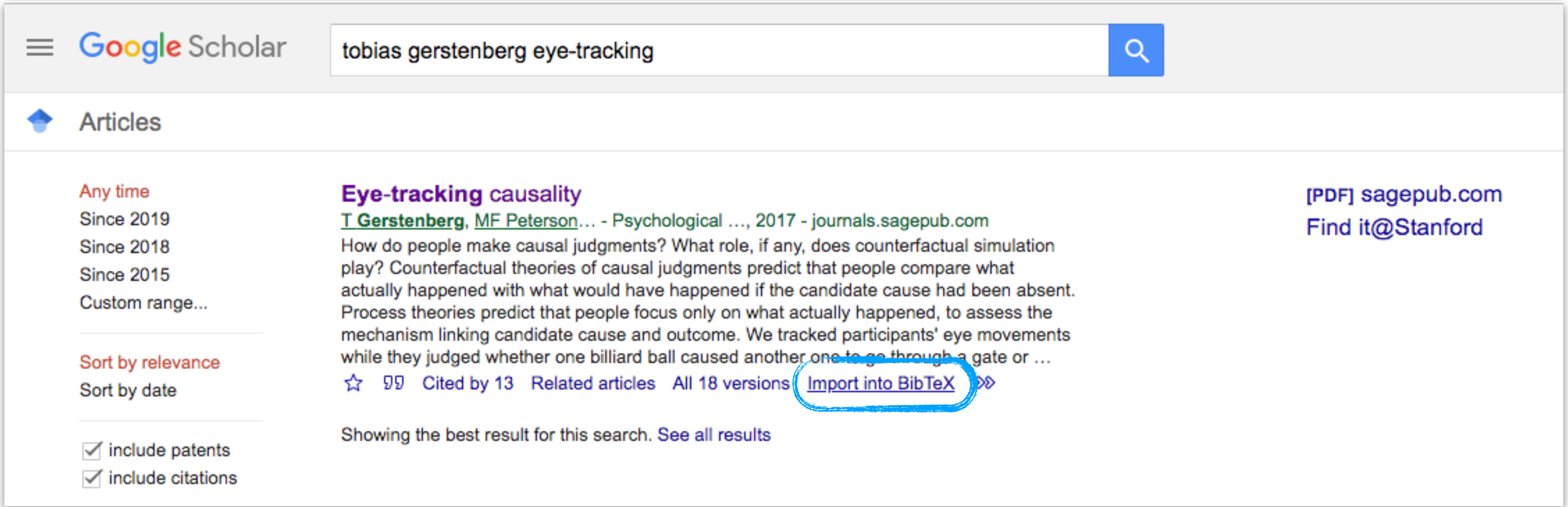
Citations in RMarkdown

Can you point us to an R cookbook resource for integrating APA citations and a bibliography into our R Markdown scripts for the final project?

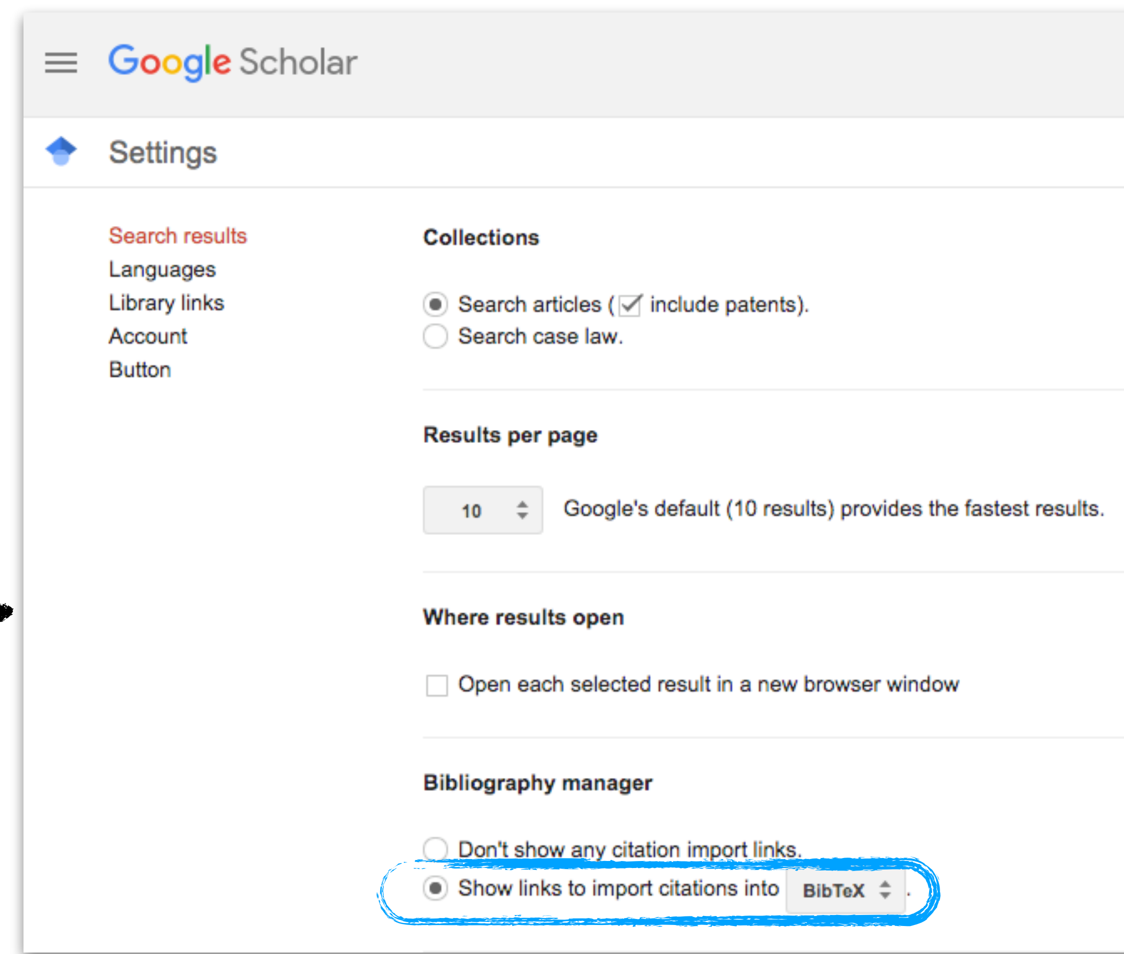
Yes!



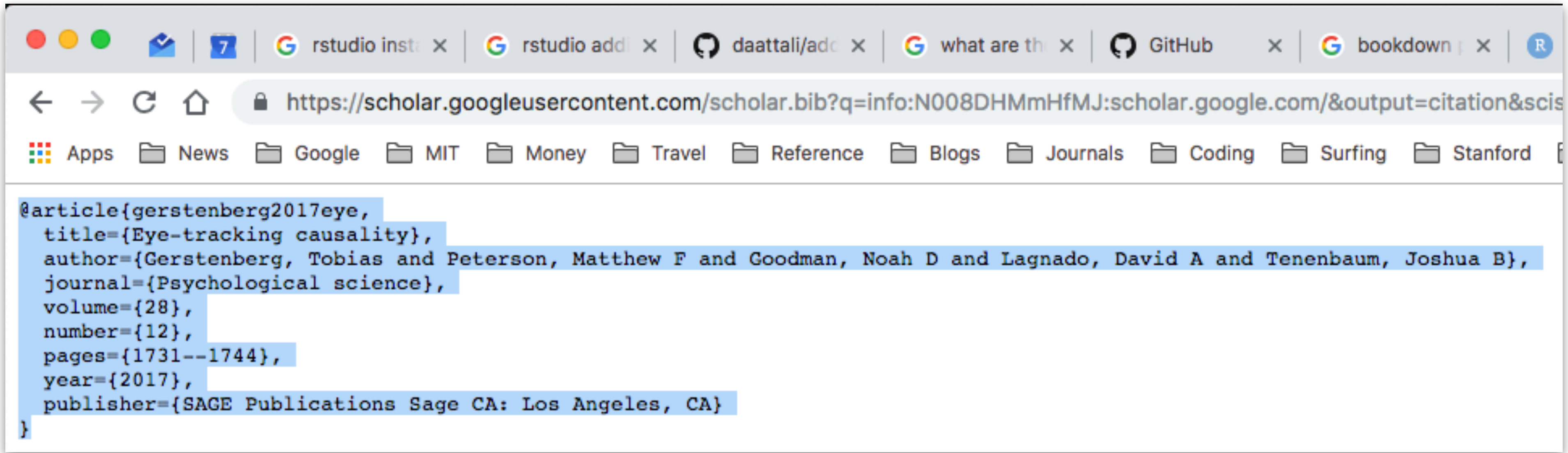
Citations in RMarkdown



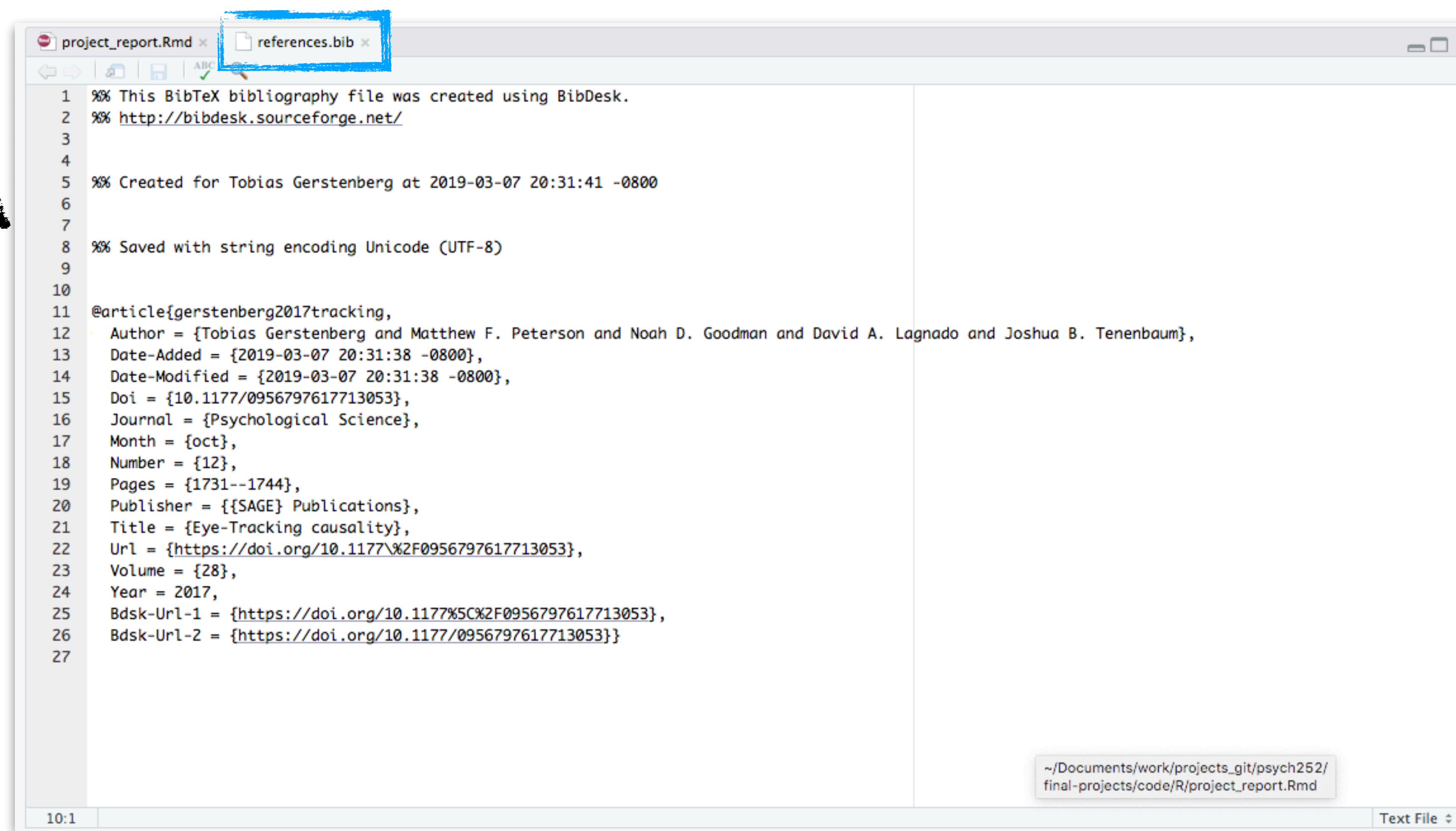
If "Import into BibTex" link is not shown



