

Package: valueSetCompare (via r-universe)

November 1, 2024

Type Package

Title Comparing HRQoL Instrument Value Sets

Version 1.0.0

Maintainer Kim Rand <krand@mathsinhealth.com>

Description The number of countries with multiple Health Related Quality of Life (HRQL) value sets is growing, and this trend is expected to continue. Each instrument and value set characterizes and values health differently. Identical health states can yield different utility values when valued using different value sets. The 'valueSetCompare' package facilitates comparisons of HRQoL value sets, enabling both theoretical and empirical comparisons. For empirical comparisons, it employs a novel simulation-based method by Jiang et al. (2022) <[doi:10.1186/s12955-022-02031-8](https://doi.org/10.1186/s12955-022-02031-8)>, allowing users to investigate the responsiveness of HRQoL instruments across the entire health spectrum using cross-sectional data with external health anchors.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Depends R (>= 3.5)

Imports dplyr, ggplot2, rlang, eq5dsuite

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository <https://mathsinhealth.r-universe.dev>

RemoteUrl <https://github.com/mathsinhealth/valuesetcompare>

RemoteRef HEAD

RemoteSha 94e84debdab23b355e90ce3f211c7bc093963178

Contents

<code>.add_EQ5D_utilities</code>	2
<code>.cut_variable</code>	3
<code>.makeWeightsGradient</code>	3
<code>.makeWeightsMixed</code>	4
<code>.makeWeightsTriangular</code>	4
<code>cdta</code>	5
<code>compute_F_statistics</code>	6
<code>compute_utility_stats</code>	8
<code>density_plot_empirical</code>	9
<code>density_plot_theoretical</code>	10
<code>example_data</code>	12
<code>severity_ribbon_plot</code>	13
<code>single_transition_plots</code>	15
Index	17

<code>.add_EQ5D_utilities</code>	<i>.add_EQ5D_utilities</i>
----------------------------------	----------------------------

Description

This function calculates utilities for the provided EQ5D version and value sets.

Usage

```
.add_EQ5D_utilities(
  df,
  value_sets,
  version,
  dim.names = c("mo", "sc", "ua", "pd", "ad"),
  colnames = NULL
)
```

Arguments

<code>df</code>	A data frame to which the utilities will be added.
<code>value_sets</code>	A character vector specifying the country value sets for the given EQ5D version.
<code>version</code>	A character string specifying the EQ5D version. Valid versions are "5L", "3L", "XW", and "XWR".
<code>dim.names</code>	A vector of dimension names to identify dimension columns in the <code>df</code> data frame
<code>colnames</code>	(Optional) A character vector specifying the column names for the added utilities.

Value

The `df` data frame with the added utility columns.

.cut_variable *.cut_variable*

Description

This function cuts a numeric variable into intervals based on the provided breaks.

Usage

```
.cut_variable(variable, breaks)
```

Arguments

variable A numeric vector that you want to cut into intervals.
breaks A numeric vector specifying the breakpoints for cutting the variable.

Value

A factor vector representing the intervals into which the variable has been cut.

.makeWeightsGradient *.makeWeightsGradient*

Description

A function to calculate weights based on a gradient approach.

Usage

```
.makeWeightsGradient(x, pointval, rng)
```

Arguments

x A numeric vector of input values.
pointval A numeric point value within the range of *x*.
rng A numeric vector indicating the range of *x*.

Value

A numeric vector of calculated weights.

.makeWeightsMixed *.makeWeightsMixed*

Description

A function to calculate weights based on a mixed approach.

Usage

```
.makeWeightsMixed(x, pointval, rng)
```

Arguments

x	A numeric vector of input values.
pointval	A numeric point value within the range of x.
rng	A numeric vector indicating the range of x.

Value

A numeric vector of calculated weights.

.makeWeightsTriangular
.makeWeightsTriangular

Description

A function to calculate weights based on a triangular approach.

Usage

```
.makeWeightsTriangular(x, pointval, rng)
```

Arguments

x	A numeric vector of input values.
pointval	A numeric point value within the range of x.
rng	A numeric vector indicating the range of x.

Value

A numeric vector of calculated weights.

 cdta

 cdta

Description

A dataset with 3749 participants from six countries containing EQ-5D-3L and EQ-5D-5L data.

