Package 'REDCapTidieR'

December 7, 2022

Type Package

Title Extract 'REDCap' Databases into Tidy 'Tibble's

Version 0.2.0

Description Convert 'REDCap' exports into tidy tables for easy handling of 'REDCap' repeat instruments and event arms.

License MIT + file LICENSE

URL https://github.com/CHOP-CGTInformatics/REDCapTidieR, https://chop-cgtinformatics.github.io/REDCapTidieR/

BugReports https://github.com/CHOP-CGTInformatics/REDCapTidieR/issues

Depends R (>= 3.5.0)

Imports checkmate, cli, dplyr, lobstr, lifecycle, purrr, REDCapR (>= 1.1.0), rlang, stringi, stringr, tibble, tidyr, tidyselect, formattable

Suggests covr, httptest, knitr, labelled, readr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.2

NeedsCompilation no

Author Richard Hanna [aut, cre], Stephan Kadauke [aut], Ezra Porter [aut]

Maintainer Richard Hanna <richardshanna91@gmail.com>

Repository CRAN

Date/Publication 2022-12-07 22:22:28 UTC

R topics documented:

bind_tibbles 2
extract_tibble 3
extract_tibbles 4
format-helpers 5
make_labelled 6
read_redcap 7

Index **9**

`bind_tibbles` *Extract data tibbles from a REDCapTidieR supertibble and bind them to an environment*

Description

Take a supertibble generated with `read_redcap()` and bind its data tibbles (i.e. the tibbles in the `redcap_data` column) to an environment. The default is the global environment.

Usage

```
bind_tibbles(supertbl, environment = global_env(), tbls = NULL)
```

Arguments

`supertbl`	A supertibble generated by `read_redcap()`. Required.
`environment`	The environment to bind the tibbles to. Default is `rlang::global_env()`.
`tbls`	A vector of the `redcap_form_names` of the data tibbles to bind to the environment. Default is `NULL` which binds all data tibbles.

Value

This function returns nothing as it's used solely for its side effect of modifying an environment.

Examples

```
# Create an empty environment
my_env <- new.env()

ls(my_env)

# Mock up a supertibble
supertbl <- tibble::tribble(
  ~redcap_form_name,    ~redcap_data,   ~structure,
  "super_hero_powers",  list(),         "repeating",
  "heroes_information", list(),         "nonrepeating"
)
```

```
bind_tibbles(supertbl, my_env)

ls(my_env)
```

`extract_tibble` *Extract a single data tibble from a REDCapTidieR supertibble*

Description

Take a supertibble generated with `read_redcap()` and return one of its data tibbles.

Usage

```
extract_tibble(supertbl, tbl)
```

Arguments

`supertbl`	A supertibble generated by `read_redcap()`. Required.
`tbl`	The `redcap_form_name` of the data tibble to extract. Required.

Details

This function makes it easy to extract a single instrument's data from a REDCapTidieR supertibble.

Value

A `tibble`.

Examples

```
# Mock up a supertibble
sample_data <- tibble::tribble(
  ~redcap_form_name,    ~redcap_data,   ~structure,
  "super_hero_powers",  list(),         "repeating",
  "heroes_information", list(),         "nonrepeating"
)

extract_tibble(sample_data, "heroes_information")
```

`extract_tibbles` *Extract data tibbles from a REDCapTidieR supertibble into a list*

Description

Take a supertibble generated with `read_redcap()` and return a named list of data tibbles.

Usage

```
extract_tibbles(supertbl, tbls = everything())
```

Arguments

`supertbl`	A supertibble generated by `read_redcap()`. Required.
`tbls`	A vector of `form_names` or a tidyselect helper. Default is `dplyr::everything()`.

Details

This function makes it easy to extract a multiple instrument's data from a REDCapTidieR supertibble into a named list. Specifying instruments using tidyselect helper functions such as `dplyr::starts_with()` or `dplyr::ends_with()` is supported.

Value

A named list of `tibbles`

Examples

```
# Mock up a supertibble
sample_data <- tibble::tribble(
  ~redcap_form_name,    ~redcap_data, ~structure,
  "super_hero_powers",  list(),       "repeating",
  "heroes_information", list(),       "nonrepeating"
)

# Extract all data tibbles
extract_tibbles(sample_data)

# Only extract data tibbles starting with "heroes"
extract_tibbles(sample_data, starts_with("heroes"))
```

format-helpers *Format REDCap variable labels*

Description

Use these functions with the `format_labels` argument of `make_labelled()` to define how variable labels should be formatted before being applied to the data columns of `redcap_data`. These functions are helpful to create pretty variable labels from REDCap field labels.

- `fmt_strip_whitespace()` removes extra white space inside and at the start and end of a string. It is a thin wrapper of `stringr::str_trim()` and `stringr::str_squish()`.
- `fmt_strip_trailing_colon()` removes a colon character at the end of a string.
- `fmt_strip_trailing_punct()` removes punctuation at the end of a string.
- `fmt_strip_html()` removes html tags from a string.
- `fmt_strip_field_embedding()` removes text between curly braces `{}` which REDCap uses for special "field embedding" logic. Note that `read_redcap()` removes html tags and field embedding logic from field labels in the metadata by default.

