Package ‘eq5dsuite’

June 24, 2025

Type Package

Title Handling and Analysing EQ-5d Data

Version 1.0.1

Maintainer Kim Rand <krand@mathsinhealth.com>

Description The EQ-5D is a widely-used standarized 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> (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> (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 analyze 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>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Depends R (>= 3.5)

Imports dplyr, ggplot2, moments, RColorBrewer, rlang, scales, stringr, tidyr, rappdirs

NeedsCompilation no

Author Kim Rand [aut, cre] (ORCID: <https://orcid.org/0000-0001-7692-4099>),
Iryna Schlackow [aut] (ORCID: <https://orcid.org/0000-0002-4154-1431>),
Anabel Estévez-Carrillo [aut] (ORCID:
<https://orcid.org/0000-0001-8778-5055>)

Repository CRAN

Date/Publication 2025-06-24 13:30:18 UTC

Contents

`.add_utility` *Add utility values to a data frame*

Description

This function adds utility values to a data frame based on a specified version of EQ-5D and a country name.

Usage

```
.add_utility(df, eq5d_version, country)
```

Arguments

`df` A data frame containing the state data. The state must be included in the data frame as a character vector under the column named ‘state‘.

`eq5d_version` A character string specifying the version of EQ-5D, i.e. 3L or 5L.

`country` A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

A data frame with an additional column named ‘utility‘ containing the calculated utility values. If the input country name is not found in the country_codes dataset, a list of available codes is printed, and subsequentyl an error message is displayed and the function stops.

Examples

```
df <- data.frame(state = c("11111", "11123", "32541"))
.add_utility(df, "5L", "DK")
```

.check_uniqueness *Check the uniqueness of groups This function takes a data frame ‘df‘ and a vector of columns ‘group_by‘, and checks whether the combinations of values in the columns specified by ‘group_by‘ are unique. If the combinations are not unique, a warning message is printed.*

Description

Check the uniqueness of groups This function takes a data frame ‘df‘ and a vector of columns ‘group_by‘, and checks whether the combinations of values in the columns specified by ‘group_by‘ are unique. If the combinations are not unique, a warning message is printed.

Usage

```
.check_uniqueness(df, group_by)
```

Arguments

`df` A data frame.

`group_by` A character vector of column names in ‘df‘ that specify the groups to check for uniqueness.

Value

No return value, called for side effects: it will stop with an error if any group combinations are not unique.

Examples

```
df <- data.frame(id = c(1, 1, 1, 1, 2, 2),
                 fu = rep(c("baseline", "follow-up"), 3),
                 value = rnorm(6))
.check_uniqueness(df, c("id", "fu"))
```

.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
)
```

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_fu	Character string for the follow-up column. If NULL, no grouping is used, and the table reports for the total population.
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.
eq5d_version	Version of the EQ-5D instrument

Value

Summary data frame.

.gen_colours *Generate colours for PCHC figures*

Description

This internal function generates a vector of colours based on the specified base colour. Currently only green and orange colours are implemented. The wrapper is used in Figures 2.2-2.4.

Usage

```
.gen_colours(col, n)
```

Arguments

col	A character string specifying the base colour. Only "green" or "orange" is accepted.
n	A positive integer specifying the number of colours to generate.

Value

A vector of colours generated based on the specified base colour and number of colours.

Examples

```
# generate 10 colours for base colour "green"
.gen_colours("green", 10)
# generate 7 colours for base colour "orange"
.gen_colours("orange", 7)
```

`.getmode` *Get the mode of a vector.*

Description

This function calculates the mode of a numeric or character vector. If there are multiple modes, the first one is returned. The code is taken from an R help page.

Usage

```
.getmode(v)
```

Arguments

`v`	A numeric or character vector.

Value

The mode of ‘v‘.

Examples

```
.getmode(c(1, 2, 3, 3))
.getmode(c("a", "b", "b", "c"))
```

`.get_lfs` *Calculate the Level Frequency Score (LFS)*

Description

This function calculates the Level Frequency Score (LFS) for a given EQ-5D state and a specified version of EQ-5D. If at least one domain contains a missing entry, the whole LFS is set to be NA.

Usage

```
.get_lfs(s, eq5d_version)
```

Arguments

`s`	A character vector representing the EQ-5D state, e.g. 11123.
`eq5d_version`	A character string specifying the version of EQ-5D, i.e. 3L or 5L.

Value

A character vector representing the calculated LFS.

Examples

```
.get_lfs("333", "3L") # returns 003
.get_lfs("333", "5L") # returns 00300
.get_lfs("12345", "5L") # returns 11111
```

`.get_names` *Replace NULL names with default values*

Description

This function takes in a list of parameters, which would be column names of the input data frame, and checks if they are null. Any nulls are replaced with default values, and the updated list of parameters is returned.

Usage

```
.get_names(df = NULL, ...)
```

Arguments

df	a data frame; only used/supplied if levels_fu needs to be defined
...	a list of parameters consisting of any/all of ‘names_eq5d‘, ‘name_fu‘, ‘levels_fu‘, ‘eq5d_version‘, and ‘name_vas‘.

Value

a list of parameters with null entries replaced with default values.

Examples

