Package 'HARplus'

March 9, 2025

Title Enhanced Processing of GEMPACK .HAR and .SL4 Files

Version 1.0.1

Author Pattawee Puangchit <ppuangch@purdue.edu>

Description HARplus is an advanced package for processing GEMPACK `.HAR` and `.SL4` files, designed to streamline large-scale economic modeling workflows. It extends beyond traditional `.HAR` readers by offering efficient data extraction, structured metadata management, and flexible data transformations. The package supports multi-file analysis, subtotal level handling, dimension renaming, and optimized memory usage to ensure seamless handling of complex datasets. Unlike standard `.HAR` readers, HARplus enables advanced manipulations such as pivoting, renaming dimensions, exporting to multiple formats, and cross-comparing structured data across different sources. The package is fully independent and does not require the HARr package.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

BugReports https://github.com/bodysbobb/HARplus/issues

URL https://github.com/bodysbobb/HARplus, https://bodysbobb.github.io/HARplus/

Imports openxlsx, haven, stats, utils, tidyr, tools, tidyselect

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Maintainer Pattawee Puangchit <ppuangch@purdue.edu>

Contents

`compare_var_structure` *Compare Variable Structures Across SL4 and HAR Objects*

Description

Compares variable structures across multiple SL4 and HAR datasets to ensure compatibility. Identifies matching and mismatched variable structures, helping users diagnose inconsistencies.

Usage

```
compare_var_structure(variables = NULL, ..., keep_unique = FALSE)
```

Arguments

`variables`	Character vector. Variable names to compare. Use `NULL` or `"ALL"` to compare all variables.
`...`	Named SL4 or HAR objects to compare.
`keep_unique`	Logical. If `TRUE`, returns unique variable structures across datasets instead of checking for compatibility. Default is `FALSE`.

Details

- Verifies whether variables have consistent structures across multiple datasets.
- Ensures correct ordering of dimensions and checks for structural compatibility.
- If `keep_unique = TRUE`, returns a list of unique variable structures instead of performing a direct comparison.
- Useful for merging or aligning datasets before further processing.
- Helps detect differences in variable dimensions, which may arise due to model updates or dataset variations.

Value

A list containing:

- `match`: A data frame listing variables with identical structures across datasets.
- `diff`: A data frame listing variables with mismatched structures, useful for debugging and alignment.
- If `keep_unique = TRUE`, instead of `match` and `diff`, returns a data frame with distinct variable structures across datasets.

Author(s)

Pattawee Puangchit

See Also

`get_var_structure`, `get_dim_patterns`, `get_dim_elements`

Examples

```
# Import sample data:
har_data1 <- load_harx(system.file("extdata", "TAR10-WEL.har", package = "HARplus"))
har_data2 <- load_harx(system.file("extdata", "SUBT10-WEL.har", package = "HARplus"))

# Compare structure for a single variable across multiple datasets
compare_var_structure("A", har_data1, har_data2)

# Compare structure for multiple variables across multiple datasets
comparison_multiple <- compare_var_structure(c("A", "E1"), har_data1, har_data2)

# Extract unique variable structures across multiple datasets
unique_vars <- compare_var_structure("ALL", har_data1, har_data2, keep_unique = TRUE)
```

---

export_data *Export Data to Various Formats (CSV/STATA/TEXT/RDS/XLSX)*

Description

Exports structured SL4 or HAR data to multiple file formats, including CSV, Stata, TXT, RDS, and XLSX. Supports nested lists, automatic subfolder creation, and multi-sheet Excel exports.

Usage

```
export_data(
  data,
  output_path,
  format = "csv",
  prefix = "",
  create_subfolder = FALSE,
  multi_sheet_xlsx = FALSE,
  xlsx_filename = NULL,
  report_output = FALSE
)
```

Arguments

`data`	A list or data frame. The SL4 or HAR data to export.
`output_path`	Character. The base output directory or file path.
`format`	Character. The export format (`"csv"`, `"stata"`, `"txt"`, `"rds"`, `"xlsx"`). Default is `"csv"`.
`prefix`	Character. An optional prefix added to exported file names. Default is `""` (empty).
`create_subfolder`	Logical. If `TRUE`, creates a subfolder for each format. Default is `FALSE`.
`multi_sheet_xlsx`	Logical. If TRUE, exports lists as multi-sheet XLSX files.
`xlsx_filename`	An optional filename for the XLSX file (used when `multi_sheet_xlsx = TRUE`).
`report_output`	Logical. If TRUE, generates an export report.

Details

- Supports exporting data in `"csv"`, `"stata"`, `"txt"`, `"rds"`, and `"xlsx"` formats.
- Handles nested lists and exports each data frame individually.
- Optionally creates subfolders for each format (`create_subfolder = TRUE`).
- Customizes file names using `prefix`.
- When multi_sheet_xlsx = TRUE, all exported data is stored in a **single Excel workbook**, with each dataset as a separate sheet.
- If exporting to Stata (`"stata"` format), column names containing `.` will be replaced with `_` to ensure compatibility.
- If `multi_sheet_xlsx = TRUE`, list elements are exported as separate sheets in a single XLSX file.
- The function creates necessary directories if they do not exist.

