Introduction to Bayesian Analysis in Stata

Gustavo Sánchez

StataCorp LLC

September 15 , 2017
Porto, Portugal

Outline

1. Bayesian analysis: The general idea
2. Basic Concepts
   - The Method
   - The tools
     - Stata 14: The `bayesmh` command
     - Stata 15: The `bayes` prefix
     - Postestimation commands
3. A few examples
   - Linear regression
   - Panel data random effect probit model
   - Change point model

The general idea

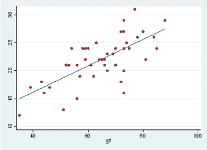
Frequentist

Theoretical Model →

```
. list ln_wage union hours wks_work tenure race grade ttl_exp in 1/15, noobs
```

ln_wage	union	hours	wks_work	tenure	race	grade	ttl_exp
1.451214	.	20	27	.0833333	black	12	1.083333
1.02862	.	44	10	.0833333	black	12	1.275641
1.589977	1	40	51	.9166667	black	12	2.25641
1.780273	.	40	3	.0833333	black	12	2.314102
1.777012	.	10	24	.1666667	black	12	2.775641
1.778681	0	32	52	1.5	black	12	3.775641
2.493976	.	52	4	.0833333	black	12	3.852564
2.551715	1	45	75	1.833333	black	12	5.294872
2.420261	1	49	101	.6666667	black	12	5.294872
2.614172	1	42	97	1.916667	black	12	7.160256
2.536374	1	45	95	3.916667	black	12	8.98718
2.462927	1	48	70	5.333333	black	12	10.33333
1.360348	0	40	13	.25	black	12	.7115384
1.206198	.	40	22	1	black	12	1.134615
1.549883	.	40	17	.3333333	black	12	1.461538

→

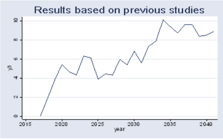


The general idea

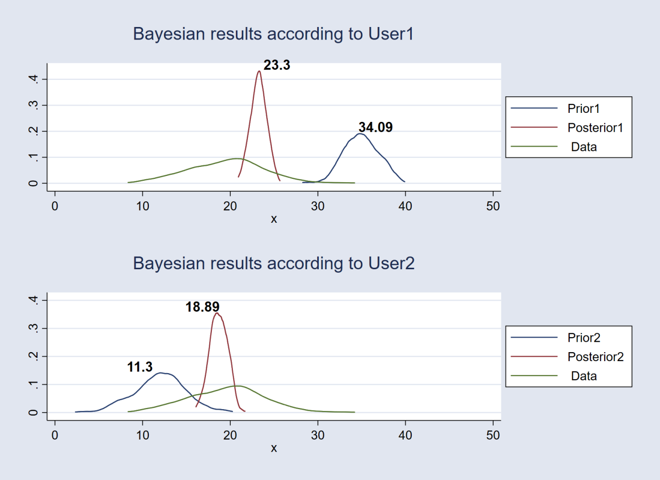
Bayesian

```
. list ln_wage union hours wks_work tenure race grade ttl_exp in 1/15, noobs
```

ln_wage	union	hours	wks_work	tenure	race	grade	ttl_exp
1.451214	.	20	27	.0833333	black	12	1.083333
1.02862	.	44	10	.0833333	black	12	1.275641
1.589977	1	40	51	.9166667	black	12	2.25641
1.780273	.	40	3	.0833333	black	12	2.314102
1.777012	.	10	24	.1666667	black	12	2.775641
1.778681	0	32	52	1.5	black	12	3.775641
2.493976	.	52	4	.0833333	black	12	3.852564
2.551715	1	45	75	1.833333	black	12	5.294872
2.420261	1	49	101	.6666667	black	12	5.294872
2.614172	1	42	97	1.916667	black	12	7.160256
2.536374	1	45	95	3.916667	black	12	8.98718
2.462927	1	48	70	5.333333	black	12	10.33333
1.360348	0	40	13	.25	black	12	.7115384
1.206198	.	40	22	1	black	12	1.134615
1.549883	.	40	17	.3333333	black	12	1.461538



The general idea



Bayesian Analysis vs Frequentist Analysis

Frequentist Analysis

- Results are based on estimations for unknown fixed parameters.
- The data are considered to be a (hypothetical) repeatable random sample.
- Uses the data to obtain estimates about the unknown fixed parameters.
- Depends on whether the data satisfies the assumptions for the specified model.

"Frequentists base their conclusions on the distribution of statistics derived from random samples, assuming that the parameters are unknown but fixed."

Bayesian Analyis

- Results are based on probability distributions about unknown random parameters
- The data are considered to be fixed.
- The results are produced by combining the data with prior beliefs about the parameters.
- The posterior distribution is used to make explicit probabilistic statements

"Bayesian analysis answers questions based on the distribution of parameters conditional on the observed sample."

Some Advantages

- Based on the Bayes rule, which applies to all parametric models.
- Inference is exact, estimation and prediction are based on posterior distribution.
- Provides more intuitive interpretation in terms of probabilities (e.g Credible intervals).
- It is not limited by the sample size.

Some Disadvantages

- Subjectivity in specifying prior beliefs.
- Computationally challenging.
- Setting up a model and performing analysis could be an involving task.

Some Examples (Taken from Hahn, 2014)

- TranScan Medical use small dataset and priors based on previous studies to determine the efficacy of its 2000 device for mammografy (FDA 1999).
- homeprice.com.hk used Bayesian analysis for pricing information on over a million real state properties in Hong Kong and surrounding areas (Shamdasany, 2011).
- Researchers in the energy industry have used Bayesian analysis to understand petroleum reservoir parameters (Glinsky and Gunning, 2011).

