

Package: eq5dsuite (via r-universe)

October 23, 2026

Type Package

Title Handling and Analysing 'EQ-5D' Data

Version 2.1.0

Maintainer Kim Rand <krand@mathsinhealth.com>

Description The EQ-5D is a widely-used standardised instrument for measuring Health Related Quality Of Life (HRQOL), developed by the EuroQol group <<https://euroqol.org/>>. It assesses five dimensions; mobility, self-care, usual activities, pain/discomfort, and anxiety/depression, using either a three-level (EQ-5D-3L) or five-level (EQ-5D-5L) scale. Scores from these dimensions are commonly converted into a single utility index using country-specific value sets, which are critical in clinical and economic evaluations of healthcare and in population health surveys. The eq5dsuite package enables users to calculate utility index values for the EQ-5D instruments, including crosswalk utilities using the original crosswalk developed by van Hout et al. (2012) <[doi:10.1016/j.jval.2012.02.008](https://doi.org/10.1016/j.jval.2012.02.008)> (mapping EQ-5D-5L responses to EQ-5D-3L index values), or the recently developed reverse crosswalk by van Hout et al. (2021) <[doi:10.1016/j.jval.2021.03.009](https://doi.org/10.1016/j.jval.2021.03.009)> (mapping EQ-5D-3L responses to EQ-5D-5L index values). Users are allowed to add and/or remove user-defined value sets. Additionally, the package provides tools to analyse EQ-5D data according to the recommended guidelines outlined in ``Methods for Analyzing and Reporting EQ-5D data" by Devlin et al. (2020) <[doi:10.1007/978-3-030-47622-9](https://doi.org/10.1007/978-3-030-47622-9)>.

URL <https://github.com/MathsInHealth/eq5dsuite-r>,
<https://github.com/MathsInHealth/eq5dsuite>

BugReports <https://github.com/MathsInHealth/eq5dsuite-r/issues>

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)
Imports curl, ggplot2, moments, RColorBrewer, rlang, scales, rappdirs
Suggests shiny, DT, bslib, readxl, spelling, testthat (>= 3.0.0),
 withr, knitr, rmarkdown, writexl, zip
VignetteBuilder knitr
Config/testthat/edition 3
Language en-GB
Config/roxygen2/version 8.1.0
Config/pak/sysreqs libssl-dev
Repository <https://mathsinhealth.r-universe.dev>
Date/Publication 2026-10-21 18:04:01 UTC
RemoteUrl <https://github.com/mathsinhealth/eq5dsuite-r>
RemoteRef HEAD
RemoteSha aebe99192260ba2636fb924254e4e3d7a7024bba

Contents

eq5dsuite-package	3
.EQxwrprob	7
.freqtab	8
.pstate3t5	9
.pstate5t3	9
eq5d	10
eq5d_profile_better_dimensions_by_group_plot	11
eq5d_profile_change_summary	12
eq5d_profile_density_curve	14
eq5d_profile_dimension_change_table	15
eq5d_profile_health_profile_grid	17
eq5d_profile_level_summary	18
eq5d_profile_level_summary_by_group	19
eq5d_profile_lfs_distribution	20
eq5d_profile_lfs_mean_utility	21
eq5d_profile_lfs_utility_plot	22
eq5d_profile_lfs_utility_summary	23
eq5d_profile_lss_utility_plot	24
eq5d_profile_lss_utility_summary	25
eq5d_profile_mixed_dimensions_by_group_plot	26
eq5d_profile_pchc_by_group_plot	27
eq5d_profile_pchc_table	29
eq5d_profile_pchc_with_no_problems_table	30
eq5d_profile_shannon	31
eq5d_profile_top_states	33
eq5d_profile_worse_dimensions_by_group_plot	34
eq5d_utility_by_group_plot	35

eq5d_utility_change_by_group_plot	36
eq5d_utility_distribution_plot	37
eq5d_utility_norms_comparison	38
eq5d_utility_over_time_plot	39
eq5d_utility_summary	40
eq5d_utility_summary_by_group	41
eq5d_utility_vas_scatter_plot	42
eq5d_vas_distribution_table	43
eq5d_vas_grouped_distribution_plot	44
eq5d_vas_histogram	44
eq5d_vas_summary	45
eq5d3l	46
eq5d5l	47
eq5dy3l	49
eqvs_add	51
eqvs_display	53
eqvs_drop	54
eqvs_load	55
eqxw	56
eqxw_UK	57
eqxwr	59
eqxwr_UK	61
example_data	63
make_all_EQ_indexes	64
make_all_EQ_states	64
make_dummies	65
run_app	66
toEQ5Ddims	69
toEQ5Dindex	69
update_value_sets	70

Index	73
--------------	-----------

Description

The **eq5dsuite** package provides a comprehensive set of functions for calculating EQ-5D preference-based values and analysing EQ-5D data following the recommendations of Devlin et al. (2020).

The package supports three EQ-5D instruments:

- EQ-5D-3L
- EQ-5D-5L
- EQ-5D-Y-3L

Value calculation

Functions for calculating EQ-5D preference-based values:

- [eq5d](#) — Generic calculator for all instruments
- [eq5d3l](#) — EQ-5D-3L value calculation
- [eq5d5l](#) — EQ-5D-5L value calculation
- [eq5dy3l](#) — EQ-5D-Y-3L value calculation
- [eqxw](#) — Original crosswalk method (van Hout et al., 2012)
- [eqxwr](#) — Reverse crosswalk method (van Hout and Shaw, 2021)
- [eqxwr_UK](#) — UK mapping from EQ-5D-3L responses to values on the UK EQ-5D-5L value set (NICE Decision Support Unit; Hernández Alava et al., 2023). This is the mapping NICE’s interim methods statement of 27 August 2026 specifies when only EQ-5D-3L data are available
- [eqxw_UK](#) — UK mapping in the opposite direction, from EQ-5D-5L responses to values on the UK EQ-5D-3L value set (Hernández Alava et al., 2023). Under NICE’s interim methods statement of 27 August 2026, EQ-5D-5L data collected in a study should instead be valued directly with [eq5d5l\(country = "GB"\)](#); [eqxw_UK](#) remains for topics started before that date, for which NICE says 5L data should still be mapped to 3L utility values, and for reproducing earlier analyses

Value set management

Functions for managing country-specific value sets:

- [eqvs_display](#) — List all available value sets
- [eqvs_add](#) — Add a custom value set
- [eqvs_drop](#) — Remove a custom value set
- [eqvs_load](#) — Load a saved custom value set into the current session
- [update_value_sets](#) — Check for and install new value sets from the online repository

Profile analysis

Functions for analysing EQ-5D health state profiles, including cross-sectional summaries, longitudinal change analyses, and severity metrics:

- [eq5d_profile_level_summary](#) — Cross-sectional frequency summary by dimension
- [eq5d_profile_level_summary_by_group](#) — Stratified frequency summary by subgroup
- [eq5d_profile_top_states](#) — Most frequently observed health state profiles
- [eq5d_profile_change_summary](#) — Dimension-level frequencies at two time points
- [eq5d_profile_dimension_change_table](#) — Changes in levels per dimension as percentages of total and of type of change
- [eq5d_profile_pchc_table](#) — Paretian Classification of Health Change table
- [eq5d_profile_pchc_with_no_problems_table](#) — PCHC table with no-problems category
- [eq5d_profile_pchc_by_group_plot](#) — PCHC bar chart by subgroup

- [eq5d_profile_better_dimensions_by_group_plot](#) — Dimensions improved among Better patients
- [eq5d_profile_worse_dimensions_by_group_plot](#) — Dimensions worsened among Worse patients
- [eq5d_profile_mixed_dimensions_by_group_plot](#) — Dimensions changed among Mixed patients
- [eq5d_profile_health_profile_grid](#) — Health Profile Grid of individual transitions
- [eq5d_profile_lss_utility_summary](#) — Descriptive statistics by Level Sum Score
- [eq5d_profile_lss_utility_plot](#) — EQ-5D values plotted against LSS
- [eq5d_profile_lfs_distribution](#) — Level Frequency Score distribution table
- [eq5d_profile_lfs_mean_utility](#) — Mean EQ-5D values by LFS category
- [eq5d_profile_lfs_utility_summary](#) — Descriptive statistics by Level Frequency Score
- [eq5d_profile_lfs_utility_plot](#) — EQ-5D values plotted against LFS
- [eq5d_profile_density_curve](#) — Health State Density Curve
- [eq5d_profile_shannon](#) — Shannon’s informativity indices H' and J' , by dimension and for the health state

EQ-5D value analysis

Functions for analysing EQ-5D preference-based values across time points, subgroups, and population norms:

- [eq5d_utility_summary](#) — Descriptive statistics at each time point
- [eq5d_utility_summary_by_group](#) — Descriptive statistics stratified by subgroup
- [eq5d_utility_norms_comparison](#) — Comparison with population norms by age and sex
- [eq5d_utility_over_time_plot](#) — Mean values with confidence intervals over time
- [eq5d_utility_by_group_plot](#) — Mean values with confidence intervals by subgroup
- [eq5d_utility_change_by_group_plot](#) — Longitudinal change by subgroup
- [eq5d_utility_distribution_plot](#) — Histogram of EQ-5D value distribution
- [eq5d_utility_vas_scatter_plot](#) — Scatter plot of EQ-5D values against EQ VAS

EQ VAS analysis

Functions for analysing EQ VAS self-rated health scores:

- [eq5d_vas_summary](#) — Descriptive statistics at each time point
- [eq5d_vas_distribution_table](#) — Frequency distribution in pre-defined ranges
- [eq5d_vas_histogram](#) — Histogram of EQ VAS scores
- [eq5d_vas_grouped_distribution_plot](#) — Bar chart of grouped EQ VAS frequencies

Helper functions

Utility functions for preparing EQ-5D data:

- `toEQ5Dindex` — Combine dimension columns into a 5-digit profile code
- `toEQ5Ddims` — Split a 5-digit profile code into separate dimension columns
- `make_all_EQ_indexes` — Generate all valid EQ-5D state codes as a vector
- `make_all_EQ_states` — Generate all valid EQ-5D states as a data frame
- `make_dummies` — Create dummy variables from EQ-5D dimension columns

Data

- `example_data` — A sample of EQ-5D-3L data from the NHS Patient Reported Outcome Measures (PROMs) programme, used throughout the examples and vignettes. Missing values are recorded as 9 in the five dimensions and 999 in the EQ VAS

Interactive application

A Shiny application providing point-and-click access to the same analytical workflow without requiring R coding:

- `run_app` — Launch the Shiny application

Cross-platform suite

eq5dsuite is part of a cross-platform suite also available in Stata and Excel:

- Full suite: <https://github.com/MathsInHealth/eq5dsuite>
- R package: <https://github.com/MathsInHealth/eq5dsuite-r>
- Value sets repository: <https://github.com/MathsInHealth/eq5dsuite-value-sets>

Author(s)

Maintainer: Kim Rand <krand@mathsinhealth.com> ([ORCID](#))

Authors:

- Kim Rand <krand@mathsinhealth.com> ([ORCID](#))
- Oliver Rivero-Arias ([ORCID](#))
- Iryna Schlackow ([ORCID](#))
- Anabel Estévez-Carrillo ([ORCID](#))

References

Devlin N, Parkin D, Janssen B (2020). *Methods for Analysing and Reporting EQ-5D Data*. Springer, Cham. [doi:10.1007/9783030476229](https://doi.org/10.1007/9783030476229)

van Hout B, Janssen MF, Feng YS, et al. (2012). Interim scoring for the EQ-5D-5L: mapping the EQ-5D-5L to EQ-5D-3L value sets. *Value in Health*, 15(5), 708–715. [doi:10.1016/j.jval.2012.02.008](https://doi.org/10.1016/j.jval.2012.02.008)

van Hout B, Shaw JW (2021). Mapping EQ-5D-3L to EQ-5D-5L. *Value in Health*, 24(9), 1285–1293. doi:10.1016/j.jval.2021.03.009

Hernández Alava M, Pudney S, Wailoo A (2023). Estimating the relationship between EQ-5D-5L and EQ-5D-3L: results from a UK population study. *Pharmacoeconomics*, 41(2), 199–207. doi:10.1007/s40273022012187

Rowen D, Mukuria C, Bray N, et al. (2026). A UK value set for the EQ-5D-5L. *Value in Health*. doi:10.1016/j.jval.2026.03.008

National Institute for Health and Care Excellence (2026). Interim methods statement: implementing the EQ-5D-5L value set (PMG51). Published 27 August 2026. <https://www.nice.org.uk/process/pmg51>

See Also

Useful links:

- <https://github.com/MathsInHealth/eq5dsuite-r>
- <https://github.com/MathsInHealth/eq5dsuite>
- Report bugs at <https://github.com/MathsInHealth/eq5dsuite-r/issues>

.EQxwrprob

.EQxwrprob

Description

Takes a matrix of parameters for reverse crosswalk model, returns 243 x 25 matrix of state/level transition probabilities.

Usage

```
.EQxwrprob(par = NULL)
```

Arguments

par	Matrix of model parameters
-----	----------------------------

Value

An 243 * 25 matrix with probabilities for state level transitions.

`.freqtab`*Helper function for frequency of levels by dimensions tables*

Description

Helper function for frequency of levels by dimensions tables

Usage

```
.freqtab(  
  df,  
  names_eq5d = NULL,  
  name_fu = NULL,  
  levels_fu = NULL,  
  eq5d_version = NULL,  
  add_summary_problems_change = TRUE  
)
```

Arguments

<code>df</code>	Data frame with the EQ-5D and follow-up columns
<code>names_eq5d</code>	Character vector of column names for the EQ-5D dimensions
<code>name_fu</code>	Character string for the follow-up column. If NULL, no grouping is used, and the table reports for the total population.
<code>levels_fu</code>	Character vector containing the order of the values in the follow-up column. If NULL (default value), the levels will be ordered in the order of appearance in <code>df</code> .
<code>eq5d_version</code>	Version of the EQ-5D instrument: "3L", "5L" or "Y3L". Matching is case-insensitive.
<code>add_summary_problems_change</code>	If set to false, the resulting dataframe does not include a row on problems change.

