

# Package ‘clickR’

December 7, 2020

**Type** Package

**Title** Fix Data and Create Report Tables from Different Objects

**Version** 0.5.27

**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** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-12-07 16:00:02 UTC

## R topics documented:

antimoda . . . . .	3
check_quality . . . . .	4
cluster_var . . . . .	5
coefplot . . . . .	5
descriptive . . . . .	6
extreme_values . . . . .	7
fix.dates . . . . .	8
fix.factors . . . . .	8
fix.levels . . . . .	9
fix.NA . . . . .	10
fix.numerics . . . . .	10

forge . . . . .	11
fxd . . . . .	12
GK_assoc . . . . .	12
good2go . . . . .	13
ipboxplot . . . . .	13
is.it . . . . .	14
kill.factors . . . . .	14
kurtosis . . . . .	15
make_csv_table . . . . .	15
make_latex_table . . . . .	16
make_table . . . . .	16
make_word_table . . . . .	17
matrixPaste . . . . .	17
may.numeric . . . . .	18
mine.plot . . . . .	18
moda . . . . .	19
moda_cont . . . . .	20
mtapply . . . . .	20
nearest . . . . .	21
nice_names . . . . .	21
numeros . . . . .	22
peek . . . . .	22
plot.reportmodel . . . . .	23
prop_may . . . . .	23
prop_min . . . . .	24
report . . . . .	24
report.betareg . . . . .	25
report.brmsfit . . . . .	26
report.clm . . . . .	26
report.clmm . . . . .	27
report.coxph . . . . .	28
report.data.frame . . . . .	29
report.default . . . . .	30
report.factor . . . . .	30
report.glm . . . . .	31
report.glmerMod . . . . .	32
report.glmmbadm . . . . .	33
report.glmnet . . . . .	34
report.lm . . . . .	35
report.lmerMod . . . . .	36
report.lqmm . . . . .	37
report.merModLmerTest . . . . .	38
report.numeric . . . . .	39
report.rlm . . . . .	39
report.rq . . . . .	40
restore_changes . . . . .	41
rob.ci . . . . .	41
rob.pvals . . . . .	42

<i>antimoda</i>	3
scale_01 . . . . .	42
search_scripts . . . . .	43
set_noms . . . . .	43
skewness . . . . .	44
track_changes . . . . .	44
ttrue . . . . .	45
unforge . . . . .	45
VarCorr . . . . .	46
workspace . . . . .	46
workspace_sapply . . . . .	47
%>NA% . . . . .	48
%>=NA% . . . . .	48
%<NA% . . . . .	49
%<=NA% . . . . .	49
<b>Index</b>	<b>50</b>

---

<i>antimoda</i>	<i>Get anti-mode</i>
-----------------	----------------------

---

### Description

Returns the least repeated value

### Usage

```
antimoda(x)
```

### Arguments

x                    A categorical variable

### Value

The anti-mode (least repeated value)

---

check_quality	<i>Checks data quality of a variable</i>
---------------	--

---

## 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	<i>Clustering of variables</i>
-------------	--------------------------------

---

**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**

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

---

coefplot	<i>Plot of the coefficients of a model</i>
----------	--

---

**Description**

Creates a plot of the coefficients of a model

**Usage**

```
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

```

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

```
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**

```
descriptive(iris)
descriptive(iris, by="Species")
```

---

extreme_values	<i>Extreme values from a numeric vector</i>
----------------	---

---

**Description**

Returns the nth lowest and highest values from a vector

**Usage**

```
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	<i>Fix dates</i>
-----------	------------------

---

**Description**

Fixes dates

**Usage**

```
fix.dates(
  x,
  max.NA = 0.8,
  min.obs = nrow(x) * 0.05,
  locale = "C",
  use.probs = TRUE,
  track = 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
use.probs	Solve ambiguities by similarity to the most frequent formats
track	Track changes?

**Examples**

```
mydata<-data.frame(Dates1=c("25/06/1983", "25-08/2014", "2001/11/01", "2008-10-01"),
  Dates2=c("01/01/85", "04/04/1982", "07/12-2016", NA),
  Numeric1=rnorm(4))
fix.dates(mydata)
```

---

fix.factors	<i>Fix factors imported as numerics</i>
-------------	---

---

**Description**

Fixes factors imported as numerics

**Usage**

```
fix.factors(x, k = 5, drop = TRUE, track = TRUE)
```



**Arguments**

x	A data.frame
k	Maximum number of numeric values to be converted to factor
drop	Drop similar levels?
track	Keep track of changes?

**Examples**

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

---

fix.levels	<i>Fix levels</i>
------------	-------------------

---

**Description**

Fixes levels of a factor

**Usage**

