

Package ‘ggview’

October 2, 2024

Type Package

Title 'ggplot2' Picture Previewer

Version 0.2.1

Description Preview what a 'ggplot2' plot would look like if you save it to a file.
Attach picture dimensions as a canvas() element and get an instant preview.
These dimensions will then be used when you save the plot.

License GPL-2 | file LICENSE

Encoding UTF-8

RoxygenNote 7.2.1

Imports ggplot2, rstudioapi

URL <https://github.com/idmn/ggview>

BugReports <https://github.com/idmn/ggview/issues>

Suggests testthat (>= 3.0.0), png

Config/testthat/edition 3

NeedsCompilation no

Author Iaroslav Domin [aut, cre, cph]

Maintainer Iaroslav Domin <ya.domin@gmail.com>

Repository CRAN

Date/Publication 2024-10-02 17:00:10 UTC

Contents

canvas	2
save_ggplot	3

Index	5
--------------	----------

canvas *Add a canvas specification to a ggplot object*

Description

A canvas specification essentially comprises a set of parameters from `ggplot2::ggsave()`. When a plot with this canvas specification is printed, it is rendered as it would appear if saved to a file with the specified dimensions.

Usage

```
canvas(  
  width,  
  height,  
  units = c("in", "cm", "mm", "px"),  
  dpi = 300,  
  scale = 1,  
  bg = "white"  
)
```

Arguments

width, height	Plot size in units expressed by the <code>units</code> argument. If not supplied, uses the size of the current graphics device.
units	One of the following units in which the width and height arguments are expressed: "in", "cm", "mm" or "px".
dpi	Plot resolution. Also accepts a string input: "retina" (320), "print" (300), or "screen" (72). Applies only to raster output types.
scale	Multiplicative scaling factor.
bg	Background colour. If NULL, uses the <code>plot.background</code> fill value from the plot theme.

Value

An object of class `canvas` that can be added to a `ggplot` object to specify the plot dimensions.

Examples

```
library(ggplot2)  
p <-  
  ggplot(mtcars, aes(wt, mpg)) +  
  geom_point() +  
  ggtitle("My awesome plot")  
  
p + canvas(3, 3)  
p + canvas(5, 3, dpi = 400)
```

save_ggplot

*Save a ggplot***Description**

Saves a ggplot object just like `ggplot2::ggsave()`. If the plot has a `canvas()` specified, these canvas parameters are used. User-specified parameters will override the canvas defaults.

Usage

```
save_ggplot(
  plot,
  file,
  device = NULL,
  scale = NULL,
  width = NULL,
  height = NULL,
  units = NULL,
  dpi = NULL,
  limitsize = TRUE,
  bg = NULL,
  create.dir = FALSE,
  ...
)
```

Arguments

plot	The ggplot object to save.
file	File to save the plot to.
device	Device to use. Can either be a device function (e.g. <code>png</code>), or one of "eps", "ps", "tex" (pictex), "pdf", "jpeg", "tiff", "png", "bmp", "svg" or "wmf" (windows only). If NULL (default), the device is guessed based on the filename extension.
scale	Multiplicative scaling factor.
width, height	Plot size in units expressed by the <code>units</code> argument. If not supplied, uses the size of the current graphics device.
units	One of the following units in which the width and height arguments are expressed: "in", "cm", "mm" or "px".
dpi	Plot resolution. Also accepts a string input: "retina" (320), "print" (300), or "screen" (72). Applies only to raster output types.
limitsize	When TRUE (the default), <code>ggsave()</code> will not save images larger than 50x50 inches, to prevent the common error of specifying dimensions in pixels.
bg	Background colour. If NULL, uses the <code>plot.background</code> fill value from the plot theme.

<code>create.dir</code>	Whether to create new directories if a non-existing directory is specified in the filename or path (TRUE) or return an error (FALSE, default). If FALSE and run in an interactive session, a prompt will appear asking to create a new directory when necessary.
<code>...</code>	Other arguments passed on to the graphics device function, as specified by device.

Value

The function is called for its side effects: it saves the plot to a file and returns the file path invisibly.

Examples

```
library(ggplot2)
p <-
  ggplot(mtcars, aes(wt, mpg)) +
  geom_point() +
  ggtitle("My awesome plot") +
  canvas(8, 6)

temp_file <- tempfile(fileext = ".png")
save_ggplot(p, temp_file)
```

Index

canvas, [2](#)

canvas(), [3](#)

ggplot2::ggsave(), [2](#), [3](#)

png, [3](#)

save_ggplot, [3](#)