Package 'retroharmonize'

May 21, 2026

Type Package

Title Ex Post Survey Data Harmonization

Version 0.2.8

Date 2026-05-19

Maintainer Daniel Antal <daniel.antal@dataobservatory.eu>

Description Assist in reproducible retrospective (ex-post) harmonization of data, particularly individual level survey data, by providing tools for organizing metadata, standardizing the coding of variables, and variable names and value labels, including missing values, and documenting the data transformations, with the help of comprehensive s3 classes.

License GPL-3

URL https://retroharmonize.dataobservatory.eu/

BugReports https://github.com/dataobservatory-eu/retroharmonize/issues

Depends R (>= 3.5.0)

Imports assertthat, cli, dataset, dplyr (>= 1.0.0), fs, glue, haven, here, labelled, magrittr, purrr, rlang, snakecase, stats, stringr, tibble, tidyr, tidyselect, utils, vctrs

Suggests covr, ggplot2, knitr, markdown, png, rmarkdown, pillar, spelling, statcodelists, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Daniel Antal [aut, cre] (ORCID: <https://orcid.org/0000-0001-7513-6760>), Marta Kolczynska [ctb] (ORCID: <https://orcid.org/0000-0003-4981-0437>)

Repository CRAN

Date/Publication 2026-05-21 07:20:09 UTC

Contents

`as_labelled_spss_survey`

Labelled to labelled_spss_survey

Description

Labelled to labelled_spss_survey

Usage

```
as_labelled_spss_survey(x, id)
```

Arguments

x	A vector of class haven_labelled or haven_labelled_spss.
id	The survey identifier.

Value

A vector of labelled_spss_survey

See Also

Other type conversion functions: labelled_spss_survey_coercion

---

collect_val_labels *Collect labels from metadata file*

Description

Collect labels from metadata file

Usage

```
collect_val_labels(metadata)

collect_na_labels(metadata)
```

Arguments

metadata	A metadata data frame created by metadata_create.

Value

The unique valid labels or the user-defined missing labels found in all the files analyzed in metadata.

See Also

Other harmonization functions: crosswalk_surveys(), harmonize_na_values(), harmonize_survey_values(), harmonize_values(), harmonize_var_names(), is.crosswalk_table(), label_normalize()

Examples

```
test_survey <- retroharmonize::read_rds(
  file = system.file("examples", "ZA7576.rds",
    package = "retroharmonize"
  ),
  id = "test"
)
example_metadata <- metadata_create(test_survey)

collect_val_labels(metadata = example_metadata)
collect_na_labels(metadata = example_metadata)
```

concatenate *Concatenate haven_labelled_spss vectors*

Description

Concatenate haven_labelled_spss vectors

Usage

```
concatenate(x, y)
```

Arguments

x	A haven_labelled_spss vector.
y	A haven_labelled_spss vector.

Value

A concatenated haven_labelled_spss vector. Returns an error if the attributes do not match. Gives a warning when only the variable label do not match.

Examples

```
v1 <- labelled::labelled(
  c(3, 4, 4, 3, 8, 9),
  c(YES = 3, NO = 4, `WRONG LABEL` = 8, REFUSED = 9)
)
v2 <- labelled::labelled(
  c(4, 3, 3, 9),
  c(YES = 3, NO = 4, `WRONG LABEL` = 8, REFUSED = 9)
)
s1 <- haven::labelled_spss(
  x = unclass(v1), # remove labels from earlier defined
  labels = labelled::val_labels(v1), # use the labels from earlier defined
  na_values = NULL,
  na_range = 8:9,
  label = "Variable Example"
)

s2 <- haven::labelled_spss(
  x = unclass(v2), # remove labels from earlier defined
  labels = labelled::val_labels(v2), # use the labels from earlier defined
  na_values = NULL,
  na_range = 8:9,
  label = "Variable Example"
)
concatenate(s1, s2)
```

`create_codebook` *Create a survey codebook*

Description

Expand survey metadata into a long-format codebook of value labels.

Usage

```
create_codebook(metadata = NULL, survey = NULL)

codebook_waves_create(waves)

codebook_surveys_create(survey_list)
```

Arguments

`metadata`	A metadata table created by [metadata_create()]. If supplied, ‘survey‘ must be ‘NULL‘.
`survey`	A survey object of class ‘"survey"‘. If supplied, metadata is generated internally using [metadata_create()].
`waves`	A list of surveys.
`survey_list`	A list containing surveys of class survey.

Details

‘create_codebook()‘ takes survey-level metadata and returns a tidy data frame describing all labelled variables and their associated value labels. Each row corresponds to a single value label, classified as either a valid value or a missing value.

Unlabelled numeric and character variables are excluded.

For multiple survey waves, use [codebook_surveys_create()].

If both ‘metadata‘ and ‘survey‘ are provided, ‘survey‘ takes precedence.

Value

A data frame with one row per value label, including:

- survey identifiers (‘id‘, ‘filename‘)
- original variable names and labels
- value codes and value labels
- label type (‘"valid"‘ or ‘"missing"‘)
- summary counts of labels

Additional user-defined metadata columns present in the input metadata are preserved.

See Also

[metadata_create()], [codebook_surveys_create()]

Other metadata functions: `is.crosswalk_table()`, `metadata_create()`, `metadata_survey_create()`

Examples

```
survey <- read_rds(
  system.file("examples", "ZA7576.rds", package = "retroharmonize")
)

cb <- create_codebook(survey = survey)
head(cb)


examples_dir <- system.file("examples", package = "retroharmonize")
survey_list <- dir(examples_dir)[grepl("\\.rds", dir(examples_dir))]

example_surveys <- read_surveys(
  file.path(examples_dir, survey_list),
  save_to_rds = FALSE
)

codebook_surveys_create(example_surveys)
```

`crosswalk_surveys` *Crosswalk and harmonize surveys*

Description

Harmonize one or more surveys using a crosswalk table that defines how variable names, value labels, numeric codes, and variable classes should be aligned across surveys.

