New Bayesian features: Predictions, multiple chains, and more

Yulia Marchenko

StataCorp LLC

2020 London Stata Conference

Outline

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

New Bayesian features in a nutshell

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, `bayesmh` and `bayes:` automatically compute and report the maximum Gelman–Rubin statistic across model parameters.

Type

```
. bayesstats grubin
```

to obtain the Gelman–Rubin diagnostic for each model parameter.

Bayesian predictions. Use `bayespredict` to compute various Bayesian predictions and their posterior summaries.

- Compute and save simulated outcomes, their expected values, and residuals in a new dataset:

  ```
  . bayespredict {_ysim} {_mu} {_resid}, saving(filename)
  ```

- Or compute posterior summaries of simulated outcomes and save them in a new variable in the current dataset:

  ```
  . bayespredict pmean, mean
  ```

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

- Summarize predicted quantities as any other model parameter:

  ```
  . 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

Command	*Description*
Estimation	
`bayes:`	Bayesian regression models (with multiple chains in Stata 16)
`bayesmh`	General Bayesian models using MH (with multiple chains in Stata 16)
`bayesmh` *evaluators*	User-defined Bayesian models using MH
Postestimation	
`bayesgraph`	Graphical convergence diagnostics
`bayesstats ess`	Effective sample sizes and more
`bayesstats summary`	Summary statistics
`bayesstats ic`	Information criteria and Bayes factors
`bayesstats ppvalues`	Posterior predictive *p*-values New in Stata 16
`bayestest model`	Model posterior probabilities
`bayestest interval`	Interval hypothesis testing
`bayespredict`	Bayesian predictions New in Stata 16
`bayesreps`	MCMC replicates New in Stata 16

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 $\boldsymbol{\theta}$ 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 $\boldsymbol{\theta}$ formulated as a **prior distribution** $p(\boldsymbol{\theta})$.
- After data D are observed, the information about $\boldsymbol{\theta}$ is updated based on the **likelihood** $f(D|\boldsymbol{\theta})$.
- Information is updated by using the Bayes rule to form a **posterior distribution** $p(\boldsymbol{\theta}|D)$:

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

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

Inference

- Estimating a posterior distribution $p(\boldsymbol{\theta}|D)$ 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

```
. use diabetes_std
(Diabetes data from Efron et al. (2004) with standardized covariates)
. describe
Contains data from diabetes_std.dta
  obs:           442                          Diabetes data from Efron et al.
                                                (2004) with standardized
                                                covariates
 vars:            12                          9 Sep 2020 17:11
                                               (_dta has notes)
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
y               int     %8.0g                 Measure of disease progression
age             float   %9.0g                 Age
sex             float   %9.0g                 Sex
bmi             float   %9.0g                 Body mass index
map             float   %9.0g                 Mean arterial pressure
tc              float   %9.0g                 Total cholesterol
ldl             float   %9.0g                 LDL cholesterol level
hdl             float   %9.0g                 HDL cholesterol level
tch             float   %9.0g                 TCh blood serum level
ltg             float   %9.0g                 LTG blood serum level
glu             float   %9.0g                 Glucose blood serum level
id              int     %9.0g               * Subject ID
                                            * 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

```
. 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*)

```
Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
Simulation 10000 .........1000.........2000.........3000.........4000.........5
> 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

MCMC iterations	12,500
Burn-in	2,500
MCMC sample size	10,000
Number of obs	442
Acceptance rate	.4379
Efficiency: min	.0152
avg	.1025
max	.2299

Log marginal-likelihood = -2415.7171

	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
y						
age	-2.478525	52.97851	1.26623	-2.593401	-108.3303	104.2442
sex	-209.4461	61.21979	1.70006	-211.1479	-330.2515	-85.52568
bmi	522.1367	66.76557	1.8115	520.6348	393.4224	656.4993
map	304.1617	65.26244	1.77912	306.1749	175.0365	432.3554
tc	-172.2847	157.5097	12.7739	-159.4816	-523.0447	110.226
ldl	1.304382	128.3598	9.96343	-7.796492	-251.2571	298.4382
hdl	-158.8146	112.6562	6.82563	-158.1347	-378.4126	48.93263
tch	91.27437	111.8483	6.06667	86.32462	-114.675	319.0824
ltg	515.5167	94.06607	5.83902	509.9952	342.9893	715.739
glu	67.94583	62.86024	1.69235	66.11433	-51.1174	197.7894
_cons	152.0964	2.545592	.053095	152.0963	146.9166	157.1342
sigma2	2961.246	207.0183	4.79372	2949.282	2587.023	3389.206
lam2	.0889046	.055257	.001899	.0769573	.020454	.229755

Note: Adaptation tolerance is not met in at least one of the blocks.

