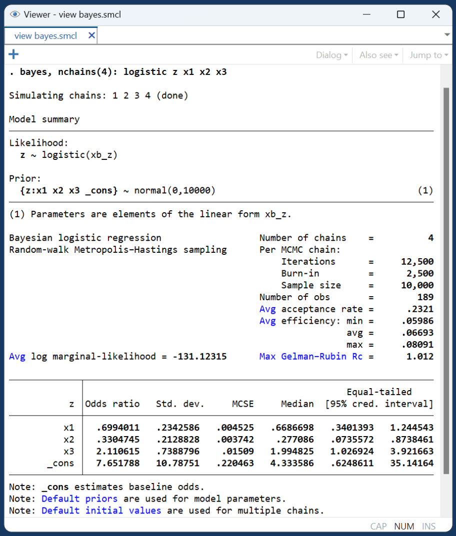
Stata Features

Bayesian analysis

Your Bayesian analysis in Stata can be as simple or as complex as your research problem.

- Thousands of built-in models
- Add your own models
- Prefix your command with **bayes:**
- Adaptive Metropolis–Hastings
- Gibbs sampling
- Multiple chains
- Convergence diagnostics
- Explore distributions
- Model goodness of fit
- Posterior predictive *p*-values
- Posterior summaries
- Hypothesis testing
- Model comparison
- Predictions
- Model averaging New
- More



Viewer - view bayes.smcl

```
. bayes, nchains(4): logistic z x1 x2 x3

Simulating chains: 1 2 3 4 (done)

Model summary
------------------------------------------------------------------------------
Likelihood:
  z ~ logistic(xb_z)

Prior:
  {z:x1 x2 x3 _cons} ~ normal(0,10000)                                      (1)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_z.

Bayesian logistic regression                     Number of chains    =       4
Random-walk Metropolis-Hastings sampling         Per MCMC chain:
                                                     Iterations      =  12,500
                                                     Burn-in         =   2,500
                                                     Sample size     =  10,000
                                                 Number of obs       =     189
                                                 Avg acceptance rate =   .2321
                                                 Avg efficiency: min =  .05986
                                                                 avg =  .06693
                                                                 max =  .08091
Avg log marginal-likelihood = -131.12315         Max Gelman-Rubin Rc =   1.012

------------------------------------------------------------------------------
             |                                                Equal-tailed
           z | Odds ratio   Std. dev.      MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
          x1 |   .6994011   .2342586    .004525   .6686698   .3401393   1.244543
          x2 |   .3304745   .2128828    .003742    .277086   .0735572   .8738461
          x3 |   2.110615   .7388796     .01509   1.994825   1.026924   3.921663
       _cons |   7.651788   10.78751    .220463   4.333586   .6248611   35.14164
------------------------------------------------------------------------------
Note: _cons estimates baseline odds.
Note: Default priors are used for model parameters.
Note: Default initial values are used for multiple chains.
```

Fit regression models

Linear regression

```
. bayes: regress y x1 x2 x3
```

Logistic regression

```
. bayes: logistic z x1 x2 x3
```

Multilevel regression

```
. bayes: mixed y x1 x2 x3 || id:
```

Vector autoregression (VAR)

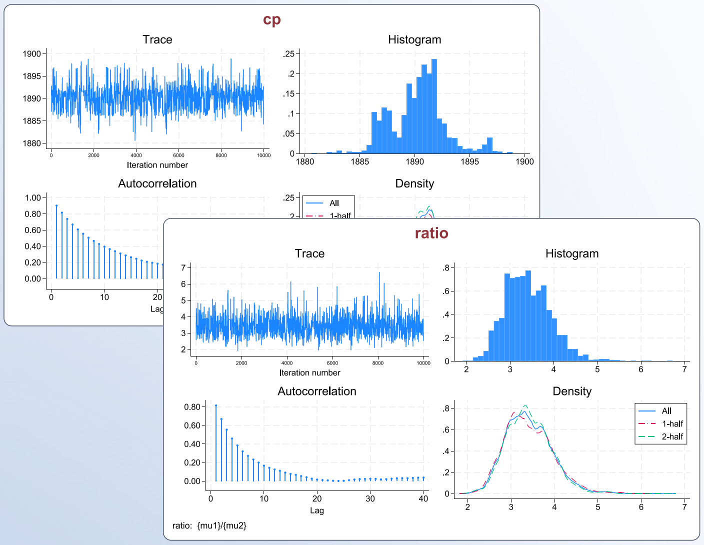
```
. bayes: var y1 y2 y3, lags(1/3) exog(x1 x2)
```

