Part A Simulation and Statistical Programming HT14

Lecturer: Geoff Nicholls

University of Oxford

Lecture 12: MCMC, convergence and mixing

Notes and Problem sheets are available at

www.stats.ox.ac.uk\ ∼nicholls\PartASSP

Recall the Metropolis Hastings MCMC algorithm

MCMC targeting $p(x) = \tilde{p}(x)/Z_p$ using proposal $Y \sim q(y|x)$.

Let $X_t = x$. X_{t+1} is determined in the following way.

[1] Draw $y \sim q(\cdot|x)$ and $u \sim U[0,1]$.
[2] If

$$u \leq \alpha(y|x) \text{ where } \alpha(y|x) = \min\left\{1, \frac{\tilde{p}(y)q(x|y)}{\tilde{p}(x)q(y|x)}\right\}$$

set $X_{t+1} = y$, otherwise set $X_{t+1} = x$.

We initialise this with $X_0 = x_0, p(x_0) > 0$ and iterate for $t = 1, 2, 3, ...n$ to simulate the samples $X_0, X_1, X_2, ...X_n$ we need.

MCMC for the Normal distribution

Suppose want to simulate the standard normal distribution $X \sim N(0,1)$. The target density is

$$\tilde{p}(x) \propto \exp(-x^2/2).$$

Step 1: Choose the proposal distribution. We need something simple and irreducible. Fix a constant $a > 0$ and choose a new point uniformly at random in a window of length $2a$ centred at x. The proposal density is

$$q(y|x) = \frac{1}{2a}\mathbb{I}_{x-a<y<x+a}$$

Notice that $q(y|x) = q(x|y)$.

Step 2: give the MCMC algorithm. If $X_t = x$ then X_{t+1} is determined in the following way:

[1] Simulate $Y \sim U(x-a, x+a)$ and $U \sim U(0,1)$.

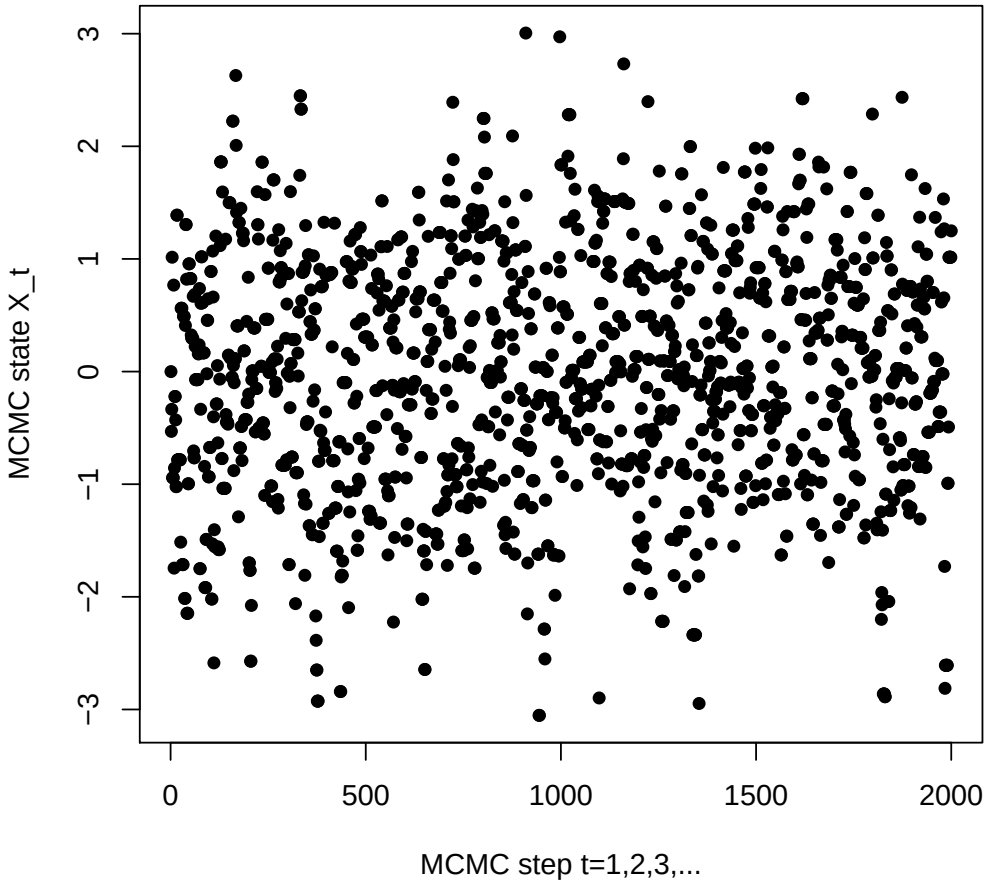
[2] If $U \leq \alpha(y|x)$ set $X_{t+1} = y$ and otherwise set $X_{t+1} = x$.

