14 Clinical Biostatistics

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

Diagnostic test accuracy, agreement (Bland-Altman, kappa, ICC), biomarker development, prediction-model reporting under TRIPOD-AI, and fairness audits. The chapter sits at the intersection of ML and biostatistics for a reason: that is where most regulatory submissions live.

This chapter contains **36 method pages** and **4 labs**. If you are not sure which method to read, return to Chapter 0 and follow the decision tree to the right node.

14.1 Method pages

Method	Source slug
Adaptive Trial Designs	`adaptive-design`
Alpha-Spending Functions	`alpha-spending`
Baseline Adjustment with ANCOVA	`baseline-adjustment-ancova`
Bland-Altman Limits of Agreement	`bland-altman-analysis`
Blinding Procedures	`blinding-procedures`
Block Randomisation	`block-randomization`
Clinical Equivalence Trials	`equivalence-clinical`
Cluster-Randomised Trials	`cluster-rct`
Cohen’s Kappa	`cohens-kappa`
Conditional Power	`conditional-power`
Crossover RCT Design	`rct-design-crossover`
Cutpoint Selection	`cutpoint-selection`
Diagnostic Test Accuracy	`diagnostic-accuracy`
Factorial Trials	`factorial-trial`
Interim Analyses and Group Sequential Designs	`interim-analysis-group-sequential`
Intraclass Correlation Coefficient (ICC)	`icc-continuous-agreement`
ITT vs Per-Protocol Analysis	`itt-vs-pp-analysis`
Likelihood Ratios	`likelihood-ratios`
Minimisation Algorithm	`minimization-algorithm`
Missing Data in RCTs	`missing-data-rct`
Multiple Imputation	`multiple-imputation`
Non-Inferiority Margin Selection	`non-inferiority-margin`
O’Brien-Fleming Boundary	`obrien-fleming-boundary`
Parallel-Group RCT Design	`rct-design-parallel`

Method	Source slug
Pocock Boundary	`pocock-boundary`
Predictive Values and Prevalence	`predictive-values-prevalence`
Randomisation Methods	`randomization-methods`
Reliability and Cronbach's Alpha	`reliability-cronbach-alpha`
ROC Analysis	`roc-analysis`
Sample Size Re-Estimation	`sample-size-reestimation`
Sensitivity Analyses in Clinical Trials	`sensitivity-analysis-clinical`
Stepped-Wedge Trial	`stepped-wedge-trial`
Stratified Randomisation	`stratified-randomization`
Subgroup Analyses	`subgroup-analysis`
Subgroup Forest Plots	`forest-plot-subgroup`
Weighted Kappa	`weighted-kappa`

14.2 Labs

Lab
Diagnostic testing: Se, Sp, PPV, NPV, LR
Kappa, ICC, Bland–Altman
Biomarker statistics (Youden, NRI, decision curves)
TRIPOD-AI, fairness auditing, reproducibility at scale

14.3 Adaptive Trial Designs

14.3.1 Introduction

Adaptive designs allow pre-specified modifications – sample size, randomisation ratios, treatment-arm selection – based on data accrued during the trial. They promise efficiency but require careful statistical control to preserve Type I error. The FDA and EMA both provide detailed guidance on acceptable adaptive modifications.

14.3.2 Prerequisites

Group-sequential trials; alpha spending; sample-size calculation.

14.3.3 Theory

Common adaptive features: - **Sample-size re-estimation** (blinded or unblinded). - **Early stopping** for efficacy or futility. - **Arm selection** (drop inferior arms in multi-arm trials). - **Response-adaptive randomisation** (shift allocation toward effective arms). - **Population**

enrichment (restrict to a responsive subgroup).

Maintaining Type I error under pre-specified adaptations requires methods like group-sequential boundaries, combination tests, or conditional error functions.

14.3.4 Assumptions

Adaptation rules are fully pre-specified (protocol, SAP); unblinded information access is tightly controlled (Data Monitoring Committee); adjustments are statistically valid.

14.3.5 R Implementation

```r
library(rpact)

# Group-sequential design with O'Brien-Fleming boundary, 3 analyses
design ← getDesignGroupSequential(
  sided = 2, alpha = 0.025, beta = 0.2,
  typeOfDesign = "OF",
  informationRates = c(0.5, 0.75, 1)
)
kable_summary ← summary(design)
print(design)

# Plan sample size for a two-arm trial with continuous outcome
ssr ← getSampleSizeMeans(design = design,
                         alternative = 0.3, stDev = 1)
print(ssr)
```

14.3.6 Output & Results

Group-sequential boundaries and associated sample sizes per stage; cumulative Type I error preserved at alpha.

14.3.7 Interpretation

“The adaptive design applied O’Brien-Fleming boundaries at 50 %, 75 %, and 100 % information; stage-1 interim inefficacy boundary would stop at p > 0.0002, preserving overall alpha = 0.025.”

14.3.8 Practical Tips

- Pre-specify every adaptation (including the decision rule) in the protocol.
- FDA/EMA require a detailed justification of the adaptive feature and its operating characteristics.
- Independent data monitoring committee is essential for efficacy or futility stopping.
- Simulation is often used to verify operating characteristics; report them.
- Post-hoc adaptations (“seamless” trial extensions) are exploratory, not confirmatory.

14.3.9 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.

14.3.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.4 Alpha-Spending Functions

14.4.1 Introduction

Alpha-spending functions (Lan & DeMets, 1983) allow interim analyses at unscheduled times while preserving overall Type I error. The spending function $f(t)$ specifies the cumulative alpha budget at information fraction $t \in [0, 1]$; nominal alpha at each analysis is the increment $f(t_k) - f(t_{k-1})$.

14.4.2 Prerequisites

Group-sequential designs; information fraction.

14.4.3 Theory

Common spending functions: - **OF-type**: $f(t) = 2 - 2\Phi(z_{\alpha/2}/\sqrt{t})$. Approximates the OF boundary. - **Pocock-type**: $f(t) = \alpha \log(1 + (e-1)t)$. Approximates Pocock. - **Power family**: $f(t) = \alpha t^{\rho}$ with $\rho > 0$. - **Custom**: any non-decreasing f with $f(0) = 0, f(1) = \alpha$.

Flexibility: interim analyses can occur at arbitrary information fractions, re-solving the boundary each time.

14.4.4 Assumptions

Information fractions are known (approximately); stopping rule applied as specified; test statistic is normal.

14.4.5 R Implementation

```r
library(rpact)

# OF spending, flexible information fractions
design ← getDesignGroupSequential(
  sided = 1, alpha = 0.025, beta = 0.2,
  typeOfDesign = "asOF",          # alpha-spending OF-type
  informationRates = c(0.4, 0.75, 1)
)

print(design$stageLevels)
plot(design, type = 1)

# Compare spending functions
f_of  ← getDesignGroupSequential(sided = 1, alpha = 0.025,
                                 typeOfDesign = "asOF",
                                 informationRates = seq(0.1, 1, 0.1))
f_poc ← getDesignGroupSequential(sided = 1, alpha = 0.025,
                                 typeOfDesign = "asP",
                                 informationRates = seq(0.1, 1, 0.1))
cbind(OF = cumsum(f_of$stageLevels),
      Pocock = cumsum(f_poc$stageLevels))
```

14.4.6 Output & Results

Cumulative alpha at each information fraction for both spending families; OF delays alpha consumption, Pocock distributes it earlier.

14.4.7 Interpretation

“The alpha-spending design at information fractions 0.4, 0.75, 1.0 under OF-type spending allocated cumulative alpha of 0.001, 0.013, 0.025 respectively, enabling flexibility in interim timing without inflating Type I error.”

14.4.8 Practical Tips

- Alpha spending is the **standard** for modern confirmatory group-sequential trials.
- Re-estimate information fractions at each interim if accrual deviates from plan.
- Information is usually subject-count for continuous outcomes, event-count for time-to-event.
- Never spend more alpha than the planned cumulative function at the current information fraction.
- For futility, use separate **beta-spending** functions (non-binding boundaries are standard).

14.4.9 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.

14.4.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.5 Baseline Adjustment with ANCOVA

14.5.1 Introduction

Analysis of Covariance (ANCOVA) regresses the outcome on treatment and baseline value jointly. Compared to a naive change-from-baseline t-test, ANCOVA gains precision, handles regression to the mean correctly, and reduces SE roughly by a factor of $\sqrt{1-\rho^2}$ where ρ is baseline-outcome correlation.

14.5.2 Prerequisites

Linear regression; correlation; change scores.

14.5.3 Theory

Naive change-score analysis: compare $\Delta = Y_{\text{post}} - Y_{\text{baseline}}$ across arms. Unbiased under randomisation but less efficient than ANCOVA when baseline correlates with outcome.

ANCOVA: $Y_{\text{post}} = \alpha + \beta_{\text{trt}} \cdot T + \beta_{\text{base}} \cdot Y_{\text{baseline}} + \varepsilon$. Treatment effect β_{trt} has lower SE than the change-score test.

Regression to the mean: if groups have different baselines by chance, change scores bias toward the difference; ANCOVA corrects for this.

14.5.4 Assumptions

Linear relationship between baseline and outcome; no treatment-by-baseline interaction (common extension: stratify or add interaction term).

14.5.5 R Implementation

```r
set.seed(2026)
n <- 200
baseline <- rnorm(n, 10, 2)
arm      <- factor(rep(c("ctrl", "trt"), each = n/2))

# Outcome correlated with baseline; true trt effect = 1
outcome <- 0.7 * baseline + ifelse(arm == "trt", 1, 0) +
           rnorm(n, 0, 1)

# Naive change-score analysis
change <- outcome - baseline
t.test(change ~ arm)

# ANCOVA
fit <- lm(outcome ~ arm + baseline)
summary(fit)$coefficients

# SE comparison
sd(change[arm == "trt"] - mean(change[arm == "trt"]))
```

14.5.6 Output & Results

ANCOVA estimates the treatment effect with substantially smaller SE than the change-score t-test when baseline-outcome correlation is non-trivial.

14.5.7 Interpretation

“ANCOVA estimated the treatment effect as 1.02 (95 % CI 0.74-1.30, p < 0.001) with ~40 % lower SE than the change-score analysis, leveraging the 0.7 baseline-outcome correlation.”

14.5.8 Practical Tips

- Use ANCOVA for any continuous outcome where a baseline measurement is available.
- Even with small baseline-outcome correlation (0.3), ANCOVA improves power.
- The “change from baseline” as an outcome is a special case of ANCOVA with $\beta_{\text{base}} = 1$ forced; ANCOVA with free β is preferred.
- Pre-specify baseline adjustment in the SAP; post-hoc addition risks bias.
- EMA guidance on baseline adjustment: stratification and ANCOVA are both acceptable; ANCOVA is more efficient when baseline is continuous.

14.5.9 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.

14.5.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.6 Bland-Altman Limits of Agreement

14.6.1 Introduction

The Bland-Altman plot (1986) compares two measurement methods by plotting their difference against their mean. It exposes systematic bias, proportional bias, and the 95 % limits of agreement within which most differences fall. It is the standard graphical summary for method-comparison studies and has largely replaced correlation-based summaries.

14.6.2 Prerequisites

Paired measurements; method-comparison basics.

14.6.3 Theory

For paired measurements (A_i, B_i): - **Bias**: mean difference $\bar{d} = \overline{A_i - B_i}$. - **Limits of agreement**: $\bar{d} \pm 1.96 \cdot s_d$.

If 95 % of differences lie within the limits and the limits are clinically acceptable, the two methods can be used interchangeably. Proportional bias shows as a trend in the scatter.

14.6.4 Assumptions

Differences are approximately normal; differences do not systematically depend on the mean (check with regression); replicates are handled appropriately if present.

14.6.5 R Implementation

```r
library(ggplot2)

set.seed(2026)
n <- 100
truth <- rnorm(n, 10, 2)
A <- truth + rnorm(n, 0, 0.5)            # method A
B <- truth + 0.3 + rnorm(n, 0, 0.5)      # method B (small bias)

df <- data.frame(A, B,
                 mean = (A + B) / 2,
                 diff = A - B)

bias <- mean(df$diff)
sd_d <- sd(df$diff)
loa  <- c(bias - 1.96 * sd_d, bias + 1.96 * sd_d)

ggplot(df, aes(mean, diff)) +
  geom_point(colour = "#2A9D8F") +
  geom_hline(yintercept = bias, linetype = 1) +
  geom_hline(yintercept = loa, linetype = 2) +
  labs(x = "Mean of A and B", y = "A - B",
       title = "Bland-Altman plot",
       subtitle = sprintf("Bias %.2f; 95%% LoA [%.2f, %.2f]",
                          bias, loa[1], loa[2])) +
  theme_minimal()
```

14.6.6 Output & Results

A scatter of differences vs means with solid bias line and dashed LoA; the simulated systematic bias of -0.3 is recovered.

14.6.7 Interpretation

“Bland-Altman analysis revealed a bias of -0.3 units with 95 % limits of agreement (-1.7, 1.1). If the clinically acceptable limit is +-2 units, the methods are interchangeable for most practical purposes.”

14.6.8 Practical Tips

- Always show both bias and LoA; correlation alone does not reveal systematic bias.
- Check for proportional bias by regressing differences on means; a non-zero slope indicates non-constant bias.
- For replicated measurements per subject, adjust the LoA calculation (Bland-Altman 1999 extension).
- Report the clinical acceptance criterion **before** calculating LoA; otherwise post-hoc thresholding biases conclusions.
- Paired with ICC, Bland-Altman gives both a quantitative and visual summary of agreement.

14.6.9 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.

14.6.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.7 Blinding Procedures

14.7.1 Introduction

Blinding (masking) keeps trial participants and personnel unaware of treatment assignments to minimise performance, assessment, and analyst biases. Single, double, triple, and quadruple blinding refer to which stakeholder groups are masked; each layer blocks a specific bias pathway.

14.7.2 Prerequisites

RCT design; sources of bias in clinical research.

14.7.3 Theory

- **Single blinding**: participant unaware; investigator aware.
- **Double blinding**: participant and investigator both unaware. Standard for drug vs placebo.
- **Triple blinding**: adds blinded outcome assessors (PROBE designs invert this).
- **Quadruple blinding**: adds blinded statisticians / analysts.

Each additional layer addresses bias but also increases operational complexity.

14.7.4 Assumptions

Identical appearance, taste, and packaging of active and placebo; emergency unblinding procedures are in place.

14.7.5 R Implementation

Blinding is operational – not a statistical analysis per se – but the **success of blinding** should be audited.

```r
# Simulate an end-of-study blinding-integrity questionnaire
set.seed(2026)
n ← 200
guess ← factor(sample(c("active", "placebo", "don't know"),
                      n, replace = TRUE,
                      prob = c(0.4, 0.35, 0.25)))
true  ← factor(rep(c("active", "placebo"), each = n/2))

# James blinding index (range 0-1; 0.5 = good blinding)
tab ← table(guess, true)
n ← sum(tab)
# Simpler chi-square test of correct-guess rate
correct ← sum(guess == true)
binom.test(correct, length(guess), p = 0.5)
```

14.7.6 Output & Results

Binomial test of whether the correct-guess rate exceeds chance; p > 0.05 consistent with effective blinding.

14.7.7 Interpretation

“End-of-study unblinding revealed 56 % correct guesses (95 % CI 49-62 %, p = 0.12 vs chance), consistent with successful blinding. The study reports the result per CONSORT recommendations.”

14.7.8 Practical Tips

- Match active and placebo precisely (taste, colour, packaging, schedule); any difference leaks information.
- For difficult-to-blind interventions (surgery, behavioural), blind at least outcome assessment.
- Test blinding at study end (e.g., James/Bang blinding index); report the result.
- Pre-specify emergency unblinding procedures; document all unblinding events.
- A statistician blinded to allocation prevents analysis-choice bias even in open-label trials.

14.7.9 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.

14.7.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)

- TRIPOD-AI, fairness auditing, reproducibility at scale

14.8 Block Randomisation

14.8.1 Introduction

Block randomisation allocates clinical-trial participants in random permutations within fixed-size blocks, guaranteeing equal numbers of subjects in each treatment arm at every block boundary. It is the standard alternative to simple randomisation in clinical trials because it prevents the substantial arm-size imbalance that simple randomisation can produce in small trials, in early enrolment phases, or at any moment when the trial is paused for an interim analysis. Block randomisation is now mandated or strongly recommended by the ICH-E9 statistical-principles guideline, by CONSORT for randomisation reporting, and by virtually every regulatory authority's clinical-trial guidance.

14.8.2 Prerequisites

A working understanding of randomisation as the foundation of causal inference in trials, allocation concealment as the procedural safeguard against selection bias, and the difference between simple, blocked, and stratified randomisation.

14.8.3 Theory

With block size B and two equally-allocated arms, each block contains $B/2$ assignments of each arm in random order. At every block boundary the arm counts are exactly balanced; between boundaries the maximum imbalance is $B/2$. With fixed block size, an investigator who knows the block size and observes $B-1$ allocations within a block can predict the final allocation — a serious threat in open-label trials. **Variable block sizes** (mixing, e.g., blocks of 4 and 6) defeat this predictability while preserving the boundary-balance guarantee.

For multi-arm trials with k arms in equal allocation, blocks must be multiples of k; for unequal allocation (e.g., $2:1$), blocks are multiples of the sum of allocation ratios.

14.8.4 Assumptions

Allocation concealment is preserved (the randomisation list is prepared in advance, kept off-site, and never available to enrolling investigators), the block-size distribution is documented in the statistical analysis plan but withheld from those who could exploit it, and randomisation is implemented through an interactive web-response system (IWRS) or equivalent with audit trail.

14.8.5 R Implementation

```
library(blockrand)

set.seed(2026)

fix ← blockrand(n = 40, num.levels = 2,
                levels = c("A", "B"),
                block.sizes = 2)
table(fix$treatment)
head(fix, 8)

var_b ← blockrand(n = 40, num.levels = 2,
                  levels = c("A", "B"),
                  block.sizes = c(2, 3))
head(var_b, 10)
```

14.8.6 Output & Results

`blockrand()` returns an allocation schedule with exactly equal arm counts across the requested n participants. With variable block sizes, the schedule blends blocks of different lengths in random order, preventing investigators from predicting the next allocation late in any single block. The schedule is typically exported to an IWRS and made available only to the trial pharmacist or unblinded statistician.

14.8.7 Interpretation

A reporting sentence: “Treatment allocation used block randomisation with variable block sizes of 4 and 6, generated by `blockrand` and managed via the trial’s interactive web-response system. Allocation was stratified by site and disease severity, with blocks nested within strata. The block-size distribution was documented in the SAP and concealed from enrolling investigators throughout the trial. Final arm counts were balanced (200 in each arm of the 400-patient trial).” Always report block-size distribution and concealment procedure.

14.8.8 Practical Tips

- Avoid fixed block size alone in open-label trials; variable block sizes are now the de facto standard for randomised clinical trials and are explicitly recommended by ICH-E9 because they prevent end-of-block predictability without sacrificing balance.
- Document the block-size distribution in the SAP but withhold the actual block sequence from enrolling investigators; sharing the block sequence (even informally) compromises allocation concealment and is a recurring cause of CONSORT-cited methodological flaws.
- For stratified designs (by site, disease severity, age category), nest blocks within strata so that each stratum maintains independent arm balance; this is standard practice in multicentre trials and prevents centre-by-treatment confounding.
- Very large blocks reduce guessability further but also relax the boundary-balance guarantee at any given moment in enrolment; small to moderate blocks (2 to 6 in two-arm trials) are the standard compromise and adequately balance most trials.

