MCMC - Gibbs Sampling, Metropolis-Hastings Sampling

Teodor Chakarov

2024-01-10

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(mvtnorm)
```

```
## Warning: package 'mvtnorm' was built under R version 4.2.3
```

```
library(mcmcplots)
```

```
## Warning: package 'mcmcplots' was built under R version 4.2.3
```

```
## Loading required package: coda
```

```
## Warning: package 'coda' was built under R version 4.2.3
```

```
set.seed(12141198)
```

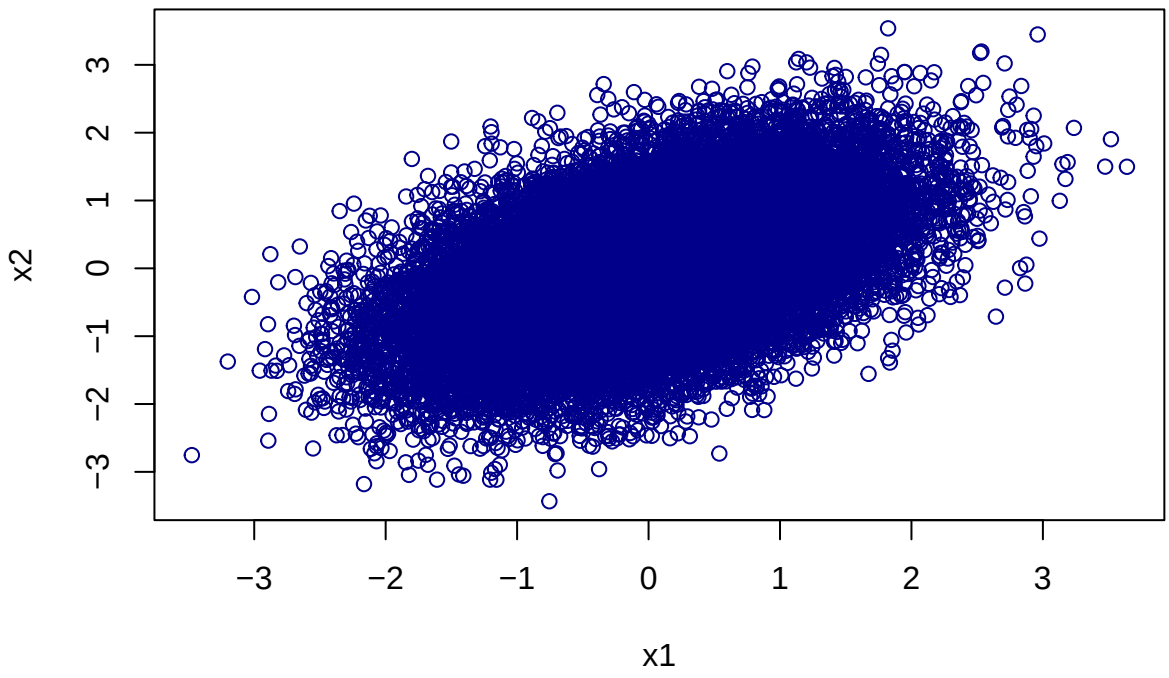
Set up the experimental parameters:

```
p = 0.5
M = 30000
```

Implement a Gibbs sampler to sample from this distribution

The Gibbs sampler is an algorithm from Markov chain Monte Carlo framework, designed for generating samples from a multivariate probability distribution. It is used for estimating the posterior distribution of various interconnected variables. In its operation, the algorithm samples from the conditional distributions of each variable, holding the values of the other variables constant. This process is being repeated iteratively.

```
x1 <- c(rnorm(1, 0, 1))
x2 <- c(rnorm(1, 0, 1))

