Package 'retroharmonize'

December 15, 2021

Type Package

Title Ex Post Survey Data Harmonization

Version 0.2.4

Date 2021-12-14

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

URL https://retroharmonize.dataobservatory.eu/, https://github.com/rOpenGov/retroharmonize

BugReports https://github.com/rOpenGov/retroharmonize/issues

Depends R (>= 3.5.0)

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

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

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.1.2

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

X-schema.org-keywords ropengov

R topics documented:

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: • "default": uses labels where available, otherwise the values. Labels are sorted by value. • "both": like "default", but pastes together the level and value • "label": use only the labels; unlabelled values become NA • "values: use only the values
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: `crosswalk_surveys()`, `crosswalk_table_create()`, `harmonize_na_values()`, `harmonize_survey_values()`, `harmonize_values()`, `harmonize_var_names()`, `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 codebook*

Description

Create a codebook from one or more survey data files.

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`. Defaults to `NULL`.
`survey`	A survey data frame, defaults to `NULL`. If the survey is given as parameter, the metadata will be set to the metadata of this particular survey by `metadata_create`.
`waves`	A list of surveys.
`survey_list`	A list containing surveys of class survey.

Details

For a list of survey waves, use `codebook_waves_create`. The returned codebook contains only labelled variables, i.e., numeric and character types are not included, because they do not require coding.

Value

A codebook for the survey as a data frame, including the metadata, and all found SPSS-type valid or missing labels.

See Also

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

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

Examples

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

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 surveys*

Description

Harmonize surveys with crosswalk tables.

Usage

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

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

Arguments

`crosswalk_table`
: A table created with `crosswalk_table_create`, or a data frame with at least the following columns: `var_name_orig`, `var_name_target`, for harmonizing the variable names. If `val_label_orig`, `val_label_target` are present, the value labels will be harmonized, too. If `var_numeric_orig`, `var_numeric_target` are present, the numeric codes of the variable will be harmonized. If `class_target` is present, then the class of the variable will be harmonized to any of `factor`, `numeric` or `character` using `as_factor`, `as_numeric`, or `as_character`.

`survey_list`
: A list of surveys imported with `read_surveys`. If set to `NULL`, the `survey_path` should give full path to the surveys.

`na_values`
: A named vector of `na_values`, the observations that are defined to be treated as missing in the SPSS-style coding. Defaults to `NULL`.

Details

Harmonize a survey or a list of surveys with the help of a crosswalk table. You can create the crosswalk table with `crosswalk_table_create`, or manually create a crosswalk table as a data frame including at least the following columns: `id` for identifying a survey, `var_name_orig` for the original variable name and `var_name_target` for the new (target) variable name. Optionally you can harmonize the value labels, the numeric codes, and the special missing labels, too.

Value

crosswalk will return a data frame, and crosswalk_surveys a list of data frames, where the variable names, and optionally the variable labels, and the missing value range is harmonized (the same names, labels, codes are used.)

See Also

Other harmonization functions: collect_val_labels(), crosswalk_table_create(), harmonize_na_values(), harmonize_survey_values(), harmonize_values(), harmonize_var_names(), 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),
  save_to_rds = FALSE)

## Compare with documentation:
documented_surveys <- metadata_surveys_create(example_surveys)
documented_surveys <- documented_surveys[
documented_surveys$var_name_orig %in% c( "rowid", "isocntry", "w1", "qd3_4",
                                         "qd3_8" , "qd7.4", "qd7.8", "qd6.4", "qd6.8"),
                                         ]
crosswalk_table <- crosswalk_table_create ( metadata = documented_surveys )
```

---

crosswalk_table_create

Create a crosswalk table

Description

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

Usage

```
crosswalk_table_create(metadata)

is.crosswalk_table(ctable)
```

Arguments

metadata	A metadata table created by metadata_create.
ctable	A table to validate if it is a crosswalk table.

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

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

---

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_surveys <- document_surveys(survey_list=example_surveys)

attr(documented_surveys, "original_list")
documented_surveys

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\\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)
```

`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(), crosswalk_table_create(), harmonize_survey_values(), harmonize_values(), harmonize_var_names(), 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 the value codes and value labels across multiple surveys.

