Bayesian Econometrics

Andrés Ramírez Hassan

Universidad Eafit
Departamento de Economía

March 10, 2017

Outline

Random Variable Generation

- Simulation methods allows expanding the scope of Bayesian inference. Because it's usual that the posterior distribution does not have known form.
- Methods of simulation are based on the production of random variables, originally independent random variables, that are distributed according to a distribution f that is not necessarily known.[1]

[1] Robert, C. Casella G. (2004). 'Monte Carlo Statistical Methods '. *Springer*. Second Edition, pag 36.

Random Variable Generation
Probability Integral Transform Method

The most basic method of generating samples takes advantage of the ability of computers to generate values that can be regarded as drawn independently from a uniform distribution on $(0, 1)$, $U(0, 1)$. Such numbers are called pseudo-random numbers, because they are produced as deterministic sequences, but they reproduce the behavior of an iid sample from uniform variable random (see Casella (2004)).[2]

[2]

- Robert, C. Casella G. (2004). 'Monte Carlo Statistical Methods '. *Springer*. Second Edition, pag 36.
- Greenberg, E. (2008). 'Introduction to Bayesian Econometrics '. *Springer*. pag 63.

Ramdom Variable Generation

The Inverse Transform

Probability integral transformation allows us to transform any random variable into a uniform random variable and, more importantly, vice versa. For example, suppose we wish to draw a sample of values from a random variable that has d.f $F(\cdot)$, assumed to be nondecreasing.

$$F(x) = \int_{-\infty}^{x} f(t)dt$$

Consider the distribution of X, which is obtained by drawing U from $U(0,1)$ and setting $X = F^{-1}(U)$, which implies $U = F(X)$

Definition

For a non-decresing function on $\mathbb{R}$ the *generalized inverse* of F, F^{-1} is the function defined by,

$$F^{-1}(u) = inf\{x : F(x) \geqslant u\}$$

Lemma

If $U \sim U(0,1)$ then the random variable $F^{-1}(U)$ has the distribution F

Proof. For all $u \in [0,1]$ and for all $x \in F^{-1}([0,1])$, the generalized inverse satisfies $F(F^{-1}(u)) \geqslant u$ and $F^{-1}(F(x)) \leqslant x$, therefore

$$\{(u,x) : F^{-1}(u) \leqslant x\} = \{(u,x) : F(x) \geqslant u\}$$

and,

$$P(F^{-1}(U) \leqslant x) = P(U \leqslant F(x)) = F(x)$$

Algorithm: Probability integral transform method

1. Draw u from $U(0,1)$.
2. Return $y = F^{-1}(u)$ as a draw from $f(y)$

Example 1. If $X \sim Exp(1)$, $F(x) = 1 - e^{-x}$. Solving for x in $u = 1 - e^{-x}$ gives $x = -log(1-u)$. Therefore, if $U \sim U(0,1)$, then

$$X = -log(U) \sim Exp(1)$$

Random Variable Generation
Probability Integral Transform Method

Example 2. Suppose we wish to draw a sample from a random variable y with density function

$$f(y) = \left\{ \begin{array}{ll} \frac{3}{8}y^2 & \textbf{if} \quad 0 \leqslant y \leqslant 2, \\ \\ 0, & \textbf{otherwise}, \end{array} \right\}$$

We first find the c.d.f for $0 \leqslant y \leqslant 2$ by computing

Random Variable Generation
Probability Integral Transform Method

$$F(y) = \frac{3}{8}\int_0^y t^2 dt = \frac{1}{8}y^3.$$

The next step is to draw a value U from $U(0,1)$ and set $U = \frac{1}{8}Y^3$. We then solve to find $Y = 2U^{1/3}$, which is a draw from $f(y)$.

Truncated distribution Suppose X has a d.f. $F(X)$ and that we wish to generate values of X restricted to $c_1 \leq X \leq c_2$. The distribution of the truncated values is $[F(X) - F(c_1)] / [F(c_2) - F(c_1)]$ for $c_1 \leq X \leq c_2$. We generate $U \sim U(0, 1)$ and set,

$$U = \frac{F(X) - F(c_1)}{F(c_2) - F(c_1)},$$

which implies that,

$$X = F^{-1}(F(c_1) + U\,[F(c_2) - F(c_1)])$$

is a drawing from the truncated distribution.

Multivariate Simulation

The multivariate most studied is the multivariate normal $N_p(\mu, \Sigma)$. To draw a sample from $N_p(\mu, \Sigma)$, first draw p values from $N(0, 1)$ and place them in a $p \times 1$ vector Z, so that $Z \sim N_p(0, I_p)$. Next write $\Sigma = C'C$, where C is a $p \times p$ upper-triangular Cholesky matrix. Finally, compute $X = \mu + C'Z$, then

$$X \sim N_p(\mu, \Sigma)$$

Method of Composition

The method of composition uses the relationship

$$f(x) = \int g(x|y)h(y)dy,$$

where f, g, and h are densities. The method is useful when we know how to sample y from $h(y)$ and x from $g(x|y)$. By drawing a y from $h(y)$ and then an x from $g(x|y)$, the value of x is a drawing from $f(x)$ (see Greenberg (2008)).[3]

3

- Greenberg, E. (2008). 'Introduction to Bayesian Econometrics '. *Springer*. pag 65.

Method of Composition

Example: For the heteroskedastic regression linear model we will show that if $\varepsilon_i|\lambda_i \sim N(0, \lambda_i^{-1}\sigma^2)$, $\lambda_i \sim G(\nu/2, \nu/2)$, and

$$f(\varepsilon_i|\sigma^2) = \int g(\varepsilon_i|\lambda_i, \sigma^2)h(\lambda_i)d\lambda_i,$$

where $g(\varepsilon_i|\lambda_i, \sigma^2)$ is the density function of $N(\varepsilon_i|0, \lambda_i^{-1}\sigma^2)$ and $h(\lambda_i)$ is the density function of $G(\lambda_i|\nu/2, \nu/2)$, then $f(\varepsilon_i|\sigma^2)$ is the density function of $t(v, 0, \sigma^2)$. This result shows that we can simulate draws from a t-distribution with ν degrees of freedom if we know how to simulate draws from a gamma distribution and from a normal distribution.

Accept- Reject

There are many distributions for which the inverse transform method fails to generate the required random variables. For these cases, we must turn to *indirect* methods, that is, methods in which we generate a candidate random variable and only accept it subject to passing a test.

