

# Package ‘ClinReport’

September 3, 2019

**Type** Package

**Title** Statistical Reporting for Scientific Publications,  
Epidemiological Studies and Clinical Trials

**Version** 0.9.1.14

**Date** 2019-09-03

**Author** Jean-Francois Collin

**Maintainer** Jean-Francois Collin <jfcollin@live.fr>

## Description

Export to 'Microsoft Word' or R markdown documents nicely formatted tables of descriptive or inferential statistics. Can be used within the scope of clinical trials, or for any scientific publications. Descriptive tables for quantitative statistics (mean, median, max etc..) and/or qualitative statistics (frequencies and percentages) are available along with formatted tables of Least Square Means of Linear Models, Linear Mixed Models, Cox Models and Generalized Linear Mixed Models. It works mainly with 'emmeans' 'officer' and 'flextable' packages.

**Depends** R (>= 3.5.0)

**License** GPL-3

**Imports** stats, ggplot2, reshape2, dplyr (>= 0.8.0.1), emmeans (>= 1.3.2), officer (>= 0.3.3), utils, flextable (>= 0.5.1), xtable, lme4, car

**Suggests** nlme, magrittr, knitr, rmarkdown, survival

**VignetteBuilder** knitr

**URL** <https://jfrancoiscollin.github.io/ClinReport>

**BugReports** <https://github.com/jfrancoiscollin/ClinReport/issues>

**Encoding** UTF-8

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-09-03 11:50:02 UTC

**R topics documented:**

add.stat . . . . .	2
adverse_event . . . . .	4
at.least.one . . . . .	5
check.x . . . . .	5
ClinReport . . . . .	6
datafake . . . . .	6
define.key . . . . .	7
desc . . . . .	8
dimnames.desc . . . . .	9
gg_desc_lsmeans . . . . .	10
gg_desc_quali . . . . .	11
gg_desc_quanti . . . . .	12
is.desc . . . . .	13
make.label . . . . .	13
plot.desc . . . . .	14
pretty . . . . .	16
prettyround . . . . .	17
print.desc . . . . .	17
regroup . . . . .	18
report.doc . . . . .	19
report.lsmeans . . . . .	27
report.modelinfo . . . . .	29
report.quali . . . . .	31
report.quali.hlev . . . . .	33
report.quanti . . . . .	34
report.sessionInfo . . . . .	37
spacetable . . . . .	39
space_vline . . . . .	40
split.desc . . . . .	40
time_to_cure . . . . .	41
transpose . . . . .	42
<b>Index</b>	<b>44</b>

---

 add.stat

*Add a new statistic to an existing table*


---

**Description**

add.stat.desc enables to add a specific quantitative statistic into an existing table of statistics.

**Usage**

```
add.stat(tab, data, func.stat, func.stat.name, pos, ...)  
  
## S3 method for class 'desc'  
add.stat(tab, data, func.stat, func.stat.name, pos = NULL,  
        ...)
```

**Arguments**

tab	A desc object
data	The data frame used to create tab
func.stat	Passed to report.quantile function
func.stat.name	Passed to report.quantile function
pos	Numeric used to specify the position of the new statistics
...	Other parameters

**Details**

Add a specific statistic in a desc object (only works for quantitative statistics for now). It calls the [report.quantile](#) function with the same attributes as the tab argument and uses the [regroup](#) function to bind the two results.

**Value**

A "quantile" desc object (desc\$type.desc equal to "quantile")

**See Also**

[regroup](#) [report.quantile](#)

**Examples**

```
# Load the data  
data(datafake)  
  
# The default statistics are given here:  
tab1=report.quantile(data=datafake,y="y_numeric",x1="GROUP",total=TRUE,subjid="SUBJID")  
  
# Define the function corresponding to the coefficient of variation for example  
cv=function(y) sd(y,na.rm=TRUE)/mean(y,na.rm=TRUE)  
  
# We use the add.stat function to add CV at the second row:  
tab1.cv=add.stat(tab1,datafake,func.stat=cv,func.stat.name="Coef. Var",  
pos=2)
```

```

tab1.cv

# Same with 2 explicative variables

tab=report.quanti(data=datafake,y="y_numeric",x1="GROUP",
x2="TIMEPOINT",total=TRUE,subjid="SUBJID",
at.row="TIMEPOINT")

tab=add.stat(tab,datafake,func.stat=cv,func.stat.name="Coef. Var",
pos=2)
tab

# And on position 5, we can add for example the mode

mode=function(x)
{
  x=na.omit(x)
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

tab=add.stat(tab,datafake,func.stat=mode,func.stat.name="Mode",
pos=5)
tab

```

---

adverse\_event

*adverse\_event*


---

## Description

Fake adverse event data example

## Usage

```
data(adverse_event)
```

## Format

A data frame #'

**SOCNAME** Fake System Organ Class variable

**HLTNAME** Fake factor

**PTNAME** Fake Preferred Terms variable

**randtrt** Fake treatment group factor

**SUBJID** Fake subject Id

---

at.least.one	<i>Create a key from two hierarchical factors</i>
--------------	---

---

**Description**

Used to calculate the 'at least one' statistics for several occurrence of an event per statistical unit.

**Usage**

```
at.least.one(data, subjid = NULL, var, total = FALSE,
             var.label = "var")
```

**Arguments**

data	A data frame
subjid	A character. Indicates the ID
var	A character. Indicates a factor in the data frame
total	Logical. if TRUE the total number of events is calculated
var.label	A character. The label of var

**Details**

Used internally by report.quali.hlev

---

check.x	<i>Check if the variable argument supplied in report.quantis and report.qualis is ok</i>
---------	--

---

**Description**

Check if the variable argument supplied in report.quantis and report.qualis is ok

**Usage**

```
check.x(data, x, substitute = "data")
```

**Arguments**

data	Data.frame object
x	Character
substitute	Character. The name of the data frame

**Details**

Used internally by report.quantis and report.qualis

---

ClinReport

*R documentation for ClinReport package*

---

## Description

The aim is to create easily a statistical reporting in Microsoft Word documents with tables in a pretty format according to "clinical standards".

## Details

It can also be used outside the scope of clinical trials, for any statistical reporting in Word.

Descriptive tables for quantitative statistics (mean, median, max etc..) and qualitative statistics (frequencies and percentages) are available and pretty tables of Least Square Means of LM, LME, Cox and GLM models coming from emmeans function in emmeans package are also available.

The package works with officer and flextable packages to export the outputs into Microsoft Word documents.

There are two main functions for getting descriptive statistics : `report.quant` and `report.qual`

There is one unique function for prettyfying LS means: `report.lsm`

Each of those 3 functions creates a desc object that can be used with `report.doc` to export the results. This is the unique function to export the outputs into a Word or an R markdown document.

Future versions will improve the "standard" graphic reporting and specific statistics tables used for reporting Adverse Events.

You can visit the website for more examples: [ClinReport website](#)

## Author(s)

Jean-Francois Collin, <jfcollin@live.fr>

---

datafake

*datafake*

---

## Description

Fictive data created only for the purpose of testing the package and showing examples.

