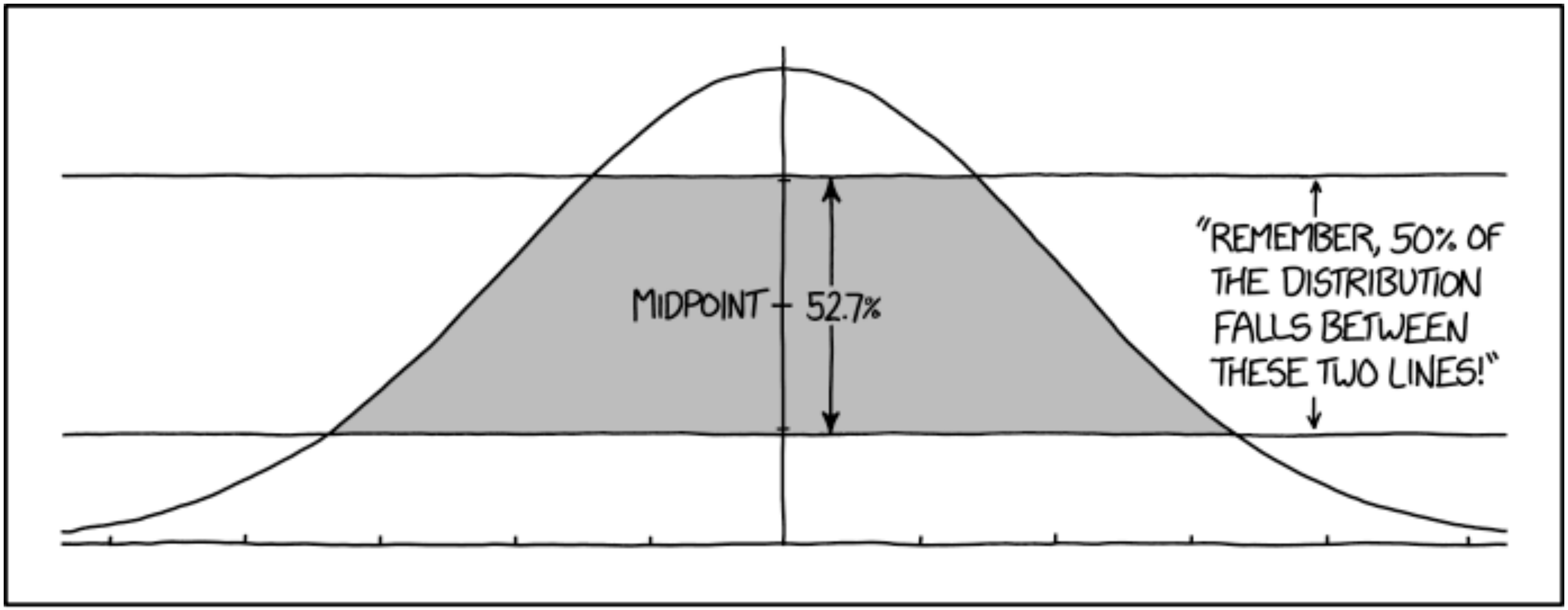
Bayesian data analysis 2



HOW TO ANNOY A STATISTICIAN

03/06/2019

Logistics

Final projects

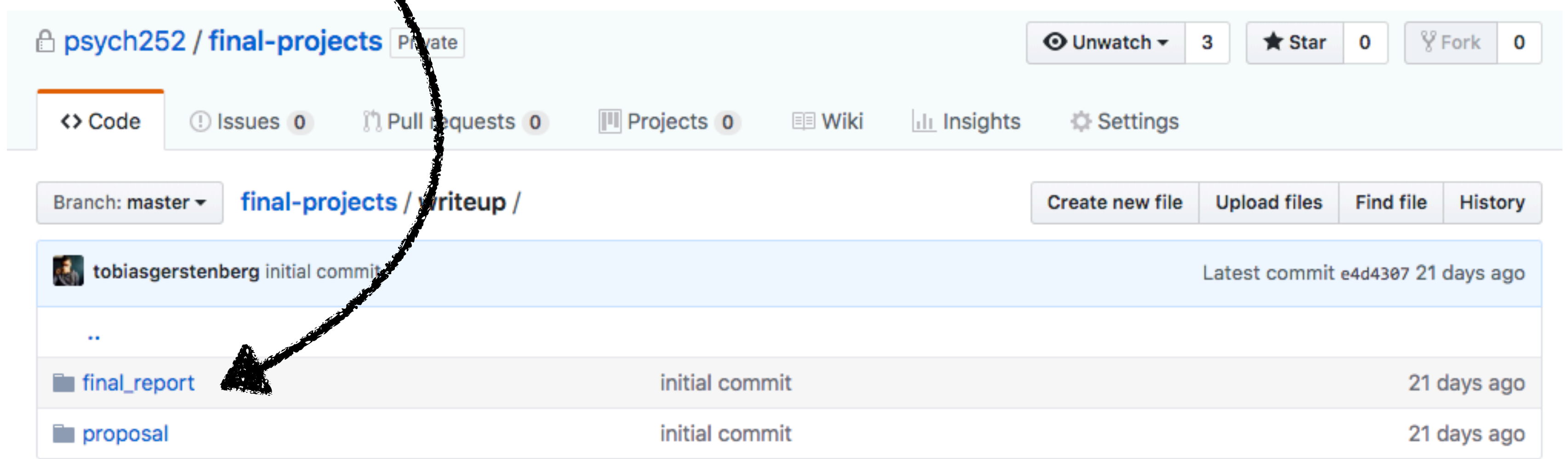
Project writeup

Project writeup

- Should have the form of a research report:
    1. Introduction / Background
        a) Research question / Hypotheses
    2. Methods
    3. Results
        a) Confirmatory analysis
        b) Exploratory analysis
    4. Discussion
- ~1000 - 2000 words long
- We expect more from teams compared to individuals (but need not be *more text*; more sophisticated analysis, visualizations, wrangling, ...)

Project writeup

- written as an RMarkdown document
- including all the code for wrangling, visualization, statistical analysis, and reporting
- save your writeup here



Project presentations

Project presentations

When/how will you present? *

- On March 21st (Final presentations day)
- On March 15th (Final class)
- On March 19th at 2pm (Stats instructors meeting)
- I will record the presentation and submit a video.

didn't mention this option initially

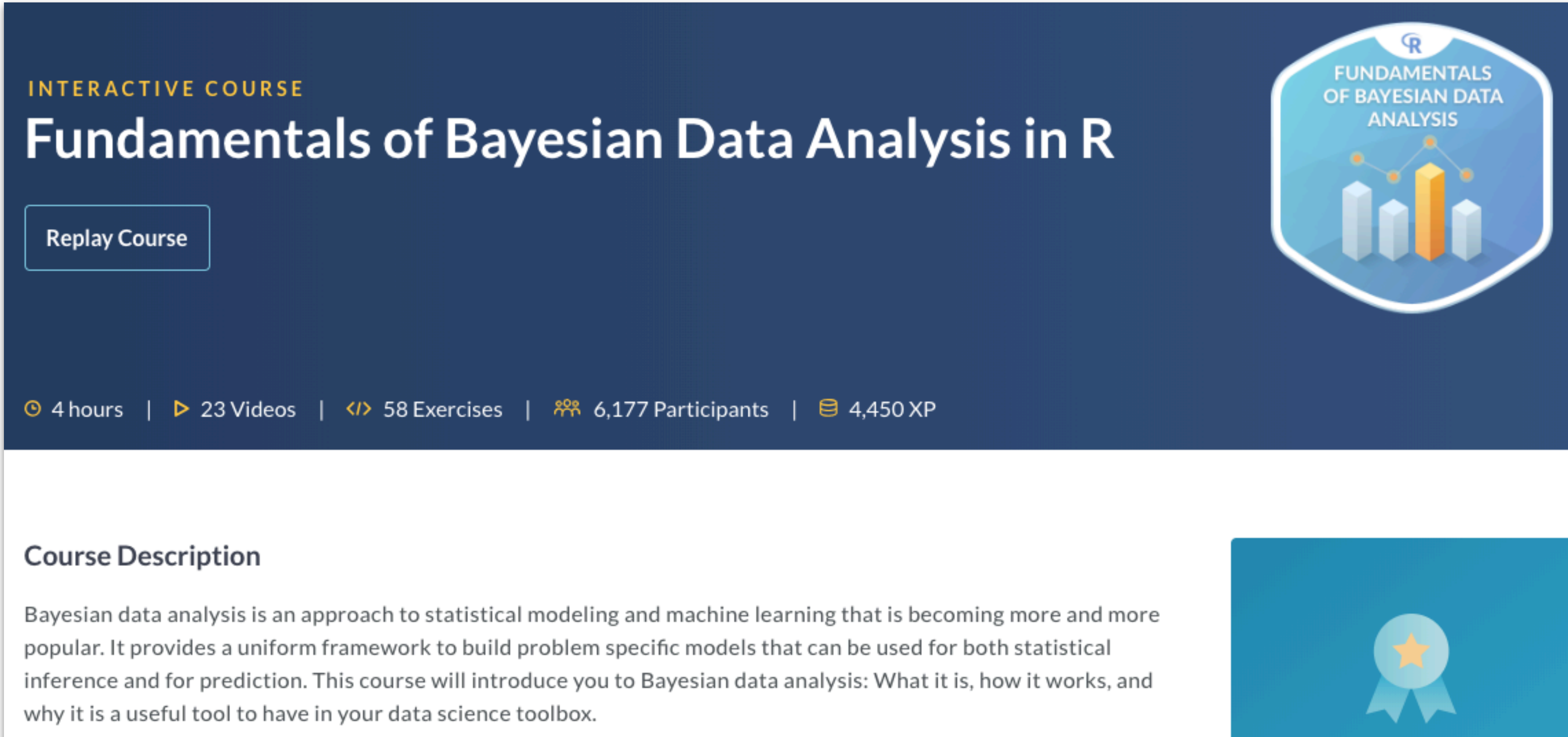
you can still change your response

survey link

https://tinyurl.com/psych252presentation

Datacamp

Datacamp course



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

Plan for today

- Flipping coins: Simple Bayesian inference
- Bayes' rule
  - Weighting of prior knowledge and evidence
  - Ingredients:
    - likelihood
    - prior
    - inference
- Doing Bayesian data analysis
  - A simple linear regression
  - Making comparisons

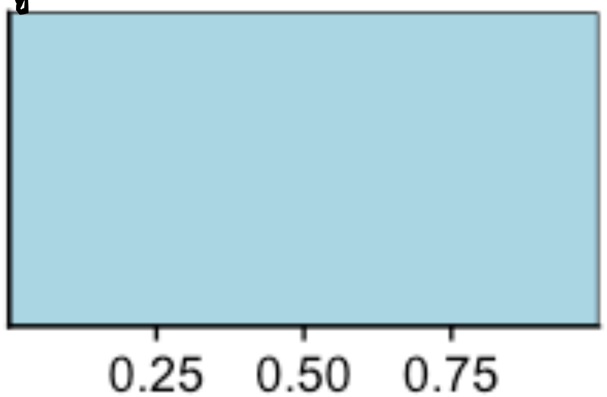
Flipping coins

Flipping coins

Learning from data

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

prior



Today's posterior is tomorrow's prior.

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

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

posterior

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

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

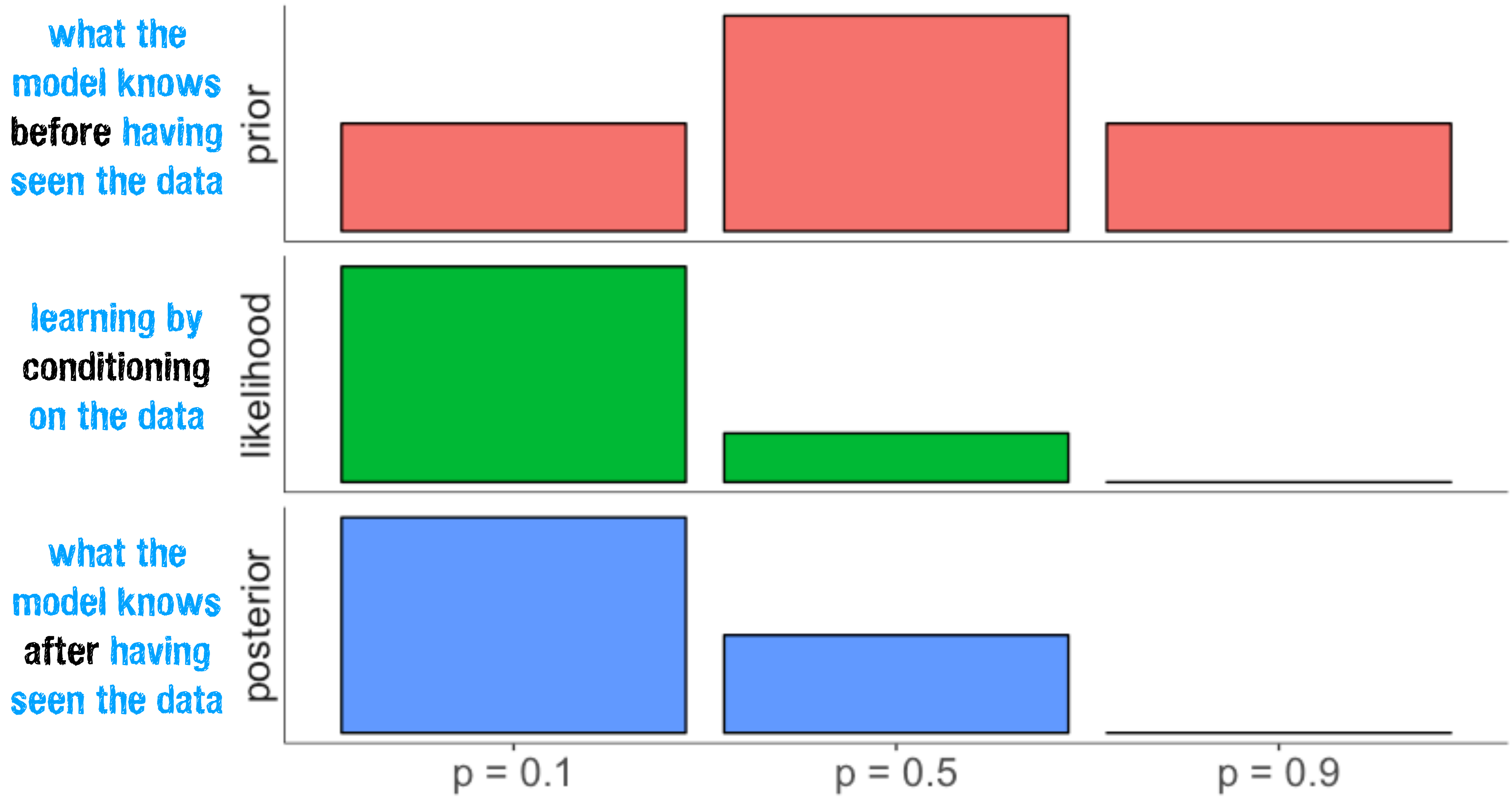
posterior

theta	prior	likelihood	posterior
0.1	0.25	0.19	0.69
0.5	0.50	0.04	0.31
0.9	0.25	0.00	0.00

Coin flip example

data: #8 tails, #2 heads

Which coin was flipped?

