

Package ‘inspectr’

January 30, 2017

Type Package

Title Perform Basic Checks of Dataframes

Version 1.0.0

Maintainer Jennifer Brussow <jbrussow@gmail.com>

Description Check one column or multiple columns of a dataframe using the preset basic checks or your own functions. Enables checks without knowledge of lapply() or sapply().

License CC BY-SA 4.0

Encoding UTF-8

LazyData true

Imports openxlsx

Depends R (>= 3.0.1)

RoxygenNote 5.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Jennifer Brussow [aut, cre]

Repository CRAN

Date/Publication 2017-01-30 21:41:32

R topics documented:

character_blanks_check	2
character_check	2
col_check	3
dataset	4
date_check	4
greater_than	5
greater_than_equalto	5
inspectr	6

less_than	6
less_than_equalto	7
numeric_check	7
three_col_check	8
two_col_check	9
val_check	10

Index	11
--------------	-----------

character_blanks_check

Check whether all values are either character strings. Blanks ("") and NA values are not permitted.

Description

To be used with [col_check](#).

Usage

character_blanks_check(x)

Arguments

x the data that enters the function (the column specified in [col_check](#))

character_check

Check whether all values are character strings or blanks (""). NA values are not permitted.

Description

To be used with [col_check](#).

Usage

character_check(x)

Arguments

x the data that enters the function (the column specified in [col_check](#))

col_check	<i>Check a single column for data fidelity.</i>
-----------	---

Description

Check a single column for data fidelity.

Usage

```
col_check(colname, data, fun, output = FALSE, stage = NULL, ...)
```

Arguments

colname	character string specifying the name of the column within your dataframe.
data	the dataframe containing the data.
fun	the check function you'd like to apply to the data.
output	logical. If FALSE, the function returns a dataframe containing only records that failed the specified check. If TRUE, invisibly prints an excel output file containing only the records that failed the specified check.
stage	An optional character string that can be used to specify the stage of the checking process in which the check is occurring. Only useful if output = TRUE. If a value is specified, a that value is prefixed to the output file; if no value is given, no stage prefix is attached.
...	arguments to be passed through to the function specified in fun

Value

col_check(output = FALSE) returns a dataframe containing only records that failed the specified check.

col_check(output = TRUE) invisibly prints an excel output file containing only records that failed the specified check.

Examples

```
col_check(colname = "ID_var", data = dataset, fun = numeric_check,  
output = TRUE, stage = "1-Reasonableness")
```

```
col_check(colname = "FName", data = dataset, fun = character_check,  
output = FALSE)
```

dataset	<i>Demonstration data created to resemble data collected from an educational assessment.</i>
---------	--

Description

Demonstration data created to resemble data collected from an educational assessment.

Usage

dataset

Format

A data frame with 20 rows and 5 variables:

ID_var A student identification variable

FName First names of each student

Var1 One score

Var2 A second score

Perf_Lvl Each student's performance level

dates Birthdates

date_check	<i>Check whether all values fall within a date range. NA values are also accepted.</i>
------------	--

Description

To be used with [col_check](#).

Usage

```
date_check(x, begin, end, format = "%m/%d/%Y")
```

Arguments

x	the data that enters the function (the column specified in col_check)
begin	the beginning acceptable date.
end	the last acceptable date.
format	a character string specifying the date format.

greater_than	<i>Check whether values in column one are greater than their corresponding values in the second column.</i>
--------------	---

Description

To be used with [two_col_check](#).

Usage

```
greater_than(col1, col2)
```

Arguments

col1	the first column of data, specified in col_check)
col2	the second column of data, specified in col_check)

greater_than_equalto	<i>Check whether values in column one are greater than or equal to their corresponding values in the second column.</i>
----------------------	---

Description

To be used with [two_col_check](#).

Usage

```
greater_than_equalto(col1, col2)
```

Arguments

col1	the first column of data, specified in col_check)
col2	the second column of data, specified in col_check)

inspectr	<i>inspectr: A package for performing fidelity checks on messy dataframes.</i>
----------	--

Description

The inspectr package contains two classes of functions: column checks and basic fidelity checks.

Column check functions

These are the basic functions used to perform checks. Each function checks one column for data fidelity, and functions exist to check that column against one or two additional columns. A data frame and a column name (or names) go in; a filtered set of records exhibiting issues comes out (either as a dataframe or as an .xlsx document - your choice!)

Basic fidelity checks

These functions are designed to be used with the column check functions. They perform basic checks on the data, like ensuring that all data in a column are of the same type or ensuring that all values in column 1 are less than their corresponding values in column 2.

less_than	<i>Check whether values in column one are less than their corresponding values in the second column.</i>
-----------	--

Description

To be used with [two_col_check](#).

