Package 'matriz'

May 8, 2026

Type Package

Title Literature Matrix Synthesis Tools for Epidemiology and Health Science Research

Version 1.0.1

Maintainer JP Monteagudo <jpmonteagudo2014@gmail.com>

Description An easy-to-use workflow that provides tools to create, update and fill literature matrices commonly used in research, specifically epidemiology and health sciences research. The project is born out of need as an easy–to–use tool for my research methods classes.

License AGPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

URL https://github.com/jpmonteagudo28/matriz

BugReports https://github.com/jpmonteagudo28/matriz/issues

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports dplyr, readr, readxl, rlang, stringr, writexl

Depends R (>= 4.1.0)

NeedsCompilation no

Author JP Monteagudo [aut, cre, cph] (ORCID: <https://orcid.org/0009-0003-6465-6658>)

Repository CRAN

Date/Publication 2025-02-05 18:30:01 UTC

Contents

add_batch_record 2
add_empty_row 3
add_record 3
delete_record 4

export_matrix 5
import_matrix 6
init_matrix 8
matriz_message 9
matriz_names 9
merge_matrix 10
process_batch_citation 11
process_citation 12
search_record 12
truncate 13
update_record 14
validate_columns 15

Index **17**

`add_batch_record` *Add Multiple Records to a literature matrix*

Description

Adds one or more records to a literature matrix at a specified position. Records can be provided as lists or data frames, and can be inserted before or after specific rows.

Usage

```
add_batch_record(.data, ..., .before = NULL, .after = NULL)
```

Arguments

`.data`	A data frame to which records will be added
`...`	One or more records to add. Each record can be either: • A list with the same length as the number of columns in '.data' • A data frame with the same column structure as '.data'
`.before`	Row number before which to insert the new records. If NULL (default), and '.after' is also NULL, records are appended to the end.
`.after`	Row number after which to insert the new records. If NULL (default), and '.before' is also NULL, records are appended to the end.

Value

A data frame with the new records added at the specified position

Examples

```
# Create sample data frame
df <- data.frame(
  name = c("John", "Jane"),
  age = c(25, 30)
)

# Add a single record as a list
df <- add_batch_record(df, list(name = "Bob", age = 35))

# Add multiple records as data frames
new_records <- data.frame(
  name = c("Alice", "Charlie"),
  age = c(28, 40)
)
df <- add_batch_record(df, new_records, .before = 2)
```

---

add_empty_row *Add an Empty Row to a Data Frame*

Description

Adds a single row of NA values to a data frame

Usage

```
add_empty_row(.data)
```

Arguments

.data	A data frame to which an empty row will be added

Value

Modified data frame with an additional empty row

---

add_record *Add a Record to a Data Frame*

Description

Adds a new row to a data frame at a specified position

Usage

```
add_record(.data, ..., .before = NULL, .after = NULL)
```

Arguments

.data	A data frame to which a record will be added
...	New record to be added (vector, list, or data frame)
.before	Optional. Row number before which to insert the new record
.after	Optional. Row number after which to insert the new record

Value

Modified data frame with the new record inserted

Examples

```
df <- data.frame(x = 1:3, y = 4:6)
add_record(df, c(4, 7))
add_record(df, c(4, 7), .before = 2)
```

---

delete_record *Delete Records from a Data Frame*

Description

Deletes specific rows from a data frame or clears the entire data frame by leveraging the ‘truncate‘ function. If no position is provided, it will issue a message and either return the unchanged data or use ‘truncate‘ to empty the data frame, depending on additional arguments.

Usage

```
delete_record(.data, position = NULL, ...)
```

Arguments

.data	A data frame from which records will be deleted.
position	A numeric vector specifying the row positions to be deleted. If ‘NULL‘, behavior is determined by the number of rows in the data frame and additional arguments passed to the ‘truncate‘ function.
...	Additional arguments passed to the ‘truncate‘ function. Specifically, the ‘keep_rows‘ argument can be used to decide whether non-NA cells in the data frame are cleared when truncating.

Details

- If ‘position‘ is ‘NULL‘ and the data frame has more than one row, a message is issued, and no records are deleted. - If ‘position‘ is a numeric vector, the specified rows are deleted using ‘dplyr::slice()‘. - If ‘position‘ is empty or invalid (e.g., not numeric), the function stops with an appropriate error message. - When no rows remain after deletion, the function calls ‘truncate‘ to handle the data frame, with behavior controlled by the ‘keep_rows‘ argument passed through ‘...‘.

