Package 'psHarmonize'

October 16, 2025

Title Creates a Harmonized Dataset Based on a Set of Instructions

Version 0.3.6

Description Functions which facilitate harmonization of data from multiple different datasets. Data harmonization involves taking data sources with differing values, creating coding instructions to create a harmonized set of values, then making those data modifications. 'psHarmonize' will assist with data modification once the harmonization instructions are written. Coding instructions are written by the user to create a ``harmonization sheet''. This sheet catalogs variable names, domains (e.g. clinical, behavioral, outcomes), provides R code instructions for mapping or conversion of data, specifies the variable name in the harmonized data set, and tracks notes. The package will then harmonize the source datasets according to the harmonization sheet to create a harmonized dataset. Once harmonization is finished, the package also has functions that will create descriptive statistics using 'RMarkdown'. Data Harmonization guidelines have been described by Fortier I, Raina P, Van den Heuvel ER, et al. (2017) <doi:10.1093/ije/dyw075>. Additional details of our R package have been described by Stephen JJ, Carolan P, Krefman AE, et al. (2024) <doi:10.1016/j.patter.2024.101003>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports dplyr, glue, magrittr, purrr, RColorBrewer, rlang, rmarkdown, stringr, tidyr

Suggests testthat (>= 3.0.0), knitr

VignetteBuilder knitr

Depends R (>= 2.10)

Config/testthat/edition 3

URL https://github.com/NUDACC/psHarmonize

BugReports https://github.com/NUDACC/psHarmonize/issues

NeedsCompilation no

Author John Stephen [aut, cre] (ORCID:
<https://orcid.org/0000-0001-7309-9193>),
Maxwell Mansolf [ctb] (ORCID: <https://orcid.org/0000-0001-6861-8657>)

Maintainer John Stephen <John.Stephen@northwestern.edu>

Repository CRAN

Date/Publication 2025-10-16 21:50:02 UTC

Contents

`code_modify_func` *Code modify function. To be called by the harmonization function.*

Description

Code modify function. To be called by the harmonization function.

Usage

```
code_modify_func(
  data = temp_dataset,
  instruction = code_instruct,
  old_var = source_item_long
)
```

Arguments

data	Dataframe to be modified
instruction	Coding instruction from harmonization sheet
old_var	Name of original variable

Value

Vector of length equal to old_var

Examples

```
# Allows the user to define a function with `instruction`
# The `old_var` in `data` will be used in place of `x` in `instruction`.

code_modify_func(data = cohort_a, instruction = 'x + 5', old_var = 'age')
```

---

code_modify_func_multi

Code modify function (multiple variables). To be called by the harmonization function.

Description

Code modify function (multiple variables). To be called by the harmonization function.

Usage

```
code_modify_func_multi(
  data = temp_dataset,
  instruction = code_instruct,
  old_var = source_item_long,
  user_args = source_item,
  sourcedataset = source_dataset,
  subdomain = subdomain,
  visit = visit,
  cohort = cohort
)
```

Arguments

data	Dataframe to be modified
instruction	Coding instruction from harmonization sheet
old_var	Name of original variable
user_args	Character vector of input variables

sourcedataset	Dataframe created so far
subdomain	Category of variable
visit	Visit number
cohort	Cohort name

Value

Vector of length equal to old_var

Examples

```
# Example calculating BMI in cohort a for visit 1

code_modify_func_multi(data = cohort_a,
                       instruction = '(x1 / 2.205)/((x2 / 39.37)**2)',
                       old_var = 'weight_1; height_1',
                       user_args = c('weight_1','height_1'),
                       sourcedataset = 'cohort_a',
                       subdomain = 'clinical',
                       visit = 1,
                       cohort = 'cohort_a')
```

---

code_modify_recode *Code modify recode. To be called by harmonization function.*

Description

Code modify recode. To be called by harmonization function.

Usage

```
code_modify_recode(
  data = temp_dataset,
  instruction = code_instruct,
  old_var = source_item_long,
  new_var = item,
  na_string = NULL
)
```

Arguments

data	Dataframe to be modified
instruction	Coding instruction from harmonization sheet
old_var	Name of original variable
new_var	Name of new variable
na_string	Character string of final recode value to be set to NA.

