Package 'epmfd'

October 20, 2025

Title Exploratory and Person/Item Misfit Diagnostics for Polytomous Data

Version 1.1.1

Description
Analysis of items and persons in data. To identify and remove person misfit in polytomous item-response data using either 'mokken' or a graded response model (GRM, via 'mirt'). Provides automatic thresholds, visual diagnostics (2D/3D), and export utilities.
Methods build on Mokken scaling as in Mokken (1971, ISBN:9789027968821) and on the graded response model of Samejima (1969) <doi:10.1007/BF03372160>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.2

URL https://github.com/hsnbulut/epmfd

BugReports https://github.com/hsnbulut/epmfd/issues

Imports dplyr, fs, ggplot2, mirt, mokken, PerFit, readr, rlang, tibble

Suggests plotly, ggrepel, openxlsx, haven, patchwork, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 4.1.0)

LazyData true

NeedsCompilation no

Author Hasan Bulut [aut, cre] (ORCID: <https://orcid.org/0000-0002-6924-9651>), Asiye Şengül Avşar [aut] (ORCID: <https://orcid.org/0000-0001-5522-2514>)

Maintainer Hasan Bulut <hasan.bulut@omu.edu.tr>

Repository CRAN

Date/Publication 2025-10-20 19:40:08 UTC

Contents

clean_epmfd 2
export_epmfd 3
load_epmfd 6
misfit_epmfd 7
plot_epmfd 8
plot_misfit 10
print.epmfd_misfit 12
sampledata 12
scale_epmfd 13
summary.epmfd_clean 14

Index **15**

`clean_epmfd` *Remove misfitting persons from an epmfd_misfit object*

Description

`clean_epmfd()` removes individuals flagged as misfitting according to a chosen decision rule and returns a cleaned dataset that can be passed directly to `scale_epmfd()`.

Usage

```
clean_epmfd(misfit, criterion = c("union", "intersection"), clean_item = FALSE)
```

Arguments

`misfit`	An `epmfd_misfit` object returned by `misfit_epmfd()`.
`criterion`	Character string, either `"union"` (default) or `"intersection"`.
`clean_item`	is a logical argument. If clean_item=TRUE, then the function can clean items. The defaul value is FALSE.

Details

The function uses logical misfit indicators stored in `misfit$table`, including:

- `misfit_any`: `TRUE` if at least one statistic flagged the person.
- Statistic-specific columns (e.g., `Gnp`, `U3p`, `lpz`) indicating per-statistic misfit decisions.

The set of statistics actually considered is taken from `misfit$stats`. Under the `"intersection"` rule, a person is removed only if **all** of those statistics are `TRUE`. Internally, `rowSums(..., na.rm = TRUE)` is used so that `NA` values do not force removal (i.e., `NA` behaves as "not flagged" in the intersection count).

Only items listed in `misfit$scaled$kept` are retained in the output. Person identifiers from the original raw object are preserved for the kept rows.

Value

An epmfd_clean list with:

- raw: An epmfd_raw object containing only the retained persons and items, directly usable in scale_epmfd().
- clean_data: The cleaned raw data frame (persons × kept items).
- n_removed: Number of persons removed.
- criterion: The applied decision rule.
- misfit: The original epmfd_misfit object (as provided).

Criterion

- "union" (default): A person is removed if **at least one** statistic (e.g., Gnp, U3p, lpz) flags them as misfitting. This is stricter.
- "intersection": A person is removed **only if all** statistics flag them as misfitting. This is more lenient.

See Also

misfit_epmfd(), scale_epmfd()

Examples

```
library(epmfd)
data<-load_epmfd(sampledata)
scaling_data<-scale_epmfd(data)
misfit_result<-misfit_epmfd(scaling_data)
clean_data<-clean_epmfd(misfit_result)
head(clean_data$clean_data)
dim(data$data)  # the dimension of raw data
dim(clean_data$clean_data)  # the dimension of clean data
```

---

export_epmfd *Export epmfd objects to disk*

Description

export_epmfd() writes commonly used tables from epmfd_* objects to CSV / Excel / SPSS files, and (optionally) saves the object itself as an RDS.