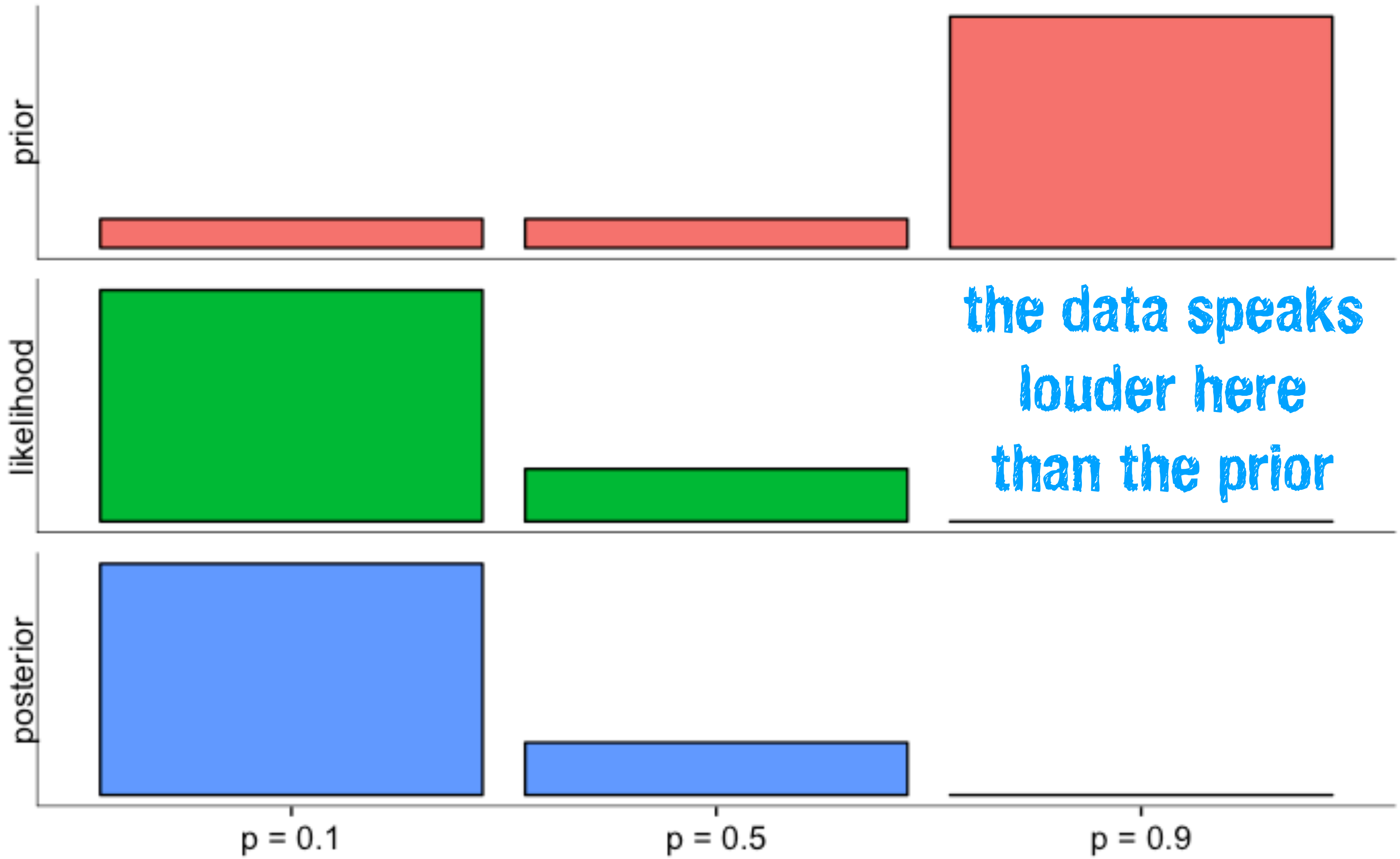


posterior = multiplicative weighting of prior and likelihood

Coin flip example

data: #8 tails, #2 heads

Which coin was flipped?

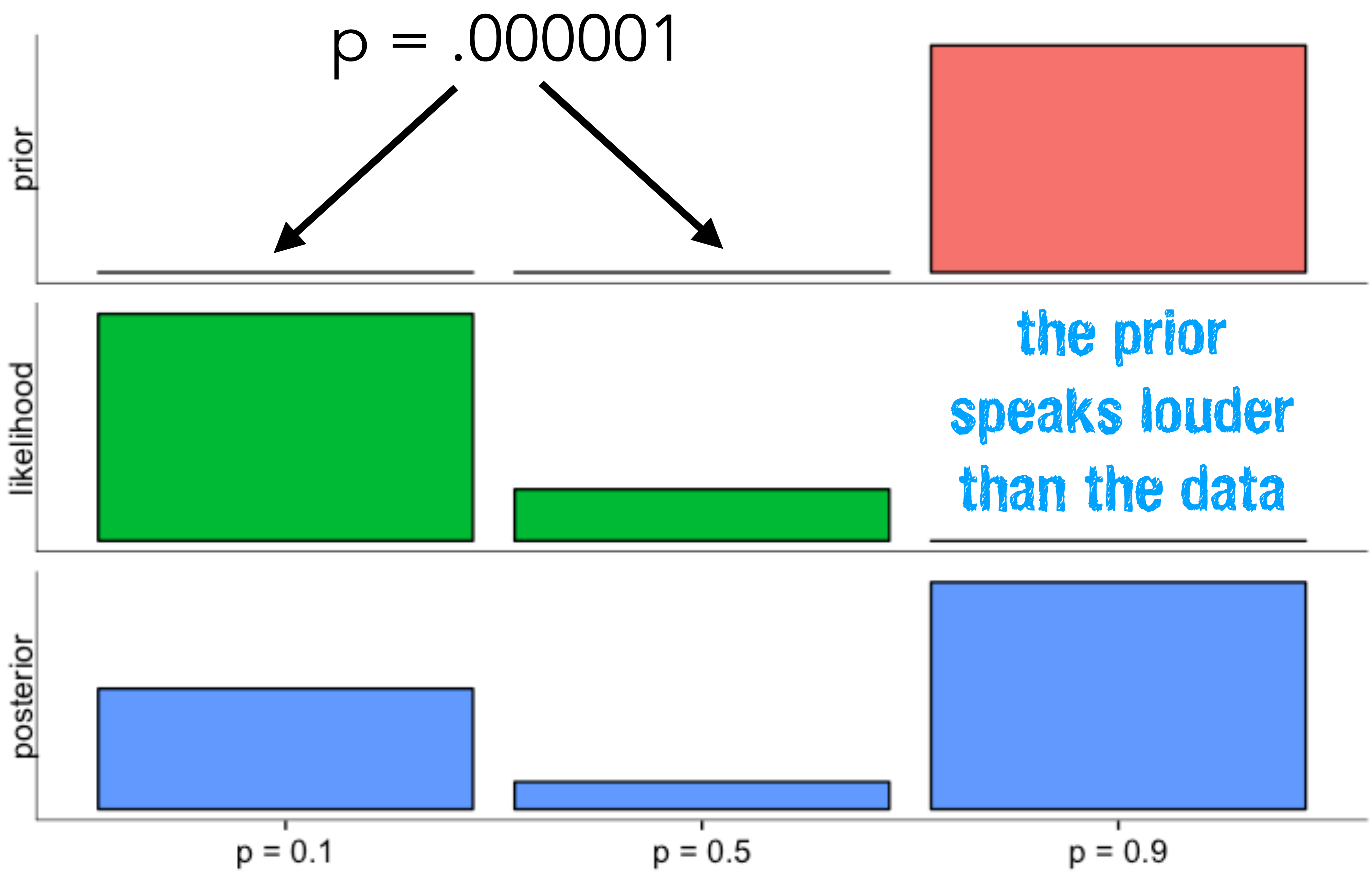


posterior = multiplicative weighting of prior and likelihood

Coin flip example

data: #8 tails, #2 heads

Which coin was flipped?



posterior = multiplicative weighting of prior and likelihood

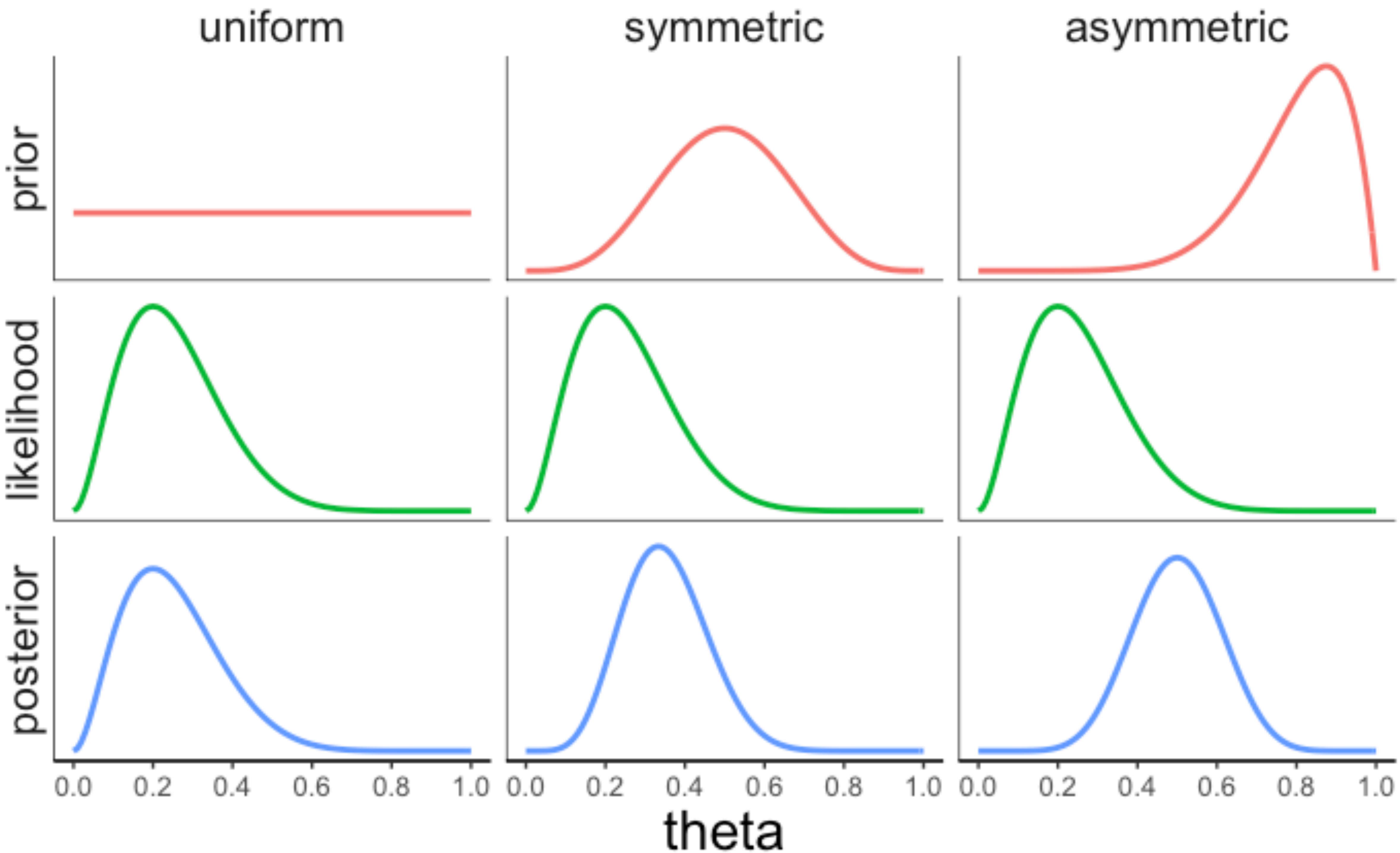
What affects the posterior?

What affects the posterior?

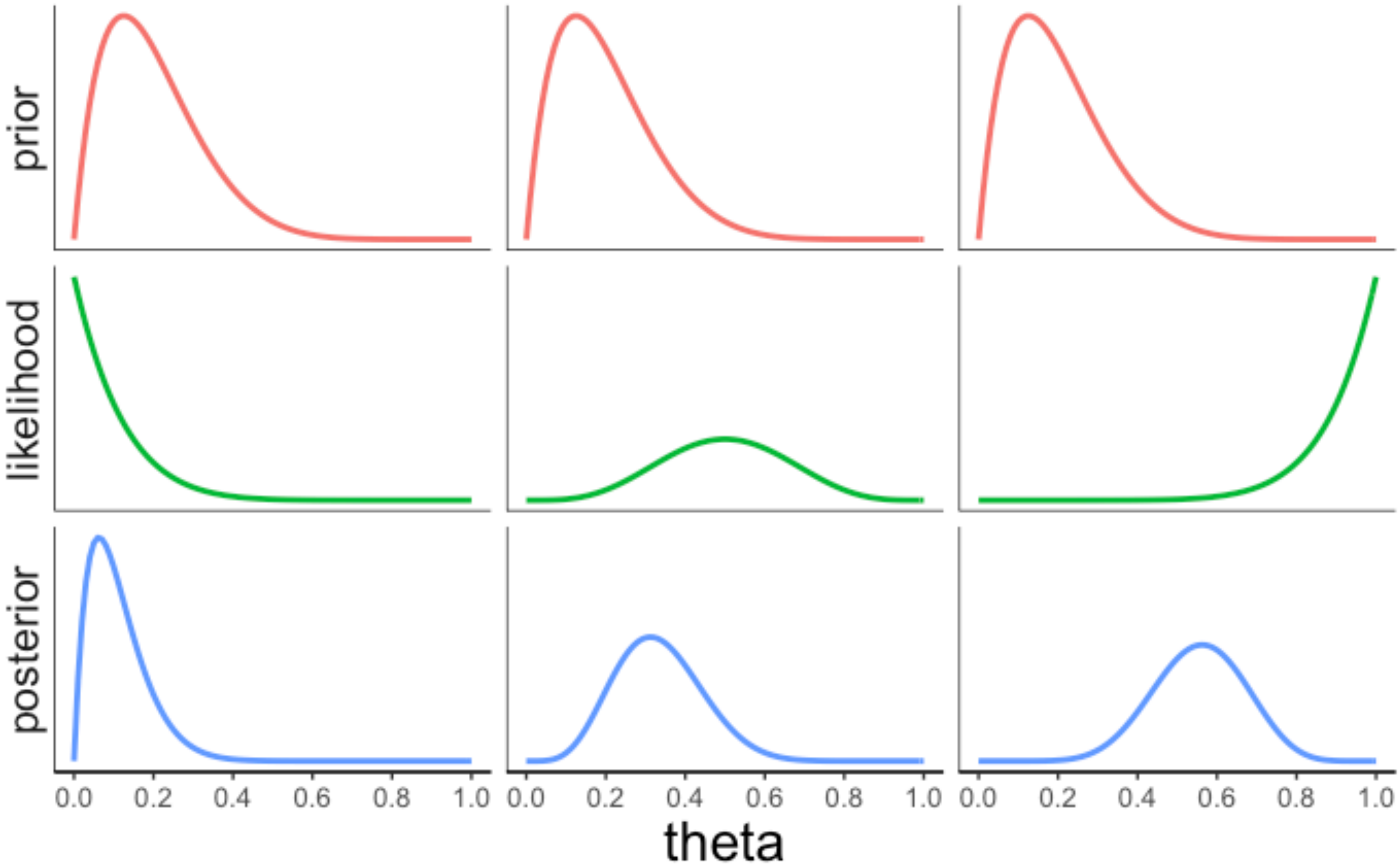
1. the prior over hypotheses
2. the likelihood of the data given each hypothesis

