New Bayesian features: Predictions, multiple chains, and more

Yulia Marchenko

StataCorp LLC

2020 London Stata Conference
New Bayesian features in a nutshell
Stata’s Bayesian suite of commands
Introduction to Bayesian analysis
Motivating example: Bayesian lasso
Bayesian predictions
Multiple chains
Summary
Additional resources
References
Stata 16 provides many new Bayesian features: multiple chains, Gelman–Rubin convergence diagnostic, predictions, posterior predictive checks, and more.
Multiple chains. Simulate multiple chains conveniently using new option nchains() with bayes: and bayesmh.

- Type

  . bayes, nchains(#) : ...

or

  . bayesmh ..., nchains(#) ...

The commands will properly combine all chains to produce a more precise final result.

- Use default chain-specific initial values or use new options initall() and init#() to specify your own.
Bayesian postestimation features will automatically handle multiple chains properly. For instance, simply type

```
.bayesgraph diagnostics ...
```

to see graphical diagnostics for all chains. Or use new options `chains()` and `sepchains` to obtain results for specific chains.

Use unofficial command `bayesparallel` to simulate chains in parallel using multiple processors:

```
.net install bayesparallel, from("https://www.stata.com/users/nbalov")
.bayesparallel, nproc(#): bayes, nchains(#): ...
.bayesparallel, nproc(#): bayesmh ..., nchains(#)
```
Gelman–Rubin convergence diagnostic. When you run multiple chains, \texttt{bayesmh} and \texttt{bayes:} automatically compute and report the maximum Gelman–Rubin statistic across model parameters.

Type

\begin{verbatim}
. bayesstats grubin
\end{verbatim}

to obtain the Gelman–Rubin diagnostic for each model parameter.
Bayesian predictions. Use \texttt{bayespredict} to compute various Bayesian predictions and their posterior summaries.

- Compute and save simulated outcomes, their expected values, and residuals in a new dataset:
  \[
  \texttt{bayespredict \{\_ysim\} \{\_mu\} \{\_resid\}, saving(filename)}
  \]

- Or compute posterior summaries of simulated outcomes and save them in a new variable in the current dataset:
  \[
  \texttt{bayespredict pmean, mean}
  \]

Compute posterior means, medians, credible intervals, and more.

- Summarize predicted quantities as any other model parameter:
  \[
  \texttt{bayesstats summary \{\_ysim\} using filename}
  \]

Use with any other Bayesian postestimation command.
**Posterior predictive checks.** Use bayespredict to compute replicated outcomes for comparison with the observed outcomes. Follow up with bayesstats ppvalues to compute posterior predictive $p$-values for a more formal comparison.

**MCMC replicates.** Use bayesreps to generate a subset of Markov chain Monte Carlo (MCMC) replicates for a quick comparison of the observed and replicated data.

**New priors:** Pareto for continuous positive parameters, pareto(); multivariate beta (Dirichlet) for probability vectors, dirichlet(); and geometric for count parameters, geometric().

**Faster Bayesian multilevel models.** bayes: with multilevel models such as bayes: mixed now runs faster!
### Stata’s Bayesian suite of commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Estimation</strong></td>
<td></td>
</tr>
<tr>
<td>bayes:</td>
<td>Bayesian regression models</td>
</tr>
<tr>
<td>bayesmh</td>
<td>General Bayesian models using MH</td>
</tr>
<tr>
<td>bayesmh evaluators</td>
<td>User-defined Bayesian models using MH</td>
</tr>
<tr>
<td><strong>Postestimation</strong></td>
<td></td>
</tr>
<tr>
<td>bayesgraph</td>
<td>Graphical convergence diagnostics</td>
</tr>
<tr>
<td>bayesstats ess</td>
<td>Effective sample sizes and more</td>
</tr>
<tr>
<td>bayesstats summary</td>
<td>Summary statistics</td>
</tr>
<tr>
<td>bayesstats ic</td>
<td>Information criteria and Bayes factors</td>
</tr>
<tr>
<td>bayesstats ppvalues</td>
<td>Posterior predictive p-values New in Stata 16</td>
</tr>
<tr>
<td>bayestest model</td>
<td>Model posterior probabilities</td>
</tr>
<tr>
<td>bayestest interval</td>
<td>Interval hypothesis testing</td>
</tr>
<tr>
<td>bayespredict</td>
<td>Bayesian predictions New in Stata 16</td>
</tr>
<tr>
<td>bayesreps</td>
<td>MCMC replicates New in Stata 16</td>
</tr>
</tbody>
</table>

New Bayesian features
What is Bayesian analysis?

Bayesian analysis is a statistical paradigm that answers research questions about unknown parameters using probability statements.

- What is the probability that a person accused of a crime is guilty?
- What is the probability that treatment A is more cost effective than treatment B for a specific health care provider?
- What is the probability that the odds ratio is between 0.3 and 0.5?
- What is the probability that three out of five quiz questions will be answered correctly by students?
Why Bayesian analysis?

You may be interested in Bayesian analysis if

- you have some prior information available from previous studies that you would like to incorporate in your analysis. For example, in a study of preterm birthweights, it would be sensible to incorporate the prior information that the probability of a mean birthweight above 15 pounds is negligible. Or,

- your research problem may require you to answer a question: What is the probability that my parameter of interest belongs to a specific range? For example, what is the probability that an odds ratio is between 0.2 and 0.5? Or,

- you want to assign a probability to your research hypothesis. For example, what is the probability that a person accused of a crime is guilty?

