#### **mfp** — Multivariable fractional polynomial models

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Acknowledgments
References	Also see		

### Description

mfp selects the multivariable fractional polynomial (MFP) model that best predicts the outcome variable from the right-hand-side variables in *xvarlist*.

For univariate fractional polynomials, fp can be used to fit a wider range of models than mfp. See [R] fp for more details.

# Quick start

Find optimal MFP model for regression of y on fractional polynomials of x1, x2, and x3 mfp: regress y x1 x2 x3

Same as above, but search only powers of -1, -0.5, 1, and 2

mfp, xpowers(-1 - .5 1 2): regress y x1 x2 x3

Allow a maximum of 2 degrees of freedom for all covariates mfp, dfdefault(2): regress y x1 x2 x3

Allow 4 degrees of freedom for x1 and 2 degrees of freedom for x2 and x3 mfp, dfdefault(2) df(x1:4): regress y x1 x2 x3

Same as above

mfp, df(x1:4, x2 x3:2): regress y x1 x2 x3

- Use a 10% significance level when testing between fractional polynomials of different degrees mfp, alpha(0.1): regress y x1 x2 x3
- Perform backward selection using a nominal *p*-value of 0.05 for all variables mfp, select(0.05): regress y x1 x2 x3
- Same as above, but force x3 into the model by setting its nominal *p*-value to 1 mfp, select(0.05, x3:1): regress y x1 x2 x3

Note: In the above examples, regress could be replaced with any estimation command allowing the mfp prefix.

### Menu

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

## **Syntax**

```
mfp [, options]: regression_cmd [yvar<sub>1</sub> [yvar<sub>2</sub>]] xvarlist [if] [in] [weight]
```

[, regression\_cmd\_options]

```
regression_cmd may be clogit, glm, intreg, logistic, logit, mlogit, nbreg, ologit, oprobit,
poisson, probit, qreg, regress, rreg, stcox, stcrreg, streg, or xtgee.
```

 $yvar_1$  is not allowed for streg, stcrreg, and stcox. For these commands, you must first stset your data.

yvar<sub>1</sub> and yvar<sub>2</sub> must both be specified when *regression\_cmd* is intreg.

*xvarlist* has elements of type *varlist* or (*varlist*) or both, for example, x1 x2 (x3 x4 x5). Elements enclosed in parentheses are tested jointly for inclusion in the model and are not eligible for fractional polynomial transformation.

options	Description
Model 2	
<u>sequential</u>	use the Royston and Altman model-selection algorithm; default uses closed-test procedure
cycles(#)	maximum number of iteration cycles; default is cycles(5)
dfdefault(#)	default maximum degrees of freedom; default is dfdefault(4)
<pre>center(cent_list)</pre>	specification of centering for the independent variables
<u>al</u> pha( <i>alpha_list</i> )	<i>p</i> -values for testing between FP models; default is alpha(0.05)
df ( <i>df_list</i> )	degrees of freedom for each predictor
powers(numlist)	list of FP powers to use; default is powers (-2 -1(.5)123)
Adv. model	
<u>xo</u> rder(+ - n)	order of entry into model-selection algorithm; default is xorder(+)
<u>sel</u> ect( <i>select_list</i> )	nominal <i>p</i> -values for selection on each predictor
<pre>xpowers(xp_list)</pre>	FP powers for each predictor
<u>zer</u> o(varlist)	treat nonpositive values of specified predictors as zero when FP is transformed
<u>cat</u> zero( <i>varlist</i> )	add indicator variable for specified predictors
all	include out-of-sample observations in generated variables
Reporting	
<u>lev</u> el(#)	set confidence level; default is level(95)
display_options	control column formats and line width
regression_cmd_options	Description
Adv. model	
regression_cmd_options	options appropriate to the regression command in use

collect is allowed; see [U] 11.1.10 Prefix commands.

All weight types supported by regression\_cmd are allowed; see [U] 11.1.6 weight.

See [U] 20 Estimation and postestimation commands for more capabilities of estimation commands.

fp generate may be used to create new variables containing fractional polynomial powers. See [R] fp.