# generate the rest of the values
for (i in 2:M){
  x1 <- c(x1, rnorm(1, x2[i-1] * p, 1-p**2))
  x2 <- c(x2, rnorm(1, x1[i] * p, 1-p**2))
}
plot(x1,x2, col="blue4")
```



Use the Metropolis-Hastings algorithm with block-wise update to simulate from this distribution

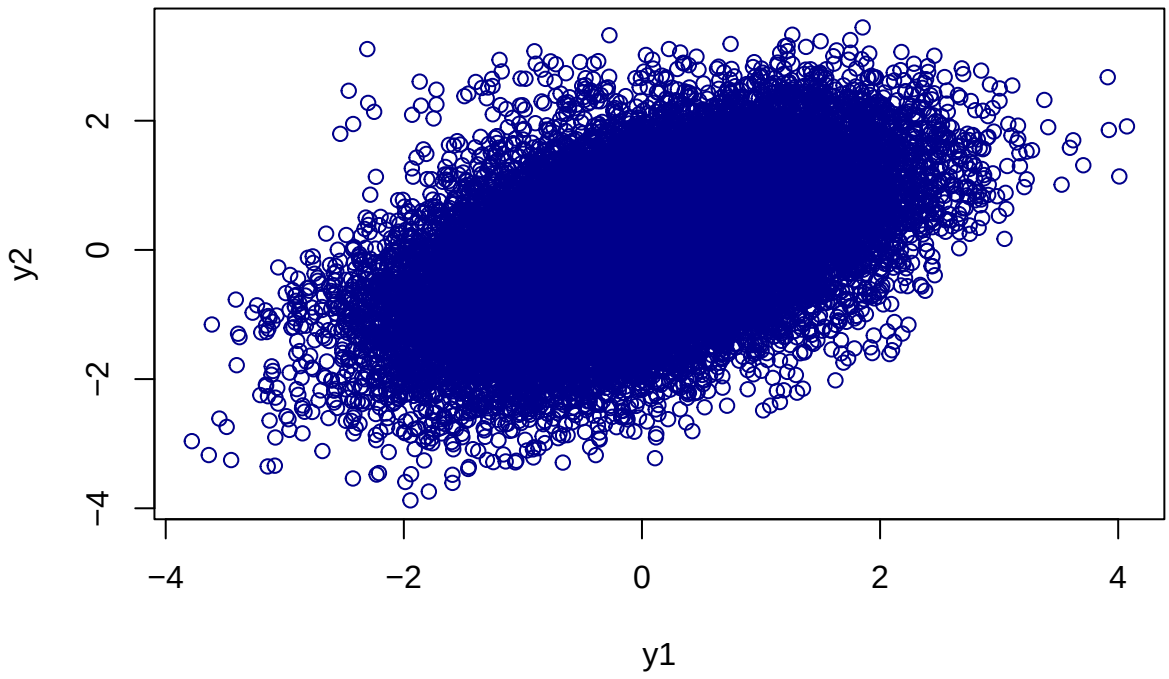
The Metropolis-Hastings algorithm is another variant within the Markov Chain Monte Carlo methodology. It is used for sampling from a specified target distribution. This algorithm functions by proposing potential values, which are subsequently accepted based on a defined probability. The sequence produced by the algorithm ultimately converges to represent the target distribution.

```
y1 <- c(rnorm(1, 0, 1))
y2 <- c(rnorm(1, 0, 1))

distrib <- dmvnorm(c(y1, y2),mean=c(0, 0),sigma = matrix(c(1, p, p, 1),ncol=2))
distrib
```

```
## [1] 0.0457918
```

```
#set param
i <- 2
while (i <= M){
  y1_selected <- rnorm(1, y1[i-1], 0.5)
  y2_selected <- rnorm(1, y2[i-1], 0.5)

  # define the proposed bivariate normal distribution
```

```
  distrib_selected <- dmvnorm(c(y1_selected,y2_selected),sigma = matrix(c(1,p,p,1),ncol=2))
  ratio <- distrib_selected / distrib

  # compute acceptence likelihood and randomly generate acceptance threshhold
  alpha <- min(1,ratio)
  thresh <- runif(1)
  if (thresh <= alpha){
    y1[i] <- y1_selected
    y2[i] <- y2_selected
    distrib <- distrib_selected
    i <- i + 1
    }
}