Usage

```
crosswalk_surveys(
  crosswalk_table,
  survey_list = NULL,
  survey_paths = NULL,
  import_path = NULL,
  na_values = NULL
)

crosswalk(survey_list, crosswalk_table, na_values = NULL)
```

Arguments

crosswalk_table	A crosswalk table created with [crosswalk_table_create()] or a data frame containing at least the columns ‘id‘, ‘var_name_orig‘, and ‘var_name_target‘. If the columns ‘val_label_orig‘ and ‘val_label_target‘ are present, value labels are harmonized. If ‘val_numeric_orig‘ and ‘val_numeric_target‘ are present, numeric codes are harmonized. If ‘class_target‘ is present, variables are coerced to the specified target class (‘"factor"‘, ‘"numeric"‘, or ‘"character"‘) using [as_factor()], [as_numeric()], or [as_character()].
survey_list	A list of survey objects to be harmonized.
survey_paths	Optional character vector of file paths to surveys. Used when surveys must be read from disk before harmonization.
import_path	Optional base directory used to resolve ‘survey_paths‘. This is primarily intended for workflows where surveys are stored outside the current working directory.
na_values	Optional named vector defining numeric codes to be treated as missing values. Names correspond to missing-value labels.

Details

A crosswalk table can be created with [crosswalk_table_create()] or supplied manually as a data frame. At a minimum, the table must contain columns ‘id‘, ‘var_name_orig‘, and ‘var_name_target‘. Additional columns enable harmonization of value labels, numeric codes, missing values, and variable classes.

Value

‘crosswalk_surveys()‘ returns a list of harmonized survey data frames. ‘crosswalk()‘ returns either a single data frame (if only one survey is harmonized) or a merged data frame combining all harmonized surveys.

See Also

[crosswalk_table_create()] to create a crosswalk table, [harmonize_survey_variables()] for lower-level variable harmonization.

Other harmonization functions: `collect_val_labels()`, `harmonize_na_values()`, `harmonize_survey_values()`, `harmonize_values()`, `harmonize_var_names()`, `is.crosswalk_table()`, `label_normalize()`

Examples

```
## Not run:
examples_dir <- system.file("examples", package = "retroharmonize")
survey_files <- dir(examples_dir, pattern = "\\.rds$")

surveys <- read_surveys(
  file.path(examples_dir, survey_files),
  save_to_rds = FALSE
)
```

```
metadata <- metadata_create(survey_list = surveys)

crosswalk_table <- crosswalk_table_create(metadata)

harmonized <- crosswalk_surveys(
  crosswalk_table = crosswalk_table,
  survey_list = surveys
)

## End(Not run)
```

`document_surveys` *Document survey lists*

Description

Document the key attributes surveys in a survey list.

Usage

```
document_surveys(survey_list = NULL, survey_paths = NULL, .f = NULL)

document_waves(waves)
```

Arguments

`survey_list`	A list of `survey` objects.
`survey_paths`	A vector of full file paths to the surveys to subset, defaults to `NULL`.
`.f`	A function to import the surveys with. Defaults to `'read_rds'`. For SPSS files, `read_spss` is recommended, which is a well-parameterized version of `read_spss` that saves some metadata, too. For STATA files use `read_dta`.
`waves`	A list of `survey` objects.

Details

The function has two alternative input parameters. If `survey_list` is the input, it returns the name of the original source data file, the number of rows and columns, and the size of the object as stored in memory. In case `survey_paths` contains the source data files, it will sequentially read those files, and add the file size, the last access and the last modified time attributes.

The earlier form `document_waves` is deprecated. Currently called `document_surveys`.

Value

Returns a data frame with the key attributes of the surveys in a survey list: the name of the data file, the number of rows and columns, and the size of the object as stored in memory.

See Also

Other documentation functions: document_survey_item()

Examples

```
examples_dir <- system.file("examples", package = "retroharmonize")

my_rds_files <- dir(examples_dir)[grepl(
  ".rds",
  dir(examples_dir)
)]

example_surveys <- read_surveys(file.path(examples_dir, my_rds_files))

documented <- document_surveys(example_surveys)

attr(documented, "original_list")
documented

document_surveys(survey_paths = file.path(examples_dir, my_rds_files))
```

---

document_survey_item *Document survey item provenance*

Description

Document the current and historical coding, labels, missing values, and survey provenance of a harmonized survey variable.

Usage

```
document_survey_item(x)
```

Arguments

x	A 'labelled_spss_survey' vector originating from a single survey or concatenated from multiple surveys.

Value

A named list containing:

- current and historical value coding,
- variable labels,
- valid and missing value definitions,
- original variable names,
- survey identifiers.

See Also

Other documentation functions: document_surveys()

Examples

```
var1 <- labelled::labelled_spss(
  x = c(1, 0, 1, 1, 0, 8, 9),
  labels = c(
    "TRUST" = 1,
    "NOT TRUST" = 0,
    "DON'T KNOW" = 8,
    "INAP. HERE" = 9
  ),
  na_values = c(8, 9)
)

var2 <- labelled::labelled_spss(
  x = c(2, 2, 8, 9, 1, 1),
  labels = c(
    "Tend to trust" = 1,
    "Tend not to trust" = 2,
    "DK" = 8,
    "Inap" = 9
  ),
  na_values = c(8, 9)
)

harmonization <- list(
  from = c(
    "^tend\\sto|^trust",
    "^tend\\snot|not\\strust",
    "^dk|^don",
    "^inap"
  ),
  to = c(
    "trust",
    "not_trust",
    "do_not_know",
    "inap"
  ),
  numeric_values = c(1, 0, 99997, 99999)
)

missing_values <- c(
  "do_not_know" = 99997,
  "inap" = 99999
)

h1 <- harmonize_values(
  x = var1,
  harmonize_label = "Do you trust the European Union?",
  harmonize_labels = harmonization,
```

```
  na_values = missing_values,
  id = "survey1"
)

h2 <- harmonize_values(
  x = var2,
  harmonize_label = "Do you trust the European Union?",
  harmonize_labels = harmonization,
  na_values = missing_values,
  id = "survey2"
)

h3 <- concatenate(h1, h2)

document_survey_item(h3)
```

`harmonize_na_values` *Harmonize na_values in haven_labelled_spss*

Description

Harmonize na_values in haven_labelled_spss

Usage

```
harmonize_na_values(df)
```

Arguments

`df`	A data frame that contains haven_labelled_spss vectors.

Value

A tibble where the na_values are consistent

See Also

Other harmonization functions: `collect_val_labels()`, `crosswalk_surveys()`, `harmonize_survey_values()`, `harmonize_values()`, `harmonize_var_names()`, `is.crosswalk_table()`, `label_normalize()`

Examples

```
examples_dir <- system.file(
  "examples",
  package = "retroharmonize"
)

test_read <- read_rds(
  file.path(examples_dir, "ZA7576.rds"),
```

```
  id = "ZA7576",
  doi = "test_doi"
)

harmonize_na_values(test_read)
```

`harmonize_survey_values` *Harmonize values in surveys*

Description

Harmonize value codes and value labels across multiple surveys and combine them into a single data frame.

