Introduction to Bayesian Analysis using Stata

Gustavo Sánchez

StataCorp LLC

Virtual|November 19, 2020
Swiss Stata Conference

Outline

1. Bayesian analysis: Basic concepts
    - The general idea
    - The method

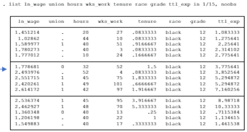
2. The Stata tools
    - The general command `bayesmh`
    - The `bayes` prefix
    - Postestimation commands
    - New in Stata 16
        - Multiple chains
        - Bayes predictions

3. A few examples
    - Linear regression
    - Random-effects probit model
    - Population mean

The general idea

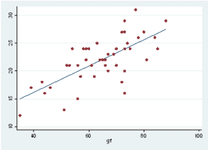
Frequentist

Theoretical Model →



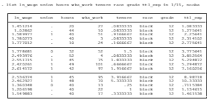
```
. list ln_wage union hours wks_work tenure race grade ttl_exp in 1/15, noobs

  ┌─────────────────────────────────────────────────────────────────────────┐
  │  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 │
  └─────────────────────────────────────────────────────────────────────────┘
```

→

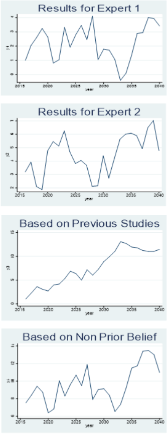


The general idea

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 (hypothetically repeatable).
- Uses data to estimate unknown fixed parameters.
- *p*-values are conditional probability statements that assume Ho to be true.

"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.
- It allows formulating probabilistic statements for the hypothesis of interest.

"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 (Fundamental Equation)

- 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; so 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 (Fundamental Equation)

- 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; so 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 (Fundamental Equation)

- 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; so 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

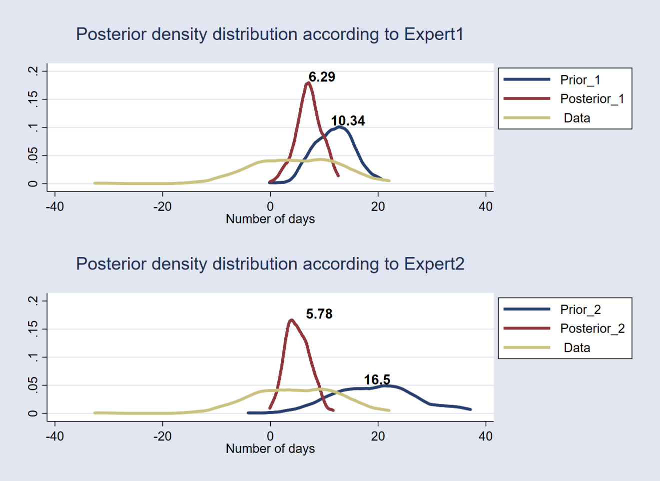
- Let's assume that both the data and the prior beliefs are normally distributed:
  - **The data**: $y \sim N\left(\theta, \sigma_d^2\right)$
  - **The prior**: $\theta \sim N\left(\mu_p, \sigma_p^2\right)$
- Homework...: Doing the algebra with the fundamental equation, we find that the posterior distribution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

- 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

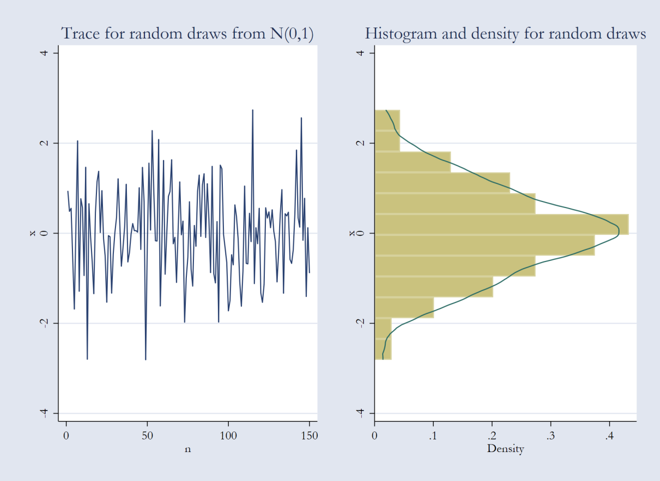
- Links for Bayesian analysis and MCMC on our YouTube channel:
  - Introduction to Bayesian statistics, part 1: The basic concepts.

    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

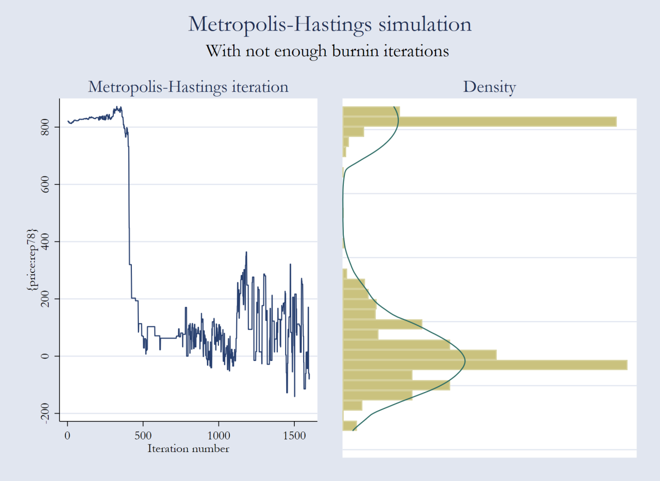
The method

- Monte Carlo simulation



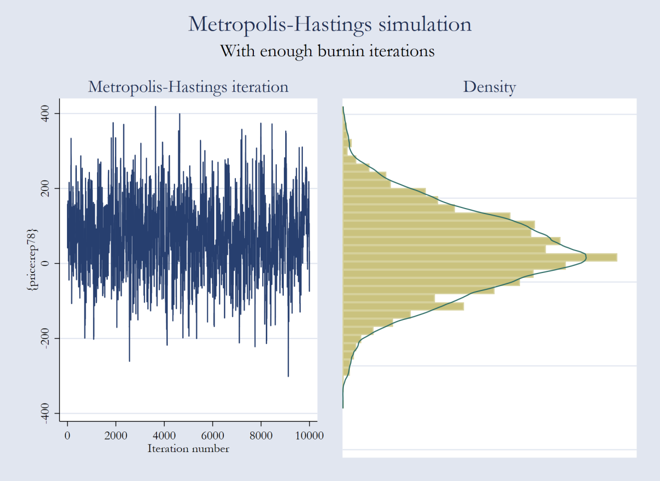
The method

- Metropolis–Hastings simulation
  - The trace plot illustrates the sequence of accepted proposal states for a simulation with not enough burnin iterations.



