

Package ‘retroharmonize’

January 14, 2026

Type Package

Title Ex Post Survey Data Harmonization

Version 0.2.7

Date 2026-01-12

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, methods, 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

RoxygenNote 7.3.3

X-schema.org-isPartOf <http://ropengov.org/>

X-schema.org-keywords ropengov

NeedsCompilation no

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

Repository CRAN

Date/Publication 2026-01-14 08:10:07 UTC

Contents

as_labelled_spss_survey	3
collect_val_labels	3
concatenate	4
create_codebook	5
crosswalk_surveys	6
document_surveys	8
document_survey_item	9
harmonize_na_values	11
harmonize_survey_values	12
harmonize_survey_variables	13
harmonize_values	15
harmonize_var_names	17
is.crosswalk_table	18
is.survey_df	19
labelled_spss_survey	20
labelled_spss_survey_coercion	24
label_normalize	25
merge_surveys	26
metadata_create	27
metadata_survey_create	28
na_range_to_values	30
pull_survey	31
read_csv	32
read_dta	33
read_rds	34
read_spss	35
retroharmonize	36
subset_surveys	38
survey	40

Index	42
--------------	-----------

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	<i>Concatenate haven_labelled_spss vectors</i>
-------------	--

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	<i>Create a survey codebook</i>
-----------------	---------------------------------

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\(\)](#)

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)
```

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

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

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

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

Description

Document the current and historic coding and labelling of the variable.

Usage

```
document_survey_item(x)
```

Arguments

x A labelled_spss_survey vector from a single survey or concatenated from several surveys.

Value

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.

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)
)

h1 <- harmonize_values(
  x = var1,
  harmonize_label = "Do you trust the European Union?",
  harmonize_labels = list(
    from = c("^tend\\sto|^trust", "^tend\\snot|not\\s\\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 = "survey1",
)
```

```

h2 <- harmonize_values(
  x = var2,
  harmonize_label = "Do you trust the European Union?",
  harmonize_labels = list(
    from = c("^tend\\sto|^trust", "^tend\\snot|not\\sitrust", "^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 = "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)
```

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

<code>crosswalk_table</code>	A crosswalk table created with <code>[crosswalk_table_create()]</code> .
<code>subset_name</code>	Character string appended to filenames of subsetted surveys. Defaults to <code>"subset"</code> .
<code>survey_list</code>	A list containing surveys of class <code>survey</code> .
<code>survey_paths</code>	Optional character vector of file paths to surveys.
<code>import_path</code>	Optional base directory used to resolve <code>'survey_paths'</code> .
<code>export_path</code>	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	<i>Harmonize the values and labels of labelled vectors</i>
------------------	--

Description

Create a labelled vector with harmonized numeric coding and value labels.

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
harmonize_label	A character vector of 1L containing the new, harmonize variable label. Defaults to NULL, in which case it uses the variable label of x, unless it is also NULL.
harmonize_labels	A list of harmonization values
na_values	A named vector of na_values, the observations that are defined to be treated as missing in the SPSS-style coding.

na_range	A min, max range of na_range, the continuous missing value range. In most surveys this should be left NULL.
id	A survey ID, defaults to survey_id
name_orig	The original name of the variable. If left NULL it uses the latest name of the object x.
remove	Defaults to NULL. A character or regex that will be removed from all old value labels, like "\(" \)" for (and).
perl	Use perl-like regex? Defaults to FALSE.

Details

Create a labelled vector that contains in its metadata attributes the original labelling, the original numeric coding and the current labelling, with the numerical values representing the harmonized coding.

Value

A labelled vector that contains in its metadata attributes the original labelling, the original numeric coding and the current labelling, with the numerical values representing the harmonized coding.

See Also

Other harmonization functions: [collect_val_labels\(\)](#), [crosswalk_surveys\(\)](#), [harmonize_na_values\(\)](#), [harmonize_survey_values\(\)](#), [harmonize_var_names\(\)](#), [is.crosswalk_table\(\)](#), [label_normalize\(\)](#)

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	<i>Harmonize the variable names of surveys</i>
---------------------	--

Description

The function harmonizes the variable names of surveys (of class `survey`) that are imported from an external file as a wave.

Usage

```

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

```

Arguments

<code>survey_list</code>	A list of surveys imported with read_surveys
<code>metadata</code>	A metadata table created by <code>metadata_create</code> and binded together for all surveys in <code>survey_list</code> .
<code>old</code>	The column name in metadata that contains the old, not harmonized variable names.
<code>new</code>	The column name in metadata that contains the new, harmonized variable names.
<code>rowids</code>	Rename var labels of original vars <code>rowid</code> to simply <code>uniqid</code> ?

Details

If the metadata that contains subsetting information is subsetting, then it will subset the surveys in `survey_list`.

