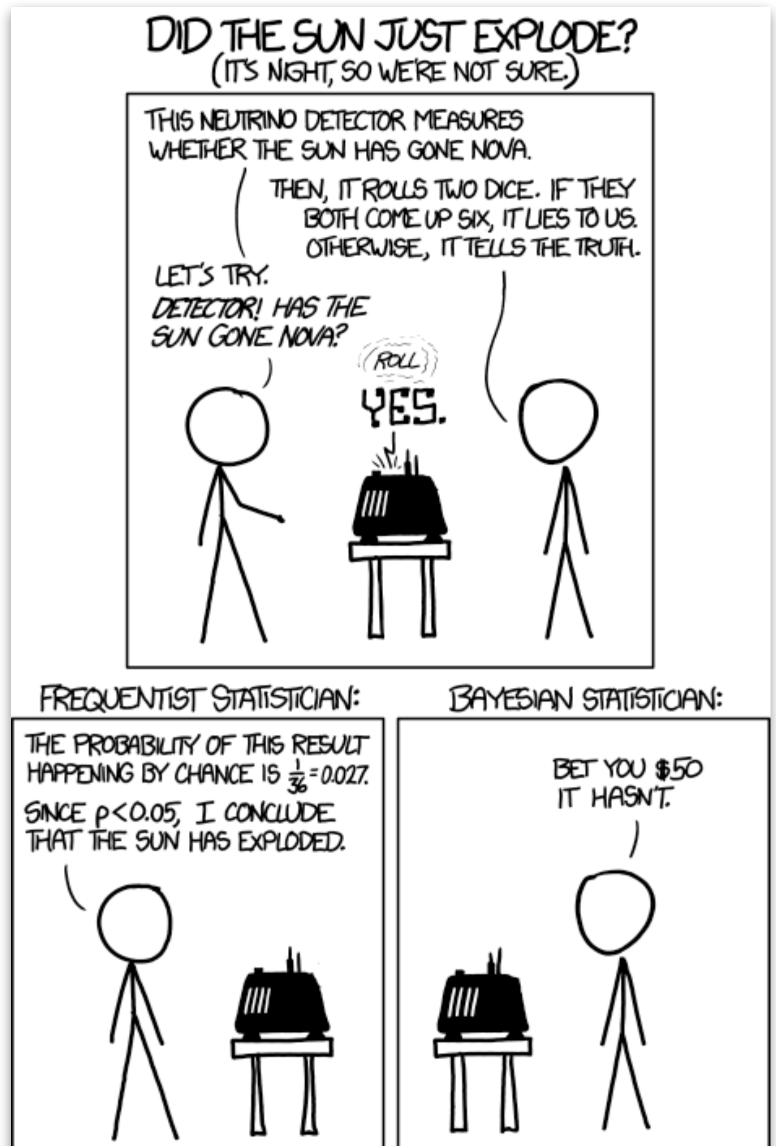
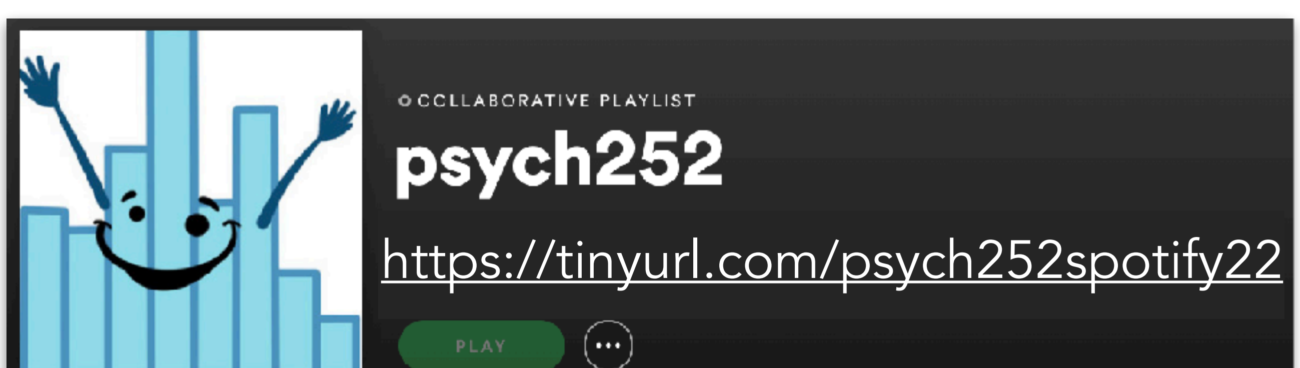
PSYCH 252: Statistical Methods for Behavioral and Social Sciences

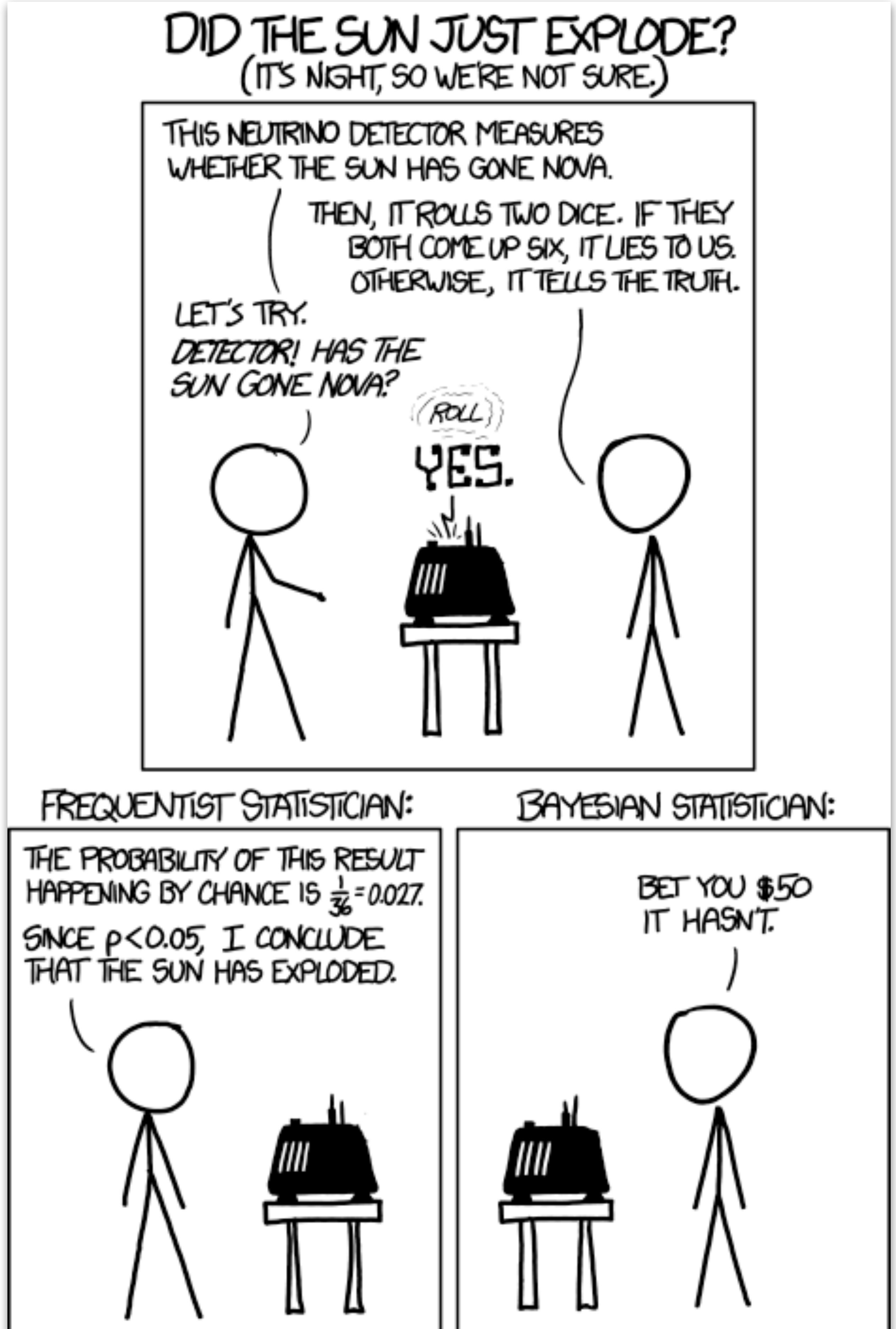
Bayesian data analysis 1





We're listening to "The Silent Orchestra" by "Hamilton Leithauser"

02/28/2022


DID THE SUN JUST EXPLODE?
(IT'S NIGHT, SO WE'RE NOT SURE.)
THIS NEUTRINO DETECTOR MEASURES WHETHER THE SUN HAS GONE NOVA.
THEN, IT ROLLS TWO DICE. IF THEY BOTH COME UP SIX, IT LIES TO US. OTHERWISE, IT TELLS THE TRUTH.
LET'S TRY. DETECTOR! HAS THE SUN GONE NOVA?
ROLL
YES.
FREQUENTIST STATISTICIAN:
THE PROBABILITY OF THIS RESULT HAPPENING BY CHANCE IS $\frac{1}{36}$ = 0.027. SINCE $p<0.05$, I CONCLUDE THAT THE SUN HAS EXPLODED.
BAYESIAN STATISTICIAN:
BET YOU $50 IT HASN'T.

Feedback

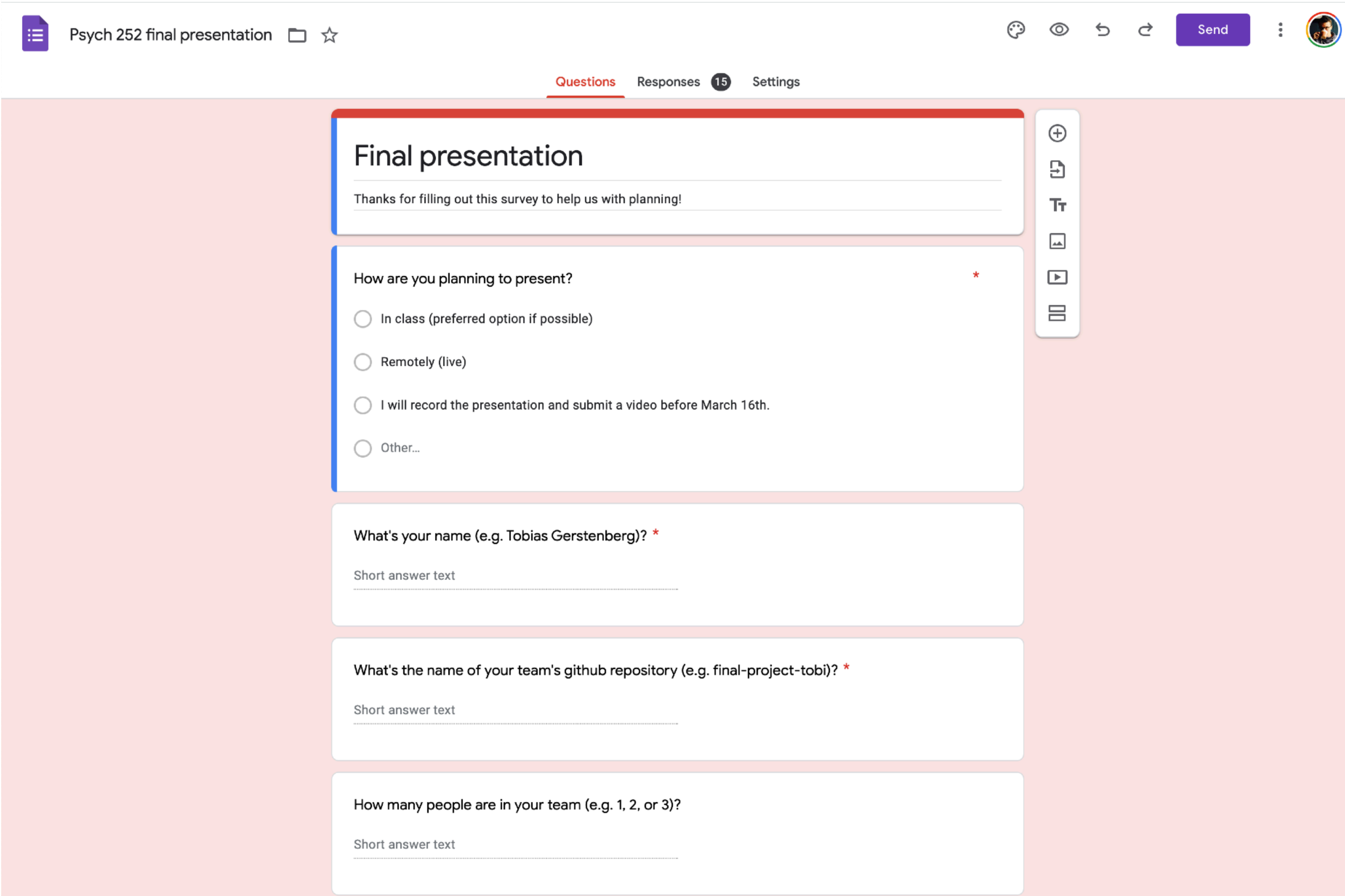
Your feedback

The last part of the lecture felt very rushed. Would you be able to review that part again next time? Thank you.

Logistics

Final presentations

29 students enrolled



https://forms.gle/Kz83x77cmLDxqZ8d6

Plan for today

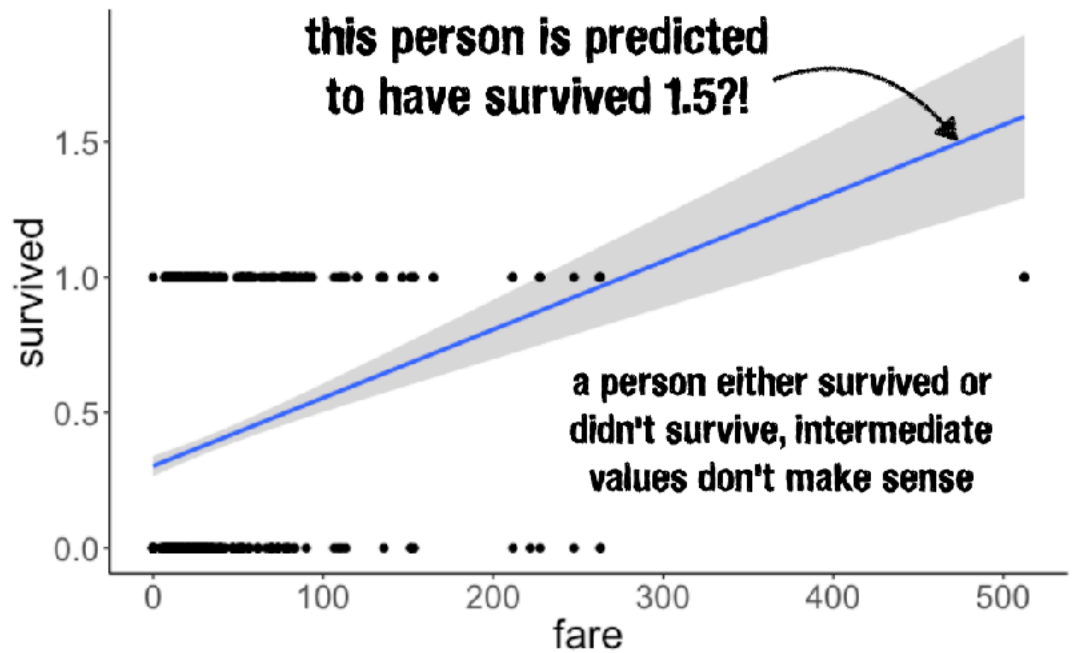
- Quick recap: Generalized linear model
- Bayesian data analysis
  - Comparison between frequentist and Bayesian data analysis
  - Quick flash from the past
  - Flipping coins
  - What affects the posterior?
  - Ingredients: likelihood, prior, inference
  - Doing Bayesian data analysis

Quick recap

Quick recap: Logistic regression

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

```
fit.lm = lm(formula = survived ~ 1 + fare, data = df.titanic)
```



22

Logit transform

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

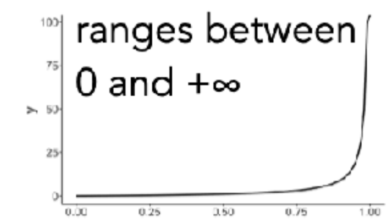
predict the probability of Y

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

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)

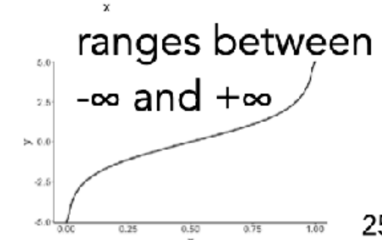
Step 1: Calculate the "odds"

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



Step 2: Take the (natural) log

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



25

Fitting a logistic regression in R

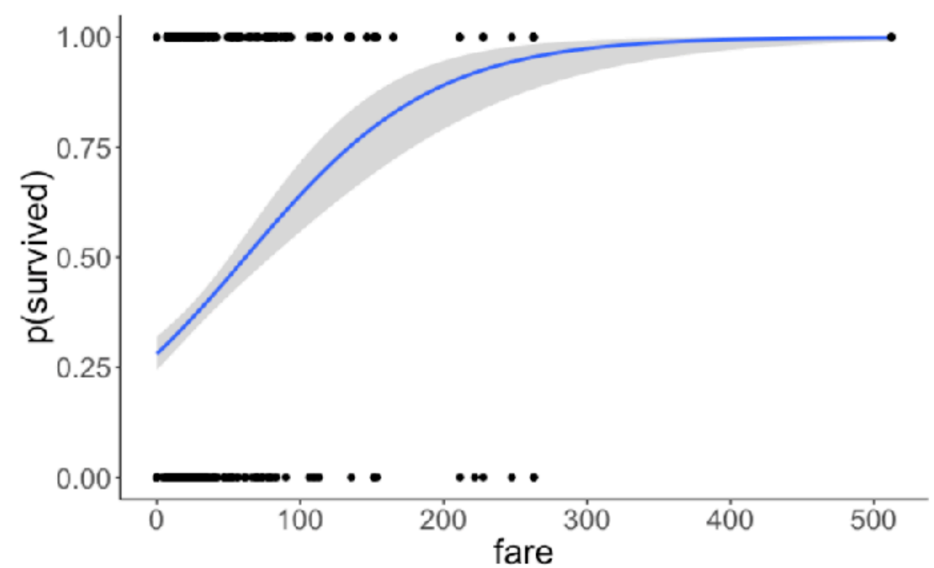
```
fit.glm = glm(formula = survived ~ 1 + fare,
              family = "binomial",
              data = df.titanic)

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

```
Call:
glm(formula = survived ~ 1 + fare, family = "binomial", data = df.titanic)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4906  -0.8878  -0.8531   1.3429   1.5942

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.941330   0.095129  -9.895  < 2e-16 ***
fare         0.015197   0.002232   6.810 9.79e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1186.7  on 890  degrees of freedom
Residual deviance: 1117.6  on 889  degrees of freedom
AIC: 1121.6

Number of Fisher Scoring iterations: 4
```

27

Visualize the model's predictions

