Package 'ItemRest'

May 7, 2026

Title Automated Item Removal Strategies for Exploratory Factor Analysis

Version 0.2.5

Description Automates the identification and comparative evaluation of item-removal strategies in exploratory factor analysis, producing transparent summaries (explained variance, loading ranges, reliability) to support comfortable, reproducible decisions. The criteria are based on best practices and established heuristics (e.g., Costello & Osborne (2005) <doi:10.7275/jyj1-4868>, Howard (2016) <doi:10.1080/10447318.2015.1087664>). Includes flexible thresholds for factor loadings (min_loading) and cross-loading differences (loading_diff).

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1)

Imports gtools, psych, qgraph, stats, utils

URL https://github.com/ahmetcaliskan1987/ItemRest

BugReports https://github.com/ahmetcaliskan1987/ItemRest/issues

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.3

NeedsCompilation no

Author Ahmet Çalışkan [aut, cre], Abdullah Faruk Kılıç [aut]

Maintainer Ahmet Çalışkan <ahmetcaliskan1987@gmail.com>

Repository CRAN

Date/Publication 2026-04-14 13:30:08 UTC

Contents

itemrest 2
print.itemrest_result 3

Index **5**

`itemrest` *Evaluate Item Removal Strategies for Exploratory Factor Analysis (EFA)*

Description

This function automates the process of identifying low-quality items (those with low factor loadings or significant cross-loadings) in an Exploratory Factor Analysis (EFA). It systematically tests various combinations of removing these problematic items and evaluates the impact on model fit, returning a comprehensive summary of all tested strategies.

Usage

```
itemrest(
  data,
  cor_method = "pearson",
  n_factors = NULL,
  extract = "uls",
  rotate = "oblimin",
  min_loading = 0.3,
  loading_diff = 0.1
)
```

Arguments

`data`	A numeric `data.frame` or `matrix` for the analysis.
`cor_method`	The correlation method to use, e.g., `"pearson"` or `"polychoric"`.
`n_factors`	The number of factors. If `NULL`, it's determined automatically by parallel analysis.
`extract`	The factor extraction (estimation) method. See `psych::fa`. Default is `"uls"`.
`rotate`	The rotation method. See `psych::fa`. Default is `"oblimin"`.
`min_loading`	Minimum factor loading threshold. Default is 0.30.
`loading_diff`	Threshold for cross-loading identification. Default is 0.10. Can be "howard".

Value

An object of class `itemrest_result`. This is a list containing the following components:

`descriptive_stats`	Basic descriptive statistics of the input data.
`initial_efa`	The results of the initial EFA before any items are removed.
`problem_items`	A list of items identified as low-loading or cross-loading.
`removal_summary`	A data.frame summarizing the results of all tested removal strategies.

optimal_strategy
: The best-performing strategy that resulted in a clean factor structure (no cross-loadings).

settings
: A list of the settings used for the analysis.

Examples

```
# We will use the 'bfi' dataset from the 'psych' package.
# This requires the 'psych' package to be installed.
if (requireNamespace("psych", quietly = TRUE)) {
  data(bfi, package = "psych")

  # 1. Prepare the data: Select the personality items (first 25 columns)
  #    and remove rows with missing values for this example.
  example_data <- bfi[, 1:25]
  example_data <- na.omit(example_data)

  # 2. Run the item removal analysis.
  #    Based on theory, the Big Five model has 5 factors.
  results <- itemrest(
    data = example_data,
    n_factors = 5,
    cor_method = "pearson" # Data is not ordinal, so pearson is appropriate
  )

  # 3. Print the report for optimal strategies (default).
  print(results)

  # 4. Print the report for all tested strategies.
  print(results, report = "all")
}
```

---

print.itemrest_result *Print method for* itemrest_result *class*

Description

Print method for itemrest_result class

Usage

```
## S3 method for class 'itemrest_result'
print(x, report = "optimal", ...)
```

Arguments

x
: An object of class itemrest_result.

report
: The type of report to generate: "optimal" (default) or "all".

...
: Other arguments (not used).

Value

No return value, called for side effects (prints the report to the console).

Examples

```
if (requireNamespace("psych", quietly = TRUE)) {
  data(bfi, package = "psych")
  example_data <- na.omit(bfi[, 1:25])
  results <- itemrest(example_data, n_factors = 5)

  # Print the default optimal report
  print(results)

  # Print the report of all strategies
  print(results, report = "all")
}
```

Index

itemrest, 2

print.itemrest_result, 3