

Package ‘GTAPViz’

February 27, 2025

Title Visualization of GTAP Model Results from HAR and SL4 Files

Version 0.0.0.9000

Author Pattawee Puangchit <ppuangch@purdue.edu>

Description HARplusViz is an extension package of HARplus, designed to provide seamless data visualization for GTAP model results from ``.HAR`` and ``.SL4`` files. This package focuses on user-friendly visualization tools, enabling GTAP users to analyze and compare simulation results efficiently without requiring deep coding knowledge. With built-in functions for pivoting, mapping, unit assignments, and graphical representation, HARplusViz enhances economic model interpretation by generating structured, publication-ready plots. It is ideal for GTAP researchers, policymakers, and analysts looking for intuitive data exploration and visual insights from GEMPACK outputs.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

BugReports <https://github.com/bodysbobb/GTAPViz/issues>

URL <https://github.com/bodysbobb/GTAPViz>, <https://bodysbobb.github.io/GTAPViz/>

Imports HARplus,

stats,
ggplot2,
dplyr,
tidyr,
openxlsx,
utils,
tidyselect,
tools,
colorspace,
grDevices,
rlang,
scales,
purrr,
readxl,
writexl,
magrittr

Suggests knitr,
rmarkdown

VignetteBuilder knitr

Exports add_unit_col,
comparison_plot,
detail_plot,
macro_plot,
gtap_macros_data,
add_mapping_info

ImportFrom HARplus get_data_by_var load_sl4x
stats setNames
ggplot2 ggplot aes aes_string geom_bar geom_text geom_vline geom_col geom_hline facet_wrap
theme_minimal theme element_text element_rect element_blank unit margin labs
scale_fill_manual scale_y_continuous position_dodge expansion
dplyr filter mutate case_when
colorspace hex2RGB hex polarLUV sRGB
grDevices col2rgb rgb
rlang sym
scales squish

Depends R (>= 3.5)

Contents

add_mapping_info	2
comparison_plot	4
convert_units	5
decomp_table	6
detail_plot	7
gtap_macros_data	9
macro_plot	10
multi_variable_plot	11
plot_gtap_data	13
process_gtap_data	14
rename_GTAP_bilateral	16
rename_value	17
report_table	18
scalar_table	20
stack_plot	21
Index	23

add_mapping_info	<i>Add Mapping Information</i>
------------------	--------------------------------

Description

The description and unit are based on GTAPv7 model variables. Full list refer to <https://www.gtap.agecon.purdue.edu/mo>
If any variable is added apart from the default GTAPv7 model, the information will be left blank.

Usage

```
add_mapping_info(
  data_list,
  external_map = NULL,
  mapping = "GTAPv7",
  description_info = TRUE,
  unit_info = TRUE
)
```

Arguments

data_list	A list or data frame containing GTAP variables.
external_map	Optional. A data frame containing external mapping information with columns "Variable", "Description", and "Unit". If NULL, the default GTAPv7 mapping is used.
mapping	Optional. Either "GTAPv7", "No", "Yes", or "Mix". Determines the mapping approach. Default is "GTAPv7".
description_info	Optional. Logical. If TRUE, adds description information. Default is TRUE.
unit_info	Optional. Logical. If TRUE, adds unit information. Default is TRUE.

Details

For the "E1" header in HAR files, if using GTAPv7 or Mix mapping mode, it renames the "FORM" column to "Unit". When using "Yes" mapping mode with an external map, it will use provided Description and Unit values.

Value

A modified list or data frame with added description and/or unit information.

Author(s)

Pattawee Puangchit

Examples

```
sl4_data1 <- HARplus::load_sl4x(system.file("extdata", "TAR10.sl4", package = "HARplus"))
result <- add_mapping_info(sl4_data1, mapping = "GTAPv7", unit_info = FALSE) # Add only Description
result <- add_mapping_info(sl4_data1, mapping = "GTAPv7", description_info = FALSE) # Add only Unit
custom_mapping <- data.frame(Variable = c("qgdp", "EV"),
                             Description = c("Real GDP", "Economic Value"),
                             Unit = c("Percent", "million USD"))
result <- add_mapping_info(sl4_data1, external_map = custom_mapping, mapping = "Mix") # Custom + GTAPv7
```

comparison_plot	<i>Generate Comparison Plots</i>
-----------------	----------------------------------

Description

Creates bar plots comparing values across categories with flexible panel configuration.

