Introduction to Bayesian Analysis in Stata

Gustavo Sánchez

StataCorp LLC

January 24, 2019
College Station, Texas

Outline

1. Bayesian analysis: Basic concepts
   - The general idea
   - The method
2. The Stata tools
   - The general command `bayesmh`
   - The `bayes` prefix
   - Postestimation commands
3. A few examples
   - Probit regression
   - Panel data random-effects Poisson model
   - Change-point model

The general idea

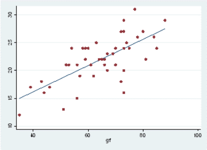
Frequentist

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

→

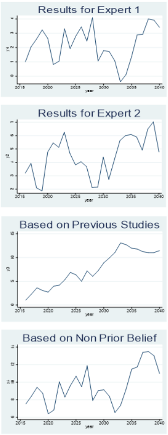


The general idea

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



Bayesian Analysis vs Frequentist Analysis

Frequentist Analysis

- Estimates unknown fixed parameters.
- The data come from a random sample (hypothetical repeatable).
- Uses data to estimate unknown fixed parameters.
- Data expected to satisfy the assumptions for the specified model.

"Conclusions are based on the distribution of statistics derived from random samples, assuming unknown but fixed parameters."

Bayesian Analysis

- Probability distributions for unknown random parameters.
- The data are fixed.
- Combines data with prior beliefs to get updated probability distributions for the parameters.
- Posterior distribution is used to make explicit probabilistic statements.

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

Stata's convenient syntax: `bayes:`

```
regress y x1 x2 x3

bayes: regress y x1 x2 x3
```

```
logit y x1 x2 x3

bayes: logit y x1 x2 x3
```

```
mixed y x1 x2 x3 || region:

bayes: mixed y x1 x2 x3 || region:
```

The method

The method

- Inverse law of probability (Bayes' Theorem):

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

Where:

$f(y;\theta)$: probability density function for y given θ.

$\pi(\theta)$: prior distribution for θ

- The marginal distribution of y, f(y), does not depend on θ; then we can write the fundamental equation for Bayesian analysis:

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

Where:

$L(\theta|y)$: likelihood function of the parameters given the data.

The method

- Inverse law of probability (Bayes' Theorem):

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

Where:

$f(y;\theta)$: probability density function for y given θ.

$\pi(\theta)$: prior distribution for θ

- The marginal distribution of y, f(y), does not depend on θ; then we can write the fundamental equation for Bayesian analysis:

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

Where:

$L(\theta|y)$: likelihood function of the parameters given the data.

The method

- Inverse law of probability (Bayes' Theorem):

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

Where:

$f(y;\theta)$: probability density function for y given θ.

$\pi(\theta)$: prior distribution for θ

- The marginal distribution of y, f(y), does not depend on θ; then we can write the fundamental equation for Bayesian analysis:

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

Where:

$L(\theta|y)$: likelihood function of the parameters given the data.

The method

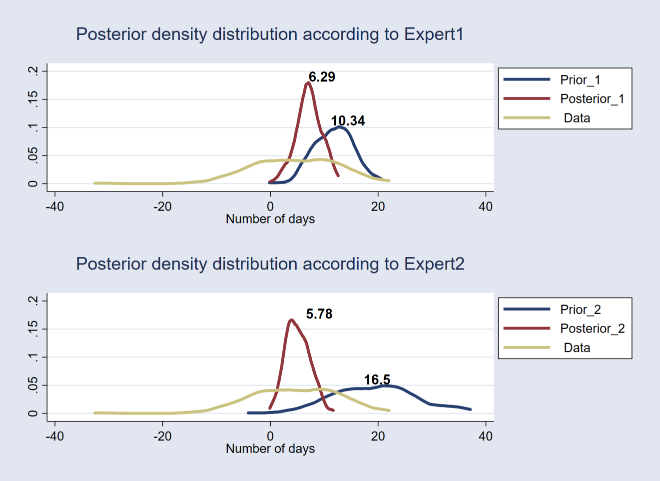
- Let's assume that both the data and the prior beliefs are normally distributed:
  - **The data**: $y \sim N\left(\theta, \sigma_d^2\right)$
  - **The prior**: $\theta \sim N\left(\mu_p, \sigma_p^2\right)$
- Homework...: Doing the algebra with the fundamental equation, we find that the posterior distribution would be normal with (see for example Cameron & Trivedi 2005):
  - **The posterior**: $\theta|y \sim N\left(\mu, \sigma^2\right)$

  Where:

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

Example (Prior distributions)

Example (Posterior distributions)



The method

- The previous example has a closed form solution.
- What about the cases with non-closed solutions, or 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 implements two alternatives:
    - Metropolis–Hastings (MH) algorithm
    - Gibbs sampling

The method

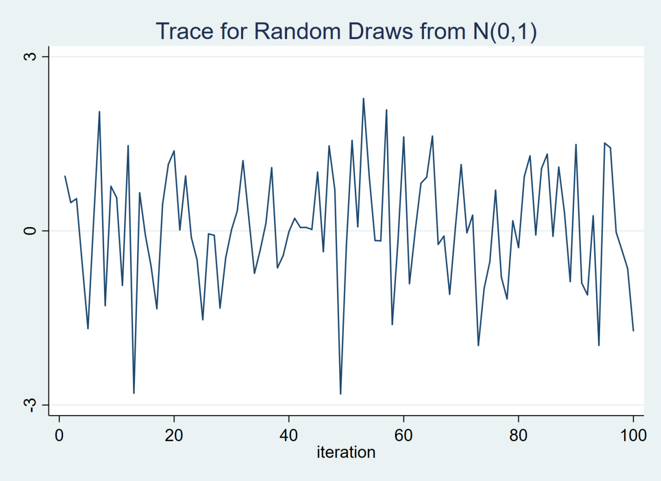
- Links for Bayesian analysis and MCMC on our YouTube channel:
  - Introduction to Bayesian statistics, part 1: The basic concepts

    https://www.youtube.com/watch?v=0F0QoMCSKJ4&feature=youtu.be
  - Introduction to Bayesian statistics, part 2: MCMC and the Metropolis–Hastings algorithm.

