Bayesian multilevel modeling using Stata

Yulia Marchenko

Vice President, Statistics and Data Science
StataCorp LLC

2022 UK Stata Conference

What's new in Bayesian multilevel modeling in Stata 17?

- New flexible and powerful multilevel syntax in `bayesmh` allows you to fit:
- nonlinear multilevel models;
- SEM-type models;
- joint longitudinal and survival models; and, more generally,
- multivariate (multiple-equation) linear and nonlinear multilevel models.
- New multivariate normal prior distributions for random effects with specialized covariance matrices such as exchangeable and identity
- Exchangeable covariance structures, `covariance(exchangeable)`, are now supported with `bayes:mixed`.
- Gibbs sampling for normally-distributed random effects with multilevel models with normal error terms.

What is a Bayesian multilevel model?

- Multilevel models are regression models that incorporate group-specific effects at different levels of hierarchy.
- Group-specific effects at different hierarchical levels may be nested or crossed.
- Group-specific effects are assumed to vary randomly across groups according to some a priori distribution, commonly a normal distribution.
- This assumption makes multilevel models natural candidates for Bayesian analysis.
- Bayesian multilevel models additionally assume that other model parameters such as regression coefficients and variance components—variances of group-specific effects—are also random.

Why Bayesian multilevel models?

- You might want to use Bayesian analysis:
  - to incorporate external prior information;
  - when it is more natural to express a research objective using probability statements such as how likely a product is to fail under warranty.
  - to compute an actual probability for a hypothesis of interest;
  - and more.
- In addition to standard reasons for Bayesian analysis, Bayesian multilevel modeling is often used when the number of groups is small or in the presence of many hierarchical levels.
- Various Bayesian information criteria are popular for comparing multilevel models.
- When the comparison of groups is of main interest, Bayesian multilevel modeling can provide entire distributions of group-specific effects.
- Also, variances of group-specific effects incorporate the uncertainty about all estimated model parameters!

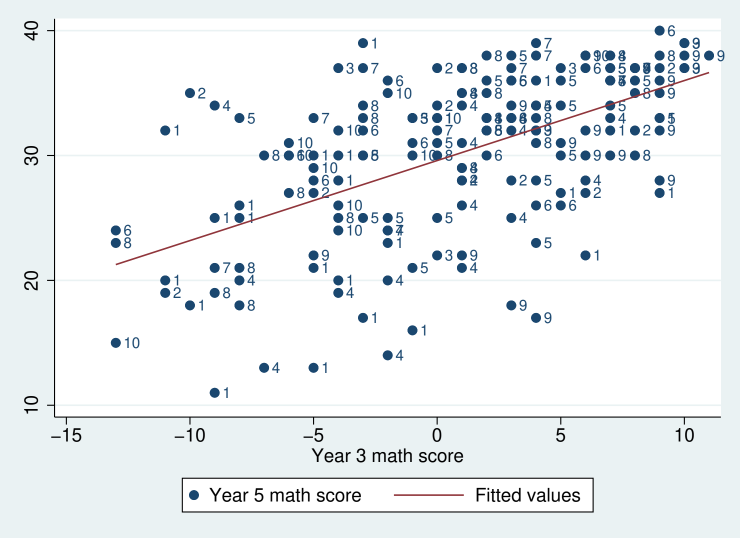
School data

- Consider data from Mortimore et al. (1988) on 887 math scores of pupils in the third and fifth years from 48 different schools in inner London.

```
. webuse mathscores

. describe

Contains data from https://www.stata-press.com/data/r17/mathscores.dta
 Observations:           887
    Variables:             3                  9 May 2020 23:31
-------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------------
school          float   %9.0g                 School ID
math3           float   %9.0g                 Year 3 math score
math5           float   %9.0g                 Year 5 math score
-------------------------------------------------------------------------------
Sorted by:
```

- Let's examine the first 10 schools:

```
. sort school math3

. graph twoway (scatter math5 math3, mlabel(school)) (lfit math5 math3) if school<=1
```


Year 3 math score
Year 5 math score
Fitted values
10
20
30
40
–15
–10
–5
0
5
10

Linear random-intercept model

- Suppose we are interested in estimating school-specific effects.
- We can fit a linear random-intercept model:

$$
\begin{aligned}
\texttt{math5}_{ij} &= \beta_0 + u_{0j} + \beta_1 \texttt{math3}_{ij} + \epsilon_{ij} \\
\epsilon_{ij} &\sim \text{Normal}(0, \sigma^2) \\
u_{0j} &\sim \text{Normal}(0, \sigma_0^2)
\end{aligned}
$$

  for $j = 1, ..., 48$ schools and $i = 1, ..., n_j$ pupils in school j.
- u_{0j} is a random effect (intercept) at the school level, and represents an upward or downward shift in performance from the overall regression line.
- σ^2 represents the within-school variability and σ_0^2—the between-school variability.

- Let's first use `mixed` to fit this model:

```
. mixed math5 math3 || school:

Mixed-effects ML regression                     Number of obs     =        887
Group variable: school                          Number of groups  =         48
                                                Obs per group:
                                                              min =          5
                                                              avg =       18.5
                                                              max =         62
                                                Wald chi2(1)      =     347.92
Log likelihood = -2767.8923                     Prob > chi2       =     0.0000

------------------------------------------------------------------------------
       math5 | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
       math3 |   .6088066   .0326392    18.65   0.000     .5448349    .6727783
       _cons |   30.36495   .3491544    86.97   0.000     29.68062    31.04928
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects parameters  |   Estimate   Std. err.     [95% conf. interval]
-----------------------------+------------------------------------------------
school: Identity             |
                  var(_cons) |   4.026853   1.189895      2.256545    7.186004
-----------------------------+------------------------------------------------
               var(Residual) |   28.12721    1.37289       25.5611    30.95094
------------------------------------------------------------------------------
LR test vs. linear model: chibar2(01) = 56.38         Prob >= chibar2 = 0.0000
```