Specify multiple chains

```
. bayes, nchains(4): logistic z x1 x2 x3
```

Check convergence

```
. bayesgraph diagnostics {cp}
             (ratio: {mu1}/{mu2})
```



Fit general models

Multilevel meta-analysis model

```
. bayesmh lnOR U[trial], noconstant likelihood(normal(var))
                         prior({U[trial]}, normal({theta},{tau2}))
                         prior({theta}, normal(0,10000))
                         prior({tau2}, igamma(0.0001,0.0001))
                         block({theta tau2}, gibbs split)
```

Nonlinear Poisson model: Change-point analysis

```
. bayesmh count, likelihood(dpoisson({mu1}*sign(year<{cp})+{mu2}*sign(year>={cp})))
                 prior({mu1 mu2}, flat)
                 prior({cp}, uniform(1851,1962))
                 initial({mu1 mu2} 1 {cp} 1906)
```

Program your own models

Hurdle model

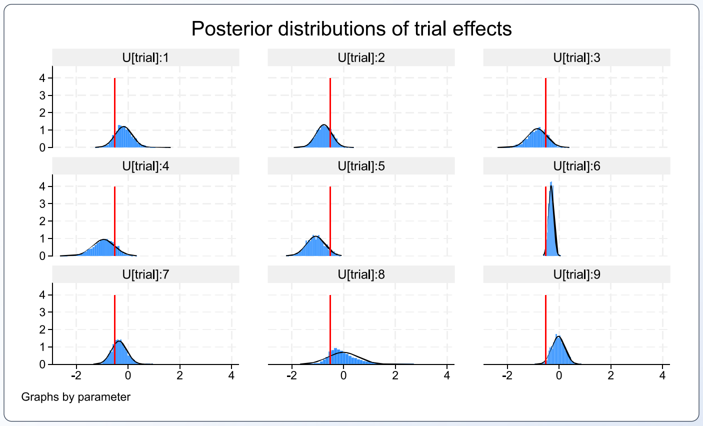
```
. bayesmh (hours age) (hours0 commute), llevaluator(mychurdle, parameters({lnsig}))
                                         prior({hours:} {hours0:} {lnsig}, flat)
```

```
program mychurdle
        version 18.0
        args lnf xb xg lnsig
        tempname sig
        scalar `sig' = exp(`lnsig')
        tempvar lnfj
        qui gen double `lnfj' = normal(`xg')
        qui replace    `lnfj' = log(1 - `lnfj') if $MH_y1 <= 0
        qui replace    `lnfj' = log(`lnfj') - log(normal(`xb'/`sig')) +
            log(normalden($MH_y1,`xb',`sig')) if $MH_y1 > 0
        summarize `lnfj', meanonly
        if r(N) < $MH_n {
            scalar `lnf' = .
            exit
        }
        scalar `lnf' = r(sum)
end
```