    https://www.youtube.com/watch?v=OTO1DygELpY&feature=youtu.be

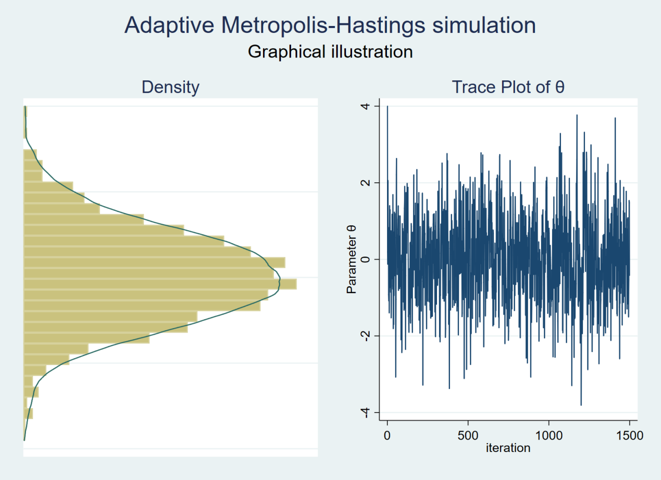
The method

- Monte Carlo Simulation



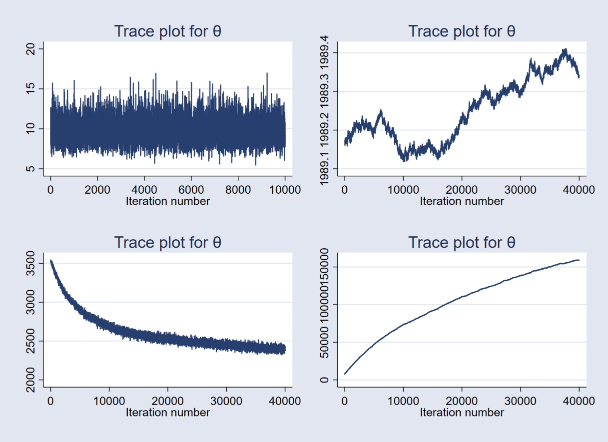
The method

- Metropolis–Hastings simulation
  - The trace plot illustrates the sequence of accepted proposal states.



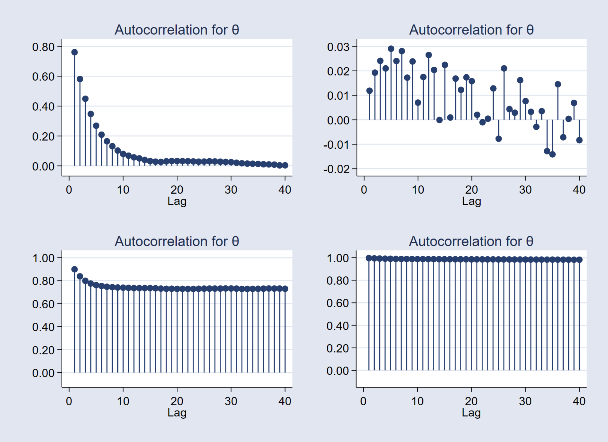
The method

- 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: `bayes:` `bayesmh`

- `bayes:` Convenient syntax for Bayesian regressions
    - Estimation command defines the likelihood for the model.
    - Default priors are assumed to be "weakly informative"'.
    - Other model specifications are set by default depending on the model defined by the estimation command.
    - Alternative specifications may need to be evaluated.

- `bayesmh` General purpose command for Bayesian analysis
    - You need to specify all the components for the Bayesian regression: likelihood, priors, hyperpriors, blocks, etc.

The Stata tools: Postestimation commands

- bayesstats ess
- bayesgraph
- bayesstats ic
- bayestest model
- bayestest interval
- bayesstats summary

Examples

Example 1: Probit regression

- Let's look at our first example:
  - We have stats on scores, strength of schedule, and bowl game result (win/loss) for the Texas A&M University football team.
  - We fit a probit model for the probability to win the bowl game.
  - Let's consider a couple of model specifications for a binary dependent variable, whose values depend on a linear latent variable:

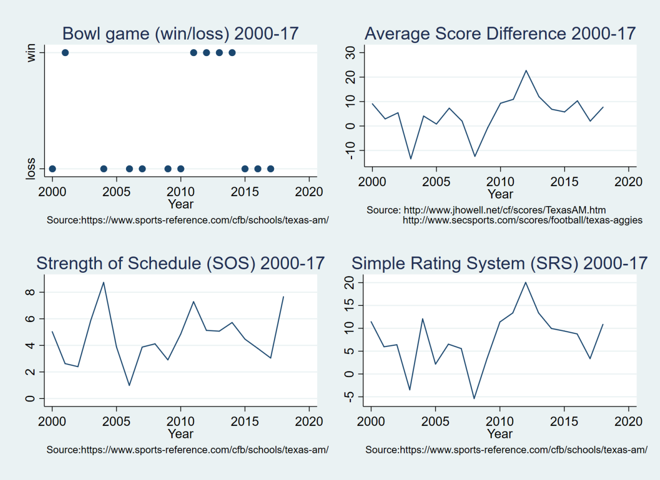
$$
\begin{aligned}
win\_bowl^* &= \alpha_1 + \beta_{sc\_dif} * score\_dif + \beta_{sos} * sos + \epsilon_1 \\
win\_bowl^* &= \alpha_2 + \beta_{scored} * score\_avg + \beta_{against} * against\_avg + \epsilon_2
\end{aligned}
$$

$$
win\_bowl = \begin{cases} 1 & \text{if } win\_bowl^* > 0 \\ 0 & \text{otherwise} \end{cases}
$$

Where:

win_bowl	: result in the bowl game (winloss).
score_dif	: Average score difference during the regular season.
sos	: Strength of schedule.
score_avg	: Average points scored during the regular season.
against_avg	: Average points against during the regular season.

