Chapter 4

Non-conjugate multi-parameter problems

Chapter 4. Non-conjugate multi-parameter problems

- Here we will study some multi-parameter problems in which the prior distribution does not have to be conjugate
- Inferences are made by using techniques which simulate realisations from the posterior distribution
- These methods are generally referred to as Markov Chain Monte Carlo techniques, and often abbreviated to MCMC
- Two main MCMC techniques:
  1. the Gibbs sampler – at the forefront of the recent MCMC revolution
  2. Metropolis-Hastings sampling
- MCMC schemes based on the combination of these two fundamental techniques are still at the forefront of MCMC research

4.1 Why is inference not straightforward in non-conjugate problems?

Example 4.1

- Consider again the problem in section 2.2

  $X_i|\mu,\tau \sim N(\mu, 1/\tau)$, $i = 1, 2, \ldots, n$ (independent)
- Here we showed that a NGa prior for $(\mu,\tau)^T$ was conjugate
- But what if a $NGa(b, c, g, h)$ prior distribution does not adequately represent our prior beliefs?
- Suppose instead that our prior beliefs are represented by independent priors for the parameters

$$\mu \sim N\left(b, \frac{1}{c}\right) \qquad \text{and} \qquad \tau \sim Ga(g, h)$$

What is the posterior distribution for $(\mu,\tau)^T$?

Solution

. . .

In this case . . .

- What is the posterior mean of μ and of τ?
- What are their marginal distributions?
- How can we calculate the moments $E(\mu^{m_1}\tau^{m_2}|\boldsymbol{x})$ of this posterior distribution?

In this case . . .

- What is the posterior mean of μ and of τ?
- What are their marginal distributions?
- How can we calculate the moments $E(\mu^{m_1}\tau^{m_2}|\boldsymbol{x})$ of this posterior distribution?

Marginal posterior density for μ

$$\begin{aligned}\pi(\mu|\boldsymbol{x}) &= \int_0^\infty \pi(\mu,\tau|\boldsymbol{x})\,d\tau \\ &= \frac{\displaystyle\int_0^\infty \tau^{g+\frac{n}{2}-1}\exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2+(\bar{x}-\mu)^2\right]\right\}d\tau}{\displaystyle\int_{-\infty}^\infty\int_0^\infty \tau^{g+\frac{n}{2}-1}\exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2+(\bar{x}-\mu)^2\right]\right\}d\tau\,d\mu}\end{aligned}$$

Marginal posterior density for τ

$$\pi(\tau|\boldsymbol{x}) = \int_{-\infty}^{\infty} \pi(\mu, \tau|\boldsymbol{x})\, d\mu$$

$$= \frac{\displaystyle\int_{-\infty}^{\infty} \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\} d\mu}{\displaystyle\int_{-\infty}^{\infty}\int_{0}^{\infty} \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\} d\tau\, d\mu}$$

Marginal posterior density for τ

$$\pi(\tau|\boldsymbol{x}) = \int_{-\infty}^{\infty} \pi(\mu, \tau|\boldsymbol{x})\, d\mu$$

$$= \frac{\displaystyle\int_{-\infty}^{\infty} \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\} d\mu}{\displaystyle\int_{-\infty}^{\infty}\int_{0}^{\infty} \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\} d\tau\, d\mu}$$

General moments

$$E(\mu^{m_1}\tau^{m_2}|\boldsymbol{x})$$

$$= \int_{-\infty}^{\infty}\int_{0}^{\infty} \mu^{m_1}\tau^{m_2}\pi(\mu, \tau|\boldsymbol{x})\, d\tau\, d\mu$$

$$= \frac{\displaystyle\int_{-\infty}^{\infty}\int_{0}^{\infty} \mu^{m_1}\tau^{m_2} \times \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\} d\tau\, d\mu}{\displaystyle\int_{-\infty}^{\infty}\int_{0}^{\infty} \tau^{g+\frac{n}{2}-1} \exp\left\{-\frac{c}{2}(\mu-b)^2 - h\tau - \frac{n\tau}{2}\left[s^2 + (\bar{x}-\mu)^2\right]\right\} d\tau\, d\mu}$$

Comments

- These integrals cannot be determined analytically
- It is possible to use numerical integration methods or approximations (for large n)
- However, in general, the accuracy of the numerical approximation to the integral deteriorates as the dimension of the integral increases
- How do we analyse models with a large number of parameters?

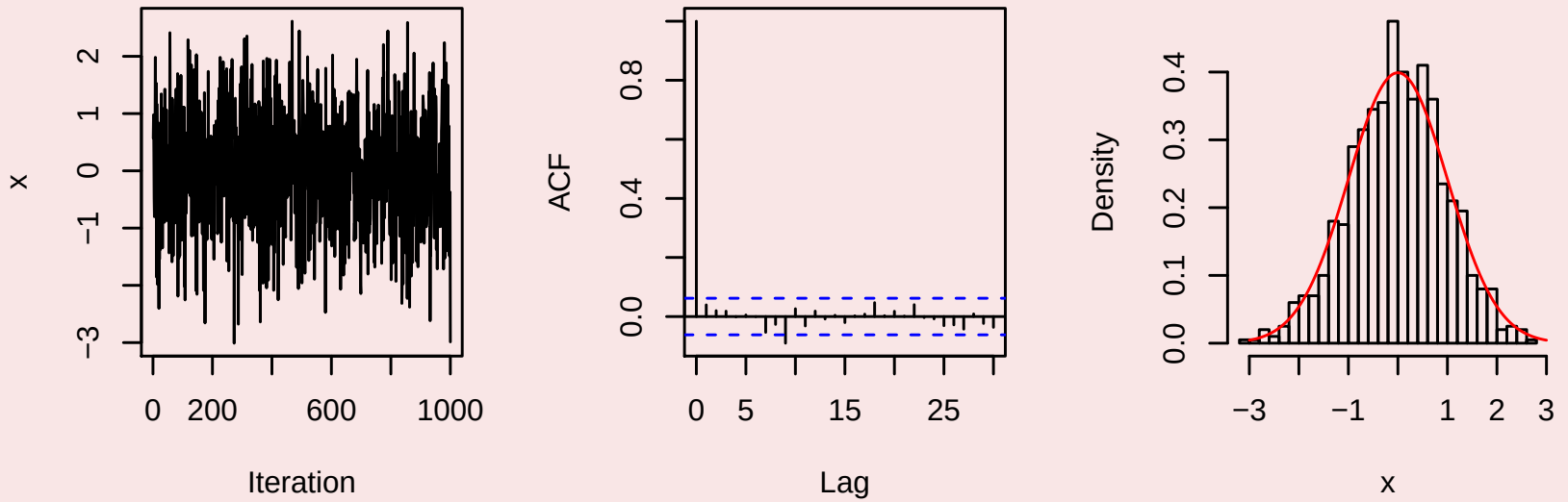
Comment

- Not using a conjugate prior distribution can cause many basic problems such as difficulty in plotting the marginal posterior density or determining posterior moments
- But having to use conjugate priors is far too restrictive for many real data analyses:
  (i) our prior beliefs may not be captured using a conjugate prior
  (ii) most models for complex data do not have conjugate priors
- Until relatively recently, practical Bayesian inference for real complex problems was either not feasible or only undertaken by the dedicated few prepared to develop bespoke computer code to numerically evaluate all the integrals etc.

4.2 Simulation-based inference

- Can get around the problem of having to work out integrals
- Base inferences on simulated realisations from the posterior distribution
- This is the fundamental idea behind MCMC methods
- If we could simulate from the posterior distribution then we could use a very large sample of realisations to determine posterior means, standard deviations, correlations, joint densities, marginal densities etc.

- Imagine you wanted to know about the standard normal distribution – its shape, its mean, its standard deviation
- But didn't know any mathematics so that you couldn't derive say the distribution's zero mean and unit variance
- However you've been given a "black box" which can simulate realisations from this distribution
- Here we'll use the `R` function `rnorm()` as the black box simulator
- If you decide to generate 1K realisations the output might look something like



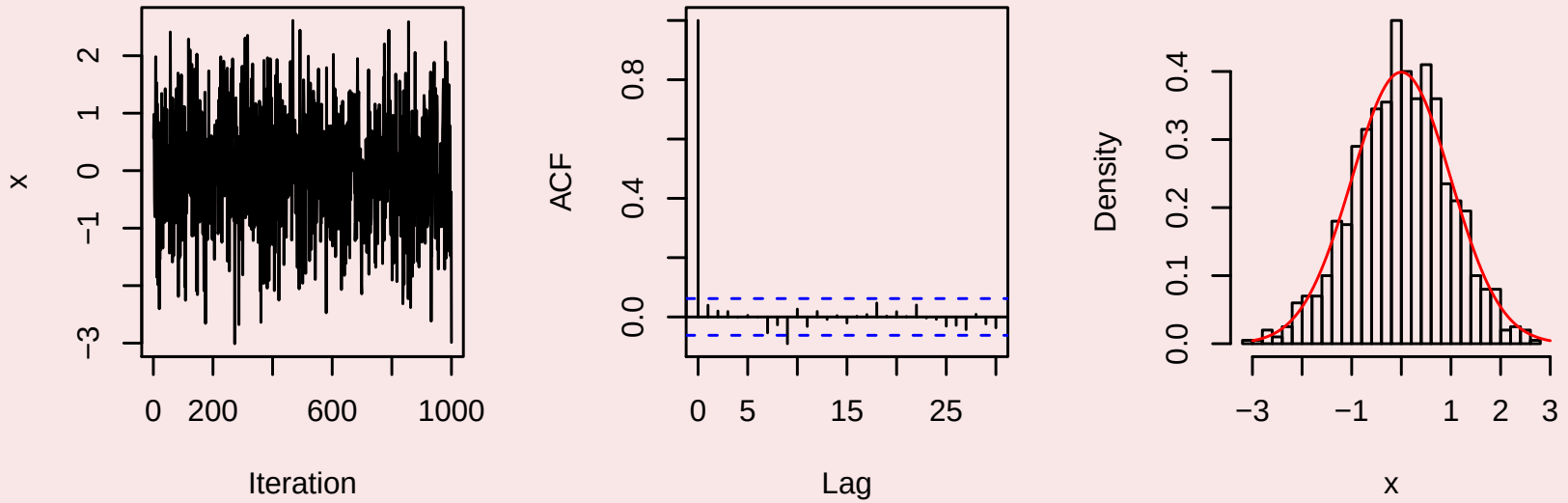
In these plots

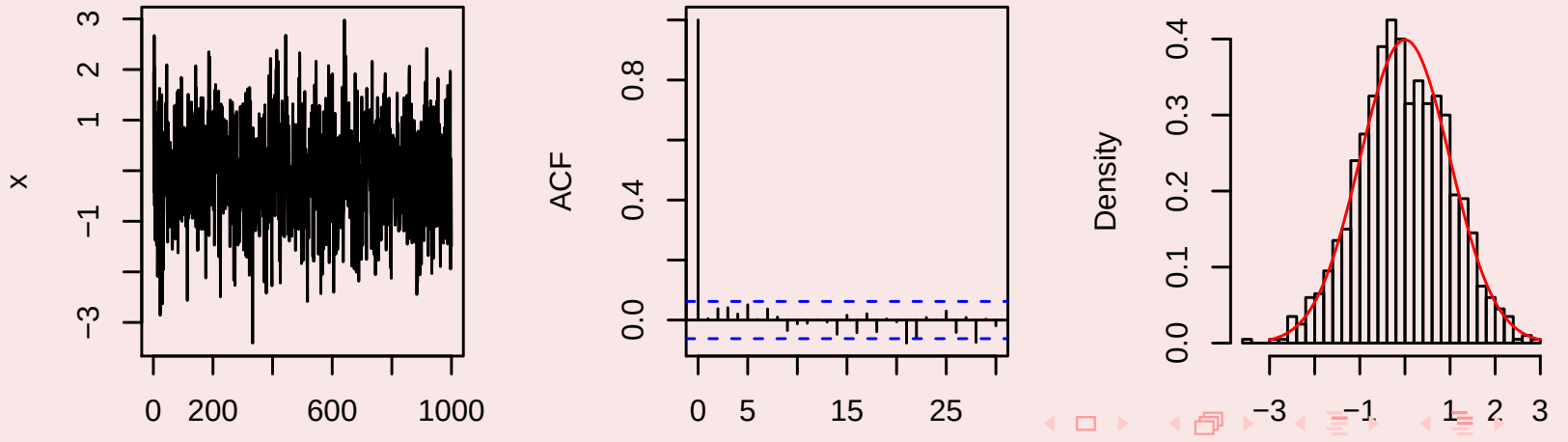
Left: the trace plot of the output – the realisations from the black box sampler in the order they are produced

Mid: the autocorrelation (ACF) plot – shows the correlation between $\{x_1, \ldots, x_{N-k}\}$ and $\{x_{k+1}, \ldots, x_N\}$ for $k \geq 0$. Call k as the lag. It shows how correlated the realisations are at different lags. The lag 0 autocorrelation $corr(x_i, x_i)$ must be one (by definition). Here the (sample) correlation between say consecutive values $corr(x_i, x_{i+1})$ will be almost zero, which shows that the simulator `rnorm()` produces independent realisations. This is also the case for correlations at all positive lags. This sample ACF plot tells that the simulator generates independent sample.

Right: the density histogram of the realisations. This tells that what the standard normal distribution is like.

- If you simulate another 1K realisations then the output you get will be a different set of realisations but from the same distribution
- Both sets of realisations might look like





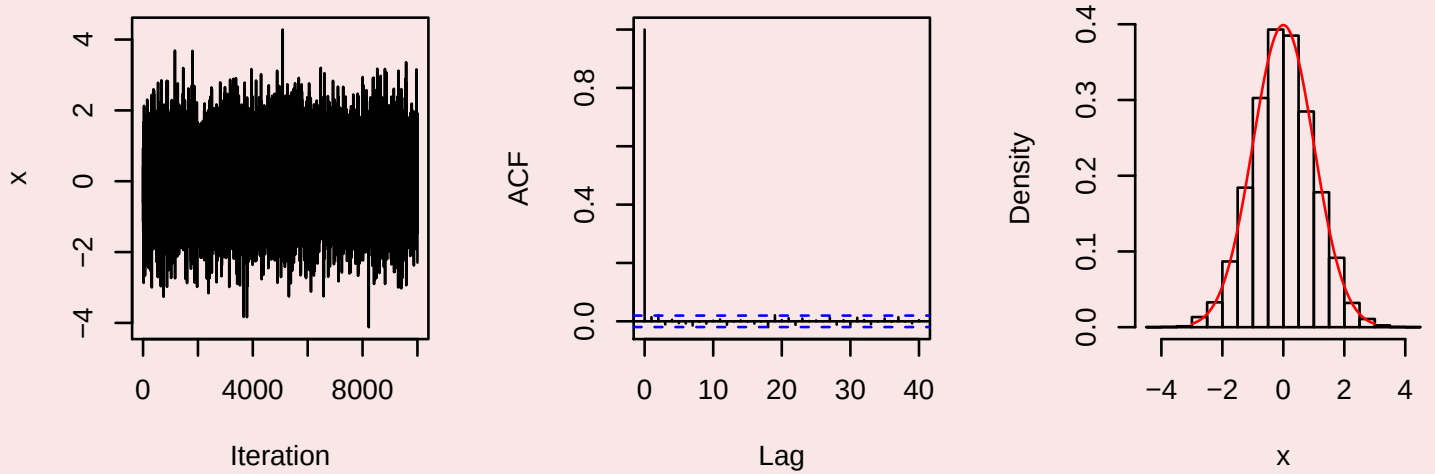
- We can also estimate various quantities of the standard normal distribution from these simulated datasets of 1K realisations

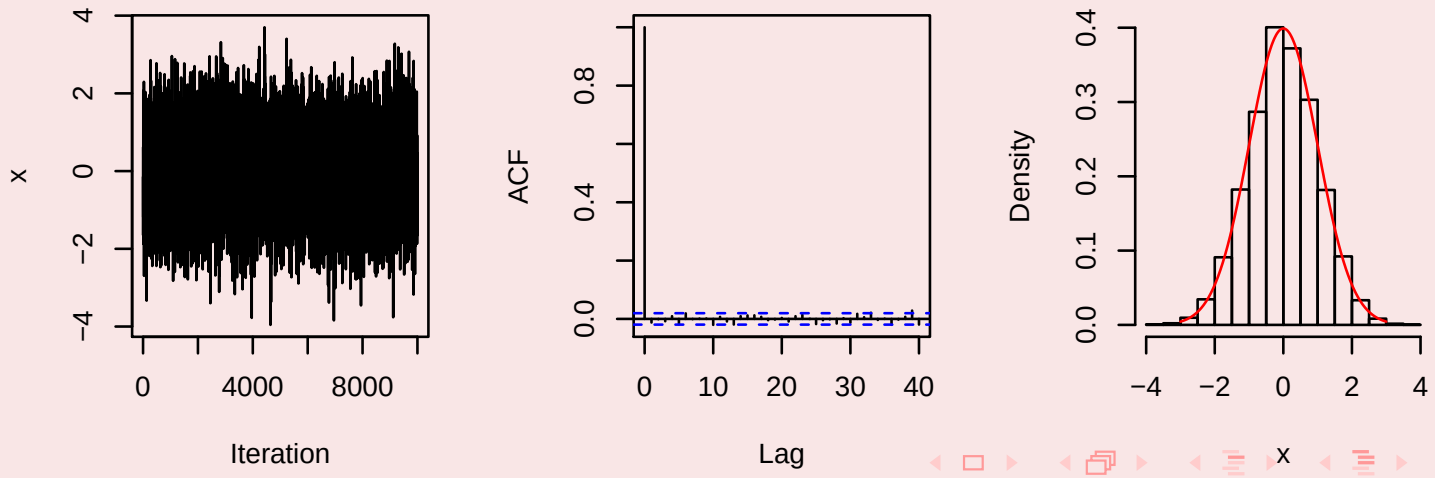
	1st Qu.	Median	Mean	3rd Qu.	St.Dev.
Dataset 1	-0.65240	-0.00130	-0.00192	0.64810	0.96049
Dataset 2	-0.69880	-0.09637	-0.03274	0.67330	0.99599

- The numerical summaries are slightly different but essentially tell the same story
- In each Dataset, the mean and median are around zero and the standard deviation is around one.
- We know from previous modules that there is sample variability in estimates of means from random samples

- Now look at two datasets with 10K simulated realisations:

	1st Qu.	Median	Mean	3rd Qu.	St.Dev.
Dataset 3	-0.66820	-0.00048	0.00370	0.68070	0.99593
Dataset 4	-0.67130	0.01354	0.01008	0.67920	1.00691





- Now estimates have much less sampling variability due to the larger sample size
- In fact we can estimate any "population" quantity to any required accuracy simply by simulating a large enough collection of realisations

- These analyses show how we can make inferences, calculate means, variances, densities etc by using realisations from a distribution. In the rest of this chapter, we will look into how we can construct algorithms for simulating from (complex) posterior distributions, from which we can then make inferences

4.3 Motivation for MCMC methods

- We consider a generic case where we want to simulate realisations of two random variables X and Y with joint density $f(x, y)$
- This joint density can be factorised as

$$f(x, y) = f(x)\, f(y|x)$$

  and so we can simulate from $f(x, y)$ by first simulating $X = x$ from $f(x)$, and then simulating $Y = y$ from $f(y|x)$
- On the other hand, we can write

$$f(x, y) = f(y)\, f(x|y)$$

  and so simulate $Y = y$ from $f(y)$ and then $X = x$ from $f(x|y)$
- Assume that simulating from $f(y|x)$ and $f(x|y)$ is straightforward

- The key problem: in general, we can't simulate from the marginal distribution, $f(x)$ and $f(y)$
- Suppose we have a single simulated sample point from the marginal distribution for X, that is, we have an $X = x$ from $f(x)$. We can now simulate a $Y = y$ from $f(y|x)$ to give a pair (x, y) from the bivariate density $f(x, y)$.
- Given that this pair is from the bivariate density, the y value must be from the marginal $f(y)$, and so we can simulate an $X = x'$ from $f(x|y)$ to give a new pair (x', y) also from the joint density $f(x, y)$.
- But now x' is from the marginal $f(x)$, and so we can simulate a $Y = y'$ from $f(y|X = x')$ to give a new pair (x', y') also from the joint density $f(x, y)$.
- And we can keep going.