Value

Summary data frame.

.pstate3t5

.pstate3t5

Description

Takes a N x 25 matrix with probabilities per level/dimension, and creates an N * 3125 matrix with probabilities per state

Usage

```
.pstate3t5(PPP)
```

Arguments

PPP N x 25 matrix with probabilities per level/dimension created by EQrxwprobs

Value

An N * 3125 matrix with probabilities per state

.pstate5t3

.pstate5t3

Description

Takes a 15 x 5 matrix with probabilities per level/dimension, and creates an 3125x243 matrix with probabilities per state

Usage

```
.pstate5t3(probs = .EQxwprob)
```

Arguments

probs 15 x 5 matrix with probabilities per level/dimension, typically saved in .EQxwprob

Value

An 3125x243 matrix with probabilities per state

eq5d

*eq5d***Description**

Get EQ-5D values for the -3L, -5L, crosswalk (-3L value set applied to -5L health states), reverse crosswalk (-5L value set applied to -3L health states), and -Y-3L

Usage

```
eq5d(
  x,
  country = NULL,
  version = "5L",
  dim.names = c("mo", "sc", "ua", "pd", "ad")
)
```

Arguments

x	A vector of 5-digit EQ-5D-3L state indexes or a matrix/data.frame with columns corresponding to EQ-5D state dimensions
country	A country code or value set code identifying the value set to use, as listed by eqvs_display(). Matching is case-insensitive. For countries with more than one value set (for example Germany, with "DE_TTO" and "DE_VAS"), the value set code must be given; supplying the country code alone raises an error listing the available codes.
version	String indicating which version to use. Options are '5L' (default), '3L', 'xw', 'xwr', and 'Y3L'. Matching is case-insensitive.
dim.names	A vector of dimension names to identify dimension columns.

Value

A numeric vector of values, or a data.frame with one column for each value set requested. The result is an unnamed numeric vector, one element per element or row of x and in the same order.

Missing and invalid responses

Each health state is checked before it is scored, and a state that cannot be scored gives NA for that element or row; the others are scored as usual.

- A missing response (NA in any dimension, or an NA state index) gives NA without a warning.
- An invalid response gives NA with a warning, which gives the number of such states and examples. Invalid means a level the instrument does not have (for example 4 in the EQ-5D-3L, or a missing-value code such as 9), a level or index that is not a whole, finite number, or an index that is not five digits long. Each problem is reported once, however many value sets are requested.

- State indexes and levels given as text or as factors are read through their labels, so "21111" and factor(2) are scored as 21111 and 2.

Recode missing-value codes to NA first if the warning is not wanted.

Examples

```
# US -3L value set
eq5d(c(11111, 12321, 32123, 33333), 'US', '3L')
# Danish and US -5L value sets applied to -3L descriptives, i.e. reverse crosswalk
eq5d(make_all_EQ_states('3L'), c('DK', 'US'), 'XWR')
# US -5L value set
eq5d(c(11111, 12321, 32153, 55555), 'US', '5L')
```

```
eq5d_profile_better_dimensions_by_group_plot
```

eq5d_profile_better_dimensions_by_group_plot: Percentage of Respondents Who Improved in Each EQ-5D Dimension, by Group This function calculates how many respondents improved in each dimension between two time points and summarizes the results for each group. It then produces a dimension-focused chart illustrating improvement percentages by dimension.

Description

eq5d_profile_better_dimensions_by_group_plot: Percentage of Respondents Who Improved in Each EQ-5D Dimension, by Group This function calculates how many respondents improved in each dimension between two time points and summarizes the results for each group. It then produces a dimension-focused chart illustrating improvement percentages by dimension.

Usage

```
eq5d_profile_better_dimensions_by_group_plot(
  df,
  name_id,
  name_groupvar = NULL,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL
)
```

Arguments

df	Data frame containing EQ-5D columns, a grouping variable, an ID column, and a follow-up column
name_id	Character string for the patient ID column
name_groupvar	Character string for the grouping column (e.g. procedure). If NULL (default), the analysis is performed on the full population.

names_eq5d	Character vector of EQ-5D dimension names
name_fu	Character string for the follow-up column
levels_fu	Character vector of length 2, specifying the order of the follow-up levels (e.g. c("Pre-op", "Post-op"))

Value

A list containing:

plot_data	A data frame of improvements by group and dimension
p	A ggplot2 object produced by <code>‘.pchc_plot_by_dim()’</code>

Pairing records

Each respondent's records are compared in timepoint order, using only the timepoints in `levels_fu`. The function stops, naming the respondents, if a respondent has more than one record at the same timepoint: it cannot tell which to compare, and does not choose. Where the analysis is split by `name_groupvar`, it also stops if a respondent is not in the same group at every timepoint (a missing group at one timepoint counts as a change), because the change is attributed to the respondent's group.

Examples

```
result <- eq5d_profile_better_dimensions_by_group_plot(
  df = example_data,
  name_id = "id",
  name_groupvar = "procedure",
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op", "Post-op")
)
result$p
result$plot_data
```

eq5d_profile_change_summary

*eq5d_profile_change_summary: Frequency of levels by dimensions,
by follow-up*

Description

eq5d_profile_change_summary: Frequency of levels by dimensions, by follow-up

Usage

```
eq5d_profile_change_summary(
  df,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL,
  eq5d_version = NULL
)
```

Arguments

<code>df</code>	Data frame with the EQ-5D and follow-up columns
<code>names_eq5d</code>	Character vector of column names for the EQ-5D dimensions
<code>name_fu</code>	Character string for the follow-up column. If NULL, the function will check if there is a column named "follow-up" or "fu", in which case the first of those will be used.
<code>levels_fu</code>	Character vector containing the order of the values in the follow-up column. If NULL (default value), the levels will be ordered in the order of appearance in <code>df</code> .
<code>eq5d_version</code>	Version of the EQ-5D instrument

Details

Counts are made against the number of non-missing responses on each dimension at each follow-up. Where that number is positive, every level of the instrument has a row and every count is a number: a level, or a problem, that nobody reported is 0. Where it is zero – nobody recorded at that follow-up, or everyone missing on that dimension – the counts are NA, because the data say nothing about them.

The change in the number reporting problems is between consecutive levels of `levels_fu`, and is NA where either count is. The relative change is NA where the earlier count is 0. These are counts at each follow-up, not changes within respondents; for those, see [eq5d_profile_pchc_table](#) and [eq5d_profile_dimension_change_table](#).

Value

Summary data frame.

Examples

```
eq5d_profile_change_summary(
  df = example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op", "Post-op"),
  eq5d_version = "3L"
)
```

eq5d_profile_density_curve

eq5d_profile_density_curve: Generate a Health State Density Curve (HSDC) for EQ-5D Data

Description

Calculates and plots the Health State Density Curve (HSDC) and the Health State Density Index (HSDI) of Zamora et al. (2018), which describe how evenly the observations are spread over the health state profiles that occur in the data. Profiles are formed from the five dimensions; a row whose profile is not valid for the instrument is excluded with a warning.

Usage

```
eq5d_profile_density_curve(df, names_eq5d, eq5d_version)
```

Arguments

df	Data frame with the EQ-5D columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
eq5d_version	Version of the EQ-5D instrument

Details

The curve. The S distinct profiles observed are ranked from the most to the least frequent. For $i = 1, \dots, S$, x_i is the cumulative proportion of observations in the first i profiles and $y_i = i/S$ the cumulative proportion of profiles. The HSDC plots y against x , from $(0, 0)$ to $(1, 1)$. The diagonal is perfect evenness, every profile equally frequent; the further the curve lies below it, the more the observations are concentrated in a few profiles.

The index. Following Zamora et al. (2018, equation 1),

$$HSDI = \sum_{i=1}^S (x_i - x_{i-1})(y_i + y_{i-1}),$$

with $x_0 = y_0 = 0$: twice the area under the curve. It is analogous to the Gini coefficient, but runs the other way: the HSDI is $1 - G$, where G is twice the area between the curve and the diagonal.

Direction. Higher is more even. The HSDI is 1 when every observed profile is equally frequent, including the case of a single profile, and falls as the observations concentrate in fewer of the observed profiles. Profiles of equal frequency may be ranked in either order without changing it.

Attainable bounds. With N observations over S observed profiles, the HSDI lies in

$$1/S + (S - 1)/N \leq HSDI \leq 1.$$

The lower bound is reached when one profile holds $N - S + 1$ observations and each of the others one. The HSDI is therefore never below $1/S$, and never 0: the 0 of "total inequality" in Zamora et al. is a limit, approached only as S and N/S both grow. With two profiles it cannot fall below 0.5.

What it is relative to. Only the profiles that occur in these data count: y_i is i/S for the S profiles observed, not for the 243 (EQ-5D-3L, EQ-5D-Y-3L) or 3,125 (EQ-5D-5L) the instrument allows, and a profile nobody reported does not enter. The index describes how evenly observations are spread over the profiles that appear. Because its attainable range depends on S and N , compare it between samples with care. Repeating a whole distribution over more profiles leaves it exactly unchanged (Zamora et al.'s independence of S), but samples whose distributions differ in shape are not on a common scale.

Worked values. Two profiles observed equally often: 1. Frequencies 9 and 1: $0.9 \times 0.5 + 0.1 \times 1.5 = 0.6$. Frequencies 999 and 1: 0.501, the paper's "0.5" for this case. Frequencies 2, 1 and 1: $(0.5 \times 1 + 0.25 \times 3 + 0.25 \times 5)/3 = 5/6$, the lower bound for $N = 4$, $S = 3$.

Value

A list containing:

plot_data	A data frame with the cumulative distribution of profiles
hsdi	The Health State Density Index, unrounded. The plot subtitle shows it rounded to three decimal places.
p	A ggplot2 object showing the Health State Density Curve

References

Zamora B, Parkin D, Feng Y, Bateman A, Herdman M, Devlin N (2018). New methods for analysing the distribution of EQ-5D observations. OHE Research Paper 18/03. London: Office of Health Economics. <https://www.ohe.org/publications/new-methods-analysing-distribution-eq-5d-observati>

Devlin N, Parkin D, Janssen B (2020). Methods for Analysing and Reporting EQ-5D Data. Cham: Springer. doi:10.1007/9783030476229

Examples

```
figure <- eq5d_profile_density_curve(
  df = example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L"
)
figure$plot_data
figure$hsdi
figure$p
```

eq5d_profile_dimension_change_table

eq5d_profile_dimension_change_table: Changes in levels in each dimension, percentages of total and of type of change

Description

eq5d_profile_dimension_change_table: Changes in levels in each dimension, percentages of total and of type of change

Usage

```
eq5d_profile_dimension_change_table(
  df,
  name_id,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL
)
```

Arguments

<code>df</code>	Data frame with the EQ-5D, id and follow-up columns
<code>name_id</code>	Character string for the patient id column
<code>names_eq5d</code>	Character vector of column names for the EQ-5D dimensions
<code>name_fu</code>	Character string for the follow-up column
<code>levels_fu</code>	Character vector containing the order of the values in the follow-up column. If NULL (default value), the levels will be ordered in the order of appearance in <code>df</code> .

Value

Summary data frame

Pairing records

Each respondent's records are compared in timepoint order, using only the timepoints in `levels_fu`. The function stops, naming the respondents, if a respondent has more than one record at the same timepoint: it cannot tell which to compare, and does not choose. Where the analysis is split by `name_groupvar`, it also stops if a respondent is not in the same group at every timepoint (a missing group at one timepoint counts as a change), because the change is attributed to the respondent's group.

Examples

```
eq5d_profile_dimension_change_table(
  df = example_data,
  name_id = "id",
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op", "Post-op")
)
```

eq5d_profile_health_profile_grid

eq5d_profile_health_profile_grid: Health Profile Grid (HPG) for Two Time Points

Description

This function creates a Health Profile Grid (HPG) for EQ-5D data, plotting each individual's change in health states (ranked from best to worst) between two time points. A diagonal reference line indicates no change; points above the line reflect improvement, and points below indicate deterioration.

Usage

```
eq5d_profile_health_profile_grid(
  df,
  names_eq5d,
  name_fu,
  levels_fu = NULL,
  name_id,
  eq5d_version,
  country
)
```

Arguments

df	A data frame containing EQ-5D columns, an ID column and a follow-up column
names_eq5d	A character vector of EQ-5D dimension names
name_fu	A character string for the follow-up column
levels_fu	A character vector of length 2, specifying the order of the follow-up levels (e.g., c("Pre-op", "Post-op"))
name_id	A character string for the patient ID column
eq5d_version	Version of the EQ-5D instrument: "3L", "5L" or "Y3L". It decides how many states are ranked.
country	Value set code used to rank the states; see eqvs_display for the codes available.

Details

The ranking is over **every health state the instrument allows**, not only those the data contain: all 243 states for the EQ-5D-3L and the EQ-5D-Y-3L, and all 3,125 for the EQ-5D-5L. Each is valued with the value set named by country and eq5d_version, and ordered best to worst, so the best possible state is rank 1 and the worst is rank 243 or 3,125. A rank therefore means the same thing in every dataset valued with the same value set, and the axes span the whole classification system rather than the part of it a particular sample happens to occupy.

This is why the function takes a value set rather than a column of pre-calculated values: the states that do *not* occur in the data still have to be valued in order to place the states that do.

Value

A list with components:

plot_data	The plot data with ranks and classification.
p	A ggplot2 object displaying the HPG scatter plot.

