27 Cheatsheets

R. Heller · Date: 2026-05-08 · Last changed: 2026-10-30

One-page topical summaries, aggregated here as a printable reference. Each is a self-contained reminder of the syntax, the assumptions, and the diagnostics that go with the methods it covers.

27.1 Foundations

27.1.1 Toolchain, tidy data, and `ggplot2`

27.1.1.1 The research workflow

`Question → Measurements → Design → Acquisition → Description → Analysis → Interpretation → Validation → Knowledge`

Biological vs statistical significance: "$p < 0.05$" ≠ "effect matters".

27.2 R / RStudio / Quarto / renv

Task	Command
Install packages	`install.packages("pkg")`
Pin project	`renv::init()` / `renv::snapshot()` / `renv::restore()`
Render one file	`quarto render path/to/file.qmd`
Preview with live reload	`quarto preview`
Render only slides	`quarto render file.qmd --to revealjs`

27.3 Data types and scales

Scale	Example	Statistic
Nominal	sex, site	counts, proportions
Ordinal	pain 0–10	median, IQR, rank tests
Interval	°C	mean, SD
Ratio	kg, time	mean, SD, ratios

Accuracy ≠ precision. Bias shifts the target; variance spreads the shots.

27.4 Tidy data

1. One variable per column.
2. One observation per row.
3. One observational unit per table.

27.5 `dplyr` verbs

```
df |>
  filter(age >= 18) |>
  select(id, age, sbp) |>
  mutate(sbp_z = (sbp - mean(sbp)) / sd(sbp)) |>
  group_by(arm) |>
  summarise(mean_sbp = mean(sbp, na.rm = TRUE), n = n())
```

Joins: `left_join`, `inner_join`, `anti_join` by a key.

Missingness: `is.na()`, `drop_na()`, `tidyr::complete()`.

27.6 `ggplot2` grammar

```
ggplot(data, aes(x, y, colour = group)) +
  geom_point() +
  facet_wrap(~ factor) +
  labs(x = "x label", y = "y label") +
  theme_minimal()
```

Layer order: `data → aes → geom → facet → scales → labels → theme`.

27.7 Plot families

- One variable continuous → **histogram**, **density**.
- Two continuous → **scatter**, sometimes binned / smoothed.
- One categorical + one continuous → **boxplot**, **violin**, **strip**.
- Two categorical → **bar with fill**, **mosaic**.

27.8 Decision rule

- Cannot read the data into a tibble → fix data entry first.
- Tibble reads but has missing or mixed types → clean before modelling.
- Plot is ugly or cluttered → trust the plot, not the summary statistic.

27.9 Common pitfalls

- Running statistics on untyped / partially-missing data.
- Adjusting colour scales instead of simplifying the chart.

- Reporting mean + SD for a skewed variable.
- Forgetting `set.seed()` in any simulation chunk.

27.10 Further reading

- Wickham, *R for Data Science* (2e).
- Posit ggplot2 and dplyr cheatsheets.

27.10.1 Descriptive statistics, Bayes, and distributions

27.10.1.1 Descriptive statistics

Situation	Summary
Roughly symmetric, no outliers	mean ± SD
Skewed or has outliers	median, IQR, range
Categorical	counts and proportions
Grouped	Table 1 via `gtsummary::tbl_summary()`

27.11 Contingency tables

```
table(df$x, df$y)
prop.table(table(df$x, df$y), margin = 1)
```

27.12 Bayes' theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

In odds form: **posterior odds = prior odds × likelihood ratio.**

27.13 Diagnostic testing

Metric	Formula
Sensitivity	TP / (TP + FN)
Specificity	TN / (TN + FP)
PPV	TP / (TP + FP)
NPV	TN / (TN + FN)
LR+	Se / (1 − Sp)
LR−	(1 − Se) / Sp

PPV collapses at low prevalence — even a 95%-accurate test is mostly false positives.

27.14 Discrete distributions

Distribution	Use	R
Bernoulli(p)	single trial	`rbinom(n, 1, p)`
Binomial(n, p)	# successes	`dbinom / pbinom / rbinom`
Poisson(λ)	rare events	`dpois / ppois / rpois`
Negative binomial	overdispersed counts	`MASS::rnegbin`, `dnbinom`

Binomial → Poisson as n grows and p shrinks with $np = \lambda$.

27.15 Continuous distributions

Distribution	Use
Normal(μ, σ)	almost everything via the CLT
Student t(v)	inference about means, small n
$\chi^2(v)$	variance, counts, GoF
F(v_1, v_2)	variance ratios, ANOVA
Exponential(λ)	time between events, memoryless

R uses the `d / p / q / r` prefix: density, CDF, quantile, random.

27.16 Q-Q plot

```
ggplot(df, aes(sample = x)) + stat_qq() + stat_qq_line()
```

S-shape → heavy tails. U-shape → skew. Plot before the test.

27.17 Decision rule

- Reporting a mean? First check a histogram.
- Testing a proportion? First sketch a 2×2 table.
- Choosing a distribution? Simulate before believing.

27.18 Common pitfalls

- Quoting PPV without disclosing prevalence.
- Assuming normality because $n > 30$.
- Using mean ± SD on skewed data.

27.19 Further reading

- Altman, *Practical Statistics for Medical Research*.
- Gelman et al., *Bayesian Data Analysis*, ch. 1.

27.19.1 Sampling, estimation, and one-sample tests

27.19.1.1 Populations vs samples

- **Parameter** = truth about the population (μ, σ, π).
- **Estimator** = statistic computed on a sample ($\bar{x}$, s, $\hat{p}$).
- **Bias** = $E[\hat{\theta}] - \theta$. **MSE** = bias² + variance.

