Introduction to Bayesian models with Stata

Ernesto F. L. Amaral
Katherine A. C. Willyard

May 15, 2018

www.ernestoamaral.com/stata2018b.html

TEXAS A&M UNIVERSITY

Bayesian analysis

- Bayesian analysis is a statistical procedure that answers research questions by expressing uncertainty about unknown parameters using probabilities

- It is based on the fundamental assumption that not only the outcome of interest but also all the unknown parameters in a statistical model are essentially random and are subject to prior beliefs

- Observed data sample y is fixed and model parameters θ are random
  - y is viewed as a result of a one-time experiment
  - A parameter is summarized by an entire distribution of values instead of one fixed value as in classical frequentist analysis

How to do Bayesian analysis

- Bayesian analysis starts with the specification of a posterior model
- The posterior model describes the probability distribution of all model parameters conditional on the observed data and some prior knowledge
- The **posterior distribution** has two components
  – A **likelihood**, which includes information about model parameters based on the observed data
  – A **prior**, which includes prior information (before observing the data) about model parameters
- The likelihood and prior models are combined using the Bayes rule to produce the posterior distribution

$$\text{Posterior} \propto \text{Likelihood} \times \text{Prior}$$

Bayes rule

- **Prior distribution: $p(\theta) = \pi(\theta)$**
  - Some prior knowledge about θ
  - Probability distribution of θ
- **Likelihood: $p(y|\theta) = f(y;\theta)$**
  - Observed sample data y about unknown parameter θ
  - Probability density function of y given θ
- **Posterior distribution: $p(\theta|y)$**

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})} = \frac{f(\mathbf{y};\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{m(\mathbf{y})}$$

- **Marginal distribution of y: $p(y) \equiv m(y)$**
  - It does not depend on the parameter of interest θ, so equation can be reduced to

$$p(\theta|y) \propto f(y;\theta)\pi(\theta)$$

Markov chain Monte Carlo

- Posterior distributions are rarely available in analytical forms and often involve multidimensional integrals
  - They are commonly estimated via simulation

- Markov chain Monte Carlo (MCMC) sampling is often used to simulate potentially very complex high-dimensional posterior distributions
  - MCMC is a simulation-based method of estimating posterior distributions
  - It produces a sequence or a chain of simulated values (MCMC estimates) of model parameters from the estimated posterior distribution
  - If the chain "converges", the sequence represents a sample from the desired posterior distribution

MCMC methods in Stata

- There are different MCMC methods to estimate the chains of simulated values

- Two more commonly used MCMC methods are
  - Metropolis-Hastings (MH) algorithm
  - Gibbs algorithm

- MCMC methods in Stata
  - Adaptive MH
  - Adaptive MH with Gibbs updates–hybrid
  - Full Gibbs sampling for some models

Stata's Bayesian commands

Estimation

`bayesian estimation`	Bayesian estimation commands
`bayes`	Bayesian regression models using the `bayes` prefix
`bayesmh`	Bayesian models using MH
`bayesmh evaluators`	User-defined Bayesian models using MH

Convergence tests and graphical summaries

`bayesgraph`	Graphical summaries

Postestimation statistics

`bayesstats ess`	Effective sample sizes and related statistics
`bayesstats summary`	Bayesian summary statistics
`bayesstats ic`	Bayesian information criteria and Bayes factors

Hypothesis testing

`bayestest model`	Hypothesis testing using model posterior probabilities
`bayestest interval`	Interval hypothesis testing

General syntax

- Built-in models
  - Fitting regression models

    bayes: *stata_command* ...

  - Fitting general models

    bayesmh ..., likelihood() prior() ...

- User-defined models
  - Posterior evaluator

    bayesmh ..., evaluator() ...

  - Likelihood evaluator with built-in priors

    bayesmh ..., llevaluator() prior() ...

- Postestimation
  - Features are the same whether you use a built-in model or program your own

Bayesian models in Stata

- Over 50 built-in likelihoods: normal, lognormal, exponential, multivariate normal, probit, logit, oprobit, ologit, Poisson, Bernoulli, binomial, and more
- Many built-in priors: normal, lognormal, uniform, gamma, inverse gamma, exponential, beta, chi square, Jeffreys, multivariate normal, Zellner's g, Wishart, inverse Wishart, multivariate Jeffreys, Bernoulli, discrete, Poisson, flat, and more
- 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

Bayesian estimation in Stata

- Bayesian estimation in Stata is similar to standard estimation, simply prefix command with "bayes:"
- For example, if your estimation command is a linear regression of y on x

```
regress y x
```

- Bayesian estimates for this model can be obtained with

```
bayes: regress y x
```

- You can also refer to "bayesmh" and "bayesmh evaluators" for fitting more general Bayesian models
- The following estimation commands support the bayes prefix...

Command	Entry	Description
Linear regression models		
regress	[BAYES] **bayes: regress**	Linear regression
hetregress	[BAYES] **bayes: hetregress**	Heteroskedastic linear regression
tobit	[BAYES] **bayes: tobit**	Tobit regression
intreg	[BAYES] **bayes: intreg**	Interval regression
truncreg	[BAYES] **bayes: truncreg**	Truncated regression
mvreg	[BAYES] **bayes: mvreg**	Multivariate regression
Binary-response regression models		
logistic	[BAYES] **bayes: logistic**	Logistic regression, reporting odds ratios
logit	[BAYES] **bayes: logit**	Logistic regression, reporting coefficients
probit	[BAYES] **bayes: probit**	Probit regression
cloglog	[BAYES] **bayes: cloglog**	Complementary log-log regression
hetprobit	[BAYES] **bayes: hetprobit**	Heteroskedastic probit regression
binreg	[BAYES] **bayes: binreg**	GLM for the binomial family
biprobit	[BAYES] **bayes: biprobit**	Bivariate probit regression
Ordinal-response regression models		
ologit	[BAYES] **bayes: ologit**	Ordered logistic regression
oprobit	[BAYES] **bayes: oprobit**	Ordered probit regression
zioprobit	[BAYES] **bayes: zioprobit**	Zero-inflated ordered probit regression
Categorical-response regression models		
mlogit	[BAYES] **bayes: mlogit**	Multinomial (polytomous) logistic regression
mprobit	[BAYES] **bayes: mprobit**	Multinomial probit regression
clogit	[BAYES] **bayes: clogit**	Conditional logistic regression
Count-response regression models		
poisson	[BAYES] **bayes: poisson**	Poisson regression
nbreg	[BAYES] **bayes: nbreg**	Negative binomial regression
gnbreg	[BAYES] **bayes: gnbreg**	Generalized negative binomial regression
tpoisson	[BAYES] **bayes: tpoisson**	Truncated Poisson regression
tnbreg	[BAYES] **bayes: tnbreg**	Truncated negative binomial regression
zip	[BAYES] **bayes: zip**	Zero-inflated Poisson regression
zinb	[BAYES] **bayes: zinb**	Zero-inflated negative binomial regression

