Package: showtext (via r-universe)

October 17, 2026

Type Package

Title Using Fonts More Easily in R Graphs

Version 0.9-8

Date 2026-03-21

Description Making it easy to use various types of fonts ('TrueType', 'OpenType', Type 1, web fonts, etc.) in R graphs, and supporting most output formats of R graphics including PNG, PDF and SVG. Text glyphs will be converted into polygons or raster images, hence after the plot has been created, it no longer relies on the font files. No external software such as 'Ghostscript' is needed to use this package.

Depends sysfonts (>= 0.7.1), showtextdb (>= 2.0)

Imports grDevices

Suggests knitr, rmarkdown, prettydoc, curl, jsonlite

SystemRequirements zlib, libpng, FreeType

URL https://github.com/yixuan/showtext

BugReports https://github.com/yixuan/showtext/issues

License Apache License (>= 2.0)

Copyright see file COPYRIGHTS

VignetteBuilder knitr, rmarkdown

RoxygenNote 7.3.3

Encoding UTF-8

Config/pak/sysreqs libfreetype6-dev libpng-dev zlib1g-dev

Repository https://yixuan.r-universe.dev

Date/Publication 2026-03-21 05:32:03 UTC

RemoteUrl https://github.com/yixuan/showtext

RemoteRef HEAD

RemoteSha f4c498d9be9f249de1c3f9cba2bebd7f0e0ab980

Contents

showtext_auto 2
showtext_begin 3
showtext_end 5
showtext_opts 5

Index **7**

---

showtext_auto *Automatically Using 'showtext' for New Graphics Devices*

Description

The two versions of this function are equivalent, but the "underscore" naming is preferred.

This function could turn on/off the automatic use of **showtext** functionality. If turned on, any newly opened graphics devices will use **showtext** to draw text. This helps to avoid the repeated calls of `showtext_begin()` and `showtext_end()`.

Usage

```
showtext_auto(enable = TRUE, record = TRUE)

showtext.auto(enable = TRUE, record = TRUE)
```

Arguments

`enable`	`TRUE` to turn on and `FALSE` to turn off.
`record`	If `TRUE`, then `showtext_begin()` is added to the display list of the current graphics device (via `recordGraphics()`), so that it may be "replayed" at a later time point (via `replayPlot()`).

Author(s)

Yixuan Qiu <https://statr.me/>

See Also

`showtext_begin()`, `showtext_end()`

Examples

```
## Not run:
pdf("test1.pdf")
plot(1, main = "\u6b22\u8fce")  ## may not render properly
dev.off()

## Automatically use showtext for future devices
showtext_auto()
```

```
plot(1, main = "\u6b22\u8fce", family = "wqy-microhei")

pdf("test2.pdf")
plot(1, main = "\u6b22\u8fce", family = "wqy-microhei")
dev.off()

## Turn off if needed
showtext_auto(FALSE)

## End(Not run)
```

`showtext_begin` *Rendering Text for R Graphics Devices*

Description

The two versions of this function are equivalent, but the "underscore" naming is preferred.

Calling this function will use **showtext** to render text for the current graphics device. The main advantage of **showtext** is that user can use any supported font file for the text rendering, and text glyphs will be converted into polygons (for vector graphics) or raster images (for bitmap and on-screen graphics), thus producing device independent output on all platforms. This function would be useful if you want to use non-standard fonts in the graphics device. The usage of this function is easy: simply open the graphics device, and "claim" that you want to use **showtext** by calling this function. See the **Examples** section for details.

Usage

```
showtext_begin()

showtext.begin()
```

Details

This package uses FreeType to load font files and render text characters. The font loading part is done by function `font_add()` in the **sysfonts** package. Users could read the help page of `font_paths()`, `font_files()` and `font_add()` to learn how to load a font file into R. **showtext** package has a built-in font file of the WenQuanYi Micro Hei family, and it will be loaded automatically, so users can use this function directly without any extra settings.

The mechanism of this function is that it can replace the text rendering functions contained in the current device. **showtext** will first use FreeType to analyze the outline of each character in the text, and then call some low-level drawing functions in the current device to draw the glyph. As a result, glyphs of the text will be finally converted into polygons or raster images, which means that the system where the graph is viewed does not need to install the fonts that create the graph.

Notice that this function is only effective to the current **ACTIVE** device. So to use this function, the device you want to work with must have already been opened (through functions like `png()`, `pdf()`, etc.).

To switch back, users can call `showtext_end()` to restore the original device functions. See examples below for the usage of these functions.

