Visualizing Data Distributions

Summer Institute in Data Science

Rolando J. Acosta

HARVARD
SCHOOL OF PUBLIC HEALTH

June 24, 2020

@RJANunez

What to expect today

- Often vectors of data are summarized by two numbers: the average and the standard deviation
- Ex 1: The average test score in the midterm was 86% with 11%
- Today we will learn effective data visualization techniques:
  - Properties of distributions
  - How to visualize distributions

Variable types

- We will be working with two types of variables: **categorical** and **numeric**
- **categorical:**
  1. **nominal:** values fall into unordered categories or classes
  2. **ordinal:** values fall into ordered categories or classes
- **numeric:**
  1. **discrete:** quantities that take on only specified values (e.g, integers or counts)
  2. **continuous:** quantities that can take on any value

Variable types: Examples

- We will be working with two types of variables: **categorical** and **numeric**
- **categorical:**
  1. **nominal (Ex):** cardinal points, blood type
  2. **ordinal (Ex):** categories of a hurricane and difficulty of a video game
- **numeric:**
  1. **discrete (Ex):** population size of municipios in Puerto Rico
  2. **continuous (Ex):** a person's height and birth weight

Distribution function

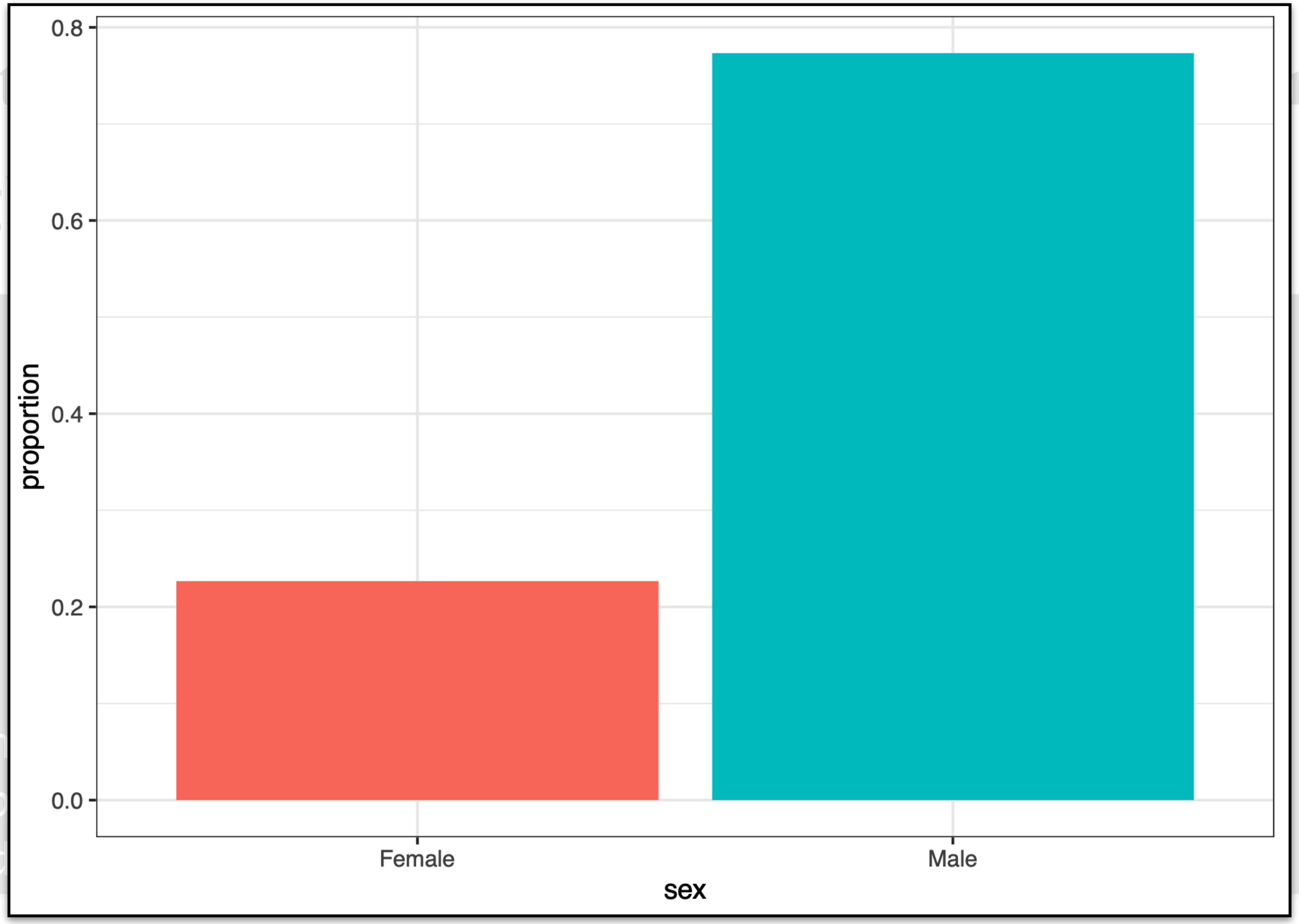
- The most basic, and perhaps most important, summary of data is its distribution
- For categorical data, the distribution corresponds to the proportion of each class
- Example with heights data:

```r
library(tidyverse)
library(dslabs)
data("heights")
ds_theme_set()

heights %>%
   group_by(sex) %>%
   summarize(proportion = n()/nrow(.)) %>%
   ggplot(aes(sex, proportion, fill=sex)) +
   geom_col(show.legend = FALSE)
```

Distribution function

- The most basic, and perhaps most important, summary of data is its distribution
- For categorical dat rtion of each class
- Example with heig

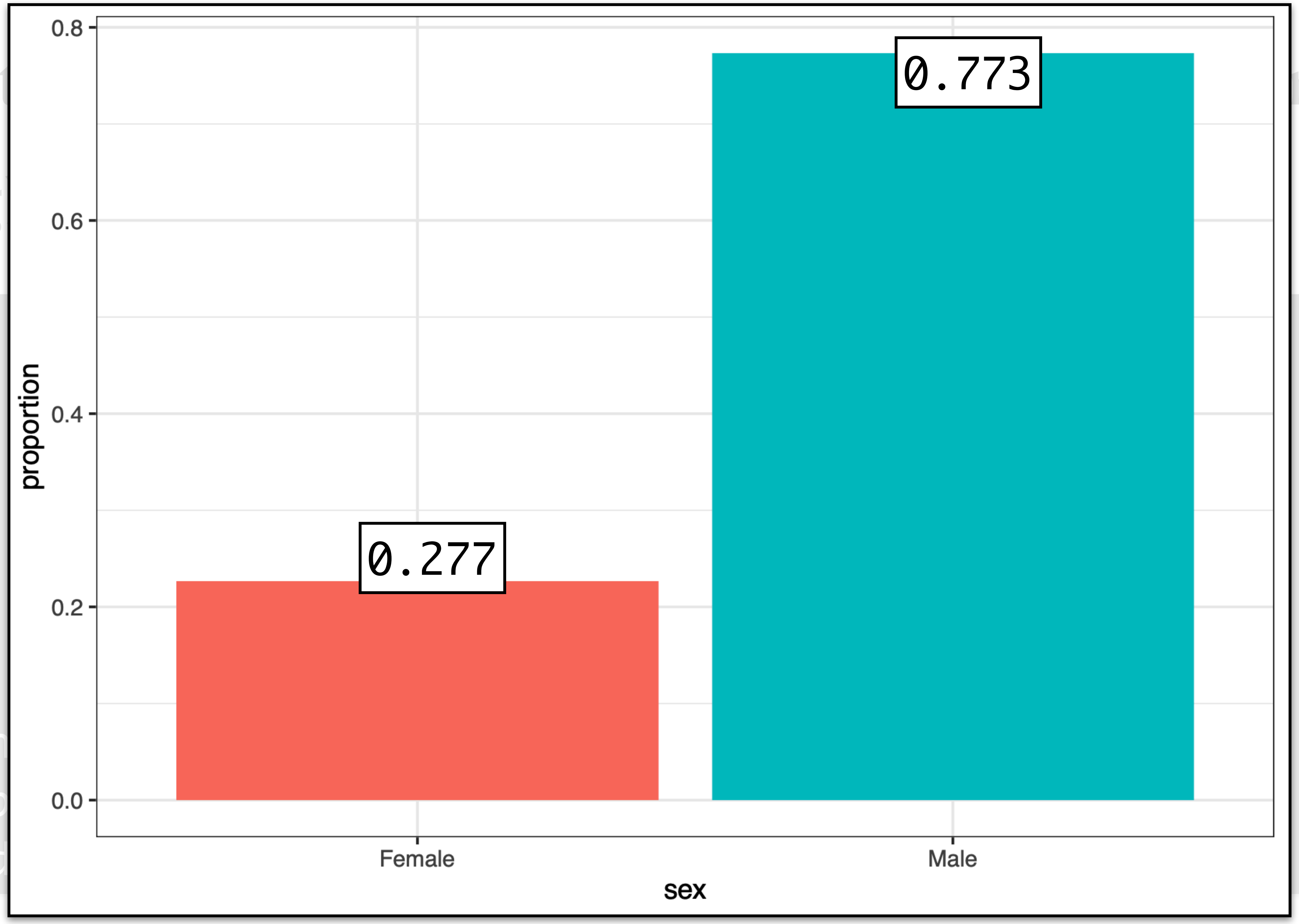
```
library(tidyverse)
library(dslabs)
data("heights")
ds_theme_set()

heights %>%
   group_by(sex) %>%
   summarize(proportion
   ggplot(aes(sex, prop
   geom_col(show.legend
```



Distribution function

- The most basic, and perhaps most important, summary of data is its distribution
- For categorical dat… rtion of each class
- Example with heig…

```
library(tidyverse)
library(dslabs)
data("heights")
ds_theme_set()

heights %>%
  group_by(sex) %>%
  summarize(proportion
  ggplot(aes(sex, prop
  geom_col(show.legenc
```



Distribution function

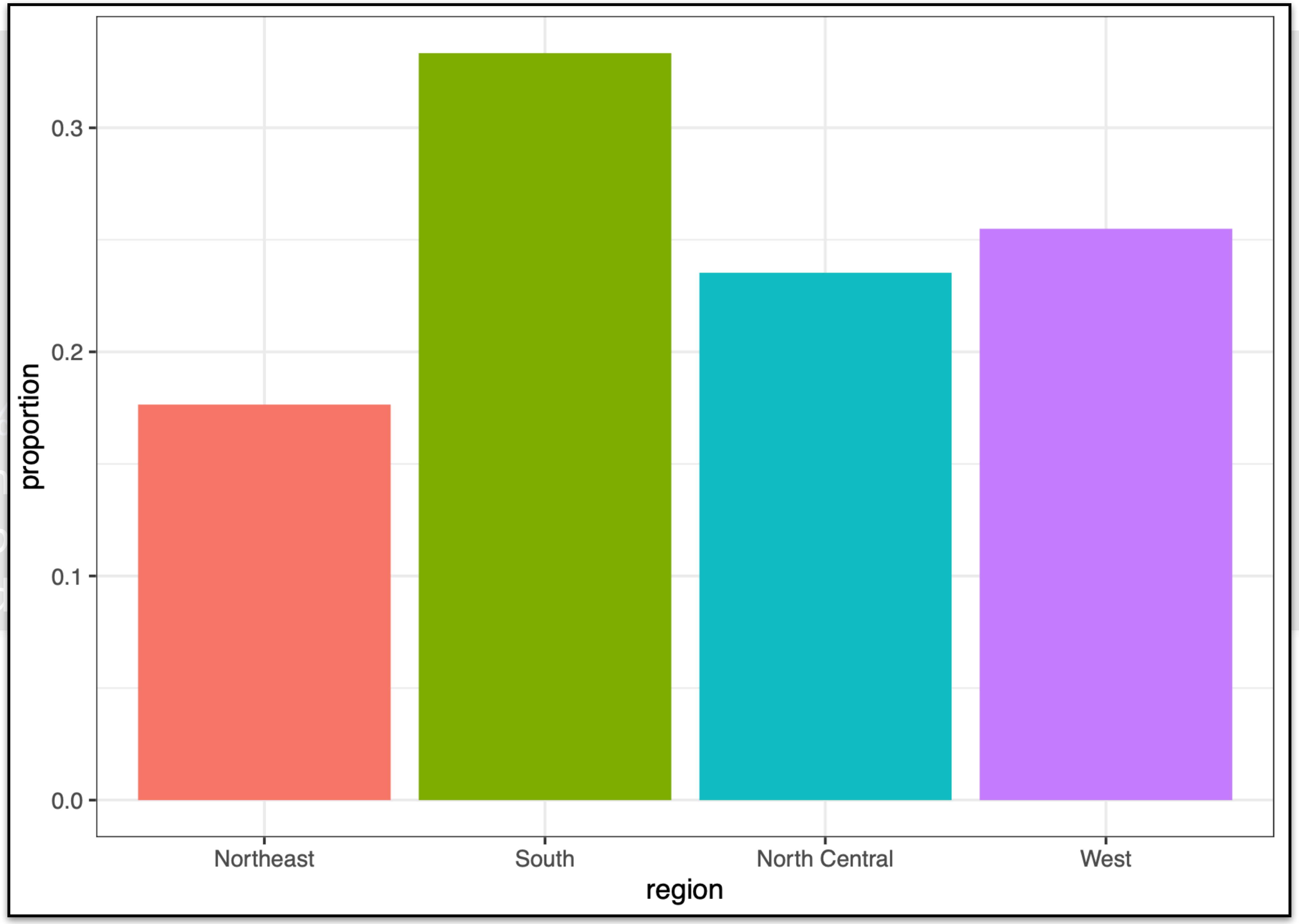
- Another example with the murders dataset:

```
library(tidyverse)
library(dslabs)
data("murders")
ds_theme_set()

murders %>%
   group_by(region) %>%
   summarize(proportion = n()/nrow(.)) %>%
   ggplot(aes(region, proportion, fill=region)) +
   geom_col(show.legend = FALSE)
```

Distribution function

- Another example with the murders dataset:

```
library(tidyverse)
library(dslabs)
data("murders")
ds_theme_set()

murders %>%
   group_by(region) %>%
   summarize(proportion
   ggplot(aes(region, p
   geom_col(show.legend
```



Distribution function

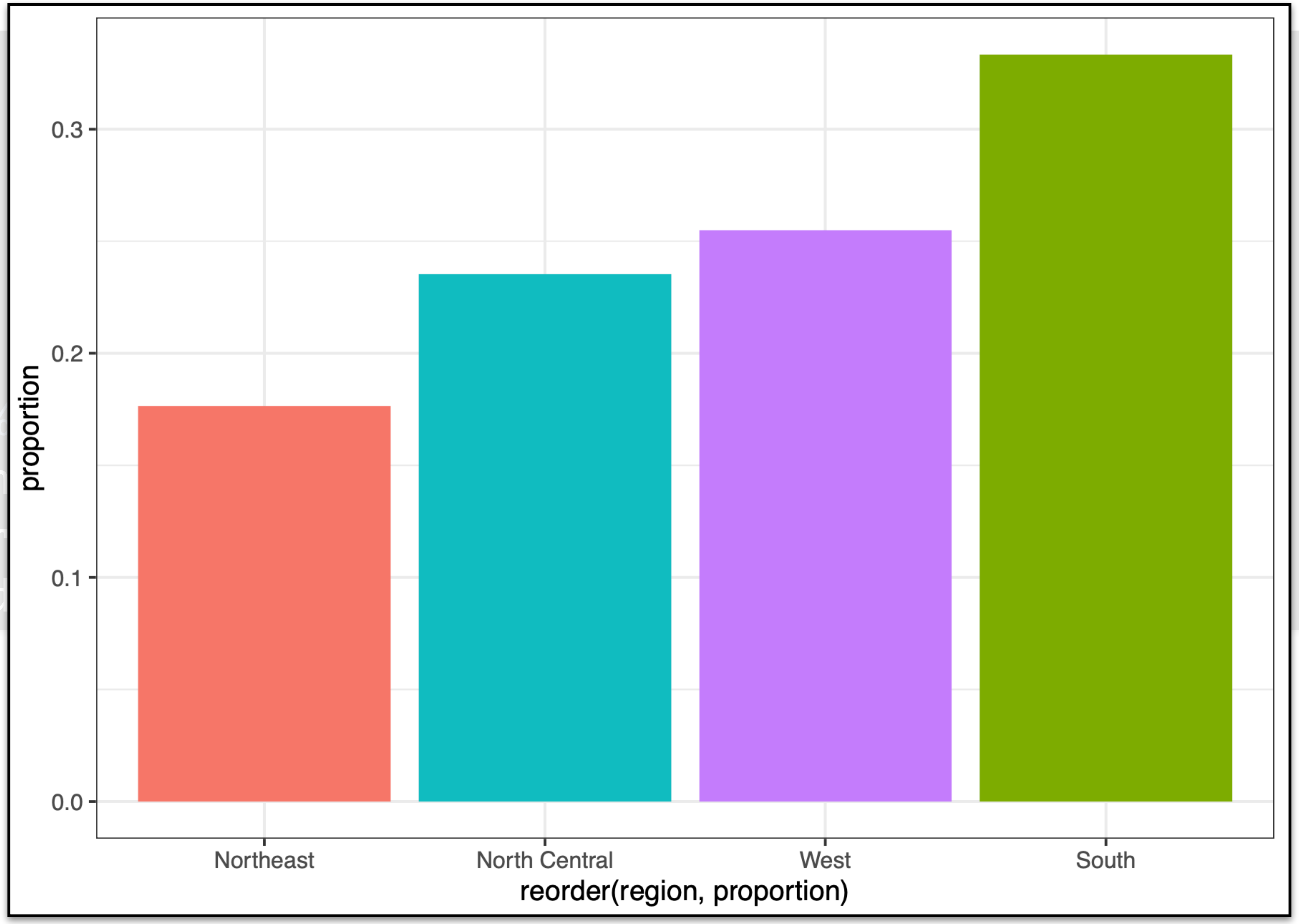
- Let's add a small tweak

```
library(tidyverse)
library(dslabs)
data("murders")
ds_theme_set()

murders %>%
   group_by(region) %>%
   summarize(proportion = n()/nrow(.)) %>%
   ggplot(aes(reorder(region, proportion), proportion, fill=region)) +
   geom_col(show.legend = FALSE)
```

