2 Descriptive Statistics

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

Methods for summarising a dataset before any inference: Table 1 construction, numerical summaries, and visual exploratory data analysis. Most papers in biomedicine open with a Table 1 — this chapter covers what belongs in it and how to render it cleanly with `gtsummary`.

This chapter contains **25 method pages** and **1 lab**. If you are not sure which method to read, return to Chapter 0 and follow the decision tree to the right node.

2.1 Method pages

Method	Source slug
Coefficient of Variation	`coefficient-of-variation`
Contingency Tables	`contingency-tables`
Cross-Tabulations	`cross-tabulations`
Descriptive vs Inferential Statistics	`descriptive-vs-inferential`
Five-Number Summary	`five-number-summary`
Frequency Tables	`frequency-tables`
Kurtosis	`kurtosis`
Mean Absolute Deviation	`mean-absolute-deviation`
Measures of Central Tendency	`measures-of-location`
Measures of Dispersion	`measures-of-dispersion`
Measures of Shape	`measures-of-shape`
Outlier Detection Rules	`outlier-detection-rules`
Percentile Ranks	`percentile-rank`
Quantiles and Percentiles	`quantiles-and-percentiles`
Robust Statistics: An Overview	`robust-statistics-overview`
Skewness	`skewness`
Standardisation and z-Scores	`standardisation-zscore`
Stem-and-Leaf Plots	`stem-and-leaf-plots`
Summary by Group	`summary-by-group`
The Geometric Mean	`geometric-mean`
The Harmonic Mean	`harmonic-mean`
The Interquartile Range	`interquartile-range`
The Weighted Mean	`weighted-mean`
Trimmed and Winsorised Means	`trimmed-winsorized-mean`

Method	Source slug
Variance and Standard Deviation	`variance-and-sd`

2.2 Labs

Lab
Descriptive statistics and Table 1

2.3 Coefficient of Variation

2.3.1 Introduction

The coefficient of variation (CV) is the ratio of the standard deviation to the mean, usually expressed as a percentage. It provides a scale-free measure of dispersion, useful for comparing variability across variables measured on different units or at very different magnitudes. In laboratory medicine, the CV is the standard metric for assay precision; a CV below 5% is typical for a clinical chemistry assay.

2.3.2 Prerequisites

The reader should know what the mean and SD are.

2.3.3 Theory

For a positive-valued variable with mean μ and SD σ,

$$\mathrm{CV} = \sigma/\mu \quad (\text{or } 100 \cdot \sigma/\mu \text{ as a percentage}).$$

The CV is unit-less: comparing CVs across weight-in-kg and weight-in-pounds gives the same value. It is meaningful only for variables on a ratio scale (true zero) and is unstable when the mean is close to zero.

A related quantity is the **relative standard error (RSE)** = SE / mean. It measures the precision of the mean estimate itself rather than the dispersion of individuals.

2.3.4 Assumptions

- The variable must be on a ratio scale (mass, concentration, time, count).
- The mean must be comfortably far from zero; CV approaches infinity as the mean approaches zero.
- For CV comparisons across groups to be meaningful, the variables must be of the same type.

2.3.5 R Implementation

```r
set.seed(2026)

# Two assays with different concentration ranges
low  <- rnorm(50, mean = 2.5, sd = 0.25)
high <- rnorm(50, mean = 85.0, sd = 6.8)

cv_pct <- function(x) 100 * sd(x) / mean(x)

c(low_cv  = cv_pct(low),
  high_cv = cv_pct(high))
```

2.3.6 Output & Results

```
low_cv high_cv
  10.0    8.0
```

The low-concentration assay has a slightly higher CV (10%) than the high-concentration one (8%); the absolute SD of low is only 0.25 vs. 6.8 for high, but after scaling by the mean the two are comparable.

2.3.7 Interpretation

Report CV for inter-assay variability, instrument precision, or any context where comparing variability across scales matters. A CV of 10% is conventional for clinical-chemistry imprecision; a CV over 20% signals an assay in need of optimisation.

2.3.8 Practical Tips

- CV is undefined or nonsensical for variables centred near zero; use absolute SD instead.
- Do not report CV for interval-scale variables (temperature in Celsius).
- Log-CV = SD of log-values is approximately equal to CV on the original scale for small CVs; for large CVs the approximation breaks down.
- In intra-assay vs. inter-assay contexts, report both CVs separately.
- The CV of a mean $\mathrm{CV}(\bar{x})$ is CV(x) / $\sqrt{n}$; this connects precision to sample size.

2.3.9 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.

2.3.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.4 Contingency Tables

2.4.1 Introduction

A contingency table is a cross-tabulation treated as the input to a statistical test of independence. The counts in the cells, the expected counts under the independence null, and the standardised residuals that localise any deviation are its three essential components.

2.4.2 Prerequisites

The reader should know what a cross-tabulation is.

2.4.3 Theory

For an $r \times c$ contingency table with observed counts n_{ij}, row totals n_{i+}, column totals n_{+j}, and grand total n:

- **Expected count** under independence: $E_{ij} = n_{i+} \cdot n_{+j} / n$.
- **Pearson residual**: $e_{ij} = (n_{ij} - E_{ij}) / \sqrt{E_{ij}}$.
- **Standardised residual**: $e_{ij} / \sqrt{(1 - n_{i+}/n)(1 - n_{+j}/n)}$; approximately standard normal under the null.
- **Chi-squared statistic**: $\chi^2 = \sum (n_{ij} - E_{ij})^2 / E_{ij}$; under the null, it follows χ^2 with $(r-1)(c-1)$ df.

Standardised residuals beyond ± 2 flag cells that substantially exceed or fall short of independence.

2.4.4 Assumptions

For the chi-squared approximation: independent observations and all expected counts ≥ 5. For small expected counts, use Fisher's exact test.

2.4.5 R Implementation

```r
library(vcd)

cohort ← matrix(c(24, 18, 36, 32, 18, 42, 16, 24, 40),
                nrow = 3, byrow = TRUE,
                dimnames = list(Stage = c("I", "II", "III"),
                                Response = c("CR", "PR", "SD")))
cohort

chi ← chisq.test(cohort)
chi
chi$expected
chi$residuals         # Pearson residuals
chi$stdres            # standardised residuals

# Visual representation of residuals
mosaic(cohort, shade = TRUE, legend = TRUE)
```

2.4.6 Output & Results

```
        Response
Stage    CR PR SD
  I      24 18 36
  II     32 18 42
  III    16 24 40

Expected:
       CR   PR   SD
  I   23.5 18.3 36.2
  II  27.4 21.4 43.2
  III 21.1 16.5 33.4

Standardised residuals:
       CR    PR    SD
  I   0.15 -0.09 -0.08
  II  1.32 -1.02 -0.30
  III -1.50  2.18  1.53
```

No cell's standardised residual exceeds ± 2.5, so none drives a substantial departure from independence. The chi-squared p-value ($p \approx 0.22$ here) is non-significant.

2.4.7 Interpretation

Report the contingency table alongside the chi-squared statistic and a measure of association (Cramer's V). Standardised residuals are reported when the omnibus test is significant and the investigator wants to localise the effect.

2.4.8 Practical Tips

- `chisq.test()` returns the expected counts, Pearson residuals, and standardised residuals as attributes.
- Mosaic plots (`vcd::mosaic`) encode cell deviations with colour – a visual alternative to residual tables.
- For small tables with sparse cells, `fisher.test()` (or Monte Carlo chi-squared) is required.
- For stratified 2×2 tables across a third variable, use Cochran-Mantel-Haenszel (`mantelhaen.test`).
- `vcd::assocstats()` returns phi, contingency coefficient, and Cramer's V in one call.

2.4.9 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.

2.4.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.5 Cross-Tabulations

2.5.1 Introduction

A cross-tabulation shows the joint distribution of two categorical variables. It is the first look at whether the variables are associated and is the input to chi-squared, Fisher's exact, and other tests of independence.

2.5.2 Prerequisites

The reader should know what a one-way frequency table is.

2.5.3 Theory

For two categorical variables X with r levels and Y with c levels, the cross-tabulation is an $r \times c$ table whose (i, j)-th cell holds the count of observations with $X = i$ and $Y = j$. Typically reported forms:

- **Joint counts**: raw n_{ij}.
- **Row percentages**: $100 \cdot n_{ij} / \sum_j n_{ij}$. Answers "given $X = i$, what is the distribution of Y?"
- **Column percentages**: $100 \cdot n_{ij} / \sum_i n_{ij}$. Answers "given $Y = j$, what is the distribution of X?"
- **Overall percentages**: $100 \cdot n_{ij} / n$. Joint probabilities.

2.5.4 Assumptions

Purely descriptive; no distributional assumption.

2.5.5 R Implementation

```r
library(dplyr)
library(janitor)
library(gtsummary)

set.seed(2026)
cohort <- tibble::tibble(
  sex   = factor(sample(c("Female", "Male"), 300, replace = TRUE)),
  arm   = factor(sample(c("Placebo", "Active"), 300, replace = TRUE)),
  event = factor(sample(c("Yes", "No"), 300, replace = TRUE, prob = c(0.25, 0.75)))
)

# Base R
table(cohort$arm, cohort$event)

# Percentages
addmargins(prop.table(table(cohort$arm, cohort$event), margin = 1), margin = 2)

# Tidy with janitor
cohort |>
  tabyl(arm, event) |>
  adorn_percentages("row") |>
  adorn_pct_formatting(digits = 1) |>
  adorn_ns()

# gtsummary
cohort |>
  tbl_cross(row = arm, col = event, percent = "row", margin = "column")

# Three-way via xtabs
xtabs(~ sex + arm + event, data = cohort)
```

2.5.6 Output & Results

A typical 2×2 table:

	Event: No	Event: Yes	Total
Placebo	118 (78.7%)	32 (21.3%)	150
Active	104 (69.3%)	46 (30.7%)	150

Row percentages show the event rate within each arm; column percentages would instead show what fraction of events came from each arm.

2.5.7 Interpretation

For association analysis, report counts and one of row, column, or joint percentages – the one that matches the causal direction of your research question. “Among placebo patients, 21.3% experienced an event” is a row-percentage statement appropriate when arm precedes event.

2.5.8 Practical Tips

- Always state whether percentages are row, column, or overall; it is a common source of reader confusion.
- Sparse cells (count < 5) motivate Fisher's exact test rather than chi-squared.
- For three-way or higher tables, consider log-linear models (`loglm` in `MASS`).
- `gtsummary::tbl_cross()` produces publication-ready output directly from a data frame.
- Report totals on the margins; they let readers check arithmetic quickly.

2.5.9 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.