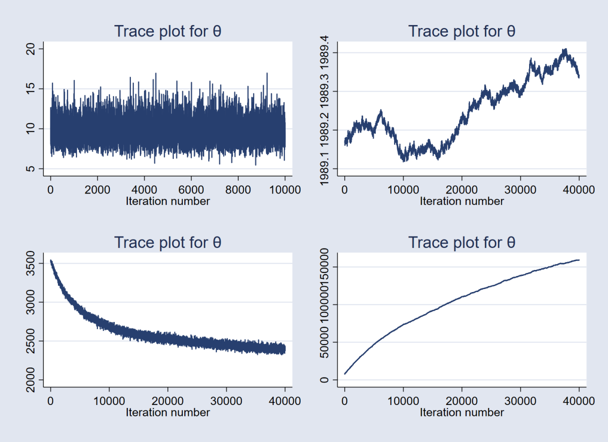
The method

- Metropolis–Hastings simulation
  - The trace plot illustrates the sequence of accepted proposal states for a simulation with enough burnin iterations.



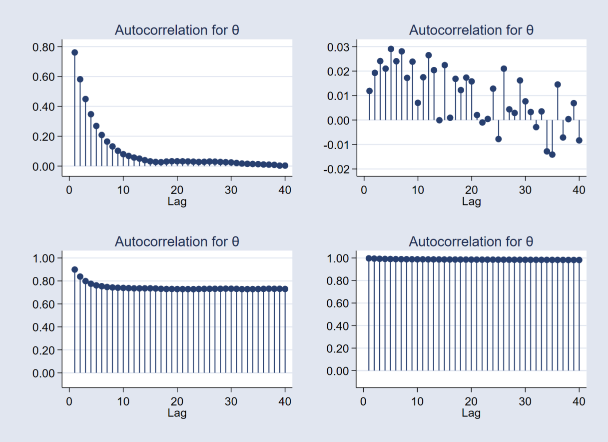
The method

- 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

Stata's Bayesian suite consists of the following commands

Command	*Description*
Estimation	
`bayes:`	Bayesian regression models using the `bayes` prefix
`bayesmh`	General Bayesian models using MH
`bayesmh` *evaluators*	User-defined Bayesian models using MH
Postestimation	
`bayesgraph`	Graphical convergence diagnostics
`bayesstats ess`	Effective sample sizes and more
`bayesstats grubin`	Gelman–Rubin convergence diagnostics
`bayesstats ic`	Information criteria and Bayes factors
`bayestest model`	Model posterior probabilities
`bayestest interval`	Interval hypothesis testing
`bayesstats summary`	Summary statistics
`bayespredict`	Bayesian predictions (available only after `bayesmh`)
`bayesreps`	Bayesian replications (available only after `bayesmh`)
`bayesstats ppvalues`	Bayesian predictive *p*-values (available only after `bayesmh`)

Built-in models and methods available in Stata

- Over 50 built-in likelihoods: normal, logit, ologit, Poisson, . . .
- Many built-in priors: normal, gamma, Wishart, Zellner's g, . . .
- Continuous, binary, ordinal, categorical, count, censored, truncated, zero-inflated, and survival outcomes.
- Univariate, multivariate, and multiple-equation models.
- Linear, nonlinear, generalized linear and nonlinear, sample-selection, panel-data, and multilevel models.
- Continuous univariate, multivariate, and discrete priors.
- User-defined models: likelihoods and priors.

MCMC methods:

- Adaptive MH.
- Adaptive MH with Gibbs updates—hybrid.
- Full Gibbs sampling for some models.

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.

Example 1: Life expectancy in the U.S.

- Let's work with a simple linear regression for the life expectancy in the U.S. We are going to be considering the following model specifications:

$$\begin{aligned} life\_exp &= \alpha_1 + \beta_{health\_cons} * health\_cons + \beta_{school} * school \\ &+ \beta_{pop\_growth} * pop\_growth + \epsilon_1 \end{aligned}$$

$$\begin{aligned} life\_exp &= \alpha_1 + \beta_{health\_educ} * health\_educ + \beta_{school} * school \\ &+ \beta_{pop\_growth} * pop\_growth + \epsilon_2 \end{aligned}$$

$$\begin{aligned} life\_exp &= \alpha_1 + \beta_{gdp\_capita} * gdp\_capita + \beta_{school} * school \\ &+ \beta_{pop\_growth} * pop\_growth + \epsilon_3 \end{aligned}$$

Where:

life_exp : Life expectancy at birth. Total for U.S.
health_cons : Real health consumption expenditure. Total for U.S.
health_educ : Real health and education expenditure. Total for U.S.
gdp_capita : Real GDP per capita for U.S.
school : School enrollment ratio female/male for U.S.
pop_growth : Population growth for U.S.

Example 1: Life expectancy in the U.S.

- Let's work with a simple linear regression for the life expectancy in the U.S. We are going to be considering the following model specifications:

$$\begin{aligned} life\_exp &= \alpha_1 + \beta_{health\_cons} * health\_cons + \beta_{school} * school \\ &+ \beta_{pop\_growth} * pop\_growth + \epsilon_1 \end{aligned}$$

$$\begin{aligned} life\_exp &= \alpha_1 + \beta_{health\_educ} * health\_educ + \beta_{school} * school \\ &+ \beta_{pop\_growth} * pop\_growth + \epsilon_2 \end{aligned}$$

$$\begin{aligned} life\_exp &= \alpha_1 + \beta_{gdp\_capita} * gdp\_capita + \beta_{school} * school \\ &+ \beta_{pop\_growth} * pop\_growth + \epsilon_3 \end{aligned}$$

Where:

life_exp : Life expectancy at birth. Total for U.S.
health_cons : Real health consumption expenditure. Total for U.S.
health_educ : Real health and education expenditure. Total for U.S.
gdp_capita : Real GDP per capita for U.S.
school : School enrollment ratio female/male for U.S.
pop_growth : Population growth for U.S.

Data

- We used `import fred` to get data from the Federal Reserve Economic Data (FRED).

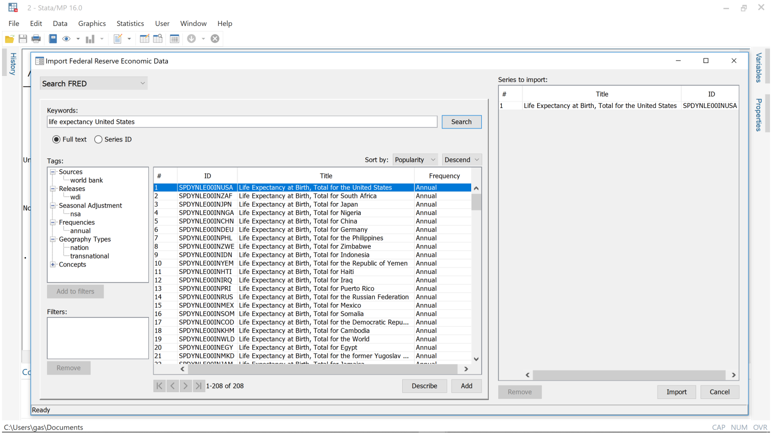
```
import fred SPDYNLE00INUSA DEDURX1A020NBEA          ///
   DHLTRX1A020NBEA NYGDPPCAPKDUSA SEENRSECOFMZSUSA ///
   SPPOPGROWUSA, daterange(2002-01-01 2016-01-01)  ///
   aggregate(annual,avg) clear
generate year=year(daten)
tsset year
rename SPDYNLE00INUSA   life_exp
rename DEDURX1A020NBEA  educ_cons
rename DHLTRX1A020NBEA  health_cons
rename NYGDPPCAPKDUSA   gdp_capita
rename SEENRSECOFMZSUSA school
rename SPPOPGROWUSA     pop_growth
generate health_educ = health_cons+educ_cons
replace  health_cons = health_cons/1000
replace  health_educ = health_educ/1000
replace  gdp_capita  = gdp_capita/1000
```