Usage

`data(cdta)`

Format

A data frame with 3749 rows and 20 variables:

`conditiongroups` integer An integer variable representing different health conditions. Values range from 1 to 10, where each value corresponds to the following conditions, in order:

1. COPD/Asthma
2. Diabetes
3. Liver disease
4. RA/A
5. CVD
6. Stroke
7. Depression
8. Personality Disorder
9. Other
10. Students

`studyID` integer Identification of the original study

`profile3L` integer EQ-5D-3L Health state

`profile5L` integer EQ-5D-5L Health state

`respID` integer Participant's ID in the original study

`condition` integer An integer variable representing different health conditions before grouping

`age` integer Participant's age in years

`gender` integer Participant's gender (Female / Male)

`education` integer Participant's educational level

`mobility` integer EQ-5D-3L Mobility dimension

`selfcare` integer EQ-5D-3L Self-Care dimension

`activity` integer EQ-5D-3L Usual activities dimension

`pain` integer EQ-5D-3L Pain/discomfort dimension

`anxiety` integer EQ-5D-3L Anxiety/depression dimension

`VAS` integer Value of the VAS scale measurement

mobility5L integer EQ-5D-5L Mobility dimension
 selfcare5L integer EQ-5D-5L Self-Care dimension
 activity5L integer EQ-5D-5L Usual activities dimension
 pain5L integer EQ-5D-5L Pain/discomfort dimension
 anxiety5L integer EQ-5D-5L Anxiety/depression dimension

compute_F_statistics *compute_F_statistics*

Description

This function computes F-statistics for specified utility columns in a data frame.

Usage

```
compute_F_statistics(  
  df,  
  utility_columns,  
  utility_combinations = NULL,  
  weight_column = "VAS",  
  weight_range = c(0:100),  
  sample_size = nrow(df),  
  number_of_samples = 1000,  
  variant_fun = .cut_variable,  
  breaks = c(0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100),  
  graph_title = "",  
  x_axis_title = "",  
  y_axis_title = "",  
  y_min_value = NULL,  
  y_max_value = NULL,  
  F_stats_groups = NULL  
)
```

Arguments

df A data frame containing the utility and weight columns.

utility_columns A character vector specifying the names of utility columns.

utility_combinations A matrix with two rows indicating the utility columns combinations. Default is all possible combinations of the elements of utility_columns taken 2 at a time.

weight_column A character string specifying the name of the weight column. Default is "VAS".

weight_range A numeric vector specifying the range of weights. Default is c(0:100).

sample_size An integer specifying the sample size for bootstrapping. Default is 1000.

number_of_samples	An integer specifying the number of bootstrap samples. Default is 1000.
variant_fun	A function to be applied for factorizing the weight column. Default is .cut_variable.
breaks	A numeric vector specifying the breaks for the 'cut' method. Default is c(0,10,20,30,40,50,60,70,80,90,100).
graph_title	A character string specifying the title of the plot. Default is an empty string.
x_axis_title	A character string specifying the title for the x-axis. Default is an empty string.
y_axis_title	A character string specifying the title for the y-axis. Default is an empty string.
y_min_value	A numeric value specifying the minimum value for the y-axis. Default is NULL.
y_max_value	A numeric value specifying the maximum value for the y-axis. Default is NULL.
F_stats_groups	An optional data frame of pre-computed group-based F-statistics. Default is NULL.

Value

A list containing two elements: 'df' which is a data frame of weighted statistics, and 'plot' which is the ggplot object representing the ribbon plot.

