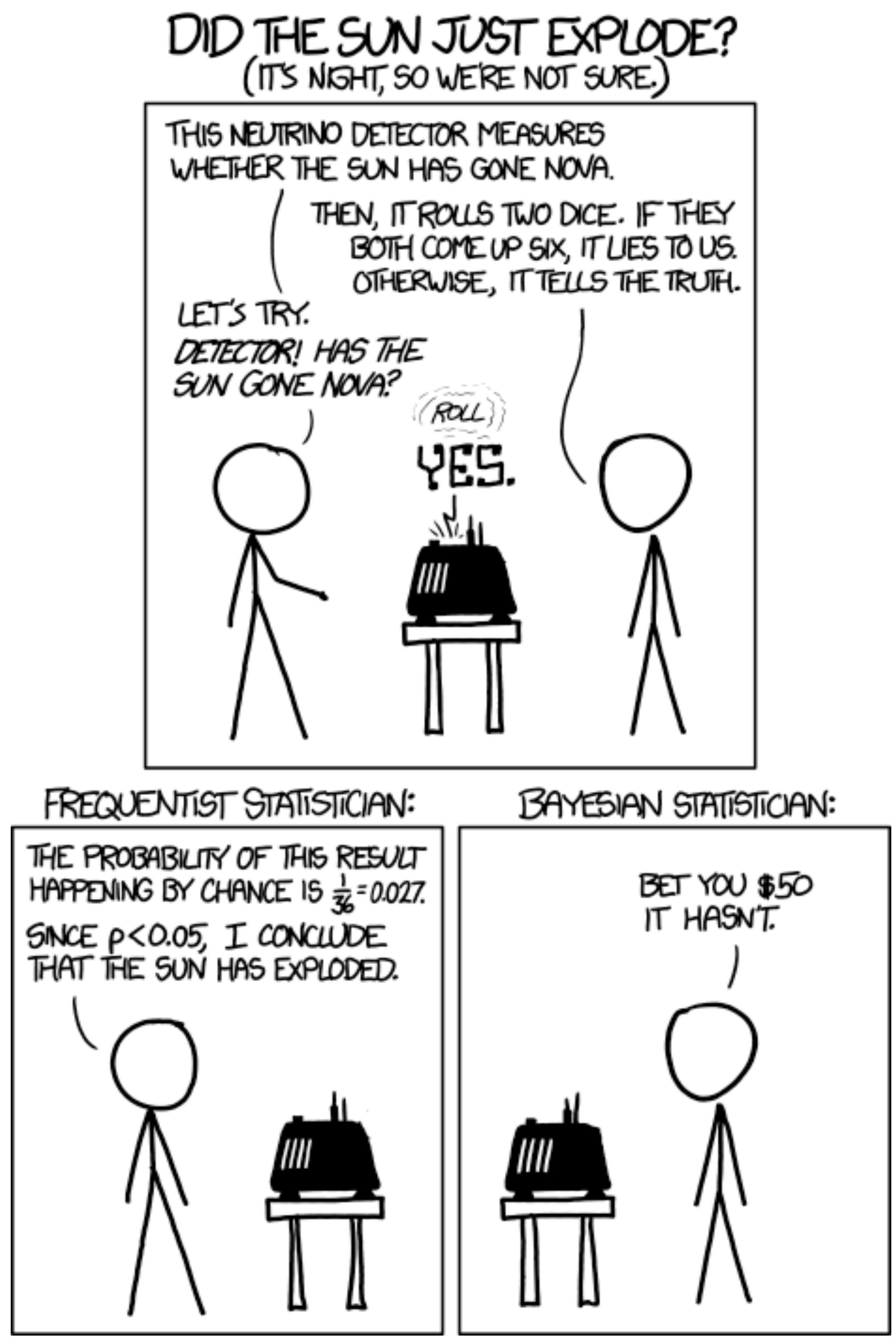
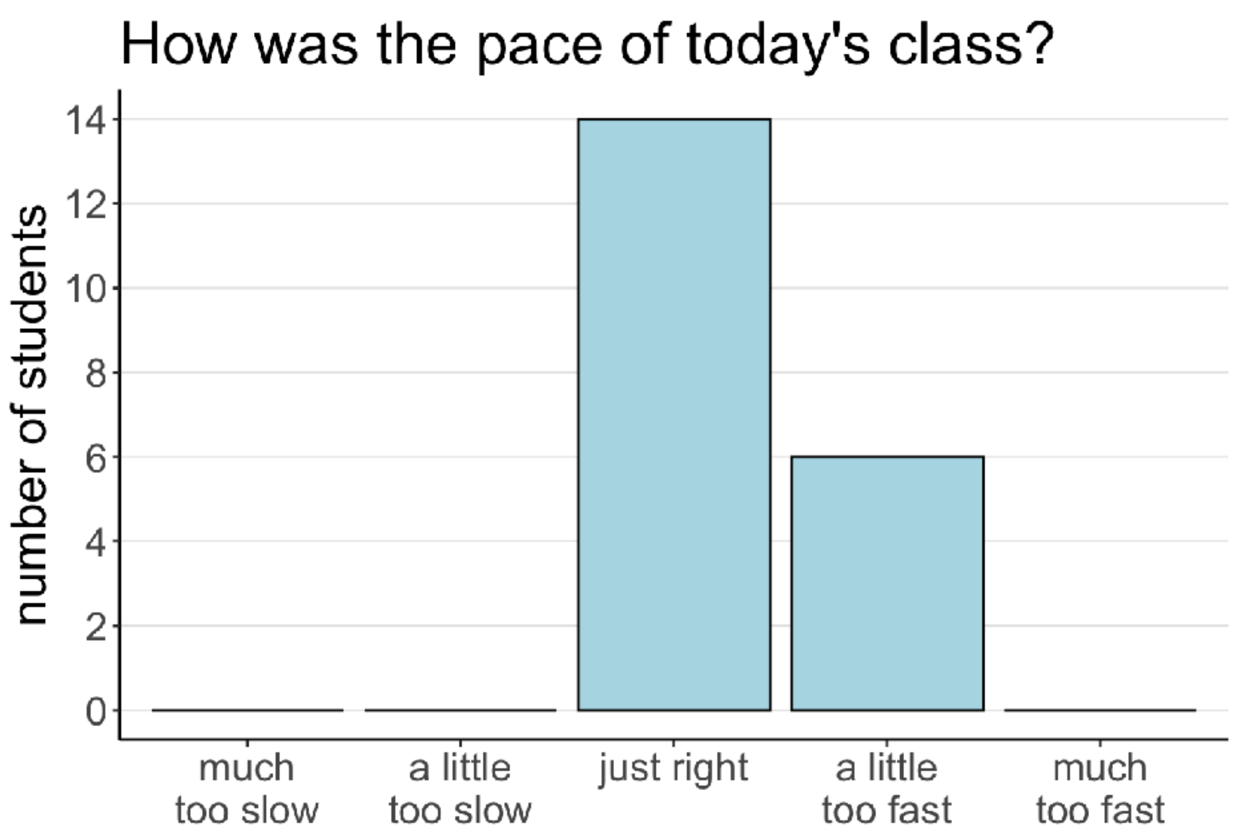
# Bayesian data analysis 1

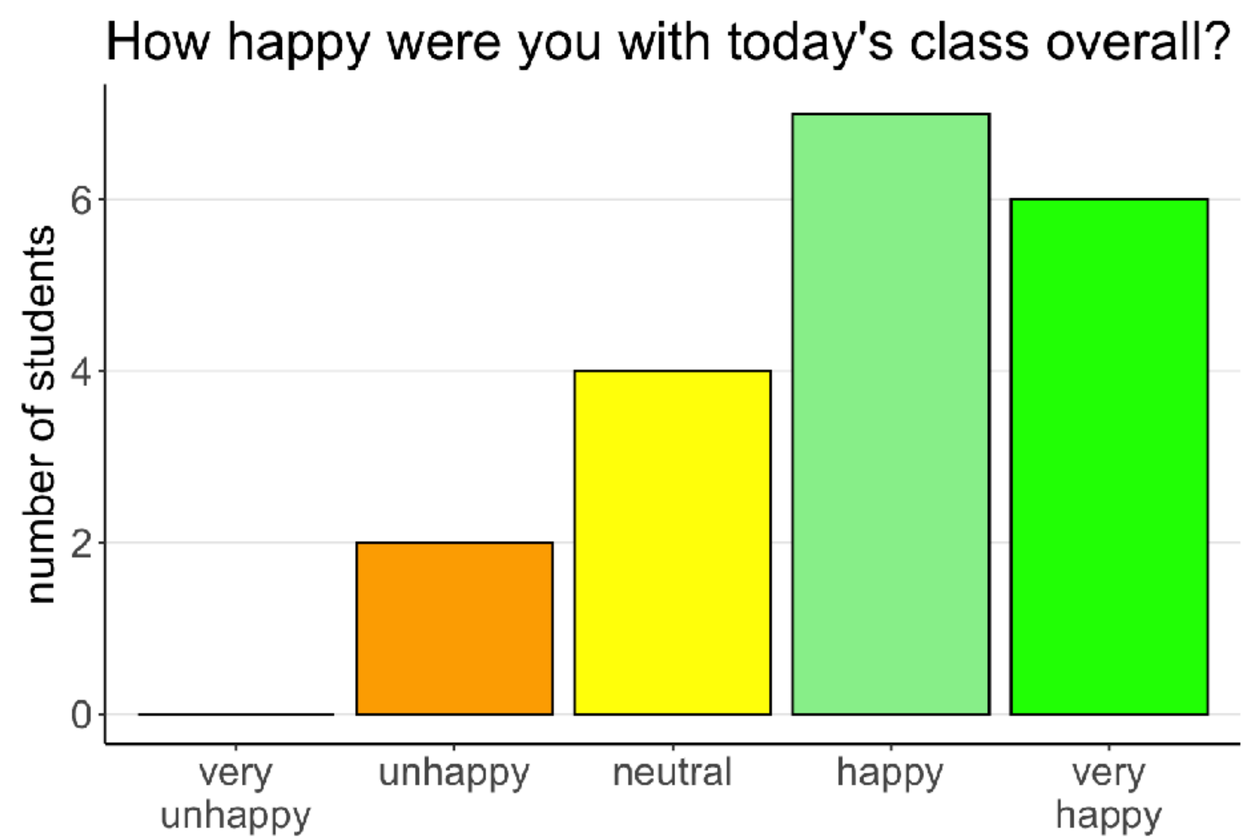



02/28/2020

# Your feedback

# Your feedback



# Feedback

not about class, but the way that you have been announcing dates for things is not ideal. You said the homework would be released on saturday night. When I look at the Canvas folders for the homework assignment, they all say Sunday or Monday. It was late. You moved the due date for the homework back one day because the homework was to be posted one day late. But it was more than that. I think, when I first checked, it said files were uploaded Sunday morning at 1am. For all intents and purposes, that's a homework pushed back from Friday to Sunday morning.

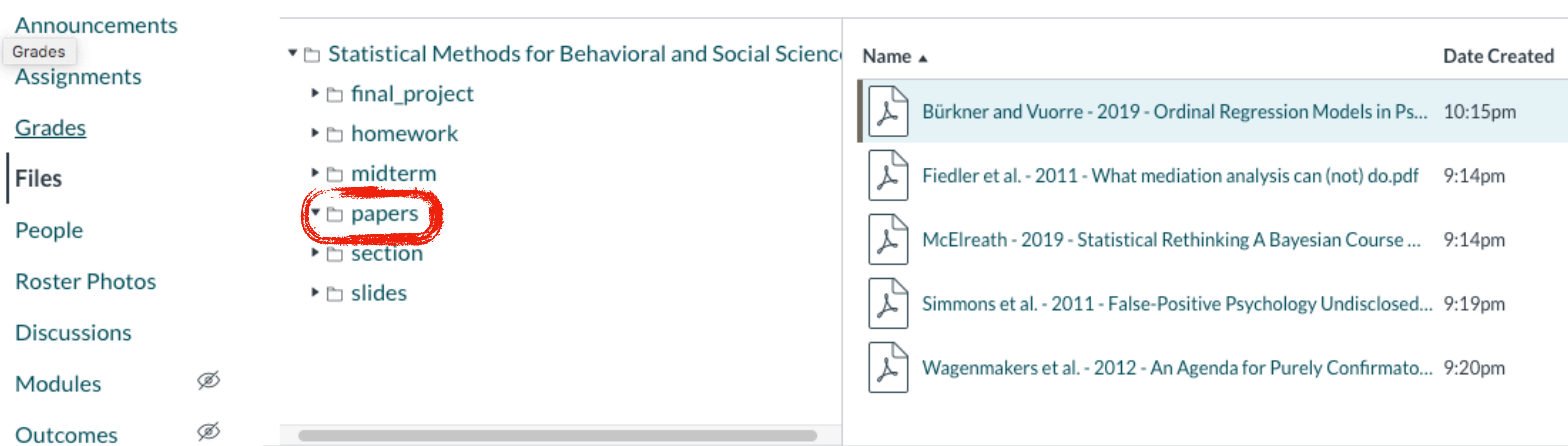
That's fine, I understand life happens. On our homework assignments, we lose points for the style of our coding. **Unfortunately, I will have to deduct 3 points from the teaching team for assignment and grade posting style.** It would be better if you just posted things during the day, or simply adjust your messaging. Some of us care about grades, and if you are planing to post grades at 11:45 on Wednesday night, just tell us they'll be posted Thursday.

I will try my best to give more realistic estimates

# Feedback

One thing in general that I feel like I still don't have a good sense of us how to model likert style data. I don't really have an idea of what a good strategy would be.

**I'll discuss one approach using Bayesian data analysis**




Bürkner & Vuorre (2019) Ordinal Regression Models in Psychology: A Tutorial. Advances in Methods and Practices in Psychological Science

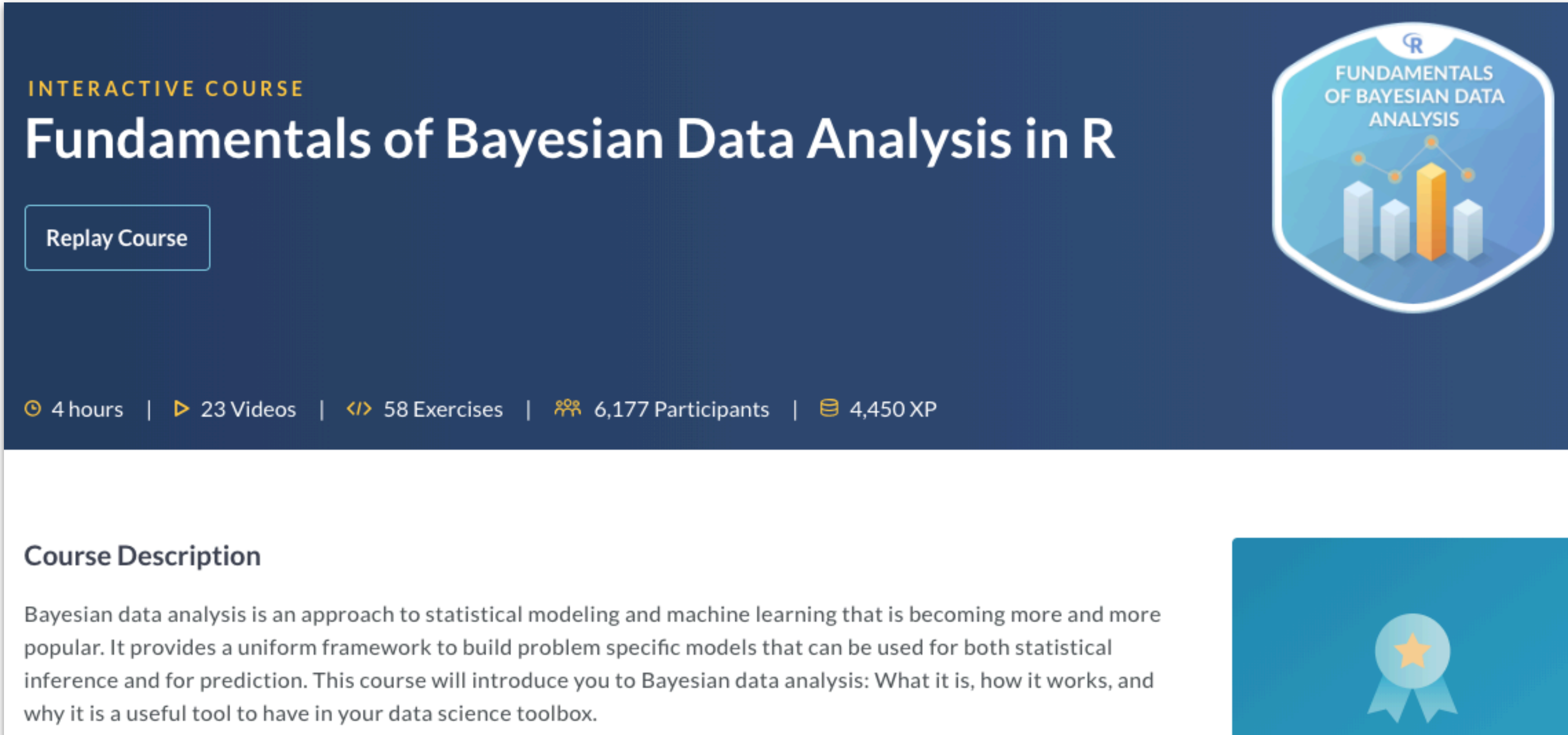
# Logistics

# Homework 6

- Topic: Linear mixed effects models
- will be released tonight (before midnight)
- will be due **Thursday, March 5th, at 8pm**

# Datacamp course

recommended!!

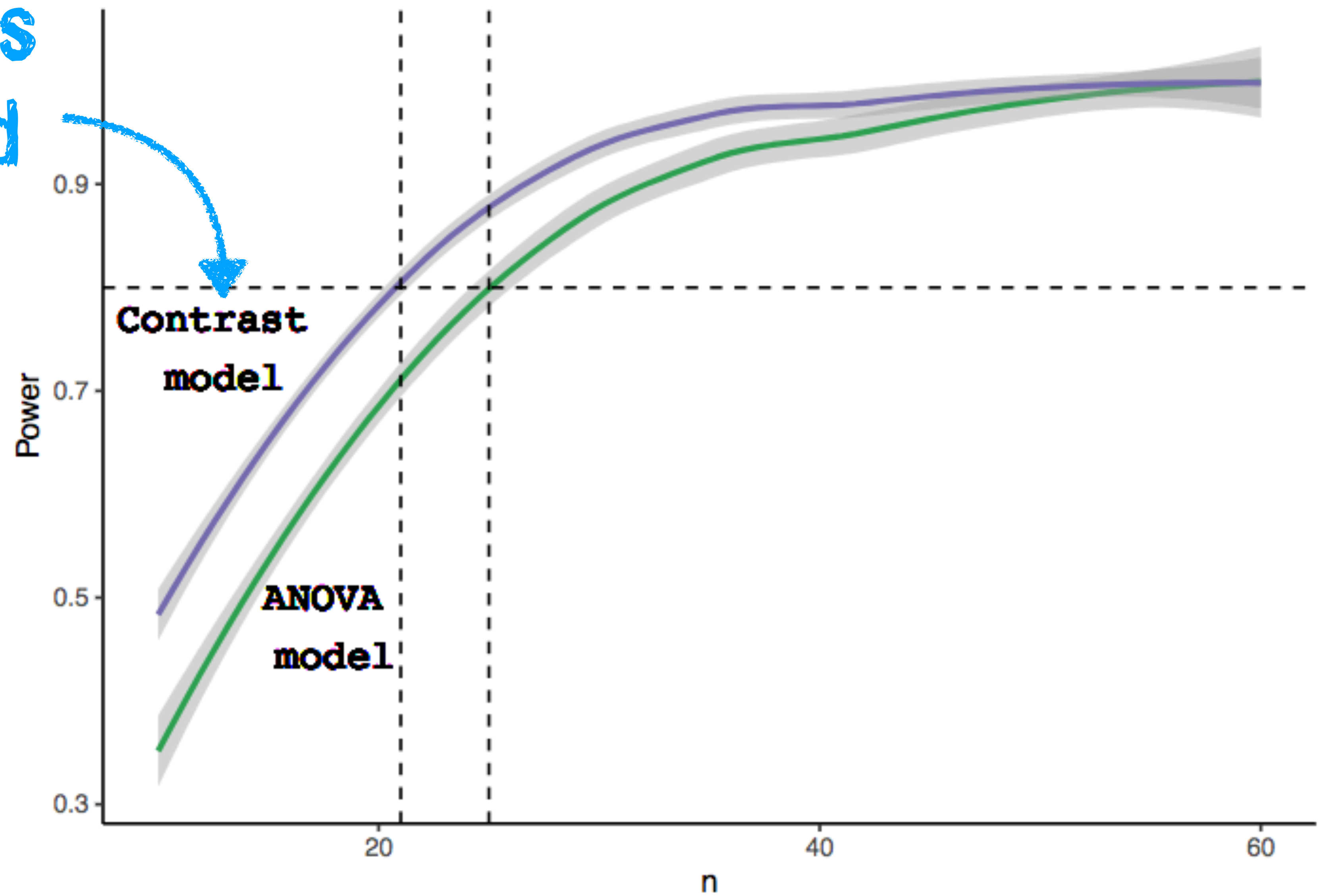



https://www.datacamp.com/courses/fundamentals-of-bayesian-data-analysis-in-r

# Things that came up

# Text in `ggplot()`

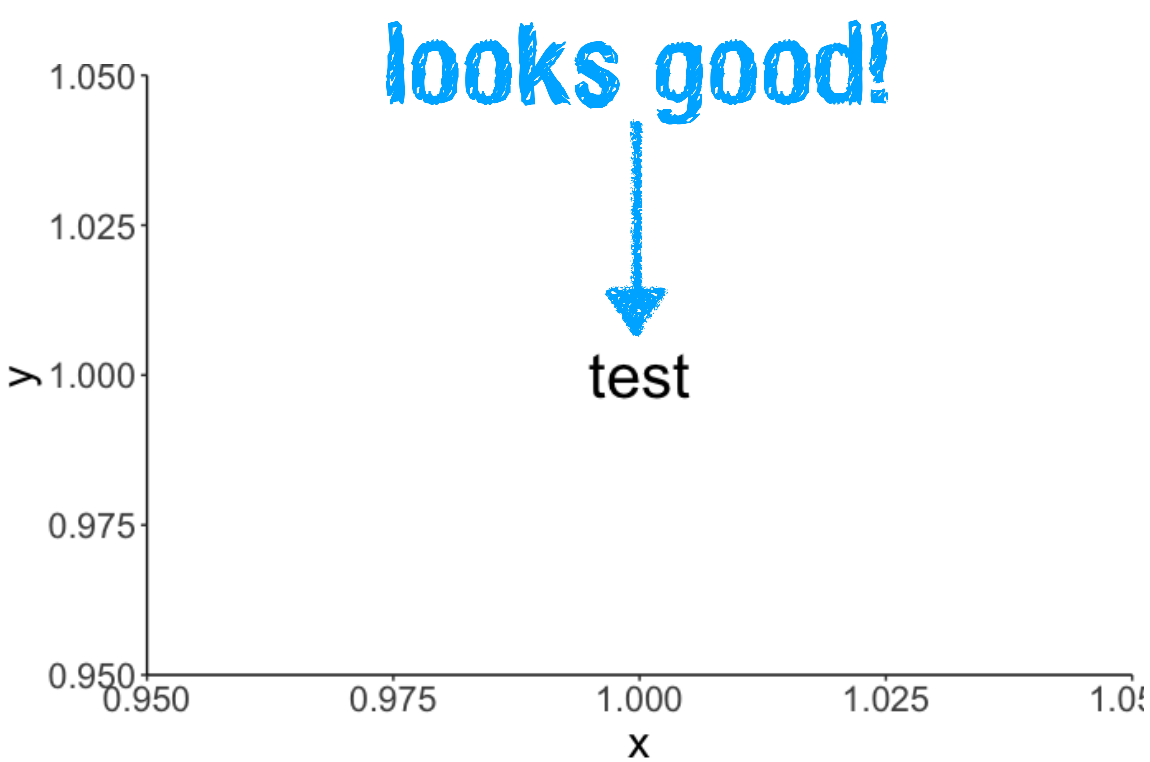
text looks pixelated



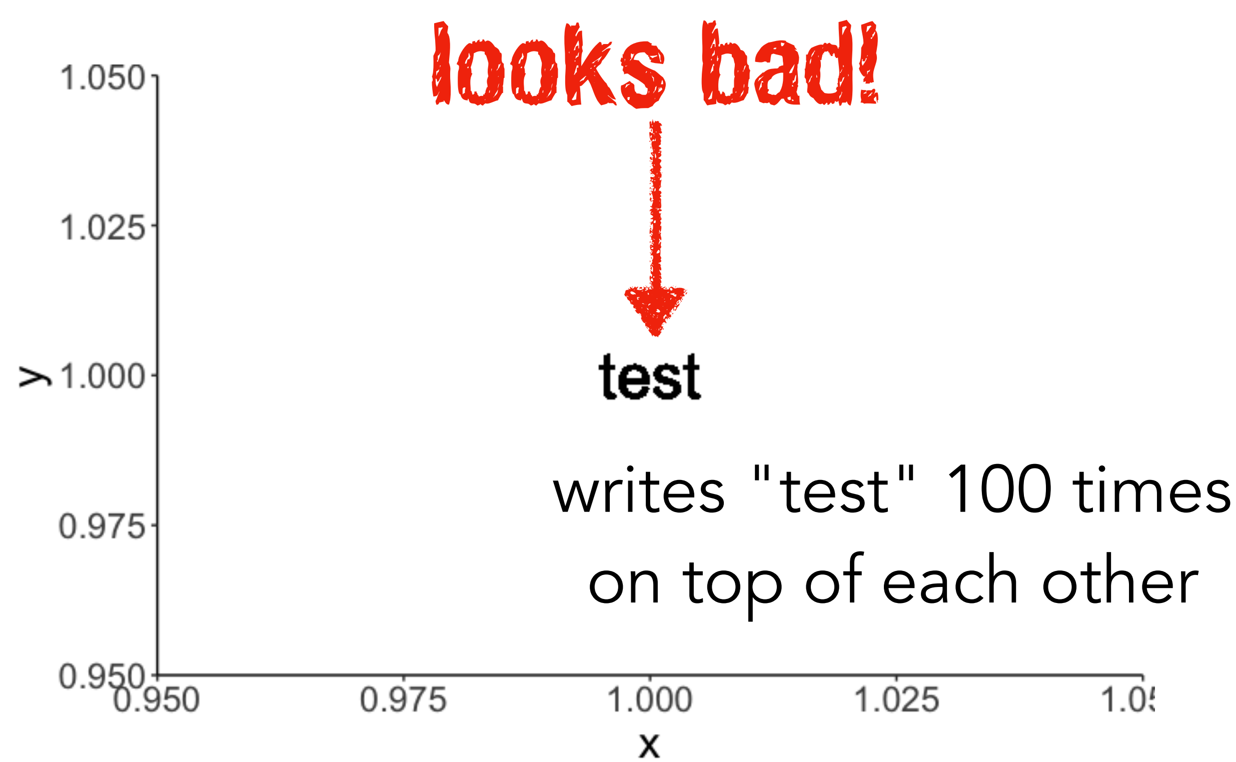
# Text in `ggplot()`

```r
df.plot = tibble(x = rep(1, n),
                 y = rep(1, n),
                 text = "test")

ggplot(data = df.plot,
       mapping = aes(x = x,
                     y = y)) +
  geom_text(aes(label = text),
            size = 10)
```