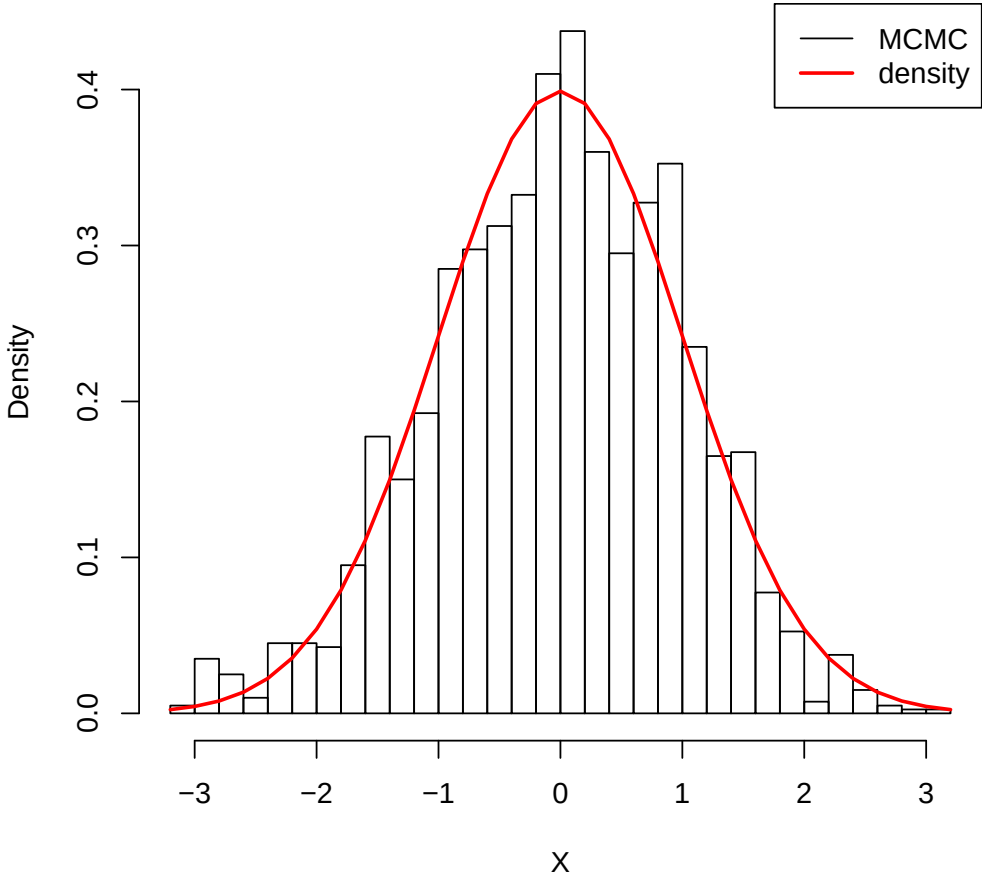
Here

$$\begin{aligned} \alpha(y|x) &= \min\left(1, \frac{\tilde{p}(y)q(x|y)}{\tilde{p}(x)q(y|x)}\right) \\ &= \min\left(1, \exp(-y^2/2 + x^2/2)\right). \end{aligned}$$

```
#MCMC simulate X_t ~ N(0,1)
a=3; n=2000
X=numeric(n); X[1]=0;
for (t in 1:(n-1)) {
   x<-X[t]
   y<-x+(2*runif(1)-1)*a
   if (runif(1)<exp((x^2-y^2)/2)) {
      X[t+1]<-y
   } else {
      X[t+1]<-x
   }
}
```