- The key problem: in general, we can't simulate from the marginal distribution, $f(x)$ and $f(y)$
- Suppose we have a single simulated sample point from the marginal distribution for X, that is, we have an $X = x$ from $f(x)$. We can now simulate a $Y = y$ from $f(y|x)$ to give a pair (x, y) from the bivariate density $f(x, y)$.
- Given that this pair is from the bivariate density, the y value must be from the marginal $f(y)$, and so we can simulate an $X = x'$ from $f(x|y)$ to give a new pair (x', y) also from the joint density $f(x, y)$.
- But now x' is from the marginal $f(x)$, and so we can simulate a $Y = y'$ from $f(y|X = x')$ to give a new pair (x', y') also from the joint density $f(x, y)$.
- And we can keep going.

- This alternate sampling from conditional distributions defines a bivariate Markov chain, and the above is an intuitive explanation for why $f(x, y)$ is its stationary distribution

4.4 The Gibbs sampler

- Suppose we want to generate realisations from the posterior density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_p)^T$, and that we can simulate from the full conditional distributions (FCDs)

$$\pi(\theta_i|\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_p, \boldsymbol{x}) = \pi(\theta_i|\cdot), \qquad i = 1, 2, \ldots, p$$

- The Gibbs sampler follows the following algorithm:
  1. Initialise the iteration counter to $j = 1$.
     Initialise the state of the chain to $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})^T$.
  2. Obtain a new value $\boldsymbol{\theta}^{(j)}$ from $\boldsymbol{\theta}^{(j-1)}$ by successive generation of values

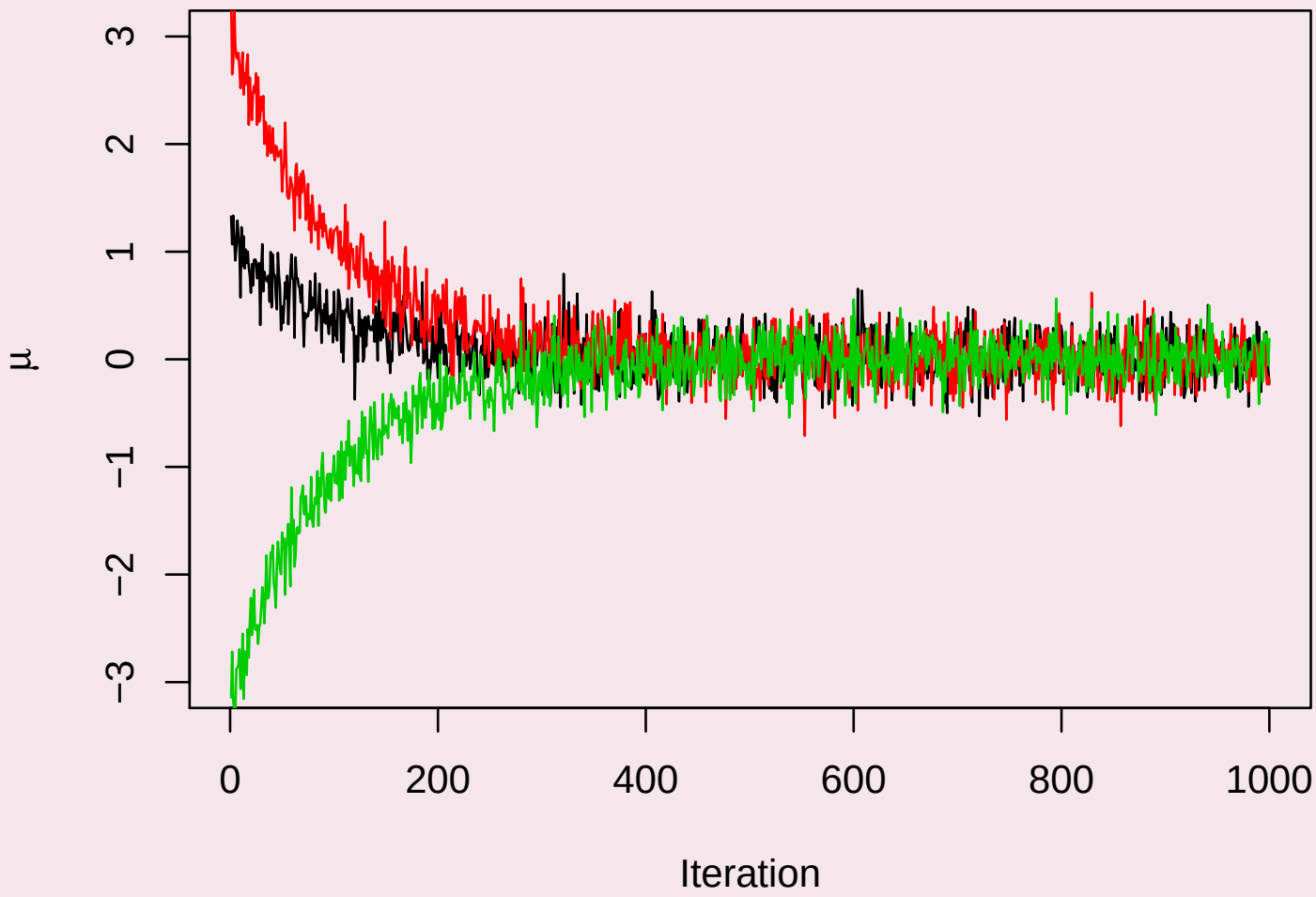
$$\begin{aligned}
\theta_1^{(j)} &\sim \pi(\theta_1|\theta_2^{(j-1)}, \theta_3^{(j-1)}, \ldots, \theta_p^{(j-1)}, \boldsymbol{x}) \\
\theta_2^{(j)} &\sim \pi(\theta_2|\theta_1^{(j)}, \theta_3^{(j-1)}, \ldots, \theta_p^{(j-1)}, \boldsymbol{x}) \\
&\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
\theta_p^{(j)} &\sim \pi(\theta_p|\theta_1^{(j)}, \theta_2^{(j)}, \ldots, \theta_{p-1}^{(j)}, \boldsymbol{x})
\end{aligned}$$

  3. Change counter j to $j + 1$, and return to step 2.

Comments

- This algorithm defines a homogeneous Markov chain as each simulated value depends only on the previous simulated value and not on any other previous values or the iteration counter j
- It can be shown that $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ is the stationary distribution of this chain and so if we simulate realisations by using a Gibbs sampler, eventually the Markov chain will converge to the required posterior distribution

Illustration of burn-in



- Different starting points
- But distributions of the realisations after iteration 500 are the "same"

Illustration of burn-in in two dimension

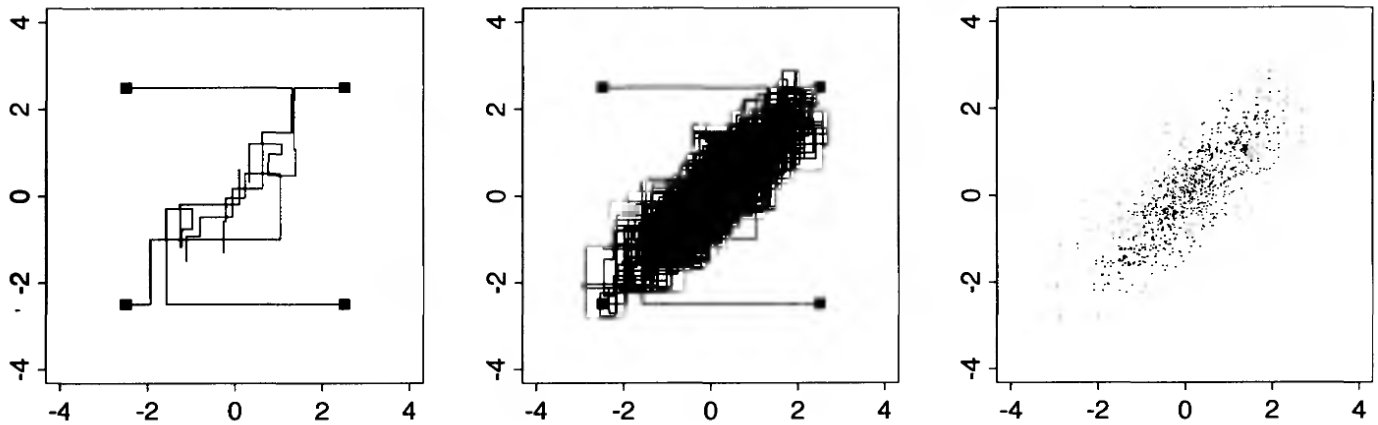


Figure 11.3 *Four independent sequences of the Gibbs sampler for a bivariate normal distribution with correlation $\rho = 0.8$, with overdispersed starting points indicated by solid squares. (a) First 10 iterations, showing the component-by-component updating of the Gibbs iterations. (b) After 500 iterations, the sequences have reached approximate convergence. Figure (c) shows the iterates from the second halves of the sequences.*

- Different starting points
- Chains converge to the "same" distribution

More comments

- This is an iterative scheme
- It needs a period to get to convergence: the *burn-in* period
- As it is a Markov chain, successive iterates are not independent
- We can get accurate values for $E(\boldsymbol{\theta}|\boldsymbol{x})$, $SD(\boldsymbol{\theta}|\boldsymbol{x})$ or even $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ by running the sampler for a long time

4.4.1 Processing output from a Gibbs sampler

- Consider a $p = 2$ parameter problem, with $\boldsymbol{\theta} = (\mu, \tau)^T$
- Run Gibbs sampler for N iterations after burn-in, giving

$$\{(\mu^{(1)}, \tau^{(1)}), (\mu^{(2)}, \tau^{(2)}), \ldots, (\mu^{(N)}, \tau^{(N)})\}$$

- Can calculate features of the posterior distribution using their sample equivalents: $\bar{\mu}$, $\bar{\tau}$, s_μ^2, s_τ^2 and $r_{\mu\tau}$

4.4.1 Processing output from a Gibbs sampler

- Consider a $p = 2$ parameter problem, with $\boldsymbol{\theta} = (\mu, \tau)^T$
- Run Gibbs sampler for N iterations after burn-in, giving

$$\{(\mu^{(1)}, \tau^{(1)}), (\mu^{(2)}, \tau^{(2)}), \ldots, (\mu^{(N)}, \tau^{(N)})\}$$

- Can calculate features of the posterior distribution using their sample equivalents: $\bar{\mu}$, $\bar{\tau}$, s_μ^2, s_τ^2 and $r_{\mu\tau}$

How accurate are these "estimates"?

- Difficult to determine exactly as they are not a *random sample*
- If it was a random sample then (for large N) approximate 95% CI for μ is

$$\bar{\mu} \pm \frac{1.96 s_\mu}{\sqrt{N}}$$

- But we don't have a random sample! What to do?

How to deal with autocorrelation

- But we don't have a random sample! What to do?
- Method one: Assume the output follows some Markov chain model, and use CI of this model to approximate the true CI.
- E.g. the autoregressive model with order 1, AR(1), is

$$\mu^{(t)} - \mu = \theta(\mu^{(t-1)} - \mu) + \varepsilon_t, \text{ where } \varepsilon_t \sim N(0,1).$$

The approximate 95% CI of μ depends on the sample autocorrelation $r(1) = Corr(\mu^{(t)}, \mu^{(t+1)})$, and is

$$\bar{\mu} \pm \frac{1.96 s_\mu}{\sqrt{N\{1 - r(1)\}^2}}.$$

Effective sample size: $N_{eff} = N\{1 - r(1)\}^2$.

- In general, MCMC output with positive autocorrelations has $N_{eff} < N$.

How to deal with autocorrelation

- In practise, the autocorrelation structure of the output is far too complex to determine a simple formula for the accuracy of $\bar{\mu}$
- Method two (more popular): `thin` the output – don't take every iterate
- Sample autocorrelation function with lag k for μ is

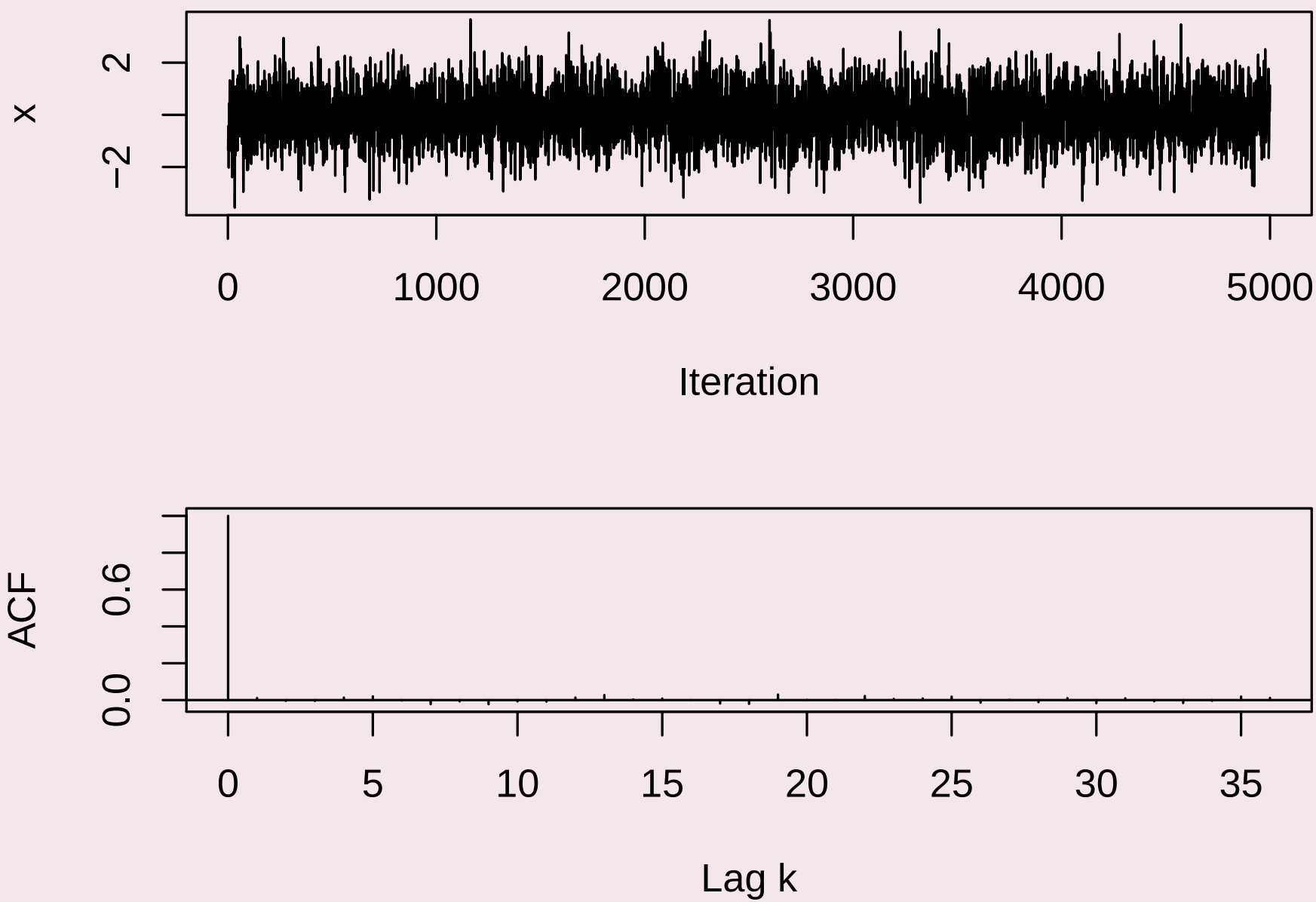
$$r(k) = Corr(\mu^{(j)}, \mu^{(j+k)}).$$

- For example, if $r(k) \simeq 0$, $k \geq 2$ then taking every 2nd iterate would give a posterior sample

$$\{(\mu^{(1)}, \tau^{(1)}), (\mu^{(3)}, \tau^{(3)}), \ldots, (\mu^{(2j+1)}, \tau^{(2j+1)}, \ldots)\}$$

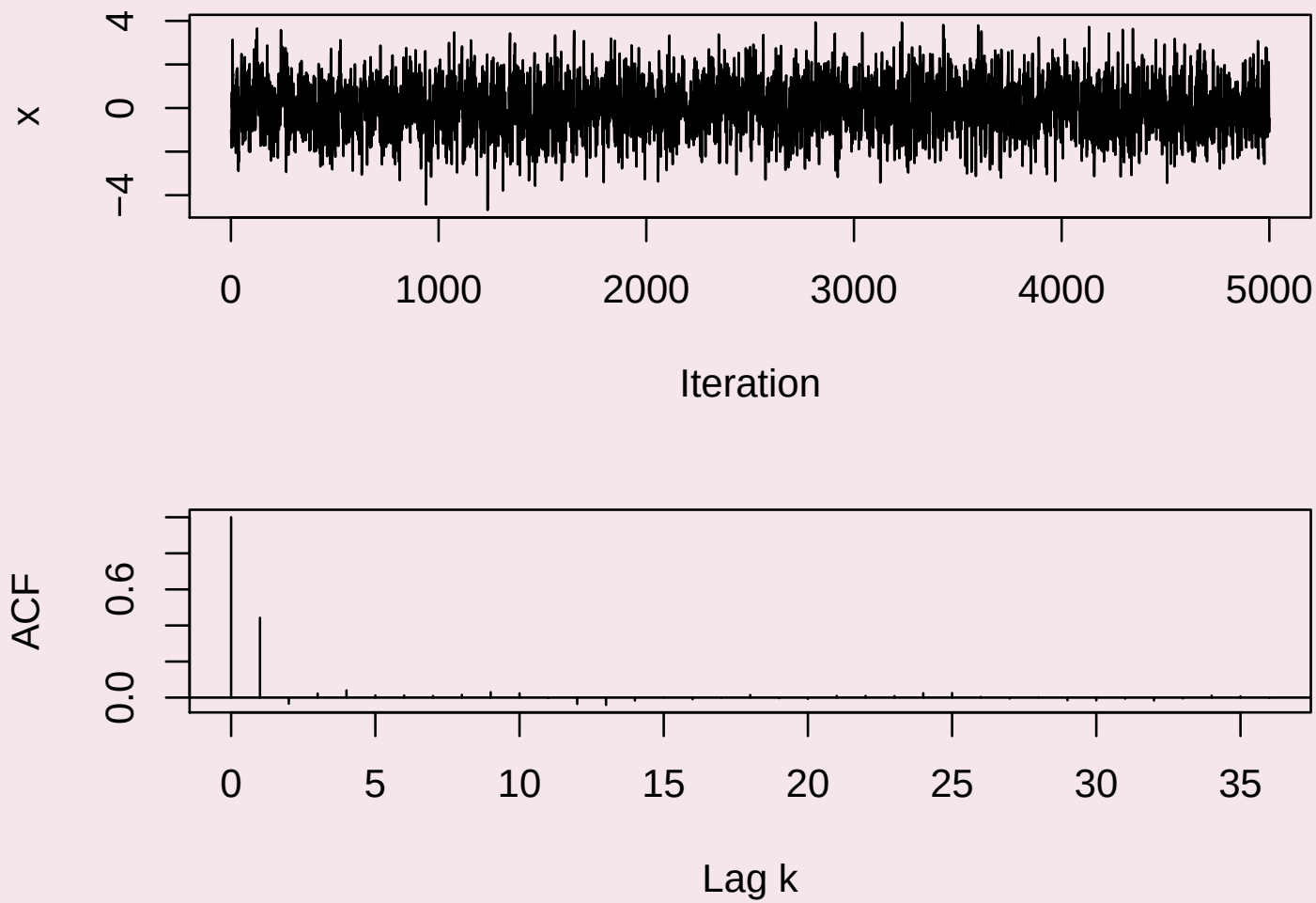
  with autocorrelation function $r^*(k) = 0$, $k \geq 1$, that is, this output has very low autocorrelation
- Generally we look at the ACF plot and choose the level of thinning needed to give low autocorrelations
- We will see how this works in some examples