2.5.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.6 Descriptive vs Inferential Statistics

2.6.1 Introduction

Every applied statistics course opens with a distinction that sounds obvious but is routinely blurred in papers: **descriptive** statistics summarise the sample at hand; **inferential** statistics make claims about the population from which the sample was drawn. Confusing them is the single most common reporting error in published research.

2.6.2 Prerequisites

The reader should know what a sample and a population are, and have encountered at least one summary statistic (mean, proportion).

2.6.3 Theory

A **descriptive statistic** is a numerical summary computed from the sample: mean, SD, median, IQR, counts, percentages. It is deterministic given the data; the same dataset always produces the same descriptive summary.

An **inferential statistic** uses the sample to say something about the population: a confidence interval, a hypothesis test, a regression coefficient with a standard error, a Bayesian credible interval. It carries an explicit or implicit uncertainty quantification.

The distinction maps cleanly onto notation: descriptive statistics (Latin letters $\bar{x}$, s, $\hat{p}$) versus population parameters (Greek letters μ, σ, π) that inferential procedures estimate.

Both play complementary roles in a paper:

- **Descriptive**: Table 1 characterising the sample; figures showing distributions.
- **Inferential**: Primary analysis results; p-values and confidence intervals for effect sizes.

2.6.4 Assumptions

Descriptive statistics require only that the data be correctly transcribed. Inferential statistics require a sampling-model assumption: that the sample reasonably represents the population (random sampling, randomised assignment, or a defensible non-random scheme).

2.6.5 R Implementation

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

trial ← data.frame(
  arm    = factor(rep(c("Placebo", "Active"), each = 60)),
  sbp    = c(rnorm(60, 138, 14), rnorm(60, 130, 14))
)

# Descriptive: summarise the sample
aggregate(sbp ~ arm, data = trial,
          FUN = function(x) c(n = length(x), mean = mean(x), sd = sd(x)))

# Inferential: make a claim about the population
fit ← t.test(sbp ~ arm, data = trial)
tidy(fit)
```

2.6.6 Output & Results

```
      arm n    mean    sd
  Placebo 60  137.82 14.05
   Active 60  130.39 13.28

 # A tibble: 1 x 10
   estimate estimate1 estimate2 statistic p.value parameter conf.low conf.high
      <dbl>     <dbl>     <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl>
 1     7.43      138.      130.      2.99 0.00333      118.     2.52      12.3
```

Descriptive: placebo arm had a mean SBP of 137.8 mmHg (SD 14.1); active arm 130.4 (SD 13.3). Inferential: the difference is 7.4 mmHg with a 95 % CI of 2.5 to 12.3 mmHg, $p = 0.003$.

2.6.7 Interpretation

A paper's **Results** section typically progresses from descriptive to inferential: first describe the sample (Table 1), then test the pre-specified hypotheses. Mislabelling the two – treating a confidence interval as if it described the sample, or treating a sample mean as a population parameter – is a common reviewer complaint.

2.6.8 Practical Tips

- When you write "mean SBP was 137.8 mmHg" you have made a descriptive statement; when you write "SBP is 137.8 mmHg in the population" you have made an inferential statement (and should include a CI).
- Inferential statistics depend on a well-defined sampling frame; when the sample is a convenience sample, the inference is about the sample itself, not a broader population.
- Use distinct notation in equations: $\bar{x}$ for the sample mean, μ for the population mean.
- Every p-value and every confidence interval should be accompanied by a descriptive summary (counts, means, proportions) so the reader has both the effect and its context.
- A purely descriptive paper (audit, registry summary) is legitimate; pretending it is inferential is not.

2.6.9 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.

2.6.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.7 Five-Number Summary

2.7.1 Introduction

The five-number summary is a compact description of a univariate distribution in five values: minimum, first quartile, median, third quartile, and maximum. It was popularised by John Tukey in his exploratory data analysis tradition and is the numeric backbone of the boxplot. For skewed or non-normal data it conveys more than the mean and SD in the same space.

2.7.2 Prerequisites

The reader should know what quantiles are and how to sort a vector in R.

2.7.3 Theory

Given a sorted sample $x_{(1)} \le \ldots \le x_{(n)}$, the five-number summary is

$$(x_{(1)}, Q_1, Q_2, Q_3, x_{(n)}),$$

where Q_1, Q_2, Q_3 are the 0.25, 0.50, and 0.75 quantiles. Tukey's **hinges** are slightly different from the Type-7 quantiles R uses by default; the `fivenum()` function returns hinges, while `quantile(..., probs = c(0, 0.25, 0.5, 0.75, 1))` returns Type-7 quantiles.

The **interquartile range** $\mathrm{IQR} = Q_3 - Q_1$ measures the width of the central 50% of the data. The **Tukey fences** at $Q_1 - 1.5 \cdot \mathrm{IQR}$ and $Q_3 + 1.5 \cdot \mathrm{IQR}$ define the whiskers of a boxplot; points beyond them are flagged as outliers.

2.7.4 Assumptions

The five-number summary is purely descriptive and assumes nothing about the distribution.

2.7.5 R Implementation

```
library(dplyr)

set.seed(2026)
x <- rnorm(80, mean = 70, sd = 12)

fivenum(x)
summary(x)

iqr_x <- IQR(x)
fences <- c(lower = quantile(x, 0.25) - 1.5 * iqr_x,
            upper = quantile(x, 0.75) + 1.5 * iqr_x)
fences
sum(x < fences["lower"] | x > fences["upper"])

boxplot(x, horizontal = TRUE,
        main = "Five-number summary of x")
```

For grouped summaries:

```
trial <- tibble::tibble(
  arm = factor(rep(c("Placebo", "Active"), each = 40)),
  fpg = c(rnorm(40, 8.1, 0.9), rnorm(40, 7.0, 1.0))
)

trial |>
  group_by(arm) |>
  summarise(min = min(fpg), q25 = quantile(fpg, 0.25),
            med = median(fpg), q75 = quantile(fpg, 0.75),
            max = max(fpg))
```

2.7.6 Output & Results

Typical output of `fivenum(x)`:

```
[1] 42.8 62.1 70.4 78.9 101.2
```

The boxplot displays each of these five numbers: the whiskers at min and max (or the Tukey fence if outliers are flagged), the hinges at Q1 and Q3, and the thick line at the median.

2.7.7 Interpretation

Use the five-number summary for skewed or small-sample data. Reporting mean and SD for such data misrepresents typicality; the median and IQR are more informative.

2.7.8 Practical Tips

- The boxplot is the default graphical display of the five-number summary; use it early in exploratory analysis.
- When comparing groups, put boxplots side by side and notice whether the boxes overlap and how the medians align.
- Many journals accept boxplots only with the underlying data overlaid (e.g., jittered points or raincloud plots); check the submission guidelines.
- For small samples ($n < 10$), Tukey fences flag many normal observations as outliers; rely on the data, not the fence.
- Report all five numbers in text or in a table alongside the figure, so readers have the exact values.

2.7.9 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.

2.7.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.8 Frequency Tables

2.8.1 Introduction

A frequency table is the most basic summary of a categorical or discrete-numeric variable: for each distinct value, how many times it occurred (absolute frequency), what share of the sample it represents (relative frequency), and the running total up to and including that value (cumulative frequency). Frequency tables are the building blocks of chi-squared tests, goodness-of-fit analyses, and most epidemiological prevalence reporting.

2.8.2 Prerequisites

The reader should know what a categorical variable is and be comfortable with R factors.

2.8.3 Theory

Given a categorical variable X with levels $c_1, \ldots, c_k$ observed in a sample of size n:

- **Absolute frequency** f_j: number of times c_j appears.
- **Relative frequency** $r_j = f_j / n$: the proportion.
- **Cumulative frequency** $F_j = f_1 + \ldots + f_j$: running count. Meaningful only for ordered variables.
- **Cumulative relative frequency** $R_j = F_j / n$: running proportion, equivalent to the empirical CDF at c_j.

2.8.4 Assumptions

Frequency tables are descriptive; no distributional assumption. The variable must be categorical (nominal or ordinal) or discrete numeric.

2.8.5 R Implementation

```r
library(dplyr)
library(janitor)
library(gt)

set.seed(2026)
cohort <- tibble::tibble(
  stage = factor(sample(c("I", "II", "III", "IV"), 250, replace = TRUE,
                        prob = c(0.20, 0.35, 0.30, 0.15)),
                 levels = c("I", "II", "III", "IV"), ordered = TRUE)
)

# Base R
table(cohort$stage)
prop.table(table(cohort$stage))
cumsum(prop.table(table(cohort$stage)))

# Tidy with janitor
cohort |>
  tabyl(stage) |>
  adorn_pct_formatting(digits = 1) |>
  adorn_totals("row")

# Publication-ready via gt
cohort |> count(stage) |>
  mutate(pct = 100 * n / sum(n),
         cum_pct = cumsum(pct)) |>
  gt() |>
  fmt_number(columns = c(pct, cum_pct), decimals = 1)
```

2.8.6 Output & Results

```
 I  II III  IV
53  87  77  33

   I    II   III    IV
0.21  0.35  0.31  0.13

   I    II   III    IV
0.21  0.56  0.87  1.00
```

For the ordered stages, the cumulative column shows that 87% of patients are stage III or earlier.

2.8.7 Interpretation

Report frequency tables for every categorical variable in a manuscript's Table 1. Include both counts and percentages: "Stage III: 77 (30.8%)". For ordered variables, the cumulative column often tells the story more clearly than the raw counts.

2.8.8 Practical Tips

- Preserve factor level ordering when reporting; ordered factors in R maintain the sequence automatically.

- Report the denominator used for percentages (total sample or column total) to avoid confusion.
- For sparse levels, consider collapsing categories; report both the original and collapsed table.
- Missing values: `table()` excludes NAs by default; use `useNA = "always"` to include them.
- Publication tables benefit from `gtsummary::tbl_summary()` which handles both frequencies and continuous variables.

2.8.9 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.

2.8.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.9 The Geometric Mean

2.9.1 Introduction

The geometric mean is the arithmetic mean on the log scale, back-transformed to the original scale. It is the appropriate summary for variables that combine multiplicatively rather than additively: antibody titres, concentrations, growth rates, investment returns. For log-normal data, the geometric mean is the median and is thus a natural measure of typicality.

2.9.2 Prerequisites

The reader should know the arithmetic mean and be comfortable with natural logarithms.

2.9.3 Theory

For positive values $x_1, \ldots, x_n$:

$$\mathrm{GM}(x) = \left(\prod x_i\right)^{1/n} = \exp\left(\frac{1}{n}\sum \log x_i\right).$$

The second form is numerically stable and matches how the geometric mean is computed in practice. Key properties:

- $\mathrm{GM}(x) \leq \bar{x}$, with equality only when all x_i are equal (AM-GM inequality, a consequence of Jensen).

- For log-normal data, the geometric mean equals the population median.
- The ratio of the geometric mean to the arithmetic mean is a simple measure of skewness.

2.9.4 Assumptions

- All values must be strictly positive; a single zero or negative value renders the geometric mean undefined.
- For inference (confidence intervals), log-normality of the data is typically assumed.

2.9.5 R Implementation

