

Computational Statistics. Chapter 5: MCMC. Solution of exercises

Thierry Denoeux

2024-03-14

```
set.seed(2021)
```

Exercise 1

As the density of ϵ is symmetric, the MH ratio is the ratio of the densities at x^* and $x^{(t-1)}$, i.e., we have

$$R(x^{(t-1)}, x^*) = \frac{f(x^*)}{f(x^{(t-1)})} = \exp(|x^{(t-1)}| - |x^*|).$$

The following function `MH_Laplace` implements the random walk MH algorithm for this problem:

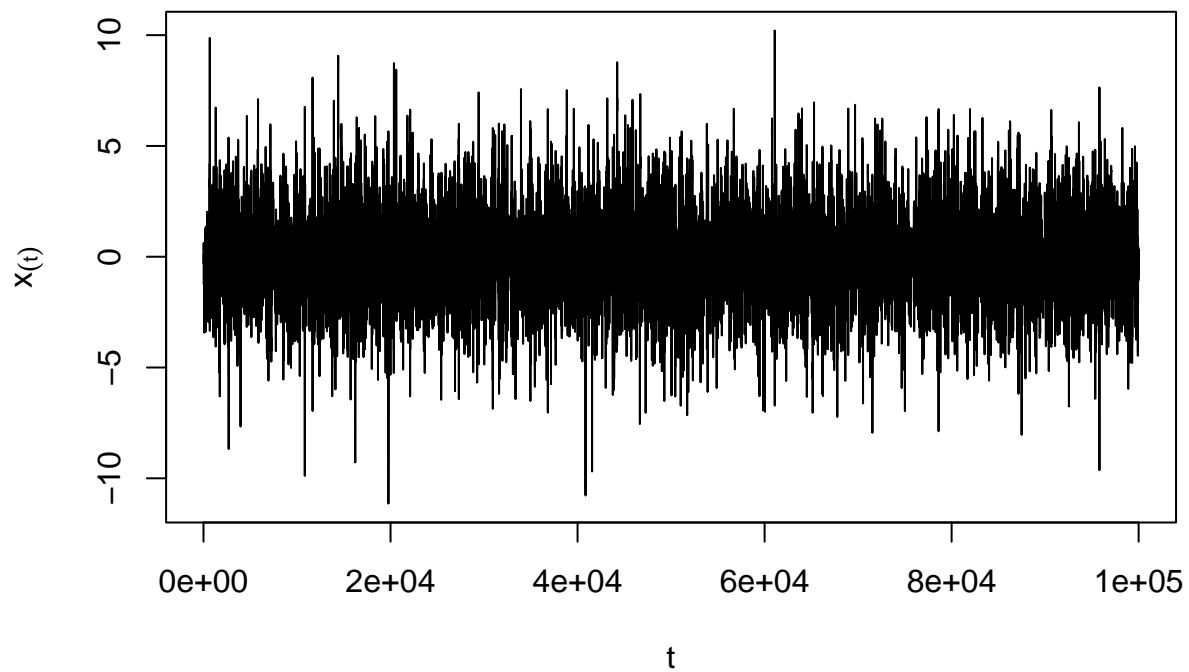
```
MH_Laplace <- function(N,sig){  
  x<-vector(N,mode="numeric")  
  x[1]<-rnorm(1,mean=0,sd=sig)  
  for(t in (2:N)){  
    epsilon<-rnorm(1,mean=0,sd=sig)  
    xstar<-x[t-1]+ epsilon  
    U<-runif(1)  
    R<-exp(abs(x[t-1]) - abs(xstar))  
    if(U <= R) x[t]<-xstar else x[t]<-x[t-1]  
  }  
  return(x)  
}
```

Let us generate a sample of size 10^5 with $\sigma = 10$:

```
x <- MH_Laplace(100000,10)
```