n = 1



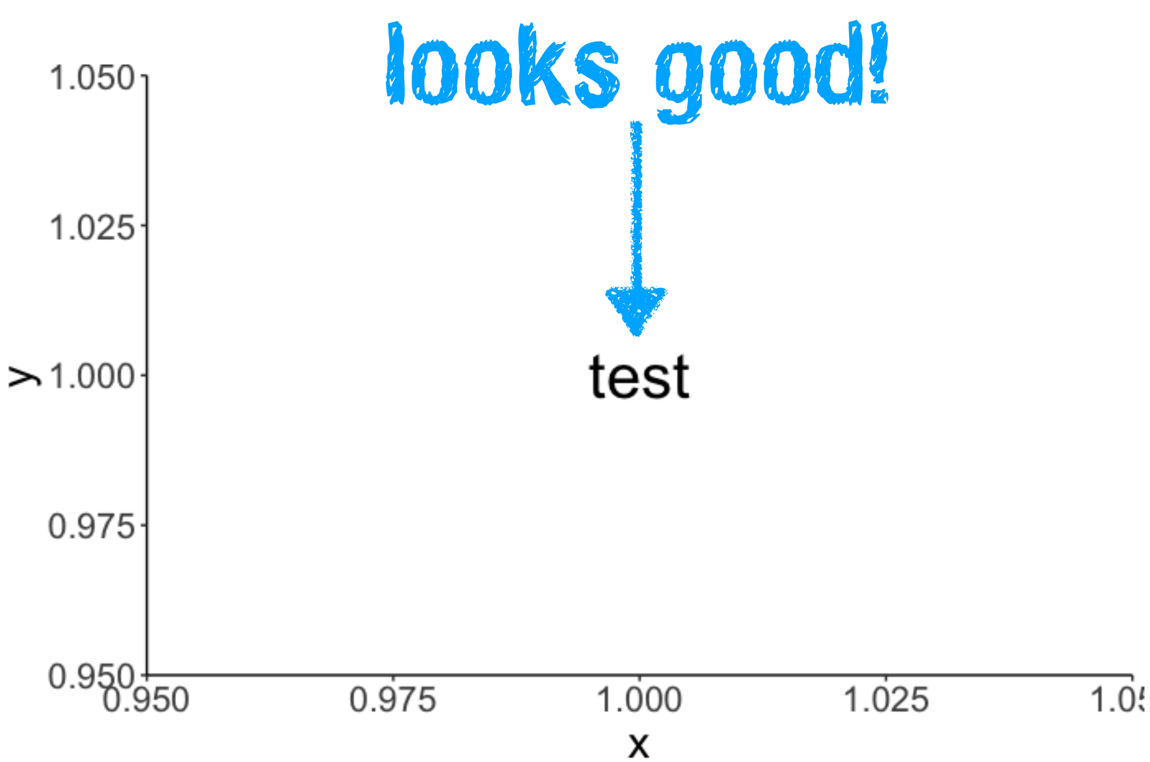

n = 100



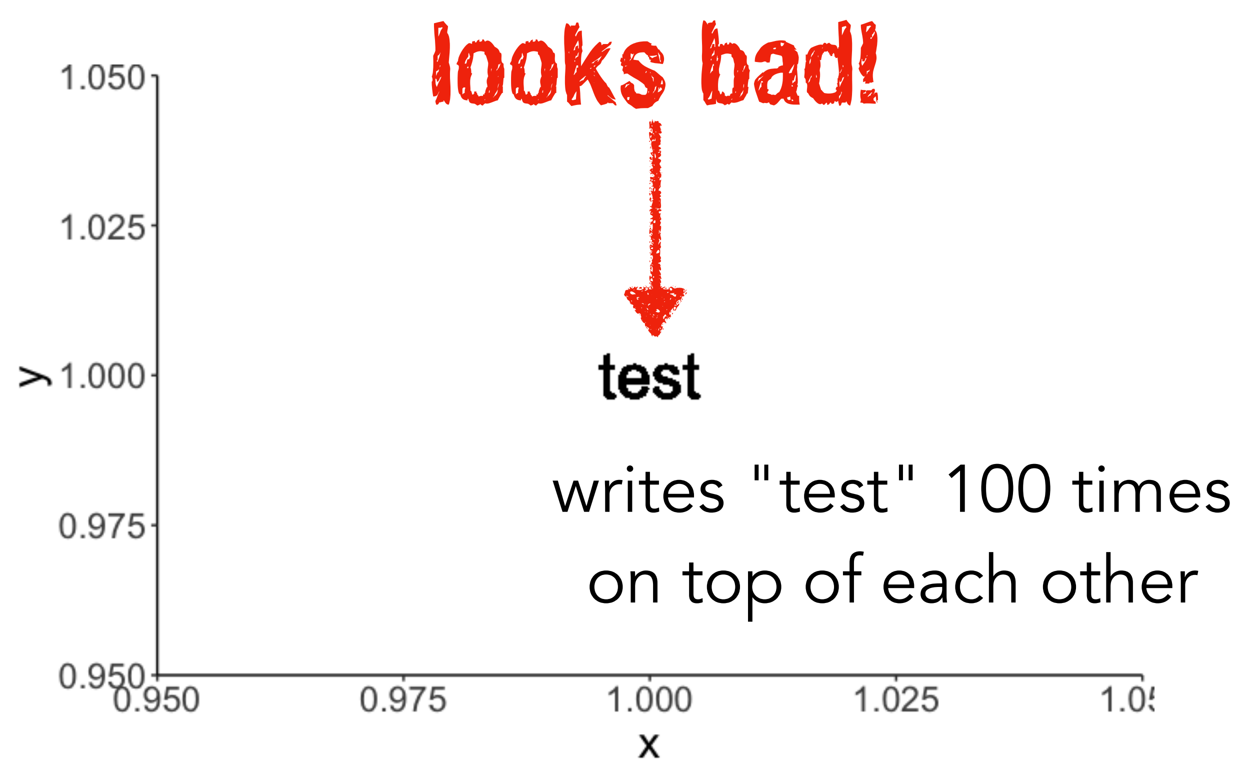
# Text in `ggplot()`

```r
df.plot = tibble(x = rep(1, n),
                 y = rep(1, n),
                 text = "test")

ggplot(data = df.plot,
       mapping = aes(x = x,
                     y = y)) +
  geom_text(aes(label = text),
            size = 10)
```

`n = 1`




`n = 100`



# Text in `ggplot()`

use `annotate()` for single text labels

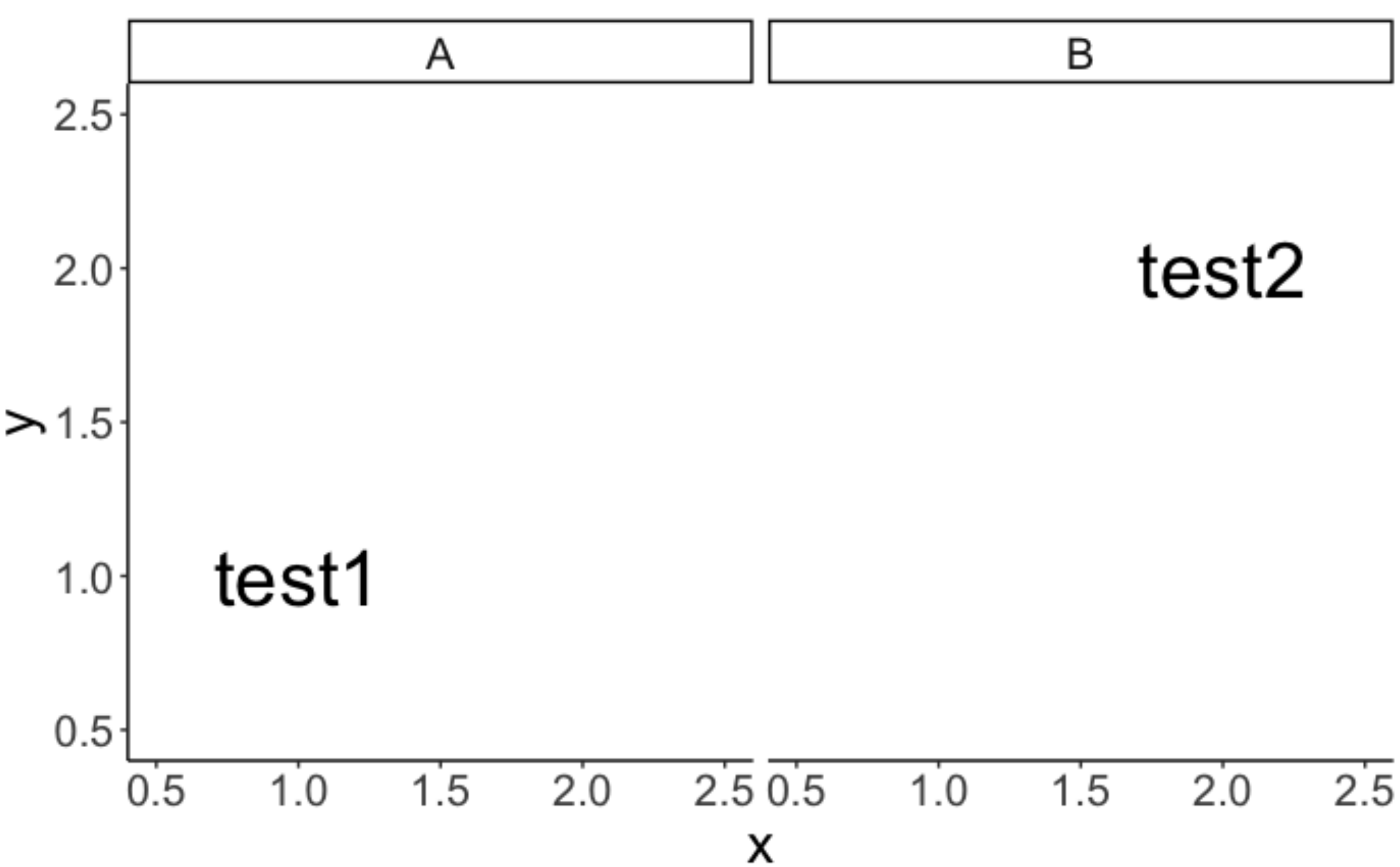
```r
ggplot(data = df.plot,
       mapping = aes(x = x,
                     y = y)) +
  annotate(geom = "text",
           x = 1,
           y = 1,
           label = "test",
           size = 10)
```

# Text in `ggplot()`

# When to use `ggtext()`?

```r
df.plot = tibble(x = c(1, 2),
                 y = c(1, 2),
                 panel = c("A", "B"),
                 text = c("test1", "test2"))

ggplot(data = df.plot,
       mapping = aes(x = x,
                     y = y)) +
  geom_text(aes(label = text),
            size = 10) +
  facet_grid(cols = vars(panel)) +
  coord_cartesian(xlim = c(0.5, 2.5),
                  ylim = c(0.5, 2.5))
```

| x | y | panel | text |
|---|---|---|---|
| 1 | 1 | A | test1 |
| 2 | 2 | B | test2 |

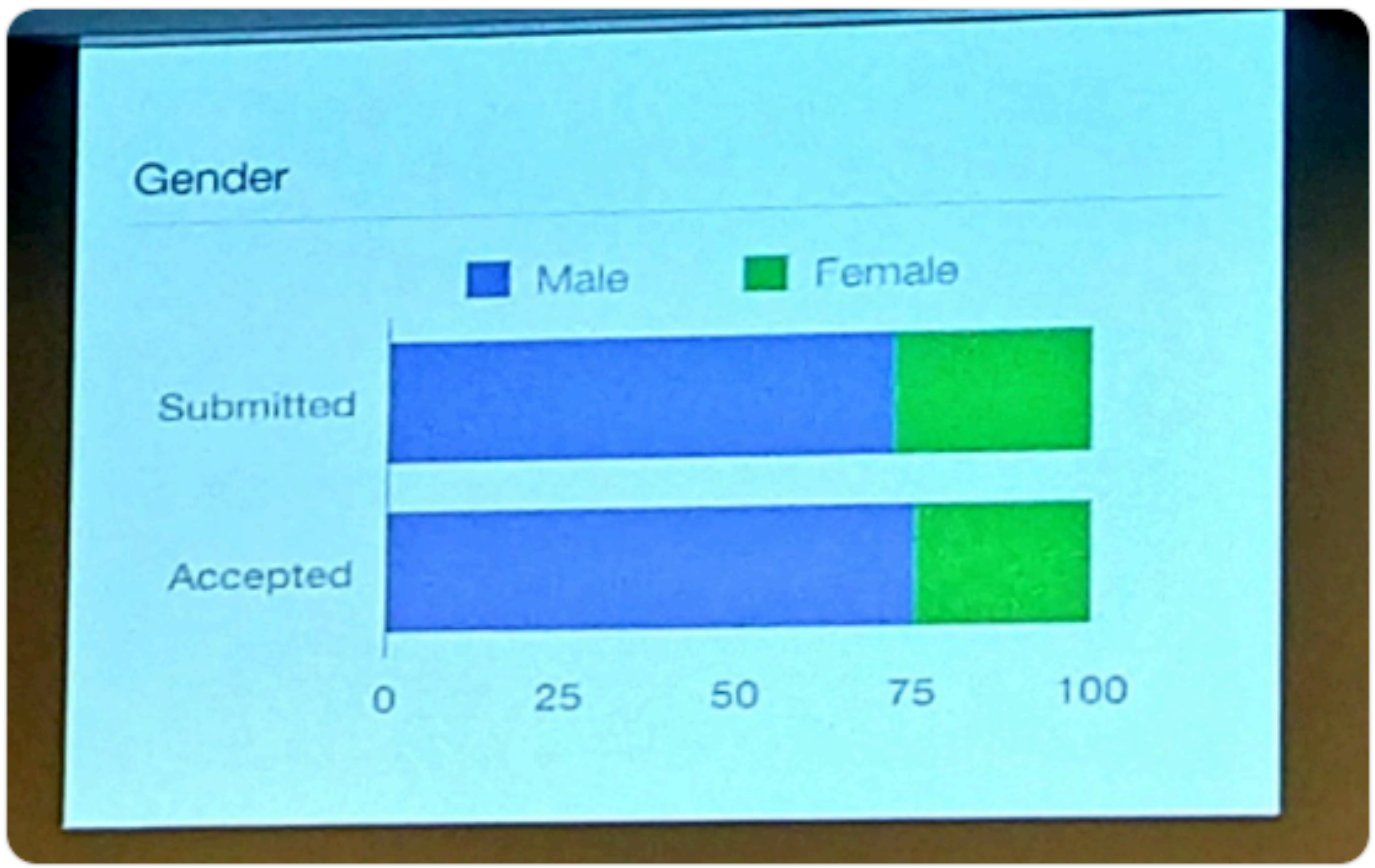


# Bias in conference admission?

**Adam J Calhoun**
@neuroecology

Follow

Accepted and submitted abstracts by gender roughly the same at #cosyne19 cc @neimarkgeffen @TrackingActions




10:29 AM - 28 Feb 2019

# Bias in conference admission?

**Adam J Calhoun** @neuroecology · Feb 28

Accepted and submitted abstracts by gender roughly the same at #cosyne19 cc @neimarkgeffen @TrackingActions


4 10

**Yael Niv**
@yael_niv

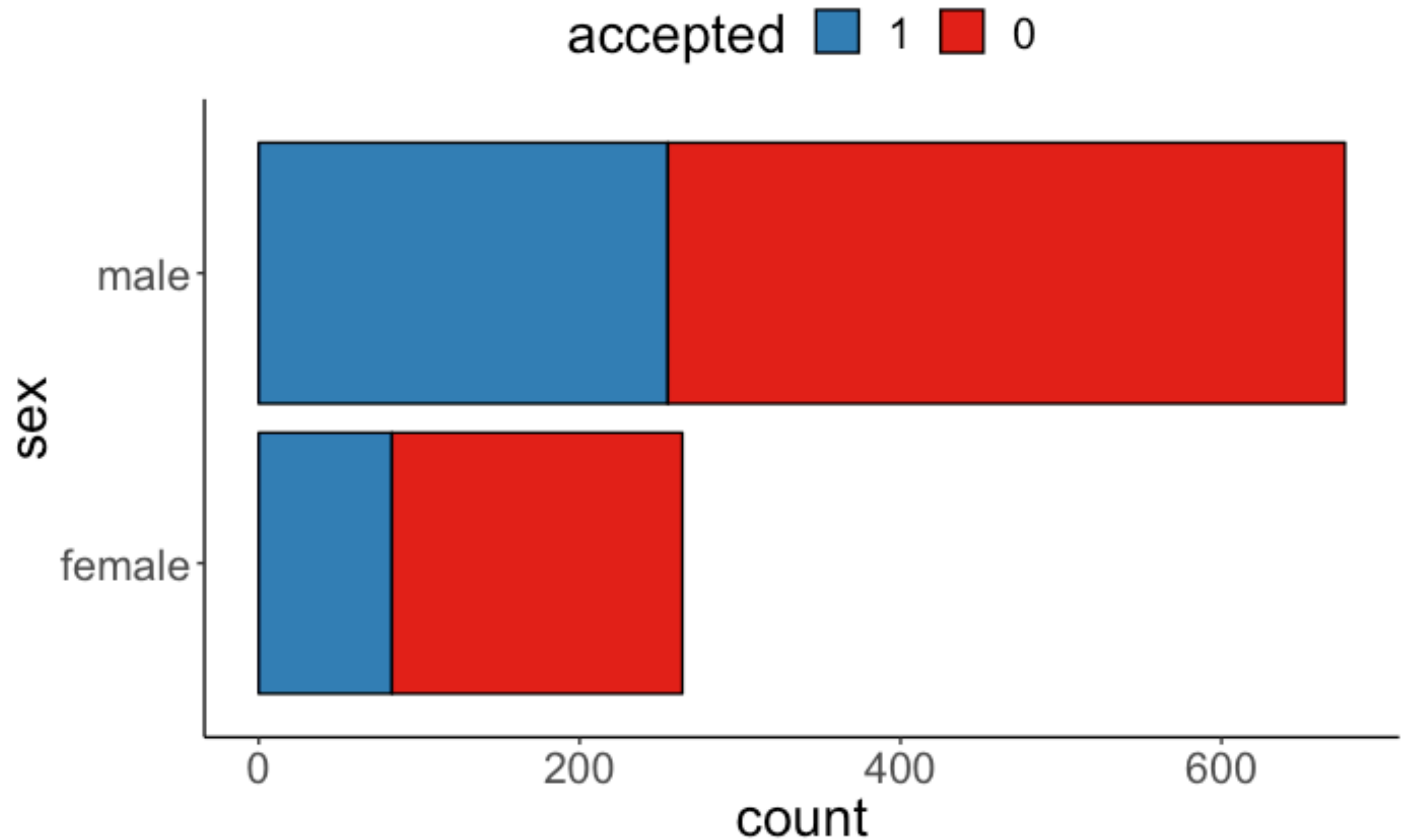
Follow

Replying to @neuroecology @neimarkgeffen @TrackingActions

Doesn't look the same to me...

10:55 AM - 2 Mar 2019

# Bias in conference admission?




**different representation of the data**

# Bias in conference admission?

```r
# logistic regression
fit.glm = glm(formula = accepted ~ 1 + sex,
              family = "binomial",
              data = df.conference)

# model summary
fit.glm %>%
  summary()
```

```
Call:
glm(formula = accepted ~ 1 + sex, family = "binomial", data = df.conference)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.9723  -0.9723  -0.8689   1.3974   1.5213

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.7797     0.1326  -5.881 4.07e-09 ***
sexmale       0.2759     0.1545   1.786   0.0741 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1228.9  on 940  degrees of freedom
Residual deviance: 1225.6  on 939  degrees of freedom
AIC: 1229.6

Number of Fisher Scoring iterations: 4
```

# Bias in conference admission?

**Megan Carey** @meganinlisbon · Mar 2

I presented the math for this at the #cosyne19 diversity lunch today.

Success rates for first authors with known gender:
Female: 83/264 accepted = 31.4%
Male: 255/677 accepted = 37.7%

37.7/31.4 = a 20% higher success rate for men

**Adam J Calhoun** @neuroecology
Accepted and submitted abstracts by gender roughly the same at #cosyne19 cc @neimarkgeffen @TrackingActions
Show this thread

9 37 83

**Mehrdad Jazayeri**
@mjaztwit

Following

Replying to @meganinlisbon

That's a really large difference. It's seems like this year we really messed up as a community. What's the distribution of difference under the null (if you do the same analysis but shuffle the gender labels)?

