Package ‘clickR’

February 29, 2020

Type  Package
Title  Fix Data and Create Report Tables from Different Objects
Version 0.4.47
Author Victoria Fornes Ferrer, David Hervas Marin
Maintainer David Hervas Marin <ddhervas@yahoo.es>
Imports beeswarm, boot, flextable, lme4, lmerTest, methods, officer, xtable
Description Tools for assessing data quality, performing exploratory analysis, fixing data errors in numerical, factor and date variables and creating report tables from models and summaries.
License GPL (>= 2)
Encoding UTF-8
LazyData true
RoxygenNote 7.0.2
NeedsCompilation no
Repository CRAN
Date/Publication 2020-02-29 19:10:02 UTC

R topics documented:

antimoda ................................................. 3
check_quality .......................................... 3
cluster_var ............................................ 4
coefplot ................................................. 5
descriptive ............................................ 6
descriptivo ............................................. 7
extreme_values ........................................ 7
fix.dates ................................................ 8
fix.factors ............................................. 8
fix.levels .............................................. 9
fix.numerics .......................................... 10
R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>forge</td>
<td>10</td>
</tr>
<tr>
<td>fxd</td>
<td>11</td>
</tr>
<tr>
<td>GK_assoc</td>
<td>11</td>
</tr>
<tr>
<td>good2go</td>
<td>12</td>
</tr>
<tr>
<td>ipboxplot</td>
<td>12</td>
</tr>
<tr>
<td>is.it</td>
<td>13</td>
</tr>
<tr>
<td>kill.factors</td>
<td>13</td>
</tr>
<tr>
<td>kurtosis</td>
<td>14</td>
</tr>
<tr>
<td>make_csv_table</td>
<td>14</td>
</tr>
<tr>
<td>make_latex_table</td>
<td>15</td>
</tr>
<tr>
<td>make_table</td>
<td>15</td>
</tr>
<tr>
<td>make_word_table</td>
<td>16</td>
</tr>
<tr>
<td>matrixPaste</td>
<td>16</td>
</tr>
<tr>
<td>may.numeric</td>
<td>17</td>
</tr>
<tr>
<td>mine.plot</td>
<td>17</td>
</tr>
<tr>
<td>moda</td>
<td>18</td>
</tr>
<tr>
<td>moda_cont</td>
<td>19</td>
</tr>
<tr>
<td>mtapply</td>
<td>19</td>
</tr>
<tr>
<td>nearest</td>
<td>20</td>
</tr>
<tr>
<td>nice_names</td>
<td>20</td>
</tr>
<tr>
<td>numeros</td>
<td>21</td>
</tr>
<tr>
<td>peek</td>
<td>21</td>
</tr>
<tr>
<td>plot.reportmodel</td>
<td>22</td>
</tr>
<tr>
<td>prop_may</td>
<td>22</td>
</tr>
<tr>
<td>prop_min</td>
<td>23</td>
</tr>
<tr>
<td>report</td>
<td>23</td>
</tr>
<tr>
<td>report.cutcorr</td>
<td>24</td>
</tr>
<tr>
<td>report.brmsfit</td>
<td>25</td>
</tr>
<tr>
<td>report.clm</td>
<td>25</td>
</tr>
<tr>
<td>report.clmm</td>
<td>26</td>
</tr>
<tr>
<td>report.coxph</td>
<td>27</td>
</tr>
<tr>
<td>report.data.frame</td>
<td>28</td>
</tr>
<tr>
<td>report.default</td>
<td>29</td>
</tr>
<tr>
<td>report.factor</td>
<td>29</td>
</tr>
<tr>
<td>report.glm</td>
<td>30</td>
</tr>
<tr>
<td>report.glmerMod</td>
<td>31</td>
</tr>
<tr>
<td>report.glmmadmb</td>
<td>32</td>
</tr>
<tr>
<td>report.glmmnet</td>
<td>33</td>
</tr>
<tr>
<td>report.lm</td>
<td>34</td>
</tr>
<tr>
<td>report.lmerMod</td>
<td>35</td>
</tr>
<tr>
<td>report.lqmm</td>
<td>36</td>
</tr>
<tr>
<td>report.merModLmerTest</td>
<td>37</td>
</tr>
<tr>
<td>report.numeric</td>
<td>38</td>
</tr>
<tr>
<td>report.rlm</td>
<td>38</td>
</tr>
<tr>
<td>report.rq</td>
<td>39</td>
</tr>
<tr>
<td>rob.ci</td>
<td>40</td>
</tr>
<tr>
<td>rob.pvals</td>
<td>40</td>
</tr>
<tr>
<td>scale_01</td>
<td>41</td>
</tr>
</tbody>
</table>
antimoda

`antimoda`  

Get anti-mode

Description

Returns the least repeated value

Usage

`antimoda(x)`

Arguments

- `x`: A categorical variable

Value

The anti-mode (least repeated value)