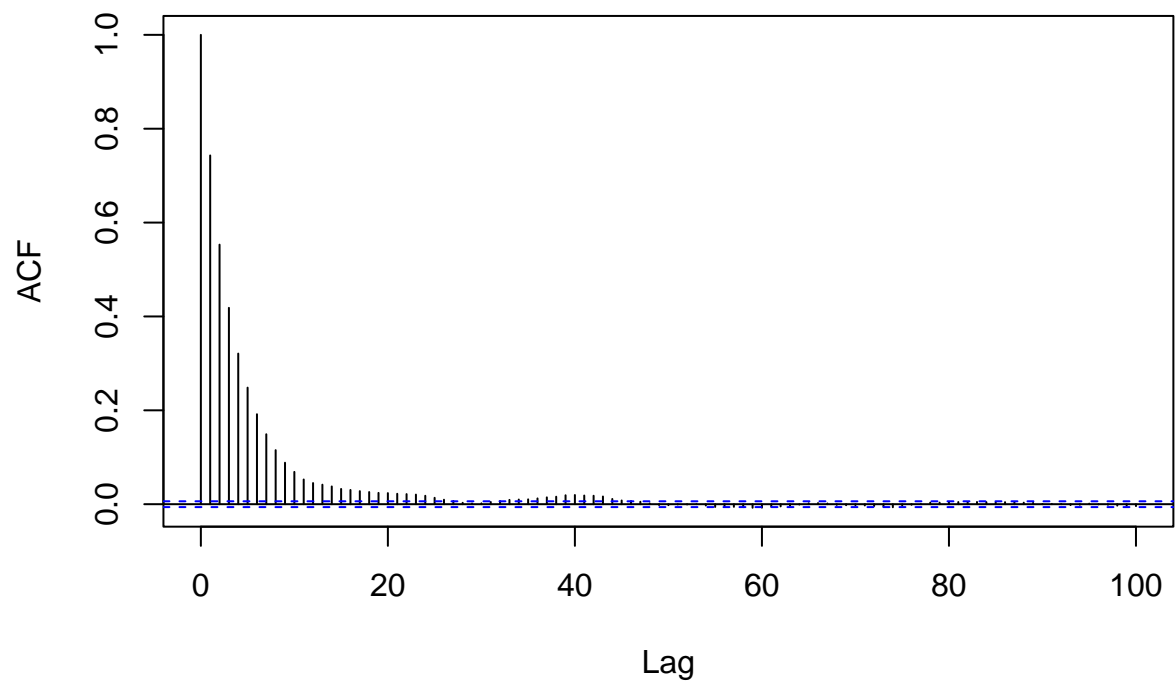
The sample path and correlation plots show good mixing (the chain quickly moves away from its starting value, and the autocorrelation decreases quickly as the lag between iterations increases):

```
plot(x,type="l",xlab='t',ylab=expression(x[(t)]))
```