# Options

Model 2

- sequential chooses the sequential fractional polynomial (FP) selection algorithm (see Methods of FP
  model selection).
- cycles(#) sets the maximum number of iteration cycles permitted. cycles(5) is the default.
- dfdefault(#) determines the default maximum degrees of freedom (df) for a predictor. The default is dfdefault(4) (second-degree FP).
- center (*cent\_list*) defines the centering of the covariates *xvar*<sub>1</sub>, *xvar*<sub>2</sub>, ... of *xvarlist*. The default is center (mean), except for binary covariates, where it is center (#), with # being the lower of the two distinct values of the covariate. A typical item in *cent\_list* is *varlist*: {mean | # | no}. Items are separated by commas. The first item is special in that *varlist* is optional, and if it is omitted, the default is reset to the specified value (mean, #, or no). For example, center (no, age:mean) sets the default to no (that is, no centering) and the centering of age to mean.
- alpha(*alpha\_list*) sets the significance levels for testing between FP models of different degrees. The rules for *alpha\_list* are the same as those for *df\_list* in the df() option (see below). The default nominal *p*-value (significance level, selection level) is 0.05 for all variables.

Example: alpha(0.01) specifies that all variables have an FP selection level of 1%.

Example: alpha(0.05, weight:0.1) specifies that all variables except weight have an FP selection level of 5%; weight has a level of 10%.

df (*df\_list*) sets the df for each predictor. The df (not counting the regression constant, \_cons) is twice the degree of the FP, so, for example, an *xvar* fit as a second-degree FP (FP2) has 4 df. The first item in *df\_list* may be either # or *varlist* : #. Subsequent items must be *varlist* : #. Items are separated by commas, and *varlist* is specified in the usual way for variables. With the first type of item, the df for all predictors is taken to be #. With the second type of item, all members of *varlist* (which must be a subset of *xvarlist*) have # df.

The default number of degrees of freedom for a predictor of type *varlist* specified in *xvarlist* but not in *df\_list* is assigned according to the number of distinct (unique) values of the predictor, as follows:

# of distinct values	Default df
1	(invalid predictor)
2-3	1
4–5	<pre>min(2, dfdefault())</pre>
$\geq 6$	dfdefault()

Example: df (4) All variables have 4 df.

Example: df(2, weight displ:4)

weight and displ have 4 df; all other variables have 2 df.

Example: df(weight displ:4, mpg:2)

weight and displ have 4 df, mpg has 2 df; all other variables have default df.

powers (*numlist*) is the set of FP powers to be used. The default set is -2, -1, -0.5, 0, 0.5, 1, 2, 3 (0 means log).

Adv. model

- xorder(+|-|n) determines the order of entry of the covariates into the model-selection algorithm. The default is xorder(+), which enters them in decreasing order of significance in a multiple linear regression (most significant first). xorder(-) places them in reverse significance order, whereas xorder(n) respects the original order in *xvarlist*.
- select (select\_list) sets the nominal p-values (significance levels) for variable selection by backward elimination. A variable is dropped if its removal causes a nonsignificant increase in deviance. The rules for select\_list are the same as those for  $df_{list}$  in the df() option (see above). Using the default selection level of 1 for all variables forces them all into the model. Setting the nominal p-value to be 1 for a given variable forces it into the model, leaving others to be selected or not. The nominal p-value for elements of xvarlist bound by parentheses is specified by including (varlist) in select\_list.

Example: select(0.05) All variables have a nominal *p*-value of 5%.

Example: select(0.05, weight:1) All variables except weight have a nominal *p*-value of 5%; weight is forced into the model.

Example: select(a (b c):0.05)

All variables except a, b, and c are forced into the model. b and c are tested jointly with 2 df at the 5% level, and a is tested singly at the 5% level.

xpowers  $(xp\_list)$  sets the permitted FP powers for covariates individually. The rules for  $xp\_list$  are the same as for  $df\_list$  in the df() option. The default selection is the same as that for the powers() option.

Example: xpowers  $(-1 \ 0 \ 1)$ All variables have powers -1, 0, 1.

Example: xpowers (x5:-1 0 1) All variables except x5 have default powers; x5 has powers -1, 0, 1.

- zero(varlist) treats negative and zero values of members of varlist as zero when FP transformations are applied. By default, such variables are subjected to a preliminary linear transformation to avoid negative and zero values, as described in the scale option of [R] fp. varlist must be part of xvarlist.
- catzero(varlist) is a variation on zero(); see Zeros and zero categories below. varlist must be part of xvarlist.
- regression\_cmd\_options may be any of the options appropriate to regression\_cmd.
- all includes out-of-sample observations when generating the FP variables. By default, the generated FP variables contain missing values outside the estimation sample.

