

Package: REDCapDM (via r-universe)

October 12, 2024

Type Package

Title 'REDCap' Data Management

Version 0.9.9

Maintainer João Carmezim <jcarmezim@igtp.cat>

Description REDCap Data Management - REDCapDM is an R package that allows users to manage data exported directly from REDCap or using an API connection. This package includes several functions designed for pre-processing data, generating reports of queries such as outliers or missing values, and following up on the identified queries. 'REDCap' (Research Electronic Data CAPture; <<https://projectredcap.org>>) is a web application developed at Vanderbilt University, designed for creating and managing online surveys and databases and the REDCap API is an interface that allows external applications to connect to REDCap remotely, and is used to programmatically retrieve or modify project data or settings within REDCap, such as importing or exporting data.

License MIT + file LICENSE

URL <https://bruigtp.github.io/REDCapDM/>,
<https://doi.org/10.1186/s12874-024-02178-6>

BugReports <https://github.com/bruigtp/REDCapDM/issues>

Encoding UTF-8

RoxygenNote 7.3.2

Imports dplyr, janitor, magrittr, openxlsx, purrr, REDCapR, rlang, stringr, tibble, tidyr, tidyselect, labelled, utils, stringi, cli, forcats

Suggests knitr, rmarkdown, kableExtra

VignetteBuilder knitr

Depends R (>= 3.6)

LazyData true

Repository <https://bruigtp.r-universe.dev>
RemoteUrl <https://github.com/bruigtp/redcapdm>
RemoteRef HEAD
RemoteSha 154a7785650db6b6c3376c91c85fab046ac3f1ce

Contents

checkbox_names	2
check_queries	3
covican	4
fill_data	5
rd_event	6
rd_export	7
rd_insert_na	8
rd_query	9
rd_rlogic	11
rd_transform	12
recalculate	13
redcap_data	14
split_event	15
split_form	16
to_factor	17
transform_checkboxes	17

Index 18

checkbox_names	<i>Change Checkboxes Names to Option Names</i>
----------------	--

Description

This function updates the names of checkboxes in the dataset and dictionary to reflect the names of their options.

Usage

```
checkbox_names(data, dic, labels, checkbox_labels = c("No", "Yes"))
```

Arguments

data	Dataset containing the REDCap data.
dic	Dataset containing the REDCap dictionary.
labels	Named character vector with the names of the variables in the data and their corresponding REDCap labels.
checkbox_labels	Character vector specifying the names for the two options of each checkbox variable. The default is 'c('No', 'Yes')'.

check_queries	<i>Check for Changes Between Two Query Reports</i>
---------------	--

Description

This function compares an old report of queries with a new one. It allows you to identify which queries are new, which have been modified, and which remain unchanged.

Usage

```
check_queries(old, new, report_title = NULL)
```

Arguments

old	Previous version of the queries report.
new	New version of the queries report. This object is used to determine the status of each query.
report_title	Character string specifying the title of the report.

Value

A list consisting of a dataframe containing each individual query from both reports and a column showing the status of the queries (new, solved, miscorrected or pending) compared to the previous query report. In addition to this dataframe, there is also a summary of the total number of queries per category.

Examples

```
# Example of a query
data_old <- rd_query(covican,
                    variables = "copd",
                    expression = "is.na(x)",
                    event = "baseline_visit_arm_1")
data_new <- rbind(data_old$queries[1:5,], c("100-20", rep("abc", 8)))

# Control of queries
check <- check_queries(old = data_old$queries,
                      new = data_new)
```

 covican

Subset of COVICAN's Database

Description

A random sample of the COVICAN study. An international, multicentre cohort study of cancer patients with COVID-19 to describe the epidemiology, risk factors, and clinical outcomes of co-infections and superinfections in onco-haematological patients with COVID-19.

