Package 'gridExtra'

July 2, 2014

Maintainer baptiste <baptiste.auguie@gmail.com>

License GPL (>= 2)

Title functions in Grid graphics

Type Package

Author Baptiste Auguie

Description misc. high-level Grid functions

Version 0.9.1

URL http://code.google.com/p/gridextra/

Date 2012-09-08

Depends R(>= 2.5.0), grid

Suggests RGraphics, pixmap, EBImage, ggplot2, lattice

Collate 'arcText.r' 'arrange.r' 'barbedGrob.r' 'borderGrob.r' 'colorstripGrob.r' 'ebimage.r' 'ellipseGrob.r' 'gridExtra-package.r' 'grob-utils.r' 'polygon.regular.r' 'ngonGrob.r' 'patternGrob.r' 'pixmap.r' 'rpattern.r' 'stextGrob.r' 'tableGrob.r'

Repository CRAN

Date/Publication 2012-08-09 05:38:06

NeedsCompilation no

R topics documented:

`gridExtra-package` *misc. high-level functions for Grid graphics*

Description

functions for Grid graphics

Author(s)

baptiste Auguie <baptiste.auguie@gmail.com>

References

R Graphics by Paul Murrell, ggplot2 source code

See Also

`Grid`

arcTextGrob *place text labels on a circle and display relations with arcs*

Description

place text labels on a circle and display relations with arcs

Usage

```
arcTextGrob(x = unit(0.5, "npc"), y = unit(0.5, "npc"),
  labels = LETTERS,
  links = sample(seq_along(labels), 50, replace = TRUE),
  default.units = "npc", gp = gpar(), ...)
```

Arguments

x	x unit
y	y unit
labels	text labels
links	integer vector
default.units	default units
gp	gpar
...	additional params (unused)

Value

grob

See Also

Other grob userlevel: barbedGrob, borderGrob, colorstripGrob, ebimageGrob, ellipseGrob, interleaven, ngonGrob, patternGrob, pixmapGrob, rpatternGrob, stextGrob, tableGrob, virtualGrob

Examples

```
library(grid)
set.seed(1234)
grid.newpage()
grid.arcText()
```

arrangeGrob *arrangeGrob*

Description

arrange ggplot2, lattice, and grobs on a page

Usage

```
arrangeGrob(..., as.table = FALSE, clip = TRUE,
  main = NULL, sub = NULL, left = NULL, legend = NULL)
```

Arguments

...	plots of class ggplot2, trellis, or grobs, and valid arguments to grid.layout
main	string, or grob (requires a well-defined height, see example)
sub	string, or grob (requires a well-defined height, see example)
legend	string, or grob (requires a well-defined width, see example)
left	string, or grob (requires a well-defined width, see example)
as.table	logical: bottom-left to top-right or top-left to bottom-right
clip	logical: clip every object to its viewport

Value

return a frame grob; side-effect (plotting) if plot=T

Examples

```
## Not run:
require(ggplot2)
plots = lapply(1:5, function(.x) qplot(1:10,rnorm(10), main=paste("plot",.x)))
require(gridExtra)
do.call(grid.arrange,  plots)
require(lattice)
grid.arrange(qplot(1:10), xyplot(1:10~1:10),
tableGrob(head(iris)), nrow=2, as.table=TRUE, main="test main",
sub=textGrob("test sub", gp=gpar(font=2)))

## End(Not run)
```

barbedGrob *barbedGrob*

Description

plot lines and symbols

Usage

```
barbedGrob(x = stats::runif(10), y = stats::runif(10),
  size = unit(sample(1:4, 10, replace = TRUE), "char"),
  pch = 21, arrow = NULL, space = 1, only.lines = FALSE,
  gp = gpar(), name = NULL, default.units = "npc",
  vp = NULL)
```

Arguments

x	coordinates
y	coordinates
size	unit vector for the symbols
pch	vector of symbol types
space	numeric scaling factor for the exclusion zone
only.lines	logical: should only split lines be returned?
arrow	arrow passed to grid.segments
gp	gpar() object for the symbols
name	grob name
default.units	default units
vp	viewport