Reporting

- level(#) specifies the confidence level, as a percentage, for confidence intervals. The default is level(95) or as set by set level; see [U] 20.8 Specifying the width of confidence intervals.
- display\_options: cformat(% fmt), pformat(% fmt), sformat(% fmt), and nolstretch; see [R] Estimation options.

## **Remarks and examples**

Remarks are presented under the following headings:

Iteration report Estimation algorithm Methods of FP model selection Zeros and zero categories

For elements in *xvarlist* not enclosed in parentheses, mfp leaves variables in the data named  $Ixvar_1$ ,  $Ixvar_2$ ,..., where *xvar* represents the first four letters of the name of *xvar*<sub>1</sub>, and so on, for *xvar*<sub>2</sub>, *xvar*<sub>3</sub>, etc. The new variables contain the best-fitting FP powers of *xvar*<sub>1</sub>, *xvar*<sub>2</sub>, ....

#### Iteration report

By default, for each continuous predictor, *x*, mfp compares null, linear, and FP1 models for *x* with an FP2 model. The deviance for each of these nested submodels is given in the column labeled "Deviance". The line labeled "Final" gives the deviance for the selected model and its powers. All the other predictors currently selected are included, with their transformations (if any). For models specified as having 1 df, the only choice is whether the variable enters the model.

#### Estimation algorithm

The estimation algorithm in mfp processes the *xvars* in turn. Initially, mfp silently arranges *xvarlist* in order of increasing *p*-value (that is, of decreasing statistical significance) for omitting each predictor from the model comprising *xvarlist*, with each term linear. The aim is to model relatively important variables before unimportant ones. This approach may help to reduce potential model-fitting difficulties caused by collinearity or, more generally, "concurvity" among the predictors. See the *xorder()* option above for details on how to change the ordering.

At the initial cycle, the best-fitting FP function for  $xvar_1$  (the first of xvarlist) is determined, with all the other variables assumed to be linear. Either the default or the alternative procedure is used (see *Methods* of *FP* model selection below). The functional form (but not the estimated regression coefficients) for  $xvar_1$  is kept, and the process is repeated for  $xvar_2$ ,  $xvar_3$ , etc. The first iteration concludes when all the variables have been processed in this way. The next cycle is similar, except that the functional forms from the initial cycle are retained for all variables except the one currently being processed.

A variable whose functional form is prespecified to be linear (that is, to have 1 df) is tested for exclusion within the above procedure when its nominal p-value (selection level) according to select() is less than 1; otherwise, it is included.

Updating of FP functions and candidate variables continues until the functions and variables included in the overall model do not change (convergence). Convergence is usually achieved within 1–4 cycles.

### Methods of FP model selection

mfp includes two algorithms for FP model selection, both of which combine backward elimination with the selection of an FP function. For each continuous variable in turn, they start from a most-complex permitted FP model and attempt to simplify the model by reducing the degree. The default algorithm resembles a closed-test procedure, a sequence of tests maintaining the overall type I error rate at a prespecified nominal level, such as 5%. All significance tests are approximate; therefore, the algorithm is not precisely a closed-test procedure (Royston and Sauerbrei 2008, chap. 6).

The closed-test algorithm for choosing an FP model with maximum permitted degree m = 2 (that is, an FP2 model with 4 df) for one continuous predictor, x, is as follows:

- 1. Inclusion: Test FP2 against the null model for x on 4 df at the significance level determined by select(). If x is significant, continue; otherwise, drop x from the model.
- 2. Nonlinearity: Test FP2 against a straight line in x on 3 df at the significance level determined by alpha(). If significant, continue; otherwise, stop, with the chosen model for x being a straight line.
- 3. Simplification: Test FP2 against FP1 on 2 df at the significance level determined by alpha(). If significant, the final model is FP2; otherwise, it is FP1.

The first step is omitted if x is to be retained in the model, that is, if its nominal p-value, according to the select() option, is 1.