Examples

```
# Define dimension names for EQ-5D-3L and EQ-5D-5L
dim.names.3L <- c("mobility", "selfcare", "activity", "pain", "anxiety")
dim.names.5L <- c("mobility5L", "selfcare5L", "activity5L", "pain5L", "anxiety5L")
# Compute EQ-5D scores using the eq5dsuite package
cdta$EQ5D3L <- eq5dsuite::eq5d3l(x = cdta,
                                country = "US",
                                dim.names = dim.names.3L)
cdta$EQ5D5L <- eq5dsuite::eq5d5l(x = cdta,
                                country = "US",
                                dim.names = dim.names.5L)
cdta$EQXW <- eq5dsuite::eqxw(x = cdta,
                             country = "US",
                             dim.names = dim.names.5L)
# Define combinations of utility columns for F-statistics calculation
utility_combinations <- matrix(c("EQ5D5L", "EQ5D3L", "EQ5D5L", "EQXW"), nrow = 2)
# Compute F-statistics for the utility columns
result <- compute_F_statistics(df = cdta,
                              utility_columns = c("EQ5D3L", "EQ5D5L", "EQXW"),
                              utility_combinations = utility_combinations)

# Plot the results
print(result$plot)
```

```
compute_utility_stats compute_utility_stats
```

Description

Compute utility statistics for given EQ5D versions and value sets.

Usage

```
compute_utility_stats(
  value_sets_3L = NULL,
  value_sets_5L = NULL,
  value_sets_XW = NULL,
  value_sets_XWR = NULL,
  value_sets_others = NULL,
  format_results = FALSE
)
```

Arguments

`value_sets_3L` (Optional) A character vector specifying the country value sets for EQ5D-3L version.

`value_sets_5L` (Optional) A character vector specifying the country value sets for EQ5D-5L version.

`value_sets_XW` (Optional) A character vector specifying the country value sets for EQ5D-XW version.

`value_sets_XWR` (Optional) A character vector specifying the country value sets for EQ5D-XWR version.

`value_sets_others` (Optional) A list of lists specifying the inputs for other instruments. Each list within the main list should be named and contain a data frame ('df'), a column specifying the health states ('stateColumn'), and a column specifying the utility values ('utilityColumn').

`format_results` (Optional) A logical indicating whether the result should be formatted. Default is FALSE.

Value

A data frame containing utility statistics for each provided value set.

Note

Mean single level transitions are calculated only for EQ-5D.

Examples

```
compute_utility_stats(value_sets_3L = "ES", value_sets_5L = "ES")
value_set_other <- list(instrument1 = list(df = data.frame(state=c(1, 2, 3), utility = c(-1, 0, 1)),
  stateColumn = "state",
  utilityColumn = "utility"))
compute_utility_stats(value_sets_3L = "AR", value_sets_others = value_set_other)
```

```
density_plot_empirical
  density_plot_empirical
```

Description

This function creates a smoothed kernel density plot of the empirical distribution of utility values in a given data frame.

Usage

```
density_plot_empirical(
  df,
  utility_columns,
  graph_title = "",
  x_axis_title = "Index Value",
  x_min_value = NULL,
  x_max_value = NULL,
  y_axis_title = "Density",
  y_min_value = NULL,
  y_max_value = NULL,
  legend_name = "",
  color_palette = NULL,
  line_types = NULL
)
```

Arguments

<code>df</code>	A data frame containing the utility and weight columns.
<code>utility_columns</code>	A character vector specifying the names of utility columns.
<code>graph_title</code>	A character string specifying the title of the graph. Default is an empty string.
<code>x_axis_title</code>	A character string specifying the title of the x-axis. Default is "Index Value".
<code>x_min_value</code>	A numeric specifying the minimum value for the x-axis. Default is NULL.
<code>x_max_value</code>	A numeric specifying the maximum value for the x-axis. Default is NULL.
<code>y_axis_title</code>	A character string specifying the title of the y-axis. Default is "Density".
<code>y_min_value</code>	A numeric specifying the minimum value for the y-axis. Default is NULL.
<code>y_max_value</code>	A numeric specifying the maximum value for the y-axis. Default is NULL.

legend_name	A character string specifying the name of the legend. Default is "".
color_palette	A character vector specifying the colors for the density lines. Default is a pre-defined color palette.
line_types	A character vector specifying the line types for the density lines. Default is solid.

Value

A ggplot object visualizing the density of utilities for the specified EQ5D versions and other instruments value sets.

