fracreg — Fractional response regression

Description

fracreg fits a fractional response model for a dependent variable that is greater than or equal to 0 and less than or equal to 1. It uses a probit, logit, or heteroskedastic probit model for the conditional mean. These models are often used for outcomes such as rates, proportions, and fractional data.

Quick start

Fractional probit model for y with values between 0 and 1 on continuous variable x1
fracreg probit y x1

As above, but use logit distribution
fracreg logit y x1

Fractional probit model for y on x1 and use x2 to model the variance of y
fracreg probit y x1, het(x2)

Menu

Statistics > Fractional outcomes > Fractional regression
Syntax

Syntax for fractional probit regression

```
fracreg probit depvar [indepvars] [if] [in] [weight] [, options]
```

Syntax for fractional logistic regression

```
fracreg logit depvar [indepvars] [if] [in] [weight] [, options]
```

Syntax for fractional heteroskedastic probit regression

```
fracreg probit depvar [indepvars] [if] [in] [weight],
het(varlist [, offset(varname)]) [options]
```

**options**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>suppress constant term</td>
</tr>
<tr>
<td>include varname in model with coefficient constrained to 1</td>
</tr>
<tr>
<td>apply specified linear constraints</td>
</tr>
<tr>
<td>independent variables to model the variance and optional offset variable with fracreg probit</td>
</tr>
</tbody>
</table>

SE/Robust

```
vce(vcetype)
```

**vcetype** may be robust, cluster clustvar, bootstrap, or jackknife

Reporting

```
level(#) or nocnsreport
display_options
```

set confidence level; default is level(95)
do not display constraints
control columns and column formats, row spacing, line width,
display of omitted variables and base and empty cells, and factor-variable labeling

Maximization

```
maximize_options
nocoef
collinear
coefflegend
```

control the maximization process; seldom used
do not display the coefficient table; seldom used
keep collinear variables
display legend instead of statistics

het() may be used only with fracreg probit to compute fractional heteroskedastic probit regression. indepvars may contain factor variables; see [U] 11.4.3 Factor variables. depvar and indepvars may contain time-series operators; see [U] 11.4.4 Time-series varlists. bayes, bootstrap, by, fp, jackknife, mi estimate, rolling, statsby, and svy are allowed; see [U] 11.1.10 Prefix commands. For more details, see [BAYES] bayes: fracreg.

vce(bootstrap) and vce(jackknife) are not allowed with the mi estimate prefix; see [MI] mi estimate.

Weights are not allowed with the bootstrap prefix; see [R] bootstrap. vce(), nocoef, and weights are not allowed with the svy prefix; see [SVY] svy.
fweights, iweights, and pweights are allowed; see [U] 11.1.6 weight.
nocnsreport; see [R] Estimation options.
indepvars, collinear, and coeflegend do not appear in the dialog box.

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

Options

Model

noconstant, offset(varname), constraints(constraints); see [R] Estimation options.

het(varlist[ , offset(varname_o) ]) specifies the independent variables and, optionally, the offset variable in the variance function. het() may only be used with fracreg probit to compute fractional heteroskedastic probit regression.

offset(varname_o) specifies that selection offset varname_o be included in the model with the coefficient constrained to be 1.

SE/Robust

vce(vcetype) specifies the type of standard error reported, which includes types that are robust to some kinds of misspecification (robust), that allow for intragroup correlation (cluster clustvar), and that use bootstrap or jackknife methods (bootstrap, jackknife); see [R] vce_option.

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

or reports the estimated coefficients transformed to odds ratios, that is, \( e^b \) rather than \( b \). Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. or may be specified at estimation or when replaying previously estimated results. This option may only be used with fracreg logit.

nocnsreport; see [R] Estimation options.

display_options: noci, nopvalues, noomitted, vsquish, noemptycells, baselevels, allbaselevels, nofvlabel, fvwrap(#), fvwrapon(style), cformat(%fmt), pformat(%fmt), sformat(%fmt), and nolstretch; see [R] Estimation options.