- And more.
Assumptions

- Observed data sample $D$ is fixed and model parameters $\theta$ are random.
- $D$ is viewed as a result of a one-time experiment.
- A parameter is summarized by an entire distribution of values instead of one fixed value as in classical frequentist analysis.
- There is some prior (before seeing the data!) knowledge about $\theta$ formulated as a prior distribution $p(\theta)$.
- After data $D$ are observed, the information about $\theta$ is updated based on the likelihood $f(D|\theta)$.
- Information is updated by using the Bayes rule to form a posterior distribution $p(\theta|D)$:

$$p(\theta|D) = \frac{f(D|\theta)p(\theta)}{p(D)}$$

where $p(D)$ is the marginal distribution of the data $D$. 
Inference

- Estimating a posterior distribution $p(\theta|D)$ is at the heart of Bayesian analysis.
- Various summaries of this distribution are used for inference.
- Point estimates: posterior means, modes, medians, percentiles.
- Interval estimates: credible intervals (CrI)—(fixed) ranges to which a parameter is known to belong with a pre-specified probability.
- Monte-Carlo standard error (MCSE)—represents precision about posterior mean estimates.
- Hypothesis testing—assign probability to any hypothesis of interest
- Model comparison: model posterior probabilities, Bayes factors
- **Prediction**: out-of-sample, future observations, posterior predictive $p$-values, and more
Challenges

- Potential subjectivity in specifying prior information—noninformative priors or sensitivity analysis to various choices of informative priors.
- Computationally demanding—involves intractable integrals that can only be computed using intensive numerical methods such as MCMC.
Advantages

Bayesian inference:

- is universal—it is based on the Bayes rule which applies equally to all models;
- incorporates prior information;
- provides the entire posterior distribution of model parameters;
- is exact, in the sense that it is based on the actual posterior distribution rather than on asymptotic normality in contrast with many frequentist estimation procedures; and
- provides straightforward and more intuitive interpretation of the results in terms of probabilities.
Diabetes data (Efron et al. 2004)

- 442 diabetes patients
- Outcome of interest: Measure of disease progression (one year after baseline)
- 10 baseline covariates: age, sex, body mass index, mean arterial pressure, and 6 blood serum measurements
- Covariates standardized to have mean zero and a sum of squares across all observations of one
- Objectives: Determine which variables are important to predict the outcome and obtain accurate predictions for future patients
New Bayesian features

Motivating example: Bayesian lasso

Diabetes data

. use diabetes_std
  (Diabetes data from Efron et al. (2004) with standardized covariates)

. describe

Contains data from diabetes_std.dta
  Diabetes data from Efron et al. (2004) with standardized covariates
  obs: 442
  vars: 12
  9 Sep 2020 17:11
  (_dta has notes)

<table>
<thead>
<tr>
<th>variable</th>
<th>storage</th>
<th>display</th>
<th>value</th>
<th>variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>int</td>
<td>%8.0g</td>
<td></td>
<td>Measure of disease progression</td>
</tr>
<tr>
<td>age</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>Age</td>
</tr>
<tr>
<td>sex</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>Sex</td>
</tr>
<tr>
<td>bmi</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>Body mass index</td>
</tr>
<tr>
<td>map</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>Mean arterial pressure</td>
</tr>
<tr>
<td>tc</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>Total cholesterol</td>
</tr>
<tr>
<td>ldl</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>LDL cholesterol level</td>
</tr>
<tr>
<td>hdl</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>HDL cholesterol level</td>
</tr>
<tr>
<td>tch</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>TCh blood serum level</td>
</tr>
<tr>
<td>ltg</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>LTG blood serum level</td>
</tr>
<tr>
<td>glu</td>
<td>float</td>
<td>%9.0g</td>
<td></td>
<td>Glucose blood serum level</td>
</tr>
<tr>
<td>id</td>
<td>int</td>
<td>%9.0g</td>
<td></td>
<td>* Subject ID</td>
</tr>
</tbody>
</table>

* indicated variables have notes
Bayesian lasso (Park and Casella 2008)

- **Idea**: Use Laplace prior with penalty parameter for regression coefficients to mimic the L1 penalty used in classical lasso
- **Advantages**: Proper inference for model parameters and estimating uncertainty for predictions
- Linear regression to model outcome $y$—likelihood function
- Priors for regression coefficients—Laplace prior with zero mean and the scale parameter that depends on the error variance and penalty parameter
- Prior for intercept—vague Normal prior, $N(0, 10^6)$
- Prior for error variance—Jeffreys, $1/\sigma^2$
- Prior for penalty parameter—Gamma prior with shape 1 and scale $1/1.78$ (per authors); 1.78 is specific to this dataset
- We sample all parameters in separate blocks to improve sampling efficiency
New Bayesian features

Motivating example: Bayesian lasso

Bayesian lasso

.bayesmh y age sex bmi map tc ldl hdl tch ltg glu, ///
> likelihood(normal({sigma2})) ///
> prior({y:age sex bmi map tc ldl hdl tch ltg glu}, ///
>     laplace(0, (sqrt({sigma2}/{lam2})))) ///
> prior({sigma2}, jeffreys) ///
> prior({y:_cons}, normal(0, 1e6)) ///
> prior({lam2=1}, gamma(1, 1/1.78)) ///
> block({y:} {sigma2} {lam2}, split) ///
> rseed(16) dots

(Continued on next page)
New Bayesian features

Motivating example: Bayesian lasso

Bayesian lasso

Burn-in 2500 aaaaaaaaaa1000aaaaaaaaaa2000aaaaa done
> 000..........6000..........7000..........8000..........9000..........10000 done

Model summary

Likelihood:
   y ~ normal(xb_y,{sigma2})

Priors:
   {y:age sex bmi map tc ldl hdl tch ltg glu} ~ laplace(0,<expr1>) (1)
   {y:_cons} ~ normal(0,1e6) (1)
   {sigma2} ~ jeffreys

Hyperprior:
   {lam2} ~ gamma(1,1/1.78)

Expression:
   expr1 : sqrt({sigma2}/{lam2})

(1) Parameters are elements of the linear form xb_y.