---

check_quality

Checks data quality of a variable

Description

Returns different data quality details of a numeric or categorical variable
Usage

check_quality(
  x,
  id = 1:length(x),
  plot = TRUE,
  numeric = NULL,
  n = ifelse(is.numeric(x) | ttrue(numeric) | class(x) %in% "Date", 5, 2),
  output = FALSE,
  ...
)

Arguments

x                A variable from a data.frame
id               ID column to reference the found extreme values
plot             If the variable is numeric, should a boxplot be drawn?
numeric          If set to TRUE, forces the variable to be considered numeric
n                Number of extreme values to extract
output           Format of the output. If TRUE, optimize for exporting as csv
...               further arguments passed to boxplot()

Value

A list of a data.frame with information about data quality of the variable

Examples

check_quality(airquality$Ozone)  #For one variable
lapply(airquality, check_quality)  #For a data.frame
lapply(airquality, check_quality, output=TRUE)  #For a data.frame, one row per variable

---

cluster_var  

Description

Displays associations between variables in a data.frame in a heatmap with clustering

Usage

cluster_var(x, margins = c(8, 1))

Arguments

x                A data.frame
margins          Margins for the plot
Value

A heatmap with the variable associations

Examples

```r
cluster_var(iris)
cluster_var(mtcars)
```

---

**coefplot**  
Plot of the coefficients of a model

Description

Creates a plot of the coefficients of a model

Usage

```r
coefplot(
  coefs,
  lwr.int = coefs,
  upper.int = coefs,
  offset = 0,
  coefnames = names(coefs),
  abline.pos = 0,
  sorted = FALSE,
  reverse = FALSE,
  pch = 16,
  xlim = c(min(lwr.int, na.rm = TRUE), max(upper.int, na.rm = TRUE)),
  ylim = c(1, length(coefs)),
  color = "black",
  ...
)
```

Arguments

- `coefs`: A vector with each coefficient
- `lwr.int`: A vector with the lower end of the CI
- `upper.int`: A vector with the upper end of the CI
- `offset`: Y-axis offset for the coefficients
- `coefnames`: Name for each variable
- `abline.pos`: Position for the vertical reference line
- `sorted`: Should the coefficients be sorted from highest to lowest?
- `reverse`: Should the order be reversed?
- `pch`: Type of point
xlim Limits of the X-axis
ylim Limits of the Y-axis
color Color for the points
... Further arguments passed to axis()

Value
A plot of the coefficients with their CI

Examples

```r
lm1 <- lm(Petal.Length ~ Sepal.Width + Species, data=iris)
a<-report(lm1)
par(mar=c(4, 10, 3, 2))
#Coefplot calling plot.reportmodel
plot(a)
#Manual coefplot
coefplot(coefs=c(1, 2, 3), lwr.int=c(0, 1, 2), upper.int=c(5, 6, 7), coefnames=c("A", "B", "C"))
```

---

descriptive Detailed summary of the data

Description
Creates a detailed summary of the data

Usage

```r
descriptive(x, z = 3, ignore.na = TRUE, by = NULL)
```

Arguments

- `x` A data.frame
- `z` Number of decimal places
- `ignore.na` If TRUE NA values will not count for relative frequencies calculations
- `by` Factor variable defining groups for the summary

Value
Summary of the data

Examples

```r
descriptive(iris)
descriptive(iris, by="Species")
```
**descriptivo**

*Defunct function for creating data summaries*

---

**Description**

Creates a detailed summary of the data

**Usage**

```r
descriptivo(x)
```

**Arguments**

- `x`: A data.frame

**Value**

Nothing, the function is defunct. Use descriptive() instead.

---

**extreme_values**

*Extreme values from a numeric vector*

---

**Description**

Returns the nth lowest and highest values from a vector

**Usage**

```r
extreme_values(x, n = 5, id = NULL)
```

**Arguments**

- `x`: A vector
- `n`: Number of extreme values to return
- `id`: ID column to reference the found extreme values

**Value**

A matrix with the lowest and highest values from a vector
fix.dates

*Fix dates*

**Description**

Fixes dates

**Usage**

```r
fix.dates(
  x,
  max.NA = 0.8,
  min.obs = nrow(x) * 0.05,
  locale = "C",
  info = TRUE,
  use.probs = TRUE
)
```

**Arguments**

- `x`: A data.frame
- `max.NA`: Maximum allowed proportion of NA values created by coercion
- `min.obs`: Minimum number of non-NA observations allowed per variable
- `locale`: Locale to be used for month names
- `info`: Add generated missing values an excluded variable information as attributes
- `use.probs`: Solve ambiguities by similarity to the most frequent formats

**Examples**

```r
                    Dates2=c("01/01/85", "04/04/1982", "07/12-2016", NA),
                    Numeric1=rnorm(4))
fix.dates(mydata)
```

fix.factors

*Fix factors imported as numerics*

**Description**

Fixes factors imported as numerics

**Usage**

```r
fix.factors(x, k = 5, drop = TRUE)
```
**fix.levels**

Arguments

- **x**: A data.frame
- **k**: Maximum number of numeric values to be converted to factor
- **drop**: Drop similar levels?