The Method

The Method

- Let's start by writing the Bayes' Rule:

$$p(B|A) = \frac{p(A|B)\,p(B)}{p(A)}$$

Where:

$p(A|B)$: conditional probability of A given B
$p(B|A)$: conditional probability of B given A
$p(B)$: marginal probability of B
$p(A)$: marginal probability of A

The Method

- If we have a probability model for a vector of observations **y** and a vector of unknown parameters θ, we can represent the model with a likelihood function:

$$L(\theta; y) = f(y; \theta) = \prod_{i=1}^{n} f(y_i|\theta)$$

Where:

$f(y;\theta)$: conditional probability of y give θ

- Let's assume that θ has a probability distribution $\pi(\theta)$, and that denote **m(y)** denote the marginal distribution of y, such that:

$$m(y) = \int f(y;\theta)\,\pi(\theta)\,d\theta$$

The Method

- Let's now write the inverse law of probability (Bayes' Theorem):

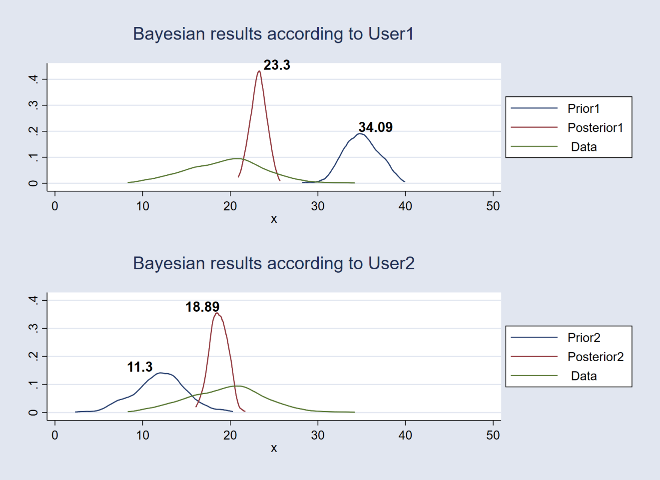
$$f(\theta|y) \quad = \quad \frac{f(y;\theta)\,\pi(\theta)}{f(y)}$$

- But notice that the marginal distribution of y, f(y), does not depend on (θ)

- Then, we can write the fundamental equation for Bayesian analysis:

$$p(\theta|y) \propto L(y|\theta)\,\pi(\theta)$$

Let's go back to our initial example



The Method

- In the example we have the data (the likelihood component)
- We also have the experts belief (the prior component)
- Then, how do we get the posterior distribution?
- We use the fundamental equation

$$p(\theta|y) \propto L(y|\theta)\pi(\theta)$$

The Method

- Let's assume that both, the data and the prior beliefs, are normally distributed:
  - **The data**: $y \sim N\left(\theta, \sigma_d^2\right)$
  - **The prior**: $\theta \sim N\left(\mu_p, \sigma_p^2\right)$
- Homework...: Doing the algebra with the fundamental equation we find that the posterior distribution would be normal with:
  - **The posterior**: $\theta|y \sim N\left(\mu, \sigma^2\right)$

  Where:

$$\begin{aligned} \mu &= \sigma^2\left(N\bar{y}/\sigma_d^2 + \mu_p/\sigma_p^2\right) \\ \sigma^2 &= \left(N/\sigma_d^2 + 1/\sigma_p^2\right)^{-1} \end{aligned}$$

The Method

- Doing the algebra was relatively straightforward in the previous case.

- What about more complex distributions?
  - Integration is performed via simulation
  - We need to use intensive computational simulation tools to find the posterior distribution in most cases.

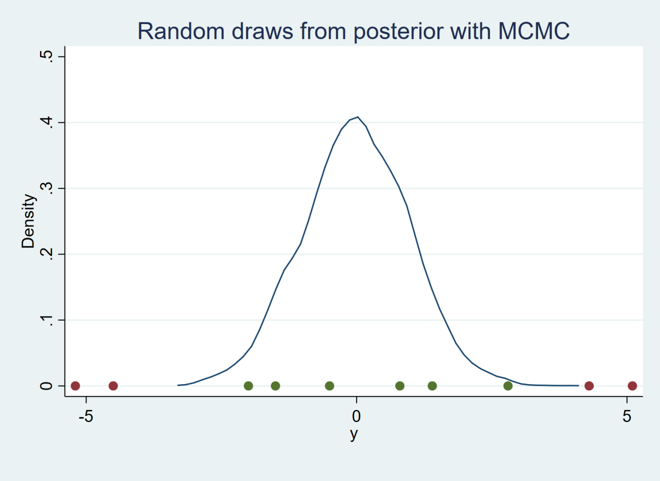
  - Markov chain Monte Carlo (MCMC) methods are the current standard in most software. Stata implement two alternatives:
    - Metropolis-Hastings (MH) algorithm
    - Gibbs sampling

The Method

- Metropolis-Hastings (MH) algorithm
  1. Specify a proposal probability distribution q(.)
  2. Set an initial state within the domain of the posterior distribution θ_0
  3. Propose a new state for the posterior distribution θ_t ; t=1,2,...
  4. Compute an aceptance rate based on the ratio of the posterior distribution evaluated at the proposed state θ_t and at the previous state θ_{t-1}.
  5. If the ratio is:
     - Greater than 1 –> keep the proposed value (state)
     - Less than one –> draw a random number from U(0,1) and keep θ_t if the ratio is greater than the random draw.
  6. Repeat the process from 3 with the selected θ_t

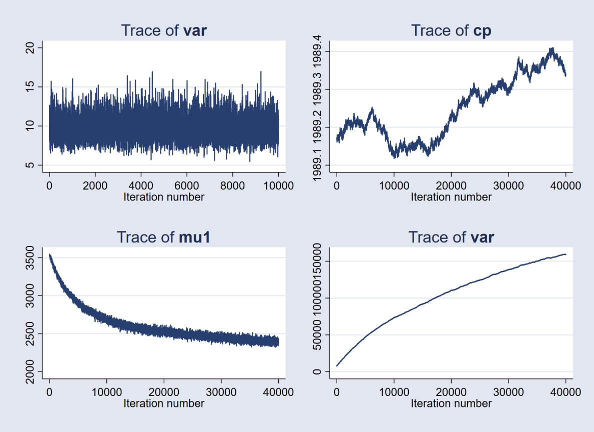
The Method

- Green points represent accepted proposal states and red points represent rejected proposal states.



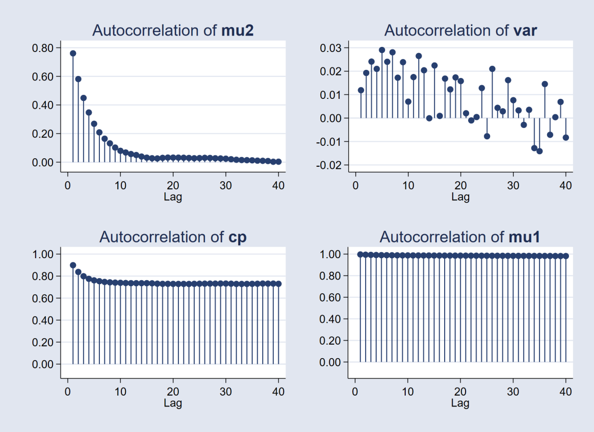
The Method

- The trace plot illustrates the sequence of accepted proposal states.
- We expect to obtain a stationary sequence when convergence is achieved.



The Method

- An efficient MCMC should have small autocorrelation.
- We expect autocorrelation to become negligible after a few lags.



