fp postestimation — Postestimation tools for fp

Postestimation commands

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

Command	Description
`fp plot`	component-plus-residual plot from most recently fit fractional polynomial model
`fp predict`	create variable containing prediction or SEs of fractional polynomials

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

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat ic`	Akaike's, consistent Akaike's, corrected Akaike's, and Schwarz's Bayesian information criteria (AIC, CAIC, AICc, and BIC, respectively)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`forecast`	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`linktest`	link test for model specification
`lrtest`	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`suest`	seemingly unrelated estimation
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

The behavior of `predict` following `fp` is determined by *est_cmd*. See the corresponding *est_cmd* postestimation entry for available `predict` options.

Also see information on `fp predict` below.

margins

The behavior of `margins` following `fp` is determined by *est_cmd*. See the corresponding *est_cmd* postestimation entry for available `margins` options.

fp plot and fp predict

Description for fp plot and fp predict

`fp plot` produces a component-plus-residual plot. The fractional polynomial comprises the component, and the residual is specified by the user in `residuals()`. The component-plus-residuals are plotted against the fractional polynomial variable. If you only want to plot the component fit, without residuals, you would specify `residuals(none)`.

`fp predict` generates the fractional polynomial or the standard error of the fractional polynomial. The fractional polynomial prediction is equivalent to the fitted values prediction given by `predict, xb`, with the covariates other than the fractional polynomial variable set to zero. The standard error may be quite large if the range of the other covariates is far from zero. In this situation, the covariates would be centered and their range would include, or come close to including, zero.

These postestimation commands can be used only when the fractional polynomial variables do not interact with other variables in the specification of *est_cmd*. See **[U] 11.4.3 Factor variables** for more information about interactions.

Menu for fp plot and fp predict

fp plot

Statistics > Linear models and related > Fractional polynomials > Component-plus-residual plot

fp predict

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

Syntax for fp plot and fp predict

Component-plus-residual plot for most recently fit fractional polynomial model

`fp plot` [*if*] [*in*], `residuals(`*res_option*`)` [*graph_options*]

Create variable containing the prediction or SEs of fractional polynomials

`fp predict` [*type*] *newvar* [*if*] [*in*] [, *predict_options*]

graph_options	Description
Main	
* `residuals(`*res_option*`)`	residual option name to use in `predict` after *est_cmd*, or `residuals(none)` if residuals are not to be graphed
`equation(`*eqno*`)`	specify equation
`level(`#`)`	set confidence level; default is `level(95)`
Plot	
`plotopts(`*scatter_options*`)`	affect rendition of the component-plus-residual scatter points
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***

*`residuals(`*res_option*`)` is required.

predict_options	Description
Main	
`fp`	calculate the fractional polynomial; the default
`stdp`	calculate the standard error of the fractional polynomial
`equation(`*eqno*`)`	specify equation

Options for fp plot

Main

`residuals(`*res_option*`)` specifies what type of residuals to plot in the component-plus-residual plot. *res_option* is the same option that would be specified to `predict` after *est_cmd*. Residuals can be omitted from the plot by specifying `residuals(none)`. `residuals()` is required.

`equation(`*eqno*`)` is relevant only when you have previously fit a multiple-equation model in *est_cmd*. It specifies the equation to which you are referring.

`equation(#1)` would mean that the calculation is to be made for the first equation, `equation(#2)` would mean the second, and so on. You could also refer to the equations by their names: `equation(income)` would refer to the equation named `income`, and `equation(hours)` would refer to the equation named `hours`.

If you do not specify `equation()`, the results are the same as if you specified `equation(#1)`.

`level(`#`)`; see [R] **Estimation options**.

Plot

`plotopts(`*scatter_options*`)` affects the rendition of the component-plus-residual scatter points; see [G-2] **graph twoway scatter**.

Fitted line

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

CI plot

`ciopts(`*area_options*`)` affects 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 fp predict

Main

`fp` calculates the fractional polynomial, the linear prediction with other variables set to zero. This is the default.

`stdp` calculates the standard error of the fractional polynomial.

`equation(`*eqno*`)` is relevant only when you have previously fit a multiple-equation model in *est_cmd*. It specifies the equation to which you are referring.

`equation(#1)` would mean that the calculation is to be made for the first equation, `equation(#2)` would mean the second, and so on. You could also refer to the equations by their names: `equation(income)` would refer to the equation named `income`, and `equation(hours)` would refer to the equation named `hours`.

If you do not specify `equation()`, the results are the same as if you specified `equation(#1)`.

Remarks and examples

After a model is fit using `fp`, the estimated fractional polynomial may be of interest. This is the linear combination of the fractional polynomial terms and the constant intercept using the model coefficients estimated by `fp`. It is equivalent to the fitted values prediction given by `predict,xb`, with the covariates and the fractional polynomial variable set to zero. When these other covariates have been centered, the prediction is made at the centering values of the covariates.

A component-plus-residual plot is generated by `fp plot`. The fractional polynomial comprises the component, and the residual is specified by the user in `residuals()`. The `residuals()` option takes the same argument that would be supplied to `predict` after *est_cmd* to obtain the desired type of residuals. If you only want to plot the component fit, without residuals, you would specify `residuals(none)`.