plot(y1,y2,col="blue4")
```

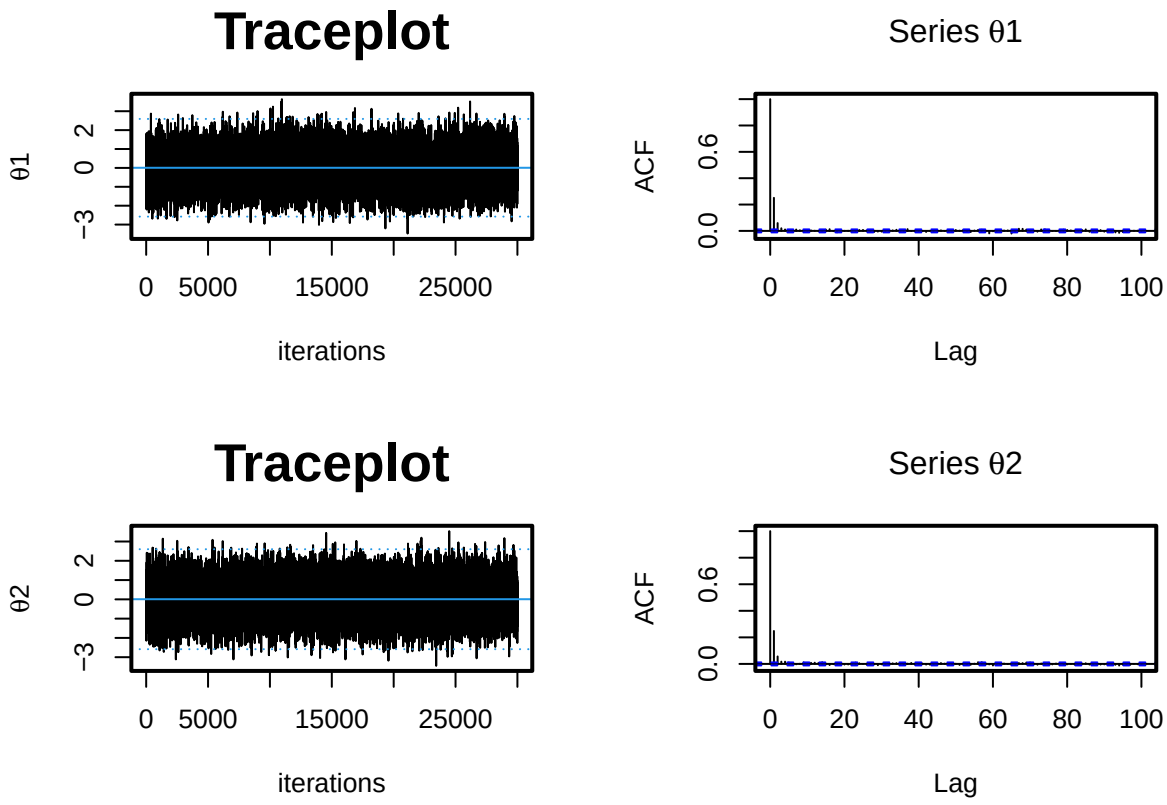


Perform chain diagnostics on the resulting chain

Lets perform a diagnostics of the given sampling methods with help form the code given on slide 44 from the lecture materials.

Gibbs sampler:

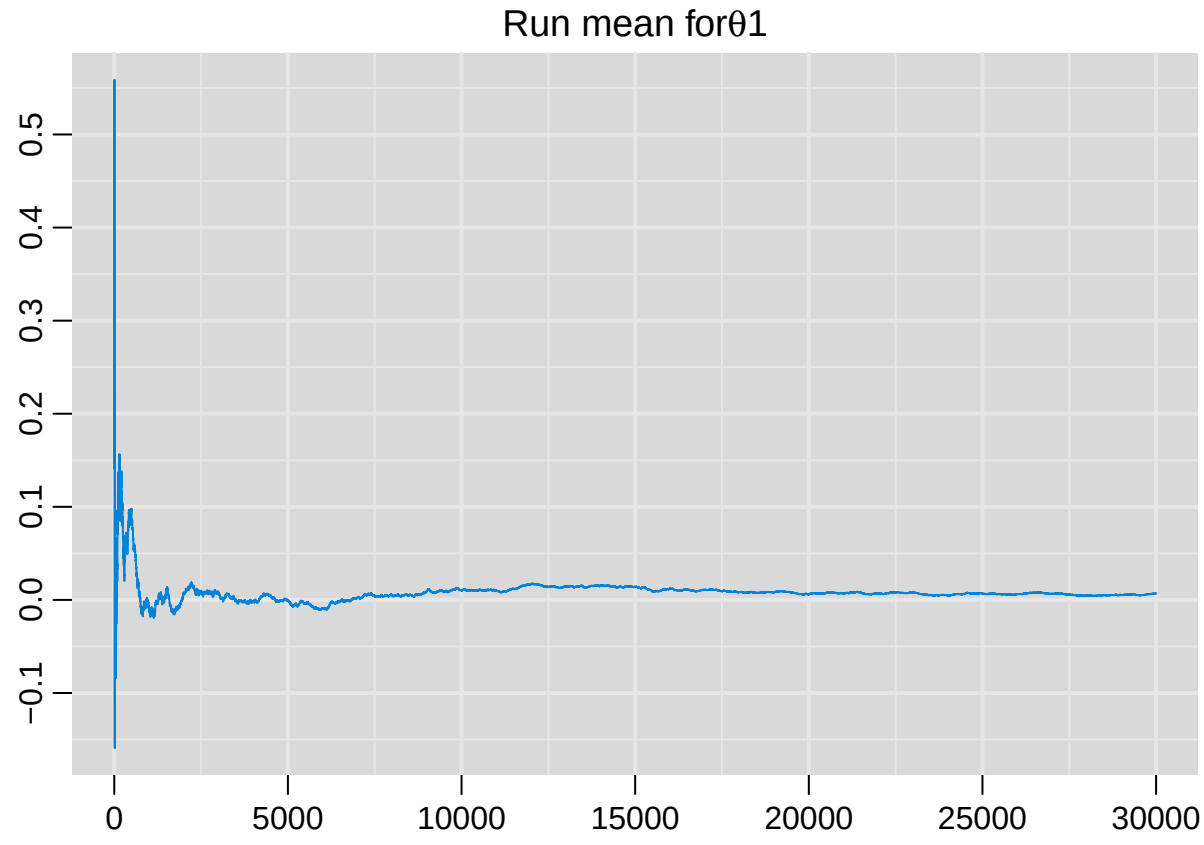
```
t1 <- x1
t2 <- x2