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

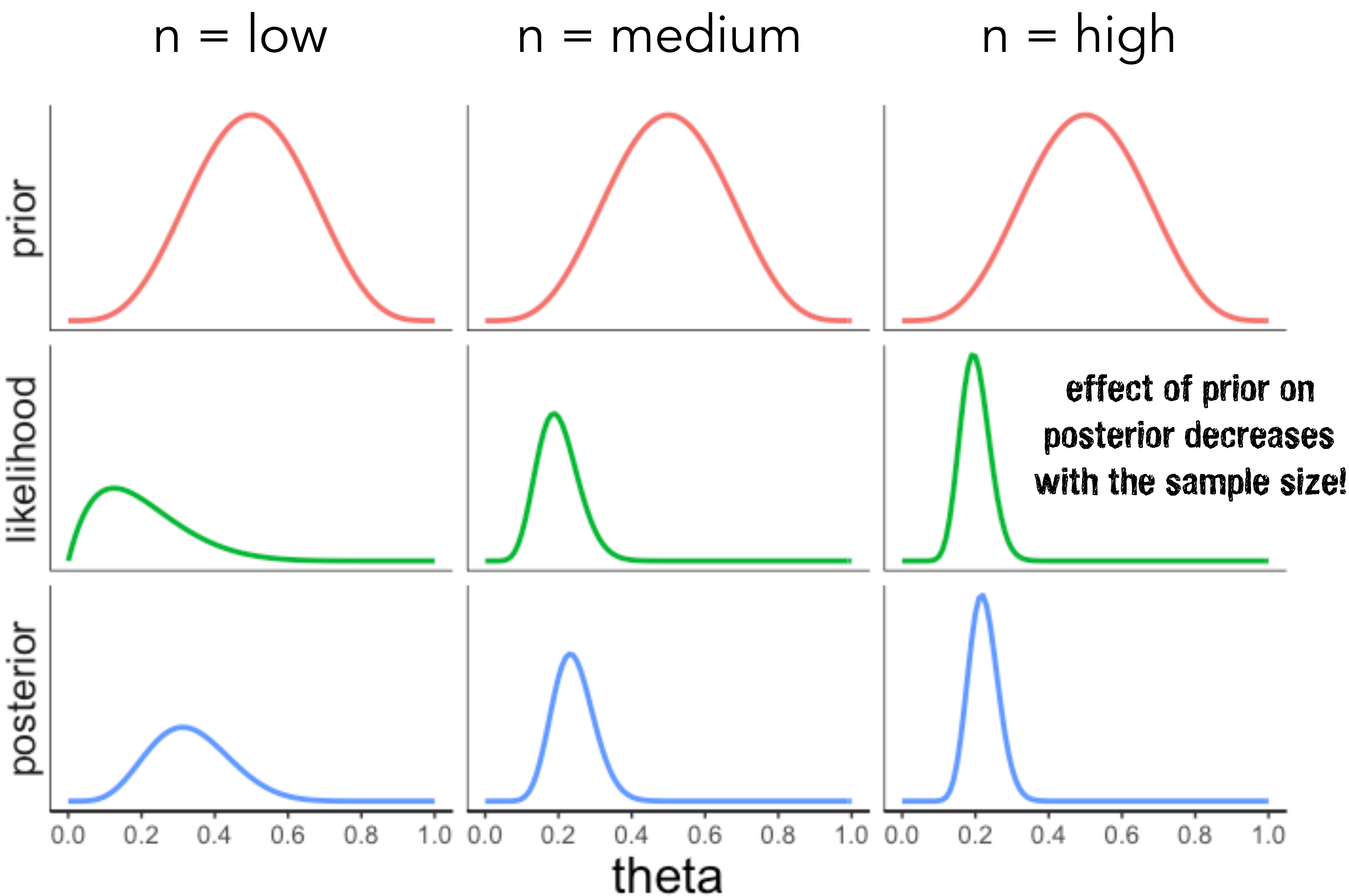
The effect of the **prior**



The effect of the **likelihood**



The effect of **sample size**



Ingredients

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

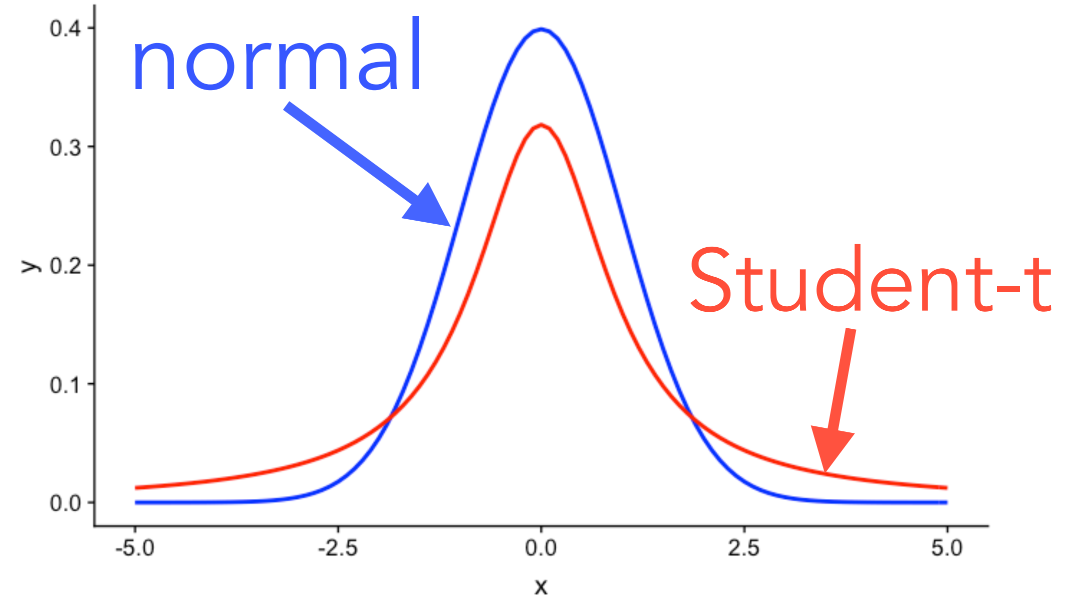
- **What probabilistic model describes best how the data were generated?**
- How to build a (Bayesian) model?
  - What real-life behavior should the model explain?
  - What assumptions can you make about the behavior?
    - What's the nature of your dependent variable (e.g. binary, ordered, continuous)?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

