bayes: fracreg — Bayesian fractional response regression

Description

`bayes: fracreg` fits a Bayesian fractional response regression to a fractional outcome whose values are greater than or equal to 0 and less than or equal to 1; see [BAYES] **bayes** and [R] **fracreg** for details.

Quick start

Bayesian fractional probit regression of y on x1 and x2, using default normal priors for regression coefficients

```
bayes: fracreg probit y x1 x2
```

Use a standard deviation of 10 instead of 100 for the default normal priors

```
bayes, normalprior(10): fracreg probit y x1 x2
```

Use uniform priors for the slopes and a normal prior for the intercept

```
bayes, prior({y: x1 x2}, uniform(-10,10)) ///
prior({y:_cons}, normal(0,10)): fracreg probit y x1 x2
```

Save simulation results to `simdata.dta`, and use a random-number seed for reproducibility

```
bayes, saving(simdata) rseed(123): fracreg probit y x1 x2
```

Specify 20,000 Markov chain Monte Carlo (MCMC) samples, set length of the burn-in period to 5,000, and request that a dot be displayed every 500 simulations

```
bayes, mcmcsize(20000) burnin(5000) dots(500): fracreg probit y x1 x2
```

In the above, request that the 90% highest posterior density (HPD) credible interval be displayed instead of the default 95% equal-tailed credible interval

```
bayes, clevel(90) hpd
```

Fit a fractional logistic regression and display results as odds ratios

```
bayes: fracreg logit y x1 x2, or
```

Display odds ratios on replay

```
bayes, or
```

Also see *Quick start* in [BAYES] **bayes** and *Quick start* in [R] **fracreg**.

Menu

Statistics > Fractional outcomes > Bayesian fractional regression

Syntax

Syntax for fractional probit regression

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

Syntax for fractional logistic regression

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

Syntax for fractional heteroskedastic probit regression

```
bayes [ , bayesopts ] : fracreg probit depvar [indepvars] [if] [in] [weight],
    het(varlist[ , offset(varname_o) ]) [options]
```

options	Description
Model	
noconstant	suppress constant term
offset(*varname*)	include *varname* in model with coefficient constrained to 1
* het(*varlist*[, offset(*varname*$_o$)])	independent variables to model the variance and possible offset variable with `fracreg probit`
Reporting	
or	report odds ratios; only valid with `fracreg logit`
display_options	control spacing, line width, and base and empty cells
level(#)	set credible level; default is `level(95)`

* `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**.

`fweights` are allowed; see **[U] 11.1.6 weight**.

`bayes: fracreg, level()` is equivalent to `bayes, clevel(): fracreg`.

For a detailed description of *options*, see *Options* in [R] **fracreg**.

bayesopts	Description
Priors	
* normalprior(#)	specify standard deviation of default normal priors for regression coefficients; default is `normalprior(100)`
prior(*priorspec*)	prior for model parameters; this option may be repeated
dryrun	show model summary without estimation

Simulation

nchains(#)	number of chains; default is to simulate one chain
mcmcsize(#)	MCMC sample size; default is mcmcsize(10000)
burnin(#)	burn-in period; default is burnin(2500)
thinning(#)	thinning interval; default is thinning(1)
rseed(#)	random-number seed
exclude(*paramref*)	specify model parameters to be excluded from the simulation results

Blocking

* blocksize(#)	maximum block size; default is blocksize(50)
block(*paramref* [, *blockopts*])	specify a block of model parameters; this option may be repeated
blocksummary	display block summary
* noblocking	do not block parameters by default

Initialization

initial(*initspec*)	specify initial values for model parameters with a single chain
init#(*initspec*)	specify initial values for #th chain; requires nchains()
initall(*initspec*)	specify initial values for all chains; requires nchains()
nomleinitial	suppress the use of maximum likelihood estimates as starting values
initrandom	specify random initial values
initsummary	display initial values used for simulation
* noisily	display output from the estimation command during initialization

Adaptation

adaptation(*adaptopts*)	control the adaptive MCMC procedure
scale(#)	initial multiplier for scale factor; default is scale(2.38)
covariance(*cov*)	initial proposal covariance; default is the identity matrix

Reporting

clevel(#)	set credible interval level; default is clevel(95)
hpd	display HPD credible intervals instead of the default equal-tailed credible intervals
* or	report odds ratio; only valid with fracreg logit
eform[(*string*)]	report exponentiated coefficients and, optionally, label as *string*
batch(#)	specify length of block for batch-means calculations; default is batch(0)
saving(*filename*[, replace])	save simulation results to *filename*.dta
nomodelsummary	suppress model summary
chainsdetail	display detailed simulation summary for each chain
[no]dots	suppress dots or display dots every 100 iterations and iteration numbers every 1,000 iterations; default is nodots
dots(#[, every(#)])	display dots as simulation is performed
[no]show(*paramref*)	specify model parameters to be excluded from or included in the output
notable	suppress estimation table
noheader	suppress output header
title(*string*)	display *string* as title above the table of parameter estimates
display_options	control spacing, line width, and base and empty cells

Advanced

`search(`*search_options*`)`	control the search for feasible initial values
`corrlag(#)`	specify maximum autocorrelation lag; default varies
`corrtol(#)`	specify autocorrelation tolerance; default is `corrtol(0.01)`

* Starred options are specific to the `bayes` prefix; other options are common between `bayes` and `bayesmh`.

Options `prior()` and `block()` may be repeated.

priorspec and *paramref* are defined in [BAYES] **bayesmh**.

paramref may contain factor variables; see **[U] 11.4.3 Factor variables**.

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

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

Model parameters are regression coefficients {*depvar*:*indepvars*} and, if option `het()` is specified, regression coefficients {`lnsigma`:*varlist*} for the log-standard-deviation equation. Use the `dryrun` option to see the definitions of model parameters prior to estimation.

For a detailed description of *bayesopts*, see *Options* in [BAYES] **bayes**.

Remarks and examples

For a general introduction to Bayesian analysis, see [BAYES] **Intro**. For a general introduction to Bayesian estimation using an adaptive Metropolis–Hastings algorithm, see [BAYES] **bayesmh**. For remarks and examples specific to the `bayes` prefix, see [BAYES] **bayes**. For details about the estimation command, see [R] **fracreg**.

For a simple example of the `bayes` prefix, see *Introductory example* in [BAYES] **bayes**.

Stored results

See *Stored results* in [BAYES] **bayes**.

Methods and formulas

See *Methods and formulas* in [BAYES] **bayesmh**.

Also see

[BAYES] **bayes** — Bayesian regression models using the bayes prefix

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

[BAYES] **Bayesian postestimation** — Postestimation tools after Bayesian estimation

[BAYES] **Bayesian estimation** — Bayesian estimation commands

[BAYES] **Bayesian commands** — Introduction to commands for Bayesian analysis

[BAYES] **Intro** — Introduction to Bayesian analysis

[BAYES] **Glossary**