Graphics before and after model fitting

Nicholas J. Cox
University of Durham

n.j.cox@durham.ac.uk

It is commonplace to compute various flavours of residual and predicted values after fitting many different kinds of model.

This allows production of a great variety of diagnostic graphics, used to examine the general and specific fit between data and model and to seek possible means of improving the model.

Several different graphs may be inspected in many modelling exercises, partly because each kind may be best for particular purposes, and partly because in many analyses a variety of models – in terms of functional form, choice of predictors, and so forth – may be entertained, at least briefly.

It is therefore helpful to be able to produce such graphs very rapidly.

Official Stata supplies as built-ins a bundle of commands originally written for use after `regress`:

◇ `avplot` and `avplots`

◇ `cprplot` and `acprplot`

◇ `lvr2plot`

◇ `rvfplot` and `rvpplot`.

These were introduced in Stata 3.0 in 1992 and are documented at [R] **regdiag**.

More recently, in an update to Stata 7.0 on 6 September 2001, all but the first two have been modified so that they may be used after `anova`.

Despite their many uses, this suite omits some very useful kinds of plot, while none of the commands may be used after other modelling commands.

Here the focus is on a new set of commands, which are biased to graphics useful for models predicting **continuous** response variables.

Note well other work for **categorical** dependent variable models by J. Scott Long and Jeremy Freese.

The **ideal** followed in this project is to make **minimal assumptions** about which modelling command has been issued previously.

The **down-side** for users is that if the data and the previous model results do not match the assumptions, it is possible to get either bizarre results or an error message.

Programs (other than those in progress) may be downloaded from SSC.

The principles followed include

◇ as far as possible, the command name by itself should produce a useful plot

◇ `predict` is used to produce temporary variables for residuals, fitted values, etc.

◇ each graph refers to the last model fitted

◇ each graph has reasonably smart default axis titles, etc.

◇ options are provided for key needs, especially `separate()` to produce separate predicted curves given dummy variables

The commands which have been written include

1. `anovaplot`

   shows fitted or predicted values from an immediately previous one-, two- or three-way `anova`. By default the data for the response are also plotted.

   In particular, `anovaplot` can show interaction plots.

2. `indexplot`

   plots estimation results (by default whatever `predict` produces by default) from an immediately previous `regress` or similar command versus a numeric index or identifier variable, if that is supplied, or observation number, if that is not supplied.

   Values are shown, by default, as vertical spikes starting at 0.

3. `ovfplot`

plots observed vs fitted or predicted values for the response from an immediately previous `regress` or similar command, with by default a line of equality superimposed.

4. `qfrplot`

plots quantile plots of fitted values, minus their mean, and residuals from the previous estimation command.

Fitted values are whatever `predict` produces by default and residuals are whatever `predict, res` produces. Comparing the distributions gives an overview of their variability and some idea of their fine structure. By default plots are side-by-side.

Quantile plots may be observed vs normal (Gaussian).

(work in progress: separate plots)

5. `rdplot`

graphs residual distributions. The residuals are, by default, those calculated by `predict, residuals` or (if the previous estimation command was `glm`) by `predict, response`.

The graph by default is a single or multiple dotplot, as produced by `dotplot`: histograms or box plots may be selected by specifying either the `histogram` or the `box` option.

6. `regplot`

plots fitted or predicted values from an immediately previous **`regress`** or similar command. By default the data for the response are also plotted.

With one syntax, no *varname* is specified. **`regplot`** shows the response and predicted values on the y axis and the covariate named first in the **`regress`** or similar command on the x axis. Thus with this syntax the plot shown is sensitive to the order in which covariates are specified in the estimation command.

With another syntax, a *varname* is supplied, which may name any numeric variable. This is used as the variable on the x axis.

Thus in practice `regplot` is most useful when the fitted values are a smooth function of the variable shown on the x axis, or a set of such functions given also one or more dummy variables as covariates.

However, other applications also arise, such as plotting observed and predicted values from a time series model versus time.

7. `rvfplot2`

graphs a residual-versus-fitted plot, a graph of the residuals versus the fitted values.

The residuals are, by default, those calculated by `predict, residuals` or (if the previous estimation command was `glm`) by `predict, response`.

The fitted values are those produced by `predict` by default after each estimation command. `rvfplot2` is offered as a generalisation of `rvfplot` in official Stata.

(work in progress: `rac` graphs residual autocorrelations)

In earlier work, two commands were produced with the main aim of producing graphs useful **before** model fitting.

`sparl`, the name being a strained acronym for scatter plot and regression line, was designed to display basic regression results, including nicely formatted summary statistics, log transformations on the fly, quadratic fit and confidence interval options, and so forth. Originally designed to be very simple, this command is currently suffering from featuritis. (A much simpler and loosely similar program is included in StataQuest.)

`tsgraph` (with Kit Baum) shows time series plots. It depends on data having been `tsset`, and correspondingly is smart about the time variable (and where relevant the panel variable).