Perform inference

Explore distributions

```
. bayesgraph histogram {U[trial]}, ...
```



Test hypothesis

```
. bayestest interval {mu1}/{mu2}, lower(3)
```

Interval tests   MCMC sample size = 10,000

      prob1 : {mu1}/{mu2} > 3

	Mean	Std. dev.	MCSE
prob1	.7147	0.45158	.0216545

Compare models

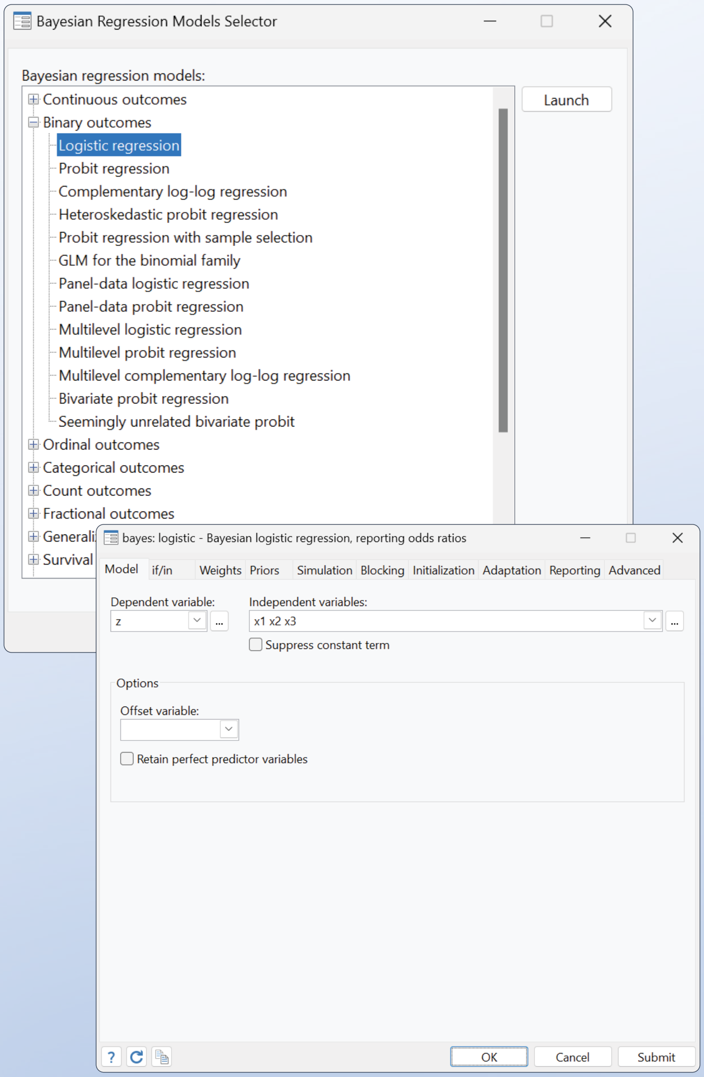
```
. bayesstats ic model1 model2
```

Bayesian information criteria

	DIC	log(ML)	log(BF)
model1	472.0359	-242.5827	.
model2	470.8157	-235.7438	6.838942

Note: Marginal likelihood (ML) is computed
      using Laplace-Metropolis approximation.

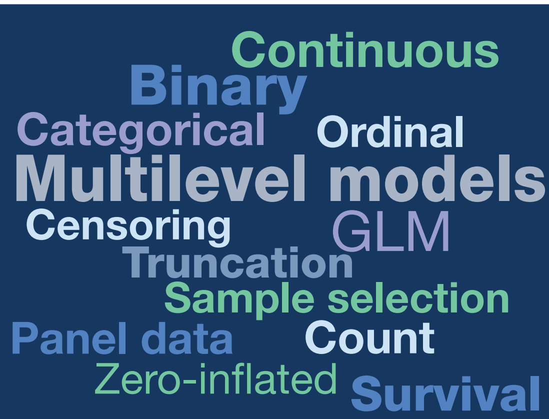
Perform any analyses using GUI



Regression models

Simply prefix your regression command with **bayes:**

- Over 60 likelihood models supported, including multilevel, survival, GLM, VAR, DSGE, and more
- Censoring, truncation, sample selection
- Intuitive and elegant model specification
- Default and custom priors
- Comprehensive Bayesian-features support



Linear regression

Use default normal priors for coefficients and inverse-gamma prior for variance

```
. bayes: regress y x1 x2
```

```
Viewer - view bayes_regress.smcl
view bayes_regress.smcl
Dialog  Also see  Jump to

. bayes: regress y x1 x2

Burn-in ...
Simulation ...

Model summary
------------------------------------------------------------------------------
Likelihood:
  y ~ regress(xb_y,{sigma2})

Priors:
  {y:x1 x2 _cons} ~ normal(0,10000)                                       (1)
         {sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_y.

Bayesian linear regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =         74
                                                 Acceptance rate  =      .3855
                                                 Efficiency:  min =      .07633
                                                              avg =      .1159
Log marginal-likelihood = -218.13296                          max =      .1694

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
y            |
          x1 | -.3951938   .1637576   .004942  -.3960287  -.7201108  -.0847018
          x2 | -.7656255   .5779397   .017594  -.7682354  -1.867728   .3611014
       _cons |  47.60132    6.36683   .230446    47.5305   35.33225     60.082
-------------+----------------------------------------------------------------
      sigma2 |  11.97461   2.055562    .04994   11.77334   8.507135     16.573
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.

CAP  NUM  INS
```

