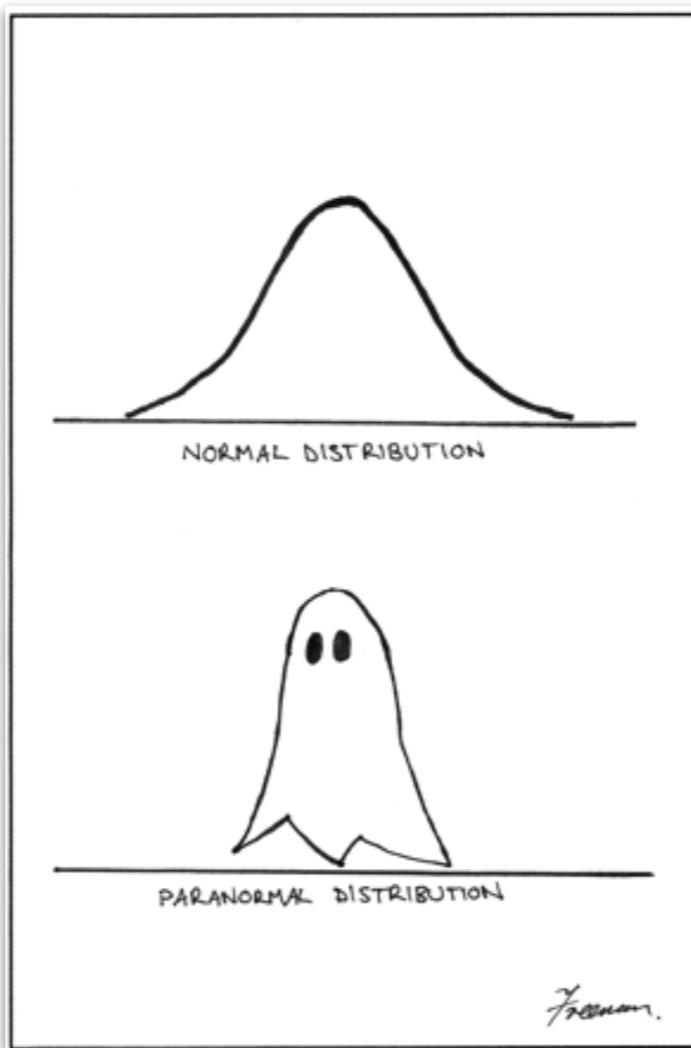


# Simulation 1



A screenshot of a Spotify interface showing a collaborative playlist titled "psych252". The interface includes a play button, a "PLAY" button, and a three-dot menu icon. The URL <https://tinyurl.com/psych252spotify24> is displayed below the title.

# **Logistics**

# Homework 3



## 1 Distributions (1 point)

When we have empirical data, we can compute cumulative probabilities and create probability density functions using `quantile()` and `density()`, respectively. Take a look at the help files for both of these functions to better understand what they're doing.

Consider the following data set:

```
df.p1 = tibble(observation = 1:20,
               rating = c(0.3775909, 0.5908214, 0.07285336, 0.06989763, 0.2180343,
                         1.447484, 0.614781, 0.2698414, 0.4782837, 0.073523,
                         0.6953676, 0.3810149, 0.6188018, 2.211967, 0.5272716,
                         0.517622, 0.9380176, 0.3273733, 0.1684667, 0.2942399))
```

### 1.1 Quantile (0.5 points)

What's the 60% percentile of the `rating` variable? (60% of the values are lower than that value?)

```
### YOUR CODE HERE ###
```

```
#####
```

### 1.2 Density (0.5 points)

Plot the density of the `rating` variable. Use

```
### YOUR CODE HERE ###
```

```
#####
```

## 2 Sampling distribution (7 points)

The sample standard deviation  $s = \sqrt{\frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{n-1}}$  is an unbiased estimator of the population standard deviation. In this exercise, we will run a simulation to compare  $s$  with another estimator  $s' = \sqrt{\frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{n}}$  and show that  $s'$  is biased. Note that the only difference between  $s$  and  $s'$  is in the denominator.

1

## 3 Permutation test (7 points)

Imagine that you collected data about people's heights from three different places and you are interested whether there are any differences in people's height between the three places.

By visualizing the data, we can see that the variances between the three groups differ considerably, which is troublesome for parametric tests (e.g. a t-test). However, we can perform a permutation test, which is non-parametric. In this case, we are interested in whether the maximal difference between each of the pairs of group means, is greater than we would expect to see by chance.

```
set.seed(1)

df.heights = read_csv("data/df_heights.csv")

df.heights %>%
  ggplot(data = .,
         mapping = aes(x = group,
                       y = height)) +
  geom_point(position = position_jitter(height = 0,
                                         width = 0.1),
             alpha = 0.5) +
  stat_summary(fun.data = "mean_cl_boot",
              shape = 21,
              fill = "lightblue",
              size = 1)
```

# Homework 3

remember to set eval = T  
when knitting the file

```
```{r p2.1, eval = F}
set.seed(1)

n_simulations = 10000 # number of simulations
n_samples = 40 # number of samples in each simulation
population_mean = 0 # ground truth mean
population_sd = 1 # ground truth standard deviation

### YOUR CODE HERE ###
df.samples =
#####
# df.samples %>%
#   head(5)

df.samples %>%
  summary()
```
```

# Your feedback

# Your feedback

If there was recordings of this classes (from the previous years), it will be great. Gives us a chance to relook at the content.

I really liked the examples and when we were solving the problems! I think it would be nice to walk through the little screenshots of R code a little slower if possible!

Nice-- Even though we had some tech issues we made it through without feeling too rushed!

I think it was out of our control bc tech issues but the speed was a bit high

# Things that came up

# PDFs for homework

PDF style #17



Anonymous

17 hours ago in Homework - HW2



9

PIN

STAR

WATCH

VIEWS



What should the PDF that we submit look like? Should we get rid of most of the instructions so it's a cleaner presentation of our code and figure, or does that not matter? Also, given that there are clear instructions for some of the data wrangling steps, should we duplicate that info in our code comments? Or should comments only focus on more specific implementation details that might otherwise be opaque?

Thanks!

[Comment](#) [Edit](#) [Delete](#) [Endorse](#) [...](#)

1 Answer



Tobi Gerstenberg STAFF

19 seconds ago



Thanks for checking!



I'd suggest to include everything in your pdf for homework 2. This makes it easier for TAs to evaluate your work. As for the data wrangling instructions, you can either keep them or remove them (both works).

[Comment](#) [Edit](#) [Delete](#) [Endorse](#) [...](#)

Add comment

You have already written an answer to this question. Write another?

[Write another answer](#)

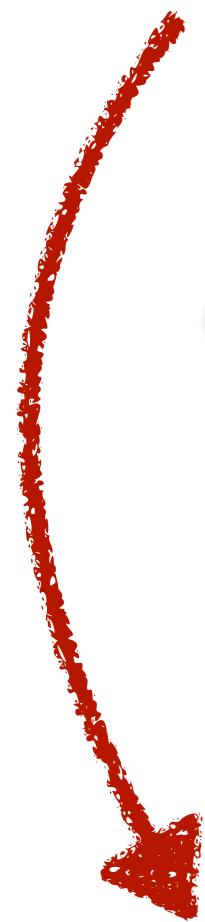
# My apologies

I'm done.

Please help.

Please help.

I'm done.



# Plan for today

- Quick recap
- Simulating data
  - Drawing samples
  - Working with probability distributions
  - Quick detour: understanding density()
  - Asking probability distributions for answers
- Doing Bayesian inference
  - Analytic solution
  - Sampling solution

# Quick recap

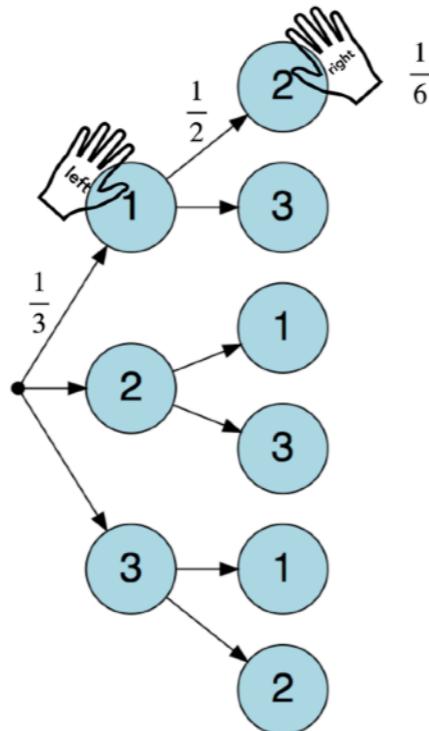
# Quick recap

## Sampling **without** replacement

$$p(\text{left} = 1, \text{right} = 2) = ?$$

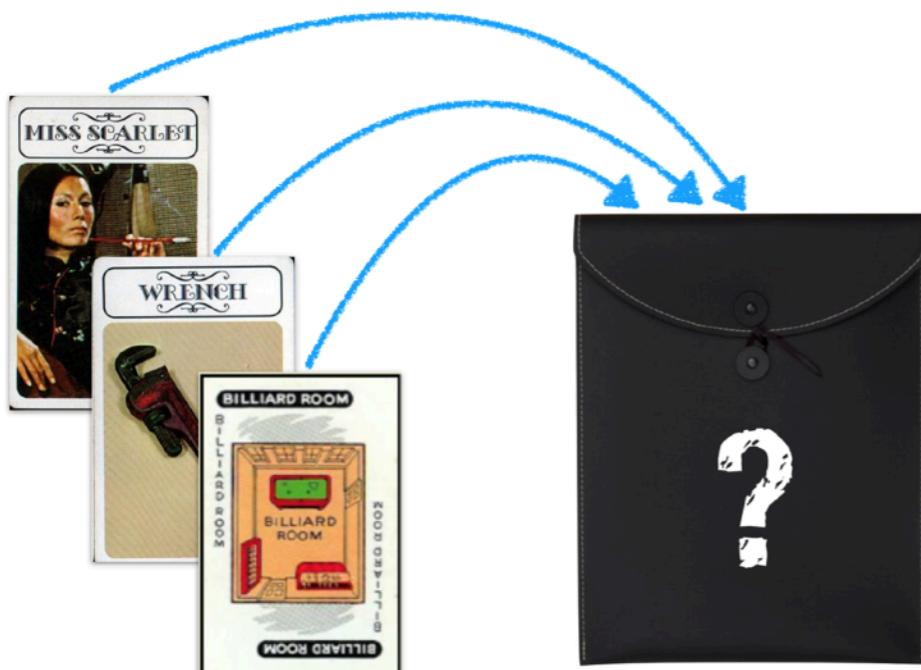


What is the probability that I first draw the 1 with my left hand, and then, without putting the 1 back into the urn, draw the 2 with my right hand?



15

## Clue guide to probability



28

## Definitions

If  $P(X_i)$  is the probability of event  $X_i$

1. Probability cannot be negative.

$$P(X_i) \geq 0$$



2. Total probability of all outcomes in the sample space is 1.

$$\sum_{i=1}^N P(X_i) = P(X_1) + P(X_2) + \dots + P(X_N) = 1$$

23

## Clue guide to probability

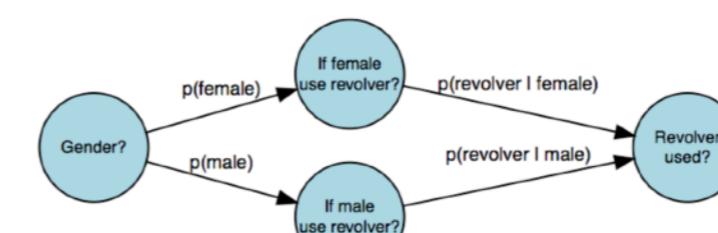
**Who?**



**What?**



$$p(\text{what} = \text{Revolver}) = ?$$



12

39

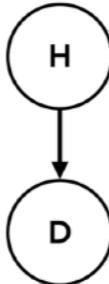
# Quick recap

## Clue guide to probability

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

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

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

44

## Clue guide to probability

### what we know

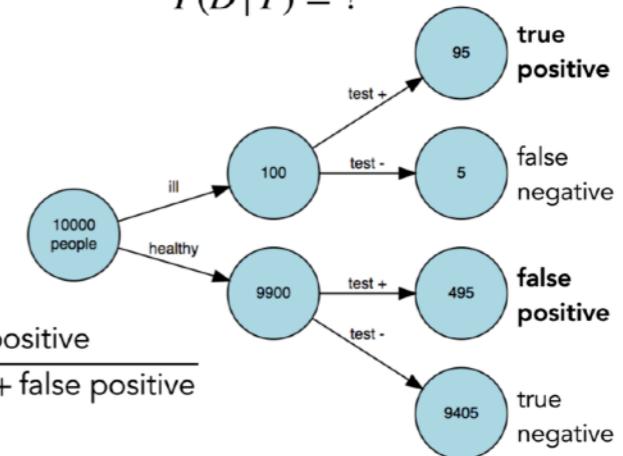
$$P(D) = 0.01$$

$$P(T|D) = 0.95$$

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

### what we want to know

$$P(D|T) = ?$$



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

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

$$\approx 0.16$$

51

## Getting Bayes right matters!



### Original claim:

Requires Bayes' rule

$$\Pr(\text{shot}|\text{minority civilian, white officer}, X) - \Pr(\text{shot}|\text{minority civilian, minority officer}, X) \\ = \frac{\Pr(\text{min. civ.}|\text{shot, white off., } X) \times \Pr(\text{shot}|\text{white off., } X)}{\Pr(\text{minority civilian}|\text{white officer}, X)} \\ - \frac{\Pr(\text{min. civ.}|\text{shot, min. off., } X) \times \Pr(\text{shot}|\text{min. off., } X)}{\Pr(\text{minority civilian}|\text{minority officer}, X)} \quad [2]$$

### Claim:

"White officers are not more likely to shoot minority civilians than non-White officers"

$$\Pr(\text{shot}|\text{minority civilian, white officer}, X) - \Pr(\text{shot}|\text{minority civilian, minority officer}, X) \leq 0, \quad [1]$$

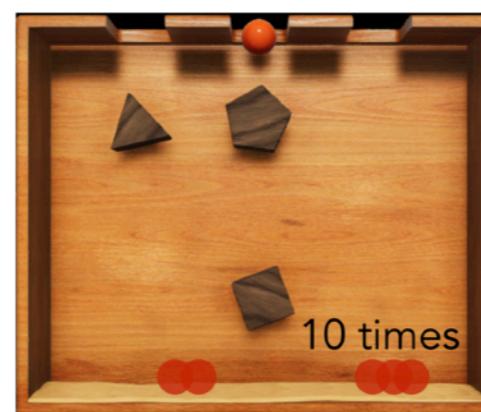
### What the statistic says:

"whether a person fatally shot was more likely to be Black (or Hispanic) than White"

authors didn't have the relevant data to support their claim!

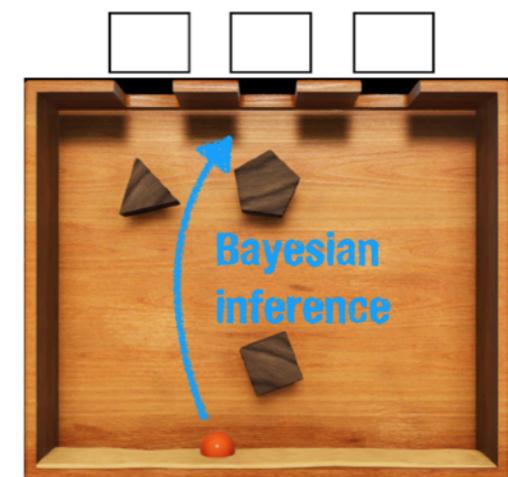
*paper was retracted*

### Prediction



Where will the ball land?

### Inference



In which hole was the ball dropped?

# Simulating data

# Drawing samples

# Simulating data: Why?



- helps us to:
  - better understand statistical concepts (e.g. p-values, confidence intervals)
  - check how accurately our statistical model can infer the ground truth
  - do power analysis
  - get one step closer to being able to develop our own probabilistic models of an interesting phenomenon

# Simulating data: How?



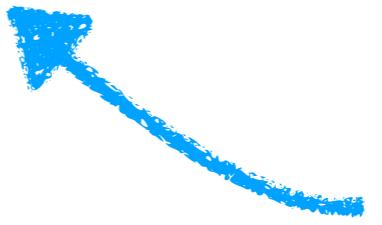
line numbers

```
1 numbers = 1:3
2
3 numbers %>%
4   sample(size = 10,
5         replace = T)
[1] 3 3 1 2 2 3 2 3 1 2
```

sample 10 times  
with replacement please  
thank you

# Simulating data: How?

```
1 numbers = 1:3  
2  
3 numbers %>%  
4   sample(size = 10,  
5           replace = T,  
6           prob = c(0.8, 0.1, 0.1))  
  
[1] 3 1 1 1 1 2 2 1 1 1
```



I want mostly 1s

# Simulating data: How?

sets the seed of the random number generator

```
1  `` {r no-seed}
2 numbers = 1:5
3
4 numbers %>%
5   sample(5)
6 ````
```

```
[1] 1 4 5 3 2
[1] 5 3 4 2 1
```

1 `` {r with-seed}
2 set.seed(1)
3
4 numbers = 1:5
5
6 numbers %>%
7 sample(5)
8 ````

```
[1] 1 4 3 5 2
[1] 1 4 3 5 2
```

every time I run this code chunk, I may get a different outcome

every time I run this code chunk, I get the same outcome 19

# set the seed for reproducible code!

sets the seed of the random number generator

```
1 `-- {r no-seed}
2 numbers = 1:5
3
4 numbers %>%
5   sample(5)
6`--
```

[1] 1 4 5 3 2  
[1] 5 3 4 2 1

1`-- {r with-seed}
2 set.seed(1)
3
4 numbers = 1:5
5
6 numbers %>%
7 sample(5)
8`--

every time I run this code chunk, I may get a different outcome

every time I run this code chunk, I get the same outcome 20

# Simulating data: How?

## Sampling rows from a data frame

```
1 set.seed(1)
2 n = 10
3 df.data = tibble(trial = 1:n,
4                   stimulus = sample(c("flower", "pet"), size = n, replace = T),
5                   rating = sample(1:10, size = n, replace = T))
```

| trial | stimulus | rating |
|-------|----------|--------|
| 1     | flower   | 3      |
| 2     | pet      | 1      |
| 3     | flower   | 5      |
| 4     | flower   | 5      |
| 5     | pet      | 10     |
| 6     | flower   | 6      |
| 7     | flower   | 10     |
| 8     | flower   | 7      |
| 9     | pet      | 9      |
| 10    | pet      | 5      |

sample 6 rows with replacement

```
1 df.data %>%
2   slice_sample(n = 6,
3                 replace = T)
```

| trial | stimulus | rating |
|-------|----------|--------|
| 9     | pet      | 9      |
| 4     | flower   | 5      |
| 7     | flower   | 10     |
| 1     | flower   | 3      |
| 2     | pet      | 1      |
| 7     | flower   | 10     |

sample 50% of the rows

```
1 df.data %>%
2   slice_sample(prop = 0.5)
```

| trial | stimulus | rating |
|-------|----------|--------|
| 9     | pet      | 9      |
| 4     | flower   | 5      |
| 7     | flower   | 10     |
| 1     | flower   | 3      |
| 2     | pet      | 1      |