$N(0, 1)$ random sample

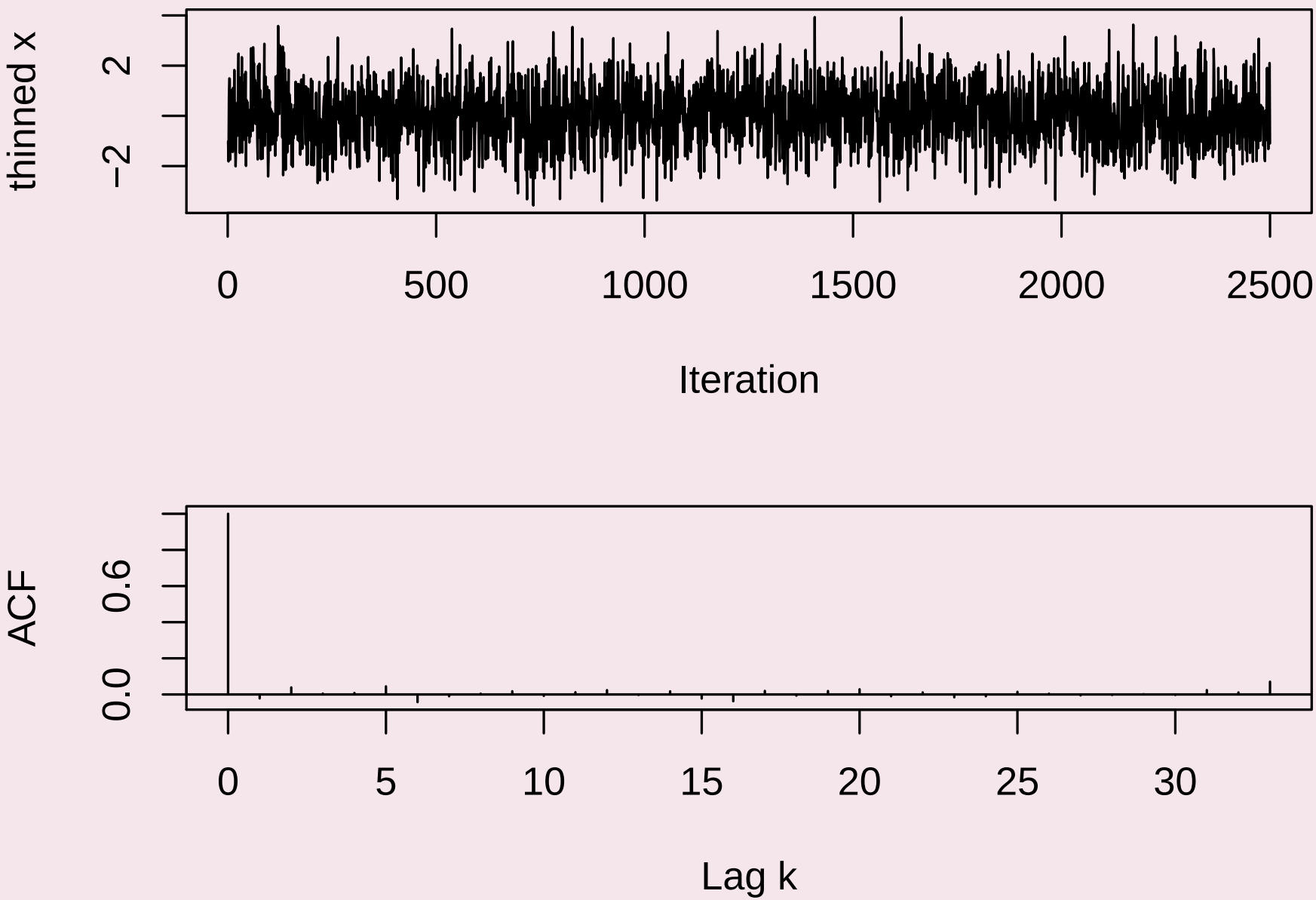


Sample from a $MA(1)$ process

$$x_t = \varepsilon_t + \theta\varepsilon_{t-1}, \text{ where } \varepsilon_t \sim N(0,1), \text{and } r(1) = \theta, r(k) = 0, k \geq 2.$$

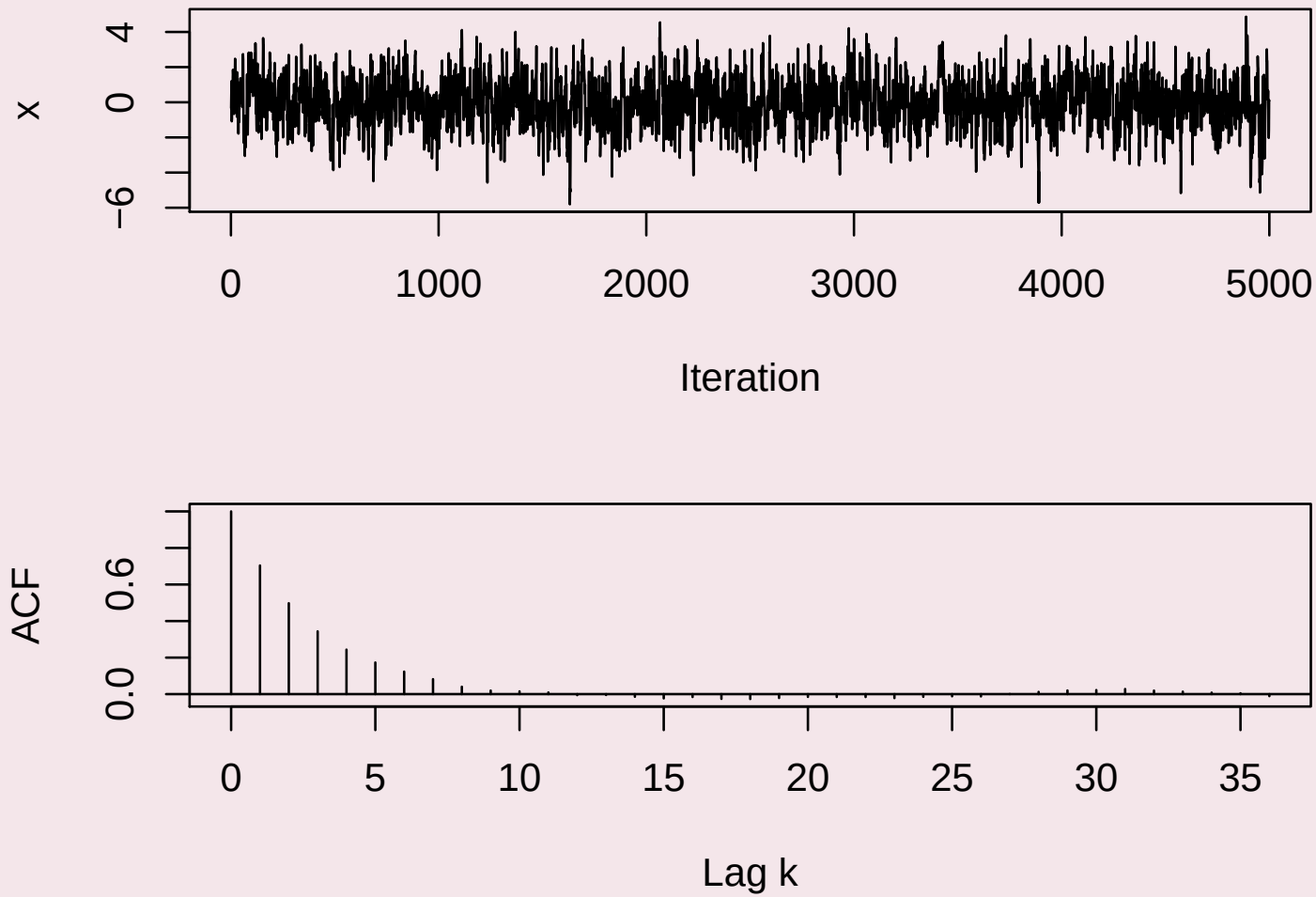


Thinned sample from a $MA(1)$ process ($\texttt{thin} = 2$)

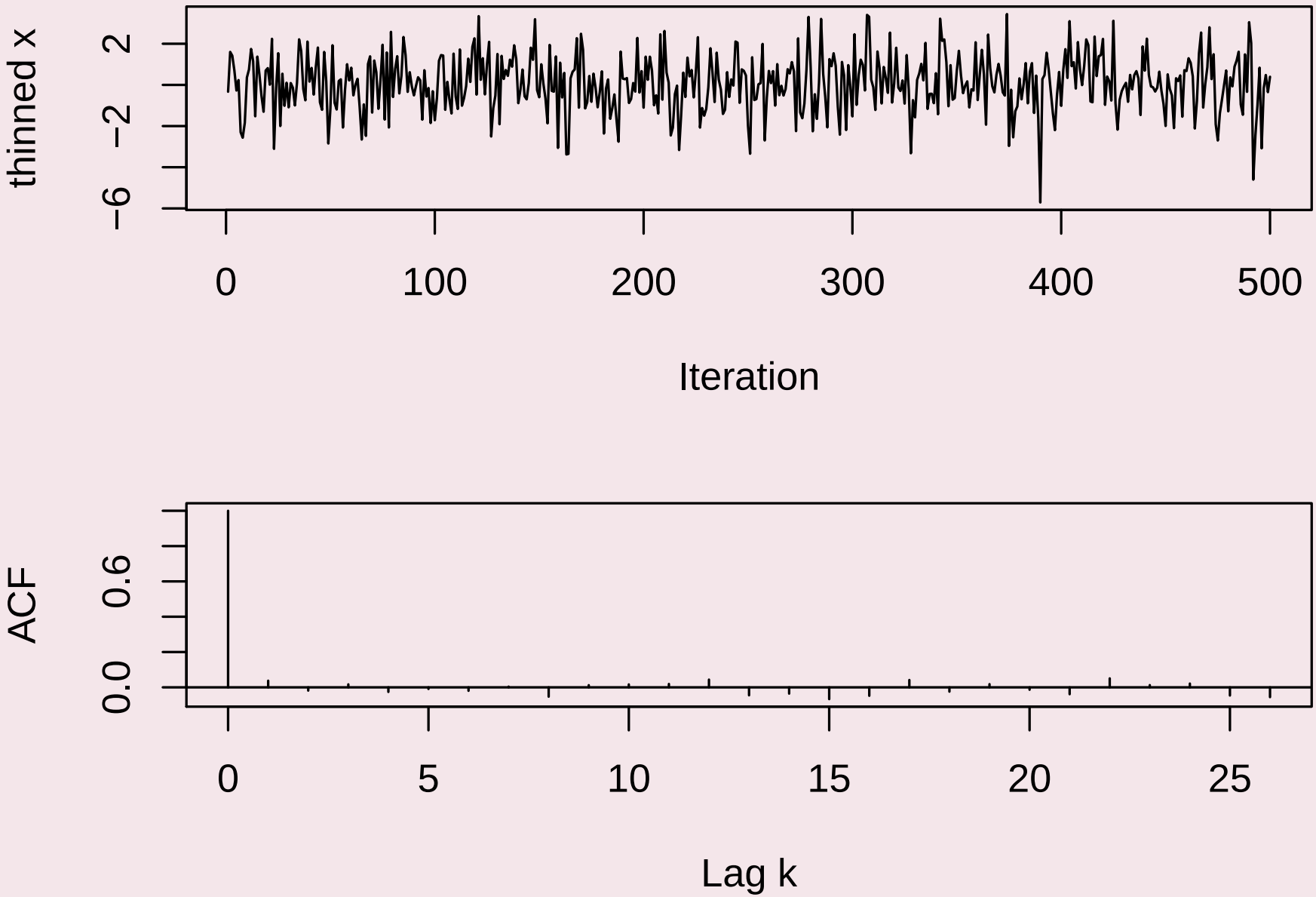


Sample from a $AR(1)$ process

$$x_t = \theta x_{t-1} + \varepsilon_t, \text{ where } \varepsilon_t \sim N(0,1), \text{and } r(k) \simeq r(1)^k, k \geq 2.$$



Thinned sample from a $AR(1)$ process (`thin` $= 10$)



Strategy

1. Determine the *burn-in* period, after which the Gibbs sampler has reached its stationary distribution
   This may involve thinning the posterior sample as slowly snaking trace plots may be due to high autocorrelations rather than a lack of convergence
2. After this, determine the level of thinning to obtain a posterior sample whose autocorrelations are roughly zero
3. Repeat steps 1 and 2 several times using different initial values to make sure that the sample really is from the posterior distribution

4.4.2 Bayesian inference using a Gibbs sampler

Example 4.2

Construct a Gibbs sampler for the posterior distribution in Example 4.1:

- Data: $X_i|\mu,\tau \sim N(\mu, 1/\tau)$, $i = 1, 2, \ldots, n$ (independent)
- Prior:

$$\mu \sim N\left(b, \frac{1}{c}\right) \qquad \text{and} \qquad \tau \sim Ga(g, h), \qquad \text{independent}$$

Solution

. . .

R functions in the nclbayes library

- `gibbsNormal` – implements this algorithm
- `mcmcProcess` – used to remove the `burn-in` and `thin` the output
- `mcmcAnalysis` – analyses the MCMC output

R functions in the nclbayes library

- `gibbsNormal` – implements this algorithm
- `mcmcProcess` – used to remove the `burn-in` and `thin` the output
- `mcmcAnalysis` – analyses the MCMC output

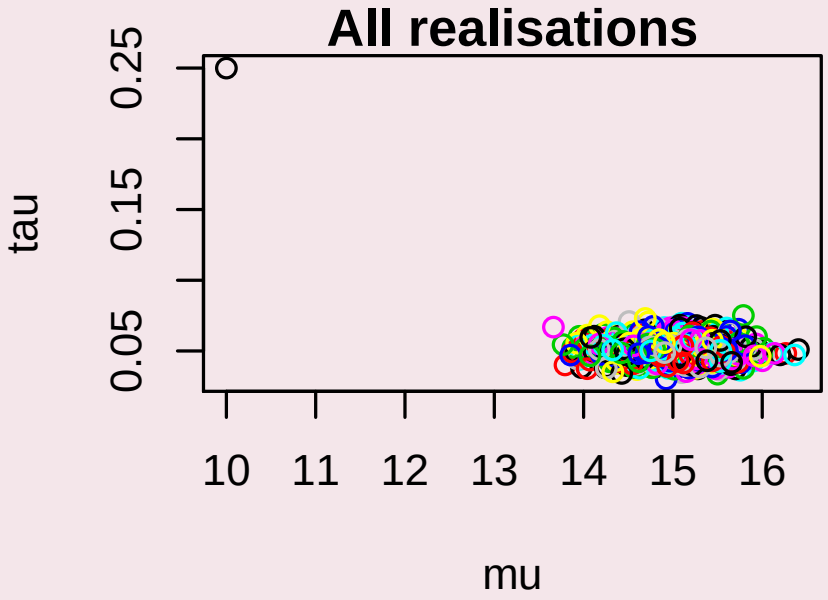
Look at a particular numerical example

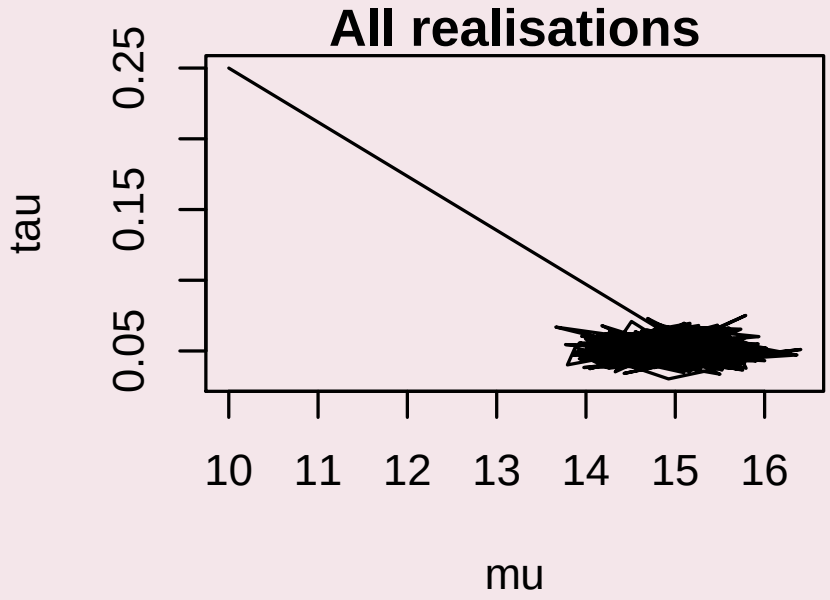
- Data: $n = 100$, $\bar{x} = 15$ and $s = 4.5$
- Prior: $\mu \sim N(10, 1/100)$ and $\tau \sim Ga(3, 12)$, independently
- Initialise Gibbs sampler at $(10, 0.25)$