Usage

```
export_epmfd(
  object,
  dir = NULL,
  prefix = NULL,
  format = c("csv", "xlsx", "sav"),
  save_rds = FALSE,
  include_misfit = FALSE
)
```

Arguments

object	One of: epmfd_scaled, epmfd_misfit, epmfd_clean.
dir	Target directory. If NULL (default), **no files are written**; instead, the function returns the tables as a named list. If provided, the directory must exist or will be created.
prefix	File name prefix (without extension). If NULL, the first class name of object is used (e.g., "epmfd_clean").
format	Output format; one of "csv" (default), "xlsx", or "sav". • "csv": written via **readr** (readr::write_csv()). • "xlsx": requires **openxlsx** (openxlsx::write.xlsx()). • "sav": SPSS format; requires **haven** (haven::write_sav()).
save_rds	Logical; if TRUE and dir is provided, also saves the object as <prefix>.rds in dir via saveRDS().
include_misfit	Logical; if TRUE, writes/returns misfit tables when available (see Details). Default = FALSE.

Details

What is produced depends on the object class:

- epmfd_clean: cleaned person-by-item data (clean); if include_misfit = TRUE and a misfit object is attached, also misfit.
- epmfd_misfit: if include_misfit = TRUE, misfit.
- epmfd_scaled: item status summary (scale).

When format = "sav", logical columns are converted to labelled factors (FALSE/TRUE) for SPSS compatibility. Writing .sav does not support list columns; the function aborts if such columns are present.

Value

If dir is NULL, a **named list** containing the tables that would be written (e.g., clean, misfit, scale). If dir is non-NULL, (invisibly) a character vector of file paths that were written.

File naming (when `dir` is provided)

Files are named `<prefix>_<name>.<format>` under `dir`. For example: `study1_clean.csv`, `study1_misfit.xlsx`, or `study1_scale.sav`.

See Also

`saveRDS()`, **readr**, **openxlsx**, **haven**

Examples

```
# Minimal toy objects created inside the example ----
set.seed(1)
toy_clean <- data.frame(
  I1 = sample(0:1, 6, TRUE),
  I2 = sample(0:1, 6, TRUE)
)
toy_misfit <- data.frame(
  person = 1:6, Gpn = runif(6), U3p = runif(6)
)

clean_obj <- structure(
  list(clean_data = toy_clean,
       misfit     = list(table = toy_misfit)),
  class = "epmfd_clean"
)

misfit_obj <- structure(
  list(table = toy_misfit, method = "mokken"),
  class = "epmfd_misfit"
)

scaled_obj <- structure(
  list(kept = c("I1", "I2"), removed = character()),
  class = "epmfd_scaled"
)

# 1) No writing: return list
lst <- export_epmfd(clean_obj, dir = NULL, include_misfit = TRUE)
str(lst)

# 2) Write to a temporary directory (CRAN policy)
tmpdir <- tempdir()
export_epmfd(clean_obj,  dir = tmpdir, prefix = "study1", format = "csv",
             save_rds = TRUE)

# Optional formats guarded by Suggests (run only if installed)
if (requireNamespace("haven", quietly = TRUE)) {
  export_epmfd(misfit_obj, dir = tmpdir, format = "sav",
               include_misfit = TRUE)
}
if (requireNamespace("openxlsx", quietly = TRUE)) {
```

```
  export_epmfd(scaled_obj, dir = tmpdir, prefix = "scaleA",
               format = "xlsx")
}
```

---

load_epmfd *Load and validate raw data for the epmfd workflow*

Description

load_epmfd() prepares raw item-response data for subsequent functions in the epmfd workflow. It validates input, ensures that all item responses fall within the expected range of categories, converts items to ordered factors, and attaches person IDs.

Usage

```
load_epmfd(data, id_col = NULL, likert_levels = NULL)
```

Arguments

data	A data.frame or tibble with persons in rows and items in columns. All item responses must be integers in 1:K, possibly with missing values.
id_col	Optional character string giving the column name containing unique person identifiers. If NULL, a simple integer sequence 1:n is used.
likert_levels	Optional integer specifying the maximum category value (K). If NULL, K is inferred automatically as the maximum observed value in the data.

Details

Each column of data is validated to ensure responses are within 1:K. Values outside this range cause an error. Missing values are allowed and reported.