Value

A list containing the file paths of the exported data.

Author(s)

Pattawee Puangchit

See Also

`pivot_data`, `get_data_by_var`, `get_data_by_dims`

Examples

```
## Not run:
# Import sample data:
sl4_data <- load_sl4x(system.file("extdata", "TAR10.sl4", package = "HARplus"))

# Extract data
data_multiple <- get_data_by_var(c("qo", "pca"), sl4_data)
```

```
# Export
export_data(data_multiple, file.path(tempdir(), "output_directory"),
           format = c("csv", "xlsx", "stata", "txt", "rds"),
           create_subfolder = TRUE,
           multi_sheet_xlsx = TRUE)

## End(Not run)
```

`export_hierarchy_to_excel` *Export Hierarchical Pivot Table to Excel (Internal)*

Description

Internal function to export hierarchical pivot tables to Excel with formatted headers.

Usage

```
export_hierarchy_to_excel(pivot_df, file_path, xlsx_filename = NULL)
```

Arguments

`pivot_df`	A hierarchical pivot object created by pivot_data_hierarchy().
`file_path`	Character. The file path for Excel export.
`xlsx_filename`	Character. The name for the Excel file when using multi_sheet_xlsx. If NULL, uses the name of the dataset. Default is `NULL`.

Value

Invisibly returns NULL.

`get_data_by_dims` *Extract Data by Dimension Patterns from SL4 or HAR Objects*

Description

Retrieves structured data from SL4 or HAR objects based on specified dimension patterns. Supports multiple experiments and merging datasets while maintaining structured dimension metadata.

Usage

```
get_data_by_dims(
  patterns = NULL,
  ...,
  experiment_names = NULL,
  subtotal_level = FALSE,
  rename_cols = NULL,
  merge_data = FALSE,
  pattern_mix = FALSE
)
```

Arguments

patterns	Character vector. Dimension patterns to extract. Use "ALL" or NULL to extract all available patterns.
...	One or more SL4 or HAR data objects loaded using load_sl4x() or load_harx().
experiment_names	Character vector. Names assigned to each dataset. If NULL, names are inferred.
subtotal_level	Character or logical. Determines which decomposition levels to retain: • "total": Keeps only "TOTAL" values. • "decomposed": Keeps only decomposed values (excludes "TOTAL"). • "all": Keeps all rows. • TRUE: Equivalent to "all", retaining both "TOTAL" and decomposed values. • FALSE: Equivalent to "total", keeping only "TOTAL" values.
rename_cols	Named vector. Column name replacements (c("old_name" = "new_name")).
merge_data	Logical. If TRUE, attempts to merge data across multiple experiments. Default is FALSE.
pattern_mix	Logical. If TRUE, allows flexible pattern matching, ignoring dimension order. Default is FALSE.

Details

- Extracts variables matching specified dimension patterns.
- Allows for flexible pattern matching (pattern_mix = TRUE).
- Supports merging data across multiple experiments (merge_data = TRUE).
- Provides column renaming functionality (rename_cols).
- Handles subtotal filtering (subtotal_level), controlling whether "TOTAL" or decomposed values are retained.

Value

A structured list of extracted data:

- If merge_data = FALSE, returns a named list where each element corresponds to an experiment.
- If merge_data = TRUE, returns a named list of all merged data

Author(s)

Pattawee Puangchit

See Also

get_data_by_var, group_data_by_dims

Examples

```
# Import sample data:
sl4_data <- load_sl4x(
  system.file("extdata", "TAR10.sl4", package = "HARplus")
)
sl4_data1 <- load_sl4x(
  system.file("extdata", "SUBT10.sl4", package = "HARplus")
)

# Extract data for a single dimension pattern
data_single_pattern <- get_data_by_dims(
  "comm*reg",
  sl4_data
)

# Extract multiple dimension patterns
data_multiple_patterns <- get_data_by_dims(
  c("comm*reg", "REG*ACTS"),
  sl4_data
)

# Extract all dimension patterns separately from multiple datasets
data_all_patterns <- get_data_by_dims(
  NULL,
  sl4_data, sl4_data1,
  merge_data = FALSE
)

# Merge data for identical patterns across multiple datasets
data_merged_patterns <- get_data_by_dims(
  NULL,
  sl4_data, sl4_data1,
  merge_data = TRUE
)

# Merge data while allowing interchangeable dimensions (e.g., A*B = B*A)
data_pattern_mixed <- get_data_by_dims(
  NULL,
  sl4_data, sl4_data1,
  merge_data = TRUE,
  pattern_mix = TRUE
)

# Retain only "TOTAL" values
data_total_only <- get_data_by_dims(
  "comm*reg",
  sl4_data,
  subtotal_level = "total"
)
data_total_only_alt <- get_data_by_dims(
  "comm*reg",
  sl4_data,
  subtotal_level = FALSE
)

# Retain only decomposed components
```

```
data_decomposed_only <- get_data_by_dims(
  "comm*reg",
  sl4_data,
  subtotal_level = "decomposed"
)

# Retain all value levels
data_all_decomp <- get_data_by_dims(
  "comm*reg",
  sl4_data,
  subtotal_level = "all"
)
data_all_decomp_alt <- get_data_by_dims(
  "comm*reg",
  sl4_data,
  subtotal_level = TRUE
)

# Rename specific columns
data_renamed <- get_data_by_dims(
  "comm*reg",
  sl4_data,
  rename_cols = c(REG = "Region", COMM = "Commodity")
)

# Merge data with custom experiment names
data_merged_experiments <- get_data_by_dims(
  "comm*reg",
  sl4_data, sl4_data1,
  experiment_names = c("EXP1", "EXP2"),
  merge_data = TRUE
)
```