## Usage

```
data(datafake)
```

**Format**

A data frame

**y\_numeric** Fake numerical response

**y\_logistic** Fake Logistic response

**y\_poisson** Fake Poisson response

**baseline** A baseline covariate

**VAR** A factor

**GROUP** A fake treatment factor

**TIMEPOINT** A fake time factor

**SUBJID** A fake id factor

---

define.key	<i>Create a key from two hierarchical factors</i>
------------	---

---

**Description**

Used to create a single key from two hierarchical factors.

**Usage**

```
define.key(data, upper = "", lower = "",
           upper.levels = "Upper.Levels", lower.levels = "Lower.Levels")
```

**Arguments**

data	A data frame
upper	A character. Indicates a factor in the data frame which corresponds to the factor with the higher number of levels (typically SOC variable)
lower	A character. Indicates a factor in the data frame which corresponds to the factor with the lower number of levels (typically PT variable)
upper.levels	A character. The label to be displayed in the table for the upper terms
lower.levels	A character. The label to be displayed in the table for the lower terms

**Details**

Used internally by report.quali.hlev

---

 desc

*Constructor function for desc objects*


---

## Description

Creates a desc object.

## Usage

```
desc(output = NULL, total = NULL, nbc0l = NULL, y = NULL,
      x1 = NULL, x2 = NULL, at.row = NULL, subjid = NULL,
      type.desc = NULL, type = NULL, y.label = NULL, type.mod = NULL,
      raw.output = NULL, contrast = NULL, contrast.name = NULL,
      stat.name = NULL, regrouped = F, rbind.label = NULL,
      title = NULL, y.levels.label = NULL)
```

## Arguments

output	a data.frame object
total	Boolean. Indicates if a column Total should be added or not
nbc0l	Numeric. number of columns that are not statistics
y	Character indicating the response variable
x1	Character indicating a factor in the data
x2	Character indicating a factor in the data
at.row	Character. Pass to spacetable function. Used to space the results (see <a href="#">spacetable</a> )
subjid	Character. Indicates in the data.frame the name of the column used to identify the Id of the subjects. If not null, it adds in the headers the number of unique subject per levels of x1 or y (if x1 is null).
type.desc	Character. Can be "quali" "quanti" "lsmeans" or "quali_quanti"
type	Character. Can be "response" or "link"
y.label	Character. The label for y response variable
type.mod	Character. The type of models the LS means are coming from (can be 'quali' or 'quanti')
raw.output	Data.frame. The raw statistics unformatted from report.quali, report.quanti and report.lsmeans
contrast	Logical. Specify if the contrast function has been used after the emmeans function
contrast.name	Character. Corresponds to the label of the column in which the contrasts are specified Default value is 'contrast'.
stat.name	Character. Indicates the name of the variable that report the statistics in report.quanti function. Default = "Statistics"



regrouped	Logical. used to indicate if the desc object regroup several desc objects (see <a href="#">regroup.desc</a> )
rbind.label	Character. The name of the label in case several desc objects are regrouped
title	Character. The title of the desc object that will be printed and used in <a href="#">report.doc</a>
y.levels.label	Character. The label to be displayed in the table which indicates the level of y

**Details**

Used internally by codereport.XXX functions

**Value**

A desc object

**See Also**

[report.quali](#) [report.quanti](#) [report.lsmmeans](#)

**Examples**

```
## Not run:
x=desc()

## End(Not run)
```

---

dimnames.desc	<i>dimnames method for desc object</i>
---------------	--

---

**Description**

dimnames method for desc object

**Usage**

```
## S3 method for class 'desc'
dimnames(x)
```

**Arguments**

x                    A desc object

**See Also**

[desc](#)

---

gg_desc_lsmeans	<i>Creates a ggplot object corresponding to a LS Means desc object</i>
-----------------	--

---

### Description

gg\_desc\_lsmeans ggplot object is created. It is used internally in function [plot.desc](#). It's easier to use this last one.

### Usage

```
gg_desc_lsmeans(desc, title = "", ylim = NULL, xlim, xlab = "",  
  ylab = "", legend.label = "Group", add.ci = F, add.line = T)
```

### Arguments

desc	Desc object
title	Character The title of the plot
ylim	Numeric of length 2 for setting y axis limits
xlim	Numeric of length 2 for setting x axis limits
xlab	Character Label for x-axis
ylab	Character Label for y-axis
legend.label	Character Label for the legend (used only if x1 and x2 are not NULL in the desc object)
add.ci	Logical. If TRUE it adds bars to the means representing 95% CI
add.line	Logical. If TRUE it joins the dots with a line (default to TRUE)

### Details

It is used internally in function [plot.desc](#). It's easier to use this last one.

### Value

A ggplot object.

### See Also

[plot.desc](#) [report.lsmmeans](#) [gg\\_desc\\_quali](#) [gg\\_desc\\_quant](#)

### Examples

---

gg_desc_quali	<i>Creates a ggplot object corresponding to a qualitative desc object</i>
---------------	---

---

### Description

gg\_desc\_quali ggplot object is printed. It is used internally in function [plot.desc](#). It's easier to use this last one.

### Usage

```
gg_desc_quali(desc, title = "", ylim = NULL, xlim = NULL,  
              xlab = "", ylab = "Percentage", legend.label = "Group")
```

### Arguments

desc	Desc object
title	Character The title of the plot
ylim	Numeric of length 2 for setting y axis limits
xlim	Numeric of length 2 for setting x axis limits
xlab	Character Label for x-axis
ylab	Character Label for y-axis
legend.label	Character Label for the legend (used only if x1 and x2 are not NULL in the desc object)

### Details

No detail

### See Also

[report.quali](#) [plot.desc](#) [gg\\_desc\\_quant](#) [gg\\_desc\\_quali](#) [gg\\_desc\\_lsmeans](#)

### Examples

---

gg\_desc\_quanti      *Creates a ggplot object corresponding to a quantitative desc object*

---

### Description

gg\_desc\_quanti ggplot object is printed

### Usage

```
gg_desc_quanti(desc, title = "", ylim = NULL, xlim = NULL,  
              xlab = "", ylab = "", legend.label = "Group", add.sd = F)
```

### Arguments

desc	Desc object
title	Character The title of the plot
ylim	Numeric of length 2 for setting y axis limits
xlim	Numeric of length 2 for setting x axis limits
xlab	Character Label for x-axis
ylab	Character Label for y-axis
legend.label	Character Label for the legend (used only if x1 and x2 are not NULL in the desc object)
add.sd	Logical. If TRUE it adds bars to the means representing +/-SD

### Details

It is used internally in function [plot.desc](#). It's easier to use this last one.

### Value

A ggplot object.

