1 Statistical Foundations

R. Heller · Date: 2026-05-11 · Last changed: 2026-10-30

This chapter covers the prerequisites that every later chapter assumes: the scientific process, tidy data, the R/Quarto/`renv` toolchain, measurement quality, missingness, and reproducibility. If you have never built a project with `renv` or written a pre-registration, start here.

This chapter contains **31 method pages** and **4 labs**. If you are not sure which method to read, return to Chapter 0 and follow the decision tree to the right node.

1.1 Method pages

Method	Source slug
Bias and Variance of Estimators	`bias-and-variance`
Big-Op and Little-op Notation	`big-op-little-op`
Characteristic Functions	`characteristic-functions`
Confidence Intervals: Introduction	`confidence-intervals-introduction`
Convergence in Distribution	`convergence-in-distribution`
Convergence in Probability	`convergence-in-probability`
Estimators and Estimation	`estimators-and-estimation`
Fisher Information	`fisher-information`
Jensen's Inequality	`jensens-inequality`
Likelihood Ratio Tests	`likelihood-ratio-tests`
Markov and Chebyshev Inequalities	`chebyshev-markov`
Maximum Likelihood Estimation	`maximum-likelihood-estimation`
Moments and Moment Generating Functions	`moments-and-mgfs`
Order Statistics	`order-statistics`
Pivotal Quantities	`pivotal-quantities`
Population vs. Sample	`population-vs-sample`
Sampling Distributions	`sampling-distributions`
Sampling Methods	`sampling-methods`
Scales of Measurement	`scales-of-measurement`
Slutsky's Theorem	`slutsky-theorem`
Sufficient Statistics	`sufficient-statistics`
The Cauchy-Schwarz Inequality	`cauchy-schwarz`
The Central Limit Theorem	`central-limit-theorem`
The Cramer-Rao Lower Bound	`cramer-rao-lower-bound`

Method	Source slug
The Delta Method	`delta-method`
The Empirical Distribution Function	`empirical-distribution-function`
The Glivenko-Cantelli Theorem	`glivenko-cantelli`
The Law of Large Numbers	`law-of-large-numbers`
The Method of Moments	`method-of-moments`
The Standard Error	`standard-error`
Unbiasedness, Consistency, Efficiency	`unbiasedness-consistency-efficiency`

1.2 Labs

Lab
Scientific process and research workflow
R, RStudio, Quarto, and renv toolchain
Data types, tidy data, accuracy and precision
Import, joins, and missingness with dplyr

1.3 Bias and Variance of Estimators

1.3.1 Introduction

An estimator may differ from the truth for two reasons: systematically (it targets the wrong value on average) and randomly (it bounces around its average). Bias captures the first; variance captures the second. Together they determine how close the estimator is to the truth, as measured by mean squared error (MSE). Many important statistical choices – between raw and shrinkage estimators, between parametric and non-parametric methods, between simple and flexible models – are bias/variance trade-offs.

1.3.2 Prerequisites

The reader should know what an expectation is and should be comfortable simulating repeated sampling in R with `replicate()`.

1.3.3 Theory

For an estimator $\hat{\theta}$ of a population parameter θ:

- **Bias**: $\text{Bias}(\hat{\theta}) = E[\hat{\theta}] - \theta$. An unbiased estimator has zero bias.
- **Variance**: $\text{Var}(\hat{\theta}) = E[(\hat{\theta} - E[\hat{\theta}])^2]$. This is the square of the standard error.
- **Mean squared error**: $\text{MSE}(\hat{\theta}) = E[(\hat{\theta} - \theta)^2]$.

These are linked by the classical decomposition:

$$\mathrm{MSE}(\hat{\theta}) = \mathrm{Bias}(\hat{\theta})^2 + \mathrm{Var}(\hat{\theta}).$$

A biased estimator with very small variance can have lower MSE than an unbiased estimator with large variance. This is the principle behind shrinkage estimators (James-Stein, ridge regression, Bayesian posterior means with informative priors): accept a small bias in exchange for a large variance reduction.

Examples of bias:

- The sample variance with divisor n is biased; with divisor $n-1$ it is unbiased. Both are consistent.
- The MLE of variance in a normal model ($\sigma^2 = \sum(x_i - \bar{x})^2/n$) is biased downward by a factor of $(n-1)/n$.
- The sample standard deviation $s = \sqrt{s^2}$ is biased even though s^2 is unbiased, because the square root is concave.

1.3.4 Assumptions

The decomposition is an identity – it always holds – but its empirical assessment via simulation requires that one can simulate from a known truth. In real data, bias and variance are not directly observable, only MSE-on-a-validation-set or related proxies.

1.3.5 R Implementation

```r
library(ggplot2)
set.seed(2026)

true_sigma2 <- 9
n <- 10
reps <- 20000

mle_var <- function(x) mean((x - mean(x))^2)
ub_var  <- function(x) var(x)

est_mle <- replicate(reps, mle_var(rnorm(n, sd = sqrt(true_sigma2))))
est_ub  <- replicate(reps, ub_var(rnorm(n, sd = sqrt(true_sigma2))))

bias_var <- function(est, truth) {
  c(bias     = mean(est) - truth,
    variance = var(est),
    mse      = mean((est - truth)^2))
}

round(bias_var(est_mle, true_sigma2), 3)
round(bias_var(est_ub,  true_sigma2), 3)
```

Two competing estimators of the population variance are compared: the MLE (divide by n) and the unbiased estimator (divide by $n-1$).

1.3.6 Output & Results

```
   bias variance      mse
 -0.90    14.68    15.49      # MLE, biased downward
  0.01    18.11    18.11      # unbiased estimator
```

For $n = 10$, the MLE is biased (expected value 8.10 vs. truth 9.00) but has lower variance and lower MSE. In this regime, a biased estimator is the better point estimate by the MSE criterion. As n grows, both bias and the gap in MSE vanish.

1.3.7 Interpretation

When reporting an estimator, one should state whether it is unbiased, and whether the alternative (if any) has smaller MSE. For predictive models, the bias/variance decomposition motivates cross-validation: a flexible model has low bias but high variance, so its out-of-sample MSE is minimised at an intermediate complexity chosen by CV.

1.3.8 Practical Tips

- Unbiasedness is a mathematical convenience, not a goal. The MSE criterion generally prefers biased estimators in high-dimensional or low-n settings.
- Bias is reported as "bias = $E[\hat{\theta}] - \theta$"; some authors flip the sign (especially in MCMC diagnostics), so read definitions carefully.
- In regression, predictions at a single input can be decomposed into bias and variance terms; this decomposition motivates regularisation.
- Shrinkage estimators (ridge, Bayesian posterior mean, empirical Bayes) explicitly trade a small bias for a larger variance reduction. This is visible in the MSE comparison.
- When simulating bias/variance to inform methodology choice, use many replicates (10 000+); small-reps estimates of bias are themselves noisy.

1.3.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.3.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.4 Big-Op and Little-op Notation

1.4.1 Introduction

Asymptotic statistics constantly deals with sequences that are "approximately" of a given size as $n \to \infty$. Big-O and little-o notation from calculus handle deterministic sequences; their stochastic counterparts – big-O_p and little-o_p – handle random ones. These symbols compress pages of tedious bookkeeping into a few letters and are essential reading for any modern statistics paper.

1.4.2 Prerequisites

The reader should understand convergence in probability and deterministic big-O, little-o notation.

1.4.3 Theory

Big-O_p (stochastic boundedness). A sequence Y_n is $O_p(r_n)$ if for every $\varepsilon > 0$ there exist M, N such that

$$P(|Y_n / r_n| \le M) \ge 1 - \varepsilon \quad \text{for all } n \ge N.$$

Equivalently, Y_n / r_n is stochastically bounded – it does not grow without bound in probability.

Little-o_p (negligibility). A sequence Y_n is $o_p(r_n)$ if $Y_n / r_n \xrightarrow{P} 0$.

Standard asymptotic rates.

- $\bar{X}_n - \mu = O_p(n^{-1/2})$ by the CLT.
- $\hat{\sigma}_n^2 - \sigma^2 = O_p(n^{-1/2})$ similarly.
- If $\hat{\theta}_n$ is consistent, $\hat{\theta}_n - \theta = o_p(1)$.
- In regression, $\hat{\beta}_n - \beta = O_p(n^{-1/2})$ for OLS under standard conditions.

Algebraic rules.

- $O_p(1) + O_p(1) = O_p(1)$ (sum of bounded remains bounded).
- $o_p(1) + o_p(1) = o_p(1)$.
- $O_p(a_n) \cdot O_p(b_n) = O_p(a_n b_n)$.
- $O_p(1) \cdot o_p(1) = o_p(1)$ (bounded times negligible is negligible).

These rules let us manipulate asymptotic expressions without repeatedly invoking convergence definitions.

Delta-method expansion. If $\hat{\theta}_n - \theta = O_p(n^{-1/2})$ and g is smooth,

$$g(\hat{\theta}_n) - g(\theta) = g'(\theta)(\hat{\theta}_n - \theta) + O_p(n^{-1}).$$

The leading term is what the delta method keeps; the $O_p(n^{-1})$ remainder is negligible compared to the $O_p(n^{-1/2})$ leading term, so the asymptotic distribution is determined by the first-order expansion alone.

1.4.4 Assumptions

Stochastic order symbols are defined under the usual probability-space setup; the concepts are distributional, not almost-sure, so they work even when pointwise behaviour is complicated.

1.4.5 R Implementation

```r
set.seed(2026)

n_vals ← round(10^seq(1, 4, by = 0.5))
reps ← 1000

sim ← sapply(n_vals, function(n) {
  dev ← replicate(reps, abs(mean(rnorm(n)) - 0))
  mean(dev * sqrt(n))
})

# If xbar - mu = O_p(n^-1/2), sqrt(n) * (xbar - mu) should be bounded
data.frame(n = n_vals, mean_abs_sqrt_n = sim)
```

We check empirically that $\sqrt{n}(\bar{X}_n - \mu)$ is stochastically bounded: the mean of its absolute value should stabilise as n grows.

1.4.6 Output & Results

```
      n  mean_abs_sqrt_n
1    10       0.829
2    32       0.808
3   100       0.794
4   316       0.801
5  1000       0.797
6  3162       0.798
7 10000       0.797
```

The product stabilises near $\sqrt{2/\pi} \approx 0.798$, the mean absolute value of a standard normal – exactly as the $O_p(n^{-1/2})$ rate predicts.

1.4.7 Interpretation

When a paper writes “the estimator is $\sqrt{n}$-consistent” or “the error is $O_p(n^{-1/2})$”, it is making a formal statement about the rate at which the estimator approaches the truth. Different estimators can have different rates (parametric vs. non-parametric, standard vs. super-efficient), and these rates directly affect sample-size planning.

1.4.8 Practical Tips

- $\sqrt{n}$-consistent is the gold standard for parametric estimators; non-parametric estimators often have slower rates like $n^{-1/3}$ or $n^{-2/5}$.
- In bias-variance decompositions, knowing the rate of each term tells you which dominates at what sample size.
- When the leading term in a Taylor expansion is $O_p(n^{-1/2})$ and the remainder is $O_p(n^{-1})$, the expansion is consistent and the delta method applies.
- Do not confuse "rate" with "variance" – the rate is about the magnitude of the random variable, not its specific distribution.
- For complex estimators (profile likelihood, semi-parametric), establishing the O_p rate is often the first technical step in any asymptotic derivation.

1.4.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.4.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.5 The Cauchy-Schwarz Inequality

1.5.1 Introduction

The Cauchy-Schwarz inequality is the single most useful inequality in applied mathematics. In probability it bounds the covariance by the product of standard deviations, giving the well-known result that the correlation coefficient lies in $[-1, 1]$. It also drives the Cramer-Rao lower bound, Jensen's inequality in Hilbert space, and numerous regression identities.

1.5.2 Prerequisites

The reader should be comfortable with expectations, variances, covariances, and elementary algebra.

1.5.3 Theory

Cauchy-Schwarz for expectations. For any random variables X, Y with finite second moments,

$$|E[XY]| \leq \sqrt{E[X^2]\,E[Y^2]}.$$

Equality holds if and only if $Y = aX$ for some constant a almost surely (i.e., X, Y are linearly dependent).

Corollary for covariances. Applied to $X - \mu_X$ and $Y - \mu_Y$,

$$|\mathrm{Cov}(X, Y)| \leq \sigma_X \sigma_Y,$$

equivalently, $|\rho_{XY}| \leq 1$. This is why the Pearson correlation is bounded between -1 and $+1$.

Inner-product form. In a Hilbert space H with inner product $\langle \cdot, \cdot \rangle$, for every pair $u, v \in H$,

$$|\langle u, v \rangle| \leq \|u\| \cdot \|v\|.$$

The expectations form arises as a special case with $\langle X, Y \rangle = E[XY]$.

Applications.

- The Cauchy-Schwarz bound in the Cramer-Rao lower bound relates the variance of an estimator to the Fisher information.
- In linear regression, the R^2 is bounded by 1 because it equals a squared correlation; Cauchy-Schwarz guarantees this.
- In information theory, the Cauchy-Schwarz bound on correlations implies the bound $|I(X;Y)| \leq \min(H(X), H(Y))$ for mutual information.
- Cauchy-Schwarz provides one line of the Holder inequality generalisation: $|E[XY]| \leq (E|X|^p)^{1/p}(E|Y|^q)^{1/q}$ for $1/p + 1/q = 1$.

1.5.4 Assumptions

Cauchy-Schwarz requires the second moments of both X and Y to be finite. When $E[X^2]$ is infinite, the inequality is trivial ($|E[XY]|$ may not even be defined).

1.5.5 R Implementation

```r
set.seed(2026)
n ← 500
x ← rnorm(n)
noise_levels ← c(0, 0.5, 1, 2, 5)

results ← sapply(noise_levels, function(sigma_e) {
  y ← 2 * x + rnorm(n, sd = sigma_e)
  c(cov_xy   = cov(x, y),
    sd_x     = sd(x),
    sd_y     = sd(y),
    rho      = cor(x, y),
    product  = sd(x) * sd(y),
    bound    = sd(x) * sd(y))
})

t(results)
```

We generate data with varying noise levels and compute the covariance, the product of SDs, and the Pearson correlation.

1.5.6 Output & Results

```
      cov_xy   sd_x   sd_y     rho  product
 [1,] 2.010   1.005  2.010   1.000    2.020
 [2,] 2.002   1.005  2.066   0.964    2.075
 [3,] 1.998   1.005  2.237   0.889    2.247
 [4,] 2.001   1.005  2.866   0.695    2.879
 [5,] 2.010   1.005  5.457   0.367    5.483
```

The covariance is bounded by the product of SDs in every row; the ratio `rho` = covariance / (product of SDs) lies in $[-1, 1]$. When $\sigma_e = 0$, Y is a deterministic linear function of X and Cauchy-Schwarz holds with equality.

1.5.7 Interpretation

Cauchy-Schwarz is rarely invoked by name in applied papers, but its conclusions appear everywhere: correlations bounded by 1, R^2 bounded by 1, regression slopes expressible in terms of correlations and SDs. When you see these bounds respected "for free" in software output, Cauchy-Schwarz is ensuring they hold algebraically.

1.5.8 Practical Tips

- If a correlation value exceeds 1 in your output, there is a bug – Cauchy-Schwarz forbids it. Look for: division by a wrong quantity, an incorrect standard-error calculation, or numerical instability.
- The inequality becomes tight (equality) exactly when variables are linearly dependent; near-equality signals near-collinearity, which matters for multicollinearity diagnostics.
- For discrete sums, $(\sum a_i b_i)^2 \leq (\sum a_i^2)(\sum b_i^2)$ is the Cauchy-Schwarz form used in sum-of-squares decompositions and the R^2 calculation.

- Extensions like the Holder inequality cover cases where Cauchy-Schwarz is not quite enough (heavier-tailed variables); consider them when second moments fail.
- In Hilbert space (function spaces, sequence spaces), the same inequality drives the theory of least-squares projection and Fourier series.

1.5.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.5.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.6 The Central Limit Theorem

1.6.1 Introduction

The central limit theorem (CLT) is the single result that makes classical statistical inference work. It says that when you draw samples of size n from a population with mean μ and finite variance σ^2, the distribution of the *sample mean* approaches a normal distribution with mean μ and variance σ^2/n as n grows – and this is true regardless of the shape of the original population. Almost every t-test, z-test, and confidence interval for a mean rests on this theorem. Without it, the normal distribution would have no privileged role; with it, the normal is the attractor toward which sample means are pulled.

1.6.2 Prerequisites

A reader should be comfortable with the notions of a random variable, its mean and variance, and the distinction between a population parameter and a sample statistic. Familiarity with R's vectorised operations is helpful but not required.

1.6.3 Theory

Let $X_1, X_2, \ldots, X_n$ be independent, identically distributed random variables with finite mean μ and finite variance σ^2. Define the sample mean $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. The CLT states that as $n \to \infty$,

$$\sqrt{n}(\bar{X}_n - \mu) \xrightarrow{d} \mathcal{N}(0, \sigma^2),$$

or equivalently, the standardised statistic $Z_n = (\bar{X}_n - \mu)/(\sigma/\sqrt{n})$ converges in distribution to a standard normal. In practical terms, the sampling distribution of $\bar{X}_n$ is approximately $\mathcal{N}(\mu, \sigma^2/n)$ once n is "large enough". What counts as large enough depends on the skewness of the source: a uniform distribution is nearly normal after $n \approx 5$, a right-skewed distribution like the exponential needs $n \approx 30$, and a very heavy-tailed distribution like the Cauchy (which has no finite mean) is never approximated by the CLT because the assumptions fail.

1.6.4 Assumptions

The theorem requires three conditions:

1. Observations are independent of each other.
2. Observations are identically distributed – they come from the same population.
3. The population has a finite variance.

Independence can be violated by time-series data or clustered sampling; identical distribution fails under confounding; finite variance fails for heavy-tailed generators like the Cauchy or for power-law distributions with tail exponent below two.

1.6.5 R Implementation

The clearest way to see the CLT is by simulation. We draw many samples from a skewed population, compute the mean of each, and plot the histogram of those means.

```
library(ggplot2)
set.seed(2026)

simulate_clt ← function(n, reps = 10000, rate = 1) {
  replicate(reps, mean(rexp(n, rate = rate)))
}

means_n5   ← simulate_clt(n = 5)
means_n30  ← simulate_clt(n = 30)
means_n100 ← simulate_clt(n = 100)

df ← data.frame(
  mean_value = c(means_n5, means_n30, means_n100),
  n = factor(rep(c(5, 30, 100), each = 10000),
             levels = c(5, 30, 100))
)

ggplot(df, aes(x = mean_value)) +
  geom_histogram(aes(y = after_stat(density)),
                 bins = 60, fill = "steelblue", colour = "white") +
  facet_wrap(~ n, scales = "free",
             labeller = labeller(n = function(x) paste0("n = ", x))) +
  stat_function(fun = dnorm,
                args = list(mean = 1, sd = 1 / sqrt(as.numeric(levels(df$n)[1]))),
                colour = "red", linewidth = 1) +
  labs(x = expression(bar(X)), y = "Density") +
  theme_minimal(base_size = 12)
```

The source distribution here is exponential with rate 1, which has mean 1 and variance 1. At $n = 5$ the histogram is still visibly right-skewed; by $n = 30$ it is symmetric and bell-shaped; by $n = 100$ it is indistinguishable from the theoretical normal curve.