Value

Returns vector of new variable after recoding as needed.

Examples

```
test_data <- data.frame(val = c('a','b','c','d'))

code_modify_recode(data = test_data,
  instruction = 'a = apple; c = carrot', old_var = 'val', new_var = 'new')
```

---

cohort_a *Cohort A*

Description

Example dataset of cohort data

Usage

```
cohort_a
```

Format

A data frame with 10,000 rows and 9 columns

Source

Simulated data

---

cohort_b *Cohort B*

Description

Example dataset of cohort data

Usage

```
cohort_b
```

Format

A data frame with 5,000 rows and 5 columns

Source

Simulated data

cohort_c *Cohort C*

Description

Example dataset of cohort data

Usage

```
cohort_c
```

Format

A data frame with 7,000 rows and 5 columns

Source

Simulated data

cohort_shell_func *Cohort sheet create. To be called by harmonization function.*

Description

Created dataframe "shell" of IDs, study/cohort name, and visit. Harmonized variables will be joined onto this dataset.

Usage

```
cohort_shell_func(sheet)
```

Arguments

sheet	Harmonization sheet

Value

Data.frame with IDs, study/cohort name, and visit.

Examples

```
# Using example harmonization sheet
cohort_shell_func(harmonization_sheet_example)
```

cont_or_cat *Continuous or categorical*

Description

Continuous or categorical

Usage

```
cont_or_cat(data, var)
```

Arguments

data	Data frame
var	Variable

Value

Returns "continuous" or "categorical"

Examples

```
# Function can help determine which kind of output is
# most appropriate

cont_or_cat(data = cohort_a, var = 'height_1')

cont_or_cat(data = cohort_a, var = 'education')
```

create_error_log_report *Error log report creation*

Description

This function will create an RMarkdown error log. It takes the harmonization object as the input, and will knit an RMarkdown html file to the path specified.

Note: The error log will only be able to detect "processing" errors, and not "content" errors. For example, if the user enters coding instructions that are nonsensical or incorrect, but are still able to be executed, this function will not be able to detect it.

Usage

```
create_error_log_report(
  harmonization_object,
  path = "./",
  file = "error_log_report.html"
)
```

Arguments

harmonization_object	Harmonization object
path	Path of output R Markdown report
file	Filename of output R Markdown report

Value

Does not return an object, but instead knits html RMarkdown report to specified path and file name.

Examples

```
# Examples not run

# Creating harmonized object using harmonization sheet with errors.
# harmonized_obj <- harmonization(harmonization_sheet = error_harmonization_sheet_example)

# Knitting error log report
# create_error_log_report(harmonization_object = harmonized_obj,
#   path = './',
#   file = 'example_output.html)
```

---

create_long_dataset *Create long dataset.*

Description

This function is usually not called by the user. Instead it is usually called by harmonization() function.

Usage

```
create_long_dataset(
  vars_interest,
  subdomain,
  previous_dataset,
  error_log,
  na_string,
  verbose = TRUE
)
```

Arguments

vars_interest	Variable currently being harmonized
subdomain	Category of variable
previous_dataset	Dataframe created so far
error_log	Error log
na_string	Character string of final recode value to be set to NA.
verbose	(TRUE/FALSE) Should the function print the current progress to the console?

Details

The function takes the harmonization sheet, and input dataframe, and creates a dataframe with the harmonized variable.

Value

Returns a list with the harmonized long dataset, and error log.

