Package 'SurveyNCD'

November 3, 2026

Title Survey-Weighted Analysis of Self-Reported Health Indicators

Version 0.2.0

Description Design-based analysis of population health survey data from the World Health Organization (WHO) Stepwise Approach to Non-Communicable Disease (NCD) Risk Factor Surveillance (STEPS), Demographic and Health Surveys (DHS), Multiple Indicator Cluster Surveys (MICS) and similar complex sample surveys, where chronic conditions are self-reported rather than coded with the International Classification of Diseases (ICD). Provides survey design constructors for DHS, MICS and STEPS microdata; design-based prevalence for any binary indicators and groupings; a self-reported multimorbidity count based on the Functional Comorbidity Index (FCI) of Groll et al. (2005) <doi:10.1016/j.jclinepi.2004.10.018>, with condition-pattern analysis, sensitivity bounds for missing answers, and direct age standardisation to the WHO World Standard Population; standard, generalized, Erreygers and Wagstaff concentration indices with design-based standard errors following Kakwani, Wagstaff and van Doorslaer (1997) <doi:10.1016/S0304-4076(96)01807-6> and O'Donnell et al. (2016) <doi:10.1177/1536867X1601600112>, concentration curves, and slope and relative indices of inequality; categorisation of DHS-coded child growth z-scores; publication-ready tables and plots. Simulated example data with a known population are included. Experimental helpers for choropleth maps and case-weighted gradient boosting with SHapley Additive exPlanations (SHAP) are also provided.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

VignetteBuilder knitr

Depends R (>= 3.5)

Imports dplyr, ggplot2, magrittr, rlang, stats, survey, tidyr

Suggests knitr, rmarkdown, sf, srvyr, testthat (>= 3.0.0), xgboost

Config/testthat/edition 3

URL https://github.com/StatAid-Research-Lab/SurveyNCD

BugReports https://github.com/StatAid-Research-Lab/SurveyNCD/issues

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.1

NeedsCompilation no

Author Sujon Mia [aut, cre] (ORCID: <https://orcid.org/0009-0001-1032-2818>), Md. Atiqul Islam [ctb] (Professor & Academic Mentor, Department of Statistics)

Maintainer Sujon Mia <sujonsgc@gmail.com>

Repository CRAN

Date/Publication 2026-11-03 09:00:23 UTC

Contents

autoplot.surveyncd . . . 3
child_anthro . . . 4
design_helpers . . . 5
fci_items . . . 6
mm_bounds . . . 7
mm_patterns . . . 8
mm_prevalence . . . 9
mm_standardize . . . 10
multimorbidity_index . . . 11
ncd_survey . . . 12
ncd_table . . . 14
plot_shap_summary . . . 15
recode_binary . . . 16
survey_concentration_curve . . . 17
survey_concentration_index . . . 18
survey_map_indicator . . . 20
survey_prevalence . . . 21
survey_shap . . . 22
survey_slope_index . . . 23
survey_xgboost . . . 24
who_anthro_score . . . 25
who_standard_population . . . 26

Index **28**

`autoplot.surveyncd` *Plot SurveyNCD results with ggplot2*

Description

`autoplot()` methods for the results of `survey_prevalence()`, `mm_prevalence()`, `survey_concentration_index()`, `survey_concentration_curve()` and `mm_patterns()`. Each returns a `ggplot` object that can be modified with the usual **ggplot2** layers.

Usage

```
## S3 method for class 'surveyncd_prevalence'
autoplot(object, ...)

## S3 method for class 'surveyncd_mm_prevalence'
autoplot(object, ...)

## S3 method for class 'surveyncd_ci'
autoplot(object, ...)

## S3 method for class 'surveyncd_curve'
autoplot(object, ...)

## S3 method for class 'surveyncd_patterns'
autoplot(object, ...)
```

Arguments

`object`	A result object from one of the functions above.
`...`	Not used.

Value

A `ggplot` object.

Examples

```
d <- ncd_survey
d$htn <- recode_binary(d$htn_raw, yes = 1, no = 2)
d$dm  <- recode_binary(d$dm_raw,  yes = 1, no = 2)
des <- survey::svydesign(ids = ~psu, strata = ~stratum, weights = ~wt,
                         data = d, nest = TRUE)
ggplot2::autoplot(survey_prevalence(des, c("htn", "dm"), by = "residence"))
ggplot2::autoplot(survey_concentration_curve(des, dm, wealth_score))
```

child_anthro *Simulated child anthropometry (DHS-style)*

Description

Children under five living in the households of ncd_survey, with height-for-age, weight-for-age and weight-for-height z-scores stored in the DHS recode convention: multiplied by 100, with flag codes 9996 (height out of plausible range) and 9998 (missing), and a few biologically implausible values. **The data are entirely synthetic.**

Usage

```
child_anthro
```

Format

A data frame with 1,034 rows and 14 variables:

child_id Child identifier.

hh_id Household identifier (links to ncd_survey).

region, residence, stratum, psu, wt Design variables, inherited from the household.

sex "Female" or "Male".

age_months Age in months (0-59).

wealth_score, wealth_quintile Household wealth, as in ncd_survey.

hw70 Height-for-age z-score x 100 (DHS variable name).

hw71 Weight-for-age z-score x 100.

hw72 Weight-for-height z-score x 100.