`get_data_by_var` *Extract Variable Data from SL4 or HAR Objects*

Description

Extracts structured data for one or more variables from SL4 or HAR objects, transforming array-like data into a tidy format.

Usage

```
get_data_by_var(
  var_names = NULL,
  ...,
  experiment_names = NULL,
  subtotal_level = FALSE,
  rename_cols = NULL,
  merge_data = FALSE
)
```

Arguments

var_names	Character vector. Variable names to extract. Use "ALL" or NULL to extract all available variables.
...	One or more SL4 or HAR data objects loaded using load_sl4x() or load_harx().
experiment_names	Character vector. Names assigned to each dataset. If NULL, names are inferred.
subtotal_level	Character or logical. Determines which decomposition levels to retain: • "total": Keeps only "TOTAL" values. • "decomposed": Keeps only decomposed values (excludes "TOTAL"). • "all": Keeps all rows. • TRUE: Equivalent to "all", retaining both "TOTAL" and decomposed values. • FALSE: Equivalent to "total", keeping only "TOTAL" values.
rename_cols	Named vector. Column name replacements (c("old_name" = "new_name")).
merge_data	Logical. If TRUE, attempts to merge data across multiple experiments. Default is FALSE.

Details

- Retrieves specific variables, multiple variables, or all available variables from SL4 or HAR datasets.
- Supports merging data from multiple experiments (merge_data = TRUE).
- Allows renaming of column names (rename_cols).
- Handles subtotal filtering (subtotal_level), controlling whether "TOTAL" or decomposed values are retained.

Value

A list of structured data:

- If merge_data = FALSE, returns a named list where each element corresponds to an experiment.
- If merge_data = TRUE, returns a named list of all merged data

Author(s)

Pattawee Puangchit

See Also

get_data_by_dims, , group_data_by_dims, load_sl4x, load_harx

Examples

```
# Import sample data:
sl4_data <- load_sl4x(system.file("extdata", "TAR10.sl4", package = "HARplus"))
sl4_data1 <- load_sl4x(system.file("extdata", "SUBT10.sl4", package = "HARplus"))

# Extract a single variable
data_qo <- get_data_by_var("qo", sl4_data)
```

```
# Extract multiple variables
data_multiple <- get_data_by_var(c("qo", "qgdp"), sl4_data)

# Extract all variables separately from multiple datasets
data_all <- get_data_by_var(NULL, sl4_data, sl4_data1, merge_data = FALSE)

# Merge variable data across multiple datasets
data_merged <- get_data_by_var(NULL, sl4_data, sl4_data1, merge_data = TRUE)

# Retain only "TOTAL" values, removing decomposed components (subtotal_level = "total" or FALSE)
data_total_only <- get_data_by_var("qo", sl4_data, subtotal_level = "total")
data_total_only_alt <- get_data_by_var("qo", sl4_data, subtotal_level = FALSE)

# Retain only decomposed components, removing "TOTAL" (subtotal_level = "decomposed")
data_decomposed_only <- get_data_by_var("qo", sl4_data, subtotal_level = "decomposed")

# Retain all value levels (subtotal_level = "all" or TRUE)
data_all_decomp <- get_data_by_var("qo", sl4_data, subtotal_level = "all")
data_all_decomp_alt <- get_data_by_var("qo", sl4_data, subtotal_level = TRUE)

# Rename specific columns
data_renamed <- get_data_by_var("qo", sl4_data, rename_cols = c(REG = "Region", COMM = "Commodity"))

# Merge data across multiple datasets with custom experiment names
data_merged_experiments <- get_data_by_var("qo", sl4_data, sl4_data1,
experiment_names = c("EXP1", "EXP2"),
merge_data = TRUE)
```

`get_dim_elements` *Get Dimension Elements from SL4 and HAR Objects*

Description

Extracts and lists unique dimension elements (e.g., `REG`, `COMM`, `ACTS`) from one or more datasets.

Usage

```
get_dim_elements(..., keep_unique = FALSE)
```

Arguments

`...`	One or more structured SL4 or HAR objects containing dimension information.
`keep_unique`	Logical. If `TRUE`, returns only unique dimension elements across inputs. Default is `FALSE`.

