

Package ‘dprint’

October 13, 2022

Type Package

Title Print Tabular Data to Graphics Device

Version 0.0.4

Date 2018-01-08

Author Carlin Brickner, Rocco Napoli

Maintainer Carlin Brickner <carlin.brickner.rpackage@gmail.com>

Depends R (>= 3.0)

Imports grid, stats, grDevices

Description Provides a generalized method for printing tabular data within the R environment in order to make the process of presenting high quality tabular output seamless for the user. Output is directed to the R graphics device so that tables can be exported to any file format supported by the graphics device. Utilizes a formula interface to specify the contents of tables often found in manuscripts or business reports. In addition, formula interface provides inline formatting of the numeric cells of a table and renaming column labels.

License GPL (>= 2)

LazyLoad yes

RoxygenNote 6.0.1

Suggests testthat

Repository CRAN

Repository/R-Forge/Project tabular

Repository/R-Forge/Revision 86

Repository/R-Forge/DateTimeStamp 2018-01-08 05:23:24

Date/Publication 2018-01-08 15:18:38 UTC

NeedsCompilation no

R topics documented:

char.dim	3
char.height	3
char.width	4
char.width1	4
colnames.linebreak	5
colnames.row	5
colnames.struct	5
conditional.struct	6
consect.struct	6
dborder	7
dprint	7
dprint.data.frame	12
dprint_simp	16
Fb	17
Fc	17
Fci	18
fitpage	18
fmla_inter	19
fmla_lhs	19
fmla_rhs	20
Fr	20
frmt	21
ft	22
hdr	23
insert.grp	24
insert.grp1	24
kill.multiregx	25
list.to.df	25
pagelayout	26
rbin	26
rdesc	27
row.hl	27
size.simp	28
style	28
tablebreak	29
tbl.struct	30
tbl.struct.simp	30
vector.linebreak	31
vector.struct	31

char.dim	<i>Character Dimensions of Table</i>
----------	--------------------------------------

Description

Given the table object and style object, will return the maximum width of columns, the height of characters in the units specified by user

Usage

```
char.dim(obj, style, cx = 1)
```

Arguments

obj	Table Object
style	style
cx	character expansion

char.height	<i>Character Height</i>
-------------	-------------------------

Description

Returns height in units(inches) based on a gpar setting

Usage

```
char.height(charact = "A", frmt, cx = 1)
```

Arguments

charact	Table Object
frmt	format object
cx	shinkage parameter

char.width	<i>Character Width</i>
------------	------------------------

Description

Returns the max width of all the elements in a vector in units, handles data types.

Usage

```
char.width(obj, frmt, cx = 1)
```

Arguments

obj	Table Object
frmt	format object
cx	shinkage parameter

char.width1	<i>Character Width</i>
-------------	------------------------

Description

Returns the max width in units, given gpar settings

Usage

```
char.width1(vctr, frmt, cx = 1)
```

Arguments

vctr	vector
frmt	format object
cx	shrinkage parameter

colnames.linebreak *Column Names Line Break*

Description

Inserts a line break in column names when the escape character [backslash] n is found.

Usage

```
colnames.linebreak(colnames.obj)
```

Arguments

colnames.obj column names object

colnames.row *Column Names Row*

Description

Adjust index (reference number of rows above table for colnames) to account for line breaks

Usage

```
colnames.row(colnames.obj)
```

Arguments

colnames.obj column names object

colnames.struct *Column Names Structure*

Description

Creates a structure for printing column names and their hierachies

Usage

```
colnames.struct(col.names, linebreak = TRUE)
```

Arguments

col.names vector of strings (result of colnames(data.frame))
linebreak boolean

`conditional.struct` *Conditional Structure*

Description

For variables defined as conditional variables in the formula, function to drive creating a list of simple table structures

Usage

```
conditional.struct(data, byvars)
```

Arguments

`data` `data.frame`

`byvars` variables in addition to the group and label variables listed in the formula

`consect.struct` *Consecutive Structure*

Description

Given a vector returns several objects describing how(if) consecutive elements appear. This is a helper function for `tbl.struct` and is important in defining the column hierarchy.

Usage

```
consect.struct(vct)
```

Arguments

`vct` `vector`

dborder	<i>Draw Border</i>
---------	--------------------

Description

Draws a border style around a section of table

Usage

```
dborder(cord1, cord2, frmt)
```

Arguments

cord1	Vector (x,y) indicating position of top left point of rectangle
cord2	Vector (x,y) indicating position of bottom right point of rectangle
frmt	Format

dprint	<i>Print Table to Graphics Device (dprint)</i>
--------	------------------------------------------------

Description

Prints tabular data to the graphics device by translating an R object to a tabular presentation.

