Lecture 01b – The fundamentals

ENVX2001 Applied Statistical Methods

Januar Harianto

Feb 2026

A refresher (for most)

You should be familiar with the following concepts:

1. Populations, samples and statistical inference
2. Probability distributions
3. Parameter estimation: central tendency, spread and variability
4. Sampling distribution of the mean: standard error, confidence intervals
5. Central Limit Theorem

That’s it!

Why review these concepts?

- These are concepts taught in first year statistics courses which we consider to be fundamental
- Even if you come from a unit/course that does not do R, **you should at least be familiar with the basic building blocks of statistical thinking**

Let's go through these concepts together!

Samples, populations

Populations

- **All** the possible units and their associated observations of interest
- Scientists are often interested in making *inferences* about populations, but measuring every unit is impractical

Samples

- A collection of observations from any population is a sample, and the number of observations in it is the **sample size**
- We assume samples that we collect can be used to make inferences about the population
- **NEW**: Samples need to be *representative* of the population

Statistics vs parameters

- Characteristics of the **population** are called *parameters* (e.g. the true average height of all trees in a forest)
- Characteristics of the **sample** are called *statistics* (e.g. the average height of 30 trees that we measured)
- In practice, we usually don't know the true population parameters, so we use sample statistics as our best estimates
- Drawing conclusions about a population from sample data is called **statistical inference**

Not all statistical techniques are inferential – some are purely descriptive (e.g. summarising data with graphs or tables) – but inference is the main focus of this course.

Sample data

Sample data are usually collected as **variables** – the characteristics we measure or record from each unit in our sample.

Variables can be:

Categorical Variables

- **Nominal**: categories without a natural order (e.g. soil type, land use)
- **Ordinal**: categories with a natural order (e.g. soil quality: poor, fair, good)

Numerical Variables

- **Continuous**: can take any value within a range (e.g. height, temperature)
- **Discrete**: can take only specific values (e.g. number of species, presence/absence)

Variable types can overlap

The categories on the previous slide are useful guidelines, but in practice, variables can cross between types depending on how you use them.

Examples

- **height (in cm)** – a numerical, **continuous** variable, but can be treated as categorical if you group it into categories (short, medium, tall)
- **age (in years)** – a numerical, **discrete** variable, but often treated as continuous because the gaps between values are small relative to the range
- **treatment (A, B, C)** – a categorical variable, but can be treated as numerical if we assign numbers to the treatments (1, 2, 3) and assume they are **ordered** e.g. effect of 1 < 2 < 3

Be careful when converting variables

Converting a numerical variable to a categorical one means **losing information** – once you group heights into “short”, “medium” and “tall”, the exact measurements are gone.

- A tree that is 5.1 m and a tree that is 9.9 m might both be called “tall”, but they are very different
- Only categorise when there is good reason to do so, as this reduces your analysis power
- Going the other way (categorical ⟶ numerical) doesn’t lose data, but it does **add assumptions** – e.g. that your categories have a meaningful order

Distribution of data

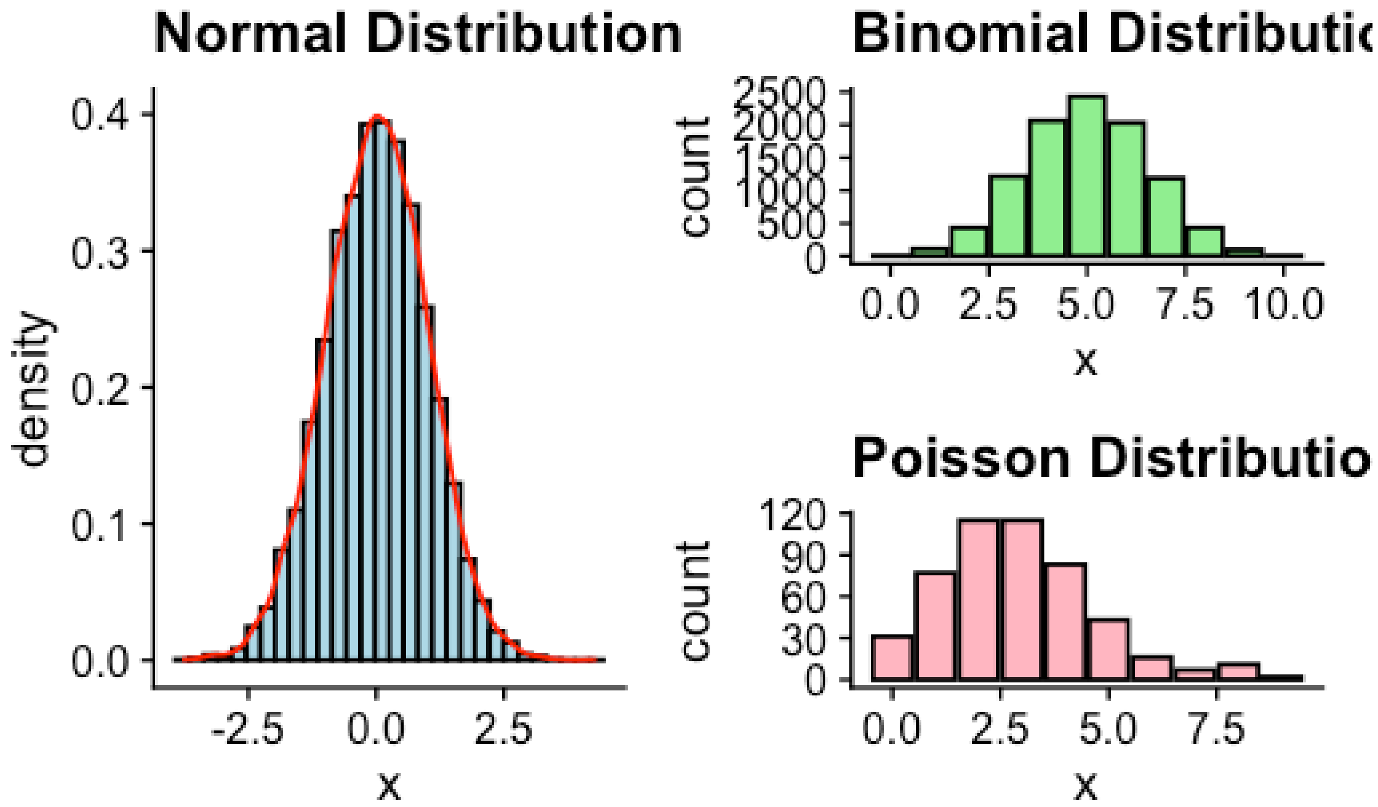
Types of probability distributions

Populations can be described by probability distributions. You may have encountered these common ones in first-year statistics:

- **Normal Distribution**: Bell-shaped curve, symmetric around the mean. **Data is continuous** (e.g. heights, weights)
- **Binomial Distribution**: Models success/failure outcomes in a fixed number of trials. **Data is discrete** (e.g. how many seeds germinate out of 10 planted)
- **Poisson Distribution**: Models count data over a fixed area or time period. **Data is discrete** (e.g. number of weeds found in a 1 m^2 quadrat)

Knowing the distribution of your data helps you choose the right statistical model. We will return to this throughout the course.

```r
library(tidyverse)

set.seed(908)
normal_data <- data.frame(x = rnorm(10000, mean = 0, sd = 1))
binomial_data <- data.frame(x = rbinom(10000, size = 10, prob = 0.5))
poisson_data <- data.frame(x = rpois(500, lambda = 3))

# normal
p1 <- ggplot(normal_data, aes(x = x)) +
  geom_histogram(aes(y = after_stat(density)), bins = 30, fill = "lightblue", color = "black") +
  geom_density(color = "red") +
  ggtitle("Normal Distribution")

# binomial
p2 <- ggplot(binomial_data, aes(x = x)) +
  geom_bar(fill = "lightgreen", color = "black") +
  ggtitle("Binomial Distribution")

# poisson
p3 <- ggplot(poisson_data, aes(x = x)) +
  geom_bar(fill = "lightpink", color = "black") +
  ggtitle("Poisson Distribution")

# Arrange plots
library(patchwork)
p1 | p2 / p3
```


Normal Distribution
density
0.4
0.3
0.2
0.1
0.0
-2.5
0.0
2.5
x
Binomial Distributic
count
2500
2000
1500
1000
500
0
0.0
2.5
5.0
7.5
10.0
x
Poisson Distributio
count
120
90
60
30
0
0.0
2.5
5.0
7.5
x

Parameter estimation

Measures of central tendency

Central tendency tells you where the "middle" or "typical" value of your data falls. There are several ways to measure it, each with trade-offs.

