New in

STATA 16

NEW IN BAYESIAN ANALYSIS:

Multiple chains, predictions, & more

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

Multiple chains

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

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

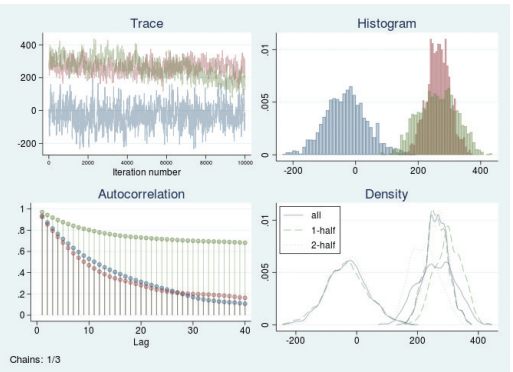
```
. bayes, nchains(3): regress y x1-x10

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   4.06045   71.7336  -50.84804   234.4335
          x2 |  -167.0296   71.9136   6.76093  -167.887   -280.008  -42.73281
          x3 |   352.3836   90.04654   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 |   100.925   162.0595   13.0143   134.8212  -155.8764   332.5071
          x9 |   483.243   102.1253   6.00414   495.4546   307.5461   627.0424
         x10 |   42.76467   117.208   6.29098   65.24344  -146.8023   198.615
       _cons |   152.4057   2.780968   .103842   152.4606   147.3315   157.9054
-------------+----------------------------------------------------------------
      sigma2 |   3110.56   221.9696   3.79219   3100.067   2707.455   3566.951
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
Note: Default initial values are used for multiple chains.
```

Check Gelman–Rubin convergence diagnostics

```
. 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
```

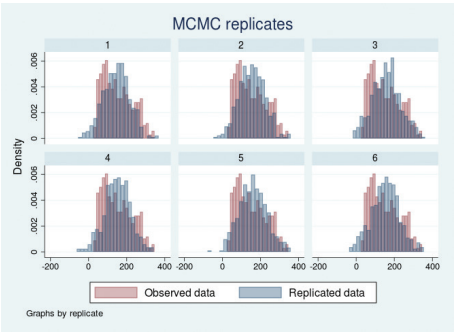
Explore convergence visually for coefficient of **x6**

```
. 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
```

view pmeancri.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

view 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

view 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.

view 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
```

Perform analyses using GUI