Objective 1—Important predictors:

```
. bayesstats summary (age:{y: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
```

	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred. Interval]	
age	.5277	.4992571	.011353	1	0	1
sex	.9997	.0173188	.000224	1	1	1
bmi	0	0	0	0	0	0
map	0	0	0	0	0	0
tc	.8815	.3232154	.018971	1	0	1
ldl	.5301	.4991181	.029122	1	0	1
hdl	.9226	.2672384	.01198	1	0	1
tch	.1992	.3994187	.016507	0	0	1
ltg	0	0	0	0	0	0
glu	.1417	.3487596	.008577	0	0	1

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

What are Bayesian predictions?

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)

Posterior predictive distribution (PPD) for a new outcome value y^{new} given observed data $\mathbf{y}$:

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

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

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 $p(\boldsymbol{\theta}|\mathbf{y})$, PPD $p(y^{new}|\mathbf{y})$ usually does not have a closed form and must be approximated. Formula

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

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

1. Simulate $\boldsymbol{\theta}^t$ from $p(\boldsymbol{\theta}|\mathbf{y})$
2. Simulate y^t from $f(y^{new}|\boldsymbol{\theta}^t)$
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 $p(y^{new}|\mathbf{y})$. Unlike classical predictions, we will have a sample of T values for each (new) observation.

An MCMC sample $\{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^T\}$ 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:

```
. 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
```

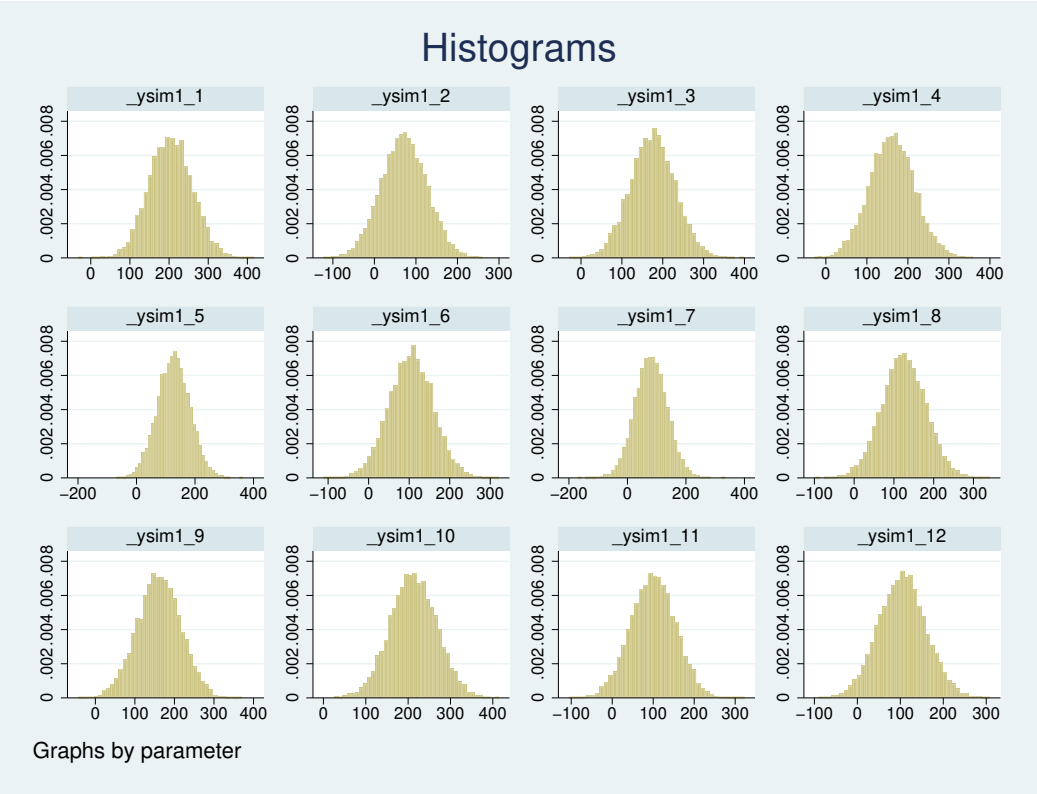
variable name	storage type	display format	value label	variable label
_chain	int	%8.0g		Chain identifier
_index	int	%8.0g		Iteration number
_ysim1_1	double	%10.0g		Simulated y, obs. #1
_ysim1_2	double	%10.0g		Simulated y, obs. #2
		(*output omitted*)		
_ysim1_441	double	%10.0g		Simulated y, obs. #441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

```
Sorted by:
```