Citations in RMarkdown

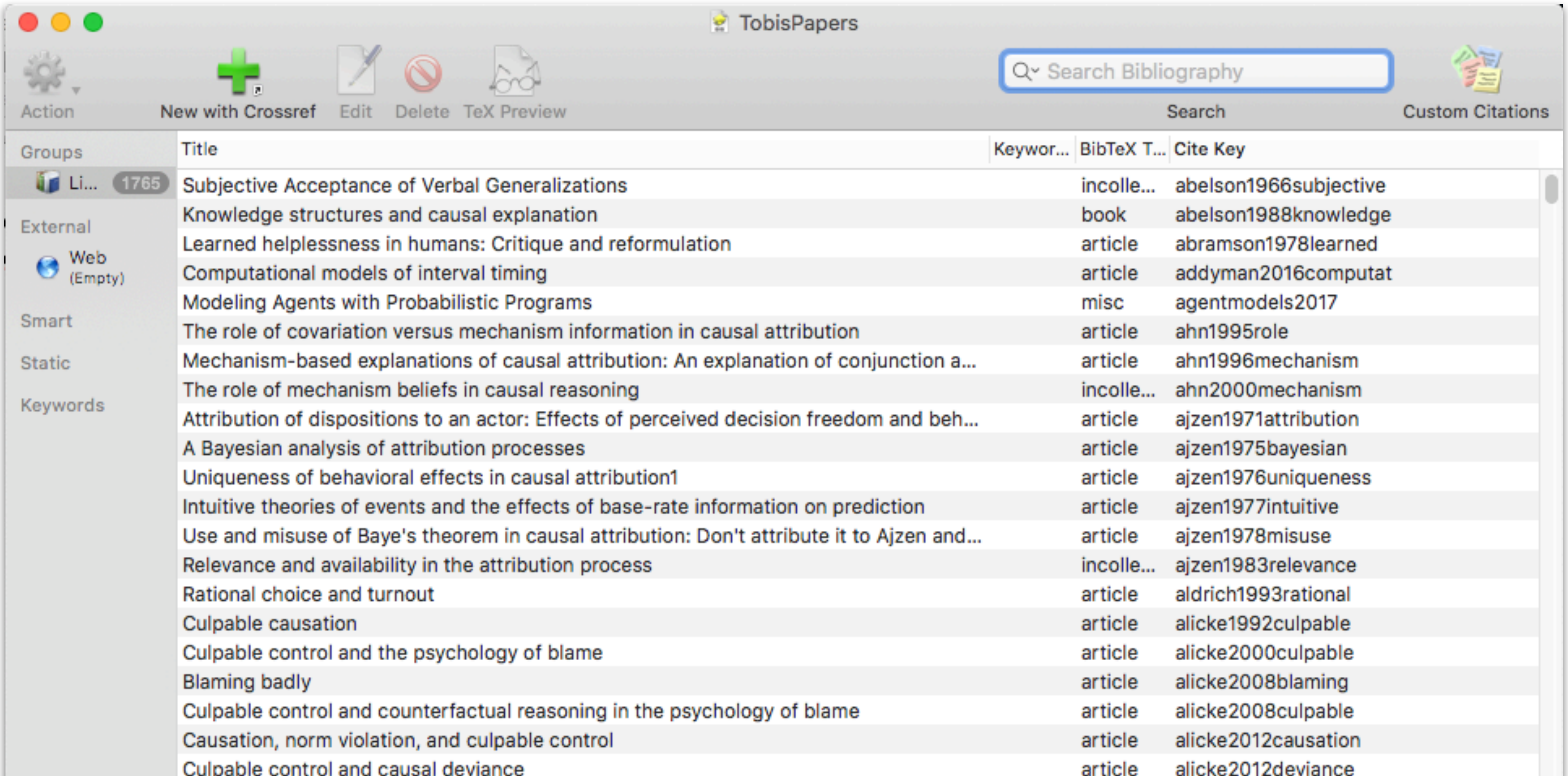


copy and paste



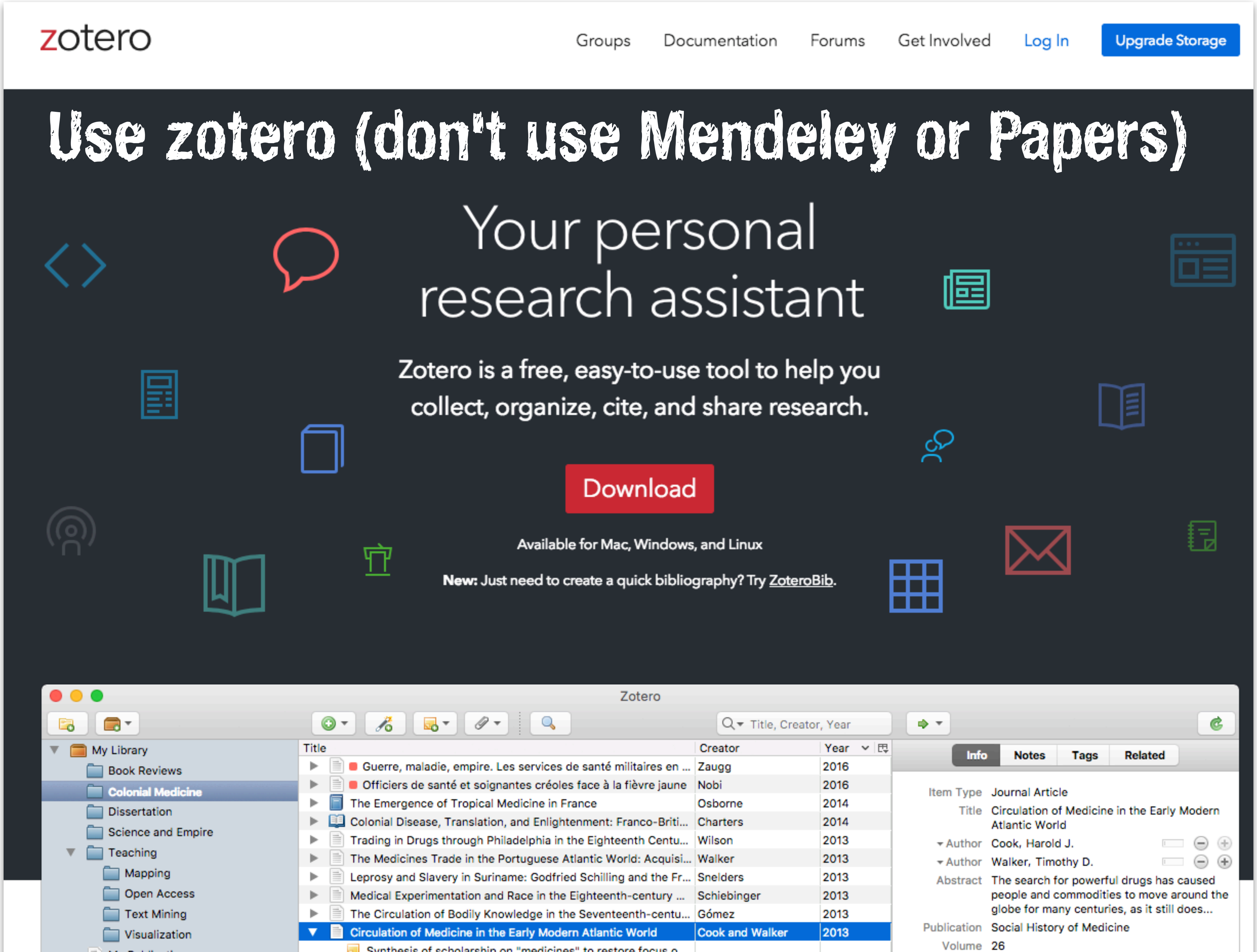
Citations in RMarkdown

Nice reference manager for bibtex files



https://bibdesk.sourceforge.io/

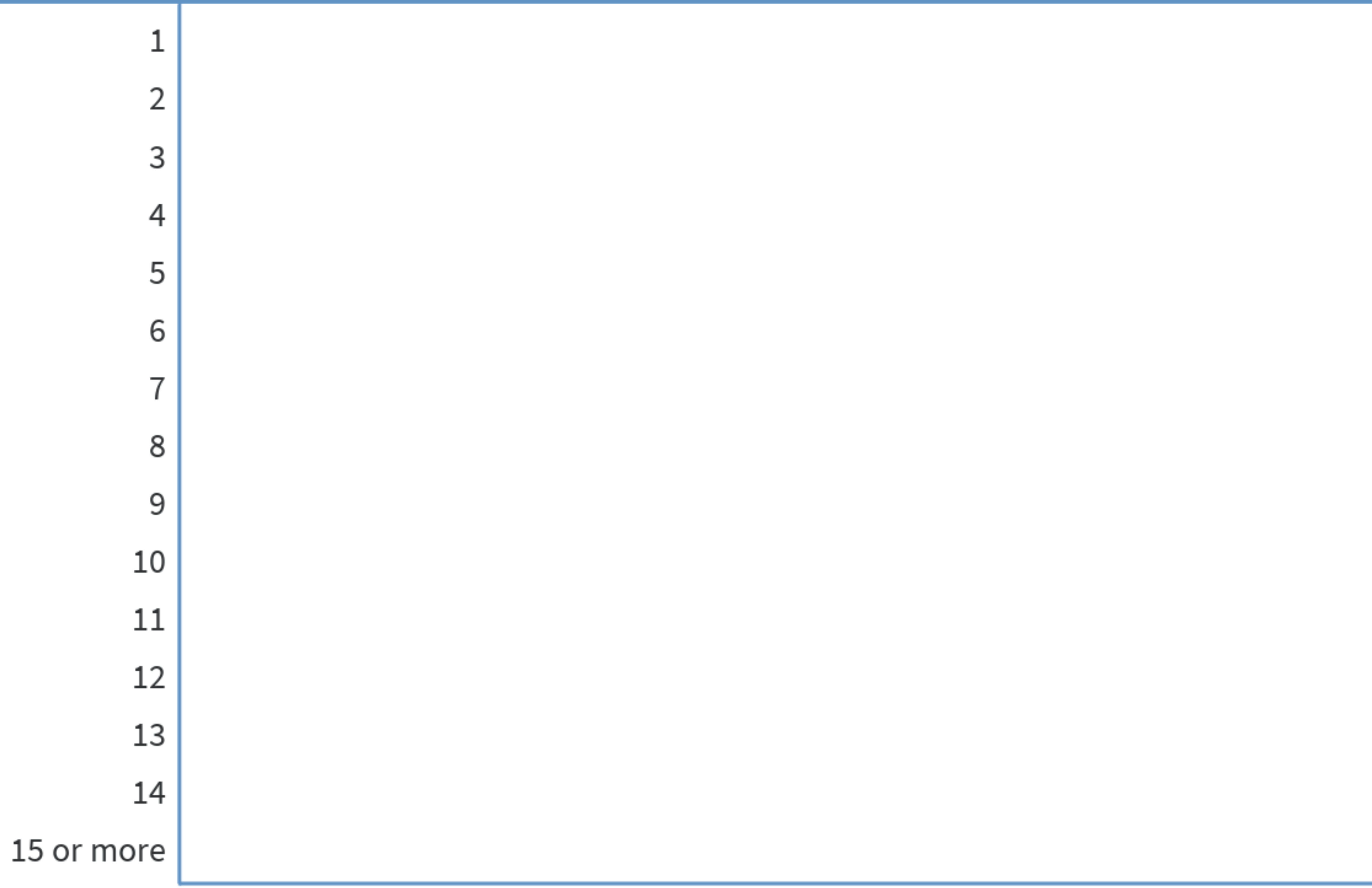
Reference manager



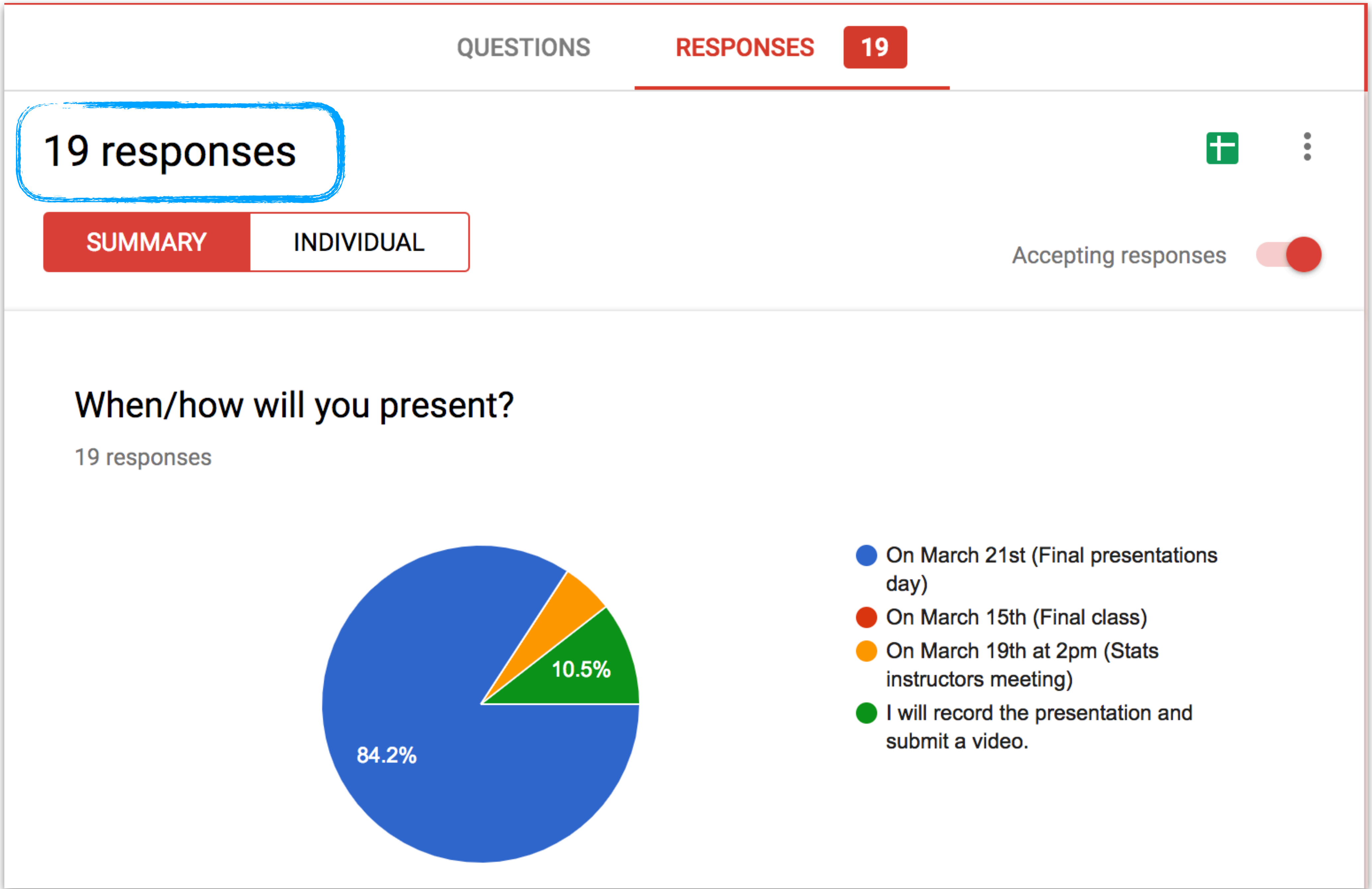
https://www.zotero.org/

Homework 6
Mixed effects models

How many hours did it take you to complete Homework 6?



Final presentations



https://tinyurl.com/psych252presentation

Plan for today

- Doing Bayesian data analysis
  - A simple linear regression
  - Measuring uncertainty: Confidence interval vs. credible interval
- Building Bayesian models with `brms`
  - Model evaluation:
    - Visualizing and interpreting results
    - Testing hypotheses
  - Inference evaluation: Did things work out?
- Some cool examples
  - Evidence for null results
  - Dealing with unequal variance
  - Zero-one inflated beta binomial model
  - Ordinal logistic regression

Doing Bayesian data analysis

A simple linear regression

Model specification

```r
library("greta")
library("tidybayes")

# variables & priors
b0 = normal(0, 10)
b1 = normal(0, 10)
sd = cauchy(0, 3, truncation = c(0, Inf))

# linear predictor
mu = b0 + b1 * attitude$complaints

# observation model (likelihood)
distribution(attitude$rating) = normal(mu, sd)

# define the model
m = model(b0, b1, sd)
```

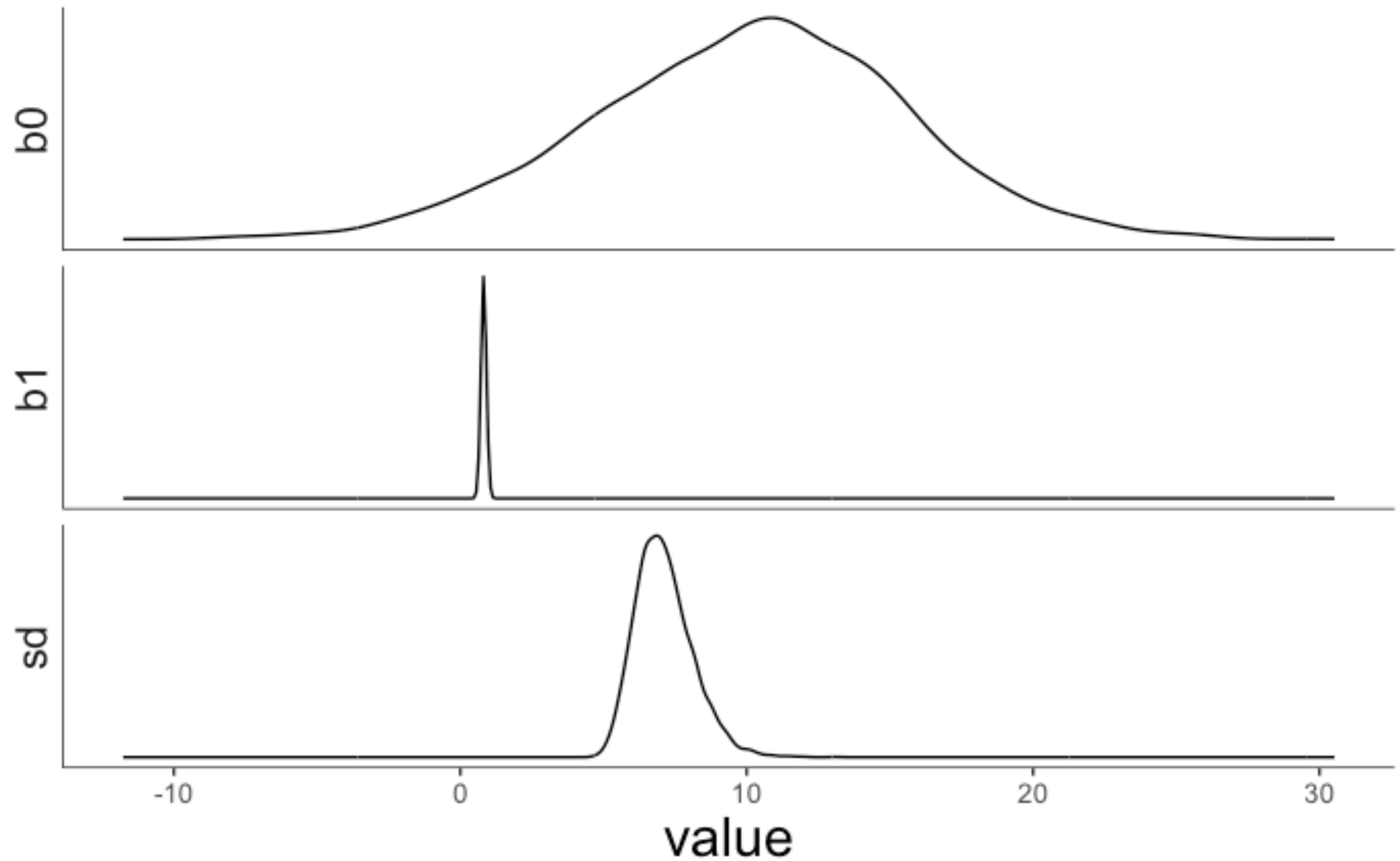
priors

linear combination

Gaussian likelihood

build the model

Summarizing results



- Posterior over each parameter is the result of the Bayesian data analysis.
- no p-values
- no confidence intervals

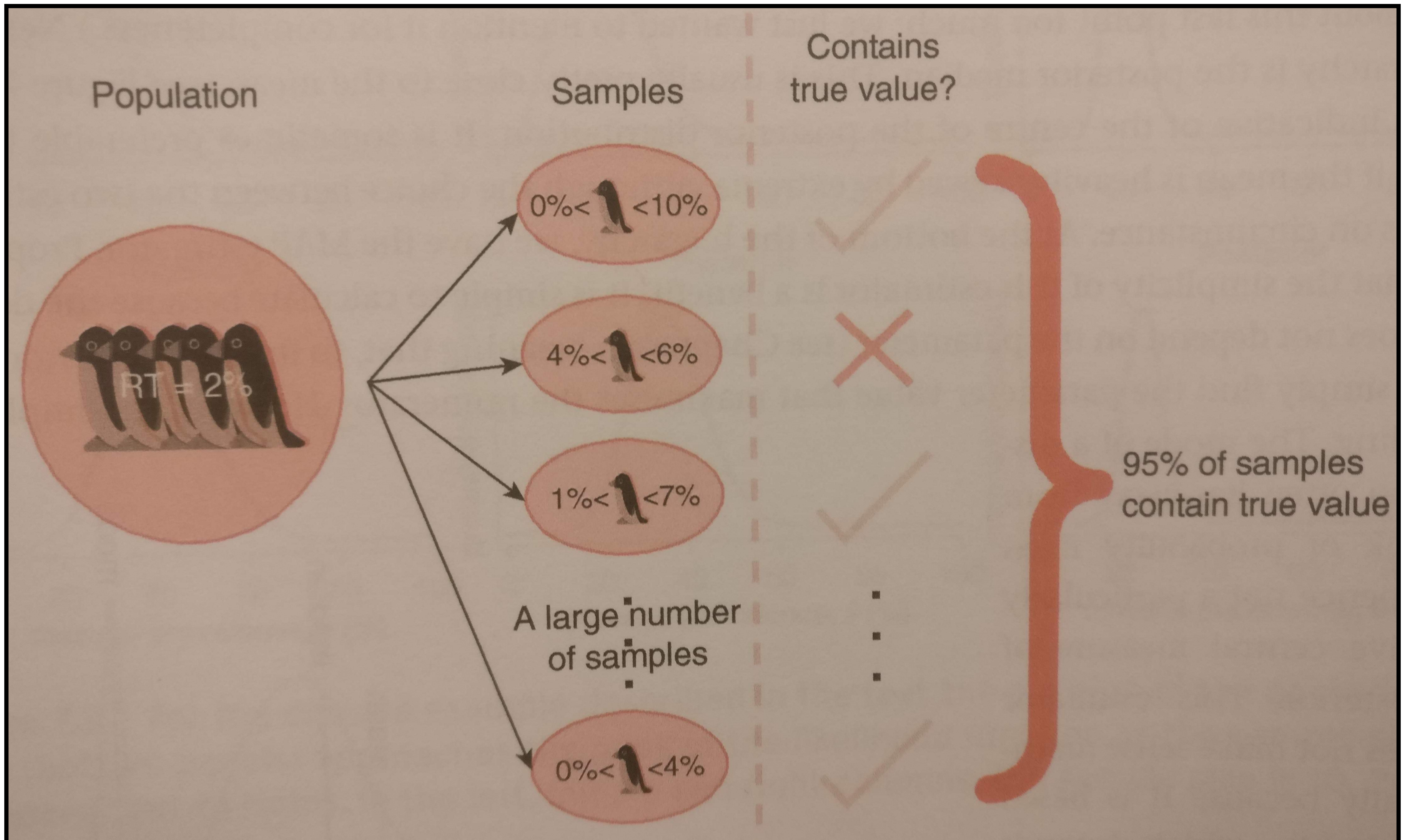
Measuring uncertainty:
Confidence interval vs. credible interval

booh!

yay!

Confidence interval vs. credible interval

"From our research, we concluded that the percentage of penguins with red tails, RT, has a 95% **confidence interval** of 1% < RT < 5%."



For 95% of the (hypothetical) samples, the confidence interval contains the true value.

Confidence interval vs. credible interval

"From our research, we concluded that the percentage of penguins with red tails, RT, has a 95% **credible interval** of 0% < RT < 4%."

Straightforward interpretation

There is a 95% probability that the percentage of penguins with red tails lies in the range of 0% < RT < 4%.