Value

a grob

See Also

grid.segments, grid.points

Other grob userlevel: arcTextGrob, borderGrob, colorstripGrob, ebimageGrob, ellipseGrob, interleaven, ngonGrob, patternGrob, pixmapGrob, rpatternGrob, stextGrob, tableGrob, virtualGrob

Examples

```
set.seed(1234)
grid.barbed(name="test")
grid.edit("test", gp=gpar(fill="blue", lwd=3))
grid.edit("test::points", pch=22)
grid.newpage()
g <-
barbedGrob(size=unit(1:5, "char"), only=FALSE,
gp=gpar(col="red", lex=3, fill="blue", alpha=0.5, pch=3))

pushViewport(vp=viewport(width=1, height=1))
grid.rect(gp=gpar(fill="thistle2"))
grid.grill(gp=gpar(col="lavenderblush1", lwd=3, lty=3))
grid.draw(g)
x <- c(0.2, 0.7)
y <- x
dev.new(width=3, height=7)
grid.newpage()
g <-
barbedGrob(x, y, size=unit(c(2, 10), "mm"))
pushViewport(vp=viewport(width=1, height=1))
grid.draw(g)
grid.points(x, y, pch=3)
```

`borderGrob` *borderGrob*

Description

an open rectangular borderdraw

Usage

```
borderGrob(type = 1, colour = "white", vp = NULL, ...)
```

Arguments

`type`	which borders to draw
`colour`	colour
`...`	additional arguments passed to gpar()
`vp`	viewport

Value

a grob

See Also

grid.segments, grid.points

Other grob userlevel: arcTextGrob, barbedGrob, colorstripGrob, ebimageGrob, ellipseGrob, interleaven, ngonGrob, patternGrob, pixmapGrob, rpatternGrob, stextGrob, tableGrob, virtualGrob

Examples

```
pushViewport(viewport(width=0.5, height=0.5, layout=grid.layout(4, 4, w=0.9, height=0.9)))
vp = viewport(width=0.9, height=0.9)
type <- 1
for(ii in 1:4){
 for(jj in 1:4){
  pushViewport(viewport(layout.pos.r=ii, layout.pos.c=jj))
  grid.rect(gp=gpar(col="grey",fill="black"))
  grid.text(paste("t = ", type), gp=gpar(col="white"))
  grid.border(type, vp=vp)
  upViewport()
  type <- type + 1
 }
}
```

---

colorstripGrob *colorstrip*

Description

strip of colors

Usage

```
colorstripGrob(fill = 1:3, colour = fill, draw = TRUE,
  raster = FALSE,
  direction = c("vertical", "horizontal"))
```

Arguments

fill	vector of colours
colour	vector of colours
draw	logical
direction	direction (horizontal or vertical)
raster	logical, use grid.raster (with interpolation)

Value

a grob

See Also

grid.rect

Other grob userlevel: arcTextGrob, barbedGrob, borderGrob, ebimageGrob, ellipseGrob, interleaven, ngonGrob, patternGrob, pixmapGrob, rpatternGrob, stextGrob, tableGrob, virtualGrob

Examples

```
colorstripGrob(c("#E41A1C", "#377EB8", "#4DAF4A"))
```

---

ebimageGrob *create a grob from EBImage object*

Description

create a grob from EBImage object

Usage

```
ebimageGrob(pic, x = 0.5, y = 0.5, scale = 1,
  raster = FALSE, angle = NULL, ...)
```

Arguments

pic	object of class Image
x	x unit
y	y unit
scale	numeric scale factor
raster	logical: use rasterGrob(R>=2.11) or imageGrob(RGraphics)
angle	numeric: angle in degrees
...	optional grob parameters, passed to imageGrob or rasterGrob

Details

Very primitive function, using RGraphics' imageGrob or rasterGrob (R>2.11)

Value

a gTree of class 'ebimage', with natural width and height in points

See Also

Other grob userlevel: arcTextGrob, barbedGrob, borderGrob, colorstripGrob, ellipseGrob, interleaven, ngonGrob, patternGrob, pixmapGrob, rpatternGrob, stextGrob, tableGrob, virtualGrob