Examples

```r
report(mtcars)
report(fix.factors(mtcars))
```

---

**fix.levels**  
Fix levels

Description

Fixes levels of a factor

Usage

```r
fix.levels(
  x, 
  levels = NULL, 
  plot = FALSE, 
  k = ifelse(!is.null(levels), length(levels), 2) 
)
```

Arguments

- **x**: A factor vector
- **levels**: Optional vector with the levels names
- **plot**: Optional: Plot cluster dendrogram?
- **k**: Number of levels for clustering

Examples

```r
factor1<-factor(c("Control", "Treatment", "Tretament", "Tratment", "treatment", 
                   "teatment", "contrl", "cntrol", "CONTol", "not available", "na"))
fix.levels(factor1, k=4, plot=TRUE)  #Choose k to select matching levels 
fix.levels(factor1, levels=c("Control", "Treatment"), k=4)
```
fix.numerics

Description

Fixes numeric data

Usage

fix.numerics(x, k = 8, max.NA = 0.2, info = TRUE)

Arguments

x A data.frame
k Minimum number of different values to be considered numerical
max.NA Maximum allowed proportion of NA values created by coercion
info Add generated missing values an excluded variable information as attributes

Examples

mydata<-data.frame(Numeric1=c(7.8, 9.2, 5.4, 3.3, "6,8", "3..3"),
Numeric2=c(3.1, 1.2, "3.s4", "a48,s5", 7, "6,,4"), stringsAsFactors=TRUE)
report(mydata)
report(fix.numerics(mydata, k=5))

forge

Description

Reshapes a data frame from wide to long format

Usage

forge(data, affixes, force.fixed = NULL, var.name = "time")

Arguments

data data.frame
affixes Affixes for repeated measures
force.fixed Variables with matching affix to be excluded
var.name Name for the new created variable (repetitions)
**Examples**

```r
# Data frame in wide format
df1 <- data.frame(id = 1:4, age = c(20, 30, 30, 35), score1 = c(2, 2, 3, 4),
                  score2 = c(2, 1, 3, 1), score3 = c(1, 1, 0, 1))
df1

# Data frame in long format
forge(df1, affixes= c("1", "2", "3"))

# Data frame in wide format with two repeated measured variables
df2 <- data.frame(df1, var1 = c(15, 20, 16, 19), var3 = c(12, 15, 15, 17))
df2

# Missing times are filled with NAs
forge(df2, affixes = c("1", "2", "3"))

# Use of parameter force.fixed
df3 <- df2[, -7]
df3

forge(df3, affixes=c("1", "2", "3"))
forge(df3, affixes=c("1", "2", "3"), force.fixed = c("var1"))
```

---

**fxd**

*Internal function to fix.dates*

**Description**

Function to format dates

**Usage**

```r
fxd(d, locale = "C", use.probs = TRUE)
```

**Arguments**

- `d`: A character vector
- `locale`: Locale to be used for month names
- `use.probs`: Solve ambiguities by similarity to the most frequent formats

---

**GK_assoc**

*Computes Goodman and Kruskal’s tau*

**Description**

Returns Goodman and Kruskal’s tau measure of association between two categorical variables

**Usage**

```r
GK_assoc(x, y)
```
ipboxplot

Arguments

x         A categorical variable
y         A categorical variable

Value

Goodman and Kruskal’s tau

Examples

data(infert)
GK_assoc(infert$education, infert$case)
GK_assoc(infert$case, infert$education) #Not the same

good2go     Good to go

Description

Loads all libraries used in scripts inside the selected path

Usage

good2go(path = getwd(), info = TRUE, load = TRUE)

Arguments

path  Path where the scripts are located
info  List the libraries found?
load  Should the libraries found be loaded?

ipboxplot      Improved boxplot

Description

Creates an improved boxplot with individual data points

Usage

ipboxplot(formula, boxwex = 0.6, ...)
Arguments

- **formula**: Formula for the boxplot
- **boxwex**: Width of the boxes
- **...**: further arguments passed to beeswarm()

Examples

```
ipboxplot(Sepal.Length ~ Species, data=iris)
ipboxplot(mpg ~ gear, data=mtcars)
```

Description

Internal function for mine.plot

Usage

```
is.it(x)
```

Arguments

- **x**: logical expression

kill.factors

Arguments

- **dat**: A data.frame
- **k**: Maximum number of levels for factors

Examples

```
d <- data.frame(Letters=letters[1:20], Nums=1:20)
d$Letters
d <- kill.factors(d)
d$Letters
```
### kurtosis

Computes kurtosis

**Description**

Calculates kurtosis of a numeric variable

**Usage**

`kurtosis(x)`

**Arguments**

- `x` A numeric variable

**Value**

kurtosis value

### make_csv_table

Export a table to excel

**Description**

Exports a table to Excel

**Usage**

`make_csv_table(x, file, info)`

**Arguments**

- `x` A data.frame object
- `file` Name of the file
- `info` Footer for the table

