Package 'Rmonize'

July 21, 2025

Type Package

Title Tools for Data Harmonization

Version 2.0.0

Maintainer Guillaume Fabre <guijoseph.fabre@gmail.com>

Description Integrated tools to support rigorous and well documented data harmonization based on Maelstrom Research guidelines. The package includes functions to assess and prepare input elements, apply specified processing rules to generate harmonized datasets, validate data processing and identify processing errors, and document and summarize harmonized outputs. The harmonization process is defined and structured by two key user-generated documents: the DataSchema (specifying the list of harmonized variables to generate across datasets) and the Data Processing Elements (specifying the input elements and processing algorithms to generate harmonized variables in DataSchema formats). The package was developed to address key challenges of retrospective data harmonization in epidemiology (as described in Fortier I and al. (2017) <doi:10.1093/ije/dyw075>) but can be used for any data harmonization initiative.

License GPL-3

LazyData true

Depends R (>= 3.5)

Imports dplyr (>= 1.1.0), rlang, stringr, tidyr, crayon, haven, utils, fs, fabR (>= 2.0.0), madshapR (>= 2.0.0)

Suggests janitor, car, lubridate, knitr

URL https://github.com/maelstrom-research/Rmonize/

BugReports https://github.com/maelstrom-research/Rmonize/issues

RoxygenNote 7.2.3

VignetteBuilder knitr

Encoding UTF-8

Language en-US

NeedsCompilation no

Author Guillaume Fabre [aut, cre] (ORCID:
<https://orcid.org/0000-0002-0124-9970>),
Maelstrom Research [aut, fnd, cph]

Repository CRAN

Date/Publication 2025-06-30 18:50:02 UTC

Contents

as_dataschema . . . 3
as_dataschema_mlstr . . . 4
as_dataset . . . 5
as_data_dict . . . 5
as_data_proc_elem . . . 5
as_dossier . . . 6
as_harmonized_dossier . . . 6
bookdown_open . . . 8
dataschema_evaluate . . . 8
dataschema_extract . . . 9
dataset_evaluate . . . 10
dataset_summarize . . . 11
dataset_visualize . . . 11
data_dict_apply . . . 11
data_dict_evaluate . . . 11
data_dict_extract . . . 12
data_proc_elem_evaluate . . . 12
dossier_create . . . 13
dossier_evaluate . . . 13
dossier_summarize . . . 13
harmonized_dossier_evaluate . . . 14
harmonized_dossier_summarize . . . 15
harmonized_dossier_visualize . . . 16
harmo_process . . . 18
is_dataschema . . . 20
is_dataschema_mlstr . . . 21
is_data_proc_elem . . . 22
pooled_harmonized_dataset_create . . . 23
Rmonize_examples . . . 24
Rmonize_templates . . . 25
Rmonize_website . . . 26
show_harmo_error . . . 27

Index **28**

`as_dataschema` *Validate and coerce as a DataSchema object*

Description

Checks if an object is a valid DataSchema and returns it with the appropriate `Rmonize::class` attribute. This function mainly helps validate inputs within other functions of the package but could be used separately to ensure that an object has an appropriate structure.

Usage

```
as_dataschema(object, as_dataschema_mlstr = FALSE)
```

Arguments

`object`	A potential DataSchema object to be coerced.
`as_dataschema_mlstr`	Whether the output DataSchema should be coerced with specific format restrictions for compatibility with other Maelstrom Research software. FALSE by default.

Details

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function. In 'Variables' the `name` column must also have unique entries, and in 'Categories' the combination of `variable` and `name` columns must also be unique.

The object may be specifically formatted to be compatible with additional Maelstrom Research software, in particular Opal environments.

Value

A list of data frame(s) named 'Variables' and (if any) 'Categories', with `Rmonize::class` 'dataschema'.

Examples

```
{

# Use Rmonize_examples to run examples.
library(dplyr)

dataschema <- as_dataschema(Rmonize_examples$`DataSchema`)
glimpse(dataschema)

}
```

`as_dataschema_mlstr` *Validate and coerce as a DataSchema object with specific format restrictions*

Description

Checks if an object is a valid DataSchema with specific format restrictions for compatibility with other Maelstrom Research software and returns it with the appropriate `Rmonize::class` attribute. This function mainly helps validate inputs within other functions of the package but could be used separately to ensure that an object has an appropriate structure.