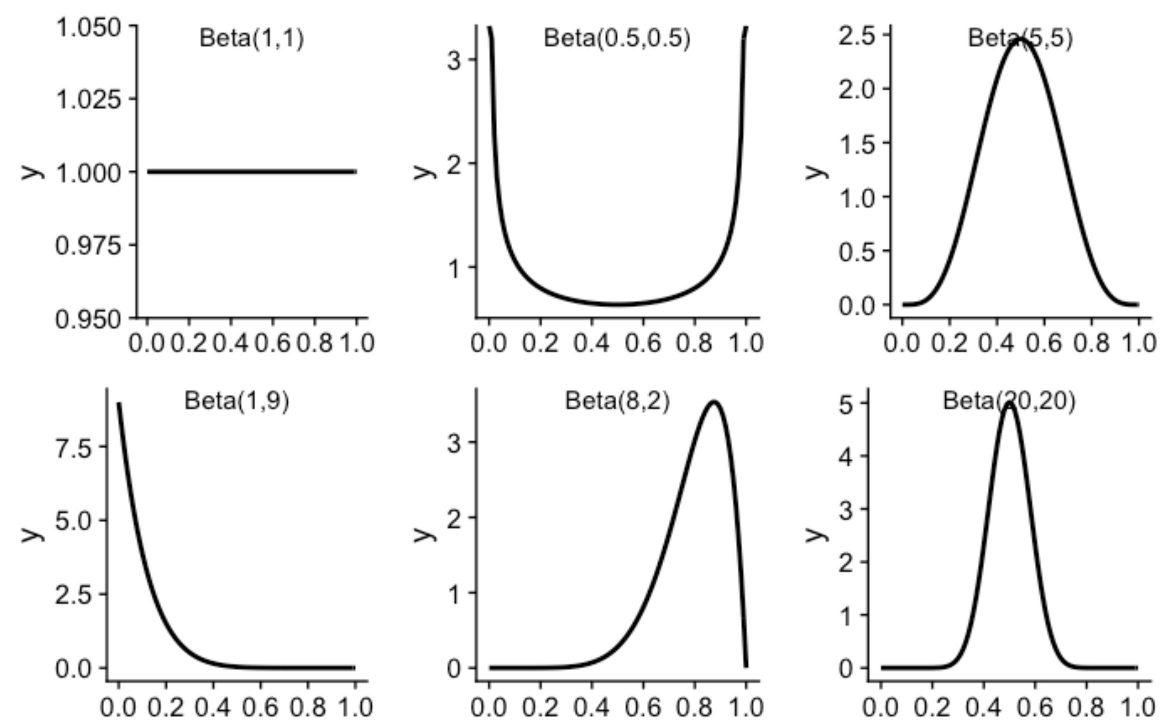


Prior

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

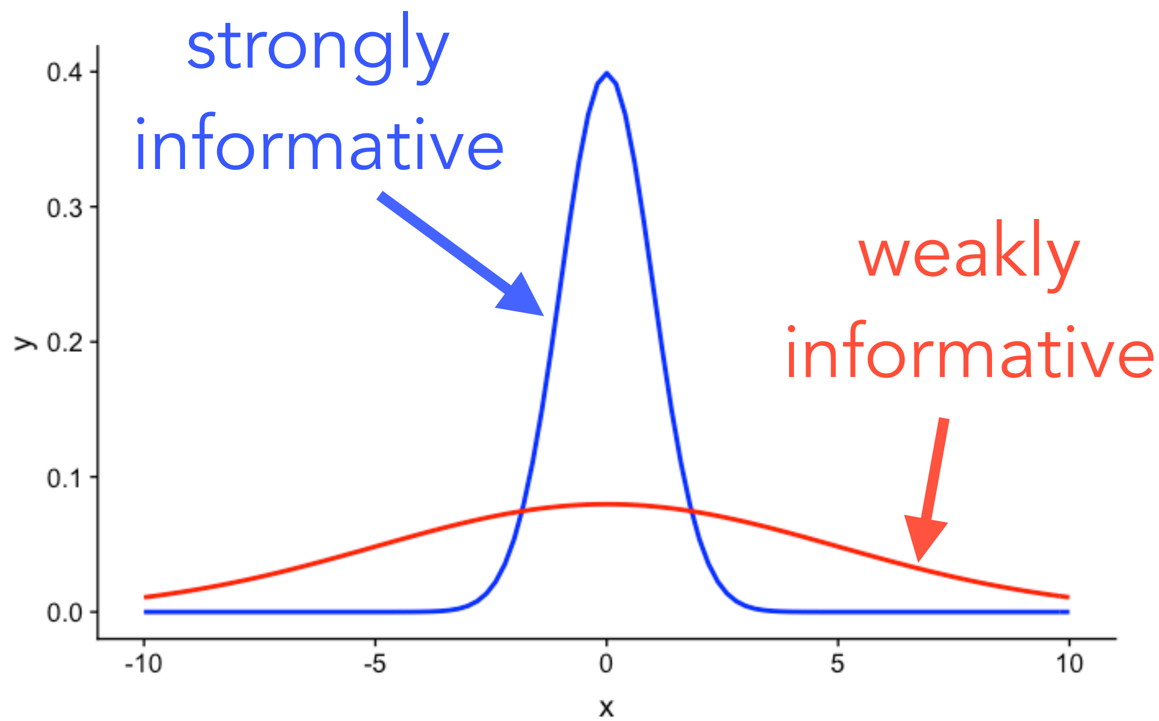
Prior

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



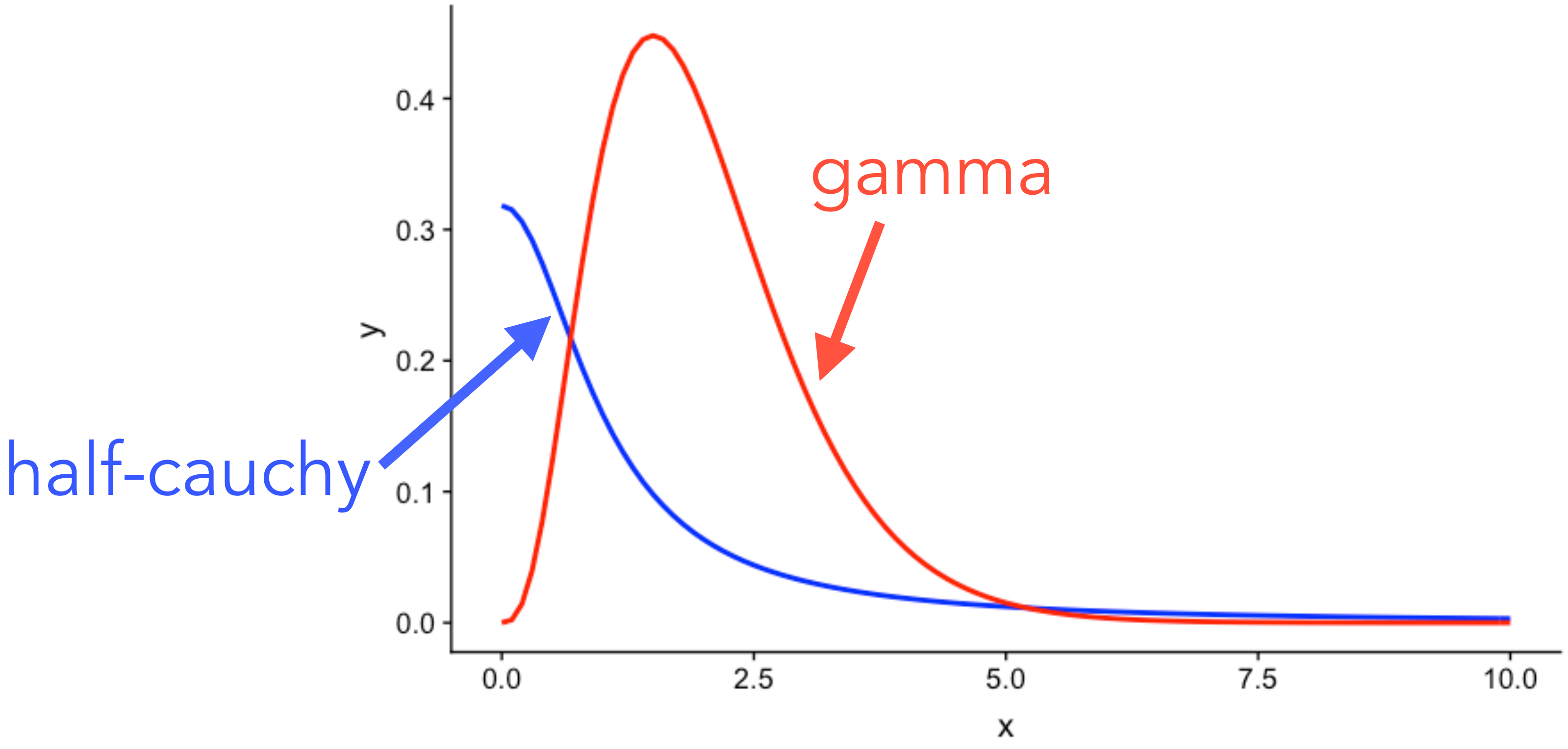
Prior

- **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

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

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

sum over all possibilities

Continuous hypothesis space

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

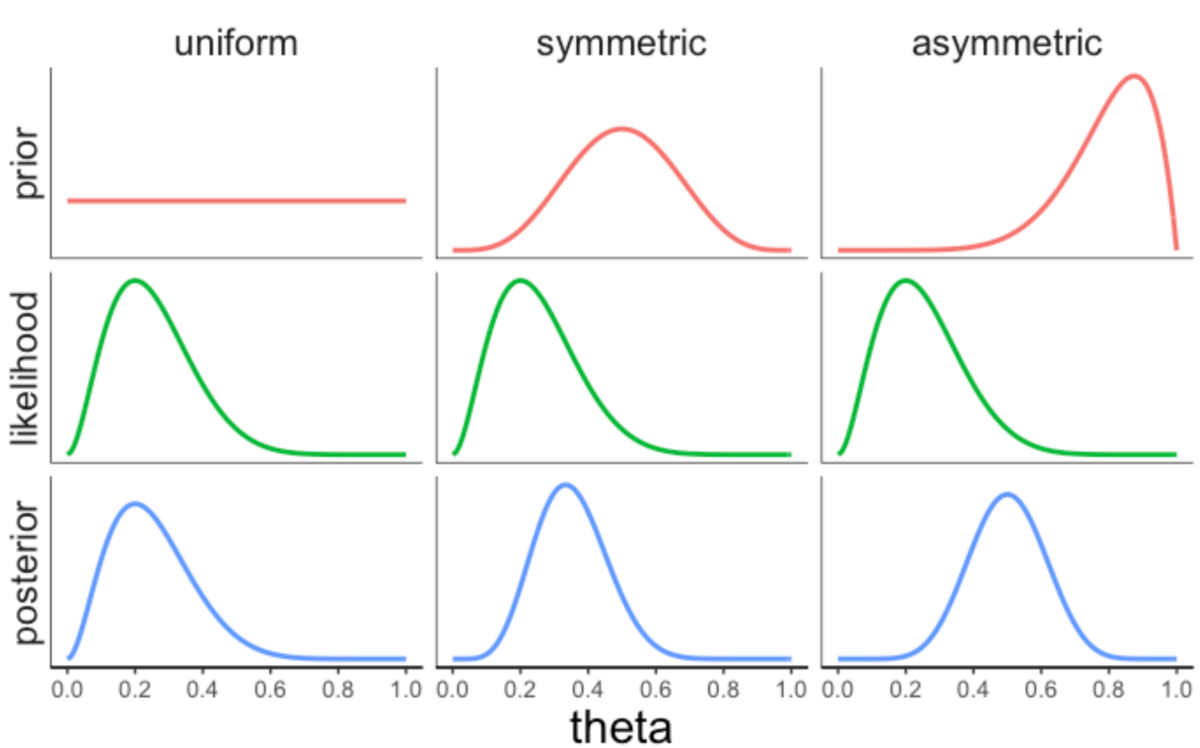
integral over all possibilities

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)) %>%
  gather("prior_index", "prior", -theta) %>%
  mutate(likelihood = dbinom(sum(data == 1),
                             size = length(data),
                             prob = theta)) %>%
  group_by(prior_index) %>%
  mutate(posterior = likelihood * prior / sum(likelihood * prior)) %>%
  ungroup() %>%
  gather("index", "value", -c(theta, prior_index))
```

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

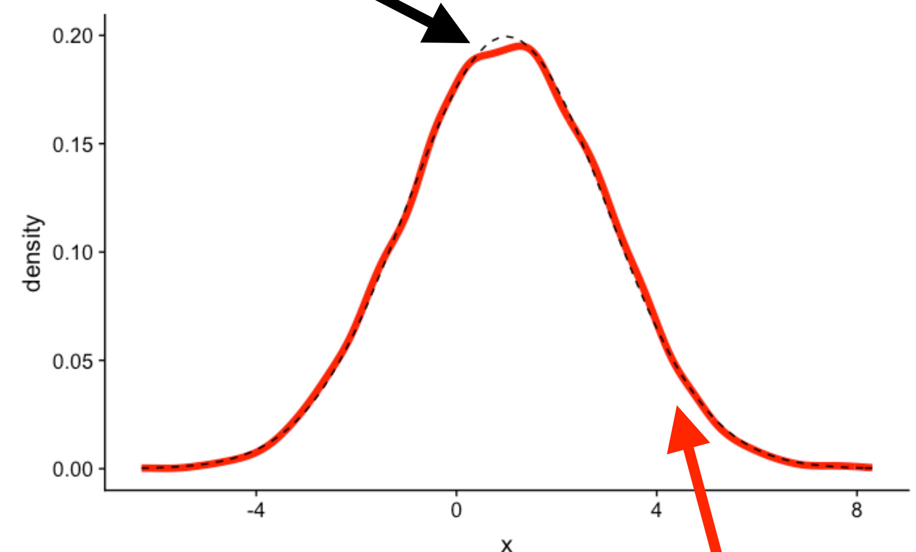
Inference via sampling

- 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)
```

theoretical distribution



distribution based on samples

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

both methods yield ≅63%

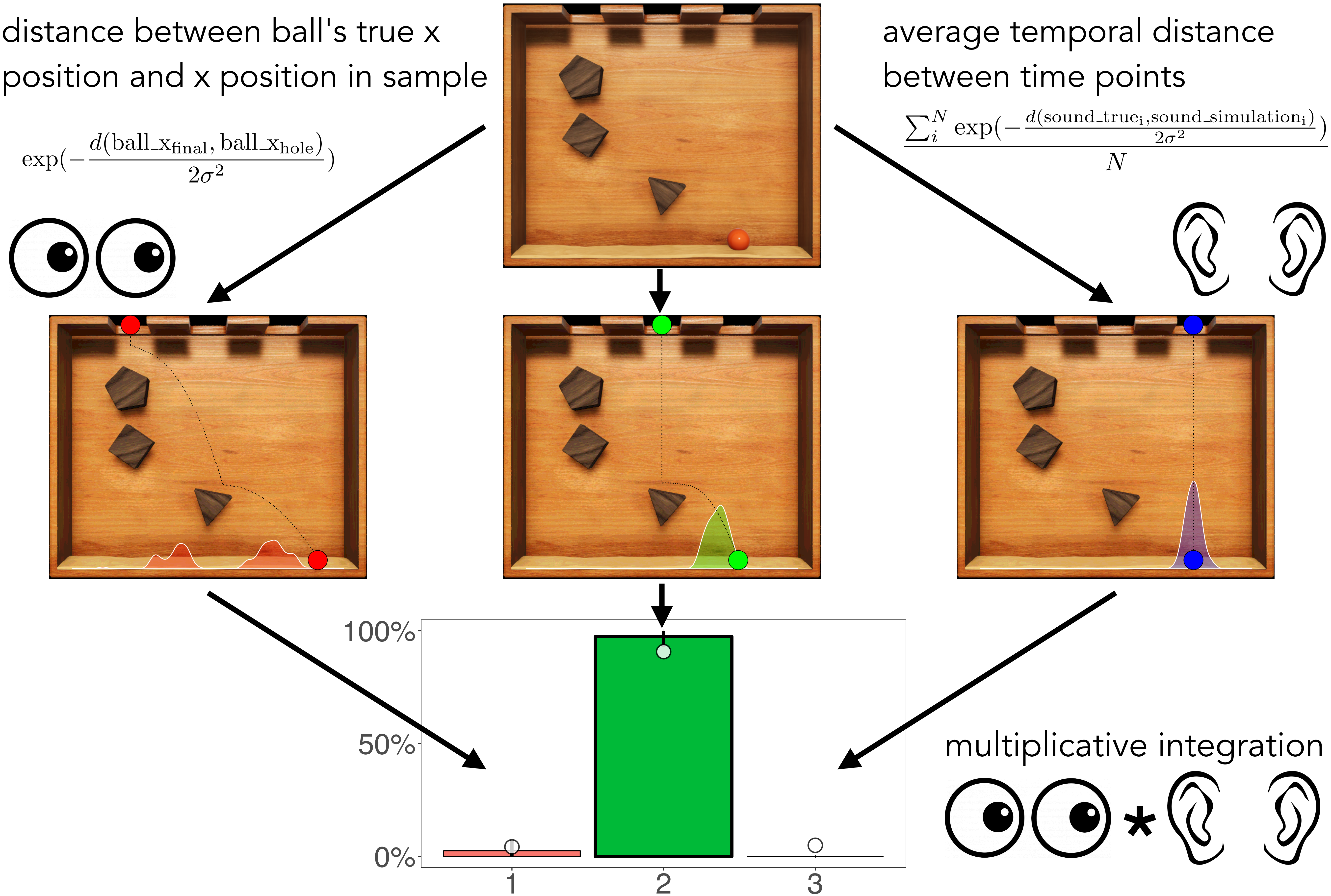
Inference via sampling

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

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

average temporal distance between time points

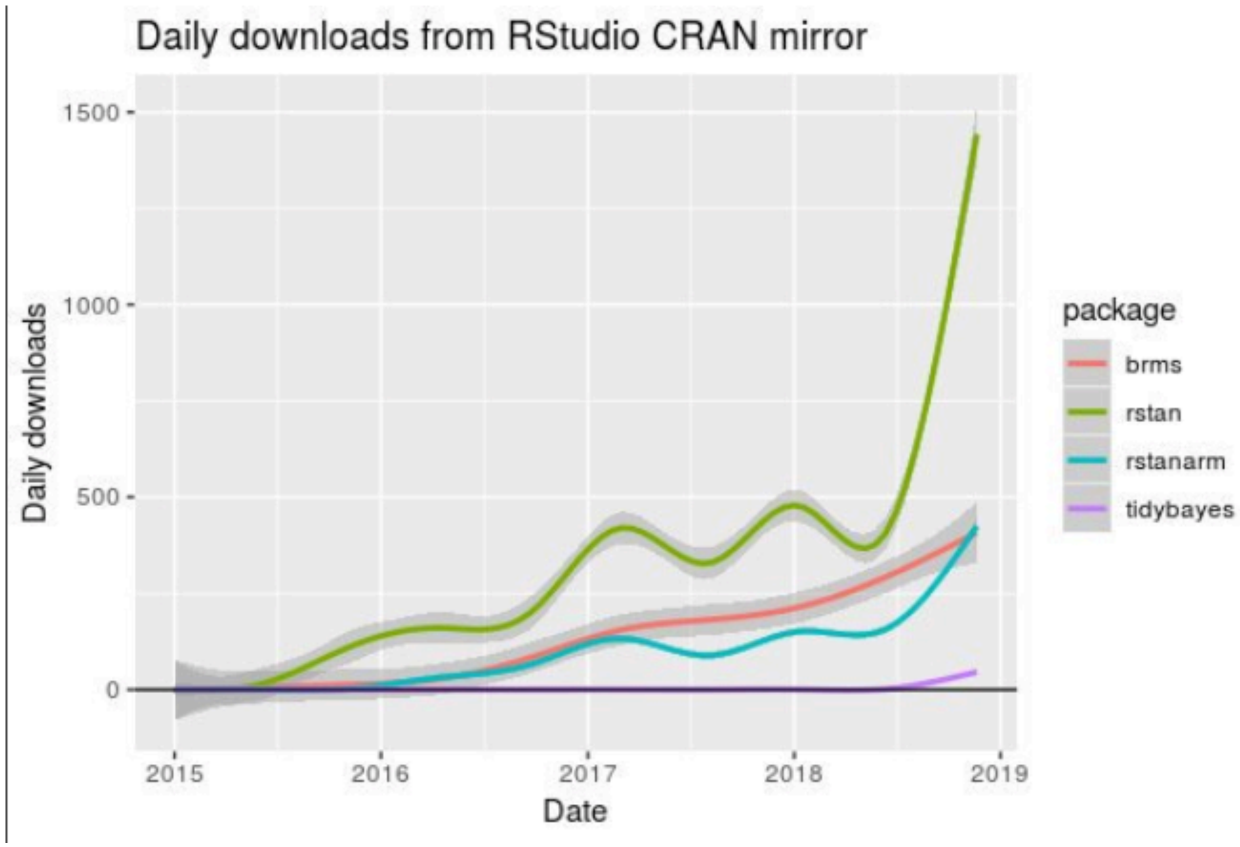
$$\frac{\sum_i^N \exp\left(-\frac{d(\text{sound\_true}_{\text{i}}, \text{sound\_simulation}_{\text{i}})}{2\sigma^2}\right)}{N}$$



multiplicative integration

Inference via sampling

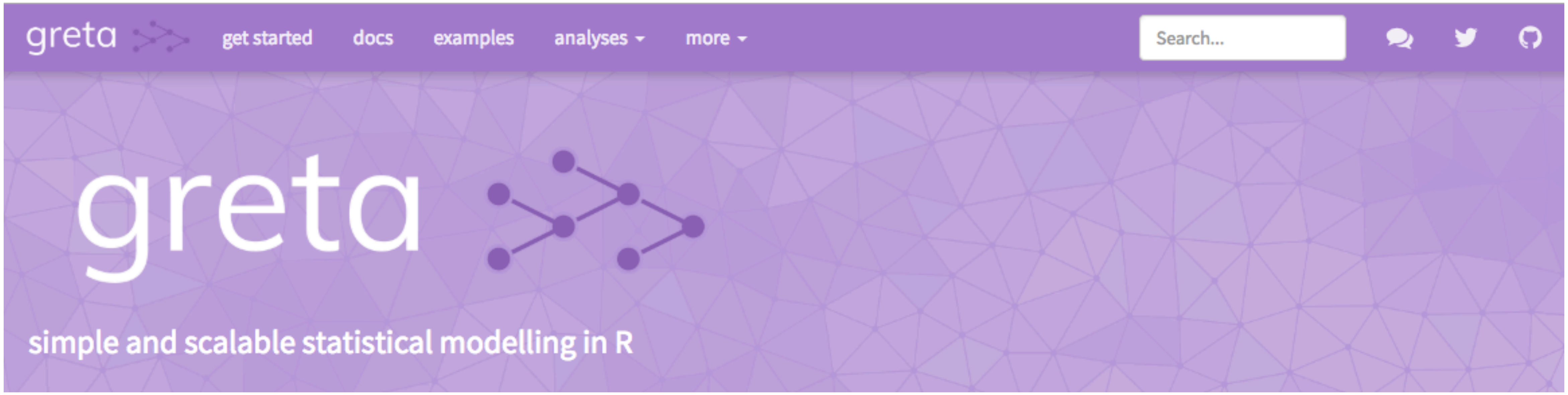
- Bayesian data analysis is becoming more popular because:
  - computers are getting more powerful
  - inference techniques are getting better
  - software packages become easier to use