# **Working with probability distributions**

# MOST POPULAR MARVEL MOVIE ACTORS

## HEIGHT COMPARISION

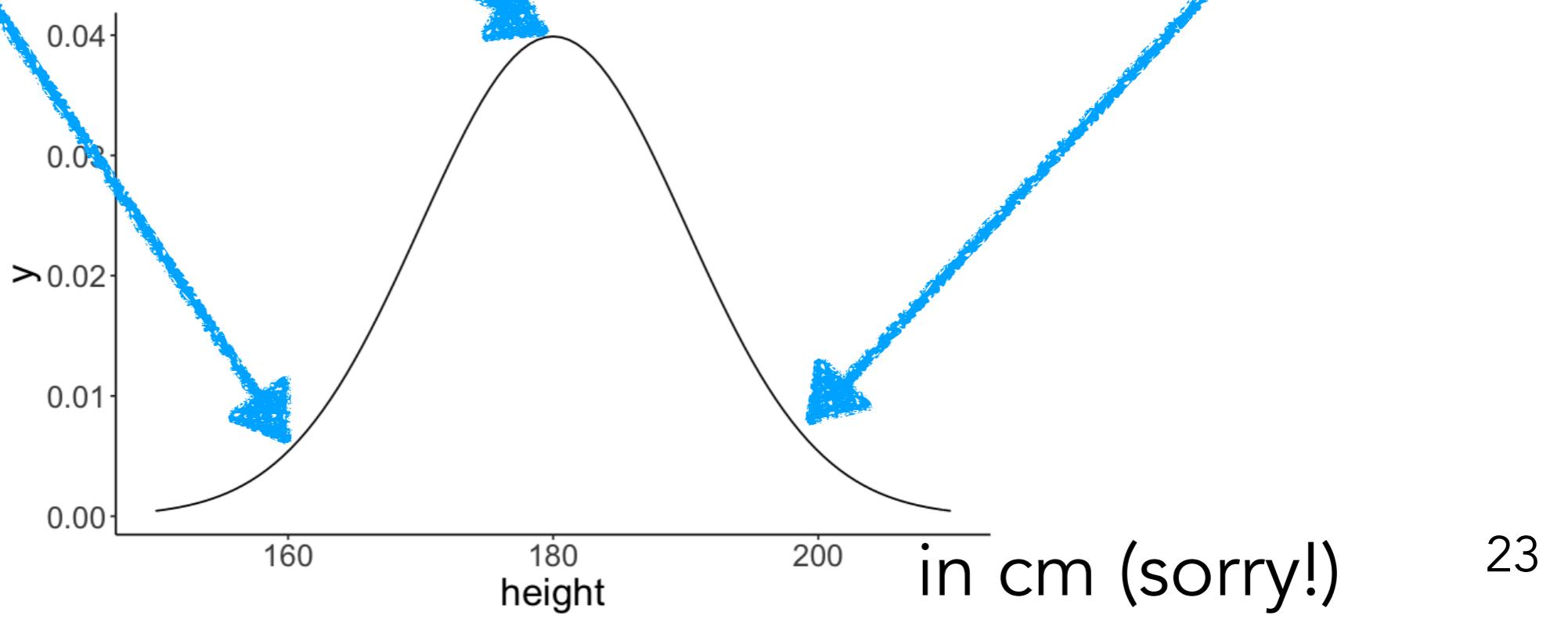
Who's the tallest and shortest actor in the Marvel Cinematic Universe?



Copyright 2018 Maurice Mitchell

[TheGeekTwins.com](http://TheGeekTwins.com)

Not Affiliated With Marvel Studios. All Rights Reserved



# Working with probability distributions

| letter | description   | example               |
|--------|---|-----------------------|
| d      | for “density”, the density function (probability mass function (for <i>discrete</i> variables) or probability density function (for <i>continuous</i> variables)) | <code>dnorm( )</code> |
| p      | for “probability”, the cumulative distribution function   | <code>pnorm( )</code> |
| q      | for “quantile”, the inverse cumulative distribution function  | <code>qnorm( )</code> |
| r      | for “random”, a random variable having the specified distribution   | <code>rnorm( )</code> |

# Normal distribution

make data frame with minimum and maximum x-value

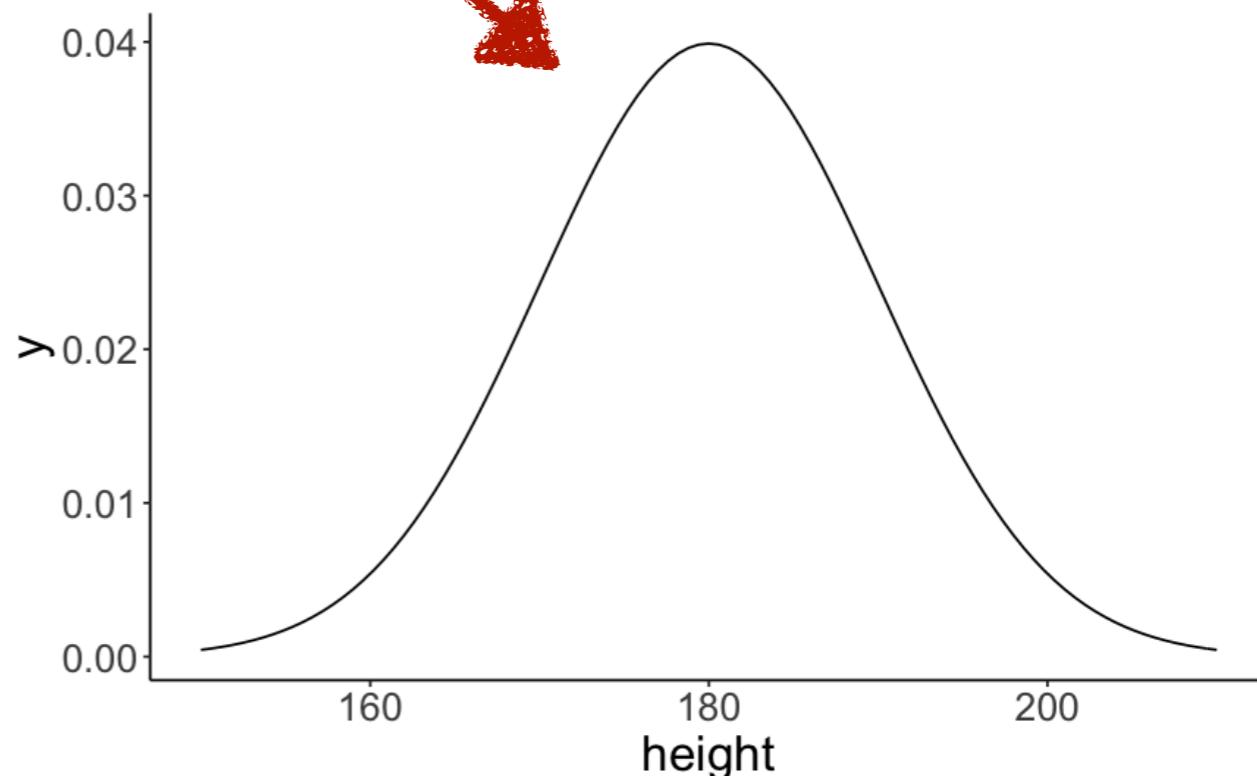
```
1 ggplot(data = tibble(height = c(150, 210)),  
2         mapping = aes(x = height)) +  
3     stat_function(fun = ~ dnorm(x = .,  
4                                     mean = 180,  
5                                     sd = 10))
```

function for plotting  
functions

what function  
should be plotted?

any parameters for  
the function?