**Value**

Creates a .csv file with the table
**make_latex_table**

*Export a table to latex*

**Description**

Exports a table to latex

**Usage**

```r
make_latex_table(x, file)
```

**Arguments**

- `x`: A data.frame object
- `file`: Name of the file

**Value**

Creates a .txt file with latex code for the table

---

**make_table**

*Make a table from report*

**Description**

Auxiliary function to create tables

**Usage**

```r
make_table(x, file, type, info = NULL, ...)
```

**Arguments**

- `x`: A data.frame object
- `file`: Name of the file
- `type`: Type of file
- `info`: Footer for the table
- `...`: Additional parameters passed to `make_word_table`

**Value**

Creates a file with the table
make_word_table  

Export a table to word

Description

Exports a table to Word

Usage

make_word_table(x, file, info = NULL, use.rownames = TRUE)

Arguments

x  
A data.frame object

file  
Name of the file

info  
Footer for the table

use.rownames  
Should row names be added to the output?

Value

Creates a word file with the table

matrixPaste  

Auxiliary matrix paste function

Description

Internal function for report.table

Usage

matrixPaste(..., sep = rep(" ", length(list(...)) - 1))

Arguments

...  
Matrices to paste

sep  
Separator for the paste function
may.numeric

Checks if each value might be numeric

Usage

may.numeric(x)

Arguments

x A vector

Value

A logical vector

mine.plot

Mine plot

Description

Creates a heatmap-like plot for exploring the data

Usage

mine.plot(
x,
what = "is.na(x)",
spacing = 5,
sort = F,
list = FALSE,
show.x = TRUE,
show.y = TRUE,
...
moda

Description

Returns the most repeated value

Usage

moda(x)

Arguments

x A categorical variable

Value

The mode
moda_cont

**Estimates number of modes**

**Description**
Estimates the number of modes

**Usage**
moda_cont(x)

**Arguments**
x A numeric variable

**Value**
Estimated number of modes. If unclear, marked with an '*'

---

mtapply

**Multiple tapply**

**Description**
Modification of the tapply function to use with data.frames

**Usage**
mtapply(x, group, fun)

**Arguments**
x A data.frame
group Grouping variable
fun Function to apply by group

**Examples**
mtapply(mtcars, mtcars$gear, mean)
nice_names

Description

Changes names of a data frame to ease work with them

Usage

nice_names(dat)

Arguments

dat A data.frame

Examples

d <- data.frame('Variable 1'=NA, ' % Response'=NA, ' Variable 3'=NA,check.names=FALSE)
names(d)
names(nice_names(d))

nearest

Internal function for descriptive()

Description

Finds positions for substitution of characters in Distribution column

Usage

nearest(x, to = seq(0, 1, length.out = 30))

Arguments

x A numeric value between 0-1
to Range of reference values

Value

The nearest position to the input value
**numeros**  

**Brute numeric coercion**

**Description**
If possible, coerces values from a vector to numeric

**Usage**

```r
numeros(x)
```

**Arguments**

- `x` A vector

**Value**

A numeric vector

---

**peek**  

**Peek**

**Description**
Takes a peek into a data.frame returning a concise visualization about it

**Usage**

```r
peek(x, n = 10, which = 1:ncol(x))
```

**Arguments**

- `x` A data.frame
- `n` Number of rows to include in output
- `which` Columns to include in output

**Examples**

```r
peek(iris)
```
plot.reportmodel  

Coefplot for reportmodel objects

Description

Creates a coefplot from the reportmodel object

Usage

## S3 method for class 'reportmodel'
plot(x, ...)

Arguments

x  
A reportmodel object

...  
Further arguments passed to coefplot

Examples

lm1 <- lm(Petal.Length ~ Sepal.Width + Species, data=iris)
a<-report(lm1)
par(mar=c(4, 10, 3, 2))
plot(a)  #Coefplot calling plot.reportmodel

prop_may  

Gets proportion of most repeated value

Description

Returns the proportion for the most repeated value

Usage

prop_may(x, ignore.na = TRUE)

Arguments

x  
A categorical variable

ignore.na  
Should NA values be ignored for computing proportions?

Value

A proportion
prop_min

Gets proportion of least repeated value

Description

Returns the proportion for the least repeated value

Usage

\[
\text{prop_min}(x, \text{ignore.na} = \text{TRUE})
\]

Arguments

- \(x\): A categorical variable
- \text{ignore.na}: Should NA values be ignored for computing proportions?