An alternative algorithm is available with the sequential option, as originally suggested by Royston and Altman (1994):

- 1. Test FP2 against FP1 on 2 df at the alpha() significance level. If significant, the final model is FP2; otherwise, continue.
- 2. Test FP1 against a straight line on 1 df at the alpha() level. If significant, the final model is FP1; otherwise, continue.
- 3. Test a straight line against omitting x on 1 df at the select() level. If significant, the final model is a straight line; otherwise, drop x.

The final step is omitted if x is to be retained in the model, that is, if its nominal p-value, according to the select() option, is 1.

If x is uninfluential, the overall type I error rate of this procedure is about double that of the closedtest procedure, for which the rate is close to the nominal value. This inflated type I error rate confers increased apparent power to detect nonlinear relationships.

#### Zeros and zero categories

The zero() option permits fitting an FP model to the positive values of a covariate, taking nonpositive values as zero. An application is the assessment of the effect of cigarette smoking as a risk factor in an epidemiological study. Nonsmokers may be qualitatively different from smokers, so the effect of smoking (regarded as a continuous variable) may not be continuous between one and zero cigarettes. To allow for this, the risk may be modeled as constant for the nonsmokers and as an FP function of the number of cigarettes for the smokers:

. generate byte nonsmokr = cond(n\_cigs==0, 1, 0) if n\_cigs != .
. mfp, zero(n cigs) df(4, nonsmokr:1): logit case n cigs nonsmokr age

Omission of  $zero(n_cigs)$  would cause  $n_cigs$  to be transformed before analysis by the addition of a suitable constant, probably 1.

A closely related approach involves the catzero() option. The command

. mfp, catzero(n\_cigs): logit case n\_cigs age

would achieve a similar result to the previous command but with important differences. First, mfp would create the equivalent of the binary variable nonsmokr automatically and include it in the model. Second, the two smoking variables would be treated as one predictor in the model. With the select() option active, the two variables would be tested jointly for inclusion in the model. A modified version is described in Royston and Sauerbrei (2008, sec. 4.15).

#### Example 1

We illustrate two of the analyses performed by Sauerbrei and Royston (1999). We use brcancer.dta, which contains prognostic factors data from the German Breast Cancer Study Group of patients with node-positive breast cancer. The response variable is recurrence-free survival time (rectime), and the censoring variable is censrec. There are 686 patients with 299 events. We use Cox regression to predict the log hazard of recurrence from prognostic factors of which five are continuous (x1, x3, x5, x6, x7) and three are binary (x2, x4a, x4b). Hormonal therapy (hormon) is known to reduce recurrence rates and is forced into the model. We use mfp to build a model from the initial set of eight predictors by using the backfitting model-selection algorithm. We set the nominal *p*-value for variable and FP selection to 0.05 for all variables except hormon, which it is set to 1:

```
. use https://www.stata-press.com/data/r19/brcancer
(German breast cancer data)
. stset rectime, fail(censrec)
 (output omitted)
```

. mfp, alpha > nohr	(.05) s	elect(.	05, hormon	:1): stcox	x1 x2 x3	3 x4a x4b	x5 x6 x7 hormon,
Deviance for	model	with al	l terms un	transforme	d = 3471	.637, 686	observations
Variable	Model	(vs.)	Deviance	Dev diff.	Ρ	Powers	(vs.)
x5	null Lin. FP1 Final	FP2	3503.610 3471.637 3449.203 3442.244	61.366 29.393 6.959	0.000* 0.000+ 0.031+	1 0 .5 3	.5 3
x6	null Lin. FP1 Final	FP2	3464.113 3442.244 3435.550 3435.550	29.917 8.048 1.354	0.000* 0.045+ 0.508	1 .5 .5	-2 .5
[hormon incl	uded wi	th 1 df	in model]				
x4a	null Final	Lin.	3440.749 3435.550	5.199	0.023*	1	1
х3	null Final	FP2	3436.832 3436.832	3.560	0.469		-2 3
x2	null Final	Lin.	3437.589 3437.589	0.756	0.384		1
x4b	null Final	Lin.	3437.848 3437.848	0.259	0.611		1
x1	null Lin. FP1 Final	FP2	3437.893 3437.848 3433.628 3419.808	18.085 18.040 13.820	0.001* 0.000+ 0.001+	1 -2 -25	-25
x7	null Final	FP2	3420.805 3420.805	3.715	0.446		5 3
End of Cycle	1: Dev	iance =	3420.805				
x5	null Lin. FP1 Final	FP2	3494.867 3451.795 3428.023 3420.724	74.143 31.071 7.299	0.000* 0.000+ 0.026+	1 0 -2 -1	-2 -1
x6	null Lin. FP1 Final	FP2	3452.093 3427.703 3420.724 3420.724	32.704 8.313 1.334	0.000* 0.040+ 0.513	1 .5 .5	0 0
[hormon incl	uded wi	th 1 df	in model]				
x4a	null Final	Lin.	3425.310 3420.724	4.586	0.032*	1	1
х3	null Final	FP2	3420.724 3420.724	5.305	0.257		5 0
x2	null Final	Lin.	3420.724 3420.724	0.214	0.644		1
x4b	null Final	Lin.	3420.724 3420.724	0.145	0.703		1
x1	null Lin. FP1 Final	FP2	3440.057 3440.038 3436.949 3420.724	19.333 19.314 16.225	0.001* 0.000+ 0.000+	1 -2 -25	-25
x7	null Final	FP2	3420.724 3420.724	2.152	0.708		-1 3