Distribution function

- Let's add a small tweak

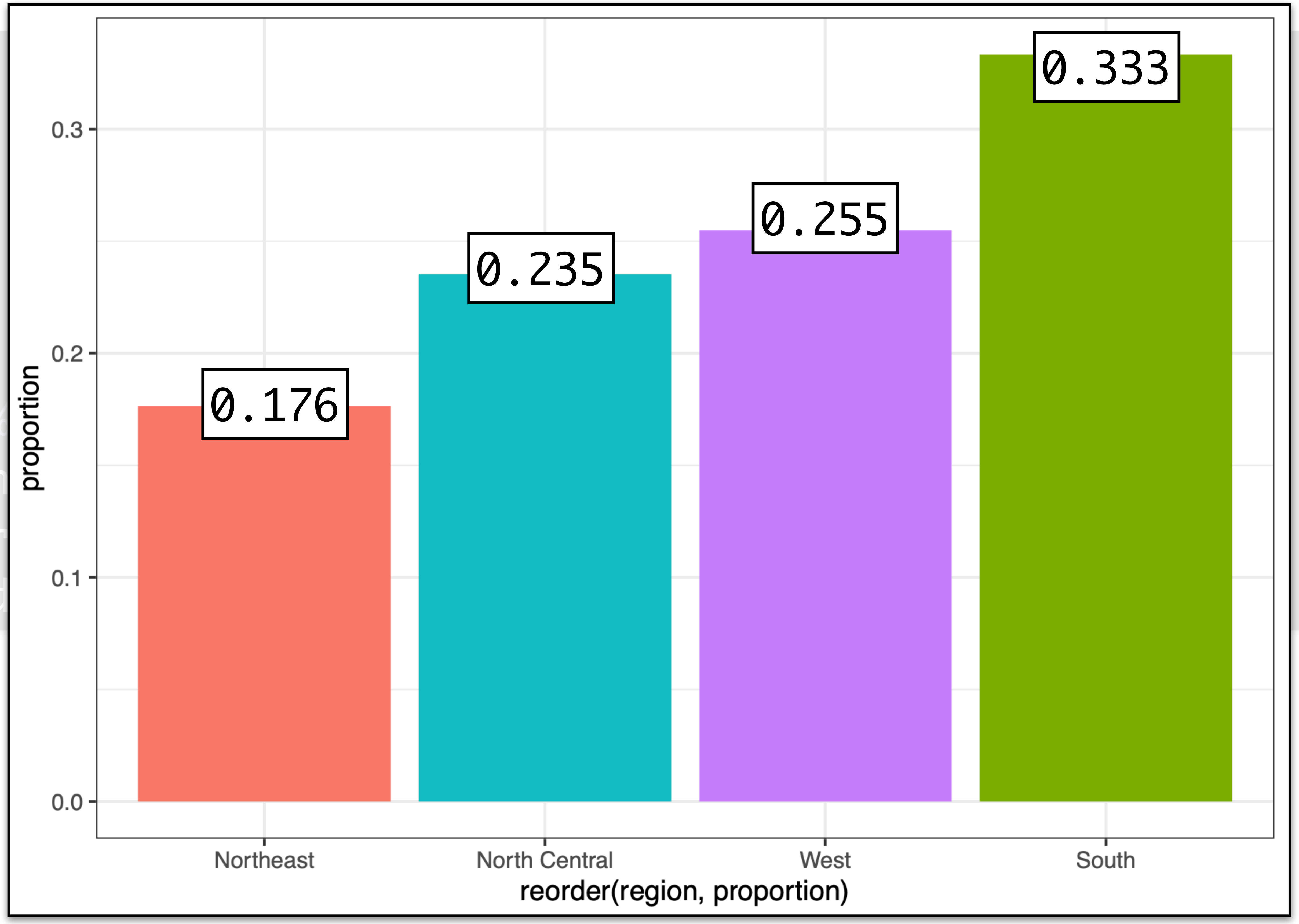
```
library(tidyverse)
library(dslabs)
data("murders")
ds_theme_set()

murders %>%
   group_by(region) %>%
   summarize(proportion
   ggplot(aes(reorder(r
   geom_col(show.legend
```



Distribution function

- Let's add a small tweak

```
library(tidyverse)
library(dslabs)
data("murders")
ds_theme_set()

murders %>%
  group_by(region) %>%
  summarize(proportion
  ggplot(aes(reorder(r
  geom_col(show.legend
```



- Note that this figure only show us four numbers
- A frequency table may be better

Cumulative distribution function

- The cumulative distribution function (CDF) is applicable only to ordered data and its define as the proportion of data below some value *a*:

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

- Example: CDF of male heights

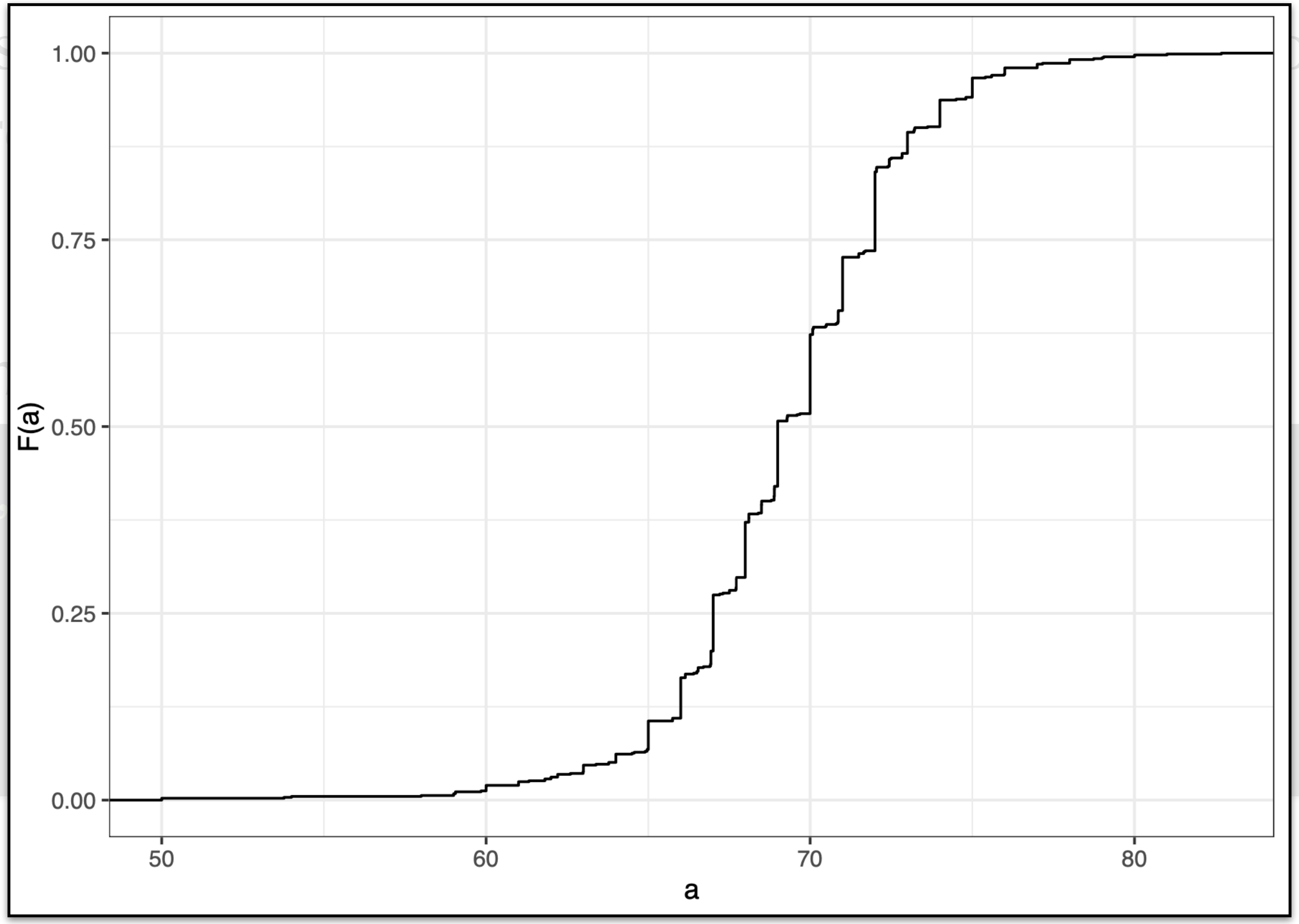
```
heights %>%
   filter(sex == “Male”) %>%
   ggplot(aes(height)) +
   stat_ecdf() +
   xlab(“a”) +
   ylab(“F(a)”)
```

Cumulative distribution function

- The cumulative dis… o ordered data and its define as the pr…

- Example: CDF of m…

```
heights %>%
  filter(sex == “Male”
  ggplot(aes(height))
  stat_ecdf() +
  xlab(“a”) +
  ylab(“F(a)”)
```

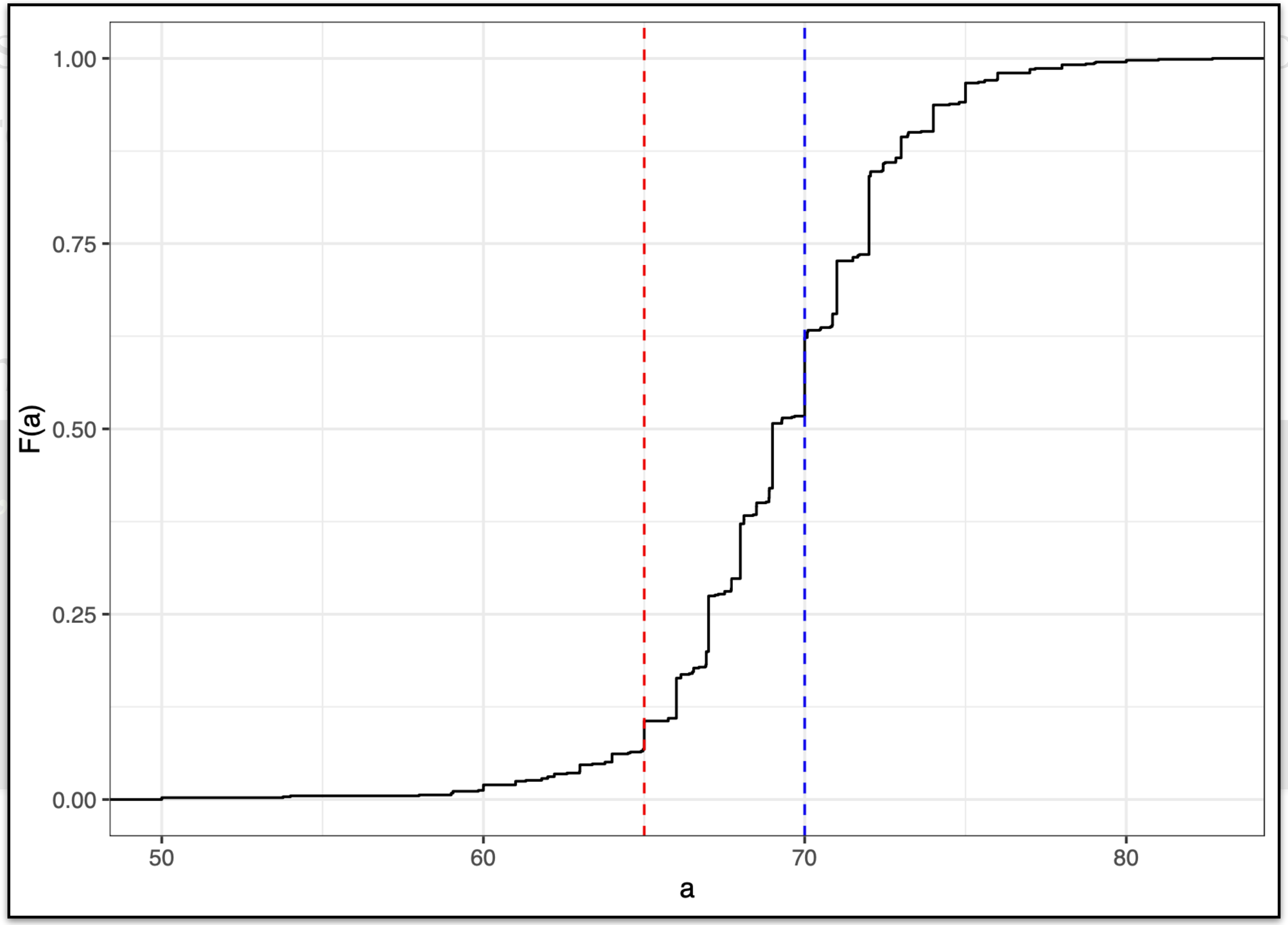


Cumulative distribution function

- The cumulative dis… ordered data and its define as the pr…

- Example: CDF of r…

```
heights %>%
  filter(sex == “Male”
  ggplot(aes(height))
  stat_ecdf() +
  xlab(“a”) +
  ylab(“F(a)”)
```



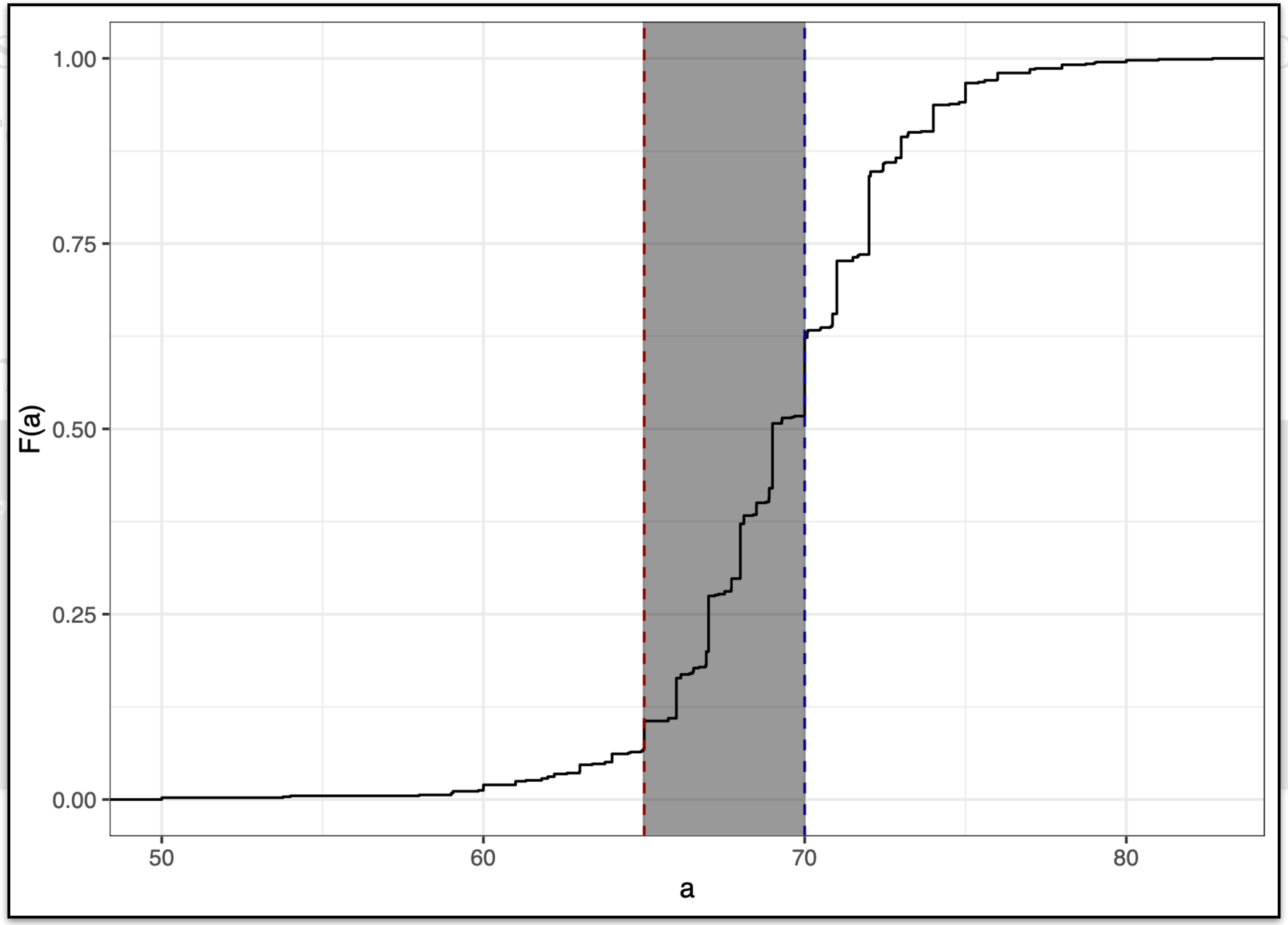
- $F(65) = 0.106$: This implies that 10.6% of male student’s height is less than 65in
- $F(70) = 0.623$: This implies that 62.3% of male student’s height is less than 70in

Cumulative distribution function

- The cumulative dis… ordered data and its define as the pr…

- Example: CDF of m…

```
heights %>%
   filter(sex == "Male"…
   ggplot(aes(height))…
   stat_ecdf() +
   xlab("a") +
   ylab("F(a)")
```



- $F(70)$ - $F(65)$ = `0.517`: This implies that 51.7% of male student's height is between 65 and 70 inches

Histograms

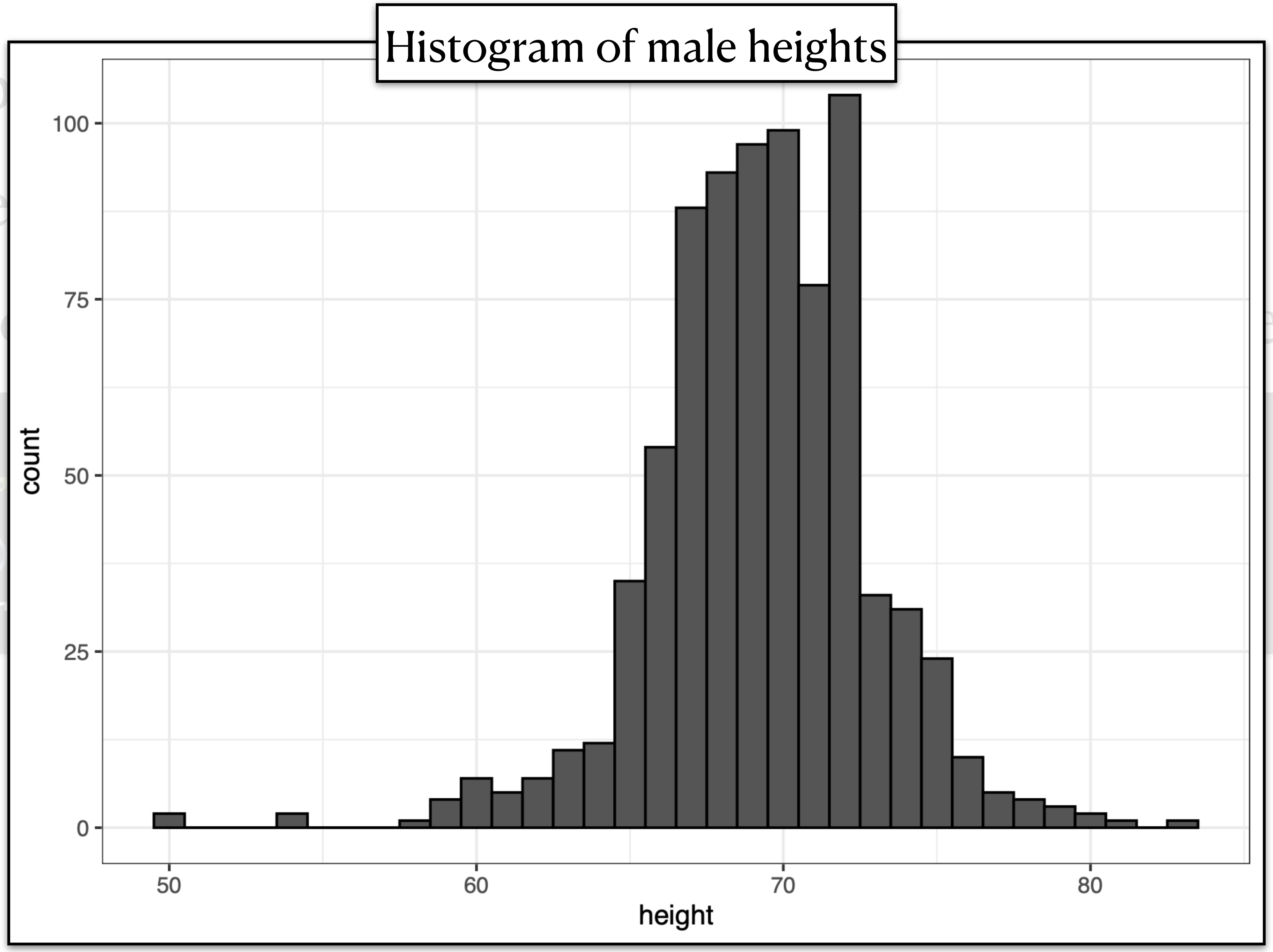
- A histogram depicts a frequency distribution of numeric data
- The *x-axis* is divided into non-overlapping bins of the same size
- The *y-axis* corresponds to the number of values that fall within each bin

```
heights %>%
   filter(sex == “Male”) %>%
   ggplot(aes(x=height)) +
   geom_histogram(color="black", binwidth = 1)
```

