Package 'REDCapDM'

March 3, 2023

Type Package

Title 'REDCap' Data Management

Version 0.4.0

Maintainer João Carmezim <ubidi@idibell.cat>

Description Access and manage 'REDCap' data. 'REDCap' (Research Electronic Data CAPture; <https://projectredcap.org>) is a web application for building and managing online surveys and databases developed at Vanderbilt University. The API allows users to programmatic access data and project meta data (such as the data dictionary) from the web. This package allows us to read 'REDCap' data, exported or using an API connection, identify missing or extreme values, identify missing 'REDCap' events in each observation, do a follow-up of the queries initially identified and it also facilitates the process of data management.

License MIT + file LICENSE

BugReports https://github.com/ubidi/REDCapDM/issues

Encoding UTF-8

RoxygenNote 7.2.3

Imports dplyr, REDCapR, janitor, stringr, magrittr, tidyr, Hmisc, utils, purrr, tidyselect, tibble, rlang

Suggests knitr, rmarkdown, kableExtra

VignetteBuilder knitr

Depends R (>= 2.10)

LazyData true

NeedsCompilation no

Author João Carmezim [aut, cre],
Judith Peñafiel [aut],
Pau Satorra [aut],
Esther García [aut],
Natàlia Pallarés [aut],
Cristian Tebé [aut]

Repository CRAN

Date/Publication 2023-03-03 14:10:05 UTC

R topics documented:

`checkbox_names` *Change checkboxes names into the name of their options*

Description

Function that returns both data and dictionary with the name of the checkboxes transformed by the name 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 name of the variables in the data and the REDCap label in its name.
`checkbox_labels`	Character vector with the names that will have the two options of every checkbox variable. Default is c('No', 'Yes').

check_queries *Check modifications between two report of queries*

Description

This function compares a previous report of queries with a new one and allows you to check 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 report of queries.
new	New version of the report of queries. This object will be used to determine the status of each query.
report_title	Character string with the report's title.

Value

A list composed by a data frame that combines all queries and includes a column that shows the status of the queries (new, modified, or unchanged) when compared to the previous report of queries. In addition to this data frame, it also has a summary of the total number of queries per category.

Examples

```
data_old <- rd_query(covican,
                     variables = "copd",
                     expression = "%in%NA",
                     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-hematological 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 containing three dataframes: the first one with the data, the second one with the dictionary (‘codebook‘) of the REDCap project and the final 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

`dic_checkboxes` *Change the names of checkboxes variables in the REDCap dictionary*

Description

Auxiliary function to checkbox_names. Adds to the dictionary all variables that correspond to all the options of checkbox (with the name as it is in the data) and remove the original general checkbox variable.

Usage

```
dic_checkboxes(var_check, dic, labels, checkbox_labels = c("No", "Yes"))
```

Arguments

`var_check`	Character vector containing the names of those variables that are checkboxes.
`dic`	Dataset containing the REDCap dictionary.
`labels`	Named character vector with the name of the variables in the data and the REDCap label in its name.
`checkbox_labels`	Character vector with the names that will have the two options of every checkbox variable. Default is c('No', 'Yes').

`fill_data` *Fill rows with the values in one event*

Description

Function that with one particular variable and event it fills all the rows in the data with the value in that particular event

Usage

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

Arguments

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

---

rd_event *Identification of missing event/s*

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 will not export events that do not have information. This function allows you to point out which record identifiers do not have information of a determined 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 and the dictionary and the event if it's needed. Can 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	Vector with the REDCap's events names to be analyzed.
filter	A filter to apply to the dataset. This argument can be used to identify the missing events on a subgroup of the dataset.
query_name	Description of the query. It can be defined as the same one for all events or you can define one for each event By default, the function will define the description as 'The event [event] is missing' for each event'.

addTo	Data frame corresponding to a prior report of queries to which you can add the new data frame of queries. By default, the function will always generate a new data frame without taking into account former reports.
report_title	Character string with the report's title.
report_zeros	Logical. If 'TRUE', it returns a report including events with zero queries.
link	List containing project information used to create a web link to each query.

Value

A dataframe with 9 columns meant to help the user identify each missing event and a table with the total of queries per variable.

Examples

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

---

rd_insert_na *Insert missing using a filter*

Description

Function that allows you to manually input a missing to some variables ('vars') when some filters ('filter') are satisfied. Useful for checkboxes without a gatekeeper question in the branching logic. Take in account that the variable will be transformed only in the events where both the variable and the filter evaluation are present, so they need to 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 and the dictionary and the event if it's needed. Can 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 those variables to transform.
filter	Character vector containing the logic to be directly evaluated. When each logic is TRUE the corresponding variable specified in 'vars' will be put to missing.

Value

transformed data with the specified variables converted.

Examples

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

`rd_query` *Identification of queries*

Description

This function allows you to identify queries by using a specific expression. It can be used to identify missing values or values that fall outside the lower and upper limit of a variable.

Usage