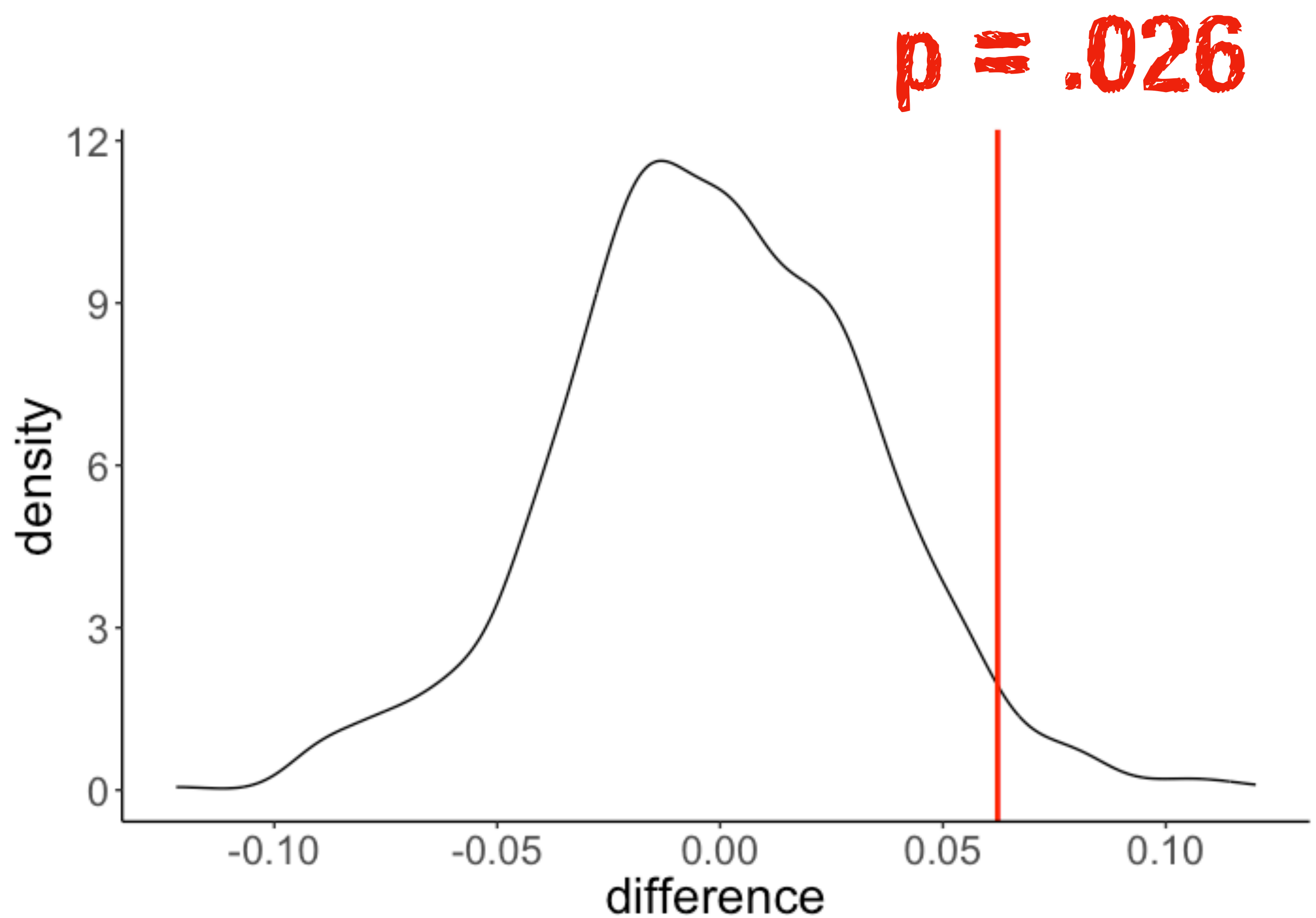
8:06 AM - 2 Mar 2019

8 Likes

**permutation test!**

# Bias in conference admission?

```r
# difference in proportion
fun.difference = function(df){
  df %>%
    as_tibble() %>%
    count(sex, accepted) %>%
    group_by(sex) %>%
    mutate(proportion = n / sum(n)) %>%
    filter(accepted == 1) %>%
    select(sex, proportion) %>%
    pivot_wider(names_from = sex,
                values_from = proportion) %>%
    mutate(difference = male - female) %>%
    pull(difference)
}

# actual difference
difference = df.conference %>%
  fun.difference()

# permutation test
df.permutation = df.conference %>%
  permute(n = 1000, sex) %>%
  mutate(difference = map_dbl(perm, ~ fun.difference(.)))
```




**significant association between sex and acceptance rate**

# Plan for today

- Generalized linear model
  - Logistic regression
  - Mixed effects logistic regression
- Bayesian data analysis
  - Comparison between frequentist and Bayesian data analysis
  - Quick flash from the past
  - Flipping coins
  - What affects the posterior?
  - Ingredients: likelihood, prior, inference
  - Doing Bayesian data analysis

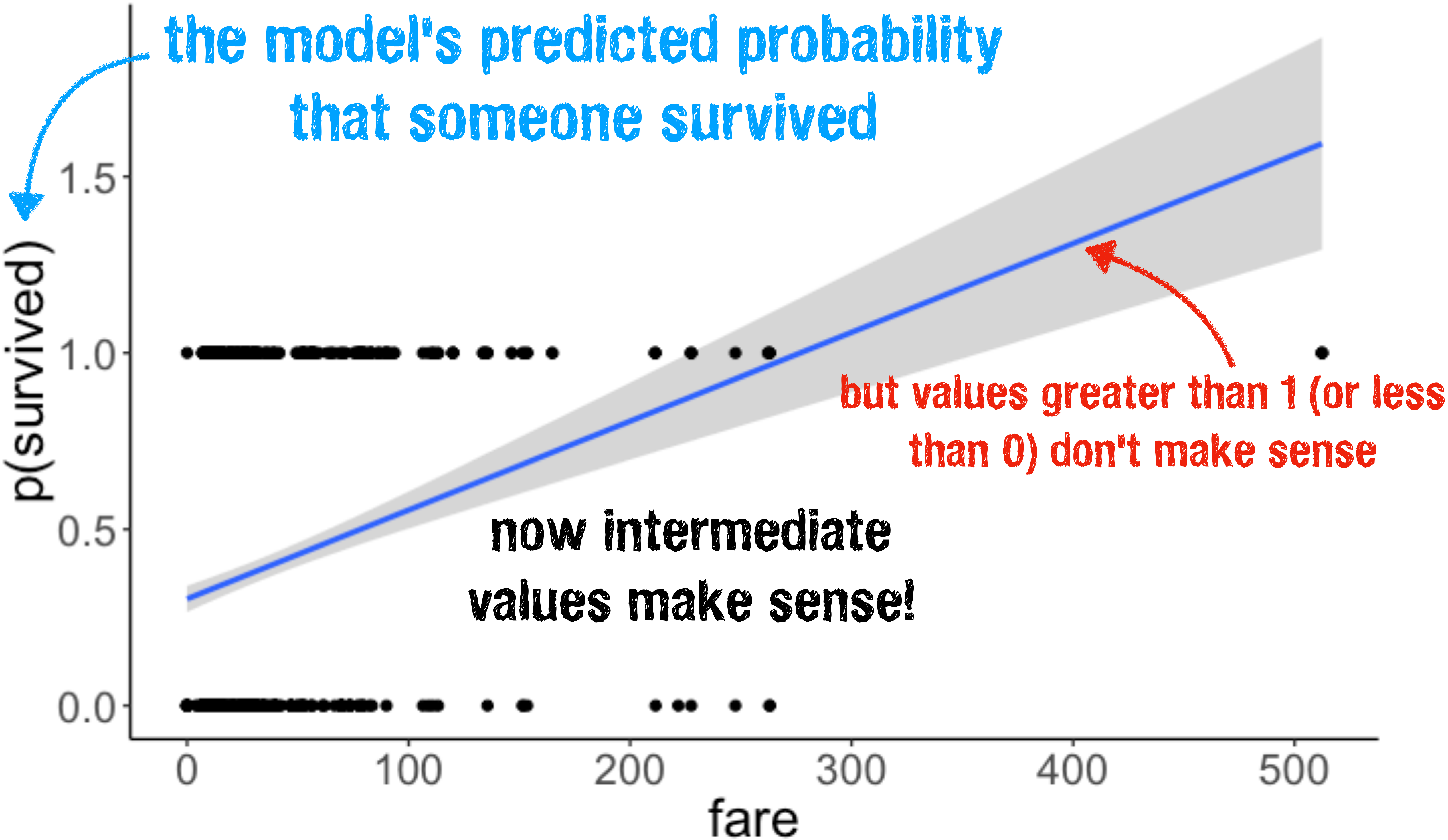
# Plan for today

- **Generalized linear model**
  - Logistic regression
  - Mixed effects logistic regression
- Bayesian data analysis
  - Comparison between frequentist and Bayesian data analysis
  - Quick flash from the past
  - Flipping coins
  - What affects the posterior?
  - Ingredients: likelihood, prior, inference
  - Doing Bayesian data analysis

# Generalized linear model

# Logistic regression

# Is there a relationship between `fare` and `survived`?



# Logit transform

$$\pi_i = b_0 + b_1 \cdot X_i + e_i$$

**predict the probability of Y**

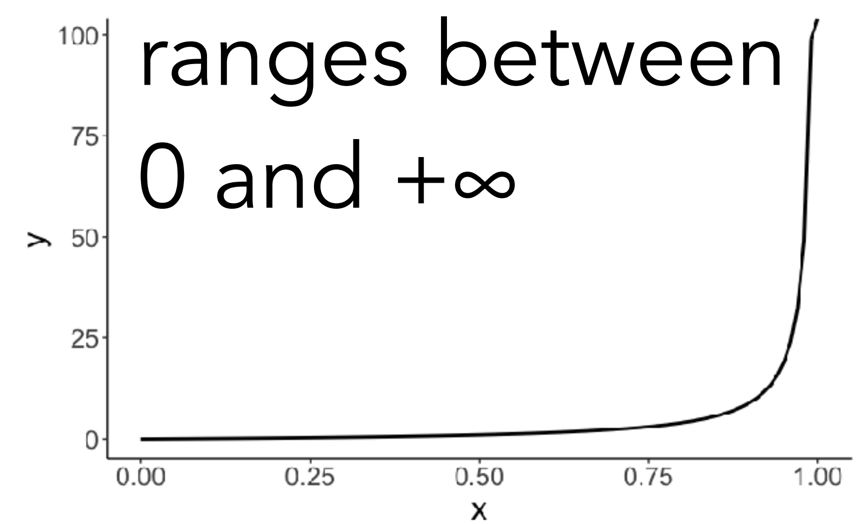
we need to transform the dependent variable so that it can take any value between -∞ and +∞ (we can then transform it back into a probability later)

$$\pi_i = P(Y_i = 1)$$

**Step 1: Calculate the "odds"**

$$\frac{P(Y_i = 1)}{P(Y_i = 0)} = \frac{\pi_i}{1 - \pi_i}$$

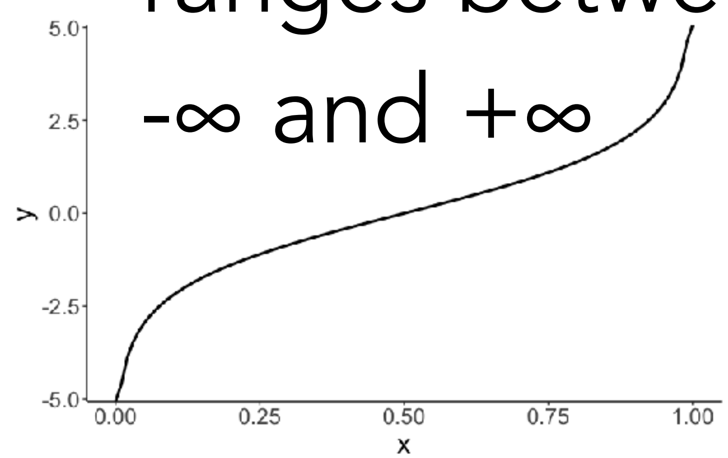
ranges between 0 and +∞




**Step 2: Take the (natural) log**

$$\ln\left(\frac{\pi_i}{1 - \pi_i}\right) = b_0 + b_1 \cdot X_i + e_i$$

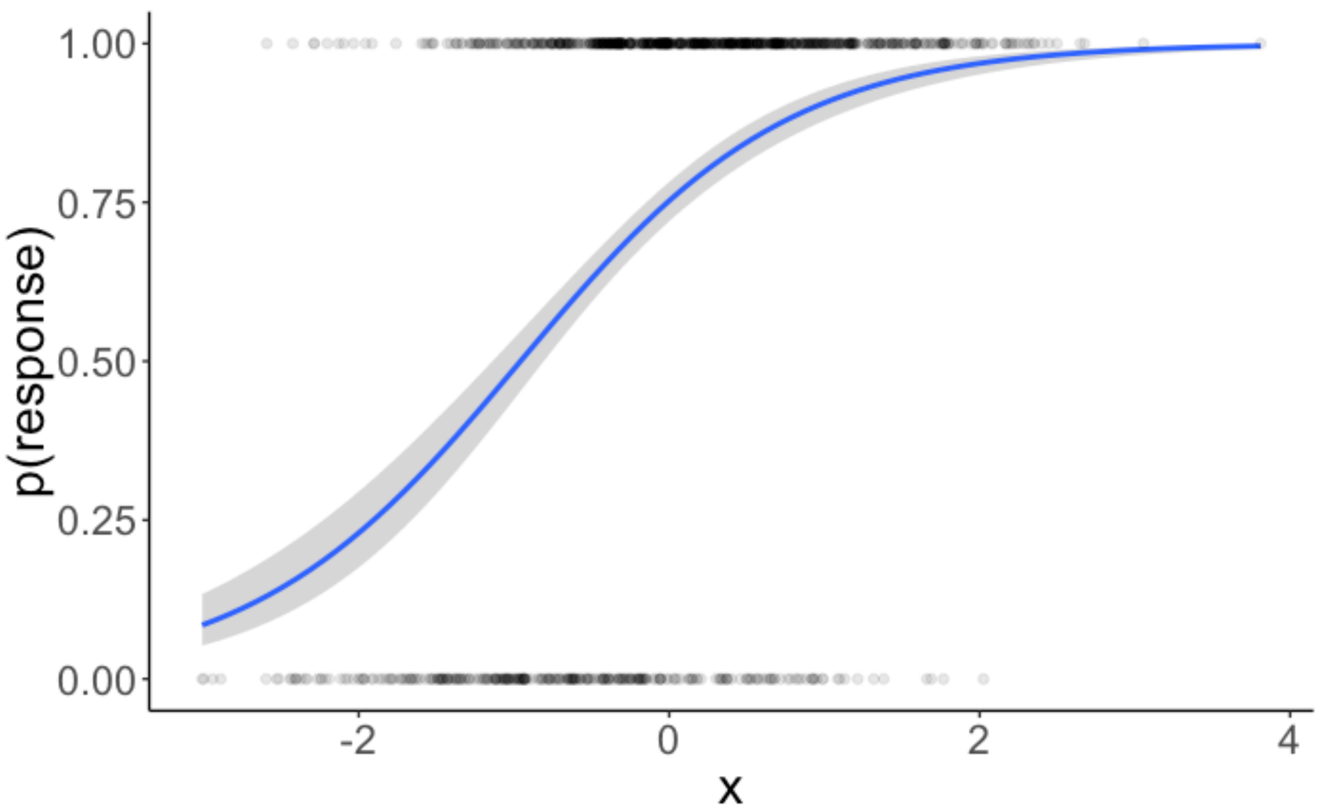
ranges between -∞ and +∞



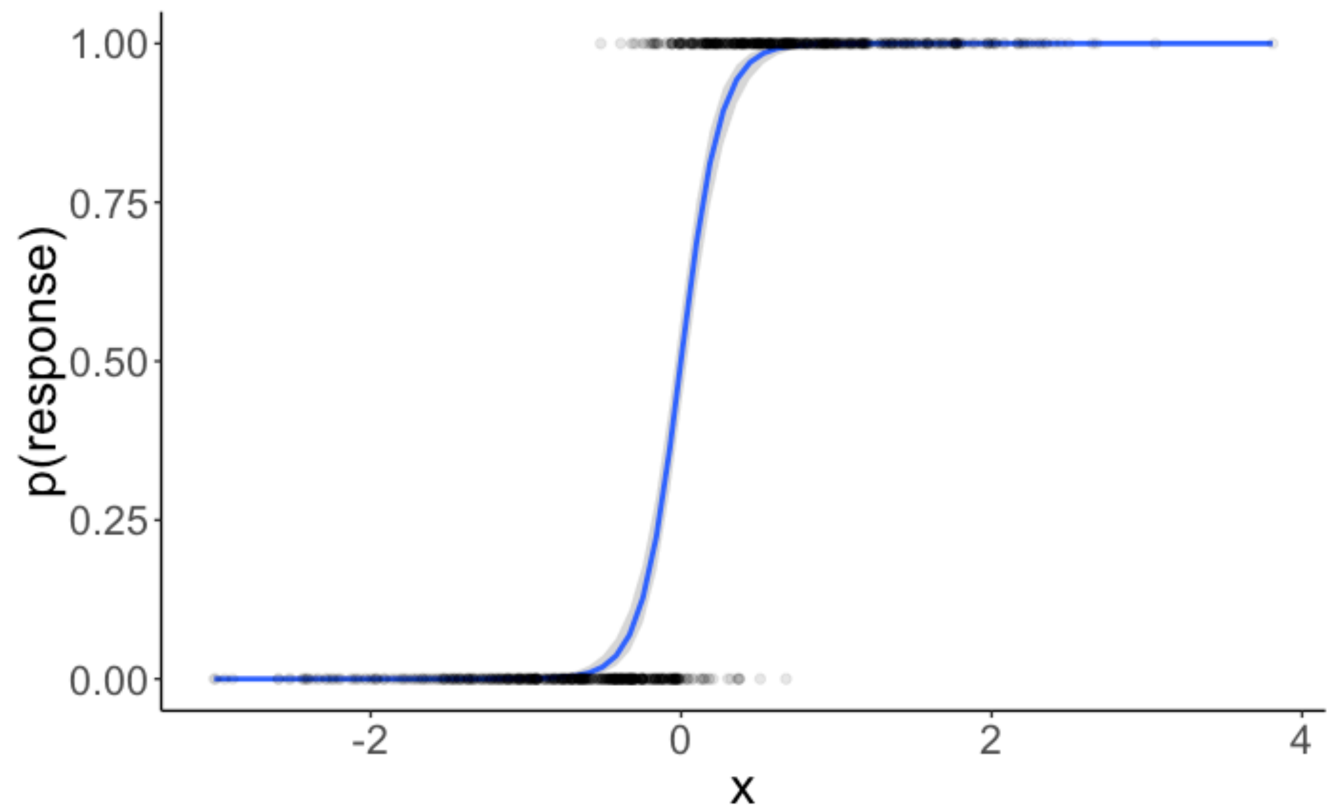
# Assessing the model fit

doesn't predict the response very well

predicts the response much better




| logLik | AIC | BIC |
|---|---|---|
| -501.65 | 1007.3 | 1017.12 |




| logLik | AIC | BIC |
|---|---|---|
| -156.37 | 316.74 | 326.55 |

# Testing hypotheses

aka checking whether it's **worth it**

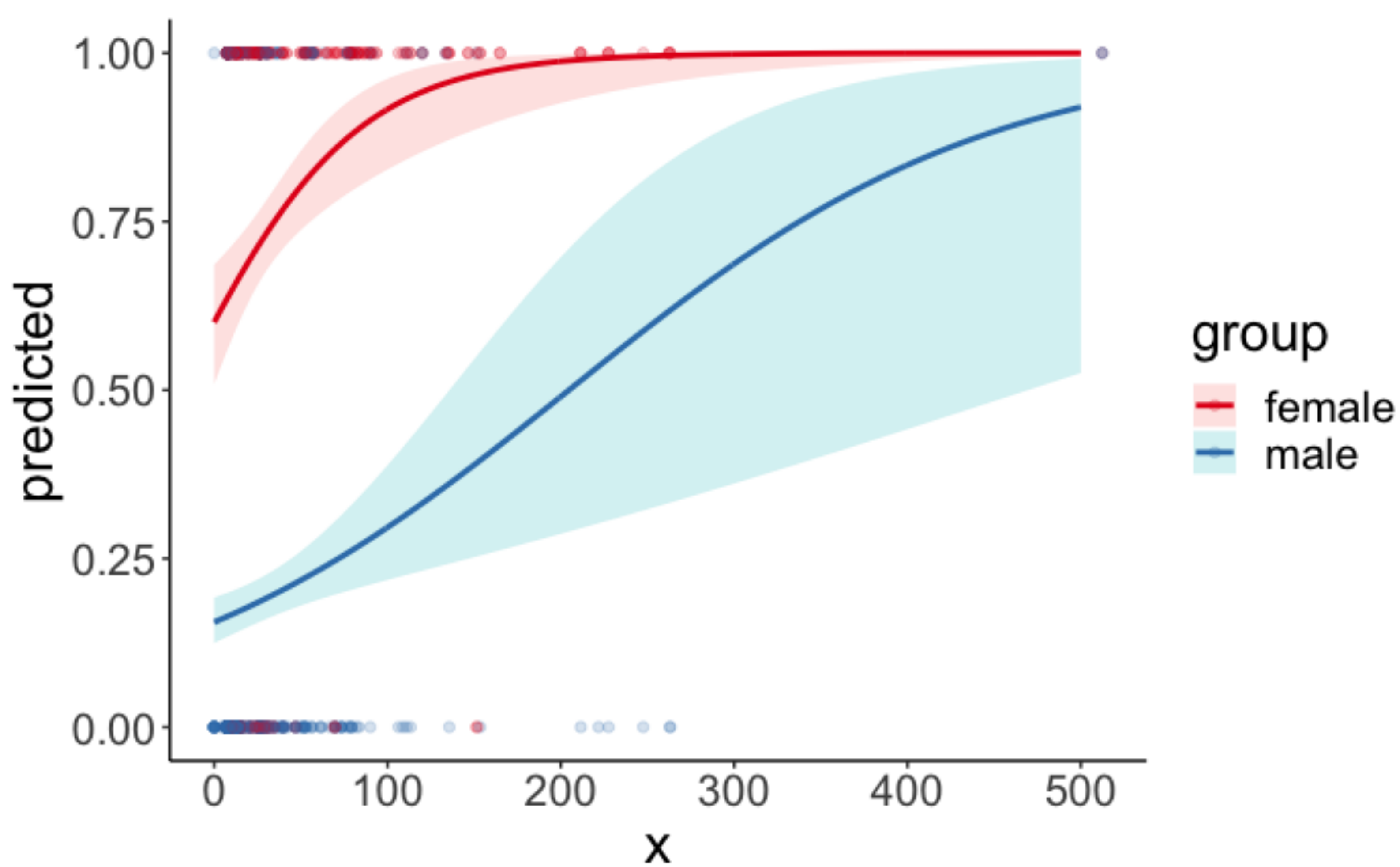
```r
# fit compact model
fit.compact = glm(formula = survived ~ 1 + fare,
                  family = "binomial",
                  data = df.titanic)

# fit augmented model
fit.augmented = glm(formula = survived ~ 1 + sex + fare,
                    family = "binomial",
                    data = df.titanic)

# likelihood ratio test
anova(fit.compact, fit.augmented, test = "LRT")
```

**we need to specify that we want a likelihood ratio test**

```
Analysis of Deviance Table

Model 1: survived ~ 1 + fare
Model 2: survived ~ 1 + sex + fare
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1       889    1117.57
2       888     884.31  1   233.26 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

# Reporting results

- Visualize the data
- Show a table with the regression results
- Report significance of different factors
- Interpreting parameter estimates is tricky -- probably best to report probabilities for a few example cases




```
# Predicted values of survived
# x = fare

# sex = female

  x | Predicted |   SE |        95% CI
----------------------------------------
  0 |      0.60 | 0.19 | [0.51, 0.69]
100 |      0.92 | 0.42 | [0.83, 0.96]
200 |      0.99 | 0.95 | [0.93, 1.00]
300 |      1.00 | 1.48 | [0.97, 1.00]
400 |      1.00 | 2.02 | [0.99, 1.00]
500 |      1.00 | 2.55 | [1.00, 1.00]

# sex = male

  x | Predicted |   SE |        95% CI
----------------------------------------
  0 |      0.16 | 0.13 | [0.12, 0.19]
100 |      0.30 | 0.21 | [0.22, 0.39]
200 |      0.49 | 0.44 | [0.29, 0.70]
300 |      0.69 | 0.69 | [0.36, 0.90]
400 |      0.83 | 0.94 | [0.44, 0.97]
500 |      0.92 | 1.19 | [0.53, 0.99]
```

# Assumptions

- linearity (between predictors and log odds)
- independence (for logistic regression)
- no (strong) multi-collinearity (= high correlation between predictors)

- model fails to converge when there is **complete separation**:
  - if outcome variable can be perfectly predicted by a (combination of) predictor(s)

# Different kinds of generalized models

## Different linking functions

```
binomial(link = "logit")

gaussian(link = "identity")

Gamma(link = "inverse")

inverse.gaussian(link = "1/mu^2")

poisson(link = "log")

quasi(link = "identity", variance = "constant")

quasibinomial(link = "logit")

quasipoisson(link = "log")
```

**apply different transformations to the dependent variable**

# Plan for today

- Generalized linear model
  - Logistic regression
  - **Mixed effects logistic regression**
- Bayesian data analysis
  - Comparison between frequentist and Bayesian data analysis
  - Quick flash from the past
  - Flipping coins
  - What affects the posterior?
  - Ingredients: likelihood, prior, inference
  - Doing Bayesian data analysis

# Mixed effects logistic regression

# Mixed effects logistic regression

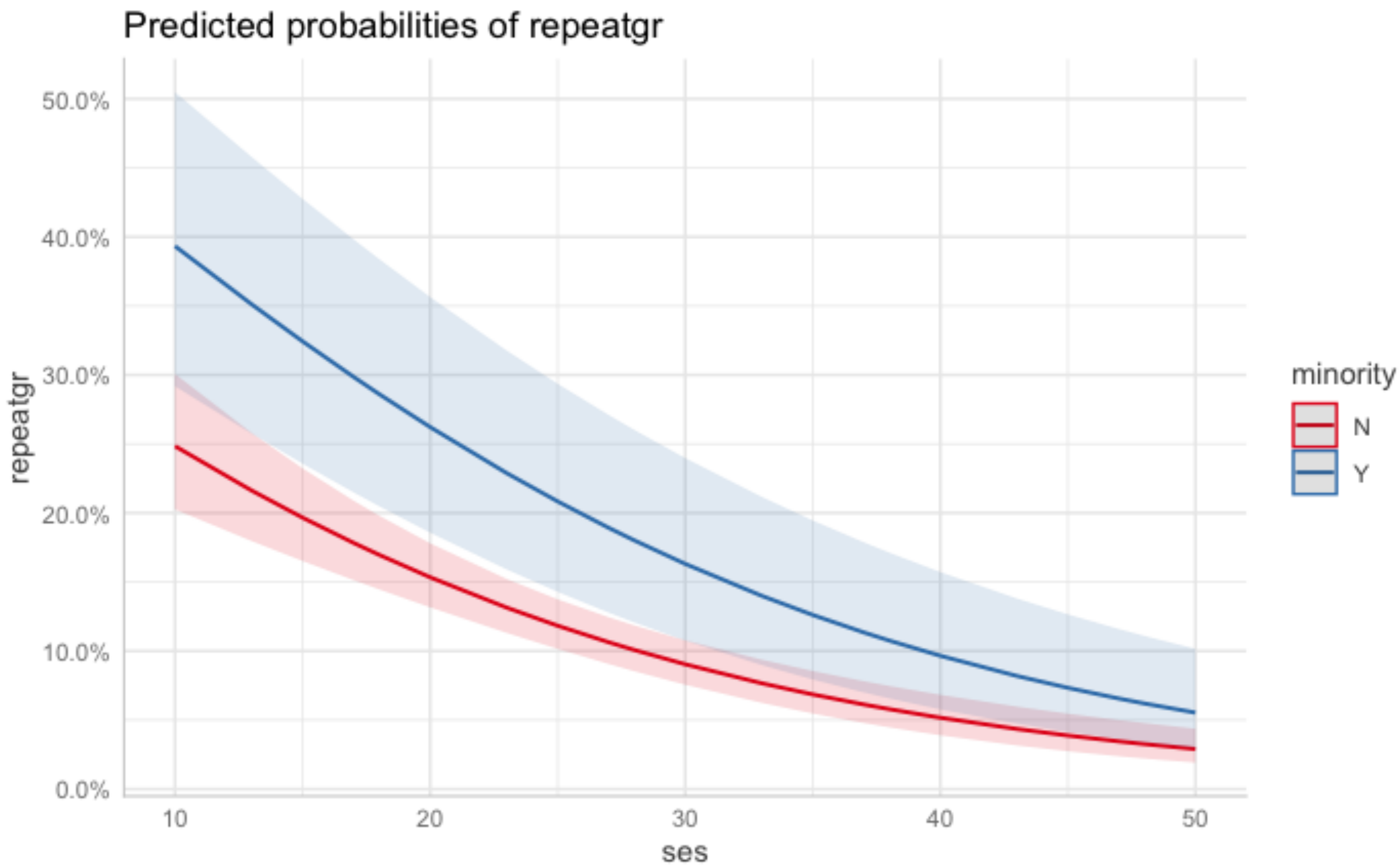
repeated a grade: yes / no