Mean $\bar{x}$

- The sum of all values divided by the number of observations – what `mean()` does in R
- Sensitive to extreme values (outliers)

⚠ Formula

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

- where n is the number of observations
- x_i represents each individual value
- $\sum$ means we add up all values from $i = 1$ to n
- Example: for data {2,4,6,8}, $n = 4$ and $\bar{x} = \frac{2+4+6+8}{4} = 5$

Median and mode

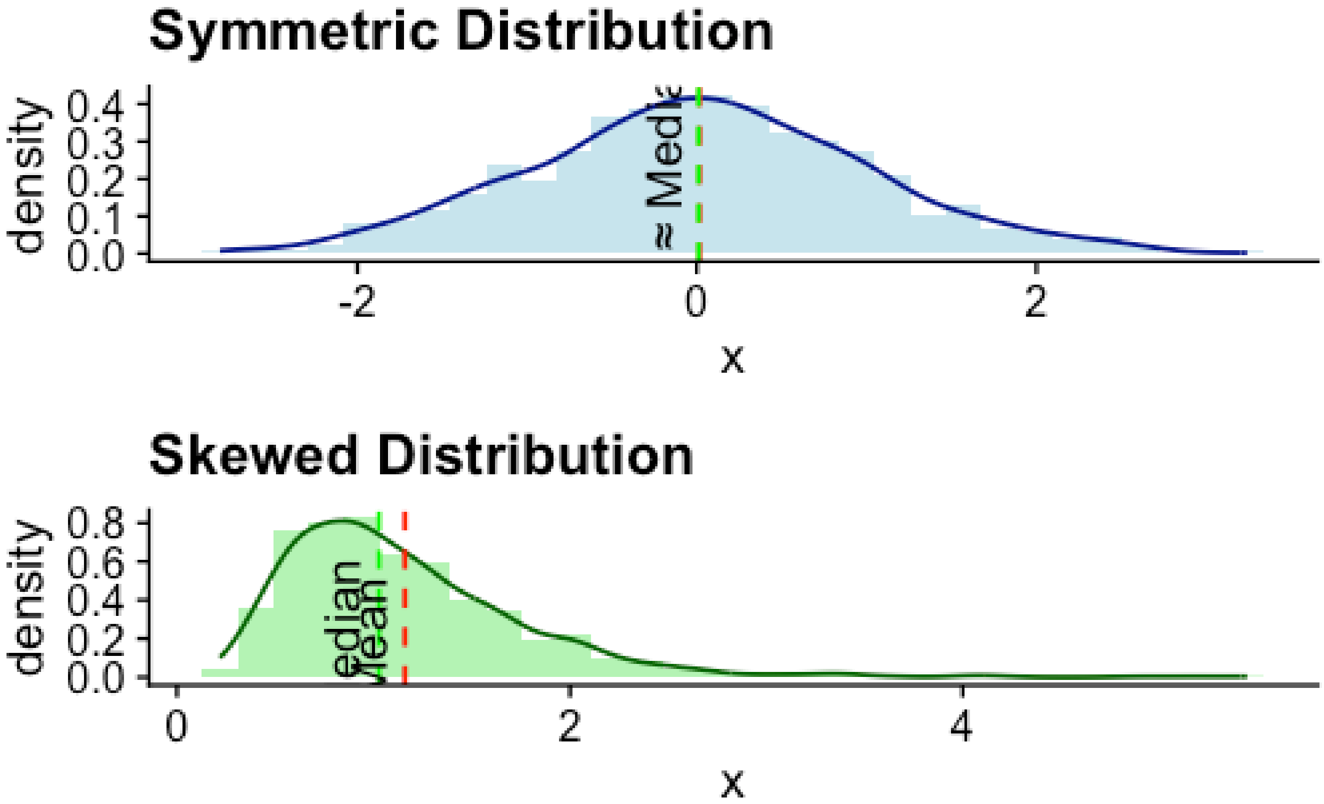
Median

- Sort your data from smallest to largest and take the middle value
- The median is resistant to extreme values, making it a better representation of typical observations when data is skewed

Mode

- The most frequently occurring value in a dataset
- Can have multiple modes (e.g. a bimodal distribution has two peaks)
- The only measure of central tendency that works for categorical data

How they compare



When the mean and median are close, your data is roughly symmetric. When they differ substantially, the data is skewed – the mean shifts toward extreme values.

Measures of dispersion

Dispersion measures how spread out your data is. Two datasets can have identical means but very different spreads.

Variance s^2

- The average squared distance of each value from the mean
- Since the result is in squared units (e.g. m^2), it’s hard to interpret directly – we usually take the square root to get standard deviation

Measures of dispersion

⚠ Formula

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

- where n is the number of observations
- x_i represents each individual value
- $\bar{x}$ is the sample mean
- For data {2,4,6,8}:
  - mean = 5
  - differences = (-3,-1,1,3), squares = (9,1,1,9), sum = 20
  - Therefore, $s^2 = \frac{20}{3} \approx 6.67$

Measures of dispersion

Standard Deviation s

- The square root of variance – expressed in your data's original units (e.g. metres, not metres²)
- Shows how far a typical observation is from the mean

⚠ Formula

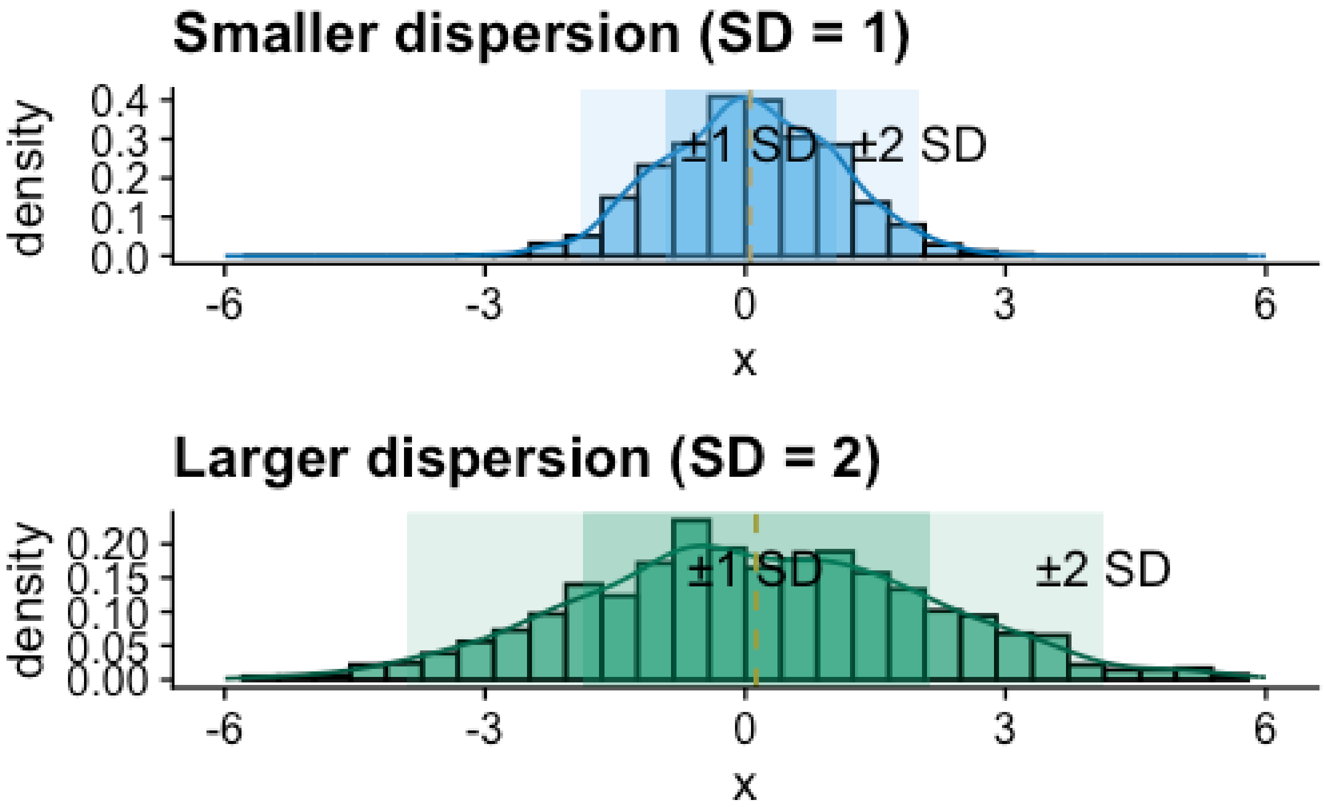
$$s = \sqrt{s^2}$$

- For our example data {2,4,6,8}: $s = \sqrt{6.67} \approx 2.58$

Visualising standard deviation

For normally distributed data, standard deviation tells you what proportion of values fall within each range of the mean:

- about **68%** of values fall within ±1 SD
- about **95%** within ±2 SD
- about **99.7%** within ±3 SD



Smaller dispersion (SD = 1)

Larger dispersion (SD = 2)

Population parameters vs sample statistics

Throughout the course you'll see Greek letters (μ, σ) for population parameters and Roman letters ($\bar{x}$, s) for sample statistics:

Mean

Population Parameter	Sample Statistic
$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$	$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$

Variance

Population Parameter	Sample Statistic
$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2$	$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$

Standard Deviation

Population Parameter	Sample Statistic
$\sigma = \sqrt{\sigma^2}$	$s = \sqrt{s^2}$

Note

Notice the use of $n - 1$ in sample variance and standard deviation

Why n-1?