Generalized linear models		
`glm`	[BAYES] **bayes: glm**	Generalized linear models
Fractional-response regression models		
`fracreg`	[BAYES] **bayes: fracreg**	Fractional response regression
`betareg`	[BAYES] **bayes: betareg**	Beta regression
Survival regression models		
`streg`	[BAYES] **bayes: streg**	Parametric survival models
Sample-selection regression models		
`heckman`	[BAYES] **bayes: heckman**	Heckman selection model
`heckprobit`	[BAYES] **bayes: heckprobit**	Probit regression with sample selection
`heckoprobit`	[BAYES] **bayes: heckoprobit**	Ordered probit model with sample selection
Multilevel regression models		
`mixed`	[BAYES] **bayes: mixed**	Multilevel linear regression
`metobit`	[BAYES] **bayes: metobit**	Multilevel tobit regression
`meintreg`	[BAYES] **bayes: meintreg**	Multilevel interval regression
`melogit`	[BAYES] **bayes: melogit**	Multilevel logistic regression
`meprobit`	[BAYES] **bayes: meprobit**	Multilevel probit regression
`mecloglog`	[BAYES] **bayes: mecloglog**	Multilevel complementary log-log regression
`meologit`	[BAYES] **bayes: meologit**	Multilevel ordered logistic regression
`meoprobit`	[BAYES] **bayes: meoprobit**	Multilevel ordered probit regression
`mepoisson`	[BAYES] **bayes: mepoisson**	Multilevel Poisson regression
`menbreg`	[BAYES] **ba** [illegible]	Multilevel negative binomial regression
`meglm`	[BAYES] **bayes: meglm**	Multilevel generalized linear model
`mestreg`	[BAYES] **bayes: mestreg**	Multilevel parametric survival regression

Go to page 433

Summary

- Stata provides an entire suite of commands for Bayesian analysis
- The **bayesmh** command and the **bayes:** prefix are the main estimation commands
- You can use **bayesmh** to fit built-in models or to program your own
- **bayesgraph diagnostics** produces graphical MCMC diagnostics including trace and auto-correlation plots
- **bayesstats ess** computes MCMC efficiencies for all model parameters
- **bayesstats summary** provides MCMS point and interval estimates for model parameters and their functions
- **bayestest interval** performs interval hypothesis testing
- **bayestest model** computes model posterior probabilities for model comparison
- **bayesstats ic** computes BFs and DICs for model comparison

Example of logistic regression

- Study of risk factors of mother (age and smoke) associated with low birthweight of child (low) from Hosmer, Lemeshow, and Sturdivant (2013, 24)

```
. use lbw, clear
(Hosmer & Lemeshow data)

. describe low age smoke

              storage   display    value
variable name   type    format     label      variable label
-------------------------------------------------------------------------
low             byte    %8.0g                 birthweight<2500g
age             byte    %8.0g                 age of mother
smoke           byte    %9.0g      smoke      smoked during pregnancy
```

Classical logistic regression

```
. logit low age smoke

Iteration 0:   log likelihood =   -117.336
Iteration 1:   log likelihood = -113.66733
Iteration 2:   log likelihood = -113.63815
Iteration 3:   log likelihood = -113.63815

Logistic regression                             Number of obs     =        189
                                                LR chi2(2)        =       7.40
                                                Prob > chi2       =     0.0248
Log likelihood = -113.63815                     Pseudo R2         =     0.0315
```

low	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
age	-.0497792	.031972	-1.56	0.119	-.1124431	.0128846
smoke	.6918486	.3218061	2.15	0.032	.0611202	1.322577
_cons	.0609051	.7573199	0.08	0.936	-1.423415	1.545225

Bayesian logistic regression

- Fit a Bayesian logistic regression using fairly noninformative normal priors for all regression coefficients

```
set seed 14
bayesmh low age smoke, likelihood(logit) prior({low:}, normal(0,10000))
```

```
Bayesian logistic regression                   MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling       Burn-in          =      2,500
                                               MCMC sample size =     10,000
                                               Number of obs    =        189
                                               Acceptance rate  =      .1827
                                               Efficiency:  min =     .06358
                                                            avg =     .06847
Log marginal likelihood = -133.87215                        max =     .07231
```

low	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
age	-.0529104	.0320853	.001193	-.0534339	-.1167257	.0101978
smoke	.7025298	.3220161	.012771	.6947374	.0858349	1.344506
_cons	.1201885	.7574915	.028731	.1204548	-1.39823	1.529904

Bayesian logistic regression

- Fit a Bayesian logistic regression with **`bayes:`** prefix

```
set seed 14
bayes: logit low age smoke
```

```
Bayesian logistic regression                    MCMC iterations  =     12,500
Random-walk Metropolis-Hastings sampling        Burn-in          =      2,500
                                                MCMC sample size =     10,000
                                                Number of obs    =        189
                                                Acceptance rate  =      .1827
                                                Efficiency:  min =     .06358
                                                             avg =     .06847
Log marginal likelihood = -133.87215                         max =     .07231
```

low	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred.	Interval]
age	-.0529104	.0320853	.001193	-.0534339	-.1167257	.0101978
smoke	.7025298	.3220161	.012771	.6947374	.0858349	1.344506
_cons	.1201885	.7574915	.028731	.1204548	-1.39823	1.529904

Note: Default priors are used for model parameters.

Bayesian logistic results

- Results are comparable with the classical logistic regression because we used fairly noninformative priors
- Specifying informative priors may be useful in the presence of perfect predictors
  - E.g. “Logistic regression model: A case of nonidentifiable parameters” (https://www.stata.com/manuals/bayesbayesmh.pdf)
- **`bayesmh`** automatically creates parameters associated with the regression function–regression coefficients–following the style **`{`***depvar*:*varname***`}`**. The intercept **`{`***depvar*:**`_cons}`** is automatically included unless option **`noconstant`** is specified
- In our example, **`bayesmh`** automatically created regression coefficients **`{low:age}`**, **`{low:smoke}`**, and **`{low:_cons}`**
- **`{low:}`** is a shortcut for all parameters with equation label **`low`**
  - We used this shortcut in option **`prior()`** to apply the same normal prior distribution to all coefficients

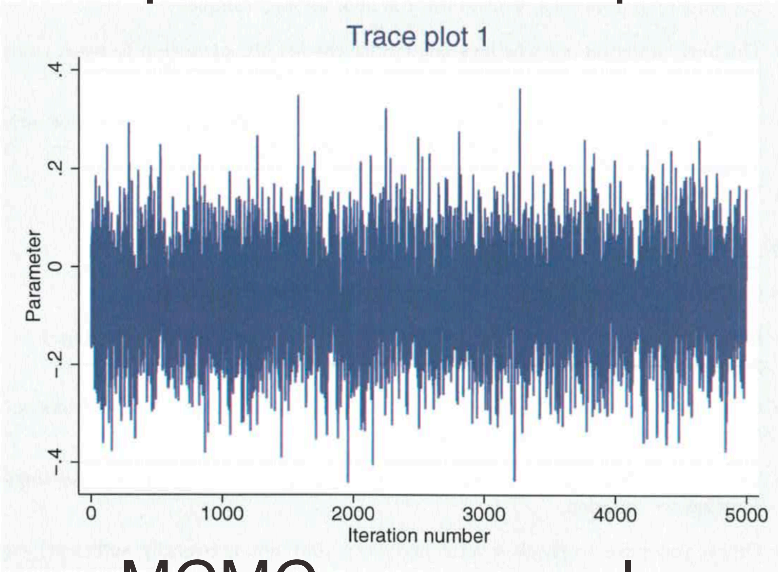
Trace plots

- A trace plot illustrates the values of the simulated parameters against the iteration number and connects consecutive values with a line