Value

A proportion

report

Generic function for reporting of models

Description

Generic function for reporting of models

Usage

\[
\text{report}(x, \ldots)
\]

Arguments

- \(x\): A model object
- \(\ldots\): further arguments passed to make_table

Value

A data frame with the report table

Examples

\[
\text{report}(\text{iris}) \quad \# \text{Report of descriptive statistics}
\]

\[
\text{lm1 <- lm(Petal.Length ~ Sepal.Width + Species, data=iris)}
\]

\[
\text{report(lm1)} \quad \# \text{Report of model}
\]
Description

Creates a report table from a beta regression model

Usage

```r
## S3 method for class 'betareg'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```

Arguments

- `x`: A betareg model object
- `file`: Name of the file to export the table
- `type`: Format of the file
- `digits`: Number of decimals
- `digitspvals`: Number of decimals for p-values
- `info`: If TRUE, include call in the exported table
- `print`: Should the report table be printed on screen?
- `...`: Further arguments passed to `make_table`

Value

A data frame with the report table
**report.brmsfit**

Report models from brms package

---

**Description**

Creates a report table from model fitted by brms

**Usage**

```r
## S3 method for class 'brmsfit'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  info = TRUE,
  print = TRUE,
  ...)
```

**Arguments**

- `x`: A brms model object
- `file`: Name of the file to export the table
- `type`: Format of the file
- `digits`: Number of decimals
- `info`: If TRUE, include call in the exported table
- `print`: Should the report table be printed on screen?
- `...`: Further arguments passed to make_table

**Value**

A data frame with the report table

---

**report.clm**

Report from ordinal model

---

**Description**

Creates a report table from an ordinal model
Usage

## S3 method for class 'clm'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)

Arguments

x: An ordinal model object
file: Name of the file to export the table
type: Format of the file
digits: Number of decimals
digitspvals: Number of decimals for p-values
info: If TRUE, include call in the exported table
print: Should the report table be printed on screen?
...
Further arguments passed to make_table

Value

A data frame with the report table

---

report.clmm: Report from ordinal mixed model

Description

Creates a report table from an ordinal mixed model

Usage

## S3 method for class 'clmm'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
repoort.coxph

```r
  print = TRUE,
  ...
)
```

**Arguments**

- `x`: An ordinal model object
- `file`: Name of the file to export the table
- `type`: Format of the file
- `digits`: Number of decimals
- `digitspvals`: Number of decimals for p-values
- `info`: If TRUE, include call in the exported table
- `print`: Should the report table be printed on screen?
- `...`: Further arguments passed to `make_table`

**Value**

A data frame with the report table

---

**Description**

Creates a report table from a cox model

**Usage**

```r
## S3 method for class 'coxph'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```
report.data.frame

Description

Creates a report table ready for publication

Usage

```r
# S3 method for class 'data.frame'
report(
  x,
  by = NULL,
  file = NULL,
  type = "word",
  digits = 2,
  digitscat = digits,
  print = TRUE,
  ...
)
```

Arguments

- **x**: A data.frame object
- **by**: Grouping variable for the report
- **file**: Name of the file to export the table
- **type**: Format of the file
- **digits**: Number of decimal places
- **digitscat**: Number of decimal places for categorical variables (if different to digits)
- **print**: Should the report table be printed on screen?
- **...**: Further arguments passed to make_table()
Examples

```r
report(iris)
(reporTable<-report(iris, by="Species"))
class(reporTable)
```

---

**report.default**  
*Default function for report*

**Description**  
This is a default function for calling `summary(x)` on non-implemented classes

**Usage**  
```r
## Default S3 method:
report(x, ...)
```

**Arguments**  
- `x`  
  Any object without specific report function  
- `...`  
  Further arguments passed to `summary`

**Value**  
A summary of the object

---

**report.factor**  
*Report from categorical variable*

**Description**  
Creates a report table

**Usage**  
```r
## S3 method for class 'factor'
report(x, ...)
```

**Arguments**  
- `x`  
  A categorical variable  
- `...`  
  Further arguments passed to `make_table`

**Value**  
A data frame with the report table
**Description**

Creates a report table from a generalized linear model

**Usage**

```r
## S3 method for class 'glm'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```

**Arguments**

- **x**: A generalized linear model object
- **file**: Name of the file to export the table
- **type**: Format of the file
- **digits**: Number of decimals
- **digitspvals**: Number of decimals for p-values
- **info**: If TRUE, include call in the exported table
- **print**: Should the report table be printed on screen?
- **...**: Further arguments passed to make_table

**Value**

A data frame with the report table
report.glmerMod

Report from generalized linear mixed model

Description

Creates a report table from a generalized linear mixed model

Usage

```r
## S3 method for class 'glmerMod'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```

Arguments

- `x`: A generalized linear mixed model object
- `file`: Name of the file to export the table
- `type`: Format of the file
- `digits`: Number of decimals
- `digitspvals`: Number of decimals for p-values
- `info`: If TRUE, include call in the exported table
- `print`: Should the report table be printed on screen?
- `...`: Further arguments passed to `make_table`

Value

A data frame with the report table
Description

Creates a report table from a glmmadmb model.