- Sample variance uses $n - 1$ instead of n in the denominator
- This correction (Bessel's correction) accounts for bias when estimating the mean from the sample itself

- Here's the logic: once you know the mean and all but one value, the last value is determined – it must satisfy the mean. Only $n - 1$ values are truly free to vary
- These independent values are your **degrees of freedom**

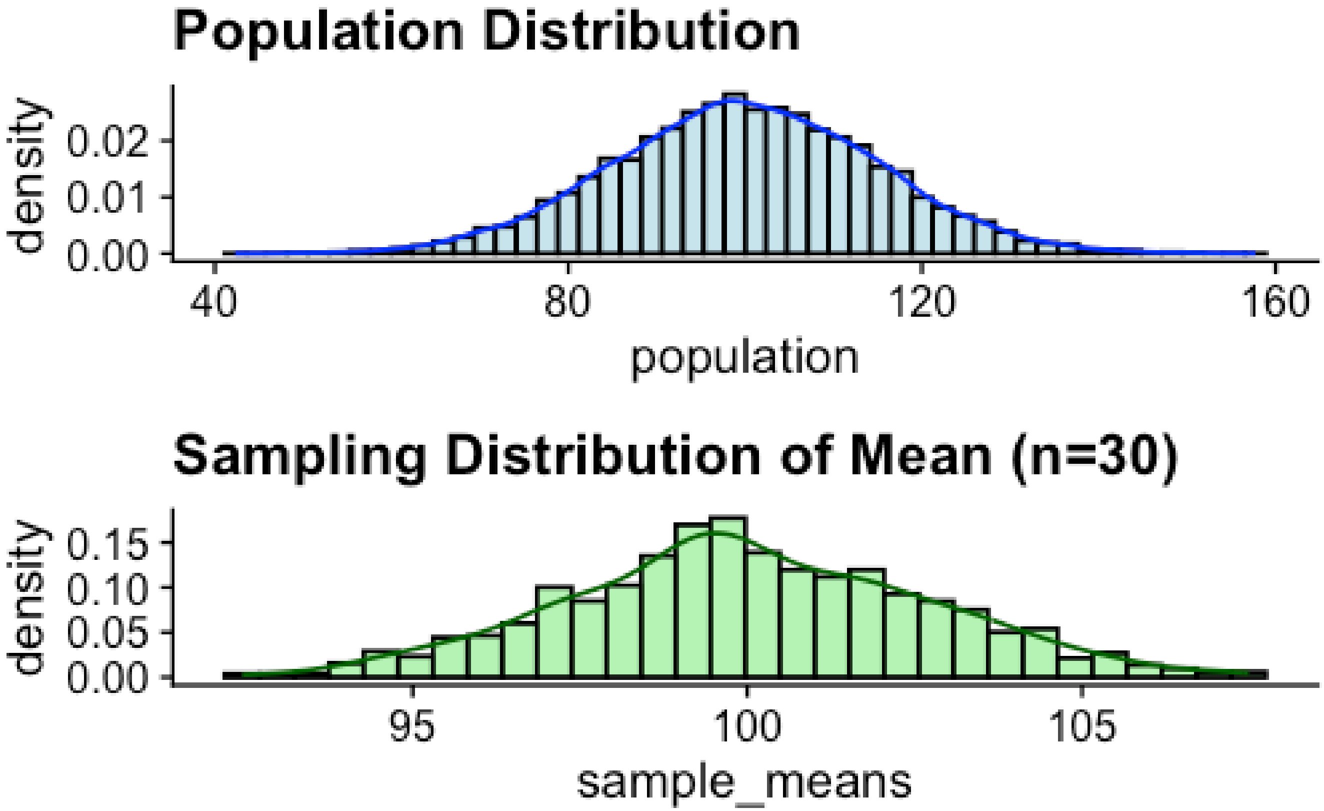
Sampling distributions and CLT

What is a sampling distribution?

Imagine you collected your sample of 30 trees, calculated the mean height, then did it again – 1000 times. Each time, you'd get a slightly different mean.

- The distribution of all those means is a **sampling distribution**
- It shows how much your estimate (the sample mean) varies due to sampling randomness
- This foundation of statistical inference lets you quantify confidence in your estimates

Sampling distribution of the mean



Central Limit Theorem

> I know of scarcely anything so apt to impress the imagination as the wonderful form of cosmic order expressed by the Central Limit Theorem. The law would have been personified by the Greeks and deified, if they had known of it."

– Sir Francis Galton, 1889, Natural Inheritance

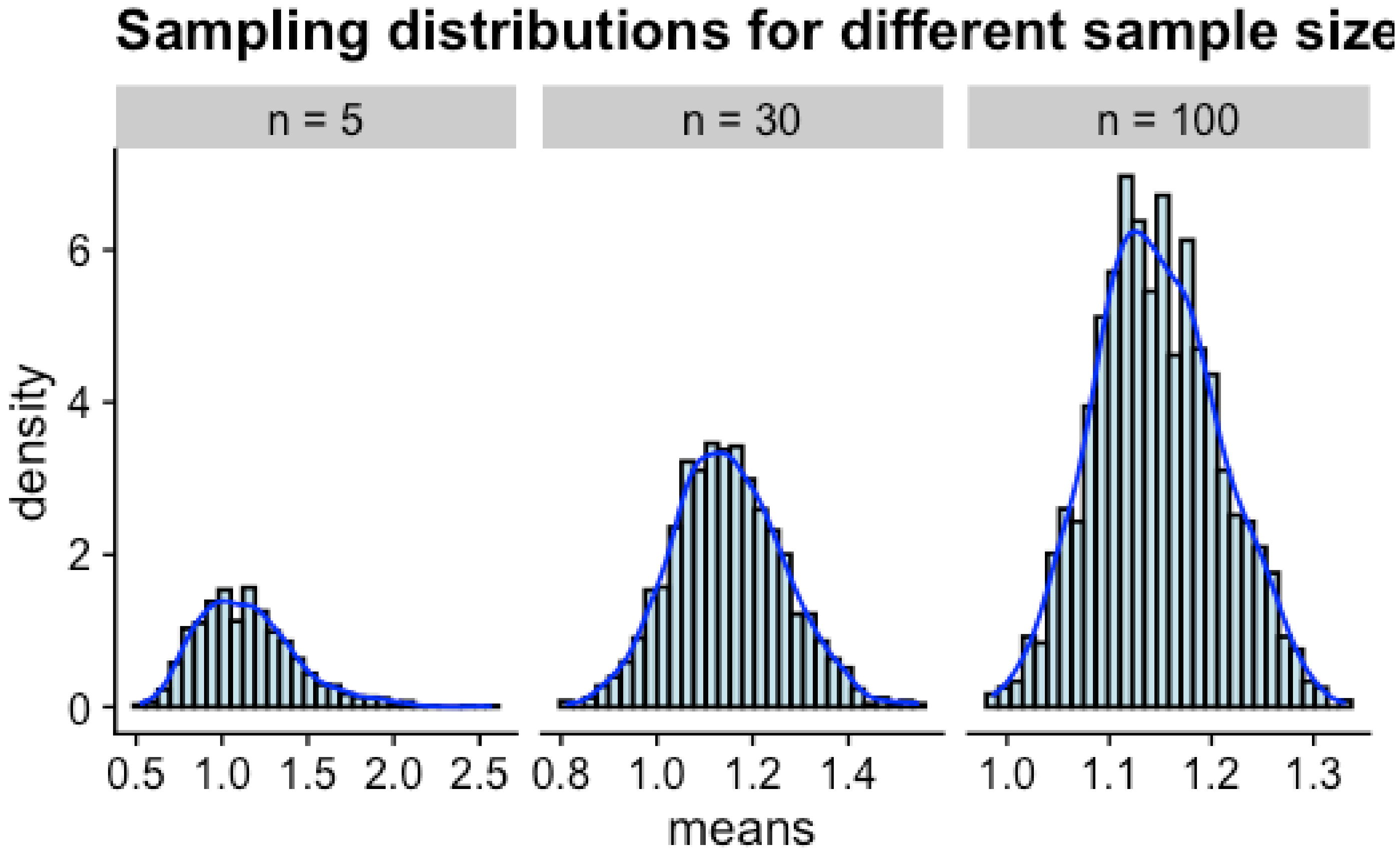
The Central Limit Theorem (CLT) states that for sufficiently large samples:

1. The distribution of sample means becomes **approximately normal**, regardless of the population's shape
2. The average of those sample means equals the **true population mean**
3. The spread of the sampling distribution decreases as sample size increases: $SE = \frac{\sigma}{\sqrt{n}}$

In short: with larger samples, their means form a predictable, bell-shaped pattern – even if the original data is skewed.

