Package: dataMeta (via r-universe)

August 31, 2024

Title Create and Append a Data Dictionary for an R Dataset

Date 2017-08-11

Version 0.1.1.9000

Description Designed to create a basic data dictionary and append to
the original dataset's attributes list. The package makes use
of a tidy dataset and creates a data frame that will serve as a
linker that will aid in building the dictionary. The dictionary
is then appended to the list of the original dataset's
attributes. The user will have the option of entering variable
and item descriptions by writing code or use alternate
functions that will prompt the user to add these.

Depends R (>= 3.3.2)

License GPL-3

URL https://github.com/dmrodz/dataMeta

BugReports https://github.com/dmrodz/dataMeta/issues

Encoding UTF-8

LazyData true

Imports dplyr

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 6.1.1

Repository https://dmrodz.r-universe.dev

RemoteUrl https://github.com/dmrodz/datameta

RemoteRef HEAD

RemoteSha f4a2d66e67f53da9c962d1aa4e44f83a15292cea

Contents

| build_dict | *Build a data dictionary for a dataset.* |

Description

build_dict constructs a data dictionary for a dataset with the aid of a data linker. This is the second
function used in this package. For the function to run, the following parameters are needed.

Usage

```
build_dict(my.data, linker, option_description = NULL,
  prompt_varopts = TRUE, na.rm = FALSE)
```

Arguments

my.data	Data.frame. The data set for which the user is creating the dictionary for.
linker	Data.frame. A data frame that has the variable names from the original dataset, and also a avriable type that will tell the dictionary whether to list unique item options or a range of values for each variable name.
option_description	
	A vector that has the description of each variable option in the order in which these appear and depending on how the variable type was set while building the linker data frame. If using the prompt_varopts option, this value must be NULL.
prompt_varopts	Logical. Whether to add the option_description manually as prompted by R. Default is set to TRUE. If FALSE, an option_description vector must be provided.
na.rm	Logical. Whether to remove NA when determining the range for variables with variable_type == 0 in linker

Value

A data frame that will serve as a data dictionary for an original dataset. The user will have the
option to add this dictionary as an attribute to the original dataset with the other package functions.

Examples

```
# example original data set for which a dictionary will be made
data("esoph")
my.data <- esoph

# Linker: Add description for each variable names and variable type
variable_description <- c("age group in years", "alcohol consumption in gm/day",
"tobacco consumption in gm/day", "number of cases (showing a range)",
"number of controls (showing range)")

variable_type <- c(0, 0, 0, 0, 0)
linker <- build_linker(my.data = my.data, variable_description = variable_description,
variable_type = variable_type)
linker

# Data dictionary
# For this data set, no further option description is needed.
dictionary <- build_dict(my.data = my.data, linker = linker, option_description = NULL,
prompt_varopts = FALSE)
dictionary
```

---

build_linker	*Build a linker data frame.*

Description

build_linker constructs a data frame that will be an intermediary between the original dataset and the data dictionary. This is the first function used in this package. For the function to run, the following parameters are needed.

Usage

```
build_linker(my.data, variable_description, variable_type)
```

Arguments

my.data   Data.frame. The data set for which the user is creating the dictionary for.

variable_description

A string vector representing the different descriptions that the user will give to each variable name from the original dataset. These need to be in the same order as the original dataset's variable names.

variable_type   A vector of integers with values 0 or 1, only. Use 0 for variable names for which a range of values will be presented and 1 to show unique cases of each variable name option. See examples, below.

Value

If the original dataset supplied as my.data is of class data.frame; the variable description items are in the same order as the orignal dataset's variable names; and the variable_type intgeer vector values are 0 or 1, then a small data frame is produced with variable_names, variable_description, variable_type columns. This dataframe will serve as a linker data frame to be able to construct the data dictionary.

Examples

```
# example original data set for which a dictionary will be made
data("esoph")
my.data <- esoph

# Add description for each variable names and variable type
variable_description <- c("age group", "alcohol consumption", "tobacco consumption",
"number of cases", "number of controls")

variable_type <- c(0, 0, 0, 0, 0)

linker <- build_linker(my.data = my.data, variable_description = variable_description,
variable_type = variable_type)
linker

## Not run:
variable_description <- c("age group", "alcohol consumption", "tobacco consumption",
"number of cases", "number of controls")
variable_type <- c(0, 2, 0, 0, 0)
linker <- build_linker(my.data = my.data, variable_description = variable_description,
variable_type = variable_type)
linker

## End(Not run)
```

---

dataMeta	*dataMeta: Create and Append a Data Dictionary for an R Dataset*

Description

The dataMeta package provides three main functions: build_linker, build_dict and incorporate_attr. The build_linker and incorporate_attr functions have prompt options called: prompt_linker and prompt_attr, respectively.

dataMeta functions

build_linker This function will build a data frame that will serve as a link between your dataset to the creation of the data dictionary. prompt_linker This function is an alternate function to build the linker. It will prompt you for variable name descriptions in real time. build_dict This function will build a data dictionary using the linker and the original dataset. incorporate_attr This function will

incorporate the data dictionary that is created with the build_dict option into the R dataset as an attribute, along with other metadata that may be needed. prompt_attr This function will prompt the user for options related to the metadata that will be added to the R dataset. This is an alternative to the incorporate_attr function. save_it This function will save your new data with its attributes as an R dataset.

---

incorporate_attr	*Incorporate attributes as metadata to an original dataset.*

Description

`incorporate_attr` adds attributes to an original dataset as metadata, including a data dictionary, among other attributes. This is the third function used in this package. For the function to run, the following parameters are needed.