Pairing records

Each respondent's records are compared in timepoint order, using only the timepoints in `levels_fu`. The function stops, naming the respondents, if a respondent has more than one record at the same timepoint: it cannot tell which to compare, and does not choose. Where the analysis is split by `name_groupvar`, it also stops if a respondent is not in the same group at every timepoint (a missing group at one timepoint counts as a change), because the change is attributed to the respondent's group.

Examples

```
tmp <- eq5d_profile_health_profile_grid(
  df = example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op", "Post-op"),
  name_id = "id",
  eq5d_version = "3L",
  country = "GB"
)
```

eq5d_profile_level_summary

eq5d_profile_level_summary: Frequency of levels by dimensions, cross-sectional

Description

eq5d_profile_level_summary: Frequency of levels by dimensions, cross-sectional

Usage

```
eq5d_profile_level_summary(df, names_eq5d = NULL, eq5d_version = NULL)
```

Arguments

df	Data frame with the EQ-5D and follow-up columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
eq5d_version	Version of the EQ-5D instrument

Value

Summary data frame.

Examples

```
eq5d_profile_level_summary(
  df = example_data[example_data$time == "Pre-op",],
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L"
)
```

```
eq5d_profile_level_summary_by_group
```

eq5d_profile_level_summary_by_group: Frequency of levels by dimensions, separated by category

Description

eq5d_profile_level_summary_by_group: Frequency of levels by dimensions, separated by category

Usage

```
eq5d_profile_level_summary_by_group(
  df,
  names_eq5d = NULL,
  name_cat = NULL,
  levels_cat = NULL,
  eq5d_version = NULL
)
```

Arguments

df	Data frame with the EQ-5D and follow-up columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
name_cat	Character string for the category column. If NULL, no grouping is used, and the table reports for the total population, i.e. equal to table 1.1.1.
levels_cat	Character vector containing the order of the values in the category column, if the wish is to have these presented in a particular order. If NULL (default value), unless the variable is a factor, the levels will be ordered in the order of appearance in df.
eq5d_version	Version of the EQ-5D instrument

Value

Summary data frame.

Examples

```
eq5d_profile_level_summary_by_group(
  df = example_data[example_data$time == "Pre-op",],
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_cat = "procedure",
  levels_cat = c("Hip Replacement", "Knee Replacement"),
  eq5d_version = "3L"
)
```

```
eq5d_profile_lfs_distribution
```

eq5d_profile_lfs_distribution: Distribution of the EQ-5D states by LFS (Level Frequency Score)

Description

eq5d_profile_lfs_distribution: Distribution of the EQ-5D states by LFS (Level Frequency Score)

Usage

```
eq5d_profile_lfs_distribution(df, names_eq5d = NULL, eq5d_version = NULL)
```

Arguments

df	Data frame with the EQ-5D columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
eq5d_version	Version of the EQ-5D instrument

Value

Summary data frame

Examples

```
eq5d_profile_lfs_distribution(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L"
)
```

eq5d_profile_lfs_mean_utility

eq5d_profile_lfs_mean_utility: Number of observations in the LFS (Level Frequency Score) according to the EQ-5D values

Description

eq5d_profile_lfs_mean_utility: Number of observations in the LFS (Level Frequency Score) according to the EQ-5D values

Usage

```
eq5d_profile_lfs_mean_utility(
  df,
  names_eq5d = NULL,
  name_utility = NULL,
  eq5d_version = NULL
)
```

Arguments

df	Data frame with the EQ-5D dimension and value columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
eq5d_version	Version of the EQ-5D instrument

Value

Summary data frame

Examples

```
example_data$value <- eq5d3l(example_data[, c("mo", "sc", "ua", "pd", "ad")],
                             country = "GB")
eq5d_profile_lfs_mean_utility(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_utility = "value",
  eq5d_version = "3L"
)
```

eq5d_profile_lfs_utility_plot

eq5d_profile_lfs_utility_plot: EQ-5D values plotted against LFS

Description

eq5d_profile_lfs_utility_plot: EQ-5D values plotted against LFS

Usage

```
eq5d_profile_lfs_utility_plot(
  df,
  names_eq5d = NULL,
  name_utility = NULL,
  eq5d_version = NULL
)
```

Arguments

df	Data frame with the EQ-5D dimension and value columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
eq5d_version	Version of the EQ-5D instrument

Value

Summary plot and data used for plotting

Examples

```
example_data$value <- eq5d3l(example_data[, c("mo", "sc", "ua", "pd", "ad")],
                             country = "GB")
tmp <- eq5d_profile_lfs_utility_plot(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_utility = "value",
  eq5d_version = "3L"
)
tmp$p
tmp$plot_data
```

eq5d_profile_lfs_utility_summary

eq5d_profile_lfs_utility_summary: Summary statistics of EQ-5D values by LFS (Level Frequency Score)

Description

eq5d_profile_lfs_utility_summary: Summary statistics of EQ-5D values by LFS (Level Frequency Score)

Usage

```
eq5d_profile_lfs_utility_summary(
  df,
  names_eq5d = NULL,
  name_utility = NULL,
  eq5d_version = NULL
)
```

Arguments

df	Data frame with the EQ-5D dimension and value columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
eq5d_version	Version of the EQ-5D instrument

Value

Summary data frame

Examples

```
example_data$value <- eq5d3l(example_data[, c("mo", "sc", "ua", "pd", "ad")],
                             country = "GB")
eq5d_profile_lfs_utility_summary(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_utility = "value",
  eq5d_version = "3L"
)
```

eq5d_profile_lss_utility_plot

eq5d_profile_lss_utility_plot: EQ-5D values plotted against LSS

Description

eq5d_profile_lss_utility_plot: EQ-5D values plotted against LSS

Usage

```
eq5d_profile_lss_utility_plot(
  df,
  names_eq5d = NULL,
  name_utility = NULL,
  eq5d_version = NULL
)
```

Arguments

df	Data frame with the EQ-5D dimension and value columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
eq5d_version	Version of the EQ-5D instrument

Value

Summary plot and data used for plotting

Examples

```
df <- data.frame(make_all_EQ_states(version = "5L"))
df$value <- eq5d5l(df[, c("mo", "sc", "ua", "pd", "ad")], country = "US")
tmp <- eq5d_profile_lss_utility_plot(
  df,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_utility = "value",
  eq5d_version = "5L"
)
tmp$p
tmp$plot_data
```

eq5d_profile_lss_utility_summary

eq5d_profile_lss_utility_summary: Summary statistics for the EQ-5D values by all the different LSSs (Level Sum Scores)

Description

eq5d_profile_lss_utility_summary: Summary statistics for the EQ-5D values by all the different LSSs (Level Sum Scores)

Usage

```
eq5d_profile_lss_utility_summary(
  df,
  names_eq5d = NULL,
  name_utility = NULL,
  eq5d_version = NULL
)
```

Arguments

df	Data frame with the EQ-5D dimension and value columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
eq5d_version	Version of the EQ-5D instrument

Value

Summary data frame

Examples

```
df <- data.frame(make_all_EQ_states(version = "5L"))
df$value <- eq5d5l(df[, c("mo", "sc", "ua", "pd", "ad")], country = "US")
eq5d_profile_lss_utility_summary(
  df,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_utility = "value",
  eq5d_version = "5L"
)
```

eq5d_profile_mixed_dimensions_by_group_plot

eq5d_profile_mixed_dimensions_by_group_plot: Percentage of Respondents Who Had a Mixed Change Overall, by Dimension Improved or Worsened, Grouped by Procedure (or Other Grouping)

Description

This function focuses on patients classified as having "Mixed change" overall (i.e., some dimensions improved, others worsened). It then examines which dimensions improved vs. worsened for each subject. Results are summarized by a grouping variable (e.g., procedure) and time points. The final output is a table plus a ggplot object.

Usage

```
eq5d_profile_mixed_dimensions_by_group_plot(
  df,
  name_id,
  name_groupvar = NULL,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL
)
```

Arguments

df	Data frame containing columns for EQ-5D dimensions, a grouping variable, a patient ID, and a follow-up variable
name_id	Character string indicating the patient ID column
name_groupvar	Character string for the grouping column (e.g. "procedure"). If NULL (default), the analysis is performed on the full population.
names_eq5d	Character vector naming the EQ-5D dimensions (e.g. c("mo", "sc", "ua", "pd", "ad"))
name_fu	Character string for the follow-up column (e.g. "time")
levels_fu	Character vector of length 2 specifying the time order (e.g. c("Pre-op", "Post-op"))

Value

A list with two elements:

plot_data	A wide-format data frame of dimension-specific improvements/worsenings for "Mixed change"
p	A ggplot2 object showing a dimension-level bar chart from .pchc_plot_by_dim

Pairing records

Each respondent's records are compared in timepoint order, using only the timepoints in `levels_fu`. The function stops, naming the respondents, if a respondent has more than one record at the same timepoint: it cannot tell which to compare, and does not choose. Where the analysis is split by `name_groupvar`, it also stops if a respondent is not in the same group at every timepoint (a missing group at one timepoint counts as a change), because the change is attributed to the respondent's group.

Examples

```
result <- eq5d_profile_mixed_dimensions_by_group_plot(
  df = example_data,
  name_id = "id",
  name_groupvar = "procedure",
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op", "Post-op")
)
result$plot_data
result$p
```

eq5d_profile_pchc_by_group_plot

eq5d_profile_pchc_by_group_plot: Paretian Classification of Health Change (PCHC) by Group This function computes PCHC categories between two time points for each subject, stratifies them by a grouping variable, and produces a single bar chart with side-by-side bars showing the distribution of PCHC categories.

Description

`eq5d_profile_pchc_by_group_plot`: Paretian Classification of Health Change (PCHC) by Group This function computes PCHC categories between two time points for each subject, stratifies them by a grouping variable, and produces a single bar chart with side-by-side bars showing the distribution of PCHC categories.

Usage

```
eq5d_profile_pchc_by_group_plot(
  df,
  name_id,
  name_groupvar = NULL,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL
)
```

Arguments

<code>df</code>	Data frame containing EQ-5D dimensions, a grouping variable, patient ID, and follow-up columns
<code>name_id</code>	Character string for the patient ID column
<code>name_groupvar</code>	Character string for the grouping column (e.g., procedure). If NULL (default), the analysis is performed on the full population.
<code>names_eq5d</code>	Character vector of column names for the EQ-5D dimensions
<code>name_fu</code>	Character string for the follow-up column
<code>levels_fu</code>	Character vector of length 2 indicating the order of follow-up time points (e.g., <code>c("Pre-op", "Post-op")</code>)

Value

A list with two elements:

<code>plot_data</code>	A tibble of PCHC percentages by group
<code>p</code>	A <code>ggplot2</code> object showing a bar chart with side-by-side bars for each PCHC category

Pairing records

Each respondent's records are compared in timepoint order, using only the timepoints in `levels_fu`. The function stops, naming the respondents, if a respondent has more than one record at the same timepoint: it cannot tell which to compare, and does not choose. Where the analysis is split by `name_groupvar`, it also stops if a respondent is not in the same group at every timepoint (a missing group at one timepoint counts as a change), because the change is attributed to the respondent's group.

Examples

```
result <- eq5d_profile_pchc_by_group_plot(
  df = example_data,
  name_id = "id",
  name_groupvar = "procedure",
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op", "Post-op")
)
result$p           # shows the plot
result$plot_data   # shows the summary table
```

eq5d_profile_pchc_table

eq5d_profile_pchc_table: Changes in health according to the PCHC (Paretian Classification of Health Change)

Description

eq5d_profile_pchc_table: Changes in health according to the PCHC (Paretian Classification of Health Change)

Usage

```
eq5d_profile_pchc_table(
  df,
  name_id,
  name_groupvar = NULL,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL
)
```

Arguments

df	Data frame with the EQ-5D, grouping, id and follow-up columns
name_id	Character string for the patient id column
name_groupvar	Character string for the grouping column. If NULL (default), the analysis is performed on the full population.
names_eq5d	Character vector of column names for the EQ-5D dimensions
name_fu	Character string for the follow-up column
levels_fu	Character vector containing the order of the values in the follow-up column. If NULL (default value), the levels will be ordered in the order of appearance in df.

Value

Summary data frame

Pairing records

Each respondent's records are compared in timepoint order, using only the timepoints in `levels_fu`. The function stops, naming the respondents, if a respondent has more than one record at the same timepoint: it cannot tell which to compare, and does not choose. Where the analysis is split by `name_groupvar`, it also stops if a respondent is not in the same group at every timepoint (a missing group at one timepoint counts as a change), because the change is attributed to the respondent's group.

Examples

```
eq5d_profile_pchc_table(
  df = example_data,
  name_id = "id",
  name_groupvar = "procedure",
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op", "Post-op")
)
```

```
eq5d_profile_pchc_with_no_problems_table
```

eq5d_profile_pchc_with_no_problems_table: Changes in health according to the PCHC, taking account of those with no problems

Description

eq5d_profile_pchc_with_no_problems_table: Changes in health according to the PCHC, taking account of those with no problems

Usage

```
eq5d_profile_pchc_with_no_problems_table(
  df,
  name_id,
  name_groupvar = NULL,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL
)
```

Arguments

df	Data frame with the EQ-5D, grouping, id and follow-up columns
name_id	Character string for the patient id column
name_groupvar	Character string for the grouping column. If NULL (default), the analysis is performed on the full population.
names_eq5d	Character vector of column names for the EQ-5D dimensions
name_fu	Character string for the follow-up column
levels_fu	Character vector containing the order of the values in the follow-up column. If NULL (default value), the levels will be ordered in the order of appearance in df.