- Commercial randomisation services (IWRS / IRT) manage the list, preserve concealment, and provide a tamper-proof audit trail; for sponsor-led trials the cost is justified by the regulatory protection.
- For trials with more than two arms, use stratified blocked randomisation with appropriately sized blocks; permuted-block randomisation extends naturally to any number of arms with equal or unequal allocation ratios.

14.8.9 R Packages Used

`blockrand` for canonical fixed and variable-block randomisation with built-in stratification support; `randomizr` for tidyverse-friendly randomisation including blocked and stratified designs; `bcrm` for biased-coin and minimisation alternatives; `ldhmm` and `psborrow` for adaptive randomisation in more complex designs; `Mediana` for trial-design simulation including randomisation strategies.

14.8.10 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.

14.8.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.9 Cluster-Randomised Trials

14.9.1 Introduction

Cluster-randomised trials (CRTs) randomise groups – clinics, schools, villages – rather than individuals. Used when an intervention must be delivered at cluster level (implementation, educational campaign) or when contamination between individuals would bias a standard RCT. Clustering inflates variance and must be accounted for in sample-size and analysis.

14.9.2 Prerequisites

Intraclass correlation (ICC); mixed-effects models.

14.9.3 Theory

Design effect: $DE = 1 + (m - 1)\rho$, where m is average cluster size and ρ is the ICC. Effective sample size = actual N / DE. Sample-size calculations inflate by DE relative to individually-randomised trials.

Analysis accounts for clustering via mixed-effects models (random cluster intercept) or GEE with cluster-robust SE.

14.9.4 Assumptions

Clusters are exchangeable; intervention is applied uniformly within cluster; ICC estimate from pilot / literature is approximately correct.

14.9.5 R Implementation

```r
library(lme4); library(lmerTest)

set.seed(2026)
# 20 clusters, avg 15 patients per cluster
n_clusters <- 20
m_per <- 15
cluster <- factor(rep(1:n_clusters, each = m_per))
arm     <- factor(rep(c("ctrl", "trt"), each = (n_clusters/2) * m_per))
clust_re <- rep(rnorm(n_clusters, 0, 0.8), each = m_per)

y <- clust_re + ifelse(arm == "trt", 0.5, 0) +
     rnorm(n_clusters * m_per, 0, 1)

df <- data.frame(cluster, arm, y)
fit <- lmer(y ~ arm + (1 | cluster), data = df)
summary(fit)$coefficients

# Empirical ICC
vc <- as.data.frame(VarCorr(fit))
icc <- vc$vcov[1] / sum(vc$vcov)
cat("Estimated ICC:", round(icc, 3), "\n")
```

14.9.6 Output & Results

Cluster-random-effect-adjusted treatment effect (~0.5) with SE accounting for clustering; ICC estimate ~0.4.

14.9.7 Interpretation

“Cluster-randomised analysis estimated a 0.49 SD improvement (95 % CI 0.21-0.77, p = 0.001), accounting for the intra-cluster correlation of 0.42 via a random-cluster intercept.”

14.9.8 Practical Tips

- Even a small ICC (0.01) inflates required sample size substantially; budget accordingly.

- Pilot data or literature usually provides ICC; report both the planning value and the observed value.
- For few clusters (< 30), mixed-effects SE underestimates; use Kenward-Roger or Satterthwaite DF.
- Report per CONSORT extension for cluster trials; include number of clusters, cluster sizes, ICC.
- Stratified or matched-pair cluster designs improve balance when cluster count is small.

14.9.9 Reporting

A defensible cluster-trial report names the unit of randomisation, the unit of analysis, the planning ICC, and the achieved ICC, and explains how the analytical model handles their potential mismatch. Where the number of clusters is below thirty, state which small-sample correction was used for standard errors and degrees of freedom, since naive likelihood-based intervals are anti-conservative in this regime. If clusters varied substantially in size, mention whether weights were applied and how missing clusters or partial cluster dropout were handled, because differential cluster attrition can bias the estimated treatment effect even when individual-level missingness is modest.

14.9.10 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.

14.9.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.10 Cohen's Kappa

14.10.1 Introduction

Cohen's kappa, introduced by Jacob Cohen in 1960, measures agreement between two raters on a categorical scale, with a correction for the agreement that would be expected by chance alone. Raw percent agreement can look impressive even when most of it reflects coincidence — two raters who both diagnose 90 % of patients as healthy will agree on at least 81 % of cases purely by chance. Kappa subtracts this chance baseline, leaving a more honest measure of the genuine signal in inter-rater agreement. It is now

the de facto standard for inter-rater reliability on nominal categorical outcomes, widely used in imaging-rater studies, pathology grading, diagnostic-criteria validation, and any reliability assessment with two raters and a categorical scale.

14.10.2 Prerequisites

A working understanding of categorical data, contingency-table summaries, observed agreement as a percentage, and the concept of chance-expected agreement under independent raters.

14.10.3 Theory

Cohen's kappa is

$$\kappa = \frac{p_o - p_e}{1 - p_e},$$

where p_o is the observed proportion of cases on which the two raters agreed and p_e is the proportion expected by chance, computed as $p_e = \sum_k p_{1k} p_{2k}$ from the marginal proportions. Kappa ranges from -1 to $+1$: $\kappa = 1$ is perfect agreement, $\kappa = 0$ is exactly chance-level, and negative values indicate systematic disagreement worse than chance.

Landis and Koch (1977) proposed widely used (and equally widely criticised) benchmarks: 0.01–0.20 slight, 0.21–0.40 fair, 0.41–0.60 moderate, 0.61–0.80 substantial, and 0.81–1.00 almost perfect. These thresholds are descriptive heuristics, not strict cut-offs.

14.10.4 Assumptions

There are exactly two raters, the categorical scale has mutually exclusive and exhaustive categories, ratings between the two raters are independent (one did not see or influence the other), and both raters classify the same set of subjects.

14.10.5 R Implementation

```
library(psych)

set.seed(2026)
n ← 100
rater1 ← factor(sample(c("pos", "neg"), n, replace = TRUE))
agree ← rbinom(n, 1, 0.8)
rater2 ← ifelse(agree == 1, as.character(rater1),
                ifelse(rater1 == "pos", "neg", "pos"))
rater2 ← factor(rater2, levels = levels(rater1))

tab ← table(rater1, rater2)
tab

cohen.kappa(cbind(rater1, rater2))
```

14.10.6 Output & Results

`cohen.kappa()` returns the unweighted (and weighted, where applicable) kappa statistic, its standard error, and a confidence interval. Reporting the contingency table alongside the kappa value gives readers the raw evidence; a small or imbalanced contingency table can produce surprising kappa behaviour and the table is the only diagnostic that reveals it.

14.10.7 Interpretation

A reporting sentence: "Inter-rater agreement on the binary classification was substantial (Cohen's $\kappa = 0.58$, 95 % CI 0.41 to 0.75) per the Landis-Koch benchmarks, with observed agreement 80 % and chance-expected agreement 52 %. The contingency table showed approximately balanced marginals (rater 1: 51 % positive; rater 2: 53 % positive), so the kappa-paradox concern that affects skewed-marginal samples does not apply here." Always report observed agreement, marginals, and the kappa value together.

14.10.8 Practical Tips

- Kappa depends on the prevalence and balance of the categories — the well-known "kappa paradox": very low-prevalence categories can produce small kappa values even when observed agreement is high, because the chance-expected agreement is also high. Always report kappa alongside the observed agreement and the marginals so readers can diagnose this.
- For nominal scales with more than two categories, the unweighted kappa treats every disagreement equally; for ordinal scales (mild / moderate / severe) use the weighted kappa to credit partial agreement, with quadratic weights as the conventional default.
- For more than two raters, use Fleiss's kappa (a generalisation of Cohen's kappa) or, when the rating is on an interval-like scale, the intraclass correlation coefficient (ICC); these handle multi-rater designs that Cohen's kappa cannot.
- Confidence intervals on kappa via the delta method are routinely reported by `psych::cohen.kappa()`; bootstrap CIs are preferable for small samples or when the marginal distributions are very imbalanced.
- Distinguish inter-rater reliability (different raters on the same subjects) from intra-rater reliability (the same rater on different occasions); both can be assessed by kappa, but the design and inferential implications differ.
- For continuous outcomes use a Bland-Altman analysis or the ICC; kappa is appropriate only for categorical scales and is misleading when applied to continuous data after dichotomisation.

14.10.9 R Packages Used

`psych::cohen.kappa()` for the canonical Cohen's and weighted kappa with confidence intervals; `irr::kappa2()` and `irr::kappam.fleiss()` for an alternative interface and multi-rater extensions; `vcd::Kappa()` for kappa within the `vcd` contingency-table ecosystem; `epibasix` for kappa with epidemiological reporting; `DescTools::CohenKappa()` for fast computation alongside related descriptive statistics.

14.10.10 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.

14.10.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.11 Conditional Power

14.11.1 Introduction

Conditional power is the probability of rejecting the null at the final analysis given the observed interim data and an assumed treatment effect for the remainder. It is the standard tool for futility stopping: a low conditional power indicates the trial is unlikely to succeed even with favourable future data.

14.11.2 Prerequisites

Group-sequential designs; interim analyses; power calculations.

14.11.3 Theory

For a two-sided Z-test at fraction t of information, conditional power under assumption θ is

$$\mathrm{CP}(\theta) = 1 - \Phi\left(\frac{z_{1-\alpha}\sqrt{1} - \sqrt{t}\,Z_t - (1-t)\theta/\sqrt{V}}{\sqrt{1-t}}\right)$$

where Z_t is the observed interim statistic and V is the information.

Typical futility trigger: stop if CP(assumed effect) < 20 %.

14.11.4 Assumptions

Assumed effect for the remainder of the trial is appropriate (observed, target, or conservative); test is Z-like.

14.11.5 R Implementation

```r
library(rpact)

design <- getDesignGroupSequential(
  sided = 1, alpha = 0.025, beta = 0.2,
  typeOfDesign = "OF",
  informationRates = c(0.5, 1)
)

# Interim analysis results: observed Z = 0.4 (weak signal)
results <- getDataSet(n1 = c(50, 50), means1 = c(0.1, NA),
                      stDevs1 = c(1, NA), n2 = c(50, 50),
                      means2 = c(0, NA), stDevs2 = c(1, NA))

ana <- getAnalysisResults(design = design, dataInput = results,
                          thetaH0 = 0, stage = 1)

# Conditional power at planned effect (delta = 0.3) and observed trend
cond <- getConditionalPower(ana,
                            nPlanned = c(50, 50),
                            thetaH1 = c(0.1, 0.3))
cond
```

14.11.6 Output & Results

Conditional power at two assumed future effects; if CP at the planned effect is low (say < 20 %), a DMC might recommend futility stopping.

14.11.7 Interpretation

“Interim conditional power under the originally planned effect was 0.18; under the observed interim trend, 0.11. The DMC recommended futility stopping at the pre-specified < 20 % threshold.”

14.11.8 Practical Tips

- Always pre-specify the futility threshold and assumed effect in the SAP.
- CP under the **observed effect** is the “optimistic” view; CP under zero effect is the conservative view.
- **Predictive power** (Bayesian analogue) averages CP over a posterior for the effect – often preferred in modern trials.
- Futility boundaries are typically **non-binding** (can be overridden by DMC) to preserve alpha.
- CP-based futility can save substantial cost in otherwise failing trials.

14.11.9 Reporting

A clear conditional-power report distinguishes the assumed effect from the observed interim effect, and presents both anchors so reviewers can judge the futility decision against the original design intent and against the trial’s actual interim trajectory. Quote the threshold prospectively recorded in the statistical analysis plan and state whether crossing it triggered a binding stop or only a recommendation that the data monitoring

committee could override. Where Bayesian predictive power was computed, report the prior used for the effect and explain why that prior was deemed plausible at the design stage, since the futility decision is only as defensible as the assumptions feeding it.

14.11.10 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.

14.11.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.12 Cutpoint Selection

14.12.1 Introduction

Once a continuous diagnostic test's discrimination is established, a clinical cutpoint must be chosen to turn continuous scores into binary decisions. The optimal cutpoint depends on the trade-off between sensitivity and specificity and on relative costs of false positives vs false negatives.

14.12.2 Prerequisites

Sensitivity and specificity; ROC curves.

14.12.3 Theory

Common criteria: - **Youden's J**: maximise $\text{Sens} + \text{Spec} - 1$. Implicitly assumes equal cost of FN and FP. - **Closest to (0, 1)**: minimise $\sqrt{(1-\text{Sens})^2 + (1-\text{Spec})^2}$. - **Cost-weighted**: minimise $c_{FN}(1-\text{Sens}) \cdot p_D + c_{FP}(1-\text{Spec}) \cdot (1-p_D)$ where p_D is prevalence. - **Target specificity (or sensitivity)**: fix one and maximise the other.

14.12.4 Assumptions

Target population has a known prevalence; costs of misclassification are elicited or set by convention; cutpoint will generalise to a new sample.

14.12.5 R Implementation

```r
library(pROC); library(cutpointr)

set.seed(2026)
n ← 300
disease ← factor(sample(c(0, 1), n, replace = TRUE, prob = c(0.6, 0.4)))
score   ← rnorm(n, mean = ifelse(disease == 1, 1.2, 0))

roc_obj ← roc(response = disease, predictor = score,
              levels = c("0", "1"), direction = "<")
youden_thr ← coords(roc_obj, "best", best.method = "youden",
                    transpose = FALSE)
youden_thr

# cutpointr package: multiple criteria in one call
cp ← cutpointr(data.frame(score, disease),
               x = score, class = disease,
               method = maximize_metric, metric = youden)
summary(cp)
```

14.12.6 Output & Results

Cutpoint at Youden's J and associated sensitivity/specificity. Common cutpoint in this simulation: ~0.6 with sensitivity ~0.7 and specificity ~0.7.

14.12.7 Interpretation

"Maximum Youden's J was achieved at a cutpoint of 0.63 (sensitivity 0.72, specificity 0.71). In a population with 40 % prevalence, this yields PPV 0.62 and NPV 0.80."

14.12.8 Practical Tips

- Internal cutpoints overfit the training data; cross-validate or use a separate holdout set.
- Clinical cutpoints should be stable, rounded to a meaningful precision, and validated prospectively.
- For screening tests, favour sensitivity; for confirmatory tests, favour specificity.
- Report both the cutpoint and its downstream metrics (Sens, Spec, PPV, NPV, LR+, LR-).
- Decision-curve analysis (Vickers-Elkin) incorporates clinical utility across a range of threshold probabilities.

14.12.9 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.

14.12.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.13 Diagnostic Test Accuracy

14.13.1 Introduction

When a new diagnostic test is proposed, the first question is: how accurately does it classify disease status against a reference standard? The answer comes in several complementary numbers – sensitivity, specificity, positive and negative predictive values, likelihood ratios, and the area under the ROC curve. Each answers a different question, and reporting any one alone is inadequate.

14.13.2 Prerequisites

The reader should understand the distinction between a test result (positive/negative) and disease status (truly positive/negative), and should be comfortable with 2×2 tables and proportions.

14.13.3 Theory

Given a binary test and a binary gold-standard diagnosis, every study produces a 2×2 table:

	Disease +	Disease -
Test +	TP	FP
Test -	FN	TN

From this table:

- **Sensitivity** = TP / (TP + FN). The probability that the test is positive in a diseased person.
- **Specificity** = TN / (TN + FP). The probability that the test is negative in a healthy person.
- **Positive predictive value (PPV)** = TP / (TP + FP). The probability of disease given a positive test.
- **Negative predictive value (NPV)** = TN / (TN + FN). The probability of no disease given a negative test.
- **Positive likelihood ratio (LR+)** = sensitivity / (1 - specificity). How many times more likely a positive test is in diseased versus healthy people.
- **Negative likelihood ratio (LR-)** = (1 - sensitivity) / specificity.

Sensitivity and specificity are properties of the test that are (approximately) invariant to prevalence. Predictive values depend strongly on prevalence: a test with 99% sensitivity and 99% specificity still has PPV below 50% when disease prevalence is 1%. Likelihood ratios, via Bayes' theorem, convert a pre-test probability into a post-test probability and thus tie the test-level quantities to the clinical reasoning a doctor actually does.

For a continuous marker, each possible threshold produces a pair (sensitivity, 1 - specificity). Plotting these across all thresholds gives the **receiver operating characteristic (ROC) curve**. The **area under the ROC curve (AUC)** is the probability that a randomly chosen diseased person has a higher marker value than a randomly chosen healthy person – an interpretable summary of discrimination independent of any threshold.

14.13.4 Assumptions

- The reference standard is a true gold standard (otherwise sensitivity and specificity are biased).
- The test is evaluated on a representative spectrum of diseased and healthy individuals (spectrum bias can inflate apparent accuracy).
- Test results are read blinded to the reference standard (verification and review bias are the two most common threats in reporting).

14.13.5 R Implementation

```
library(pROC)
library(cutpointr)

set.seed(2026)
n ← 200
disease ← rbinom(n, 1, 0.3)
marker ← ifelse(disease == 1,
                rnorm(n, mean = 60, sd = 12),
                rnorm(n, mean = 45, sd = 10))

df ← data.frame(disease = factor(disease, levels = c(0, 1),
                                 labels = c("healthy", "diseased")),
                marker = marker)

roc_obj ← roc(df$disease, df$marker,
              levels = c("healthy", "diseased"), direction = "<")

auc(roc_obj)
ci.auc(roc_obj)

plot(roc_obj, print.auc = TRUE, ci = TRUE)

cp ← cutpointr(df, marker, disease,
               pos_class = "diseased",
               method = maximize_metric,
               metric = youden)
summary(cp)
plot(cp)
```

`pROC::roc()` constructs the ROC object from marker values and disease labels. `auc()` and `ci.auc()` give the point estimate and 95% CI. The `cutpointr` package finds the threshold that maximises Youden's index (sensitivity + specificity - 1) and reports the operating characteristics at that threshold.

14.13.6 Output & Results

The simulated example gives an AUC of approximately 0.85 (95% CI 0.79 to 0.90). The Youden-optimal cutoff is around 52, with sensitivity ~0.75 and specificity ~0.80 at that threshold.

14.13.7 Interpretation

A manuscript table should report sensitivity, specificity, PPV, NPV, LR+, LR-, and the AUC, each with 95% confidence intervals. For a binary test:

"Sensitivity was 75% (95% CI 66-83%), specificity 80% (74-86%), PPV 62% (52-72%), NPV 88% (82-93%), LR+ 3.8 (2.7-5.2), LR- 0.31 (0.22-0.44), AUC 0.85 (0.79-0.90)."

Crucially, the PPV depends on the disease prevalence in the study population. If the intended clinical use is in a lower-prevalence setting, report the projected PPV at that prevalence using Bayes' theorem.

14.13.8 Practical Tips

- Always report the reference standard explicitly and justify it as a gold standard.
- Report sensitivity and specificity *with* predictive values, not instead of them. Predictive values are what the clinician uses; sensitivity and specificity are what the test provides.
- Use 95% Wilson or Clopper-Pearson intervals for proportions, not the Wald interval, which can extend outside $[0, 1]$ or have poor coverage near 0 and 1.
- Avoid choosing a threshold from the same data that will report its performance; hold out a validation set or use cross-validation.
- Follow STARD reporting guidelines: flow diagram, blinding, reference-standard description, thresholds, and indeterminate results.

14.13.9 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.