```
ggplot(data = df.titanic,
       mapping = aes(x = fare,
                     y = survived)) +
  geom_smooth(method = "glm",
              method.args = list(family = "binomial")) +
  geom_point() +
  labs(y = "p(survived)")
```

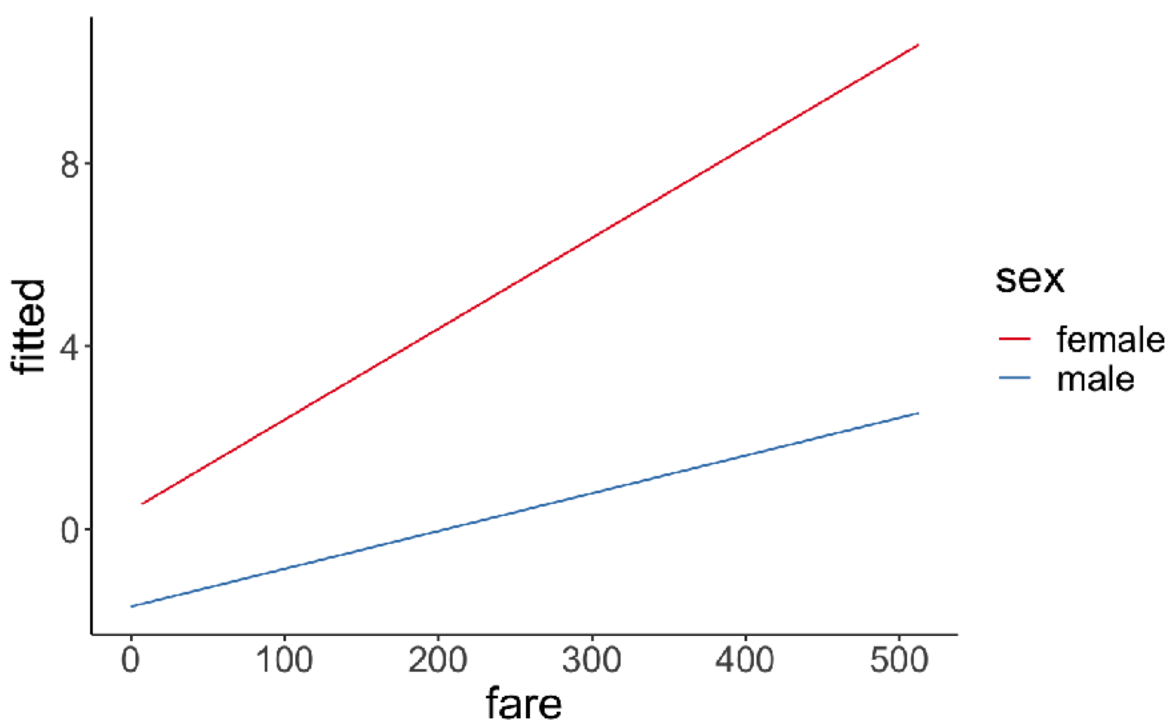


28

Quick recap: Logistic regression

```
fit.glm3 = glm(formula = survived ~ 1 + sex * fare,
               family = "binomial",
               data = df.titanic)

fit.glm3 %>%
  summary()
```

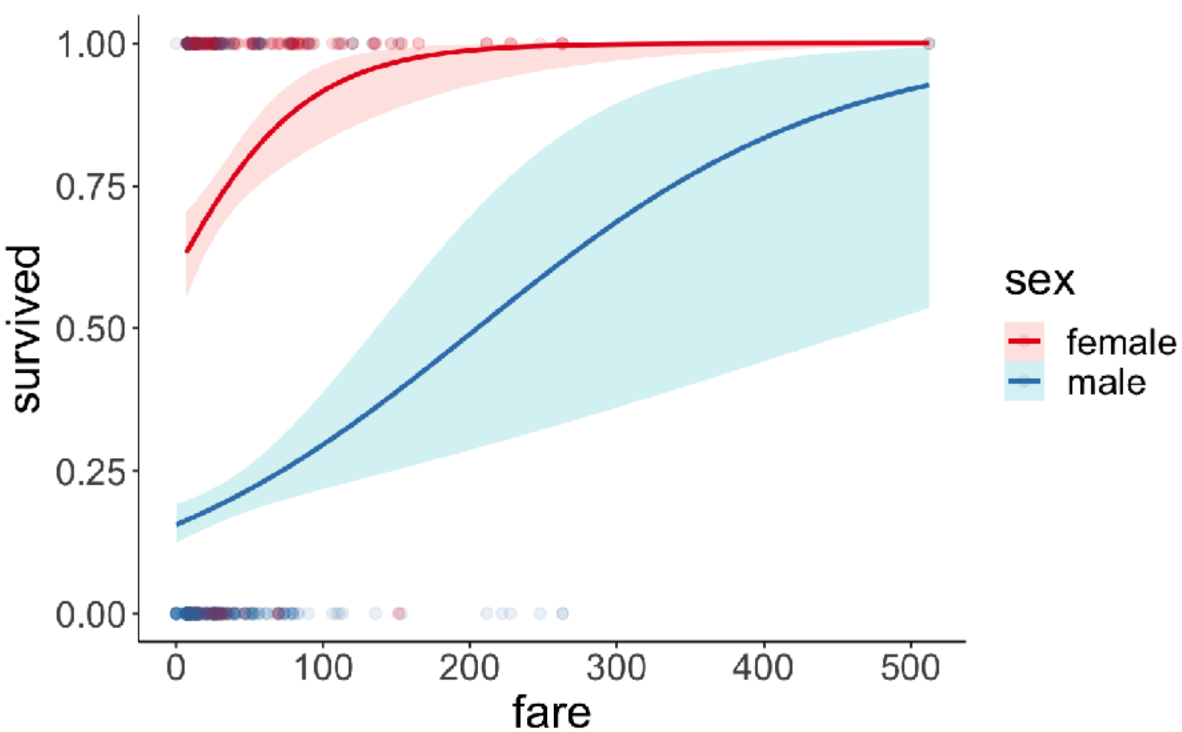
```
Call:
glm(formula = survived ~ 1 + sex * fare, family = "binomial",
    data = df.titanic)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6280  -0.6279  -0.5991   0.8172   1.9288

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.408428   0.189999   2.150 0.031584 *
sexmale      -2.099345   0.230291  -9.116  < 2e-16 ***
fare          0.019878   0.005372   3.701 0.000215 ***
sexmale:fare -0.011617   0.005934  -1.958 0.050252 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1186.66  on 890  degrees of freedom
Residual deviance:  879.85  on 887  degrees of freedom
AIC: 887.85

Number of Fisher Scoring iterations: 5
```



lines in log odds space

Quick recap: Logistic regression

```
fit.glm3 = glm(formula = survived ~ 1 + sex * fare,
               family = "binomial",
               data = df.titanic)

fit.glm3 %>%
  summary()
```



after inverse logit transformation

```
Call:
glm(formula = survived ~ 1 + sex * fare, family = "binomial",
    data = df.titanic)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6280  -0.6279  -0.5991   0.8172   1.9288

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   0.408428   0.189999   2.150 0.031584 *
sexmale      -2.099345   0.230291  -9.116  < 2e-16 ***
fare          0.019878   0.005372   3.701 0.000215 ***
sexmale:fare -0.011617   0.005934  -1.958 0.050252 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1186.66  on 890  degrees of freedom
Residual deviance:  879.85  on 887  degrees of freedom
AIC: 887.85

Number of Fisher Scoring iterations: 5
```

Quick recap: Interpreting the model output

Let's consider a binary predictor

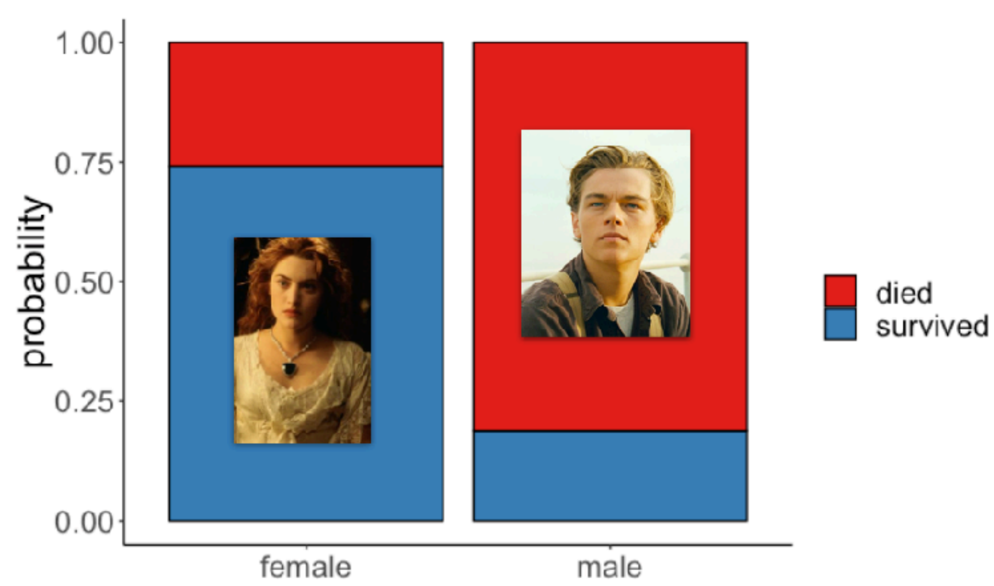
```r
fit.glm2 = glm(formula = survived ~ sex,
               family = "binomial",
               data = df.titanic)

fit.glm2 %>% summary()
```

```
Call:
glm(formula = survived ~ sex, family = "binomial", data = df.titanic)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6462  -0.6471  -0.6471   0.7725   1.8256

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.0566     0.1290   8.191 2.58e-16 ***
sexmale      -2.5137     0.1672 -15.036  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1186.7  on 890  degrees of freedom
Residual deviance:  917.8  on 889  degrees of freedom
AIC: 921.8

Number of Fisher Scoring iterations: 4
```

sex was significantly associated with survival

35

Was the probability of survival different between female and male passengers on the Titanic?



Let's consider a binary predictor

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

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.0566     0.1290   8.191 2.58e-16 ***
sexmale      -2.5137     0.1672 -15.036  < 2e-16 ***
```

sex	survived	n	p	p(survived I sex)
female	0	81	0.09	0.26
female	1	233	0.26	0.74
male	0	468	0.53	0.81
male	1	109	0.12	0.19

if sex == 0:

$$\ln\left(\frac{\widehat{\text{p(survived)}}_i}{1-\text{p(survived)}_i}\right) = b_0$$

$$\text{p(survived)}_i = \frac{e^{b_0}}{1+e^{b_0}} = 0.74$$

36

Now let's go back to a continuous predictor

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

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.941330   0.095129  -9.895  < 2e-16 ***
fare         0.015197   0.002232   6.810 9.79e-12 ***
```

fare	prediction	p(survival)
0	-0.94	0.28
10	-0.79	0.31
50	-0.18	0.45
100	0.58	0.64
500	6.66	1.00

$$\ln\left(\frac{\widehat{\text{p(survived)}}}{1-\text{p(survived)}}\right) = -0.94 + 0.015 \cdot 10$$

$$\text{p(survived)}_i = \frac{e^{-0.94+0.015\cdot 10}}{1+e^{-0.94+0.015\cdot 10}} = 0.31$$

38

Quick recap: Fitting and reporting models

Simulating a logistic regression

```r
# make example reproducible
set.seed(1)

# set parameters
sample_size = 1000
b0 = 0
b1 = 1

# generate data
df.data = tibble(
  x = rnorm(n = sample_size),
  y = b0 + b1 * x,
  p = inv.logit(y)) %>%
  mutate(response = rbinom(n(), size = 1, p = p))

# fit model
fit = glm(formula = response ~ 1 + x,
          family = "binomial",
          data = df.data)

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

set some parameters

linear model (y is in log odds)

transform into probability

randomly draw response

fit a logistic regression

summarize the result

45

Assessing the model fit

actual value

predicted value

$$\text{log-likelihood} = \sum_{i=1}^{n} [Y_i \cdot \ln(P(Y_i)) + (1 - Y_i) \cdot \ln(1 - P(Y_i))]$$

- calculate the probability of the observed response
- take the log of these probabilities
- sum them up to get the log-likelihood of the data (given the model)

response	p(Y = 1)	p(Y = response)	log(p(Y = response))
1	0.34	0.34	-1.07
0	0.53	0.47	-0.75
1	0.30	0.30	-1.20
1	0.81	0.81	-0.22
1	0.56	0.56	-0.58
0	0.30	0.70	-0.36
1	0.60	0.60	-0.52
1	0.65	0.65	-0.43
1	0.62	0.62	-0.48
0	0.41	0.59	-0.54

47

Testing hypotheses

`joint_tests()`

```r
glm(formula = survived ~ 1 + sex + fare,
  family = "binomial",
  data = df.titanic) %>%
  joint_tests()
```

```
 model term df1 df2 F.ratio p.value
 sex          1 Inf 201.881  <.0001
 fare         1 Inf  23.869  <.0001
```

not quite the same value

`anova(test = "LRT")`

```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")
```

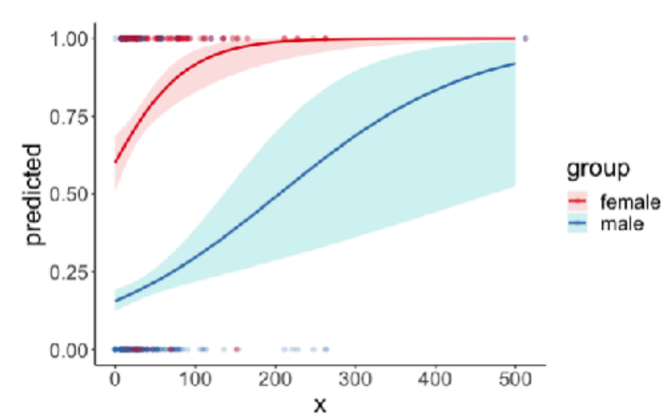
```
Analysis of Deviance Table

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
```

https://stats.stackexchange.com/questions/400101/using-emmeans-with-clmm-to-look-at-joint-effects

51

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 | 0.13 | [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 | [0.98, 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 | [0.28, 0.70]
300 |      0.69 | 0.63 | [0.35, 0.90]
400 |      0.83 | 0.94 | [0.44, 0.97]
500 |      0.92 | 1.13 | [0.53, 0.99]
```

54

Quick recap: Mixed effects logistic regression

Mixed effects logistic regression

repeated a grade: yes / no

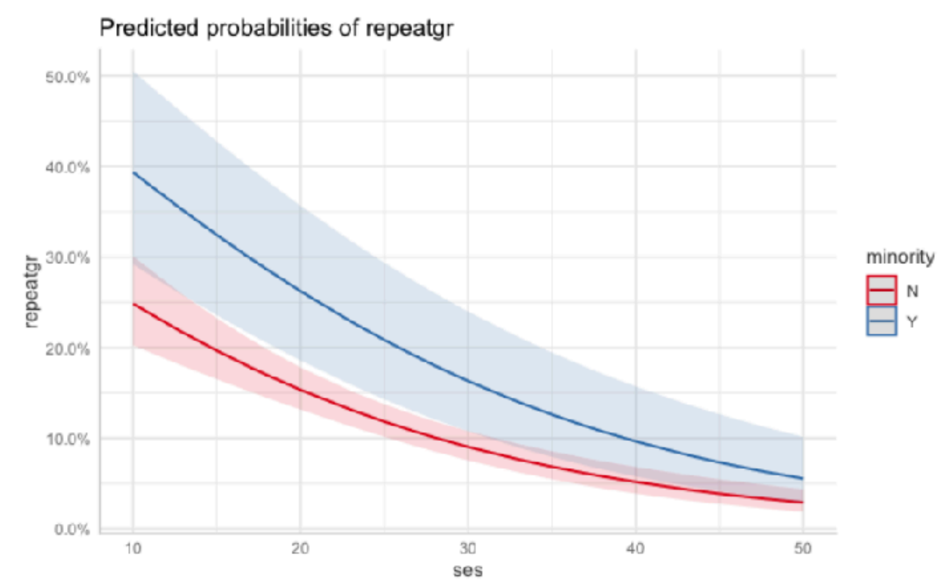
```r
fit =  glmer(repeatgr ~ 1 + ses * Minority + (1 | schoolNR),
              data = df.language,
              family = "binomial")

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
```

58

Mixed effects logistic regression

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



59

Hypothesis test: `joint_tests()`

