Package: ipf (via r-universe)

July 9, 2026

Title Iterative Proportional Fitting

Version 0.0.1

Description Fast raking for survey weighting. The computational core is written in Rust for speed. Supports multiple raking variables, automatic variable selection, weight bounding, and comprehensive diagnostics.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Config/rextendr/version 0.4.2

SystemRequirements Cargo (Rust's package manager), rustc >= 1.65

Depends R (>= 4.2)

Imports cli, generics, tibble

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/testthat/edition 3

URL https://christophertkenny.com/ipf/

VignetteBuilder knitr

LazyData true

Config/Needs/website christopherkenny/ctktemplate

NeedsCompilation yes

Author Christopher T. Kenny [aut, cre] (ORCID: <https://orcid.org/0000-0002-9386-6860>)

Maintainer Christopher T. Kenny <ctkenny@proton.me>

Config/pak/sysreqs libclang-dev

Repository https://cran.r-universe.dev

Date/Publication 2026-07-09 09:40:02 UTC

RemoteUrl https://github.com/cran/ipf

RemoteRef HEAD

RemoteSha 7f2571e0d74da7b6f3c611ff29a5c0838d674733

Contents

`anes24` *ANES 2024 Time Series Study (subset)*

Description

A subset of the 2024 American National Election Study (ANES) Time Series face-to-face sample, containing demographic and vote choice variables for 966 respondents. Useful for demonstrating survey raking workflows.

Usage

```
anes24
```

Format

A tibble with 966 rows and 7 columns:

state Two-letter US state abbreviation. `NA` for respondents whose state is not identified (106 missing).

sex Respondent sex: `"Male"` or `"Female"` (5 missing).

race Race/ethnicity: `"White"`, `"Black"`, `"Hispanic"`, `"Asian"`, or `"Other"` (11 missing).

income Household income bracket: `"Under $50k"`, `"$50k-$100k"`, or `"Over $100k"` (47 missing).

education Education: `"Less than HS"`, `"High school"`, `"Some college"`, `"Bachelor's"`, or `"Graduate"` (451 missing).

married Marital status: `"Married"`, `"Widowed"`, `"Divorced"`, `"Separated"`, or `"Never married"` (277 missing).

presidential 2024 presidential vote choice: `"Harris"` or `"Trump"` (335 missing).

Source

https://electionstudies.org/data-center/2024-time-series-study/

References

American National Election Studies. 2025. *ANES 2024 Time Series Study Full Release* (dataset and documentation). August 8, 2025 version. https://www.electionstudies.org/

`augment.ipf_rake` *Augment data with raked weights*

Description

Returns the original data frame with a `.weight` column appended.

Usage

```
## S3 method for class 'ipf_rake'
augment(x, ...)
```

Arguments

`x`	An `ipf_rake` object.
`...`	Additional arguments (ignored).

Value

A tibble with all original columns plus `.weight`.

Examples

```
data <- data.frame(
  gender = sample(c('M', 'F'), 100, replace = TRUE, prob = c(0.6, 0.4))
)
targets <- list(gender = c(M = 0.5, F = 0.5))
result <- rake(data, targets)
augment(result)
```

`design_effect` *Compute design effect and effective sample size*

Description

The design effect (deff) measures the variance inflation factor due to unequal weighting. The effective sample size is `n / deff`.

Usage

```
design_effect(weights)
```

Arguments

weights	Numeric weight vector.

Value

A list with deff (design effect) and n_eff (effective sample size).

Examples

```
w <- c(1.2, 0.8, 1.5, 0.5, 1.0)
design_effect(w)
```

---

find_discrepant_vars *Find discrepant variables and their aggregate discrepancy scores*

Description

Calculates discrepancy between the current weighted distribution and target distributions for each variable, then aggregates using the chosen method.