27.20 Central limit theorem

For iid samples with finite variance, as $n \to \infty$: $\bar{X} \sim N\left(\mu, \sigma/\sqrt{n}\right)$

Holds roughly by $n \approx 30$ for most non-pathological distributions.

27.21 Standard error

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

27.22 Bootstrap

```r
B ← 2000
boot_mean ← replicate(B, mean(sample(x, replace = TRUE)))
quantile(boot_mean, c(0.025, 0.975))   # 95% percentile CI
sd(boot_mean)                           # bootstrap SE
```

27.23 Permutation test

```r
obs  ← mean(a) - mean(b)
pool ← c(a, b)
null ← replicate(5000, {
  idx ← sample(seq_along(pool), length(a))
  mean(pool[idx]) - mean(pool[-idx])
})
mean(abs(null) ⩾ abs(obs))   # two-sided p
```

27.24 Maximum likelihood

Given data x and model $f(x;\theta)$:

- $\ell(\theta) = \sum \log f(x_i;\theta)$.

- $\hat{\theta}_{\text{MLE}} = \arg\max_\theta \ell(\theta)$.
- Fisher information $I(\theta)$; asymptotic SE = $1/\sqrt{I(\hat{\theta})}$.

27.25 One-sample tests (five-step template)

Hypothesis → Visualise → Assumptions → Conduct → Conclude.

Test	R	Assumptions
One-sample *t*	`t.test(x, mu = 0)`	roughly normal or large *n*
One-proportion	`prop.test(k, n, p = 0.5)` or `binom.test`	$np, n(1-p) \geq 5$ for `prop.test`

27.26 Hypothesis-testing vocabulary

- **Type I**: reject true H_0 (probability α).
- **Type II**: accept false H_0 (probability β).
- **Power** = $1 - \beta$.
- A *p*-value is **not** the probability that H_0 is true.

27.27 Decision rule

- Want a CI? Bootstrap first, then ask whether a formula applies.
- Want a test? State H_0 and α in writing before running it.
- Want to know if *n* is big enough? Simulate the CLT for your outcome.

27.28 Common pitfalls

- Reporting SE when you meant SD (or vice versa).
- Using the 95% CI to "prove" there is no effect.
- Running many tests and quoting the smallest *p*-value.

27.29 Further reading

- Harrell, *Biostatistics for Biomedical Research*, ch. 3–5.
- Efron & Tibshirani, *An Introduction to the Bootstrap*.

27.29.1 Two-group comparisons, effect sizes, and power

27.29.1.1 Choosing a comparison

Design	Continuous outcome	Binary outcome
Two independent groups	`t.test(y ~ g)` (Welch); Wilcoxon rank-sum	`prop.test` / `fisher.test`
Paired / pre-post	`t.test(pre, post, paired = TRUE)`; Wilcoxon signed-rank	McNemar
> 2 groups, continuous	`aov(y ~ g)`; Kruskal-Wallis	chi-square

27.30 Effect sizes

Statistic	What it reports
Mean difference + 95% CI	primary for continuous
Cohen's *d*	standardised mean difference
Hedges' *g*	small-sample correction of *d*
Risk ratio (RR) / odds ratio (OR)	two proportions
Risk difference	absolute, clinically intuitive

```
effectsize::cohens_d(y ~ g)
```

27.31 Two proportions

```
prop.test(c(tA, tB), c(nA, nB))          # asymptotic
fisher.test(matrix(c(tA, nA - tA,
                     tB, nB - tB), 2))  # exact, small cells
```

Report RR (or OR) with 95% CI, not just the *p*-value.

27.32 Correlation

Method	Captures	Assumptions
Pearson	linear association, continuous	bivariate normal, no outliers
Spearman	monotonic association	rank-based, robust
Kendall	concordant/discordant pairs	robust, slow on large data

```
cor.test(x, y, method = "spearman")
```

27.33 Non-parametric tests

Test	Replaces
Wilcoxon rank-sum (Mann-Whitney)	two-sample *t*
Wilcoxon signed-rank	paired *t*
Kruskal-Wallis	one-way ANOVA
Sign test	paired *t*, when even ranks fail

27.34 Power and sample size

```
pwr::pwr.t.test(d = 0.5, power = 0.80, sig.level = 0.05,
                type = "two.sample")
```

Family	Function
two-sample *t*	`pwr.t.test`
two proportions	`pwr.2p.test`, `pwr.2p2n.test`
correlation	`pwr.r.test`
one-way ANOVA	`pwr.anova.test`

Simulation-based power for anything the textbooks skip: `simr::powerSim`.

27.35 Reporting with `gtsummary`

```
library(gtsummary)
trial ▷
  tbl_summary(by = arm, statistic = list(all_continuous() ~ "{mean} ({sd})")) ▷
  add_p() ▷
  add_overall()
```

27.36 Decision rule

- Ask: one-group, two-group, paired, or many-group?
- Check: normal-ish, or should I use a rank-based test?
- Report: point estimate + 95% CI + effect size; *p*-value last, not first.

27.37 Common pitfalls

- Equal-variance *t*-test (default in some languages) when variances differ.
- Reporting OR when the audience expects RR (or vice versa).
- Forgetting that the *p*-value of a paired test depends on the pairing.

- Chaining correlations until one is “significant”.

27.38 Further reading

- Altman, Machin et al., *Statistics with Confidence*.
- `gtsummary` docs.

27.39 Regression

27.39.1 Linear regression and diagnostics

27.39.1.1 Correlation vs regression

- Correlation: symmetric; both variables random (**Model II**).
- Regression: asymmetric; predictor fixed, outcome random (**Model I**).
- Use Model II (e.g. `smatr::sma`) when both variables have error.