```
rd_query(
  ...,
  data = NULL,
  dic = NULL,
  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()
)
```

Arguments

`...`	List containing the data and the dictionary and the event if it's needed. Can 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.

`variables` Vector with variables names from the database that will be checked. If this argument alongside with the argument 'expression' are unspecified, this function will look for abnormal values using the minimum and maximum of each variable in the dataset (information contained in the dictionary).

`expression` Expression that will be applied to the chosen variables, for example, "<170". If this argument is unspecified, this function will look for abnormal values using the minimum and maximum of each variable in the dataset (information contained in the dictionary).

`negate` Logical value indicating whether or not to negate the defined expression. Defaults to 'FALSE'.

`event` REDCap's event name to be analyzed. If your REDCap project has events, you should use this argument in order to name the event to which the defined variables belong.

`filter` A filter to apply to the dataset. This argument can be used to, for example, apply the branching logic of a determined variable.

`addTo` Data frame corresponding to a prior report of queries to which you can add the new data frame of queries. By default, the function will always generate a new data frame without taking into account former reports.

`variables_names` Vector with the description of each variable. By default, the function will automatically pick the description of each variable from the dictionary of the dataset.

`query_name` Description of the query. It can be defined as the same one for all variables or you can define one for each variable. By default, the function will define the description as 'The value is [value] and it should not be [expression]' for each one of the variables'.

`instrument` REDCap's instrument to which the variables belong. It can be defined as the same one for all variables or you can define one for each variable. By default, the function will automatically pick the corresponding instrument of each variable from the dictionary of the dataset.

`report_title` Character string with the report's title.

`report_zeros` Logical. If 'TRUE', it returns a report including 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.

Value

A list with a data frame with 9 columns meant to help the user identify each query and a table with the total of queries per variable.

Examples

```
# Missings
example <- rd_query(covican,
                    variables = c("copd", "age"),
                    expression = c("%in%NA", "%in%NA"),
                    event = "baseline_visit_arm_1")
example

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

# Using a filter
example <- rd_query(covican,
                    variables = "potassium",
                    expression = "%in%NA",
                    event = "baseline_visit_arm_1",
                    filter = "available_analytics=='1'")
example
```

`rd_rlogic` *REDCap logic into R logic*

Description

This function allows you to transcribe REDCap logic to R logic. WARNING: If the REDCap logic involves some smart-variables this function will not be able to transform it.

Usage

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

Arguments

`...`	List containing the data and the dictionary and the event if it's needed. Can 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.
`logic`	String containing a logic in REDCap format.
`var`	string containing the name of the variable that contains 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

Function that transforms the raw data from REDCap read by the function 'redcap_data'. It returns the transformed data and dictionary along with the summary of the results of each step.

Usage

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

Arguments

`...`	List containing the data and the dictionary and the event_form if it's needed. Can 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.

`event_path` Character string with the pathname of the file with the correspondence between each event and each form (it can be downloaded in 'Designate Instruments for My Events' inside the 'Project Setup' section of REDCap).

`checkbox_labels` Character vector with the names that will have the two options of every checkbox variable. Default is 'c('No', 'Yes')'.

`checkbox_na` Logical indicating if values of checkboxes that have a branching logic have to set to missing only when the branching logic is missing (meaning that some of the variables specified in it are missing) or also when the branching logic isn't satisfied (true).

`exclude_to_factor` Character vector with the names of the variables that do not have to be transformed to factors.

`keep_labels` Logical indicating if the labels have to be kept or not.

`delete_vars` Character vector specifying the pattern that will contain variables to exclude. By default, variables ending up with '_complete' will be removed.

`final_format` Character string indicating the final arrangement format of the data that the function will return. Choose one of 'raw', 'by_event' or 'by_form'. 'raw' (default) will return the transformed data with the original structure. 'by_event' will return the transformed data as a nested data frame by event. 'by_form' will return the transformed data as a nested data frame by form.

`which_event` Character string indicating if only one event has to be returned if the final format selected is 'by_event'.

`which_form` Character string indicating if only one form has to be returned if the final format selected is 'by_form'.

`wide` Logical indicating if the data split by form (if selected) has to be in a wide format or in a long one.

Value

List with the transformed dataset, dictionary and the results

Examples

```
rd_transform(covican)

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

recalculate *Recalculate REDCap calculated fields*

Description

Function that recalculates every calculated field if the logic can be transcribed to R. Recall that calculated fields with smart-variables in the logic or variables in other events cannot be transcribed.