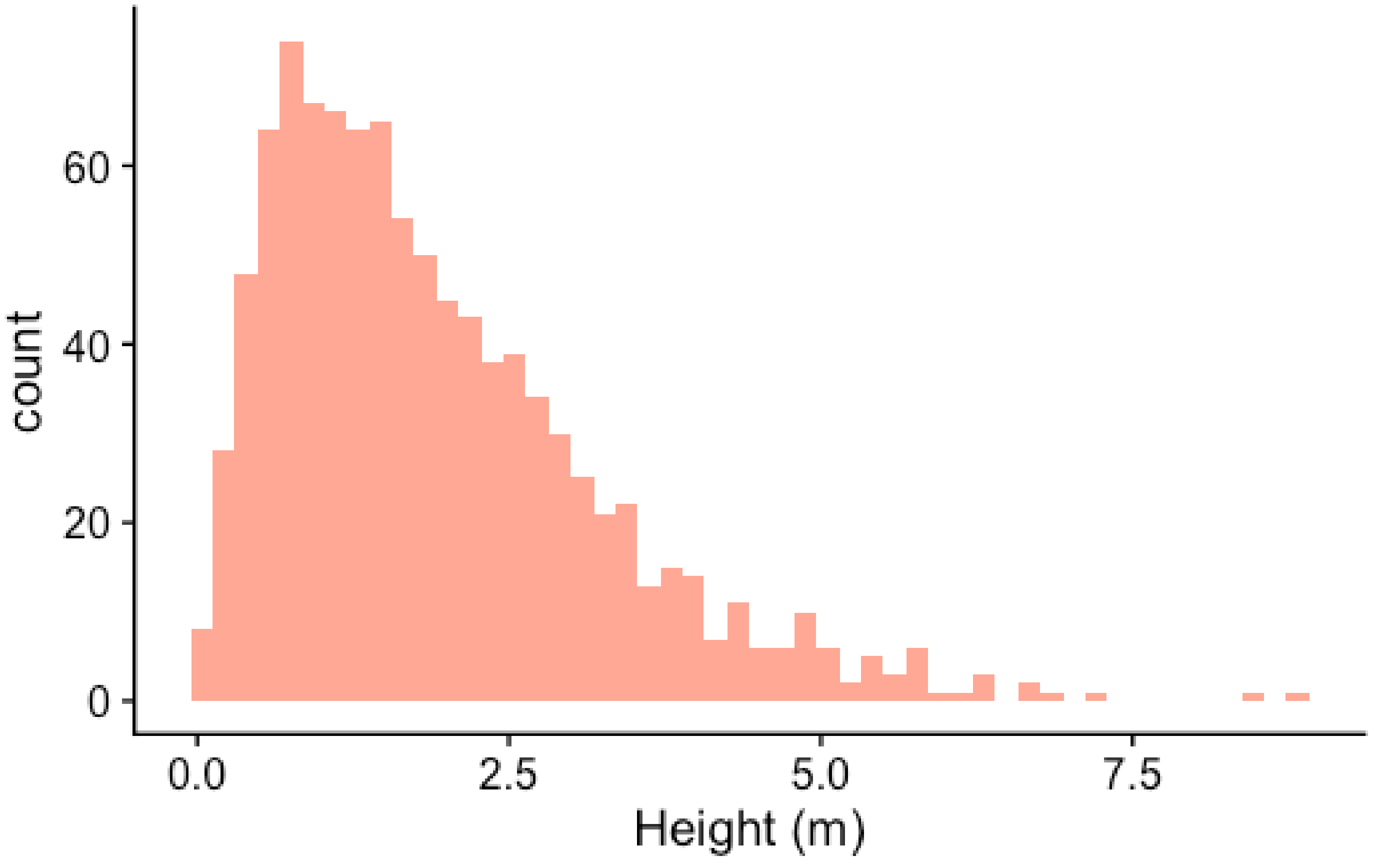
CLT in action

```r
# Create a skewed population
set.seed(456)
skewed_pop ← exp(rnorm(10000, mean = 0, sd = 0.5))

# Sample means for different sample sizes (ordered small to large)
sample_sizes ← c(5, 30, 100)
sample_labels ← factor(paste("n =", sample_sizes),
  levels = paste("n =", sample_sizes)
) # preserve order
sample_dist_data ← lapply(sample_sizes, function(n) {
  means ← replicate(1000, mean(sample(skewed_pop, size = n)))
  data.frame(means = means, size = factor(paste("n =", n), levels = levels(sample_labels)))
})
sample_dist_df ← do.call(rbind, sample_dist_data)

# Plot
ggplot() +
  geom_histogram(aes(x = means, y = after_stat(density)),
    data = sample_dist_df,
    bins = 30, fill = "lightblue", color = "black", alpha = 0.7
  ) +
  geom_density(aes(x = means), data = sample_dist_df, color = "blue") +
  facet_wrap(~size, scales = "free_x") +
  ggtitle("Sampling distributions for different sample sizes")
```


Sampling distributions for different sample size
n = 5
n = 30
n = 100
density
0
2
4
6
0.5
1.0
1.5
2.0
2.5
0.8
1.0
1.2
1.4
1.0
1.1
1.2
1.3
means

Example

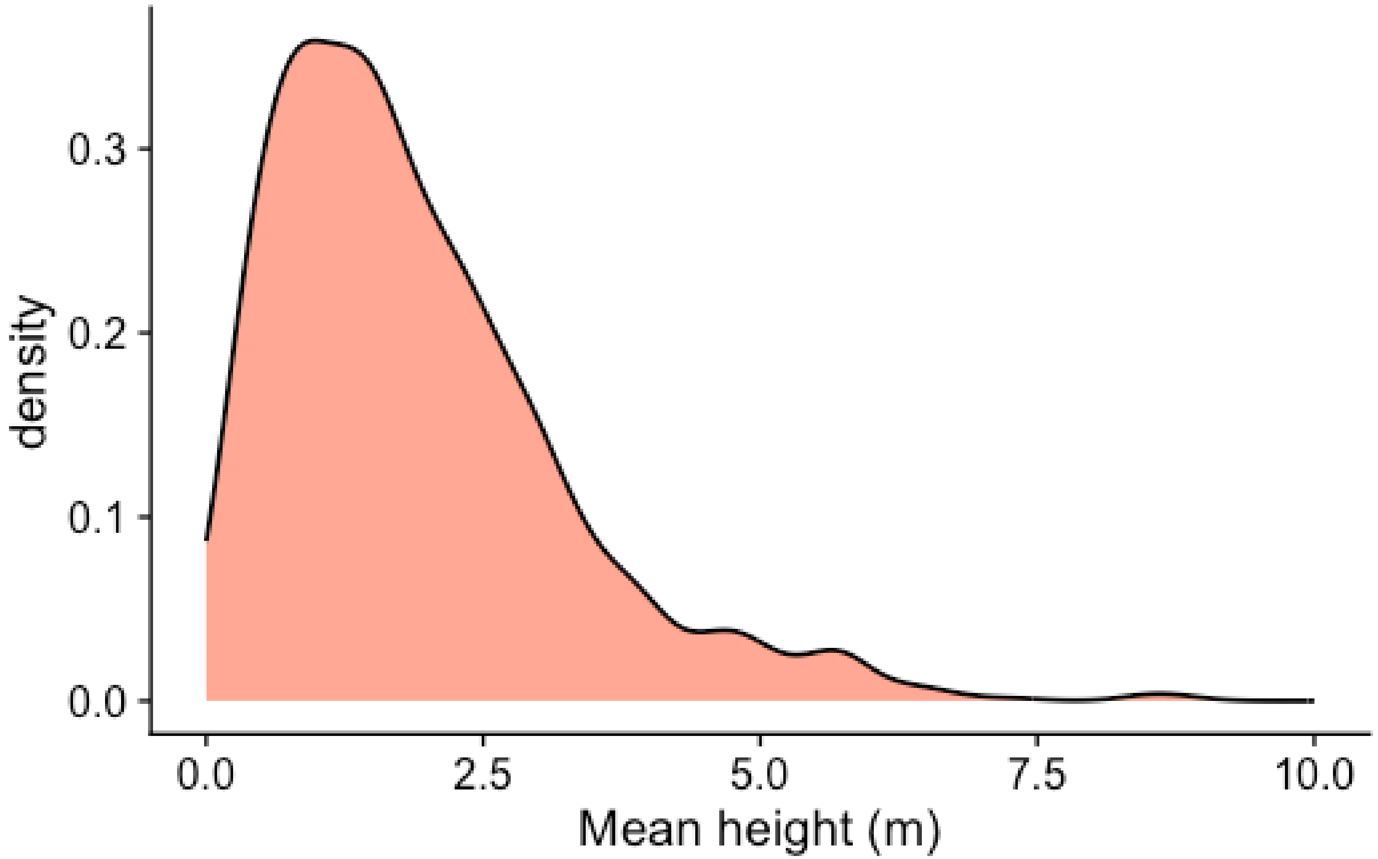
```r
set.seed(239)
# Generate a skewed distribution
skewed <- tibble(
  x = rgamma(1000, shape = 2, scale = 1)
)

# plot in ggplot2
ggplot(data = skewed, aes(x = x)) +
  geom_histogram(
    fill = "orangered",
    alpha = 0.5, bins = 50
  ) +
  xlab("Height (m)")
```



- Skewed population distribution for tree heights.
- We want to estimate the mean height of the trees in the forest.