14.13.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR

- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.14 Clinical Equivalence Trials

14.14.1 Introduction

Clinical equivalence trials test whether two treatments are clinically interchangeable within a pre-specified equivalence margin — narrow enough that any difference inside the margin is regarded as clinically meaningless. Bioequivalence studies, used to support generic-drug approval, are the archetypal application: regulators require that the ratio of pharmacokinetic parameters between a generic and the reference product fall within 80 % to 125 % of unity, demonstrating that the generic delivers essentially the same exposure as the originator. Equivalence frameworks also apply in clinical contexts where two interventions are competing on safety or convenience and an investigator wishes to show "no clinically meaningful difference" rather than the more usual "test product is better".

14.14.2 Prerequisites

A working understanding of non-inferiority trial design, confidence-interval logic, and the distinction between absence of evidence (high p-value) and evidence of absence (CI within an equivalence margin).

14.14.3 Theory

The two one-sided tests (TOST) procedure rejects the null of non-equivalence if and only if the two-sided 90 % confidence interval of the treatment effect lies entirely within the equivalence margin $[-\Delta, +\Delta]$. This is equivalent to two one-sided tests, each at $\alpha = 0.05$, against the lower and upper non-equivalence boundaries; the overall type-I error is preserved at 0.05 because at most one of the two boundaries can be violated under any single state of the world.

For bioequivalence on log-transformed pharmacokinetic parameters such as AUC and $C_{\max}$, the ratio μ_T/μ_R must lie within $(0.80, 1.25)$, corresponding to $\pm\log(1.25) \approx \pm 0.223$ on the natural-log scale. Log-transformation is mandated by regulators because it makes the test-to-reference ratio symmetric around unity and renders the inference Normal-theory tractable.

14.14.4 Assumptions

The outcome (typically a log-transformed pharmacokinetic parameter) is approximately Normal, the design is a crossover with adequate washout to eliminate carryover, the within-subject variance is reasonably estimated, and observations are independent across subjects.

14.14.5 R Implementation

```r
library(PowerTOST)

n ← sampleN.TOST(alpha = 0.05, targetpower = 0.80,
                 theta0 = 0.95,
                 theta1 = 0.80, theta2 = 1.25,
                 CV = 0.20,
                 design = "2x2")
n

set.seed(2026)
n_sub ← 30
subject_re ← rnorm(n_sub, 0, 0.15)
period1 ← exp(subject_re + rnorm(n_sub, 0, 0.1))
period2 ← exp(subject_re + log(0.95) + rnorm(n_sub, 0, 0.1))

log_diff ← log(period1) - log(period2)
m  ← mean(log_diff); sd_d ← sd(log_diff)
ci ← m + c(-1, 1) * qt(0.95, df = n_sub - 1) * sd_d / sqrt(n_sub)
exp(ci)
```

14.14.6 Output & Results

`sampleN.TOST()` returns the sample size required to achieve target power for the bioequivalence hypothesis under specified true ratio and within-subject coefficient of variation. The simulation block then computes a 90 % confidence interval on the test-to-reference ratio, which is the regulatory-relevant inference. Reporting both the point ratio and the CI is the standard expected by FDA and EMA.

14.14.7 Interpretation

A reporting sentence: "The 90 % confidence interval of the test-to-reference ratio was 0.92 to 1.08, fully within the regulatory bioequivalence window of 0.80 to 1.25 for both AUC and $C_{\max}$. The geometric mean ratio was 1.00, with within-subject coefficient of variation 18 %. Bioequivalence was therefore demonstrated under standard FDA and EMA criteria; the formal TOST procedure rejected non-equivalence on both boundaries at $\alpha = 0.05$." Always report the 90 % CI on the back-transformed scale.

14.14.8 Practical Tips

- Always analyse log-transformed PK parameters rather than raw values; log-ratios are symmetric around unity, the regulatory equivalence window translates to a symmetric range on the log scale, and Normal-theory inference is tractable on the log scale.
- Use the 90 % confidence interval, not the 95 %; the TOST procedure at $\alpha = 0.05$ corresponds exactly to a two-sided 90 % CI lying entirely within the equivalence margin, and this is the regulatory standard.
- The bioequivalence margin (0.80 to 1.25) is fixed by FDA and EMA regulation; clinical equivalence margins for non-PK outcomes must be pre-specified and justified clinically, because the equivalence claim hinges entirely on the margin width.

- Reference-scaled bioequivalence (RSABE) is used for highly variable drugs with within-subject CV above 30 %; the equivalence margin is then widened proportionally to the reference-product variability, preserving statistical feasibility for inherently variable products.
- Replicate crossover designs (each subject receives test and reference twice) reduce within-subject variance and improve efficiency; they are the standard for highly variable drugs and increasingly the default in modern bioequivalence trials.
- Pre-specify the equivalence margin, washout period, and analysis model in the protocol; FDA and EMA scrutinise these choices closely, and post-hoc margin selection is grounds for rejection.

14.14.9 R Packages Used

`PowerTOST` for canonical TOST sample-size calculation, power analysis, and simulation across crossover and parallel-group equivalence designs; `bear` for end-to-end FDA-compliant bioequivalence analysis with all standard reporting; `bioequivalence` and `bioequivalenceR` for alternative interfaces; `replicateBE` for replicate-design bioequivalence with reference-scaled procedures; `BE` for Bayesian bioequivalence approaches.

14.14.10 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.

14.14.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.15 Factorial Trials

14.15.1 Introduction

Factorial trials test two or more interventions in a single experiment: participants are randomised to each factor independently, producing 2×2 (or higher) cells. The design is efficient if the interventions act independently (no interaction); otherwise the interaction itself becomes the primary finding.

14.15.2 Prerequisites

ANOVA; interactions.

14.15.3 Theory

In a 2×2 design, four cells: control, A only, B only, A + B. Main effects are estimated by averaging over the other factor; the interaction compares observed A + B effect to the sum of individual effects.

If interaction is negligible, factorial is efficient: same power as two separate trials with half the total sample size.

14.15.4 Assumptions

Treatments do not interact (or the interaction is the inferential target); randomisation is to each factor independently; outcome is measured under identical conditions across cells.

14.15.5 R Implementation

```r
set.seed(2026)
n_per_cell ← 40
A ← factor(rep(c("no", "yes"), each = 2 * n_per_cell))
B ← factor(rep(rep(c("no", "yes"), each = n_per_cell), 2))

# Simulate additive effects, mild positive interaction
y ← rnorm(4 * n_per_cell) +
     ifelse(A == "yes", 0.5, 0) +
     ifelse(B == "yes", 0.3, 0) +
     ifelse(A == "yes" & B == "yes", 0.2, 0)

fit ← lm(y ~ A * B)
summary(fit)$coefficients
anova(fit)
```

14.15.6 Output & Results

Main-effect estimates for A and B plus the interaction term. The interaction is small, consistent with the simulated +0.2.

14.15.7 Interpretation

“The factorial analysis estimated a main effect of A = 0.48, B = 0.31, with a small positive interaction (0.22, p = 0.14). Main-effect analyses are interpretable in the absence of significant interaction.”

14.15.8 Practical Tips

- Pre-specify the interaction test and its interpretation; a non-significant test does not guarantee additivity.
- Factorial trials are under-powered for interactions unless specifically designed for them.

- Report both main effects (averaged over the other factor) and cell means.
- Partial factorial ('unbalanced') designs drop problematic cells – useful when certain combinations are unethical or impractical.
- For > 3 factors, fractional factorial designs (Taguchi) reduce cell count at the cost of confounding higher-order interactions.

14.15.9 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.

14.15.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.16 Subgroup Forest Plots

14.16.1 Introduction

Subgroup forest plots display, on a single figure, the treatment effect estimate and confidence interval for each pre-specified subgroup of a clinical trial alongside the overall effect. They are the standard tool for visually communicating effect modification — whether the intervention works differently in men and women, in older and younger patients, in mild and severe disease — and they are now a near-universal element of CONSORT-compliant trial reporting. The compact horizontal-error-bar layout makes the magnitude and precision of each subgroup-specific effect immediately legible, while a vertical reference line at the null anchors interpretation. The plot's strength is also its risk: readers eye-ball heterogeneity at a glance and may infer effect modification where the formal interaction test does not support it.

14.16.2 Prerequisites

A working understanding of pre-specified subgroup analysis, the difference between within-subgroup tests and the test for interaction, and confidence-interval visualisation.

14.16.3 Theory

Each row of a forest plot shows the subgroup name, sample sizes per arm, the point estimate of the treatment effect (or the within-subgroup analogue), and its 95 % confidence interval as a horizontal whisker. An overall effect — the marginal estimate

across the full trial population — appears at the top or bottom of the plot for reference. A vertical line at the null value (0 for differences, 1 for ratios) anchors interpretation, and the formal test for interaction (whether the effect varies across subgroups beyond chance) is reported either in the figure or alongside it.

14.16.4 Assumptions

Subgroups are pre-specified in the trial protocol or statistical analysis plan rather than chosen post hoc, the effect estimates and confidence intervals are correctly computed for each subgroup, and the formal interaction test (rather than within-subgroup p-value comparisons) is the basis for any claim of effect modification.

14.16.5 R Implementation

```
library(ggplot2); library(dplyr)

set.seed(2026)
subgroups ← data.frame(
  group   = c("Overall", "Male", "Female",
              "Age < 65", "Age ⩾ 65",
              "Mild", "Moderate", "Severe"),
  n1      = c(200, 105, 95, 90, 110, 70, 80, 50),
  n2      = c(200, 98, 102, 92, 108, 72, 79, 49),
  effect  = c(0.30, 0.18, 0.43, 0.22, 0.38, 0.12, 0.32, 0.55),
  lower   = c(0.15, -0.02, 0.21, 0.02, 0.18, -0.14, 0.10, 0.28),
  upper   = c(0.45, 0.38, 0.65, 0.42, 0.58, 0.38, 0.54, 0.82)
)

ggplot(subgroups, aes(y = factor(group, levels = rev(group)),
                      x = effect)) +
  geom_vline(xintercept = 0, linetype = 2, colour = "grey50") +
  geom_errorbarh(aes(xmin = lower, xmax = upper), height = 0.15) +
  geom_point(size = 3, colour = "#2A9D8F") +
  labs(x = "Treatment effect (with 95% CI)", y = NULL,
       title = "Subgroup forest plot") +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())
```

14.16.6 Output & Results

The resulting plot is a vertical list of subgroups with horizontal confidence intervals referenced to the null line. Combining this with a column of sample sizes and an interaction-test p-value column produces a publication-ready forest plot. Most clinical-trial reports also include the formal interaction p-value at the right of the figure or in the figure caption.

14.16.7 Interpretation

A reporting sentence: “The forest plot showed the treatment effect was present across all eight pre-specified subgroups; directional heterogeneity was observed by sex (point estimate 0.43 in women vs 0.18 in men) and disease severity (0.55 in severe vs 0.12 in mild patients). Formal tests for interaction were not significant ($p_{\text{sex}} = 0.11$, $p_{\text{severity}} = 0.07$),

suggesting the observed within-subgroup differences may reflect sampling rather than genuine effect modification. Subgroups were pre-specified in the SAP." Always report formal interaction tests, not within-subgroup p-values.

14.16.8 Practical Tips

- Always display subgroup sample sizes alongside each row; a small subgroup with a wide CI can look extreme on the plot but carry very little weight in the overall conclusion, and readers benefit from seeing the precision context directly.
- Order subgroups by category (sex, age, disease severity, geographic region) — never by point estimate. Post-hoc ordering is a recurring source of biased visual interpretation and is increasingly flagged by trial reviewers.
- Show the overall trial effect prominently, at the top or bottom of the plot, as the reference against which subgroup deviations are read; subgroup forest plots without an overall reference line are difficult to interpret.
- For odds ratios, hazard ratios, or risk ratios, use a logarithmic scale on the horizontal axis; on the linear scale, ratios appear asymmetric and visual judgements of "large" vs "small" effects become misleading.
- Include the formal interaction p-value in the figure or directly beside it; this discourages the well-known fallacy of comparing within-subgroup p-values, which are always under-powered and routinely yield "significant in one subgroup, not the other" patterns by chance alone.
- Keep the number of subgroups manageable (typically 5–10); too many subgroups crowd the plot and create false-positive risk through multiple comparisons even with correct interaction-test reporting.

14.16.9 R Packages Used

`ggplot2` for custom forest plots with full layout control; `forestplot` for highly customised clinical-trial forest plots with multiple columns and risk-of-bias annotation; `forester` for tidyverse-friendly forest-plot construction with built-in subgroup-table integration; `metafor::forest()` when the underlying analysis is meta-analytic; `survminer` for survival-specific forest plots when subgroup effects are hazard ratios.

14.16.10 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.

14.16.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman

- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.17 Intraclass Correlation Coefficient (ICC)

14.17.1 Introduction

The Intraclass Correlation Coefficient (ICC) quantifies inter-rater reliability on continuous measurements. Unlike Pearson's correlation, it penalises systematic rater bias (two raters who differ by a constant still have Pearson = 1 but ICC < 1). Several ICC forms reflect different study designs and questions.

14.17.2 Prerequisites

Variance components; random vs fixed effects.

14.17.3 Theory

Shrout-Fleiss (1979) forms: - **ICC(1, 1)**: one-way random effects; each subject rated by a different random rater. - **ICC(2, 1)**: two-way random effects; **absolute agreement** between raters. - **ICC(3, 1)**: two-way mixed effects; **consistency** (ignores systematic rater bias).

Single rater vs average of k raters: ICC(x, 1) vs ICC(x, k).

14.17.4 Assumptions

Subjects and raters are drawn from appropriate populations; ratings are independent given subject; ICC form matches the intended use.

14.17.5 R Implementation

```
library(psych)

set.seed(2026)
n ← 30
subject_re ← rnorm(n, 0, 1)

# 3 raters, each with own systematic bias
r1 ← subject_re + rnorm(n, 0, 0.4)
r2 ← subject_re + 0.3 + rnorm(n, 0, 0.4)    # rater 2 higher by 0.3
r3 ← subject_re - 0.2 + rnorm(n, 0, 0.4)

df ← cbind(r1, r2, r3)

icc_res ← ICC(df, lmer = FALSE)
icc_res$results[, c("type", "ICC", "lower bound", "upper bound")]
```

14.17.6 Output & Results

Six ICC forms with 95 % CIs. Agreement ICC (2, 1) is lower than consistency ICC (3, 1) when raters have systematic biases.

14.17.7 Interpretation

“Single-rater absolute-agreement ICC(2, 1) = 0.78 (95 % CI 0.63-0.88); consistency ICC(3, 1) = 0.83. Systematic rater differences moderately lowered absolute agreement.”

14.17.8 Practical Tips

- Choose ICC form by study question:
  - **ICC(2, 1)** if you want to quantify agreement including systematic rater differences.
  - **ICC(3, 1)** if you will remove systematic rater effects in practice (e.g., calibrate each rater).
- For clinical usability, average-of-k-raters ICCs are the most interpretable.
- Thresholds (Koo-Li): < 0.5 poor, 0.5-0.75 moderate, 0.75-0.9 good, $>$ 0.9 excellent.
- ICC $<$ 0.7 is usually insufficient for individual decision-making.
- Pair ICC with Bland-Altman for graphical assessment of rater-specific bias.

14.17.9 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.

14.17.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.18 Interim Analyses and Group Sequential Designs

14.18.1 Introduction

Group-sequential designs schedule a sequence of interim analyses at pre-specified information fractions, each with efficacy and/or futility boundaries. Trials can stop early for benefit, harm, or futility while preserving overall Type I error. They are the standard for confirmatory trials with ethical imperatives for early stopping.

14.18.2 Prerequisites

Type I error; sample-size calculation; information fraction.

14.18.3 Theory

With K analyses and overall two-sided alpha α, boundaries are chosen so the union of rejection events has probability α under the null.

Common families: - **O'Brien-Fleming**: conservative early, near-nominal late – preserves nominal alpha at final analysis. - **Pocock**: constant boundary across looks – more early stopping but harder to reach final. - **Alpha-spending** (Lan-DeMets): flexible timing; spending function $f(t)$ at information fraction t.

14.18.4 Assumptions

Analyses occur at pre-specified information fractions; test statistic is approximately normal at each look.

14.18.5 R Implementation

```
library(rpact)

# Group-sequential design: 3 analyses, O'Brien-Fleming, alpha=0.025 one-sided
design ← getDesignGroupSequential(
  sided = 1, alpha = 0.025, beta = 0.2,
  informationRates = c(0.33, 0.67, 1),
  typeOfDesign = "OF"
)
print(design)

# Corresponding sample sizes for two-mean comparison
ss ← getSampleSizeMeans(design = design,
                        alternative = 0.3, stDev = 1)
print(ss)
```

14.18.6 Output & Results

Three-stage design with cumulative alpha budgets per stage summing to 0.025; stage sample sizes scale with the chosen information fractions.

14.18.7 Interpretation

"The group-sequential design with O'Brien-Fleming boundary allocated very little alpha to the first two interims (< 0.001 each), preserving nearly the full alpha for the final analysis. Early stopping is extremely unlikely unless the effect is large."

14.18.8 Practical Tips

- OF boundaries are standard for confirmatory trials; Pocock may suit exploratory ones or single-arm phase II.
- Alpha-spending (Lan-DeMets) is more flexible – timing does not need to be exact, only pre-specified.
- Independent DMC must oversee interim analyses; investigators remain blinded.
- Adjust point estimates and CIs for stopping (median-unbiased, repeated CI).

- Futility boundaries (betagamma spending) complement efficacy and can be non-binding.

14.18.9 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.

14.18.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.19 ITT vs Per-Protocol Analysis

14.19.1 Introduction

The **intent-to-treat (ITT)** principle analyses participants according to their randomised assignment, irrespective of actual treatment received or adherence. It estimates the effect of *prescribing* the intervention under real-world conditions. **Per-protocol (PP)** restricts analysis to adherent participants and estimates the effect of *receiving* the intervention – a different estimand.

14.19.2 Prerequisites

Randomisation; estimands; missing data.

14.19.3 Theory

ITT preserves randomisation and tends to be conservative in superiority trials (dilution by non-adherers). PP can exaggerate efficacy or introduce selection bias because adherence is post-randomisation.

Modified ITT (mITT) excludes participants who never started or have no post-baseline data; common but can reintroduce bias if exclusion correlates with arm.

Non-inferiority trials: PP and ITT are both reported; the more conservative result (less close to margin) drives inference.

14.19.4 Assumptions

Randomisation is properly concealed; adherence classification is unaffected by knowledge of outcome; missingness mechanism is characterised.

14.19.5 R Implementation

```r
set.seed(2026)
n_per ← 100
arm ← rep(c("trt", "ctrl"), each = n_per)

# 20 % non-adherence in trt arm; 5 % in ctrl
adhered ← ifelse(arm == "trt", rbinom(n_per * 2, 1, 0.8),
                 rbinom(n_per * 2, 1, 0.95))
adhered[1:n_per] ← rbinom(n_per, 1, 0.8)
adhered[(n_per+1):(2 * n_per)] ← rbinom(n_per, 1, 0.95)

# True effect when actually received
true_effect ← ifelse(adhered == 1 & arm == "trt", 0.7, 0)
y ← rnorm(2 * n_per) + true_effect

# ITT analysis (analyse as randomised)
itt ← t.test(y ~ arm)
# Per-protocol analysis (adherers only)
pp  ← t.test(y[adhered == 1] ~ arm[adhered == 1])

rbind(ITT = c(est = diff(itt$estimate), p = itt$p.value),
      PP  = c(est = diff(pp $estimate), p = pp $p.value))
```

14.19.6 Output & Results

ITT effect is diluted by non-adherence; PP effect recovers the on-treatment effect but is subject to selection bias.