Value

A data frame containing unique dimension elements.

Author(s)

Pattawee Puangchit

See Also

get_dim_patterns, get_var_structure

Examples

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


# Extract dimension elements from a single dataset
get_dim_elements(sl4_data1)

# Extract dimension elements from multiple datasets
get_dim_elements(sl4_data1, sl4_data2)

# Extract unique dimension elements across datasets
get_dim_elements(sl4_data1, sl4_data2, keep_unique = TRUE)
```

---

get_dim_info *Extract and Organize Dimension Metadata (Internal)*

Description

A helper function that extracts and structures dimension-related metadata from a given dimension structure. Used internally in `get_var_structure()` and `compare_var_structure()`.

Usage

```
get_dim_info(dim_info)
```

Arguments

`dim_info`	A list containing dimension metadata, including: • `dimension_string`: A textual representation of dimensions (e.g., `"REG*COMM*YEAR"`). • `dimension_names`: A character vector of dimension names. • `dimension_sizes`: A numeric vector indicating the size of each dimension.

Details

- Retrieves structured metadata for variables in SL4 and HAR datasets.
- Computes data shape and ensures consistency in dimension structures.
- Helps determine observation counts and column sizes for variable summaries.

Value

A structured list containing:

- `dimension_string`: The original dimension string.
- `dim_size`: The number of dimensions.
- `data_shape`: A formatted string representing the data shape (e.g., `"10x20x30"`).
- `col_size`: The product of all dimension sizes except the first, representing column count.
- `n_obs`: The first dimension size, typically representing the number of observations.

Author(s)

Pattawee Puangchit

See Also

`get_var_structure`, `compare_var_structure`

---

get_dim_patterns *Get Dimension Patterns from SL4 and HAR Objects*

Description

Extracts and lists unique dimension patterns (e.g., `REG*COMM`, `REG*REG*ACTS`) from one or more datasets.

Usage

```
get_dim_patterns(..., keep_unique = FALSE)
```

Arguments

`...`	One or more structured SL4 or HAR objects containing dimension information.
`keep_unique`	Logical. If `TRUE`, returns only unique dimension patterns. Default is `FALSE`.

Details

- Extracts dimension structure details from the dataset.
- If multiple datasets are provided, combines their dimension information.
- If `keep_unique = TRUE`, returns only distinct dimension patterns.

Value

A data frame containing:

- `DimPattern`: The unique dimension patterns.

Author(s)

Pattawee Puangchit

See Also

get_dim_elements, get_var_structure

Examples

```
# Import sample data:
sl4_data <- load_sl4x(system.file("extdata", "TAR10.sl4", package = "HARplus"))
sl4_data2 <- load_sl4x(system.file("extdata", "SUBT10.sl4", package = "HARplus"))

# Extract dimension patterns
get_dim_patterns(sl4_data)

# Extract only unique dimension patterns across datasets
get_dim_patterns(sl4_data, sl4_data2, keep_unique = TRUE)
```

---

get_original_pattern *Retrieve the Original Dimension Pattern (Internal)*

Description

A helper function that finds the original dimension pattern name in an SL4 or HAR dataset that matches a given pattern. Used internally in `get_data_by_dims()`.

Usage

```
get_original_pattern(pattern, data_obj, mix_patterns = FALSE)
```

Arguments

`pattern`	Character. The pattern to search for in dimension structures.
`data_obj`	An SL4 or HAR object containing dimension information.
`mix_patterns`	Logical. If `TRUE`, allows dimension order to vary when matching patterns.

Details

- Performs a case-insensitive comparison to identify matching dimension patterns.
- Supports flexible pattern matching when `mix_patterns = TRUE`, allowing dimension order to vary.
- Returns the standardized dimension pattern name as stored in the dataset.

Value

The original dimension pattern name as a character string, or `NULL` if no match is found.

Author(s)

Pattawee Puangchit

See Also

pattern_match, process_pattern, get_data_by_dims

`get_var_structure` *Get Variable Structure Summary from SL4 and HAR Objects*

Description

Generates a summary of the variables within one or more SL4 or HAR objects, listing their dimension sizes, structures, and optionally, column and observation counts.

Usage

```
get_var_structure(variables = NULL, ..., include_col_size = FALSE)
```

Arguments

`variables`	Character vector. Variable names to summarize. Use `NULL` or `"ALL"` to summarize all variables.
`...`	One or more SL4 or HAR objects created using `load_sl4x()` or `load_harx()`.
`include_col_size`	Logical. If `TRUE`, includes column and observation counts. Default is `FALSE`.

Details

- Extracts dimension structures for variables in one or more SL4 or HAR datasets.
- If `include_col_size = TRUE`, adds column and observation counts.
- Supports multiple datasets and returns results as a named list, with each dataset's summary stored separately.
- Can summarize specific variables or `"ALL"`.

Value

A named list, where each element contains a data frame with:

- `Variable`: The variable name.
- `Dimensions`: The associated dimensions.
- `DimSize`: The number of dimensions.
- `DataShape`: The shape of the data (e.g., `10x20x30`).
- `No.Col`: (Optional) The number of columns.
- `No.Obs`: (Optional) The number of observations.

Author(s)

Pattawee Puangchit

See Also

`get_dim_patterns`, `get_dim_elements`

Examples