Source

Simulated; see data-raw/simulate_survey.R in https://github.com/StatAid-Research-Lab/SurveyNCD.

Examples

```
table(who_anthro_score(child_anthro$hw70, indicator = "stunting"),
      useNA = "ifany")
```

`design_helpers` *Build a survey design from DHS, MICS or STEPS microdata*

Description

Thin wrappers around `survey::svydesign()` that use each programme's standard variable names, so the design (clusters, strata, weights) is specified correctly in one line.

Usage

```
dhs_design(
  data,
  psu = "v021",
  strata = "v023",
  weight = "v005",
  weight_scale = 1e+06,
  ...
)

mics_design(
  data,
  weight,
  psu = "HH1",
  strata = "stratum",
  weight_scale = 1,
  ...
)

steps_design(
  data,
  weight,
  psu = "psu",
  strata = "stratum",
  weight_scale = 1,
  ...
)
```

Arguments

`data`	A data frame of survey microdata.
`psu`, `strata`, `weight`	Names of the cluster, stratum and weight columns. Set `strata = NULL` for an unstratified design.
`weight_scale`	Number the weight is divided by (1e6 for raw DHS weights; 1 if the weight has already been rescaled).
`...`	Further arguments passed to `survey::svydesign()`.

Details

- **DHS**: the Guide to DHS Statistics identifies the primary sampling unit as `v021` (`hv021`/`mv021` in household and men's files), the strata as `v023` (`hv023`/`mv023`) and the weight as `v005` (`hv005`/`mv005`) divided by 1,000,000. Check the survey report (Appendix A) where strata were not consistently defined.
- **MICS**: clusters are `HH1` and strata `stratum`; the weight depends on the file (`hhweight`, `wmweight`, `mnweight`, `chweight`, `fsweight`), so it must be given.
- **STEPS**: clusters are usually `psu` and strata `stratum`; the weight depends on the step analysed (`wstep1`, `wstep2`, `wstep3`), so it must be given. A design built with the **stepssurvey** package can be passed to SurveyNCD functions directly instead.

All three set `nest = TRUE`, because cluster numbers are typically reused across strata, and drop rows with a missing or non-positive weight with a message.

Value

A `survey.design2` object.

References

ICF. Guide to DHS Statistics: Analyzing DHS data. https://dhsprogram.com/data/Guide-to-DHS-Statistics/Analyzing_DHS_Data.htm

Examples

```
# A toy individual-recode extract with DHS variable names
set.seed(1)
ir <- data.frame(
  v021 = rep(1:20, each = 10),
  v023 = rep(1:4, each = 50),
  v005 = round(runif(200, 5e5, 2e6)),
  htn  = rbinom(200, 1, 0.2)
)
des <- dhs_design(ir)
survey_prevalence(des, "htn")

# MICS and STEPS: same idea, weight column chosen for the file or step
names(ir) <- c("HH1", "stratum", "wmweight", "htn")
ir$wmweight <- ir$wmweight / 1e6
mics_design(ir, weight = "wmweight")
```

---

fci_items *The 18 conditions of the Functional Comorbidity Index*

Description

Returns the standard condition list for the Functional Comorbidity Index.

Usage

```
fci_items()
```

Value

A character vector of the 18 FCI condition descriptions.

References

Groll, D. L., To, T., Bombardier, C., & Wright, J. G. (2005). The development of a comorbidity index with physical function as the outcome. *Journal of Clinical Epidemiology*, 58(6), 595-602. doi:10.1016/j.jclinepi.2004.10.018

Examples

```
fci_items()
length(fci_items())  # 18
```

---

mm_bounds *Sensitivity of multimorbidity prevalence to "don't know" answers*

Description

Missing answers (for example "don't know") make multimorbidity prevalence uncertain beyond sampling error. This function reports three design-based estimates: the complete-case estimate (respondents with any missing answer excluded), a lower bound (every missing answer treated as "no") and an upper bound (every missing answer treated as "yes"). The bounds contain the prevalence under any assumption about the missing answers; the complete-case estimate is unbiased only if answers are missing completely at random.

Usage

```
mm_bounds(design, conditions, threshold = 2, by = NULL, level = 0.95)
```

Arguments

design	A survey design containing the 0/1 condition columns (with NA for missing answers).
conditions	Character vector of condition columns.
threshold	Number of conditions defining multimorbidity.
by	Optional character vector of grouping columns.
level	Confidence level.

Value

A data frame with `scenario` ("complete case", "lower bound", "upper bound"), any by columns, `n`, `estimate`, `se`, `lower` and `upper`.

Examples

```
d <- ncd_survey
for (v in c("htn", "dm", "arth"))
  d[[v]] <- suppressWarnings(recode_binary(d[[paste0(v, "_raw")]], yes = 1, no = 2))
des <- survey::svydesign(ids = ~psu, strata = ~stratum, weights = ~wt,
                         data = d, nest = TRUE)
mm_bounds(des, c("htn", "dm", "arth"))
```

---

mm_patterns *Most common combinations of conditions (multimorbidity patterns)*

Description

Design-based prevalence of every pair (or triad, ...) of conditions, with the ratio of observed to expected prevalence. The expected prevalence is the product of the single-condition prevalences, i.e. what would be seen if the conditions occurred independently; ratios above 1 indicate conditions that cluster together.