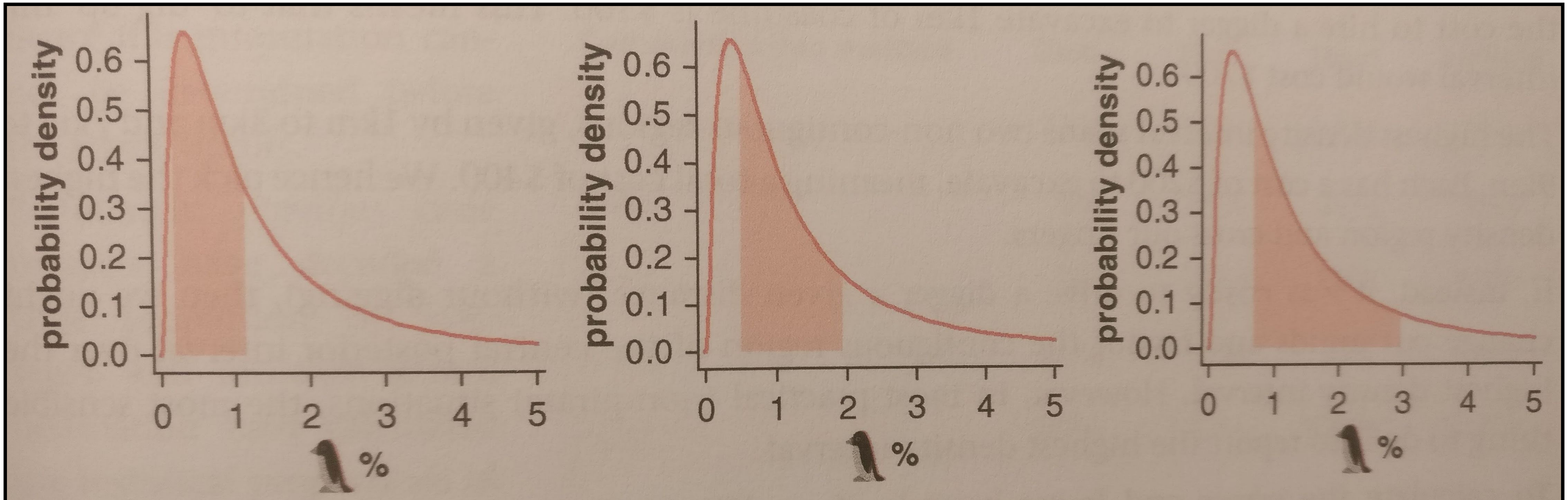
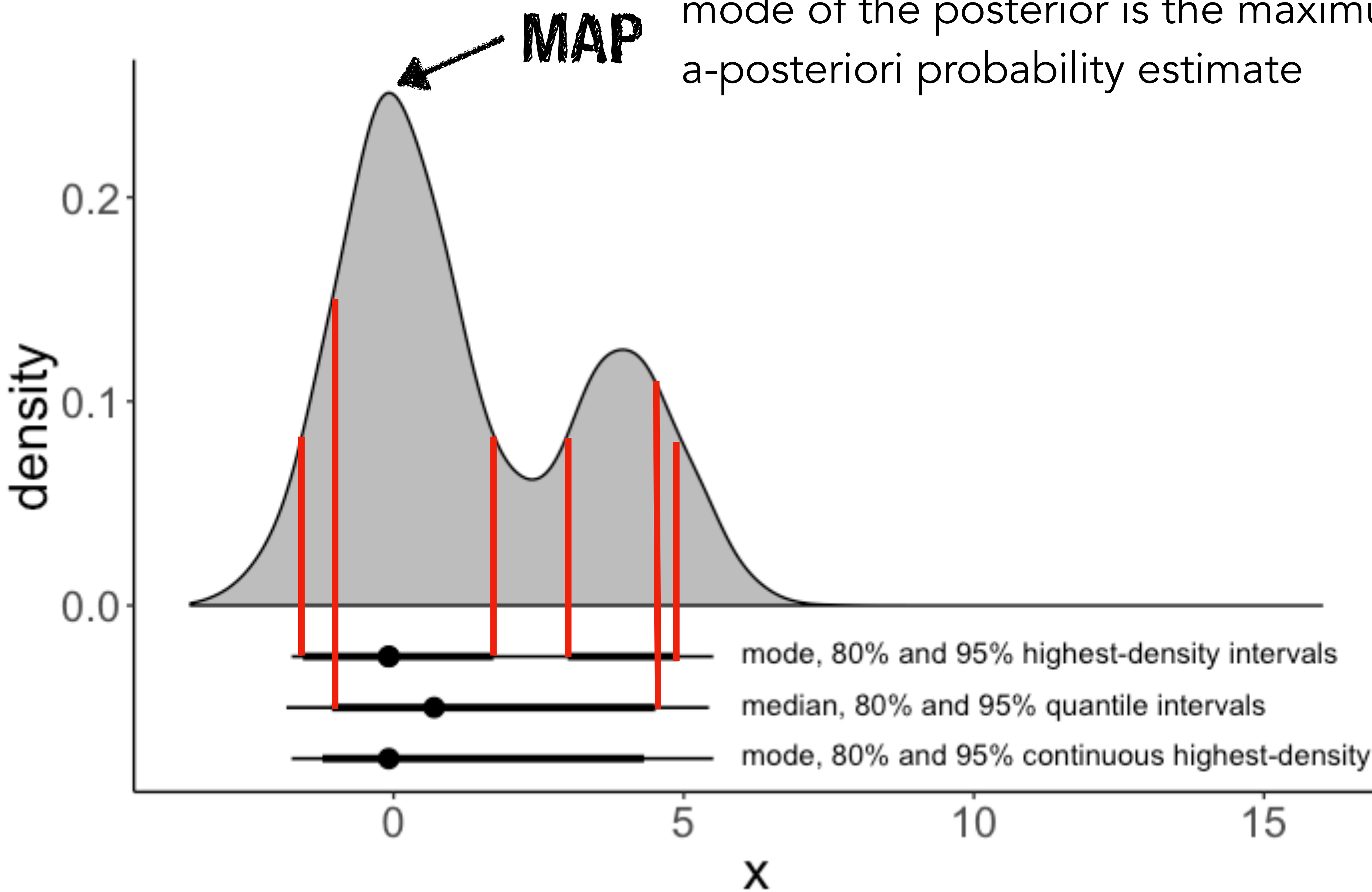


Figure 7.8 Three examples of 50% credible intervals for a parameter representing the proportion of penguins with red tails.

Different kinds of credible intervals

mode of the posterior is the maximum a-posteriori probability estimate



Building Bayesian models with `brms`

Software packages

`library("brms")`

Bayesian regression modeling with Stan

- 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

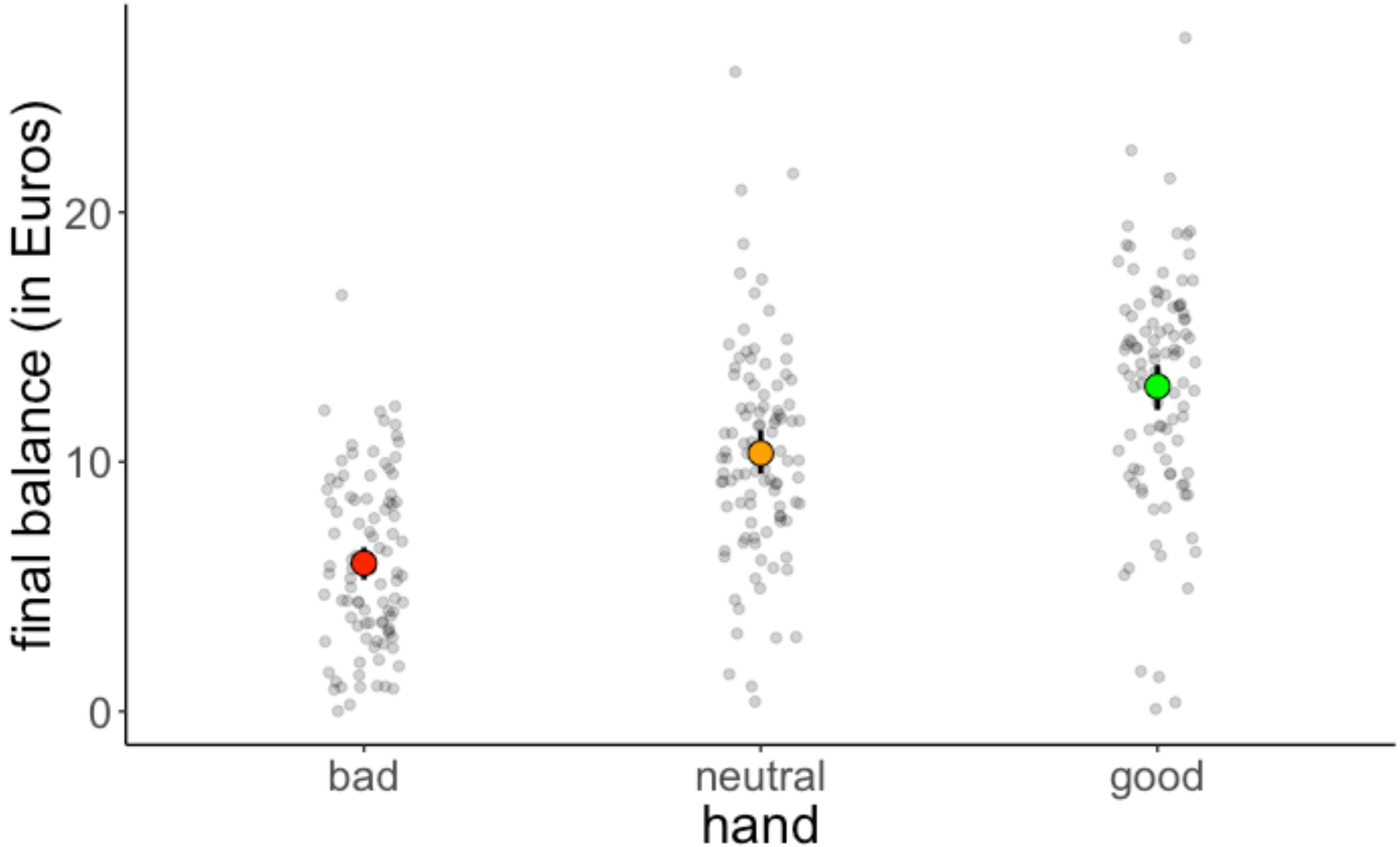
Software packages

Stan

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

https://mc-stan.org/

Poker data



Using `lm()`

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

fit.lm %>% summary()
```

```
Call:
lm(formula = balance ~ 1 + hand, data = df.poker)

Residuals:
     Min       1Q   Median       3Q      Max
-12.9264  -2.5902  -0.0115   2.6573  15.2834

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.9415     0.4111  14.451  < 2e-16 ***
handneutral   4.4051     0.5815   7.576 4.55e-13 ***
handgood      7.0849     0.5815  12.185  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.111 on 297 degrees of freedom
Multiple R-squared:  0.3377, Adjusted R-squared:  0.3332
F-statistic:  75.7 on 2 and 297 DF,  p-value: < 2.2e-16
```

Using `brm()`

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

fit.brm %>% summary()
```

```
Family: gaussian
  Links: mu = identity; sigma = identity
Formula: balance ~ 1 + hand
   Data: df.poker (Number of observations: 300)
Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
         total post-warmup samples = 4000

Population-Level Effects:
            Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
Intercept       5.94      0.41     5.14     6.74       3495 1.00
handneutral     4.41      0.59     3.23     5.53       3125 1.00
handgood        7.09      0.59     5.92     8.24       3830 1.00

Family Specific Parameters:
      Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
sigma     4.12      0.17     3.81     4.47       3745 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).
```

Comparison between `lm()` and `brm()`

`lm()`

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      5.9415     0.4111  14.451  < 2e-16 ***
handneutral      4.4051     0.5815   7.576 4.55e-13 ***
handgood         7.0849     0.5815  12.185  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

`brm()`

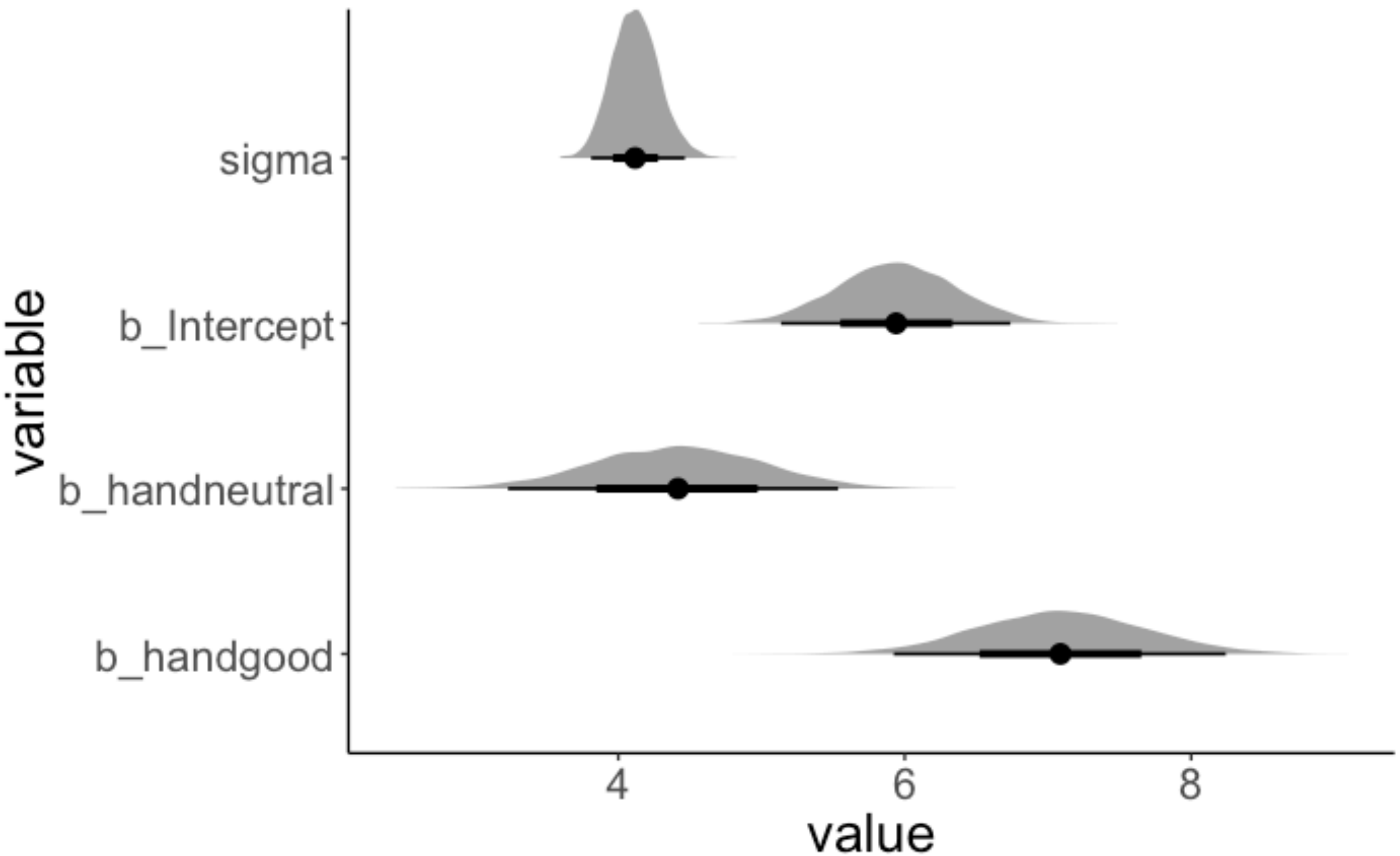
```
Population-Level Effects:
            Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
Intercept       5.94      0.41     5.14     6.74       3495 1.00
handneutral     4.41      0.59     3.23     5.53       3125 1.00
handgood        7.09      0.59     5.92     8.24       3830 1.00
```

almost identical results!