Example 1: Probit regression



Example 1: Probit regression

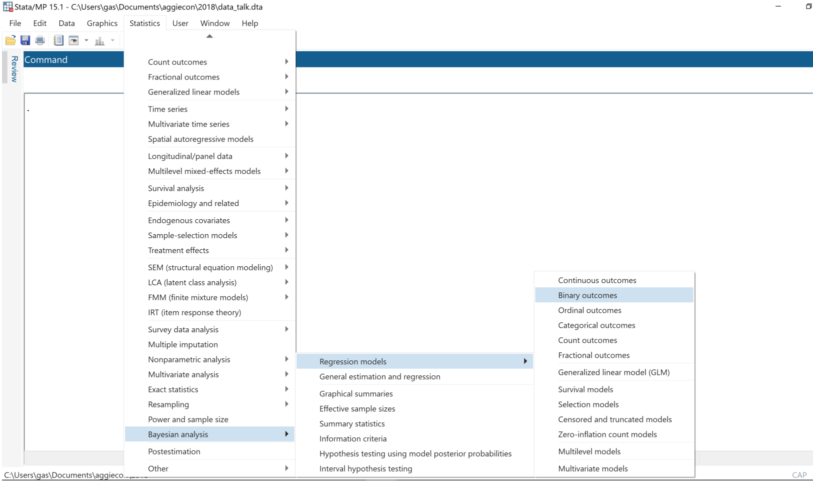
- Probit regression with the `bayes:` prefix

```
bayes, rseed(123): probit win_bowl score_diff sos
```

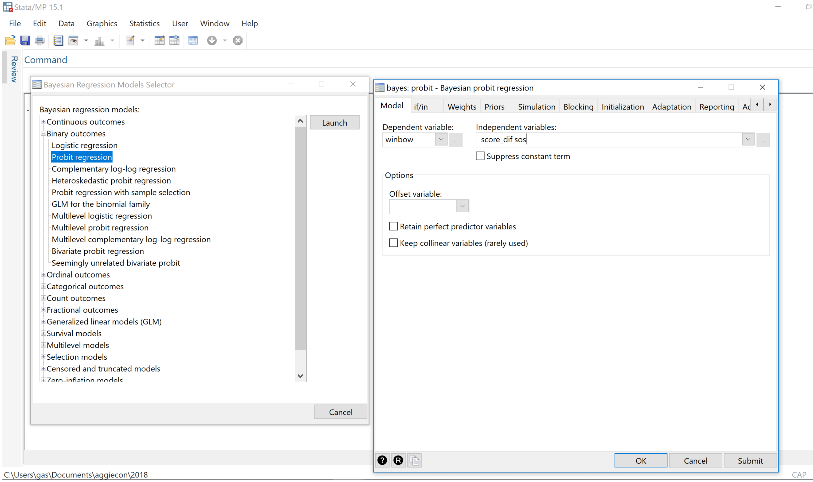
- Equivalent model with `bayesmh`

```
bayesmh win_bowl score_diff sos, rseed(123)       ///
    likelihood(probit)                            ///
    prior({win_bowl:score_diff}, normal(0,10000)) ///
    prior({win_bowl:sos}, normal(0,10000))        ///
    prior({win_bowl:_cons}, normal(0,10000))
```

Example 1: Menu for Bayesian regression



Example 1: Menu for Bayesian regression



Example 1: Menu for Bayesian regression

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

   Statistics > Bayesian analysis > Regression models
2. Select "Binary outcomes"
3. Select "Probit regression"
4. Click on "Launch"
5. Specify the dependent variable (win_bowl) and the explanatory variables (score_dif sos)
6. Click on "OK"

Example 1: bayes: prefix

```
. bayes, rseed(123):probit win_bowl score_dif sos

Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  win_bowl ~ probit(xb_win_bowl)
Prior:
  {win_bowl:score_dif sos _cons} ~ normal(0,10000)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_win_bowl.
```

Example 1: bayes: prefix

```
. bayes, rseed(123):probit win_bowl score_dif sos

Bayesian probit regression                       MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling         Burn-in          =      2,500
                                                 MCMC sample size =     10,000
                                                 Number of obs    =         14
                                                 Acceptance rate  =      .2522
                                                 Efficiency:  min =     .06504
                                                              avg =     .07364
Log marginal likelihood = -25.891444                          max =     .07973

------------------------------------------------------------------------------
             |                                                Equal-tailed
    win_bowl |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
   score_dif |  .1722847   .1011987   .003668   .1633205   .0064462   .4011969
         sos |  .0797042   .2138371   .007573   .0882321  -.3346481   .4871838
       _cons |  -2.08378   1.128949   .044266  -2.033869  -4.501485   .0358983
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
```

Example 1: bayesstats ess

- Let's evaluate the effective sample size.

```
. bayesstats ess
Efficiency summaries    MCMC sample size =    10,000
```

