

[Description](#)[Remarks and examples](#)[Quick start](#)[Stored results](#)[Menu](#)[Methods and formulas](#)[Syntax](#)[Also see](#)

Description

`bayes: probit` fits a Bayesian probit regression to a binary outcome; see [\[BAYES\] bayes](#) and [\[R\] probit](#) for details.

Quick start

Bayesian probit regression of y on x_1 and x_2 , using default normal priors for regression coefficients

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

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

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

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

```
bayes, saving(simdata) rseed(123): 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): 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
```

Also see [Quick start](#) in [\[BAYES\] bayes](#) and [Quick start](#) in [\[R\] probit](#).

Menu

Statistics > Binary outcomes > Bayesian regression > Probit regression

Syntax

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

<i>options</i>	Description
Model	
<code>noconstant</code>	suppress constant term
<code>offset(<i>varname</i>)</code>	include <i>varname</i> in model with coefficient constrained to 1
<code>asis</code>	retain perfect predictor variables
Reporting	
<code>display_options</code>	control spacing, line width, and base and empty cells
<code>level(#)</code>	set credible level; default is level(95)
<i>indepvars</i> may contain factor variables; see [U] 11.4.3 Factor variables.	
<i>depvar</i> and <i>indepvars</i> may contain time-series operators; see [U] 11.4.4 Time-series varlists.	
fweights are allowed; see [U] 11.1.6 weight.	
bayes: probit, level() is equivalent to bayes, clevel(): probit.	
For a detailed description of <i>options</i> , see <i>Options</i> in [R] probit.	
<i>bayesopts</i>	Description
Priors	
* <code>normalprior(#)</code>	specify standard deviation of default normal priors for regression coefficients; default is normalprior(100)
<code>prior(<i>priorspec</i>)</code>	prior for model parameters; this option may be repeated
<code>dryrun</code>	show model summary without estimation
Simulation	
<code>nchains(#)</code>	number of chains; default is to simulate one chain
<code>mcmcsize(#)</code>	MCMC sample size; default is mcmcsize(10000)
<code>burnin(#)</code>	burn-in period; default is burnin(2500)
<code>thinning(#)</code>	thinning interval; default is thinning(1)
<code>rseed(#)</code>	random-number seed
<code>exclude(<i>paramref</i>)</code>	specify model parameters to be excluded from the simulation results
Blocking	
* <code>blocksize(#)</code>	maximum block size; default is blocksize(50)
<code>block(<i>paramref</i> [, <i>blockopts</i>])</code>	specify a block of model parameters; this option may be repeated
<code>blocksummary</code>	display block summary
* <code>noblocking</code>	do not block parameters by default
Initialization	
<code>initial(<i>initspec</i>)</code>	specify initial values for model parameters with a single chain
<code>init#(<i>initspec</i>)</code>	specify initial values for #th chain; requires nchains()
<code>initall(<i>initspec</i>)</code>	specify initial values for all chains; requires nchains()
<code>nomleinitial</code>	suppress the use of maximum likelihood estimates as starting values
<code>initransom</code>	specify random initial values
<code>initsummary</code>	display initial values used for simulation
* <code>noisily</code>	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
`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
`corrto1(#)` specify autocorrelation tolerance; default is `corrto1(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 `{devar: indepvars}`. 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] `probit`.

For a simple example of the `bayes` prefix, see *Introductory example* in [BAYES] `bayes`. Also see *Logistic regression with perfect predictors* 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] **probit** — Probit 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**

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](#).