Value

The list of surveys with harmonized variable names.

See Also

`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_list <- dir(examples_dir)[grepl("\\.rds", dir(examples_dir))]

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

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"

harmonize_var_names(
  survey_list = example_surveys,
  metadata = metadata
)
```

is.crosswalk_table	<i>Validate a crosswalk table</i>
--------------------	-----------------------------------

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	<i>Create a survey object</i>
--------------	-------------------------------

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

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

Description

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

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

Usage

```
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, ...]

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'
print(x, ...)

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

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

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

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

## 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'
print(x, ...)

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

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

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

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

## 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

<code>x</code>	A vector of values.
<code>labels</code>	A named vector of value labels.
<code>na_values</code>	A vector of values to be treated as missing.
<code>na_range</code>	A numeric range defining missing values.
<code>label</code>	A variable label.
<code>id</code>	A character scalar identifying the survey.
<code>name_orig</code>	Original variable name. Defaults to the name of 'x'.
<code>i</code>	Index vector used for subsetting.
<code>...</code>	potentially further arguments for methods; not used in the default method.
<code>object</code>	A <code>labelled_spss_survey</code> to summarize.
<code>value</code>	Replacement values used when assigning names.
<code>digits</code>	Number of digits to use in string representation in the format method.
<code>na.rm</code>	a logical value indicating whether NA values should be stripped before the computation proceeds.
<code>probs</code>	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.)
<code>w</code>	a numerical vector of weights the same length as <code>x</code> giving the weights to use for elements of <code>x</code> .

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 (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.

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 (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()].

An object of class “retroharmonize_labelled_spss_survey”, extending [haven::labelled_spss()].

See Also

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

[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)
```

```
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)
```

```
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	<i>Normalize value and variable labels</i>
-----------------	--

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\(\)](#)

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 surveys

Description

Merge a list of surveys into a list with harmonized variable names, variable labels, and survey identifiers.

Usage

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

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

Arguments

`survey_list` A list of surveys.

`var_harmonization`

A metadata table describing how variables should be harmonized. It must contain at least the columns `filename`, `var_name_orig`, `var_name_target`, and `var_label`.

`waves` Deprecated.

Details

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

Value

A list of surveys with harmonized variable names and labels.

See Also

metadata_create

Examples

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

example_surveys <- read_surveys(
  survey_files,
  save_to_rds = FALSE
)

# Create metadata from surveys
metadata <- metadata_create(survey_list = example_surveys)

# Select and harmonize a subset of variables
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 = var_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]]
```

metadata_create

Create a metadata table from several surveys

Description

Create a metadata table from several surveys

Usage

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

metadata_waves_create(survey_list)
```

Arguments

`survey_list` A list containing surveys of class `survey`.

`survey_paths` Optional character vector of file paths to surveys.

`.f` A function to import the surveys with.

Details

The form `metadata_waves_create` is deprecated.

See Also

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 a metadata table

Description

Create a metadata table from the survey data files.

Usage

```
metadata_survey_create(survey)
```

Arguments

`survey` A survey data frame. You receive a survey object with any importing function, i.e. [read_rds](#), [read_spss](#), [read_dta](#), [read_csv](#) or their common wrapper [read_survey](#). You can construct it with [survey](#) from a data frame, too.

Details

A data frame like tibble object is returned. In case you are working with several surveys, a list of surveys or a vector of file names containing the full path to the survey must be called with `[metadata_create()]`, which is a wrapper around a list of `[metadata_survey_create()]` calls.

The structure of the returned tibble:

filename The original file name; if present; missing, if a non-[survey](#) data frame is used as input survey.

id The ID of the survey, if present; missing, if a non-[survey](#) data frame is used as input survey.

var_name_orig The original variable name in SPSS.

class_orig The original variable class after importing with [read_spss](#).

var_label_orig The original variable label in SPSS.

labels A list of the value labels.

valid_labels A list of the value labels that are not marked as missing values.

na_labels A list of the value labels that refer to user-defined missing values.

na_range An optional range of a continuous missing range, if present in the vector.

n_labels Number of categories or unique levels, which may be different from the sum of missing and category labels.

n_valid_labels Number of categories in the non-missing range.

n_na_labels Number of categories of the variable, should be the sum of the former two.

na_levels A list of the user-defined missing values.

Value

A nested data frame with metadata and the range of labels, `na_values` and the `na_range` itself.

See Also

Other metadata functions: [create_codebook\(\)](#), [is.crosswalk_table\(\)](#), [metadata_create\(\)](#)

Examples

```
metadata_create(
  survey_list = read_rds(
    system.file("examples", "ZA7576.rds",
      package = "retroharmonize"
    )
  )
)
```

na_range_to_values	<i>Harmonize SPSS-style missing value ranges</i>
--------------------	--

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 <code>[haven::labelled_spss()]</code> or <code>'retroharmonize_labelled_spss_survey'</code> .
---	--

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\(\)](#)

pull_survey	<i>Pull a survey from a survey list</i>
-------------	---

Description

Pull a survey by survey code or id.