27.40 Simple linear regression

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2)$$

```
fit ← lm(y ~ x, data = df)
broom::tidy(fit, conf.int = TRUE)
broom::glance(fit)
```

27.41 Multiple regression

```
fit ← lm(y ~ x1 + x2 + x1:x2, data = df)
```

Concept	Fix
Confounding	adjust by including the confounder
Interaction	include `x1:x2` (or `x1 * x2`)
Non-linearity	`splines::ns(x, 4)` or `I(x^2)`
Scale sensitivity	centre / standardise predictors

27.42 Diagnostics

```
plot(fit)                    # base: 4 diagnostic plots
performance::check_model(fit)   # tidy overview
car::vif(fit)                # multicollinearity
```

Plot	What it flags
Residuals vs fitted	non-linearity, non-constant variance
Normal Q-Q	non-normal residuals
Scale-location	heteroscedasticity
Leverage / Cook's	influential outliers

VIF > 5 → investigate collinearity; > 10 → fix it.

27.43 Robust / weighted

```
MASS::rlm(y ~ x, data = df)          # robust to outliers
lm(y ~ x, weights = 1 / var_i)        # weighted LS

# Heteroscedasticity-consistent SEs
library(sandwich); library(lmtest)
coeftest(fit, vcov = vcovHC(fit, type = "HC3"))
```

27.44 Decision rule

- Residual plot ugly? Transform, add terms, or switch link.
- VIF explodes? Drop one of the collinear predictors or combine them.
- Outlier with high Cook's distance? Investigate before deleting.
- Variance depends on X? HC3 SEs or weighted LS.

27.45 Common pitfalls

- Interpreting β_1 as "effect of X" under confounding.
- Reporting R^2 on in-sample data as if it were generalisable.
- Including an interaction but only reporting main effects.
- Centring categorical predictors (don't — use reference coding).

27.46 Further reading

- Harrell, *Regression Modeling Strategies*, ch. 4–5.
- Fox, *Applied Regression Analysis*.

27.46.1 ANOVA, mixed models, and non-linear regression

27.46.1.1 One-way ANOVA

```
aov(y ~ group, data = df) ▷ summary()
```

ANOVA is a linear model with a categorical predictor. The F-test compares between-group to within-group variance.

27.47 Contrasts with `emmeans`

```
library(emmeans)
emm ← emmeans(fit, ~ group)
pairs(emm, adjust = "tukey")   # all pairwise
contrast(emm, list(TrtVsCtrl = c(-1, 1, 1, 1) / 3))
```

Pre-specify contrasts before looking at the data; correct for multiplicity.

27.48 Two-way / factorial ANOVA

```
aov(y ~ A * B, data = df) ▷ summary()
emmip(fit, A ~ B)   # interaction plot
```

Interaction means "effect of A differs by level of B". Report the interaction first; main effects are conditional.

27.49 Repeated measures / blocking

- RCBD: `aov(y ~ treatment + Error(block))`.
- Repeated measures: move to a mixed model.

```
library(lme4); library(lmerTest)
lmer(y ~ treatment + time + (1 | subject), data = df)
```

27.50 GAMs — smooth non-linear terms

```
library(mgcv); library(gratia)
fit ← gam(y ~ s(x, k = 10) + z, data = df)
summary(fit)      # edf tells you how "wiggly"
draw(fit)         # smooth + CI
```

edf ≈ 1 → nearly linear; > 4 → clearly non-linear.

27.51 Non-linear regression (`nls`)

```
# Michaelis-Menten: y = Vmax * x / (K + x)
fit ← nls(y ~ Vmax * x / (K + x),
          data = df, start = list(Vmax = 1, K = 1))
```

Start values matter. If it fails, plot first to guess reasonable starts.

27.52 Decision rule

- Categorical predictor, > 2 levels → ANOVA + contrasts.
- Factorial design → include interaction, report it first.
- Effect obviously curved → GAM with spline; else try `nls`.
- Repeated measures → mixed model, not repeated-measures ANOVA.

27.53 Common pitfalls

- Tukey HSD without pre-specified contrasts of interest.
- ANOVA p < 0.05 reported alone — without naming *which* groups differ.
- Forcing a GAM onto monotonic data that `nls` fits cleanly.
- Ignoring the random effect in clustered designs (pseudoreplication).

27.54 Further reading

- Wood, *Generalized Additive Models*, 2e.
- `emmeans` vignette.

27.54.1 Generalised linear models and prediction evaluation

27.54.1.1 GLM link functions

Outcome	Family	Link	R
Continuous	gaussian	identity	`lm()`
Binary	binomial	logit	`glm(y ~ x, family = binomial)`
Count	poisson	log	`glm(y ~ x, family = poisson, offset = log(t))`
Overdispersed count	neg. binomial	log	`MASS::glm.nb`
Ordinal	cumulative	logit	`MASS::polr`
Nominal	multinomial	logit	`nnet::multinom`

27.55 Logistic regression

```
fit ← glm(y ~ x1 + x2, data = df, family = binomial)
exp(coef(fit))                       # odds ratios
exp(confint(fit))                    # 95% CI on OR
predict(fit, newdata = nd, type = "response")
```

- Check for perfect separation (huge SEs).
- Interpret ORs cautiously; RR is more intuitive for the audience.

27.56 ANCOVA in an RCT

Adjust for baseline; **do not** analyse the change score.

```
lm(y_followup ~ arm + y_baseline, data = trial)
```

More efficient than simple t-test on change scores when baseline and follow-up are correlated.