Sample size: n = 1

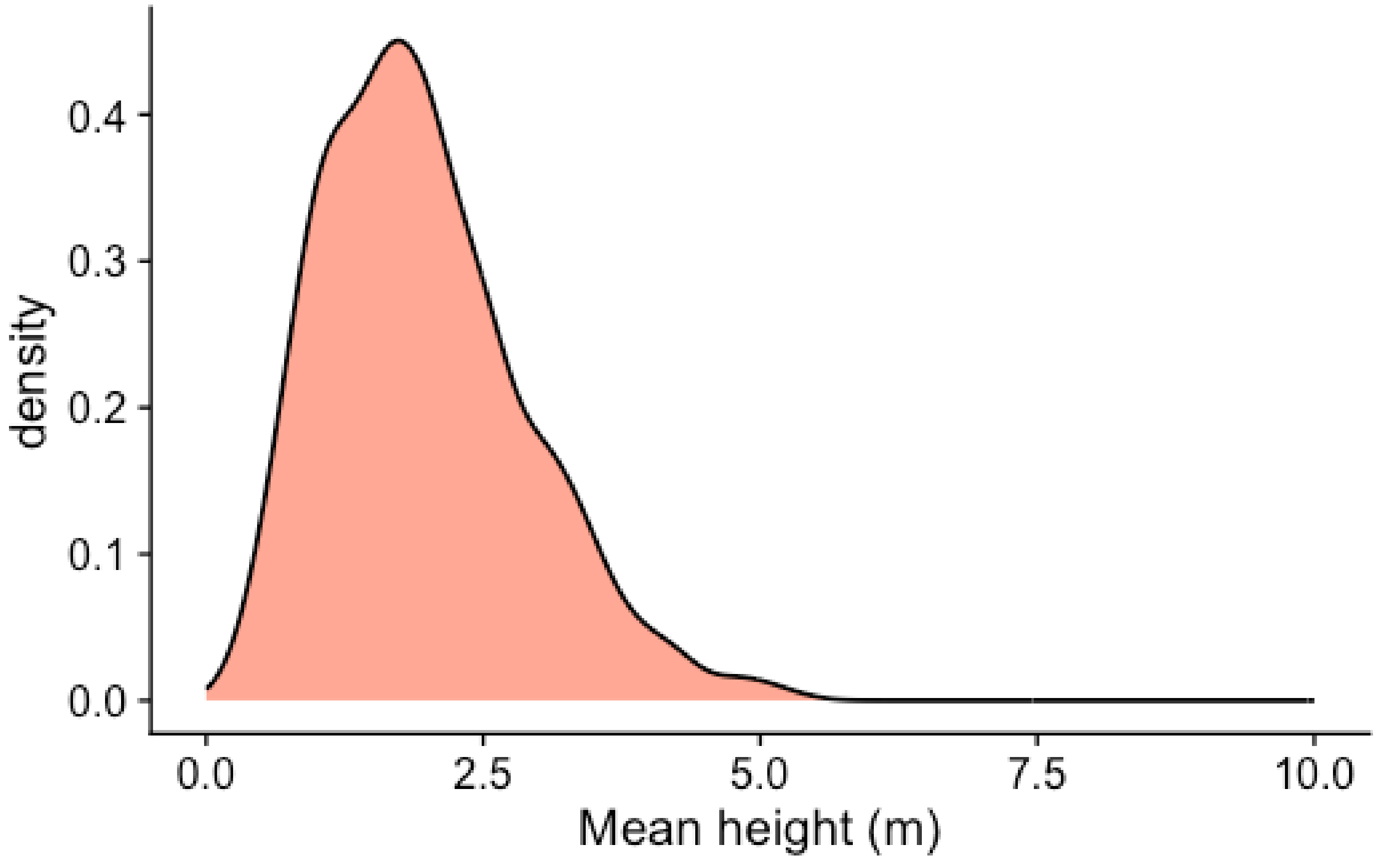
```r
skewed |>
  infer::rep_sample_n(
    size = 1,
    reps = 1000
  ) |>
  group_by(replicate) |>
  summarise(xbar = mean(x)) |>
  ggplot(aes(x = xbar)) +
  geom_density(
    fill = "orangered",
    alpha = 0.5, bins = 50
  ) +
  xlim(0, 10) +
  xlab("Mean height (m)")
```



With a sample size of one, the sampling distribution mirrors the population – larger samples are needed to see convergence to normality.

Sample size: n = 2

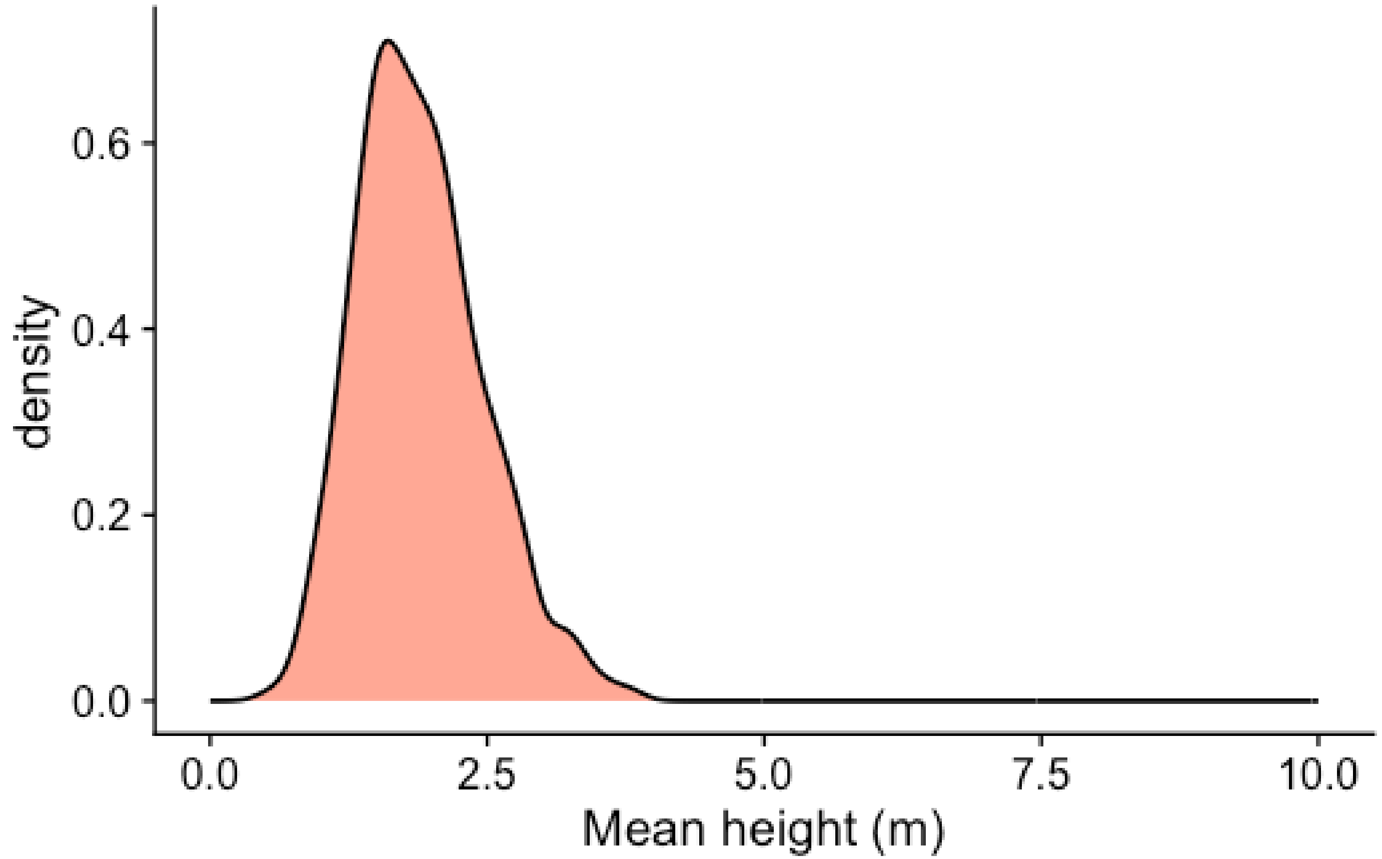
```r
skewed |>
  infer::rep_sample_n(
    size = 2,
    reps = 1000
  ) |>
  group_by(replicate) |>
  summarise(xbar = mean(x)) |>
  ggplot(aes(x = xbar)) +
  geom_density(
    fill = "orangered",
    alpha = 0.5, bins = 50
  ) +
  xlim(0, 10) +
  xlab("Mean height (m)")
```



- We sample 2 trees, calculate the sample mean, and repeat 1000 times.
- The distribution of sample means is starting to look more like a normal distribution.