`fp predict` generates the fractional polynomial. If the `stdp` option is specified, the standard error of the fractional polynomial is generated instead. This standard error may be quite large if the range of the other covariates is far from zero. In this situation, the covariates would be centered and their range would include, or come close to including, zero.

These postestimation commands can be used only when the fractional polynomial terms do not interact with other variables in the specification of *est_cmd*. See **[U] 11.4.3 Factor variables** for more information about interactions.

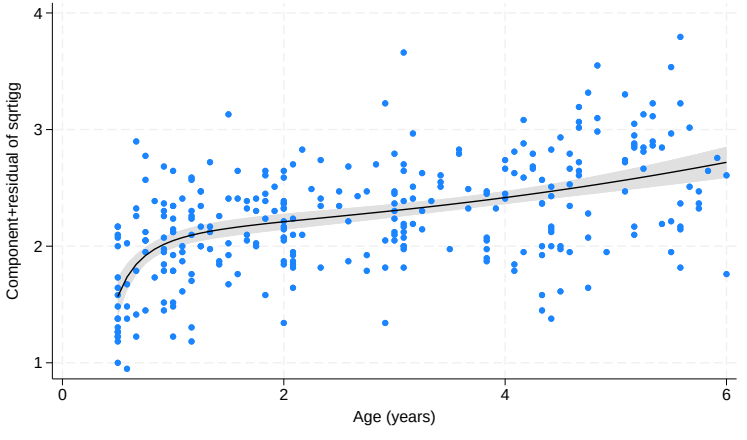
Examples

▷ Example 1: fp plot after linear regression

In example 1 of [R] **fp**, we modeled the mean of the square root of a child's serum immunoglobulin G (IgG) level as a fractional polynomial function of the child's age. An FP2 model with powers $(-2, 2)$ is chosen.

We load the data and then fit the model with `fp`. Then, we use `fp plot` to draw the component-plus-residual plot. A 95% confidence interval is produced for the fractional polynomial in age (the component). The `residuals` prediction option for `regress` is specified in the `residuals()` option in `fp plot` so that the residuals are rendered.

```
. use https://www.stata-press.com/data/r19/igg
(Immunoglobulin in children)
. fp <age>, scale center: regress sqrtigg <age>
  (output omitted)
. fp plot, residuals(residuals)
```



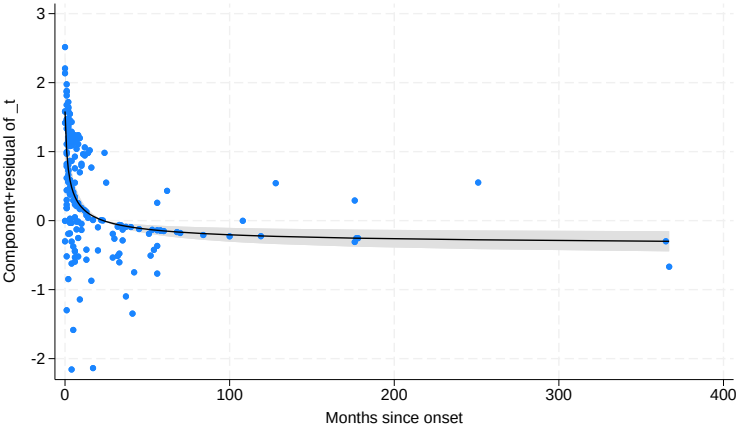
◁

Example 2: fp plot after Cox regression

In example 2 of [R] **fp**, we modeled the time to complete healing of leg ulcers for 192 elderly patients using a Cox regression. A one-degree fractional polynomial in `mthson`, the number of months since the onset of the ulcer, is used as a predictor in the regression. The power −0.5 is used for `mthson`. Other covariates are age (`age`), ulcer area (`ulcarea`), treatment type, and a binary indicator of deep vein involvement (`deepppg`).

We load the data and then demean ulcer area and age. Then, we fit the model with `fp` and draw the component-plus-residual plot with `fp plot`. `mgale` is specified in the `residuals()` option to obtain martingale residuals. See [ST] **stcox postestimation** for more details.

```
. use https://www.stata-press.com/data/r19/legulcer2, clear
(Leg ulcer clinical trial)
. quietly stset ttevent, failure(healed)
. quietly summarize age
. replace age = age - r(mean)
variable age was byte now float
(192 real changes made)
. quietly summarize ulcarea
. replace ulcarea = ulcarea - r(mean)
variable ulcarea was int now float
(192 real changes made)
. fp <mthson>, replace center scale nohr fp(-.5): stcox <mthson> age ulcarea
> deepppg treat
  (output omitted)
. fp plot, residuals(mgale)
```



Example 3: fp plot and fp predict after logistic regression

In example 3 of [R] **fp**, we used logistic regression to model the odds of death for male civil servants in Britain conditional on cigarette consumption. The dependent variable `all10` is an indicator of whether the individual passed away in the 10 years under study.