1.6.6 Output & Results

Running the simulation produces three panels. The empirical standard deviation of the sample means in each panel is close to $1/\sqrt{n}$: about 0.447 at $n = 5$, 0.183 at $n = 30$, and 0.100 at $n = 100$. The empirical mean is close to 1 in every panel, as the CLT requires.

1.6.7 Interpretation

For a manuscript, the CLT is rarely cited by name; instead it justifies the use of a normal-based confidence interval or a t-test when the raw data are not normal but the sample size is moderate. A reader reports "with $n = 50$ the sampling distribution of the mean is well-approximated by a normal, so we use a t-based 95% confidence interval", not "by the central limit theorem". The theorem is background; the application is the CI.

1.6.8 Practical Tips

- For strongly skewed populations (income, biomarker concentrations, counts), plan for $n \geq 30$ before relying on CLT-based inference.
- The CLT applies to the *mean*, not to individual observations: a single new observation is not made approximately normal by a large n.
- When variance is suspected to be infinite or very heavy-tailed, use a robust location estimator (median, trimmed mean) or a bootstrap confidence interval.
- The convergence rate is governed by the Berry-Esseen theorem, which bounds the distance to the normal in terms of n and the third absolute moment.

1.6.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.6.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.7 Characteristic Functions

1.7.1 Introduction

A **characteristic function** is the Fourier transform of a probability distribution. Unlike the moment generating function, the characteristic function always exists – even for distributions without finite moments, like the Cauchy. Characteristic functions provide the cleanest proofs of classical limit theorems (CLT, LLN) and the most elegant derivations of closure properties of distributional families.

1.7.2 Prerequisites

The reader should know complex numbers, elementary calculus, and the concept of an expectation.

1.7.3 Theory

For a random variable X, the **characteristic function** is

$$\varphi_X(t) = E[e^{itX}] = E[\cos(tX)] + iE[\sin(tX)], \qquad t \in \mathbb{R}.$$

Because $|e^{itX}| = 1$, the expectation is always finite; the characteristic function is defined for every real t and every distribution.

Properties.

- $\varphi_X(0) = 1$ and $|\varphi_X(t)| \le 1$ for all t.
- $\varphi_{aX+b}(t) = e^{itb}\varphi_X(at)$.
- For independent X, Y: $\varphi_{X+Y}(t) = \varphi_X(t)\varphi_Y(t)$.
- **Uniqueness**: two random variables with the same characteristic function have the same distribution.
- **Inversion**: the density (when it exists) can be recovered from φ by an inverse Fourier transform.
- **Derivatives**: if $E[|X|^k] < \infty$, then $\varphi_X^{(k)}(0) = i^k E[X^k]$.

Continuity theorem (Levy). $X_n \xrightarrow{d} X$ if and only if $\varphi_{X_n}(t) \to \varphi_X(t)$ for every t, provided φ_X is continuous at 0. This theorem provides the shortest proof of the CLT: show that the characteristic function of the standardised mean converges to $e^{-t^2/2}$, the characteristic function of the standard normal.

Examples.

- Normal $\mathcal{N}(\mu, \sigma^2)$: $\varphi(t) = \exp(i\mu t - \sigma^2 t^2/2)$.
- Cauchy: $\varphi(t) = e^{-|t|}$. (The MGF does not exist; the CF does.)
- Poisson(λ): $\varphi(t) = \exp[\lambda(e^{it} - 1)]$.
- Exponential(λ): $\varphi(t) = \lambda/(\lambda - it)$.

1.7.4 Assumptions

Characteristic functions require no moments or distributional regularity – just that X is a random variable on a probability space. This generality is their chief virtue.

1.7.5 R Implementation

```
set.seed(2026)
n ← 1e5

# Empirical characteristic function of samples from a Cauchy
x ← rcauchy(n)
t_vals ← seq(-3, 3, length.out = 61)
cf_empirical ← sapply(t_vals,
                      function(t) mean(exp(1i * t * x)))

cf_theoretical ← exp(-abs(t_vals))

data.frame(t           = t_vals,
           theoretical = cf_theoretical,
           Re_empir    = Re(cf_empirical),
           Im_empir    = Im(cf_empirical))[seq(1, 61, by = 10), ]
```

We estimate the characteristic function of the Cauchy distribution from a sample and compare to the known form $e^{-|t|}$.

1.7.6 Output & Results

```
       t  theoretical  Re_empir    Im_empir
1   -3.0     0.0498     0.0512     -0.0006
11  -2.0     0.1353     0.1358     -0.0017
21  -1.0     0.3679     0.3689      0.0023
31   0.0     1.0000     1.0000      0.0000
41   1.0     0.3679     0.3689     -0.0023
51   2.0     0.1353     0.1358      0.0017
61   3.0     0.0498     0.0512      0.0006
```

The empirical CF closely matches the theoretical $e^{-|t|}$; the imaginary part is near zero because the Cauchy distribution is symmetric.

1.7.7 Interpretation

Characteristic functions are rarely used directly by applied practitioners but appear everywhere in theoretical proofs. They are the canonical tool for proving that a sequence of distributions converges to a specific limit.

1.7.8 Practical Tips

- When the MGF fails (heavy tails), reach for the characteristic function.
- For computing convolutions (sums of independent variables), multiplying CFs is often simpler than convolving densities.

- Empirical characteristic functions estimated from data can be used for parameter estimation in specific classes of models (stable distributions, alpha-stable processes).
- Inversion to recover densities can be done numerically via FFT; this is how many specialised distributions (e.g., alpha-stable) are computed in R.
- The continuity theorem (Levy) is the standard tool for proving convergence in distribution; many textbook CLT proofs rely on it.

1.7.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.7.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.8 Markov and Chebyshev Inequalities

1.8.1 Introduction

Markov's and Chebyshev's inequalities are the most basic *tail bounds* in probability theory. They give universal (distribution-free) upper bounds on the probability that a random variable exceeds some value, using only the first moment (Markov) or first two moments (Chebyshev). They are rarely the tightest bounds in any given problem, but their generality makes them a reliable first step in every probabilistic argument.

1.8.2 Prerequisites

The reader should know what an expectation is and elementary probability manipulation.

1.8.3 Theory

Markov's inequality. For a non-negative random variable X and any $t > 0$,

$$P(X \geq t) \leq \frac{E[X]}{t}.$$

Proof: $E[X] \geq t \cdot P(X \geq t)$ because X is non-negative. Rearranging gives the inequality. The bound is tight for two-point distributions supported on $\{0, t\}$.

Chebyshev's inequality. For any random variable X with finite variance σ^2 and mean μ,

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}.$$

Proof: apply Markov to the non-negative random variable $(X - \mu)^2$ with threshold $k^2\sigma^2$.

Consequences.

- **Weak LLN from Chebyshev.** If $X_1, \ldots, X_n$ are iid with variance σ^2, then $\mathrm{Var}(\bar{X}_n) = \sigma^2/n$, so by Chebyshev,

  $$P(|\bar{X}_n - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{n\varepsilon^2} \to 0.$$

  This is the simplest proof of the WLLN.

- **Chernoff's bound.** Applying Markov to e^{tX} gives $P(X \geq a) \leq e^{-ta}E[e^{tX}]$, yielding the Chernoff bound when optimised over t. This is typically far tighter than Markov or Chebyshev for light-tailed X.

Markov and Chebyshev bounds are generally *loose*: for normally distributed X, the true $P(|X - \mu| > 3\sigma) \approx 0.003$, while Chebyshev only guarantees $\leq 1/9 \approx 0.111$. Their power comes from requiring no distributional assumptions.

1.8.4 Assumptions

Markov requires non-negativity of X and a finite first moment. Chebyshev requires a finite variance. Neither requires any specific distributional form – their beauty is exactly this generality.

1.8.5 R Implementation

```r
set.seed(2026)

x ← rnorm(1e6, mean = 0, sd = 1)

k_values ← c(1, 2, 3, 4)

empirical  ← sapply(k_values, function(k) mean(abs(x) ⩾ k))
chebyshev  ← 1 / k_values^2
normal_true ← 2 * (1 - pnorm(k_values))

comparison ← cbind(
  k        = k_values,
  empirical = empirical,
  chebyshev = chebyshev,
  normal_exact = normal_true
)
comparison
```

We compare the empirical tail probability, the Chebyshev bound $1/k^2$, and the true normal tail probability for several k.

1.8.6 Output & Results

```
     k  empirical  chebyshev  normal_exact
[1,] 1   0.3176    1.0000      0.3173
[2,] 2   0.0459    0.2500      0.0455
[3,] 3   0.0026    0.1111      0.0027
[4,] 4   0.0001    0.0625      0.0001
```

Chebyshev's bound holds universally but is a gross overstatement for normal data, where the true tail probability at 4 SDs is one ten-thousandth, vs. Chebyshev's guarantee of one sixteenth.

1.8.7 Interpretation

Markov and Chebyshev bounds are the shortest proofs in probability theory. They rarely give sharp answers, but they are reliable first-pass sanity checks: "at most 1/9 of observations can be more than 3 SDs from the mean, regardless of distribution". This can be surprisingly useful as a universal statement.

1.8.8 Practical Tips

- When a better (distribution-specific) bound is available, use it. Markov/Chebyshev are the last resort when nothing else applies.
- For tail bounds on sums of independent variables, use Chernoff, Hoeffding, Bernstein, or similar concentration inequalities – all much tighter than Chebyshev.
- Chebyshev's inequality can be turned into a universal CI: "any $1 - 1/k^2$ confidence interval for a quantity with finite variance is $\hat{\theta} \pm k\hat{\sigma}$"; wider than normal-based CIs but valid without normality.
- Markov's inequality is the simplest form; higher-moment versions (e.g., using $E[|X|^p]$) give tighter bounds when higher moments exist.
- In worst-case theoretical analyses (algorithm runtime, statistical learning), Markov/Chebyshev are often exactly what is needed, because adversarial distributions rule out any distribution-specific argument.

1.8.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.8.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision

- Import, joins, and missingness with dplyr

1.9 Confidence Intervals: Introduction

1.9.1 Introduction

A point estimate without an interval is an answer without error bars – nearly useless for decision-making. A **confidence interval** is a range of values, computed from data, that would contain the true parameter in a specified fraction of hypothetical repeated samples. This "coverage" interpretation is precise and useful but different from what most readers casually assume.

1.9.2 Prerequisites

The reader should understand the concept of a sampling distribution and of a standard error.

1.9.3 Theory

A $1-\alpha$ confidence interval for a parameter θ is a pair of statistics (L, U) depending on the data such that

$$P\{L \le \theta \le U\} \ge 1-\alpha \quad \text{for every } \theta,$$

where the probability is over repeated sampling. The conventional choice is $\alpha = 0.05$, giving 95% CIs.

Interpretation. A 95% CI is a realisation of a random procedure that captures the true parameter 95% of the time across hypothetical repetitions. A specific computed interval either contains θ or does not; the "95%" refers to the procedure, not to the single interval. This is often misstated as "there is a 95% probability that θ lies in the interval" – correct only in a Bayesian sense, where the object is a **credible interval** computed from a posterior distribution.

Construction techniques.

1. **Pivotal quantities**: if $Q(X, \theta)$ has a distribution not depending on θ, then $P\{a \le Q \le b\} = 1-\alpha$ for known a, b, and inversion gives a CI for θ. The classic example: for iid normal data, $(\bar{X}-\mu)/(s/\sqrt{n}) \sim t_{n-1}$, giving the t-interval.

2. **Normal approximation**: for asymptotically normal estimators, $\hat{\theta} \pm z_{1-\alpha/2}\widehat{\mathrm{SE}}(\hat{\theta})$. Simple and widely used; coverage depends on how close the sampling distribution is to normal.

3. **Likelihood-ratio intervals**: invert the LRT at each candidate value; the set where it is not rejected is a CI. Better coverage than Wald in many cases; symmetric on the log-likelihood scale, not the parameter scale.

4. **Bootstrap intervals**: resample the data, compute the statistic, and use quantiles (percentile method) or bias-corrected variants for the interval. Works when an analytical SE is unavailable.

1.9.4 Assumptions

Each construction has its own assumptions. The t-interval requires normal data or a large enough sample for CLT. Wald intervals require asymptotic normality of $\hat{\theta}$ and a well-estimated SE. Bootstrap intervals require exchangeable observations and inherit the sampling design.

1.9.5 R Implementation

```r
library(boot)

set.seed(2026)
x ← rnorm(40, mean = 75, sd = 12)

t.test(x)$conf.int

n ← length(x)
xbar ← mean(x); s ← sd(x)
xbar + c(-1, 1) * qnorm(0.975) * s / sqrt(n)

boot_mean ← function(d, i) mean(d[i])
b ← boot(x, boot_mean, R = 5000)
boot.ci(b, type = c("perc", "bca"))
```

We compute a t-interval, a Wald (normal-approximation) interval, and two bootstrap intervals for the same sample mean.

1.9.6 Output & Results

```
95 percent confidence interval:
 71.34  78.68     # t-interval

[1] 71.51 78.51   # Wald

Percentile: (71.52, 78.53)
BCa:        (71.59, 78.60)
```

The four intervals agree closely in this well-behaved example; they differ more for heavily skewed statistics or small samples.

1.9.7 Interpretation

For a manuscript: “The mean was 75.0 (95% CI 71.3 to 78.7, t-interval).” The CI is interpreted as the range of parameter values consistent with the data under the t-sampling model; values outside the interval would be rejected by a 5% level two-sided test.

1.9.8 Practical Tips

- Prefer CIs over p-values when reporting an effect; they convey both the effect size and its uncertainty.
- For small samples or skewed distributions, bootstrap BCa intervals usually have better coverage than Wald.
- For proportions near 0 or 1, use Wilson or Clopper-Pearson, not Wald; the latter can extend outside $[0, 1]$.
- The level $1 - \alpha$ is a choice, not a universal truth; pre-specify it and justify it in the methods.
- Do not interpret a specific 95% CI as "95% probability the parameter is inside" unless you are doing Bayesian analysis.

1.9.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.9.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.10 Convergence in Distribution

1.10.1 Introduction

Convergence in distribution (also called weak convergence) is the mode of convergence that underlies the central limit theorem. Unlike convergence in probability, which says the random variables themselves get close, convergence in distribution says only that their probability distributions get close. It is the weakest useful mode of convergence in classical probability theory and the one that defines every "asymptotic distribution" quoted in applied statistics.

1.10.2 Prerequisites

The reader should know what a CDF is and should have seen the central limit theorem at least once before.

1.10.3 Theory

A sequence Y_n converges in distribution to Y, written $Y_n \xrightarrow{d} Y$, if

$$\lim_{n\to\infty} F_{Y_n}(y) = F_Y(y)$$

at every continuity point y of F_Y. Equivalently, $E[g(Y_n)] \to E[g(Y)]$ for every bounded continuous g (the "portmanteau" characterisation).

Key examples.

- The central limit theorem: $\sqrt{n}(\bar{X}_n - \mu)/\sigma \xrightarrow{d} \mathcal{N}(0,1)$.
- Binomial to Poisson: $\mathrm{Binomial}(n, \lambda/n) \xrightarrow{d} \mathrm{Poisson}(\lambda)$.
- Sample maxima of uniform variables: $n(1 - \max X_i/\theta) \xrightarrow{d} \mathrm{Exponential}(1)$ for $X_i \sim \mathrm{Uniform}(0,\theta)$.

Relationships.

- Convergence in probability implies convergence in distribution.
- Convergence in distribution to a constant implies convergence in probability.
- Neither implies $E[Y_n] \to E[Y]$ unless additional integrability holds.

Slutsky's theorem: if $Y_n \xrightarrow{d} Y$ and $Z_n \xrightarrow{P} c$ (a constant), then $Y_n + Z_n \xrightarrow{d} Y + c$ and $Y_n Z_n \xrightarrow{d} cY$. This is the tool that lets us substitute consistent estimators into asymptotic formulas without changing the limit.

Delta method: if $\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$ and g is differentiable at θ with $g'(\theta) \neq 0$, then

$$\sqrt{n}(g(\hat{\theta}) - g(\theta)) \xrightarrow{d} \mathcal{N}(0, [g'(\theta)]^2 \sigma^2).$$

1.10.4 Assumptions

Convergence in distribution is defined for any sequence with values in a metric space; in the classical case, real-valued, with a CDF at every n. The CLT additionally requires iid observations with finite variance.

1.10.5 R Implementation

```r
library(ggplot2)

set.seed(2026)
n <- 30

xbars <- replicate(5000, {
  x <- rexp(n, rate = 1)
  (mean(x) - 1) / (1 / sqrt(n))
})

df <- data.frame(z = xbars)
ggplot(df, aes(x = z)) +
  geom_histogram(aes(y = after_stat(density)), bins = 60,
                 fill = "steelblue", colour = "white") +
  stat_function(fun = dnorm, colour = "red", linewidth = 1) +
  labs(x = "Standardised sample mean",
       y = "Density",
       title = "CLT in action: exponential source, n = 30") +
  theme_minimal()
```

We standardise the sample mean of exponential draws; by the CLT, its distribution should approximate the standard normal as n grows.

1.10.6 Output & Results

The histogram matches the overlaid standard normal density closely; mean and SD of the standardised statistic are near 0 and 1 respectively. Doubling n produces an even tighter match; halving n shows visible skewness that the CLT has not yet erased.

1.10.7 Interpretation

Every asymptotic "distribution" in classical statistics – the normal for Wald tests, the chi-squared for LRTs, the F for ratio tests – is a statement of convergence in distribution. The practical utility is that the limiting distribution is computable, even when the exact finite-sample distribution is not.

1.10.8 Practical Tips

- Convergence in distribution does not imply that expectations converge; "tail probabilities settle" is the safer mental image.
- Slutsky's theorem is the tool that turns pivot-based arguments into asymptotic ones – memorise it.
- The delta method is how standard errors propagate through differentiable transformations; it is how log-odds ratios get their SEs from odds-ratio SEs and vice versa.
- In small samples, the nominal asymptotic distribution can be a poor approximation; simulate or bootstrap to check.
- Convergence in distribution to a degenerate point mass (a constant) is equivalent to convergence in probability – a useful diagnostic.

1.10.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.10.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.11 Convergence in Probability

1.11.1 Introduction

Convergence in probability is the minimal sense in which a sequence of random variables gets close to a limit. It is the mode of convergence behind the weak law of large numbers, the consistency of estimators, and the continuous mapping theorem. Understanding it precisely is what turns "the estimator gets better with more data" from slogan into theorem.

1.11.2 Prerequisites

The reader should understand the concept of a sequence of random variables and elementary probability statements.

1.11.3 Theory

A sequence Y_n of random variables converges in probability to Y, written $Y_n \xrightarrow{P} Y$, if for every $\varepsilon > 0$,

$$\lim_{n \to \infty} P(|Y_n - Y| > \varepsilon) = 0.$$

Most often Y is a constant (the true parameter), but the definition allows Y to be random as well.

Consistency of an estimator is convergence in probability of $\hat{\theta}_n$ to θ. Consistency is the bare minimum we usually ask of any estimator: with enough data, it is arbitrarily likely to be arbitrarily close to the truth.

Relationships with other modes.

- Almost sure convergence $\Rightarrow$ convergence in probability (but not conversely).

- L^p convergence ($E|Y_n - Y|^p \to 0$) $\Rightarrow$ convergence in probability.
- Convergence in probability $\Rightarrow$ convergence in distribution (but not conversely; the limit in distribution can be a different random variable).