### See Also

[plot.desc](#) [desc](#) [gg\\_desc\\_quali](#) [gg\\_desc\\_lsmeans](#)

### Examples

---

is.desc	<i>Check if it is really a desc object</i>
---------	--

---

**Description**

Check if it is really a desc object

**Usage**

```
is.desc(x)
```

**Arguments**

x	Normally, a desc object, but it can be anything...
---	--

**See Also**

[desc](#)

---

make.label	<i>Make pretty labels from variable names</i>
------------	---

---

**Description**

```
make.label
```

Mainly used to prettyfy the rownames of anova tables.

**Usage**

```
make.label(x, l = list(l1 = c("_", " at "), l2 = c(":",  
" interaction with "), l3 = c(".", " ")))
```

**Arguments**

x	Character
l	List Each first element is gsubised by the second element of the list

**Details**

Replace with gsub function each first element of `l[[i]]` by the second element of `l[[i]]` for each element of `x`.

**Value**

A character

**Examples**

```

data(datafake)
an=anova(lm(y_numeric~y_logistic+GROUP:TIMEPOINT,data=datafake))

# Raw output:
an

rownames(an)=make.label(rownames(an),
l=list(
l1=c("y_logistic","A logistic variable"),
l2=c(":", " interaction with "))
)

# Gives:

an

```

---

plot.desc

*Display the corresponding graphic of statistical table*


---

**Description**

Display the corresponding graphic of statistical table

**Usage**

```

## S3 method for class 'desc'
plot(x, ..., title = "", ylim = NULL, xlim = NULL,
     xlab = "", ylab = "", legend.label = "Group", add.sd = F,
     add.ci = F, size.title = 10, add.line = T)

```

**Arguments**

x	A desc object
...	Other parameters
title	Character The title of the plot
ylim	Numeric of length 2 for setting y axis limits
xlim	Numeric of length 2 for setting x axis limits
xlab	Character Label for x-axis
ylab	Character Label for y-axis
legend.label	Character Label for the legend (used only if x1 and x2 are not NULL in the desc object)
add.sd	Logical. If TRUE it adds bars to the means representing +/-SD (for desc quanti reporting only)

add.ci	Logical. If TRUE it adds bars to the means representing 95% CI (for LS Means reporting only)
size.title	Numeric. used to specify the font size of the title
add.line	Logical. If TRUE it joins the dots with a line (default to TRUE)

## Details

It's a wrapper function which uses [gg\\_desc\\_quant](#), [gg\\_desc\\_quali](#) or [gg\\_desc\\_lsmeans](#) depending if the desc object is of type "quali", "quanti" or "lsmeans" (argument `type.desc` in `desc` object, see examples below).

## See Also

[desc](#)

## Examples

```
data(datafake)

desc=report.quali(data=datafake,y="y_logistic",x1="GROUP",
x2="TIMEPOINT")

plot(desc,
title="Qualitative desc object with 2 explicative variables",
legend.label="Y levels")

desc2=report.quanti(data=datafake,y="y_numeric",x1="GROUP",
x2="TIMEPOINT")

plot(desc2,
title="Quantitative desc object with 2 explicative variables",
legend.label="Treatment groups")

#Removing baseline data in the response, for the model

data.mod=droplevels(datafake[datafake$TIMEPOINT!="D0",])

library(nlme)
library(emmeans)

mod3=lme(y_numeric~baseline+GROUP+TIMEPOINT+GROUP*TIMEPOINT,
random=~1|SUBJID,data=data.mod,na.action=na.omit)

test3=emmeans(mod3,~GROUP|TIMEPOINT)

tab.mod3=report.lsmmeans(lsm=test3)
```

```
gg=plot(tab.mod3,title="LS Means plot example")  
  
#Print  
  
gg
```

---

pretyp                      *Return a p-value vector in nice format*

---

### Description

pretyp Return a formatted version of pvalues with nice format.

### Usage

```
pretyp(p, r = 3)
```

### Arguments

p	A vector of numerical p-values
r	A numeric values that indicates the number of digits to round to

### Details

It takes a p and return another one, but better.

### Value

Return a formated p-value vector with rounded numbers and <0.001 instead of 0.00

### Examples

```
pretyp(c(0.05,0.001,0.00001),3)
```



---

prettyround	<i>Return a rounded vector with equal number of digits</i>
-------------	--

---

**Description**

prettyround Return a formatted version of the numeric vector.

**Usage**

```
prettyround(p, r = 3)
```

**Arguments**

p	Vector of numerical values
r	Integer indicates the number of digits to round to

**Details**

No detail.

**Value**

Return a formatted vector with rounded numbers

**Examples**

```
prettyround(c(-0.05,0.001,0.00001),3)
```

---

print.desc	<i>Print method for desc object</i>
------------	-------------------------------------

---

**Description**

Print method for desc object

**Usage**

```
## S3 method for class 'desc'  
print(x, ...)
```

**Arguments**

x	A desc object
...	Other parameters

**See Also**[desc](#)

---

**regroup***Regroup two descriptive tables into one*

---

**Description**

regroup regroup two descriptive tables (qualitative or quantitative) into one

**Usage**

```
regroup(x, y, ...)

## S3 method for class 'desc'
regroup(x, y, rbind.label = "Response", ...)
```

**Arguments**

x	A desc object
y	A desc object
...	Other parameters
rbind.label	Character. The label for rbind column header

**Details**

Regroup a quantitative table and a qualitative table, is only possible if there is one and only one explicative variable. So it works if and only if x1 argument in x and y objects are not NULL, are the same and if x2 argument is NULL in both x and y objects.

The function takes the y.label argument of object x and y respectively as label for the levels of the new column created under the name of rbind.label (see example below)

It's also possible to regroup two quantitative tables, in this case it's possible if there is one or two explicative variables.

For now it's not possible to regroup two qualitative tables.

**Value**

A desc object corresponding to a table of statistics.

**See Also**

[report.quali](#) [report.quanti](#) [report.doc](#) [desc](#)

## Examples

```

data(datafake)

# Example with a qualitative and a quantitative tables
#The argument y.label is stored in the desc object and
# only used after by the regroup function

tab1=report.quanti(data=datafake,y="y_numeric",
x1="GROUP",subjid="SUBJID",y.label="Y numeric")

tab2=report.quali(data=datafake,y="y_logistic",
x1="GROUP",subjid="SUBJID",y.label="Y logistic")

regroup(tab1,tab2,rbind.label="The label of your choice")

# Example with 2 quantitative tables

tab1=report.quanti(data=datafake,y="y_numeric",
x1="GROUP",subjid="SUBJID",y.label="Y numeric")

datafake$y_numeric2=rnorm(length(datafake$y_numeric))

tab2=report.quanti(data=datafake,y="y_numeric2",
x1="GROUP",subjid="SUBJID",y.label="Y Numeric 2")

regroup(tab1,tab2,rbind.label="The label of your choice")

```

---

report.doc

*Export a statistical table into a 'Microsoft Word' or a R markdown document*

---

## Description

report.doc This function enables to export the table created with [report.quali](#) [report.quanti](#) or [report.lsmmeans](#) to a Microsoft Word or a R markdown document.

It's also possible to use report.doc to have a preview of the table in HTML format if the doc argument is NULL.