Usage

```
comparison_plot(
  data,
  plot_var = NULL,
  x_axis_from,
  split_by,
  title_prefix = "",
  title_suffix = "",
  compare_by_experiment = FALSE,
  separate_figure = FALSE,
  invert_panel = FALSE,
  description_as_title = FALSE,
  output_dir = NULL,
  panel_rows = NULL,
  panel_cols = NULL,
  color_tone = NULL,
  legend_position = "none",
  width = NULL,
  height = NULL
)
```

Arguments

data	Data frame containing "Variable", "Value", "Experiment", "Unit" columns plus x_axis_from and split_by variables.
plot_var	Optional. Character vector or data frame with "Variable" column to specify which variables to plot.
x_axis_from	Column name to use for x-axis values (or panels if compare_by_experiment = TRUE).
split_by	Column name for separating plots (or panels if invert_panel = TRUE).
title_prefix	Optional text to prepend to plot titles.
title_suffix	Optional text to append to plot titles.
compare_by_experiment	If TRUE, Experiment is plotted on x-axis, x_axis_from values as panels.
separate_figure	If TRUE, creates individual figures for each panel value.
invert_panel	If TRUE, uses split_by for panels and Experiment for separation.
description_as_title	If TRUE, uses "Description" column for titles instead of "Variable".
output_dir	Optional path to save plots. If NULL, plots are only returned.

panel_rows	Optional number of panel rows. Auto-calculated if NULL.
panel_cols	Optional number of panel columns. Auto-calculated if NULL.
color_tone	Optional base color for the plot. NULL uses default ggplot2 colors.
legend_position	Position of legend: "none" (default), "bottom", "top", "left", "right".
width	Optional width in inches. Auto-calculated if NULL.
height	Optional height in inches. Auto-calculated if NULL.

Value

A list of ggplot objects or a single ggplot object depending on settings

Examples

```
## Not run:
comparison_plot(sl4.plot.data$REG,
  x_axis_from = "Region",
  plot_var = sl4plot,
  title_prefix = "Impact on",
  output_dir = output.folder,
  compare_by_experiment = FALSE,
  description_as_title = TRUE,
  separate_figure = FALSE,
  color_tone = "grey",
  width = 20,
  height = 12,
  legend_position = "bottom")

## End(Not run)
```

 convert_units

Convert Multiple Units in Nested Data Structures

Description

Convert Multiple Units in Nested Data Structures

Usage

```
convert_units(
  data,
  change_unit_from,
  change_unit_to,
  adjustment,
  value_col = "Value",
  unit_col = "Unit"
)
```

Arguments

data	A data structure (list, data.frame, or nested combination)
change_unit_from	Character vector of units to change (case-insensitive)
change_unit_to	Character vector of new units (same length as change_unit_from)
adjustment	Character vector of operations or numeric vector (same length as change_unit_from)
value_col	Column name containing values to adjust (default: "Value")
unit_col	Column name containing unit information (default: "Unit")

Value

Data structure with same format as input but with adjusted values and units

Author(s)

Pattawee Puangchit

decomp_table	<i>Create Decomposition Table with Multi-Data Support</i>
--------------	---

Description

Processes decomposition data, allowing multiple data frames as input. Converts specified columns into wide format per dataset while supporting multiple headers, independent column selections, and optional total calculations.

Usage

```
decomp_table(
  data_list,
  header,
  wide_cols,
  total_column = FALSE,
  export_table = FALSE,
  multi_sheet = FALSE,
  output_dir = NULL
)
```

Arguments

header	Optional. A character vector specifying headers to filter each dataset. Required if data is a list.
wide_cols	A named list specifying the column(s) to be converted into wide format for each dataset. Example: <code>list(A = "COLUMN", E1 = "PRICES")</code> .
total_column	Logical. If TRUE, calculates a total column for each dataset. Default is FALSE.
export_table	Logical. If TRUE, exports the result to an Excel file. Default is FALSE.
multi_sheet	Logical. If TRUE, exports each dataset to a separate sheet in the Excel file. Default is FALSE.

output_dir	Optional character. Directory to save the output file if export_table = TRUE.
data	A data frame or a named list of data frames, each containing decomposition data.
workbook_name	Optional character. Name of the output Excel workbook if export_table = TRUE. Default is "decomp_results.xlsx".