Usage

```
less_than(col1, col2)
```

Arguments

col1	the first column of data, specified in col_check)
col2	the second column of data, specified in col_check)

less_than_equalto	<i>Check whether values in column one are less than or equal to their corresponding values in the second column.</i>
-------------------	--

Description

To be used with [two_col_check](#).

Usage

```
less_than_equalto(col1, col2)
```

Arguments

col1	the first column of data, specified in col_check)
col2	the second column of data, specified in col_check)

numeric_check	<i>Check whether all values are numeric.</i>
---------------	--

Description

To be used with [col_check](#).

Usage

```
numeric_check(x)
```

Arguments

x	the data that enters the function (the column specified in col_check)
---	--

three_col_check	<i>Check a column for data fidelity using criteria related to two additional columns.</i>
-----------------	---

Description

Check a column for data fidelity using criteria related to two additional columns.

Usage

```
three_col_check(colname1, colname2, colname3, data = data, fun,
  output = FALSE, stage = NULL, ...)
```

Arguments

colname1	character string specifying the name of the column within your dataframe that will be the subject of the checks.
colname2	character string specifying the name of a second column to be used in the check criteria.
colname3	character string specifying the name of a third column to be used in the check criteria.
data	the dataframe containing the data.
fun	the check function you'd like to apply to the data.
output	logical. If FALSE, the function returns a dataframe containing only records that failed the specified check. If TRUE, invisibly prints an excel output file containing only the records that failed the specified check.
stage	An optional character string that can be used to specify the stage of the checking process in which the check is occurring. Only useful if output = TRUE. If a value is specified, a that value is prefixed to the output file; if no value is given, no stage prefix is attached.
...	arguments to be passed through to the function specified in fun

Value

col_check(output = FALSE) returns a dataframe containing only records that failed the specified check.

col_check(output = TRUE) invisibly prints an excel output file containing only records that failed the specified check.

Examples

```
three_col_check(colname1 = "Perf_Lvl", colname2 = "Var1", colname3 = "Var2",
  data = dataset, fun = function(col1, col2, col3){
  col1 %in% c("Basic", "Intermediate", "Advanced") |
  (is.na(col1) & (col3 %% 2 ==0) & (col2 %% 2 ==1 ))
})
```

two_col_check	<i>Check a column for data fidelity using criteria related to a second column.</i>
---------------	--

Description

Check a column for data fidelity using criteria related to a second column.

Usage

```
two_col_check(colname1, colname2, data, fun, output = FALSE, stage = NULL,
  ...)
```

Arguments

colname1	character string specifying the name of the column within your dataframe that will be the subject of the checks.
colname2	character string specifying the name of a second column to be used in the check criteria.
data	the dataframe containing the data.
fun	the check function you'd like to apply to the data.
output	logical. If FALSE, the function returns a dataframe containing only records that failed the specified check. If TRUE, invisibly prints an excel output file containing only the records that failed the specified check.
stage	An optional character string that can be used to specify the stage of the checking process in which the check is occurring. Only useful if output = TRUE. If a value is specified, a that value is prefixed to the output file; if no value is given, no stage prefix is attached.
...	arguments to be passed through to the function specified in fun

Value

col_check(output = FALSE) returns a dataframe containing only records that failed the specified check.

col_check(output = TRUE) invisibly prints an excel output file containing only records that failed the specified check.

Examples

```
two_col_check("Var1", "Var2", dataset, less_than_equalto, output = FALSE)
```

```
two_col_check("Var2", "Var1", dataset, greater_than, output = TRUE,
  stage = "1-Reasonableness")
```

val_check	<i>Check whether all values in the column fall within a set of user-defined values.</i>
-----------	---

Description

To be used with [col_check](#).

Usage

```
val_check(x, values)
```

Arguments

x	the data that enters the function (the column specified in col_check)
values	contains a value or vector of values that contain the acceptable value(s) that may be found in the column. These values may be any data type - character strings, numeric values, etc.

Index

*Topic **datasets**

dataset, [4](#)

character_blanks_check, [2](#)

character_check, [2](#)

col_check, [2](#), [3](#), [4–7](#), [10](#)

dataset, [4](#)

date_check, [4](#)

greater_than, [5](#)

greater_than_equalto, [5](#)

inspectr, [6](#)

inspectr-package (inspectr), [6](#)

less_than, [6](#)

less_than_equalto, [7](#)

numeric_check, [7](#)

three_col_check, [8](#)

two_col_check, [5–7](#), [9](#)

val_check, [10](#)