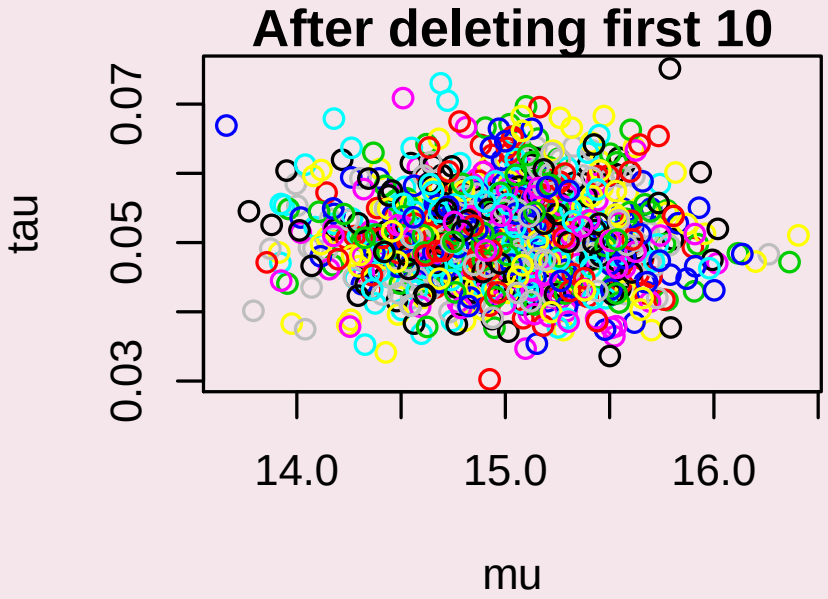
R commands used

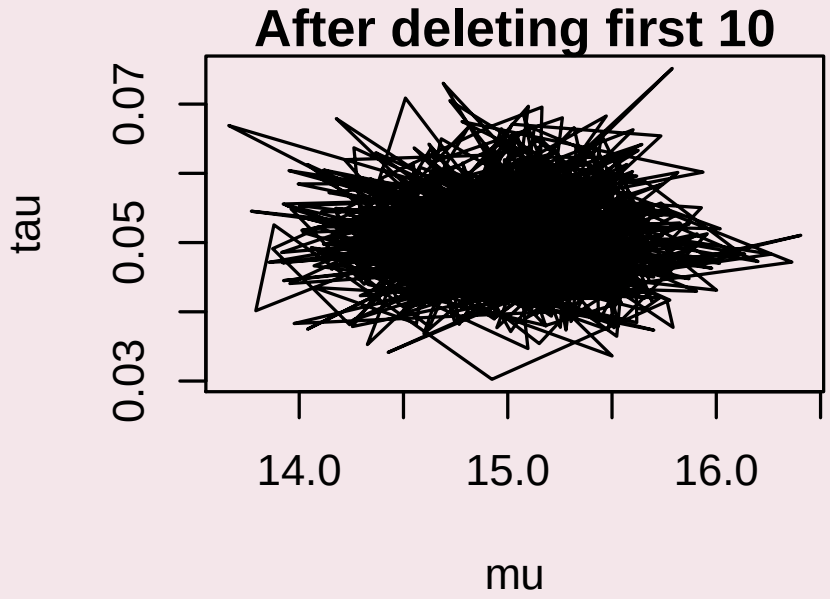
```
library(nclbayes)
posterior=gibbsNormal(N=2000,initial=c(10,0.25),
    priorparam=c(10,1/100,3,12),n=100,xbar=15,s=4.5)
posterior2=mcmcProcess(input=posterior,burnin=1000,thin=1)

op=par(mfrow=c(2,2))
plot(posterior,col=c(1:length(posterior)),
    main="All realisations")
plot(posterior,type="l",main="All realisations")
plot(posterior2,col=c(1:length(posterior2)),
    main="After deleting first 1000")
plot(posterior2,type="l",main="After deleting first 1000")
par(op)

mcmcAnalysis(posterior,rows=2,show=F)
mcmcAnalysis(posterior2,rows=2,show=F)
```

Progress of the MCMC scheme









Analysis of MCMC output (all realisations)

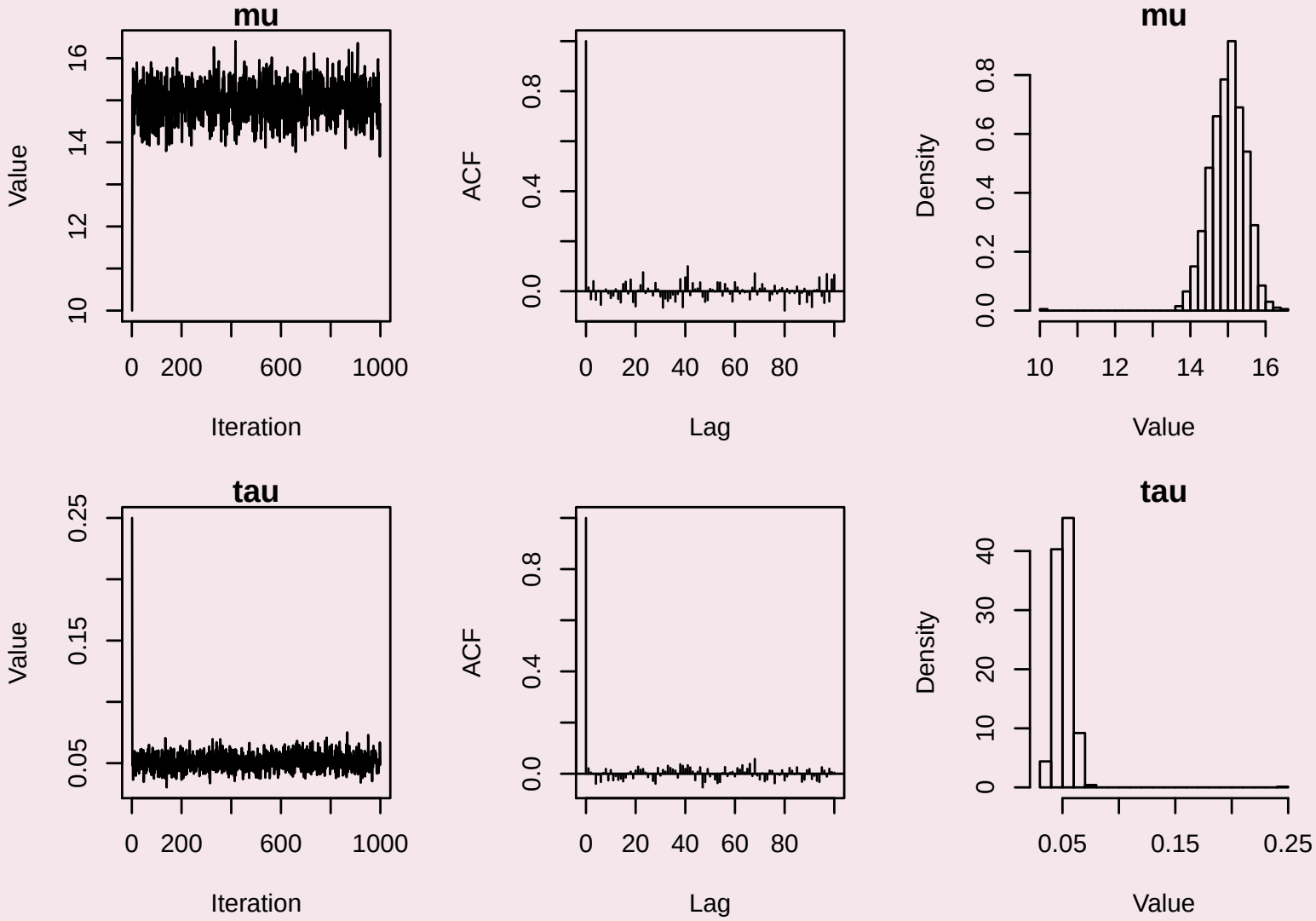


Figure: Trace plots, autocorrelation plots and histograms of the Gibbs sampler output

Analysis of MCMC output (after deleting the first 10 iterations)

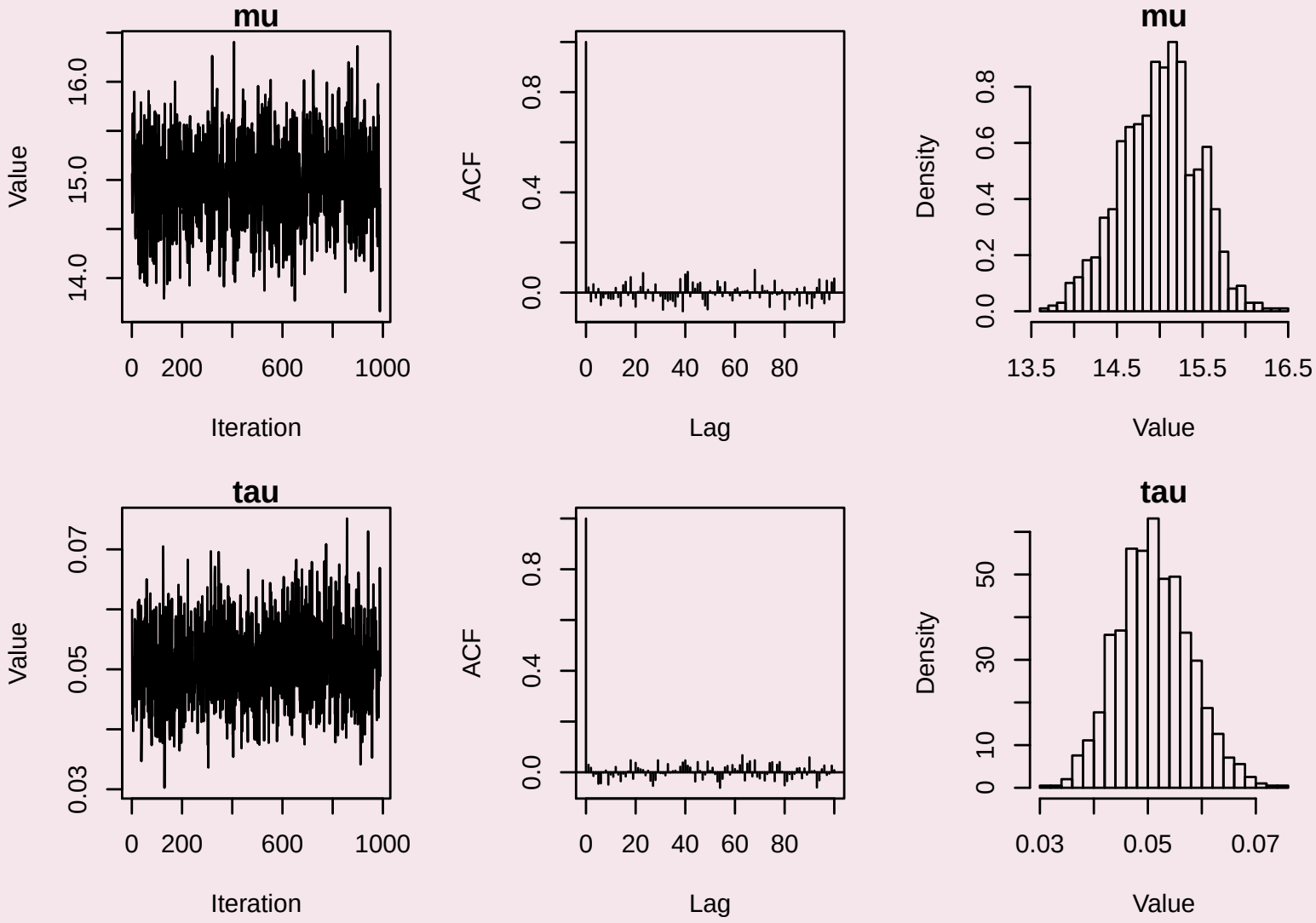


Figure: Trace plots, autocorrelation plots and histograms of the Gibbs sampler output

- Summaries of the output (after deleting the first 10 iterations)

```
N = 1000 iterations
       mu                tau
 Min.   :13.54   Min.   :0.03017
 1st Qu.:14.69   1st Qu.:0.04595
 Median :15.01   Median :0.05089
 Mean   :14.99   Mean   :0.05110
 3rd Qu.:15.28   3rd Qu.:0.05562
 Max.   :16.52   Max.   :0.07253
Standard deviations:
          mu          tau
0.444299473 0.007124057
```

- Also $Corr(\mu, \tau|\boldsymbol{x}) = -0.02153$

- Equi-tailed confidence intervals are calculated as follows:
  - MCMC output has N realisations $(\mu^{(j)}, \tau^{(j)})$
  - Sort the $\mu^{(j)}$ and the $\tau^{(j)}$ into increasing order
  - CI end-points will be the $N\alpha/2$th and the $N(1-\alpha/2)$th values
  - Use `mcmcCi` command in the `nclbayes` package
  - Gives 95% confidence intervals

$$\begin{aligned} \mu: &\quad (14.123,\ 15.847) \\ \tau: &\quad (0.037949,\ 0.065571) \end{aligned}$$

Distributions of functions of the parameters

- Now we have a sample from the posterior distribution, we can determine the posterior distribution for any function of the parameters
- For example, if we want the posterior distribution for $\sigma = 1/\sqrt{\tau}$ then we can easily obtain realisations of σ as

$$\sigma^{(j)} = 1 \Big/ \sqrt{\tau^{(j)}}$$

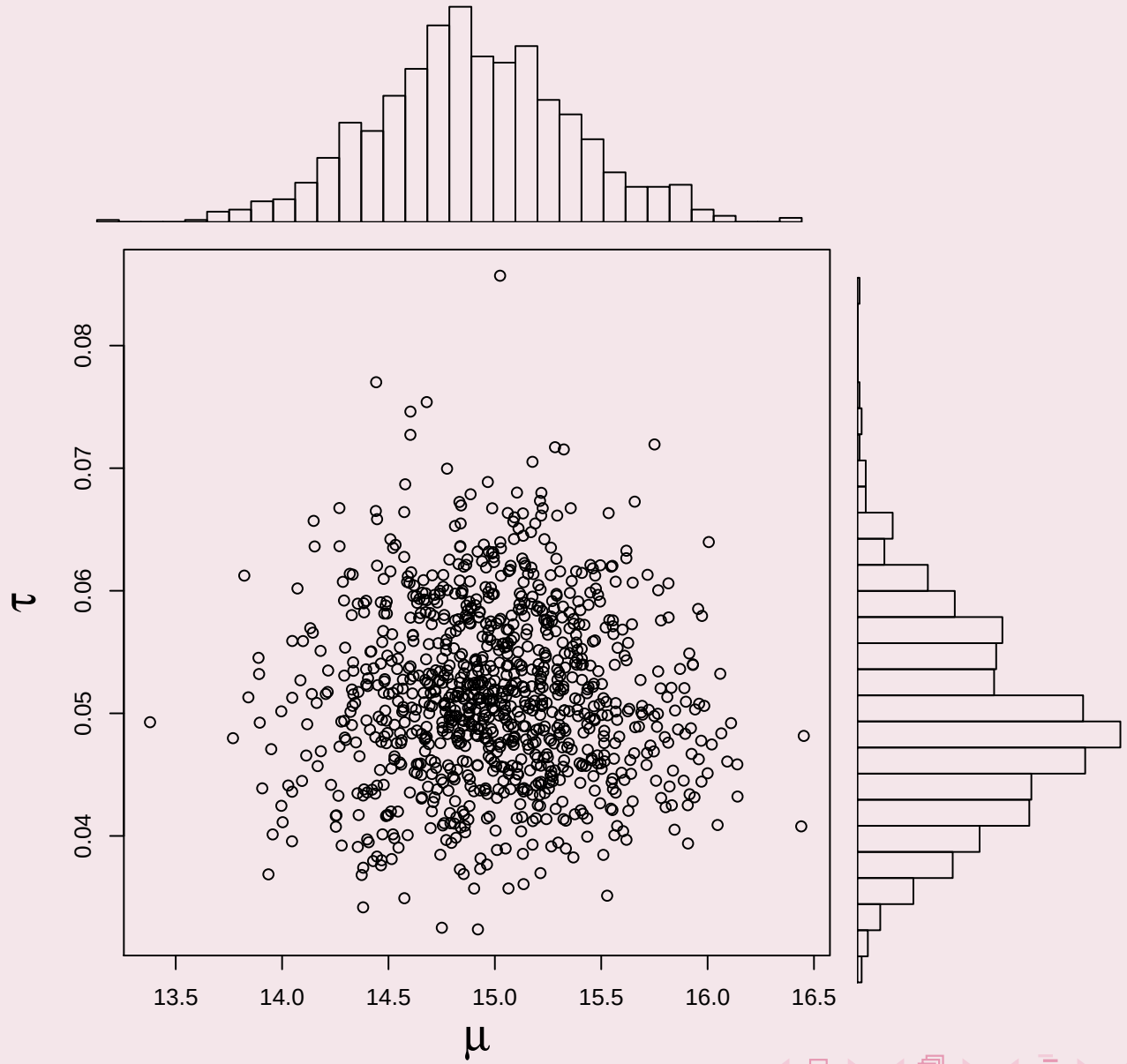
- Summaries of the output for σ

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.713   4.240   4.433   4.457   4.665   5.758
```

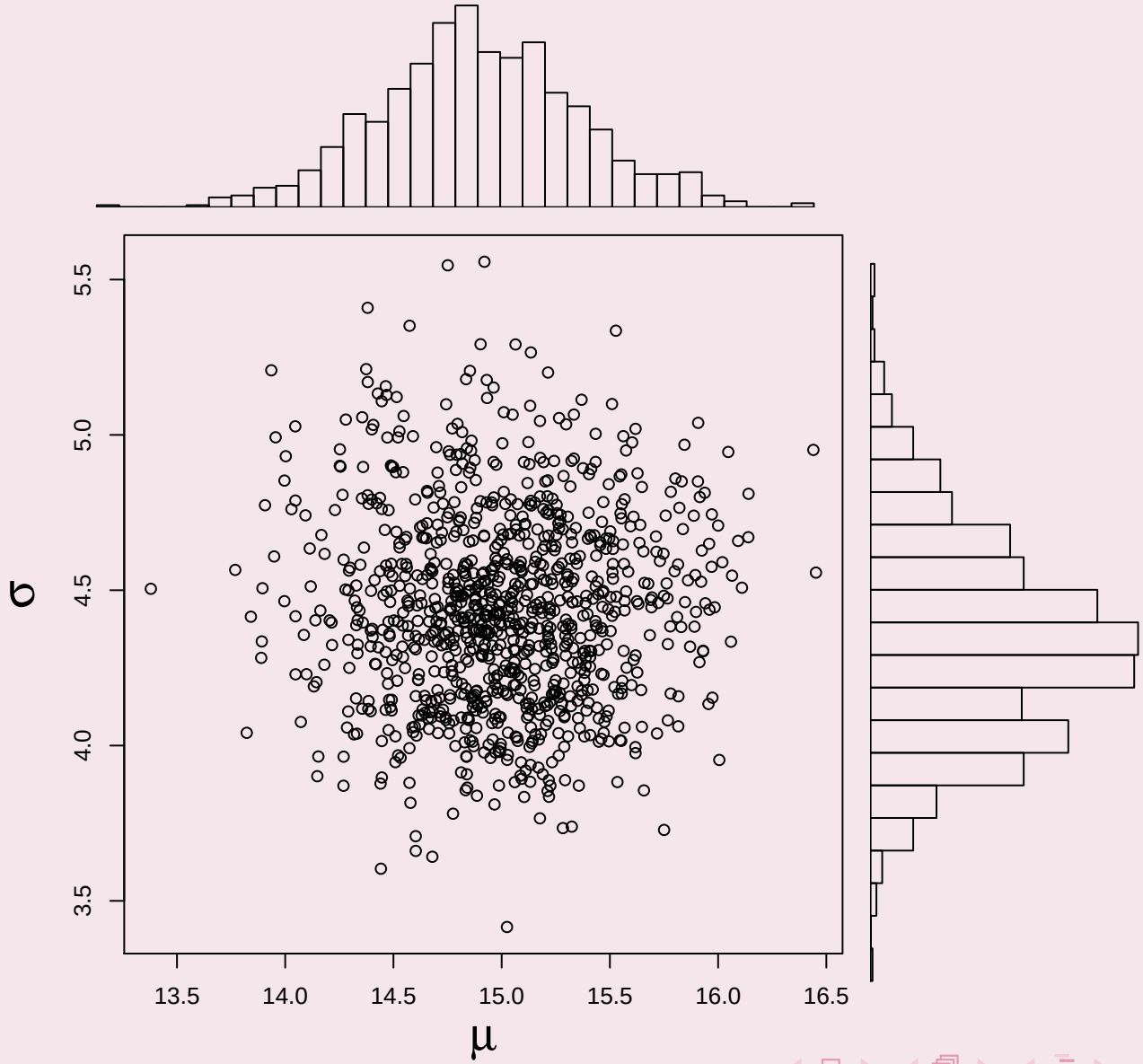
st.dev. 0.3178429

Equi-tailed 95% confidence interval for σ is $(3.9001,\ 5.1192)$.

Joint and marginal distributions of $(\mu, \tau)^T$



Joint and marginal distributions of $(\mu, \sigma)^T$



R commands used