Examples

```
dim.names.3L <- c("mobility", "selfcare", "activity", "pain", "anxiety")
cdta$EQ5D3L <- eq5dsuite::eq5d3l(x = cdta,
                               country = "US",
                               dim.names = dim.names.3L)
dim.names.5L <- c("mobility5L", "selfcare5L", "activity5L", "pain5L", "anxiety5L")
cdta$EQ5D5L <- eq5dsuite::eq5d5l(x = cdta,
                               country = "US",
                               dim.names = dim.names.5L)
cdta$EQXW <- eq5dsuite::eqxw(x = cdta,
                             country = "US",
                             dim.names = dim.names.5L)
density_plot_empirical(df = cdta, utility_columns = c("EQ5D3L", "EQ5D5L", "EQXW"))
```

density_plot_theoretical

density_plot_theoretical

Description

This function creates a smoothed kernel density plot of utilities for different EQ5D versions and specified value sets.

Usage

```
density_plot_theoretical(
  value_sets_3L = NULL,
  value_sets_5L = NULL,
  value_sets_XW = NULL,
  value_sets_XWR = NULL,
  value_sets_others = NULL,
  graph_title = "",
  x_axis_title = "Index Value",
  x_min_value = NULL,
  x_max_value = NULL,
  y_axis_title = "Density",
```

```

    y_min_value = NULL,
    y_max_value = NULL,
    legend_name = "",
    color_palette = NULL,
    line_types = NULL
  )

```

Arguments

`value_sets_3L` A character vector specifying the country value sets for the EQ5D-3L version.

`value_sets_5L` A character vector specifying the country value sets for the EQ5D-5L version.

`value_sets_XW` A character vector specifying the country value sets for the EQ5D-XW version.

`value_sets_XWR` A character vector specifying the country value sets for the EQ5D-XWR version.

`value_sets_others` A list of lists specifying the inputs for other instruments. Each list within the main list should be named and contain a data frame ('df'), a column specifying the health states ('stateColumn'), and a column specifying the utility values ('utilityColumn').

`graph_title` A character string specifying the title of the graph. Default is an empty string.

`x_axis_title` A character string specifying the title of the x-axis. Default is "Index Value".

`x_min_value` A numeric specifying the minimum value for the x-axis. Default is NULL.

`x_max_value` A numeric specifying the maximum value for the x-axis. Default is NULL.

`y_axis_title` A character string specifying the title of the y-axis. Default is "Density".

`y_min_value` A numeric specifying the minimum value for the y-axis. Default is NULL.

`y_max_value` A numeric specifying the maximum value for the y-axis. Default is NULL.

`legend_name` A character string specifying the name of the legend. Default is "".

`color_palette` A character vector specifying the colors for the density lines. Default is a pre-defined color palette.

`line_types` A character vector specifying the line types for the density lines. Default is solid.

Value

A ggplot object visualizing the density of utilities for the specified EQ5D versions and other instruments value sets.

Examples

```

density_plot_theoretical(value_sets_3L = "NL", value_sets_5L = "NL")
instrument <- data.frame(HS=c(123, 456, 789), val = c(-0.3, 0.1, 0.75))
value_set_other <- list(test_instrument = list(df = instrument,
      stateColumn = "HS",
      utilityColumn = "val"))
density_plot_theoretical(value_sets_3L = "HU", value_sets_others = value_set_other)

```

 example_data

example_data

Description

A dataset with 3749 participants from six countries containing EQ-5D-3L and EQ-5D-5L data.

Usage

```
data(example_data)
```

Format

A data frame with 3749 rows and 20 variables:

`conditiongroups` integer An integer variable representing different health conditions. Values range from 1 to 10, where each value corresponds to the following conditions, in order:

1. 1: COPD/Asthma
2. 2: Diabetes
3. 3: Liver disease
4. 4: RA/A
5. 5: CVD
6. 6: Stroke
7. 7: Depression
8. 8: Personality Disorder
9. 9: Other
10. 10: Students

`studyID` integer Identification of the original study

`profile3L` integer EQ-5D-3L Health state

`profile5L` integer EQ-5D-5L Health state

`respID` integer Participant's ID in the original study

`condition` integer An integer variable representing different health conditions before grouping

`age` integer Participant's age in years

`gender` integer Participant's gender (Female / Male)

`education` integer Participant's educational level

`mobility` integer EQ-5D-3L Mobility dimension

`selfcare` integer EQ-5D-3L Self-Care dimension

`activity` integer EQ-5D-3L Usual activities dimension

`pain` integer EQ-5D-3L Pain/discomfort dimension

`anxiety` integer EQ-5D-3L Anxiety/depression dimension

`VAS` integer Value of the VAS scale measurement

mobility5L integer EQ-5D-5L Mobility dimension
 selfcare5L integer EQ-5D-5L Self-Care dimension
 activity5L integer EQ-5D-5L Usual activities dimension
 pain5L integer EQ-5D-5L Pain/discomfort dimension
 anxiety5L integer EQ-5D-5L Anxiety/depression dimension

severity_ribbon_plot *severity_ribbon_plot*

Description

This function generates a ribbon plot for given utility columns, based on weighted statistics.