Value

A modified data frame with the specified rows removed. If ‘position‘ is ‘NULL‘, the function either returns the original data frame or an empty data frame, based on the ‘keep_rows‘ argument in the ‘truncate‘ function.

Examples

```
df <- data.frame(A = 1:5, B = letters[1:5])

# Delete a specific row
delete_record(df, position = 2)

# Delete multiple rows
delete_record(df, position = c(2, 4))

# Use truncate to clear the data frame
delete_record(df, position = NULL, keep_rows = FALSE)

# Keep non-NA cells but empty rows
delete_record(df, position = NULL, keep_rows = TRUE)
```

---

export_matrix *Export a Data Matrix to Various File Formats*

Description

This function exports a data frame to a specified file format, including CSV, TSV, RDS, XLSX, and TXT. If the format is not provided, it is inferred from the file extension.

Usage

```
export_matrix(
  .data,
  file,
  format = NULL,
  drop_extra = FALSE,
  extra_columns = NULL,
  silent = FALSE,
  ...
)
```

Arguments

.data	A data frame or tibble to be exported.
file	A character string specifying the file name and path.

format	A character string specifying the file format. If ‘NULL‘, the format is inferred from the file extension. Supported formats: ‘"csv"‘, ‘"tsv"‘, ‘"rds"‘, ‘"xlsx"‘, ‘"txt"‘.
drop_extra	Logical. If ‘TRUE‘, removes columns not listed in ‘extra_columns‘ before exporting. Default is ‘FALSE‘.
extra_columns	A character vector specifying additional columns to retain if ‘drop_extra = TRUE‘. Default is ‘NULL‘.
silent	Logical. If ‘TRUE‘, suppresses messages. Default is ‘FALSE‘.
...	Additional arguments passed to the underlying export functions (‘write.csv‘, ‘writexl::write_xlsx‘, etc.).

Value

Exports the data to a file and returns ‘NULL‘ invisibly.

`import_matrix` *This function imports a matrix (data frame) from various file formats (CSV, TSV, RDS, XLSX, XLS, TXT) and ensures it contains the required columns. It also allows the user to control whether extra columns should be dropped or kept.*

Description

This function imports a matrix (data frame) from various file formats (CSV, TSV, RDS, XLSX, XLS, TXT) and ensures it contains the required columns. It also allows the user to control whether extra columns should be dropped or kept.

Usage

```
import_matrix(
  path,
  format = NULL,
  drop_extra = FALSE,
  extra_columns = NULL,
  remove_dups = TRUE,
  silent = FALSE,
  ...
)
```

Arguments

path	A character string specifying the path to the file to be imported.
format	A character string specifying the file format. If not provided, the format is automatically detected based on the file extension. Supported formats: "csv", "tsv", "rds", "xlsx", "xls", "txt".

drop_extra	A logical value indicating whether extra columns (not in the list of required columns) should be dropped. Default is 'FALSE'.
extra_columns	A character vector of column names that are allowed in addition to the required columns. By default, no extra columns are allowed.
remove_dups	A logical value indicating whether to remove duplicate columns before merging. Default is 'TRUE'.
silent	A logical value indicating whether to suppress messages. Default is 'FALSE'.
...	Additional arguments passed to the specific file-reading functions (e.g., 'read.csv', 'read.delim', 'readRDS', 'readxl::read_xlsx', 'readxl::read_xls', 'read.table'). Refer to the documentation of the corresponding read function for the list of valid arguments.