Usage

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

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

Arguments

survey_list	A list of surveys. In the deprecated form the parameter was called waves.
.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.
waves	A list of surveys. Deprecated.

Details

The functions binds together variables that are all present in the surveys, and applies a harmonization function .f on them. Till retroharmonize 0.2.0 called harmonize_waves.

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

Value

A natural full join of all surveys in a single data frame.

See Also

Other harmonization functions: collect_val_labels(), crosswalk_surveys(), crosswalk_table_create(), harmonize_na_values(), harmonize_values(), harmonize_var_names(), 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),
  save_to_rds = FALSE)

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

require(dplyr)

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

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

merged_surveys <- merge_surveys ( example_surveys, var_harmonization = to_harmonize  )

harmonized <- harmonize_survey_values(survey_list = merged_surveys,
                              .f = harmonize_eb_trust,
                              status_message = FALSE)

# For details see Afrobarometer and Eurobarometer Case Study vignettes.
```

harmonize_survey_variables *Harmonize survey variables*

Description

Similar to subset_surveys, but will not only remove the variables that cannot be harmonized, but renames the remaining variables.

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 by crosswalk_table_create or a manually created crosstable including at least filename, var_name_orig, var_name_target and optionally var_label_orig and var_label_target. This parameter is optional and defaults to NULL.
subset_name	An identifier for the survey subset.
survey_list	A list of surveys imported with read_surveys. If set to NULL, the survey_path should give full path to the surveys.
survey_paths	A vector of full file paths to the surveys to subset.

Value

A list of surveys or save individual rds files on the export_path.

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,
  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.

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(), crosswalk_table_create(), harmonize_na_values(), harmonize_survey_values(), harmonize_var_names(), label_normalize()

Other harmonization functions: collect_val_labels(), crosswalk_surveys(), crosswalk_table_create(), harmonize_na_values(), harmonize_survey_values(), harmonize_var_names(), 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 the variable names of surveys*

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

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

Details

If the `metadata` that contains subsetting information is subsetted, 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(), crosswalk_table_create(), harmonize_na_values(), harmonize_survey_values(), harmonize_values(), 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 )
```

---

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

x	A vector to label. Must be either numeric (integer or double) or character.

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

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 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(), crosswalk_table_create(), harmonize_na_values(), harmonize_survey_values(), harmonize_values(), harmonize_var_names()

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

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`	Metadata of surveys, including at least `filename`, `var_name_orig`, `var_name_target`, `var_label`.
`waves`	Deprecated.

Details

The function was called till version 0.2.0 `merge_waves()`, which reflects the vocabulary of Eurobarometer surveys.

Value

A list of surveys with harmonized names and variable labels.

See Also

survey

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),
  save_to_rds = FALSE)

metadata <- metadata_surveys_create(example_surveys)

require(dplyr)

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_target = val_label_normalize(var_label) ) %>%
  mutate ( var_name_target = ifelse(.data$var_name_orig %in% c("rowid", "w1", "wex"),
                                    .data$var_name_orig, .data$var_name_target) )

merge_surveys ( example_surveys, to_harmonize )
```

`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

`inheritParams`	read_surveys

Details

The form `metadata_waves_create` is deprecated.

See Also

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

Examples

```
metadata_create (
 survey_list = 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 variable label harmonization functions: `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_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_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 *Read csv file*

Description

Import a survey from a csv file.

Usage

```
read_csv(
  file,
  id = NULL,
  filename = NULL,
  doi = NULL,
  header = FALSE,
  sep = "",
  quote = "\"'",
  dec = ".",
  numerals = c("allow.loss", "warn.loss", "no.loss"),
  na.strings = "NA",
  skip = 0,
  check.names = TRUE,
  strip.white = FALSE,
  blank.lines.skip = TRUE,
  stringsAsFactors = FALSE,
  fileEncoding = "",
  encoding = "unknown"
)
```

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.