Examples

```
## Not run:
library(EBImage)
library(RGraphics)
 x <- readImage("http://www.google.com/logos/teachersday09.gif")
g1 <- ebimageGrob(x)
dev.new(width=g1$width/72, height=g1$height/72)
grid.draw(g1)

## End(Not run)
```

`ellipseGrob` *ellipseGrob*

Description

an ellipse grob

Usage

```
ellipseGrob(x, y, size = 1, angle = rep(pi/2, length(x)),
  ar = 1, gp = gpar(colour = "grey50", fill = "grey90"),
  default.units = "native", size.units = "mm")
```

Arguments

`x`	x unit
`y`	y unit
`size`	size
`angle`	angle
`ar`	aspec ratio
`gp`	gpar
`default.units`	default units
`size.units`	size units

Value

grob

See Also

Other grob userlevel: `arcTextGrob`, `barbedGrob`, `borderGrob`, `colorstripGrob`, `ebimageGrob`, `interleaven`, `ngonGrob`, `patternGrob`, `pixmapGrob`, `rpatternGrob`, `stextGrob`, `tableGrob`, `virtualGrob`

Examples

```
g = ellipseGrob(1:10/11,1:10/11,size=5,ar=1:5,angle=rnorm(10),
                def="npc", gp=gpar(fill=grey(1:10/11)))
grid.draw(g)
```

`expand.arguments` *expand.arguments*

Description

expand a list of arguments to full length (and misc. undocumented functions)

Usage

```
expand.arguments(...)
```

Arguments

`...`	arguments

Value

...

`interleaven` *interleaven*

Description

mix two vectors

Usage

```
interleaven(x = letters[1:3], y = 1:12, n = 4)
```

Arguments

`x`	vector
`y`	vector
`n`	integer

Value

a vector

See Also

rep, ggplot2:interleave

Other grob userlevel: arcTextGrob, barbedGrob, borderGrob, colorstripGrob, ebimageGrob, ellipseGrob, ngonGrob, patternGrob, pixmapGrob, rpatternGrob, stextGrob, tableGrob, virtualGrob

Examples

```
interleaven(replicate(3, rectGrob(), simplify=FALSE), replicate(12, virtualGrob, simplify=FALSE), 4)
```

---

marrangeGrob *marrangeGrob*

Description

Interface to arrangeGrob that can dispatch on multiple pages

Usage

```
marrangeGrob(..., as.table = FALSE, clip = TRUE,
  top = quote(paste("page", g, "of", pages)),
  bottom = NULL, left = NULL, right = NULL)
```

Arguments

...	grobs
as.table	see arrangeGrob
clip	see arrangeGrob
top	see arrangeGrob
bottom	see arrangeGrob
left	see arrangeGrob
right	see arrangeGrob

Details

If the layout specifies both nrow and ncol, the list of grobs can be split in multiple pages. Interactive devices print open new windows, whilst non-interactive devices such as pdf call grid.newpage() between the drawings.

Value

a list of class arrangelist

Author(s)

baptiste Auguie

Examples

```
## Not run:
require(ggplot2)
pl <- lapply(1:11, function(.x) qplot(1:10,rnorm(10), main=paste("plot",.x)))
ml <- do.call(marrangeGrob, c(pl, list(nrow=2, ncol=2)))
## interactive use; open new devices
ml
## non-interactive use, multipage pdf
ggsave("multipage.pdf", ml)

## End(Not run)
```

---

ngonGrob *ngonGrob*

Description

regular polygon grob

Usage

```
ngonGrob(x, y, sides = 5, size = 1,
  angle = rep(pi/2, length(x)), ar = rep(1, length(x)),
  gp = gpar(colour = "grey50", fill = "grey90", linejoin = "mitre"),
  units.def = "native", units.size = "mm")
```

Arguments

x	x unit
y	y unit
sides	sides
size	size
angle	angle
ar	ar
units.def	units.def
units.size	units.size
gp	gpar