the result



**dnorm**(x, mean = 180, sd = 10)

d = density

norm = normal distribution

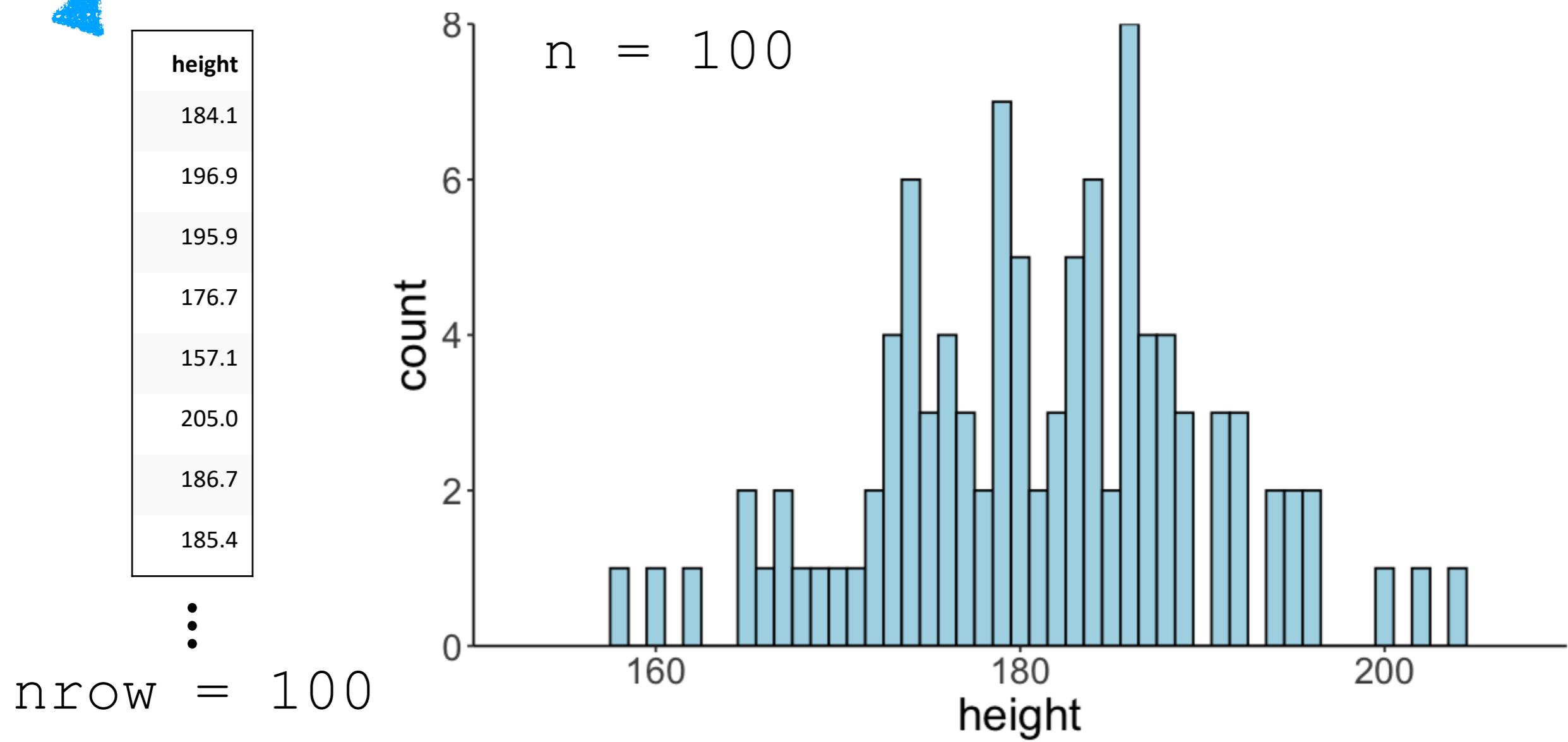
# Sampling from distributions

**rnorm**(n, mean = 180, sd = 10)

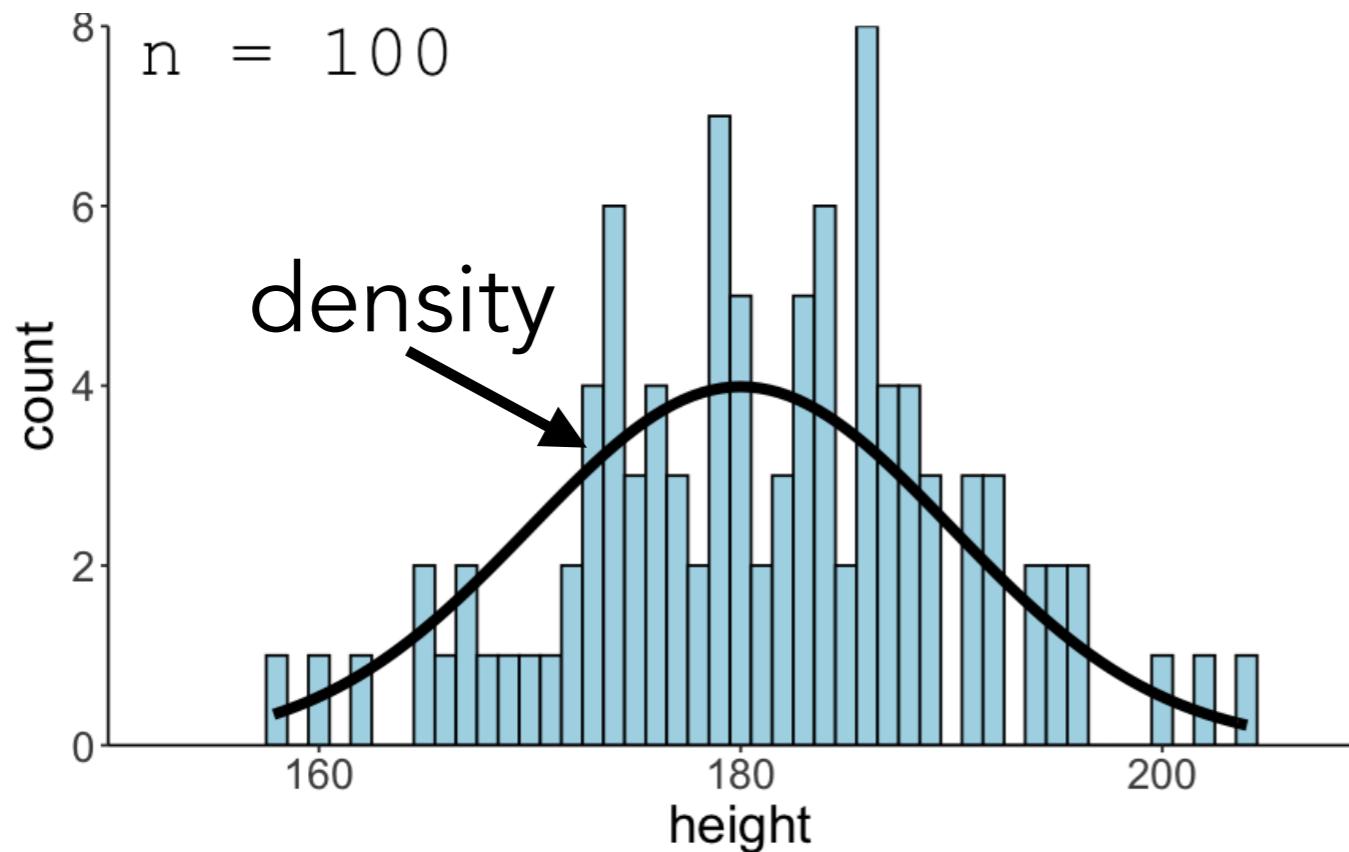
n = number of samples

r = random samples

norm = normal distribution

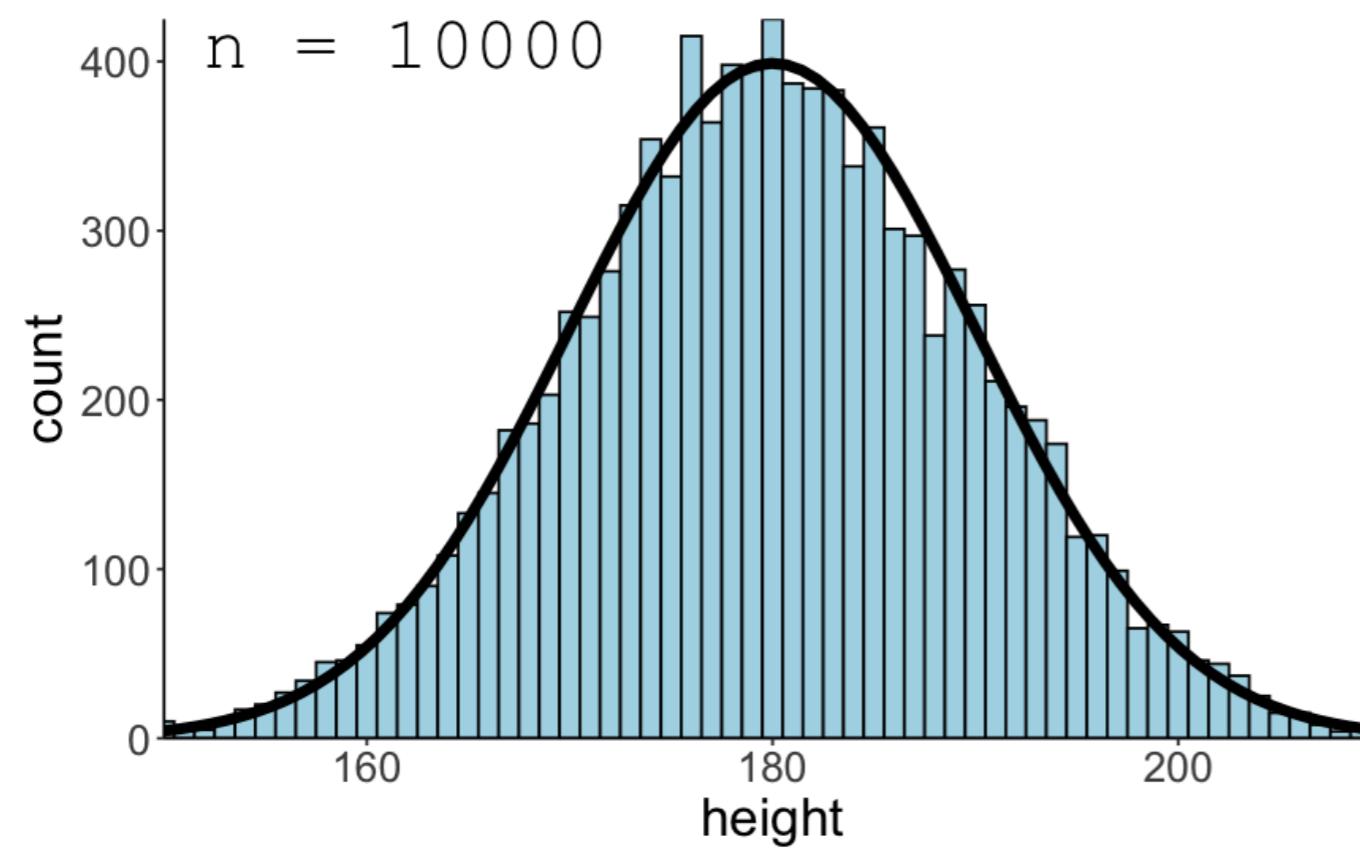


# Sampling from distributions

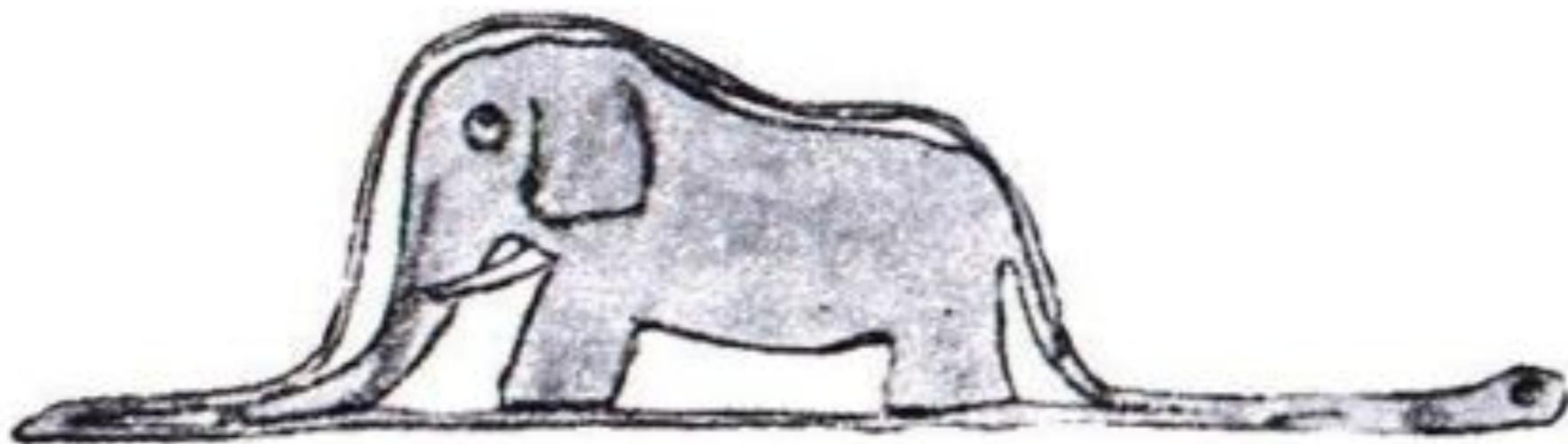
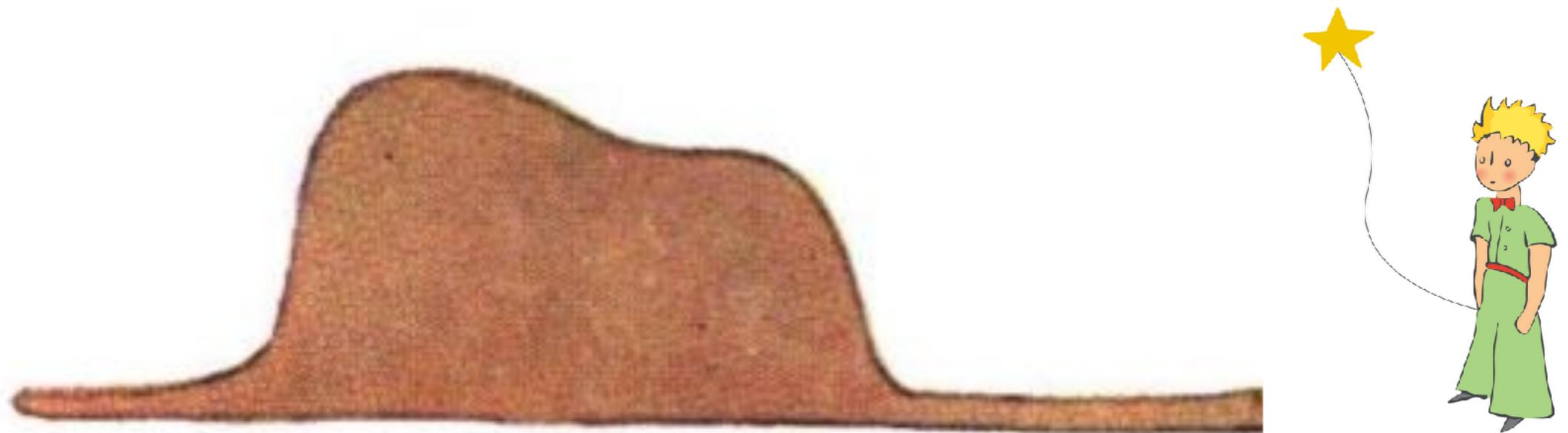


## law of large numbers

approximation to true underlying distribution improves with increased sample size



# Quick detour: understanding `density()`

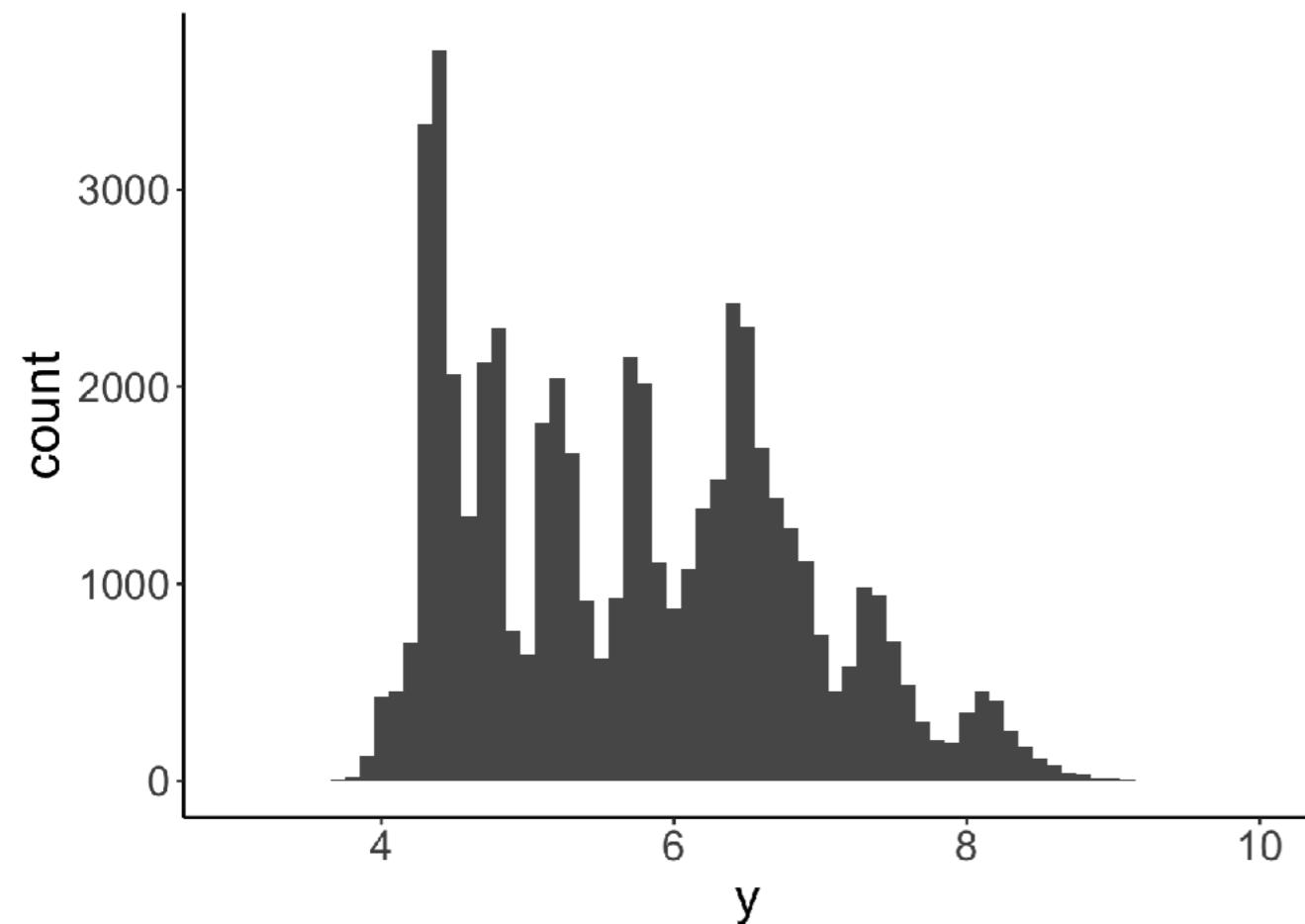


What's underneath the hood (or hat)?

# You've seen `density()` before

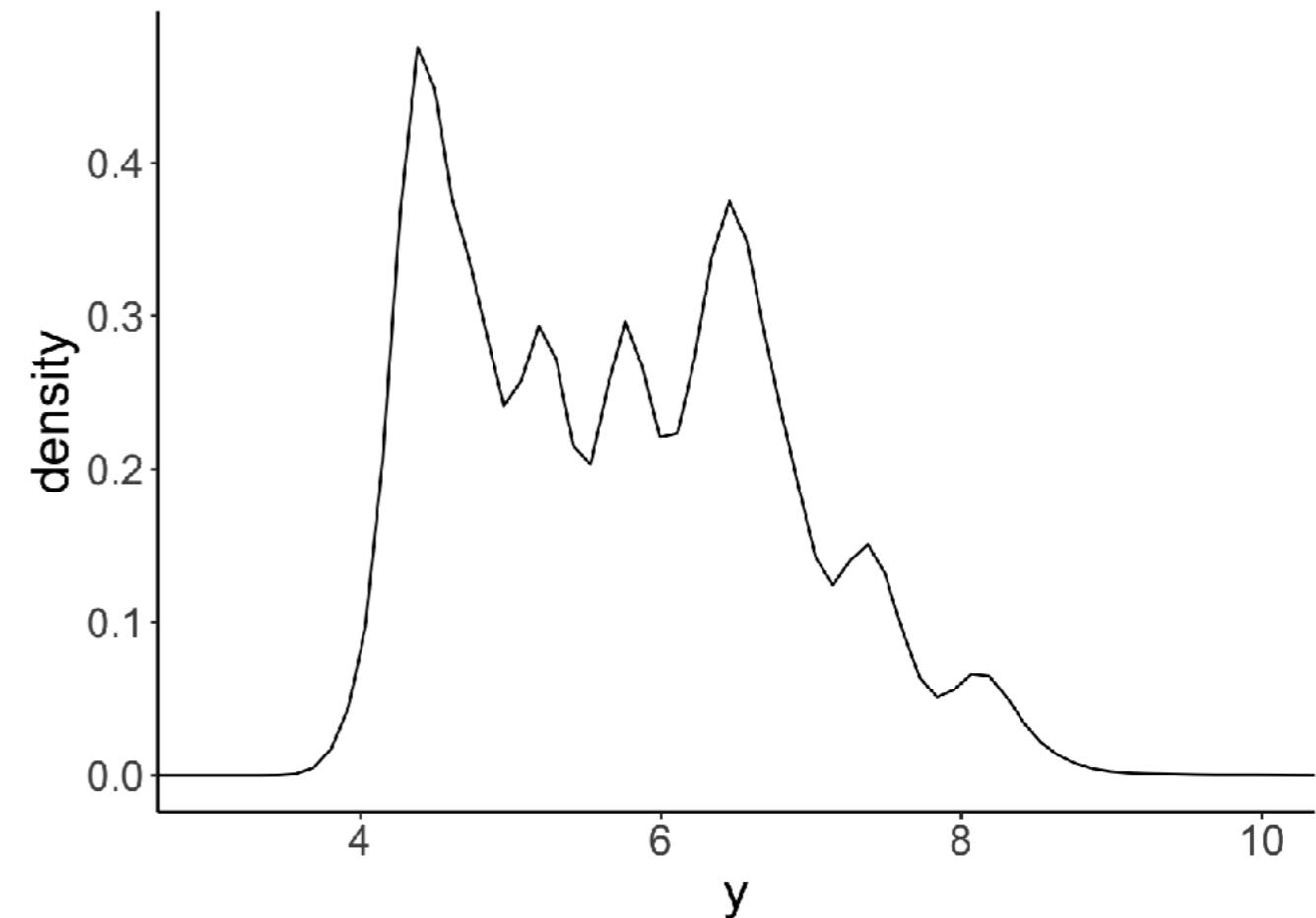
## Histogram

```
1 ggplot(data = df.diamonds,  
2         mapping = aes(x = y)) +  
3         geom_histogram(binwidth = 0.1) +  
4         coord_cartesian(xlim = c(3, 10))
```



## Density

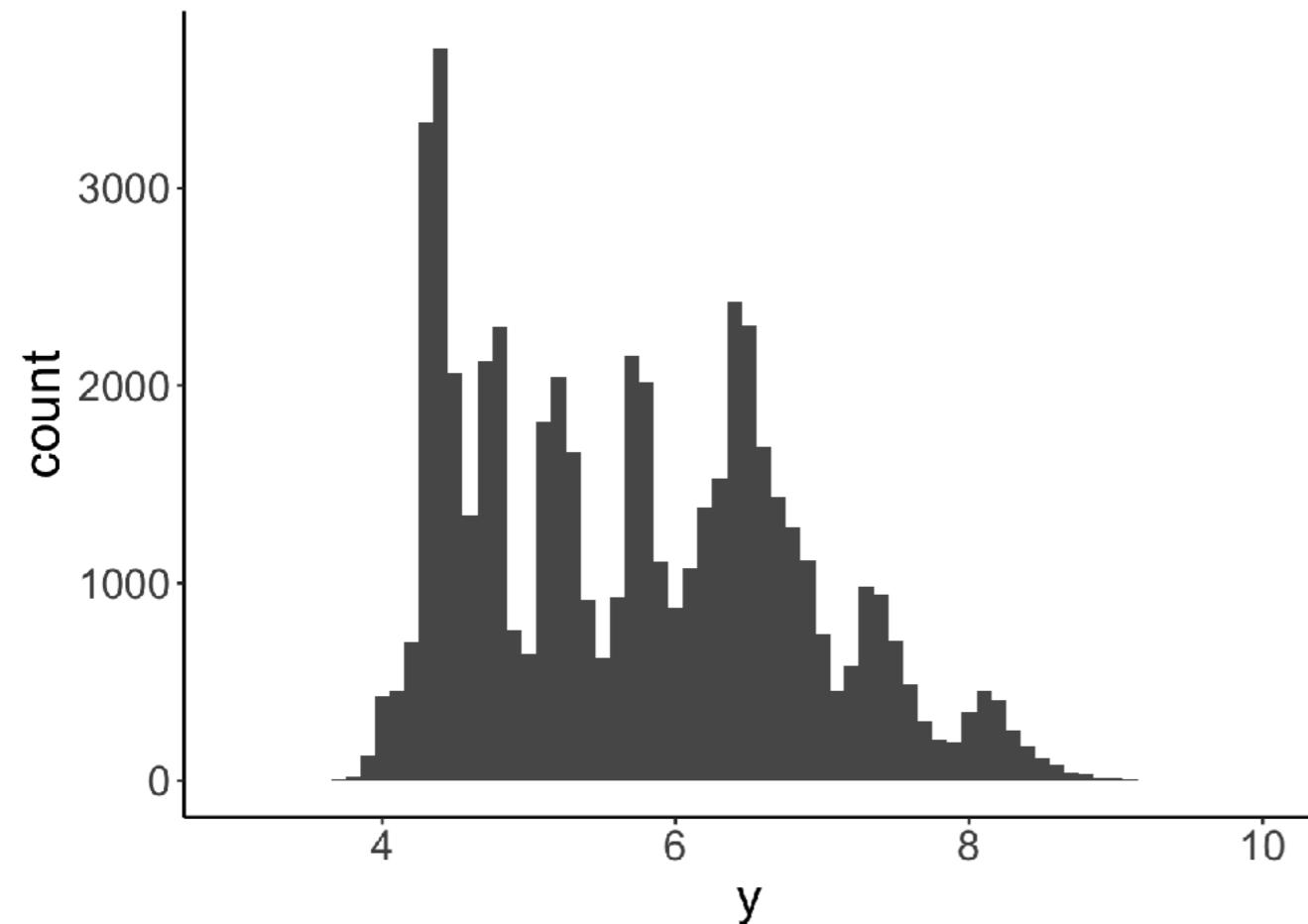
```
1 ggplot(data = df.diamonds,  
2         mapping = aes(x = y)) +  
3         geom_density() +  
4         coord_cartesian(xlim = c(3, 10))
```



# You've seen `density()` before

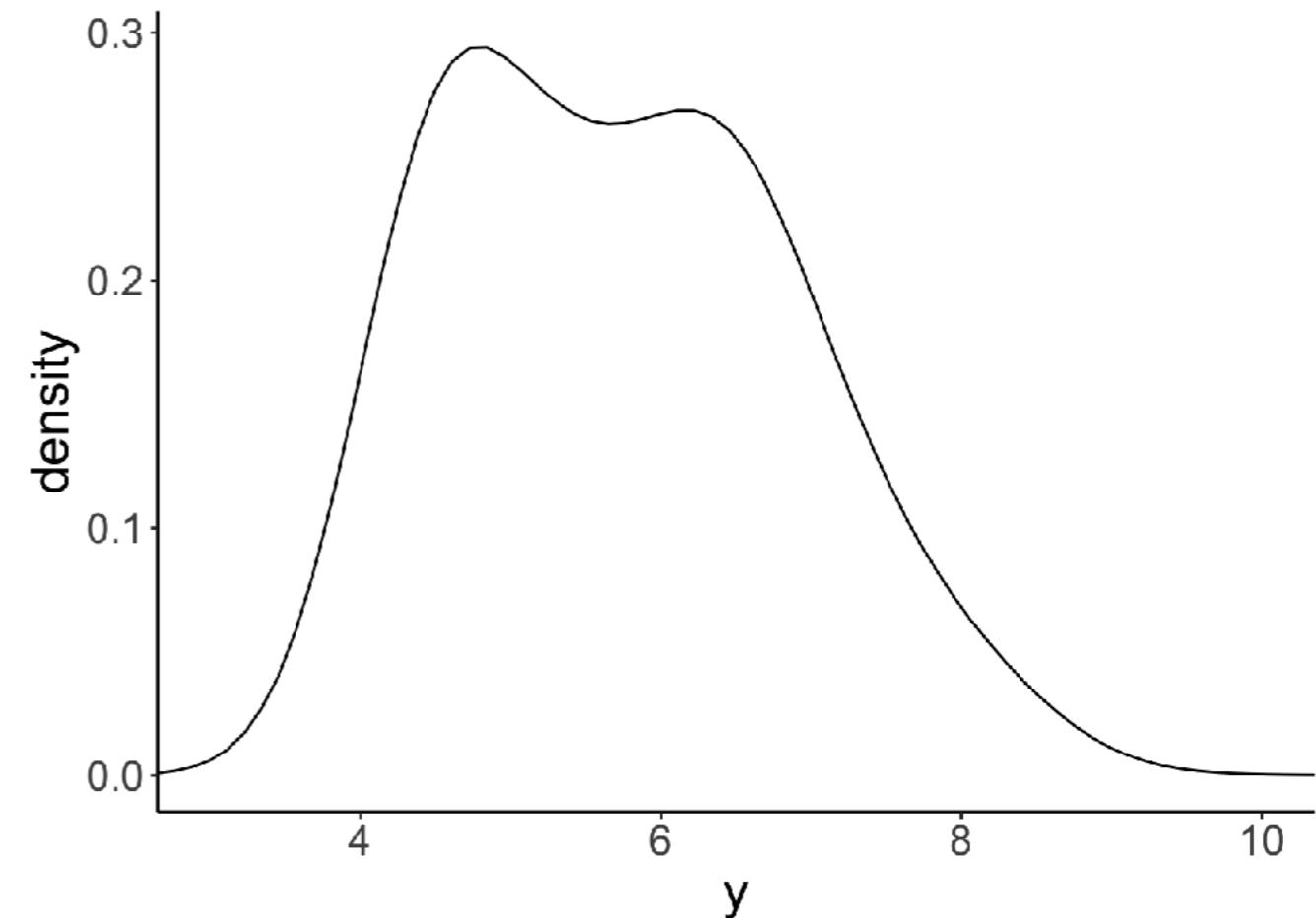
## Histogram

```
1 ggplot(data = df.diamonds,  
2         mapping = aes(x = y)) +  
3         geom_histogram(binwidth = 0.1) +  
4         coord_cartesian(xlim = c(3, 10))
```



## Density

```
1 ggplot(data = df.diamonds,  
2         mapping = aes(x = y)) +  
3         geom_density(bw = 0.5) +  
4         coord_cartesian(xlim = c(3, 10))
```