```r
library(psych)

set.seed(2026)
titres ← 2^sample(4:12, 50, replace = TRUE)  # antibody titres in 2^n form

mean(titres)
median(titres)
psych::geometric.mean(titres)
exp(mean(log(titres)))

# 95% CI for the geometric mean
n ← length(titres)
log_x ← log(titres)
log_mean ← mean(log_x)
log_se   ← sd(log_x) / sqrt(n)
exp(log_mean + c(-1, 1) * qt(0.975, n - 1) * log_se)
```

2.9.6 Output & Results

```
[1] 1157         # arithmetic mean
[1] 512          # median
[1] 588          # geometric mean
[1] 588          # (equivalent computation)

[1] 441 782      # 95% CI for GM
```

The geometric mean (588) sits close to the median (512) and well below the arithmetic mean (1157). This is typical of log-normal or multiplicative data: the arithmetic mean is pulled up by the upper tail.

2.9.7 Interpretation

Report the geometric mean for titres, viral loads, drug concentrations, and any other multiplicatively combining variable. In a paper: “the geometric mean antibody titre was 588 (95% CI 441-782)”.

2.9.8 Practical Tips

- Non-positive values require imputation or exclusion; adding a small constant (e.g., 0.5) before log is common but distorts small values.

- The geometric mean of ratios is the exponent of the mean of log-ratios; reporting log-scale effects is cleaner.
- For growth rates, the geometric mean equals the arithmetic mean of $(1 + r_i)$ values minus 1.
- In meta-analysis of ratios, log transformation + normal approximation is the standard pipeline.
- When data are centred around 1 (e.g., hazard ratios), the difference between AM and GM is small.

2.9.9 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.

2.9.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.10 The Harmonic Mean

2.10.1 Introduction

The harmonic mean is the reciprocal of the arithmetic mean of reciprocals. It is appropriate when averaging rates, speeds, or ratios where the denominator is fixed – for example, when a traveller covers equal distances at different speeds and you want the average speed. In machine learning, the harmonic mean of precision and recall gives the F1 score.

2.10.2 Prerequisites

The reader should know the arithmetic mean and be comfortable taking reciprocals in R.

2.10.3 Theory

For positive values $x_1, \ldots, x_n$,

$$\mathrm{HM}(x) = \frac{n}{\sum 1/x_i}.$$

The harmonic mean is always less than or equal to the geometric mean, which is less than or equal to the arithmetic mean: HM $\leq$ GM $\leq$ AM, with equality when all x_i are equal.

The F1 score in binary classification is the harmonic mean of precision and recall:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} = \frac{1}{\frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)} \cdot (\text{rescaled}).$$

2.10.4 Assumptions

All values must be strictly positive. Negative values or zeros make the harmonic mean undefined or infinite.

2.10.5 R Implementation

```
library(psych)

speeds ← c(40, 60, 80)  # km/h over three equal distances
psych::harmonic.mean(speeds)
mean(speeds)                  # incorrect for average speed over equal distances

# F1 score
precision ← 0.80
recall    ← 0.60
f1 ← 2 * precision * recall / (precision + recall)
f1

# Harmonic mean is appropriate for average rates
n ← 3
hm_speed ← n / sum(1 / speeds)
hm_speed
```

2.10.6 Output & Results

```
[1] 55.38       # harmonic mean of speeds
[1] 60          # arithmetic mean (incorrect for equal-distance travel)
[1] 0.686       # F1 score with P=0.80, R=0.60
```

Travelling equal distances at 40, 60, 80 km/h gives an average speed of 55.4 km/h, not 60 km/h. The F1 of 0.69 splits the difference between P and R, penalising the lower value more than the arithmetic mean would.

2.10.7 Interpretation

Use the harmonic mean when averaging rates with a fixed numerator or denominator (e.g., average speed when distance is constant). Use the F1 form when balancing two normalised scores where both values matter.

2.10.8 Practical Tips

- Zero or negative inputs break the harmonic mean; guard against them.
- The harmonic mean weights small values more heavily; a single small observation drives the result down.
- In meta-analysis of rates per unit time, harmonic-mean sample size is sometimes used in effective-sample-size calculations.

- The F-beta score generalises F1: $F_\beta = (1+\beta^2)PR/(\beta^2 P + R)$, with $\beta = 2$ emphasising recall and $\beta = 0.5$ emphasising precision.
- Reporting a harmonic mean without explanation is rare; specify why it is the appropriate summary in the methods.

2.10.9 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.

2.10.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.11 The Interquartile Range

2.11.1 Introduction

The interquartile range (IQR) is the distance between the first and third quartiles, $Q_3 - Q_1$. It measures the width of the central 50% of the distribution and is robust to outliers, since extreme values do not affect the quartiles themselves. The IQR appears in boxplots, Tukey fences, and the median-IQR convention for reporting skewed biomedical variables.

2.11.2 Prerequisites

The reader should know what quantiles and quartiles are, and be comfortable with R's `quantile()` function.

2.11.3 Theory

For a sample, Q_1 and Q_3 are the 0.25 and 0.75 quantiles of the empirical CDF. The IQR is

$$\mathrm{IQR} = Q_3 - Q_1.$$

Tukey's hinges differ slightly from R's default quantiles; `fivenum()` returns the hinges and `IQR()` uses `quantile()` with `type = 7`.

For a normal distribution with SD σ, the IQR equals $2 \cdot 0.6745 \cdot \sigma \approx 1.349\sigma$. Equivalently, dividing the IQR by 1.349 gives a rough, robust estimate of the SD.

The **Tukey fences** at $Q_1 - 1.5 \cdot \mathrm{IQR}$ and $Q_3 + 1.5 \cdot \mathrm{IQR}$ define the whiskers of a standard boxplot; points beyond are flagged as outliers.

2.11.4 Assumptions

The IQR is purely descriptive and requires no distributional assumption.

2.11.5 R Implementation

```r
set.seed(2026)
x ← rnorm(80, mean = 75, sd = 12)

quantile(x, probs = c(0.25, 0.75))
IQR(x)

# Tukey fences
q1 ← quantile(x, 0.25); q3 ← quantile(x, 0.75)
fences ← c(lower = q1 - 1.5 * IQR(x),
           upper = q3 + 1.5 * IQR(x))
fences
sum(x < fences[1] | x > fences[2])

# Robust SD from IQR (approximately)
IQR(x) / 1.349
sd(x)
```

2.11.6 Output & Results

```
   25%      75%
 66.4     83.1

[1] 16.7       # IQR

lower  upper
 41.4  108.1

[1] 12.4       # SD from IQR (robust)
[1] 11.9       # sample SD
```

The two SD estimates agree closely for approximately normal data; divergence signals heavy tails or outliers.

2.11.7 Interpretation

Report the IQR alongside the median for skewed data. A typical biomedical manuscript uses “median 6.8 days (IQR 4.2-10.1 days)” to summarise length-of-stay or similar variables.

2.11.8 Practical Tips

- The boxplot visually displays the IQR as the box; whiskers extend to the Tukey fences or to the nearest non-outlier data point.
- For very small samples ($n < 10$), the IQR is unstable; report the order statistics directly.
- The IQR is the scale estimator implicit in the Tukey boxplot; it is one of the most interpretable dispersion measures for audiences less familiar with SDs.

- In grouped summaries, compare IQRs across groups alongside medians; equal medians with different IQRs indicate different spreads at the same centre.
- Do not confuse IQR (dispersion) with inter-quartile ratio (Q_3/Q_1) occasionally used in log-skewed contexts.

2.11.9 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.

2.11.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.12 Kurtosis

2.12.1 Introduction

Kurtosis is the fourth standardised moment of a distribution and quantifies tail heaviness relative to the normal. High kurtosis means a few observations contribute disproportionately to the variance (the "fat tails" of finance). Low kurtosis means the distribution has fewer extreme values than the normal. Kurtosis matters because many estimators and tests assume normal-like tails; heavy tails break asymptotic approximations and inflate Type I error.

2.12.2 Prerequisites

The reader should know what mean, SD, and skewness are, and should be comfortable computing summary statistics in R.

2.12.3 Theory

The population kurtosis is

$$\gamma_2 = E\left[\left(\frac{X-\mu}{\sigma}\right)^4\right].$$

For a standard normal distribution, $\gamma_2 = 3$. The convention is to report **excess kurtosis** $\gamma_2 - 3$, where 0 indicates normal-like tails, positive indicates heavier tails (leptokurtic), and negative indicates lighter tails (platykurtic).

Classical values:

- Normal: excess kurtosis = 0.
- Uniform: excess kurtosis = -1.2 (light tails).
- Logistic: excess kurtosis = 1.2.
- t-distribution with ν df: excess kurtosis = $6/(\nu - 4)$ for $\nu > 4$. At $\nu = 5$ the excess kurtosis is 6; at $\nu = 10$ it is 1.
- Exponential: excess kurtosis = 6.
- Laplace: excess kurtosis = 3.

2.12.4 Assumptions

Kurtosis as a descriptive statistic has no assumption. As an estimator of a population parameter, it requires finite fourth moments.

2.12.5 R Implementation

```r
library(moments)

set.seed(2026)
norm_sample ← rnorm(5000)
t4_sample   ← rt(5000, df = 4)
unif_sample ← runif(5000)

c(normal  = kurtosis(norm_sample) - 3,
  t_df4   = kurtosis(t4_sample) - 3,
  uniform = kurtosis(unif_sample) - 3)

moments::anscombe.test(t4_sample)
```

2.12.6 Output & Results

```
  normal    t_df4  uniform
   -0.04     6.12    -1.21
```

The t-distribution with 4 df has massively heavier tails than normal, as predicted. The uniform distribution has lighter tails. Anscombe's test on the t-sample gives $p < .001$, formally rejecting normal kurtosis.

2.12.7 Interpretation

Report excess kurtosis alongside skewness whenever distribution shape matters. A value near zero is "normal-like"; a value above 1 is a heavy-tailed distribution for which outlier-sensitive methods (mean, SD, OLS) may perform poorly.

2.12.8 Practical Tips

- Kurtosis is sensitive to outliers; a single extreme value can inflate it dramatically.
- Formal normality tests (Shapiro-Wilk, Jarque-Bera) implicitly test kurtosis and skewness jointly.

- Heavy tails call for robust methods: median, MAD, robust regression, rank-based tests.
- Leptokurtic distributions sometimes motivate using a Student-t rather than normal likelihood in regression (see `brms::student`).
- Different software sometimes report kurtosis with/without the "-3" correction; always check whether excess or raw kurtosis is being reported.

2.12.9 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.

2.12.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.13 Mean Absolute Deviation

2.13.1 Introduction

The mean absolute deviation (MAD) is the average distance of observations from a measure of central tendency. It uses absolute values instead of squares, making it less sensitive to outliers than the standard deviation. In the modern literature, the term "MAD" almost always means the **median absolute deviation from the median**, a robust scale estimator.

2.13.2 Prerequisites

The reader should know what a mean, median, and SD are, and how to compute them in R.

2.13.3 Theory

Two versions exist.

MAD about the mean: $\mathrm{MAD}_{\text{mean}} = \frac{1}{n}\sum |x_i - \bar{x}|$. More stable than the SD under heavy-tailed distributions but still sensitive to outliers because it uses the mean as the centre.

MAD about the median (modern default): $\mathrm{MAD} = \text{median}|x_i - \text{median}(x)|$. Highly robust; breakdown point of 50%. When multiplied by 1.4826, it is a consistent estimator of the SD for normal data.

R's `mad()` function computes MAD about the median with the 1.4826 scaling constant by default.

2.13.4 Assumptions

MAD is purely descriptive; no distributional assumption.

2.13.5 R Implementation