Usage

```
harmonize_survey_values(survey_list, .f, status_message = FALSE)

harmonize_waves(waves, .f, status_message = FALSE)
```

Arguments

`survey_list`	A list of surveys (data frames). In earlier versions this argument was called `waves`.
`.f`	A function applied to each labelled variable (class `"retroharmonize_labelled_spss_survey"`). The function must not change the length of the input vector.
`status_message`	Logical. If `TRUE`, prints the identifier of each survey as it is processed.
`waves`	A list of surveys. Deprecated.

Details

The function first aligns the structure of all surveys by ensuring that they contain the same set of variables. Missing variables are added and filled with appropriate missing values depending on their type.

Variables of class `"retroharmonize_labelled_spss_survey"` are then harmonized by applying a user-supplied function `.f` to each variable separately within each survey.

The harmonization function `.f` must return a vector of the same length as its input. If `.f` returns `NULL`, the original variable is kept unchanged.

Prior to version 0.2.0 this function was called `harmonize_waves`.

The earlier form `harmonize_waves` is deprecated. The function is currently called `harmonize_waves`.

Value

A data frame containing the row-wise combination of all surveys, with harmonized labelled variables and preserved attributes describing the original surveys.

See Also

Other harmonization functions: collect_val_labels(), crosswalk_surveys(), harmonize_na_values(), harmonize_values(), harmonize_var_names(), is.crosswalk_table(), label_normalize()

Examples

```
examples_dir <- system.file("examples", package = "retroharmonize")
survey_files <- dir(examples_dir, pattern = "\\.rds$", full.names = TRUE)

surveys <- read_surveys(
  survey_files,
  export_path = NULL
)

# Keep only supported variable types
surveys <- lapply(
  surveys,
  function(s) {
    s[, vapply(
      s,
      function(x) {
        inherits(x, c(
          "retroharmonize_labelled_spss_survey",
          "numeric",
          "character",
          "Date"
        ))
      },
      logical(1)
    )]
  }
)

# Identity harmonization (no-op)
harmonized <- harmonize_survey_values(
  survey_list = surveys,
  .f = function(x) x,
  status_message = FALSE
)

head(harmonized)
```

`harmonize_survey_variables`

Read a survey from a CSV file

Description

Import a survey stored in a CSV file and return it as a survey object with attached dataset- and survey-level metadata.

Usage

```
harmonize_survey_variables(
  crosswalk_table,
  subset_name = "subset",
  survey_list = NULL,
  survey_paths = NULL,
  import_path = NULL,
  export_path = NULL
)
```

Arguments

crosswalk_table	A crosswalk table created with [crosswalk_table_create()].
subset_name	Character string appended to filenames of subsetted surveys. Defaults to ‘"subset"‘.
survey_list	A list containing surveys of class survey.
survey_paths	Optional character vector of file paths to surveys.
import_path	Optional base directory used to resolve ‘survey_paths‘.
export_path	Optional directory where subsetted surveys are exported to

Details

The CSV file is read using [utils::read.csv()]. Character variables with more than one unique value are automatically converted to labelled factors. A unique row identifier is added and labelled.

If the file cannot be read, an empty survey object is returned with a warning.

If a column named ‘"X"‘ is present (commonly created by ‘write.csv()‘), it is removed automatically.

Value

An object of class ‘"survey"‘, which is a data frame with attached survey- and dataset-level metadata.

See Also

[read_rds()] for importing surveys from RDS files, [survey_df()] for constructing survey objects manually.

Other import functions: `pull_survey()`, `read_csv()`, `read_dta()`, `read_rds()`, `read_spss()`, `read_surveys()`

Examples

```
# Create a temporary CSV file from an example survey
path <- system.file("examples", "ZA7576.rds",
  package = "retroharmonize"
)
```

```
survey <- read_rds(path)

tmp <- tempfile(fileext = ".csv")
write.csv(survey, tmp, row.names = FALSE)

# Read the CSV file back as a survey
re_read <- read_csv(
  file = tmp,
  id = "ZA7576",
  doi = "10.0000/example"
)
```

`harmonize_values` *Harmonize values and labels of labelled vectors*

Description

'harmonize_values()' converts heterogeneous labelled survey vectors into a harmonized representation suitable for cross-survey integration.

The function:

- harmonizes value labels using regex-based matching; - assigns harmonized numeric codes; - preserves original coding metadata; - standardizes user-defined missing values; - preserves SPSS-style labelled metadata; - and records provenance attributes.

Usage

```
harmonize_values(
  x,
  harmonize_label = NULL,
  harmonize_labels = NULL,
  na_values = c(do_not_know = 99997, declined = 99998, inap = 99999),
  na_range = NULL,
  id = "survey_id",
  name_orig = NULL,
  remove = NULL,
  perl = FALSE
)
```

Arguments

`x`	A labelled vector, typically of class '"haven_labelled"' or '"haven_labelled_spss"'.
`harmonize_label`	Optional harmonized variable label. Defaults to the original variable label.
`harmonize_labels`	A list describing harmonization rules. Must contain the elements: - 'from' - 'to' - 'numeric_values'

na_values	Named numeric vector defining harmonized missing value codes.
na_range	Optional SPSS-style missing value range. Usually left ‘NULL‘.
id	Survey identifier. Defaults to ‘"survey_id"‘.
name_orig	Optional original variable name. Defaults to the object name supplied to ‘x‘.
remove	Optional regex pattern removed from original labels before harmonization.
perl	Logical. Use Perl-compatible regular expressions? Defaults to ‘FALSE‘.

Details

Create a harmonized labelled vector with standardized value labels, numeric coding, and missing value definitions.

Harmonization is performed using a harmonization table supplied via ‘harmonize_labels‘.

The harmonization table must contain:

- ‘from‘: regex patterns matching original labels; - ‘to‘: harmonized labels; - ‘numeric_values‘: harmonized numeric codes.

Original labels and numeric codes are preserved in attributes attached to the returned vector.

If no harmonization table is supplied, the function still attempts to normalize common missing value labels such as:

- ‘"inap"‘ - ‘"declined"‘ - ‘"do_not_know"‘

Value

A harmonized ‘haven_labelled_spss‘ vector.

The returned vector preserves:

- harmonized value labels; - harmonized numeric coding; - SPSS missing value metadata; - original coding metadata; - survey provenance metadata.