Fractional polynomial fitting algorithm converged after 2 cycles.

Transformations of covariates:

```
-> gen double Ix1_1 = X^-2-.0355294635 if e(sample)
```

- -> gen double Ix1\_2 = X^-.5-.4341573547 if e(sample)
   (where: X = x1/10)
- -> gen double Ix5\_\_1 = X^-2-3.983723313 if e(sample)
- -> gen double Ix5\_2 = X^-1-1.99592668 if e(sample)
   (where: X = x5/10)
- -> gen double Ix6\_\_1 = X<sup>.5-.3331600619</sup> if e(sample)
   (where: X = (x6+1)/1000)

Variable		-Initial-			Final-	
	df	Select	Alpha	Status	df	Powers
x1	4	0.0500	0.0500	in	4	-25
x2	1	0.0500	0.0500	out	0	
x3	4	0.0500	0.0500	out	0	
x4a	1	0.0500	0.0500	in	1	1
x4b	1	0.0500	0.0500	out	0	
x5	4	0.0500	0.0500	in	4	-2 -1
x6	4	0.0500	0.0500	in	2	.5
x7	4	0.0500	0.0500	out	0	
hormon	1	1.0000	0.0500	in	1	1

Final multivariable fractional polynomial model for \_t

Cox regression	n Breslow m	ethod for t	ies				
Entry time _t(	C			Numbe	r of obs	=	686
				LR ch	i2(7)	=	155.62
				Prob	> chi2	=	0.0000
Log likelihood	d = -1710.3619			Pseud	lo R2	=	0.0435
_t	Coefficient	Std. err.	Z	P> z	[95%	conf.	interval]
Ix11	44.73377	8.256682	5.42	0.000	28.55	097	60.91657
Ix12	-17.92302	3.909611	-4.58	0.000	-25.58	571	-10.26032
x4a	.5006982	.2496324	2.01	0.045	.0114	276	.9899687
Ix51	.0387904	.0076972	5.04	0.000	.0237	041	.0538767
Ix52	5490645	.0864255	-6.35	0.000	7184	554	3796736
Ix61	-1.806966	.3506314	-5.15	0.000	-2.494	191	-1.119741
hormon	4024169	.1280843	-3.14	0.002	6534	575	1513763

Deviance = 3420.724.

Some explanation of the output from the model-selection algorithm is desirable. Consider the first few lines of output in the iteration log:

1.	Deviance	for	model	with a	ll terms un	transforme	d = 347	1.637, 686	observations
	Variable		Model	(vs.)	Deviance	Dev diff.	Р	Powers	(vs.)
2.	x5		null	FP2	3503.610	61.366	0.000*		.5 3
з.			lin.		3471.637	29.393	0.000+	1	
4.			FP1		3449.203	6.959	0.031+	0	
5.			Final		3442.244			.5 3	

Line 1 gives the deviance  $(-2 \times \log \text{ partial likelihood})$  for the Cox model with all terms linear, the place where the algorithm starts. The model is modified variable by variable in subsequent steps. The most significant linear term turns out to be x5, which is therefore processed first. Line 2 compares the best-fitting FP2 for x5 with a model omitting x5. The FP has powers (0.5, 3), and the test for inclusion of x5 is highly significant. The reported deviance of 3,503.610 is of the null model, not for the FP2 model.