Usage

```
mm_patterns(design, conditions, order = 2, top = NULL, level = 0.95)
```

Arguments

`design`	A survey design containing the 0/1 condition columns.
`conditions`	Character vector of 0/1 condition columns.
`order`	Size of the combinations: 2 for pairs (default), 3 for triads.
`top`	Keep only the `top` most prevalent combinations (`NULL` keeps all).
`level`	Confidence level.

Details

A respondent counts towards a combination only if all of its conditions are answered; combinations are estimated by domain estimation on those respondents. Single-condition prevalences for the expected value are each estimated on respondents who answered that condition.

Value

A data frame of class `surveyncd_patterns` with columns `combination`, `n`, `prevalence`, `se`, `lower`, `upper`, `expected` and `oe_ratio`, sorted by prevalence.

Examples

```
d <- ncd_survey
for (v in c("htn", "dm", "arth", "angina"))
  d[[v]] <- suppressWarnings(recode_binary(d[[paste0(v, "_raw")]], yes = 1, no = 2))
des <- survey::svydesign(ids = ~psu, strata = ~stratum, weights = ~wt,
                         data = d, nest = TRUE)
mm_patterns(des, c("htn", "dm", "arth", "angina"))
```

`mm_prevalence` *Population-level multimorbidity estimates from complex survey data*

Description

Design-weighted prevalence of multimorbidity and mean number of conditions, accounting for stratification, clustering and sampling weights. Run `multimorbidity_index()` first to create `mm_category` and `mm_n_conditions`.

Usage

```
mm_prevalence(
  data,
  ids = NULL,
  strata = NULL,
  weights = NULL,
  by = NULL,
  nest = TRUE,
  level = 0.95
)
```

Arguments

`data`	A survey design whose data contain the output of `multimorbidity_index()`, or such a data frame.
`ids`, `strata`, `weights`	Column names of the clusters, strata and weights (only when `data` is a data frame). `ids = NULL` means no clustering and `strata = NULL` no stratification.
`by`	Optional character vector of one or more grouping columns.
`nest`	Passed to `survey::svydesign()` when `data` is a data frame.
`level`	Confidence level.

Details

`data` can be either a survey design object (recommended; for example from `dhs_design()` or `survey::svydesign()`) or a data frame plus the names of the design columns. Respondents with a missing multimorbidity score are excluded by domain estimation: the full design is kept for variance estimation and the number excluded is reported.

Value

A data frame of class `surveyncd_mm_prevalence` with any `by` columns, `n`, `prevalence`, `prevalence_se`, `prevalence_lower`, `prevalence_upper`, `mean_conditions` and `mean_conditions_se`.

Examples

```
set.seed(1)
n <- 40
df <- data.frame(
  psu    = rep(1:8, each = 5),
  region = rep(c("A", "B"), each = 20),
  sex    = rep(c("F", "M"), 20),
  wt     = round(runif(n, 0.8, 1.4), 2),
  htn    = rbinom(n, 1, 0.3),
  dm     = rbinom(n, 1, 0.2)
)
scored <- multimorbidity_index(df, conditions = c("htn", "dm"))
mm_prevalence(scored, ids = "psu", strata = "region", weights = "wt")

# Design-first interface, several grouping variables
des <- survey::svydesign(ids = ~psu, strata = ~region, weights = ~wt,
                         data = scored, nest = TRUE)
mm_prevalence(des, by = c("region", "sex"))
```

`mm_standardize` *Age-standardised prevalence from a survey design*

Description

Direct age standardisation of one or more 0/1 indicators using `survey::svystandardize()`, so that standard errors reflect the survey design. Crude (unstandardised) estimates are returned alongside.

Usage

```
mm_standardize(design, vars, age_group, standard, by = NULL, level = 0.95)
```

Arguments

`design`	A survey design.
`vars`	Character vector of 0/1 indicator columns (for example `"mm_multimorbid"` from `multimorbidity_index()`).
`age_group`	Name of a factor (or character) column of age groups.
`standard`	Standard population: a data frame with columns `age_group` and `weight` (as returned by `who_standard_population()`), or a named numeric vector. Every age group in the data must appear in it.
`by`	Optional character vector of grouping columns (standardisation is done within each group).
`level`	Confidence level.

Value

A data frame with `indicator`, any `by` columns, `n`, `crude`, `standardized`, `se`, `lower` and `upper` (Wald interval for the standardised estimate).

Examples

```
d <- ncd_survey
d$htn <- recode_binary(d$htn_raw, yes = 1, no = 2)
d$age_group <- cut(d$age, c(18, 30, 45, 60, 70), right = FALSE,
                   labels = c("18-29", "30-44", "45-59", "60-69"))
des <- survey::svydesign(ids = ~psu, strata = ~stratum, weights = ~wt,
                         data = d, nest = TRUE)
std <- who_standard_population(breaks = c(18, 30, 45, 60, 70))
mm_standardize(des, "htn", age_group = "age_group", standard = std, by = "sex")
```

`multimorbidity_index` *Calculate a self-reported multimorbidity index*

Description

Computes an individual-level multimorbidity score from a set of self-reported binary condition indicators (e.g. "Has a doctor ever told you that you have hypertension?"). This is designed for population health survey data (WHO STEPS, DHS, SAGE, etc.), where multimorbidity is captured through self-report rather than ICD-coded diagnoses – a data shape that existing comorbidity packages (which all assume ICD claims data) don't handle.