```r
glmer(formula = repeatgr ~ 1 + ses + minority + (1 | school_nr),
      data = df.language,
      family = "binomial") %>%
  joint_tests()
```

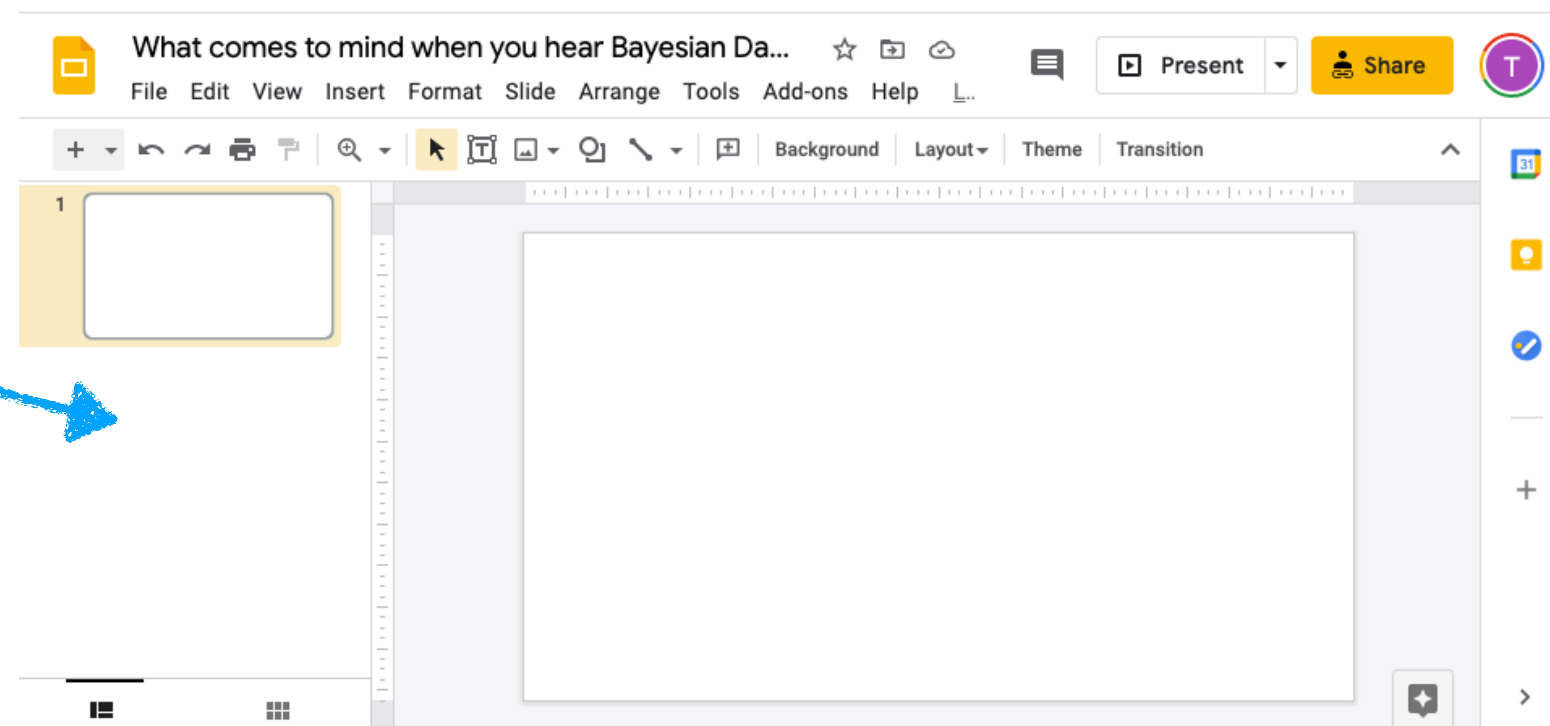
model term	df1	df2	F.ratio	p.value
ses	1	Inf	63.784	<.0001
minority	1	Inf	7.967	0.0048

Bayesian Data Analysis

Breakout rooms

Tasks:

What comes to mind when you hear Bayesian Data Analysis?



Size: ~4 people

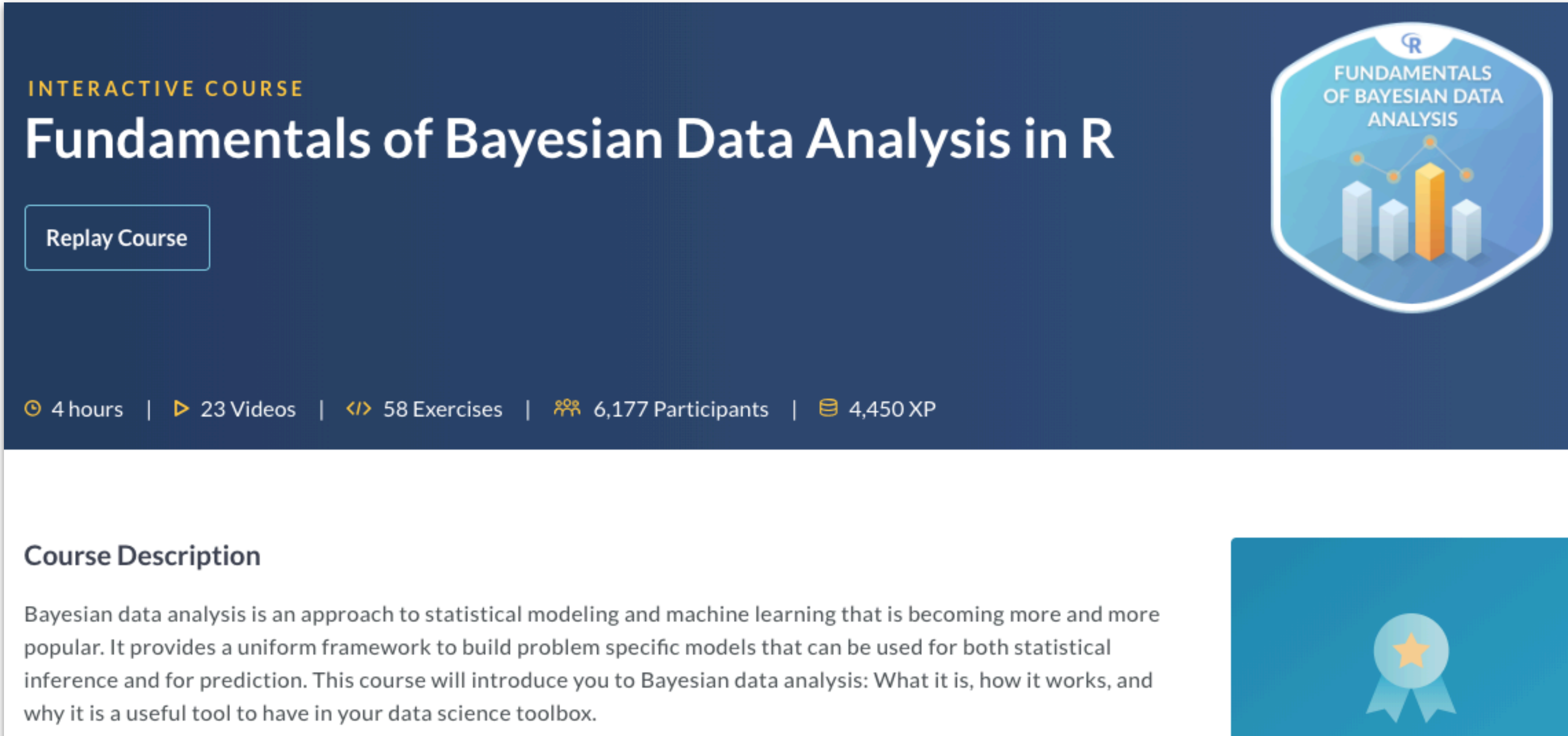
Time: 5 minutes

https://tinyurl.com/psych252bayes

Report: We will take a look together.

Datacamp course

recommended!!



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

Great online book

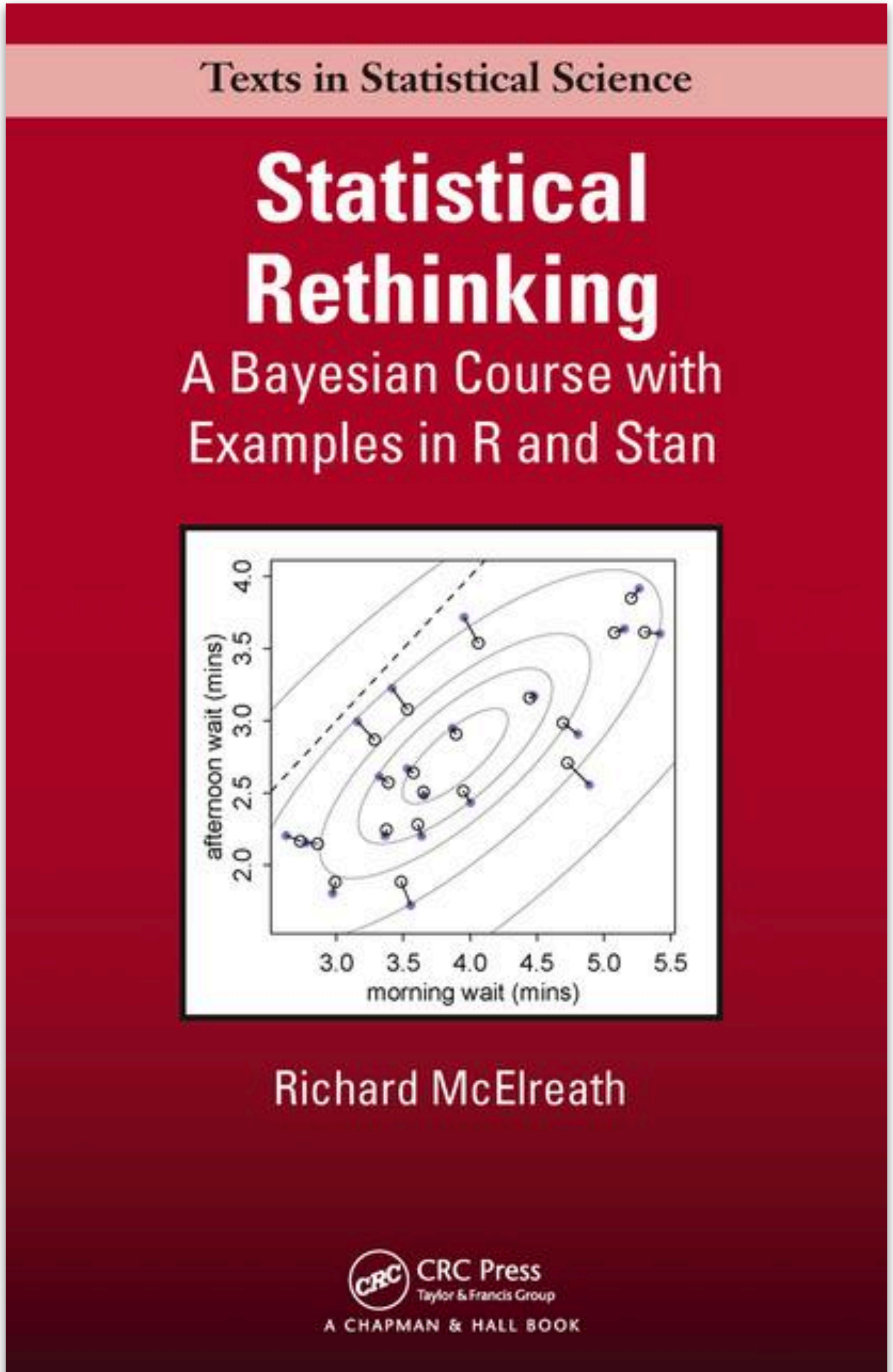
An Introduction to Data Analysis

Michael Franke

last rendered at: 2021-02-23 12:07:27

https://michael-franke.github.io/intro-data-analysis/index.html

Great book on Bayesian data analysis



- nice hands-on book (which uses R throughout)
- rewrite of all the code with `"tidyverse"` and `"BRMS"` is here: https://bookdown.org/content/4857/
- video lectures are available here: https://github.com/rmcelreath/stat_rethinking_2022

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 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)**

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) to make a decision
- Bayesians use Bayes' rule

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
- better predictions!

Flash from the past

Clue guide to probability

we derived this using the definition of conditional probability

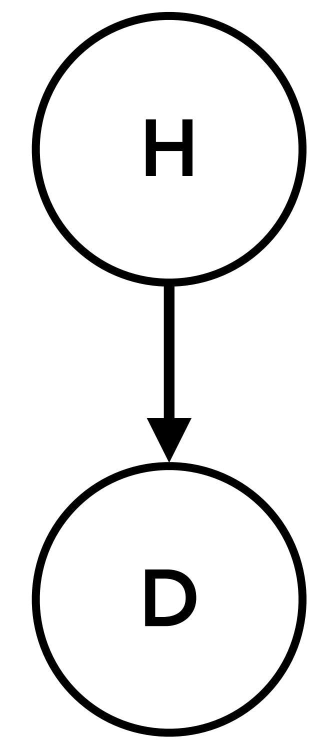
$$p(B\,|\,A) = \frac{p(A\,|\,B) \cdot p(B)}{p(A)}$$

$$\overset{\text{posterior}}{p(H\,|\,D)} = \frac{\overset{\text{likelihood}}{p(D\,|\,H)} \cdot \overset{\text{prior}}{p(H)}}{p(D)}$$

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

what we know

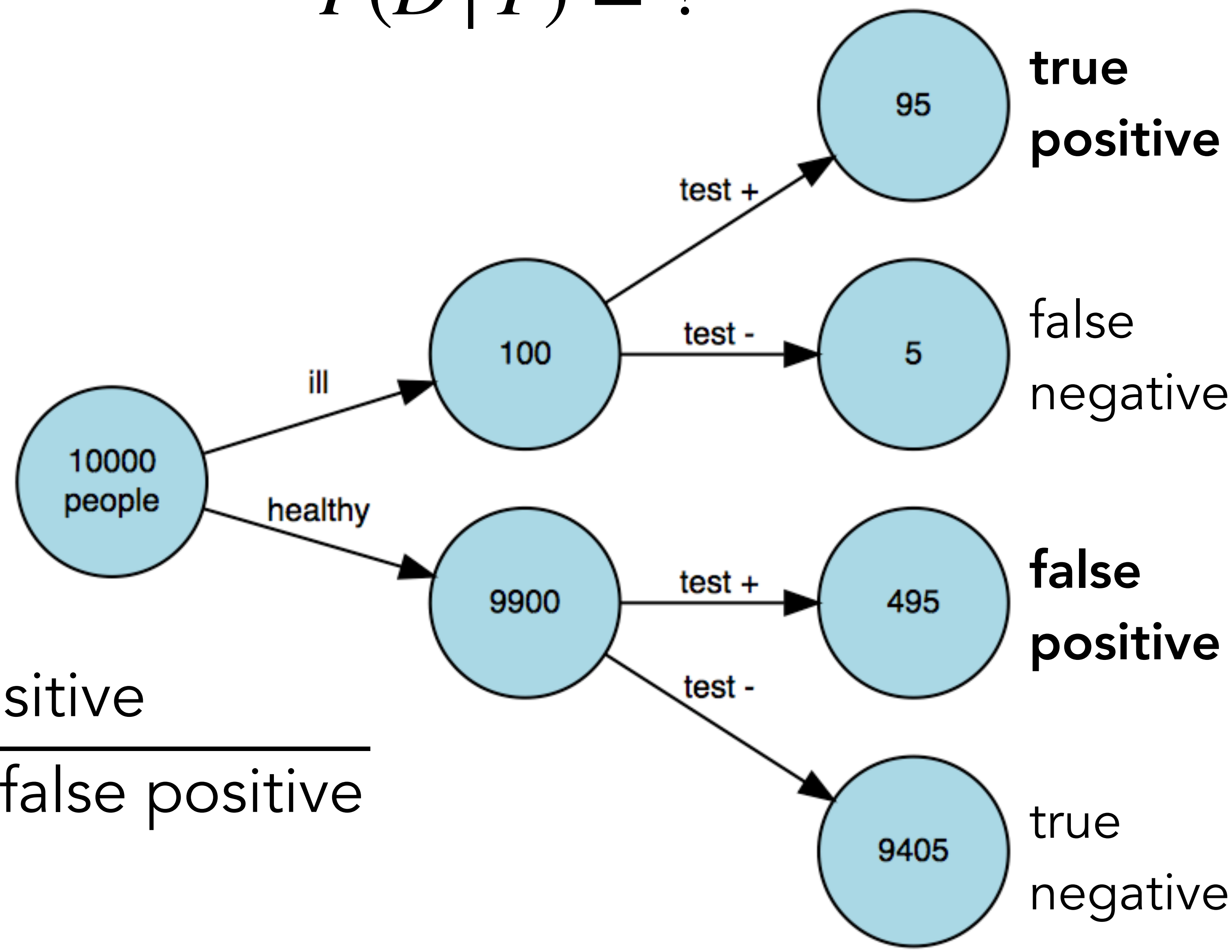
$$P(D) = 0.01$$

$$P(T \mid D) = 0.95$$

$$P(T \mid \neg D) = 0.05$$

what we want to know

$$P(D \mid T) = ?$$



$$P(D \mid T) = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

$$= \frac{95}{95 + 495}$$

$$\approx 0.16$$

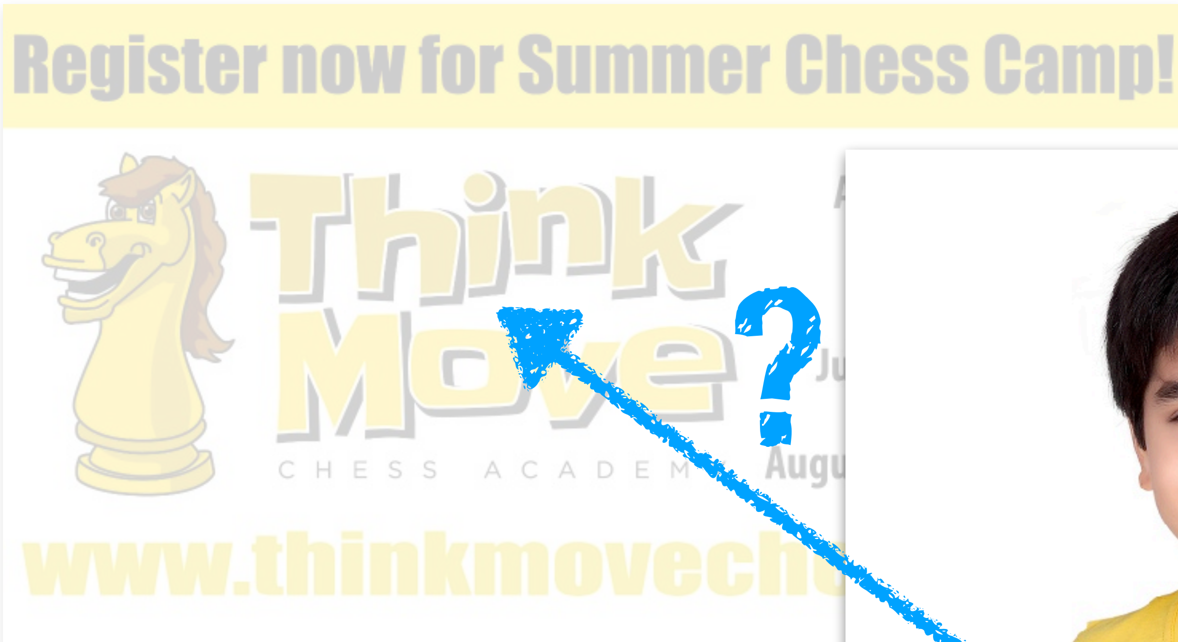
Summer camp

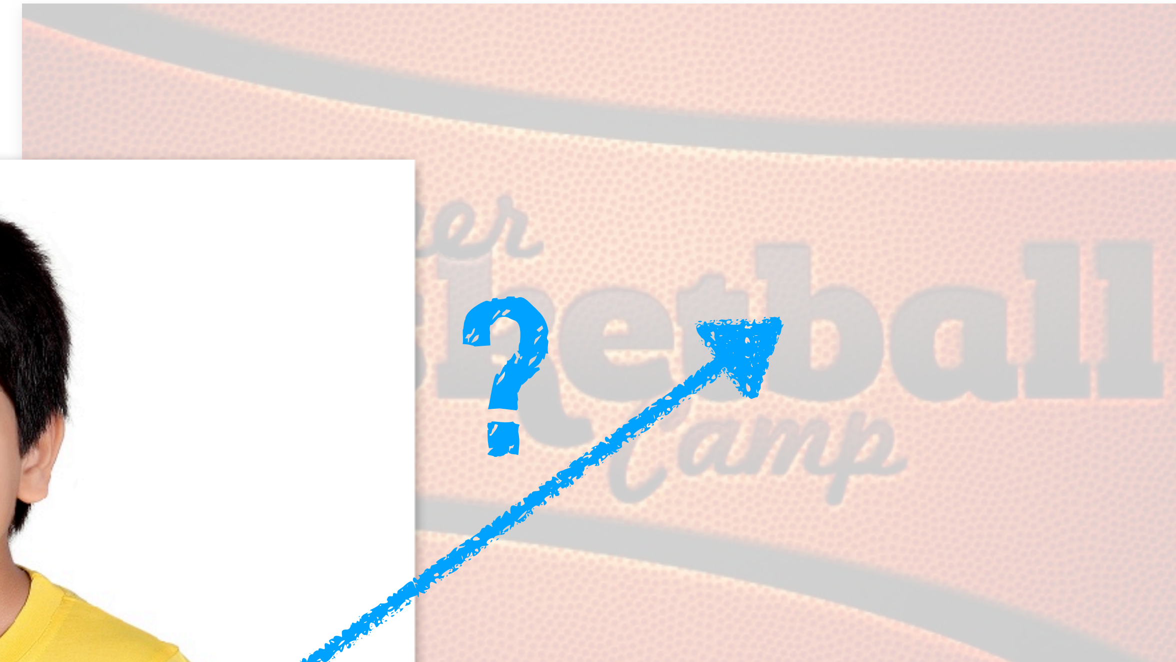
twice as many kids go to the basketball camp

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

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

Summer camp





twice as ma etball camp

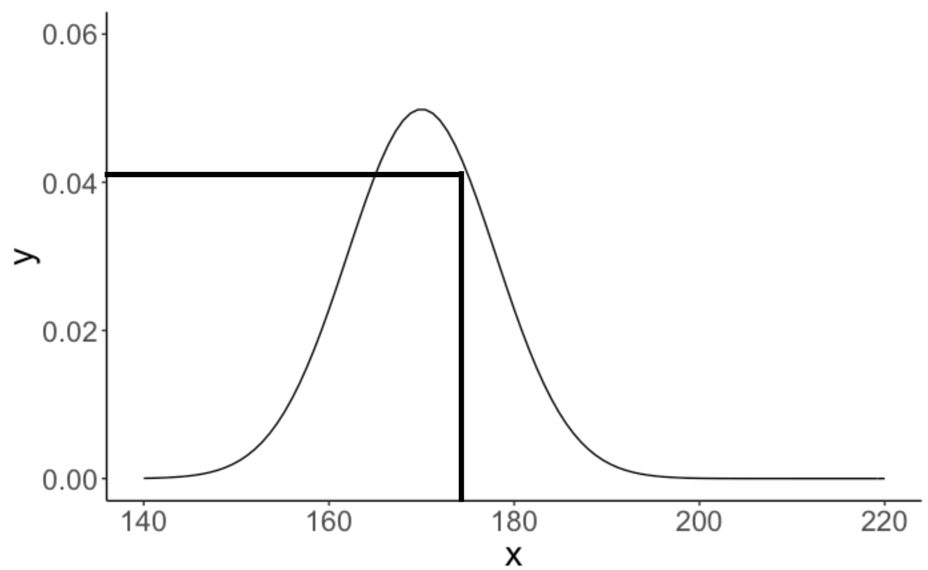
$X \sim$ Normal($\mu = 170$ nal($\mu = 180$, $\sigma = 10$)

Summer camp

prior

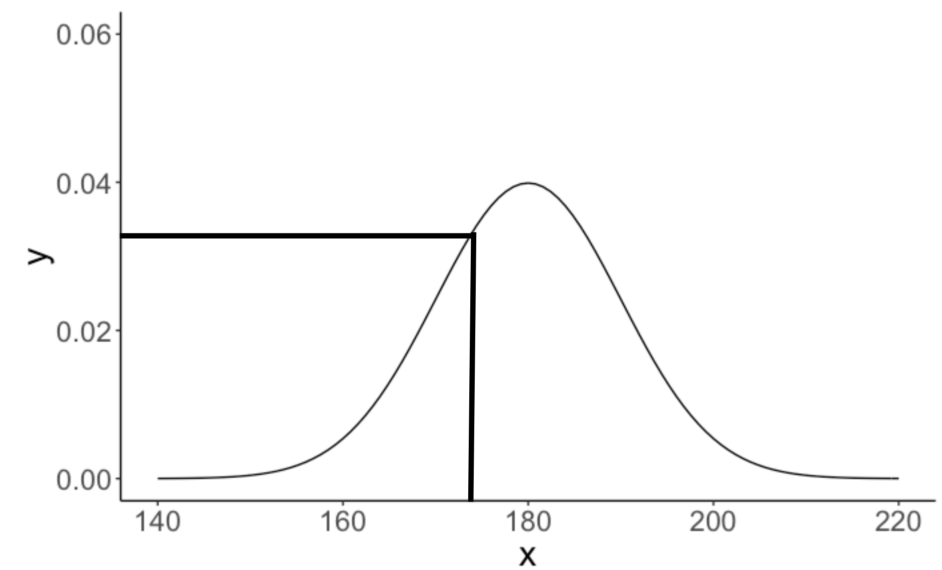
$$p(\text{chess}) = \frac{1}{3} \qquad p(\text{basketball}) = \frac{2}{3}$$

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!

Flipping coins

Flipping coins

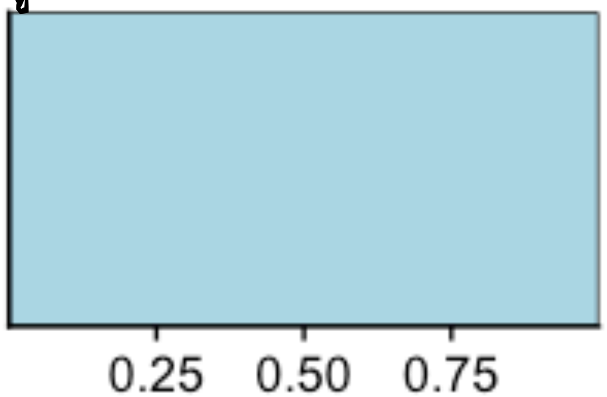
What's the probability of my coin coming up heads?

$$\theta = ?$$

Learning from data

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

prior



Today's posterior is tomorrow's prior.

$$p(\theta \,|\, \mathrm{n}_{\mathrm{success}} = 6,\ \mathrm{n}_{\mathrm{trials}} = 8)$$

Coin flip example

Which coin did I flip?

Hypotheses

$\theta = 0.3$

$\theta = 0.5$

$\theta = 0.9$