```
.get_names(names_eq5d = c("mo", "sc", "ua", "pd", "ad"))
.get_names(names_eq5d = NULL, eq5d_version = NULL, name_vas = NULL)
```

`.modify_ggplot_theme` *Modify ggplot2 theme*

Description

Modify ggplot2 theme

Usage

```
.modify_ggplot_theme(p)
```

Arguments

`p` ggplot2 plot

Value

ggplot2 plot with modified theme

`.pchc` *Wrapper to determine Paretian Classification of Health Change*

Description

This internal function determines Paretian Classification of Health Change (PCHC) for each combination of the variables specified in the ‘group_by‘ argument. It is used in the code for table_2_4-table_2_5 and figure_2_1-figure_2_4. An EQ-5D health state is deemed to be ‘better‘ than another if it is better on at least one dimension and is no worse on any other dimension. An EQ-5D health state is deemed to be ‘worse‘ than another if it is worse in at least one dimension and is no better in any other dimension.

Usage

```
.pchc(df, level_fu_1, add_noprobs = FALSE)
```

Arguments

`df` A data frame with EQ-5D states and follow-up variable. The dataset is assumed to be have been ordered correctly.

`level_fu_1` Value of the first (i.e. earliest) follow-up. Would normally be defined as levels_fu[1].

`add_noprobs` Logical value indicating whether to include a separate classification for those without problems (default is FALSE)

Value

A data frame with PCHC value for each combination of the grouping variables. If ’add_noprobs’ is TRUE, a separate classification for those without problems is also included.

Examples

```
df <- data.frame(id = c(1, 1, 2, 2),
                 fu = c(1, 2, 1, 2),
                 mo = c(1, 1, 1, 1),
                 sc = c(1, 1, 5, 1),
                 ua = c(1, 1, 4, 3),
                 pd = c(1, 1, 1, 3),
                 ad = c(1, 1, 1, 1))
.pchc(df, level_fu_1 = 1, add_noprobs = TRUE)
```

.pchctab *.pchctab: Changes in health according to the PCHC (Paretian Classification of Health Change)*

Description

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

Usage

```
.pchctab(
  df,
  name_id,
  name_groupvar,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL,
  add_noprobs = FALSE
)
```

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
`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.
`add_noprobs`	if set to TRUE, level corresponding to "no problems" will be added to the table

Value

Summary data frame

Examples

```
.pchctab(df = example_data,
  name_id = "id",
  name_groupvar = "procedure",
  name_fu = "time",
  levels_fu = c('Pre-op', 'Post-op')
)
```

`.pchc_plot_by_dim` *Wrapper to generate Paretian Classification of Health Change plot by dimension*

Description

This internal function plots Paretian Classification of Health Change (PCHC) by dimension. The input is a data frame containing the information to plot, and the plot will contain bars representing the proportion of the total data that falls into each dimension, stacked by covariate. The wrapper is used in Figures 2.2-2.4.

Usage

```
.pchc_plot_by_dim(plot_data, ylab, title, cols, text_rotate = FALSE)
```

Arguments

`plot_data`	A data frame containing information to plot, with columns for name (the dimensions to plot), p (the proportion of the total data falling into each dimension), and fu (the follow-up).
`ylab`	The label for the y-axis.
`title`	The plot title.
`cols`	A vector of colors to use for the bars.
`text_rotate`	A logical indicating whether to rotate the text labels for the bars.

Value

A ggplot object containing the PCHC plot.

Examples

```
df <- data.frame(
  name = rep(c("Dim1", "Dim2"), each = 2),
  p = c(0.6, 0.4, 0.7, 0.3),
  groupvar = rep(c("Group A", "Group B"), 2)
)
colors <- c("Group A" = "#1b9e77", "Group B" = "#d95f02")
.pchc_plot_by_dim(df, ylab = "Proportion", title = "Example Plot", cols = colors)
```

.prep_eq5d *Data checking/preparation: EQ-5D variables*

Description

This function prepares a data frame for analysis by extracting, processing, and adding columns for EQ-5D variables, including state, LSS (Level Sum Score), LFS (Level Frequency Score) and utility.

Usage

```
.prep_eq5d(
  df,
  names,
  add_state = FALSE,
  add_lss = FALSE,
  add_lfs = FALSE,
  add_utility = FALSE,
  eq5d_version = NULL,
  country = NULL
)
```

Arguments

`df` a data frame of EQ-5D scores

`names` character vector of length 5 with names of EQ-5D variables in the data frame. The variables should be in an integer format.

`add_state` logical indicating whether the EQ-5D state should be added

`add_lss` logical indicating whether the LSS (Level Sum Score) should be added

`add_lfs` logical indicating whether the LFS (Level Frequency Score) should be added

`add_utility` logical indicating whether the utility should be added

`eq5d_version` character indicating the version of the EQ-5D questionnaire to use (either "3L" or "5L")

`country` character indicating the country to retrieve the quality of life score for

Value

a modified data frame with EQ-5D domain columns renamed to default names, and, if necessary, with added columns for state, LSS, LFS, and/or utility. If any of the checks fail (e.g. EQ-5D columns are not in an integer format), an error message is displayed and the function is stopping.

Examples