Histograms

- A histogram is dep
- The *x-axis* is divide
- The *y-axis* correspo … each bin

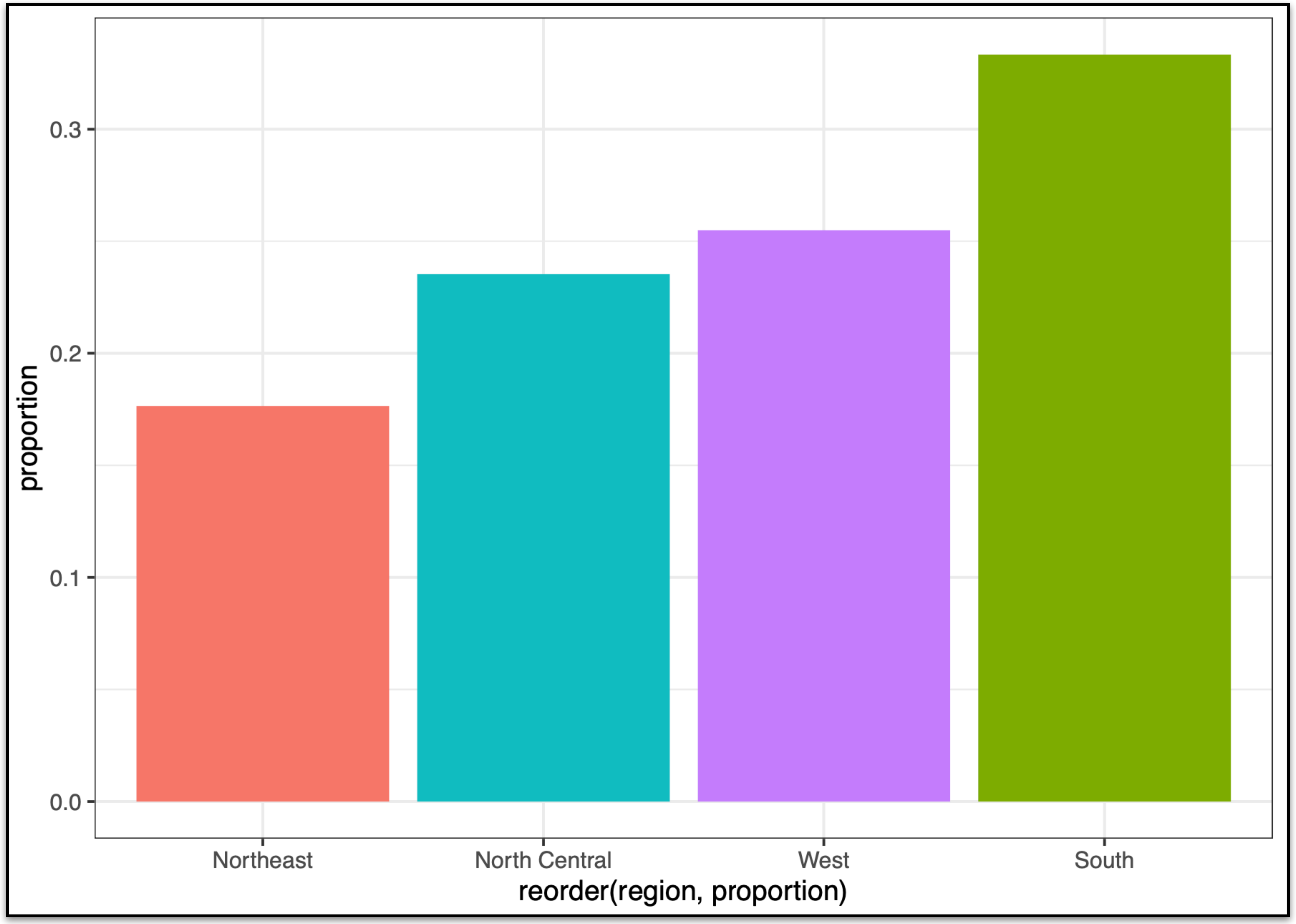
```
heights %>%
   filter(sex == "Male"
   ggplot(aes(x=height)
   geom_histogram(color
```

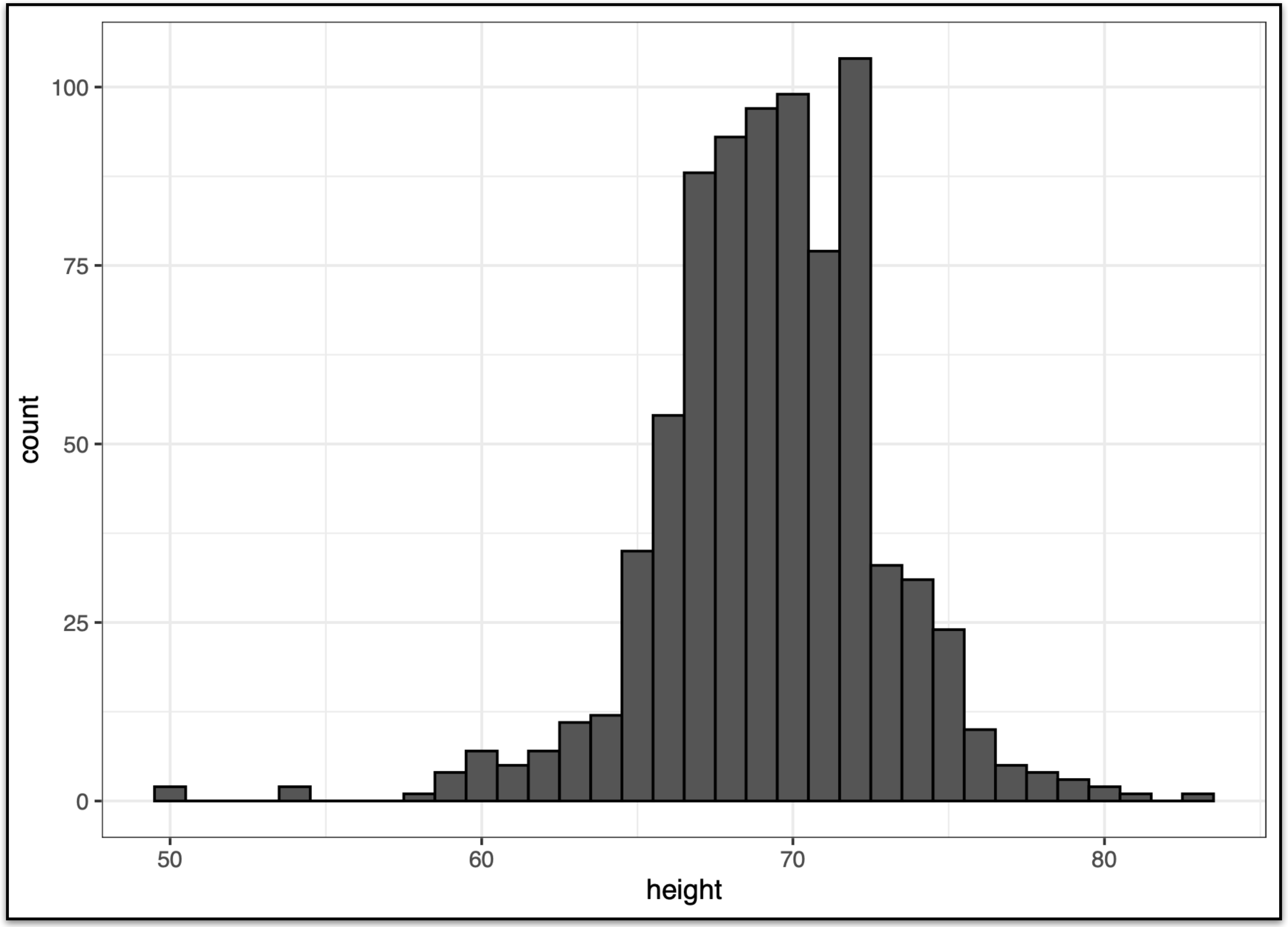


- The *x-axis* is split into 1 inch bins
- (49.5, 50.5], (50.5, 51.5], …, (82.5, 83.5]

Histograms

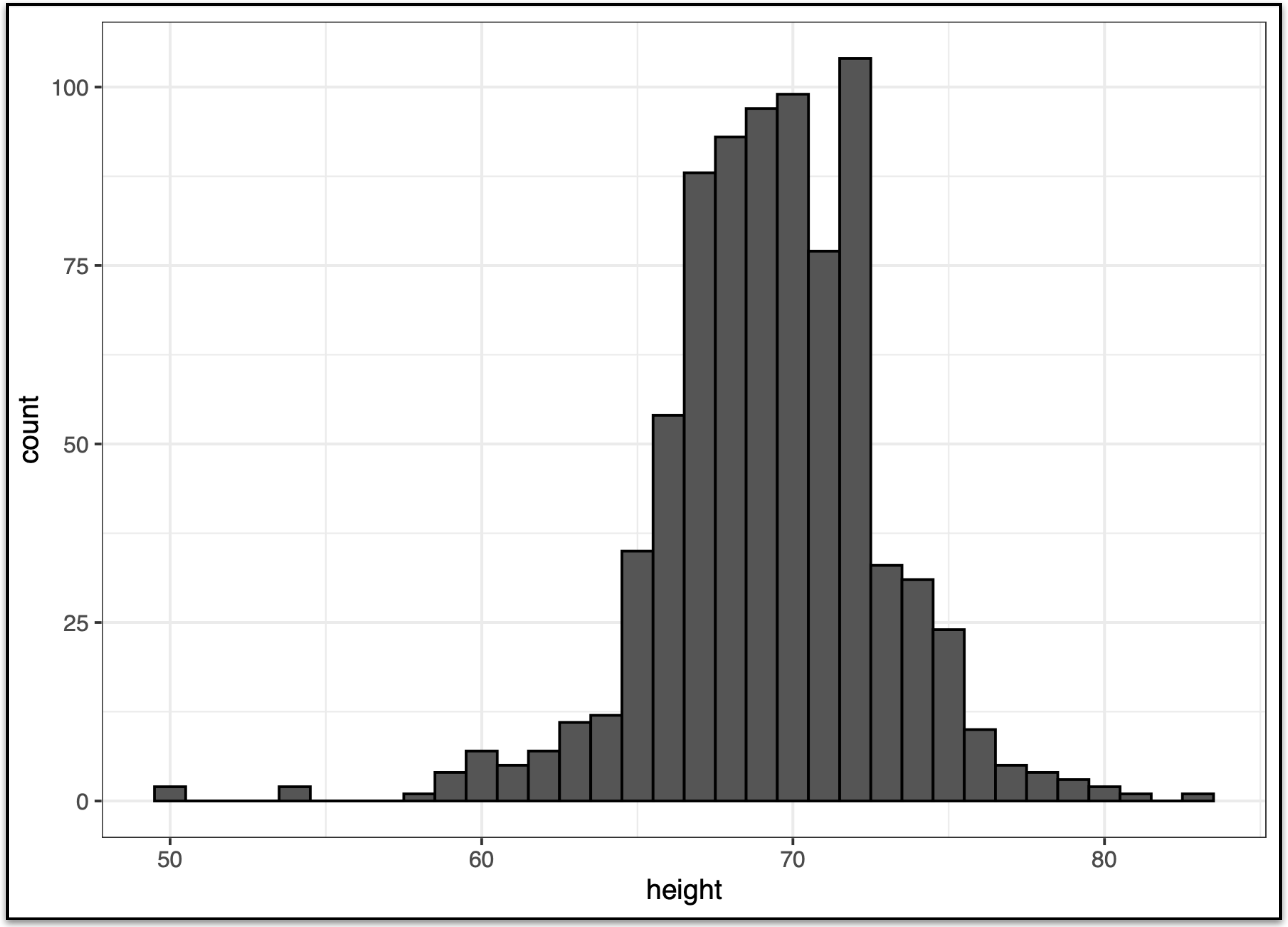
- Note that a histogram is similar to a barplot but the *x-axis* is numerical





Histograms

- From this we see:
  - Range: [50in, 84in]
  - Most of the data is between 63in and 75in
  - The distribution is more or less symmetric around 69in



Smoothed density

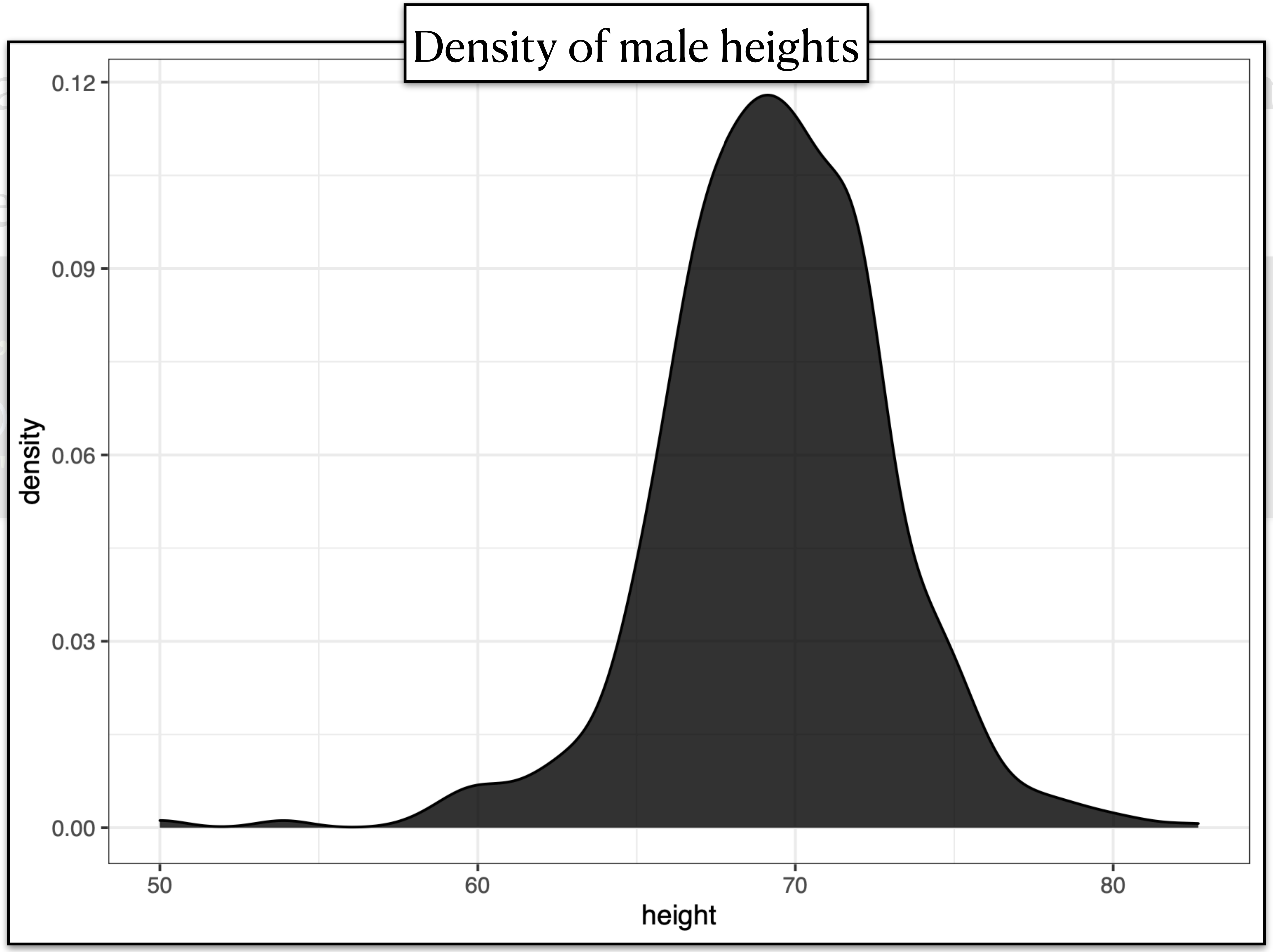
- Smooth densities are more aesthetically pleasing than histograms
- Here is an example:

```
heights %>%
   filter(sex == “Male”) %>%
   ggplot(aes(x=height)) +
   geom_density(fill="black", alpha = 0.80)
```

Smoothed density

- Smooth densities a ... ns
- Here is an example

```
heights %>%
   filter(sex == "Male"
   ggplot(aes(x=height)
   geom_density(alpha="
```



Smoothed density

- Smooth densities are more aesthetically pleasing than histograms
- Here is an example:

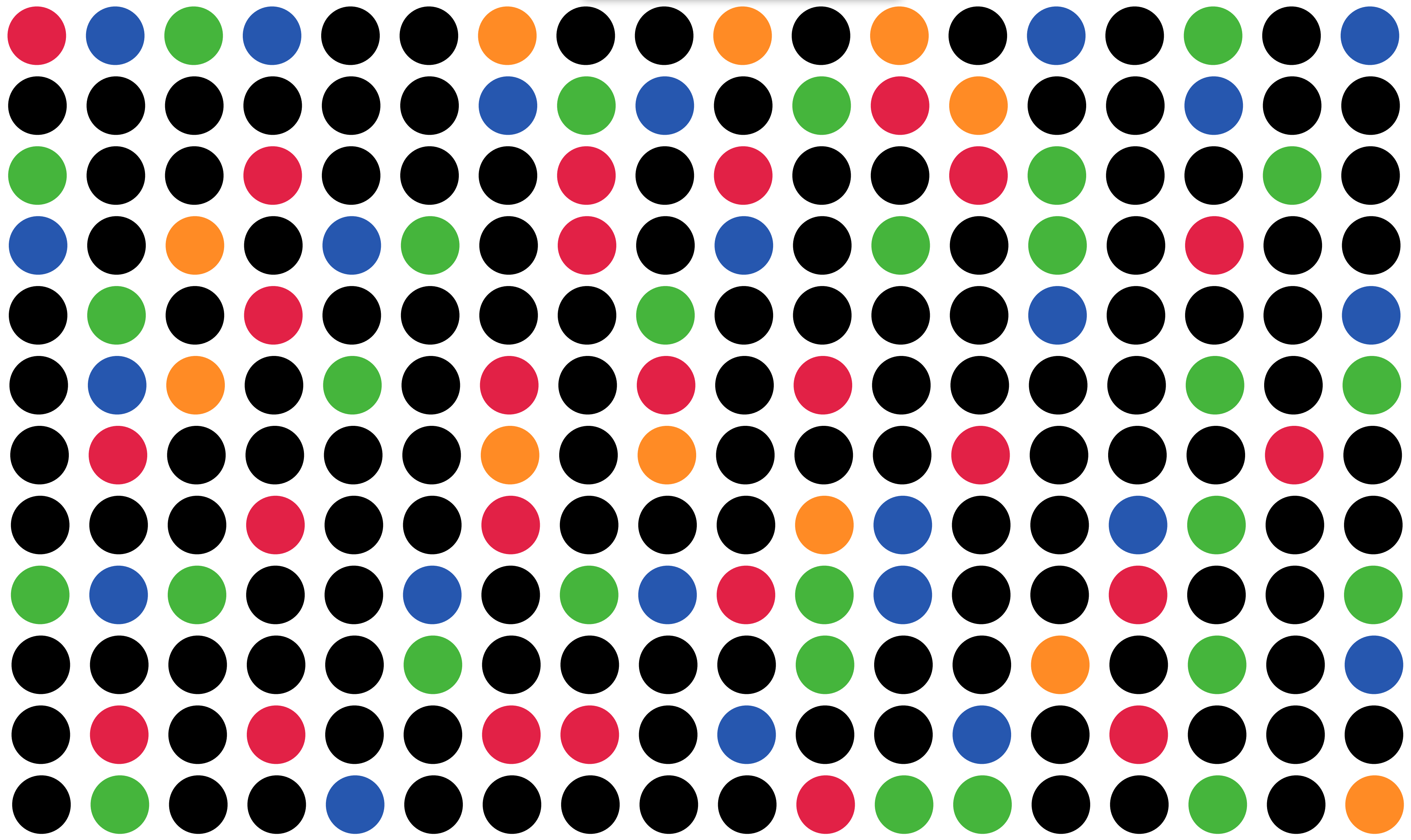
```
heights %>%
   filter(sex == “Male”) %>%
   ggplot(aes(x=height)) +
   geom_density(fill="black", alpha = 0.80)
```

- Note that the sharp edges at the interval boundaries are gone
- The local peaks are no more
- The *y-axis* changed from counts to density

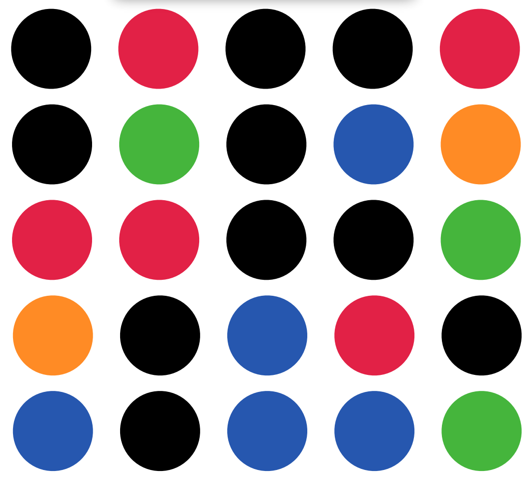
Understanding smooth densities

- We assume that our list of observed values is a subset of much larger list of unobserved values

Population

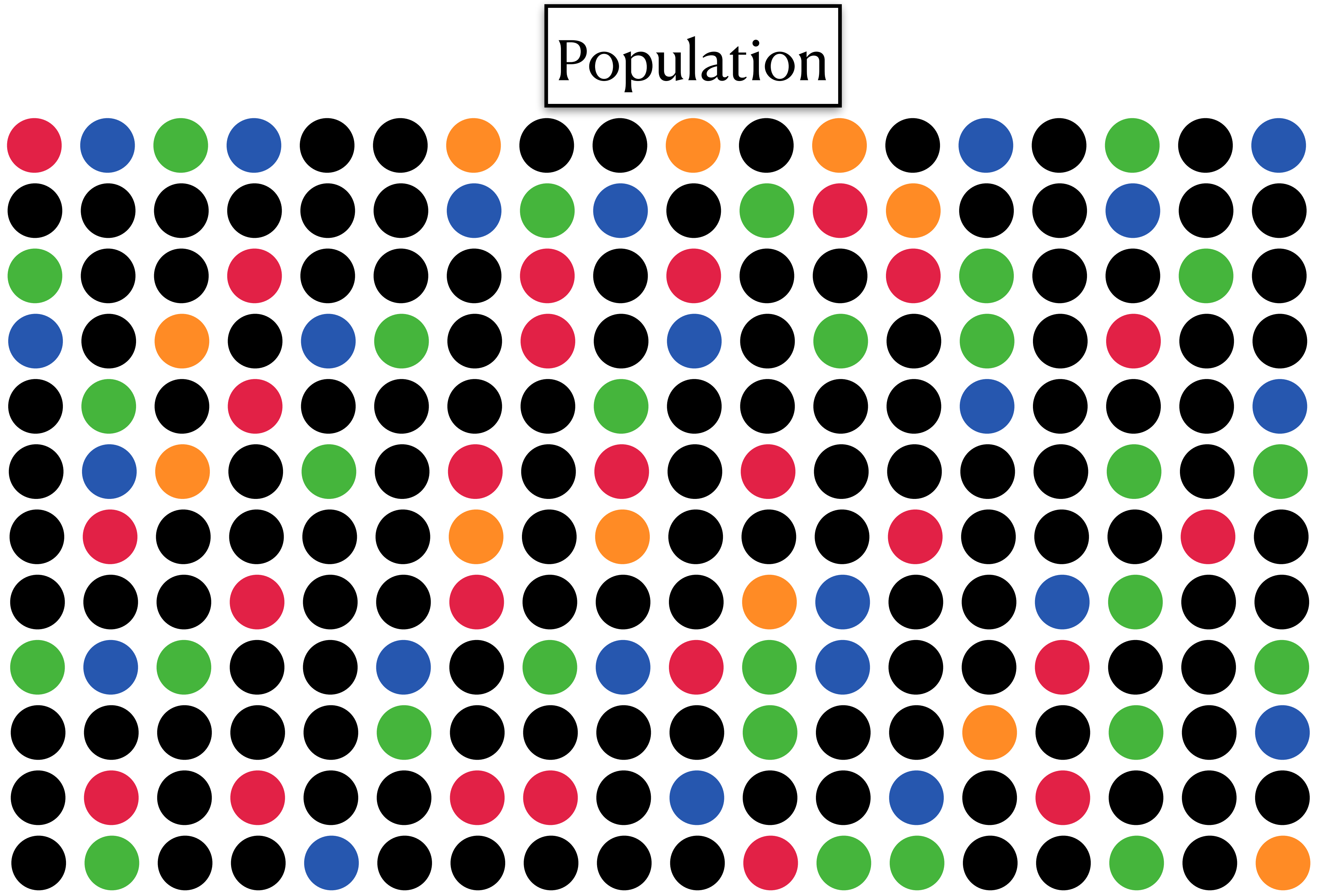


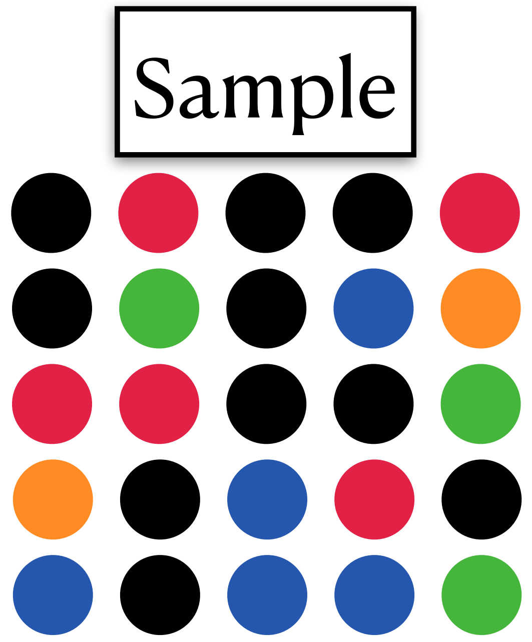
Sample



Understanding smooth densities

- We assume that our list of observed values is a subset of much larger list of unobserved values





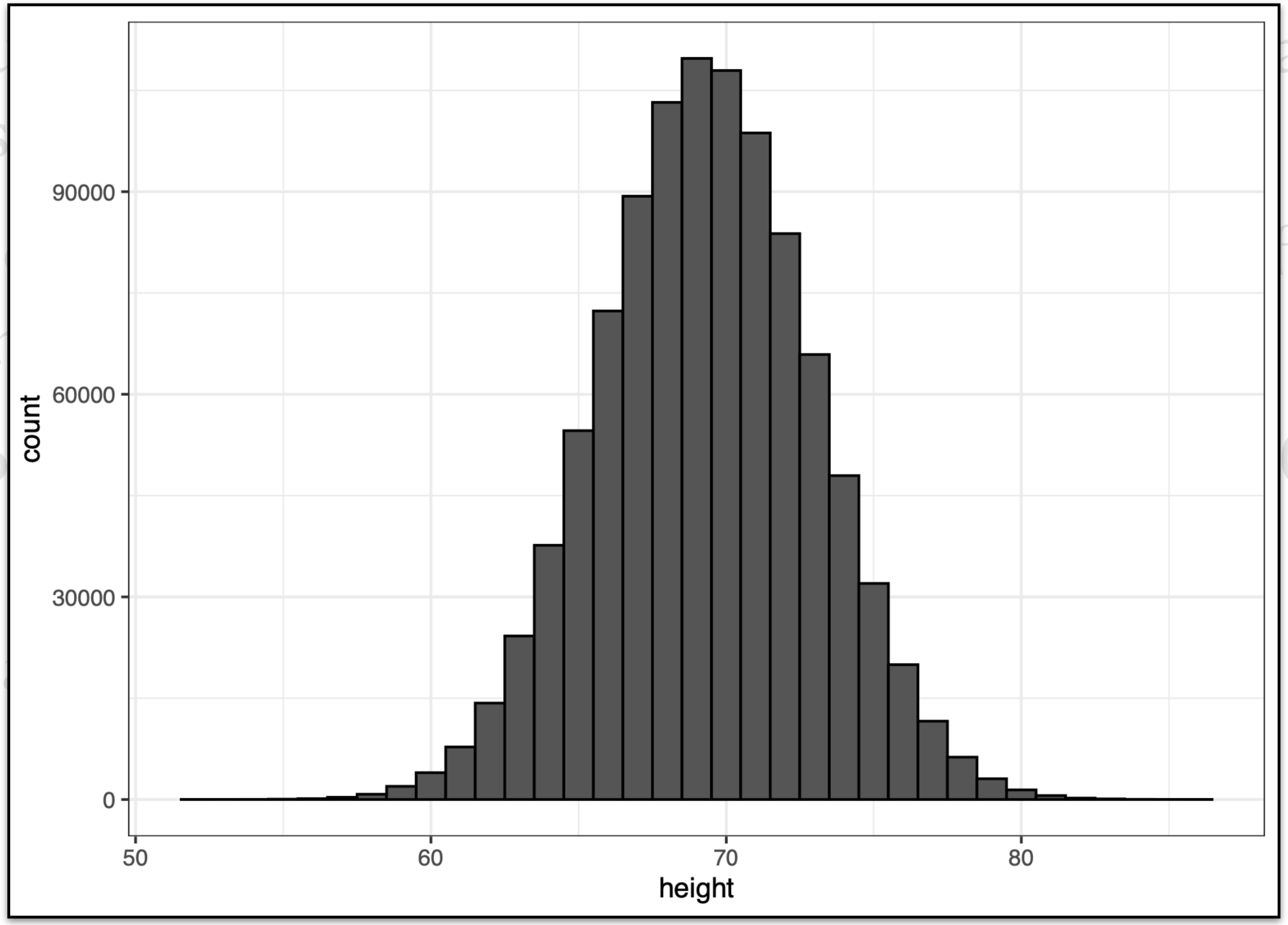
- For example, suppose that our list of 812 male students comes from a hypothetical population of all male students in the world

Understanding smooth densities

- We assume that our list of observed values is a subset of much larger list of unobserved values
- For example, suppose that our list of `812` male students comes from a hypothetical population of all male students in the world
- Specifically, suppose that the size of the population is `1,000,000` and that we have access to it
- Here is the histogram

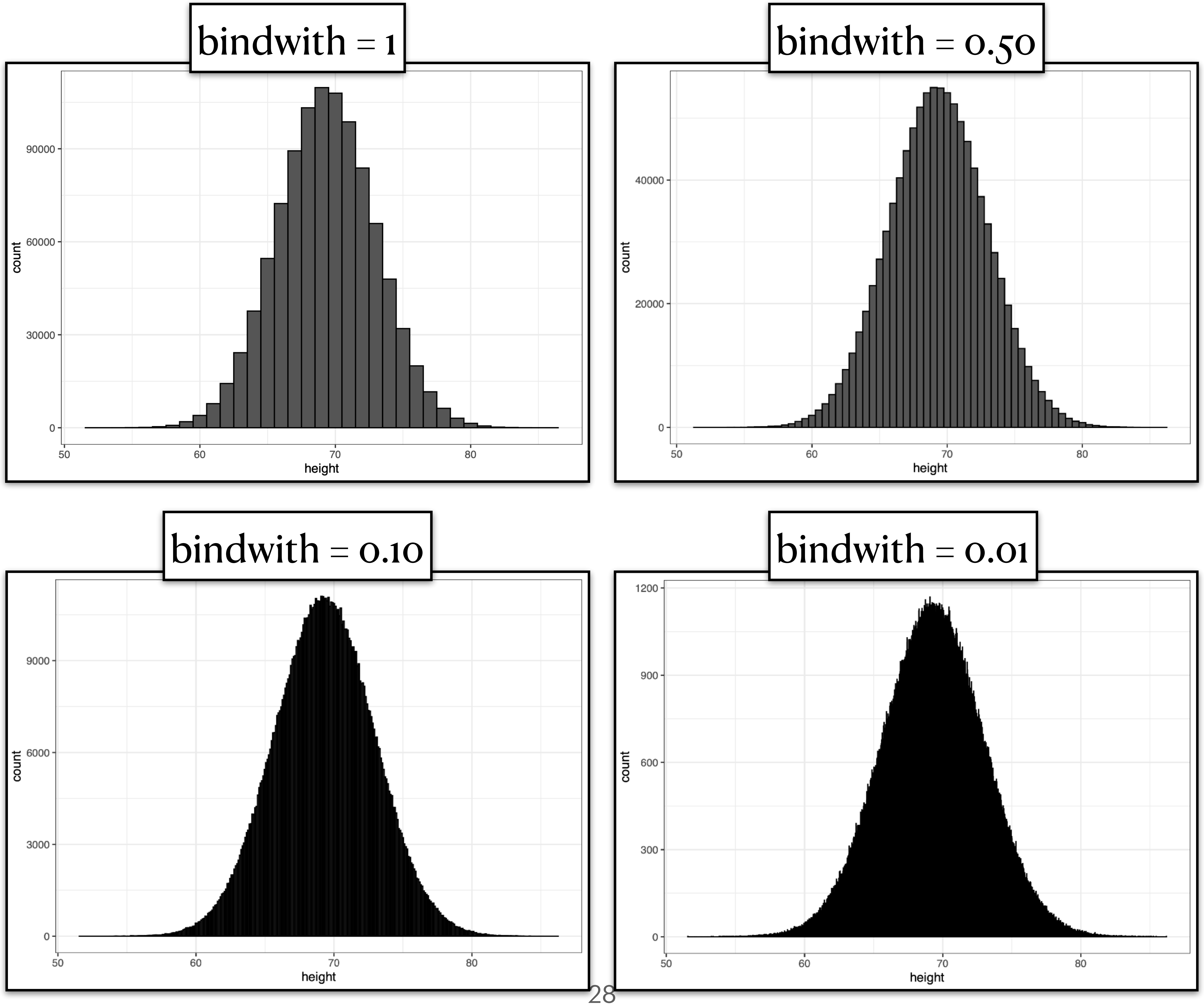
Understanding smooth densities

- We assume that ou arger list of unobserved values
- For example, supp from a hypothetical population of all m
- Specifically, suppo 00 and that we have access to it
- Here is the histogr