```
# Import data sample:
sl4_data <- load_sl4x(system.file("extdata", "TAR10.sl4", package = "HARplus"))
sl4_data1 <- load_sl4x(system.file("extdata", "SUBT10.sl4", package = "HARplus"))

# Get summary for all variables in a single dataset
get_var_structure(data_obj = sl4_data)

# Get summary for specific variables
get_var_structure(c("gdp", "trade"), sl4_data)

# Include column and observation counts
get_var_structure("ALL", sl4_data, include_col_size = TRUE)

# Compare structures across multiple datasets
get_var_structure("ALL", sl4_data, sl4_data1)

# Include column and observation counts across multiple datasets
get_var_structure("ALL", sl4_data, sl4_data1, include_col_size = TRUE)
```

`group_data_by_dims` *Group Data by Dimension Patterns in SL4 or HAR Objects*

Description

Groups extracted SL4 or HAR data based on specified dimension structures and priority rules. Supports automatic renaming, merging, subtotal filtering, and structured metadata handling.

Usage

```
group_data_by_dims(
  patterns = NULL,
  ...,
  priority,
  rename_cols = NULL,
  experiment_names = NULL,
  subtotal_level = FALSE,
  auto_rename = FALSE
)
```

Arguments

`patterns`	Character vector. Dimension patterns to extract. Use `"ALL"` or `NULL` to extract all available patterns.
`...`	One or more SL4 or HAR objects loaded using `load_sl4x()` or `load_harx()`.
`priority`	Named list. Specifies priority dimension elements (`c("group_name" = c("dim1", "dim2"))`).
`rename_cols`	Named vector. Column name replacements (`c("old_name" = "new_name")`).
`experiment_names`	Character vector. Names assigned to each dataset. If `NULL`, names are inferred.

`subtotal_level` Character or logical. Determines which decomposition levels to retain:

- `"total"`: Keeps only `"TOTAL"` values.
- `"decomposed"`: Keeps only decomposed values (excludes `"TOTAL"`).
- `"all"`: Keeps all rows.
- `TRUE`: Equivalent to `"all"`, retaining both `"TOTAL"` and decomposed values.
- `FALSE`: Equivalent to `"total"`, keeping only `"TOTAL"` values.

`auto_rename` Logical. If `TRUE`, automatically renames dimensions for consistency. Default is `FALSE`.

Details

- Groups extracted variables based on dimension elements.
- Applies predefined priority rules to structure the data.
- Allows automatic renaming of dimensions (`auto_rename = TRUE`).
- Supports merging of grouped data across multiple experiments.
- Handles subtotal filtering (`subtotal_level`), controlling whether `"TOTAL"` or decomposed values are retained.

Value

A structured list of grouped data:

- A named list where each element corresponds to a dimension size group (e.g., "2D", "3D").
- Each group contains dimension-grouped data based on priority rules.
- If unmerged data exists, includes a report attribute detailing merge issues.

Author(s)

Pattawee Puangchit

See Also

`get_data_by_dims`, `get_data_by_var`, `load_sl4x`, `load_harx`

Examples

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

# Case 1: Multiple priority levels (Sector then Region) with auto_rename
priority_list <- list(
  "Sector" = c("COMM", "ACTS"),
  "Region" = c("REG")
)
grouped_data_multiple <- group_data_by_dims(
  patterns = "ALL",
  sl4_data1,
  priority = priority_list,
  auto_rename = TRUE
)
```

```
# Case 2: Single priority (Region only) with auto_rename
priority_list <- list("Region" = c("REG"))
grouped_data_single <- group_data_by_dims(
  patterns = "ALL",
  sl4_data1, sl4_data2,
  priority = priority_list,
  auto_rename = TRUE
)

# Case 3: Multiple priorities without auto_rename
priority_list <- list(
  "Sector" = c("COMM", "ACTS"),
  "Region" = c("REG")
)
grouped_data_no_rename <- group_data_by_dims(
  patterns = "ALL",
  sl4_data1,
  priority = priority_list,
  auto_rename = FALSE
)
```

`load_harplus` *Load and Process GEMPACK HAR Files (Internal)*

Description

Reads a GEMPACK HAR file and efficiently extracts structured data while maintaining compatibility with standard HAR formats. This implementation builds upon the foundational work of the **HARr** package, reorganizing the process for improved execution speed, memory management, and handling of sparse data structures.

Usage

```
load_harplus(con, coefAsname = FALSE, lowercase = TRUE, select_header = NULL)
```

Arguments

`con`	Character or connection. The file path to the HAR file or an open binary connection.
`coefAsname`	Logical. If `TRUE`, replaces four-letter headers with coefficient names when available. Default is `FALSE`.
`lowercase`	Logical. If `TRUE`, converts all string values to lowercase. Default is `TRUE`.
`select_header`	Character vector. Specific headers to extract; if `NULL`, reads all headers.