```
set.seed(1234)
df <- data.frame(mo = sample(1:5, 3), sc = sample(1:5, 3),
  ua = sample(1:5, 3), pd = sample(1:5, 3), ad = sample(1:5, 3))
```

```
.prep_eq5d(df, names = c("mo", "sc", "ua", "pd", "ad"),
  add_state = TRUE, add_lss = TRUE)
.prep_eq5d(df, names = c("mo", "sc", "ua", "pd", "ad"),
  add_state = TRUE, add_lss = TRUE, add_lfs = TRUE, add_utility = TRUE,
  eq5d_version = "5L", country = "ES")
```

.prep_fu *Data checking/preparation: follow-up variable*

Description

This function prepares the follow-up (FU) variable for analysis by giving it a default name (‘fu‘) and factorising

Usage

```
.prep_fu(df, name = NULL, levels = NULL)
```

Arguments

df	A data frame.
name	Column name in the data frame that contains follow-up information.
levels	Levels to factorise the FU variable into.

Value

A data frame with the follow-up variable renamed as "fu" and factorised.

Examples

```
df <- data.frame(id = c(1, 1, 2, 2),
  visit = c("baseline", "follow-up", "baseline", "follow-up"))
.prep_fu(df = df, name = "visit", levels = c("baseline", "follow-up"))
```

.prep_vas *Data checking/preparation: VAS variable*

Description

The function prepares the data for VAS (Visual Analogue Scale) analyses.

Usage

```
.prep_vas(df, name)
```

Arguments

df	A data frame.
name	Column name in the data frame that holds the VAS score. The column can only contain integers or NAs

Value

A modified data frame with the VAS score renamed to "vas". If any checks fail (e.g. column is not numeric), an error message is displayed and the function is stopping.

Examples

```
df <- data.frame(vas_score = c(20, 50, 80, NA, 100))
.prep_vas(df = df, name = "vas_score")
df <- data.frame(vas_score = c(20.5, 50, 80, NA, 100))
.prep_vas(df = df, name = "vas_score")
```

.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

`.summary_cts_by_fu` *Wrapper to summarise a continuous variable by follow-up (FU)*

Description

This function summarizes a continuous variable for each follow-up (FU) and calculates various statistics such as mean, standard deviation, median, mode, kurtosis, skewness, minimum, maximum, range, and number of observations. It also reports the total sample size and the number (and proportion) of missing values for each FU. The input ‘df‘ must contain an ordered FU variable and the continuous variable of interest. The name of the continuous variable must be specified using ‘name_v‘. The wrapper is used in Table 3.1 (for VAS) or Table 4.2 (for EQ-5D utility)

Usage

```
.summary_cts_by_fu(df, name_v)
```

Arguments

`df`	A data frame containing the FU and continuous variable of interest. The dataset must contain an ordered ‘fu‘ variable.
`name_v`	A character string with the name of the continuous variable in ‘df‘ to be summarised.

Value

Data frame with one row for each statistic and one column for each FU.

Examples

```
df <- data.frame(fu = c(1,1,2,2,3,3),
                 vas = c(7,8,9,NA,7,6))
.summary_cts_by_fu(df, name_v = "vas")
```

.summary_mean_ci *Wrapper to calculate summary mean with 95% confidence interval*

Description

This internal function calculates summary mean and 95% confidence interval of the utility variable, which can also be grouped. The function is used in Figures 4.2-4.4.

Usage

```
.summary_mean_ci(df, group_by)
```

Arguments

`df`	A data frame containing a ‘utility‘ column.
`group_by`	A character vector of column names to group by.

Value

A data frame with the mean, lower bound, and upper bound of the 95

Examples

```
df <- data.frame(group = c("A", "A", "B", "B"),
                 utility = c(0.5, 0.7, 0.8, 0.9))
.summary_mean_ci(df, group_by = "group")
```

.summary_table_2_1 *Wrapper for the repetitive code in function_table_2_1. Data frame summary*

Description

This internal function summarises a data frame by grouping it based on the variables specified in the 'group_by' argument and calculates the frequency of each group. The output is used in Table 2.1

Usage

```
.summary_table_2_1(df, group_by)
```

Arguments

df	A data frame
group_by	A character vector of variables in ‘df‘ to group by. Should contain 'eq5d' and 'fu'.

Value

A summarised data frame with groups defined by ‘eq5d‘ and ‘fu‘ variables, the count of observations in each group, and the frequency of each group.

Examples