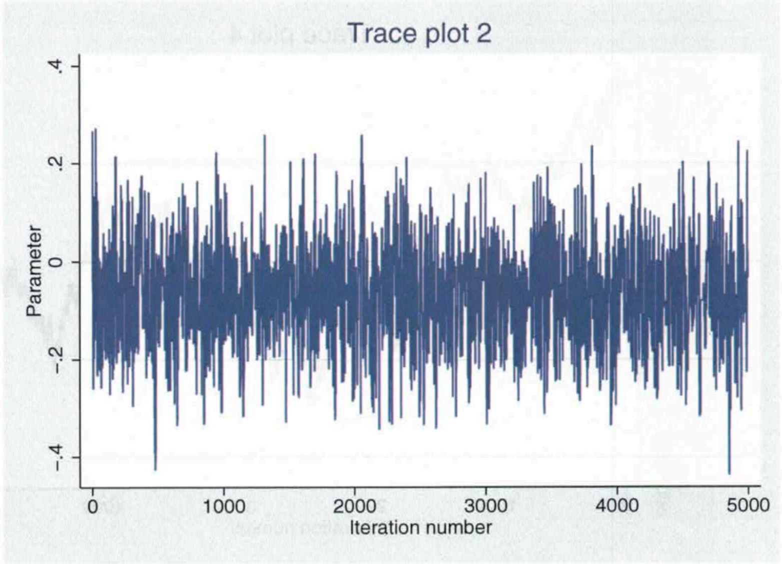
- For a well-mixing parameter, the range of the parameter is traversed rapidly by the MCMC chain, which makes the drawn lines look almost vertical and dense

- Sparseness and trends in the trace plot of a parameter suggest convergence problems

