

# Package ‘retroharmonize’

September 11, 2020

**Type** Package

**Title** Ex Post Survey Data Harmonization

**Version** 0.1.13

**Maintainer** Daniel Antal <daniel.antal@ceemid.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

**Encoding** UTF-8

**Language** en-US

**URL** <https://retroharmonize.dataobservatory.eu/>

**BugReports** <https://github.com/antaldaniel/retroharmonize/issues>

**LazyData** true

**Imports** vctrs,  
haven,  
dplyr (>= 1.0.0),  
magrittr,  
stats,  
tibble,  
labelled,  
methods,  
rlang,  
fs,  
assertthat,  
tidyselect,  
pillar,  
snakecase,  
purrr,  
tidyr,  
here

**RoxygenNote** 7.1.1

**Depends** R (>= 3.5.0)

**Suggests** knitr,  
 rmarkdown,  
 covr,  
 testthat,  
 png

**VignetteBuilder** knitr

## R topics documented:

as_factor . . . . .	2
as_labelled_spss_survey . . . . .	3
collect_val_labels . . . . .	3
concatenate . . . . .	4
document_survey_item . . . . .	5
document_waves . . . . .	6
harmonize_na_values . . . . .	7
harmonize_values . . . . .	8
harmonize_waves . . . . .	9
labelled_spss_survey . . . . .	10
label_normalize . . . . .	12
merge_waves . . . . .	13
metadata_create . . . . .	14
na_range_to_values . . . . .	15
pull_survey . . . . .	16
read_rds . . . . .	17
read_spss . . . . .	17
read_surveys . . . . .	19
retrohamonize . . . . .	20
subset_save_surveys . . . . .	21
survey . . . . .	22
<b>Index</b>	<b>23</b>

---

as\_factor

*Convert labelled\_spss\_survey vector To Factor*

---

### Description

Convert a [labelled\\_spss\\_survey](#) vector to a type of factor. Keeps only the levels and class attributes.

### Usage

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

**Arguments**

x	Object to coerce to a factor.
levels	How to create the levels of the generated factor: <ul style="list-style-type: none"><li>• "default": uses labels where available, otherwise the values. Labels are sorted by value.</li><li>• "both": like "default", but pastes together the level and value</li><li>• "label": use only the labels; unlabelled values become NA</li><li>• "values": use only the values</li></ul>
ordered	If TRUE create an ordered (ordinal) factor, if FALSE (the default) create a regular (nominal) factor.

**See Also**

as\_factor is imported from haven: [as\\_factor](#)

---

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\(\)](#)

---

`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: [harmonize\\_na\\_values\(\)](#), [harmonize\\_values\(\)](#), [harmonize\\_waves\(\)](#), [label\\_normalize\(\)](#), [merge\\_waves\(\)](#), [na\\_range\\_to\\_values\(\)](#)

### 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)
```

---

document\_survey\_item    *Document survey item harmonization*

---

**Description**

Document survey item harmonization

**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\\_waves\(\)](#)

**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\\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\\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 = "survey2"
)

h3 <- concatenate(h1, h2)
document_survey_item(h3)
```

---

document_waves	<i>Document survey lists</i>
----------------	------------------------------

---

**Description**

Document survey lists

**Usage**

```
document_waves(survey_list)
```

**Arguments**

survey\_list     A list of [survey](#) objects.

**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))
waves_document <- document_waves(example_surveys)
attr(waves_document, "original_list" )
waves_document
```

---

harmonize_na_values	<i>Harmonize na_values in haven_labelled_spss</i>
---------------------	---------------------------------------------------

---

**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\(\)](#), [harmonize\\_values\(\)](#), [harmonize\\_waves\(\)](#), [label\\_normalize\(\)](#), [merge\\_waves\(\)](#), [na\\_range\\_to\\_values\(\)](#)

**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\_values

*Harmonize the values and labels of labelled vectors*

---

**Description**

Harmonize the values and labels of labelled vectors

**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
)
```