Usage

```
incorporate_attr(my.data, data.dictionary, main_string)
```

Arguments

`my.data`	Data.frame. The data set to add attributes as metadata.
`data.dictionary`	
	Data frame. The data dictionary has all variable names, and variable descriptions that will explain an original dataset.
`main_string`	A character string describing the original dataset.

Value

This function will return an R dataset containing metadata stored in its attributes. Attributes added will include: a data dictionary, number of columns, number of rows, the name of the author or user who created the dictionary and added it, the time when it was last edited and a brief description of the original dataset.

Examples

```
# example original data set for which a dictionary will be made
data("esoph")
my.data <- esoph

# Linker: Add description for each variable names and variable type
variable_description <- c("age group in years", "alcohol consumption in gm/day",
"tobacco consumption in gm/day", "number of cases (showing range)",
"number of controls (showing range)")
variable_type <- c(0, 0, 0, 0, 0)
linker <- build_linker(my.data = my.data, variable_description = variable_description,
variable_type = variable_type)
linker
```

```
# Data dictionary
# For this data set, no further option description is needed.
dictionary <- build_dict(my.data = my.data, linker = linker, option_description = NULL,
prompt_varopts = FALSE)
dictionary

# Create main_string for attributes
main_string <- "This dataset describes tobacco and alcohol consumption at different age groups."
complete_dataset <- incorporate_attr(my.data = my.data, data.dictionary = dictionary,
main_string = main_string)
complete_dataset
attributes(complete_dataset)
```

---

my.data   *my.data*

Description

Data containing Zika cases as reported by the United States Virgin Islands department of Health
and scraped into CDC's public Zika data github repository.

Usage

```
data(my.data)
```

Format

A data frame of 32 observations and 9 variables.

report_date Date when report was published by USVI Department of Health, YYYY-mm-dd

location Regional location by name

location_type The type of location

data_field The type of case presented

data_field_code A code to identify the data_field

time_period The time period of the cases

time_period_type The units of the time period

value The number of observations under a specificc data_field

unit The unit of the number of observations, cases, municipalities...

Source

https://github.com/cdcepi/zika/blob/master/USVI/USVI_Zika/data/USVI_Zika-2017-01-03.csv

References

Dania M. Rodriguez, Michael A Johansson, Luis Mier-y-Teran-Romero, moiradillon2, eyq9, Yo-JimboDurant, . . . Daniel Mietchen. (2017). cdcepi/zika: March 31, 2017 [Data set]. Zenodo. (zenodo)

---

prompt_attr	*Incorporate attributes as metadata to an original dataset as prompted by the function.*

Description

`prompt_attr` adds attributes to an original dataset as metadata, including a data dictionary, among other attributes as prompted by the function.

Usage

```
prompt_attr(my.data, data.dictionary)
```

Arguments

my.data   Data.frame. The data set to add attributes as metadata.

data.dictionary

Data frame. The data dictionary has all variable names, and variable descriptions that will explain an original dataset.

Details

This is a variation of the third function used in this package. For the function to run, the following parameters are needed.

Value

This function will return an R dataset containing metadata stored in its attributes. The function will prompt the user for a main description. Attributes added will include: a data dictionary, the name of the author or user who created the dictionary and added it, the time when it was last edited and a brief description of the original dataset.

---

prompt_linker   *Build a linker data frame: prompt option.*

Description

prompt_linker this function will prompt the user for a variable description and variable type to construct a data frame that will be an intermediary between the original dataset and the data dictionary. This is a variation of the first function used in this package. For the function to run, the following parameters are needed.

Usage

```
prompt_linker(my.data)
```

Arguments

my.data   Data.frame. The dataset for which the user is creating the dictionary for.

Value

If the original dataset supplied as my.data is of class data.frame; the variable description items are in the same order as the orignal dataset's variable names; and the variable_type intgeer vector values are 0 or 1, then a small data frame is produced with variable_names, variable_description, variable_type columns. This dataframe will serve as a linker daata frame to be able to construct the data dictionary.

---

save_it   *Save dataset with attributes.*

Description

save_it saves datset with attributes stored as metadata as an R dataset (.rds) into the current working directory. This is the final function used in this package. For the function to run, the following parameters are needed.

Usage

```
save_it(x, name_of_file)
```

Arguments

x   Data.frame. Dataset that has attributes added, including a data dictionary.

name_of_file   Text string to name the file.

Value

This function will save the dataset along with its attributes as an R dataset (.rds) to the current working directory.

Examples

```
# example original data set for which a dictionary will be made
data("esoph")
my.data <- esoph

# Linker: Add description for each variable names and variable type
variable_description <- c("age group in years", "alcohol consumption in gm/day",
"tobacco consumption in gm/day", "number of cases (showing a range)",
"number of controls (showing range)")

variable_type <- c(0, 0, 0, 0, 0)

linker <- build_linker(my.data = my.data, variable_description = variable_description,
variable_type = variable_type)
linker

# Data dictionary
# For this data set, no further option description is needed.
dictionary <- build_dict(my.data = my.data, linker = linker, option_description = NULL,
prompt_varopts = FALSE)
dictionary

# Create main_string for attributes
main_string <- "This dataset describes tobacco and alcohol consumption at different age groups."
complete_dataset <- incorporate_attr(my.data = my.data, data.dictionary = dictionary,
main_string = main_string)
complete_dataset
attributes(complete_dataset)

# Save it
# Name of file
name_of_file <- "my new data set"
save_it(x = complete_dataset, name_of_file = name_of_file)
```

Index