Package 'mpindex'

June 15, 2026

Type Package

Title Multidimensional Poverty Index (MPI) via the Alkire-Foster Method

Version 0.3.0

Author Bhas Abdulsamad [aut, cre, cph] (ORCID: <https://orcid.org/0009-0002-5891-8124>)

Maintainer Bhas Abdulsamad <aeabdulsamad@gmail.com>

Description Estimate Multidimensional Poverty Index (MPI) measures from household survey microdata using the Alkire-Foster dual-cutoff counting method (Alkire and Foster, 2011). Load indicator specifications from CSV, Excel, JSON, or plain-text files; compute the headcount ratio (H), intensity (A), and MPI = H x A across any subgroup; and export results to formatted Excel reports. Supports complex survey designs — stratification, clustering, and probability weights — and optionally appends design-based standard errors and confidence intervals.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports dplyr, stringr, jsonlite, tibble, openxlsx, tsg, rlang, lifecycle

Suggests tidyr, survey, testthat (>= 3.0.0), knitr, rmarkdown, gt

Config/testthat/edition 3

RoxygenNote 7.3.3

BugReports https://github.com/yng-me/mpindex/issues

URL https://github.com/yng-me/mpindex, https://yng-me.github.io/mpindex/

VignetteBuilder knitr

Depends R (>= 4.1.0)

NeedsCompilation no

Repository CRAN

Date/Publication 2026-06-15 07:50:15 UTC

Contents

compute_mpi 2
define_deprivation 4
define_mpi_specs 6
deprived 7
df_household 8
df_household_roster 9
global_mpi_specs 10
save_mpi 11
use_global_mpi_specs 12

Index **13**

`compute_mpi` *Compute Multidimensional Poverty Index (MPI)*

Description

The primary single-call API for computing the MPI using the Alkire-Foster (AF) counting method. Deprivation cutoffs are specified inline using the `deprived` helper, making the workflow self-contained and readable.

Usage

```
compute_mpi(
  .data,
  mpi_specs,
  deprivations,
  ...,
  by = NULL,
  include_deprivation_matrix = FALSE,
  weight = NULL,
  strata = NULL,
  cluster = NULL,
  fpc = NULL,
  survey_design = NULL,
  inference = FALSE,
  ci_level = 0.95
)
```

Arguments

`.data`	A data frame where each row is the unit of analysis.
`mpi_specs`	MPI specifications from `define_mpi_specs`.
`deprivations`	A named list of `deprived` calls. Each name must exactly match a `variable` in `mpi_specs`.

`...`	*(Optional)* Extra columns to carry through into the deprivation matrix (tidyselect). These columns are included in the matrix after by columns but do **not** affect grouping of summary outputs. Also catches old dotted argument names.
`by`	*(Optional)* Columns to group summary outputs by (tidyselect), e.g. `c(region, sex)`. These columns are also included in the deprivation matrix, before any `...` columns.
`include_deprivation_matrix`	Whether to include deprivation matrices. Default `FALSE`.
`weight`	Name of the sampling-weight column in `.data`. When supplied, all estimates are survey-weighted. Requires the **survey** package.
`strata`	Name of the stratum column in `.data`.
`cluster`	Name of the cluster / PSU column in `.data`.
`fpc`	Name of the finite-population correction column in `.data`.
`survey_design`	A pre-built `survey::svydesign()` object. Provide either `weight` / `strata` / `cluster` / `fpc` *or* `survey_design`, not both.
`inference`	Logical. When `TRUE` (and a survey design is supplied), standard errors and confidence intervals are appended as `*_se`, `*_ci_low`, `*_ci_high` columns. Default `FALSE`.
`ci_level`	Confidence level for intervals. Default `0.95`.

Value

A named list of class `mpi_output` with components:

`$index` Named list keyed by `k_*`: MPI, H, A, n.

`$contribution` Named list keyed by `k_*`: contribution by indicator/dimension.

`$headcount_ratio` Named list with `uncensored` and per-`k_*` censored ratios.

`$deprivation_matrix` Named list with `uncensored` and per-`k_*` matrices.