Details

- If data is a single data frame, header is ignored, and wide_cols should be a single column name.
- If data is a list of data frames, header must be specified to select relevant subsets of each dataset.
- wide_cols must be a named list matching the names of data (e.g., "A", "E1").
- When total_column = TRUE, numerical values in each row are summed into a "Total" column.
- If export_table = TRUE, the function saves the result to an Excel file, either as a single sheet or multiple sheets (if multi_sheet = TRUE).

Value

A list of data frames, each containing wide-format decomposition data. If export_table = TRUE, the results are also saved to an Excel file.

Examples

```
## Not run:
# Example list of data frames
har.plot.data <- list(
  A = data.frame(Experiment = c("Base", "Policy"),
    Header = "A", COLUMN = c("Tariff", "NTM"), Value = c(10, 5)),
  E1 = data.frame(Experiment = c("Base", "Policy"),
    Header = "E1", PRICES = c("GDP", "Consumption"), Value = c(105, 55))
)

# Convert decomposition tables for both datasets
result <- decomp_table(data = har.plot.data, header = c("A", "E1"),
  wide_cols = list(A = "COLUMN", E1 = "PRICES"), total_column = TRUE)

# Export as an Excel file with multiple sheets
decomp_table(data = har.plot.data, header = c("A", "E1"),
  wide_cols = list(A = "COLUMN", E1 = "PRICES"),
  total_column = TRUE, export_table = TRUE, multi_sheet = TRUE,
  output_dir = "results", workbook_name = "decomposition_summary.xlsx")

## End(Not run)
```

detail_plot

Generate Detailed Plot

Description

Creates a detailed plot reporting all regions or sectors depending on the variable selection. Uses a separate dataframe to specify variables to plot and their titles.

Usage

```

detail_plot(
  data,
  plot_var = NULL,
  y_axis_from,
  split_by,
  title_prefix = "",
  title_suffix = "",
  description_as_title = TRUE,
  panel_rows = NULL,
  panel_cols = NULL,
  positive_color = "#2E8B57",
  negative_color = "#CD5C5C",
  output_dir = NULL,
  top_impact = NULL,
  width = NULL,
  height = NULL,
  separate_figure = FALSE,
  invert_panel = FALSE,
  legend_position = "none",
  width_dodge = 0.5,
  width_bar = 0.4
)

```

Arguments

<code>data</code>	A data frame containing the full dataset.
<code>plot_var</code>	A data frame with "Variable" and "PlotTitle" columns specifying which variables to plot and their titles. If NULL, all variables in the data will be plotted with default titles.
<code>y_axis_from</code>	Character vector. Column names to use for the y-axis.
<code>split_by</code>	Character. Column used to separate different plots.
<code>title_prefix</code>	Optional character string to prepend to the title.
<code>title_suffix</code>	Optional character string to append to the title.
<code>description_as_title</code>	Logical. If TRUE, uses the "Description" column from <code>plot_var</code> as the plot title.
<code>panel_rows</code>	Optional numeric. Number of panel rows. If left blank, it will be automatically adjusted.
<code>panel_cols</code>	Optional numeric. Number of panel columns. If left blank, it will be automatically adjusted.
<code>positive_color</code>	Character. Color for positive values.
<code>negative_color</code>	Character. Color for negative values.
<code>output_dir</code>	Optional character. Directory to save the output plot.
<code>top_impact</code>	Optional numeric. Number of top impact factors to include. If left blank, all factors will be included.
<code>width</code>	Optional numeric. Width of the output plot. If left blank, it will be automatically adjusted.
<code>height</code>	Optional numeric. Height of the output plot. If left blank, it will be automatically adjusted.

separate_figure

Logical. If TRUE, creates separate figures for each combination of variable and category.

invert_panel

Logical. If TRUE, swaps the roles of Experiment and split_by in faceting and figure separation.

Value

A ggplot object or NULL if no plots could be generated.

gtap_macros_data	<i>Extract and Aggregate Scalar Macroeconomic Variables</i>
------------------	---

Description

Extracts scalar macroeconomic variables from multiple SL4 datasets and aggregates them into a structured data frame.