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
```

	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
_ysim1_1	203.7014	54.97776	.558382	203.1887	98.05673	312.5681
_ysim1_2	71.25238	55.05362	.550536	70.95502	-35.91061	179.3321
_ysim1_3	175.3088	55.09297	.55093	175.3602	67.95822	284.0823
_ysim1_4	161.6022	55.3058	.577559	161.149	53.56015	273.678
_ysim1_5	127.0087	55.34909	.553491	127.128	18.79625	235.7958
_ysim1_6	104.5405	54.35416	.543542	105.0999	-2.76073	211.2213
_ysim1_7	80.33914	55.64711	.55933	80.04908	-28.58423	189.9953
_ysim1_8	124.9409	55.43717	.554372	124.5975	16.57208	234.4114
_ysim1_9	160.7804	55.37094	.561705	160.6604	51.05089	269.877
_ysim1_10	212.4139	54.70172	.554331	211.9675	105.3659	319.5306
_ysim1_11	99.4205	54.82602	.574513	99.69542	-9.512793	203.8927
_ysim1_12	104.5963	55.64342	.588025	104.4855	-4.058576	215.4143

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)
Interval tests      MCMC sample size =     10,000
       prob1 : {_ysim[1]} > 100
```

	Mean	Std. Dev.	MCSE
prob1	.9733	0.16121	.0016834

Posterior predictive checks

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

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 $T(\mathbf{y})$ such as a sample mean or median. PPP for $T(\mathbf{y})$ is defined as

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

- In a Bayesian context, $T(\mathbf{y}) = T(\mathbf{y}, \boldsymbol{\theta})$ may also depend on model parameters $\boldsymbol{\theta}$ 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.

PPPs for mean and variance

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

T	Mean	Std. Dev.	E(T_obs)	P(T>=T_obs)
mean	152.0664	3.635561	152.1335	.4969
var	5934.96	487.52	5943.331	.4808

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

PPPs for other statistics

Define Mata function `skew()` that computes skewness:

```
. mata:
------------------------------------------------- mata (type end to exit) -----
: 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
```

T	Mean	Std. Dev.	E(T_obs)	P(T>=T_obs)
skewness	.0899553	.1045585	.4390664	.0002
min	-62.72501	24.93968	25	0
max	381.8695	26.75575	346	.9319

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
```

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

```
* 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)
```

Bayesian lasso

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 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
Simulation 10000 .........1000.........2000.........3000.........4000.........5
> 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

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

	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
y						
age	22.90211	70.97209	1.65398	18.49825	-109.8378	167.3328
sex	-147.1624	91.8814	2.60312	-144.2808	-335.182	20.59289
bmi	523.9505	99.27119	2.74922	526.4859	326.6205	724.9461
map	279.1178	100.3706	2.87727	280.5395	73.24508	477.0849
tc	-10.10333	150.7421	9.50814	-7.539645	-329.4431	290.428
ldl	-39.61316	130.3894	7.07092	-29.77871	-317.1814	205.3837
hdl	-149.4535	136.8265	6.65266	-145.7912	-437.285	108.0459
tch	161.9482	155.7781	7.82029	147.5103	-115.6871	500.1832
ltg	312.8631	124.2238	5.34773	315.6055	72.72891	559.7625
glu	24.37885	79.45832	1.99475	20.4382	-125.1158	191.8139
_cons	149.7803	3.844837	.078828	149.7844	141.9715	157.2013
sigma2	3310.895	329.993	8.00504	3285.107	2733.671	4015.07
lam2	.1320636	.0896626	.003105	.1123817	.0282763	.3578873

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*
```