The function will return the dataset and dictionary with the added recalculated variables (the name of the calculated field + '_recalc') along with a table that shows the summary of the results.

Usage

```
recalculate(data, dic, event_form = 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.

REDCapDM *REDCapDM: A package to create queries reports of a determined REDCap dataset.*

Description

The REDCapDM package provides three main functions that allow us to read a dataset exported from REDCap, identify a variety of queries, check over time which of the old queries were resolved and even do a pre-processing of the data. This package was built and tested with databases created using REDCap v12.4.17.

Details

REDCapDM functions:

- redcap_data: used to read data exported from REDCap or through an API connection.

- rd_expression: identification of queries.

- rd_event: identification of missing events per record identifier.

- check_queries: used to compare current queries with an old report to determine which ones have been modified, which remain unchanged, and if there are any new queries.

- rd_transform: pre-processing a dataset.

- rd_rlogic: transcribes redcap logic to R logic.

- rd_insert_na: manually input a missing value for specified variables using a filter.

`redcap_data` *Read REDCap data*

Description

This function allows users to read datasets from a REDCap project, through exported data or an API connection, into R for analysis.

The REDCap API is an interface that allows communication with REDCap and server without going through the interactive REDCap interface.

Usage

```
redcap_data(
  data_path = NA,
  dic_path = NA,
  event_path = NA,
  uri = NA,
  token = NA
)
```

Arguments

`data_path`	Character string with the pathname of the R file to read the dataset from.
`dic_path`	Character string with the pathname of the dictionary.
`event_path`	Character string with the pathname of the file with the correspondence between each event and each form (it can be downloaded in 'Designate Instruments for My Events' inside the 'Project Setup' section of REDCap)
`uri`	The URI (uniform resource identifier) of the REDCap project.
`token`	Character vector with the code of the token.

Value

List with the dataset and the dictionary of the corresponding REDCap project. If the event_path is specified it will also contain a third element with the correspondence of the events & forms of the project.

Note

If you will give further use to the package, we advise you to use the argument 'dic_path' to read your dictionary, since all other functions need it in order to run properly.

To read exported data you have to first use REDCap's 'Export Data' function and select the format 'R Statistical Software', then it will generate one CSV file with all observations and an R file with the necessary code to complete each variable information.

split_event — *Creation of a data frame with variables of all the forms of a specified event*

Description

Function that given the data, the dictionary and the mapping between forms and events it creates a dataset with the variables of all the forms that are in this event. It can be chosen to return only the data from the specified 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	Specify an event if only data for the desired event is wanted.

split_form — *Creation of a data frame with variables of a specified form*

Description

Function that given the data, the dictionary and the mapping between forms and events it creates a dataset with the variables that are in this form for all events. It can be chosen to return only the data from the specified 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	Specify a form if only data for the desired form is wanted.
wide	If the dataset needs to be in a wide format or not (long).

to_factor *Convert variables to factors*

Description

Function that converts every variable except those specified to factor.

Usage

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

Arguments

data	Dataset containing the REDCap data.
exclude	Character vector containing the names of those variables that will not be converted to factors. If 'NULL', all variables will be converted.

transform_checkboxes *Transformation of checkboxes in case of having a branching logic*

Description

Inspects all the checkboxes of the study and looks if there is a branching logic. If there is one, when the logic of the branching logic is missing it directly inputs a missing to the checkbox. If checkbox_na is TRUE additionally it will put a missing when the branching logic isn't satisfied and not only when the logic is missing. If a branching logic cannot be found or the logic cannot be transcribed because of the presence of some smart variables, the variable is added in the list of the reviewable ones that will be printed.

The function will return the dataset with the transformed checkboxes along with a table that shows a summary of the results.

Usage

```
transform_checkboxes(
  data,
  dic,
  event_form,
  checkbox_labels = c("No", "Yes"),
  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_labels	Character vector with the names that will have the two options of every checkbox variable. Default is 'c('No', 'Yes')'.
checkbox_na	Logical indicating if values of checkboxes that have a branching logic have to set to missing only when the branching logic is missing (meaning that some of the variables specified in it are missing) or also when the branching logic isn't satisfied (true).

---

transform_name *Auxiliary function to 'checkbox_names'*

Description

Auxiliary function to checkbox_names. It changes the name of the checkbox variable to the name of the option it corresponds

Usage

```
transform_name(var_check, name, labels)
```

Arguments

var_check	a character vector containing the names of those variables that are checkboxes
name	a character element with the original name of the checkbox variable that has to be changed
labels	a named character vector with the name of the variables in the data and the REDCap label in its name

Index