```r
set.seed(2026)
x ← c(rnorm(50, mean = 100, sd = 10), 500)  # one outlier

sd(x)
mad(x, constant = 1.4826)
mad(x, constant = 1)
mean(abs(x - mean(x)))
```

2.13.6 Output & Results

```
[1] 56.4      # SD inflated by outlier
[1] 9.78      # MAD estimator of SD (robust)
[1] 6.60      # raw MAD (unscaled)
[1] 22.4      # MAD about the mean (less robust)
```

The classical SD is nearly six times the robust MAD estimator because a single extreme point at 500 dominates the sum of squares. The MAD about the median is essentially unaffected.

2.13.7 Interpretation

Report MAD when the data contain outliers or when robustness to contamination is desirable. In biomedical chemistry, MAD is the default measure of analytical imprecision for certain platforms.

2.13.8 Practical Tips

- `mad()` defaults: `constant = 1.4826`, `center = median`. Always check whether this is what you want.
- For a normal sample, MAD $\times$ 1.4826 approximates the SD; the two agree closely at moderate n.
- The MAD is a robust estimator but not robust to the estimator of the *centre*: use the median, not the mean, as the centre.
- For positive-skewed data (lab values), a raw MAD on the original scale can still be informative; log-transform the data first if the skew is severe.
- Do not confuse the MAD with the variance; they are in the same units as the data (unlike the variance, which is squared units).

2.13.9 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.

2.13.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.14 Measures of Dispersion

2.14.1 Introduction

A measure of central tendency tells you where the data are centred; a measure of dispersion tells you how far they spread around that centre. Two samples can share the same mean yet behave differently in every downstream calculation because their variability is different. This tutorial covers the five dispersion measures used routinely in applied work, explains when each is appropriate, and shows how to compute them in R.

2.14.2 Prerequisites

The reader should know what the mean and median are, and should be comfortable computing them on an R vector.

2.14.3 Theory

For a sample $x_1, \ldots, x_n$:

- **Range** $= \max x_i - \min x_i$. Simplest, most fragile: uses only two values, is dominated by outliers, has no information about the bulk of the distribution.
- **Variance** $s^2 = \frac{1}{n-1}\sum(x_i - \bar{x})^2$. The average squared deviation from the mean. Additive over independent sums. Divisor $n-1$ gives an unbiased estimator of the population variance; divisor n gives the MLE under normality (biased).
- **Standard deviation** $s = \sqrt{s^2}$. Same units as the data. The SD of a normally distributed variable corresponds to the 68-95-99.7 rule.
- **Median absolute deviation (MAD)** $= \text{median}|x_i - \text{median}(x)|$. Robust to outliers; breakdown point of 50%. Multiplied by 1.4826 to estimate the SD of a normal distribution.
- **Interquartile range (IQR)** $= Q_3 - Q_1$. The width of the central 50% of the data. Used in the Tukey boxplot fence for outlier identification ($\pm 1.5 \cdot \text{IQR}$).

The SD is the default for approximately normal data. For skewed or contaminated data, the MAD and IQR are far more stable. Reporting the range is informative only for small, clean samples or for summarising audit ranges.

2.14.4 Assumptions

All dispersion measures are purely descriptive; they do not require any distributional assumption. Their *interpretation*, however, depends on the distribution: an SD of 10 on a normal distribution implies 68% of values within ± 10 of the mean, while an SD of 10 on a heavy-tailed distribution does not.

2.14.5 R Implementation

```r
library(dplyr)
library(robustbase)

set.seed(2026)

df ← tibble::tibble(
  symmetric    = rnorm(200, mean = 50, sd = 10),
  right_skewed = rlnorm(200, meanlog = log(50), sdlog = 0.6)
)

summarise_dispersion ← function(x) {
  tibble::tibble(
    n        = length(x),
    range    = max(x) - min(x),
    variance = var(x),
    sd       = sd(x),
    iqr      = IQR(x),
    mad      = mad(x, constant = 1.4826),
    mad_raw  = mad(x, constant = 1)
  )
}

dispersion_tbl ← df ▷
  tidyr::pivot_longer(everything(), names_to = "variable", values_to = "value") ▷
  group_by(variable) ▷
  summarise(
    summarise_dispersion(value),
    .groups = "drop"
  )

dispersion_tbl

Qn_scale ← robustbase::Qn(df$right_skewed)
Sn_scale ← robustbase::Sn(df$right_skewed)
c(Qn = Qn_scale, Sn = Sn_scale)
```

The `mad()` function defaults to a scaling constant of 1.4826, which makes it consistent with the SD for normal data. `robustbase::Qn()` and `Sn()` are alternative robust scale estimators with higher efficiency than MAD.

2.14.6 Output & Results

Typical output:

variable	n	range	variance	sd	iqr	mad
symmetric	200	57.3	98.5	9.93	13.2	10.1

variable	n	range	variance	sd	iqr	mad
right_skewed	200	260.2	1143.2	33.8	34.5	23.7

For the symmetric normal sample, SD and MAD are nearly identical as expected under normality. For the right-skewed log-normal sample, the SD (33.8) is larger than the MAD (23.7), because the SD is pulled up by the right tail.

2.14.7 Interpretation

Report dispersion appropriate to the distribution shape:

- **Approximately normal data**: mean (SD).
- **Skewed or contaminated data**: median (IQR) or median (MAD).
- Never report mean (SD) for a distribution whose histogram is obviously asymmetric; the summary gives a misleading impression of typicality.

2.14.8 Practical Tips

- Plot the histogram before choosing the summary; numerical dispersion alone can disguise bimodality or heavy tails.
- For small samples, IQR is preferred over range: a single extreme value dominates the range.
- The SD and variance are measured in the original units squared (variance) or original units (SD); do not mix them in a single summary.
- When reporting a coefficient of variation (CV = SD/mean), ensure the mean is far from zero; CV is meaningless for variables centred near zero.
- In R, `sd()` uses the $n - 1$ divisor; the MLE version is `sqrt(mean((x - mean(x))^2))`.

2.14.9 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.

2.14.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.15 Measures of Central Tendency

2.15.1 Introduction

A measure of central tendency is a single number that summarises "where the data are". The three classical choices – arithmetic mean, median, and mode – are not interchangeable: each answers a slightly different question, and each is appropriate in a different context. Choosing the right one matters because readers of your results will form mental pictures of the data that the number implies.

2.15.2 Prerequisites

The reader should know the difference between a numeric, an ordered categorical, and a nominal categorical variable, and should be able to read a histogram. Familiarity with R vectors is assumed.

2.15.3 Theory

The **arithmetic mean** of observations $x_1, \ldots, x_n$ is $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. It is the centre of mass of the data and minimises the sum of squared deviations from any candidate centre. The mean uses all observations, but it is sensitive to outliers: a single extreme value can drag it arbitrarily far from the bulk of the data.

The **median** is the value that separates the lower half of the data from the upper half: $x_{((n+1)/2)}$ for odd n, the average of the two middle values for even n. The median minimises the sum of absolute deviations. Unlike the mean, it is robust to outliers: moving the largest observation to infinity leaves the median unchanged.

The **mode** is the most frequently occurring value. For continuous data, the mode is typically defined from a kernel density estimate rather than from the raw values, because ties are unlikely. The mode is the only measure of central tendency that makes sense for nominal categorical variables.

The **trimmed mean** computes the mean after removing a proportion α of the smallest and largest observations – a compromise between the efficiency of the mean and the robustness of the median. A 10% trimmed mean removes the bottom and top 10% of the observations; a 50% trimmed mean is the median.

2.15.4 Assumptions

These are descriptive statistics, not tests, so they do not carry distributional assumptions. They do carry interpretive assumptions: the mean is most useful when the distribution is roughly symmetric; the median is preferred for skewed distributions and for ordinal data; the mode is preferred for categorical data and for multimodal distributions where a single centre is misleading.

2.15.5 R Implementation

