

# Package ‘clickR’

March 12, 2021

**Type** Package

**Title** Semi-Automatic Preprocessing of Messy Data with Change Tracking  
for Dataset Cleaning

**Version** 0.6.64

**Author** David Hervas Marin

**Maintainer** David Hervas Marin <ddhervas@yahoo.es>

**Imports** beeswarm, boot, flextable, lme4, lmerTest, methods, officer,  
stringdist, xtable

**Description** Tools for assessing data quality, performing exploratory analysis, and  
semi-automatic preprocessing of messy data with change tracking for integral dataset cleaning.

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-03-12 15:40:02 UTC

## R topics documented:

antimoda . . . . .	3
check_quality . . . . .	4
clickR-deprecated . . . . .	5
cluster_var . . . . .	6
coefplot . . . . .	7
descriptive . . . . .	8
extreme_values . . . . .	9
fix_all . . . . .	9
fix_concat . . . . .	10
fix_dates . . . . .	10
fix_factors . . . . .	11

fix_levels	12
fix_NA	13
fix_numerics	14
forge	15
fxd	16
GK_assoc	16
good2go	17
ipboxplot	17
kill.factors	18
kurtosis	18
make_csv_table	19
make_latex_table	19
make_table	20
make_word_table	20
manual_fix	21
matrixPaste	22
may.numeric	22
mine.plot	23
moda	23
moda_cont	24
mtapply	24
nearest	25
nice_names	25
numeros	26
outliers	26
peek	27
plot.reportmodel	27
prop_may	28
prop_min	28
remove_empty	29
report	29
report.betareg	30
report.brmsfit	31
report.clm	32
report.clmm	33
report.coxph	34
report.data.frame	35
report.default	36
report.factor	36
report.glm	37
report.glmerMod	38
report.glmmdmb	39
report.glmnet	40
report.lm	41
report.lmerMod	42
report.lqmm	43
report.merModLmerTest	44
report.numeric	45

report.rlm . . . . .	45
report.rq . . . . .	46
restore_changes . . . . .	47
rob.ci . . . . .	48
rob.pvals . . . . .	48
scale_01 . . . . .	49
search_scripts . . . . .	49
set_noms . . . . .	50
skewness . . . . .	50
text_date . . . . .	51
track_changes . . . . .	51
ttrue . . . . .	52
unforge . . . . .	52
VarCorr . . . . .	53
workspace . . . . .	54
workspace_sapply . . . . .	54
%>NA% . . . . .	55
%>=NA% . . . . .	55
%<NA% . . . . .	56
%<=NA% . . . . .	56
<b>Index</b>	<b>57</b>

---

antimoda	<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
```

## Description

These functions still work but will be removed (defunct) in the next version. Please use the package `repmud` to find the updated versions of these functions.

## Details

- `report.lm`: This function is deprecated, and will be removed in the next version of this package.
- `report.glm`: This function is deprecated, and will be removed in the next version of this package.
- `report.coxph`: This function is deprecated, and will be removed in the next version of this package.
- `report.merModLmerTest`: This function is deprecated, and will be removed in the next version of this package.
- `report.lmerMod`: This function is deprecated, and will be removed in the next version of this package.
- `report.glmMod`: This function is deprecated, and will be removed in the next version of this package.
- `report.lqmm`: This function is deprecated, and will be removed in the next version of this package.
- `report.clm`: This function is deprecated, and will be removed in the next version of this package.
- `report.clmm`: This function is deprecated, and will be removed in the next version of this package.
- `report.rq`: This function is deprecated, and will be removed in the next version of this package.
- `report.betareg`: This function is deprecated, and will be removed in the next version of this package.
- `report.brmsfit`: This function is deprecated, and will be removed in the next version of this package.
- `report.glmnet`: This function is deprecated, and will be removed in the next version of this package.
- `report.rlm`: This function is deprecated, and will be removed in the next version of this package.
- `report.glmadmb`: This function is deprecated, and will be removed in the next version of this package.
- `rob.pvals`: This function is deprecated, and will be removed in the next version of this package.

- `rob.ci`: This function is deprecated, and will be removed in the next version of this package.
- `make_word_table`: This function is deprecated, and will be removed in the next version of this package.
- `make_latex_table`: This function is deprecated, and will be removed in the next version of this package.
- `make_csv_table`: This function is deprecated, and will be removed in the next version of this package.
- `make_table`: This function is deprecated, and will be removed in the next version of this package.
- `VarCorr`: This function is deprecated, and will be removed in the next version of this package.
- `set_noms`: This function is deprecated, and will be removed in the next version of this package.
- `plot.reportmodel`: This function is deprecated, and will be removed in the next version of this package.
- `coefplot`: This function is deprecated, and will be removed in the next version of this package.
- `report.data.frame`: This function is deprecated, and will be removed in the next version of this package.

---

 cluster\_var

*Clustering of variables*


---

### Description

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

### Usage

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

### Arguments

<code>x</code>	A <code>data.frame</code>
<code>margins</code>	Margins for the plot

### Value

A heatmap with the variable associations

### Examples

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

---

`coefplot`*Deprecated functions in package 'clickR'*

---

**Description**

Creates a plot of the coefficients of a model. This function will be defunct in the next version. Updated functions are available in 'repmud' package.

**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**

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

**Details**

Plot of the coefficients of a model

**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 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**