Value

Summary data frame. For each group and later timepoint, `_n` is a count and `_p` a proportion: of everyone for "No problems" and "Total with problems", and of those with problems for the four change categories. Where nobody in a group had problems at a timepoint, those four proportions are NA, as they have no denominator.

Pairing records

Each respondent's records are compared in timepoint order, using only the timepoints in `levels_fu`. The function stops, naming the respondents, if a respondent has more than one record at the same timepoint: it cannot tell which to compare, and does not choose. Where the analysis is split by `name_groupvar`, it also stops if a respondent is not in the same group at every timepoint (a missing group at one timepoint counts as a change), because the change is attributed to the respondent's group.

Examples

```
eq5d_profile_pchc_with_no_problems_table(
  df = example_data,
  name_id = "id",
  name_groupvar = "procedure",
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op", "Post-op")
)
```

eq5d_profile_shannon *eq5d_profile_shannon: Shannon's informativity indices*

Description

Reports Shannon's index H' , its maximum $H'_{\max}$, and Shannon's evenness index J' , for each EQ-5D dimension and for the health state as a whole. These are the informativity measures of chapter 4 of Devlin et al. (2020): how much of the classification system a sample actually uses.

Usage

```
eq5d_profile_shannon(
  df,
  names_eq5d = NULL,
  eq5d_version = NULL,
  name_fu = NULL,
  levels_fu = NULL
)
```

Arguments

<code>df</code>	Data frame with the EQ-5D columns
<code>names_eq5d</code>	Character vector of column names for the EQ-5D dimensions
<code>eq5d_version</code>	Version of the EQ-5D instrument: "3L", "5L" or "Y3L". Matching is case-insensitive. It decides L , so it changes H' max and J' .
<code>name_fu</code>	Character string for the follow-up column. If NULL (default), the indices are reported for the sample as a whole.
<code>levels_fu</code>	Character vector containing the order of the values in the follow-up column. If NULL (default value), the levels will be ordered in the order of appearance in <code>df</code> .

Details

For a variable with categories i holding proportions p_i of the observations,

$$H' = - \sum_i p_i \log_2 p_i, \quad H'_{max} = \log_2 L, \quad J' = H' / H'_{max}$$

where L is the number of categories the instrument allows. For a dimension that is 3 (EQ-5D-3L, EQ-5D-Y-3L) or 5 (EQ-5D-5L); for the health state it is $3^5 = 243$ or $5^5 = 3125$. L is what the instrument allows, not what the sample happens to contain, so J' is comparable between samples.

H' is in bits and rises with both the number of categories used and how evenly they are used. J' is H' as a fraction of its maximum, from 0 when every observation falls in one category to 1 when they are spread evenly over all L . Reporting both is the recommendation: H' alone cannot be compared between instruments with different numbers of categories.

Each dimension is computed over its own non-missing responses. The health state row uses complete profiles only, since a profile with any dimension missing is not a health state.

Value

A data frame with one row per EQ-5D dimension and a final row for the health state, and three columns per follow-up level: `H_<level>`, `Hmax_<level>` and `J_<level>`. Without `name_fu` there is a single set of columns, suffixed `All`.

References

Devlin N, Parkin D, Janssen B (2020). Methods for Analysing and Reporting EQ-5D Data. Springer. doi:[10.1007/9783030476229](https://doi.org/10.1007/9783030476229)

Examples

```
eq5d_profile_shannon(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L"
)

# By time point
```

```
eq5d_profile_shannon(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L",
  name_fu = "time",
  levels_fu = c("Pre-op", "Post-op")
)
```

eq5d_profile_top_states

eq5d_profile_top_states: Prevalence of the 10 most frequently observed self-reported health states

Description

eq5d_profile_top_states: Prevalence of the 10 most frequently observed self-reported health states

Usage

```
eq5d_profile_top_states(df, names_eq5d = NULL, eq5d_version = NULL, n = 10)
```

Arguments

df	Data frame with the EQ-5D columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
eq5d_version	Version of the EQ-5D instrument
n	Number of most frequently observed states to display (default 10)

Value

Summary data frame

Examples

```
eq5d_profile_top_states(
  df = example_data[example_data$time == "Pre-op",],
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L",
  n = 10
)
```

eq5d_profile_worse_dimensions_by_group_plot

This function identifies respondents with a "Worsen" PCHC state (i.e., overall health state got worse between levels_fu[1] and levels_fu[2]), checks dimension-specific changes (e.g., mo_diff < 0), and summarizes by a grouping variable (e.g., procedure) and time points. It returns a data table and a ggplot object.

Description

This function identifies respondents with a "Worsen" PCHC state (i.e., overall health state got worse between levels_fu[1] and levels_fu[2]), checks dimension-specific changes (e.g., mo_diff < 0), and summarizes by a grouping variable (e.g., procedure) and time points. It returns a data table and a ggplot object.

Usage

```
eq5d_profile_worse_dimensions_by_group_plot(
  df,
  name_id,
  name_groupvar = NULL,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL
)
```

Arguments

df	A data frame containing EQ-5D columns, a grouping variable, an ID column, and a follow-up column
name_id	A character string for the patient ID column
name_groupvar	A character string for the grouping column (e.g., procedure). If NULL (default), the analysis is performed on the full population.
names_eq5d	A character vector of EQ-5D dimension names
name_fu	A character string for the follow-up column
levels_fu	A character vector of length 2, specifying the order of the follow-up levels (e.g., c("Pre-op", "Post-op"))

Value

A list containing:

plot_data	A data frame of "Worsen" percentages by group and dimension
p	A ggplot2 object produced by 'pchc_plot_by_dim()'

Pairing records

Each respondent's records are compared in timepoint order, using only the timepoints in `levels_fu`. The function stops, naming the respondents, if a respondent has more than one record at the same timepoint: it cannot tell which to compare, and does not choose. Where the analysis is split by `name_groupvar`, it also stops if a respondent is not in the same group at every timepoint (a missing group at one timepoint counts as a change), because the change is attributed to the respondent's group.

Examples

```
result <- eq5d_profile_worse_dimensions_by_group_plot(
  df = example_data,
  name_id = "id",
  name_groupvar = "procedure",
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op", "Post-op")
)
result$p           # shows the plot
result$plot_data   # shows the summary table
```

eq5d_utility_by_group_plot

eq5d_utility_by_group_plot: Mean EQ-5D values and 95% confidence intervals: all vs by groupvar

Description

eq5d_utility_by_group_plot: Mean EQ-5D values and 95% confidence intervals: all vs by groupvar

Usage

```
eq5d_utility_by_group_plot(df, name_utility = NULL, name_groupvar = NULL)
```

Arguments

<code>df</code>	Data frame with the EQ-5D value and grouping columns
<code>name_utility</code>	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
<code>name_groupvar</code>	Character string for the grouping column. If NULL (default), the analysis is performed on the full population.

Value

Summary plot and data used for plotting

Confidence intervals

The 95% confidence interval of a mean is the mean plus or minus 1.96 standard errors, the standard error being the standard deviation divided by the square root of the number of values: a normal approximation, which is less accurate for small groups.

Examples

```
example_data$value <- eq5d31(example_data[, c("mo", "sc", "ua", "pd", "ad")],
                             country = "GB")
tmp <- eq5d_utility_by_group_plot(
  example_data,
  name_utility = "value",
  name_groupvar = "procedure"
)
tmp$p
tmp$plot_data
```

```
eq5d_utility_change_by_group_plot
```

*eq5d_utility_change_by_group_plot: EQ-5D values: smoothed lines
and confidence intervals by groupvar*

Description

eq5d_utility_change_by_group_plot: EQ-5D values: smoothed lines and confidence intervals by groupvar

Usage

```
eq5d_utility_change_by_group_plot(
  df,
  name_utility = NULL,
  name_fu = NULL,
  levels_fu = NULL,
  name_groupvar = NULL
)
```

Arguments

df	Data frame with the EQ-5D value, follow-up and grouping columns
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
name_fu	Character string for the follow-up column
levels_fu	Character vector containing the order of the values in the follow-up column. If NULL (default value), the levels will be ordered in the order of appearance in df.

name_groupvar Character string for the grouping column. If NULL (default), the analysis is performed on the full population.

Value

Summary plot and data used for plotting

Confidence intervals

The 95% confidence interval of a mean is the mean plus or minus 1.96 standard errors, the standard error being the standard deviation divided by the square root of the number of values: a normal approximation, which is less accurate for small groups.

Examples

```
example_data$value <- eq5d3l(example_data[, c("mo", "sc", "ua", "pd", "ad")],
                             country = "GB")
tmp <- eq5d_utility_change_by_group_plot(
  example_data,
  name_utility = "value",
  name_fu = "time",
  levels_fu = c('Pre-op', 'Post-op'),
  name_groupvar = "procedure"
)
tmp$p
tmp$plot_data
```

eq5d_utility_distribution_plot

eq5d_utility_distribution_plot: EQ-5D values: smoothed lines and confidence intervals by groupvar

Description

eq5d_utility_distribution_plot: EQ-5D values: smoothed lines and confidence intervals by groupvar

Usage

```
eq5d_utility_distribution_plot(df, name_utility = NULL)
```

Arguments

df Data frame with the EQ-5D value column

name_utility Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see [eq5d](#) for calculating them from the dimensions.

Value

Summary plot and data used for plotting

Examples

```
example_data$value <- eq5d3l(example_data[, c("mo", "sc", "ua", "pd", "ad")],
                             country = "GB")
tmp <- eq5d_utility_distribution_plot(
  example_data,
  name_utility = "value"
)
tmp$p
tmp$plot_data
```

eq5d_utility_norms_comparison

eq5d_utility_norms_comparison:EQ-5D values: by age and groupvar

Description

eq5d_utility_norms_comparison:EQ-5D values: by age and groupvar

Usage

```
eq5d_utility_norms_comparison(
  df,
  name_utility = NULL,
  name_fu = NULL,
  levels_fu = NULL,
  name_groupvar = NULL,
  name_age
)
```

Arguments

df	Data frame with the EQ-5D value, age, follow-up and grouping columns
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
name_fu	Character string for the follow-up column
levels_fu	Character vector containing the order of the values in the follow-up column. If NULL (default value), the levels will be ordered in the order of appearance in df. The first level is the baseline, and only records at the baseline are summarised.
name_groupvar	Character string for the grouping column. If NULL (default), the analysis is performed on the full population.
name_age	Character string for the age column

Details

The comparison with population norms describes the sample at baseline: only records at the first timepoint in `levels_fu` are used, so each respondent counts once, and a message says which timepoint that is. Records at other timepoints, or with a missing timepoint, are not included.

Value

Summary data frame

Examples

```
example_data$ageband <- factor(
  example_data$ageband,
  levels = c("20 to 29", "30 to 39", "40 to 49", "50 to 59", "60 to 69", "70 to 79", "80 to 89")
)
example_data <- example_data[example_data$gender %in% c("Male", "Female"),]
example_data$value <- eq5d3l(example_data[, c("mo", "sc", "ua", "pd", "ad")],
  country = "GB")
eq5d_utility_norms_comparison(
  example_data,
  name_utility = "value",
  name_fu = "time",
  levels_fu = c('Pre-op', 'Post-op'),
  name_groupvar = "gender",
  name_age = "ageband"
)
```

eq5d_utility_over_time_plot

eq5d_utility_over_time_plot: EQ-5D values by timepoints: mean values and 95% confidence intervals

Description

eq5d_utility_over_time_plot: EQ-5D values by timepoints: mean values and 95% confidence intervals

Usage

```
eq5d_utility_over_time_plot(
  df,
  name_utility = NULL,
  name_fu = NULL,
  levels_fu = NULL
)
```

Arguments

df	Data frame with the EQ-5D value and follow-up columns
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
name_fu	Character string for the follow-up column
levels_fu	Character vector containing the order of the values in the follow-up column. If NULL (default value), the levels will be ordered in the order of appearance in df.

Value

Summary plot and data used for plotting

Confidence intervals

The 95% confidence interval of a mean is the mean plus or minus 1.96 standard errors, the standard error being the standard deviation divided by the square root of the number of values: a normal approximation, which is less accurate for small groups.

Examples

```
example_data$value <- eq5d3l(example_data[, c("mo", "sc", "ua", "pd", "ad")],
                             country = "GB")
tmp <- eq5d_utility_over_time_plot(
  example_data,
  name_utility = "value",
  name_fu = "time",
  levels_fu = c('Pre-op', 'Post-op')
)
tmp$p
tmp$plot_data
```

eq5d_utility_summary *eq5d_utility_summary: EQ-5D values: by timepoints*

Description

eq5d_utility_summary: EQ-5D values: by timepoints

Usage

```
eq5d_utility_summary(df, name_utility = NULL, name_fu = NULL, levels_fu = NULL)
```

Arguments

df	Data frame with the EQ-5D value and follow-up columns
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
name_fu	Character string for the follow-up column
levels_fu	Character vector containing the order of the values in the follow-up column. If NULL (default value), the levels will be ordered in the order of appearance in df.

Details

Skewness and kurtosis are the population (biased) estimators $m_3/m_2^{3/2}$ and m_4/m_2^2 , as returned by `moments::skewness()` and `moments::kurtosis()`.

The kurtosis is **non-excess**: a normal distribution gives 3, not 0. The row is labelled "Kurtosis (non-excess)" so that the output says which convention it uses. Stata's `summarize`, `detail` reports the same quantity. Excel's `KURT()` reports **excess** kurtosis with a sample-bias correction, so it is roughly 3 lower for the same data.

Value

Summary data frame with one row per statistic and one column per follow-up level, in the order given by `levels_fu`. "Missing (%)" is a percentage, from 0 to 100, of the rows at that follow-up (the "Total sample" row) with no usable value; it is NA for a follow-up with no rows. Before 2.1.0 it was a proportion.