```
set.seed(1234)
df <- data.frame(eq5d = rep(rnorm(5), 2),
                 fu = rep(c(1, 0, 1, 0, 1), 2))
.summary_table_2_1(df, c("eq5d", "fu"))
```

.summary_table_4_3 *Summary wrapper for Table 4.3*

Description

This internal function creates a summary of the data frame for Table 4.3. It groups the data by the variables specified in ‘group_by‘ and calculates various summary statistics.

Usage

```
.summary_table_4_3(df, group_by)
```

Arguments

df	A data frame.
group_by	A character vector of names of variables by which to group the data.

Value

A data frame with the summary statistics.

Examples

```
df <- data.frame(group = c("A", "A", "B", "B"),
                 utility = c(0.5, 0.7, 0.8, 0.9))
.summary_table_4_3(df, group_by = "group")
```

.summary_table_4_4 *Summary wrapper for Table 4.4*

Description

This internal function creates a summary of the data frame for Table 4.4. It groups the data by the variables specified in ‘group_by‘ and calculates various summary statistics.

Usage

```
.summary_table_4_4(df, group_by)
```

Arguments

df	A data frame.
group_by	A character vector of names of variables by which to group the data.

Value

A data frame with the summary statistics.

Examples

```
df <- data.frame(group = c("A", "A", "B", "B"),
                 utility = c(0.5, 0.7, 0.8, 0.9))
.summary_table_4_4(df, group_by = "group")
```

eq5d *eq5d*

Description

Get EQ-5D index values for the -3L, -5L, crosswalk (-3L value set applied to -5L health states), reverse crosswealk (-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	String vector indicating country names or ISO3166 Alpha 2 / 3 country codes.
version	String indicating which version to use. Options are '5L' (default), '3L', 'xw', 'xwr', and 'Y3L'.
dim.names	A vector of dimension names to identify dimension columns.

Value

A vector of values or data.frame with one column for each value set requested.

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')
```

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 EQ-5D-3L state dimensions.
`country`	String vector indicating country names or ISO3166 Alpha 2 / 3 country 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 vector of EQ-5D-3L values or data.frame with one column for each value set requested.

Examples

```
eq5d3l(c(11111, 12321, 32123, 33333), country = "US")
eq5d3l(make_all_EQ_states('3L'), c('DK', 'CA')) # Danish and Canada -3L value sets
```

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 EQ-5D-5L state dimensions.
`country`	String vector indicating country names or ISO3166 Alpha 2 / 3 country codes.
`dim.names`	A character vector specifying the names of the EQ-5D-5L dimensions. Default is c("mo", "sc", "ua", "pd", "ad").

Value

A vector of EQ-5D-5L values or data.frame with one column for each value set requested.

Examples

```
eq5d5l(c(11111, 12321, 32423, 55555), 'IT') # Italy -5L value set
eq5d5l(make_all_EQ_states('5L'), c('ES', 'DE')) # Spanish and german value sets
```

eq5dy3l *eq5dy3l*

Description

Get EQ-5D-Y3L index values from individual responses to the five dimensions of the EQ-5D-Y3L.

Usage

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

Arguments

`x`	A vector of 5-digit EQ-5D-Y3L state indexes or a matrix / data frame with columns for each dimension.
`country`	String vector indicating country names or ISO3166 Alpha 2 / 3 country codes.
`dim.names`	A character vector specifying the names of the EQ-5D-Y3L dimensions. Default is c("mo", "sc", "ua", "pd", "ad").

Value

A vector of EQ-5D-Y3L values or data.frame with one column for each value set requested.

Examples

```
# Slovenia -Y3L value set
eq5dy3l(x = c(11111, 12321, 33333), country = 'SI')
# Germany and Spain -Y3L value sets
eq5dy3l(make_all_EQ_states('3L'), c('ES', 'DE')) # Spanish and german value sets
```

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) EQ-5D health state vectors, and a column of corresponding utility values, with a suitable name.
`version`	Version of the EQ-5D instrument. Can take values 5L (default) or 3L.
`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 three-digit code for the value set. Must be different from any existing national value set code.
`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 package folder, 3: Save in another path.
`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.

Value

True/False, indicating success or error.

Examples

```
# make nonsense value set
new_df <- data.frame(state = make_all_EQ_indexes(), TEST = runif(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")
```

`eqvs_display` *eqvs_display*

Description

Display available value sets, which can also be used as (reverse) crosswalks.

Usage

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

Arguments

`version`	Version of the EQ-5D instrument. Can take values 5L (default) or 3L.
`return_df`	If set to TRUE, the function will return information on the names of the available value sets in a data.frame. Defaults to FALSE

Value

Default NULL, if return_df == TRUE, returns a data.frame with the displayed information.