27.57 Poisson / negative binomial

```
glm(cases ~ x + offset(log(person_years)),
    family = poisson, data = df)
MASS::glm.nb(cases ~ x + offset(log(person_years)), data = df)
```

Check dispersion: `sum(residuals(fit, type = "pearson")^2) / df.residual`. If > 1.5, switch to NB.

27.58 Evaluation — calibration + discrimination

Metric	Means	R
Calibration plot	predicted vs observed	`rms::val.prob`, manual bin
ROC / AUC	rank ordering	`pROC::roc(y, phat)`
Brier score	overall accuracy	`mean((phat - y)^2)`
Calibration slope / intercept	systematic bias	from logistic recalibration

```
library(pROC)
roc_obj ← roc(y, phat)
auc(roc_obj); ci.auc(roc_obj)
plot(roc_obj)
```

27.59 Decision rule

- Binary outcome → logistic; report OR + 95% CI.
- Count outcome → Poisson; check overdispersion; NB if needed.
- Trial analysis → ANCOVA, not change score.
- Prediction model → calibration curve first, ROC second, decision curves third.

27.60 Common pitfalls

- Quoting AUC without calibration (a discriminating but miscalibrated model is dangerous).
- Ignoring offsets in count data.

- Using ordinal logit when the proportional-odds assumption fails.
- Presenting logistic regression coefficients on the log-odds scale without OR.

27.61 Further reading

- Harrell, *RMS*, ch. 10–12.
- Steyerberg, *Clinical Prediction Models*.

27.61.1 Agreement, survival primer, and reporting

27.61.1.1 Don’t dichotomise

Cutting a continuous variable at the median loses ≈ 37% of the information for a linear association. Keep predictors continuous; model non-linearity with a spline if needed.

27.62 Change scores vs ANCOVA

- Change scores suffer from **regression to the mean**.
- ANCOVA (adjust for baseline) is the efficient analysis in an RCT.
- In observational studies with unequal baselines, both approaches answer subtly different questions — state which.

27.63 Agreement & reliability

Statistic	Use	R
Cohen’s κ	two raters, categorical	`irr::kappa2(cbind(r1, r2))`
Weighted κ	ordinal categorical	`irr::kappa2(..., weight = "squared")`
ICC(2,1) / ICC(3,1)	continuous	`psych::ICC(df)`
Bland-Altman	two methods, continuous	plot $\bar{x}y$ vs $y - x$

```r
mean_diff ← mean(y - x)
loa ← mean_diff + c(-1.96, 1.96) * sd(y - x)
ggplot(df, aes((x + y)/2, y - x)) + geom_point() +
  geom_hline(yintercept = c(mean_diff, loa), linetype = 2)
```

27.64 Survival primer

Concept	R
Kaplan-Meier	`survfit(Surv(time, event) ~ g) + ggsurvfit::ggsurvfit`
Log-rank test	`survdiff(Surv(time, event) ~ g)`

Concept	R
Cox PH model	`coxph(Surv(time, event) ~ x1 + x2)`
PH check	`cox.zph(fit)`

- Report a hazard ratio with 95% CI.
- PH violated? → time-varying coefficient or stratify.
- Interpret HR only after checking for non-informative censoring.

27.65 Decision curves, NRI, IDI

- **Decision curve**: net benefit vs threshold probability — dominates "treat all" and "treat none" when useful.
- **NRI**: how many events move up / non-events move down in risk.
- **IDI**: mean change in predicted probability by class.
- Report decision curve first; NRI/IDI as secondary.

27.66 Explanation vs prediction (Shmueli)

Explanatory	Predictive
Goal: inference about β	Goal: minimise out-of-sample loss
Tools: ANOVA, diagnostics, intervals	Tools: CV, regularisation, ensembles
Metric: interval coverage, bias	Metric: RMSE, AUC, Brier on hold-out

27.67 Reporting guidelines

Design	Guideline
Randomised trial	CONSORT
Observational	STROBE
Diagnostic-accuracy	STARD
Prediction model (incl. AI)	TRIPOD / TRIPOD-AI
Systematic review	PRISMA
Animal research	ARRIVE

27.68 Decision rule

- Continuous predictor + continuous outcome → no median splits.
- Two-group with repeated measurement → ANCOVA.
- New rater / device → Bland-Altman + ICC.

- Time-to-event → KM + Cox, always check PH.
- Prediction model → TRIPOD checklist at submission.

27.69 Common pitfalls

- Reporting κ on near-constant outcomes (high % agreement, low κ).
- Quoting median survival that is never reached.
- Forgetting to mention PH assumption checking.
- Claiming a "prediction model" built on the same data used for evaluation.

27.70 Further reading

- Harrell, *BBR*, ch. 17–18.
- Royston & Altman, *Prognosis research*.

27.71 Design & Causal Inference

27.71.1 Study designs and power

27.71.1.1 Observational designs

Design	Strengths	Weaknesses
Cohort	temporality, incidence	expensive, loss to follow-up
Case-control	rare outcomes, cheap	recall + selection bias
Cross-sectional	prevalence, screening	no temporality
Case-crossover	within-person comparison	for transient exposures only

STROBE checklist: https://www.strobe-statement.org/

27.72 Trial designs

Type	When
Parallel-group	most common; independent arms
Crossover	stable chronic condition, carry-over washable
Cluster	intervention at group level (schools, clinics)
Factorial	two or more interventions, test interactions
Adaptive	pre-specified modifications based on interim data
Non-inferiority	new treatment not worse than control by margin Δ
Equivalence	two-sided non-inferiority

CONSORT for RCTs: https://www.consort-statement.org/

27.73 Bench / translational

- **Blocking** reduces nuisance variation (plate, day, operator).
- **Factorial** tests interactions efficiently.
- **Split-plot** handles two levels of randomisation.
- **Pseudoreplication**: technical replicates ≠ biological replicates.