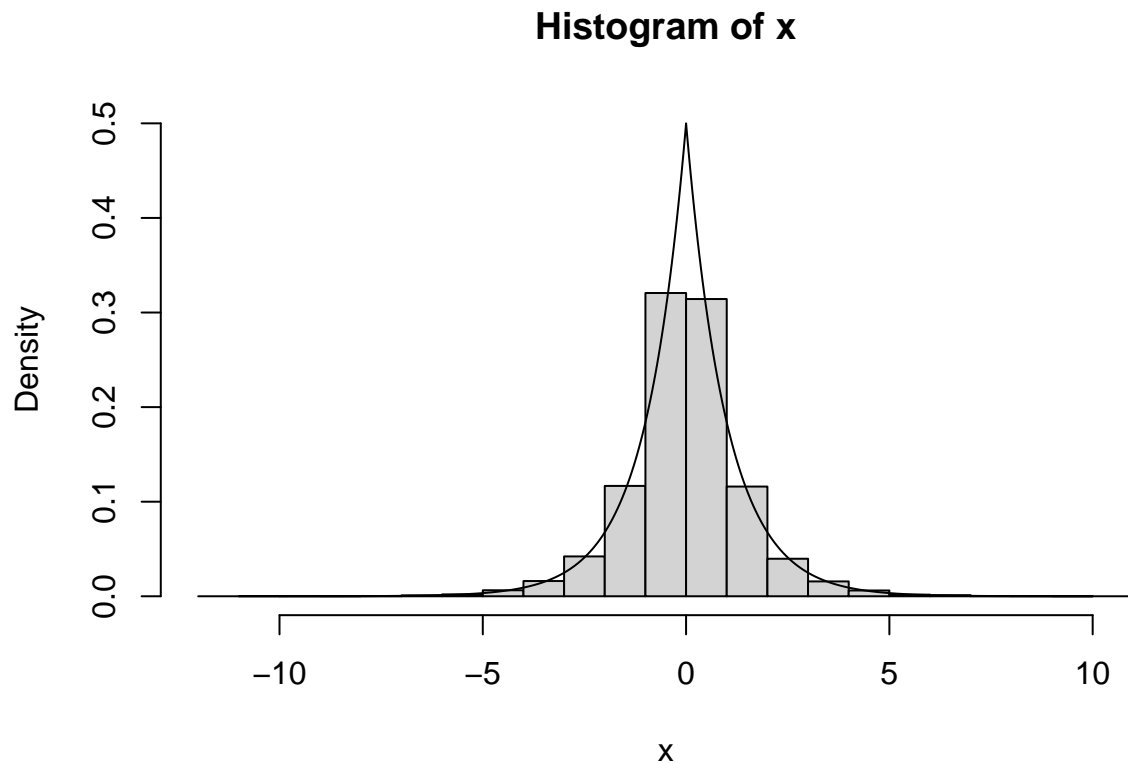
```
acf(x,lag.max=100)
```

Series x



Plot of the histogram with the Laplace density:

```
u<-seq(-10,10,0.01)
fu<-0.5*exp(-abs(u))
hist(x,freq=FALSE,ylim=range(fu))
lines(u,0.5*exp(-abs(u)))
```

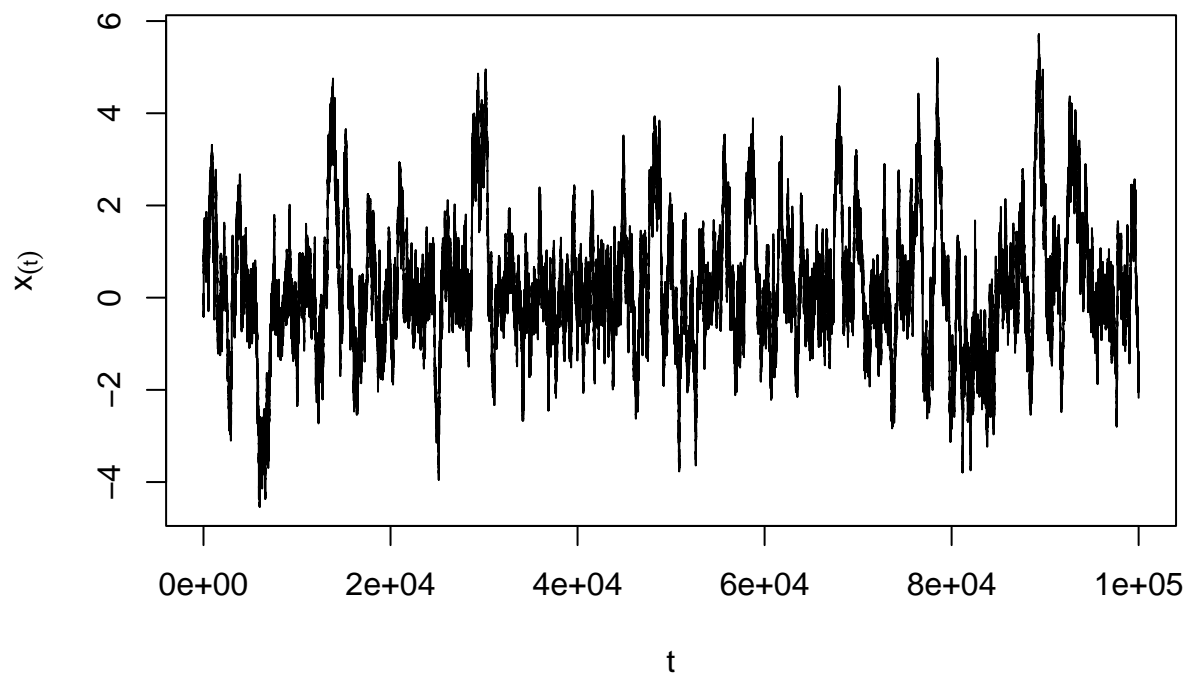


Let us now generate another sample of the same size, this time with $\sigma = 0.1$:

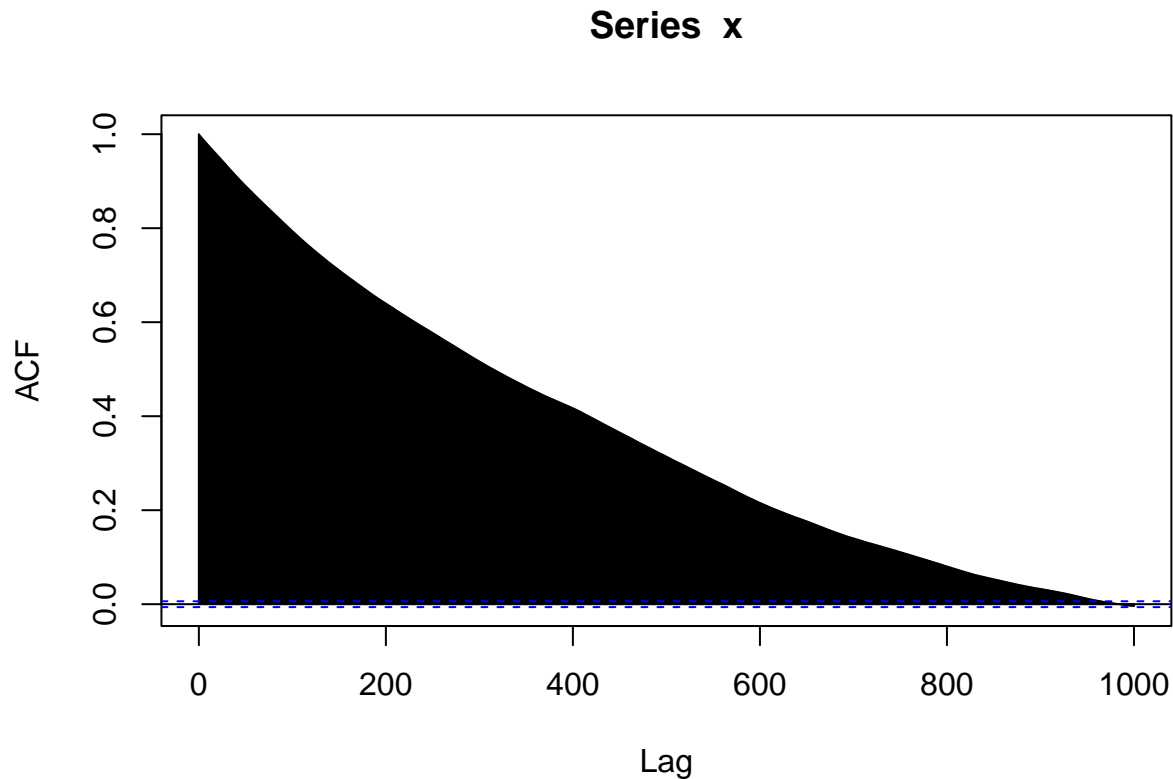
```
x<-MH_Laplace(100000,0.1)
```

This time, the sample path and correlation plots show poor mixing (the chain remains at or near the same value for many iterations, and the autocorrelation decays very slowly):

```
plot(x,type="l",xlab='t',ylab=expression(x[(t)]))
```



```
acf(x, lag.max=1000)
```



Exercise 2

Question a

The likelihood function is

$$L(\beta; y_1, \dots, y_n) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y_i - \beta x_i)^2\right) = (2\pi)^{-n/2} \exp\left(-\frac{1}{2} \sum_{i=1}^n (y_i - \beta x_i)^2\right).$$

The density of the Gamma distribution with shape parameter a and rate b is $f(\beta) \propto \beta^{a-1} \exp(-b\beta) I(\beta > 0)$. Here $a = 2$ and $b = 1$, so $f(\beta) \propto \beta \exp(-\beta) I(\beta > 0)$. Consequently, the posterior density is

$$f(\beta \mid y_1, \dots, y_n) \propto \beta \exp(-\beta) \exp\left(-\frac{1}{2} \sum_{i=1}^n (y_i - \beta x_i)^2\right) I(\beta > 0).$$