The Stata tools for Bayesian regression

The Stata tools: `bayesmh`

- In Stata 14 we introduce `bayesmh`.
- This is a general purpose command to perform Bayesian analysis using MCMC (MH or Gibbs).
- We are going to work with a few examples to show different facilities available in Stata for the analysis.
- Let's look at our first example:
  - We have stats on number of wins by the Porto soccer team.
  - We fit a linear regression for yearly wins.
  - Let's consider three specifications:

$$\begin{aligned} wins &= \alpha_1 + \beta_{gs} * goals\_scored + \epsilon_1 \\ wins &= \alpha_2 + \beta_{ga} * goals\_against + \epsilon_2 \\ wins &= \alpha_3 + \beta_{gs2} * goals\_scored + \beta_{ga2} * goals\_against + \epsilon_3 \end{aligned}$$

The Stata tools: Regression with `bayesmh`

- Here is one syntax with `bayesmh` to fit this model:

```
bayesmh wins gs,likelihood(normal({sigma2}))    ///
    prior({wins:gs _cons}, normal(0,10000))      ///
    prior({sigma2}, igamma(.01,.01))             ///
    rseed(123)
```

- But let's use the Graphical User Interface (GUI) (Menus and dialog boxes):

The Stata tools: Menu for Bayesian regression

1. Make the following sequence of selection from the main menu:

   Statistics > Bayesian analysis
   > General estimation and regression
2. Select 'Univariate linear models'
3. Specify the dependent variable (wins) and the explanatory variable (gs)
4. Select the 'Likelihood model' (Normal regression)
   - For 'Variance' click on 'Create' and select 'Specify as a model parameter'
   - Type 'sigma2' in 'Parameter name'

The Stata tools: Menu for Bayesian regression

5. For "'Priors of model parameters' click on 'Create'
   - Select wins:gs and wins:_cons
   - Select the 'Normal distribution'
   - write '0' for the mean and '10000' for the variance.
6. Next, create the prior for the variance of the likelihood sigma2
   - Select the Inverse gamma distribution
   - Specify .01 and .01 for the 'Shape' and 'Scale' parameters.
7. Click on the 'Simulation' tab and set the 'Random-number seed' to 123

The Stata tools: Regression output

```
. bayesmh wins gs,likelihood(normal({sigma2}))    ///
>         prior({wins:gs _cons}, normal(0,10000)) ///
>         prior({sigma2}, igamma(.01,.01))        ///
>         rseed(123)

Burn-in ...
Simulation ...

Model summary
------------------------------------------------------------------
Likelihood:
  wins ~ normal(xb_wins,{sigma2})

Priors:
  {wins:gs _cons} ~ normal(0,10000)
         {sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_wins.
```

The Stata tools: Regression output

```
. bayesmh wins gs,likelihood(normal({sigma2}))    ///
>          prior({wins:gs _cons}, normal(0,10000)) ///
>          prior({sigma2}, igamma(.01,.01))        ///
>          rseed(123)

Bayesian normal regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =         47
                                                 Acceptance rate  =      .2222
                                                 Efficiency:  min =     .04521
                                                              avg =     .06161
Log marginal likelihood = -135.77023                          max =     .07185

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
wins         |
          gs |  .2360223   .0365801   .001405   .2363132    .162386   .3096086
       _cons |  6.711756   2.417745   .090197   6.704956   1.923868   11.53032
-------------+----------------------------------------------------------------
      sigma2 |  9.380877   2.089641   .098277   9.040789   6.262636   14.55403
------------------------------------------------------------------------------
```

The Stata tools: `bayesstats ess`

- Let's use the postestimation command `bayesstats ess` to evaluate the effective sample size

```
. bayesstats ess

Efficiency summaries    MCMC sample size =     10,000
```

	ESS	Corr. time	Efficiency
wins			
gs	677.68	14.76	0.0678
_cons	718.51	13.92	0.0719
sigma2	452.10	22.12	0.0452

- We expect to have an acceptance rate (see previous slide) that is neither to small nor too large.
- We also expect to have low correlation
- Efficiencies over 10% are considered good for MH. Efficiencies under 1% would be a source of concern.

The Stata tools: Blocking of parameters

- Blocking of parameters
  - The update steps for MH are performed simultaneously for all parameters.
  - For high dimensional models this may result in poor mixing.
  - Blocking of parameters helps improving mixing efficiency

The Stata tools: Blocking of parameters

- Blocking of parameters
  - How it works?
    - It separates the model parameters into two or more subsets of blocks.
    - MH updates are applied to each block separately
    - Computations are performed in the order the blocks are specified

```
bayesmh wins gs,likelihood(normal({sigma2}))   ///
    prior({wins:gs _cons}, normal(0,10000))    ///
    prior({sigma2}, igamma(.01,.01))           ///
    block({wins:gs _cons}) block({sigma2})     ///
    rseed(123)
```

The Stata tools: Menu for Blocking of parameters

Let's go back to our previous example:

1. Click on the 'Blocking' tab
2. Select 'Display block summary'
3. Click on 'Create'
4. Select wins:gs and wins:_cons and click 'OK'
5. Click on 'Create'
6. Select sigma2 and click 'OK'

The Stata tools: Blocking of parameters

```
. bayesmh wins gs,likelihood(normal({sigma2}))                                ///
>    prior({wins:gs _cons},normal(0,10000)) prior({sigma2},igamma(.01,.01)) ///
>    block({wins:gs _cons}) block({sigma2}) rseed(123) blocksummary

Burn-in ...
Simulation ...
Block summary
------------------------------------------------------------------------------
  1:   {wins:gs _cons}
  2:   {sigma2}
------------------------------------------------------------------------------

Bayesian normal regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =         47
                                                 Acceptance rate  =      .3426
                                                 Efficiency:  min =     .09882
                                                              avg =      .1156
Log marginal likelihood =  -135.7408                          max =      .1464
```

	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
wins						
gs	.2363963	.0373595	.001188	.2366527	.1626758	.3109461
_cons	6.690619	2.452853	.076957	6.69004	1.683672	11.63661
sigma2	9.392034	2.136876	.05585	9.09526	6.170093	14.36234

The Stata tools: `bayesstats ess`

- Let's evaluate again the effective sample size

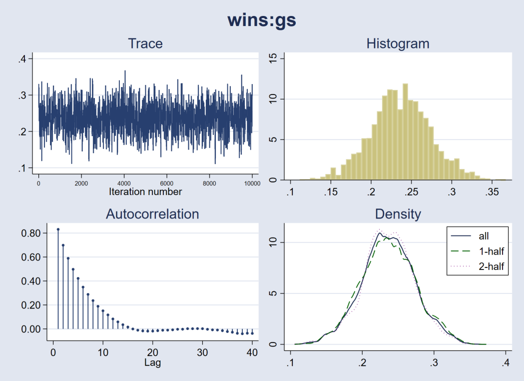
```
. bayesstats ess

Efficiency summaries    MCMC sample size =    10,000

------------------------------------------------------
             |        ESS   Corr. time    Efficiency
-------------+----------------------------------------
wins         |
          gs |     988.18        10.12        0.0988
       _cons |    1015.90         9.84        0.1016
-------------+----------------------------------------
      sigma2 |    1463.91         6.83        0.1464
------------------------------------------------------
```

- The efficiency is now around 10% or more for all the parameters.
- Correlation was reduced
- The effective sample size is also higher for all the parameters.