Usage

```
multimorbidity_index(
  data,
  conditions,
  weights = NULL,
  na_action = c("ignore", "na"),
  threshold = 2
)
```

Arguments

`data`	A data frame, one row per individual.
`conditions`	A character vector of column names in `data`. Each column must already be coded 0/1/NA (1 = condition present). Use `recode_binary()` first if your raw data uses Yes/No, 1/2, etc.
`weights`	Optional named numeric vector giving a weight for each condition (names must match `conditions`). If `NULL` (the default), every condition counts equally – i.e. a simple unweighted sum, which is how the published Functional Comorbidity Index (Groll et al. 2005) is scored. See `fci_items()`.

na_action	How to handle a row with at least one missing condition. "ignore" (default) sums whatever conditions are non-missing for that person. "na" returns NA for that person's index entirely, which is the safer choice once you move to population-level (survey-weighted) estimates, since silently treating missing as absent can bias prevalence downward.
threshold	Number of conditions at or above which a respondent is classified as multimorbid. The default, 2, is the most common definition; 3 is often used for "complex multimorbidity".

Value

data with four columns appended: mm_n_conditions (raw count of conditions present), mm_index (the possibly weighted score), mm_category (factor: "None", "Single condition" or, when threshold > 2, "1-k conditions", and "Multimorbid") and mm_multimorbid (1 if the count is at least threshold, else 0).

References

Groll, D. L., To, T., Bombardier, C., & Wright, J. G. (2005). The development of a comorbidity index with physical function as the outcome. *Journal of Clinical Epidemiology*, 58(6), 595-602. doi:10.1016/j.jclinepi.2004.10.018

Examples

```
df <- data.frame(
  id           = 1:5,
  hypertension = c(1, 0, 1, 1, 0),
  diabetes     = c(0, 0, 1, 1, 0),
  arthritis    = c(1, 0, 0, 1, NA)
)
multimorbidity_index(df, conditions = c("hypertension", "diabetes", "arthritis"))
```

---

ncd_survey *Simulated adult NCD survey (STEPS-style)*

Description

A simulated nationally representative-style survey of adults aged 18-69, drawn from a known synthetic population by stratified two-stage cluster sampling. It mimics the structure of WHO STEPS and DHS microdata (strata, primary sampling units, unequal sampling weights, raw response codes with "don't know" answers) so that every function in the package can be demonstrated. **The data are entirely synthetic**; regions are fictitious and no estimate describes any real country.

Usage

```
ncd_survey
```

Format

A data frame with 2,400 rows (one adult per household) and 18 variables:

hh_id Household identifier (links to `child_anthro`).

region Region (6 fictitious regions, "Region A" to "Region F").

residence "Urban" or "Rural".

stratum Sampling stratum (region x residence, 12 strata).

psu Primary sampling unit (enumeration area), 120 in total.

wt Sampling weight, normalised to mean 1 (as in DHS).

sex "Female" or "Male".

age Age in completed years (18-69).

wealth_score Continuous household wealth index (higher = richer).

wealth_quintile Weighted wealth quintile, 1 (poorest) to 5 (richest).

bmi Body mass index (kg/m^2).

htn_raw, dm_raw, arth_raw, asthma_raw, angina_raw, stroke_raw Raw answers to "Has a doctor or health worker ever told you that you have hypertension / diabetes / arthritis / asthma / angina or heart disease / stroke?", coded 1 = yes, 2 = no, 9 = don't know.

dep_raw Raw answer for depression, coded as text: "Yes", "No", "Don't know".

Details

The population has 6 regions of unequal size, each divided into urban and rural enumeration areas (12 strata). Ten enumeration areas were selected by simple random sampling in every stratum and 20 households in each, so small regions and urban areas are over-sampled and the weights vary about 15-fold. The code that generates the population and the sample is in `data-raw/` in the package source repository.

Source

Simulated; see `data-raw/simulate_survey.R` in https://github.com/StatAid-Research-Lab/SurveyNCD.

See Also

`child_anthro`

Examples

```
head(ncd_survey)
ncd_survey$htn <- recode_binary(ncd_survey$htn_raw, yes = 1, no = 2)
```

`ncd_table` *Publication-ready prevalence table*

Description

Builds a formatted table of design-based prevalence (with confidence intervals) for several indicators, overall and by group, ready to paste into a manuscript or pass to **knitr**, **flextable** or **gt**.

Usage

```
ncd_table(
  design,
  vars,
  by = NULL,
  labels = NULL,
  digits = 1,
  level = 0.95,
  n_row = TRUE
)
```

Arguments

`design`	A survey design (see `dhs_design()` or `survey::svydesign()`); **srvyr** designs are accepted.
`vars`	Character vector of 0/1 (or logical) indicator columns.
`by`	Optional single grouping column; one column of the table per group, plus an "Overall" column.
`labels`	Optional named character vector of row labels (`c(htn = "Hypertension")`).
`digits`	Decimal places for percentages.
`level`	Confidence level.
`n_row`	Add a first row with the unweighted number of respondents.

Value

A data frame of class `ncd_table` whose first column is `Indicator` and whose other columns hold "estimate (lower-upper)" strings in percent. The underlying numeric estimates are in attribute `"estimates"`.