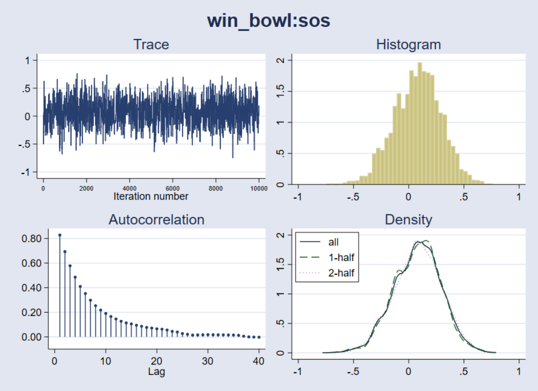
winbowl	ESS	Corr. time	Efficiency
score_dif	761.28	13.14	0.0761
sos	797.34	12.54	0.0797
_cons	650.45	15.37	0.0650

- We expect to have an acceptance rate (see previous slide) that is neither too 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.

Example 1: `bayesgraph`

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

. bayesgraph diagnostic {sos}

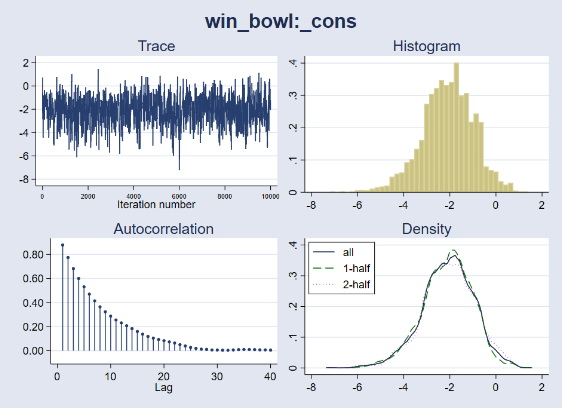


- The trace indicates that convergence was achieved.
- Correlation dies out after around 10 periods.

Example 1: `bayesgraph`

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

. bayesgraph diagnostic {_cons}



- Correlation dies out after around 15 periods.

Example 1: `bayestest model`

- `bayestest model` is another postestimation command to compare different models.
- `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.

Example 1: bayestest model

- Let's fit two other models and compare them with the one we already fit.
- We store the results for the three models, and we use the postestimation command `bayestest model` to select one of them.

```
quietly {
    bayes , rseed(123) saving(dif_sos,replace):      ///
        probit winbowl score_dif sos
    estimates store dif_sos

    bayes , rseed(123) saving(score,replace):        ///
        probit winbowl scored_avg against_avg
    estimates store scored_against

    bayes , rseed(123) saving(srs_linear,replace)   ///
        prior({winbowl:srs}, normal(10,20)):         ///
        block({winbowl:srs _cons}):                  ///
        regress winbowl srs
    estimates store srs_linear
}
bayestest model dif_sos scored_against srs_linear
```

Example 1: `bayestest model`

- Here is the output for `bayestest model`

```
. quietly {
. bayestest model dif_sos scored_against srs_linear
Bayesian model tests
```

	log(ML)	P(M)	P(M|y)
dif_sos	-25.9158	0.3333	0.3679
scored_against	-26.7528	0.3333	0.1593
srs_linear	-25.6652	0.3333	0.4727

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

- We could also assign different priors for the models:

```
. bayestest model dif_sos scored_against srs_linear, ///
                 prior(.3 .5 .2)
Bayesian model tests
```

	log(ML)	P(M)	P(M|y)
dif_sos	-25.9158	0.3000	0.3879
scored_against	-26.7528	0.5000	0.2799
srs_linear	-25.6652	0.2000	0.3322

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

Example 1: `bayestest model`

- Here is the output for `bayestest model`

```
. quietly {
. bayestest model dif_sos scored_against srs_linear
Bayesian model tests
```

	log(ML)	P(M)	P(M|y)
dif_sos	-25.9158	0.3333	0.3679
scored_against	-26.7528	0.3333	0.1593
srs_linear	-25.6652	0.3333	0.4727

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

- We could also assign different priors for the models:

```
. bayestest model dif_sos scored_against srs_linear, ///
                    prior(.3 .5 .2)
Bayesian model tests
```

	log(ML)	P(M)	P(M|y)
dif_sos	-25.9158	0.3000	0.3879
scored_against	-26.7528	0.5000	0.2799
srs_linear	-25.6652	0.2000	0.3322

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

Example 2: Random-effects Poisson model

Example 2: Random-effects Poisson model

- Let's use `bayes:` to fit a random-effects Poisson model for a count dependent variable.

$$Pr(y_{it} = y | x_{it}, \alpha_i) = \frac{e^{-\mu_{it}} \mu_{it}^{y}}{y!}$$

Where:

$$\mu_{i,t} = exp(x_{i,t}\beta + \alpha_i)$$

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

- This is also referred to as a two-level random intercept model.
- We can also fit this model with `mepoisson` or `xtpoisson,re normal`.