See Also

[harmonize_var_names()]

Other harmonization functions: `collect_val_labels()`, `crosswalk_surveys()`, `harmonize_na_values()`, `harmonize_survey_values()`, `harmonize_var_names()`, `is.crosswalk_table()`, `label_normalize()`

Examples

```
var1 <- labelled::labelled_spss(
  x = c(1, 0, 1, 1, 0, 8, 9),
  labels = c(
    "TRUST" = 1,
    "NOT TRUST" = 0,
    "DON'T KNOW" = 8,
    "INAP. HERE" = 9
  ),
  na_values = c(8, 9)
)
```

```
harmonize_values(
  var1,
  harmonize_labels = list(
    from = c(
      "^tend\\sto|^trust",
      "^tend\\snot|not\\strust",
      "^dk|^don",
      "^inap"
    ),
    to = c(
      "trust",
      "not_trust",
      "do_not_know",
      "inap"
    ),
    numeric_values = c(
      1,
      0,
      99997,
      99999
    )
  ),
  na_values = c(
    "do_not_know" = 99997,
    "inap" = 99999
  ),
  id = "survey_id"
)
```

`harmonize_var_names` *Harmonize variable names across surveys*

Description

'harmonize_var_names()' renames variables across multiple surveys to a shared harmonized naming scheme.

The harmonization rules are defined in a metadata table, typically created with [metadata_create()].

Usage

```
harmonize_var_names(
  survey_list,
  metadata,
  old = "var_name_orig",
  new = "var_name_suggested",
  rowids = TRUE
)
```

Arguments

survey_list	A list of survey objects, typically imported with [read_surveys()].
metadata	A metadata table containing harmonization rules. Typically created with [metadata_create()] and combined across surveys.
old	Name of the column in ‘metadata‘ containing the original variable names.
new	Name of the column in ‘metadata‘ containing the harmonized variable names.
rowids	Logical. Should original ‘rowid‘ variables be renamed to ‘"uniqid"‘?

Details

Harmonize variable names in a list of survey objects using a metadata crosswalk table.

The function can also be used for survey subsetting workflows. If ‘metadata‘ contains only a subset of variables for a survey, only those variables are retained in the harmonized output.

Value

A list of surveys with harmonized variable names.

See Also

[metadata_create()], [crosswalk()]

Other harmonization functions: collect_val_labels(), crosswalk_surveys(), harmonize_na_values(), harmonize_survey_values(), harmonize_values(), is.crosswalk_table(), label_normalize()

Examples

```
examples_dir <- system.file(
  "examples",
  package = "retroharmonize"
)

survey_files <- dir(
  examples_dir,
  pattern = "\\.rds$"
)

example_surveys <- read_surveys(
  file.path(examples_dir, survey_files)
)

metadata <- metadata_create(
  example_surveys
)

metadata$var_name_suggested <-
  label_normalize(metadata$var_name)

metadata$var_name_suggested[
  metadata$label_orig == "age_education"
```

```
] <- "age_education"

harmonized_surveys <- harmonize_var_names(
  survey_list = example_surveys,
  metadata = metadata
)

harmonized_surveys[[1]]
```

`is.crosswalk_table` *Validate a crosswalk table*

Description

Create a crosswalk table with the source variable names and variable labels.

Usage

```
is.crosswalk_table(ctable)

crosswalk_table_create(metadata)
```

Arguments

`ctable`	A table to validate if it is a crosswalk table.
`metadata`	A metadata table created by [metadata_create()].

Details

The table contains a `var_name_target` and `val_label_target` column, but these values need to be set by further manual or reproducible harmonization steps.

Value

A tibble with raw crosswalk table. It contains all harmonization tasks, but the target values need to be set by further manipulations.

See Also

Other metadata functions: `create_codebook()`, `metadata_create()`, `metadata_survey_create()`

Other harmonization functions: `collect_val_labels()`, `crosswalk_surveys()`, `harmonize_na_values()`, `harmonize_survey_values()`, `harmonize_values()`, `harmonize_var_names()`, `label_normalize()`

`is.survey_df` *Create a survey object*

Description

Construct a survey object from a data frame or tibble by attaching survey-level metadata such as an identifier, source filename, and basic dataset-level descriptive metadata.

Usage

```
is.survey_df(x)

survey_df(
  x,
  title = NULL,
  creator = person("Unknown", "Creator"),
  dataset_bibentry = NULL,
  dataset_subject = NULL,
  identifier,
  filename
)

is.survey_df(x)

## S3 method for class 'survey_df'
print(x, ...)
```

Arguments

`x`	A data frame or tibble containing the survey data.
`title`	Optional title for the survey. Defaults to ‘"Untitled Survey"‘.
`creator`	A [utils::person()] object describing the dataset creator. Defaults to ‘person("Unknown", "Creator")‘.
`dataset_bibentry`	Optional dataset-level bibliographic metadata. If ‘NULL‘, a minimal DataCite entry is created automatically using ‘title‘, ‘creator‘, and ‘dataset_subject‘.
`dataset_subject`	Dataset subject metadata. If ‘NULL‘, defaults to the Library of Congress Subject Heading Surveys.
`identifier`	A character scalar identifying the survey.
`filename`	A character scalar giving the source filename, or ‘NULL‘ if unknown.
`...`	potentially further arguments for methods.

Details

This function is primarily intended for use by import helpers such as [read_rds()], [read_spss()], [read_dta()], and [read_csv()]. Most users will not need to call it directly.

Value

An object of class ‘"survey_df"‘, which is a data frame with additional survey-level metadata stored as attributes and dataset-level metadata stored using the ‘dataset‘ package.

See Also

[read_survey()] for importing survey data from external files.

Other importing functions: `survey()`

Examples

```
survey_df(
  x = data.frame(
    rowid = 1:6,
    observations = runif(6)
  ),
  identifier = "example",
  filename = "no_file"
)
```

---

labelled_spss_survey_coercion

Coercion methods for labelled survey vectors

Description

Convert labelled SPSS-style survey vectors to common R data types. These helpers provide consistent coercion behavior for ‘"retroharmonize_labelled_spss_survey"‘ objects while respecting labelled missing values.

Usage

```
as_numeric(x)

as_character(x)

as_factor(x, levels = "default", ordered = FALSE)
```

Arguments

x	A labelled survey vector created with [labelled_spss_survey()].
levels	Character string indicating how factor levels should be constructed. Currently retained for compatibility.
ordered	Logical; whether the resulting factor should be ordered. Currently ignored.

Value

* ‘as_numeric()‘ returns a numeric vector with labelled missing values converted to ‘NA‘. * ‘as_character()‘ returns a character vector based on the factor representation of ‘x‘. * ‘as_factor()‘ returns a factor with levels derived from value labels.