Use Gibbs sampling

```
. bayes, gibbs: regress y x1 x2
```

Logistic regression

Use default normal priors for coefficients

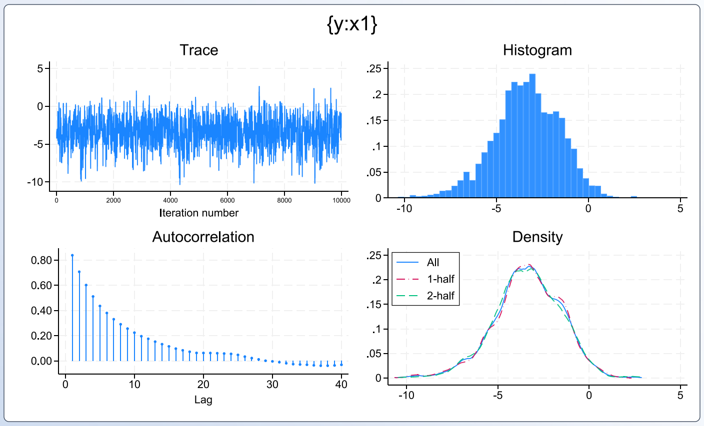
```
. bayes: logistic y x1 x2
```

Use custom Cauchy priors for coefficients on **x1** and **x2**

```
. bayes, prior({y:x1 x2}, cauchy(0,2.5)):
         logistic y x1 x2
```

Check convergence of coefficient on **x1**

```
. bayesgraph diagnostics {y:x1}
```



Generalized linear model

Use burn-in of 1,000 and MCMC size of 5,000

```
. bayes, burnin(1000) mcmcsize(5000):
         glm y x1 x2, family(binomial) link(log)
```

Test that coefficient **{y:x1}** is greater than 4

```
. bayestest interval {y:x1}, lower(4)
```

Viewer - view bayes_interval.smcl

view bayes_interval.smcl

Dialog · Also see · Jump to

```
. bayestest interval {y:x1}, lower(4)

Interval tests     MCMC sample size =     10,000

       prob1 : {y:x1} > 4

-----------------------------------------
             |      Mean    Std. dev.      MCSE
-------------+---------------------------
       prob1 |     .7881     0.40867   .0125451
-----------------------------------------
```

CAP NUM INS

Survival regression

Declare survival data

```
. stset time, failure(died)
```

Fit Bayesian exponential regression

```
. bayes, saving(mcmc_exp): streg x1 x2,
         distribution(exponential)

. estimates store exp
```

Fit Bayesian Weibull regression

```
. bayes, saving(mcmc_weibull): streg x1 x2,
         distribution(weibull)

. estimates store weibull
```

Compare models using the Bayes factor

```
. bayesstats ic exp weibull, bayesfactor
```

Viewer - view bayes_ic.smcl

view bayes_ic.smcl

Dialog · Also see · Jump to

```
. bayesstats ic exp weibull, bayesfactor

Bayesian information criteria

----------------------------------------------
             |       DIC    log(ML)        BF
-------------+--------------------------------
         exp |  103.4405  -71.04365         .
     weibull |  92.02015  -74.79336  .0235246
----------------------------------------------
Note: Marginal likelihood (ML) is computed
      using Laplace-Metropolis approximation.
```

CAP NUM INS

Compare models using posterior probabilities

```
. bayestest model exp weibull
```

Viewer - view bayes_model.smcl

view bayes_model.smcl

Dialog · Also see · Jump to

```
. bayestest model exp weibull

Bayesian model tests

----------------------------------------------
             |   log(ML)      P(M)    P(M|y)
-------------+--------------------------------
         exp |  -71.0436    0.5000    0.9770
     weibull |  -74.7934    0.5000    0.0230
----------------------------------------------
Note: Marginal likelihood (ML) is computed using
      Laplace-Metropolis approximation.
```

CAP NUM INS

Other regression models

Ordered logistic regression

```
. bayes: ologit y x1 x2
```

Conditional logistic regression

```
. bayes: clogit y x1 x2, group(id)
```

Poisson regression

```
. bayes: poisson y x1 x2
```

Truncated Poisson regression

```
. bayes: tpoisson y x1 x2, ll(10)
```

Zero-inflated negative binomial regression

```
. bayes: zinb y x1 x2, inflated(z1 z2)
```

Tobit regression

```
. bayes: tobit y x1 x2, ul(20)
```

Heteroskedastic probit regression

```
. bayes: hetprobit y x1 x2, het(xhet)
```

Heckman selection model

```
. bayes: heckman y x1 x2, select(x1 x2 x3)
```

Multivariate regression

```
. bayes: mvreg y1 y2 y3 = x1 x2
```

Multilevel regression

```
. bayes: mixed y x1 x2 || id:
```

Vector autoregression (VAR)

```
. bayes: var y1 y2 y3, lags(1/3) exog(x1 x2)
```