(see the associated R-file for plotting commands)





MH example: an equal mixture of bivariate normals

$$\pi(\theta) = (2\pi)^{-1}\left(0.5e^{-(\theta-\mu_1)\Sigma_1^{-1}(\theta-\mu_1)/2} + 0.5e^{-(\theta-\mu_2)\Sigma_2^{-1}(\theta-\mu_2)/2}\right)$$

with $\theta = (\theta_1, \theta_2)$. Use $\mu_1 = (1,1)^T$, $\mu_2 = (5,5)^T$ and $\Sigma_1 = \Sigma_2 = I_2$ for this illustration.

Step 1. For a proposal distribution q we want something simple to sample. The simplest thing I can think of is the same as before:

$$\theta_i' \sim U(\theta_i - a, \theta_i + a)$$

with a a fixed constant. Note that this time we are proposing in a box of side $2a$. That is easy to sample, and certainly $q(\theta'|\theta) > 0 \Leftrightarrow q(\theta|\theta') > 0$ since $q(\theta'|\theta) = q(\theta|\theta') = 1/4a^2$.

Step 2. The algorithm is, given $\theta^{(n)} = \theta$,
[1] for $i = 1, 2$ simulate $\theta_i' \sim U(\theta_i - a, \theta_i + a)$
[2] with probability

$$\alpha(\theta'|\theta) = \min\left\{1, \ \frac{\pi(\theta')}{\pi(\theta)}\right\}$$

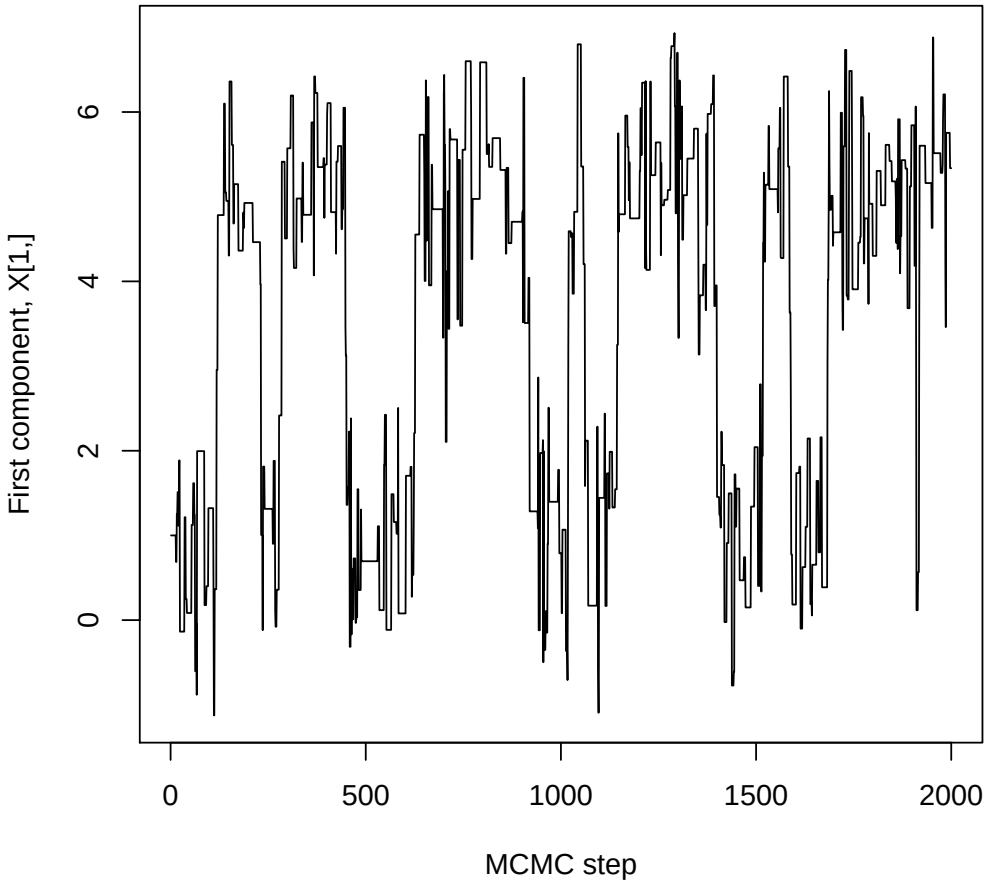
set $\theta^{(n+1)} = \theta'$ otherwise set $\theta^{(n+1)} = \theta$.