```r
fit =  glmer(repeatgr ~ 1 + ses * Minority + (1 | schoolNR),
             data = df.language,
             family = "binomial")

fit %>% summary()
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace Approximation) ['glmerMod']
 Family: binomial  ( logit )
Formula: repeatgr ~ 1 + ses + minority + (1 | school_nr)
   Data: df.language

     AIC      BIC   logLik deviance df.resid
  1659.1   1682.1   -825.6   1651.1     2279

Scaled residuals:
    Min      1Q  Median      3Q     Max
-0.9235 -0.4045 -0.3150 -0.2249  5.8372

Random effects:
 Groups    Name        Variance Std.Dev.
 school_nr (Intercept) 0.2489   0.4989
Number of obs: 2283, groups:  school_nr, 131

Fixed effects:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.506291   0.197570  -2.563  0.01039 *
ses         -0.060086   0.007524  -7.986 1.39e-15 ***
minorityY    0.673612   0.238660   2.822  0.00477 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
          (Intr) ses
ses       -0.898
minorityY -0.308  0.208
```

# Mixed effects logistic regression

```
ggpredict(model = fit,
          terms = c("ses [all]", "minority")) %>%
  plot()
```



# Plan for today

- Generalized linear model
  - Logistic regression
  - Mixed effects logistic regression
- **Bayesian data analysis**
  - Comparison between frequentist and Bayesian data analysis
  - Quick flash from the past
  - Flipping coins
  - What affects the posterior?
  - Ingredients: likelihood, prior, inference
  - Doing Bayesian data analysis

# What is Bayesian data analysis?

Discuss with your neighbor for 3 minutes

# Plan for today

- Generalized linear model
    - Logistic regression
    - Mixed effects logistic regression
- Bayesian data analysis
    - **Comparison between frequentist and Bayesian data analysis**
    - Quick flash from the past
    - Flipping coins
    - What affects the posterior?
    - Ingredients: likelihood, prior, inference
    - Doing Bayesian data analysis

# Comparison between frequentist and Bayesian data analysis

# Goal of data analysis: Inference about the world

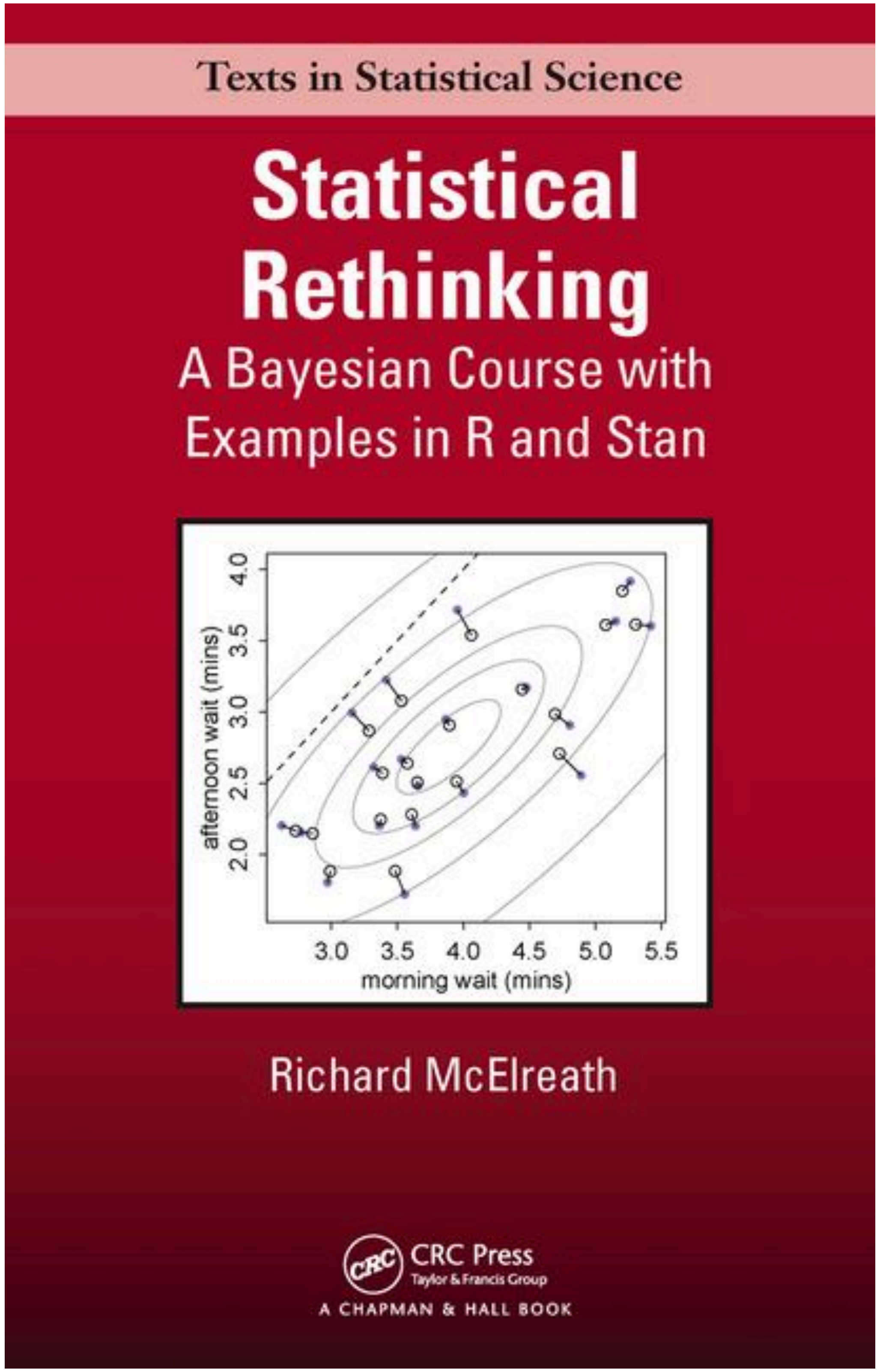
## Frequentist statistics

- 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 $\alpha$

## Bayesian statistics

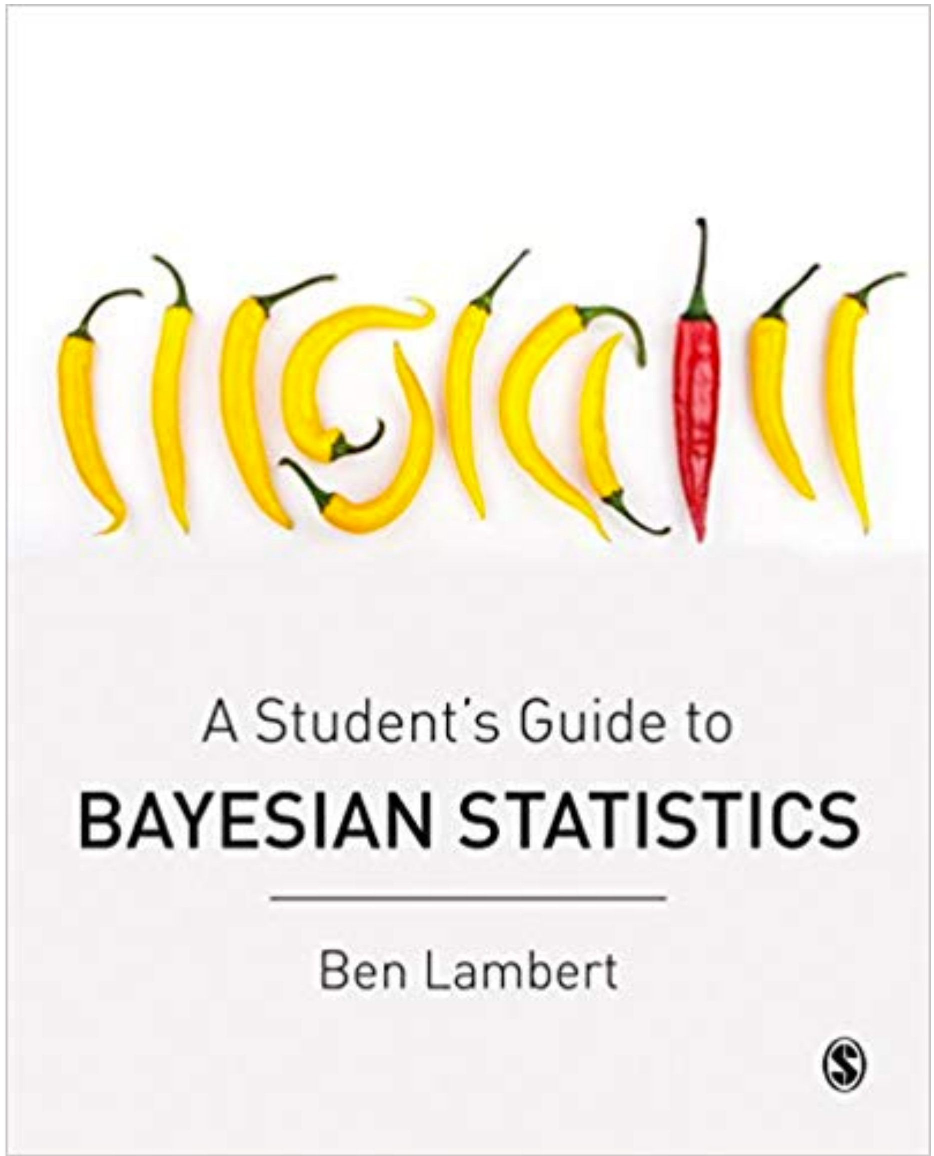
- directly test hypotheses of interest
- define prior over hypotheses **p(H)**
- compute likelihood of the data for each hypothesis **p(D|H)**
- use Bayes' rule to infer the posterior over hypotheses given the data **p(H|D)**

# Good books on Bayesian data analysis




- nice hands-on book (which uses R throughout)
- unfortunately, mostly in base R
- however, rewrite of all the code with "`tidyverse`" and "`BRMS`" is here: https://bookdown.org/connect/#/apps/1850/access
- video lectures are available here: https://goo.gl/4zZWTv
- Version 2 is almost out (a pre-print is on Canvas in the `papers/` folder)

# Good books on Bayesian data analysis




- less hands-on (no R code)
- very nice visualizations of key concepts
- goes a little more into theoretical details (but in a mostly non-mathy way)
- also comes with video lectures

# Objections to frequentist NHST

null hypothesis significance testing

- p-value is not a measure of evidential support
  - becomes smaller as $N$ increases
- results are often misinterpreted (both p-values and confidence intervals are not particularly intuitive)
- what we want to know: $p(\text{Hypothesis} \mid \text{Data})$
- what we calculate: $p(\text{Data} \mid \text{Null Hypothesis})$

Wasserstein, Schirm, & Lazar (2019) Moving to a World Beyond p < 0.05. *The American Statistician*

# Frequentists vs. Bayesian

- both want to evaluate the evidence for a hypothesis using a sample of data **p(H|D)**
- it's often easier to calculate the inverse: the probability of the data given a hypothesis **p(D|H)**
- frequentists use a rule of thumb (p-value)
- Bayesians use Bayes' rule
- Bayesians insist on describing uncertainty explicitly through probability distributions

# Why don't more people use Bayesian Statistics?

- supposedly more difficult
  - relies on the logic of probability theory
- reliance on a *prior*
- reliance on computing and simulation
  - we can't just use SPSS
  - but we can use JASP (Just Another Statistics Program)

**and we've already learned how to simulate and visualize data in this class!**

# What are (some of) the benefits of Bayesian data analysis?

- intuitive model testing and comparison
  - compare simulated data with the real data
- straightforward interpretation of results
  - Bayesian credible intervals vs. Confidence intervals
- more model flexibility
  - adequately express assumptions about the data-generating process
- less opportunity for misuse of test
  - we build models from the ground up, making our assumptions explicit
- better predictions!

# Plan for today

- Generalized linear model
  - Logistic regression
  - Mixed effects logistic regression
- Bayesian data analysis
  - Comparison between frequentist and Bayesian data analysis
  - **Quick flash from the past**
  - Flipping coins
  - What affects the posterior?
  - Ingredients: likelihood, prior, inference
  - Doing Bayesian data analysis

# Flash from the past

# Clue guide to probability

we derived this using the definition of conditional probability

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

posterior

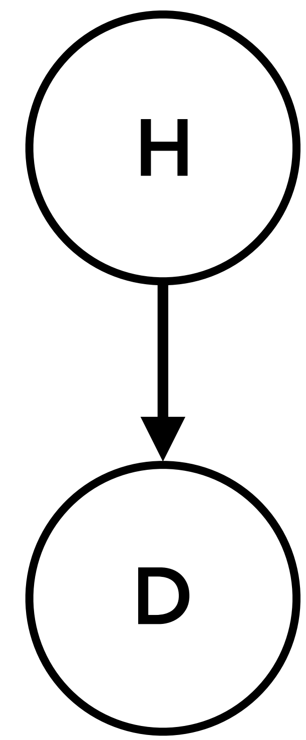
likelihood prior

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

subjective probability interpretation

$H$ = Hypothesis

$D$ = Data




**formal framework for learning from data**

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

# Clue guide to probability

A patient named Fred is tested for a disease called *conditionitis*, a medical condition that afflicts **1% of the population**. The test result is positive, i.e., the test claims that Fred has the disease. Let **D** be the event that Fred has the disease and **T** be the event that he tests positive.

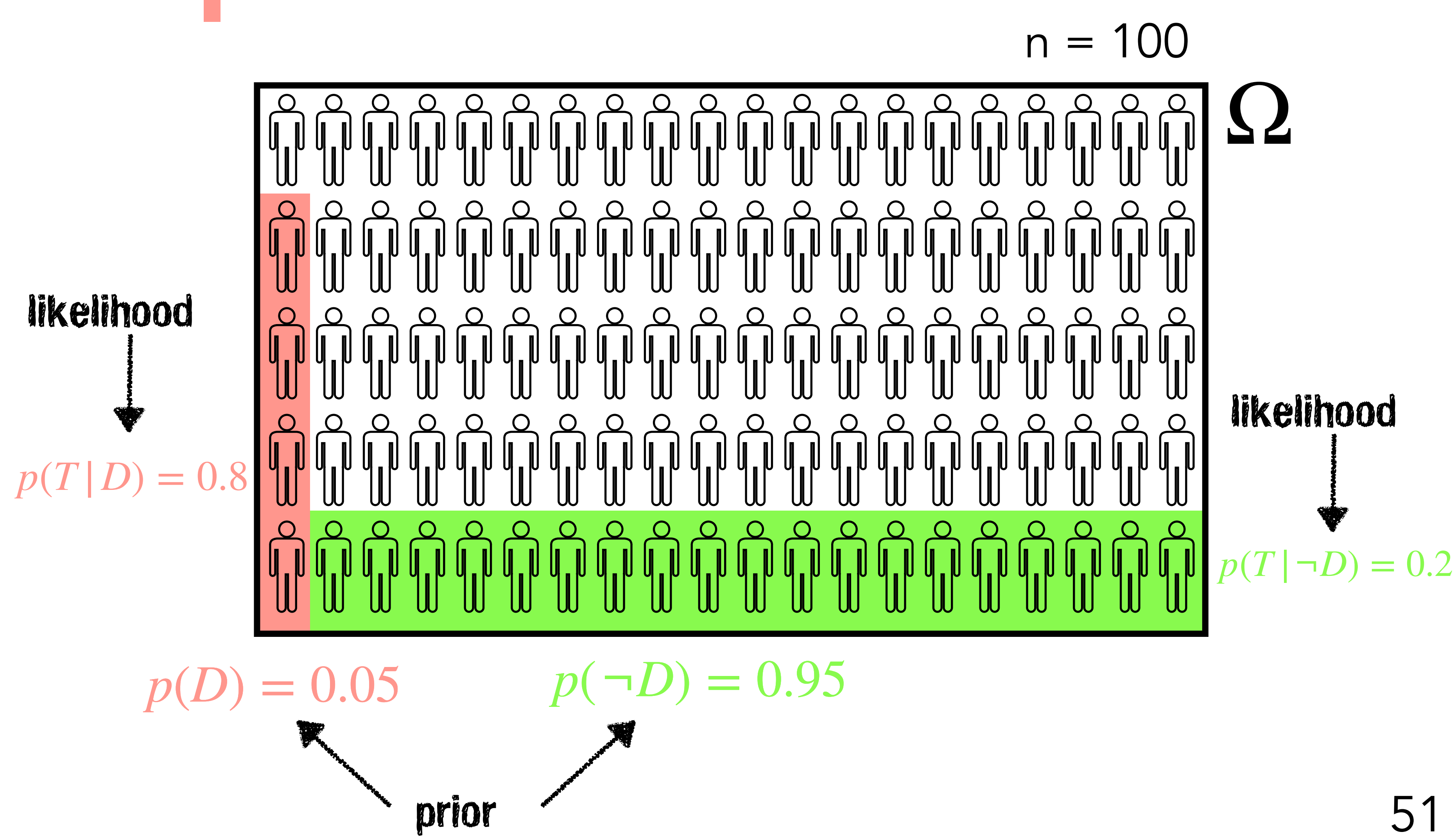
Suppose that the test is "95% accurate"; there are different measures of the accuracy of a test, but in this problem it is assumed to mean that $\mathbf{P(T|D) = 0.95}$ and $\mathbf{P(\neg T|\neg D) = 0.95}$. The quantity $P(T|D)$ is known as the *sensitivity* (= true positive rate) of the test, and $P(\neg T|\neg D)$ is known as the *specificity* (= true negative rate).

Find the conditional probability that Fred has *conditionitis*, given the evidence provided by the positive test result.

# Clue guide to probability

$$p(D\,|\,T) = \frac{p(T\,|\,D)\cdot p(D)}{p(T\,|\,D)\cdot p(D) + p(T\,|\,\neg D)\cdot p(\neg D)}$$

$$p(D\,|\,T) = \frac{4}{4 + 19} = 0.174$$



# Summer camp

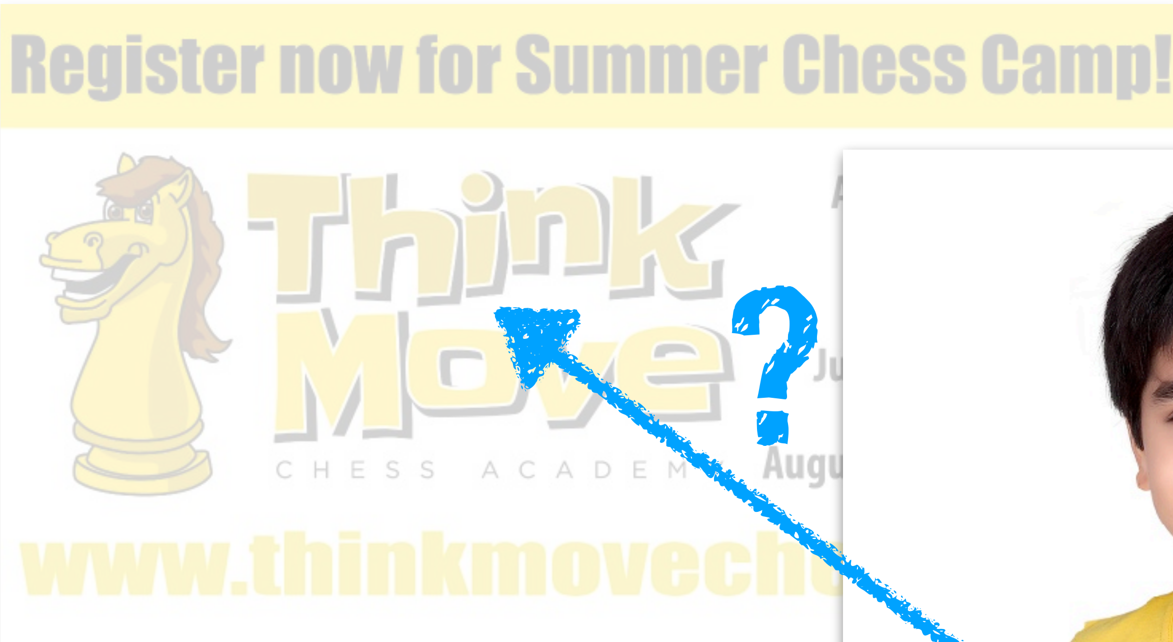


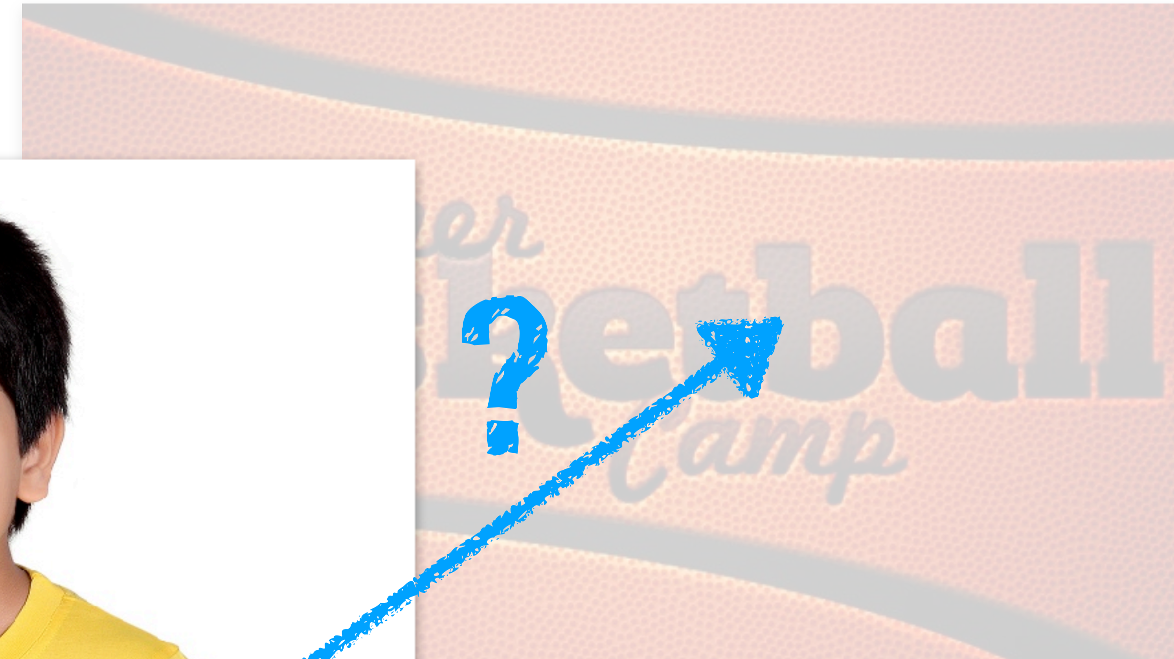
twice as many kids go to the basketball camp

$X \sim$ Normal($\mu = 170$, $\sigma = 8$)

$X \sim$ Normal($\mu = 180$, $\sigma = 10$)

# Summer camp



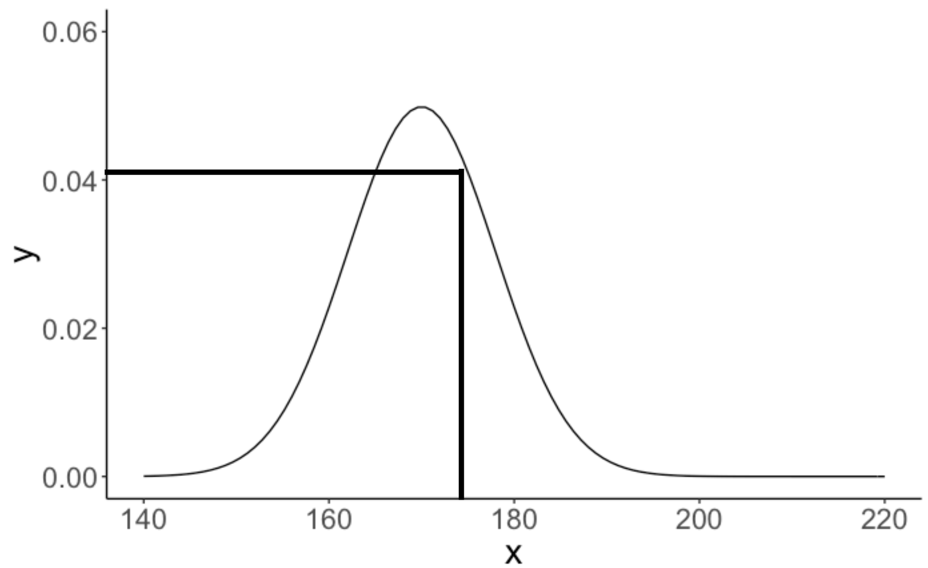

twice as ma ... etball camp

$X \sim$ Normal($\mu = 170$ ... nal($\mu = 180, \sigma = 10$)

# Summer camp

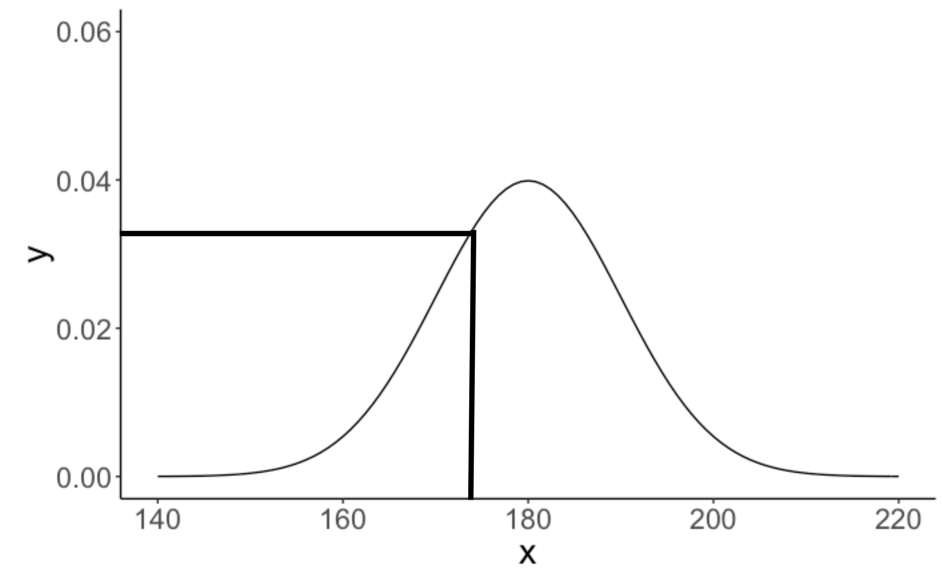
**prior** $p(\text{chess}) = \frac{1}{3}$ $p(\text{basketball}) = \frac{2}{3}$

**likelihood**




`dnorm(175, mean = 170, sd = 8)`

= 0.041




`dnorm(175, mean = 180, sd = 10)`

= 0.035

**posterior**

$$p(\text{basketball} \mid 175) = \frac{p(175 \mid \text{basketball}) \cdot p(\text{basketball})}{p(175 \mid \text{basketball}) \cdot p(\text{basketball}) + p(175 \mid \text{chess}) \cdot p(\text{chess})}$$

$$p(\text{basketball} \mid 175) = \frac{0.035 \cdot 2/3}{0.035 \cdot 2/3 + 0.041 \cdot 1/3} \approx 0.63$$

send the kid to the basketball gym!

# Plan for today

- Generalized linear model
  - Logistic regression
  - Mixed effects logistic regression
- Bayesian data analysis
  - Comparison between frequentist and Bayesian data analysis
  - Quick flash from the past
  - **Flipping coins**
  - What affects the posterior?
  - Ingredients: likelihood, prior, inference
  - Doing Bayesian data analysis

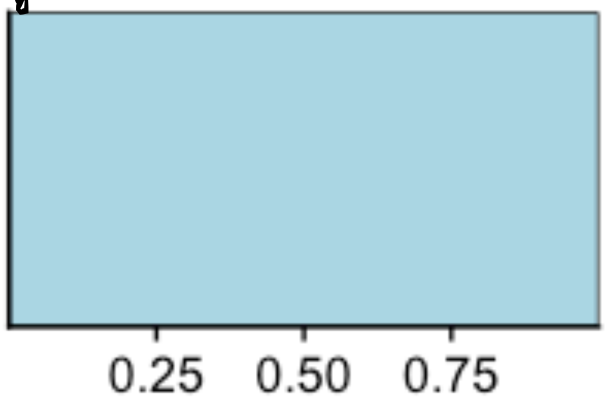
# Flipping coins

# Flipping coins

# Learning from data

How does/should our belief change as evidence comes in?

prior




Today's posterior is tomorrow's prior.

$p(\theta \mid \mathrm{n_{success}} = 6, \mathrm{n_{trials}} = 8)$

# Coin flip example

## Which coin did I flip?

**Hypotheses**

p = 0.1


p = 0.5


p = 0.9


**Data**

## #8 tails, #2 heads

# Bayesian Recipe

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

**+ a healthy dose of Bayes' rule**

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

# Coin flip example

