Package ‘tidyREDCap’

September 16, 2024

Title Helper Functions for Working with 'REDCap' Data

Version 1.1.2

Description Helper functions for processing 'REDCap' data in R. 'REDCap' is a web-enabled application for building and managing surveys and databases developed at Vanderbilt University.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.5.0)

Imports cli, dplyr, janitor, labelVector, magrittr, purrr, REDCapR, rlang, stringr, tibble, tidyr, tidyselect, vctrs

Suggests knitr, redcapAPI, rmarkdown, skimr, testthat (>= 3.0.0)

VignetteBuilder knitr

URL `https://raymondbalise.github.io/tidyREDCap/index.html`

BugReports `https://github.com/RaymondBalise/tidyREDCap/issues`

Config/testthat/edition 3

NeedsCompilation no

Author Raymond Balise [aut, cre] (<`https://orcid.org/0000-0002-9856-5901`>),
Gabriel Odom [aut] (<`https://orcid.org/0000-0003-1341-4555`>),
Anna Calderon [aut] (<`https://orcid.org/0000-0002-0139-3841`>),
Layla Bouzoubaa [aut] (<`https://orcid.org/0000-0002-6616-0950`>),
Wayne DeFreitas [aut] (<`https://orcid.org/0000-0002-2584-6278`>),
Lauren Nahodyl [ctb] (<`https://orcid.org/0000-0001-6241-2615`>),
Kyle Grealis [aut] (<`https://orcid.org/0000-0002-9223-8854`>)

Maintainer Raymond Balise <`balise@miami.edu`>

Repository CRAN

Date/Publication 2024-09-16 19:00:02 UTC

Contents

`drop_label` *Drop the label from a variable*

Description

There is a reported issues with joins on data (without a reprex) that seem to be caused by the labels. As a possible solution this can be used to drop labels.

Usage

```
drop_label(df, x)
```

Arguments

`df`	the name of the data frame
`x`	the quoted name of the variable

Value

df

drop_labels *Drop all the labels from a variable*

Description

There is an issue with the function we are using to add column labels. If you run into problems processing the labels.

Usage

```
drop_labels(df)
```

Arguments

df	The data frame with column labels that you want to drop

Value

df without column labels

Examples

```
## Not run:
demographics |>
  drop_labels() |>
  skimr::skim()

## End(Not run)
```

import_instruments *Import all instruments into individual R tables*

Description

This function takes the url and key for a REDCap project and returns a table for each instrument/form in the project.

Usage

```
import_instruments(
  url,
  token,
  drop_blank = TRUE,
  record_id = "record_id",
  first_record_id = 1,
  envir = .GlobalEnv
)
```

Arguments

url The API URL for your the instance of REDCap

token The API security token

drop_blank Drop records that have no data. TRUE by default.

record_id Name of record_id variable (if it was changed in REDCap).

first_record_id
A value of the custom record_id variable (if changed in REDCap). To improve the speed of the import, tidyREDCap pulls in a single record twice. By default if uses the first record. If you have a custom record_id variable and if its the first record identifier is not 1, specify a record identifier value here. For example if you are using dude_id instead of record_id and dude_id has a value of "first dude" for one of its records this argument would be first_record_id = "first dude".

envir The name of the environment where the tables should be saved.

Value

one data.frame for each instrument/form in a REDCap project. By default the datasets are saved into the global environment.

Examples

```
## Not run:
import_instruments(
  "https://redcap.miami.edu/api/",
  Sys.getenv("test_API_key")
)

## End(Not run)
```

make_binary_word *Convert a "choose all that apply" Question Into a Binary Word*

Description

This function takes a data frame holding binary variables with values corresponding to a dummy-coded "choose all that apply" question. It can be used for any *binary word* problem.

Usage

```
make_binary_word(df, yes_value = "Checked", the_labels = letters)
```

Arguments

`df`	A data frame with the variables corresponding to binary indicators (the dummy coded variables) for a "choose all that apply" question.
`yes_value`	A character string that corresponds to choosing "yes" in the binary variables of `df`. Defaults to the REDCap "Checked" option.
`the_labels`	A character vector of single letters holding the letters used to make the binary word. See the article/vignette called "Make Binary Word" for an example: `https://raymondbalise.github.io/tidyREDCap/articles/makeBinaryWord.html`.