Examples

```
d <- ncd_survey
d$htn <- recode_binary(d$htn_raw, yes = 1, no = 2)
d$dm  <- recode_binary(d$dm_raw,  yes = 1, no = 2)
des <- survey::svydesign(ids = ~psu, strata = ~stratum, weights = ~wt,
                         data = d, nest = TRUE)
ncd_table(des, c("htn", "dm"), by = "sex",
          labels = c(htn = "Hypertension", dm = "Diabetes"))
```

plot_shap_summary *Plot SHapley Additive exPlanations (SHAP) Summary*

Description

Experimental. Generates a SHAP summary plot (similar to the Python 'shap' package) where features are ranked on the y-axis by their overall importance (mean absolute SHAP value), SHAP values are shown on the x-axis, and each point (representing a respondent) is colored by its relative value for that feature (blue for low, red for high).

Usage

```
plot_shap_summary(
  shap_matrix,
  data,
  low_color = "#1e88e5",
  high_color = "#ff0052",
  title = "SHAP Summary Plot",
  subtitle = "Feature impact on model predictions (ranked by mean absolute SHAP)"
)
```

Arguments

shap_matrix	The matrix output from survey_shap().
data	The matrix or data frame of training features (e.g. the X matrix returned by survey_xgboost()) or the original data frame. If the row count does not match the SHAP matrix, rows with missing values will be automatically filtered out.
low_color	A character string for the color representing low feature values. Default is "#1e88e5" (standard SHAP blue).
high_color	A character string for the color representing high feature values. Default is "#ff0052" (standard SHAP red).
title	A character string for the plot title. Default is "SHAP Summary Plot".
subtitle	A character string for the plot subtitle. Default is "Feature impact on model predictions (ranked by mean absolute SHAP)".

Value

A ggplot2 object.

Examples

```
if (requireNamespace("xgboost", quietly = TRUE)) {
  set.seed(1)
  n <- 30
  df <- data.frame(
    outcome = rnorm(n, 100, 10),
```

```
    age     = round(runif(n, 18, 80)),
    bmi     = round(rnorm(n, 24, 4), 1),
    wt      = round(runif(n, 0.5, 2), 2)
  )
  design <- survey::svydesign(ids = ~1, weights = ~wt, data = df)
  model  <- survey_xgboost(design, outcome ~ age + bmi, nrounds = 5)
  shap   <- survey_shap(model)
  plot_shap_summary(shap, model$X)
}
```

`recode_binary` *Recode a messy survey column into clean 0/1*

Description

Converts alternative survey response values into clean standard numeric 0 or 1 scores. When `no` is not specified, every value that is neither in `yes` nor `NA` is treated as 0. When `no` *is* specified, values matching neither `yes` nor `no` are left as `NA` and a warning is issued.

Usage

```
recode_binary(x, yes, no = NULL)
```

Arguments

`x`	A vector coding a single condition.
`yes`	The raw value(s) that should become 1.
`no`	Optional raw value(s) that should become 0. If `NULL` (the default), every non-`yes`, non-`NA` value becomes 0.

Value

A numeric vector of 0, 1, or `NA`.

Examples

```
# STEPS-style coding: 1 = yes, 2 = no
recode_binary(c(1, 2, 1, NA), yes = 1, no = 2)

# When `no` is omitted, everything that isn't `yes` becomes 0
recode_binary(c(1, 2, 9, NA), yes = 1)

# With `no` specified, unrecognised codes (9) become NA with a warning
recode_binary(c(1, 2, 9, NA), yes = 1, no = 2)
```

survey_concentration_curve *Concentration curve from a survey design*

Description

Coordinates of the concentration curve: the cumulative share of the population ranked from poorest to richest (x) against the cumulative share of the health variable (y), both using sampling weights. A curve above the diagonal means the variable is concentrated among poorer people. Tied respondents are grouped into one point, consistent with `survey_concentration_index()`: the concentration index equals one minus twice the area under the curve.

Usage

```
survey_concentration_curve(design, outcome, wealth, by = NULL)
```

Arguments

`design`	A survey design object (**survey** or **srvyr**).
`outcome`	Unquoted name of the health variable.
`wealth`	Unquoted name of the ranking variable.
`by`	Optional character vector of grouping columns; ranks are computed within each group.

Value

A data frame of class `surveyncd_curve` with any `by` columns, `cum_population` and `cum_outcome`, starting at (0, 0).

Examples

```
set.seed(1)
df <- data.frame(wealth = rnorm(300), wt = runif(300, 0.5, 2))
df$sick <- rbinom(300, 1, plogis(-1 - 0.6 * df$wealth))
des <- survey::svydesign(ids = ~1, weights = ~wt, data = df)
cc <- survey_concentration_curve(des, sick, wealth)
head(cc)
```

`survey_concentration_index` *Survey-weighted concentration index*

Description

Computes the concentration index of a health variable across a socioeconomic ranking (for example a wealth score), with design-based standard errors, confidence intervals and p-values that account for stratification and clustering. The standard, generalized, Erreygers and Wagstaff indices are available.