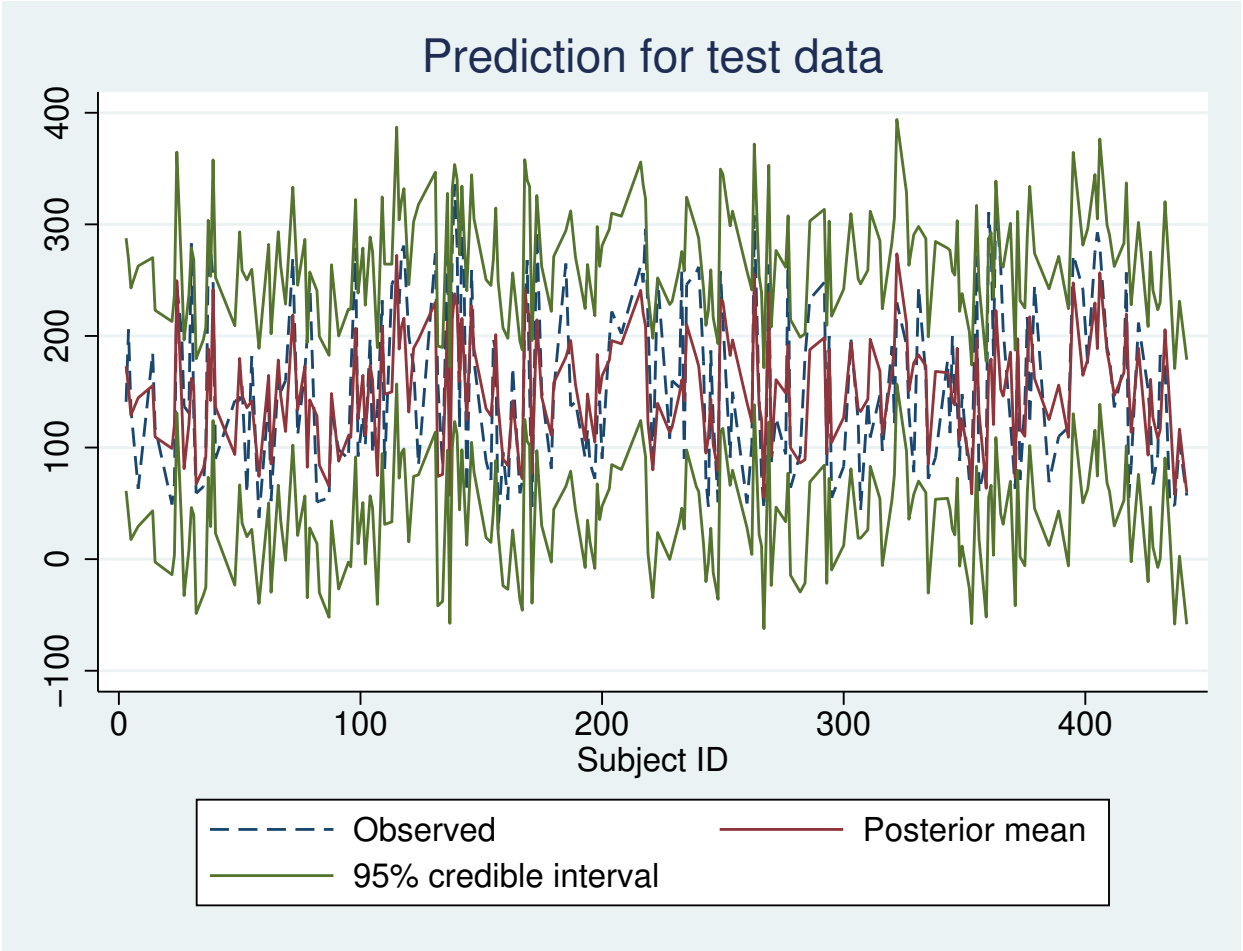
Variable	Obs	Mean	Std. Dev.	Min	Max
err_lasso	221	2875.555	3645.928	.3942694	20429.17
err_blasso	221	2854.459	3622.219	.2838339	19689.97