Usage

```
data(covican)
```

Format

A data frame with 342 rows and 56 columns

record_id: Identifier of each record. This information does not match the real data.

redcap_event_name: Auto-generated name of the events

redcap_data_access_group: Auto-generated name of each center. This information does not match the real data.

inc_1: Inclusion criteria of 'Patients older than 18 years' (0 = No ; 1 = Yes)

inc_2: Inclusion criteria of 'Cancer patients' (0 = No ; 1 = Yes)

inc_3: Inclusion criteria of 'Diagnosed of COVID-19' (0 = No ; 1 = Yes)

exc_1: Exclusion criteria of 'Solid tumour remission >1 year' (0 = No ; 1 = Yes)

screening_fail_crit: Indicator of non-compliance with inclusion and exclusion criteria (0 = compliance ; 1 = non-compliance)

d_birth: Date of birth (y-m-d). This date does not correspond to the original.

d_admission: Date of first visit (y-m-d). This date does not correspond to the original.

age: Age in years

dm: Indicator of diabetes (0 = No ; 1 = Yes)

type_dm: Type of diabetes (1 = No complications ; 2 = End-organ diabetes-related disease)

copd: Indicator of chronic obstructive pulmonary disease (0 = No ; 1 = Yes)

fio2: Fraction of inspired oxygen in percentage

available_analytics: Indicator of blood test available (0 = No ; 1 = Yes)

potassium: Potassium in mmol/L

resp_rate: Respiratory rate in bpm

leuk_lymph: Indicator of leukemia or lymphoma (0 = No ; 1 = Yes)

acute_leuk: Indicator of acute leukemia (0 = No ; 1 = Yes)

type_underlying_disease[...]: Checkbox with the type of underlying disease (0 = Haematological cancer ; 1 = Solid tumour)

underlying_disease_hemato[...] Checkbox with the type of underlying disease (1 = Acute myeloid leukemia ; 2 = Myelodysplastic syndrome ; 3 = Chronic myeloid leukaemia ; 4 = Acute lymphoblastic leukaemia ; 5 = Hodgkin lymphoma ; 6 = Non Hodgkin lymphoma ; 7 = Multiple myeloma ; 8 = Myelofibrosis ; 9 = Aplastic anaemia ; 10 = Chronic lymphocytic leukaemia ; 11 = Amyloidosis ; 12 = Other)

urine_culture: Indicator of urine culture: (0 = Not done ; 1 = Done)

[... .factor:] Labels of the different variables

Note

List with three data frames: the first one with the data, the second one with the dictionary ('code-book') of the REDCap project and the last one with the instrument-event mappings of the REDCap project.

References

Gudiol, C., Durà-Miralles, X., Aguilar-Company, J., Hernández-Jiménez, P., Martínez-Cutillas, M., Fernandez-Avilés, F., Machado, M., Vázquez, L., Martín-Dávila, P., de Castro, N., Abdala, E., Sorli, L., Andermann, T. M., Márquez-Gómez, I., Morales, H., Gabilán, F., Ayaz, C. M., Kayaaslan, B., Aguilar-Guisado, M., Herrera, F. Royo-Cebrecos C, Peghin M, González-Rico C, Goikoetxea J, Salgueira S, Silva-Pinto A, Gutiérrez-Gutiérrez B, Cuellar S, Haidar G, Maluquer C, Marin M, Pallarès N, Carratalà J. (2021). Co-infections and superinfections complicating COVID-19 in cancer patients: A multicentre, international study. *The Journal of infection*, 83(3), 306–313. <https://doi.org/10.1016/j.jinf.2021.07.014>

fill_data

Fill Rows with Values from One Event

Description

This function fills all rows in the dataset with the value of a particular variable in a specified event. It is an auxiliary function used in the 'rd_rlogic' function.

Usage

```
fill_data(which_event, which_var, data)
```

Arguments

which_event	String specifying the name of the event.
which_var	String specifying the name of the variable.
data	Dataset containing the REDCap data.

rd_event	<i>Identification of Missing Event(s)</i>
----------	---

Description

When working with a longitudinal REDCap project, the exported data has a structure where each row represents one event per record. However, by default REDCap does not export events for which there is no information available. This function allows you to identify which records do not contain information about a particular event.