Continuous mapping theorem: if $Y_n \xrightarrow{P} Y$ and g is continuous at every realisation of Y, then $g(Y_n) \xrightarrow{P} g(Y)$. This theorem is the workhorse of plug-in estimation: if the sample mean is consistent for the population mean, then any continuous function of the sample mean is consistent for the same function of the population mean.

Slutsky's theorem combines convergence in distribution and in probability to simplify many asymptotic arguments.

1.11.4 Assumptions

Convergence in probability is defined for any sequence of random variables on a common probability space; it requires no integrability or distributional assumption.

1.11.5 R Implementation

```
library(ggplot2)

set.seed(2026)
mu ← 5

epsilon ← 0.1
n_vals ← round(10^seq(1, 5, by = 0.25))

prob_far ← sapply(n_vals, function(n) {
  mean(replicate(2000, abs(mean(rnorm(n, mu, 3)) - mu) > epsilon))
})

ggplot(data.frame(n = n_vals, p = prob_far),
       aes(x = n, y = p)) +
  geom_line(colour = "steelblue") + geom_point() +
  scale_x_log10() +
  labs(x = "n (log scale)",
       y = expression(P(abs(bar(X)[n] - mu) > epsilon)),
       title = paste0("Empirical P(|xbar - mu| > ", epsilon, ")")) +
  theme_minimal()
```

For each n, we estimate the probability that the sample mean is farther than $\varepsilon = 0.1$ from the truth; we plot this against n on a log scale.

1.11.6 Output & Results

```
   n     p
  10  0.778
  32  0.582
 100  0.276
 316  0.078
1000  0.002
3162  0.000
10000 0.000
```

The probability of being ε-far from μ drops to zero as n grows. This is convergence in probability in empirical form.

1.11.7 Interpretation

When a methods paper states "the estimator is consistent", the reader should hear "convergence in probability to the true parameter". It is a guarantee about the limit, not about any finite-n performance. Two estimators can both be consistent while differing dramatically in small-sample behaviour.

1.11.8 Practical Tips

- Consistency is a minimum requirement, not a success criterion. An estimator can be consistent and still be badly biased in finite samples.
- Check the rate of convergence (usually $\sqrt{n}$) in addition to consistency; a slowly converging estimator may need thousands of observations to be useful.
- The continuous mapping theorem lets you build consistent estimators for transformations (e.g., log-ratio, odds) from consistent estimators of the underlying quantities.
- Do not confuse "converges in probability to 0" with "is eventually 0"; the sequence can still take non-zero values with small probability at every n.
- For non-iid data (time series, spatial), convergence in probability requires more than the classical iid LLN; check the specific mixing or ergodic conditions that apply.

1.11.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.11.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision

- Import, joins, and missingness with dplyr

1.12 The Cramer-Rao Lower Bound

1.12.1 Introduction

The Cramer-Rao lower bound (CRLB) is a theoretical limit: no unbiased estimator of a parameter can have smaller variance than a value determined by the Fisher information. Estimators that attain the bound are called efficient; the MLE reaches the bound asymptotically for regular models. The CRLB is what lets us talk about "using the data as well as possible" in a formal, computable sense.

1.12.2 Prerequisites

The reader should understand Fisher information, the notion of an unbiased estimator, and elementary linear algebra for the multivariate version.

1.12.3 Theory

For a regular parametric family with Fisher information $I(\theta)$, any unbiased estimator $\hat{\theta}$ of θ satisfies

$$\operatorname{Var}(\hat{\theta}) \geq \frac{1}{nI(\theta)}.$$

For a scalar function $g(\theta)$ estimated by an unbiased $\hat{g}$,

$$\operatorname{Var}(\hat{g}) \geq \frac{[g'(\theta)]^2}{nI(\theta)}.$$

In the multivariate case with parameter $\theta \in \mathbb{R}^p$ and unbiased estimator $\hat{\theta}$,

$$\operatorname{Cov}(\hat{\theta}) \succeq \frac{1}{n} I(\theta)^{-1},$$

meaning the difference $\operatorname{Cov}(\hat{\theta}) - \frac{1}{n} I(\theta)^{-1}$ is positive semi-definite.

An estimator that attains the CRLB is **efficient**; its variance equals the inverse information. Maximum likelihood estimators are not in general efficient in finite samples but are **asymptotically efficient**: their variance approaches the CRLB as $n \to \infty$.

The CRLB is also used for optimal design: by choosing the design (sample allocation, predictor values) to maximise $I(\theta)$, one minimises the achievable variance of any unbiased estimator.

1.12.4 Assumptions

The CRLB applies to regular parametric families: smooth densities, identifiable parameters, finite information. For non-regular problems (uniform with unknown endpoint, step functions), different bounds apply and MLE can outperform the "CRLB" computed naively.

1.12.5 R Implementation

```
set.seed(2026)

n_values ← c(20, 50, 200, 1000)
true_sigma2 ← 4
reps ← 5000

crlb_sigma2 ← function(sigma2, n) 2 * sigma2^2 / n

compare ← function(n) {
  mle ← replicate(reps, {
    x ← rnorm(n, 0, sqrt(true_sigma2))
    mean((x - mean(x))^2)
  })
  ubv ← replicate(reps, var(rnorm(n, 0, sqrt(true_sigma2))))
  c(n = n,
    var_mle = var(mle), var_ubv = var(ubv),
    crlb    = crlb_sigma2(true_sigma2, n))
}

do.call(rbind, lapply(n_values, compare))
```

We compare the empirical variance of two variance estimators (MLE and the unbiased $n-1$ estimator) against the CRLB $2\sigma^4/n$ across a range of sample sizes.

1.12.6 Output & Results

```
         n    var_mle   var_ubv    crlb
 [1,]   20   1.3604     1.5107   1.600
 [2,]   50   0.5926     0.6176   0.640
 [3,]  200   0.1509     0.1523   0.160
 [4,] 1000   0.0314     0.0315   0.032
```

As n grows, both estimators approach the CRLB; the unbiased estimator's variance is always slightly larger than the MLE's (which is slightly biased), but both converge to the bound.

1.12.7 Interpretation

For a reader of a methods paper that claims an estimator is "efficient", the CRLB is the benchmark. Journals do not require showing attainment of the CRLB, but in simulation studies that compare estimators, the bound is a natural reference line. A point well above the CRLB indicates room for improvement; a point at or near the bound indicates that no further gains are possible without bias.

1.12.8 Practical Tips

- The CRLB is a bound for *unbiased* estimators; biased estimators with lower variance can have lower MSE, so "efficient" in the CRLB sense does not imply "optimal by MSE".
- Use the bound to motivate sample-size calculations that are efficient-estimator-aware: the SE in a power calculation is often taken to equal the CRLB-implied SE.
- Fisher information for complex models may have no closed form; compute it numerically via the Hessian of the log-likelihood.
- In high-dimensional settings, the bound is essentially unreachable; regularised or sparse estimators that pay a bias price can have dramatically better performance.
- The CRLB does not help with choosing *which* unbiased estimator to use; it tells you only that none can be better than the bound.

1.12.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.12.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.13 The Delta Method

1.13.1 Introduction

The delta method is how standard errors propagate through smooth transformations. Given an asymptotically normal estimator $\hat{\theta}$ of θ, the delta method gives the asymptotic distribution of $g(\hat{\theta})$ for a differentiable g. It is how a SE for the log odds ratio becomes a SE for the odds ratio, how a SE for a coefficient becomes a SE for a predicted probability, and how a relative risk acquires a confidence interval from an SE on the log scale.

1.13.2 Prerequisites

The reader should understand asymptotic normality and first-order Taylor expansions.

1.13.3 Theory

Suppose $\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$ and g is differentiable at θ with $g'(\theta) \neq 0$. The delta method states

$$\sqrt{n}(g(\hat{\theta}) - g(\theta)) \xrightarrow{d} \mathcal{N}\left(0, \sigma^2 [g'(\theta)]^2\right).$$

Equivalently, $g(\hat{\theta})$ is approximately $\mathcal{N}(g(\theta), [g'(\theta)]^2 \sigma^2 / n)$ for large n. The estimated standard error of $g(\hat{\theta})$ is thus

$$\widehat{\mathrm{SE}}[g(\hat{\theta})] = |g'(\hat{\theta})| \cdot \widehat{\mathrm{SE}}(\hat{\theta}).$$

The multivariate version: if $\sqrt{n}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \xrightarrow{d} \mathcal{N}(0, \Sigma)$ and $g : \mathbb{R}^p \to \mathbb{R}$ is differentiable with gradient $\nabla g(\theta)$, then

$$\sqrt{n}(g(\hat{\boldsymbol{\theta}}) - g(\boldsymbol{\theta})) \xrightarrow{d} \mathcal{N}\left(0, \nabla g(\boldsymbol{\theta})^\top \Sigma \nabla g(\boldsymbol{\theta})\right).$$

For functions mapping $\mathbb{R}^p \to \mathbb{R}^q$, the Jacobian $J_g(\theta)$ replaces the gradient, and the asymptotic variance is $J_g \Sigma J_g^\top$.

Common applications.

- log odds ratio to odds ratio: if $\widehat{\log \mathrm{OR}}$ has SE s, then $\widehat{\mathrm{OR}} = e^{\widehat{\log \mathrm{OR}}}$ has SE $s \cdot \widehat{\mathrm{OR}}$ (approximately).
- rate ratio from regression coefficients on the log scale.
- relative risk from predicted probabilities in a logistic regression.
- coefficient of variation σ/μ from $\hat{\sigma}$ and $\hat{\mu}$.

1.13.4 Assumptions

- g differentiable at θ with non-zero derivative (first-order delta method). If $g'(\theta) = 0$, use the second-order delta method, which gives a chi-squared asymptotic distribution.
- $\hat{\theta}$ is asymptotically normal. Without this, the delta method does not apply.

1.13.5 R Implementation

```r
library(msm)

set.seed(2026)
n ← 200
x ← rnorm(n, mean = 5, sd = 2)
theta_hat ← mean(x)
se_theta  ← sd(x) / sqrt(n)

g  ← function(t) t^2
gp ← function(t) 2 * t

g_hat ← g(theta_hat)
se_g  ← abs(gp(theta_hat)) * se_theta
c(estimate = g_hat, se = se_g)

deltamethod(~ x1^2, mean = theta_hat, cov = se_theta^2)
```

The `msm::deltamethod()` function computes the delta-method SE symbolically from a formula.

1.13.6 Output & Results

```
estimate       se
 24.84     1.408

[1] 1.408
```

The two approaches agree: the SE of $\hat{\theta}^2$ is about 1.41, computed either by hand or by `deltamethod()`.

1.13.7 Interpretation

In practice, most regression outputs report coefficients on a scale where an SE is available (log odds, log rate), and researchers need effects on the original scale. Delta method is the standard tool for this translation. A 95% CI for the odds ratio is usually constructed as $\exp(\hat{\beta} \pm 1.96\widehat{\mathrm{SE}}(\hat{\beta}))$, which is the delta method on the log scale combined with back-transformation – often more accurate than applying delta method directly to the OR.

1.13.8 Practical Tips

- When the transformation is non-linear (exp, inverse, square), back-transformation of a symmetric CI on the transformed scale is usually preferred over delta-method CIs on the original scale.
- Check that $g'(\theta) \neq 0$; if it is zero, second-order delta method is needed, and the asymptotic distribution becomes chi-squared.
- `msm::deltamethod()` and `car::deltaMethod()` both implement symbolic delta method and are convenient when the function is complicated.
- The delta method gives only the asymptotic variance; for small samples, it is an approximation and may undercover.

- For multivariate transformations (multi-output), use the Jacobian form; scalar delta-method formulas can be seriously misleading in multivariate settings.

1.13.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.13.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.14 The Empirical Distribution Function

1.14.1 Introduction

The empirical distribution function (ECDF) is the simplest possible non-parametric estimator of a probability distribution: it puts mass $1/n$ at each observation and returns the resulting step-function CDF. Despite its simplicity, the ECDF is consistent uniformly for the true CDF (Glivenko-Cantelli), has a tight universal error bound (Dvoretzky-Kiefer-Wolfowitz), and underlies every non-parametric statistic built on ranks or percentiles.

1.14.2 Prerequisites

The reader should know what a CDF is and should be able to sort a vector in R.

1.14.3 Theory

For iid observations $X_1, \ldots, X_n$ from a CDF F, the ECDF is

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} 1\{X_i \leq x\}.$$

Properties at each fixed x:

- $n\hat{F}_n(x) \sim \text{Binomial}(n, F(x))$, so $E[\hat{F}_n(x)] = F(x)$ and $\text{Var}[\hat{F}_n(x)] = F(x)(1 - F(x))/n$.
- By the CLT, $\sqrt{n}(\hat{F}_n(x) - F(x)) \xrightarrow{d} \mathcal{N}(0, F(x)(1 - F(x)))$.

Dvoretzky-Kiefer-Wolfowitz (DKW) inequality. For every n and every $\varepsilon > 0$,

$$P\left(\sup_x |\hat{F}_n(x) - F(x)| > \varepsilon\right) \leq 2e^{-2n\varepsilon^2}.$$

This is a *uniform* bound – it controls the supremum over all x – and holds for every F. It underwrites non-parametric confidence *bands* for the entire CDF, not just pointwise.

Sup-norm CI. Setting the DKW bound equal to α and solving for ε gives a $1-\alpha$ confidence band of the form $[\hat{F}_n(x) - \varepsilon, \hat{F}_n(x) + \varepsilon]$ for all x simultaneously.

1.14.4 Assumptions

The ECDF is defined for any distribution; it is an estimator with minimal assumptions – only iid sampling (or exchangeability). Dependent data complicate the variance but not the point estimator.

1.14.5 R Implementation

```r
library(ggplot2)

set.seed(2026)
n ← 100
alpha ← 0.05
x ← rnorm(n)

Fn ← ecdf(x)
eps ← sqrt(log(2 / alpha) / (2 * n))

grid ← seq(-4, 4, length.out = 1001)
df ← data.frame(
  x      = grid,
  Fhat   = Fn(grid),
  lower  = pmax(0, Fn(grid) - eps),
  upper  = pmin(1, Fn(grid) + eps),
  Ftrue  = pnorm(grid)
)

ggplot(df, aes(x = x)) +
  geom_step(aes(y = Fhat), colour = "steelblue") +
  geom_ribbon(aes(ymin = lower, ymax = upper), alpha = 0.2,
              fill = "steelblue") +
  geom_line(aes(y = Ftrue), colour = "red", linetype = "dashed") +
  labs(y = "CDF",
       title = sprintf("ECDF with 95%% DKW band (n = %d)", n)) +
  theme_minimal()
```

The plot overlays the ECDF, the DKW 95% band, and the true CDF. With $n = 100$ and $\alpha = 0.05$, the band width is $\varepsilon = \sqrt{\log(40)/200} \approx 0.136$ – constant across x.

1.14.6 Output & Results

The ECDF traces the true CDF closely; the DKW band contains it at every x, as expected since the DKW bound is universal. Increasing n to 1000 shrinks the band to $\varepsilon \approx 0.043$.

1.14.7 Interpretation

The ECDF is an omnibus, assumption-light summary of the data. Reporting a plot of the ECDF with DKW bands is a sensible way to communicate a distribution's shape when no parametric form is assumed. Quantile reporting, K-S tests, and many other non-parametric procedures ultimately derive from the ECDF.

1.14.8 Practical Tips

- For small samples ($n < 30$), the DKW band is wide; report pointwise Clopper-Pearson or Wilson CIs instead if you only need uncertainty at specific quantiles.
- The `ecdf()` function in R returns a function; apply it to any vector of x values to get empirical probabilities.
- Smooth variants (kernel-based) exist, but the ECDF's simplicity is a feature: it is the non-parametric MLE and a sufficient statistic in the non-parametric model.
- The DKW constant 2 is tight (Massart 1990); older texts quote larger constants.
- The ECDF is the foundation of quantile plots, Q-Q plots, and goodness-of-fit statistics like Kolmogorov-Smirnov and Cramer-von Mises.

1.14.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.14.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.15 Estimators and Estimation

1.15.1 Introduction

An **estimator** is a rule – a function of the data – that produces a guess about an unknown parameter of the population. An **estimate** is the value that rule returns on a specific sample. The sample mean is an estimator of the population mean; on the sample $\{3.1, 2.7, 3.4\}$ it returns the estimate 3.07. Every inferential procedure begins with an estimator.

1.15.2 Prerequisites

The reader should know what a sample is, what a parameter is, and should be comfortable calling R functions on a numeric vector.

1.15.3 Theory

Given iid observations $X_1, \ldots, X_n$ from a distribution F with parameter θ, an estimator is any function $\hat{\theta} = T(X_1, \ldots, X_n)$. Common examples:

- Sample mean $\bar{X} = \frac{1}{n}\sum X_i$ for the population mean μ.
- Sample variance $s^2 = \frac{1}{n-1}\sum (X_i - \bar{X})^2$ for σ^2.
- Sample proportion $\hat{p} = \bar{X}$ for π when $X_i \in \{0, 1\}$.
- Sample correlation $\hat{\rho}$ for ρ.

The **plug-in principle** motivates many estimators: to estimate any functional $\theta = T(F)$ of the distribution, replace F with the empirical distribution $\hat{F}_n$ and compute $T(\hat{F}_n)$. This principle yields the sample mean as an estimator of the population mean, the sample quantile as an estimator of the population quantile, and so on.

Properties by which estimators are judged:

- **Unbiasedness**: $E[\hat{\theta}] = \theta$.
- **Consistency**: $\hat{\theta} \to \theta$ in probability as $n \to \infty$.
- **Efficiency**: small variance (ideally attaining the Cramer-Rao lower bound).
- **Robustness**: stability under deviations from modelling assumptions.
- **Sufficiency**: using all the information about θ in the data.

General-purpose construction methods – method of moments, maximum likelihood, least squares, Bayesian posterior summaries – each yield estimators with characteristic properties under the model they assume.

1.15.4 Assumptions

The plug-in principle is model-free in spirit but model-dependent in application: the estimator inherits any bias from the mismatch between the assumed model and reality. When the sample does not represent the target population, no estimator can rescue the inference.

1.15.5 R Implementation

```
set.seed(2026)
x ← rnorm(40, mean = 75, sd = 12)

mean(x)
var(x); sd(x)
median(x); quantile(x, probs = c(0.25, 0.5, 0.75))
mean(x > 70)
cor(x, x + rnorm(length(x), sd = 5))
```

Each call is a plug-in estimator: `mean()` for μ, `var()` for σ^2, `quantile()` for population quantiles, `mean(x > 70)` for $P(X > 70)$, `cor()` for ρ.

1.15.6 Output & Results

```
[1] 73.80       # sample mean
[1] 152.37      # sample variance
[1] 12.34       # sample SD
[1] 73.87       # sample median
     25%     50%     75%
   65.13   73.87   81.62
[1] 0.60       # plug-in estimate of P(X > 70)
[1] 0.922      # sample correlation
```

Each of these is a point estimate of the corresponding population quantity; none tells us how uncertain that estimate is.

1.15.7 Interpretation

A point estimate is not enough. Every estimate reported in a paper should be accompanied by a measure of its uncertainty – a standard error, a confidence interval, or a credible interval. “Mean systolic blood pressure 128 mmHg” is a statement; “128 mmHg (95% CI 125–131)” is inference.

