



help mcmcsum

Title

mcmcsum - **runmlwin** postestimation - MCMC summary statistics and plots

Syntax

mcmcsum in parameter mode

mcmcsum *parameter_list* [*if*] [*in*] [, *plot* **thinning**(#) **getchains**]

mcmcsum in variable mode

mcmcsum *varlist* [*if*] [*in*] [, *plot* **thinning**(#)] **variables**

where *plot* is one of:

trajectories trajectory plot for each parameter/variable

densities kernel density plot for each parameter/variable

fiveway trajectory, kernel density, ACF, PACF, and MCSE plots for a single chosen parameter/variable.

If no plot is specified a table of MCMC summary statistics is displayed.

Description

mcmcsum is a postestimation command for **runmlwin**. **mcmcsum** calculates and displays a variety of MCMC summary statistics and plots. These statistics and plots can be based on either the current **runmlwin** estimation results or the variables currently in memory.

Options

Main

trajectories display a trajectory plot for each parameter/variable.

densities displays a kernel density plot for each parameter/variable.

fiveway displays a five-way plot containing the MCMC trajectory plot, kernel density plot, ACF plot, PACF plot, and MCSE plot for the chosen parameter. The trajectory and kernel density plots are as described above. Note that only one parameter/variable can be specified when using this option.

thinning(#) specifies that thinning every # iterations was used when storing the MCMC chains.

getchains save the MCMC parameter chains from the current **runmlwin** estimation results as variables in the current data set. Note this will overwrite the current data set.

variables reads the data from the variables currently in memory instead of the current estimation results. The latter is useful if the MCMC chains for a set of model parameters have been saved to disk.

Remarks

Remarks are presented under the following headings:

Remarks on MCMC summary statistics

Remarks on MCMC plots

Remarks on referencing specific parameters when using parameters mode

Remarks on referencing specific parameters when using variables mode

Remarks on MCMC summary statistics

mcmcsum calculates and displays a variety of MCMC summary statistics.

- (1) The chain mean (posterior mean). This gives the parameter point estimate.
- (2) The MCSE of this mean. The MCSE will decrease as the chain length is increased.
- (3) The chain standard deviation. This gives the parameter standard error.
- (4) The chain mode.
- (5) The proportion of chain values of the opposite sign to the chain mean.
- (6) The proportion of chain values of the opposite sign to the chain mode.
- (7) The proportion of chain values of the opposite sign to the chain median.
- (8) The 0.5th, 2.5th, 5th, 25th, 50th, 75th, 95th, 97.5th, and 99.5th quantiles. The 2.5th and 97.5th quantiles give the 95% Bayesian credible interval. This is equivalent to a 95% confidence interval in maximum likelihood. Unlike maximum likelihood, this is not based on a normal sampling distribution assumption.
- (9) The thinned chain length.
- (10) The Effective Sample Size (ESS) gives an estimate of the equivalent number of independent iterations that the chain represents. The ESS will typically be less than the number of actual iterations because the chain is positively autocorrelated (it is a Markov chain).
- (11) Brooks-Draper (mean): This statistic is based on the mean of the distribution. It is used to estimate the length of chain required to produce a mean estimate to 2 significant figures with a given accuracy.
- (12) Raftery-Lewis (quantile): This statistic is based on the 2.5th and 97.5th quantiles of the posterior distribution (i.e. the 95% credible interval). It is used to estimate the length of chain required to estimate the boundaries of the 95% credible interval to a given accuracy.

We recommend users seeking further information to consult the comprehensive MLwiN MCMC manual by Browne (2009).

Remarks on MCMC plots

mcmcsum calculates and displays a variety of MCMC plots.

Trajectory plots can be thought of as "time series" plots of each chain. The chain values are plotted against the iteration number. Healthy chains are those that resemble white noise.

Kernel density plots are smoothed histograms of the chains. They plot the posterior distributions, the fundamental things of interest. Note that posterior distributions for variance parameters will typically be right skewed.

The ACF plot shows the autocorrelation between iteration t and $t - k$. A Markov chain should have a power relationship in the lags i.e. if $ACF(1) = \rho$ then $ACF(2) = \rho^2$ etc. This is known as an AR(1) process. The less correlated the chain the better.

The PACF plot shows the autocorrelation between iteration t and $t - k$, having accounted for $t - 1, \dots, t - (k - 1)$. It is used to identify the extent to which the chain departs from an ACF(1). That is, it is used to identify the extent of the lag in the chain. Look for the point on the plot where the partial autocorrelations for all higher lags are essentially zero.

The MCSE is an indication of how much error is in the mean estimate due to the fact that MCMC is used. As the number of iterations increases the MCSE tends to 0. The MCSE is used to calculate how long to run the chain to achieve a mean estimate with a particular desired MCSE.

We recommend users seeking further information to consult the comprehensive MLwiN MCMC manual by Browne (2009).

Remarks on referencing specific parameters when using parameters mode

You can find the names assigned to parameters by `runmlwin` using the `mat list e(b)` command. For example, if your model contains the parameter `FP1:cons`, you would refer to this as `[FP1]cons`. Similarly, the parameter `RP2:var(cons)` would be referred to as `[RP2]var(cons)`. See the **Examples** section for an example.

Remarks on referencing specific parameters when using variables mode

An alternative to referencing the parameters in the current `runmlwin` estimation results is to use the `getchains` option to save these parameter chains as variables in the current data set. Note this will overwrite the current data set. For example, if your model contains the parameter `FP1:cons`, this would be saved as the variable `FP1_cons` in your current data set. Similarly, the parameter `RP2:var(cons)` would be saved as `RP2_var_cons_`. See the **Examples** section for an example.