Maximization

maximize_options: difficult, technique(algorithm_spec), iterate(#), [no]log, trace, gradient, showstep, hessian, showtolerance, tolerance(#), ltolerance(#), nrtolerance(#), nonrtolerance, and from(init_specs); see [R] Maximize. These options are seldom used.
The following options are available with `fracreg` but are not shown in the dialog box:

- `nocoef` specifies that the coefficient table not be displayed. This option is sometimes used by programmers but is of no use interactively.

- `collinear`, `coeflegend`; see [R] Estimation options.

Remarks and examples

Fractional response data may occur when the outcome of interest is measured as a fraction, for example, a patient’s oxygen saturation or Gini coefficient values. These data are also often observed when proportions are generated from aggregated binary outcomes. For example, rather than having data on whether individual students passed an exam, we might simply have data on the proportion of students in each school that passed.

These models are appropriate when you have a dependent variable that takes values between 0 and 1 and may also be equal to 0 or 1, denoted for conciseness with the notation $[0, 1]$. If the dependent variable takes only values between 0 and 1, `betareg` might be a valid alternative. `betareg` provides more flexibility in the distribution of the mean of the dependent variable but is misspecified if the dependent variable is equal to 0 or 1. See [R] `betareg` for more information.

These models have been applied to various topics. For example, Papke and Wooldridge (1996) studied the participation rates of employees in firms’ 401(k) retirement plans. Papke and Wooldridge (2008) also evaluated an education policy by studying the pass rates for an exam administered to fourth grade Michigan students over time.

The models fit by `fracreg` are quasilikelihood estimators like the generalized linear models described in [R] `glm`. Fractional regression is a model of the mean of the dependent variable $y$ conditional on covariates $x$, which we denote by $\mu_x$. Because $y$ is in $[0, 1]$, we must ensure that $\mu_x$ is also in $[0, 1]$. We do this by using a probit, logit, or heteroskedastic probit model for $\mu_x$.

The key insight from quasilikelihood estimation is that you do not need to know the true distribution of the entire model to obtain consistent parameter estimates. In fact, the only information that you need is the correct specification of the conditional mean.

This means that the true model does not need to be, for example, a probit. If the true model is a probit, then fitting a probit regression via maximum likelihood gives you consistent parameter estimates and asymptotically efficient standard errors.

By contrast, if the conditional mean of the model is the same as the conditional mean of a probit but the model is not a probit, the point estimates are consistent, but the standard errors are not asymptotically efficient. The standard errors are not efficient, because no assumptions about the distribution of the unobserved components in the model are made. Thus `fracreg` uses robust standard errors by default.

For further discussion on quasilikelihood estimation in the context of fractional regression, please see Papke and Wooldridge (1996) and Wooldridge (2010).
Example 1: Fractional probit model of rates

In this example, we look at the expected participation rate in 401(k) plans for a cross-section of firms. Participation rate (prate) is defined as the fraction of eligible employees in a firm that participate in a 401(k) plan. We use `summarize` to see the range of the participation rate.

```
. use https://www.stata-press.com/data/r16/401k
(Firm-level data on 401k participation)
. summarize prate

Variable | Obs  | Mean   | Std. Dev. | Min   | Max
---------|------|--------|-----------|-------|-----
prate    | 4,075| .840607| .1874841  | .0036364| 1   
```

The variable has values between 0 and 1 but also has at least 1 firm for which the participation rate is exactly 1.

As in Papke and Wooldridge (1996), we surmise that the expected participation rate depends on the matching rate of employee 401(k) contributions (mrate), the natural log of the total number of employees (ltotemp), the age of the plan (age), and whether the 401(k) plan is the only retirement plan offered by the employer (sole). We include ltotemp and age, along with their squares, using factor-variable notation; see [U] 11.4.3 Factor variables.