Doing Bayesian data analysis

Software packages

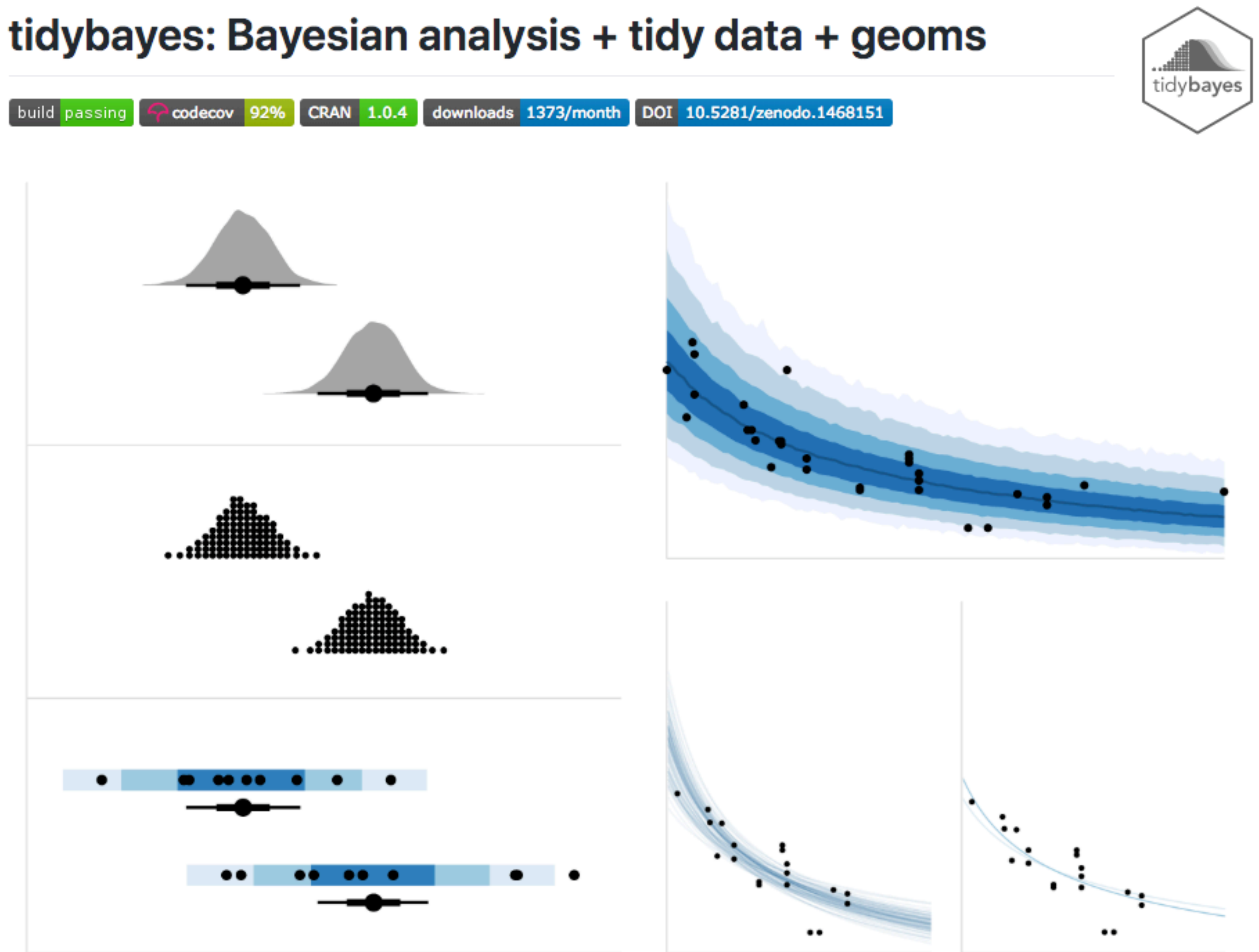
```
library("greta")
```



- 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

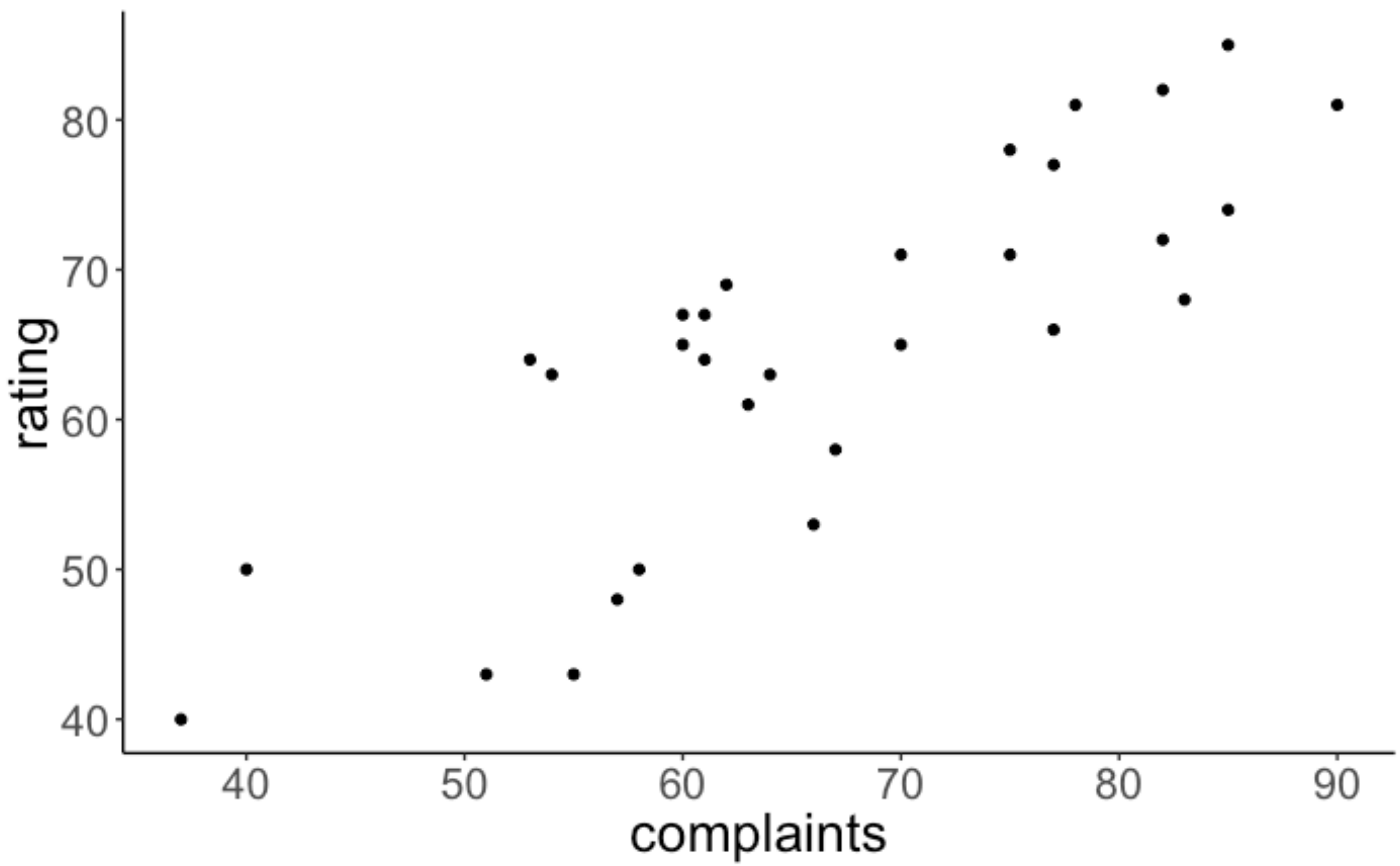
`library("tidybayes")`



- 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?



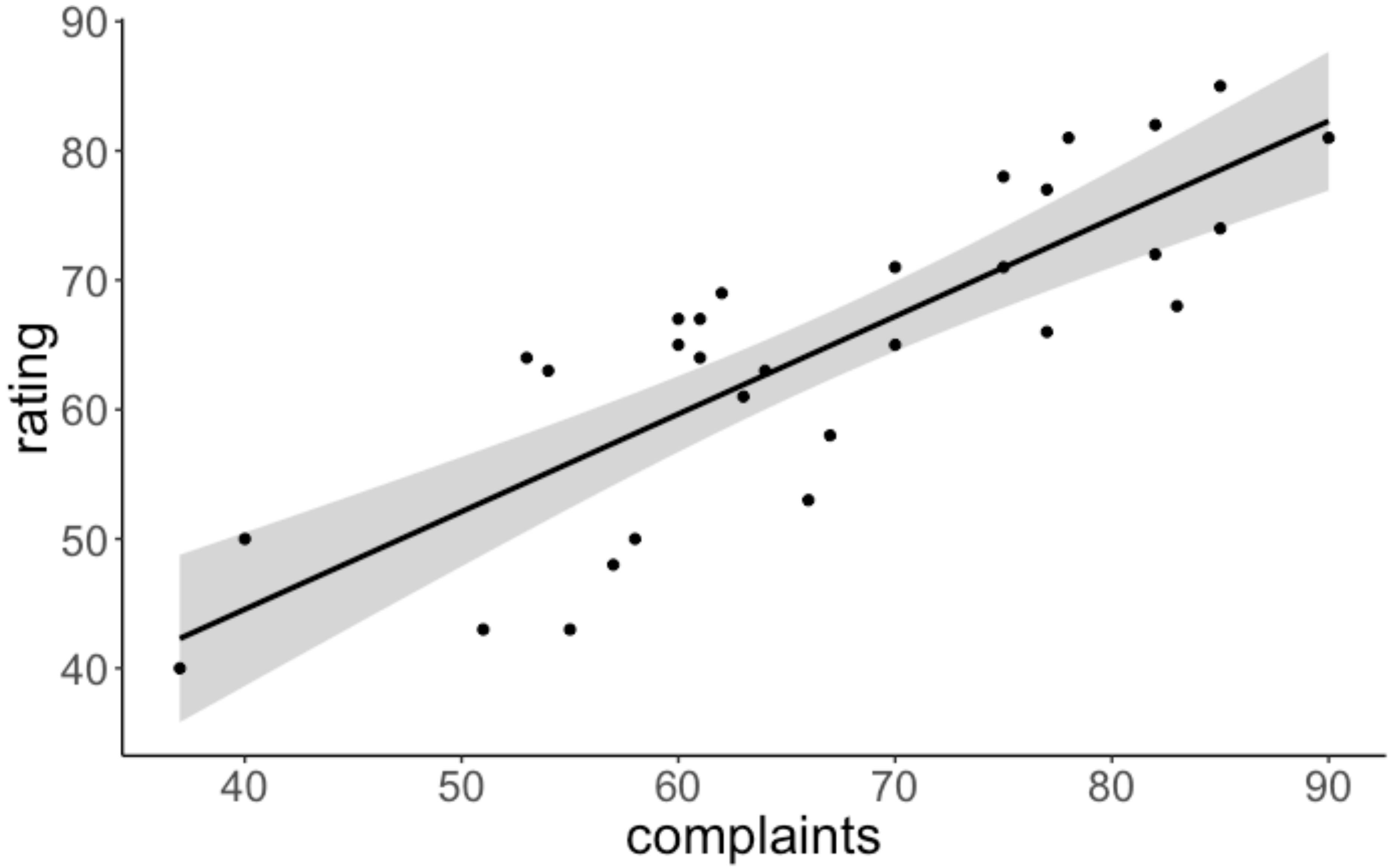
Frequentist analysis

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)
```

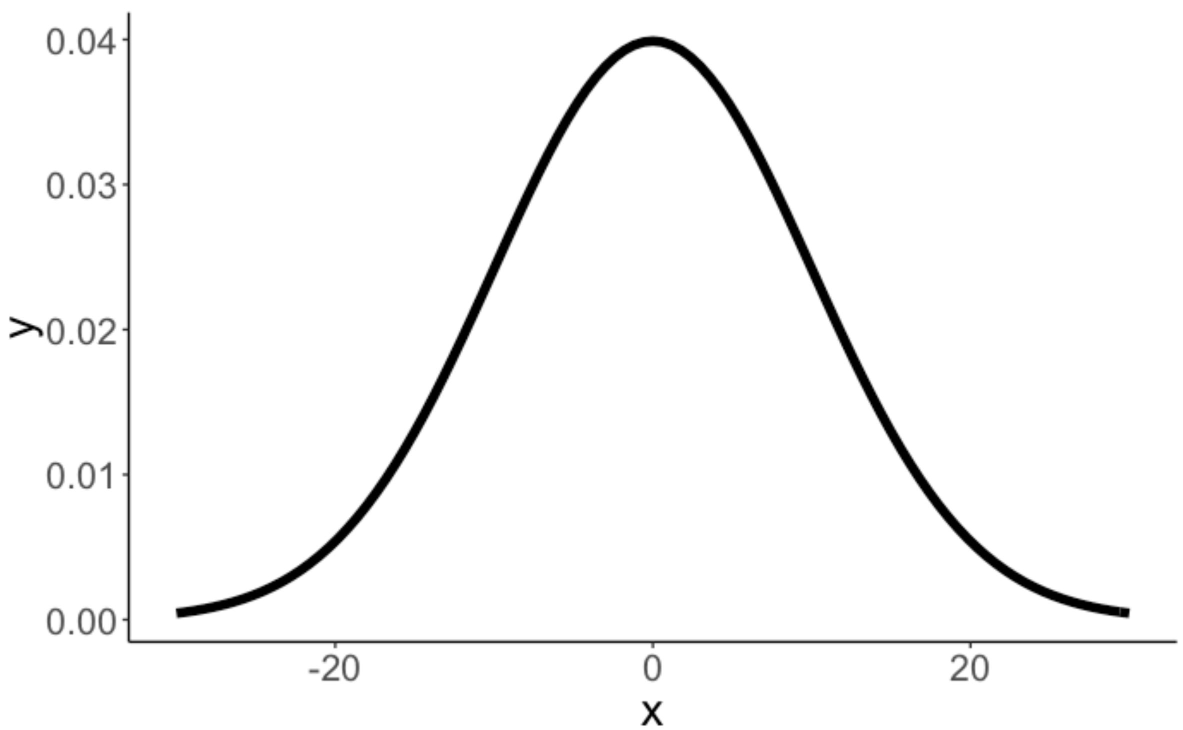
priors

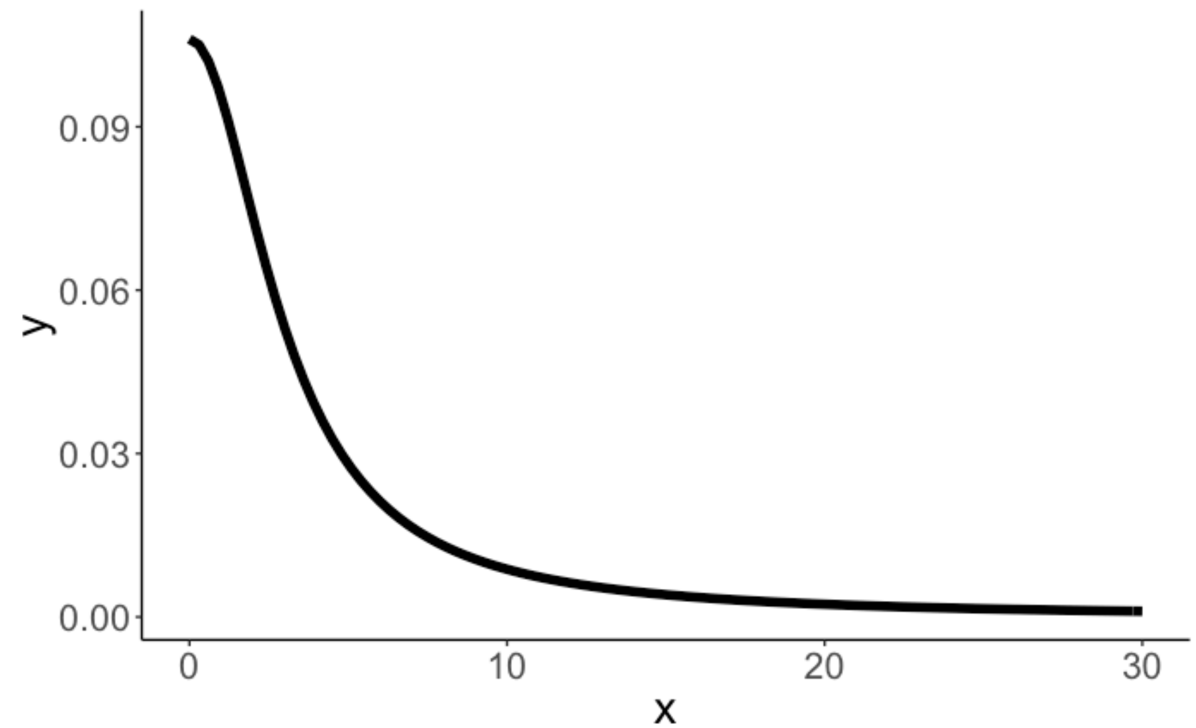
linear combination

Gaussian likelihood

build the model

Priors





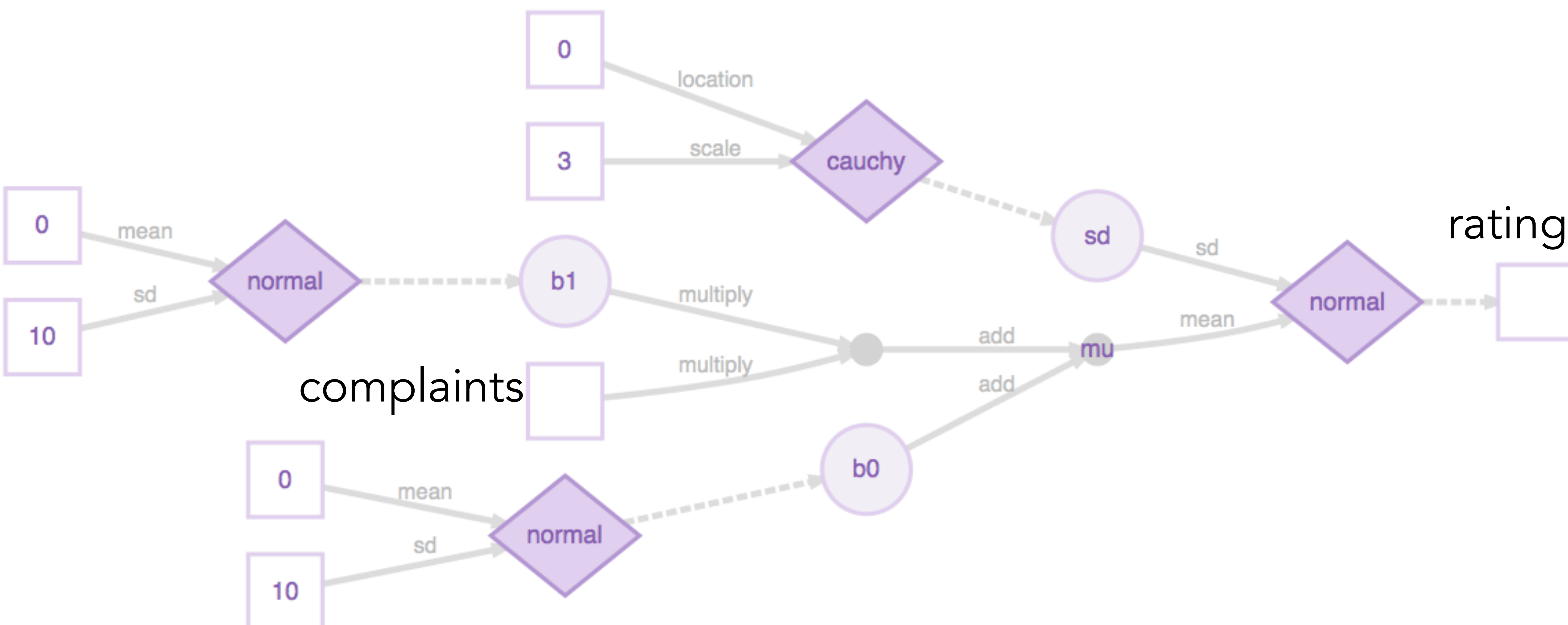
Gaussian prior on intercept and coefficient

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)
```