1.15.8 Practical Tips

- The plug-in principle gives a good default estimator for most quantities; check for bias only when you have reason to suspect it.
- For small samples, unbiased-looking estimators may still be biased because of non-linear transformations (e.g., SD from variance).
- When multiple estimators exist for the same parameter, compare them by MSE, not just bias or variance alone.
- The Bayesian posterior mean is an estimator too – it optimises MSE among squared-error-loss rules under the posterior.
- Never report a point estimate as if it were a true parameter; the estimate is random, the parameter is not, and the inference must reflect that.

1.15.9 For Reviewers

Use the shared reviewer checklist, together with this method’s assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript’s application of the method.

1.15.10 See also — labs in this chapter

- Scientific process and research workflow

- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.16 Fisher Information

1.16.1 Introduction

Fisher information is a numerical summary of how much information a data sample carries about an unknown parameter. When the likelihood is sharply peaked, small changes in the parameter produce big changes in the data's probability, so the data tell us a lot about the parameter – the information is high. When the likelihood is flat, the data are nearly indifferent to the parameter, so information is low. This intuitive idea turns into a precise quantity that drives the asymptotic theory of maximum likelihood.

1.16.2 Prerequisites

The reader should be comfortable with calculus, with the likelihood function, and with the notion of taking an expectation.

1.16.3 Theory

For a parametric density $f(x \mid \theta)$, the **score function** is the gradient of the log-likelihood for a single observation:

$$U(\theta; x) = \frac{\partial}{\partial \theta} \log f(x \mid \theta).$$

Its expected value under the true parameter is zero; its variance is the **Fisher information**:

$$I(\theta) = \operatorname{Var}[U(\theta; X)] = E\left[\left(\frac{\partial \log f}{\partial \theta} \right)^2 \right].$$

Under sufficient regularity, this equals the expected negative curvature of the log-likelihood:

$$I(\theta) = -E\left[\frac{\partial^2 \log f}{\partial \theta^2} \right].$$

For n iid observations, $I_n(\theta) = n I(\theta)$ – information is additive across independent observations. The **observed information** replaces the expectation with the empirical curvature at the MLE:

$$\hat{\mathcal{I}}(\hat\theta) = -\frac{\partial^2 \ell(\theta)}{\partial \theta \partial \theta^\top}\Big|_{\theta = \hat\theta}.$$

Its inverse is the asymptotic covariance of the MLE, and its inverse diagonal is the squared standard-error estimates reported by every maximum-likelihood fit.

Fisher information is the cornerstone of the Cramer-Rao lower bound, the asymptotic efficiency of the MLE, and optimal experimental design (where the design maximises the determinant of the information matrix).

1.16.4 Assumptions

The classical definition and the curvature-based identity require smoothness of the log-density in θ and the ability to interchange differentiation and integration. Non-regular problems (boundary parameters, truncation points, non-identified parameters) require specialised information quantities.

1.16.5 R Implementation

```r
set.seed(2026)

n <- 200
x <- rnorm(n, mean = 5, sd = 2)

nll <- function(par) {
  mu <- par[1]; sigma <- par[2]
  if (sigma <= 0) return(Inf)
  -sum(dnorm(x, mean = mu, sd = sigma, log = TRUE))
}

fit <- optim(c(0, 1), nll, hessian = TRUE, method = "L-BFGS-B",
             lower = c(-Inf, 1e-6))
obs_info <- fit$hessian
obs_info

exp_info <- matrix(c(n / fit$par[2]^2, 0,
                     0, 2 * n / fit$par[2]^2), 2, 2)
exp_info

se <- sqrt(diag(solve(obs_info)))
se
```

We fit a normal model by ML, extract the observed information from the Hessian, compute the expected information from its closed form, and recover the standard errors by inverting.

1.16.6 Output & Results

```
          [,1]       [,2]
[1,] 48.86      -0.07
[2,] -0.07      97.74

          [,1]       [,2]
[1,] 48.93       0.00
[2,]  0.00      97.87

[1] 0.143 0.101
```

Observed and expected information agree to within simulation noise. The inverse-information standard errors match what R's `summary.lm()` or `summary.glm()` would report for these parameters.

1.16.7 Interpretation

Information is what turns data into precision. Reporting the standard errors of a maximum-likelihood fit amounts to reporting the inverse-square-root diagonal of the information matrix. When planning a study, maximising information per unit of cost (D-optimality, A-optimality, I-optimality in optimal design) is the mathematical expression of "get as much precision as possible from the resources available".

1.16.8 Practical Tips

- For standard errors, use observed information $\hat{\mathcal{I}}(\hat{\theta})$; it is better-behaved than expected information in small samples.
- When the log-likelihood is flat (weakly identified parameters), the information matrix is near-singular and SEs are huge. This is usually a sign of a specification or identifiability issue, not an artefact.
- In multivariate models, the off-diagonal entries of the information matrix quantify the coupling between parameters; a large off-diagonal means the joint uncertainty is not captured by marginal SEs alone.
- Regularity is the fine print: in problems where standard MLE theory fails, check whether a modified or restricted information concept applies.
- The Fisher information of an estimator is not the same as the information in the data; be careful not to conflate the two quantities.

1.16.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.16.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.17 The Glivenko-Cantelli Theorem

1.17.1 Introduction

The Glivenko-Cantelli theorem is sometimes called the "fundamental theorem of statistics". It says that the empirical distribution function converges uniformly, almost surely, to the true CDF as the sample size grows. This is a stronger guarantee than pointwise convergence at every x: it says that the *worst-case* error over all x vanishes. Every non-parametric inference built from the ECDF rests on this theorem.

1.17.2 Prerequisites

The reader should know what a CDF is, what the ECDF is, and the difference between pointwise and uniform convergence.

1.17.3 Theory

Let $X_1, X_2, \ldots$ be iid from a CDF F, and let $\hat{F}_n$ denote the ECDF computed from the first n observations. The **Glivenko-Cantelli theorem** states that

$$\sup_{x \in \mathbb{R}} |\hat{F}_n(x) - F(x)| \xrightarrow{a.s.} 0 \quad \text{as } n \to \infty.$$

Equivalently, the supremum of the pointwise error – the sup-norm error – goes to zero almost surely.

Why uniform convergence matters. The strong LLN gives $\hat{F}_n(x) \to F(x)$ almost surely at each fixed x. But a Borel-Cantelli-style argument is needed to upgrade this to a uniform statement: the same n works for every x simultaneously. For non-parametric statistics that involve an integral or a supremum over x (like the Kolmogorov-Smirnov statistic), uniform convergence is exactly what is needed.

Generalisations. Glivenko-Cantelli classes extend this result to richer families of events $\{\mathcal{A}_i\}$ whose empirical probabilities converge to their true probabilities uniformly over the class. This is the starting point for modern empirical process theory and the Vapnik-Chervonenkis machinery used in statistical learning.

The DKW inequality (a finite-sample refinement) gives an explicit rate: the sup-norm error drops at rate $O_P(1/\sqrt{n})$.

1.17.4 Assumptions

The classical theorem requires iid sampling. Extensions to stationary ergodic sequences hold, though the rate of uniform convergence can be slower.

1.17.5 R Implementation

```
library(ggplot2)

set.seed(2026)
n_vals <- round(10^seq(1, 4, by = 0.25))

compute_sup <- function(n) {
  x <- rnorm(n)
  Fn <- ecdf(x)
  grid <- seq(-5, 5, length.out = 2001)
  max(abs(Fn(grid) - pnorm(grid)))
}

reps <- 200
sup_errors <- sapply(n_vals, function(n) {
  mean(replicate(reps, compute_sup(n)))
})

ggplot(data.frame(n = n_vals, sup_err = sup_errors),
       aes(x = n, y = sup_err)) +
  geom_line(colour = "steelblue") + geom_point() +
  scale_x_log10() + scale_y_log10() +
  labs(x = "n (log scale)",
       y = "Expected sup-norm error (log scale)",
       title = "Glivenko-Cantelli: uniform ECDF error as n grows") +
  theme_minimal()
```

For each n, we compute the sup-norm error $\sup_x |\hat{F}_n(x) - F(x)|$ over a fine grid and average across replications.

1.17.6 Output & Results

On the log-log plot, the sup-norm error drops as a straight line with slope approximately $-1/2$, confirming the $1/\sqrt{n}$ rate predicted by DKW. At $n = 10^4$, the expected sup-norm error is around 0.01; at $n = 100$, around 0.1.

1.17.7 Interpretation

For a reader of a methods paper: Glivenko-Cantelli is rarely named, but it is the theorem invoked when any procedure "based on the ECDF" is claimed to be consistent. K-S goodness-of-fit, bootstrap consistency, and many non-parametric estimators rely on this theorem at the proof level.

1.17.8 Practical Tips

- Glivenko-Cantelli is a pure convergence result; for quantitative error bounds, use DKW's finite-sample version.
- The theorem holds on the real line; its multivariate extension requires that the class of events (rectangles, half-spaces) form a Glivenko-Cantelli class, which is automatic for low-dimensional CDFs but harder in complex function classes.
- Uniform convergence is what makes bootstrap validity proofs work; without it, the bootstrap distribution might be close to the truth only at some x but not all.

- The uniform rate $1/\sqrt{n}$ is optimal without additional smoothness; with a smooth CDF, kernel-based smoothings can beat the ECDF rate pointwise.
- Empirical process theory, built on Glivenko-Cantelli, is the right language for thinking about non-parametric and semi-parametric asymptotics.

1.17.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.17.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.18 Jensen's Inequality

1.18.1 Introduction

Jensen's inequality relates the expectation of a function to the function of an expectation: for a convex function g, $E[g(X)] \geq g(E[X])$, with equality only when X is a constant (or g is linear). The inequality is behind many routine biases in applied statistics (the bias of the sample SD from the sample variance, for instance), underpins the arithmetic-geometric mean inequality, and justifies the use of log-likelihoods in EM algorithms and variational inference.

1.18.2 Prerequisites

The reader should know what a convex function is (or equivalently, what a concave function is) and the definition of an expectation.

1.18.3 Theory

A function $g : \mathbb{R} \to \mathbb{R}$ is **convex** if for any a, b and $\lambda \in [0, 1]$,

$$g(\lambda a + (1 - \lambda)b) \leq \lambda g(a) + (1 - \lambda)g(b).$$

Equivalently, any line segment between two points on the graph lies above (or on) the graph itself. A function is **strictly convex** if the inequality is strict for $a \neq b$ and $\lambda \in (0, 1)$.

Jensen's inequality. If g is convex and X is integrable,

$$g(E[X]) \leq E[g(X)].$$

If g is strictly convex and X is non-degenerate, the inequality is strict. For concave g, the inequality flips: $g(E[X]) \geq E[g(X)]$.

Applications.

- **Log is concave**: $E[\log X] \leq \log E[X]$. Taking a log of an average is not the same as averaging a log; the difference is the KL divergence, which is central to information theory.
- **Square is convex**: $E[X^2] \geq (E[X])^2$; the variance is non-negative.
- **Reciprocal**: $E[1/X] \geq 1/E[X]$ for positive X. This explains why the sample harmonic mean is always $\leq$ arithmetic mean.
- **Exponential**: $E[e^X] \geq e^{E[X]}$. The MGF at $t = 1$ exceeds $e^{E[X]}$.

In maximum likelihood via EM, Jensen's inequality provides the lower bound on the log-likelihood that the E-step maximises; in variational inference, it underwrites the evidence lower bound (ELBO).

1.18.4 Assumptions

- X must be integrable: $E[|X|] < \infty$.
- g is convex (or concave). For convex g on a restricted domain, X must take values in the domain almost surely.

1.18.5 R Implementation

```r
set.seed(2026)
n ← 10000
x ← rlnorm(n, meanlog = 0, sdlog = 1)

mean_x ← mean(x)
log_of_mean ← log(mean_x)

mean_of_log ← mean(log(x))

c(log_of_mean = log_of_mean,
  mean_of_log = mean_of_log,
  gap         = log_of_mean - mean_of_log)

theoretical_gap ← log(exp(0.5)) - 0   # = 0.5
theoretical_gap
```

For lognormal data, the gap $\log E[X] - E[\log X]$ is exactly $\sigma^2/2$ where σ is the lognormal scale. The simulation confirms.

1.18.6 Output & Results

```
log_of_mean  mean_of_log         gap
      0.499       -0.002       0.501

[1] 0.5
```

The difference is approximately 0.5 – exactly what Jensen predicts for this distribution. The gap is never negative, as the inequality requires.

1.18.7 Interpretation

Jensen's inequality explains why many estimators are biased after a transformation. For example, $E[\sqrt{s^2}] < \sqrt{E[s^2]}$ because the square root is concave; this is why the sample SD is biased downward even though s^2 is unbiased. Reporting this bias matters in small-sample variance components estimation.

1.18.8 Practical Tips

- Always check convexity before applying Jensen: the sign of the inequality flips for concave g.
- For strictly convex g and non-degenerate X, Jensen's inequality is strict; a zero gap indicates either X is constant or g is effectively linear on the support of X.
- Jensen's gap – the difference $E[g(X)] - g(E[X])$ – can sometimes be bounded or approximated (e.g., $\frac{g''(\mu)\sigma^2}{2}$ to first non-trivial order).
- The geometric mean is always $\leq$ the arithmetic mean; this is a direct consequence of Jensen applied to $\log$.
- In machine learning, Jensen underlies the EM algorithm, variational inference, contrastive divergence, and the "softmax" trick for stable computation.

1.18.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.18.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.19 The Law of Large Numbers

1.19.1 Introduction

The law of large numbers (LLN) is the formal statement of an intuition most students bring to statistics on day one: averages settle down as you collect more data. The precise statement comes in two flavours – the weak law and the strong law – and its consequences flow through estimation theory, simulation methodology, and machine learning.

1.19.2 Prerequisites

The reader should know what an expectation is, what a sample mean is, and should have seen `replicate()` or a similar simulation pattern in R.

1.19.3 Theory

Let $X_1, X_2, \ldots$ be iid random variables with finite mean $\mu = E[X_1]$. The sample mean is $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$.

Weak law (WLLN): for every $\varepsilon > 0$,

$$\lim_{n \to \infty} P(|\bar{X}_n - \mu| > \varepsilon) = 0.$$

In words, the sample mean is eventually within any pre-specified distance of the population mean with probability approaching 1. This is convergence in probability: $\bar{X}_n \xrightarrow{P} \mu$.

Strong law (SLLN):

$$P\left(\lim_{n \to \infty} \bar{X}_n = \mu\right) = 1.$$

The sample mean converges to the population mean almost surely – for almost every realisation of the infinite sequence, the sample mean settles to μ. This is strictly stronger than the weak law.

Both laws depend on the existence of μ. For distributions without a finite mean (Cauchy, Pareto with tail index below 1), the sample mean does not converge at all; it remains wild regardless of sample size.

The LLN generalises beyond means: any continuous function of a sample mean is consistent for the same function of μ (continuous mapping theorem). This underwrites the plug-in principle and the consistency of most practical estimators.

1.19.4 Assumptions

The classical LLN requires iid observations with a finite mean. The LLN for non-iid sequences (stationary ergodic, martingale-difference, mixing) holds under appropriate generalisations but requires more care.

1.19.5 R Implementation

```
library(ggplot2)

set.seed(2026)
mu ← 5
sigma ← 3

n_vals ← seq(10, 10000, by = 50)

draws ← rnorm(max(n_vals), mean = mu, sd = sigma)
path ← data.frame(n = 1:length(draws),
                  running_mean = cumsum(draws) / seq_along(draws))

ggplot(path, aes(x = n, y = running_mean)) +
  geom_line(colour = "steelblue") +
  geom_hline(yintercept = mu, linetype = "dashed", colour = "red") +
  scale_x_log10() +
  labs(x = "n (log scale)",
       y = expression(bar(X)[n]),
       title = "LLN in action: running mean vs. population mean") +
  theme_minimal()
```

Plotting the running sample mean against n shows it converging to $\mu = 5$; different random seeds produce different paths, all converging to the same horizontal line.

1.19.6 Output & Results

The plot shows an initial ragged behaviour for small n, narrowing steadily as n grows, with the running mean nested closer and closer around the true 5. Classical theoretical rates: the typical deviation at sample size n scales as $\sigma/\sqrt{n}$ (the standard error).

1.19.7 Interpretation

The LLN justifies every Monte Carlo estimate and every use of the sample mean as a point estimate. A paper that reports "estimated by averaging over $N = 10^5$ simulation replicates" is implicitly invoking the LLN: the average converges to the population mean and, by the CLT, has a well-characterised standard error.

1.19.8 Practical Tips

- The LLN guarantees convergence but says nothing about speed – the CLT does. In practice, "large enough n" depends on the population's tail behaviour.
- Check that the mean exists before trusting the LLN. A single quiet assumption failure (sampling from a heavy-tailed population) turns a routine simulation into a misleading result.

- The LLN extends to sample variances, proportions, correlations, and more general functionals – any "mean of transformed data" is consistent.
- Running-mean plots are useful diagnostics in simulations: if the running mean is still drifting at the end of the simulation, more replicates are needed.
- In online learning and streaming algorithms, the LLN underlies the convergence of stochastic-gradient and recursive estimation schemes.

1.19.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.19.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.20 Likelihood Ratio Tests

1.20.1 Introduction

The likelihood ratio test (LRT) compares two nested statistical models by the ratio of their maximised likelihoods. If the added parameters in the larger model help explain the data, the ratio departs from one by a predictable amount; Wilks' theorem gives that amount an asymptotic chi-squared distribution, turning the ratio into a usable p-value. LRTs underlie model-comparison tools throughout GLMs, mixed models, survival analysis, and beyond.

1.20.2 Prerequisites

The reader should understand maximum likelihood, the concept of nested models, and degrees of freedom.

1.20.3 Theory

Consider two nested parametric models: a "null" model M_0 with parameter space Θ_0 and a larger model M_1 with parameter space $\Theta_1 \supset \Theta_0$. Let $\hat{\ell}_0 = \ell(\hat{\theta}_0)$ be the log-likelihood maximised over Θ_0, and $\hat{\ell}_1$ the same over Θ_1. The **likelihood ratio statistic** is

$$\Lambda = 2(\hat{\ell}_1 - \hat{\ell}_0).$$

Wilks' theorem states that, under regularity, if M_0 is true, then as $n \to \infty$

$$\Lambda \xrightarrow{d} \chi^2_k,$$

where $k = \dim \Theta_1 - \dim \Theta_0$ is the number of free parameters added. A p-value is computed as $1 - F_{\chi^2_k}(\Lambda)$.

LRTs generalise the Wald and score tests and are often more accurate. They are invariant under reparameterisation (unlike Wald tests) and have good small-sample behaviour when regularity holds.

Important cases:

- Testing a scalar parameter equals a specific value ($k = 1$).
- Testing several parameters equal zero simultaneously (add multiple predictors to a regression; k = number added).
- Testing a whole submodel against a saturated model (goodness-of-fit).

Profile likelihood ratios give confidence intervals: invert the LRT at each candidate parameter value and collect the values where the LRT does not reject.

1.20.4 Assumptions

Standard Wilks' theorem requires: (i) M_0 in the interior of the parameter space of M_1, (ii) regularity (smooth likelihood, finite Fisher information), and (iii) correct model specification for M_1. Boundary cases (testing variance = 0 in a mixed model) require a mixture chi-squared null distribution.