Value

grob

See Also

Other grob userlevel: arcTextGrob, barbedGrob, borderGrob, colorstripGrob, ebimageGrob, ellipseGrob, interleaven, patternGrob, pixmapGrob, rpatternGrob, stextGrob, tableGrob, virtualGrob

Examples

```
pushViewport(dataViewport(0:1, 0:1, width=unit(2, "cm"), height=unit(2, "cm")))

xy <- polygon.regular(6, TRUE)
grid.ngon(0.5, 0.5, 6, 10, units.size="mm")
for(ii in 1:NROW(xy)){
 grid.ngon(xy[ii, 1]+0.5, xy[ii, 2]+0.5, 6, 10, units.size="mm")
}
upViewport()
```

---

patternGrob *patternGrob*

Description

pattern grob

Usage

```
patternGrob(x = unit(0.5, "npc"), y = unit(0.5, "npc"),
  width = unit(1, "npc"), height = unit(1, "npc"),
  pattern = 2, orientation = 45,
  granularity = unit(5, "mm"), motif.col = "black",
  motif.fill = NULL, motif.alpha = 1, motif.cex = 1,
  pattern.offset = c(0, 0), default.units = "npc",
  clip = TRUE, gp = gpar(fill = NA), ...)
```

Arguments

x	x unit
y	y unit
width	width
height	height
pattern	integer
orientation	orientation
granularity	unimplemented
motif.col	motif.col
motif.fill	motif.fill
motif.alpha	motif.alpha
motif.cex	motif.cex
pattern.offset	pattern.offset
default.units	default.units
clip	clip
gp	gp
...	additional params to the grob

Value

grob of class pattern

See Also

Other grob userlevel: arcTextGrob, barbedGrob, borderGrob, colorstripGrob, ebimageGrob, ellipseGrob, interleaven, ngonGrob, pixmapGrob, rpatternGrob, stextGrob, tableGrob, virtualGrob

Examples

```
grid.pattern(x=seq(1/6, 5/6, length=6), width=unit(1/8,"npc"), height=unit(0.5,"npc"),
           motif.width=unit(10, "mm"),  pattern=c(1:6), orientation=45, motif.alpha=0.5,
            motif.cex=c(1, 0.5), motif.col=1:2, motif.fill=NA,
            gp=gpar(fill="blue", lwd=2, alpha=0.5),  clip=TRUE)
```

---

pixmapGrob *create a grob from pixmap object*

Description

create a grob from pixmap object

Usage

```
pixmapGrob(pic, x = 0.5, y = 0.5, scale = 1,
  raster = FALSE, angle = 0, vp = NULL, ...)
```

Arguments

pic	pixmap object
x	x unit
y	y unit
scale	scale
raster	raster
angle	angle
vp	viewport
...	optional grob parameters, passed to imageGrob or rasterGrob

Details

Very primitive function, using RGraphics' imageGrob or rasterGrob (R>=2.11)

Value

a gTree of class 'pixmap', with natural width and height in points

See Also

Other grob userlevel: arcTextGrob, barbedGrob, borderGrob, colorstripGrob, ebimageGrob, ellipseGrob, interleaven, ngonGrob, patternGrob, rpatternGrob, stextGrob, tableGrob, virtualGrob

Examples

```
library(pixmap)
library(RGraphics)
x <- read.pnm(system.file("pictures/logo.ppm", package="pixmap")[1])
g1 <- pixmapGrob(x)
dev.new(width=g1$width/72, height=g1$height/72)
grid.draw(g1)
```

---

polygon.regular *polygon.regular*

Description

compute vertices of a polygon

Usage

```
polygon.regular(sides = 4, scale.area = TRUE,
  star = FALSE)
```

Arguments

sides	int
scale.area	logical
star	logical

Value

a data.frame

References

ngon from the 'maptree' package

See Also

star

Examples

```
polygon.regular(5)
```

polygon.star *polygon.star*

Description

compute vertices of a starred polygon

Usage

```
polygon.star(n = 4, adj = pi/2, r = 1)
```

Arguments

n	int
adj	numeric angle
r	radius

Value

a data.frame

Examples

```
polygon.star(5)
```

polygon1 *polygon1*

Description

compute vertices of a convex polygon

Usage

```
polygon1(n = 5, ang = 0, ...)
```

Arguments

n	integer
ang	numeric angle
...	unused

Value

a data.frame

Examples

```
polygon2(5)
```

`read.tiff` *read tiff files with pixmap*

Description

read tiff files with pixmap

Usage

```
read.tiff(con, ...)
```

Arguments

`con`	connection
`...`	unused

Value

...