The fundamental theorem of simulation

Since $f(x) = \int_0^{f(x)} du$, then simulating $X \sim f(x)$ is equivalent to simulate

$$(X, U) \sim U\{(x, u) : 0 < u < f(x)\}$$

Accept-Reject

The accept-reject algorithm can be used to simulate values from a density function $f(\cdot)$ (called *target density*). We use a simpler density g (called *instrumental* or *candidate density*), to generate the random variable. The only constraints we impose on this candidate density g are that,

1. f and g have compatible supports (i.e, $g(x) > 0$ when $f(x) > 0$).
2. There is a constant c with $f(x)/g(x) \leq c$ for all x.

Algorithm: Accept-Reject method

1. Generate $Y \sim g$, $U \sim U(0,1)$;
2. Accept $X = Y$ if $U \leq f(Y)/cg(Y)$;
3. Return to 1 otherwise.

Why this method work?. Consider the distribution of the accepted values of y, $h[y|u \leq f(y)/cg(y)]$. By Bayes theorem and the property of the uniform distribution, $P(u \leq t) = t$, $0 \leq t \leq 1$, we have

$$\begin{aligned} h[y|u \leq f(y)/cg(y)] &= \frac{P[u \leq f(y)/cg(y)|y]g(y)}{\int P[u \leq f(y)/cg(y)|y]g(y)dy} \\ &= \frac{[f(y)/cg(y)]g(y)}{(1/c)\int f(y)dy} \\ &= f(y). \end{aligned}$$

Note that

$$\int P[u \leq f(y)/cg(y)|y]g(y)dy = \frac{1}{c}$$

is the probability that a generated value of y is accepted.

Accept-Reject Algorithm

Example 3. Let the target density be $N(0, 1)$ and the proposal density be the Laplace distribution, $g(y) = (1/2)e^{-|y|}$.

Importance Sampling

Suppose that $X \sim f(X)$ and we wish to estimate

$$E[g(X)] = \int g(x)f(x)dx,$$

but the integral is not computable analytically and the method of composition is not available because we cannot sample from $f(x)$. The importance sampling method, a type of Monte Carlo integration, works as follows.

Importance Sampling

Let $h(X)$ be a distribution from which we know how to simulate and consider the integral

$$E[g(X)] = \int \frac{g(x)f(x)}{h(x)} h(x)dx.$$

This integral can be approximated by drawing a sample of G values from $h(X)$, with values $X^{(g)}$, and computing

Importance Sampling

$$E[g(X)] \approx \frac{1}{G} \sum g(X^{(g)}) \frac{f(X^{(g)})}{h(X^{(g)})}.$$

This expression can be regarded as a weighted average of the $g(X^{(g)})$, where the importance weights are $f(X^{(g)})/h(X^{(g)})$. The main issue in implementation of importance sampling is the choice of $h(\cdot)$. To find the suitable distribution we examine the variance of the estimate.[4]

4

- Greenberg, E. (2008). 'Introduction to Bayesian Econometrics '. *Springer*. pag 70.

Importance Sampling

Since $var(\hat{g}) = E(\hat{g}^2) - E(\hat{g})^2$ and the latter converges to $E[g(X)]^2$, we may concetrate on

$$E(\hat{g}^2) = \int g(x)^2 \left(\frac{f(x)}{h(x)}\right)^2 h(x)dx.$$

This integral is large when $f(x)/h(x)$ is large, a situation that tends to occur when the tail values of $h(\cdot)$ are very small compared to the tail values of $f(\cdot)$. In general, $Var(\hat{g})$ is small when $f(\cdot)/g(\cdot)$ does not vary greatly.

Importance Sampling

Example: we wish to approximate $E[(1+x^2)^{-1}]$, where $x \sim exp(1)$, truncated to $[0,1]$; that is, we approximate the integral

$$\frac{1}{1-e^{-1}} \int_0^1 \frac{1}{1+x^2} e^{-x} dx.$$

Importance Sampling

We choose as an importance function $Beta(2,3)$ because it is defined on $[0,1]$ and because, for this choice of parameters, the match between the beta function and target density is good over part of the $[0,1]$ interval. Algorithm:

- Generate a sample of G values, $X^{(1)}, \ldots, X^{(G)}$ from $Beta(2,3)$.
- Calculate

$$\frac{1}{G}\sum_{1}^{G}\left(\frac{1}{1+(X^{(g)})^2}\right)\left(\frac{e^{-X^{(g)}}}{1-e^{-1}}\right)\left(\frac{B(2,3)}{X^{(g)}(1-(X^{(g)})^2)}\right).$$

Importance Sampling

By applaying the previous algorithm and setting $G = 10000$, we obtain an estimate of 0.8268.

Finite State Spaces

Consider a stochastic process indexed by t, X_t that takes values in the finite set $S = [1, 2, ..., s]$.

p_{ij} is the probability that $X_{t+1} = j$ given that $X_t = i$ (p_{ij} is a transition probability).

$$p_{ij} = P(X_{t+1} = j | X_t = i), i, j \in S$$

Additionally, since the process remains in S:

$$\sum_{j=1}^{s} p_{ij} = 1$$

Finite State Spaces

The $s \times s$ transition matrix:

$$P = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}$$

where the ith row represents the distribution of the process at $t+1$, given that it is in state i at t.

The distribution of the state at $t+2$ $p_{ij}^{(2)}$ is given by the expression:

$$p_{ij}^{(2)} = \sum_k p_{ik} p_{kj}$$

The matrix of $p_{ij}^{(2)}$ is given by $PP \equiv P^2$. The values of $p_{ij}^{(n)}$ are the ijth entries in the matrix P^n.

Finite State Spaces

When $p_{ij} = p_j$ for all i, the matrix is of completely random motion or independence.
If $p_{ij}^{(n)} > 0$ for some $n \geq 1$, j is accessible from i:

$$i \longrightarrow j$$

If $i \longrightarrow j$ and $j \longrightarrow i$, then: $i \longleftrightarrow j$

Finite State Spaces

An **irreducible** Markov process is a process where starting from state i, the process can reach any other state with positive probability.
Another important property of a chain is the periodicity. For example, whenever there are positive probabilities of returning to a state in either of two subsets exist only at even values of n. If the period is 1 for all states, the chain is said to be **aperiodic**.
More formally, if $i \longrightarrow j$, then the period of i is the greatest common divisor of the integers in the set $A = [n \geq 1 : p_{ij}^{(n)} > 0]$. If d_i is the period of i, then $p_{ii}^{(n)} = 0$ whenever n is not a multiple of d_i, and d_i is the largest integer with this property. Note that a chain is aperiodic if $p_{ij}^{(n)} \geq 0$ for all i and for sufficiently large n.