```
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_all	<i>fix_all</i>
---------	----------------

---

**Description**

Tries to automatically fix all problems in the data.frame

**Usage**

```
fix_all(x, track = TRUE)
```

**Arguments**

x	A data.frame
track	Track changes?

---

fix_concat	<i>fix_concat</i>
------------	-------------------

---

**Description**

Fixes concatenated values in a variable

**Usage**

```
fix_concat(x, varname, sep = ", |; | ", track = TRUE)
```

**Arguments**

x	A data.frame
varname	Variable name
sep	Separator for the different values
track	Track changes?

**Examples**

```
mydata <- data.frame(concat=c("a", "b", "a b" , "a b, c", "a; c"),
  numeric = c(1, 2, 3, 4, 5))
fix_concat(mydata, "concat")
```

---

fix_dates	<i>Fix_dates</i>
-----------	------------------

---

**Description**

Fixes dates. Dates can be recorded in numerous formats depending on the country, the traditions and the field of knowledge. `fix_dates` tries to detect all possible date formats and transforms all of them in the ISO standard favored by R (yyyy-mm-dd).

**Usage**

```
fix_dates(
  x,
  max.NA = 0.8,
  min.obs = nrow(x) * 0.05,
  use.probs = TRUE,
  track = TRUE
)

fix_dates(
  x,
```

```

max.NA = 0.8,
min.obs = nrow(x) * 0.05,
use.probs = TRUE,
track = TRUE
)

```

### Arguments

x	A data.frame
max.NA	Maximum allowed proportion of NA values created by coercion. If the coercion to date creates more NA values than those specified in max.NA, then all changes will be reverted and the variable will remain unchanged.
min.obs	Minimum number of non-NA observations allowed per variable. If the variable has fewer non-NA observations, then it will be ignored by fix.dates.
use.probs	When there are multiple date formats in the same column, there can be ambiguities. For example, 04-06-2015 can be interpreted as 2015-06-04 or as 2015-04-06. If use.probs=TRUE, ambiguities will be solved by assigning to the most frequent date format in the column.
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", "September 24, 2020"),
                  Numeric1=rnorm(4))
fix_dates(mydata)

```

---

fix_factors	<i>Fix factors imported as numerics</i>
-------------	---

---

### Description

Fixes factors imported as numerics. It is usual in some fields to encode factor variables as integers. This function detects such variables and transforms them into factors. When drop=TRUE (by default) it detects multiple versions of the same levels due to different capitalization, whitespaces or non-ASCII characters.

### Usage

```

fix_factors(x, k = 5, drop = TRUE, track = TRUE)

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

```

**Arguments**

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

**Examples**

```
# mtcars data has all variables encoded as numeric, even the factor variables.
descriptive(mtcars)
# After using fix_factors, factor variables are recognized as such.
descriptive(fix_factors(mtcars))
```

---

fix\_levels

*Fix levels*

---

**Description**

Fixes levels of a factor

**Usage**

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

fix.levels(
  data,
  factor_name,
  method = "dl",
  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)
method	Method from stringdist package to estimate distances
levels	Optional vector with the levels names. If "auto", levels are assigned based on frequency
plot	Optional: Plot cluster dendrogram?
k	Number of levels for clustering
track	Keep track of changes?
...	Further parameters passed to stringdist::stringdistmatrix function

**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

*fix\_NA*


---

**Description**

Fixes miscoded missing values

**Usage**

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

```
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

*Fix numeric data*


---

**Description**

Fixes numeric data. In many cases, numeric data are not recognized by R because there are data inconsistencies (wrong decimal separator, whitespaces, typos, thousand separator, etc.). `fix_numerics` detects and corrects these variables, making them numeric again.