Examples

```
example_data$value <- eq5d3l(example_data[, c("mo", "sc", "ua", "pd", "ad")],
                             country = "GB")

eq5d_utility_summary(
  example_data,
  name_utility = "value",
  name_fu = "time",
  levels_fu = c('Pre-op', 'Post-op')
)
```

```
eq5d_utility_summary_by_group
```

```
eq5d_utility_summary_by_group:EQ-5D values: by groupvar
```

Description

```
eq5d_utility_summary_by_group:EQ-5D values: by groupvar
```

Usage

```
eq5d_utility_summary_by_group(df, name_utility = NULL, name_groupvar = NULL)
```

Arguments

df	Data frame with the EQ-5D value and grouping columns
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
name_groupvar	Character string for the grouping column. If NULL (default), the analysis is performed on the full population.

Value

Summary data frame

Examples

```
example_data$value <- eq5d3l(example_data[, c("mo", "sc", "ua", "pd", "ad")],
                             country = "GB")
eq5d_utility_summary_by_group(
  example_data,
  name_utility = "value",
  name_groupvar = "procedure"
)
```

```
eq5d_utility_vas_scatter_plot
```

eq5d_utility_vas_scatter_plot: EQ-5D values: smoothed lines and confidence intervals by groupvar

Description

eq5d_utility_vas_scatter_plot: EQ-5D values: smoothed lines and confidence intervals by groupvar

Usage

```
eq5d_utility_vas_scatter_plot(df, name_utility = NULL, name_vas = NULL)
```

Arguments

df	Data frame with the EQ-5D value and EQ VAS columns
name_utility	Character string naming the column of pre-calculated EQ-5D values. If NULL (default), the column "utility" is used. The values are analysed as they stand; see eq5d for calculating them from the dimensions.
name_vas	Character string for the VAS column

Value

Summary plot and data used for plotting

Examples

```
example_data$value <- eq5d3l(example_data[, c("mo", "sc", "ua", "pd", "ad")],
                             country = "GB")
tmp <- eq5d_utility_vas_scatter_plot(
  example_data,
  name_utility = "value",
  name_vas = "vas"
)
tmp$p
tmp$plot_data
```

eq5d_vas_distribution_table

eq5d_vas_distribution_table: EQ VAS Scores frequency of mid-points

Description

eq5d_vas_distribution_table: EQ VAS Scores frequency of mid-points

Usage

```
eq5d_vas_distribution_table(df, name_vas = NULL, add_na_total = TRUE)
```

Arguments

df	Data frame with the VAS column
name_vas	Character string for the VAS column
add_na_total	Logical, whether to add summary of the missing, and across the Total, data

Value

Summary data frame

Examples

```
eq5d_vas_distribution_table(
  example_data,
  name_vas = 'vas',
  add_na_total = TRUE
)
```

`eq5d_vas_grouped_distribution_plot`*eq5d_vas_grouped_distribution_plot: Mid-point EQ VAS scores*

Description

eq5d_vas_grouped_distribution_plot: Mid-point EQ VAS scores

Usage

```
eq5d_vas_grouped_distribution_plot(df, name_vas = NULL)
```

Arguments

df	Data frame with the VAS column
name_vas	Character string for the VAS column

Value

Summary plot and data used for plotting

Examples

```
tmp <- eq5d_vas_grouped_distribution_plot(example_data, name_vas = 'vas')
tmp$p
tmp$plot_data
```

`eq5d_vas_histogram`*eq5d_vas_histogram: EQ VAS scores*

Description

eq5d_vas_histogram: EQ VAS scores

Usage

```
eq5d_vas_histogram(df, name_vas = NULL)
```

Arguments

df	Data frame with the VAS column
name_vas	Character string for the VAS column

Value

Summary plot and data used for plotting

Examples

```
tmp <- eq5d_vas_histogram(example_data, name_vas = 'vas')
tmp$p
tmp$plot_data
```

eq5d_vas_summary	<i>eq5d_vas_summary: EQ VAS Score by timepoints</i>
------------------	---

Description

eq5d_vas_summary: EQ VAS Score by timepoints

Usage

```
eq5d_vas_summary(df, name_vas = NULL, name_fu = NULL, levels_fu = NULL)
```

Arguments

df	Data frame with the VAS and the follow-up columns
name_vas	Character string for the VAS column
name_fu	Character string for the follow-up column
levels_fu	Character vector containing the order of the values in the follow-up column.

Details

Skewness and kurtosis are the population (biased) estimators $m_3/m_2^{3/2}$ and m_4/m_2^2 , as returned by `moments::skewness()` and `moments::kurtosis()`.

The kurtosis is **non-excess**: a normal distribution gives 3, not 0. The row is labelled "Kurtosis (non-excess)" so that the output says which convention it uses. Stata's `summarize`, `detail` reports the same quantity. Excel's `KURT()` reports **excess** kurtosis with a sample-bias correction, so it is roughly 3 lower for the same data.

Value

Summary data frame with one row per statistic and one column per follow-up level, in the order given by `levels_fu`. "Missing (%)" is a percentage, from 0 to 100, of the rows at that follow-up (the "Total sample" row) with no usable value; it is NA for a follow-up with no rows. Before 2.1.0 it was a proportion.

Examples

```
eq5d_vas_summary(
  example_data,
  name_vas = 'vas',
  name_fu = 'time',
  levels_fu = c('Pre-op', 'Post-op')
)
```

eq5d3l

*eq5d3l***Description**

Get EQ-5D-3L index values from individual responses to the five dimensions of the EQ-5D-3L.

Usage

```
eq5d3l(x, country = NULL, dim.names = c("mo", "sc", "ua", "pd", "ad"))
```

Arguments

x	A vector of 5-digit EQ-5D-3L state indexes, or a matrix/data.frame with columns corresponding to the EQ-5D-3L dimensions.
country	A country code or value set code identifying the value set to use, as listed by <code>eqvs_display()</code> . Matching is case-insensitive. For countries with more than one value set (for example Germany, with "DE_TT0" and "DE_VAS"), the value set code must be given; supplying the country code alone raises an error listing the available codes.
dim.names	A character vector specifying the names of the EQ-5D-3L dimensions. Default is 'c("mo", "sc", "ua", "pd", "ad")'.

Value

A numeric vector of EQ-5D-3L values, or a data.frame with one column for each requested value set. The result is an unnamed numeric vector, one element per element or row of x and in the same order.

Missing and invalid responses

Each health state is checked before it is scored, and a state that cannot be scored gives NA for that element or row; the others are scored as usual.

- A missing response (NA in any dimension, or an NA state index) gives NA without a warning.
- An invalid response gives NA with a warning, which gives the number of such states and examples. Invalid means a level the instrument does not have (for example 4 in the EQ-5D-3L, or a missing-value code such as 9), a level or index that is not a whole, finite number, or an index that is not five digits long. Each problem is reported once, however many value sets are requested.
- State indexes and levels given as text or as factors are read through their labels, so "21111" and `factor(2)` are scored as 21111 and 2.

Recode missing-value codes to NA first if the warning is not wanted.

Examples

```
# Example 1: utility values from EQ-5D-3L profile codes
eq5d3l(c(11111, 12321, 32123, 33333), country = "US")

# Example 2: request multiple value sets
eq5d3l(make_all_EQ_states("3L"), country = c("DK", "CA"))

# Example 3: use a data.frame with dimension columns
df3l <- data.frame(
  mo = c(1, 2, 3),
  sc = c(1, 2, 2),
  ua = c(1, 3, 1),
  pd = c(2, 2, 3),
  ad = c(1, 1, 2)
)
eq5d3l(df3l, country = "US")

# Example 4: use custom dimension column names
df3l_named <- data.frame(
  mobility = c(1, 2, 3),
  self_care = c(1, 2, 2),
  usual_activities = c(1, 3, 1),
  pain_discomfort = c(2, 2, 3),
  anxiety_depression = c(1, 1, 2)
)
eq5d3l(
  df3l_named,
  country = "US",
  dim.names = c(
    "mobility", "self_care", "usual_activities",
    "pain_discomfort", "anxiety_depression"
  )
)
```

eq5d5l

*eq5d5l***Description**

Get EQ-5D-5L index values from individual responses to the five dimensions of the EQ-5D-5L.

Usage

```
eq5d5l(x, country = NULL, dim.names = c("mo", "sc", "ua", "pd", "ad"))
```

Arguments

x A vector of 5-digit EQ-5D-5L state indexes, or a matrix/data.frame with columns corresponding to the EQ-5D-5L dimensions.

country	A country code or value set code identifying the value set to use, as listed by <code>eqvs_display()</code> . Matching is case-insensitive. For countries with more than one value set (for example Germany, with "DE_TTO" and "DE_VAS"), the value set code must be given; supplying the country code alone raises an error listing the available codes.
dim.names	A character vector specifying the names of the EQ-5D-5L dimensions. Default is <code>c("mo", "sc", "ua", "pd", "ad")</code> .

Value

A numeric vector of EQ-5D-5L values, or a data.frame with one column for each requested value set. The result is an unnamed numeric vector, one element per element or row of `x` and in the same order.

Missing and invalid responses

Each health state is checked before it is scored, and a state that cannot be scored gives NA for that element or row; the others are scored as usual.

- A missing response (NA in any dimension, or an NA state index) gives NA without a warning.
- An invalid response gives NA with a warning, which gives the number of such states and examples. Invalid means a level the instrument does not have (for example 4 in the EQ-5D-3L, or a missing-value code such as 9), a level or index that is not a whole, finite number, or an index that is not five digits long. Each problem is reported once, however many value sets are requested.
- State indexes and levels given as text or as factors are read through their labels, so "21111" and `factor(2)` are scored as 21111 and 2.

Recode missing-value codes to NA first if the warning is not wanted.

Examples

```
# Example 1: utility values from EQ-5D-5L profile codes
eq5d5l(c(11111, 12321, 32423, 55555), country = "IT")

# Example 2: request multiple value sets
eq5d5l(make_all_EQ_states("5L"), country = c("ES", "DE"))

# Example 3: use a data.frame with dimension columns
df5l <- data.frame(
  mo = c(1, 2, 5),
  sc = c(1, 2, 4),
  ua = c(1, 3, 3),
  pd = c(2, 4, 2),
  ad = c(1, 5, 1)
)
eq5d5l(df5l, country = "ES")

# Example 4: use custom dimension column names from a real-world style dataset
df5l_named <- data.frame(
  mobility = c(1, 5, 3),
```

```

    self_care = c(2, 4, 2),
    usual_activities = c(3, 3, 1),
    pain_discomfort = c(4, 2, 2),
    anxiety_depression = c(5, 1, 3)
  )
  eq5d5l(
    df5l_named,
    country = "ES",
    dim.names = c(
      "mobility", "self_care", "usual_activities",
      "pain_discomfort", "anxiety_depression"
    )
  )
)

```

eq5dy3l

eq5dy3l

Description

Get EQ-5D-Y-3L index values from individual responses to the five dimensions of the EQ-5D-Y-3L.

Usage

```
eq5dy3l(x, country = NULL, dim.names = c("mo", "sc", "ua", "pd", "ad"))
```

Arguments

x	A vector of 5-digit EQ-5D-Y-3L state indexes, or a matrix/data.frame with columns corresponding to the EQ-5D-Y-3L dimensions.
country	A country code or value set code identifying the value set to use, as listed by <code>eqvs_display()</code> . Matching is case-insensitive. For countries with more than one value set (for example Germany, with "DE_TT0" and "DE_VAS"), the value set code must be given; supplying the country code alone raises an error listing the available codes.
dim.names	A character vector specifying the names of the EQ-5D-Y-3L dimensions. Default is 'c("mo", "sc", "ua", "pd", "ad")'.

Value

A numeric vector of EQ-5D-Y-3L values, or a data.frame with one column for each requested value set. The result is an unnamed numeric vector, one element per element or row of x and in the same order.

Missing and invalid responses

Each health state is checked before it is scored, and a state that cannot be scored gives NA for that element or row; the others are scored as usual.

- A missing response (NA in any dimension, or an NA state index) gives NA without a warning.
- An invalid response gives NA with a warning, which gives the number of such states and examples. Invalid means a level the instrument does not have (for example 4 in the EQ-5D-3L, or a missing-value code such as 9), a level or index that is not a whole, finite number, or an index that is not five digits long. Each problem is reported once, however many value sets are requested.
- State indexes and levels given as text or as factors are read through their labels, so "21111" and factor(2) are scored as 21111 and 2.

Recode missing-value codes to NA first if the warning is not wanted.

Examples

```
# Example 1: utility values from EQ-5D-Y-3L profile codes
eq5dy3l(x = c(11111, 12321, 33333), country = "SI")
```

```
# Example 2: request multiple value sets
eq5dy3l(make_all_EQ_states("3L"), country = c("ES", "DE"))
```

```
# Example 3: use a data.frame with dimension columns
dfy3l <- data.frame(
  mo = c(1, 2, 3),
  sc = c(1, 1, 2),
  ua = c(1, 2, 3),
  pd = c(2, 2, 3),
  ad = c(1, 3, 2)
)
eq5dy3l(dfy3l, country = "SI")
```

```
# Example 4: use custom dimension column names
dfy3l_named <- data.frame(
  mobility = c(1, 2, 3),
  self_care = c(1, 1, 2),
  usual_activities = c(1, 2, 3),
  pain_discomfort = c(2, 2, 3),
  anxiety_depression = c(1, 3, 2)
)
eq5dy3l(
  dfy3l_named,
  country = "SI",
  dim.names = c(
    "mobility", "self_care", "usual_activities",
    "pain_discomfort", "anxiety_depression"
  )
)
```

eqvs_add

*eqvs_add***Description**

Add user-defined EQ-5D value set and corresponding crosswalk option.