See Also

[labelled_spss_survey()], [haven::as_factor()]

Other type conversion functions: `as_labelled_spss_survey()`

`label_normalize` *Normalize value and variable labels*

Description

`label_normalize` removes special characters, whitespace, and other typical typing errors.

Usage

```
label_normalize(x)

var_label_normalize(x)

val_label_normalize(x)
```

Arguments

x	A character vector of labels to be normalized.

Details

`var_label_normalize` and `val_label_normalize` removes possible chunks from question identifiers.

The functions `var_label_normalize` and `val_label_normalize` may be differently implemented for various survey series.

Value

Returns a suggested, normalized label without special characters. The `var_label_normalize` and `val_label_normalize` returns them in `snake_case` for programmatic use.

See Also

Other variable label harmonization functions: `na_range_to_values()`

Other harmonization functions: `collect_val_labels()`, `crosswalk_surveys()`, `harmonize_na_values()`, `harmonize_survey_values()`, `harmonize_values()`, `harmonize_var_names()`, `is.crosswalk_table()`

Examples

```
label_normalize(
  c(
    "Don't know", " TRUST", "DO NOT  TRUST",
    "inap in Q.3", "Not 100%", "TRUST < 50%",
    "TRUST >=90%", "Verify & Check", "TRUST 99%+"
  )
)

var_label_normalize(
  c(
    "Q1_Do you trust the national government?",
    " Do you trust the European Commission"
  )
)

val_label_normalize(
  c(
    "Q1_Do you trust the national government?",
    " Do you trust the European Commission"
  )
)
```

`merge_surveys` *Merge and harmonize surveys*

Description

'merge_surveys()' applies a harmonization specification to a list of survey objects and returns harmonized survey datasets with aligned variable names and metadata.

Usage

```
merge_surveys(survey_list, var_harmonization)
```

Arguments

`survey_list`	A list of survey objects.
`var_harmonization`	A metadata table describing the harmonization rules. The table must contain at least: - 'filename' - 'var_name_orig' - 'var_name_target' - 'var_label'

Details

Harmonize variable names, labels, and identifiers across multiple surveys using a metadata crosswalk table.

Prior to version 0.2.0 this function was called ‘merge_waves()‘, reflecting terminology commonly used in Eurobarometer surveys.

The harmonization table supplied in ‘var_harmonization‘ typically originates from [metadata_create()] and contains mappings between original and harmonized variable names.

Value

A list of harmonized survey objects with standardized variable names and variable labels.

See Also

[metadata_create()]

Other survey harmonization functions: `merge_waves()`

Examples

```
examples_dir <- system.file(
  "examples",
  package = "retroharmonize"
)

survey_files <- dir(
  examples_dir,
  pattern = "\\.rds$",
  full.names = TRUE
)

example_surveys <- read_surveys(
  survey_files
)

metadata <- metadata_create(
  survey_list = example_surveys
)

to_harmonize <- metadata %>%
  dplyr::filter(
    var_name_orig %in% c("rowid", "w1") |
      grepl("^trust", var_label_orig)
  ) %>%
  dplyr::mutate(
    var_label = var_label_normalize(var_label_orig),
    var_name_target = val_label_normalize(var_label),
    var_name_target = ifelse(
      .data$var_name_orig %in%
        c("rowid", "w1", "wex"),
      .data$var_name_orig,
      .data$var_name_target
    )
  )
```

```
merged_surveys <- merge_surveys(
  survey_list = example_surveys,
  var_harmonization = to_harmonize
)

merged_surveys[[1]]
```

`merge_waves` *Deprecated wrapper for 'merge_surveys()'*

Description

'merge_waves()' has been renamed to [merge_surveys()] for more general survey harmonization workflows.

Usage

```
merge_waves(waves, var_harmonization)
```

Arguments

`waves`	Deprecated alias for 'survey_list'.
`var_harmonization`	A metadata table describing the harmonization rules. The table must contain at least: - 'filename' - 'var_name_orig' - 'var_name_target' - 'var_label'

Value

A list of harmonized survey objects.

See Also

[merge_surveys()]

Other survey harmonization functions: `merge_surveys()`

`metadata_create` *Create metadata tables from survey datasets*

Description

Create a variable-level metadata table from one or more survey datasets. Metadata are extracted either from survey objects already loaded into memory or directly from survey files.

Usage

```
metadata_create(survey_list = NULL, survey_paths = NULL, .f = NULL)

metadata_waves_create(survey_list)
```

Arguments

`survey_list`	Optional list of survey objects of class [survey()].
`survey_paths`	Optional character vector containing paths to survey files.
`.f`	Import function used to read surveys from ‘survey_paths‘. When ‘NULL‘, the import function is inferred from the file extension.

Details

The resulting metadata table contains information about:

- variable names and labels,
- storage classes,
- value labels,
- user-defined missing values,
- and missing value ranges.

‘metadata_create()‘ is a convenience wrapper around repeated [metadata_survey_create()] calls.

The form `metadata_waves_create` is deprecated.

Value

A data frame containing variable-level survey metadata.

See Also

[metadata_survey_create()], [create_variable_catalog()]

Other metadata functions: `create_codebook()`, `is.crosswalk_table()`, `metadata_survey_create()`

Examples

```
examples_dir <- system.file(
  "examples",
  package = "retroharmonize"
)

my_rds_files <- dir(examples_dir)[grepl(
  "\\.rds$",
  dir(examples_dir)
)]

example_surveys <- read_surveys(
  file.path(examples_dir, my_rds_files)
)

metadata_create(example_surveys)
```

`metadata_survey_create` *Create variable-level metadata from a survey dataset*

Description

Extract variable-level metadata from a survey dataset and return the result as a nested data frame.

Usage

```
metadata_survey_create(survey)
```

Arguments

`survey`	A survey object of class [survey()]. Survey objects are typically created with: • [read_rds()] • [read_spss()] • [read_dta()] • [read_csv()] • [read_survey()] Survey objects can also be created manually from a data frame with [survey()].

Details

The metadata table contains:

- variable names and labels,

- imported storage classes,
- value labels,
- user-defined missing values,
- missing value ranges,
- and summary counts of labelled categories.

For multiple surveys, use [metadata_create()], which applies ‘metadata_survey_create()‘ across a list of surveys or survey files.

Value

A nested data frame containing:

filename Original survey file name.

id Survey identifier.

var_name_orig Original variable name.

class_orig Imported storage class.

var_label_orig Original variable label.

labels List column of value labels.

valid_labels List column of non-missing value labels.

na_labels List column of user-defined missing labels.

na_range List column containing user-defined missing ranges.

n_labels Number of labelled categories.

n_valid_labels Number of non-missing categories.

n_na_labels Number of missing categories.

