Package 'Rmonize'

May 1, 2024

Type Package

Title Support Retrospective Harmonization of Data

Version 1.1.0

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

Description Functions to support rigorous retrospective data harmonization processing, evaluation, and documentation across datasets from different studies based on Maelstrom Research guidelines. The package includes the core functions to evaluate and format the main inputs that define the harmonization process, apply specified processing rules to generate harmonized data, diagnose processing errors, and summarize and evaluate harmonized outputs. The main inputs that define the processing are a DataSchema (list and definitions of harmonized variables to be generated) and Data Processing Elements (processing rules to be applied to generate harmonized variables from study-specific variables). The main outputs of processing are harmonized datasets, associated metadata, and tabular and visual summary reports. As described in Maelstrom Research guidelines for rigorous retrospective data harmonization (Fortier I and al. (2017) <doi:10.1093/ije/dyw075>).

License GPL-3

LazyData true

Depends R (>= 3.4)

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

Suggests janitor, car, 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] (<https://orcid.org/0000-0002-0124-9970>), Maelstrom-research group [fnd, cph]

Repository CRAN

Date/Publication 2024-05-01 16:12:08 UTC

R topics documented:

`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_DEMO to run examples.
library(dplyr)

glimpse(as_dataschema(Rmonize_DEMO$`dataschema - final`))

}
```

`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_DEMO to run examples.
library(dplyr)

glimpse(as_dataschema_mlstr(Rmonize_DEMO$`dataschema - final`))

}
```

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 algorithms used to process input variables into harmonized variables in the DataSchema format. 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_DEMO to run examples.
library(dplyr)

glimpse(head(as_data_proc_elem(Rmonize_DEMO$`data_processing_elements - final`),3))

}
```

---

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 dataset 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 algorithms used to process input variables into harmonized variables in the DataSchema format. 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 output harmonized dossier.

Examples

```
{

# Use Rmonize_DEMO to run examples.
library(dplyr)

glimpse(as_harmonized_dossier(Rmonize_DEMO$harmonized_dossier))

}
```

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_DEMO provided by the package

library(dplyr)
library(madshapR) # data_dict_filter

dataschema <-
  Rmonize_DEMO$`dataschema - final` %>%
  data_dict_filter("name == 'adm_unique_id'")

dataschema_evaluate(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 algorithms used to process input variables into harmonized variables in the DataSchema format. 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) named 'Variables' and (if any) 'Categories', with `Rmonize::class` 'dataschema'.

Examples

```
{

# Use Rmonize_DEMO to run examples.
library(dplyr)

glimpse(dataschema_extract(
  data_proc_elem = Rmonize_DEMO$`data_processing_elements - final`))
}
```

`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`

`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,
  dataschema = attributes(harmonized_dossier)$`Rmonize::DataSchema`,
  taxonomy = NULL,
  as_dataschema_mlstr = TRUE
)
```

Arguments

harmonized_dossier	A list containing the harmonized dataset(s).
dataschema	A DataSchema object.
taxonomy	An optional data frame identifying a variable classification schema.
as_dataschema_mlstr	Whether the output DataSchema should be coerced with specific format restrictions for compatibility with other Maelstrom Research software. 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.

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_DEMO provided by the package
library(dplyr)

glimpse(harmonized_dossier_evaluate(Rmonize_DEMO$harmonized_dossier))

}
```

`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,
  group_by = attributes(harmonized_dossier)$`Rmonize::harmonized_col_dataset`,
  dataschema = attributes(harmonized_dossier)$`Rmonize::DataSchema`,
 data_proc_elem = attributes(harmonized_dossier)$`Rmonize::Data Processing Element`,
  taxonomy = NULL,
  valueType_guess = FALSE
)
```

Arguments

harmonized_dossier	A list containing the harmonized dataset(s).
group_by	A character string identifying the column in the dataset to use as a grouping variable. Elements will be grouped by this column.
dataschema	A DataSchema object.
data_proc_elem	A Data Processing Elements object.
taxonomy	An optional data frame identifying a variable classification schema.
valueType_guess	Whether the output should include a more accurate valueType that could be applied to the 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 algorithms used to process input variables into harmonized variables in the DataSchema format. 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

```
{

harmonized_dossier <- Rmonize_DEMO$harmonized_dossier

# summary harmonization
harmonized_dossier_summarize(harmonized_dossier)

}
```

---

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,
  group_by = attributes(harmonized_dossier)$`Rmonize::harmonized_col_dataset`,
  harmonized_dossier_summary = NULL,
  dataschema = attributes(harmonized_dossier)$`Rmonize::DataSchema`,
 data_proc_elem = attributes(harmonized_dossier)$`Rmonize::Data Processing Element`,
  valueType_guess = FALSE,
  taxonomy = 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.
group_by	A character string identifying the column in the dataset to use as a grouping variable. Elements will be grouped by this column.
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.
dataschema	A DataSchema object.
data_proc_elem	A Data Processing Elements object.
valueType_guess	Whether the output should include a more accurate valueType that could be applied to the dataset. FALSE by default.
taxonomy	An optional data frame identifying a variable classification schema.