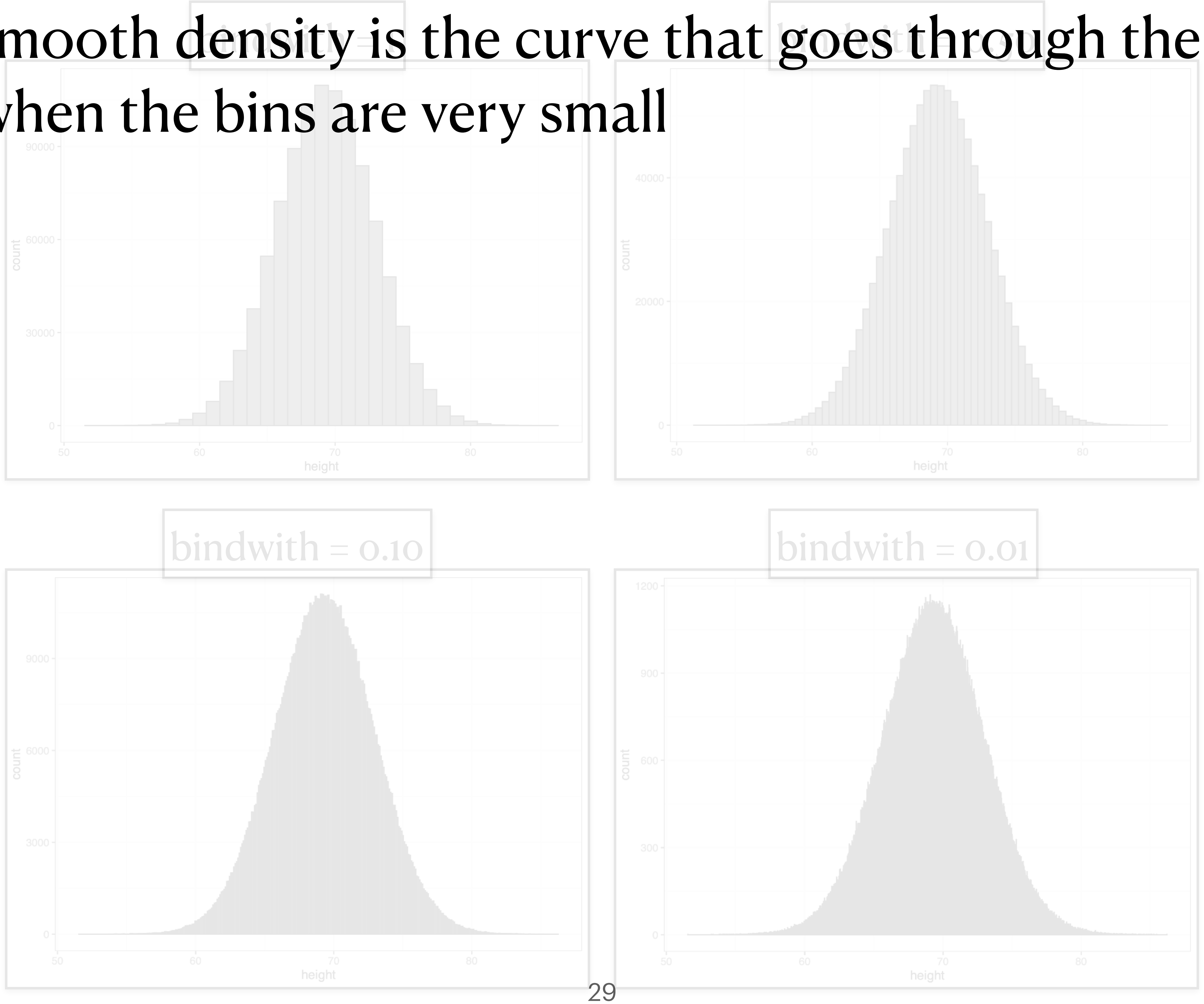
- Bins of 1 inch

Understanding smooth densities



Understanding smooth densities

- Essentially, the smooth density is the curve that goes through the top of the histogram bars when the bins are very small



Understanding smooth densities

- Now back to reality, all we have access to is the list of 812 male students

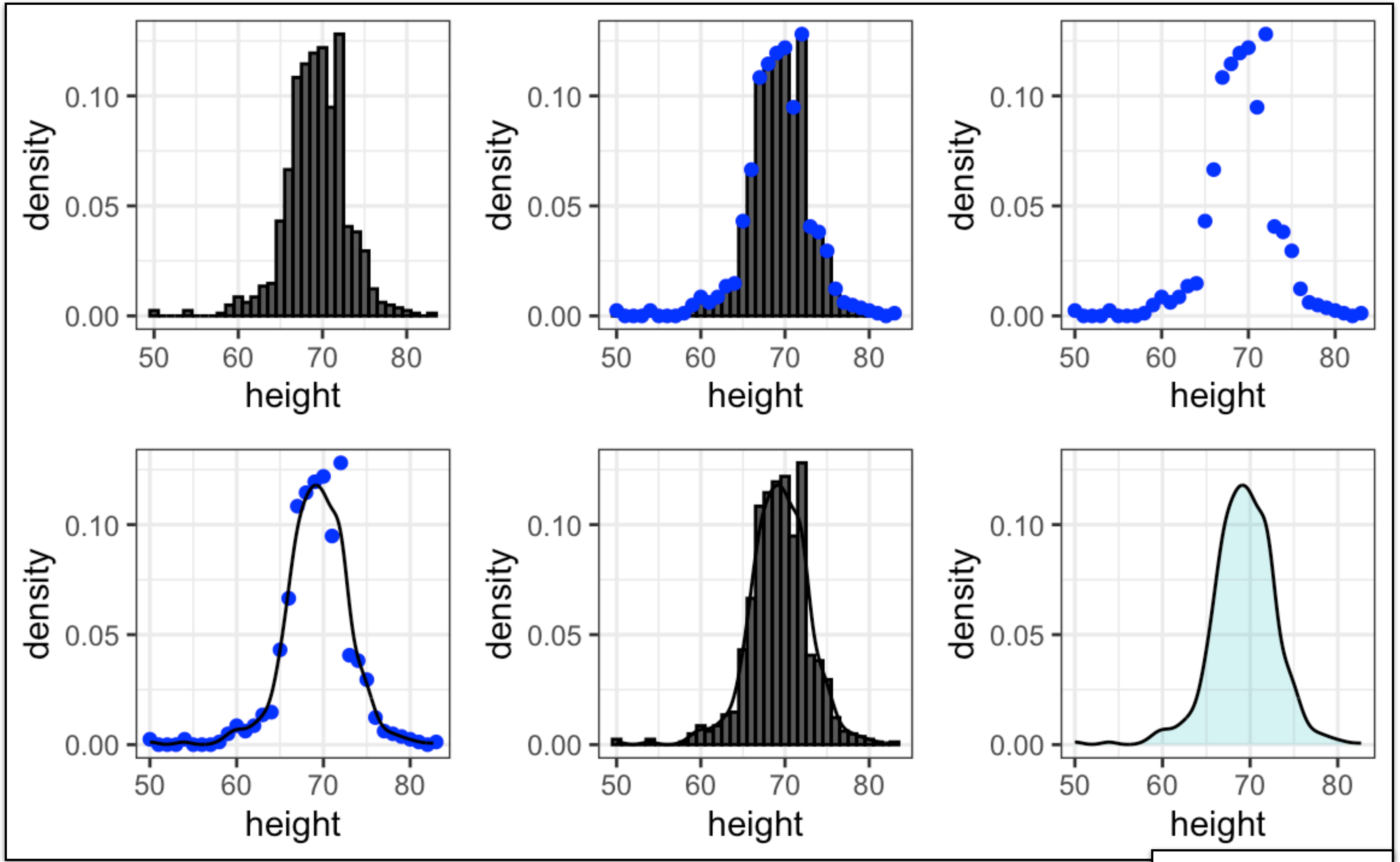


Figure taken from here

Understanding smooth densities

- However, *smooth* is relative term. We can control the smoothness of the curve

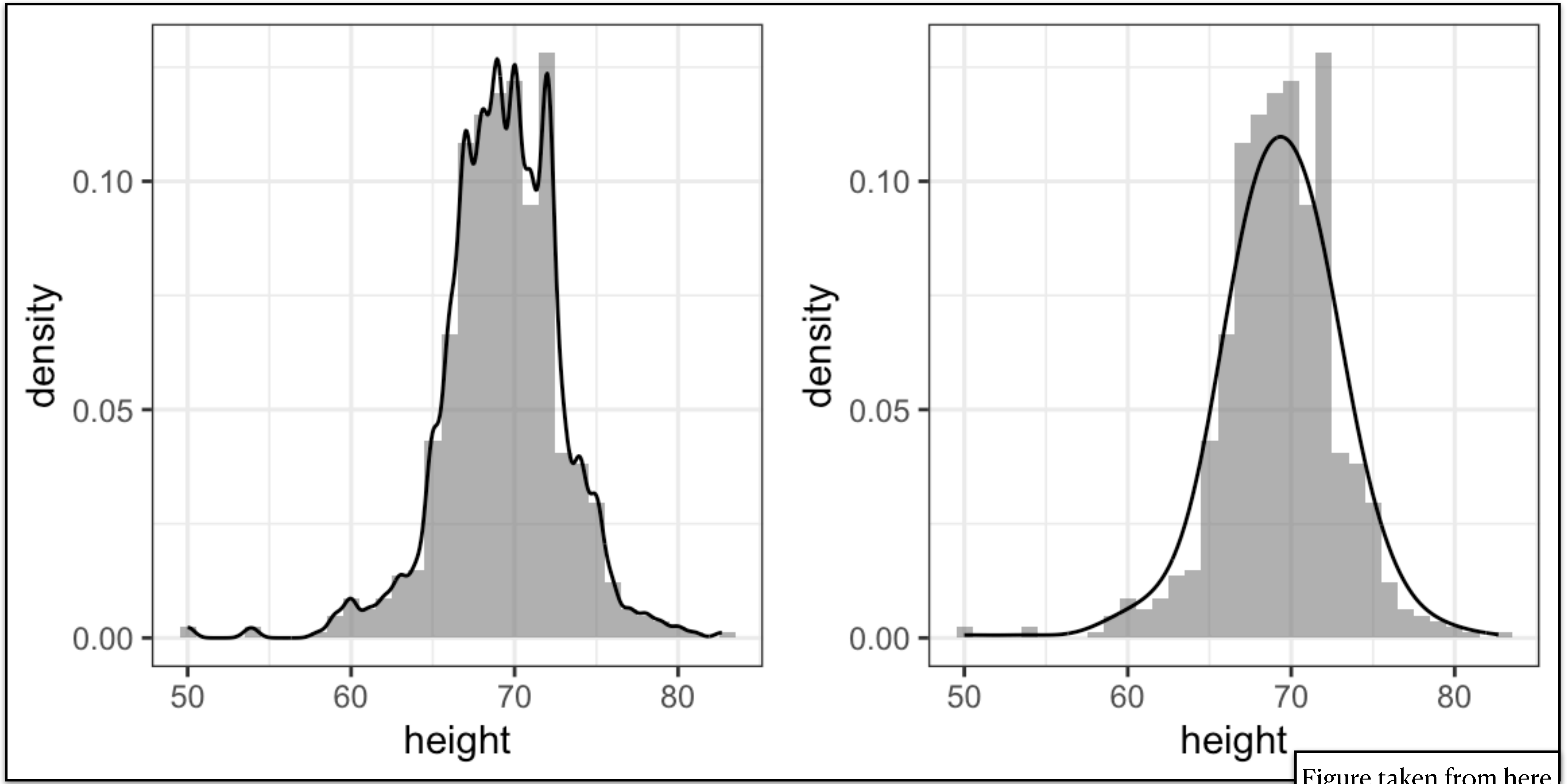


Figure taken from here

Smooth densities: The *y-axis*

- Densities are scaled so that the area under the curve is equal to 1
- To know the proportion of data in some interval [a, b] we have to compute the proportion of the total area that's inside the interval
- For example: the proportion between 65 and 68 is around 0.30

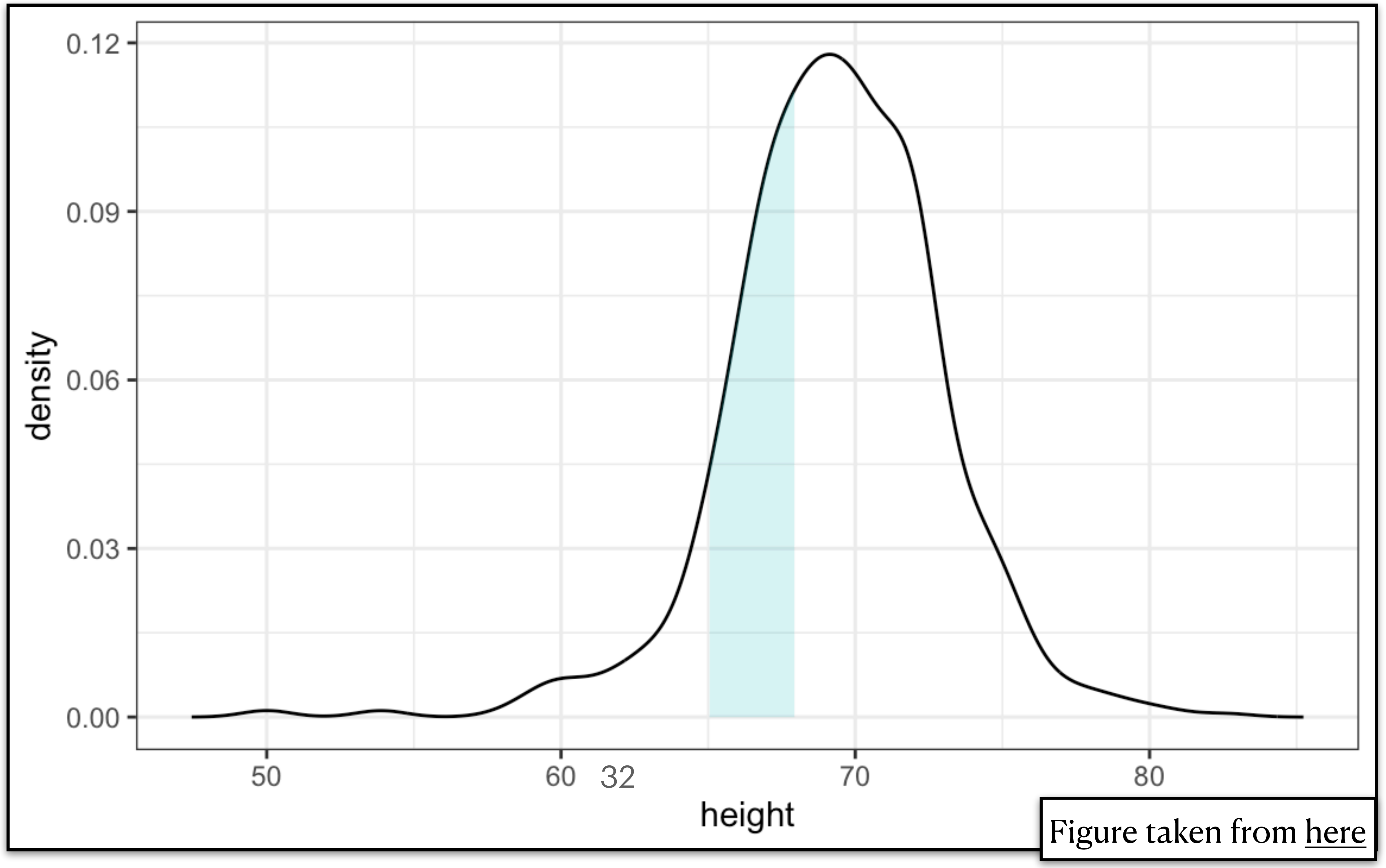


Figure taken from here

Smooth densities: Stratification

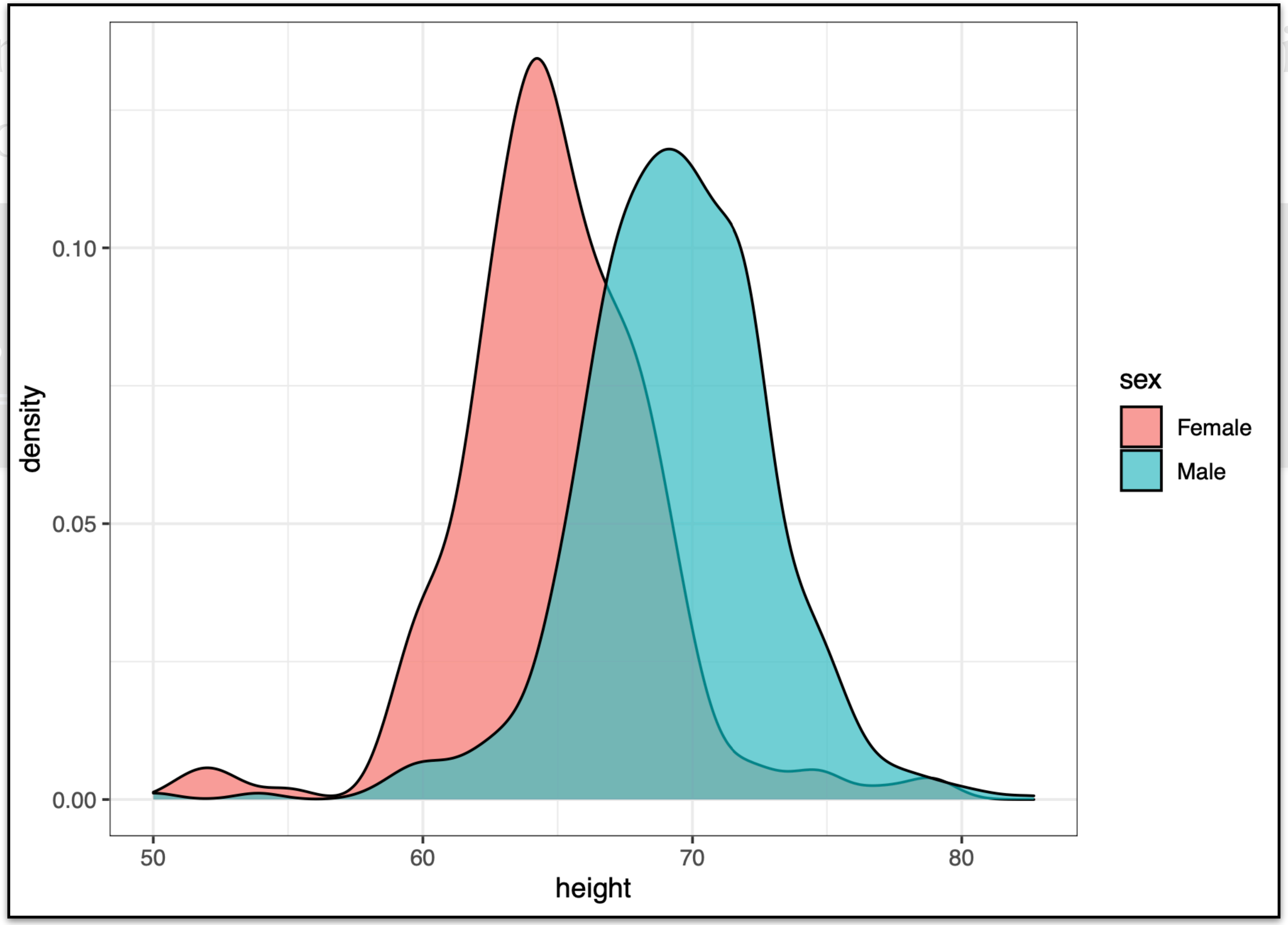
- One more appealing aspect of densities over histograms is that its easier to visualize multiple distributions

```
heights %>%
   ggplot(aes(x=height, fill=sex)) +
   geom_density(alpha = 0.80)
```

Smooth densities: Stratification

- One more appeali ... its easier to visualize multiple distributi

```
heights %>%
  ggplot(aes(x=height,
  geom_density(alpha =
```



The Normal distribution

- Histograms and densities provide excellent ways of summarizing data
- Can we further summarize our data? A two-number summary?
- The Normal distribution, also known as the bell curve or the Gaussian distribution, has the following density function:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