Model evaluation

Visualizing and interpreting results

Summary of the posterior

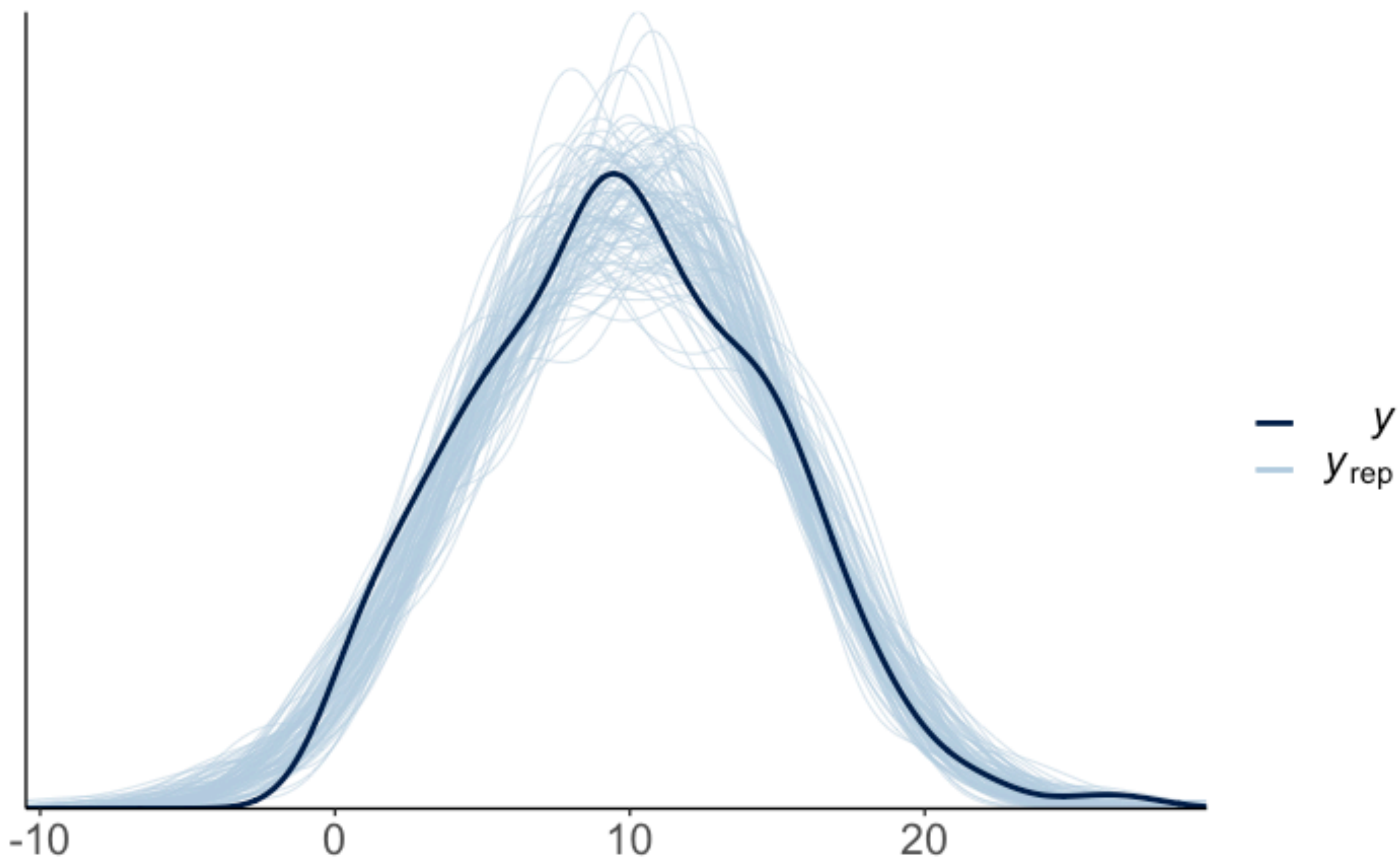


MAP estimate and 95% highest density interval

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

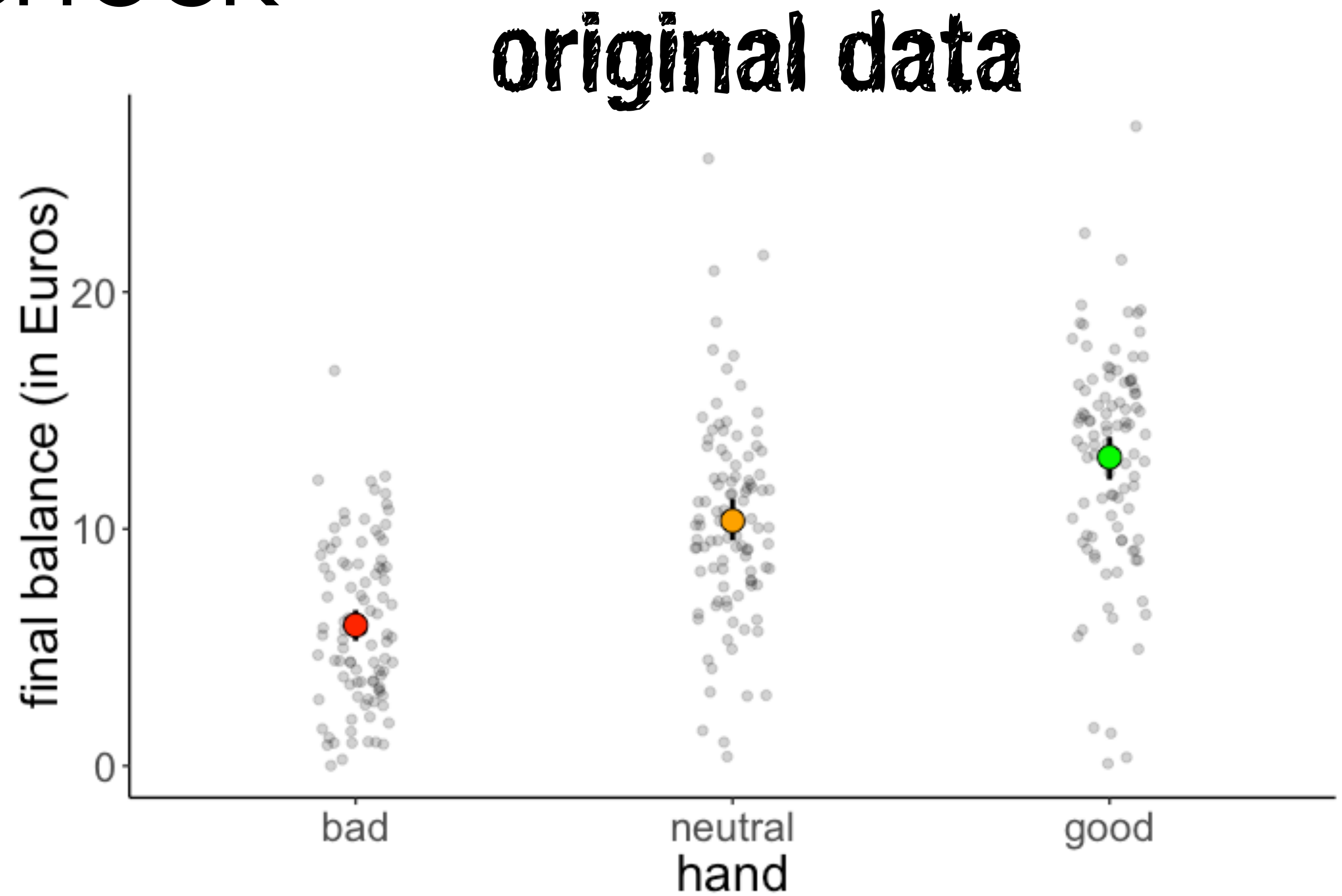
Posterior predictive check

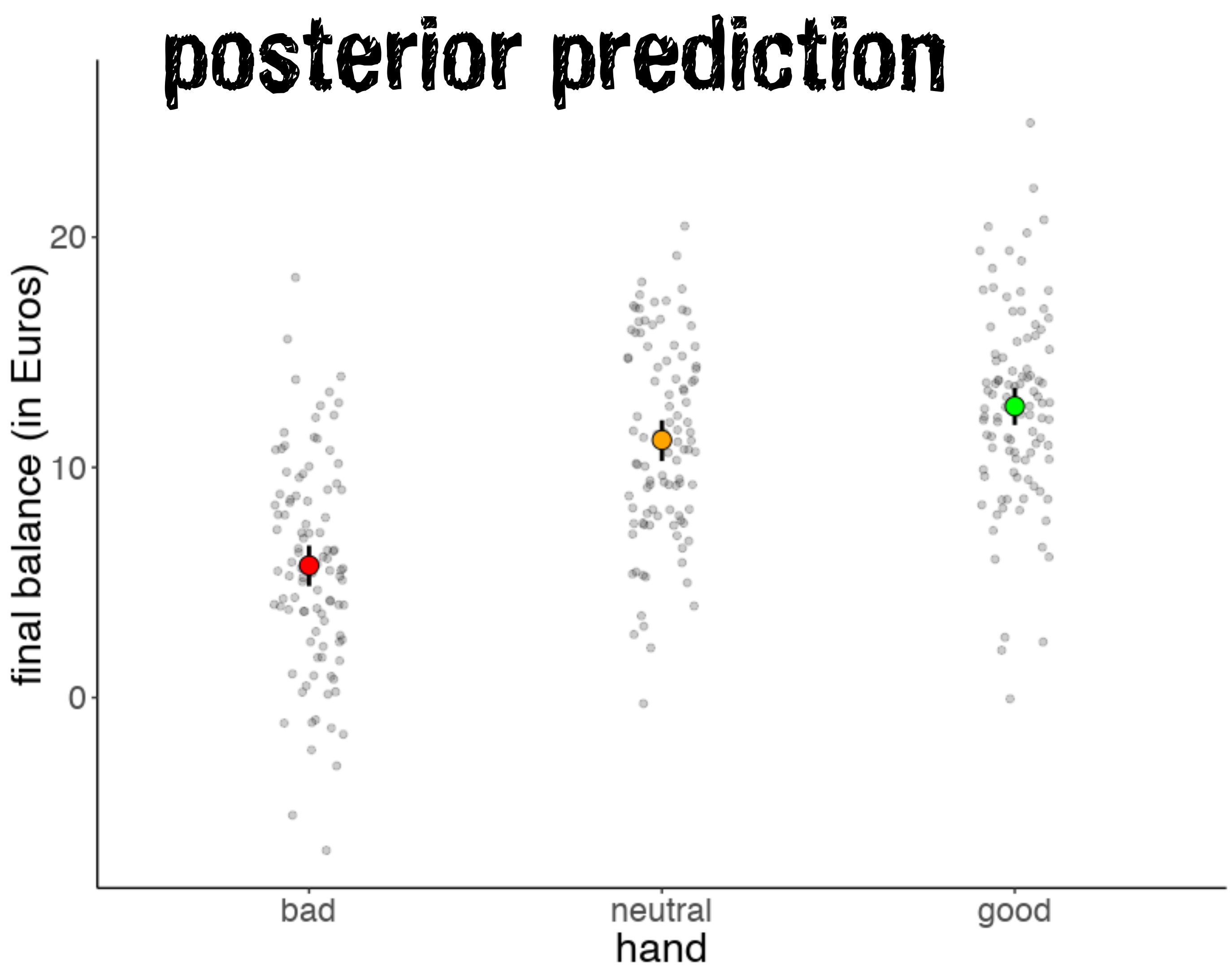
```
pp_check(fit.brm, nsamples = 100)
```



The model accurately captures the distribution of the response variable

Posterior predictive check

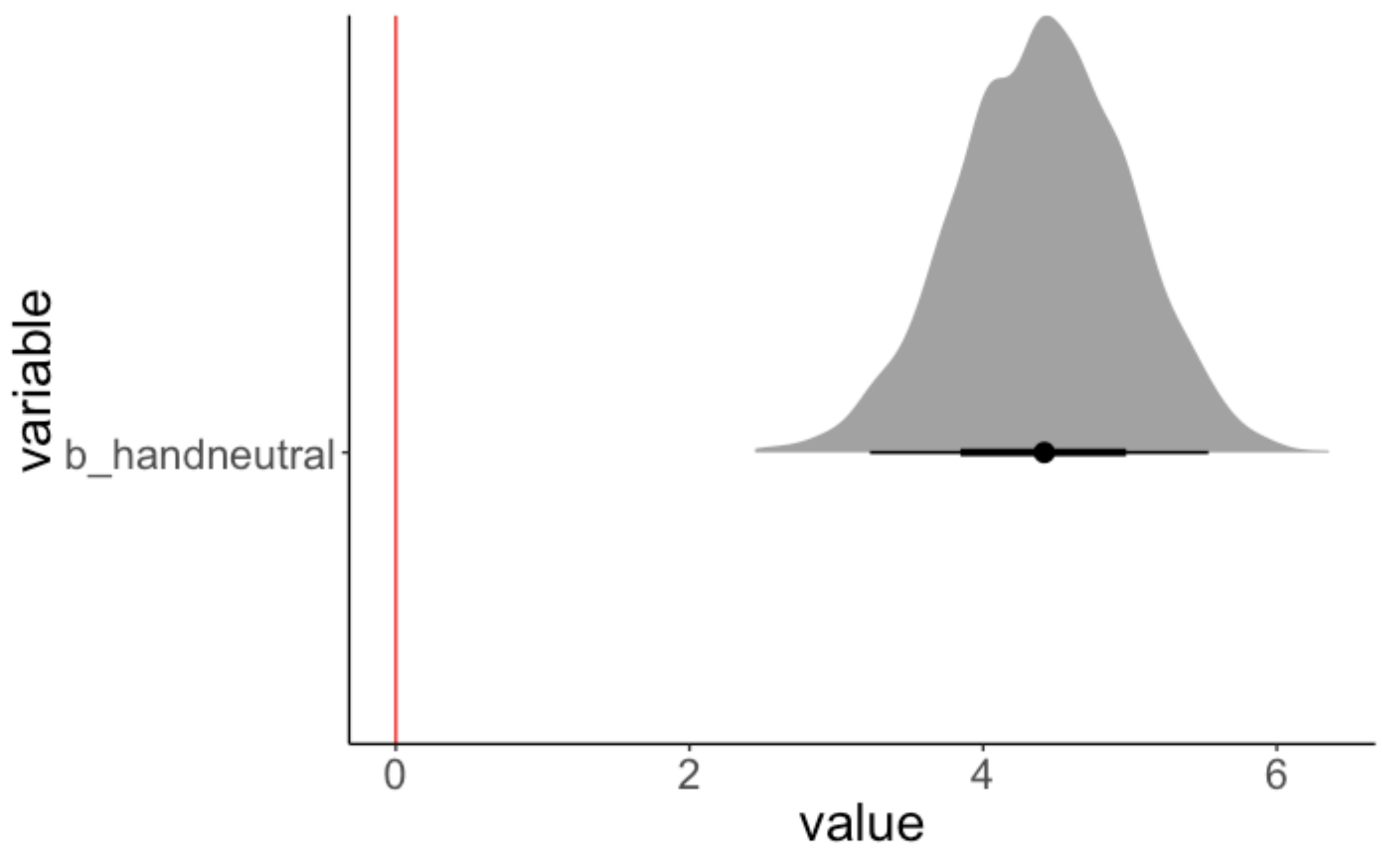




Testing hypotheses

Asking questions based on the posterior

Do neutral hands earn more money than bad hands?



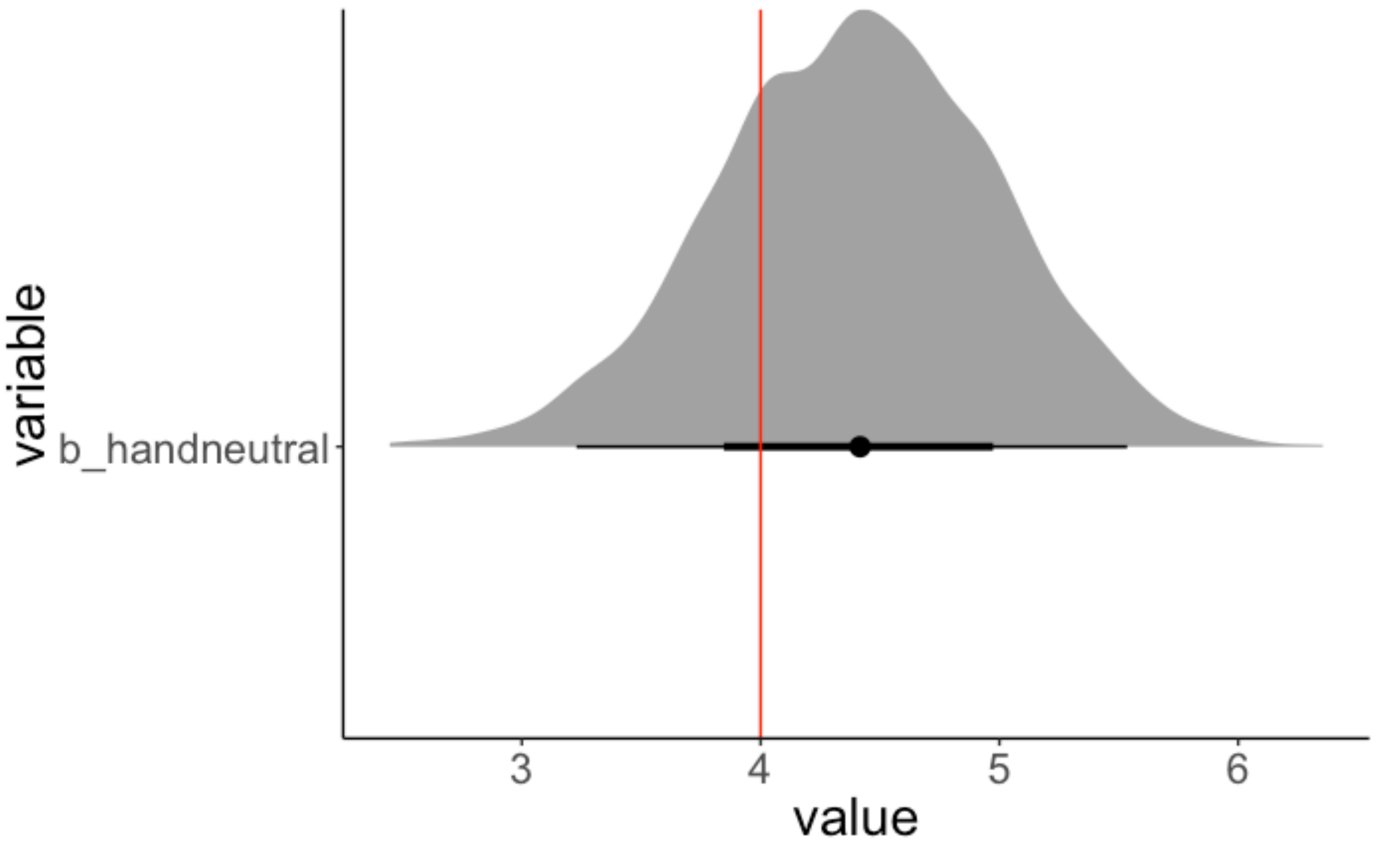
What's the probability that handneutral is less than 0?

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

$p = 0$

Asking questions based on the posterior

Do neutral hands earn much more money than bad hands?



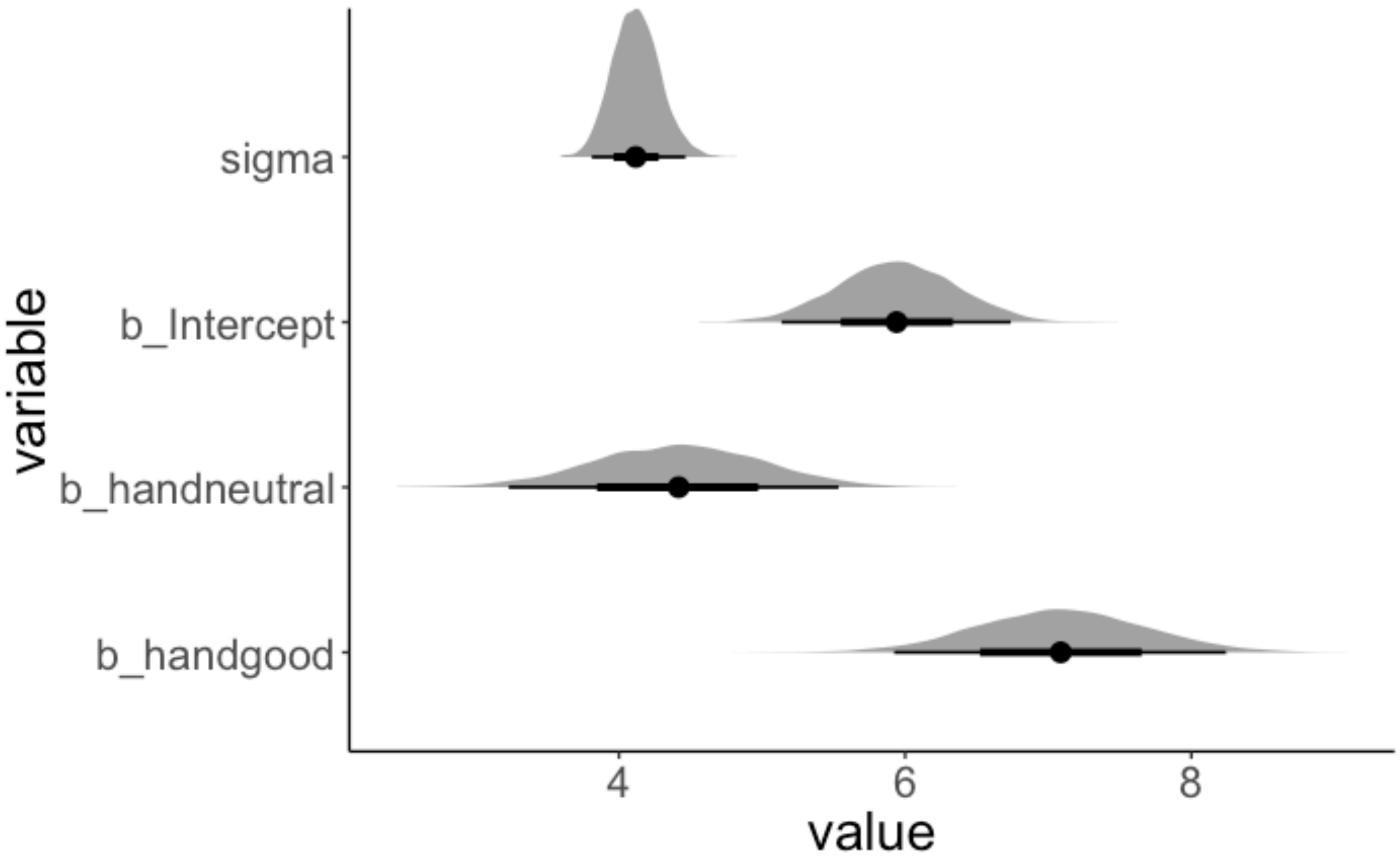
What's the probability that handneutral is **more than 4**?

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

p = 0.75

Asking questions based on the posterior

Do good hands make twice as much as bad hands?

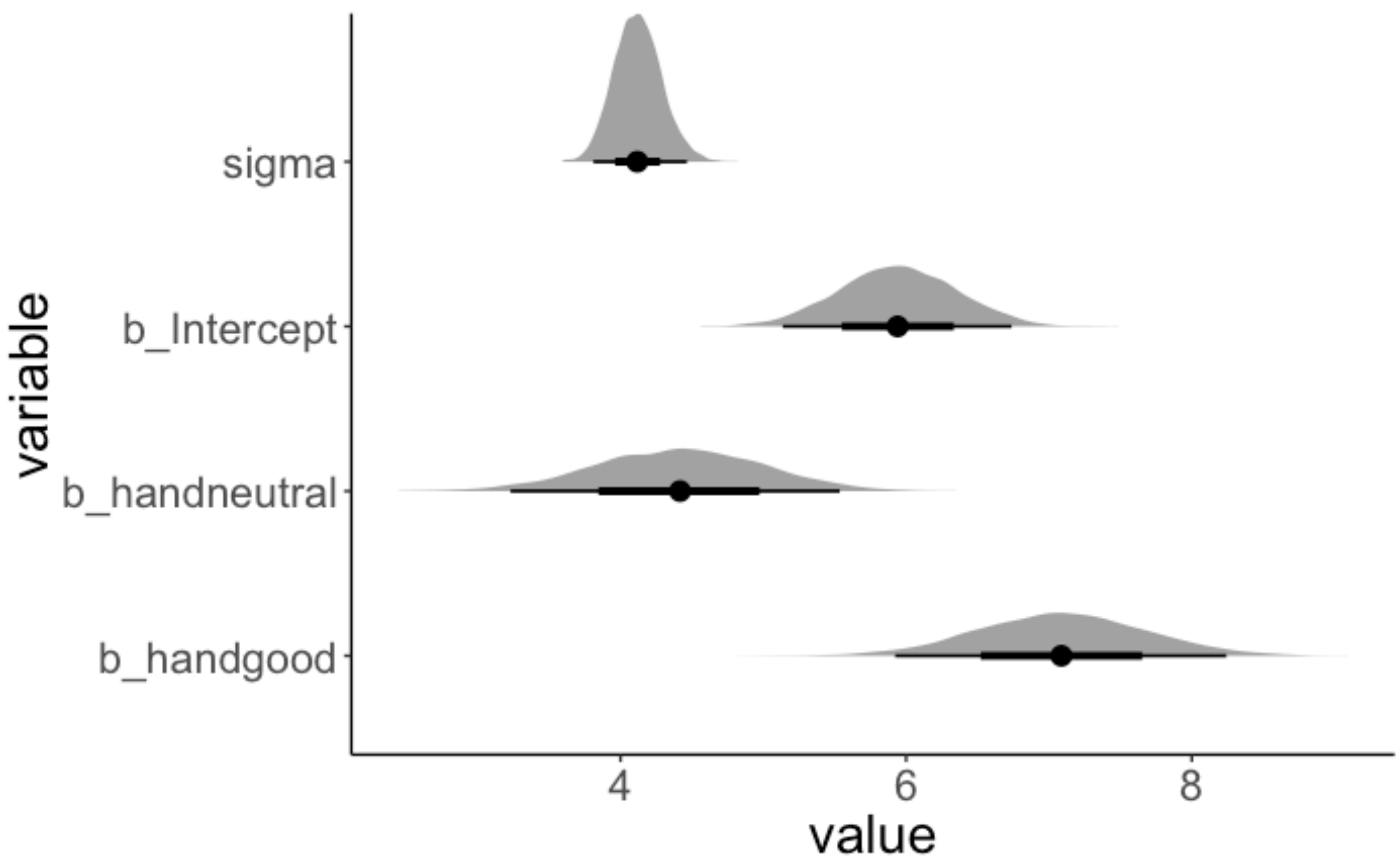



```r
hypothesis(fit.brm,
           hypothesis = "handgood + Intercept > 2 * Intercept")
```

p = 0.89

Asking questions based on the posterior

Are neutral hands worse than bad and good hands combined?