Usage

```
rd_event(
  ...,
  data = NULL,
  dic = NULL,
  event,
  filter = NA,
  query_name = NA,
  addTo = NA,
  report_title = NA,
  report_zeros = FALSE,
  link = list()
)
```

Arguments

...	List containing the data, dictionary and event mapping (if required) of the REDCap project. This should be the output of the 'redcap_data' function.
data	Data frame containing the data read from REDCap. If the list is specified, this argument is not required.
dic	Data frame containing the dictionary read from REDCap. If the list is specified, this argument is not required.
event	Character vector with the name of the REDCap event(s) to be analyzed.
filter	A filter to be applied to the dataset. This argument can be used to identify the missing events on a subset of the dataset.
query_name	Description of the query. It can be the same for all variables, or you can define a different one for each variable. By default, the function defines it as 'The event [event] is missing' for each event'.
addTo	Data frame corresponding to a previous query data frame to which you can add the new query data frame. By default, the function always generates a new data frame without taking into account previous reports.
report_title	Character string specifying the title of the report.
report_zeros	Logical. If 'TRUE', the function returns a report containing variables with zero queries.
link	List of project information used to create a web link for each missing event.

Value

A list with a data frame of 9 columns (10 columns if the link argument is specified) to help the user identify each missing event and a table with the total number of missing events per event analyzed.

Examples

```
example <- rd_event(covican,
                    event = "follow_up_visit_da_arm_1")
example
```

rd_export

*Exporting Query Dataset***Description**

This function exports a query report, generated using the ‘rd_query’ or ‘rd_event’ functions, to an .xlsx file.

Usage

```
rd_export(
  ...,
  queries = NULL,
  column = NULL,
  sheet_name = NULL,
  path = NULL,
  password = NULL
)
```

Arguments

...	List containing the data frame of queries. This list must be the output of the ‘rd_query’ or ‘rd_event’ functions.
queries	Data frame containing the identified queries. If the list is specified, this argument is not required.
column	Character element specifying the column containing the link for each query.
sheet_name	Character element specifying the sheet name of the resulting xlsx file.
path	Character element specifying the file path to save the xlsx file. If ‘NULL’, the file will be created in the current working directory.
password	String with the password to protect the worksheet and prevent others from making changes.

Value

An .xlsx file containing all the queries and, if available, hyperlinks to each of them.

 rd_insert_na

Insert Missing Values Using a Filter

Description

This function allows you to manually insert a missing value into certain variables ('vars') if the specified filter/s ('filter') are satisfied. It's particularly useful for checkboxes without a gatekeeper question in the branching logic. Note that the variable is only transformed in the events where both the variable and the filter evaluation are present, so they must have at least one event in common.

Usage

```
rd_insert_na(..., data = NULL, dic = NULL, event_form = NULL, vars, filter)
```

Arguments

...	List containing the data, the dictionary and the event if it's needed. Should be the output of the function 'redcap_data'.
data	Data frame containing data from REDCap. If the list is specified, this argument is not needed.
dic	Data frame containing the dictionary read from REDCap. If the list is specified, this argument is not needed.
event_form	Data frame containing the correspondence of each event with each form. If the list is specified, this argument is not needed.
vars	Character vector containing the names of the variables to be transformed.
filter	Character vector containing the logic to be evaluated directly. If each logic is TRUE, the corresponding variable in 'vars' is set to missing.

Value

Transformed data with the specified variables converted.

Examples

```
table(is.na(covican$data$potassium))
data <- rd_insert_na(covican,
  vars = "potassium",
  filter = "age < 65")
table(data$potassium)
```

rd_query	<i>Identification of Queries</i>
----------	----------------------------------

Description

This function allows you to identify queries using a particular expression/filter. It can be used to identify missing values or to identify values outside the lower and upper limits of a variable.

Usage

```
rd_query(
  ...,
  variables = NA,
  expression = NA,
  negate = FALSE,
  event = NA,
  filter = NA,
  addTo = NA,
  variables_names = NA,
  query_name = NA,
  instrument = NA,
  report_title = NA,
  report_zeros = FALSE,
  by_dag = FALSE,
  link = list(),
  data = NULL,
  dic = NULL,
  event_form = NULL
)
```