Finite State Spaces

Markov Chains Monte Carlo methods (MCMC) are based on the following statement. $\pi = (\pi_1, \pi_2, \ldots, \pi_s)'$ is an invariant distribution for P if $\pi' = \pi' P$, or:

$$\pi_j = \sum_i \pi_i p_{ij}, \quad j = 1, ..., s$$

It can be interpreted as the probability of starting the process at state i with probability π_i and then moving to state j with distribution p_{ij}.
A necessary condition for P being a unique invariant distribution is to be irreducible.

Finite State Spaces

Theorem (Theorem 6.1)

Suppose S is finite and $p_{ij}>0$ for all i,j. Then there exists a unique probability distribution π_j, $j \in S$, such that $\sum_i \pi_i p_{ij} = \pi_j$ for all $j \in S$. Moreover,

$$|p_{ij}^{(n)} - \pi_j| \leq r^n$$

where $0<r<1$ for all i, j and $n \geq 1$.

In a finite state space with positive probabilities there is a unique invariant distribution, and p_{ij}^n convergences at a geometric rate.

Finite State Spaces

If we can find a Markov process for which the invariant distribution is the target distribution, we can simulate draws from the process to generate values from the target distribution.

Finite State Spaces

Theorem (Theorem 6.2)

Let P be irreducible and aperiodic over a finite state space. Then there is a unique probability distribution π such that $\sum_i \pi_i p_{ij} = \pi_j$ for all $j \in S$ and

$$|p_{ij}^{(n)} - \pi_j| \leq r^{n/\nu}$$

for all $i,j \in S$, where $0<r<1$, for some positive integer ν

Countable State Spaces

Irreducibility and aperiodicity no longer imply the existence of a unique invariant distribution when S is countable but not finite.
Let $P_j(A)$ denote the probability that event A occurs, given that the process starts at j. Then state j is called **recurrent** if:

$$P_j(X_n = j \quad i.o) = 1$$

Where i.o. means "infinitely often".
The latter means that the process will return to state j an infinite number of times with probability 1. If a state is not recurrent, it is then called *transient*.

Countable State Spaces

Recurrence is not strong enough to imply a unique invariant distribution. To specify a stronger condition, let $\tau_j^{(1)}$ be the time it takes for the process to make its first return to state j:

$$\tau_j^{(1)} = min\{n > 0 : X_n = j\}$$

A state j is called **positive recurrent** if $E\tau_j^{(1)} < \infty$.
Otherwise, it is **null recurrent**.

Countable State Spaces

Theorem (Theorem 6.3)

Assume that the process is irreducible. Then:

1. *If all states are recurrent, they are either all positive recurrent or all null recurrent.*
2. *There exists an invariant distribution if and only if all states are positive recurrent. In that case, the invariant distribution π is unique and is given by:*

$$\pi_j = (E_j \tau_j^{(1)})^{-1}$$

3. *In case the states are positive recurrent, for any initial distribution, if $E_\pi |f(X_1)| < \infty$*

$$\lim_{n \longrightarrow \infty} \frac{1}{n} \sum_{m=1}^{n} f(X_m) = E_\pi f(X_1)$$

Countable State Spaces

Under the conditions stated in the theorem, we know that there is a unique invariant distribution and that averages of functions evaluated at sample values converge to their expected values under the invariant distribution.

Since a possible function is the indicator function $1(X_n = i)$ which has expected value π_i. This value can be estimated from sample data.

Countable State Spaces

Theorem (Theorem 6.4)

If P is an aperiodic recurrent chain, $\lim_{n \longrightarrow \infty} P^{(n)}$ exists. If P is an aperiodic positive-recurrent chain, then $\lim_{n \longrightarrow \infty} P^{(n)} = A$, where A is a matrix whose rows are the invariant distribution

Theorem (Theorem 6.5)

Suppose P is π-irreducible and that π is an invariant distribution for P. Then P is positive recurrent and π is the unique invariant distribution of P. If P is also aperiodic, then for π-almost all x,

$$||P^n(x, .) - \pi|| \longrightarrow 0.$$

Countable State Spaces

In the latter theorem, which also applies to the continuous case, π-irreducible means that for some n, $P^n(x, A)>0$ for any set A such that $\pi(A)>0$. This implies that recurrence need not be assumed explicitly if it is known that an invariant distribution exists.

Continuous State Spaces

Now suppose that the states of a Markov process take values in $\mathcal{R}$. The counterpart of the transition probabilities is the *transition kernel* or *transition density* $p(x, y)$. It is denoted by $p(x, y)$ because it is the counterpart of p_{ij}, but it is more instructive to interpret it as the conditional density $p(x|y)$. The Markov property is captured by assuming that the joint density, conditional on the initial value $X_0 = x_0$, is given by:

$$f_{(X_1,\ldots,X_n|X_0=x_0)}(X_1, \ldots, X_n) = p(x_0, x_1)p(x_1, x_2)\ldots p(x_{n-1}, x_n)$$

Continuous State Spaces

Given that the process is currently at state x, the probability that it moves to a point in $A \subseteq \mathcal{R}$ is given by:

$$P(x, A) = \int_A p(x, y)dy$$

The nth step ahead transition is computed analogously as that in the Finite State Spaces case:

$$P^{(n)}(x, A) = \int_R P(x, dy)P^{(n-1)}(y, A)$$

An invariant density $\pi(y)$ for the transition kernel $p(x, y)$ is a density that satisfies:

$$\pi(y) = \int_R \pi(x)p(x, y)dx$$

Continuous State Spaces

For process in continuous state spaces, the definitions of irreducibility and aperiodicity are as before with $p(x, y)$ in place of p_{ij}. To define recurrence for continuous state spaces, let $P_x(A)$ denote the probability of event A given that the process started at x. Then, a π-irreducible chain with invariant distribution π is recurrent if for each B with $\pi(B)>0$,

$$P_x(X_n \in B \quad i.o)>0 \quad \text{for all x,}$$

$$P_x(X_n \in B \quad i.o) = 1 \quad \text{for } \pi\text{-almost all x}$$

The chain is *Harris Recurrent* if $P_x(X_n \in B \quad i.o.) = 1$ for all x

Continuous State Spaces

Following theorems use the total *variation distance* between two measures, defined as follows:

The total variation norm of a bounded, signed measure λ is $||\lambda|| = sup_A \lambda(A) - inf_A \lambda(A)$, and the total variation distance between two such measures λ_1 and λ_2 is $||\lambda_1 - \lambda_2||$

Theorem (Theorem 6.6)

Suppose that P is π-irreducible and that π is an invariant distribution for P. Then P is positive recurrent and π is the unique invariant distribution of P. If P is also aperiodic, then for π-almost all x,

$$||P^{(n)}(x, .) - \pi|| \longrightarrow 0,$$

With $||.||$ denoting the total variation distance.

