Graphics Devices in R

Roger D. Peng, Associate Professor of Biostatistics
Johns Hopkins Bloomberg School of Public Health

What is a Graphics Device?

- A graphics device is something where you can make a plot appear
  - A window on your computer (screen device)
  - A PDF file (file device)
  - A PNG or JPEG file (file device)
  - A scalable vector graphics (SVG) file (file device)
- When you make a plot in R, it has to be "sent" to a specific graphics device
- The most common place for a plot to be "sent" is the *screen device*
  - On a Mac the screen device is launched with the `quartz()`
  - On Windows the screen device is launched with `windows()`
  - On Unix/Linux the screen device is launched with `x11()`

What is a Graphic Device?

- When making a plot, you need to consider how the plot will be used to determine what device the plot should be sent to.
  - The list of devices is found in `?Devices`; there are also devices created by users on CRAN
- For quick visualizations and exploratory analysis, usually you want to use the screen device
  - Functions like `plot` in base, `xyplot` in lattice, or `qplot` in ggplot2 will default to sending a plot to the screen device
  - On a given platform (Mac, Windows, Unix/Linux) there is only one screen device
- For plots that may be printed out or be incorporated into a document (e.g. papers/reports, slide presentations), usually a *file device* is more appropriate
  - There are many different file devices to choose from
- NOTE: Not all graphics devices are available on all platforms (i.e. you cannot launch the `windows()` on a Mac)

How Does a Plot Get Created?

There are two basic approaches to plotting. The first is most common:

1. Call a plotting function like `plot`, `xyplot`, or `qplot`
2. The plot appears on the screen device
3. Annotate plot if necessary
4. Enjoy

```
library(datasets)
with(faithful, plot(eruptions, waiting))  ## Make plot appear on screen device
title(main = "Old Faithful Geyser data")  ## Annotate with a title
```

How Does a Plot Get Created?

The second approach to plotting is most commonly used for file devices:

1. Explicitly launch a graphics device
2. Call a plotting function to make a plot (Note: if you are using a file device, no plot will appear on the screen)
3. Annotate plot if necessary
4. Explicitly close graphics device with `dev.off()` (this is very important!)

```
pdf(file = "myplot.pdf")  ## Open PDF device; create 'myplot.pdf' in my working directory
## Create plot and send to a file (no plot appears on screen)
with(faithful, plot(eruptions, waiting))
title(main = "Old Faithful Geyser data")  ## Annotate plot; still nothing on screen
dev.off()  ## Close the PDF file device
## Now you can view the file 'myplot.pdf' on your computer
```

Graphics File Devices

There are two basic types of file devices: *vector* and *bitmap* devices

Vector formats: good for resize but will be very large if many many points

- `pdf`: useful for line-type graphics, resizes well, usually portable, not efficient if a plot has many objects/points
- `svg`: XML-based scalable vector graphics; supports animation and interactivity, potentially useful for web-based plots
- `win.metafile`: Windows metafile format (only on Windows)
- `postscript`: older format, also resizes well, usually portable, can be used to create encapsulated postscript files; Windows systems often don't have a postscript viewer

Graphics File Devices

Bitmap formats Not good for resizable

- `png`: bitmapped format, good for line drawings or images with solid colors, uses lossless compression (like the old GIF format), most web browsers can read this format natively, good for plotting many many many points, does not resize well
- `jpeg`: good for photographs or natural scenes, uses lossy compression, good for plotting many many many points, does not resize well, can be read by almost any computer and any web browser, not great for line drawings
- `tiff`: Creates bitmap files in the TIFF format; supports lossless compression
- `bmp`: a native Windows bitmapped format

Multiple Open Graphics Devices

- It is possible to open multiple graphics devices (screen, file, or both), for example when viewing multiple plots at once
- Plotting can only occur on one graphics device at a time
- The **currently active** graphics device can be found by calling `dev.cur()`
- Every open graphics device is assigned an integer ≥ 2.
- You can change the active graphics device with `dev.set(<integer>)` where `<integer>` is the number associated with the graphics device you want to switch to

Copying Plots

Copying a plot to another device can be useful because some plots require a lot of code and it can be a pain to type all that in again for a different device.

- `dev.copy`: copy a plot from one device to another
- `dev.copy2pdf`: specifically copy a plot to a PDF file

NOTE: Copying a plot is not an exact operation, so the result may not be identical to the original.

```
library(datasets)
with(faithful, plot(eruptions, waiting))  ## Create plot on screen device
title(main = "Old Faithful Geyser data")  ## Add a main title
dev.copy(png, file = "geyserplot.png")  ## Copy my plot to a PNG file
dev.off()  ## Don't forget to close the PNG device!
```

Summary

- Plots must be created on a graphics device
- The default graphics device is almost always the screen device, which is most useful for exploratory analysis
- File devices are useful for creating plots that can be included in other documents or sent to other people
- For file devices, there are vector and bitmap formats
    - Vector formats are good for line drawings and plots with solid colors using a modest number of points
    - Bitmap formats are good for plots with a large number of points, natural scenes or web-based plots