mfrmr quick reference

mfrmr package

Data to fitted results

A compact reference for the data review / fit / summary / plot / diagnose / report loop in mfrmr. Use `plot()` on fitted models or saved analysis results. For old `plot_*()` calls, use `help("mfrmr_plot_migration")`; the 33 drawing helpers are internal in 0.2.4. Every snippet assumes `library(mfrmr)` and a long-format `data.frame` with `Person`, one or more facet columns, and `Score`. The main route below is ordinary RSM/PCM; optional snippets require their named study inputs. For a complete first analysis, use `vignette("mfrmr-workflow", package = "mfrmr")`; for equations and references, open `help("mfrmr_methods_reference")`.

1. Check data roles, score support, and missing codes

```
toy <- load_mfrmr_data("example_operational")

data_review <- describe_mfrm_data(
  toy, person = "Person", facets = c("Rater", "Criterion"),
  score = "Score", rating_min = 1, rating_max = 4, category_policy = "preserve"
)
summary(data_review)

# Declare missing SCORE codes; preserve IDs with the same text.
clean_data <- recode_missing_codes(dirty_data, columns = "Score",
                                   codes = c(99, 999, -1, "N/A", ""))
dirty_review <- describe_mfrm_data(
  clean_data, "Person", c("Rater", "Criterion"), "Score",
  rating_min = 1, rating_max = 4, category_policy = "preserve"
)
```

Confirm retained and zero-count categories and inspect the score map. Use the same cleaned data, rubric bounds and `category_policy` in `fit_mfrm()`. Supply your own `expected_design` when the assignment roster is known; missing rows cannot be counted from column missingness alone.

2. Fit

```
fit <- fit_mfrm(
  toy,
  person = "Person",
  facets = c("Rater", "Criterion"),
  score  = "Score",
  method = "MML",          # recommended starting method
  model  = "RSM",          # use PCM/GPCM only with a model rationale
  rating_min = 1,
  rating_max = 4, category_policy = "preserve"
)
```

For PCM, choose an explicit `step_facet` from the rubric and question. For GPCM, read `gpcm_capability_matrix()` before using downstream helpers: one slope owner and two slope families have different estimation and output routes. The former "bounded GPCM" label described implementation scope, not a distinct model or a finite slope box. Two-family GPCM and corrected JML do not support the ordinary diagnostic and Wright-map calls below.

3. Read summaries, then choose the Wright display

```
fit_summary <- summary(fit, profile = "fit", detail = "brief")

# Comprehensive FACETS-organized review; mfrmr remains the estimator.
review <- summary(fit, profile = "facets", detail = "brief")
res <- review$results
diag <- res$diagnostics

# Primary final-scale display: all locations plus available facet uncertainty.
plot(res, type = "wright", renderer = "native",
     show_ci = TRUE, top_n = Inf, preset = "publication")
```

Read optimizer status and terminal-gradient guidance before interpreting estimates. The native map is the primary result because it keeps facet uncertainty and fitted step locations visible. MML Person uncertainty is posterior SD; facet SEs describe structural parameters. High rater separation reliability means distinguishable severities, not agreement. Low separation can also reflect poor precision; strata is not a set of discovered groups.

4. Add only the diagnostics needed for the question

```
# Reuse review diagnostics; rerun only for explicit custom settings.
diag_pca <- diagnose_mfrm(fit, residual_pca = "both")

# Optional follow-up: x = Infit, y = measure. Persons are opt-in.
plot(fit, type = "fit_pathway", diagnostics = diag,
     fit_stat = "Infit", include_person = FALSE)
```

5. Report / APA

```r
chk  <- reporting_checklist(fit, diagnostics = diag)
apa  <- build_apa_outputs(fit, diag)
cat(apa$report_text)

tbl  <- apa_table(fit, which = "facets", diagnostics = diag)
kab  <- as_kable(tbl)       # Markdown / HTML via kableExtra
ft   <- as_flextable(tbl)   # Word / PowerPoint via flextable
```

6. Bias / DFF / Equivalence

```r
bias <- estimate_bias(fit, diag, facet_a = "Rater", facet_b = "Criterion")
summary(bias)
plot(bias, type = "bias", view = "ranked", show_ci = TRUE)

# Requires a study data frame containing the named grouping column.
dff <- analyze_dff(fit, diagnostics = diag,
                   facet = "Rater", group = "Group", data = your_data)
plot(dff, type = "summary")

# Requires inference-ready MML and unregularized joint covariance.
# Choose the practical bound before inspecting equivalence results.
eq <- analyze_facet_equivalence(fit, facet = "Rater",
                                equivalence_bound = 0.5, ci_level = 0.95)
plot(eq, type = "facet_equivalence", view = "forest")
```

Bias screens hold calibration fixed; their displayed intervals do not propagate structural-parameter uncertainty. Equivalence uses its separate joint-covariance checks. A flag or overlapping individual intervals is not a test that two raters are interchangeable.

7. Hierarchical structure and small-N review

```
review <- facet_small_sample_review(fit)
summary(review)

icc <- compute_facet_icc(
  toy, facets = c("Rater", "Criterion"),
  score = "Score", person = "Person",
  ci_method = "boot", ci_boot_seed = 2026
)

h <- analyze_hierarchical_structure(
  toy, facets = c("Rater", "Criterion"),
  ci_method = "boot", ci_boot_seed = 2026
)
```

Read `ICC_CI_Status` and `attr(icc, "icc_ci")` before reporting intervals. Bootstrap intervals require the fitted Gaussian model and complete, converged refits without warnings. The former `"profile"` method is no longer supported; rerun saved interval analyses with the original data and settings.