27.74 Power — closed form

```
library(pwr)
pwr.t.test(d = 0.5, power = 0.80, sig.level = 0.05,
           type = "two.sample")
pwr.2p.test(h = ES.h(0.3, 0.2), power = 0.80)
pwr.r.test(r = 0.3, power = 0.80)
pwr.anova.test(k = 4, f = 0.25, power = 0.80)
```

Effect-size conventions (Cohen): small d = 0.2, medium 0.5, large 0.8.

27.75 Power — simulation

```
library(simr)
# Build a pilot model, increase N, or tweak fixef, then:
ps ← powerSim(model, nsim = 500, test = fixed("arm"))
ps
```

Simulation wins for any design the textbook skips: mixed models, adaptive rules, non-standard outcomes.

27.76 Decision rule

- Randomise if you can. If not, draw a DAG and name the biases.
- Power calculation before the protocol freeze, not after data collection.
- Cluster randomisation → design effect $1 + (\bar{m} - 1)\rho$; inflate N.
- Bench experiments → treat batch, plate, and operator as random effects.

27.77 Common pitfalls

- Planning a cluster RCT without inflating N for design effect.
- Borrowing a pilot effect size without acknowledging noise.
- Running a non-inferiority trial as if it were superiority.
- Treating technical replicates as if they were biological.

27.78 Further reading

- Matthews, *Introduction to Randomized Controlled Clinical Trials*.
- Chow & Liu, *Design and Analysis of Clinical Trials*.

27.78.1 Missing data, mixed models, and longitudinal

27.78.1.1 Missingness mechanisms

Mechanism	Definition	Complete-case analysis?
MCAR	missingness independent of data	unbiased but inefficient
MAR	missingness depends on **observed** data	biased, imputation helps
MNAR	missingness depends on **unobserved** values	imputation under model only

Test is impossible without assumptions — reason about the mechanism.

27.79 Multiple imputation with `mice`

```
library(mice)
imp ← mice(df, m = 20, seed = 42, printFlag = FALSE)
fit ← with(imp, lm(y ~ x1 + x2))
pooled ← pool(fit)
summary(pooled, conf.int = TRUE)
```

Rule of thumb: $m \geq 100 \times$ fraction of missing information, but 20 is a fine starting point.

27.80 Linear mixed models

```
library(lme4); library(lmerTest)
lmer(y ~ treatment * time + (1 | subject), data = df)
lmer(y ~ treatment * time + (time | subject), data = df)  # random slope
```

- Random intercept: each cluster has its own baseline.
- Random slope: each cluster has its own trajectory.
- ICC $\rho = \sigma_u^2 / (\sigma_u^2 + \sigma_e^2)$.

REML for variance components; ML only for likelihood-ratio tests on fixed effects.

27.81 GLMMs & GEE

```
library(lme4)
glmer(y ~ x + (1 | cluster), data = df, family = binomial)

library(geepack)
geeglm(y ~ x, id = cluster, data = df,
       family = binomial, corstr = "exchangeable")
```

- GLMM: subject-specific effects, good for prediction.
- GEE: population-average effects, robust to working-correlation misspecification (when combined with robust SEs).

27.82 Time series basics

```
library(forecast)
ts_obj ← ts(x, frequency = 12)
decompose(ts_obj) ▷ plot()
auto.arima(ts_obj)

library(changepoint)
cpt.mean(x, method = "PELT") ▷ plot()
```

27.83 Decision rule

- Missing data > 5% → at least sensitivity analysis; prefer MI.
- Repeated measurements → mixed model, not repeated-measures ANOVA.
- Binary outcome with clusters → GLMM (if subject effects matter) or GEE (if marginal effect is the estimand).
- Time series with trend / seasonality → decompose before modelling.

27.84 Common pitfalls

- Last-observation-carried-forward; usually biased.
- Imputing after transformation rather than in the right scale.
- Ignoring random effects when clusters are small in number (degrees of freedom correction).
- Quoting GEE coefficients as if they were subject-specific.

27.85 Further reading

- van Buuren, *Flexible Imputation of Missing Data*.
- Fitzmaurice, Laird, Ware, *Applied Longitudinal Analysis*.

27.85.1 Causal inference methods

27.85.1.1 Time-varying covariates / landmarking

- Counting-process data: split follow-up into intervals.
- Landmark analysis: define a landmark time t^*, condition on surviving to t^*, classify by status at t^*.

```
library(survival)
coxph(Surv(tstart, tstop, event) ~ x, data = long)
```

- Immortal-time bias: exposed group can only be exposed if they live long enough to be exposed → spurious survival benefit.

27.86 Competing risks

```
library(tidycmprsk); library(ggsurvfit)
cif ← cuminc(Surv(time, event_f) ~ x, data = df)
ggsurvfit::ggcuminc(cif, outcome = "event1")

# Fine-Gray subdistribution hazard for event1
crr(Surv(time, event_f) ~ x, data = df, failcode = "event1")
```

- Cause-specific HR answers "given you are event-free, what is the hazard for event *k*?"
- Subdistribution HR answers "what is the hazard for event *k* in the whole cohort, treating competing events as obstacles?"

27.87 DAGs

```
library(dagitty); library(ggdag)
dag ← dagitty("dag { X → Y ; C → X ; C → Y }")
adjustmentSets(dag, exposure = "X", outcome = "Y")
impliedConditionalIndependencies(dag)
```

- Adjust for confounders, **not** colliders.
- Adjusting for a mediator estimates the direct effect only.

