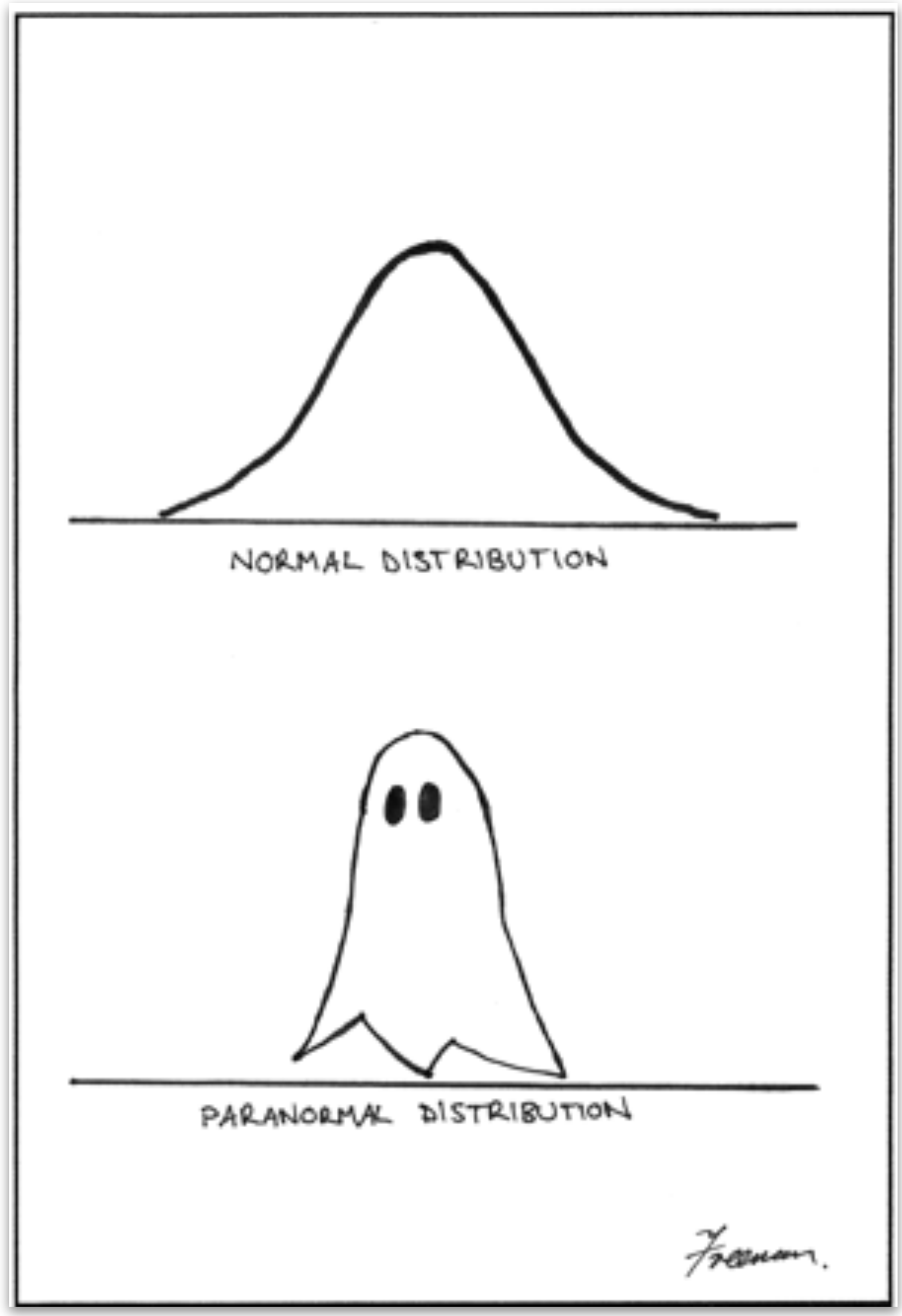
Sit in a different row this time!

Simulation 1



01/23/2019

1. How was your long weekend?
2. How annoying was homework 2?

Feedback

Your feedback

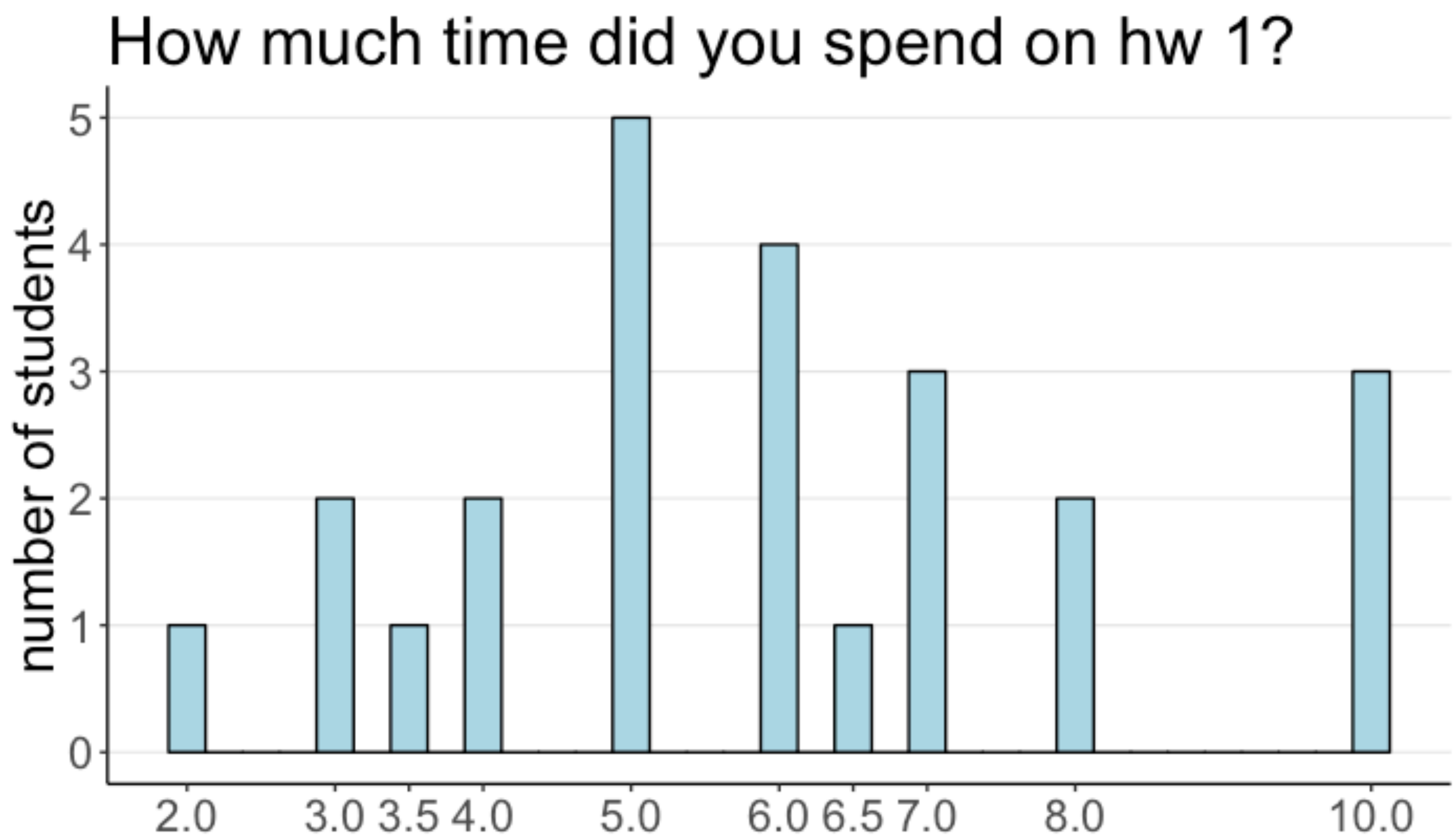
This is very trivial, but it would be helpful if the **slides are numbered** so we can jot down the number of specific slides to review later. Still wrapping my mind around Bayesian but it's a good start!

you got it

Logistics

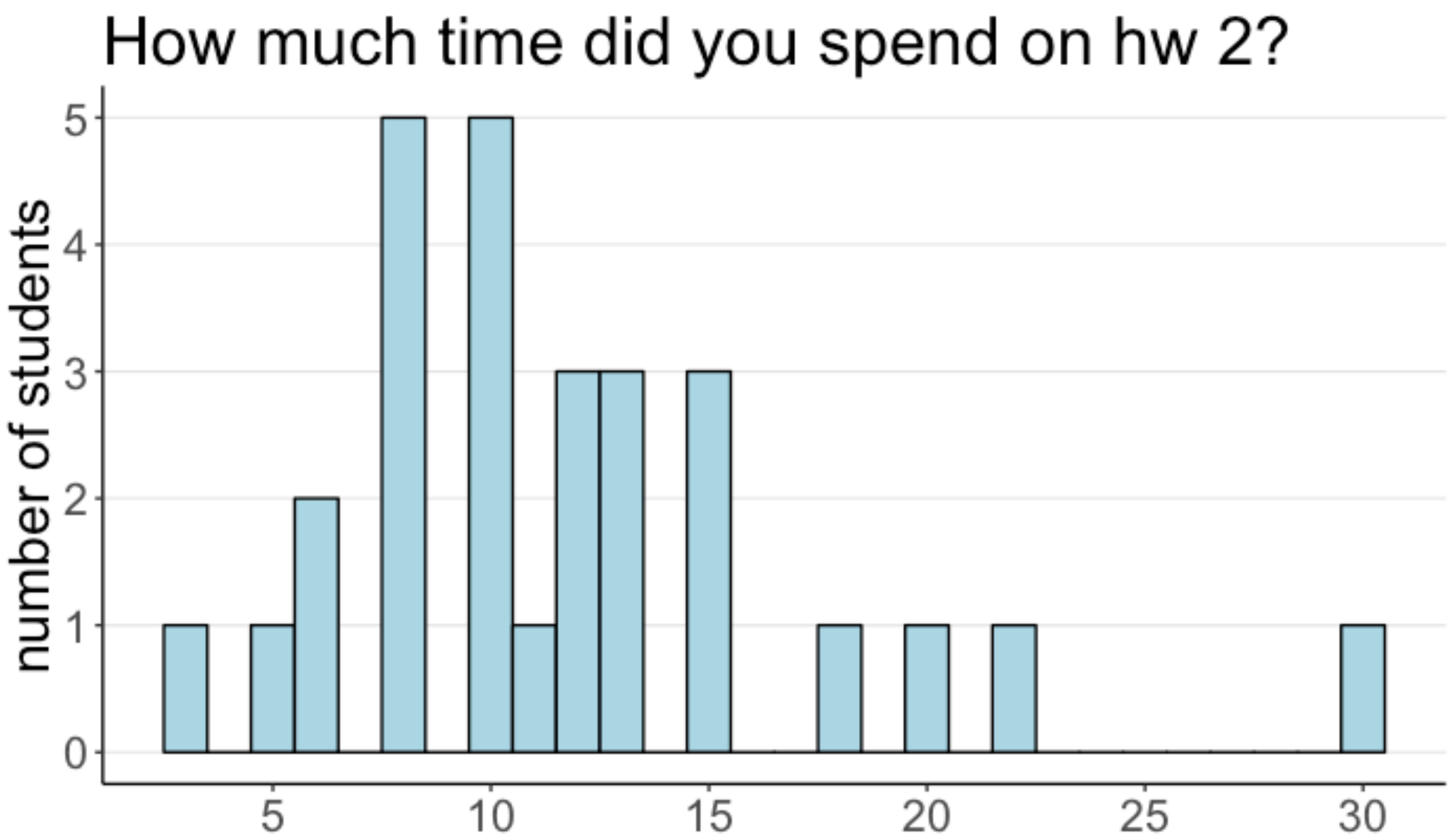
Homework

How many hours did it take you to complete Homework 1? (respond with a single number, e.g. write 5 if it took you five hours)



Students spent and average of 5.92 (SD = 2.21) on homework 1.

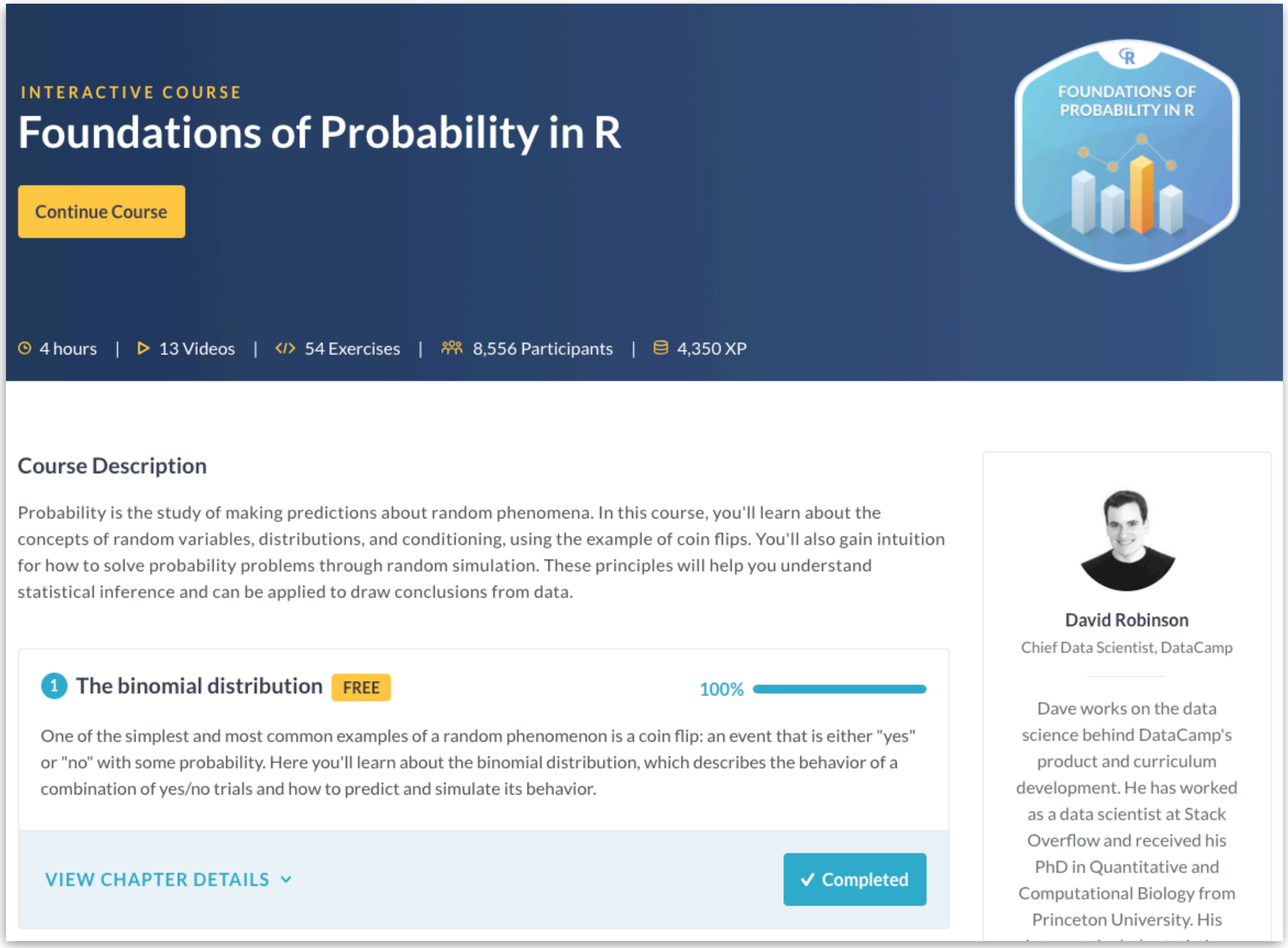
How many hours did it take you to complete Homework 2? (respond with a single number, e.g. write 5 if it took you five hours)



Students spent and average of 11.84 (SD = 5.62) on homework 2.

Homework 3

Will be released after class today and is due this **Tuesday (January 29th) at 8pm**.



Make sure to sign up for datacamp with your stanford.edu email address!

https://tinyurl.com/psych252datacamp

Plan for today

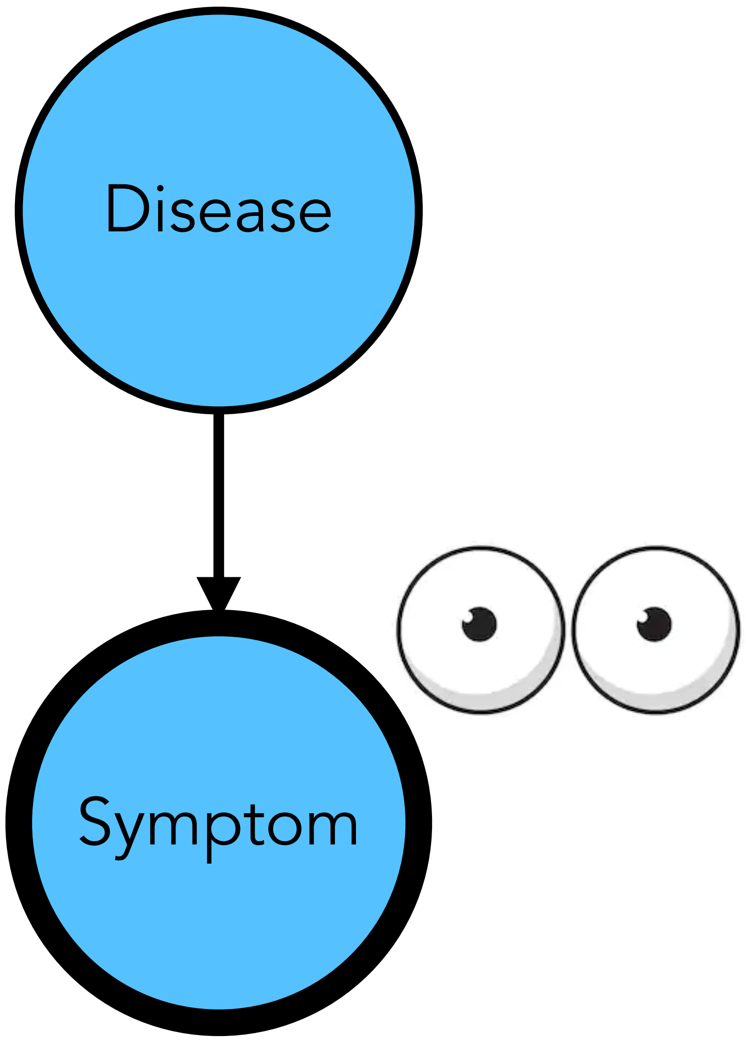
- Quick review of causality
- Working with probability distributions
  - `dnorm()`, `pnorm()`, `qnorm()`, `rnorm()`
  - computing probabilities
- Bayesian inference
  - analytic solution
  - via sampling
- Working with samples
  - Understanding `density()`
  - Understanding `quantile()`
  - Comparing distributions

Plan for today

- **Quick review of causality**
- Working with probability distributions
  - `dnorm()`,`pnorm()`,`qnorm()`,`rnorm()`
  - computing probabilities
- Bayesian inference
  - analytic solution
  - via sampling
- Working with samples
  - Understanding `density()`
  - Understanding `quantile()`
  - Comparing distributions