Usage

```
as_dataschema_mlstr(object)
```

Arguments

`object`	A potential DataSchema object to be coerced.

Details

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function. In 'Variables' the `name` column must also have unique entries, and in 'Categories' the combination of `variable` and `name` columns must also be unique.

The object may be specifically formatted to be compatible with additional Maelstrom Research software, in particular Opal environments.

Value

A list of data frame(s) named 'Variables' and (if any) 'Categories', with `Rmonize::class` 'dataschema_mlstr'.

Examples

```
{

# Use Rmonize_examples to run examples.
library(dplyr)

dataschema_mlstr <- as_dataschema_mlstr(Rmonize_examples$`DataSchema`)
glimpse(dataschema_mlstr)

}
```

as_dataset *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR as_dataset

as_data_dict *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR as_data_dict

as_data_proc_elem *Validate and coerce as a Data Processing Elements object*

Description

Checks if an object is a valid Data Processing Elements and returns it with the appropriate `Rmonize::class` attribute. This function mainly helps validate inputs within other functions of the package but could be used separately to ensure that an object has an appropriate structure.

Usage

```
as_data_proc_elem(object)
```

Arguments

`object`	A potential Data Processing Elements object to be coerced.

Details

The Data Processing Elements specifies the input elements and processing algorithms to generate harmonized variables in the DataSchema formats. It is also contains metadata used to generate documentation of the processing. A Data Processing Elements object is a data frame with specific columns used in data processing: `dataschema_variable`, `input_dataset`, `input_variables`, `Mlstr_harmo::rule_category` and `Mlstr_harmo::algorithm`. To initiate processing, the first entry must be the creation of a harmonized primary identifier variable (e.g., participant unique ID).

Value

A data frame with `Rmonize::class` 'data_proc_elem'.

Examples

```
{

# Use Rmonize_examples to run examples.
library(dplyr)

data_proc_elem <- as_data_proc_elem(Rmonize_examples$`Data_Processing_Elements_no_errors`)

head(data_proc_elem)

}
```

---

as_dossier *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR as_dossier

---

as_harmonized_dossier *Validate and coerce as a harmonized dossier object*

Description

Checks if an object is a valid harmonized dossier and returns it with the appropriate `Rmonize::class` attribute. This function mainly helps validate inputs within other functions of the package but could be used separately to ensure that an object has an appropriate structure. The function has two arguments that can optionally be declared by the user (`unique_col_dataset` and `unique_col_id`). `unique_col_dataset` refers to the columns which contains name of each harmonized dataset. `unique_col_id` refers to the column in harmonized datasets which identifies unique combinations of observation/dataset. These two columns are added to ensure that there is always a unique entity identifier when datasets are pooled.

Usage

```
as_harmonized_dossier(
  object,
  dataschema = attributes(object)$`Rmonize::DataSchema`,
  data_proc_elem = attributes(object)$`Rmonize::Data Processing Elements`,
  harmonized_col_id = attributes(object)$`Rmonize::harmonized_col_id`,
 harmonized_col_dataset = attributes(object)$`Rmonize::harmonized_col_dataset`,
  harmonized_data_dict_apply = FALSE
)
```

Arguments

`object`	A A potential harmonized dossier object to be coerced.
`dataschema`	A DataSchema object.
`data_proc_elem`	A Data Processing Elements object.
`harmonized_col_id`	A character string identifying the name of the column present in every dataset to use as a participant identifier.
`harmonized_col_dataset`	A character string identifying the column to use for dataset names.
`harmonized_data_dict_apply`	Whether to apply the dataschema to each harmonized dataset. FALSE by default.

Details

A harmonized dossier is a named list containing one or more data frames, which are harmonized datasets. A harmonized dossier is generally the product of applying processing to a dossier object The name of each harmonized dataset (data frame) is taken from the reference input dataset. A harmonized dossier also contains the DataSchema and Data Processing Elements used in processing as attributes.

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function. In 'Variables' the `name` column must also have unique entries, and in 'Categories' the combination of `variable` and `name` columns must also be unique.