Value

An object of class epmfd_raw, a list with elements:

- data: A data.frame of ordered-factor responses
- id: Vector of person IDs
- K: Maximum number of categories per item

See Also

scale_epmfd(), misfit_epmfd()

Examples

```
# Example: 5 persons × 3 items, responses 1–4
df <- data.frame(
  Pid = paste0("P", 1:5),
  Item1 = c(1, 2, 3, 2, 1),
  Item2 = c(2, 3, 4, 2, 1),
  Item3 = c(3, 4, 1, 2, 2)
)

raw <- load_epmfd(df, id_col = "Pid", likert_levels = 4)
str(raw)
```

`misfit_epmfd` *Compute person-fit statistics (polytomous data)*

Description

`misfit_epmfd()` computes selected person-fit statistics for polytomous responses and returns an `epmfd_misfit` object with scores, thresholds, and logical flags per person.

Usage

```
misfit_epmfd(object, stats = c("auto", "lpz", "Gnp", "U3p"), cut.off = "auto")
```

Arguments

`object`	An `epmfd_scaled` object (output of your scaling step).
`stats`	Character vector choosing which statistics to compute. Allowed values: `"auto"`, `"lpz"`, `"Gnp"`, `"U3p"`. If `"auto"` is present, the set is chosen based on the detected scaling method: • for `"mirt"`: `c("lpz","Gnp","U3p")` • for `"mokken"`: `c("Gnp","U3p")`
`cut.off`	Cut-off for `Gnp` and `U3p`. Either `"auto"` (default; uses **PerFit**'s `cutoff()` with its implied tail), or a single numeric value (interpreted with tail `"upper"` for both `Gnp` and `U3p`). `lpz` uses a fixed lower-tail cut-off of `-1.645`.

Details

Auto vs manual decision for `misfit_final`:

- If `stats` contains `"auto"`:
  - for `"mokken"`: `misfit_final` = `Gnp` & `U3p`
  - for `"mirt"`: `misfit_final` = (`lpz` & `Gnp` & `U3p`) if any such rows exist, otherwise fallback to `lpz` only.

- If stats is manual (no "auto"): misfit_final is the AND over the selected statistics (if only one selected, it is used directly).

Polytomous PerFit statistics assume a **common design K** (number of categories) across items. This function uses objectrawK as the global design K and maps item responses to **0..K-1** without compressing per-item gaps (unused categories are allowed and do not trigger an error).

Value

An epmfd_misfit list with:

- scaled: the input epmfd_scaled object
- method: detected method ("mirt" or "mokken")
- stats: actually computed statistics (subset of c("lpz","Gnp","U3p"))
- thresholds: named list of lists with value and tail
- scores: named list of numeric score vectors per statistic
- table: a tibble with id, one logical column per statistic, misfit_any (OR over selected stats), and misfit_final (see Details)

Examples

```
library(epmfd)
data<-load_epmfd(sampledata)
scaling_data<-scale_epmfd(data)
misfit_result<-misfit_epmfd(scaling_data)
misfit_result
plot_misfit(misfit_result,threeD=TRUE)
```

---

plot_epmfd *Plot methods for epmfd objects*

Description

Quick visual summaries for three object classes:

- **epmfd_scaled**: Item-level retention summary (Kept vs Removed) and a quality-statistic histogram (either discrimination a for *mirt* or scalability H_i for *mokken*).
- **epmfd_misfit**: Bar plot of misfit counts per statistic and a global bar summarizing overall misfit ratio.
- **epmfd_clean**: Bar plot comparing remaining vs removed persons.

Usage

```
## S3 method for class 'epmfd_scaled'
plot(x, ...)

## S3 method for class 'epmfd_misfit'
plot(x, ...)

## S3 method for class 'epmfd_clean'
plot(x, ...)
```

Arguments

x	An epmfd_scaled, epmfd_misfit, or epmfd_clean object.
...	Additional aesthetics or layers forwarded to the underlying **ggplot2** geoms (e.g., alpha, linewidth).

Details

If the **patchwork** package is installed, paired plots are stacked vertically and returned as a single *patchwork* object; otherwise a list of two ggplot2 objects is returned.

Value

A single ggplot2 object, a patchwork object (if available), or a list of ggplot2 objects—depending on the class and whether combined layout is possible.