14.19.7 Interpretation

“The primary ITT analysis estimated a 0.56 point advantage for the intervention (95 % CI 0.28-0.84, p < 0.001); PP analysis gave 0.71 (CI 0.38-1.04). ITT is the primary inference; PP is supportive.”

14.19.8 Practical Tips

- Pre-specify ITT as primary in the SAP; never switch post-hoc.
- Report a flow diagram (CONSORT) showing how participants were classified.
- Handle missing outcome data with multiple imputation, not naive exclusion.
- **Complier-average causal effect (CACE)** via instrumental variables estimates the effect among adherers without PP’s selection bias.
- Non-inferiority trials commonly report both ITT and PP; both must meet the margin.

14.19.9 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.

14.19.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.20 Likelihood Ratios

14.20.1 Introduction

Likelihood ratios (LR+ and LR-) summarise the information content of a diagnostic test in a way that is independent of disease prevalence. Sensitivity and specificity describe how the test performs in known-disease and known-no-disease populations, but they do not directly tell a clinician what to believe after observing a positive or negative test result; predictive values do, but they depend on prevalence and are therefore not portable across populations. Likelihood ratios bridge this gap: they combine multiplicatively with the pre-test odds to yield the post-test odds via Bayes' theorem, making them the natural building blocks of clinical Bayesian reasoning. Modern evidence-based-medicine guidance and core teaching texts present LRs as the preferred diagnostic-performance summary precisely because of this prevalence-independence.

14.20.2 Prerequisites

A working understanding of sensitivity and specificity, the relationship between probability and odds, and Bayes' theorem in odds form.

14.20.3 Theory

The two basic likelihood ratios are

$$\mathrm{LR}^{+} = \frac{\mathrm{Sens}}{1 - \mathrm{Spec}}, \qquad \mathrm{LR}^{-} = \frac{1 - \mathrm{Sens}}{\mathrm{Spec}}.$$

The Bayes-theorem update is

$$\mathrm{Odds}_{\mathrm{post}} = \mathrm{Odds}_{\mathrm{pre}} \times \mathrm{LR}.$$

Conventional clinical interpretation: $\mathrm{LR}^{+} > 10$ or $\mathrm{LR}^{-} < 0.1$ is often decisive; 5–10 or 0.1–0.2 is moderate evidence; 2–5 or 0.2–0.5 is weak; near 1 is uninformative. Fagan's nomogram graphically converts pre-test probability and LR directly to post-test probability and is a useful bedside tool.

14.20.4 Assumptions

The same assumptions as for sensitivity and specificity: a reliable gold-standard reference for disease status, accurate test classification, and that the test characteristics estimated in one population generalise to the patients to whom the LR is being applied. Verification bias and spectrum bias both threaten this generalisation.

14.20.5 R Implementation

```
library(epiR)

tab ← as.table(matrix(c(80, 20,
                        20, 880),
                      nrow = 2, byrow = FALSE,
                      dimnames = list(Test = c("+", "-"),
                                      Disease = c("yes", "no"))))
epi.tests(tab)$detail[c("lrpos", "lrneg"), ]

sens ← 0.8; spec ← 0.978
lr_pos ← sens / (1 - spec); lr_neg ← (1 - sens) / spec

prior_odds ← 0.1 / 0.9
post_odds  ← prior_odds * lr_pos
post_prob  ← post_odds / (1 + post_odds)
c(lr_pos = lr_pos, lr_neg = lr_neg, post_prob_after_pos = post_prob)
```

14.20.6 Output & Results

`epi.tests()` reports LR+ and LR- with their confidence intervals from the input contingency table. The Bayesian update example shows how a 10 % pre-test probability rises to roughly 80 % post-test after a positive result with LR+ = 36, illustrating the multiplicative-on-odds nature of the update.

14.20.7 Interpretation

A reporting sentence: “The diagnostic test had sensitivity 80 % (95 % CI 71 to 87 %) and specificity 97.8 % (96.4 to 98.7), corresponding to LR+ = 36 (95 % CI 16 to 81) and LR- = 0.20 (95 % CI 0.13 to 0.31). A positive test raised the pre-test probability of 10 % to a post-test probability of 80 %, while a negative test reduced it to roughly 2 %. The test is therefore strongly informative in both directions for the typical pre-test probability range of this clinical setting.” Always report LRs with CIs.

14.20.8 Practical Tips

- Report likelihood ratios with their 95 % confidence intervals; wide CIs indicate fragile diagnostic estimates and should temper interpretation, especially when small sample sizes or rare disease drive uncertainty in sensitivity or specificity.
- For multi-category or ordinal tests (rating scales, semi-quantitative biomarker results), compute stratum-specific LRs for each score level rather than collapsing to a single binary LR; this preserves the information content of the gradation.

- Likelihood ratios generalise across populations as long as the test characteristics (sensitivity, specificity) hold in the new population; this is their primary advantage over predictive values, which depend on local prevalence and do not transfer.
- Clinical decision thresholds are often pre-specified in terms of required LR (e.g., LR+ ≥ 10 to justify initiating treatment, LR- ≤ 0.1 to confidently rule out disease); building these thresholds into the diagnostic pathway is the operational analogue of Bayesian reasoning at the bedside.
- Chain multiple test results by multiplying their LRs only if the tests are conditionally independent given disease status; in practice this assumption is often violated (a second test of the same type is correlated with the first), and joint LRs from a multivariable predictor are often more honest.
- For complex multi-variable diagnostic tools (clinical prediction rules), the LR concept generalises naturally — the rule's score corresponds to a stratum-specific LR — and is a useful way to communicate the rule's discrimination at decision-relevant cut-points.

14.20.9 R Packages Used

`epiR::epi.tests()` for canonical sensitivity, specificity, predictive values, and likelihood ratios with confidence intervals from a 2 × 2 table; `epibasix` for an alternative interface; `caret::confusionMatrix()` for ML-style classification metrics including LRs; `pROC` for LRs across the full ROC operating range; `bayesmeta` and related packages for Bayesian meta-analytic pooling of likelihood ratios across diagnostic-accuracy studies.

14.20.10 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.

14.20.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.21 Minimisation Algorithm

14.21.1 Introduction

Minimisation, introduced by Taves (1974) and refined by Pocock and Simon (1975), is a covariate-adaptive allocation procedure that assigns each new participant to the treatment arm that prospectively makes the distribution of pre-specified prognostic covariates most

balanced across arms. Unlike stratified block randomisation, which requires a separate randomisation list for every stratum and quickly becomes impractical when more than two or three covariates need balancing, minimisation handles many prognostic factors simultaneously in a single allocation framework. It is now widely used in trials with several important baseline covariates and small-to-moderate sample sizes, where stratified randomisation would create too many empty strata to be useful.

14.21.2 Prerequisites

A working understanding of simple, block, and stratified randomisation; balance metrics for cross-tabulated baseline covariates; and the regulatory framework around covariate-adaptive allocation.

14.21.3 Theory

For each candidate arm assignment, the algorithm computes a balance score — typically the sum across covariates of marginal imbalances that would result from that assignment. The new participant is allocated to the balance-minimising arm with high probability p (commonly 0.8 or 0.9), and to the other arm with probability $1 - p$ to preserve a degree of allocation unpredictability. The probabilistic element matters: a deterministic minimisation that always chooses the balance-minimising arm becomes predictable to investigators who know the algorithm and the previous assignments, compromising allocation concealment.

14.21.4 Assumptions

The covariates to be balanced are pre-specified in the protocol, allocation is performed centrally (typically through an interactive web-response system) rather than manually, and the trial's analysis model includes all minimisation covariates as fixed effects to preserve valid inference under randomisation theory.

14.21.5 R Implementation

```r
library(Minirand)

set.seed(2026)
n <- 60
covmat <- data.frame(
  centre = sample(c("A", "B", "C"), n, replace = TRUE),
  sex    = sample(c("M", "F"), n, replace = TRUE),
  age_g  = sample(c("young", "old"), n, replace = TRUE)
)

res <- character(n)
for (j in 1:n) {
  res[j] <- Minirand(covmat = covmat, covwt = rep(1, 3),
                     ntrt = 2, trtseq = c("A", "B"),
                     ratio = c(1, 1),
                     p = 0.9, j = j, result = res)
}

table(trt = res, centre = covmat$centre)
table(trt = res, sex = covmat$sex)
```

14.21.6 Output & Results

`Minirand()` allocates each subject sequentially based on the prior allocation history and the new subject's covariate profile. Cross-tabulations of treatment by each covariate show approximately equal arm counts within every covariate level — the design's primary objective — even when no single covariate combination has many subjects.

14.21.7 Interpretation

A reporting sentence: "Treatment allocation used Pocock-Simon minimisation balancing on three pre-specified covariates (centre, sex, age category), each with equal weight; the random component used probability 0.9 of allocation to the balance-minimising arm. The trial's analysis model retained centre, sex, and age category as fixed-effect covariates to preserve valid inference under minimisation; in the final 60-subject sample, the maximum marginal arm imbalance on any covariate was 1 subject." Always state the random-component probability and the analysis-model covariates.

14.21.8 Practical Tips

- Always analyse the trial with the minimisation covariates as fixed-effect adjustments in the primary analysis model; omitting them violates the randomisation-inference framework that minimisation relies on, and the resulting standard errors are typically too small.
- Pre-specify the covariates, their weights, and the random-component probability p in the protocol and SAP; adding covariates post hoc defeats minimisation's protective role and is disallowed by most regulators.
- Commercial IWRS systems are required for robust minimisation in any non-trivial trial; manual implementation is error-prone, especially as the trial grows, and a single manual error can compromise allocation concealment for the entire study.

- FDA and EMA accept minimisation when the method, covariates, and analysis model are pre-specified; the regulatory concern about covariate-adaptive allocation is largely addressed by transparent documentation and analysis-model adjustment.
- Minimisation is less transparent than stratified block randomisation — investigators cannot reproduce the allocation list from a simple description — so the protocol should describe the algorithm carefully, including the balance metric, weights, and probability parameter.
- For trials with very few prognostic covariates and moderate sample size, stratified block randomisation is often preferable because of its operational simplicity; minimisation's advantage grows with the number of covariates and with smaller per-stratum sample sizes.

14.21.9 R Packages Used

`Minirand` for canonical Pocock-Simon minimisation with arbitrary weights and ratio support; `randomizeR` for an alternative interface integrated with broader randomisation simulation; `bcrm` for biased-coin and minimisation alternatives; `RandomizationLogic` for full IWRS-style simulation including audit trail; `Mediana` for trial-design simulation with minimisation-allocation strategies.

14.21.10 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.

14.21.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.22 Missing Data in RCTs

14.22.1 Introduction

Missing data are ubiquitous in RCTs and can invalidate inference if not handled carefully. Three missingness mechanisms characterise the problem: **MCAR** (missingness is random), **MAR** (missingness depends on observed variables), and **MNAR** (missingness depends on the unobserved value). Valid analyses require assumptions on the mechanism.

14.22.2 Prerequisites

Basic probability; estimands framework (ICH E9 R1).

14.22.3 Theory

- **MCAR**: $P(\text{missing}) = P(\text{missing} \mid X, Y)$. Complete-case analysis is unbiased but inefficient.
- **MAR**: $P(\text{missing} \mid X, Y) = P(\text{missing} \mid X)$. Multiple imputation, ML, or weighting is unbiased.
- **MNAR**: $P(\text{missing})$ depends on unobserved Y. Sensitivity analyses with assumed MNAR mechanisms are needed.

Missingness is rarely MCAR in practice; MAR is the default operating assumption, with MNAR as sensitivity.

14.22.4 Assumptions

Missingness pattern is characterised by the analyst; auxiliary variables are included in the imputation model; the mechanism assumption matches the method.

14.22.5 R Implementation

```r
library(mice); library(naniar)

set.seed(2026)
n ← 300
baseline ← rnorm(n, 5, 1)
arm ← rep(c("ctrl", "trt"), each = n/2)
outcome ← baseline + ifelse(arm == "trt", 1, 0) + rnorm(n, 0, 1)

# MAR: missingness depends on baseline
prob_missing ← plogis(-2 + 0.3 * baseline)
outcome[rbinom(n, 1, prob_missing) == 1] ← NA

df ← data.frame(arm = factor(arm), baseline, outcome)

# Missing-data summary
miss_var_summary(df)

# Multiple imputation
imp ← mice(df, m = 10, method = "pmm", printFlag = FALSE)
pool(with(imp, lm(outcome ~ arm + baseline))) %>% summary()
```

14.22.6 Output & Results

Missing-data summary (outcome has ~20 % missingness); pooled MI estimate for treatment effect after adjusting for baseline.

14.22.7 Interpretation

"Under MAR, multiple imputation with 10 imputations gave a treatment effect of 0.94 (95 % CI 0.62-1.26, p < 0.001); complete-case analysis gave a similar estimate, consistent with MAR assumption."

14.22.8 Practical Tips

- Prevent missing data by design (pre-specified follow-up, low attrition) before analysis tricks.
- Distinguish missing at random from missing-completely-at-random; CCA needs the stronger MCAR.
- Always report the missingness rate and pattern by arm; differential missingness is a red flag.
- Pre-specify primary analysis under MAR; sensitivity analyses under MNAR (tipping-point).
- ICH E9 R1 estimands framework formalises how missing data interacts with the estimand; align analysis to the estimand, not vice versa.

14.22.9 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.

14.22.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.23 Multiple Imputation

14.23.1 Introduction

Multiple imputation (MI; Rubin 1987) replaces each missing value with $m > 1$ plausible values drawn from the posterior predictive distribution of the missing data given the observed. The analysis is run on each of the m completed datasets; Rubin's rules combine the results into a single inference reflecting both within-imputation and between-imputation uncertainty.

14.23.2 Prerequisites

Missing-data mechanisms (MAR); Bayesian predictive distributions.

14.23.3 Theory

MI procedure: 1. Impute: create m completed datasets via a predictive model. 2. Analyse: apply the intended model to each dataset. 3. Pool: combine $\bar{\hat{\beta}} = (1/m)\sum \hat{\beta}_k$; total variance $T = \bar{W} + (1 + 1/m)B$, where $\bar{W}$ is mean within-imputation variance and B is between-imputation variance.

`mice` uses chained equations: iteratively impute each variable using the others as predictors.

14.23.4 Assumptions

Missing at random (MAR); imputation model is correctly specified (correct functional form, includes all predictors that might correlate with missingness).

14.23.5 R Implementation

```
library(mice)

set.seed(2026)
n ← 300
df ← data.frame(
  x1 = rnorm(n), x2 = rnorm(n),
  x3 = sample(c("a", "b", "c"), n, replace = TRUE),
  y  = rnorm(n)
)
df$y[sample(n, 60)] ← NA       # 20% missing
df$x2[sample(n, 30)] ← NA      # 10% missing

# Chained-equations imputation, 20 imputations
imp ← mice(df, m = 20, method = c("pmm", "pmm", "polyreg", "pmm"),
           printFlag = FALSE)

# Fit the model on each imputation and pool
fit ← with(imp, lm(y ~ x1 + x2 + x3))
pooled ← pool(fit)
summary(pooled)
```

14.23.6 Output & Results

Pooled regression coefficients with SEs that correctly reflect imputation uncertainty; `fmi` (fraction of missing information) indicates how much of the variance comes from imputation.

14.23.7 Interpretation

“Multiple imputation (m = 20) under MAR gave a pooled coefficient of 0.94 (SE 0.10, 95 % CI 0.74-1.14); fraction of missing information 0.18 suggests reasonable efficiency.”

14.23.8 Practical Tips

- Include **all** variables used in the substantive model, plus auxiliary variables correlated with missingness, in the imputation model.

- m should be ≥ 100 when a large fraction is missing; 20 is a minimum for exploratory work.
- For regression on interactions or non-linearity, include those terms in the imputation model too (impute so the model matches).
- Use predictive mean matching (pmm) for continuous variables to avoid unrealistic extrapolations.
- Always check convergence of chained equations via trace plots.

14.23.9 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.

14.23.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.24 Non-Inferiority Margin Selection

14.24.1 Introduction

Non-inferiority (NI) trials test whether a new treatment is not worse than an active comparator by more than a pre-specified margin Δ. Margin selection is the most consequential and scrutinised part of NI trial design: too loose and a genuinely inferior treatment is approved; too tight and the study is infeasible.

14.24.2 Prerequisites

Hypothesis testing; active-controlled trials; historical-control data.

14.24.3 Theory

Two common approaches: - **Fixed-margin (synthesis) method**: margin Δ is chosen based on the historical effect of the active comparator vs placebo, typically preserving 50-75 % of that effect. Example: if active reduces mortality by 10 % vs placebo, Δ might be set at 5 %. - **Clinical margin**: a clinically judged threshold of practical importance, independent of historical data.

Both require regulatory justification; FDA and EMA typically require the synthesis approach with supporting clinical judgement.

14.24.4 Assumptions

Historical active-vs-placebo effect is consistent and generalisable to the current trial population; assay sensitivity (ability to detect a difference if truly present) is preserved.

14.24.5 R Implementation

```
# Synthesis-method margin calculation
# Historical effect: placebo-controlled active gives risk reduction 10% (95% CI
7%-13%)
# Conservative estimate: lower bound 7%
preservation ← 0.5                    # preserve at least 50% of effect
margin_synthesis ← 0.07 * (1 - preservation)
cat("Synthesis-based NI margin:", margin_synthesis, "\n")

# Sample size for an NI trial with continuous outcome
# Expected true difference 0; margin 0.25 SD; alpha=0.025; power=0.80
library(pwr)
pwr.t.test(d = 0.25, sig.level = 0.025, power = 0.80,
           type = "two.sample", alternative = "greater")
```

14.24.6 Output & Results

Synthesis-based margin (0.035 = 3.5 % preserving 50 % of historical effect); sample-size calculation gives ~252/arm for a 0.25 SD margin.

14.24.7 Interpretation

“The non-inferiority margin was prospectively set at 3.5 percentage points, preserving at least 50 % of the historical benefit of the active comparator (7 % lower bound of historical effect). Sample size was 504 based on 80 % power to rule out a margin with one-sided alpha 0.025.”

14.24.8 Practical Tips

- Margin selection must be pre-specified, regulator-reviewed, and clinically justified.
- Run both ITT and per-protocol analyses; NI requires consistency.
- Beware “biocreep”: repeated non-inferiority approvals without placebo anchoring drift from effective therapy.
- Assay sensitivity is hard to demonstrate without a placebo arm; three-arm trials (NI + placebo) are ideal but often unethical.
- Non-inferiority claims should report the effect estimate and CI, not just “p < 0.025”.

14.24.9 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.

14.24.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.25 O'Brien-Fleming Boundary

14.25.1 Introduction

The O'Brien-Fleming (OF) boundary, introduced by Peter O'Brien and Thomas Fleming in 1979, is the most widely used efficacy stopping boundary for group-sequential clinical trials and the de facto regulatory default for confirmatory phase-3 trials. It is conservative early and liberal late: very little type-I error budget is spent at early interim analyses, so stopping early for efficacy requires an unusually large effect, while the final analysis uses nearly the full nominal alpha. The practical consequence is that interim stops are rare and convincing — they happen only when the treatment effect is much larger than the powered alternative — while the final analysis suffers only a tiny multiplicity penalty if no early stop occurs.

14.25.2 Prerequisites

A working understanding of group-sequential trial design, the alpha-spending framework, and the trade-off between early-stopping ease and final-analysis stringency in repeated-look hypothesis testing.