Usage

```
gtap_macros_data(..., experiment_names = NULL)
```

Arguments

...

One or more SL4 data objects. Can be either:

- A single list containing multiple SL4 datasets
- Multiple SL4 datasets as separate arguments

experiment_names

Optional character vector of experiment names. If NULL, names are inferred.

Value

A data frame with three columns:

- "Experiment": The scenario name
- "Variable": The macroeconomic variable name
- "Value": The extracted numeric value

Author(s)

Pattawee Puangchit

See Also

[add_unit_col](#), [add_mapping_info](#)

Examples

```
# Import sample data
sl4_data1 <- HARplus::load_sl4x(system.file("extdata", "TAR10.sl4", package = "HARplus"))
sl4_data2 <- HARplus::load_sl4x(system.file("extdata", "SUBT10.sl4", package = "HARplus"))

# Method 1: Using a list of datasets
data_list <- list(base = sl4_data1, policy = sl4_data2)
Macros1 <- gtap_macros_data(data_list)

# Method 2: Using multiple arguments
Macros2 <- gtap_macros_data(sl4_data1, sl4_data2,
                             experiment_names = c("Base", "Policy"))
```

macro_plot

Generate Macro Plot with Flexible Title Options

Description

Creates figures of macro variables from the Macros header in the SL4. Uses a separate dataframe to specify variables to plot with options for using either variable names, PlotTitle, or Description columns for plot titles.

Usage

```
macro_plot(
  data,
  plot_var = NULL,
  title_prefix = "",
  title_suffix = "",
  compare_by_experiment = FALSE,
  description_as_title = FALSE,
  output_dir = NULL,
  panel_rows = NULL,
  panel_cols = NULL,
  color_tone = NULL,
  separate_figure = FALSE,
  width = NULL,
  height = NULL,
  legend_position = "none"
)
```

Arguments

data	A data frame containing the full dataset with macroeconomic variables.
plot_var	A data frame with "Variable" column and optionally "PlotTitle" or "Description" columns specifying which variables to plot and their titles. If NULL, all variables in the data will be plotted with default titles.
title_prefix	Optional character string to prepend to the title.
title_suffix	Optional character string to append to the title.

compare_by_experiment	Logical. If TRUE, compares experiments within x-axis categories.
description_as_title	Logical. If TRUE, uses the "Description" column for plot titles instead of "Plot-Title". Default is FALSE.
output_dir	Optional character. Directory to save the output plot.
panel_rows	Optional numeric. Number of panel rows. If left blank, it will be automatically adjusted.
panel_cols	Optional numeric. Number of panel columns. If left blank, it will be automatically adjusted.
color_tone	Optional character. Base color for the plot. If left blank, a default color will be used.
separate_figure	Logical. If FALSE, combines multiple variables into one plot.
width	Optional numeric. Width of the output plot. If left blank, it will be automatically adjusted.
height	Optional numeric. Height of the output plot. If left blank, it will be automatically adjusted.
legend_position	Position of the legend: "none", "bottom", "top", "left", or "right"

Value

A ggplot object or a list of plots if separate_figure = TRUE.

Examples

```
## Not run:
macro_plot(Macros,
  plot_var = macro.map,
  compare_by_experiment = FALSE,
  color_tone = "grey",
  output_dir = output.folder,
  description_as_title = FALSE,
  panel_rows = 2,
  width = 45,
  separate_figure = FALSE,
  legend_position = "bottom")

## End(Not run)
```

multi_variable_plot	<i>Generate Multi-Variable Plot for Regions</i>
---------------------	---

Description

Creates plots showing multiple variables for a specific region across experiments. Designed to work with data from get_data_by_dims for the REG dimension.