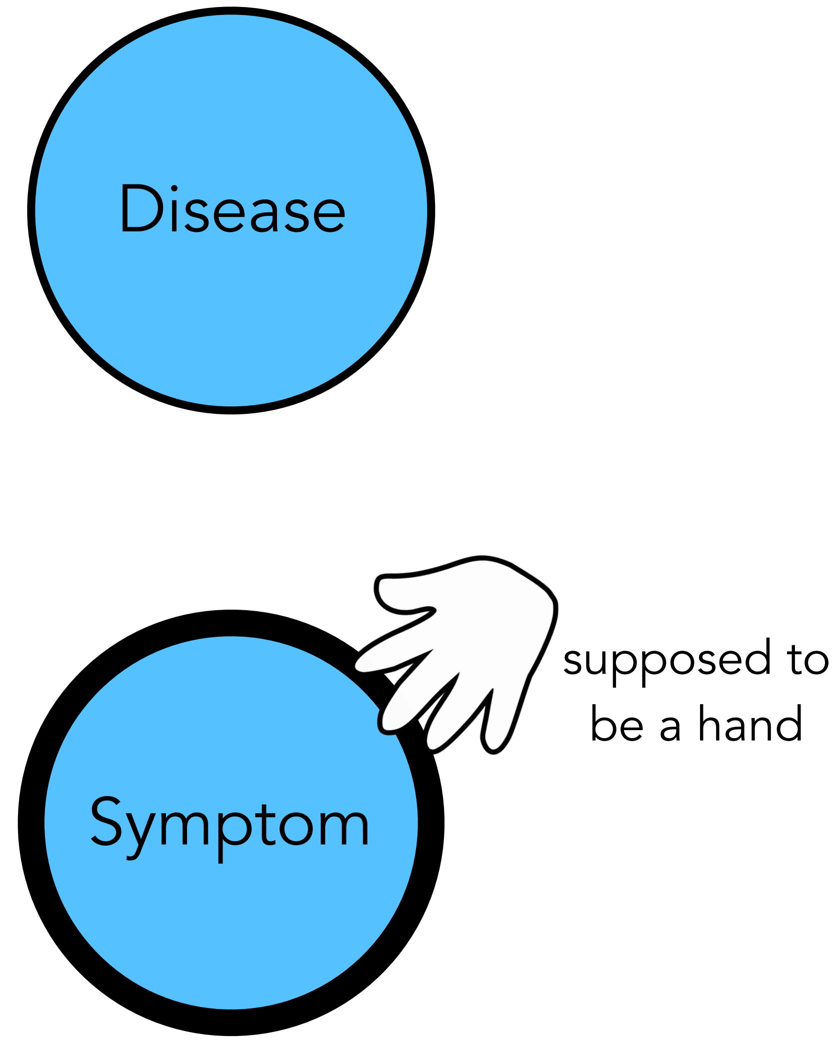
Observation vs. Intervention

seeing



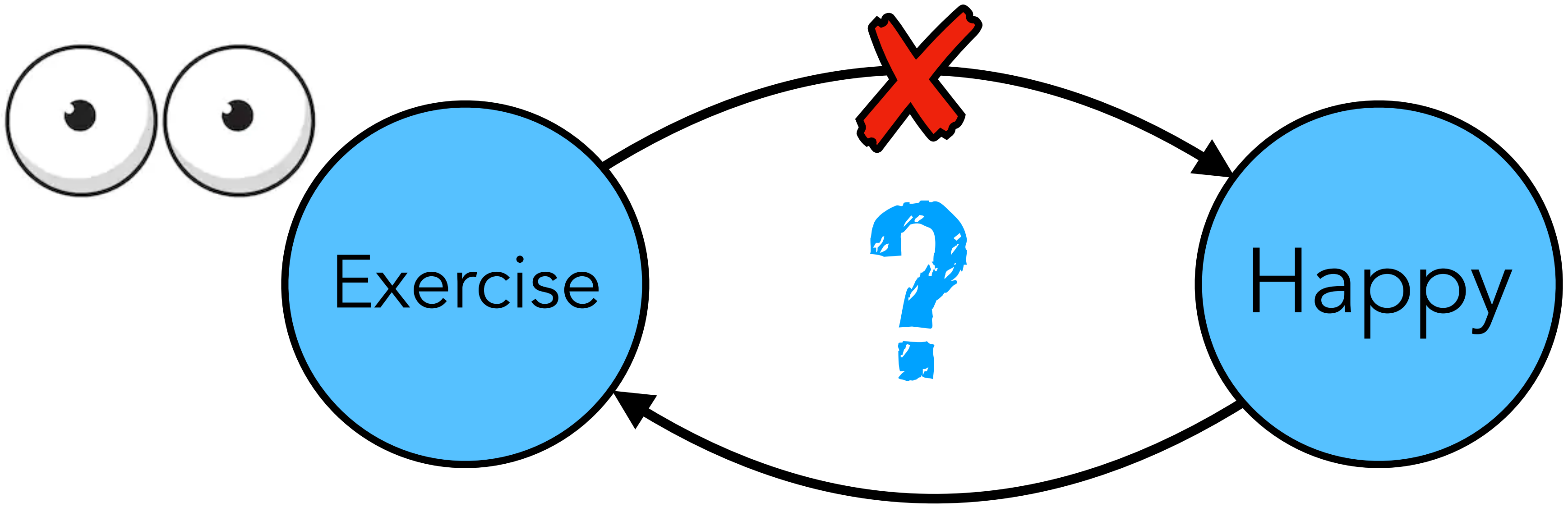
$$p(D \mid S) > p(D)$$

doing

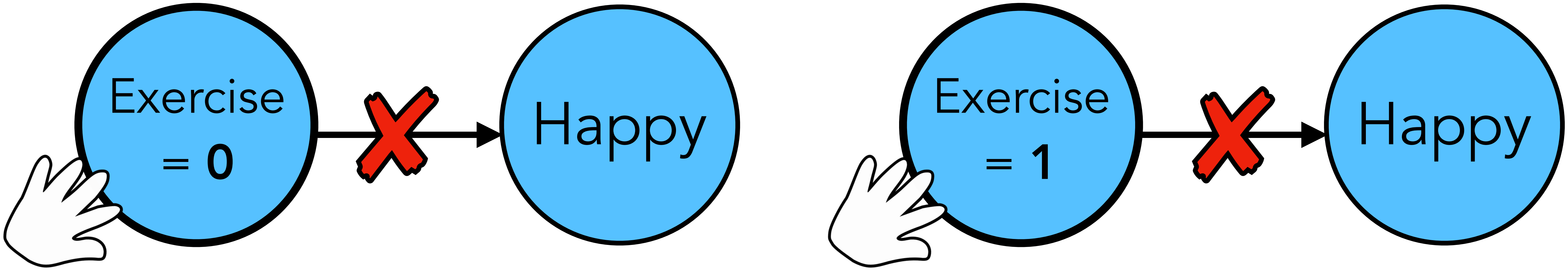


$$p(D \mid \mathrm{do}(S)) = p(D)$$

Inferring causal structure through intervention



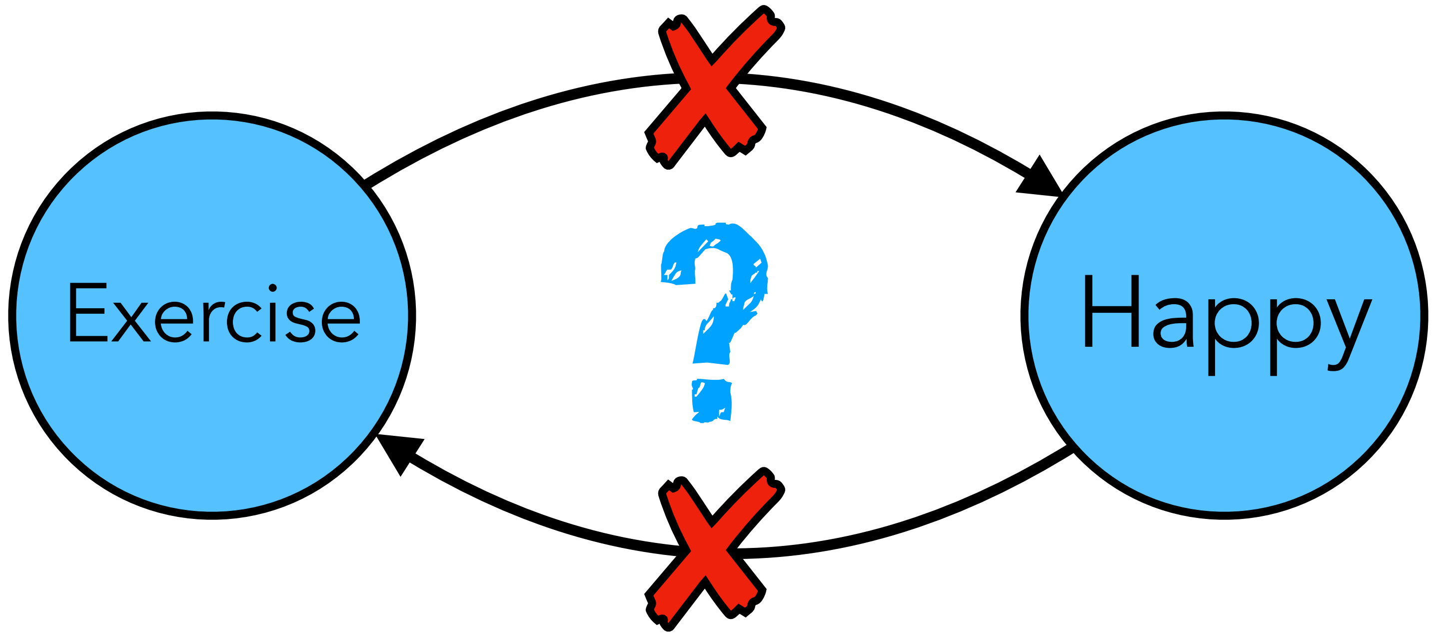
Experiment 1



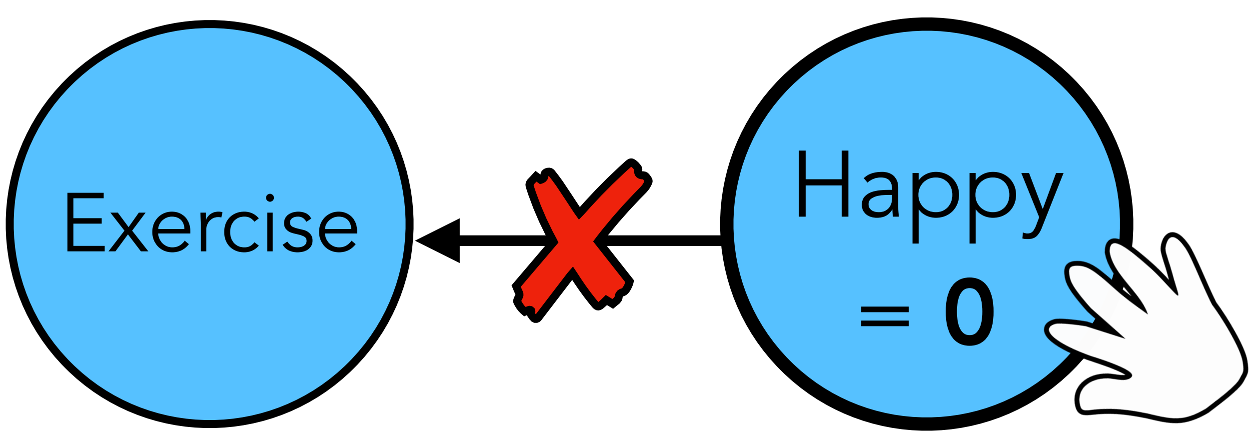
$p(\text{Happy} \mid \text{do}(\text{Exercise} = 0)) = 0.3$

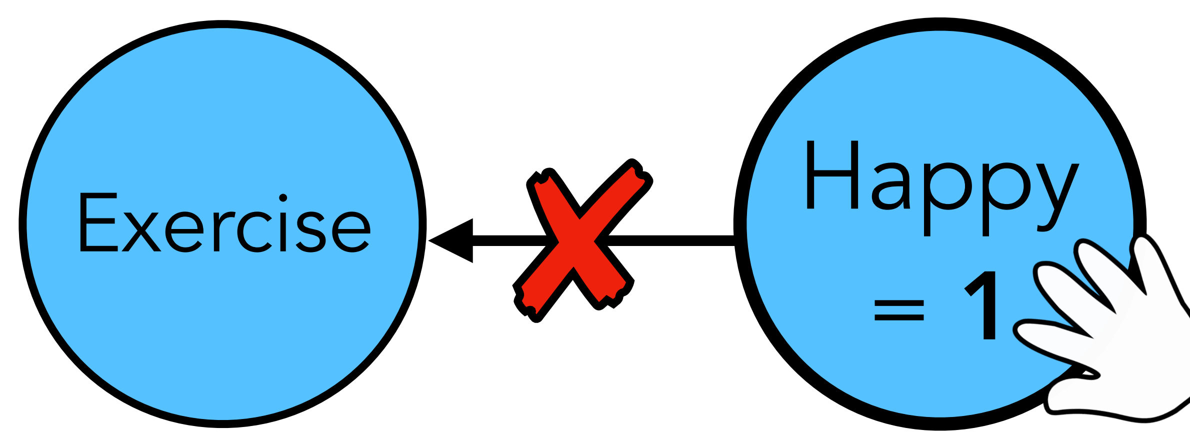
$p(\text{Happy} \mid \text{do}(\text{Exercise} = 1)) = 0.3$

Inferring causal structure through intervention



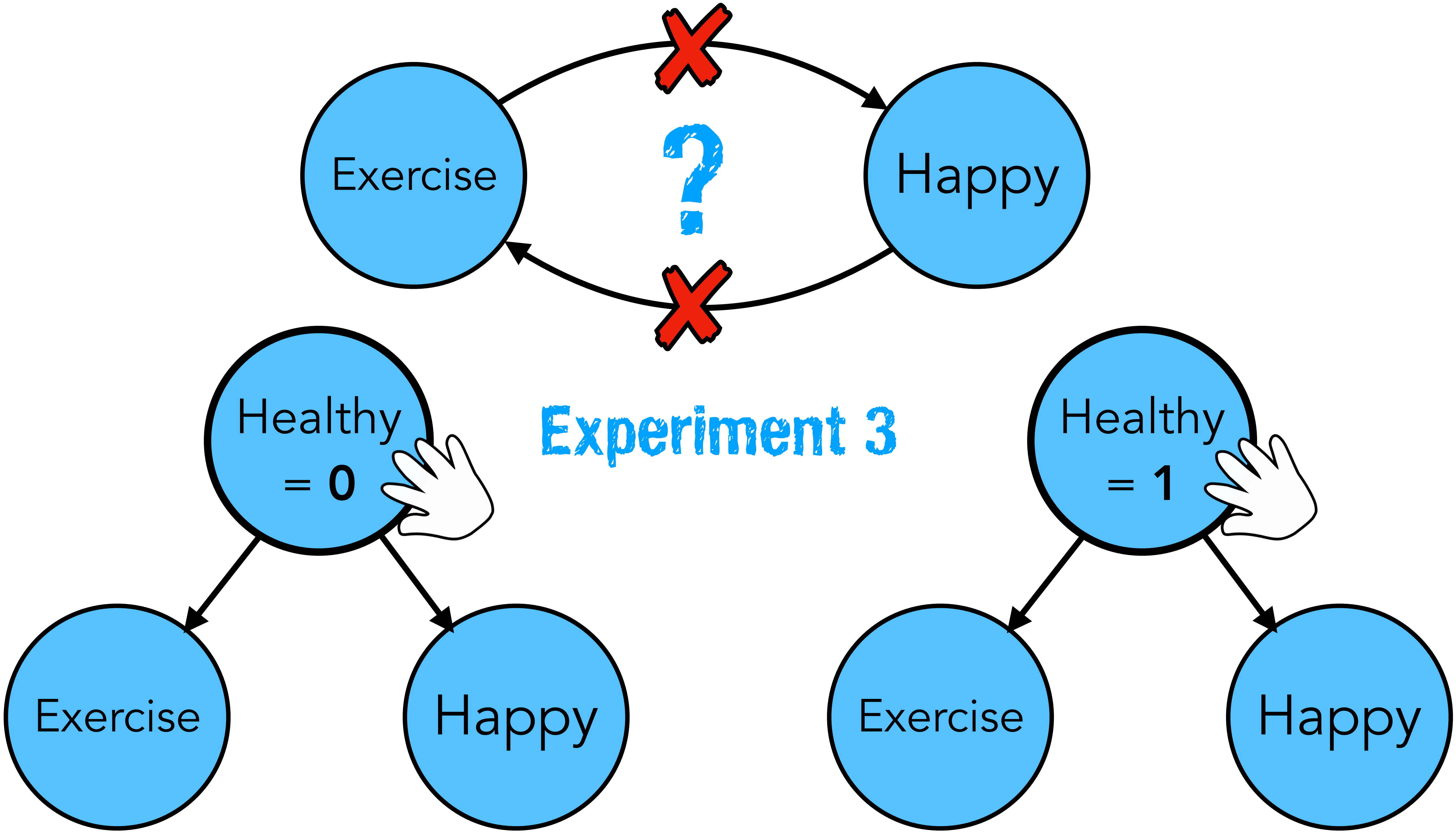
Experiment 2





$p(\text{Exercise} \mid \text{do}(\text{Happy} = 0)) = 0.1$ $\quad$ $p(\text{Exercise} \mid \text{do}(\text{Happy} = 1)) = 0.1$

Inferring causal structure through intervention

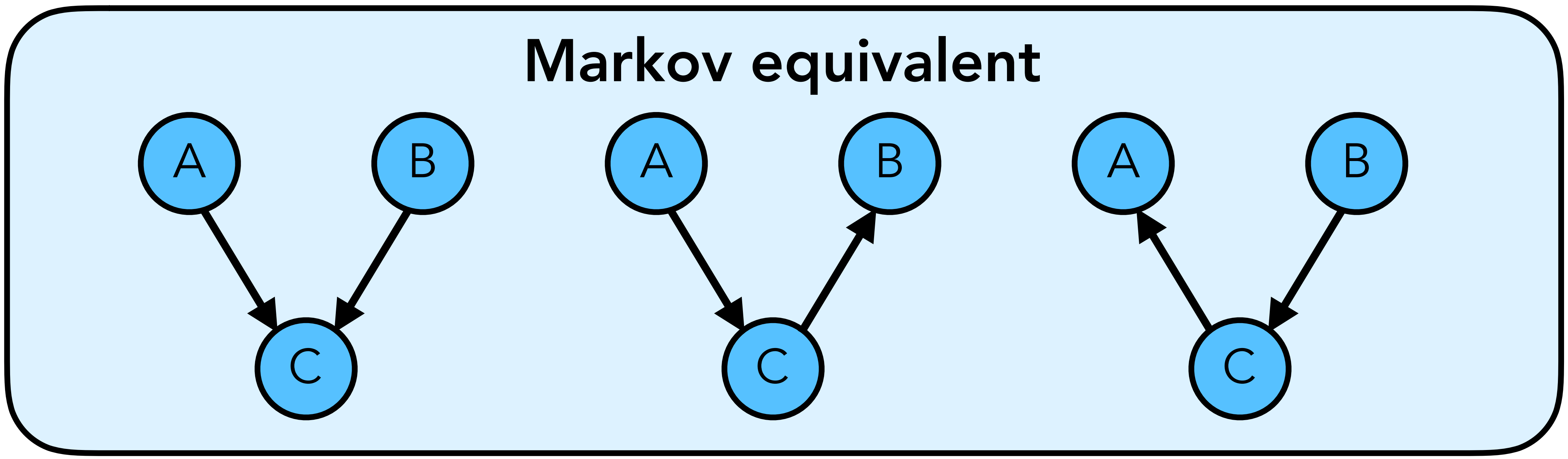


$p(\text{Happy} \,|\, \text{do}(\text{Healthy} = 0)) = 0.05$ $\qquad$ $p(\text{Happy} \,|\, \text{do}(\text{Healthy} = 0)) = 0.5$

$p(\text{Exercise} \,|\, \text{do}(\text{Healthy} = 0)) = 0.1$ $\qquad$ $p(\text{Exercise} \,|\, \text{do}(\text{Healthy} = 0)) = 0.75$

Important take home message

- correlation is not causation
- correlation (= probabilistic dependence) suggests that there is *some* causal relationship
- but we don't know which one it is



- **causal interventions** / experiments can reveal the underlying causal structure

The three layer causal hierarchy

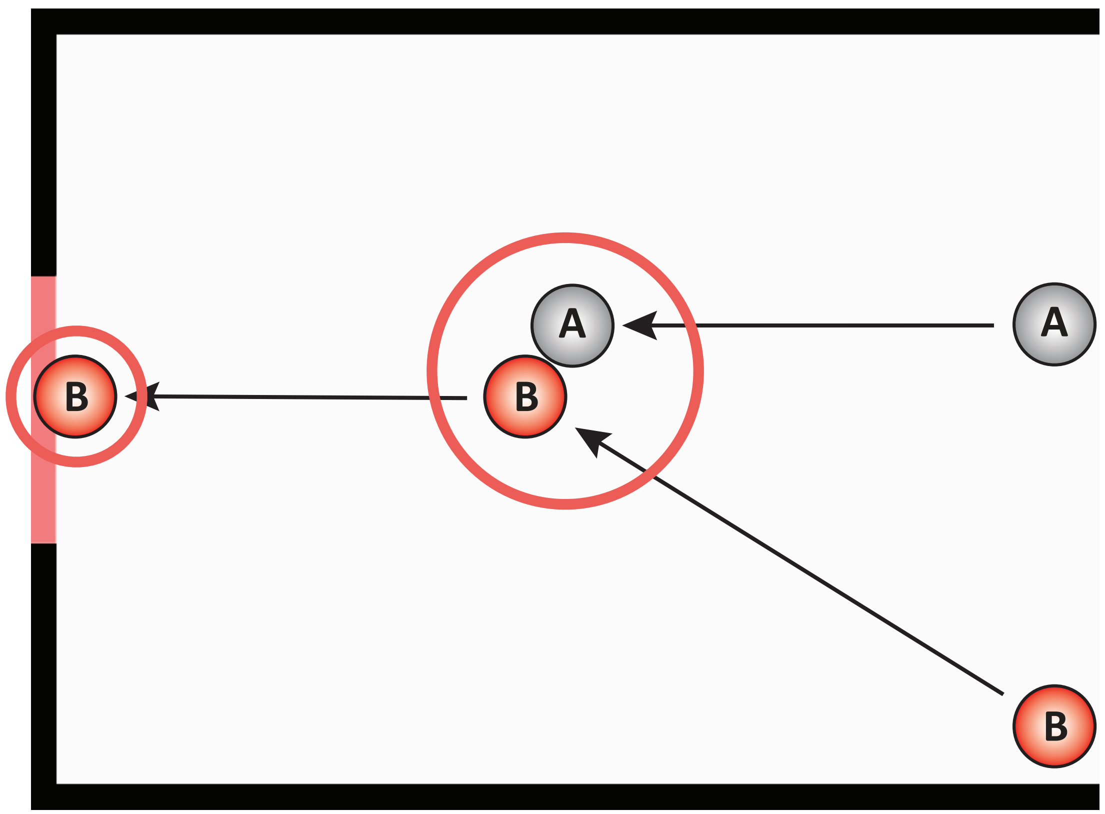
Level (Symbol)	Typical Activity	Typical Questions	Examples
1. Association $P(y\|x)$	Seeing	What is? How would seeing X change my belief in$Y$?	What does a symptom tell me about a disease? What does a survey tell us about the election results?
2. Intervention $P(y\|do(x), z)$	Doing Intervening	What if? What if I do X?	What if I take aspirin, will my headache be cured? What if we ban cigarettes?
3. Counterfactuals $P(y_x\|x', y')$	Imagining, Retrospection	Why? Was it X that caused Y? What if I had acted differently?	Was it the aspirin that stopped my headache? Would Kennedy be alive had Oswald not shot him? What if I had not been smoking the past 2 years?

Pearl (2018) Theoretical Impediments to Machine Learning With Seven Sparks from the Causal Revolution. *Proceedings of the Eleventh International Conference on Web Search and Data Mining*

Did A cause B to go through the gate?

Counterfactual Simulation Model

What happened?

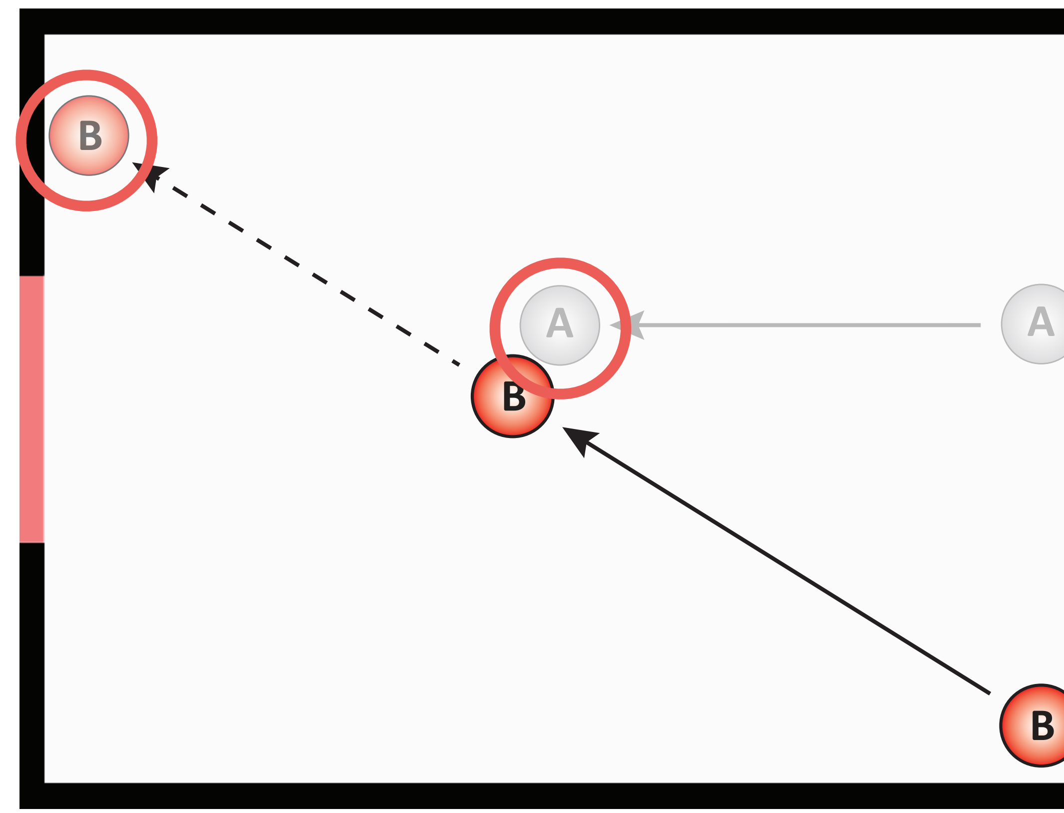


Actual situation

B went through the gate

≠

What would have happened?