Author(s)

Yixuan Qiu <https://statr.me/>

See Also

showtext_opts(), showtext_auto(), showtext_end()

Examples

```
## Not run:
old = setwd(tempdir())

###  Enable pdf() to draw Chinese characters nicely  ###
###  Requires the simkai.ttf font file, usually      ###
###  installed in Windows                            ###

## First, open the device
pdf("showtext-ex1.pdf")

## For now we are using the original device functions to draw axis labels
plot(1, type = "n")

## Then turn showtext on and draw some characters
showtext_begin()
text(1, 1.2, intToUtf8(c(21315, 31179, 19975, 36733)), cex = 5)

## Use another font
if("simkai.ttf" %in% font_files()) font_add("kaishu", "simkai.ttf")
text(1, 0.8, intToUtf8(c(19968, 32479, 27743, 28246)),
     cex = 5, family = "kaishu")

## Turn showtext off
showtext_end()

## Also turn off the device
dev.off()


###  Download font file from web  ###

download.file("http://fontpro.com/download-family.php?file=36289",
              "newrocker.ttf", mode="wb")
download.file("http://fontpro.com/download-family.php?file=35128",
              "cutetumblr.ttf", mode ="wb")

font_add("newrocker", "newrocker.ttf")
font_add("cutetumblr", "cutetumblr.ttf")

png("showtext-ex2.png", 800, 500)
plot(1, type = "n")
showtext_begin()
text(1, 1.2, "Let me tell you a story", cex = 4, family = "newrocker")
```

```
text(1, 0.8, "Long long ago...", cex = 4, family = "cutetumblr")
showtext_end()
dev.off()

setwd(old)


## End(Not run)
```

---

showtext_end *Turning Off 'showtext' Text Rendering*

Description

The two versions of this function are equivalent, but the "underscore" naming is preferred.

This function will turn off the **showtext** functionality of rendering text. When you call this function, the current active device should be the same as the one when you call `showtext_begin()`, or an error will be issued. See the example in `showtext_begin()`.

Usage

```
showtext_end()

showtext.end()
```

Author(s)

Yixuan Qiu <https://statr.me/>

See Also

`showtext_begin()`

---

showtext_opts *Setting Options for 'showtext' package*

Description

The two versions of this function are equivalent, but the "underscore" naming is preferred.

This function sets parameters that will affect the appearance of the graphs created with **showtext**.

Usage

```
showtext_opts(...)

showtext.opts(...)
```

Arguments

`...`	Options to be set, expressed in `name = value` pairs. It can also be a list containing these pairs (for example, the list returned by a previous call of `showtext_opts()`). Currently accepted parameters are `nseg` and `dpi`. See the **Options Used** section.

Options Used

`nseg` Parameter to control the smoothness of the outlines of glyphs, typically used in vector graphics devices such as `pdf()` and `svg()`. It is the number of line segments to approximate a piece of curve in the glyph. The larger `nseg` is, the smoother text outlines would be, but also with larger file size for vector graphics. Usually a value between 5~20 would be enough.

`dpi` An integer that gives the resolution of the device. This parameter is only used in bitmap and on-screen graphics devices such as `png()` and `x11()`, to determine the pixel size of text from point size. For example, if `dpi` is set to 96, then a character with 12 point size will have a pixel size of `12 * 96 / 72 = 16`.

Author(s)

Yixuan Qiu <https://statr.me/>

Examples

```
## Not run:
## Set dpi to 200
op = showtext_opts(dpi = 200)

png("dpi_200.png", 800, 800, res = 200)
par(family = "sans")
showtext_begin()

set.seed(123)
plot(density(rnorm(100)))

showtext_end()
dev.off()

## Restore old options
showtext_opts(op)

png("dpi_96.png", 800, 800, res = 96)
par(family = "sans")
showtext_begin()

set.seed(123)
plot(density(rnorm(100)))

showtext_end()
dev.off()

## End(Not run)
```

Index

font_add, *3*
font_files, *3*
font_paths, *3*

pdf, *3*
png, *3*

recordGraphics, *2*
replayPlot, *2*

showtext.auto *(*showtext_auto*)*, 2
showtext.begin *(*showtext_begin*)*, 3
showtext.end *(*showtext_end*)*, 5
showtext.opts *(*showtext_opts*)*, 5
showtext_auto, 2, *4*
showtext_begin, *2*, 3, *5*
showtext_end, *2–4*, 5
showtext_opts, *4*, 5