Details

- **Efficient File Reading:** Reads large HAR files in chunks for better performance.
- **Optimized Memory Usage:** Reduces unnecessary allocations and improves cleanup.
- **Streamlined Header Processing:** Ensures accurate extraction of dimension metadata.
- **Supports Sparse Data Structures:** Handles `RESPSE` and `REFULL` headers efficiently.

Supported HAR Header Types:

- 1CFULL: Character headers
- 2IFULL: Integer headers
- 2RFULL: Real headers
- REFULL: Real headers with extended metadata
- RESPSE: Sparse real headers

Value

A structured list where:

- Each element corresponds to a header in the HAR file.
- Names are either header names or coefficient names (if coefAsname = TRUE).
- Data maintains its original dimensions and attributes.

Author(s)

Pattawee Puangchit

See Also

load_sl4x, load_harx

---

load_harx *Load and Process HAR Files with Header Selection*

Description

Reads a GEMPACK HAR file and extracts structured data while maintaining compatibility with standard HAR formats. Provides flexibility in naming conventions and header selection.

Usage

```
load_harx(
  file_path,
  coefAsname = FALSE,
  lowercase = FALSE,
  select_header = NULL
)
```

Arguments

file_path	Character. The file path to the HAR file.
coefAsname	Logical. If TRUE, replaces four-letter headers with coefficient names when available. Default is FALSE.
lowercase	Logical. If TRUE, converts all variable names to lowercase. Default is FALSE.
select_header	Character vector. Specific headers to read; if NULL, all headers are read. Example: select_header = c("A", "E1").

Details

- Uses `load_harplus()` internally for efficient HAR file reading.
- Allows optional conversion of variable names to lowercase (`lowercase = TRUE`).
- Supports coefficient-based naming (`coefAsname = TRUE`).
- Enables selective header extraction via `select_header = c("A", "E1")`.
- Returns structured data with explicit dimension names and sizes.

Value

A structured list containing:

- `data`: Extracted HAR variable data stored as matrices, arrays, or vectors.
- `dimension_info`: A list with:
  - `dimension_string`: A textual representation of dimensions (e.g., "REG*COMM*YEAR").
  - `dimension_names`: The names of each dimension.
  - `dimension_sizes`: The size of each dimension.

Author(s)

Pattawee Puangchit

See Also

`load_sl4x`, `get_data_by_var`, `get_data_by_dims`

Examples

```
# Path to example files
har_path <- system.file("extdata", "TAR10-WEL.har", package = "HARplus")

# Basic loading
har_data <- load_harx(har_path)

# Load with coefficient names
har_data_coef <- load_harx(har_path, coefAsname = TRUE)

# Load with lowercase names
har_data_lower <- load_harx(har_path, lowercase = TRUE)

# Load specific headers
har_selected <- load_harx(har_path, select_header = c("A", "E1"))

# Load with multiple options
har_combined <- load_harx(har_path,
                          coefAsname = TRUE,
                          lowercase = TRUE,
                          select_header = c("A", "E1"))
```

load_sl4x *Load and Process SL4 Files with Enhanced Options*

Description

Reads an SL4 file and processes its structured data into an enhanced SL4 object. Extracts structured variable information, dimensions, and handles subtotal columns.

Usage

```
load_sl4x(file_path, lowercase = FALSE, select_header = NULL)
```

Arguments

file_path	Character. The full path to the SL4 file to be read.
lowercase	Logical. If TRUE, converts all variable names to lowercase. Default is FALSE.
select_header	Character vector. Specific headers to extract; if NULL, all headers are read.

Details

- Uses load_harplus() internally for optimized SL4 file reading.
- Extracts variable names, dimension structures, and metadata.
- Converts variable names to lowercase if lowercase = TRUE.
- Allows the selection of specific headers using select_header.
- Returns structured data with explicit dimension names and sizes.

Value

A structured list containing:

- data: Extracted SL4 variable data, stored as arrays or matrices.
- dimension_info: A list with:
  - dimension_string: A textual representation of dimensions (e.g., "REG*COMM*YEAR").
  - dimension_names: The names of each dimension.
  - dimension_sizes: The size of each dimension.

Author(s)

Pattawee Puangchit

See Also

load_harx, get_data_by_var, get_data_by_dims

Examples

```
# Path to example files
sl4_path <- system.file("extdata", "TAR10.sl4", package = "HARplus")

# Basic loading
sl4_data <- load_sl4x(sl4_path)

# Load with lowercase names
sl4_data_lower <- load_sl4x(sl4_path, lowercase = TRUE)

# Load specific headers
sl4_selected <- load_sl4x(sl4_path, select_header = c("qo", "qgdp"))
```

`pattern_match` *Match Patterns with Optional Mixing (Internal)*

Description

Compares two patterns to determine if they match, with an option to allow flexible dimension order.