Continuous State Spaces

Theorem (Theorem 6.7)

If $||P^{(n)}(x,.) - \pi|| \longrightarrow 0$ *for all* x*, the chain is* π*-irreducible, aperiodic, positive recurrent, and has invariant distribution* π*.*

Continuous State Spaces

These theorems form the basis of MCMC methods. In practice, the researcher seeks to *construct an irreducible, aperiodic and positive recurrent transition kernel for which the invariant distribution is the target distribution*.

Introduction

The Gibbs sampler algorithm is one of the most used (and useful) Markov chain Monte Carlo (MCMC) methods available to sample from non-standard distributions in Bayesian analysis. It is a special case of the Metropolis-Hastings (MH) algorithm, but originated from a different background.

Problem

The problem we are faced with in MCMC theory is to construct a kernel or transition density $p(x, y)$ for which the invariant distribution π is the target distribution. Remember that π is given by

$$\pi(y) = \int \pi(x)p(x, y)dx$$

where x and y are the "previous" and "current" states respectively. In our case, the random variables of interest are the parameters θ and $\pi(\theta|y)$ is the target distribution.

Gibbs Sampler

The Gibbs algorithm proposes the following transition kernel for two parameter blocks

$$p(x, y) = \pi(y_2|y_1)\pi(y_1|x_2)$$

where $x = (x_1, x_2)$ and $y = (y_1, y_2)$. We can see that in order for the Gibbs sampler to be of use, we must first obtain the conditional distributions of each parameter block in terms of the others.

Gibbs Sampler

Proof that the Gibbs kernel leads to the invariant distribution:

$$
\begin{aligned}
\pi(y) &= \int \pi(x)p(x,y)dx \\
&= \int \pi(x_1,x_2)\pi(y_1|x_2)\pi(y_2|y_1)dx_1dx_2 \\
&= \pi(y_2|y_1)\int \pi(y_1|x_2)\pi(x_1,x_2)dx_1dx_2 \\
&= \pi(y_2|y_1)\int \pi(y_1|x_2)\pi(x_2)dx_2 \\
&= \pi(y_2|y_1)\int \pi(y_1,x_2)dx_2 \\
&= \pi(y_2|y_1)\pi(y_1) = \pi(y_1,y_2) = \pi(y)
\end{aligned}
$$

Gibbs Sampler

A word of caution on the careless use of the Gibbs sampler algorithm:

Caution

Even when the conditional distributions $\pi(y_1|x_2)$ and $\pi(y_2|y_1)$ are well defined and can be simulated from, the joint distribution $\pi(y)$ may not correspond to any proper distribution. This is specially true when using improper priors, so care is to be taken!

Algorithm

For two parameter blocks

1. Choose a starting value $x_2^{(0)}$.
2. At the first iteration, draw

$$x_1^{(1)} \text{ from } \pi(x_1|x_2^{(0)}),$$
$$x_2^{(1)} \text{ from } \pi(x_2|x_1^{(1)}).$$

3. At the gth iteration, draw

$$x_1^{(g)} \text{ from } \pi(x_1|x_2^{(g-1)}),$$
$$x_2^{(g)} \text{ from } \pi(x_2|x_1^{(g)}).$$

Algorithm

For d parameter blocks

1. Choose starting values $x_2^{(0)}, \ldots, x_d^{(0)}$.
2. At the gth iteration, draw

$$
\begin{aligned}
x_1^{(g)} &\text{ from } \pi\left(x_1 | x_2^{(g-1)}, \ldots, x_d^{(g-1)}\right), \\
x_2^{(g)} &\text{ from } \pi\left(x_2 | x_1^{(g)}, x_3^{(g-1)}, \ldots, x_d^{(g-1)}\right), \\
&\vdots \\
x_d^{(g)} &\text{ from } \pi\left(x_d | x_1^{(g)}, \ldots, x_{d-1}^{(g)}\right).
\end{aligned}
$$

Simulation Exercise

Initial setting for the simulation:

- $N = 1000$
- $\beta = (1.5, -3.5, 2)'$
- $x_1 \sim \mathcal{N}_N(0, 2^2)\,, x_2 \sim \mathcal{N}_N(0, 3^2)\,, X = (\mathbf{1}, x_1, x_2)$
- $y = X\beta + \mu\,, \quad \mu \sim \mathcal{N}_N(0, 1)$

Prior distributions:

$$\beta \sim \mathcal{N}_3(\beta_0, B_0)$$
$$\sigma^2 \sim \mathcal{IG}(\alpha_0/2, \delta_0/2)$$

Simulation Exercise

Which results in posterior distributions

$$\beta|\sigma^2, y, X \sim \mathcal{N}_3(\bar{\beta}, B_1)$$
$$\sigma^2|\beta, y, X \sim \mathcal{IG}(\alpha_1/2, \delta_1/2)$$

with

$$B_1 = (\sigma^{-2}X'X + B_0^{-1})^{-1}$$
$$\bar{\beta} = B_1(\sigma^{-2}X'y + B_0^{-1}\beta_0)$$
$$\alpha_1 = \alpha_0 + N$$
$$\delta_1 = \delta_0 + (y - X\beta)'(y - X\beta)$$