```r
# data
data = rep(0:1, c(8, 2))

# parameters
theta = c(0.1, 0.5, 0.9)

# prior
prior = c(0.25, 0.5, 0.25)

# likelihood
likelihood = dbinom(sum(data == 1), size = length(data), prob = theta)

# posterior
posterior = likelihood * prior / sum(likelihood * prior)
```

**8 tails and 2 heads**

**hypotheses**

**prior over the hypotheses**

**binomial distribution**

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

law of total probability:

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

# Coin flip example

```r
# data
data = rep(0:1, c(8, 2))

# parameters
theta = c(0.1, 0.5, 0.9)

# prior
prior = c(0.25, 0.5, 0.25)

# likelihood
likelihood = dbinom(sum(data == 1), size = length(data), prob = theta)

# posterior
posterior = likelihood * prior / sum(likelihood * prior)
```

8 tails and 2 heads

hypotheses

prior over the hypotheses

binomial distribution

multiply

re-normalize

| theta | prior | likelihood | prior_x_like | posterior |
|---|---|---|---|---|
| 0.1 | 0.25 | 0.19 | 0.0475 | 0.69 |
| 0.5 | 0.50 | 0.04 | 0.02 | 0.31 |
| 0.9 | 0.25 | 0.00 | 0.00 | 0.00 |

# Coin flip example

data: #8 tails, #2 heads

## Which coin was flipped?

what the model knows before having seen the data


learning by conditioning on the data

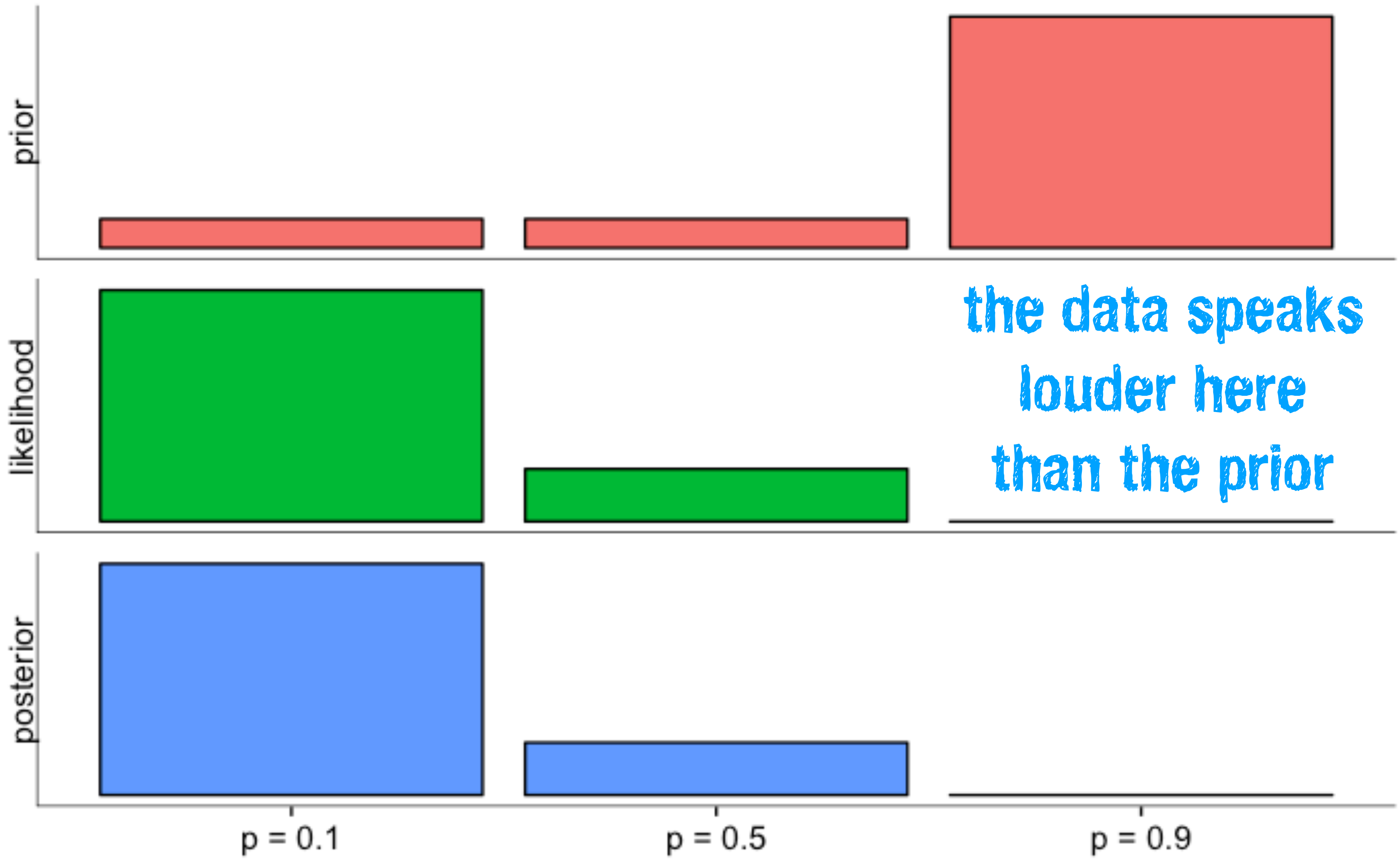
what the model knows after having seen the data

**posterior = multiplicative weighting of prior and likelihood**

# Coin flip example

data: #8 tails, #2 heads

## Which coin was flipped?

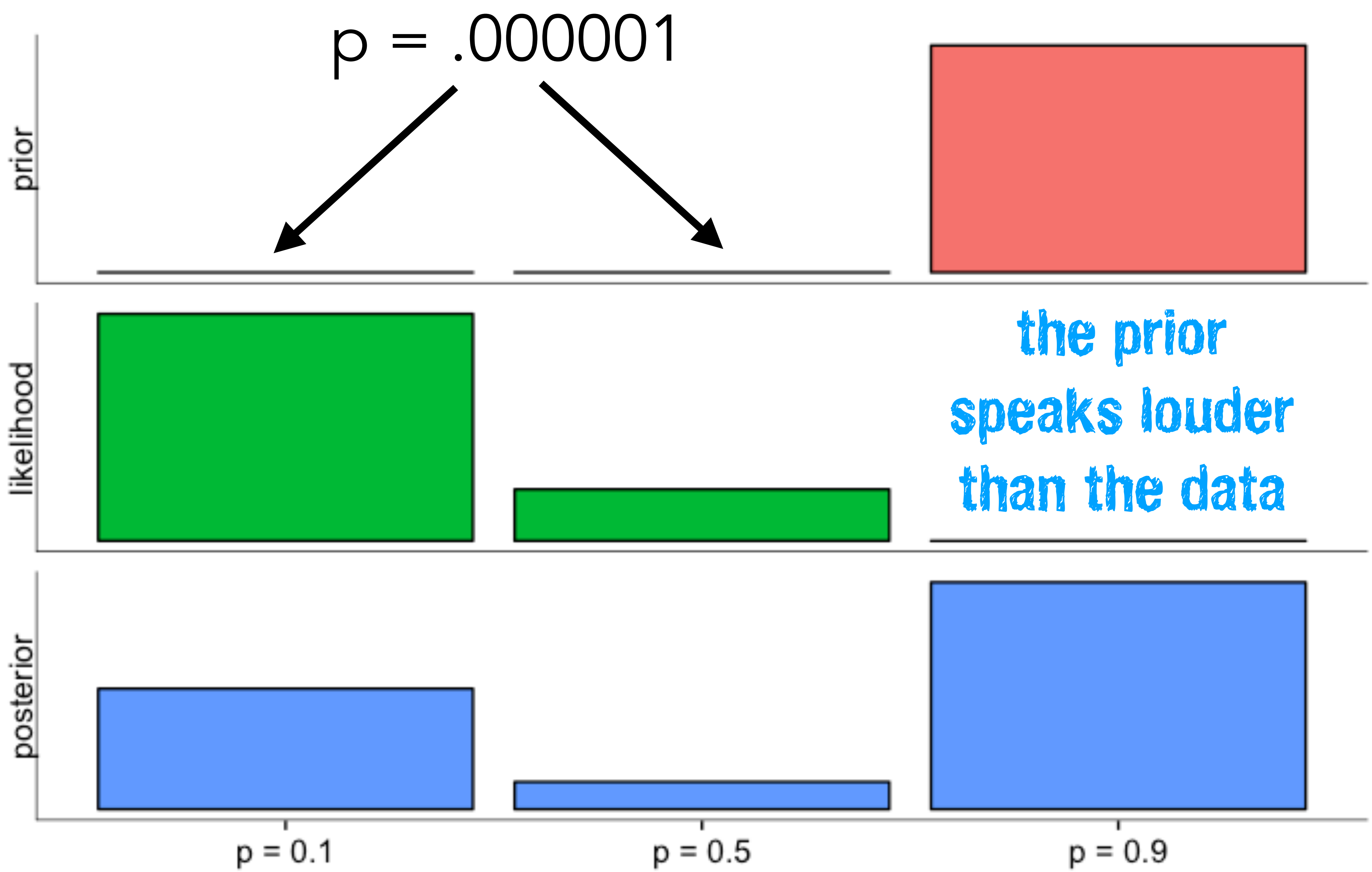



**posterior = multiplicative weighting of prior and likelihood**

# Coin flip example

**data: #8 tails, #2 heads**

**Which coin was flipped?**




**posterior = multiplicative weighting of prior and likelihood**

# Plan for today

- Generalized linear model
  - Logistic regression
  - Mixed effects logistic regression
- Bayesian data analysis
  - Comparison between frequentist and Bayesian data analysis
  - Quick flash from the past
  - Flipping coins
  - **What affects the posterior?**
  - Ingredients: likelihood, prior, inference
  - Doing Bayesian data analysis

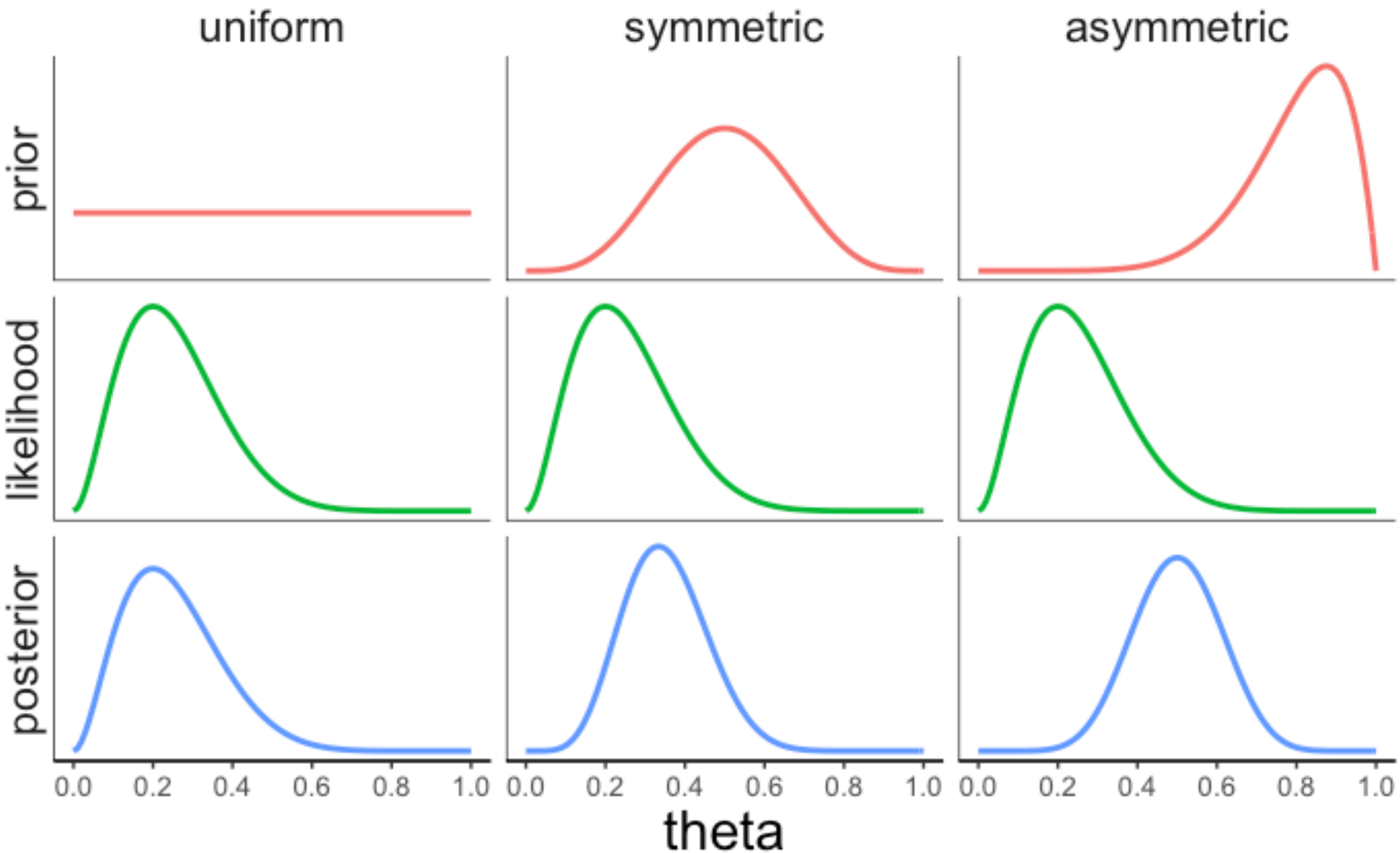
# What affects the posterior?

# What affects the posterior?

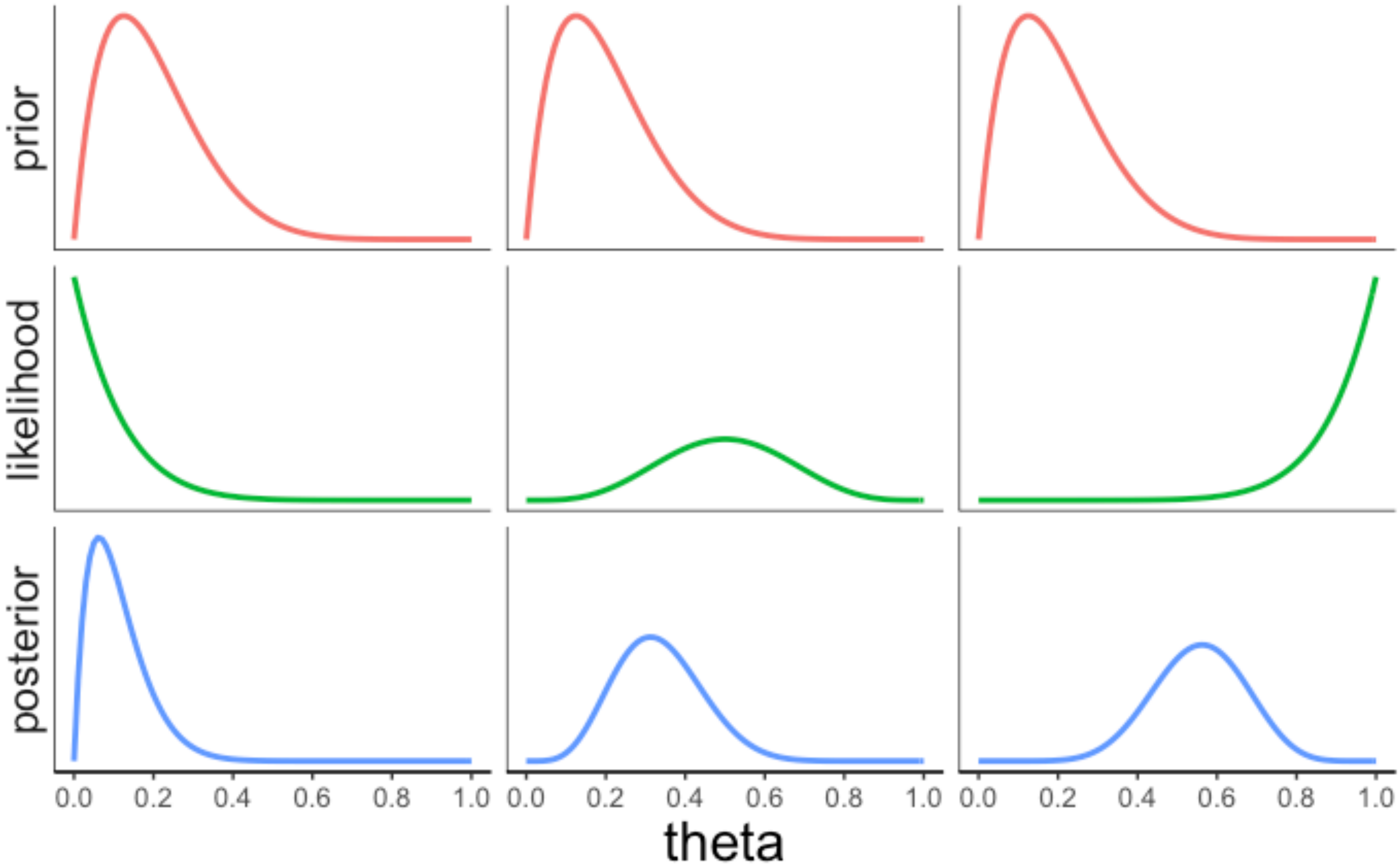
1. the prior over hypotheses
2. the likelihood of the data given each hypothesis

$$\underbrace{p(H \mid D)}_{\text{Posterior}} = \frac{\overbrace{p(D \mid H)}^{\text{Likelihood}} \cdot \overbrace{p(H)}^{\text{Prior}}}{\underbrace{p(D)}_{\text{Normalizing constant}}}$$

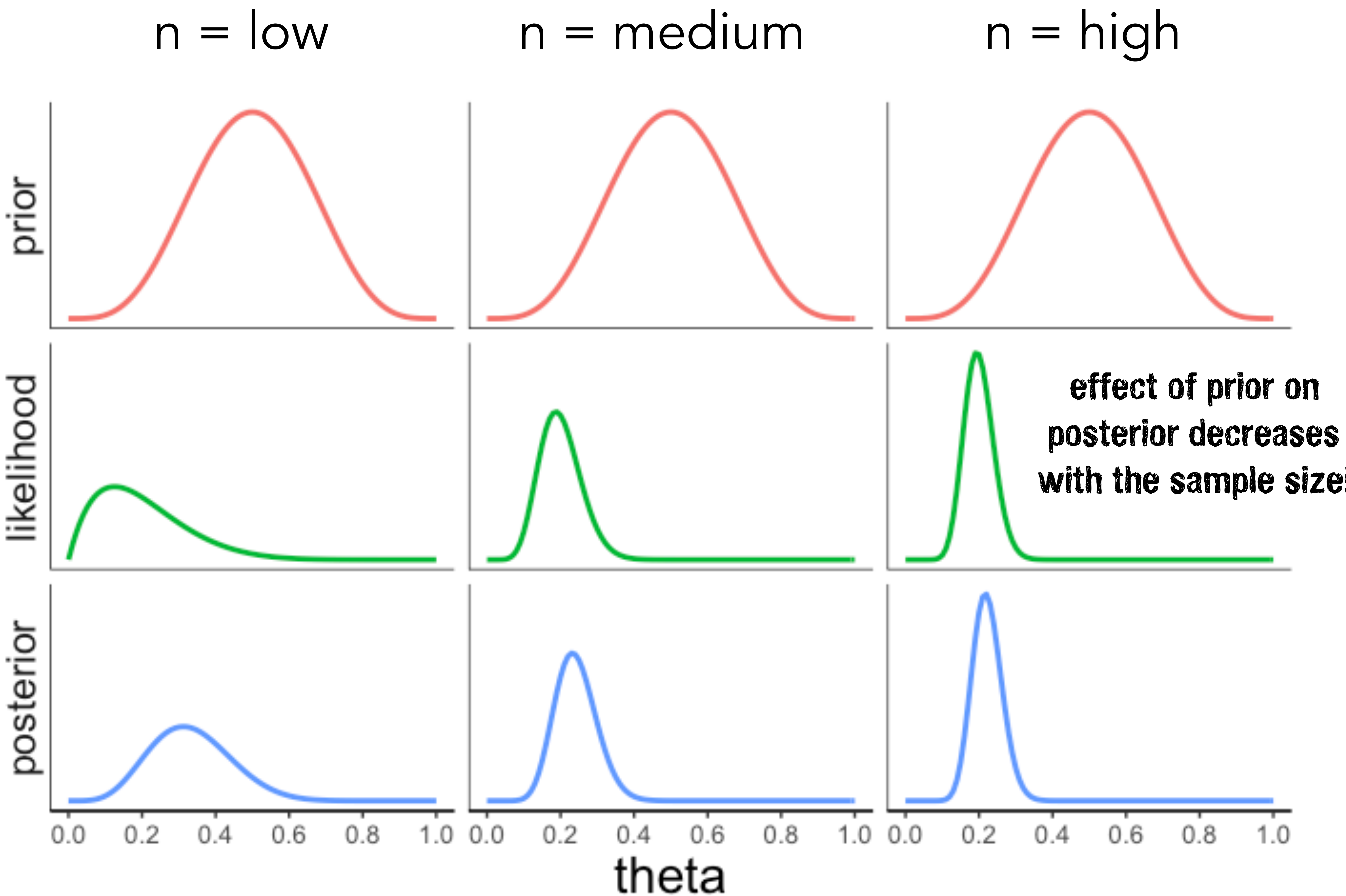
# The effect of the **prior**



# The effect of the **likelihood**



# The effect of **sample size**



# Plan for today

- Generalized linear model
  - Logistic regression
  - Mixed effects logistic regression
- Bayesian data analysis
  - Comparison between frequentist and Bayesian data analysis
  - Quick flash from the past
  - Flipping coins
  - What affects the posterior?
  - **Ingredients: likelihood, prior, inference**
  - Doing Bayesian data analysis

**Ingredients:** likelihood, prior, inference

# Ingredients

Likelihood Prior

Posterior

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

Normalizing constant

Likelihood Prior

Posterior

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

Normalizing constant

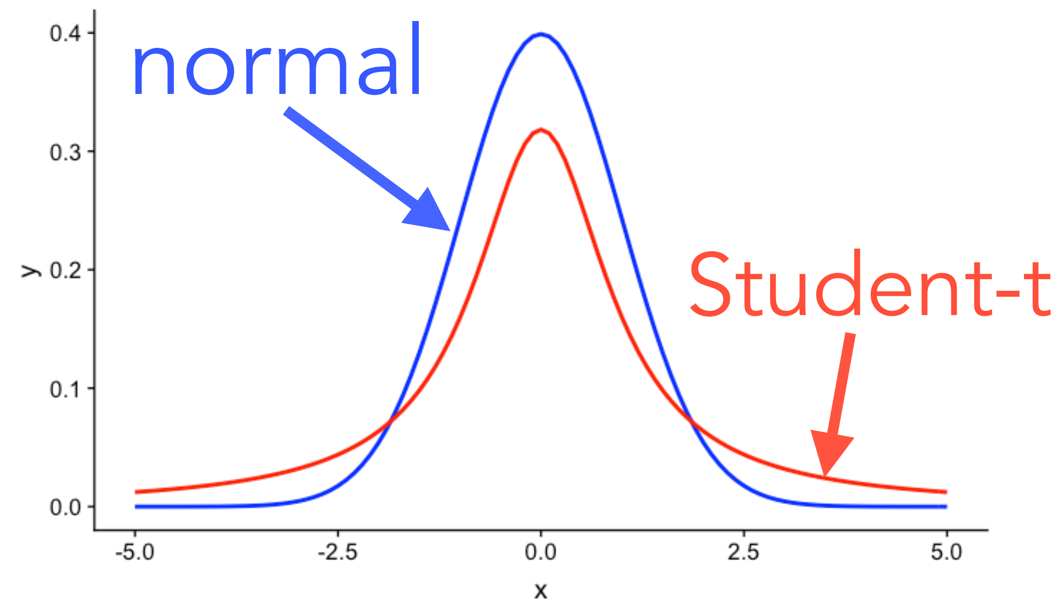
# Likelihood

- **What probabilistic model describes best how the data were generated?**
- How to build a (Bayesian) model?
    - What real-life behavior should the model explain?
    - What assumptions can you make about the behavior?
        - What's the nature of your dependent variable (e.g. binary, ordered, continuous)?
    - Does the model re-create the behavior of interest?

# Likelihood

- **Bernoulli**:
  - binary data
  - a single trial
- **Binomial**:
  - binary data
  - fixed number of total trials
  - trial outcomes are independent
  - probability of success is the same in each trial
- **Poisson**: count of discrete events
- **Beta-binomial**: like binomial but probability of success may change across trials
- ...

# Likelihood




- **Normal**:
    - continuous data
    - unbounded outcomes
    - outcome is the result of a large number of additive factors
- **Student-*t***:
    - same as Normal
    - handles greater variability in the data (distribution has **fat tails**)

# Prior

Likelihood     Prior

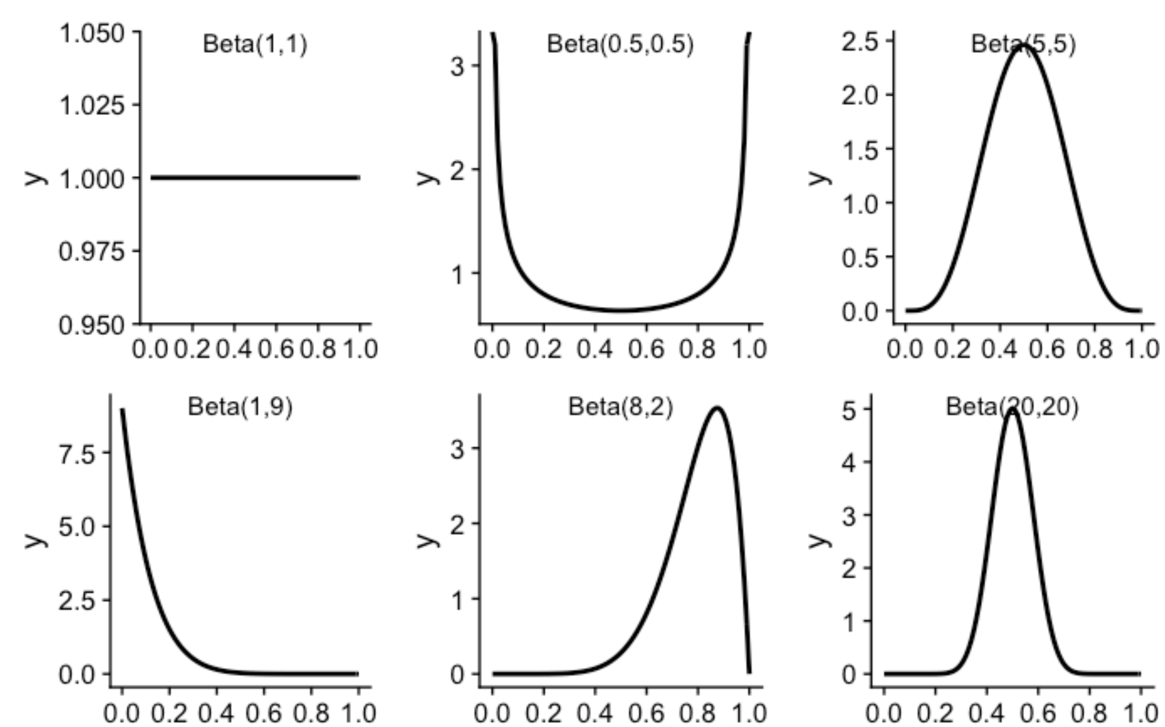
Posterior

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

Normalizing constant

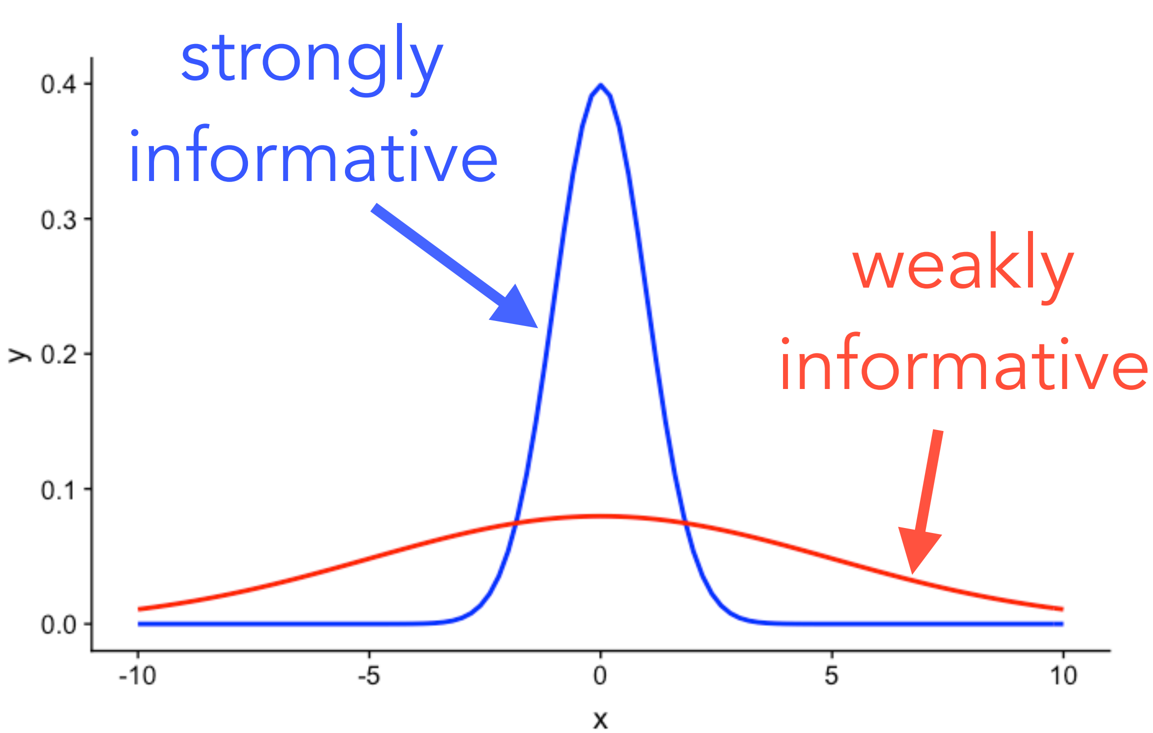
# Prior

- **uniform**:
  - continuous or discrete
  - bounded between minimum and maximum
- **beta:**
  - continuous parameters
  - bounded between 0 and 1
  - can model a wide range of priors



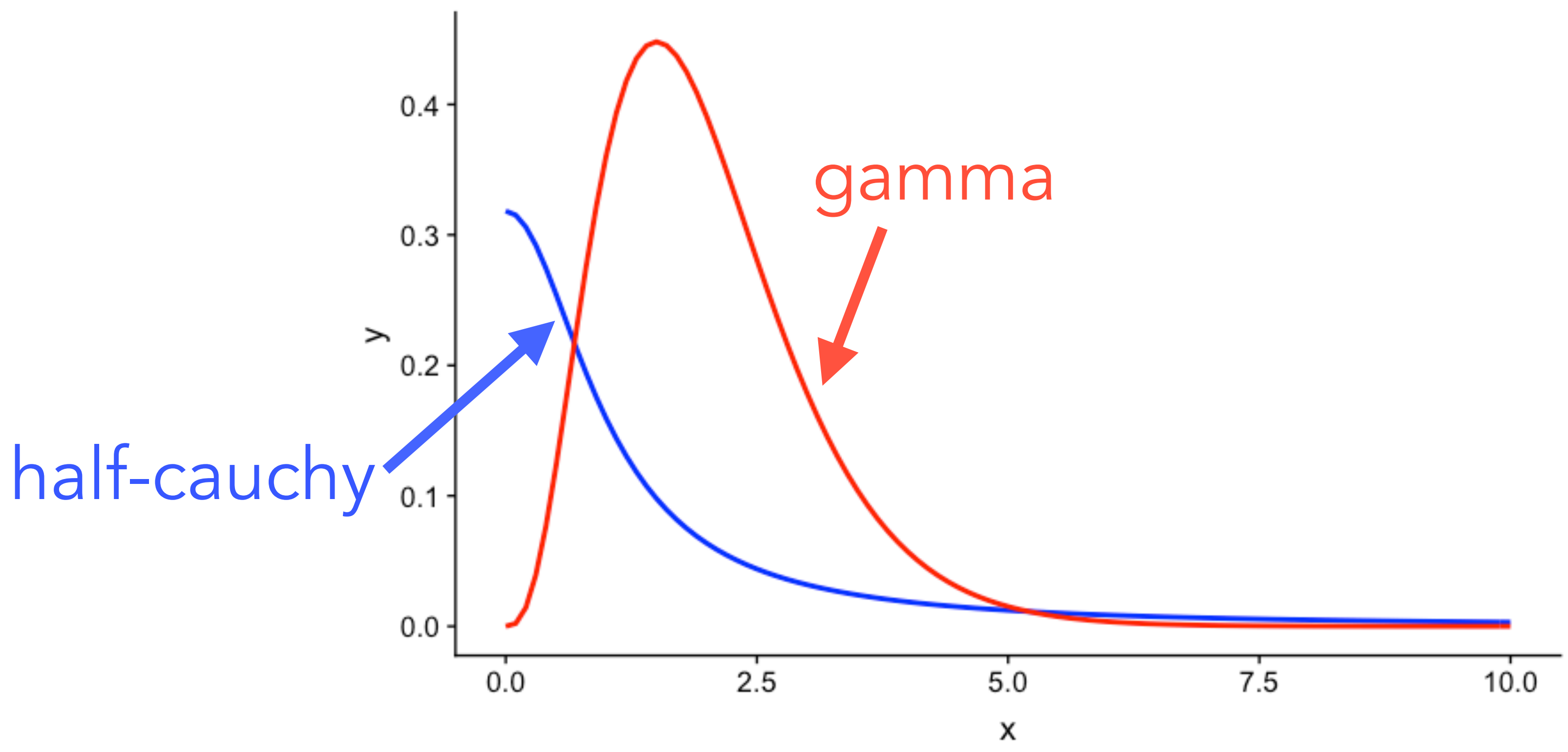
# Prior

- **normal**:
  - continuous
  - unbounded outcomes
  - can range from weakly informative to strongly informative



# Prior

- What prior should we use for inferring the standard deviation?
  - **uniform** (positive)
    - but: large values might be less plausible a priori than smaller values
  - **cauchy** (truncated)
  - **gamma**



# Inference

$$\overbrace{p(H \mid D)}^{\text{Posterior}} = \frac{\overbrace{p(D \mid H)}^{\text{Likelihood}} \cdot \overbrace{p(H)}^{\text{Prior}}}{\underbrace{p(D)}_{\text{Normalizing constant}}}$$

the devil is in the denominator ...

# Doing Bayesian inference

## Discrete hypothesis space

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

sum over all possibilities

## Continuous hypothesis space

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

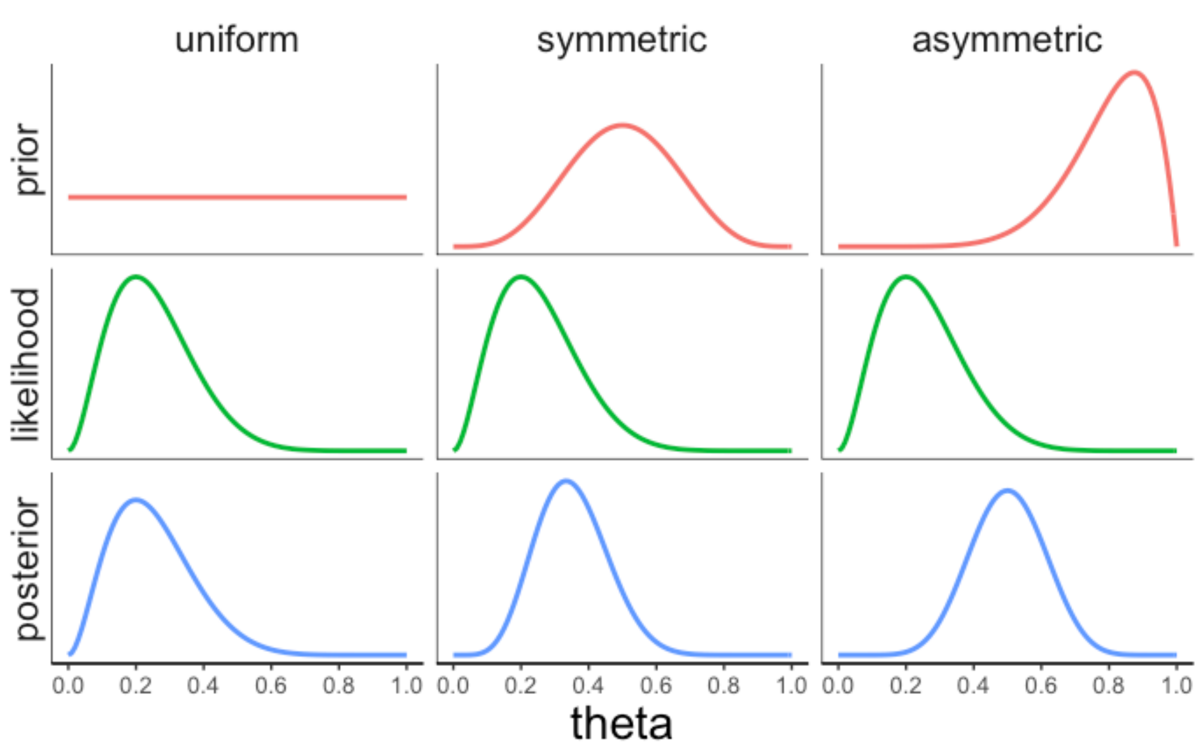
integral over all possibilities

# Discretizing the parameters