# trace plots
par(mfrow=c(2,2),mar=c(5,5,4,2))
plot(1:M,t1,type="l",ylim=c(min(t1,mean(t1)-3*sd(t1)),
max(t1,mean(t1)+3*sd(t1))),xlab="iterations",
ylab=expression(paste(theta,"1")),cex.main=2,main="Traceplot")
abline(h=mean(t1),lty=1,col=4);
abline(h=mean(t1)+3*sd(t1),lty=3,col=4)
abline(h=mean(t1)-3*sd(t1),lty=3,col=4); box(lwd=2)
# ACF plots with run mean
acf(t1,lag.max=100,main=expression(paste("Series ",theta,"1")),cex.main=2); box(lwd=2)
plot(1:M,t2,type="l",ylim=c(min(t2,mean(t2[-(1:200)])-3*sd(t2[-(1:200)])),
max(t2,mean(t2[-(1:200)])+3*sd(t2[-(1:200)]))),xlab="iterations",
ylab=expression(paste(theta,"2")),cex.main=2,main="Traceplot")
abline(h=mean(t2[-(1:200)]),lty=1,col=4)
abline(h=mean(t2[-(1:200)])+3*sd(t2[-(1:200)]),lty=3,col=4)
abline(h=mean(t2[-(1:200)])-3*sd(t2[-(1:200)]),lty=3,col=4); box(lwd=2)
acf(t2,lag.max=100,main=expression(paste("Series ",theta,"2")),cex.main=2); box(lwd=2)
```




```
# histrograms of marginal posteriors
rmeanplot(t1,lwd=2,main=expression(paste("Run mean for",theta,"1")),mar=c(2,2,1.5,1)+ 0.1)
```

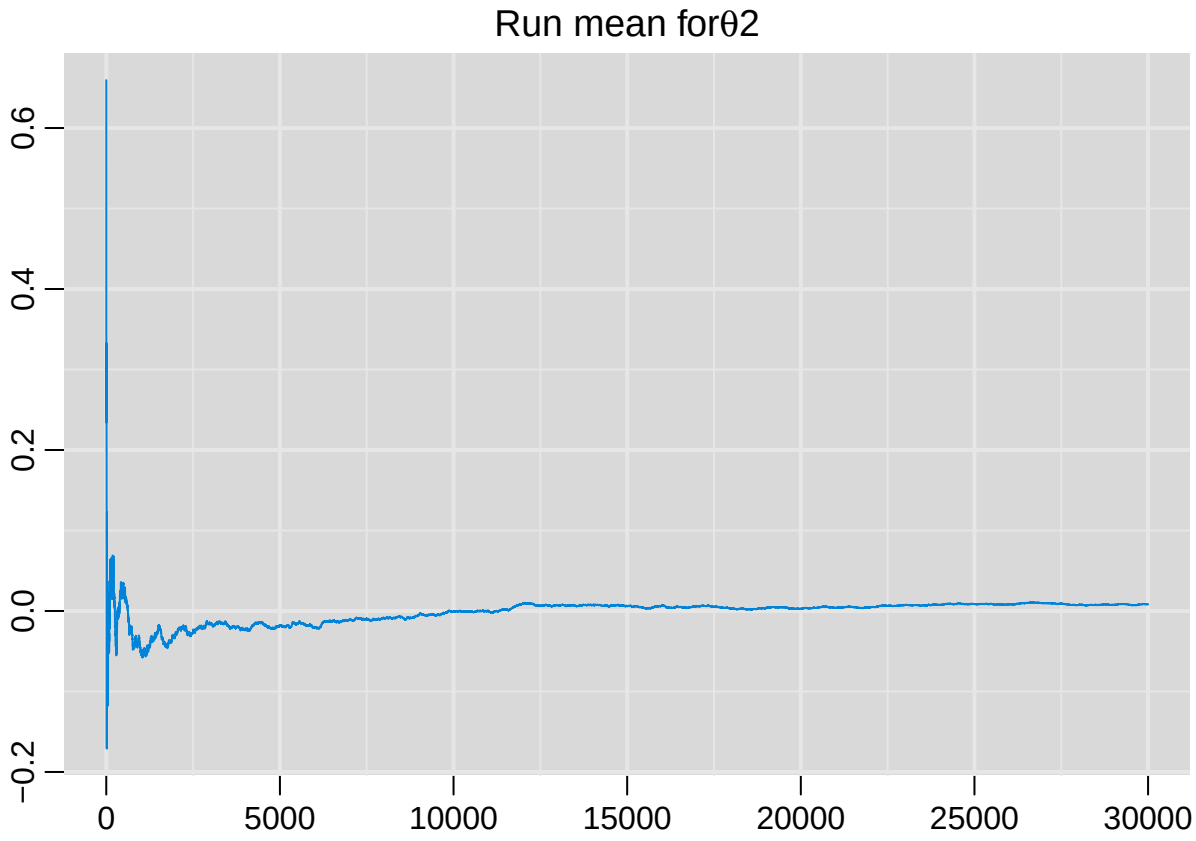
```
## Warning in rmeanplot(t1, lwd = 2, main = expression(paste("Run mean for", :
```