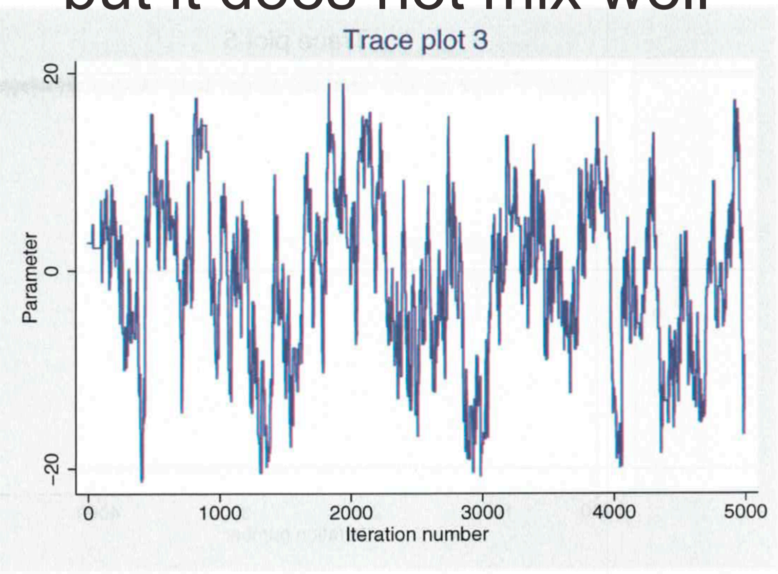
Ideal parameter trace plot



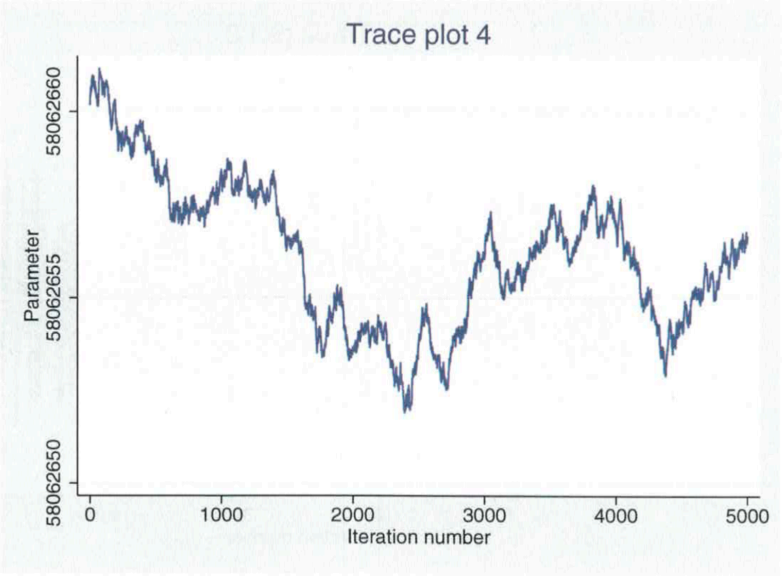
Very good parameter trace plot



MCMC converged,
but it does not mix well



MCMC did not converge



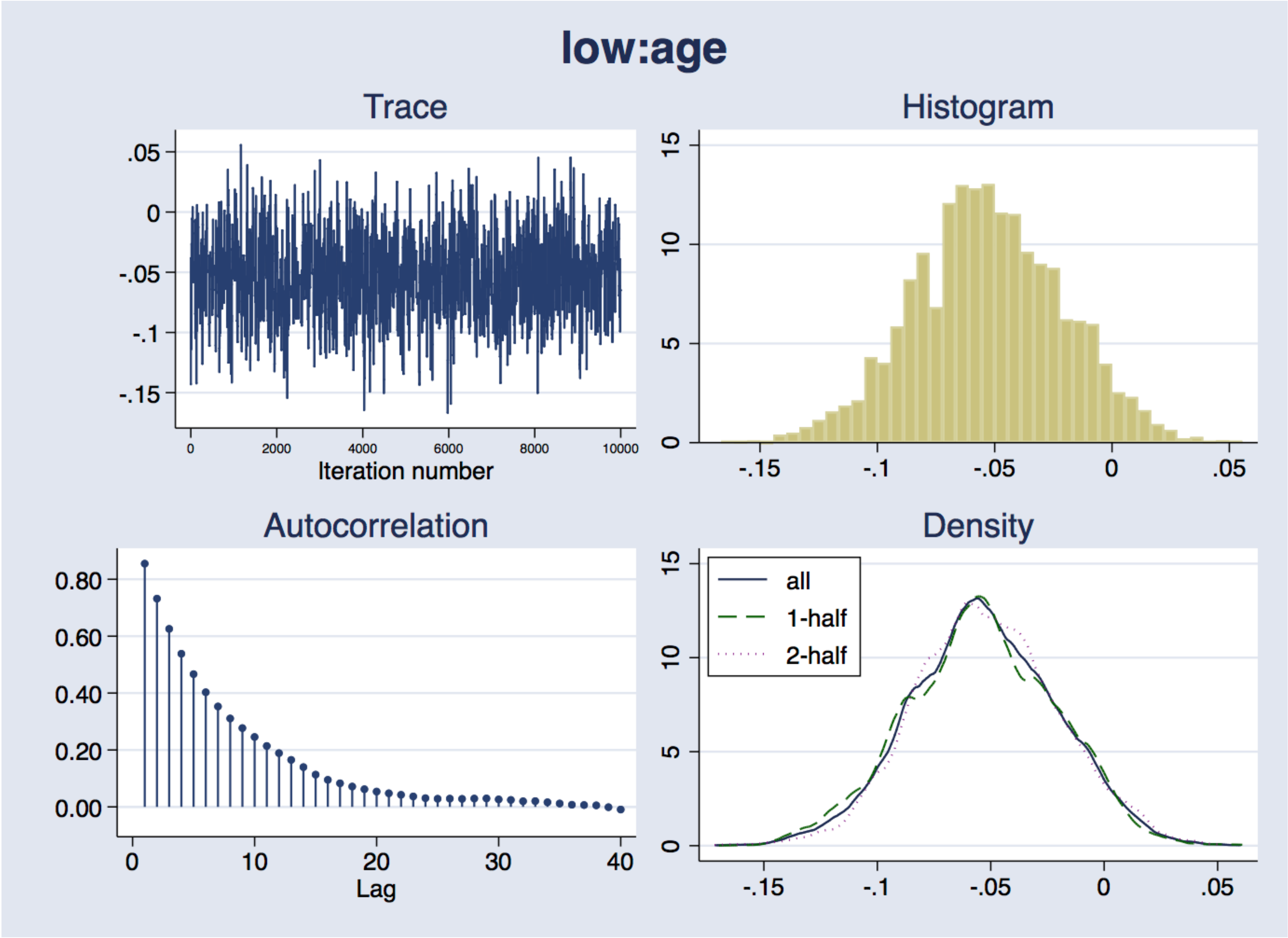
MCMC convergence

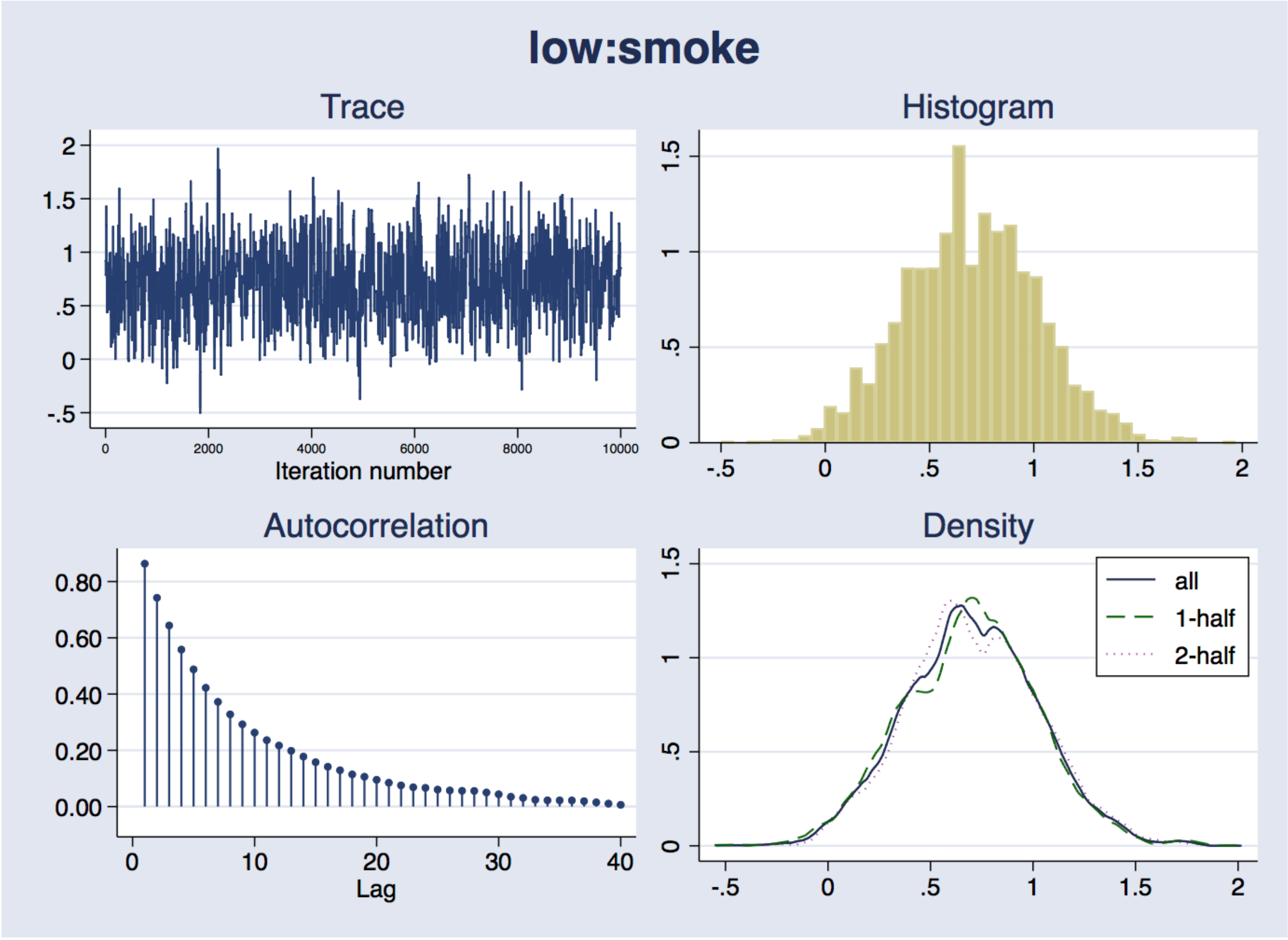
- We can check MCMC convergence for each coefficient separately

```
bayesgraph diagnostics {low:age}
bayesgraph diagnostics {low:smoke}
bayesgraph diagnostics {low:_cons}
```