Details

The matrix includes the following predefined columns:

- 'year': Numeric. Year of publication. - 'citation': Character. Citation or reference details. - 'keywords': Character. Keywords or tags for the study. - 'profession': Character. Profession of the study participants or target audience. - 'electronic': Logical. Indicates whether the study is available electronically. - 'purpose': Character. Purpose or objective of the study. - 'study_design': Character. Study design or methodology. - 'outcome_var': Character. Outcome variables measured in the study. - 'predictor_var': Character. Predictor variables considered in the study. - 'sample': Numeric. Sample size. - 'dropout_rate': Numeric. Dropout or attrition rate. - 'setting': Character. Study setting (e.g., clinical, educational). - 'inclusion_criteria': Character. Inclusion criteria for participants. - 'ethnicity': Character. Ethnic background of participants. - 'age': Numeric. Age of participants. - 'sex': Factor. Sex of participants. - 'income': Factor. Income level of participants. - 'education': Character. Educational background of participants. - 'measures': Character. Measures or instruments used for data collection. - 'analysis': Character. Analytical methods used. - 'results': Character. Summary of results or findings. - 'limitations': Character. Limitations of the study. - 'implications': Character. Implications or recommendations from the study. - 'ethical_concerns': Character. Ethical concerns addressed in the study. - 'biases': Character. Potential biases in the study. - 'notes': Character. Additional notes or observations.

Extra columns beyond the required ones are handled via the 'extra_columns' argument. If the 'drop_extra' argument is set to 'TRUE', extra columns will be removed. If 'drop_extra' is 'FALSE', extra columns will remain in the imported data, and a message will be shown.

The '...' argument allows you to pass additional parameters directly to the read functions. For instance: - For 'read.csv', '...' could include 'header = TRUE', 'sep = ","', or 'stringsAsFactors = FALSE'. - For 'read.delim', '...' could include 'header = TRUE', 'sep ', or 'stringsAsFactors = FALSE'. - For 'readRDS', '...' could include 'refhook = NULL'. - For 'readxl::read_xlsx', '...' could include 'sheet = 1' or 'col_names = TRUE'. - For 'readxl::read_xls', '...' could include 'sheet = 1' or 'col_Names = TRUE'. - For 'read.table', '...' could include 'header = TRUE', 'sep', or 'stringsAsFactors = FALSE'.

Value

A data frame containing the imported matrix, with the required columns and any allowed extra columns.

`init_matrix` *Initialize a Literature Review Matrix*

Description

Creates a standardized data frame for systematic literature review with predefined columns, allowing the addition of custom columns if needed.

Usage

```
init_matrix(...)
```

Arguments

`...`	Optional. Additional column names (as character strings) to be appended to the matrix.

Details

The matrix includes the following predefined columns: - ‘year‘: Numeric. Year of publication. - ‘citation‘: Character. Citation or reference details. - ‘keywords‘: Character. Keywords or tags for the study. - ‘profession‘: Character. Profession of the study participants or target audience. - ‘electronic‘: Logical. Indicates whether the study is available electronically. - ‘purpose‘: Character. Purpose or objective of the study. - ‘study_design‘: Character. Study design or methodology. - ‘outcome_var‘: Character. Outcome variables measured in the study. - ‘predictor_var‘: Character. Predictor variables considered in the study. - ‘sample‘: Numeric. Sample size. - ‘dropout_rate‘: Numeric. Dropout or attrition rate. - ‘setting‘: Character. Study setting (e.g., clinical, educational). - ‘inclusion_criteria‘: Character. Inclusion criteria for participants. - ‘ethnicity‘: Character. Ethnic background of participants. - ‘age‘: Numeric. Age of participants. - ‘sex‘: Factor. Sex of participants. - ‘income‘: Factor. Income level of participants. - ‘education‘: Character. Educational background of participants. - ‘measures‘: Character. Measures or instruments used for data collection. - ‘analysis‘: Character. Analytical methods used. - ‘results‘: Character. Summary of results or findings. - ‘limitations‘: Character. Limitations of the study. - ‘implications‘: Character. Implications or recommendations from the study. - ‘ethical_concerns‘: Character. Ethical concerns addressed in the study. - ‘biases‘: Character. Potential biases in the study. - ‘notes‘: Character. Additional notes or observations.

Custom columns can also be added by passing their names via the ‘...‘ argument.

Value

A data frame with predefined columns for literature review analysis.

Examples

```
# Create a basic literature review matrix
lit_matrix <- init_matrix()
```

matriz_message *Display package version for* **matriz**

Description

`matriz_message()` produces a message about the package version and the version of R making use of this package.

Usage

```
matriz_message()
```

Value

`dmatriz_message()` returns a message about the install version of **matriz**.