For more examples see the website: [ClinReport website](#)

## Usage

```

report.doc(table, ...)

## S3 method for class 'desc'
report.doc(table, title = NULL, colspan.value = NULL,

```

```

doc = NULL, init.numbering = F, numbering = T,
font.name = "Times", page.break = T, font.size = 10, valign = F,
...)

## S3 method for class 'anova'
report.doc(table, title = "Anova table",
  type.anova = 3, doc = NULL, numbering = T, init.numbering = F,
  font.name = "Times", font.size = 10, page.break = T,
  pretty.label = FALSE, ...)

## S3 method for class 'anova.lme'
report.doc(table, title = "Anova table",
  type.anova = 3, doc = NULL, numbering = T, init.numbering = F,
  font.name = "Times", font.size = 10, page.break = T,
  pretty.label = FALSE, ...)

```

## Arguments

table	A desc object that report statistics (the results of <code>report.quantil</code> or <code>report.quali</code> )
...	Other arguments
title	Character. The title of the table
colspan.value	Character. Add the label of the x1 variable levels (typically "Treatment Groups")
doc	NULL or a rdocx object
init.numbering	Logical. If TRUE Start numbering of the output at 1, otherwise it increase the output numbering of 1 unit
numbering	Logical. If TRUE Output numbers are added before the title.
font.name	Character. Passed to <code>font</code> function. Set the font of the output in Word
page.break	Logical. If TRUE it adds a page break after the output. Default to TRUE
font.size	Numeric. Passed to <code>fontsize</code> function. Set the font size of the output in Word
valign	Logical. If TRUE it aligns vertically the levels of the merged cells in the first column
type.anova	Passed to Anova function from car package (see its documentation).
pretty.label	Logical. Default to FALSE. If TRUE, use the function <code>make.label</code> with default option on the rownames of the anova table
anova	Logical. Used to specify if the table is an anova table. By default it's not

## Details

This function creates a flextable object from a desc object.

### For Microsoft Word documents:

The argument `doc` should be used so the flextable is added to a rdocx object.

Like:

```
doc=read_docx()
```

```
tab=report.quanti(data=data,y="y_numeric",x1="GROUP")
doc=report.doc(tab,doc=doc)
```

#### For R markdown documents:

Just don't use the doc argument. Something like:

```
““{r, include=TRUE}
tab=report.quanti(data=data,y="y_numeric",x1="GROUP")
doc=report.doc(tab)
doc
““
```

### Value

A flextable object (if doc=NULL) or a rdocx object (if doc= an rdocx object).

### See Also

[report.quali](#) [report.quanti](#) [report.lsmmeans](#) [desc](#)