If we believe that the functional form of the expected participation rate is a cumulative normal density, we may use `fracreg probit`.

```
. fracreg probit prate mrate c.ltotemp##c.ltotemp c.age##c.age i.sole
```

Fractional probit regression

```
Fractional probit regression

<table>
<thead>
<tr>
<th>Prate</th>
<th>Robust</th>
<th>[95% Conf. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Coef.</td>
<td>Std. Err.</td>
</tr>
<tr>
<td>mrate</td>
<td>.5859715</td>
<td>.0387616</td>
</tr>
<tr>
<td>ltotemp</td>
<td>-.6102767</td>
<td>.0615052</td>
</tr>
<tr>
<td>c.ltotemp#</td>
<td>.0313576</td>
<td>.003975</td>
</tr>
<tr>
<td>c.ltotemp#</td>
<td>.0313576</td>
<td>.003975</td>
</tr>
<tr>
<td>c.age#c.age</td>
<td>-.0003159</td>
<td>.0000875</td>
</tr>
<tr>
<td>sole</td>
<td>.0683196</td>
<td>.0272091</td>
</tr>
<tr>
<td>_cons</td>
<td>3.25991</td>
<td>.2323929</td>
</tr>
</tbody>
</table>
```

Like those obtained from `probit`, the parameters provide the sign of the marginal effect of the covariates on the outcome, but the magnitude is difficult to interpret. We can use `margins` to estimate conditional or population-averaged effects; see example 2. The standard errors are robust by default because the true data-generating process need not be a probit, even though we use the probit likelihood to obtain our parameter estimates.
Example 2: Changing the distribution of the conditional mean

Continuing with example 1, we may instead believe that the expected participation rate follows a fractional logistic response. In this case, we should use fractional logistic regression instead of fractional probit regression to obtain consistent estimates of the parameters of the conditional mean.

```
.fracreg logit prate mrate c.ltotemp##c.ltotemp c.age##c.age i.sole
```

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

|                 | Coef. | Std. Err. | z    | P>|z|   | [95% Conf. Interval] |
|-----------------|-------|-----------|------|-------|---------------------|
| prate           | 1.143516 | .074748 | 15.30 | 0.000 | .9970125 - 1.290019 |
| mrate           | -1.103275 | .1130667 | -9.76 | 0.000 | -1.324882 - .8816687 |
| ltotemp         | .0565782 | .0072883 | 7.76  | 0.000 | .0422934 - .070863  |
| c.ltotemp#      | .0512643 | .0059399 | 8.63  | 0.000 | .0396223 - .0629064 |
| c.age#c.age     | -.0005891 | .0001645 | -3.58 | 0.000 | -.0009114 - -.0002667 |
| sole only plan  | .1137479 | .0507762 | 2.24  | 0.025 | .0142284 - .2132674 |
| _cons           | 5.747761 | .4294386 | 13.38 | 0.000 | 4.906077 - 6.589445 |

Like those obtained from `logit`, the parameters provide the sign of the marginal effect of the covariates on the outcome, but the magnitude is again difficult to interpret. As with `fracreg probit` in example 1, we would use `margins` to obtain the marginal effects or other predictions of interest.

Example 3: Odds ratios from a fractional logit model

When the conditional mean of our outcome is interpretable as a probability, it is possible to adopt an odds-ratio interpretation of the results of a fractional logit model. In example 2, this is plausible because expected participation rates can be viewed as estimates of the probability of participation. We obtain the odds ratios by specifying the option `or`.
. 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#     | 1.058209   | .0077125         | 7.76 | 0.000| 1.043201, 1.073434|
| c.ltotemp#     | 1.0052601  | .0062524         | 8.63 | 0.000| 1.040418, 1.064927|
| 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.

Among other things, we see that if the 401(k) is the only plan offered by the employer, then the odds of an employee participating increase by a factor of 1.12. We can also see that if the matching rate goes from 0 to 1:1 (exactly matching employee contributions) or from 1:1 to 2:1 (doubling employee contributions), then the odds of participating increase by 3.1.