Arguments

...	List containing the data, dictionary and event mapping (if required) of the REDCap project. This should be the output of the 'redcap_data' function.
variables	Character vector containing the names of the database variables to be checked.
expression	Character vector of expressions to apply to the selected variables.
negate	Logical value indicating whether the defined expression should be negated. Default value is 'FALSE'.
event	The name of the REDCap event to analyze. If there are events in your REDCap project, you should use this argument to name the event to which the defined variables belong.
filter	A filter to be applied to the dataset. For example, this argument can be used to apply the branching logic of a defined variable.
addTo	Data frame corresponding to a previous query data frame to which you can add the new query data frame. By default, this function always creates a new data frame regardless previous reports.

variables_names	Character vector containing the description of each selected variable. By default, the function automatically takes the description of each variable from of the REDCap project dictionary.
query_name	Description of the query. It can be the same for all variables, or you can define a different one for each variable. By default, this function defines it as ‘The value is [value] and it should not be [expression]’.
instrument	REDCap instrument to which the variables belong. It can be the same for all variables, or you can define a different one for each variable. By default, the function automatically selects the corresponding instrument of each variable from the REDCap project dictionary.
report_title	Character string specifying the title of the report.
report_zeros	Logical. If ‘TRUE’, the function returns a report containing variables with zero queries.
by_dag	Logical. If ‘TRUE’, both elements of the output will be grouped by the Data Access Groups (DAGs) of the REDCap project.
link	List containing project information used to create a web link to each query.
data	Data frame containing the data read from REDCap. If the list is given, this argument is not required.
dic	Data frame containing the dictionary read from REDCap. If the list is given, this argument is not required.
event_form	Data frame containing the correspondence of each event with each form. If the list is specified, this argument is not required.

Value

A list with a data frame of 9 columns (10 columns, if the link argument is specified) meant to help the user identify each query and a table with the total number of queries per variable.

Examples

```
# Missing values
example <- rd_query(covican,
  variables = c("copd", "age"),
  expression = c("is.na(x)", "x %in% NA"),
  event = "baseline_visit_arm_1")
example

# Expression
example <- rd_query(covican,
  variables="age",
  expression="x>20",
  event="baseline_visit_arm_1")
example

# Using the filter argument
example <- rd_query(covican,
  variables = "potassium",
```

```

expression = "is.na(x)",
event = "baseline_visit_arm_1",
filter = "available_analytics=='1'"
example

```

rd_rlogic

Translate REDCap Logic to R Logic

Description

This function allows you to convert REDCap logic into R logic. **WARNING:** Please note that if the REDCap logic involves smart variables, this function may not be able to transform it accurately.

Usage

```
rd_rlogic(..., data = NULL, dic = NULL, event_form = NULL, logic, var)
```

Arguments

...	List containing the data, dictionary and event mapping (if applicable) of the REDCap project. This should be the output of the 'redcap_data' function.
data	Data frame containing data from REDCap. If the list is specified, this argument is not required.
dic	Data frame containing the dictionary read from REDCap. If the list is specified, this argument is not required.
event_form	Data frame containing the correspondence of each event with each form. If the list is specified, this argument is not required.
logic	String containing logic in REDCap format.
var	String with the name of the variable containing the logic.

Value

List containing the logic in R format and its evaluation.

Examples

```

rd_rlogic(covican,
  logic = "if([exc_1]='1' or [inc_1]='0' or [inc_2]='0' or [inc_3]='0',1,0)",
  var = "screening_fail_crit")

```

rd_transform *Transformation of the Raw Data*

Description

This function transforms the raw REDCap data read by the ‘redcap_data’ function. It returns the transformed data and dictionary, along with a summary of the results of each step.

Usage

```
rd_transform(
  ...,
  data = NULL,
  dic = NULL,
  event_form = NULL,
  checkbox_labels = c("No", "Yes"),
  checkbox_na = FALSE,
  exclude_recalc = NULL,
  exclude_to_factor = NULL,
  delete_vars = NULL,
  delete_pattern = c("_complete", "_timestamp"),
  final_format = "raw",
  which_event = NULL,
  which_form = NULL,
  wide = NULL
)
```

Arguments

...	Output of the ‘redcap_data’ function, which is a list containing the data frames of the data, dictionary and event_form (if needed) of the REDCap project.
data	Data frame containing the data read from REDCap. If the list is specified, this argument is not necessary.
dic	Data frame containing the dictionary read from REDCap. If the list is specified, this argument is not necessary.
event_form	Data frame containing the correspondence of each event with each form. If the list is specified, this argument is not necessary.
checkbox_labels	Character vector with the names for the two options of every checkbox variable. Default is ‘c(‘No’, ‘Yes’)’.
checkbox_na	Logical indicating if checkboxes values with branching logic should be set to missing only when the branching logic is missing (‘FALSE’), or also when the branching logic isn’t satisfied (‘TRUE’). The default is ‘FALSE’.
exclude_recalc	Character vector with the names of variables that should not be recalculated. Useful for projects with time-consuming recalculations of certain calculated fields.