The Data Processing Elements specifies the input elements and processing algorithms to generate harmonized variables in the DataSchema formats. It is also contains metadata used to generate documentation of the processing. A Data Processing Elements object is a data frame with specific columns used in data processing: `dataschema_variable`, `input_dataset`, `input_variables`, `Mlstr_harmo::rule_category` and `Mlstr_harmo::algorithm`. To initiate processing, the first entry must be the creation of a harmonized primary identifier variable (e.g., participant unique ID).

Value

A list of data frame(s) containing harmonized dataset(s). The DataSchema and Data Processing Elements are preserved as attributes of the harmonized dossier.

Examples

```
{

# Use Rmonize_examples to run examples.

library(dplyr)

harmonized_dossier <- Rmonize_examples[["harmonized_dossier"]]
harmonized_dossier <- as_harmonized_dossier(harmonized_dossier)

glimpse(harmonized_dossier$dataset_study1)

}
```

---

bookdown_open *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR bookdown_open

---

dataschema_evaluate *Generate an assessment report for a DataSchema*

Description

Assesses the content and structure of a DataSchema object and generates reports of the results. This function can be used to evaluate data structure, presence of specific fields, coherence across elements, and data dictionary formats.

Usage

```
dataschema_evaluate(dataschema, taxonomy = NULL)
```

Arguments

dataschema	A DataSchema object.
taxonomy	An optional data frame identifying a variable classification schema.

Details

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function. In 'Variables' the `name` column must also have unique entries, and in 'Categories' the combination of `variable` and `name` columns must also be unique.

A taxonomy is a classification schema that can be defined for variable attributes. A taxonomy is usually extracted from an Opal environment, and a taxonomy object is a data frame that must contain at least the columns `taxonomy`, `vocabulary`, and `terms`. Additional details about Opal taxonomies are available online.

Value

A list of data frames containing assessment reports.

Examples

```
{
# Use Rmonize_examples to run examples.
library(dplyr)

dataschema <- Rmonize_examples$`DataSchema`
eval_dataschema <- dataschema_evaluate(dataschema)

glimpse(eval_dataschema)

}
```

`dataschema_extract` *Generate a DataSchema based on Data Processing Elements*

Description

Generates a DataSchema from a Data Processing Elements.

Usage

```
dataschema_extract(data_proc_elem)
```

Arguments

`data_proc_elem`	A Data Processing Elements object.

Details

The Data Processing Elements specifies the input elements and processing algorithms to generate harmonized variables in the DataSchema formats. It is also contains metadata used to generate documentation of the processing. A Data Processing Elements object is a data frame with specific columns used in data processing: `dataschema_variable`, `input_dataset`, `input_variables`, `Mlstr_harmo::rule_category` and `Mlstr_harmo::algorithm`. To initiate processing, the first entry must be the creation of a harmonized primary identifier variable (e.g., participant unique ID).

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function. In 'Variables' the `name` column must also have unique entries, and in 'Categories' the combination of `variable` and `name` columns must also be unique. The object may be specifically formatted to be compatible with additional Maelstrom Research software, in particular Opal environments.

Value

A list of data frame(s) named 'Variables' and (if any) 'Categories', with `Rmonize::class` 'dataschema'.

Examples

```
{

# Use Rmonize_examples to run examples.
library(dplyr)

dataschema <- dataschema_extract(Rmonize_examples$`Data_Processing_Elements_no_errors`)
glimpse(dataschema)

}
```

`dataset_evaluate` *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR `dataset_evaluate`

dataset_summarize *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR dataset_summarize

dataset_visualize *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR dataset_visualize

data_dict_apply *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR data_dict_apply

data_dict_evaluate *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR data_dict_evaluate

data_dict_extract *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR data_dict_extract

data_proc_elem_evaluate *Generate an assessment report for Data Processing Elements*

Description

Assesses the content and structure of a Data Processing Elements object and generates reports of the results. This function can be used to evaluate data structure, presence of specific fields, coherence across elements, and data dictionary formats.

Usage

```
data_proc_elem_evaluate(data_proc_elem, taxonomy = NULL)
```

Arguments

data_proc_elem	A Data Processing Elements object.
taxonomy	An optional data frame identifying a variable classification schema.