The use of an odds-ratio interpretation is not appropriate if the conditional mean cannot be viewed as a probability. For example, if the fractional outcome were a Gini coefficient, we could not interpret the expected values of our outcomes as probabilities. The Gini coefficient is a measure of inequality between zero and one and cannot be interpreted as a probability. In this case, using the odds-ratio option would not be sensible.
Stored results

fracreg stores the following in e():

Scalars
- e(N) number of observations
- e(k) number of parameters
- e(k_eq) number of equations in e(b)
- e(k_eq_model) number of equations in overall model test
- e(k_dv) number of dependent variables
- e(df_m) model degrees of freedom
- e(r2_p) pseudo-$R^2$-squared
- e(ll) log likelihood
- e(ll_0) log likelihood, constant-only model
- e(N_clust) number of clusters
- e(chi2) $\chi^2$
- e(p) p-value for model test
- e(rank) rank of e(V)
- e(ic) number of iterations
- e(rc) return code
- e(converged) 1 if converged, 0 otherwise

Macros
- e(cmd) fracreg
- e(cmdline) command as typed
- e(estimator) model for conditional mean; logit, probit, or hetprobit
- e(depvar) name of dependent variable
- e(wtype) weight type
- e(wexp) weight expression
- e(title) title in estimation output
- e(clustvar) name of cluster variable
- e(offset) offset
- e(chi2type) Wald; type of model $\chi^2$ test
- e(vce) vcetype specified in vce()
- e(vcetype) title used to label Std. Err.
- e(opt) type of optimization
- e(which) max or min; whether optimizer is to perform maximization or minimization
- e(ml_method) type of ml method
- e(user) name of likelihood-evaluator program
- e(technique) maximization technique
- e(properties) b V
- e(estat_cmd) program used to implement estat
- e(predict) program used to implement predict
- e(marginsnotok) predictions disallowed by margins
- e(asbalanced) factor variables fvset as asbalanced
- e(asobserved) factor variables fvset as asobserved

Matrices
- e(b) coefficient vector
- e(mns) vector of means of the independent variables
- e(Cns) constraints matrix
- e(lllog) iteration log (up to 20 iterations)
- e(gradient) gradient vector
- e(V) variance–covariance matrix of the estimators
- e(V_modelbased) model-based variance

Functions
- e(sample) marks estimation sample
Methods and formulas

The log-likelihood function for fractional models is of the form

\[
\ln L = \sum_{j=1}^{N} w_j y_j \ln \left\{ G \left( x_j' \beta \right) \right\} + w_j (1 - y_j) \ln \left\{ 1 - G \left( x_j' \beta \right) \right\}
\]

where \( N \) is the sample size, \( y_j \) is the dependent variable, \( w_j \) denotes the optional weights, \( \ln L \) is maximized, as described in [R]\ Maximize, and \( G(\cdot) \) can be

<table>
<thead>
<tr>
<th>Model</th>
<th>Functional form for ( G \left( x_j' \beta \right) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>probit</td>
<td>( \Phi \left( x_j' \beta \right) )</td>
</tr>
<tr>
<td>logit</td>
<td>( \exp(x_j' \beta)/(1 + \exp(x_j' \beta)) )</td>
</tr>
<tr>
<td>hetprobit</td>
<td>( \Phi \left{ x_j' \beta / \exp(z_j' \gamma) \right} )</td>
</tr>
</tbody>
</table>

where \( x_j \) are the covariates for individual \( j \), \( z_j \) are the covariates used to model the variance of the outcome for the heteroskedastic probit model, and \( \Phi \) is the standard normal cumulative density function.