14.25.3 Theory

The O'Brien-Fleming boundary on the standardised Wald-statistic scale takes the form $c_k = c/\sqrt{t_k}$ at information fraction t_k, so the nominal p-value threshold required at interim k shrinks with $1/\sqrt{t_k}$. In practice, a five-look equally-spaced OF design has nominal α values approximately 5×10^{-6}, 0.0013, 0.008, 0.018, 0.041 at the five sequential analyses (for one-sided $\alpha = 0.025$). The final analysis therefore uses 0.041 instead of 0.025 — a mild penalty for the option to stop early — while early looks are protected against premature rejection.

14.25.4 Assumptions

Information accrues as planned (regular interim spacing or alpha-spending implementation that handles irregular timing), the test statistic is approximately Normal at each look, and the multiplicity correction is pre-specified before any data are unblinded.

14.25.5 R Implementation

```r
library(rpact)

design <- getDesignGroupSequential(
  sided = 1, alpha = 0.025, beta = 0.2,
  typeOfDesign = "OF",
  informationRates = seq(0.2, 1, by = 0.2)
)

print(design$stageLevels)

print(design$criticalValues)

plot(design, type = 1)
```

14.25.6 Output & Results

`rpact` returns the stage-specific nominal alphas (very conservative at early looks, near-nominal at the final analysis) and the corresponding critical values on the standardised test-statistic scale. The boundary plot makes the asymmetric "high early, low late" shape visually obvious and is a standard supplementary figure in trial design documents.

14.25.7 Interpretation

A reporting sentence: "The five-stage group-sequential design with O'Brien-Fleming efficacy boundaries required nominal $p < 5 \times 10^{-6}$ at the 20 % information interim, relaxing to $p < 0.041$ at the final analysis to maintain overall one-sided $\alpha = 0.025$. Early stopping was therefore triggered only by treatment effects substantially larger than the powered alternative; the final analysis suffered only a 0.009 multiplicity penalty (0.041 vs unadjusted 0.025) if no earlier stop occurred. The maximum sample size was 6 % larger than a fixed-design equivalent." Always justify boundary choice.

14.25.8 Practical Tips

- O'Brien-Fleming is the default efficacy boundary in confirmatory phase-3 trials and is virtually always the regulatory expectation; deviations should be justified in the protocol with explicit reasoning about why a different boundary (Pocock, Hwang-Shih-DeCani, custom) is preferred.
- Pair O'Brien-Fleming efficacy boundaries with a non-binding futility boundary (gamma-spending or beta-spending) to detect hopeless trials early; this combination preserves the type-I error of the efficacy analysis while allowing the trial to stop for futility when the conditional power is low.
- Alpha-spending implementations of O'Brien-Fleming (Lan-DeMets with the OF-shape spending function) preserve the OF behaviour under irregular interim timing and are the modern default for handling unscheduled looks.
- Post-stopping treatment-effect point estimates are upwardly biased — the trial stopped early precisely because the random fluctuation of the effect was large. Report repeated confidence intervals (Jennison-Turnbull) or median-unbiased

estimates rather than the naive maximum-likelihood estimate when stopping for efficacy.
- Compared with the Pocock boundary, OF makes early stopping substantially harder but preserves near-nominal final-analysis alpha and a smaller maximum sample size; Pocock makes early stopping easier but at higher final-analysis cost. Choice should reflect the trial's ethical and practical priorities.
- Stage-specific information fractions need not be equally spaced; alpha-spending OF accommodates whatever schedule the data-monitoring committee prefers, but the schedule should be pre-specified or generated from a pre-specified spending function.

14.25.9 R Packages Used

`rpact` for canonical group-sequential design with O'Brien-Fleming, Pocock, Hwang-Shih-DeCani, and custom alpha-spending boundaries; `gsDesign` for an alternative comprehensive group-sequential framework; `ldbounds` for Lan-DeMets alpha-spending; `gsbDesign` for Bayesian group-sequential alternatives; `Mediana` for trial-design simulation including OF and other boundary comparisons.

14.25.10 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.

14.25.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.26 Pocock Boundary

14.26.1 Introduction

The Pocock boundary, introduced by Stuart Pocock in 1977, was the first systematic group-sequential efficacy stopping rule for clinical trials. It uses the same nominal type-I error level at every interim and final analysis, distributing the type-I error budget approximately uniformly across looks. The result is a design that makes early stopping for efficacy comparatively easy — a constant relatively-low nominal threshold across all looks — at the cost of requiring an unusually stringent nominal p-value at the final analysis if no earlier

look has stopped the trial. Pocock boundaries are conceptually simple, ethically attractive when early stopping is a priority, but somewhat costly in terms of maximum sample size compared with the more conservative O'Brien-Fleming alternative.

14.26.2 Prerequisites

A working understanding of group-sequential trial designs, the alpha-spending framework, and the trade-off between early-stopping ease and final-analysis stringency in repeated-look hypothesis testing.

14.26.3 Theory

With K planned analyses and overall two-sided type-I error α, the Pocock boundary uses a constant nominal alpha α^* at every look such that the probability of any rejection event under the null exactly equals α. For $K = 5$ and $\alpha = 0.05$ (two-sided), $\alpha^* \approx 0.0158$ at every look — substantially below the unadjusted 0.05 because of the multiplicity correction. Compared with O'Brien-Fleming, Pocock rejects more easily at early interim analyses (where O'Brien-Fleming demands extreme test statistics) but less easily at the final analysis (where O'Brien-Fleming approaches the unadjusted threshold).

14.26.4 Assumptions

Information accrues as planned (regular interim spacing or alpha-spending implementation that handles irregular timing), the test statistic is approximately Normal at each look, and the multiplicity correction is pre-specified before any data are unblinded.

14.26.5 R Implementation

```
library(rpact)

design ← getDesignGroupSequential(
  sided = 1, alpha = 0.025, beta = 0.2,
  typeOfDesign = "P",
  informationRates = seq(0.2, 1, by = 0.2)
)

print(design$stageLevels)
print(design$criticalValues)

plot(design, type = 1)

ss_pocock ← getSampleSizeMeans(design = design,
                               alternative = 0.3, stDev = 1)

design_of ← getDesignGroupSequential(
  sided = 1, alpha = 0.025, beta = 0.2,
  typeOfDesign = "OF",
  informationRates = seq(0.2, 1, by = 0.2)
)
ss_of ← getSampleSizeMeans(design = design_of,
                           alternative = 0.3, stDev = 1)

c(pocock_max_n = ss_pocock$maxNumberOfSubjects,
  of_max_n    = ss_of   $maxNumberOfSubjects)
```

14.26.6 Output & Results

`rpact` returns the constant nominal alpha at each stage, the corresponding critical values on the test-statistic scale, and the maximum sample size needed to maintain the targeted overall power. Comparing Pocock and O'Brien-Fleming sample-size calculations side by side quantifies the cost of the more aggressive early-stopping property.

14.26.7 Interpretation

A reporting sentence: "The five-stage group-sequential design with Pocock boundaries required nominal $p < 0.0158$ at every interim and final analysis to maintain overall one-sided $\alpha = 0.025$. This design enables earlier efficacy stopping than the equivalent O'Brien-Fleming design, but requires approximately 15 % more maximum sample size if no early stop is triggered. The choice was justified by the ethical imperative to halt enrolment as soon as a clinically meaningful benefit is established, given the trial's seriously-ill population." Always justify the boundary choice ethically.

14.26.8 Practical Tips

- Pocock boundaries favour early stopping for efficacy; O'Brien-Fleming favours late stopping with high final-analysis power. The choice should reflect whether the trial prioritises ethical termination of clearly beneficial interventions (Pocock) or efficient final-analysis confirmation (O'Brien-Fleming).
- The maximum sample size is larger under Pocock than under O'Brien-Fleming for the same overall power; O'Brien-Fleming is usually preferred when a meaningful effect is expected only at the final analysis and ethics permit waiting.
- Alpha-spending implementations (Lan-DeMets with the Pocock-shape spending function αt or related linear forms) approximate Pocock behaviour under irregular interim timing and are the modern default in regulatory submissions.
- Pocock boundaries are now rarely the primary choice in confirmatory phase-3 trials; O'Brien-Fleming has become the regulatory default. Pocock remains useful in phase-2 trials and futility-stopping contexts where early termination is a stronger priority.
- A hybrid Hwang-Shih-DeCani family interpolates between Pocock and O'Brien-Fleming behaviour via a single shape parameter γ, allowing trial designers to dial the trade-off explicitly.
- Pre-specify the boundary type, the spending function, and any contingency for unscheduled interims in the protocol; ad hoc adjustments after data inspection inflate type-I error and are increasingly flagged by regulators.

14.26.9 R Packages Used

`rpact` for canonical group-sequential design with Pocock, O'Brien-Fleming, Hwang-Shih-DeCani, and custom alpha-spending boundaries; `gsDesign` for an alternative comprehensive group-sequential framework; `ldbounds` for Lan-DeMets alpha-spending implementation; `gsbDesign` for Bayesian group-sequential alternatives; `Mediana` for trial-design simulation including Pocock and O'Brien-Fleming boundary comparisons.

14.26.10 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.

14.26.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.27 Predictive Values and Prevalence

14.27.1 Introduction

Sensitivity and specificity are properties of a diagnostic test, not its clinical utility. **Positive predictive value (PPV)** – the probability that a positive test indicates disease – and **negative predictive value (NPV)** depend on disease prevalence. The same test can have very high PPV in a high-prevalence setting and very low PPV in a screening setting.

14.27.2 Prerequisites

Sensitivity and specificity; Bayes' theorem.

14.27.3 Theory

$$\mathrm{PPV} = \frac{\mathrm{Sens} \cdot \mathrm{Prev}}{\mathrm{Sens} \cdot \mathrm{Prev} + (1 - \mathrm{Spec}) \cdot (1 - \mathrm{Prev})}.$$

$$\mathrm{NPV} = \frac{\mathrm{Spec} \cdot (1 - \mathrm{Prev})}{(1 - \mathrm{Sens}) \cdot \mathrm{Prev} + \mathrm{Spec} \cdot (1 - \mathrm{Prev})}.$$

For a fixed test, PPV rises with prevalence and NPV falls. A high-specificity test is essential for screening in low-prevalence populations.

14.27.4 Assumptions

Test characteristics (Sens, Spec) generalise to the target population; prevalence is correctly estimated.

14.27.5 R Implementation

```r
library(epiR)

# 2x2 table from a diagnostic study
tab ← as.table(matrix(c(90, 10,     # TP, FP
                        20, 880),   # FN, TN
                      nrow = 2, byrow = FALSE,
                      dimnames = list(Test = c("+", "-"),
                                      Disease = c("yes", "no"))))

epi.tests(tab, conf.level = 0.95)

# What happens when prevalence is only 1%?
sens ← 0.82; spec ← 0.99
for (p in c(0.5, 0.2, 0.05, 0.01)) {
  ppv ← sens * p / (sens * p + (1 - spec) * (1 - p))
  npv ← spec * (1 - p) / ((1 - sens) * p + spec * (1 - p))
  cat(sprintf("Prev=%.2f  PPV=%.3f  NPV=%.3f\n", p, ppv, npv))
}
```

14.27.6 Output & Results

Test statistics including PPV/NPV in the sample; manual computation shows PPV dropping sharply as prevalence falls, from 0.98 at 50 % prevalence to 0.45 at 1 %.

14.27.7 Interpretation

“In a population with 1 % prevalence, a positive test (sens 82 %, spec 99 %) has a PPV of only 0.45; most positives are false. For screening, test specificity drives PPV far more than sensitivity does.”

14.27.8 Practical Tips

- Always report PPV and NPV at the **target** population’s prevalence, not the study sample’s.
- For low-prevalence settings, confirmatory testing after a positive screen is usually essential.
- Likelihood ratios avoid dependence on prevalence and combine multiplicatively with prior odds.
- Bayes’ post-test probability: $P(D \mid +) = \mathrm{LR}(+) \cdot \mathrm{Prev} / (1 + \mathrm{LR}(+) \cdot \mathrm{Prev})$ using prior odds.
- Decision curves (Vickers-Elkin) integrate PPV / NPV at different thresholds into a utility measure.

14.27.9 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.

14.27.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.28 Randomisation Methods

14.28.1 Introduction

Randomisation is the defining feature of an RCT: participants are allocated by chance rather than choice. Proper randomisation prevents selection bias (investigators cannot predict allocation) and ensures balance of known and unknown confounders in expectation. Several schemes trade simplicity, balance, and unpredictability.

14.28.2 Prerequisites

Probability; allocation concealment.

14.28.3 Theory

Simple randomisation: each participant flips a fair coin. Easy but can produce imbalance in small trials.

Block randomisation: within blocks of size B, allocate equal counts to each arm. Guarantees balance at block boundaries.

Stratified randomisation: block within strata defined by baseline covariates (sex, age, centre). Balances the stratification variables without post-hoc adjustment.

Minimisation (covariate-adaptive): allocate each new participant to minimise covariate imbalance; quasi-random, less transparent.

14.28.4 Assumptions

Allocation is concealed until the participant is enrolled; blocks / strata definitions are pre-specified.

14.28.5 R Implementation

```r
library(blockrand)

set.seed(2026)
# Block randomisation with variable block sizes (4, 6)
alloc ← blockrand(n = 60, num.levels = 2,
                  levels = c("ctrl", "trt"),
                  block.sizes = c(2, 3))
head(alloc, 10)
table(alloc$treatment)

# Stratified randomisation: stratify by sex
library(randomizeR)
pbr ← pbrPar(rb = c(4, 6), K = 2, ratio = c(1, 1))
rand_m ← genSeq(pbr, r = 1, seed = 2026)
rand_f ← genSeq(pbr, r = 1, seed = 2027)
```

14.28.6 Output & Results

Allocation sequence with approximately equal arm counts; variable block sizes prevent predictability at block boundaries.

14.28.7 Interpretation

“Block-randomisation with variable block sizes (4, 6) was used to allocate 60 participants, guaranteeing equal arm counts at every 20-participant batch. Stratification by centre ensured balance across sites.”

14.28.8 Practical Tips

- Use variable block sizes (e.g., 4 and 6) to prevent predictability; investigators guessing the next allocation defeats concealment.
- Stratify on a small number of strong prognostic variables (typically 2-3); over-stratification creates empty cells.
- Centralise the allocation schedule; local administration risks unblinding.
- Document randomisation method and concealment in the published paper and trial registry.
- For small trials, stratified block randomisation is usually the best choice.

14.28.9 For Reviewers

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

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

14.28.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR

- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.29 Crossover RCT Design

14.29.1 Introduction

A crossover randomised controlled trial assigns each participant to receive two or more treatments in sequence, separated by a washout period, so that every subject acts as their own control across the treatment comparison. The within-subject comparison removes between-subject variability — typically the dominant source of variance in clinical-pharmacology and chronic-disease studies — from the treatment estimate, giving the crossover design substantially more power than a parallel-group RCT of equal total sample size. Crossover designs are particularly common in early-phase clinical pharmacology, bioequivalence studies, sleep and migraine research, and other contexts in which the underlying condition is stable, treatment effects are reversible, and an adequate washout can eliminate pharmacological carryover.

14.29.2 Prerequisites

A working understanding of parallel-group RCT design, within-subject paired comparisons, mixed-effects models with subject as a random effect, and the concepts of period, sequence, and carryover effects.

14.29.3 Theory

The standard 2×2 crossover randomises participants to sequence AB (treatment A in period 1, B in period 2) or BA. The within-subject treatment contrast is the primary inference; analysis is by Grizzle's classical t-test on within-subject differences, or — preferably — a mixed-effects model with subject random intercept and fixed effects for period and treatment. The model is

$$y_{ijk} = \mu + \pi_j + \tau_k + s_i + \varepsilon_{ijk},$$

with period π_j, treatment τ_k, subject random intercept s_i, and residual error ε_{ijk}. **Carryover** — a residual treatment effect from period 1 lingering into period 2 — biases the within-subject estimate; it is formally tested by the sequence × period interaction but the test is under-powered, and adequate washout is the primary defence.

14.29.4 Assumptions

No carryover (washout long enough to eliminate the first-period treatment's effect), the condition is stable between periods (no progressive disease, no natural recovery), treatment effects are independent of period, and observations within each subject share a Normal distribution with constant variance.

14.29.5 R Implementation

```r
library(nlme)

set.seed(2026)
n <- 20
sequence <- sample(c("AB", "BA"), n, replace = TRUE)
subj_eff <- rnorm(n, 0, 1)

y_A <- subj_eff + rnorm(n, 0, 0.5)
y_B <- subj_eff + 0.5 + rnorm(n, 0, 0.5)

df <- data.frame(
  subject   = rep(1:n, each = 2),
  period    = rep(1:2, n),
  treatment = unlist(lapply(sequence, function(s) strsplit(s, "")[[1]])),
  sequence = rep(sequence, each = 2),
  y         = unlist(lapply(1:n, function(i)
    if (sequence[i] == "AB") c(y_A[i], y_B[i]) else c(y_B[i], y_A[i])))
)

fit <- lme(y ~ treatment + period, random = ~ 1 | subject, data = df)
summary(fit)$tTable
```

14.29.6 Output & Results

The mixed-effects fit returns the treatment effect with within-subject standard error and a separate period effect that adjusts for any drift between the two periods. The random subject intercept absorbs between-subject variation and is the source of the crossover design's power advantage; reporting the variance components alongside the fixed-effect estimate makes the design's gain explicit.

14.29.7 Interpretation

A reporting sentence: "The two-period crossover analysis with mixed-effects modelling estimated the B–A treatment difference as 0.48 (95 % CI 0.22 to 0.74, $p = 0.002$), achieving over three-fold more precision than an equivalent parallel-group design with the same number of subjects. The period effect was small and non-significant ($p = 0.51$), and the sequence × period interaction (carryover diagnostic) was non-significant ($p = 0.78$), supporting the no-carryover assumption. Reporting follows the CONSORT extension for crossover trials." Always report period and carryover.

14.29.8 Practical Tips

- Test the carryover hypothesis formally via the sequence × period interaction, but rely on design — an adequately long washout, conventionally at least five half-lives of the active compound — as the primary defence rather than the underpowered post-hoc test.
- Unbalanced sequences (very different counts of AB and BA participants) reduce design efficiency and complicate analysis; aim for sequence balance via stratified randomisation on sequence.

- More than two periods (Latin-square or Williams designs) improve power and allow comparison of more than two treatments, at the cost of complexity, longer trial duration, and more potential for dropout — which crossover designs handle poorly because dropouts lose paired information.
- If the underlying condition evolves substantially within the trial timeframe (progressive disease, recovery, growth), the crossover design is inappropriate; the stability assumption is hard to defend and biases the estimate.
- Report per the CONSORT extension for crossover trials, including the trial flow diagram (per period), the sequence allocation, washout duration, and the carryover diagnostic.
- For ordinal or binary outcomes in a crossover design, generalised mixed-effects models (`glmer`) or paired analyses on the within-subject contingency table (McNemar) are the appropriate analysis approaches.

14.29.9 R Packages Used

`nlme::lme()` and `lme4::lmer()` for mixed-effects analysis with subject random intercepts; `Crossover` for canonical crossover-design construction including Williams squares and higher-order designs; `crossdes` for systematic generation of balanced crossover layouts; `bear` for end-to-end bioequivalence analysis on crossover data; `Mediana` for trial-design simulation including crossover designs.

14.29.10 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.

14.29.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.30 Parallel-Group RCT Design

14.30.1 Introduction

The parallel-group randomised controlled trial (RCT) is the gold standard for evaluating interventions. Participants are randomly allocated to one of two (or more) arms – typically a new intervention vs control – and followed for a pre-specified outcome. Randomisation balances known and unknown confounders in expectation; blinding further reduces bias.