Simulation Exercise

The Gibbs algorithm for this simulation is therefore

1. Choose a starting value $\sigma^{2(0)}$.
2. At the gth iteration, draw

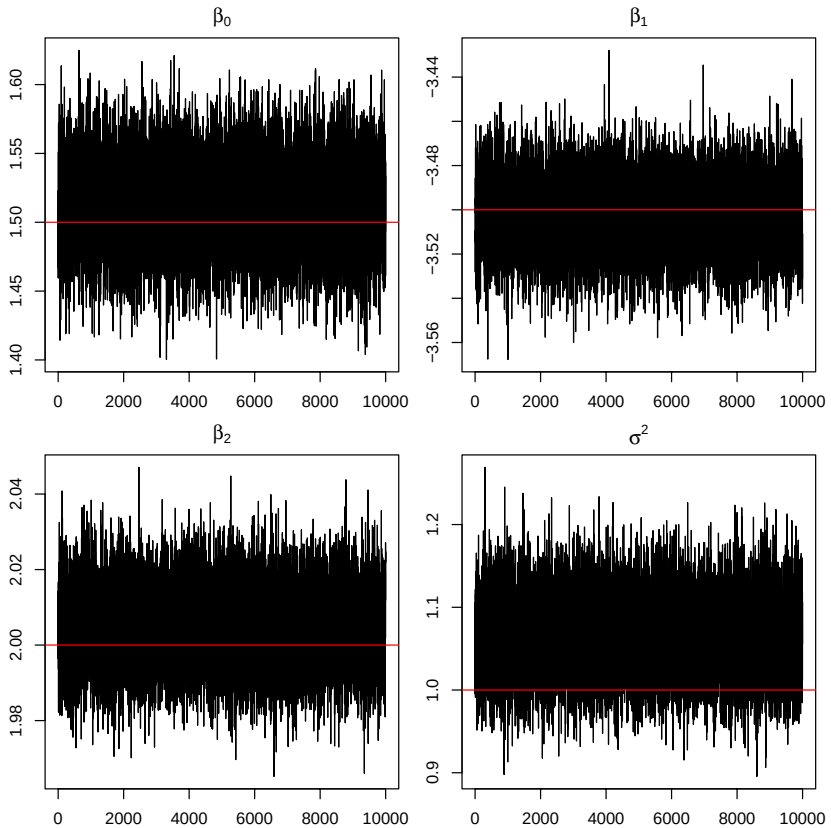
$$\beta(g) \text{ from } \mathcal{N}_3(\bar{\beta}^{(g)}, B_1^{(g)}),$$
$$\sigma^{2(g)} \text{ from } \mathcal{IG}(\alpha_1/2, \delta_1^{(g)}/2).$$

with

$$B_1^{(g)} = (\sigma^{-2(g-1)} X'X + B_0^{-1})^{-1}$$
$$\bar{\beta} = B_1^{(g)}(\sigma^{-2(g-1)} X'y + B_0^{-1}\beta_0)$$
$$\delta_1^{(g)} = \delta_0 + (y - X\beta^{(g)})'(y - X\beta^{(g)})$$

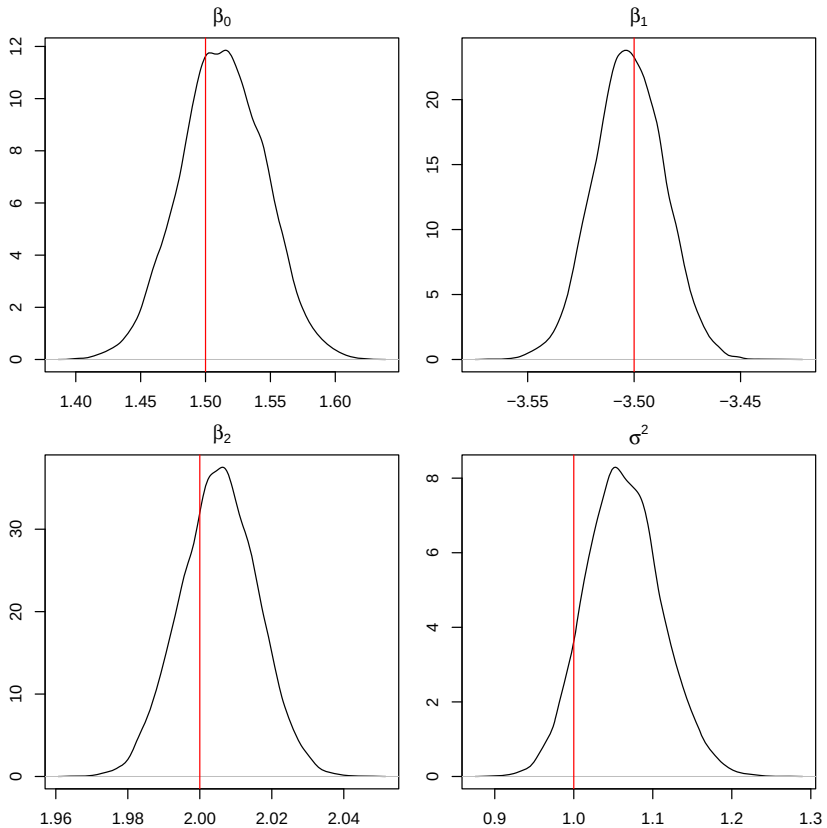
Simulation Exercise

Figure: Trace plots for the parameters in 10,000 draws



Simulation Exercise

Figure: Density plots for the parameters in 10,000 draws



Metropolis-Hastings Algorithm

- The MH algorithm is more general than the Gibbs sampler because it does not require that the full set of conditional distributions be available for sampling.
- To generate a sample from $f(X)$, where X may be a scalar or vector random variable, the first step is to find a kernel $p(X, Y)$ that has $f(\cdot)$ as its invariant distribution. To that end, we introduce the idea of a reversible kernel, defined as a kernel $q(\cdot, \cdot)$ for which:

Metropolis-Hastings Algorithm

$$f(x)q(x,y) = f(y)q(y,x).$$

if q is reversible,

$$\begin{aligned} P(y \in A) &= \int_A \int_{R^d} f(x)q(x,y)dxdy \\ &= \\ &= \int_A \int_{R^d} f(y)q(y,x)dxdy \\ &= \\ &= \int_A f(y)dy. \end{aligned}$$

Metropolis-Hastings Algorithm

- This shows that $f(\cdot)$ is the invariant distribution for a kernel $q(\cdot,\cdot)$ because the probability that y is contained in A is computed from $f(\cdot)$.
- The fact that a reversible kernel has this property can help in finding a kernel that has the desired target distribution. We now follow the derivation of the algorithm. The trick is to make an irreversible kernel reversible.[5]

5

- Greenberg, E. (2008). 'Introduction to Bayesian Econometrics '. *Springer*. pag 96-99.