Data

- We used `import fred` to get data from the Federal Reserve Economic Data (FRED).

```
import fred SPDYNLE00INUSA DEDURX1A020NBEA          ///
   DHLTRX1A020NBEA NYGDPPCAPKDUSA SEENRSECOFMZSUSA ///
   SPPOPGROWUSA, daterange(2002-01-01 2016-01-01)  ///
   aggregate(annual,avg) clear
generate year=year(daten)
tsset year
rename SPDYNLE00INUSA   life_exp
rename DEDURX1A020NBEA  educ_cons
rename DHLTRX1A020NBEA  health_cons
rename NYGDPPCAPKDUSA   gdp_capita
rename SEENRSECOFMZSUSA school
rename SPPOPGROWUSA     pop_growth
generate health_educ = health_cons+educ_cons
replace  health_cons = health_cons/1000
replace  health_educ = health_educ/1000
replace  gdp_capita  = gdp_capita/1000
```

`import fred`: Dialog box



Keywords: life expectancy United States

● Full text ○ Series ID

Sort by: Popularity Descend

#	ID	Title	Frequency
1	SPDYNLE00INUSA	Life Expectancy at Birth, Total for the United States	Annual
2	SPDYNLE00INZAF	Life Expectancy at Birth, Total for South Africa	Annual
3	SPDYNLE00INJPN	Life Expectancy at Birth, Total for Japan	Annual
4	SPDYNLE00INNGA	Life Expectancy at Birth, Total for Nigeria	Annual
5	SPDYNLE00INCHN	Life Expectancy at Birth, Total for China	Annual
6	SPDYNLE00INDEU	Life Expectancy at Birth, Total for Germany	Annual
7	SPDYNLE00INPHL	Life Expectancy at Birth, Total for the Philippines	Annual
8	SPDYNLE00INZWE	Life Expectancy at Birth, Total for Zimbabwe	Annual
9	SPDYNLE00INIDN	Life Expectancy at Birth, Total for Indonesia	Annual
10	SPDYNLE00INYEM	Life Expectancy at Birth, Total for the Republic of Yemen	Annual
11	SPDYNLE00INHTI	Life Expectancy at Birth, Total for Haiti	Annual
12	SPDYNLE00INIRQ	Life Expectancy at Birth, Total for Iraq	Annual
13	SPDYNLE00INPRI	Life Expectancy at Birth, Total for Puerto Rico	Annual
14	SPDYNLE00INRUS	Life Expectancy at Birth, Total for the Russian Federation	Annual
15	SPDYNLE00INMEX	Life Expectancy at Birth, Total for Mexico	Annual
16	SPDYNLE00INSOM	Life Expectancy at Birth, Total for Somalia	Annual
17	SPDYNLE00INCOD	Life Expectancy at Birth, Total for the Democratic Repu...	Annual
18	SPDYNLE00INKHM	Life Expectancy at Birth, Total for Cambodia	Annual
19	SPDYNLE00INWLD	Life Expectancy at Birth, Total for the World	Annual
20	SPDYNLE00INEGY	Life Expectancy at Birth, Total for Egypt	Annual
21	SPDYNLE00INMKD	Life Expectancy at Birth, Total for the former Yugoslav ...	Annual

1-208 of 208

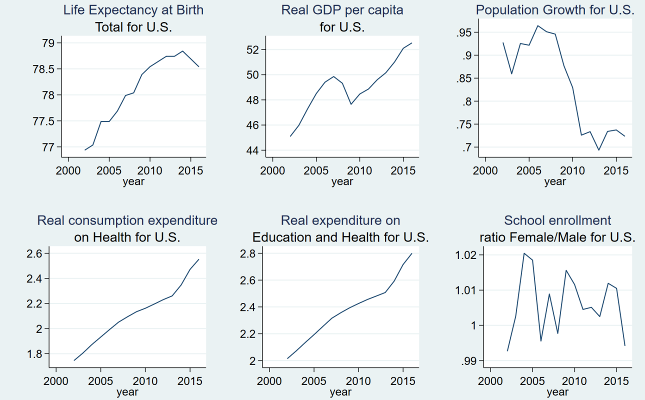
Series to import:

#	Title	ID
1	Life Expectancy at Birth, Total for the United States	SPDYNLE00INUSA

Ready

C:\Users\gas\Documents

Graphs



Source: Federal Reserve Economic Data (FRED)
Downloaded using -import fred- in Stata

- Linear regression with the `bayes`: prefix

```
bayes ,rseed(1) blocksummary:                              ///
  regress life_exp health_cons pop_growth school
```

- Equivalent model with `bayesmh`

```
bayesmh life_exp health_cons pop_growth school,            ///
  likelihood(normal({sigma2}))                             ///
  prior({life_exp:health_cons}, normal(0,10000))           ///
  prior({life_exp:pop_growth}, normal(0,10000))            ///
  prior({life_exp:school}, normal(0,10000))                ///
  prior({life_exp:_cons}, normal(0,10000))                 ///
  prior({sigma2}, igamma(.01,.01))                         ///
  block({sigma2}) rseed(1)                                 ///
  block({life_exp:health_cons pop_growth school _cons})
```

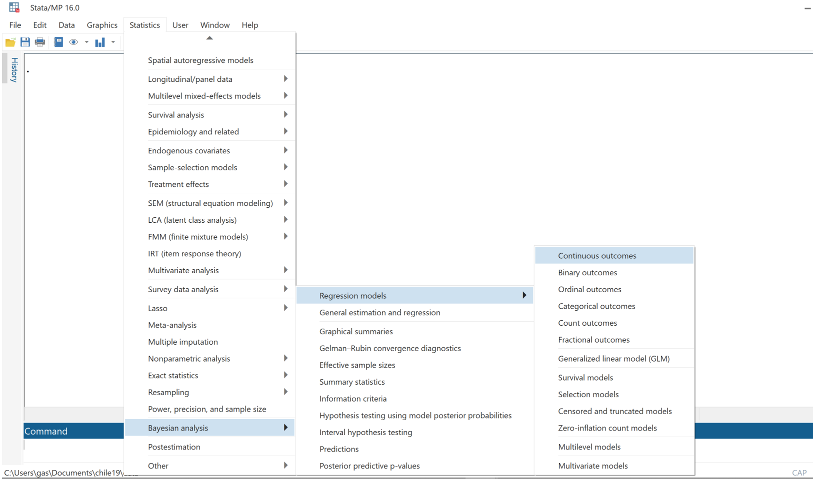
- Linear regression with the `bayes:` prefix

```
bayes ,rseed(1) blocksummary:                              ///
    regress life_exp health_cons pop_growth school
```