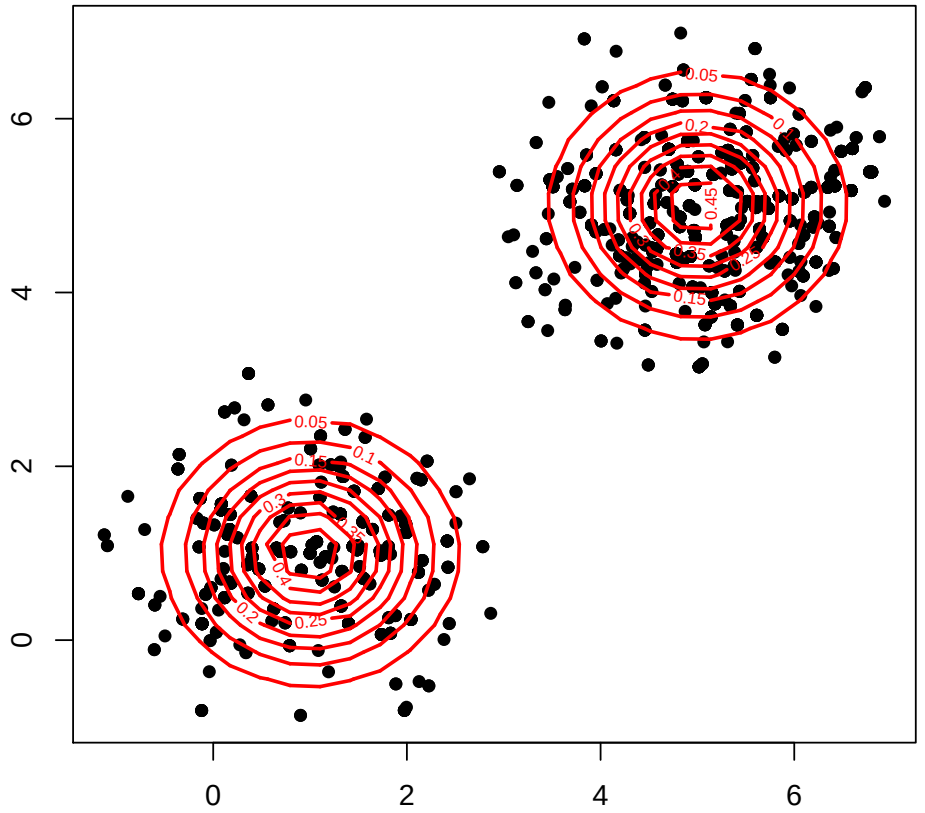
This algorithm is ergodic for any $a > 0$ but we will see that the choice of a makes a difference to efficiency.

```
a=3; n=2000
mu1=c(1,1); mu2=c(5,5); S=diag(2); S1i=S2i=solve(S);
X=matrix(NA,2,n); X[,1]=x=mu1
for (t in 1:(n-1)) {
   y<-x+(2*runif(2)-1)*a
   MHR<-f(y,mu1,mu2,S1i,S2i)/f(x,mu1,mu2,S1i,S2i)
   if (runif(1)<MHR) x<-y
   X[,t+1]<-x
}

#MCMC simulate X_t according to a mixture of normals
f<-function(x,mu1,mu2,S1i,S2i,p1=0.5) {
  #mixture of normals, density up to constant factor
  c1<-exp(-t(x-mu1)%*%S1i%*%(x-mu1))
  c2<-exp(-t(x-mu2)%*%S2i%*%(x-mu2))
  return(p1*c1+(1-p1)*c2)
}
```

(see the associated R-file for plotting commands)





Convergence and mixing

We want to estimate $E_p(f(X))$ using our MCMC samples $X_0, X_1, X_2, ..., X_n$ targeting $p(x)$ and calculate the estimate $\bar{f}_n = n^{-1} \sum_t f(X_t)$. The ergodic theorem tells us this estimate converges in probability to $E_p(f(X))$.