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

$p = 0.04$

Testing hypothesis

```r
df.hypothesis = fit.brm %>%
  posterior_samples() %>%
  clean_names() %>%
  select(starts_with("b_")) %>%
  mutate(neutral = b_intercept + b_handneutral,
         bad_good_average = (b_intercept + b_intercept + b_handgood)/2,
         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

```r
df.hypothesis %>%
  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
- No need to control for multiple comparisons (since we don't use a heuristic criterion for rejecting null hypotheses)!
- No need for setting up planned linear contrasts!

Bayes factor

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

Two hypotheses

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

Odds form = Bayes factor

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

If priors are equal

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

Fit two models and compare

```r
fit.brm1 = brm(formula = balance ~ 1 + hand,
               data = df.poker,
               save_all_pars = T,
               file = "brm_factor1")

fit.brm2 = brm(formula = balance ~ 1 + hand + skill,
               data = df.poker,
               save_all_pars = T,
               file = "brm_factor2")
```

```r
bayes_factor(fit.brm2, fit.brm1)
```

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

relative evidence for one model over the other

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

Wetzels, R. & Wagenmakers, E.-J. (2012). A default Bayesian hypothesis test for correlations and partial correlations. Psychonomic Bulletin & Review.

Inference evaluation:
Did things work out?

What about the priors?

```r
fit.brm = brm(formula = balance ~ 1 + hand,
              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

```r
fit.brm2 = brm(
  formula = balance ~ 1 + hand,
  family = "gaussian",
  data = df.poker,
  prior = c(
    prior(normal(0, 10), class = "b", coef = "handgood"),
    prior(normal(0, 10), class = "b", coef = "handneutral"),
    prior(student_t(3, 3, 10), class = "Intercept"),
    prior(student_t(3, 0, 10), class = "sigma")
  ),
  inits = list(
    list(Intercept = 0, sigma = 1, handgood = 5, handneutral = 5),
    list(Intercept = -5, sigma = 3, handgood = 2, handneutral = 2),
    list(Intercept = 2, sigma = 1, handgood = -1, handneutral = 1),
    list(Intercept = 1, sigma = 2, handgood = 2, handneutral = -2)
  ),
  iter = 4000,
  warmup = 1000,
  chains = 4,
  file = "brm2",
  seed = 1
)
```

priors

initialization

how many runs in the inference chain

how long for the warmup

how many chains

save the model result

make reproducible

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

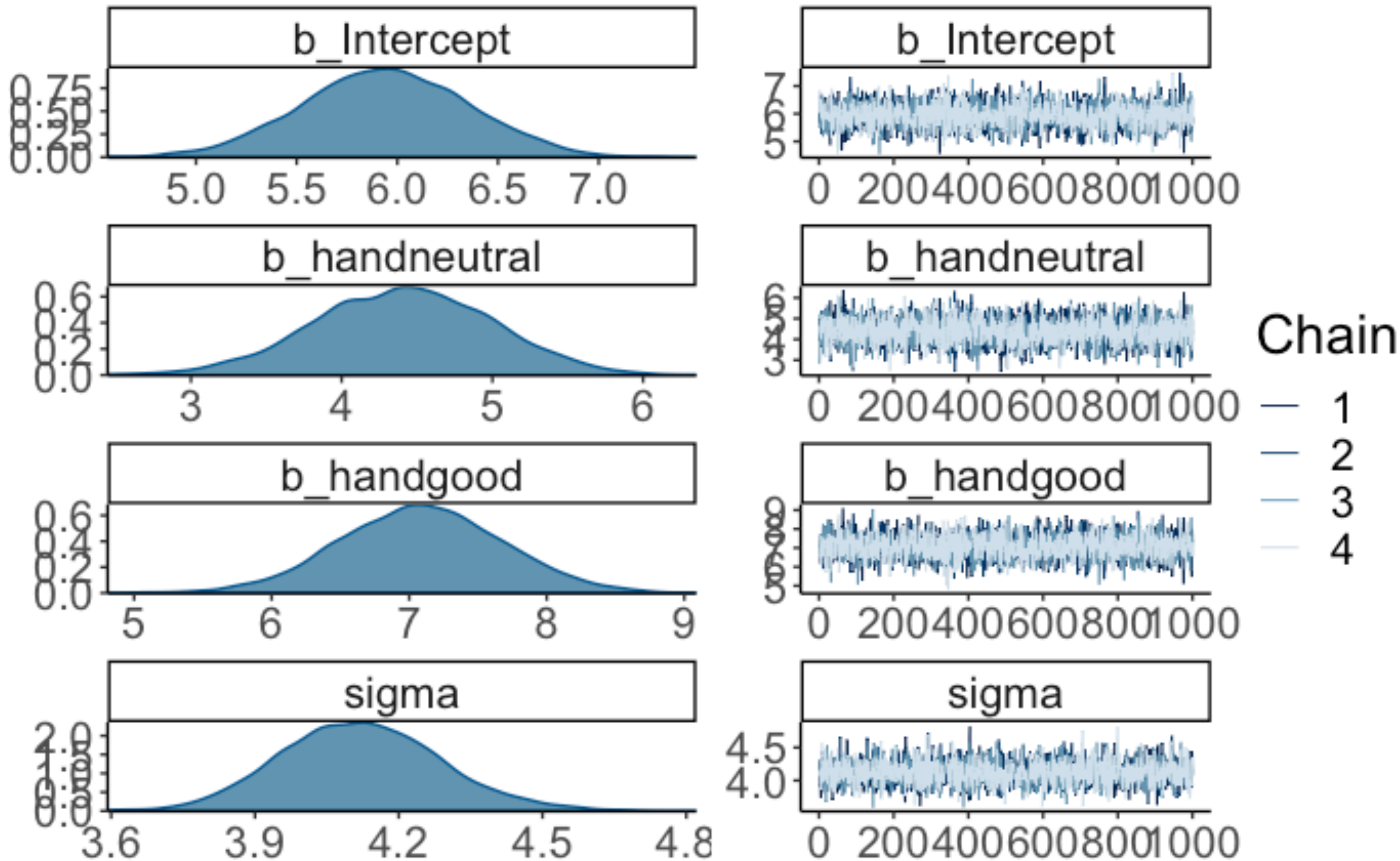
Turned into Stan code

```
// 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

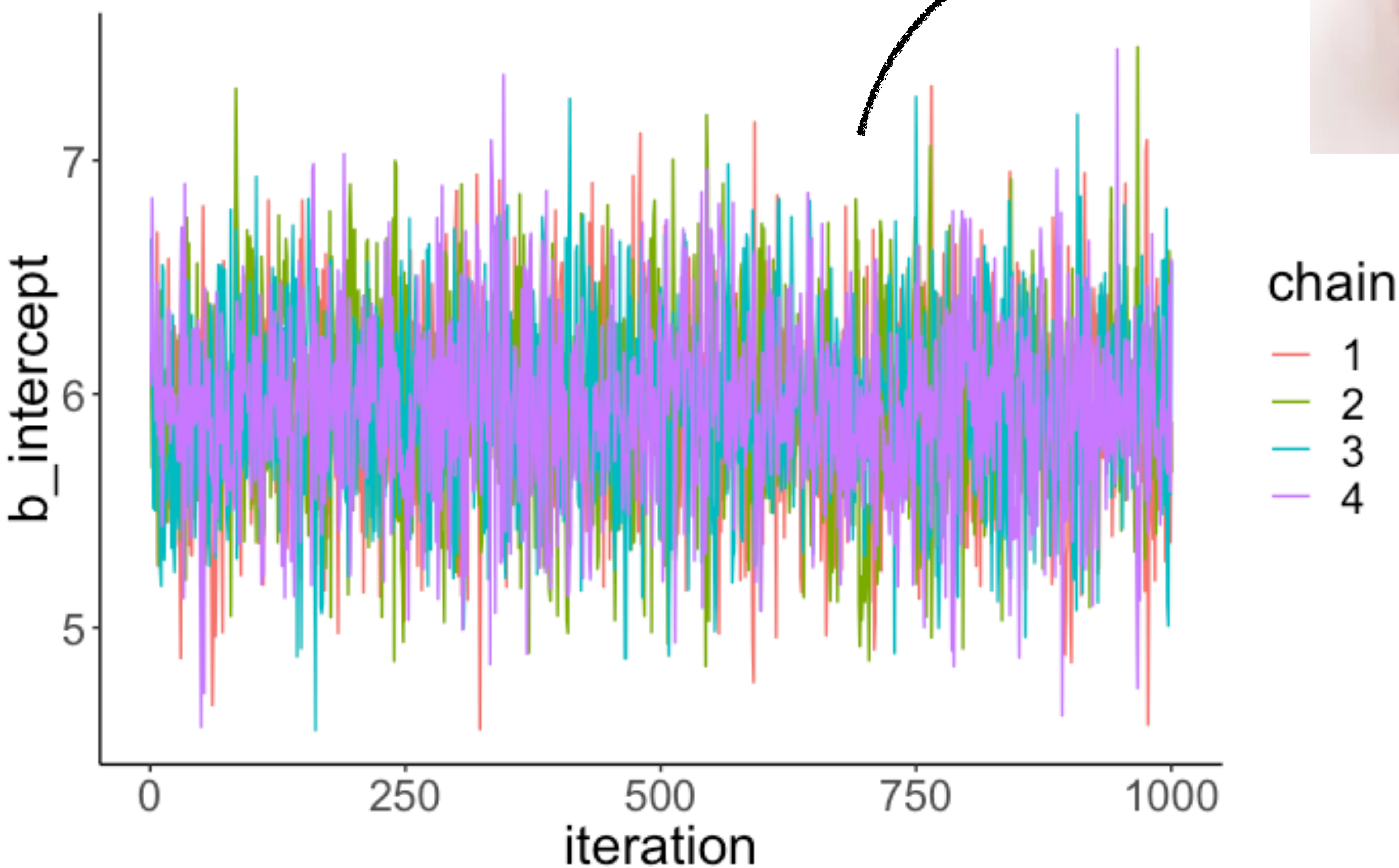
Can we trust the model results?

```r
plot(fit.brm)
```



Can we trust the model results?

looks like a fuzzy carterpillar



When things don't work out

```r
df.data = tibble(y = c(-1, 1))

fit.brm1 = brm(
  data = df.data,
  family = gaussian,
  formula = y ~ 1,
  prior = c(
    prior(uniform(-1e10, 1e10), class = Intercept),
    prior(uniform(0, 1e10), class = sigma)
  ),
  inits = list(
    list(Intercept = 0, sigma = 1),
    list(Intercept = 0, sigma = 1)
  ),
  iter = 4000,
  warmup = 1000,
  chains = 2,
  file = "fit_brm1"
)
```

only two data points!

incredibly wide uniform priors

When things don't work out

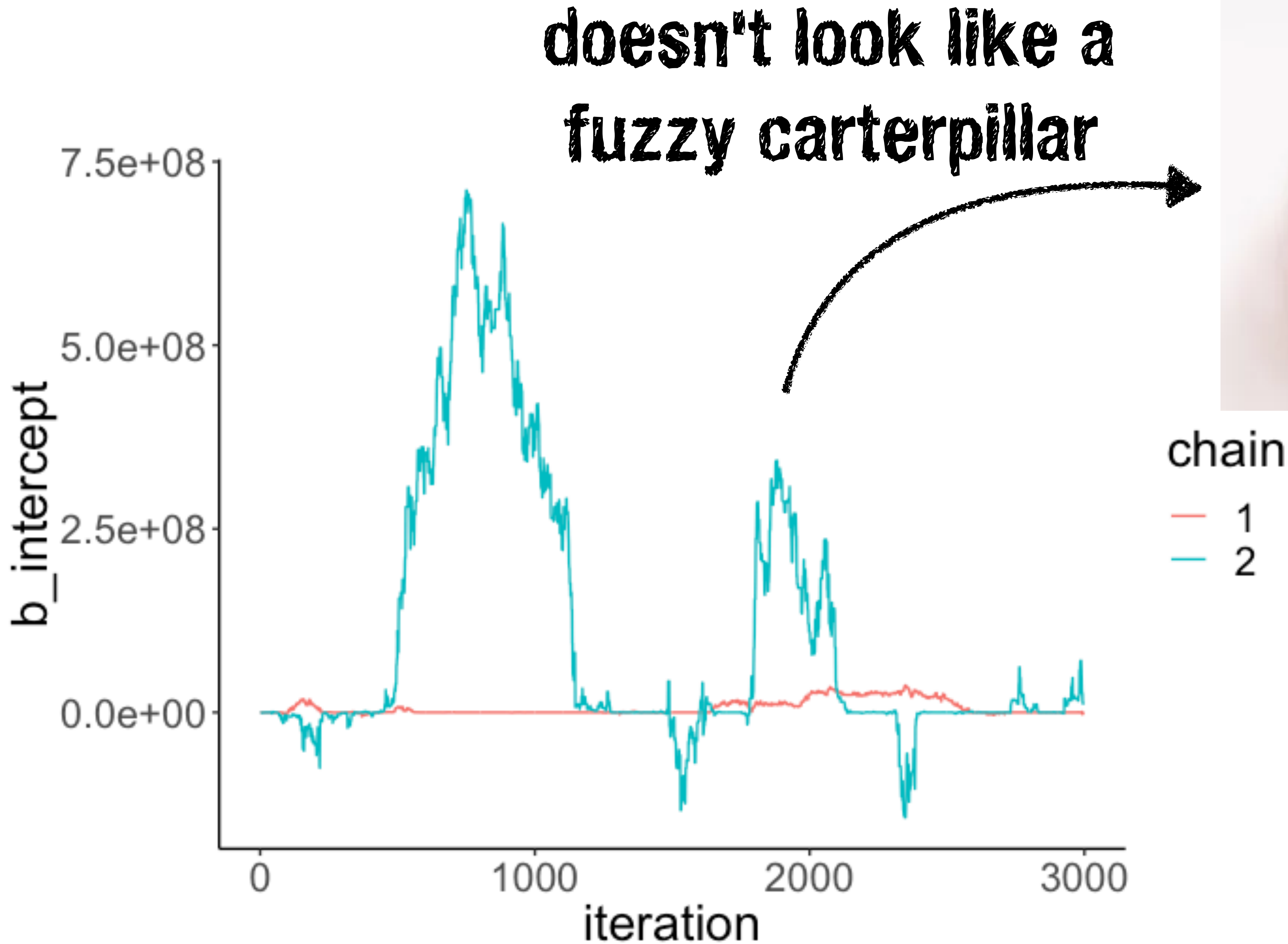
`summary(fit.brm1)`

```
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 362 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: 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 56599494.07 142352414.04 -36858534.50 548777591.78         12 1.20

Family Specific Parameters:
          Estimate    Est.Error l-95% CI       u-95% CI Eff.Sample Rhat
sigma 269115068.56 931020177.84   757.39 2652989409.36         37 1.06

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).
```