```
fix.levels(
  data,
  factor_name,
  levels = NULL,
  plot = FALSE,
  k = ifelse(!is.null(levels), length(levels), 2),
  track = TRUE
)
```

**Arguments**

data	data.frame with the factor to fix
factor_name	Name of the factor to fix (as character)
levels	Optional vector with the levels names
plot	Optional: Plot cluster dendrogram?
k	Number of levels for clustering
track	Keep track of changes?

**Examples**

```
mydata <- data.frame(factor1=factor(c("Control", "Treatment", "Tretament", "Tratment", "treatment",
  "teatment", "contr1", "cntrol", "CONTol", "not available", "na")))
fix.levels(mydata, "factor1", k=4, plot=TRUE) #Chose k to select matching levels
fix.levels(mydata, "factor1", levels=c("Control", "Treatment"), k=4)
```

---

fix.NA	<i>fix.NA</i>
--------	---------------

---

**Description**

Fixes miscoded missing values

**Usage**

```
fix.NA(
  x,
  na.strings = c("^$", "^ $", "^\\?$", "^-$", "^\\.?$", "^NaN$", "^NULL$", "^N/A$"),
  track = TRUE
)
```

**Arguments**

x	A data.frame
na.strings	Strings to be considered NA
track	Track changes?

**Examples**

```
mydata <- data.frame(prueba = c("", NA, "A", 4, " ", "?", "-", "+"),
  casa = c("", 1, 2, 3, 4, " ", 6, 7))
fix.NA(mydata)
```

---

fix.numerics	<i>Fix numeric data</i>
--------------	-------------------------

---

**Description**

Fixes numeric data

**Usage**

```
fix.numerics(x, k = 8, max.NA = 0.2, track = 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
track	Keep track of changes?

**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

*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**

```
#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	<i>Internal function to fix.dates</i>
-----	---------------------------------------

---

**Description**

Function to format dates

**Usage**

```
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	<i>Computes Goodman and Kruskal's tau</i>
----------	---

---

**Description**

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

**Usage**

```
GK_assoc(x, y)
```

**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**

<code>path</code>	Path where the scripts are located
<code>info</code>	List the libraries found?
<code>load</code>	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**

<code>formula</code>	Formula for the boxplot
<code>boxwex</code>	Width of the boxes
<code>...</code>	further arguments passed to <code>beeswarm()</code>

**Examples**

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

`is.it`*is.it*

---

**Description**

Internal function for mine.plot

**Usage**

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

**Arguments**

x                    logical expression

---

`kill.factors`*Kill factors*

---

**Description**

Changes factor variables to character

**Usage**

```
kill.factors(dat, k = 10)
```

**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	<i>Computes kurtosis</i>
----------	--------------------------

---

**Description**

Calculates kurtosis of a numeric variable

**Usage**

```
kurtosis(x)
```

**Arguments**

x	A numeric variable
---	--------------------

**Value**

kurtosis value

---

make_csv_table	<i>Export a table to excel</i>
----------------	--------------------------------

---

**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	<i>Export a table to latex</i>
------------------	--------------------------------

---

**Description**

Exports a table to latex

**Usage**

```
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	<i>Make a table from report</i>
------------	---------------------------------

---

**Description**

Auxiliary function to create tables

**Usage**

```
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	<i>Export a table to word</i>
-----------------	-------------------------------

---

**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	<i>Auxiliary matrix paste function</i>
-------------	--

---

**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	<i>Checks if each value might be numeric</i>
-------------	--

---

**Description**

Checks if each value from a vector might be numeric

**Usage**

```
may.numeric(x)
```

**Arguments**

x                    A vector

**Value**

A logical vector

---

mine.plot	<i>Mine plot</i>
-----------	------------------

---

**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,  
  ...  
)
```

**Arguments**

x	A data.frame
what	A logical expression that will be depicted in the plot
spacing	Numerical separation between lines at the y-axis
sort	If TRUE, variables are sorted according to their results
list	If TRUE, creates a vector with the results
show.x	Should the x-axis be plotted?
show.y	Should the y-axis be plotted?
...	further arguments passed to order()

**Examples**

```
mine.plot(airquality) #Displays missing data
mine.plot(airquality, what="x>mean(x)+2*sd(x) | x<mean(x)-2*sd(x)") #Shows extreme values
```

---

moda

*Get mode*

---

**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.

---

mtapply                      *Multiple tapply*

---

**Description**

Modification of the tapply function to use with data.frames. Consider using aggregate()

**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)
```

---

nearest	<i>Internal function for descriptive()</i>
---------	--

---

**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

---

nice_names	<i>Nice names</i>
------------	-------------------

---

**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))
```

---

numeros	<i>Brute numeric coercion</i>
---------	-------------------------------

---

**Description**

If possible, coerces values from a vector to numeric

**Usage**