Usage

```r
## S3 method for class 'glmmadmb'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```

Arguments

- `x`: A generalized linear mixed model object (glmmadmb)
- `file`: Name of the file to export the table
- `type`: Format of the file
- `digits`: Number of decimals
- `digitspvals`: Number of decimals for p-values
- `info`: If TRUE, include call in the exported table
- `print`: Should the report table be printed on screen?
- `...`: Further arguments passed to make_table

Value

A data frame with the report table
Report models from glmnet package

Description

Creates a report table from models fitted by glmnet

Usage

```r
## S3 method for class 'glmnet'
report(
  x,
  s,
  gamma = 1,
  drop.zero = TRUE,
  file = NULL,
  type = "word",
  digits = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```

Arguments

- `x`: A glmnet model object
- `s`: Value of lambda for estimating the coefficients
- `gamma`: Value of gamma for estimating the coefficients (only used in relaxed fits)
- `drop.zero`: Should zero coefficients be dropped?
- `file`: Name of the file to export the table
- `type`: Format of the file
- `digits`: Number of decimals
- `info`: If TRUE, include call in the exported table
- `print`: Should the report table be printed on screen?
- `...`: Further arguments passed to make_table

Value

A data frame with the report table
Description

Creates a report table from a linear model

Usage

```r
## S3 method for class 'lm'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```

Arguments

- `x` A linear model object
- `file` Name of the file to export the table
- `type` Format of the file
- `digits` Number of decimals
- `digitspvals` Number of decimals for p-values
- `info` If TRUE, include call in the exported table
- `print` Should the report table be printed on screen?
- `...` Further arguments passed to `make_table`

Value

A data frame with the report table
Report from linear mixed model

Description

Creates a report table from a linear mixed model

Usage

```r
## S3 method for class 'lmerMod'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitpvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```

Arguments

- `x`: A linear mixed model object
- `file`: Name of the file to export the table
- `type`: Format of the file
- `digits`: Number of decimals
- `digitpvals`: Number of decimals for p-values
- `info`: If TRUE, include call in the exported table
- `print`: Should the report table be printed on screen?
- `...`: Further arguments passed to make_table

Value

A data frame with the report table
**Description**

Creates a report table from a quantile mixed model

**Usage**

```r
## S3 method for class 'lqmm'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```

**Arguments**

- `x`: A quantile model object
- `file`: Name of the file to export the table
- `type`: Format of the file
- `digits`: Number of decimals
- `digitspvals`: Number of decimals for p-values
- `info`: If TRUE, include call in the exported table
- `print`: Should the report table be printed on screen?
- `...`: Further arguments passed to `make_table`

**Value**

A data frame with the report table
Report from linear mixed model with pvalues

Description

Creates a report table from a linear mixed model

Usage

```r
## S3 method for class 'merModLmerTest'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```

Arguments

- `x`: A linear mixed model object
- `file`: Name of the file to export the table
- `type`: Format of the file
- `digits`: Number of decimals
- `digitspvals`: Number of decimals for p-values
- `info`: If TRUE, include call in the exported table
- `print`: Should the report table be printed on screen?
- `...`: Further arguments passed to `make_table`

Value

A data frame with the report table
report.numeric  Report from numeric variable

Description

Creates a report table

Usage

## S3 method for class 'numeric'
report(x, ...)

Arguments

x              A numeric variable
...            Further arguments passed to make_table

Value

A data frame with the report table

report.rlm  Report from robust linear model (rlm)

Description

Creates a report table from a robust linear model

Usage

## S3 method for class 'rlm'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
Argument

- **x**: A rlm object
- **file**: Name of the file to export the table
- **type**: Format of the file
- **digits**: Number of decimals
- **digitspvals**: Number of decimals for p-values
- **info**: If TRUE, include call in the exported table
- **print**: Should the report table be printed on screen?
- **...**: Further arguments passed to make_table

Value

A data frame with the report table

---

Description

Creates a report table from a quantile regression model

Usage

```r
## S3 method for class 'rq'
report(
  x,
  file = NULL,
  type = "word",
  digits = 3,
  digitspvals = 3,
  info = TRUE,
  print = TRUE,
  ...
)
```

Arguments

- **x**: A quantreg model object
- **file**: Name of the file to export the table
- **type**: Format of the file
- **digits**: Number of decimals
- **digitspvals**: Number of decimals for p-values
- **info**: If TRUE, include call in the exported table
- **print**: Should the report table be printed on screen?
- **...**: Further arguments passed to make_table
rob.ci  
A data frame with the report table

Function to compute bootstrap confidence intervals for robust linear regression models

Description
Estimates confidence intervals for rlm models

Usage
rob.ci(x, level = 0.95, maxit = 200, R = 2000)

Arguments
x  A rlm object  
level  Confidence level for the interval  
maxit  Maximum number of iterations per fit  
R  Number of bootstrap samples

Value
A matrix with bootstrap confidence intervals for each variable in the model

rob.pvals  
Function to compute p-values for robust linear regression models

Description
Estimates p-values for rlm models

Usage
rob.pvals(x)

Arguments
x  A rlm object

Value
A vector of p-values
**scale_01**

Scales data between 0 and 1

**Description**

Escales data to 0-1

**Usage**

\[ \text{scale}_01(x) \]

**Arguments**

\[ x \]

A numeric variable

**Value**

Scaled data

---

**search_scripts**

Search scripts

**Description**

Searches for strings in R script files

**Usage**

\[ \text{search}_scripts(\text{string}, \text{path} = \text{getwd()}, \text{recursive} = \text{TRUE}) \]

**Arguments**

\[ \text{string} \]

Character string to search

\[ \text{path} \]

Character vector with the path name

\[ \text{recursive} \]

Logical. Should the search be recursive into subdirectories?