should be 1

should be much higher

When things don't work out



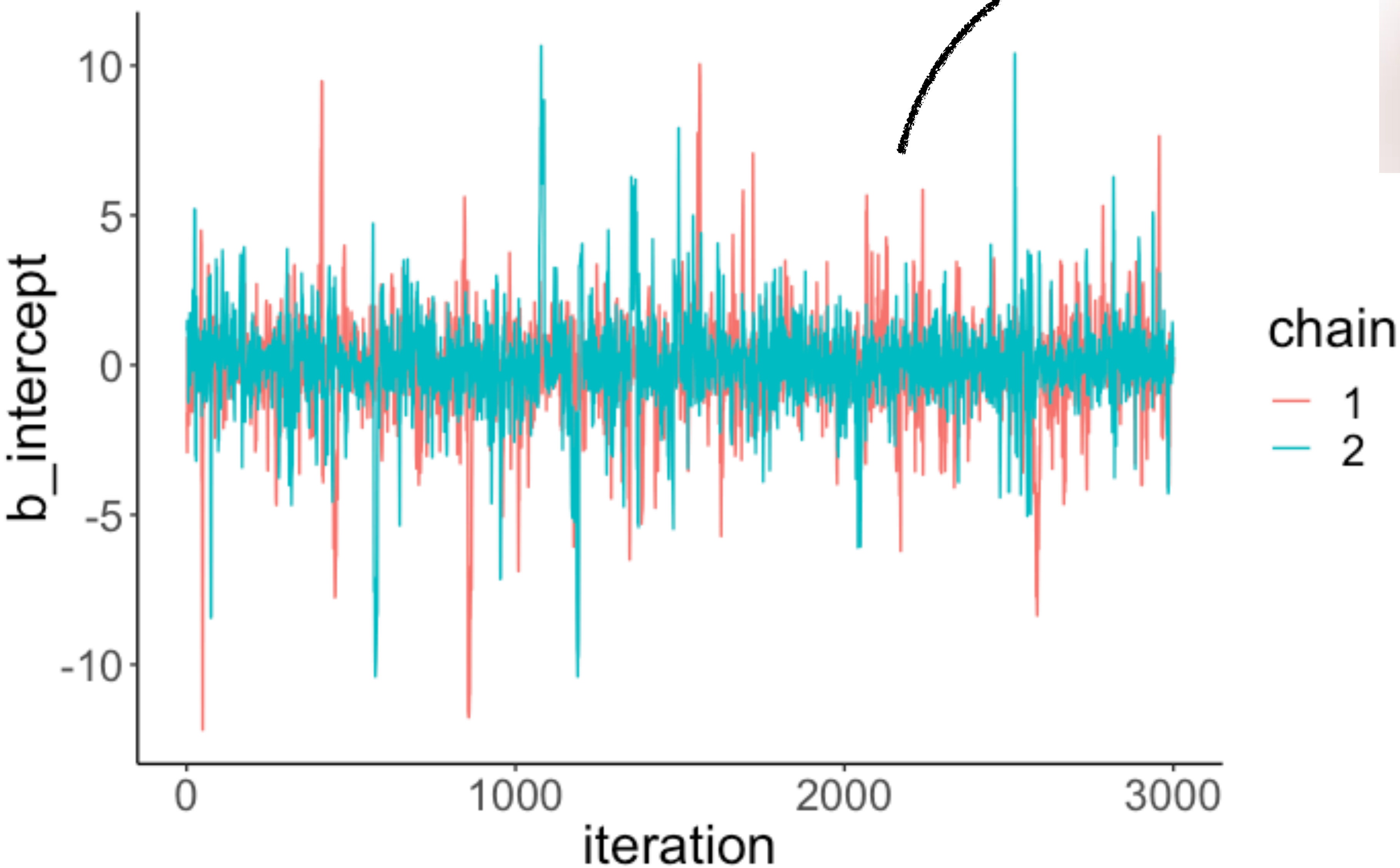
Having somewhat informative priors fixes things

```r
fit.brm2 = brm(
  data = df.data,
  family = gaussian,
  formula = y ~ 1,
  prior = c(
    prior(normal(0, 10), class = Intercept), # more reasonable priors
    prior(cauchy(0, 1), class = sigma)
  ),
  iter = 4000,
  warmup = 1000,
  chains = 2,
  seed = 1,
  file = "fit_brm2"
)
```

```
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

looks like a fuzzy carterpillar



Interim summary

- Bayesian models are very flexible: They support asking many different kinds of questions about the data
- Asking these questions is easy: we just directly use the samples from the posterior
- We can use the Bayes factor to compare different models (they need not be nested)
- There are ways for us to check whether the inference worked
- If we have little data, then weakly informative priors help

Some cool examples

Evidence for null results

frontiers in Psychology

Front Psychol. 2014; 5: 781.
Published online 2014 Jul 29. doi: 10.3389/fpsyg.2014.00781

PMCID: PMC4114196
PMID: 25120503

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

Zoltan Dienes*

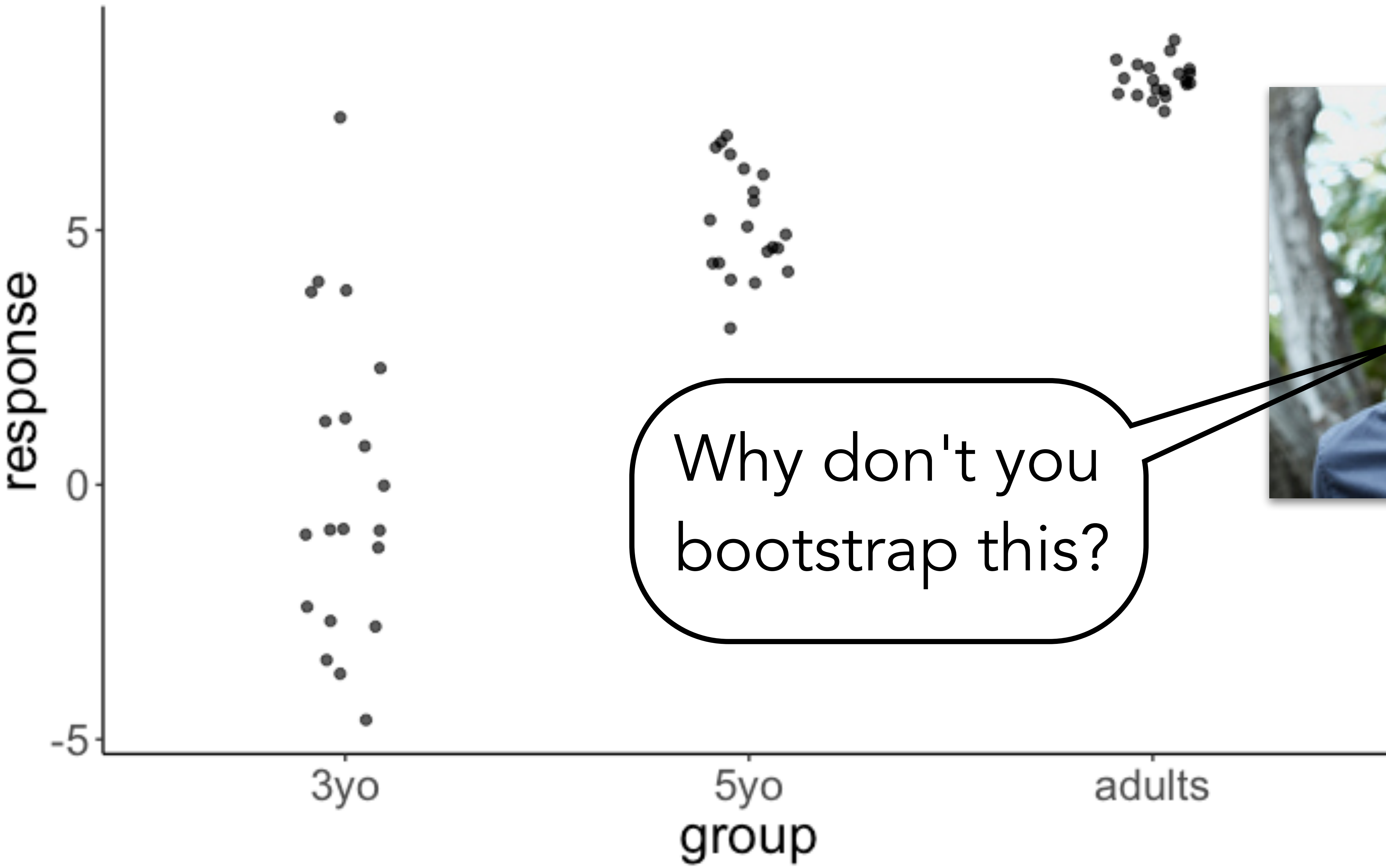
‣ Author information ‣ Article notes ‣ Copyright and License information Disclaimer

[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 649 Related articles All 14 versions Import into BibTeX »

- 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)

Dealing with unequal variance

Unequal variance aka heteroscedasticity



Why don't you bootstrap this?

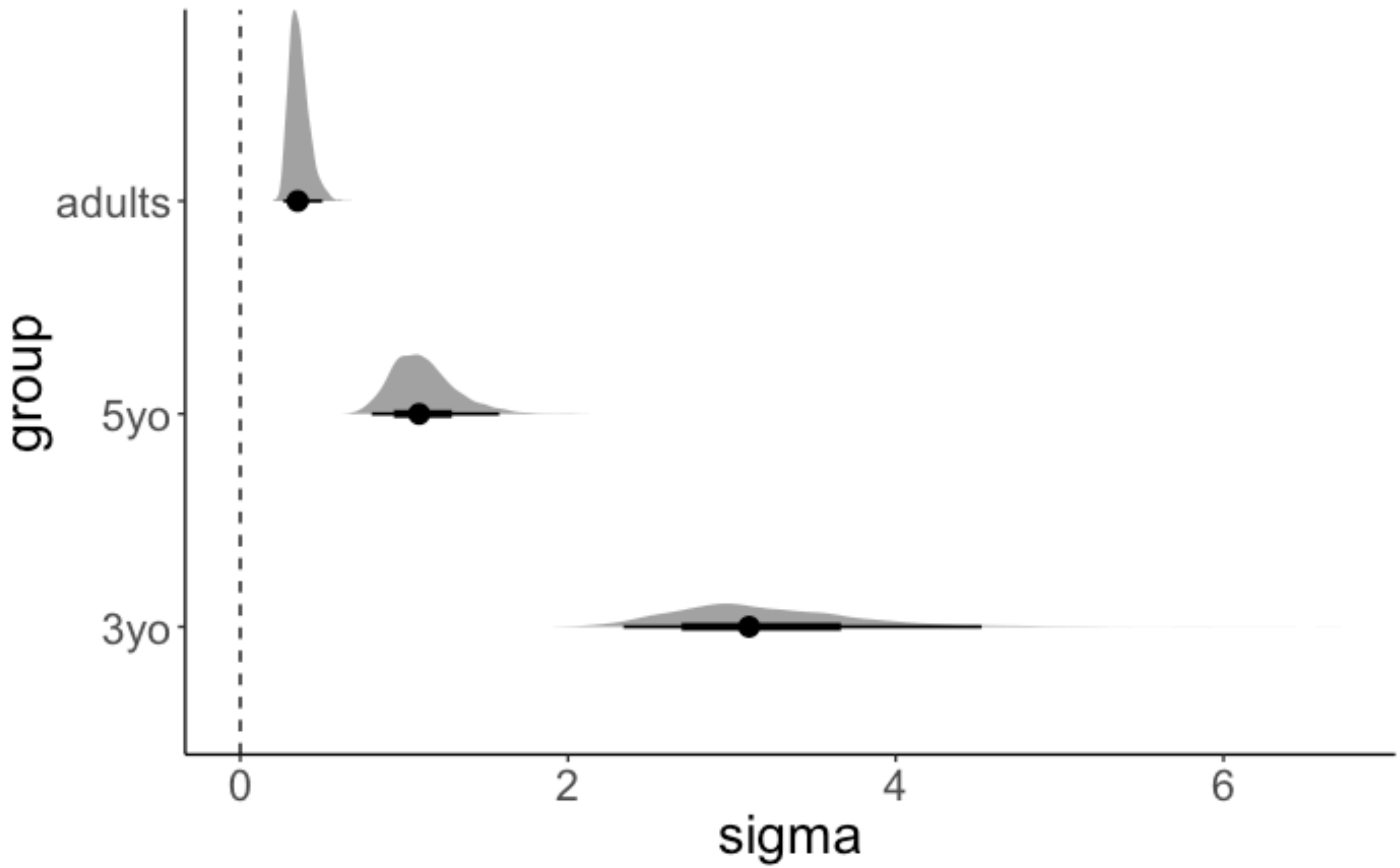
Unequal variance aka heteroscedasticity

```
fit.variance = brm(
  formula = bf(response ~ group,
               sigma ~ group),
  data = df.variance,
  file = "variance")
```