<code>exclude_to_factor</code>	Character vector with the names of variables that should not be transformed to factors.
<code>delete_vars</code>	Character vector specifying the variables to exclude.
<code>delete_pattern</code>	Character vector specifying the regex pattern for variables to be excluded. By default, variables ending with <code>'_complete'</code> and <code>'_timestamp'</code> will be removed.
<code>final_format</code>	Character string indicating the final format of the data. Options are <code>'raw'</code> , <code>'by_event'</code> or <code>'by_form'</code> . <code>'raw'</code> (default) returns the transformed data in its original structure, <code>'by_event'</code> returns it as a nested data frame by event, and <code>'by_form'</code> returns it as a nested data frame by form.
<code>which_event</code>	Character string indicating a specific event to return if the final format is <code>'by_event'</code> .
<code>which_form</code>	Character string indicating a specific form to return if the final format is <code>'by_form'</code> .
<code>wide</code>	Logical indicating if the data split by form (if selected) should be in a wide format (<code>'TRUE'</code>) or a long format (<code>'FALSE'</code>).

Value

A list with the transformed dataset, dictionary, `event_form`, and the results of each transformation step.

Examples

```
# Basic transformation
rd_transform(covican)

# For customization of checkbox labels (example)
rd_transform(covican,
             checkbox_labels = c("Not present", "Present"))
```

<code>recalculate</code>	<i>Recalculate REDCap Calculated Fields</i>
--------------------------	---

Description

This function recalculates each calculated field if the logic can be transcribed to R. Note that calculated fields containing smart-variables or variables from other events cannot be transcribed.

The function returns the dataset and dictionary with the recalculated variables appended (named as the original field plus `'_recalc'`), along with a summary table of the recalculation results.

Usage

```
recalculate(data, dic, event_form = NULL, exclude_recalc = NULL)
```

Arguments

<code>data</code>	Data frame containing data from REDCap.
<code>dic</code>	Data frame containing the dictionary read from REDCap.
<code>event_form</code>	Data frame containing the correspondence of each event with each form.
<code>exclude_recalc</code>	Character vector with the names of the variables that should not be recalculated. Useful for projects with time-consuming recalculations for certain calculated fields.

<code>redcap_data</code>	<i>Read REDCap data</i>
--------------------------	-------------------------

Description

This function allows users to read datasets from a REDCap project into R for analysis, either by exporting the data or via an API connection.

The REDCap API serves as an interface for communication with REDCap and the server without requiring interaction through the REDCap interface.

[Important] To read exported data from REDCap, please follow these steps:

- Use REDCap's 'Export Data' function.
- Select the 'R Statistical Software' format.
- REDCap will then generate two files:
 - A CSV file containing all observations of the REDCap project.
 - An R file with the necessary code to complete each variable's information and import them.
- Ensure these files, along with the dictionary and event-mapping, are in the same directory.

Usage

```
redcap_data(
  data_path = NA,
  dic_path = NA,
  event_path = NA,
  uri = NA,
  token = NA,
  filter_field = NULL,
  survey_fields = FALSE
)
```

Arguments

<code>data_path</code>	Character string specifying the path of the R file from which the dataset will be read.
<code>dic_path</code>	Character string with the path of the dictionary.

event_path	Character string specifying the path of the file containing the correspondence between each event and each form (downloadable via the ‘Designate Instruments for My Events‘ tab within the ‘Project Setup‘ section of REDCap).
uri	The URI (Uniform Resource Identification) of the REDCap project.
token	Character vector containing the generated token.
filter_field	Character vector specifying the fields of the REDCap project desired to be imported into R (via API connection only).
survey_fields	Logical indicating whether the function should download all the survey-related fields of the REDCap project (via API connection only).