Usage

```
eqvs_add(
  df,
  version = "5L",
  country = NULL,
  countryCode = NULL,
  VSCode = NULL,
  description = NULL,
  saveOption = 1,
  savePath = NULL
)
```

Arguments

df	A data.frame or file name pointing to csv file. The contents of the data.frame or csv file should be exactly two columns: state, containing a list of all 3125 (for 5L) or 243 (for 3L and Y3L) EQ-5D health state vectors, and a column of corresponding utility values, with a suitable name. The EQ-5D-Y-3L uses the same 243 health states as the EQ-5D-3L.
version	Version of the EQ-5D instrument. Can take values 5L (default), 3L or Y3L. Matching is case-insensitive.
country	Optional string. If not NULL, will be used as a country description for the user-defined value set.
countryCode	Optional string. If not NULL, will be used as the two-digit code for the value set. Must be different from any existing national value set code.
VSCode	Optional string. If not NULL, will be used as the code for the value set; otherwise the name of the second column of df is used. It must differ from every built-in value set code, compared without regard to case and across all three instruments, and from the codes of your own value sets for this instrument. eqvs_display() lists the codes already in use.
description	Optional string. If not NULL, will be used as a descriptive text for the user-defined value set.
saveOption	Integer indicating how the cache data should be saved. 1: Do not save (default). 2: Save in this package's user cache directory, as returned by tools::R_user_dir("eq5dsuite", "cache"). It is read back automatically when the package is next loaded. Nothing is written inside the installed package, which CRAN does not permit. 3: Save in the directory given by savePath, to be read back with eqvs_load().

`savePath` A path where the cache data should be saved when ‘`saveOption`’ is 3. Please use ‘`eqvs_load`’ to load it in your next session.

Details

Every check below is made before anything is changed, so a value set that fails one is not added, in part or in full, and nothing is saved.

- `df`, if a file name, must name an existing csv file, and must have exactly two columns.
- It must have exactly one row per health state of the instrument (3,125 for the EQ-5D-5L, 243 for the EQ-5D-3L and EQ-5D-Y-3L), and every state must be present exactly once. The rows may be in any order, and the states may be given as numbers, text or factors; they must be whole numbers, so 11111.9 is refused rather than read as 11111.
- The value set code (VSCode, or the name of the second column) must not be a built-in value set code, compared without regard to case and across all three instruments; this is an error. If it is the code of one of your own value sets for the same instrument, again without regard to case, the function warns and returns `0` without adding it.
- Every value must be numeric and finite: NA, NaN, text, Inf and -Inf are errors.
- With `saveOption` = 3, `savePath` must name an existing directory. With `saveOption` = 2 or 3, if the cache cannot be saved, the value set is removed again.

The values themselves are not checked further: nothing requires full health (11111) to be 1, or the values to fall as the levels worsen.

Value

Called for its side effect. When the value set is added, NULL, invisibly, with a message. When VSCode is already used by one of your own value sets for this instrument, `0`, with a warning, and nothing is added.

Examples

```
# Make a nonsense value set. The values are a plain sequence rather than
# runif(), which would move the random seed of whoever runs the example.
new_df <- data.frame(state = make_all_EQ_indexes(),
                     TEST = seq(1, -0.5, length.out = 3125))
# Add as value set for Fantasia
eqvs_add(
  new_df,
  version = "5L",
  country = 'Fantasia',
  countryCode = "MyCountry",
  VSCode = "FAN",
  saveOption = 1
)
eq5d5l(55555, country = "FAN")
# Remove it again, so the example leaves no value set behind.
eqvs_drop(country = "FAN", version = "5L", saveOption = 1, ask = FALSE)
```

eqvs_display	<i>eqvs_display</i>
--------------	---------------------

Description

Display available value sets, which can also be used as (reverse) crosswalks. Built-in value sets are shown first, followed by any user-defined value sets added via `eqvs_add()`.

Usage

```
eqvs_display(version = "5L", return_df = FALSE, show_citation = FALSE)
```

Arguments

<code>version</code>	Version of the EQ-5D instrument. One of "3L", "5L" (default), or "Y3L". Matching is case-insensitive.
<code>return_df</code>	Logical. If FALSE (default), the summary table is printed to the console. If TRUE, nothing is printed and the value set information is returned as a <code>data.frame</code> instead, with all columns (including <code>citation</code> if present), so that it can be used in further code without cluttering scripts, reports or the console.
<code>show_citation</code>	Logical. If TRUE, prints the full AMA citation for each value set after the summary table. Defaults to FALSE. Applies only when <code>return_df = FALSE</code> : the returned <code>data.frame</code> always includes the <code>citation</code> column, so this argument is ignored when <code>return_df = TRUE</code> .

Value

When `return_df = FALSE`, NULL invisibly, called for the printed output. When `return_df = TRUE`, a `data.frame`, returned visibly so that it still prints when the call is made at the console.

Examples

```
# Print the available EQ-5D-5L value sets.
eqvs_display(version = "5L")

# Get the same information as a data.frame, printing nothing.
vs <- eqvs_display(version = "5L", return_df = TRUE)
head(vs)
```

eqvs_drop

*eqvs_drop***Description**

Drop user-defined EQ-5D value set to reverse crosswalk options.

Usage

```
eqvs_drop(
  country = NULL,
  version = "5L",
  saveOption = 1,
  savePath = NULL,
  ask = TRUE
)
```

Arguments

country	A country code or value set code identifying the user-defined value set to remove, as listed by <code>eqvs_display()</code> . If more than one user-defined value set shares the country code, the value set code must be given; otherwise the function stops and lists the matching codes.
version	Version of the EQ-5D instrument. Can take values 5L (default), 3L or Y3L. Matching is case-insensitive.
saveOption	Integer indicating how the cache data should be saved. 1: Do not save (default). 2: Save in this package's user cache directory, as returned by <code>tools::R_user_dir("eq5dsuite", "cache")</code> . It is read back automatically when the package is next loaded. Nothing is written inside the installed package, which CRAN does not permit. 3: Save in the directory given by <code>savePath</code> , to be read back with <code>eqvs_load()</code> .
savePath	A path where the cache data should be saved when 'saveOption' is 3. Please use 'eqvs_load' to load it in your next session.
ask	Logical. Whether to ask for confirmation before removing the value set. Confirmation is only requested when <code>ask</code> is TRUE <i>and</i> the session is interactive; with <code>ask = FALSE</code> , or in a non-interactive session, the value set is removed without prompting. Defaults to TRUE.

Value

Invisibly TRUE if a value set was removed, and FALSE otherwise (for example if no matching value set exists, or the user declined the confirmation).

Examples

```
# Make a nonsense value set, without moving the random seed.
new_df <- data.frame(state = make_all_EQ_indexes(),
                     TEST = seq(1, -0.5, length.out = 3125))

# Add as value set for Fantasia
eqvs_add(
  new_df,
  version = "5L",
  country = 'Fantasia',
  countryCode = "MyCountry",
  VSCode = "FAN",
  saveOption = 1
)

# Test the new value set
eq5d5l(55555, country = "FAN")

# Drop value set for Fantasia, without asking for confirmation
eqvs_drop(country = 'FAN', saveOption = 1, ask = FALSE)
```

eqvs_load

*eqvs_load***Description**

Load cache data from a specified path.

Usage

```
eqvs_load(loadPath)
```

Arguments

loadPath The path from which to load the cache data.

Details

loadPath is the **directory** a cache was written to with `eqvs_add(saveOption = 3, savePath = ...)` or `eqvs_drop(saveOption = 3, savePath = ...)`, not a single value set. Everything saved there is restored at once.

A cache written with `saveOption = 2` goes to the package's user cache directory and is read back automatically when the package loads, so it needs no call to `eqvs_load()`.

Value

TRUE if loading is successful, FALSE otherwise.

See Also

[eqvs_add](#), [eqvs_drop](#)

Examples

```
# Save a custom value set to a directory of your choosing, then read it
# back. tempdir() is used here so the example leaves nothing behind.
path <- file.path(tempdir(), "eq5d-value-sets")
dir.create(path, showWarnings = FALSE)

my_vs <- data.frame(state = make_all_EQ_indexes("3L"),
                    MY_VS = round(seq(1, -0.5, length.out = 243), 4))
eqvs_add(my_vs, version = "3L", country = "My Country",
         countryCode = "MC", VSCode = "MY_VS",
         saveOption = 3, savePath = path)

# In a later session, restore it from that directory.
eqvs_load(loadPath = path)
eq5d3l(c(11111, 33333), country = "MY_VS")

# Tidy up.
eqvs_drop(country = "MY_VS", version = "3L", ask = FALSE)
unlink(path, recursive = TRUE)
```

eqxw	<i>eqxw</i>
------	-------------

Description

Get crosswalk values

Usage

```
eqxw(x, country = NULL, dim.names = c("mo", "sc", "ua", "pd", "ad"))
```

Arguments

x	A vector of 5-digit EQ-5D-5L state indexes or a matrix/data.frame with columns corresponding to EQ-5D state dimensions
country	A country code or value set code identifying the value set to use, as listed by eqvs_display(). Matching is case-insensitive. For countries with more than one value set (for example Germany, with "DE_TTO" and "DE_VAS"), the value set code must be given; supplying the country code alone raises an error listing the available codes.
dim.names	A vector of dimension names to identify dimension columns

Value

A numeric vector of crosswalk values, or a data.frame with one column per crosswalk set requested. The result is an unnamed numeric vector, one element per element or row of x and in the same order.

Missing and invalid responses

Each health state is checked before it is scored, and a state that cannot be scored gives NA for that element or row; the others are scored as usual.

- A missing response (NA in any dimension, or an NA state index) gives NA without a warning.
- An invalid response gives NA with a warning, which gives the number of such states and examples. Invalid means a level the instrument does not have (for example 4 in the EQ-5D-3L, or a missing-value code such as 9), a level or index that is not a whole, finite number, or an index that is not five digits long. Each problem is reported once, however many value sets are requested.
- State indexes and levels given as text or as factors are read through their labels, so "21111" and factor(2) are scored as 21111 and 2.

Recode missing-value codes to NA first if the warning is not wanted.

Examples

```
eqxw(c(11111, 12521, 32123, 55555), 'US')
eqxw(make_all_EQ_states('5L'), c('DK', 'US'))
```

eqxw_UK	<i>eqxw_UK</i>
---------	----------------

Description

Map EQ-5D-5L health states, or mean EQ-5D-5L values, to EQ-5D values based on the UK EQ-5D-3L value set (Dolan 1997), using the mapping published by the NICE Decision Support Unit (Hernández Alava et al. 2023).

Usage

```
eqxw_UK(x, age, male, dim.names = c("mo", "sc", "ua", "pd", "ad"), bwidth = 0)
```

Arguments

x	EQ-5D-5L health states, as a vector of 5-digit state indexes or as a matrix or data frame with one column per EQ-5D dimension; or a vector of mean EQ-5D values on the UK EQ-5D-5L value set. See Details.
age	Age in years, as a vector or, when x is a data frame, the name of a column in it.
male	1 for male and 0 for female (TRUE/FALSE are accepted), as a vector or, when x is a data frame, a column name.
dim.names	A vector of dimension names to identify dimension columns.

bwidth Kernel half-width, used only for value input: a finite number of at least 0, or one per row. 0 (the default) matches values exactly. "default" applies the DSU's recommended bandwidths for mean values exactly as the DSU's own R command (April 2026 release) does: 0.1 for a value above 0.6 and 0.4 for a value of 0.6 or below. NICE's FAQ to its interim methods statement (PMG51) describes 0.6 itself as in the 0.1 range; the two differ only for a value of exactly 0.6, where a numeric bwidth can be given instead.

Details

NICE's interim methods statement of 27 August 2026 (NICE 2026, PMG51) makes the UK EQ-5D-5L value set the reference case: EQ-5D-5L data collected in a study should be valued directly with it, that is `eq5d5l(x, country = "GB")`. The statement applies to topics started after it was published – for technology appraisals, those whose invitation to participate was issued after that date. For topics started before it, NICE says EQ-5D-5L data should continue to be mapped to EQ-5D-3L utility values, which is what `eqxw_UK()` does; it also reproduces earlier analyses. (Checked against the statement on 4 October 2026; NICE's methods manuals PMG36 and PMG20 are to be updated to match it.)

The mapping is age- and sex-specific. It is not the crosswalk of van Hout et al. (2012) implemented in `eqxw`, and it is not the reverse crosswalk of van Hout and Shaw (2021) implemented in `eqxwr`; those take neither age nor sex. See `eqxwr_UK` for the opposite direction.

Two kinds of input: **Health states**, for individual respondents. Give `x` as 5-digit EQ-5D-5L state indexes, or as a matrix or data frame with one column per dimension. Each respondent's state, age and sex are looked up directly and `bwidth` is ignored. This is the usual case.

EQ-5D values, for a *mean* value such as one reported in a published study. Give `x` as values on the UK EQ-5D-5L value set. There is no unique health state behind a mean, so the mapped value is a weighted average of the mapped values of the states whose EQ-5D value lies near it, using an Epanechnikov kernel of half-width `bwidth` within the age band and sex given. Set `bwidth = "default"` to apply the DSU's recommended bandwidths for mean values: 0.1 for values above 0.6, and 0.4 for values at or below 0.6. **This input is not intended for individual-level data;** map individual respondents from their health states.

Age and sex: Both are required. Age is used in years and grouped into the five bands of the mapping: 16-34, 35-44, 45-54, 55-64 and 65 or over. There is no upper limit. Ages below 16 are outside the estimation sample and return NA with a warning; values of 1 to 5 are treated as ages, never as band numbers. Sex must be 1 for male and 0 for female; TRUE and FALSE are accepted. Any other value returns NA with a warning.

Invalid or missing health states, ages or sexes return NA.

Value

An unnamed numeric vector of mapped EQ-5D values, one per row of `x` and in the same order.