Usage

```
multi_variable_plot(
  data,
  x_axis_from,
  plot_var = NULL,
  title_prefix = "",
  title_suffix = "",
  separate_figure = FALSE,
  sort_by = "value",
  limit_vars = NULL,
  output_dir = NULL,
  color_tone = NULL,
  same_scale = FALSE,
  panel_rows = NULL,
  panel_cols = NULL,
  width = NULL,
  height = NULL,
  legend_position = "bottom"
)
```

Arguments

<code>data</code>	A data frame containing GTAP output data with Region, Variable, and Value columns.
<code>plot_var</code>	A data frame with "Variable" and "PlotTitle" columns specifying which variables to plot and their titles. If NULL, all variables for the given region will be plotted with default titles.
<code>title_prefix</code>	Optional character string to prepend to the title.
<code>title_suffix</code>	Optional character string to append to the title.
<code>sort_by</code>	Character. How to sort the variables: "name" (alphabetical), "value" (by magnitude), or "none" (as provided). Default is "value".
<code>limit_vars</code>	Numeric. Maximum number of variables to include. If NULL, all variables are included.
<code>output_dir</code>	Optional character. Directory to save the output plot.
<code>color_tone</code>	Optional character. Base color for the plot.
<code>same_scale</code>	Logical. If TRUE, all variables use the same y-axis scale. Default is FALSE.
<code>panel_rows</code>	Optional numeric. Number of panel rows for the plot.
<code>panel_cols</code>	Optional numeric. Number of panel columns for the plot.
<code>width</code>	Optional numeric. Width of the output plot.
<code>height</code>	Optional numeric. Height of the output plot.
<code>legend_position</code>	Character. Position of the legend: "none", "bottom", "top", "left", or "right".
<code>region</code>	Character. The specific region/country to plot.

Value

A ggplot object or a list of ggplot objects (one per unit type).

plot_gtap_data	<i>Extract GTAP Data</i>
----------------	--------------------------

Description

Extracts data from GTAP SL4 and HAR files with simplified filtering options.

Usage

```
plot_gtap_data(
  sl4file,
  harfile,
  experiment,
  region_select = NULL,
  sector_select = NULL,
  mapping_info = "GTAPv7",
  project_dir = NULL,
  input_dir = NULL,
  subtotal = FALSE,
  sl4_list_name = "sl4.plot.data",
  har_list_name = "har.plot.data",
  sl4_structure_name = "sl4.structure",
  har_structure_name = "har.structure"
)
```

Arguments

sl4file	A data frame containing SL4 mapping information, or FALSE to skip SL4 processing.
harfile	A data frame containing HAR mapping information, or FALSE to skip HAR processing.
experiment	A character vector of experiment names to be processed.
region_select	Optional. A vector specifying regions to filter the data.
sector_select	Optional. A vector specifying sectors to filter the data.
mapping_info	A character string indicating the mapping mode (e.g., "GTAPv7"). Default is "GTAPv7".
project_dir	Optional. Path to the project directory containing the "in" folder.
input_dir	Optional. Directory path for input files; overrides project_dir/in if provided.
sl4_list_name	A character string specifying the global variable name for SL4 plotting data. Default is "sl4.plot.data".
har_list_name	A character string specifying the global variable name for HAR plotting data. Default is "har.plot.data".
sl4_structure_name	A character string specifying the global variable name for SL4 structure. Default is "sl4.structure".
har_structure_name	A character string specifying the global variable name for HAR structure. Default is "har.structure".

Value

A list containing extracted data with applied filters.

Examples

```
# Extract specific variables with region filters
## Not run:
sl4_selected <- data.frame(
  Variable = c("qo", "qgdp", "EV"),
  stringsAsFactors = FALSE
)

result <- plot_gtap_data(
  sl4file = sl4_selected,
  harfile = FALSE,
  experiment = c("baseline", "policy"),
  region_select = c("USA", "CHN", "EU"),
  input_dir = "D:/GTAP_inputs"
)

## End(Not run)
```

process_gtap_data

Process GTAP Data Automation

Description

Processes GTAP data from SL4 and HAR files automatically using external mapping files. By default, uses GTAPv7 definition for variable descriptions and units.

Usage

```
process_gtap_data(
  sl4file,
  harfile,
  experiment,
  mapping_info = "GTAPv7",
  project_dir = NULL,
  input_dir = NULL,
  output_dir = NULL,
  output_formats = NULL,
  export_data = TRUE
)
```

Arguments

sl4file	Data frame, NULL, or FALSE. Mapping for SL4 variables. Set to NULL to extract all variables, or FALSE to skip SL4 processing.
harfile	Data frame, NULL, or FALSE. Mapping for HAR variables. Set to NULL to extract all variables, or FALSE to skip HAR processing.
experiment	Character vector. Case names to process.

mapping_info	Character. Mapping mode: "GTAPv7" (default), "Yes", "No", or "Mix".
project_dir	Character. Path to the project folder with "in" and "out" subfolders.
input_dir	Character. Path to the input folder. Overrides project_dir/in if specified.
output_dir	Character. Path to the output folder. Overrides project_dir/out if specified.
output_formats	Character vector. Output formats (e.g., c("csv", "stata", "rds", "txt")).
export_data	Logical. If TRUE, exports processed data to files.