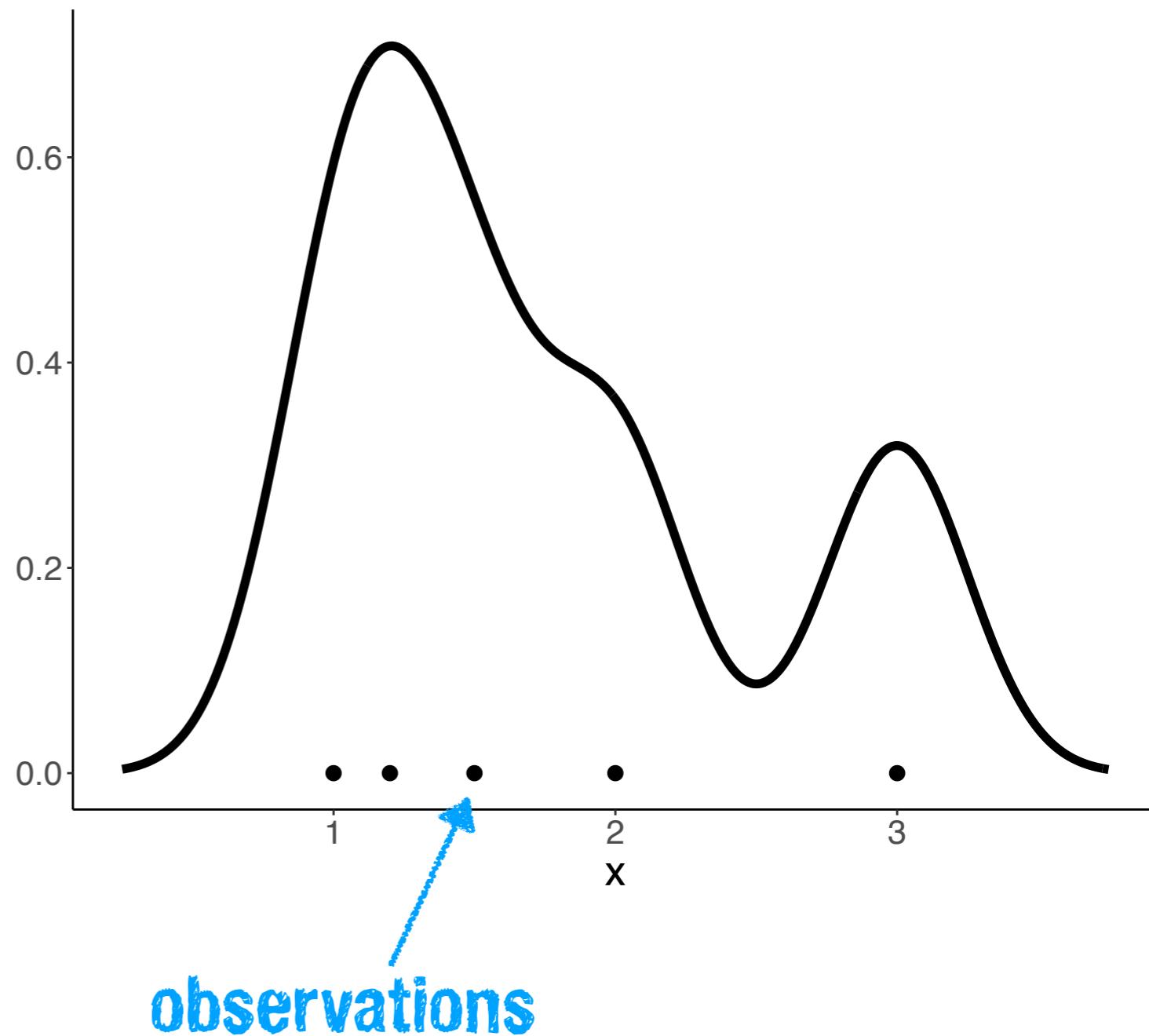


# Understanding density()

```
1 # calculate density  
2 observations = c(1, 1.2, 1.5, 2, 3)  
3 bandwidth = 0.25  
4 density = density(observations,  
5   kernel = "gaussian",  
6   bw = bandwidth,  
7   n = 512)
```

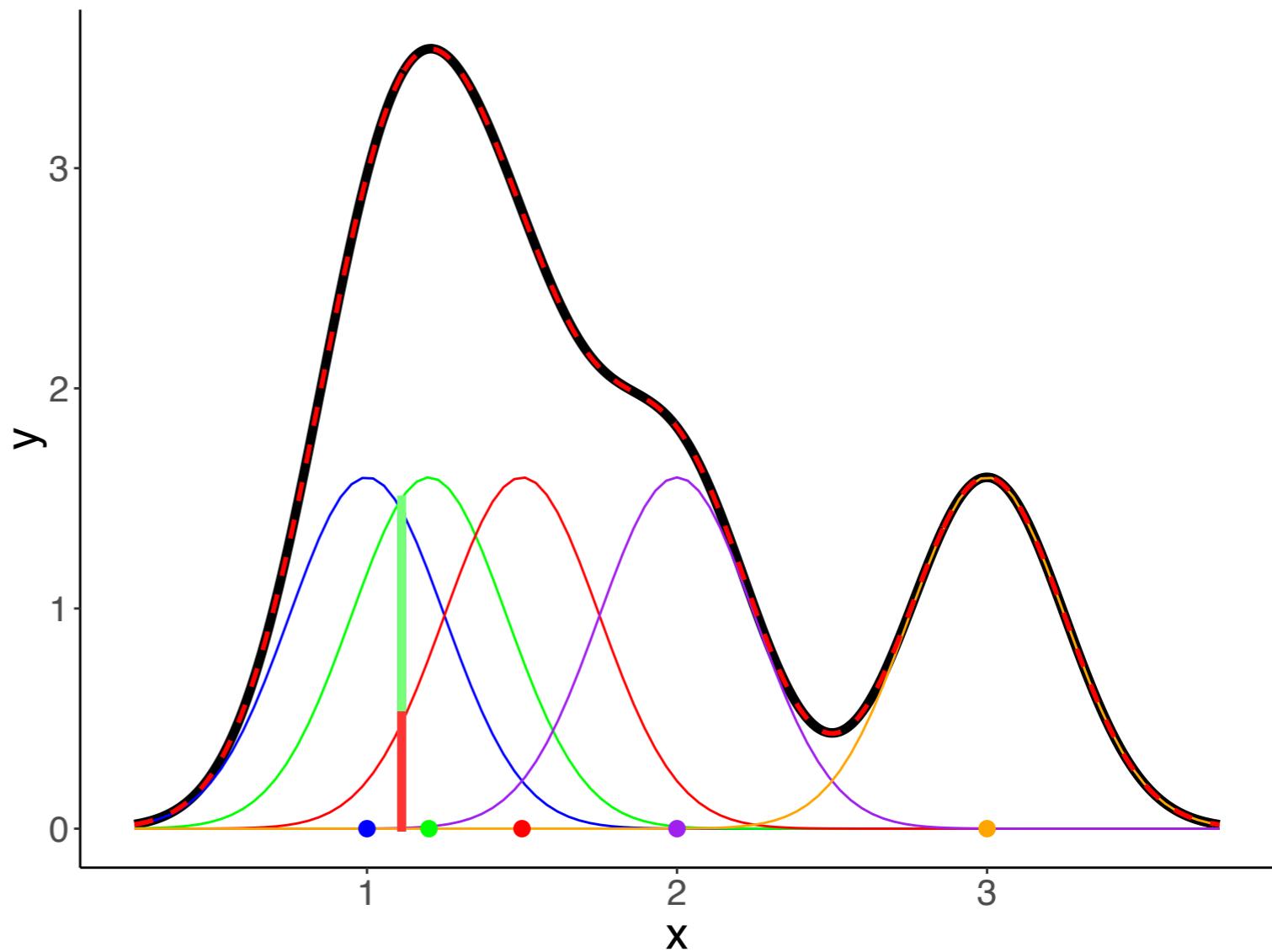
| x     | y     |
|-------|-------|
| 0.250 | 0.004 |
| 0.257 | 0.004 |
| 0.264 | 0.005 |
| 0.271 | 0.005 |
| 0.277 | 0.005 |
| 0.284 | 0.006 |

nrow = 512



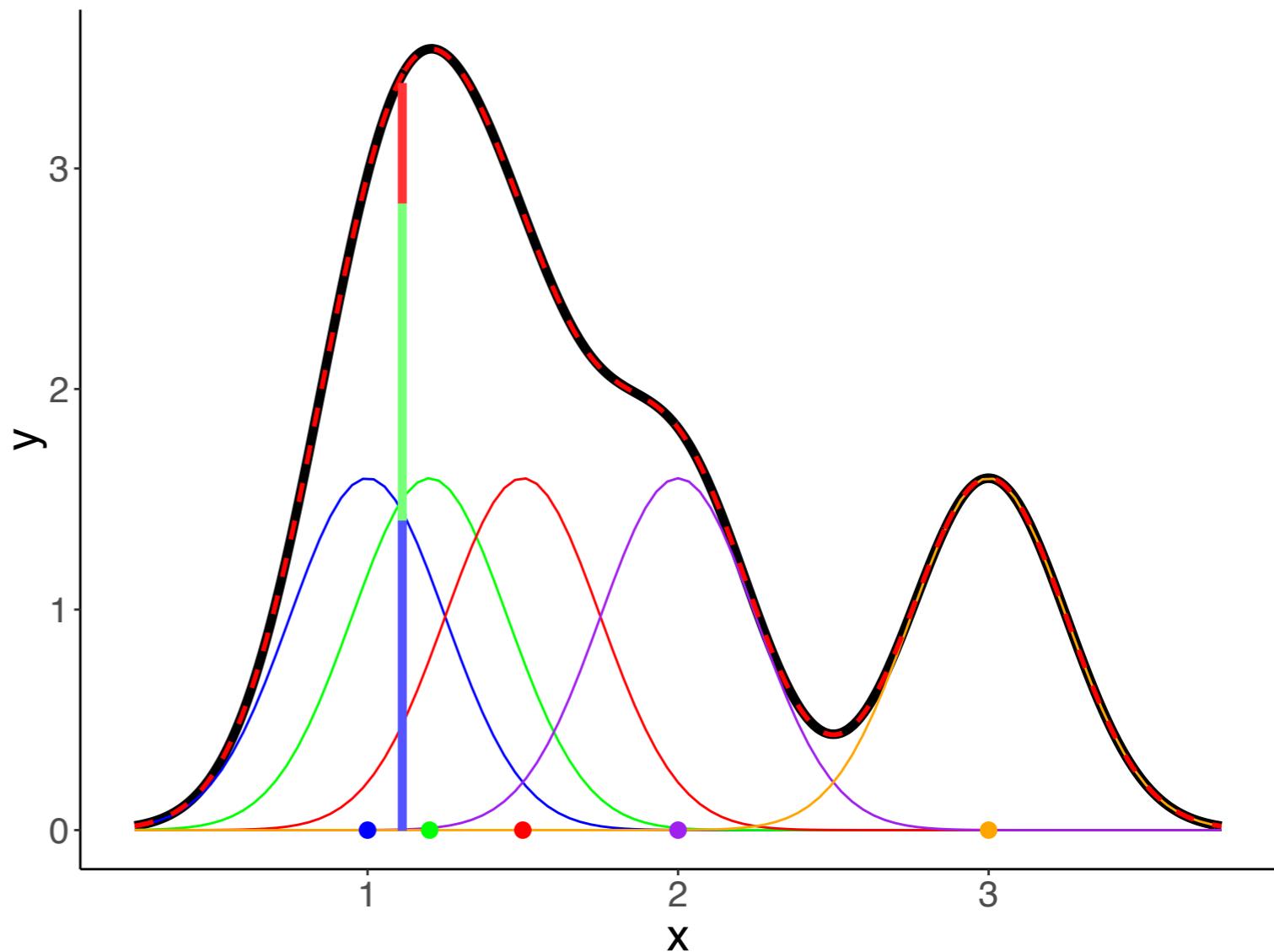
# Understanding density()

| x     | y     | observation_1 | observation_2 | observation_3 | observation_4 | observation_5 | sum_norm |
|-------|-------|---------------|---------------|---------------|---------------|---------------|----------|
| 0.250 | 0.019 | 0.018         | 0.001         | 0             | 0             | 0             | 0.019    |
| 0.257 | 0.021 | 0.019         | 0.001         | 0             | 0             | 0             | 0.021    |
| 0.264 | 0.023 | 0.021         | 0.001         | 0             | 0             | 0             | 0.022    |
| 0.271 | 0.024 | 0.023         | 0.002         | 0             | 0             | 0             | 0.024    |
| 0.277 | 0.027 | 0.024         | 0.002         | 0             | 0             | 0             | 0.026    |
| 0.284 | 0.029 | 0.026         | 0.002         | 0             | 0             | 0             | 0.028    |

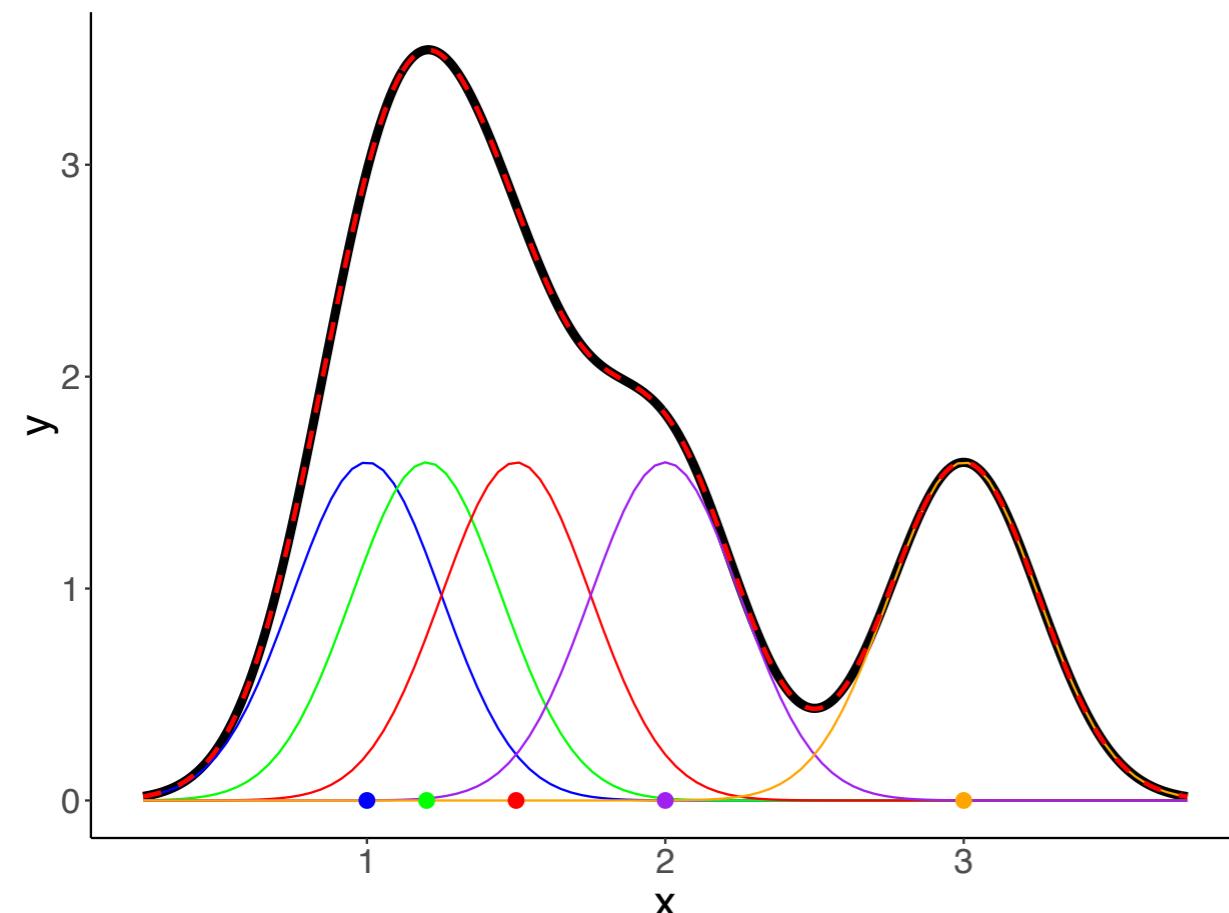


# Understanding density()

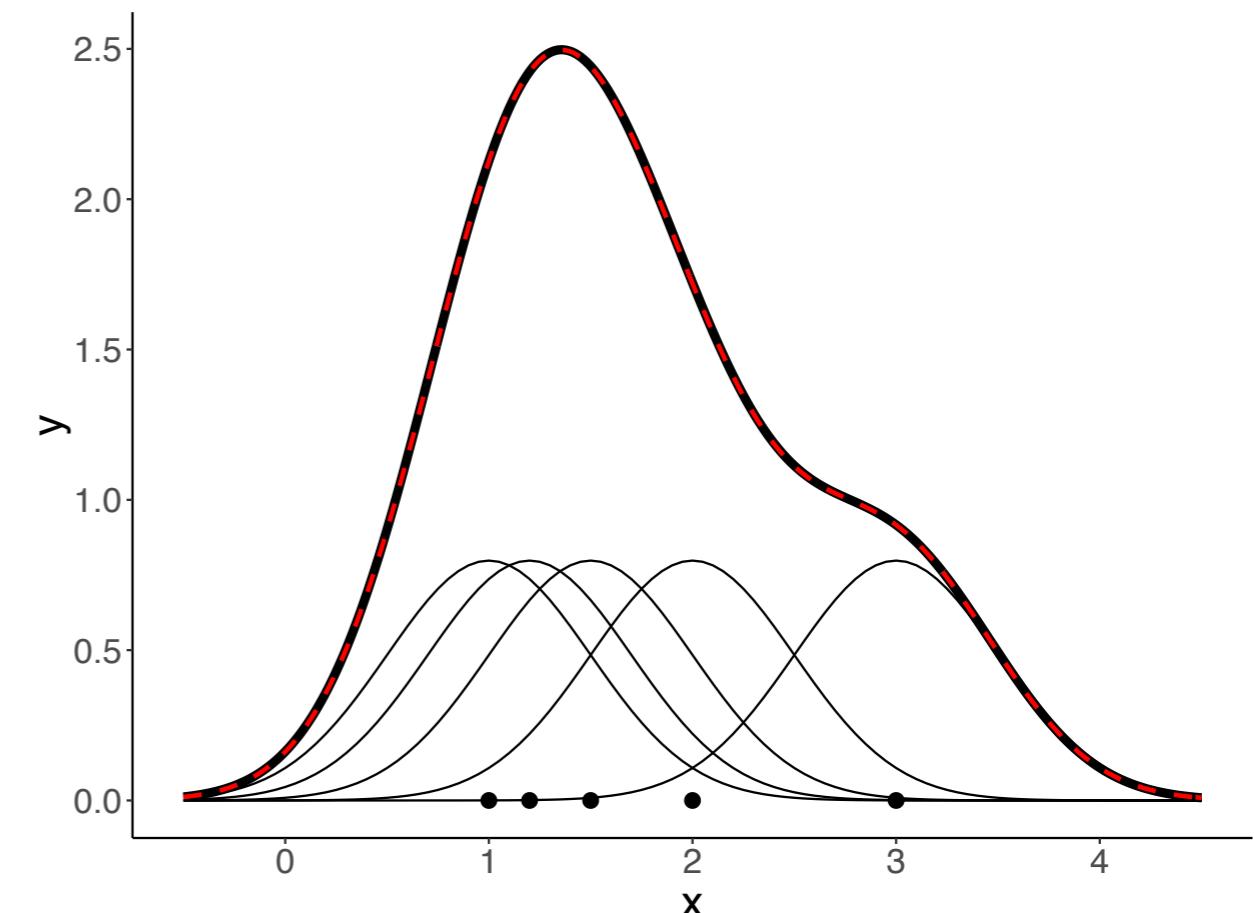
| x     | y     | observation_1 | observation_2 | observation_3 | observation_4 | observation_5 | sum_norm |
|-------|-------|---------------|---------------|---------------|---------------|---------------|----------|
| 0.250 | 0.019 | 0.018         | 0.001         | 0             | 0             | 0             | 0.019    |
| 0.257 | 0.021 | 0.019         | 0.001         | 0             | 0             | 0             | 0.021    |
| 0.264 | 0.023 | 0.021         | 0.001         | 0             | 0             | 0             | 0.022    |
| 0.271 | 0.024 | 0.023         | 0.002         | 0             | 0             | 0             | 0.024    |
| 0.277 | 0.027 | 0.024         | 0.002         | 0             | 0             | 0             | 0.026    |
| 0.284 | 0.029 | 0.026         | 0.002         | 0             | 0             | 0             | 0.028    |



