fracreg postestimation — Postestimation tools for fracreg

Postestimation commands predict margins Remarks and examples Also see

Postestimation commands

The following standard postestimation commands are available after `fracreg`:

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)
`estat (svy)`	postestimation statistics for survey data
`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
`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`	conditional means, linear predictions, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

*`forecast` and `hausman` are not appropriate with `svy` estimation results. `forecast` is also not appropriate with `mi` estimation results.

predict

Description for predict

`predict` creates a new variable containing predictions such as conditional means, linear predictions, standard errors, and equation-level scores.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic* `nooffset`]

`predict` [*type*] *stub*`*` [*if*] [*in*], `scores`

statistic	Description
Main	
`cm`	conditional mean; the default
`xb`	linear prediction
`sigma`	standard deviation of the error term (for `het()`)
`stdp`	standard error of the linear prediction

Options for predict

Main

`cm`, the default, calculates the conditional mean of the outcome.

`xb` calculates the linear prediction.

`sigma` calculates the standard deviation of the error term. It is available only when `het()` is specified.

`stdp` calculates the standard error of the linear prediction.

`nooffset` is relevant only if you specified `offset(`*varname*`)`. It modifies the calculations made by `predict` so that they ignore the offset variable; the linear prediction is treated as $\mathbf{x}_j\mathbf{b}$ rather than as $\mathbf{x}_j\mathbf{b} + \text{offset}_j$.

`scores` calculates the equation-level scores. In the case of `fracreg probit` and `fracreg logit`, $\partial \ln L/\partial(\mathbf{x}_j\boldsymbol{\beta})$ is calculated, and if the option `het()` was specified with `fracreg probit`, then $\partial \ln L/\partial(\mathbf{z}_j\boldsymbol{\gamma})$ is also calculated.

margins

Description for margins

`margins` estimates margins of response for conditional means and linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

statistic	Description
`cm`	conditional mean; the default
`xb`	linear prediction
`sigma`	standard deviation of the error term (for `het()`)
`stdp`	not allowed with `margins`
`scores`	not allowed with `margins`

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

Remarks and examples

Remarks are presented under the following headings:

Obtaining predicted values
Performing hypothesis tests

Obtaining predicted values

Once you have fit a model using `fracreg`, you can obtain the conditional mean of the fractional response by using the `predict` command for both the estimation sample and other samples; see **[U] 20 Estimation and postestimation commands** and [R] **predict**.

When you use the fractional probit estimator, `fracreg probit`, with the option `het()`, there is an additional statistic available, `sigma`. With the `sigma` option, `predict` calculates the predicted standard deviation, $\sigma_j = \exp(\mathbf{z}_j\boldsymbol{\gamma})$.

Performing hypothesis tests

▷ Example 1: Conditional means

In example 1 of [R] **fracreg**, we fit a fractional probit model to see how participation rate (`prate`) in 401(k) plans is affected by the matching rate of employer contributions (`mrate`). To obtain the predicted conditional means, we use `predict` and do not specify the default `cm` option.

```
. use https://www.stata-press.com/data/r19/401k
(Firm-level data on 401k participation)
. fracreg probit prate mrate c.ltotemp##c.ltotemp c.age##c.age i.sole
Iteration 0:  Log pseudolikelihood = -1769.6832
Iteration 1:  Log pseudolikelihood = -1675.2763
Iteration 2:  Log pseudolikelihood = -1674.6234
Iteration 3:  Log pseudolikelihood = -1674.6232
Iteration 4:  Log pseudolikelihood = -1674.6232
Fractional probit regression                    Number of obs =  4,075
                                                Wald chi2(6)  = 815.88
                                                Prob > chi2   = 0.0000
Log pseudolikelihood = -1674.6232               Pseudo R2     = 0.0632
```

prate	Coefficient	Robust std. err.	z	P>|z|	[95% conf.	interval]
mrate	.5859715	.0387616	15.12	0.000	.5100002	.6619429
ltotemp	-.6102767	.0615052	-9.92	0.000	-.7308246	-.4897288
c.ltotemp# c.ltotemp	.0313576	.003975	7.89	0.000	.0235667	.0391484
age	.0273266	.0031926	8.56	0.000	.0210691	.033584
c.age#c.age	-.0003159	.0000875	-3.61	0.000	-.0004874	-.0001443
sole						
Only plan	.0683196	.0272091	2.51	0.012	.0149908	.1216484
_cons	3.25991	.2323929	14.03	0.000	2.804429	3.715392

```
. predict mpart
(option cm assumed)
```

We can then summarize these conditional mean estimates (`cmean`) over the population to get the population average conditional mean participation rate in our sample.

```
. summarize mpart
```

Variable	Obs	Mean	Std. dev.	Min	Max
mpart	4,075	.8405767	.0828094	.6251739	.9964518

The average of the conditional mean of participation rate in our sample is 84% with a range between 62.5% and 99.6%.

◁

Example 2: Average marginal effects

In example 2 of [R] **fracreg**, we used the outcome variable and covariates of example 1 but instead of fitting a fractional probit regression, we fit a fractional logit. Using `margins`, we explore the average marginal effect of `mrate` on `prate` for both specifications.

