

Package: **plotscale** (via **r-universe**)

September 15, 2024

Type Package

Title Scale Graphics Devices Using Plot Dimensions

Version 0.1.6

Author Tim Bergsma

Maintainer Tim Bergsma <bergsmat@gmail.com>

Description Figures rendered on graphics devices are usually rescaled to fit pre-determined device dimensions. 'plotscale' implements the reverse: desired plot dimensions are specified and device dimensions are calculated to accommodate marginal material, giving consistent proportions for plot elements. Default methods support grid graphics such as lattice and ggplot. See ``example('devsize')" and ``vignette('plotscale')".

Imports grid, graphics

License GPL-3

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, lattice, ggplot2

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Date/Publication 2018-08-31 18:10:03 UTC

Repository <https://bergsmat.r-universe.dev>

RemoteUrl <https://github.com/cran/plotscale>

RemoteRef HEAD

RemoteSha f8796cf483efcc09e060e50092c066dcdcf3fc001

Contents

as.pdf	2
as.pdf.default	3

as.png	4
as.png.default	4
as.size	5
as.size.list	6
devsize	6
devsize.default	7
plotsize	8
plotsize.default	9
print.size	10
unscaled	10
unscaled.default	11

Index	12
--------------	-----------

as.pdf	<i>Coerce to PDF</i>
--------	----------------------

Description

Coerces to PDF. Generic, with default method.

Usage

```
as.pdf(x, ...)
```

Arguments

x	object
...	passed arguments

See Also

Other as.pdf: [as.pdf.default](#)
 Other devices: [as.pdf.default](#), [as.png.default](#), [as.png](#)

Examples

```
example(as.pdf.default)
```

as.pdf.default	<i>Convert to PDF by Default</i>
----------------	----------------------------------

Description

Converts object to PDF. By default, device is scaled by plot size. Device is automatically closed and file argument is returned.

Usage

```
## Default S3 method:
as.pdf(x, width = getOption("plotscale_pdf_width", 3),
       height = getOption("plotscale_pdf_height", 3), devsize = if (scaled)
       "devsize" else "unscaled", scaled = getOption("plotscale_scaled", TRUE),
       onefile = TRUE, file = if (onefile) "Rplots.pdf" else "Rplot%03d.pdf",
       ...)
```

Arguments

x	object
width	desired plot width
height	desired plot height
devsize	a function of x, width, height, and ... that calculates device size
scaled	whether to rescale width and height by plot size
onefile	logical: passed to pdf . If true (the default) allow multiple figures in one file. If false, generate a file with name containing the page number for each page. Defaults to TRUE, and forced to true if file is a pipe.
file	file name passed to pdf
...	other arguments to pdf and fun

Value

(invisible) the file argument

See Also

Other as.pdf: [as.pdf](#)

Other devices: [as.pdf](#), [as.png.default](#), [as.png](#)

Examples

```
options(example.ask = FALSE)
options(device.ask.default = FALSE)
library(lattice)
as.pdf(xyplot(2~3),file = tempfile())
```

 as.png

Coerce to PNG

Description

Coerces to PNG. Generic, with default method.

Usage

```
as.png(x, ...)
```

Arguments

x	object
...	passed arguments

See Also

Other as.png: [as.png.default](#)

Other devices: [as.pdf.default](#), [as.pdf](#), [as.png.default](#)

Examples

```
example(as.png.default)
```

 as.png.default

Convert to PNG by Default

Description

Converts object to PNG. By default, device is scaled by plot size. Device is automatically closed and filename argument is returned.

Usage

```
## Default S3 method:
as.png(x, width = getOption("plotscale_png_width", 3),
       height = getOption("plotscale_png_height", 3), devsize = if (scaled)
       "devsize" else "unscaled", res = getOption("plotscale_res", 300),
       units = "in", scaled = getOption("plotscale_scaled", TRUE),
       filename = "Rplot%03d.png", ...)
```

Arguments

x	object
width	desired plot width
height	desired plot height
devsize	a function of x, width, height, and ... that calculates device size
res	passed to png
units	NOT passed to png ; width and height always treated as inches regardless
scaled	whether to rescale width and height by plot size
filename	file name passed to png
...	other arguments to png and devsize

Value

(invisible) the filename argument

See Also

Other as.png: [as.png](#)

Other devices: [as.pdf.default](#), [as.pdf](#), [as.png](#)

Examples

```
options(example.ask = FALSE)
options(device.ask.default = FALSE)
library(lattice)
as.png(xyplot(2~3), filename = tempfile())
```

as.size

Coerce to Size

Description

Coerce to class 'size'. Generic, with method for list.