```
library(nclbayes)
posterior=gibbsNormal(N=2000,initial=c(10,0.25),
    priorparam=c(10,1/100,3,12),n=100,xbar=15,s=4.5)
posterior2=mcmcProcess(input=posterior,burnin=1000,thin=1)

mcmcAnalysis(posterior,rows=2,show=F)
mcmcAnalysis(posterior2,rows=2,show=F)

cor(posterior2)
mcmcCi(posterior2,level=0.95)

sigma=1/sqrt(posterior2[,2])
summary(sigma)
sd(sigma)
```

Summary

We can use the (converged and thinned) MCMC output to do the following

- Obtain the posterior distribution for any (joint) functions of the parameters (such as $\sigma = 1/\sqrt{\tau}$ or $(\theta_1 = \mu - \tau, \theta_2 = e^{\mu+\tau/2})^T$)
- Look at bivariate posterior distributions via scatter plots
- Look at univariate marginal posterior distributions via histograms or boxplots
- Obtain numerical summaries such as the mean, standard deviation and confidence intervals for single variables and correlations between variables

Example 4.3

- Gibbs sampling can also be used when using a conjugate prior
- Construct a Gibbs sampler for the problem in Example 3.2: Cavendish's data on the earth's density
- Data: random sample from a normal distribution with unknown mean μ and precision τ, that is

$$X_i|\mu,\tau \sim N(\mu, 1/\tau), \quad i = 1, 2, \ldots, n \quad \text{(independent)}$$

- Prior: *NGa* distribution for $(\mu, \tau)^T$

Solution

. . .

R function in the nclbayes library

- `gibbsNormal2` – implements this algorithm

R function in the nclbayes library

- `gibbsNormal2` – implements this algorithm

Reanalysis of Cavendish's data

- Data: $n = 23$, $\bar{x} = 5.4848$, $s = 0.1882$
- Prior: $NGa(b = 5.41, c = 0.25, g = 2.5, h = 0.1)$
- Have seen in Example 3.2 that
  - posterior is $NGa(B = 5.4840, C = 23.25, G = 14, H = 0.5080)$
  - marginals are

$$\mu|\boldsymbol{x} \sim t_{2G=28}(B = 5.4840, H/(GC) = 0.001561)$$

$$\tau|\boldsymbol{x} \sim Ga(G = 14, H = 0.5080)$$

- We now analyse this problem using the Gibbs sampler
- Can check that the Gibbs sampler is producing realisations from the correct distribution by comparing its output with the above theoretical marginal distributions

Main R commands used

```
library(nclbayes)
posterior=gibbsNormal2(N=2000,initial=c(5.41,25),
    priorparam=c(5.41,0.25,2.5,0.1),n=23,xbar=5.4848,s=0.1882)
posterior2=mcmcProcess(input=posterior,burnin=1000,thin=1)