**Value**

A list with each element being one of the files containing the search string
set_noms  Set header names for word tables

Description
Internal function for make_word_table

Usage
set_noms(x, args)

Arguments
- x: A flextable object
- args: A names list with the header names

Value
A flextable object with assigned header names

skewness  Computes skewness

Description
Calculates skewness of a numeric variable

Usage
skewness(x)

Arguments
- x: A numeric variable

Value
skewness value
ttrue

Description

Makes possible logical comparisons against NULL and NA values

Usage

ttrue(x)

Arguments

x  A logical vector

Value

A logical vector

def

Description

Reshapes a data frame from long to wide format

Usage

unforge(data, origin, variables, prefix = origin)

Arguments

data  data.frame

origin  Character vector with variable names in data containing the values to be assigned to the different new variables

variables  Variable in data containing the variable names to be created

prefix  Vector with prefixes for the new variable names
Examples

```r
# Data frame in wide format
df1 <- data.frame(id = 1:4, age = c(20, 30, 30, 35), score1 = c(2,2,3,4),
                  score2 = c(2,1,3,1), score3 = c(1,1,0,1))
df1
# Data frame in long format
df2 <- forge(df1, affixes= c("1", "2", "3"))
df2
# Data frame in wide format again
df3 <- unforge(df2, "score", "time", prefix="score")
```

---

VarCorr

**Generic VarCorr function**

**Description**

Extract Variance-Covariance Matrix

**Usage**

```
VarCorr(x, sigma = 1, ...)
```

**Arguments**

- `x`: A model object
- `sigma`: Optional value used as a multiplier for the standard deviations
- `...`: Further arguments passed to `VarCorr` methods

**Value**

A Variance-Covariance Matrix

---

workspace

**Explores global environment workspace**

**Description**

Returns information regarding the different objects in global environment

**Usage**

```
workspace(table = FALSE)
```
workspace_sapply

Arguments

- table: If TRUE a table with the frequencies of each type of object is given

Value

A list of object names by class or a table with frequencies if table = TRUE

Examples

df1 <- data.frame(x=rnorm(10), y=rnorm(10, 1, 2))
df2 <- data.frame(x=rnorm(20), y=rnorm(20, 1, 2))
workspace(table=TRUE)  #Frequency table of the different object classes
workspace()  #All objects in the global object separated by class

workspace_sapply

Applies a function over objects of a specific class

Description

Applies a function over all objects of a specific class in the global environment

Usage

workspace_sapply(object_class, action = "summary")

Arguments

- object_class: Class of the objects where the function is to be applied
- action: Name of the function to apply

Value

Results of the function

Examples

df1 <- data.frame(x=rnorm(10), y=rnorm(10, 1, 2))
df2 <- data.frame(x=rnorm(20), y=rnorm(20, 1, 2))
workspace_sapply("data.frame", "summary")  #Gives a summary of each data.frame
\%>\text{NA}\% \quad \text{greater} \ & \text{NA}

\textbf{Description}

‘\textgreater’ operator where NA values return FALSE

\textbf{Usage}

\texttt{x \%>\text{NA}\% y}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \quad \text{Vector for the left side of the operator}
  \item \texttt{y} \quad \text{A Scalar or vector of the same length as \texttt{x} for the right side of the operator}
\end{itemize}

\textbf{Value}

A logical vector of the same length as \texttt{x}

\\hline
\%>=\text{NA}\%
\quad \text{geq} \ & \text{not NA}
\hline

\textbf{Description}

‘\textgreater=’ operator where NA values return FALSE

\textbf{Usage}

\texttt{x \%>=\text{NA}\% y}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{x} \quad \text{Vector for the left side of the operator}
  \item \texttt{y} \quad \text{A Scalar or vector of the same length as \texttt{x} for the right side of the operator}
\end{itemize}

\textbf{Value}

A logical vector of the same length as \texttt{x}
%<NA%  

Description

'<' operator where NA values return FALSE

Usage

x %<NA% y

Arguments

x  Vector for the left side of the operator
y  A Scalar or vector of the same length as x for the right side of the operator

Value

A logical vector of the same length as x

%<=NA%  

Description

'<=' operator where NA values return FALSE

Usage

x %<=NA% y

Arguments

x  Vector for the left side of the operator
y  A Scalar or vector of the same length as x for the right side of the operator