Usage

```
survey_concentration_index(
  design,
  outcome,
  wealth,
  conf.level = 0.95,
  type = c("CI", "generalized", "erreygers", "wagstaff"),
  bounds = NULL,
  by = NULL
)
```

Arguments

`design`	A survey design object (**survey** or **srvyr**).
`outcome`	Unquoted name of the health variable.
`wealth`	Unquoted name of the ranking variable.
`conf.level`	Confidence level.
`type`	One of `"CI"` (default), `"generalized"`, `"erreygers"`, `"wagstaff"`.
`bounds`	Numeric vector `c(a, b)` of the outcome's bounds, needed for `"erreygers"` and `"wagstaff"`. Defaults to `c(0, 1)` when the outcome is binary.
`by`	Optional character vector of grouping columns; ranks are computed within each group.

Details

The standard concentration index is

$$C = \frac{2}{\mu}\mathrm{cov}_w(y_i, R_i),$$

where μ is the weighted mean of y and R_i the weighted fractional rank: the mid-point of each respondent's interval of the cumulative weight distribution (Lerman and Yitzhaki 1989). Respondents tied on the ranking variable share the mid-point of their tied group's interval, so results do not depend on the order of the data.

Each index is the slope β of the "convenient regression" of Kakwani, Wagstaff and van Doorslaer (1997), $2\sigma_R^2 h(y_i) = \alpha + \beta R_i + \varepsilon_i$, fitted with `survey::svyglm()` so that its standard error is design-based. With bounds $a \le y \le b$ (O'Donnell et al. 2016):

- `"CI"` (standard): $h(y) = y/\mu$.
- `"generalized"`: $h(y) = y$, giving μC.
- `"erreygers"`: $h(y) = 4y/(b-a)$, giving $4\mu C/(b-a)$ (Erreygers 2009).
- `"wagstaff"`: $h(y) = (b-a)y/((b-\mu)(\mu-a))$, giving $(b-a)\mu C/((b-\mu)(\mu-a))$; for binary outcomes $C/(1-\mu)$ (Wagstaff 2005).

For bounded outcomes such as binary indicators the Erreygers or Wagstaff index is usually preferred. The standard error treats the ranks and μ as fixed.

Value

A tibble of class `surveyncd_ci` with any by columns, `Concentration_Index`, `Standard_Error`, `Lower_CI`, `Upper_CI`, `p_value`, `Outcome_Mean`, `n` and `Type`.

References

Kakwani, N., Wagstaff, A., & van Doorslaer, E. (1997). Socioeconomic inequalities in health: Measurement, computation, and statistical inference. *Journal of Econometrics*, 77(1), 87-103. doi:10.1016/S03044076(96)018076

Erreygers, G. (2009). Correcting the concentration index. *Journal of Health Economics*, 28(2), 504-515. doi:10.1016/j.jhealeco.2008.02.003

Wagstaff, A. (2005). The bounds of the concentration index when the variable of interest is binary, with an application to immunization inequality. *Health Economics*, 14(4), 429-432. doi:10.1002/hec.953

O'Donnell, O., O'Neill, S., Van Ourti, T., & Walsh, B. (2016). conindex: Estimation of concentration indices. *The Stata Journal*, 16(1), 112-138. doi:10.1177/1536867X1601600112

Examples

```
set.seed(42)
n <- 50
df <- data.frame(
  wealth  = rnorm(n, 50, 15),
  outcome = pmax(0, 0.3 * rnorm(n, 50, 15) + rnorm(n, 0, 5)),
  sick    = rbinom(n, 1, 0.3),
  wt      = sample(1:5, n, replace = TRUE)
)
design <- survey::svydesign(ids = ~1, weights = ~wt, data = df)
survey_concentration_index(design, outcome = outcome, wealth = wealth)
survey_concentration_index(design, outcome = sick, wealth = wealth,
                           type = "erreygers")
```

`survey_map_indicator` *Map Survey Indicators Universally*

Description

Experimental. This function merges calculated survey indicators with any provided spatial shapefile to generate a publication-ready thematic map.

Usage

```
survey_map_indicator(
  survey_data,
  shapefile,
  join_by,
  fill_var,
  palette = c("magma", "viridis", "plasma", "inferno", "cividis"),
  legend_title = NULL,
  border_color = "white",
  border_width = 0.2
)
```

Arguments

`survey_data`	A data frame containing the aggregated survey indicators.
`shapefile`	An `sf` spatial object containing regional boundaries.
`join_by`	A character string of the column name present in both datasets to merge on.
`fill_var`	A character string of the variable in `survey_data` to map (e.g., "Concentration_Index").
`palette`	A character string specifying the viridis color palette option to use: `"magma"` (default), `"viridis"`, `"plasma"`, `"inferno"`, or `"cividis"`.
`legend_title`	A character string for the legend title. If `NULL` (default), the `fill_var` name is used.
`border_color`	A character string for the region border line color. Default is `"white"`.
`border_width`	A numeric value for the border line width. Default is `0.2`.