References


Also see

[R]\ fracreg postestimation — Postestimation tools for fracreg

[R]\ betareg — Beta regression

[R]\ glm — Generalized linear models

[BAYES] bayes: fracreg — Bayesian fractional response regression

[MJ]\ Estimation — Estimation commands for use with mi estimate

[SVY]\ svy estimation — Estimation commands for survey data

[U]\ 20 Estimation and postestimation commands
## Postestimation commands

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

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>contrast</td>
<td>contrasts and ANOVA-style joint tests of estimates</td>
</tr>
<tr>
<td>estat ic</td>
<td>Akaike’s and Schwarz’s Bayesian information criteria (AIC and BIC)</td>
</tr>
<tr>
<td>estat summarize</td>
<td>summary statistics for the estimation sample</td>
</tr>
<tr>
<td>estat vce</td>
<td>variance–covariance matrix of the estimators (VCE)</td>
</tr>
<tr>
<td>estat (svy)</td>
<td>postestimation statistics for survey data</td>
</tr>
<tr>
<td>estimates</td>
<td>cataloging estimation results</td>
</tr>
<tr>
<td><em>forecast</em></td>
<td>dynamic forecasts and simulations</td>
</tr>
<tr>
<td><em>hausman</em></td>
<td>Hausman’s specification test</td>
</tr>
<tr>
<td>lincom</td>
<td>point estimates, standard errors, testing, and inference for linear combinations</td>
</tr>
<tr>
<td>margins</td>
<td>marginal means, predictive margins, marginal effects, and average marginal effects</td>
</tr>
<tr>
<td>marginsplot</td>
<td>graph the results from margins (profile plots, interaction plots, etc.)</td>
</tr>
<tr>
<td>nlcom</td>
<td>point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients</td>
</tr>
<tr>
<td>predict</td>
<td>predictions, residuals, influence statistics, and other diagnostic measures</td>
</tr>
<tr>
<td>predictnl</td>
<td>point estimates, standard errors, testing, and inference for generalized predictions</td>
</tr>
<tr>
<td>pwcompare</td>
<td>pairwise comparisons of estimates</td>
</tr>
<tr>
<td>test</td>
<td>Wald tests of simple and composite linear hypotheses</td>
</tr>
<tr>
<td>testnl</td>
<td>Wald tests of nonlinear hypotheses</td>
</tr>
</tbody>
</table>

* 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

\[ \text{predict [type] newvar [if] [in], statistic nooffset] } \]

\[ \text{predict [type] { stub*|newvar|newvarlist} [if] [in], scores} \]

<table>
<thead>
<tr>
<th>statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>conditional mean; the default</td>
</tr>
<tr>
<td>xb</td>
<td>linear prediction</td>
</tr>
<tr>
<td>sigma</td>
<td>standard deviation of the error term (for het())</td>
</tr>
<tr>
<td>stdp</td>
<td>standard error of the linear prediction</td>
</tr>
<tr>
<td>scores</td>
<td>equation-level score variables</td>
</tr>
</tbody>
</table>

Options for predict

- 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.
- scores calculates the equation-level score, \( \partial \ln L / \partial (x_j\beta) \), in the case of fracreg probit and fracreg logit, and can also calculate \( \partial \ln L / \partial (z_j\gamma) \) if the option het() is specified.
- 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 \( x_jb \) rather than as \( x_jb + offset_j \).
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]

<table>
<thead>
<tr>
<th>statistic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cm</td>
<td>conditional mean; the default</td>
</tr>
<tr>
<td>xb</td>
<td>linear prediction</td>
</tr>
<tr>
<td>sigma</td>
<td>standard deviation of the error term (for het())</td>
</tr>
<tr>
<td>stdp</td>
<td>not allowed with margins</td>
</tr>
<tr>
<td>scores</td>
<td>not allowed with margins</td>
</tr>
</tbody>
</table>

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, $\sigma_j = \exp(z_j\gamma)$. 
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 \texttt{predict} and do not specify the default \texttt{cm} option.

\[ \begin{align*}
\text{use} \: \texttt{https://www.stata-press.com/data/r16/401k} \\
\text{(Firm-level data on 401k participation)} \\
\text{fracreg} \: \texttt{probit} \: \texttt{prate} \: \texttt{mrate} \: \texttt{c.ltotemp##c.ltotemp} \: \texttt{c.age##c.age} \: \texttt{i.sole} \\
\text{Iteration 0:} \: \text{log pseudolikelihood} = -1769.6832 \\
\text{Iteration 1:} \: \text{log pseudolikelihood} = -1675.2763 \\
\text{Iteration 2:} \: \text{log pseudolikelihood} = -1674.6234 \\
\text{Iteration 3:} \: \text{log pseudolikelihood} = -1674.6232 \\
\text{Iteration 4:} \: \text{log pseudolikelihood} = -1674.6232 \\
\text{Fractional probit regression} \: \text{Number of obs} = 4,075 \\
\text{Wald chi2(6)} = 815.88 \\
\text{Prob > chi2} = 0.0000 \\
\text{Log pseudolikelihood} = -1674.6232 \\
\text{Pseudo R2} = 0.0632
\end{align*} \]