The deviance for the FP2 model may be calculated by subtracting the deviance difference (Dev diff.) from the reported deviance, giving 3,503.610 - 61.366 = 3,442.244. Line 3 shows that the FP2 model is also a significantly better fit than a straight line (lin.) and line 4 that FP2 is also somewhat better than FP1 (p = 0.031). Thus at this stage in the model-selection procedure, the final model for x5 (line 5) is FP2 with powers (0.5, 3). The overall model with an FP2 for x5 and all other terms linear has a deviance of 3,442.244.

After all the variables have been processed (cycle 1) and reprocessed (cycle 2) in this way, convergence is achieved because the functional forms (FP powers and variables included) after cycle 2 are the same as they were after cycle 1. The model finally chosen is Model II as given in tables 3 and 4 of Sauerbrei and Royston (1999). Because of scaling of variables, the regression coefficients reported there are different, but the model and its deviance are identical. The model includes x1 with powers (-2, -0.5), x4a, x5 with powers (-2, -1), and x6 with power 0.5. There is strong evidence of nonlinearity for x1 and for x5, the deviance differences for comparison with a straight-line model (FP2 vs lin.) being, respectively, 19.3 and 31.1 at convergence (cycle 2). Predictors x2, x3, x4b, and x7 are dropped, as may be seen from their status out in the table Final multivariable fractional polynomial model for \_t (the assumed *depvar* when using stcox).

All predictors except x4a and hormon, which are binary, have been centered on the mean of the original variable. For example, the mean of x1 (age) is 53.05 years. The first FP-transformed variable for x1 is x1^-2 and is created by the expression generate double  $Ix1_1 = X^-2 - .0355$  if e(sample). The value 0.0355 is obtained from  $(53.05/10)^{-2}$ . The division by 10 is applied automatically to improve the scaling of the regression coefficient for  $Ix1_1$ .

According to Sauerbrei and Royston (1999), medical knowledge dictates that the estimated risk function for x5 (number of positive nodes), which was based on the above FP with powers (-2, -1), should be monotonic, but it was not. They improved Model II by estimating a preliminary exponential transformation, x5e = exp $(-0.12 \cdot x5)$ , for x5 and fitting a degree 1 FP for x5e, thus obtaining a monotonic risk function. The value of -0.12 was estimated univariately using nonlinear Cox regression with the ado-file boxtid (Royston and Ambler 1999b, 1999d). To ensure a negative exponent, Sauerbrei and Royston (1999) restricted the powers for x5e to be positive. Their Model III may be fit by using the following command:

. mfp, alpha(.05) select(.05, hormon:1) df(x5e:2) xpowers(x5e:0.5 1 2 3):

> stcox x1 x2 x3 x4a x4b x5e x6 x7 hormon

Number of obs

=

Other than the customization for x5e, the command is the same as it was before. The resulting model is as reported in table 4 of Sauerbrei and Royston (1999):

```
. use https://www.stata-press.com/data/r19/brcancer, clear
(German breast cancer data)
. stset rectime, fail(censrec)
(output omitted)
. mfp, alpha(.05) select(.05, hormon:1) df(x5e:2) xpowers(x5e:0.5 1 2 3):
> stcox x1 x2 x3 x4a x4b x5e x6 x7 hormon, nohr
```

```
(output omitted)
```

Final multivariable fractional polynomial model for \_t

Variable		-Initial-			Final-	
	dī	Select	Alpha	Status	dÍ	Powers
x1	4	0.0500	0.0500	in	4	-25
x2	1	0.0500	0.0500	out	0	
x3	4	0.0500	0.0500	out	0	
x4a	1	0.0500	0.0500	in	1	1
x4b	1	0.0500	0.0500	out	0	
x5e	2	0.0500	0.0500	in	1	1
x6	4	0.0500	0.0500	in	2	.5
x7	4	0.0500	0.0500	out	0	
hormon	1	1.0000	0.0500	in	1	1

```
Cox regression -- Breslow method for ties Entry time _t0
```

Log likelihood	d = −1711.6186			LR cl Prob Pseud	ni2(6) > chi2 do R2	= = =	153.11 0.0000 0.0428
_t	Coefficient	Std. err.	Z	P> z	[95%	conf.	interval]
Ix11 Ix1_2	43.55382 -17 48136	8.253433	5.28 -4 47	0.000	27.3	7738 1851	59.73025 -9.814212
x4a Ix5e1	.5174351	.2493739	2.07	0.038	.0286	5713 5909	1.006199
1x61 hormon	-1.84008 3944998	.3508432 .128097	-5.24 -3.08	0.000	-2.52 645	2772 5654	-1.15244 1434342

Deviance = 3423.237.