Metropolis-Hastings Algorithm

If a kernel is not reversible, for some pair (x, y) we have

$$f(x)q(x, y) > f(y)q(y, x).$$

The MH algorithm deals with this situation by multiplying both sides by a function $\alpha(\cdot, \cdot)$ that turns the irreversible kernel $q(\cdot, \cdot)$ into the reversible kernel $p(x, y) = \alpha(x, y)q(x, y)$:

$$f(x)\alpha(x, y)q(x, y) = f(y)\alpha(y, x)q(y, x). \tag{1}$$

Metropolis-Hastings Algorithm

- The expression $\alpha(x, y)q(x, y)$ is interpreted as follows: if the present state of the process is x, generate a value y from the kernel $q(x, y)$ and make the move to y with probability $\alpha(x, y)$. If the move to y is rejected, the process remains at x.
- Note that this transition kernel combines a continuous kernel $q(x, y)$ and a probability mass function $\alpha(x, y)$.

Metropolis-Hastings Algorithm

- How to defined $\alpha(x, y)$ is the next question. Suppose that

$$f(x)q(x, y) > f(y)q(y, x).$$

- Roughly speaking, this means that the kernel goes from x to y with greater probability than it goes from y to x.
- Accordingly, if the process is at y and the kernel proposes a move to x, that move should be made with high probability. This can be done by setting $\alpha(y, x) = 1$. But then, $\alpha(x, y)$ is determined because, from (2),

$$f(x)q(x, y)\alpha(x, y) = f(y)q(y, x)$$

Metropolis-Hastings Algorithm

implies

$$\alpha(x,y) = \left\{ \begin{array}{ll} \min\left\{\frac{f(y)q(y,x)}{f(x)q(x,y)}, 1\right\} & \textbf{if} \quad f(x)q(x,y) \neq 0, \\ \\ 0, & \textbf{otherwise}. \end{array} \right\}$$

The condition that $f(x)q(x,y) \neq 0$ is usually satisfied in practice because the starting value is always chosen in the support of the distribution and the kernel usually generates values in the support of the distribution.

Metropolis-Hastings Algorithm

MH algorithm

1. Given x, generate Y from $q(x, y)$.
2. Generate U from $U(0, 1)$. If

$$U \leqslant \alpha(x, Y) = \min\left\{\frac{f(Y)q(Y, x)}{f(x)q(x, Y)}, 1\right\}$$

return Y. Otherwise, return x and go to 1.

Although we have shown that the MH kernel has the desired target distribution, this is only a necessary condition for convergence to the target.

Metropolis-Hastings Algorithm

Example: MH for $Beta(3,4)$ with $U(0,1)$ proposal

1. Set $x^{(0)}$ equal to a number between zero and one.
2. At the *gth* iteration, generate U_1 and U_2 from $U(0,1)$.
3. If
$$U_1 \leqslant \alpha(x^{(g-1)}, U_2) = \frac{U_2^2(1-U_2)^3}{(x^{(g-1)})^2(1-x^{(g-1)})^3},$$
set $x^{(g)} = U_2$. Otherwise set $x^{(g)} = x^{(g-1)}$.
4. Go to 2 and continue until the desired number of iterations is obtained.

Metropolis-Hastings Algorithm

Theorem (Theorem 7.2, Greenberg)

Suppose P is a π-irreducible Metropolis kernel. Then P is Harris recurrent.

Metropolis-Hastings Algorithm

The next implementation issue is how to choose the proposal density $q(. \quad , .)$. There are several possible choices and the selection is a matter of judgment. Several factors need to be taken into account:

1. The kernel should generate proposals that have a reasonably good probability of acceptance; if not, the same value will be returned often, and the algorithm will mix poorly
2. There may be a high acceptance rate if the kernel generates only proposals that are close to the current point, but the sampling may then be confined to a small part of the support, again leading to poor mixing.

Metropolis-Hastings Algorithm

Two straightforward (not necessarily good) kernels are the random-walk kernel and the independence kernel. For the former, the proposal y is generated from the current value x by the addition of a random variable or vector u, $y = x + u$, where the distribution of u is specified. If that distribution is symmetric around zero, $(h(u) = h(-u))$, the kernel has the property that $q(x, y) = q(y, x)$,which implies that $\alpha(x, y) = f(y)/f(x)$. Accordingly, with a random-walk kernel, a move from x to y is made for certain if $f(y) > f(x)$. A move from a higher density point to a lower density point is not ruled out, but the probability of such a move $f(y)/f(x)$ is less than one.

Metropolis-Hastings Algorithm

The independence kernel has the property $q(x, y) = q(y)$; that is, the proposal density is independent of the current state of the chain. For this type of kernel:

$$\alpha(x, y) = \frac{f(y)/q(y)}{f(x)/q(x)},$$

and our comments about the probability of a move are similar to those about the random-walk chain if f(.) is replaced by f(.)/q(.)

Metropolis-Hastings Algorithm

A "tailored" kernel is recommended: construct a kernel that approximates the target distribution. This may be done by choosing a fat-tailed distribution, such as the multivariate t with small ν, whose mean and scale matrix are chosen to coincide with the mode and negative inverse of the second-derivative matrix at the mode, respectively. An example of a tailored kernel may be found in section 9.2 (Greenberg). If there is just one parameter block, the tailored kernel is an independence kernel. If there is more than one block, the tailored kernel for the block being updated may depend on the current values of parameters in the other blocks.

Metropolis-Hastings Algorithm

MH algorithm with two blocks

1. Let the state be (x_1, x_2) at the gth iteration and (y_1, y_2) at the $g+1$st iteration. Draw Z_1 from $q_1(x_1, Z_1|x_2)$ and U_1 from $U(0,1)$
2. If
$$U_1 \leq \alpha(x_1, Z_1|x_2) = \frac{f(Z_1, x_2)q_1(Z_1, x_1|x_2)}{f(x_1, x_2)q_1(x_1, Z_1|x_2)},$$
return $y_1 = Z_1$. Otherwise return $y_1 = x_1$
3. Draw Z_2 from $q_2(x_2, Z_2|y_1)$ and U_2 from U(0,1).
4. If
$$U_2 \leq \alpha(x_2, Z_2|y_1) = \frac{f(y_1, Z_2)q_2(Z_2, x_2|y_1)}{f(y_1, x_2)q_2(x_2, Z_2|y_1)},$$
return $y_2 = Z_2$. Otherwise return $y_2 = x_2$.