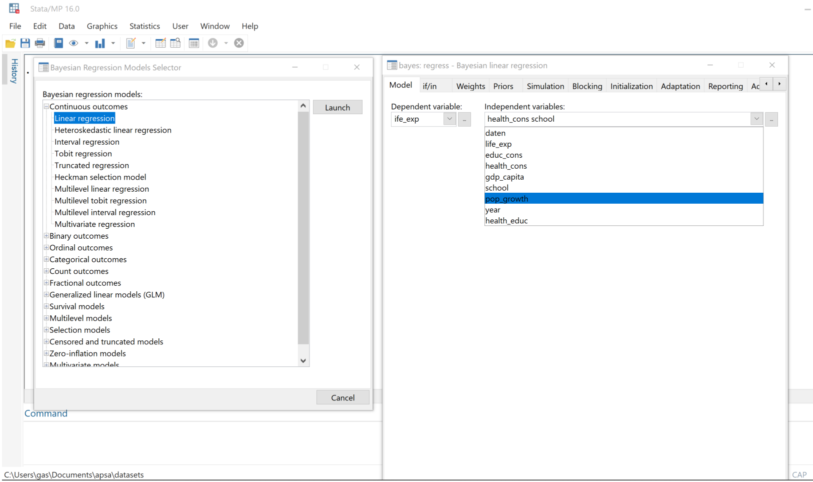
- Equivalent model with `bayesmh`

```
bayesmh life_exp health_cons pop_growth school,            ///
    likelihood(normal({sigma2}))                           ///
    prior({life_exp:health_cons}, normal(0,10000))         ///
    prior({life_exp:pop_growth}, normal(0,10000))          ///
    prior({life_exp:school}, normal(0,10000))              ///
    prior({life_exp:_cons}, normal(0,10000))               ///
    prior({sigma2}, igamma(.01,.01))                       ///
    block({sigma2}) rseed(1)                               ///
    block({life_exp:health_cons pop_growth school _cons})
```

Menu for Bayesian regression



Menu for Bayesian regression



Bayesian Regression Models Selector

Bayesian regression models:

- Continuous outcomes
  - Linear regression
  - Heteroskedastic linear regression
  - Interval regression
  - Tobit regression
  - Truncated regression
  - Heckman selection model
  - Multilevel linear regression
  - Multilevel tobit regression
  - Multilevel interval regression
  - Multivariate regression
- Binary outcomes
- Ordinal outcomes
- Categorical outcomes
- Count outcomes
- Fractional outcomes
- Generalized linear models (GLM)
- Survival models
- Multilevel models
- Selection models
- Censored and truncated models
- Zero-inflation models
- Multivariate models

bayes: regress - Bayesian linear regression

Dependent variable: ife_exp

Independent variables: health_cons school

daten
life_exp
educ_cons
health_cons
gdp_capita
school
pop_growth
year
health_educ

C:\Users\gas\Documents\apsa\datasets

Menu sequence for Bayesian regression

1. Make the following sequence of selection from the main menu:
   Statistics > Bayesian analysis > Regression models
2. Select 'Continuous outcomes'.
3. Select 'Linear regression'.
4. Click on 'Launch'.
5. Specify the dependent variable (life_exp) and the explanatory variables (health_cons school pop_growth).
6. Click on 'OK'.

Prefix command `bayes:`

```
. bayes,rseed(1) blocksummary:                          ///
>       regress life_exp health_cons pop_growth school

Burn-in ...
Simulation ...
Model summary
------------------------------------------------------------------------------
Likelihood:
  life_exp ~ regress(xb_life_exp,{sigma2})
Priors:
  {life_exp:health_cons pop_growth school _cons} ~ normal(0,10000)
                                        {sigma2} ~ igamma(.01,.01)
------------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_life_exp.
Block summary
------------------------------------------------------------------------------
   1:  {life_exp:health_cons pop_growth school _cons}
   2:  {sigma2}
------------------------------------------------------------------------------
```

```
. bayes,rseed(1) blocksummary:                   ///
>       regress life_exp health_cons pop_growth school

Bayesian linear regression                   MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling     Burn-in          =      2,500
                                             MCMC sample size =     10,000
                                             Number of obs    =         15
                                             Acceptance rate  =      .3118
                                             Efficiency:  min =     .05276
                                                          avg =     .06011
Log marginal-likelihood = -24.244226                      max =     .07019

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
life_exp     |
 health_cons |  2.072218   .5749819   .022738   2.100761   .8911282    3.19791
  pop_growth | -1.298569   1.301589    .04913  -1.228649   -4.00535   1.254212
      school |  12.77527   9.605456   .410609   13.04013  -6.617371   32.14734
       _cons |   61.9527    9.83164   .428044   62.02925    42.3255    81.8623
-------------+----------------------------------------------------------------
      sigma2 |  .1043956   .0519073   .002138   .0911482   .0443204   .2389263
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
```

We expect to have an acceptance rate that is neither too small nor too large.

`bayesstats ess`

- Let's evaluate the effective sample size.

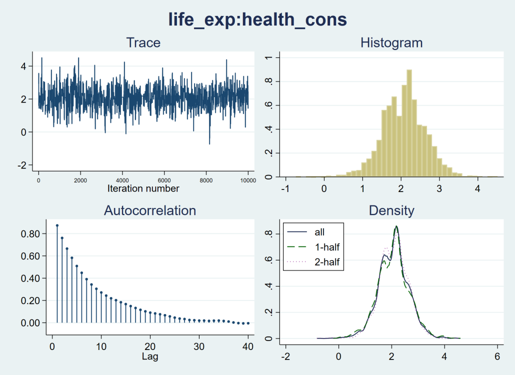
```
. bayesstats ess
Efficiency summaries    MCMC sample size =    10,000
                        Efficiency:  min =    .05276
                                     avg =    .06011
                                     max =    .07019

-------------------------------------------------
             |        ESS   Corr. time    Efficiency
-------------+-----------------------------------
life_exp     |
 health_cons |     639.46        15.64        0.0639
  pop_growth |     701.85        14.25        0.0702
      school |     547.24        18.27        0.0547
       _cons |     527.56        18.96        0.0528
-------------+-----------------------------------
      sigma2 |     589.34        16.97        0.0589
-------------------------------------------------
```

- We expect to have low autocorrelation. Correlation time provides an estimate for the lag after which autocorrelation in an MCMC sample is small.
- Efficiencies over 10% are considered good for MH. Efficiencies under 1% would be a source of concern.

`bayesgraph`

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

```
. bayesgraph diagnostic {health_cons}
```

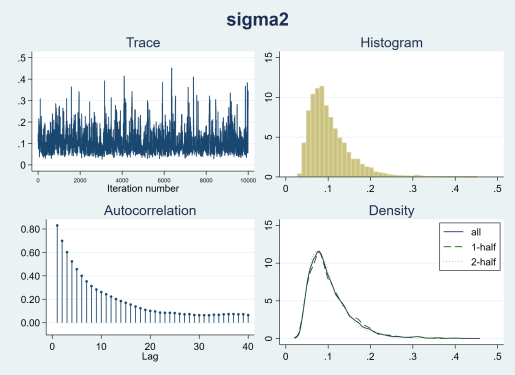


- The trace indicates that convergence was achieved.
- Correlation becomes negligible after 20 periods.