Usage

```
dprint(data, fmla = NULL, label = NULL, group = NULL, regx = NA,
       style = NULL, main = NA, footnote = NA, dtype = "rgraphics",
       pg.dim = NULL, margins = NULL, showmargins = FALSE, row.hl = NULL,
       fit.width = FALSE, fit.height = FALSE, fit = FALSE, newpage = FALSE,
       center.horz = FALSE, center.vert = FALSE, center = FALSE,
       f.hdr = NULL, f.ftr = NULL, pagenum = NULL, lastcall = NULL)
```

Arguments

data	An object of class found among methods(dprint)
fmla	An object of class "formula". Formula interface is used to describe the properties of tabular data to be printed from the <i>data</i> object.
label	Character vector of length 1 used to reference the name of column containing row labels. Optional to <i>fmla</i> . Set to NULL to when using <i>fmla</i> or when no row labels exist. Default value is NULL.
group	Character vector of length 1 used to reference the name of column containing grouping of row labels. Optional to <i>fmla</i> . Set to NULL to when using <i>fmla</i> or when no row labels exist. Default value is NULL.

regx	Character vector of length 1 used to provide regular expression(s) to remove unwanted text displayed from original column names (e.g. merge applied with .x and .y appended to duplicate column names)
style	Style sheet object used to define font and other settings of the table. See style and frmt
main	Table title defined by character vector of length 1. String will be placed on top of table
footnote	Footnote defined by character vector finite length. The text will be printed immediately underneath the tabular presentation. Each position in the vector will force a new line break.
dtype	Named references to preset <i>pg.dim</i> settings. Graphics device type referred to by names, sets default page settings. Device types, currently available "rdevice", "portrait", "landscape" which sets <i>pg.dim</i> to c(8,8), (11, 8.5) and (8.5, 11)
pg.dim	A vector of c(height,width) units used to describe the dimensions of a custom page and over ride dtype. When printing to a multiple page pdf with custom dimensions, dtype should be set to some character other than "rdevice" (i.e. "custom") because dev.new() will be used to start a new window instead of grid.text().
margins	A numerical vector of the form c(bottom, left, top, right) which gives the margin size specified in inches. Other declarations assume a constant for all margins or c(top/bottom,left/right)
showmargins	Boolean, displays margins on R device. Useful for tinkering with presentation
row.hl	Conditional highlight row highlight object. See row.hl
fit.width	Boolean. If TRUE, forces the table to fit the table horizontally within the <i>pg.dim</i> and <i>margins</i> . Exactly fits when vector formats are used, approximation otherwise.
fit.height	Boolean. If TRUE, forces the table to fit the table vertically within the <i>pg.dim</i> and <i>margins</i> . Exactly fits when vector formats are used, approximation otherwise.
fit	Boolean. If TRUE, forces the table to fit both horizontally and vertically within the <i>pg.dim</i> and <i>margins</i> .
newpage	Boolean. If TRUE, when the presentation of the table runs out of space on the current page, within the <i>pg.dim</i> and <i>margins</i> , a new page will automatically be started. Designed for multiple page pdf reports.
center.horz	Boolean, If TRUE, center table horizontally
center.vert	Boolean, If TRUE, center table vertically. Only available for single table.
center	Boolean, If TRUE, center both vertically and horizontally. Does not consider the fit.* parameters
f.hdr	Pass a function for printing header. See examples and hdr
f.ftr	Pass a function for printing footer. See examples and ftr
pagenum	Starting page number, will override page number from last call
lastcall	Object returned from last call from dprint. Can use this as reference for where a second table should be presented on the same device as the previous call. dprint continues printing to device with fixed separation between tables

Details

The available method functions for dprint are given by methods(dprint).

Formula Interface: Operator Definitions

- “~” separates row versus column definitions (LHS vs RHS of equation)
- Left Hand Side (LHS) - row attributes
 - “+” delimits up to two row labeling hierarchies (group + level)
- Right Hand Side (RHS) - column attributes
 - “+” delimits column placement
 - “:” spanning attribute of a columns
 - “.” all columns in data frame should be included
 - “_” drops following column
 - “|” list variables (delimited by "+") to condition on when print multiple simple tables

Formula Interface: Embedded Functions

- Any algebraic manipulation available in [formula](#) through [I](#) is also available here
- Formatting
 - Rounding & place holders: [Fr](#), [Fci](#), [Fc](#), and [Fb](#)
 - Use [paste](#) to concatenate one of more fields with other text
- Renaming data frame column names to presentable labels. Names default to column names otherwise
 - `Rn(column name, "Presentation Label")`

“\n” can be used in *main* or *footnote* paramters or embedded function *Rn()* to force additional line breaks

Author(s)