- We can predict the random (school) effects and their standard errors after fitting `mixed`:

```
. predict re, reffects reses(se)
. sort school
. list school re se if school!=school[_n+1] & school<10
```

	school	re	se
25.	1	-2.676116	.9377579
35.	2	-.0152072	1.286861
43.	3	1.058414	1.370049
67.	4	-2.122366	.9527702
92.	5	-.0924746	.9377579
105.	6	.6523949	1.186348
115.	7	1.536003	1.286861
141.	8	.4360111	.9234335
162.	9	-1.988043	1.002539

- In the above, we listed only the first nine schools.
- The reported random-effects standard errors are conditional on the estimated regression coefficients and variance components.

Bayesian random-intercept model

- To fit a Bayesian random-intercept model, we need to formulate prior distributions in addition to the likelihood model.
- Let's consider the following prior distributions:

$$
\begin{aligned}
\beta_i &\sim \text{Normal}(0{,}10000),\ i = 0, 1 \\
\sigma^2 &\sim \text{InvGamma}(0.01, 0.01) \\
\sigma_0^2 &\sim \text{InvGamma}(0.01, 0.01)
\end{aligned}
$$

- To fit the model, we simply prefix `mixed` with `bayes:`.

```
. set seed 12345

. bayes, melabel: mixed math5 math3 || school:
note: Gibbs sampling is used for regression coefficients and variance
      components.

Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
Simulation 10000 .........1000.........2000.........3000.........4000.........5
> 000.........6000.........7000.........8000.........9000.........10000 done

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 =         48
                                                 Obs per group:
                                                              min =          5
                                                              avg =       18.5
                                                              max =         62
                                                 Number of obs    =        887
                                                 Acceptance rate  =      .8102
                                                 Efficiency:  min =     .03923
                                                              avg =      .3628
Log marginal-likelihood                                       max =      .7226

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
math5        |
       math3 |  .6081648   .0327236   .000428   .6077114   .5450497   .6728088
       _cons |  30.36912   .3570318   .018026   30.36653   29.67139   31.08675
-------------+----------------------------------------------------------------
school       |
   var(_cons)|  4.314708   1.337713   .041102   4.146655   2.215281   7.493345
-------------+----------------------------------------------------------------
var(Residual)|  28.26249   1.387124   .016318   28.22386   25.67543   31.11543
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
```

- We used option `melabel` to obtain output similar to that of `mixed` for easier comparison of the results.
- The reported estimates of posterior means and posterior standard deviations for model parameters are similar to the corresponding MLEs and standard errors reported by `mixed`.

- Here is the output from `bayes:mixed` without the `melabel` option.

```
. bayes
Multilevel structure
------------------------------------------------------------------------------
school
    {U0}: random intercepts
------------------------------------------------------------------------------

Model summary
------------------------------------------------------------------------------
Likelihood:
  math5 ~ normal(xb_math5,{e.math5:sigma2})

Priors:
  {math5:math3 _cons} ~ normal(0,10000)                                   (1)
                 {U0} ~ normal(0,{U0:sigma2})                             (1)
     {e.math5:sigma2} ~ igamma(.01,.01)

Hyperprior:
  {U0:sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_math5.
```

```
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 =         48
                                                 Obs per group:
                                                              min =          5
                                                              avg =       18.5
                                                              max =         62
                                                 Number of obs    =        887
                                                 Acceptance rate  =      .8102
                                                 Efficiency:  min =     .03923
                                                              avg =      .3628
Log marginal-likelihood                                       max =      .7226
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
math5						
math3	.6081648	.0327236	.000428	.6077114	.5450497	.6728088
_cons	30.36912	.3570318	.018026	30.36653	29.67139	31.08675
school						
U0:sigma2	4.314708	1.337713	.041102	4.146655	2.215281	7.493345
e.math5						
sigma2	28.26249	1.387124	.016318	28.22386	25.67543	31.11543

Note: Default priors are used for model parameters.

- Let's describe it in pieces.

```
. bayes
Multilevel structure
------------------------------------------------------------------------------
school
    {U0}: random intercepts
------------------------------------------------------------------------------

Model summary
------------------------------------------------------------------------------
Likelihood:
  math5 ~ normal(xb_math5,{e.math5:sigma2})
Priors:
  {math5:math3 _cons} ~ normal(0,10000)                                    (1)
                 {U0} ~ normal(0,{U0:sigma2})                              (1)
     {e.math5:sigma2} ~ igamma(.01,.01)
Hyperprior:
  {U0:sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_math5.
```

- The header includes additional information about the fitted Bayesian model.
- Parameter `{U0}` represents random intercepts in the model.
- Regression coefficients `{math5:math3}` and `{math5:_cons}` are assigned default normal priors with zero means and variances of 10,000.
- The variance component for schools `{U0:sigma2}` and error variance `{e.math5:sigma2}` are assigned default inverse-gamma priors with 0.01 for both the shape and scale parameters.

- The rest of the header is the same as with option `melabel`.