### Examples

```
#####
# Import libraries
#####

library(officer)
library(flextable)
library(reshape2)
library(emmeans)
library(lme4)
library(nlme)
library(ggplot2)
library(car)
library(xtable)

#####
# Load data
#####

data(datafake)
head(datafake)

# Removing baseline data for the model

data.mod=droplevels(datafake[datafake$TIMEPOINT!="D0",])
```

```
#####  
# Create your stats tables and graphics  
#####  
  
# Quantitative stats (2 explicative variables) #####  
# since it's a big enough table, we don't want it to overlap 2 pages  
# so we split it in two with split.desc function  
  
tab1=report.quanti(data=datafake,y="y_numeric",  
x1="GROUP",x2="TIMEPOINT",at.row="TIMEPOINT",subjid="SUBJID")  
  
s=split(tab1,variable="TIMEPOINT",at=3)  
  
tab1.1=s$x1  
tab1.2=s$x2  
  
gg=plot(tab1,title="Mean response evolution as a function of time by treatment group",  
legend.label="Treatment groups",ylab="Y mean")  
  
# Qualitative stats (2 explicative variables) #####  
  
tab2=report.quali(data=datafake,y="y_logistic",  
x1="GROUP",x2="TIMEPOINT",at.row="TIMEPOINT",total=TRUE,subjid="SUBJID")  
  
gg2=plot(tab2,title="Response distribution (%) by day and treatment group",  
legend.label="Y levels")  
  
# Qualitative stats (no explicative variable) #####  
  
tab3=report.quali(data=datafake,y="VAR",y.label="Whatever")  
  
# Qualitative stats (no explicative variables ; add number of subjects in header)#  
  
tab4=report.quali(data=datafake,y="VAR",y.label="Whatever",  
subjid="SUBJID")  
  
# Qualitative stats (1 explicative variable)#####  
  
tab5=report.quali(data=datafake,y="VAR",y.label="Whatever",x1="GROUP",  
subjid="SUBJID")  
  
# Quantitative stats (1 explicative variable)#####  
  
tab6=report.quanti(data=datafake,y="y_numeric",y.label="Whatever 2",x1="GROUP",  
subjid="SUBJID")  
  
# Quali-Quanti table  
  
tab5.6=regroup(tab5,tab6)
```

```
# Linear model (order 2 interaction): Anova and LS-Means reporting #####

mod1=lm(y_numeric~baseline+GROUP+TIMEPOINT+GROUP*TIMEPOINT,data=data.mod)
test1=emmeans(mod1,~GROUP|TIMEPOINT)

anov1=Anova(mod1)

tab.mod1=report.lsmmeans(lsm=test1)

gg.mod1=plot(tab.mod1,title="LS-Means response evolution as a function of time\n
by treatment group (95% CI)",
legend.label="Treatment groups",ylab="Y mean",add.ci=TRUE)

# Linear model (1 group only): Anova and LS-Means and graph reporting #####

mod2=lm(y_numeric~baseline+GROUP,data=data.mod)

anov2=Anova(mod2,type=3)

test2=emmeans(mod2,~GROUP)
tab.mod2=report.lsmmeans(lsm=test2)

gg.mod2=plot(tab.mod2,title="LS-Means response\nby treatment group (95% CI)",
legend.label="Treatment groups",ylab="Y mean",add.ci=TRUE)

# Linear mixed model (order 2 interaction):
# Anova and LS-Means and graph reporting #####

mod3=lme(y_numeric~baseline+GROUP+TIMEPOINT+GROUP*TIMEPOINT,
random=~1|SUBJID,data=data.mod,na.action=na.omit)

anov3=Anova(mod3,3)

test3=emmeans(mod3,~GROUP|TIMEPOINT)

tab.mod3=report.lsmmeans(lsm=test3)

gg.mod3=plot(tab.mod3,title="LS-Means response evolution as a function of time\n
by treatment group (95% CI Mixed model)",
legend.label="Treatment groups",ylab="Y mean",add.ci=TRUE)

# Contrast example

contr=contrast(test3, "trt.vs.ctr1", ref = "A")

tab.mod3.contr=report.lsmmeans(lsm=contr)

gg.mod3.contr=plot(tab.mod3.contr,title="LS-Means contrast versus reference A\n
(95% CI Mixed model)",
legend.label="Treatment groups",ylab="Y mean",add.ci=TRUE,add.line=FALSE)
```

```
#####  
# Generalized Logistic Linear model (order 2 interaction):  
#####  
  
# Anova LS-Means and graph reporting #####  
  
mod4=glm(y_logistic~baseline+GROUP+TIMEPOINT+GROUP*TIMEPOINT,  
family=binomial,data=data.mod,na.action=na.omit)  
  
anov4=Anova(mod4,3)  
  
test4=emmeans(mod4,~GROUP|TIMEPOINT)  
  
tab.mod4=report.lsmmeans(lsm=test4,at.row="TIMEPOINT")  
  
gg.mod4=plot(tab.mod4,title="LS-Means response evolution as a function of time\  
by treatment group (95% CI Logistic model)",  
legend.label="Treatment groups",ylab="Y mean",add.ci=TRUE)  
  
# Generalized Poisson Linear model (order 2 interaction):  
# Anova LS-Means and graph reporting #'  
  
mod5=glm(y_poisson~baseline+GROUP+TIMEPOINT+GROUP*TIMEPOINT,  
family=poisson,data=data.mod,na.action=na.omit)  
  
anov5=Anova(mod5,3)  
  
test5=emmeans(mod5,~GROUP|TIMEPOINT)  
  
tab.mod5=report.lsmmeans(lsm=test5,at.row="TIMEPOINT")  
  
gg.mod5=plot(tab.mod5,title="LS-Means response evolution as a function of time\  
by treatment group (95% CI Poisson model)",  
legend.label="Treatment groups",ylab="Y mean",add.ci=TRUE)  
  
#####  
# Create your report  
#####  
  
doc=read_docx()  
doc=body_add_toc(doc)  
  
doc=body_add_par(doc,"A beautiful reporting using ClinReport", style = "heading 1")  
  
doc=body_add_par(doc,"Descriptive statistics", style = "heading 2")
```



```
doc=report.doc(tab1.1,title="Quantitative statistics (2 explicative variables) (Table 1/2)",
colspan.value="Treatment group",doc=doc,init.numbering=TRUE,
page.break=FALSE)

doc=report.doc(tab1.2,title="Quantitative statistics (2 explicative variables) (Table 2/2)",
colspan.value="Treatment group",doc=doc)

doc=body_add_par(doc,"Corresponding graphic of outputs 1 & 2", style ="Normal")

doc=body_add_gg(doc, value = gg, style = "centered" )

doc=body_add_break(doc)

doc=report.doc(tab2,title="Qualitative statistics (2 explicative variables)",
colspan.value="Treatment group",doc=doc)

doc=report.doc(tab2,title="The same with smaller font size",
colspan.value="Treatment group",doc=doc,font.size=8)

doc=body_add_par(doc,"Corresponding graphic of output 3", style ="Normal")

doc=body_add_gg(doc, value = gg2, style = "centered" )

doc=body_add_break(doc)

doc=body_add_par(doc,"Example of mixing qualitative and quantitative
statistics with the function regroup", style ="Normal")

doc=report.doc(tab5.6,title="Quali-Qanti statistics (1 variable only)",doc=doc)

doc=body_add_par(doc,"Statistical model results", style = "heading 2")

doc=body_add_par(doc,"Model 1", style = "heading 3")

doc=body_add_par(doc,"Anova table example", style = "Normal")

doc=report.doc(anov1,doc=doc)

doc=body_add_par(doc,"LS-Means example", style = "Normal")

doc=report.doc(tab.mod1,title="Linear Model LS-Means results using lm with interactions",
colspan.value="Treatment group",doc=doc)

doc=body_add_gg(doc, value = gg.mod1, style = "centered" )

doc=body_add_break(doc)

doc=body_add_par(doc,"Model 2", style = "heading 3")

doc=report.doc(anov2,doc=doc)
```

```
doc=report.doc(tab.mod2,title="Linear Model LS-Means results using lm without interaction",
colspan.value="Treatment group",doc=doc)
```

```
doc=body_add_gg(doc, value = gg.mod2, style = "centered" )
```

```
doc=body_add_break(doc)
```

```
doc=body_add_par(doc,"Model 3", style = "heading 3")
```

```
doc=report.doc(anov3,doc=doc)
```

```
doc=report.doc(tab.mod3,title="Linear Mixed Model LS-Means results using lme",
colspan.value="Treatment group",doc=doc)
```

```
doc=body_add_gg(doc, value = gg.mod3, style = "centered" )
```

```
doc=body_add_break(doc)
```

```
doc=report.doc(tab.mod3.contr,title="LS-Means Contrast example",
colspan.value="Timepoints",doc=doc)
```

```
doc=body_add_gg(doc, value = gg.mod3.contr, style = "centered" )
```

```
doc=body_add_break(doc)
```

```
doc=body_add_par(doc,"Model 4", style = "heading 3")
```

```
doc=report.doc(anov4,doc=doc)
```

```
doc=report.doc(tab.mod4,title="Generalized Linear Mixed Model LS-Means results using glm",
colspan.value="Treatment group",doc=doc)
```

```
doc=body_add_gg(doc, value = gg.mod4, style = "centered" )
```

```
doc=body_add_break(doc)
```

```
doc=body_add_par(doc,"Model 5", style = "heading 3")
```

```
doc=report.doc(anov5,doc=doc)
```

```
doc=report.doc(tab.mod5,title="Poisson Model LS-Means results",
colspan.value="Treatment group",doc=doc)
```

```

doc=body_add_gg(doc, value = gg.mod5, style = "centered" )

file=paste(tempfile(),".docx",sep="")
print(doc, target =file)
shell.exec(file)

```

---

report.lsmeans	<i>'LS Means' statistics reporting</i>
----------------	--

---

### Description

Creates a desc object for "LS Means" statistics reporting.

For more examples see the website: [ClinReport website](#)

### Usage

```

report.lsmeans(lsm, at.row = NULL, infer = c(T, T), round = 2, x1,
  x2, x3, x1.name, x2.name, x3.name, data, contrast, contrast.name, type,
  transpose = FALSE, y.label = NULL)

```

### Arguments

lsm	emmGrid object (result of a emmeans call)
at.row	Character. Passed to spacetable function. Used to space the results per levels of the mentioned variable
infer	A vector of one or two logical values. Passed to summary.emmGrid function.
round	Numeric. Specify the number of digits to round the statistics
x1	deprecated
x2	deprecated
x3	deprecated
x1.name	deprecated
x2.name	deprecated
x3.name	deprecated
data	deprecated
contrast	deprecated
contrast.name	deprecated
type	deprecated
transpose	Logical. If TRUE Statistics will be reported in columns
y.label	Character Indicates the label for y parameter to be displayed in the title of the table

**Details**

You can produce formatted Least Square Means table for up to 3 factors. It doesn't work for quantitative covariates.

See examples below.

**Value**

A desc object that can be used by the `report.doc` function.