`bayesgraph`

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

. bayesgraph diagnostic {sigma2}



- The trace indicates that convergence was achieved.
- Correlation becomes negligible after 20 periods.

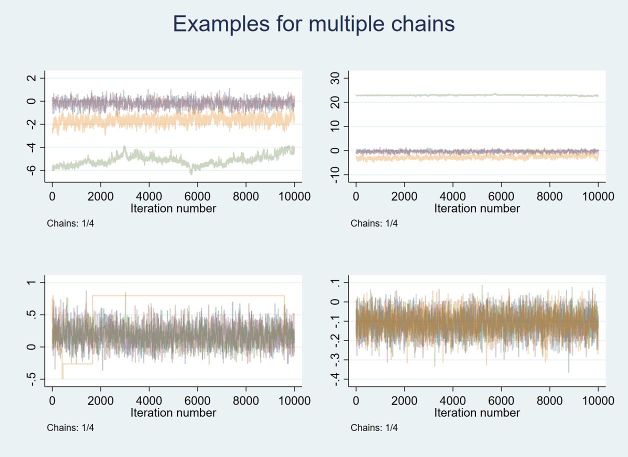
Multiple Markov chains

Multiple Markov chains

- Convergence requires the chains to be stationary and well mixed.
- Performing the estimation on multiple chains allows checking for convergence (stationarity).
- In general, three to four chains should be enough to check for convergence.
- The Gelman–Rubin convergence diagnostic statistic (R_c) helps in deciding whether convergence was reached.
  - Compares variances for the weighted average of between-chains and within-chains variances.
  - R_c greater than 1.1 indicates convergence problems.

Trace for multiple chains

- We expect to see similar trace plots for all the chains:



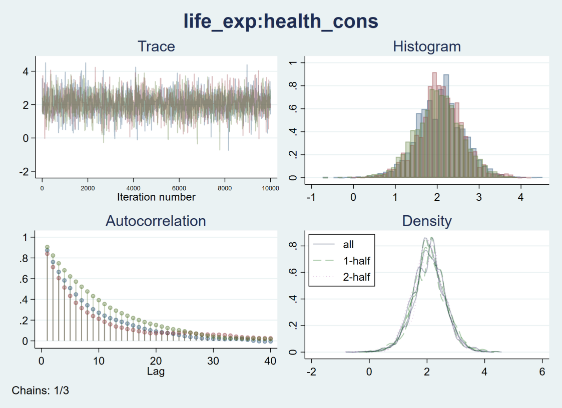
Example 2: Multiple chains with `bayes:` prefix

```
. bayes, rseed(1) nchains(3):                                    ///
>       regress life_exp health_cons pop_growth school

Chain 1
  Burn-in ...
  Simulation ...

Chain 2
  Burn-in ...
  Simulation ...

Chain 3
  Burn-in ...
  Simulation ...
Model summary
---------------------------------------------------------------------------
Likelihood:
  life_exp ~ regress(xb_life_exp,{sigma2})

Priors:
  {life_exp:health_cons pop_growth school _cons} ~ normal(0,10000)
                                        {sigma2} ~ igamma(.01,.01)
---------------------------------------------------------------------------
(1) Parameters are elements of the linear form xb_life_exp.
```

```
. bayes, rseed(1) nchains(3):                                   ///
>       regress life_exp health_cons pop_growth school

Bayesian linear regression                       Number of chains    =          3
Random-walk Metropolis-Hastings sampling         Per MCMC chain:
                                                     Iterations      =     12,500
                                                     Burn-in         =      2,500
                                                     Sample size     =     10,000
                                                 Number of obs       =         15
                                                 Avg acceptance rate =      .3361
                                                 Avg efficiency: min =     .05592
                                                                 avg =     .05928
                                                                 max =     .06243
Avg log marginal-likelihood = -24.228225         Max Gelman-Rubin Rc =      1.012

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
life_exp     |
 health_cons |  2.061216   .5616096   .013198   2.067605    .933505    3.18182
  pop_growth | -1.285638   1.293889   .029899  -1.258758  -3.969357   1.244759
      school |  13.04088    9.76268   .231469   13.01902  -6.213398   32.40011
       _cons |  61.69646   9.936567   .242602   61.67632   42.07788   81.67689
-------------+----------------------------------------------------------------
      sigma2 |  .1054626   .0537058   .001283   .0921645    .044239   .2470984
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
Note: Default initial values are used for multiple chains.
```

`bayesgraph` with multiple chains

- We expect to see similar diagnostic plots for all the chains:
  . bayesgraph diagnostic {health_cons}



- The trace indicates that convergence was achieved.
- Correlation decays for all the chains and the histograms and densities seem to indicate convergence.

Postestimation

`bayestest model`

- `bayestest model` is a 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 posteriors.
- It can be used to compare models with
  - different priors, different posterior distributions, or both;
  - different regression functions, and
  - different covariates.
- MCMC convergence should be verified before comparing the models.

Example 3: `bayestest model`

- Let's fit two other models and compare them with the one we already fit.
- Store the results for the three models and use the post-estimation command `bayestest model` to select one.

```
quietly {
        bayes , rseed(1) saving(health):                    ///
            regress life_exp health_cons pop_growth school
        estimates store health

        bayes , rseed(1) saving(health_educ):               ///
            regress life_exp health_educ pop_growth school
        estimates store health_educ

        bayes , rseed(1) saving(gdp_capita):                ///
            regress life_exp gdp_capita pop_growth school
        estimates store gdp_capita
}
bayestest model health health_educ gdp_capita
```

Here is the output for `bayestest model`:

```
. quietly {
. bayestest model health health_educ gdp_capita
Bayesian model tests

─────────────┬──────────────────────────────────────
             │      log(ML)        P(M)      P(M|y)
─────────────┼──────────────────────────────────────
      health │     -24.2442      0.3333      0.4384
 health_educ │     -24.0065      0.3333      0.5561
  gdp_capita │     -28.6256      0.3333      0.0055
─────────────┴──────────────────────────────────────
Note: Marginal likelihood (ML) is computed using
      Laplace-Metropolis approximation.
```

We could also assign different priors for the models:

```
. bayestest model health health_educ gdp_capita, ///
          prior(.3 .2 .5)
Bayesian model tests

─────────────┬──────────────────────────────────────
             │      log(ML)        P(M)      P(M|y)
─────────────┼──────────────────────────────────────
      health │     -24.2442      0.3000      0.5358
 health_educ │     -24.0065      0.2000      0.4530
  gdp_capita │     -28.6256      0.5000      0.0112
─────────────┴──────────────────────────────────────
Note: Marginal likelihood (ML) is computed using
      Laplace-Metropolis approximation.
```

Here is the output for `bayestest model`:

```
. quietly {
. bayestest model health health_educ gdp_capita
Bayesian model tests
------------------------------------------------
               |    log(ML)      P(M)    P(M|y)
---------------+--------------------------------
        health |   -24.2442    0.3333    0.4384
   health_educ |   -24.0065    0.3333    0.5561
    gdp_capita |   -28.6256    0.3333    0.0055
------------------------------------------------
Note: Marginal likelihood (ML) is computed using
      Laplace-Metropolis approximation.
```