See Also

define_mpi_specs, deprived, save_mpi

Examples

```
specs <- define_mpi_specs(
  system.file("extdata", "global-mpi-specs.csv", package = "mpindex"),
  uid = "uuid"
)

## Not run:
mpi_result <- compute_mpi(
  df_household,
  mpi_specs = specs,
  deprivations = list(
    nutrition = deprived(
      undernourished == 1 & age < 70,
```

```
      .data = df_household_roster,
      collapse_fn = max
    ),
    child_mortality = deprived(with_child_died == 1),
    year_schooling = deprived(
      completed_6yrs_schooling == 2,
      .data = df_household_roster,
      collapse_fn = max
    ),
    school_attendance = deprived(
      attending_school == 2 & age %in% c(5:24),
      .data = df_household_roster,
      collapse_fn = max
    ),
    cooking_fuel   = deprived(cooking_fuel %in% c(4:6, 9)),
    sanitation     = deprived(toilet > 1),
    drinking_water = deprived(drinking_water == 2),
    electricity    = deprived(electricity == 2),
    housing        = deprived(
      roof %in% c(5, 7, 9) | walls %in% c(5, 8, 9, 99) == 2 | floor %in% c(5, 6, 9)
    ),
    assets = deprived(!(
      (asset_tv + asset_telephone + asset_mobile_phone + asset_computer +
         asset_animal_cart + asset_bicycle + asset_motorcycle +
         asset_refrigerator) > 1 &
        (asset_car + asset_truck) > 0
    ))
  )
)

## End(Not run)
```

`define_deprivation` *Define deprivation cutoffs*

Description

Sets a deprivation cutoff for a single indicator. For each unit of analysis, the result is `0` (not deprived), `1` (deprived), or `NA` (missing). An additional weighted column (indicator value × weight) is also computed.

Usage

```
define_deprivation(
  .data,
  indicator,
  cutoff,
  mpi_specs = NULL,
  collapse_fn = NULL,
```

```
  set_na_equal_to = 0,
  ...
)
```

Arguments

.data	A data frame or tibble.
indicator	Name of the indicator as defined in the MPI Specs (must exactly match the variable column).
cutoff	A logical expression that evaluates to TRUE for deprived units.
mpi_specs	MPI specifications from define_mpi_specs.
collapse_fn	An optional function to collapse roster-level data to the unit-of-analysis level (e.g. max). NULL (default) means no collapsing.
set_na_equal_to	Coerce NA values to 0 (not deprived, default) or 1 (deprived).
...	Reserved; passing old dotted names triggers a helpful error.

Value

A data frame with columns *_unweighted and *_weighted.

See Also

define_mpi_specs

Examples

```
specs_file <- system.file(
 "extdata",
 "global-mpi-specs.csv",
 package = "mpindex"
)

specs <- define_mpi_specs(specs_file, uid = "uuid")

df_household |>
  define_deprivation(
    indicator  = drinking_water,
    cutoff     = drinking_water == 2,
    mpi_specs  = specs
  )

df_household_roster |>
  define_deprivation(
    indicator   = school_attendance,
    cutoff      = attending_school == 2,
    mpi_specs   = specs,
    collapse_fn = max
  )
```

`define_mpi_specs` *Define MPI specifications: dimensions, indicators, and weights*

Description

Use to define MPI dimensions, indicators and its corresponding weights using any of the supported file types: `.xlsx` (Excel), `.json`, `.csv`, or `.txt` (TSV). You can also set the poverty cutoff or list of poverty cutoffs.

Usage

```
define_mpi_specs(
  mpi_specs_file = NULL,
  indicators = NULL,
  poverty_cutoffs = NULL,
  unit_of_analysis = NULL,
  uid = NULL,
  source_of_data = NULL,
  names_separator = getOption("mpindex.options")$names_separator %||% "__",
  save_as_global_options = FALSE,
  ...
)
```

Arguments

`mpi_specs_file`	Path to a `.xlsx`, `.json`, `.csv`, or `.txt` (TSV) file. The file must contain columns: `Dimension`, `Indicator`, `Variable`, `Weight`, and optionally `Description`.
`indicators`	A data frame of MPI indicators. Alternative to `mpi_specs_file` when you prefer to define indicators inline.
`poverty_cutoffs`	Single value or vector of poverty cutoffs (k). All values must be in (0, 1]. Default is `NULL` which will be automatically set to `1/n`, where `n` is the total number of dimensions.
`unit_of_analysis`	e.g. `"individuals"`, `"households"`. Default `NULL`.
`uid`	Column name containing the unique ID (unit of analysis).
`source_of_data`	Source of data; used in output footnotes.
`names_separator`	**[Deprecated]** Column separator for the header hierarchy.
`save_as_global_options`	**[Deprecated]** No longer has any effect.
`...`	Reserved for forward-compatibility; passing old dotted argument names (e.g. `.uid`) triggers a helpful error.