**Usage**

```
fix_numerics(x, k = 8, max.NA = 0.2, track = TRUE)
```

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

**Arguments**

x	A data.frame
k	Minimum number of different values a variable has to have to be considered numerical
max.NA	Maximum allowed proportion of NA values created by coercion. If the coercion to numeric creates more NA values than those specified in <code>max.NA</code> , then all changes will be reverted and the variable will remain unchanged.
track	Keep track of changes?

**Examples**

```
mydata<-data.frame(Numeric1=c(7.8, 9.2, "5.4e+2", 3.3, "6,8", "3..3"),
  Numeric2=c(3.1, 1.2, "3.4s", "48,500.04 $", 7, "$ 6.4"))
descriptive(mydata)
descriptive(fix_numerics(mydata, k=5))
```

---

forge	<i>Forge</i>
-------	--------------

---

## 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, use.probs = TRUE)
```

**Arguments**

d	A character vector
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 beeswarm()

**Examples**

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

---

kill.factors	<i>Kill factors</i>
--------------	---------------------

---

**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>Deprecated functions in package 'clickR'</i>
----------------	---

---

**Description**

Exports a table to Excel. This function will be defunct in the next version. Updated functions are available in 'repmol' package.

**Usage**

```
make_csv_table(x, file, info)
```

**Arguments**

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

**Details**

Export a table to excel

**Value**

Creates a .csv file with the table

---

make_latex_table	<i>Deprecated functions in package 'clickR'</i>
------------------	---

---

**Description**

Exports a table to latex. This function will be defunct in the next version. Updated functions are available in 'repmol' package.

**Usage**

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

**Arguments**

x	A data.frame object
file	Name of the file

**Details**

Export a table to latex

**Value**

Creates a .txt file with latex code for the table

---

make_table	<i>Deprecated functions in package 'clickR'</i>
------------	---

---

**Description**

Auxiliary function to create tables. This function will be defunct in the next version. Updated functions are available in 'repmoD' package.

**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

**Details**

Make a table from report

**Value**

Creates a file with the table

---

make_word_table	<i>Deprecated functions in package 'clickR'</i>
-----------------	---

---

**Description**

Exports a table to Word. This function will be defunct in the next version. Updated functions are available in 'repmoD' package.

**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?

**Details**

Export a table to word

**Value**

Creates a word file with the table

---

manual_fix	<i>Tracked manual fixes to data</i>
------------	-------------------------------------

---

**Description**

Tracks manual fixes performed on a variable in a data.frame

**Usage**

```
manual_fix(data, variable, subset, newvalues = NULL)
```

**Arguments**

data	A data.frame
variable	A character string with the name of the variable to be fixed
subset	A logical expression for selecting the cases to be fixed
newvalues	New value or values that will take the cases selected by subset parameter.

**Examples**

```
iris2 <- manual_fix(iris, "Petal.Length", Petal.Length < 1.2, 0)
track_changes(iris2)
```

---

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,  
  fun = is.na,  
  spacing = 5,  
  sort = F,  
  show.x = TRUE,  
  show.y = TRUE,  
  ...  
)
```

**Arguments**

x	A data.frame
fun	A function that evaluates a vector and returns a logical vector
spacing	Numerical separation between lines at the y-axis
sort	If TRUE, variables are sorted according to their 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, fun=outliers) #Shows extreme values
```

---

moda	<i>Get mode</i>
------	-----------------

---

**Description**

Returns the most repeated value

**Usage**

```
moda(x)
```

**Arguments**

x                    A categorical variable

**Value**

The mode

---

moda_cont	<i>Estimates number of modes</i>
-----------	----------------------------------

---

**Description**

Estimates the number of modes

**Usage**

```
moda_cont(x)
```

**Arguments**

x                    A numeric variable

**Value**

Estimated number of modes.

---

mtapply	<i>Multiple tapply</i>
---------	------------------------

---

**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(x, track = TRUE)
```

**Arguments**

x	A data.frame
track	Track changes?