**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.

### 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\(\)](#), [harmonize\\_na\\_values\(\)](#), [harmonize\\_waves\(\)](#), [label\\_normalize\(\)](#), [merge\\_waves\(\)](#), [na\\_range\\_to\\_values\(\)](#)

Other harmonization functions: [collect\\_val\\_labels\(\)](#), [harmonize\\_na\\_values\(\)](#), [harmonize\\_waves\(\)](#), [label\\_normalize\(\)](#), [merge\\_waves\(\)](#), [na\\_range\\_to\\_values\(\)](#)

### 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\_waves

*Harmonize waves*

---

### Description

Harmonize the values of surveys. It binds together variables that are all present in the surveys, and applies a harmonization function on them.

### Usage

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

**Arguments**

**waves**                A list of surveys  
**.f**                    A function to apply for the harmonization.  
**status\_message**    Defaults to FALSE. If set to TRUE it shows the id of the survey that is being joined.

**Value**

A natural full join of all surveys into a data frame.

**See Also**

Other harmonization functions: [collect\\_val\\_labels\(\)](#), [harmonize\\_na\\_values\(\)](#), [harmonize\\_values\(\)](#), [label\\_normalize\(\)](#), [merge\\_waves\(\)](#), [na\\_range\\_to\\_values\(\)](#)

**Examples**

```

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

example_surveys <- read_surveys(
  here( examples_dir, survey_list))

metadata <- lapply ( X = example_surveys, FUN = metadata_create )
metadata <- do.call(rbind, metadata)

to_harmonize <- metadata %>%
  filter ( var_name_orig %in%
           c("rowid", "w1") |
           grepl("trust ", label_orig ) ) %>%
  mutate ( var_label = var_label_normalize(label_orig)) %>%
  mutate ( var_name = val_label_normalize(var_label))

harmonize_eb_trust <- function(x) {
  label_list <- list(
    from = c("^tend\\snot", "^cannot", "^tend\\sto", "^can\\srely",
             "^dk", "^inap", "na"),
    to = c("not_trust", "not_trust", "trust", "trust",
           "do_not_know", "inap", "inap"),
    numeric_values = c(0,0,1,1, 99997,99999,99999)
  )

  harmonize_values(x,
    harmonize_labels = label_list,
    na_values = c("do_not_know"=99997,
                  "declined"=99998,
                  "inap"=99999)
  )
}

merged_surveys <- merge_waves ( example_surveys, var_harmonization = to_harmonize )

harmonized <- harmonize_waves(waves = merged_surveys,
  .f = harmonize_eb_trust,
  status_message = FALSE)

# For details see Afrobarometer and Eurobarometer Case Study vignettes.

```

---

labelled\_spss\_survey *Labelled vectors for multiple SPSS surveys*

---

## Description

This class is amending `haven::labelled_spss` with a unique object identifier `id` to make later binding or joining reproducible and well-documented.

## Usage

```
labelled_spss_survey(  
  x = double(),  
  labels = NULL,  
  na_values = NULL,  
  na_range = NULL,  
  label = NULL,  
  id = NULL,  
  name_orig = NULL  
)  
  
as_character(x)  
  
is.labelled_spss_survey(x)  
  
as_numeric(x)
```

## Arguments

<code>x</code>	A vector to label. Must be either numeric (integer or double) or character.
<code>labels</code>	A named vector or NULL. The vector should be the same type as <code>x</code> . Unlike factors, labels don't need to be exhaustive: only a fraction of the values might be labelled.
<code>na_values</code>	A vector of values that should also be considered as missing.
<code>na_range</code>	A numeric vector of length two giving the (inclusive) extents of the range. Use <code>-Inf</code> and <code>Inf</code> if you want the range to be open ended.
<code>label</code>	A short, human-readable description of the vector.
<code>id</code>	Survey ID
<code>name_orig</code>	The original name of the variable. If left NULL it uses the latest name of the object <code>x</code> .

## Details

It inherits many methods from `labelled`, but uses more strict coercion and validation rules.