Counterfactual situation

B would have missed the gate

Gerstenberg, Goodman, Lagnado, & Tenenbaum (2012) Noisy Newtons: Unifying process and dependency accounts of causal attribution. Cognitive Science Proceedings

Gerstenberg, Goodman, Lagnado, & Tenenbaum (2014) From counterfactual simulation to causal judgment. Cognitive Science Proceedings

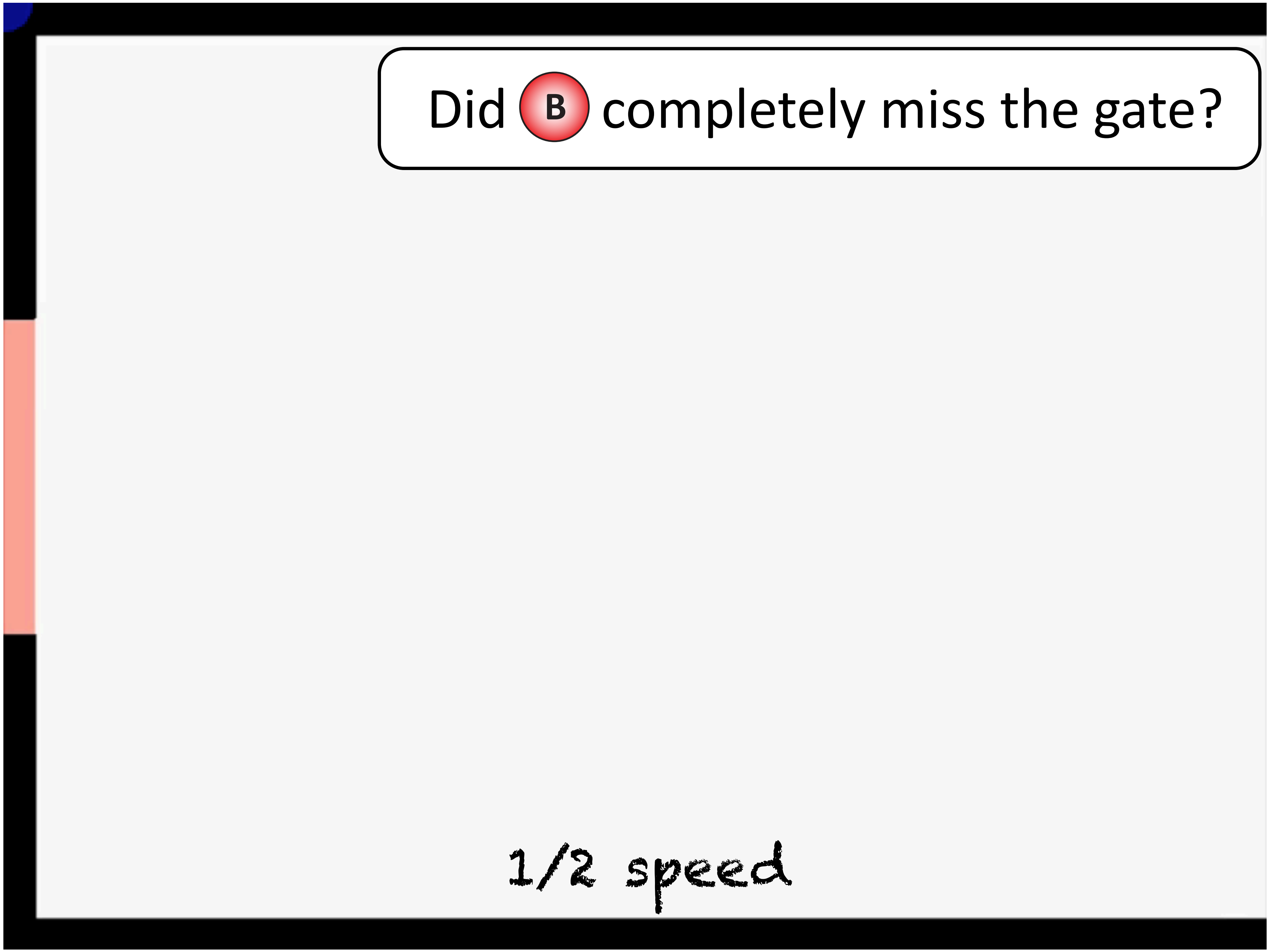
Gerstenberg, Goodman, Lagnado, & Tenenbaum (2015) How, whether, why: Causal judgments as counterfactual contrasts. Cognitive Science Proceedings

Gerstenberg & Tenenbaum (2016) Understanding ``almost'': Empirical and computational studies of near misses. Cognitive Science Proceedings

Gerstenberg & Tenenbaum (2017) Intuitive Theories. Oxford Handbook of Causal Reasoning

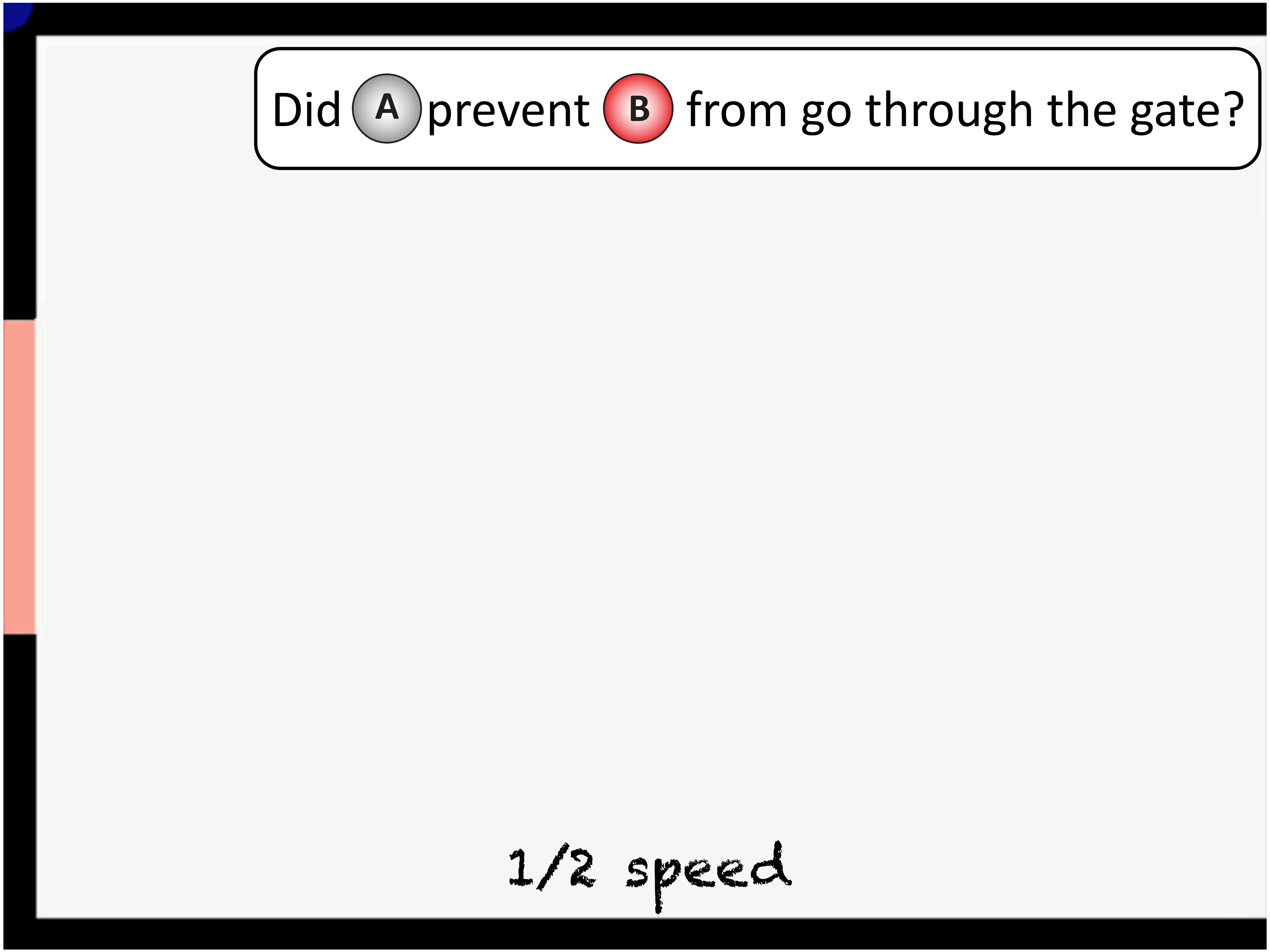
Gerstenberg, Goodman, Lagnado, & Tenenbaum (in preparation) A counterfactual simulation model of causal judgment.

Spontaneous counterfactual simulation



Gerstenberg, Peterson, Goodman, Lagnado, & Tenenbaum (2017) Eye-tracking causality. *Psychological Science*

Spontaneous counterfactual simulation



Gerstenberg, Peterson, Goodman, Lagnado, & Tenenbaum (2017) Eye-tracking causality. *Psychological Science*

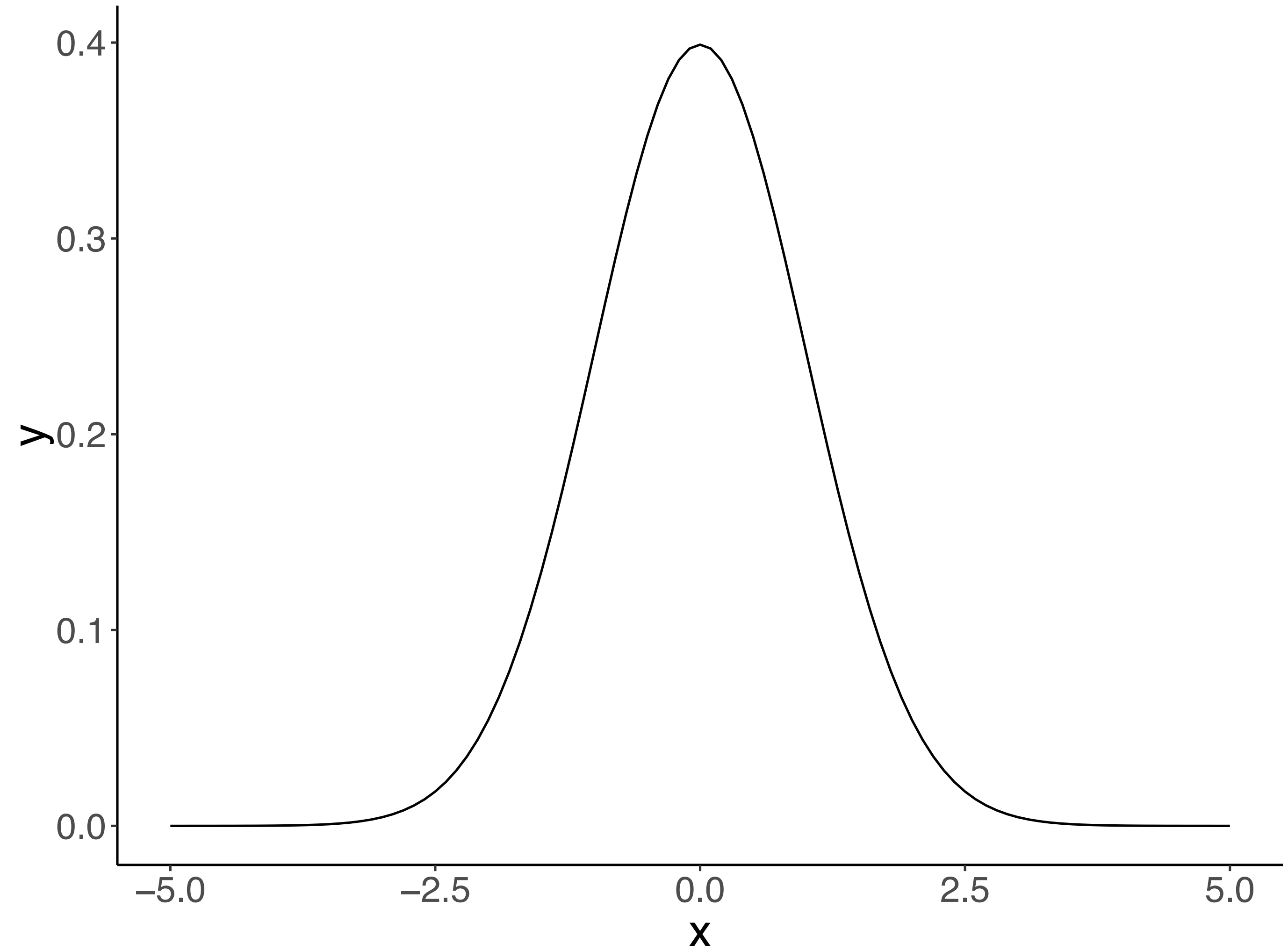
Plan for today

- Quick review of causality
- **Working with probability distributions**
  - `dnorm()`, `pnorm()`, `qnorm()`, `rnorm()`
  - computing probabilities
- Bayesian inference
  - analytic solution
  - via sampling
- Working with samples
  - Understanding `density()`
  - Understanding `quantile()`
  - Comparing distributions

Working with probability distributions

letter	description	example
`d`	for "density", the density function (probability mass function (for *discrete* variables) or probability density function (for *continuous* variables))	`dnorm()`
`p`	for "probability", the cumulative distribution function	`pnorm()`
`q`	for "quantile", the inverse cumulative distribution function	`qnorm()`
`r`	for "random", a random variable having the specified distribution	`rnorm()`

Normal distribution



mu sigma

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

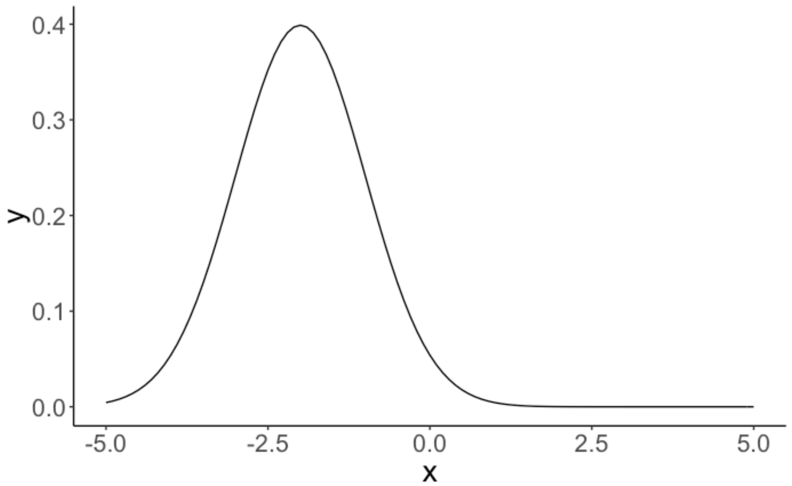
```
dnorm(x, mean = 0, sd = 1)
```

d = density **norm** = normal distribution

we use **Greek** letters to refer to parameters of the population (or the theoretical distributions)

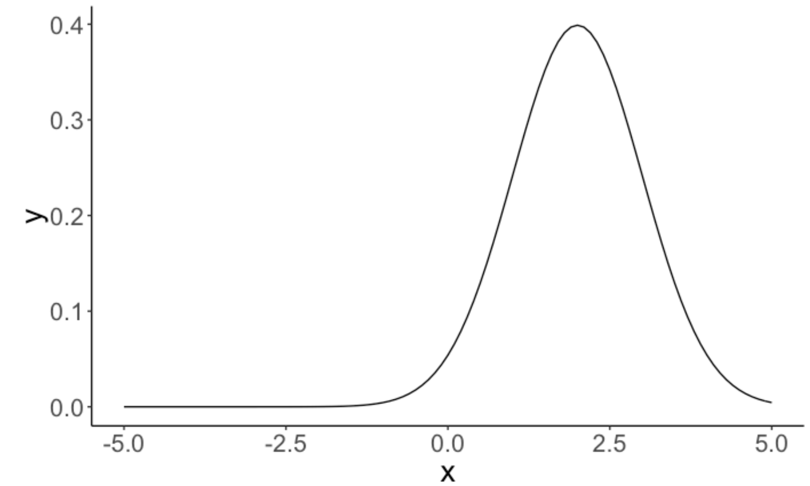
and **Roman** letters to refer to parameters in our sample from the population

Normal distribution

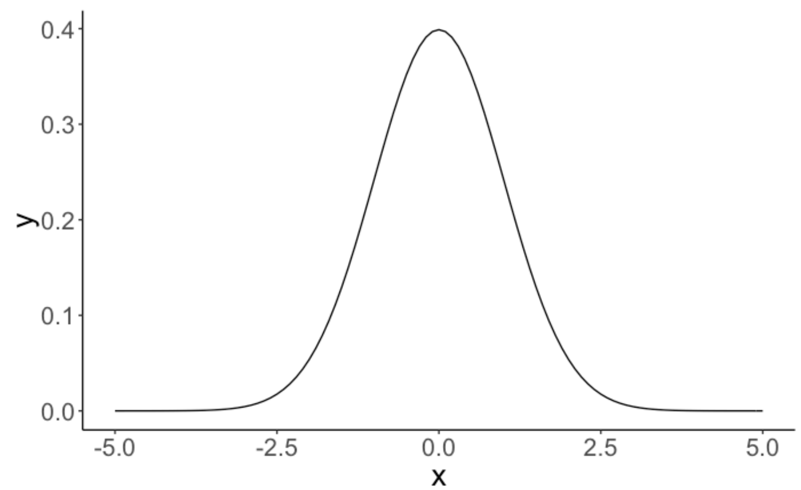


dnorm(x, mean = **-2,** sd = 1)

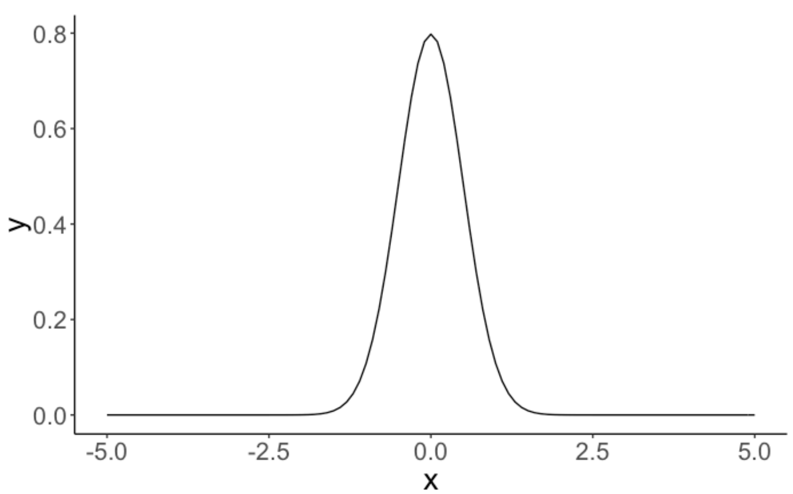
manipulating `mean`



dnorm(x, mean = **2,** sd = 1)

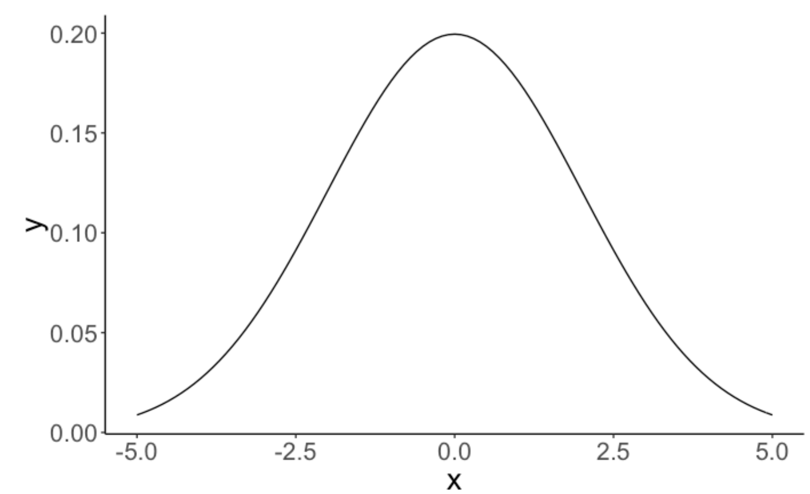


dnorm(x, mean = **0,** sd = 1)



dnorm(x, mean = 0, sd = **0.5**)

manipulating `sd`



dnorm(x, mean = 0, sd = **2**)

Plotting distributions

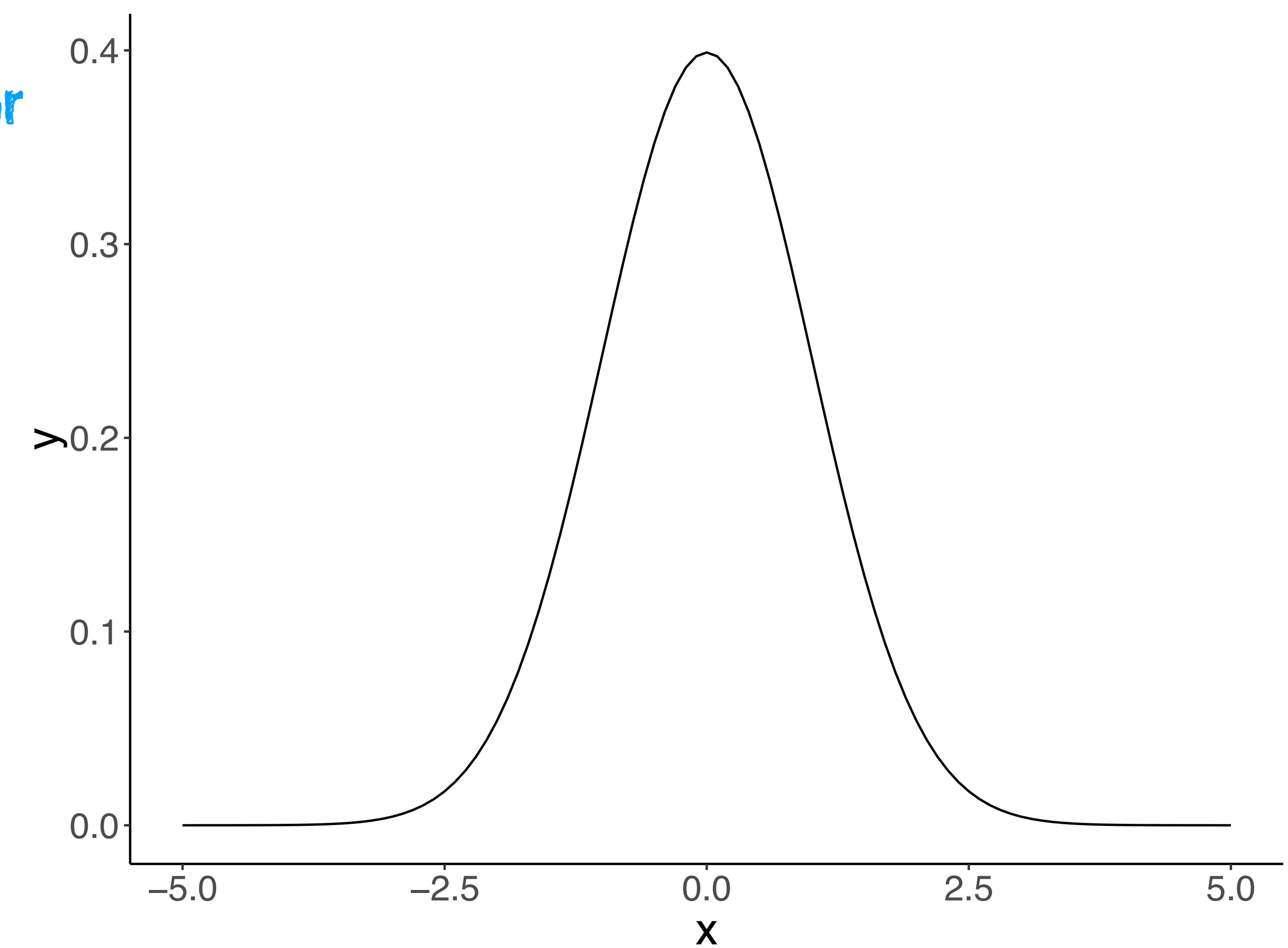
```r
ggplot(data = tibble(x = c(-5, 5)),
       mapping = aes(x = x)) +
  stat_function(fun = "dnorm",
                args = list(mean = 0,
                            sd = 1))
```

make data frame with minimum and maximum x-value

what function should be plotted?

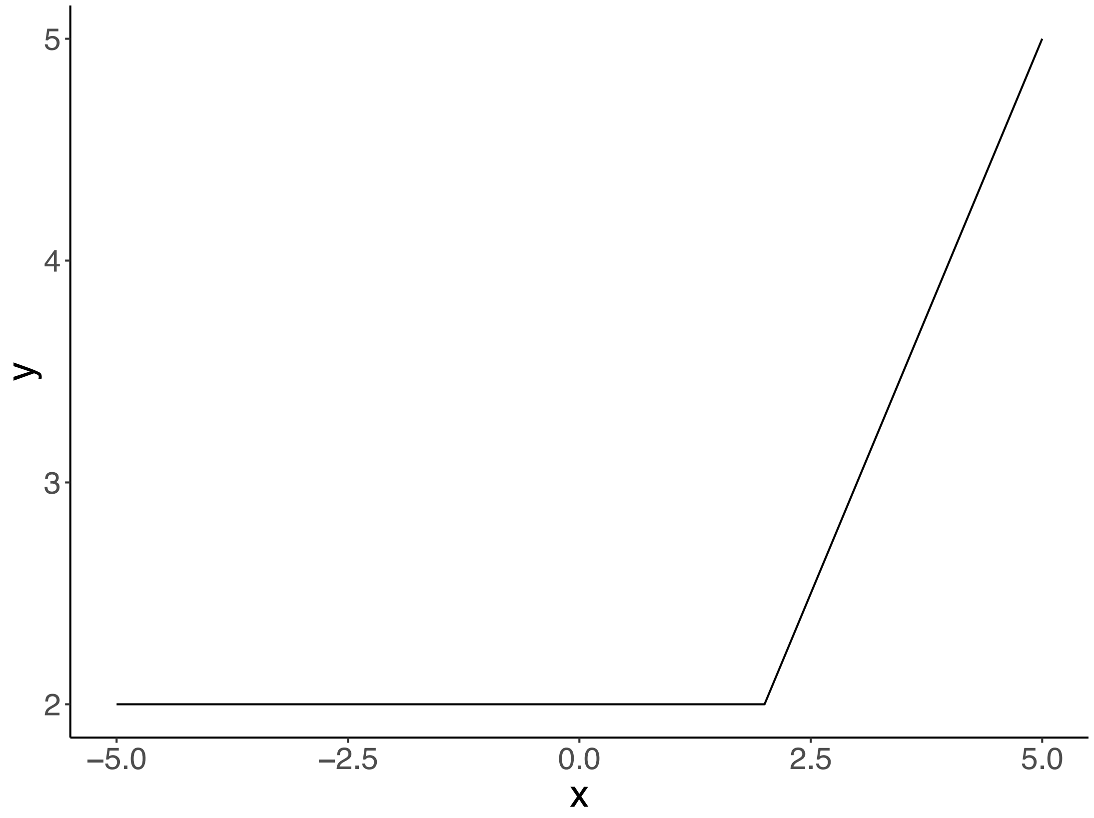
function for plotting functions

any parameters for the function?