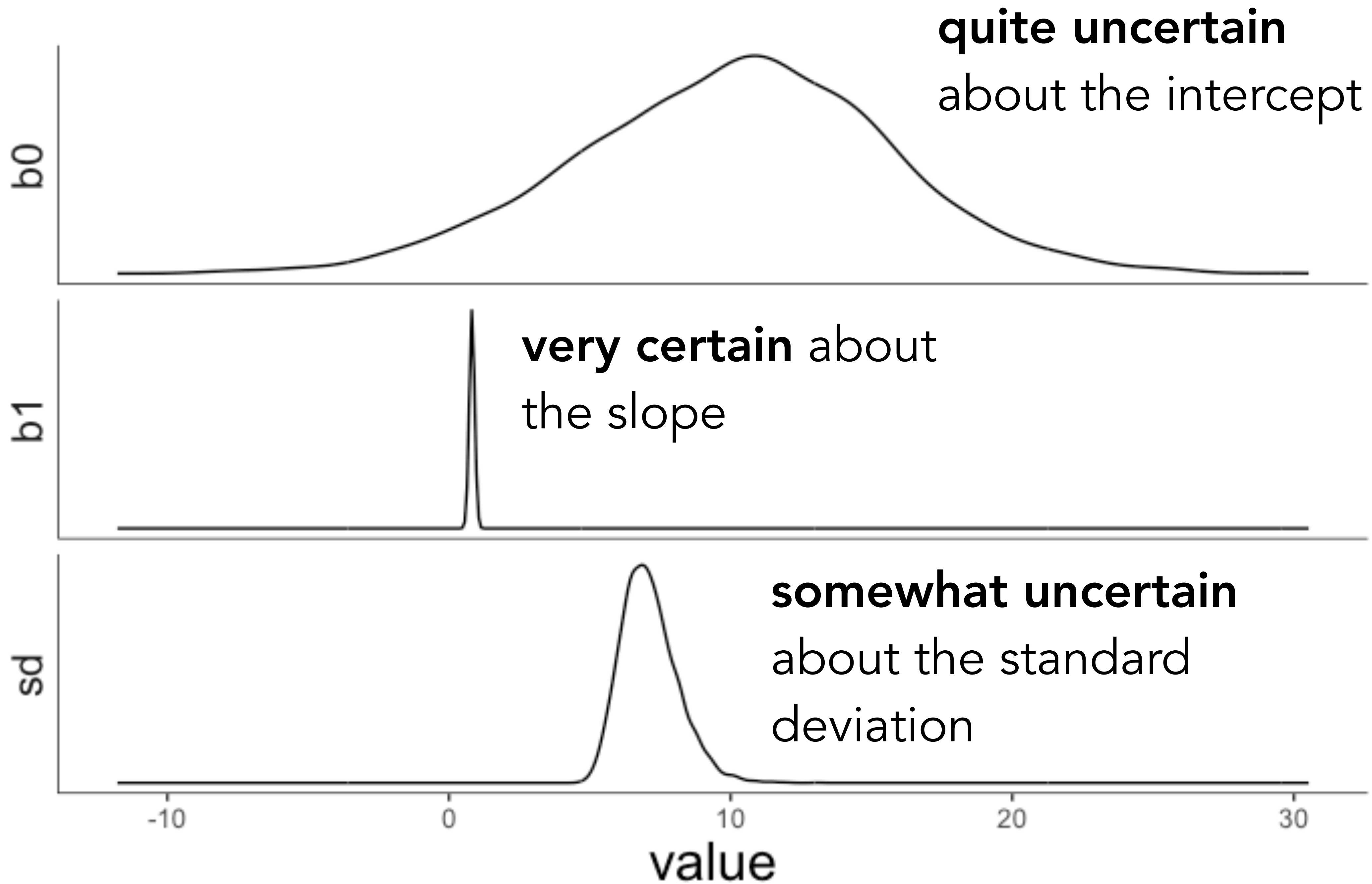
Inference via sampling

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

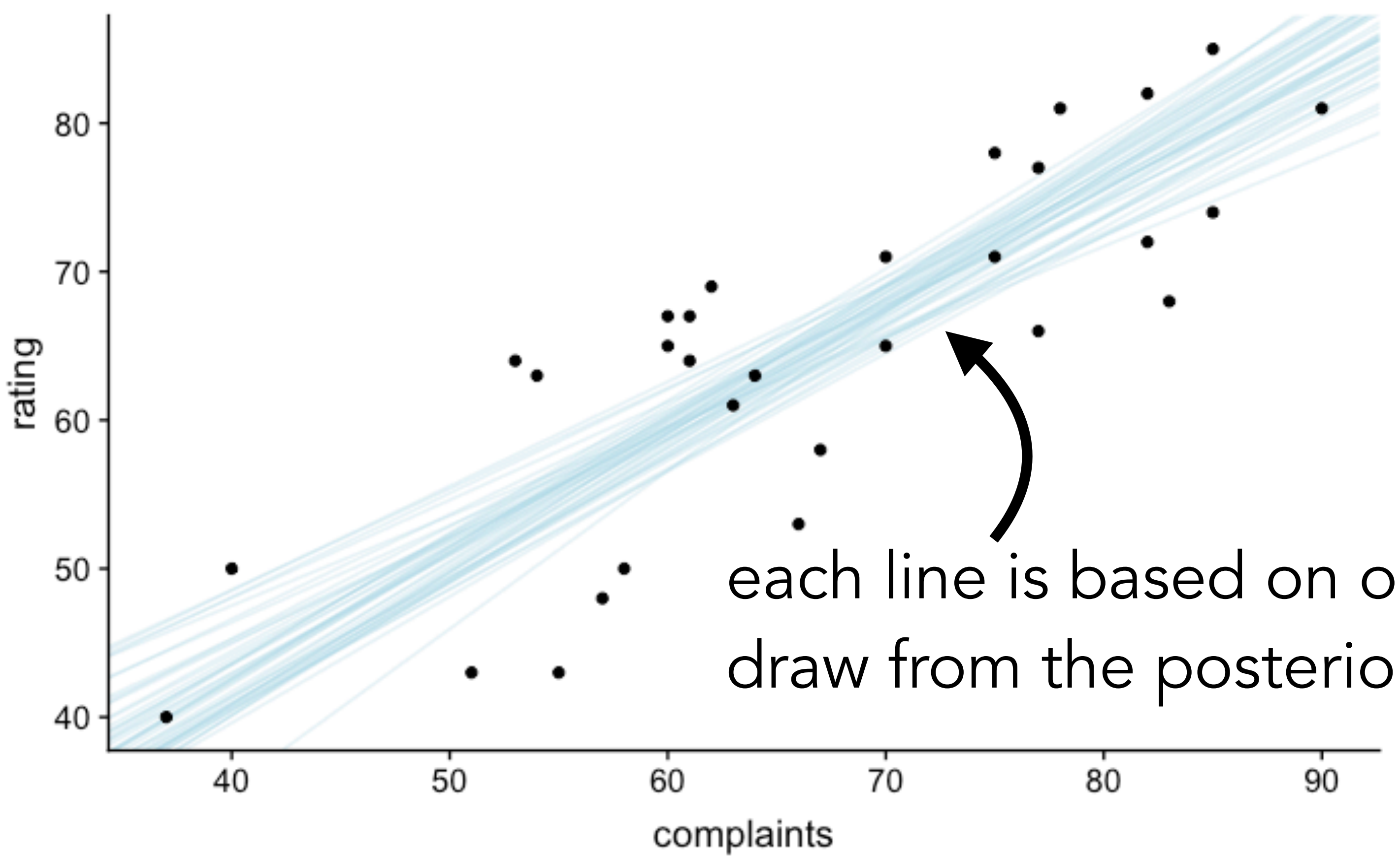
Visualize the posterior



Posterior distribution over the three parameters in the model

Visualize the model predictions

```
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()
```

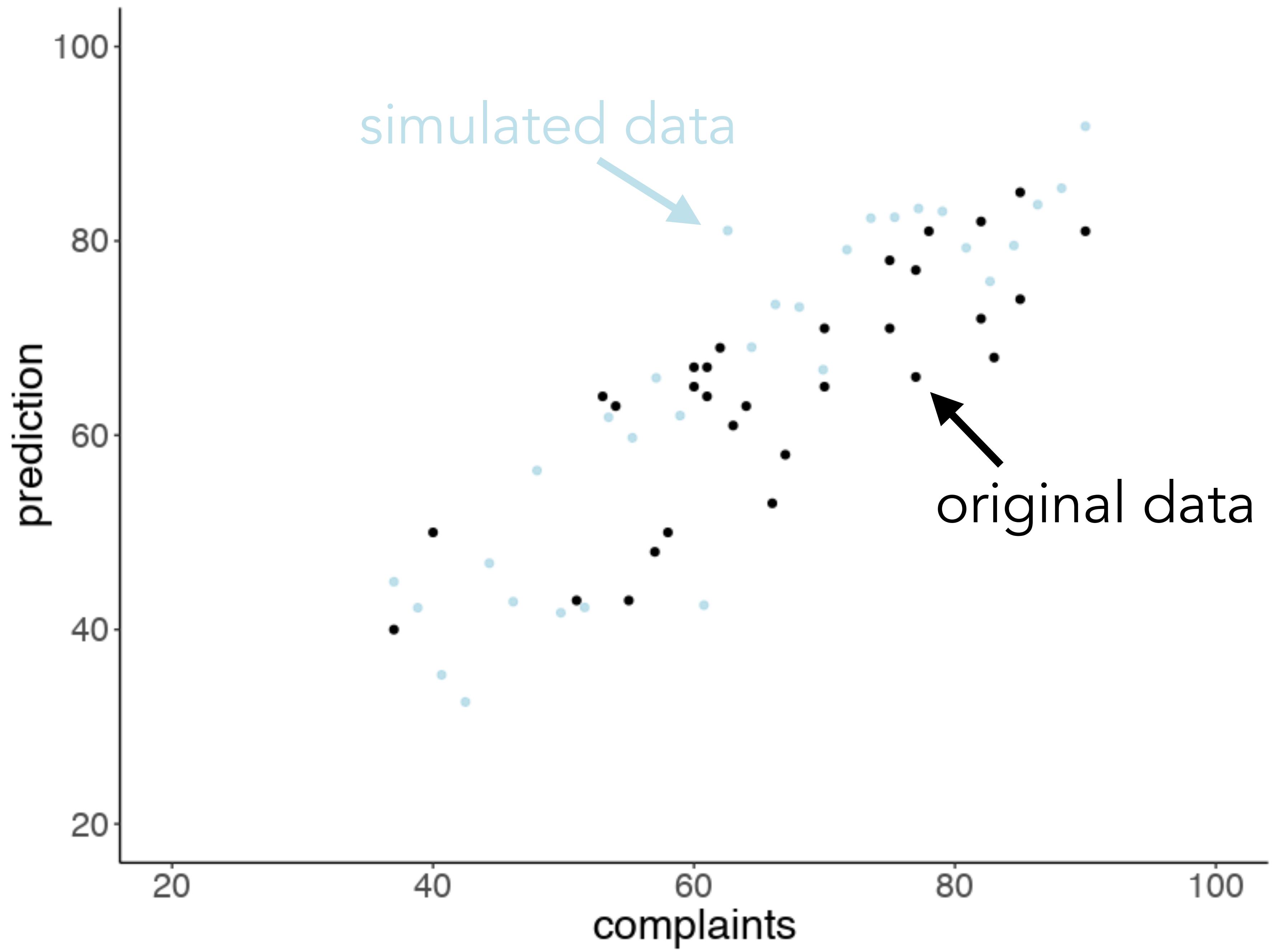


each line is based on one draw from the posterior

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

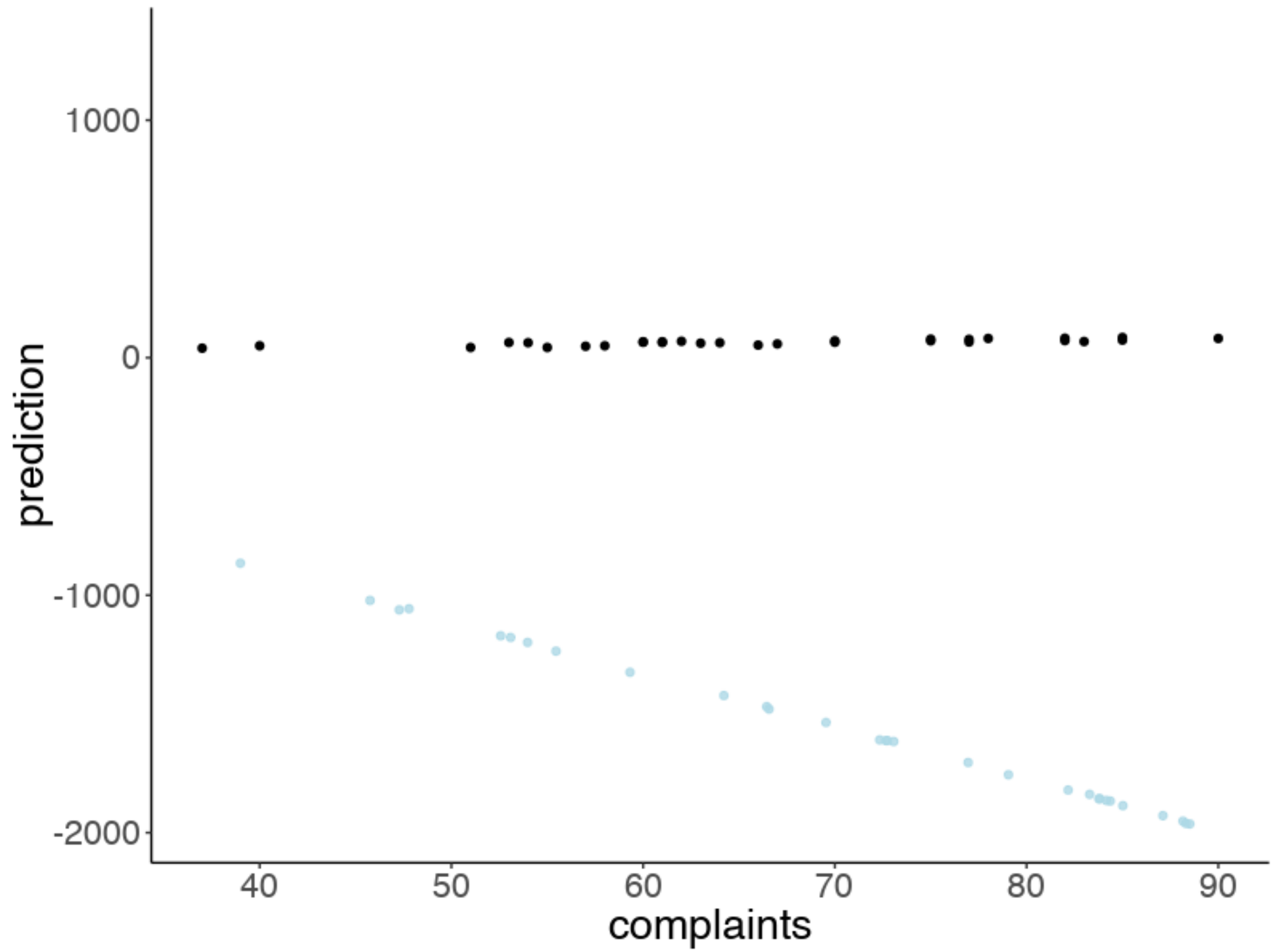
1. sample parameters from the posterior distribution
2. generate data using these parameters (using the likelihood function)