Source

The mapping tables are taken from the publicly available commands for mapping between the EQ-5D-3L and the EQ-5D-5L published by the NICE Decision Support Unit (April 2026 release, which

uses the UK EQ-5D-5L value set of Rowen et al. 2026): <https://sheffield.ac.uk/nice-dsu/methods-development/mapping-eq-5d-5l-3l>

As the DSU requests, work using these mappings cites Hernández Alava, Pudney and Wailoo (2023) and, for the underlying model, Hernández-Alava and Pudney (2017). See `file.show(system.file("COPYRIGHTS", package = "eq5dsuite"))`.

References

Hernández Alava M, Pudney S, Wailoo A (2023). Estimating the Relationship Between EQ-5D-5L and EQ-5D-3L: Results from a UK Population Study. *Pharmacoeconomics* 41(2):199-207. doi:10.1007/s40273022012187

Hernández-Alava M, Pudney S (2017). Econometric modelling of multiple self-reports of health states: The switch from EQ-5D-3L to EQ-5D-5L in evaluating drug therapies for rheumatoid arthritis. *Journal of Health Economics* 55:139-152. doi:10.1016/j.jhealeco.2017.06.013

Rowen D, Mukuria C, Bray N, et al. (2026). A UK value set for the EQ-5D-5L. *Value in Health* 29(5):858-869. doi:10.1016/j.jval.2026.03.008

National Institute for Health and Care Excellence (2026). Interim methods statement: implementing the EQ-5D-5L value set (PMG51). Published 27 August 2026. <https://www.nice.org.uk/process/pmg51>

See Also

[eqxwr_UK](#) for EQ-5D-3L to EQ-5D-5L, [eqxw](#) and [eqxwr](#) for the van Hout crosswalks, and [eq5d5l\(\)](#) for direct valuation.

Examples

```
# health states, for individual respondents
eqxw_UK(c(11111, 12345, 32423, 55555),
        age = c(30, 40, 55, 70), male = c(1, 0, 1, 0))

# a mean EQ-5D-5L value from a published study, with the recommended bandwidth
eqxw_UK(0.72, age = 55, male = 1, bwidth = "default")
```

eqxwr

eqxwr

Description

Get reverse crosswalk values

Usage

```
eqxwr(x, country = NULL, dim.names = c("mo", "sc", "ua", "pd", "ad"))
```

Arguments

x	A vector of 5-digit EQ-5D-3L state indexes or a matrix/data.frame with columns corresponding to EQ-5D state dimensions
country	A country code or value set code identifying the value set to use, as listed by <code>eqvs_display()</code> . Matching is case-insensitive. For countries with more than one value set (for example Germany, with "DE_TTO" and "DE_VAS"), the value set code must be given; supplying the country code alone raises an error listing the available codes.
dim.names	A vector of dimension names to identify dimension columns

Details

This is the reverse crosswalk of van Hout and Shaw (2021), which maps EQ-5D-3L responses to EQ-5D-5L values for any available value set. It does not use age or sex. It is a different mapping from [eqxwr_UK](#), which implements the NICE Decision Support Unit mapping for the United Kingdom only and is age- and sex-specific. Do not substitute one for the other.

Value

A numeric vector of reverse crosswalk values, or a data.frame with one column per reverse crosswalk set requested. The result is an unnamed numeric vector, one element per element or row of x and in the same order.

Missing and invalid responses

Each health state is checked before it is scored, and a state that cannot be scored gives NA for that element or row; the others are scored as usual.

- A missing response (NA in any dimension, or an NA state index) gives NA without a warning.
- An invalid response gives NA with a warning, which gives the number of such states and examples. Invalid means a level the instrument does not have (for example 4 in the EQ-5D-3L, or a missing-value code such as 9), a level or index that is not a whole, finite number, or an index that is not five digits long. Each problem is reported once, however many value sets are requested.
- State indexes and levels given as text or as factors are read through their labels, so "21111" and `factor(2)` are scored as 21111 and 2.

Recode missing-value codes to NA first if the warning is not wanted.

References

van Hout B, Shaw JW (2021). Mapping EQ-5D-3L to EQ-5D-5L. *Value in Health* 24(9):1285-1293. [doi:10.1016/j.jval.2021.03.009](https://doi.org/10.1016/j.jval.2021.03.009)

See Also

[eqxwr_UK](#), [eqxw](#)

Examples

```
eqxwr(c(11111, 12321, 32123, 33333), 'US')
eqxwr(make_all_EQ_states('3L'), c('DK', 'US'))
```

eqxwr_UK	<i>eqxwr_UK</i>
----------	-----------------

Description

Map EQ-5D-3L health states, or mean EQ-5D-3L values, to EQ-5D values based on the UK EQ-5D-5L value set (Rowen et al. 2026), using the mapping published by the NICE Decision Support Unit (Hernández Alava et al. 2023).

Usage

```
eqxwr_UK(x, age, male, dim.names = c("mo", "sc", "ua", "pd", "ad"), bwidth = 0)
```

Arguments

x	EQ-5D-3L health states, as a vector of 5-digit state indexes or as a matrix or data frame with one column per EQ-5D dimension; or a vector of mean EQ-5D values on the UK EQ-5D-3L value set. See Details.
age	Age in years, as a vector or, when x is a data frame, the name of a column in it.
male	1 for male and 0 for female (TRUE/FALSE are accepted), as a vector or, when x is a data frame, a column name.
dim.names	A vector of dimension names to identify dimension columns.
bwidth	Kernel half-width, used only for value input: a finite number of at least 0, or one per row. 0 (the default) matches values exactly. "default" applies the DSU's recommended bandwidths for mean values exactly as the DSU's own R command (April 2026 release) does: 0.1 for a value above 0.6 and 0.4 for a value of 0.6 or below. NICE's FAQ to its interim methods statement (PMG51) describes 0.6 itself as in the 0.1 range; the two differ only for a value of exactly 0.6, where a numeric bwidth can be given instead.

Details

This is the mapping NICE's interim methods statement of 27 August 2026 (NICE 2026, PMG51) specifies when EQ-5D-5L data from a relevant study are not available and EQ-5D-3L data are used instead: the 3L responses are mapped to utility values on the UK EQ-5D-5L value set, the reference case for topics started after that date. (Checked against the statement on 4 October 2026.)

The mapping is age- and sex-specific. It is **not** the reverse crosswalk of van Hout and Shaw (2021) implemented in [eqxwr](#), which takes neither age nor sex and is available for many value sets; the two give different answers and are not interchangeable. See [eqxw_UK](#) for the opposite direction.

Two kinds of input: Health states, for individual respondents. Give `x` as 5-digit EQ-5D-3L state indexes, or as a matrix or data frame with one column per dimension. Each respondent's state, age and sex are looked up directly and `bwidth` is ignored. This is the usual case.

EQ-5D values, for a *mean* value such as one reported in a published study. Give `x` as values on the UK EQ-5D-3L value set. There is no unique health state behind a mean, so the mapped value is a weighted average of the mapped values of the states whose EQ-5D value lies near it, using an Epanechnikov kernel of half-width `bwidth` within the age band and sex given. Set `bwidth = "default"` to apply the DSU's recommended bandwidths for mean values: 0.1 for values above 0.6, and 0.4 for values at or below 0.6. **This input is not intended for individual-level data;** map individual respondents from their health states.

Age and sex: Both are required. Age is used in years and grouped into the five bands of the mapping: 16-34, 35-44, 45-54, 55-64 and 65 or over. There is no upper limit. Ages below 16 are outside the estimation sample and return NA with a warning; values of 1 to 5 are treated as ages, never as band numbers. Sex must be 1 for male and 0 for female; TRUE and FALSE are accepted. Any other value returns NA with a warning.

Invalid or missing health states, ages or sexes return NA.

Value

An unnamed numeric vector of mapped EQ-5D values, one per row of `x` and in the same order.

Source

The mapping tables are taken from the publicly available commands for mapping between the EQ-5D-3L and the EQ-5D-5L published by the NICE Decision Support Unit (April 2026 release, which uses the UK EQ-5D-5L value set of Rowen et al. 2026): <https://sheffield.ac.uk/nice-dsu/methods-development/mapping-eq-5d-5l-3l>

As the DSU requests, work using these mappings cites Hernández Alava, Pudney and Wailoo (2023) and, for the underlying model, Hernández-Alava and Pudney (2017). See `file.show(system.file("COPYRIGHTS", package = "eq5dsuite"))`.

References

- Hernández Alava M, Pudney S, Wailoo A (2023). Estimating the Relationship Between EQ-5D-5L and EQ-5D-3L: Results from a UK Population Study. *Pharmacoeconomics* 41(2):199-207. doi:10.1007/s40273022012187
- Hernández-Alava M, Pudney S (2017). Econometric modelling of multiple self-reports of health states: The switch from EQ-5D-3L to EQ-5D-5L in evaluating drug therapies for rheumatoid arthritis. *Journal of Health Economics* 55:139-152. doi:10.1016/j.jhealeco.2017.06.013
- Rowen D, Mukuria C, Bray N, et al. (2026). A UK value set for the EQ-5D-5L. *Value in Health* 29(5):858-869. doi:10.1016/j.jval.2026.03.008
- National Institute for Health and Care Excellence (2026). Interim methods statement: implementing the EQ-5D-5L value set (PMG51). Published 27 August 2026. <https://www.nice.org.uk/process/pmg51>

See Also

[eqxw_UK](#) for EQ-5D-5L to EQ-5D-3L, and [eqxwr](#) for the van Hout and Shaw reverse crosswalk.

Examples

```
# health states, for individual respondents
eqxwr_UK(c(11111, 12321, 32123, 33333),
         age = c(30, 40, 55, 70), male = c(1, 0, 1, 0))

# a mean EQ-5D-3L value from a published study, with the recommended bandwidth
eqxwr_UK(0.54, age = 35, male = 0, bwidth = "default")
```

example_data

example_data

Description

A sample of patient-level EQ-5D-3L data from the NHS Patient Reported Outcome Measures (PROMs) programme, in long format: 5,000 patients, each with one pre-operative and one post-operative record.

Missing values in the EQ-5D dimensions and in the EQ VAS are recorded with numeric codes rather than NA: 9 in mo, sc, ua, pd and ad (1,323 cells across 497 records), and 999 in vas (671 records). Both the analysis functions and the value functions (`eq5d3l()`, `eq5d()`, `eqxwr()` and so on) treat these as invalid: they set them to NA, with a warning. Recode them to NA first to avoid the warning. ageband and gender use NA directly (970 records each).

Usage

```
data(example_data)
```

Format

A data frame with 10000 rows and 12 variables:

```
id   integer Patient id; each id appears twice
time character Follow-up (Pre-op / Post-op)
mo   integer EQ-5D-3L Mobility dimension (1-3; 9 = missing)
sc   integer EQ-5D-3L Self-care dimension (1-3; 9 = missing)
ua   integer EQ-5D-3L Usual activities dimension (1-3; 9 = missing)
pd   integer EQ-5D-3L Pain/discomfort dimension (1-3; 9 = missing)
ad   integer EQ-5D-3L Anxiety/depression dimension (1-3; 9 = missing)
vas  integer EQ VAS score (0-100; 999 = missing)
procedure character Type of surgery (Groin Hernia / Hip Replacement / Knee Replacement /
  Varicose Vein)
year  character Financial year of the operation, April to March (2013/14 to 2016/17)
ageband character Age in pre-defined ranges (20 to 29, ..., 80 to 89; NA where not recorded)
gender character Patient's gender (Female / Male / Not specified; NA where not recorded)
```

make_all_EQ_indexes	<i>make_all_EQ_indexes</i>
---------------------	----------------------------

Description

Make a vector containing all 5-digit EQ-5D indexes for -3L or -5L version.

Usage

```
make_all_EQ_indexes(  
  version = "5L",  
  dim.names = c("mo", "sc", "ua", "pd", "ad")  
)
```

Arguments

version	"3L", "5L" or "Y3L", to signify whether 243 or 3125 states should be generated. The EQ-5D-Y-3L has the same 243 states as the EQ-5D-3L. Matching is case-insensitive.
dim.names	A vector of dimension names to be used as names for output columns.

Value

A vector with 5-digit state indexes for all 243 (-3L, -Y-3L) or 3125 (-5L) EQ-5D health states

Examples

```
make_all_EQ_indexes('3L')
```

make_all_EQ_states	<i>make_all_EQ_states</i>
--------------------	---------------------------

Description

Make a data.frame with all health states defined by dimensions

Usage

```
make_all_EQ_states(  
  version = "5L",  
  dim.names = c("mo", "sc", "ua", "pd", "ad"),  
  append_index = FALSE  
)
```

Arguments

version	"3L", "5L" or "Y3L", to signify whether 243 or 3125 states should be generated. The EQ-5D-Y-3L has the same 243 states as the EQ-5D-3L. Matching is case-insensitive.
dim.names	A vector of dimension names to be used as names for output columns.
append_index	Boolean to indicate whether a column of 5-digit EQ-5D health state indexes should be added to output.

Value

A data.frame with 5 columns and 243 (-3L, -Y-3L) or 3125 (-5L) health states

Examples

```
make_all_EQ_states('3L')
```

make_dummies	<i>EQ_dummies</i>
--------------	-------------------

Description

Make a data.frame of all EQ-5D dummies relevant for e.g. regression modelling.