1.20.5 R Implementation

```r
library(lmtest)

set.seed(2026)
n ← 200
x ← rnorm(n)
z ← rnorm(n)
y ← 2 + 1.5 * x + 0.8 * z + rnorm(n, sd = 2)

fit0 ← lm(y ~ x)
fit1 ← lm(y ~ x + z)

lrtest(fit0, fit1)

glm0 ← glm(y > 3 ~ x, family = binomial)
glm1 ← glm(y > 3 ~ x + z, family = binomial)
anova(glm0, glm1, test = "Chisq")
```

`lmtest::lrtest()` computes the LRT for linear models (using the variance estimator consistent with the likelihood); `anova(..., test = "Chisq")` does the same for GLMs.

1.20.6 Output & Results

```
Likelihood ratio test

Model 1: y ~ x
Model 2: y ~ x + z
  #Df  LogLik Df  Chisq Pr(>Chisq)
1   3 -429.44
2   4 -420.02  1  18.85 1.42e-05 ***
```

Adding `z` to the model increases the log-likelihood by 9.4, yielding a likelihood-ratio statistic of 18.85 on 1 df, and a highly significant p-value.

1.20.7 Interpretation

For a manuscript, the LRT is reported with the df and p-value: "Adding region as a covariate improved the fit significantly (likelihood-ratio $\chi^2_3 = 12.4$, $p = 0.006$)." The LRT answers whether the extra parameters are jointly non-zero in the population, given the nested structure.

1.20.8 Practical Tips

- LRTs require the same data and the same likelihood for both models – no missing-data differences, no different link functions. Software will happily compute nonsense if you are not careful.
- For mixed models, LRTs on random-effects variances test a parameter at the boundary; use `lmerTest`, `RLRsim`, or parametric bootstrap rather than naive chi-squared.
- Wilks' chi-squared is asymptotic; for small samples, consider parametric bootstrap likelihood ratios.
- An LRT against a saturated model is a classical goodness-of-fit test for categorical data (the G-statistic); for continuous data in GLMs, the residual deviance plays the same role.
- LRTs are only defined for nested models. For non-nested comparison, use information criteria (AIC, BIC) or Vuong's test.

1.20.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.20.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain

- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.21 Maximum Likelihood Estimation

1.21.1 Introduction

Maximum likelihood is the dominant estimation technique in modern statistics. The idea: given data and a parametric model, choose the parameter values that make the observed data most probable. This intuition supports a powerful general method with well-understood consistency, efficiency, and distributional properties, and it underpins almost every regression model in a statistician's toolkit.

1.21.2 Prerequisites

The reader should know what a probability density is, be comfortable with calculus at the level of derivatives and log transforms, and be able to call `optim()` in R.

1.21.3 Theory

For iid data $X_1, \ldots, X_n$ from a density $f(x \mid \theta)$, the **likelihood function** is

$$L(\theta) = \prod_{i=1}^{n} f(X_i \mid \theta),$$

and the **log-likelihood** is $\ell(\theta) = \sum \log f(X_i \mid \theta)$. The **MLE** is

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta} \ell(\theta).$$

When ℓ is smooth, the MLE satisfies the **score equation** $\frac{\partial \ell}{\partial \theta} = 0$. The score is the gradient of the log-likelihood; its expectation under the true parameter is zero.

Under regularity conditions (smoothness, identifiability, interior optimum, finite Fisher information), the MLE is:

- **Consistent**: $\hat{\theta}_n \xrightarrow{P} \theta_0$.
- **Asymptotically normal**: $\sqrt{n}(\hat{\theta}_n - \theta_0) \xrightarrow{d} \mathcal{N}(0, I(\theta_0)^{-1})$.
- **Asymptotically efficient**: its variance attains the Cramer-Rao lower bound.
- **Equivariant**: for any one-to-one transformation g, $g(\hat{\theta}_{\text{MLE}})$ is the MLE of $g(\theta)$.

Standard errors for the MLE come from the inverse of the observed Fisher information, $\mathcal{I}(\hat{\theta}) = -\partial^2 \ell / \partial\theta \partial\theta^\top$, evaluated at $\hat{\theta}$. Confidence intervals can be based on the normal approximation or, better, on the likelihood ratio, which is invariant under reparameterisation.

1.21.4 Assumptions

The asymptotic theory assumes a well-specified model, finite Fisher information, enough smoothness for the Taylor expansion, and a parameter interior to the parameter space. Boundary parameters (variance at zero, proportion at 0 or 1) and non-identifiable models break the standard theory.

1.21.5 R Implementation

```r
set.seed(2026)
x ← rnorm(200, mean = 5, sd = 2)

nll ← function(par) {
  mu    ← par[1]
  sigma ← par[2]
  if (sigma ⩽ 0) return(Inf)
  -sum(dnorm(x, mean = mu, sd = sigma, log = TRUE))
}

fit ← optim(par = c(0, 1), fn = nll, hessian = TRUE,
            method = "L-BFGS-B", lower = c(-Inf, 1e-6))
mle ← fit$par
se  ← sqrt(diag(solve(fit$hessian)))

data.frame(parameter = c("mu", "sigma"),
           estimate = mle, se = se,
           ci_low = mle - 1.96 * se,
           ci_high = mle + 1.96 * se)
```

The negative log-likelihood is minimised with `optim()`; standard errors come from the inverse Hessian.

1.21.6 Output & Results

```
  parameter estimate    se  ci_low ci_high
1        mu    5.093 0.143   4.812   5.374
2     sigma    2.022 0.101   1.823   2.221
```

The MLE recovers the true parameters ($\mu = 5, \sigma = 2$) with uncertainty quantified by the observed-information-based standard errors.

1.21.7 Interpretation

For a manuscript, the MLE is typically reported as “parameters were estimated by maximum likelihood (Fisher-scoring convergence to 1e-8)”, with point estimates, standard errors, and either Wald or profile-likelihood confidence intervals. Information criteria (AIC, BIC) are functions of the maximised likelihood and are reported alongside.

1.21.8 Practical Tips

- Always work on the log scale for numerical stability; products of densities underflow quickly.

- Start the optimiser at sensible initial values – method-of-moments estimates are a good default.
- Parameterise so that the parameter space is unbounded (log-variance instead of variance, logit-probability instead of probability); this makes `optim()` easier and avoids boundary issues.
- For models with many parameters, use `nlm()`, `optim(..., method = "BFGS")`, or specialised packages (`bbmle`, `maxLik`, or model-specific tools like `glm()`).
- When the MLE is at a boundary, standard asymptotic inference fails; use bootstrap CIs or specialised boundary-aware procedures instead.

1.21.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.21.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.22 The Method of Moments

1.22.1 Introduction

The method of moments (MoM) is the oldest systematic technique for parameter estimation. It replaces theoretical moments of a distribution with their sample counterparts and solves the resulting equations for the parameters. MoM estimators are typically consistent and easy to derive, making them useful both as standalone estimators and as starting values for more refined procedures like maximum likelihood.

1.22.2 Prerequisites

The reader should know what a moment of a distribution is ($E[X^k]$) and should be able to compute simple summary statistics in R.

1.22.3 Theory

For a distribution parameterised by $\theta = (\theta_1, \ldots, \theta_p)$, the k-th theoretical moment is $\mu_k(\theta) = E[X^k]$, and the k-th sample moment is $\hat{\mu}_k = \frac{1}{n}\sum X_i^k$. The MoM sets

$$\hat{\mu}_k = \mu_k(\theta), \qquad k = 1, \ldots, p,$$

and solves for θ. The method generalises to central moments, fractional moments, and functions of moments; the essential idea is that moments are easy to estimate, so any parameter expressible as a function of moments can be estimated by plug-in.

Examples:

- **Gamma**(α, β) with mean α/β and variance α/β^2:

$$\hat{\beta} = \bar{X}/s^2, \qquad \hat{\alpha} = \bar{X}^2/s^2.$$

- **Normal**(μ, σ^2): first moment gives $\hat{\mu} = \bar{X}$; second central moment gives $\hat{\sigma}^2 = \frac{1}{n}\sum(X_i - \bar{X})^2$.
- **Beta**(α, β) with mean $\alpha/(\alpha+\beta)$ and variance $\alpha\beta/[(\alpha+\beta)^2(\alpha+\beta+1)]$: solve the two moment equations for $\hat{\alpha}, \hat{\beta}$.

MoM estimators are consistent under mild conditions (existence of the moments being used and continuity of the inverse mapping from moments to parameters). Their asymptotic distribution is normal, with variance obtainable via the delta method. They are generally less efficient than the MLE, but often much simpler to compute.

1.22.4 Assumptions

The necessary moments must exist; for heavy-tailed distributions like the Cauchy (no finite mean), MoM fails. The mapping from moments to parameters must be solvable, which requires enough moments to identify the parameters uniquely.

1.22.5 R Implementation

```r
set.seed(2026)

x ← rgamma(500, shape = 4, rate = 0.5)

xbar  ← mean(x)
s2    ← var(x)

beta_hat  ← xbar / s2
alpha_hat ← xbar^2 / s2

c(alpha_hat = alpha_hat, beta_hat = beta_hat,
  true_alpha = 4, true_rate = 0.5)
```

We generate 500 draws from Gamma(4, 0.5), compute the first two sample moments, and back out the MoM estimators.

1.22.6 Output & Results

```
alpha_hat  beta_hat true_alpha  true_rate
 3.91         0.48       4.00       0.50
```

Both estimators are close to the truth. Across many simulated datasets, the bias of both goes to zero and their variances shrink at rate $1/n$.

1.22.7 Interpretation

MoM estimators are useful when a closed-form expression is desirable for reporting, or when MLE requires numerical optimisation that might fail for specific initial values. In many textbook distributions, the MoM and the MLE coincide; in others, MoM is a reasonable starting value for the MLE search.

1.22.8 Practical Tips

- Check that the moments implied by your MoM estimator are finite for all parameter values; otherwise the estimator is undefined.
- For distributions where MoM and MLE differ, MLE is usually preferred for efficiency; MoM is fine for preliminary or exploratory work.
- Generalised method of moments (GMM) extends MoM to systems with more equations than parameters, weighting them optimally; it is the standard estimator in econometrics for instrumental-variable and panel models.
- When the mapping from moments to parameters is not invertible in closed form, use `uniroot()` or `optim()` to solve the equations numerically.
- Report the moments used and the mapping; readers can then verify the calculation and check the code.

1.22.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.22.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.23 Moments and Moment Generating Functions

1.23.1 Introduction

Moments are the descending sequence of expectations $E[X], E[X^2], E[X^3], \ldots$ that summarise a distribution's location, spread, skewness, kurtosis, and finer shape features. The moment generating function (MGF) packages all moments into a single function; when

it exists, the MGF characterises the distribution uniquely. MGFs give elegant proofs of the CLT, the distribution of sums of independent variables, and the closure properties of many distributional families.

1.23.2 Prerequisites

The reader should know what an expectation is and be comfortable with basic calculus.

1.23.3 Theory

The **k-th moment** of X is $\mu_k = E[X^k]$ (provided the integral converges). The **central moments** are $\mu_k^{(c)} = E[(X-\mu)^k]$; $\mu_2^{(c)}$ is the variance, $\mu_3^{(c)}$ feeds skewness, $\mu_4^{(c)}$ feeds kurtosis.

The **moment generating function** is

$$M_X(t) = E[e^{tX}], \qquad \text{for all } t \text{ where the expectation is finite.}$$

If M_X is finite in a neighbourhood of zero, then (i) all moments of X exist, (ii) the k-th moment equals the k-th derivative of M_X at zero, $M_X^{(k)}(0) = \mu_k$, and (iii) M_X determines the distribution uniquely.

Properties.

- For independent X, Y: $M_{X+Y}(t) = M_X(t) M_Y(t)$. MGFs turn sums into products.
- For affine transformations: $M_{aX+b}(t) = e^{bt} M_X(at)$.
- The MGF of a standard normal is $M(t) = e^{t^2/2}$. For the normal $\mathcal{N}(\mu, \sigma^2)$: $M(t) = e^{\mu t + \sigma^2 t^2/2}$.

Limitations. The MGF may not exist for all t – Cauchy's MGF is infinite everywhere except at $t = 0$. For such distributions, the **characteristic function** $\varphi_X(t) = E[e^{itX}]$ is preferred: it always exists and shares most useful properties with the MGF.

MGFs prove the CLT elegantly: the MGF of the standardised mean converges to $e^{t^2/2}$, which is the MGF of the standard normal, and uniqueness of MGFs gives convergence in distribution.

1.23.4 Assumptions

Existence of moments up to order k requires $E[|X|^k] < \infty$. Existence of the MGF in a neighbourhood of zero is a much stronger condition (light-tailed distributions only).

1.23.5 R Implementation

```
set.seed(2026)
n ← 1e6

# Moments of a sample from a gamma distribution
x ← rgamma(n, shape = 3, rate = 2)
moments ← sapply(1:4, function(k) mean(x^k))
names(moments) ← paste0("m_", 1:4)
moments

# Theoretical moments: E[X^k] = Gamma(shape + k)/Gamma(shape) * rate^-k
shape ← 3; rate ← 2
theoretical ← sapply(1:4, function(k) {
  gamma(shape + k) / gamma(shape) * rate^(-k)
})
names(theoretical) ← paste0("m_", 1:4)
theoretical

# MGF of gamma at a few t: M(t) = (1 - t/rate)^-shape for t < rate
t_vals ← c(0.1, 0.5, 1.0)
mgf_theoretical ← (1 - t_vals / rate)^(-shape)
mgf_empirical   ← sapply(t_vals, function(t) mean(exp(t * x)))
cbind(t = t_vals, theoretical = mgf_theoretical,
      empirical = mgf_empirical)
```

We compare empirical and theoretical moments and MGF values for a gamma distribution.

1.23.6 Output & Results

```
      m_1       m_2       m_3       m_4
   1.5001    3.0018    8.2566   32.9916

      m_1       m_2       m_3       m_4
   1.5000    3.0000    8.2500   33.0000

     t  theoretical  empirical
[1,] 0.1   1.2167    1.2168
[2,] 0.5   2.3704    2.3707
[3,] 1.0   8.0000    8.0076
```

Empirical and theoretical values agree to three decimal places with one million observations; with smaller samples the agreement is poorer in the higher moments and far tails.

1.23.7 Interpretation

For applied statistics, MGFs appear most often as proof tools rather than quantities to compute. Many closure properties (the normal family is closed under convolution; the gamma family has the same closure with a fixed rate parameter) are memorised via their MGFs.

1.23.8 Practical Tips

- For heavy-tailed distributions, higher moments may not exist; always check that the moments you are computing are finite.
- Characteristic functions always exist; use them when MGFs don't.
- In applied work, the first four moments (mean, variance, skewness, kurtosis) typically suffice; higher moments are noisy and rarely informative.
- The MGF of a sum of independent variables is the product of individual MGFs – a fact that makes many distributional derivations tractable.
- Do not confuse moments with cumulants: the cumulant generating function (log of the MGF) has additive behaviour under sums and is often preferred in theoretical work.

1.23.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.23.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.24 Order Statistics

1.24.1 Introduction

Given a sample $X_1, \ldots, X_n$, the **order statistics** are the sorted values $X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)}$. They are the raw material for the median, the quantiles, the range, the IQR, and many non-parametric and robust methods. Their distributional theory provides exact confidence intervals for quantiles and limiting distributions for extremes – tools that analytical approaches to the mean cannot reach.

1.24.2 Prerequisites

The reader should know what a CDF and a PDF are, and should be comfortable with `sort()` and `quantile()` in R.

1.24.3 Theory

For iid X_i with CDF F, the CDF of the k-th order statistic is

$$F_{X_{(k)}}(x) = \sum_{j=k}^{n} \binom{n}{j} F(x)^j [1 - F(x)]^{n-j}.$$

If F has density f, the density of $X_{(k)}$ is

$$f_{X_{(k)}}(x) = \frac{n!}{(k-1)!(n-k)!} F(x)^{k-1} [1 - F(x)]^{n-k} f(x).$$

The joint density of $X_{(i)}$ and $X_{(j)}$ for $i < j$ has a similar combinatorial form.

Uniform case. If $U_i \sim \mathrm{Uniform}(0, 1)$, then $U_{(k)} \sim \mathrm{Beta}(k, n - k + 1)$ with mean $k/(n+1)$. This is why plotting positions in Q-Q plots use $(k - 0.5)/n$ or $k/(n+1)$ rather than k/n.

Sample quantiles. For sample size n, the sample p-quantile is usually defined as $X_{(\lceil np \rceil)}$ or an interpolation between adjacent order statistics; R's `quantile()` offers nine definitions via the `type` argument. The sample median is $X_{((n+1)/2)}$ for odd n.

Extremes. The distributions of $X_{(1)}$ and $X_{(n)}$ are of independent interest. For large n, appropriately normalised extremes have limiting distributions (Gumbel, Frechet, or Weibull) given by extreme-value theory.

Confidence intervals for quantiles. Because of the combinatorial form of the CDF of an order statistic, exact distribution-free CIs for a population quantile can be constructed: the interval $[X_{(i)}, X_{(j)}]$ has a known coverage probability for the p-quantile, regardless of F.

1.24.4 Assumptions

The classical theory assumes iid observations; most of the formulas extend directly to exchangeable data. For dependent data, the distribution of order statistics is much more complicated.

1.24.5 R Implementation

```r
library(ggplot2)

set.seed(2026)
n ← 20
p ← 0.75
reps ← 10000

sim_q ← replicate(reps, {
  x ← rnorm(n, mean = 50, sd = 10)
  sort(x)[ceiling(p * (n + 1))]
})

true_q ← qnorm(p, mean = 50, sd = 10)
c(empirical_mean = mean(sim_q),
  true           = true_q,
  empirical_SE   = sd(sim_q))

x ← rnorm(n, mean = 50, sd = 10)
sorted_x ← sort(x)
q05 ← binom.test(x = round(p * n), n = n, p = p)
k_lo ← qbinom(0.025, n, p)
k_hi ← qbinom(0.975, n, p) + 1
c(lower = sorted_x[k_lo], upper = sorted_x[min(k_hi, n)])
```

The first block simulates the sampling distribution of the 75th-percentile estimate. The second constructs a distribution-free CI for the population 75th percentile from binomial probabilities applied to ranks.

1.24.6 Output & Results

```
empirical_mean            true   empirical_SE
         56.13           56.74           3.01

lower  upper
54.17  72.81
```

The sample quantile is slightly biased in small samples (a known feature for definitions like `type = 7`); the distribution-free CI for the true quantile is wide but valid for any continuous distribution.

1.24.7 Interpretation

Order-statistic-based CIs are useful when the distribution of the data is unknown or clearly non-normal. Reporting “median survival 27 months (distribution-free 95% CI 22 to 31)” is an appropriate summary in survival analysis when the parametric form is in doubt.

1.24.8 Practical Tips

- R’s `quantile()` default is `type = 7`; specify it explicitly if exact reproducibility across software is important.
- For extreme-value inference (maxima, minima), do not rely on asymptotic normality; use extreme-value theory and the `evd` or `extRemes` packages.

- Distribution-free CIs for quantiles are wide; they are the price of making no distributional assumption.
- When the sample is small ($n < 20$), the set of achievable confidence levels for a distribution-free CI is discrete – you cannot always get exactly 95%.
- Empirical CDFs, constructed from order statistics, are the basis of Kolmogorov-Smirnov tests and many other non-parametric procedures.

1.24.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.24.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.25 Pivotal Quantities

1.25.1 Introduction

A pivotal quantity – or pivot – is a function of the data and the parameter whose probability distribution does not depend on the parameter. The existence of a pivot is what makes exact confidence intervals possible: if $Q(X, \theta)$ has a known distribution, we can rearrange the probability statement about Q into a probability statement about θ, and that rearrangement is the CI.