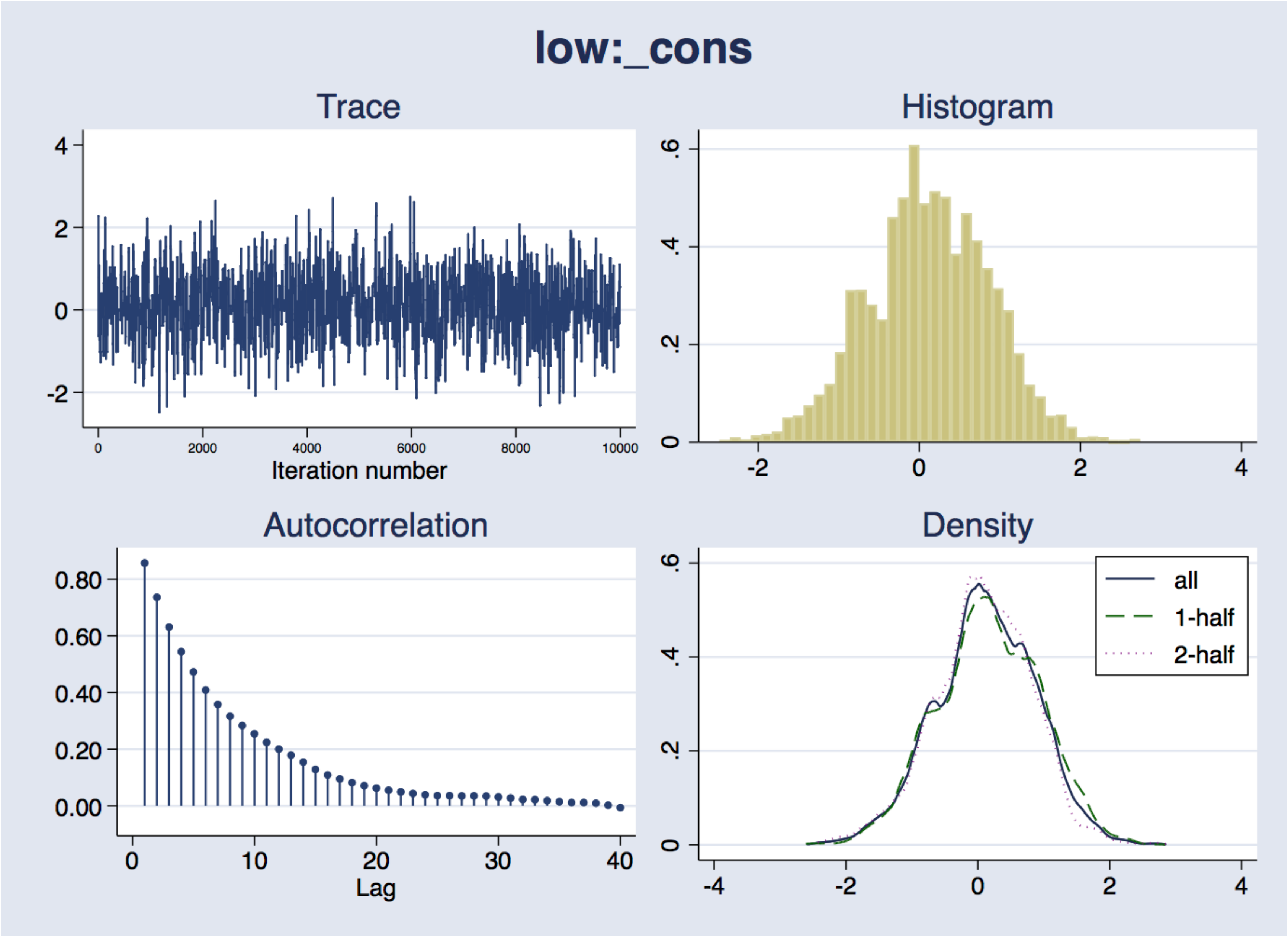
- Or altogether

```
bayesgraph diagnostics {low:}
bayesgraph diagnostics _all
```


low:age
Trace
.05
0
-.05
-.1
-.15
0
2000
4000
6000
8000
10000
Iteration number
Histogram
15
10
5
0
-.15
-.1
-.05
0
.05
Autocorrelation
0.80
0.60
0.40
0.20
0.00
0
10
20
30
40
Lag
Density
all
1-half
2-half
15
10
5
0
-.15
-.1
-.05
0
.05



low:smoke
Trace
2
1.5
1
.5
0
-.5
0
2000
4000
6000
8000
10000
Iteration number
Histogram
1.5
1
.5
0
-.5
0
.5
1
1.5
2
Autocorrelation
0.80
0.60
0.40
0.20
0.00
0
10
20
30
40
Lag
Density
1.5
1
.5
0
-.5
0
.5
1
1.5
2
all
1-half
2-half



low:_cons
Trace
4
2
0
-2
0
2000
4000
6000
8000
10000
Iteration number
Histogram
.6
.4
.2
0
-2
0
2
4
Autocorrelation
0.80
0.60
0.40
0.20
0.00
0
10
20
30
40
Lag
Density
.6
.4
.2
0
-4
-2
0
2
4
all
1-half
2-half

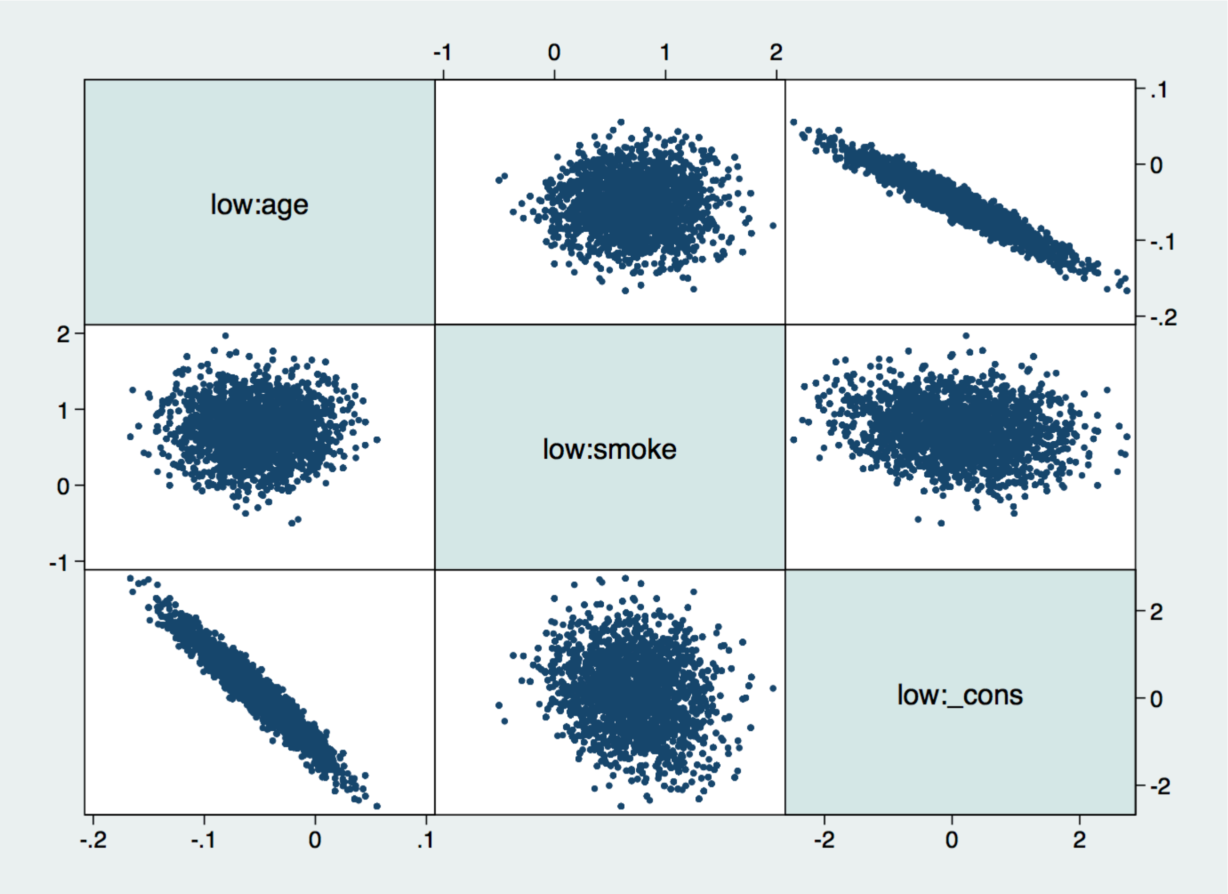
Convergence results

- Trace plots looked reasonable (homogenous)
  - They depict no trends and traverse the parameter range fairly well
- Autocorrelation plots indicated good convergence
  - They reached zero after some lag numbers
  - Specifically, autocorrelations become very small after lag 20
- Density plots illustrated good convergence
  - We want the overall density, the density for the first half and the density for the second half to be similar