```r
# grid
theta = seq(0, 1, 0.01)

# data
data = rep(0:1, c(8, 2))

# calculate posterior
df.prior = tibble(theta = theta,
                  prior_uniform = dbeta(grid, shape1 = 1, shape2 = 1),
                  prior_normal = dbeta(grid, shape1 = 5, shape2 = 5),
                  prior_biased = dbeta(grid, shape1 = 8, shape2 = 2)) %>%
  pivot_longer(cols = -theta,
               names_to = "prior_index",
               values_to = "prior") %>%
  mutate(likelihood = dbinom(sum(data == 1),
                             size = length(data),
                             prob = theta)) %>%
  group_by(prior_index) %>%
  mutate(posterior = likelihood * prior / sum(likelihood * prior)) %>%
  ungroup() %>%
  pivot_longer(cols = -c(theta, prior_index),
               names_to = "index",
               values_to = "value")
```

100 discrete values

**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

# Inference via sampling

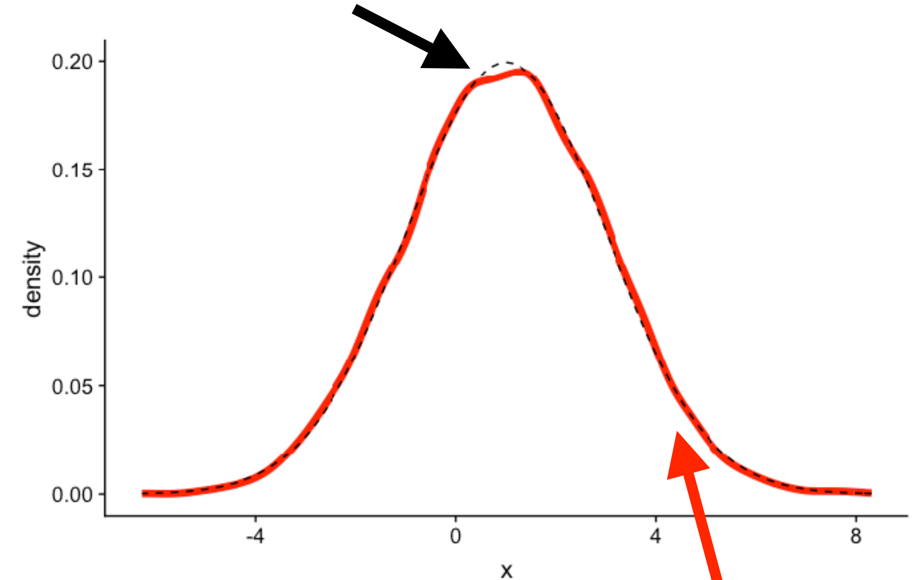
- imagine that we don't have a **`dnorm`**`()` function in R but we want to compute probabilities
- luckily, we do have the **`rnorm`**`()` function, so we can create random samples

# Inference via sampling

```r
# generate samples
df.samples = tibble(x = rnorm(n = 10000, mean = 1, sd = 2))

# visualize distribution
ggplot(data = df.samples,
       mapping = aes(x = x)) +
  stat_density(geom = "line",
               color = "red",
               size = 2) +
  stat_function(fun = "dnorm",
                args = list(mean = 1, sd = 2),
                color = "black",
                linetype = 2)