686

## Stored results

In addition to what *regression\_cmd* stores, mfp stores the following in e():

Scalars	
e(fp_nx)	number of predictors in xvarlist
e(fp_dev)	deviance of final model fit
e(Fp_id#)	initial degrees of freedom for the #th element of xvarlist
e(Fp_fd#)	final degrees of freedom for the #th element of xvarlist
e(Fp_al#)	FP selection level for the #th element of xvarlist
e(Fp_se#)	backward elimination selection level for the #th element of xvarlist
Macros	
e(fp_cmd)	fracpoly
e(fp_cmd2)	mfp
e(cmdline)	command as typed
e(fracpoly)	command used to fit the selected model using fracpoly
e(fp_fvl)	variables in final model
e(fp_depv)	$yvar_1(yvar_2)$
e(fp_opts)	estimation command options
e(fp_x1)	first variable in xvarlist
e(fp_x2)	second variable in xvarlist
e(fp_xN)	last variable in xvarlist, $N = e(fp_nx)$
e(fp_k1)	power for first variable in xvarlist (*)
e(fp_k2)	power for second variable in xvarlist (*)
e(fp_kN)	power for last var. in xvarlist (*), $N = e(fp_nx)$

Note: (\*) contains '.' if the variable is not selected in the final model.

In addition to the above, the following is stored in r():

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, p-values, and
	confidence intervals

Note that results stored in r() are updated when the command is replayed and will be replaced when any r-class command is run after the estimation command.

## Acknowledgments

mfp is an update of mfracpol by Royston and Ambler (1998).

### References

Ambler, G., and P. Royston. 2001. Fractional polynomial model selection procedures: Investigation of Type I error rate. Journal of Statistical Computation and Simulation 69: 89–108. https://doi.org/10.1080/00949650108812083.

Royston, P., and D. G. Altman. 1994. Regression using fractional polynomials of continuous covariates: Parsimonious parametric modelling. *Journal of the Royal Statistical Society*, C ser., 43: 429–467. https://doi.org/10.2307/2986270.

Royston, P., and G. Ambler. 1998. sg81: Multivariable fractional polynomials. *Stata Technical Bulletin* 43: 24–32. Reprinted in *Stata Technical Bulletin Reprints*, vol. 8, pp. 123–132. College Station, TX: Stata Press.

Royston, P., and W. Sauerbrei. 2007. Multivariable modeling with cubic regression splines: A principled approach. Stata Journal 7: 45–70.

——. 2008. Multivariable Model-Building: A Pragmatic Approach to Regression Analysis Based on Fractional Polynomials for Modelling Continuous Variables. Chichester, UK: Wiley.

——. 2009a. Two techniques for investigating interactions between treatment and continuous covariates in clinical trials. *Stata Journal* 9: 230–251.

. 2009b. Bootstrap assessment of the stability of multivariable models. Stata Journal 9: 547–570.

——. 2016. mfpa: Extension of mfp using the ACD covariate transformation for enhanced parametric multivariable modeling. *Stata Journal* 16: 72–87.

Sauerbrei, W., and P. Royston. 1999. Building multivariable prognostic and diagnostic models: Transformation of the predictors by using fractional polynomials. *Journal of the Royal Statistical Society*, A ser., 162: 71–94. https://doi.org/ 10.1111/1467-985X.00122.

— 2002. Corrigendum: Building multivariable prognostic and diagnostic models: Transformation of the predictors by using fractional polynomials. *Journal of the Royal Statistical Society*, A ser., 165: 399–400. https://doi.org/10.1111/1467-985X.02026.

## Also see

- [R] **mfp postestimation** Postestimation tools for mfp
- [R] **fp** Fractional polynomial regression
- [U] 20 Estimation and postestimation commands

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on citing Stata documentation.