Dependencies

These methods use **ggplot2**. For epmfd_scaled objects fitted with *mirt*, the method accesses model coefficients via **mirt** if that package is installed (it is not required for *mokken*). Stacking multiple plots uses **patchwork** when available.

See Also

plot_misfit for 2D/3D scatter visualizations of person-level misfit, and misfit_epmfd / clean_epmfd for producing the inputs to these plots.

Examples

```
# Scaled object
p_scaled <- plot(scaled_obj)              # item retention + quality histogram

# Misfit object
p_mf <- plot(misfit_obj)                  # per-statistic counts + overall ratio

# Cleaned object
p_cl <- plot(clean_obj)                   # remaining vs removed persons

# Add ggplot2 options through '...'
```

```
plot(misfit_obj, alpha = 0.8)
```

`plot_misfit` *Plot person misfit in 2D/3D using stored thresholds*

Description

`plot_misfit()` visualizes person-level misfit statistics stored in an `epmfd_misfit` object. It supports:

- **2D**: scatter plots for all pairwise combinations of the selected statistics (or a single 2D plot if exactly two are given). Points are coloured by the joint exceedance logic (see `any`). Axis titles are the statistic names; the main title shows the cut-offs in parentheses, and dashed lines mark the cut-offs per axis.
- **3D**: an interactive scatter using **plotly** if three statistics are supplied; optionally adds three semi-transparent planes at the x/y/z cut-offs when `planes = TRUE`.

Usage

```
plot_misfit(
  object,
  stats = NULL,
  threeD = FALSE,
  any = FALSE,
  planes = TRUE,
  label_ids = FALSE,
  ...
)
```

Arguments

`object`	An `epmfd_misfit` or `epmfd_clean` object. If `epmfd_clean` is supplied, its `$misfit` component is used.
`stats`	Character vector of length 2 or 3 naming statistics found in `object$scores` (e.g., `c("Gnp","U3p","lpz")`). If `NULL`, the first up to three available statistics are used.
`threeD`	Logical. If `TRUE` and three statistics are available, a 3D **plotly** plot is drawn; otherwise the function falls back to 2D and emits a warning.
`any`	Logical. Colouring rule: • `FALSE` (default): only two classes - *all* cut-offs exceeded (red) vs *none* exceeded (blue). • `TRUE`: adds an intermediate class (orange) for *partial* exceedance (in 2D: exactly one; in 3D: one or two).

planes	Logical (3D only). If TRUE, draw three semi-transparent planes at the x, y, and z cut-off values; if FALSE, no planes are shown. Ignored for 2D plots.
label_ids	Logical. If TRUE, label points by id in 2D plots (uses **ggrepel** when available).
...	Additional aesthetics passed to ggplot2::geom_point() (2D) or plotly::add_markers() (3D), such as alpha, size, etc.

Details

Cut-off logic. For each selected statistic, a person is deemed to exceed if its score is greater than the cut-off for upper-tailed statistics or less than the cut-off for lower-tailed statistics. In 2D, dashed vertical and horizontal lines indicate the cut-offs; the plot title shows "Y (cutY) vs X (cutX)" with formatted values. In 3D, axis titles include the cut-off values in parentheses, and (optionally) three grey planes make the cut-offs explicit.

Returned value. With two statistics, a single ggplot is returned; with three statistics and threeD = FALSE, a named list of ggplots is returned for all 2D pairs. With threeD = TRUE and three statistics, a plotly object is returned.

Dependencies. This function uses **ggplot2** for 2D plots and, for 3D, **plotly** (required only when threeD = TRUE). Optional labels in 2D use **ggrepel** when installed.

Value

A ggplot object (2D), a named list of ggplots (all 2D pairs), or a plotly object (3D), depending on stats and threeD.

See Also

misfit_epmfd() for computing statistics and thresholds; clean_epmfd() for removing misfitting persons.

Examples

```
# Suppose 'mf' is an epmfd_misfit with scores Gnp, U3p, lpz

# 2D: single plot
plot_misfit(mf, stats = c("Gnp","U3p"), any = TRUE)

# 2D: all pairwise plots
plot_misfit(mf, stats = c("Gnp","U3p","lpz"))

# 3D: with cut-off planes
plot_misfit(mf, stats = c("Gnp","U3p","lpz"), threeD = TRUE, planes = TRUE)

# 3D: points only (no planes)
plot_misfit(mf, stats = c("Gnp","U3p","lpz"), threeD = TRUE, planes = FALSE)
```

print.epmfd_misfit *Print Method for epmfd_misfit Objects*

Description

Prints summary information for an `epmfd_misfit` object.