Plotting functions

```
# define the breakpoint function
fun.breakpoint = function(x, breakpoint){
  x[x < breakpoint] = breakpoint
  return(x)
}

# plot the function
ggplot(data = tibble(x = c(-5, 5)),
       mapping = aes(x = x)) +
  stat_function(fun = "fun.breakpoint",
                args = list(breakpoint = 2)
                )
```



Sampling from distributions

n = number of samples

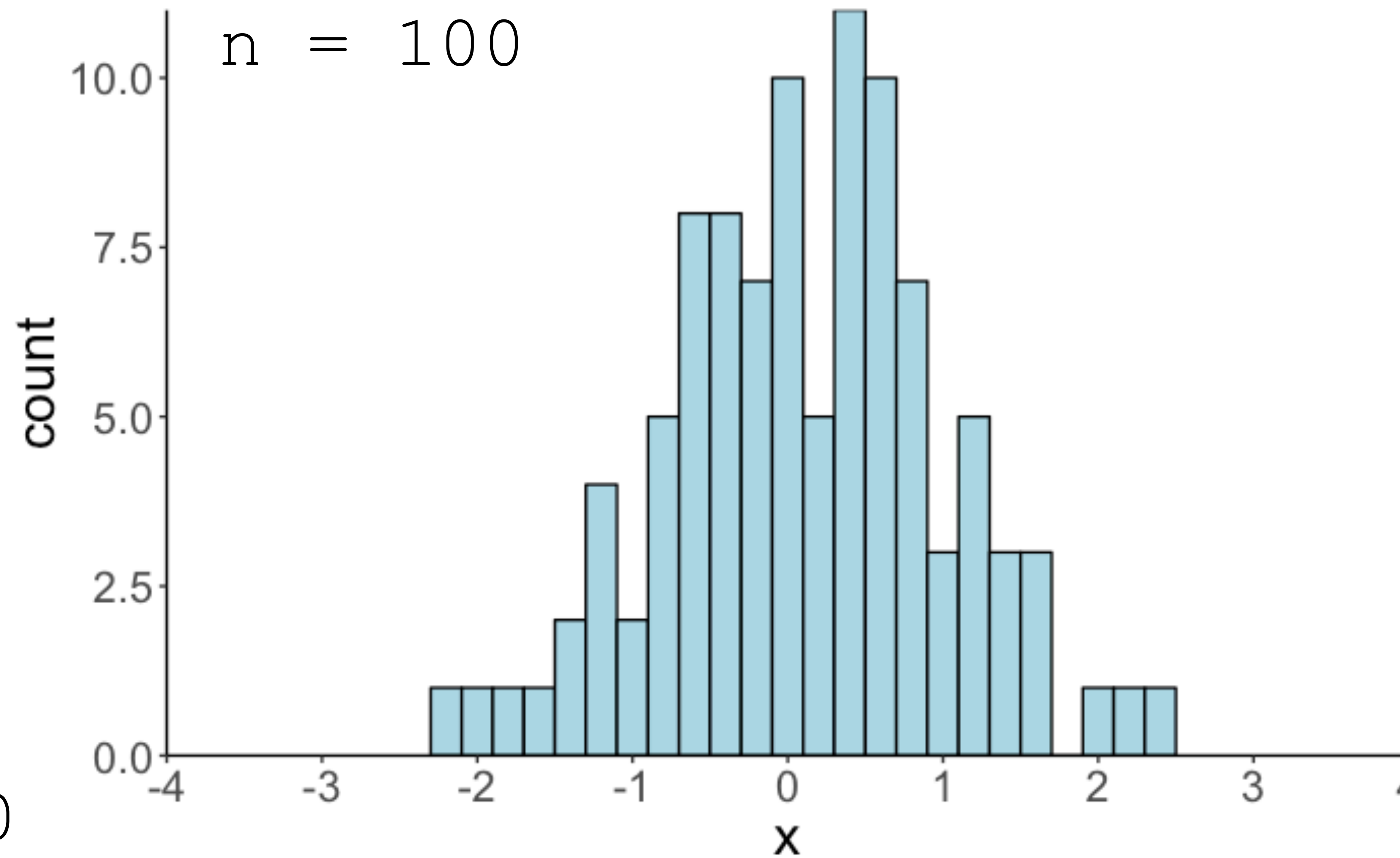
```
rnorm(n, mean = 0, sd = 1)
```

r = random samples　**norm** = normal distribution

x
-0.63
0.18
-0.84
1.60
0.33
-0.82
0.49
0.74

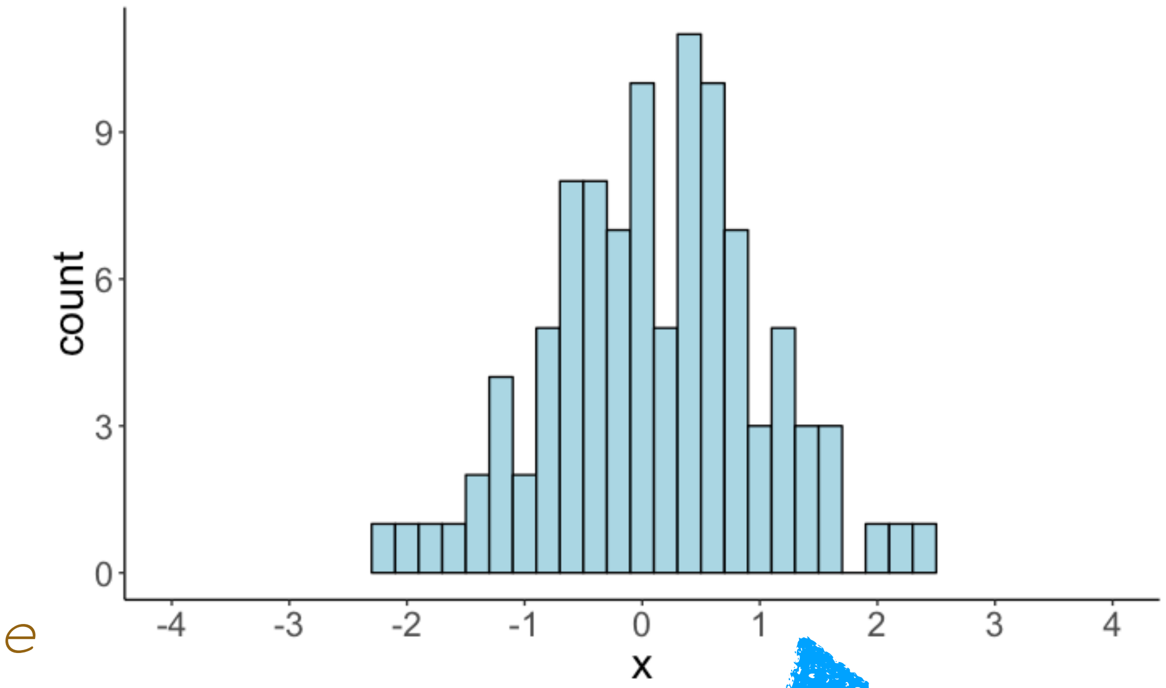
⋮

`nrow = 100`



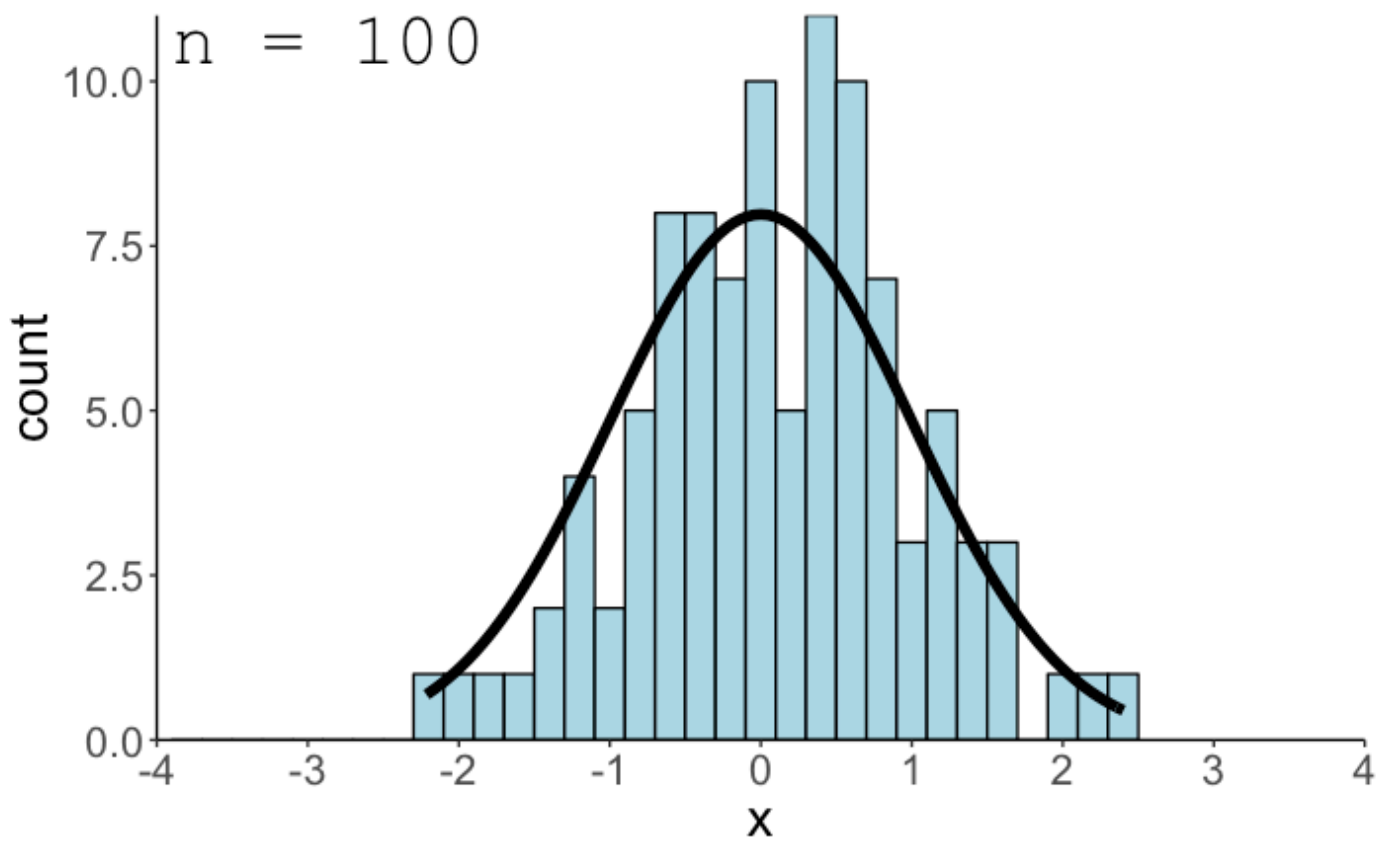
Sampling from distributions

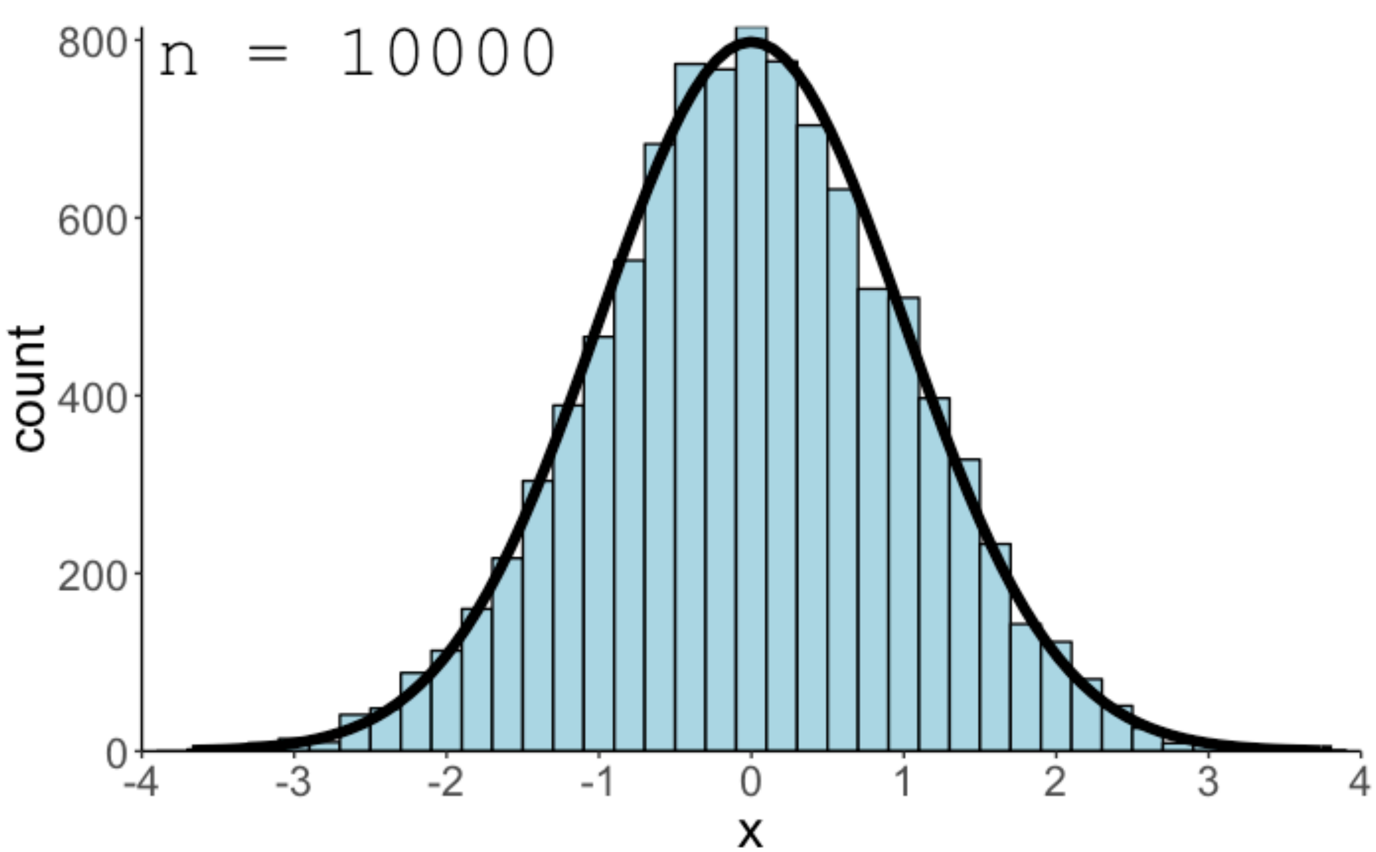
```r
# make this example reproducible
set.seed(1)

# define how many samples to draw
nsamples = 100

# make a data frame with the sample
df.plot = tibble(
  x = rnorm(n = nsamples, mean = 0, sd = 1)
)

# plot the samples using a histogram
ggplot(data = df.plot,
       mapping = aes(x = x)) +
  geom_histogram(binwidth = 0.2,
                 color = "black",
                 fill = "lightblue") +
  scale_x_continuous(breaks = -4:4, labels = -4:4) +
  coord_cartesian(xlim = c(-4, 4), expand = F)
```



expand = T

Sampling from distributions





law of large numbers

approximation to true underlying distribution improves with increased sample size

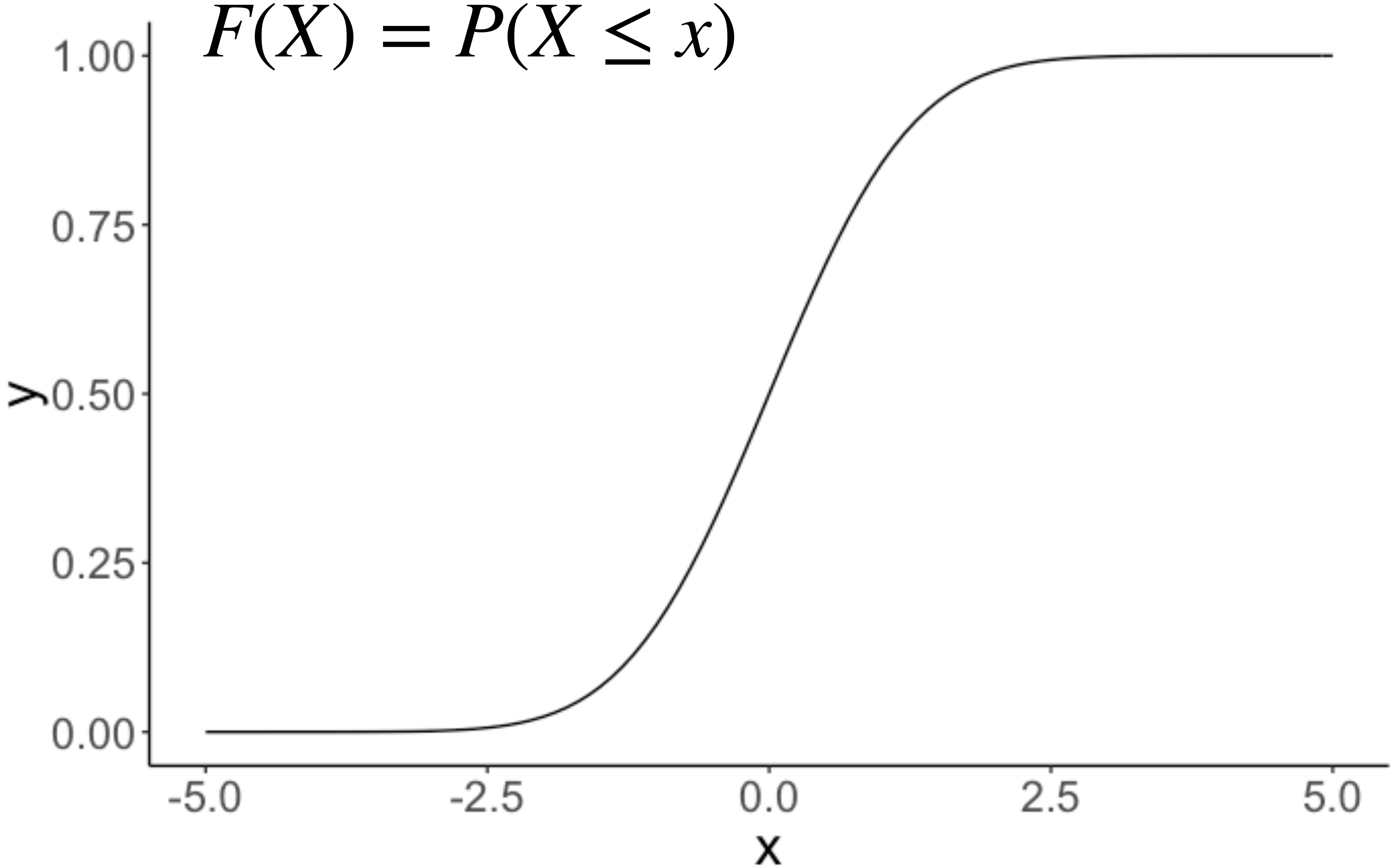
Cumulative probability distribution

```r
ggplot(data = tibble(x = c(-5, 5)),
       mapping = aes(x = x)) +
  stat_function(fun = "pnorm",
                args = list(mean = 0,
                            sd = 1))
```

p = probability

cumulative distribution function

$$F(X) = P(X \leq x)$$

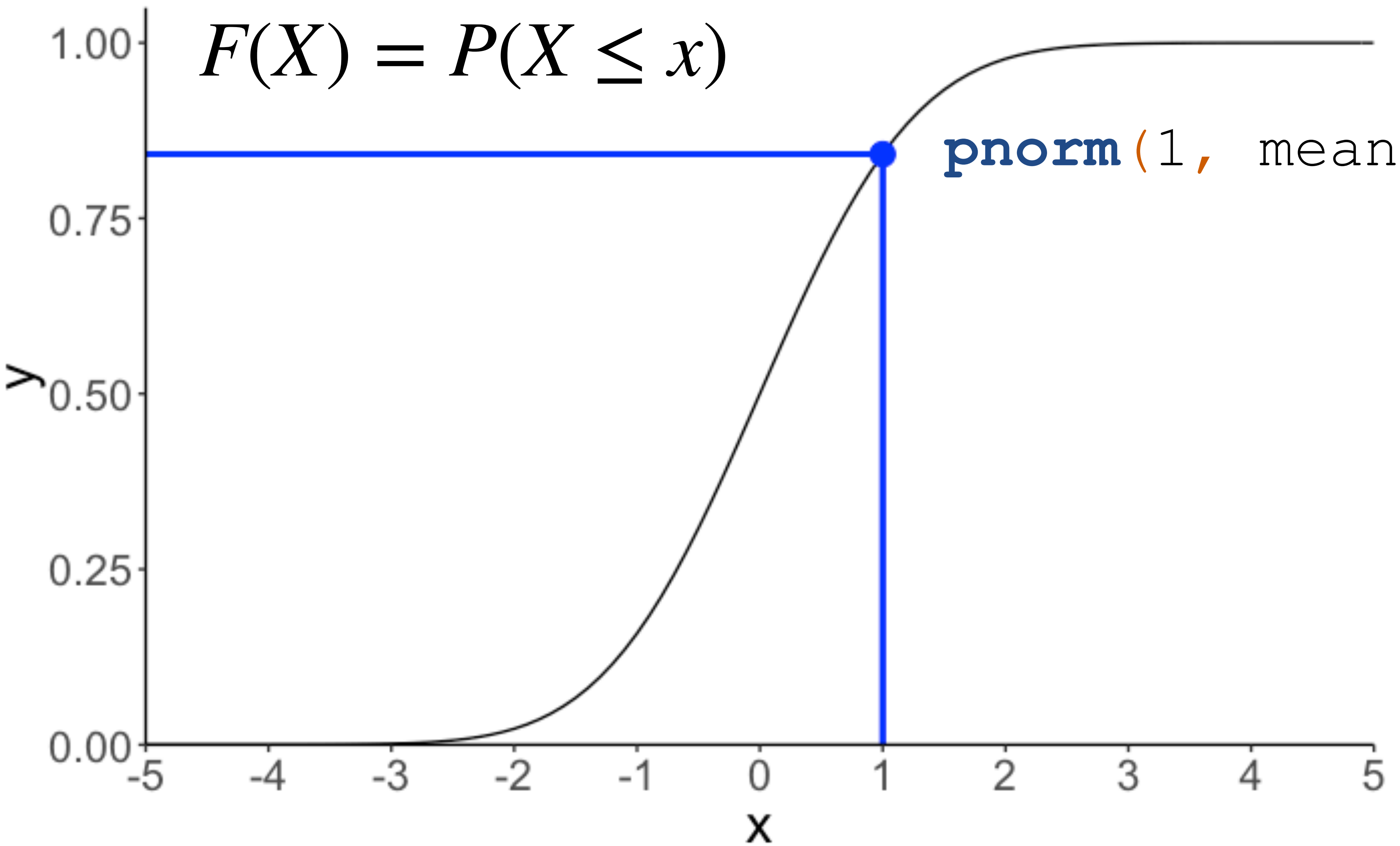


Cumulative probability distribution

```
ggplot(data = tibble(x = c(-5, 5)),
       mapping = aes(x = x)) +
  stat_function(fun = "pnorm",
                args = list(mean = 0,
                            sd = 1))
```

p = probability

cumulative distribution function



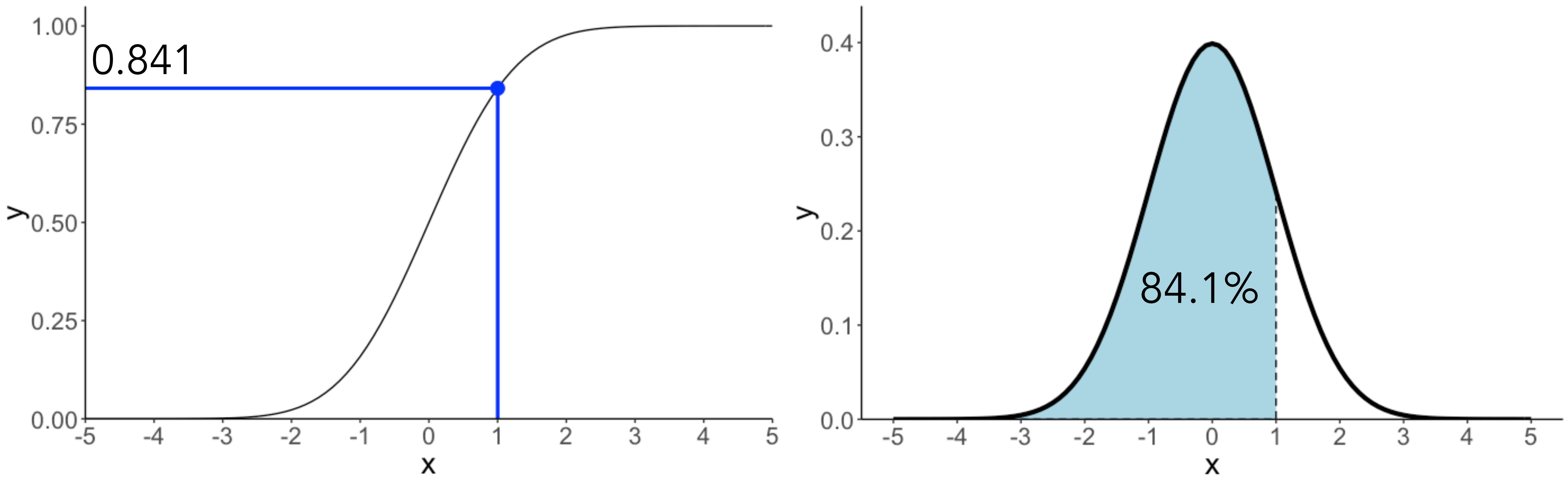
$$F(X) = P(X \leq x)$$

`pnorm(1, mean = 0, sd = 1)`

= 0.841

Computing probabilities

```
pnorm(1, mean = 0, sd = 1)
```



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

Inverse cumulative distribution function

```r
ggplot(data = tibble(x = c(0, 1)),
       mapping = aes(x = x)) +
  stat_function(fun = "qnorm",
                args = list(mean = 0,
                            sd = 1))
```