Usage

```
as.size(x, ...)
```

Arguments

x	object
...	ignored

See Also

Other generic functions: [devsize](#), [plotsize](#), [unscaled](#)

Other devsize: [devsize.default](#), [devsize](#), [print.size](#), [unscaled.default](#), [unscaled](#)

Other plotsize: [plotsize.default](#), [plotsize](#), [print.size](#)

Other size: [as.size.list](#)

<code>as.size.list</code>	<i>Coerce List to Size</i>
---------------------------	----------------------------

Description

Coerce list to class 'size'.

Usage

```
## S3 method for class 'list'
as.size(x, ...)
```

Arguments

<code>x</code>	object
<code>...</code>	ignored

See Also

Other size: [as.size](#)

<code>devsize</code>	<i>Determine Device Size</i>
----------------------	------------------------------

Description

Determines the size of a device. Generic, with default method.

Usage

```
devsize(x, ...)
```

Arguments

<code>x</code>	object
<code>...</code>	other arguments

See Also

Other generic functions: [as.size](#), [plotsize](#), [unscaled](#)

Other devsize: [as.size](#), [devsize.default](#), [print.size](#), [unscaled.default](#), [unscaled](#)

Examples

```
example(devsize.default)
```

devsize.default *Determine Device Size by Default*

Description

Determines smallest device width and height that give a plot width and height at least as large as the supplied width and height. Plot width and height are determined by `plotsize` (e.g., by the dimensions of the first panel as for [plotsize.default](#)).

Usage

```
## Default S3 method:
devsize(x, width, height, device = list(width = width * 1.1,
    height = height * 1.1), digits = getOption("plotscale_devsize_digits", 2),
    plotsize = getOption("plotscale_plotsize", "plotsize"),
    width.major = TRUE, optimize.minor = TRUE, check.aspect = TRUE,
    verbose = getOption("plotsize_verbose", FALSE), ...)
```

Arguments

<code>x</code>	object
<code>width</code>	desired plot width in inches
<code>height</code>	desired plot height in inches
<code>device</code>	length 2 named list suggesting initial width and height for device
<code>digits</code>	round width, height, device, and result to this many decimal places; passed to <code>plotsize</code>
<code>plotsize</code>	a function of <code>x</code> , <code>width</code> , <code>height</code> , <code>digits</code> and ... that returns a named list with elements 'width' and 'height' giving dimensions of <code>x</code>
<code>width.major</code>	whether to optimize width first for fixed aspect plots (used internally)
<code>optimize.minor</code>	whether to minor dimension for fixed aspect plots (used internally)
<code>check.aspect</code>	whether to adjust width and height to conform to fixed aspect (used internally)
<code>verbose</code>	whether to explain the process
...	other arguments passed to fun

Value

length 2 named list giving device width and height in inches (class 'size')

See Also

Other devsize: [as.size](#), [devsize](#), [print.size](#), [unscaled.default](#), [unscaled](#)

Examples

```
options(example.ask = FALSE)
options(device.ask.default = FALSE)
library(lattice)
data(iris)
x <- xyplot(Petal.Width~Petal.Length, groups = Species, data = iris)

devsize(x, width = 4, height = 4, digits = 1, verbose = TRUE)

devsize(x, width = 4, height = 5, digits = 1, verbose = TRUE)

x <- update(x, aspect = 1)
devsize(x, width = 4, height = 4, digits = 1, verbose = TRUE)

devsize(x, width = 4, height = 5, digits = 1, verbose = TRUE)

devsize(splom(iris[,1:4]), width = 4, height = 4)
devsize(xyplot(Sepal.Length ~ Sepal.Width | Species, data = iris), width = 4, height = 4)
library(ggplot2)
p <- ggplot(data = iris, mapping = aes(y = Sepal.Length, x = Sepal.Width)) +
  geom_point() + facet_wrap(~Species)
devsize(p, width = 4, height = 4)
devsize(p + theme(aspect.ratio = 1), width = 4, height = 4)
```

plotsize

Determine Plot Size

Description

Determines the size of a plot. Generic, with default method.

