R.devices

Henrik Bengtsson

November 19, 2012

Abstract

The *R.devices* package provides utility methods that enhance the existing graphical device functions already available in R for the purpose of simplifying the creation of figure files of various image formats and much more.

Keywords: devices, graphics, plots, figures

This vignette is distributed as part of the R.devices package, which is available on CRAN (http://cran.r-project.org/). Feedback is very much appreciated.

Contents

1 Creating image files

When creating image files using one of the built-in R device functions (e.g. `pdf()`) several device specific arguments needs to the specified. This makes it tedious to change the output format. For instance, when first creating a PDF file (`file="GaussianDensity.pdf"`) with aspect ratio 0.6 (`width=7, height=0.6*7`), it is necessary to modify at least three of the arguments (`file="GaussianDensity.png", width=480, height=0.6*480`) in order to create a PNG image file.

1.1 devEval()

To overcome the above and other hurdles, the `devEval()` function was created. When using `devEval()` it is only argument that specify the image format that needs to be modified. For instance,

```
devEval("pdf", name="GaussianDensity", aspectRatio=0.6, {
  curve(dnorm, from=-5, to=+5)
})
```

creates a PDF file named *GaussianDensity.pdf* that is 7.0 inches wide and 4.2 inches tall, whereas

```
devEval("png", name="GaussianDensity", aspectRatio=0.6, {
  curve(dnorm, from=-5, to=+5)
})
```

creates a PNG file named *GaussianDensity.png* that is 480 pixels wide and 288 pixels tall. For default dimensions, see Section 2. By specifying the `scale` argument, it is possible to create an image file with a smaller or a larger dimension relative to that of the default, e.g.

```
devEval("png", name="GaussianDensity,large", aspectRatio=0.6, scale=2, {
  curve(dnorm, from=-5, to=+5)
})
```

creates a PNG file named *GaussianDensity,large.png* that is 960 pixels wide and 576 pixels tall. Note also how there is in none of the above examples a need for closing the device via `dev.new()`, which is sometimes forgotten by newcomers. The graphical device opened is also guaranteed to be closed by `devEval()`.

1.2 toEPS(), toPDF(), toPNG() etc.

For conveniency, there exists a set of `toNNN()` functions that basically are wrappers for `devEval()`. For instance, instead of calling `devEval("png", ...)` one can use `toPNG(...)` as

```
toPNG("GaussianDensity,large", aspectRatio=0.6, scale=2, {
  curve(dnorm, from=-5, to=+5)
})
```

The following `toNNN()` functions are currently available: `toBMP()`, `toEPS()`, `toPDF()`, `toPNG()`, `toSVG()`, and `toTIFF()`.

1.3 Setting default output directory

All figures created by `devEval()/toNNN()` are by default written to the *figures/* directory (created if missing), which can be overridden by passing argument `path` to `devEval()`. The default figure path can be change by setting option `"devEval/args/path"`, which will be created if needed, e.g.

```
options("devEval/args/path"="figures/col/")
```

1.4 Names and comma-separated tags

The filename used by `devEval()/toNNN()`, is made up of argument `name`, followed by comma-separated argument `tags` (an optional character vector) and a filename extension (specified by the device type). Argument `tags` provides a convenient way to adjust the filename, e.g.

```
for (ar in c(0.6, 0.8)) {
  arTag <- sprintf("aspect=%g", ar)
  for (sc in 2:4) {
    scaleTag <- sprintf("scale=%d", sc)
    toEPS("GaussianDensity", tags=c(arTag, scaleTag), aspectRatio=ar, scale=sc, {
      curve(dnorm, from=-5, to=+5)
    })
  }
})
```

which creates six images files (*GaussianDensity,aspect=0.6,scale=2.eps*, *GaussianDensity,aspect=0.6,scale=3.eps*, ..., and *GaussianDensity,aspect=0.8,scale=4.eps*).

1.5 Overwriting existing figure files

By default, existing figure files created by `devEval()/toNNN()` are overwritten wihtout notice. By passing argument `force=FALSE` to `devEval()`, existing figure files will be skipped. To change the default, set option `"devEval/args/force"`, e.g.

```
 options("devEval/args/force"=FALSE)
```

Note that whenever a figure is skipped this way, it also means that none of the expressions in `devEval(..., {<exprs>})` are executed. This will speed up the processing, but it also means that the rest of your code must not rely on such code being executed.

1.6 No more incomplete image files

The `devEval()/toNNN()` functions create image files atomically. When creating image files by opening a device, calling a set of plot functions and then closing the device (`png(...); {...}; dev.off()`), there is a risk of creating an incomplete file whenever an error or an interrupt occurs while plotting. By contrast, `devEval()/toNNN()` is fault tolerant and guarantees that the image file created is complete; if an error or an interrupt occurs, then the default is to remove the incomplete image. For instance, the following will not result in an image file:

```
toPDF("GaussianDensity", {
  curve(dnorm, from=-5, to=+5)
  abline(v=log("a"))
})
```

because the last plot statement generates an error. To further lower the risk for incomplete image files, for instance due to abrupt power failures, all image files are first written to a temporary file which is renamed to the final file only when the plotting is complete. This is useful when for instance running large non-interactive batch jobs that creates hundreds or thousands of image files.