(Continued on next page)
Bayesian normal regression
Random-walk Metropolis-Hastings sampling

<table>
<thead>
<tr>
<th>MCMC iterations</th>
<th>= 12,500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burn-in</td>
<td>= 2,500</td>
</tr>
<tr>
<td>MCMC sample size</td>
<td>= 10,000</td>
</tr>
<tr>
<td>Number of obs</td>
<td>= 442</td>
</tr>
<tr>
<td>Acceptance rate</td>
<td>= .4379</td>
</tr>
<tr>
<td>Efficiency: min</td>
<td>= .0152</td>
</tr>
<tr>
<td>avg</td>
<td>= .1025</td>
</tr>
<tr>
<td>max</td>
<td>= .2299</td>
</tr>
</tbody>
</table>

Log marginal-likelihood = -2415.7171

<p>| Equal-tailed | | | | |</p>
<table>
<thead>
<tr>
<th>Mean</th>
<th>Std. Dev.</th>
<th>MCSE</th>
<th>Median</th>
<th>[95% Cred. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>-2.478525</td>
<td>52.97851</td>
<td>1.26623</td>
<td>-2.593401</td>
</tr>
<tr>
<td>sex</td>
<td>-209.4461</td>
<td>61.21979</td>
<td>1.70006</td>
<td>-211.1479</td>
</tr>
<tr>
<td>bmi</td>
<td>522.1367</td>
<td>66.76557</td>
<td>1.8115</td>
<td>520.6348</td>
</tr>
<tr>
<td>map</td>
<td>304.1617</td>
<td>65.26244</td>
<td>1.77912</td>
<td>306.1749</td>
</tr>
<tr>
<td>tc</td>
<td>-172.2847</td>
<td>157.5097</td>
<td>12.7739</td>
<td>-159.4816</td>
</tr>
<tr>
<td>ldl</td>
<td>1.304382</td>
<td>128.3598</td>
<td>9.96343</td>
<td>-7.796492</td>
</tr>
<tr>
<td>hdl</td>
<td>-158.8146</td>
<td>112.6562</td>
<td>6.82563</td>
<td>-158.1347</td>
</tr>
<tr>
<td>tch</td>
<td>91.27437</td>
<td>111.8483</td>
<td>6.06667</td>
<td>86.32462</td>
</tr>
<tr>
<td>ltg</td>
<td>515.5167</td>
<td>94.06607</td>
<td>5.83902</td>
<td>509.9952</td>
</tr>
<tr>
<td>glu</td>
<td>67.94583</td>
<td>62.86024</td>
<td>1.69235</td>
<td>66.11433</td>
</tr>
<tr>
<td>_cons</td>
<td>152.0964</td>
<td>2.545592</td>
<td>.053095</td>
<td>152.0963</td>
</tr>
<tr>
<td>sigma2</td>
<td>2961.246</td>
<td>207.0183</td>
<td>4.79372</td>
<td>2949.282</td>
</tr>
<tr>
<td>lam2</td>
<td>0.0889046</td>
<td>0.055257</td>
<td>0.001899</td>
<td>0.0769573</td>
</tr>
</tbody>
</table>

Note: Adaptation tolerance is not met in at least one of the blocks.
Objective 1—Important predictors:

```
.bayesstats summary (age:<0) (sex:{y:sex}<0) (bmi:{y:bmi}<0) ///
> (map:{y:map}<0) (tc:{y:tc}<0) (ldl:{y:ldl}<0) (hdl:{y:hdl}<0) ///
> (tch:{y:tch}<0) (ltg:{y:ltg}<0) (glu:{y:glu}<0), nolegend
Posterior summary statistics MCMC sample size = 10,000
```

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>MCSE</th>
<th>Median</th>
<th>Equal-tailed [95% Cred. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>.5277</td>
<td>.4992571</td>
<td>.011353</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>sex</td>
<td>.9997</td>
<td>.0173188</td>
<td>.000224</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>bmi</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>map</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>tc</td>
<td>.8815</td>
<td>.3232154</td>
<td>.018971</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>ldl</td>
<td>.5301</td>
<td>.4997118</td>
<td>.029122</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>hdl</td>
<td>.9226</td>
<td>.2672384</td>
<td>.01198</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>tch</td>
<td>.1992</td>
<td>.3994187</td>
<td>.016507</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ltg</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>glu</td>
<td>.1417</td>
<td>.3487596</td>
<td>.008577</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

The probabilities that the coefficients for `age` and `ldl` are less than 0 are close to 0.5, so we may consider these two variables not important.

Objective 2—Predictions; see next
Bayesian predictions play two important roles in Bayesian analysis:

- Prediction (estimation) of new or future outcomes, and
- Model goodness of fit, also known as posterior predictive model checks.

*Bayesian predictions* are outcome values simulated from the posterior predictive distribution, which is the distribution of the unobserved (future) data given the observed data.

More generally, Bayesian predictions can be viewed as any function of simulated outcomes.
Posterior predictive distribution (PPD) for a new outcome value $y^{new}$ given observed data $y$:

$$p(y^{new}|y) = \int f(y^{new}|\theta)p(\theta|y)d\theta$$

where $f(y^{new}|\theta)$ is the likelihood of $y^{new}$ given $\theta$ and $p(\theta|y)$ is the posterior distribution of $\theta$ given $y$.

Bayesian prediction for $y^{new}$ is a realization from the above PPD. I will also use the term *simulated outcomes* to refer to such realizations.
Simulating from PPD

Like posterior distribution of model parameters $p(\theta|y)$, PPD $p(y_{\text{new}}|y)$ usually does not have a closed form and must be approximated. Formula

$$p(y_{\text{new}}|y) = \int f(y_{\text{new}}|\theta)p(\theta|y)d\theta$$

provides a way to simulate values from PPD using the following two-step procedure.

1. Simulate $\theta^t$ from $p(\theta|y)$
2. Simulate $y^t$ from $f(y_{\text{new}}|\theta^t)$
3. Repeat steps 1 and 2 for $t = 1, 2, \ldots, T$ MCMC iterations

A sample $\{y^1, y^2, \ldots, y^T\}$ represents a sample from $p(y_{\text{new}}|y)$. Unlike classical predictions, we will have a sample of $T$ values for each (new) observation.