Scatterplot matrix

`bayesgraph matrix _all`

- High correlation between constant and age coefficient
  - It generates inefficiency and could affect smoke coefficient



MCMC efficiency

- We can use **`bayesstats ess`** to check MCMC efficiency of regression coefficients
- Effective sample size (ESS)
  - It informs the amount of independent observations we have within MCMC sample size
- Efficiency = ESS / MCMC sample size
  - Efficiency closer to 1 is better
  - Efficiency > 0.1 is good
  - Efficiency < 0.01 is a concern
- If 0.01 > efficiency < 0.1, we have to look at MCSE (digits of precision)
  - Do we want more digits of precision?
  - It depends on the scales of our parameters of estimation

MCMC efficiency results

```
. bayesstats ess

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

low	ESS	Corr. time	Efficiency
age	723.10	13.83	0.0723
smoke	635.79	15.73	0.0636
_cons	695.13	14.39	0.0695

- All efficiencies look reasonable (none below 0.01)
  - Efficiencies decrease if we add more parameters to the model
  - We want to keep them above 0.01, at least for main parameters

- ESS informs that posterior estimates are based on at least 600 independent observations for each coefficient

Functions of model parameters

- We can use **`bayesstats summary`** to obtain estimates of any function of model parameters
- E.g., estimate odds ratios (exponentiated coefficients)

```
. bayesstats summary (OR_age:exp({low:age})) (OR_smoke:exp({low:smoke}))

Posterior summary statistics                      MCMC sample size =    10,000

      OR_age : exp({low:age})
    OR_smoke : exp({low:smoke})
```

	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred. Interval]	
OR_age	.9489532	.0304503	.001134	.9479686	.8898292	1.01025
OR_smoke	2.127093	.7120785	.02777	2.003183	1.089626	3.836291

Multiple chains

- Run multiple chains and compute Gelman-Rubin statistic to verify convergence to a single stationary distribution

```
***Chain 1
bayesmh low age smoke, likelihood(logit) ///
                       prior({low:}, normal(0,10000)) rseed(14) ///
                            mcmcsize(20000) saving(chain1_mcmc, replace) ///
                            initial({low:} 0)
estimates store chain1


***Chain 2
bayesmh low age smoke, likelihood(logit) ///
                       prior({low:}, normal(0,10000)) rseed(14) ///
                            mcmcsize(20000) saving(chain2_mcmc, replace) ///
                            initial({low:} 10)
estimates store chain2


***Chain 3
bayesmh low age smoke, likelihood(logit) ///
                       prior({low:}, normal(0,10000)) rseed(14) ///
                            mcmcsize(20000) saving(chain3_mcmc, replace) ///
                            initial({low:} -10)
estimates store chain3
```

Gelman-Rubin statistic

```
***Install command
net install grubin, from(http://www.stata.com/users/nbalov)