```
## Argument 'mcmcout' did not have valid variable names, so names have been
## created for you.
```

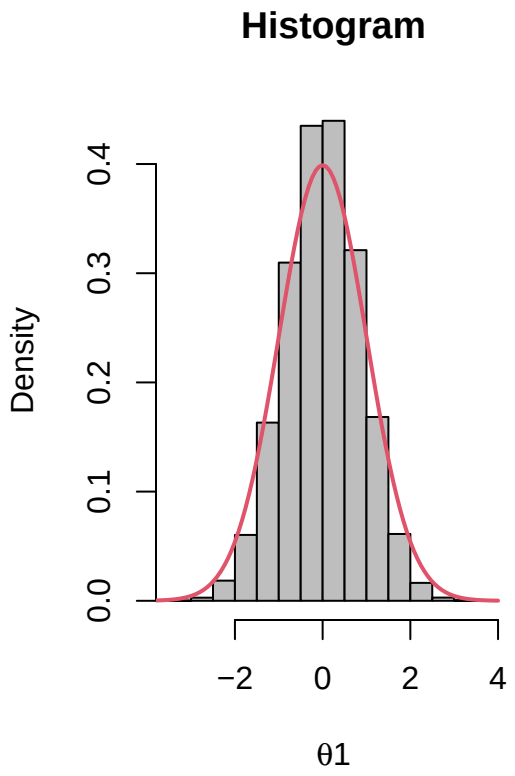


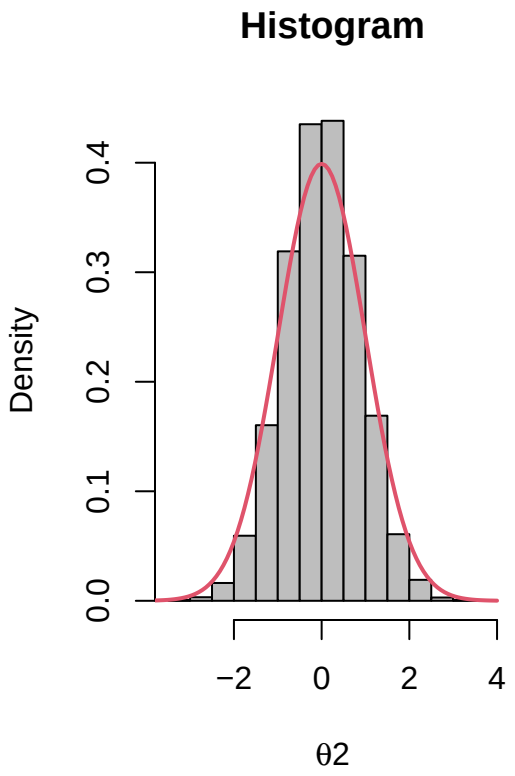

```
rmeanplot(t2,lwd=2,main=expression(paste("Run mean for",theta,"2")),mar=c(2,2,1.5,1)+ 0.1)
```