Value

An `mpi_specs` object. Pass this directly as the `mpi_specs` argument in `compute_mpi`, `define_deprivation`, and `save_mpi`.

See Also

compute_mpi

Examples

```
specs_file <- system.file(
  "extdata",
  "global-mpi-specs.csv",
  package = "mpindex"
)
system.file("extdata", package = "mpindex") |> list.files()
```

`deprived` *Specify a deprivation cutoff for use in* `compute_mpi`

Description

A helper that captures a bare deprivation cutoff expression and optional per-indicator settings for use inside the `deprivations` argument of `compute_mpi`.

Usage

```
deprived(.cutoff, .data = NULL, collapse_fn = NULL, set_na_equal_to = 0, ...)
```

Arguments

`.cutoff`	A bare logical expression evaluated against the indicator's data frame. Rows where this evaluates to `TRUE` are considered deprived.
`.data`	An optional data frame to use for this indicator instead of the primary `.data` passed to `compute_mpi`. Useful when one or more indicators are at a different unit of analysis (e.g. person-level roster).
`collapse_fn`	An optional function applied to collapse roster-level data to the unit-of-analysis level (e.g. `max` to flag a household as deprived if any member is deprived). If `NULL` (default), no collapsing is performed. NAs are removed before calling the function; if all values are `NA` the result is `NA`.
`set_na_equal_to`	Coerce `NA` deprivation values to `0` (not deprived, default) or `1` (deprived).
`...`	Reserved; passing old dotted names (e.g. `.collapse_fn`) triggers a helpful error.

Value

An object of class `mpi_d`.

See Also

compute_mpi

Examples

```
deprived(drinking_water == 2)
deprived(undernourished == 1 & age < 70, .data = df_household_roster, collapse_fn = max)
```

---

df_household *Sample dataset of households*

Description

This is a synthetic dataset containing household information primarily used for demonstration purposes on how to use the `mpindex` package.

Usage

```
df_household
```

Format

A tibble with 198 rows and 21 variables:

uuid Unique ID

class Urbanity: `Rural` or `Urban`

drinking_water Acess to drinking water: `1` - improved; `2` - unimproved

toilet Service level of toilet or sanitation facility: `1` - basic; `2` - limited; `3` - unimproved; `4` - open defecation

with_child_died With at least one (1) child died in the last five (5) years: `1` - with child died; `2` - without child died

roof Main construction material of the roof: `1` - galvanized iron/aluminum; `2` - concrete/clay tile; `3` - half galvanized iron and half concrete; `4` - wood/bamboo; `5` - cogon/nipa/anahaw; `6` - asbestos; `7` - makeshift/salvaged/improvised materials; `9` - other construction material

walls Main construction material of the outer walls: `1` - concrete/brick/stone; `2` - wood; `3` - half concrete/brick/stone and half wood; `4` - Galvanized iron/aluminum; `5` - bamboo/sawali/cogon/nipa; `6` - asbestos; `7` - glass; `8` - makeshift/salvaged/improvised materials; `9` - none; `10` - concrete hollow blocks; `11` - concrete hollow blocks/wood; `12` - shear walls; `99` - other construction material

floor Main construction material of the floor: 1 - concrete; 2 - wood; 3 - coconut lumber; 4 - bamboo; 5 - earth/sand/mud; 6 - makeshift/salvaged/improvised materials; 9 - other construction material

electricity Access to electricity: 1 - with access to electricity; 2 - without access to electricity

cooking_fuel Fuel use for cooking: 1 - electricity; 2 - kerosene (gaas); 3 - liquified petroleum gas (LPG); 4 - charcoal; 5 - wood; 6 - none; 9 - other cooking fuel such as dung, agricultural crop, or shrubs