Carlin Brickner

Examples

```
### Generate Sample Data Structures ###
# Generate some example data frames
table1 <- rdesc(4, 5)          # Numeric
table1f <- rdesc(4, 5, rnd=TRUE) # Rounded and pretty format so values are character
table1a <- table1; table1b<-table1;
table1a$group2 <- 1; table1b$group2 <- 2;
table2 <- rbind(table1a, table1b)
table2a <- table2; table2b<-table2
table2a$group3 <- "Zebra"; table2b$group3 <- "Elephant";
table3 <- rbind(table2a, table2b)
# Create style object
CBs <- style(frmt.bdy=frmt(fontfamily="HersheySans"), frmt.tbl=frmt(bty="o", lwd=1),
            frmt.col=frmt(fontfamily="HersheySans", bg="khaki", fontface="bold", lwd=2, bty="_"),
            frmt.grp=frmt(fontfamily="HersheySans", bg="khaki", fontface="bold"),
```

```

frmt.main=frmt(fontfamily="HersheySans", fontface="bold", fontsize=12),
frmt.ftn=frmt(fontfamily="HersheySans"),
justify="right")

# dev.new()# All variables, no group or label
dprint(~., data=table1f)
dev.off()
# dev.new() # Spanning, group level, and apply control and treatments to hierchaies on right
dprint(group+level~Control:(Mean1 + Median1 + Variance1) +
  Treatment:(Mean2 + Median2 + Variance2) + p.value, data=table1f)
dev.off()
# dev.new(); #Illegal Names, remove expression
dprint(group+level~`This is a Control`: (Mean1 + Median1 + Variance1) +
  Treatment.y:(Mean2 + Median2 + Variance2), data=table1f, regx="1|2|.y")
dev.off()
# dev.new(); #Illegal Names, no group label
dprint( ~ `This is a Control`: (Mean1 + Median1 + Variance1) +
  Treatment.y:(Mean2 + Median2 + Variance2), data=table1f, regx="1|2|.y")
# dev.new(); # all on rhs with exception of p.value
dev.off()
dprint(group+level~.-p.value, data=table1f)
dev.off()
## Not run:
# dev.new();
dprint(fmla=group+level~., data=table1)
dev.off()
# dev.new()
dprint(fmla=group+level~Rn(round(Mean1, 2), "Mean Trt")+Rn(round(Variance1,2), "Variance"),
  data=table1)
dev.off()
# dev.new()
dprint(group+level~Rn(round(Mean1, 2), "Mean Trt")+
  Variance1+Rn(round(I((Mean1+Mean2)/2),2), "Average of Averages"), data=table1, main="Don't Do this")
dev.off()
# dev.new()
dprint(level~.|group2, data=table2)
dev.off()
# dev.new();
dprint(level~Mean1+Median2|group2, data=table2, main="Descriptives")
dev.off()
# dev.new(); # Spanning, embedded fuctions, and conditional
dprint(group+level~Treatment:Rn(paste(round(Mean1, 2),"(", round(Variance1, 2),")"),
  "Mean Trt (Std)")|group2, data=table2)
dev.off()
# dev.new(); # Spanning, embedded fuctions, and conditional
dprint(~Treatment:Rn(paste(round(Mean1, 2),"(", round(Variance1, 2),")"),
  "Mean Trt (Std)")|group2, data=table2)
# dev.new(); # Spanning, embedded fuctions, and conditional
dev.off()
dprint(~Treatment:(Rn(paste(round(Mean1, 2),"(", round(Variance1, 2),")"), "Mean Trt (Std)")+
  Rn(round(Median1,2), "Median"))|group2, data=table2)
dev.off()
# dev.new()

```

```

dprint(~Treatment:Rn(paste(round(Mean1, 2),"(", round(Variance1, 2),")"), "Mean Trt (Std)")+
Control:Rn(paste(round(Mean2, 2),"(", round(Variance2, 2),")"), "Mean Trt (Std)")|group2,
  data=table2)
dev.off()

f1 <- group+level~Treatment:Rn(Fc(Mean1, Variance1), "Mean (Std)")+
Control:Rn(Fc(Mean2, Variance2), "Mean (Std)") + Rn(round(p.value,2), "P-value")
# dev.new()
dprint(fmla=f1, data=table1,margins=.2, main="Justify Center")
dev.off()
# dev.new()
dprint(fmla=f1, data=table1,margins=.2, main="Justify Right",
style=style(justify="right", frmt.tbl=frmt(bty="o")))
dev.off()
# dev.new()
dprint(fmla=f1, data=table1,margins=.2, main="Justify Left",
style=style(justify="left", frmt.tbl=frmt(bty="o")))
dev.off()

h <- expression(hdr("Test Header",
pagelayout.obj=pagelayout(dtype="rgraphics", margins=c(1, .5))))
f <- expression(ftr("R Package tabular",
pagelayout.obj=pagelayout(dtype="rgraphics", margins=c(1.25, 1, 1.25,1)),
pagenum=eval.parent(pagenum, 1)))
# dev.new()
dprint(fmla=f1, data=table1,margins=c(1.25, 1, 1.25,1), showmargins=TRUE, main="Table Left",
  style=style(justify="left", frmt.tbl=frmt(bty="o"),
  frmt.bdy=frmt(linespace=1.5, bty="X")),
  f.hdr = h, f.ftr=f, pagenum=1)
dev.off()

# dev.new()
dprint(fmla=f1, data=table1,margins=c(1.25, 1, 1.25,1), showmargins=TRUE, main="Table Left",
  style=CBs,
  f.hdr = h, f.ftr=f, pagenum=1)

dev.new()
by_var_f1 <- level~Mean1+Median1|group
by_var_f2 <- level~Mean1+Median1|group+group2
# If main is default (null) than do not print titles
dprint(fmla=by_var_f1, data=table2)
dev.off()
# dev.new()
# When a title is defined, and only one conditional variable is defined, just print the values
# concatenated to the text
dprint(fmla=by_var_f1, data=table2,main=" ")
dev.off()
# dev.new()
# When more than one conditional variable, concatenate the variable name and the
# current combination of values
dprint(fmla=by_var_f2, data=table2,main="Descriptives for: ")

## End(Not run)

```

dprint.data.frame *Print Table to Graphics Device (dprint)*

Description

Prints tabular data to the graphics device by translating an R object to a tabular presentation.