Details

The Data Processing Elements specifies the input elements and processing algorithms to generate harmonized variables in the DataSchema formats. It is also contains metadata used to generate documentation of the processing. A Data Processing Elements object is a data frame with specific columns used in data processing: dataschema_variable, input_dataset, input_variables, Mlstr_harmo::rule_category and Mlstr_harmo::algorithm. To initiate processing, the first entry must be the creation of a harmonized primary identifier variable (e.g., participant unique ID).

A taxonomy is a classification schema that can be defined for variable attributes. A taxonomy is usually extracted from an Opal environment, and a taxonomy object is a data frame that must contain at least the columns taxonomy, vocabulary, and terms. Additional details about Opal taxonomies are available online.

Value

A list of data frames containing assessment reports.

Examples

```
{
# Use Rmonize_examples to run examples.
library(dplyr)

data_proc_elem <- Rmonize_examples$`Data_Processing_Elements_no_errors`

glimpse(data_proc_elem)
}
```

`dossier_create` *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR dossier_create

`dossier_evaluate` *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR dossier_evaluate

`dossier_summarize` *Objects exported from other packages*

Description

These objects are imported from other packages. Follow the links below to see their documentation.

madshapR dossier_summarize

`harmonized_dossier_evaluate` *Generate an assessment report for a harmonized dossier*

Description

Assesses the content and structure of a harmonized dossier and generates reports of the results. This function can be used to evaluate data structure, presence of specific fields, coherence across elements, and data dictionary formats.

Usage

```
harmonized_dossier_evaluate(harmonized_dossier)
```

Arguments

`harmonized_dossier` A list containing the harmonized dataset(s).

Details

A harmonized dossier is a named list containing one or more data frames, which are harmonized datasets. A harmonized dossier is generally the product of applying processing to a dossier object. The name of each harmonized dataset (data frame) is taken from the reference input dataset. A harmonized dossier also contains the DataSchema and Data Processing Elements used in processing as attributes.

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function. In 'Variables' the `name` column must also have unique entries, and in 'Categories' the combination of `variable` and `name` columns must also be unique.

A taxonomy is a classification schema that can be defined for variable attributes. A taxonomy is usually extracted from an Opal environment, and a taxonomy object is a data frame that must contain at least the columns `taxonomy`, `vocabulary`, and `terms`. Additional details about Opal taxonomies are available online. The object may be specifically formatted to be compatible with additional Maelstrom Research software, in particular Opal environments.

Value

A list of data frames containing assessment reports for each harmonized dataset.

Examples

```
# Use Rmonize_examples to run examples.
library(dplyr)
```

```
# Perform data processing
harmonized_dossier <- Rmonize_examples$`harmonized_dossier`

eval_harmo <- harmonized_dossier_evaluate(harmonized_dossier)

glimpse(eval_harmo)
```

`harmonized_dossier_summarize` *Generate an assessment report and summary of a harmonized dossier*

Description

Assesses and summarizes the content and structure of a harmonized dossier and generates reports of the results. This function can be used to evaluate data structure, presence of specific fields, coherence across elements, and data dictionary formats, and to summarize additional information about variable distributions and descriptive statistics.

Usage

```
harmonized_dossier_summarize(harmonized_dossier)
```

Arguments

`harmonized_dossier` A list containing the harmonized dataset(s).

Details

A harmonized dossier is a named list containing one or more data frames, which are harmonized datasets. A harmonized dossier is generally the product of applying processing to a dossier object. The name of each harmonized dataset (data frame) is taken from the reference input dataset. A harmonized dossier also contains the DataSchema and Data Processing Elements used in processing as attributes.

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function. In 'Variables' the `name` column must also have unique entries, and in 'Categories' the combination of `variable` and `name` columns must also be unique.

The Data Processing Elements specifies the input elements and processing algorithms to generate harmonized variables in the DataSchema formats. It is also contains metadata used to generate documentation of the processing. A Data Processing Elements object is a data frame with specific

columns used in data processing: `dataschema_variable`, `input_dataset`, `input_variables`, `Mlstr_harmo::rule_category` and `Mlstr_harmo::algorithm`. To initiate processing, the first entry must be the creation of a harmonized primary identifier variable (e.g., participant unique ID).