Question b

We first write a function that computes the log-likelihood:

```
loglik <- function(beta,x,y){  
  n<- length(x)  
  return(-0.5 * sum((y-beta*x)^2) - n/2*log(2*pi))  
}
```

We then write a function that generates a MC of size N for a given data set:

```

gen_MH<-function(x,y,N){
  beta <- vector(N,mode="numeric")
  beta[1] <- rgamma(1,shape=2,rate=1)
  for(t in (2:N)){
    beta_star <- rgamma(1,shape=2,rate=1)
    u <- runif(1)
    logR <- loglik(beta_star,x,y)-loglik(beta[t-1],x,y)
    if( log(u) <= logR ) beta[t] <- beta_star   else beta[t]<- beta[t-1]
  }
  return(beta)
}

```

Question c

Data generation:

```

beta0<- rgamma(1,shape=2,rate=1) # Generation of beta
cat(beta0)

```

```
## 1.085444
```

```

n<-50
x<-rnorm(n)
y<-x*beta0+rnorm(n)

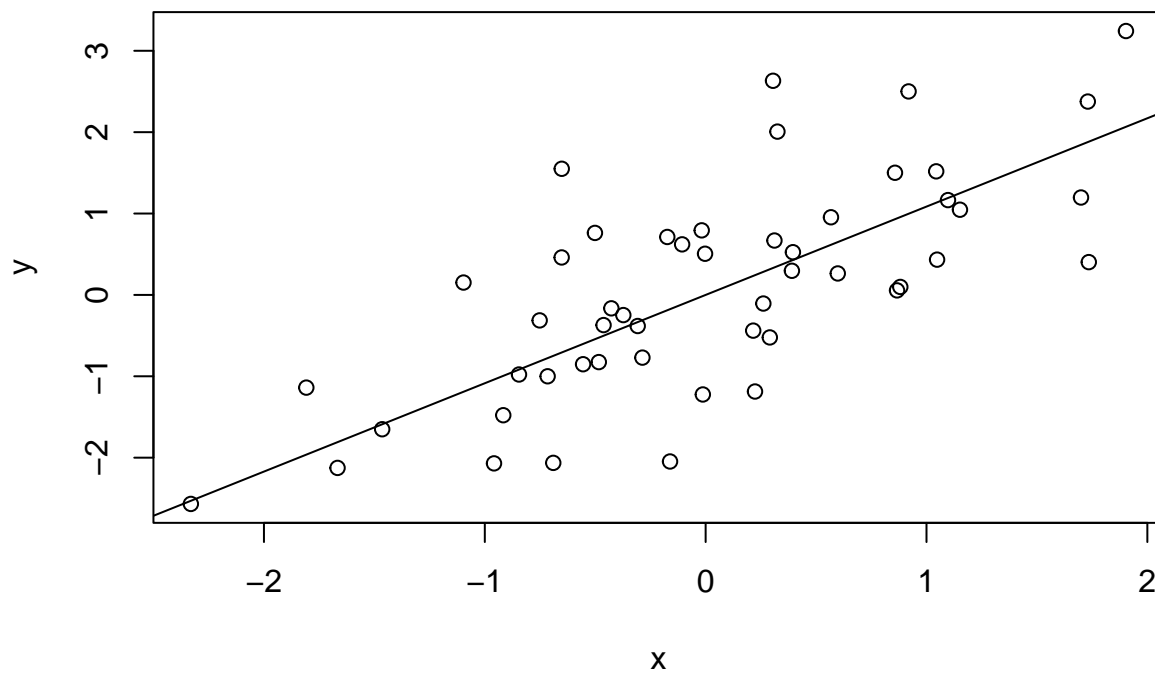
```

Plot of the data:

```

plot(x,y)
abline(0,beta0)