Data

#8 tails, #2 heads

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

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

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

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

law of total probability:

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

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

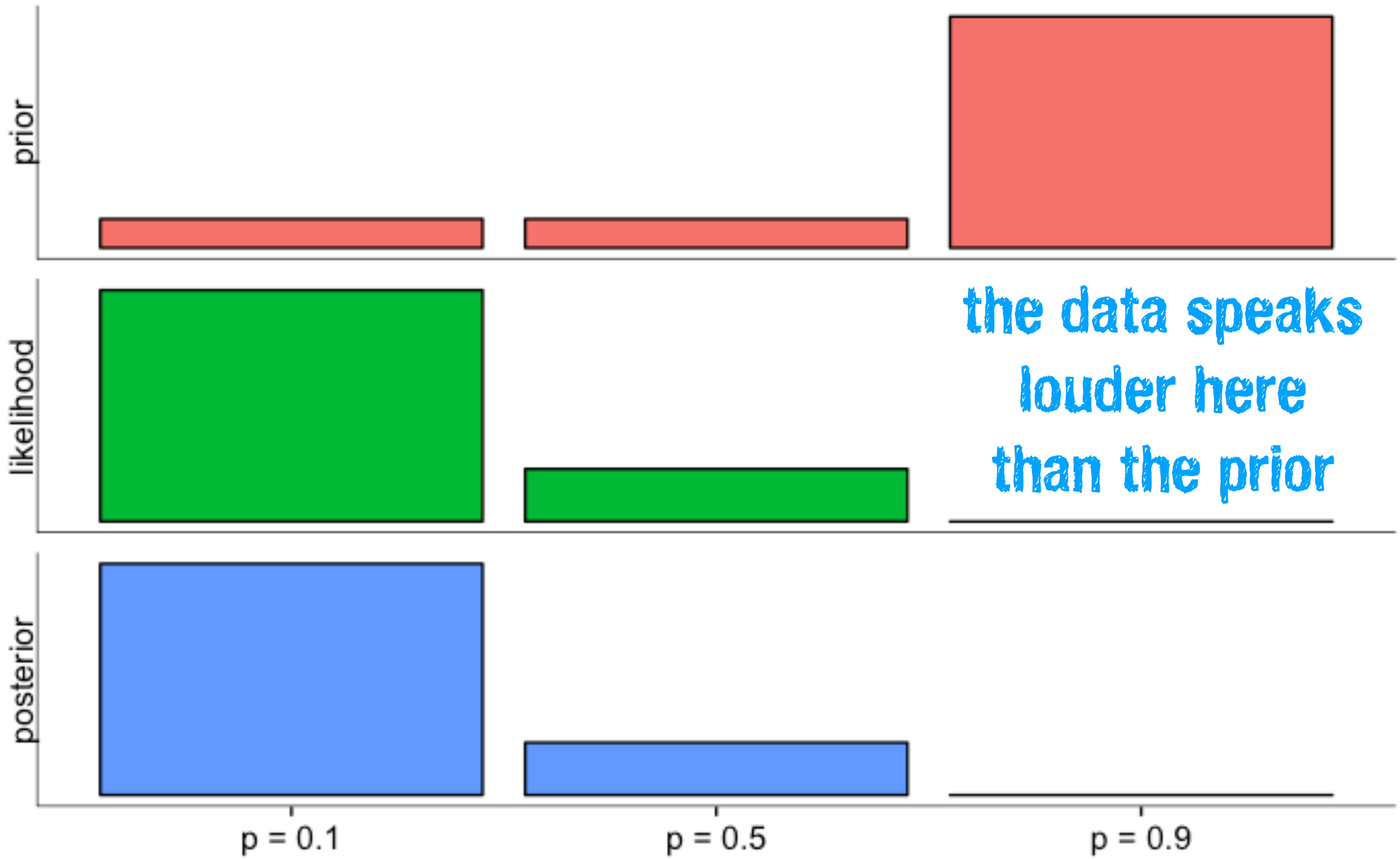
what the model knows **after** having seen the data

posterior = multiplicative weighting of prior and likelihood

Coin flip example

data: #8 tails, #2 heads

Which coin was flipped?

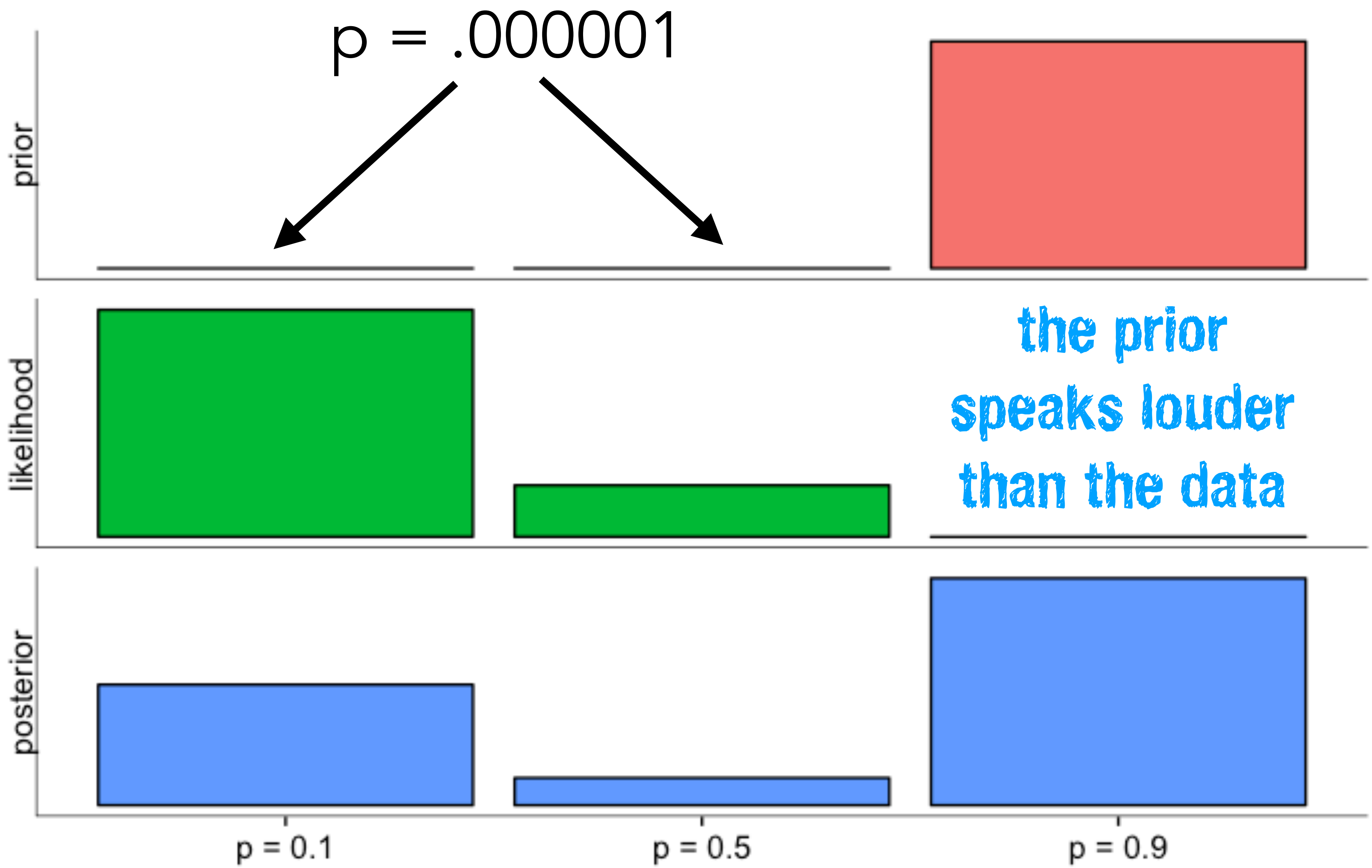


posterior = multiplicative weighting of prior and likelihood

Coin flip example

data: #8 tails, #2 heads

Which coin was flipped?

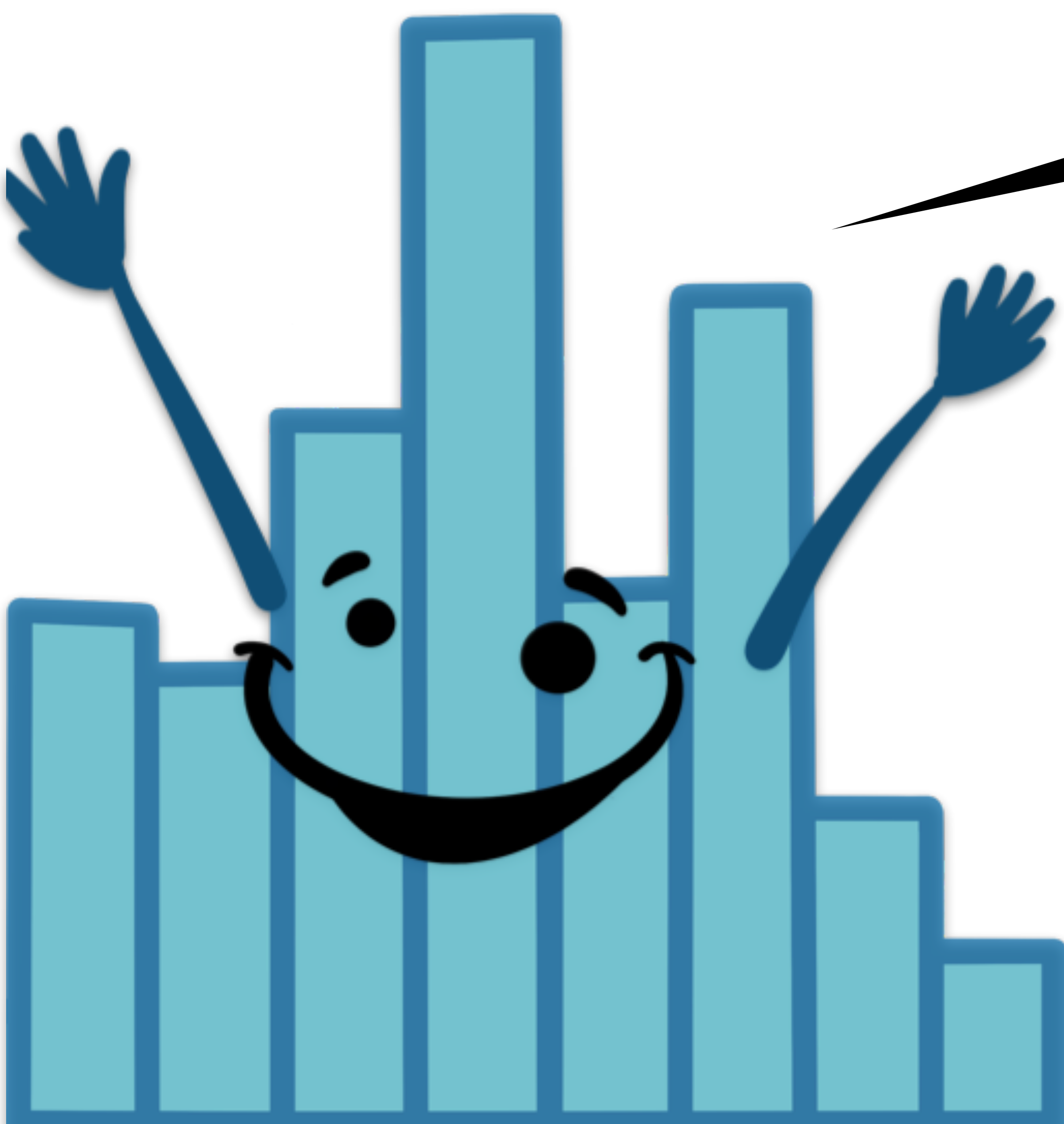


posterior = multiplicative weighting of prior and likelihood

We're listening to "Sunkissed" by "khai dreams" submitted by Sarah Wu

02:00

stretch break!



What affects the posterior?

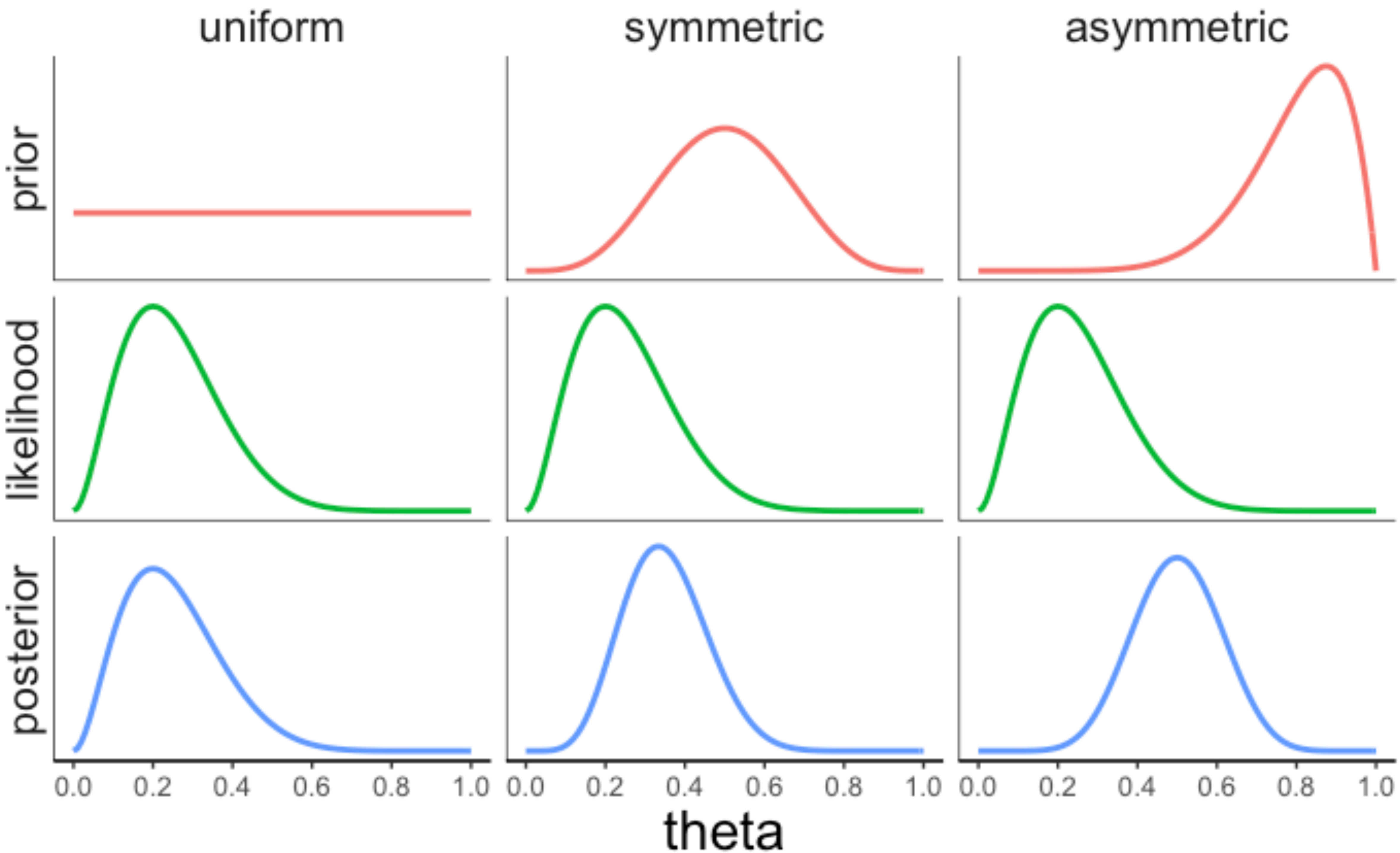
What affects the posterior?

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

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

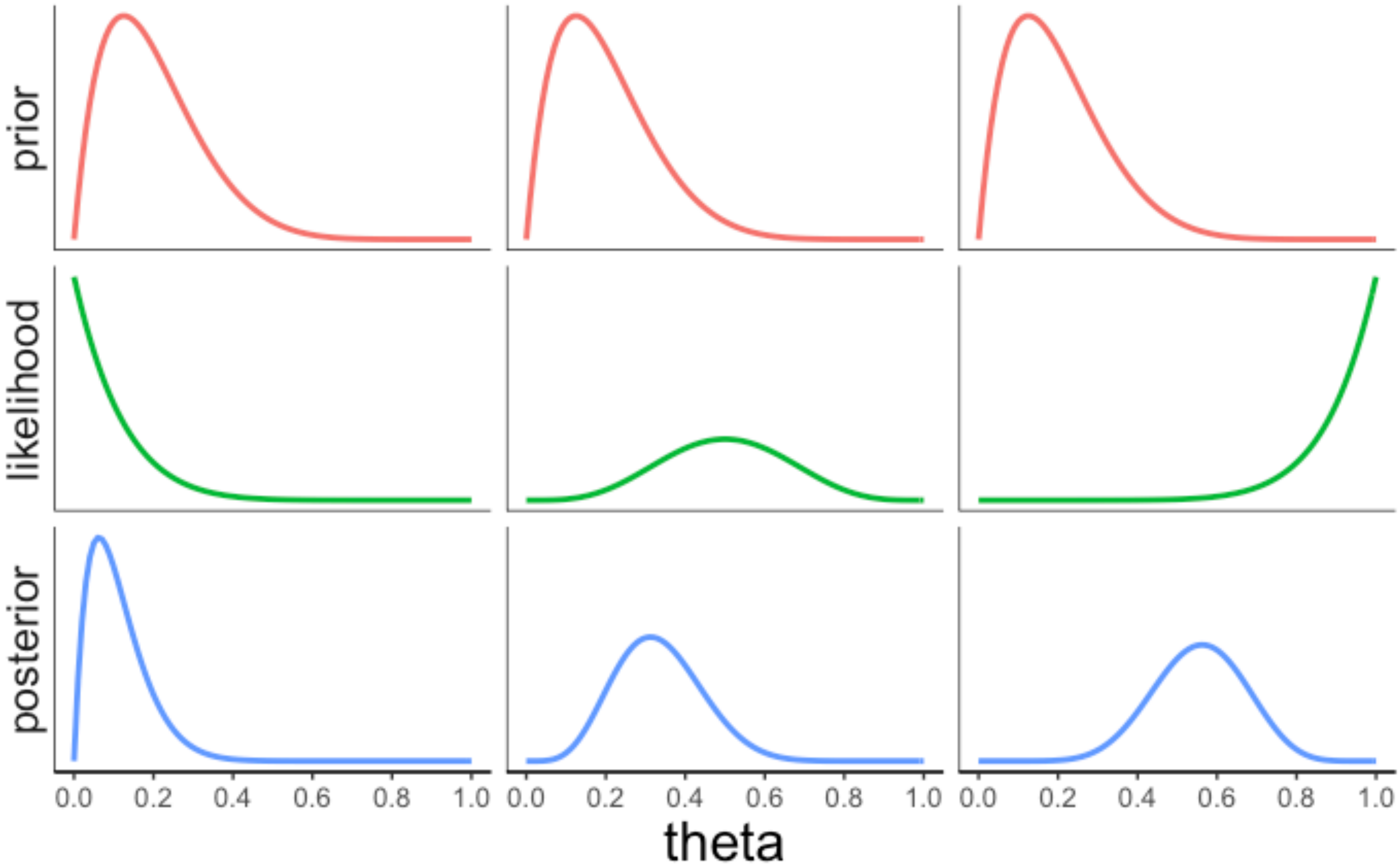
The effect of the **prior**

same data, different priors

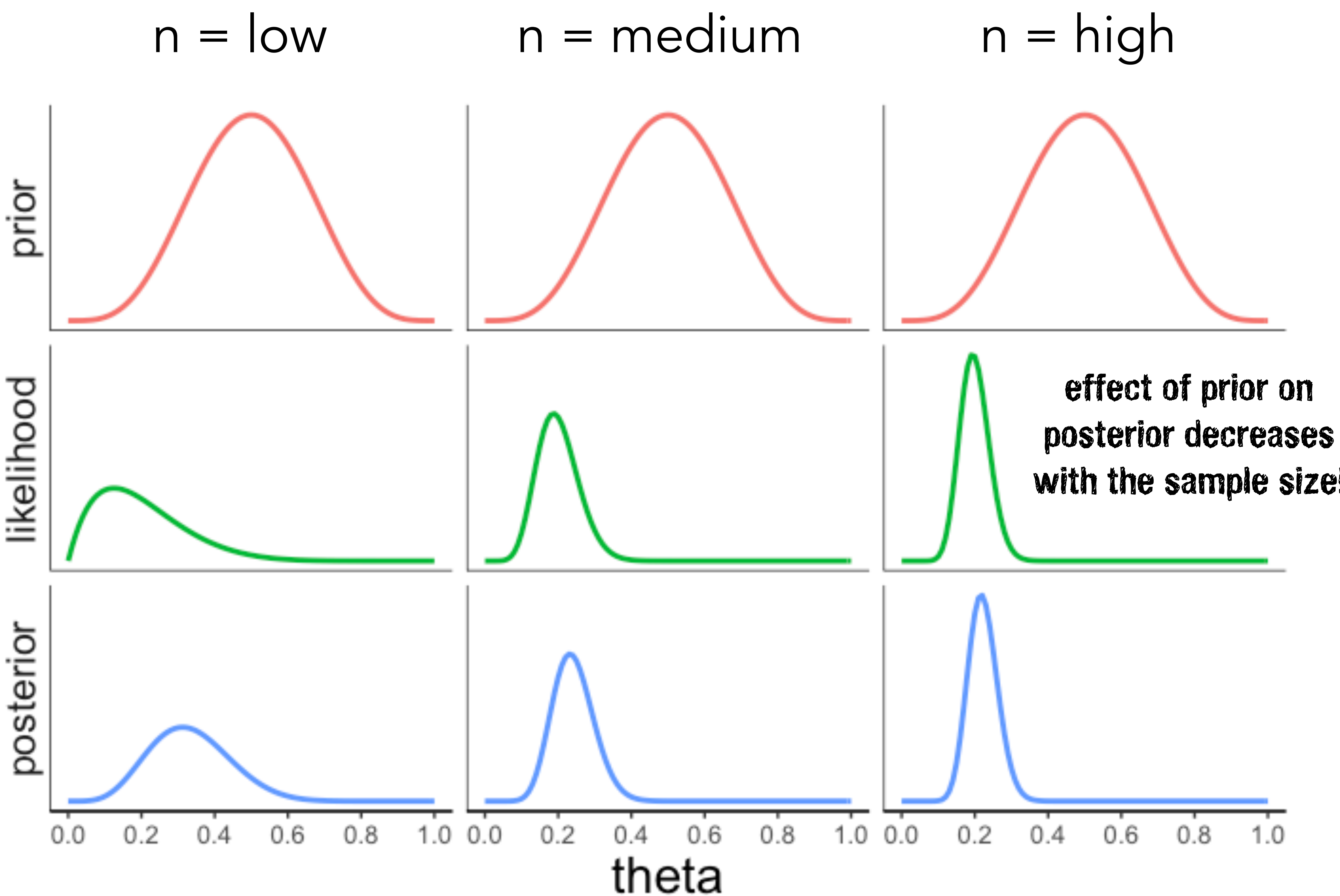


The effect of the **likelihood**

same prior, different data



The effect of **sample size**



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

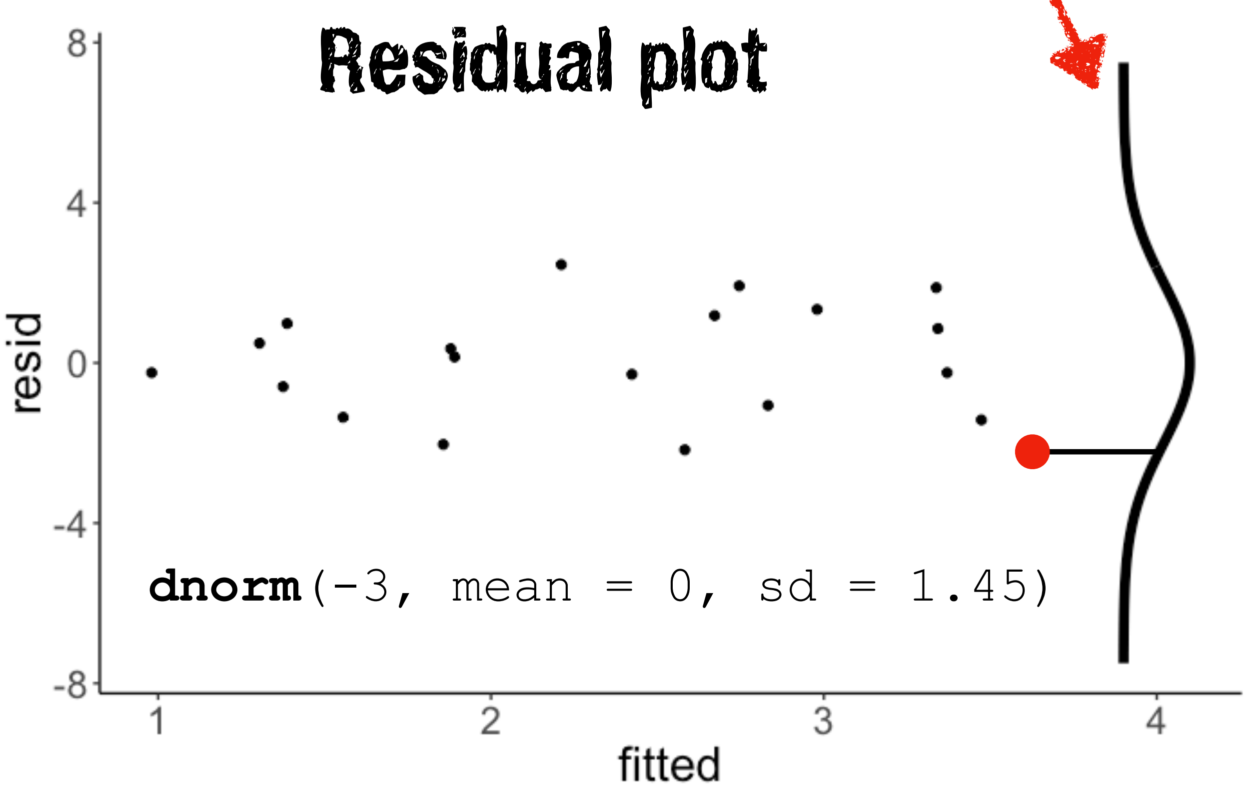
- **What probabilistic model describes best how the data were generated?**
  - What assumptions can you make about the data?
    - What's the nature of your dependent variable (e.g. binary, ordered, continuous)?
  - Does the model re-create the behavior of interest?

Likelihood

Gaussian distribution

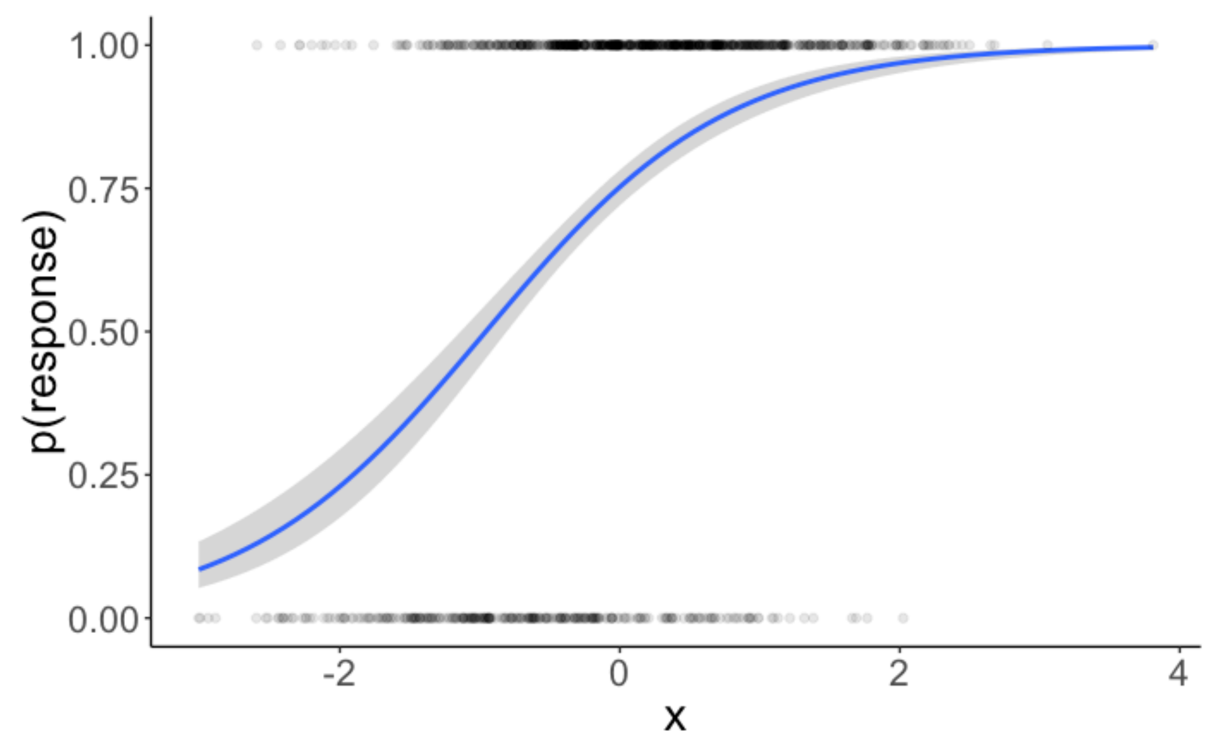
$$Y_i = b_0 + b_1 \cdot x_i + e_i$$

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

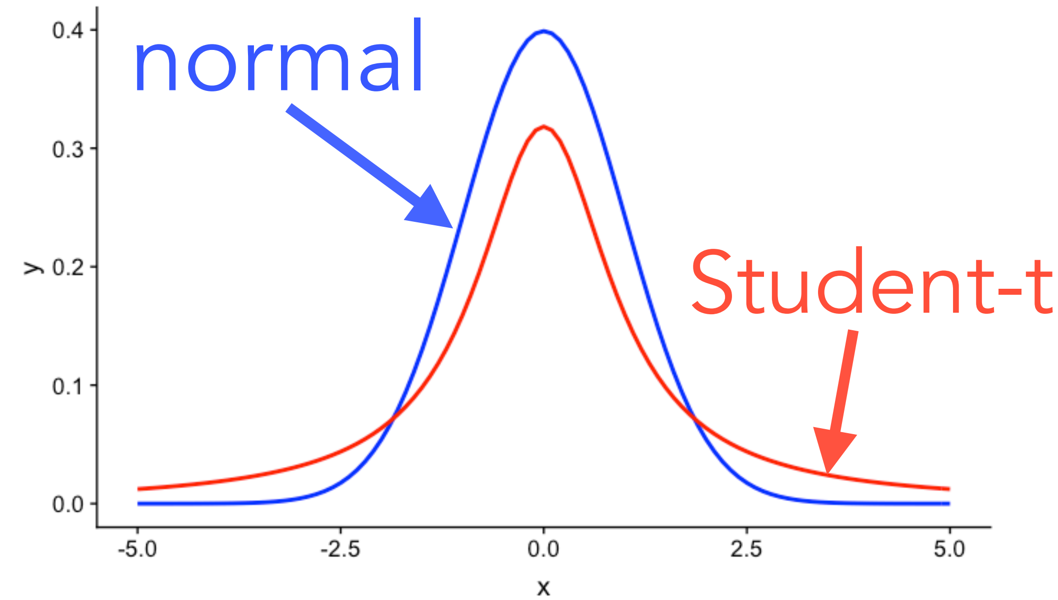


Binomial distribution

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

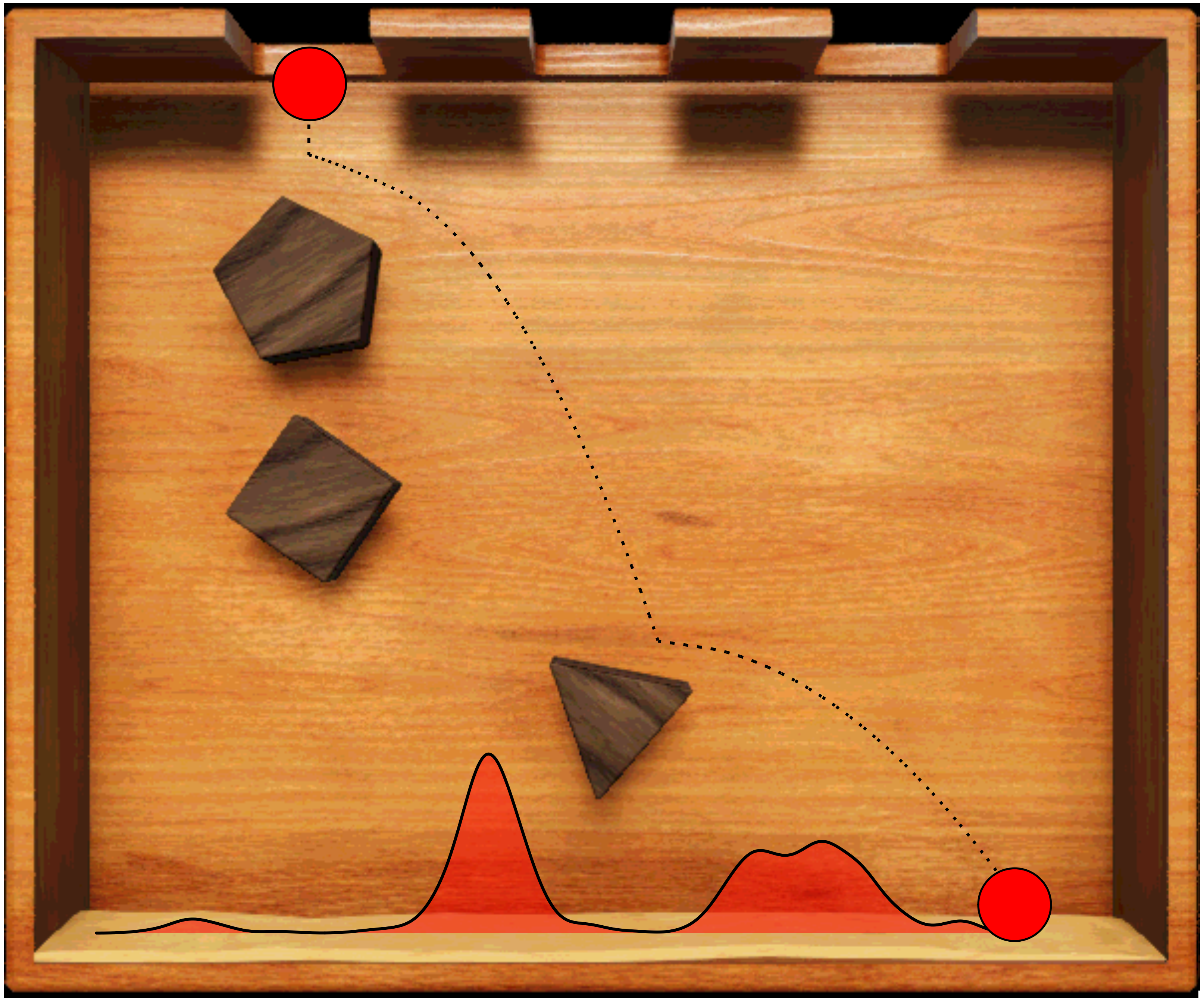


Likelihood



- **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**)
- …

Prediction: Where will the ball land?

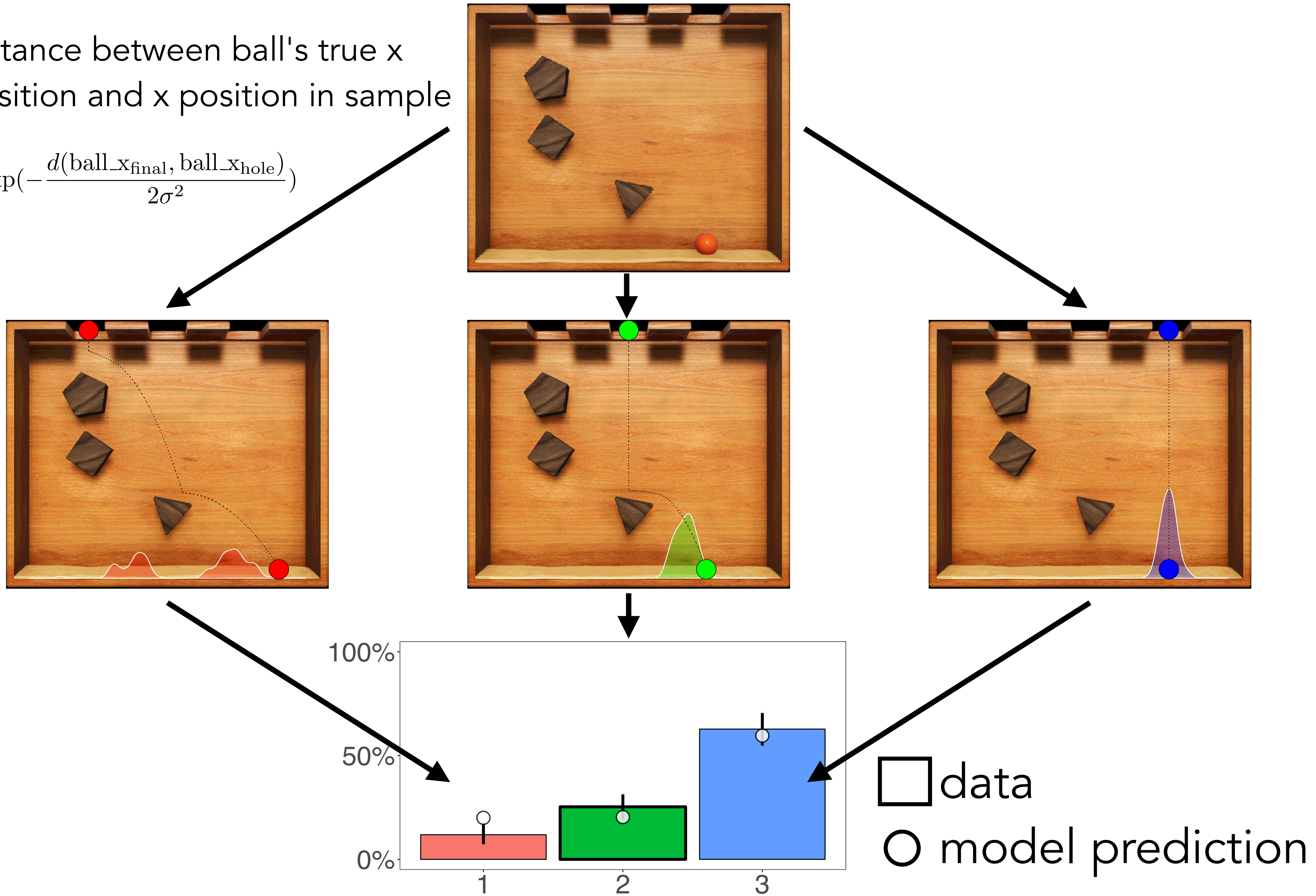


Aggregated responses

Inference: In which hole was the ball dropped?

distance between ball's true x position and x position in sample

$$\exp\left(-\frac{d(\text{ball\_x}_{\text{final}}, \text{ball\_x}_{\text{hole}})}{2\sigma^2}\right)$$



Prior

Likelihood Prior

Posterior

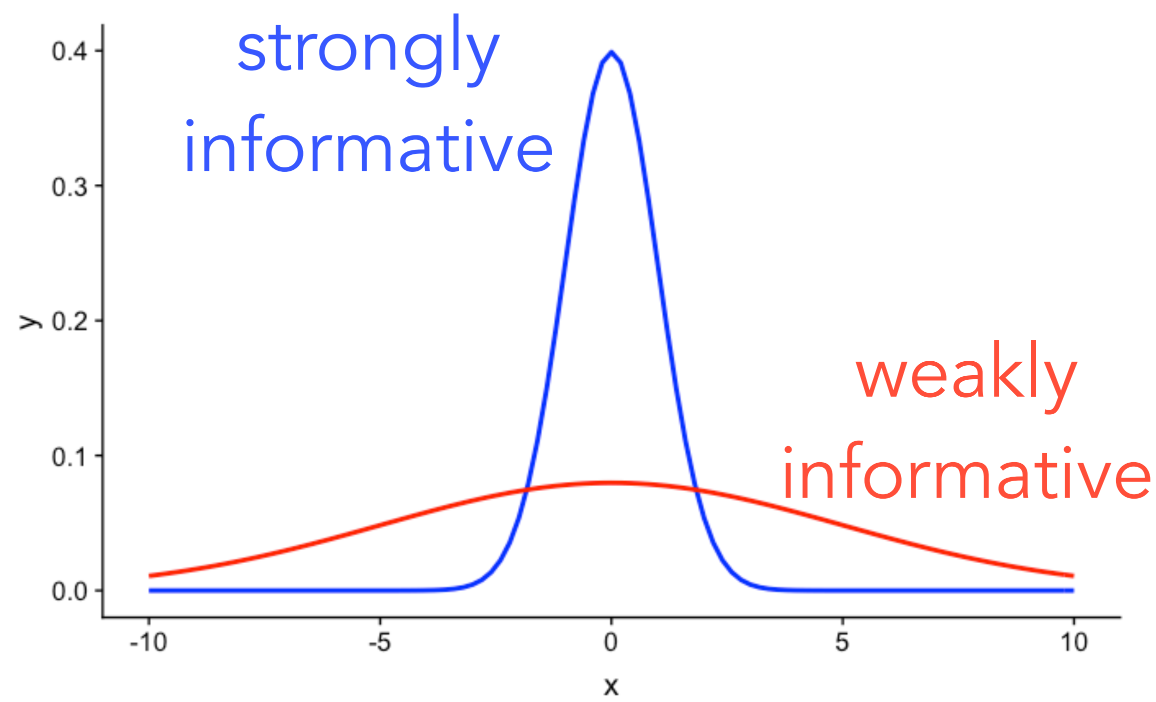
$$p(H \mid D) = \frac{p(D \mid H) \cdot \color{red}{p(H)}}{p(D)}$$

Normalizing constant

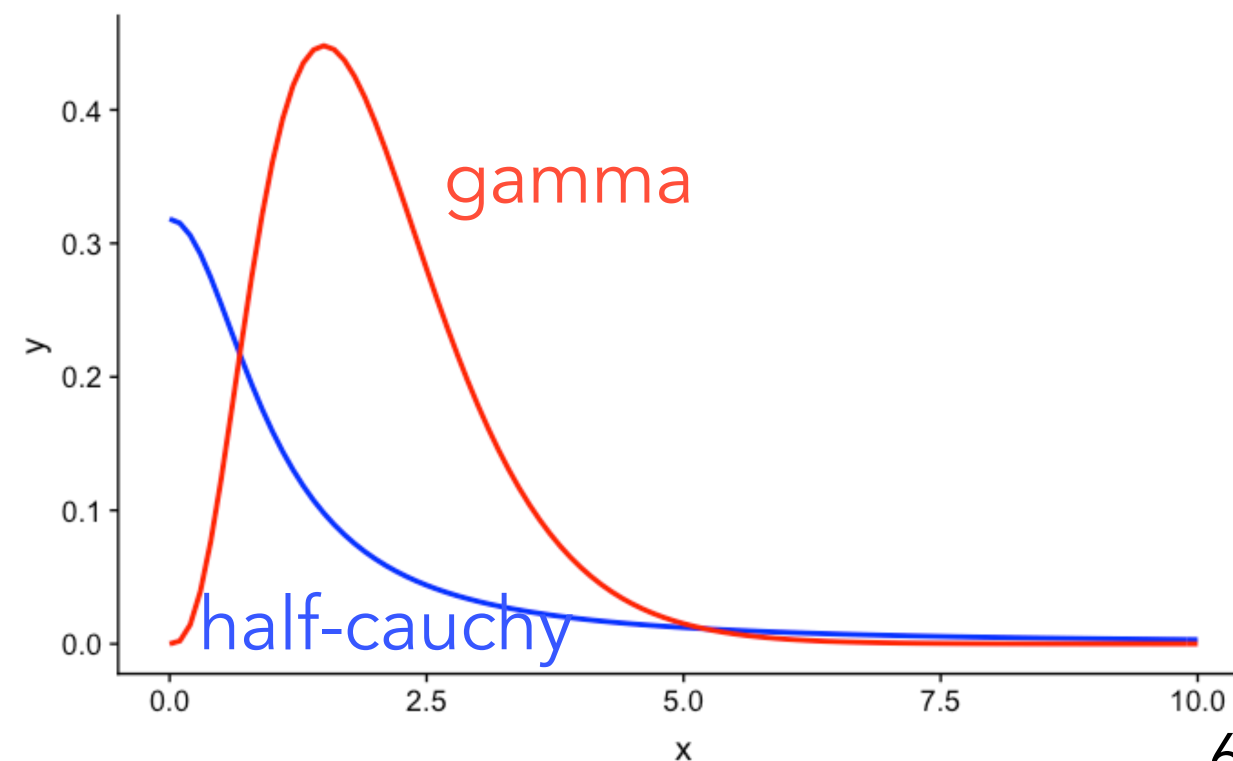
Prior

- **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



Inference

Likelihood **Prior**

Posterior

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

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

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

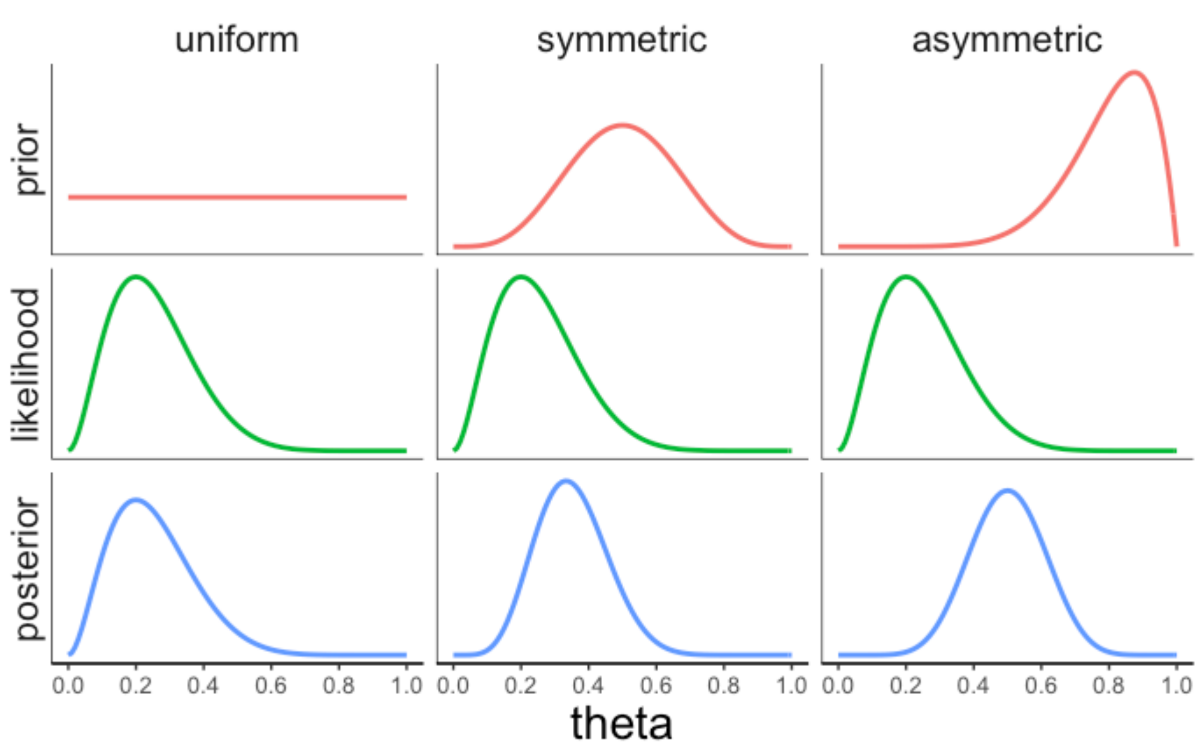
integral over all possibilities

Discretizing the parameters