A taxonomy is a classification schema that can be defined for variable attributes. A taxonomy is usually extracted from an Opal environment, and a taxonomy object is a data frame that must contain at least the columns `taxonomy`, `vocabulary`, and `terms`. Additional details about Opal taxonomies are available online.

The valueType is a declared property of a variable that is required in certain functions to determine handling of the variables. Specifically, valueType refers to the OBiBa data type of a variable. The valueType is specified in a data dictionary in a column 'valueType' and can be associated with variables as attributes. Acceptable valueTypes include 'text', 'integer', 'decimal', 'boolean', datetime', 'date'. The full list of OBiBa valueType possibilities and their correspondence with R data types are available using `valueType_list`. The valueType can be used to coerce the variable to the corresponding data type.

Value

A list of data frames containing overall assessment reports and summaries grouped by harmonized dataset.

Examples

```
# Use Rmonize_examples to run examples.
library(dplyr)

# Perform data processing
harmonized_dossier <- Rmonize_examples$`harmonized_dossier`
summary_harmo <- harmonized_dossier_summarize(harmonized_dossier)

glimpse(summary_harmo)
```

---

harmonized_dossier_visualize

Generate a web-based visual report for a harmonized dossier

Description

Generates a visual report of a harmonized dossier in an HTML bookdown document, with summary figures and statistics for each harmonized variable. The report outputs can be grouped by a categorical variable.

Usage

```
harmonized_dossier_visualize(
  harmonized_dossier,
  bookdown_path,
  harmonized_dossier_summary = NULL
)
```

Arguments

harmonized_dossier	A list containing the harmonized dataset(s).
bookdown_path	A character string identifying the folder path where the bookdown report files will be saved.
harmonized_dossier_summary	A list which identifies an existing summary produced by harmonized_dossier_summarize() of the harmonized variables. Using this parameter can save time in generating the visual report.

Details

A harmonized dossier is a named list containing one or more data frames, which are harmonized datasets. A harmonized dossier is generally the product of applying processing to a dossier object. The name of each harmonized dataset (data frame) is taken from the reference input dataset. A harmonized dossier also contains the DataSchema and Data Processing Elements used in processing as attributes.

Value

A folder containing files for the bookdown document. To open the bookdown document in a browser, open 'docs/index.html', or use bookdown_open() with the folder path.

See Also

dataset_visualize() bookdown_open()

Examples

```
library(fs)

# Use Rmonize_examples to run examples.
# Perform data processing

harmonized_dossier <- Rmonize_examples$`harmonized_dossier`
harmonized_dossier_summary <- Rmonize_examples$`summary_report_harmonized_dossier`

# Create a folder where the visual report will be placed

if(dir_exists(tempdir())) dir_delete(tempdir())
bookdown_path <- tempdir()
```

```
# Generate the visual report
harmonized_dossier_visualize(
  harmonized_dossier = harmonized_dossier,
  bookdown_path = bookdown_path,
  harmonized_dossier_summary = harmonized_dossier_summary)

# To open the file in a browser, open 'bookdown_path/docs/index.html'.
# Or use bookdown_open(bookdown_path) function.
```

`harmo_process` *Generate harmonized dataset(s) and associated metadata*

Description

Reads a DataSchema and Data Processing Elements to generate a harmonized dossier from input dataset(s) in a dossier and associated metadata. The function has one argument that can optionally be declared by the user (`unique_col_dataset`). It refers to the columns which contains name of each harmonized dataset. These two columns are added to ensure that there is always a unique entity identifier when datasets are pooled.

Usage

```
harmo_process(
  object,
  dataschema = attributes(object)$`Rmonize::DataSchema`,
  data_proc_elem = attributes(object)$`Rmonize::Data Processing Elements`,
 harmonized_col_dataset = attributes(object)$`Rmonize::harmonized_col_dataset`,
  harmonized_col_id = attributes(object)$`Rmonize::harmonized_col_id`,
  .debug = FALSE
)
```

Arguments

`object`	Data frame(s) or list of data frame(s) containing input dataset(s).
`dataschema`	A DataSchema object.
`data_proc_elem`	A Data Processing Elements object.
`harmonized_col_dataset`	A character string identifying the column to use for dataset names.
`harmonized_col_id`	A character string identifying the name of the column present in every dataset to use as a participant identifier.
`.debug`	Allow user to test the inputs before processing harmonization.