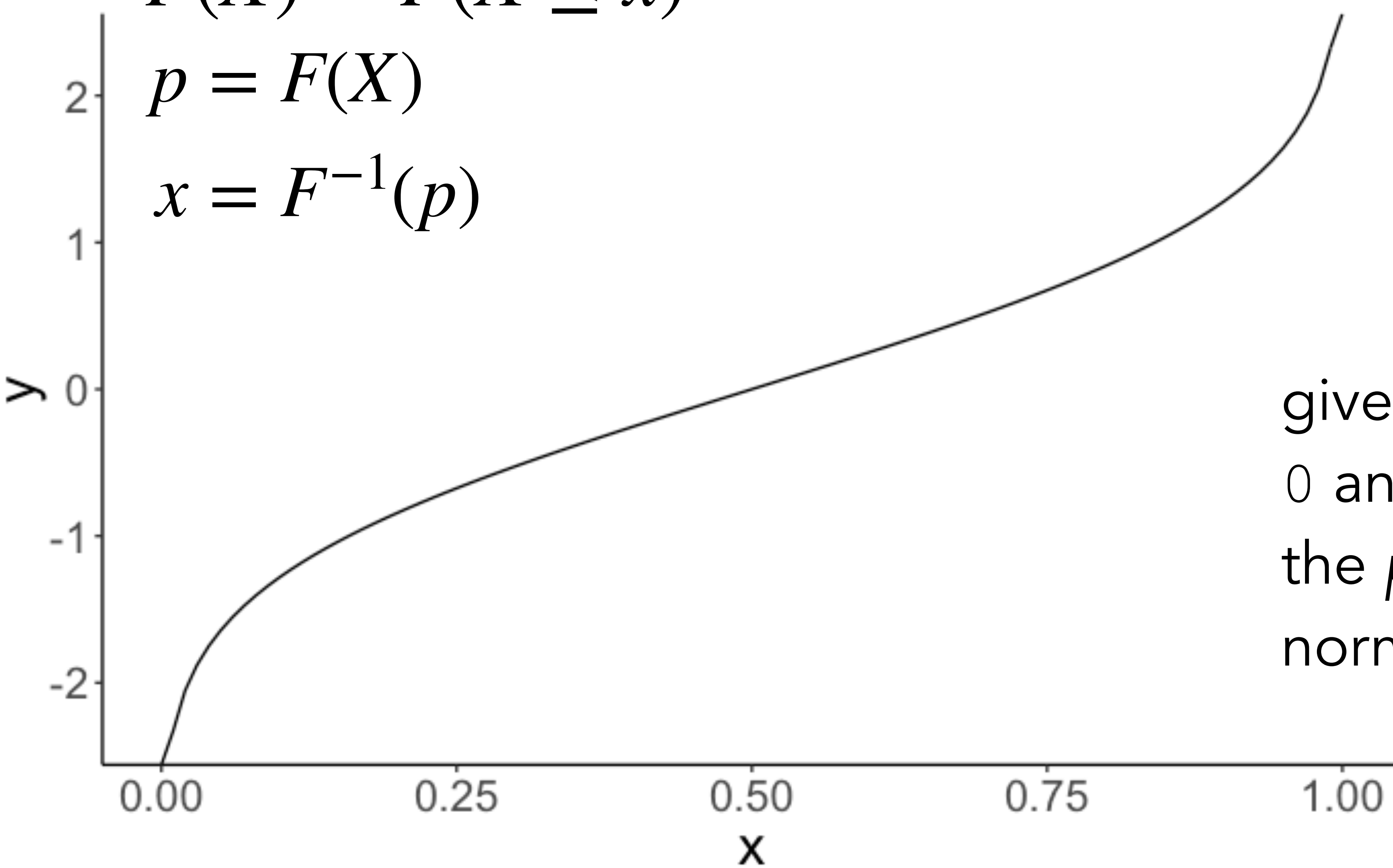
q = quantile

inverse cumulative distribution function

$$F(X) = P(X \leq x)$$

$$p = F(X)$$

$$x = F^{-1}(p)$$



given a number p between `0` and `1`, `qnorm` looks up the p-th quantile of the normal distribution

Inverse cumulative distribution function

```r
ggplot(data = tibble(x = c(0, 1)),
       mapping = aes(x = x)) +
  stat_function(fun = "qnorm",
                args = list(mean = 0,
                            sd = 1))
```

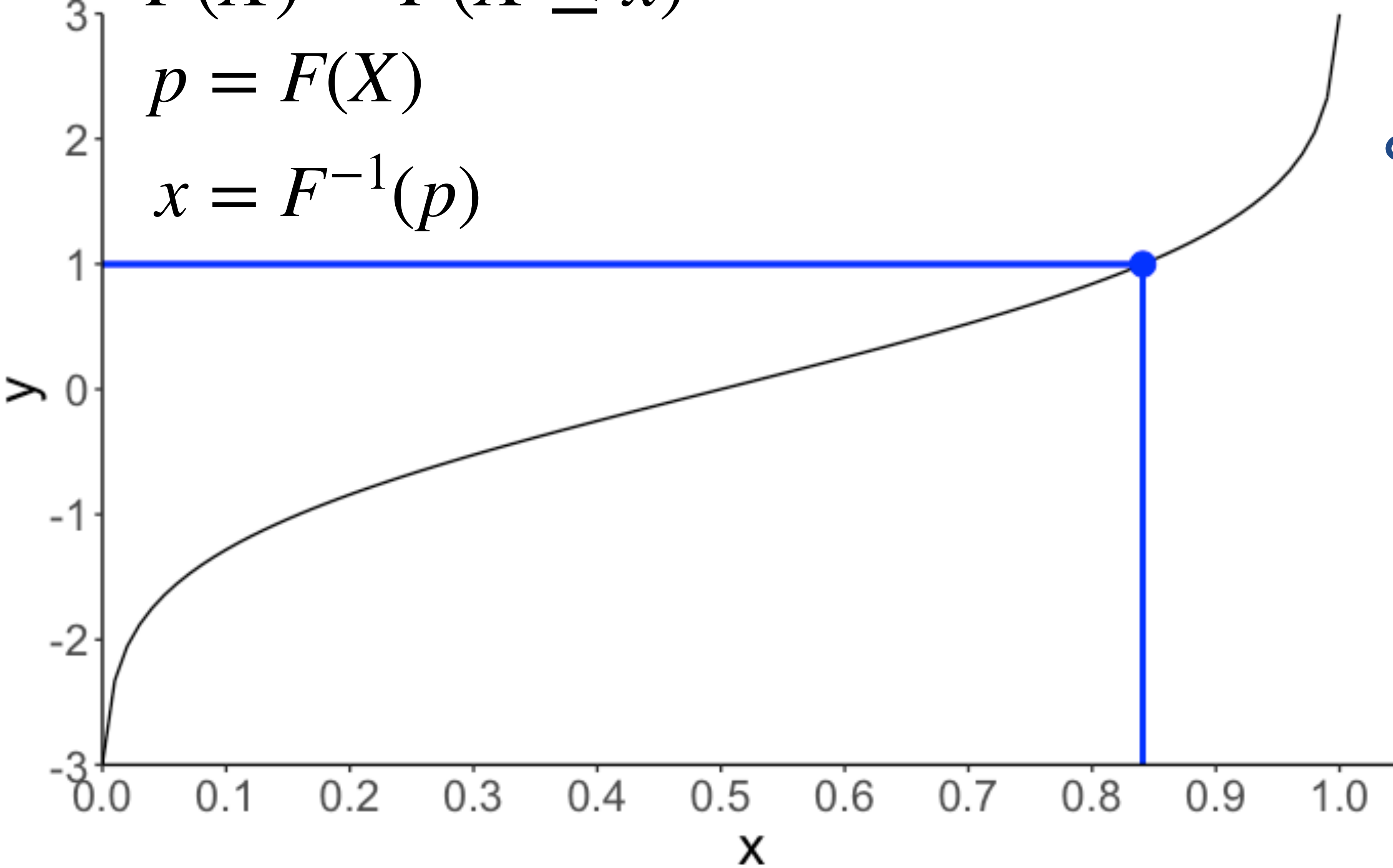
q = quantile

inverse cumulative distribution function

$$F(X) = P(X \leq x)$$

$$p = F(X)$$

$$x = F^{-1}(p)$$

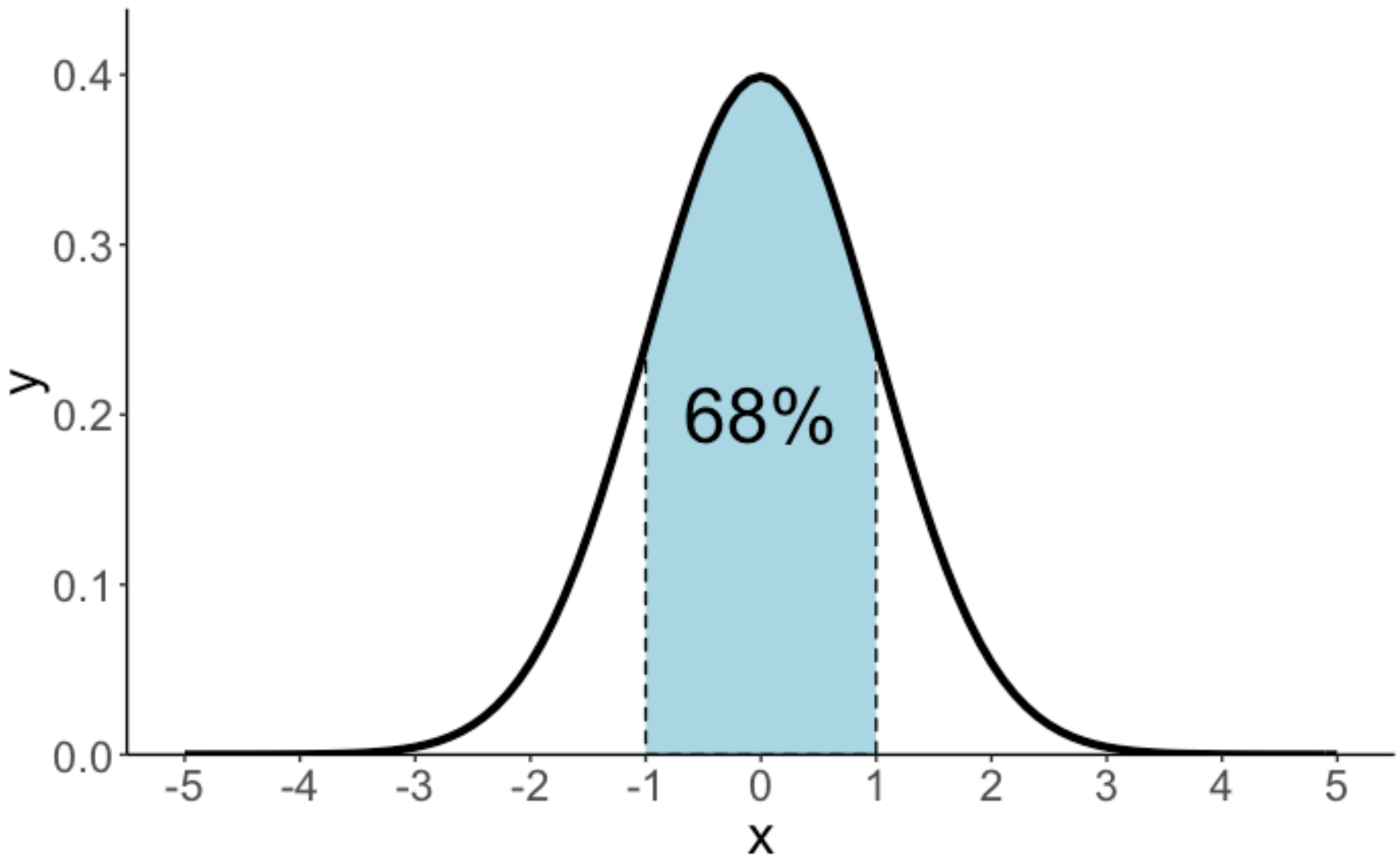



```r
qnorm(0.841,
      mean = 0,
      sd = 1)
```
≈ 1

Computing probabilities

Find the probability between two values of interest:

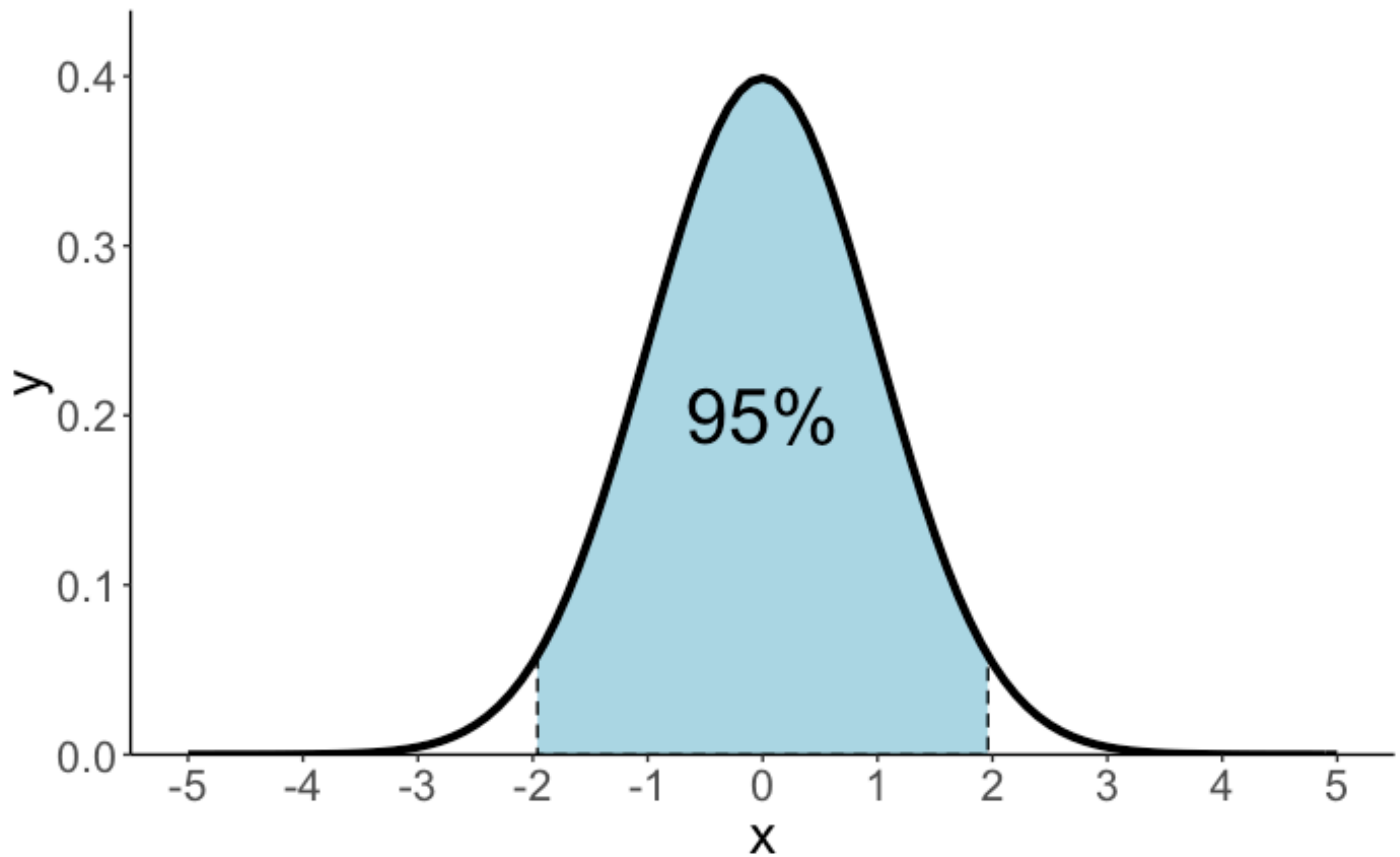
```
pnorm(1) - pnorm(-1) =
0.84     -  0.16     = 0.68
```



Computing probabilities

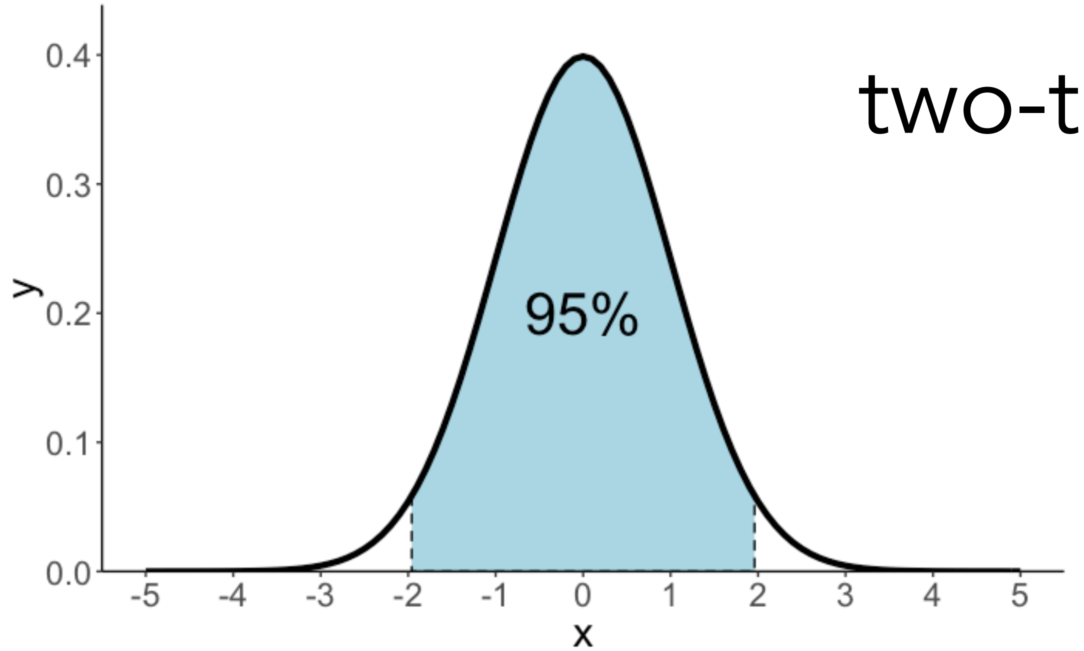
Find the lower and upper value so that 95% of the probability are contained between these values.

```
qnorm(0.025)= -1.96
qnorm(0.975)= 1.96
```



Computing probabilities

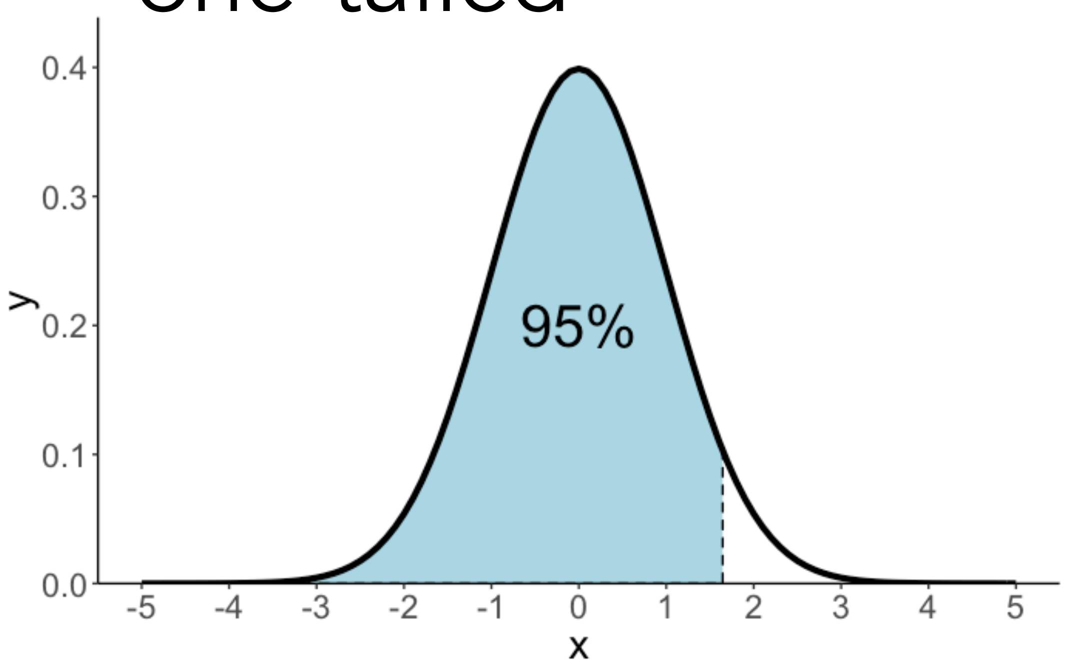
two-tailed



`qnorm(0.025)= -1.96`

`qnorm(0.975)= 1.96`

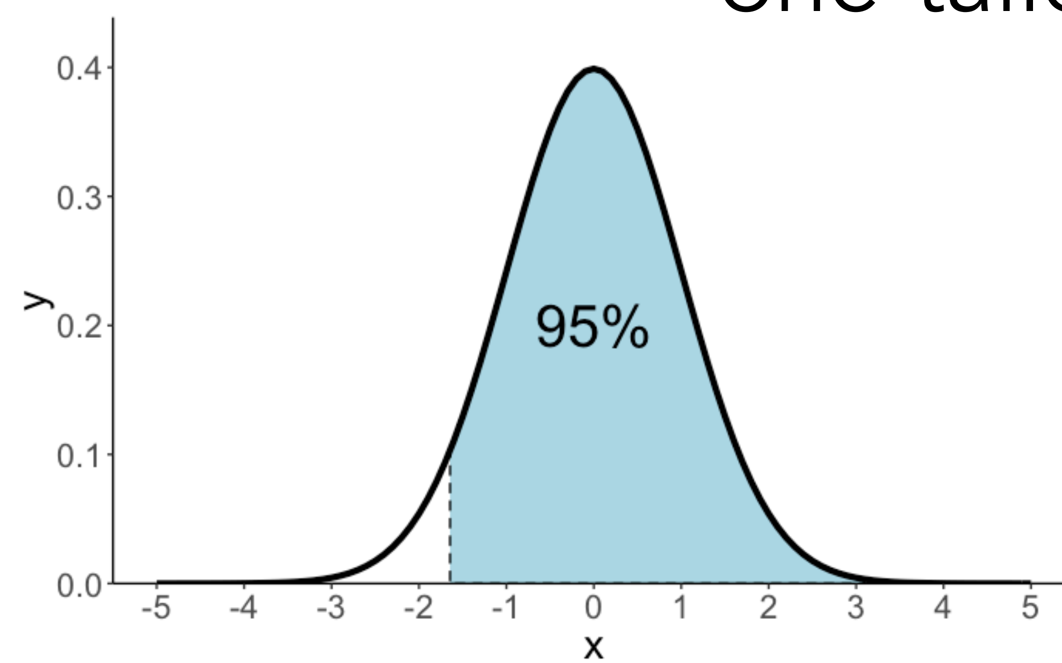
one-tailed



`qnorm(0)= -Inf`

`qnorm(0.95)= 1.64`

one-tailed



`qnorm(0.05)= -1.64`

`qnorm(1)= Inf`

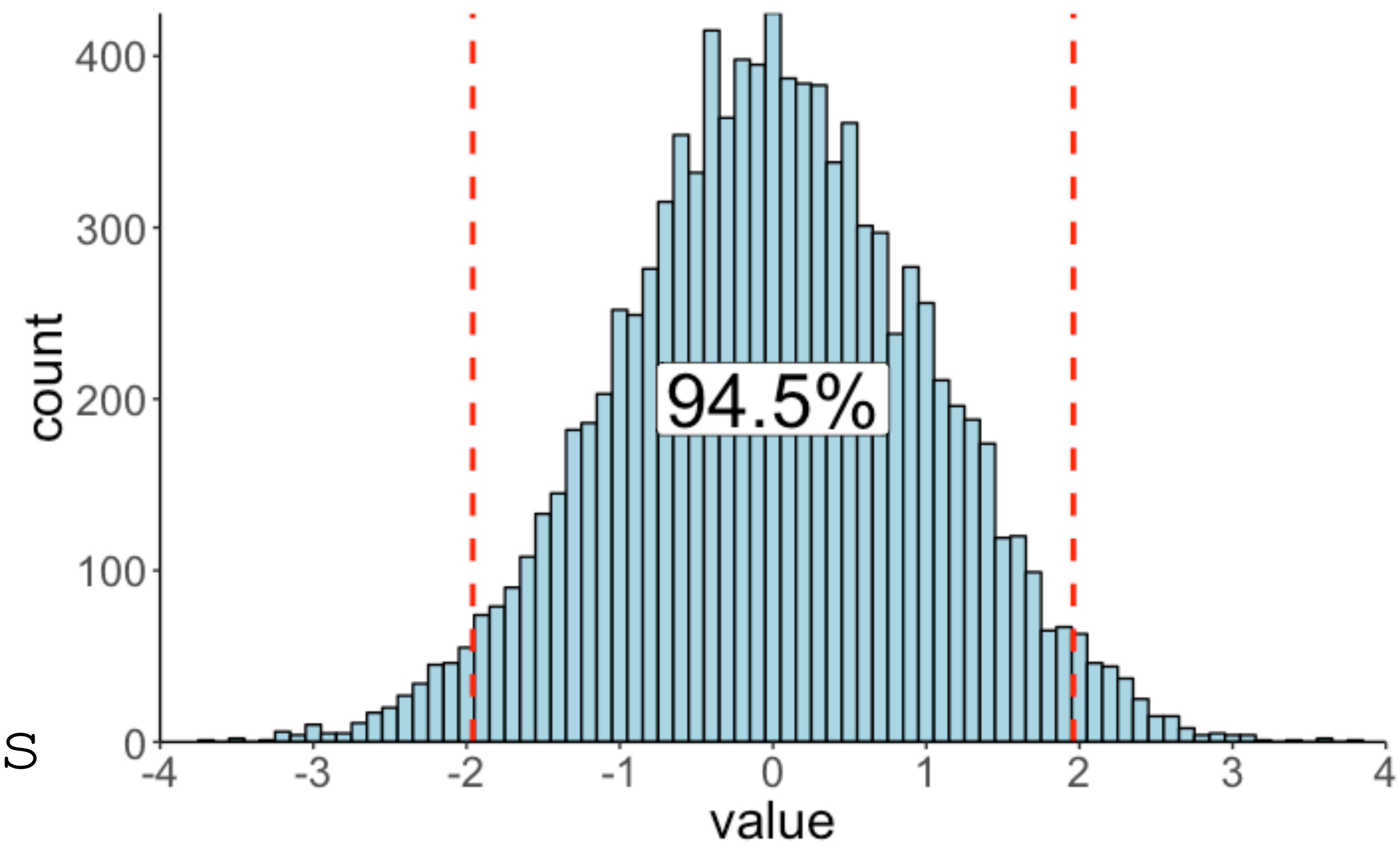
Computing probabilities: Simulation

```r
df.samples = tibble(
  sample = 1:nsamples,
  value = rnorm(n = nsamples, mean = 0, sd = 1)
)
```