Metropolis-Hastings Algorithm

MH algorithm with two blocks

In this algorithm, the kernel $q_1(x_1, Y_1|x_2)$ is analogous to $q(x, Y)$; it generates a value Y_1 conditional on the current value x_1 in the same block and the current value x_2 in the other block. If "tailored" proposal densities are used, new densities are specified for $q_1(x_1, Z_1|x_2)$ and $q_2(x_2, Z_2|y_1)$ for each value of x_2 and y_1, respectively. This algorithm can be extended to an arbitrary number of blocks.

Metropolis-Hastings Algorithm

Having introduced blocks of parameters, we can show that the Gibbs sampler is a special case of the Metropolis-Hastings Algorithm. Consider $\alpha(.,.)$ when the kernel for moving from the current state or value x_1 to the proposal value Y_1 is the conditional distribution $f(x_1|x_2)$, which is assumed to be available for sampling. Then

$$\alpha(x_1, Y_1|x_2) = \frac{f(Y_1, x_2)f(x_1|x_2)}{f(x_1, x_2)f(Y_1|x_2)},$$

but, since $f(Y_1|x_2) = f(Y_1, x_2)/f(x_2)$ and $f(x_1|x_2) = f(x_1, x_2)/f(x_2)$ it follows that $\alpha(x_1, Y_1|x_2) = 1$, showing that the Gibbs algorithm is an MH algorithm where the proposal is always accepted.

Metropolis-Hastings Algorithm

When implementing the MH algorithm for two blocks of parameters, Gibbs sampling may be employed in any block for which the conditional distributions are available for sampling. In the remaining blocks, the MH algorithm may be employed in the usual way, that is, by finding suitable proposal densities and accepting with probability $\alpha(x, y)$. At each iteration, the algorithm works through the blocks, either moving to a new value or retaining the current value of the variables in the block.

Convergence Diagnostics

To check whether the chain has converged to its posterior distribution, we use the following methods:

- Visual inspection.
- Gelman and Rubin Diagnostic.
- Geweke Diagnostic.

Convergence Diagnostics
Visual Inspection

- One way to see if our chain has converged is to see how well our chain is **mixing**, or moving around the parameter space.
- If our chain takes a long time to move around the parameter space, then it will take longer to converge.
- We can see how well our chain is mixing through visual inspection.
- We need to do inspection for every parameter.

Convergence Diagnostics
Visual Inspection (Traceplots)

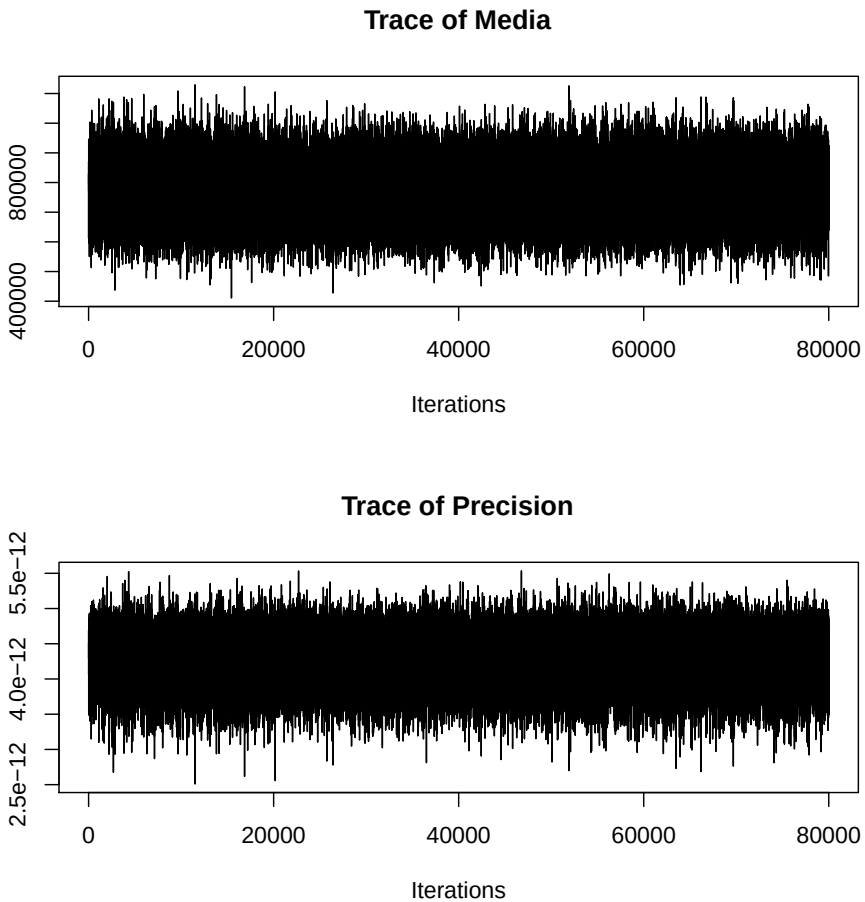


Figure: Traceplots for mean and precision.

Convergence Diagnostics
Visual Inspection (Autocorrelation plots)