**See Also**

[report.quali emmeans](#) [report.doc desc](#)

**Examples**

```
library(emmeans)
library(lme4)

data(datafake)

#Simple lm model

mod=lm(Petal.Width~Species,data=iris)
raw.lsm=emmeans(mod,~Species)
report.lsmeans(raw.lsm)

# You can display the Statistics in columns

report.lsmeans(raw.lsm,transpose=TRUE)

# In case of just one intercept

mod=glm(Species~1,data=iris,family=binomial)
raw.lsm=emmeans(mod,~1)
report.lsmeans(raw.lsm)

# Display statistics in columns

report.lsmeans(raw.lsm,transpose=TRUE)

#Mixed model example using lme4

mod=lmer(y_numeric~GROUP+TIMEPOINT+GROUP*TIMEPOINT+(1|SUBJID),data=datafake)
raw.lsm=emmeans(mod,~GROUP|TIMEPOINT)
report.lsmeans(lsm=raw.lsm,at="TIMEPOINT")

# Display statistics in columns

report.lsmeans(lsm=raw.lsm,at="TIMEPOINT",transpose=TRUE)
```

```

# LM model with specific contrast

warp.lm <- lm(breaks ~ wool+tension+wool:tension, data = warpbreaks)
warp.emm <- emmeans(warp.lm, ~ tension | wool)
contr=contrast(warp.emm, "trt.vs.ctrl", ref = "M")
report.lsmeans(lsm=contr,at="wool")

# Display statistics in columns

report.lsmeans(lsm=contr,at="wool",transpose=TRUE)

# Cox model

library(survival)

data(time_to_cure)

fit <- coxph(Surv(time, status) ~ Group, data = time_to_cure)
em=emmeans(fit,~Group,type="response")
pairs=pairs(em,adjust="none",exclude="Untreated")
pairs

report.lsmeans(pairs)

# Display statistics in columns

report.lsmeans(pairs,transpose=TRUE)

```

---

```
report.modelinfo      Export model information (still experimental)
```

---

## Description

report.doc This function enables to export the information of the model (the package, the name of the function, the call etc...)

## Usage

```

report.modelinfo(object, ...)

## S3 method for class 'lme'
report.modelinfo(object, doc = NULL, page.break = TRUE,
  ...)

## S3 method for class 'lmerMod'

```

```
report.modelinfo(object, doc = NULL,
  page.break = TRUE, ...)

## S3 method for class 'glm'
report.modelinfo(object, doc = NULL, page.break = TRUE,
  ...)

## S3 method for class 'coxph'
report.modelinfo(object, doc = NULL, page.break = TRUE,
  ...)
```

### Arguments

object	A model (for now glm, lme, lmer and coxph models are available)
...	Other arguments
doc	NULL or a rdocx object
page.break	Logical. If TRUE it adds a page break after the output. Default to TRUE

### Details

Compatible only (for now) with GLM, LME and Cox models. For now those output are not numbered.

### Value

A flextable object (if doc=NULL) or a rdocx object (if doc= an rdocx object).

### See Also

[report.doc](#)

### Examples

```
library(officer)
library(flextable)
library(nlme)
library(lme4)

data(datafake)

#Model info for lme model

mod=lme(y_numeric~GROUP+TIMEPOINT,random=~1|SUBJID,data=datafake,na.action=na.omit)

# Show in HTML (can be inserted in an R markdown or a MS Word document)

report.modelinfo(mod)

#Model info for lmer model
```

```
mod=lmer(y_numeric~GROUP+TIMEPOINT+(1|SUBJID),data=datafake,na.action=na.omit)

report.modelinfo(mod)
```

---

report.quali	<i>Descriptive "Qualitative" statistics (frequencies and percentages) reporting</i>
--------------	---

---

### Description

Compute and report frequencies and percentages by levels of y (in rows) and by levels of x1 (in columns) and x2 in rows.

For more examples see the website: [ClinReport website](#)

### Usage

```
report.quali(data, y = NULL, x1 = NULL, x2 = NULL, y.label = y,
  x2.label = NULL, y.levels.label = "Levels", total = F, round = 2,
  at.row = NULL, percent.col = T, subjid = NULL, remove.zero = F,
  drop.y = NULL, drop.x1 = NULL, drop.x2 = NULL,
  remove.missing = F)
```

### Arguments

data	a data.frame object
y	Character indicating a factor in the data (the response)
x1	Character indicating a factor in the data (levels will be displayed in columns)
x2	Character indicating a factor in the data (levels will be displayed in rows). Only possible if x1 is not NULL.
y.label	Character indicating the label for y to be displayed in the title of the table
x2.label	Character indicating the label for x2 parameter, only if x2 is not null
y.levels.label	Character. Indicates the label of the column in which the levels of y are displayed
total	Logical Indicates if a column Total should be added or not
round	Integer Indicates the number of digits to round percentages
at.row	Character. Pass to spacetable function. Used to space the results (see example below)
percent.col	Logical By default it is set to T to indicate that column percentages should be reported. If set to False, row percentages are reported.
subjid	Character Indicates in the data.frame the name of the column used to identify the Id of the subjects. If not null, it adds in the headers the number of unique subject per levels of x1 or y (if x1 is null).

remove.zero	Logical. Remove the levels in the contingency table for which there is no observation.
drop.y	Character. Indicates one or several levels of the y factor that you want to drop in the result
drop.x1	Character. Indicates one or several levels of the x1 factor that you want to drop in the result
drop.x2	Character. Indicates one or several levels of the x2 factor that you want to drop in the result
remove.missing	Logical. default to TRUE. If TRUE number of missing values are reported and percentages take into account the number of missing value in the calculation. If set to FALSE, the missing values regarding the response factor y are ignored and percentages are computed on non missing values only.

### Details

This function computes and reports qualitative statistics by level of y and by level of x1 (if not null) and x2 (if not null).

See examples to show the results. If total=T, the last column is the statistics performed overall levels of the explicative variables x1.

Note that missing values are counted in the calculation of the percentages.

### Value

A desc object

### See Also

[report.quant](#) [report.doc](#) [desc](#)

### Examples

```
data(datafake)

# No explicative variable changing y labels (option y.levels.label)
report.quali(data=datafake,y="y_logistic",
y.label="Clinical cure",y.levels.label="Levels")

# 1 explicative variable (option x1: levels in columns)
report.quali(data=datafake,y="y_logistic",x1="GROUP",y.levels.label="Clinical cure")

# 2 explicative variables (x1, levels in columns, option x2, levels in rows),
# spacing the results (option at.row)
report.quali(data=datafake,y="y_logistic",x1="GROUP",
x2="TIMEPOINT",y.levels.label="Clinical cure",x2.label="Days",at.row="Days")

# 2 explicative variables with row percentages (option percent.col=F)
report.quali(data=datafake,y="y_logistic",x1="GROUP",
```



```

x2="TIMEPOINT",percent.col=FALSE,x2.label="Days",y.levels.label="Clinical cure")

# Add Total number of subjects in headers (option subjid="SUBJID")
report.quali(data=datafake,y="y_logistic",x1="GROUP",
x2="TIMEPOINT",at.row="TIMEPOINT",subjid="SUBJID")

# Round percentages to 1 digit (option round=1)
report.quali(data=datafake,y="y_logistic",x1="GROUP",
x2="TIMEPOINT",at.row="TIMEPOINT",subjid="SUBJID",round=1)

# Qualitative statistics with a response with more than one levels
tab=report.quali(data=datafake,y="y_poisson",x1="GROUP",
x2="TIMEPOINT",at.row="TIMEPOINT",subjid="SUBJID",round=1)

# Print formatted results
tab

#Getting raw output (unformatted)
tab$raw.output

```

---

report.quali.hlev      *'Hierarchical Qualitative' statistics reporting (experimental)*

---

## Description

This function is mainly used to compute qualitative statistics when there are several events per statistical unit. Often used for reporting adverse events, medical history or concomitant treatments.