Example 2: Random-effects Poisson model

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

```
   clear
   set obs 300
   set seed 123

*Panel level*
   generate id     = _n
   generate alpha  = rnormal(0,.33)

*Observation level*
   expand 5
   bysort id:generate year = _n
   xtset id year
   generate x1 = rnormal()
   generate x2 = runiform()
   generate x3 = rnormal()


*Generate dependent variable*

   generate y = rpoisson(exp(.1*x1-.1*x2+.1*x3+.75+alpha))
```

Example 2: Random-effects Poisson model

Let's show the results with `mepoisson`:

```
. mepoisson y x1 x2 x3 || id:,nolog
Mixed-effects Poisson regression                Number of obs     =      1,500
Group variable:             id                  Number of groups  =        300
                                                Obs per group:
                                                              min =          5
                                                              avg =        5.0
                                                              max =          5
Integration method: mvaghermite                 Integration pts.  =          7
                                                Wald chi2(3)      =      68.33
Log likelihood = -2646.5534                     Prob > chi2       =     0.0000
------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .0806379   .0192914     4.18   0.000     .0428275    .1184484
          x2 |  -.1134928     .06522    -1.74   0.082    -.2413217    .0143361
          x3 |   .1285766   .0187383     6.86   0.000     .0918502    .1653029
       _cons |   .7373862   .0416085    17.72   0.000      .655835    .8189375
-------------+----------------------------------------------------------------
id           |
   var(_cons)|   .1087738   .0171051                      .0799226      .14804
------------------------------------------------------------------------------
LR test vs. Poisson model: chibar2(01) = 116.41       Prob >= chibar2 = 0.0000
```

Example 2: Random-effects Poisson model

- We now fit the model with `bayes`:

```
bayes, nodots rseed(123): ///
        mepoisson y x1 x2 x3 || id:
```

- Equivalent model with `bayesmh`

```
bayesmh y x1 x2 x3, rseed(123)                       ///
    likelihood(poisson) reffects(id)                 ///
    prior({y:x1 x2 x3 _cons}, normal(0,10000))       ///
    prior({y:i.id}, normal(0,{sigma2}))              ///
    prior({sigma2}, igamma(.01,.01))                 ///
    block({sigma2}) nodots
```

Example 2: Random-effects Poisson model

```
. bayes, nodots rseed(123) :      ///
>         mepoisson y x1 x2 x3 || id:

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

Model summary
------------------------------------------------------------------------------
Likelihood:
  y ~ mepoisson(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.
```

Example 2: Random-effects Poisson model

```
. bayes, nodots rseed(123) :      ///
>         mepoisson y x1 x2 x3 || id:

Bayesian multilevel Poisson 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 =        300

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

Family : Poisson                                Number of obs    =      1,500
Link   : log                                    Acceptance rate  =      .2715
                                                Efficiency:  min =     .02614
                                                             avg =      .0409
Log marginal likelihood                                      max =     .05729

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
y            |
          x1 |  .0810731   .0192223   .000803   .0805926   .0448467   .1195346
          x2 | -.1137537   .0648044   .003071  -.1128703  -.2428485   .0164924
          x3 |  .1296011   .0183267    .00082   .1294387   .0931207    .167355
       _cons |  .7368688   .0427745   .002624   .7378466   .6528039   .8186462
-------------+----------------------------------------------------------------
id           |
   U0:sigma2 |  .1099352   .0177164   .001096   .1093387   .0765145   .1469857
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
```

Example 2: Random-effects Poisson model

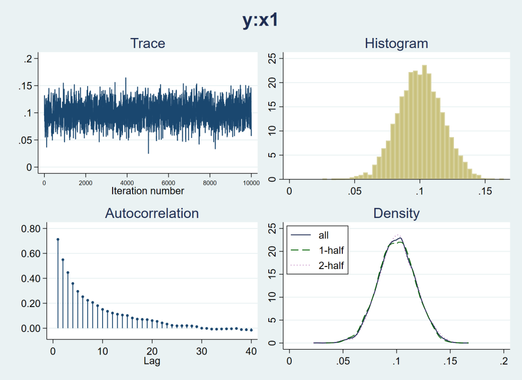
```
. bayesstats ess
Efficiency summaries    MCMC sample size =    10,000
```

	ESS	Corr. time	Efficiency
y			
x1	572.89	17.46	0.0573
x2	445.22	22.46	0.0445
x3	499.81	20.01	0.0500
_cons	265.72	37.63	0.0266
id			
U0:sigma2	261.41	38.25	0.0261

Example 2: `bayesgraph diagnostic`

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

. bayesgraph diagnostic {y:x1}

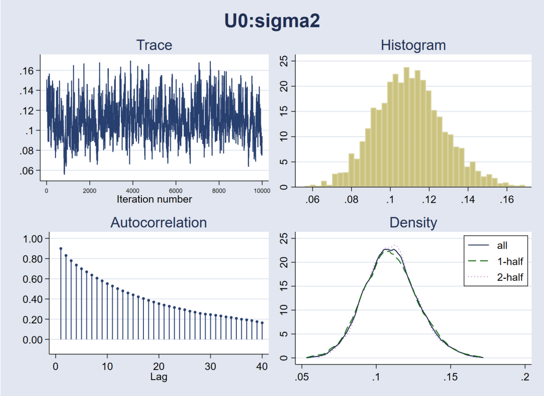


- The trace seems to indicate convergence.
- Autocorrelation becomes negligible after about 15 periods.
- Densities are similar for first and second halves of the MCMC sample.

Example 2: `bayesgraph diagnostic`

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

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



- The trace seems to indicate convergence.
- Autocorrelation is slightly high, but decays steadily.
- Densities are similar for first and second halves of the MCMC sample.

Example 2: `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 hypothesis about intervals for one or more parameters .
- `bayestest interval` reports the estimated posterior mean probability for Ho.

```
bayestest interval ({y:x1},lower(.08) upper(.12)) ///
                   ({y:x2},lower(-.12) upper(-.09))
```

Example 2: bayestest interval

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

```
. bayestest interval  ({y:x1},lower(.08) upper(.12)) ///
>                     ({y:x2},lower(-.12) upper(-.09))
Interval tests     MCMC sample size =    10,000

       prob1 : .08 < {y:x1} < .12
       prob2 : -.12 < {y:x2} < -.09
```

	Mean	Std. Dev.	MCSE
prob1	.4909	0.49994	.0199632
prob2	.1926	0.39436	.0145117

- If we draw θ_1 from the specified prior and we use the data to update the knowledge about θ_1, then there is a 49% chance that θ_1 belongs to the interval (.08,.12).