Usage

```
plotsize(x, ...)
```

Arguments

x	object
...	other arguments

See Also

Other generic functions: [as.size](#), [devsize](#), [unscaled](#)

Other plotsize: [as.size](#), [plotsize.default](#), [print.size](#)

Examples

```
example(plotsize.default)
```

```
plotsize.default      Determine Plot Size by Default
```

Description

Determines the size of a grid graphics plot conditional on device dimensions. Size is defined as physical width and height in inches of the first encountered panel as rendered on a device with the specified width and height. `pdf()` is used as the evaluation device. 'First encountered panel' is resolved by evaluating the output of a call to `current.vpTree` for the first viewport name containing 'panel' or 'subpanel'. The latter takes precedence if available, for intuitive handling of output from `splom`.

Usage

```
## Default S3 method:
plotsize(x, width, height,
         digits = getOption("plotscale_plotsize_digits", 3),
         pattern = c("subpanel\\b", "panel\\b"), index = 1, ...)
```

Arguments

<code>x</code>	object
<code>width</code>	device width in inches
<code>height</code>	device height in inches
<code>digits</code>	round result to this many decimal places
<code>pattern</code>	character: vector of search patterns to identify critical viewport set; first non-empty set is retained
<code>index</code>	integer: select this element of the critical viewport set
<code>...</code>	other arguments passed to pdf

Value

length 2 named list giving width and height in inches (class 'size') for the first panel

See Also

Other plotsize: [as.size](#), [plotsize](#), [print.size](#)

Examples

```
options(ask.default = FALSE)
options(device.ask.default = FALSE)
library(lattice)
p <- xyplot(2~3)
plotsize(p, width = 7, height = 7)
```

<code>print.size</code>	<i>Print Size</i>
-------------------------	-------------------

Description

Print object of class 'size', as returned by [devsize](#) and [plotsize](#).

Usage

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

Arguments

<code>x</code>	object
<code>...</code>	ignored

See Also

Other `plotsize`: [as.size](#), [plotsize.default](#), [plotsize](#)

Other `devsize`: [as.size](#), [devsize.default](#), [devsize](#), [unscaled.default](#), [unscaled](#)

<code>unscaled</code>	<i>Determine Unscaled Device Size</i>
-----------------------	---------------------------------------

Description

Determines the size of an unscaled device. Generic, with default method.

Usage

```
unscaled(x, ...)
```

Arguments

<code>x</code>	object
<code>...</code>	other arguments

See Also

Other generic functions: [as.size](#), [devsize](#), [plotsize](#)

Other `devsize`: [as.size](#), [devsize.default](#), [devsize](#), [print.size](#), [unscaled.default](#)

Examples

```
example(plotsize.default)
```

unscaled.default *Determine Unscaled Device Size by Default*

Description

Determines the size of an unscaled device. Width and height are interpreted directly as device dimensions rather than plot dimensions.

Usage

```
## Default S3 method:  
unscaled(x, width, height, ...)
```

Arguments

x	object
width	desired width in inches
height	desired height in inches
...	other arguments

See Also

Other devsize: [as.size](#), [devsize.default](#), [devsize](#), [print.size](#), [unscaled](#)

Index

as.pdf, [2](#), [3–5](#)
as.pdf.default, [2](#), [3](#), [4](#), [5](#)
as.png, [2](#), [3](#), [4](#), [5](#)
as.png.default, [2–4](#), [4](#)
as.size, [5](#), [6–11](#)
as.size.list, [6](#), [6](#)

current.vpTree, [9](#)

devsize, [6](#), [6](#), [8](#), [10](#), [11](#)
devsize.default, [6](#), [7](#), [7](#), [10](#), [11](#)

pdf, [3](#), [9](#)
plotsize, [6](#), [7](#), [8](#), [9](#), [10](#)
plotsize.default, [6–8](#), [9](#), [10](#)
png, [5](#)
print.size, [6–10](#), [10](#), [11](#)

round, [7](#), [9](#)

splom, [9](#)

unscaled, [6–8](#), [10](#), [10](#), [11](#)
unscaled.default, [6–8](#), [10](#), [11](#)