Details

This function provides a complete automation workflow for processing GTAP model outputs. It handles both SL4 files (solution files) and HAR files (welfare decomposition files).

The key parameters `sl4file` and `harfile` each accept three different input types:

- **Data frame:** Contains variable mappings with required "Variable" column and optional "Description" and "Unit" columns. When provided, only specified variables will be extracted.
- **NULL:** Extracts all available variables from the respective file type, using GTAPv7 definitions for descriptions and units.
- **FALSE:** Completely skips processing of that file type, allowing the function to focus only on the other file type.

The `mapping_info` parameter controls how descriptions and units are assigned:

- **GTAPv7:** Uses standard GTAPv7 definitions (default)
- **Yes:** Uses only the supplied descriptions and units from `sl4file/harfile`
- **No:** Does not add any descriptions or units
- **Mix:** Prioritizes supplied descriptions and units, falling back to GTAPv7 for any missing values

Value

If `export_data=TRUE`, returns NULL invisibly. If `export_data=FALSE`, returns a list of processed data.

Author(s)

Pattawee Puangchit

See Also

[add_mapping_info](#), [gtap_macros_data](#)

Examples

```
# Example 1: Process all variables from both SL4 and HAR files
## Not run:
process_gtap_data(
  sl4file = NULL,
  harfile = NULL,
  experiment = c("baseline", "scenario1"),
  project_dir = "D:/GTAP_Project",
  mapping_info = "GTAPv7"
)
```

```

## End(Not run)

# Example 2: Process only specific variables using custom mapping
## Not run:
# Create mapping for selected SL4 variables
sl4_selected <- data.frame(
  Variable = c("qo", "qgdp", "EV"),
  stringsAsFactors = FALSE
)

# Create mapping for selected HAR variables
har_selected <- data.frame(
  Variable = c("A", "E1"), # Welfare and ToT decomposition
  stringsAsFactors = FALSE
)

process_gtap_data(
  sl4file = sl4_selected,
  harfile = har_selected,
  experiment = c("baseline", "policy"),
  input_dir = "D:/GTAP_inputs",
  output_dir = "D:/GTAP_results"
)

## End(Not run)

# Example 3: Custom descriptions and units instead of GTAPv7 defaults
## Not run:
# Create custom mapping with descriptions and units
custom_mapping <- data.frame(
  Variable = c("qo", "qgdp", "EV"),
  Description = c("Output by sector", "Real GDP", "Equivalent Variation"),
  Unit = c("% change", "% change", "million USD"),
  stringsAsFactors = FALSE
)

process_gtap_data(
  sl4file = custom_mapping,
  harfile = FALSE, # Skip HAR processing
  experiment = c("baseline", "policy"),
  project_dir = "D:/GTAP_Project",
  mapping_info = "Yes" # Use only supplied descriptions and units
)

## End(Not run)

```

rename_GTAP_bilateral *Rename GTAP Bilateral Trade Columns*

Description

Renames bilateral trade columns in GTAP data output, specifically targeting REG columns to Source and Destination in data frames with COMMREGREG dimension structure.

Usage

```
rename_GTAP_bilateral(data)
```

Arguments

data List or data frame. Output from `get_data_by_var`, `get_data_by_dims`, or `group_data_by_dims`.

Value

Data with renamed columns, maintaining the original structure type.

Author(s)

Your Name

Examples

```
# For direct data frame output
df <- get_data_by_var("qxs", sl4_data)
df_renamed <- rename_GTAP_bilateral(df)

# For list output with multiple variables
data_list <- get_data_by_var(c("qxs", "viws"), sl4_data)
renamed_list <- rename_GTAP_bilateral(data_list)

# For grouped dimension output
grouped_data <- group_data_by_dims("COMM*REG*REG", sl4_data)
renamed_grouped <- rename_GTAP_bilateral(grouped_data)
```

rename_value	<i>Rename Values in a Data Frame or List of Data Frames</i>
--------------	---

Description

Replaces specified values in a given column with new values based on a mapping file. If `column_name` is not provided, it will be automatically extracted from `mapping.file` if available.