***Estimate Gelman-Rubin statistic
grubin, estnames(chain1 chain2 chain3)
```

Gelman-Rubin convergence diagnostic

MCMC sample size = 20000
Number of chains = 3

	Rc	95% Ru
low		
age	1.000179	1.000104
smoke	1.000558	1.000161
_cons	1.000346	1.000114

- All estimated Rc values are close to 1, which indicates that there is convergence

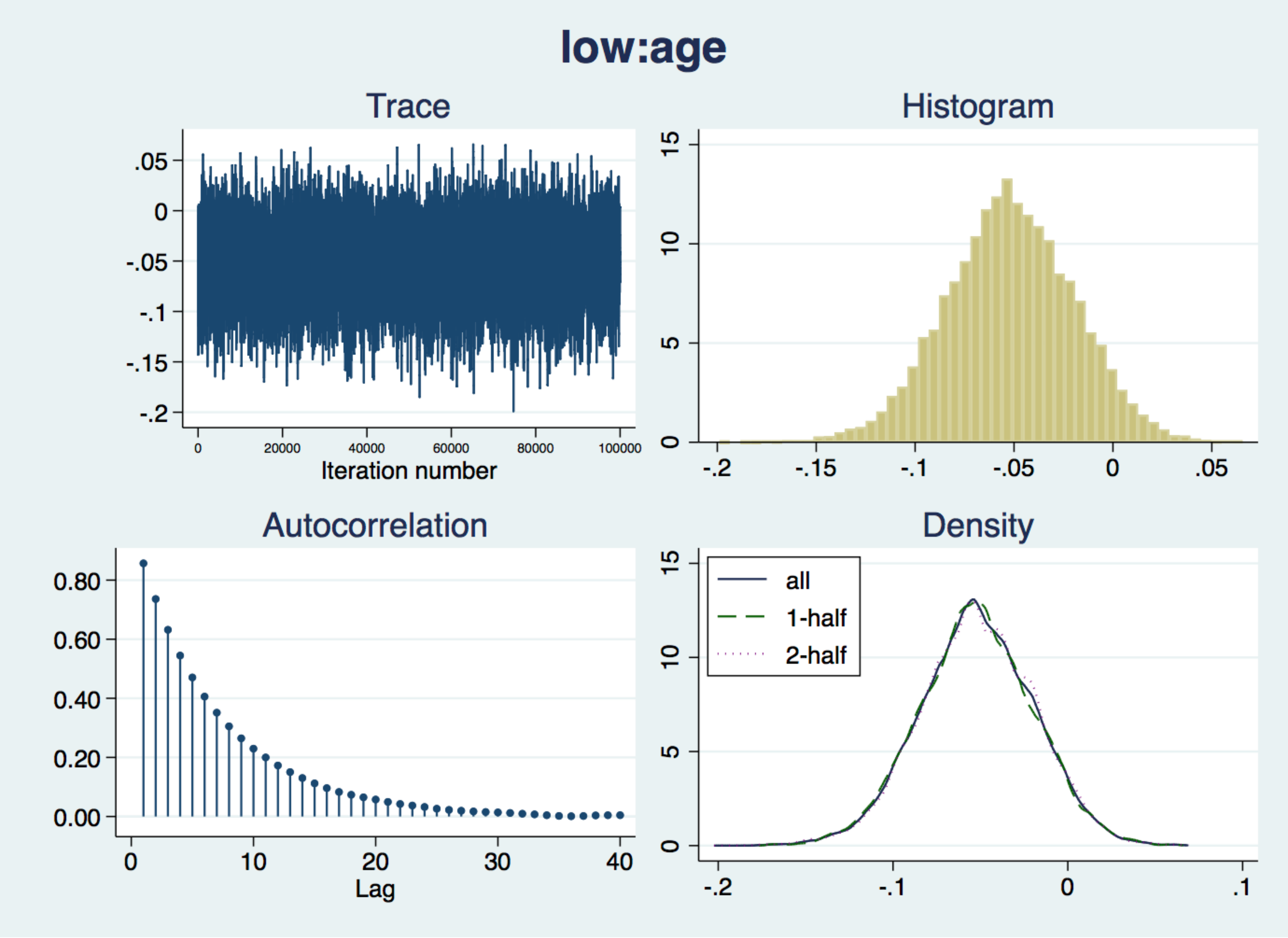
Increase MCMC sample size

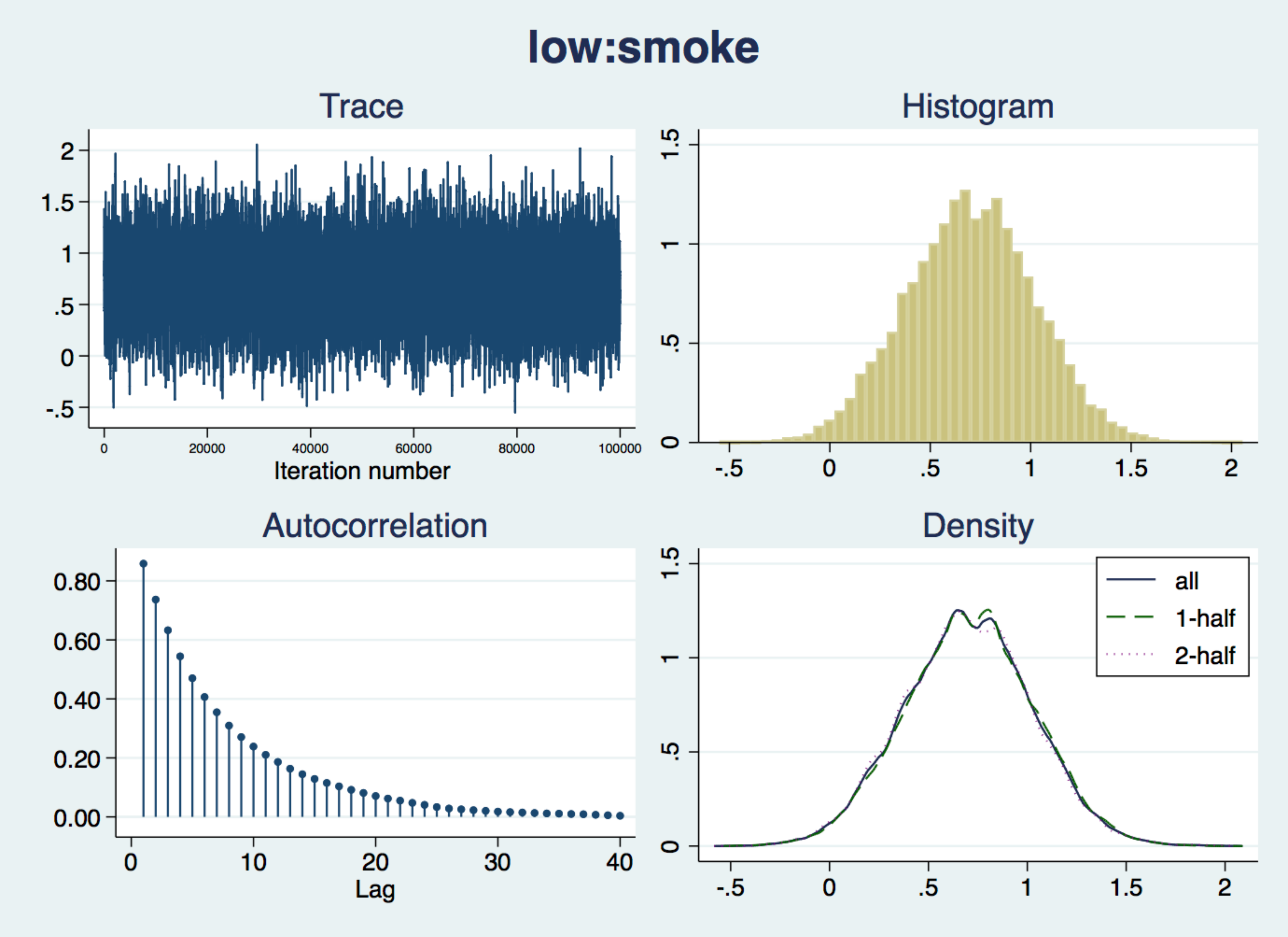
- We can increase MCMC sample size to improve precision of our posterior estimates (reduce MCSE)

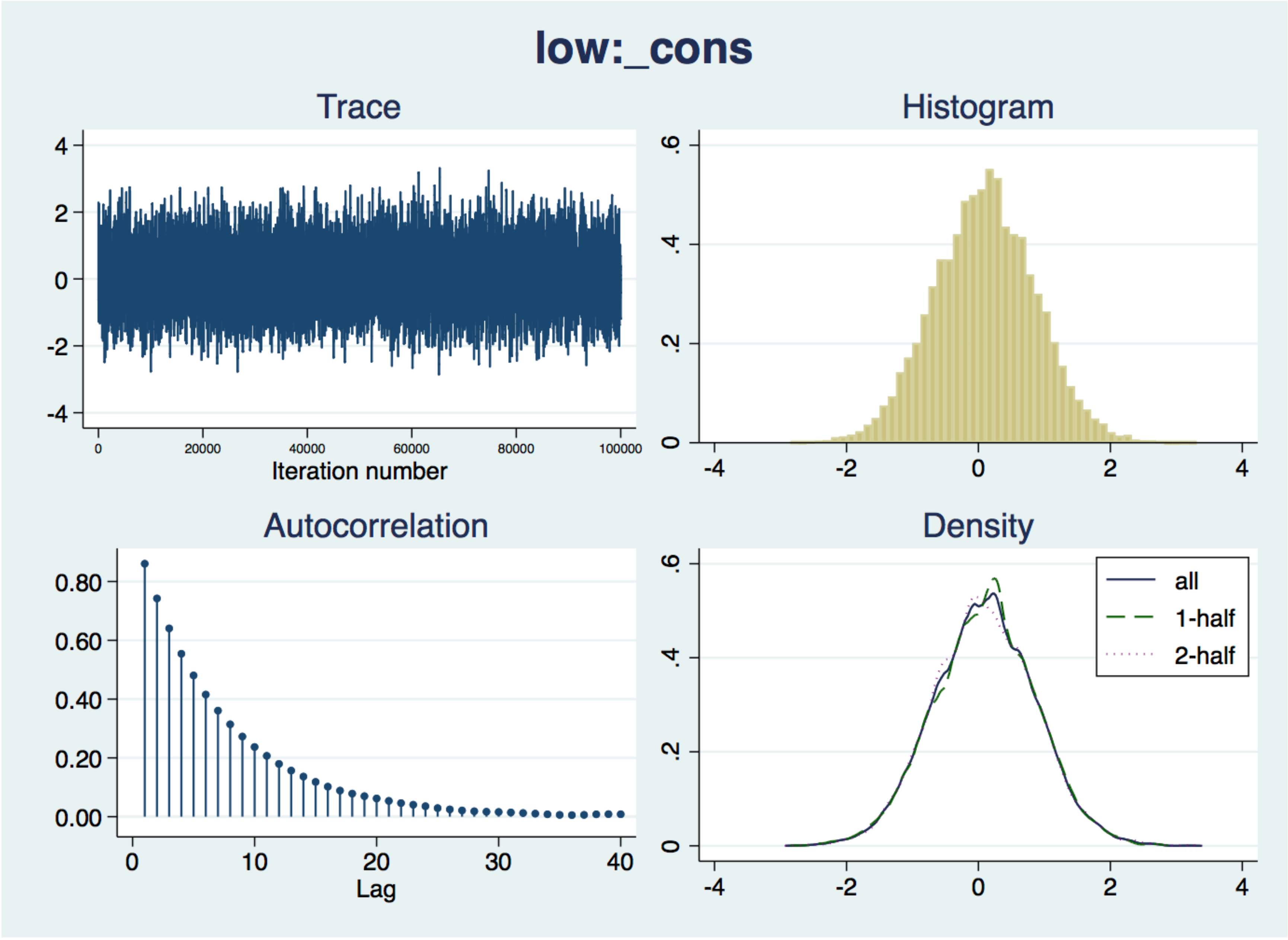
```
set seed 14
bayesmh low age smoke, likelihood(logit) ///
                       prior({low:}, normal(0,10000)) ///
                        mcmcsize(100000)
```