Examples

```
# Example sheet
test_sheet <- harmonization_sheet_example[harmonization_sheet_example$study == 'Cohort A',]

# Example dataset
test_data <- cohort_a

# create error log
test_error_log <- test_sheet[,c('item','study','visit','possible_range')]

test_error_log$completed_status <- NA_character_
test_error_log$completed_reason <- NA_character_
test_error_log$range_set_to_na <- NA_integer_
test_error_log$range_out_of_range_warning <- NA

long_dataset <- create_long_dataset(vars_interest = test_sheet,
                                    subdomain = 'age',
                                    previous_dataset = test_data,
                                    error_log = test_error_log,
                                    na_string = 'NA',
                                    verbose = TRUE)
```

`create_summary_report` *Summary report creation*

Description

Summary report creation

Usage

```
create_summary_report(
  harmonization_object,
  path = "./",
  file = "summary_report.html",
  compare = FALSE
)
```

Arguments

`harmonization_object`	Harmonization object
`path`	Path of output R Markdown report
`file`	Filename of output R Markdown report
`compare`	Creates summary report with comparison of raw values with modified values

Value

Does not return an object, but instead knits html RMarkdown report to specified path and file name.

Examples

```
# Examples not run

# Creating harmonized object
# harmonized_obj <- harmonization(harmonization_sheet = harmonization_sheet_example)

# Knitting summary report
# create_summary_report(harmonization_object = harmonized_obj,
#   path = './',
#   file = 'example_output.html)

# Use `compare` option to create comparison summary report.
# create_summary_report(harmonization_object = harmonized_obj,
#   path = './',
#   file = 'example_output.html,
#   compare = TRUE)
```

error_harmonization_sheet_example *Error harmonization sheet example*

Description

Example of harmonization sheet. This harmonization sheet has a few typos present (incorrect name of variable and/or dataset). This can be used to demonstrate how errors are presented in the error log and/or the summary method.

Usage

```
error_harmonization_sheet_example
```

Format

A data frame with 16 rows and 12 columns

Source

Created data

harmonization *Harmonization Function*

Description

This is the main function in the psHarmonize package. Takes a harmonization sheet as input, and returns a harmonization object (list with S3 class of 'psHarmonize'). Requires source data.frames to be in the global environment.

Usage

```
harmonization(
  harmonization_sheet,
  long_dataset = TRUE,
  wide_dataset = TRUE,
  error_log = TRUE,
  source_variables = TRUE,
  na_string = "NA",
  verbose = TRUE
)
```

Arguments

harmonization_sheet	Harmonization sheet input. Set of coding instructions
long_dataset	(TRUE/FALSE) Should the function return a long dataset?
wide_dataset	(TRUE/FALSE) Should the function return a wide dataset?
error_log	(TRUE/FALSE) Should the function return an error log?
source_variables	(TRUE/FALSE) Should the output datasets contain the original non modified values?
na_string	Character string of final recode value to be set to missing. Default is 'NA'. For example, if you use code_type of 'recode', and some of your final values are 'NA', they will be set to missing.
verbose	(TRUE/FALSE) Should the harmonization() function print the current progress to the console?

Details

Note: psHarmonize evaluates and runs code entered in the harmonization sheet. Make sure to only use harmonization sheets from authors you trust.

Value

List of return objects with S3 class of 'psHarmonize'. Can be used as input for report function create_summary_report() and create_error_log_report().

Examples

```
# Running harmonization function with example harmonization sheet
harmonization_obj <- harmonization(harmonization_sheet = harmonization_sheet_example)

# Extracting harmonized long dataset (each row is a visit)
long_dataset <- harmonization_obj$long_dataset

# Extracting harmonized wide dataset (each row is a person)
# Visits are expressed in multiple columns
wide_dataset <- harmonization_obj$wide_dataset
```

---

harmonization_sheet_example

Harmonization sheet example

Description

Example of a harmonization sheet. This serves as the input file for the harmonization function.

Usage

```
harmonization_sheet_example
```

Format

A data frame with 16 rows and 12 columns

Source

Created data

---

range_function *Range function. To be called by harmonization function.*

Description

Range function. To be called by harmonization function.

Usage

```
range_function(
  data = temp_dataset,
  min_max_range = possible_range,
  new_var = item
)
```

Arguments

data	Data to be modified
min_max_range	Range of allowed values
new_var	New variable

Value

Returns a list with the new vector (values outside of range set to NA), and the number of values set to NA.

