Package ‘mde’
February 29, 2020

Title Missing Data Explorer
Version 0.1.0
Description Correct identification and handling of missing data is one of the most important steps in any analysis. To aid this process, ’mde’ provides a very easy to use yet robust framework to quickly get an idea of where the missing data lies and therefore find the most appropriate action to take.

License GPL-3
Imports purrr(>= 0.3.3), dplyr(>= 0.8.3), magrittr(>= 1.5)
Encoding UTF-8
LazyData true
RoxygenNote 7.0.2
URL https://github.com/Nelson-Gon/mde
BugReports https://github.com/Nelson-Gon/mde/issues
Suggests knitr, rmarkdown, testthat
VignetteBuilder knitr
NeedsCompilation no
Author Nelson Gonzabato [aut, cre]
Maintainer Nelson Gonzabato <gonzabato@hotmail.com>
Repository CRAN
Date/Publication 2020-02-29 13:50:05 UTC

R topics documented:
drop_na_at .......................................................... 2
drop_na_if .......................................................... 2
get_na_counts ...................................................... 3
percent_missing ................................................... 4
recode_as_na ....................................................... 5
recode_as_na_for ................................................ 6
**drop_na_at**

*Drop missing values at columns that match a given pattern*

**Description**

Provides a simple yet efficient way to drop missing values("NA"s) at columns that match a given pattern.

**Usage**

```r
drop_na_at(x, pattern_type = "contains", pattern = NULL, ...)
```

**Arguments**

- `x`: A data.frame object
- `pattern_type`: One of "contains", "ends_with" or "starts_with"
- `pattern`: The type of pattern to use when matching the pattern_type. The pattern is case sensitive
- `...`: Other params to other methods

**Value**

A data.frame object containing only columns that match the given pattern with the missing values removed.

**Examples**

```r
drop_na_at(airquality,pattern_type = "starts_with","O")
```

---

**drop_na_if**

*Condition based dropping of columns with missing values*

**Description**

"drop_na_if" provides a simple way to drop columns with missing values if they meet certain criteria/conditions.

**Usage**

```r
drop_na_if(df, sign = "gteq", percent_na = 50, keep_columns = NULL, ...)
```
**get_na_counts**

**Arguments**
- **df** A data.frame object
- **sign** Character. One of gteq,lteq,lt,gt or eq which refer to greater than(gt) or equal(eq) or less than(lt) or equal to(eq) respectively.
- **percent_na** The percentage to use when dropping columns with missing values
- **keep_columns** Columns that should be kept despite meeting the target percent_na criterion(criteria)
- **...** Other arguments to "percent_missing"

**Value**
A data.frame object with columns that meet the target criteria dropped

**See Also**
- percent_missing

**Examples**
```
drop_na_if(airquality, percent_na = 24)
#drop columns that have less than or equal to 4%
drop_na_if(airquality,sign="lteq", percent_na = 4)
# Drop all except Ozone
drop_na_if(airquality, percent_na = 24, keep_columns = "Ozone")
```

---

**get_na_counts**  
*Add columnwise/groupwise counts of missing values*

**Description**
This function takes a ‘data.frame’ object as an input and returns the corresponding ‘NA’ counts. ‘NA’ refers to R’s built-in missing data holder.

**Usage**
```
get_na_counts(x, grouped = FALSE, grouping_cols = NULL)
```

**Arguments**
- **x** A valid R ‘object’ for which ‘na_counts’ are needed.
- **grouped** Logical. Do you need to add grouped counts? Defaults to ‘FALSE’
- **grouping_cols** If ‘grouped’ is set to ‘TRUE’, one can provide the columns by which to group the data.

**Value**
An object of the same type as ‘x’ showing the respective number of missing values. If grouped is set to ‘TRUE’, the results are returned by group.
Examples

get_na_counts(airquality)
# Grouped counts
test <- data.frame(Subject = c("A", "A", "B", "B"), res = c(NA, 1, 2, 3),
    ID = c("1", "1", "2", "2"))
geat_na_counts(test, grouped = FALSE, grouping_cols =
c("ID", "Subject"))

percent_missing

columnwise missingness percentages

Description

A convenient way to obtain percent missingness columnwise.

Usage

percent_missing(df, grouping_cols = NULL, exclude_cols = NULL)

Arguments

df A valid R ‘object’ for which the percentage of missing values is required.
grouping_cols If ‘grouped’ is set to ‘TRUE’, one can provide the columns by which to group
    the data.
exclude_cols A character vector indicating columns to exclude when returning results.

Value

An object of the same class as x showing the percentage of missing values.

Examples

percent_missing(test, grouping_cols = "ID")
percent_missing(airquality)
percent_missing(airquality, exclude_cols = c("Day", "Temp"))
**recode_as_na**

**Description**

This provides a convenient way to convert a number/value that should indeed be an "NA" to "NA". In otherwords, it converts a value to R’s recognized NA.