```
# 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(theta, shape1 = 1, shape2 = 1),
                  prior_normal = dbeta(theta, shape1 = 5, shape2 = 5),
                  prior_biased = dbeta(theta, 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

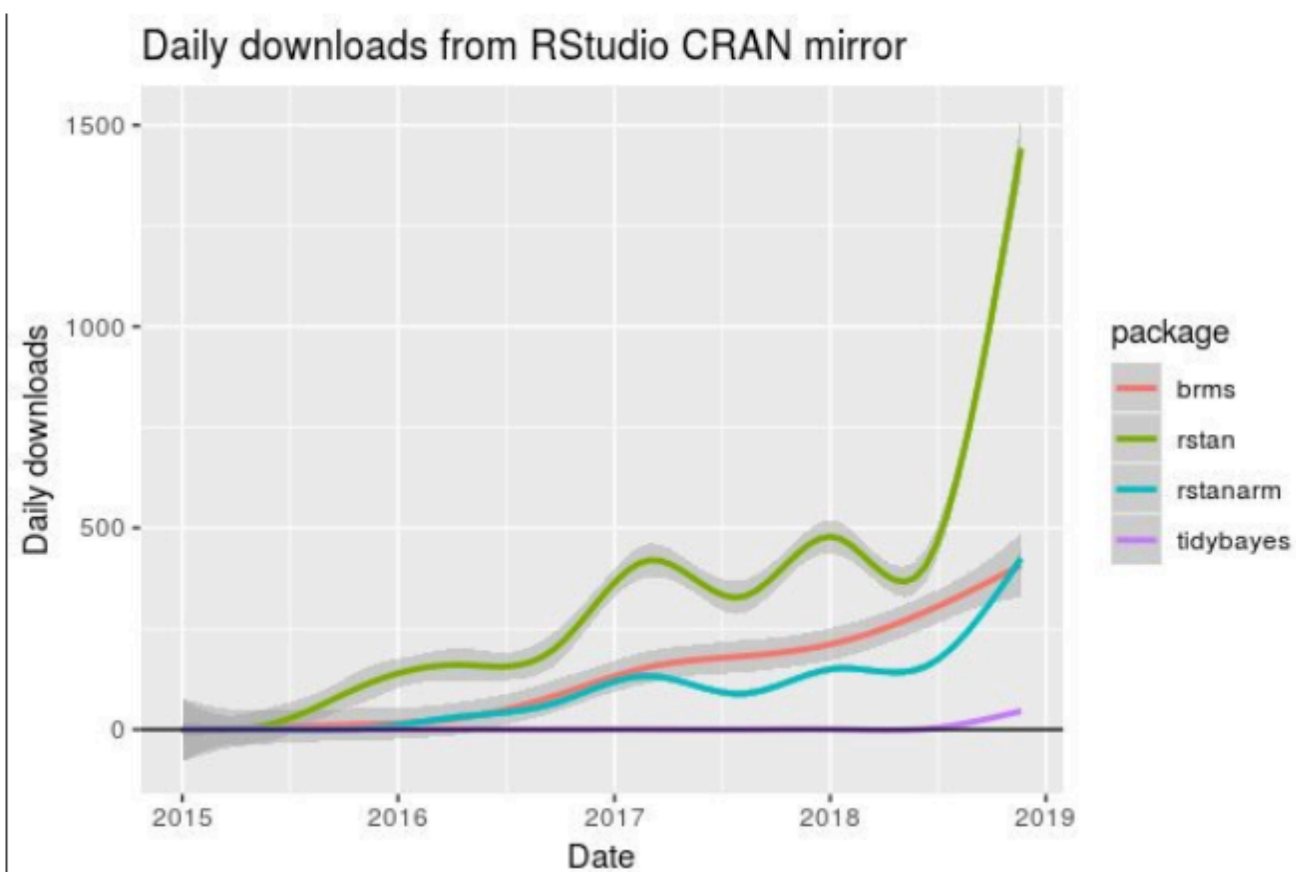
- we cannot directly calculate the probability of the posterior (because it might have a pretty weird shape)
- **but**: we can draw random samples from the posterior
- we can then use our data wrangling and visualization skills to make inferences based on these samples