```r
library(DescTools)
library(psych)

x ← c(2.1, 2.4, 2.5, 2.7, 2.9, 3.1, 3.3, 3.4, 3.7, 42.0)

mean(x)
median(x)
mean(x, trim = 0.1)
Mode(x)
psych::geometric.mean(x)
psych::harmonic.mean(x)
```

The vector `x` is a small sample with a single outlier at 42. Running the above shows a mean of 6.71, a median of 3.00, and a 10% trimmed mean of 2.94. The geometric mean (3.44) and harmonic mean (3.15) are close to the median rather than the mean, since both give less weight to the outlier.

2.15.6 Output & Results

For a skewed or contaminated distribution, the three classical measures of central tendency can differ substantially:

Statistic	Value
Mean	6.71
Median	3.00
10% trimmed mean	2.94
Geometric mean	3.44
Harmonic mean	3.15

The mean is pulled far from the bulk of the data by the single outlier; every other measure stays close to the main cluster.

2.15.7 Interpretation

For a manuscript, the choice of statistic should be justified in the methods section. A paper reporting the “mean serum creatinine” in a cohort where several dialysis patients are included should either exclude those patients explicitly or report the median with IQR. Never mix conventions across papers in a series: if you reported medians in one table you should not swap to means in the next.

2.15.8 Practical Tips

- Plot the data before choosing the statistic. A histogram or boxplot will show immediately whether the mean is a sensible summary.
- Report the median and IQR (not the mean and SD) for variables with strong right skew: lab values, income, incubation times.

- For small samples ($n < 10$), any central tendency measure is uncertain; consider bootstrapping to get a confidence interval for the statistic of choice.
- Do not report the mode for continuous data unless it comes from a density estimate – the sample mode of continuous data is the lowest value in the tied set, which is meaningless.

2.15.9 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.

2.15.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.16 Measures of Shape

2.16.1 Introduction

Central tendency and dispersion alone do not describe a distribution completely: two samples with the same mean and SD can look very different if one is symmetric and the other skewed, or if one has thin tails and the other heavy tails. Skewness and kurtosis are the conventional third- and fourth-moment summaries that capture these shape features.

2.16.2 Prerequisites

The reader should know what the mean, variance, and standard deviation are, and should understand what a histogram shows.

2.16.3 Theory

The **skewness** of a distribution measures its asymmetry:

$$g_1 = \frac{1}{n} \sum \left(\frac{x_i - \bar{x}}{s} \right)^3 .$$

Positive skewness indicates a right-tailed distribution (mean > median); negative skewness indicates a left tail. For a normal distribution, $g_1 = 0$. Absolute values greater than about 1 are considered substantial.

The **kurtosis** measures tail heaviness:

$$g_2 = \frac{1}{n}\sum\left(\frac{x_i - \bar{x}}{s}\right)^4.$$

For a normal, $g_2 = 3$. By convention, **excess kurtosis** is reported as $g_2 - 3$: zero means normal-like tails, positive means heavier (leptokurtic), negative means lighter (platykurtic). High kurtosis signals that a few observations contribute a large share of the variance; in finance this is the “fat tails” that break many standard models.

Both statistics are moment-based and therefore sensitive to outliers. In small samples, their estimates are noisy.

2.16.4 Assumptions

Skewness and kurtosis are descriptive and require no distributional assumption. Their asymptotic standard errors assume independent observations with finite sixth moment for skewness and eighth for kurtosis.

2.16.5 R Implementation

```r
library(moments)
library(psych)

set.seed(2026)

symmetric   <- rnorm(5000, mean = 0, sd = 1)
right_skewed <- rlnorm(5000, meanlog = 0, sdlog = 0.6)
heavy_tailed <- rt(5000, df = 4)

shape_summary <- function(x, label) {
  data.frame(
    variable = label,
    skewness = moments::skewness(x),
    kurtosis = moments::kurtosis(x),
    excess_kurtosis = moments::kurtosis(x) - 3
  )
}

rbind(
  shape_summary(symmetric,    "Normal"),
  shape_summary(right_skewed, "Log-normal"),
  shape_summary(heavy_tailed, "t, df=4")
)

psych::describe(data.frame(symmetric, right_skewed, heavy_tailed),
                skew = TRUE, ranges = FALSE)
```

2.16.6 Output & Results

variable	skewness	kurtosis	excess kurtosis
Normal	0.01	3.02	0.02
Log-normal	1.95	9.96	6.96
t, df=4	-0.02	7.41	4.41

The normal sample has skewness and excess kurtosis near zero. The log-normal is strongly right-skewed and heavy-tailed. The t-distribution with 4 df is symmetric but very heavy-tailed.

2.16.7 Interpretation

Report shape statistics alongside location and dispersion for any distribution whose histogram is not obviously normal. In a methods section: "Serum CRP was right-skewed (skewness = 1.8, excess kurtosis = 3.2), so log-transformation was applied before parametric testing."

2.16.8 Practical Tips

- Skewness and kurtosis are sensitive to outliers; a single extreme point can drive g_1 past 1.
- In small samples ($n < 30$), estimates are noisy; do not over-interpret small deviations from zero.
- For normality assessment, prefer graphical methods (Q-Q plots) over numerical skewness and kurtosis thresholds.
- Different software use slightly different bias corrections; `moments::skewness` uses the g_1 form, `psych::describe` uses the G_1 form with a small-sample correction.
- Log transformation often reduces both skewness and excess kurtosis in right-skewed biomedical data.

2.16.9 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.

2.16.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.17 Outlier Detection Rules

2.17.1 Introduction

Outlier detection is a diagnostic step, not a cleaning step: flagged points should be investigated, not automatically removed. Several rules produce flags from different premises; each has a different trade-off between sensitivity and robustness. This page covers the common univariate rules and their assumptions.

2.17.2 Prerequisites

The reader should know what mean, median, SD, and IQR are.

2.17.3 Theory

- **1.5 x IQR fence (Tukey)**: observations outside $[Q_1 - 1.5 \cdot \mathrm{IQR},\ Q_3 + 1.5 \cdot \mathrm{IQR}]$ are mild outliers; $3 \cdot \mathrm{IQR}$ marks extremes. Robust and distribution-free.
- **Z-score (3-sigma)**: $|z_i| > 3$ where $z_i = (x_i - \bar{x})/s$. Sensitive to the outliers it is trying to flag (masking).
- **Hampel identifier**: $|x_i - \mathrm{median}| > 3 \cdot \mathrm{MAD}$. Robust variant of the z-score that uses median and MAD.
- **Grubbs' test**: formal test of the most extreme value under normality. Gives a p-value but is sensitive to its own normality assumption.

Masking and swamping are two pitfalls: masking occurs when true outliers inflate the SD and hide themselves; swamping occurs when one outlier drags neighbours across the threshold, falsely flagging them.

2.17.4 Assumptions

- IQR fence and Hampel: no distributional assumption; robust.
- Z-score: approximately normal distribution.
- Grubbs' test: normality and a single suspected outlier.

2.17.5 R Implementation

```r
library(outliers)

set.seed(2026)
x ← c(rnorm(48, mean = 50, sd = 8), 95, 110)

# IQR fence
q ← quantile(x, c(0.25, 0.75))
iqr ← IQR(x)
fences ← c(q[1] - 1.5 * iqr, q[2] + 1.5 * iqr)
which(x < fences[1] | x > fences[2])

# Z-score
z ← (x - mean(x)) / sd(x)
which(abs(z) > 3)

# Hampel
h ← (x - median(x)) / mad(x)
which(abs(h) > 3)

# Grubbs' test for the single most extreme
grubbs.test(x)
```

2.17.6 Output & Results

```
IQR fence flags:   [49, 50]
z-score flags:     [50]
Hampel flags:      [49, 50]
Grubbs: G = 2.81, p = 0.03
```

The IQR and Hampel rules identify both extreme values; the z-score, inflated by those same values, identifies only the most extreme. Grubbs' test rejects the null that the extreme point is from the same distribution.

2.17.7 Interpretation

Report which rule you applied and how many observations it flagged. "Two observations exceeded the Tukey 1.5 x IQR upper fence; both were verified as accurate data-entry values and retained in the primary analysis" is a defensible sentence.

2.17.8 Practical Tips

- Flag, investigate, decide. Never delete silently.
- For skewed data, IQR fences still flag normal-range observations as outliers; consider log-transformation first.
- For multivariate outliers, use Mahalanobis distance; for regression diagnostics, use Cook's distance.
- Grubbs' test assumes normality; for non-normal data prefer the IQR or Hampel rule.
- Report sensitivity analyses that re-run the primary analysis with and without flagged observations; the conclusion should be robust.

2.17.9 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.

2.17.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.18 Percentile Ranks

2.18.1 Introduction

A percentile rank answers "what fraction of observations fall at or below this value?" It converts an absolute measurement to a relative position in the sample distribution. In paediatrics, growth charts give a child's weight percentile; in cognitive testing, an IQ score

comes with a percentile rank; in bioinformatics, a gene's expression is often reported as a percentile across a reference panel.

2.18.2 Prerequisites

The reader should know what a quantile is and be comfortable with sorted vectors in R.

2.18.3 Theory

For a sample $x_1, \dots, x_n$, the percentile rank of a value y is

$$\mathrm{PR}(y) = 100 \cdot \frac{\#\{x_i \le y\}}{n}.$$

Several refinements exist: the "mid-rank" form counts half of any ties; the "plotting position" form uses $(i - a)/(n + 1 - 2a)$ for various a (e.g., Blom's $a = 3/8$).

For a single value, computing the percentile rank is the inverse of computing a quantile: if the 75th percentile is $q_{0.75}$, then $\mathrm{PR}(q_{0.75}) \approx 75$.

2.18.4 Assumptions

Purely descriptive; assumes only that the reference distribution is appropriate to the value being ranked.

2.18.5 R Implementation

```
library(dplyr)

set.seed(2026)
growth_chart <- rnorm(1000, mean = 15.0, sd = 2.2)   # reference weights (kg)
patient      <- 17.8                                   # a patient's weight

pr_patient <- 100 * mean(growth_chart <= patient)
pr_patient

# Percentile ranks for all observations in a cohort
cohort <- data.frame(weight = rnorm(30, 15.5, 2.0))
cohort$rank    <- rank(cohort$weight)
cohort$pr      <- 100 * cohort$rank / nrow(cohort)
cohort$pr_mid  <- 100 * (rank(cohort$weight) - 0.5) / nrow(cohort)
head(cohort)

# ECDF equivalent
ec <- ecdf(growth_chart)
100 * ec(patient)
```

2.18.6 Output & Results

```
[1] 89.4     # 17.8 kg is at the 89th percentile of the reference

  weight rank    pr pr_mid
1  13.29    1   3.3   1.67
2  14.15    6  20.0  18.33
3  15.01   11  36.7  35.00
```

2.18.7 Interpretation

Report percentile ranks in growth, development, and norming contexts. In a clinical note: “body weight 17.8 kg (89th percentile for age/sex)”. The percentile tells the reader how unusual the value is relative to the reference.

2.18.8 Practical Tips

- Percentile ranks depend on the reference sample; always state which reference was used (national norms, internal cohort, etc.).
- Ties are handled by `rank()` via averaging, minimum, or maximum; specify `ties.method` when the convention matters.
- For small reference samples, percentile ranks are unstable; use smoothing (LMS method) for growth chart applications.
- A percentile rank of exactly 100 is usually not reported; “above the 99th percentile” is the conventional phrasing for the sample maximum.
- Do not confuse percentile rank (position in a distribution) with percentile (value at a specified position).

2.18.9 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.

2.18.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.19 Quantiles and Percentiles

2.19.1 Introduction

A quantile is a cut-point that divides a distribution by cumulative probability: the p-quantile is the value below which a proportion p of the data lies. The median is the 0.5-quantile; quartiles are the 0.25, 0.50, and 0.75 quantiles; percentiles are quantiles rescaled to the 0-

100 range. Quantiles underlie boxplots, five-number summaries, reference ranges in lab medicine, and non-parametric confidence intervals.

2.19.2 Prerequisites

The reader should know what a cumulative distribution function is and be comfortable with sorted vectors in R.

2.19.3 Theory

For a sample $x_1, \ldots, x_n$ sorted as $x_{(1)} \le x_{(2)} \le \ldots \le x_{(n)}$, the sample p-quantile is not uniquely defined when $p \cdot (n+1)$ is not an integer. R offers nine definitions, selected via the `type` argument of `quantile()`:

- `type = 1`: inverse of the empirical CDF, giving a step-function quantile.
- `type = 4`: linear interpolation of the empirical CDF (SAS default).
- `type = 6`: linear interpolation based on plotting positions (Minitab, SPSS).
- `type = 7`: R's default; linear interpolation between order statistics with no offset.
- `type = 8`: approximately unbiased for continuous distributions (Hyndman-Fan recommendation).

The different types give slightly different numeric values in small samples; they converge as n grows. Type 7 is the default in R's `quantile()`, but reproducibility across software requires specifying the type.

2.19.4 Assumptions

Quantiles are purely descriptive; they require no distributional assumption.

2.19.5 R Implementation