**Usage**

```r
recode_as_na(
  df,
  value = NULL,
  subset_df = FALSE,
  tidy = FALSE,
  subset_cols = NULL,
  pattern_type = NULL,
  pattern = NULL,
  ...
)
```

**Arguments**

- `df` A data.frame object for which recoding is to be done.
- `value` The value to convert to ‘NA’. We can for instance change "n/a" to ‘NA’ or any other value.
- `subset_df` Logical. Use only specific columns? Defaults to FALSE. All "value"s everywhere are "recoded".
- `tidy` If set to TRUE, then one can provide additional arguments to match specific patterns using pattern and pattern_type
- `subset_cols` Character. If subset_df is TRUE, then this provides the columns for which changes are required.
- `pattern_type` One of contains, starts_with or ends_with. Used only if subset_cols is set to tidy.
- `pattern` A character pattern to match
- `...` Other arguments to other functions

**Value**

An object of the same class as x with values changed to ‘NA’.
Examples

dummy_test <- data.frame(ID = c("A","B","B","A"),
values = c("n/a",NA,"Yes","No"))
# Replace n/a with "NA". We assume n/a is treated as missing(subjective)
recode_as_na(dummy_test,"n/a")
# Recode only at specific columns
another_dummy <- data.frame(ID = 1:5, Subject = 7:11,
Change = c("missing", "n/a", 2:4 ))
# Change missing and n/a to NA only for the column named Change.
recode_as_na(another_dummy, subset_df = TRUE,
subset_cols = "Change", value = c("n/a",
"missing"))
recode_as_na(airquality, subset_df = TRUE,
tidy=TRUE, pattern_type="starts_with",
pattern="Solar")

recode_as_na_for  Recode Values as NA if they meet defined criteria

Description

Recode Values as NA if they meet defined criteria

Usage

recode_as_na_for(df, criteria = "gt", value = 0, subset_cols = NULL)

Arguments

  df
A data.frame object to manipulate

criteria
One of gt,gteq,lt,lteq to define greater than, greater than or equal to, less than or
less than or equal to.

criteria
The value to convert to ‘NA’. We can for instance change "n/a" to ‘NA’ or any
other value.

Value

A data.frame object with the required changes.

Examples

recode_as_na_for(airquality,value=36, criteria = "gteq",
subset_cols = c("Ozone","Solar.R"))
**recode_na_as**  
*Replace missing values with another value*

**Description**

This provides a convenient way to recode "NA" as another value for instance "NaN", "n/a" or any other value a user wishes to use.

**Usage**

```r
recode_na_as(
  df,
  value = 0,
  subset_df = FALSE,
  tidy = FALSE,
  subset_cols = NULL,
  pattern_type = NULL,
  pattern = NULL,
  ...
)
```

**Arguments**

- `df`: A data.frame object for which recoding is to be done.
- `value`: The value to convert to 'NA'. We can for instance change "n/a" to 'NA' or any other value.
- `subset_df`: Logical. Use only specific columns? Defaults to FALSE. All "value"s everywhere are "recoded".
- `tidy`: If set to TRUE, then one can provide additional arguments to match specific patterns using pattern and pattern_type
- `subset_cols`: Character. If subset_df is TRUE, then this provides the columns for which changes are required.
- `pattern_type`: One of contains, starts_with or ends_with. Used only if subset_cols is set to tidy.
- `pattern`: A character pattern to match
- `...`: Other arguments to other functions

**Value**

An object of the same type as x with NAs replaced with the desired value.
Examples

recode_na_as(airquality, "n/a")
recode_na_as(airquality, subset_df = TRUE, 
subset_cols = "Ozone", value = "N/A")
recode_na_as(airquality, subset_df=TRUE, tidy=TRUE, 
value=0, pattern_type="starts_with", 
pattern="solar",ignore.case=TRUE)

recode_na_if

Recode NA as another value with some conditions

Description

Recode NA as another value with some conditions

Usage

recode_na_if(x, grouping_col = NULL, target_groups = NULL, replacement = 0)

Arguments

x A data.frame object with missing values


grouping_col Character column to use for grouping the data

target_groups Character Recode NA as if and only if the grouping column is in this vector of 
values

replacement Values to use to replace NAs for IDs that meet the requirements. Defaults to 0.

Examples

some_data <- data.frame(ID=c("A1","A2","A3", "A4"),
A=c(5,NA,0,8), B=c(10,0,0,1),C=c(1,NA,NA,25))
# Replace NAs with 0s only for IDs in A2 and A3
recode_na_if(some_data,"ID",c("A2","A3"),replacement=0)

sort_by_missingness

Sort Variables according to missingness

Description

Provides a useful way to sort the variables(columns) according to their missingness.

Usage

sort_by_missingness(x, sort_by = "counts", descending = FALSE, ...)

sort_by_missingness

Arguments

- **x**: A valid R 'object' for which 'na_counts' are needed.
- **sort_by**: One of counts or percents. This determines whether the results are sorted by counts or percentages.
- **descending**: Logical. Should missing values be sorted in decreasing order i.e. largest to smallest? Defaults to FALSE.
- **...**: Other arguments to specific functions. See "See also below"

Value

A 'data.frame' object sorted by number/percentage of missing values

See Also

- get_na_counts
- percent_missing

Examples

```r
sort_by_missingness(airquality, sort_by = "counts")
# sort by percents
sort_by_missingness(airquality, sort_by="percents")
# descending order
sort_by_missingness(airquality, descend = TRUE)
```
Index

drop_na_at, 2
drop_na_if, 2

get_na_counts, 3, 9

percent_missing, 3, 4, 9

recode_as_na, 5
recode_as_na_for, 6
recode_na_as, 7
recode_na_if, 8

sort_by_missingness, 8