It's as if ...

we don't have **`pnorm`**`()` but we do have **`rnorm`**`()`

Inference via sampling

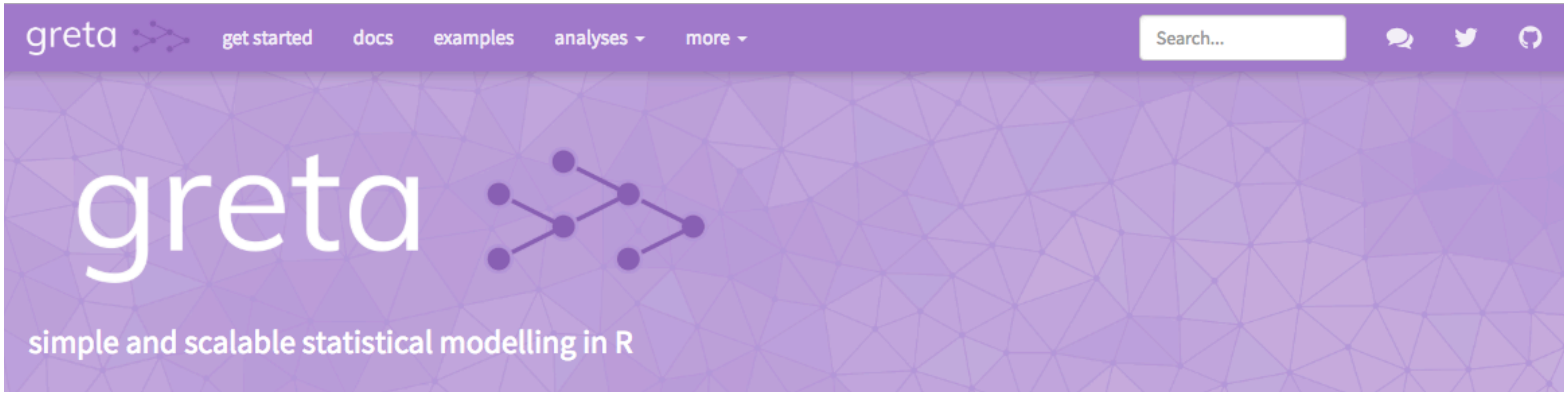
- Bayesian data analysis is becoming more popular because:
  - computers are getting more powerful
  - inference techniques are getting better
  - software packages become easier to use



Doing Bayesian data analysis

Software packages

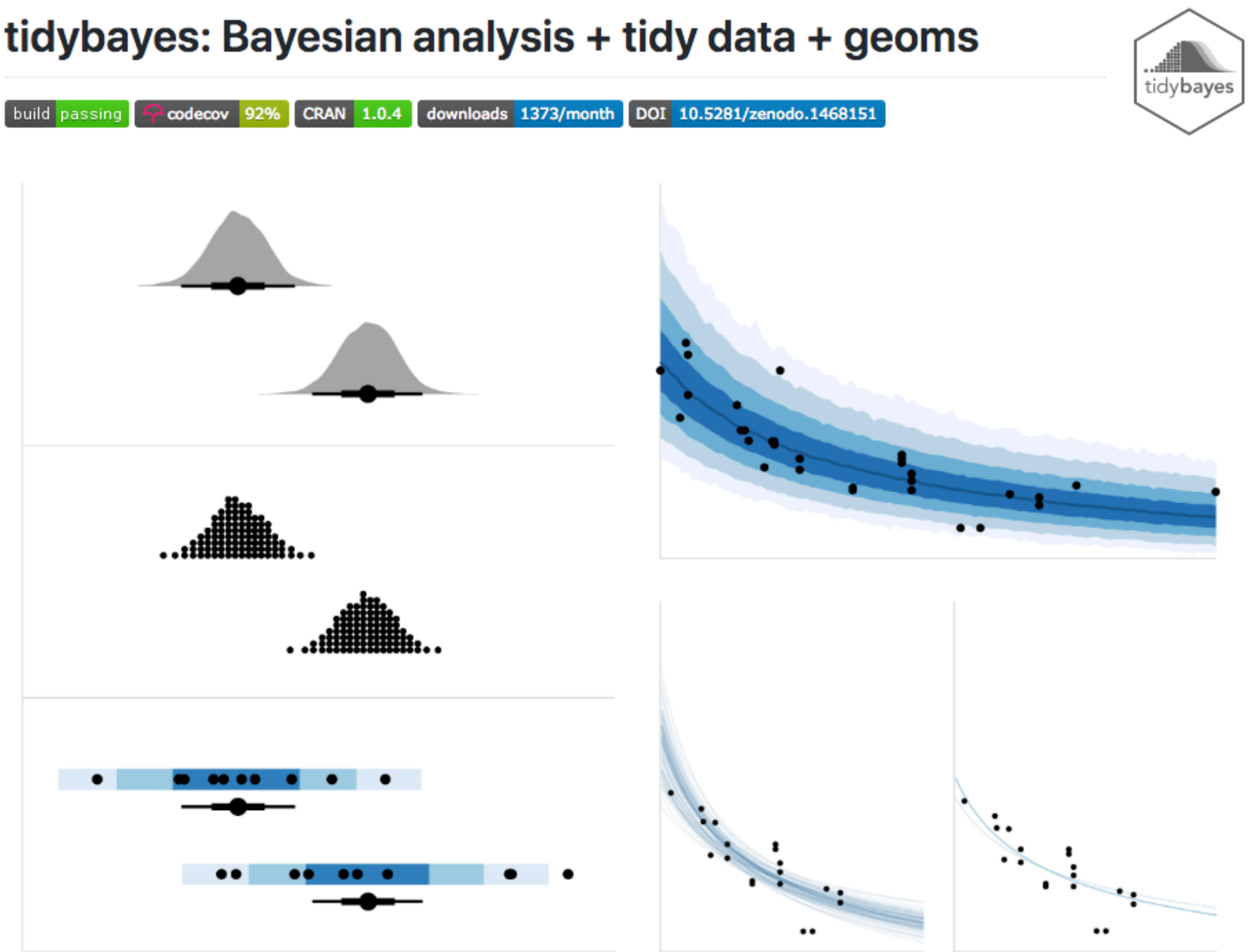
`library("greta")`



- 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 ...)