```r
library(dplyr)

set.seed(2026)
x ← rnorm(60, mean = 75, sd = 10)

# Default quantile types
quantile(x, probs = c(0.25, 0.50, 0.75))

# Compare five R types
sapply(c(1, 4, 6, 7, 8), function(t) quantile(x, probs = 0.25, type = t))

# Five-number summary via fivenum
fivenum(x)

# Deciles (10th, 20th, ..., 90th percentiles)
quantile(x, probs = seq(0.1, 0.9, by = 0.1))

# Quantiles by group
diabetes ← tibble::tibble(
  arm = factor(rep(c("Placebo", "Active"), each = 40)),
  hba1c = c(rnorm(40, 7.4, 0.6), rnorm(40, 6.8, 0.7))
)
diabetes ▷ group_by(arm) ▷
  summarise(q25 = quantile(hba1c, 0.25),
            q50 = quantile(hba1c, 0.50),
            q75 = quantile(hba1c, 0.75))
```

2.19.6 Output & Results

```
25% 50% 75%
71.2 74.8 80.6

   1    4    6    7    8
72.3 71.8 71.0 71.2 71.3
```

The five types differ by less than 2 units at $n = 60$. For `diabetes`, the active arm's quartiles are shifted lower than placebo's, as expected.

2.19.7 Interpretation

Report quantiles when the distribution is skewed, ordinal, or otherwise not well-summarised by a mean: median with IQR (25th and 75th percentiles) is the standard form. In reference-range contexts (clinical chemistry), the 2.5 and 97.5 percentiles define the central 95% "normal range".

2.19.8 Practical Tips

- Always specify `type` in `quantile()` when reproducing results across software.
- The boxplot uses Type 2 (`fivenum()`) for its hinges; this differs from Type 7 quantiles.
- Quantiles respect monotonic transformations: `quantile(log(x), 0.5) == log(quantile(x, 0.5))` (exactly for the median, approximately for others depending

on `type`).
- For very small samples ($n < 10$), quantile estimates are unstable; report the order statistics directly instead.
- Empirical quantiles are biased estimators of population quantiles; use Harrell-Davis (`Hmisc::hdquantile`) for a smoother estimator.

2.19.9 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.

2.19.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.20 Robust Statistics: An Overview

2.20.1 Introduction

A robust statistic is one that remains useful when the data deviate from the idealised model – heavy tails, outliers, skewed contamination. Classical estimators like the mean and SD are efficient under normality but break down under even modest contamination. Robust alternatives sacrifice a little efficiency for dramatic stability, and are the default in many applied fields where data quality cannot be guaranteed.

2.20.2 Prerequisites

The reader should know the mean, median, SD, and MAD. Familiarity with outlier concepts helps.

2.20.3 Theory

Two key concepts from robustness theory:

Breakdown point. The largest fraction of contaminated observations an estimator can tolerate before giving an arbitrary answer. The arithmetic mean has breakdown 0: a single extreme point can push it arbitrarily far. The median has breakdown 0.5: up to half the sample can be contaminated before the median is affected.

Influence function. Measures the infinitesimal effect of adding a single observation at x on the estimator. The arithmetic mean has a linear, unbounded influence function; the median has a bounded (step-shaped) influence function.

Robust estimators aim for high breakdown point and bounded influence function, typically with moderate loss of efficiency at the normal model.

Robust alternatives to classical summaries:

Classical	Robust	Breakdown	Efficiency at normal
Mean	Median	0.5	64%
Mean	20% trimmed mean	0.2	~95%
Mean	Huber M-estimator	0	~95%
SD	MAD	0.5	37%
SD	Qn, Sn	0.5	~82%
OLS	Robust regression (`rlm` , `lmrob`)	0 to 0.5	95%

2.20.4 Assumptions

Robustness is an explicit relaxation of distributional assumptions; it asks how the estimator behaves under deviation, not whether deviation is present.

2.20.5 R Implementation

```r
library(robustbase)
library(MASS)

set.seed(2026)
clean ← rnorm(100, mean = 50, sd = 10)
contaminated ← c(clean[1:95], rnorm(5, mean = 200, sd = 5))

classical ← c(mean = mean(contaminated), sd = sd(contaminated))
robust    ← c(median = median(contaminated),
              mad    = mad(contaminated),
              Qn     = Qn(contaminated),
              Sn     = Sn(contaminated))

rbind(classical = c(classical, NA, NA), robust)

# Robust regression example
n ← 80
x ← rnorm(n)
y ← 2 + 1.5 * x + rnorm(n)
y[c(5, 10, 15)] ← y[c(5, 10, 15)] + c(15, -15, 20)  # inject outliers

coef(lm(y ~ x))
coef(rlm(y ~ x))          # M-estimator
coef(lmrob(y ~ x))        # MM-estimator with high breakdown
```

2.20.6 Output & Results

```
      mean    sd   median  mad    Qn    Sn
57.84   32.9   50.28  10.21  10.45 10.72
```

Classical mean is pulled to 57.8 by the five extreme values; every robust estimator remains near the true 50.

For the regression: OLS slope is 1.23 (pulled by outliers); `rlm` gives 1.47 (close to truth of 1.5); `lmrob` gives 1.48.

2.20.7 Interpretation

Use robust statistics as the default in biomedical work where outliers or heavy tails are typical. A robust summary plus a classical one can be reported side by side; divergence between them signals contamination that warrants investigation.

2.20.8 Practical Tips

- MM-estimators (`lmrob`) combine high breakdown with high efficiency and are the modern robust-regression default.
- Robust methods are not a license to ignore outliers; always investigate before reporting.
- For heavy-tailed data, a Student-t likelihood (`brms` or `heavy` package) is an alternative to robust M-estimators.
- Robust confidence intervals are usually narrower than a non-robust CI on contaminated data, reflecting the estimator's stability.
- Specialised packages: `robustbase`, `WRS2` (Wilcox), `quantreg` (quantile regression), `MASS::rlm`.

2.20.9 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.

2.20.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.21 Skewness

2.21.1 Introduction

Skewness is the third standardised moment of a distribution; it quantifies how asymmetric the distribution is around its mean. Zero skewness corresponds to perfect symmetry; positive skewness indicates a long right tail; negative skewness indicates a long left tail. Many biomedical variables (lab values, incubation times, survival times) are right-skewed; income and wealth distributions are classically right-skewed with very long tails.

2.21.2 Prerequisites

The reader should know what a mean and SD are, and should have seen a few histograms.

2.21.3 Theory

The population skewness is

$$\gamma_1 = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right].$$

Several sample estimators exist with different small-sample biases. The classical moment estimator is $g_1 = m_3/m_2^{3/2}$ where $m_k = \frac{1}{n}\sum(x_i-\bar{x})^k$. SAS/SPSS/`psych` use a bias-corrected form $G_1 = \sqrt{n(n-1)}/(n-2)\cdot g_1$. Interpretation: $|g_1| < 0.5$ is approximately symmetric, $0.5 < |g_1| < 1$ is moderately skewed, $|g_1| > 1$ is highly skewed.

2.21.4 Assumptions

As a descriptive statistic, skewness has no assumptions. As an estimator, it requires finite third moments, which fails for heavy-tailed distributions like the Cauchy.

2.21.5 R Implementation

```
library(moments)

set.seed(2026)
sym   <- rnorm(2000, 0, 1)
rskew <- rlnorm(2000, meanlog = 0, sdlog = 0.7)
lskew <- -rlnorm(2000, meanlog = 0, sdlog = 0.7)

c(symmetric = skewness(sym),
  right     = skewness(rskew),
  left      = skewness(lskew))

# log transform reduces right skewness
skewness(log(rskew))
```

2.21.6 Output & Results

```
 symmetric      right       left
     0.013      2.134     -2.090
```

Log-transforming the right-skewed sample drops skewness from 2.1 to approximately 0; this is the standard remedy for biomedical lab values.

2.21.7 Interpretation

Report skewness for any variable whose histogram departs from symmetry. When $|g_1| > 1$, the mean is pulled away from the centre of mass; the median is a better point summary, or the variable should be transformed.

2.21.8 Practical Tips

- Log transformation is the default remedy for right-skewed positive data.
- Visual inspection beats numerical thresholds; a histogram makes the shape immediately visible.
- Skewness in small samples is noisy; avoid thresholds below $n = 30$.
- The Pearson skewness coefficient $3(\text{mean} - \text{median})/\text{SD}$ is a robust alternative.
- For formal tests of skewness, use D'Agostino's test (`moments::agostino.test`).

2.21.9 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.

2.21.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.22 Standardisation and z-Scores

2.22.1 Introduction

A z-score expresses each observation as its distance from the mean in units of the standard deviation. Standardisation is often applied before clustering, PCA, ridge/lasso regression, and any procedure whose output is unit-sensitive. For data with outliers, a robust variant using the median and MAD is safer.

2.22.2 Prerequisites

The reader should know what the mean, SD, median, and MAD are.

2.22.3 Theory

The classical z-score for observation x_i is

$$z_i = (x_i - \bar{x})/s.$$

After standardisation, the sample mean is 0 and the sample SD is 1. The values are dimensionless and comparable across variables measured on different scales.

The **robust z** uses the median and scaled MAD:

$$z_i^{\text{robust}} = (x_i - \text{median}(x))/(1.4826 \cdot \text{MAD}(x)).$$

It reduces to the classical z for normal data but resists the inflation of the scale estimator by outliers.

Standardisation is not the same as normalisation. **Min-max normalisation** rescales to $[0, 1]$: $(x_i - \min(x))/(\max(x) - \min(x))$. Each has its use; z-scores preserve relative distances better.

2.22.4 Assumptions

Purely algebraic; no distributional assumption. The choice between classical and robust z depends on whether outliers are present.

2.22.5 R Implementation