An MCMC sample $\{\theta^1, \theta^2, \ldots, \theta^T\}$ is usually available after the main estimation, so Bayesian prediction simplifies to step 2 only.
Example: Bayesian lasso prediction

- Recall our Bayesian lasso model:

  . bayesmh y age sex bmi map tc ldl hdl tch ltg glu, ///
    likelihood(normal({sigma2})) ///
    prior({y:age sex bmi map tc ldl hdl tch ltg glu}, ///
       laplace(0, (sqrt({sigma2}/lam2)))) ///
    prior({sigma2}, jeffreys) ///
    prior({y:_cons}, normal(0, 1e6)) ///
    prior({lam2=1}, gamma(1, 1/1.78)) ///
    block({y:} {sigma2} {lam2}, split) ///
    rseed(16) dots

  (output omitted)

- Save MCMC posterior sample of model parameters:

  . bayesmh, saving(blasso_mcmc)
  note: file blasso_mcmc.dta saved
Compute Bayesian predictions

- Use `bayespredict` to simulate and save outcome values:
  
  ```bash
  . bayespredict {_ysim}, saving(blasso_pred) rseed(16)
  Computing predictions ...
  file blasso_pred.dta saved
  file blasso_pred.ster saved
  ```

- **Option `saving()`** is required when simulating outcome values `{_ysim}`.

- Simulated values and other system variables are saved in `blasso_pred.dta`.

- Auxiliary estimation results used by `bayespredict` are saved in `blasso_pred.ster`.

- Remember to erase these files when you no longer need them.
Stata dataset created by `bayespredict` with simulated values:

```
. describe using blasso_pred
Contains data
    obs:     10,000  9 Sep 2020 14:30
    vars:      887

<table>
<thead>
<tr>
<th>variable name</th>
<th>storage</th>
<th>display format</th>
<th>value label</th>
</tr>
</thead>
<tbody>
<tr>
<td>_chain</td>
<td>int</td>
<td>%8.0g</td>
<td>Chain identifier</td>
</tr>
<tr>
<td>_index</td>
<td>int</td>
<td>%8.0g</td>
<td>Iteration number</td>
</tr>
<tr>
<td>_ysim1_1</td>
<td>double</td>
<td>%10.0g</td>
<td>Simulated y, obs. #1</td>
</tr>
<tr>
<td>_ysim1_2</td>
<td>double</td>
<td>%10.0g</td>
<td>Simulated y, obs. #2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(output omitted)</td>
</tr>
<tr>
<td>_ysim1_441</td>
<td>double</td>
<td>%10.0g</td>
<td>Simulated y, obs. #441</td>
</tr>
<tr>
<td>_ysim1_442</td>
<td>double</td>
<td>%10.0g</td>
<td>Simulated y, obs. #442</td>
</tr>
<tr>
<td>_mu1_1</td>
<td>double</td>
<td>%10.0g</td>
<td>Expected values for y, obs. #1</td>
</tr>
<tr>
<td>_mu1_2</td>
<td>double</td>
<td>%10.0g</td>
<td>Expected values for y, obs. #2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(output omitted)</td>
</tr>
<tr>
<td>_mu1_441</td>
<td>double</td>
<td>%10.0g</td>
<td>Expected values for y, obs. #441</td>
</tr>
<tr>
<td>_mu1_442</td>
<td>double</td>
<td>%10.0g</td>
<td>Expected values for y, obs. #442</td>
</tr>
<tr>
<td>_frequency</td>
<td>byte</td>
<td>%8.0g</td>
<td>Frequency weight</td>
</tr>
</tbody>
</table>
```

Sorted by:
New Bayesian features

- Bayesian predictions
- Example: Bayesian lasso prediction

Histograms of Bayesian predictions

Histograms of simulated values for the first 12 observations:

```
. bayesgraph histogram {_ysim[1/12]} using blasso_pred, byparm
```
## Summary of Bayesian predictions

```
.bayesstats summary {_ysim[1/12]} using blasso_pred
Posterior summary statistics MCMC sample size = 10,000

<table>
<thead>
<tr>
<th>_ysim1_1</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>MCSE</th>
<th>Median</th>
<th>Equal-tailed [95% Cred. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>203.7014</td>
<td>54.97776</td>
<td>.558382</td>
<td>203.1887</td>
<td>98.05673 - 312.5681</td>
</tr>
<tr>
<td>_ysim1_2</td>
<td>71.25238</td>
<td>55.05362</td>
<td>.550536</td>
<td>70.95502</td>
<td>-35.91061 - 179.3321</td>
</tr>
<tr>
<td>_ysim1_3</td>
<td>175.3088</td>
<td>55.09297</td>
<td>.55093</td>
<td>175.3602</td>
<td>67.95822 - 284.0823</td>
</tr>
<tr>
<td>_ysim1_4</td>
<td>161.6022</td>
<td>55.3058</td>
<td>.577559</td>
<td>161.3602</td>
<td>67.95822 - 284.0823</td>
</tr>
<tr>
<td>_ysim1_5</td>
<td>127.0087</td>
<td>55.34909</td>
<td>.553491</td>
<td>127.128</td>
<td>18.79625 - 235.7958</td>
</tr>
<tr>
<td>_ysim1_6</td>
<td>104.5405</td>
<td>54.35416</td>
<td>.543542</td>
<td>105.0999</td>
<td>-2.76073 - 211.2213</td>
</tr>
<tr>
<td>_ysim1_7</td>
<td>80.33914</td>
<td>55.64711</td>
<td>.55933</td>
<td>80.04908</td>
<td>-28.58423 - 189.9953</td>
</tr>
<tr>
<td>_ysim1_8</td>
<td>124.9409</td>
<td>55.43717</td>
<td>.554372</td>
<td>124.5975</td>
<td>16.57208 - 234.4114</td>
</tr>
<tr>
<td>_ysim1_9</td>
<td>160.7804</td>
<td>55.37094</td>
<td>.561705</td>
<td>160.6604</td>
<td>51.05089 - 269.877</td>
</tr>
<tr>
<td>_ysim1_10</td>
<td>212.4139</td>
<td>54.70172</td>
<td>.554331</td>
<td>211.9675</td>
<td>105.3659 - 319.5306</td>
</tr>
<tr>
<td>_ysim1_11</td>
<td>99.4205</td>
<td>54.82602</td>
<td>.574513</td>
<td>99.69542</td>
<td>-9.512793 - 203.8927</td>
</tr>
<tr>
<td>_ysim1_12</td>
<td>104.5963</td>
<td>55.64342</td>
<td>.588025</td>
<td>104.4855</td>
<td>-4.058576 - 215.4143</td>
</tr>
</tbody>
</table>
```
Hypothesis testing for Bayesian predictions

Compute probability that the first simulated value is greater than 100:

```
    . bayestest interval {_ysim[1]} using blasso_pred, lower(100)
```

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>MCSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>prob1</td>
<td>.9733</td>
<td>0.16121</td>
<td>.0016834</td>
</tr>
</tbody>
</table>

Interval tests MCMC sample size = 10,000

prob1 : {_ysim[1]} > 100
Bayesian predictions are useful for model checking by performing so-called posterior predictive checks.