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
. list y yhat pmean cri* if sample==2 & id<10
```

	y	yhat	pmean	cri_l	cri_u
3.	141	171.31446	172.9158	61.21823	287.624
4.	206	155.51477	153.5821	39.70884	268.9385
5.	135	130.15709	129.081	17.57684	243.0658
8.	63	147.79128	144.6209	29.40267	262.7248


Prediction for test data
−100
0
100
200
300
400
0
100
200
300
400
Subject ID
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.

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 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
  Simulation 3500 .........1000.........2000.........3000..... done

Chain 3
  Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa 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*)

```
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*)

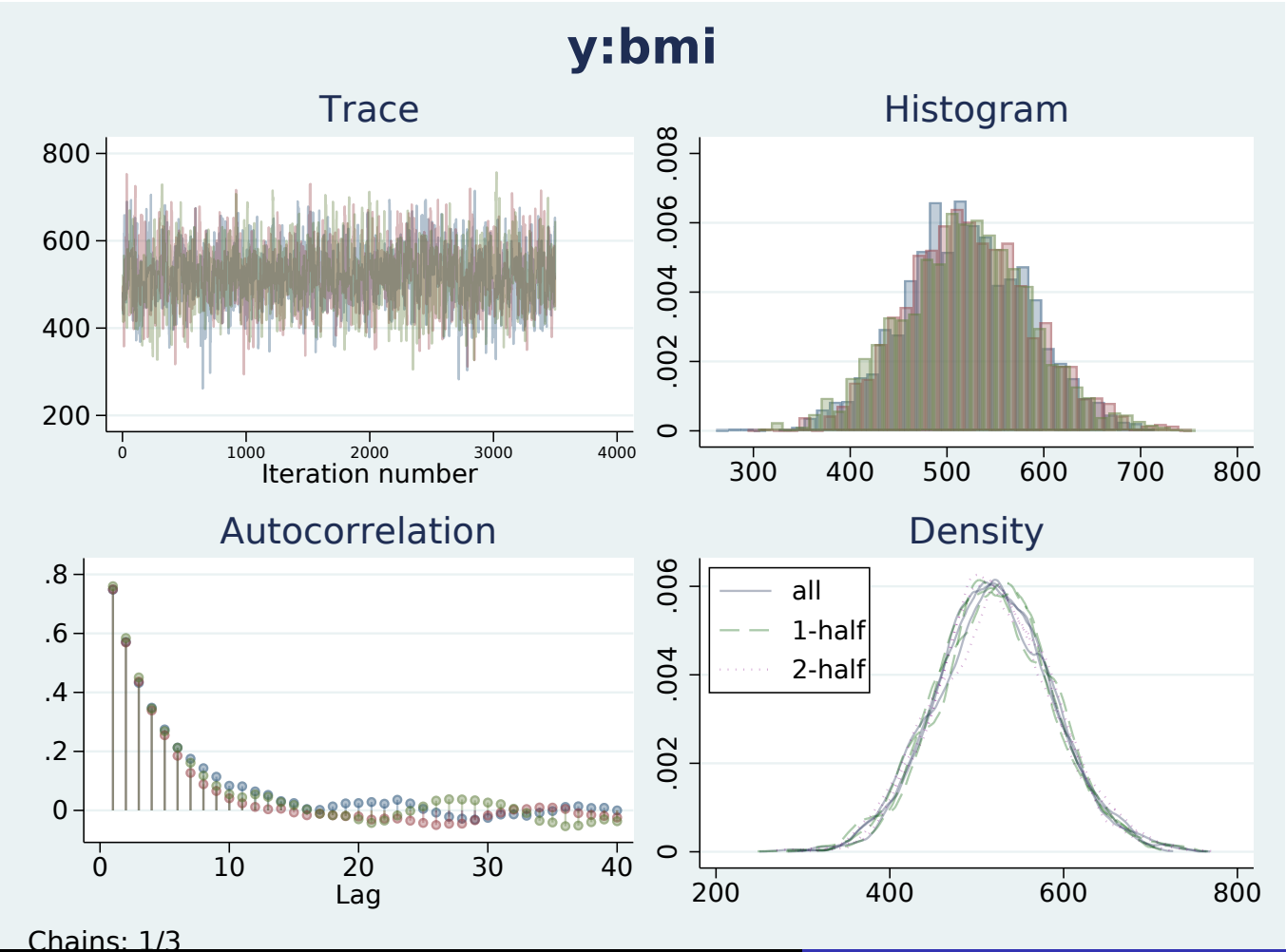
```
Bayesian linear regression                      Number of chains    =          3
Random-walk Metropolis-Hastings sampling        Per MCMC chain:
                                                     Iterations     =      6,000
                                                     Burn-in        =      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
```