Sample size: n = 5

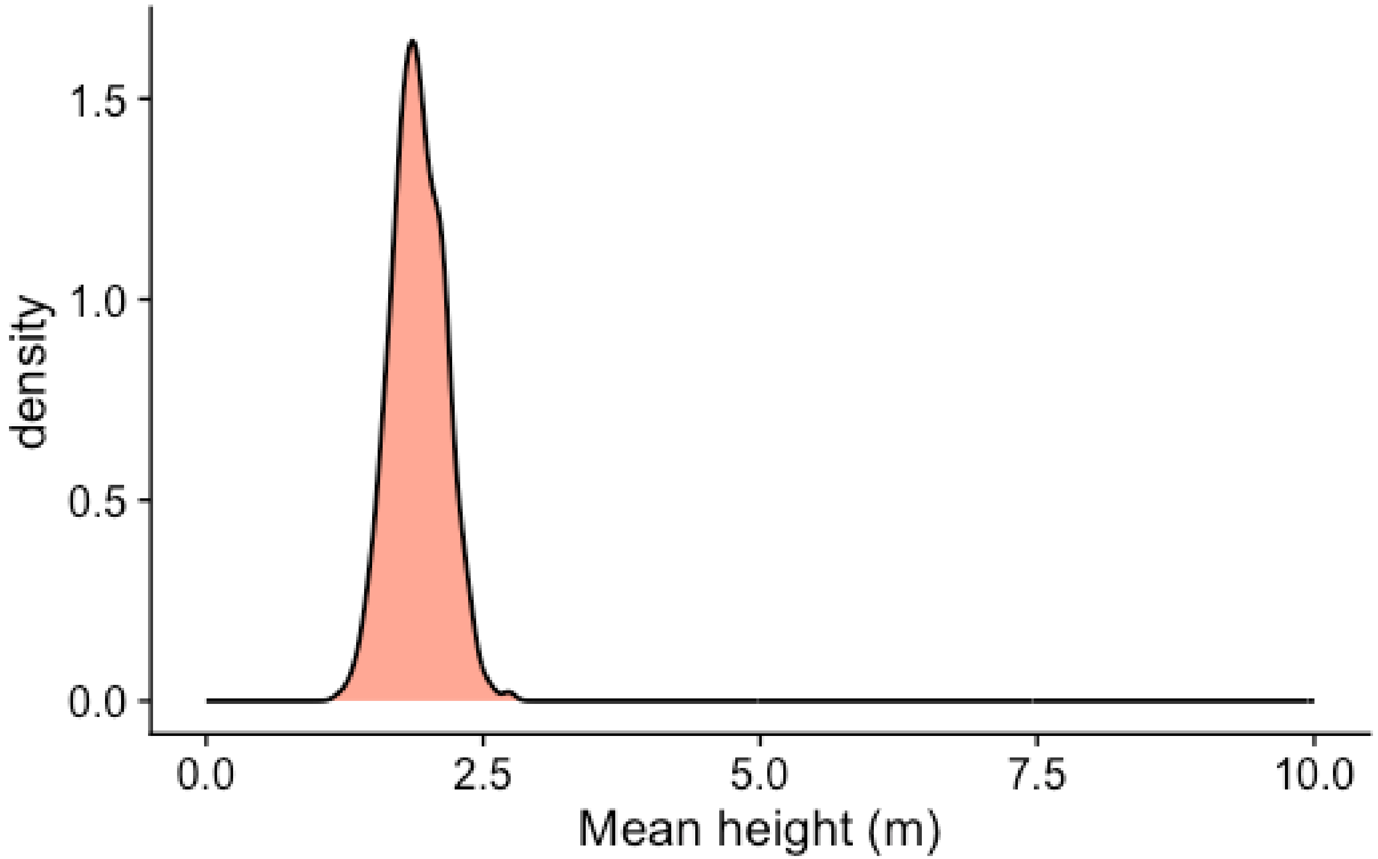
```r
skewed |>
  infer::rep_sample_n(
    size = 5,
    reps = 1000
  ) |>
  group_by(replicate) |>
  summarise(xbar = mean(x)) |>
  ggplot(aes(x = xbar)) +
  geom_density(
    fill = "orangered",
    alpha = 0.5, bins = 50
  ) +
  xlim(0, 10) +
  xlab("Mean height (m)")
```



- Each sample mean is based on 5 observations, repeated 1000 times.
- The distribution is becoming more normal, and the spread is decreasing: estimate is getting more **precise**.

Sample size: n = 30

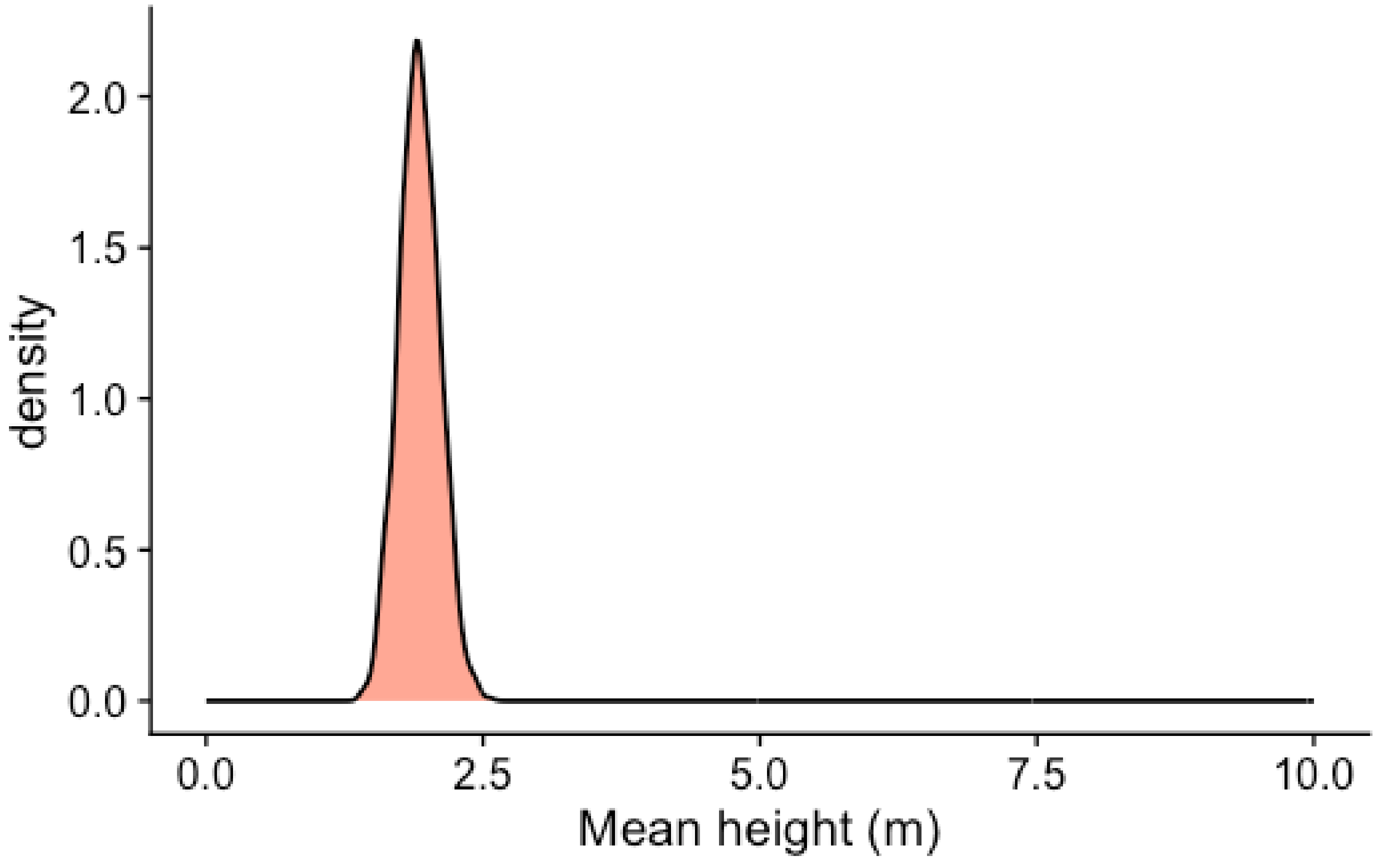
```r
skewed |>
  infer::rep_sample_n(
    size = 30,
    reps = 1000
  ) |>
  group_by(replicate) |>
  summarise(xbar = mean(x)) |>
  ggplot(aes(x = xbar)) +
  geom_density(
    fill = "orangered",
    alpha = 0.5, bins = 50
  ) +
  xlim(0, 10) +
  xlab("Mean height (m)")
```



- Each sample mean is based on 30 observations, repeated 1000 times.
- The distribution of sample means is very close to a normal distribution.