Usage

```
## S3 method for class 'data.frame'
dprint(data, fmla = NULL, label = NULL, group = NULL,
       regx = NA, style = NULL, main = NA, footnote = NA,
       dtype = "rgraphics", pg.dim = NULL, margins = NULL,
       showmargins = FALSE, row.hl = NULL, fit.width = FALSE,
       fit.height = FALSE, fit = FALSE, newpage = FALSE, center.horz = FALSE,
       center.vert = FALSE, center = FALSE, f.hdr = NULL, f.ftr = NULL,
       pagenum = NULL, lastcall = NULL)
```

Arguments

data	An object of class found among methods(dprint)
fmla	An object of class "formula". Formula interface is used to describe the properties of tabular data to be printed from the <i>data</i> object.
label	Character vector of length 1 used to reference the name of column containing row labels. Optional to <i>fmla</i> . Set to NULL to when using <i>fmla</i> or when no row labels exist. Default value is NULL.
group	Character vector of length 1 used to reference the name of column containing grouping of row labels. Optional to <i>fmla</i> . Set to NULL to when using <i>fmla</i> or when no row labels exist. Default value is NULL.
regx	Character vector of length 1 used to provide regular expression(s) to remove unwanted text displayed from original column names (e.g. merge applied with .x and .y appended to duplicate column names)
style	Style sheet object used to define font and other settings of the table. See style and frmt
main	Table title defined by character vector of length 1. String will be placed on top of table
footnote	Footnote defined by character vector finite length. The text will be printed immediately underneath the tabular presentation. Each position in the vector will force a new line break.
dtype	Named references to preset <i>pg.dim</i> settings. Graphics device type referred to by names, sets default page settings. Device types, currently available "rdevice", "portrait", "landscape" which sets <i>pg.dim</i> to c(8,8), (11, 8.5) and (8.5, 11)

pg.dim	A vector of c(height,width) units used to describe the dimensions of a custom page and over ride dtype. When printing to a multiple page pdf with custom dimensions, dtype should be set to some character other than "rdevice" (i.e. "custom") because dev.new() will be used to start a new window instead of grid.text().
margins	A numerical vector of the form c(bottom, left, top, right) which gives the margin size specified in inches. Other declarations assume a constant for all margins or c(top/bottom,left/right)
showmargins	Boolean, displays margins on R device. Useful for tinkering with presentation
row.hl	Conditional highlight row highlight object. See row.hl
fit.width	Boolean. If TRUE, forces the table to fit the table horizontally within the <i>pg.dim</i> and <i>margins</i> . Exactly fits when vector formats are used, approximation otherwise.
fit.height	Boolean. If TRUE, forces the table to fit the table vertically within the <i>pg.dim</i> and <i>margins</i> . Exactly fits when vector formats are used, approximation otherwise.
fit	Boolean. If TRUE, forces the table to fit both horizontally and vertically within the <i>pg.dim</i> and <i>margins</i> .
newpage	Boolean. If TRUE, when the presentation of the table runs out of space on the current page, within the <i>pg.dim</i> and <i>margins</i> , a new page will automatically be started. Designed for multiple page pdf reports.
center.horz	Boolean, If TRUE, center table horizontally
center.vert	Boolean, If TRUE, center table vertically. Only available for single table.
center	Boolean, If TRUE, center both vertically and horizontally. Does not consider the fit.* parameters
f.hdr	Pass a function for printing header. See examples and hdr
f.ftr	Pass a function for printing footer. See examples and ftr
pagenum	Starting page number, will override page number from last call
lastcall	Object returned from last call from dprint. Can use this as reference for where a second table should be presented on the same device as the previous call. dprint continues printing to device with fixed separation between tables

Details

The available method functions for dprint are given by methods(dprint).

Formula Interface: Operator Definitions

- “~” separates row versus column definitions (LHS vs RHS of equation)
- Left Hand Side (LHS) - row attributes
 - “+” delimits up to two row labeling hierarchies (group + level)
- Right Hand Side (RHS) - column attributes
 - “+” delimits column placement
 - “:” spanning attribute of a columns

- “.” all columns in data frame should be included
- “-” drops following column
- “|” list variables (delimited by "+") to condition on when print multiple simple tables

Formula Interface: Embedded Functions

- Any algebraic manipulation available in `formula` through `I` is also available here
- Formatting
 - Rounding & place holders: `Fr`, `Fci`, `Fc`, and `Fb`
 - Use `paste` to concatenate one or more fields with other text
- Renaming data frame column names to presentable labels. Names default to column names otherwise
 - `Rn(column name, "Presentation Label")`

“\n” can be used in `main` or `footnote` parameters or embedded function `Rn()` to force additional line breaks

Author(s)