Usage

```
fmt_strip_whitespace(x)

fmt_strip_trailing_colon(x)

fmt_strip_trailing_punct(x)

fmt_strip_html(x)

fmt_strip_field_embedding(x)
```

Arguments

x	a character vector

Value

a modified character vector

Examples

```
fmt_strip_whitespace("Poorly Spaced   Label ")

fmt_strip_trailing_colon("Label:")

fmt_strip_trailing_punct("Label-")
```

```
fmt_strip_html("<b>Bold Label</b>")

fmt_strip_field_embedding("Label{another_field}")

supertbl <- tibble::tribble(
  ~redcap_data, ~redcap_metadata,
 tibble::tibble(x = letters[1:3]), tibble::tibble(field_name = "x", field_label = "X Label:")
)

make_labelled(supertbl, format_labels = fmt_strip_trailing_colon)
```

`make_labelled` *Apply variable labels to a REDCapTidieR supertibble*

Description

Take a supertibble and use the `labelled` package to apply variable labels to the columns of the supertibble as well as to each tibble in the `redcap_data`, `redcap_metadata`, and `redcap_events` columns of that supertibble.

Usage

```
make_labelled(supertbl, format_labels = NULL)
```

Arguments

`supertbl`	a supertibble generated using `read_redcap()`
`format_labels`	one or multiple optional label formatting functions. A label formatting function is a function that takes a character vector and returns a modified character vector of the same length. This function is applied to field labels before attaching them to variables. One of: • `NULL` to apply no additional formatting. Default. • A label formatting function. • A character with the name of a label formatting function. • A vector or list of label formatting functions or function names to be applied in order. Note that ordering may affect results.

Details

The variable labels for the data tibbles are derived from the `field_label` column of the metadata tibble.

Value

A labelled supertibble.

Examples

```
supertbl <- tibble::tribble(
  ~redcap_data, ~redcap_metadata,
 tibble::tibble(x = letters[1:3]), tibble::tibble(field_name = "x", field_label = "X Label"),
 tibble::tibble(y = letters[1:3]), tibble::tibble(field_name = "y", field_label = "Y Label")
)

make_labelled(supertbl)

make_labelled(supertbl, format_labels = tolower)

## Not run:
redcap_uri <- Sys.getenv("REDCAP_URI")
token <- Sys.getenv("REDCAP_TOKEN")

supertbl <- read_redcap(redcap_uri, token)
make_labelled(supertbl)

## End(Not run)
```

`read_redcap` *Import a REDCap database into a tidy supertibble*

Description

Query the REDCap API to retrieve data and metadata about a project, and transform the output into a "supertibble" that contains data and metadata organized into tibbles, broken down by instrument.

Usage

```
read_redcap(
  redcap_uri,
  token,
  raw_or_label = "label",
  forms = NULL,
  export_survey_fields = TRUE,
  suppress_redcapr_messages = TRUE
)
```

Arguments

`redcap_uri`	The URI/URL of the REDCap server (e.g., "https://server.org/apps/redcap/api/"). Required.
`token`	The user-specific string that serves as the password for a project. Required.
`raw_or_label`	A string (either 'raw' or 'label') that specifies whether to export the raw coded values or the labels for the options of categorical fields. Default is 'label'.
`forms`	A character vector of REDCap instrument names that specifies which instruments to import. Default is `NULL` which imports all instruments in the project.

`export_survey_fields`
: A logical that specifies whether to export the survey identifier and timestamp fields if available. Default is `TRUE`.

`suppress_redcapr_messages`
: A logical to control whether to suppress messages from REDCapR API calls. Default `TRUE`.

Details

This function uses the REDCapR package to query the REDCap API. The REDCap API returns a block matrix that mashes data from all data collection instruments together. The `read_redcap()` function deconstructs the block matrix and splices the data into individual tibbles, where one tibble represents the data from one instrument.

Value

A `tibble` in which each row represents a REDCap instrument. It contains the following columns:

- `redcap_form_name`, the name of the instrument
- `redcap_form_label`, the label for the instrument
- `redcap_data`, a tibble with the data for the instrument
- `redcap_metadata`, a tibble of data dictionary entries for each field in the instrument
- `redcap_events`, a tibble with information about the arms and longitudinal events represented in the instrument. Only if the project has longitudinal events enabled
- `structure`, the instrument structure, either "repeating" or "nonrepeating"
- `data_rows`, the number of rows in the instrument's data tibble
- `data_cols`, the number of columns in the instrument's data tibble
- `data_size`, the size in memory of the instrument's data tibble computed by `lobstr::obj_size()`
- `data_na_pct`, the percentage of cells in the instrument's data columns that are `NA` excluding identifier and form completion columns

Examples

```
## Not run:
redcap_uri <- Sys.getenv("REDCAP_URI")
token <- Sys.getenv("REDCAP_TOKEN")

read_redcap(
  redcap_uri,
  token,
  raw_or_label = "label"
)

## End(Not run)
```

Index

bind_tibbles, 2

extract_tibble, 3
extract_tibbles, 4

fmt_strip_field_embedding
        *(*format-helpers*)*, 5
fmt_strip_html *(*format-helpers*)*, 5
fmt_strip_trailing_colon
        *(*format-helpers*)*, 5
fmt_strip_trailing_punct
        *(*format-helpers*)*, 5
fmt_strip_whitespace *(*format-helpers*)*, 5
format-helpers, 5

make_labelled, 6

read_redcap, 7