27.88 Propensity scores / IPTW

```
library(MatchIt); library(cobalt)
m ← matchit(treat ~ x1 + x2, data = df, method = "nearest")
love.plot(m)   # check balance: SMD < 0.1 is the goal
fit ← lm(y ~ treat, data = match.data(m))
```

IPTW: weight by $1/\hat{e}_i$ for treated, $1/(1-\hat{e}_i)$ for untreated. Stabilised weights stay closer to 1; trim extremes to avoid outliers.

27.89 G-methods, IV, DiD, RDD

Method	Identifying assumption
G-formula / IPTW	positivity + no unmeasured confounding
Instrumental variables	exclusion restriction + relevance
Difference-in-differences	parallel trends
Regression discontinuity	continuity at the cutoff

27.90 Heterogeneous treatment effects

- Causal forest (`grf::causal_forest`) or meta-learners (S-, T-, X-).
- Report conditional ATEs with adjusted intervals; avoid spurious subgroup p-values.

27.91 Decision rule

- Treatment assigned over time → time-varying Cox.
- Multiple competing events → cumulative incidence, Fine-Gray.
- Observational causal question → DAG first, choose identification strategy second.
- Want the whole distribution of treatment effect? → causal forest.

27.92 Common pitfalls

- Adjusting for a post-treatment variable (induces collider bias).
- Quoting HR from a Cox model that violates PH.
- Matching but never checking balance.
- Calling IV estimates "generalisable" when they are local to compliers.

27.93 Further reading

- Hernán & Robins, *Causal Inference: What If*.
- Therneau & Grambsch, *Modeling Survival Data*.

27.93.1 Synthesis and pre-registration

27.93.1.1 Systematic review workflow

1. PICO: Population, Intervention, Comparator, Outcome.
2. Register protocol on PROSPERO.
3. Build search across ≥ 2 databases; peer-review with a librarian.
4. Two-screener title/abstract and full-text.
5. Extract, appraise (RoB 2 / ROBINS-I), synthesise.

6. Report with the PRISMA 2020 checklist and flow diagram.

27.94 Meta-analysis

```r
library(metafor)
dat ← escalc(measure = "RR",
             ai = tpos, bi = tneg, ci = cpos, di = cneg,
             data = studies)
fit ← rma(yi, vi, data = dat, method = "REML")
fit
forest(fit, header = TRUE)
funnel(fit); regtest(fit)
```

Statistic	Meaning
τ^2	between-study variance
I^2	% of total variance due to heterogeneity
Egger's test	funnel-plot asymmetry (small-study effects)

Fixed-effect only when heterogeneity is effectively zero. Default to random effects.

27.95 Network meta-analysis

```r
library(netmeta)
net ← netmeta(TE, seTE, treat1, treat2, studlab,
              data = dat, sm = "MD", random = TRUE)
netgraph(net)
netrank(net, small.values = "good")    # SUCRA ranking
decomp.design(net)                     # inconsistency
```

Transitivity + consistency are the core assumptions.

27.96 Infectious disease — SIR / SEIR

```r
library(deSolve)
sir ← function(t, y, p) with(as.list(c(y, p)), {
  N ← S + I + R
  list(c(-beta * S * I / N,
         beta * S * I / N - gamma * I,
         gamma * I))
})
out ← ode(c(S = 999, I = 1, R = 0),
          times = 0:120,
          func = sir,
          parms = c(beta = 0.3, gamma = 0.1))
```

$R_0 = \beta/\gamma$. Herd-immunity threshold $\approx 1 - 1/R_0$.

27.97 Pre-registration and SAP

A pre-registration specifies, before data analysis:

- Primary outcome + exact estimand.
- Analysis model (fixed terms, adjustments, interactions).
- Handling of missing data.
- Multiplicity control.
- Sample-size justification.

OSF and ClinicalTrials.gov are the two common registries.

27.98 Decision rule

- More than a handful of studies on a question → plan a systematic review.
- Multiple comparators of interest → network meta-analysis.
- Outbreak modelling → start with SIR, extend only as data justify.
- Confirmatory analysis → pre-registered; exploratory analyses are labelled as such in the manuscript.

27.99 Common pitfalls

- Declaring a review “systematic” without a pre-registered protocol.
- Pooling studies with fundamentally different populations / outcomes.
- Interpreting SUCRA rankings as if they were certain.
- Calling a compartmental model a “forecast” without uncertainty.

27.100 Further reading

- Higgins et al., *Cochrane Handbook for Systematic Reviews*.
- Keeling & Rohani, *Modeling Infectious Diseases in Humans and Animals*.
- Nosek et al. (2018), *The preregistration revolution*.

27.101 Machine Learning & High-Dimensional Data

27.101.1 Cross-validation, regularisation, and unsupervised

27.101.1.1 Cross-validation

Scheme	Use
k-fold (k = 5 or 10)	baseline, fast
Repeated k-fold	reduces Monte Carlo noise
Leave-one-out	small n, high variance

Scheme	Use
Stratified k-fold	imbalanced class labels
Grouped / blocked	correlated units (patients, sites)
Nested	honest evaluation of tuned models

Nested CV is the only defensible way to estimate generalisation error when you are also tuning hyperparameters.

27.102 Regularisation

```r
library(glmnet)
x ← model.matrix(y ~ . - 1, df); y ← df$y
fit ← cv.glmnet(x, y, alpha = 1)     # alpha = 1 = lasso; 0 = ridge
coef(fit, s = "lambda.1se")          # sparse solution
plot(fit)                             # CV curve
```

- `lambda.min` : empirical minimum of CV error.
- `lambda.1se` : more regularised, simpler, within 1 SE of min.
- Elastic net: `alpha` between 0 and 1 blends ridge and lasso.