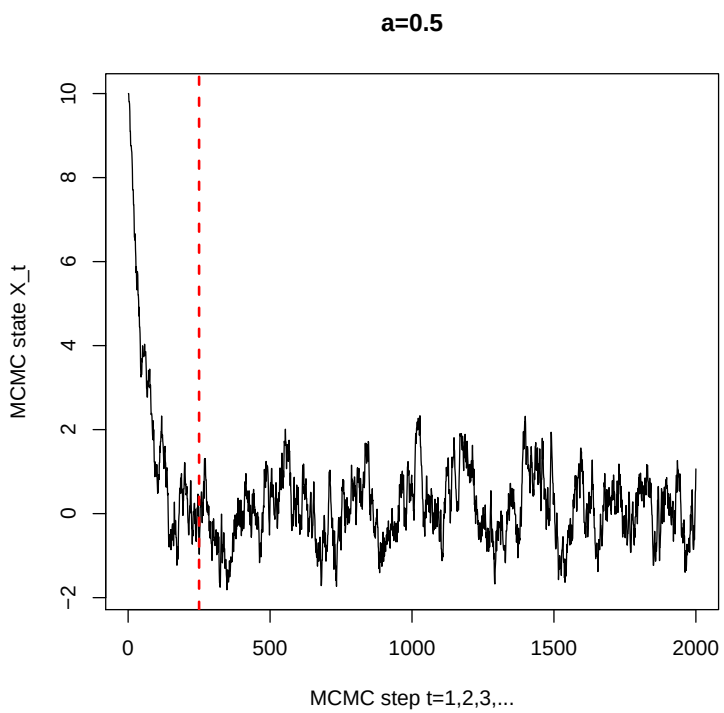
How large should we take n? There are two issues.

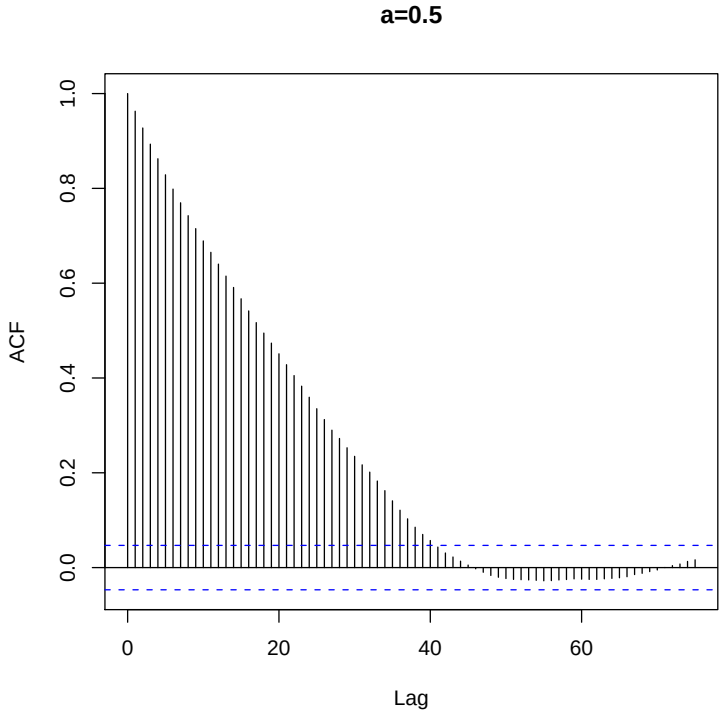
First, suppose $p^{(0)}(x) = p(x)$, so we start the chain in equilibrium. The variance, $\text{var}(\bar{f}_n)$, of $\bar{f}_n$ will get smaller as n increases. We should choose n large enough to ensure $\text{var}(\bar{f}_n)$ is small enough so that $\bar{f}_n$ has useful precision. However, calculating $\text{var}(\bar{f}_n)$ wont be completely straightforward as the MCMC samples are correlated.