```
## Warning in rmeanplot(t2, lwd = 2, main = expression(paste("Run mean for", :
## Argument 'mcmcout' did not have valid variable names, so names have been
## created for you.
```



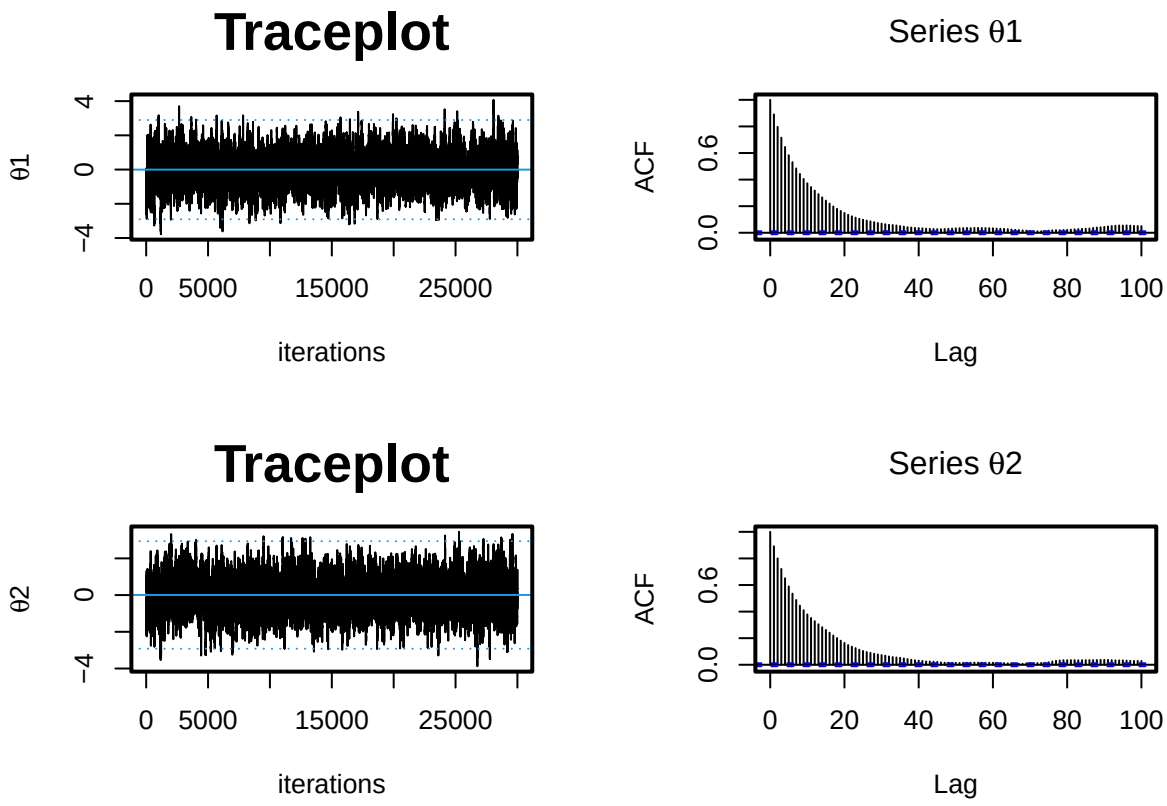

```
par(mfrow=c(1,2),mar=c(5,5,4,2))
hist(t1,freq=F,col="grey",main="Histogram",xlab=expression(paste(theta,"1")))
curve(dnorm(x,0,1),from=-4,4,add=T,lwd=2,col=2)
hist(t2,freq=F,col="grey",main="Histogram",xlab=expression(paste(theta,"2")))
curve(dnorm(x,0,1),from=-4,4,add=T,lwd=2,col=2)
```





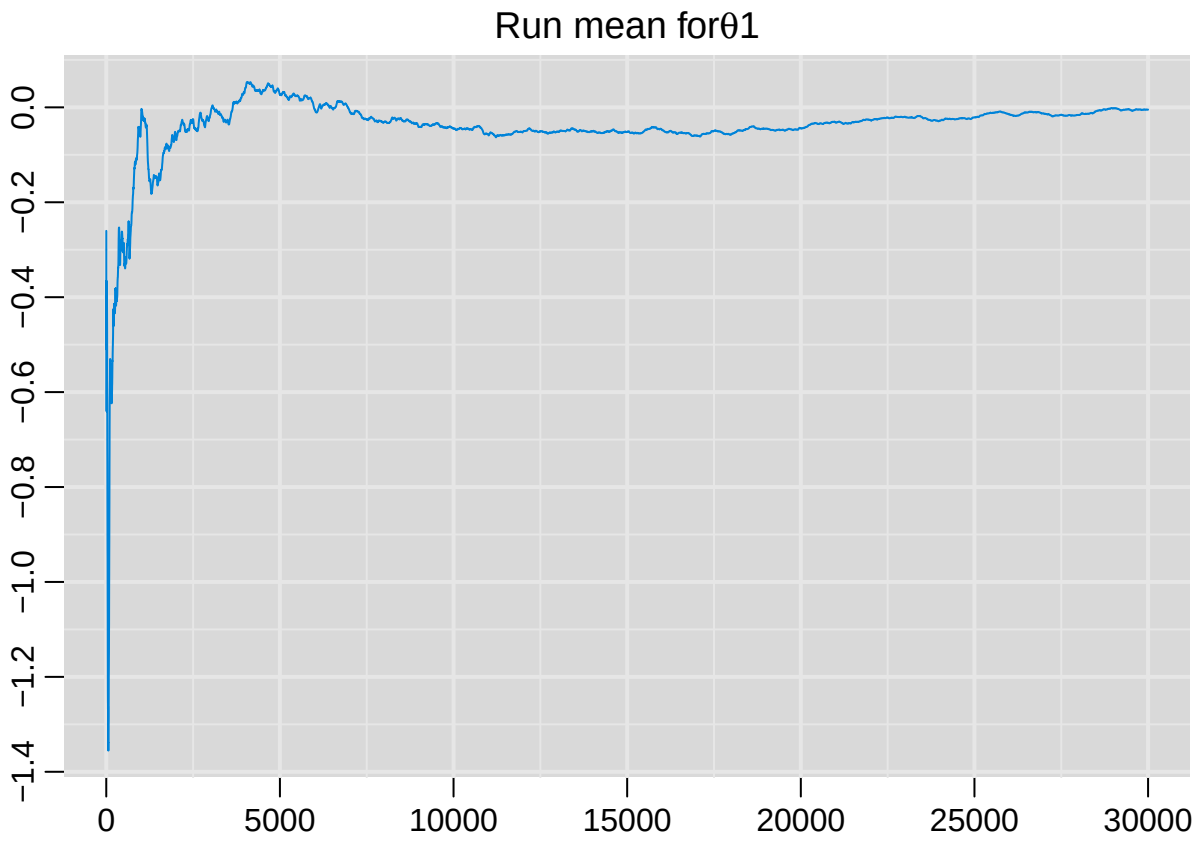
Metropolis-Hastings sampler