The Stata tools: `bayesgraph`

- We can use `bayesgraph` to look at the trace, the correlation, and the density. For example:

. bayesgraph diagnostic {gs}



- The trace indicates that convergence was achieved
- Correlation becomes negligible after 10 periods

The Stata tools: `bayes:` prefix

- In Stata 15 we introduce the prefix command `bayes:`
- This is a simple syntax to perform Bayesian analysis.
- You specify the prefix followed by your estimation command.
- The specified estimation defines the likelihood for the model.
- The default priors are assumed to be noninformative in many cases.
- But the priors may become informative due to the scale of the parameters.
- The default priors could be consider a starting point.
- However, alternative priors may need to be considered.
- Postestimation commands would help decide on the final model.
- Let's use `bayes:` to fit our previous model:

The Stata tools: `bayes:` prefix

```
. bayes,rseed(123) nomodelsummary: regress wins gs

Burn-in ...
Simulation ...
Bayesian linear regression                    MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling      Burn-in          =      2,500
                                              MCMC sample size =     10,000
                                              Number of obs    =         47
                                              Acceptance rate  =      .3426
                                              Efficiency:  min =     .09882
                                                           avg =      .1156
Log marginal likelihood =  -135.7408                       max =      .1464
```

	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
wins						
gs	.2363963	.0373595	.001188	.2366527	.1626758	.3109461
_cons	6.690619	2.452853	.076957	6.69004	1.683672	11.63661
sigma2	9.392034	2.136876	.05585	9.09526	6.170093	14.36234

Note: Default priors are used for model parameters.

The Stata tools: `bayesstats ic`

- Let's fit now the other two models that we specify at the beginning of this example.
- We will store the results for the three models and we will use the postestimation command `bayesstats ic` to select one of them.

```
quietly {
    bayes , rseed(123): regress wins gs
    estimates store m_gs

    bayes , rseed(123): regress wins ga
    estimates store m_ga

    bayes , rseed(123): regress wins gs ga
    estimates store m_full
}
bayesstats ic m_gs m_ga m_full,basemodel(m_ga)
```

The Stata tools: `bayesstats ic`

- `bayesstats ic` reports three statistics
  - Log of the marginal likelihood
  - DIC:
    - It is designed for Bayesian estimation involving MCMC simulations.
    - It Has a penalty term based on the difference between the expected log likelihood and the likelihod at the posterior mean point.
    - You shoud select the model with the lowest DIC.
  - Bayes factors (BF)
    - Incorporates information about model priors.
    - Ratio of the marginal likelihood of two models (fit on the same sample).
    - It can be used to compare nested and nonnested models.
    - Not applicable to models with improper priors.

The Stata tools: `bayesstats ic`

- Here is the output for `bayesstats ic`

```
. quietly {
. bayesstats ic m_gs m_ga m_full,basem(m_full) bayesf
Bayesian information criteria
```

	DIC	log(ML)	BF
m_gs	240.6314	-135.7408	5.015833
m_ga	268.5267	-148.5384	.0000139
m_full	230.9162	-137.3534	.

```
Note: Marginal likelihood (ML) is computed
      using Laplace-Metropolis approximation.
```

- Interpretation for Bayes Factors (Jeffreys 1961)

log10(BF_jb)	BF_jb	Evidence against M_b
0 to 1/2	1 to 3.2	Bare mention
1/2 to 1	3.2 to 10	Substantial
1 to 2	10 to 100	Strong
> 2	> 100	Decisive

The Stata tools: `bayestest model`

- `bayestest model` is another postestimation command to compare different models.
- We can again store the results for our alternative models, and then use `bayestest model`.

```
quietly {
    bayes , rseed(123): regress wins gs
     estimates store m_gs

     bayes , rseed(123): regress wins ga
     estimates store m_ga

     bayes , rseed(123): regress wins gs ga
     estimates store m_full

     bayes , prior({wins:gs _cons}, normal(20,10)) ///
              rseed(123): regress wins
     estimates store m_meanonly
 }
 bayestest model m_gs m_ga m_full m_meanonly
```

The Stata tools: `bayestest model`

- `bayestest model` computes the posterior probabilities for each model.
- The result indicates which model is more likely.
- It requires that the models use the same data and that they have proper posterior.
- It can be used to compare models with:
  - Different priors and/or different posterior distributions.
  - Different regression functions.
  - Different covariates
- MCMC convergence should be verified before comparing the models.

The Stata tools: `bayestest model`

- Here is the output for `bayestest model`

```
. bayestest model m_gs m_ga m_full m_meanonly
```

	log(ML)	P(M)	P(M|y)
m_gs	-135.7408	0.2500	0.8211
m_ga	-148.5384	0.2500	0.0000
m_full	-137.3534	0.2500	0.1637
m_meanonly	-139.7326	0.2500	0.0152

Note: ML is computed using Laplace-Metropolis approximation.

- We could also assign different priors for the models:

```
. bayestest model m_gs m_ga m_full m_meanonly, ///
        prior(.2 .1 .4 .3)
```

	log(ML)	P(M)	P(M|y)
m_gs	-135.7408	0.2000	0.7010
m_ga	-148.5384	0.1000	0.0000
m_full	-137.3534	0.4000	0.2795
m_meanonly	-139.7326	0.3000	0.0194

Note: ML is computed using Laplace-Metropolis approximation.