Examples

The following examples will only work on your computer if you have installed `runmlwin`.

Two-level random-intercept model, analogous to `xtreg`

Setup

```
. use http://www.bristol.ac.uk/cmm/media/runmlwin/tutorial, clear
```

Fit model using IGLS

```
. runmlwin normexam cons standlrt, level2(school: cons) level1(student: cons) nopause
```

Fit model using MCMC

```
. runmlwin normexam cons standlrt, level2(school: cons) level1(student: cons) mcmc(on) initsprevious nopause
```

Calculate and display a variety of MCMC summary statistics for all model parameters

```
. mcmcsun
```

Trajectory plots for all model parameters

```
. mcmcsun, trajectories
```

Kernel density plots for all model parameters

```
. mcmcsun, densities
```

Fiveway plot for the level 2 variance parameter (`[RP2]var(cons)`)

```
. mcmcsun [RP2]var(cons), fiveway
```

Save the MCMC parameter chains from the current `runmlwin` model as variables in the current data set

```
. mcmcsun, getchains
```

Compute the intraclass correlation (a non-linear combination of model parameters)

```
. gen icc = RP2 var cons / (RP2 var cons + RP1 var cons )
```

Calculate and display a variety of MCMC summary statistics for the derived ICC parameter

```
. mcmcsun icc, variables
```

Fiveway plot for the ICC parameter

```
. mcmcsun icc, fiveway variables
```

Saved results

mcmcsun saves the following in **r()** when no **plot** is specified:

Scalars

r(thinnedchain)	length of chain after thinning
r(mean)	mean of parameter chain
r(mode)	mode of parameter chain
r(sd)	standard deviation of parameter chain
r(ess)	effective sample size
r(meanmce)	mean Monte-Carlo standard error
r(bd)	Brook-Draper diagnostic statistic
r(rlub)	Raftery-Lewis upper bound
r(rllb)	Raftery-Lewis lower bound
r(p99_5)	99.5% quantile of the chain
r(p95)	95% quantile of the chain
r(p75)	75% quantile of the chain
r(p50)	50% quantile (median) of the chain
r(p25)	25% quantile of the chain
r(p5)	5% quantile of the chain
r(p2_5)	2.5% quantile of the chain
r(p0_5)	0.5% quantile of the chain

About the Centre for Multilevel Modelling

The MLwiN software is developed at the Centre for Multilevel Modelling. The Centre was established in 1986, and has been supported largely by project grants from the UK Economic and Social Research Council. The Centre has been based at the University of Bristol since 2005.

The Centre's website:

<http://www.bristol.ac.uk/cmm>

contains much of interest, including new developments, and details of courses and workshops. This website also contains the latest information about the MLwiN software, including upgrade information, maintenance downloads, and documentation.

The Centre also runs a free online multilevel modelling course:

<http://www.bristol.ac.uk/cmm/learning/course.html>

which contains modules starting from an introduction to quantitative research progressing to multilevel modelling of continuous and categorical data. Modules include a description of concepts and models and instructions of how to carry out analyses in MLwiN, Stata and R. There is also a user forum, videos and interactive quiz questions for learners' self-assessment.

Citation of runmlwin and MLwiN

runmlwin (and **mcmcsun**) is not an official Stata command. It is a free contribution to the research community, like a paper. Please cite it as such:

Leckie, G. and Charlton, C. 2011. **runmlwin**: Stata module for fitting multilevel models in the MLwiN software package. Centre for Multilevel Modelling, University of Bristol.

Similarly, please also cite the MLwiN software:

Rasbash, J., Charlton, C., Browne, W.J., Healy, M. and Cameron, B. 2009. MLwiN Version 2.1. Centre for Multilevel Modelling, University of Bristol.

For models fitted using MCMC estimation, we ask that you additionally cite:

Browne, W.J. 2009. MCMC Estimation in MLwiN, v2.13. Centre for Multilevel Modelling, University of Bristol.

The runmlwin user forum

Please use the **runmlwin** user forum to post any questions you have about **mcmcsum** (or **runmlwin**). We will try to answer your questions as quickly as possible, but where you know the answer to another user's question please also reply to them!

<http://www.cmm.bristol.ac.uk/forum/>

Authors

Chris Charlton
Centre for Multilevel Modelling
University of Bristol
c.charlton@bristol.ac.uk

George Leckie
Centre for Multilevel Modelling
University of Bristol

Acknowledgments

The code to calculate the MCMC summary statistics was adapted from that written by Bill Browne for the MCMC engine in the MLwiN software (Browne, 2009). We are very grateful to colleagues at the Centre for Multilevel Modelling and the University of Bristol for their useful comments.

The development of this command was funded under the LEMMA project, a node of the UK Economic and Social Research Council's National Centre for Research Methods (grant number RES-576-25-0003).

Disclaimer

mcmcsum comes with no warranty. Where users are using **mcmcsum** after fitting a model by **runmlwin**, we recommend that users check their results with those obtained through operating MLwiN by its graphical user interface.

References

Browne, W.J. 2009. MCMC Estimation in MLwiN, v2.13. Centre for Multilevel Modelling, University of Bristol.
<http://www.bristol.ac.uk/cmm/software/mlwin/download/manuals.html>

Also see

Online: **runmlwin**, **reffadjust**, **winbugs**