# calculate probability based on samples
df.samples %>%
  summarize(prob = sum(x >= 0 & x < 4)/n())

# calculate probability based on theoretical distribution
pnorm(4, mean = 1, sd = 2) - pnorm(0, mean = 1, sd = 2)
```

| x |
|---|
| 0.07 |
| 0.10 |
| -0.56 |
| -0.26 |
| 1.59 |
| 0.37 |
| 0.01 |
| 3.68 |
| 0.87 |
| 1.24 |

theoretical distribution

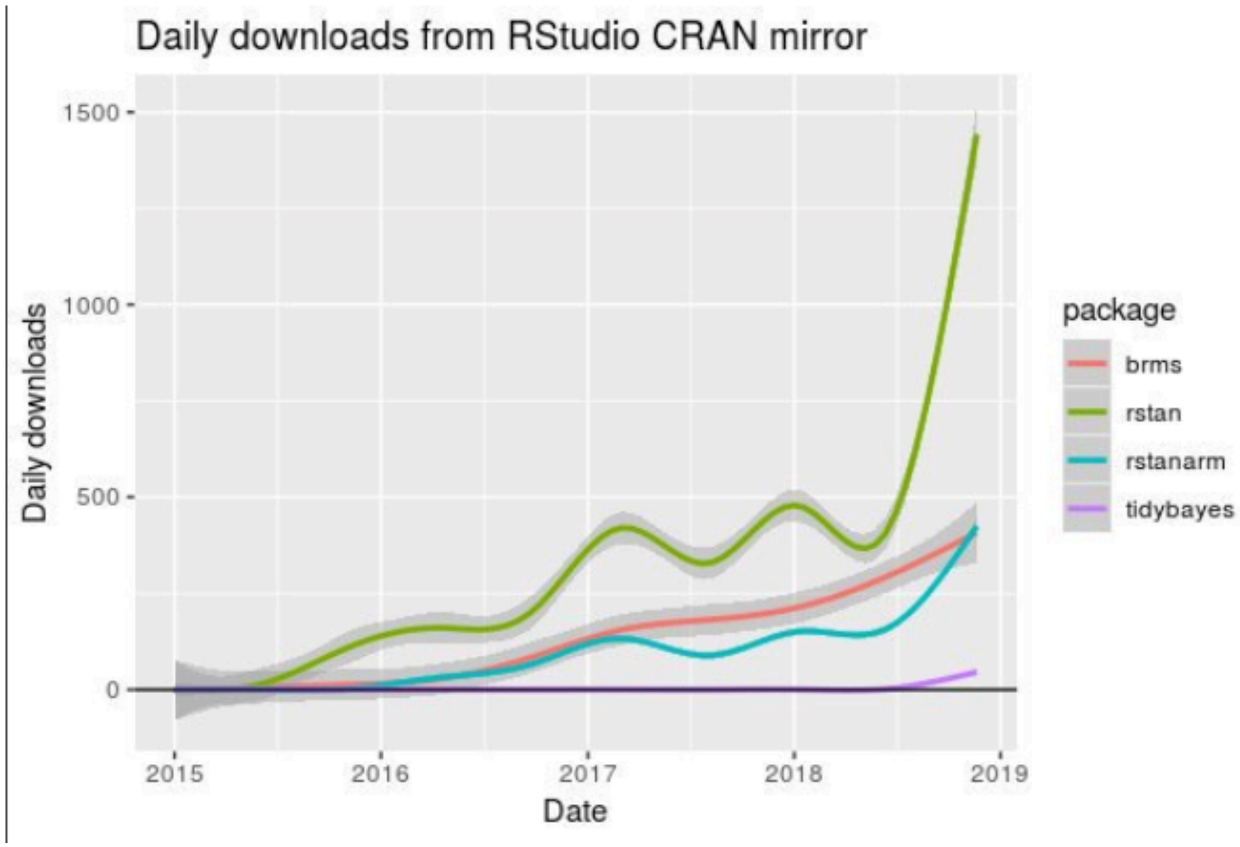



distribution based on samples

both methods yield ≅63%

# 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



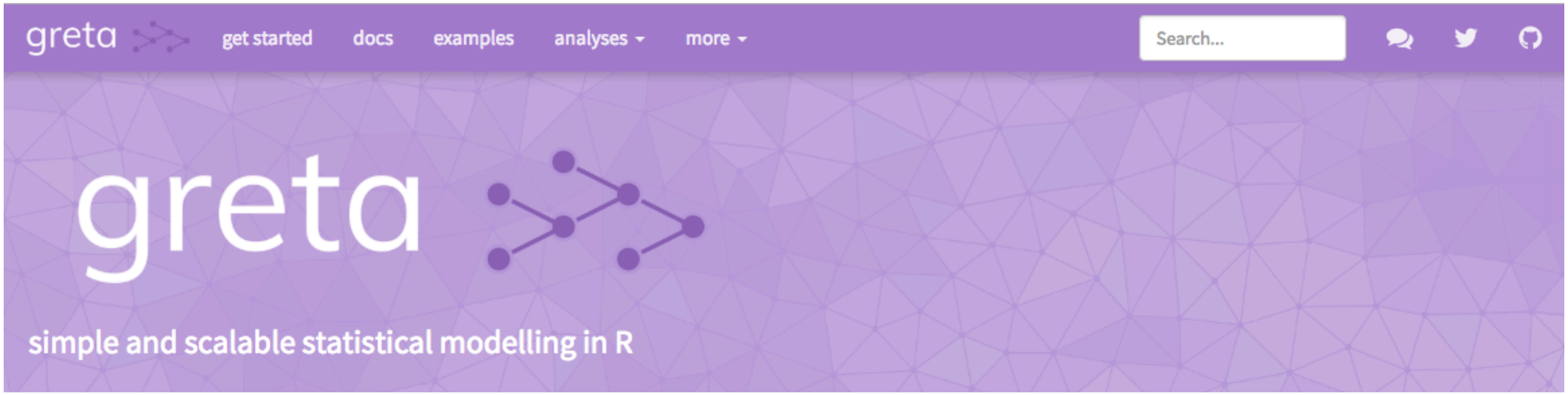
# Plan for today

- Generalized linear model
  - Logistic regression
  - Mixed effects logistic regression
- Bayesian data analysis
  - Comparison between frequentist and Bayesian data analysis
  - Quick flash from the past
  - Flipping coins
  - What affects the posterior?
  - Ingredients: likelihood, prior, inference
  - **Doing Bayesian data analysis**

# Doing Bayesian data analysis

# Software packages

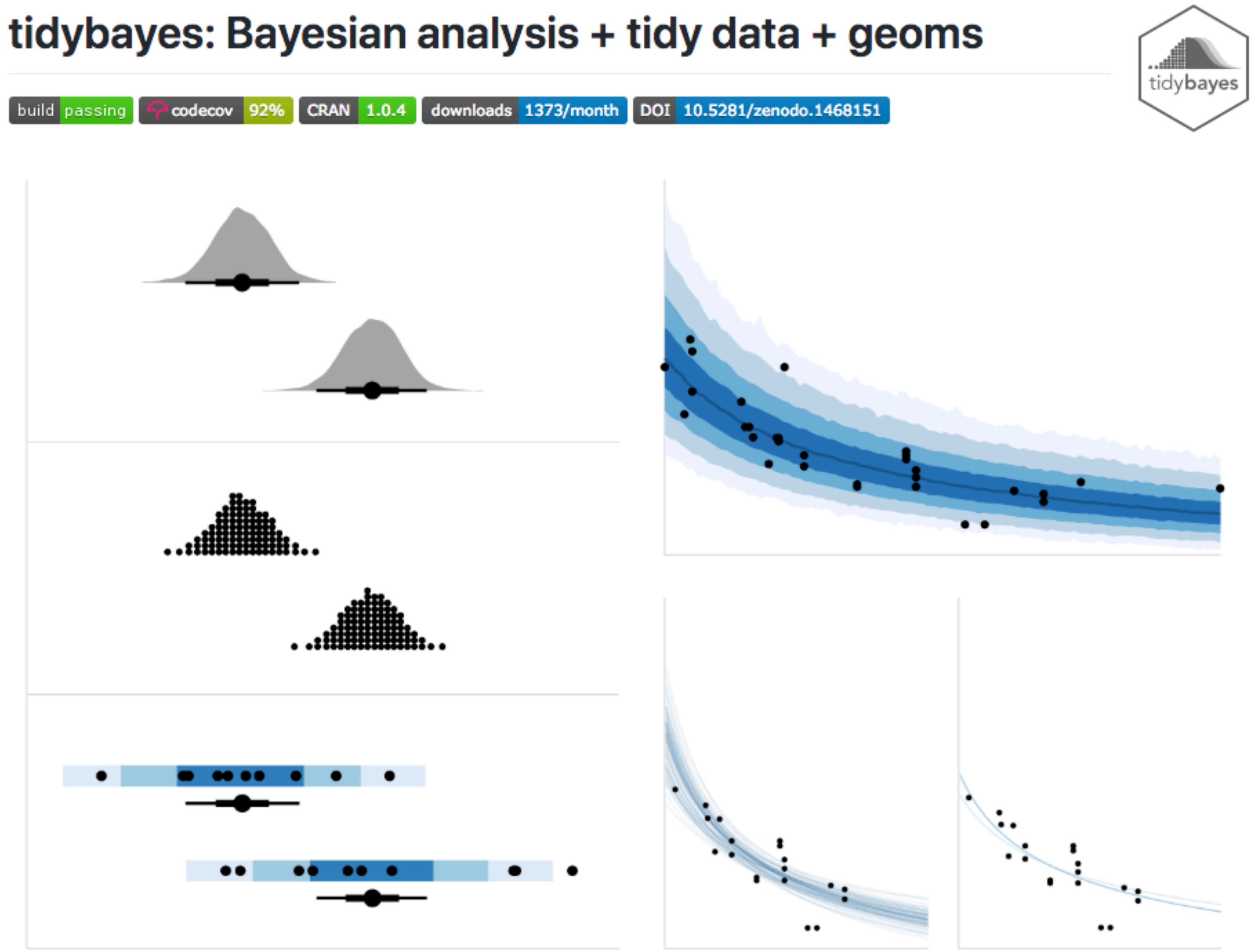
`library("greta")`




- 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 ...)

# Software packages

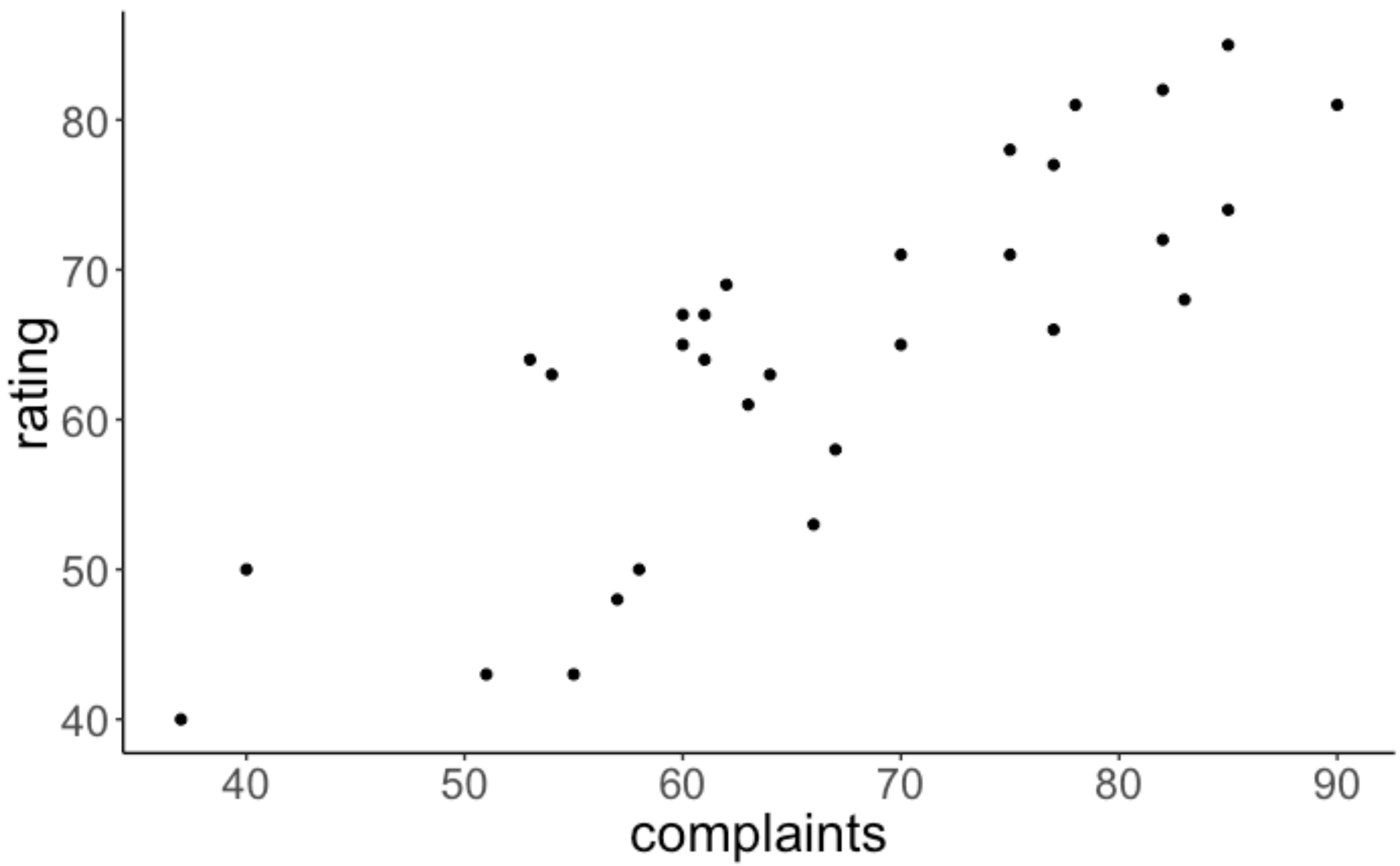
`library("tidybayes")`




- great tool for wrangling and visualizing the results of Bayesian data analysis

# Attitude data set

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



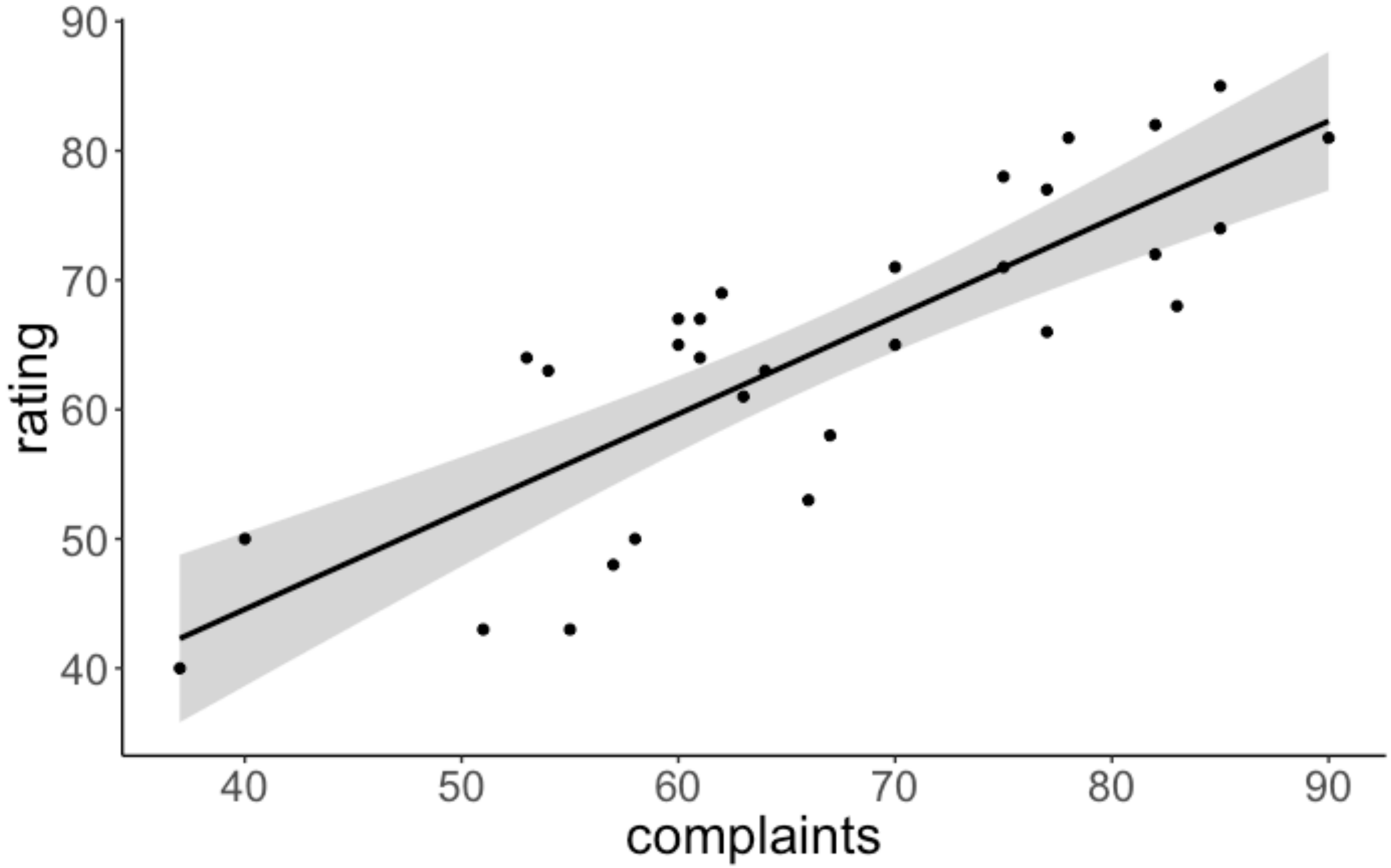
# Frequentist analysis

# Frequentist analysis

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

# print summary
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

```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)
```

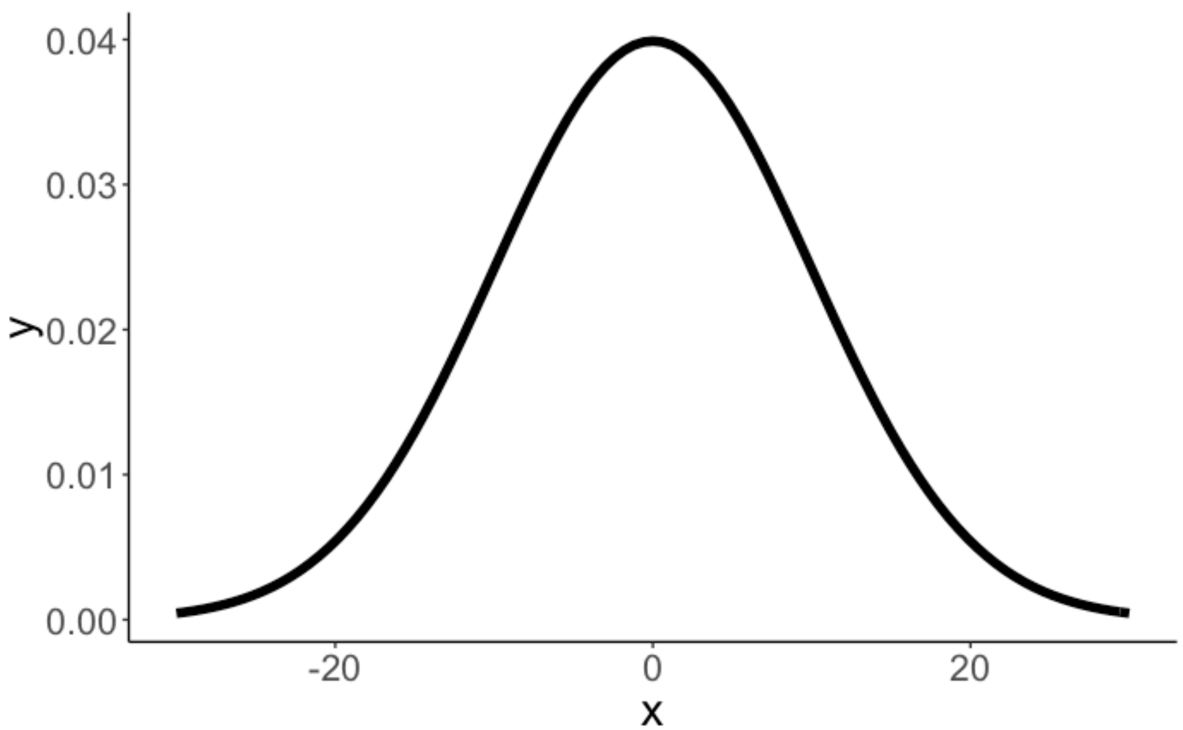
priors

linear combination

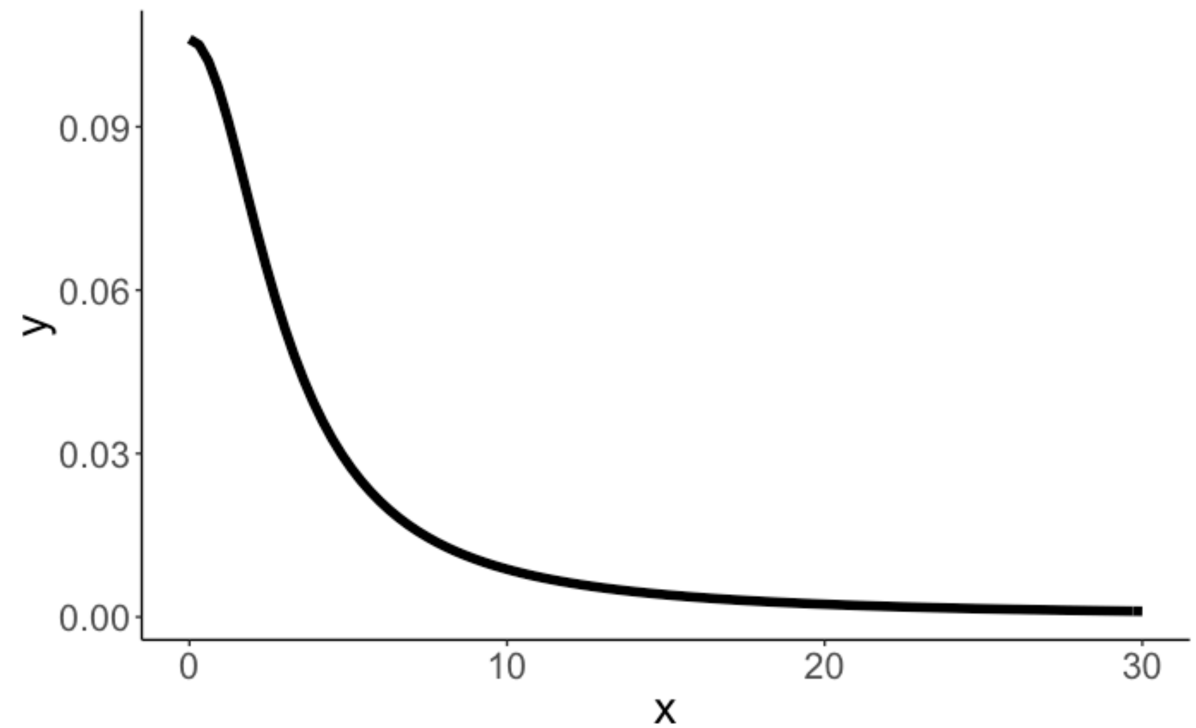
Gaussian likelihood

build the model

# Priors




**Gaussian prior on intercept and coefficient**

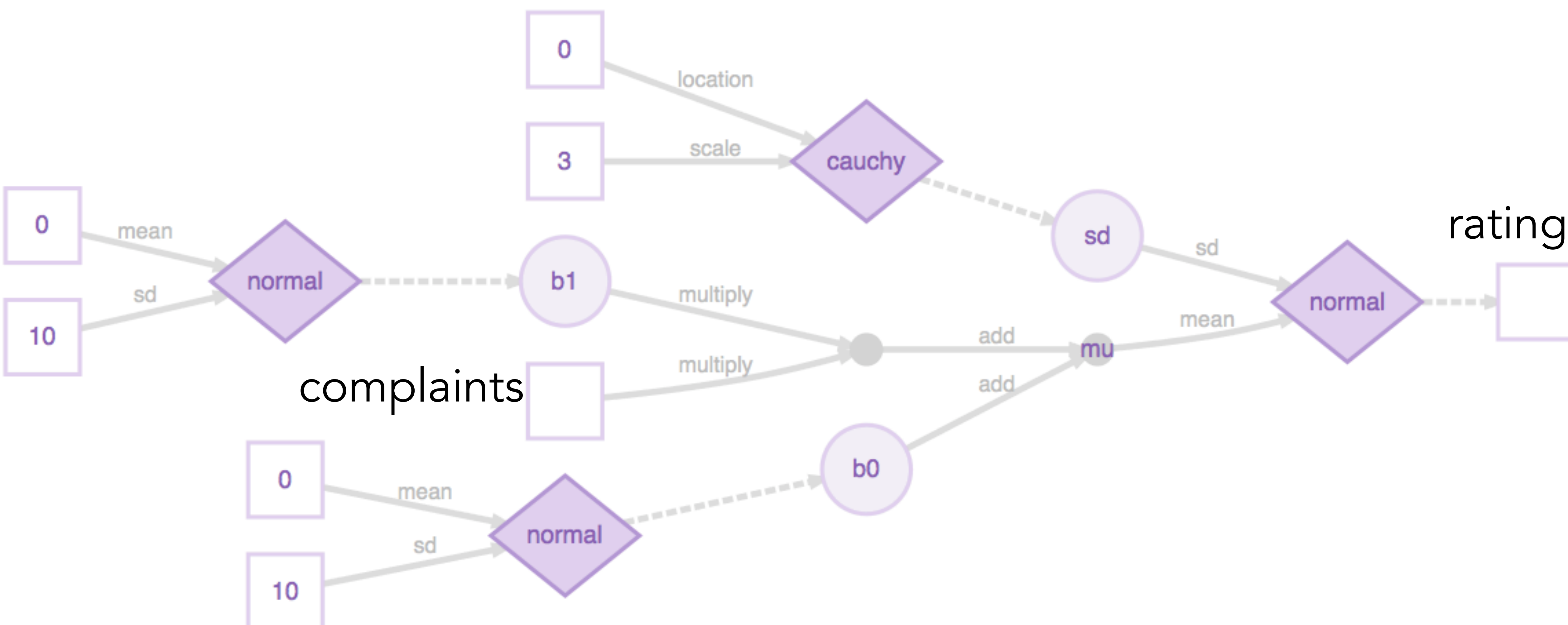



**Truncated Cauchy prior on the standard deviation**

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

# Graphical representation of the model

```
# plotting
plot(m)
```



# Inference via sampling

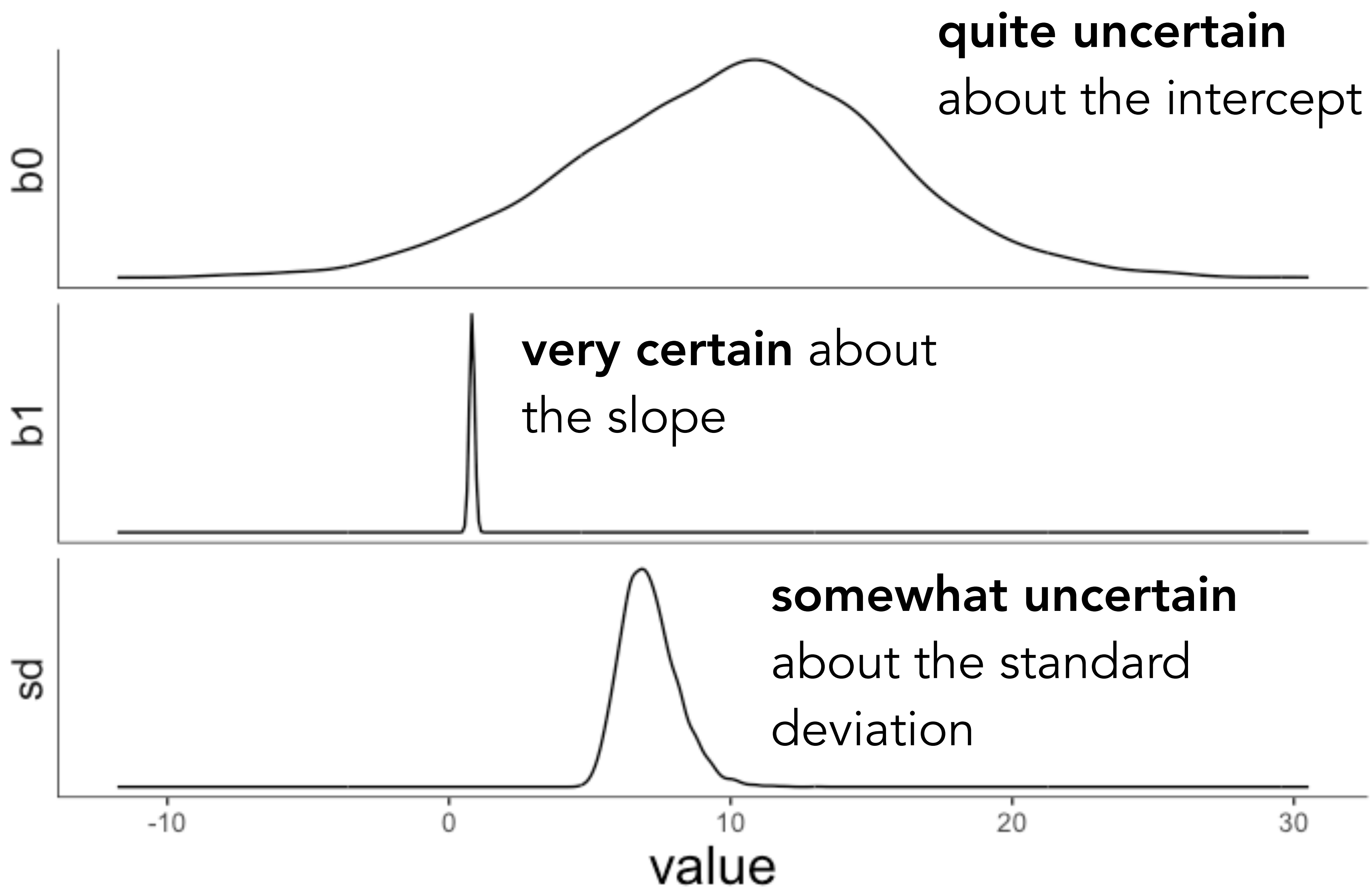
Markov Chain
Monte Carlo
inference

```
# sampling
draws = mcmc(m, n_samples = 1000)