27.103 Multivariate workhorses

Method	Purpose
PCA	variance → low-d summary; `prcomp(x, scale. = TRUE)`
FA	latent-variable modelling; `psych::fa`
CCA	correlation structure between two sets
LDA	supervised low-d projection; `MASS::lda`

```r
pc ← prcomp(x, scale. = TRUE)
summary(pc)   # proportion of variance per PC
```

Scale before PCA when variables are on different units.

27.104 Clustering

```r
km ← kmeans(scale(x), centers = 4, nstart = 25)
hc ← hclust(dist(scale(x)), method = "ward.D2")
library(mclust); bic ← mclustBIC(x)      # model-based, BIC-selected
```

- Choose k by silhouette, gap statistic, or BIC.
- Scale first. K-means assumes round clusters of similar size.

27.105 Non-linear embeddings

```
library(uwot)
umap_xy ← umap(scale(x), n_neighbors = 15, min_dist = 0.1)

library(Rtsne)
ts ← Rtsne(scale(x), perplexity = 30)$Y
```

Embeddings are *visualisations*, not distances you should input into downstream stats. Global geometry is often distorted.

27.106 Decision rule

- Lots of predictors, small n → lasso; report `lambda.1se`.
- Exploring structure → PCA + UMAP on scaled data.
- Clustering for biology → model-based + sensitivity to k.
- Tuning anything → nested CV.

27.107 Common pitfalls

- Selecting features on full data, then estimating error by k-fold.
- PCA before scaling when features are on different units.
- Interpreting UMAP distances as real distances.
- Using AUC on rank-ordered hold-out error as if it were CV error.

27.108 Further reading

- Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning*.
- James et al., *ISLR2*.

27.108.1 Trees, boosting, and interpretability

27.108.1.1 Tree-based models

Model	R	Strengths
CART	`rpart::rpart`	interpretable single tree
Random forest	`ranger::ranger`	robust, OOB error, variable importance
XGBoost	`xgboost::xgb.train`	top tabular performance, careful tuning
LightGBM	`lightgbm::lgb.train`	fast on large data

```r
library(ranger)
fit <- ranger(y ~ ., data = df, importance = "permutation",
              num.trees = 500, mtry = floor(sqrt(ncol(df) - 1)))
fit$prediction.error   # OOB error
```

27.109 Interpretability

```r
library(DALEX); library(iml)

ex  <- explain(fit, data = df |> select(-y), y = df$y)
ip  <- model_parts(ex)            # permutation importance
pd  <- model_profile(ex, variables = "x1")   # partial dependence
sh  <- Shapley$new(Predictor$new(fit, data = df, y = df$y),
                   x.interest = df[1, ])     # SHAP
```

PDP assumes feature independence; if features are correlated, use ALE plots.

27.110 Tabular neural networks with `torch`

```r
# sketch only — typically in a loop on GPU
library(torch)
nn <- nn_sequential(
  nn_linear(p, 64), nn_relu(),
  nn_linear(64, 1)
)
opt <- optim_adam(nn$parameters)
```

On small biomedical tabular data, boosted trees usually beat NN. Use NN when you need end-to-end training with images, sequences, or text.

27.111 tidymodels pipeline

```r
library(tidymodels)
rec  <- recipe(y ~ ., data = df) |>
  step_normalize(all_numeric_predictors()) |>
  step_dummy(all_nominal_predictors())

mod  <- rand_forest(mtry = tune(), trees = 500) |>
  set_engine("ranger") |> set_mode("classification")

wf   <- workflow() |> add_recipe(rec) |> add_model(mod)
cv   <- vfold_cv(df, v = 5, strata = y)
tuned <- tune_grid(wf, cv, grid = 20)
collect_metrics(tuned)
```

27.112 Decision rule

- Tabular, < 100k rows → gradient-boosted trees.
- Need explanations → SHAP + PDP, validate with held-out.
- Images / sequences → CNN / transformer, use `torch`.

- Reproducibility → `tidymodels` pipeline + fixed seed + locked recipe.

27.113 Common pitfalls

- Reporting feature importance from a model trained on the full data.
- Interpreting PDPs when features are strongly correlated.
- Claiming NN superiority without CV on the same split as a tree model.
- Forgetting to set a seed for any tune / split operation.

27.114 Further reading

- Biecek & Burzykowski, *Explanatory Model Analysis*.
- Molnar, *Interpretable Machine Learning*.

27.114.1 Bayesian and survival ML

27.114.1.1 Bayesian thinking

$$\underbrace{p(\theta \mid y)}_{\text{posterior}} \propto \underbrace{p(y \mid \theta)}_{\text{likelihood}} \underbrace{p(\theta)}_{\text{prior}}$$

- Priors are part of the model — specify and defend them.
- Posterior summaries: mean, median, 95% credible interval.
- No *p*-values; use posterior probability of direction, Bayes factors, or LOO.

27.115 `brms` / Stan

```r
library(brms)
fit <- brm(y ~ x + (1 | group), data = df,
           prior = c(prior(normal(0, 1), class = "b"),
                     prior(exponential(1), class = "sd")),
           chains = 4, iter = 2000, seed = 42)

summary(fit)
loo(fit)               # leave-one-out cross-validation
pp_check(fit)           # posterior predictive checks
```

Prior predictive check before fitting: simulate y from the prior and confirm the implications are plausible.