8. Empirical-Bayes shrinkage for small-N facets

```
# Integrated:
fit_eb <- fit_mfrm(toy, "Person", c("Rater", "Criterion"), "Score",
                   method = "MML", facet_shrinkage = "empirical_bayes")
shrinkage_report(fit_eb)

# Post-hoc:
fit_eb2 <- apply_empirical_bayes_shrinkage(fit)
```

9. G theory: components, future plans and plots

G/D studies analyze observed scores and their assignment design; they do not require an MFRM fit. Here V and W are two fixed score components, and Task is random. Choose weights before comparing plans.

```
tasks <- read.csv(system.file("extdata", "mgenova-table12.csv", package = "mfrmr"))
g <- mfrm_multivariate_gstudy(tasks, scores = c("V", "W"),
                              person = "Person", task = "Task", rater = NULL)
plot(g, score = c("V", "W"), ncol = 2, preset = "monochrome")
d <- mfrm_multivariate_d_study(g, design_grid = data.frame(Tasks = c(6, 12)),
                               weights = c(V = .5, W = .5))
summary(d) # Read G/Phi and SEM status columns as well as their values
p <- plot(d, type = "coefficients", preset = "monochrome")
plot_data(p, "notes")
saveRDS(list(gstudy = g, dstudy = d, figure = p), "g-study.rds")
saved_g <- readRDS("g-study.rds")
summary(saved_g$dstudy)
```

G concerns relative ordering; Phi concerns absolute score levels. SEMs are in original score units. The highest interaction and residual share a component when they cannot be separated. Raw negative components remain visible. Component plots have no component SEs or confidence intervals; paired plan-difference intervals have a separate normal-theory scope. See `vignette("mfrmr-gtheory", package = "mfrmr")` for crossed/nested designs, covariance heatmaps, color, D-study curves and facet-count panels. The older `mfrm_generalizability()` route uses a different main-effects decomposition.

10. Plot surface (selected)

Function	Purpose
`plot(fit)`	Wright map (default)
`plot(fit_rsm, comparison = fit_pcm, type = "ccc", view = "difference")`	Conditional probability differences; paired Wright views also available
`plot(fit, type = "wright", view = "unified", renderer="facets")`	Optional FACETS-style * ruler; visual convention only
`plot(fit, type="fit_pathway", diagnostics=diag, include_person=TRUE)`	Infit-vs-measure pathway; person rows opt-in
`plot(fit, type = "thresholds")`	Category threshold ordering
`plot(fit, type = "person_fit", diagnostics = diag)`	Per-person Infit/Outfit bubble
`plot(fit, type = "rater_severity", diagnostics = diag)`	Rater ranking + CI bands
`plot(bias, type = "bias", view ="heatmap")`	Bias cell heatmap
`plot(fit, type = "displacement", show_ci=TRUE)`	Anchor displacement lollipop
`plot(fit, type = "fair_average", show_ci=TRUE)`	Fair-score diagnostic intervals; full-refit coverage unverified
`plot(drift, type="forest")`	Wave-level anchor CI forest
`plot(chain, type="graph")`	Wave/common-element graph with per-link screening
`plot(chain, type="links")`	Recorded wave links; counts of retained elements
`plot(chain, type="anchor_removal")`	Newly disconnected wave pairs after each common-element deletion; no refitting
`plot(chain, type="offset_sensitivity")`	Rescreened offset changes after deletion; source estimates/SEs fixed

Titles and cautions are returned as metadata by default. Save the plot object and read `plot_data(p, "notes")`; keep its caveats with the figure when exporting.

11. Reproducibility and export

```
# The default warning reminds you that this does not deidentify the files.
export_mfrm_bundle(fit, diagnostics = diag,
                   output_dir = "mfrmr_bundle/")
export_mfrm_results(res, output_dir = "mfrmr-results",
                    preset = "starter", overwrite = TRUE)
# Add acknowledge_sensitive = TRUE only after reviewing every intended file
# and confirming that the destination follows the study's handling policy.
manifest <- build_mfrm_manifest(fit, diagnostics = diag)
# manifest$environment, $dependencies, $input_summary, $session_info,
# $hierarchical_review, $missing_recoding, $shrinkage_review
replay_script <- build_mfrm_replay_script(fit, path = "replay_mfrmr.R")
```

12. FACETS and ConQuest users

- Keep the native Wright map with available facet uncertainty as the primary mfrmr display. Use `renderer = "facets"` when readers benefit from FACETS-style asterisks and ruler layout; this does not reproduce FACETS estimation.

```
rubric_labels <- setNames(
  your_rubric_labels,
  fit$prep$score_map$OriginalScore
)
plot(fit, type = "wright", view = "unified", renderer = "facets", show_ci = FALSE, category_labels = rubric_labels)
```

- Use `facets_feature_coverage()` for feature-by-feature alternatives.
- Use `build_conquest_overlap_bundle(fit_lr)` only after fitting the documented unidimensional MML comparison. Run the generated `.cqc` file in ConQuest, then pass its four comparison CSV exports to `normalize_conquest_overlap_exports()` and `review_conquest_overlap()`; retain the additional history CSV for objective/free-dimension verification.
- mfrmr does not execute FACETS or ConQuest and does not claim general numerical equivalence with either program.

Package links

- Source and documentation: https://github.com/Ryuya-dot-com/mfrmr
- Citation: `citation("mfrmr")`

mfrmr quick reference. Distributed under the MIT License.