Software packages

```
library("tidybayes")
```

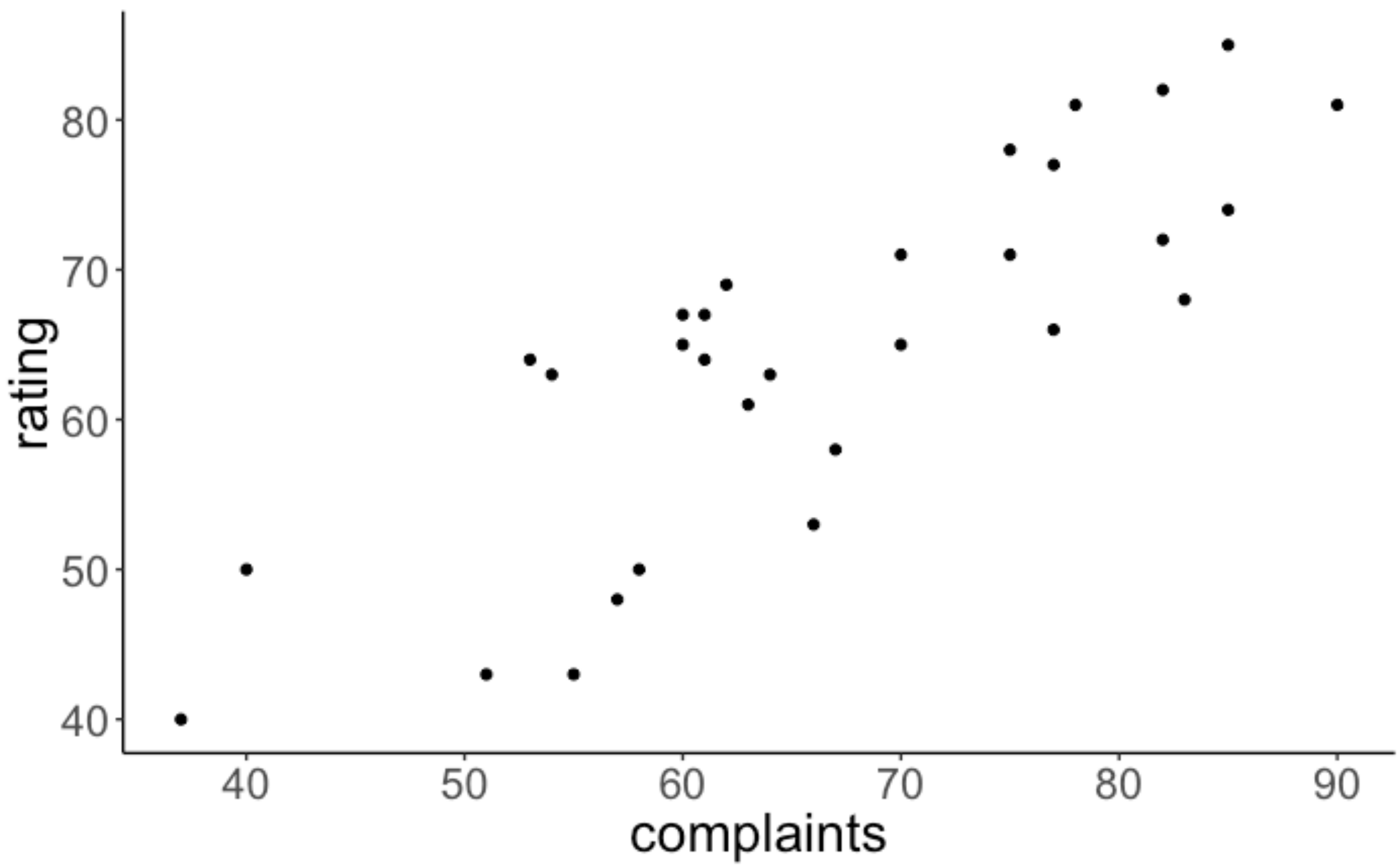


Matthew Kay

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?



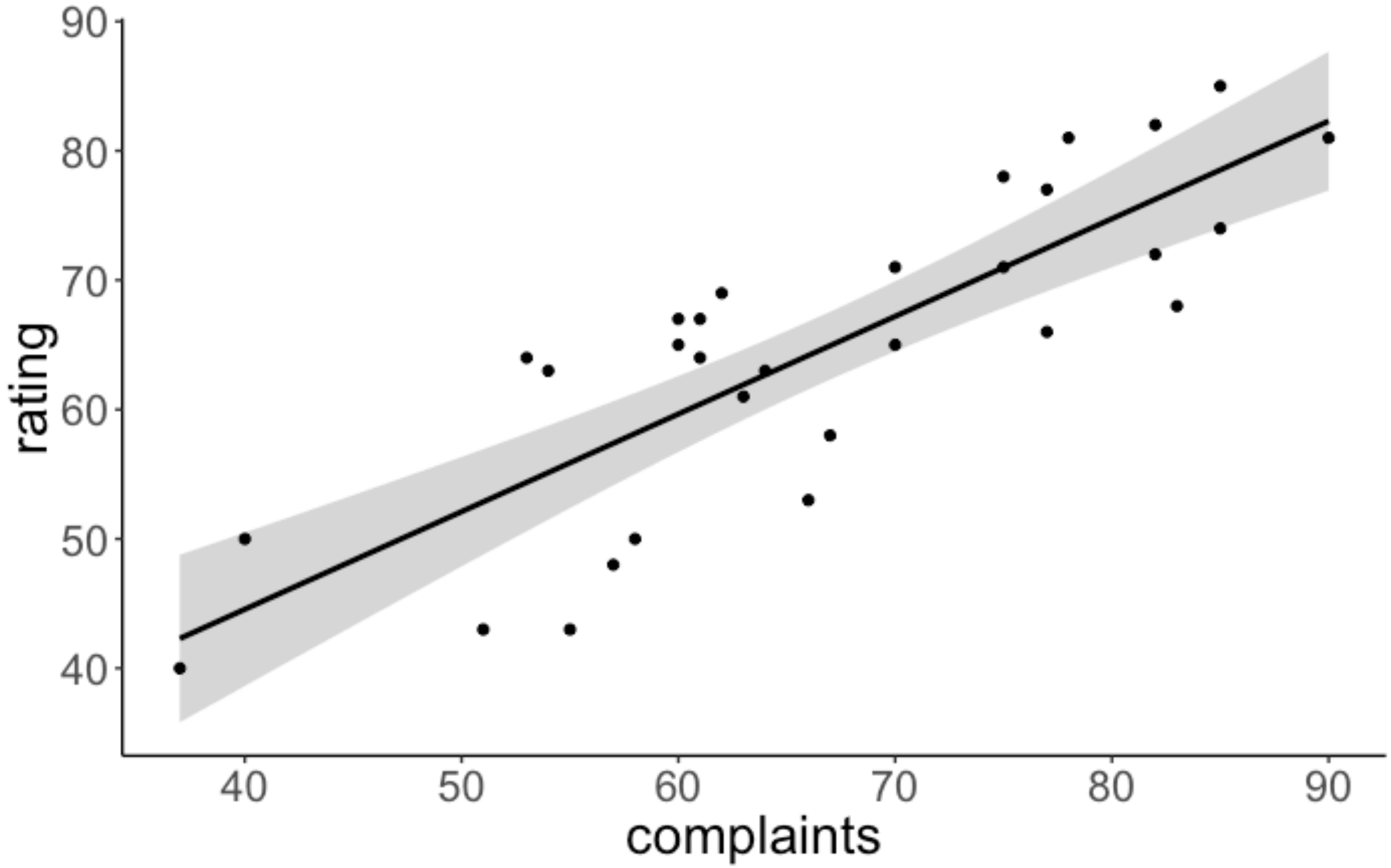
Frequentist analysis

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)
```

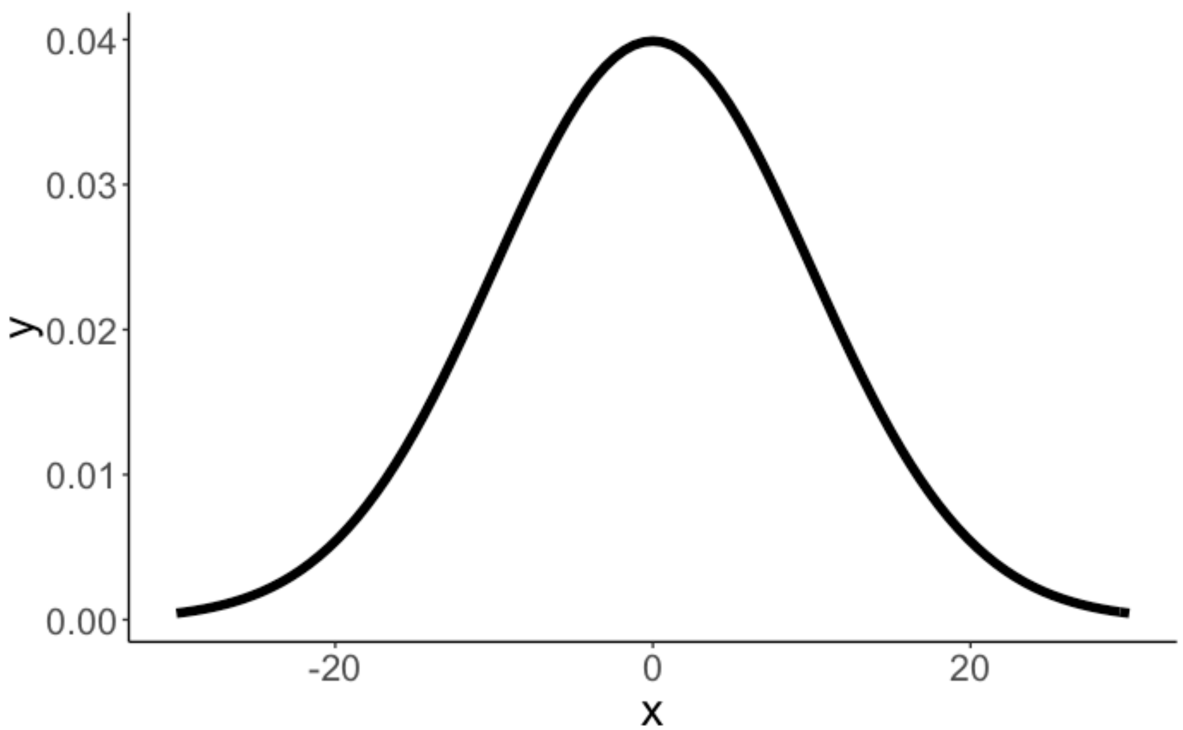
priors

linear combination

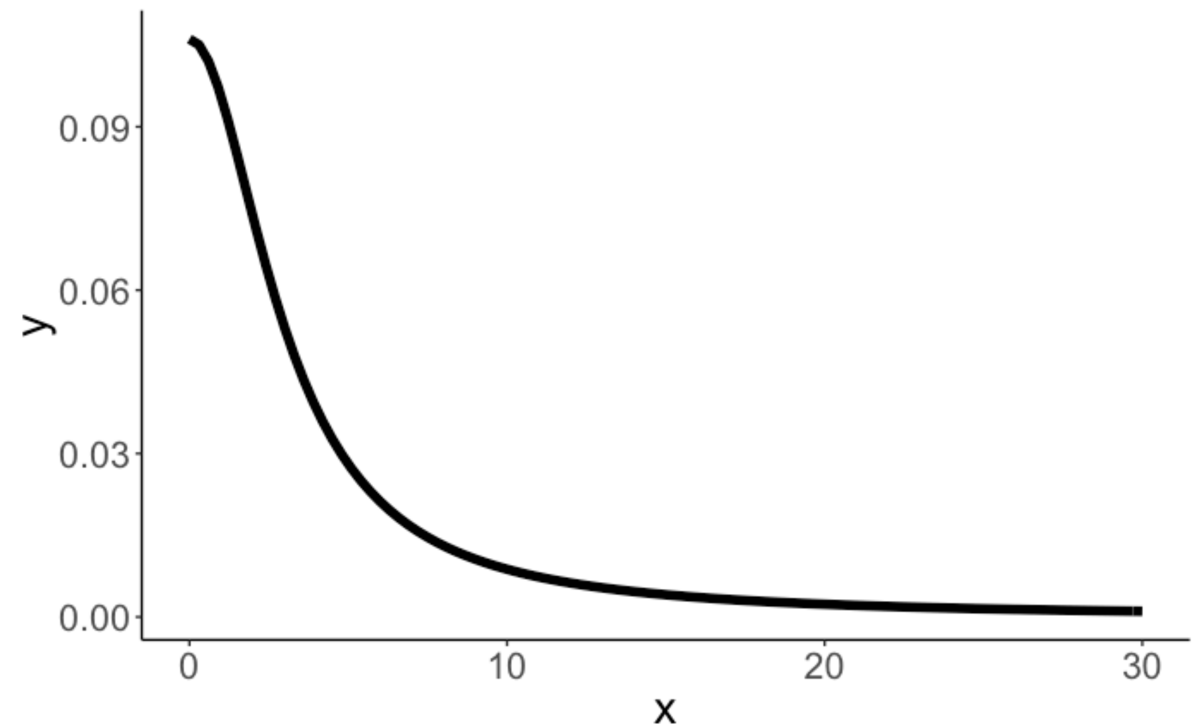
Gaussian likelihood

build the model

Priors



Gaussian prior on intercept and coefficient

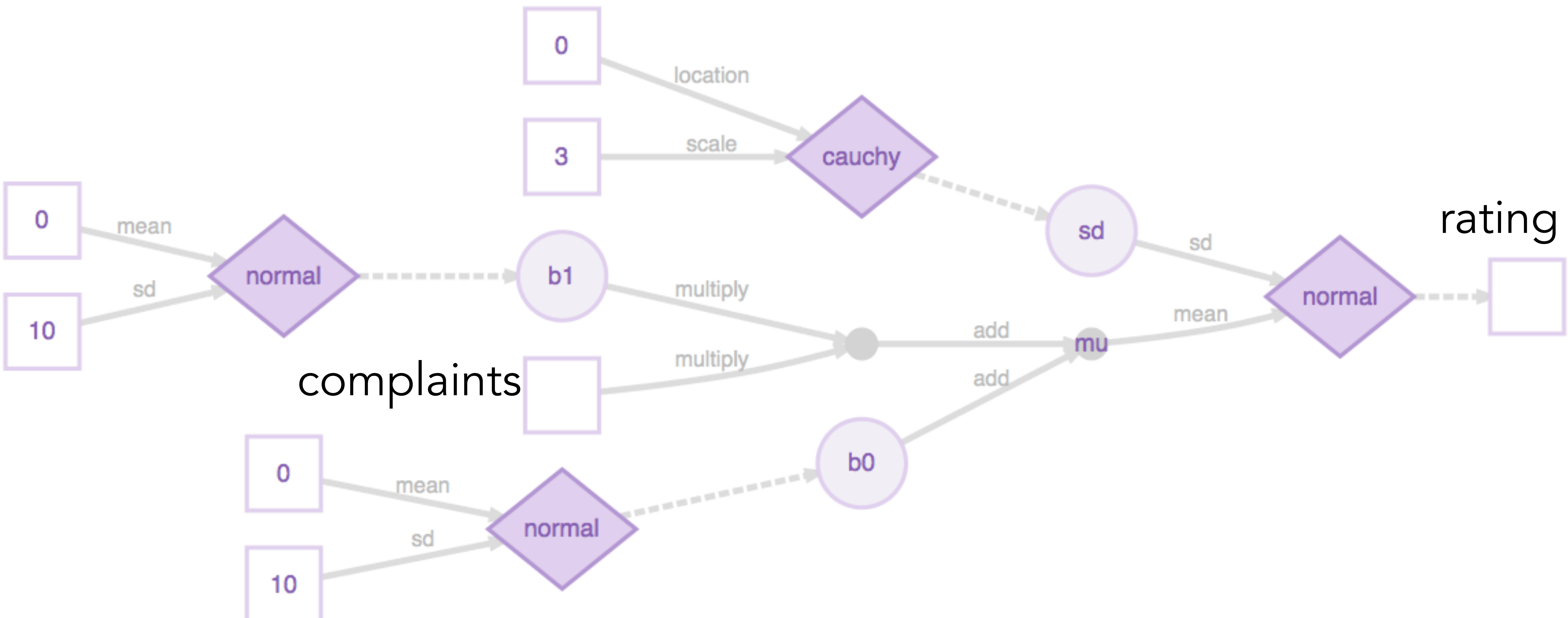


Truncated Cauchy prior on the standard deviation

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

Graphical representation of the model

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



Inference via sampling

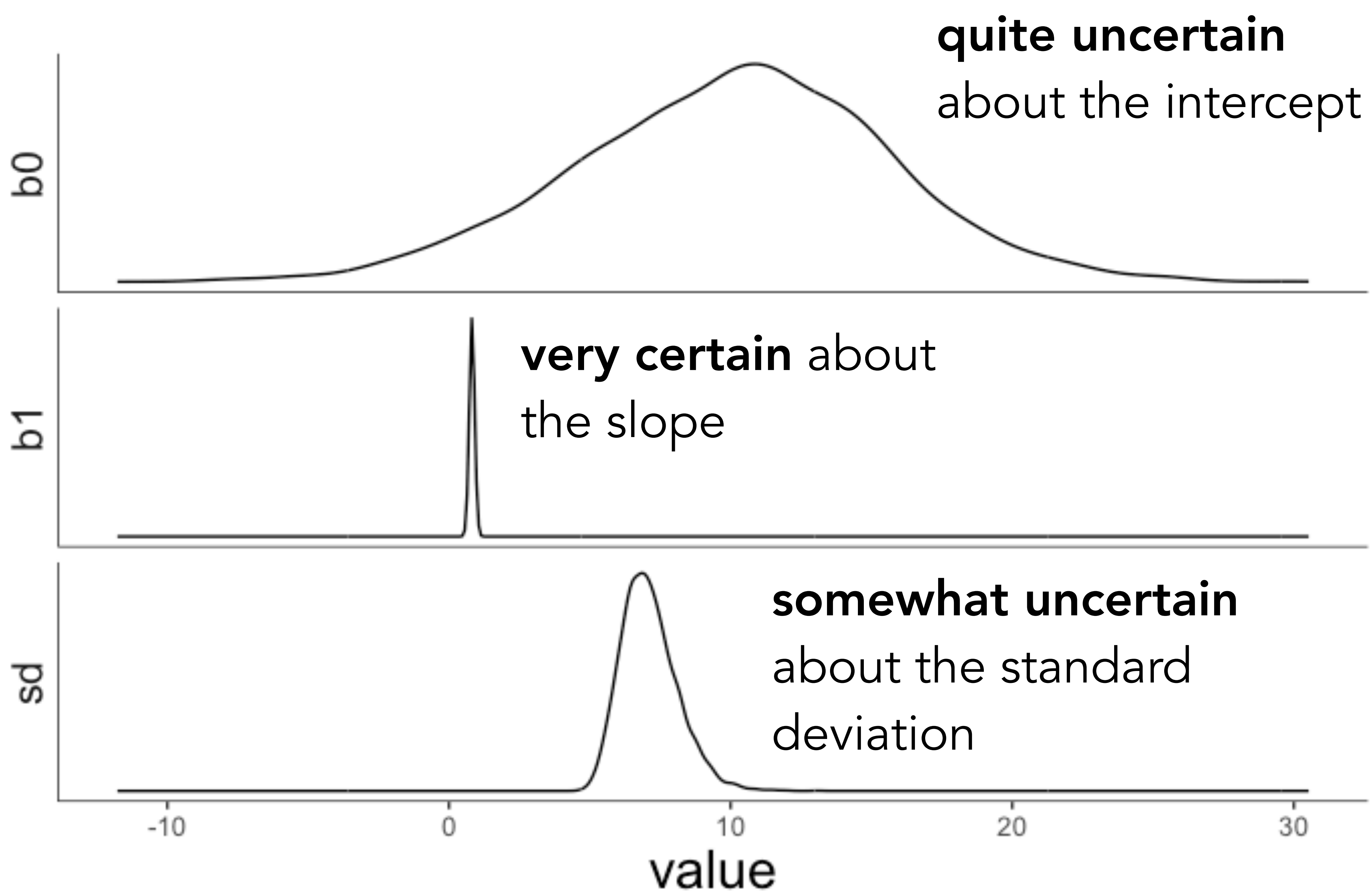
Markov Chain
Monte Carlo
inference