Examples

```
# Display available value sets.
eqvs_display
```

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)
```

Arguments

country	Optional string. If NULL, a list of current user-defined value sets will be provided for selection. If set, and matching an existing user-defined value set, a prompt will be given as to whether the value set should be deleted.
version	Version of the EQ-5D instrument. Can take values 5L (default) or 3L.
saveOption	Integer indicating how the cache data should be saved. 1: Do not save (default), 2: Save in package folder, 3: Save in another path.
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.

Value

True/False, indicating success or error.

Examples

```
# make nonsense value set
new_df <- data.frame(state = make_all_EQ_indexes(), TEST = runif(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
eqvs_drop(country = 'FAN', saveOption = 1)
```

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.

Value

TRUE if loading is successful, FALSE otherwise.

eqxw *eqxw*

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	String vector indicating country names or ISO3166 Alpha 2 / 3 country codes.
dim.names	A vector of dimension names to identify dimension columns

Value

A vector of reverse crosswalk values or data.frame with one column per reverse crosswalk set requested.

Examples

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

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` String vector indicating country names or ISO3166 Alpha 2 / 3 country codes.

`dim.names` A vector of dimension names to identify dimension columns

Value

A vector of reverse crosswalk values or data.frame with one column per reverse crosswalk set requested.

Examples

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

eqxw_UK *eqxw_UK*

Description

Crosswalks EQ-5D-5L responses to EQ-5D-3L utilities using NICE's mapping.

Usage

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

Arguments

`x` A vector of 5-digit EQ-5D-5L states (domain scores) or a summary score.

`age` A numeric vector or column name (if ‘x‘ is a data frame). Can be either: (1) a numeric age between 18 and 100, which will be automatically grouped into NICE-defined age bands (18-35, 35-45, 45-55, 55-65, +65), or (2) a factor/character/numeric vector already representing the NICE age bands with values 1-5 indicating age bands (18-35, 35-45, 45-55, 55-65, +65).

`male` A numeric vector (1=male, 0=female) or column name indicating gender.

`dim.names` A vector of dimension names for EQ-5D states (default: c("mo", "sc", "ua", "pd", "ad")).

`bwidth` Numeric. Bandwidth for kernel smoothing when using summary scores.

Value

A vector or data frame with crosswalked EQ-5D-3L utilities.

Examples

```
eqxw_UK(c(11111, 12345, 32423, 55555), age = c(30, 40, 55, 70), male = c(1, 0, 1, 0))
```

`example_data` *example_data*

Description

A dataset containing patient-level data in a long format.

Usage

```
data(example_data)
```

Format

A data frame with 10000 rows and 14 variables:

`id` double Patient id

`time` character Follow-up (Pre-op / Post-op)

`mo` double EQ-5D-5L Mobility dimension

`sc` double EQ-5D-5L Self-care dimension

`ua` double EQ-5D-5L Usual activities dimension

`pd` double EQ-5D-5L Pain / discomfort dimension

`ad` double EQ-5D-5L Anxiety/depression dimension

`vas` double Value of the VAS scale measurememnt

`providercode` character Provider code

`procedure`	character Type of surgery
`year`	character Year of intervention
`ageband`	character Age in pre-defined ranges
`gender`	character Patient's gender (Female / Male)

`figure_1_2_1` *Figure 1.2.1: 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

Figure 1.2.1: 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

```
figure_1_2_1(
  df,
  name_id,
  name_groupvar,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL
)
```

Arguments

`df`	Data frame containing EQ-5D dimensions, a grouping variable, patient ID, and follow-up columns
`name_id`	Character string for the patient ID column
`name_groupvar`	Character string for the grouping column (e.g., procedure)
`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 of length 2 indicating the order of follow-up time points (e.g., c("Pre-op", "Post-op"))

Value

A list with two elements:

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

Examples

```
result <- figure_1_2_1(
  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
```

figure_1_2_2 *Figure 1.2.2: 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. The, it prodcuces a dimension-focused chart illustrating improvement percentages by dimension.*

Description

Figure 1.2.2: 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. The, it prodcuces a dimension-focused chart illustrating improvement percentages by dimension.

Usage

```
figure_1_2_2(
  df,
  name_id,
  name_groupvar,
  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)
`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 '.pchc_plot_by_dim()'

Examples

```
result <- figure_1_2_2(
  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
```

figure_1_2_3   *Figure 1.2.3: Percentage of Respondents Who Worsened in Each EQ-5D Dimension, by Group*

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

```
figure_1_2_3(
  df,
  name_id,
  name_groupvar,
  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)

`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()‘

Examples

```
result <- figure_1_2_3(
  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
```

`figure_1_2_4` *Figure 1.2.4: 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

```
figure_1_2_4(
  df,
  name_id,
  name_groupvar,
  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")

`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

Examples

```
result <- figure_1_2_4(
  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
```

figure_1_2_5 *Figure 1.2.5: 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

```
figure_1_2_5(
  df,
  names_eq5d,
  name_fu,
  levels_fu = NULL,
  name_id,
  eq5d_version,
  country
)
```

Arguments

`df`	A data frame containing EQ-5D columns, a grouping variable, 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
`country`	A character string representing the name of the country.