Value

A logical vector of the same length as x
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>%&lt;=NA%</td>
<td>47</td>
</tr>
<tr>
<td>%&lt;NA%</td>
<td>47</td>
</tr>
<tr>
<td>%&gt;=NA%</td>
<td>46</td>
</tr>
<tr>
<td>%&gt;NA%</td>
<td>46</td>
</tr>
<tr>
<td>antimoda</td>
<td>3</td>
</tr>
<tr>
<td>check_quality</td>
<td>3</td>
</tr>
<tr>
<td>cluster_var</td>
<td>4</td>
</tr>
<tr>
<td>coefplot</td>
<td>5</td>
</tr>
<tr>
<td>descriptive</td>
<td>6</td>
</tr>
<tr>
<td>descriptivo</td>
<td>7</td>
</tr>
<tr>
<td>extreme_values</td>
<td>7</td>
</tr>
<tr>
<td>fix.dates</td>
<td>8</td>
</tr>
<tr>
<td>fix.factors</td>
<td>8</td>
</tr>
<tr>
<td>fix.levels</td>
<td>9</td>
</tr>
<tr>
<td>fix.numerics</td>
<td>10</td>
</tr>
<tr>
<td>forge</td>
<td>10</td>
</tr>
<tr>
<td>fxd</td>
<td>11</td>
</tr>
<tr>
<td>GK_assoc</td>
<td>11</td>
</tr>
<tr>
<td>good2go</td>
<td>12</td>
</tr>
<tr>
<td>ipboxplot</td>
<td>12</td>
</tr>
<tr>
<td>is.it</td>
<td>13</td>
</tr>
<tr>
<td>kill.factors</td>
<td>13</td>
</tr>
<tr>
<td>kurtosis</td>
<td>14</td>
</tr>
<tr>
<td>make_csv_table</td>
<td>14</td>
</tr>
<tr>
<td>make_latex_table</td>
<td>15</td>
</tr>
<tr>
<td>make_table</td>
<td>15</td>
</tr>
<tr>
<td>make_word_table</td>
<td>16</td>
</tr>
<tr>
<td>matrixPaste</td>
<td>16</td>
</tr>
<tr>
<td>may.numeric</td>
<td>17</td>
</tr>
<tr>
<td>mine.plot</td>
<td>17</td>
</tr>
<tr>
<td>moda</td>
<td>18</td>
</tr>
<tr>
<td>moda_cont</td>
<td>19</td>
</tr>
<tr>
<td>mtapply</td>
<td>19</td>
</tr>
<tr>
<td>nearest</td>
<td>20</td>
</tr>
<tr>
<td>nice_names</td>
<td>20</td>
</tr>
<tr>
<td>numeros</td>
<td>21</td>
</tr>
<tr>
<td>peek</td>
<td>21</td>
</tr>
<tr>
<td>plot.reportmodel</td>
<td>22</td>
</tr>
<tr>
<td>prop_may</td>
<td>22</td>
</tr>
<tr>
<td>prop_min</td>
<td>23</td>
</tr>
<tr>
<td>report</td>
<td>23</td>
</tr>
<tr>
<td>report.betareg</td>
<td>24</td>
</tr>
<tr>
<td>report.brmsfit</td>
<td>25</td>
</tr>
<tr>
<td>report.clm</td>
<td>25</td>
</tr>
<tr>
<td>report.clmm</td>
<td>26</td>
</tr>
<tr>
<td>report.coxph</td>
<td>27</td>
</tr>
<tr>
<td>report.data.frame</td>
<td>28</td>
</tr>
<tr>
<td>report.default</td>
<td>29</td>
</tr>
<tr>
<td>report.factor</td>
<td>29</td>
</tr>
<tr>
<td>report.glm</td>
<td>30</td>
</tr>
<tr>
<td>report.glmerMod</td>
<td>31</td>
</tr>
<tr>
<td>report.glmmadmb</td>
<td>32</td>
</tr>
<tr>
<td>report.glmnet</td>
<td>33</td>
</tr>
<tr>
<td>report.lm</td>
<td>34</td>
</tr>
<tr>
<td>report.lmerMod</td>
<td>35</td>
</tr>
<tr>
<td>report.lqmm</td>
<td>36</td>
</tr>
<tr>
<td>report.merModLmerTest</td>
<td>37</td>
</tr>
<tr>
<td>report.numeric</td>
<td>38</td>
</tr>
<tr>
<td>report.rlm</td>
<td>38</td>
</tr>
<tr>
<td>report.rq</td>
<td>39</td>
</tr>
<tr>
<td>robust.ci</td>
<td>40</td>
</tr>
<tr>
<td>robust.pvals</td>
<td>40</td>
</tr>
<tr>
<td>scale_01</td>
<td>41</td>
</tr>
<tr>
<td>search_scripts</td>
<td>41</td>
</tr>
<tr>
<td>set_noms</td>
<td>42</td>
</tr>
<tr>
<td>skewness</td>
<td>42</td>
</tr>
<tr>
<td>ttrue</td>
<td>43</td>
</tr>
</tbody>
</table>
INDEX

unforge, 43
VarCorr, 44
workspace, 44
workspace_sapply, 45