```r
# 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

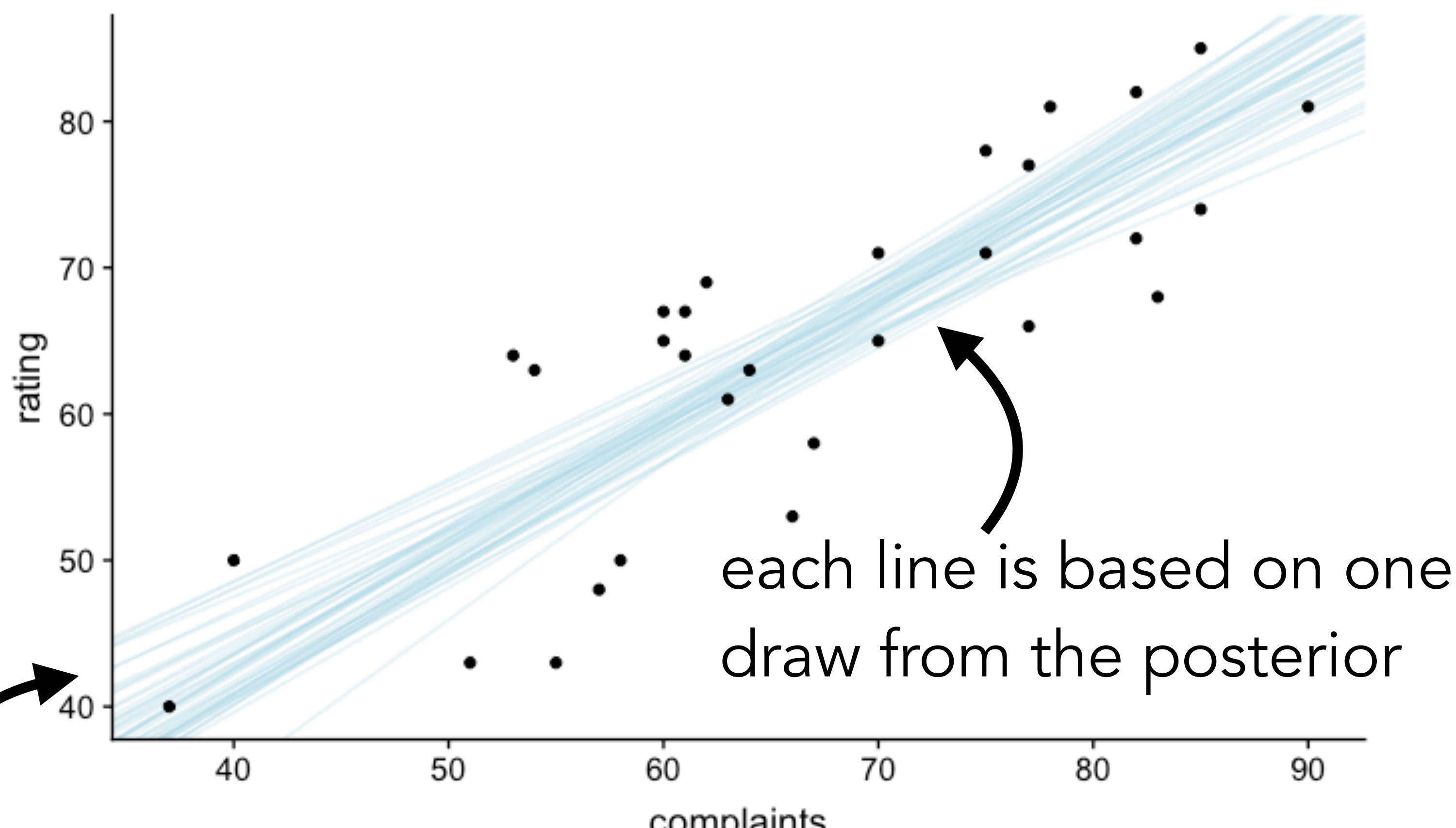
Visualize the posterior



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()
```

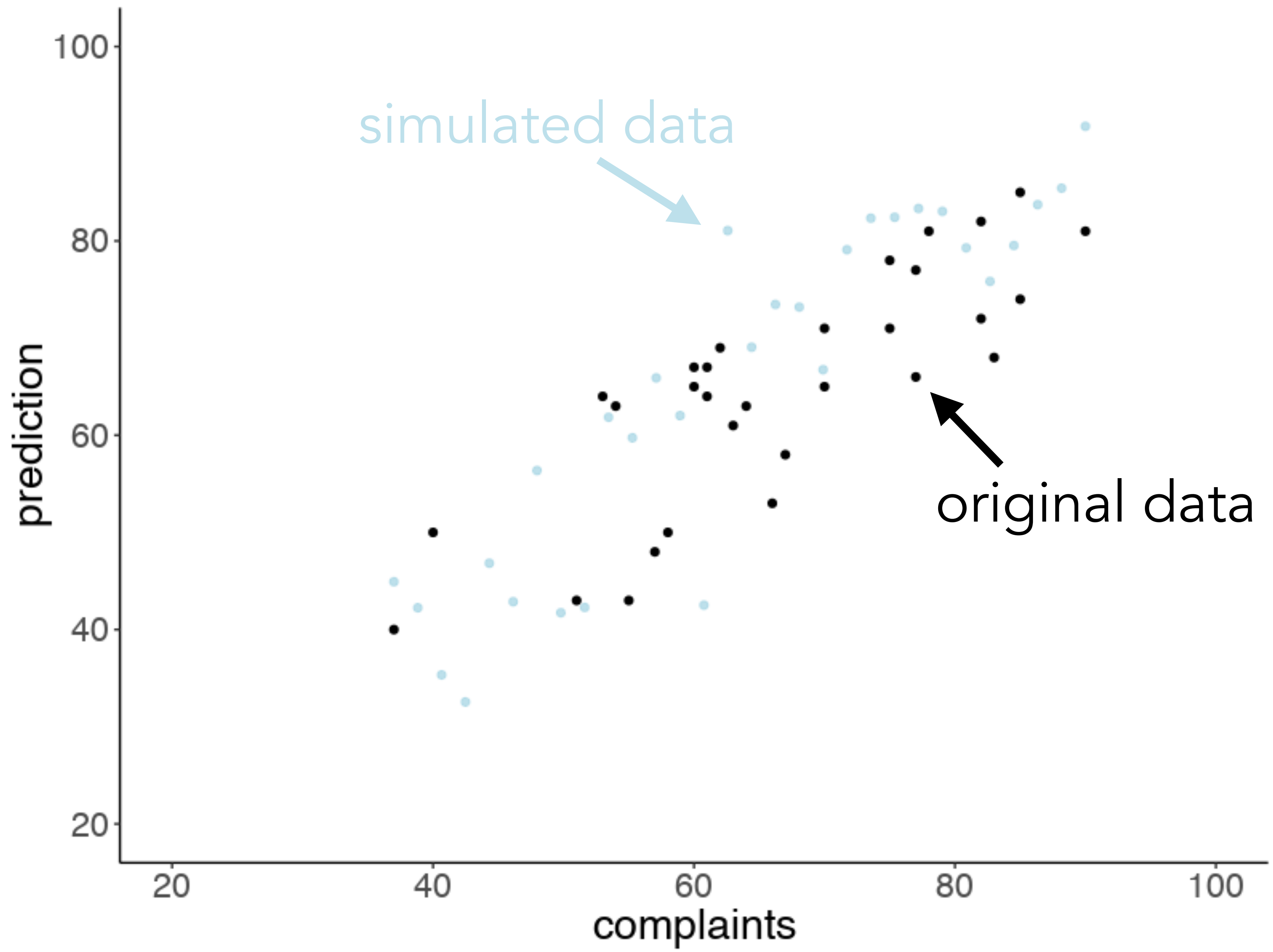


explains the "high" uncertainty about the intercept

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

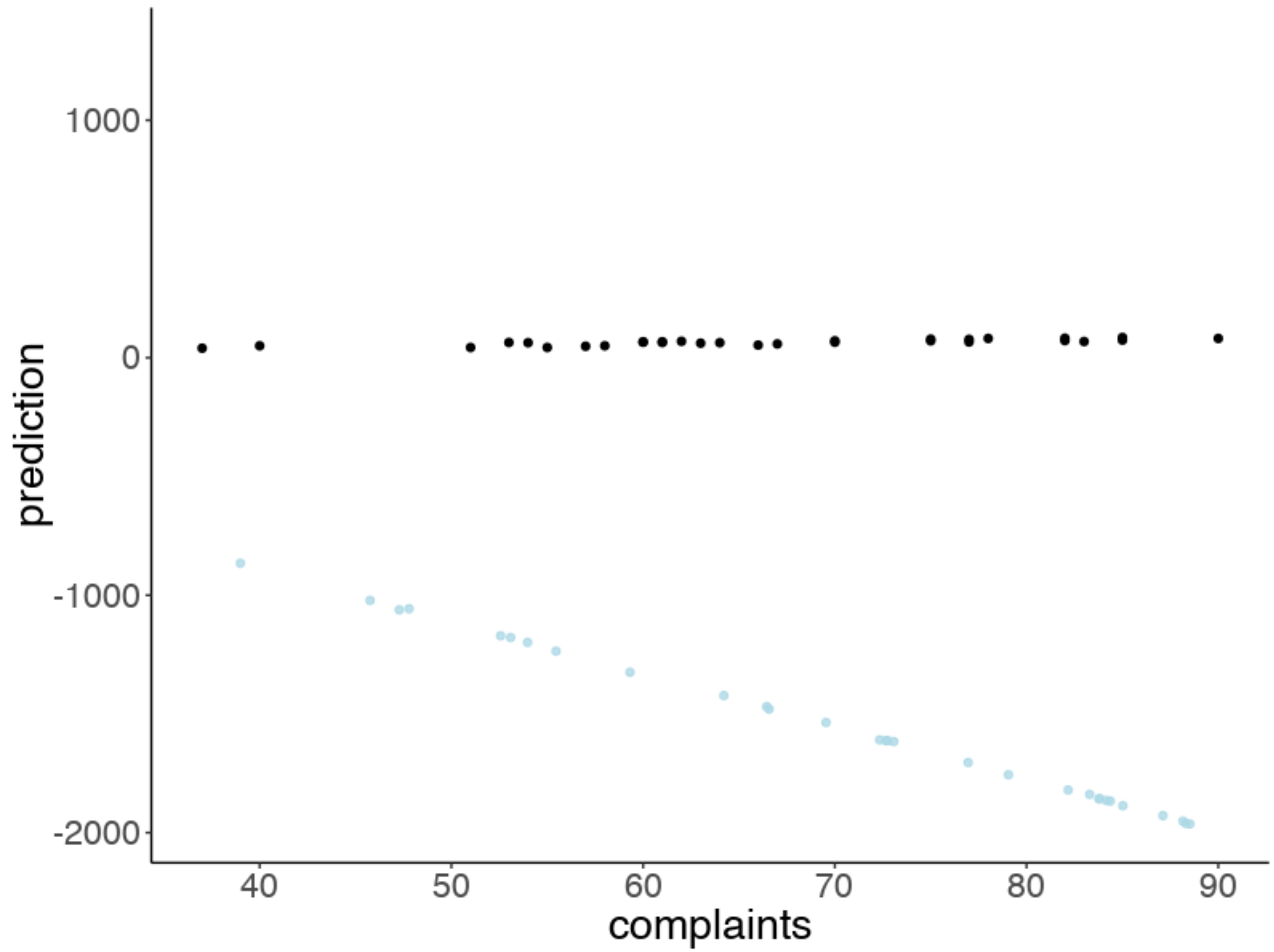
1. sample parameters from the **posterior distribution**
2. generate data using these parameters (using the likelihood function)



Hypothetical outcome plot

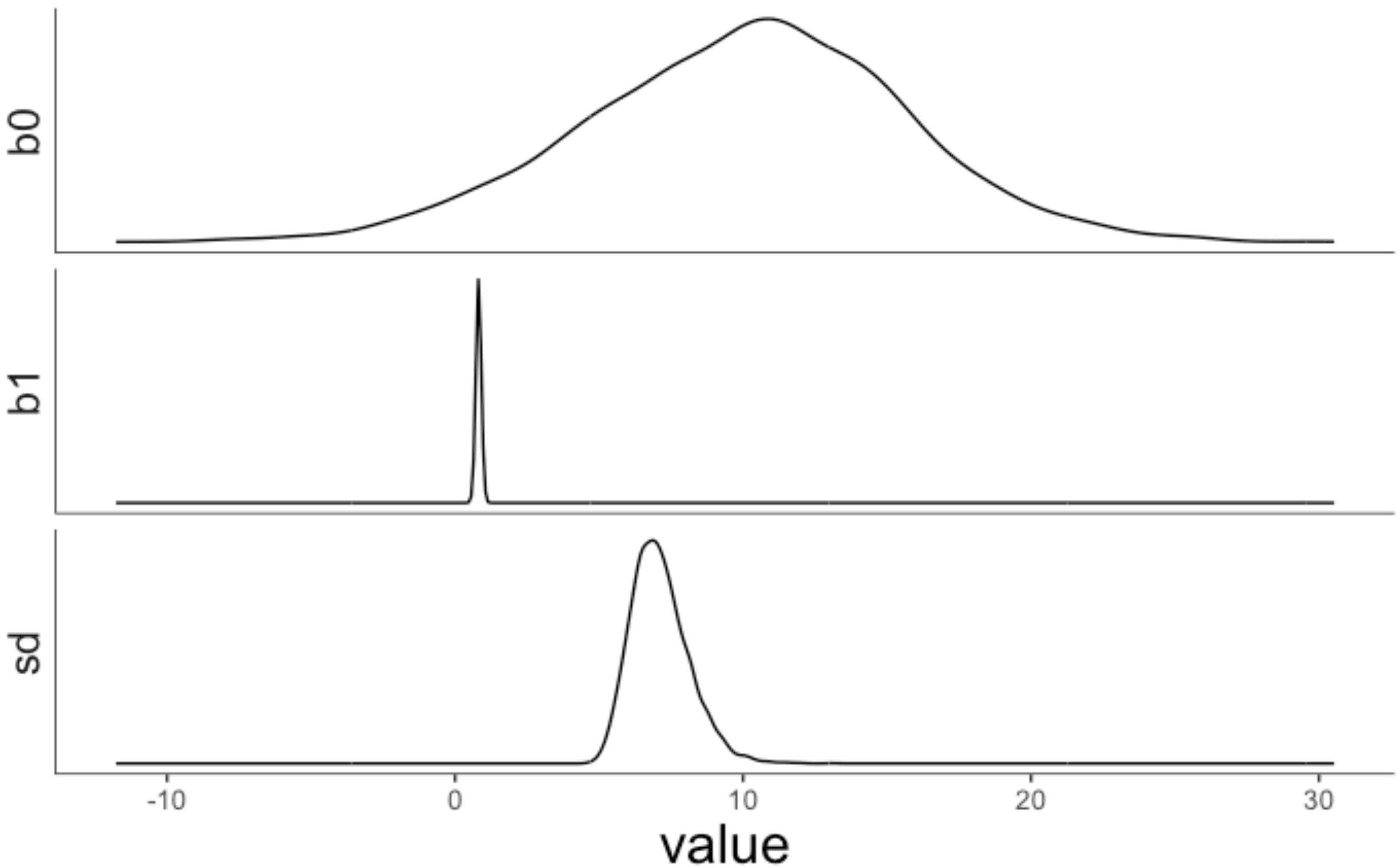
Prior predictive check

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



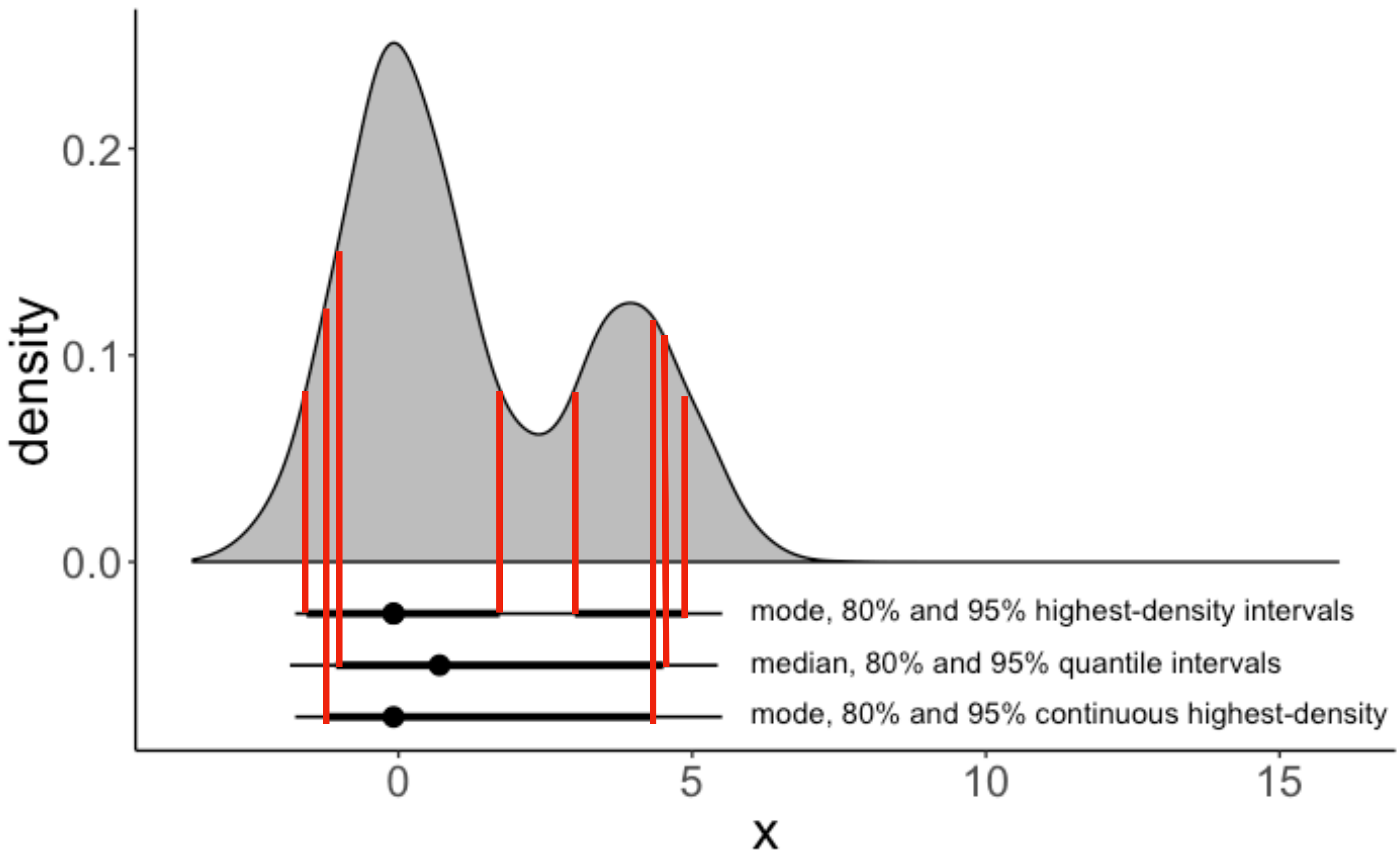
Hypothetical outcome plot

Summarizing results



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

Different kinds of credible intervals



Summary

- Quick recap: Generalized linear model
- 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

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

Start the presentation to see live content. Still no live content? Install the app or get help at **PollEv.com/app**

Thank you!