```
t1 <- y1
t2 <- y2

# trace plots
par(mfrow=c(2,2),mar=c(5,5,4,2))
plot(1:M,t1,type="l",ylim=c(min(t1,mean(t1)-3*sd(t1)),
max(t1,mean(t1)+3*sd(t1))),xlab="iterations",
ylab=expression(paste(theta,"1")),cex.main=2,main="Traceplot")
abline(h=mean(t1),lty=1,col=4);
abline(h=mean(t1)+3*sd(t1),lty=3,col=4)
abline(h=mean(t1)-3*sd(t1),lty=3,col=4); box(lwd=2)
# ACF plots with run mean
acf(t1,lag.max=100,main=expression(paste("Series ",theta,"1")),cex.main=2); box(lwd=2)
plot(1:M,t2,type="l",ylim=c(min(t2,mean(t2[-(1:200)])-3*sd(t2[-(1:200)])),
max(t2,mean(t2[-(1:200)])+3*sd(t2[-(1:200)]))),xlab="iterations",
ylab=expression(paste(theta,"2")),cex.main=2,main="Traceplot")
abline(h=mean(t2[-(1:200)]),lty=1,col=4)
abline(h=mean(t2[-(1:200)])+3*sd(t2[-(1:200)]),lty=3,col=4)
abline(h=mean(t2[-(1:200)])-3*sd(t2[-(1:200)]),lty=3,col=4); box(lwd=2)
acf(t2,lag.max=100,main=expression(paste("Series ",theta,"2")),cex.main=2); box(lwd=2)
```



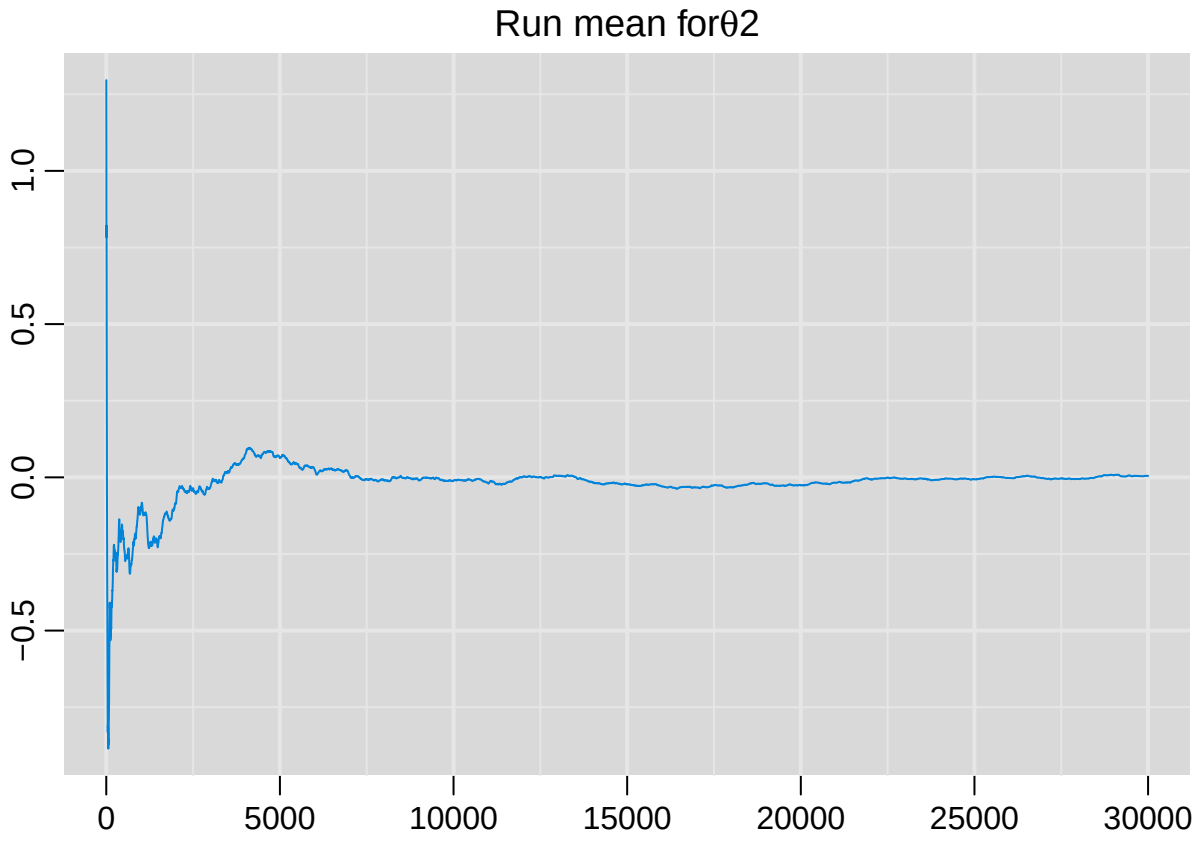