`rowMax.units` *rowMax.units*

Description

calculates the max of a list of units arranged in a matrix

Usage

```
rowMax.units(u, nrow)
```

Arguments

`u`	list of units
`nrow`	nrow

Value

a vector of units

See Also

`unit.c`, `unit`

rpatternGrob *rpatternGrob*

Description

rectangular grob with raster fill pattern

Usage

```
rpatternGrob(x = unit(0.5, "npc"), y = unit(0.5, "npc"),
  width = unit(1, "npc"), height = unit(1, "npc"),
  motif = matrix("white"), AR = 1,
  motif.width = unit(5, "mm"),
  motif.height = AR * motif.width,
  pattern.offset = c(0, 0), default.units = "npc",
  clip = TRUE, gp = gpar(fill = NA), ...)
```

Arguments

x	x unit
y	y unit
width	width
height	height
motif	motif
AR	AR
motif.width	motif.width
motif.height	motif.height
pattern.offset	pattern.offset
default.units	default.units
clip	clip
gp	gp
...	additional params passed to the grob

Value

grob of class rpattern

See Also

Other grob userlevel: arcTextGrob, barbedGrob, borderGrob, colorstripGrob, ebimageGrob, ellipseGrob, interleaven, ngonGrob, patternGrob, pixmapGrob, stextGrob, tableGrob, virtualGrob

Examples

```
.lines45 <- matrix("white", ncol=100, nrow=100)
diag(.lines45) <- "black"
grid.rpattern(motif=.lines45)
```

---

stextGrob *stextGrob*

Description

shadow text

Usage

```
stextGrob(label, r = 0.1, x = unit(0.5, "npc"),
  y = unit(0.5, "npc"), just = "centre", hjust = NULL,
  vjust = NULL, rot = 0, check.overlap = FALSE,
  default.units = "npc", name = NULL, gp = gpar(),
  vp = NULL)
```

Arguments

label	see textGrob
r	blur radius
x	see textGrob
y	see textGrob
just	see textGrob
hjust	see textGrob
vjust	see textGrob
rot	see textGrob
check.overlap	see textGrob
default.units	see textGrob
name	see textGrob
gp	see textGrob
vp	see textGrob

Details

adds a blurred white version of a label below the text

Value

gTree

Author(s)

Baptiste Auguie

See Also

Other grob userlevel: arcTextGrob, barbedGrob, borderGrob, colorstripGrob, ebimageGrob, ellipseGrob, interleaven, ngonGrob, patternGrob, pixmapGrob, rpatternGrob, tableGrob, virtualGrob

Examples

```
grid.rect(gp=gpar(fill="grey"))
grid.stext("test")
```

---

tableGrob *tableGrob*

Description

create a list of text and fill grobs and calculates the sizes for a table display

Usage

```
tableGrob(d, rows = rownames(d), cols = colnames(d),
  parse = FALSE, show.rownames = TRUE,
  show.colnames = TRUE, row.just = "center",
  col.just = "center", core.just = "center",
  separator = "white", show.box = FALSE,
  show.vlines = FALSE, show.hlines = FALSE,
  show.namesep = FALSE, show.csep = FALSE,
  show.rsep = FALSE, equal.width = FALSE,
  equal.height = FALSE, padding.h = unit(4, "mm"),
  padding.v = unit(4, "mm"), gp = NULL,
  gpar.coretext = gpar(col = "black", cex = 1),
  gpar.coltext = gpar(col = "black", cex = 1, fontface = "bold"),
  gpar.rowtext = gpar(col = "black", cex = 0.8, fontface = "italic"),
  h.odd.alpha = 1, h.even.alpha = 1, v.odd.alpha = 1,
  v.even.alpha = 1,
  gpar.corefill = gpar(fill = "grey95", col = "white"),
  gpar.rowfill = gpar(fill = "grey90", col = "white"),
  gpar.colfill = gpar(fill = "grey90", col = "white"),
  theme = NULL, ...)
```