The Normal distribution

- Histograms and densities provide excellent ways of summarizing data
- Can we further summarize our data? A two-number summary?
- The Normal distribution, also known as the bell curve or the Gaussian distribution, has the following density function:

$$f(x) = \frac{1}{\underbrace{\sigma}_{\text{standard deviation}}\sqrt{2\pi}} \exp\left\{ -\frac{1}{2\sigma^2}(x - \underbrace{\mu}_{\text{mean}})^2 \right\}$$

- No need to memorize this, just note that this distribution is completely defined by two numbers

The Normal distribution

- Histograms and densities provide excellent ways of summarizing data
- Can we further summarize our data? A two-number summary?
- The Normal distribution, also known as the bell curve or the Gaussian distribution, has the following density function:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

mean

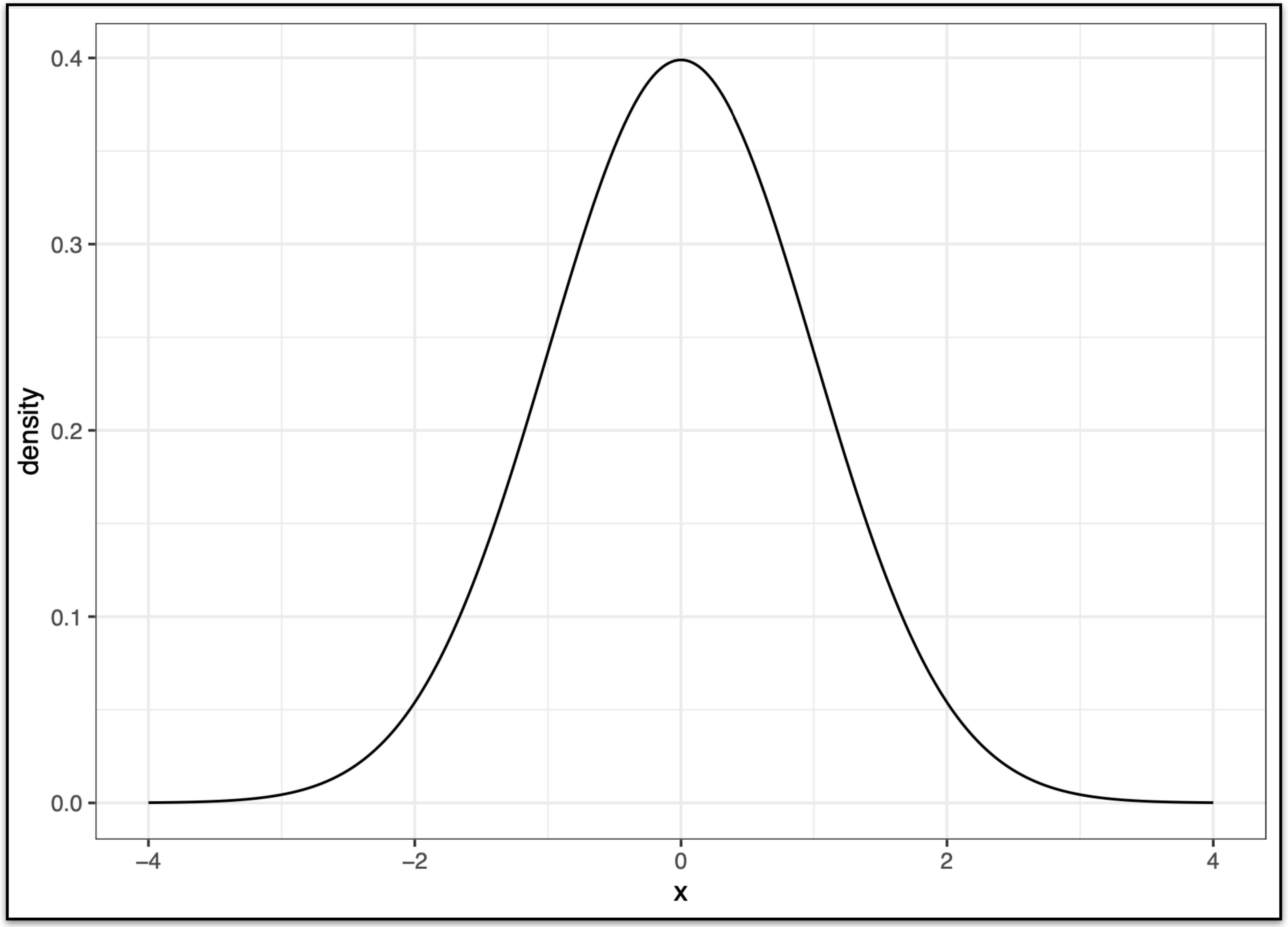
standard deviation

- We find the proportion between two numbers (a, b) with:

$$P(a < x < b) = \int_a^b \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

The Normal distribution

- Here is the density of a normal distribution with mean `0` and standard deviation `1`



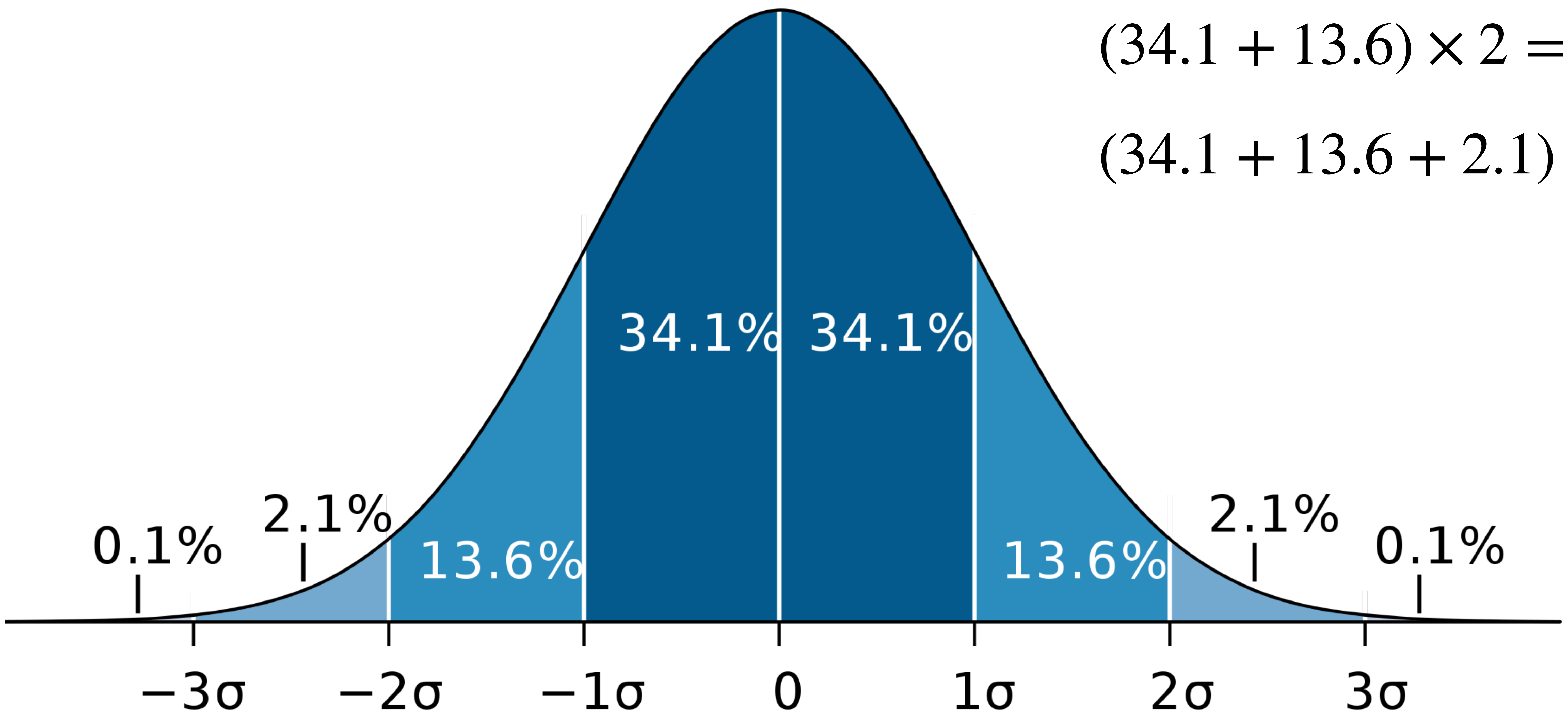
The Normal distribution

- Here is the density of a normal distribution with mean `0` and standard deviation `1`

$$34.1 \times 2 = 68.2$$

$$(34.1 + 13.6) \times 2 = 95.4$$

$$(34.1 + 13.6 + 2.1) \times 2 = 99.6$$



The Normal distribution

- Suppose that we have a list of data: $\{x_1, \ldots, x_n\}$
- Then we can compute the average with:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

- and we can compute the standard deviation with:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2}$$

- In R we can compute these quantities with:

```
mean(x)
sd(x)
```

The Normal distribution

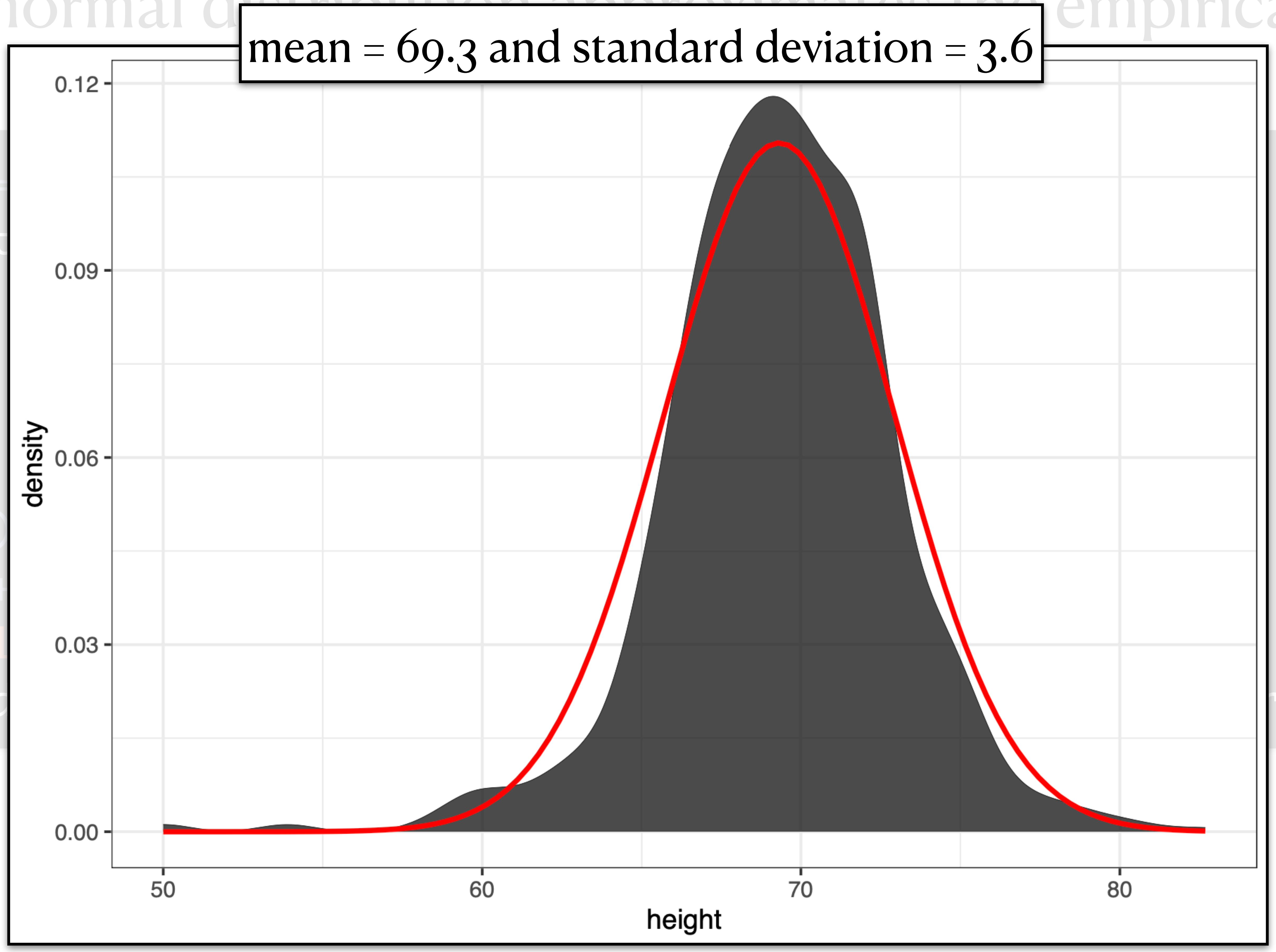
- Let's see how the normal distribution approximates the empirical distribution of male heights

```
index <- heights$sex == "Male"
x     <- heights$height[index]
m     <- mean(x)
s     <- sd(x)

heights %>%
  filter(sex == "Male") %>%
  ggplot(aes(height)) +
  geom_density(size=0.10, fill="black", alpha=0.70) +
  stat_function(fun = dnorm, size=1, args = list(mean = m, sd = s), color="red")
```

The Normal distribution

- Let's see how the normal d… empirical distribution of male heights

```
index <- heights$sex ==
x     <- heights$height
m     <- mean(x)
s     <- sd(x)

heights %>%
  filter(sex == "Male")
  ggplot(aes(height)) +
  geom_density(size=0.1
  stat_function(fun = c                                              ="red")
```



- The normal distribution seems to be a good approximation for male heights

Standard units

- We can talk in terms of *standard units* for data that is approximately normally distributed.
- The standard unit of a value tell us how many standard deviations away from the average it is.
- For a value x from a list of values $\{x_1, \ldots, x_n\}$ we define the value of x in standard units as:

$$z = \frac{x - \mu}{\sigma}$$

mean

standard deviation

Standard units

- Recall the density function of a normal distribution:

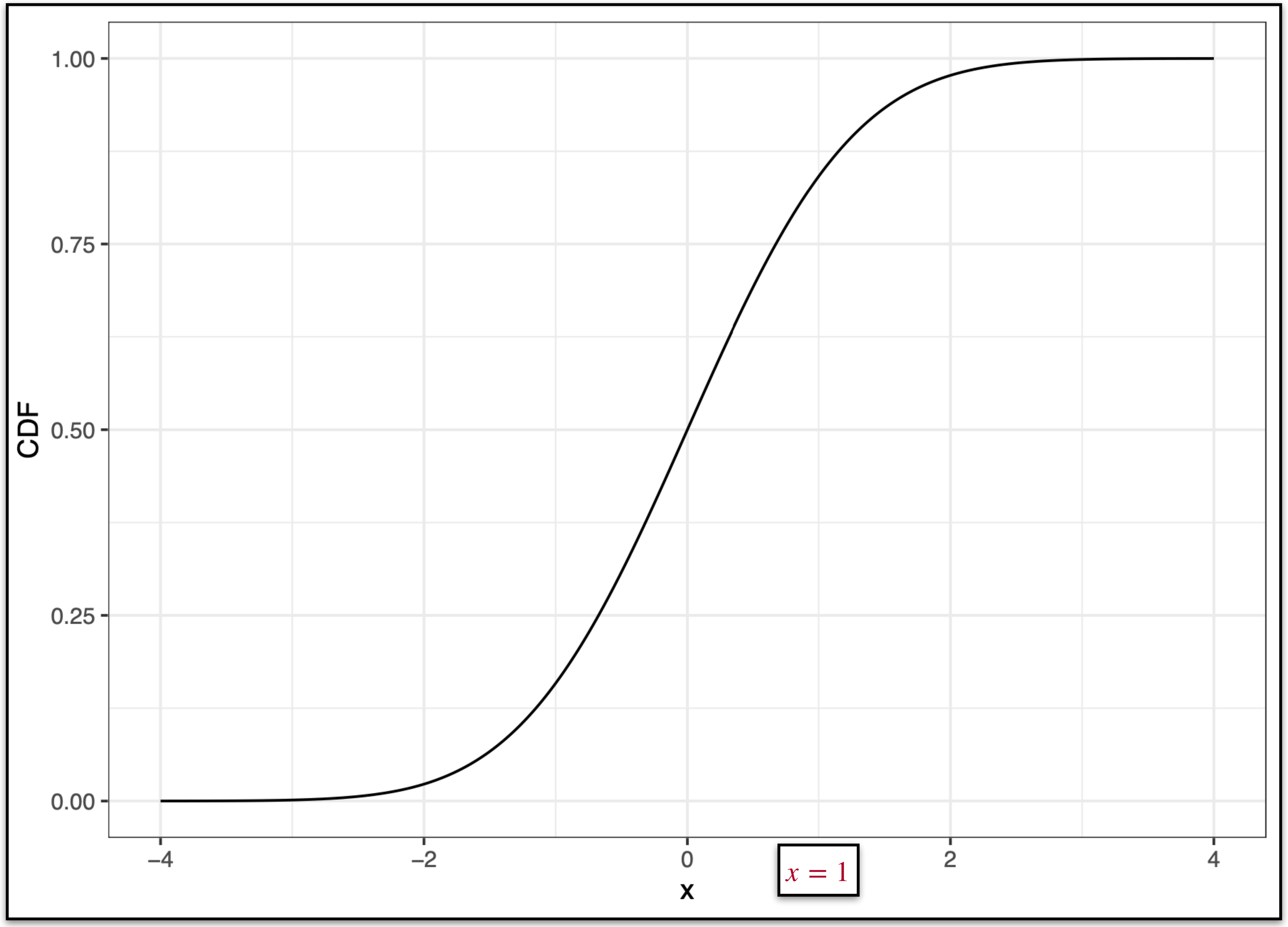
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}z^2\right\}$$

- The maximum of $f(x)$ is when $z = 0$ or $x = \mu$
  - The maximum occurs at the average
- Further, note that $-z^2/2$ is symmetric around 0
- Finally, standard units allow us to assess if an observation is about average ($z \approx 0$), very large or very small ($|z| \approx 2$), or an extremely race occurrence ($|z| > 3$)
- Note that this is true regardless of the original distribution of the data, as long as it is approximately normal.