14.30.2 Prerequisites

Randomisation; hypothesis testing; sample-size calculation.

14.30.3 Theory

Essential elements: - **Primary outcome** (continuous, binary, time-to-event) with a clinically meaningful effect size. - **Allocation ratio** (usually 1:1, occasionally 2:1 or 3:1 for rare interventions). - **Randomisation list** (pre-generated, concealed at allocation time). - **Blinding** (single, double, triple). - **Pre-registered statistical analysis plan**.

Primary analysis is typically intent-to-treat (ITT), comparing groups as randomised.

14.30.4 Assumptions

Participants are exchangeable post-randomisation; allocation is fully concealed; outcome assessment is blinded; follow-up is complete or MAR.

14.30.5 R Implementation

```
library(pwr)

# Sample-size for two-arm comparison of means
# Effect: difference of 0.5 SD, alpha = 0.05, power = 0.80
pwr.t.test(d = 0.5, sig.level = 0.05, power = 0.80,
           type = "two.sample", alternative = "two.sided")

# Simulate an RCT with continuous outcome
set.seed(2026)
n_per_arm ← 64
arm ← rep(c("ctrl", "trt"), each = n_per_arm)
y   ← rnorm(2 * n_per_arm, mean = ifelse(arm == "trt", 0.5, 0))

# ITT analysis: two-sample t-test
t.test(y ~ arm, var.equal = TRUE)

# Adjusted analysis with a pre-specified covariate (ANCOVA)
covar ← rnorm(2 * n_per_arm)
summary(lm(y ~ arm + covar))$coefficients
```

14.30.6 Output & Results

Sample size ~64 per arm for 80 % power; ITT t-test recovers the effect; ANCOVA-adjusted analysis gives a similar point estimate with smaller SE when the covariate is prognostic.

14.30.7 Interpretation

“The trial randomised 128 participants 1:1 to intervention vs control; the intervention arm showed a 0.52 SD improvement (95 % CI 0.18-0.86, p = 0.003) on the primary outcome, analysed by ANCOVA adjusting for baseline value.”

14.30.8 Practical Tips

- Register the protocol and SAP **before** enrolment (clinicaltrials.gov, EUDRACT).
- Pre-specify the primary outcome and analysis; secondary outcomes are exploratory.
- Blinding is protective; document how it was broken (unblinding events, assessment).
- Report per CONSORT 2010 guidelines; flow diagram is mandatory.
- ITT is the primary analysis; supportive per-protocol analyses are sensitivity.

14.30.9 For Reviewers

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

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

14.30.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.31 Reliability and Cronbach's Alpha

14.31.1 Introduction

Cronbach's alpha, introduced by Lee Cronbach in 1951, summarises the internal consistency of a multi-item scale by quantifying how strongly the items co-vary after accounting for the total number of items. The intuition is that items intended to measure the same underlying construct should correlate with each other; alpha rises with the average inter-item correlation and with the number of items. Cronbach's alpha is now ubiquitous in questionnaire validation, patient-reported-outcome (PRO) instrument development, psychometric evaluation of clinical scales, and any setting where a composite score is formed from multiple ordinal or continuous items. Despite well-known statistical limitations, it remains the single most reported reliability statistic in the clinical-research literature.

14.31.2 Prerequisites

A working understanding of classical test theory, the concept of a true score plus measurement error, and the construction of composite scores from multi-item rating scales.

14.31.3 Theory

Cronbach's alpha is

$$\alpha = \frac{k}{k-1}\left(1 - \frac{\sum_{i=1}^{k}\sigma_i^2}{\sigma_T^2}\right),$$

with k the number of items, σ_i^2 the variance of item i, and σ_T^2 the variance of the total summed score. The statistic ranges from $-\infty$ (theoretically; in practice 0) to 1. Conventional thresholds are 0.70 (acceptable), 0.80 (good), and 0.90 (excellent), with values above 0.95 typically indicating redundant items rather than superior reliability. McDonald's omega is a more flexible alternative when the assumption of tau-equivalence (equal true-score variances across items) is doubtful.

14.31.4 Assumptions

Items are tau-equivalent (each item measures the same true construct with the same loading), the scale is unidimensional (a single underlying factor explains the inter-item covariance structure), and items are continuous or quasi-continuous (Likert with at least 5 levels). Violations are common, and McDonald's omega or hierarchical-omega estimators give a more honest reliability estimate when the assumptions are not met.

14.31.5 R Implementation

```
library(psych)

set.seed(2026)
n ← 200
theta ← rnorm(n)
items ← sapply(1:8, function(i) 0.7 * theta + rnorm(n, 0, 0.7))
colnames(items) ← paste0("q", 1:8)

alpha_res ← psych::alpha(items)
alpha_res$total
alpha_res$alpha.drop[, c("raw_alpha", "std.alpha")]
```

14.31.6 Output & Results

`psych::alpha()` returns the raw and standardised Cronbach's alpha for the full scale, plus an "alpha drop" table showing how alpha would change if each item were removed. A large positive drop (alpha increases without an item) flags that item as inconsistent with the rest of the scale and a candidate for revision or removal. The output also includes 95 % confidence intervals (via Feldt's method) and the average inter-item correlation, which is often more informative than alpha itself.

14.31.7 Interpretation

A reporting sentence: "The eight-item PRO scale had Cronbach's alpha 0.82 (95 % CI 0.78 to 0.86, Feldt method), indicating good internal consistency. The average inter-item correlation was 0.36, supporting the tau-equivalence assumption qualitatively. No single item substantially changed alpha when dropped (largest drop -0.01), so all items were retained for the final scale. McDonald's omega-total was 0.83, in close agreement with alpha." Always report both alpha and omega when feasible.

14.31.8 Practical Tips

- Alpha depends on the number of items: long scales inflate alpha mechanically, even when items are only weakly inter-correlated. Reporting the average inter-item correlation alongside alpha gives readers a fairer picture of true item coherence.
- Low alpha (< 0.70) may reflect a multi-dimensional scale rather than unreliable items. Always check the factor structure with exploratory or confirmatory factor analysis before concluding that items are unreliable.
- Very high alpha (> 0.95) suggests redundant items measuring essentially the same content; consider trimming the scale by removing the most redundant items, which improves administrative efficiency without sacrificing reliability.
- For ordinal items with fewer than five categories, the standard Pearson-correlation-based alpha is biased downward; use ordinal alpha (computed from polychoric correlations) or McDonald's omega instead.
- Pair Cronbach's alpha with confirmatory factor analysis (CFA) to verify the assumed unidimensional structure; alpha computed on a multidimensional scale is uninterpretable as a reliability statistic.
- For test-retest reliability and inter-rater reliability, the appropriate statistics are the intraclass correlation coefficient (ICC) and Cohen's or Fleiss's kappa, respectively; alpha measures only internal consistency, not stability or agreement.

14.31.9 R Packages Used

`psych::alpha()` for the canonical Cronbach's alpha with drop analysis, `psych::omega()` for McDonald's omega and hierarchical-omega; `ltm::cronbach.alpha()` as a lightweight alternative; `MBESS::ci.reliability()` for advanced CIs (Feldt, bootstrap, Bonett); `lavaan` for confirmatory factor analysis to verify the unidimensional assumption.

14.31.10 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.

14.31.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.32 ROC Analysis

14.32.1 Introduction

The receiver-operating-characteristic (ROC) curve traces sensitivity against $1 - $ specificity across all possible decision thresholds of a continuous-valued diagnostic test or risk score. Where a single sensitivity-specificity pair characterises performance at one cut-off, the ROC curve summarises performance across every cut-off and therefore captures the discriminative ability of the underlying continuous measurement independently of any chosen threshold. The area under the curve (AUC) reduces this curve to a single number with a clean probabilistic interpretation: it equals the probability that a randomly chosen diseased case has a higher biomarker value than a randomly chosen non-diseased case. AUC values of 0.5 indicate chance-level discrimination and 1.0 indicates perfect separation; values between 0.7 and 0.9 are typical for clinically useful biomarkers.

14.32.2 Prerequisites

A working understanding of sensitivity and specificity, the role of decision thresholds in diagnostic-test performance, and the trade-off between true and false positive rates that the ROC curve makes explicit.

14.32.3 Theory

For a continuous biomarker X and binary disease status D, the ROC curve is the parametric plot

$$\mathrm{ROC}(c) = \big(1 - F_0(c),\, 1 - F_1(c)\big) \quad \text{for all } c,$$

where F_0 and F_1 are the CDFs of X in non-diseased and diseased populations, respectively. The AUC has the equivalent representation

$$\mathrm{AUC} = P(X_1 > X_0),$$

with X_1 a random observation from the diseased and X_0 from the non-diseased population — the probability of correct ranking. Confidence intervals on AUC are typically computed via the DeLong non-parametric method or by bootstrap; bootstrap is also the standard approach for inference on the curve itself.

14.32.4 Assumptions

Test values are measured on a continuous scale (or at least an ordinal scale with many levels), disease status is correctly classified by a reliable gold-standard reference, and observations are independent. The interpretation as a discrimination measure is independent of disease prevalence — a feature of AUC that distinguishes it from predictive values.

14.32.5 R Implementation

```r
library(pROC)

set.seed(2026)
n <- 200
disease <- factor(sample(c(0, 1), n, replace = TRUE, prob = c(0.6, 0.4)))
biomarker <- rnorm(n, mean = ifelse(disease == 1, 1.0, 0), sd = 1)

roc_obj <- roc(response = disease, predictor = biomarker,
               levels = c("0", "1"), direction = "<")
auc(roc_obj)
ci.auc(roc_obj, method = "delong")

plot(roc_obj, col = "#2A9D8F", lwd = 2, legacy.axes = TRUE,
     main = "ROC curve for biomarker")
abline(0, 1, lty = 2, col = "grey60")
```

14.32.6 Output & Results

`roc()` constructs the ROC object; `auc()` returns the area under the curve and `ci.auc()` provides the DeLong or bootstrap confidence interval. The standard plot shows sensitivity on the vertical axis and $1 - $ specificity on the horizontal axis, with the chance-diagonal as a reference line. Points along the curve correspond to sensitivity-specificity trade-offs at different cut-off values.

14.32.7 Interpretation

A reporting sentence: “The biomarker showed good discrimination for the binary disease outcome with AUC 0.77 (95 % CI 0.70 to 0.84, DeLong method); this exceeds the conventional ‘fair’ threshold of 0.70 but falls short of ‘excellent’ (≥ 0.90). At the Youden-optimal cut-off (biomarker ≥ 0.42), sensitivity was 73 % and specificity 70 %; alternative cut-offs prioritising specificity (e.g., ≥ 1.0, sensitivity 51 %, specificity 87 %) may be preferred for screening applications.” Always report both AUC and at least one operating point.

14.32.8 Practical Tips

- Report AUC with a 95 % confidence interval — DeLong’s non-parametric method is the standard for paired comparisons and bootstrap is preferable for very small or imbalanced samples; AUC without uncertainty bounds is uninterpretable.
- Test AUC against 0.5 (chance) using `pROC::roc.test()`; compare two AUCs from the same cases using a paired DeLong test, which respects the correlation induced by shared subjects.
- Partial AUC over a clinically relevant region — for example, AUC restricted to specificity above 0.9 in a screening context — is often more informative than full AUC, because clinical use rarely spans the full operating range.
- AUC is threshold-independent; combine it with a calibration analysis (Hosmer-Lemeshow, calibration intercept and slope, calibration plot) or a decision-curve analysis when threshold-dependent decisions matter for the clinical context.

- For heavily imbalanced data (rare disease, screening contexts), precision-recall AUC is often more informative than ROC AUC because the ROC can look deceptively good when most subjects are non-diseased; PRAUC focuses on the positive class.
- When comparing biomarkers, include the increment in AUC (Δ AUC), the integrated discrimination index (IDI), and the net reclassification improvement (NRI); each captures a different facet of incremental performance.

14.32.9 R Packages Used

`pROC` for canonical ROC analysis with DeLong and bootstrap CIs, partial AUC, and paired comparisons; `ROCR` for an alternative interface with comprehensive performance-measure support; `PRROC` for precision-recall AUC and area-under-the-PR-curve analyses; `cutpointr` for principled threshold selection; `rms::lrm()` for AUC reporting integrated with logistic-regression model evaluation.

14.32.10 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.

14.32.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.33 Sample Size Re-Estimation

14.33.1 Introduction

Sample-size re-estimation (SSR) is an adaptive-design technique that updates a clinical trial's planned sample size mid-study using interim estimates of nuisance parameters such as the within-group standard deviation (continuous outcomes) or the control-arm event rate (binary outcomes). It is one of the most common and least controversial trial adaptations because, when done in the blinded form, it does not access treatment-effect information and therefore preserves the type-I error rate under mild and well-understood conditions. SSR is now standard practice when pilot-data estimates of nuisance parameters are uncertain at the planning stage and the trial is otherwise large enough that the consequences of an under- or over-estimated nuisance parameter would be substantial.

14.33.2 Prerequisites

A working understanding of sample-size calculation, adaptive trial design, the distinction between blinded and unblinded interim analyses, and the regulatory framework around protocol-pre-specified adaptations.

14.33.3 Theory

Blinded SSR uses interim estimates of nuisance parameters from pooled (across-arm) data only, without revealing any arm-level information. For continuous outcomes the pooled standard deviation suffices; for binary outcomes, the pooled event rate. Blinded SSR does not inflate type-I error under standard conditions and is widely accepted by regulators with minimal formal control.

Unblinded SSR lets a Data Monitoring Committee see interim results by arm, supporting more flexible re-estimation rules at the cost of formal multiplicity control. The standard implementation is a combination-test or promising-zone design (Mehta and Pocock, 2011) that preserves conditional type-I error through explicit weighting of the interim and final test statistics.

14.33.4 Assumptions

The relevant nuisance parameter is unknown at planning but estimable from interim data; the interim analysis preserves blinding where required; the SSR rule is pre-specified in the protocol and statistical analysis plan; and a maximum sample size cap is set in advance to avoid unlimited re-estimation.

14.33.5 R Implementation

```
library(rpact)

n_initial ← ceiling(
  getSampleSizeMeans(alternative = 0.3, stDev = 1,
                     alpha = 0.025, beta = 0.2,
                     groups = 2)$numberOfSubjects
)
n_initial

n_updated ← ceiling(
  getSampleSizeMeans(alternative = 0.3, stDev = 1.4,
                     alpha = 0.025, beta = 0.2,
                     groups = 2)$numberOfSubjects
)
n_updated

c(planned = n_initial, revised = n_updated)
```

14.33.6 Output & Results

The script computes the planned sample size given the protocol-assumed standard deviation and the revised sample size given the interim-observed standard deviation. The ratio of the two ($1.4^2 = 1.96$) drives the proportional increase in required sample size — a

familiar “variance is squared in the sample-size formula” relationship that makes SSR especially valuable when the within-group SD was uncertain at planning.

14.33.7 Interpretation

A reporting sentence: “A pre-specified blinded sample-size re-estimation at 50 % information accrual revealed a pooled within-group SD of 1.38, compared with the protocol-assumed 1.00. Per the pre-specified rule, the sample size was increased from 350 to 680 to maintain 80 % power against the originally specified treatment effect of 0.3 SD. The increase did not access treatment-arm information and therefore did not inflate the type-I error rate. The blinded SSR was conducted by an unblinded statistician within the DMC operating manual.” Always report the blinding status and the cap.

14.33.8 Practical Tips

- Use blinded SSR whenever possible; it is operationally simpler, less controversial with regulators, and adequate for the most common SSR application (revising the within-group SD or pooled event rate).
- Unblinded SSR requires a statistical method that explicitly preserves the type-I error rate — typically a combination-test design (Cui-Hung-Wang or Mehta-Pocock promising-zone) that weights the interim and final test statistics in a pre-specified way.
- Pre-specify the SSR trigger condition, the re-estimation rule, and the maximum sample-size cap in the protocol and SAP; unlimited re-estimation without a cap is not acceptable to regulators and creates an open-ended commitment that sponsors rarely want to make.
- Document the SSR decision rationale in the final trial report — whether the SSR triggered, what the interim nuisance-parameter estimate was, and what the revised sample size became — so reviewers can assess the decision.
- SSR is especially valuable when pilot-data estimates of nuisance parameters are uncertain or when the trial population is expected to differ from the pilot in ways that affect variance or event rate; reliable prior data make SSR less necessary.
- Combine SSR with group-sequential efficacy and futility boundaries for a fully adaptive design that handles both nuisance-parameter uncertainty and effect-size revision; the combination is now standard in many phase-3 trials.

14.33.9 R Packages Used

`rpact` for canonical adaptive-design analysis including blinded and unblinded SSR with built-in type-I error control; `gsDesign` and `adaptTest` for combination-test SSR with promising-zone analysis; `Mediana` for trial-design simulation including SSR strategies; `RPACT::getDataset()` for stage-data integration in re-estimation workflows; `Hmisc` and `pwr` for the underlying classical sample-size formulas.

14.33.10 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.

14.33.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.34 Sensitivity Analyses in Clinical Trials

14.34.1 Introduction

Sensitivity analyses complement the primary analysis by varying key assumptions – missing-data mechanism, model form, inclusion criteria – to assess how robust conclusions are. Tipping-point analyses identify how extreme assumptions must be before the primary conclusion flips, providing a concrete interpretation.

14.34.2 Prerequisites

Primary analysis; missing-data mechanisms; multiple imputation.

14.34.3 Theory

Tipping-point analysis under MI: impute missing outcomes in the experimental arm with an increasing shift δ (less favourable); re-pool. The smallest δ that makes the treatment effect no longer significant is the "tipping point". A large tipping point means the conclusion is robust.

Other sensitivity analyses: different imputation methods, PP vs ITT, different covariate adjustments, varying inclusion criteria, alternative parametric models.

14.34.4 Assumptions

Sensitivity analyses are pre-specified; tipping points are interpreted clinically, not mechanically.

14.34.5 R Implementation

```r
library(mice)

set.seed(2026)
n <- 200
arm <- factor(rep(c("ctrl", "trt"), each = n/2))
baseline <- rnorm(n, 5, 1)
outcome  <- 0.6 * baseline + ifelse(arm == "trt", 0.8, 0) +
            rnorm(n, 0, 1)
outcome[sample(n, 40)] <- NA

df <- data.frame(arm, baseline, outcome)

# Base analysis under MAR
imp <- mice(df, m = 20, method = "pmm", printFlag = FALSE)
summary(pool(with(imp, lm(outcome ~ arm + baseline))))$estimate[2]

# Tipping-point analysis: penalise imputed trt-arm outcomes by delta
deltas <- seq(0, 2, by = 0.25)
effs <- sapply(deltas, function(d) {
  imp2 <- imp
  for (k in 1:20) {
    idx <- which(is.na(df$outcome) & df$arm == "trt")
    imp2$imp$outcome[[k]][df$arm[idx] == "trt"] <-
      imp2$imp$outcome[[k]][df$arm[idx] == "trt"] - d
  }
  summary(pool(with(imp2, lm(outcome ~ arm + baseline))))$estimate[2]
})
data.frame(delta = deltas, trt_effect = round(effs, 3))
```

14.34.6 Output & Results

Treatment effect declines linearly with δ; the tipping point is where the effect crosses zero (or loses significance).

14.34.7 Interpretation

“A delta shift of 1.6 on the imputed intervention-arm outcomes was required to eliminate significance; clinically this would require intervention dropouts to fare 1.6 SD worse than MAR predicts. The primary conclusion is robust to plausible MNAR mechanisms.”