\[ \begin{align*}
\text{prate} & \quad \text{Robust} \quad \text{Std. Err.} \\
\text{mrate} & \quad 0.5859715 \quad 0.0387616 \quad 15.12 \quad 0.000 \quad 0.5100002 \quad 0.6619429 \\
\text{ltotemp} & \quad -0.6102767 \quad 0.0615052 \quad -9.92 \quad 0.000 \quad -0.7308246 \quad -0.4897288 \\
\text{c.ltotemp##c.ltotemp} & \quad 0.0313576 \quad 0.003975 \quad 7.89 \quad 0.000 \quad 0.0235667 \quad 0.0391484 \\
\text{age} & \quad 0.0273266 \quad 0.0031926 \quad 8.56 \quad 0.000 \quad 0.0210691 \quad 0.033584 \\
\text{c.age##c.age} & \quad -0.0003159 \quad 0.0000875 \quad -3.61 \quad 0.000 \quad -0.0004874 \quad -0.0001443 \\
\text{sole only plan} & \quad 0.0683196 \quad 0.0272091 \quad 2.51 \quad 0.012 \quad 0.0149908 \quad 0.1216484 \\
\text{cons} & \quad 3.25991 \quad 0.232929 \quad 14.03 \quad 0.000 \quad 2.804429 \quad 3.715392 \quad 4,075 \\
\end{align*} \]

\[ \begin{align*}
\text{predict} \: \texttt{mpart} \\
\text{(option \texttt{cm} assumed)} \\
\text{summarize mpart} \\
\text{Variable} \quad \text{Obs} \quad \text{Mean} \quad \text{Std. Dev.} \quad \text{Min} \quad \text{Max} \\
\text{mpart} \quad 4,075 \quad 0.8405767 \quad 0.0828094 \quad 0.6251739 \quad 0.9964518
\end{align*} \]

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

\[ \begin{align*}
\text{summarize mpart} \\
\text{Variable} \quad \text{Obs} \quad \text{Mean} \quad \text{Std. Dev.} \quad \text{Min} \quad \text{Max} \\
\text{mpart} \quad 4,075 \quad 0.8405767 \quad 0.0828094 \quad 0.6251739 \quad 0.9964518
\end{align*} \]

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/r16/401k, 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

|                  | Odds Ratio  | Std. Err. |  z   | P>|z|  | [95% Conf. Interval] |
|------------------|-------------|-----------|------|------|----------------------|
| prate            |             |           |      |      |                      |
| 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 w.r.t. : mrate

| Delta-method  | dy/dx | Std. Err. |  z   | P>|z|  | [95% Conf. Interval] |
|---------------|-------|-----------|------|------|----------------------|
| mrable        | .1450106 | .0094558  | 15.34| 0.000| .1264776             | .1635436     |
```

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 w.r.t. : mrate

| dy/dx  | 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

<table>
<thead>
<tr>
<th>Variable</th>
<th>logit</th>
<th>probit</th>
</tr>
</thead>
<tbody>
<tr>
<td>mrate</td>
<td>.14501059</td>
<td>.13355046</td>
</tr>
<tr>
<td></td>
<td>.00945578</td>
<td>.00873852</td>
</tr>
</tbody>
</table>

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))
```

| _at | Margin | Std. Err. | z | P>|z| | [95% Conf. Interval] |
|-----|--------|-----------|---|-----|----------------------|
| 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
```

![Graph showing Predictive Margins with 95% CIs](image)

**Conditional Mean Of Prate**

- Predictive Margins with 95% CIs
- plan match rate, per $

Also see

[R] `fracreg` — Fractional response regression

[U] 20 Estimation and postestimation commands