Quantile-quantile plot

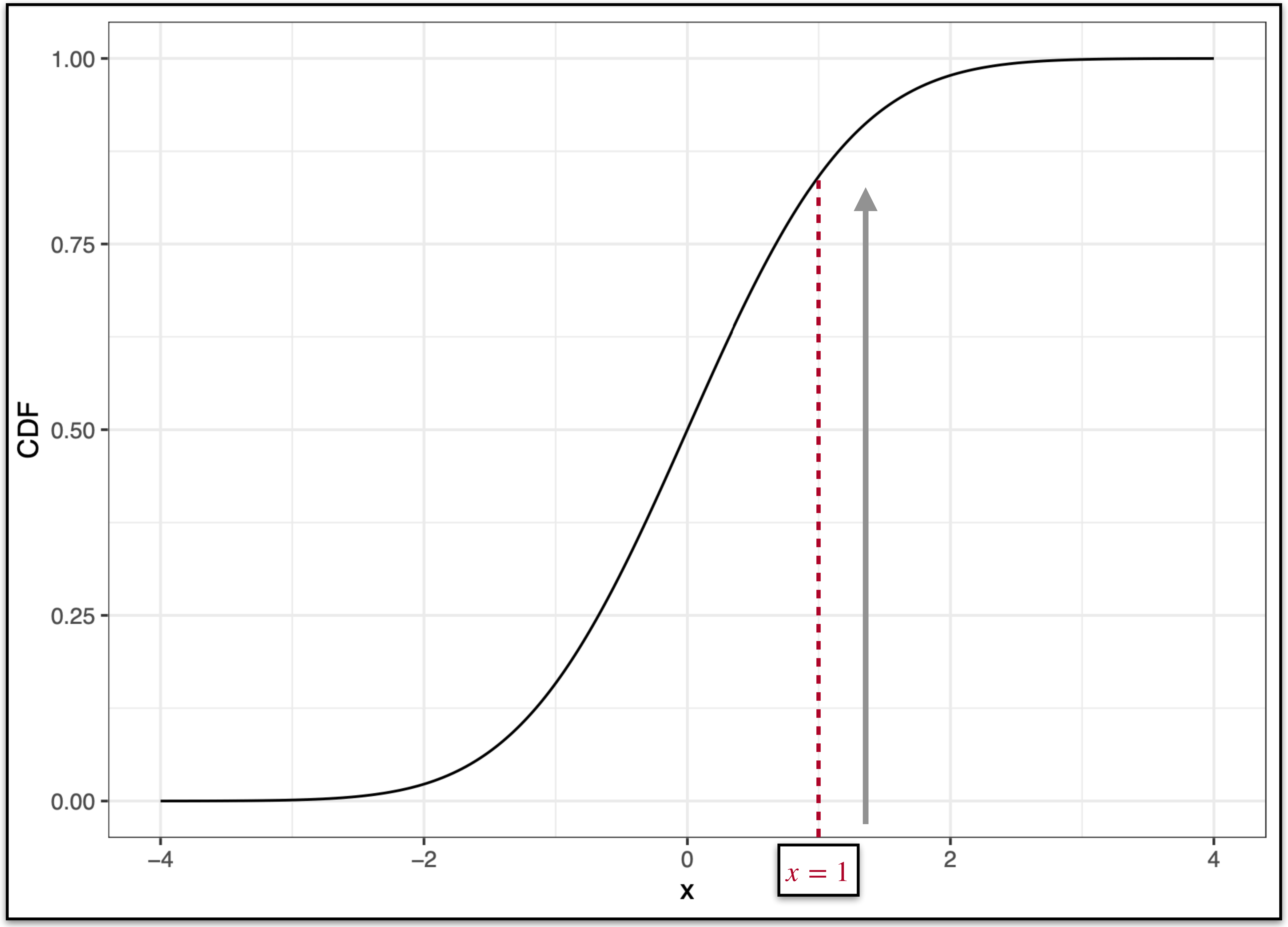
- A systematic way to assess the normal approximation is to see if the observed and theoretical proportions match.
- These proportions are known as quantiles, hence the name
- Denote $\Phi(x)$ as the function that gives us the probability of a standard normal variables being smaller than x. We've seen this before, any ideas?
- Then, the inverse function $\Phi^{-1}(x)$ yields the theoretical quantiles of a normal distribution
- Let me provide some intuition

Quantile-quantile plot



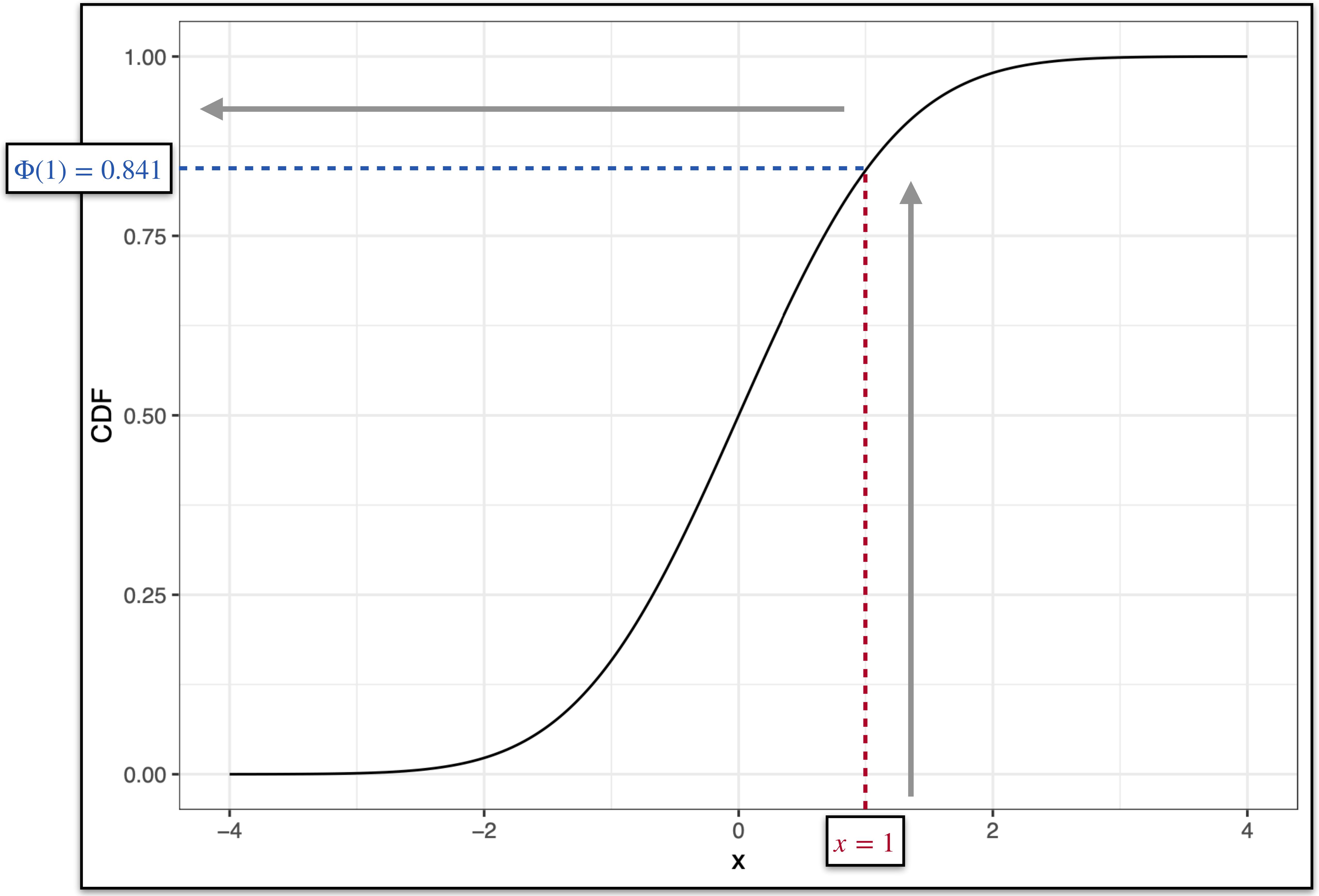
- $\Phi(1) = 0.841$: Again, this tell us the proportion of values that is less than 1

Quantile-quantile plot



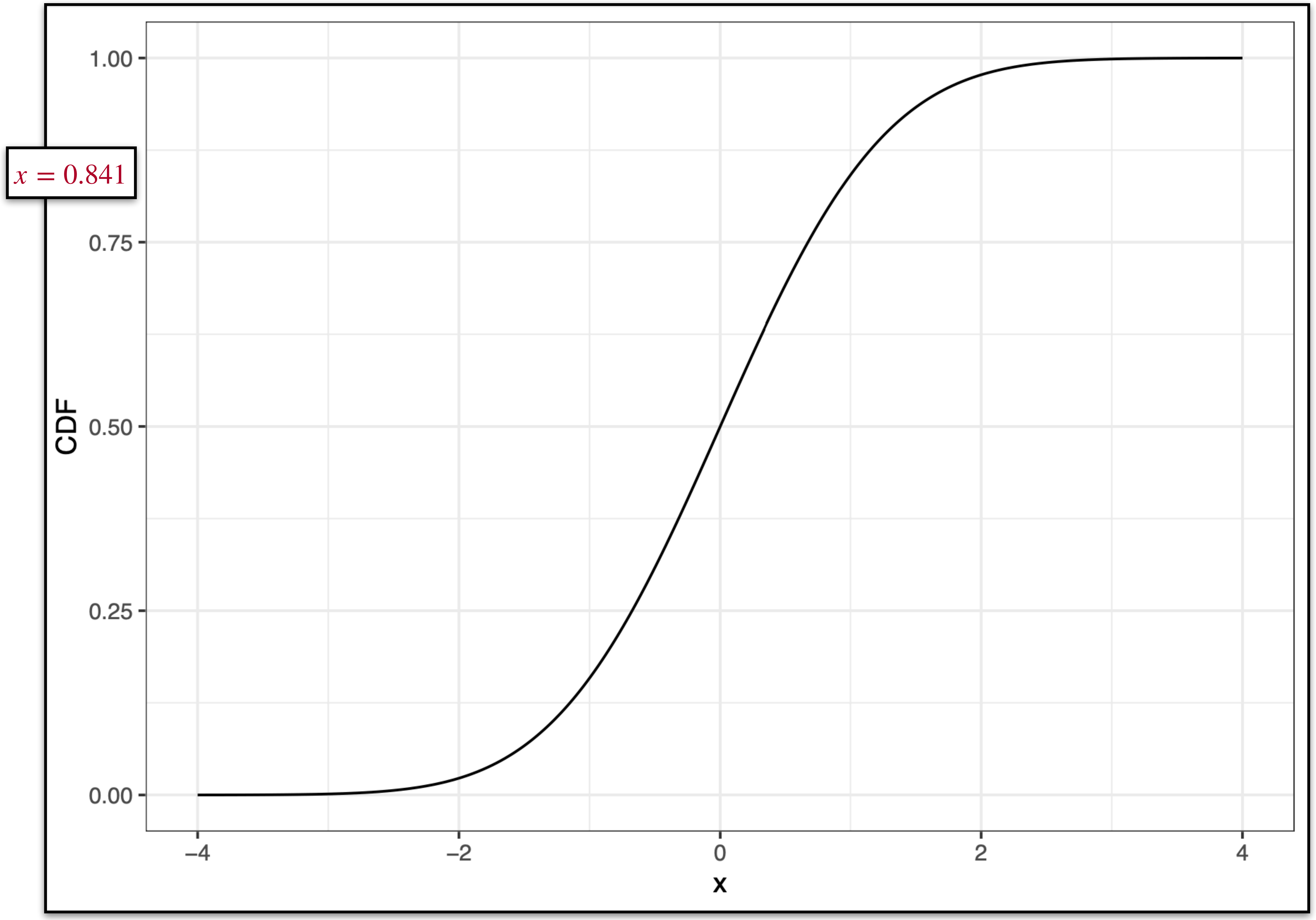
- $\Phi(1) = 0.841$: Again, this tell us the proportion of values that is less than 1

Quantile-quantile plot



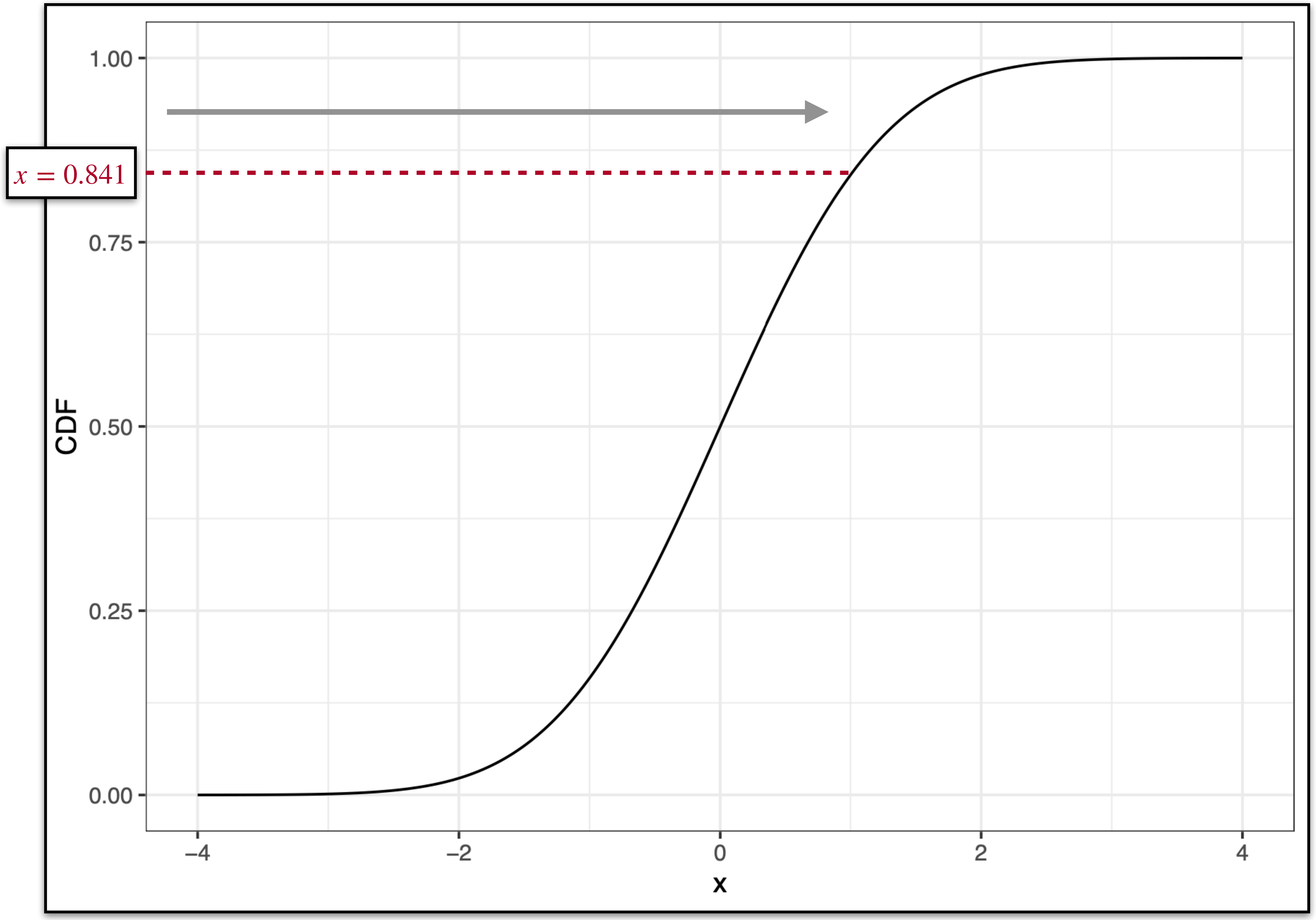
- $\Phi(1) = 0.841$: Again, this tell us the proportion of values that is less than 1

Quantile-quantile plot



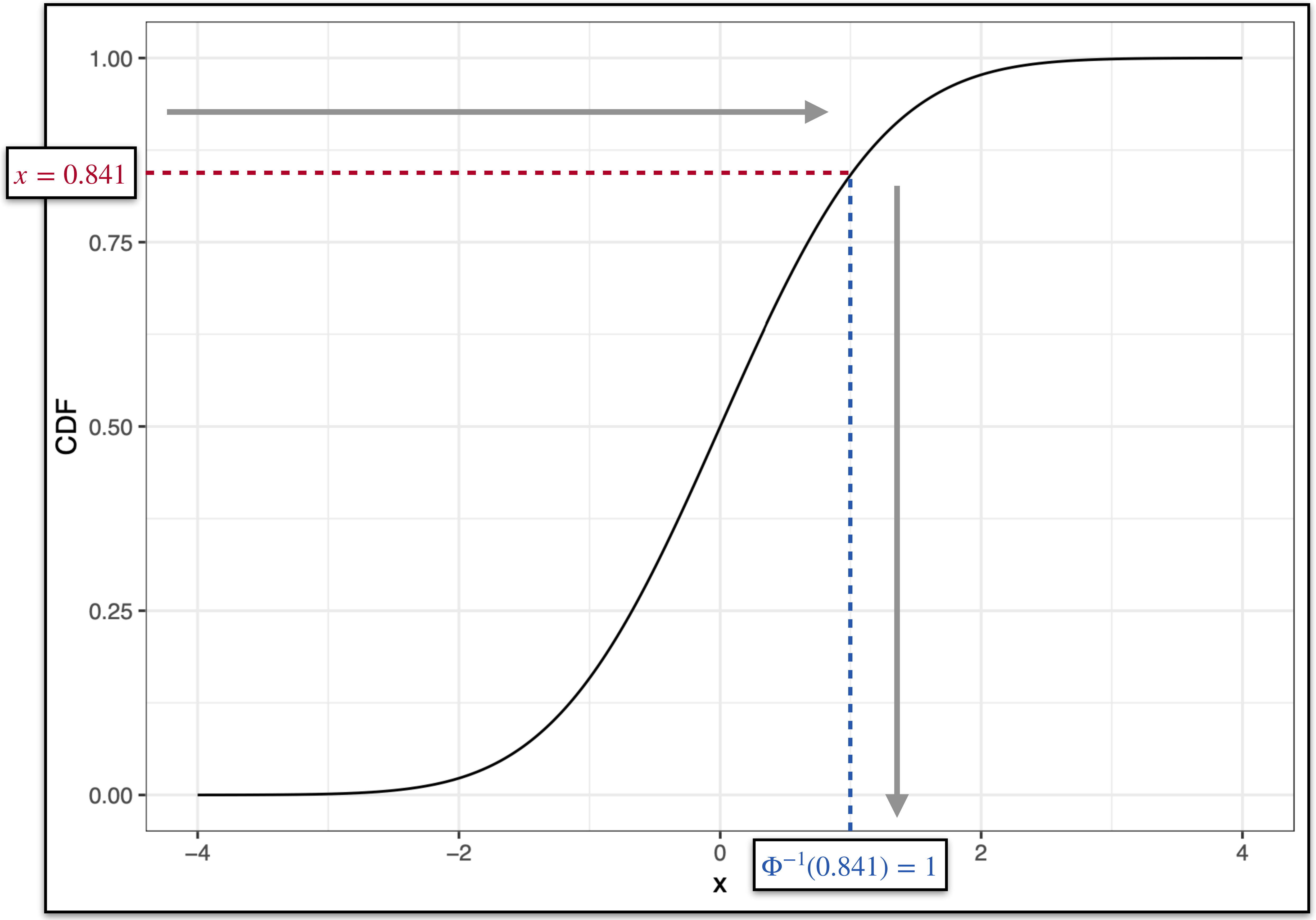
- $\Phi^{-1}(0.841) = 1$: This tell us, what is the value that is bigger than 84.1% of all values