Hypothetical outcome plot

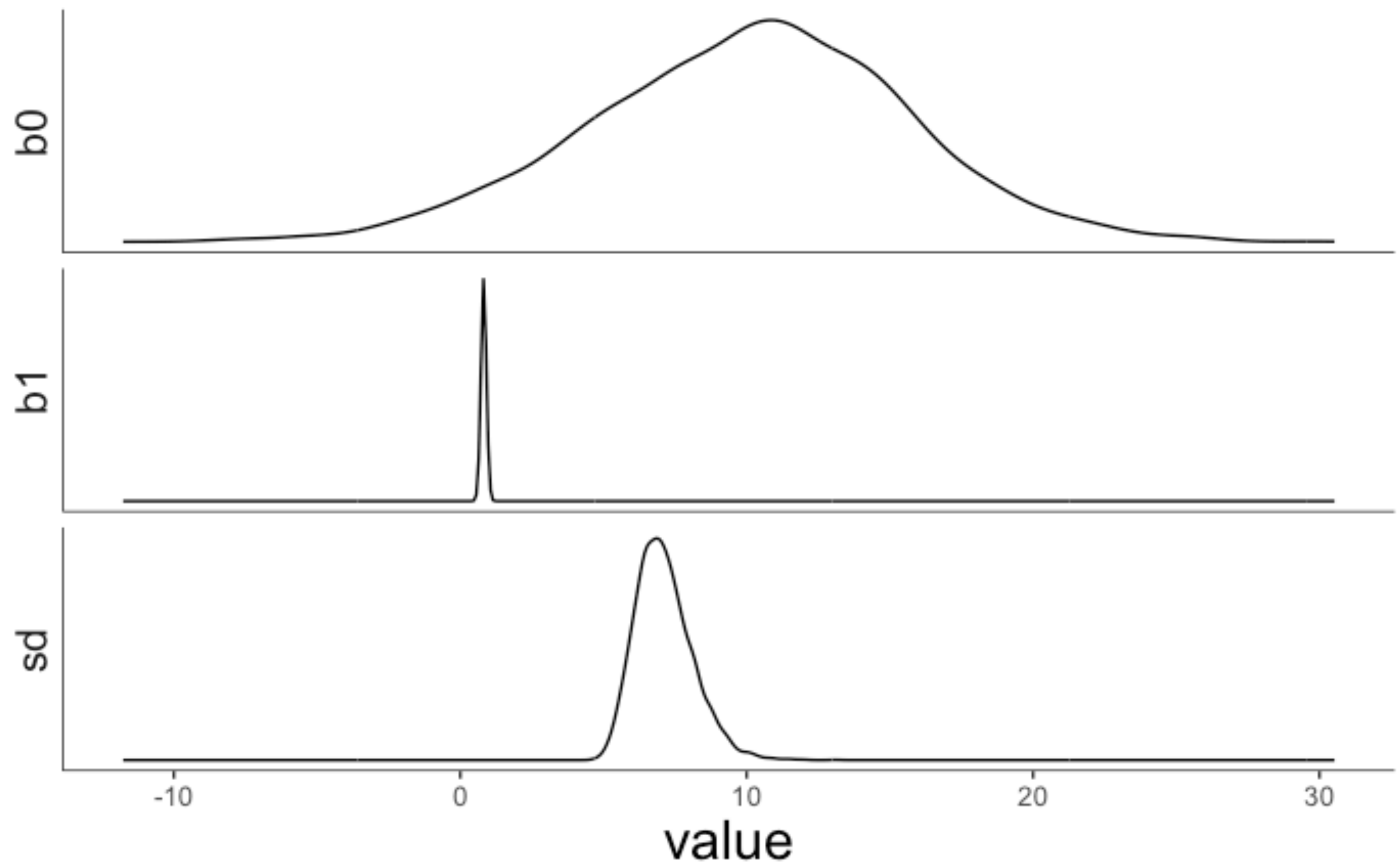
Prior predictive check

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



Hypothetical outcome plot

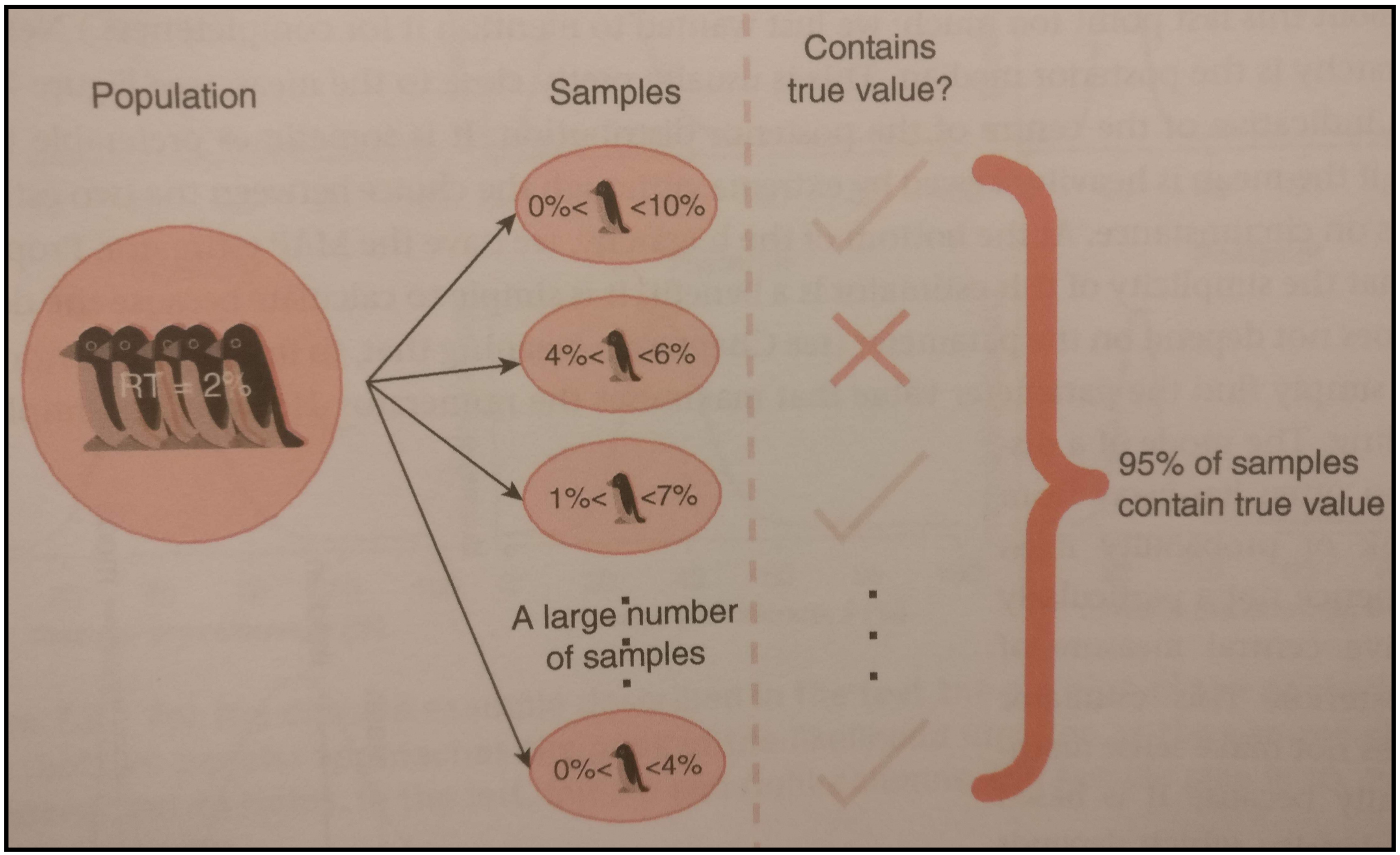
Summarizing results



- 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%.

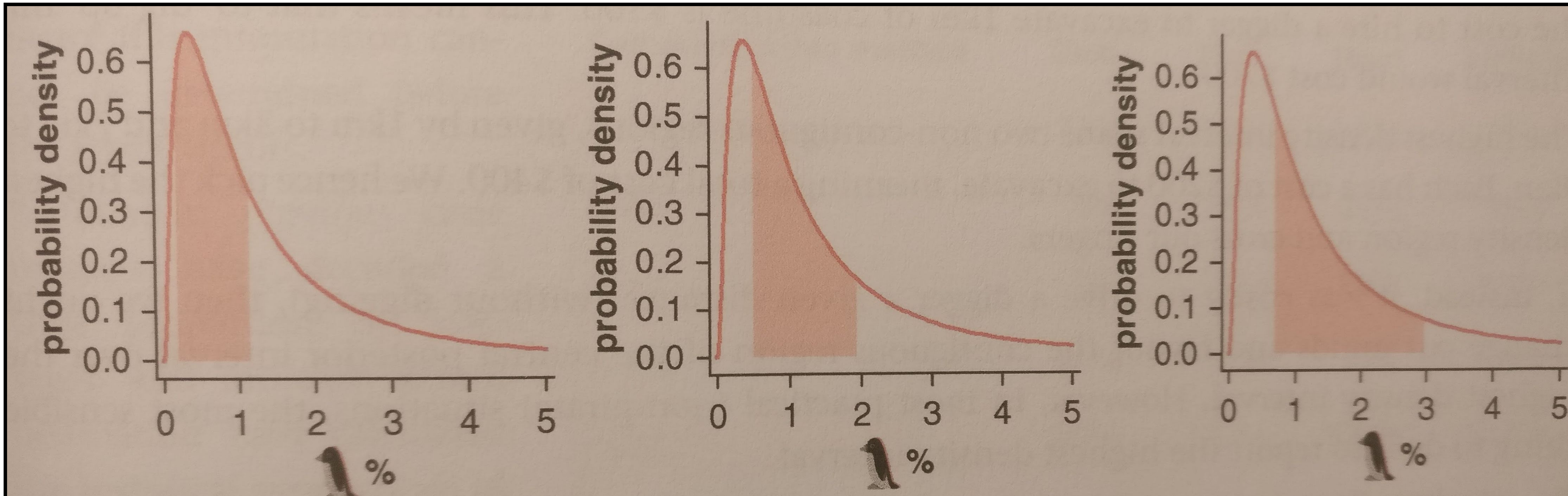
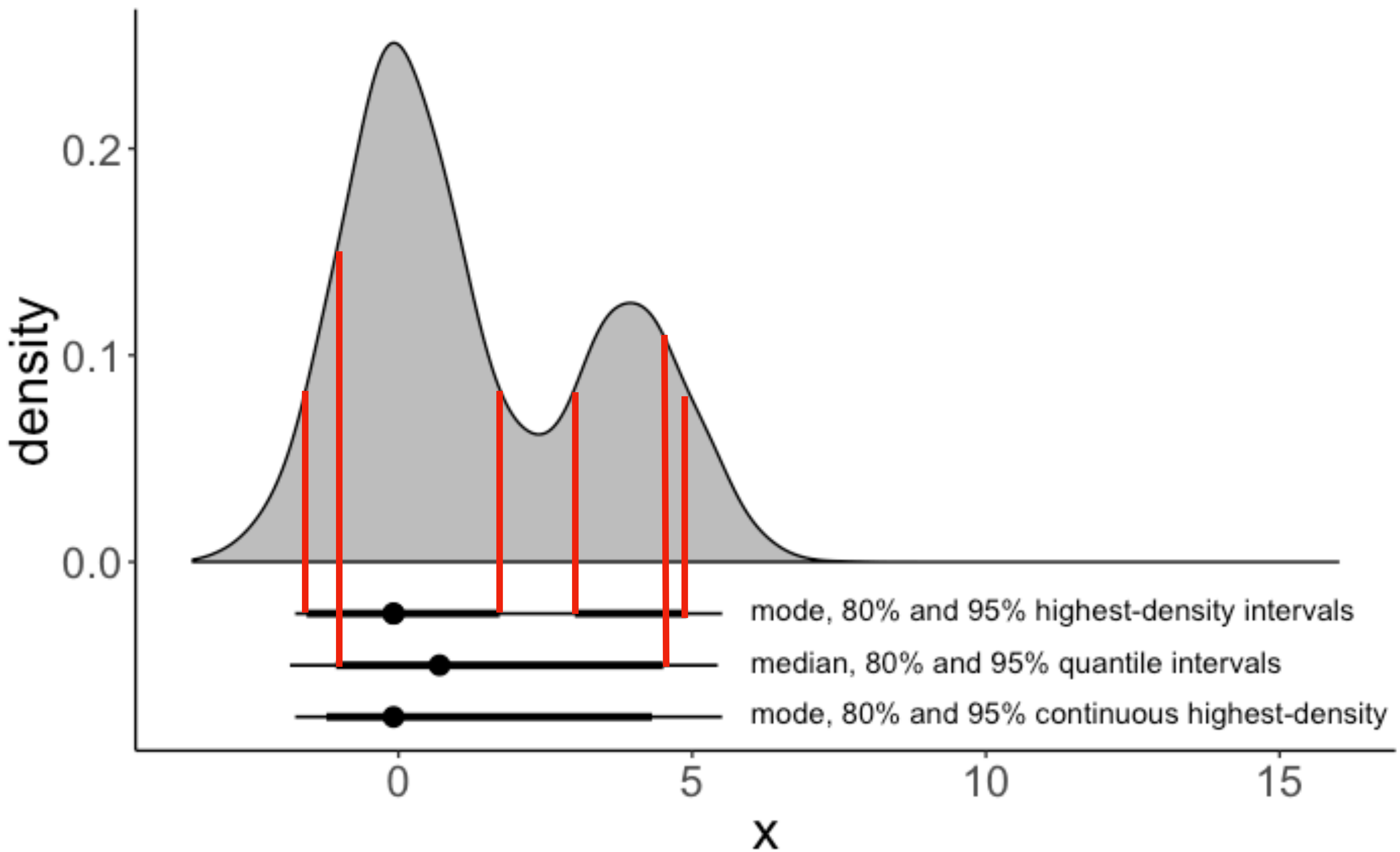


Figure 7.8 Three examples of 50% credible intervals for a parameter representing the proportion of penguins with red tails.