Value

A ggplot2 spatial map object.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  regional_map <- sf::st_read(
    system.file("shape/nc.shp", package = "sf"), quiet = TRUE
  )
  metrics <- data.frame(
```

```
    NAME = c("Ashe", "Alleghany", "Surry"),
    prevalence = c(0.12, 0.08, 0.15)
  )
  survey_map_indicator(metrics, regional_map, "NAME", "prevalence")
}
```

`survey_prevalence` *Design-based prevalence of one or more binary indicators*

Description

Estimates the population prevalence of each 0/1 indicator in `vars`, overall or within groups defined by one or more variables, using the full survey design (strata, clusters, weights). Respondents with a missing value for an indicator are excluded from that indicator only, by domain estimation, so the design is never broken by dropping rows.

Usage

```
survey_prevalence(design, vars, by = NULL, level = 0.95, method = "logit")
```

Arguments

`design`	A survey design (see `dhs_design()` or `survey::svydesign()`); **srvyr** designs are accepted.
`vars`	Character vector of 0/1 (or logical) indicator columns.
`by`	Optional character vector of grouping columns; estimates are produced for every observed combination.
`level`	Confidence level.
`method`	Confidence-interval method passed to `survey::svyciprop()`; `"logit"` (default) keeps intervals inside 0-1.

Value

A data frame of class `surveyncd_prevalence` with columns `indicator`, any `by` columns, `n` (unweighted respondents with a valid answer), `estimate`, `se`, `lower`, `upper` and `deff` (design effect relative to simple random sampling with replacement).

See Also

`ncd_table()` for a formatted table and `autoplot()` for a plot.

Examples

```
d <- ncd_survey
d$htn <- recode_binary(d$htn_raw, yes = 1, no = 2)
d$dm  <- recode_binary(d$dm_raw,  yes = 1, no = 2)
des <- survey::svydesign(ids = ~psu, strata = ~stratum, weights = ~wt,
                         data = d, nest = TRUE)
survey_prevalence(des, c("htn", "dm"))
survey_prevalence(des, "htn", by = c("sex", "residence"))
```

`survey_shap` *Extract SHAP Values from a Survey-Weighted XGBoost Model*

Description

Experimental. This function cracks open a trained survey-weighted XGBoost model and calculates the SHapley Additive exPlanations (SHAP values) for every respondent.

Usage

```
survey_shap(sxgb_model)
```

Arguments

`sxgb_model`	The list output from the `survey_xgboost()` function.

Value

A matrix of SHAP values detailing the marginal contribution of each feature to the final prediction for every observation.

Examples

```
if (requireNamespace("xgboost", quietly = TRUE)) {
  set.seed(1)
  n <- 30
  df <- data.frame(
    outcome = rnorm(n, 100, 10),
    age     = round(runif(n, 18, 80)),
    bmi     = round(rnorm(n, 24, 4), 1),
    wt      = round(runif(n, 0.5, 2), 2)
  )
  design <- survey::svydesign(ids = ~1, weights = ~wt, data = df)
  model  <- survey_xgboost(design, outcome ~ age + bmi, nrounds = 5)
  shap   <- survey_shap(model)
  head(shap)
}
```

`survey_slope_index` *Slope and relative index of inequality from a survey design*

Description

Regression-based measures of socioeconomic inequality on the weighted fractional rank R (0 = poorest, 1 = richest; tie-aware as in `survey_concentration_index()`):

- the slope index of inequality (SII), the slope of a linear regression of the outcome on R: the absolute difference in the outcome between the richest and the poorest end of the distribution;
- the relative index of inequality (RII), $\exp(\beta)$ from a log-link quasi-Poisson regression on R: the ratio of the outcome at the richest to the poorest end.

Both models are fitted with `survey::svyglm()`, giving design-based confidence intervals. Values of SII above 0 (RII above 1) mean higher levels among richer people.

Usage

```
survey_slope_index(design, outcome, wealth, conf.level = 0.95, by = NULL)
```

Arguments

`design`	A survey design object (**survey** or **srvyr**).
`outcome`	Unquoted name of the health variable.
`wealth`	Unquoted name of the ranking variable.
`conf.level`	Confidence level.
`by`	Optional character vector of grouping columns; ranks are computed within each group.

Value

A data frame with any by columns, `n`, `SII`, `SII_lower`, `SII_upper`, `RII`, `RII_lower`, `RII_upper`.

References

Mackenbach, J. P., & Kunst, A. E. (1997). Measuring the magnitude of socio-economic inequalities in health: An overview of available measures illustrated with two examples from Europe. *Social Science & Medicine*, 44(6), 757-771. doi:10.1016/S02779536(96)000731

Examples

```
set.seed(1)
df <- data.frame(wealth = rnorm(300), wt = runif(300, 0.5, 2))
df$sick <- rbinom(300, 1, plogis(-1 - 0.6 * df$wealth))
des <- survey::svydesign(ids = ~1, weights = ~wt, data = df)
survey_slope_index(des, sick, wealth)
```

survey_xgboost *Train an XGBoost Model with Complex Survey Weights*

Description

Experimental. This function trains an Extreme Gradient Boosting (XGBoost) model while strictly enforcing sampling weights from a complex survey design object.

Usage

```
survey_xgboost(design, formula, params = list(), nrounds = 100)
```

Arguments

design	A survey.design object containing the data and weights.
formula	A formula specifying the response and predictor variables.
params	A list of XGBoost parameters (e.g., objective, eta, max_depth).
nrounds	The number of boosting iterations.

Value

A list containing the trained xgb.Booster model, the xgb.DMatrix, the feature names, the observation count, and the cleaned training feature matrix X.

Examples