Value

A character vector with length equal to the rows of `df`, including one letter or underscore for each column of `df`. For instance, if `df` has one column for each of the eight options of the Nacho Craving Index example instrument (`https://libguides.du.edu/c.php?g=948419&p=6839916`), with a row containing the values "Chips" (checked), "Yellow cheese" (unchecked), "Orange cheese" (checked), "White cheese" (checked), "Meat" (checked), "Beans" (unchecked), "Tomatoes" (unchecked) and "Peppers" (checked), then the character string corresponding to that row will be `"a_cde__h"`. The underscores represent that the options for "Yellow cheese", "Beans", and "Tomatoes" were left unchecked.

Examples

```
test_df <- tibble::tibble(
  q1 = c("Unchecked", "Checked"),
  q2 = c("Unchecked", "Unchecked"),
  q3 = c("Checked", "Checked"),
  q4 = c("Checked", "Unchecked")
)
make_binary_word(test_df)
```

`make_choose_all_table` *Count The Responses to a Choose All That Apply Question*

Description

This will tally the number of responses on a choose all that apply question. This function extracts the option name from the variable labels. So the data set needs to be labeled. See the Make a 'Choose All' Table vignette for help.

Usage

```
make_choose_all_table(df, variable)
```

Arguments

`df`	The name of the data set (it needs labels)
`variable`	The name of the REDCap variable

Value

A variable's response label without the choose all the question

make_choose_one_table *Make a frequency table for a categorical variable*

Description

Pass this function either 1) a labeled factor or 2) a data frame and also a factor in the frame, and it will return a `janitor`-style table. Use subset = TRUE if you are making a report on a variable that is part of a *choose all that apply* question.

Usage

```
make_choose_one_table(arg1, arg2, subset = FALSE)
```

Arguments

`arg1`	data frame that has a factor or a factor name
`arg2`	if arg1 is a data frame, this is a factor name
`subset`	can be equal to TRUE/FALSE. This option removes extra variable name text from the label. This option is useful for *choose all that apply* questions.

Value

a table

make_instrument *Extract an Instrument from an REDCap Export*

Description

This function takes a data frame and the names of the first and last variables in an instrumnt and returns a data frame with the instrument.

Usage

```
make_instrument(
  df,
  first_var,
  last_var,
  drop_which_when = FALSE,
  record_id = "record_id"
)
```

Arguments

`df` A data frame with the instrument

`first_var` The name of the first variable in an instrument

`last_var` The name of the last variable in an instrument

`drop_which_when` Drop the `record_id` and `redcap_event_name` variables

`record_id` Name of `record_id` variable (if it was changed in REDCap)

Value

A data frame that has an instrument (with at least one not NA value)

`make_instrument_auto` *Extract an Instrument from an REDCap Export without specifying Variables*

Description

This function takes a data frame holding REDCap data, checks if it is a longitudinal study, and returns records that have values.

Usage

```
make_instrument_auto(df, drop_which_when = FALSE, record_id = "record_id")
```

Arguments

`df` A data frame with the instrument

`drop_which_when` Drop the `record_id` and `redcap_event_name` variables

`record_id` Name of `record_id` variable (if it was changed in REDCap)

Value

A data frame that has an instrument (with at least one not NA value).

make_yes_no *make_yes_no*

Description

Convert a "Yes-No", "True-False" or "Checkboxes (Multiple Answers)" question in REDCap to a factor holding "Yes" or "No or Unknown". Technically "yes" or "checked" (ignoring case), 1 or TRUE responses are converted to "Yes" and all other values to "No or Unknown". Also see `make_yes_no_unknown()`.

Usage

```
make_yes_no(x)
```

Arguments

x	x variable to be converted to hold "Yes" or "No or Unknown"

Value

a factor with "Yes" or "No or Unknown"

Examples

```
make_yes_no(c(0, 1, NA))
make_yes_no(c("unchecked", "Checked", NA))
```

make_yes_no_unknown *make_yes_no_unknown*

Description

Convert a "Yes-No", "True-False" or "Checkboxes (Multiple Answers)" question in REDCap to a factor holding "No" or "Yes" or "Unknown". Technically "yes" or "checked" (ignoring case), 1 or TRUE responses are converted to "Yes". "No" or "unchecked" (ignoring case), 0 or FALSE are converted to "No". All other values are set to "Unknown". Also see `make_yes_no()`.

Usage

```
make_yes_no_unknown(x)
```

Arguments

x	variable to be converted to hold "No", "Yes", or Unknown"

Value

a factor with "No", "Yes", or Unknown"

Examples

```
make_yes_no_unknown(c(0, 1, NA))
make_yes_no_unknown(c("unchecked", "Checked", NA))
```

Index