See Also

[metadata_create()], [create_variable_catalog()]

Other metadata functions: `create_codebook()`, `is.crosswalk_table()`, `metadata_create()`

Examples

```
metadata_survey_create(
  survey = read_rds(
    system.file(
      "examples",
      "ZA7576.rds",
      package = "retroharmonize"
    )
  )
)
```

`na_range_to_values` *Harmonize SPSS-style missing value ranges*

Description

Ensure consistency between SPSS-style missing value ranges (‘na_range‘) and explicit missing values (‘na_values‘) for labelled survey vectors.

Usage

```
na_range_to_values(x)
```

Arguments

x	A labelled vector created with [haven::labelled_spss()] or ‘retroharmonize_labelled_spss_survey‘.

Details

When both attributes are present, this function:

- adjusts the missing range if it conflicts with existing missing values,
- derives missing values from the range when necessary,
- leaves non-SPSS-labelled vectors unchanged.

This harmonization is important before joining, binding, or summarizing survey data.

Value

The input vector with harmonized ‘na_values‘ and ‘na_range‘ attributes. If no harmonization is needed, ‘x‘ is returned unchanged.

See Also

[labelled::na_range()], [labelled::na_values()], [as_numeric()]

Other variable label harmonization functions: `label_normalize()`

`print.retroharmonize_labelled_spss_survey` *Labelled SPSS-style vectors with survey provenance*

Description

Create a labelled vector compatible with [haven::labelled_spss()] that carries additional survey-level provenance metadata.

Usage

```
## S3 method for class 'retroharmonize_labelled_spss_survey'
print(x, ...)

labelled_spss_survey(
  x = double(),
  labels = NULL,
  na_values = NULL,
  na_range = NULL,
  label = NULL,
  id = NULL,
  name_orig = NULL
)

## S3 method for class 'retroharmonize_labelled_spss_survey'
x[i, ...]

## S3 method for class 'retroharmonize_labelled_spss_survey'
summary(object, ...)

## S3 replacement method for class 'retroharmonize_labelled_spss_survey'
names(x) <- value

## S3 method for class 'retroharmonize_labelled_spss_survey'
is.na(x)

## S3 method for class 'retroharmonize_labelled_spss_survey'
levels(x)

## S3 method for class 'retroharmonize_labelled_spss_survey'
format(x, ..., digits = getOption("digits"))

is.labelled_spss_survey(x)

## S3 method for class 'retroharmonize_labelled_spss_survey'
median(x, na.rm = TRUE, ...)
```

```
## S3 method for class 'retroharmonize_labelled_spss_survey'
quantile(x, probs, ...)

## S3 method for class 'retroharmonize_labelled_spss_survey'
weighted.mean(x, w, ...)

## S3 method for class 'retroharmonize_labelled_spss_survey'
mean(x, ...)

## S3 method for class 'retroharmonize_labelled_spss_survey'
sum(x, ...)
```

Arguments

x	A vector of values.
...	potentially further arguments for methods; not used in the default method.
labels	A named vector of value labels.
na_values	A vector of values to be treated as missing.
na_range	A numeric range defining missing values.
label	A variable label.
id	A character scalar identifying the survey.
name_orig	Original variable name. Defaults to the name of 'x'.
i	Index vector used for subsetting.
object	A labelled_spss_survey to summarize.
value	Replacement values used when assigning names.
digits	Number of digits to use in string representation in the format method.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
probs	numeric vector of probabilities with values in $[0, 1]$. (Values up to '2e-14' outside that range are accepted and moved to the nearby endpoint.)
w	a numerical vector of weights the same length as x giving the weights to use for elements of x.

Details

The resulting object behaves like a 'haven_labelled_spss' vector, but stores:

- a survey identifier;
- the original variable name;
- the original value coding.

Several arithmetic and statistical summary methods operate on the numeric representation of labelled survey vectors, converting SPSS-style missing values to 'NA' before computation.

You can coerce 'labelled_spss_survey' vectors to numeric, character or factor representation.

Value

An object of class ‘"retroharmonize_labelled_spss_survey"‘, extending [haven::labelled_spss()].

See Also

[haven::labelled_spss()], [as_factor()], [as_numeric()], [as_character()]

Examples

```
x <- labelled_spss_survey(
  x = c(1, 2, 9),
  labels = c(Yes = 1, No = 2),
  na_values = 9,
  id = "survey_1"
)

is.na(x)
as_factor(x)
```

`pull_survey` *Retrieve a survey from a survey list*

Description

‘pull_survey()‘ retrieves a survey object from a list created with [read_surveys()].

Surveys can be selected using:

- the survey identifier stored in the ‘"id"‘ attribute, or - the original source file name stored in the ‘"filename"‘ attribute.

Usage

```
pull_survey(survey_list, id = NULL, filename = NULL)
```

Arguments

`survey_list`	A list of ‘survey‘ objects.
`id`	Optional survey identifier.
`filename`	Optional source file name.

Details

Extract a single ‘survey‘ object from a list of surveys using either its survey identifier or source file name.

Either ‘id‘ or ‘filename‘ must be supplied.

The function throws an error if:

- neither argument is provided; - the requested survey cannot be found; - or multiple surveys match the query.

Value

A single ‘survey‘ object.

See Also

[read_surveys()]

Other import functions: harmonize_survey_variables(), read_csv(), read_dta(), read_rds(), read_spss(), read_surveys()

Examples

```
examples_dir <- system.file(
  "examples",
  package = "retroharmonize"
)

survey_files <- dir(
  examples_dir,
  pattern = "\\.rds$"
)

example_surveys <- read_surveys(
  file.path(examples_dir, survey_files)
)

pull_survey(
  example_surveys,
  id = "ZA5913"
)
```

---

read_csv *Read a survey dataset from a CSV file*

Description

Import a survey dataset stored in comma-separated value (‘.csv‘) format and convert it into a survey-compatible tibble with reproducibility metadata retained as attributes.

Usage

```
read_csv(file, id = NULL, doi = NULL, dataset_bibentry = NULL, ...)
```

Arguments

file	Path to a ‘.csv‘ file.
id	Optional dataset identifier. When omitted, the file name without extension is used.

doi Optional dataset DOI identifier.

dataset_bibentry
Optional bibliographic metadata created with [dataset::dublincore()] or [dataset::datacite()].

... Additional arguments passed to [utils::read.csv()].

Details

The imported object is returned as a tibble with additional survey metadata such as identifiers, DOI references, and optional dataset bibliographic metadata.