Usage

```
pattern_match(pattern1, pattern2, mix_patterns = FALSE)
```

Arguments

`pattern1`	A character string representing the first pattern.
`pattern2`	A character string representing the second pattern.
`mix_patterns`	Logical; if `TRUE`, allows dimension order to be ignored during comparison.

Details

- Performs case-insensitive pattern matching.
- If `mix_patterns = TRUE`, allows patterns to match even if dimensions are in different order.

Value

Logical; `TRUE` if the patterns match, `FALSE` otherwise.

Author(s)

Pattawee Puangchit

See Also

`get_original_pattern`, `process_pattern`, `get_data_by_dims`

`pivot_data` *Pivot Data from SL4 or HAR Objects*

Description

Transforms long-format SL4 or HAR data into wide format by pivoting selected columns. Supports both single data frames and nested lists.

Usage

```
pivot_data(data_obj, pivot_cols, name_repair = "unique")
```

Arguments

`data_obj`	A list or data frame. The SL4 or HAR data to pivot.
`pivot_cols`	Character vector. Column names to use as pivot keys.
`name_repair`	Character. Method for handling duplicate column names (`"unique"`, `"minimal"`, `"universal"`). Default is `"unique"`.

Details

- Uses `tidyr::pivot_wider()` internally to reshape data.
- Allows multiple columns to be pivoted simultaneously.
- Recursively processes nested lists, ensuring all data frames are transformed.

Value

A transformed data object where the specified `pivot_cols` are pivoted into wide format.

Author(s)

Pattawee Puangchit

See Also

`get_data_by_var`, `get_data_by_dims`

Examples

```
# Import sample data:
sl4_data <- load_sl4x(system.file("extdata", "TAR10.sl4", package = "HARplus"))

# Extract multiple variables
data_multiple <- get_data_by_var(c("qo", "qxs"), sl4_data)

# Pivot a single column
pivoted_data <- pivot_data(data_multiple, pivot_cols = "REG")

# Pivot multiple columns
pivoted_data_multi <- pivot_data(data_multiple, pivot_cols = c("REG", "COMM"))
```

pivot_data_hierarchy *Create Hierarchical Pivot Table from SL4 or HAR Objects*

Description

Creates hierarchical pivot tables from structured SL4 or HAR data, with optional Excel export. Supports both single data frames and nested lists, preserving dimension hierarchies.

Usage

```
pivot_data_hierarchy(
  data_obj,
  pivot_cols,
  name_repair = "unique",
  export = FALSE,
  file_path = NULL,
  xlsx_filename = NULL
)
```

Arguments

data_obj	A list or data frame. The SL4 or HAR data to pivot.
pivot_cols	Character vector. Column names to use as pivot keys in order of hierarchy.
name_repair	Character. Method for handling duplicate column names ("unique", "minimal", "universal"). Default is "unique".
export	Logical. If TRUE, exports result to Excel. Default is FALSE.
file_path	Character. Required if export = TRUE. The path for Excel export.
xlsx_filename	Character. The name for the Excel file when using multi_sheet_xlsx. If NULL, uses the name of the dataset. Default is NULL.

Details

- Transforms data into hierarchical pivot format with nested column headers.
- Supports multiple pivot columns in specified order (e.g., REG > COMM).
- Handles both single data frames and nested list structures.
- Optional direct export to Excel with formatted hierarchical headers.
- Uses efficient data processing with tidyr and openxlsx.

Value

A pivoted data object with hierarchical structure:

- If input is a data frame: Returns a hierarchical pivot table.
- If input is a list: Returns a nested list of hierarchical pivot tables.
- If export = TRUE: Invisibly returns the pivoted object after Excel export.

Author(s)

Pattawee Puangchit

See Also

pivot_data

Examples

```
## Not run:
# Import sample data:
sl4_data <- load_sl4x(system.file("extdata", "TAR10.sl4", package = "HARplus"))

# Extract data
data_multiple <- get_data_by_var(c("qo", "pca"), sl4_data)

# Create hierarchical pivot without export
pivot_hier <- pivot_data_hierarchy(data_multiple,
                                   pivot_cols = c("REG", "COMM"))

# Create and export to Excel in one step
pivot_export <- pivot_data_hierarchy(data_multiple,
                                     pivot_cols = c("REG", "COMM"),
                                     export = TRUE,
                                    file_path = file.path(tempdir(), "pivot_output.xlsx"))

## End(Not run)
```

---

process_decomp_level *Process Decomposition Levels in Data Frames (Internal)*

Description

A helper function that filters data based on decomposition levels in the "Subtotal" column. Used internally in get_data_by_var(), get_data_by_dims(), and group_data_by_dims().

Usage

```
process_decomp_level(df, subtotal_level)
```

Arguments

df	A data frame containing a "Subtotal" column.
subtotal_level	Character or logical. Determines which values to retain:

- "total": Keeps only "TOTAL" values.
- "decomposed": Keeps only decomposed values (excludes "TOTAL").
- "all": Keeps all rows.
- TRUE: Equivalent to "all" (keeps both "TOTAL" and decomposed values).
- FALSE: Equivalent to "total" (keeps only "TOTAL" values, removing decomposed components).