- We can also perform a joint test:

```
. bayestest interval (({y:x1},lower(.08) upper(.12)) ///
>                    ({y:x2},lower(-.12) upper(-.09)),joint)
Interval tests     MCMC sample size =    10,000

       prob1 : .08 < {y:x1} < .12, -.12 < {y:x2} < -.09
```

	Mean	Std. Dev.	MCSE
prob1	.0885	0.28403	.0098171

Example 2: bayestest interval

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

```
. bayestest interval  ({y:x1},lower(.08) upper(.12)) ///
>                     ({y:x2},lower(-.12) upper(-.09))
Interval tests     MCMC sample size =    10,000

       prob1 : .08 < {y:x1} < .12
       prob2 : -.12 < {y:x2} < -.09
```

	Mean	Std. Dev.	MCSE
prob1	.4909	0.49994	.0199632
prob2	.1926	0.39436	.0145117

  - If we draw θ_1 from the specified prior and we use the data to update the knowledge about θ_1, then there is a 49% chance that θ_1 belongs to the interval (.08,.12).

- We can also perform a joint test:

```
. bayestest interval (({y:x1},lower(.08) upper(.12)) ///
>                    ({y:x2},lower(-.12) upper(-.09)),joint)
Interval tests     MCMC sample size =    10,000

       prob1 : .08 < {y:x1} < .12, -.12 < {y:x2} < -.09
```

	Mean	Std. Dev.	MCSE
prob1	.0885	0.28403	.0098171

Example 2: Show random effects

```
. bayes, show({U0[1/6]}) noheader
```

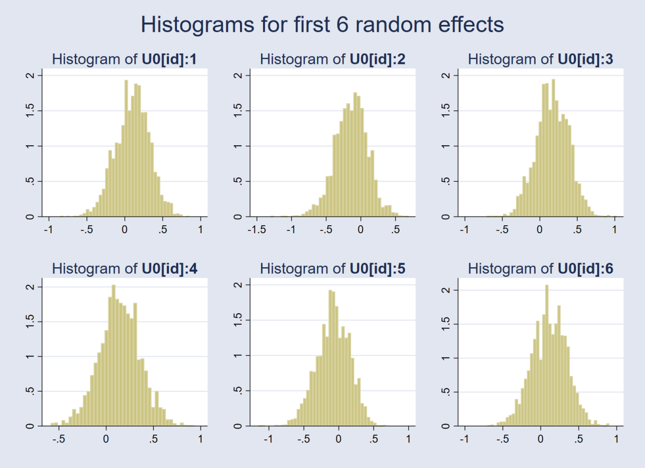
U0[id]	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
1	.1005875	.2248611	.005989	.1137852	-.3503203	.5382369
2	-.1376598	.2372418	.006347	-.1312831	-.6391449	.3238192
3	.1669656	.2171576	.006349	.1645487	-.2620912	.5840191
4	.1415134	.2192747	.006385	.1401843	-.3075952	.5717826
5	-.0802774	.2361239	.007224	-.0747518	-.5665242	.3531596
6	.1128583	.2338012	.006719	.1093227	-.3585934	.5664554

Note: Default priors are used for model parameters.

Example 2: Histograms for random effects

- `bayesgraph histogram`

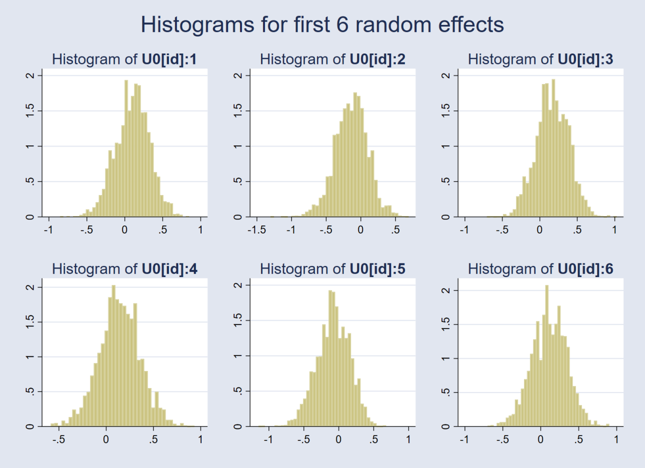
```
. bayesgraph histogram {U0[1/6]},name(g1 g2 g3 g4 g5 g6,replace)
. graph combine g1 g2 g3 g4 g5 g6, ///
>       title("Histograms for first 6 random effects")
```



Example 2: Histograms for random effects

- `bayesgraph histogram`

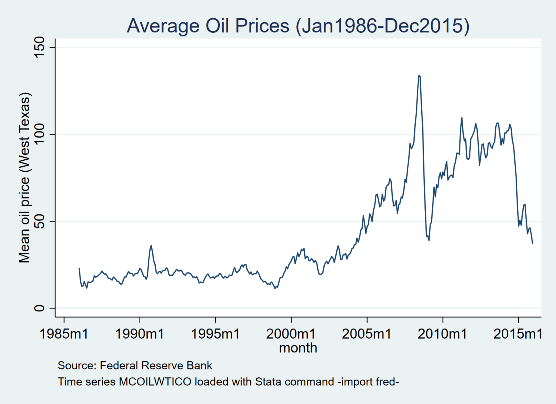
```
. bayesgraph histogram {U0[1/6]},name(g1 g2 g3 g4 g5 g6,replace)
. graph combine g1 g2 g3 g4 g5 g6, ///
>       title("Histograms for first 6 random effects")
```



Example 3: Change-point model

Example 3: Change-point model

- Let's work now with an example where we write our model using a substitutable expression.
- We have average oil prices for January 1986 to December 2015:



- The series has a significant increase around 2005.
- We may consider fitting a change-point model.