Random Effects Probit model

The Stata tools: Random effects probit model

- We are going to use `bayes:` to fit a random effects probit model for a binary variable y_{it}, which depends on the latent variable .

$$y^*_{it} = \beta_0 + \beta_1 x_{it1} + \beta_2 x_{it2} + \ldots + \beta_k x_{itk} + \alpha_i + \epsilon_{it}$$

Where:

$$y_{it} = \begin{cases} 1 & \text{if } y^*_{it} > 0 \\ 0 & \text{otherwise} \end{cases}$$

$\alpha_i \sim N\left(0, \sigma^2_\alpha\right)$ is the individual random panel effect

$\epsilon_{it} \sim N\left(0, \sigma^2_e\right)$ is the idiosyncratic error term

- This is also referred as a two-level random intercept model.
- We can also fit this model with `meprobit` or `xtprobit,re`

The Stata tools: Random effects probit model

- This time we are going to work with simulated data.
- Here is the code to simulate the panel dataset:

```
  clear
  set obs 100
  set seed 1

* Panel level *
  generate id=_n
  generate alpha=rnormal()
  expand 5

* Observation level *
  bysort id:generate year=_n
  xtset id year
  generate x1=rnormal()
  generate x2=runiform()>.5
  generate x3=uniform()
  generate u=rnormal()

* Generate dependent variable *
  generate y=.5+1*x1+(-1)*x2+1*x3+alpha+u>0
```

The Stata tools: Random effects probit model

Let's show the results with `meprobit`:

```
. meprobit y x1 x2 x3 || id:,nolog
Mixed-effects probit regression                 Number of obs     =        500
Group variable:                 id              Number of groups  =        100

                                                Obs per group:
                                                              min =          5
                                                              avg =        5.0
                                                              max =          5

Integration method: mvaghermite                 Integration pts.  =          7

                                                Wald chi2(3)      =      82.83
Log likelihood = -236.88589                     Prob > chi2       =     0.0000
------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .9769118   .1143889      0.000     .7527138     1.20111
          x2 |  -.9896286   .1853433      0.000    -1.352895   -.6263625
          x3 |   .9426958   .2941061      0.001     .3662584    1.519133
       _cons |   .5220418   .2187448      0.017     .0933098    .9507738
-------------+----------------------------------------------------------------
id           |
   var(_cons)|       1.31   .3835866                 .7379508    2.325494
------------------------------------------------------------------------------
LR test vs. probit model: chibar2(01) = 67.24    Prob >= chibar2 = 0.0000
```

The Stata tools: Random effects probit model

We now fit the model with `bayes:`

```
. bayes , dryrun: meprobit y x1 x2 x3 || id:
Multilevel structure
------------------------------------------------------------------------
id
    {U0}: random intercepts
------------------------------------------------------------------------

Model summary
------------------------------------------------------------------------
Likelihood:
  y ~ meprobit(xb_y)

Priors:
  {y:x1 x2 x3 _cons} ~ normal(0,10000)                              (1)
                {U0} ~ normal(0,{U0:sigma2})                        (1)
.
Hyperprior:
  {U0:sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_y.
```

The Stata tools: Random effects probit model

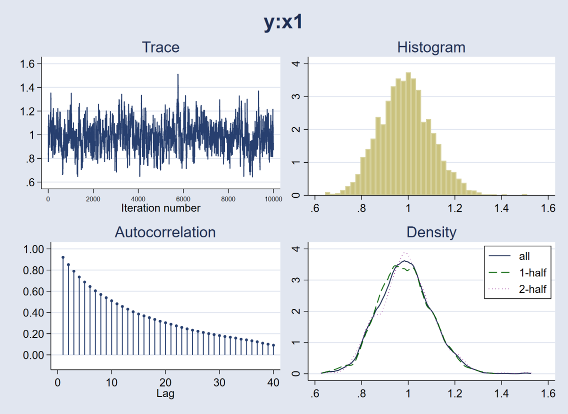
We now fit the model with `bayes`:

```
. bayes ,nomodelsummary nodots rseed(123): meprobit y x1 x2 x3 || id:
Burn-in ...
Simulation ...
Multilevel structure
------------------------------------------------------------------------------
id
    {U0}: random intercepts
------------------------------------------------------------------------------

Bayesian multilevel probit regression            MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
Group variable: id                               Number of groups =        100
                                                 Obs per group:
                                                              min =          5
                                                              avg =        5.0
                                                              max =          5
Family : Bernoulli                               Number of obs    =        500
Link   : probit                                  Acceptance rate  =      .3247
                                                 Efficiency:  min =     .01333
                                                              avg =     .02736
Log marginal likelihood                                       max =     .04012

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
y            |
          x1 |  .9866518   .1129356   .006316   .9850336   .7789124   1.215904
          x2 | -1.005328   .1793814   .009673  -1.003398  -1.357951  -.6617393
          x3 |  .9856235   .2968089   .014819   .9666234   .4282133   1.591159
       _cons |  .5051288   .2055344   .017802   .5032979   .0933563    .889766
-------------+----------------------------------------------------------------
id           |
   U0:sigma2 |  1.432124   .4234419   .032504   1.388553   .7326054   2.388284
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
```

The Stata tools: `bayesgraph diagnostic`

- We can look at the diagnostic graph for a couple of variables:

. bayesgraph diagnostic {y:x1}

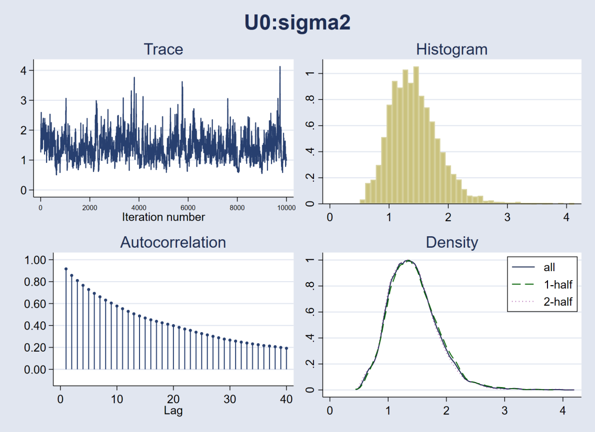


- The trace shows periods with trends.
- Correlation is persistent for around 25 periods.

The Stata tools: `bayesgraph diagnostic`

- Look now at the diagnostic graphs for U0:sigma2

. bayesgraph diagnostic {U0:sigma2}



- The trace also shows periods with trends.
- Correlation is persistent for around 30 periods.