**Examples**

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

---

numeros

*Brute numeric coercion*

---

### Description

If possible, coerces values from a vector to numeric

### Usage

```
numeros(x)
```

### Arguments

x                    A vector

### Value

A numeric vector

---

outliers

*outliers*

---

### Description

Function for detecting outliers based on the boxplot method

### Usage

```
outliers(x)
```

### Arguments

x                    A vector

### Examples

```
outliers(iris$Petal.Length)
outliers(airquality$Ozone)
```

---

`peek`*Peek*

---

**Description**

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

**Usage**

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

**Arguments**

<code>x</code>	A data.frame
<code>n</code>	Number of rows to include in output
<code>which</code>	Columns to include in output

**Examples**

```
peek(iris)
```

---

`plot.reportmodel`*Deprecated functions in package 'clickR'*

---

**Description**

Creates a coefplot from the reportmodel object. This function will be defunct in the next version. Updated functions are available in 'repmo' package.

**Usage**

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

**Arguments**

<code>x</code>	A reportmodel object
<code>...</code>	Further arguments passed to coefplot

**Details**

Coefplot for reportmodel objects

**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	<i>Gets proportion of most repeated value</i>
----------	---

---

**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

---

remove_empty	<i>remove_empty</i>
--------------	---------------------

---

**Description**

Removes empty rows or columns from data.frames

**Usage**

```
remove_empty(x, track = TRUE)
```

**Arguments**

x	A data.frame
track	Track changes?

**Examples**

```
mydata <- data.frame(a = c(NA, NA, NA, NA, NA), b = c(1, NA, 3, 4, 5),
  c=c(NA, NA, NA, NA, NA), d=c(4, NA, 5, 6, 3))
remove_empty(mydata)
```

---

report	<i>Deprecated functions in package 'clickR'</i>
--------	---

---

**Description**

Generic function for creating reporting tables. This function will be defunct in the next version. Updated functions are available in 'repmol' package.

**Usage**

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

**Arguments**

x	An compatibleobject that can be summarized
...	further arguments passed to make_table

**Details**

Generic function for creating reporting tables

**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                      *Deprecated functions in package 'clickR'*

---

**Description**

Creates a report table from a beta regression model. This function will be defunct in the next version. Updated functions are available in 'repmud' package.

**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

**Details**

Report from beta regression model

**Value**

A data frame with the report table

---

report.brmsfit      *Deprecated functions in package 'clickR'*

---

### Description

Creates a report table from model fitted by brms. This function will be defunct in the next version. Updated functions are available in 'repmol' package.

### 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

### Details

Report models from brms package

### Value

A data frame with the report table

---

`report.clm`*Deprecated functions in package 'clickR'*

---

### Description

Creates a report table from an ordinal model. This function will be defunct in the next version. Updated functions are available in 'repmol' package.

### Usage

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

### Arguments

<code>x</code>	An ordinal 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>

### Details

Report from ordinal model

### Value

A data frame with the report table



---

`report.clmm`*Deprecated functions in package 'clickR'*

---

### Description

Creates a report table from an ordinal mixed model. This function will be defunct in the next version. Updated functions are available in 'repmud' package.

### Usage

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

### Arguments

<code>x</code>	An ordinal 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>

### Details

Report from ordinal mixed model

### Value

A data frame with the report table

---

`report.coxph`*Deprecated functions in package 'clickR'*

---

### Description

Creates a report table from a cox model. This function will be defunct in the next version. Updated functions are available in 'repmo' package.

### Usage

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

### Arguments

<code>x</code>	A cox 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>

### Details

Report from cox regression model

### Value

A data frame with the report table

---

report.data.frame      *Deprecated functions in package 'clickR'*

---

### Description

Creates a report table ready for publication. This function will be defunct in the next version. Updated functions are available in 'repmud' package.

### 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()

### Details

Report tables of summary data

### Examples

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

---

report.default	<i>Deprecated functions in package 'clickR'</i>
----------------	---

---

**Description**

This is a default function for calling `summary(x)` on non-implemented classes. This function will be defunct in the next version. Updated functions are available in 'repmo' package.