# tidy up the draws
df.draws = tidy_draws(draws) %>%
  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

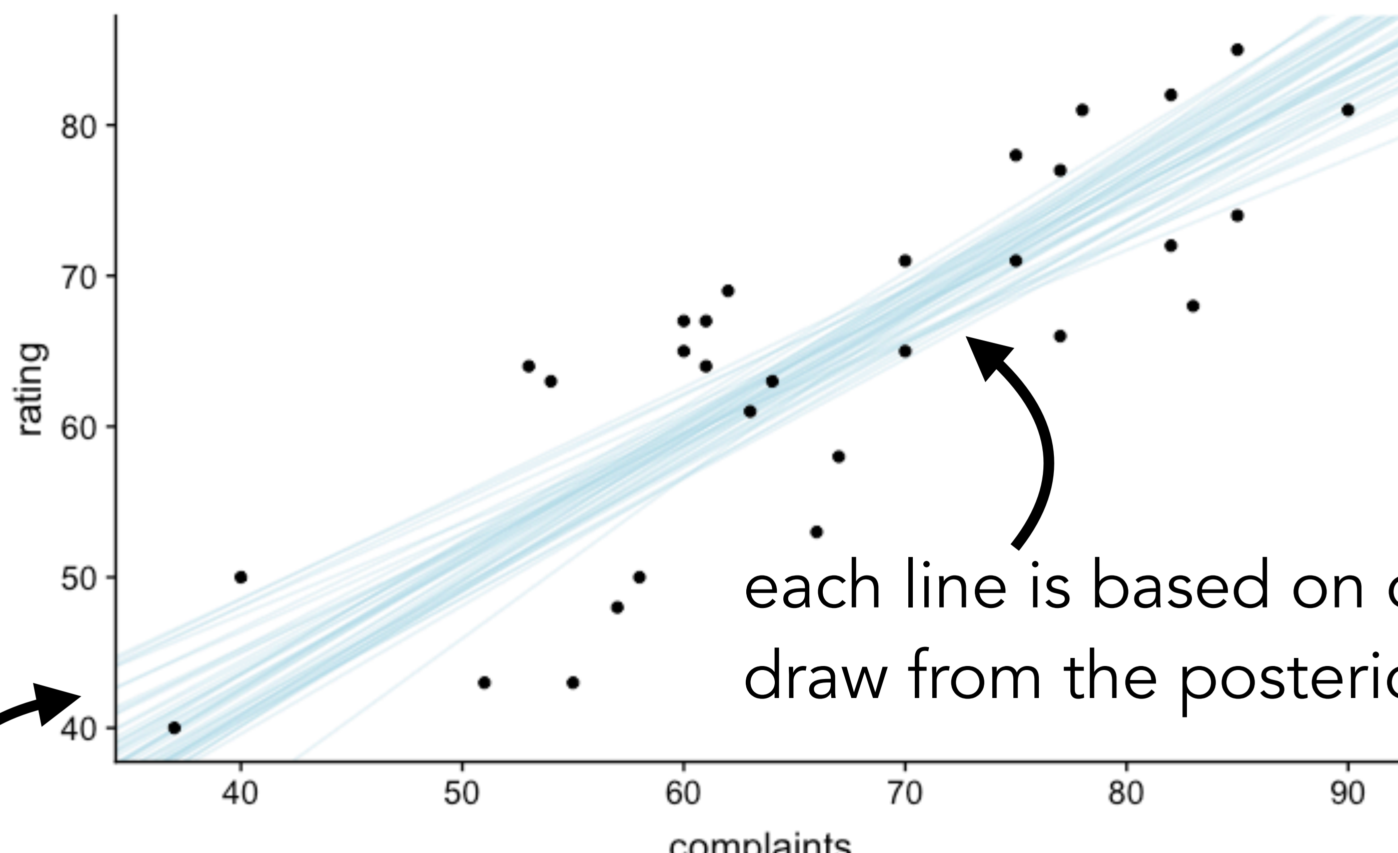
# Visualize the posterior




**Posterior distribution over the three parameters in the model**

# Visualize the model predictions

```r
ggplot(data = df.attitude,
       mapping = aes(x = complaints,
                     y = rating)) +
  geom_abline(data = df.draws %>%
                sample_n(size = 50),
              aes(intercept = b0,
                  slope = b1),
              alpha = 0.3,
              color = "lightblue") +
  geom_point()
```



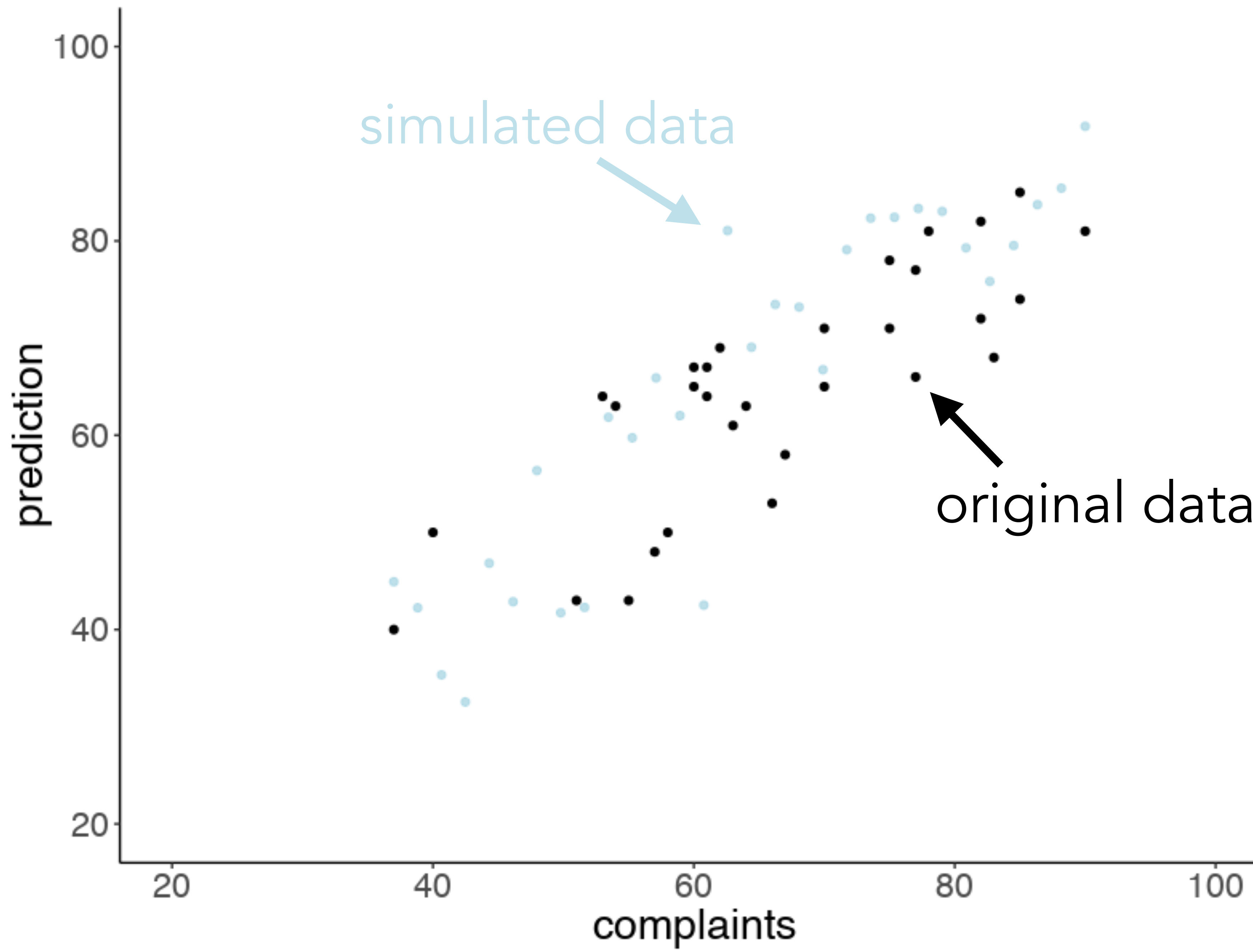

each line is based on one draw from the posterior

explains the "high" uncertainty about the intercept

| 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

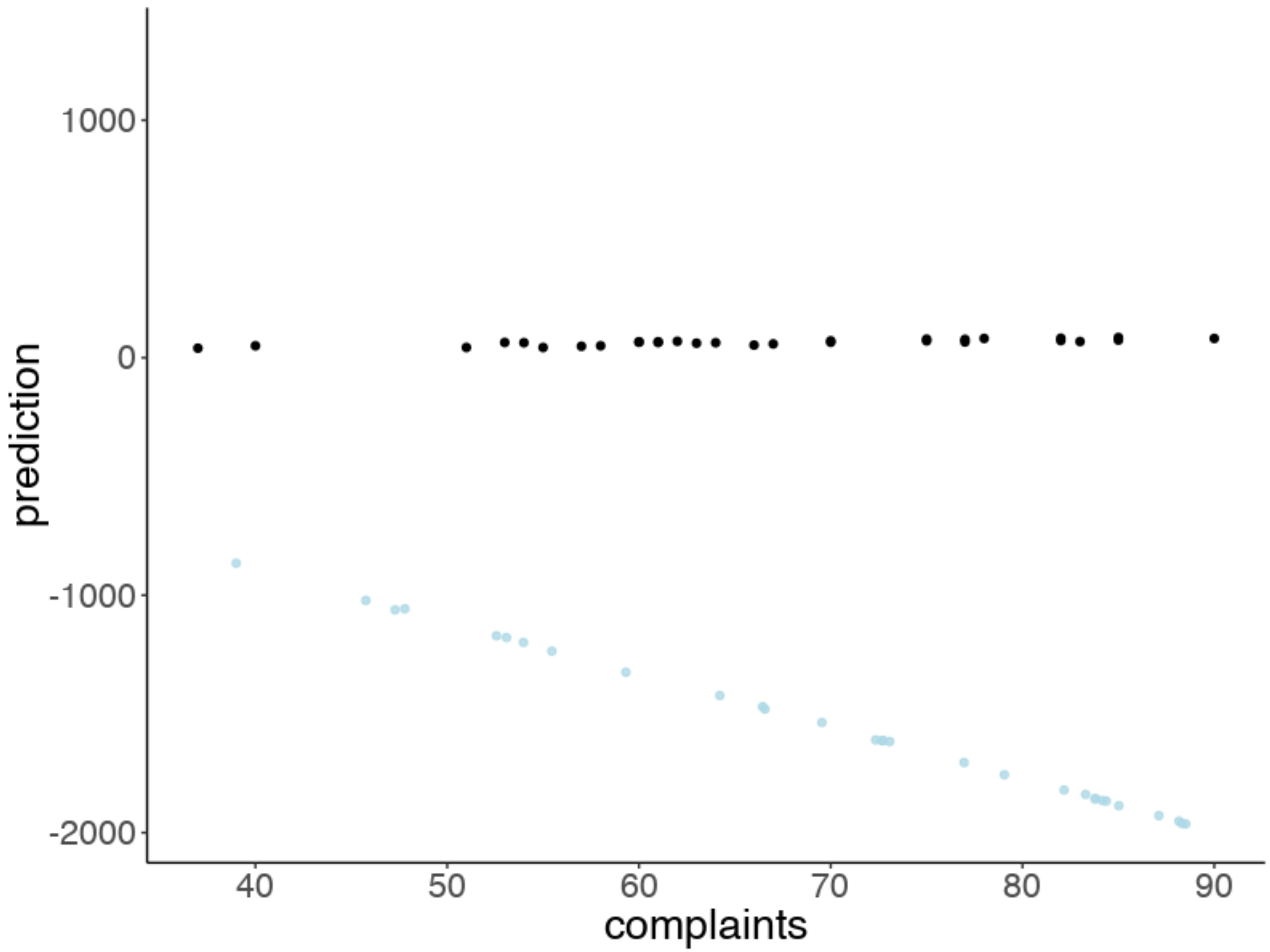
1. sample parameters from the posterior distribution
2. generate data using these parameters (using the likelihood function)




**Hypothetical outcome plot**

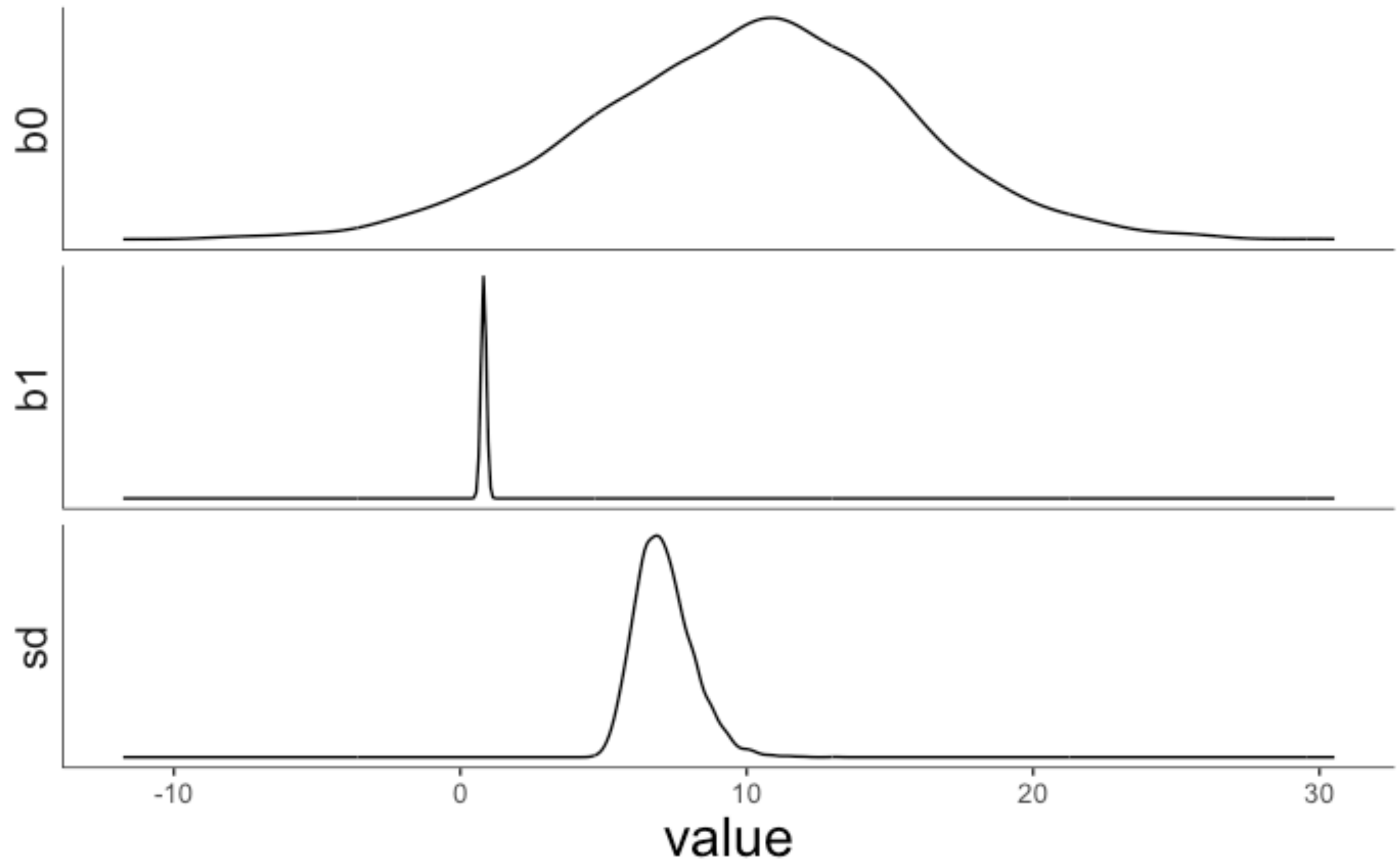
# **Prior** predictive check

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




Hypothetical outcome plot

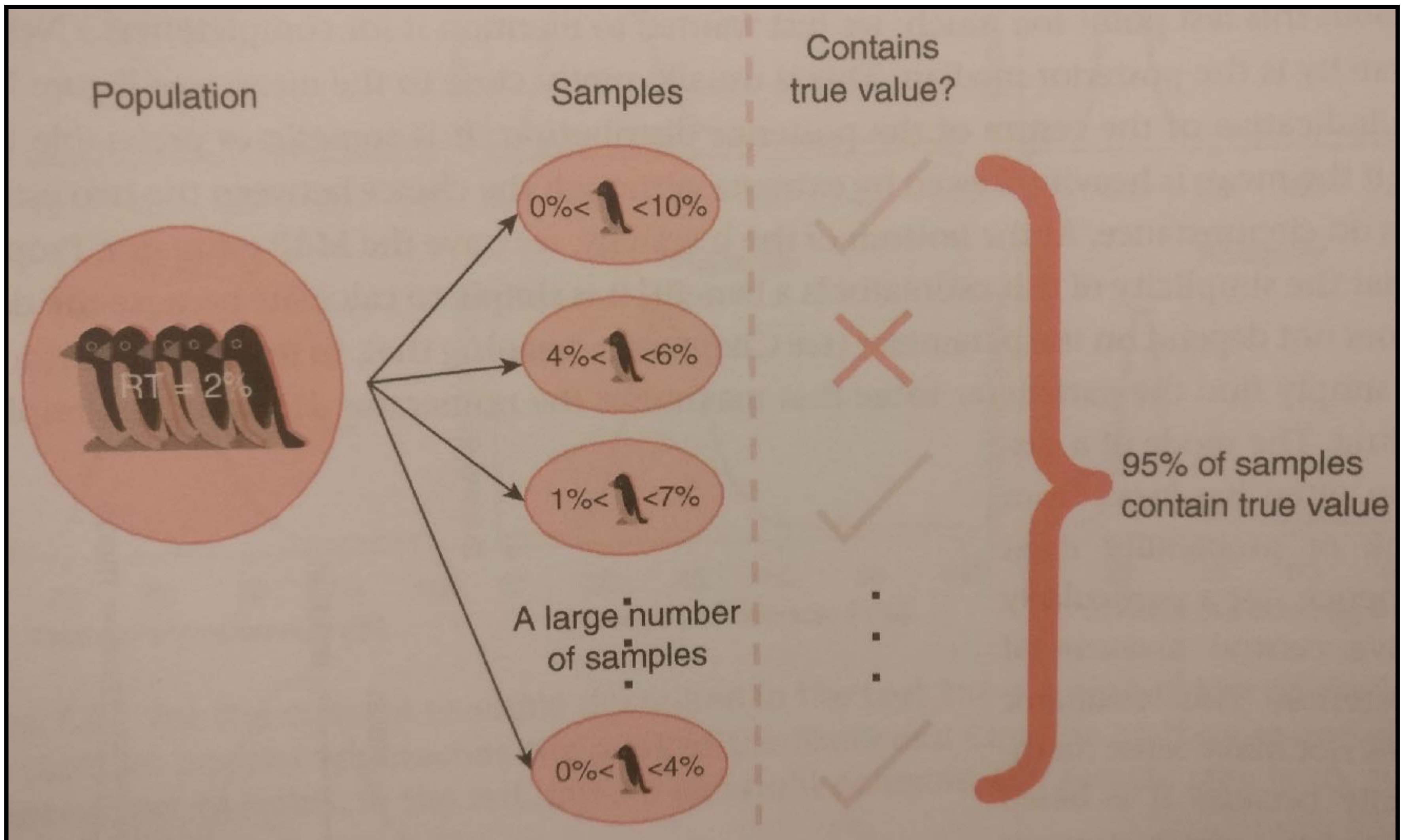
# Summarizing results




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

# 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%.

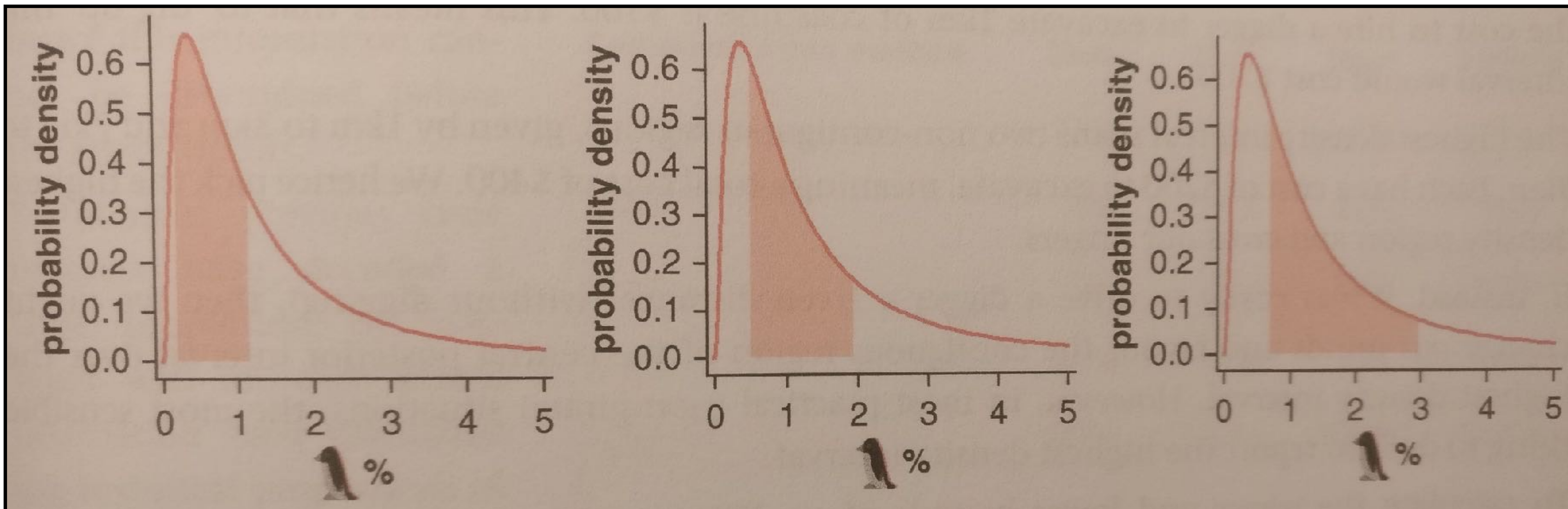
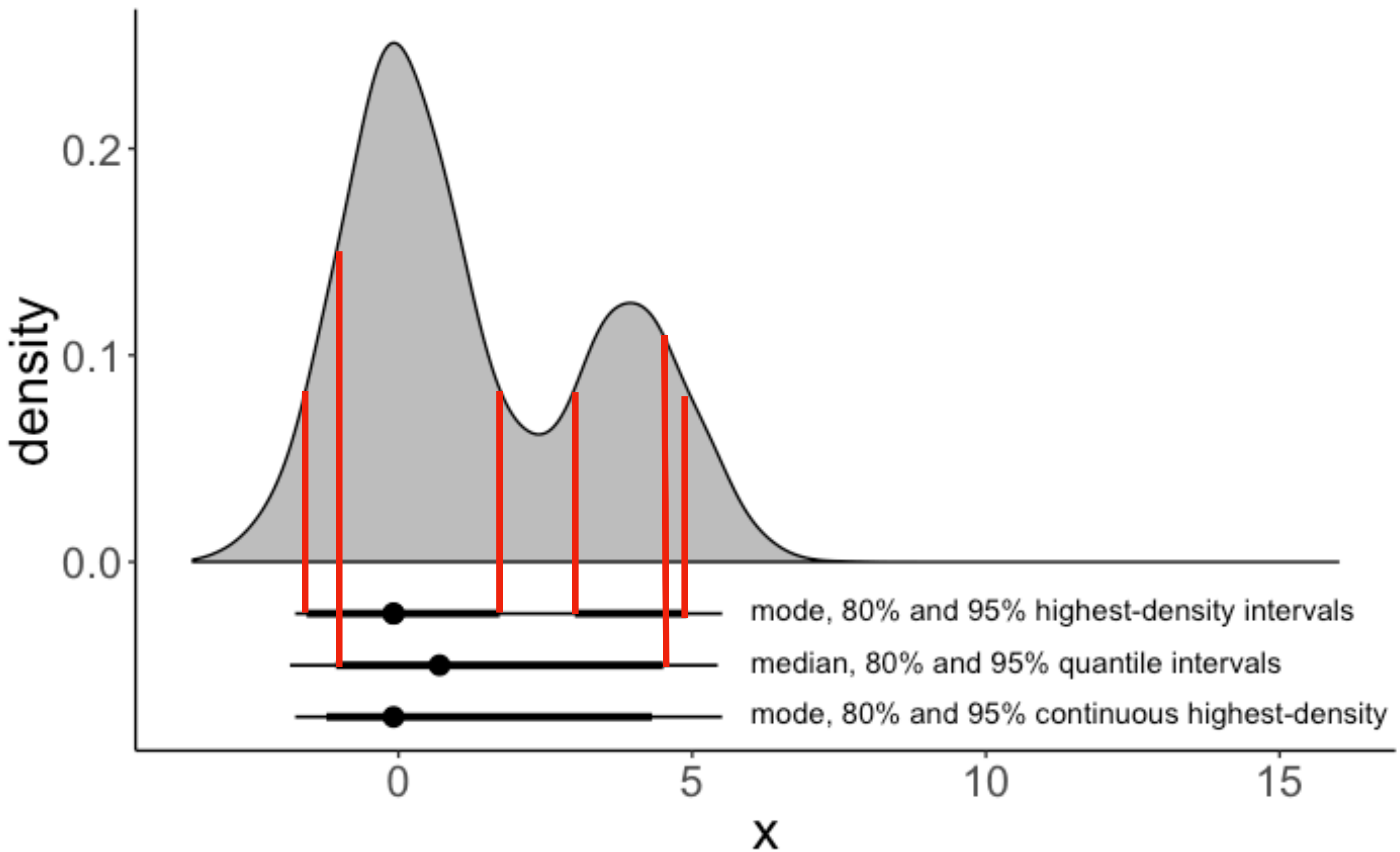



**Figure 7.8** Three examples of 50% credible intervals for a parameter representing the proportion of penguins with red tails.

# Different kinds of credible intervals



# Sensitivity analysis

- How much does the result depend on the choice of priors?
- (I won't do this here, but you will do so in the homework)

# Summary

- Generalized linear model
  - Logistic regression
  - Mixed effects logistic regression
- Bayesian data analysis
  - Comparison between frequentist and Bayesian data analysis
  - Quick flash from the past
  - Flipping coins
  - What affects the posterior?
  - Ingredients: likelihood, prior, inference
  - Doing Bayesian data analysis

# 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!