```
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 =         48
                                                 Obs per group:
                                                              min =          5
                                                              avg =       18.5
                                                              max =         62
                                                 Number of obs    =        887
                                                 Acceptance rate  =      .8102
                                                 Efficiency:  min =     .03923
                                                              avg =      .3628
Log marginal-likelihood                                       max =      .7226
```

- In the output table, the results are the same, but the parameter labels are different.

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
math5						
math3	.6081648	.0327236	.000428	.6077114	.5450497	.6728088
_cons	30.36912	.3570318	.018026	30.36653	29.67139	31.08675
school						
U0:sigma2	4.314708	1.337713	.041102	4.146655	2.215281	7.493345
e.math5						
sigma2	28.26249	1.387124	.016318	28.22386	25.67543	31.11543

Note: Default priors are used for model parameters.

- Without option `melabel`, `bayes:mixed` displays results using parameter names as you would use when referring to these parameters in `bayes`'s options or during postestimation.
- For example, you would use `{U0:sigma2}` to refer to the variance component for schools and `{e.math5:sigma2}` to refer to the error variance.

Random-effects parameters

- The term *random effects*, representing subject-specific effects, is not well suited for Bayesian multilevel models, because all effects or parameters are considered random within the Bayesian framework. But we will use it for consistency with classical multilevel models.
- Unlike frequentist multilevel models, Bayesian multilevel models do not integrate "random effects" out but estimate them together with other model parameters.
- Thus, random effects are treated as model parameters just like regression coefficients and variance components.
- The `bayes` prefix does not report them by default because there are often too many of them.
- But you can display them during or after estimation.

- Let's replay the estimation, adding option `showreffects()` to display the estimates of the first nine random intercepts.

```
. bayes, showreffects({U0[1/9]})
```

(header omitted)

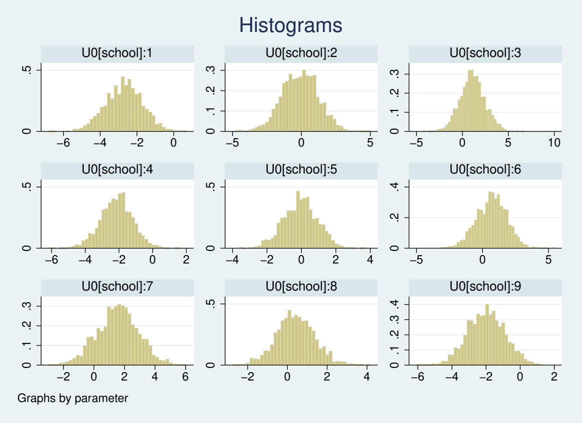
	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
math5						
math3	.6081648	.0327236	.000428	.6077114	.5450497	.6728088
_cons	30.36912	.3570318	.018026	30.36653	29.67139	31.08675
U0[school]						
1	-2.701971	1.007584	.033658	-2.688302	-4.677831	-.7650245
2	-.0929602	1.323015	.031586	-.0791454	-2.6954	2.444911
3	1.083833	1.367884	.03335	1.078927	-1.577482	3.887134
4	-2.143734	.9656367	.029863	-2.107032	-4.124594	-.2722371
5	-.0531935	1.00176	.031622	-.0485564	-2.078467	1.801151
6	.69567	1.178609	.030849	.7392095	-1.686727	2.894298
7	1.552796	1.352609	.033462	1.595416	-1.145189	4.14544
8	.4141391	.9612695	.030266	.391018	-1.505493	2.302713
9	-1.992744	1.086797	.032749	-1.969235	-4.130665	.124332
school						
U0:sigma2	4.314708	1.337713	.041102	4.146655	2.215281	7.493345
e.math5						
sigma2	28.26249	1.387124	.016318	28.22386	25.67543	31.11543

Note: Default priors are used for model parameters.

- Posterior mean estimates of random effects are similar to the ones predicted after `mixed`.
- Posterior standard deviations tend to be larger than the corresponding standard errors of the random effects predicted after `mixed` because they incorporate the uncertainty about the estimated regression coefficients and variance components.

- We can even plot the posterior distributions of the random effects. For example, let's look at the posterior distributions of the random intercepts for the first nine schools.

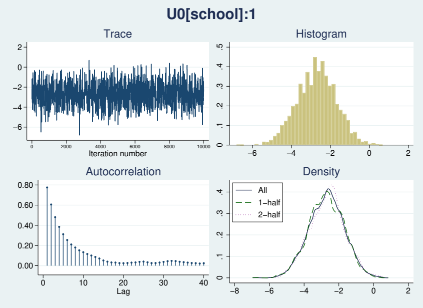
```
. bayesgraph histogram {U0[1/9]}, byparm
```



MCMC convergence

- We can check convergence and sampling efficiency of the MCMC for random-effects parameters just like any other model parameter.
- For example, here are graphical MCMC diagnostics for the first random effect:

```
. bayesgraph diagnostics {U0[1]}
```



Log marginal likelihood

- Notice from the header that the LML is not reported.
- As I mentioned earlier, Bayesian multilevel models may contain many model parameters, which include random-effects parameters.
- For models with many parameters, the computation of the LML can be time consuming, and its accuracy may become unacceptably low.
- Thus, the LML is not computed by default for multilevel models, but you can specify option `remargl` during estimation or on replay to compute it.
- LML is needed, for instance, if you want to compare Bayesian models using Bayes factors or using model posterior probabilities.

- To demonstrate, let's compute the LML on replay.

```
. bayes, remargl
```

  (*output omitted*)

```
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 =         48
                                                 Obs per group:
                                                              min =          5
                                                              avg =       18.5
                                                              max =         62
                                                 Number of obs    =        887
                                                 Acceptance rate  =      .8102
                                                 Efficiency:  min =     .03923
                                                              avg =      .3628
Log marginal-likelihood = -2801.7616                          max =      .7226
```

  (*output omitted*)

Convergence diagnostics using multiple chains

- We can use option `nchains()` with `bayes:` or `bayesmh` to generate multiple chains.
- And we can then use command `bayesstats grubin` to compute Gelman–Rubin convergence diagnostics for model parameters.
- Continuing with our previous linear random-intercept model, let's generate four chains and check convergence more formally.