Details

- If `subtotal_level = "total"`, **keeps only `"TOTAL"` values**, removing all decomposed components.
- If `subtotal_level = "decomposed"`, **keeps only decomposed components**, removing `"TOTAL"`.
- If `subtotal_level = "all"`, **keeps both `"TOTAL"` and decomposed values** (no filtering).
- If `subtotal_level = TRUE` (logical), it is **equivalent to `"all"`**, meaning all values are kept.
- If `subtotal_level = FALSE` (logical), it is **equivalent to `"total"`**, meaning only `"TOTAL"` values are kept, and decomposed components are removed.
- Filtering is applied **only when both `"TOTAL"` and decomposed values exist**.

Value

A filtered data frame based on the specified decomposition level.

Author(s)

Pattawee Puangchit

See Also

`get_data_by_var`, `get_data_by_dims`, `group_data_by_dims`

`process_export_report` *Process and Export Report (Internal)*

Description

Generates a summary report of variables and their corresponding output files from structured SL4 or HAR data. The report is saved as an Excel file.

Usage

```
process_export_report(data, output_path, prefix = "", data_name = "data")
```

Arguments

`data`	A structured SL4 or HAR object or a nested list of data frames.
`output_path`	A character string specifying the output directory or file path.
`prefix`	A character string to prepend to the exported filenames.

Details

- Extracts unique variable names and their associated export filenames.
- Handles nested list structures by recursively traversing them.
- The output report is saved as an Excel file named `"Report_<data_name>.xlsx"`.

Author(s)

Pattawee Puangchit

See Also

export_data

---

process_pattern *Extract and Process Pattern-Matched Variables (Internal)*

Description

A helper function that extracts and processes variables matching a specified pattern within an SL4 or HAR data object. Used internally in `get_data_by_dims()`.

Usage

```
process_pattern(pattern, data_obj, exp_name, pattern_mix = FALSE)
```

Arguments

`pattern`	Character. The pattern to match against dimension structures.
`data_obj`	An SL4 or HAR object containing dimension information and data.
`exp_name`	Character. The experiment name assigned to the extracted data.
`pattern_mix`	Logical. If `TRUE`, allows pattern matching to ignore dimension order.

Details

- Searches for variables whose dimension structures match the given pattern.
- Supports flexible pattern matching when `pattern_mix = TRUE`, allowing dimension order to vary.
- Extracted data is converted into a tidy format, preserving dimension structures.
- Standardizes `"Subtotal"` column naming for consistency across datasets.
- Ensures only non-empty extracted variables are retained.

Value

A data frame containing processed data for the matching pattern, or `NULL` if no matches are found.

Author(s)

Pattawee Puangchit

See Also

get_original_pattern, pattern_match, get_data_by_dims

`rename_col` *Rename Columns in a Data Frame (Internal)*

Description

A helper function that renames columns in a data frame based on a specified mapping. Used internally in `get_var_structure()`, `get_data_by_dims()`, and `rename_dims()`.

Usage

```
rename_col(df, rename_cols)
```

Arguments

`df`	A data frame containing columns to be renamed.
`rename_cols`	A named vector where names are existing column names, and values are the corresponding new names.

Details

- Replaces column names according to the provided `rename_cols` mapping.
- Ensures no duplicate column names by appending numerical suffixes when necessary.
- Helps standardize column names across SL4 and HAR datasets.

Value

A modified data frame with renamed columns.

Author(s)

Pattawee Puangchit

See Also

`get_var_structure`, `get_data_by_dims`, `rename_dims`

`rename_dims` *Rename Dimensions in SL4 or HAR Objects*

Description

Renames dimension and list names in structured SL4 or HAR objects.

Usage

```
rename_dims(data_obj, mapping_df, rename_list_names = FALSE)
```

Arguments

data_obj	A structured SL4 or HAR object.
mapping_df	A two-column data frame where the first column (old) contains the current names, and the second column (new) contains the new names.
rename_list_names	Logical. If TRUE, renames list element names. Default is FALSE.

Details

- Replaces old dimension names with new ones as specified in mapping_df.
- If rename_list_names = TRUE, renames list element names as well.
- Ensures consistency across SL4 and HAR datasets.

Value

The modified SL4 or HAR object with updated dimension names and, optionally, updated list names.

Author(s)

Pattawee Puangchit

See Also

get_data_by_var, get_data_by_dims

Examples

```
# Import sample data:
sl4_data <- load_sl4x(system.file("extdata", "TAR10.sl4", package = "HARplus"))

# Define a renaming map
mapping_df <- data.frame(
  old = c("REG", "COMM"),
  new = c("Region", "Commodity")
)

# Rename columns in the dataset
rename_dims(sl4_data, mapping_df)

# Rename both columns and list names
rename_dims(sl4_data, mapping_df, rename_list_names = TRUE)
```

Index