Usage

```
rename_value(data, column_name = NULL, mapping.file)
```

Arguments

data A data frame or a list of data frames containing the column to modify.

column_name Optional. Character. The name of the column where values will be changed. If NULL, the function will attempt to extract `ColumnName` from `mapping.file`.

mapping.file A data frame containing `OldName` and `NewName` columns that specify the mapping of values to be replaced. Optionally, a `ColumnName` column may define which column should be modified.

Value

A modified data frame or list of data frames with specified values replaced.

Author(s)

Pattawee Puangchit

Examples

```
# Example mapping file
wefare.decomp.rename <- data.frame(
  OldName = c("alloc_A1", "ENDWB1", "tech_C1"),
  NewName = c("Allocation", "Endowment", "Technology"),
  ColumnName = "Variable",
  stringsAsFactors = FALSE
)

# Load example data
har_data <- HARplus::load_harx(c("globalcgds", "vgdpwld", "WEV"))
har_data <- HARplus::get_data_by_var(c("globalcgds", "vgdpwld", "WEV"), har_data)

# Apply renaming
modified_list <- rename_value(har_data, mapping.file = wefare.decomp.rename)
```

report_table

Create Report Table

Description

Generates a structured report table from a given dataset, allowing for flexible formatting and customization of column names. The function reshapes the data to present variables in a wide format with experimental groups as rows.

Usage

```
report_table(
  data,
  vars,
  x_axis_from,
  output_dir = NULL,
  workbook_name = NULL,
  sheet_name = "Results",
  col_names = NULL
)
```

Arguments

data	A data frame or list containing the dataset. If a list, it is converted into a data frame before processing.
vars	A character vector specifying the variables to include in the report.
x_axis_from	A character string specifying the column used for grouping (e.g., "REG", "COMM").

output_dir	Optional character string specifying the directory where the report will be saved as an Excel file. If NULL, the table is not exported.
workbook_name	Optional character string for the Excel workbook name. Default is "Report_table".
sheet_name	Optional character string for the worksheet name. Default is "Results".
col_names	Optional data frame or character vector for renaming columns. If a data frame, it must contain "Variable" and "PlotTitle" columns to map variable names to custom titles. If a character vector, it must match the length of vars.

Details

- The function checks for the required columns "Experiment", "Variable", "Value", and the specified x_axis_from column.
- The table is formatted to ensure numerical values are rounded to two decimal places.
- If output_dir is provided, the table is exported as an Excel file with enhanced formatting.
- If column renaming (col_names) is specified, it is applied to improve readability.

Value

A formatted data frame in wide format with grouped variables as columns and experimental groups as rows. If output_dir is provided, the function also saves the output to an Excel file.

Author(s)

Pattawee Puangchit

See Also

[scalar_table](#), [decomp_table](#)

Examples

```
## Not run:
# Example dataset
data <- data.frame(
  Experiment = rep(c("Baseline", "Policy"), each = 3),
  REG = rep(c("USA", "CHN", "EU"), 2),
  Variable = rep(c("GDP", "Consumption", "Investment"), 2),
  Value = c(100, 50, 30, 105, 55, 35)
)

# Generate report table
report <- report_table(data, vars = c("GDP", "Consumption"),
  x_axis_from = "REG", output_dir = "results")

# Custom column names
col_names_df <- data.frame(
  Variable = c("GDP", "Consumption"),
  PlotTitle = c("Gross Domestic Product", "Household Consumption")
)

report_table(data, vars = c("GDP", "Consumption"), x_axis_from = "REG",
  output_dir = "results", col_names = col_names_df)

## End(Not run)
```

scalar_table	Create Scalar Result Table
--------------	----------------------------

Description

Generates a formatted table summarizing scalar results from a given dataset. The function transforms long-format data into a wide format with scenarios as columns, making comparisons across experiments more accessible.