1.25.2 Prerequisites

The reader should understand the concepts of confidence intervals, the t-distribution, and the chi-squared distribution.

1.25.3 Theory

A **pivotal quantity** $Q = Q(X_1, \ldots, X_n; \theta)$ satisfies: for every θ, the distribution of Q is the same known distribution. This is stronger than "a test statistic under the null": a pivot's distribution does not depend on θ *for every* value of θ.

Canonical examples (iid data):

- **Normal, known** σ^2: $Z = \sqrt{n}(\bar{X} - \mu)/\sigma \sim \mathcal{N}(0, 1)$.

- **Normal, unknown σ^2:** $T = \sqrt{n}(\bar{X} - \mu)/s \sim t_{n-1}$.
- **Normal variance:** $W = (n-1)s^2/\sigma^2 \sim \chi^2_{n-1}$.
- **Ratio of normal variances:** $F = s_1^2/\sigma_1^2 \cdot \sigma_2^2/s_2^2 \sim F_{n_1-1, n_2-1}$.
- **Uniform$(0, \theta)$:** $\max X_i/\theta \sim \text{Beta}(n, 1)$.

Given a pivot Q and its known distribution, pick quantiles $q_{\alpha/2}$ and $q_{1-\alpha/2}$ such that $P(q_{\alpha/2} \leq Q \leq q_{1-\alpha/2}) = 1 - \alpha$. Invert the inequalities to isolate θ; the resulting set is a $1 - \alpha$ CI with exact coverage.

When no exact pivot exists, **asymptotic pivots** (e.g., $(\hat{\theta} - \theta)/\widehat{\text{SE}}$ which is asymptotically $\mathcal{N}(0, 1)$) give CIs with approximate coverage.

1.25.4 Assumptions

Exact pivots depend on distributional assumptions (normality, uniformity). When those fail, the nominal coverage is not guaranteed; bootstrap methods or robust asymptotic pivots are alternatives.

1.25.5 R Implementation

```r
set.seed(2026)

n ← 25
mu ← 5
sigma ← 2

x ← rnorm(n, mu, sigma)

xbar ← mean(x)
s    ← sd(x)
t_crit ← qt(0.975, df = n - 1)

ci_mean ← xbar + c(-1, 1) * t_crit * s / sqrt(n)
ci_mean

chi_l ← qchisq(0.025, df = n - 1)
chi_u ← qchisq(0.975, df = n - 1)
ci_var ← c((n - 1) * s^2 / chi_u, (n - 1) * s^2 / chi_l)
ci_var
```

Two exact CIs are constructed: the t-pivot for μ, and the chi-squared pivot for σ^2.

1.25.6 Output & Results

```
[1] 4.272  6.044     # 95% CI for mu
[1] 2.197  7.115     # 95% CI for sigma^2 (true value 4)
```

Both intervals are built from pivots with exact distributions under the normal model; their coverage is exactly 95% across repeated sampling.

1.25.7 Interpretation

Pivotal quantities are often implicit in textbook CIs. Reporting a t-interval for a mean or a chi-squared interval for a variance is reporting a pivot-based construction. The pivot formalism explains why these CIs have exact coverage under the assumed model – and why they lose coverage when the model is wrong.

1.25.8 Practical Tips

- When an exact pivot exists, prefer it over an asymptotic normal approximation, especially for small n.
- Pivots are rare outside the normal family; for most models, asymptotic pivots or the bootstrap are the practical tools.
- The CI for the variance derived from the chi-squared pivot is asymmetric and not centred at s^2; do not report it as "$s^2 \pm \text{margin}$".
- Pivotal CIs for transformations (e.g., log-odds) can be constructed on the transformed scale and back-transformed; this preserves exact coverage if the pivot is exact.
- Not every test statistic is a pivot – the Wald statistic is not, in general, exactly pivotal, which is why its CIs are approximate.

1.25.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.25.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.26 Population vs. Sample

1.26.1 Introduction

Everything in applied statistics rests on a distinction that sounds obvious until one tries to pin it down: the difference between the population a claim is about and the sample from which the claim is drawn. A *parameter* summarises the population; a *statistic* summarises the sample. The goal of inference is to use the statistic to say something defensible about the parameter. Confusion between the two is the single most common source of overreach in scientific reporting.

1.26.2 Prerequisites

The reader should know what an average is and should be comfortable with simple R arithmetic. No previous inferential statistics is assumed.

1.26.3 Theory

The **target population** is the set of units about which we want to make a claim: “all adults in Germany”, “all HER2-positive breast cancer patients treated at this hospital in 2020”, “all spikes recorded from V1 neurons in macaque M during task T”. It is defined *first*, in scientific terms, not by what is easy to measure.

The **sampling frame** is the operational list from which units are actually drawn: the register of registered voters; the hospital’s electronic health record; the 48 neurons that were successfully isolated. When the sampling frame differs from the target population, the sample may be representative of the frame but not of the intended population – *coverage bias*.

The **sample** is the subset of the frame actually observed. Its summary statistics (mean, proportion, correlation) are estimates of the corresponding population parameters. Inference is the machinery that quantifies how close the statistic is likely to be to the parameter, given the sampling scheme and the observed variability.

By convention, Greek letters denote population parameters and Latin letters denote sample statistics:

- Population mean μ, sample mean $\bar{x}$.
- Population variance σ^2, sample variance s^2.
- Population proportion π, sample proportion $\hat{p}$.
- Population regression coefficient β, estimate $\hat{\beta}$.

1.26.4 Assumptions

Inference about the population from the sample requires that the sampling scheme be *representative* – either by probability sampling (each unit in the frame has a known, positive probability of selection) or by an identified mechanism under which conclusions transfer (e.g., random assignment within a convenience sample, which supports causal but not prevalence claims).

1.26.5 R Implementation

```r
library(ggplot2)

set.seed(2026)
population <- rnorm(1e6, mean = 75, sd = 10)
mu    <- mean(population)
sigma <- sd(population)

sample_n  <- 50
one_sample <- sample(population, sample_n)
xbar <- mean(one_sample)
s    <- sd(one_sample)

c(population_mean = mu, sample_mean = xbar,
  population_sd   = sigma, sample_sd   = s)

sampling_distribution <- replicate(5000, mean(sample(population, sample_n)))
quantile(sampling_distribution, c(0.025, 0.975))
```

We treat `population` as a fictional census of one million values. A single sample of $n = 50$ gives a point estimate $\bar{x}$; repeated sampling traces out the sampling distribution of $\bar{x}$.

1.26.6 Output & Results

```
  population_mean       sample_mean  population_sd       sample_sd
        74.99806          76.51094       10.00139          9.47318

      2.5%       97.5%
 73.23143    76.78120
```

The sample mean is close to but not equal to the population mean; the sampling distribution tells us that over repeated sampling, 95% of sample means fall within about ± 1.4 of the truth.

1.26.7 Interpretation

When a paper reports "mean systolic blood pressure in the cohort was 128 mmHg (SD 14)", the reader should understand that this is a *statistic*, and that the population parameter it estimates has some uncertainty, quantified by the standard error or confidence interval. The methods section should state explicitly what population is targeted and through what frame.

1.26.8 Practical Tips

- Define the target population in plain scientific terms before the data arrive. Retrospective definitions invite bias.
- Document the sampling frame, inclusion/exclusion criteria, and selection probabilities. This is what makes a study reproducible.
- If the frame deviates from the target population (convenience sampling in clinical trials, online panels, self-selected respondents), state the deviation and the likely

direction of bias.
- Sample statistics are random; reporting only a point estimate without an SE or CI gives a false sense of precision.
- Never confuse "the distribution of the data" with "the sampling distribution of a statistic". The first describes individuals, the second describes averages (or other summaries) across repeated samples.

1.26.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.26.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.27 Sampling Distributions

1.27.1 Introduction

The sampling distribution of a statistic is the distribution of its values across hypothetical repetitions of the sampling process. Every confidence interval, every p-value, and every standard error is a statement about the sampling distribution of some statistic. Understanding this concept makes inference intuitive; ignoring it makes it a ritual of incantations.

1.27.2 Prerequisites

The reader should know what a sample mean is and should be comfortable with histograms and R's `replicate()` function.

1.27.3 Theory

Let $T = T(X_1, \ldots, X_n)$ be a statistic computed from a sample of size n drawn from a population with some distribution. The **sampling distribution** of T is the distribution of values T would take if the sampling process were repeated many times, each time drawing a fresh sample of size n from the same population.

Key facts:

- The sampling distribution is a property of T and of the sampling design, not of any particular observed sample.
- Its mean is the expectation $E[T]$. For unbiased estimators, $E[T]$ equals the population parameter.
- Its standard deviation is called the **standard error** of T: $\mathrm{SE}(T) = \sqrt{\mathrm{Var}(T)}$.
- Its shape depends on the statistic, on n, and on the population distribution. For the sample mean from a population with finite variance, the central limit theorem guarantees approximate normality as n grows.

For the sample mean $\bar{X}$ of n iid observations with mean μ and variance σ^2:

$$E[\bar{X}] = \mu, \qquad \mathrm{SE}(\bar{X}) = \sigma/\sqrt{n}.$$

The standard error tells us the typical distance between $\bar{X}$ and μ – it is the unit in which confidence intervals are measured.

1.27.4 Assumptions

The theoretical sampling distribution derived in most textbooks assumes iid sampling from a population with finite variance. Deviations (heavy tails, dependence, non-iid sampling) change the shape or the formulas; the bootstrap lets us approximate the sampling distribution empirically without these assumptions.

1.27.5 R Implementation

```
library(ggplot2)

set.seed(2026)
population ← rgamma(1e5, shape = 2, rate = 0.1)
mu ← mean(population)
sigma ← sd(population)

draw_mean ← function(n) mean(sample(population, n))

reps ← 10000
means_n10  ← replicate(reps, draw_mean(10))
means_n100 ← replicate(reps, draw_mean(100))

cat(sprintf("n=10  : mean=%.2f, SE=%.2f; theoretical SE=%.2f\n",
            mean(means_n10), sd(means_n10), sigma / sqrt(10)))
cat(sprintf("n=100 : mean=%.2f, SE=%.2f; theoretical SE=%.2f\n",
            mean(means_n100), sd(means_n100), sigma / sqrt(100)))

df ← data.frame(xbar = c(means_n10, means_n100),
                n = factor(rep(c(10, 100), each = reps)))
ggplot(df, aes(x = xbar, fill = n)) +
  geom_histogram(alpha = 0.6, bins = 50, position = "identity") +
  labs(x = expression(bar(X)), y = "Frequency") +
  theme_minimal()
```

For a skewed gamma population, the sampling distribution of the mean concentrates as n grows, and its spread matches the theoretical standard error.

1.27.6 Output & Results

```
n=10  : mean=19.98, SE=4.42; theoretical SE=4.47
n=100 : mean=20.01, SE=1.41; theoretical SE=1.41
```

Increasing n by a factor of 10 decreases the standard error by $\sqrt{10} \approx 3.16$, exactly as the $1/\sqrt{n}$ rule predicts.

1.27.7 Interpretation

The standard deviation of the sampling distribution – the SE – is what a confidence interval is built from. A 95% CI for the mean is approximately $\bar{x} \pm 1.96 \cdot \mathrm{SE}(\bar{X})$. When a manuscript reports "mean 19.98 (SE 1.41, 95% CI 17.22–22.74)", every number in that summary is derived from the sampling distribution of the statistic.

1.27.8 Practical Tips

- Do not confuse the standard deviation of the data with the standard error of a statistic. The SD describes variability across individuals; the SE describes variability across hypothetical samples of size n.
- When the theoretical sampling distribution is unknown, approximate it by the bootstrap: resample with replacement from the observed sample, compute the statistic, and use the empirical distribution of the resampled statistics as an estimate.
- The sampling distribution depends on n – larger samples yield tighter distributions, as the $1/\sqrt{n}$ rule captures for the mean.
- For statistics other than the mean, the sampling distribution may be harder to derive analytically but is equally well-defined; simulation or bootstrap is the general-purpose tool.
- The t distribution that appears in small-sample inference is the sampling distribution of $(\bar{X} - \mu)/(s/\sqrt{n})$ under normality with unknown variance, not of the mean itself.

1.27.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.27.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.28 Sampling Methods

1.28.1 Introduction

How a sample is drawn matters as much as how big it is. The same 500 observations can support valid inference about a population or produce a misleading summary, depending entirely on the sampling mechanism. Classical sampling theory names a handful of probability designs, each with characteristic strengths, variance properties, and analysis requirements.

1.28.2 Prerequisites

The reader should know the difference between a population and a sample, and should be familiar with random-number generation in R.

1.28.3 Theory

Simple random sampling (SRS) draws n units from a frame of N such that every subset of size n is equally likely. The sample mean is unbiased for the population mean, with standard error $\sigma/\sqrt{n} \cdot \sqrt{1 - n/N}$ (the finite-population correction is negligible for $n \ll N$). SRS is the theoretical baseline against which other designs are compared.

Stratified sampling partitions the population into non-overlapping strata – e.g., by region, age band, or hospital – and draws an SRS from each. The pooled estimator has *lower* variance than SRS whenever the strata are more homogeneous within than the population at large, which is typical. Stratification also guarantees representation of small subgroups that might be missed by chance under SRS.

Cluster sampling draws whole clusters (schools, villages, GP practices) and then includes some or all units within each. Variance is *higher* than SRS for the same n because units within a cluster are correlated; the loss is quantified by the design effect $\mathrm{deff} = 1 + (\bar{m} - 1)\rho$, where $\bar{m}$ is average cluster size and ρ is the intra-cluster correlation. Cluster sampling trades statistical efficiency for dramatically lower data-collection cost.

Systematic sampling picks every k-th unit from a list after a random start. It is equivalent to SRS when the list is in random order; when the list has periodicity matching k, it can give badly biased estimates.

Convenience sampling recruits whatever units are available – internet panels, patients at one hospital, undergraduates in a course. It supports no probability-based inference about a broader population; any generalisation must be argued on non-statistical grounds (mechanism, theory, replication).

1.28.4 Assumptions

Probability designs assume a complete sampling frame and known selection probabilities. Analysis must incorporate design weights ($w_i = 1/\pi_i$) and the design variance; standard software that ignores weighting (a t-test on weighted survey data, say) produces incorrect standard errors.

1.28.5 R Implementation

```r
library(sampling)

set.seed(2026)
N <- 10000
pop <- data.frame(
  id     = 1:N,
  stratum = sample(c("A", "B", "C"), N, replace = TRUE, prob = c(0.5, 0.3, 0.2)),
  y      = rnorm(N, mean = c(A = 50, B = 60, C = 70)[sample(c("A", "B", "C"), N,
           replace = TRUE, prob = c(0.5, 0.3, 0.2))], sd = 5)
)

srs_idx <- sample(N, 200)
srs <- pop[srs_idx, ]
mean(srs$y); sd(srs$y) / sqrt(200)

str_idx <- strata(pop, stratanames = "stratum",
                  size = c(A = 100, B = 60, C = 40),
                  method = "srswor")
strat <- pop[str_idx$ID_unit, ]
tapply(strat$y, strat$stratum, mean)
weighted.mean(tapply(strat$y, strat$stratum, mean),
              w = table(pop$stratum))
```

The code draws a 200-unit SRS and a 200-unit stratified sample from the same population, then computes the design-consistent stratified estimator via weighted averages.

1.28.6 Output & Results

```
[1] 57.41       # SRS mean
[1] 0.533       # SE of SRS mean

     A      B      C
 49.97  59.86  70.22   # stratum means

[1] 56.94       # stratified mean (weighted by stratum population proportions)
```

Both estimators target the same quantity; the stratified design has systematically smaller variance across repeated sampling.

1.28.7 Interpretation

A methods section should state the sampling design precisely: “Patients were sampled by stratified random sampling, stratified by treatment centre with proportional allocation, from a frame of 4 832 eligible records identified by the inclusion criteria.” This lets a reviewer evaluate whether analysis choices respect the design.

1.28.8 Practical Tips

- When using stratified or cluster designs, analyse with the `survey` package (or equivalent) so standard errors incorporate weights and clustering.
- Proportional allocation is a sensible default for stratified sampling; optimal (Neyman) allocation weights strata by $N_h\sigma_h$, which can be used when prior variance estimates are available.
- For cluster designs, report the design effect and the ICC; reviewers will ask.
- Systematic sampling is fine for logistical convenience, but inspect the sampling list for periodicity before using it.
- Convenience samples are not disqualified from publication, but their external validity must be argued explicitly; treating them as if they were random is the root cause of many replication failures.

1.28.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.28.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.29 Scales of Measurement

1.29.1 Introduction

Every variable in a dataset carries a scale of measurement, and the scale dictates which statistics are meaningful to compute. Stevens' 1946 taxonomy – nominal, ordinal, interval, ratio – remains the default vocabulary for this idea, and even when its strict version is criticised, the underlying distinctions are unavoidable in applied work. Computing a mean of jersey numbers is nonsense because jersey numbers are nominal; computing a median of exam grades is sensible because grades are at least ordinal.

1.29.2 Prerequisites

A reader should know what a variable is and should be able to tell numeric data from categorical data in R (as `numeric` vs. `factor` or `character`).

1.29.3 Theory

- **Nominal**: labels without order. Operations: equality. Examples: sex, blood type, country. Meaningful summaries: mode, frequency counts.
- **Ordinal**: ordered labels without meaningful distances. Operations: $=$, $<$, $>$. Examples: Likert scale, disease stage (I–IV), pain rating. Meaningful summaries: median, quantiles, rank correlations.
- **Interval**: ordered, with meaningful distances but no true zero. Operations: $+$, $-$. Examples: temperature in Celsius, calendar dates. Ratios are *not* meaningful: 40 °C is not "twice as hot" as 20 °C.
- **Ratio**: interval with a true zero. Operations: $\times$, $/$. Examples: mass, concentration, time elapsed, counts. Ratios are meaningful: 40 kg is twice 20 kg.

The scale determines permissible summary statistics and inferential procedures. A Pearson correlation requires interval or ratio data; a Spearman correlation extends to ordinal. A t-test operates on (at minimum) interval data, though in practice it is widely applied to Likert-like composites with reasonable robustness.

1.29.4 Assumptions

The taxonomy itself is descriptive rather than inferential; its use rests on the assumption that the chosen scale accurately characterises the variable. In practice, many variables sit ambiguously – is a 5-point Likert item ordinal or interval? Standard practice treats Likert *composite scales* (averages of many items) as interval and *single items* as ordinal.

1.29.5 R Implementation

```r
library(ggplot2)

df ← data.frame(
  id       = 1:8,
  country = factor(c("DE", "US", "FR", "DE", "US", "FR", "DE", "US")),
  stage    = factor(c("I", "II", "III", "II", "IV", "I", "III", "II"),
                    levels = c("I", "II", "III", "IV"), ordered = TRUE),
  temp_c   = c(36.5, 37.1, 38.2, 36.9, 37.4, 36.7, 38.0, 37.2),
  mass_kg = c(72.1, 81.3, 65.8, 74.2, 88.7, 61.4, 79.0, 84.5)
)

table(df$country)
median(as.numeric(df$stage))
mean(df$temp_c)
mean(df$mass_kg); df$mass_kg[1] / df$mass_kg[3]
```

Nominal `country` is summarised by frequencies; ordinal `stage` by the median (the raw factor must be coerced to numeric for the median call); interval `temp_c` by the mean; ratio `mass_kg` by the mean and by ratios.