Usage

```
find_discrepant_vars(
  data,
  targets,
  weights,
  choosemethod = "total",
  na_method = c("exclude", "bucket")
)
```

Arguments

data	Data frame.
targets	Named list of named numeric target vectors (proportions).
weights	Numeric weight vector.
choosemethod	Method for aggregating per-category discrepancies. One of "total", "max", "average", "totalsquared", "maxsquared", "averagesquared".
na_method	How to handle NA values. "exclude" skips NA cases from that margin. "bucket" treats missing values as an implicit extra category.

Value

Named numeric vector of aggregate discrepancy per variable.

Examples

```
data <- data.frame(
  gender = sample(c('M', 'F'), 100, replace = TRUE, prob = c(0.6, 0.4)),
  age = sample(c('young', 'old'), 100, replace = TRUE, prob = c(0.7, 0.3))
)
targets <- list(
  gender = c(M = 0.5, F = 0.5),
  age = c(young = 0.6, old = 0.4)
)
find_discrepant_vars(data, targets, weights = rep(1, 100))
```

---

glance.ipf_rake *Glance at an ipf_rake object*

Description

Returns a single-row tibble with summary statistics.

Usage

```
## S3 method for class 'ipf_rake'
glance(x, ...)
```

Arguments

x	An ipf_rake object.
...	Additional arguments (ignored).

Value

A single-row tibble with columns: converged, iterations, max_prop_err, deff, n_eff, n_obs, n_vars.

Examples

```
data <- data.frame(
  gender = sample(c('M', 'F'), 100, replace = TRUE, prob = c(0.6, 0.4))
)
targets <- list(gender = c(M = 0.5, F = 0.5))
result <- rake(data, targets)
glance(result)
```

print.ipf_rake *Print an ipf_rake object*

Description

Print an ipf_rake object

Usage

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

Arguments

x	An ipf_rake object.
...	Additional arguments (ignored).

Value

Invisibly returns x.

Examples

```
data <- data.frame(
  gender = sample(c('M', 'F'), 100, replace = TRUE, prob = c(0.6, 0.4))
)
targets <- list(gender = c(M = 0.5, F = 0.5))
result <- rake(data, targets)
print(result)
```

rake *Iterative proportional fitting (raking)*

Description

Adjusts survey weights so that weighted marginal distributions match known population targets. Supports automatic variable selection, iterative re-raking, and weight bounding.

Usage

```
rake(
  data,
  targets,
  base_weights = NULL,
  cap = 5,
  bounds = NULL,
```

```
  type = c("nolim", "pctlim", "nlim"),
  pctlim = 0.05,
  nlim = 5L,
  choosemethod = c("total", "max", "average", "totalsquared", "maxsquared",
    "averagesquared"),
  na_method = c("exclude", "bucket"),
  iterate = TRUE,
  max_iter = 1000L,
  tol = 1e-06,
  verbose = FALSE,
  diagnostics_every = 0L
)
```

Arguments

data	A data frame or tibble containing the survey data.
targets	A named list of named numeric vectors specifying target proportions for each raking variable. Names of the list must match column names in data. Each vector's names must match the levels of the corresponding variable. Values should sum to 1 (proportions); if not, they are normalized with a warning.
base_weights	Optional numeric vector of base (design) weights. If NULL (default), uniform weights of 1 are used. Centered to mean 1 before raking.
cap	Maximum weight value (ratio cap). Weights exceeding this value are trimmed and all weights are renormalized. Default 5. Ignored if bounds is specified.
bounds	Optional numeric vector of length 2, c(lo, hi), specifying minimum and maximum weight bounds. Overrides cap.
type	Variable selection method: • "nolim" (default): use all variables in targets. • "pctlim": use only variables with discrepancy >= pctlim. • "nlim": use the nlim most discrepant variables.
pctlim	Discrepancy threshold for type = "pctlim". Default 0.05 (5 percentage points).
nlim	Number of variables for type = "nlim". Default 5.
choosemethod	Method for aggregating per-category discrepancies into a single variable score. One of "total", "max", "average", "totalsquared", "maxsquared", "averagesquared".
na_method	How to handle NA values in raking variables. "exclude" (default): targets are proportions among non-NA cases only; NA cases are invisible to that margin. Matches anesrake. "bucket": NAs become a frozen extra category; their total weight is preserved and the named targets are rescaled to the remaining non-NA mass.
iterate	Logical. If TRUE and type = "pctlim", re-check discrepancies after raking and add newly discrepant variables, repeating up to 10 times. Default TRUE.
max_iter	Maximum number of raking iterations. Default 1000.
tol	Convergence tolerance (max proportional error). Default 1e-6.
verbose	Logical. If TRUE, print iteration progress. Default FALSE.

