mfp postestimation — Postestimation tools for mfp

Postestimation commands

The following postestimation commands are of special interest after `mfp`:

Command	Description
`fracplot`	plot data and fit from most recently fit fractional polynomial model
`fracpred`	create variable containing prediction, deviance residuals, or SEs of fitted values

The following standard postestimation commands are also available if available after *regression_cmd*:

Command	Description
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`linktest`	link test for model specification
`lrtest`	likelihood-ratio test
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

fracplot and fracpred

Description for fracplot and fracpred

`fracplot` plots the data and fit, with 95% confidence limits, from the most recently fit fractional polynomial (FP) model. The data and fit are plotted against *varname*, which may be $xvar_1$ or another of the covariates ($xvar_2$, ..., or a variable from *xvarlist*). If *varname* is not specified, $xvar_1$ is assumed.

`fracpred` creates *newvar* containing the fitted index or deviance residuals for the whole model, or the fitted index or its standard error for *varname*, which may be $xvar_1$ or another covariate.

Menu for fracplot and fracpred

fracplot

Statistics > Linear models and related > Fractional polynomials > Multivariable fractional polynomial plot

fracpred

Statistics > Linear models and related > Fractional polynomials > Multivariable fractional polynomial prediction

Syntax for fracplot and fracpred

Plot data and fit from most recently fit fractional polynomial model

`fracplot` [*varname*] [*if*] [*in*] [, *fracplot_options*]

Create variable containing the prediction, deviance residuals, or SEs of fitted values

`fracpred` *newvar* [, *fracpred_options*]

fracplot_options	Description
Plot	
marker_options	change look of markers (color, size, etc.)
marker_label_options	add marker labels; change look or position
Fitted line	
`lineopts(`*cline_options*`)`	affect rendition of the fitted line
CI plot	
`ciopts(`*area_options*`)`	affect rendition of the confidence bands
Add plots	
`addplot(`*plot*`)`	add other plots to the generated graph
Y axis, X axis, Titles, Legend, Overall	
twoway_options	any options other than `by()` documented in [G-3] ***twoway_options***

fracpred_options	Description
`for(`*varname*`)`	compute prediction for *varname*
`dresid`	compute deviance residuals
`stdp`	compute standard errors of the fitted values *varname*

`fracplot` is not allowed after `mfp` with `clogit`, `mlogit`, or `stcrreg`. `fracpred, dresid` is not allowed after `mfp` with `clogit`, `mlogit`, or `stcrreg`.

Options for fracplot

Plot

marker_options affect the rendition of markers drawn at the plotted points, including their shape, size, color, and outline; see [G-3] ***marker_options***.

marker_label_options specify if and how the markers are to be labeled; see [G-3] ***marker_label_options***.

Fitted line

`lineopts(`*cline_options*`)` affect the rendition of the fitted line; see [G-3] ***cline_options***.

CI plot

`ciopts(`*area_options*`)` affect the rendition of the confidence bands; see [G-3] ***area_options***.

Add plots

`addplot(`*plot*`)` provides a way to add other plots to the generated graph. See [G-3] ***addplot_option***.

Y axis, X axis, Titles, Legend, Overall

twoway_options are any of the options documented in [G-3] ***twoway_options***, excluding `by()`. These include options for titling the graph (see [G-3] ***title_options***) and for saving the graph to disk (see [G-3] ***saving_option***).

Options for fracpred

`for(`*varname*`)` specifies (partial) prediction for variable *varname*. The fitted values are adjusted to the value specified by the `center()` option in `mfp`.

`dresid` specifies that deviance residuals be calculated.

`stdp` specifies calculation of the standard errors of the fitted values *varname*, adjusted for all the other predictors at the values specified by `center()`.

Remarks and examples

stata.com

`fracplot` actually produces a component-plus-residual plot. For normal-error models with constant weights and one covariate, this amounts to a plot of the observations with the fitted line inscribed. For other normal-error models, weighted residuals are calculated and added to the fitted values.

For models with additional covariates, the line is the partial linear predictor for the variable in question ($xvar_1$ or a covariate) and includes the intercept β_0.

For generalized linear and Cox models, the fitted values are plotted on the scale of the “index” (linear predictor). Deviance residuals are added to the (partial) linear predictor to give component-plus-residual values. These values are plotted as small circles.

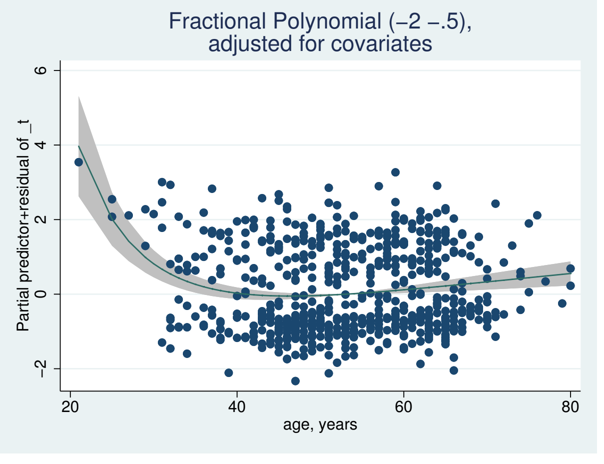
▷ Example 1

In example 1 of [R] **mfp**, we used Cox regression to predict the log hazard of breast cancer recurrence from prognostic factors of which five are continuous (`x1`, `x3`, `x5`, `x6`, `x7`) and three are binary (`x2`, `x4a`, `x4b`). We also controlled for hormonal therapy (`hormon`). We used `mfp` to build a model from the initial set of eight predictors by using the backfitting model-selection algorithm. The nominal p-value for variable and FP selection was set to 0.05 for all variables except `hormon`, which is set to 1.

```
. use http://www.stata-press.com/data/r15/brcancer
(German breast cancer data)
. stset rectime, fail(censrec)
  (output omitted )
. mfp, alpha(.05) select(.05, hormon:1): stcox x1 x2 x3 x4a x4b x5 x6 x7 hormon,
> nohr
  (output omitted )
```