Examples

```
test_data <- data.frame(val = 1:10)

range_function(data = test_data, min_max_range = '[2,8]', new_var = 'val')
```

range_function_cat *Possible values for categorical variables. To be called by harmonization function.*

Description

Possible values for categorical variables. To be called by harmonization function.

Usage

```
range_function_cat(
  data = temp_dataset,
  possible_vals_cat = possible_vals,
  new_var = item
)
```

Arguments

data	data to be modified
possible_vals_cat	vector of possible values
new_var	new variable

Value

Returns a list with the new vector (values outside of set to NA), and the number of values set to NA.

Examples

```
test_data <- data.frame(val = c('a','b','j','k','c','d'))

range_function_cat(data = test_data, possible_vals_cat = c('a','b','c','d'), new_var = 'val')
```

reorder_factors *Reorder factors*

Description

Reorder factors

Usage

```
reorder_factors(data, sheet)
```

Arguments

data	Harmonization object, or harmonized data.frame.
sheet	Factor reorder sheet.

Value

Returns harmonization object, or harmonized data.frame.

Examples

```
# Running harmonization function with example harmonization sheet
harmonization_obj <- harmonization(harmonization_sheet = harmonization_sheet_example)

long_dataset <- harmonization_obj$long_dataset

table(long_dataset$education)

# College
# 5643
#
# Graduate/Professional
# 1287
#
# High school
# 7562
#
# No education/grade school
# 7508

# Creating factor reorder sheet
edu_order <- data.frame(
 variable = 'education',
 values = c('No education/grade school', 'High school', 'College', 'Graduate/Professional'),
 order = 1:4
)

# Reorder factors
harmonization_obj <- reorder_factors(data = harmonization_obj, sheet = edu_order)

long_dataset <- harmonization_obj$long_dataset

table(long_dataset$education)

# No education/grade school
# 7508
#
# High school
# 7562
#
# College
# 5643
#
```

```
# Graduate/Professional
# 1287
```

`reorder_factors_df` *Reorder factors data.frame*

Description

Reorder factors data.frame

Usage

```
reorder_factors_df(data, sheet)
```

Arguments

`data`	Harmonized data.frame
`sheet`	Factor reorder sheet

Value

Returns harmonized data.frame.

Examples

```
# Creating example dataframe of variables, the order, and the values
# The function will reorder the factor using these values in the order
# provided.

# This would typically be created in an excel or CSV file outside of R,
# and then imported into R.
test_sheet <- data.frame(
  variable = c(rep('Education',4),rep('Class',3)),
  order = c(1,2,3,4,1,2,3),
  values = c('None','Grade','HS','College','A','B','C')
)

# I'm creating some test data to demonstrate
set.seed(1234)
test_data <- data.frame(
  ID = 1:20,
  Education = sample(c('None','Grade','HS','College'), size = 20, replace = TRUE),
  Class = sample(c('A','B','C'), size = 20, replace = TRUE)
)

# Creating factors in the test data
test_data$Education <- factor(test_data$Education)
```

```
test_data$Class <- factor(test_data$Class)

table(test_data$Education, useNA = 'ifany')
table(test_data$Class, useNA = 'ifany')


# Now reordering factors based on the sheet
test_data_mod <- reorder_factors_df(data = test_data, sheet = test_sheet)

table(test_data_mod$Education, useNA = 'ifany')
table(test_data_mod$Class, useNA = 'ifany')
```

`summary.psHarmonize` *psHarmonize summary method*

Description

psHarmonize summary method

Usage

```
## S3 method for class 'psHarmonize'
summary(object, ..., verbose = FALSE)
```

Arguments

`object`	psHarmonize object
`...`	Can provide additional arguments
`verbose`	T/F. When TRUE, will list variables for each section.

Value

Doesn't return object. Prints status of harmonization (# of harmonizated variables, etc.)

Examples

```
harmonization_obj <- harmonization(harmonization_sheet_example)

summary(harmonization_obj)

# Use verbose option to see more details
summary(harmonization_obj, verbose = TRUE)
```

Index