mcmcAnalysis(posterior,rows=2,show=F)
mcmcAnalysis(posterior2,rows=2,show=F)
```

Analysis of MCMC output (after deleting the first 1000 iterations)

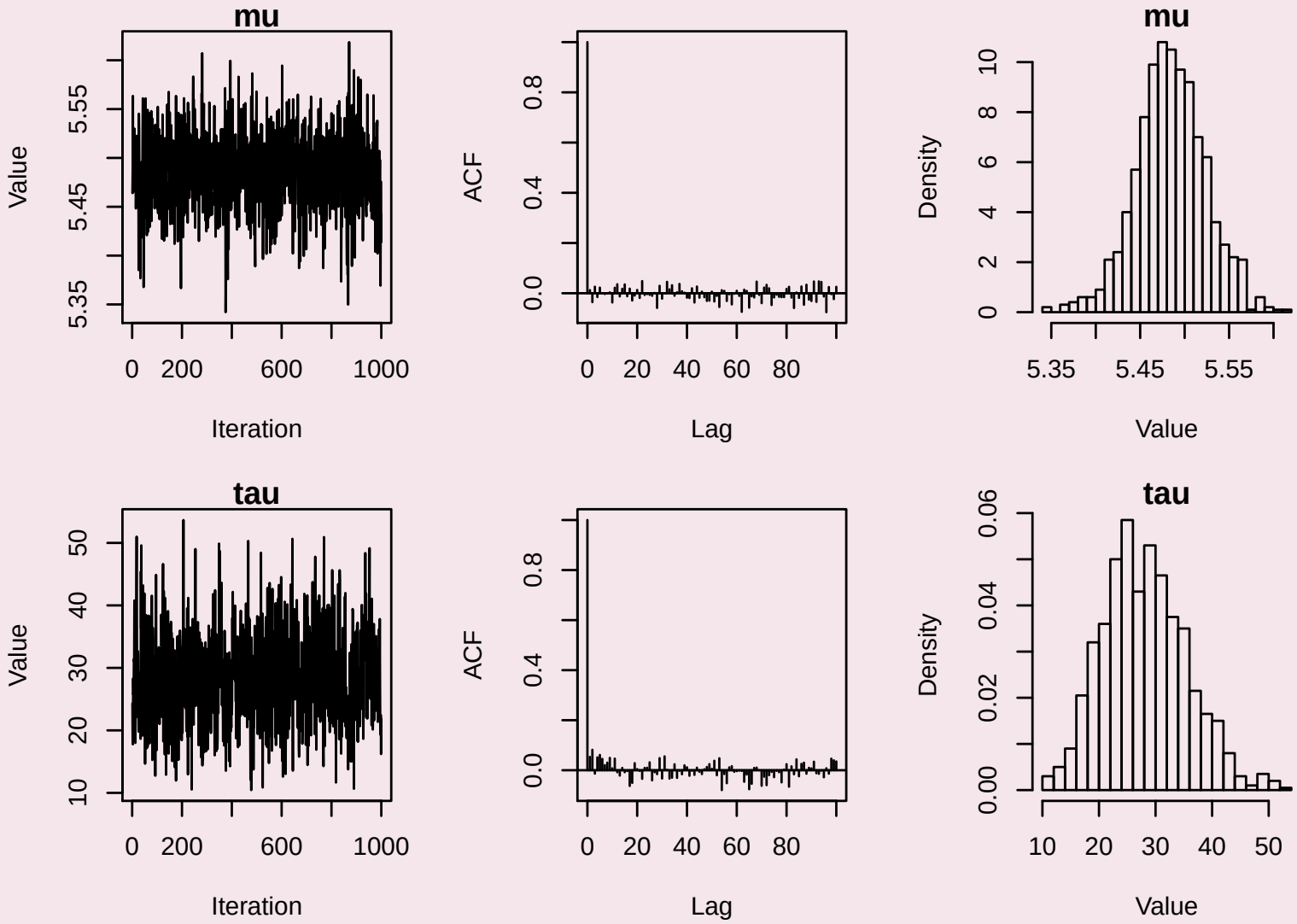


Figure: Trace plots, autocorrelation plots and histograms of the Gibbs sampler output

- Summaries of the output (after deleting the first 1000 iterations)

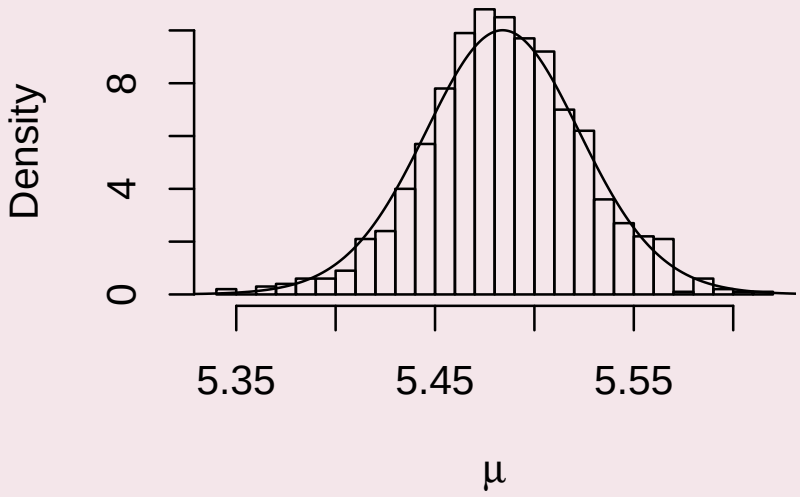
```
N = 1000 iterations
       mu              tau
 Min.   :5.376   Min.   :11.01
 1st Qu.:5.456   1st Qu.:22.09
 Median :5.483   Median :27.06
 Mean   :5.484   Mean   :27.66
 3rd Qu.:5.512   3rd Qu.:32.60
 Max.   :5.605   Max.   :56.16
Standard deviations:
        mu         tau
0.04090114 7.47836536
```

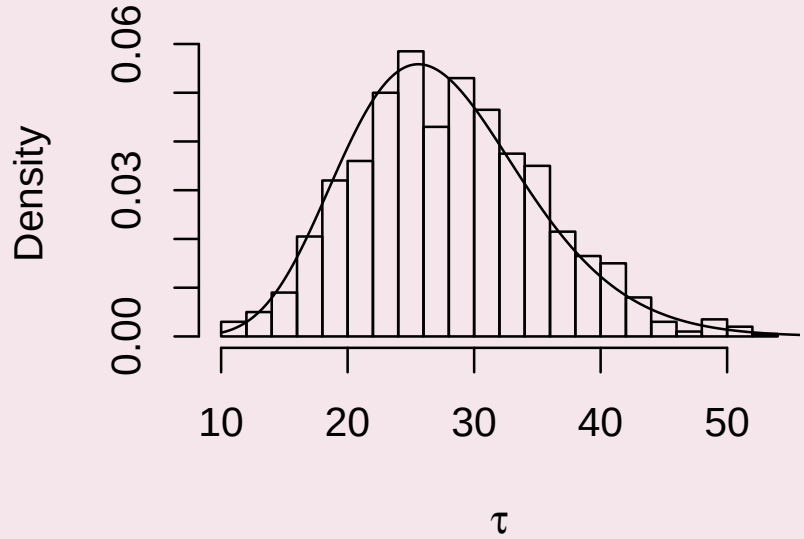
- Check:

$$E(\mu|\boldsymbol{x}) = B = 5.4840, \qquad SD(\mu|\boldsymbol{x}) = \sqrt{\frac{H}{(G-1)C}} = 0.04100$$

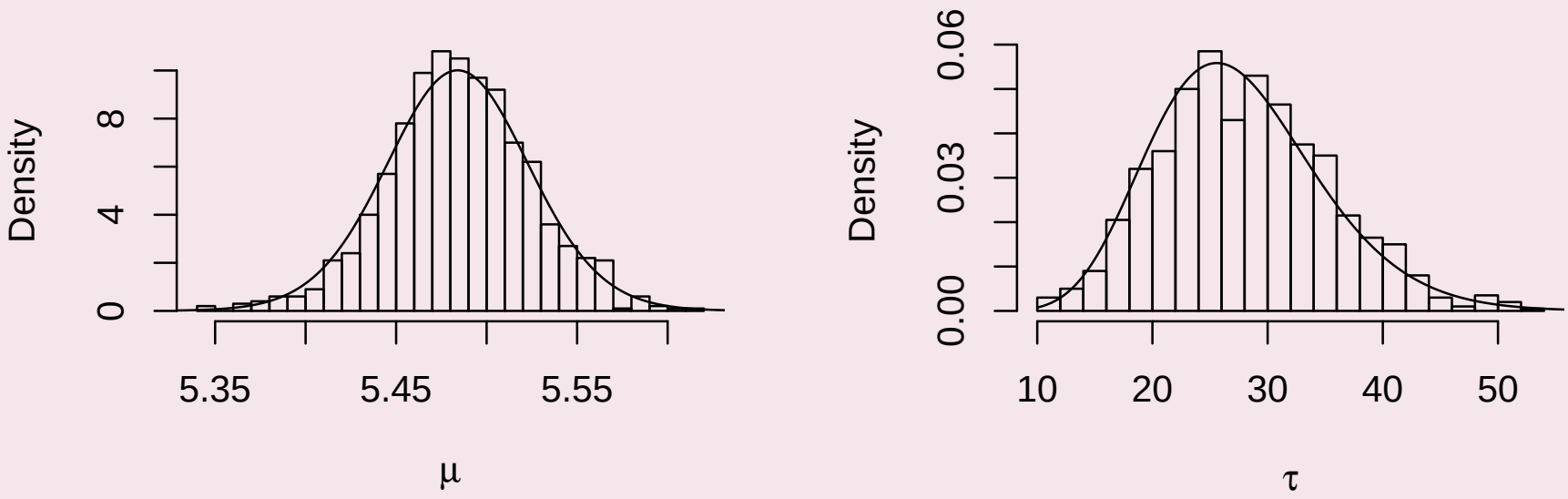
$$E(\tau|\boldsymbol{x}) = \frac{G}{H} = 27.559, \qquad SD(\tau|\boldsymbol{x}) = \frac{\sqrt{G}}{H} = 7.3655$$

Marginal posterior distributions of μ and σ





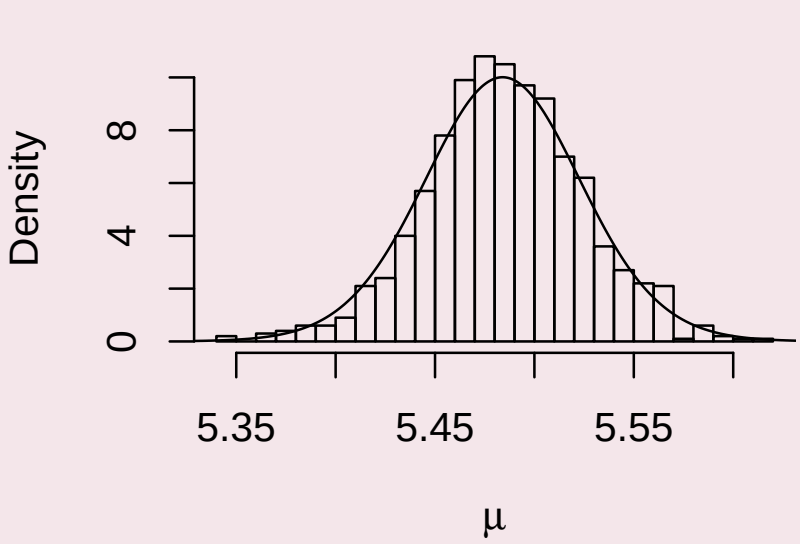
Marginal posterior distributions of μ and σ

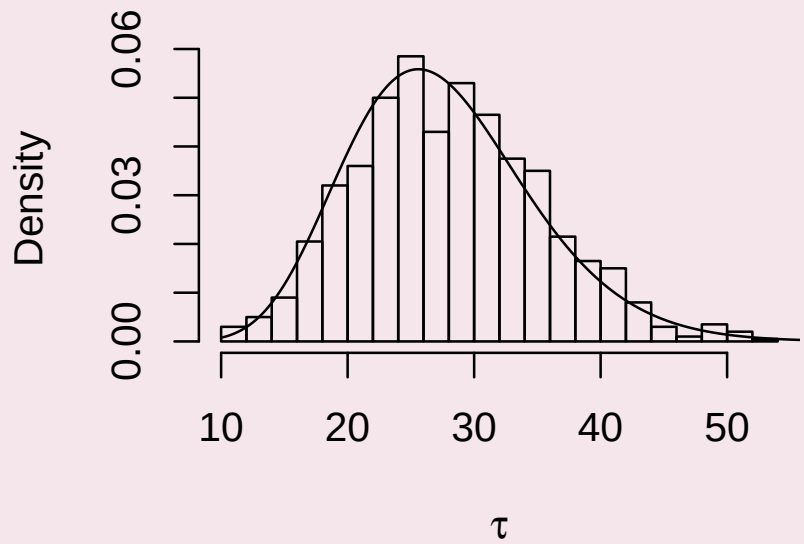


Conclusion

- Posterior means match pretty closely
- Posterior standard deviations match pretty closely
- Marginal posterior denisties match pretty closely

Marginal posterior distributions of μ and σ

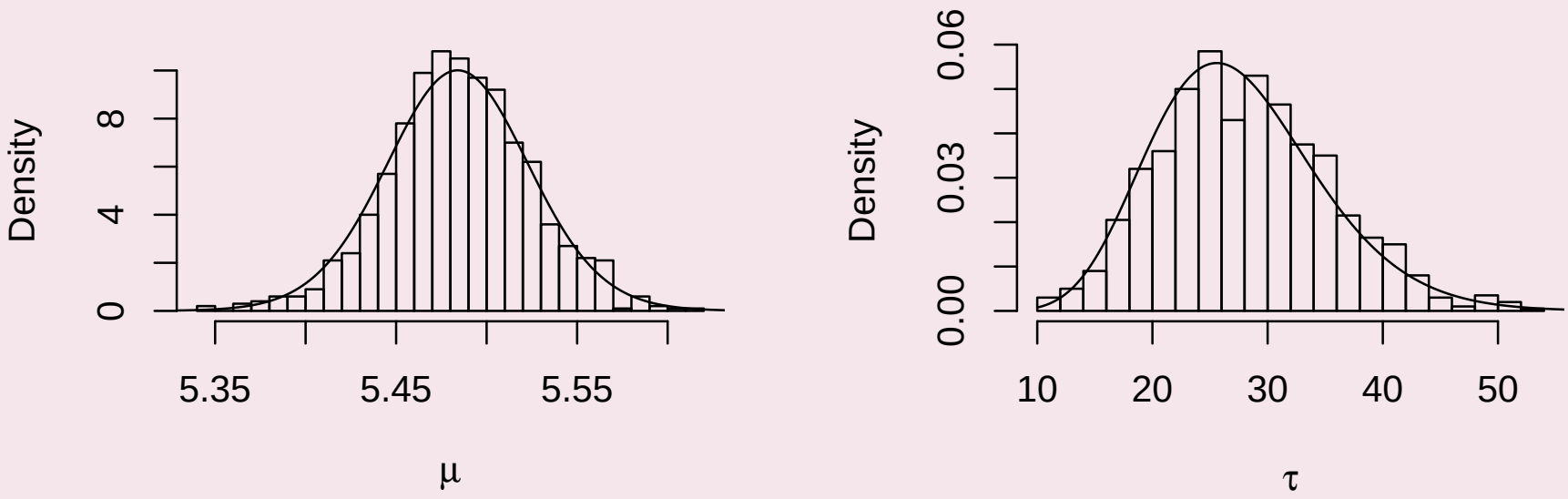




Conclusion

- Posterior means match pretty closely
- Posterior standard deviations match pretty closely
- Marginal posterior denisties match pretty closely
- They are close enough within sampling error – but could run sampler for more iterations if we wanted more accurate estimates

Marginal posterior distributions of μ and σ



Conclusion

- Posterior means match pretty closely
- Posterior standard deviations match pretty closely
- Marginal posterior denisties match pretty closely
- They are close enough within sampling error – but could run sampler for more iterations if we wanted more accurate estimates
- This confirms that the Gibbs sampler does indeed produce realisations from the correct posterior distribution

Accuracy of posterior estimates

- Posterior output: sample mean $\bar{\mu}$, standard deviation s_μ
- How accuracte are these estimates of $M = E(\mu|\boldsymbol{x})$ and $\Sigma = SD(\mu|\boldsymbol{x})$?

- Each time we run an MCMC scheme, we obtain a different sample from the posterior distribution
- Suppose we have a large sample with effective sample size N
- It's easiest to think of this as being a sample which has been thinned so that it has N values, say $\mu_1, \ldots, \mu_N$ (and effective sample size N)
- To quantify accuracy, we need to make an assumption about the posterior distribution
- If the data sample size n is large then the posterior distribution will be approximately normal
- Think of the MCMC output as being a random sample from a normal distribution

- When the random sample $\theta_1, \ldots, \theta_N$ is from the normal $N(a, 1/d)$, from example 3.1, the asymptotic posterior distribution about the mean and precision $(a, d)^T$ is

$$a|\boldsymbol{\theta} \sim N(\bar{\theta}, s^2/N), \qquad d|\boldsymbol{\theta} \sim N\{1/s^2, 2/(Ns^4)\}, \qquad \text{independently}$$

- If the Gibbs outputs $\mu_1, \ldots, \mu_N$ come from a normal distribution $N(M, \Sigma^2)$, rewriting this result in terms of the MCMC sample mean $\bar{\mu}$, standard deviation s_μ and the parameters they estimate gives posterior distributions

$$M \sim N(\bar{\mu}, s_\mu^2/N), \qquad \Sigma^{-2} \sim N\{1/s_\mu^2, 2/(Ns_\mu^4)\}, \qquad \text{independently}$$

- Therefore an approximate 95% HDI for M is

$$\bar{\mu} \pm z_{0.025} \frac{s_\mu}{\sqrt{N}} \simeq \bar{\mu} \pm \frac{2s_\mu}{\sqrt{N}}$$

since $z_{0.025} \simeq 2$.

- Also, from the posterior distribution for Σ^{-2}, we have

$$P\left(\frac{1}{s_\mu^2} - 2\sqrt{\frac{2}{Ns_\mu^4}} < \Sigma^{-2} < \frac{1}{s_\mu^2} + 2\sqrt{\frac{2}{Ns_\mu^4}}\right) \simeq 0.95$$

$$\implies \quad P\left(\frac{1 - 2\sqrt{2/N}}{s_\mu^2} < \Sigma^{-2} < \frac{1 + 2\sqrt{2/N}}{s_\mu^2}\right) \simeq 0.95$$

$$\implies \quad P\left(\frac{s_\mu}{\sqrt{1 + 2\sqrt{2/N}}} < \Sigma < \frac{s_\mu}{\sqrt{1 - 2\sqrt{2/N}}}\right) \simeq 0.95.$$

  Therefore a 95% confidence interval for Σ is

$$s_\mu\left(1 \pm 2\sqrt{2/N}\right)^{-1/2} \simeq s_\mu\left(1 \pm \frac{1}{2} \times 2\sqrt{2/N}\right) = s_\mu \pm s_\mu\sqrt{\frac{2}{N}}$$

- It can be shown that these accuracy calculations are fairly accurate even when the posterior distribution (from which we have the MCMC sample) is not particularly normal

Example 4.4

- Data: $X_i|\alpha, \lambda \sim Ga(\alpha, \lambda)$, $i = 1, 2, \ldots, n$ (independent), where the index α and scale parameter λ are unknown
- Prior: $\alpha \sim Ga(a, b)$ and $\lambda \sim Ga(c, d)$, independent
- Determine the posterior density for $(\alpha, \lambda)^T$ and hence the posterior conditional densities for $\alpha|\lambda$ and $\lambda|\alpha$

Solution

. . .

Problem

- Although the FCD for λ is a standard distribution and easy to simulate from, the FCD for α is NOT!
- Therefore we can't use a Gibbs sampler for this analysis
- Fortunately there are other methods we can use . . .

4.5 Metropolis–Hastings sampling

- The Gibbs sampler is a very powerful tool
- Only useful if the full conditional distributions (FCDs) are standard distributions (which are easy to simulate from)
- Fortunately there is a class of methods which can be used when the FCDs are non-standard
- These methods are known as Metropolis-Hastings schemes

- Want to simulate realisations from the posterior density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$
- All of the FCDs are non-standard
- Choose a proposal distribution with density $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$, which is easy to simulate from
- This distribution gives us a way of proposing new values $\boldsymbol{\theta}^*$ from the current value $\boldsymbol{\theta}$

Metropolis–Hastings algorithm

1. Initialise the iteration counter to $j = 1$, and initialise the chain to $\boldsymbol{\theta}^{(0)}$
2. Generate a proposed value $\boldsymbol{\theta}^*$ using the proposal distribution $q(\boldsymbol{\theta}^* | \boldsymbol{\theta}^{(j-1)})$

Metropolis–Hastings algorithm

1. Initialise the iteration counter to $j = 1$, and initialise the chain to $\boldsymbol{\theta}^{(0)}$
2. Generate a proposed value $\boldsymbol{\theta}^*$ using the proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(j-1)})$
3. Evaluate the acceptance probability $\alpha(\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^*)$ of the proposed move, where

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{x})\, q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta}|\boldsymbol{x})\, q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\right\}$$

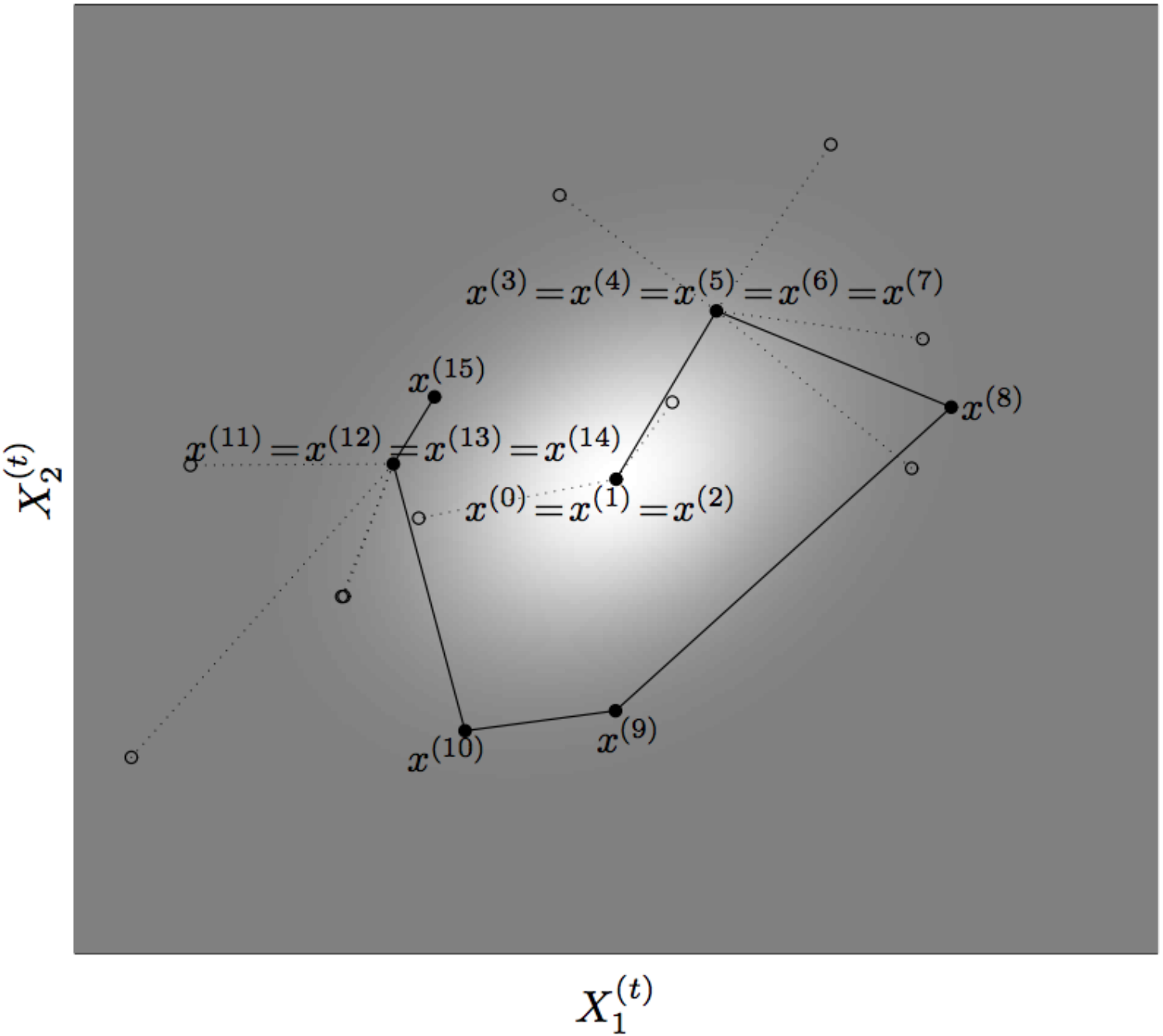
4. Set $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^*$ with probability $\alpha(\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^*)$, and set $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{(j-1)}$ otherwise

Metropolis–Hastings algorithm

1. Initialise the iteration counter to $j = 1$, and initialise the chain to $\boldsymbol{\theta}^{(0)}$
2. Generate a proposed value $\boldsymbol{\theta}^*$ using the proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(j-1)})$
3. Evaluate the acceptance probability $\alpha(\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^*)$ of the proposed move, where

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{x})\, q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta}|\boldsymbol{x})\, q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\right\}$$

4. Set $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^*$ with probability $\alpha(\boldsymbol{\theta}^{(j-1)}, \boldsymbol{\theta}^*)$, and set $\boldsymbol{\theta}^{(j)} = \boldsymbol{\theta}^{(j-1)}$ otherwise
5. Change the counter from j to $j + 1$ and return to step 2


$x^{(3)} = x^{(4)} = x^{(5)} = x^{(6)} = x^{(7)}$
$x^{(15)}$
$x^{(8)}$
$x^{(11)} = x^{(12)} = x^{(13)} = x^{(14)}$
$x^{(0)} = x^{(1)} = x^{(2)}$
$x^{(10)}$
$x^{(9)}$
$X_2^{(t)}$
$X_1^{(t)}$

Comments

- At each stage, a new value is generated from the proposal distribution
- This value is either accepted, in which case the chain moves, or rejected, in which case the chain stays where it is
- Whether or not the move is accepted or rejected depends on an acceptance probability which itself depends on the relationship between the density of interest and the proposal distribution
- The posterior density $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ only enters into the acceptance probability as a ratio, and so the method can be used when the posterior density is only known up to a scaling constant, that is,

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*)\, f(\boldsymbol{x}|\boldsymbol{\theta}^*)\, q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta})\, f(\boldsymbol{x}|\boldsymbol{\theta})\, q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\right\}$$

- This algorithm defines a Markov chain with $\pi(\boldsymbol{\theta}|\boldsymbol{x})$ as its stationary distribution
- It holds for all possible proposal distributions (subject to them generating realisations from the full parameter space)

Possible proposal distributions

- Are some choices of proposal distribution better than others?
- We now discuss some commonly used proposal distributions

Possible proposal distributions

- Are some choices of proposal distribution better than others?
- We now discuss some commonly used proposal distributions

4.5.1 Symmetric chains (Metropolis method)

- Use a symmetric proposal distribution: $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}^*), \forall\ \boldsymbol{\theta}, \boldsymbol{\theta}^*$
- The acceptance probability simplifies to

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{x})}{\pi(\boldsymbol{\theta}|\boldsymbol{x})}\right\}$$

  and hence does not involve the proposal density at all
- Proposed moves which will take the chain to a region of higher posterior density are always accepted
- Moves which take the chain to a region of lower posterior density are accepted with probability proportional to the ratio of the two densities
- Moves which will take the chain to a region of very low density will be accepted with very low probability

- Any proposal of the form $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = f(|\boldsymbol{\theta}^* - \boldsymbol{\theta}|)$ is symmetric, where $f(\cdot)$ is an arbitrary density. In this case, the proposal value is a symmetric displacement from the current value

- Any proposal of the form $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = f(|\boldsymbol{\theta}^* - \boldsymbol{\theta}|)$ is symmetric, where $f(\cdot)$ is an arbitrary density. In this case, the proposal value is a symmetric displacement from the current value

Random walk proposals

- Consider the random walk proposal in which the proposed value $\boldsymbol{\theta}^*$ at stage j is

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(j-1)} + \boldsymbol{w}_j$$

  where the $\boldsymbol{w}_j$ are independent and identically distributed random $p \times 1$ vectors (completely independent of the state of the chain)
- Suppose that the $\boldsymbol{w}_j$ have density $f(\cdot)$, which is easy to simulate from, has mean $\mathbf{0}$ and is symmetric about its mean
- We can then simulate an innovation $\boldsymbol{w}_j$, and set the proposal value to $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(j-1)} + \boldsymbol{w}_j$
- Clearly $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = f(|\boldsymbol{\theta}^* - \boldsymbol{\theta}|)$
- However, what distribution should we use for $f(\cdot)$?

Choice of innovation distribution

- A distribution which is simple and easy to simulate from is always a good idea; for example, the uniform or normal ……
  Normal is generally better, but is a bit more expensive to simulate
- What variance should we choose for the innovation distribution?
- This choice will affect the acceptance probability, and hence the overall proportion of accepted moves
- If the variance of the innovation is too low, then most proposed values will be accepted, but the chain will move very slowly around the space — the chain is said to be too "cold"
- If the variance of the innovation is too large, very few proposed values will be accepted, but when they are, they will often correspond to quite large moves — the chain is said to be too "hot"
- Theoretically it has been shown that the optimal acceptance rate is around 0.234 — this is an asymptotic result (for large samples of data) — but experience suggests that an acceptance rate of around 20–30% is okay
- Thus, the variance of the innovation distribution should be "tuned" to get an acceptance rate of around this level