Usage

```
## S3 method for class 'epmfd_misfit'
print(x, ...)
```

Arguments

`x`	An object of class `epmfd_misfit`.
`...`	Further arguments passed to or from other methods.

Value

The input object `x`, returned (invisibly) after printing.

sampledata *Example Polytomous Response Data*

Description

A small toy dataset included in the **epmfd** package, containing polytomous item responses from simulated persons.

Usage

```
sampledata
```

Format

A data frame with 20 persons (rows) and 6 items (columns). Each item takes ordered values 1–5.

Examples

```
data(sampledata)
head(sampledata)
```

`scale_epmfd` *Scale polytomous item responses*

Description

`scale_epmfd()` fits either a parametric graded response model (GRM, via **mirt**) or a nonparametric Mokken model (via **mokken**) to polytomous item-response data and filters out weak items based on user-specified thresholds.

Usage

```
scale_epmfd(
  object,
  method = c("auto", "mirt", "mokken"),
  a_thr = 0.5,
  H_thr = 0.3
)
```

Arguments

`object`	An `epmfd_raw` object created by `load_epmfd()`.
`method`	Scaling method. One of: • `"mirt"`: fit a one-factor graded response model (GRM). • `"mokken"`: perform nonparametric Mokken scale analysis. • `"auto"` (default): choose based on sample size (`n >= 500` → GRM, otherwise Mokken).
`a_thr`	Numeric. Threshold for item discrimination parameter `a` when using GRM (default = 0.5). Items with `a < a_thr` are removed.
`H_thr`	Numeric. Threshold for item scalability coefficient `H_i` when using Mokken analysis (default = 0.3). Items with `H_i < H_thr` are removed.

Details

The function converts ordered factors to numeric before analysis.

- For GRM (`mirt`), items are filtered by their discrimination parameter `a`. The overall model fit is attempted using `mirt::M2()`; if this fails (e.g., due to insufficient df), a warning is issued and `model_fit = NULL`.
- For Mokken, item scalability coefficients `H_i` are computed and compared to `H_thr`.

Value

An object of class `epmfd_scaled`, a list containing:

- `raw`: the original `epmfd_raw` object
- `method`: scaling method actually used (`"mirt"` or `"mokken"`)

- kept: names of items retained
- removed: names of items removed
- model: fitted GRM model (for "mirt"), else NULL
- ai: item parameter estimates (for "mirt")
- a_thr: discrimination threshold used (for "mirt")
- model_fit: results of mirt::M2() (if available)
- Hi: vector of item scalability coefficients (for "mokken")
- H_thr: scalability threshold used (for "mokken")
- items: the vector containing all items names.

See Also

load_epmfd(), misfit_epmfd(), plot.epmfd_scaled()

Examples

```
library(epmfd)
data<-load_epmfd(sampledata)
scale_epmfd(data)
```

---

summary.epmfd_clean *Summary method for* epmfd_clean *objects*

Description

Summary method for epmfd_clean objects

Usage

```
## S3 method for class 'epmfd_clean'
summary(object, ...)
```

Arguments

object	An object of class epmfd_clean.
...	Further arguments (ignored).

Value

Invisibly returns a named list with summary numbers.

Index

∗ **datasets**
sampledata, 12

clean_epmfd, 2, *9*
clean_epmfd(), *11*

export_epmfd, 3

load_epmfd, 6
load_epmfd(), *13, 14*

misfit_epmfd, 7, *9*
misfit_epmfd(), *2, 3*, *6*, *11*, *14*

plot.epmfd_clean *(*plot_epmfd*)*, 8
plot.epmfd_misfit *(*plot_epmfd*)*, 8
plot.epmfd_scaled *(*plot_epmfd*)*, 8
plot.epmfd_scaled(), *14*
plot_epmfd, 8
plot_misfit, *9*, 10
print.epmfd_misfit, 12

sampledata, 12
saveRDS(), *5*
scale_epmfd, 13
scale_epmfd(), *2, 3*, *6*
summary.epmfd_clean, 14