```
. bayes, nchains(4) rseed(12345): mixed math5 math3 || school:
note: Gibbs sampling is used for regression coefficients and variance
      components.

Chain 1
  Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
  Simulation 10000 .........1000.........2000.........3000.........4000........
> .5000.........6000.........7000.........8000.........9000.........10000 done

Chain 2
  Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
  Simulation 10000 .........1000.........2000.........3000.........4000........
> .5000.........6000.........7000.........8000.........9000.........10000 done

Chain 3
  Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
  Simulation 10000 .........1000.........2000.........3000.........4000........
> .5000.........6000.........7000.........8000.........9000.........10000 done

Chain 4
  Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
  Simulation 10000 .........1000.........2000.........3000.........4000........
> .5000.........6000.........7000.........8000.........9000.........10000 done
```

(prior information omitted)

```
Bayesian multilevel regression                   Number of chains    =          4
Metropolis-Hastings and Gibbs sampling           Per MCMC chain:
                                                     Iterations      =     12,500
                                                     Burn-in         =      2,500
                                                     Sample size     =     10,000
Group variable: school                           Number of groups    =         48
                                                 Obs per group:
                                                                 min =          5
                                                                 avg =       18.5
                                                                 max =         62
                                                 Number of obs       =        887
                                                 Avg acceptance rate =       .812
                                                 Avg efficiency: min =     .04044
                                                                 avg =      .3583
                                                                 max =      .7284
Log marginal-likelihood                          Max Gelman-Rubin Rc =          1
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
math5						
math3	.6087103	.0325995	.00022	.6085895	.5448259	.672861
_cons	30.36082	.3559451	.00885	30.3587	29.66591	31.07091
school						
U0:sigma2	4.306511	1.336795	.019883	4.124045	2.23513	7.476504
e.math5						
sigma2	28.26538	1.380569	.008088	28.22864	25.68559	31.10292

Note: Default priors are used for model parameters.
Note: Default initial values are used for multiple chains.

- The summary information in the header and the estimation results are based on the four simulated chains.
- `bayes:` automatically reported the maximum value of the Gelman–Rubin statistic across all model parameters (excluding random effects).
- This value is less than 1.1, so we do not suspect convergence problems with this model.

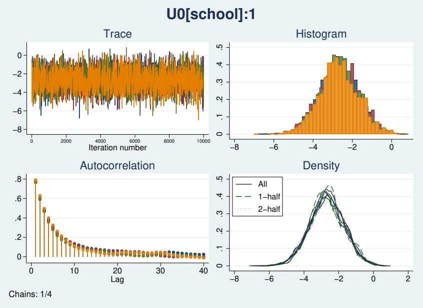
- We can use `bayesstats grubin` to check diagnostics for some of the random effects.
- Let's do this for the first nine random effects, sorted from largest to smallest diagnostic values.

```
. bayesstats grubin {U0[1/9]}, sort

Gelman-Rubin convergence diagnostic

Number of chains     =           4
MCMC size, per chain =      10,000
Max Gelman-Rubin Rc  =    1.001297

-------------------------
             |         Rc
-------------+-----------
U0[school]   |
           5 |   1.001297
           4 |   1.000745
           1 |   1.000586
           9 |   1.000525
           3 |   1.000452
           2 |    1.00044
           6 |    1.00034
           7 |   1.000297
           8 |   1.000111
-------------------------
Convergence rule: Rc < 1.1
```

- All values are less than 1.1.

- We can also explore all four chains graphically, for say, the first random effect:

```
. bayesgraph diagnostics {U0[1]}
```



- The results from all four chains agree.

Bayesian random-coefficients and higher-level models

- Similarly to classical multilevel models, we can fit other more complicated random-effects models by simply prefixing the corresponding `mixed` command with `bayes:`.
- A random-coefficient model assuming independence between random intercepts and random coefficients:

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

- A random-coefficient model with an unstructured covariance matrix for random intercepts and random coefficients:

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

- A three-level random-intercept model:

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

Bayesian multilevel models using bayesmh

- bayes: is convenient for fitting Bayesian multilevel models, but it is not as powerful or as flexible as bayesmh.
- For instance, you can relax the assumption of normality for random effects by using bayesmh.
- And now the new multilevel syntax of bayesmh allows you to fit more sophisticated models including nonlinear, multiple-equation linear, and multiple-equation nonlinear multilevel models. (The latter class of models is supported only by bayesmh!)

Multilevel syntax of `bayesmh`

- If you use `sem`, `gsem`, or `menl`, the multilevel syntax of `bayesmh` is the same, except:
  - you can specify crossed random effects `U[id1#id2]` in addition to nested random effects `U[id1>id2]`, and
  - you use `L[_n]` instead of simply `L` to specify latent factors.
- You can use any capitalized names in place of `U` and `L` above to specify random effects and latent factors.
- You can define random effects at various levels of hierarchy, `U[id1]`, `U[id1>id2]`, `U[id1>id2>id3]` and mix and match nested and crossed factors, `U[id1#id2<id3]`.
- You can interact random intercepts with covariates to include random coefficients: `c.age#U_age[id]`, `1.treat#U_trt1[id1>id2]`, and so on.
- And you can include random-effects and latent terms in nonlinear expressions!
- See section *Random effects* under *Remarks and examples* in **[BAYES] bayesmh** for details.

Random-intercept model using bayesmh

- Let's see how we can fit a random-intercept model using bayesmh:

```
. bayesmh math5 math3 U[school], likelihood(normal({var}))
> prior({math5:}, normal(10000))
> prior({var_U var}, igamma(0.01, 0.01) split)
```

- With bayesmh, we must specify the likelihood() model.
- U[school] is a random intercept {U} at the school level.
- By default, it is assumed to have a normal prior with mean 0 and variance {var_U}.
- But you must specify a prior for its variance and all other parameters!
- New suboption split within prior() specifies the same independent priors for the listed parameters.