```
if (requireNamespace("xgboost", quietly = TRUE)) {
  set.seed(1)
  n <- 30
  df <- data.frame(
    outcome = rnorm(n, 100, 10),
    age     = round(runif(n, 18, 80)),
    bmi     = round(rnorm(n, 24, 4), 1),
    wt      = round(runif(n, 0.5, 2), 2)
  )
  design <- survey::svydesign(ids = ~1, weights = ~wt, data = df)
  model  <- survey_xgboost(design, outcome ~ age + bmi, nrounds = 5)
  model$features
}
```

`who_anthro_score` *Calculate WHO Anthropometric Categories from DHS/MICS Data*

Description

Cleans raw DHS/MICS z-score variables (e.g., hw70, hw71, hw72), adjusts the decimal scaling if needed, handles DHS-specific missing flags, applies WHO biologically implausible flagging rules, and categorizes them into WHO severity tiers.

Usage

```
who_anthro_score(
  x,
  indicator = c("stunting", "wasting", "underweight"),
  scaled_by_100 = TRUE,
  remove_implausible = TRUE
)
```

Arguments

`x`	A numeric vector of raw DHS/MICS z-scores.
`indicator`	The type of indicator for labeling and flagging: `"stunting"` (height-for-age, HAZ), `"wasting"` (weight-for-height, WHZ), or `"underweight"` (weight-for-age, WAZ).
`scaled_by_100`	Logical. If `TRUE` (default), divides input values by 100 (standard for DHS raw recode files like `hw70`). If `FALSE`, assumes values are already on the standard z-score scale (like MICS or cleaned DHS data).
`remove_implausible`	Logical. If `TRUE` (default), applies WHO child growth standard flagging rules to set biologically implausible values to `NA`. WHO flags are: • Stunting (HAZ): < -6.0 or > 6.0 • Wasting (WHZ): < -5.0 or > 5.0 • Underweight (WAZ): < -6.0 or > 5.0

Value

A factor vector of WHO categories (Severe, Moderate, Normal) with the indicator label appended.

References

World Health Organization (2006). *WHO Child Growth Standards: Length/height-for-age, weight-for-age, weight-for-length, weight-for-height and body mass index-for-age: Methods and development*. Geneva: WHO.

Examples

```
# DHS-style raw z-scores (scaled by 100)
raw <- c(-310, -254, 50, 9999)
who_anthro_score(raw, indicator = "stunting")

# Already on standard z-score scale
who_anthro_score(c(-3.1, -2.5, 0.5), indicator = "wasting",
                 scaled_by_100 = FALSE)
```

---

who_standard_population

WHO World Standard Population

Description

The WHO World Standard Population (world average population 2000-2025; Ahmad et al. 2001), optionally re-aggregated to custom age groups such as the 18-29, 30-44, 45-59 and 60-69 groups used in STEPS reports.

Usage

```
who_standard_population(breaks = NULL, labels = NULL)
```

Arguments

`breaks`	Optional increasing numeric vector of age-group boundaries. Intervals are `[breaks[i], breaks[i + 1])` the last boundary is the (exclusive) upper limit. `NULL` returns the 21 published groups.
`labels`	Optional labels for the intervals.

Details

The standard is published in 5-year groups. For `breaks` that are not multiples of 5 (e.g. 18), the share of a 5-year group falling inside an interval is assumed proportional to the years it covers (uniform age distribution within the group). The 100+ group is treated as 100-104.

Value

A data frame with `age_group`, `lower`, `upper` (exclusive) and `weight`, where `weight` sums to 1.

References

Ahmad, O. B., Boschi-Pinto, C., Lopez, A. D., Murray, C. J. L., Lozano, R., & Inoue, M. (2001). *Age standardization of rates: A new WHO standard.* GPE Discussion Paper Series No. 31. Geneva: World Health Organization.

Examples

```
who_standard_population()
who_standard_population(breaks = c(18, 30, 45, 60, 70))
```

Index

∗ **datasets**
child_anthro, 4
ncd_survey, 12

autoplot.surveyncd, 3
autoplot.surveyncd_ci *(*autoplot.surveyncd*)*, 3
autoplot.surveyncd_curve *(*autoplot.surveyncd*)*, 3
autoplot.surveyncd_mm_prevalence *(*autoplot.surveyncd*)*, 3
autoplot.surveyncd_patterns *(*autoplot.surveyncd*)*, 3
autoplot.surveyncd_prevalence *(*autoplot.surveyncd*)*, 3

child_anthro, 4, *13*

design_helpers, 5
dhs_design *(*design_helpers*)*, 5
dhs_design(), *9*, *14*, *21*

fci_items, 6

mics_design *(*design_helpers*)*, 5
mm_bounds, 7
mm_patterns, 8
mm_patterns(), *3*
mm_prevalence, 9
mm_prevalence(), *3*
mm_standardize, 10
multimorbidity_index, 11
multimorbidity_index(), *9*, *10*

ncd_survey, *4*, 12
ncd_table, 14
ncd_table(), *21*

plot_shap_summary, 15

recode_binary, 16

steps_design *(*design_helpers*)*, 5
survey::svyciprop(), *21*
survey::svydesign(), *5*, *9*, *14*, *21*
survey::svyglm(), *19*, *23*
survey::svystandardize(), *10*
survey_concentration_curve, 17
survey_concentration_curve(), *3*
survey_concentration_index, 18
survey_concentration_index(), *3*, *17*, *23*
survey_map_indicator, 20
survey_prevalence, 21
survey_prevalence(), *3*
survey_shap, 22
survey_slope_index, 23
survey_xgboost, 24

who_anthro_score, 25
who_standard_population, 26
who_standard_population(), *10*