Value

A list containing the dataset and the dictionary of the REDCap project. If ‘event_path‘ is specified, it will also contain a third element with the correspondence of the events and forms of the project.

Note

For further use of the package, it’s recommended to use the ‘dic_path‘ argument to read the dictionary, as all other functions require it for proper functioning.

Examples

```
## Not run:
# Exported files from REDCap

dataset <- redcap_data(data_path = "C:/Users/username/example.r",
                      dic_path = "C:/Users/username/example_dictionary.csv",
                      event_path = "C:/Users/username/events.csv")

# API connection

dataset_api <- redcap_data(uri = "https://redcap.idibell.cat/api/",
                          token = "55E5C3D1E83213ADA2182A4BFDEA") # This token is fictitious

## End(Not run)
```

split_event	<i>Creation of a Data Frame with Variables from All Forms of a Specified Event</i>
-------------	--

Description

This function generates a nested dataset filtered by each event, containing only the variables associated with each event. It uses the provided data, dictionary, and event-form mapping. You can choose to return data for a specific event.

Usage

```
split_event(data, dic, event_form, which = NULL)
```

Arguments

data	Data frame containing data from REDCap.
dic	Data frame containing the dictionary read from REDCap.
event_form	Data frame containing the correspondence of each event with each form.
which	Character string specifying an event if only data for that event is desired.

split_form

Creation of a Data Frame with Variables from a Specified Form

Description

This function generates a nested dataset containing only the variables associated with each form, using the provided data, dictionary, and event-form mapping. You can choose to return data for a specific form.

Usage

```
split_form(data, dic, event_form = NULL, which = NULL, wide = FALSE)
```

Arguments

data	Data frame containing data from REDCap.
dic	Data frame containing the dictionary read from REDCap.
event_form	Data frame containing the correspondence of each event with each form.
which	Character string specifying a form if only data for that form is desired.
wide	Logical indicating if the dataset should be returned in a wide format ('TRUE') or long format ('FALSE').

to_factor	<i>Convert Variables to Factors</i>
-----------	-------------------------------------

Description

This function converts all variables in the dataset to factors, except those specified in the 'exclude' parameter.

Usage

```
to_factor(data, dic, exclude = NULL)
```

Arguments

data	Data frame containing the REDCap data.
dic	Data frame containing the REDCap dictionary.
exclude	Character vector specifying the names of variables that should not be converted to factors. If 'NULL', all variables will be converted.

transform_checkboxes	<i>Transformation of Checkboxes with Branching Logic</i>
----------------------	--

Description

This function inspects all the checkboxes in the study to determine if they have a branching logic. If a branching logic is present and its result is missing, the function will input a missing value into the checkbox. If 'checkbox_na' is 'TRUE', the function will additionally input a missing value when the branching logic isn't satisfied, not just when it is missing. If a branching logic cannot be found or the logic cannot be transcribed due to the presence of smart variables, the variable is added to a list of reviewable variables that will be printed.

The function returns the dataset with the transformed checkboxes and a table summarizing the results.

Usage

```
transform_checkboxes(data, dic, event_form = NULL, checkbox_na = FALSE)
```

Arguments

data	Data frame containing data from REDCap.
dic	Data frame containing the dictionary read from REDCap.
event_form	Data frame containing the correspondence of each event with each form.
checkbox_na	Logical indicating if values of checkboxes with branching logic should be set to missing only when the branching logic is missing ('FALSE'), or also when the branching logic is not satisfied ('TRUE'). The default is 'FALSE'.

Index

* datasets

covican, [4](#)

check_queries, [3](#)

checkbox_names, [2](#)

covican, [4](#)

fill_data, [5](#)

rd_event, [6](#)

rd_export, [7](#)

rd_insert_na, [8](#)

rd_query, [9](#)

rd_rlogic, [11](#)

rd_transform, [12](#)

recalculate, [13](#)

redcap_data, [14](#)

split_event, [15](#)

split_form, [16](#)

to_factor, [17](#)

transform_checkboxes, [17](#)