1.7 Including images in RSP-embedded LaTeX documents

By using RSP-markup, image files can be included in for instance LaTeX, Sweave and knitr documents in a very clean fashion, while keeping full control of all image formatting. For instance, the plot in Figure 1 was included as:

```
\includegraphics{<%=toPDF("MyGaussianDensity", aspectRatio=0.6, {
  curve(dnorm, from=-5, to=+5)
})%>}
```

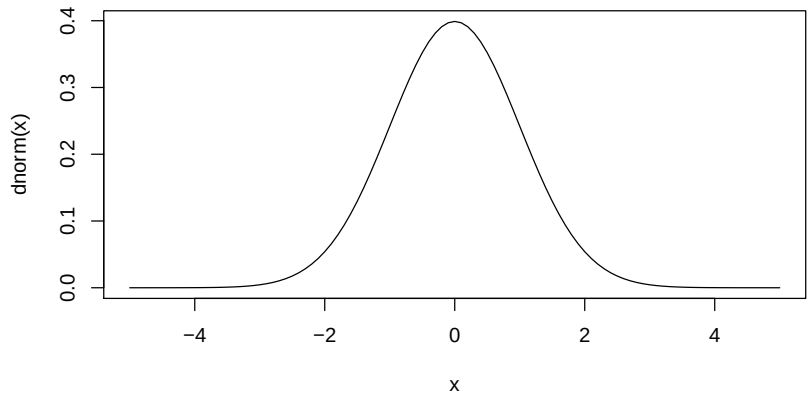

Figure 1: This graph was generated using `toPDF()` and then include into the LaTeX document using RSP.

For more details on RSP, see the vignettes of the *R.rsp* package (available on CRAN).

2 Setting default device options

The `devOptions()` function provides a unified interface to getting and settings common options for the various graphical devices available in R. When using one of the `toNNN()` functions, `devEval()` or `devNew()`, the device options used are given by `devOptions()`. For example, to see the current settings used by PNG device, do:

```
> devOptions("png")
$filename
[1] "Rplot%03d.png"
$width
[1] 480
$height
[1] 480
$units
[1] "px"
$pointsize
```

```
[1] 12
$bg
[1] "white"
$res
[1] NA
$family
[1] "sans"
$restoreConsole
[1] TRUE
$type
c("windows", "cairo", "cairo-png")
$antialias
c("default", "none", "cleartype", "grey", "subpixel")
```

To change one or several options, do:

```
> devOptions("png", width = 1024, bg = "lightblue")
```

To reset the options back to the built-in defaults, do:

```
> devOptions("png", reset = TRUE)
```

To get an overview of a set of common options for the supported devices, do:

```
> knownDevices <- eval(formals(devOptions)$type)
> fields <- c("width", "height", "bg", "fg", "pointsize")
> opts <- sapply(knownDevices, FUN = function(type) devOptions(type)[fields])
> rownames(opts) <- fields
> opts <- t(opts)
> print(opts)
           width height bg            fg      pointsize
bmp        480   480    "white"       NULL    12
cairo_pdf  7     7      "white"       NULL    12
cairo_ps   7     7      "white"       NULL    12
eps        7     7      "transparent" "black" 12
jpeg       480   480    "white"       NULL    12
jpeg2      480   480    "transparent" "black" 12
pdf        7     7      "transparent" "black" 12
pictex     5     4      "white"       "black" NULL
png        480   480    "white"       NULL    12
png2       480   480    "transparent" "black" 12
postscript 8.27  11.7   "transparent" "black" 12
quartz     NULL  NULL   NULL          NULL    NULL
svg        7     7      "white"       NULL    12
tiff       480   480    "white"       NULL    12
windows    7     7      "transparent" NULL    12
x11        7     7      "transparent" NULL    12
xfig       8.27  11.7   "transparent" "black" 12
```

2.1 Under the hood (advanced)

The `devOptions()` function tries as far as possible to infer the default options from the default arguments of the device function and any additional options for that device, e.g. `formals(pdf)` and `pdf.options()`. Likewise, when setting an option it uses the standard interfaces to do so, whenever possible. This means that for instance `pdf()` will also be affected by `devOptions("pdf", width=5)`. Note that this may not be the case for all devices, because their options cannot be set. Instead they are all specified as arguments when opening the device, e.g. `png()` will *not* be affected by `devOptions("png", width=1024)`. This is why we recommend to always use `devNew()` in place of `dev.new()`, or better, `devEval()` or the corresponding `toNNN()` function, which all respects the options set via `devOptions()`.

Appendix

Session information

- R version 3.0.0 RC (2013-03-31 r62463), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=C`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: R.devices 2.2.2, R.methodsS3 1.4.3, R.oo 1.13.1, R.rsp 0.9.1, R.utils 1.23.2
- Loaded via a namespace (and not attached): tools 3.0.0

This report was automatically generated using `rfile()` of the R.rsp package. Total processing time after RSP-to-R translation was 0.24 secs.