**Usage**

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

**Arguments**

x	Any object without specific report function
...	further arguments passed to summary

**Details**

Default function for report

**Value**

A summary of the object

---

report.factor	<i>Deprecated functions in package 'clickR'</i>
---------------	---

---

**Description**

Creates a report table. This function will be defunct in the next version. Updated functions are available in 'repmo' package.

**Usage**

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

**Arguments**

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

**Details**

Report from categorical variable

**Value**

A data frame with the report table

---

report.glm	<i>Deprecated functions in package 'clickR'</i>
------------	---

---

**Description**

Creates a report table from a generalized linear model. This function will be defunct in the next version. Updated functions are available in 'repmod' package.

**Usage**

```
## 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

**Details**

Report from generalized linear model

**Value**

A data frame with the report table

---

report.glmerMod      *Deprecated functions in package 'clickR'*

---

### Description

Creates a report table from a generalized linear mixed model. This function will be defunct in the next version. Updated functions are available in 'repmol' package.

### 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

### Details

Report from generalized linear mixed model

### Value

A data frame with the report table

---

report.glmmb *Deprecated functions in package 'clickR'*

---

### Description

Creates a report table from a glmmb model. This function will be defunct in the next version. Updated functions are available in 'repmob' package.

### 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

### Details

Report from generalized linear mixed model from ADMB

### Value

A data frame with the report table

---

`report.glmnet`*Deprecated functions in package 'clickR'*

---

**Description**

Creates a report table from models fitted by glmnet. This function will be defunct in the next version. Updated functions are available in 'repmud' package.

**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>

**Details**

Report models from glmnet package

**Value**

A data frame with the report table



---

`report.lm`*Deprecated functions in package 'clickR'*

---

### Description

Creates a report table from a linear model. This function will be defunct in the next version. Updated functions are available in 'repmo' package.

### 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>

### Details

Report from linear model

### Value

A data frame with the report table

---

report.lmerMod      *Deprecated functions in package 'clickR'*

---

### Description

Creates a report table from a linear mixed model. This function will be defunct in the next version. Updated functions are available in 'repmol' package.

### 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

### Details

Report from linear mixed model

### Value

A data frame with the report table

---

`report.lqmm`*Deprecated functions in package 'clickR'*

---

### Description

Creates a report table from a quantile mixed model. This function will be defunct in the next version. Updated functions are available in 'repmol' package.

### Usage

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

### Arguments

<code>x</code>	A quantile 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>

### Details

Report from quantile mixed model

### Value

A data frame with the report table

---

report.merModLmerTest *Deprecated functions in package 'clickR'*

---

### Description

Creates a report table from a linear mixed model. This function will be defunct in the next version. Updated functions are available in 'repmol' package.

### 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

### Details

Report from linear mixed model with pvalues

### Value

A data frame with the report table

---

report.numeric	<i>Deprecated functions in package 'clickR'</i>
----------------	---

---

**Description**

Creates a report table. This function will be defunct in the next version. Updated functions are available in 'repmud' package.

**Usage**

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

**Arguments**

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

**Details**

Report from numeric variable

**Value**

A data frame with the report table

---

report.rlm	<i>Deprecated functions in package 'clickR'</i>
------------	---

---

**Description**

Creates a report table from a robust linear model. This function will be defunct in the next version. Updated functions are available in 'repmud' package.

**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

**Details**

Report from robust linear model (rlm)

**Value**

A data frame with the report table

---

report.rq

*Deprecated functions in package 'clickR'*

---

**Description**

Creates a report table from a quantile regression model. This function will be defunct in the next version. Updated functions are available in 'repmud' package.

**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

**Details**

Report from quantile regression model

**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(tracking)
```

**Arguments**

tracking	A data.frame generated by track_changes() function
----------	--

**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
tracking <- track_changes(mydata)
mydata_r <- restore_changes(tracking)
mydata_r
```

---

rob.ci *Deprecated functions in package 'clickR'*

---

**Description**

Estimates confidence intervals for rlm models. This function will be defunct in the next version. Updated functions are available in 'repmo' package.

**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

**Details**

Function to compute bootstrap confidence intervals for robust linear regression models

**Value**

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

---

rob.pvals *Deprecated functions in package 'clickR'*

---

**Description**

Estimates p-values for rlm models. This function will be defunct in the next version. Updated functions are available in 'repmo' package.

**Usage**