Usage

```

severity_ribbon_plot(
  df,
  utility_columns,
  weight_column = "VAS",
  weight_range = c(0:100),
  weight_values = NULL,
  weight_function = .makeWeightsMixed,
  sample_size = 1000,
  number_of_samples = 1000,
  probability_levels = c(min = 0, `2.5%` = 0.025, `25%` = 0.25, median = 0.5, `75%` =
    0.75, `97.5%` = 0.975, max = 1),
  graph_title = "",
  x_axis_title = "",
  y_axis_title = "",
  legend_name = "Type",
  legend_labels = NULL,
  y_axis_limits = c(0.15, 0.95),
  y_min_value = "2.5%",
  y_max_value = "97.5%",
  alpha_1 = 0.15,
  alpha_2 = 0.05,
  linetype_1 = 1,
  linetype_2 = 2,
  interpretation_quartiles = c(0, 0.25, 0.5, 0.75, 1),
  elevation_threshold = 0.05,
  slope_threshold = 0.02,
  color_palette = NULL,
  weighted_statistics = NULL
)

```

Arguments

<code>df</code>	A data frame containing the utility and weight columns.
<code>utility_columns</code>	A character vector specifying the utility columns for which the ribbon plot will be generated.
<code>weight_column</code>	A string specifying the column that contains the weights. Default is "VAS".
<code>weight_range</code>	A numeric vector specifying the range of weights. Default is <code>c(0:100)</code> .
<code>weight_values</code>	A numeric vector specifying the weight values to be used. Default is <code>NULL</code> , in which case the <code>weight_range</code> will be used.
<code>weight_function</code>	A function to generate weights. Default is <code>.makeWeightsMixed</code> .
<code>sample_size</code>	An integer specifying the sample size for bootstrapping. Default is 1000.
<code>number_of_samples</code>	An integer specifying the number of bootstrap samples. Default is 1000.
<code>probability_levels</code>	A named vector specifying the probability levels for quantile.
<code>graph_title</code>	A string specifying the title of the graph. Default is an empty string.
<code>x_axis_title</code>	A string specifying the title for the x-axis. Default is an empty string.
<code>y_axis_title</code>	A string specifying the title for the y-axis. Default is an empty string.
<code>legend_name</code>	A string specifying the name for the legend. Default is "Type".
<code>legend_labels</code>	A character vector specifying the labels for the legend. Default is <code>NULL</code> .
<code>y_axis_limits</code>	A numeric vector specifying the limits for the y-axis. Default is <code>c(0.15, 0.95)</code> .
<code>y_min_value</code>	A string specifying the minimum value for the y-axis.
<code>y_max_value</code>	A string specifying the maximum value for the y-axis.
<code>alpha_1</code>	A numeric value between 0 and 1 to define the transparency of the interquartile range. Default is 0.15.
<code>alpha_2</code>	A numeric value between 0 and 1 to define the transparency of the confidence interval range. Default is 0.05.
<code>linetype_1</code>	A numeric value between 0 and 1 to define the line type of the interquartile range. Default is 1 "solid".
<code>linetype_2</code>	A numeric value between 0 and 1 to define the line type of the confidence interval range. Default 2 "dashed".
<code>interpretation_quartiles</code>	A numeric vector of values between 0 and 1 to define the quantile ranges for the figure interpretation. Default is <code>c(0, 0.25, 0.5, 0.75, 1)</code>
<code>elevation_threshold</code>	A numeric value specifying the threshold for elevation differences. Default is 0.05.
<code>slope_threshold</code>	A numeric value specifying the threshold for slope differences. Default is 0.02.
<code>color_palette</code>	A character vector specifying the color palette for the plot. Default is <code>c("#8dd3c7", "#bebada", "#80b1d3", "#fb8072", "#ffff67", "#fdb462", "#b3de69", "#fccde5", "#d9d9d9", "#bc80bd")</code> .
<code>weighted_statistics</code>	An optional data frame of pre-computed weighted statistics. Default is <code>NULL</code> .