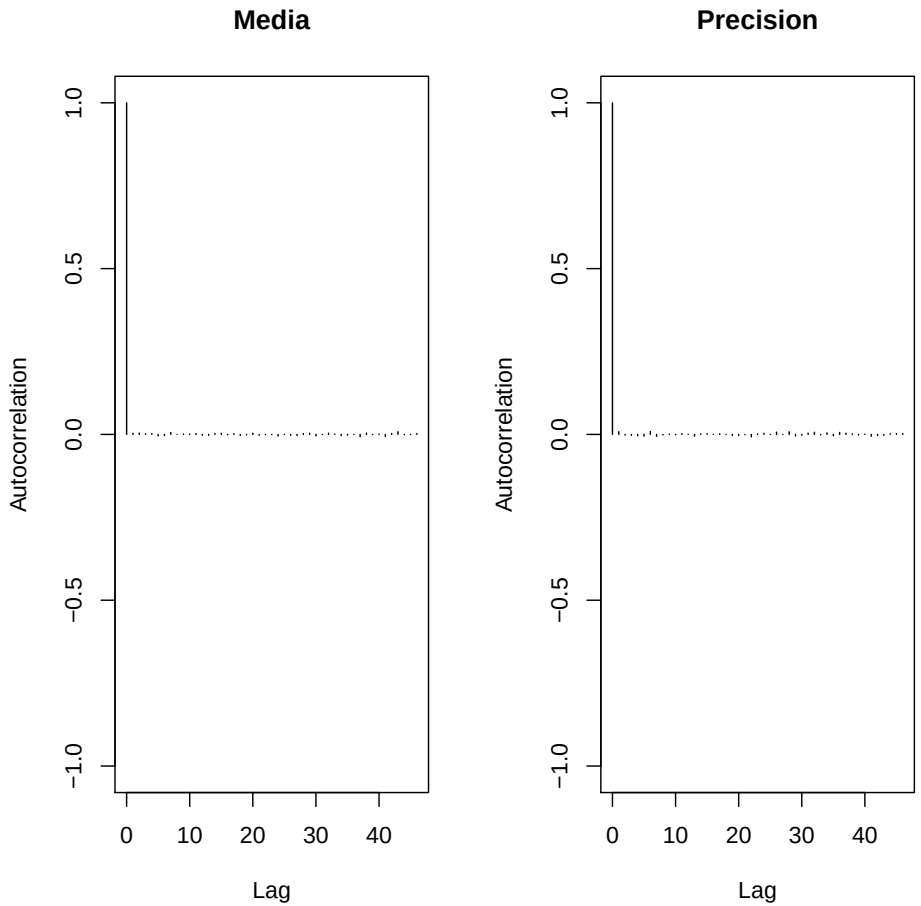


Figure: Autocorrelation plots for mean and precision.

Convergence Diagnostics
Gelman and Rubin Diagnostic

- Gelman (especially) argues that the best way to identify non-convergence is to simulate multiple sequences for over-dispersed starting points.
- The intuition is that the behavior of all of the chains should be basically the same.
- Or, as Gelman and Rubin put it, the variance within the chains should be the same as the variance across the chains.

Convergence Diagnostics
Gelman and Rubin Diagnostic

- Run $m \geqslant 2$ chains of length $2n$ from overdispersed starting values.
- Discard the first n draws in each chain.
- Calculate the within-chain and between-chain variance.
- Calculate the estimated variance of the parameter as a weighted sum of the within-chain and between-chain variance.
- Calculate the potential scale reduction factor.

Convergence Diagnostics

Gelman and Rubin Diagnostic (Within Chain Variance)

$$W = \frac{1}{m}\sum_{j=1}^{m} s_j^2,$$

where

$$s_j^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\theta_{ij} - \bar{\theta}_j)^2,$$

s_j^2 is just the formula for the variance of the *jth* chain. W is then just the mean of the variances of each chain.

Convergence Diagnostics
Gelman and Rubin Diagnostic (Between Chain Variance)

$$B = \frac{n}{m-1}\sum_{j=1}^{m}(\bar{\theta}_j - \bar{\bar{\theta}})^2,$$

where

$$\bar{\bar{\theta}} = \frac{1}{m}\sum_{j=1}^{m}\bar{\theta}_j$$

This is the variance of the chain means multiplied by n because each chain is based on n draws.

Convergence Diagnostics
Gelman and Rubin Diagnostic (Estimated Variance)

We can then estimate the variance of the stationary distribution as a weighted average of W and B.

$$\hat{var}(\theta) = \left(1 - \frac{1}{n}\right) W + \frac{1}{n} B$$

Because of overdispersion of the starting values, this overestimates the true variance, but is unbiased if the starting distribution equals the stationary distribution (if starting values were not overdispersed).

Convergence Diagnostics

Gelman and Rubin Diagnostic (Potential Scale Reduction Factor)

The potencial scale reduction factor is

$$\hat{R} = \sqrt{\frac{\hat{var}(\theta)}{W}}$$

When $\hat{R}$ is high (perhaps greater than 1.1 or 1.2), then we should run our chains out longer to improve convergence to the stationary distribution.

Convergence Diagnostics
Gelman and Rubin Diagnostic (Potential Scale Reduction Factor)

- If we have more than one parameter, then we need to calculate the potential scale reduction factor for each parameter.
- We should run our chains out long enough so that all the potential scale reduction factors are small enough.
- We can then combine the mn total draws from our chains to produce one chain from the stationary distribution.

Convergence Diagnostics
Gelman and Rubin Diagnostic (Potential Scale Reduction Factor)

```
Potential scale reduction factors:

          Point est. 97.5% quantile
Media              1                1
Precision          1                1

Multivariate psrf

1
```

Convergence Diagnostics
Gelman and Rubin Diagnostic (Potential Scale Reduction Factor)

We can see how the **psrf** evolves through the iterations using the **gelman.plot()** function.

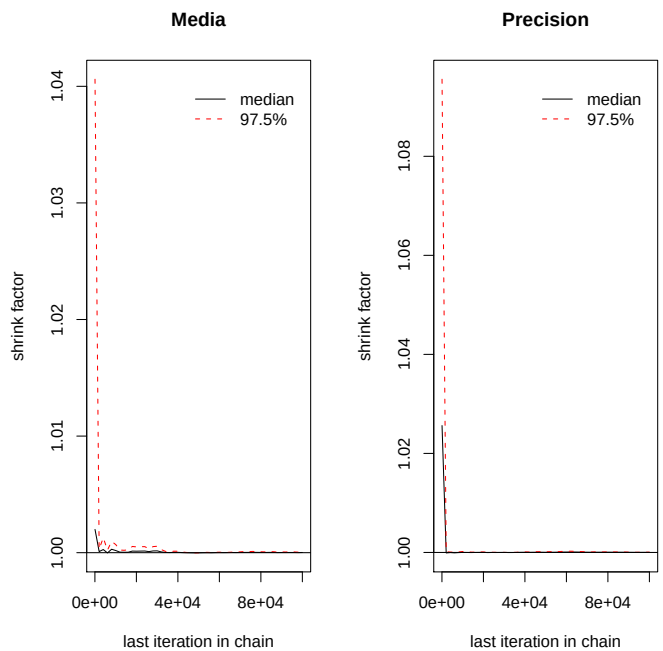


Figure: Gelman Plot

Example of the Gibbs Sampler

Geweke Diagnostic

The Geweke test takes two parts of the chain (usually the first 10 percent and last 50 percent) and compares the mean of both parts, using the differences of means test in order to see if the two parts of Markov Chain are from the same distribution (null hypothesis). The test statistic is a standard Z-score with the standard errors adjusted for autocorrelation.

```
Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

Media  Precision
-1.315e+00  1.011e-07
```