sample	value
1	-0.626
2	0.184
3	-0.836
4	1.595
5	0.330
6	-0.820

⋮

nrow = nsamples



Plan for today

- Quick review of causality
- Working with probability distributions
  - `dnorm()`, `pnorm()`, `qnorm()`, `rnorm()`
  - computing probabilities
- **Bayesian inference**
  - analytic solution
  - via sampling
- Working with samples
  - Understanding `density()`
  - Understanding `quantile()`
  - Comparing distributions

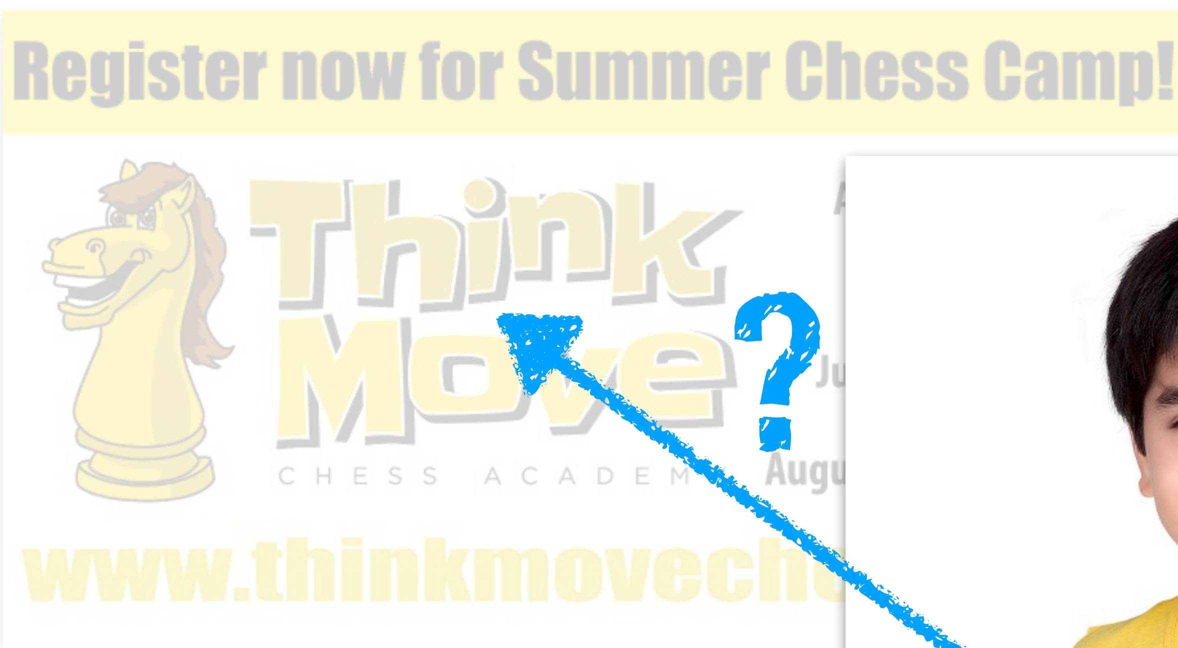
Summer camp

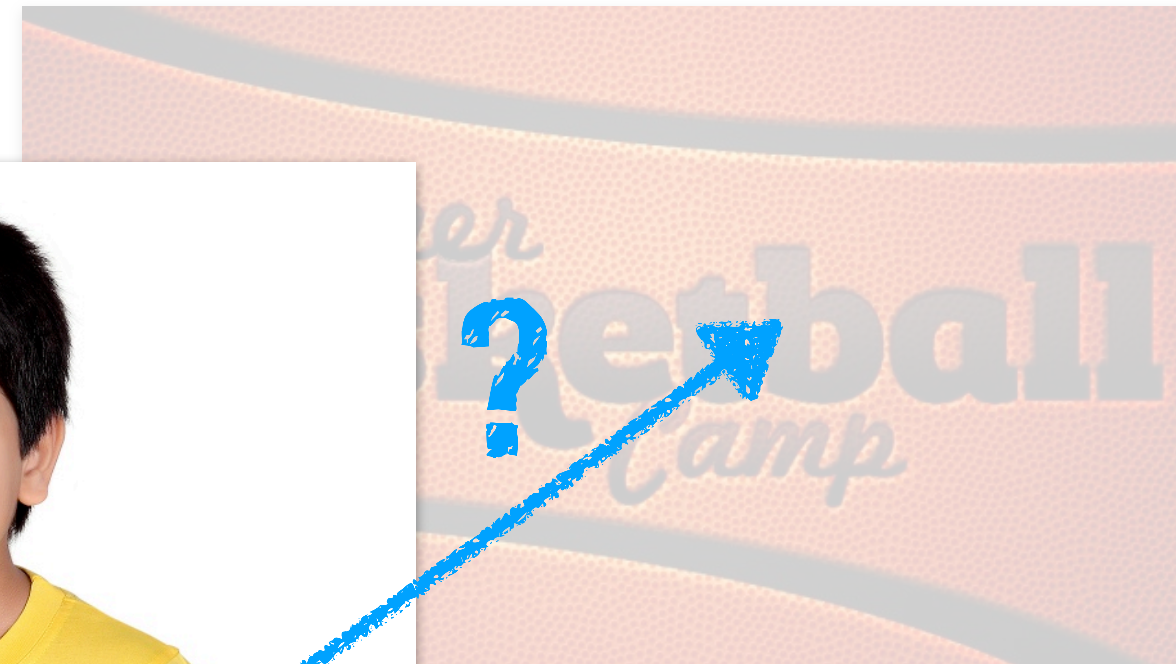
twice as many kids go to the basketball camp

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

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

Summer camp





twice as ma etball camp

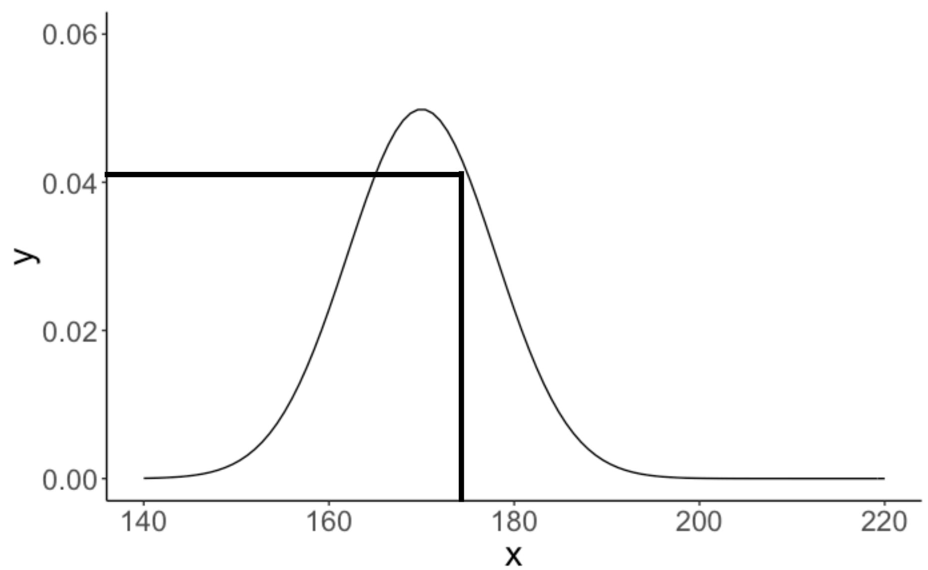
$X \sim$ Normal($\mu = 170$ al($\mu = 180$, $\sigma = 10$)

Analytic solution

Summer camp

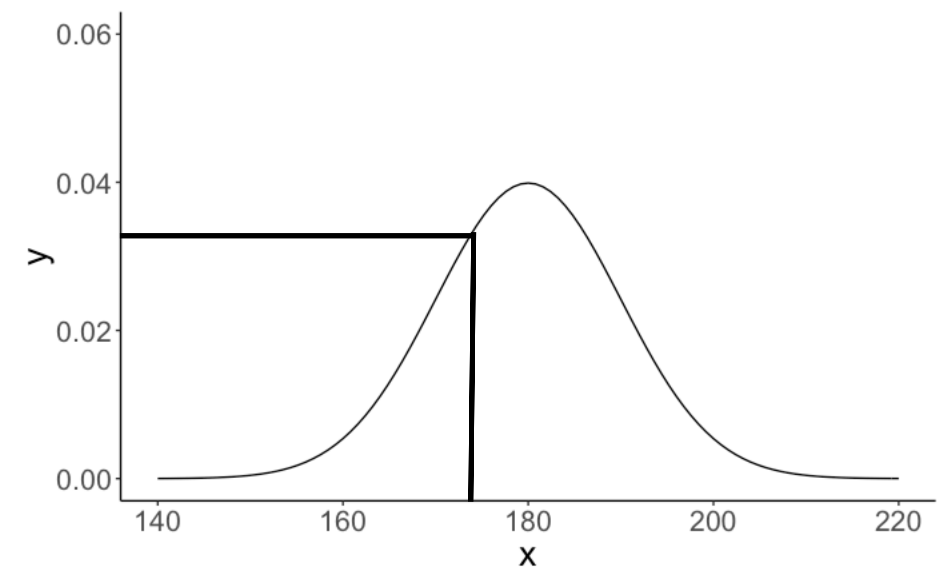
prior $p(\text{chess}) = \dfrac{1}{3}$ $p(\text{basketball}) = \dfrac{2}{3}$

likelihood



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

= 0.041



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

= 0.035

posterior

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

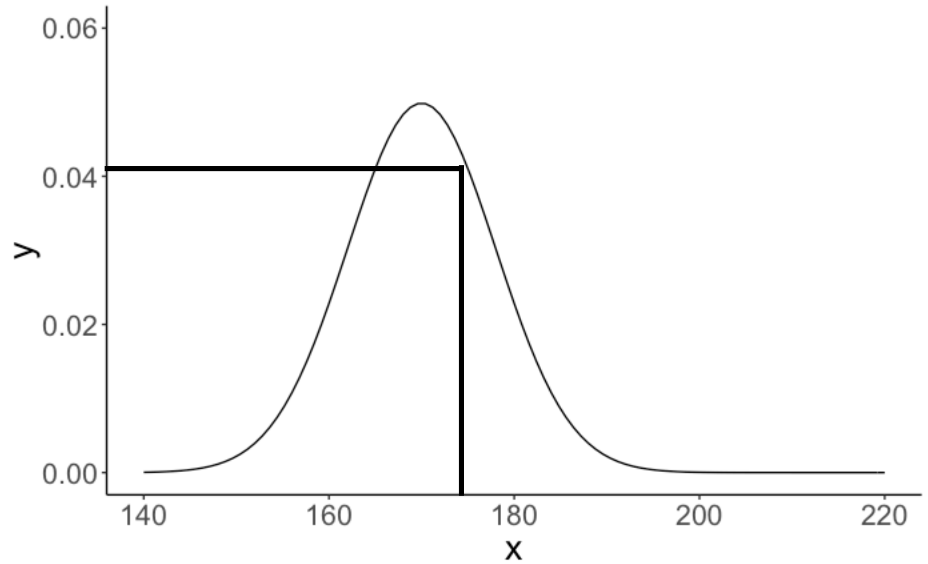
$$p(\text{basketball} \mid 175) = \frac{p(175 \mid \text{basketball}) \cdot p(\text{basketball})}{p(175 \mid \text{basketball}) \cdot p(\text{basketball}) + p(175 \mid \text{chess}) \cdot p(\text{chess})}$$

Summer camp

prior

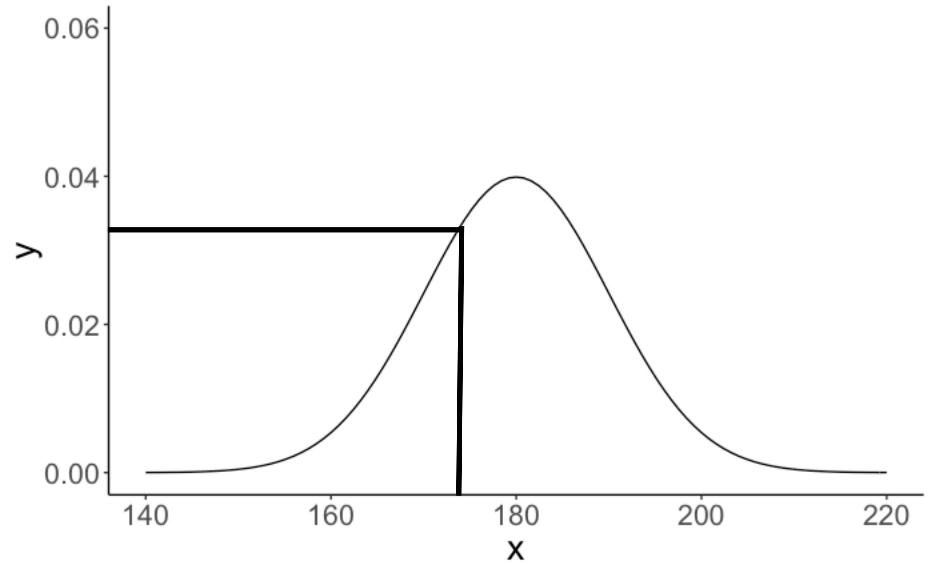
$$p(\text{chess}) = \frac{1}{3} \qquad p(\text{basketball}) = \frac{2}{3}$$

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!

Sampling solution

Summer camp: Via sampling

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

```r
df.camp %>%
  filter(height == 175) %>%
  count(sport) %>%
  spread(sport, n) %>%
  summarize(prob_basketball =
            basketball/(basketball
            + chess))
```

doesn't work!

Summer camp: Via sampling

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

```
df.camp %>%
  filter(between(height,
                 left = 174,
                 right = 176)) %>%
  count(sport) %>%
  spread(sport, n) %>%
  summarize(prob_basketball =
              basketball/(basketball
              + chess))
```

this works!

≈ 0.63

Plan for today

- Quick review of causality
- Working with probability distributions
  - `dnorm()`, `pnorm()`, `qnorm()`, `rnorm()`
  - computing probabilities
- Bayesian inference
  - analytic solution
  - via sampling
- **Working with samples**
  - Understanding `density()`
  - Understanding `quantile()`
  - Comparing distributions