Value

A list with components:

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

Examples

```
tmp <- figure_1_2_5(
           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 = "UK"
       )
```

figure_1_3_1 *Figure 1.3.1: EQ-5D values plotted against LSS*

Description

Figure 1.3.1: EQ-5D values plotted against LSS

Usage

```
figure_1_3_1(df, names_eq5d = NULL, eq5d_version = NULL, country)
```

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
country	A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary plot and data used for plotting

Examples

```
df <- data.frame(make_all_EQ_states(version = "5L"))
tmp <- figure_1_3_1(
 df,
 names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
 eq5d_version = "5L",
 country = "US"
)
tmp$p
tmp$plot_data
```

figure_1_3_2 *Figure 1.3.2: EQ-5D values plotted against LFS*

Description

Figure 1.3.2: EQ-5D values plotted against LFS

Usage

```
figure_1_3_2(df, names_eq5d = NULL, eq5d_version = NULL, country)
```

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
`country`	A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary plot and data used for plotting

Examples

```
tmp <- figure_1_3_2(
 example_data,
 names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
 eq5d_version = "3L",
 country = "UK"
)
tmp$p
tmp$plot_data
```

`figure_1_4_1` *Figure 1.4.1: Generate a Health State Density Curve (HSDC) for EQ-5D Data*

Description

This function calculates and plots the Health State Density Curve (HSDC) for a given EQ-5D dataset. It concatenates dimension values to form health state profiles, filters out invalid states based on the specified EQ-5D version, then computes the cumulative distribution of profiles (profiles vs. observations). A diagonal reference line indicates a perfectly even distribution. The function also calculates the Health State Density Index (HSDI), representing how sharply the observed distribution deviates from the diagonal.

Usage

```
figure_1_4_1(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

Value

A list containing:

plot_data	A data frame with the cumulative distribution of profiles
p	A ggplot2 object showing the Health State Density Index

Examples

```
figure <- figure_1_4_1(
            df = example_data,
            names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
            eq5d_version = "3L"
          )
figure$plot_data
figure$p
```

figure_2_1 *Figure 2.1: EQ VAS scores*

Description

Figure 2.1: EQ VAS scores

Usage

```
figure_2_1(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 <- figure_2_1(example_data, name_vas = 'vas')
tmp$p
tmp$plot_data
```

figure_2_2 *Figure 2.2: Mid-point EQ VAS scores*

Description

Figure 2.2: Mid-point EQ VAS scores

Usage

```
figure_2_2(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 <- figure_2_2(example_data, name_vas = 'vas')
tmp$p
tmp$plot_data
```

figure_3_1 *Figure 3.1: EQ-5D values by timepoints: mean values and 95% confidence intervals*

Description

Figure 3.1: EQ-5D values by timepoints: mean values and 95% confidence intervals

Usage

```
figure_3_1(
  df,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL,
  eq5d_version = NULL,
  country
)
```

Arguments

`df` Data frame with the VAS columns

`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.

`eq5d_version` Version of the EQ-5D instrument

`country` A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary plot and data used for plotting

Examples

```
tmp <- figure_3_1(
 example_data,
 names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
 name_fu = "time",
 levels_fu = c('Pre-op', 'Post-op'),
 eq5d_version = "3L",
 country = "UK"
)
tmp$p
tmp$plot_data
```

---

figure_3_2 *Figure 3.2: Mean EQ-5D values and 95% confidence intervals: all vs by groupvar*

Description

Figure 3.2: Mean EQ-5D values and 95% confidence intervals: all vs by groupvar

Usage

```
figure_3_2(df, names_eq5d = NULL, name_groupvar, eq5d_version = NULL, country)
```

Arguments

`df` Data frame with the EQ-5D and grouping columns

`names_eq5d` Character vector of column names for the EQ-5D dimensions

`name_groupvar` Character string for the grouping column

`eq5d_version` Version of the EQ-5D instrument

`country` A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary plot and data used for plotting

Examples

```
tmp <- figure_3_2(
 example_data,
 names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
 name_groupvar = "procedure",
 eq5d_version = "3L",
 country = "UK"
)
tmp$p
tmp$plot_data
```

---

figure_3_3 *Figure 3.3: EQ-5D values: smoothed lines and confidence intervals by groupvar*

Description

Figure 3.3: EQ-5D values: smoothed lines and confidence intervals by groupvar

Usage

```
figure_3_3(
  df,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL,
  name_groupvar,
  eq5d_version = NULL,
  country
)
```

Arguments

df	Data frame with the EQ-5D, follow-up and grouping columns
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.
name_groupvar	Character string for the grouping column
eq5d_version	Version of the EQ-5D instrument
country	A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary plot and data used for plotting

Examples

```
tmp <- figure_3_3(
 example_data,
 names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
 name_fu = "time",
 levels_fu = c('Pre-op', 'Post-op'),
 name_groupvar = "procedure",
 eq5d_version = "3L",
 country = "UK"
)
tmp$p
tmp$plot_data
```

figure_3_4 *Figure 3.4: EQ-5D values: smoothed lines and confidence intervals by groupvar*

Description

Figure 3.4: EQ-5D values: smoothed lines and confidence intervals by groupvar

Usage