It reports frequencies and percentages according to hierarchical levels of two factors.

Typically, adverse event are classified according to System Organ Class (SOC) and then sub classified by Preferred Terms (PT). Several observations of a same adverse event can be observed several times on the same subject. It's then useful to know how many persons are concerned by at least one of those adverse events and report the frequencies for each classifications: SOC and PT.

This is exactly what this function does.

For more examples see the website: [ClinReport website](#)

## Usage

```

report.quali.hlev(data, subjid = NULL, x1 = NULL, var_upper, var_lower,
  lower.levels = "Lower.Levels", upper.levels = "Upper.Levels",
  x1.label = NULL)

```

## Arguments

data	A data frame
subjid	A character

x1	A character. Indicates a factor in the data frame.
var_upper	A character. Indicates a factor in the data frame which corresponds to the factor with the higher number of levels (typically SOC variable)
var_lower	A character. Indicates a factor in the data frame which corresponds to the factor with the lower number of levels (typically PT variable)
lower.levels	A character. The label to be displayed in the table for the lower terms
upper.levels	A character. The label to be displayed in the table for the upper terms
x1.label	A character. Not used for now

### Details

The subjid argument is mandatory for this function.

### Value

A desc object that can be used by the report.doc function.

### See Also

[report.quali emmeans report.doc desc](#)

### Examples

```
data(adverse_event)

test=report.quali.hlev(data=adverse_event,subjid="SUBJID",var_upper="PTNAME",
var_lower="SOCNAME",lower.levels="System Organ Class",upper.levels="Preferred Terms",x1="randtrt")

# show results in console
test

# show formatted results in HTML
ft=report.doc(test, valign=TRUE)
ft
```

---

report.quant

*Descriptive "Quantitative" statistics (mean, SD, median...) reporting*

---

### Description

report.quant Returns quantitative descriptive statistics such as mean, median, standard deviation etc...

For more examples see the website: [ClinReport website](#)

**Usage**

```
report.quanti(data, y, x1 = NULL, x2 = NULL, y.label = y,
  round = 2, total = F, scientific = F, digits = NULL,
  at.row = NULL, subjid = NULL, geomean = F, add.mad = F,
  default.stat = T, func.stat = NULL, stat.name = "Statistics",
  func.stat.name = "", drop.x1 = NULL, drop.x2 = NULL)
```

**Arguments**

data	Data.frame object
y	Character indicating a numerical vector in the data frame passed to data argument
x1	Character indicating a factor in the data (levels will be displayed in columns)
x2	Character indicating a factor in the data (levels will be displayed in lines)
y.label	Character Indicates the label for y parameter to be displayed in the title of the table
round	Numeric to indicate how to round statistics
total	Logical to indicate if a "Total" column should be added
scientific	Logical Indicates if statistics should be displayed in scientific notations or not
digits	Numeric (used if scientific=TRUE) to indicate how many digits to use in scientific notation
at.row	Character Used to space the results (see examples)
subjid	Character Indicates the column in which there is the subject Id to add the number of subjects in the column header if x1 and x2 are not null.
geomean	Logical If yes geometric mean is calculated instead of arithmetic mean: $\exp(\text{mean}(\log(x)), \text{na.rm}=\text{TRUE})$ for $x>0$
add.mad	Logical If yes the Median Absolute Deviance is added to the median statistics (see function <a href="#">mad</a> )
default.stat	Logical (default to TRUE). If FALSE you can specify your own example
func.stat	Function. If specified then default.stat=FALSE and only the specified statistic is reported
stat.name	Character. Indicates the name of the variable that report the statistics Default = "Statistics"
func.stat.name	Character. Used only if default.stat=FALSE. Indicates the name of specific statistic you want to report
drop.x1	Character. Indicates one or several levels of the x1 factor that you want to drop in the result
drop.x2	Character. Indicates one or several levels of the x2 factor that you want to drop in the result

## Details

This function computes and reports quantitative statistics on  $y$ . And can gives the statistics by level of two factors ( $x_1$  in columns and/or  $x_2$  in rows). See the example to show the results. If `total=TRUE`, the last column is the statistics performed overall levels of  $x_1$  for each levels of  $x_2$ . Quantiles are calculated using type 3 (SAS presumed definition) algorithms, but even though, some differences between SAS and R can appear on quantile values.

"`geomean`" compute the geometric mean defined as  $\exp(\text{mean}(\log(y)))$ . The values below or equal 0 are removed and a message is printed to indicate how many values were deleted to calculate the geometric mean.

`N` returns the number of observations (including NA values)

`stat.name` is automatically transformed using `make.names` function.

## Value

A desc object.

## See Also

[report.quali](#) [report.doc desc](#)

## Examples

```
data(datafake)

# Quantitative statistics with no factor
report.quanti(data=datafake,y="y_numeric",total=TRUE,y.label="Awesome results")

#' # Quantitative statistics with no factor with geometric mean (option geomean=TRUE)
report.quanti(data=datafake,y="y_numeric",y.label="Awesome results",geomean=TRUE)

# Quantitative statistics with one factor
report.quanti(data=datafake,y="y_numeric",x1="GROUP")

# One factor with total column
report.quanti(data=datafake,y="y_numeric",x1="GROUP",total=TRUE)

# Quantitative statistics with two factors
report.quanti(data=datafake,y="y_numeric",x1="GROUP",x2="TIMEPOINT")

# Quantitative statistics with two factors and a total column
report.quanti(data=datafake,y="y_numeric",x1="GROUP",x2="TIMEPOINT",total=TRUE)

# Add median absolute deviance to the median statistics
```

```
report.quanti(data=datafake,y="y_numeric",x1="GROUP",x2="TIMEPOINT",total=TRUE,add.mad=TRUE)

# Quantitative statistics with spacing rows (option at.row)

report.quanti(data=datafake,y="y_numeric",x1="GROUP",
x2="TIMEPOINT",total=TRUE,at.row="TIMEPOINT")

# Add number of subjects in headers (option subjid)

tab=report.quanti(data=datafake,y="y_numeric",x1="GROUP",
x2="TIMEPOINT",total=TRUE,at.row="TIMEPOINT",subjid="SUBJID")

# Print tab output
tab

#Getting a specific statistic and not the default ones

mystat=function(x) quantile(x,0.99,na.rm=TRUE)

tab=report.quanti(data=datafake,y="y_numeric",x1="GROUP",
x2="TIMEPOINT",total=TRUE,subjid="SUBJID",
func.stat=mystat,func.stat.name="99% quantile")
tab

mystat2=function(x) mean(x,na.rm=TRUE)/sd(x,na.rm=TRUE)

tab=report.quanti(data=datafake,y="y_numeric",x1="GROUP",
total=TRUE,subjid="SUBJID",func.stat=mystat2,
func.stat.name="Coefficient of variation")
tab

mode=function(x) {
  x=na.omit(x)
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

tab=report.quanti(data=datafake,y="y_numeric",
func.stat=mode,func.stat.name="Mode")

#Getting raw output
tab$raw.output

#Getting a data.frame version of the output
tab$output
```