Example 4.5

- Suppose the posterior distribution is a standard normal distribution, with density $\phi(\cdot)$
- Construct a Metropolis–Hastings algorithm which samples this posterior distribution by using a uniform random walk proposal
- Examine how the acceptance rate for this algorithm depends on the width of the uniform distribution

Solution

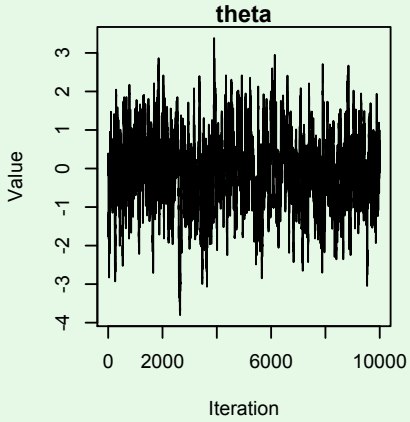
. . .

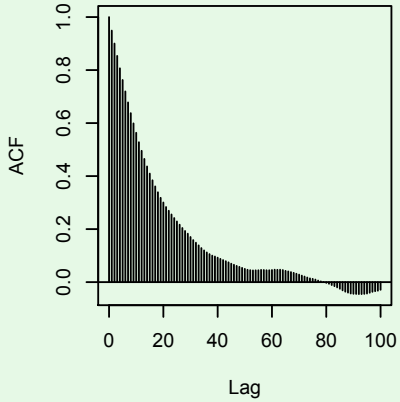
R commands

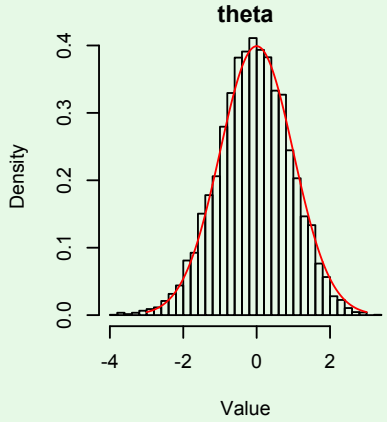
```
posterior=metropolis(N=10000,initial=0,a=1)
mcmcAnalysis(posterior,rows=1,show=F)
```

Sampling from a standard normal using Metropolis-Hastings

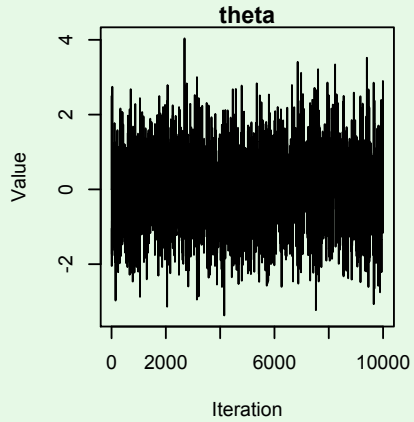
$a = 0.6$

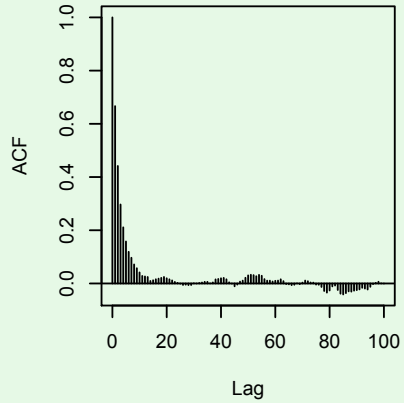


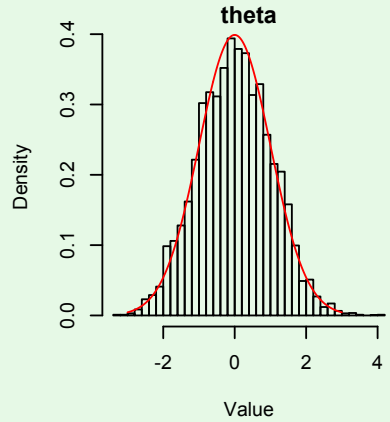




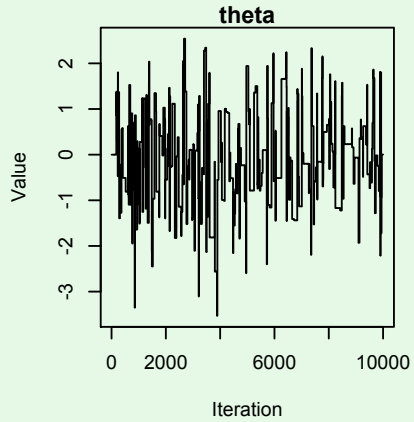
$a = 6$

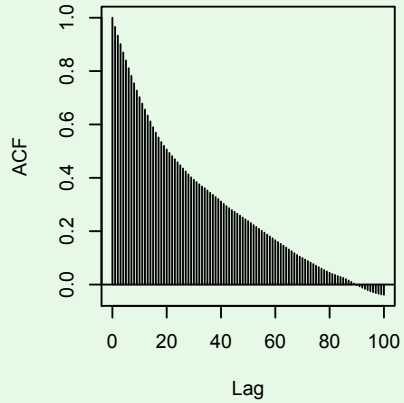


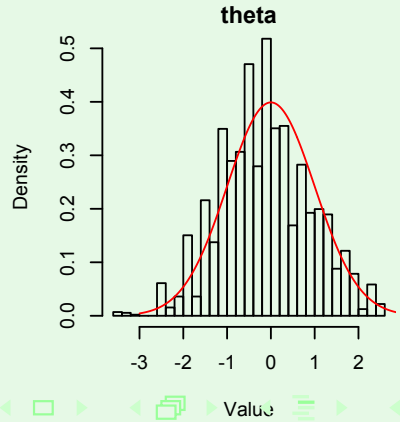




$a = 60$





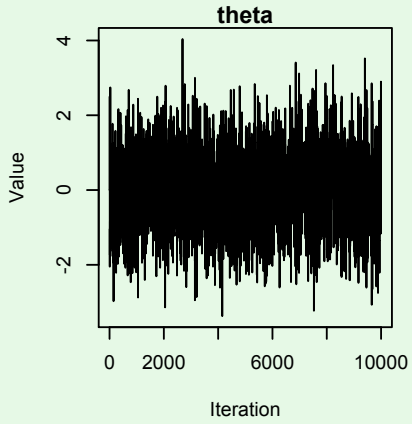


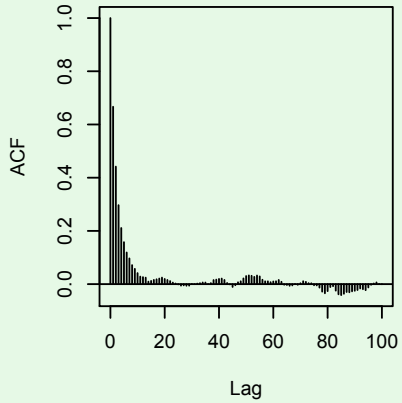
Comments

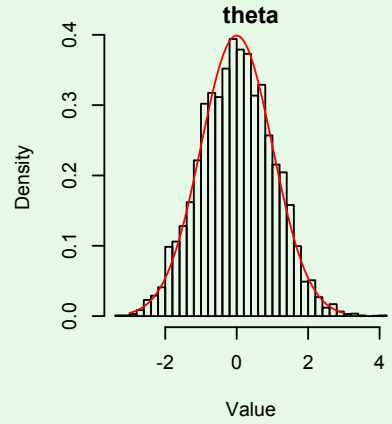
- $a = 0.6$ – this chain is too "cold"
  - The innovations are too small and are generally accepted
  - The acceptance rate for this chain was 0.911
  - The autocorrelations are too high and this chain would have to be thinned
- $a = 6$
  - The autocorrelations are much lower
  - The acceptance rate was 0.392 (nearer the asymptotic 0.234 M–H acceptance rate)
- $a = 60$ – this chain is too "hot"
  - Few proposed values are accepted (acceptance rate 0.039)
  - When they are, it results in a fairly large move to the chain
  - This gives fairly high autocorrelations and this chain would have to be thinned

Sampling from a standard normal using Metropolis-Hastings

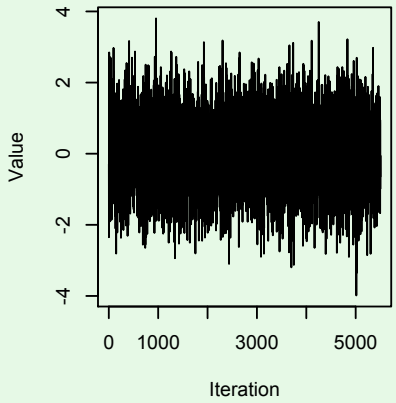
$a = 6$

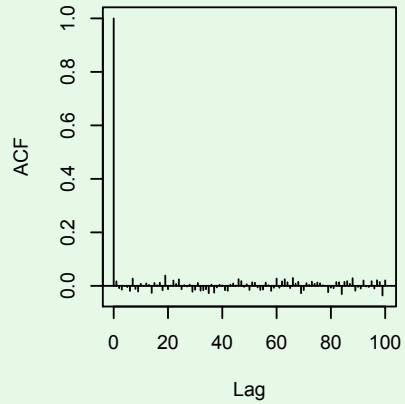


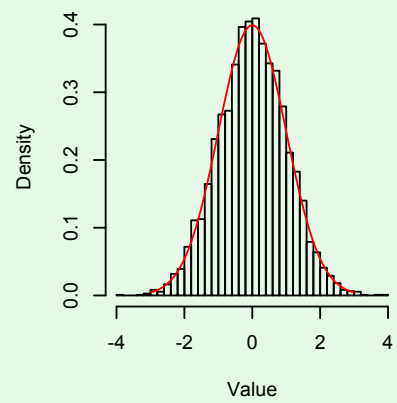




thinned







Normal random walk proposals

- Suppose we decide to use a normal random walk with $f(\cdot) = N_p(\mathbf{0}, \Sigma)$
- The proposal distribution is

$$q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = N_p(\boldsymbol{\theta}, \Sigma)$$

- Tuning this random walk requires us to choose Σ
  How can we do this?
- If the posterior distribution is approximately normally distributed (as it is with large data samples) then researchers have shown that the optimal choice is

$$\Sigma = \frac{2.38^2}{p} Var(\boldsymbol{\theta}|\boldsymbol{x})$$

- In practice, we don't know the posterior variance $Var(\boldsymbol{\theta}|\boldsymbol{x})$

- However, we could first run the MCMC algorithm substituting in the (generally much larger) prior variance $Var(\boldsymbol{\theta})$, that is, take

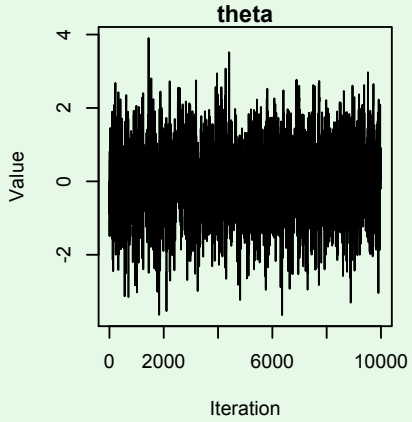
$$\Sigma = \frac{2.38^2}{p} Var(\boldsymbol{\theta})$$

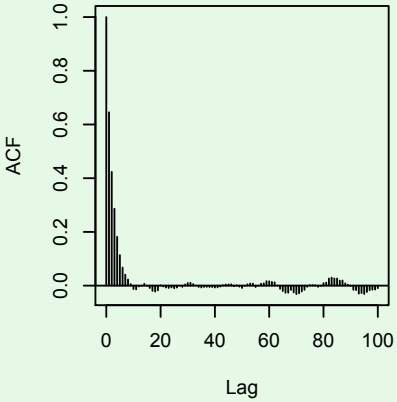
- If this chain doesn't converge quickly then we can use the output of this MCMC run to get a better idea of $Var(\boldsymbol{\theta}|\boldsymbol{x})$ and run the MCMC code again – this will have more appropriate values for the parameter variances and correlations
- It has been shown from experience that it is not vital to get an extremely accurate value for Σ
- Often just getting the correct order of magnitude for its elements will be sufficient, that is, using say 0.1 rather than 0.01 or 1

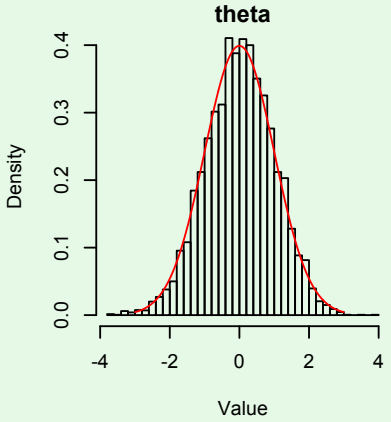
- Example: Assume the posterior distribution is a standard normal distribution. Then the optimal $\Sigma = 5.7$.

Sampling from a standard normal using normal random walk

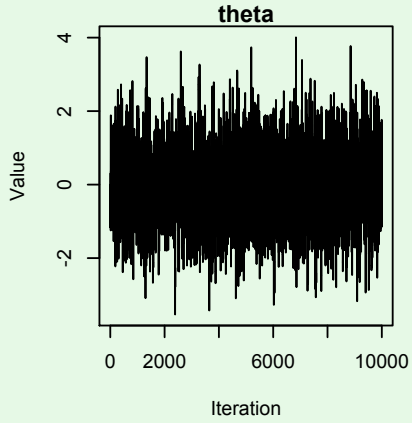
$\Sigma = 11$

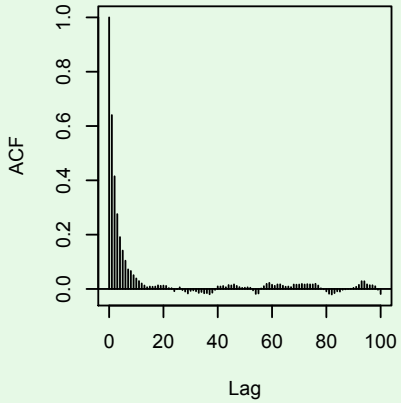


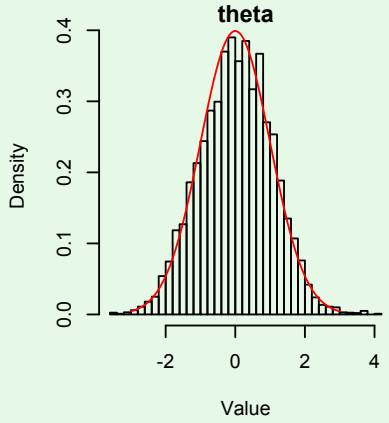




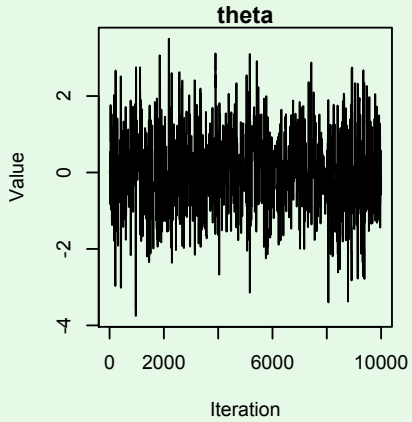
$\Sigma = 5.7$

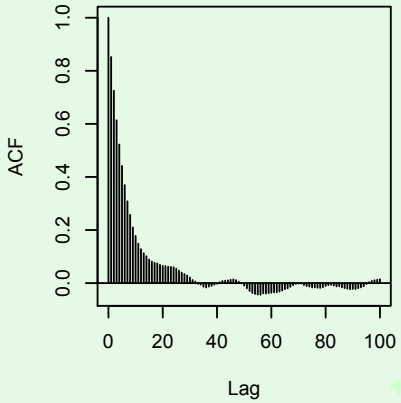


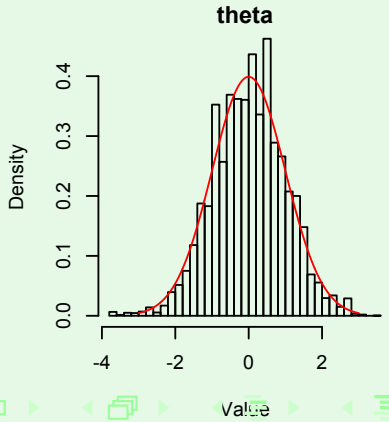




$\Sigma = 100$







4.5.2 Independence chains

- The proposal is formed independently of the position of the chain, and so $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = f(\boldsymbol{\theta}^*)$ for some density $f(\cdot)$
- The acceptance probability is

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{x})}{\pi(\boldsymbol{\theta}|\boldsymbol{x})} \Big/ \frac{f(\boldsymbol{\theta}^*)}{f(\boldsymbol{\theta})} \right\}$$

- A nice thing is that, when $f(\cdot)$ is close to $\pi(\cdot|\boldsymbol{x})$, the acceptance probability can be close to 1.

4.5.2 Independence chains

- The proposal is formed independently of the position of the chain, and so $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = f(\boldsymbol{\theta}^*)$ for some density $f(\cdot)$
- The acceptance probability is

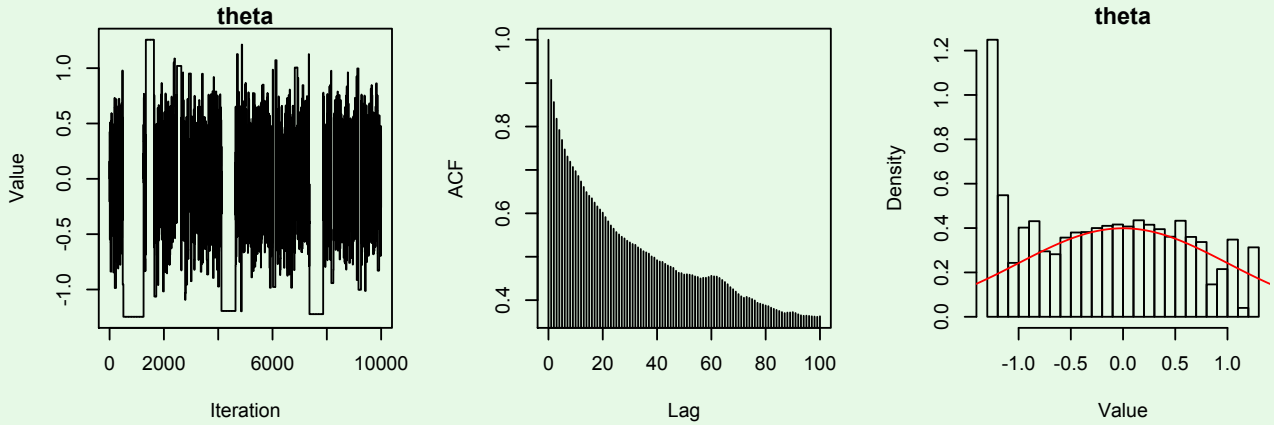
$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{x})}{\pi(\boldsymbol{\theta}|\boldsymbol{x})} \Big/ \frac{f(\boldsymbol{\theta}^*)}{f(\boldsymbol{\theta})} \right\}$$

- A nice thing is that, when $f(\cdot)$ is close to $\pi(\cdot|\boldsymbol{x})$, the acceptance probability can be close to 1.
- Principle of choosing $f(\cdot)$: Simulation from $f(\cdot)$ can easily cover the support of posterior density.
- Don't use uniform density for $f(\cdot)$, because it takes values in a bounded region and some values of θ will never be sampled.
- Similarly, don't use underdispersed density for $f(\cdot)$, e.g. $N(0, \Sigma)$ with very small Σ.
- Therefore, use overdispersed density for $f(\cdot)$, and tune it so that the higher the acceptance probability, the better.

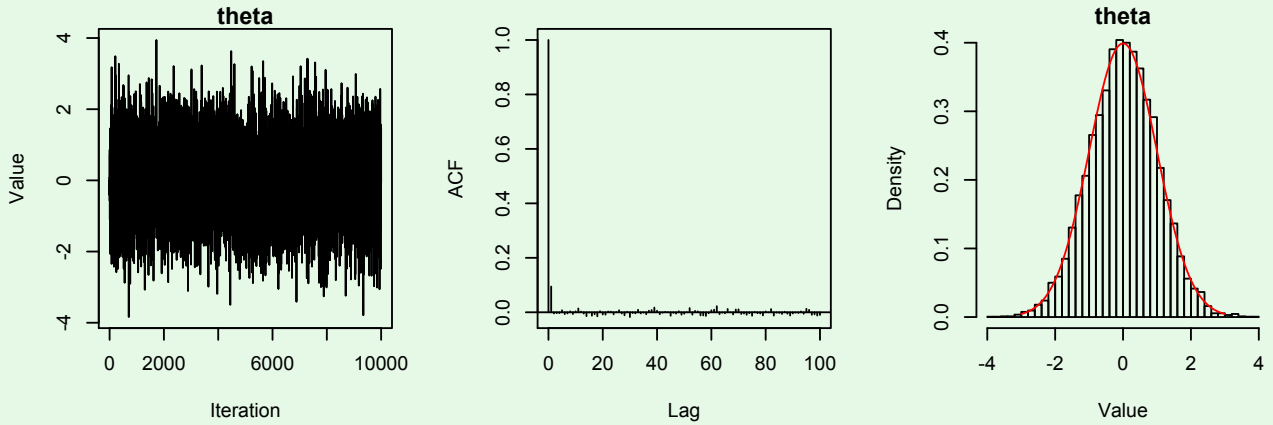
Sampling from a standard normal using normal independent chain

- Example: Consider the posterior is standard normal. Construct a Metropolis-Hastings algorithm using an independence chain with $f(\theta) = N(0, \Sigma)$.

  When $\Sigma = 0.1^2$, acceptance rate is 0.4.

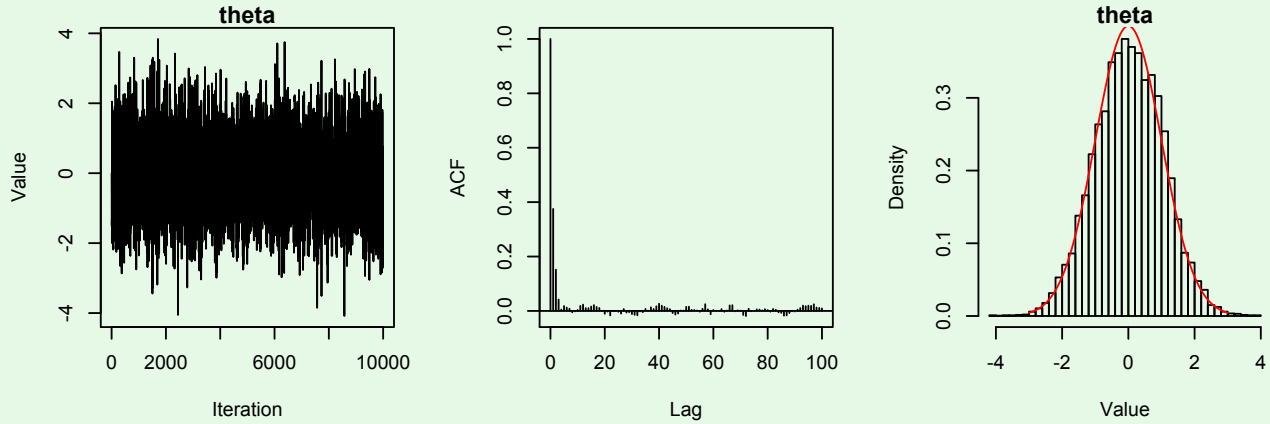


$\Sigma = 1.6^2$, acceptance rate is 0.85.

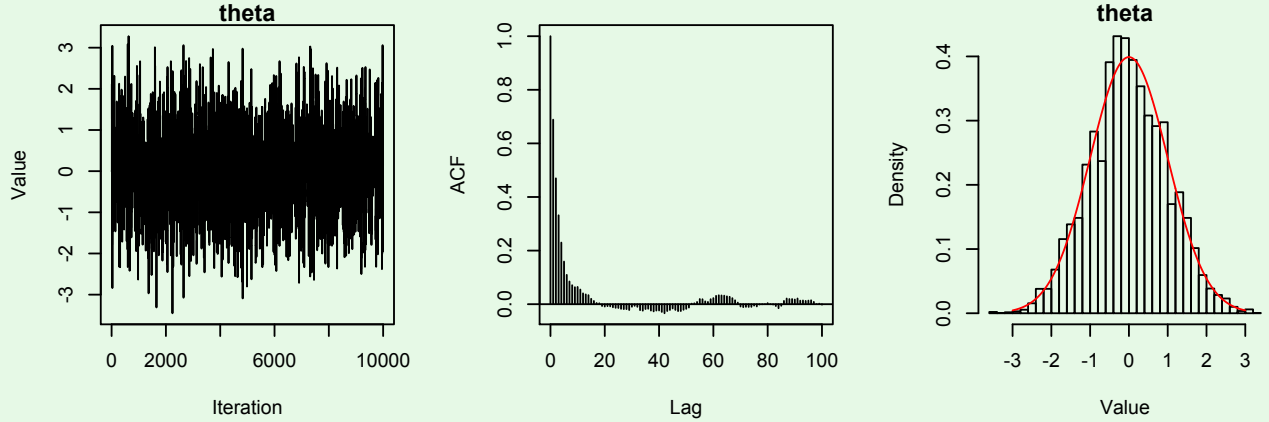


Sampling from a standard normal using normal independent chain

When $\Sigma = 6^2$, acceptance rate is 0.51.



$\Sigma = 30^2$, acceptance rate is 0.24.



- Choose $\Sigma = 1.6^2$ since the covered region is wide enough and the acceptance probability is the highest.
- Don't use underdispersed distribution, regardless its acceptance

Bayes Theorem via independence chains

- One possible choice for the proposal density is the prior density
- The acceptance probability is then

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left\{1, \frac{f(\boldsymbol{x}|\boldsymbol{\theta}^*)}{f(\boldsymbol{x}|\boldsymbol{\theta})}\right\},$$

  and hence depends only on the likelihood ratio of the proposal and the current value.
- It will not be underdispersed.

4.6 Hybrid methods

- The Metropolis-Hastings method in last section samples $\boldsymbol{\theta}$ in a single block, i.e. all parameters are updated at the same time.
- When the number of parameter is large, it is difficult to choose the proposal distribution. For example, when there are 5 parameters, the proposal distribution $N(0, \Sigma)$ has $5 \times 5 = 25$ tuning parameters in Σ.
- Recall the Gibbs sampler for how the componentwise update works.

4.6.1 Componentwise transitions

- Given a posterior distribution with FCDs that are awkward to sample from directly, we can define a Metropolis-Hastings scheme for each full conditional distribution, and apply them to each component in turn for each iteration
- This is like the Gibbs sampler, but each component update is a Metropolis-Hastings update, rather than a direct simulation from the FCD
- Each of these steps will require its own proposal distribution

Example

Componentwise Metropolist-Hasting algorithm in two-dimension.

Solution

. . .

The algorithm

1. Initialise the iteration counter to $j = 1$
   Initialise the state of the chain to $\boldsymbol{\theta}^{(0)} = (\theta_1^{(0)}, \ldots, \theta_p^{(0)})^T$
2. Let $\boldsymbol{\theta}_{-i}^{(j)} = \left(\theta_1^{(j)}, \ldots, \theta_{i-1}^{(j)}, \theta_{i+1}^{(j-1)}, \ldots, \theta_p^{(j-1)}\right)^T$, $i = 1, 2, \ldots, p$
   Obtain a new value $\boldsymbol{\theta}^{(j)}$ from $\boldsymbol{\theta}^{(j-1)}$ by successive generation of values
   - $\theta_1^{(j)} \sim \pi(\theta_1 | \boldsymbol{\theta}_{-1}^{(j)}, \boldsymbol{x})$ using a Metropolis–Hastings step with proposal distribution $q_1(\theta_1 | \theta_1^{(j-1)}, \boldsymbol{\theta}_{-1}^{(j)})$
   - $\theta_2^{(j)} \sim \pi(\theta_2 | \boldsymbol{\theta}_{-2}^{(j)}, \boldsymbol{x})$ using a Metropolis–Hastings step with proposal distribution $q_2(\theta_2 | \theta_2^{(j-1)}, \boldsymbol{\theta}_{-2}^{(j)})$

     $\vdots$
   - $\theta_p^{(j)} \sim \pi(\theta_p | \boldsymbol{\theta}_{-p}^{(j)}, \boldsymbol{x})$ using a Metropolis–Hastings step with proposal distribution $q_p(\theta_p | \theta_p^{(j-1)}, \boldsymbol{\theta}_{-p}^{(j)})$
3. Change counter j to $j + 1$, and return to step 2

The distributions $\pi(\theta_i | \boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})$ are just the FCDs

- Suppose we decide to use normal random walks for these M–H steps, that is, take $q_i(\theta_i^*|\theta_i, \boldsymbol{\theta}_{-i}^{(j)}) = N(\theta_i, \Sigma_{ij})$
- What is the appropriate value for Σ_{ij}?
- As the proposal in step j is targeting the conditional posterior density $\pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})$, the optimal choice of Σ_{ij} is

$$\Sigma_{ij} = \frac{2.38^2}{1} \, Var(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x}) = 2.38^2 \, Var(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})$$

- As these (conditional) posterior variances are not known before running the MCMC code, a sensible strategy might be to replace it with the (probably much larger) prior conditional variance or even the prior marginal variance, that is, use

$$\Sigma_{ij} = 2.38^2 \, Var(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}) \qquad \text{or} \qquad \Sigma_{ij} = 2.38^2 \, Var(\theta_i)$$

- Again, recall that these values are to be used as a guide, generally to get the order of magnitude for the innovation variance

Hybrid methods

- The Gibbs sampler can be used to sample from multivariate posterior distributions provided that we can simulate from the full conditional distributions (FCDs). The acceptance probability is 1.
- The Metropolis-Hastings method can be used to sample from awkward FCDs. The acceptance probability is less than 1.
- We can combine these, when some FCDs can be simulated from directly and some can not, to increase the acceptance probability.

4.6.2 Metropolis within Gibbs

- Given a posterior distribution with full conditional distributions,
  - some of which may be simulated from directly,
  - and others of which have Metropolis-Hastings updating schemes,

  the Metropolis within Gibbs algorithm goes through each in turn, and simulates directly from the full conditional, or carries out a Metropolis-Hastings update as necessary
- This algorithm is, in fact, just the above algorithm but uses the full conditional distributions as the proposal distributions when they are easy to simulate from
- To see this, suppose that we can simulate from the FCD $\pi(\theta_i|\boldsymbol{\theta}^{(j)}_{-i}, \boldsymbol{x})$ and use this as the proposal distribution, that is, take $\theta^*_i \sim \pi(\theta_i|\boldsymbol{\theta}^{(j)}_{-i}, \boldsymbol{x})$

- Then the acceptance probability for this step is

$$\begin{aligned}
\alpha(\theta_i, \theta_i^*) &= \min\left\{1, \frac{\pi(\theta_i^*|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})}{\pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})} \frac{q(\theta_i|\theta_i^*, \boldsymbol{\theta}_{-i}^{(j)})}{q(\theta_i^*|\theta_i, \boldsymbol{\theta}_{-i}^{(j)})}\right\} \\
&= \min\left\{1, \frac{\pi(\theta_i^*|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})}{\pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})} \frac{\pi(\theta_i|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})}{\pi(\theta_i^*|\boldsymbol{\theta}_{-i}^{(j)}, \boldsymbol{x})}\right\} \\
&= \min\{1, 1\} \\
&= 1,
\end{aligned}$$

that is, we always accept the proposal from the FCD

Example 4.6

- Construct an MCMC scheme for the problem in Example 4.4 where we had a random sample of size n from a gamma $Ga(\alpha, \lambda)$ distribution and independent gamma $Ga(a, b)$ and $Ga(c, d)$ prior distributions for α and λ respectively
- Recall that the FCDs were

$$\pi(\alpha|\lambda, \boldsymbol{x}) \propto \frac{\alpha^{a-1} e^{(-b+n\log \bar{x}_g + n\log\lambda)\alpha}}{\Gamma(\alpha)^n}, \quad \alpha > 0$$

and

$$\pi(\lambda|\alpha, \boldsymbol{x}) \propto \lambda^{c+n\alpha-1}\, e^{-(d+n\bar{x})\lambda}, \quad \lambda > 0$$

Solution

. . .

R code

- Use the function `mwgGamma` in the `nclbayes` library
- Data: $n = 50$, $\bar{x} = 0.62$, $\bar{x}_g = 0.46$ and $s = 0.4$
- Prior: $a = 2$, $b = 1$, $c = 3$, $d = 1$
- Use a normal random walk proposal with variance $\Sigma_\alpha = 0.9^2$
- This gives a reasonable acceptance probability of 0.237