Sample size: n = 50

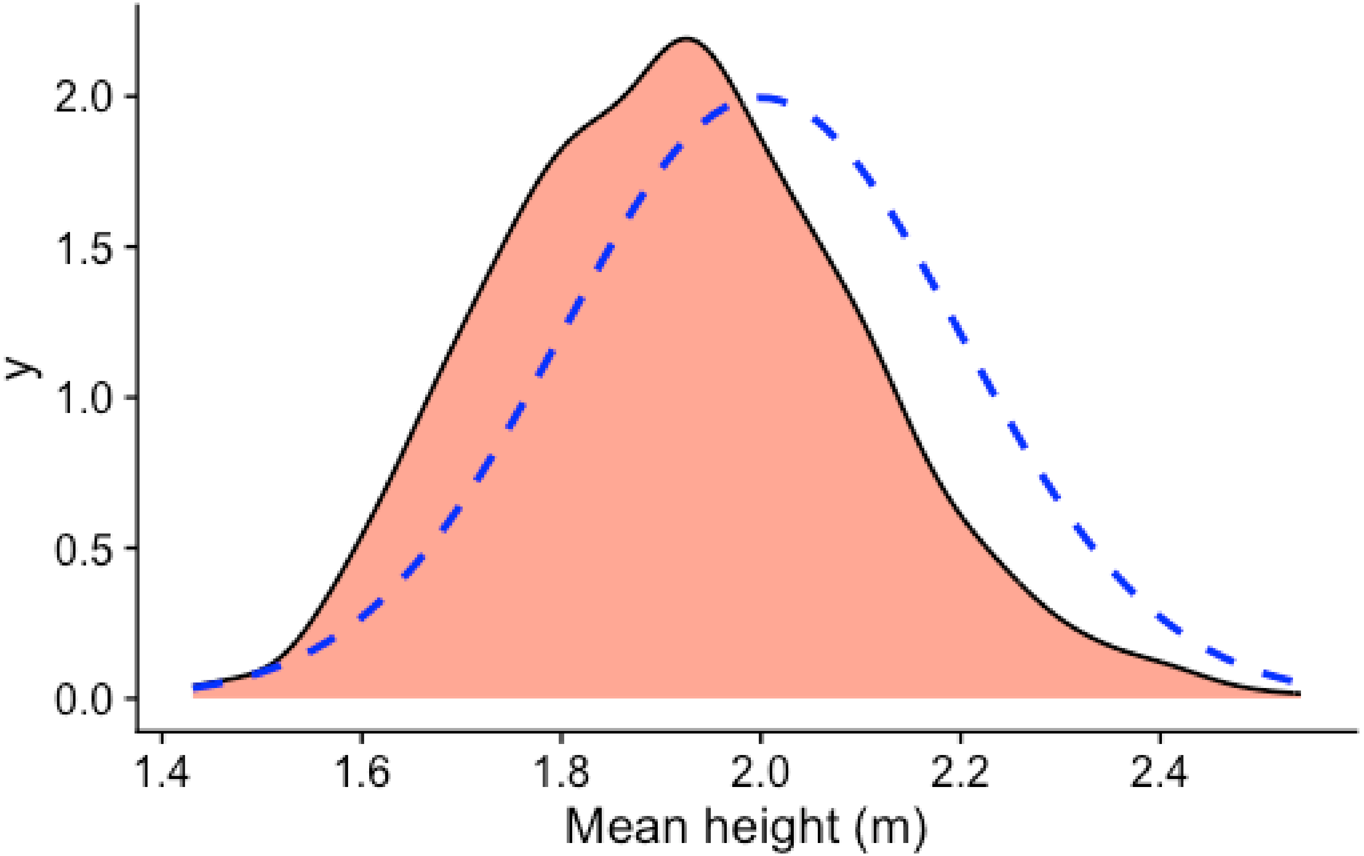
```r
skewed |>
  infer::rep_sample_n(
    size = 50,
    reps = 1000
  ) |>
  group_by(replicate) |>
  summarise(xbar = mean(x)) |>
  ggplot(aes(x = xbar)) +
  geom_density(
    fill = "orangered",
    alpha = 0.5, bins = 50
  ) +
  xlim(0, 10) +
  xlab("Mean height (m)")
```



- Each sample mean is based on 50 observations, repeated 1000 times.
- **How many samples is enough?**

Is n = 50 “normal” enough?

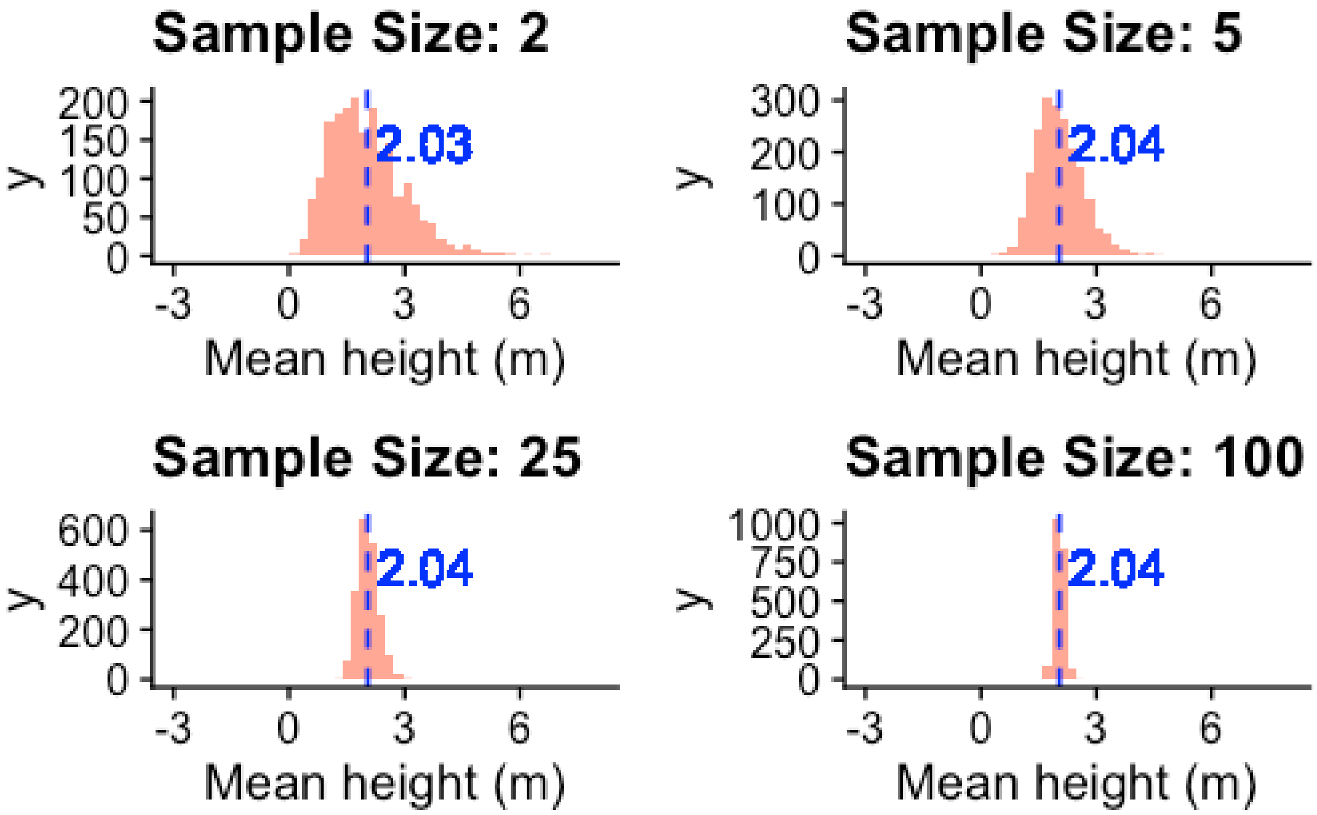
```r
skewed |>
  infer::rep_sample_n(
    size = 50,
    reps = 1000
  ) |>
  group_by(replicate) |>
  summarise(xbar = mean(x)) |>
  ggplot(aes(x = xbar)) +
  geom_density(
    fill = "orangered",
    alpha = 0.5, bins = 50
  ) +
  stat_function(
    fun = dnorm,
    args = list(
      mean = 2, # population mean for gamma(2,1)
      sd = sqrt(2) / sqrt(50) # theoretical SE for gamma(2,1)
    ),
    linewidth = 1,
    color = "blue",
    linetype = "dashed"
  ) +
  xlab("Mean height (m)")
```



- Each sample mean is based on 50 observations, repeated 1000 times.
- **How many samples is enough?**

Effect of sample size

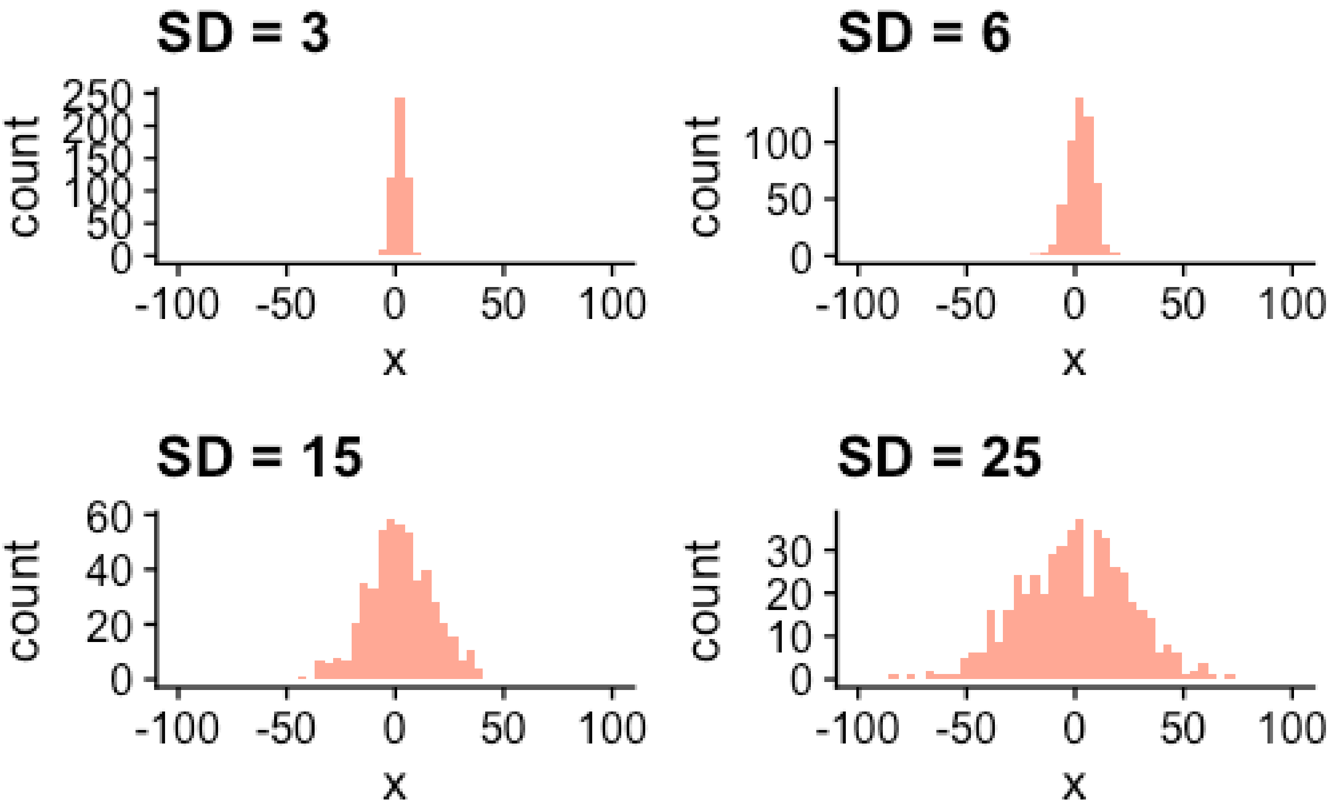
```r
library(tidymodels)
library(patchwork)
set.seed(642)

heights <- tibble(heights = rnorm(1000, 1.99, 1))
popmean <- mean(heights$heights)
sample_sizes <- c(2, 5, 25, 100)
n <- length(sample_sizes)

heights <- tibble(heights = rgamma(1000, shape = 2, scale = 1))
sample_sizes <- c(2, 5, 25, 100)
n <- length(sample_sizes)

plots <- lapply(sample_sizes, function(size) {
  df <- heights |>
    rep_sample_n(size = size, reps = 2000) |>
    group_by(replicate) |>
    summarise(xbar = mean(heights))
  mean_xbar <- mean(df$xbar)

  ggplot(df, aes(x = xbar)) +
    geom_histogram(fill = "orangered", alpha = 0.5, bins = 50) +
    geom_vline(aes(xintercept = mean_xbar), color = "blue", linetype = "dashed") +
    geom_text(aes(x = mean_xbar, label = sprintf("%.2f", mean_xbar), y = Inf), hjust = -0.1, vjust = 2, color = "blue") +
    ggtitle(paste0("Sample Size: ", size)) +
    xlab("Mean height (m)") +
    xlim(-3, 8)
})
wrap_plots(plots)
```