Value

A tibble, data frame variant with survey attributes.

See Also

Other import functions: pull_survey(), read_dta(), read_rds(), 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_dta *Read Stata DTA files ('.dta') files*

Description

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

Usage

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

Arguments

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

Details

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

The funcion 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: pull_survey(), read_csv(), read_rds(), read_spss(), read_surveys()

Examples

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

---

read_rds *Read rds file*

Description

Import a survey from an rds file.

Usage

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

Value

A tibble, data frame variant with survey attributes.

See Also

Other import functions: `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 *Read SPSS (‘.sav‘, ‘.zsav‘, ‘.por‘) files. Write ‘.sav‘ and ‘.zsav‘ files.*

Description

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

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 without its extension.
`doi`	An optional document object identifier.
`.name_repair`	Defaults to `"unique"` See `tibble::as_tibble` 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: `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)
```

---

read_surveys *Read survey file(s)*

Description

Import surveys into a list or several `.rds` files.

Usage

```
read_surveys(survey_paths, .f = NULL, export_path = NULL)

read_survey(file_path, .f = NULL, export_path = NULL)
```

Arguments

`survey_paths`	A vector of (full) file paths that contain the surveys to import.
`.f`	A function to import the surveys with. Defaults to `'NULL'`, in this case files with an extension of `'.sav'` and `'.por'` will call case `read_spss`, files with an extension of `'.dta'` will call `read_dta`, `rds` will call `read_rds` and `'.csv'` `read_csv`.
`export_path`	Defaults to `NULL`, in this case the read surveys are imported into a single list of surveys in memory. If `export_path` is a valid directory, it will instead save each survey an R object with `[base]saveRDS`.

Details

Use `read_survey` for a single survey and `read_surveys` for several surveys in in a loop. The function handle exceptions with wrong file names and not readable files. If a file cannot be read, a message is printed, and empty survey is added to the the list in the place of this file.

Value

A list of the surveys or a vector of the saved file names. 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_csv(), read_dta(), read_rds(), read_spss()

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' )
```

---

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

suggest_permanent_names: Suggest the use of variable naming conventions. 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 from one survey.
metadata_surveys_create: Create a joined metadata data frame from more than 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.

---

subset_surveys *Subset surveys*

Description

This is a wrapper function for various procedures to reduce the size of surveys by removing variables that are not harmonized.

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,
  survey_paths = NULL,
  import_path = NULL,
  export_path = NULL
)
```

Arguments

survey_list	A list of surveys imported with read_surveys. If set to NULL, the survey_path should give full path to the surveys.
survey_paths	A vector of full file paths to the surveys to subset.
rowid	The unique row (observation) identifier in the files. Defaults to "rowid", which is the default of the importing functions in this package.
subset_name	An identifier for the survey subset.
subset_vars	The names of the variables that should be kept from all surveys in the list that contains the wave of surveys. Defaults to NULL in which case it returns all variables without subsetting.
crosswalk_table	A crosswalk table created by crosswalk_table_create or a manually created crosstable including at least filename, var_name_orig, var_name_target and optionally var_label_orig and var_label_target. This parameter is optional and defaults to NULL.
waves	A list of surveys imported with read_surveys.

Details

This function allows several workflows. Subsetting can be based on a vector of variable names given by survey_path, or on the basis of a crosstable. The subset_save_surveys can be called directly.

subset_surveys will also harmonize the variable names if the var_name_target is optionally defined in the crosswalk_table input. harmonize_survey_variables is a wrapper and will require that the new (target) variable names are present in a valid crosstable.

Value

A list of surveys or save individual rds files on the `export_path`.

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)
  )

subset_surveys(survey_list = example_surveys,
               subset_vars = c("rowid", "isocntry", "qa10_1", "qa14_1"),
               subset_name = "subset_example")
```

---

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 = character(),
  filename = character(),
  doi = character()
)

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.

Examples

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

Index