# Understanding density()



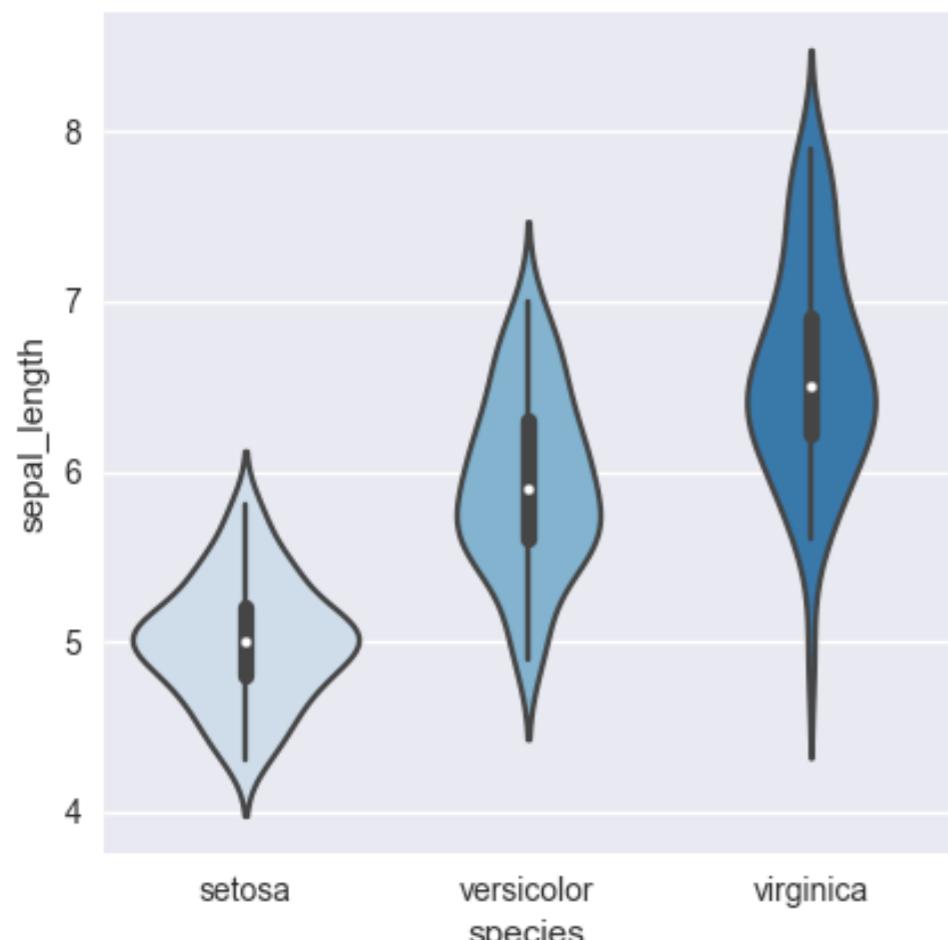
**density**(bw = 0.25)



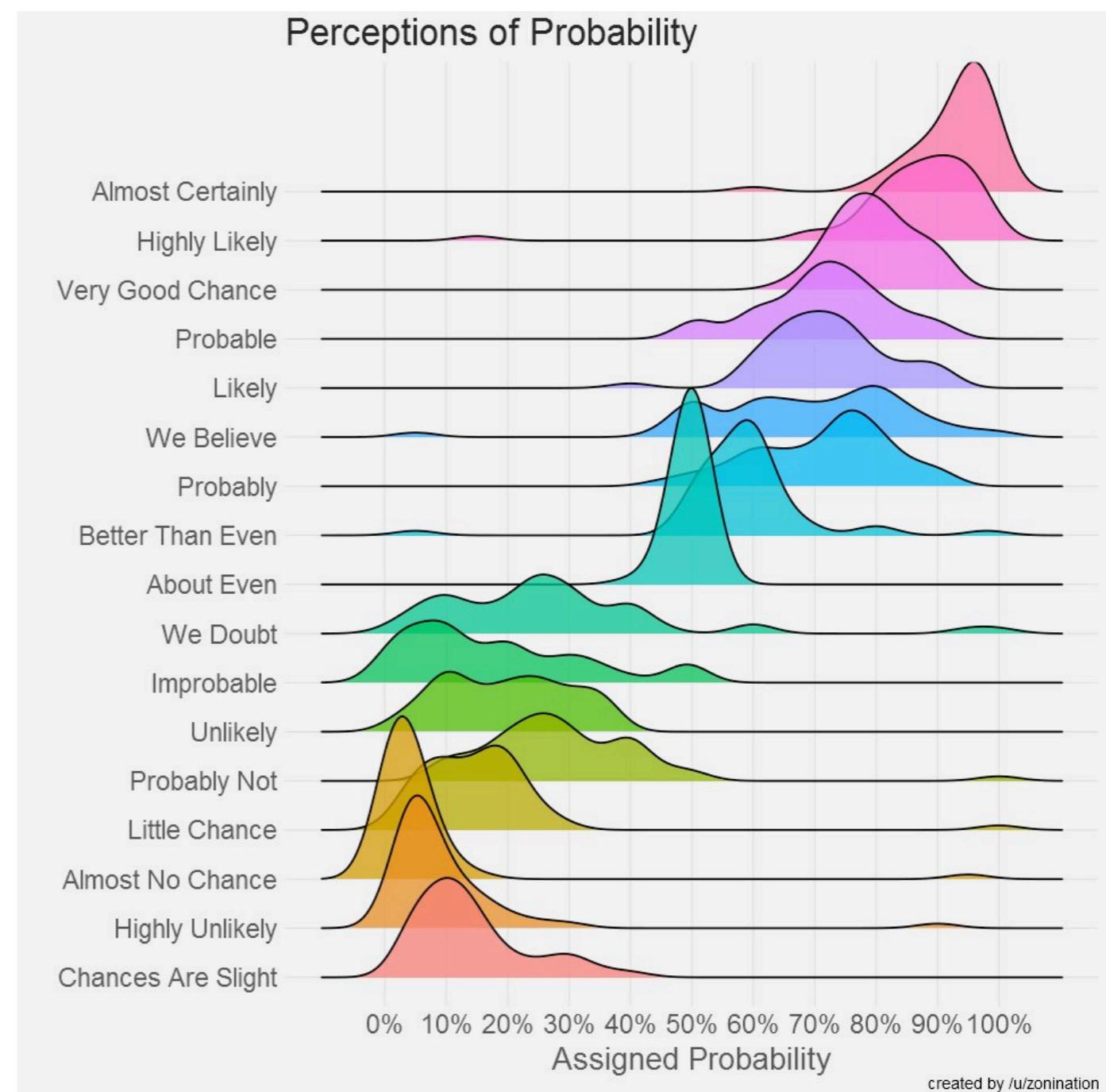
**density**(bw = 0.5)

# Understanding density()

violinplot

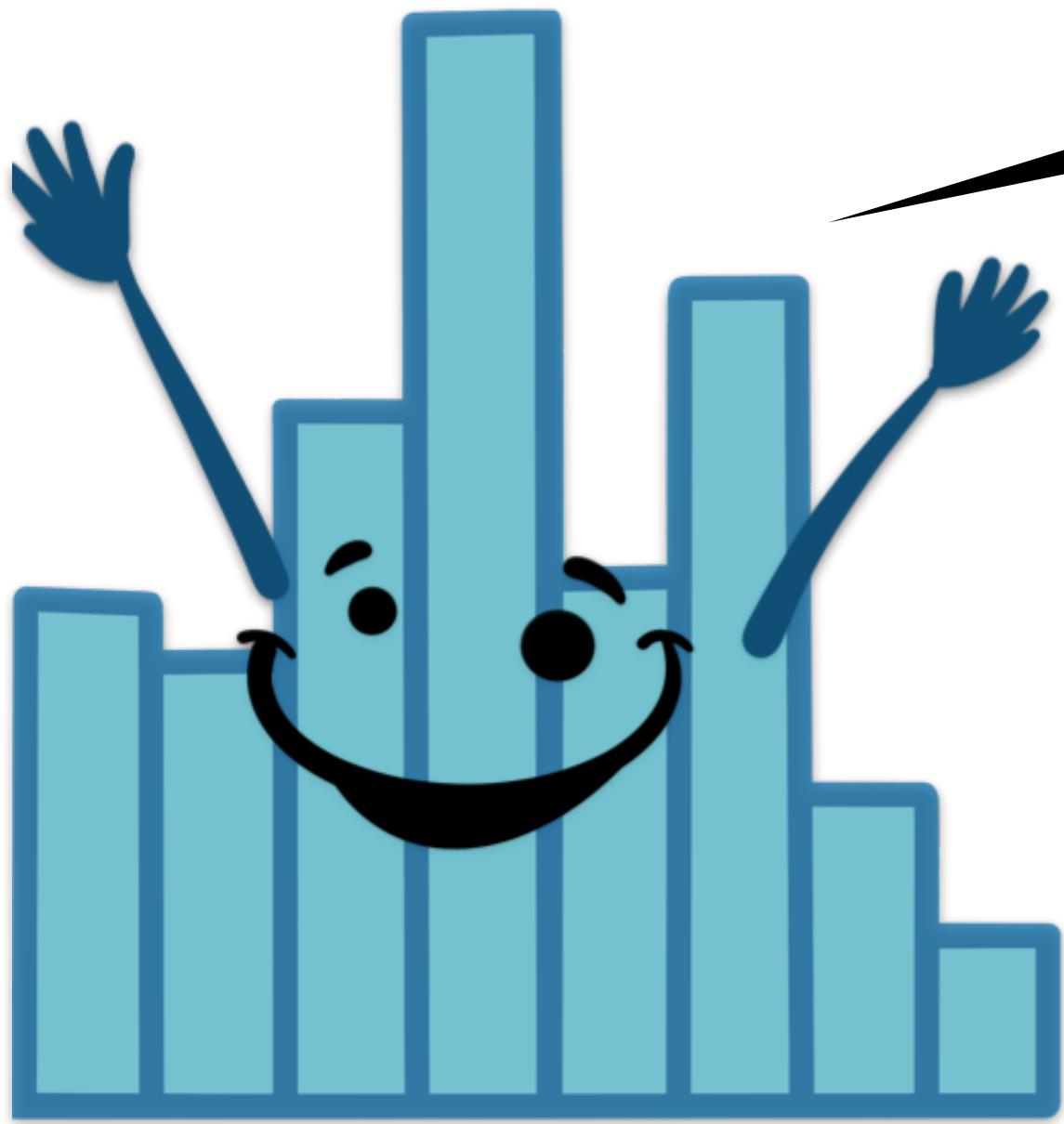


joyplot



02:00

stretch break!



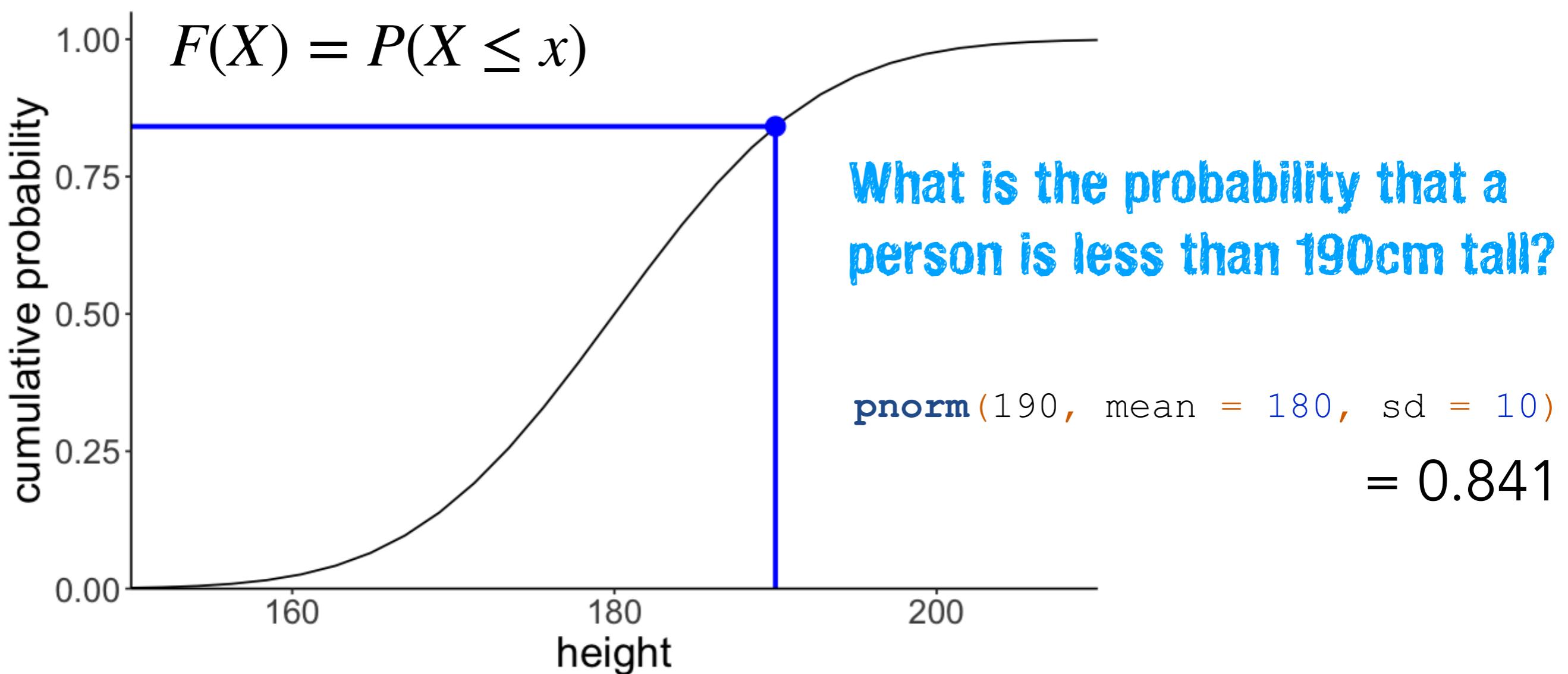
# Asking probability distributions for answers



# Cumulative probability distribution

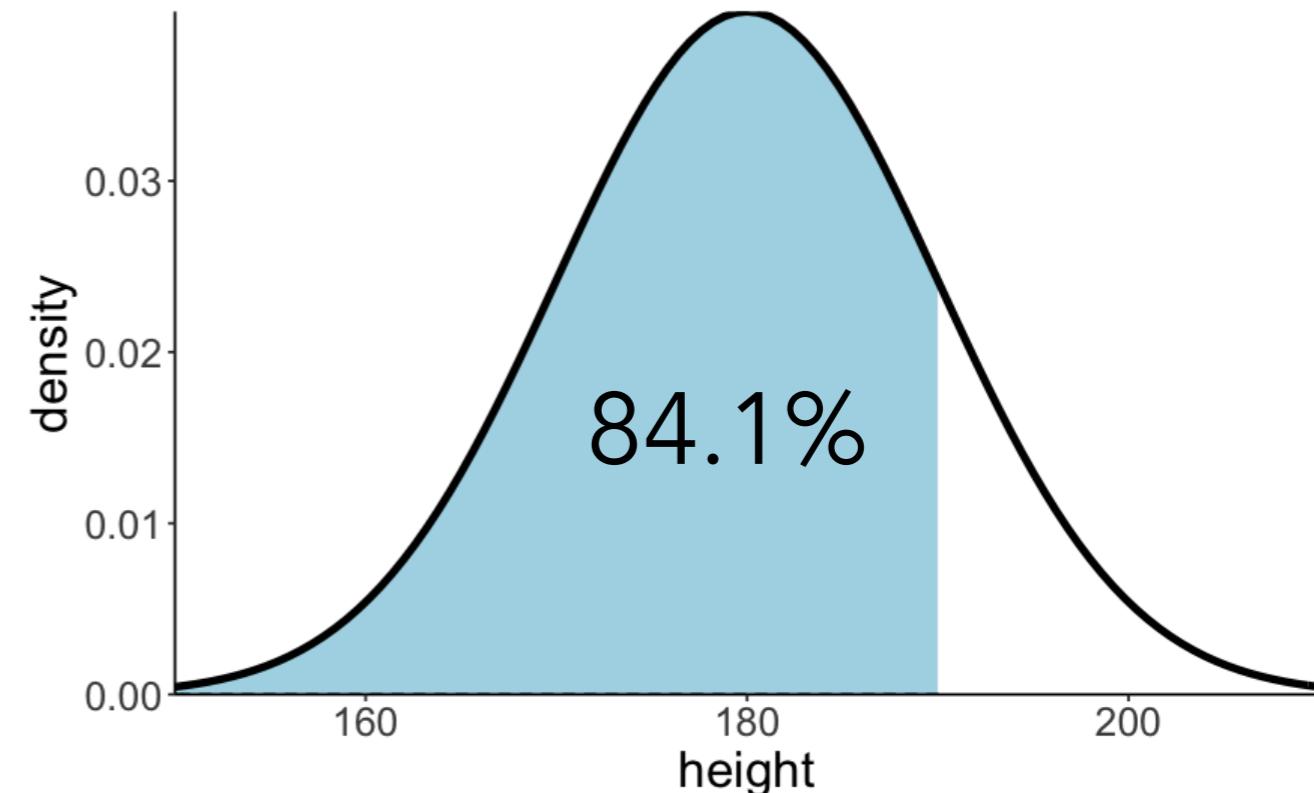
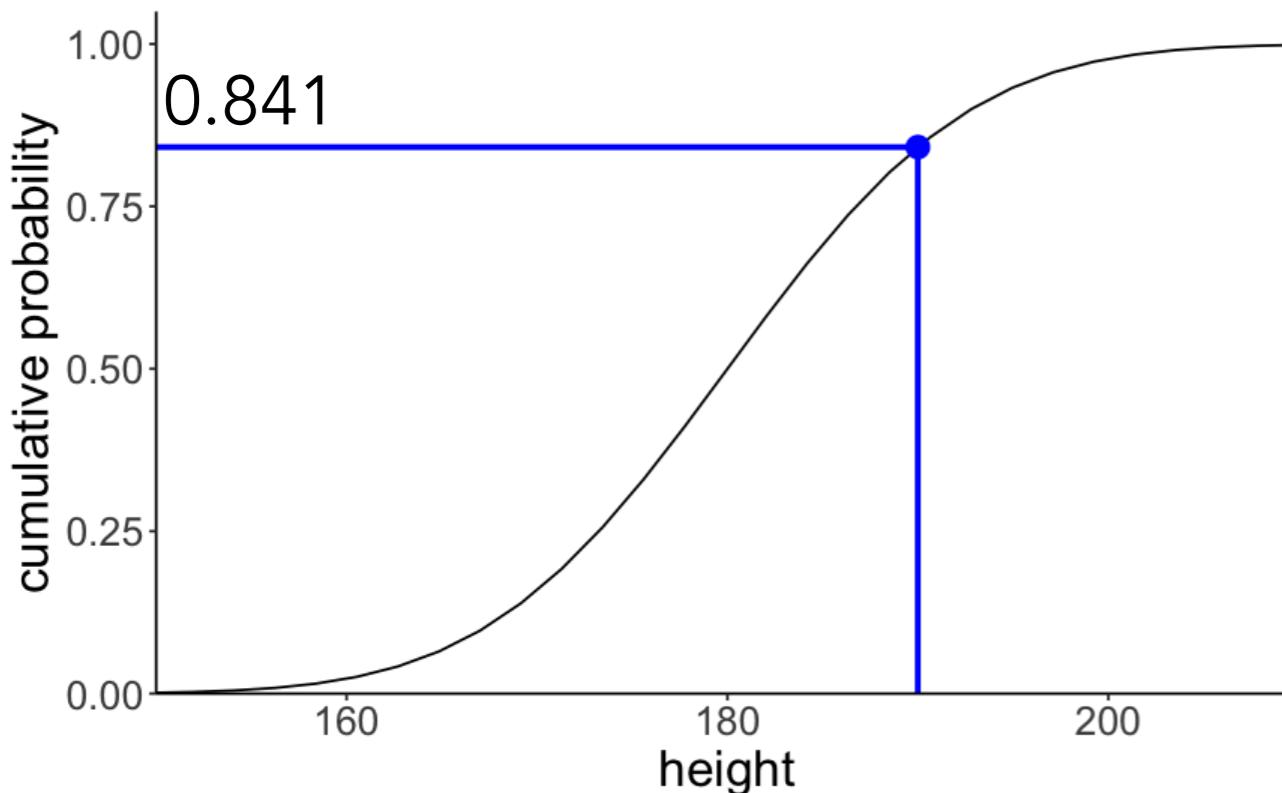
```
1 ggplot(data = tibble(x = c(150, 210)),  
2         mapping = aes(x = x)) +  
3   stat_function(fun = ~ pnorm(q = .,  
4                               mean = 180,  
5                               sd = 10))
```