Usage

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

Arguments

survey_list	A list of surveys
id	The id of the requested survey. If NULL use filename
filename	The filename of the requested survey.

Value

A single survey identified by id or filename.

See Also

Other import functions: [harmonize_survey_variables\(\)](#), [read_csv\(\)](#), [read_dta\(\)](#), [read_rds\(\)](#), [read_spss\(\)](#), [read_surveys\(\)](#)

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)
)

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

read_csv	<i>Read csv file</i>
----------	----------------------

Description

Import a survey from a csv file.

Usage

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

Arguments

file	A path to a file to import.
id	An identifier of the tibble, if omitted, defaults to the file name without its extension.
doi	An optional document object identifier.
dataset_bibentry	A bibliographic entry created with <code>dataset::dublincore</code> or <code>dataset::datacite</code> .
...	Further optional parameters to pass on to <code>utils::read.csv</code> .

Value

A tibble, data frame variant with survey attributes.

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()
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	<i>Read Stata DTA files (‘.dta’) files</i>
----------	--

Description

This is a wrapper around `haven::read_dta` with some exception handling.

Usage

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

Arguments

<code>file</code>	A STATA file.
<code>id</code>	An identifier of the tibble, if omitted, defaults to the file name without its extension.
<code>doi</code>	An optional document object identifier.
<code>.name_repair</code>	Defaults to "unique" See <code>tibble::as_tibble</code> for details.

Details

`'read_dta()'` reads both `'dta'` files.

The function is not yet tested.

Value

A tibble.

Variable labels are stored in the "label" attribute of each variable. It is not printed on the console, but the RStudio viewer will show it.

`'write_sav()'` returns the input `'data'` invisibly.

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")
read_dta(path)
```

read_rds	<i>Read rds file</i>
----------	----------------------

Description

Import a survey from an rds file.

Usage

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

Arguments

file	A path to a file to import.
dataset_bibentry	A bibliographic entry created with <code>dataset::dublincore</code> or <code>dataset::datacite</code> .
id	An identifier of the tibble, if omitted, defaults to the file name without its extension.
doi	An optional document object identifier.

Value

A tibble, data frame variant with survey 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")
read_survey <- read_rds(path)
attr(read_survey, "id")
attr(read_survey, "filename")
attr(read_survey, "doi")
```

read_spss	<i>Read SPSS ('.sav', '.zsav', '.por') files. Write '.sav' and '.zsav' files.</i>
-----------	---

Description

This is a wrapper around `haven::read_spss` with some exception handling.

Usage

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

Arguments

<code>file</code>	An SPSS file.
<code>user_na</code>	Should user-defined na_values be imported? Defaults to TRUE.
<code>dataset_bibentry</code>	A bibliographic entry created with <code>dataset::dublincore</code> or <code>dataset::datacite</code> .
<code>id</code>	An identifier of the tibble, if omitted, defaults to the file name without its extension.
<code>doi</code>	An optional document object identifier.
<code>.name_repair</code>	Defaults to "unique" See <code>tibble::as_tibble</code> for details.

Details

`'read_sav()'` reads both `'sav'` and `'zsav'` files; `'write_sav()'` creates `'zsav'` files when `'compress = TRUE'`. `'read_por()'` reads `'por'` files. `'read_spss()'` uses either `'read_por()'` or `'read_sav()'` based on the file extension.

When the SPSS file has columns which are of class labelled, but have no labels, they are read as numeric or character vectors.

Value

A tibble:

Variable labels are stored in the "label" attribute of each variable. It is not printed on the console, but the RStudio viewer will show it.

`'write_sav()'` returns the input `'data'` invisibly.

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")
haven::read_sav(path)