- Let's fit the random-intercept model using `bayesmh`.

```
. set seed 12345
. bayesmh math5 math3 U[school], likelihood(normal({var}))
> prior({math5:}, normal(10000))
> prior({var_U var}, igamma(0.01, 0.01) split) dots

Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaa. done
Simulation 10000 .........1000.........2000.........3000.........4000.........5
> 000.........6000.........7000.........8000.........9000.........10000 done
Model summary
------------------------------------------------------------------------------
Likelihood:
  math5 ~ normal(xb_math5,{var})
Priors:
  {math5:math3 _cons} ~ normal(0,10000)                                    (1)
          {U[school]} ~ normal(0,{var_U})                                  (1)
                {var} ~ igamma(0.01,0.01)
Hyperprior:
  {var_U} ~ igamma(0.01,0.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_math5.
```

```
Bayesian normal regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =        887
                                                 Acceptance rate  =      .2169
                                                 Efficiency:  min =     .01418
                                                              avg =     .01643
Log marginal-likelihood                                       max =     .01914
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
math5						
math3	.6073655	.0304029	.002197	.6081756	.546117	.6630448
_cons	30.36476	.3355963	.027645	30.3647	29.68375	31.00273
var	28.23198	1.417103	.119004	28.22811	25.48952	31.08209
var_U	4.303251	1.265079	.095186	4.142379	2.316004	7.424923

```
// save MCMC results for later comparison
. bayesmh, saving(ri_nn_mcmc)
note: file ri_nn_mcmc.dta saved.
. estimates store ri_nn
```

- The estimates are similar to those from `bayes:mixed`. We also saved MCMC results for later comparison.

Full Gibbs sampling

- To increase efficiency, we can use Gibbs sampling for all model parameters, including random effects.

```
. set seed 12345

. bayesmh math5 math3 U[school], likelihood(normal({var})) ///
> prior({math5:}, normal(10000)) ///
> prior({var_U var}, igamma(0.01, 0.01) split) dots ///
> block({math5:}, gibbs) block({var var_U U}, gibbs split)

Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
Simulation 10000 .........1000.........2000.........3000.........4000.........5
> 000.........6000.........7000.........8000.........9000.........10000 done

Model summary
------------------------------------------------------------------------------
Likelihood:
  math5 ~ normal(xb_math5,{var})

Priors:
  {math5:math3 _cons} ~ normal(0,10000)                                    (1)
          {U[school]} ~ normal(0,{var_U})                                  (1)
                {var} ~ igamma(0.01,0.01)

Hyperprior:
  {var_U} ~ igamma(0.01,0.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_math5.
```

```
Bayesian normal regression                      MCMC iterations  =     12,500
Gibbs sampling                                  Burn-in          =      2,500
                                                MCMC sample size =     10,000
                                                Number of obs    =        887
                                                Acceptance rate  =          1
                                                Efficiency:  min =       .142
                                                             avg =      .5516
Log marginal-likelihood                                      max =      .8807
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
math5						
math3	.6087841	.0325005	.000347	.6088714	.5451194	.6721316
_cons	30.36373	.3641346	.009664	30.36565	29.63633	31.06866
var	28.2785	1.377501	.014678	28.22166	25.72931	31.12469
var_U	4.343227	1.318911	.023915	4.169909	2.297055	7.386686

- Sampling efficiencies are much higher now for all parameters.

Bayesian random-intercept model with Student's t errors

- Let's relax the normality assumption of the error term in our Bayesian random-intercept model:

$$e_{ij} \sim t(0, sc^2, df)$$

where *sc* is the scale parameter and *df* is the degrees of freedom.

```
. set seed 12345
. bayesmh math5 math3 U[school], likelihood(t({sc2},{df}))
> prior({math5:}, normal(10000))
> prior({var_U} {sc2}, igamma(0.01, 0.01))
> prior({df}, uniform(0,1000))
> block({sc2 df}) saving(ri_nt_mcmc) dots

Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
Simulation 10000 .........1000.........2000.........3000.........4000.........5
> 000.........6000.........7000.........8000.........9000.........10000 done
Model summary
------------------------------------------------------------------------------
Likelihood:
  math5 ~ t(xb_math5,{sc2},{df})
Priors:
  {math5:math3 _cons} ~ normal(0,10000)                                    (1)
          {U[school]} ~ normal(0,{var_U})                                  (1)
                {sc2} ~ igamma(0.01,0.01)
                 {df} ~ uniform(0,1000)
Hyperprior:
  {var_U} ~ igamma(0.01,0.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_math5.
```

```
Bayesian t regression                          MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling       Burn-in          =      2,500
                                               MCMC sample size =     10,000
                                               Number of obs    =        887
                                               Acceptance rate  =      .2713
                                               Efficiency:  min =     .01434
                                                            avg =     .05264
Log marginal-likelihood                                     max =     .08265
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
math5						
math3	.5902118	.0322714	.001186	.5903328	.5236286	.6536404
_cons	30.7968	.3565133	.029768	30.80539	30.06264	31.47062
sc2	18.66776	1.730679	.060198	18.59467	15.31439	22.07594
df	5.85193	1.315771	.048748	5.6429	3.887774	8.945624
var_U	4.124481	1.256536	.090361	3.940338	2.269366	7.121272

```
file ri_nt_mcmc.dta saved.
. estimates store ri_nt
```

- The estimate of the df parameter is about 6 with a 95% CrI of (3.9, 8.9), which suggests somewhat heavier tails for the error-term distribution.

Student's t random-effects distribution

- The normal distribution is typically assumed for subject-specific random effects u_{0j}'s.
- But we can relax this assumption and model random effects using, say, a Student's t distribution:

$$u_{0j} \sim t(0, sc_u^2, df_u)$$

where sc_u is the scale parameter and df_u is the degrees of freedom.