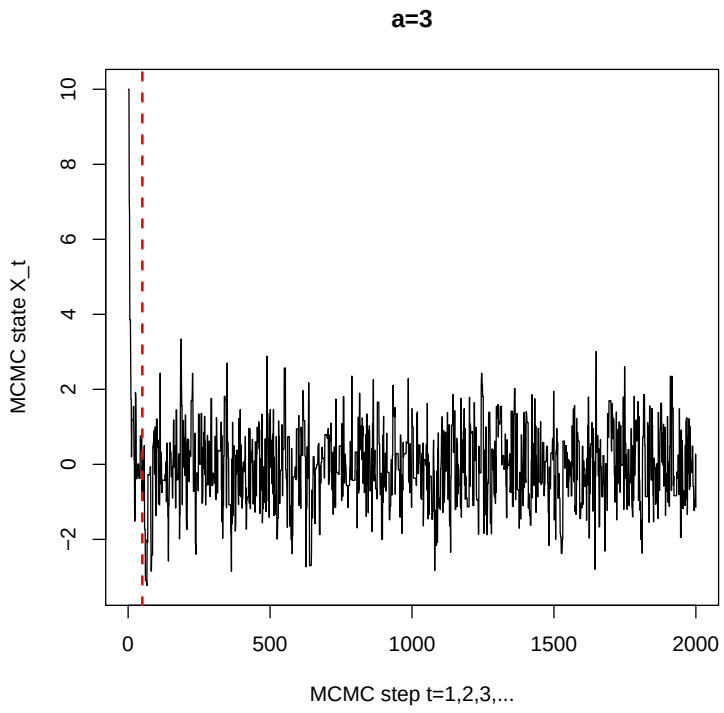
Second, we dont start the chain in equilibrium. The samples in the first part of the chain are biased by the initialization. It is common practice to drop the first part of the MCMC run (called "burn-in") to reduce the initialization bias. We know $p^{(t)} \longrightarrow p$ as $\longrightarrow \infty$ and want to choose a cut-off T beyond which $p^{(t)} \simeq p$ to a good approximation. We need $n \gg T$ so that most of the samples are representative of p.

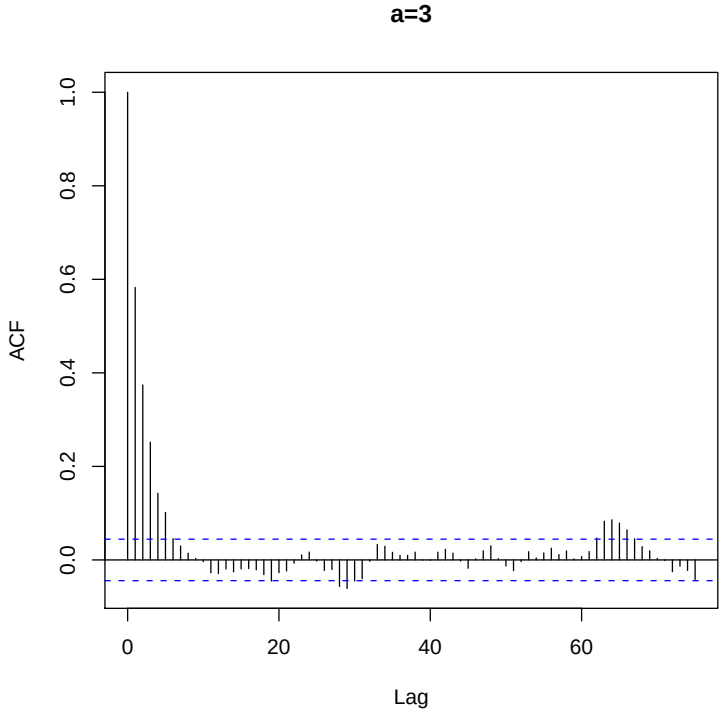
Note that if $n \gg T$ then the bias from burn-in will be slight anyway. One observation here is that if you need to drop states from the start of the chain to reduce this bias, you probably havnt run the chain long enough.