```



Running the MH algorithm:

```

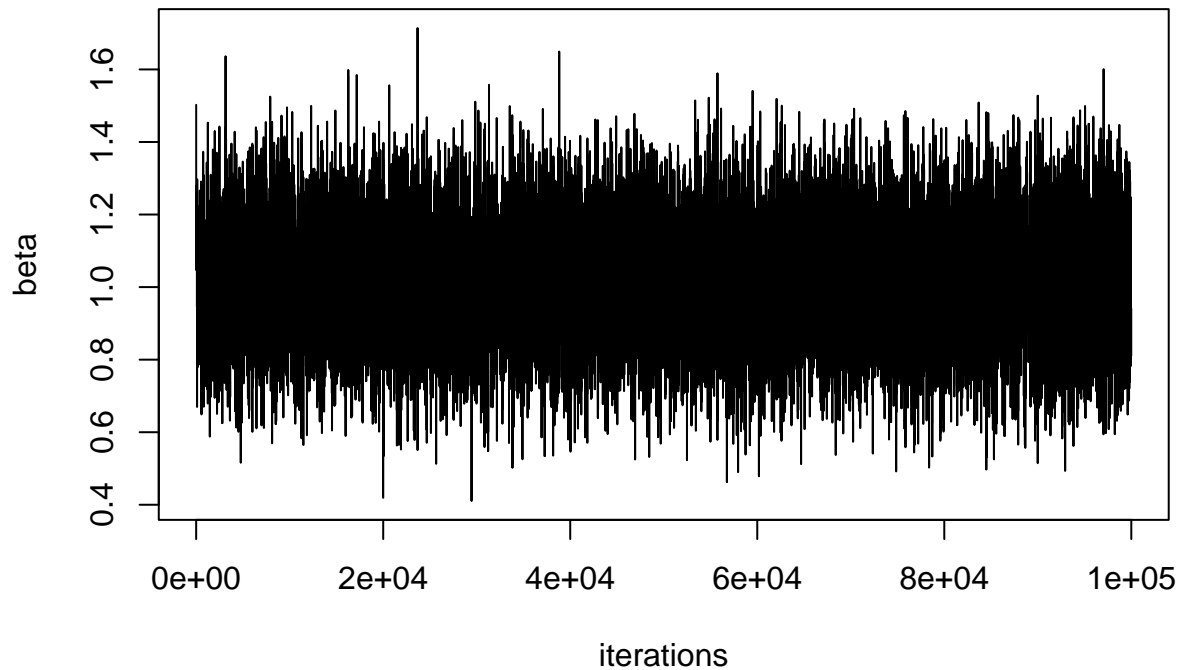
N <- 100000
beta <- gen_MH(x,y,N)

```

Question d

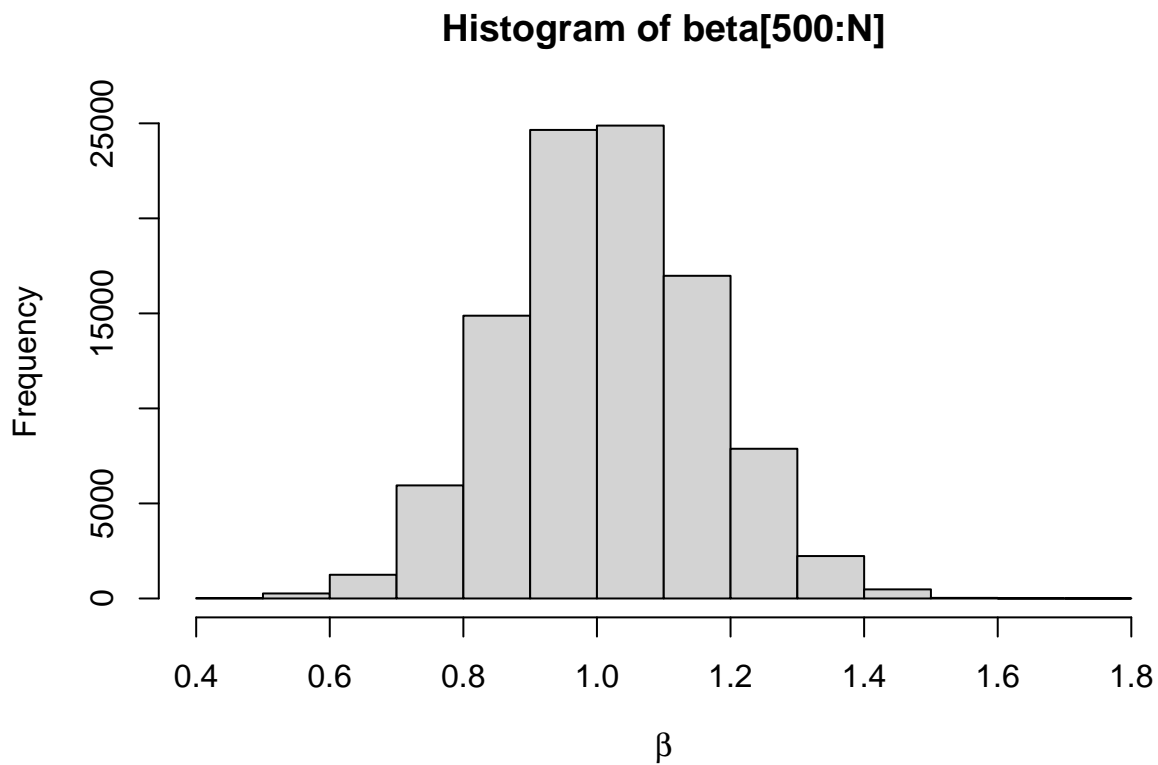
Sample path:

```
plot(beta,type="l",xlab="iterations")
```