The Stata tools: thinning

- We can reduce autocorrelation by using thinning
- This would save the random draws skipping a prespecified number of simulated values in the iteration process for the MCMC.
- We can use the option 'thinning(#)' to indicate that Stata should save simulated values from every (1+k*#)th iteration (k=0,1,2,...).
- Let's try using 'thinning(5)'

```
bayes ,nomodelsummary nodots rseed(123) ///
    thinning(5): meprobit y x1 x2 x3 || id:
```

The Stata tools: thinning

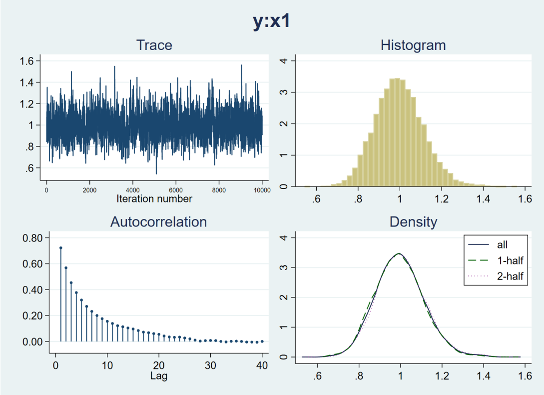
Let's show the results with 'thinning(5)'

```
. bayes,nomodelsummary nodots rseed(123) thinning(5):meprobit y x1 x2 x3 || id:
note: discarding every 4 sample observations; using observations 1,6,11,...

Burn-in ...
Simulation ...
Multilevel structure
------------------------------------------------------------------------------
id
    {U0}: random intercepts
------------------------------------------------------------------------------

Bayesian multilevel probit regression          MCMC iterations  =     52,496
Random-walk Metropolis-Hastings sampling       Burn-in          =      2,500
                                               MCMC sample size =     10,000
Group variable: id                             Number of groups =        100
                                               Obs per group:
                                                            min =          5
                                                            avg =        5.0
                                                            max =          5
Family : Bernoulli                             Number of obs    =        500
Link   : probit                                Acceptance rate  =      .3268
                                               Efficiency:  min =     .05399
                                                            avg =       .102
Log marginal likelihood                                     max =      .1628

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
y            |
          x1 |  .9977099   .1181726   .003773   .9936143   .7810441   1.242439
          x2 | -1.018063   .1892596    .00557  -1.012598  -1.396798  -.6509636
          x3 |  .9539304   .2936949   .007279   .9514395   .3823801    1.52913
       _cons |  .5433822   .2205077    .00949   .5398387   .1216346   .9847166
-------------+----------------------------------------------------------------
id           |
   U0:sigma2 |  1.456558   .4384163   .015537   1.401461   .7611919   2.463175
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
```

bayesgraph diagnostic

- We now look at the diagnostic graph for the same two variables:

. bayesgraph diagnostic {y:x1}

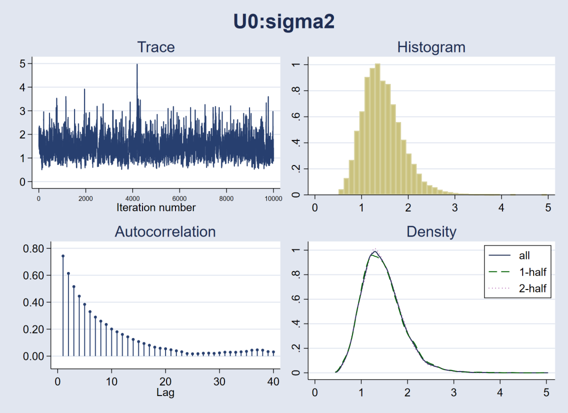


- The trace seems to indicate convergence this time.
- Autocorrelation decays quicker and becomes negligible after about 15 periods.

The Stata tools: `bayesgraph diagnostic`

- We now look now at the diagnostic graphs for U0:sigma2

```
. bayesgraph diagnostic {U0:sigma2}
```



- The trace seems to indicate convergence this time.
- Autocorrelation decays quicker and becomes negligible after about 15 periods.

The Stata tools: `bayestest interval`

- We can perform interval testing with the postestimation command `bayestest interval`.
- It estimates the probability that a model parameter lies in a particular interval.
- For continuous parameters the hypothesis is formulated in terms of intervals.
- We can perform point hypothesis testing only for parameters with discrete posterior distributions.
- `bayestest interval` estimates the posterior distribution for a null interval hypothesis.
- `bayestest interval` reports the estimated posterior mean probability for Ho.

```
bayestest interval ({y:x1},lower(.9) upper(1.02)) ///
                   ({y:x2},lower(-1.1) upper(-.8))
```

The Stata tools: `bayestest interval`

- We can, for example, perform separate tests for different parameters:

```
. bayestest interval ({y:x1},lower(.9) upper(1.02)) ///
>                     ({y:x2},lower(-1.1) upper(-.8))
Interval tests     MCMC sample size =    10,000

       prob1 :   .9 < {y:x1} < 1.02
       prob2 : -1.1 < {y:x2} < -.8
```

	Mean	Std. Dev.	MCSE
prob1	.3888	0.48750	.0077073
prob2	.5474	0.49777	.0097517

- We can also perform a joint test:

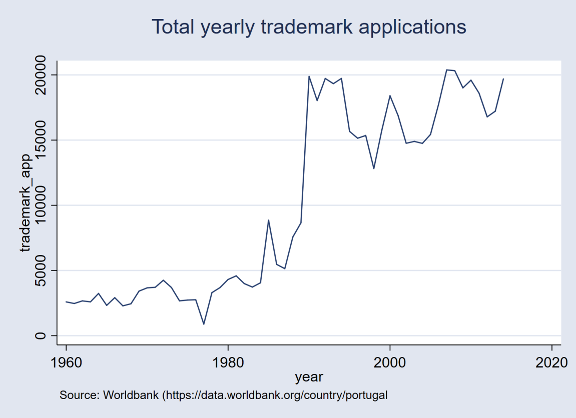
```
. bayestest interval (({y:x1},lower(.9) upper(1.02)) ///
>                 ({y:x2},lower(-1.1) upper(-.8)),joint)
Interval tests     MCMC sample size =    10,000

   prob1 : .9 < {y:x1} < 1.02, -1.1 < {y:x2} < -.8
```

	Mean	Std. Dev.	MCSE
prob1	.2249	0.41754	.0066399

Change-point model

The Stata tools: Change-point model

- Let's work now with an example where we write our model using a substitutable expression.
- We have data on yearly trademark applications in portugal:



- The series has a significant change around 1990.
- We may consider fitting a change-point model.