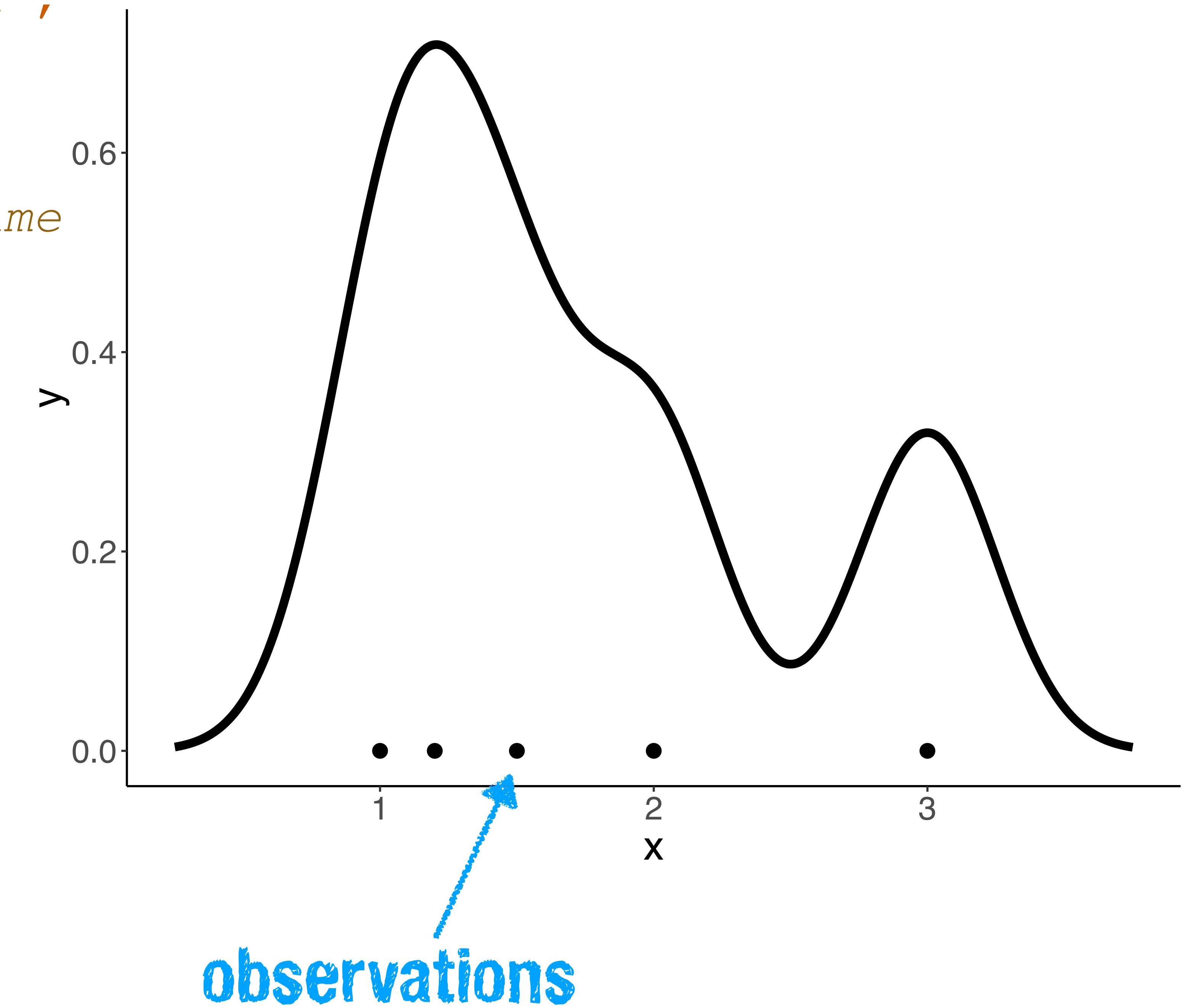
Understanding `density()`

Understanding `density()`

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

# save density as data frame
df.density = tibble(
  x = density$x,
  y = density$y
)
```

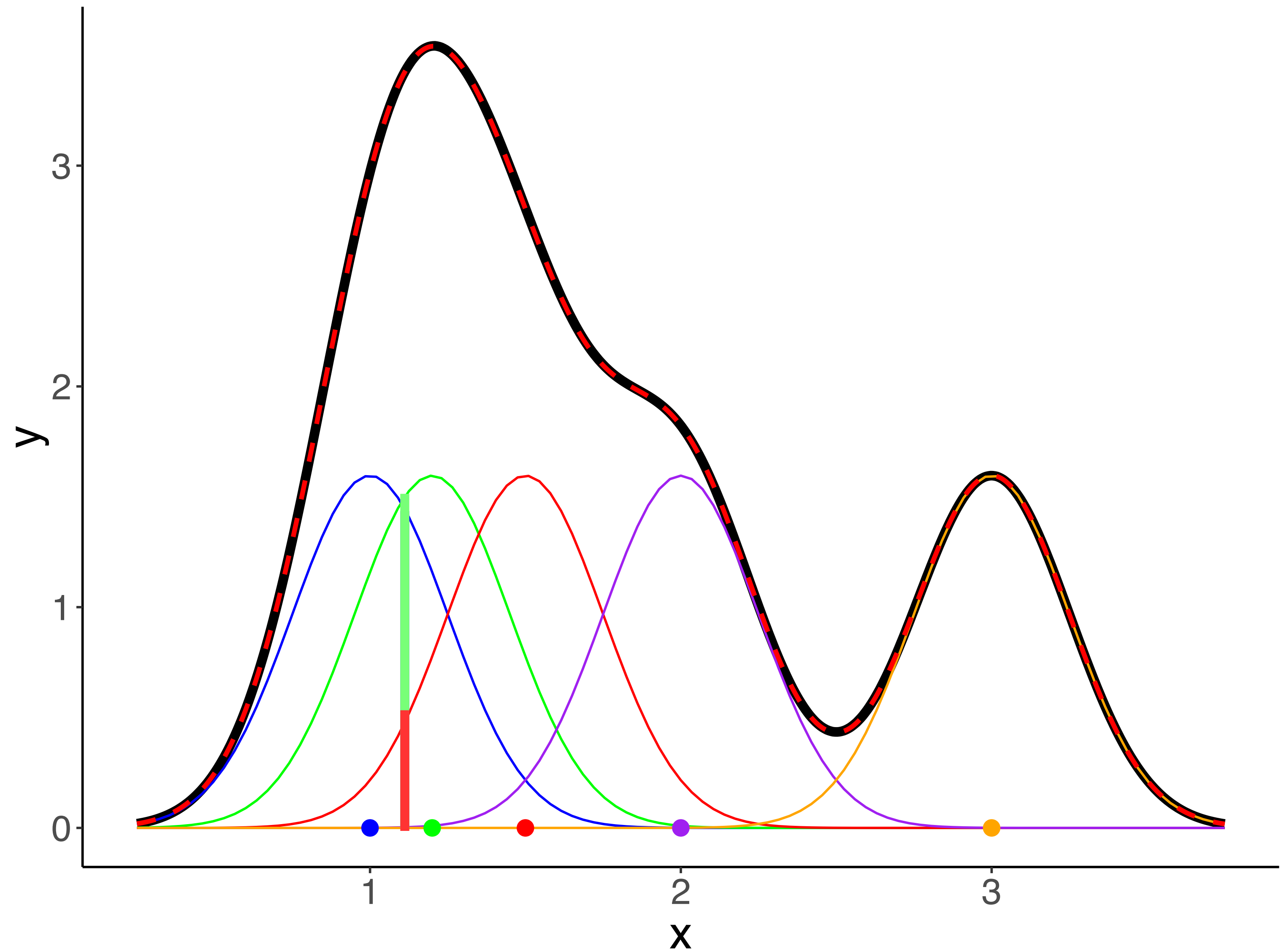
x	y
0.250	0.004
0.257	0.004
0.264	0.005
0.271	0.005
0.277	0.005
0.284	0.006

nrow = 512



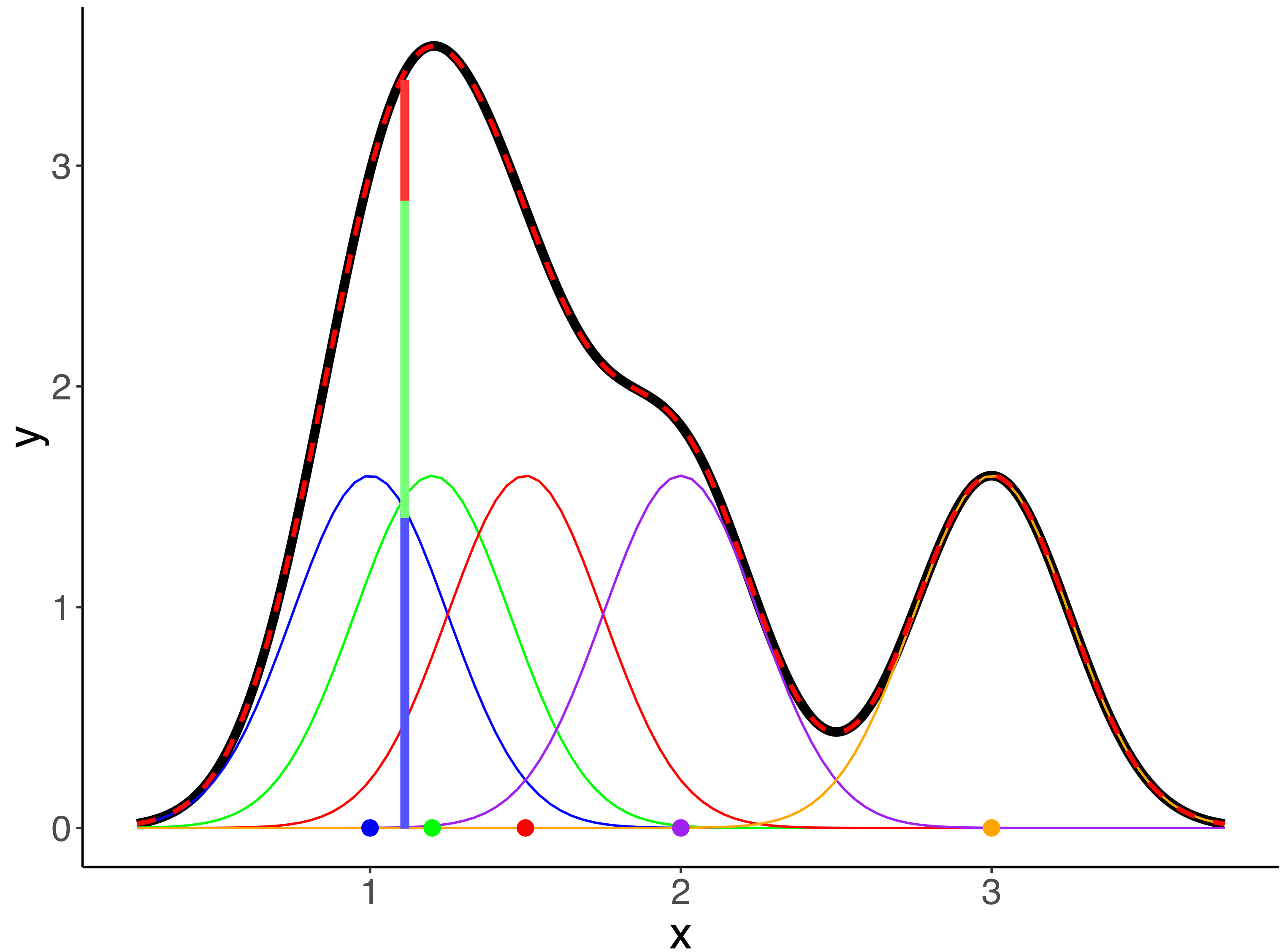
Understanding `density()`

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

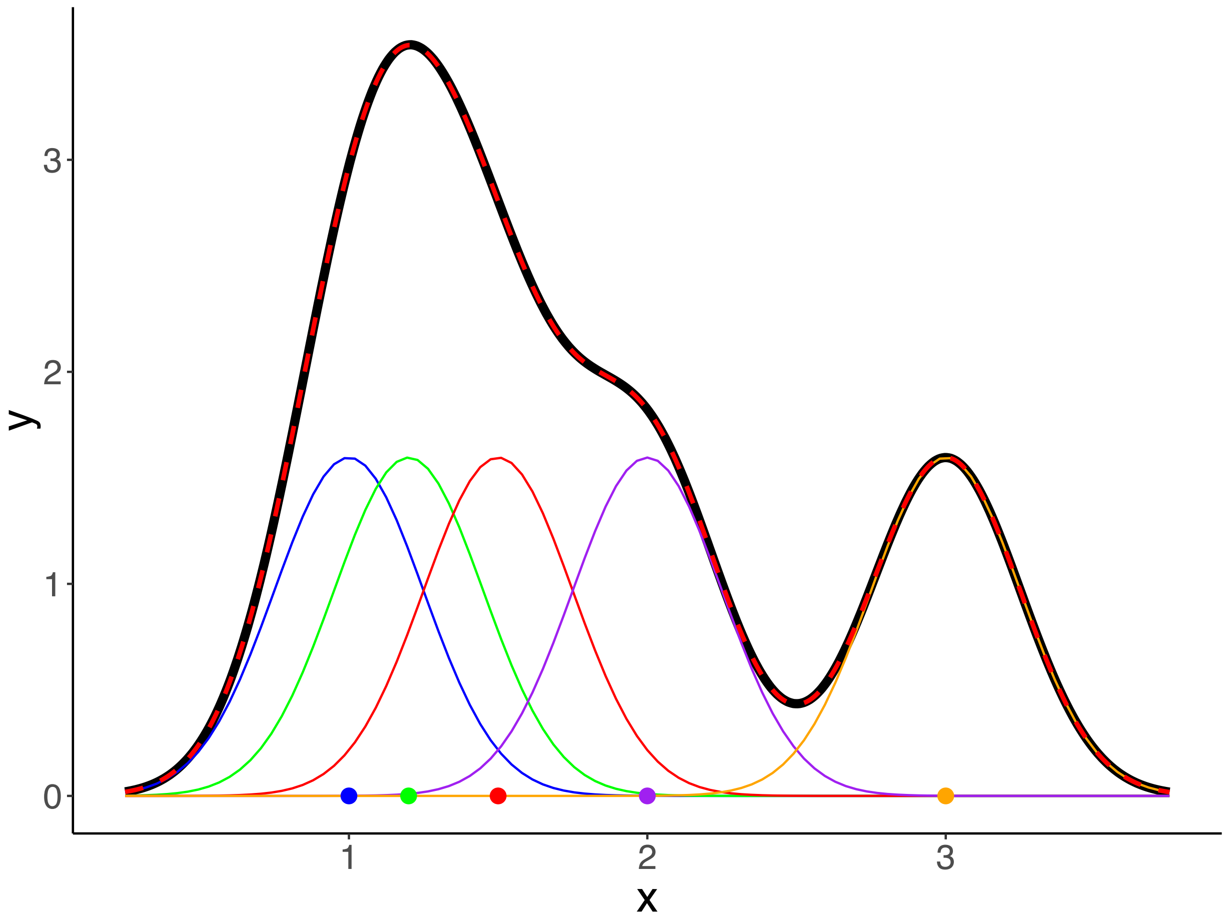


Understanding `density()`

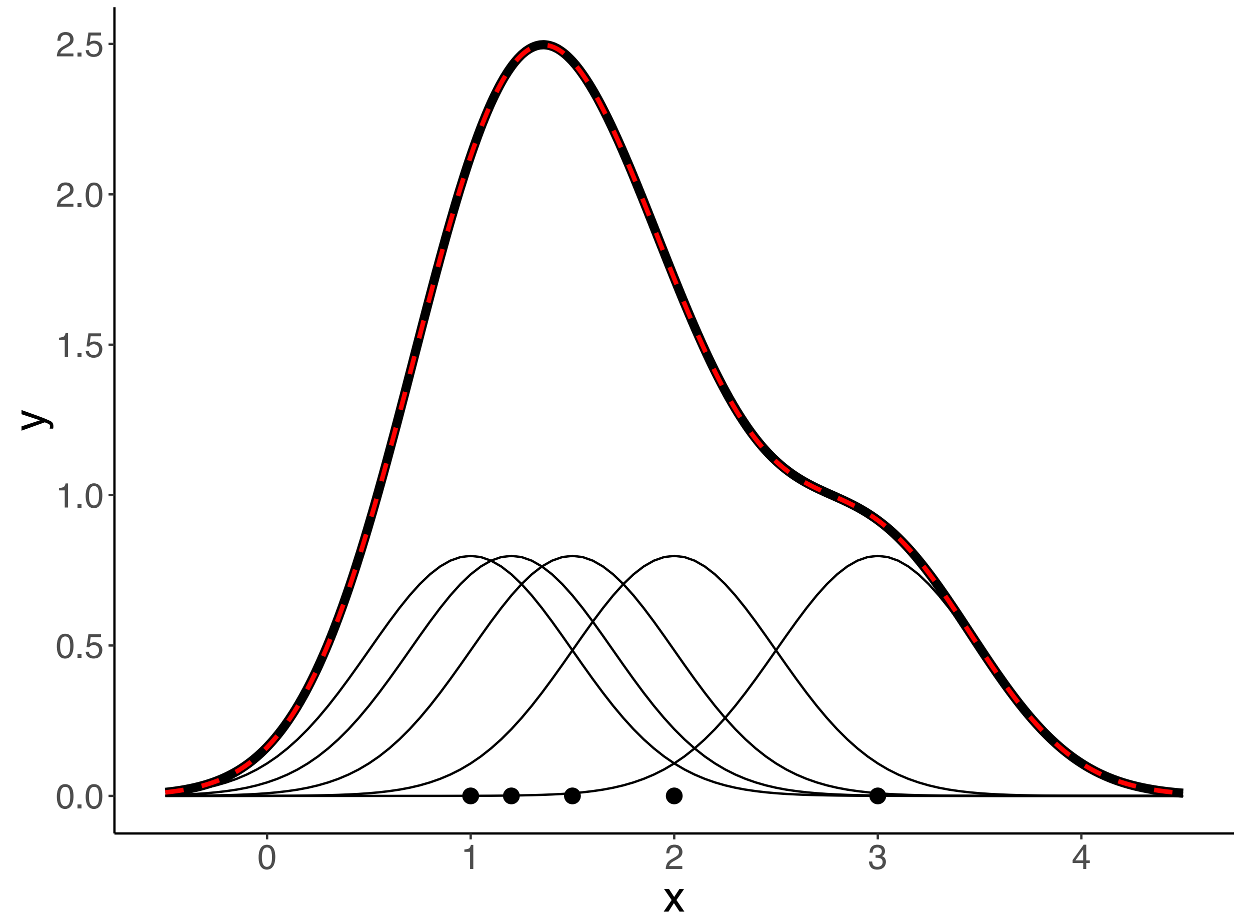
x	y	observation_1	observation_2	observation_3	observation_4	observation_5	sum_norm
0.250	0.019	0.018	0.001	0	0	0	0.019
0.257	0.021	0.019	0.001	0	0	0	0.021
0.264	0.023	0.021	0.001	0	0	0	0.022
0.271	0.024	0.023	0.002	0	0	0	0.024
0.277	0.027	0.024	0.002	0	0	0	0.026
0.284	0.029	0.026	0.002	0	0	0	0.028



Understanding `density()`



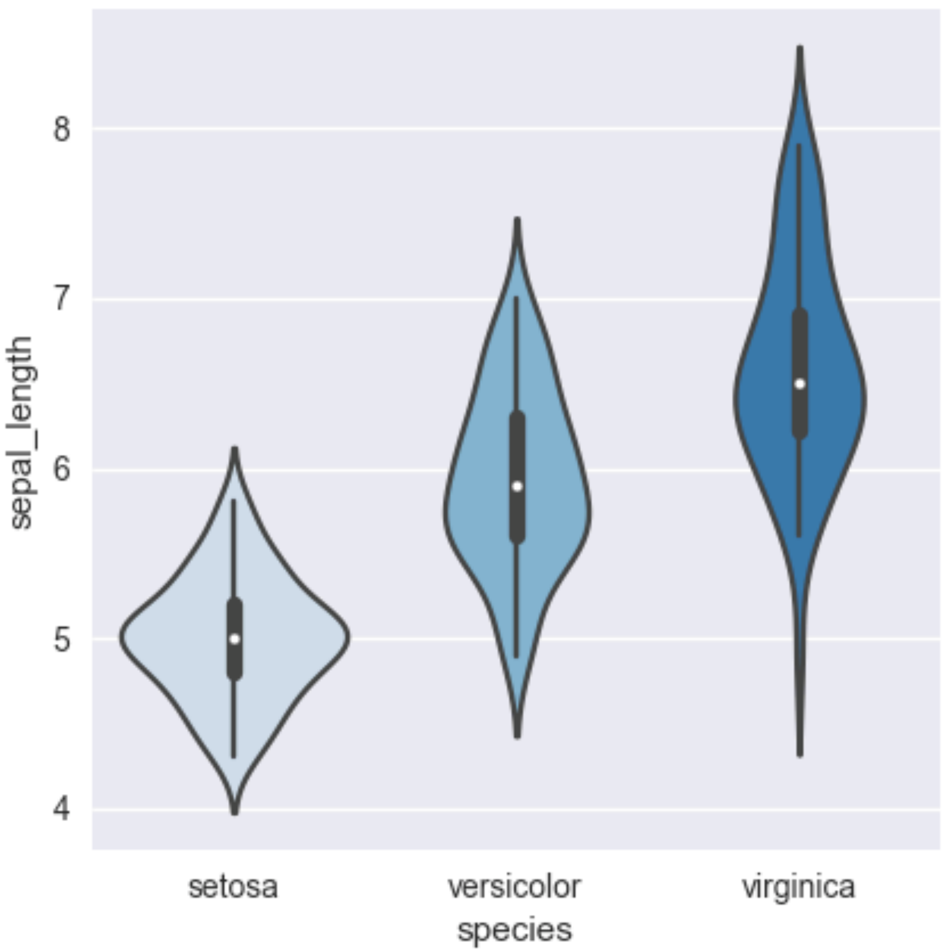
`density``(bw = 0.25)`



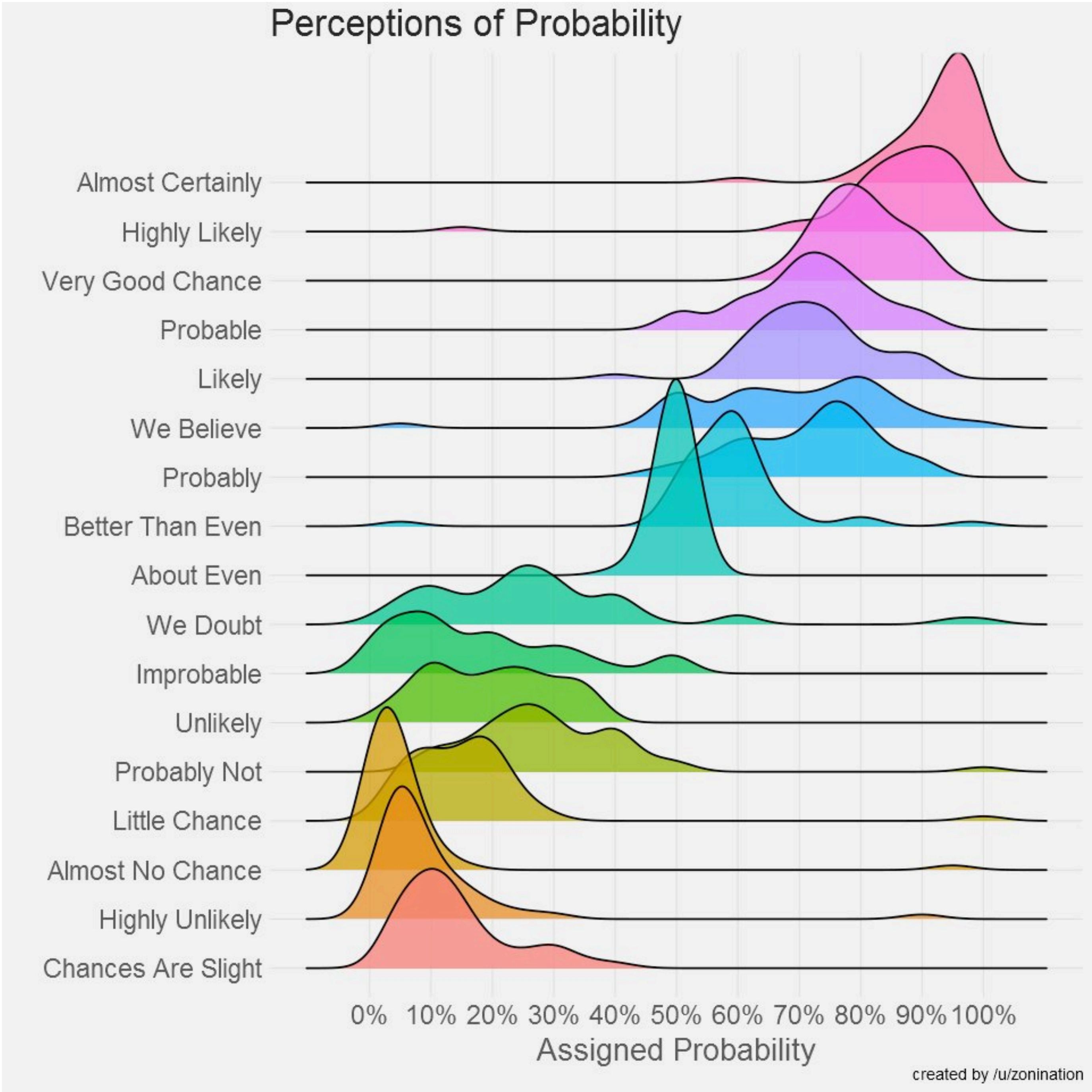
`density``(bw = 0.5)`

Understanding `density()`

violinplot



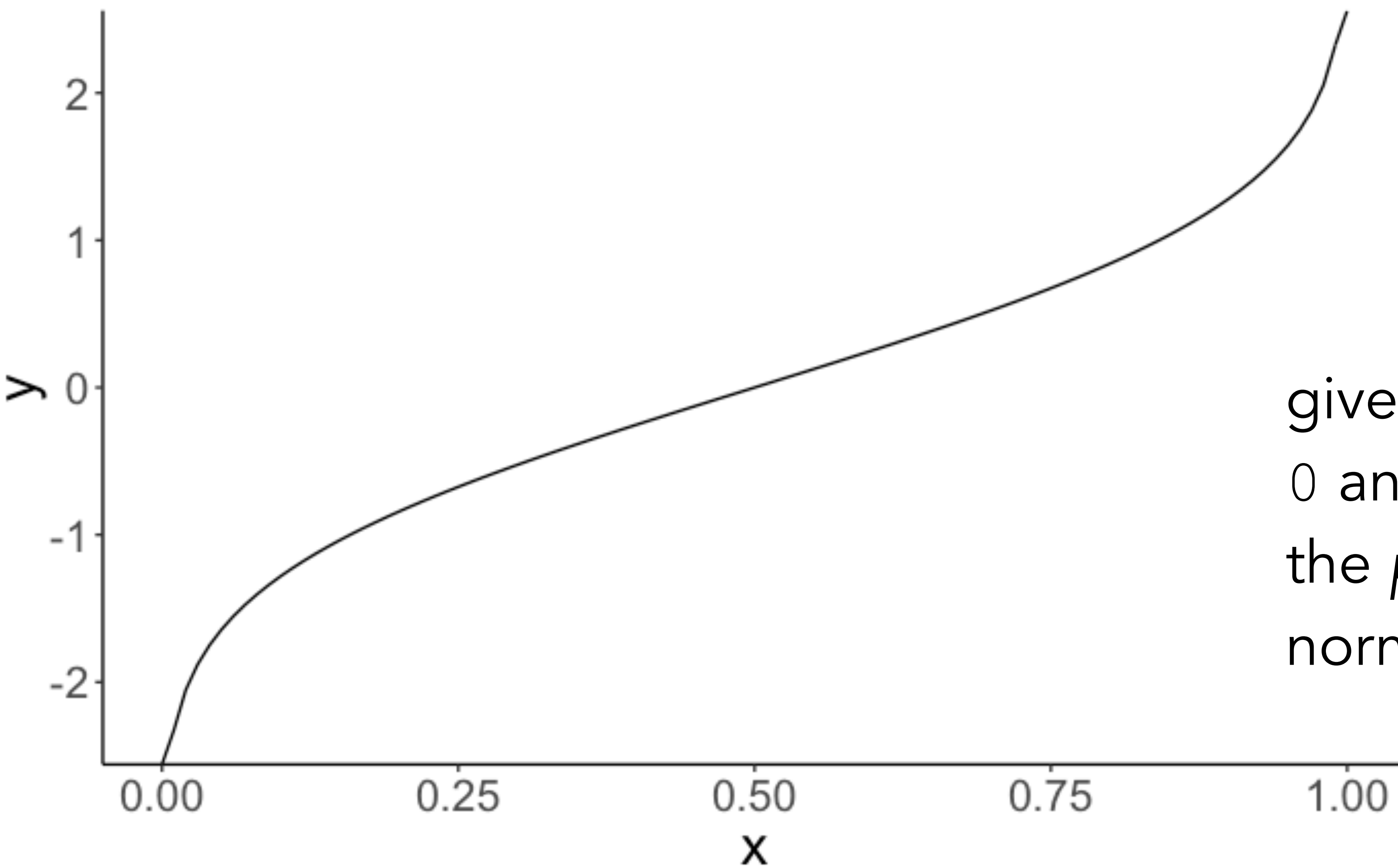
joyplot



Using `quantile()`

Using **`quantile()`**

- `quantile()` is for samples what `qnorm()` is for the normal distribution



given a number *p* between `0` and `1`, `qnorm` looks up the *p*-th quantile of the normal distribution

Using `quantile()`

- `quantile()` is for samples what `qnorm()` is for the normal distribution

```r
# a sample from the normal distribution
df.quantile = tibble(
  sample = 1:nsamples,
  value = rnorm(n = nsamples))
```

sample	value
1	-0.63
2	0.18
3	-0.84
4	1.60
5	0.33
6	-0.82
7	0.49
8	0.74
9	0.58
10	-0.3

median

1st quartile   3rd quartile