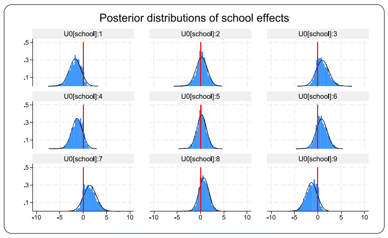
And more

```
. bayes: ...
```

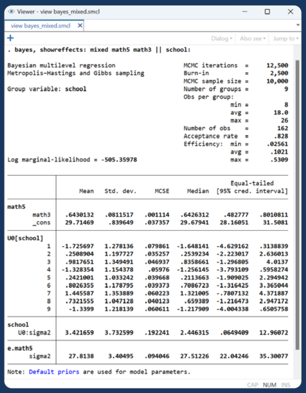
Learn more about regression models at stata.com/bayesian-regression-models

Multilevel models

Small number of groups?
Many hierarchical levels?
Want posterior distributions of random effects?



- Continuous, censored, binary, ordinal, and count outcomes
- Support for GLM and survival methods
- Random intercepts and coefficients
- Nested and crossed effects
- Multiple levels of hierarchy
- Random-effects covariance structures
- Multivariate nonlinear multilevel models
- Comprehensive Bayesian-features support



```
. bayes, showreffects: mixed math5 math3 || school:

Bayesian multilevel regression                  MCMC iterations  =     12,500
Metropolis-Hastings and Gibbs sampling          Burn-in          =      2,500
                                                MCMC sample size =     10,000
Group variable: school                          Number of groups =          9
                                                Obs per group:
                                                              min =          8
                                                              avg =       18.0
                                                              max =         26
                                                Number of obs    =        162
                                                Acceptance rate  =       .828
                                                Efficiency:  min =     .02561
                                                              avg =      .1021
Log marginal-likelihood = -505.35978                          max =      .5309

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
math5        |
       math3 |  .6430132   .0811517   .001114   .6426312    .482777   .8010811
       _cons |  29.71469    .839649   .037357   29.67941   28.16051    31.5081
-------------+----------------------------------------------------------------
U0[school]   |
           1 | -1.725697   1.278136   .079861  -1.648141  -4.629162   .3138839
           2 |  .2508904   1.197727   .035257   .2539234  -2.223017   2.636013
           3 |  .9817651   1.349491   .046937   .8358661  -1.296805     4.0137
           4 | -1.328354   1.154378    .05976  -1.256145  -3.793109   .5958274
           5 |  .2421001   1.033242   .039668   .2113663  -1.909025   2.294942
           6 |  .8026355   1.178795   .039373   .7086723  -1.316425   3.365044
           7 |  1.445587   1.353889   .060223   1.321005  -.7807132   4.371887
           8 |  .7321555   1.047128   .040123    .659389  -1.216473   2.947172
           9 |   -1.3399   1.218139   .060611  -1.217909  -4.004338   .6505758
-------------+----------------------------------------------------------------
school       |
   U0:sigma2 |  3.421659   3.732599   .192241   2.446315   .0649409   12.96072
-------------+----------------------------------------------------------------
e.math5      |
      sigma2 |   27.8138    3.40495   .094046   27.51226   22.04246   35.30077
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
```

Two-level models: Random intercepts

Fit regression of **math5** on **math3** with random intercepts by **school**

```
. bayes: mixed math5 math3 || school:
```

Display estimates of random effects

```
. bayes, showreffects:
         mixed math5 math3 || school:
```

(See output above)

Specify custom uniform priors instead of default normal priors for coefficients

```
. bayes, prior({math5:math3 _cons},
         uniform(-50,50)):
         mixed math5 math3 || school:
```

Plot posterior distributions of random intercepts

```
. bayesgraph histogram {U0}, byparm
```

(See graph above)

Two-level models: Random coefficients

Add random coefficient on **math3** by **school**

```
. bayes: mixed math5 math3 || school: math3
```

Specify unstructured covariance for random effects

```
. bayes: mixed math5 math3 || school: math3,
         covariance(unstructured)
```

Three-level models

Add random intercepts for teachers nested within schools

```
. bayes: mixed math5 math3 || school: || teacher:
```

