

autoplot : ready-made plots with ggplot2

Jean-Olivier Irisson

Laboratoire d'Océanographie de Villefranche (LOV)
Station Zoologique, 181 Chemin du Lazaret
06230 Villefranche-sur-Mer
irisson@normalesup.org

Keywords : Visualisation, Versatility, Multivariate data analysis

Generic functions with methods dedicated to specific data types are one of the great strengths of R. Neophytes can just use `plot` or `summary` on almost any kind of objects and something clever and appropriate happens. For example, “plotting” a linear model object provides the usual diagnostic plots: residuals vs. fitted, quantile-quantile plot, etc. Advanced users can program methods for their own data types and the rest of their code is cleaner because of it (no more convoluted function names!).

Many packages provide `plot` methods for the class of objects they implement. Those usually involve some computation with the data from the provided object and calls to base graphical functions. For the quantile-quantile plot defined in `plot.lm`, appropriate quantiles of the normal distribution are computed based on the number of data points in the model and are then plotted against the residuals of said model. This process results in a ready-to-use plot but provides only limited flexibility through a few arguments or the usual graphical parameters (set by `par`).

The package `ggplot2` provides a *grammar* to define a graphic. Plots are constructed by assembling building blocks and defining how each column of data should be represented on the plot (as different colours, different shapes, etc.). The intricacies such as colour choice, legends, axes ranges are dealt with automatically. This paves the way for greater flexibility in the definition and usage of graphical functions.

To fill the gap between `plot` methods defined using base graphics and the versatility (and good looks!) of `ggplot2`, `autoplot` is a generic for which methods that output `ggplot` objects can be defined. Once those methods are defined, anyone can “just use `autoplot`” and something appropriate happens, auto-magically.

But `autoplot` goes beyond what is already possible through base graphics, because it allows the user to alter the plot following the usual grammar of `ggplot2`. The author of the `autoplot` method implements the preliminary computation and the skeleton of the plot, then the user can override those choices to tailor the plot to his/her needs. Furthermore, because the computation and actual plotting are split between two functions (`fortify` for the computation and `autoplot` for the plot) it is even possible for the user to implement a completely different plot based on the usual diagnostic variables extracted from an object.

The package `autoplot` (<https://github.com/jiho/autoplot>) aims at providing `fortify` and `autoplot` methods for many kinds of objects. It started with objects resulting from multivariate

data analysis methods (Principal Component Analysis, Correspondence Analysis, etc.). These examples will be used to illustrate the versatility of the approach and how it can help gain better insight into the data. They will also demonstrate how the separation between computation (`fortify`) and plotting (`autoplot`) gives the opportunity to unify the output of many functions implementing the same analysis in various packages, at little coding cost.