Carlin Brickner

Examples

```
### Generate Sample Data Structures ###
# Generate some example data frames
table1 <- rdesc(4, 5) # Numeric
table1f <- rdesc(4, 5, rnd=TRUE) # Rounded and pretty format so values are character
table1a <- table1; table1b <- table1;
table1a$group2 <- 1; table1b$group2 <- 2;
table2 <- rbind(table1a, table1b)
table2a <- table2; table2b <- table2
table2a$group3 <- "Zebra"; table2b$group3 <- "Elephant";
table3 <- rbind(table2a, table2b)
# Create style object
CBs <- style(frmt.bdy=frmt(fontfamily="HersheySans"), frmt.tbl=frmt(bty="o", lwd=1),
            frmt.col=frmt(fontfamily="HersheySans", bg="khaki", fontface="bold", lwd=2, bty="_"),
            frmt.grp=frmt(fontfamily="HersheySans", bg="khaki", fontface="bold"),
            frmt.main=frmt(fontfamily="HersheySans", fontface="bold", fontsize=12),
            frmt.ftn=frmt(fontfamily="HersheySans"),
            justify="right")

# dev.new() # All variables, no group or label
dprint(~., data=table1f)
dev.off()
# dev.new() # Spanning, group level, and apply control and treatments to hierchaies on right
dprint(group+level~Control:(Mean1 + Median1 + Variance1) +
       Treatment:(Mean2 + Median2 + Variance2) + p.value, data=table1f)
dev.off()
# dev.new(); #Illegal Names, remove expression
dprint(group+level~`This is a Control`:(Mean1 + Median1 + Variance1) +
```

```

Treatment.y:(Mean2 + Median2 + Variance2), data=table1f, regx="1|2|.y")
dev.off()
# dev.new(); #Illegal Names, no group label
dprint( ~ `This is a Control`: (Mean1 + Median1 + Variance1) +
Treatment.y:(Mean2 + Median2 + Variance2), data=table1f, regx="1|2|.y")
# dev.new(); # all on rhs with exception of p.value
dev.off()
dprint(group+level~.-p.value, data=table1f)
dev.off()
## Not run:
# dev.new();
dprint(fm1a=group+level~., data=table1)
dev.off()
# dev.new()
dprint(fm1a=group+level~Rn(round(Mean1, 2), "Mean Trt")+Rn(round(Variance1,2), "Variance"),
data=table1)
dev.off()
# dev.new()
dprint(group+level~Rn(round(Mean1, 2), "Mean Trt")+
Variance1+Rn(round((Mean1+Mean2)/2),2), "Average of Averages"), data=table1, main="Don't Do this")
dev.off()
# dev.new()
dprint(level~.|group2, data=table2)
dev.off()
# dev.new();
dprint(level~Mean1+Median2|group2, data=table2, main="Descriptives")
dev.off()
# dev.new(); # Spanning, embedded fuctions, and conditional
dprint(group+level~Treatment:Rn(paste(round(Mean1, 2),"(", round(Variance1, 2),")"),
"Mean Trt (Std)")|group2, data=table2)
dev.off()
# dev.new(); # Spanning, embedded fuctions, and conditional
dprint(~Treatment:Rn(paste(round(Mean1, 2),"(", round(Variance1, 2),")"),
"Mean Trt (Std)")|group2, data=table2)
# dev.new(); # Spanning, embedded fuctions, and conditional
dev.off()
dprint(~Treatment:(Rn(paste(round(Mean1, 2),"(", round(Variance1, 2),")"), "Mean Trt (Std)")+
Rn(round(Median1,2), "Median"))|group2, data=table2)
dev.off()
# dev.new()
dprint(~Treatment:Rn(paste(round(Mean1, 2),"(", round(Variance1, 2),")"), "Mean Trt (Std)")+
Control:Rn(paste(round(Mean2, 2),"(", round(Variance2, 2),")"), "Mean Trt (Std)")|group2,
data=table2)
dev.off()

f1 <- group+level~Treatment:Rn(Fc(Mean1, Variance1), "Mean (Std)")+
Control:Rn(Fc(Mean2, Variance2), "Mean (Std)") + Rn(round(p.value,2), "P-value")
# dev.new()
dprint(fm1a=f1, data=table1,margins=.2, main="Justify Center")
dev.off()
# dev.new()
dprint(fm1a=f1, data=table1,margins=.2, main="Justify Right",
style=style(justify="right", frmt.tbl=frmt(bty="o")))

```

```

dev.off()
# dev.new()
dprint(fmla=f1, data=table1,margins=.2, main="Justify Left",
style=style(justify="left", frmt.tbl=frmt(bty="o")))
dev.off()

h <- expression(hdr("Test Header",
pagelayout.obj=pagelayout(dtype="rgraphics", margins=c(1, .5))))
f <- expression(ftr("R Package tabulaR",
pagelayout.obj=pagelayout(dtype="rgraphics", margins=c(1.25, 1, 1.25,1)),
pagenum=eval.parent(pagenum, 1)))
# dev.new()
dprint(fmla=f1, data=table1,margins=c(1.25, 1, 1.25,1), showmargins=TRUE, main="Table Left",
style=style(justify="left", frmt.tbl=frmt(bty="o"),
frmt.bdy=frmt(linespace=1.5, bty="X")),
f.hdr = h, f.ftr=f, pagenum=1)
dev.off()

# dev.new()
dprint(fmla=f1, data=table1,margins=c(1.25, 1, 1.25,1), showmargins=TRUE, main="Table Left",
style=CBs,
f.hdr = h, f.ftr=f, pagenum=1)
dev.new()
by_var_f1 <- level~Mean1+Median1|group
by_var_f2 <- level~Mean1+Median1|group+group2
# If main is default (null) than do not print titles
dprint(fmla=by_var_f1, data=table2)
dev.off()
# dev.new()
# When a title is defined, and only one conditional variable is defined, just print the values
# concatenated to the text
dprint(fmla=by_var_f1, data=table2,main=" ")
dev.off()
# dev.new()
# When more than one conditional variable, concatenate the variable name and the
# current combination of values
dprint(fmla=by_var_f2, data=table2,main="Descriptives for: ")

## End(Not run)

```

dprint_simp

Print Table to Graphics Device (dprint)

Description

drpint "simple table" class (to be implemented)

Usage

```
dprint_simp(tbl.obj, init = c(0, 6.9), style, char.dim.obj, size.simp.obj)
```

Arguments

tbl.obj	Simple Table object
init	Starting Position of Table
style	style object
char.dim.obj	character dimension object
size.simp.obj	size of text object passed to grid

Fb *Format Binomial*

Description

Inline formula formatting of columns representing binomial summary statistics

Usage

```
Fb(prop = NULL, r = NULL, n = NULL, digits.p = 4, nsmall.p = 2,
  digits.n = 6)
```

Arguments

prop	column name representing a proportion
r	column name representing number of binomial events observed
n	column name representing number of observations
digits.p	digits for proportion presentation. see format
nsmall.p	nsmall for proportion presentation.
digits.n	number of digits for n presentation