Details

A dossier is a named list containing one or more data frames, which are input datasets. The name of each data frame in the dossier will be used as the name of the associated harmonized dataset produced by `harmo_process()`.

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function. In 'Variables' the `name` column must also have unique entries, and in 'Categories' the combination of `variable` and `name` columns must also be unique.

The Data Processing Elements specifies the input elements and processing algorithms to generate harmonized variables in the DataSchema formats. It is also contains metadata used to generate documentation of the processing. A Data Processing Elements object is a data frame with specific columns used in data processing: `dataschema_variable`, `input_dataset`, `input_variables`, `Mlstr_harmo::rule_category` and `Mlstr_harmo::algorithm`. To initiate processing, the first entry must be the creation of a harmonized primary identifier variable (e.g., participant unique ID).

Value

A list of data frame(s) containing harmonized dataset(s). The DataSchema and Data Processing Elements are preserved as attributes of the harmonized dossier.

Examples

```
# Use Rmonize_examples to run examples.

library(dplyr)
library(stringr)
library(lubridate)

# Perform data processing
dossier <- Rmonize_examples[str_detect(names(Rmonize_examples),"input_dataset_study")]
names(dossier) <- str_remove(names(dossier), "input_")
dataschema <- Rmonize_examples$`DataSchema`
data_proc_elem <- Rmonize_examples$`Data_Processing_Elements_no_errors`

harmonized_dossier <- harmo_process(
  dossier,
  dataschema,
  data_proc_elem,
  harmonized_col_dataset = 'adm_study_id')

glimpse(harmonized_dossier$dataset_study1)
```

`is_dataschema` *Test for a valid DataSchema object*

Description

Tests if the input is a valid DataSchema object. This function mainly helps validate input within other functions of the package but could be used to check if an object is valid for use in a function.

Usage

```
is_dataschema(object)
```

Arguments

`object`	A potential DataSchema object to be evaluated.

Details

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function. In 'Variables' the `name` column must also have unique entries, and in 'Categories' the combination of `variable` and `name` columns must also be unique.

Value

A logical.

See Also

`dataschema_evaluate()`.

Examples

```
{

# Use Rmonize_examples to run examples.

is_dataschema(Rmonize_examples$`DataSchema`)
is_dataschema(Rmonize_examples$`Data_Processing_Elements_no_errors`)
is_dataschema(iris)

}
```

`is_dataschema_mlstr` *Test for a valid DataSchema object with specific format restrictions*

Description

Tests if an object is a valid DataSchema object with specific format restrictions for compatibility with other Maelstrom Research software. This function mainly helps validate input within other functions of the package but could be used to check if an object is valid for use in a function.

Usage

```
is_dataschema_mlstr(object)
```

Arguments

`object`	A potential DataSchema object to be evaluated.

Details

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function. In 'Variables' the `name` column must also have unique entries, and in 'Categories' the combination of `variable` and `name` columns must also be unique.

The object may be specifically formatted to be compatible with additional Maelstrom Research software, in particular Opal environments.

Value

A logical.

See Also

`dataschema_evaluate()`.

Examples

```
{

# use Rmonize_examples provided by the package

is_dataschema_mlstr(Rmonize_examples$`DataSchema`)
is_dataschema_mlstr(Rmonize_examples$`Data_Processing_Elements_no_errors`)
is_dataschema_mlstr(iris)

}
```

`is_data_proc_elem` *Test for a valid Data Processing Elements object*

Description

Tests if the input is a valid Data Processing Elements object. This function mainly helps validate input within other functions of the package but could be used to check if an object is valid for use in a function.

Usage

```
is_data_proc_elem(object)
```

Arguments

`object`	A potential Data Processing Elements object to be evaluated.

Details

The Data Processing Elements specifies the input elements and processing algorithms to generate harmonized variables in the DataSchema formats. It is also contains metadata used to generate documentation of the processing. A Data Processing Elements object is a data frame with specific columns used in data processing: `dataschema_variable`, `input_dataset`, `input_variables`, `Mlstr_harmo::rule_category` and `Mlstr_harmo::algorithm`. To initiate processing, the first entry must be the creation of a harmonized primary identifier variable (e.g., participant unique ID).