```
numeros(x)
```

**Arguments**

x	A vector
---	----------

**Value**

A numeric vector

---

peek	<i>Peek</i>
------	-------------

---

**Description**

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

**Usage**

```
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**

```
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	<i>Gets proportion of least repeated value</i>
----------	--

---

**Description**

Returns the proportion for the least repeated value

**Usage**

```
prop_min(x, ignore.na = TRUE)
```

**Arguments**

x	A categorical variable
ignore.na	Should NA values be ignored for computing proportions?

**Value**

A proportion

---

report	<i>Generic function for reporting of models</i>
--------	---

---

**Description**

Generic function for reporting of models

**Usage**

```
report(x, ...)
```

**Arguments**

x	A model object
...	further arguments passed to make_table

**Value**

A data frame with the report table

**Examples**

```
report(iris) #Report of descriptive statistics  
lm1 <- lm(Petal.Length ~ Sepal.Width + Species, data=iris)  
report(lm1) #Report of model
```



---

report.betareg	<i>Report from beta regression model</i>
----------------	--

---

## Description

Creates a report table from a beta regression model

## Usage

```
## 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	<i>Report models from brms package</i>
----------------	--

---

### Description

Creates a report table from model fitted by brms

### Usage

```
## 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	<i>Report from ordinal model</i>
------------	----------------------------------

---

### 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	<i>Report from ordinal mixed model</i>
-------------	--

---

**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,
```

```

    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.coxph	<i>Report from cox regression model</i>
--------------	---

---

### Description

Creates a report table from a cox model

### Usage

```

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

```

**Arguments**

x	A cox 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.data.frame	<i>Report tables of summary data</i>
-------------------	--------------------------------------

---

**Description**

Creates a report table ready for publication

**Usage**

```
## 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**

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

---

report.default	<i>Default function for report</i>
----------------	------------------------------------

---

**Description**

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

**Usage**

```
## 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	<i>Report from categorical variable</i>
---------------	---

---

**Description**

Creates a report table

**Usage**

```
## 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

---

`report.glm`*Report from generalized linear model*

---

## Description

Creates a report table from a generalized linear model

## Usage

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

## Arguments

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

## Value

A data frame with the report table

---

report.glmerMod	<i>Report from generalized linear mixed model</i>
-----------------	---

---

## Description

Creates a report table from a generalized linear mixed model

## Usage

```
## 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



---

report.glmmb	<i>Report from generalized linear mixed model from ADMB</i>
--------------	---

---

## Description

Creates a report table from a glmmb model

## Usage

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

## Arguments

x	A generalized linear mixed model object (glmmb)
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.glmnet`*Report models from glmnet package*

---

**Description**

Creates a report table from models fitted by glmnet

**Usage**

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

**Arguments**

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

**Value**

A data frame with the report table

---

`report.lm`*Report from linear model*

---

## Description

Creates a report table from a linear model

## Usage

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

## Arguments

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

## Value

A data frame with the report table

---

report.lmerMod	<i>Report from linear mixed model</i>
----------------	---------------------------------------

---

## Description

Creates a report table from a linear mixed model

## Usage

```
## S3 method for class 'lmerMod'
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.lqmm	<i>Report from quantile mixed model</i>
-------------	---

---

## Description

Creates a report table from a quantile mixed model

## Usage

```
## 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.merModLmerTest *Report from linear mixed model with pvalues*

---

## Description

Creates a report table from a linear mixed model

## Usage

```
## 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	<i>Report from numeric variable</i>
----------------	-------------------------------------

---

**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	<i>Report from robust linear model (rlm)</i>
------------	--

---

**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,  
  ...  
)
```

**Arguments**

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

---

report.rq	<i>Report from quantile regression model</i>
-----------	--

---

**Description**

Creates a report table from a quantile regression model

**Usage**

```
## 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



**Value**

A data frame with the report table

---

restore_changes	<i>Restore changes</i>
-----------------	------------------------

---

**Description**

Restores original values after using a fix function

**Usage**

```
restore_changes(x, var.names)
```

**Arguments**

x	A data.frame
var.names	Character vector with names of the variables to be restored

**Examples**

```
mydata<-data.frame(Dates1=c("25/06/1983", "25-08/2014", "2001/11/01", "2008-10-01"),
                  Dates2=c("01/01/85", "04/04/1982", "07/12-2016", NA),
                  Numeric1=rnorm(4))
mydata <- fix.dates(mydata)
mydata
mydata <- restore_changes(mydata, "Dates1")
mydata
```