Larger samples give more precise estimates of the population mean. The **narrower distribution** of sample means reflects this precision, measured by the **standard error**.

Effect of variability

```
set.seed(1221)

# Define a function to generate ggplot objects
generate_plot ← function(sd) {
  data ← rnorm(500, 1.99, sd)
  p ← ggplot(data = tibble(x = data), aes(x = x)) +
    geom_histogram(fill = "orangered", alpha = 0.5, bins = 50) +
    ggtitle(paste("SD =", sd)) +
    xlim(-100, 100)
  return(p)
}

# Apply the function to a list of standard deviations
sds ← c(3, 6, 15, 25)
plots ← lapply(sds, generate_plot)

# Wrap the plots
wrap_plots(plots)
```



More variable populations (e.g. tree heights ranging from 1 m to 50 m) produce more spread in sample means. The **standard error** captures this reduced precision.

CLT drives statistical inference

The CLT enables inference about the population mean without knowing the population distribution.

- With sufficient observations, the sampling distribution of the mean is approximately normal
- The centre of that distribution is the true population mean
- The standard error measures the spread – how precise your estimate is

Standard error and confidence intervals

Standard Error of the Mean

- The standard deviation of the sampling distribution – how much sample means vary across samples
- Decreases as sample size increases, so larger samples yield more precise estimates

⚠ Formula

$$SE_{\bar{x}} = \frac{s}{\sqrt{n}}$$

- where s is the sample standard deviation
- n is the sample size

When to report SD or SE

Standard Deviation (SD)

- Describes variability in your **data**
- Does not change with sample size

Standard Error (SE)

- Describes precision of your **mean estimate**
- Shrinks as sample size increases ($SE = \frac{SD}{\sqrt{n}}$)

When reporting:

- **mean ± SE** shows how precisely you've estimated the mean — but SE can be misleadingly small with large samples, so always report n
- **mean ± SD** shows the spread of your raw data

Confidence intervals

What is a confidence interval?

- A range of values that likely contains the true population parameter, computed from your sample
- 95% is the standard choice
- Wider intervals indicate less precision

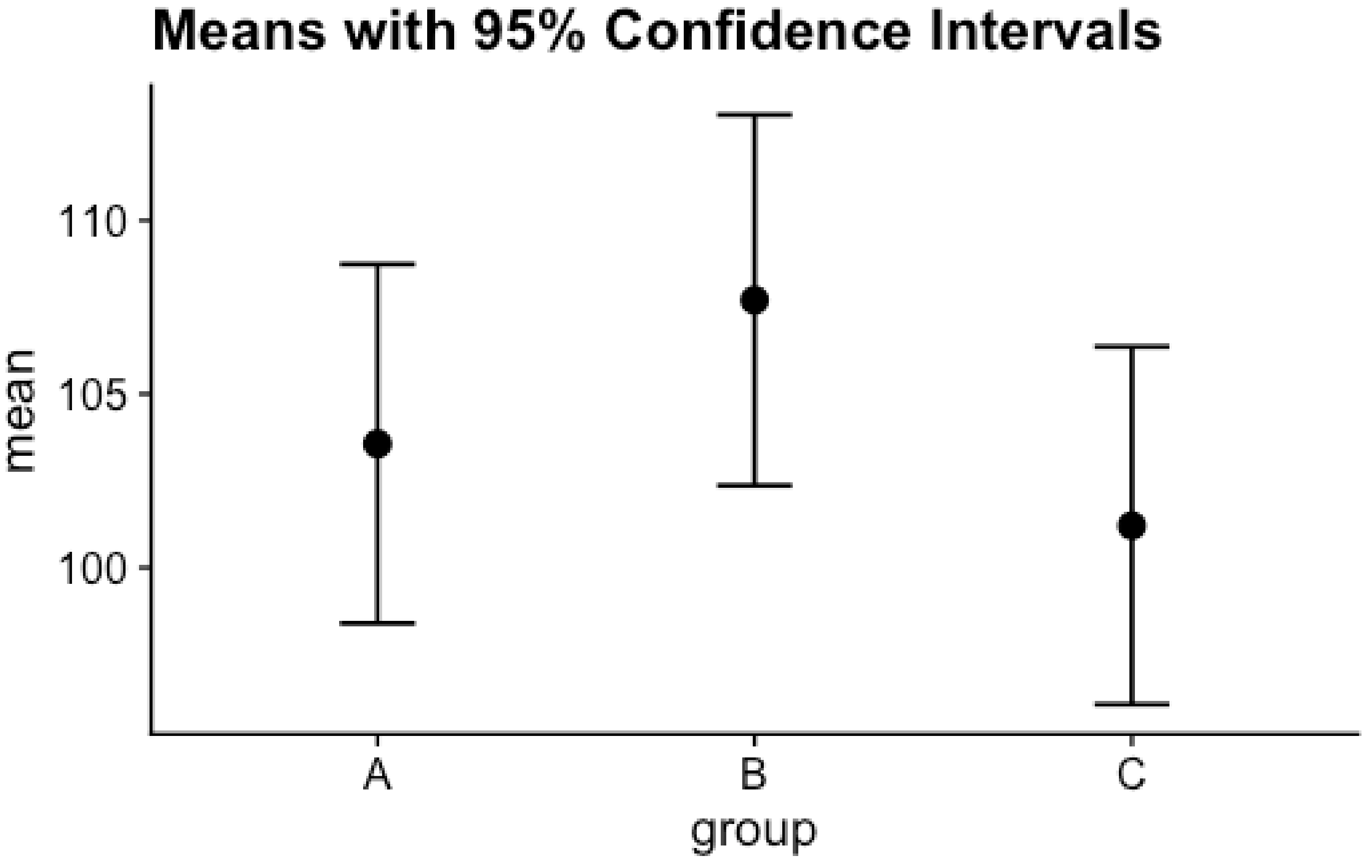
⚠ Formula for 95% CI

$$\bar{x} \pm (t_{n-1} \times SE_{\bar{x}})$$

- where t_{n-1} comes from the t-distribution (covered next lecture)

Visualising confidence intervals

```r
#| warning: false

# Generate sample data
set.seed(253)
sample_data <- data.frame(
  group = rep(c("A", "B", "C"), each = 30),
  value = c(
    rnorm(30, 100, 15),
    rnorm(30, 110, 15),
    rnorm(30, 105, 15)
  )
)

# Calculate means and CIs
ci_data <- sample_data %>%
  group_by(group) %>%
  summarise(
    mean = mean(value),
    se = sd(value) / sqrt(n()),
    ci_lower = mean - qt(0.975, n() - 1) * se,
    ci_upper = mean + qt(0.975, n() - 1) * se
  )

# Plot
ggplot(ci_data, aes(x = group, y = mean)) +
  geom_point(size = 3) +
  geom_errorbar(aes(ymin = ci_lower, ymax = ci_upper), width = 0.2) +
  ggtitle("Means with 95% Confidence Intervals")
```



We will learn more about confidence intervals in the next lecture.

Thanks for listening! Questions?