The Stata tools: Change-point model

Change point model specification

```
bayesmh                                                          ///
  trdmark=({mu1}*sign(year<{cp})+{mu2}*sign(year>={cp})), ///
  likelihood(normal({var}))                         ///
  prior({mu1}, normal(3000,2000000))                ///
  prior({mu2}, normal(16000,2000000))               ///
  prior({cp}, uniform(1960,2016))                   ///
  prior({var}, igamma(2,1))                         ///
  initial({mu1} 5000 {mu2} 10000 {cp} 1960)        ///
  rseed(123) mcmcsize(40000)                       ///
  dots(500,every(5000))                            ///
  title(Change-point analysis)
```

The Stata tools: Change-point model

Change point model specification

```
. bayesmh trdmark=({mu1}*sign(year<{cp})+{mu2}*sign(year>={cp})), ///
>         likelihood(normal({var}))                                ///
>         prior({mu1}, normal(3000,2000000))                       ///
>         prior({mu2}, normal(16000,2000000))                      ///
>         prior({cp}, uniform(1960,2016))                          ///
>         prior({var}, igamma(2,1))                                ///
>         initial({mu1} 5000 {mu2} 10000 {cp} 1960)                ///
>         rseed(123) mcmcsize(40000) dots(500,every(5000))         ///
>         title(Change-point analysis)

Burn-in 2500 aaaaa done
Simulation 40000 .........5000.........10000.........15000.........20000
> .........25000.........30000.........35000.........40000 done
Model summary
----------------------------------------------------------------------------
Likelihood:
  trdmark ~ normal({mu1}*sign(year<{cp})+{mu2}*sign(year>={cp}),{var})

Priors:
  {var} ~ igamma(2,1)
  {mu1} ~ normal(3000,2000000)
  {mu2} ~ normal(16000,2000000)
   {cp} ~ uniform(1960,2016)
----------------------------------------------------------------------------
```

The Stata tools: Change-point model

Change point model specification

```
. bayesmh trdmark=({mu1}*sign(year<{cp})+{mu2}*sign(year>={cp})), ///
>         likelihood(normal({var}))                             ///
>         prior({mu1}, normal(3000,2000000))                     ///
>         prior({mu2}, normal(16000,2000000))                    ///
>         prior({cp}, uniform(1960,2016))                        ///
>         prior({var}, igamma(2,1))                              ///
>         initial({mu1} 5000 {mu2} 10000 {cp} 1960)              ///
>         rseed(123) mcmcsize(40000) dots(500,every(5000))       ///
>         title(Change-point analysis)

Change-point analysis                          MCMC iterations  =     42,500
Random-walk Metropolis-Hastings sampling       Burn-in          =      2,500
                                               MCMC sample size =     40,000
                                               Number of obs    =         55
                                               Acceptance rate  =      .4117
                                               Efficiency:  min =    .001033
                                                            avg =     .03796
Log marginal likelihood = -621.28408                        max =      .1362
```

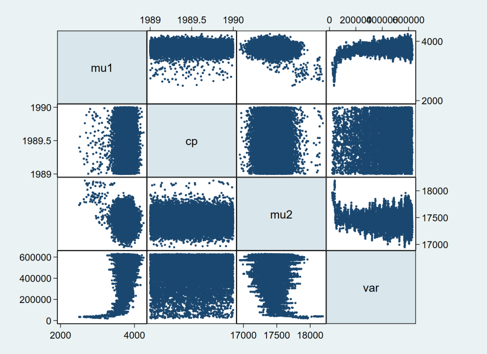
	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
cp	1989.492	.2891978	.003918	1989.492	1989.023	1989.972
mu1	3754.837	153.0364	11.9209	3761.923	3468.338	4015.751
mu2	17448.84	144.531	7.04777	17448.23	17170.98	17736.22
var	463983.1	144106.8	22418.1	487445.9	89224.3	621052.3

```
Note: There is a high autocorrelation after 500 lags.
Note: Adaptation tolerance is not met.
```

The Stata tools: `bayesgraph matrix`

- We can use `bayesgraph matrix` to look at the scatterplots for the simulated values of the coefficients and the variance.
- This may be useful to identify pairwise correlations that could suggest blocking for some of the parameters.

. bayesgraph matrix {mu1} {cp} {mu2} {var}

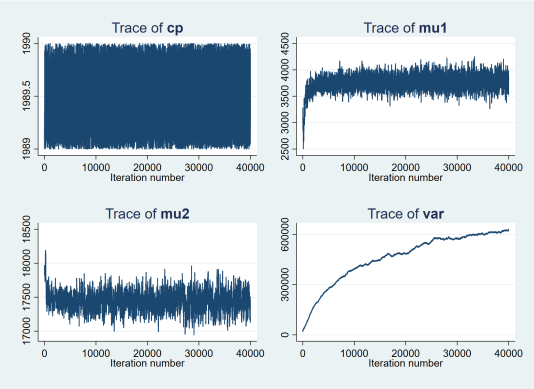


- We observe pairwise correlations for {mu1}, {mu2} and {var}
- Then, we could perform the MCMC for those three parameters as a block and {cp} in a second block. .

The Stata tools: `bayesgraph trace`

- We can use `bayesgraph trace` to look at the trace for all the parameters.
- This helps in determining convergence.

. bayesgraph trace

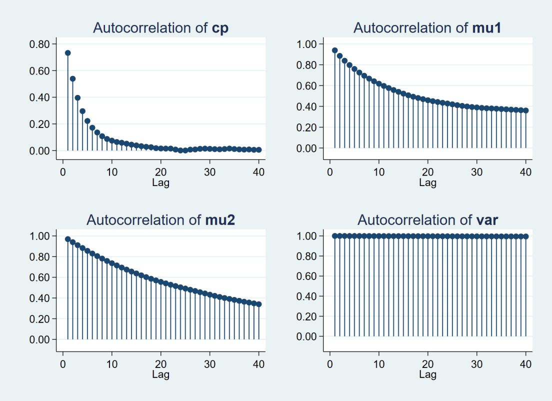


- We observe signs of lack of convergence, particularly for the variance.

The Stata tools: `bayesgraph ac`

- We can use `bayesgraph ac` to look at the autocorrelation for all the parameters.
- This also helps in determining convergence.

. bayesgraph ac



- The plot shows autocorrelation for almost all the parameters.

The Stata tools: Change-point model with MCMC Blocking

Change point model specification with blocking

```
bayesmh                                                              ///
  trdmark=({mu1}*sign(year<{cp})+{mu2}*sign(year>={cp})), ///
  likelihood(normal({var}))                                  ///
  prior({mu1}, normal(3000,2000000))                         ///
  prior({mu2}, normal(10000,2000000))                        ///
  prior({cp}, uniform(1960,2016))                            ///
  prior({var}, igamma(2,1))                                  ///
  initial({mu1} 5000 {mu2} 10000 {cp} 1960)                  ///
  block({var}, gibbs) block({cp}) blocksummary               ///
  rseed(123) mcmcsize(40000)                                 ///
  dots(500,every(5000))                                      ///
  title(Change-point analysis)
```