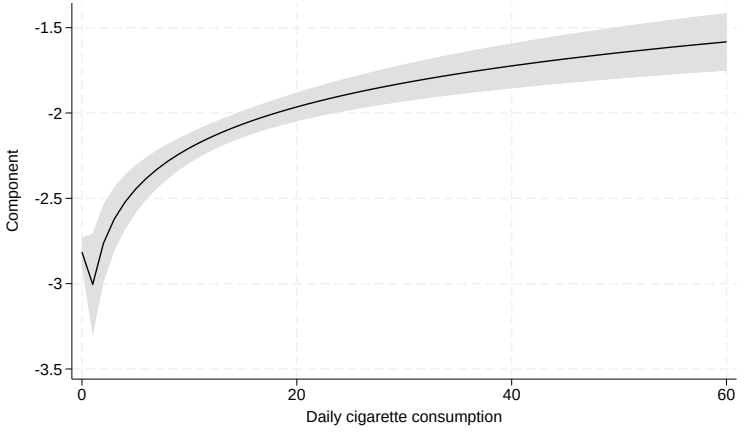
Nonsmokers may be qualitatively different from smokers, so the effect of smoking (regarded as a continuous variable) may not be continuous between zero cigarettes and one cigarette. To allow for this possibility, we model the risk as constant intercept for the nonsmokers and as a fractional polynomial function of the number of cigarettes for the smokers, `cigs`, adjusted for age. An FP1 model with power 0 is chosen.

We load the data and demean age. Then, we fit the model using `fp` and graph the fit of the model and 95% confidence interval using `fp plot`. Only the component fit is graphed by specifying `residuals(none)`.

```
. use https://www.stata-press.com/data/r19/smoking, clear
(Smoking and mortality data)
. quietly summarize age
. replace age = age - r(mean)
variable age was byte now float
(17,260 real changes made)
. fp <cigs>, catzero replace fp(0): logit all10 <cigs> age
-> logit all10 cigs_0 cigs_1 age
Logistic regression                                     Number of obs =  17,260
                                                        LR chi2(3)    = 1027.13
                                                        Prob > chi2   =  0.0000
Log likelihood = -4973.3016                             Pseudo R2     =  0.0936
```

all10	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
cigs_0	.1883732	.1553093	1.21	0.225	-.1160274	.4927738
cigs_1	.3469842	.0543552	6.38	0.000	.2404499	.4535185
age	.1194976	.0045818	26.08	0.000	.1105174	.1284778
_cons	-3.003767	.1514909	-19.83	0.000	-3.300683	-2.70685

```
. fp plot, residuals(none)
```



We see a small spike at zero for `cigs` because of the effect of `cigs_0` on the fractional polynomial; however, the high p-value for `cigs_0` in the model output indicates that we cannot reject that there is no extra effect at zero for nonsmokers.

We can also use `fp predict` to predict the fractional polynomial for nonsmokers and the mean of age. This is the value at the spike. We store the result in `fp0`. We see it is equivalent to the sum of the constant intercept estimate and the estimate of the `cigs_0` coefficient.

```
. fp predict fp0 if cigs == 0
(7,157 missing values generated)
. summarize fp0
    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
         fp0 |     10,103   -2.815393           0  -2.815393  -2.815393
. display _b[cigs_0]+_b[_cons]
-2.8153935
```

Methods and formulas

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 and x_i is the fractional polynomial variable.

`fp predict` calculates the fractional polynomial at the centering value x_0, $\widehat{\eta}_i = ({x_i}^{(p_1,\ldots,p_d)} - {x_0}^{(p_1,\ldots,p_m)})'\widehat{\boldsymbol{\beta}}$. This is equivalent to the linear predictor of the model at $\mathbf{z}_i = \mathbf{0}$. The standard error is calculated from the variance–covariance matrix of $\widehat{\boldsymbol{\beta}}$, ignoring estimation of the powers. When $x_i \leq 0$, $\mathbf{H}(x_i)$, and thus ${x_i}^{(p_1,\ldots,p_m)}$, is either undefined or zero. A zero offset term, α_0, may be added to $\widehat{\eta}_i$ for these nonpositive x_i values.

The values $\widehat{\eta}_i$ represent the behavior of the fractional polynomial model for x at fixed values $\mathbf{z} = \mathbf{0}$ of the (centered) covariates. The ith component-plus-residual is defined as $\widehat{\eta}_i + d_i$, where d_i is the residual for the ith observation. The definition of d_i will change according to the type of model used and the preference of the user. `fp plot` plots $\widehat{\eta}_i + d_i$ versus x_i, overlaying $\widehat{\eta}_i$ and its confidence interval.

Acknowledgment

We thank Patrick Royston of the MRC Clinical Trials Unit, London, and coauthor of the Stata Press book *Flexible Parametric Survival Analysis Using Stata: Beyond the Cox Model* for writing `fracplot` and `fracpred`, the commands on which `fp plot` and `fp predict` are based. We also thank Professor Royston for his advice on and review of `fp plot` and `fp predict`.

Reference

Royston, P. 2017. Model selection for univariable fractional polynomials. *Stata Journal* 17: 619–629.

Also see

[R] **fp** — Fractional polynomial regression

[U] 20 Estimation and postestimation commands