```r
quantile(df.quantile, probs = c(0.25, 0.5, 0.75))
```

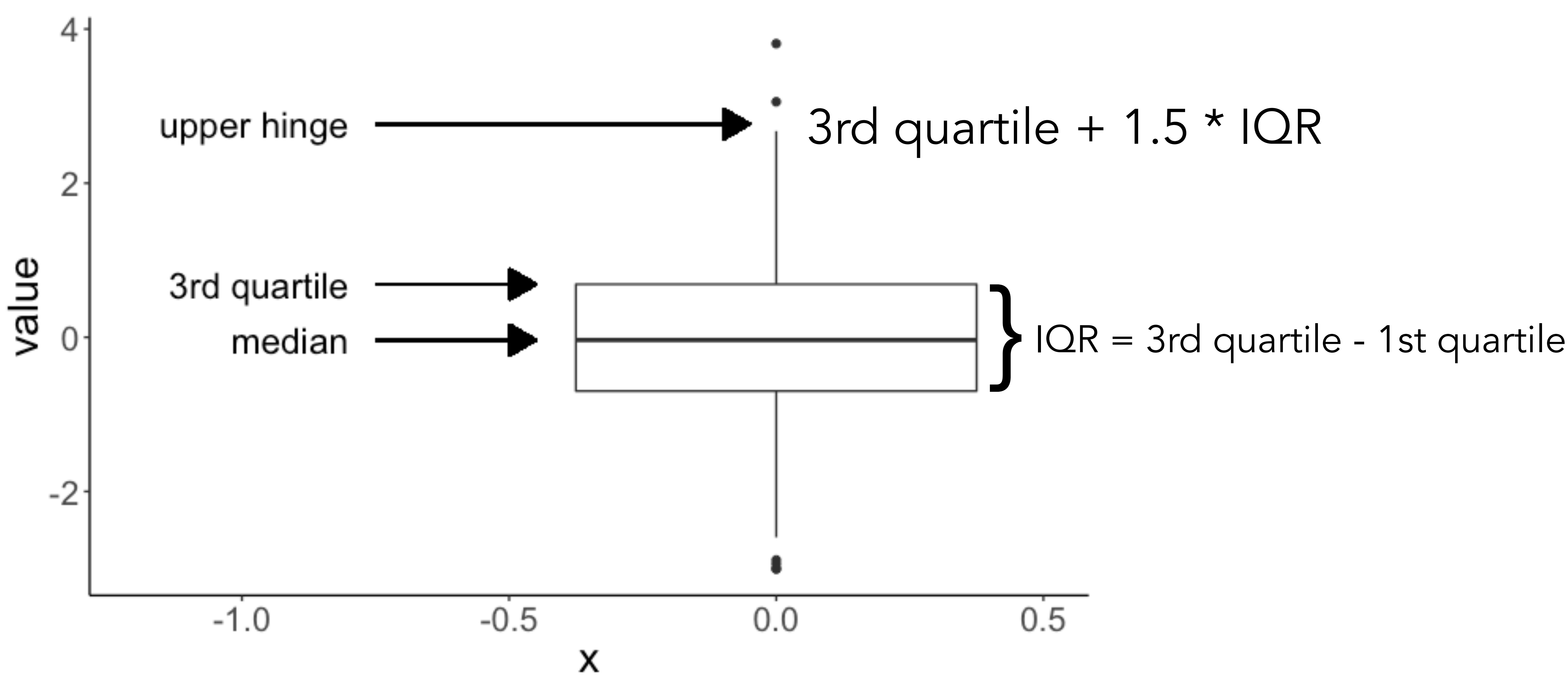
Using **`quantile()`**

median

1st quartile

3rd quartile

```
quantile(df.quantile, probs = c(0.25, 0.5, 0.75))
```

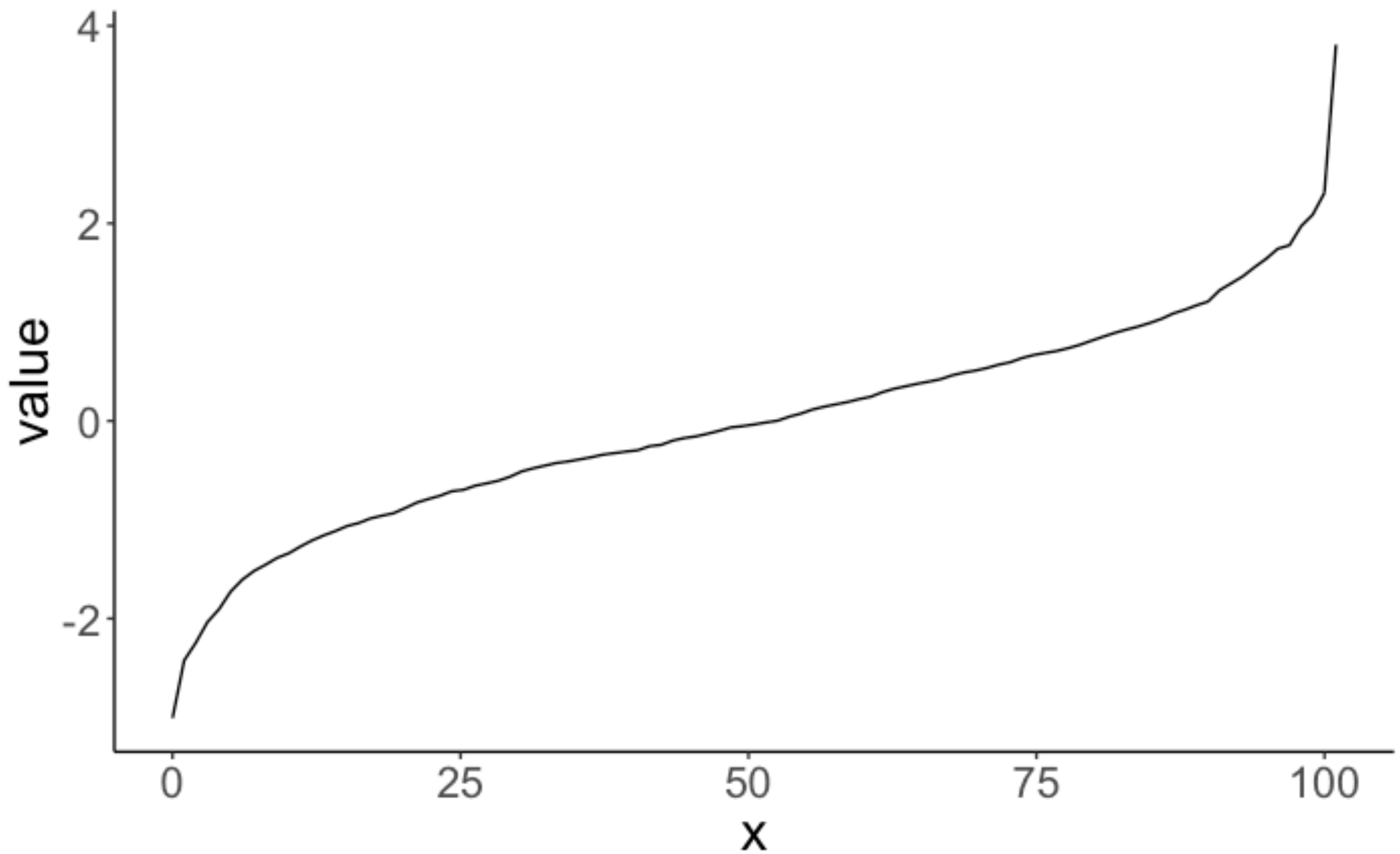


Using **quantile()**

```r
df.plot = df.quantile$value %>%
  quantile(probs = seq(0, 1, 0.01)) %>%
  as_tibble() %>%
  mutate(x = seq(0, n(), length.out = n()))
```

value	x
-3.008	0.00
-2.424	1.01
-2.246	2.02
-2.036	3.03
-1.905	4.04
1.744	95.95
1.779	96.96
1.971	97.97
2.089	98.98
2.308	99.99
3.810	101.00

```r
ggplot(data = df.plot,
       mapping = aes(x = x, y = value)) +
  geom_line()
```



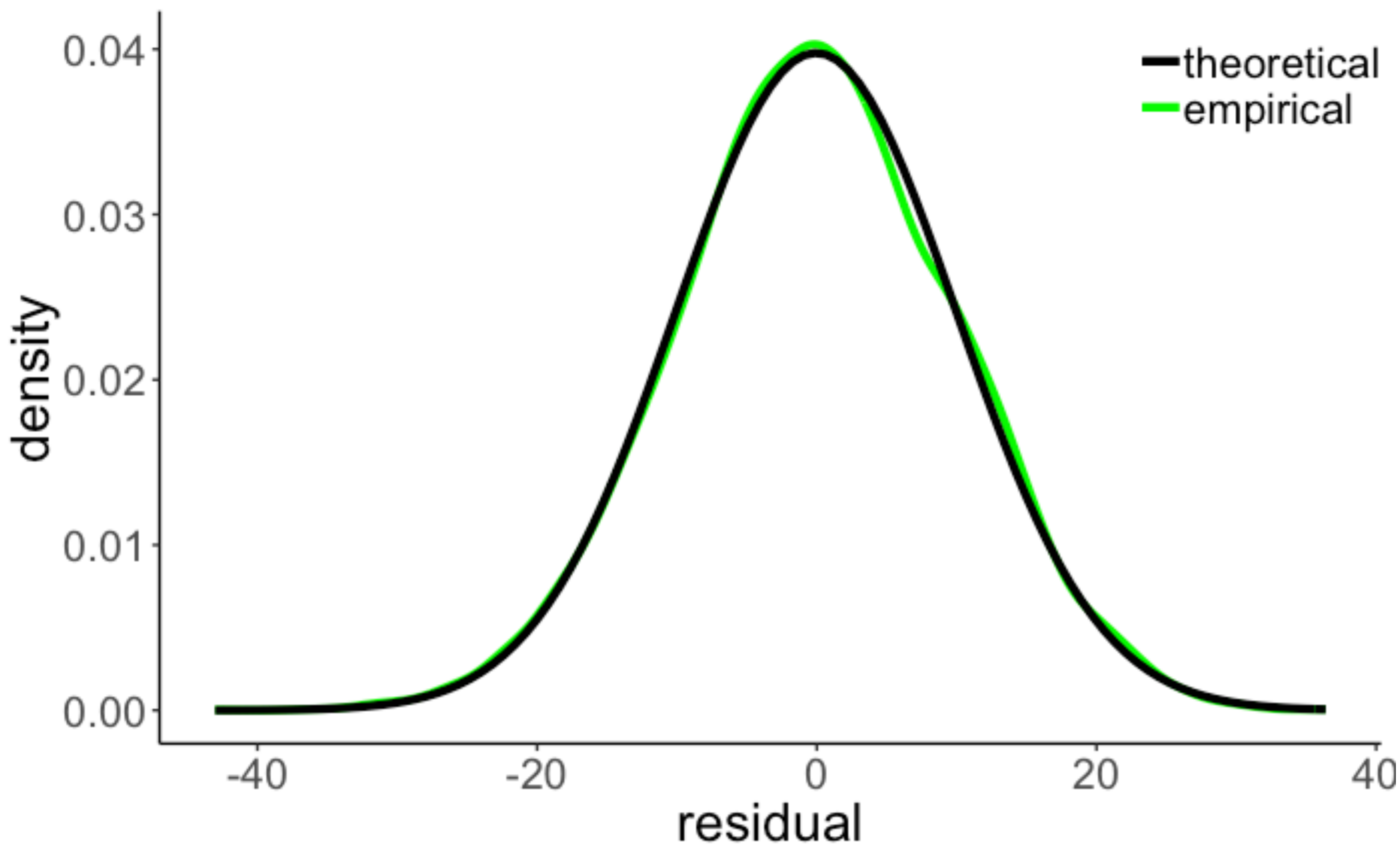
Working with samples

theoretical distribution		empirical sample
`dnorm()`	⟷	**`density()`**
`qnorm()`	⟷	**`quantile()`**

Comparing distributions

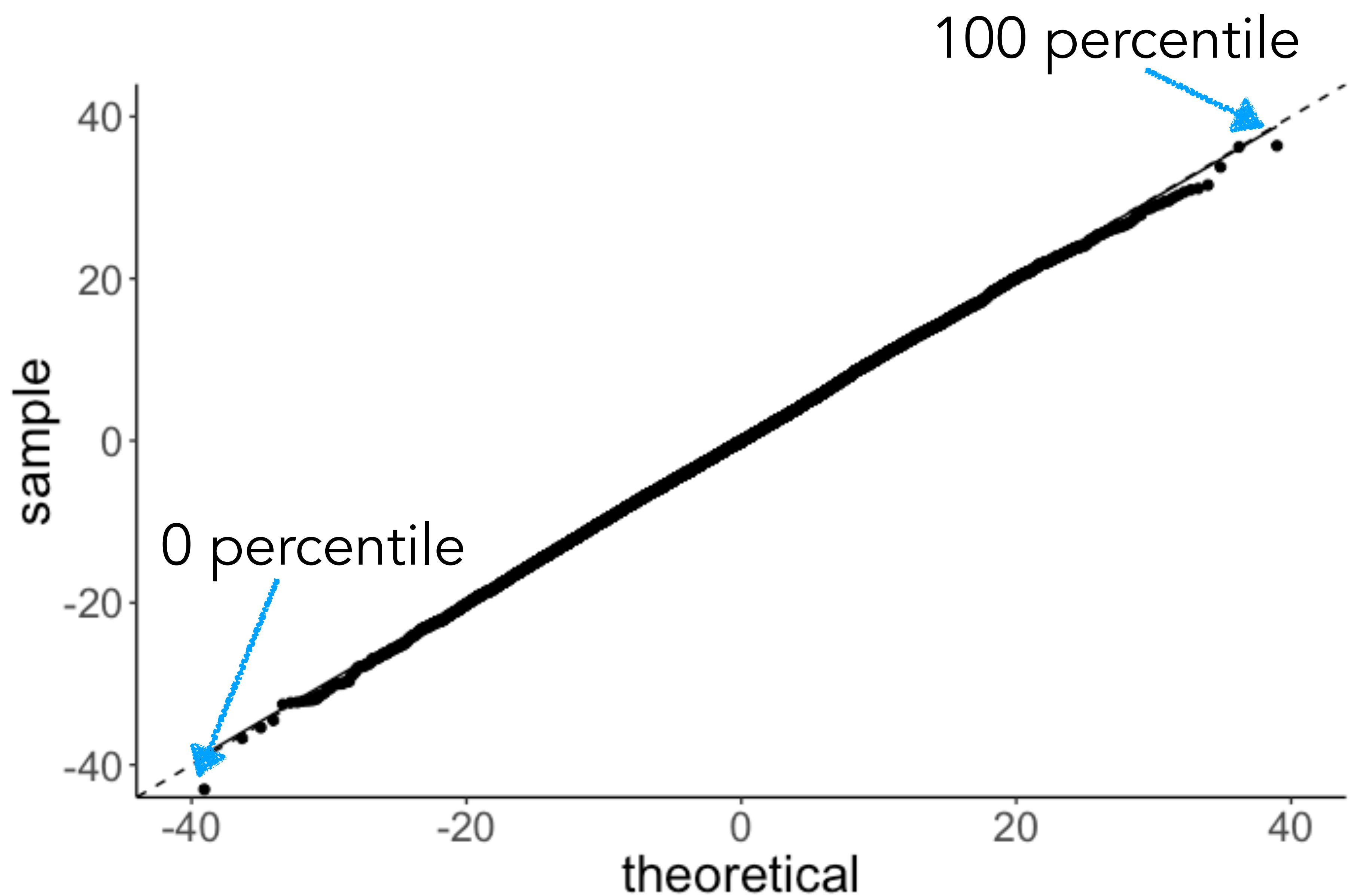
Comparing distributions

- **QQ plot** (Quantile-Quantile plot): useful for checking whether data (or errors) are normally distributed

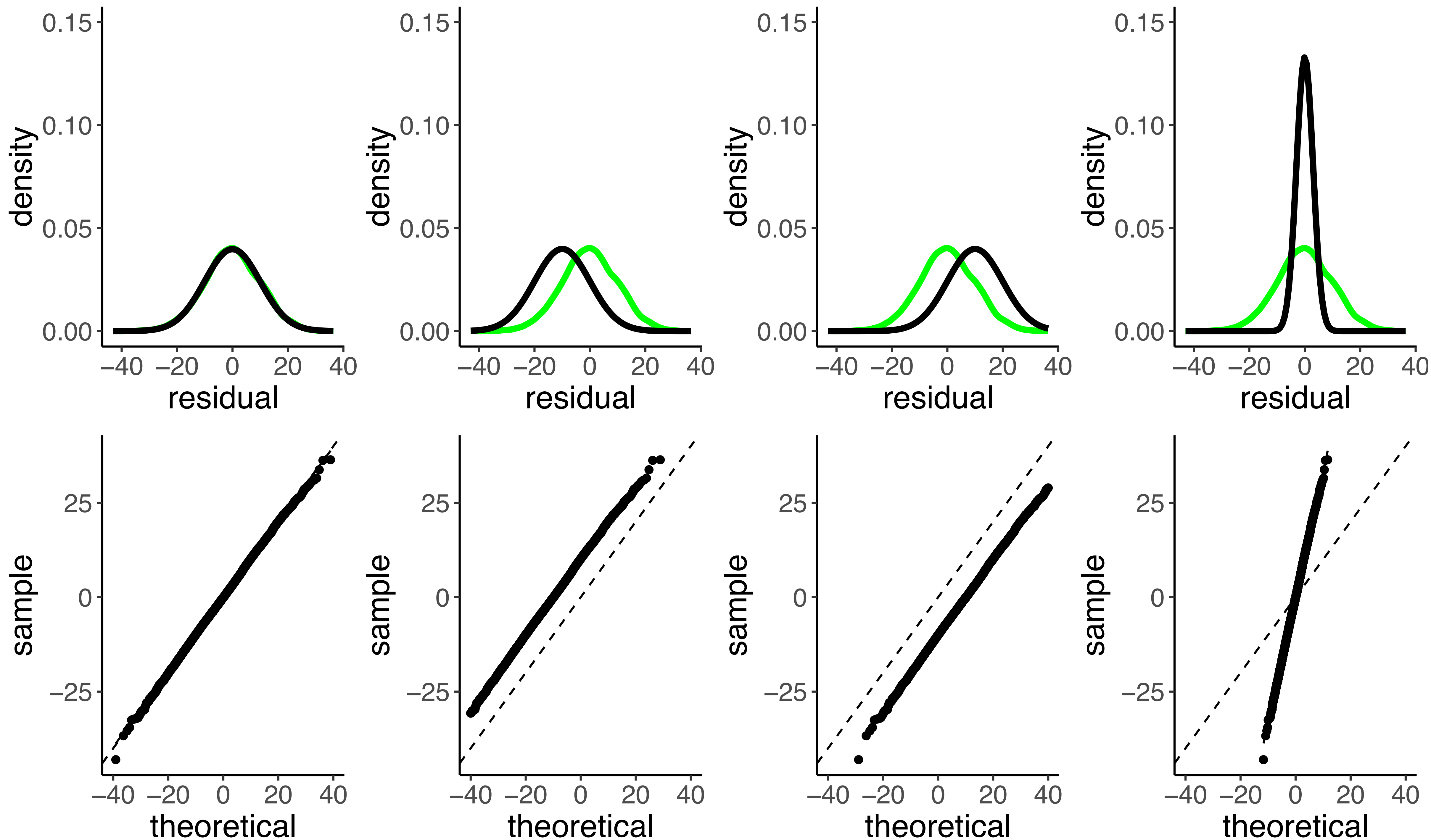


Comparing distributions

- **QQ plot** (Quantile-Quantile plot): useful for checking whether data (or errors) are normally distributed

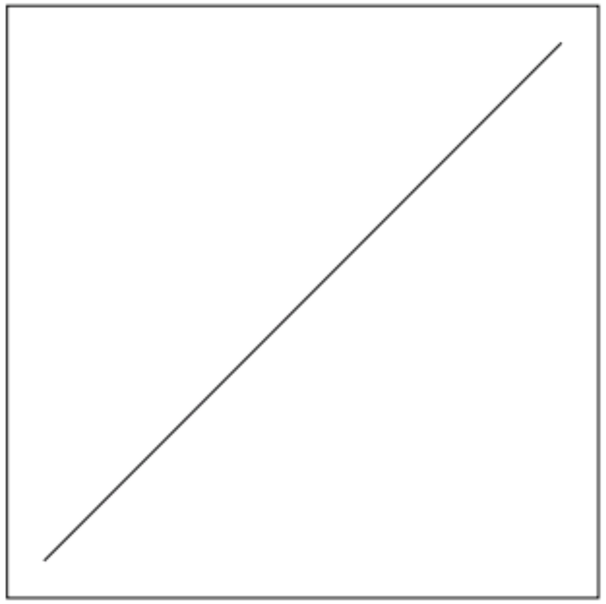


Comparing distributions



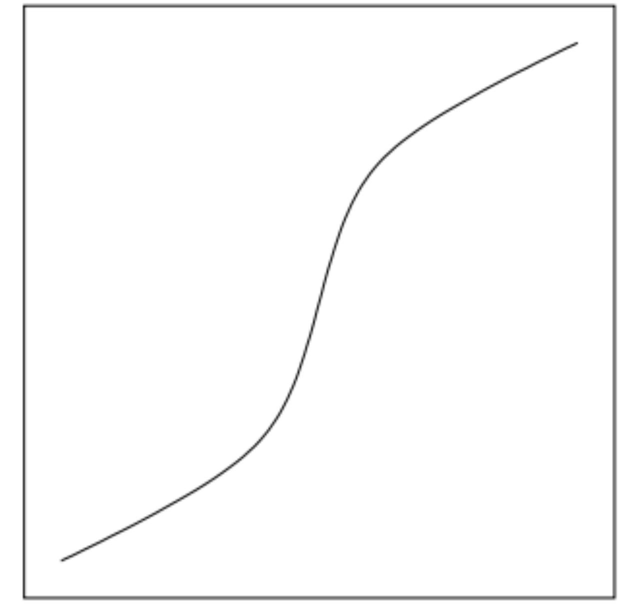
data is normally distributed if QQ plot is a line

Comparing distributions

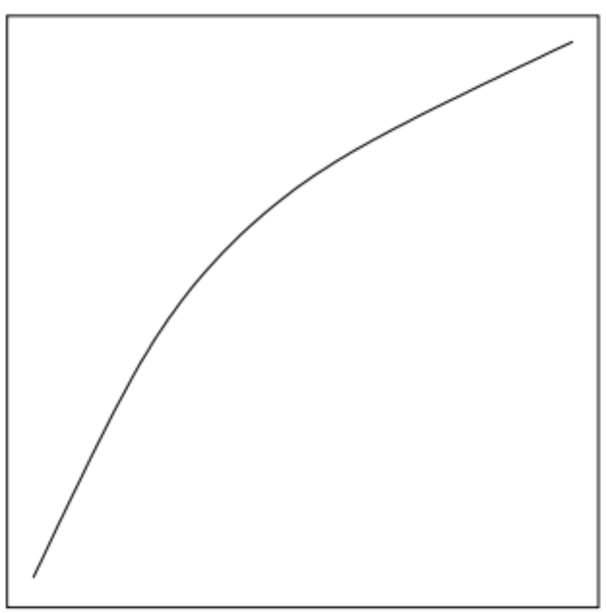


(a) Normally Distributed

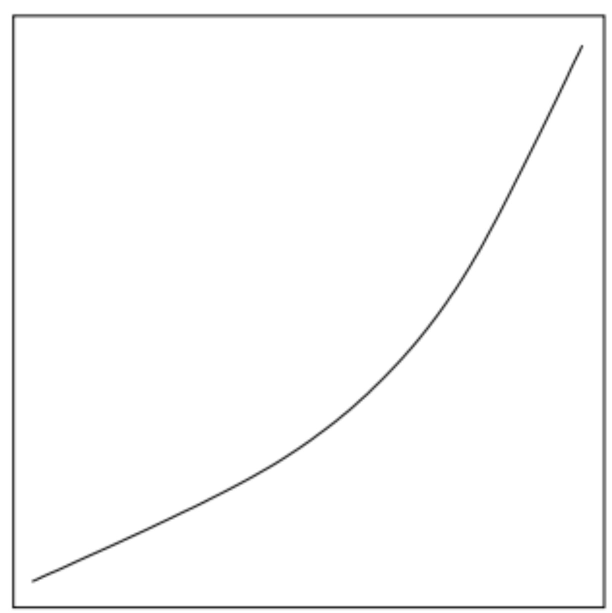
(b) Heavy Tails



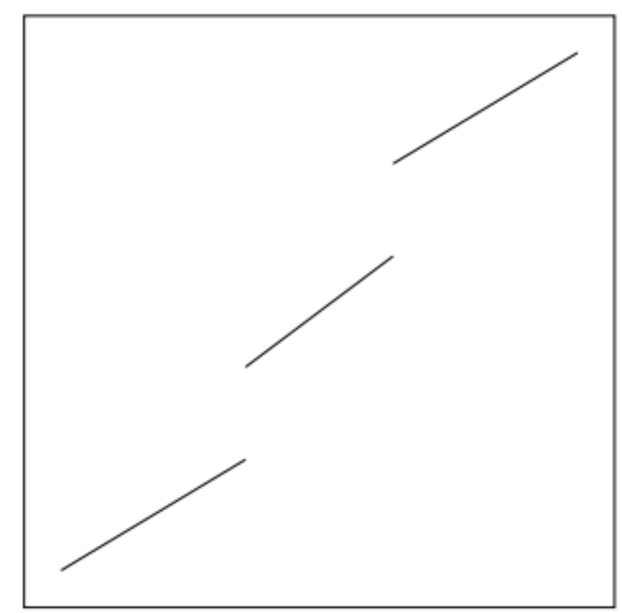
(c) Light Tails



(d) Skewed to the Left



(e) Skewed to the Right



(f) Separate Clusters

Plan for today

- **Quick review of causality**
- **Working with probability distributions**
  - `dnorm()`, `pnorm()`, `qnorm()`, `rnorm()`
  - computing probabilities
- **Bayesian inference**
  - analytic solution
  - via sampling
- **Working with samples**
  - Understanding `density()`
  - Understanding `quantile()`
  - Comparing distributions

Thank you!