Usage

```
scalar_table(
  data,
  vars = NULL,
  output_dir = NULL,
  workbook_name = "scalar_results",
  sheet_name = "Results"
)
```

Arguments

data	A data frame or list containing scalar data, where each observation consists of an experiment name, variable name, and corresponding value.
vars	Optional character vector of variable names to include in the table. If NULL, all available variables in the dataset will be used.
output_dir	Optional character. Directory where the generated table will be saved as an Excel file. If NULL, no file will be created.
workbook_name	Optional character. Name of the output Excel workbook. Default is "scalar_results".
sheet_name	Optional character. Name of the worksheet in the Excel file. Default is "Results".

Details

- If data is a list and not a data frame, the function attempts to combine it into a data frame.
- The table is formatted to ensure numerical values are rounded to two decimal places.
- If output_dir is specified, the table is exported as an Excel file with proper formatting.
- Required columns in data are "Experiment", "Variable", and "Value".

Value

A formatted data frame in wide format with variables as rows and experiments as columns. If output_dir is provided, the function also saves the output to an Excel file.

Author(s)

Pattawee Puangchit

See Also

[report_table](#), [decomp_table](#)

Examples

```
## Not run:
sl4_data1 <- HARplus::get_data_by_var(c("globalcgds", "vgdpwld", "WEV"), sl4_data)
scalar_table <- scalar_table(
  data = sl4_data1,
  vars = "vgdpwld",
  output_dir = temp_dir,
  workbook_name = "global_indicators",
  sheet_name = "Results"
)

## End(Not run)
```

stack_plot

Generate Stacked Bar Plot with Auto-Dimensions

Description

Creates stacked bar plots with automatic total calculations and flexible grouping. Uses a separate dataframe to specify variables to plot and their titles. Auto-adjusts dimensions and panel layout if not specified.

Usage

```
stack_plot(
  data,
  plot_var = NULL,
  x_axis_from,
  stack_value_from,
  title_prefix = "",
  title_suffix = "",
  y_axis_title = NULL,
  compare_by_experiment = FALSE,
  separate_figure = FALSE,
  unstack_plot = FALSE,
  output_dir = NULL,
  panel_rows = NULL,
  panel_cols = NULL,
  color_tone = NULL,
  show_total = TRUE,
  description_as_title = TRUE,
  legend_position = "bottom",
  y_limit = NULL,
  width = NULL,
  height = NULL
)
```

Arguments

data A data frame containing the full dataset.

plot_var	A data frame with "Variable" and "PlotTitle" columns specifying which variables to plot and their titles. If NULL, all variables in the data will be plotted with default titles.
x_axis_from	Character. Column name to use for x-axis grouping
stack_value_from	Character. Column name containing the stack categories
title_prefix	Optional character string to prepend to the title
title_suffix	Optional character string to append to the title
y_axis_title	Character. Y-axis label. If NULL, uses Unit if available.
compare_by_experiment	Logical. If TRUE, compares x-axis values within experiments
separate_figure	Logical. If TRUE, creates separate figures instead of panels
unstack_plot	Logical. If TRUE, displays components side by side instead of stacked and creates separate figures for each x_axis_from value
output_dir	Optional character. Directory to save the output plot
panel_rows	Optional numeric. Number of panel rows
panel_cols	Optional numeric. Number of panel columns
color_tone	Optional character. Base color for plot color generation
legend_position	Character. Position of the legend: "bottom" (default), "top", "left", "right", or "none"
y_limit	Optional numeric vector. Custom y-axis limits (e.g., c(-5, 5))
width	Optional numeric. Width of the output plot
height	Optional numeric. Height of the output plot

Value

A list of ggplot objects or a single ggplot object depending on settings

Index

`add_mapping_info`, [2](#), [9](#), [15](#)
`add_unit_col`, [9](#)

`comparison_plot`, [4](#)
`convert_units`, [5](#)

`decomp_table`, [6](#), [19](#), [20](#)
`detail_plot`, [7](#)

`gtap_macros_data`, [9](#), [15](#)

`macro_plot`, [10](#)
`multi_variable_plot`, [11](#)

`plot_gtap_data`, [13](#)
`process_gtap_data`, [14](#)

`rename_GTAP_bilateral`, [16](#)
`rename_value`, [17](#)
`report_table`, [18](#), [20](#)

`scalar_table`, [19](#), [20](#)
`stack_plot`, [21](#)