Below, we use `margins` after `fracreg logit` with the option `post` to post the average marginal effects as estimates. We then store our results with the name `logit`. We do the same with our probit estimates.

```
. use https://www.stata-press.com/data/r19/401k, clear
(Firm-level data on 401k participation)
. fracreg logit prate mrate c.ltotemp##c.ltotemp c.age##c.age i.sole, or
Iteration 0:  Log pseudolikelihood = -1983.8372
Iteration 1:  Log pseudolikelihood = -1682.4496
Iteration 2:  Log pseudolikelihood = -1673.6458
Iteration 3:  Log pseudolikelihood = -1673.5566
Iteration 4:  Log pseudolikelihood = -1673.5566
Fractional logistic regression                  Number of obs =  4,075
                                                Wald chi2(6)  = 817.73
                                                Prob > chi2   = 0.0000
Log pseudolikelihood = -1673.5566               Pseudo R2     = 0.0638
```

prate	Odds ratio	Robust std. err.	z	P>|z|	[95% conf.	interval]
mrate	3.137781	.2345429	15.30	0.000	2.710173	3.632857
ltotemp	.3317826	.0375136	-9.76	0.000	.2658343	.4140913
c.ltotemp# c.ltotemp	1.058209	.0077125	7.76	0.000	1.043201	1.073434
age	1.052601	.0062524	8.63	0.000	1.040418	1.064927
c.age#c.age	.9994111	.0001644	-3.58	0.000	.999089	.9997333
sole						
Only plan	1.12047	.0568932	2.24	0.025	1.01433	1.237716
_cons	313.4879	134.6238	13.38	0.000	135.1083	727.3771

```
Note: _cons estimates baseline odds.
. margins, dydx(mrate) post
Average marginal effects                        Number of obs = 4,075
Model VCE: Robust
Expression: Conditional mean of prate, predict()
dy/dx wrt:  mrate
```

	dy/dx	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
mrate	.1450106	.0094558	15.34	0.000	.1264776	.1635436

```
. estimates store logit
```

The marginal effects from `fracreg logit` suggest that a small change in the matching rate of employers can increase participation by more than 14%.

```
. quietly fracreg probit prate mrate c.ltotemp##c.ltotemp c.age##c.age i.sole
. margins, dydx(mrate) post
Average marginal effects                                  Number of obs = 4,075
Model VCE: Robust
Expression: Conditional mean of prate, predict()
dy/dx wrt:  mrate
```

	dy/dx	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
mrate	.1335505	.0087385	15.28	0.000	.1164233	.1506776

```
. estimates store probit
```

For the probit model, a change in the matching rate increases participation by more than 13%.

Because we stored our margins results as estimates, we can now produce a table showing both the logit and probit results.

```
. estimates table logit probit, se
```

Variable	logit	probit
mrate	.14501059	.13355046
	.00945578	.00873852

Legend: b/se

As indicated by the standard errors in the table, both average marginal effects are significant. The difference between the two estimates is approximately one percentage point. This relatively small difference is consistent with the intuition that marginal effects obtained from probit and logit conditional means give us analogous results.

▷ Example 3: Average marginal effects for different levels of participation

We can also use `margins` to find the expected participation rate for various levels of employer matching. Using our probit model, we obtain the following by typing

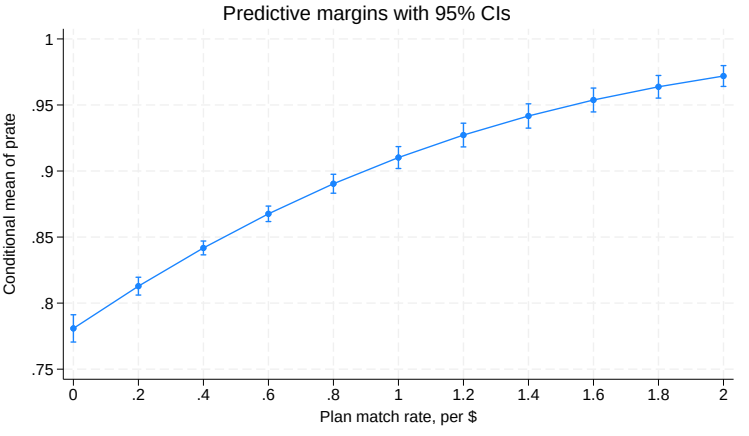
```
. quietly fracreg probit prate mrate c.ltotemp##c.ltotemp c.age##c.age i.sole
. margins, at(mrate=(0(.2)2))
Predictive margins                                      Number of obs = 4,075
Model VCE: Robust

Expression: Conditional mean of prate, predict()
1._at:  mrate =   0
2._at:  mrate =  .2
3._at:  mrate =  .4
4._at:  mrate =  .6
5._at:  mrate =  .8
6._at:  mrate =   1
7._at:  mrate = 1.2
8._at:  mrate = 1.4
9._at:  mrate = 1.6
10._at: mrate = 1.8
11._at: mrate =   2
```

	Margin	Delta-method std. err.	z	P>|z|	[95% conf.	interval]
_at						
1	.780858	.0052738	148.06	0.000	.7705216	.7911944
2	.8128364	.003441	236.22	0.000	.8060923	.8195806
3	.8417642	.002672	315.03	0.000	.8365271	.8470013
4	.8675979	.0029882	290.34	0.000	.8617412	.8734547
5	.8903734	.0036591	243.33	0.000	.8832018	.8975451
6	.9101957	.0042293	215.21	0.000	.9019065	.9184849
7	.9272265	.0045767	202.60	0.000	.9182563	.9361966
8	.9416712	.004694	200.61	0.000	.9324711	.9508712
9	.9537652	.0046115	206.82	0.000	.9447268	.9628035
10	.9637608	.004372	220.44	0.000	.9551919	.9723298
11	.9719159	.0040207	241.73	0.000	.9640355	.9797964

Going from no matching to equal matching changes the participation rate from 78% to 91%, and double matching moves participation all the way to 97.2%.

We can also see these results in a graph by using `marginsplot`.

```
. marginsplot
```



Also see

[R] **fracreg** — Fractional response regression

[U] 20 Estimation and postestimation commands