We could also assign different priors for the models:

```
. bayestest model health health_educ gdp_capita, ///
           prior(.3 .2 .5)
Bayesian model tests
------------------------------------------------
               |    log(ML)      P(M)    P(M|y)
---------------+--------------------------------
        health |   -24.2442    0.3000    0.5358
   health_educ |   -24.0065    0.2000    0.4530
    gdp_capita |   -28.6256    0.5000    0.0112
------------------------------------------------
Note: Marginal likelihood (ML) is computed using
      Laplace-Metropolis approximation.
```

`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(#) upper(#)) ///
                   ({y:x2},lower(#) upper(#))
```

Example 4: `bayestest interval`

- Separate tests for different parameters:

```
. estimates restore health
(results health are active now)

. bayestest interval                                           ///
>       ({life_exp:health_cons}, lower(1.5) upper(2.25))   ///
>       ({sigma2},lower(.075))

Interval tests     MCMC sample size =    10,000

       prob1 : 1.5 < {life_exp:health_cons} < 2.25
       prob2 : {sigma2} > .075

-------------------------------------------------
             |      Mean    Std. Dev.      MCSE
-------------+-----------------------------------
       prob1 |     .5038     0.50001   .0185749
       prob2 |     .6836     0.46509   .0145983
-------------------------------------------------
```

  - 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 50% chance that θ_1 belongs to the interval (1.5,2.25).

- We can also perform a joint test by specifying the "joint"'option.

Example 4: `bayestest interval`

- Separate tests for different parameters:

```
. estimates restore health
(results health are active now)

. bayestest interval                                                ///
>       ({life_exp:health_cons}, lower(1.5) upper(2.25))            ///
>       ({sigma2},lower(.075))

Interval tests     MCMC sample size =    10,000

       prob1 : 1.5 < {life_exp:health_cons} < 2.25
       prob2 : {sigma2} > .075

-------------------------------------------------
              |      Mean    Std. Dev.      MCSE
--------------+----------------------------------
        prob1 |     .5038     0.50001   .0185749
        prob2 |     .6836     0.46509   .0145983
-------------------------------------------------
```

  - 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 50% chance that θ_1 belongs to the interval (1.5,2.25).

- We can also perform a joint test by specifying the "joint"'option.

Example 5: `Random-effects probit`

- Consider a random-effects probit model for a binary variable, whose values depend on a linear latent variable.

$$y_{it}^{*} = \beta_0 + \beta_1 x_{1it} + \beta_2 x_{2it} + \ldots + \beta_k x_{kit} + \alpha_i + \epsilon_{it}$$

where

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

$\alpha_i \sim N\left(0, \sigma_{\alpha}^2\right)$ is the individual random panel effect and $\epsilon_{it} \sim N\left(0, \sigma_{e}^2\right)$ is the idiosyncratic error term.

- The above model is also referred to as a two-level random-intercept probit model.
- We can fit this model using `meprobit` or `xtprobit, re`.

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

```
   clear
   set obs 250
   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()*2
   generate x2 = runiform()*4
   generate x3 = runiform()*6
   generate  u = rnormal()

* Generate dependent variable *
   generate y = .25 + .05*x1 + (-.05)*x2 + .05*x3+alpha+u>0
```

- Let's first fit a classical random-effects probit model to these data using `meprobit`:

```
. meprobit y x1 x2 x3 || id:,nolog
Mixed-effects probit regression                 Number of obs     =      1,250
Group variable:                 id              Number of groups  =        250

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

Integration method: mvaghermite                 Integration pts.  =          7

                                                Wald chi2(3)      =      15.82
Log likelihood = -765.58807                     Prob > chi2       =     0.0012
------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .0554992   .0218748     2.54   0.011     .0126254     .098373
          x2 |  -.0816423   .0388118    -2.10   0.035    -.1577121   -.0055726
          x3 |   .0495629   .0253132     1.96   0.050    -.0000501    .0991758
       _cons |   .2951457   .1307708     2.26   0.024     .0388397    .5514517
-------------+----------------------------------------------------------------
id           |
   var(_cons)|   .8359797   .1469796                      .5922975    1.179917
------------------------------------------------------------------------------
LR test vs. probit model: chibar2(01) = 150.87        Prob >= chibar2 = 0.0000
```

- To fit a Bayesian random-effects probit model, we can simply prefix our previous `meprobit` specification with `bayes:`. We additionally specify a random-number seed in `rseed()` for reproducibility and suppress the display of dots by specifying `nodots`.

```
. bayes, nodots rseed(50): meprobit y x1 x2 x3 || id:

Burn-in ...
Simulation ...
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.
------------------------------------------------------------------------
```

```
. bayes, nodots rseed(50): meprobit y x1 x2 x3 || id:

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 =        250
                                                 Obs per group:
                                                              min =          5
                                                              avg =        5.0
                                                              max =          5
Family : Bernoulli                               Number of obs    =      1,250
Link   : probit                                  Acceptance rate  =      .3212
                                                 Efficiency:  min =     .03291
                                                              avg =     .04084
Log marginal-likelihood                                       max =     .04719

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
y            |
          x1 |  .0545741   .0220519    .00104   .0542829   .0107792   .0971753
          x2 | -.0814938   .0389815   .001794  -.0814345  -.1577731  -.0044158
          x3 |  .0489053   .0258258   .001218   .0495041  -.0033026   .0988736
       _cons |  .3057306   .1292624   .007125   .3049666   .0434966   .5513856
-------------+----------------------------------------------------------------
id           |
   U0:sigma2 |   .869336   .1475275   .007987   .8565905   .6122842   1.194495
------------------------------------------------------------------------------
Note: Default priors are used for model parameters.
```

- Our Bayesian results are similar to the classical results because the default priors used for parameters were noninformative.

Random effects

- During Bayesian estimation, random effects are estimated together with other model parameters instead of being predicted after estimation.

- Because there may be many random effects, `bayes` does not report them by default. But we can use option `showreffects()` to display them.

Show random effects

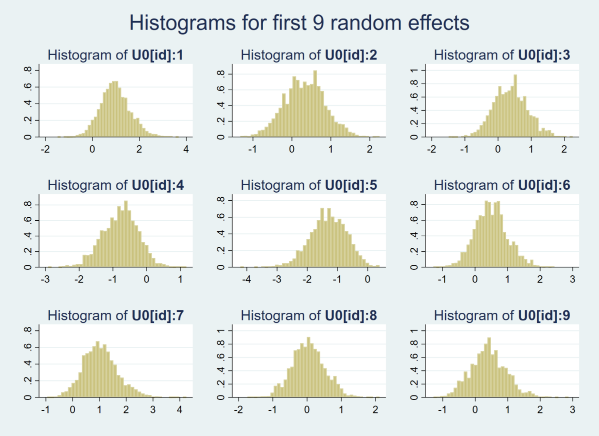
- For instance, let’s display the first 9 random effects.

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

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
y            |
          x1 |  .0545741   .0220519    .00104   .0542829   .0107792   .0971753
          x2 | -.0814938   .0389815   .001794  -.0814345  -.1577731  -.0044158
          x3 |  .0489053   .0258258   .001218   .0495041  -.0033026   .0988736
       _cons |  .3057306   .1292624   .007125   .3049666   .0434966   .5513856
-------------+----------------------------------------------------------------
U0[id]       |
           1 |  .9816318   .6483689   .018095   .9451563  -.1966993    2.35105
           2 |  .3298048   .5280284   .014699   .3250389  -.6906729   1.386742
           3 |  .3808169   .5135901   .015094    .377268  -.5926917   1.464861
           4 |  -.781506   .5283996   .016195  -.7492063  -1.842893   .1953963
           5 | -1.307104   .6082005   .017053  -1.280264  -2.570318  -.1867906
           6 |  .5024583   .5118613   .014101   .4808428  -.4400955   1.577791
           7 |   1.03784    .647897   .016973   .9924323  -.1562301   2.437312
           8 |  .0393935   .4893852   .014986   .0250356  -.8939384   .9904983
           9 |  .4053234   .5520343   .015952   .3918537  -.6268649   1.578649
-------------+----------------------------------------------------------------
id           |
   U0:sigma2 |   .869336   .1475275   .007987   .8565905   .6122842   1.194495
------------------------------------------------------------------------------
```