```
Bayesian logistic regression                    MCMC iterations  =    102,500
Random-walk Metropolis-Hastings sampling        Burn-in          =      2,500
                                                MCMC sample size =    100,000
                                                Number of obs    =        189
                                                Acceptance rate  =      .1887
                                                Efficiency:  min =     .07101
                                                             avg =     .07254
Log marginal likelihood = -133.81762                         max =     .07434
```

low	Mean	Std. Dev.	MCSE	Median	Equal-tailed [95% Cred. Interval]	
age	-.0520744	.0327172	.000379	-.0522821	-.117341	.0109098
smoke	.702268	.3242447	.003848	.7017357	.0716997	1.336714
_cons	.0954346	.7756196	.009123	.099679	-1.417087	1.625152


low:age
Trace
.05
0
-.05
-.1
-.15
-.2
0
20000
40000
60000
80000
100000
Iteration number
Histogram
15
10
5
0
-.2
-.15
-.1
-.05
0
.05
Autocorrelation
0.80
0.60
0.40
0.20
0.00
0
10
20
30
40
Lag
Density
all
1-half
2-half
15
10
5
0
-.2
-.1
0
.1


low:smoke
Trace
Histogram
Iteration number
Autocorrelation
Density
Lag
all
1-half
2-half

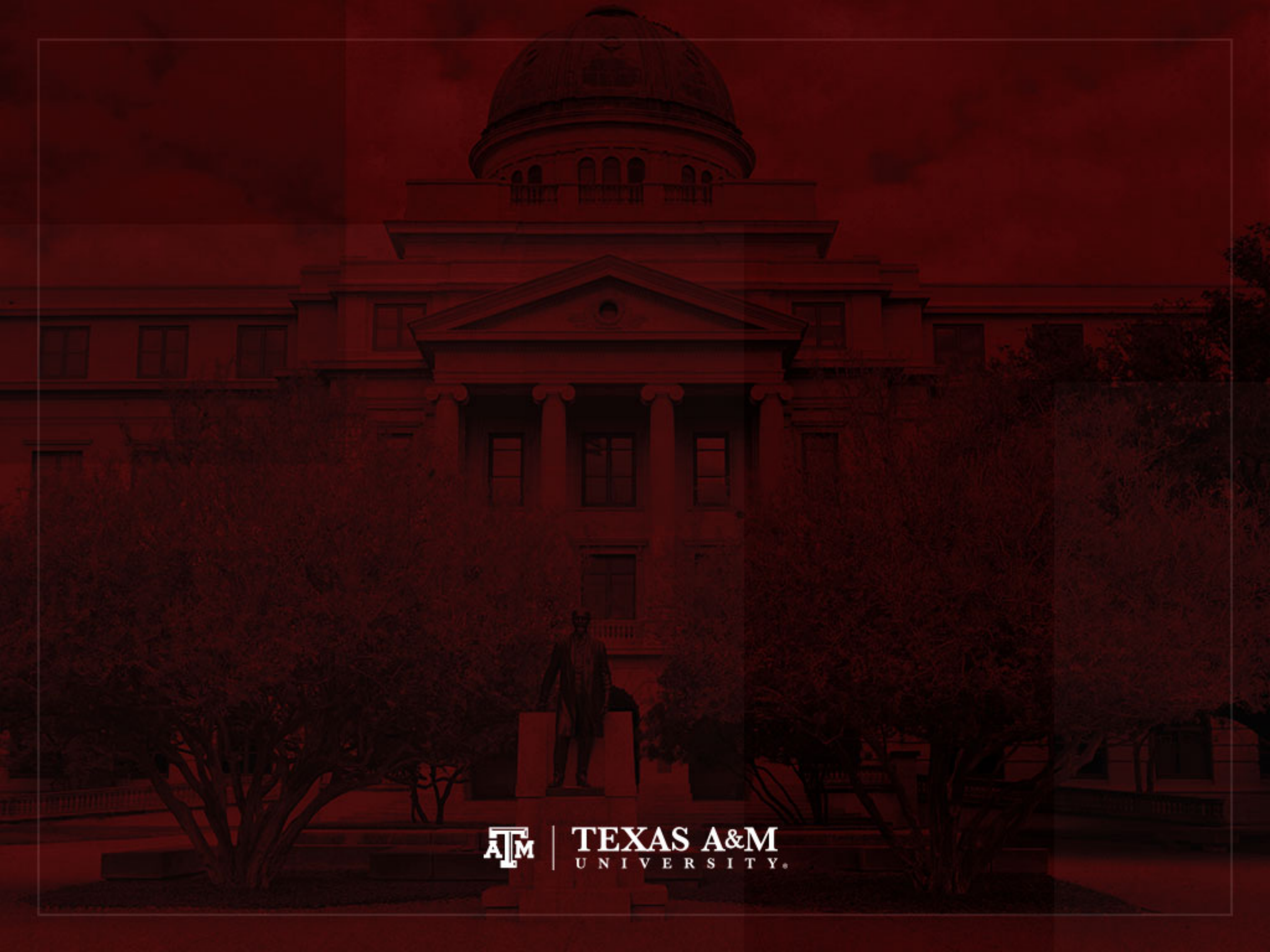


low:_cons
Trace
4
2
0
-2
-4
0
20000
40000
60000
80000
100000
Iteration number
Histogram
.6
.4
.2
0
-4
-2
0
2
4
Autocorrelation
0.80
0.60
0.40
0.20
0.00
0
10
20
30
40
Lag
Density
all
1-half
2-half

References

Marchenko, Yulia V. 2018. *Introduction to Bayesian analysis using Stata*. Web-based training, May 1–4. College Station: StataCorp LLC.

StataCorp. 2017. *Stata Bayesian Analysis Reference Manual: Release 15*. College Station: StataCorp LLC. (https://www.stata.com/manuals/bayes.pdf)


TEXAS A&M
UNIVERSITY