14.34.8 Practical Tips

- Pre-specify all sensitivity analyses in the SAP; post-hoc analyses are exploratory.
- Tipping points are most informative when expressed on the clinical scale.
- ICH E9 R1 promotes sensitivity-analysis thinking tied to the estimand.
- Running many sensitivity analyses is fine; interpret them holistically.
- Reporting a tipping-point figure in the primary paper is increasingly standard for high-missingness trials.

14.34.9 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.

14.34.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.35 Stepped-Wedge Trial

14.35.1 Introduction

Stepped-wedge cluster-randomised trials (SW-CRTs) randomise not whether each cluster receives the intervention but the time at which each cluster transitions from control to intervention. By the end of the trial all clusters have received the intervention, which gives the design particular ethical and pragmatic appeal: it is well suited to settings where investigators believe the intervention is likely beneficial (so withholding it from some clusters indefinitely would be ethically uncomfortable), where logistical constraints prevent simultaneous rollout across all clusters, or where a programme rollout is happening anyway and the trial is exploiting the staggered implementation to learn the effect. Analysis must carefully separate the secular time trend that affects all clusters from the treatment effect that is realised at different calendar times in different clusters.

14.35.2 Prerequisites

A working understanding of cluster-randomised trials, time-trend modelling, and mixed-effects models with cluster random intercepts and (optionally) cluster-time random effects.

14.35.3 Theory

Clusters are randomised to sequence rather than to arm; at each step (period) a new subset of clusters crosses over from control to intervention. The resulting data structure provides two complementary contrasts: a between-cluster comparison at each period (like a parallel-arm CRT at that period) and a within-cluster before-after comparison around each cluster's switch-point. The standard analysis is a mixed-effects model with fixed effects for time period and treatment status and a random intercept for cluster:

$$y_{ijk} = \mu + \tau_t + \beta \cdot X_{jk} + u_j + \varepsilon_{ijk},$$

with τ_t the period effect, X_{jk} the treatment indicator for cluster j at time k, and $u_j \sim N(0, \sigma_c^2)$ the cluster random effect. Including the period fixed effects is mandatory because secular trends confound the treatment estimate.

14.35.4 Assumptions

Secular time trends are common across clusters (any cluster-specific time trend should be modelled explicitly), the treatment effect is immediate and stable after switch (or any fade-in/fade-out is explicitly modelled), and the within-cluster correlation structure is correctly specified.

14.35.5 R Implementation

```r
library(lme4); library(lmerTest)

set.seed(2026)
n_cl ← 10; n_per ← 20; n_period ← 5
cluster ← rep(1:n_cl, each = n_period * n_per)
period  ← rep(rep(1:n_period, each = n_per), n_cl)

start_t ← sample(2:5, n_cl, replace = TRUE)
trt ← as.numeric(period ⩾ rep(start_t, each = n_period * n_per))

time_trend ← 0.1 * (period - 1)
cl_re      ← rep(rnorm(n_cl, 0, 0.5), each = n_period * n_per)

y ← cl_re + time_trend + 0.4 * trt +
     rnorm(length(cluster), 0, 1)

df ← data.frame(cluster = factor(cluster),
                period  = factor(period),
                trt     = trt, y = y)

fit ← lmer(y ~ trt + period + (1 | cluster), data = df)
summary(fit)$coefficients["trt", ]
```

14.35.6 Output & Results

The mixed-effects model returns the treatment effect estimate with a standard error that reflects both the within-cluster and between-cluster information available in the staggered design. Including the period fixed effects absorbs the secular time trend; the random cluster intercept absorbs the between-cluster baseline variation; the residual captures within-cluster, within-period noise.

14.35.7 Interpretation

A reporting sentence: “The stepped-wedge mixed-effects analysis estimated a treatment effect of 0.38 SD (95 % CI 0.19 to 0.57, $p < 0.001$), adjusting for the calendar-period secular trend (which itself was significant, $\hat{\tau}_5 - \hat{\tau}_1 = 0.41$) and the cluster random intercepts (cluster ICC 0.20). Reporting follows the CONSORT extension for stepped-wedge CRTs.” Always report both the secular trend and the treatment effect.

14.35.8 Practical Tips

- Always adjust for time period in the analysis; an unadjusted analysis confounds treatment with secular trend, and the bias can be substantial in any health-system

setting where outcomes are improving (or worsening) over time independent of the intervention.
- Report per the CONSORT extension for stepped-wedge cluster-randomised trials (Hemming et al., 2018), which specifies the trial design figure, time-by-cluster matrix, and standard reporting requirements.
- Consider modelling time-varying treatment effects (a ramp-up over a few periods after the switch) for interventions that take time to implement fully; assuming an immediate stable effect when the intervention requires phased rollout biases the estimate downward.
- Sample-size calculation for stepped-wedge designs is intrinsically more complex than for parallel CRTs because it depends on the design matrix, the within-cluster correlation, and the number of steps; use `swCRTdesign::swPwr()` or the Hussey-Hughes (2007) closed-form formula, and avoid naive parallel-CRT power approximations.
- The ethical advantage — all clusters eventually receive the intervention — is real but does not eliminate the need for equipoise; if investigators are confident the intervention works, the trial is arguably unnecessary regardless of design.
- Sensitivity analyses to the assumed correlation structure (compound symmetry vs Hooper-Girling vs more complex) are increasingly required by reviewers; report several specifications and check that conclusions are robust.

14.35.9 R Packages Used

`lme4::lmer()` and `lmerTest` for canonical stepped-wedge mixed-effects analysis; `swCRTdesign::swPwr()` and `swCRTdesign::swSummary()` for design and power calculation; `clusterPower` for general cluster-trial power including stepped wedge; `geepack::geeglm()` for GEE-based marginal-model analysis as an alternative; `glmmTMB` for stepped-wedge analyses with non-Normal outcomes (count, binary).

14.35.10 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.

14.35.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.36 Stratified Randomisation

14.36.1 Introduction

Stratified randomisation runs a separate block-randomisation list within each stratum defined by one or more baseline covariates — typically centre, sex, age category, disease severity, or other strong prognostic factors. By randomising within strata, the design guarantees balance of the stratification variables across treatment arms, which stabilises subgroup inference, pre-empts the situation in which a strongly prognostic covariate drives apparent arm differences, and is increasingly required by regulators and journals for any multi-centre or prognostically heterogeneous trial. The trade-off is a slight increase in implementation complexity and the risk of empty strata in small trials, both manageable with care.

14.36.2 Prerequisites

A working understanding of simple and block randomisation, the role of baseline covariates as potential confounders or effect modifiers, and the analytical principle that the design should be reflected in the analysis model.

14.36.3 Theory

Strata are defined by a cross-tabulation of one or more pre-specified factors — typically 4 to 8 strata total in a real trial, formed by combining centre with one or two prognostic factors. Within each stratum, block randomisation proceeds independently with its own variable-sized blocks, ensuring that arm counts are balanced both globally and within every stratum at every block boundary. The method trades a small amount of design simplicity for guaranteed marginal balance on the stratification variables and substantial protection against centre-by-treatment confounding in multi-centre trials.

14.36.4 Assumptions

Stratification variables are known and recorded before randomisation (post-hoc stratification is not stratified randomisation but rather post-hoc adjustment), the strata are clinically meaningful and prognostically important, and the trial is large enough that no stratum will end up with too few subjects to support stable within-stratum analysis.

14.36.5 R Implementation

```
library(blockrand)

set.seed(2026)

strata <- c("A", "B", "C")
schedule <- do.call(rbind, lapply(strata, function(s) {
  b <- blockrand(n = 20, num.levels = 2,
                 levels = c("ctrl", "trt"),
                 block.sizes = c(2, 3))
  b$centre <- s
  b
}))

table(schedule$centre, schedule$treatment)
head(schedule, 8)
```

14.36.6 Output & Results

The script generates three centre-specific allocation schedules and combines them into a master list. The cross-tabulation of centre by treatment shows equal arm counts within each centre — the design's signature property — and the master list is then exported to the trial's interactive web-response system for execution.

14.36.7 Interpretation

A reporting sentence: "Treatment allocation was stratified by centre (three sites) and by baseline disease severity (mild, moderate, severe), with variable-sized permuted blocks of 2 and 3 within each of the six strata. This guaranteed equal arm allocation at every centre and within every severity stratum, preventing centre-by-treatment and severity-by-treatment confounding. Final arm counts were exactly balanced within every stratum (50 patients per arm per stratum)." Always describe the stratification scheme.

14.36.8 Practical Tips

- Stratify on the one to three strongest prognostic variables, and not more; more strata mean smaller stratum sizes, more empty cells (especially in small trials), and progressively diminishing protection against the very imbalance the stratification was meant to prevent.
- Centre is the standard stratification variable for multi-centre trials and is virtually always recommended; centre-specific outcome differences are common and centre-by-treatment confounding can substantially bias the overall estimate.
- Always analyse with the stratification variables as covariates in the analysis model; stratified randomisation by itself does not produce the correct standard error if the analysis ignores the stratification — analysing as if the trial were simple-randomised understates the precision of the estimate.
- Do not stratify on a variable you intend to adjust for analytically without that variable being prognostic — redundant stratification dilutes randomisation entropy without analytic benefit. Conversely, every variable used for stratification should also enter the analysis as a covariate.

- For small trials (typically fewer than 100 subjects total) where stratification on multiple factors would create empty strata, minimisation (Pocock-Simon) is a compromise that balances multiple covariates without forcing strict block structure.
- The trial's interactive web-response system (IWRS) handles the multi-stratum allocation in real time; running stratified randomisation by hand in a multi-centre trial is operationally fragile and a frequent source of allocation-concealment failures.

14.36.9 R Packages Used

`blockrand` for canonical stratified block randomisation with built-in stratum looping; `randomizr` for tidyverse-friendly stratified randomisation with explicit unit-of-randomisation control; `Minirand::Pocock` for minimisation as an alternative when stratum cells would be too sparse; `bcrm` and related packages for biased-coin variants; `Mediana` for trial-design simulation including stratified-randomisation strategies.

14.36.10 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.

14.36.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.37 Subgroup Analyses

14.37.1 Introduction

Subgroup analyses in clinical trials assess whether the overall treatment effect varies across pre-specified baseline characteristics — age, sex, disease severity, comorbidity, biomarker status. They provide important insight into treatment heterogeneity, support guideline development, and inform precision-medicine decisions. They are also notoriously prone to over-interpretation: with enough subgroups and enough cut-points, false-positive heterogeneity findings appear by chance alone, and the literature is replete with cautionary tales of subgroup claims that failed to replicate. Modern guidance (CONSORT, ICH-E9, regulatory subgroup analysis frameworks) emphasises pre-specification, formal interaction tests, and graphical communication via forest plots, while warning against per-subgroup hypothesis testing as a substitute for the interaction test.

14.37.2 Prerequisites

A working understanding of treatment effect estimation in randomised trials, interaction terms in regression, the multiple-testing problem, and the regulatory framework around pre-specified versus post-hoc analyses.

14.37.3 Theory

The statistically appropriate test for effect heterogeneity is the **treatment-by-subgroup interaction** in a regression of the outcome on treatment, subgroup, and their product. The standard reporting set includes the overall treatment effect with its 95 % CI, the subgroup-specific effects with CIs, and the interaction p-value. Comparing within-subgroup p-values across strata (the "significant in one, not the other" fallacy) is statistically incorrect because each within-subgroup test is under-powered and the comparison ignores the multiplicity.

Pre-specification is the key safeguard: a small number of biologically-motivated subgroups documented in the SAP carry interpretable evidentiary weight, while post-hoc subgroup discovery is at best hypothesis-generating and at worst misleading.

14.37.4 Assumptions

The subgroups are pre-specified in the protocol or SAP, the subgrouping covariates are measured at baseline rather than on-treatment (avoiding immortal-time bias and other post-randomisation issues), and the trial is large enough that the interaction test has at least minimal power — usually not the case in practice, which is why subgroup tests rarely reach significance.

14.37.5 R Implementation

```r
set.seed(2026)
n <- 400
arm <- factor(rep(c("ctrl", "trt"), each = n/2))
sex <- factor(sample(c("M", "F"), n, replace = TRUE))

y <- ifelse(arm == "trt",
            ifelse(sex == "F", 1.0, 0.3), 0) +
     rnorm(n)

fit <- lm(y ~ arm * sex)
summary(fit)$coefficients

by(data.frame(y, arm), sex, function(df) {
  t.test(y ~ arm, data = df)$estimate
})
```

14.37.6 Output & Results

The interaction term in the regression model is the formal test of effect modification by subgroup; the per-subgroup t-tests give the subgroup-specific point estimates that populate the forest plot. Reporting both — interaction p-value plus subgroup-specific

estimates with CIs — is the standard expected by trial reporting guidelines.

14.37.7 Interpretation

A reporting sentence: "The overall treatment effect was 0.65 (95 % CI 0.45 to 0.85, $p < 0.001$); the pre-specified treatment-by-sex interaction was significant ($p = 0.02$), with the effect in women (0.98, 95 % CI 0.69 to 1.27) approximately three-fold larger than in men (0.32, 95 % CI 0.04 to 0.61). This sex-by-treatment heterogeneity was hypothesised in the protocol on the basis of pharmacokinetic differences and is reported here as a confirmatory rather than exploratory finding; replication in an independent trial is desirable." Always state pre-specification status.

14.37.8 Practical Tips

- Pre-specify all subgroup analyses in the protocol and SAP; post-hoc subgroups are exploratory at best and should be flagged as such in any reporting, ideally in a separate section labelled "exploratory."
- Always interpret the interaction p-value as the test of heterogeneity, not the per-subgroup p-values; the per-subgroup tests are nearly always under-powered, and comparing their significance across subgroups is a well-known statistical fallacy.
- Forest plots are the standard way to communicate subgroup effects visually; they make magnitudes and uncertainties immediately legible and are now expected by most clinical-trial reporting guidelines.
- Limit pre-specified subgroups to four to six biologically motivated factors; a list of 20+ subgroups is a fishing expedition that nearly guarantees at least one false-positive interaction by chance, and reviewers will flag it.
- Heterogeneity-of-treatment-effect (HTE) methods — causal forests, BART, model-based recursive partitioning, the SIDES algorithm — are emerging for principled data-driven subgroup discovery, with appropriate multiplicity control built in. They are increasingly accepted as exploratory complements to traditional pre-specified subgroup analyses.
- For survival or time-to-event subgroup analyses, fit a Cox model with treatment, subgroup, and treatment × subgroup terms; the per-subgroup hazard ratios should be reported with 95 % CIs alongside the joint interaction test.

14.37.9 R Packages Used

Base R `lm()`, `glm()`, and `t.test()` for canonical subgroup analysis; `survival::coxph()` with interaction terms for survival subgroup analyses; `forestplot` and `forester` for publication-quality subgroup forest plots; `grf` for generalised random forests with treatment-effect estimation; `SIDES` and `model4you` for principled exploratory subgroup discovery.

14.37.10 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.

14.37.11 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.38 Weighted Kappa

14.38.1 Introduction

Weighted kappa extends Cohen's kappa to ordinal data by crediting partial agreement: near-miss disagreements (mild vs moderate) weigh less than dramatic disagreements (mild vs severe). It is the standard inter-rater agreement statistic for Likert-type scales, radiology grading, and symptom-severity ratings.

14.38.2 Prerequisites

Cohen's kappa; ordinal measurement.

14.38.3 Theory

$$\kappa_w = 1 - \frac{\sum_{ij} w_{ij} f_{ij}}{\sum_{ij} w_{ij} e_{ij}},$$

where w_{ij} is the disagreement weight between category i and j, f_{ij} observed cell frequency, e_{ij} expected under chance.

Weight schemes: - **Linear** $w_{ij} = |i - j|/(k - 1)$ for k categories. - **Quadratic** $w_{ij} = (i - j)^2/(k - 1)^2$ – more forgiving of near-miss disagreements.

Quadratic is most common for multi-category ordinal scales.

14.38.4 Assumptions

Category ordering is meaningful and equally spaced; two raters; independent ratings.

14.38.5 R Implementation

```r
library(psych)

set.seed(2026)
n <- 100
# Two raters on a 5-point ordinal scale
rater1 <- sample(1:5, n, replace = TRUE,
                 prob = c(0.1, 0.2, 0.4, 0.2, 0.1))
# Rater 2 agrees within +-1 with prob 0.8, otherwise random
rater2 <- ifelse(rbinom(n, 1, 0.8) == 1,
                 pmax(1, pmin(5, rater1 + sample(-1:1, n, replace = TRUE))),
                 sample(1:5, n, replace = TRUE))

cohen.kappa(cbind(rater1, rater2))$kappa    # unweighted
cohen.kappa(cbind(rater1, rater2),
            w = "squared")$weighted.kappa
cohen.kappa(cbind(rater1, rater2),
            w = "linear")$weighted.kappa
```

14.38.6 Output & Results

Unweighted kappa (~0.35), linear-weighted (~0.55), quadratic-weighted (~0.70); quadratic weights reward near-miss agreement more heavily.

14.38.7 Interpretation

“Weighted kappa with quadratic weights was 0.72 (95 % CI 0.62-0.82), consistent with substantial agreement; unweighted kappa of 0.35 understates agreement by not crediting near-miss ratings.”

14.38.8 Practical Tips

- Use quadratic weights for most clinical ordinal scales; they reflect that 1-step disagreements matter far less than 3-step.
- Linear weights are appropriate when category spacing is more uniform-linear.
- Always specify the weight scheme when reporting weighted kappa.
- For continuous data with measurement error, use the intraclass correlation coefficient (ICC) instead.
- Quadratic-weighted kappa equals ICC(3, 1) under certain assumptions – the two methods converge for ordinal Likert scales.

14.38.9 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.

14.38.10 See also — labs in this chapter

- Diagnostic testing: Se, Sp, PPV, NPV, LR
- Kappa, ICC, Bland–Altman
- Biomarker statistics (Youden, NRI, decision curves)
- TRIPOD-AI, fairness auditing, reproducibility at scale

14.39 Week 3, Session 3 — Biomarker statistics (Youden, NRI, decision curves)

Testing labs use the main template: Hypothesis → Visualise → Assumptions → Conduct → Conclude.

14.39.1 Learning objectives

- Compute ROC-AUC, Youden's index, sensitivity, and specificity at the optimal cut-point.
- Compute a net reclassification index between two predictive models.
- Construct a decision curve and interpret net benefit at a range of threshold probabilities.

14.39.2 Prerequisites

Binary classification and ROC curves; logistic regression.

14.39.3 Background

A candidate biomarker is not useful until it is tied to a decision. ROC-AUC summarises discrimination across all thresholds but is insensitive to *where* on the curve the action happens. Youden's index (sensitivity + specificity − 1) picks the threshold that maximises the equal-weighted sum. The net reclassification index (NRI) quantifies whether a new model reclassifies cases and non-cases in the correct direction relative to a baseline. Decision curve analysis plots *net benefit* as a function of the threshold probability, and lets a reader compare strategies ("treat all", "treat none", "treat by model") across a clinically relevant range.

Discrimination, calibration, and net benefit are three complementary axes. A biomarker with high AUC that is poorly calibrated can produce harmful decisions; a perfectly calibrated biomarker with low AUC gives no useful ranking. Reporting all three keeps the evaluation honest.

Decision curves are not hypothesis tests. They are a principled way to put a clinical question — what is the harm-to-benefit ratio of acting on this prediction? — into the analysis, and to see how the answer depends on that ratio.