Value

A tibble-like survey object with metadata attributes retained for reproducible workflows.

See Also

Other import functions: harmonize_survey_variables(), pull_survey(), read_dta(), read_rds(), read_spss(), read_surveys()

Examples

```
# Create a temporary CSV file:
path <- system.file(
  "examples",
  "ZA7576.rds",
  package = "retroharmonize"
)

read_survey <- read_rds(path)

test_csv_file <- tempfile(fileext = ".csv")

write.csv(
  x = read_survey,
  file = test_csv_file,
  row.names = FALSE
)

# Read the CSV file:
re_read <- read_csv(
  file = test_csv_file,
  id = "ZA7576",
  doi = "test_doi"
)
```

`read_dta` *Read a Stata '.dta' survey file*

Description

Import a survey dataset stored in Stata '.dta' format and convert it into a 'survey' object with harmonized metadata and labelled variables.

Usage

```
read_dta(file, id = NULL, doi = NULL, .name_repair = "unique")
```

Arguments

`file`	Path to a Stata '.dta' file.
`id`	Optional survey identifier. Defaults to the file name without extension.
`doi`	Optional DOI identifier for the survey.
`.name_repair`	Strategy for repairing invalid or duplicated column names. Passed to [haven::read_dta()].

Details

This function wraps [haven::read_dta()] and adds:

- error handling, - survey metadata creation, - 'rowid' normalization, - preservation of variable labels, - conversion of labelled variables, - and provenance metadata.

Variable labels are preserved using the '"label"' attribute.

Labelled variables are converted to harmonized labelled survey vectors where possible. Variables that inherit from 'haven_labelled' but do not contain valid label definitions are converted back to standard vectors.

If the file cannot be read, the function returns an empty 'survey' object and emits a warning.

Value

A 'survey' object inheriting from 'data.frame' and 'tbl_df'.

See Also

Other import functions: `harmonize_survey_variables()`, `pull_survey()`, `read_csv()`, `read_rds()`, `read_spss()`, `read_surveys()`

Examples

```
path <- system.file(
  "examples",
  "iris.dta",
  package = "haven"
)

survey_object <- read_dta(path)

attr(survey_object, "id")
attr(survey_object, "filename")
```

`read_rds` *Read a survey from an '.rds' file*

Description

Import a serialized survey object stored in '.rds' format and return it as a 'survey' object with harmonized metadata attributes.

Usage

```
read_rds(file, dataset_bibentry = NULL, id = NULL, doi = NULL)
```

Arguments

`file`	Path to an '.rds' file containing a survey object.
`dataset_bibentry`	Optional bibliographic metadata created with [dataset::dublincore()] or [dataset::datacite()].
`id`	Optional survey identifier. Defaults to the file name without extension.
`doi`	Optional DOI identifier for the survey.

Details

This function restores survey objects previously saved with [base::saveRDS()] or exported from the 'retroharmonize' workflow. The returned object retains survey metadata and gains additional provenance attributes such as source file name and file size.

If the file cannot be read, an empty 'survey' object is returned and a warning is emitted.

The function:

- restores the serialized object, - validates source file information, - normalizes 'rowid', - records provenance metadata, - and stores object and source file sizes as attributes.

Value

A 'survey' object inheriting from 'data.frame' and 'tbl_df' with survey metadata attributes.

See Also

Other import functions: harmonize_survey_variables(), pull_survey(), read_csv(), read_dta(), read_spss(), read_surveys()

Examples

```
path <- system.file(
  "examples",
  "ZA7576.rds",
  package = "retroharmonize"
)

survey_object <- read_rds(path)

attr(survey_object, "id")
attr(survey_object, "filename")
attr(survey_object, "doi")
```

---

read_spss *Read SPSS survey files*

Description

Import SPSS survey files in '.sav', '.zsav', or '.por' format and convert them into harmonized 'survey' objects with preserved metadata, labelled variables, and provenance information.

Usage

```
read_spss(
  file,
  user_na = TRUE,
  dataset_bibentry = NULL,
  id = NULL,
  doi = NULL,
  .name_repair = "unique"
)
```

Arguments

file	Path to an SPSS survey file.
user_na	Logical. Should user-defined missing values be imported? Defaults to 'TRUE'.
dataset_bibentry	Optional bibliographic metadata created with [dataset::dublincore()] or [dataset::datacite()].
id	Optional survey identifier. Defaults to the file name without extension.
doi	Optional DOI identifier.
.name_repair	Strategy for repairing invalid or duplicated column names. Passed to [haven::read_spss()].

Details

This function wraps [haven::read_spss()] and adds:

- error handling, - harmonized survey metadata, - ‘rowid‘ creation and normalization, - preservation of variable labels, - conversion of labelled SPSS vectors, - handling of malformed labelled variables, - and provenance metadata.

‘read_sav()‘ reads both ‘.sav‘ and ‘.zsav‘ files. ‘read_por()‘ reads portable SPSS ‘.por‘ files. ‘read_spss()‘ automatically dispatches to the appropriate importer based on file extension.

Variables that inherit from ‘haven_labelled‘ but do not contain valid label definitions are converted to standard numeric or character vectors.

If a file cannot be imported, the function returns an empty ‘survey‘ object and emits a warning.

Value

A ‘survey‘ object inheriting from ‘data.frame‘ and ‘tbl_df‘.

Variable labels are stored in the ‘"label"‘ attribute of each variable.

Additional provenance metadata are stored as attributes, including:

- ‘"id"‘ - ‘"doi"‘ - ‘"object_size"‘ - ‘"source_file_size"‘

See Also

Other import functions: `harmonize_survey_variables()`, `pull_survey()`, `read_csv()`, `read_dta()`, `read_rds()`, `read_surveys()`

Examples

```
path <- system.file(
  "examples",
  "iris.sav",
  package = "haven"
)

survey_object <- read_spss(path)

attr(survey_object, "id")
attr(survey_object, "filename")
```

`retroharmonize` *retroharmonize: Retrospective harmonization of survey data files*

Description

The goal of `retroharmonize` is to facilitate retrospective (ex-post) harmonization of data, particularly survey data, in a reproducible manner. The package provides tools for organizing the metadata, standardizing the coding of variables, variable names and value labels, including missing values, and for documenting all transformations, with the help of comprehensive S3 classes.

import functions

Read data stored in formats with rich metadata, such as SPSS (.sav) files, and make them usable in a programmatic context.
`read_spss`: read an SPSS file and record metadata for reproducibility
`read_rds`: read an rds file and record metadata for reproducibility
`read_surveys`: programmatically read a list of surveys
`pull_survey`: pull a single survey from a survey list.