27.116 Biomarker statistics

Question	Statistic
Does a biomarker classify?	AUC, cut-point (Youden's index)

Question	Statistic
Does it add over an existing model?	ΔAUC, NRI, IDI, decision curves
Does it move prognosis?	Calibration plus discrimination
Is a cut-off reproducible?	Bootstrap CI on the cut-off

```r
pROC::roc(y, biomarker) ▷
  pROC::coords("best", best.method = "youden")
```

27.117 Survival ML

Model	R
Random survival forest	`randomForestSRC::rfsrc(Surv(...) ~ ., data)`
Gradient-boosted Cox	`xgboost` with `objective = "survival:cox"`
DeepSurv (conceptual)	`torch` with partial-likelihood loss

Evaluate with **time-dependent AUC**, **integrated Brier score**, and **IPA** (index of prediction accuracy = 1 − Brier(model) / Brier(null)).

```r
library(timeROC)
r ← timeROC(T = df$time, delta = df$event,
             marker = pred, cause = 1, times = c(365, 730))
```

27.118 External validation

- Never trust the apparent performance on training data.
- Minimum: split-sample or CV on internal data.
- Better: external validation in a second cohort.
- Report calibration slope / intercept, discrimination, and decision curves.

27.119 Decision rule

- Rare outcome with need for uncertainty → Bayesian, not bootstrap of MLE.
- New biomarker → NRI + decision curve, not just ΔAUC.
- Survival prediction → time-dependent Brier / IPA, not global AUC.
- Any clinical claim → external validation before publication.

27.120 Common pitfalls

- Reporting a posterior with default flat priors on unscaled predictors.
- Selecting the Youden cut-off on the full data and using the same data to evaluate sensitivity / specificity.
- Quoting C-statistic at a single time point and calling it survival ML.

- Publishing a prediction model without TRIPOD-compliant reporting.

27.121 Further reading

- Gelman et al., *Bayesian Data Analysis*, 3e.
- Royston & Altman, *External validation of a Cox prognostic model*.

27.121.1 Omics, FDR, and TRIPOD-AI

27.121.1.1 Bulk RNA-seq

```
library(DESeq2)
dds ← DESeqDataSetFromMatrix(counts, coldata, ~ condition)
dds ← DESeq(dds)
res ← results(dds, contrast = c("condition", "B", "A"))
```

```
library(edgeR)
dge ← DGEList(counts, group = coldata$condition)
dge ← calcNormFactors(dge)
design ← model.matrix(~ condition, coldata)
fit ← glmQLFit(estimateDisp(dge, design), design)
qlf ← glmQLFTest(fit, coef = 2)
topTags(qlf)
```

Package	Distribution	Notes
DESeq2	negative binomial	shrinks LFCs; standard for small cohorts
edgeR	negative binomial	QL-F test has good type-I control
limma-voom	mean-variance weights	fast on very large studies

27.122 Enrichment

```
library(fgsea)
fgsea(pathways, stats = stat_vector, nperm = 10000) ▷
  arrange(padj)

# Over-representation with a hypergeometric test
phyper(k - 1, K, N - K, n, lower.tail = FALSE)
```

Always pre-rank by a *signed* statistic (log-FC × -log10 p), not by raw p alone.

27.123 Single-cell RNA-seq (Seurat pattern)

```
library(Seurat)
so ← CreateSeuratObject(counts) ▷
  NormalizeData() ▷
  FindVariableFeatures() ▷
  ScaleData() ▷
  RunPCA() ▷
  FindNeighbors() ▷
  FindClusters(resolution = 0.4) ▷
  RunUMAP(dims = 1:20)
DimPlot(so)
```

27.124 FDR, knockoffs, replication

Correction	When
Bonferroni	few planned tests, strict control
Benjamini-Hochberg	many tests, accept proportion of false discoveries
Knockoffs	controlled variable selection with FDR
Permutation / SEA	small *n*, non-standard models

```
p.adjust(pvals, method = "BH")
```

Replication crisis shorthand: pre-register + report effect sizes + share data. Power, not p-values.

27.125 TRIPOD-AI + fairness

- TRIPOD-AI extends TRIPOD for prediction models built with ML.
- Report: data provenance, training/evaluation splits, subgroup performance, calibration, external validation.
- Fairness: stratify AUC / calibration by sex, ancestry, age bands. Disparities in calibration are often larger than in AUC.

27.126 Reproducibility at scale with `targets`

```
# _targets.R
library(targets)
tar_option_set(packages = c("tidyverse"))
list(
  tar_target(raw, read_csv("data/raw.csv")),
  tar_target(clean, clean_data(raw)),
  tar_target(fit, fit_model(clean)),
  tar_target(report, quarto::quarto_render("report.qmd"))
)
```

```
Rscript -e 'targets::tar_make()'
```

27.127 Decision rule

- Bulk DE → DESeq2 for small samples, edgeR/limma for large.
- Many hypotheses → BH by default; knockoffs for variable selection.
- scRNA-seq → Seurat pipeline with batch correction + cluster annotation.
- ML prediction model → TRIPOD-AI + subgroup analysis.
- Any multi-step analysis → `targets` pipeline.

27.128 Common pitfalls

- Running DE on a pathway-level aggregate rather than gene-level.
- Cherry-picking a pathway database post-hoc.
- Over-clustering scRNA-seq data until "known" groups appear.
- Calling an ML model "fair" after checking only overall AUC.
- Submitting a paper without a `sessionInfo()` or a lock file.

27.129 Further reading

- Love, Huber, Anders (2014), *Moderated estimation of fold change…*
- Collins et al. (2024), *TRIPOD-AI*.
- Peng, *Reproducible Research with R*.