1.29.6 Output & Results

```
DE FR US
 3  2  3

[1] 2      # median stage (II)
[1] 37.25  # mean temperature in C
[1] 75.875 # mean mass in kg
[1] 1.096  # mass ratio is meaningful for ratio data
```

1.29.7 Interpretation

A paper's methods section should list each variable and its scale, because the scale justifies the summary statistic shown in the descriptive-statistics table. "Disease stage (ordinal) is summarised by the median (IQR); body mass (ratio) by the mean (SD)." This prevents a reader from being surprised by, say, a mean of a four-level ordinal variable dressed up as a continuous measurement.

1.29.8 Practical Tips

- Encode ordinal variables in R as `factor(..., ordered = TRUE)` so that methods that respect ordering (proportional-odds models, Spearman correlation) pick up the order automatically.
- Resist the urge to compute means of ordinal variables for reporting; a mean of disease stages is uninterpretable even when it produces a number.
- Do not confuse a *count* (ratio) with a *code* (nominal): "category = 3" is nominal even though stored as an integer.
- Be careful with variables that look ratio but are not: pH is actually a logarithmic transformation of concentration, so differences on the pH scale correspond to ratios of hydrogen-ion concentration.
- When a variable's scale is ambiguous (Likert items), pre-register your analytical choice – ordinal or interval – so reviewers cannot object after the fact.

1.29.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.29.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.30 Slutsky's Theorem

1.30.1 Introduction

Slutsky's theorem is one of the workhorse results of asymptotic statistics. It says that we can substitute a consistent estimator for a constant in an asymptotic statement, without changing the limiting distribution. This simple-sounding result is what makes the Wald statistic have an asymptotic standard-normal distribution, what justifies using the sample SD in place of the population SD in a t-statistic's denominator, and what underlies countless routine asymptotic arguments in applied work.

1.30.2 Prerequisites

The reader should understand convergence in probability and convergence in distribution.

1.30.3 Theory

Slutsky's theorem. Suppose $Y_n \xrightarrow{d} Y$ and $Z_n \xrightarrow{P} c$, where c is a constant. Then:

- $Y_n + Z_n \xrightarrow{d} Y + c$,
- $Y_n Z_n \xrightarrow{d} cY$,
- $Y_n / Z_n \xrightarrow{d} Y/c$ (provided $c \neq 0$).

The key requirement is that Z_n converges to a *constant*, not to another random variable. Convergence of Z_n to a non-degenerate random variable does not generally justify these manipulations.

Typical use. Consider the Wald statistic for a mean:

$$W_n = \frac{\sqrt{n}(\bar{X}_n - \mu)}{s_n}.$$

We know $\sqrt{n}(\bar{X}_n - \mu)/\sigma \xrightarrow{d} \mathcal{N}(0,1)$ by the CLT, and $s_n \xrightarrow{P} \sigma$ by the LLN applied to the sample variance. Writing

$$W_n = \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \cdot \frac{\sigma}{s_n},$$

Slutsky's theorem gives $W_n \xrightarrow{d} \mathcal{N}(0,1) \cdot 1 = \mathcal{N}(0,1)$.

This is why the Wald statistic with an estimated SE still has a standard-normal limit. The same principle applies to Wald chi-squared statistics, score statistics, and almost every test statistic in asymptotic theory.

1.30.4 Assumptions

Slutsky's theorem requires (i) Z_n converges in probability to a constant, not to a random variable, and (ii) the combined operation (sum, product, quotient) is well-defined. For quotients the constant must not be zero; dividing by a small random denominator can still introduce tail issues at finite n that Slutsky does not warn about.

1.30.5 R Implementation

```
set.seed(2026)

n ← 500
mu ← 3; sigma ← 2
reps ← 5000

wald_stats ← replicate(reps, {
  x ← rnorm(n, mu, sigma)
  sqrt(n) * (mean(x) - mu) / sd(x)
})

qqnorm(wald_stats, pch = ".")
qqline(wald_stats, col = "red")

c(mean = mean(wald_stats),
  sd   = sd(wald_stats))
```

We simulate the Wald statistic across 5000 replications. Slutsky's theorem predicts the limiting distribution is standard normal; the simulation confirms empirically.

1.30.6 Output & Results

```
    mean       sd
 0.00411   0.9997
```

The empirical mean is near zero and SD near one, confirming standard-normal behaviour. The Q-Q plot falls on the reference line.

1.30.7 Interpretation

Slutsky's theorem is rarely cited by name in applied papers, but its conclusions appear constantly. The phrase "by the CLT, and using the consistency of $\hat{\sigma}$ for σ, the Wald statistic is asymptotically standard normal" is a Slutsky argument compressed into a single sentence.

1.30.8 Practical Tips

- When you see an asymptotic argument that swaps an estimator for a parameter, that is (usually) Slutsky at work.
- Slutsky's theorem does not help when both sequences are non-degenerate; for that, joint convergence and the continuous mapping theorem are needed.

- In finite samples, the Slutsky approximation may be crude. For small n, the t-distribution (which accounts for the estimated denominator more precisely) is preferred over the Wald normal.
- Extensions hold for continuous functions: if $Y_n \xrightarrow{d} Y$ and $Z_n \xrightarrow{P} c$, then $g(Y_n, Z_n) \xrightarrow{d} g(Y, c)$ for continuous g at (y, c).
- The theorem is sometimes stated with Z_n taking values in any fixed dimensional space – useful when the "constant" is a vector or matrix (e.g., an estimated covariance).

1.30.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.30.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.31 The Standard Error

1.31.1 Introduction

The **standard error** of a statistic is the standard deviation of its sampling distribution. It quantifies how much the statistic would vary across hypothetical repetitions of the sampling process. Every confidence interval and every t- or z-statistic is built on an estimate of the standard error, so understanding and correctly computing SEs is central to competent data analysis.

1.31.2 Prerequisites

A reader should know what the sampling distribution is, what a sample standard deviation is, and how to use the `sd()` and `mean()` functions in R.

1.31.3 Theory

For an iid sample $X_1, \ldots, X_n$ from a population with variance σ^2, the sample mean $\bar{X}$ has variance σ^2/n and therefore

$$\mathrm{SE}(\bar{X}) = \sigma/\sqrt{n}.$$

Because σ is usually unknown, it is estimated by the sample SD s, giving the plug-in estimator $\widehat{\mathrm{SE}}(\bar{X}) = s/\sqrt{n}$.

For a binary proportion $\hat{p} = k/n$, the standard error under binomial sampling is

$$\mathrm{SE}(\hat{p}) = \sqrt{p(1-p)/n}, \qquad \widehat{\mathrm{SE}}(\hat{p}) = \sqrt{\hat{p}(1-\hat{p})/n}.$$

For a difference in means across two independent samples with variances s_1^2, s_2^2,

$$\widehat{\mathrm{SE}}(\bar{X}_1 - \bar{X}_2) = \sqrt{s_1^2/n_1 + s_2^2/n_2}.$$

For regression coefficients, standard errors come from the diagonal of $\hat{\sigma}^2 (X^\top X)^{-1}$; R returns them in `summary(fit)$coefficients`.

For statistics without a closed-form SE (ratios, quantiles, custom summaries), the **bootstrap** gives a general-purpose estimator: resample with replacement B times, compute the statistic for each resample, and use the SD of the B bootstrap statistics as $\widehat{\mathrm{SE}}$.

1.31.4 Assumptions

The formula-based SEs above assume iid sampling and, for the finite-sample versions, enough regularity for the CLT to justify using them in normal-approximation CIs. The bootstrap SE is consistent under weaker conditions but inherits the design: it is accurate only when resampling units are the same as the independent sampling units (so bootstrap at the cluster level for cluster samples).

1.31.5 R Implementation

```r
library(boot)

set.seed(2026)
x <- rnorm(60, mean = 75, sd = 12)

se_mean <- sd(x) / sqrt(length(x))
se_mean

y <- c(rep(1, 24), rep(0, 76))
p_hat <- mean(y)
se_prop <- sqrt(p_hat * (1 - p_hat) / length(y))
se_prop

iqr_stat <- function(d, i) IQR(d[i])
b <- boot(x, statistic = iqr_stat, R = 5000)
se_iqr <- sd(b$t)
se_iqr
```

The first two blocks use the closed-form SEs for the mean and a proportion; the third uses the bootstrap for the IQR, which has no convenient closed form.

1.31.6 Output & Results

```
[1] 1.552        # SE(mean)
[1] 0.0427       # SE(proportion)
[1] 2.11         # bootstrap SE(IQR)
```

Each of these is the estimated typical distance between the sample statistic and the population parameter it estimates.

1.31.7 Interpretation

The standard error is what gets reported alongside the estimate: “mean 75.8 (SE 1.6)” or “proportion 0.24 (95% CI 0.16–0.32)”, where the CI is approximately the estimate plus/minus 1.96 times the SE. A reviewer glancing at the SE can judge whether the study is precise enough to support its claims.

1.31.8 Practical Tips

- Always distinguish the SD (spread of individual values) from the SE (spread of a statistic); papers regularly report the wrong one.
- For proportions near 0 or 1, the normal-approximation SE is poor; prefer Wilson or Clopper-Pearson intervals, which are available via `binom.test()` and the `binom` package.
- For differences between groups, the correct SE is the SE of the difference, not either group’s SE by itself.
- When you lack a closed-form SE, reach for the bootstrap rather than inventing one; `boot::boot()` handles the resampling machinery robustly.
- Do not report an SE without stating what statistic it refers to – “SE = 1.6” alone is ambiguous; “SE of the mean = 1.6” is clear.

1.31.9 For Reviewers

Use the shared reviewer checklist, together with this method’s assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript’s application of the method.

1.31.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.32 Sufficient Statistics

1.32.1 Introduction

A statistic is **sufficient** for a parameter if it captures all the information in the data that is relevant to that parameter. Once a sufficient statistic is computed, the raw data have nothing left to add about the parameter. Sufficiency organises the theory of estimation: it explains why the sample mean is a complete summary for the normal mean, why the sum of successes suffices for the binomial proportion, and why the exponential family of distributions is central to modern statistical modelling.

1.32.2 Prerequisites

The reader should be comfortable with the likelihood function, conditional probability, and a little abstract notation.

1.32.3 Theory

Formally, $T(X_1, \dots, X_n)$ is **sufficient for** θ if the conditional distribution of the data given $T = t$ does not depend on θ. Equivalently (Fisher-Neyman factorisation theorem), T is sufficient iff the joint density factorises as

$$f(x_1, \dots, x_n \mid \theta) = g\big(T(x), \theta\big) \cdot h(x),$$

where h does not depend on θ and g depends on the data only through T.

Examples:

- **Normal** (μ, σ^2) **with known** σ^2: $T = \bar{X}$ is sufficient for μ.
- **Normal with both unknown**: $(\bar{X}, \sum (X_i - \bar{X})^2)$ is jointly sufficient.
- **Bernoulli/Binomial**: $T = \sum X_i$ is sufficient for π.
- **Poisson**: $T = \sum X_i$ is sufficient for λ.
- **Uniform**$(0, \theta)$: $T = \max X_i$ is sufficient for θ.

A statistic is **minimal sufficient** if it is a function of every other sufficient statistic – i.e., it is the most compact sufficient summary.

The **exponential family** has density $f(x \mid \theta) = h(x) \exp[\eta(\theta)^\top T(x) - A(\theta)]$ for natural statistic $T(x)$. Almost every distribution used in applied statistics (normal, binomial, Poisson, gamma, beta, multinomial) belongs to the exponential family, and $T(x)$ is minimal sufficient for $\eta(\theta)$.

Rao-Blackwell theorem: given any unbiased estimator $\hat{\theta}$ and a sufficient statistic T, the conditional expectation $E[\hat{\theta} \mid T]$ is also unbiased and has variance no larger than $\hat{\theta}$. Conditioning on sufficient statistics never hurts and may help.

1.32.4 Assumptions

The factorisation theorem applies to proper densities on a common support. In non-regular families (uniform with an unknown upper endpoint) sufficiency still applies, but the resulting estimators inherit the non-standard behaviour of the family.

1.32.5 R Implementation

```r
set.seed(2026)

n ← 50
mu ← 5
sigma ← 2
x ← rnorm(n, mean = mu, sd = sigma)

T1 ← sum(x)
T2 ← sum(x^2)

log_lik_from_sum ← function(mu, sigma, T1, T2, n) {
  -n/2 * log(2 * pi * sigma^2) -
    (T2 - 2 * mu * T1 + n * mu^2) / (2 * sigma^2)
}

mu_hat ← T1 / n
sigma2_hat ← (T2 - T1^2/n) / n

c(mu_hat = mu_hat, sigma2_hat = sigma2_hat)

c(mle_from_data_mean = mean(x),
  mle_from_data_var  = mean((x - mean(x))^2))
```

Computing the MLE from only the two sufficient statistics $(T_1, T_2) = (\sum x, \sum x^2)$ yields the same point estimates as computing it from the full data vector.

1.32.6 Output & Results

```
    mu_hat sigma2_hat
     5.090      3.821

mle_from_data_mean  mle_from_data_var
             5.090              3.821
```

The two are numerically identical, confirming that (T_1, T_2) contains all the information about the parameters that the data can provide.

1.32.7 Interpretation

In practice, sufficiency is used under the hood. When a software package reports a confidence interval for a normal mean using `t.test()`, it relies on the fact that $\bar{X}$ and s^2 are sufficient for (μ, σ^2). For a manuscript, sufficiency is rarely mentioned explicitly but justifies the use of summary statistics as complete descriptions of the data in the parametric model assumed.

1.32.8 Practical Tips

- Sufficient statistics let you store compressed summaries (counts, sums) without losing information for inference – useful for distributed computation.
- The exponential family is the most important family to understand, because almost every GLM assumes it.
- Completeness plus sufficiency plus unbiasedness implies the UMVUE (Lehmann-Scheffe); many textbook UMVUE proofs rely on this.
- Rao-Blackwellisation is useful in MCMC: conditioning on sufficient statistics for part of the parameter vector reduces Monte Carlo variance.
- Sufficiency depends on the model. A statistic that is sufficient for μ in a normal model may carry no information about the same μ under a different data-generating process.

1.32.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.32.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.33 Unbiasedness, Consistency, Efficiency

1.33.1 Introduction

When multiple estimators are available for the same parameter, we need criteria to choose among them. Three classical criteria – unbiasedness, consistency, and efficiency – cover the most important properties: whether the estimator is correct on average (unbiasedness), whether it approaches the truth as more data arrive (consistency), and whether it uses the data as effectively as possible (efficiency).

1.33.2 Prerequisites

The reader should be comfortable with the notions of expectation, variance, and convergence in probability.

1.33.3 Theory

Unbiasedness. An estimator $\hat{\theta}$ is unbiased for θ if $E[\hat{\theta}] = \theta$ for every value of θ in the parameter space. Examples: the sample mean is unbiased for the population mean; the sample variance with divisor $n - 1$ is unbiased for σ^2; the plug-in variance with divisor n is biased.

Consistency. An estimator $\hat{\theta}_n$ is consistent for θ if it converges to θ in probability as $n \to \infty$:

$$\hat{\theta}_n \xrightarrow{P} \theta.$$

Unbiasedness and consistency are distinct: the sample median is consistent for the population median but can be biased in small samples; the MLE of σ^2 (divisor n) is biased in every finite sample but consistent.

Efficiency. Among unbiased estimators, the **efficient** one is the one with the smallest variance. The Cramer-Rao lower bound (CRLB) sets a floor on this variance:

$$\operatorname{Var}(\hat{\theta}) \geq \frac{1}{I(\theta)},$$

where $I(\theta)$ is the Fisher information for θ. An estimator that attains this bound is **UMVUE** (uniformly minimum-variance unbiased estimator). For regular models, the MLE is asymptotically efficient: its variance approaches the CRLB as n grows.

These three properties can be in tension. A biased estimator with tiny variance can have lower MSE than an unbiased estimator with large variance; a consistent estimator may converge slowly or may not exist in closed form.

1.33.4 Assumptions

These properties are defined within a model. Change the model, and an estimator that was unbiased, consistent, and efficient for one parameter may be none of the three for another.

1.33.5 R Implementation

```r
library(ggplot2)
set.seed(2026)

simulate_props <- function(n, reps = 5000) {
  mu <- 10
  sigma <- 3
  m  <- replicate(reps, mean(rnorm(n, mu, sigma)))
  md <- replicate(reps, median(rnorm(n, mu, sigma)))
  rbind(
    data.frame(n = n, est = "mean",   bias = mean(m) - mu, var = var(m)),
    data.frame(n = n, est = "median", bias = mean(md) - mu, var = var(md))
  )
}

res <- do.call(rbind, lapply(c(10, 50, 250, 1000), simulate_props))
res
```

We compare the sample mean and the sample median as estimators of the population mean μ (under a symmetric normal, so both target the same quantity). The mean is efficient; the median is less efficient but is also consistent.

1.33.6 Output & Results

```
      n    est         bias         var
1   10   mean   0.00285700   0.9066562
2   10 median   0.00162894   1.3963523
3   50   mean   0.00098014   0.1795648
4   50 median  -0.00132211   0.2743110
5  250   mean  -0.00083720   0.0359942
6  250 median  -0.00038881   0.0557902
7 1000   mean   0.00006472   0.0090152
8 1000 median  -0.00007128   0.0140841
```

Both estimators have bias near zero; variance shrinks at rate $1/n$ (the mean hits $\sigma^2/n = 9/n$; the median is higher by the factor $\pi/2 \approx 1.57$). The mean is thus more efficient by roughly 1.57.

1.33.7 Interpretation

For normally distributed data, the sample mean is both unbiased and efficient; choosing it over the median costs nothing and gains precision. For heavy-tailed data or data with outliers, the median is more robust, and the efficiency gap can reverse in the opposite direction. These are trade-offs that the three criteria make explicit.

1.33.8 Practical Tips

- Consistency is usually the bare minimum requirement; without it, more data does not help.
- Unbiasedness is a mathematical convenience that can be sacrificed for a smaller variance (lower MSE).

- The asymptotic efficiency of the MLE is why it is a default estimator in regular parametric models; the cost is specification error if the model is wrong.
- Efficiency is relative to a model: under misspecification, the "efficient" estimator may be inconsistent while a "less efficient" robust one remains consistent.
- Always compare estimators on the quantity that matters for the application – MSE, coverage of CIs, predictive error – not on a single property in isolation.

1.33.9 For Reviewers

Use the shared reviewer checklist, together with this method's assumptions and practical tips.

Method-specific reviewer notes have not yet been developed for this entry. The links above support a structured review; they do not certify the example or a manuscript's application of the method.

1.33.10 See also — labs in this chapter

- Scientific process and research workflow
- R, RStudio, Quarto, and renv toolchain
- Data types, tidy data, accuracy and precision
- Import, joins, and missingness with dplyr

1.34 Week 1, Session 3 — Data types, tidy data, accuracy and precision

Workflow labs use the variant template: Goal → Approach → Execution → Check → Report.

1.34.1 Learning objectives

- Classify a variable as nominal, ordinal, interval, or ratio and explain why the distinction dictates what summaries are legal.
- Reshape a rectangular dataset between wide and long form with `pivot_longer()` and `pivot_wider()`.
- Separate accuracy, precision, and bias by simulation, and say why a precise instrument can still be wrong.

1.34.2 Prerequisites

Lab 1.2 (toolchain).