```
figure_3_4(df, names_eq5d = NULL, eq5d_version = NULL, country)
```

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
country	A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary plot and data used for plotting

Examples

```
tmp <- figure_3_4(
 example_data,
 names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
 eq5d_version = "3L",
 country = "UK"
)
tmp$p
tmp$plot_data
```

`figure_3_5` *Figure 3.5: EQ-5D values: smoothed lines and confidence intervals by groupvar*

Description

Figure 3.5: EQ-5D values: smoothed lines and confidence intervals by groupvar

Usage

```
figure_3_5(
  df,
  names_eq5d = NULL,
  name_vas = NULL,
  eq5d_version = NULL,
  country
)
```

Arguments

`df` Data frame with the EQ-5D columns

`names_eq5d` Character vector of column names for the EQ-5D dimensions

`name_vas` Character string for the VAS column

`eq5d_version` Version of the EQ-5D instrument

`country` A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary plot and data used for plotting

Examples

```
tmp <- figure_3_5(
   example_data,
   names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
   name_vas = "vas",
   eq5d_version = "3L",
   country = "UK"
 )
tmp$p
tmp$plot_data
```

make_all_EQ_indexes *make_all_EQ_indexes*

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	Either "3L" or "5L", to signify whether 243 or 3125 states should be generated
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) or 3125 (-5L) EQ-5D health states

Examples

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

make_all_EQ_states *make_all_EQ_states*

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`	Either "3L" or "5L", to signify whether 243 or 3125 states should be generated
`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) or 3125 (-5L) health states

Examples

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

make_dummies *EQ_dummies*

Description

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

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` Either "3L" or "5L", to signify EQ-5D instrument version

`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

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_”)
```

table_1_1_1 *Table 1.1.1: Frequency of levels by dimensions, cross-sectional*

Description

Table 1.1.1: Frequency of levels by dimensions, cross-sectional

Usage

```
table_1_1_1(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

```
table_1_1_1(
 df = example_data[example_data$time == "Pre-op",],
 names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
 eq5d_version = "3L"
)
```

---

table_1_1_2 *Table 1.1.2: Frequency of levels by dimensions, separated by category*

Description

Table 1.1.2: Frequency of levels by dimensions, separated by category

Usage

```
table_1_1_2(
  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

```
table_1_1_2(
 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"
)
```

table_1_1_3 *Table 1.1.3: Prevalence of the 10 most frequently observed self-reported health states*

Description

Table 1.1.3: Prevalence of the 10 most frequently observed self-reported health states

Usage

```
table_1_1_3(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

```
table_1_1_3(
  df = example_data[example_data$time == "Pre-op",],
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L",
  n = 10
)
```

`table_1_2_1` *Table 1.2.1: Frequency of levels by dimensions, by follow-up*

Description

Table 1.2.1: Frequency of levels by dimensions, by follow-up

Usage

```
table_1_2_1(
  df,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = 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_fu` 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.

`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.

`eq5d_version` Version of the EQ-5D instrument

Value

Summary data frame.

Examples

```
table_1_2_1(
  df = example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op" , "Post-op"),
  eq5d_version = "3L"
)
```

table_1_2_2 *Table 1.2.2: Changes in health according to the PCHC (Paretian Classification of Health Change)*

Description

Table 1.2.2: Changes in health according to the PCHC (Paretian Classification of Health Change)

Usage

```
table_1_2_2(
  df,
  name_id,
  name_groupvar,
  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
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

Examples

```
table_1_2_2(
  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")
)
```

table_1_2_3 *Table 1.2.3: Changes in health according to the PCHC, taking account of those with no problems*

Description

Table 1.2.3: Changes in health according to the PCHC, taking account of those with no problems

Usage

```
table_1_2_3(
  df,
  name_id,
  name_groupvar,
  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
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

Examples

```
table_1_2_3(
  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")
)
```

table_1_2_4 *Table 1.2.4: Changes in levels in each dimension, percentages of total and of type of change*

Description

Table 1.2.4: Changes in levels in each dimension, percentages of total and of type of change

Usage

```
table_1_2_4(df, name_id, names_eq5d = NULL, name_fu = NULL, levels_fu = NULL)
```

Arguments

df	Data frame with the EQ-5D, id and follow-up columns
name_id	Character string for the patient id column
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

Examples

```
table_1_2_4(
  df = example_data,
  name_id = "id",
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c("Pre-op" , "Post-op")
)
```

table_1_3_1 *Table 1.3.1: Summary statistics for the EQ-5D values by all the different LSSs (Level Sum Scores)*

Description

Table 1.3.1: Summary statistics for the EQ-5D values by all the different LSSs (Level Sum Scores)

Usage

```
table_1_3_1(df, names_eq5d = NULL, eq5d_version = NULL, country)
```

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

`country` A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary data frame

Examples

```
df <- data.frame(make_all_EQ_states(version = "5L"))
table_1_3_1(
  df,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L",
  country = "US"
)
```

`table_1_3_2` *Table 1.3.2: Distribution of the EQ-5D states by LFS (Level Frequency Score)*