Different kinds of credible intervals

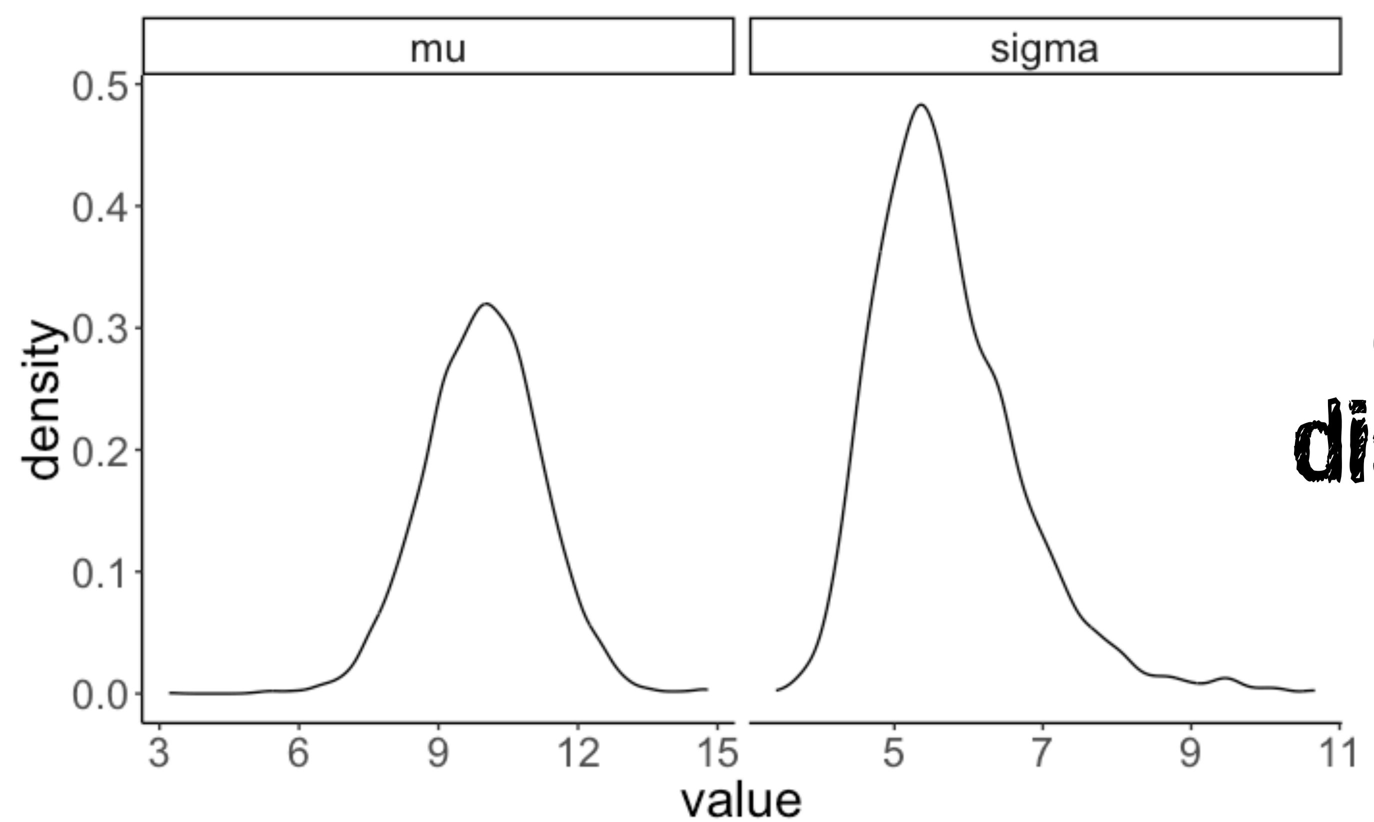


Sensitivity analysis

- 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)

Making comparisons

```
# prior
mu = normal(mean = 0, sd = 10)
sigma = cauchy(location = 0, scale = 3, truncation = c(0, Inf))

# data
data = rnorm(n = 20, mean = 10, sd = 5)

# likelihood
distribution(data) = normal(mu, sigma)

# fit model
m1 = model(mu, sigma)

# sample from the model
draws = mcmc(m1, n_samples = 1000)
```

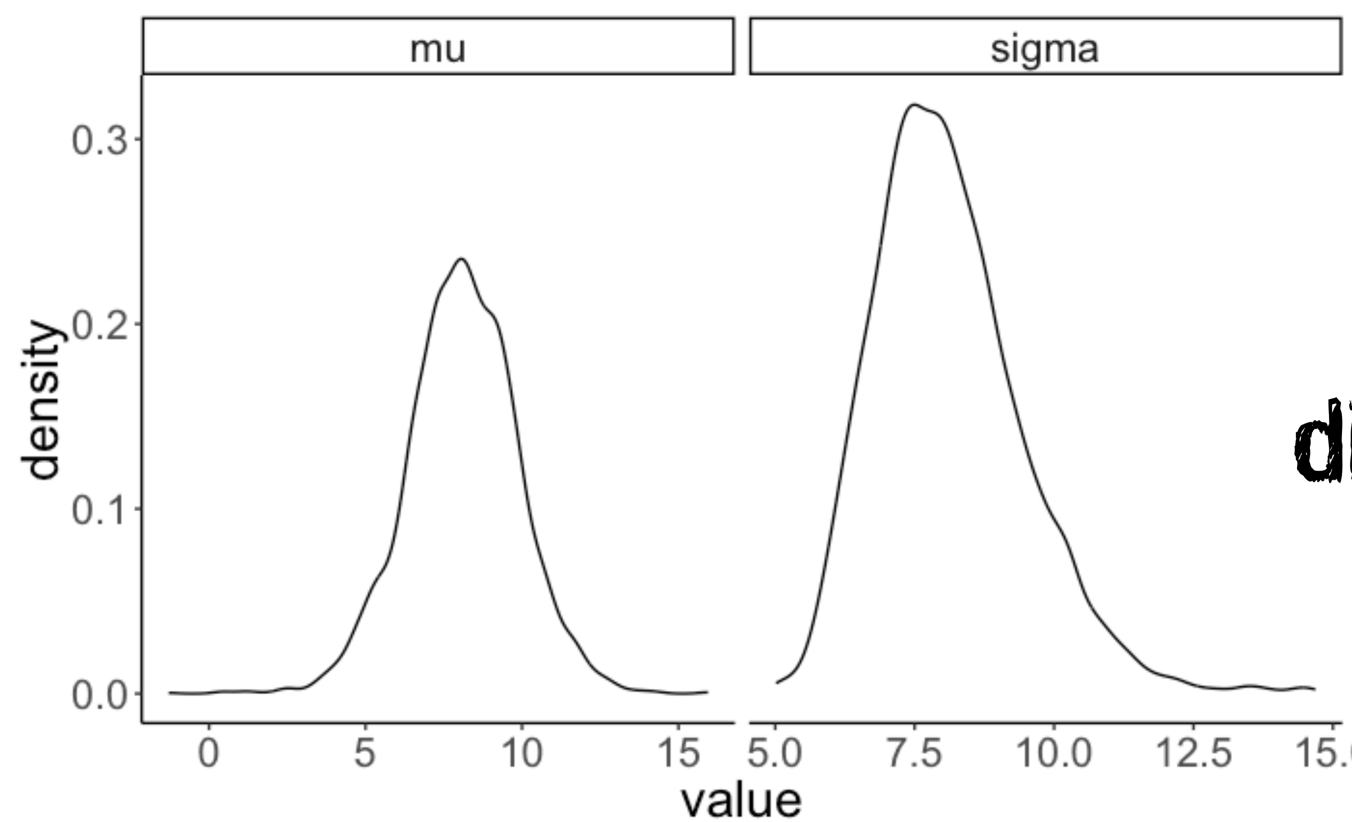


posterior distributions

Making comparisons

```
# prior
mu = normal(mean = 0, sd = 10)
sigma = cauchy(location = 0, scale = 3, truncation = c(0, Inf))

# data
data = rnorm(n = 20, mean = 6, sd = 8)

# likelihood
distribution(data) = normal(mu, sigma)

# fit model
m2 = model(mu, sigma)

# sample from the model
draws = mcmc(m2, n_samples = 1000)
```

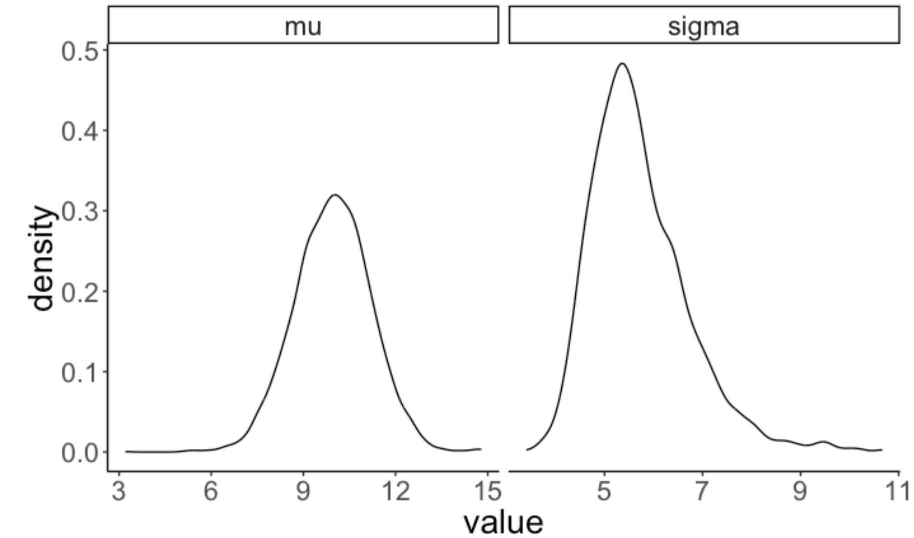
different data set

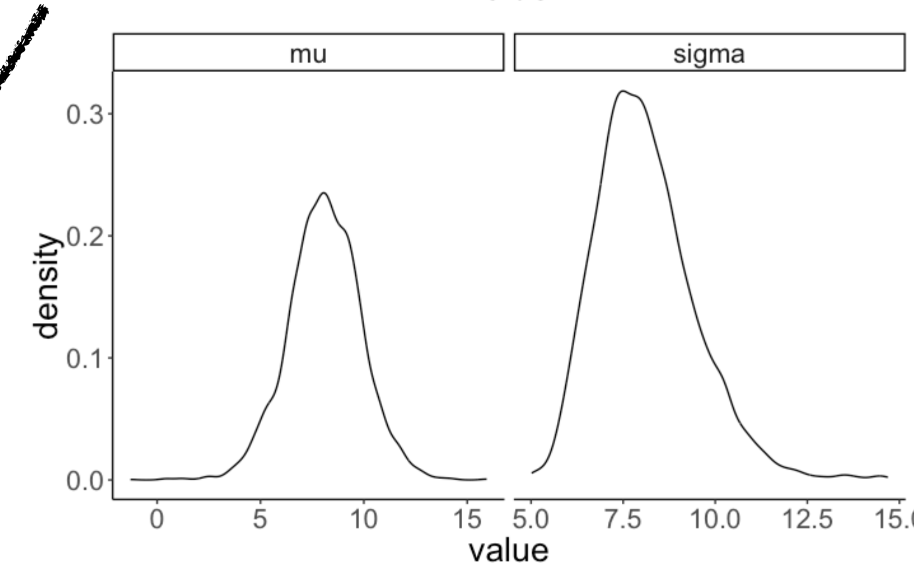


posterior distributions

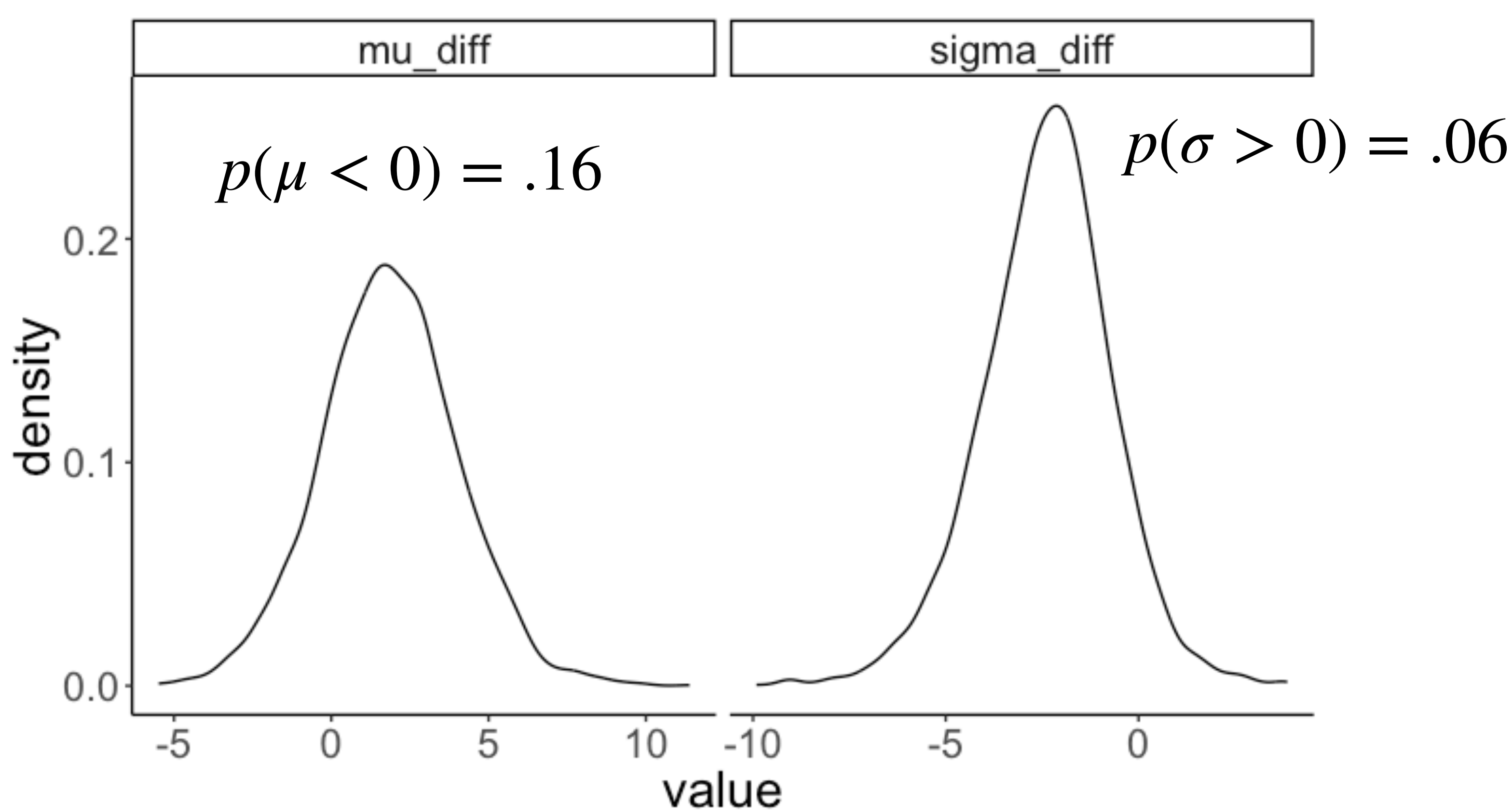
Making comparisons

draw	mu_1	sigma_1	mu_2	sigma_2	mu_diff	sigma_diff
1	8.28	7.64	5.59	8.13	2.69	-0.48
2	11.34	6.97	6.17	7.05	5.17	-0.08
3	10.93	6.63	6.40	6.92	4.52	-0.30
4	9.87	6.14	7.96	6.35	1.90	-0.20
5	9.90	6.38	9.86	9.44	0.04	-3.06
6	8.88	6.66	9.04	9.33	-0.16	-2.67
7	9.01	6.12	6.54	8.72	2.47	-2.60
8	8.75	5.93	10.55	8.19	-1.81	-2.26
9	9.69	5.62	4.99	8.56	4.71	-2.94
10	11.39	6.76	9.36	8.23	2.03	-1.47





posterior of the difference in mean and variance



Summary

- Flipping coins: Simple Bayesian inference
- Bayes' rule
  - Weighting of prior knowledge and evidence
  - Ingredients:
    - likelihood
    - prior
    - inference
- Doing Bayesian data analysis
  - A simple linear regression
  - Making comparisons

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!