1.34.3 Background

Statistics is largely the study of variation, and variation must be measured. Before we calculate anything we need to say what kind of variable we are looking at. A *nominal* variable labels groups without order. An *ordinal* variable orders them but does not quantify

the distance between them. *Interval* and *ratio* variables do quantify distances, and ratio variables have a true zero. Tools that average an ordinal variable or compute a standard deviation on a category code produce numbers, but not answers.

Tidy data is a convention, not a requirement, but adopting it pays immediate dividends: every variable gets a column, every observation a row, and every value a cell. Real datasets arrive in other shapes, so a large part of the job is reshaping them until they conform. The tools are `pivot_longer()` and `pivot_wider()`, and they are the two functions you will use most often in a working week.

The accuracy/precision/bias triangle is borrowed from measurement theory. Accuracy is how close a measurement is to truth on average. Precision is how tightly repeated measurements cluster. Bias is a systematic offset from truth. An instrument can be precise but biased (tight cluster off-target), accurate but imprecise (wide cluster centred on truth), or — the goal — both.

1.34.4 Setup

```
library(tidyverse)
set.seed(42)
theme_set(theme_minimal(base_size = 12))
```

1.34.5 1. Goal

Move a small simulated clinical dataset between wide and long forms and quantify accuracy, precision, and bias in a simulated measurement process.

1.34.6 2. Approach

We simulate three repeat measurements of blood pressure on twelve patients at two visits. This is a realistic wide-format dataset with a within-patient structure.

```
clinic ← tibble(
  id = sprintf("P%02d", 1:12),
  sex = sample(c("F", "M"), 12, replace = TRUE),
  visit1_sbp1 = rnorm(12, 140, 10),
  visit1_sbp2 = rnorm(12, 140, 10),
  visit1_sbp3 = rnorm(12, 140, 10),
  visit2_sbp1 = rnorm(12, 135, 10),
  visit2_sbp2 = rnorm(12, 135, 10),
  visit2_sbp3 = rnorm(12, 135, 10)
)
clinic ▷ select(id, starts_with("visit1")) ▷ head(3)
```

1.34.7 3. Execution

Reshape to a long form where each row is one reading.

```r
long ← clinic ▷
  pivot_longer(
    cols = starts_with("visit"),
    names_to = c("visit", "reading"),
    names_pattern = "visit(\\d)_sbp(\\d)",
    values_to = "sbp"
  ) ▷
  mutate(visit = factor(paste0("V", visit)),
         reading = as.integer(reading))
head(long)
```

```r
long ▷
  ggplot(aes(visit, sbp, colour = sex)) +
  geom_jitter(width = 0.1, alpha = 0.6) +
  labs(x = NULL, y = "SBP (mmHg)")
```

The long format makes it trivial to summarise by visit, by patient, or by visit-and-patient. The same plot in wide format would require custom code for each visit column.

```r
# Collapse three readings per visit to a mean.
by_visit ← long ▷
  group_by(id, visit, sex) ▷
  summarise(sbp_mean = mean(sbp), .groups = "drop")

# Pivot back to wide if the downstream analysis needs it.
wide ← by_visit ▷ pivot_wider(names_from = visit, values_from = sbp_mean)
head(wide)
```

1.34.8 4. Check

Simulate an instrument with a known truth (say, a calibrated reference of 120) and compare three candidate devices with different accuracy/precision profiles.

```r
truth ← 120
n ← 500
devices ← tibble(
  device_A = rnorm(n, mean = 120, sd = 1),   # accurate & precise
  device_B = rnorm(n, mean = 125, sd = 1),   # precise, biased
  device_C = rnorm(n, mean = 120, sd = 5)    # accurate on average, imprecise
) ▷
  pivot_longer(everything(), names_to = "device", values_to = "reading")

summary_table ← devices ▷
  group_by(device) ▷
  summarise(mean = mean(reading),
            sd = sd(reading),
            bias = mean(reading) - truth,
            rmse = sqrt(mean((reading - truth)^2)))
summary_table
```

```r
devices ▷
  ggplot(aes(reading, fill = device)) +
  geom_density(alpha = 0.5, colour = NA) +
  geom_vline(xintercept = truth, linetype = 2) +
  labs(x = "Reading", y = "Density")
```

1.34.9 5. Report

> In a simulated calibration experiment (*n* = `n` draws per device), the unbiased precise device had bias = `round(summary_table$bias[1], 2)` mmHg and SD = `round(summary_table$sd[1], 2)`; the biased precise device had bias = `round(summary_table$bias[2], 2)` and the same SD; the imprecise unbiased device had bias = `round(summary_table$bias[3], 2)` but SD = `round(summary_table$sd[3], 2)`. Root mean squared error captures accuracy overall and is the quantity to minimise.

The lesson is that *bias* and *variance* are separately identifiable. A device that is consistent is not necessarily correct, and a device that is correct on average may still be unusable on a single reading.

1.34.10 Common pitfalls

- Treating ordinal codes (1 = mild, 2 = moderate, 3 = severe) as if the steps were equal.
- Forcing data to stay wide because that is how the spreadsheet happened to arrive.
- Judging a measurement by precision alone. A small SD with a big bias is still the wrong number.
- Using the mean to summarise a nominal variable.

1.34.11 Further reading

- Wickham H (2014), *Tidy Data*. Journal of Statistical Software.
- Altman DG, Bland JM (1983), *Measurement in medicine*.

1.34.12 Session info

```
sessionInfo()
```

1.34.13 See also — chapter index

- Methods in this chapter
- Find your method (Chapter 0)

1.35 Week 1, Session 4 — Import, joins, and missingness with dplyr

Workflow labs use the variant template: Goal → Approach → Execution → Check → Report.

1.35.1 Learning objectives

- Use `filter()`, `select()`, `mutate()`, `arrange()`, `group_by()`, and `summarise()` to carry out routine data management with palmerpenguins.

- Join two tables on a shared key and explain the difference between inner, left, and anti joins.
- Spot, quantify, and respond to missing data without letting NAs propagate silently into a summary.

1.35.2 Prerequisites

Labs 1.2 and 1.3.

1.35.3 Background

The bulk of analytic time is spent moving data from the shape it arrived in to the shape an analysis requires. dplyr names the five verbs you need for most of that work — filter rows, select columns, mutate to create new variables, arrange to sort, summarise with an optional grouping. If you learn these six verbs well you will write less code and hide fewer bugs.

Joins are the other half of data management. Real datasets arrive in pieces — a demographics table, a laboratory table, a follow-up table — that must be connected on a shared identifier. `left_join()` keeps all rows on the left side; `inner_join()` keeps only matches; `anti_join()` returns the rows with no match, which is often where data-quality problems live.

Missingness is not rare. A realistic clinical dataset has NAs in at least half its variables, and the NAs are almost never uniformly distributed. The first question to ask of any new dataset is *how much is missing, and where?* The second is *why?* Missing-completely-at-random, missing-at-random, and missing-not-at-random are three distinct problems with different consequences for the analysis.

1.35.4 Setup

```
library(tidyverse)
library(palmerpenguins)
set.seed(42)
theme_set(theme_minimal(base_size = 12))
```

1.35.5 1. Goal

Reshape the `penguins` dataset with the standard dplyr verbs, join a derived table back on, and describe the pattern of missingness.

1.35.6 2. Approach

```
glimpse(penguins)
```

```
# Standard five verbs.
penguins |>
  filter(!is.na(body_mass_g)) |>
  select(species, sex, bill_length_mm, body_mass_g) |>
  mutate(mass_kg = body_mass_g / 1000) |>
  arrange(desc(mass_kg)) |>
  head(5)
```

```
# group_by + summarise: the grammar of a Table 1 row.
species_summary <- penguins |>
  group_by(species, sex) |>
  summarise(
    n    = n(),
    mean_mass = mean(body_mass_g, na.rm = TRUE),
    sd_mass   = sd(body_mass_g,   na.rm = TRUE),
    .groups = "drop"
  )
species_summary
```

1.35.7 3. Execution

Simulate a small “islands” table with information not in `penguins`, and join it back.

```
islands <- tibble(
  island = c("Biscoe", "Dream", "Torgersen"),
  lat    = c(-65.43, -64.73, -64.77),
  researcher = c("Anna", "Ben", "Carla")
)

penguins_aug <- penguins |>
  left_join(islands, by = "island")

penguins_aug |>
  select(species, island, researcher, lat) |>
  head(5)
```

```
# anti_join finds rows in penguins with no matching island row.
penguins |>
  anti_join(islands, by = "island")
```

```
# inner_join keeps only matches on both sides.
inner_join(penguins, islands, by = "island") |>
  nrow()
```

1.35.8 4. Check

Quantify missingness at a glance.

```
missingness <- penguins |>
  summarise(across(everything(), ~ mean(is.na(.)))) |>
  pivot_longer(everything(),
               names_to = "variable",
               values_to = "frac_missing") |>
  arrange(desc(frac_missing))
missingness
```

```
missingness ▷
  ggplot(aes(frac_missing, reorder(variable, frac_missing))) +
  geom_col(fill = "grey60") +
  labs(x = "Fraction missing", y = NULL)
```

Now inject missingness to show how NAs move through a pipeline.

```
n ← nrow(penguins)
inj ← penguins ▷
  mutate(body_mass_g = if_else(runif(n) < 0.1, NA_real_, body_mass_g))
mean(inj$body_mass_g)            # NA — propagates
mean(inj$body_mass_g, na.rm = TRUE)
```

1.35.9 5. Report

> Of the 344 rows in `palmerpenguins::penguins`, sex was missing in `sum(is.na(penguins$sex))` rows (`round(100 * mean(is.na(penguins$sex)), 1)` %) and the four morphometric variables were missing in two rows apiece. After an inner join with a three-row `islands` table, all rows matched. Missingness was quantified per-variable and any summary that took a mean used `na.rm = TRUE` explicitly; silent NA propagation would otherwise have produced an NA mean.

Always look at the pattern of missingness before running an analysis. If NAs cluster in one group, a list-wise delete changes the comparison you thought you were making.

1.35.10 Common pitfalls

- Forgetting `na.rm = TRUE` in a `mean()` and getting back an NA.
- Confusing inner and left join and silently dropping rows.
- Using row numbers as join keys after a reshape (they will not match).
- Imputing missingness by hand and forgetting to flag the imputed rows.

1.35.11 Further reading

- Wickham H, Grolemund G. *R for Data Science*, chapter on data transformation.
- Little RJA, Rubin DB. *Statistical Analysis with Missing Data*.

1.35.12 Session info

```
sessionInfo()
```

1.35.13 See also — chapter index

- Methods in this chapter
- Find your method (Chapter 0)

1.36 Week 1, Session 2 — R, RStudio, Quarto, and renv toolchain

Workflow labs use the variant template: Goal → Approach → Execution → Check → Report.

1.36.1 Learning objectives

- Describe the role of R, RStudio, Quarto, and renv in a reproducible analysis toolchain.
- Initialise a project that a collaborator can clone and run without guesswork.
- Produce a rendered Quarto document from an R chunk that draws on a pinned package environment.

1.36.2 Prerequisites

R, RStudio, and Quarto installed (see Get started).

1.36.3 Background

A statistical analysis is only as trustworthy as the environment that produced it. A result that depends on the version of a package, the operating system of the analyst, or an undocumented installation step cannot be confidently rerun by anyone else — including the analyst six months later. The job of the toolchain is to make the set of dependencies explicit and easy to restore.

R provides the language; RStudio provides an integrated environment that keeps project files, editor, console, and version control in one place; Quarto renders source documents into articles, slides, and reports while interleaving text and executable code. The last piece, renv, pins package versions to a lockfile that lives with the project. Together these give you a project that is self-contained, rebuildable on another machine, and testable against changes in its own dependencies.

Beginners often think of reproducibility as a requirement imposed by journals. The real payoff is more selfish: a colleague can pick up a project without you having to answer questions, a revision cycle does not force you to reconstruct the environment you used six months ago, and a silent upgrade of a package does not break a graph you no longer remember how to draw.

1.36.4 Setup

```
library(tidyverse)
set.seed(42)
theme_set(theme_minimal(base_size = 12))
```

1.36.5 1. Goal

Stand up a minimal reproducible R project and demonstrate the commands that keep it reproducible: version reporting, environment snapshotting, and rendering.

1.36.6 2. Approach

A reproducible project has four minimum ingredients: a working directory, a package lockfile, a document that can be re-rendered, and a record of the R session. In this lab we simulate a tiny dataset, draw a figure from it, and record the environment used to produce both.

```
lab ← tibble(
  subject = seq_len(30),
  group   = rep(c("A", "B"), each = 15),
  value   = c(rnorm(15, mean = 10, sd = 2),
              rnorm(15, mean = 12, sd = 2))
)
lab
```

1.36.7 3. Execution

```
lab ▷
  ggplot(aes(group, value, fill = group)) +
  geom_boxplot(alpha = 0.6, colour = "grey30") +
  labs(x = "Group", y = "Measured value") +
  theme(legend.position = "none")
```

The chunk above is what a reader would rerun. It must be self-contained: the data are simulated inside the document, the seed is set, and only tidyverse is required.

1.36.7.1 Project anatomy

A minimal project contains:

- `my-project.Rproj` — the RStudio project file (an anchor for the working directory).
- `renv.lock` — the JSON lockfile of pinned package versions.
- `renv/` — the project's private package library.
- An `index.qmd` or report `.qmd` — your narrative.
- A `code/` or `R/` folder — your scripts.
- A `data/` folder — ideally small and read-only; never raw CSVs you edited by hand.

1.36.7.2 renv in three commands

```
# run once per project, after creating the .Rproj
renv::init()

# after installing or upgrading packages
renv::snapshot()

# on a new machine, after cloning the repo
renv::restore()
```

The `renv::status()` command reports drift between the library and the lockfile. In a shared project, your rule of thumb is: if `status()` reports anything other than "project is synchronised with the lockfile", do not commit.

1.36.8 4. Check

```
R.version.string
```

```
# A minimal package-version report we can include in any write-up.
installed_versions ← tibble(
  package = c("tidyverse", "ggplot2", "dplyr"),
  version = sapply(c("tidyverse", "ggplot2", "dplyr"),
                   function(p) as.character(packageVersion(p)))
)
installed_versions
```

The two outputs above — R version plus a small table of key package versions — are what a reviewer needs to rerun the figure. Everything larger (`sessionInfo()`) lives at the bottom of the document.

1.36.9 5. Report

> A reproducible analysis is carried out in a project folder anchored by an `.Rproj` file, with dependencies pinned in `renv.lock` and narrative plus code in a Quarto document. In this lab, 30 simulated observations were generated from two normal distributions and plotted as a boxplot; the document was rendered with R `R.version.string` and tidyverse `packageVersion("tidyverse")` .

The important phrase is *self-contained*. A file that someone else can run by cloning the repository and typing `quarto render` is the unit of reproducibility in this course.

1.36.10 Common pitfalls

- Installing packages outside the project library, then wondering why `renv::restore()` on another machine produces different output.
- Editing `renv.lock` by hand. Do not. Use `snapshot()` .
- Committing `renv/library/` to git. The lockfile is what is tracked; the library is restored locally.
- Rendering a Quarto doc that reads a file outside the project directory, then sharing only the `.qmd` .

1.36.11 Further reading

- Quarto documentation: quarto.org.
- renv documentation: `vignette("renv")` .
- Wickham H. *R Packages*, chapter on project organisation.

1.36.12 Session info

```
sessionInfo()
```

1.36.13 See also — chapter index

- Methods in this chapter
- Find your method (Chapter 0)

1.37 Week 1, Session 1 — Scientific process and research workflow

Workflow labs use the variant template: Goal → Approach → Execution → Check → Report.

1.37.1 Learning objectives

- Name the nine stages of the research workflow and say what goes wrong if each is skipped.
- Distinguish biological from statistical significance with an example.
- Frame a research question that can survive a methods section.

1.37.2 Prerequisites

R, RStudio, and Quarto installed (see Get started).

1.37.3 Background

Statistics is a thread running through a longer process that begins before a dataset exists. The research workflow — Question → Measurements → Design → Acquisition → Description → Analysis → Interpretation → Validation → Knowledge — names the stages of that process and makes clear that errors made at one stage are paid for at the next. A well-posed question protects you from a bad design; a good design protects you from a noisy dataset; a clean dataset protects you from needing exotic methods. Much of what looks like statistical failure in the literature is in fact a workflow failure three stages earlier.

The workflow also organises how you write up the work. The Introduction names the question; the Methods describe the measurements, design, and analysis; the Results describe the data and the inference; the Discussion interprets. When you plan an analysis in this order, the paper nearly writes itself. When you skip a stage, reviewers notice exactly which one.

1.37.4 Setup

```
library(tidyverse)
set.seed(42)
theme_set(theme_minimal(base_size = 12))
```

1.37.5 1. Goal

Make the research workflow concrete by simulating a small clinical scenario and mapping each stage to a specific decision.

1.37.6 2. Approach

Imagine a new blood-pressure drug. Our simulated question: *does drug X reduce systolic blood pressure compared with placebo in adults with hypertension?* We will generate data that reflect a realistic effect and walk through the workflow.

```
n ← 120
df ← tibble(
  id    = seq_len(n),
  arm   = rep(c("placebo", "drug"), each = n / 2),
  sbp_0 = rnorm(n, mean = 150, sd = 10),
  sbp_1 = sbp_0 + ifelse(arm == "drug", -6, 0) + rnorm(n, 0, 8)
)
```

1.37.7 3. Execution

```
df ▷
  mutate(delta = sbp_1 - sbp_0) ▷
  ggplot(aes(arm, delta, fill = arm)) +
  geom_boxplot(alpha = 0.6, colour = "grey30") +
  labs(x = NULL, y = "Change in SBP (mmHg)") +
  theme(legend.position = "none")
```

The plot is the Description stage. Without it, we would skip straight from raw numbers to a test. With it, the direction and spread of the change are visible before a single calculation.

1.37.8 4. Check

```
fit ← t.test(sbp_1 - sbp_0 ~ arm, data = df)
fit
```

The drug arm has a lower change-in-SBP on average; the interval is compatible with a clinically meaningful effect.

1.37.9 5. Report

> In a simulated two-arm trial (n = 120), mean change in systolic blood pressure was `round(diff(tapply(df$sbp_1 - df$sbp_0, df$arm, mean)), 1)` mmHg lower in the drug arm than in placebo (95% CI: `round(fit$conf.int[1], 1)` to `round(fit$conf.int[2], 1)`; p = `signif(fit$p.value, 2)`).

Note the three pieces every report needs: direction and magnitude, an interval, and a p-value reported alongside — never instead of — the effect.

1.37.9.1 Biological vs statistical significance

A p-value below 0.05 does not, on its own, mean the effect matters clinically. A reduction of 0.1 mmHg in SBP would be statistically significant with a million patients but biologically meaningless. The discipline of asking *how large is the effect?* — not *is there any effect?* — is the single most important habit to form early on.

1.37.10 Common pitfalls

- Skipping the description stage and running a test on data you have never plotted.
- Reporting a *p*-value without an effect size and interval.
- Conflating biological and statistical significance.
- Treating the research workflow as optional once data are in hand.

1.37.11 Further reading

- Research workflow appendix.
- Altman DG. *Practical Statistics for Medical Research*, ch. 1.
- Wasserstein RL & Lazar NA (2016), *The ASA's Statement on p-Values*.

1.37.12 Session info

```
sessionInfo()
```

1.37.13 See also — chapter index

- Methods in this chapter
- Find your method (Chapter 0)