14.39.4 Setup

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

14.39.5 1. Hypothesis

Can a logistic model on `Pima.tr` (glucose, BMI, age) distinguish diabetic from non-diabetic patients well enough to support a screening decision?

14.39.6 2. Visualise

```
d <- as_tibble(MASS::Pima.tr)
ggplot(d, aes(glu, bmi, colour = type)) + geom_point(alpha = 0.7)
```

14.39.7 3. Assumptions

Independence of observations; probability of diabetes is monotone in the linear predictor; no missingness.

14.39.8 4. Conduct

Fit a simple logistic regression and compute discrimination.

```
fit <- glm(type ~ glu + bmi + age, data = d, family = binomial())
d$p <- predict(fit, type = "response")
r  <- roc(d$type, d$p, direction = "<", quiet = TRUE)
auc(r)
coords(r, "best", ret = c("threshold", "sensitivity", "specificity",
                          "youden"), transpose = FALSE)
```

NRI against a glucose-only baseline.

```
fit0 <- glm(type ~ glu, data = d, family = binomial())
p0 <- predict(fit0, type = "response")
p1 <- d$p
# Continuous NRI
case   <- d$type == "Yes"
nri_up <- mean(p1[case]  > p0[case])   - mean(p1[case]  < p0[case])
nri_dn <- mean(p1[!case] < p0[!case])  - mean(p1[!case] > p0[!case])
nri <- nri_up + nri_dn
c(nri_cases = nri_up, nri_noncases = nri_dn, nri_total = nri)
```

A manual decision curve.

```r
thr <- seq(0.05, 0.6, by = 0.02)
dca <- sapply(thr, function(t) {
  treat <- p1 > t
  tp <- sum(treat &  case); fp <- sum(treat & !case); N <- length(case)
  tp / N - (fp / N) * (t / (1 - t))
})
nb_all <- sapply(thr, function(t) {
  tp <- sum(case); fp <- sum(!case); N <- length(case)
  tp / N - (fp / N) * (t / (1 - t))
})
tibble(threshold = thr, model = dca, treat_all = nb_all, treat_none = 0) |>
  pivot_longer(-threshold) |>
  ggplot(aes(threshold, value, colour = name)) + geom_line() +
  labs(x = "threshold probability", y = "net benefit")
```

14.39.9 5. Concluding statement

> A logistic model using glucose, BMI, and age discriminated diabetic from non-diabetic patients in `MASS::Pima.tr` with AUC `round(as.numeric(auc(r)), 3)`. The Youden-optimal cut-point occurred at a predicted probability of `round(coords(r, "best", ret = "threshold", transpose = FALSE)[1, 1], 2)`. Adding BMI and age to a glucose-only baseline produced an NRI of `round(nri, 2)`; the decision curve showed net benefit above "treat all" for threshold probabilities from roughly 0.15 to 0.5.

Decision curves give the clinical context: if the decision to intervene at, say, p = 0.2 is under discussion, the model is useful; at p = 0.05 or p = 0.7, it is barely distinguishable from treat-all or treat-none.

14.39.10 Common pitfalls

- Reporting AUC without calibration or decision curves.
- Computing NRI with a categorical risk cut-point and failing to disclose the cut-off.
- Using the same data to develop and evaluate the biomarker (apparent performance).

14.39.11 Further reading

- Pencina MJ, D'Agostino RB Sr, et al. (2008), *Evaluating the added predictive ability of a new marker*.
- Vickers AJ, Elkin EB (2006), *Decision curve analysis*.

14.39.12 Session info

```r
sessionInfo()
```

14.39.13 See also — chapter index

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

14.40 Week 2, Session 3 — Diagnostic testing: Se, Sp, PPV, NPV, LR

Inference labs use the five-step template: Hypothesis → Visualise → Assumptions → Conduct → Conclude.

14.40.1 Learning objectives

- Compute sensitivity, specificity, PPV, NPV, and positive and negative likelihood ratios from a 2×2 table.
- Convert pre-test probability to post-test probability with an LR.
- Sketch a receiver-operating characteristic curve from a continuous test statistic.

14.40.2 Prerequisites

Lab 2.2.

14.40.3 Background

A diagnostic test has two operating characteristics intrinsic to the test itself: *sensitivity* is the probability that a diseased person tests positive; *specificity* is the probability that a disease-free person tests negative. These quantities are properties of the test. They do not change with prevalence.

Two other quantities are properties of the test *and* the population in which it is applied: *positive predictive value* is the probability of disease given a positive test; *negative predictive value* is the probability of no disease given a negative test. These change with prevalence, sometimes dramatically.

Likelihood ratios unify the two pairs. *LR+* is sens / (1 − spec); *LR−* is (1 − sens) / spec. They convert pre-test odds to post-test odds by multiplication, which is the cleanest way to combine a test result with prior information. An LR+ greater than 10 is a strong positive; less than 0.1 is a strong negative; values near 1 are uninformative.

14.40.4 Setup

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

14.40.5 1. Hypothesis

Question of interest: how does a continuous biomarker behave as a diagnostic test? We are not running an inferential test; we are characterising a test's discrimination.

14.40.6 2. Visualise

Simulate a biomarker that is higher in diseased cases than in disease-free controls, with overlap.

```
N ← 500
prev ← 0.2
pop ← tibble(
  id = seq_len(N),
  disease = rbinom(N, 1, prev),
  biomarker = rnorm(N, mean = if_else(disease == 1, 7, 5), sd = 1)
)
```

```
pop ▷
  mutate(status = if_else(disease == 1, "disease", "no disease")) ▷
  ggplot(aes(biomarker, fill = status)) +
  geom_density(alpha = 0.5, colour = NA) +
  geom_vline(xintercept = 6, linetype = 2) +
  labs(x = "Biomarker level", y = "Density", fill = NULL)
```

14.40.7 3. Assumptions

The gold standard for disease status is assumed perfect. The biomarker is continuous and must be dichotomised at some cutoff to behave like a positive/negative test. We choose 6 as the cutoff for illustration; in practice, the cutoff is itself an outcome of the analysis.

```
cutoff ← 6
pop ← pop ▷ mutate(test = as.integer(biomarker > cutoff))
tab ← table(disease = pop$disease, test = pop$test)
tab
```

14.40.8 4. Conduct

```
TP ← tab["1", "1"]; FN ← tab["1", "0"]
FP ← tab["0", "1"]; TN ← tab["0", "0"]

sens ← TP / (TP + FN)
spec ← TN / (TN + FP)
ppv  ← TP / (TP + FP)
npv  ← TN / (TN + FN)
lrp  ← sens / (1 - spec)
lrn  ← (1 - sens) / spec

diag_tbl ← tibble(
  quantity = c("Sensitivity", "Specificity",
               "PPV", "NPV", "LR+", "LR-"),
  value = c(sens, spec, ppv, npv, lrp, lrn)
)
diag_tbl
```

Convert pre-test odds to post-test odds with the LR.

```
pre_prob ← 0.1
pre_odds ← pre_prob / (1 - pre_prob)
post_odds_pos ← pre_odds * lrp
post_prob_pos ← post_odds_pos / (1 + post_odds_pos)
post_odds_neg ← pre_odds * lrn
post_prob_neg ← post_odds_neg / (1 + post_odds_neg)

tibble(
  pre_prob,
  post_prob_if_positive = post_prob_pos,
  post_prob_if_negative = post_prob_neg
)
```

Sketch an ROC by sweeping the cutoff.

```
roc ← tibble(
  cut = seq(min(pop$biomarker), max(pop$biomarker), length.out = 200)
) ▷
  rowwise() ▷
  mutate(
    tp = sum(pop$biomarker > cut & pop$disease == 1),
    fn = sum(pop$biomarker ⩽ cut & pop$disease == 1),
    fp = sum(pop$biomarker > cut & pop$disease == 0),
    tn = sum(pop$biomarker ⩽ cut & pop$disease == 0),
    sens = tp / (tp + fn),
    fpr  = fp / (fp + tn)
  ) ▷
  ungroup()
```

```
ggplot(roc, aes(fpr, sens)) +
  geom_path(linewidth = 1) +
  geom_abline(linetype = 2, colour = "grey50") +
  coord_equal() +
  labs(x = "False positive rate (1 - specificity)",
       y = "Sensitivity")
```

14.40.9 5. Concluding statement

With a cutoff of `cutoff`, the biomarker had sensitivity `round(sens, 2)`, specificity `round(spec, 2)`, PPV `round(ppv, 2)`, and NPV `round(npv, 2)`. The positive likelihood ratio was `round(lrp, 2)` and the negative `round(lrn, 2)`. A pre-test probability of 10% becomes `round(post_prob_pos, 2)` after a positive test and `round(post_prob_neg, 3)` after a negative test.

A single cutoff collapses a rich continuous score into two states. The ROC curve shows the trade-off across all cutoffs; the area under it summarises overall discrimination without committing to a threshold.

14.40.10 Common pitfalls

- Quoting a single cutoff's sensitivity and specificity as if they were fixed properties of the test, ignoring that a different cutoff gives different numbers.
- Confusing sensitivity with PPV in everyday speech.

- Forgetting that PPV and NPV depend on prevalence.
- Using an ROC to compare tests with different prevalence in each sample.

14.40.11 Further reading

- Altman DG & Bland JM, *Diagnostic tests* series, BMJ.
- Pepe MS. *The Statistical Evaluation of Medical Tests for Classification and Prediction*.

14.40.12 Session info

```
sessionInfo()
```

14.40.13 See also — chapter index

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

14.41 Week 4, Session 2 — Kappa, ICC, Bland–Altman

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

14.41.1 Learning objectives

- Compute Cohen's kappa for categorical agreement and explain its chance-correction.
- Compute an intraclass correlation coefficient for continuous agreement and distinguish consistency from absolute agreement.
- Draw a Bland–Altman plot and report limits of agreement.

14.41.2 Prerequisites

Basic R and ggplot2.

14.41.3 Background

Measurement-agreement studies ask whether two raters, two methods, or two instruments give the same answer on the same units. The choice of statistic depends on the scale of the measurement. Cohen's kappa adjusts simple percent agreement for the agreement expected by chance given the marginal frequencies; it ranges from −1 to 1 with common landmarks at 0.4 and 0.6. Its main weakness is sensitivity to prevalence.

For continuous measurements, the intraclass correlation (ICC) and the Bland–Altman plot answer complementary questions. The ICC is a single-number summary of reliability, defined in several flavours (one-way, two-way, consistency vs absolute). The Bland–Altman plot shows *pattern*: it plots the difference between two raters against their mean and marks the limits of agreement (typically mean ± 1.96 SD). It reveals bias, proportional bias, and heteroscedasticity that ICCs hide.

Reliability is not the same as agreement. Two raters can be highly correlated (one is always twice the other) and have a terrible agreement. Always report both and let the picture tell the pattern.

14.41.4 Setup

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

14.41.5 1. Goal

Build two small rater datasets — one categorical, one continuous — and compute the matching agreement statistics.

14.41.6 2. Approach

For the categorical example, simulate 100 radiograph classifications (3 categories) by two readers with substantial but not perfect agreement. For the continuous example, simulate 60 measurements by two instruments, one with a small constant bias.

```
# categorical
cats ← c("normal", "mild", "severe")
truth ← sample(cats, 100, replace = TRUE, prob = c(0.5, 0.3, 0.2))
r1 ← ifelse(runif(100) < 0.2, sample(cats, 100, replace = TRUE), truth)
r2 ← ifelse(runif(100) < 0.25, sample(cats, 100, replace = TRUE), truth)
kap_tbl ← tibble(r1 = factor(r1, levels = cats),
                 r2 = factor(r2, levels = cats))

# continuous
n ← 60
true_val ← rnorm(n, 100, 15)
inst1 ← true_val + rnorm(n, 0, 3)
inst2 ← true_val + 2 + rnorm(n, 0, 3)    # small positive bias
meas ← tibble(inst1, inst2)
```

14.41.7 3. Execution

Cohen’s kappa:

```
kappa2(kap_tbl[, c("r1", "r2")])
```

ICC via `psych`:

```
ICC(as.matrix(meas))
```

Bland–Altman:

```
ba ← meas ▷
  mutate(mean_val = (inst1 + inst2) / 2,
         diff_val = inst2 - inst1)
loa ← mean(ba$diff_val) + c(-1.96, 0, 1.96) * sd(ba$diff_val)

ggplot(ba, aes(mean_val, diff_val)) +
  geom_point(alpha = 0.7) +
  geom_hline(yintercept = loa[1], linetype = 2, colour = "firebrick") +
  geom_hline(yintercept = loa[2], linetype = 1, colour = "steelblue") +
  geom_hline(yintercept = loa[3], linetype = 2, colour = "firebrick") +
  labs(x = "Mean of two instruments",
       y = "Difference (inst2 - inst1)")
```

14.41.8 4. Check

The ICC should be high (> 0.9) because the raters are well correlated, but the Bland–Altman plot shows a small positive bias (inst2 reads about 2 units higher on average).

14.41.9 5. Report

Cohen's kappa for the two radiograph readers was `round(kappa2(kap_tbl[, c("r1","r2")])$value, 2)`. For the two instruments, the ICC (absolute agreement, two-way random) was `round(ICC(as.matrix(meas))$results$ICC[2], 2)`, but the Bland–Altman plot revealed a mean bias of `round(mean(ba$diff_val), 1)` units with 95% limits of agreement from `round(loa[1], 1)` to `round(loa[3], 1)`.

14.41.10 Common pitfalls

- Reporting percent agreement instead of kappa.
- Using Pearson *r* on two raters and calling it agreement.
- Omitting the limits of agreement from a Bland–Altman plot.

14.41.11 Further reading

- Bland JM, Altman DG (1986), *Statistical methods for assessing agreement…*
- Shrout PE, Fleiss JL (1979), *Intraclass correlations…*
- McGraw KO, Wong SP (1996), *Forming inferences about some ICCs*.

14.41.12 Session info

```
sessionInfo()
```

14.41.13 See also — chapter index

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

14.42 Week 4, Session 5 — TRIPOD-AI, fairness auditing, reproducibility at scale

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

14.42.1 Learning objectives

- Enumerate the TRIPOD-AI reporting items relevant to a prediction- model manuscript.
- Compute group-stratified AUC and calibration as a fairness audit.
- Sketch a reproducible analysis pipeline with the `targets` package.

14.42.2 Prerequisites

External validation; biomarker evaluation.

14.42.3 Background

TRIPOD-AI extends the original TRIPOD statement to cover machine- learning prediction models. It asks authors to describe the data source, the participants, the outcome, the predictors, sample size and missing data, the model specification and its hyperparameter tuning, the performance on internal and external data, and the intended use of the model. A report that fails on any of these items is difficult to reproduce and difficult to deploy safely.

Fairness auditing extends validation to population subgroups. A model with strong overall AUC can have markedly worse performance in a minority subgroup; the remedy is first to detect the gap and then to decide whether to retrain, reweight, or accept the limitation explicitly.

The `targets` package is the modern R approach to reproducible pipelines. It builds a directed acyclic graph of analysis steps, caches intermediate outputs, and reruns only what has changed. This separation between pipeline definition and execution is what lets a study survive the months between submission and revision.

Reproducibility at scale is not a purity test. It is an insurance policy: when a reviewer asks for a recomputed sensitivity, or when a colleague tries to replicate the analysis two years later, the cost of doing the work as a scripted DAG is paid back many times.

14.42.4 Setup

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

14.42.5 1. Goal

Audit a logistic prediction model on `Pima.tr` by a simulated subgroup attribute, and sketch a `targets` pipeline for the full analysis.

14.42.6 2. Approach

Attach a synthetic subgroup label — imagine this were clinic of enrolment — and compare performance.

```
d ← as_tibble(MASS::Pima.tr) ▷
  mutate(subgroup = sample(c("A", "B"), n(), replace = TRUE,
                           prob = c(0.7, 0.3)))
ggplot(d, aes(glu, fill = subgroup)) +
  geom_histogram(alpha = 0.7, bins = 20, position = "identity")
```

14.42.7 3. Execution

```
fit ← glm(type ~ glu + bmi + age, data = d, family = binomial())
d$p ← predict(fit, type = "response")

auc_overall ← as.numeric(auc(roc(d$type, d$p, quiet = TRUE)))
auc_by ← d ▷
  group_by(subgroup) ▷
  summarise(auc = as.numeric(auc(roc(type, p, quiet = TRUE))),
            n   = n(), .groups = "drop")
auc_by
```

Calibration stratified by subgroup.

```
d ▷
  mutate(bin = cut(p, quantile(p, seq(0, 1, by = 0.2)),
                   include.lowest = TRUE)) ▷
  group_by(subgroup, bin) ▷
  summarise(pred = mean(p), obs = mean(type == "Yes"),
            n = n(), .groups = "drop") ▷
  ggplot(aes(pred, obs, colour = subgroup)) +
  geom_point(aes(size = n)) + geom_line() +
  geom_abline(slope = 1, intercept = 0, colour = "grey50") +
  labs(x = "mean predicted", y = "observed proportion")
```

A minimal `targets` pipeline (sketch).

```
library(targets)
tar_script({
  library(tidyverse); library(MASS); library(pROC)
  list(
    tar_target(raw, as_tibble(MASS::Pima.tr)),
    tar_target(fit, glm(type ~ glu + bmi + age, data = raw, family = binomial())),
    tar_target(auc_overall,
               as.numeric(auc(roc(raw$type, predict(fit, type = "response"), quiet =
TRUE)))),
    tar_target(report, tibble(auc = auc_overall))
  )
})
tar_make()
tar_read(report)
```

14.42.8 4. Check

TRIPOD-AI-style checklist (abbreviated).

```
checklist ← tribble(
  ~item,                              ~status,
  "Study design stated",              "yes",
  "Source and eligibility",           "yes",
  "Outcome definition",               "yes",
  "Predictor definitions",            "yes",
  "Sample size justified",            "partial",
  "Missing-data handling",            "yes",
  "Model specification",              "yes",
  "Hyperparameter tuning",            "NA (no tuning)",
  "Internal validation",              "yes",
  "External validation",              "NOT in this lab",
  "Calibration reported",             "yes",
  "Fairness audit by subgroup",       "yes",
  "Code available",                   "yes"
)
checklist
```

14.42.9 5. Report

> A logistic prediction model on `Pima.tr` achieved overall AUC `round(auc_overall, 2)`. A fairness audit by synthetic subgroup revealed AUCs of `round(auc_by$auc[1], 2)` in subgroup A (n = `auc_by$n[1]`) and `round(auc_by$auc[2], 2)` in subgroup B (n = `auc_by$n[2]`). A `targets` pipeline capturing raw data, fit, evaluation, and report would make the entire analysis re-runnable by any collaborator.

TRIPOD-AI, fairness auditing, and a pipeline tool are not independent initiatives; they are three faces of the same commitment to make modelling decisions legible, auditable, and reproducible.

14.42.10 Common pitfalls

- Reporting overall metrics and stopping; fairness gaps are only visible after stratification.
- Using `targets` as a static pipeline and not updating the DAG when inputs change.

- Treating TRIPOD-AI as a post-hoc checklist rather than a planning document written before analysis.

14.42.11 Further reading

- Collins GS et al. (2024), *TRIPOD+AI statement: updated guidance for reporting clinical prediction models that use regression or machine learning methods*.
- Obermeyer Z et al. (2019), *Dissecting racial bias in an algorithm used to manage the health of populations*.
- Landau WM (2021), *The targets R package: a dynamic make-like function-oriented pipeline toolkit*.

14.42.12 Session info

```
sessionInfo()
```

14.42.13 See also — chapter index

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