Description

Table 1.3.2: Distribution of the EQ-5D states by LFS (Level Frequency Score)

Usage

```
table_1_3_2(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

```
table_1_3_2(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L"
)
```

`table_1_3_3` *Table 1.3.3: Number of observations in the LFS (Level Frequency Score) according to the EQ-5D values*

Description

Table 1.3.3: Number of observations in the LFS (Level Frequency Score) according to the EQ-5D values

Usage

```
table_1_3_3(df, names_eq5d = NULL, eq5d_version = NULL, country)
```

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
`country`	A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary data frame

Examples

```
table_1_3_3(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L",
  country = "UK"
)
```

`table_1_3_4` *Table 1.3.4: Summary statistics of EQ-5D values by LFS (Level Frequency Score)*

Description

Table 1.3.4: Summary statistics of EQ-5D values by LFS (Level Frequency Score)

Usage

```
table_1_3_4(df, names_eq5d = NULL, eq5d_version = NULL, country)
```

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
`country`	A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary data frame

Examples

```
table_1_3_4(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  eq5d_version = "3L",
  country = "UK"
)
```

`table_2_1` *Table 2.1: EQ VAS Score by timepoints*

Description

Table 2.1: EQ VAS Score by timepoints

Usage

```
table_2_1(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.

Value

Summary data frame

Examples

```
table_2_1(
  example_data,
  name_vas = 'vas',
  name_fu = 'time',
  levels_fu = c('Pre-op', 'Post-op')
)
```

`table_2_2` *Table 2.2: EQ VAS Scores frequency of mid-points*

Description

Table 2.2: EQ VAS Scores frequency of mid-points

Usage

```
table_2_2(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

```
table_2_2(
  example_data,
  name_vas = 'vas',
  add_na_total =  TRUE
  )
```

`table_3_1` *Table 3.1: EQ-5D values: by timepoints*

Description

Table 3.1: EQ-5D values: by timepoints

Usage

```
table_3_1(
  df,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL,
  eq5d_version = NULL,
  country
)
```

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_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.

`eq5d_version` Version of the EQ-5D instrument

`country` A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

Value

Summary data frame

Examples

```
table_3_1(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c('Pre-op', 'Post-op'),
  eq5d_version = "3L",
  country = "UK"
)
```

table_3_2 *Table 3.2 EQ-5D values: by groupvar*

Description

Table 3.2 EQ-5D values: by groupvar

Usage

```
table_3_2(df, names_eq5d = NULL, name_groupvar, eq5d_version = NULL, country)
```

Arguments

df	Data frame with the EQ-5D, follow-up and grouping columns
names_eq5d	Character vector of column names for the EQ-5D dimensions
name_groupvar	Character string for the grouping column
eq5d_version	Version of the EQ-5D instrument
country	A character string representing the name of the country.

Value

Summary data frame

Examples

```
table_3_2(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_groupvar = "procedure",
  eq5d_version = "3L",
  country = "UK"
)
```

table_3_3 *Table 3.3 EQ-5D values: by age and groupvar*

Description

Table 3.3 EQ-5D values: by age and groupvar

Usage

```
table_3_3(
  df,
  names_eq5d = NULL,
  name_fu = NULL,
  levels_fu = NULL,
  name_groupvar,
  name_age,
  eq5d_version = NULL,
  country
)
```

Arguments

`df` Data frame with the EQ-5D, age, follow-up and grouping columns

`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.

`name_groupvar` Character string for the grouping column

`name_age` Character string for the age column

`eq5d_version` Version of the EQ-5D instrument

`country` A character string representing the name of the country. This could be in a 2-letter format, full name or short name, as specified in the country_codes datasets.

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"),]
table_3_3(
  example_data,
  names_eq5d = c("mo", "sc", "ua", "pd", "ad"),
  name_fu = "time",
  levels_fu = c('Pre-op', 'Post-op'),
  name_groupvar = "gender",
  name_age = "ageband",
  eq5d_version = "3L",
  country = "UK"
)
```

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 columns, one for each EQ-5D dimension, with names from dim.names argument.

Examples

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

toEQ5Dindex *toEQ5DIndex*

Description

Generate EQ-5D state vector from data.frame/matrix or named vector with dimensions.

Usage

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

Arguments

x	A data.frame, matrix, or vector containing dimension levels. Should have column names corresponding to the dim.names argument.
dim.names	A vector of dimension names in data.frame/matrix/vector x

Value

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

Examples

```
toEQ5Dindex(c(1,2,3,4,5))

example_data <- as.data.frame(matrix(data = c(1, 2, 3, 4, 5,
                                              5, 4, 3, 2, 1,
                                              3, 2, 1, 2, 3),
                                     ncol = 5,
                                     byrow = TRUE,
                                 dimnames = list(NULL, c("mo", "sc", "ua", "pd", "ad"))))
example_data$irrelevant <- c(6,5,3)
toEQ5Dindex(example_data)
```

Index