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 algorithms used to process input variables into harmonized variables in the DataSchema format. 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).

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.

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

See Also

`dataset_visualize()` `bookdown_open()`

Examples

```
{

# Use Rmonize_DEMO to run examples.

library(fs)

harmonized_dossier <- Rmonize_DEMO$harmonized_dossier
harmonized_dossier_summary <- Rmonize_DEMO$harmonized_dossier_summary

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

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

# To open the file in 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 = NULL,
  dataschema = attributes(dossier)$`Rmonize::DataSchema`,
  data_proc_elem = attributes(dossier)$`Rmonize::Data Processing Elements`,
 harmonized_col_dataset = attributes(dossier)$`Rmonize::harmonized_col_dataset`,
  harmonized_col_id = attributes(dossier)$`Rmonize::harmonized_col_id`,
  .debug = FALSE,
  dossier = object
)
```

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. NULL by default.
`harmonized_col_id`	A character string identifying the name of the column present in every dataset to use as a dataset identifier. NULL by default.
`.debug`	Allow user to test the inputs before processing harmonization.
`dossier`	**[Deprecated]**

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 algorithms used to process input variables into harmonized variables in the DataSchema format. 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 output harmonized dossier.

Examples

```
{

# Use Rmonize_DEMO to run examples.

library(dplyr)
library(madshapR) # data_dict_filter

dataset_MELBOURNE <- Rmonize_DEMO$dataset_MELBOURNE[1]
dossier <- dossier_create(list(dataset_MELBOURNE))

dataschema <-
  Rmonize_DEMO$`dataschema - final` %>%
  data_dict_filter('name == "adm_unique_id"')

data_proc_elem <- Rmonize_DEMO$`data_processing_elements - final` %>%
  dplyr::filter(dataschema_variable == 'adm_unique_id',
         input_dataset == 'dataset_MELBOURNE')

# perform harmonization
harmonized_dossier <- harmo_process(dossier,dataschema,data_proc_elem)
glimpse(harmonized_dossier)

}
```

`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

For a better assessment, please use `dataschema_evaluate()`.

Examples

```
{

# use Rmonize_DEMO provided by the package

dataschema <- Rmonize_DEMO$`dataschema - final`
is_dataschema(dataschema)
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

For a better assessment, please use `dataschema_evaluate()`.

Examples

```
{

# use Rmonize_DEMO provided by the package

dataschema <- Rmonize_DEMO$`dataschema - final`
is_dataschema_mlstr(dataschema)
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 algorithms used to process input variables into harmonized variables in the DataSchema format. 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_DEMO provided by the package

data_proc_elem <- Rmonize_DEMO$`data_processing_elements - final`
is_data_proc_elem(data_proc_elem)
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 (`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

```
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`,
  add_col_dataset = FALSE,
  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 dataset identifier.

`add_col_dataset`
: Whether to add an extra column to each harmonized dataset. The resulting data frame will have an additional column and its data dictionary will be updated accordingly adding categories for this variable if necessary. FALSE by default.

`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 algorithms used to process input variables into harmonized variables in the DataSchema format. 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 madshapR_DEMO provided by the package
library(dplyr)

harmonized_dossier <- Rmonize_DEMO$harmonized_dossier

glimpse(pooled_harmonized_dataset_create(
  harmonized_dossier,harmonized_col_id = 'adm_unique_id'))

}
```

`Rmonize_DEMO` *Demo objects to provide illustrative examples*

Description

Demo input datasets, input data dictionaries, DataSchema, Data Processing Elements, and other objects to provide illustrative examples of objects used by Rmonize.

Usage

```
Rmonize_DEMO
```

Format

`list`:

A list with 13 elements (data frames and lists) providing example objects for testing the package:

data_processing_elements - final Example Data Processing Elements

data_processing_elements - with error Example Data Processing Elements containing errors

data_processing_elements - work in progress Example incomplete Data processing Element

dataschema - final Example DataSchema

pooled_harmonized_dataset Example pooled harmonized dataset

harmonized_dossier Example of harmonized dossier

harmonized_dossier_summary Example harmonized variables summary

data_dict_MELBOURNE Example Data dictionary for Melbourne dataset

data_dict_PARIS Example Data dictionary for Paris dataset

data_dict_TOKYO Example Data dictionary for Tokyo dataset

dataset_MELBOURNE Example Dataset for Melbourne

dataset_PARIS Example Dataset for Paris

dataset_TOKYO Example Dataset for Tokyo ...

Examples

```
{

# use madshapR_DEMO provided by the package
library(dplyr)

glimpse(Rmonize_DEMO$`dataschema - final`)

}
```

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 online documentation*

Description

Direct call to the online documentation for the package, which includes a description of the latest version of the package, vignettes, user guides, and a reference list of functions and help pages.

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 the processing.

Examples

```
{

  harmonized_dossier <- Rmonize_DEMO$harmonized_dossier
  show_harmo_error(harmonized_dossier)

}
```

Index