These checks compare various characteristics of the posterior predictive distribution with those observed in the data.

For regression models, PPD depends on covariate data $X$, $p(y^{new} | y) = p(y^{new} | y, X)$.

Thus, a concept of replicated outcomes, $y^{rep}$, is introduced to refer to simulated outcomes from PPD $p(y^{new} | y, X^{obs})$ that uses observed covariate data $X^{obs}$. 
Posterior predictive \( p \)-values (PPPs)

- Posterior predictive \( p \)-values (PPPs) formalize posterior predictive checks.
- They quantify the discrepancy between the summaries of the observed and replicated data.
- Consider a test statistic \( T(y) \) such as a sample mean or median. PPP for \( T(y) \) is defined as

\[
q(T) = Pr(T(y^{rep}) \geq T(y^{obs}) | y^{obs}, X^{obs})
\]

- In a Bayesian context, \( T(y) = T(y, \theta) \) may also depend on model parameters \( \theta \) and is then referred to as a test quantity.
- Values of PPPs close to zero or one indicate lack of fit.
- Use `bayesstats ppvalues` to compute PPPs in Stata.
New Bayesian features

Bayesian predictions

Example: PPPs to check Bayesian lasso fit

## PPPs for mean and variance

```stata
.bayesstats ppvalues (mean:@mean(_ysim)) (var:@variance(_ysim)) using blasso_pred
Posterior predictive summary   MCMC sample size = 10,000

<table>
<thead>
<tr>
<th>T</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>E(T_obs)</th>
<th>P(T&gt;=T_obs)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>152.0664</td>
<td>3.635561</td>
<td>152.1335</td>
</tr>
<tr>
<td></td>
<td>var</td>
<td>5934.96</td>
<td>487.52</td>
<td>5943.331</td>
</tr>
</tbody>
</table>

Note: P(T>=T_obs) close to 0 or 1 indicates lack of fit.
Define Mata function `skew()` that computes skewness:

```
. mata:

: real scalar skew(real colvector x) {
>     return (sqrt(length(x))*sum((x:-mean(x)):^3)/(sum((x:-mean(x)):^2)^1.5))
> }

: end
```

Use `skew()` with `bayesstats ppvalues`:

```
. bayesstats ppvalues (skewness:@skew({_ysim})) (min:@min({_ysim})) ///
>   (max:@max({_ysim})) using blasso_pred
```

Posterior predictive summary  MCMC sample size = 10,000

<table>
<thead>
<tr>
<th>T</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>E(T_obs)</th>
<th>P(T&gt;=T_obs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>skewness</td>
<td>.0899553</td>
<td>.1045585</td>
<td>.4390664</td>
<td>.0002</td>
</tr>
<tr>
<td>min</td>
<td>-62.72501</td>
<td>24.93968</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>max</td>
<td>381.8695</td>
<td>26.75575</td>
<td>346</td>
<td>.9319</td>
</tr>
</tbody>
</table>

Note: `P(T>=T_obs)` close to 0 or 1 indicates lack of fit.
Out-of-sample predictions

- Let’s check prediction accuracy of our Bayesian lasso model and compare it with classical lasso.
- We will use `splitsample` (new in Stata 16) to randomly split our diabetes data into the training (sample=1) and test (sample=2) samples.

  - `. use diabetes_std`
  - `. splitsample, generate(sample) rseed(12345)`

- We will fit classical and Bayesian lassos using the training sample.
- We will then predict the outcome using the test sample and compute mean squared prediction errors for classical and Bayesian lassos.
Classical lasso prediction

Fit classical lasso using the training sample and save predicted values from the test sample in variable *yhat*:

```
.lasso linear y age sex bmi map tc ldl hdl tch ltg glu if sample==1, nolog
Lasso linear model
No. of obs = 221
No. of covariates = 10
Selection: Cross-validation
No. of CV folds = 10
```

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>lambda</th>
<th>No. of nonzero coef.</th>
<th>Out-of-sample R-squared</th>
<th>CV mean prediction error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>first lambda</td>
<td>44.44473</td>
<td>0</td>
<td>-0.0031</td>
<td>5855.735</td>
</tr>
<tr>
<td>33</td>
<td>lambda before</td>
<td>2.264076</td>
<td>6</td>
<td>0.4237</td>
<td>3364.209</td>
</tr>
<tr>
<td>* 34</td>
<td>selected lambda</td>
<td>2.062942</td>
<td>6</td>
<td>0.4238</td>
<td>3363.86</td>
</tr>
<tr>
<td>35</td>
<td>lambda after</td>
<td>1.879676</td>
<td>6</td>
<td>0.4235</td>
<td>3365.395</td>
</tr>
<tr>
<td>38</td>
<td>last lambda</td>
<td>1.421906</td>
<td>6</td>
<td>0.4220</td>
<td>3374.295</td>
</tr>
</tbody>
</table>

* *lambda selected by cross-validation.*

```
.predict double yhat if sample==2
(options xb penalized assumed; linear prediction with penalized coefficients)
.gen double err_lasso = (y-yhat)^2
(221 missing values generated)
```
Fit Bayesian lasso using the training sample:

```
.bayesmh y age sex bmi map tc ldl hdl tch ltg glu if sample==1, \
>   likelihood(normal({sigma2})) \
>   prior({y:age sex bmi map tc ldl hdl tch ltg glu}, \
>       laplace(0, (sqrt({sigma2}/{lam2})))) \
>   prior({sigma2}, jeffreys) \
>   prior({y:_cons}, normal(0, 1e6)) \
>   prior({lam2=1}, gamma(1, 1/1.78)) \
>   block({y:} {sigma2} {lam2}, split) \
>   rseed(16) dots saving(blassosplit_mcmc)
```

(Continued on next page)
Burn-in 2500 aaaaaaaaaa1000aaaaaaaaaa2000aaaaa done
Simulation 10000 ........1000...........2000...........3000...........4000...........5
> 000...........6000...........7000...........8000...........9000...........10000 done

Model summary

Likelihood:
  \( y \sim \text{normal}(xb_y,\{\sigma^2}\) \)

Priors:
  \{y:\text{age sex bmi map tc ldl hdl tch ltg glu}\} \sim \text{laplace}(0,\text{expr1}) \quad (1)
  \{y:\_cons\} \sim \text{normal}(0,1e6) \quad (1)
  \{\sigma^2\} \sim \text{jeffreys}

Hyperprior:
  \{\text{lam2}\} \sim \text{gamma}(1,1/1.78)

Expression:
  \text{expr1} : \sqrt{\{\sigma^2/\text{lam2}\}}

(1) Parameters are elements of the linear form \(xb_y\).

(Continued on next page)
Bayesian normal regression
Random-walk Metropolis-Hastings sampling
MCMC iterations = 12,500
Burn-in = 2,500
MCMC sample size = 10,000
Number of obs = 221
Acceptance rate = .4404
Efficiency: min = .02513
avg = .1081
max = .2379
Log marginal-likelihood = -1225.3231

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>MCSE</th>
<th>Median</th>
<th>[95% Cred. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>22.90211</td>
<td>70.97209</td>
<td>1.65398</td>
<td>18.49825</td>
<td>-109.8378  167.3328</td>
</tr>
<tr>
<td>sex</td>
<td>-147.1624</td>
<td>91.8814</td>
<td>2.60312</td>
<td>-144.2808</td>
<td>-335.182  20.59289</td>
</tr>
<tr>
<td>bmi</td>
<td>523.9505</td>
<td>99.27119</td>
<td>2.74922</td>
<td>526.4859</td>
<td>326.6205  724.9461</td>
</tr>
<tr>
<td>map</td>
<td>279.1178</td>
<td>100.3706</td>
<td>2.87727</td>
<td>280.5395</td>
<td>73.24508  477.0849</td>
</tr>
<tr>
<td>tc</td>
<td>-10.10333</td>
<td>150.7421</td>
<td>9.50814</td>
<td>-7.539645</td>
<td>-329.4431 290.428</td>
</tr>
<tr>
<td>hdl</td>
<td>-149.4535</td>
<td>136.8265</td>
<td>6.65266</td>
<td>-145.7912</td>
<td>-437.285  108.0459</td>
</tr>
<tr>
<td>tch</td>
<td>161.9482</td>
<td>155.7781</td>
<td>7.82029</td>
<td>147.5103</td>
<td>-115.6871 500.1832</td>
</tr>
<tr>
<td>ltg</td>
<td>312.8631</td>
<td>124.2238</td>
<td>5.34773</td>
<td>315.6055</td>
<td>72.72891  559.7625</td>
</tr>
<tr>
<td>glu</td>
<td>24.37885</td>
<td>79.45832</td>
<td>1.99475</td>
<td>20.4382</td>
<td>-125.1158 191.8139</td>
</tr>
<tr>
<td>_cons</td>
<td>149.7803</td>
<td>3.844837</td>
<td>.078828</td>
<td>149.7844</td>
<td>141.9715  157.2013</td>
</tr>
<tr>
<td>sigma2</td>
<td>3310.895</td>
<td>329.993</td>
<td>8.00504</td>
<td>3285.107</td>
<td>2733.671  4015.07</td>
</tr>
<tr>
<td>lam2</td>
<td>.1320636</td>
<td>.0896626</td>
<td>.003105</td>
<td>.1123817</td>
<td>.0282763  .3578873</td>
</tr>
</tbody>
</table>

Note: Adaptation tolerance is not met in at least one of the blocks.
file blassosplit_mcmc.dta saved
Bayesian lasso prediction

Compute posterior means of Bayesian lasso predictions for each observation in the test sample and save them in the variable `pmean`:

```
.bayespredict pmean if sample==2, mean rseed(16) dots
Computing predictions 10000 ........1000...........2000...........3000...........400
> 0........5000...........6000...........7000...........8000...........9000...........10
> 000 done
.gen double err_blasso = (y-pmean)^2
(221 missing values generated)
```

Compare mean squared prediction error for classical and Bayesian lassos:

```
.summarize err*

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>err_lasso</td>
<td>221</td>
<td>2875.555</td>
<td>3645.928</td>
<td>.3942694</td>
<td>20429.17</td>
</tr>
<tr>
<td>err_blasso</td>
<td>221</td>
<td>2854.459</td>
<td>3622.219</td>
<td>.2838339</td>
<td>19689.97</td>
</tr>
</tbody>
</table>
```
Credible intervals for Bayesian lasso predictions

Compute 95% credible intervals for Bayesian lasso predictions:

```
.bayespredict cri_l cri_u if sample==2, cri rseed(16) dots
Computing predictions 10000 ........1000 ........2000 ........3000 ........400
> 0 ........5000 ........6000 ........7000 ........8000 ........9000 ........10
> 000 done
.b list y yhat pmean cri* if sample==2 & id<10
```

<table>
<thead>
<tr>
<th>y</th>
<th>yhat</th>
<th>pmean</th>
<th>cri_l</th>
<th>cri_u</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td>141</td>
<td>171.31446</td>
<td>172.9158</td>
<td>61.21823</td>
</tr>
<tr>
<td>4.</td>
<td>206</td>
<td>155.51477</td>
<td>153.5821</td>
<td>39.70884</td>
</tr>
<tr>
<td>5.</td>
<td>135</td>
<td>130.15709</td>
<td>129.081</td>
<td>17.57684</td>
</tr>
<tr>
<td>8.</td>
<td>63</td>
<td>147.79128</td>
<td>144.6209</td>
<td>29.40267</td>
</tr>
</tbody>
</table>
New Bayesian features

Bayesian predictions

Bayesian lasso prediction

Prediction for test data

- Observed
- Posterior mean
- 95% credible interval
Multiple chains

- Bayesian inference uses MCMC.
- MCMC convergence must be established before any inferential conclusions can be made.
- MCMC convergence is often explored visually after the simulation.
- In Stata 16, you can run multiple chains using new option `nchains()` to explore convergence both visually and more formally.
- Instead of a single longer Markov chain, you can run several shorter chains to:
  - explore convergence from different initial states and potentially detect pseudoconvergence;
  - obtain more precise results; and
  - speed up computation when running the chains in parallel using multiple processors.
Gelman–Rubin convergence diagnostic

- With multiple chains, you can compute Gelman–Rubin convergence diagnostics for all parameters and use them for a more formal assessment of MCMC convergence.

- The Gelman–Rubin diagnostic $R_c$ (Brooks and Gelman 1998) summarizes the differences between multiple chains by comparing the within-chain and between-chains variances.

- An $R_c$ greater than 1.1 for any model parameter is considered to be indicative of nonconvergence.

- In addition to MCMC nonconvergence, poor sampling efficiency may also lead to large $R_c$.

- You can use `bayesstats grubin` to compute Gelman–Rubin diagnostics for all model parameters.
New Bayesian features

- Multiple chains
- Example: Convergence of Bayesian lasso

Example: Convergence of Bayesian lasso

Let’s run multiple chains to explore convergence of our earlier Bayesian lasso model. Here, we will fit Bayesian lasso using `bayes:` and simulate three chains. We will also use shorter chains of 3,500 iterations and display initial values used for each chain.

```
. bayes, prior({y:age sex bmi map tc ldl hdl tch ltg glu}, ///
  laplace(0, (sqrt({sigma2}/{lam2}))))) ///
> prior({sigma2}, jeffreys) ///
> prior({y:_cons}, normal(0, 1e6)) ///
> prior({lam2=1}, gamma(1, 1/1.78)) ///
> block({y:} {sigma2} {lam2}, split) ///
> rseed(16) dots ///
> nchains(3) initsummary mcmcsize(3500) ///
> : regress y age sex bmi map tc ldl hdl tch ltg glu
```

(output omitted)
Chain 1
  Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
  Simulation 3500 ........1000........2000........3000..... done

Chain 2
  Burn-in 2500 aaaaaaaaaaa1000aaaaaaaaaa2000aaaaa done
  Simulation 3500 ........1000........2000........3000..... done

Chain 3
  Burn-in 2500 aaaaaaaaaaaa1000aaaaaaaaaa2000aaaaa done
  Simulation 3500 ........1000........2000........3000..... done

Model summary

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

 Priors:
  {y:age sex bmi map tc ldl hdl tch ltg glu} ~ laplace(0,<expr1>) (1)
  {y:_cons} ~ normal(0,1e6) (1)
  {sigma2} ~ jeffreys

 Hyperprior:
  {lam2} ~ gamma(1,1/1.78)

 Expression:
  expr1 : sqrt({sigma2}/{lam2})

(1) Parameters are elements of the linear form xb_y.

(output omitted)
New Bayesian features

Multiple chains

Example: Convergence of Bayesian lasso

Initial values:
Chain 1:  {y:age} -10.0122 {y:sex} -239.819 {y:bmi} 519.84 {y:map} 324.39
{y:tc} -792.184 {y:ldl} 476.746 {y:hdl} 101.045 {y:tch} 177.064 {y:ltg} 751.279
{y:glu} 67.6254 {y:_cons} 152.133 {sigma2} 2932.68 {lam2} 1

Chain 2:  {y:age} .856616 {y:sex} .141924 {y:bmi} -.210244 {y:map} -.84781
{y:tc} -3.11354 {y:ldl} .287661 {y:hdl} .007601 {y:tch} -1.11456 {y:ltg}
-1.02858 {y:glu} -.775863 {y:_cons} 428.914 {sigma2} 2943.11 {lam2} .181865

Chain 3:  {y:age} -1.69075 {y:sex} -.39084 {y:bmi} .074689 {y:map} -.371372
{y:tc} -.196243 {y:ldl} -1.50958 {y:hdl} .899462 {y:tch} -.265409 {y:ltg}
-1.53079 {y:glu} -.387231 {y:_cons} -1676.45 {sigma2} 2906.83 {lam2} .766684

(output omitted)
Bayesian linear regression

Random-walk Metropolis-Hastings sampling

<table>
<thead>
<tr>
<th>Per MCMC chain:</th>
<th>Iterations = 6,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Burn-in</td>
<td>2,500</td>
</tr>
<tr>
<td>Sample size</td>
<td>3,500</td>
</tr>
<tr>
<td>Number of obs</td>
<td>442</td>
</tr>
<tr>
<td>Avg acceptance rate</td>
<td>.4401</td>
</tr>
<tr>
<td>Avg efficiency: min = .01631</td>
<td></td>
</tr>
<tr>
<td>avg = .1081</td>
<td></td>
</tr>
<tr>
<td>max = .2282</td>
<td></td>
</tr>
</tbody>
</table>

Avg log marginal-likelihood = -2416.1455
Max Gelman-Rubin Rc = 1.06

<table>
<thead>
<tr>
<th>Mean</th>
<th>Std. Dev.</th>
<th>MCSE</th>
<th>Median</th>
<th>[95% Cred. Interval]</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>-2.217509</td>
<td>53.09013</td>
<td>1.2185</td>
<td>-2.023093</td>
</tr>
<tr>
<td>sex</td>
<td>-205.5805</td>
<td>62.72992</td>
<td>1.61611</td>
<td>-207.3534</td>
</tr>
<tr>
<td>bmi</td>
<td>520.648</td>
<td>66.46239</td>
<td>1.75237</td>
<td>519.7247</td>
</tr>
<tr>
<td>map</td>
<td>302.2608</td>
<td>64.14865</td>
<td>1.60709</td>
<td>305.1601</td>
</tr>
<tr>
<td>tc</td>
<td>-154.7892</td>
<td>156.5725</td>
<td>11.9657</td>
<td>-141.3851</td>
</tr>
<tr>
<td>hdl</td>
<td>-163.6772</td>
<td>105.5633</td>
<td>6.6184</td>
<td>-165.3641</td>
</tr>
<tr>
<td>tch</td>
<td>92.444</td>
<td>111.6599</td>
<td>5.78541</td>
<td>85.0718</td>
</tr>
<tr>
<td>ltg</td>
<td>510.006</td>
<td>93.71682</td>
<td>5.14455</td>
<td>506.4445</td>
</tr>
<tr>
<td>glu</td>
<td>66.79432</td>
<td>60.65255</td>
<td>1.59444</td>
<td>65.17973</td>
</tr>
<tr>
<td>_cons</td>
<td>152.1796</td>
<td>2.629443</td>
<td>0.053714</td>
<td>152.1549</td>
</tr>
<tr>
<td>sigma2</td>
<td>2961.205</td>
<td>212.3299</td>
<td>4.70581</td>
<td>2948.655</td>
</tr>
<tr>
<td>lam2</td>
<td>.0919228</td>
<td>.0572705</td>
<td>.001712</td>
<td>.0785825</td>
</tr>
</tbody>
</table>

Note: Default initial values are used for multiple chains.
Graphical diagnostics for multiple chains

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

- **Trace**
  - Iteration number vs trace of values

- **Histogram**
  - Distribution of values

- **Autocorrelation**
  - Lag vs autocorrelation of values

- **Density**
  - Distribution density of values

Chains: 1/3

Yulia Marchenko (StataCorp)
## Summary for each chain