---

report.sessionInfo     *Export the table of the sessionInfo() to Word or R markdown documents*

---

## Description

report.sessionInfo

This function enables to export the table of the [sessionInfo](#)

## Usage

```
report.sessionInfo(doc = NULL)
```

## Arguments

doc                    NULL or a rdocx object'

## Details

None

## Value

A flextable object (if doc=NULL) or a rdocx object (if doc= an rdocx object).

## See Also

[report.doc](#) [report.modelinfo](#)

## Examples

```
# For an R markdown document just do
report.sessionInfo()

# Fro a Word document use the doc argument
library(officer)
doc=read_docx()
doc=report.sessionInfo(doc)
```

---

spacetable	<i>Add space to the results of a statistic table</i>
------------	--

---

## Description

spacetable Add space at some interval in a table of statistics

## Usage

```
spacetable(tab, at.row = 6)
```

## Arguments

tab	a data.frame. Normally, the output part of a desc object
at.row	Numeric (between 1 and <code>ncol(tab)</code> ) or Character (a column name of tab)

## Details

Used internally into `report.lsmmeans`, `report.quantile` and `report.quali`

## Value

A data.frame with some space between levels of the variable specified by `at.row`

## See Also

[report.quantile](#) [report.quali](#) [report.lsmmeans](#)

## Examples

```
data(datafake)
spacetable(report.quali(data=datafake,y="y_logistic",x1="GROUP",
x2="TIMEPOINT")$output,at.row="TIMEPOINT")
```

space\_vline

*Compute the indices where to put blanks in report.doc*

---

**Description**

Compute the indices where to put blanks in report.doc

**Usage**

```
space_vline(tab, at.row)
```

**Arguments**

tab	data.frame from a desc object
at.row	Character. at.row option of a desc object

**Details**

None

**Value**

A vector of integers

**See Also**

[report.doc](#)

---

split.desc*Split a table of statistics in two.*

---

**Description**

Split a table of statistics in two.

**Usage**

```
## S3 method for class 'desc'  
split(x, f, drop, ..., variable, at)
```



**Arguments**

x	A desc object
f	see split documentation
drop	see split documentation
...	Other parameters
variable	Character. Indicates the variable to use for the split
at	Integer. Indicates the maximum number of levels to include in the first splitted table

**Details**

Used for splitting an output in two. It can be used for example when the table takes too much space in a page of a Word document (see the example below).

**See Also**

[desc](#)

**Examples**

```
data(datafake)

tab1=report.quantile(data=datafake,y="y_numeric",
x1="GROUP",x2="TIMEPOINT",at.row="TIMEPOINT",subjid="SUBJID")

s=split(tab1,variable="TIMEPOINT",at=3)

tab1.1=s$x1
tab1.2=s$x2

tab1.1
tab1.2
```

---

time\_to\_cure

*time\_to\_cure*

---

**Description**

Simulated survival data created only for the purpose of testing the package and showing examples.

**Usage**

```
data(time_to_cure)
```

**Format**

A data frame #'

**time** Fake numerical time values

**status** Fake status values, 1=cured 0=not cured

**block** Fake factor

**random\_number** Fake factor

**Sex** Fake sex factor

**Weight\_D0** Fake weight covariate

**Pen** Fake factor

**TCS** Fake covariate

**Group** Fake treatment group factor

**Subjid** Fake subject Id

---

transpose	<i>Transpose the table of statistics (experimental)</i>
-----------	---

---

**Description**

Used to get the statistics in column for example.

For more examples see the website: [ClinReport website](#)

**Usage**

```
transpose(desc, ...)
```

```
## S3 method for class 'desc'
```

```
transpose(desc, ...)
```

**Arguments**

desc	a desc object
...	Not used

**Details**

None

**Value**

A desc object

**See Also**

[report.quant](#) [report.doc](#) [desc](#)

**Examples**

```
library(reshape2)

data(datafake)

desc=report.quanti(data=datafake,y="y_numeric",x1="GROUP",
x2="TIMEPOINT",total=TRUE,at.row="TIMEPOINT",subjid="SUBJID")

transpose(desc)

desc=report.quanti(data=datafake,y="y_numeric",x1="GROUP")

transpose(desc)

desc=report.quanti(data=datafake,y="y_numeric")

transpose(desc)

desc=report.quali(data=datafake,y="y_logistic",x1="GROUP")

transpose(desc)

desc=report.quali(data=datafake,y="y_logistic",x1="GROUP",x2="TIMEPOINT")

transpose(desc)

desc=report.quali(data=datafake,y="y_logistic")

transpose(desc)
```

# Index

## \*Topic **datasets**

- adverse\_event, 4
  - datafake, 6
  - time\_to\_cure, 41
- add.stat, 2
- adverse\_event, 4
- at.least.one, 5
- check.x, 5
- ClinReport, 6
- ClinReport-package (ClinReport), 6
- datafake, 6
- define.key, 7
- desc, 8, 9, 12, 13, 15, 18, 21, 28, 32, 34, 36, 41, 42
- dimnames.desc, 9
- emmeans, 28, 34
- font, 20
- fontsize, 20
- gg\_desc\_lsmeans, 10, 11, 12, 15
- gg\_desc\_quali, 10, 11, 11, 12, 15
- gg\_desc\_quanti, 10, 11, 12, 15
- is.desc, 13
- mad, 35
- make.label, 13
- make.names, 36
- plot.desc, 10–12, 14
- pretty, 16
- prettyround, 17
- print.desc, 17
- regroup, 3, 18
- regroup.desc, 9
- report.doc, 9, 18, 19, 28, 30, 32, 34, 36, 38, 40, 42
- report.lsmeans, 9, 10, 19, 21, 27, 39
- report.modelinfo, 29, 38
- report.quali, 9, 11, 18, 19, 21, 28, 31, 34, 36, 39
- report.quali.hlev, 33
- report.quanti, 3, 9, 18, 19, 21, 32, 34, 39, 42
- report.sessionInfo, 37
- sessionInfo, 38
- space\_vline, 40
- spacetable, 8, 39
- split.desc, 40
- time\_to\_cure, 41
- transpose, 42