The Stata tools: Change-point model with MCMC Blocking

Change point model specification with blocking

```
. bayesmh trdmark=({mu1}*sign(year<{cp})+{mu2}*sign(year>={cp})), ///
>         likelihood(normal({var}))                                 ///
>         prior({mu1}, normal(3000,2000000))                        ///
>         prior({mu2}, normal(16000,2000000))                       ///
>         prior({cp}, uniform(1960,2016))                           ///
>         prior({var}, igamma(2,1))                                 ///
>         initial({mu1} 5000 {mu2} 10000 {cp} 1960)                 ///
>         block({var}, gibbs) block({cp}) blocksummary               ///
>         rseed(123) mcmcsize(40000) dots(500,every(5000))          ///
>         title(Change-point analysis)

Burn-in 2500 aaaaa done
Simulation 40000 .........5000.........10000.........15000.........20000
> .........25000.........30000.........35000.........40000 done
Model summary
------------------------------------------------------------------------
Likelihood:
  trdmark ~ normal({mu1}*sign(year<{cp})+{mu2}*sign(year>={cp}),{var})

Priors:
  {var} ~ igamma(2,1)
  {mu1} ~ normal(3000,2000000)
  {mu2} ~ normal(16000,2000000)
   {cp} ~ uniform(1960,2016)
------------------------------------------------------------------------

Block summary
------------------------------------------------------------------------
   1:  {var}                                                     (Gibbs)
   2:  {cp}
   3:  {mu1} {mu2}
------------------------------------------------------------------------
```

The Stata tools: Change-point model with MCMC Blocking

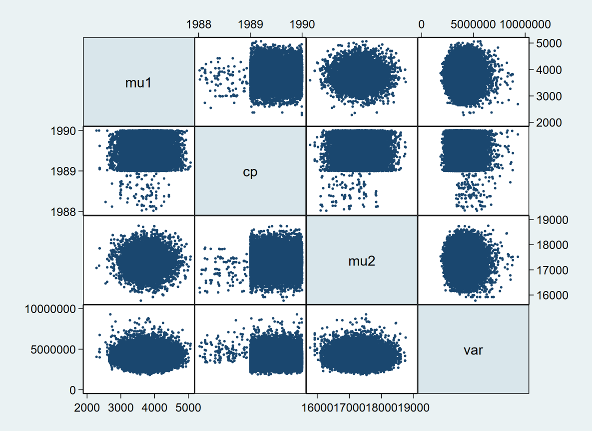
Change point model specification with blocking

```
. bayesmh trdmark=({mu1}*sign(year<{cp})+{mu2}*sign(year>={cp})),     ///
>         likelihood(normal({var}))                                  ///
>         prior({mu1}, normal(3000,2000000))                         ///
>         prior({mu2}, normal(16000,2000000))                        ///
>         prior({cp}, uniform(1960,2016))                            ///
>         prior({var}, igamma(2,1))                                  ///
>         initial({mu1} 5000 {mu2} 10000 {cp} 1960)                      ///
>         block({var}, gibbs) block({cp}) blocksummary                  ///
>         rseed(123) mcmcsize(40000) dots(500,every(5000))    ///
>         title(Change-point analysis)

Change-point analysis                             MCMC iterations  =     42,500
Metropolis-Hastings and Gibbs sampling            Burn-in          =      2,500
                                                  MCMC sample size =     40,000
                                                  Number of obs    =         55
                                                  Acceptance rate  =      .5288
                                                  Efficiency:  min =     .07912
                                                               avg =      .2638
Log marginal likelihood = -533.33098                           max =      .6717

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
          cp |  1989.496   .2944166   .003126   1989.496   1989.019   1989.975
         mu1 |   3780.26   341.1711   6.06443   3783.149   3108.712   4446.395
         mu2 |  17332.57   372.1327   6.47794   17344.63    16588.7   18068.87
         var |   3798272   739589.8   4512.18    3708037    2612399    5480970
------------------------------------------------------------------------------
```

The Stata tools: `bayesgraph matrix`

- We check the scatterplots again for the simulated values of the coefficients and the variance.

. bayesgraph matrix

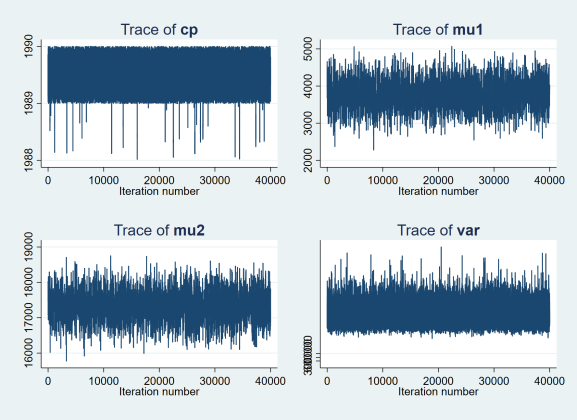


- We do not observe any pairwise correlations now.

The Stata tools: `bayesgraph trace`

- We can use `bayesgraph trace` to look at the trace for all the parameters.

. bayesgraph trace

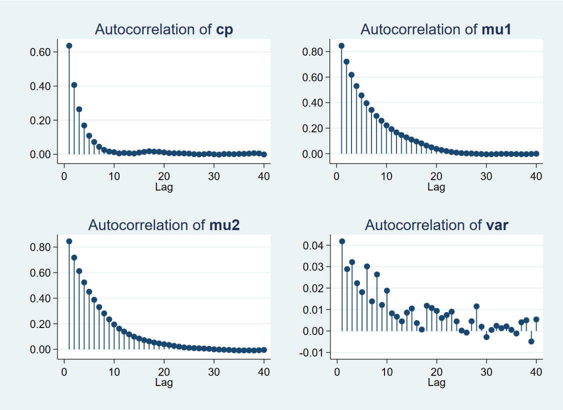


- The plots indicate that convergence seems to be achieved.

The Stata tools: `bayesgraph ac`

- We can also use `bayesgraph ac` to look at the autocorrelation for all the parameters.

. bayesgraph ac



- Autocorrelation decays and becomes negligible quickly for almost all the parameters.

Summary

1. Bayesian analysis: The general idea
2. Basic Concepts
    - The Method
    - The tools
    - Stata 14: The `bayesmh` command
    - Stata 15: The `bayes` prefix
    - Postestimation commands
3. A few examples
    - Linear regression
    - Panel data random effect probit model
    - Change point model

References

Glinsky, M. E. and Gunnin, J. 2011. *Understanding uncertainty in GSEM*. World Oil, 232(1), 57—62, Jan. http://www.worldoil.com/Understanding-uncertainty-in-CSEM-January-2011.html, assessed-Jan.18,2012.

Hahn, Eugene D. 2014. *Bayesian Methods for Management and Business: Pragmatic Solutions for Real Problems*. John Wiley and Sons.

Shamdasany, P. 2011. *Smart Money*. South China Morning Post. p. 2., Jul. 4.