We can use `fracplot` to produce component-plus-residual plots of the continuous variables. We produce the component-plus-residual plot for `x1` with `fracplot` by specifying `x1` after the command name.

```
. fracplot x1
```



We use `fracpred` with the `stdp` option to predict the standard error of the fractional polynomial prediction for `x1`. The standard error prediction will be stored in variable `sepx1`. We specify that prediction is made for `x1` with the `for()` option. After prediction, we use `summarize` to show how the standard error estimate varies over different values of `x1`.

```
. fracpred sepx1, stdp for(x1)
. summarize sepx1
```

Variable	Obs	Mean	Std. Dev.	Min	Max
sepx1	686	.0542654	.0471993	.0003304	.6862065

◁

Methods and formulas

The general definition of an FP, accommodating possible repeated powers, may be written for functions $H_1(x), \ldots, H_m(x)$ as

$$\beta_0 + \sum_{j=1}^{m} \beta_j H_j(x)$$

where $H_1(x) = x^{(p_1)}$ and for $j = 2, \ldots, m$,

$$H_j(x) = \begin{cases} x^{(p_j)} & \text{if } p_j \neq p_{j-1} \\ H_{j-1}(x)\log x & \text{if } p_j = p_{j-1} \end{cases}$$

For example, an FP of degree 3 with powers $(1, 3, 3)$ has $H_1(x) = x$, $H_2(x) = x^3$, and $H_3(x) = x^3 \log x$ and equals $\beta_0 + \beta_1 x + \beta_2 x^3 + \beta_3 x^3 \log x$.

The component-plus-residual values graphed by `fracplot` are calculated as follows: Let the data consist of triplets $(y_i, x_i, \mathbf{z}_i)$, $i = 1, \ldots, n$, where $\mathbf{z}_i$ is the vector of covariates for the ith observation, after applying possible fractional polynomial transformation and adjustment as described earlier. Let $\widehat{\eta}_i = \widehat{\beta}_0 + \{\mathbf{H}(x_i) - \mathbf{H}(x_0)\}' \widehat{\boldsymbol{\beta}} + \mathbf{z}_i' \widehat{\boldsymbol{\gamma}}$ be the linear predictor from the FP model, as given by the `fracpred` command or, equivalently, by the `predict` command with the `xb` option, following `mfp`. Here $\mathbf{H}(x_i) = \{H_1(x_i), \ldots, H_m(x_i)\}'$ is the vector of FP functions described above, $\mathbf{H}(x_0) = \{H_1(x_0), \ldots, H_m(x_0)\}'$ is the vector of adjustments to x_0 (often, x_0 is chosen to be the mean of the x_i), $\widehat{\boldsymbol{\beta}}$ is the estimated parameter vector, and $\widehat{\boldsymbol{\gamma}}$ is the estimated parameter vector for the covariates. The values $\widehat{\eta}_i^* = \widehat{\beta}_0 + \{\mathbf{H}(x_i) - \mathbf{H}(x_0)\}' \widehat{\boldsymbol{\beta}}$ represent the behavior of the FP model for x at fixed values $\mathbf{z} = \mathbf{0}$ of the (adjusted) covariates. The ith component-plus-residual is defined as $\widehat{\eta}_i^* + d_i$, where d_i is the deviance residual for the ith observation. For normal-errors models, $d_i = \sqrt{w_i}(y_i - \widehat{\eta}_i)$, where w_i is the case weight (or 1, if *weight* is not specified). For logistic, Cox, and generalized linear regression models, see [R] **logistic**, [R] **probit**, [ST] **stcox**, and [R] **glm** for the formula for d_i. The formula for `poisson` models is the same as that for `glm` with `family(poisson)`. For `stcox`, d_i is the partial martingale residual (see [ST] **stcox postestimation**).

`fracplot` plots the values of d_i and the curve represented by $\widehat{\eta}_i^*$ against x_i. The confidence interval for $\widehat{\eta}_i^*$ is obtained from the variance–covariance matrix of the entire model and takes into account the uncertainty in estimating β_0, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ (but not in estimating the FP powers for x).

`fracpred` with the `for(`*varname*`)` option calculates the predicted index at $x_i = x_0$ and $\mathbf{z}_i = \mathbf{0}$; that is, $\widehat{\eta}_i = \widehat{\beta}_0 + \{\mathbf{H}(x_i) - \mathbf{H}(x_0)\}' \widehat{\boldsymbol{\beta}}$. The standard error is calculated from the variance–covariance matrix of $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}})$, again ignoring estimation of the powers.

Also see

[R] **mfp** — Multivariable fractional polynomial models

[U] 20 Estimation and postestimation commands