```r
library(robustbase)

set.seed(2026)
x ← c(rnorm(50, mean = 100, sd = 10), 200, 250)

z_classical ← scale(x)
z_robust    ← (x - median(x)) / (1.4826 * mad(x, constant = 1))

range(z_classical)
range(z_robust)

cohort ← data.frame(
  age  = rnorm(100, 58, 12),
  sbp  = rnorm(100, 130, 18),
  crp  = rlnorm(100, log(3), 0.6)
)
z_cohort ← scale(cohort)
colMeans(z_cohort)
apply(z_cohort, 2, sd)
```

2.22.6 Output & Results

```
range(z_classical): -1.69   6.11
range(z_robust):    -2.02  15.04
```

The robust z flags the outlier at 250 as > 15 SDs from the median, while the classical z only gives 6. Classical z understates the deviation because the extreme point has inflated the sample SD.

2.22.7 Interpretation

Report z-scores when comparing observations on different scales. Biological examples: centred and scaled gene-expression values across samples; cognitive-test scores scaled to a reference population.

2.22.8 Practical Tips

- `scale()` returns a matrix; wrap with `as.data.frame()` when using downstream tidyverse functions.
- Save the `center` and `scale` attributes to apply the same standardisation to new test data – critical in predictive modelling.
- For skewed variables, log-transform before standardising.
- The robust z is a simple and effective first-pass outlier filter: $|z^{\text{robust}}| > 3$ flags candidates.
- Do not standardise the outcome in a regression if interpretability is desired; standardise predictors if comparing coefficients.

2.22.9 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.

2.22.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.23 Stem-and-Leaf Plots

2.23.1 Introduction

The stem-and-leaf plot, introduced by John Tukey, is a compact display that shows the distribution of a small sample while preserving every data value. Each observation is split into a "stem" (leading digits) and a "leaf" (trailing digit); stems are listed vertically and leaves are written next to each stem in order. For samples up to about 100 observations, the resulting ASCII plot is a poor man's histogram that can be typed into a notebook.

2.23.2 Prerequisites

The reader should know what a histogram is.

2.23.3 Theory

Given observations, pick a stem width (10s, 100s, etc.) and list each observation by its stem and leaf. For example, values 12, 15, 18, 22, 25, 29, 34 with tens-digit stems:

```
1 | 2 5 8
2 | 2 5 9
3 | 4
```

The back-to-back variant puts two groups on either side of the stem, permitting visual comparison. Modern software rarely uses stem-and-leaf for publication figures, but it remains a quick inspection tool in R's console.

2.23.4 Assumptions

Purely descriptive, requires nothing except numeric data.

2.23.5 R Implementation

```r
set.seed(2026)
x ← round(rnorm(40, mean = 100, sd = 15))
stem(x)

# With scale change
stem(x, scale = 0.5)
stem(x, scale = 2)

# Back-to-back
library(aplpack)
x ← rnorm(30, 100, 15)
y ← rnorm(30, 110, 12)
stem.leaf.backback(x, y)
```

2.23.6 Output & Results

```
The decimal point is 1 digit(s) to the right of the |

 7 | 3
 8 | 0447
 9 | 01135789
10 | 0111345899
11 | 0245578
12 | 34
```

Each leaf on the 10-row represents one observation: 101, 101, 101, 103, 104, etc. The display is itself a histogram with the same-height bars rotated 90 degrees.

2.23.7 Interpretation

Use stem-and-leaf plots for quick eyeballing in R's console; they reveal outliers, skew, and multimodality without generating a figure file.

2.23.8 Practical Tips

- For samples over 100, the plot becomes cluttered; use a histogram.
- `scale` in `stem()` adjusts the stem width; experiment to find the right granularity.
- Back-to-back plots are a useful low-overhead alternative to side-by-side boxplots.
- Stem-and-leaf is rarely used in journal articles today; it is a Tukey EDA tradition, not a reporting convention.

- In teaching contexts, it remains a useful way to introduce distribution shape before histograms.

2.23.9 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.

2.23.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.24 Summary by Group

2.24.1 Introduction

Split-apply-combine is the workhorse pattern of applied data analysis: split the data by a grouping variable, apply a summary function to each group, and combine the results. It produces the per-group means, medians, counts, and other statistics that populate Table 1 of most scientific papers.

2.24.2 Prerequisites

The reader should know what the mean, median, and SD are, and be comfortable with R's `dplyr` pipe syntax.

2.24.3 Theory

For a dataset with n observations and a grouping variable g taking values $g_1, \ldots, g_k$, split-apply-combine computes

$$s_j = f(\{x_i : g_i = g_j\})$$

for each group j, where f is the summary function (mean, median, quantile, any aggregator). The returned object has k rows, one per group.

R implements this via base `aggregate()`, `tapply()`, and `by()`, and via the modern `dplyr::group_by() ▷ summarise()` and `data.table` equivalents.

2.24.4 Assumptions

Purely descriptive; no distributional assumption. The grouping variable should have enough observations per group for the summary to be meaningful (a mean based on $n = 2$ is noisy).

2.24.5 R Implementation

```r
library(dplyr)
library(data.table)
library(gtsummary)

set.seed(2026)
trial <- tibble::tibble(
  arm      = factor(rep(c("Placebo", "Active"), each = 75)),
  sex      = factor(sample(c("F", "M"), 150, replace = TRUE)),
  age      = round(rnorm(150, 58, 12)),
  hba1c    = round(c(rnorm(75, 7.3, 0.7), rnorm(75, 6.9, 0.8)), 1),
  event    = rbinom(150, 1, 0.15)
)

# dplyr
trial |>
  group_by(arm) |>
  summarise(
    n         = n(),
    age_mean  = mean(age),
    age_sd    = sd(age),
    hba1c_med = median(hba1c),
    hba1c_iqr = IQR(hba1c),
    event_rate = mean(event)
  )

# Multi-variable grouping
trial |>
  group_by(arm, sex) |>
  summarise(across(c(age, hba1c), list(mean = mean, sd = sd)),
            .groups = "drop")

# data.table
dt <- as.data.table(trial)
dt[, .(n = .N, age_mean = mean(age), hba1c_median = median(hba1c)),
   by = .(arm)]

# Publication-ready
trial |>
  tbl_summary(by = arm, include = c(age, sex, hba1c, event),
              statistic = list(all_continuous() ~ "{mean} ({sd})"))
```

2.24.6 Output & Results

```
# A tibble: 2 x 6
  arm       n age_mean age_sd hba1c_med hba1c_iqr event_rate
  <fct> <int>    <dbl>  <dbl>     <dbl>     <dbl>      <dbl>
1 Placebo  75     57.9   12.1       7.3       0.9     0.173
2 Active   75     57.8   11.8       6.9       1.1     0.120
```

The two arms differ in HbA1c (medians 7.3 vs. 6.9) while being comparable in age and sex distribution – the typical pattern in a Table 1 balance check.

2.24.7 Interpretation

Report grouped summaries for every important subgrouping in a study: treatment arms, sex, study site, risk stratum. The `gtsummary::tbl_summary(by = ...)` call produces a fully formatted Table 1 with p-values on request.

2.24.8 Practical Tips

- Always include `n` per group so readers know the denominator.
- Handle missing values explicitly: `summarise(mean(x, na.rm = TRUE))` or pre-filter.
- For many summary statistics per variable, `across(everything(), list(mean = mean, sd = sd))` is concise.
- `data.table` is faster for very large datasets; the syntax differs but the logic is identical.
- Report median + IQR for skewed variables and mean + SD for approximately normal ones; mix the two when the Table 1 covers both types.

2.24.9 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.

2.24.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.25 Trimmed and Winsorised Means

2.25.1 Introduction

A trimmed mean drops a proportion of observations from each tail before computing the mean. A Winsorised mean replaces those observations with the nearest non-trimmed values and then averages. Both estimators compromise between the arithmetic mean (efficient but sensitive to outliers) and the median (robust but inefficient). They are useful when the data are heavy-tailed but you want to retain most of the sample's information.

2.25.2 Prerequisites

The reader should know what the arithmetic mean and the median are.

2.25.3 Theory

For a sample of size n and a trim proportion $\alpha \in [0, 0.5)$:

- **Trimmed mean**: sort the data, discard the smallest and largest $\lfloor n\alpha \rfloor$ observations, and average the rest.
- **Winsorised mean**: replace the smallest $\lfloor n\alpha \rfloor$ values with the $(\lfloor n\alpha \rfloor + 1)$-th smallest, and similarly at the upper end; then compute the mean.

Special cases: $\alpha = 0$ gives the arithmetic mean; $\alpha = 0.5$ gives the median (exactly for odd n, almost for even n). A 10% or 20% trim is a common compromise.

Trimmed means have a positive breakdown point (proportional to α): a trim of 20% tolerates up to 20% contamination without being affected. Winsorised means have the same breakdown but retain the influence of the replaced values through their "pinning" behaviour.

2.25.4 Assumptions

These estimators are descriptive; no distributional assumption. Their *efficiency* depends on the distribution: the 20% trimmed mean is nearly as efficient as the arithmetic mean under normality but much more robust under heavy tails.

2.25.5 R Implementation

```r
library(psych)
library(DescTools)

set.seed(2026)
x ← c(rnorm(48, mean = 50, sd = 8), 200, 250)  # two outliers

mean(x)
median(x)
mean(x, trim = 0.10)
mean(x, trim = 0.20)
DescTools::Winsorize(x, probs = c(0.10, 0.90)) ▷ mean()
```

2.25.6 Output & Results

```
[1] 56.62       # arithmetic mean (pulled up by outliers)
[1] 50.30       # median
[1] 51.82       # 10% trimmed mean
[1] 50.78       # 20% trimmed mean
[1] 51.50       # 10% Winsorised mean
```

The trimmed and Winsorised means align closely with the median, demonstrating their robustness. The arithmetic mean is dragged up by the two extreme points.

2.25.7 Interpretation

Report a trimmed or Winsorised mean when outliers are present, their exclusion is justified, and you want a more efficient summary than the median. Explicitly state the trim proportion.

2.25.8 Practical Tips

- The 10% trim is a useful default; 20% is stronger but loses efficiency.
- Don't let the trim serve as a clean-up for genuine heterogeneity; investigate the outliers first.
- Inference for trimmed means requires a Yuen test (`WRS2::yuen`) rather than the usual t-test.
- Winsorising is preferred when you want to retain the sample size but soften the influence of extremes.
- Combined with robust scale (MAD, Sn), the trimmed mean / SD pair gives a fully robust location / spread summary.

2.25.9 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.

2.25.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.26 Variance and Standard Deviation

2.26.1 Introduction

Variance and standard deviation are the most widely used dispersion measures in applied statistics. The variance is the mean squared deviation from the mean; the SD is its square root, returning to the original units. Both underlie the normal distribution, the central limit theorem, regression, and every parametric test that refers to "sigma".

2.26.2 Prerequisites

The reader should know the mean and be comfortable squaring, summing, and taking square roots in R.

2.26.3 Theory

For a sample $x_1, \ldots, x_n$:

- **Population variance** (when the data *are* the whole population): $\sigma^2 = \frac{1}{n}\sum(x_i - \mu)^2$.
- **Sample variance** (when the data are a sample): $s^2 = \frac{1}{n-1}\sum(x_i - \bar{x})^2$.

The $n - 1$ divisor (Bessel's correction) makes s^2 an unbiased estimator of σ^2. Using n would systematically underestimate σ^2 because $\bar{x}$ is itself estimated from the same data.

The sample **standard deviation** is $s = \sqrt{s^2}$. Even though s^2 is unbiased, s is biased downward because the square root is concave (Jensen's inequality). The bias shrinks like $1/n$ and is negligible for $n > 30$; an unbiased estimator exists (`c_4` correction in SPC) but is rarely used in applied work.

R's `var()` and `sd()` use the $n - 1$ divisor by default. To get the MLE version, compute `mean((x - mean(x))^2)`.

2.26.4 Assumptions

Variance and SD are descriptive; no distributional assumption is required. For inferential use (the standard error of the mean is $s/\sqrt{n}$), the usual iid assumption applies.

2.26.5 R Implementation