```
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 Eff.Sample Rhat
Intercept             0.00      0.72    -1.38     1.45       1273 1.00
sigma_Intercept       1.15      0.17     0.85     1.51       2033 1.00
group5yo              5.16      0.77     3.60     6.62       1424 1.00
groupadults           7.96      0.72     6.49     9.37       1276 1.00
sigma_group5yo       -1.05      0.23    -1.51    -0.59       2355 1.00
sigma_groupadults    -2.19      0.23    -2.65    -1.74       2231 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).
```

Model estimates both group effects **and** variance!

Unequal variance aka heteroscedasticity



Zero-one inflated beta binomial model

Example taken from …

How to analyze visual analog (slider) scale data?

A reasonable choice might be the zero-one-inflated beta model

Feb 18, 2019 · 25 min read · psychology, statistics

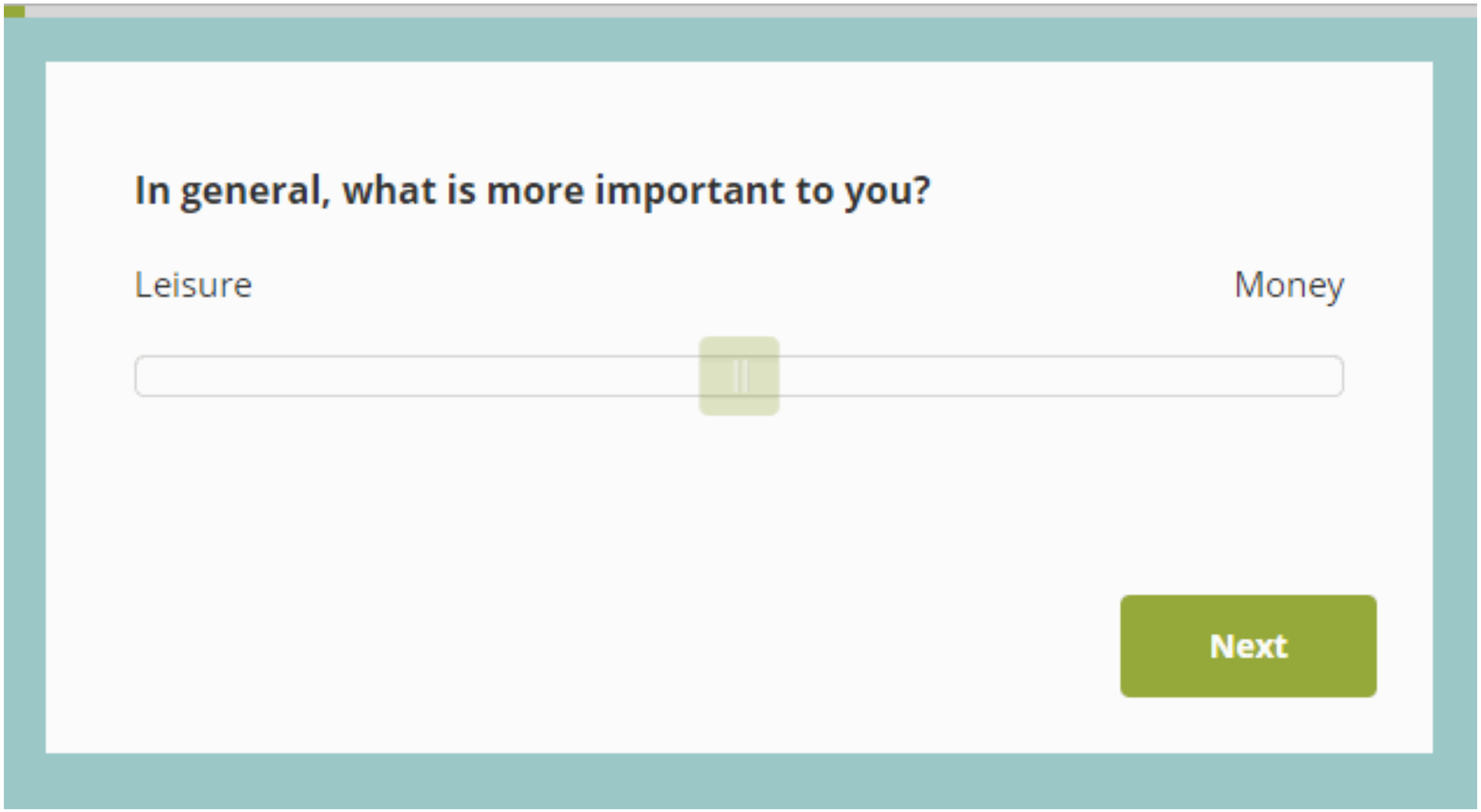
- Introduction
- The zero-one-inflated beta model
- ZOIB regression
- Simulation: Compare ZOIB and t-test performances
- Discussion
- References

Introduction

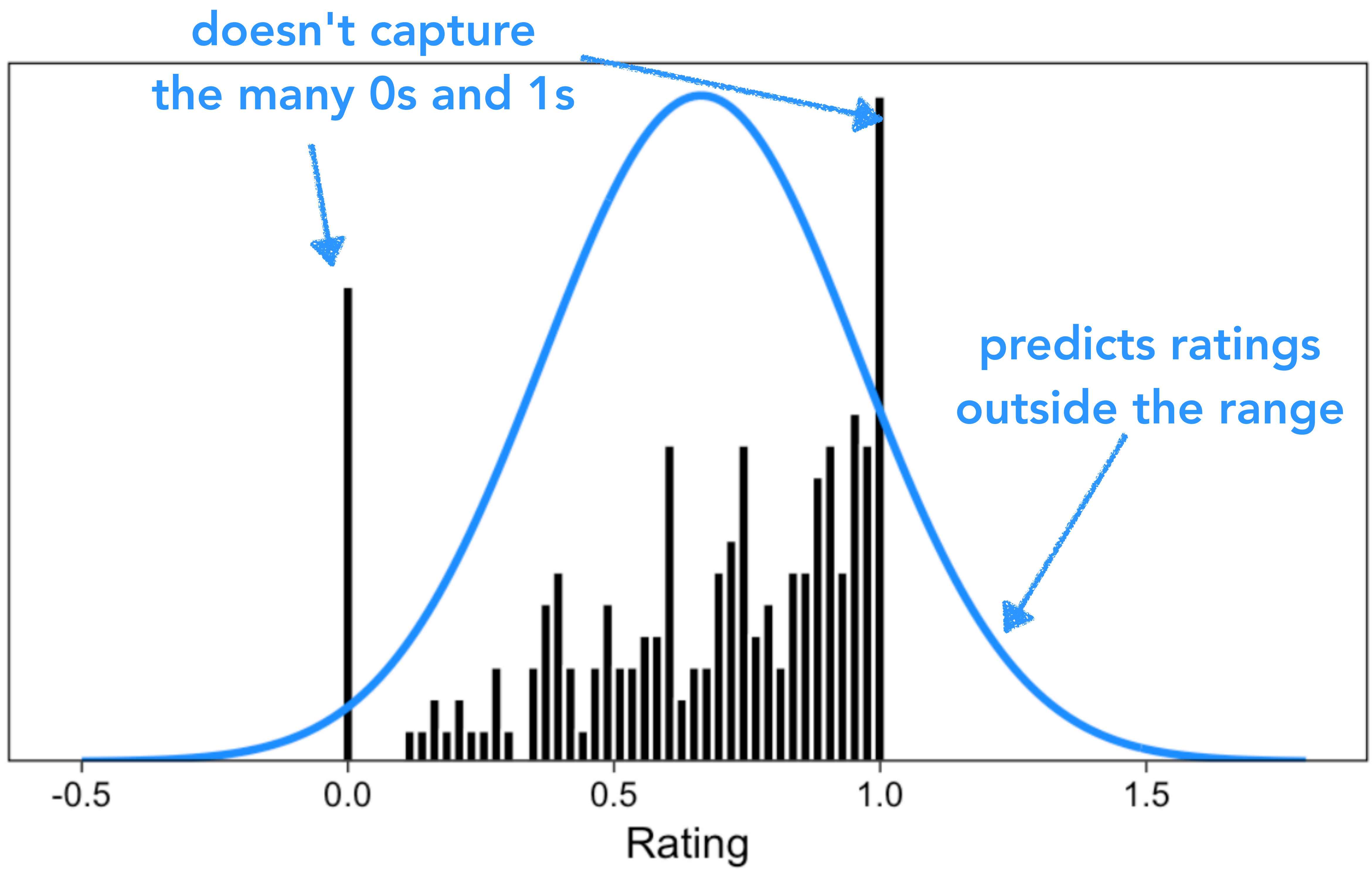
In psychological experiments, subjective responses are often collected using two types of response scales: ordinal and visual analog scales. These scales are unlikely to provide normally distributed data. However, researchers often analyze responses from these scales with models that assume normality of the data.[1]

Ordinal scales, of which binary ratings are a special case, provide ordinal data and are thus better analyzed using ordinal models (Bürkner and Vuorre 2018; Liddell and Kruschke 2018).

https://vuorre.netlify.com/post/2019/02/18/analyze-analog-scale-ratings-with-zero-one-inflated-beta-models/#zoib-regression


In general, what is more important to you?
Leisure
Money
Next

Normality assumption is (almost always) violated



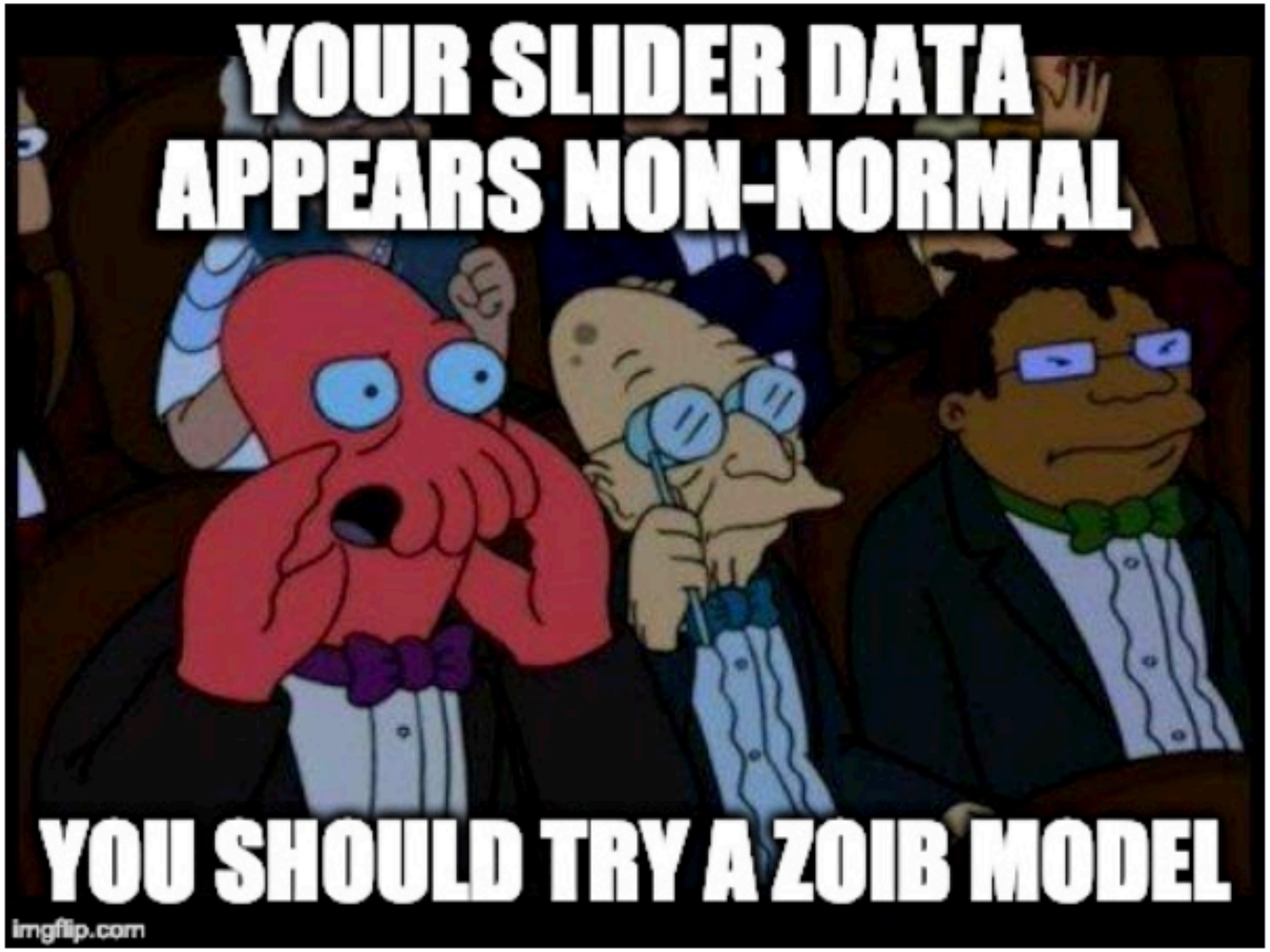


Figure 3: Dr. John A. Zoidberg thinks you should try a ZOIB model on your slider scale data.

From Nour Kteily's talk

or an OIB model?!

People can vary in how human-like they seem. Some people seem highly evolved whereas others seem no different than lower animals. Using the image below, indicate using the sliders how evolved you consider the average member of each group to be:

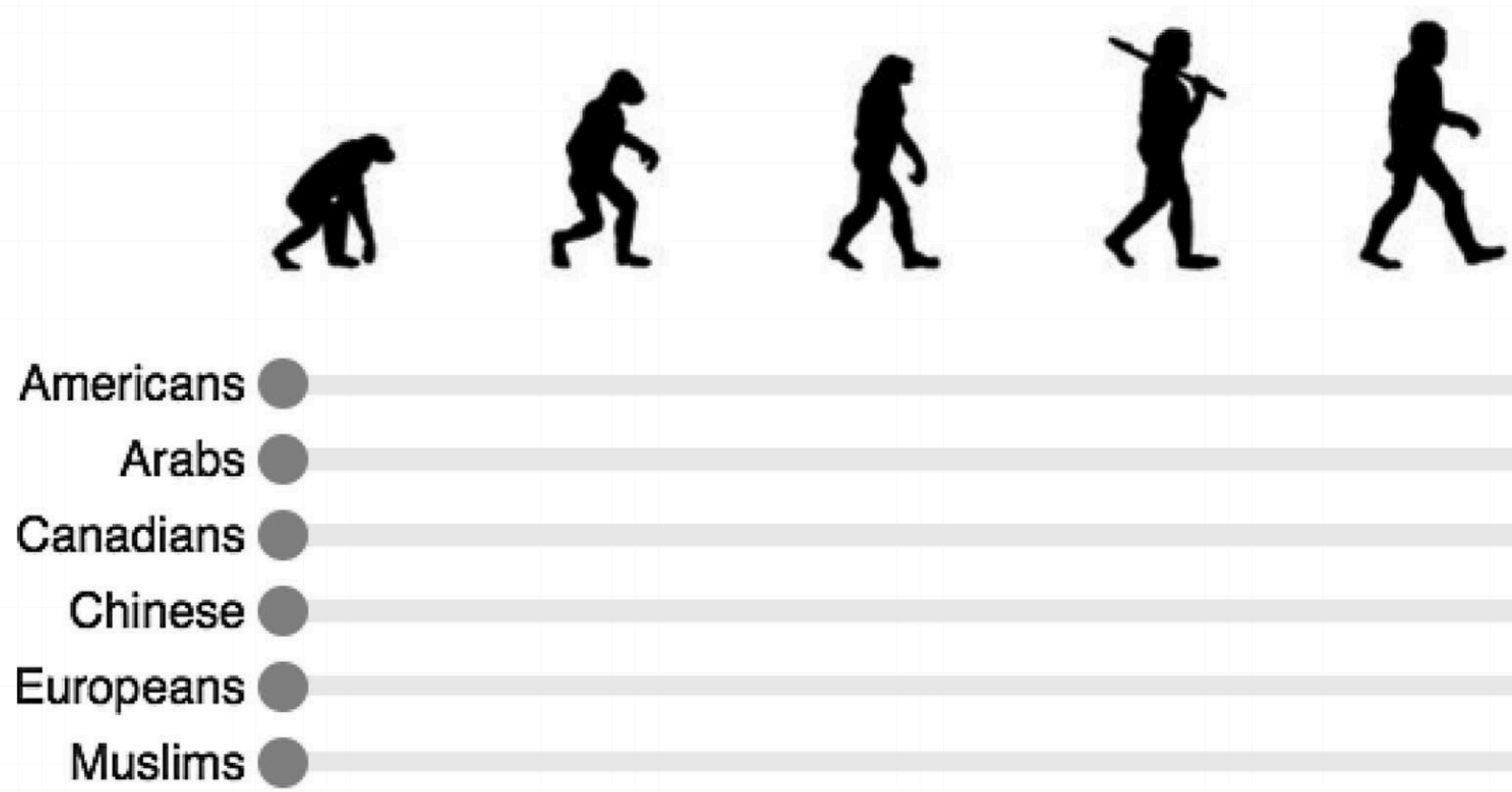
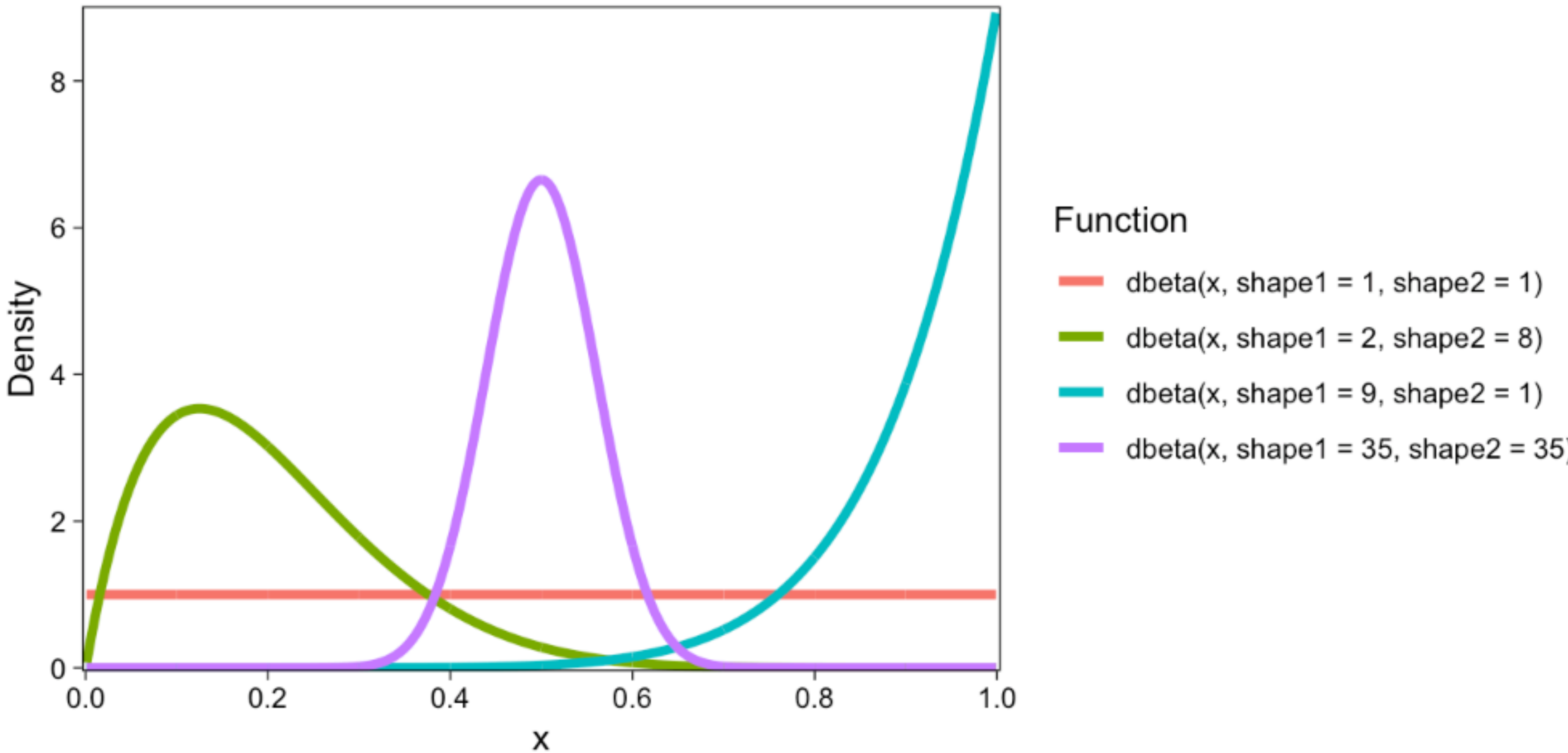


Figure 1. The Ascent measure of blatant dehumanization. Responses were made for each target group using the sliders next to the groups. Target group order was randomized across participants.

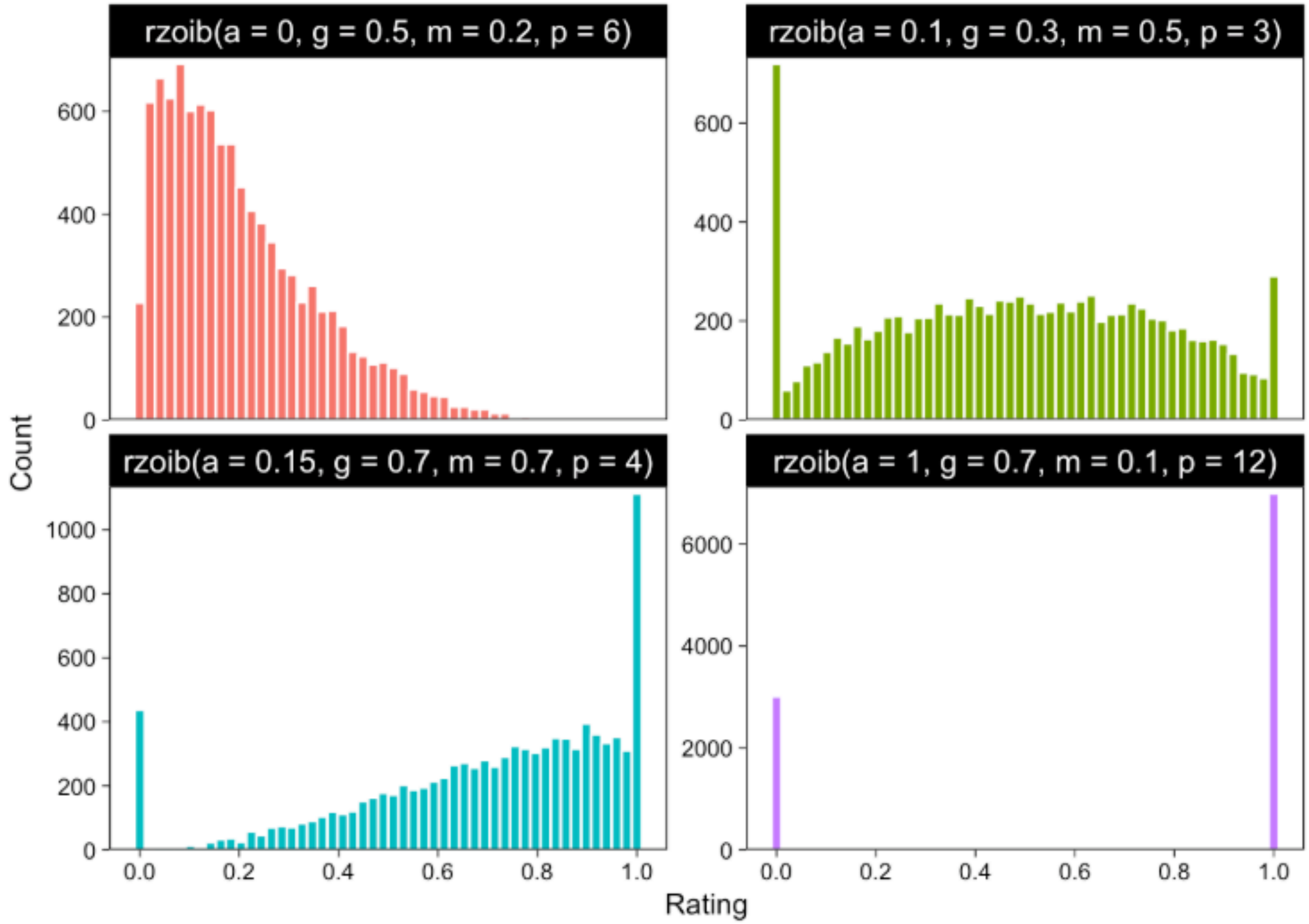
Kteily, N., Bruneau, E., Waytz, A., & Cotterill, S. (2015). The ascent of man: Theoretical and empirical evidence for blatant dehumanization. Journal of Personality and Social Psychology.

Beta distribution



Zero-one inflated **beta** binomial model

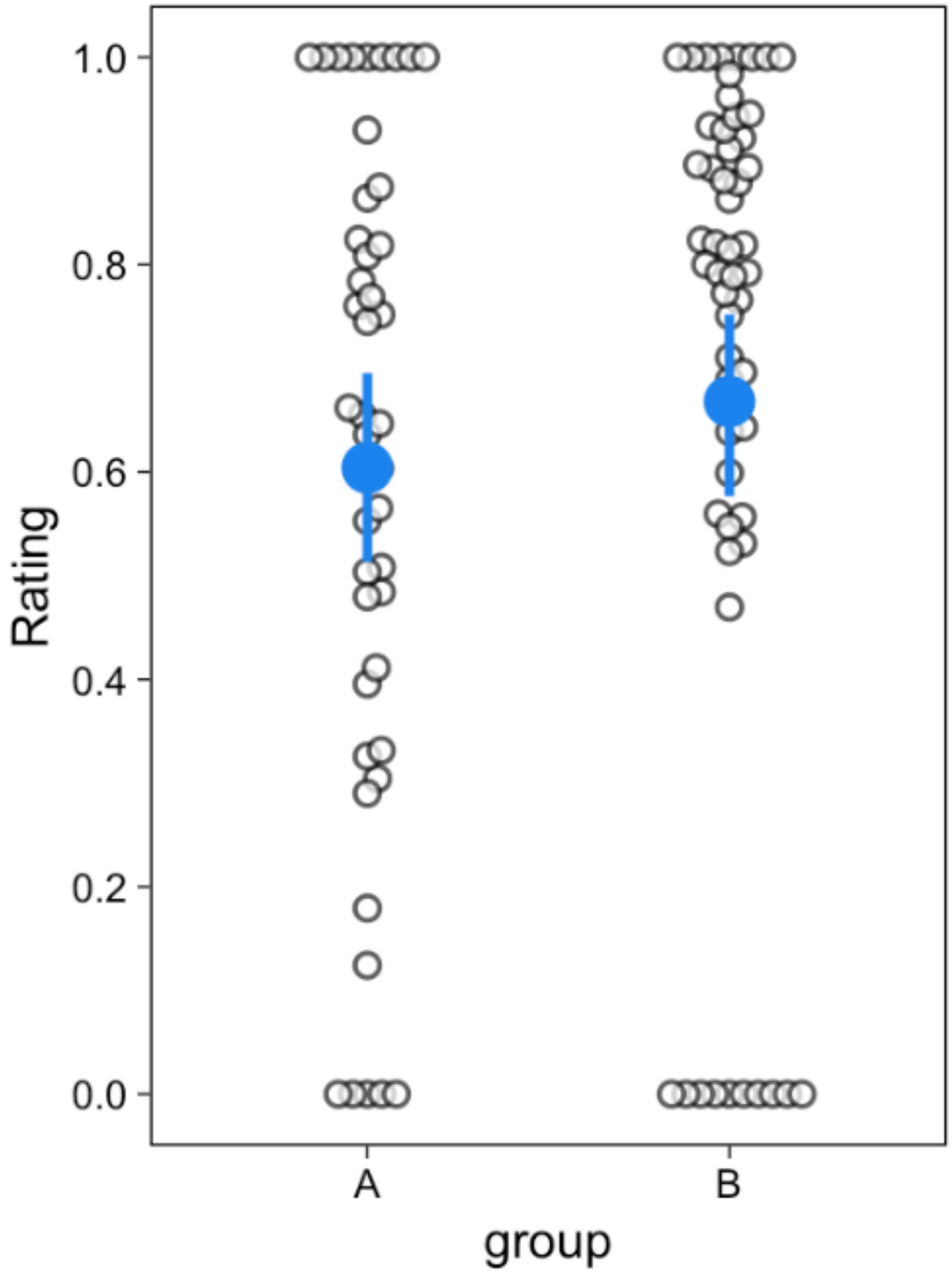
Zero-one inflated beta binomial model



Generative process

Some chance the a person will pick a 0 or 1, if not then response is determined by the beta distribution.

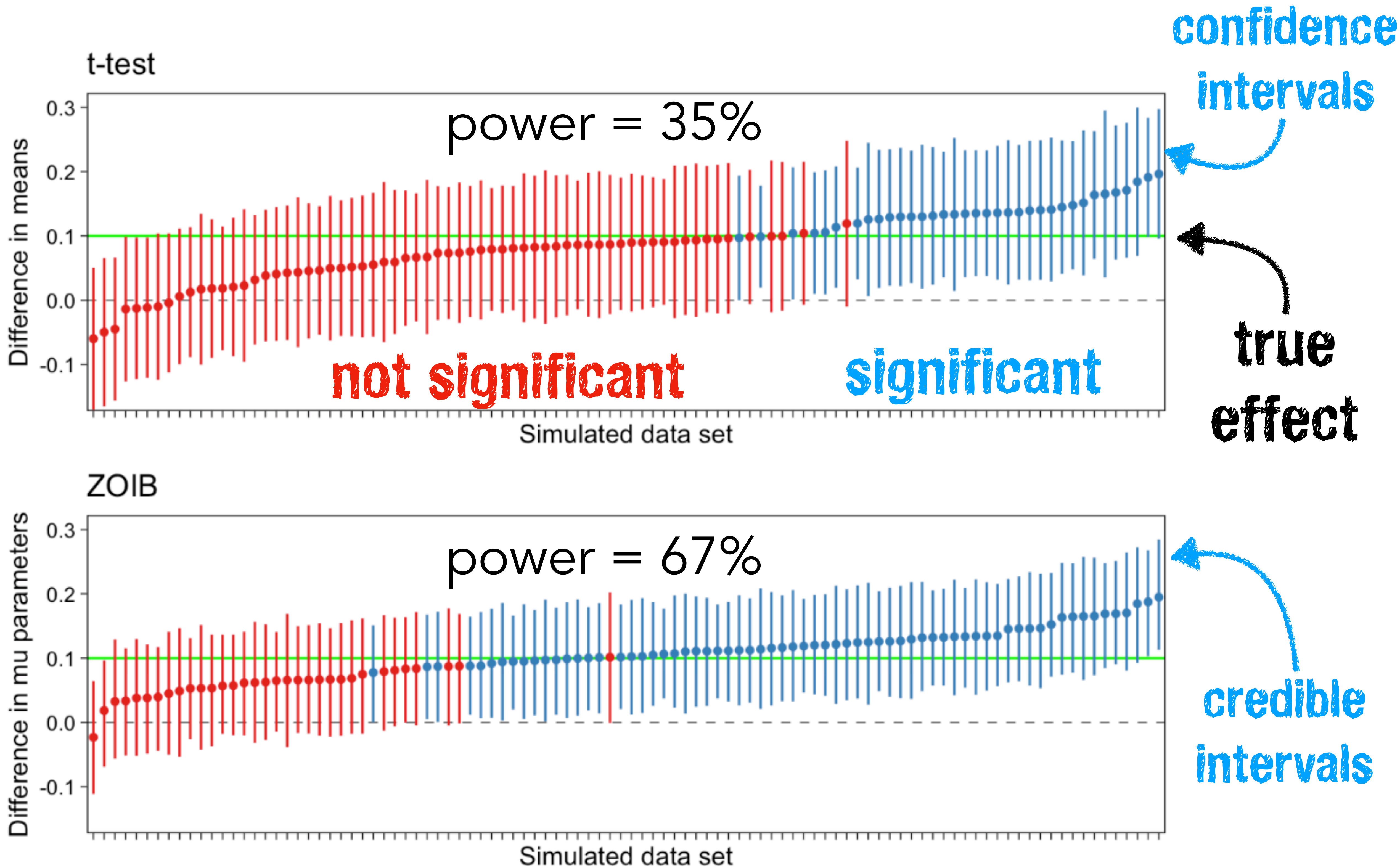
Example data



Fit the ZOIB

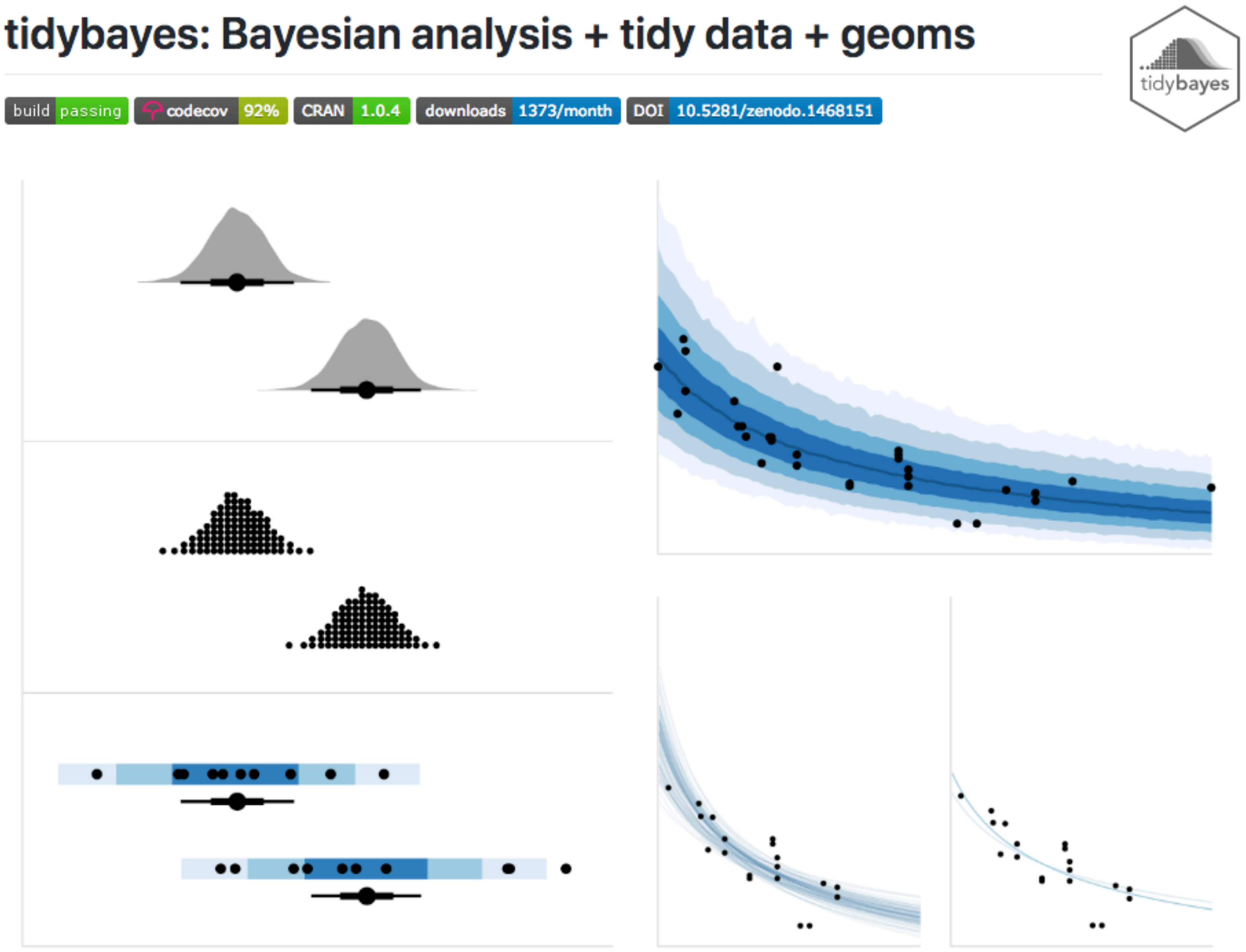
```
zoib_model = bf(
  Rating ~ group,
  phi ~ group,
  zoi ~ group,
  coi ~ group,
  family = zero_one_inflated_beta()
)