## See Also

`as_factor`  
Other type conversion functions: [as\\_labelled\\_spss\\_survey\(\)](#)  
Other type conversion functions: [as\\_labelled\\_spss\\_survey\(\)](#)

## Examples

```
x1 <- labelled_spss_survey(
  1:10, c(Good = 1, Bad = 8),
  na_values = c(9, 10),
  id = "survey1")

is.na(x1)

# Print data and metadata
print(x1)

x2 <- labelled_spss_survey( 1:10,
  labels = c(Good = 1, Bad = 8),
  na_range = c(9, Inf),
  label = "Quality rating",
  id = "survey1")

is.na(x2)

# Print data and metadata
x2
```

---

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 changes the vector to snake\_case. 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.

**See Also**

Other harmonization functions: [collect\\_val\\_labels\(\)](#), [harmonize\\_na\\_values\(\)](#), [harmonize\\_values\(\)](#), [harmonize\\_waves\(\)](#), [merge\\_waves\(\)](#), [na\\_range\\_to\\_values\(\)](#)

Other harmonization functions: [collect\\_val\\_labels\(\)](#), [harmonize\\_na\\_values\(\)](#), [harmonize\\_values\(\)](#), [harmonize\\_waves\(\)](#), [merge\\_waves\(\)](#), [na\\_range\\_to\\_values\(\)](#)

Other harmonization functions: [collect\\_val\\_labels\(\)](#), [harmonize\\_na\\_values\(\)](#), [harmonize\\_values\(\)](#), [harmonize\\_waves\(\)](#), [merge\\_waves\(\)](#), [na\\_range\\_to\\_values\(\)](#)

**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\_waves

*Merge waves*

---

**Description**

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

**Usage**

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

**Arguments**

waves                    A list of surveys

var\_harmonization

Metadata of surveys, including at least filename, var\_name\_orig, var\_name, var\_label.

**Value**

A list of surveys with harmonized names and variable labels.

**See Also**

[survey](#)

Other harmonization functions: [collect\\_val\\_labels\(\)](#), [harmonize\\_na\\_values\(\)](#), [harmonize\\_values\(\)](#), [harmonize\\_waves\(\)](#), [label\\_normalize\(\)](#), [na\\_range\\_to\\_values\(\)](#)

**Examples**

```

survey_list <- dir (
  here( "inst", "examples"))[grepl(".rds",
  dir (here( "inst", "examples")))]

example_surveys <- read_surveys(
  here( "inst", "examples", survey_list))

metadata <- lapply ( X = example_surveys, FUN = metadata_create )
metadata <- do.call(rbind, metadata)

to_harmonize <- metadata %>%
  filter ( var_name_orig %in%
    c("rowid", "w1") |
    grepl("trust ", label_orig ) ) %>%
  mutate ( var_label = var_label_normalize(label_orig) ) %>%
  mutate ( var_name = var_label_normalize(var_label) )

merge_waves ( example_surveys, to_harmonize )

```

---

metadata_create	<i>Create a metadata table</i>
-----------------	--------------------------------

---

**Description**

Create a metadata table from the survey data files.

**Usage**

```
metadata_create(survey)
```

**Arguments**

survey            A survey data frame.

**Details**

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](#).

**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.

### Examples

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

---

na\_range\_to\_values      *Harmonize user-defined missing value ranges*

---

### Description

Harmonize the na\_values attribute with na\_range, if the latter is present.

### Usage

```
na_range_to_values(x)
```

```
is.na_range_to_values(x)
```

### Arguments

x                      A labelled\_spss or labelled\_spss\_survey vector

### Details

na\_range\_to\_values() tests if the function needs to be called for na\_values harmonization. The na\_range is often missing and less likely to cause logical problems when joining survey answers.

### Value

A x with harmonized na\_values and na\_range attributes. If min(na\_values) or max(na\_values) than the left- and right-hand value of na\_range, it gives a warning and adjusts the original na\_range.