Histograms for random effects

- Just like other parameters of Bayesian models, we have an entire distribution for each random effect. Let's plot them using, for instance, `bayesgraph histogram`.

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



Efective sample size, autocorrelation, and efficiency

```
. bayesstats ess
Efficiency summaries      MCMC sample size =     10,000
                          Efficiency:  min =     .03291
                                       avg =     .04084
                                       max =     .04719

-----------------------------------------------------
             |        ESS   Corr. time    Efficiency
-------------+---------------------------------------
y            |
          x1 |     449.84        22.23        0.0450
          x2 |     471.94        21.19        0.0472
          x3 |     449.69        22.24        0.0450
       _cons |     329.11        30.39        0.0329
-------------+---------------------------------------
id           |
   U0:sigma2 |     341.21        29.31        0.0341
-----------------------------------------------------
```

- The efficiency is around 3% to 4% for all the main parameters
- Autorrelation seems to be a little high, so we may want to check the diagnostic plots for more detailed analysis, and we may also want to check convergence using multiple chains.

bayesstats grubin

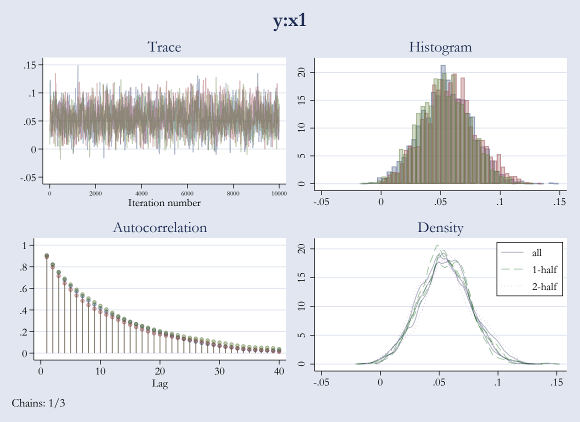
- Let’s check convergence by fitting the model with 3 chains and evaluating the Gelman–Rubin statistic:

```
. quietly bayes, nodots rseed(50) nchains(3): ///
                   meprobit y x1 x2 x3 || id:
. bayesstats grubin

Gelman-Rubin convergence diagnostic

Number of chains     =          3
MCMC size, per chain =     10,000
Max Gelman-Rubin Rc  =   1.008693

------------------------
            |         Rc
------------+-----------
y           |
         x1 |   1.008693
         x2 |   1.001802
         x3 |   1.001238
      _cons |   1.002039
------------+-----------
id          |
   U0:sigma2|   1.004256
------------------------

Convergence rule: Rc < 1.1
```

- The Gelman–Rubin statistic supports convergence for each of the main parameters.

bayesgraph diagnostics

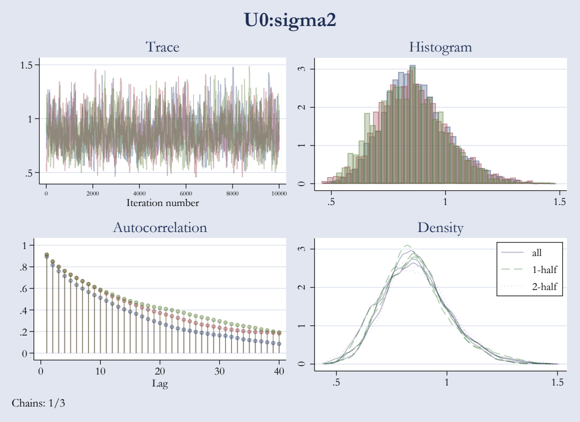
- Let's look at the diagnostic graphs for y:x1.



- All the plots support convergence for y:x1. You should also check y:x2 and y:x3.

bayesgraph diagnostics

- Let's also look at the diagnostic graphs for U0:sigma2:



- All the plots support convergence for U0:sigma2, although the autocorrelation is dying off slower for this parameter.

Bayesian predictions and replications

Use of Bayesian predictions

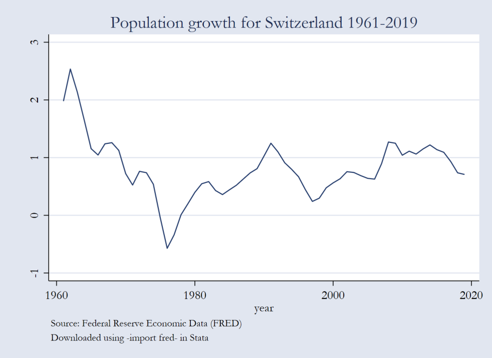
- In model diagnostic
- Optimal predictors in forecasting (Out of sample predictions)
- Optimal classifiers in classification problems
- Missing-data imputation

Computing Bayesian predictions

- Simulate outcome predictions (out of sample)
  - Obtained from posterior predictive distribution of the unobserved (future) data, based on:
    - Posterior distribution for model parameters
    - Likelihood for the outcome given model parameters and data
- Compute and save posterior summaries of simulated outcome.
- Simulate replicates (in sample) and save them in the current dataset.
- Use internal or user-defined Mata functions.
- Use user-defined Stata programs.

Example 6: MCMC sample of replicated outcome

- We can use `bayesreps` to generate a subset of MCMC replicates in the current dataset.
- Replicated data are data we would have observed if we were to repeat the same experiment that produced the observed data.
- The replicates can be used to make comparisons with the observed outcome.
- Let's see how the comparison looks with the estimate for the mean population growth for Switzerland.