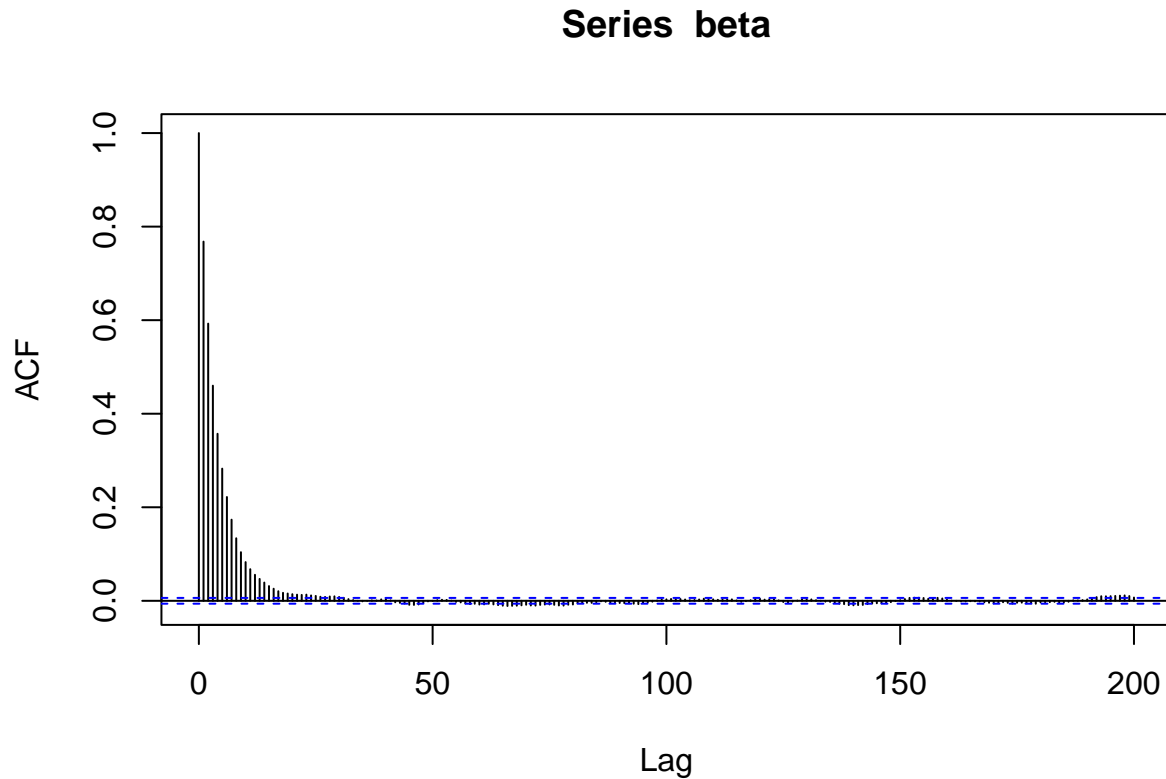
Histogram (leaving out the first 500 values):

```
hist(beta[500:N],xlab=expression(beta))
```