```
library(nclbayes)
posterior=mwgGamma(N=20000,initial=(xbar/s)^2,innov=0.9,
    priorparam=c(2,1,3,1),n=50,xbar=0.62,xgbar=0.46,show=TRUE)
mcmcAnalysis(posterior,rows=2,show=F)
posterior2=mcmcProcess(input=posterior,burnin=10,thin=20)
mcmcAnalysis(posterior2,rows=2)
```

- Using `burnin` $= 10$, `thin` $= 20$ produces (almost) un-autocorrelated posterior output

Analysis of MCMC output (all realisations)

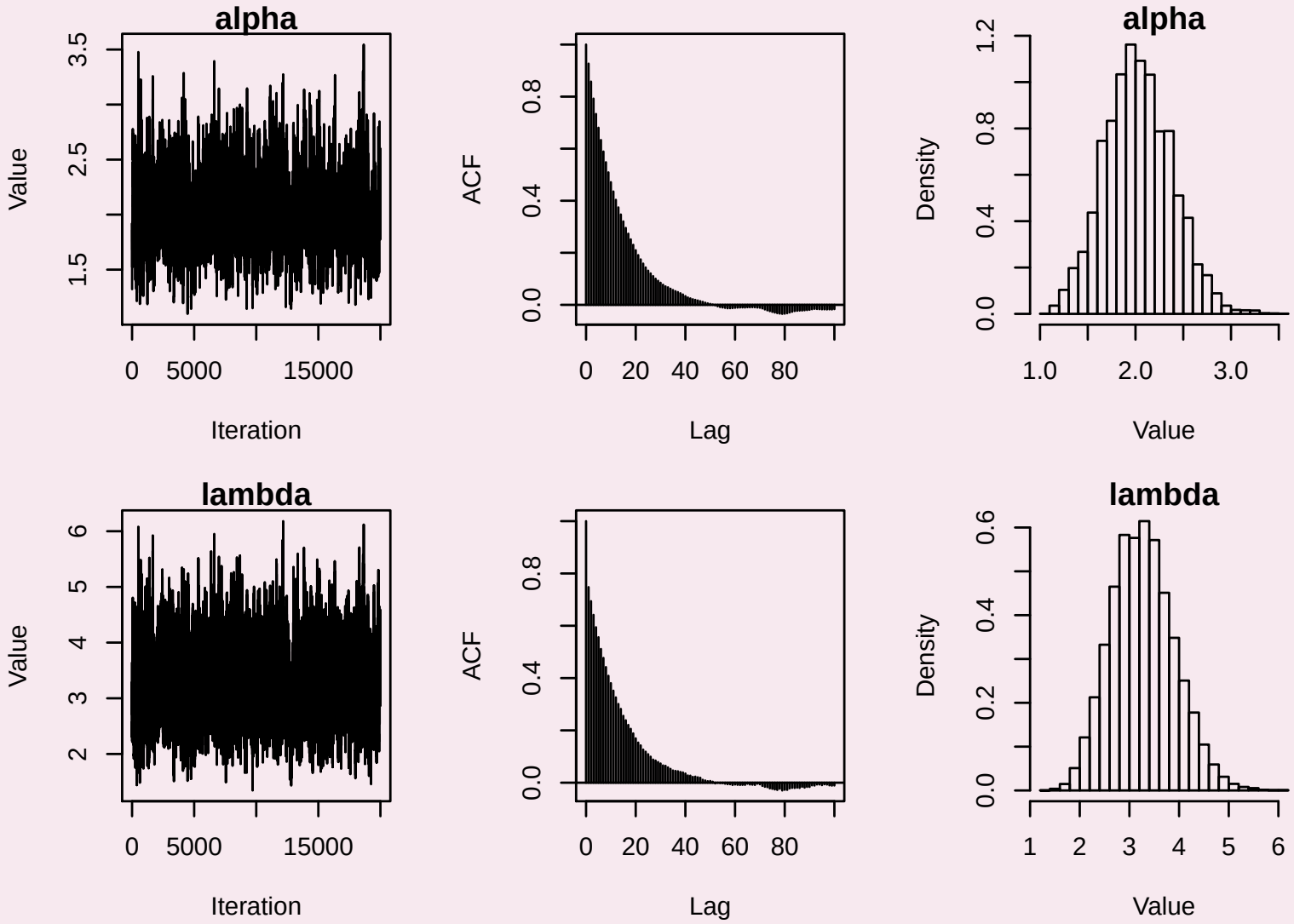


Figure: Trace plots, autocorrelation plots and histograms of the Metropolis within Gibbs sampler output

Analysis of MCMC output (with burnin = 10, thin = 20)

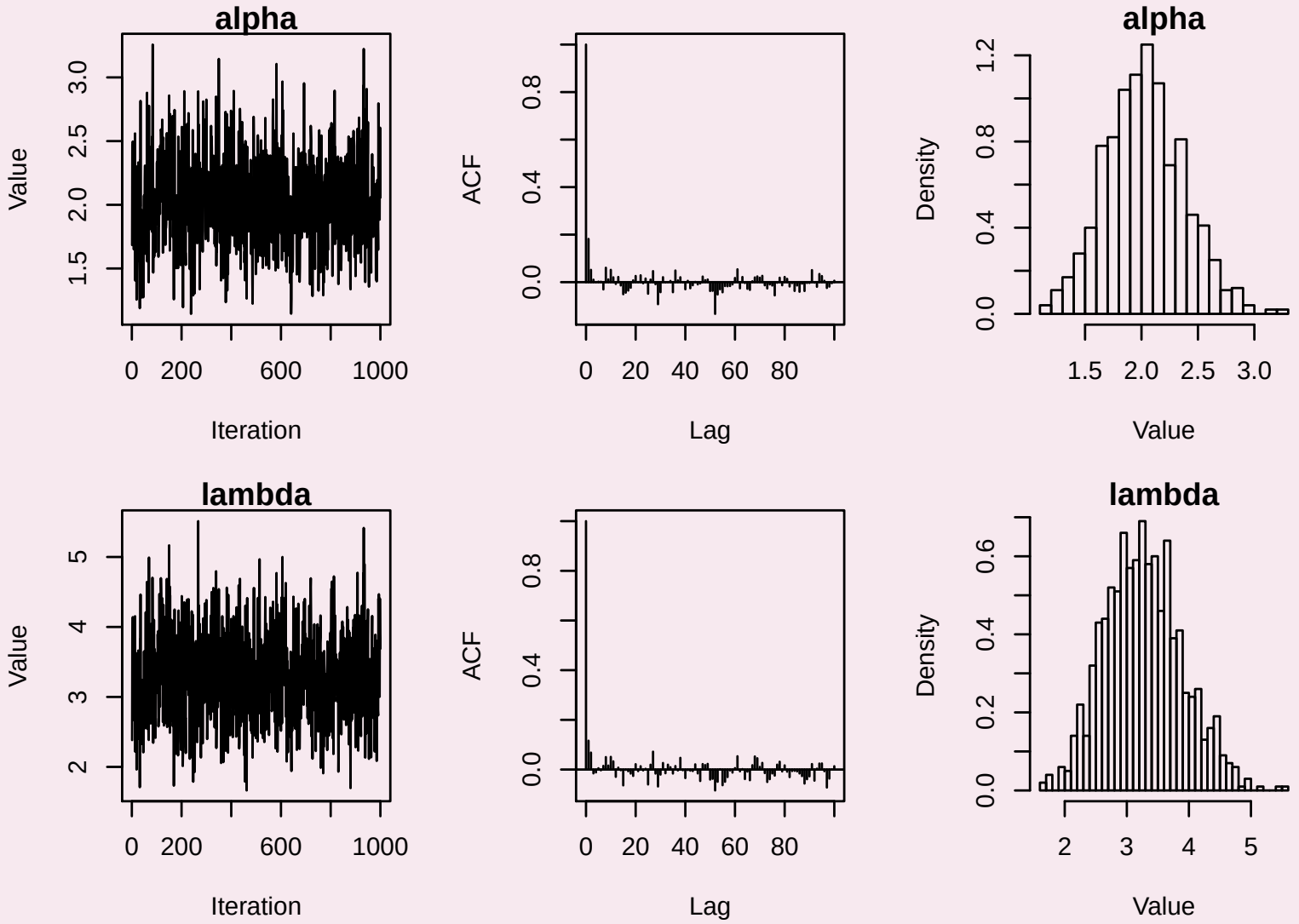


Figure: Trace plots, autocorrelation plots and histograms of the Metropolis within Gibbs sampler output

4.7 Summary

(i) Bayesian inference can be complicated when not using a conjugate prior distribution

(ii) One solution is to use Markov chain Monte Carlo (MCMC) methods

(iii) These work by producing realisations from the posterior distribution by constructing a Markov chain which has the posterior distribution as its stationary distribution

(iv) The MCMC methods we have studied are the Gibbs sampler, Metropolis within Gibbs algorithm and the Metropolis–Hastings algorithm

(v) When obtaining output from these algorithms, we need to assess whether there needs to be a burn-in and whether the output needs to be thinned (by looking at traceplots and autocorrelation plots) using `mcmcAnalysis` and `mcmcProcess`

(vi) The (converged and thinned) MCMC output are realisations from the posterior distribution. It can be used to

- obtain the posterior distribution for any (joint) functions of the parameters (such as $\sigma = 1/\sqrt{\tau}$ or $(\theta_1 = \mu - \tau, \theta_2 = e^{\mu+\tau/2})^T$)
- look at bivariate posterior distributions via scatter plots
- look at univariate marginal posterior distributions via histograms or boxplots
- obtain numerical summaries such as the mean, standard deviation and confidence intervals for single variables and correlations between variables

(vii) Equi-tailed posterior confidence intervals can be determined from the MCMC output using `mcmcCi`

4.8 Learning objectives

By the end of this chapter, you should be able to:

1. explain why not using a conjugate prior generally causes problems in determining the posterior distribution
2. describe the Gibbs sampler, explain why it is a Markov chain and give an outline as to why its stationary distribution is the posterior distribution
3. describe the issues of processing MCMC output (burn-in, autocorrelation, thinning etc.) and interpret numerical/graphical output
4. derive the full conditional densities for any posterior distribution and name these distributions if they are "standard" distributions given in the notes or on the exam paper

5. describe a Metropolis-Hastings algorithm in general terms and when using symmetric random walk proposals or independence proposals or general proposals
6. describe the hybrid methods: *componentwise transitions* and *Metropolis within Gibbs*
7. provide a detailed description of **any** of the MCMC algorithms as they apply to generating realisations from **any** posterior distribution