Example 3: Gibbs sampling

Change-point model specification with blocking

```
bayesmh oilprice = ({mu1}*sign(year<{cp})                    ///
                 + {mu2}*sign(year>={cp})),                  ///
       likelihood(normal({var}))                             ///
       prior({mu1}, normal(0,50))                            ///
       prior({mu2}, normal(50,150))                          ///
       prior({cp}, uniform(tm(1986m1),2015m12))              ///
       prior({var}, igamma(.01,.01))                         ///
       initial({mu1} =15 {mu2} =100 {cp} =tm(1986m1))        ///
       block({var}, gibbs) block({cp}) blocksummary          ///
       rseed(123) mcmcsize(40000)                            ///
       dots(500,every(5000))
quietly {
    matrix mean=e(mean)
    noisily display _n _col(10) "Date: " mean[1,1]           ///
       _n _col(17) "Cut point (Month): " %tm mean[1,1]
}
```

Example 3: Gibbs sampling

Change-point model specification with blocking

```
bayesmh oilprice = ({mu1}*sign(year<{cp})                        ///
                + {mu2}*sign(year>={cp})),                       ///
    likelihood(normal({var}))                                    ///
    prior({mu1}, normal(0,50))                                   ///
    prior({mu2}, normal(50,150))                                 ///
    prior({cp}, uniform(tm(1986m1),2015m12))                     ///
    prior({var}, igamma(.01,.01))                                ///
    initial({mu1} =15 {mu2} =100 {cp} =tm(1986m1))               ///
    block({var}, gibbs) block({cp}) blocksummary                 ///
    rseed(123) mcmcsize(40000)                                   ///
    dots(500,every(5000))
quietly {
  matrix mean=e(mean)
  noisily display _n _col(10) "Date: " mean[1,1]               ///
     _n _col(17) "Cut point (Month): " %tm mean[1,1]
}
```

Example 3: Gibbs sampling

Change-point model specification with blocking

```
bayesmh oilprice = ({mu1}*sign(year<{cp})                    ///
                 + {mu2}*sign(year>={cp})),                   ///
      likelihood(normal({var}))                               ///
      prior({mu1}, normal(0,50))                              ///
      prior({mu2}, normal(50,150))                            ///
      prior({cp}, uniform(tm(1986m1),2015m12))                ///
      prior({var}, igamma(.01,.01))                           ///
      initial({mu1} =15 {mu2} =100 {cp} =tm(1986m1))          ///
      block({var}, gibbs) block({cp}) blocksummary            ///
      rseed(123) mcmcsize(40000)                              ///
      dots(500,every(5000))
quietly {
    matrix mean=e(mean)
    noisily display _n _col(10) "Date: " mean[1,1]            ///
      _n _col(17) "Cut point (Month): " %tm mean[1,1]
}
```

Example 3: Gibbs sampling

Change-point model specification with blocking

```
bayesmh oilprice = ({mu1}*sign(year<{cp})                ///
                 + {mu2}*sign(year>={cp})),              ///
     likelihood(normal({var}))                           ///
     prior({mu1}, normal(0,50))                          ///
     prior({mu2}, normal(50,150))                        ///
     prior({cp}, uniform(tm(1986m1),2015m12))            ///
     prior({var}, igamma(.01,.01))                       ///
     initial({mu1} =15 {mu2} =100 {cp} =tm(1986m1))      ///
     block({var}, gibbs) block({cp}) blocksummary        ///
     rseed(123) mcmcsize(40000)                          ///
     dots(500,every(5000))
quietly {
   matrix mean=e(mean)
   noisily display _n _col(10) "Date: " mean[1,1]        ///
     _n _col(17) "Cut point (Month): " %tm mean[1,1]
}
```

Example 3: Gibbs sampling

Change-point model specification with blocking

```
bayesmh oilprice = ({mu1}*sign(year<{cp})                    ///
                  + {mu2}*sign(year>={cp})),                  ///
      likelihood(normal({var}))                               ///
      prior({mu1}, normal(0,50))                              ///
      prior({mu2}, normal(50,150))                            ///
      prior({cp}, uniform(tm(1986m1),2015m12))                ///
      prior({var}, igamma(.01,.01))                           ///
      initial({mu1} =15 {mu2} =100 {cp} =tm(1986m1))          ///
      block({var}, gibbs) block({cp}) blocksummary            ///
      rseed(123) mcmcsize(40000)                              ///
      dots(500,every(5000))
quietly {
    matrix mean=e(mean)
    noisily display _n _col(10) "Date: " mean[1,1]            ///
       _n _col(17) "Cut point (Month): " %tm mean[1,1]
}
```

Example 3: Gibbs sampling

Change-point model specification with blocking

```
. bayesmh oilprice=({mu1}*sign(month<{cp})+{mu2}*sign(month>={cp})), ///
>         likelihood(normal({var}))                                   ///
>         prior({mu1}, normal(0,50))                                  ///
>         prior({mu2}, normal(50,150))                                ///
>         prior({cp}, uniform(tm(1986m1),tm(2015m12)))                ///
>         prior({var}, igamma(.01,.01))                               ///
>         initial({mu1} =15 {mu2} =100 {cp} =tm(1986m1)) rseed(123)   ///
>         block({var}, gibbs) block({cp}) blocksummary                ///
>         mcmcsize(20000) dots(500, every(5000))

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

Priors:
  {var} ~ igamma(.01,.01)
  {mu1} ~ normal(0,50)
  {mu2} ~ normal(50,150)
   {cp} ~ uniform(tm(1986m1),tm(2015m12))
------------------------------------------------------------------------------

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

Example 3: Gibbs sampling

Change-point model specification with blocking