asset_radio Number of working radio owned by the household

asset_tv Number of working television owned by the household

asset_telephone Number of working telephone owned by the household

asset_mobile_phone Number of working mobile phone owned by the household

asset_computer Number of working computer owned by the household

asset_animal_cart Number of animal carts owned by the household

asset_bicycle Number of bicycle owned by the household

asset_motorcycle Number of motorcylce owned by the household

asset_refrigerator Number of working refrigerator owned by the household

asset_car Number of car owned by the household

asset_truck Number of trucks owned by the household

See Also

df_household_roster

Examples

```
df_household
```

---

df_household_roster *Sample dataset of household members*

Description

This dataset contains a many-to-one relationship with the df_household dataset. Hence, you can apply joins using the uuid.

Usage

```
df_household_roster
```

Format

A tibble with 905 rows and 8 variables:

uuid Unique ID

line_number Number identifier for each member within the household

class Urbanity: `Rural` or `Urban`

sex Sex of the household member

age Age of the household member

attending_school Whether the household member (aged 5-24 years old) is currently attending school: `1` - currently attending; `2` - currently not attending

completed_6yrs_schooling Whether completed at least six (6) years of schooling: `1` - completed; `2` -not completed

undernourished Whether the household member (aged below 70 years old) is undernourished: `1` - undernourished; `2` - not undernourished

See Also

df_household

Examples

```
df_household_roster
```

`global_mpi_specs` *Load the built-in Global MPI specification*

Description

Returns the `mpi_specs` object for the standard Global MPI (10 indicators across Health, Education, and Living Standards). Assign the result and pass it explicitly as `mpi_specs`.

Usage

```
global_mpi_specs(..., poverty_cutoffs = 1/3)
```

Arguments

`...`	Additional arguments passed to `define_mpi_specs`, e.g. `uid`, `poverty_cutoffs`.
`poverty_cutoffs`	Single value or vector of poverty cutoffs (k). All values must be in (0, 1]. Default is `1/3`.

Value

An `mpi_specs` object.

Examples

```
mpi_specs <- global_mpi_specs(uid = "uuid")
```

`save_mpi` *Save MPI output*

Description

Save the MPI output to an Excel file using the tsg package for publication-ready table formatting.

Usage

```
save_mpi(
  mpi_output,
  mpi_specs,
  ...,
  filename = NULL,
  include_specs = FALSE,
  overall_label = "Overall",
  facade = tsg::get_tsg_facade()
)
```

Arguments

`mpi_output`	An object derived from compute_mpi.
`mpi_specs`	MPI specifications defined in `define_mpi_specs`.
`...`	Reserved; passing old dotted names triggers a helpful error.
`filename`	Output filename. The `.xlsx` extension is added automatically when missing. Defaults to `"MPI Results.xlsx"` in the current working directory.
`include_specs`	Whether to include MPI specification as a separate sheet. Defaults to `FALSE`.
`overall_label`	Overall label to assign when grouping is defined in `compute_mpi()` through by argument. Default is `"Overall"`. Accepts vector of elements matching the number of grouping variables defined.
`facade`	`tsg` facade (see add_facade).

Value

Returns the normalized file path of the generated Excel file.

Examples

```
## Not run:
mpi_result <- compute_mpi(df_household, mpi_specs = specs, deprivations = deps)
save_mpi(mpi_result, mpi_specs = specs, filename = "MPI Sample Output")

## End(Not run)
```

`use_global_mpi_specs` *Use Global MPI specification*

Description

[Deprecated] Please use `global_mpi_specs` instead.

Usage

```
use_global_mpi_specs(...)
```

Arguments

`...`	Passed to `global_mpi_specs`.

Value

An `mpi_specs` object.

Examples

```
## Not run:
# Deprecated - use global_mpi_specs() instead
mpi_specs <- use_global_mpi_specs(uid = "uuid")

## End(Not run)
```

Index

∗ **datasets**
df_household, 8
df_household_roster, 9

add_facade, *11*

compute_mpi, 2, *7*, *8*, *11*

define_deprivation, 4, *7*
define_mpi_specs, *2*, *3*, *5*, 6, *10*, *11*
deprived, *2*, *3*, 7
df_household, 8, *9*, *10*
df_household_roster, *9*, 9

global_mpi_specs, 10, *12*

save_mpi, *3*, *7*, 11

use_global_mpi_specs, 12