CRAN Task View: Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization

Maintainer:	Nicholas Lewin-Koh
Contact:	nikko at hailmail.net
Version:	2015-01-07
URL:	https://CRAN.R-project.org/view=Graphics

R is rich with facilities for creating and developing interesting graphics. Base R contains functionality for many plot types including coplots, mosaic plots, biplots, and the list goes on. There are devices such as postscript, png, jpeg and pdf for outputting graphics as well as device drivers for all platforms running R. lattice and grid are supplied with R's recommended packages and are included in every binary distribution. lattice is an R implementation of William Cleveland's trellis graphics, while grid defines a much more flexible graphics environment than the base R graphics.

R's base graphics are implemented in the same way as in the S3 system developed by Becker, Chambers, and Wilks. There is a static device, which is treated as a static canvas and objects are drawn on the device through R plotting commands. The device has a set of global parameters such as margins and layouts which can be manipulated by the user using `par()` commands. The R graphics engine does not maintain a user visible graphics list, and there is no system of double buffering, so objects cannot be easily edited without redrawing a whole plot. This situation may change in R 2.7.x, where developers are working on double buffering for R devices. Even so, the base R graphics can produce many plots with extremely fine graphics in many specialized instances.

One can quickly run into trouble with R's base graphic system if one wants to design complex layouts where scaling is maintained properly on resizing, nested graphs are desired or more interactivity is needed. grid was designed by Paul Murrell to overcome some of these limitations and as a result packages like lattice, ggplot2, vcd or hexbin use grid for the underlying primitives. When using plots designed with grid one needs to keep in mind that grid is based on a system of viewports and graphic objects. To add objects one needs to use grid commands, e.g., `grid.polygon()` rather than `polygon()`. Also grid maintains a stack of viewports from the device and one needs to make sure the desired viewport is at the top of the stack. There is a great deal of explanatory documentation included with grid as vignettes.

The graphics packages in R can be organized roughly into the following topics, which range from the more user oriented at the top to the more developer oriented at the bottom. The categories are not mutually exclusive but are for the convenience of presentation:

- *Plotting* : Enhancements for specialized plots can be found in plotrix, for polar plotting, vcd for categorical data, hexbin for hexagon binning, gclus for ordering plots and gplots for some plotting enhancements. Some specialized graphs, like Chernoff faces are implemented in aplpack, which also has a nice implementation of Tukey's bag plot. For 3D plots lattice, scatterplot3d and misc3d provide a selection of plots for different kinds of 3D plotting. scatterplot3d is based on R's base graphics system, while misc3d is based on rgl. The package onion for visualizing quaternions and octonions is well suited to display 3D graphics based on derived meshes.
- *Graphic Applications* : This area is not much different from the plotting section except that these packages have tools that may not for display, but can aid in creating effective displays. Also included are packages with more esoteric plotting methods. For specific subject areas, like maps, or clustering the excellent task views contributed by other dedicated useRs is an excellent place to start.
    - *Effect ordering* : The gclus package focuses on the ordering of graphs to accentuate cluster structure or natural ordering in the data. While not for graphics directly cba and seriation have functions for creating 1 dimensional orderings from higher dimensional criteria. For ordering an array of displays, biclust can be useful.