```
. set seed 12345
. bayesmh math5 math3 U[school], likelihood(normal({var}))
> prior({math5:}, normal(10000))
> prior({U}, t(0,{sc2_U},{df_U}))
> prior({sc2_U var}, igamma(0.01, 0.01) split)
> prior({df_U}, uniform(0,1000))
> block({sc2_U df_U}) saving(ri_tn_mcmc) dots

Burn-in 2500 aaaaaaaaa1000aaa......2000..... done
Simulation 10000 .........1000.........2000.........3000.........4000.........5
> 000.........6000.........7000.........8000.........9000.........10000 done
Model summary
------------------------------------------------------------------------------
Likelihood:
  math5 ~ normal(xb_math5,{var})
Priors:
  {math5:math3 _cons} ~ normal(0,10000)                                   (1)
          {U[school]} ~ t(0,{sc2_U},{df_U})                               (1)
                {var} ~ igamma(0.01,0.01)
Hyperpriors:
  {sc2_U} ~ igamma(0.01,0.01)
   {df_U} ~ uniform(0,1000)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_math5.
```

```
Bayesian normal regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =        887
                                                 Acceptance rate  =      .2233
                                                 Efficiency:  min =     .01916
                                                              avg =     .05082
Log marginal-likelihood                                       max =      .1038
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
math5						
math3	.6096901	.0334464	.001546	.6106509	.5449196	.674595
_cons	30.3634	.3393116	.024514	30.35326	29.70554	31.0354
var	28.22595	1.313612	.06295	28.14754	25.70595	30.85718
sc2_U	4.270535	1.26931	.062858	4.159509	2.202217	7.239853
df_U	517.2	282.73	8.77644	511.7186	36.58858	975.0253

```
file ri_tn_mcmc.dta saved.
. estimates store ri_tn
```

- The estimate of df for the random-effects distribution, {df_U}, is about 517. Given its large value, the normal random-effects distribution is preferable to the *t* distribution.

Student's t random-effects and error distributions

- We can relax the normality assumption for both random effects and error terms:

$$
\begin{aligned}
e_{ij} &\sim t(0, sc^2, df) \\
u_{0j} &\sim t(0, sc_u^2, df_u)
\end{aligned}
$$

  where sc and sc_u are the respective scale parameters and df and df_u are the degrees-of-freedom parameters.

- From the previous slide, there is no need to use the t distribution for random effects in our example, but let's do this for completeness.

```
. set seed 12345
. bayesmh math5 math3 U[school], likelihood(t({sc2},{df}))
> prior({math5:}, normal(10000))
> prior({U}, t(0,{sc2_U},{df_U}))
> prior({sc2_U sc2}, igamma(0.01, 0.01) split)
> prior({df_U df}, uniform(0,1000))
> block({sc2_U df_U}) block(sc2 df) saving(ri_tt_mcmc) dots

Burn-in 2500 aaaaaaaaa1000aaaaaaa..2000..... done
Simulation 10000 .........1000.........2000.........3000.........4000.........5
> 000.........6000.........7000.........8000.........9000.........10000 done
Model summary
------------------------------------------------------------------------------
Likelihood:
  math5 ~ t(xb_math5,{sc2},{df})
Priors:
  {math5:math3 _cons} ~ normal(0,10000)                                    (1)
          {U[school]} ~ t(0,{sc2_U},{df_U})                                (1)
                {sc2} ~ igamma(0.01,0.01)
                 {df} ~ uniform(0,1000)
Hyperpriors:
  {sc2_U} ~ igamma(0.01,0.01)
   {df_U} ~ uniform(0,1000)
------------------------------------------------------------------------------
```

```
Bayesian t regression                            MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =        887
                                                 Acceptance rate  =      .2363
                                                 Efficiency:  min =     .02143
                                                              avg =     .06284
Log marginal-likelihood                                       max =     .09383
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
math5						
math3	.592927	.0329479	.001365	.5932803	.5294996	.659188
_cons	30.83308	.3391586	.023171	30.82126	30.17718	31.52452
sc2	18.52484	1.758456	.060697	18.45145	15.37084	22.17011
df	5.843741	1.374977	.047419	5.592489	3.779647	9.112901
sc2_U0	4.006495	1.12213	.059537	3.856532	2.306424	6.635418
df_U0	504.5605	287.3822	9.38173	505.4225	29.11466	970.731

```
file ri_tt_mcmc.dta saved.
. estimates store ri_tt
```

Model comparison

```
. bayesstats ic ri_nn ri_tn ri_nt ri_tt, diconly
Deviance information criterion
```

	DIC
ri_nn	5514.742
ri_tn	5515.442
ri_nt	5477.963
ri_tt	5477.744

- DIC is the smallest for the `ri_tt` model with both errors and random effects distributed according to $t()$.
- But given the large estimate of the df for the random-effects distribution, the `ri_nt` model with normal random effects and $t()$ errors would be better in practice.
- In the above, we computed the so-called conditional DIC. But other information criteria might be more suitable for some multilevel models (Merkle, Furr, Rabe-Hesketh 2019).

Random-coefficients and higher-level models

- A random-coefficient model assuming independence between random intercepts and random coefficients:

```
. bayesmh math5 math3 U0[school] c.math3#U1[school],
> likelihood(normal({var}))
> prior({math5:}, normal(10000))
> prior({var_U0 var_U1 var}, igamma(0.01, 0.01) split)
```

- A random-coefficient model with an unstructured covariance matrix for random intercepts and random coefficients:

```
. bayesmh math5 math3 U0[school] c.math3#U1[school],
> likelihood(normal({var}))
> prior({math5:}, normal(10000))
> prior({var}, igamma(0.01, 0.01))
> prior({U0 U1}, mvn(2,0,0,Sigma,m))
> prior({Sigma,m}, iwishart(2,3,I(2)))
```

- A three-level random-intercept model:

```
. bayesmh math5 math3 U[school] UU[teacher<school],
> likelihood(normal({var}))
> prior({math5:}, normal(10000))
> prior({var_U var_UU var}, igamma(0.01, 0.01) split)
```

Nonlinear multilevel models

- Consider the following logistic growth model

$$y_{ij} = \frac{C_i}{1 + d \times C_i \times e^{-B_i \times t_{ij}}} + \epsilon_{ij}$$

  for measurements y_{ij}'s on subjects $i = 1, 2, \ldots, I$ at times t_{ij} for $j = 1, 2, \ldots, n_i$.
- Error terms $\epsilon_{ij} \sim N(0, \sigma^2)$.
- Random effects $(C_i, B_i) \sim N_2(c, b, \Sigma)$.
- C_i is subject-specific maximum growth.
- B_i is subject-specific growth rate.
- d is the initial-growth multiplier.

- Jones et al. (2005) used the above formulation to model weight of black-fronted tern chicks.
- Let's see how we could fit this model using bayesmh.
- Suppose id is the chick identifier, y is the weight (g), and time is days since birth.

```
. bayesmh y = ({C[id]}/(1+{d}*{C[id]}*exp(-{B[id]}*time))),
> likelihood(normal({var}))
> prior({d}, exp(1))
> prior({var}, igamma(0.01, 0.01))
> prior({C B}, mvnormal(2,{c},{b},{Sigma,m}))
> prior({c b}, normal(0,100))
> prior({Sigma,m}, iwishart(2,3,I(2)))
> block({d b c}, split) block({Sigma,m}, gibbs)
> initial({c} 100 {d} 1) mcmcsize(2500) rseed(17)
```

```
Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
Simulation 2500 .........1000.........2000..... done

Model summary
------------------------------------------------------------------------------
Likelihood:
  y ~ normal({C[id]}/(1+{d}*{C[id]}*exp(-{B[id]}*time)),{var})

Priors:
          {var} ~ igamma(0.01,0.01)
            {d} ~ exponential(1)
  {C[id] B[id]} ~ mvnormal(2,{c},{b},{Sigma,m})

Hyperpriors:
      {c b} ~ normal(0,100)
  {Sigma,m} ~ iwishart(2,3,I(2))
------------------------------------------------------------------------------
```

```
Bayesian normal regression                          MCMC iterations  =      5,000
Metropolis-Hastings and Gibbs sampling              Burn-in          =      2,500
                                                    MCMC sample size =      2,500
                                                    Number of obs    =        414
                                                    Acceptance rate  =      .4564
                                                    Efficiency:  min =      .0305
                                                                 avg =     .08921
Log marginal-likelihood                                          max =      .1582
```

	Mean	Std. dev.	MCSE	Median	Equal-tailed [95% cred.	interval]
d	.7539755	.0006552	.000075	.7538968	.7529732	.7555175
var	344.227	26.9721	1.35627	342.3408	297.2411	402.3016
c	49.37715	7.742743	.621674	50.20698	31.32951	62.09946
b	.3865898	.0805282	.007119	.3918832	.2201669	.5332942
Sigma_1_1	1109.745	531.1578	31.4845	999.7406	483.4897	2451.883
Sigma_2_1	5.357392	3.498629	.220185	4.658808	.567264	14.1419
Sigma_2_2	.0950801	.0363054	.002212	.0882656	.0493029	.1907358

- The estimated average maximum weight {c} is 49 grams but there is certainly variability in maximum weights ({Sigma_1_1}= 1,110) and in weight gain rates ({Sigma_2_2}= .095) among chicks.

Multivariate nonlinear multilevel models

- Jones et al. (2005) actually considered a Bayesian bivariate growth model to study the growth of black-fronted tern chicks.
- A linear growth model was assumed for wing length y_1, and the earlier logistic growth model was assumed for weight y_2:

$$\begin{aligned} y_{1,ij} &= U_i + V_i \times t_{ij} + \epsilon_{1,ij} \\ y_{2,ij} &= \frac{C_i}{1 + d \times C_i \times e^{-B_i \times t_{ij}}} + \epsilon_{2,ij} \end{aligned}$$

- Error terms $(\epsilon_{1,ij}, \epsilon_{2,ij}) \sim N(0, 0, \Sigma_0)$.
- Random effects $(U_i, V_i, C_i, B_i) \sim N_4(u, v, c, b, \Sigma)$.
- U_i is chick-specific initial wing length.
- V_i is chick-specific wing growth rate.
- C_i is chick-specific maximum weight.
- B_i is chick-specific weight gain rate.
- d is the initial-weight multiplier.

- Suppose `id` is the chick identifier, `y1` is the wing length (mm), `y2` is the weight (g), and `time` is days since birth.
- The corresponding `bayesmh` specification is

```
. bayesmh (y1 = ({U[id]} + time*{V[id]}))
>         (y2 = ({C[id]}/(1+{d}*{C[id]}*exp(-{B[id]}*time)))),
> likelihood(mvnormal({Sigma0,m}))
> prior({U V C B},  mvnormal(4,{u},{v},{c},{b},{Sigma,m}))
> prior({u v c b},  normal(0, 100))
> prior({Sigma0,m}, iwishart(2,3,I(2)))
> prior({Sigma,m},  iwishart(4,5,I(4)))
> prior({d}, exp(1))
> block({d u v b c}, split) block({Sigma0,m} {Sigma,m}, gibbs split)
> init({U[id] u} -10 {V[id] v} 10 {C[id] c} 100 {d} 1) mcmcsize(2500) rseed(17)
```