```
rob.pvals(x)
```

**Arguments**

x	A rlm object
---	--------------

**Details**

Function to compute p-values for robust linear regression models



**Value**

A vector of p-values

---

scale_01	<i>Scales data between 0 and 1</i>
----------	------------------------------------

---

**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>Deprecated functions in package 'clickR'</i>
----------	---

---

**Description**

Internal function for `make_word_table`. This function will be defunct in the next version. Updated functions are available in 'repmol' package.

**Usage**

```
set_noms(x, args)
```

**Arguments**

x	A flextable object
args	A names list with the header names

**Details**

Set header names for word tables

**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

---

text_date	<i>Internal function for dates with text</i>
-----------	--

---

**Description**

Function to transform text into dates

**Usage**

```
text_date(date, format = "%d/%Y %b")
```

**Arguments**

date	A date
format	Format of the date

---

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, subset)
```

**Arguments**

x	A data.frame
subset	Logical expression for subsetting the data.frame with the 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))
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

*Deprecated functions in package 'clickR'*

---

## Description

Extract Variance-Covariance Matrix. This function will be defunct in the next version. Updated functions are available in 'repmud' package.

## 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

## Details

Generic VarCorr function

## 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%	<i>greater &amp; NA</i>
-------	-------------------------

---

**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%	<i>geq &amp; not NA</i>
--------	-------------------------

---

**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%

*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

---

%&lt;=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%`, 56  
`%<NA%`, 56  
`%>=NA%`, 55  
`%>NA%`, 55

`antimoda`, 3

`check_quality`, 4  
`clickR-deprecated`, 5  
`cluster_var`, 6  
`coefplot`, 6, 7

`descriptive`, 8

`extreme_values`, 9

`fix.dates` (`fix_dates`), 10  
`fix.factors` (`fix_factors`), 11  
`fix.levels` (`fix_levels`), 12  
`fix.NA` (`fix_NA`), 13  
`fix.numerics` (`fix_numerics`), 14  
`fix_all`, 9  
`fix_concat`, 10  
`fix_dates`, 10  
`fix_factors`, 11  
`fix_levels`, 12  
`fix_NA`, 13  
`fix_numerics`, 14  
`forge`, 15  
`fxd`, 16

`GK_assoc`, 16  
`good2go`, 17

`ipboxplot`, 17

`kill.factors`, 18  
`kurtosis`, 18

`make_csv_table`, 6, 19  
`make_latex_table`, 6, 19

`make_table`, 6, 20  
`make_word_table`, 6, 20  
`manual_fix`, 21  
`matrixPaste`, 22  
`may.numeric`, 22  
`mine.plot`, 23  
`moda`, 23  
`moda_cont`, 24  
`mtapply`, 24

`nearest`, 25  
`nice_names`, 25  
`numeros`, 26

`outliers`, 26

`peek`, 27  
`plot.reportmodel`, 6, 27  
`prop_may`, 28  
`prop_min`, 28

`remove_empty`, 29  
`report`, 29  
`report.betareg`, 5, 30  
`report.brmsfit`, 5, 31  
`report.clm`, 5, 32  
`report.clmm`, 5, 33  
`report.coxph`, 5, 34  
`report.data.frame`, 6, 35  
`report.default`, 36  
`report.factor`, 36  
`report.glm`, 5, 37  
`report.glmMod`, 5, 38  
`report.glmMadmb`, 5, 39  
`report.glmnet`, 5, 40  
`report.lm`, 5, 41  
`report.lmerMod`, 5, 42  
`report.lqmm`, 5, 43  
`report.merModLmerTest`, 5, 44  
`report.numeric`, 45

report.rlm, [5](#), [45](#)  
report.rq, [5](#), [46](#)  
restore\_changes, [47](#)  
rob.ci, [6](#), [48](#)  
rob.pvals, [5](#), [48](#)  
  
scale\_01, [49](#)  
search\_scripts, [49](#)  
set\_noms, [6](#), [50](#)  
skewness, [50](#)  
  
text\_date, [51](#)  
track\_changes, [51](#)  
ttrue, [52](#)  
  
unforge, [52](#)  
  
VarCorr, [6](#), [53](#)  
  
workspace, [54](#)  
workspace\_sapply, [54](#)