The following figures show autocorrelations for two MCMC runs of the N(0,1) sampler above, with different values of the jump size $a = 0.5, 3$.


a=0.5
MCMC state X_t
MCMC step t=1,2,3,...
−2
0
2
4
6
8
10
0
500
1000
1500
2000


a=0.5
ACF
Lag
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60


a=3
MCMC state X_t
MCMC step t=1,2,3,...
−2
0
2
4
6
8
10
0
500
1000
1500
2000


a=3
ACF
Lag
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60

MCMC variance in equilibrium

$X_0, X_1, X_2, \ldots$ are correlated so $\mathrm{var}(\bar{f}_n) \neq \mathrm{var}(f(X))/n$ in general.

Correlation at lag s

$$\rho_s^{(f)} = \frac{\mathrm{cov}(f(X_i), f(X_{i+s}))}{\mathrm{var}(f(X_i))}$$

(so $\rho_0 = 1$). Let $\sigma^2 = \mathrm{var}(f(X_i))$. This doesnt depend on i because the chain is stationary, because it was started in equilibrium.

Express $\mathrm{var}(\bar{f}_n)$ in terms of $\rho_s^{(f)}$. This gives insight and leads to an estimator for $\mathrm{var}(\bar{f}_n)$, since we can estimate $\rho_s^{(f)}$.

$$
\begin{aligned}
\mathrm{var}(\overline{f}) &= n^{-2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{cov}(f(X_i), f(X_j)) \\
&= \sigma^2 n^{-2} \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{|i-j|} \\
&= \sigma^2 n^{-1} \left[1 + 2 \sum_{s=1}^{n-1} \left(1 - \frac{s}{n} \right) \rho_s \right] \\
&\simeq \sigma^2 n^{-1} \left[1 + 2 \sum_{s=1}^{n-1} \rho_s \right] \\
&= \sigma^2 \tau_f / n,
\end{aligned}
$$

if as usual ρ_s is small when s is large. τ_f is called the integrated autocorrelation time. The quantity ESS$= n/\tau_f$ is called the effective sample size - the number of independent samples that would give the same precision for $\overline{f}$ as the n correlated samples we actually have.

MCMC convergence

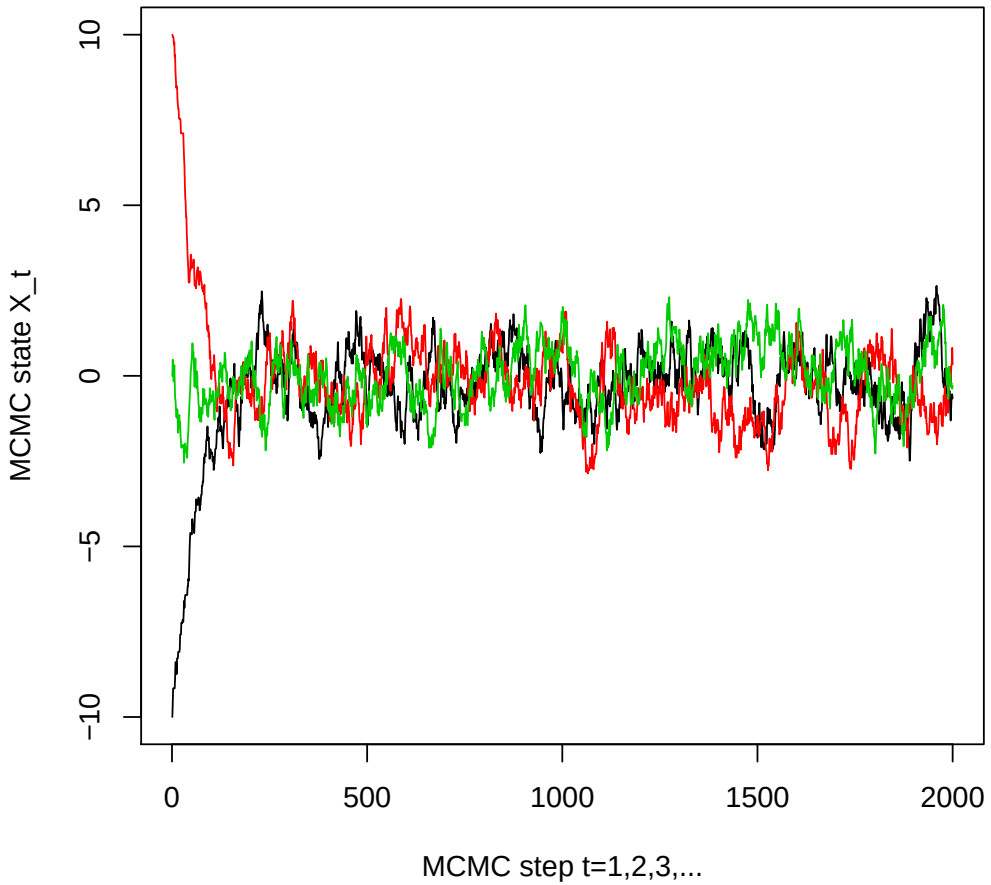
There is no simple generic sufficient condition we can test for convergence. Here some checks we can run to detect poor mixing and identify a burn-in and run length.