Fc *Format Continuous*

Description

Inline formula formatting of columns representing summary statistics for Continuous data

Usage

```
Fc(mn, std, digits = 2, nsmall = 2, NAmis = TRUE)
```

Arguments

mn	column name representing a mean
std	column name representing standard deviation (Variance, Margin of Error, etc.)
digits	see format
nsmall	see format
NAmis	present NA as white space

Fci	<i>Format Interval</i>
-----	------------------------

Description

Inline formula formatting of columns representing lower and upper bound of an interval where the results are of the form 'mn (cil, ciu)'

Usage

```
Fci(cil, ciu, mn = NA, digits = 2, nsmall = 2, NAmis = TRUE)
```

Arguments

cil	column name representing lower interval
ciu	column name representing upper interval
mn	column name representing a mean
digits	see format
nsmall	see format
NAmis	present NA as white space

fitpage	<i>Fit Table to Page</i>
---------	--------------------------

Description

Calculates character expansion to apply across table presentation so that table fits within margins

Usage

```
fitpage(size.simp, pagelayout, fit.width = FALSE, fit.height = FALSE,  
fit = FALSE, cex.0)
```

Arguments

size.simp	size object
pagelayout	dimensions of page
fit.width	boolean, force to fit to width
fit.height	boolean, force to fit height
fit	boolean, force to fit minimum of height and width
cex.0	default cex

`fmla_inter`*Formula Interface Describing the Structure of Tabular Data*

Description

Parse the left and right hand side of the formula.

Usage

```
fmla_inter(f, data = NULL, regx = NA)
```

Arguments

<code>f</code>	formula
<code>data</code>	data.frame
<code>regx</code>	remove regular expression from column name

Author(s)

Rocco Napoli

`fmla_lhs`*Formula Interface: Left Hand Side*

Description

Parses Left hand side of formula

Usage

```
fmla_lhs(f, grp = NULL, lvl = NULL, rhs.lpad = FALSE)
```

Arguments

<code>f</code>	formula
<code>grp</code>	character, name of variable which groups rows together
<code>lvl</code>	character, name of variable which labels each row
<code>rhs.lpad</code>	boolean, right pad

Author(s)

Rocco Napoli

 fmla_rhs

Formula Interface: Right Hand Side

Description

Parses Right hand side of formula

Usage

```
fmla_rhs(f, span, Rn1 = list(Rn.o = list(), Rn.n = NULL, rn.i = 0),
  byvars = list(byvars1 = NULL, byvars2 = NULL, byvars.i = 0))
```

Arguments

f	formula
span	There are two types of calls, when TRUE returns the spanning text, does not when FALSE
Rn1	Rename
byvars	Condition on these variables

Author(s)

Rocco Napoli

 Fr

Format Round

Description

Abreviation of rouding function

Usage

```
Fr(x, digits = 2, nsmall = 2, NAmiss = TRUE)
```

Arguments

x	column name
digits	see format
nsmall	see format
NAmiss	present NA as white space

frmt	<i>Format of Table</i>
------	------------------------

Description

Creates a data structure to define the gpar settings passed to the various grid.* function calls, as well as other formatting controls

Usage

```
frmt(fontfamily = NULL, fontface = NULL, fontsize = NULL,
     text.all = FALSE, col = NULL, bg = NULL, buf = NULL, bty = NULL,
     lwd = NULL, lty = NULL, lcol = NULL, linespace = NULL)
```

Arguments

fontfamily	see gpar
fontface	see gpar
fontsize	see gpar
text.all	If TRUE, when frmt.bdy is defined and nothing else, fontfamily, fontface, fontsize, col will be applied to frmt.col and frmt.row to avoid excessive parameters. Need to define some general rules - probably not implemented correct at this time.
col	Color of text
bg	Background Color
buf	number of character space (buffer) to be forced inbetween columns, and after the labels and group display. When left and right justified, .5 buffer is added and subtracted to respective side of borders
bty	style for box around section of table. "=" = above and below, "o" = completely enclosed, "_" = underneath
lwd	line size for bty, -1 suppresses
lty	line type for bty
lcol	color of lines
linespace	Number of lines between columns

ftr

*Footer***Description**

dprint is to be designed so that users can define custom functions to present footer on page. This is the out of box footer.

Usage

```
ftr(txt1, frmt1 = frmt(fontfamily = "", fontface = "plain", fontsize = 8, col
= "black", linespace = 0.75), date = TRUE,
pagelayout.obj = pagelayout(dtype = "portrait", margins = c(1, 0.5)),
pgtxt2 = "page", pagenum = NULL)
```

Arguments

txt1	A vector of text to be placed on bottom right of footer
frmt1	style sheet data type frmt for text on bottom left
date	Boolean, should today's date be placed in bottom center of page
pagelayout.obj	Tells footer what type of page dprint is working with. Has separate margins to allow for extra space between table presentation
pgtxt2	Text to be appended to page number. To suppress numbering, make NULL
pagenum	Didn't want parameter here, I wanted pagenum to be passed down the calls stack "eval.parent(pagenum, 1)" here is the only way I could get to work ~ carlin