---

rob.ci	<i>Function to compute bootstrap confidence intervals for robust linear regression models</i>
--------	---

---

**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**

Escale data to 0-1

**Usage**

```
scale_01(x)
```

**Arguments**

x                    A numeric variable

**Value**

Scaled data

---

search_scripts	<i>Search scripts</i>
----------------	-----------------------

---

**Description**

Searches for strings in R script files

**Usage**

```
search_scripts(string, path = getwd(), recursive = TRUE)
```

**Arguments**

string	Character string to search
path	Character vector with the path name
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	<i>Set header names for word tables</i>
----------	---

---

**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	<i>Computes skewness</i>
----------	--------------------------

---

**Description**

Calculates skewness of a numeric variable

**Usage**

```
skewness(x)
```

**Arguments**

x                    A numeric variable

**Value**

skewness value

---

track_changes	<i>track_changes</i>
---------------	----------------------

---

**Description**

Gets a data.frame with all the changes performed by the different fix functions

**Usage**

```
track_changes(x)
```

**Arguments**

x                    A data.frame

**Examples**

```
mydata<-data.frame(Dates1=c("25/06/1983", "25-08/2014", "2001/11/01", "2008-10-01"),
                   Dates2=c("01/01/85", "04/04/1982", "07/12-2016", NA),
                   Numeric1=rnorm(4))
mydata <- fix.dates(mydata)
mydata
track_changes(mydata)
```

---

ttrue	<i>True TRUE</i>
-------	------------------

---

**Description**

Makes possible logical comparisons against NULL and NA values

**Usage**

```
ttrue(x)
```

**Arguments**

x	A logical vector
---	------------------

**Value**

A logical vector

---

unforge	<i>Un-Forge</i>
---------	-----------------

---

**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**

```
#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 VarrCorr 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)
```

**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
```

---

`%>NA%`*greater & NA*

---

**Description**

'>' operator where NA values return FALSE

**Usage**`x %>NA% y`**Arguments**

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

**Value**

A logical vector of the same length as `x`

---

`%>=NA%`*geq & not NA*

---

**Description**

'>=' operator where NA values return FALSE

**Usage**`x %>=NA% y`**Arguments**

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

**Value**

A logical vector of the same length as `x`



---

%<NA%                      *less & 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%                      *leq & not 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

# Index

[%<=NA%](#), 49  
[%<NA%](#), 49  
[%>=NA%](#), 48  
[%>NA%](#), 48

[antimoda](#), 3

[check\\_quality](#), 4  
[cluster\\_var](#), 5  
[coefplot](#), 5

[descriptive](#), 6

[extreme\\_values](#), 7

[fix.dates](#), 8  
[fix.factors](#), 8  
[fix.levels](#), 9  
[fix.NA](#), 10  
[fix.numerics](#), 10  
[forge](#), 11  
[fxd](#), 12

[GK\\_assoc](#), 12  
[good2go](#), 13

[ipboxplot](#), 13  
[is.it](#), 14

[kill.factors](#), 14  
[kurtosis](#), 15

[make\\_csv\\_table](#), 15  
[make\\_latex\\_table](#), 16  
[make\\_table](#), 16  
[make\\_word\\_table](#), 17  
[matrixPaste](#), 17  
[may.numeric](#), 18  
[mine.plot](#), 18  
[moda](#), 19  
[moda\\_cont](#), 20

[mtapply](#), 20

[nearest](#), 21  
[nice\\_names](#), 21  
[numeros](#), 22

[peek](#), 22  
[plot.reportmodel](#), 23  
[prop\\_max](#), 23  
[prop\\_min](#), 24

[report](#), 24  
[report.betareg](#), 25  
[report.brmsfit](#), 26  
[report.clm](#), 26  
[report.clmm](#), 27  
[report.coxph](#), 28  
[report.data.frame](#), 29  
[report.default](#), 30  
[report.factor](#), 30  
[report.glm](#), 31  
[report.glmerMod](#), 32  
[report.glmmdmb](#), 33  
[report.glmnet](#), 34  
[report.lm](#), 35  
[report.lmerMod](#), 36  
[report.lqmm](#), 37  
[report.merModLmerTest](#), 38  
[report.numeric](#), 39  
[report.rlm](#), 39  
[report.rq](#), 40  
[restore\\_changes](#), 41  
[rob.ci](#), 41  
[rob.pvals](#), 42

[scale\\_01](#), 42  
[search\\_scripts](#), 43  
[set\\_noms](#), 43  
[skewness](#), 44

[track\\_changes](#), 44

ttrue, [45](#)

unforge, [45](#)

VarCorr, [46](#)

workspace, [46](#)

workspace\_sapply, [47](#)