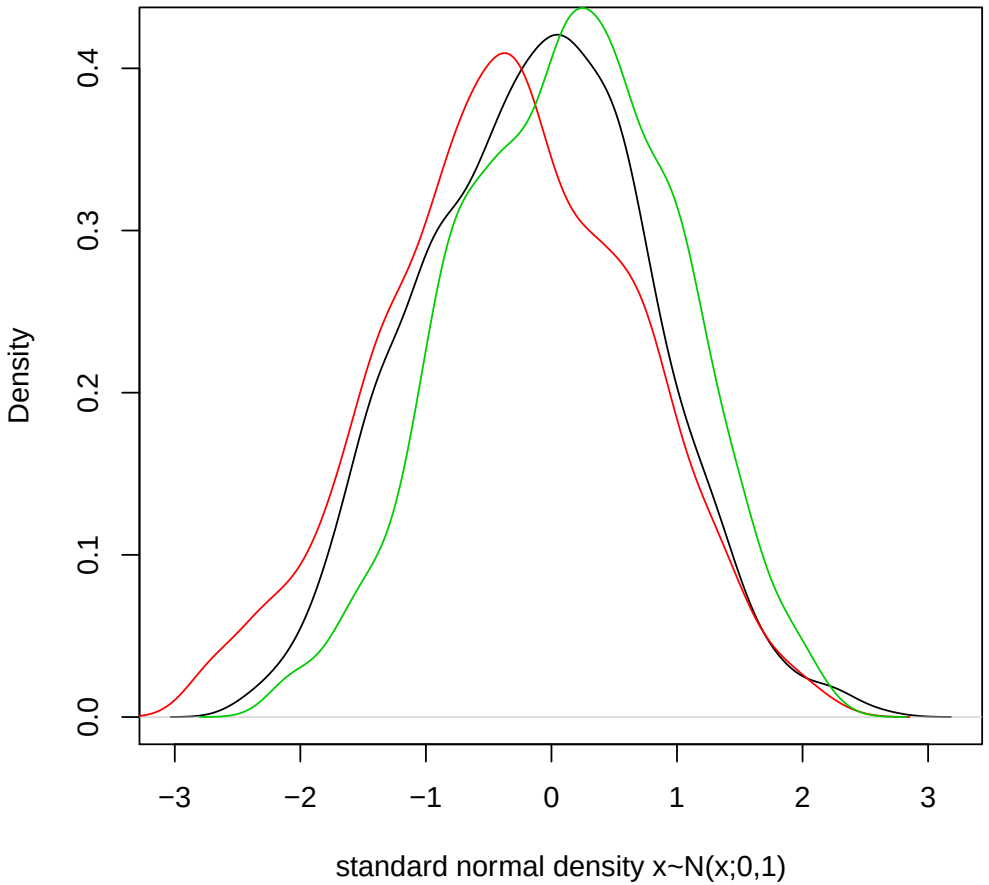
[1] Make multiple runs from different start states and check marginal distributions agree.

[2] Calculate the ESS and check it is reasonably large.

[3] Plot MCMC traces of the variables and key functions. The chain should be stationary after burn-in.

Here is an example of the plots I would use for convergence checking on the $N(x; 0, 1)$ MCMC sampler.


MCMC state X_t
10
5
0
−5
−10
0
500
1000
1500
2000
MCMC step t=1,2,3,...


Density
0.4
0.3
0.2
0.1
0.0
−3
−2
−1
0
1
2
3
standard normal density x~N(x;0,1)