### See Also

Other harmonization functions: [collect\\_val\\_labels\(\)](#), [harmonize\\_na\\_values\(\)](#), [harmonize\\_values\(\)](#), [harmonize\\_waves\(\)](#), [label\\_normalize\(\)](#), [merge\\_waves\(\)](#)

**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_range = c(8,12))

na_range_to_values(var1)
as_numeric(na_range_to_values(var1))
as_character(na_range_to_values(var1))
```

---

pull\_survey

*Pull a survey from a survey list*

---

**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: [read\\_rds\(\)](#), [read\\_spss\(\)](#), [read\\_surveys\(\)](#), [subset\\_save\\_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_rds	<i>Read survey from rds file</i>
----------	----------------------------------

---

### Description

Read survey from rds file

### Usage

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

### Arguments

file	A re-saved survey, imported with haven:: <a href="#">read_spss</a>
id	An identifier of the tibble, if omitted, defaults to the file name.
filename	An import file name.
doi	An optional document object identifier.

### Value

A tibble, data frame variant with survey attributes.

### See Also

Other import functions: [pull\\_survey\(\)](#), [read\\_spss\(\)](#), [read\\_surveys\(\)](#), [subset\\_save\\_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

'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.

## Usage

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

## Arguments

file	An SPSS file.
user_na	Should user-defined na_values be imported? Defaults to TRUE.
id	An identifier of the tibble, if omitted, defaults to the file name.
filename	An import file name.
doi	An optional document object identifier.
.name_repair	Defaults to "unique" See <code>tibble::as_tibble</code> for details.

## Details

This is a wrapper around `haven::read_spss`

## Value

A tibble, data frame variant with nice defaults.

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: `pull_survey()`, `read_rds()`, `read_surveys()`, `subset_save_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)
```

---

read_surveys	<i>Read Survey Files Import surveys into a list. Adds filename as a constant to each element of the list.</i>
--------------	---------------------------------------------------------------------------------------------------------------

---

## Description

Read Survey Files

Import surveys into a list. Adds filename as a constant to each element of the list.

## Usage

```
read_surveys(import_file_names, .f = "read_rds", save_to_rds = TRUE)
```

## Arguments

`import_file_names` A vector of file names to import.

`.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.

`save_to_rds` Should it save the imported survey to .rds? Defaults to TRUE.

## Value

A list of the surveys. Each element of the list is a data frame-like [survey](#) type object where some metadata, such as the original file name, doi identifier if present, and other information is recorded for a reproducible workflow.

## See Also

[survey](#)

Other import functions: [pull\\_survey\(\)](#), [read\\_rds\(\)](#), [read\\_spss\(\)](#), [subset\\_save\\_surveys\(\)](#)

## Examples

```
file1 <- system.file(
  "examples", "ZA7576.rds", package = "retroharmonize")
file2 <- system.file(
  "examples", "ZA5913.rds", package = "retroharmonize")

read_surveys (c(file1,file2), .f = 'read_rds' )
```

---

retrohamonize

*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

`subset_save_surveys`: programmatically read a list of surveys, and subset them (pre-harmonize the same variables.)

`pull_survey`: pull a single survey from a survey list.

## harmonization functions

Create consistent coding and labelling.

`harmonize_values`: `merge_waves`: Create a list of surveys with harmonized names and variable labels.

`harmonize_waves`: Create a list of surveys with harmonized value labels.

`label_normalize` removes special characters, whitespace, and other typical typing errors and helps the uniformization of labels and variable names.

`na_range_to_values`: Make the `na_range` attributes, as imported from SPSS, consistent with the `na_values` attributes.

## documentation functions

Make the workflow reproducible by recording the harmonization process.

## 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.

---

subset\_save\_surveys     *Subset and Save Surveys*

---

## Description

Read a predefined survey list and variables.

## Usage

```
subset_save_surveys(  
  var_harmonization,  
  selection_name = "trust",  
  import_path = "",  
  export_path = "working"  
)
```

## Arguments

`var_harmonization` Metadata of surveys, including at least filename, `var_name_orig`, `var_name`, `var_label`.

`selection_name` An identifier for the survey subset.

`import_path` The path to the survey files.

`export_path` The path where the subsets should be saved.