Author(s)

JP Monteagudo

Examples

```
matriz_message()
```

matriz_names *Retrieve Column Classes from deafult literature matrix.*

Description

This function calls `init_matrix()` to obtain a matrix or data frame, then extracts the class of each column. It returns a data frame containing the class information for each column.

Usage

```
matriz_names(...)
```

Arguments

`...`	extra arguments to pass as column names for the literature matrix

Details

The purpose of this function is to provide the user with a quick way to check the default names and classes as the matrix is being filled instead of having to type ‘str(init_matrix())‘ every time the user forgets a category in the default matrix.

Value

A data frame with one column named `class` that lists the class of each column from the matrix or data frame returned by `init_matrix()`.

Examples

```
matriz_names()
```

`merge_matrix` *Merge Two literature matrices by Common Columns*

Description

This function merges two literature matrices based on specified key columns, with options for full or inner joins and duplicate column removal.

Usage

```
merge_matrix(
  .data,
  .data2,
  by = NULL,
  all = FALSE,
  remove_dups = TRUE,
  suffixes = c(".x", ".y"),
  silent = FALSE
)
```

Arguments

`.data`	A data frame to be merged.
`.data2`	A second data frame to be merged with '.data'.
`by`	A character vector specifying the column(s) to merge by. Must exist in both data frames.
`all`	A logical value indicating whether to perform a full join ('TRUE') or an inner join ('FALSE', default).
`remove_dups`	A logical value indicating whether to remove duplicate columns before merging. Default is 'TRUE'.
`suffixes`	A character vector of length 2 specifying suffixes to apply to overlapping column names from '.data' and '.data2', respectively. Default is 'c(".x", ".y")'.
`silent`	A logical value indicating whether to suppress messages about duplicate column removal. Default is 'FALSE'.

Details

The function first ensures that '.data' and '.data2' are valid data frames and checks that the 'by' columns exist in both. If 'remove_dups = TRUE', duplicate columns are removed before merging. The function then performs either a full or inner join using 'dplyr::full_join()' or 'dplyr::inner_join()', respectively.

Value

A merged data frame with specified join conditions applied.

Examples

```
df1 <- data.frame(id = c(1, 2, 3), value1 = c("A", "B", "C"))
df2 <- data.frame(id = c(2, 3, 4), value2 = c("X", "Y", "Z"))

# Inner join (default)
merge_matrix(df1, df2, by = "id")

# Full join
merge_matrix(df1, df2, by = "id", all = TRUE)

# Remove duplicate columns before merging
df3 <- data.frame(id = c(1, 2, 3), value1 = c("A", "B", "C"), extra = c(1, 2, 3))
df4 <- data.frame(id = c(2, 3, 4), value2 = c("X", "Y", "Z"), extra = c(4, 5, 6))
merge_matrix(df3, df4, by = "id", remove_dups = TRUE)
```

---

process_batch_citation

Process Multiple BibTeX Citations and Update Literature Matrix

---

Description

Reads multiple BibTeX citations from files and updates the corresponding rows in a literature matrix with formatted citations, keywords, and years.

Usage

```
process_batch_citation(.data, citations, where = NULL)
```

Arguments

`.data`	A data frame containing at least three columns: • citation: Character column for formatted citations • keywords: List column for citation keywords • year: Numeric column for publication years
`citations`	Character vector of file paths to BibTeX citation files
`where`	Numeric vector indicating which rows to update. If NULL (default), all rows will be updated.

Value

A data frame with updated citation information in the specified rows

See Also

format_batch_ama_citation, parse_batch_citation

---

process_citation *Process a Citation Record*

Description

Takes a record list and a citation string, processes the citation into AMA format, and updates the record with the formatted citation, keywords, and year.

Usage

```
process_citation(.record, citation)
```

Arguments

.record	A list containing the record to be updated
citation	A character string containing a BibTeX citation

Value

An updated list containing the original record with added fields:

citation	The formatted AMA citation
keywords	A vector of keywords from the citation
year	The publication year

---

search_record *Search and Filter Records in a literature matrix*

Description

Filters a literature matrix based on a specified condition, with the option to restrict the search to a specific column. The function supports both column names and numeric indices for column selection.

Usage

```
search_record(.data, column = NULL, where = NULL)
```

Arguments

.data	A data frame to search within.
column	Optional. The column to search in, specified either by name or numeric index. If NULL (default), the search is performed across all columns.
where	A logical expression that defines the search condition. Must evaluate to a logical vector of the same length as the number of rows in '.data'.