diagnostics_every
: Record per-margin diagnostics every k iterations. 0 means only baseline. Default 0.

Value

An ipf_rake object (S3 class) containing:

- weights: final raked weight vector
- data: the input data frame
- converged: logical
- iterations: number of iterations
- max_prop_err: final max proportional error
- targets: normalized targets used
- vars_used: character vector of variables raked on
- base_weights: original base weights
- type, choosemethod, na_method, cap: settings used
- deff, n_eff: design effect and effective sample size
- diagnostics: tibble of per-iteration diagnostics

Examples

```
data <- data.frame(
  gender = sample(c('M', 'F'), 100, replace = TRUE, prob = c(0.6, 0.4)),
  age = sample(c('young', 'old'), 100, replace = TRUE, prob = c(0.7, 0.3))
)
targets <- list(
  gender = c(M = 0.5, F = 0.5),
  age = c(young = 0.6, old = 0.4)
)
result <- rake(data, targets)
print(result)
```

---

summary.ipf_rake *Summarize an ipf_rake object*

Description

Produces a detailed summary including per-variable diagnostic tables showing target vs. achieved distributions.

Usage

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

Arguments

object	An ipf_rake object.
...	Additional arguments (ignored).

Value

Invisibly returns a list with convergence info, weight summary, design effect, and per-variable assessment tibbles.

Examples

```
data <- data.frame(
  gender = sample(c('M', 'F'), 100, replace = TRUE, prob = c(0.6, 0.4))
)
targets <- list(gender = c(M = 0.5, F = 0.5))
result <- rake(data, targets)
summary(result)
```

---

tidy.ipf_rake *Tidy an ipf_rake object*

Description

Returns a one-row-per-variable-per-level tibble with target proportions, weighted proportions, and discrepancy.

Usage

```
## S3 method for class 'ipf_rake'
tidy(x, ...)
```

Arguments

x	An ipf_rake object.
...	Additional arguments (ignored).

Value

A tibble with columns: variable, level, target, weighted_pct, discrepancy.

Examples

```
data <- data.frame(
  gender = sample(c('M', 'F'), 100, replace = TRUE, prob = c(0.6, 0.4))
)
targets <- list(gender = c(M = 0.5, F = 0.5))
result <- rake(data, targets)
tidy(result)
```

`weight_assess` *Assess weight quality with diagnostic tables*

Description

Produces a per-variable diagnostic table comparing target distributions to unweighted and weighted distributions.

Usage

```
weight_assess(
  data,
  targets,
  weights,
  base_weights = NULL,
  na_method = c("exclude", "bucket")
)
```

Arguments

`data`	Data frame.
`targets`	Named list of named numeric target vectors (proportions).
`weights`	Final raked weight vector.
`base_weights`	Original base weights before raking. If `NULL`, uses uniform weights.
`na_method`	How to handle `NA` values. `"exclude"` skips NA cases from that margin. `"bucket"` treats missing values as an implicit extra category.

Value

Named list of tibbles, one per variable.

Examples

```
data <- data.frame(
  gender = sample(c('M', 'F'), 100, replace = TRUE, prob = c(0.6, 0.4))
)
targets <- list(gender = c(M = 0.5, F = 0.5))
result <- rake(data, targets)
weight_assess(data, targets, result$weights)
```

Index