```
. bayesmh oilprice=({mu1}*sign(month<{cp})+{mu2}*sign(month>={cp})),  ///
>         likelihood(normal({var}))                                   ///
>         prior({mu1}, normal(0,50))                                  ///
>         prior({mu2}, normal(50,150))                                ///
>         prior({cp}, uniform(tm(1986m1),tm(2015m12)))                ///
>         prior({var}, igamma(.01,.01))                               ///
>         initial({mu1} =15 {mu2} =100 {cp} =tm(1986m1)) rseed(123)   ///
>         block({var}, gibbs) block({cp}) blocksummary                ///
>         mcmcsize(20000) dots(500, every(5000))

Bayesian normal regression                       MCMC iterations  =     22,500
Metropolis-Hastings and Gibbs sampling           Burn-in          =      2,500
                                                 MCMC sample size =     20,000
                                                 Number of obs    =        360
                                                 Acceptance rate  =      .5632
                                                 Efficiency:  min =     .09094
                                                              avg =      .3304
Log marginal likelihood = -1481.9487                          max =          1
```

	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
cp	541.5063	1.806737	.037169	541.4515	536.7238	544.9228
mu1	22.07432	.936419	.01974	22.09333	20.23623	23.85525
mu2	78.69139	1.259118	.029524	78.67589	76.2043	81.19035
var	197.286	14.80914	.104716	196.6902	169.991	228.0003

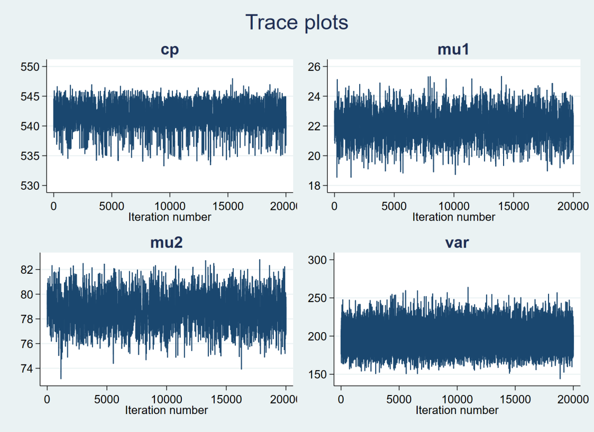
```
. quietly {

          elapsed date:   541.50629
    Cut point (Month):   2005m2
```

Example 3: `bayesgraph trace`

- Use bayesgraph trace to look at the trace for all the parameters.

. bayesgraph trace _all,combine

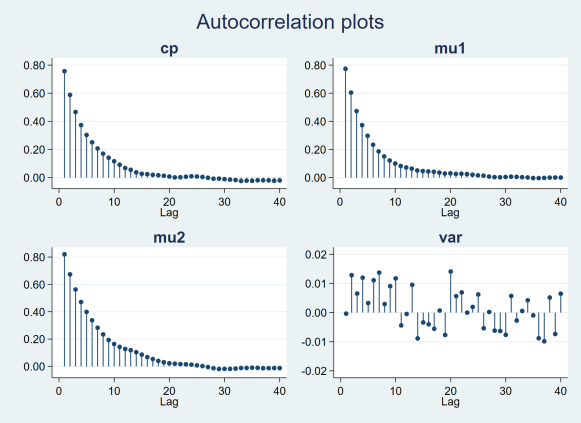


- The plots indicate that convergence seems to be achieved.

Example 3: `bayesgraph ac`

- Use bayesgraph ac to look at the autocorrelation for all the parameters.

. bayesgraph ac _all,combine

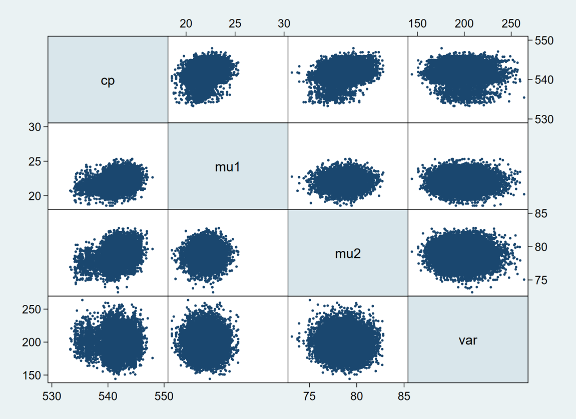


- Autocorrelation quickly becomes negligible for all the parameters.

Example 3: `bayesgraph matrix`

- Use `bayesgraph matrix` to look at pairwise correlation for the parameters.

. bayesgraph matrix _all



- The plots seem to indicate that there are no significant pairwise correlations among the parameters.

Summing up

- Bayesian analysis: A statistical approach that can be used to answer questions about unknown parameters in terms of probability statements.

- It can be used when we have prior information on the distribution of the parameters involved in the model.

- Alternative approach or complementary approach to classic/frequentist approach?

Reference

Cameron, A. and Trivedi, P. 2005. *Microeconometric Methods and Applications*. Cambridge University Press, Section 13.2.2, 422–423.

Links

https://www.stata.com/meeting/uk17/slides/uk17_Marchenko.pdf

https://www.stata.com/meeting/brazil16/slides/rising-brazil16.pdf

https://www.stata.com/meeting/spain18/slides/spain18_Sanchez.pdf