subsetting functions

`subset_surveys`: remove variables from surveys that cannot be harmonized.

variable name harmonization functions

`harmonize_survey_variables`: Create a list of surveys with harmonized variable names.

variable label harmonization functions

Create consistent coding and labelling.
`harmonize_values`: Harmonize the label list across surveys.
`harmonize_survey_values`: Create a list of surveys with harmonized value labels.
`na_range_to_values`: Make the `na_range` attributes, as imported from SPSS, consistent with the `na_values` attributes.
`label_normalize` removes special characters, whitespace, and other typical typing errors and helps the uniformization of labels and variable names.

survey harmonization functions

`merge_surveys`: Create a list of surveys with harmonized names and variable labels.
`crosswalk_surveys`: Create a list of surveys with harmonized variable names, harmonized value labels and harmonize R classes.
`crosswalk`: Create a joined data frame of surveys with harmonized variable names, harmonized value labels and harmonize R classes.

metadata functions

`metadata_create`: Createa metadata dataa from one or more `survey`.
`metadata_survey_create`: Create a joined metadata data frame from one survey.
`create_codebook` and `codebook_waves_create` `crosswalk_table_create`: Create an initial crosswalk table from a metadata data frame.

documentation functions

Make the workflow reproducible by recording the harmonization process. `document_survey_item`: Returns a list of the current and historic coding, labelling of the valid range and missing values or range, the history of the variable names and the history of the survey IDs. `document_surveys`: Document the key attributes surveys in a survey list.

type conversion functions

Consistently treat labels and SPSS-style user-defined missing values in the R language. `survey` helps constructing a valid survey data frame, and `labelled_spss_survey` helps creating a vector for a questionnaire item. `as_numeric`: convert to numeric values.
`as_factor`: convert to labels to factor levels.
`as_character`: convert to labels to characters.
`as_labelled_spss_survey`: convert labelled and labelled_spss vectors to labelled_spss_survey vectors.

Author(s)

Maintainer: Daniel Antal <daniel.antal@dataobservatory.eu> (ORCID)

Authors:

- Daniel Antal <daniel.antal@dataobservatory.eu> (ORCID)

Other contributors:

- Marta Kolczynska <mkolczynska@gmail.com> (ORCID) [contributor]

See Also

Useful links:

- https://retroharmonize.dataobservatory.eu/
- Report bugs at https://github.com/dataobservatory-eu/retroharmonize/issues

---

`subset_surveys` *Subset and optionally harmonize surveys*

Description

Subset one or more surveys by retaining a specified set of variables. Subsetting can be performed either on surveys already loaded in memory or directly from survey files on disk.

If a crosswalk table is supplied, variables are selected based on the variables listed for each survey in the crosswalk, and variable names can optionally be harmonized using ‘var_name_target‘.

This function replaces the deprecated helpers [subset_waves()] and [subset_save_surveys()].

Usage

```
subset_surveys(
  survey_list,
  survey_paths = NULL,
  rowid = "rowid",
  subset_name = "subset",
  subset_vars = NULL,
  crosswalk_table = NULL,
  import_path = NULL,
  export_path = NULL
)

subset_waves(waves, subset_vars = NULL)

subset_save_surveys(
  crosswalk_table,
  subset_name = "subset",
  survey_list = NULL,
  subset_vars = NULL,
  survey_paths = NULL,
  import_path = NULL,
  export_path = NULL
)
```

Arguments

survey_list	A list of survey objects created by [read_surveys()]. If ‘NULL‘, surveys are read from disk.
survey_paths	A character vector of full file paths to survey files. Used when ‘survey_list‘ is ‘NULL‘.
rowid	Name of the unique observation identifier column. Defaults to ‘"rowid"‘.
subset_name	Character string appended to filenames of subsetted surveys. Defaults to ‘"subset"‘.
subset_vars	Character vector of variable names to retain. If ‘NULL‘, all variables are retained.
crosswalk_table	Optional crosswalk table created with [crosswalk_table_create()]. If supplied, variables are selected per survey based on ‘var_name_orig‘, and variable names may be harmonized using ‘var_name_target‘.
import_path	Optional directory containing survey files. Used to resolve filenames when subsetting from disk.
export_path	Optional directory where subsetted surveys are saved as ‘.rds‘ files. If ‘NULL‘, surveys are returned in memory.
waves	A list of surveys imported with [read_surveys()].

Details

The function supports multiple workflows:

* **In-memory subsetting** using ‘survey_list‘ * **File-based subsetting** using ‘survey_paths‘ or ‘import_path‘ * **Crosswalk-driven subsetting**, where variables are selected per survey using a crosswalk table created by [crosswalk_table_create()]

If ‘export_path‘ is provided, subsetted surveys are written to disk as ‘.rds‘ files. Otherwise, subsetted surveys are returned in memory.

Value

Either: * a list of subsetted survey objects (if ‘export_path = NULL‘), or * a character vector of filenames written to ‘export_path‘.

See Also

[crosswalk_table_create()], [harmonize_survey_variables()], [read_surveys()]

Examples

```
examples_dir <- system.file("examples", package = "retroharmonize")
survey_files <- dir(examples_dir, pattern = "\\.rds$")

surveys <- read_surveys(
  file.path(examples_dir, survey_files),
  export_path = NULL
)

subset_surveys(
  survey_list = surveys,
  subset_vars = c("rowid", "isocntry", "qa10_1", "qa14_1"),
  subset_name = "example_subset"
)
```

`survey` *Create a survey data frame*

Description

Store the data of a survey in a tibble (data frame) with a unique survey identifier, import filename, and optional document object identifier.

Usage

```
survey(object = data.frame(), id = "survey_id", filename = NULL, doi = NULL)

is.survey(object)

## S3 method for class 'survey'
summary(object, ...)
```

Arguments

object	A tibble or data frame that contains the survey data.
id	A mandatory identifier for the survey.
filename	The import file name.
doi	Optional document object identifier (doi), can be omitted.
...	Arguments passed to summary method.

Details

Whilst you can create a survey object with this helper function, it is most likely that you will receive it with an importing function, i.e. read_rds, read_spss read_dta, read_csv or their common wrapper read_survey.

Value

A tibble with id, filename, doi metadata information.

See Also

Other importing functions: is.survey_df()

Examples

```
example_survey <- survey(
  object = data.frame(
    rowid = 1:6,
    observations = runif(6)
  ),
  id = "example",
  filename = "no_file"
)
```

Index