Value

A logical.

Examples

```
{

# Use Rmonize_examples to run examples.

is_data_proc_elem(Rmonize_examples$`Data_Processing_Elements_no_errors`)
is_data_proc_elem(Rmonize_examples$`DataSchema`)
is_data_proc_elem(iris)
}
```

`pooled_harmonized_dataset_create` *Generate a pooled harmonized dataset from a harmonized dossier*

Description

Generates a pooled harmonized dataset from a harmonized dossier. The function has two arguments that can optionally be declared by the user (`harmonized_col_dataset` and `harmonized_col_id`). `harmonized_col_dataset` refers to the columns which contains name of each harmonized dataset. `harmonized_col_id` refers to the column in harmonized datasets which identifies unique combinations of observation/dataset. These two columns are added to ensure that there is always a unique entity identifier when datasets are pooled.

Usage

```
pooled_harmonized_dataset_create(
  harmonized_dossier,
  harmonized_col_dataset =
    attributes(harmonized_dossier)$`Rmonize::harmonized_col_dataset`,
 harmonized_col_id = attributes(harmonized_dossier)$`Rmonize::harmonized_col_id`,
  dataschema = attributes(harmonized_dossier)$`Rmonize::DataSchema`,
 data_proc_elem = attributes(harmonized_dossier)$`Rmonize::Data Processing Elements`
)
```

Arguments

`harmonized_dossier`	A list containing the harmonized dataset(s).
`harmonized_col_dataset`	A character string identifying the column to use for dataset names.
`harmonized_col_id`	A character string identifying the name of the column present in every dataset to use as a participant identifier.
`dataschema`	A DataSchema object.
`data_proc_elem`	A Data Processing Elements object.

Details

A harmonized dossier is a named list containing one or more data frames, which are harmonized datasets. A harmonized dossier is generally the product of applying processing to a dossier object The name of each harmonized dataset (data frame) is taken from the reference input dataset. A harmonized dossier also contains the DataSchema and Data Processing Elements used in processing as attributes.

A DataSchema is the list of core variables to generate across datasets and related metadata. A DataSchema object is a list of data frames with elements named 'Variables' (required) and 'Categories' (if any). The 'Variables' element must contain at least the `name` column, and the 'Categories' element must contain at least the `variable` and `name` columns to be usable in any function.

In 'Variables' the name column must also have unique entries, and in 'Categories' the combination of variable and name columns must also be unique.

The Data Processing Elements specifies the input elements and processing algorithms to generate harmonized variables in the DataSchema formats. It is also contains metadata used to generate documentation of the processing. A Data Processing Elements object is a data frame with specific columns used in data processing: dataschema_variable, input_dataset, input_variables, Mlstr_harmo::rule_category and Mlstr_harmo::algorithm. To initiate processing, the first entry must be the creation of a harmonized primary identifier variable (e.g., participant unique ID).

Value

A data frame containing the pooled harmonized dataset.

Examples

```
{

# Use Rmonize_examples to run examples.

library(dplyr)

# Perform data processing
harmonized_dossier <- Rmonize_examples["harmonized_dossier"][[1]]

# create the pooled harmonized dataset from the harmonized dossier
pooled_harmonized_dataset <- pooled_harmonized_dataset_create(harmonized_dossier)

glimpse(pooled_harmonized_dataset)

}
```

Rmonize_examples *Example objects to provide an illustrative use case*

Description

Example input datasets, input data dictionaries, DataSchema, Data Processing Elements, harmonized output, and summary report.

Usage

```
Rmonize_examples
```

Format

`list`:
A list with elements (data frames and lists) providing example objects for using the package:

original_dataset_study1 Example original dataset from .sav file for "study1"

original_dataset_study2 Example original dataset from .sav file for "study2"

original_dataset_study3 Example original dataset from .csv file for "study3"

original_dataset_study4 Example original dataset from .xlsx file for "study4"

original_dataset_study5 Example original dataset from .xlsx file for "study5"

original_data_dictionary_study4 Example original data dictionary from .xlsx file, for "study4"

original_data_dictionary_study5 Example original data dictionary from .xlsx file, for "study5"

input_dataset_study1 Example input dataset ready for processing, for "study1"

input_dataset_study2 Example input dataset ready for processing, for "study2"

input_dataset_study3 Example input dataset ready for processing, for "study3"

input_dataset_study4 Example input dataset ready for processing, for "study4"

input_dataset_study5 Example input dataset ready for processing, for "study5"