Quantile-quantile plot



- $\Phi^{-1}(0.841) = 1$: This tell us, what is the value that is bigger than 84.1% of all values

Quantile-quantile plot



- $\Phi^{-1}(0.841) = 1$: This tell us, what is the value that is bigger than 84.1% of all values

Quantile-quantile plot

- In R we can evaluate $\Phi(x)$ using the following code:

```
> pnorm(1, mean=0, sd=1)
[1] 0.8413447
```

- and we can evaluate $\Phi^{-1}(x)$ with:

```
> qnorm(0.8413447, mean=0, sd=1)
[1] 0.9999998
```

- Quantiles can be defined for any distribution, including an empirical one
- If we have data $\{x_1, \ldots, x_n\}$, we can define the quantile associated with any proportion p as the q for which the proportion of values below q is p
- The idea of a QQ-plot is that if your data is well approximated by a normal distribution, then the quantiles of your data should be similar to the quantiles of a normal distribution

Quantile-quantile plot

- How to construct a QQ-plot:
  1. Define a vector of m proportions $p_1, \ldots, p_m$
  2. Define a vector of quantiles $q_1, \ldots, q_m$ for your data for the proportions above
     - These are known as ***Sample quantiles***
  3. Define a vector of quantiles for the proportions above from a normal distribution with the same mean and standard deviation as your data
     - These are known as the ***Theoretical quantiles***
  4. Plot the ***Sample quantiles*** versus the ***Theoretical quantiles***

Quantile-quantile plot

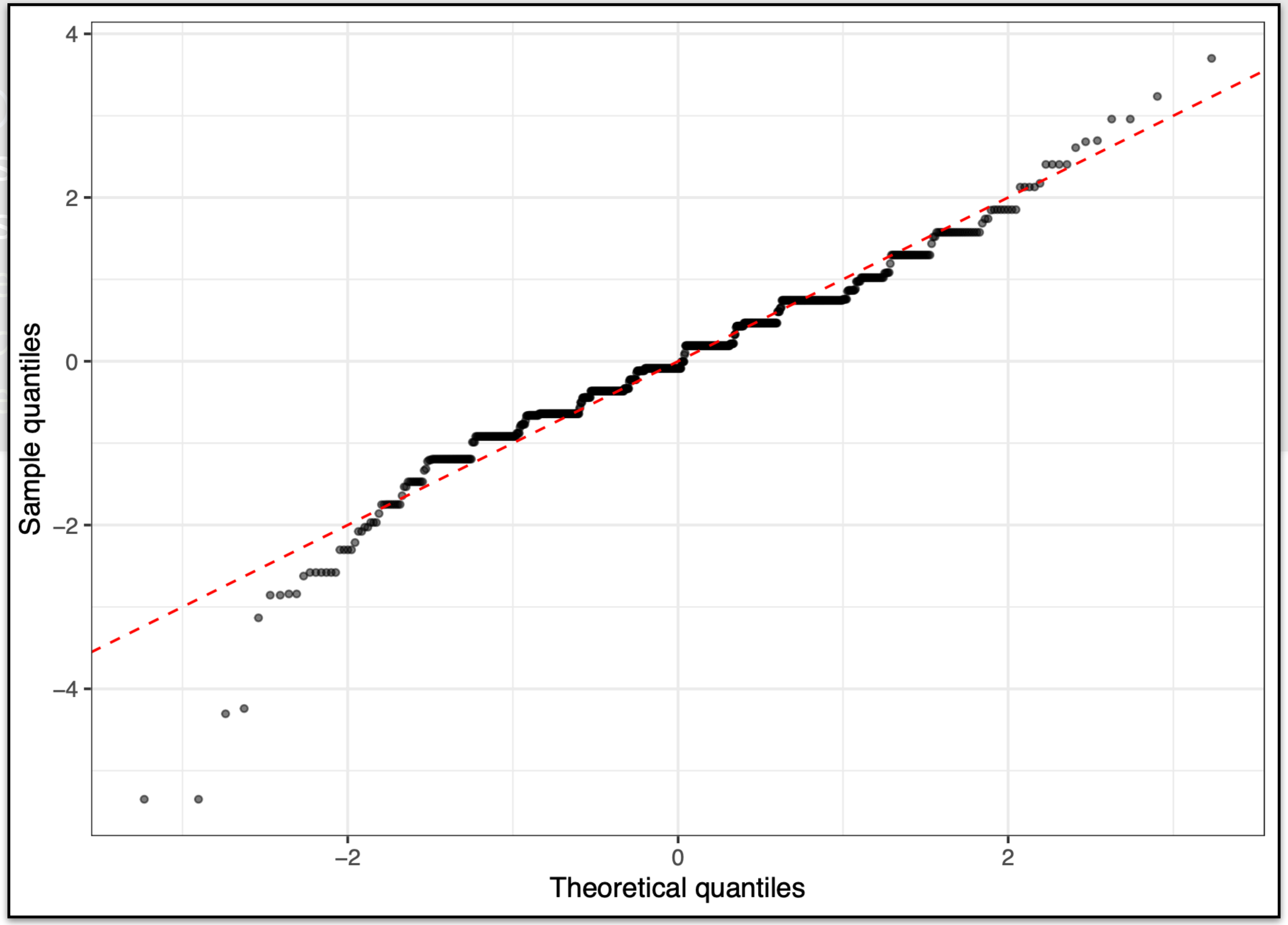
- QQ-plots in R:

```
heights %>%
  filter(sex == "Male") %>%
  ggplot(aes(sample = scale(height))) +
  geom_qq(alpha=0.50, size=1) +
  ylab("Sample quantiles") +
  xlab("Theoretical quantiles") +
  geom_abline(color="red", lty=2)
```

Quantile-quantile plot

- QQ-plots in R:

```
heights %>%
  filter(sex == "Male")
  ggplot(aes(sample = s
  geom_qq(alpha=0.50, s
  ylab("Sample quantile
  xlab("Theoretical qu
  geom_abline(color="re
```



Boxplots

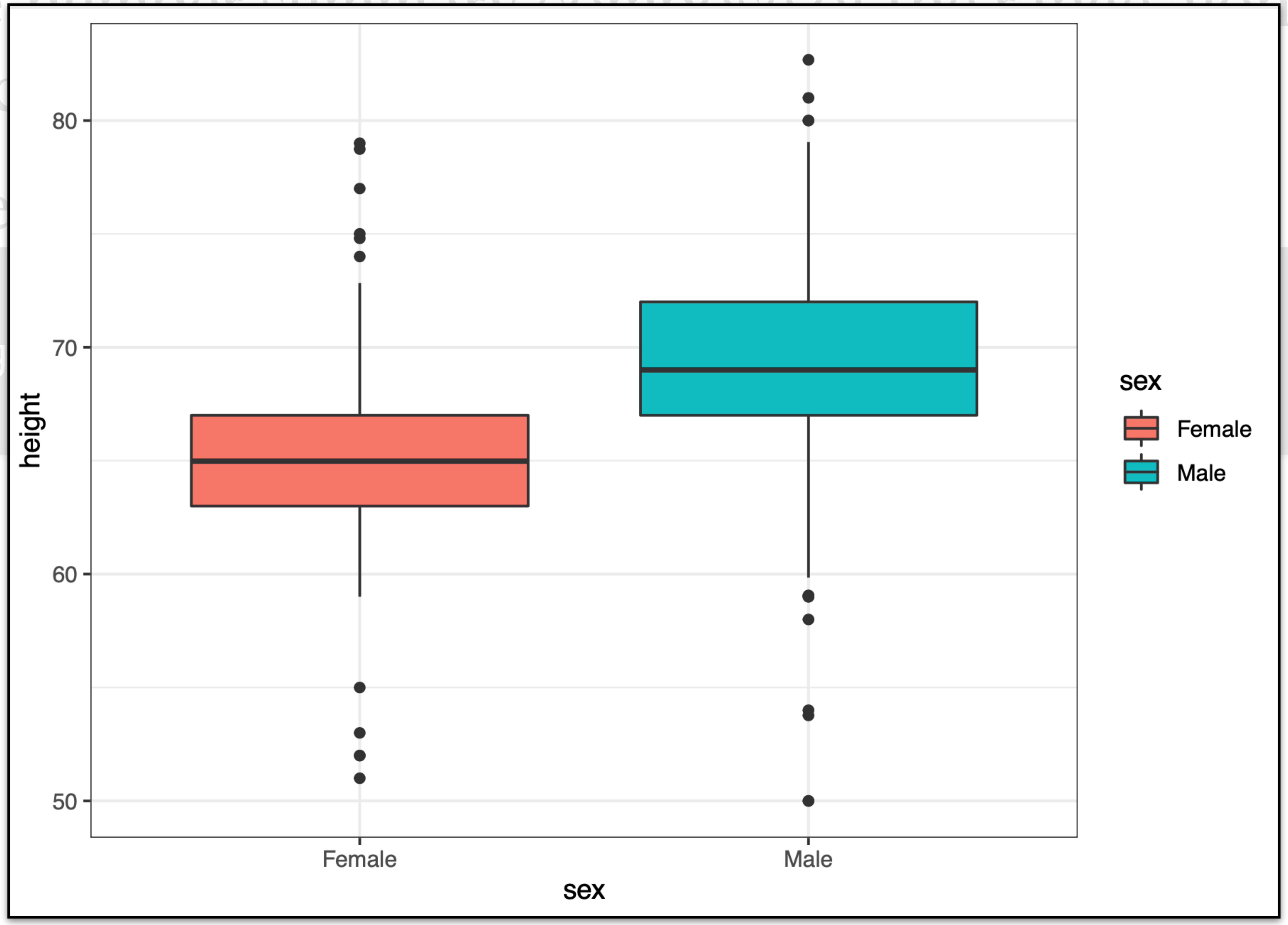
- Boxplots are a five-number summary compose of the range along with the, 25th, 50th, and 75th percentiles.
- Here is an example

```
heights %>%
   ggplot(aes(sex, height, fill = sex)) +
   geom_boxplot()
```

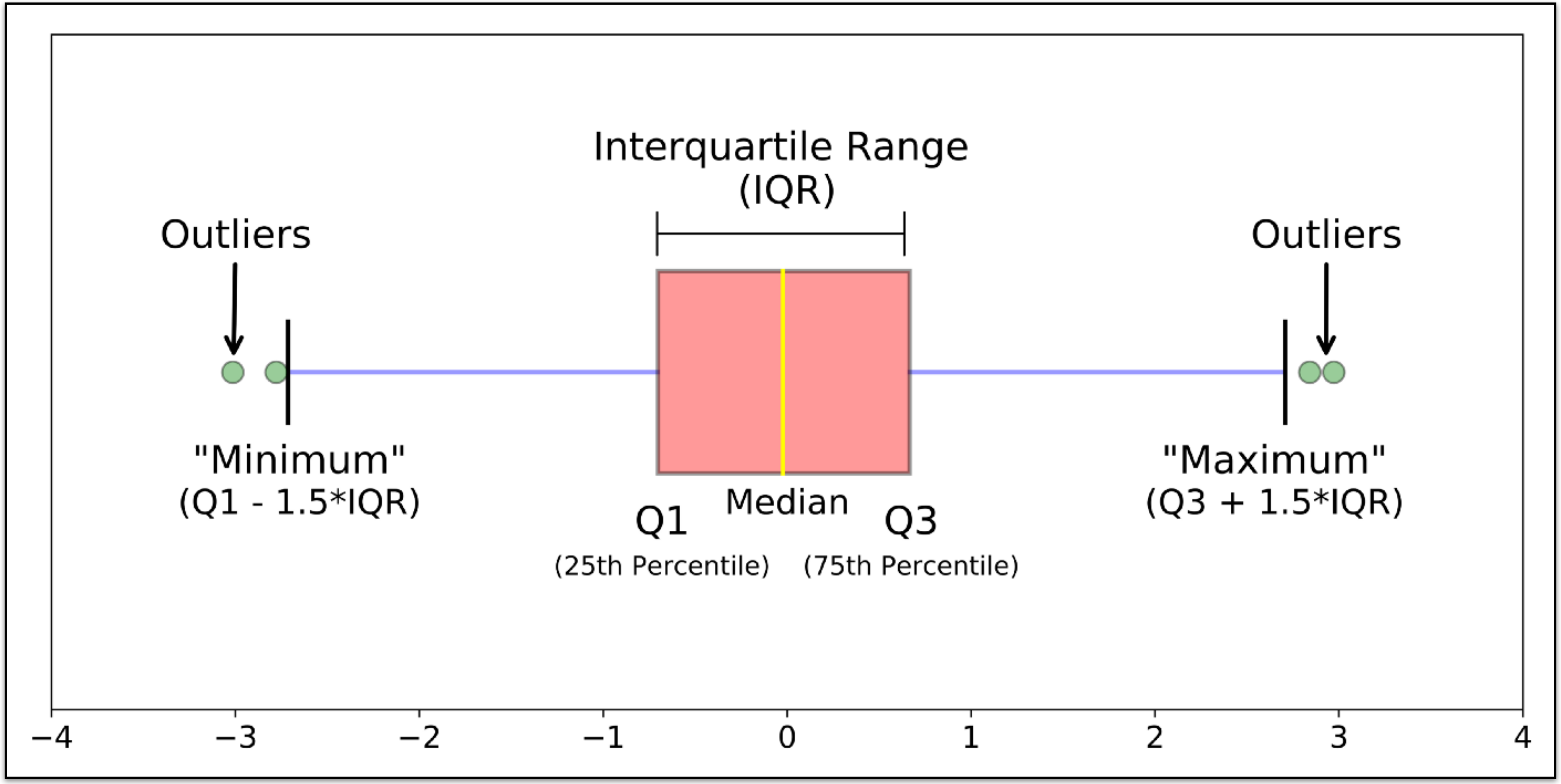
Boxplots

- Boxplots are a five … g with the, 25th, 50th, and 75th perc …
- Here is an example

```
heights %>%
  ggplot(aes(sex, heig
  geom_boxplot()
```



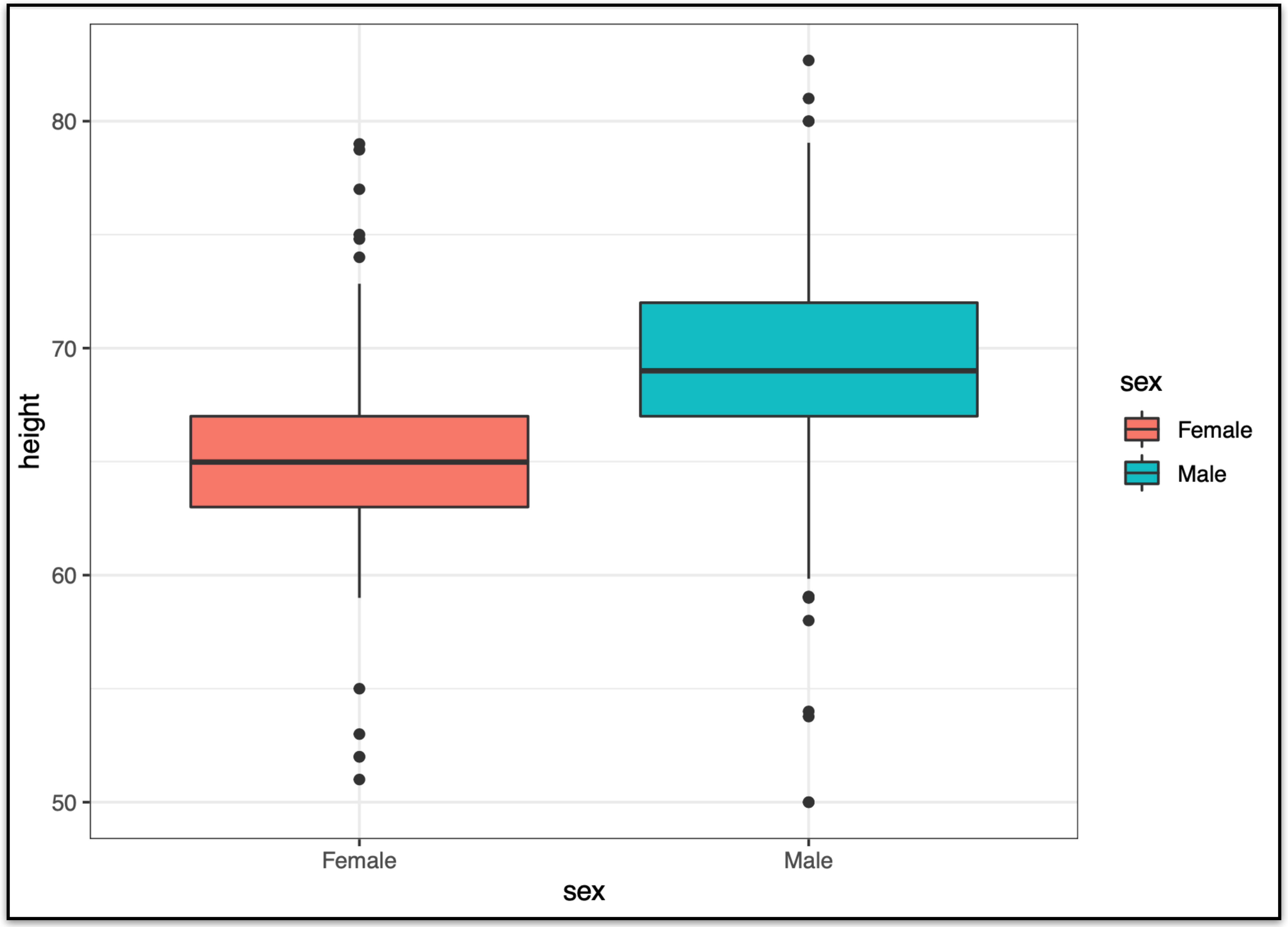
Boxplots



Interquartile Range
(IQR)
Outliers
Outliers
"Minimum"
(Q1 - 1.5*IQR)
Median
Q1
(25th Percentile)
Q3
(75th Percentile)
"Maximum"
(Q3 + 1.5*IQR)
−4
−3
−2
−1
0
1
2
3
4

Boxplots

- We see that males are, on average, higher than females
- The standard deviation seems to be similar between the two groups
- More analysis needed to assess if the normal approximation is appropriate for female data



References

1. Introduction to Data Science: Data analysis and prediction algorithms with R by Rafael A. Irizarry, Chapter 8. https://rafalab.github.io/dsbook/
2. R for Data Science by Grolemund & Wickham, Chapter 3. https://r4ds.had.co.nz/index.html
3. ggplot2: Elegant graphics for data analysis: Wickham. https://ggplot2-book.org

Referencias en español:

1. Introducción a la Ciencia de Datos: Análisis de datos y algoritmos de predicción con R por Rafael A. Irizarry, Capítulo 8. https://rafalab.github.io/dslibro/
2. R para Ciencia de Datos por Grolemund & Wickham, Capítulo 3. https://es.r4ds.hadley.nz