```
. bayesstats summary {y:bmi}, sepchains
Posterior summary statistics

<table>
<thead>
<tr>
<th>Chain 1</th>
<th>MCMC sample size = 3,500</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>Mean</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>bmi</td>
<td>519.3121</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chain 2</th>
<th>MCMC sample size = 3,500</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>Mean</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>bmi</td>
<td>522.8732</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chain 3</th>
<th>MCMC sample size = 3,500</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>Mean</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>bmi</td>
<td>519.7587</td>
</tr>
</tbody>
</table>
```
New Bayesian features

- Multiple chains
- Example: Convergence of Bayesian lasso

## Gelman–Rubin statistics

```
.bayes, nomodelsummary notable
Bayesian linear regression
Random-walk Metropolis-Hastings sampling
Number of chains = 3
Per MCMC chain:
  Iterations = 6,000
  Burn-in = 2,500
  Sample size = 3,500
Number of obs = 442
Avg acceptance rate = .4401
Avg efficiency:  min = .01631
                 avg = .1081
                 max = .2282
Avg log marginal-likelihood = -2416.1455
Max Gelman-Rubin Rc = 1.06
```

Maximum Gelman–Rubin $R_c = 1.06 < 1.1$. 
Given the maximum $R_c$ of 1.06, all other model parameters will also have $R_c$ values less than 1.1:

```
.bayesstats grubin, sort
Gelman-Rubin convergence diagnostic
Number of chains = 3
MCMC size, per chain = 3,500
Max Gelman-Rubin Rc = 1.059548

<table>
<thead>
<tr>
<th></th>
<th>Rc</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td></td>
</tr>
<tr>
<td>ldl</td>
<td>1.059548</td>
</tr>
<tr>
<td>tc</td>
<td>1.043915</td>
</tr>
<tr>
<td>ltg</td>
<td>1.017272</td>
</tr>
<tr>
<td>tch</td>
<td>1.016441</td>
</tr>
<tr>
<td>hdl</td>
<td>1.014299</td>
</tr>
<tr>
<td>map</td>
<td>1.002838</td>
</tr>
<tr>
<td>glu</td>
<td>1.001849</td>
</tr>
<tr>
<td>lam2</td>
<td>1.001789</td>
</tr>
</tbody>
</table>

|    |      |
| y  |      |
| age| 1.001506 |
| sex| 1.001356 |
| _cons| 1.000939 |
| bmi| 1.000795 |
| sigma2| 1.000684 |
```

Convergence rule: $R_c < 1.1$
Based on the Gelman–Rubin statistics and visual diagnostics, it is reasonable to assume that MCMC converged in our example. For an example of MCMC nonconvergence, see, for instance, https://www.stata.com/new-in-stata/gelman-rubin-convergence-diagnostic/
Clean up

Remove the generated files if you no longer need them:

```
. erase blasso_mcmc.dta
. erase blasso_pred.dta
. erase blasso_pred.ster
. erase blassosplit_mcmc.dta
```
Summary

- Bayesian prediction is a powerful tool not only for predicting future observations but also for model checking.
- It provides an entire distribution for each predicted observation, which allows you to assess the uncertainty about the estimated predicted values.
- Use `bayespredict` to compute various Bayesian predictions.
- Use `bayesreps` and `bayesstats ppvalues` to perform posterior model checks.
- Use new option `nchains()` with `bayesmh` and `bayes:` to simulate multiple chains.
- Use unofficial command `bayesparallel` to generate multiple chains simultaneously.
- Use `bayesstats grubin` to compute the Gelman–Rubin convergence diagnostics for all model parameters.
- Revisit section New Bayesian features in a nutshell for details.
Quick overview of new Bayesian features in Stata 16:

Bayesian predictions: [BAYES] bayespredict and

Multiple chains:

Running multiple chains in parallel:

Gelman–Rubin convergence diagnostic: [BAYES] bayesstats grubin and
https://www.stata.com/new-in-stata/gelman-rubin-convergence-diagnostic/
Additional resources (cont.)

- Overview of Bayesian features:
  https://www.stata.com/features/overview/bayesian-analysis/
  https://www.stata.com/features/bayesian-analysis/

- Stata Bayesian Analysis Reference Manual:

- YouTube: Bayesian analysis in Stata
  https://www.stata.com/links/video-tutorials
  https://www.youtube.com/playlist?list=PLN5IskQdgXWnvvLNIeGpL2u1Jg739jsqd
References