Crossed-effects models

Include crossed random effects of primary and secondary schools

```
. bayes: mixed math5 math3 ||
         _all: R.primary || secondary:
```

Other multilevel models

Logistic regression

```
. bayes: melogit y x1 x2 || id:
```

Poisson regression

```
. bayes: mepoisson y x1 x2 || id:
```

Generalized linear model

```
. bayes: meglm y x1 x2 || id:,
         family(binomial) link(cloglog)
```

Ordered logistic regression

```
. bayes: meologit y x1 x2 || id:
```

Survival regression

```
. bayes: mestreg x1 x2 || id:,
         distribution(weibull)
```

And more

```
. bayes: any multilevel command ...
```

Learn more about multilevel models at stata.com/bayesian-multilevel-models

Multiple chains, predictions, and more

- Multiple chains
- Gelman–Rubin convergence diagnostics
- Bayesian predictions
- Posterior summaries of simulated values
- MCMC replicates
- Posterior predictive *p*-values

Two-level models: Random coefficients

Use option **nchains()** with **bayes:** or **bayesmh** to simulate multiple chains

Fit regression of **y** on covariates **x1** through **x10** and generate 3 chains

Viewer - view bayes_chains.smcl

```
. bayes, nchains(3): regress y x1-x10

Chain 1
  Burn-in ...
  Simulation ...

Chain 2
  Burn-in ...
  Simulation ...

Chain 3
  Burn-in ...
  Simulation ...

Model summary
------------------------------------------------------------------------------
Likelihood:
  y ~ regress(xb_y,{sigma2})

Priors:
  {y:x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 _cons} ~ normal(0,10000)              (1)
                                  {sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_y.

Bayesian linear regression                       Number of chains    =       3
Random-walk Metropolis-Hastings sampling         Per MCMC chain:
                                                     Iterations      =  12,500
                                                     Burn-in         =   2,500
                                                     Sample size     =  10,000
                                                 Number of obs       =     442
                                                 Avg acceptance rate =    .321
                                                 Avg efficiency: min =  .003771
                                                                 avg =   .01886
                                                                 max =    .1142
Avg log marginal-likelihood = -2457.6885         Max Gelman-Rubin Rc =    4.543

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
y            |
          x1 |  81.50951   87.57016   4.06945    71.7336  -50.84804   234.4335
          x2 | -167.0296    71.9136   6.76093   -167.887   -289.098  -42.73281
          x3 |  352.3836   99.94654   6.13795    360.123    193.378   505.5432
          x4 |  286.0075   78.16619   5.23075   275.4511   174.2242   426.8137
          x5 | -273.1433   164.7453    14.618  -295.8386  -501.8144  -16.60783
          x6 |   165.232    178.513   8.84052    231.598  -125.8265   352.7025
          x7 | -94.36373   114.5865   7.16308  -106.0647  -263.5826   99.65676
          x8 |   109.925   162.0595   13.0143   134.8212  -155.8764   332.5971
          x9 |   483.243   102.1253   6.00414   495.4546   307.5461   627.0424
         x10 |  42.76467    117.208   6.29098   65.24344  -146.8923    198.615
       _cons |  152.4957   2.780968   .103842   152.4606   147.3315   157.9954
-------------+----------------------------------------------------------------
      sigma2 |   3110.56   221.9696   3.79219   3100.067   2707.435   3566.951
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
Note: Default initial values are used for multiple chains.
Note: There is a high autocorrelation after 500 lags in at least one of the
      chains.
```

Gelman–Rubin convergence diagnostics

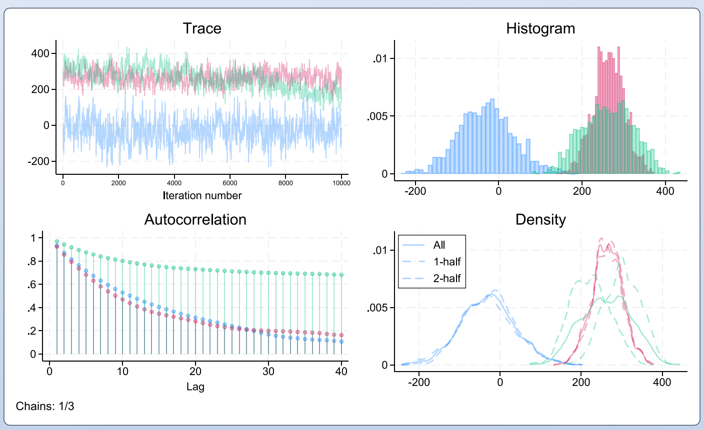
Check Gelman–Rubin convergence diagnostics