Value

A list containing three elements: 'df' which is a data frame of weighted statistics, 'plot' which is the ggplot object representing the ribbon plot and 'interpretation' with the automatic interpretation of the ribbon plot.

Examples

```
# Define dimension names for EQ-5D-3L and EQ-5D-5L
dim.names.3L <- c("mobility", "selfcare", "activity", "pain", "anxiety")
dim.names.5L <- c("mobility5L", "selfcare5L", "activity5L", "pain5L", "anxiety5L")
# Compute EQ-5D scores using the eq5dsuite package
cdta$EQ5D3L <- eq5dsuite::eq5d3l(x = cdta,
                                country = "US",
                                dim.names = dim.names.3L)
cdta$EQ5D5L <- eq5dsuite::eq5d5l(x = cdta,
                                country = "US",
                                dim.names = dim.names.5L)
cdta$EQXW <- eq5dsuite::eqxw(x = cdta,
                             country = "US",
                             dim.names = dim.names.5L)
# Get severity ribbon plot
result <- severity_ribbon_plot(df = cdta, utility_columns = c("EQ5D3L", "EQ5D5L", "EQXW"))
```

single_transition_plots

single_transition_plots

Description

This function creates a scatter plot of mean one-level transitions for different EQ5D versions and specified value sets.

Usage

```
single_transition_plots(
  value_sets_3L = NULL,
  value_sets_5L = NULL,
  value_sets_XW = NULL,
  value_sets_XWR = NULL,
  graph_title = "",
  x_axis_title = NULL,
  x_min_value = -1,
  x_max_value = 1,
  y_axis_title = "Mean 1-level transition",
  y_min_value = 0,
  y_max_value = 0.5,
  legend_name = NULL
)
```

Arguments

value_sets_3L	A character vector specifying the country value sets for the EQ5D-3L version.
value_sets_5L	A character vector specifying the country value sets for the EQ5D-5L version.
value_sets_XW	A character vector specifying the country value sets for the EQ5D-XW version.
value_sets_XWR	A character vector specifying the country value sets for the EQ5D-XWR version.
graph_title	A character string specifying the title of the graph. Default is an empty string.
x_axis_title	A character string specifying the title of the x-axis. Default is "Utility".
x_min_value	A numeric specifying the minimum value for the x-axis. Default is -1.
x_max_value	A numeric specifying the maximum value for the x-axis. Default is 1.
y_axis_title	A character string specifying the title of the y-axis. Default is "Mean 1-level transition".
y_min_value	A numeric specifying the minimum value for the y-axis. Default is 0.
y_max_value	A numeric specifying the maximum value for the y-axis. Default is 0.5.
legend_name	A character string specifying the name of the legend. Default is NULL.

Value

A list of ggplot objects visualizing the mean one-level transitions for the specified EQ5D versions and value sets.

Note

This function is primarily intended to work with EQ5D data and it is not be applicable to other instruments.

Examples

```
single_transition_plots(value_sets_5L = "IT")
single_transition_plots(value_sets_3L = c("JP", "US"))
single_transition_plots(value_sets_3L = "ES",
                        value_sets_5L = "ES",
                        value_sets_XW = "ES",
                        value_sets_XWR = "ES")
```

Index

* datasets

- cdta, [5](#)
- example_data, [12](#)
- .add_EQ5D_utilities, [2](#)
- .cut_variable, [3](#)
- .makeWeightsGradient, [3](#)
- .makeWeightsMixed, [4](#)
- .makeWeightsTriangular, [4](#)

- cdta, [5](#)
- compute_F_statistics, [6](#)
- compute_utility_stats, [8](#)

- density_plot_empirical, [9](#)
- density_plot_theoretical, [10](#)

- example_data, [12](#)

- severity_ribbon_plot, [13](#)
- single_transition_plots, [15](#)