**p** = probability  
cumulative distribution function



# Computing probabilities

`pnorm(190, mean = 180, sd = 10)`

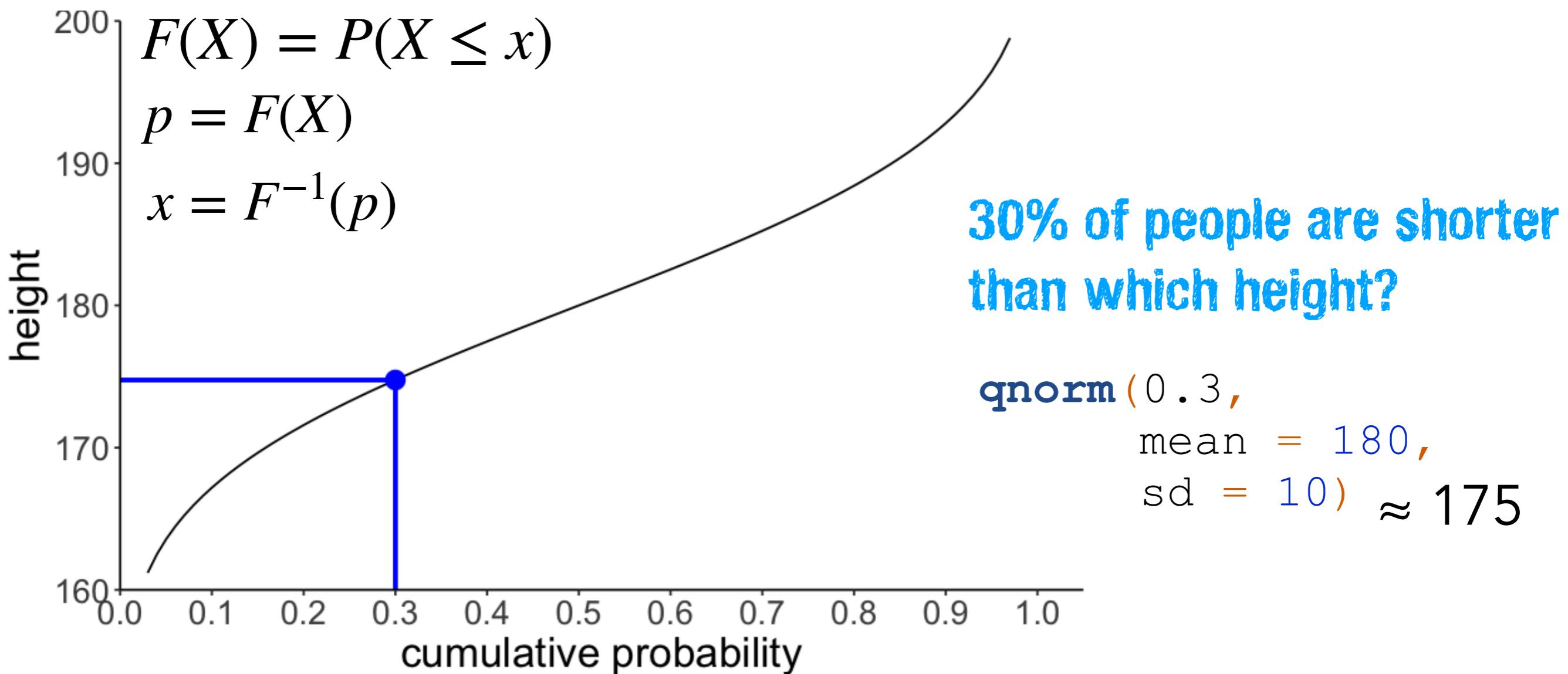


`pnorm(x)` returns the integral from  $-\infty$  to  $x$  of the probability density function

# Inverse cumulative distribution function

```
1 ggplot(data = tibble(x = c(0, 1)),  
2         mapping = aes(x = x)) +  
3   stat_function(fun = ~ qnorm(p = .,  
4                               mean = 180,  
5                               sd = 10))
```

q = quantile  
inverse cumulative distribution function



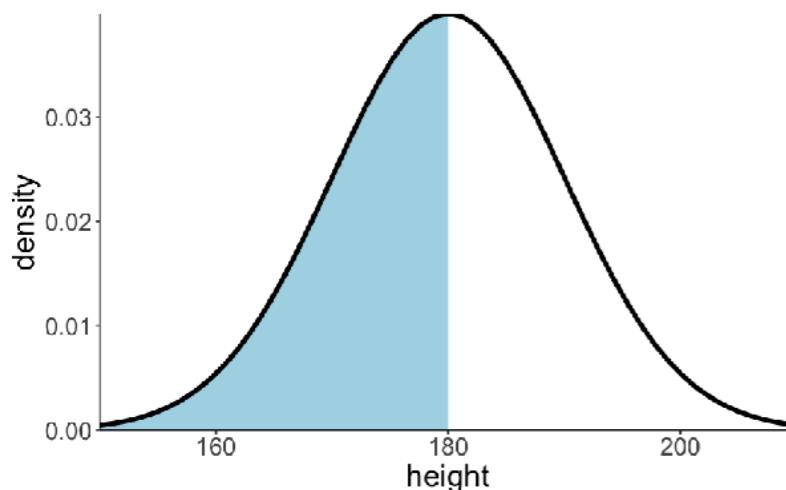
# What proportion of people are between 170cm and 180cm?

## Analytic solution

```
pnorm(180,  
      mean = 180,  
      sd = 10)
```

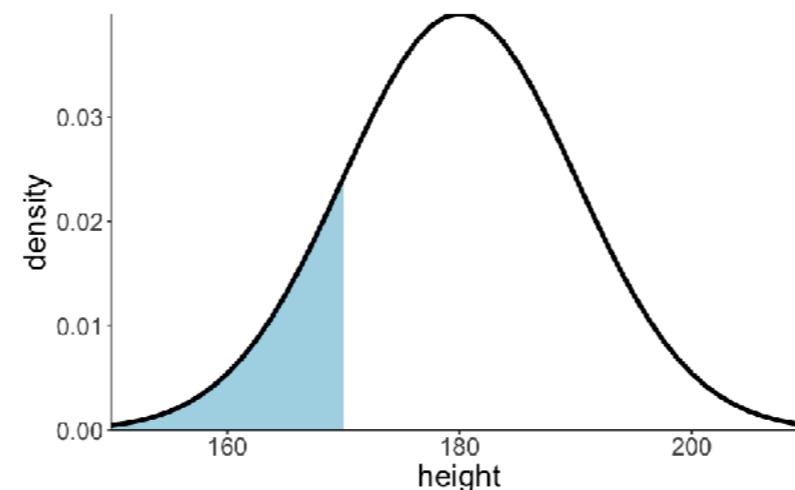
```
pnorm(170,  
      mean = 180,  
      sd = 10)
```

50%



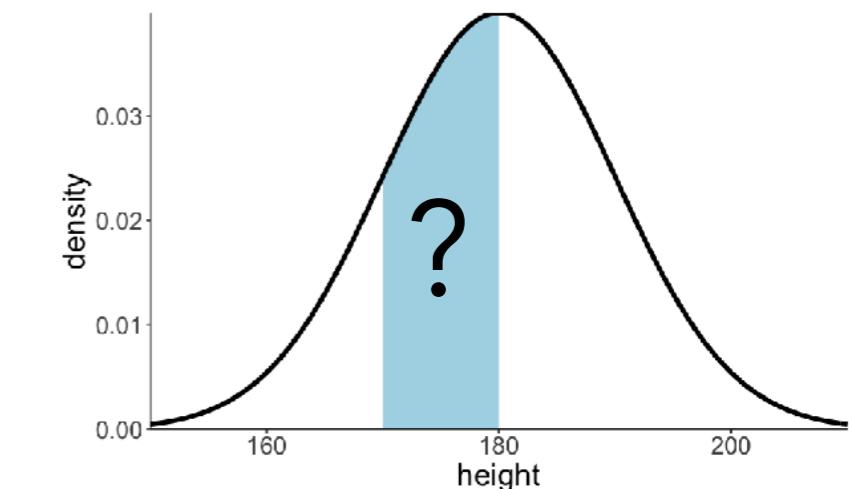
smaller than 180cm

16%



smaller than 170cm

34%

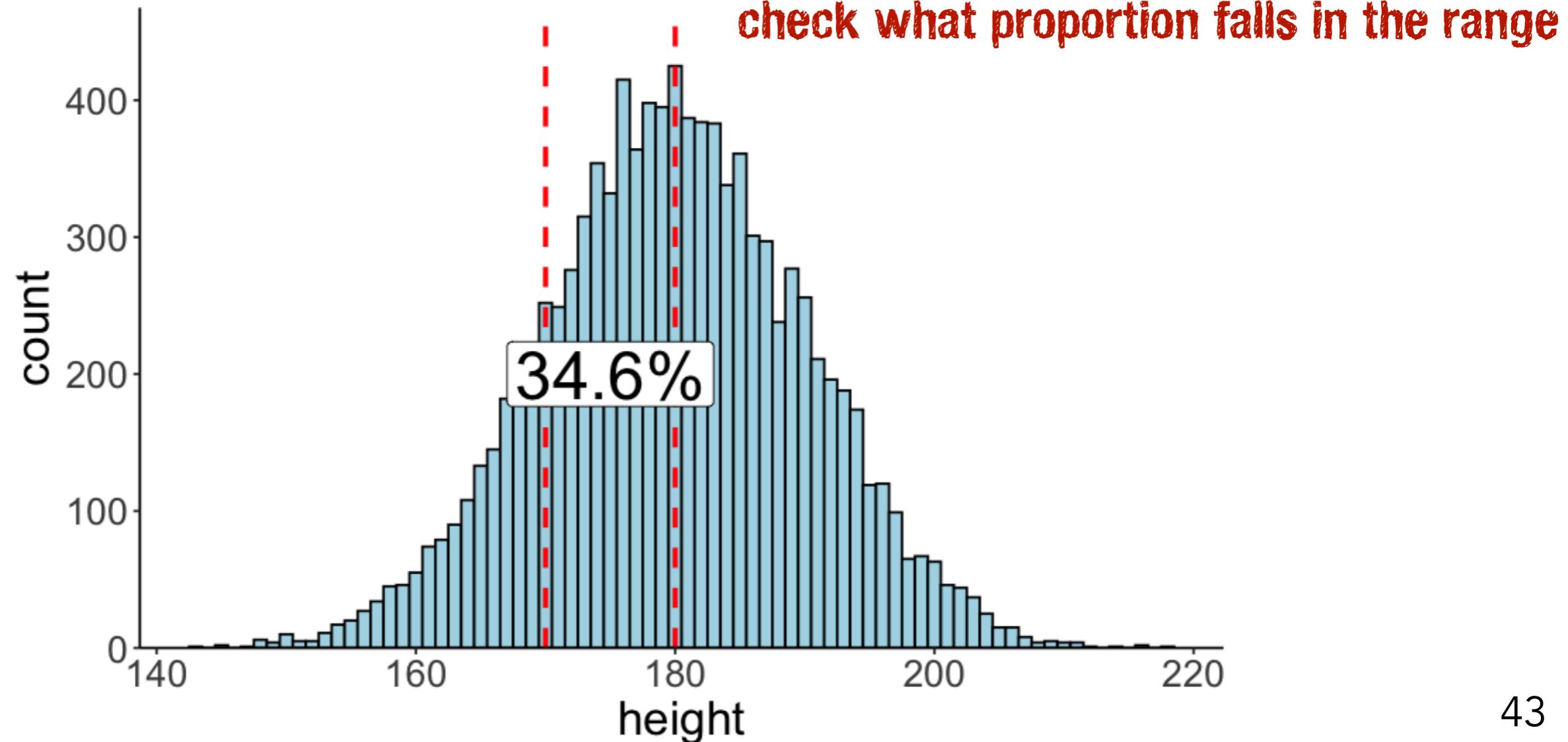


between 170cm  
and 180cm

# What proportion of people are between 170cm and 180cm?

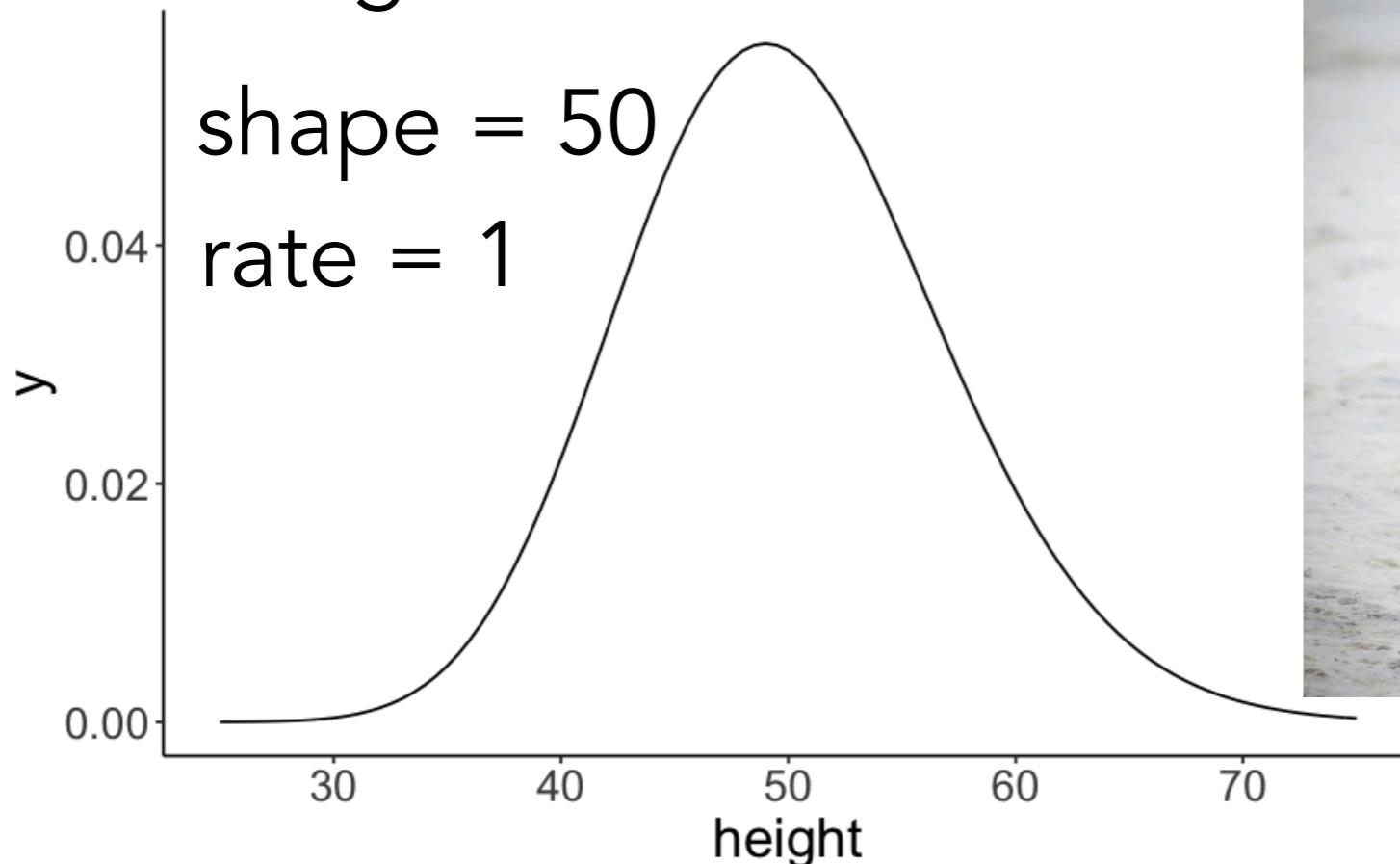
## Sampling solution

```
1 tibble(height = rnorm(n = 10000, mean = 180, sd = 10)) %>%
2   summarize(probability = sum(height > 170 & height < 180) / n())
```



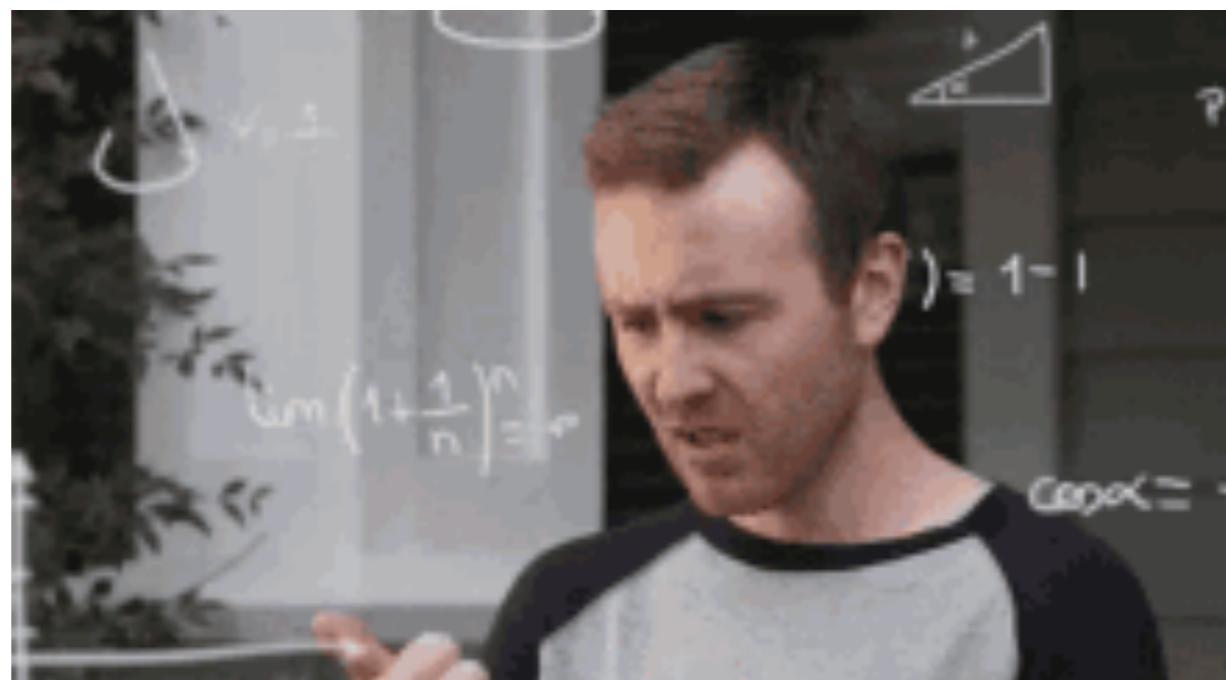
# Answering questions about Penguins

## gamma distribution



1. Make this plot
2. A 60cm tall Penguin claims that no more than 10% are taller than her. Is she correct?
3. Are there more penguins between 50 and 55cm or between 55 and 65cm?
4. What size is a Penguin who is taller than 75% of the rest?