Usage

```
make_dummies(
  df,
  version = "5L",
  dim.names = c("mo", "sc", "ua", "pd", "ad"),
  drop_level_1 = TRUE,
  add_intercept = FALSE,
  incremental = FALSE,
  prepend = NULL,
  append = NULL,
  return_df = TRUE
)
```

Arguments

df	data.frame containing EQ-5D health states.
version	EQ-5D instrument version: "3L", "5L" or "Y3L". Matching is case-insensitive.
dim.names	A vector of dimension names to be used as names for output columns.
drop_level_1	If set to FALSE, dummies for level 1 will be included. Defaults to TRUE.
add_intercept	If set to TRUE, a column containing 1s will be appended. Defaults to FALSE.

incremental	If set to TRUE, incremental dummies will be produced (e.g. MO = 3 will give mo2 = 1, mo3 = 1). Defaults to FALSE.
prepend	Optional string to be prepended to column names.
append	Optional string to be appended to column names.
return_df	If set to TRUE, data.frame is returned, otherwise matrix. Defaults to TRUE.

Value

A data.frame of dummy variables, one row per row of df. Levels are read as the value functions read them: text and factors through their labels. A level the instrument does not have (fractional, 0, negative, non-finite or above the highest level) is not a level, so that dimension's dummies are NA, with a warning; a missing level gives NA dummies without one.

Examples

```
make_dummies(make_all_EQ_states('3L'), '3L')

make_dummies(df = make_all_EQ_states('3L'),
              version = '3L',
              incremental = TRUE,
              add_intercept = TRUE,
              prepend = "d_")
```

run_app

Launch the eq5dsuite Shiny Application

Description

Opens an interactive Shiny application for uploading, validating, analysing and exporting EQ-5D data.

Usage

```
run_app(online = NULL, ...)
```

Arguments

online	Whether to run in online mode, for a public deployment. NULL (default) takes it from the eq5dsuite.online option and then the EQ5DSUITE_ONLINE environment variable, both of which default to off. The limits it puts in force are listed in DEPLOY.md.
...	Additional arguments passed to runApp , such as port or launch.browser.

Details

The app has seven pages, in the order the work is done:

Home What the app does, what data it needs, and a button to load the bundled [example_data](#).

Data Upload a .csv, .xlsx or .rds file, set the instrument version, and select its variables ("Confirm variables"). Only the five EQ-5D dimensions are required; a timepoint, patient ID, group, EQ VAS score, age, sex and an existing EQ-5D value column are each optional and each unlock further analyses.

Validation Range, missingness and duplicate-ID checks, then "Proceed", which fixes the working dataset.

Calculate EQ-5D values Turn the responses into values and append them as a column. One Method selector covers every way the package has of doing it: a published value set directly, a crosswalk ([eqxw](#), [eqxwr](#)), or the NICE Decision Support Unit's UK mapping ([eqxw_UK](#), [eqxwr_UK](#)). The direction of a mapping follows the instrument version and is stated in the method's label and on the page: EQ-5D-3L responses give EQ-5D-5L values, and EQ-5D-5L responses give EQ-5D-3L values.

The DSU mapping depends on the respondent's age and sex as well as the health state, so choosing it asks for those columns. Where the age column holds bands rather than exact ages, the midpoint of each band is used and the page says so: the DSU's age bands begin at 35, 45, 55 and 65, which are exactly the midpoints of ten-year bands, so a band straddling a boundary is placed in the upper one. Supply exact ages where they are available.

Only the EQ-5D value analyses need a value column; the profile and EQ VAS analyses can be run without one. Several columns can be added and compared.

The working dataset can be downloaded from here, as CSV, XLSX or RDS, with every column added during the session. EQ-5D values are shown to three decimal places throughout the app, but written at full precision.

Analysis All the analysis functions of Devlin et al. (2020), in one place. Pick a component (EQ-5D profiles, EQ-5D values or EQ VAS), then an output, then Run – or open "See catalogue of available analyses", which lists, in one tab per component, only what the data can support: a plain description of each and an example of its output on the bundled example data, with "Select this". An output whose variables are not selected says what it needs rather than failing. The EQ-5D value analyses take a Utility column, offering every value column in the dataset; the chosen one is passed to the analysis as `name_utility` and nothing is recalculated.

Results and export Every analysis that has been run, numbered, each with controls to move it up or down or remove it, its own downloads (CSV for a table, PNG and PDF for a figure), and "View", which shows the table or figure and the `eq5dsuite` code that produced it. That order is the export order for all three exports: a Word report, one section per result with its title and its table or figure; an archive of CSV tables and PNG figures with a `manifest.csv`; and an R script that repeats the whole session. The Word report uses the template bundled with the package and needs Pandoc, which ships with RStudio.

The script loads the data, runs the same checks, calculates the same EQ-5D values and produces the same results in the same order. Apart from the analysis functions themselves it calls nothing from `eq5dsuite`: reading the file, renaming the columns, the checks, the age-band midpoints and the display formatting are all written out in the script, so it can be read, changed and run wherever `eq5dsuite` is installed, with no part of the app involved. It is built from a structured record of what was done, so every value is quoted safely.

Value

Called for its side effect of launching a Shiny application. Returns invisibly.

Running it on a server

`run_app(online = TRUE)`, or `EQ5DSUITE_ONLINE=true` in the environment the app starts in, puts the app in online mode. That limits what can be uploaded, keeps every file the app writes inside a folder of its own that is deleted when the session ends, ends idle sessions, sanitises errors, and tells the user what happens to their data. Locally – the default – none of it applies and the app behaves as it always has.

The limits and the wording are set with options or environment variables. `DEPLOY.md` in the installed package (`system.file("shiny", "DEPLOY.md", package = "eq5dsuite")`) covers what the server itself has to be told.

The app does not attach `eq5dsuite`, so it needs no `library()` call first; `run_app()` is enough.

References

Devlin N, Parkin D, Janssen B (2020). *Methods for Analyzing and Reporting EQ-5D Data*. Springer. doi:[10.1007/9783030476229](https://doi.org/10.1007/9783030476229)

See Also

[eq5d](#) and [eqxwr_UK](#) for the EQ-5D values the app calculates, and [eq5d_profile_level_summary](#) for the analyses it runs.

[eq5d](#) for value calculation, [eqxwr_UK](#) and [eqxw_UK](#) for the UK mapping.

Examples

```
## Not run:
# Locally.
eq5dsuite::run_app()

# On a server, with the defaults.
eq5dsuite::run_app(online = TRUE)

# On a server, with its own limits.
options(eq5dsuite.max_rows = 20000, eq5dsuite.idle_minutes = 15)
eq5dsuite::run_app(online = TRUE)

## End(Not run)
```

toEQ5Ddims

toEQ5Ddims

Description

Generate dimension vectors based on state index

Usage

```
toEQ5Ddims(x, dim.names = c("mo", "sc", "ua", "pd", "ad"))
```

Arguments

x A vector of 5-digit EQ-5D state indexes.

dim.names A vector of dimension names to be used as names for output columns.

Value

A data.frame with 5 integer columns, one for each EQ-5D dimension, with names from the dim.names argument; the same layout as the output of make_all_EQ_states(). A code that is not a valid 5-digit state gives NA in every column, with a warning.

Examples

```
toEQ5Ddims(c(12345, 54321, 12321))
```

toEQ5Dindex

Convert EQ-5D dimension scores to a five-digit profile index

Description

Convert EQ-5D dimension scores to a five-digit profile index

Usage

```
toEQ5Dindex(
  x,
  dim.names = c("mo", "sc", "ua", "pd", "ad"),
  na.rm = FALSE,
  quiet = FALSE
)
```

Arguments

<code>x</code>	A data.frame, matrix, or named numeric vector of EQ-5D dimension scores. Each dimension must be a whole number from 1 to 9; a value that is fractional, non-finite or outside that range is not encoded and returns NA for its row, with a warning. Fractional values are not rounded. Factor columns are read through their labels.
<code>dim.names</code>	Character vector of length 5 giving the dimension names, in the conventional MO, SC, UA, PD, AD order.
<code>na.rm</code>	Logical. If FALSE (default), any NA in a row produces NA in the output. If TRUE, NA dimensions are treated as 0 — use with care, as this silently changes the index value.
<code>quiet</code>	Logical. Suppress informational messages about assumed column / name order. Default FALSE so existing scripts see the same messages; set TRUE inside pipelines.

Value

An integer vector with one element per row (data.frame/matrix input) or a single integer (vector input). Works inside `dplyr::mutate()` without `rowwise()`.

Examples

```
# Named vector – scalar usage unchanged
toEQ5Dindex(c(mo=1, sc=2, ua=3, pd=1, ad=2))
```

update_value_sets	<i>Update eq5dsuite value sets</i>
-------------------	------------------------------------

Description

Connects to the eq5dsuite value sets repository and checks whether new value sets are available for any of the three EQ-5D instruments. Optionally drop or rename existing value sets before checking for updates.

Usage

```
update_value_sets(
  versions = c("3L", "5L", "Y3L"),
  ask = TRUE,
  drop = NULL,
  rename = NULL
)
```

Arguments

versions	Character vector. Which instruments to check: any of "3L", "5L" and "Y3L", in any case. The date of the last check, which decides when you are reminded to check again, moves only when all three are checked. Defaults to c("3L", "5L", "Y3L").
ask	Logical. Whether to ask for confirmation before installing, dropping, or renaming. Defaults to TRUE.
drop	Character vector of VS_codes to remove, in the format "VS_code:version" e.g. c("NL:3L", "US:5L"). Defaults to NULL (no removals).
rename	Named character vector of renames in the format c("old_VS_code:version" = "new_VS_code") e.g. c("NL:3L" = "NL_2006"). Defaults to NULL (no renames).

Details

This function requires an internet connection for checking and installing new value sets. Drop and rename operations work offline.

An instrument counts as checked only when its index was downloaded and is valid. One that could not be downloaded, or whose index is malformed, is reported as failed, with the reason, and is not reported as up to date. The date of the last check, which the reminder on loading the package uses, is moved on only when every requested instrument was checked and every new value set installed; otherwise the reminder stays due.

Value

Invisibly returns a list with elements checked (the instruments whose index was read and valid), failed (a named character vector: for each instrument that could not be checked, the reason), new, installed, install_failed (new value sets that could not be installed), migration_conflicts (renames not applied because both codes are installed; both value sets are kept), migrations_unrecorded (renames made, and kept, whose record could not be saved), dropped, and renamed. A list of renames that could not be downloaded, or is malformed, or renames that could not be recorded, appear in failed as "migrations"; that, or a conflict, leaves the check incomplete, so the date of the last check is not moved on. A rename made but not recorded is recorded by the next run, without being made again.

Examples

```
## Not run:
# Check for new value sets interactively
update_value_sets()

# Check specific instrument only
update_value_sets(versions = "5L")

# Drop a value set
update_value_sets(drop = "NL:3L")

# Rename a value set
```

```
update_value_sets(rename = c("NL:3L" = "NL_2006"))

# Drop and rename in one call
update_value_sets(
  drop   = "DE:3L",
  rename = c("NL:3L" = "NL_2006", "SI:3L" = "SI_TT0")
)

# Non-interactive update
update_value_sets(ask = FALSE)

## End(Not run)
```

Index

* datasets

- example_data, 63
- .EQxwrprob, 7
- .freqtab, 8
- .pstate3t5, 9
- .pstate5t3, 9

- eq5d, 4, 10, 21–25, 35–38, 40–42, 68
- eq5d3l, 4, 46
- eq5d5l, 4, 47
- eq5d_profile_better_dimensions_by_group_plot, 5, 11
- eq5d_profile_change_summary, 4, 12
- eq5d_profile_density_curve, 5, 14
- eq5d_profile_dimension_change_table, 4, 13, 15
- eq5d_profile_health_profile_grid, 5, 17
- eq5d_profile_level_summary, 4, 18, 68
- eq5d_profile_level_summary_by_group, 4, 19
- eq5d_profile_lfs_distribution, 5, 20
- eq5d_profile_lfs_mean_utility, 5, 21
- eq5d_profile_lfs_utility_plot, 5, 22
- eq5d_profile_lfs_utility_summary, 5, 23
- eq5d_profile_lss_utility_plot, 5, 24
- eq5d_profile_lss_utility_summary, 5, 25
- eq5d_profile_mixed_dimensions_by_group_plot, 5, 26
- eq5d_profile_pchc_by_group_plot, 4, 27
- eq5d_profile_pchc_table, 4, 13, 29
- eq5d_profile_pchc_with_no_problems_table, 4, 30
- eq5d_profile_shannon, 5, 31
- eq5d_profile_top_states, 4, 33
- eq5d_profile_worse_dimensions_by_group_plot, 5, 34
- eq5d_utility_by_group_plot, 5, 35
- eq5d_utility_change_by_group_plot, 5, 36
- eq5d_utility_distribution_plot, 5, 37
- eq5d_utility_norms_comparison, 5, 38
- eq5d_utility_over_time_plot, 5, 39
- eq5d_utility_summary, 5, 40
- eq5d_utility_summary_by_group, 5, 41
- eq5d_utility_vas_scatter_plot, 5, 42
- eq5d_vas_distribution_table, 5, 43
- eq5d_vas_grouped_distribution_plot, 5, 44
- eq5d_vas_histogram, 5, 44
- eq5d_vas_summary, 5, 45
- eq5dsuite (eq5dsuite-package), 3
- eq5dsuite-package, 3
- eq5dy3l, 4, 49
- eqvs_add, 4, 51, 55
- eqvs_display, 4, 17, 53
- eqvs_drop, 4, 54, 55
- eqvs_load, 4, 55
- eqxw, 4, 56, 58–60, 67
- eqxw_UK, 4, 57, 61, 63, 67, 68
- eqxwr, 4, 58, 59, 59, 61, 63, 67
- eqxwr_UK, 4, 58–60, 61, 67, 68
- example_data, 6, 63, 67

- make_all_EQ_indexes, 6, 64
- make_all_EQ_states, 6, 64
- make_dummies, 6, 65

- run_app, 6, 66
- runApp, 66

- toEQ5Ddims, 6, 69
- toEQ5Dindex, 6, 69

- update_value_sets, 4, 70