    - *Large Data Sets* : Large data sets can present very different challenges from moderate and small datasets. Aside from overplotting, rendering 1,000,000 points can tax even modern GPU's. For bivariate data ash can produce a bivariate smoothed histogram very quickly, and hexbin can bin bivariate data onto a hexagonal lattice, the advantage being that the irregular lines and orientation of hexagons do not create linear artifacts. For multivariate data, hexbin can be used to create a scatterplot matrix, combined with lattice. An alternative is to use scagnostics to produce a scaterplot matrix of "data about the data", and look for interesting combinations of variables.
    - *Trees and Graphs* : ape and ade4 have functions for plotting phylogenetic trees, which can be used for plotting dendrograms from clustering procedures. While these packages produce decent graphics, they do not use sophisticated algorithms for node placement, so may not be useful for very large trees. igraph has the Tilford-Rheingold algorithm implementead and is useful for plotting larger trees. diagram has facilities for flow diagrams and simple graphs. For more sophisticated graphs Rgraphviz and igraph have functions for plotting and layout, especially useful for representing large networks.
- *Graphics Systems* : lattice is built on top of the grid graphics system and is an R implementation of William Cleveland's trellis system for S-PLUS. lattice allows for building many types of plots with sophisticated layouts based on conditioning. ggplot2 is an R implementation of the system described in "A Grammar of Graphics" by Leland Wilkinson. Like lattice, ggplot2 (also built on top of grid) assists in trellis-like graphics, but allows for much more. Since it is built on the idea of a semantics for graphics there is much more emphasis on reshaping data, transformation, and assembling the elements of a plot.
- *Devices* : Whereas grid is built on top of the R graphics engine, many in the R community have found the R graphics engine somewhat inflexible and have written separate device drivers that either emphasize interactivity or plotting in various graphics formats. R base supplies devices for PostScript, PDF, JPEG and other formats. Devices on CRAN include cairoDevice which is a device based libcairo, which can actually render to many device types. The cairo device is desgned to work with RGtk2, which is an interface to the Gimp Tool Kit, similar to pyGTK2. RSvgDevice is an SVG device driver and interfaces well with with vector drawing programs. When SVG devices are for web display developers should be aware that internet explorer does not support SVG, but has their own standard. Trust Microsoft. rgl provides a device driver based on OpenGL, and is good for 3D and interactive development. Lastly, the Augsburg group supplies a set of packages that includes a Java-based device, JavaGD.
- *Colors* : The package colorspace provides a set of functions for transforming between color spaces and `mixcolor()` for mixing colors within a color space. Based on the HCL colors provided in colorspace, vcd provides a set of functions for choosing color palettes suitable for coding categorical variables (`rainbow_hcl()`) and numerical information (`sequential_hcl()`, `diverge_hcl()`). Similar types of palettes are provided in RColorBrewer. dichromat is focused on palettes for color-impaired viewers.
- *Interactive Graphics* : There are several efforts to implement interactive graphics systems that interface well with R. In an interactive system the user can interactively query the graphics on the screen with the mouse, or a moveable brush to zoom, pan and query on the device as well as link with other views of the data. rggobi embeds the GGobi interactive graphics system within R, so that one can display a data frame or several in GGobi directly from R. The package has functions to support longitudinal data, and graphs using GGobi's edge set functionality. The RoSuDA repository maintained and developed by the University of Augsburg group has two packages, iplots and iwidgets as well as their Java development environment including a Java device, JavaGD. Their interactive graphics tools contain functions for alpha blending, which produces darker shading around areas with more data. This is exceptionally useful for parallel coordinate plots where many lines can quickly obscure patterns. playwith has facilities for building interactive versions of R graphics using the cairoDevice and RGtk2. Lastly, the rgl package has mechanisms for interactive manipulation of plots, especially 3D rotations and surfaces.
- *Development* : For development of specialized graphics packages in R, grid should probably be the first consideration for any new plot type. rgl has better tools for 3D graphics, since the device is interactive, though it can be slow. An alternative is to use Java and the Java device in the RoSuDA packages, though Java has its own drawbacks. For porting plotting code to grid, using the package gridBase presents a nice intermediate step to embed base graphics in grid graphics and vice versa.

CRAN packages:

- ade4
- animation
- ape
- aplpack
- ash
- biclust
- Cairo
- cairoDevice
- cba
- colorspace
- diagram
- dichromat
- gclus
- ggplot2 (core)
- gplots
- gridBase
- hexbin
- IDPmisc
- igraph
- iplots
- JavaGD
- klaR
- lattice (core)
- latticeExtra
- misc3d
- onion
- playwith
- plotrix (core)
- RColorBrewer (core)
- rggobi
- rgl (core)
- RGraphics
- RGtk2
- RSvgDevice
- RSVGTipsDevice
- scagnostics
- scatterplot3d
- seriation
- tkrplot
- vcd (core)
- vioplot
- xgobi

Related links:

- CRAN Task View: Cluster
- CRAN Task View: Multivariate
- CRAN Task View: Spatial
- Bioconductor Package: Rgraphviz
- R Graph Gallery
- Paul Murrell's home page
- RoSuDA R packages
- GGobi Data Visualization System