Viewer - view bayes_grubin.smcl

```
. bayesstats grubin, sort

Gelman-Rubin convergence diagnostic

Number of chains     =          3
MCMC size, per chain =     10,000
Max Gelman-Rubin Rc  =   4.542823

--------------------------
             |        Rc
-------------+------------
y            |
          x6 |  4.542823
          x8 |  3.376646
          x5 |  3.184339
         x10 |  3.089546
          x3 |  2.543104
          x7 |  2.447089
          x4 |  2.421061
          x1 |   2.40928
          x9 |  2.389624
          x2 |  1.680013
       _cons |  1.082658
-------------+------------
      sigma2 |  1.023543
--------------------------
Convergence rule: Rc < 1.1
```

Explore convergence visually for coefficient of **x6**

```
. bayesgraph diagnostics {y:x6}
```



Bayesian predictions

- Predict new values
- Check model fit using posterior predictive checks
- Compute functions of predicted values
- Specify your own prediction functions
- Obtain posterior summaries of predicted values
- Generate MCMC replicates
- Compute posterior predictive *p*-values

Bayesian predictions are outcome values simulated from the posterior predictive distribution. They are useful for predicting new outcome values and for checking model fit. Let's use **bayesmh** to fit a general Bayesian model.

```
. bayesmh y ..., likelihood(...) prior(...)
```

Posterior summaries of predictions

Compute posterior mean and credible intervals for all observations, and store them in variables **pmean**, **cril**, and **criu**

```
. bayespredict pmean, mean
. bayespredict cril criu, cri
```

Viewer - view bayes_predict.smcl

```
. list y pmean cril criu in 1/10
```

	y	pmean	cril	criu
1.	2.933	3.111	2.014	4.230
2.	4.614	4.478	3.362	5.576
3.	1.654	2.034	0.936	3.115
4.	2.025	2.234	1.130	3.362
5.	3.165	2.894	1.790	4.014
6.	1.372	2.337	1.227	3.452
7.	2.921	3.253	2.127	4.372
8.	2.699	2.274	1.158	3.359
9.	1.198	1.228	0.124	2.312
10.	3.097	2.767	1.655	3.872

MCMC replicates

Compute 6 MCMC replicates, and store them in variables **yrep1**, **yrep2**, and so on

```
. bayesreps yrep*, nreps(6)
```

List the first 10 observations

Viewer - view bayes_reps.smcl

```
. list y yrep* in 1/10
```

	y	yrep1	yrep2	yrep3	yrep4	yrep5	yrep6
1.	2.933	3.496	1.416	3.852	2.667	3.621	3.229
2.	4.614	4.794	3.462	4.354	6.245	3.848	4.822
3.	1.654	2.068	2.136	1.949	1.395	2.894	2.613
4.	2.025	2.568	2.234	2.780	1.966	1.804	2.230
5.	3.165	2.980	2.180	3.610	2.075	2.526	1.754
6.	1.372	1.584	2.110	2.932	0.956	2.149	2.438
7.	2.921	4.087	3.161	3.570	2.687	4.051	3.766
8.	2.699	1.731	1.846	2.216	2.065	2.109	1.994
9.	1.198	1.615	1.039	1.530	0.612	1.092	1.478
10.	3.097	2.281	2.774	2.799	2.162	4.188	3.107

Posterior predictive *p*-values

Simulate predictions for outcome **y**, and save them in **y_pred.dta**

```
. bayespredict {_ysim}, saving(y_pred)
```

Compute posterior predictive *p*-values; use Mata's built-in functions and your own

Viewer - view bayes_ppvalues.smcl

```
. bayesstats ppvalues (mean:@mean({_ysim})) (min:@min({_ysim})) (max:@max({_ysim}))
> (skew:@myskew({_ysim})) using y_pred

Posterior predictive summary   MCMC sample size =    10,000
```