tmp <- tempfile(fileext = ".sav")
haven::write_sav(mtcars, tmp)
haven::read_sav(tmp)
```

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](#): Create a metadata data frame 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](#))

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

- * **documentation functions**
 - document_survey_item, [9](#)
 - document_surveys, [8](#)
- * **harmonization functions**
 - collect_val_labels, [3](#)
 - crosswalk_surveys, [6](#)
 - harmonize_na_values, [11](#)
 - harmonize_survey_values, [12](#)
 - harmonize_values, [15](#)
 - harmonize_var_names, [17](#)
 - is.crosswalk_table, [18](#)
 - label_normalize, [25](#)
- * **import functions**
 - harmonize_survey_variables, [13](#)
 - pull_survey, [31](#)
 - read_csv, [32](#)
 - read_dta, [33](#)
 - read_rds, [34](#)
 - read_spss, [35](#)
- * **importing functions**
 - is.survey_df, [19](#)
 - survey, [40](#)
- * **joining functions**
 - concatenate, [4](#)
- * **metadata functions**
 - create_codebook, [5](#)
 - is.crosswalk_table, [18](#)
 - metadata_create, [27](#)
 - metadata_survey_create, [28](#)
- * **subsetting functions**
 - subset_surveys, [38](#)
- * **survey harmonization functions**
 - merge_surveys, [26](#)
- * **type conversion functions**
 - as_labelled_spss_survey, [3](#)
 - labelled_spss_survey_coercion, [24](#)
- * **variable label harmonization functions**
 - label_normalize, [25](#)
 - na_range_to_values, [30](#)
- [.retroharmonize_labelled_spss_survey
(labelled_spss_survey), [20](#)
- as_character, [37](#)
- as_character
(labelled_spss_survey_coercion),
[24](#)
- as_factor, [37](#)
- as_factor
(labelled_spss_survey_coercion),
[24](#)
- as_labelled_spss_survey, [3](#), [24](#), [37](#)
- as_numeric, [37](#)
- as_numeric
(labelled_spss_survey_coercion),
[24](#)
- as_tibble, [33](#), [35](#)
- codebook_surveys_create
(create_codebook), [5](#)
- codebook_waves_create, [37](#)
- codebook_waves_create
(create_codebook), [5](#)
- collect_na_labels (collect_val_labels),
[3](#)
- collect_val_labels, [3](#), [8](#), [11](#), [13](#), [16](#), [17](#), [19](#),
[25](#)
- concatenate, [4](#)
- create_codebook, [5](#), [19](#), [28](#), [29](#), [37](#)
- crosswalk, [37](#)
- crosswalk (crosswalk_surveys), [6](#)
- crosswalk_surveys, [4](#), [6](#), [11](#), [13](#), [16](#), [17](#), [19](#),
[25](#), [37](#)
- crosswalk_table_create, [37](#)
- crosswalk_table_create
(is.crosswalk_table), [18](#)
- datacite, [32](#), [34](#), [35](#)
- document_survey_item, [9](#), [9](#), [37](#)
- document_surveys, [8](#), [9](#), [10](#), [37](#)

document_waves (document_surveys), 8
 dublincore, 32, 34, 35
 format.retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20
 harmonize_na_values, 4, 8, 11, 13, 16, 17,
 19, 25
 harmonize_survey_values, 4, 8, 11, 12, 16,
 17, 19, 25, 37
 harmonize_survey_variables, 13, 31–34,
 36
 harmonize_values, 4, 8, 11, 13, 15, 17, 19,
 25, 37
 harmonize_var_names, 4, 8, 11, 13, 16, 17,
 19, 25
 harmonize_waves, 12
 harmonize_waves
 (harmonize_survey_values), 12
 is.crosswalk_table, 4, 6, 8, 11, 13, 16, 17,
 18, 25, 28, 29
 is.labelled_spss_survey
 (labelled_spss_survey), 20
 is.na.retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20
 is.survey (survey), 40
 is.survey_df, 19, 40
 label_normalize, 4, 8, 11, 13, 16, 17, 19, 25,
 30, 37
 labelled_spss_survey, 20, 37
 labelled_spss_survey_coercion, 3, 24
 levels.retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20
 mean.retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20
 median.retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20
 merge_surveys, 26, 37
 merge_waves (merge_surveys), 26
 metadata_create, 3, 6, 19, 27, 29, 37
 metadata_survey_create, 6, 19, 28, 28, 37
 metadata_waves_create
 (metadata_create), 27
 na_range_to_values, 25, 30, 37
 names<- .retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20
 print.retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20
 print.survey_df (is.survey_df), 19
 pull_survey, 14, 31, 32–34, 36
 quantile.retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20
 read.csv, 32
 read_csv, 14, 28, 31, 32, 33, 34, 36, 40
 read_dta, 14, 28, 31–33, 33, 34, 36, 40
 read_rds, 14, 28, 31–33, 34, 36, 40
 read_spss, 8, 14, 28, 29, 31–35, 35, 36, 40
 read_survey, 28, 40
 read_surveys, 14, 17, 31–34, 36
 retroharmonize, 36
 subset_save_surveys (subset_surveys), 38
 subset_surveys, 36, 38
 subset_waves (subset_surveys), 38
 sum.retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20
 summary.retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20
 summary.survey (survey), 40
 survey, 8, 20, 28, 29, 37, 40
 survey_df (is.survey_df), 19
 val_label_normalize (label_normalize),
 25
 var_label_normalize (label_normalize),
 25
 weighted.mean.retroharmonize_labelled_spss_survey
 (labelled_spss_survey), 20