```
Burn-in 2500 aaaaaaaaa1000aaaaaaaaa2000aaaaa done
Simulation 2500 .........1000.........2000..... done

Model summary
------------------------------------------------------------------------------
Likelihood:
  y1 y2 ~ mvnormal(2,<expr1>,<expr2>,{Sigma0,m})

Priors:
                 {Sigma0,m} ~ iwishart(2,3,I(2))
  {U[id] V[id] C[id] B[id]} ~ mvnormal(4,{u},{v},{c},{b},{Sigma,m})
                        {d} ~ exponential(1)

Hyperpriors:
  {u v c b} ~ normal(0,100)
  {Sigma,m} ~ iwishart(4,5,I(4))

Expressions:
  expr1 : {U[id]} + time*{V[id]}
  expr2 : {C[id]}/(1+{d}*{C[id]}*exp(-{B[id]}*time))
------------------------------------------------------------------------------
```

```
Bayesian multivariate normal regression          MCMC iterations  =      5,000
Metropolis-Hastings and Gibbs sampling           Burn-in          =      2,500
                                                 MCMC sample size =      2,500
                                                 Number of obs    =        414
                                                 Acceptance rate  =      .4713
                                                 Efficiency:  min =     .01174
                                                              avg =      .2265
Log marginal-likelihood                                       max =      .7028

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. dev.     MCSE     Median  [95% cred. interval]
-------------+----------------------------------------------------------------
           d |  .0634061   .0025888   .000478   .0635744   .0579154   .0680656
           u | -12.84796   3.011731   .255283  -12.97586  -18.25202  -6.451113
           v |  5.977761   .2446379   .023368   5.990374   5.422395   6.404792
           c |  78.42872   3.602142   .368536    78.7988   70.10973   84.34357
           b |  .2208688   .0471093   .002637   .2229167   .1242395   .3148616
  Sigma0_1_1 |  7.956314   .5825538   .017417   7.926544   6.871581   9.158582
  Sigma0_2_1 |  2.625951   .6406367   .021819   2.632427   1.430312   3.875557
  Sigma0_2_2 |  18.85203   1.342218   .038113   18.81303   16.36956   21.67296
   Sigma_1_1 |  192.8405   67.11091   2.92639   179.5316    101.754   362.8648
   Sigma_2_1 | -8.029962   4.209058    .21859  -7.334189  -17.74035  -1.783869
   Sigma_3_1 | -108.4137   63.18093   3.39159  -97.77067  -258.3206  -18.55377
   Sigma_4_1 |  .4582266   .6998019   .021466   .4405483  -.8234645   1.983518
   Sigma_2_2 |  1.193545   .4200058   .025011    1.10642   .6352668   2.223882
   Sigma_3_2 |  12.45667   5.664299   .404336   11.29209   5.259565   27.34906
   Sigma_4_2 | -.0023492   .0557342   .001842  -.0034794  -.1104773   .1078309
   Sigma_3_3 |  234.2312   95.14968   6.93288   212.8518   117.8635   471.0824
   Sigma_4_3 | -.2949588    .829987   .032991  -.2727646  -2.063978   1.386505
   Sigma_4_4 |  .0454308   .0136201   .000325   .0428103   .0257433   .0790052
------------------------------------------------------------------------------
```

- The wing-length and weight measurements appear to be correlated.

Summary

- Bayesian multilevel modeling inherits all of the benefits of generic Bayesian modeling such as
  - incorporating prior information about model parameters into your analysis;
  - providing intuitive and direct interpretations of results by using probability statements about parameters; and
  - providing a way to assign an actual probability to any hypothesis of interest.
- Compared with classical multilevel models, Bayesian multilevel models additionally assume that other model parameters—regression coefficients and variance components—are random.

Summary (cont.)

- Bayesian multilevel models do not integrate out random effects but estimate them together with other model parameters.
- Bayesian multilevel models provide marginal posterior distributions for all random effects. As such, convergence of MCMC chains for random-effects parameters should also be checked.
- LML is not reported by default for Bayesian multilevel models because its precision decreases as the number of random effects increases. For a moderate number of random effects, you can use option `remargl` to compute it.

Summary (cont.)

- High autcorrelation occurs frequently in Bayesian multilevel models. More informative priors or model simplifications are often needed.
- You can use `bayes:mixed` to fit Bayesian multilevel models to a continuous outcome.
- You can use `bayes:`*mecommand* to fit Bayesian multilevel models to other types of outcomes such as binary and ordinal.
- You can use `bayesmh` to fit more sophisticated Bayesian multilevel models.

- Read more about **Bayesian multilevel modeling** at
  `stata.com/new-in-stata/bayesian-multilevel-modeling/`
- Check out all new Bayesian features at
  `stata.com/new-in-stata/new-in-bayesian-analysis/`
- And see the **[BAYES] Bayesian analysis** manual for more examples and details about Bayesian analysis.

Jones, G., R. J. Keedwell, A. D. L. Noble, and D. I. Hedderley. 2005. Dating chicks: Calibration and discrimination in a nonlinear multivariate hierarchical growth model. *Journal of Agricultural, Biological, and Environmental Statistics* 10: 306–320.

Merkle, E.C., D. Furr, S. Rabe-Hesketh. 2019. Bayesian Comparison of Latent Variable Models: Conditional Versus Marginal Likelihoods. *Psychometrika* 84, 802–829.

Mortimore, P., P. Sammons, L. Stoll, D. Lewis, and R. Ecob. 1988. *School Matters: The Junior Years*. Wells, Somerset, UK: Open Books.