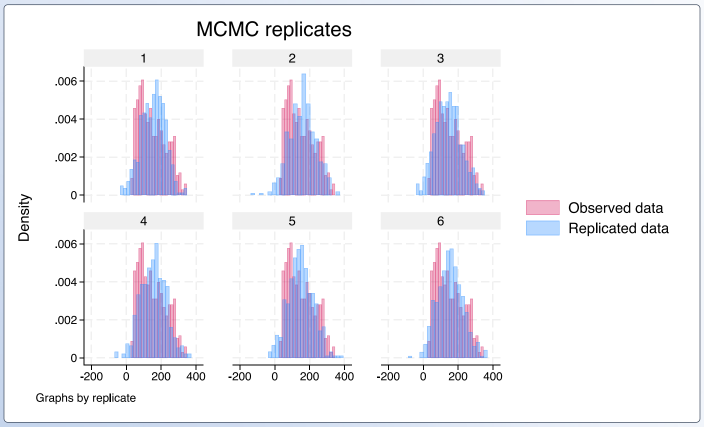
T	Mean	Std. dev.	E(T_obs)	P(T>=T_obs)
mean	3.045143	.0787588	3.044554	.5026
min	.5130189	.3401942	1.049675	.0365
max	5.84806	.3703789	5.703145	.626
skew	.1471358	.1660461	.1555046	.4806

Note: P(T>=T_obs) close to 0 or 1 indicates lack of fit.

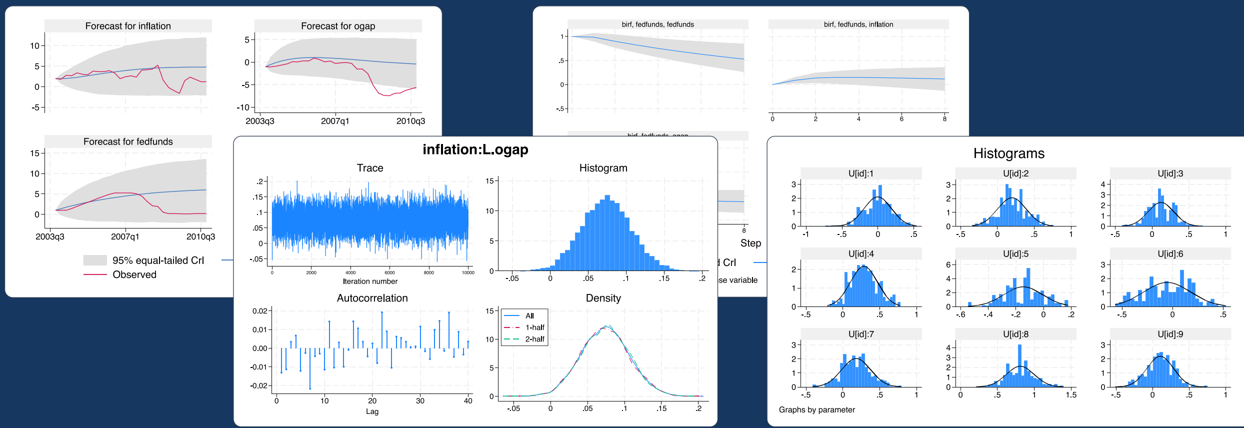
Viewer - view bayes_skew.smcl

```
. mata:
------------------------------------ mata (type end to exit) ------
: real scalar myskew(real colvector x) {
>         return (sqrt(length(x))*sum((x:-mean(x)):^3)/(sum((x:-mean(x)):^2)^1.5))
> }

: end
```

Plot distributions of MCMC replicates



Bayesian econometrics

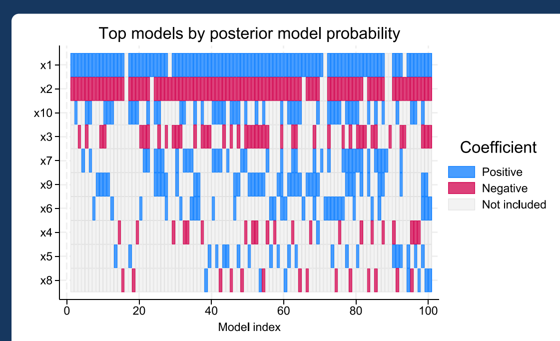


- Panel-data models
- Dynamic forecasting
- VAR models
- IRF and FEVD analysis
- Linear and nonlinear DSGE models
- And more

stata.com/bayesian-econometrics

New in Stata 18

- **Bayesian model averaging (BMA)**
  – BMA for linear regression
  – Influential models and important predictors
  – Posterior distribution plots for regression coefficients
  – Model-probability plots
  – Variable-inclusion maps
  – Model fit and predictive performance



stata.com/bma