Examples

```
longtable1 <- rdesc(15, 7)
longtable2 <- rdesc(7, 4)
h <- expression(hdr("Multiple Page Report",
                    pagelayout.obj=pagelayout(dtype="landscape", margins=c(.75, .5))))
f <- expression(ftr("R Package tabulaR",
                    pagelayout.obj=pagelayout(dtype="landscape", margins=c(.75, .5))
                    , pagenum=eval.parent(pagenum, 1)
                    ))

pdf("longtable1.pdf", height=8.5, width=11)
dp <- dprint(fmla= group+level~ `This is a Control`:(Mean1 + Variance1) +
            Treatment:(Mean2 + Variance2)+p.value,
            data=longtable1, showmargins=TRUE, dtype="landscape",
            newpage=TRUE, pagenum=1, margins=1,
            f.hdr=h, f.ftr=f
            )

dprint(fmla= group+level~ `This is a Control`:(Mean1 + Variance1) +
```

```

Treatment:(Mean2 + Variance2)+p.value,
  data=longtable2, showmargins=TRUE, dtype="landscape",
  newpage=TRUE, lastcall=dp, # Pick up with page numbering
  margins=1,
  f.hdr=h, f.ftr=f
)
x <- rnorm(100)
y <- rnorm(100)
f2 <- expression(ftr("R Package tabular",
  pagelayout.obj=pagelayout(dtype="landscape", margins=c(.5, .5))
  , pagenum=dp$pagenum+1
))
# par(mai=c(2,3,2,3))
plot(x,y, main="Scatter Plot X vs. Y")
eval(f2)
eval(h)
par(mfcol=c(1,2), pty="s", bg="grey", mai=c(1,1,1,1))
plot(density(x), "Distribution of X", xlab="x", bg="blue")
plot(x,y, main="Scatter Plot X vs. Y")
eval(f2)
eval(h)
dev.off()

```

 hdr

Header

Description

Header

Usage

```

hdr(txt1, frmt1 = frmt(fontfamily = "", fontface = "bold", fontsize = 20, col
= "blue", bty = "_", lwd = 2, linespace = 1), txt2 = NA,
frmt2 = frmt(fontfamily = "", fontface = "bold", fontsize = 16, col = "red",
bty = "_", lwd = 2, linespace = 1), pagelayout.obj = pagelayout(dtype =
"portrait", margins = c(1, 0.5)))

```

Arguments

txt1	Text 1
frmt1	Format 1
txt2	Text 2
frmt2	Format 2
pagelayout.obj	Layout Object

 insert.grp

Insert Values Table Structure

Description

Dispatcher for insert.grp1. Inserts NA positions for tbl, label, and group structure from tbl.obj.

Usage

```
insert.grp(tbl.obj)
```

Arguments

tbl.obj	Table Object
---------	--------------

 insert.grp1

Insert Values

Description

Inserts a value into record in tbl.struct objects to control for when empty rows should be inserted into the data structure. NA's will be suppressed during presentation of the table.

Usage

```
insert.grp1(obj, dx, lblescp, group = FALSE, val = NA, dx.up = FALSE)
```

Arguments

obj	vector or data frame
dx	Index where NA is to be inserted
lblescp	Label Escape, boolean
group	indicates that this object is related to the grouping vector and the insetions must go after
val	Values to be inserted
dx.up	Bolean, If true adjust an index

kill.multiregx	<i>Kill Multiple Regular Expression</i>
----------------	-----------------------------------------

Description

Compacting some regular expression logic to remove regular expressions from vector

Usage

```
kill.multiregx(string, regx)
```

Arguments

string	Input string which will have regular expressions removed
regx	Perl Regular expression

list.to.df	<i>List to Data.frame</i>
------------	---------------------------

Description

Convert a list of vectors, of varying length, to a data frame. Elements are NA where lengths of vectors are smaller than the length of the largest vector.

Usage

```
list.to.df(lst)
```

Arguments

lst	this is a list returned from string split
-----	-------------------------------------------

pagelayout *Page Layout*

Description

Page Layout

Usage

```
pagelayout(dtype = "rgraphics", margins = NULL, pg.dim = NULL)
```

Arguments

dtype	Device type, currently available "rgraphics", "portrait", "landscape"
margins	A numerical vector of the form c(bottom, left, top, right) which gives the margin size specified in inches.
pg.dim	If custom page dimensions define a vector of c(width, height) units

rbin *Simulate Descriptives: Binomial*

Description

Simulate a Data Frame containing descriptive statistics summarizing a binomial distribution

Usage

```
rbin(n.grp = 8, n.lvls = 5, n.grp2 = NULL)
```

Arguments

n.grp	number of row groups
n.lvls	number of rows in each row group
n.grp2	another row grouping, such that grp is subset of grp2

rdesc	<i>Simulate Descriptives: Continuous</i>
-------	------------------------------------------

Description

Simulate a Data Frame containing descriptive statistics summarizing a continuous data

Usage

```
rdesc(n.grp = 8, n.lvls = 5, n.grp2 = NULL, rnd = FALSE)
```

Arguments

n.grp	number of row groups
n.lvls	number of rows in each row group
n.grp2	another row grouping, such that grp is subset of grp2
rnd	boolean round results, data frame then returns as character

row.hl	<i>Row Highlight</i>
--------	----------------------

Description

Creates a data structure for highlighting rows of the table based on indices provided by the user