Arguments

d	data.frame
rows	vector of row names
cols	vector of col names
parse	logical, parse labels as expressions
row.just	justification of labels
col.just	justification of labels
core.just	justification of labels
separator	colour of the border lines
show.box	logical box surrounding the table
show.vlines	logical vertical lines
show.hlines	logical horizontal lines
show.namesep	logical draw lines to separate header(s)
show.csep	logical extend vert. separator to colnames
show.rsep	logical extend vert. separator to rownames
equal.width	logical
equal.height	logical
padding.h	unit of horizontal margin, per cell
padding.v	unit of vertical margin, per cell
gpar.coretext	gpar() for inner text
gpar.corefill	gpar() for inner fill
gpar.coltext	gpar() for colnames text
h.odd.alpha	numeric transparency factor for odd horizontal cells
h.even.alpha	numeric transparency factor for even horizontal cells
v.odd.alpha	numeric transparency factor for odd vertical cells
v.even.alpha	numeric transparency factor for even vertical cells
gpar.colfill	gpar() for colnames fill
gpar.rowtext	gpar() for rownames text
gpar.rowfill	gpar() for rownames fill
show.rownames	logical
show.colnames	logical
gp	gpar
theme	theme (list of aesthetic elements)
...	passed to grob

Value

a grob of class table

See Also

Other grob userlevel: arcTextGrob, barbedGrob, borderGrob, colorstripGrob, ebimageGrob, ellipseGrob, interleaven, ngonGrob, patternGrob, pixmapGrob, rpatternGrob, stextGrob, virtualGrob

Examples

```
grid.table(head(iris), h.even.alpha=1, h.odd.alpha=1,  v.even.alpha=0.5, v.odd.alpha=1)
grid.newpage()
grid.draw(tableGrob(head(iris, 10), name="test"))
e = expression(alpha,"testing very large width", hat(beta), integral(f(x)*dx, a, b), "abc")
grid.edit("test", cols=e, show.rownames=FALSE, rows=NULL,
           gpar.corefill = gpar(fill="white", col=NA),
           grep=TRUE, global=TRUE)
 grid.newpage()
 grid.draw(tableGrob(head(iris, 10),
                 show.csep=TRUE, show.rsep=TRUE, show.box=TRUE, separator="grey", name="test"))
 grid.edit("test",gp=gpar(fontsize=8, lwd=2),  equal.width=TRUE, grep=TRUE, global=TRUE)
# visualize themes
lg <- lapply(c("theme.blank", "theme.default", "theme.white",  "theme.vertical",  "theme.list", "theme.black"),
             function(x) tableGrob(head(iris[, 1:3]), theme=get(x)()))
grid.newpage()
do.call(grid.arrange, lg)
## Not run:
## timing: a bit slow due to repeated on-the-fly calculations
pdf("test2.pdf", height=50)
print(system.time( grid.table(iris)) ) # about 12s here
dev.off()

## End(Not run)
```

`theme.default` *theme*

Description

themes for table grob

Usage

```
theme.default(...)
```

Arguments

...	optional params to overwrite the theme defaults

Value

theme

`virtualGrob` *virtualGrob*

Description

zero grob borrowed from ggplot2

Usage

```
virtualGrob
```

Format

List of 3 $ name: chr "NULL" $ gp : NULL $ vp : NULL - attr(*, "class")= chr [1:3] "virtual" "grob" "gDesc"

See Also

Other grob userlevel: `arcTextGrob`, `barbedGrob`, `borderGrob`, `colorstripGrob`, `ebimageGrob`, `ellipseGrob`, `interleaven`, `ngonGrob`, `patternGrob`, `pixmapGrob`, `rpatternGrob`, `stextGrob`, `tableGrob`

Index