Value

A filtered data frame containing only the rows that match the search condition. If a specific column was selected, only that column is returned.

Examples

```
df <- data.frame(
  id = 1:5,
  name = c("John", "Jane", "Bob", "Alice", "John"),
  age = c(25, 30, 35, 28, 40)
)

# Search across all columns where age > 30
search_record(df, where = age > 30)

# Search only in the name column for "John"
search_record(df, column = "name", where = name == "John")

# Search using column index
search_record(df, column = 2, where = name == "Jane")
```

---

truncate *Truncate a Data Frame or Matrix*

Description

Remove all rows from a literature matrix but preserve the general structure. Mimics SQL's TRUNCATE operation by clearing data while preserving structure.

Usage

```
truncate(.data, keep_rows = FALSE)
```

Arguments

.data	A data frame or matrix to be truncated
keep_rows	Logical. If TRUE, replaces non-NA values with NA instead of removing all data

Value

An empty data frame or matrix with the same structure as the input

Examples

```
# Completely empty a data frame
df <- data.frame(x = 1:3, y = 4:6)
truncate(df)

# Replace non-NA values with NA while keeping structure
truncate(df, keep_rows = TRUE)
```

---

update_record *Update Rows in a Data Frame Based on a Condition*

Description

Modifies the values in a specified column of a data frame for rows that meet a given condition.

Usage

```
update_record(.data, column = NULL, where = NULL, set_to = NULL, ...)
```

Arguments

.data	A data frame. The dataset to modify.
column	A column in the data frame to update. Can be specified as a column name, index, or unquoted column symbol.
where	A condition that determines which rows to update. Must evaluate to a logical vector of the same length as the number of rows in '.data'.
set_to	The value to assign to the rows in the specified column where the 'where' condition is 'TRUE'.
...	Additional arguments (currently unused, reserved for future use).

Details

This function updates values in a specified column of a data frame for rows that satisfy the given condition. The 'column' parameter can be provided as: - A numeric column index (e.g., '2'). - A column name (e.g., '"value"'). - An unquoted column symbol (e.g., 'value').

Value

The modified data frame with updated values.

Examples

```
# Example data frame
df <- data.frame(
  id = 1:5,
  value = c(10, 20, 30, 40, 50)
)

# Update rows where id > 3
updated_df <- update_record(df, column = value, where = id > 3, set_to = 100)
print(updated_df)

# Using column as a string
updated_df <- update_record(df, column = "value", where = id == 2, set_to = 99)
print(updated_df)
```

---

validate_columns *Validate and Clean Imported Data Matrix*

Description

This function ensures that the imported data contains all required columns, optionally removes unwanted extra columns, and provides informative messages about the dataset's structure.

Usage

```
validate_columns(
  data,
  extra_columns = NULL,
  drop_extra = FALSE,
  silent = FALSE
)
```

Arguments

data	A data frame containing the imported matrix.
extra_columns	A character vector of allowed additional columns beyond the required ones. Defaults to NULL.
drop_extra	A logical value indicating whether to remove extra columns that are not in 'extra_columns'. Defaults to FALSE.
silent	A logical value indicating whether to suppress messages. Defaults to FALSE.

Details

The function checks whether all required columns are present in the data. If any required columns are missing, it stops execution and informs the user.

It also identifies extra columns beyond the required set and compares them against the allowed 'extra_columns'. If 'drop_extra = TRUE', it removes any extra columns not listed in 'extra_columns'. If 'drop_extra = FALSE', it retains the extra columns but issues a message unless 'silent = TRUE'.

Value

A cleaned data frame with required columns intact and, optionally, extra columns removed.

Note

The function assumes that column names in ‘data‘ are correctly formatted and case-sensitive.

Index

add_batch_record, 2
add_empty_row, 3
add_record, 3

delete_record, 4

export_matrix, 5

format_batch_ama_citation, *12*

import_matrix, 6
init_matrix, 8

matriz_message, 9
matriz_names, 9
merge_matrix, 10

parse_batch_citation, *12*
process_batch_citation, 11
process_citation, 12

search_record, 12

truncate, 13

update_record, 14

validate_columns, 15