```
# histrograms of marginal posteriors
rmeanplot(t1,lwd=2,main=expression(paste("Run mean for",theta,"1")),mar=c(2,2,1.5,1)+ 0.1)
```

```
## Warning in rmeanplot(t1, lwd = 2, main = expression(paste("Run mean for", :
## Argument 'mcmcout' did not have valid variable names, so names have been
## created for you.
```

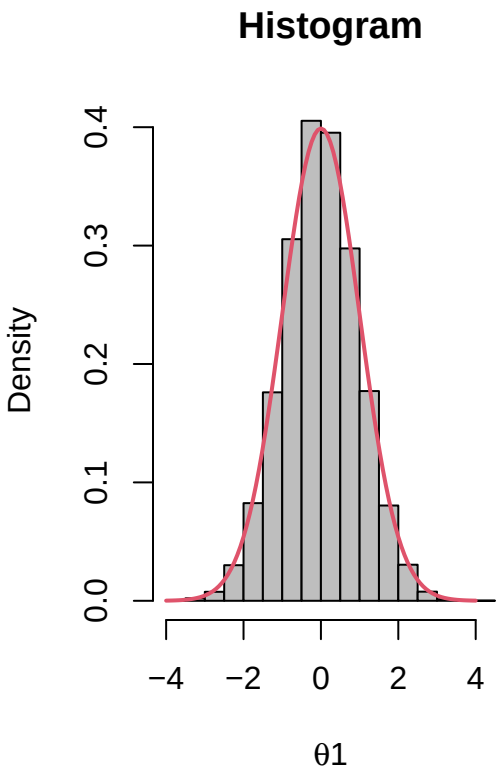


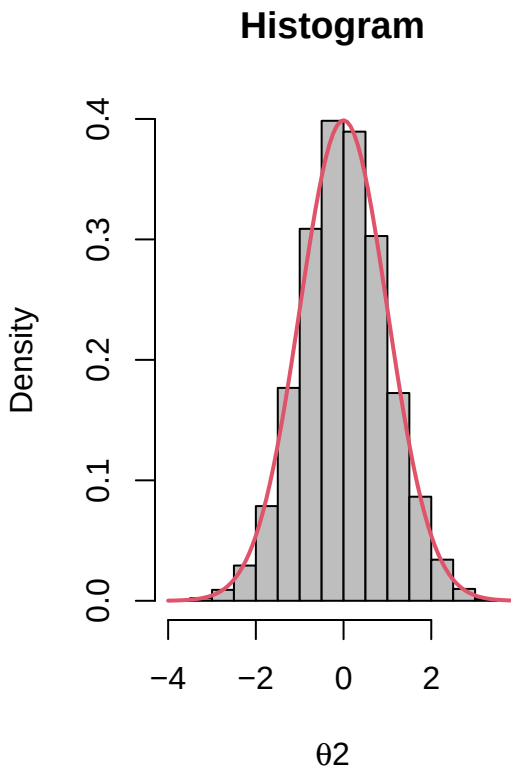

```
rmeanplot(t2,lwd=2,main=expression(paste("Run mean for",theta,"2")),mar=c(2,2,1.5,1)+ 0.1)
```

```
## Warning in rmeanplot(t2, lwd = 2, main = expression(paste("Run mean for", :
## Argument 'mcmcout' did not have valid variable names, so names have been
## created for you.
```




```
par(mfrow=c(1,2),mar=c(5,5,4,2))
hist(t1,freq=F,col="grey",main="Histogram",xlab=expression(paste(theta,"1")))
curve(dnorm(x,0,1),from=-4,4,add=T,lwd=2,col=2)
hist(t2,freq=F,col="grey",main="Histogram",xlab=expression(paste(theta,"2")))
curve(dnorm(x,0,1),from=-4,4,add=T,lwd=2,col=2)
```


Histogram
Density
0.0
0.1
0.2
0.3
0.4
−4
−2
0
2
4
θ1


Histogram
Density
0.0
0.1
0.2
0.3
0.4
−4
−2
0
2
θ2