# Show solutions in R Studio



# Quick recap

- we can draw random samples in R via
  - `sample()` from a vector
  - `slice_sample()` from a data frame
- generate random samples from a probability distribution via
  - `rnorm()`, `rbinom()`, `rgamma()`, ...
- understand how `density()` works
- answer questions about probabilities via the
  - analytic route: `qnorm()`, `pnorm()`
  - sampling route: `rnorm()` + data wrangling

# Plan for today

- Simulating data
  - Drawing samples
  - Working with probability distributions
  - Quick detour: understanding `density()`
  - Asking probability distributions for answers
- **Doing Bayesian inference**
  - Analytic solution
  - Sampling solution

# Summer camp

**Register now for Summer Chess Camp!**



**think  
Move**  
CHESS ACADEMY

All skill levels  
welcome!

July 23 - July 27  
and  
August 13 - August 17

**[www.thinkmovechess.com](http://www.thinkmovechess.com)**



twice as many kids go to the basketball camp

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

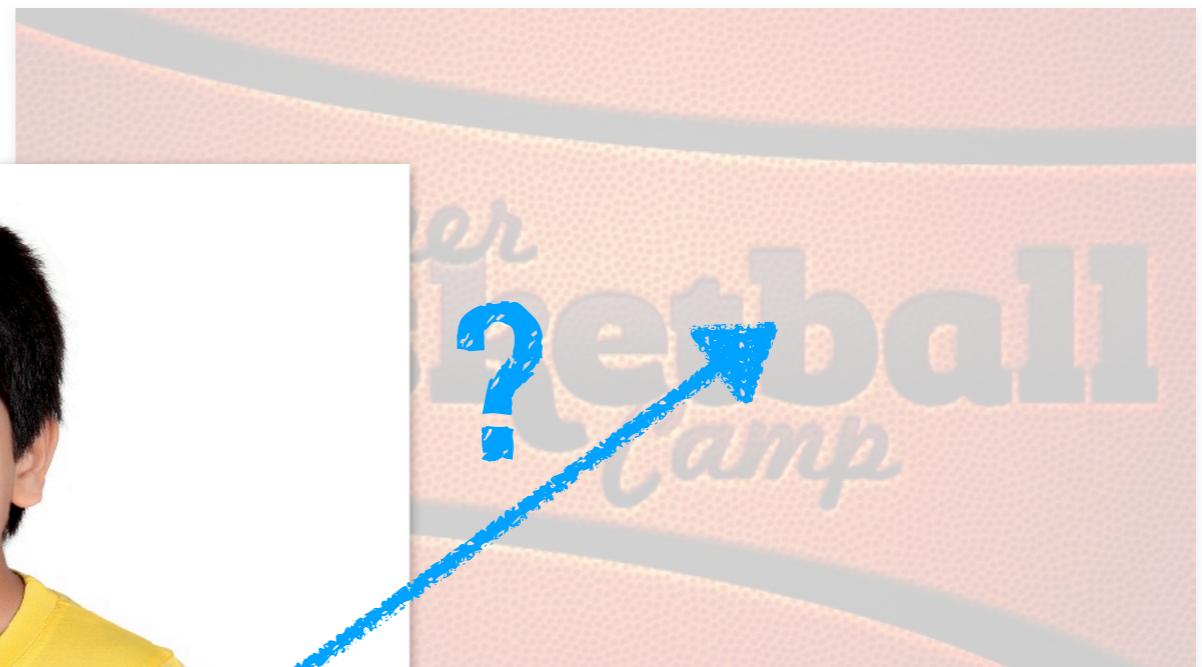
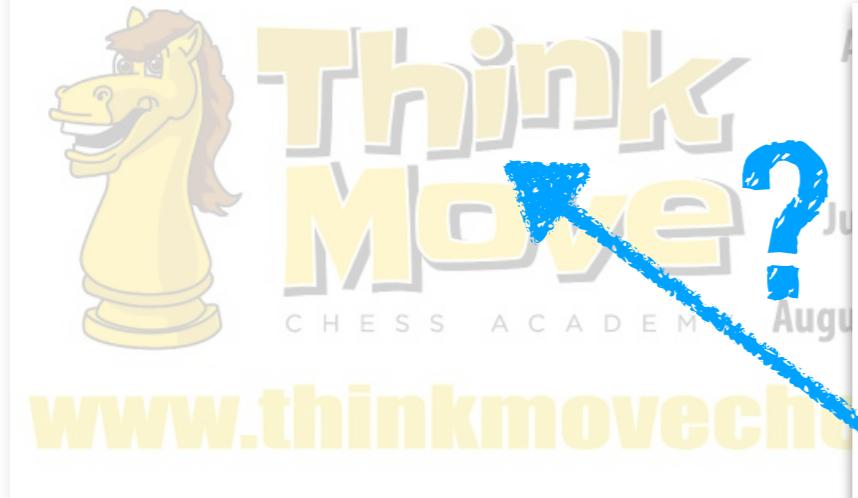


$X \sim \text{Normal}(\mu = 180, \sigma = 10)$



# Summer camp

Register now for Summer Chess Camp!



# **Analytic solution**

# Can you feel the Bayes?

$H = \{\text{basketball, chess}\}$

$D = 175 \text{ cm}$

$$\text{posterior } p(H | D) = \frac{\text{likelihood} \quad \text{prior}}{p(D)} \quad \begin{aligned} H &= \text{Hypothesis} \\ D &= \text{Data} \end{aligned}$$

**probability of the data?!**

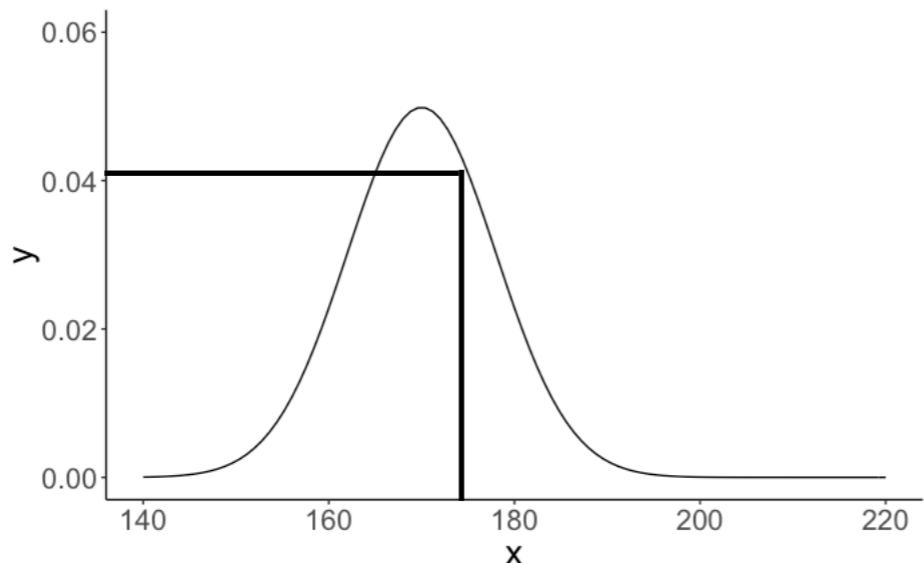
# Summer camp

**prior**

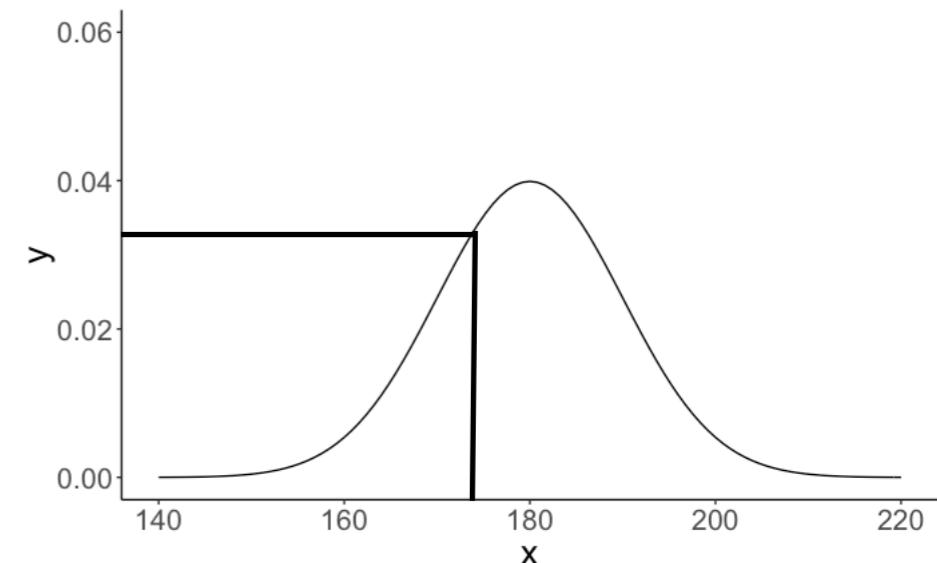
$$p(\text{chess}) = \frac{1}{3}$$

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

**likelihood**



$$\begin{aligned} \text{dnorm}(175, \text{mean} = 170, \text{sd} = 8) \\ = 0.041 \end{aligned}$$



$$\begin{aligned} \text{dnorm}(175, \text{mean} = 180, \text{sd} = 10) \\ = 0.035 \end{aligned}$$

**posterior**

$$p(\text{sport} = \text{basketball} | \text{height} = 175) = \frac{p(175 | \text{basketball}) \cdot p(\text{basketball})}{p(175)}$$

**likelihood**      **prior**

**data**

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

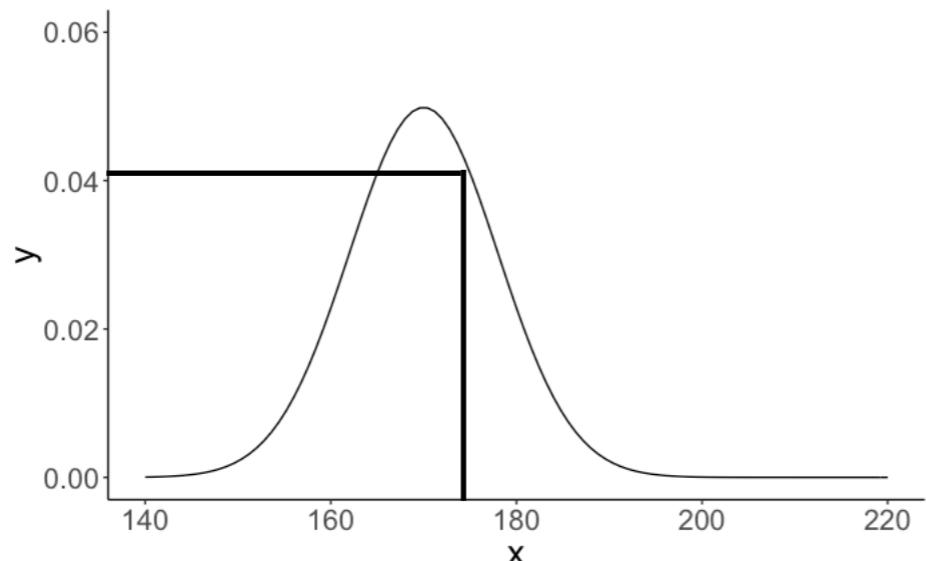
# Summer camp

**prior**

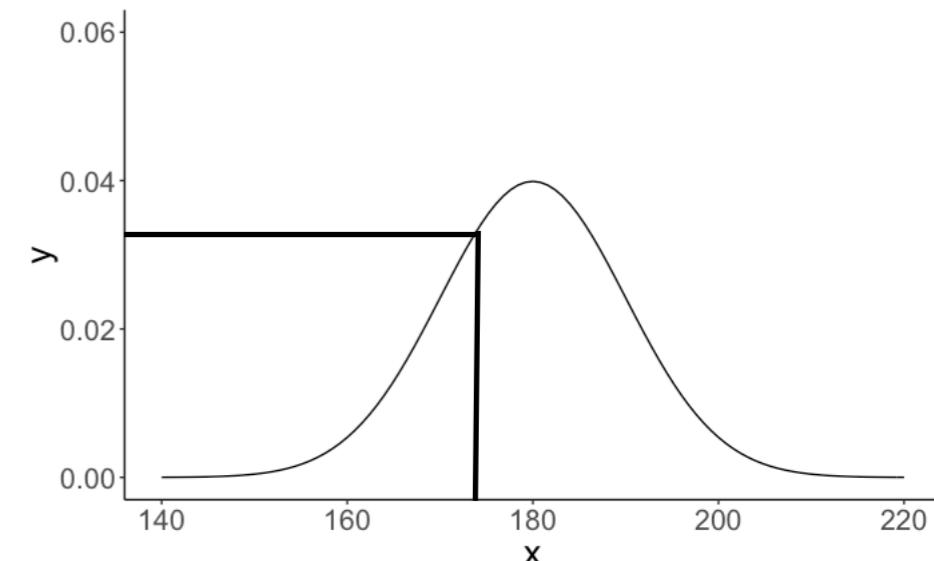
$$p(\text{chess}) = \frac{1}{3}$$

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

**likelihood**



$$\begin{aligned} \text{dnorm}(175, \text{mean} = 170, \text{sd} = 8) \\ = 0.041 \end{aligned}$$



$$\begin{aligned} \text{dnorm}(175, \text{mean} = 180, \text{sd} = 10) \\ = 0.035 \end{aligned}$$

**posterior**

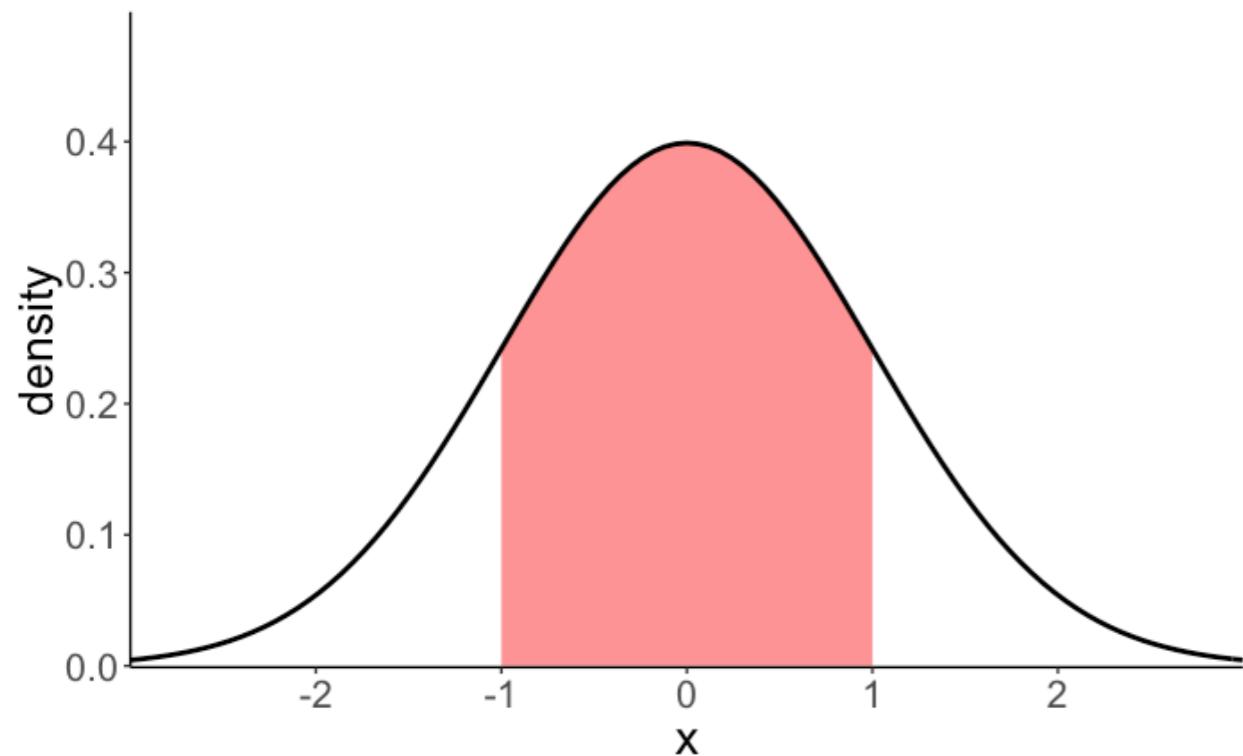
$$p(\text{basketball} | 175) = \frac{p(175 | \text{basketball}) \cdot p(\text{basketball})}{p(175 | \text{basketball}) \cdot p(\text{basketball}) + p(175 | \text{chess}) \cdot p(\text{chess})}$$

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

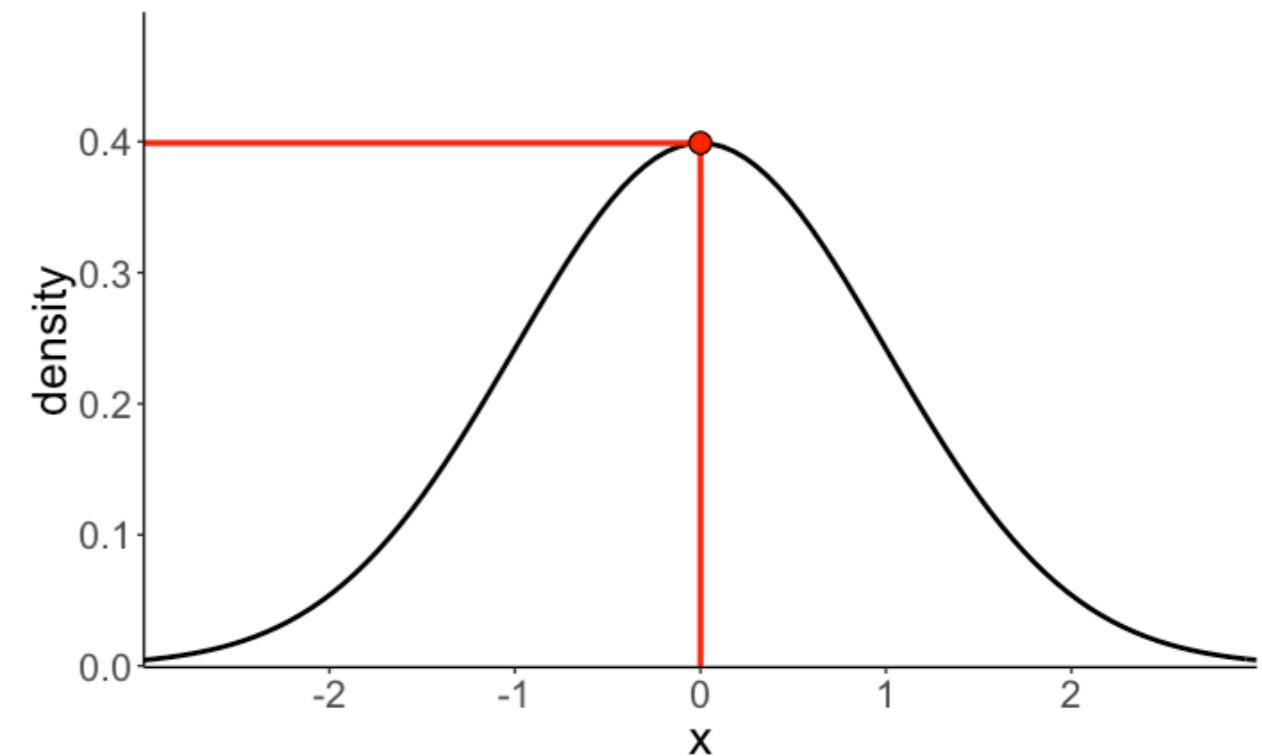
send the kid to  
the basketball  
gym!