Population growth for Switzerland 1961-2019

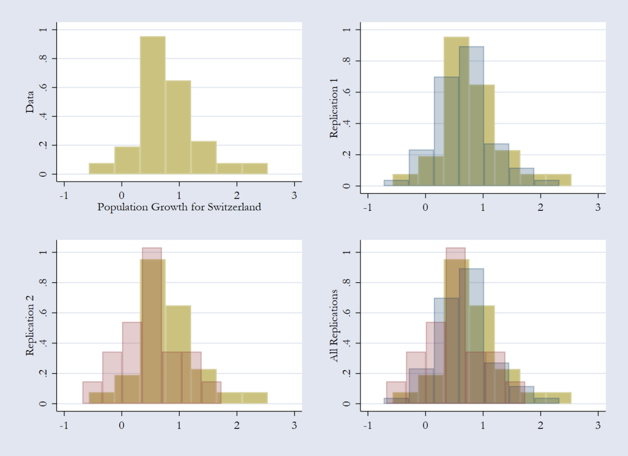
Source: Federal Reserve Economic Data (FRED)
Downloaded using -import fred- in Stata

```
. describe

Contains data from popgr_swiss.dta
  obs:            59
 vars:             4                          18 Nov 2020 12:08
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------------
datestr         str10   %-10s                 observation date
daten           int     %td                   numeric (daily) date
popgr_swiss     float   %9.0g                 Population Growth for Switzerland
year            float   %9.0g
-------------------------------------------------------------------------------
Sorted by: year


. summarize popgr_swiss if year<=1970

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
 popgr_swiss |         10    1.485167    .5701435   .7224038   2.535012


. summarize popgr_swiss if year>1970

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
 popgr_swiss |         49    .6681046    .3961815  -.5715957   1.270618
```

Code for bayesian replications

```
bayesmh popgr_swiss if year>1970, likelihood(normal({.25})) ///
             prior({popgr_swiss:_cons},normal(1.485,.325))   ///
             saving(popgr_mcmc,replace) rseed(1)

// Use -bayesreps- to get two replicates for popgr_swiss
// plot the data along with the replicates.
bayesreps yrep*, rseed(123) nreps(2)

// Plot the data along with the replicates.
twoway histogram popgr_swiss, name(data,replace)       ///
       legend(off) ytitle("Data")
twoway histogram popgr_swiss || histogram yrep1,       ///
       color(navy%25) name(rep1,replace) legend(off)   ///
       ytitle("Replication 1")
twoway histogram popgr_swiss || histogram yrep2,       ///
       color(maroon%25) name(rep2,replace) legend(off) ///
       ytitle("Replication 2")
twoway histogram popgr_swiss ||                        ///
       histogram yrep1, color(navy%25) ||              ///
       histogram yrep2, color(maroon%25) ||,           ///
       name(rep_all,replace) legend(off)               ///
       ytitle("All Replications")
graph combine data rep1 rep2 rep_all
```

We expect to see similar histograms for the data and the replicates:



Example 7.1: Predicted outcome and residuals

- We can use `bayespredict` to get predictions for simulated outcomes and residuals.

```
. quietly bayesmh popgr_swiss if year>1970,                  ///
>            likelihood(normal({.25}))                       ///
>            prior({popgr_swiss:_cons},normal(1.48,.32)) ///
>            saving(popgr_mcmc,replace) rseed(1)
. bayespredict {_ysim} if year>1970,saving(my_ysim,replace) rseed(123)
Computing predictions ...
file my_ysim.dta saved
file my_ysim.ster saved
```

- We can then use `bayesstats summary` to get summaries for the mean of the simulated outcome and residuals.

```
. bayesstats summary @mean({_ysim})        ///
>                 @mean({_resid1}) using my_ysim
Posterior summary statistics                      MCMC sample size =    10,000

------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.      MCSE     Median  [95% Cred. Interval]
-------------+----------------------------------------------------------------
 _ysim1_mean |   .681428   .1022601   .001693   .6798943   .4828026   .8834371
_resid1_mean |  .0002904   .0715301   .000715   .0005509  -.1407139   .1407014
------------------------------------------------------------------------------
.
end of do-file
. do "C:\Users\gas\AppData\Local\Temp\STD25fc_000000.tmp"
```

Example 7.1: Predicted outcome and residuals

- We can use `bayespredict` to get predictions for simulated outcomes and residuals.

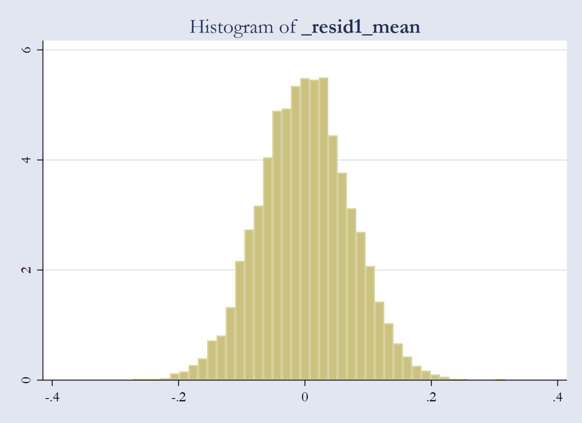
```
. quietly bayesmh popgr_swiss if year>1970,              ///
>            likelihood(normal({.25}))                   ///
>            prior({popgr_swiss:_cons},normal(1.48,.32)) ///
>            saving(popgr_mcmc,replace) rseed(1)
. bayespredict {_ysim} if year>1970,saving(my_ysim,replace) rseed(123)
Computing predictions ...
file my_ysim.dta saved
file my_ysim.ster saved
```

- We can then use `bayesstats summary` to get summaries for the mean of the simulated outcome and residuals.

```
. bayesstats summary @mean({_ysim})       ///
>                    @mean({_resid1}) using my_ysim
Posterior summary statistics                      MCMC sample size =    10,000

--------------------------------------------------------------------------------
             |                                                Equal-tailed
             |      Mean   Std. Dev.     MCSE     Median  [95% Cred. Interval]
-------------+------------------------------------------------------------------
 _ysim1_mean |   .681428   .1022601   .001693   .6798943   .4828026   .8834371
_resid1_mean |  .0002904   .0715301   .000715   .0005509  -.1407139   .1407014
--------------------------------------------------------------------------------

.
end of do-file
. do "C:\Users\gas\AppData\Local\Temp\STD25fc_000000.tmp"
```

We can also get a histogram for the mean of the simulated residuals

```
. bayesgraph histogram @mean({_resid1}) using my_ysim
```



Example 7.2: Posterior predictive p-values (PPPs)

- We can complete the analysis by using `bayesstats ppvalues` to measure discrepancies between the model and the data.

- In general, we should evaluate test quantities that correspond to relevant assumptions for the model.

- PPPs are expected to be close to .5 for a well-fitted model, but in practice PPPs between .05 and .95 are accepted as values that support the goodness of fit for the model.

- Let’s use the mean and variance for the residuals as a test quantity:

```
. bayesstats ppvalues   (mean: @mean({_resid1})) ///
>                       (var:@variance({_resid1})) using my_ysim
Posterior predictive summary   MCMC sample size =    10,000
-----------------------------------------------------------
            T |      Mean   Std. Dev.   E(T_obs)  P(T>=T_obs)
-------------+---------------------------------------------
         mean |  .0002904   .0715301   -.013033        .5458
          var |  .2496989   .0509342   .1569598        .9796
-----------------------------------------------------------
Note: P(T>=T_obs) close to 0 or 1 indicates lack of fit.
```

- For the mean the PPPs supports the model, but the variance it does not support the model.

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