DataSchema Example DataSchema

Data_Processing_Elements_no_errors Example Data Processing Elements containing no errors

Data_Processing_Elements_with_errors Example Data Processing Elements containing errors

harmonized_dossier Example harmonized dossier

pooled_harmonized_dataset Example pooled harmonized dataset

summary_report_harmonized_dossier Example summary report of harmonized dossier

Examples

```
{

library(dplyr)
glimpse(Rmonize_examples$`DataSchema`)


}
```

Rmonize_templates *Call to online documentation to download templates*

Description

Direct call to online documentation to download templates.

Usage

```
Rmonize_templates()
```

Value

Nothing to be returned. The function opens a web page.

Examples

```
{

Rmonize_templates()

}
```

`Rmonize_website` *Call to package website*

Description

Direct call to the package website, which includes an overview of the Rmonize process, vignettes and user guides, a reference list of functions and help pages, and package updates.

Usage

```
Rmonize_website()
```

Value

Nothing to be returned. The function opens a web page.

Examples

```
{

Rmonize_website()

}
```

show_harmo_error *Print a summary of data processing in the console*

Description

Reads a harmonized dossier, product of `harmo_process()`, to list processes, any errors, and an overview of each harmonization rule. The output printed in the console can help in correcting any errors that occurred during data processing.

Usage

```
show_harmo_error(harmonized_dossier, show_warnings = TRUE)
```

Arguments

harmonized_dossier	A list containing the harmonized dataset(s).
show_warnings	Whether the function should print warnings or not. TRUE by default.

Details

A harmonized dossier is a named list containing one or more data frames, which are harmonized datasets. A harmonized dossier is generally the product of applying processing to a dossier object The name of each harmonized dataset (data frame) is taken from the reference input dataset. A harmonized dossier also contains the DataSchema and Data Processing Elements used in processing as attributes.

Value

Nothing to be returned. The function prints messages in the console, showing any errors in data processing.

Examples

```
{
# Use Rmonize_examples to run examples.
library(dplyr)

# Perform data processing
harmonized_dossier <- Rmonize_examples$`harmonized_dossier`

# Show error(s) on the console
show_harmo_error(harmonized_dossier)
}
```

Index

∗ **datasets**
    Rmonize_examples, 24
∗ **imported**
    as_data_dict, 5
    as_dataset, 5
    as_dossier, 6
    bookdown_open, 8
    data_dict_apply, 11
    data_dict_evaluate, 11
    data_dict_extract, 12
    dataset_evaluate, 10
    dataset_summarize, 11
    dataset_visualize, 11
    dossier_create, 13
    dossier_evaluate, 13
    dossier_summarize, 13

as_data_dict, *5*, 5
as_data_proc_elem, 5
as_dataschema, 3
as_dataschema_mlstr, 4
as_dataset, *5*, 5
as_dossier, *6*, 6
as_harmonized_dossier, 6

bookdown_open, *8*, 8
bookdown_open(), *17*

data_dict_apply, *11*, 11
data_dict_evaluate, *11*, 11
data_dict_extract, *12*, 12
data_proc_elem_evaluate, 12
dataschema_evaluate, 8
dataschema_evaluate(), *20*, *21*
dataschema_extract, 9
dataset_evaluate, *10*, 10
dataset_summarize, *11*, 11
dataset_visualize, *11*, 11
dataset_visualize(), *17*
dossier_create, *13*, 13
dossier_evaluate, *13*, 13
dossier_summarize, *13*, 13

harmo_process, 18
harmo_process(), *19*, *27*
harmonized_dossier_evaluate, 14
harmonized_dossier_summarize, 15
harmonized_dossier_summarize(), *17*
harmonized_dossier_visualize, 16

is_data_proc_elem, 22
is_dataschema, 20
is_dataschema_mlstr, 21

pooled_harmonized_dataset_create, 23

Rmonize_examples, 24
Rmonize_templates, 25
Rmonize_website, 26

show_harmo_error, 27