fit = brm(
  formula = zoib_model,
  data = dat
)
```

Capturing the data-generating process gives you power



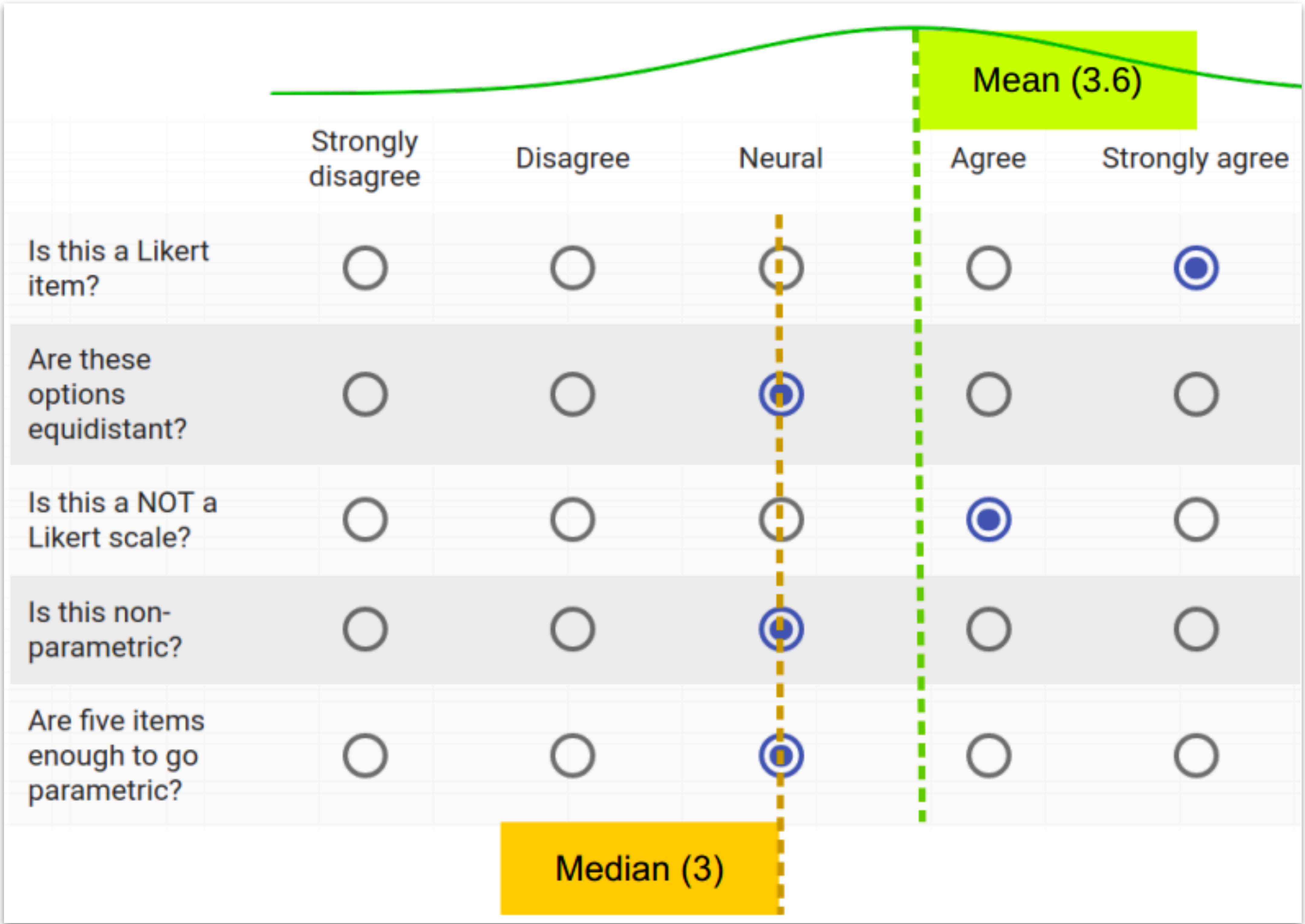
Ordinal logistic regression

Example taken from …



https://mjskay.github.io/tidybayes/articles/tidy-brms.html#ordinal-models

Likert scale data



`mtcars` -- a classic R data set

We want to predict how many cylinders a car's engine has based on the car's efficiency (miles per gallon).

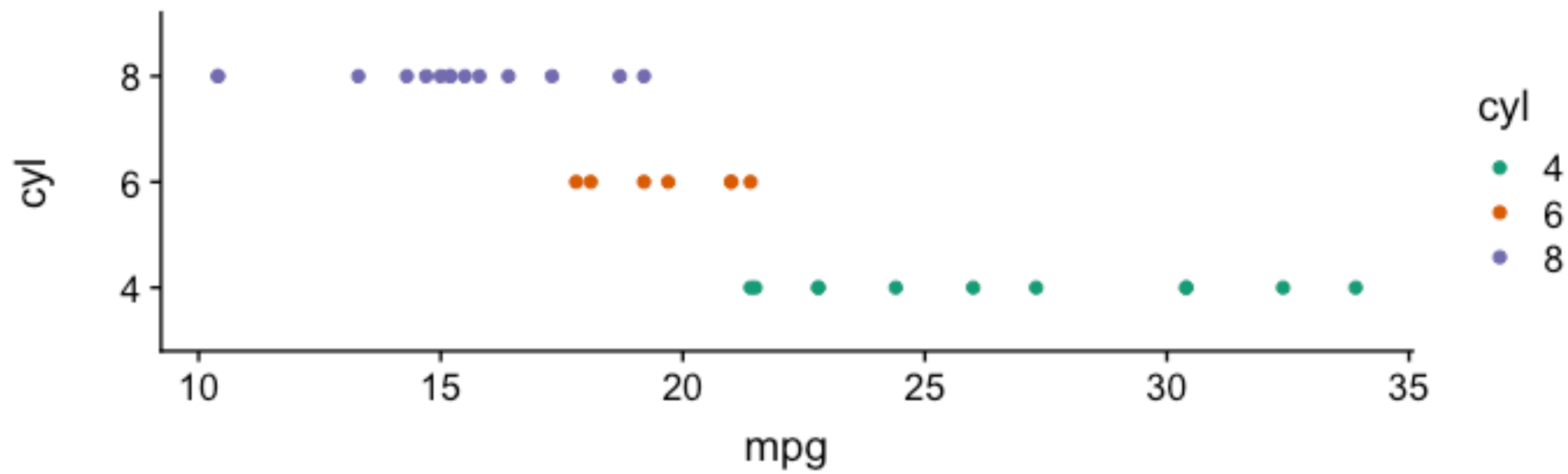
Bayesian ordinal regression

```r
df.cars = mtcars %>%
  mutate(cyl = ordered(cyl)) # creates an ordered factor
```

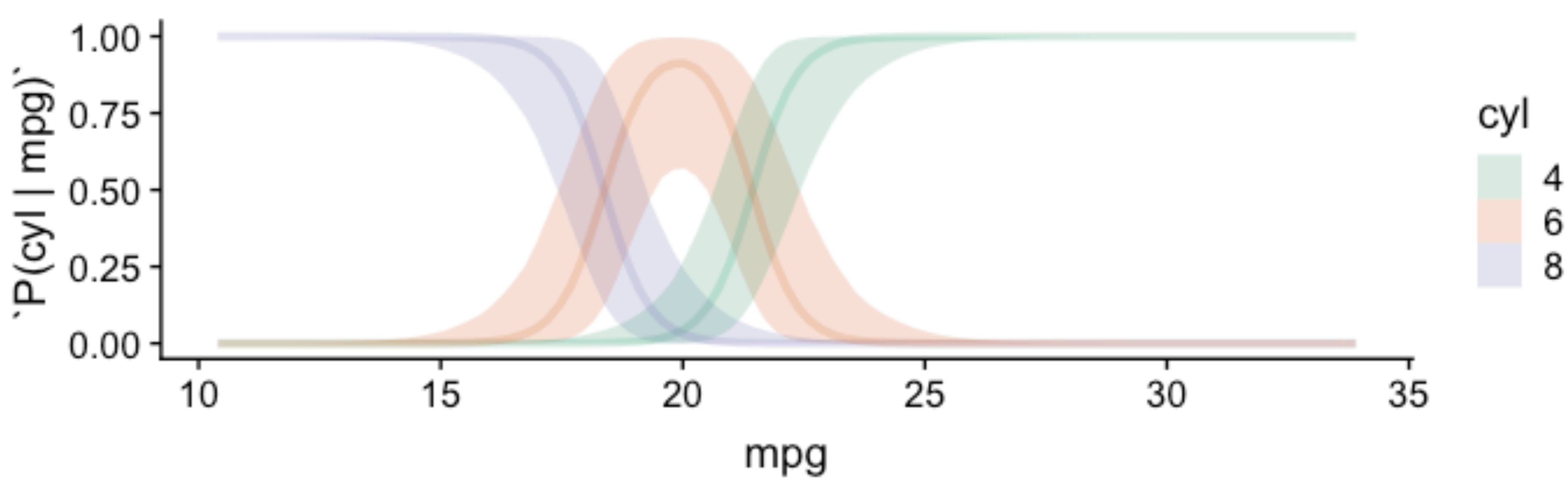
```
'data.frame':	32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : Ord.factor w/ 3 levels "4"<"6"<"8": 2 2 1 2 3 2 3 1 1 2 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

Bayesian ordinal regression

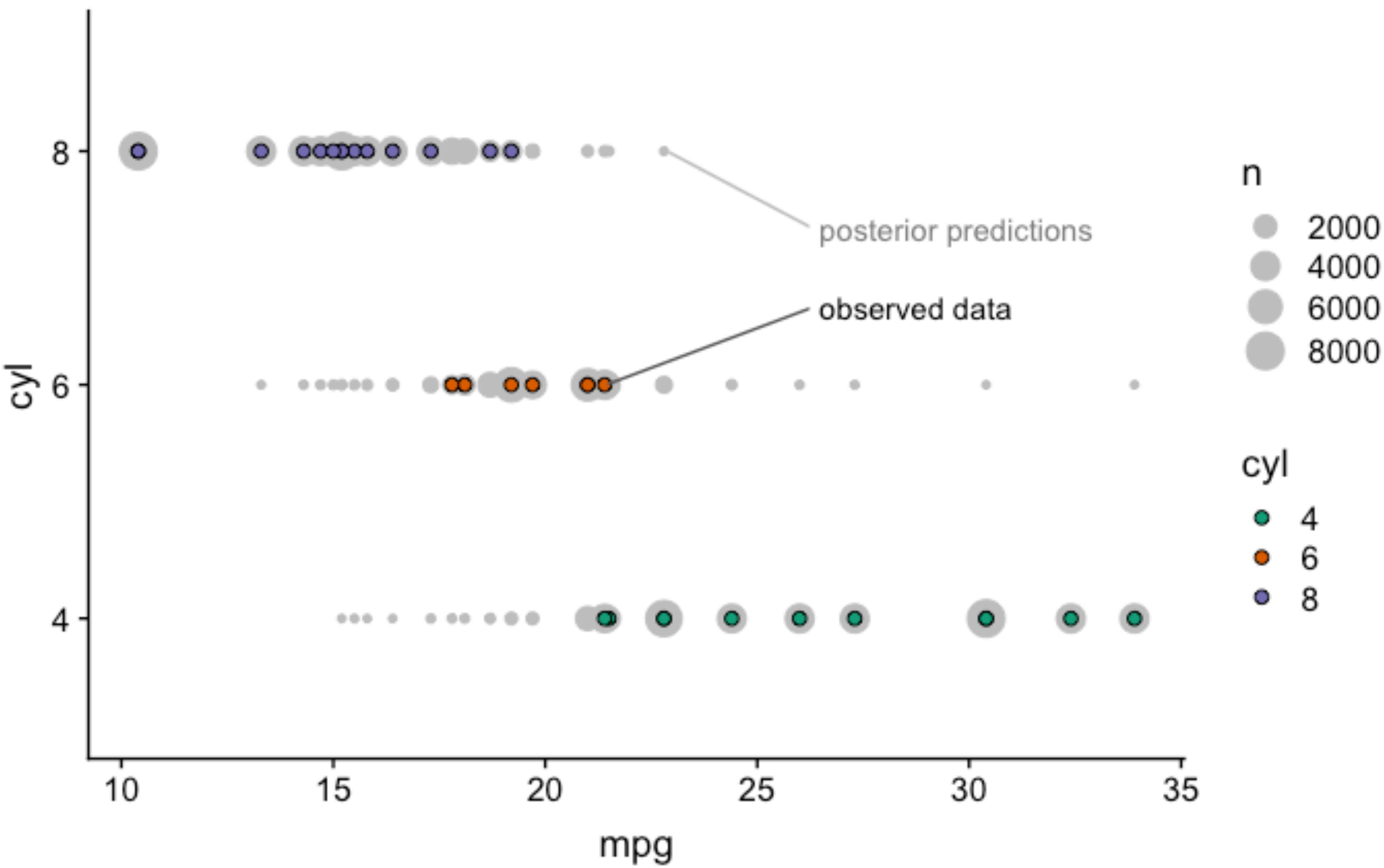
```r
fit.cars = brm(formula = cyl ~ mpg,
               data = df.cars,
               family = "cumulative",
               file = "cars",
               seed = 1)
```



probability of having 4, 6, or 8 cylinders



Posterior predictive check



Summary

- Doing Bayesian data analysis
  - A simple linear regression
  - Measuring uncertainty: Confidence interval vs. credible interval
- Building Bayesian models with `brms`
  - Model evaluation:
    - Visualizing and interpreting results
    - Testing hypotheses
  - Inference evaluation: Did things work out?
- Some cool examples
  - Evidence for null results
  - Dealing with unequal variance
  - Zero-one inflated beta binomial model
  - Ordinal logistic regression

Feedback

How was the pace of today's class?

How happy were you with today's class overall?

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

Thank you!