## Value

The function does not return a value. It saves the subsetted surveys into .rds files.

## See Also

Other import functions: [pull\\_survey\(\)](#), [read\\_rds\(\)](#), [read\\_spss\(\)](#), [read\\_surveys\(\)](#)

## Examples

```
test_survey <- read_rds (  
  file = system.file("examples", "ZA7576.rds",  
                    package = "retroharmonize")  
)  
  
test_metadata <- metadata_create ( test_survey )  
test_metadata <- test_metadata[c(18:37),]  
test_metadata$var_name <- var_label_normalize (test_metadata$var_name_orig)  
test_metadata$var_label <- test_metadata$label_orig  
  
saveRDS(test_survey, file.path(tempdir(),  
                               "ZA7576.rds"),  
        version = 2)  
  
subset_save_surveys ( var_harmonization = test_metadata,  
                    selection_name = "tested",  
                    import_path = tempdir(),
```

```
export_path = tempdir())  
file.exists ( file.path(tempdir(), "ZA7576_tested.rds"))
```

---

survey	<i>Survey data frame</i>
--------	--------------------------

---

### Description

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

### Usage

```
survey(  
  df = data.frame(),  
  id = character(),  
  filename = character(),  
  doi = character()  
)  
  
is.survey(df)
```

### Arguments

df	A tibble or data frame that contains the survey data.
id	A mandatory identifier for the survey
filename	The import file name.
doi	Optional doi, can be omitted.

### Value

A tibble with id, filename, doi metadata information.

### Examples

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

# Index

- \* **documentation functions**
  - document\_survey\_item, 5
  - document\_waves, 6
- \* **harmonization functions**
  - collect\_val\_labels, 3
  - harmonize\_na\_values, 7
  - harmonize\_values, 8
  - harmonize\_waves, 9
  - label\_normalize, 12
  - merge\_waves, 13
  - na\_range\_to\_values, 15
- \* **import functions**
  - pull\_survey, 16
  - read\_rds, 17
  - read\_spss, 17
  - read\_surveys, 19
  - subset\_save\_surveys, 21
- \* **joining functions**
  - concatenate, 4
- \* **type conversion functions**
  - as\_labelled\_spss\_survey, 3
  - labelled\_spss\_survey, 10
- as\_character, 20
- as\_character (labelled\_spss\_survey), 10
- as\_factor, 2, 3, 20
- as\_labelled\_spss\_survey, 3, 11, 20
- as\_numeric, 20
- as\_numeric (labelled\_spss\_survey), 10
- as\_tibble, 18
  
- collect\_na\_labels (collect\_val\_labels), 3
- collect\_val\_labels, 3, 7–9, 12, 13, 15
- concatenate, 4
  
- document\_survey\_item, 5, 6
- document\_waves, 5, 6
  
- harmonize\_na\_values, 4, 7, 8, 9, 12, 13, 15
- harmonize\_values, 4, 7, 8, 9, 12, 13, 15, 20
- harmonize\_waves, 4, 7, 8, 9, 12, 13, 15, 20
  
- is\_labelled\_spss\_survey (labelled\_spss\_survey), 10
  
- is.na\_range\_to\_values (na\_range\_to\_values), 15
- is\_survey (survey), 22
  
- label\_normalize, 4, 7–9, 12, 13, 15, 20
- labelled\_spss, 10
- labelled\_spss\_survey, 2, 3, 10, 20
  
- merge\_waves, 4, 7–9, 12, 13, 15, 20
- metadata\_create, 3, 14
  
- na\_range\_to\_values, 4, 7–9, 12, 13, 15, 20
  
- pull\_survey, 16, 17–21
  
- read\_rds, 16, 17, 18–21
- read\_spss, 14, 16, 17, 17, 18–21
- read\_surveys, 16–18, 19, 20, 21
- retroharmonize, 20
  
- subset\_save\_surveys, 16–20, 21
- survey, 6, 14, 19, 20, 22
  
- val\_label\_normalize (label\_normalize), 12
- var\_label\_normalize (label\_normalize), 12