Usage

```
row.hl(dx, col = "yellow")
```

Arguments

dx	index rows of data.frame to highlight
col	color

size.simp	<i>Size of Simple Table</i>
-----------	-----------------------------

Description

Measures size of properties of the simple table

Usage

```
size.simp(tbl.obj, char.dim.obj, pagelayout, loc.y)
```

Arguments

tbl.obj	Table Object
char.dim.obj	Size of character given formats
pagelayout	Page dimensions
loc.y	Y-coordinate reference

style	<i>Style Sheet</i>
-------	--------------------

Description

Control the mark up and formats for different sections of table

Usage

```
style(frmt.bdy = NULL, frmt.col = NULL, frmt.colh = NULL,
      frmt.grp = NULL, frmt.lbl = NULL, frmt.main = NULL, frmt.tbl = NULL,
      frmt.ftn = NULL, justify = "center", indent = 2, tbl.buf = 0.25,
      cex = 1)
```

Arguments

frmt.bdy	format settings for body of table
frmt.col	format settings for column heading of table
frmt.colh	Only used for borders around column spanning in hierarchy, "o" boxes entire column, "_" puts line under hierarchy
frmt.grp	format settings for row group labels
frmt.lbl	format settings for row labels
frmt.main	format settings for table title
frmt.tbl	format settings for entire table, currently only for the box around entire table (Except table title)

frmt.ftn	format settings for footnote
justify	justification of text, applies to column heading and body
indent	number of characters ("A") to indent the labels underneath the grouping variable
tbl.buf	The space (vertical) between multiple tables
cex	character expansion

Note

Defaults are listed below, unless section of table overwrites with its own defaults:

fontfamily = "" fontface = "plain" fontsize = 8 col = "black" bg = "white" buf = 3 bty = "X" (none)

lwd = 1 lty = 1 lcol="black" linespace = 2

Column heading: fontface="bold",bty="_", lwd=2

Column Spanning: bty="_",

Row Group: fontface="bold", bty="="

Entire table: bty="="

Table title (main): fontface="bold", fontsize=10, linespace=1.5

Footnote: linespace=1.5

Examples

```
# My Style
# Default, this is what is used in dprint if style parameter is not defined
style()
# Save style sheet formats in object to pass to multiple calls
CBs <- style(frmt.bdy=frmt(fontfamily="HersheySans"), frmt.tbl=frmt(bty="o", lwd=1),
            frmt.col=frmt(fontfamily="HersheySans", bg="khaki", fontface="bold",lwd=2,bty="_"),
            frmt.grp=frmt(fontfamily="HersheySans",bg="khaki", fontface="bold"),
            frmt.main=frmt(fontfamily="HersheySans", fontface="bold", fontsize=12),
            frmt.ftn=frmt(fontfamily="HersheySans"),
            justify="right")
```

tablebreak	<i>Table Break</i>
------------	--------------------

Description

Given an index, breaks table into multiple tables

Usage

```
tablebreak(tbl.obj.l, cur.tbl, dx)
```

Arguments

tbl.obj.l	list of table objects
cur.tbl	Index to current table
dx	Row to break table on

tbl.struct	<i>Table Structure</i>
------------	------------------------

Description

Generalization of table structure

Usage

```
tbl.struct(fm1a = NULL, data, label = NULL, group = NULL, regx = NA,
  main = NA, footnote = NA, row.hl = list(dx = NULL, col = NULL))
```

Arguments

fm1a	Formula interface to define table structure
data	data.frame
label	name of column containing row labels
group	name of column containing hierierarchy labels for the row names
regx	regular expression to be removed from original column names
main	Table title
footnote	footnote
row.hl	row highlight object see row.hl function

tbl.struct.simp	<i>Simple Table Structure</i>
-----------------	-------------------------------

Description

Simple Table Structure

Usage

```
tbl.struct.simp(data, label = NULL, group = NULL, main = NA,
  footnote = NA, colnames.obj = NULL)
```

Arguments

data	data.frame
label	name of column containing row labels
group	name of column containing hierierarchy labels for the row names
main	table title
footnote	foot note
colnames.obj	colnames object defined by colnames.struct

vector.linebreak	<i>Vector Line Break</i>
------------------	--------------------------

Description

To break vectors apart around an escape character [backslash]n that indicates a line break. This is designed for handling line breaks for the table title (main) and the footnotes

Usage

```
vector.linebreak(vctr)
```

Arguments

vctr	character vector
------	------------------

vector.struct	<i>Vector Structure</i>
---------------	-------------------------

Description

Defines the structure of a vector given to footnote, main, etc.

Usage

```
vector.struct(vctr = NA)
```

Arguments

vctr	character vector
------	------------------

Index

`char.dim`, 3
`char.height`, 3
`char.width`, 4
`char.width1`, 4
`colnames.linebreak`, 5
`colnames.row`, 5
`colnames.struct`, 5
`conditional.struct`, 6
`consect.struct`, 6

`dborder`, 7
`dprint`, 7
`dprint.data.frame`, 12
`dprint_simp`, 16

`Fb`, 9, 14, 17
`Fc`, 9, 14, 17
`Fci`, 9, 14, 18
`fitpage`, 18
`fmla_inter`, 19
`fmla_lhs`, 19
`fmla_rhs`, 20
`formula`, 9, 14
`Fr`, 9, 14, 20
`frmt`, 8, 12, 21
`ftr`, 8, 13, 22

`hdr`, 8, 13, 23

`I`, 9, 14
`insert.grp`, 24
`insert.grp1`, 24

`kill.multiregx`, 25

`list.to.df`, 25

`pagelayout`, 26
`paste`, 9, 14

`rbin`, 26

`rdesc`, 27
`row.hl`, 8, 13, 27

`size.simp`, 28
`style`, 8, 12, 28

`tablebreak`, 29
`tbl.struct`, 30
`tbl.struct.simp`, 30

`vector.linebreak`, 31
`vector.struct`, 31