```
set.seed(2026)
x ← rnorm(60, mean = 100, sd = 15)

var(x)
sd(x)

# MLE (biased) version with divisor n
mean((x - mean(x))^2)
sqrt(mean((x - mean(x))^2))

# Bias of s: theoretical factor c_4(n) for normal data
c4 ← function(n) sqrt(2 / (n - 1)) * gamma(n / 2) / gamma((n - 1) / 2)
sd_unbiased ← sd(x) / c4(length(x))
sd_unbiased
```

2.26.6 Output & Results

```
[1] 210.5       # s^2 with divisor n-1
[1]  14.5       # s
[1] 207.0       # MLE variance
[1]  14.4       # MLE SD (biased)
[1]  14.6       # bias-corrected SD
```

At $n = 60$, the difference between biased and unbiased SD is a fraction of a percent, irrelevant for applied work.

2.26.7 Interpretation

Report variance when the scale of the squared units is interpretable (e.g., variance of portfolio returns). Report SD when the scale of the original units is desired. In a paper: "mean 100.2 (SD 14.5)" is universally readable.

2.26.8 Practical Tips

- Always specify which divisor you mean when reporting; R's default is $n - 1$, which is the statistician's convention.

- For very small samples, the biased-vs-unbiased distinction matters; for most applied work (n > 30) it does not.
- The SD is in the same units as the data; the variance is in units squared. Don't mix them in a single report.
- When a variance is reported without an associated mean, check whether it is actually a standard error (which is smaller by $\sqrt{n}$).
- For skewed data, the SD is not a useful summary; use the IQR instead.

2.26.9 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.

2.26.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.27 The Weighted Mean

2.27.1 Introduction

A weighted mean is a generalisation of the arithmetic mean in which each observation carries a positive weight that reflects its importance, reliability, or inclusion probability. Weighted means appear in survey sampling (weights are reciprocals of inclusion probabilities), in meta-analysis (weights are inverse-variance), and in weighted regression.

2.27.2 Prerequisites

The reader should know the arithmetic mean and be comfortable with the R function `weighted.mean()`.

2.27.3 Theory

Given observations $x_1, \ldots, x_n$ with positive weights $w_1, \ldots, w_n$, the weighted mean is

$$\bar{x}_w = \frac{\sum w_i x_i}{\sum w_i}.$$

When all weights are equal, this reduces to the arithmetic mean. The weights re-scale each observation's contribution. In inverse-variance weighting, $w_i = 1/s_i^2$, which gives more precise observations more influence.

The weighted variance is

$$s_w^2 = \frac{\sum w_i (x_i - \bar{x}_w)^2}{\sum w_i - 1}$$

(using the frequency-weight convention). For survey weights, a different denominator is used.

2.27.4 Assumptions

- Weights must be positive.
- The choice of weight reflects a real asymmetry (inclusion probability, precision, repetition count); arbitrary weighting is meaningless.

2.27.5 R Implementation

```
library(survey)

# Meta-analysis: four studies with mean and SE
studies ← data.frame(
  mean = c(1.20, 0.95, 1.40, 1.10),
  se   = c(0.15, 0.25, 0.10, 0.20)
)
studies$w ← 1 / studies$se^2

pooled_mean ← weighted.mean(studies$mean, studies$w)
pooled_se   ← sqrt(1 / sum(studies$w))
c(pooled_mean = pooled_mean, pooled_se = pooled_se)

# Survey weights: 200 respondents, sample weight = 1 / inclusion probability
set.seed(2026)
resp ← data.frame(
  sbp = rnorm(200, 132, 15),
  w   = sample(c(0.5, 1.0, 2.0), 200, replace = TRUE)
)
weighted.mean(resp$sbp, resp$w)
```

2.27.6 Output & Results

```
 pooled_mean    pooled_se
       1.236        0.075
```

The inverse-variance weighted mean is 1.24 with SE 0.08 – more precise than the simple average of 1.16 that would ignore the study-level precision.

2.27.7 Interpretation

Meta-analytic pooling gives study-level weights that reflect how much each study contributes to the combined evidence. Survey weights make the computed mean estimate the population mean rather than the sample mean. Always report which weights were used.

2.27.8 Practical Tips

- Negative or zero weights are invalid; check the input.
- For survey data, use the `survey` package (`svymean`, `svydesign`) rather than `weighted.mean()` to get the correct SE accounting for design.
- Inverse-variance weighting is optimal under random-effects meta-analysis only when τ^2 is zero; otherwise weights should use $w_i = 1/(s_i^2 + \tau^2)$.
- When weights are frequencies (integer counts), the weighted mean equals the ordinary mean of the expanded dataset.
- Reporting "the weighted mean" without specifying the weights is ambiguous; include the weight definition in methods.

2.27.9 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.

2.27.10 See also — labs in this chapter

- Descriptive statistics and Table 1

2.28 Week 2, Session 1 — Descriptive statistics and Table 1

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

2.28.1 Learning objectives

- Match the right summary statistic to the right variable type — means for symmetric continuous, medians for skewed, counts and percentages for categorical.
- Build a publication-ready Table 1 with `gtsummary::tbl_summary()`.
- Defend the choice of summary with a quick visual check.

2.28.2 Prerequisites

Labs 1.3 through 1.5.

2.28.3 Background

Table 1 is the first real table of almost every clinical paper. It describes the sample: who was in the study, in how many groups, with what characteristics, and — if the study has arms — how balanced those arms are at baseline. A Table 1 that is built thoughtlessly hides an unbalanced comparison; a Table 1 that is built well anchors the rest of the paper.

The right summary depends on the variable. A continuous variable that is roughly symmetric is summarised by its mean and standard deviation; one that is skewed or has heavy tails is summarised by its median and interquartile range. A categorical variable is summarised by counts and percentages. Reporting the median of a balanced continuous variable is not wrong, but it is unusual; reporting the mean of a skewed variable is often misleading.

gtsummary packages these choices into a declarative API. You say which variables to include and which grouping variable to stratify by, and the table is built with sensible defaults. When you need non-standard behaviour, you override per-variable.

2.28.4 Setup

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

2.28.5 1. Goal

Produce a Table 1 for the `penguins` dataset stratified by species, and back the choice of summary with a diagnostic plot.

2.28.6 2. Approach

```
dat ← penguins ▷
  drop_na(species, sex, bill_length_mm, bill_depth_mm,
          flipper_length_mm, body_mass_g)
```

2.28.7 3. Execution

Pick the right summary per variable. First, eyeball the shapes.

```
dat ▷
  pivot_longer(c(bill_length_mm, bill_depth_mm,
                 flipper_length_mm, body_mass_g),
               names_to = "variable", values_to = "value") ▷
  ggplot(aes(value, fill = species)) +
  geom_histogram(bins = 20, alpha = 0.6, colour = "grey30",
                 position = "identity") +
  facet_wrap(~ variable, scales = "free") +
  labs(x = NULL, y = "Count")
```

All four continuous variables are approximately symmetric within species, so mean (SD) is defensible. If a variable were skewed, we would switch to median (IQR).

```
t1 ← dat ▷
  select(species, sex, bill_length_mm, bill_depth_mm,
         flipper_length_mm, body_mass_g) ▷
  tbl_summary(
    by = species,
    statistic = list(
      all_continuous() ~ "{mean} ({sd})",
      all_categorical() ~ "{n} ({p}%)"
    ),
    digits = all_continuous() ~ 1,
    label = list(
      bill_length_mm    ~ "Bill length (mm)",
      bill_depth_mm     ~ "Bill depth (mm)",
      flipper_length_mm ~ "Flipper length (mm)",
      body_mass_g       ~ "Body mass (g)",
      sex               ~ "Sex"
    )
  ) ▷
  add_overall() ▷
  add_p()
t1
```

`add_p()` will run a t-test or ANOVA on continuous variables and a chi-square or Fisher's exact test on categorical, with sensible defaults for sample size.

2.28.8 4. Check

Validate the gtsummary output against a hand computation.

```
dat ▷
  group_by(species) ▷
  summarise(
    n = n(),
    mean_mass = mean(body_mass_g),
    sd_mass   = sd(body_mass_g),
    .groups = "drop"
  )
```

The group means and standard deviations must match the numbers in the body-mass row of the Table 1.

2.28.9 5. Report

> Table 1 describes `nrow(dat)` penguins from three species in the Palmer Archipelago. Body mass was highest in Gentoo (`round(mean(dat$body_mass_g[dat$species == "Gentoo"]), 0)` ± `round(sd(dat$body_mass_g[dat$species == "Gentoo"]), 0)` g) and lower in Adelie and Chinstrap. All four continuous variables were approximately symmetric within species and are summarised as mean (SD); sex was approximately balanced within each species.

Table 1 is not a tool for hypothesis testing. The *p*-values in the rightmost column describe the data, not the biology. A statistically-significant difference between arms at baseline in a randomised trial is, by construction, a chance finding.

2.28.10 Common pitfalls

- Reporting a mean for a skewed variable.
- Using Table 1 as a place to "prove balance" with *p*-values in a randomised trial.
- Suppressing the missing-data row. Always report it.
- Forgetting that `tbl_summary` drops NAs by default; check the count at the bottom of each column.

2.28.11 Further reading

- Sjoberg DD et al. *Reproducible summary tables with the gtsummary package*. R Journal.
- Altman DG. *Practical Statistics for Medical Research*, chapter on describing data.

2.28.12 Session info

```
sessionInfo()
```

2.28.13 See also — chapter index

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