	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
y						
age	-2.217509	53.09013	1.2185	-2.023093	-107.2715	102.7923
sex	-205.5805	62.72992	1.61611	-207.3534	-329.1627	-83.28759
bmi	520.648	66.46239	1.75237	519.7247	393.3882	656.7785
map	302.2608	64.14865	1.60709	305.1601	174.4888	426.939
tc	-154.7892	156.5725	11.9657	-141.3851	-485.8309	114.466
ldl	-14.18946	130.7579	9.50712	-20.58282	-264.0704	265.0667
hdl	-163.6772	105.5633	6.6184	-165.3641	-362.3713	40.72074
tch	92.444	111.6599	5.78541	85.0718	-109.9963	331.8837
ltg	510.006	93.71682	5.14455	506.4445	341.4745	707.1959
glu	66.79432	60.65255	1.59444	65.17973	-44.24351	192.3846
_cons	152.1796	2.629443	.053714	152.1549	146.8562	157.2639
sigma2	2961.205	212.3299	4.70581	2948.655	2576.092	3397.092
lam2	.0919228	.0572705	.001712	.0785825	.0213069	.2371047

Note: Default initial values are used for multiple chains.

Graphical diagnostics for multiple chains

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



Summary for each chain

```
. bayesstats summary {y:bmi}, sepchains
Posterior summary statistics
Chain 1                                          MCMC sample size =     3,500
```

y	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred. Interval]	
bmi	519.3121	65.38551	3.07014	517.8948	387.1325	648.7197

```
Chain 2                                          MCMC sample size =     3,500
```

y	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred. Interval]	
bmi	522.8732	65.67219	2.89163	520.5146	400.4887	658.6483

```
Chain 3                                          MCMC sample size =     3,500
```

y	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred. Interval]	
bmi	519.7587	68.23592	3.15049	520.5898	384.0506	657.4678

Gelman–Rubin statistics

```
. bayes, nomodelsummary notable
Bayesian linear regression                      Number of chains    =          3
Random-walk Metropolis-Hastings sampling        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
```

	Rc
y	
ldl	1.059548
tc	1.043915
ltg	1.017272
tch	1.016441
hdl	1.014299
map	1.002838
glu	1.001849
lam2	1.001789
y	
age	1.001506
sex	1.001356
_cons	1.000939
bmi	1.000795
sigma2	1.000684

```
Convergence rule: Rc < 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.

Additional resources

- Quick overview of new Bayesian features in Stata 16:
  https://www.stata.com/new-in-stata/new-in-bayesian-analysis/
- Bayesian predictions: [BAYES] **bayespredict** and
  https://www.stata.com/new-in-stata/bayesian-predictions/
- Multiple chains:
  https://www.stata.com/new-in-stata/multiple-chains-in-bayesian-estimation/
- Running multiple chains in parallel:
  https://www.stata.com/support/faqs/statistics/bayesian-analysis-parallel-multiple-chains/
- 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:

  https://www.stata.com/manuals/bayes.pdf
- YouTube: Bayesian analysis in Stata

  https://www.stata.com/links/video-tutorials

  https://www.youtube.com/playlist?list=PLN5IskQdgXWnvvLNIeGpL2u1Jg739jsqd

References

Brooks, S. P., and A. Gelman. 1998. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics* 7: 434–455.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani 2004. Least angle regression. *The Annals of Statistics* 32: 407–499.

Park, T.,and G. Casella. 2008. Bayesian lasso. *Journal of the American Statistical Association* 103: 681–686.