# Probability vs. likelihood

**Probability**

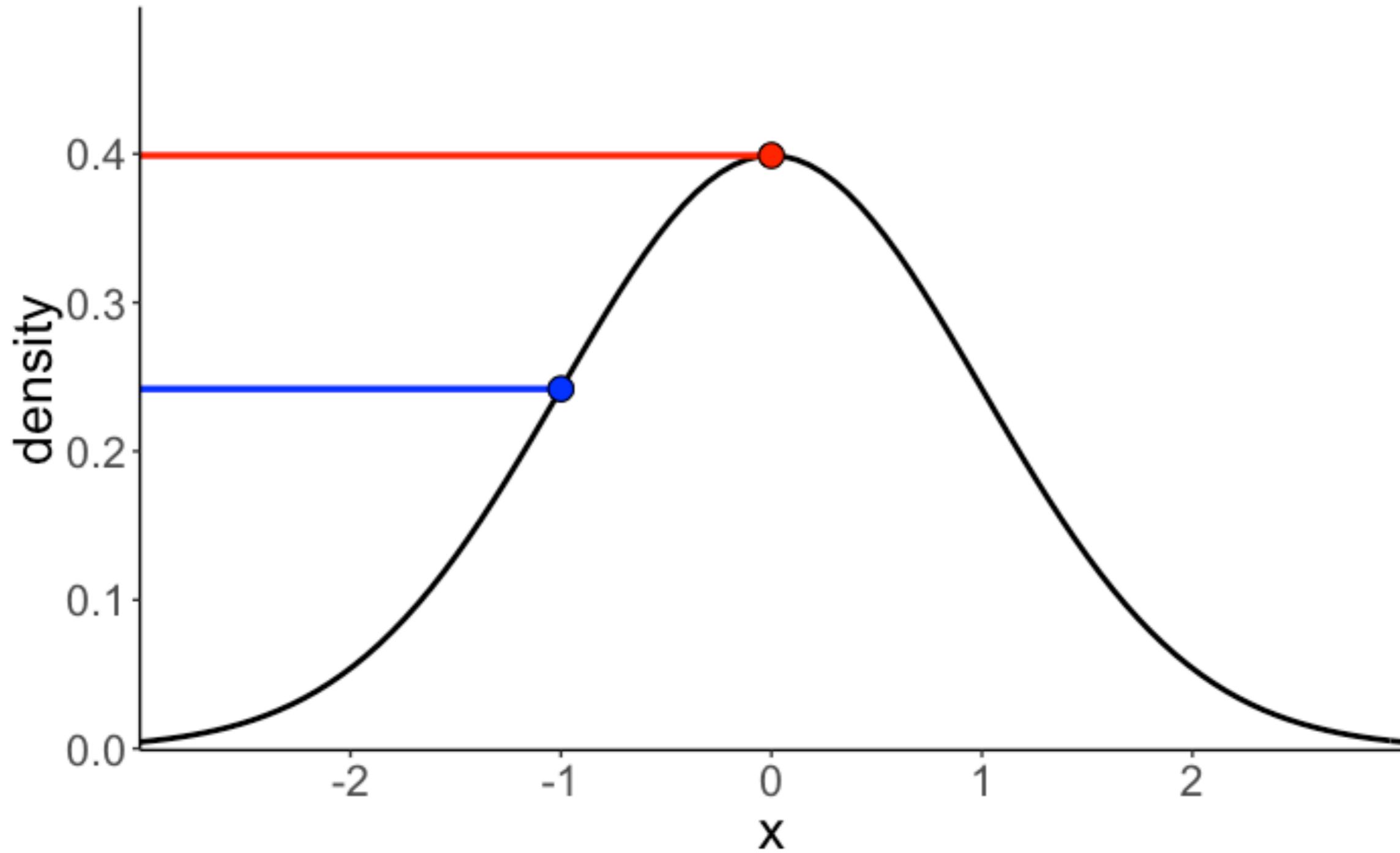


**Likelihood**



# Probability vs. likelihood

$$\text{dnorm}(0) / \text{dnorm}(-1) = 1.6487$$

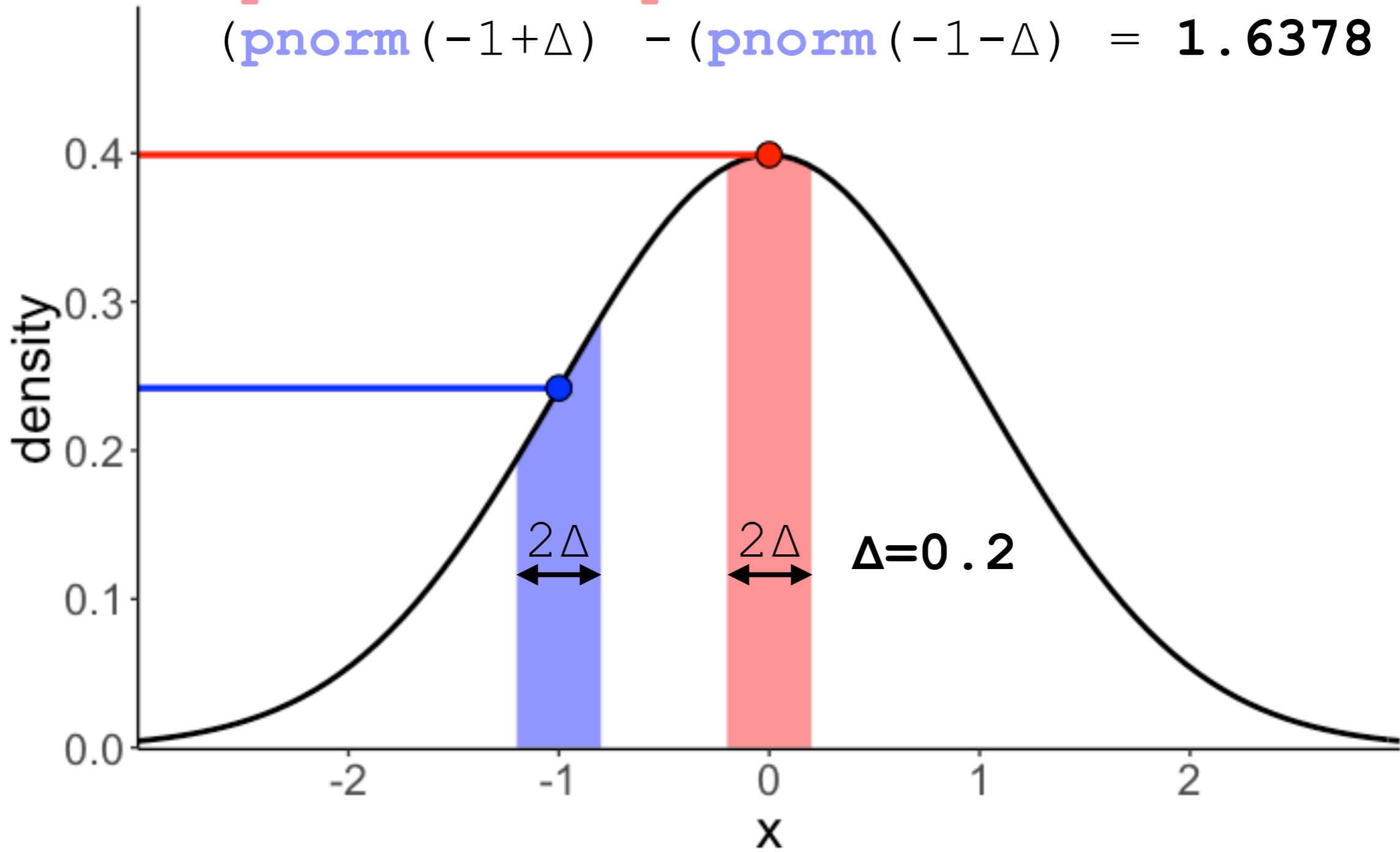


relative probability of one value vs. another

# Probability vs. likelihood

$$\text{dnorm}(0) / \text{dnorm}(-1) = 1.6487$$

$$\frac{(\text{pnorm}(0+\Delta) - \text{pnorm}(0-\Delta))}{(\text{pnorm}(-1+\Delta) - \text{pnorm}(-1-\Delta))} = 1.6378$$

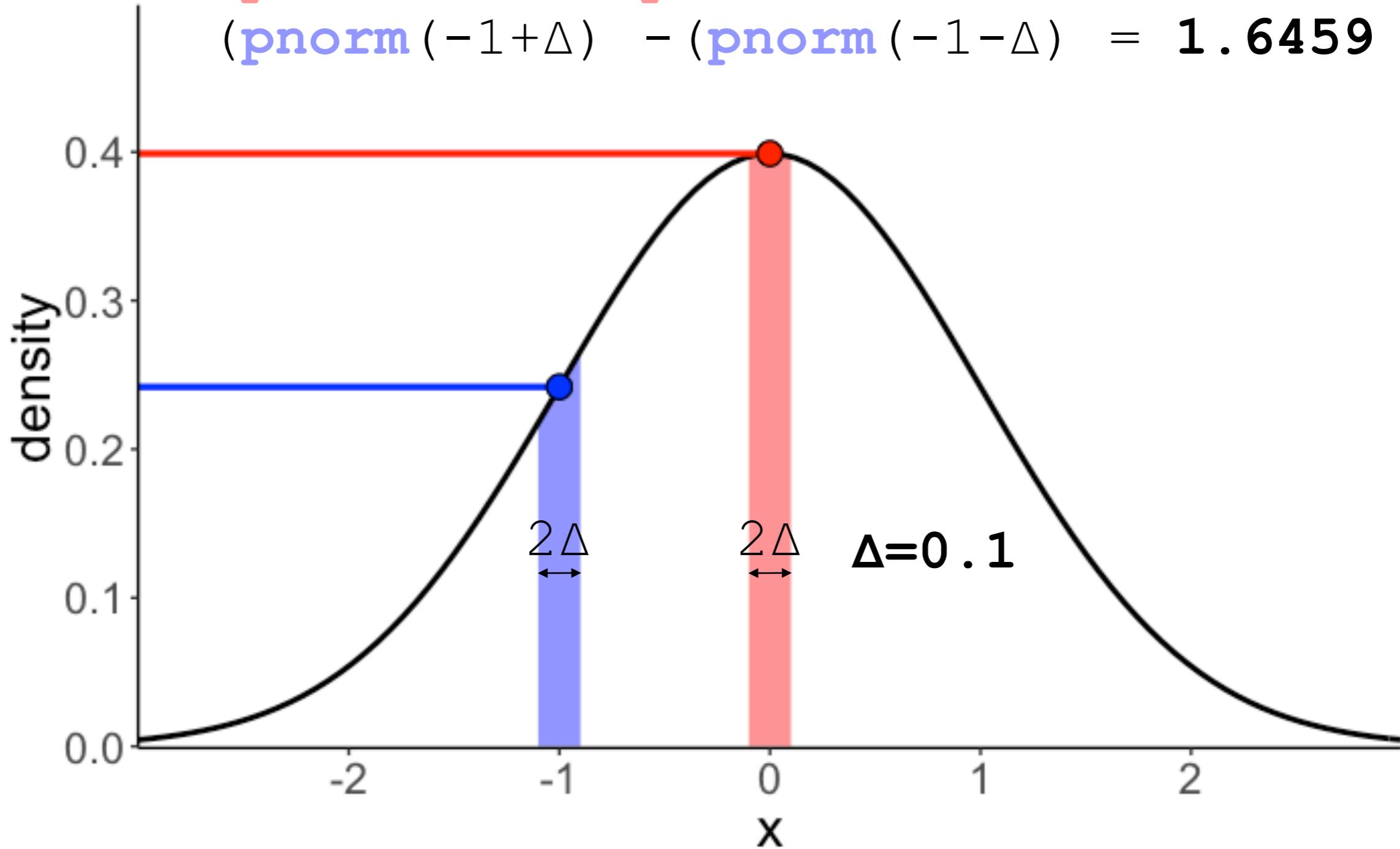


relative probability of one value vs. another

# Probability vs. likelihood

$$\text{dnorm}(0) / \text{dnorm}(-1) = 1.6487$$

$$\frac{(\text{pnorm}(0+\Delta) - \text{pnorm}(0-\Delta))}{(\text{pnorm}(-1+\Delta) - \text{pnorm}(-1-\Delta))} = 1.6459$$

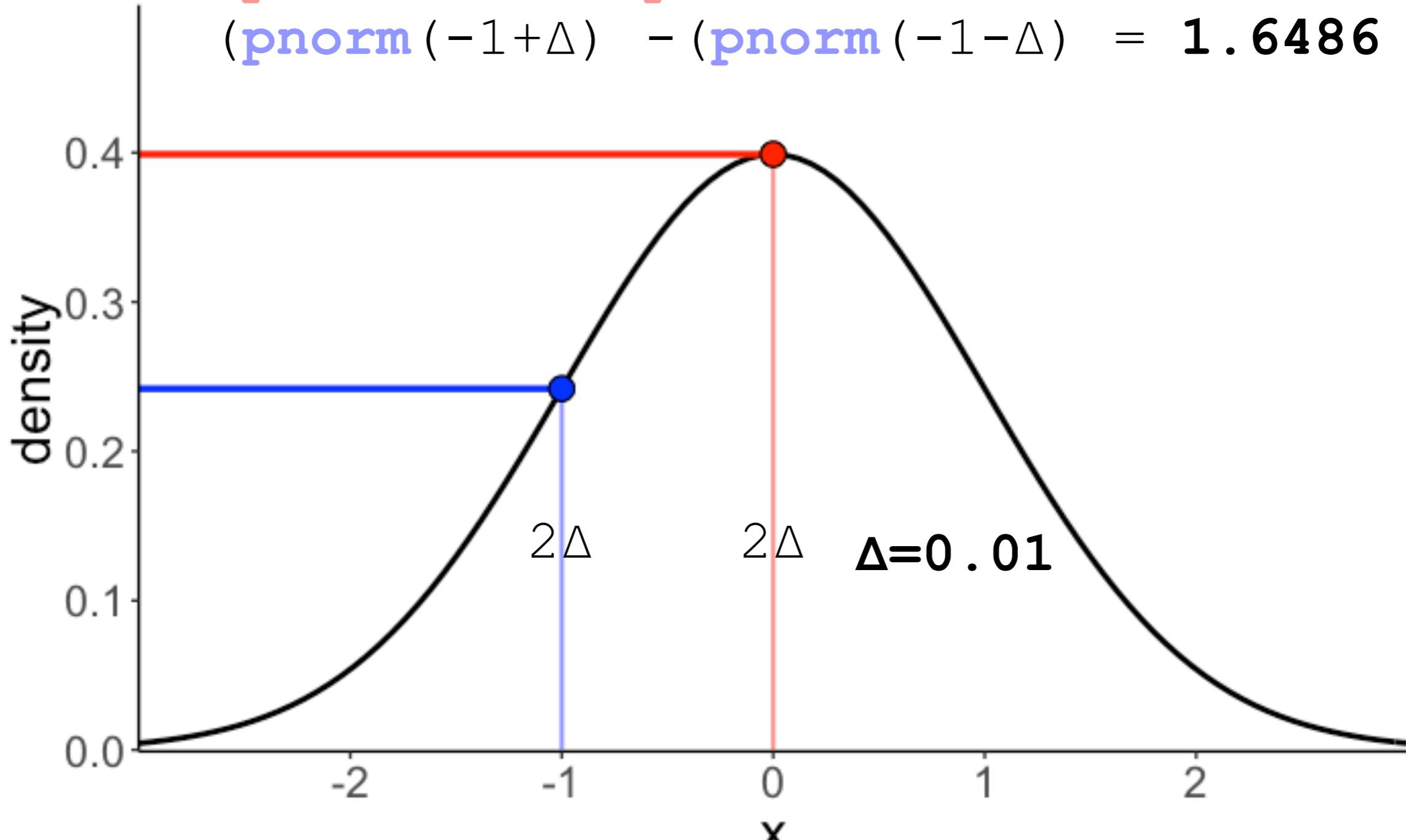


relative probability of one value vs. another

# Probability vs. likelihood

$$\text{dnorm}(0) / \text{dnorm}(-1) = 1.6487$$

$$\frac{(\text{pnorm}(0+\Delta) - \text{pnorm}(0-\Delta))}{(\text{pnorm}(-1+\Delta) - \text{pnorm}(-1-\Delta))} = 1.6486$$



relative probability of one value vs. another

# **Sampling solution**

# Summer camp: Via sampling

```
1 df.camp = tibble(  
2   kid = 1:1000,  
3   sport = sample(c("chess", "basketball"),  
4     size = 1000,  
5     replace = T,  
6     prob = c(1/3, 2/3))) %>%  
7   rowwise() %>%  
8   mutate(height = ifelse(test = sport == "chess",  
9     yes = rnorm(., mean = 170, sd = 8),  
10    no = rnorm(., mean = 180, sd = 10))) %>%  
11  ungroup())
```

| kid | sport      | height |
|-----|------------|--------|
| 1   | basketball | 164.84 |
| 2   | basketball | 163.22 |
| 3   | basketball | 191.18 |
| 4   | chess      | 160.16 |
| 5   | basketball | 182.99 |
| 6   | chess      | 163.54 |
| 7   | chess      | 168.56 |
| 8   | basketball | 192.99 |
| 9   | basketball | 171.91 |
| 10  | basketball | 177.12 |

```
1 df.camp %>%  
2   filter(height == 175) %>%  
3   count(sport)
```

doesn't work!

# Summer camp: Via sampling

```
1 df.camp = tibble(  
2   kid = 1:100000,  
3   sport = sample(c("chess", "basketball"),  
4     size = 100000,  
5     replace = T,  
6     prob = c(1/3, 2/3))) %>%  
7   rowwise() %>%  
8   mutate(height = ifelse(test = sport == "chess",  
9     yes = rnorm(., mean = 170, sd = 8),  
10    no = rnorm(., mean = 180, sd = 10))) %>%  
11 ungroup())
```

| kid | sport      | height |
|-----|------------|--------|
| 1   | basketball | 164.84 |
| 2   | basketball | 163.22 |
| 3   | basketball | 191.18 |
| 4   | chess      | 160.16 |
| 5   | basketball | 182.99 |
| 6   | chess      | 163.54 |
| 7   | chess      | 168.56 |
| 8   | basketball | 192.99 |
| 9   | basketball | 171.91 |
| 10  | basketball | 177.12 |

```
1 df.camp %>%  
2   filter(between(height,  
3     left = 174,  
4     right = 176)) %>%  
5   count(sport)
```

this works!

| sport      | n   |
|------------|-----|
| basketball | 469 |
| chess      | 273 |

$$\frac{\text{basketball}}{\text{basketball} + \text{chess}} \approx 0.63$$

# Plan for today

- Quick recap
- Simulating data
  - Drawing samples
  - Working with probability distributions
  - Quick detour: understanding density()
  - Asking probability distributions for answers
- Doing Bayesian inference
  - Analytic solution
  - Sampling solution

# **Feedback**

# How was the pace of today's class?

much      a little      just      a little      much  
too      too      right      too      too  
slow      slow

# 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!**