Autocorrelation plot:

```
acf(beta, lag.max=200)
```



Question e

We use the batch means method. We first determine the lag k_0 such that the autocorrelation is small enough to be neglected:

```
ACF<-acf(beta, lag.max=200, plot=FALSE)
k0<-ACF$lag[min(which(abs(ACF$acf)<0.01))]
cat(k0)
```

```
## 26
```

We fix the burn-in period and we compute the number of batches:

```
D<-1000 # burn in
B<-floor((N-D)/k0)
```

We compute the means within each block:

```
Z<-vector(B, mode="numeric")
for(b in 1:B) Z[b]<-mean(beta[(D+(b-1)*k0+1):(D+b*k0)])
```

The estimated simulation standard error is the standard deviation of the batch means divided by the square root of the number of batches:

```
se <- sd(Z)/sqrt(B)
```

Estimated posterior expectation of β and simulation standard error:

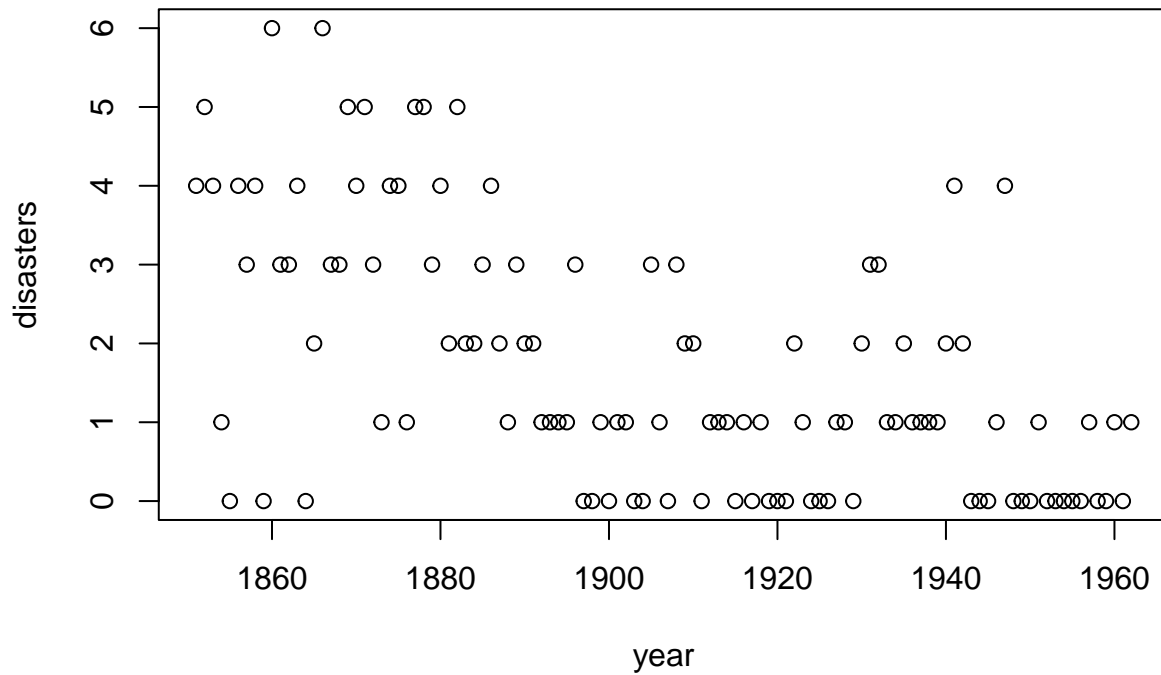
```
print(c(mean(beta[(D+1):N]),se),3)
```

```
## [1] 1.01312 0.00123
```

Exercise 3

Question a

```
coal <- read.table("/Users/Thierry/Documents/R/Data/Compstat/coal.dat",header=TRUE)
plot(coal)
```



Question b

The likelihood function is

$$L(\theta_1, \theta_2, k | \mathbf{x}) \propto \prod_{i=1}^k e^{-\theta_1} \theta_1^{x_i} \prod_{i=k+1}^n e^{-\theta_2} \theta_2^{x_i}.$$

We obtain the posterior distribution by multiplying the likelihood and the prior:

$$f(\theta_1, \theta_2, k | \mathbf{x}) \propto \underbrace{\theta_1^{\alpha_{01}-1} e^{-\beta_{01}\theta_1}}_{f(\theta_1)} \underbrace{\theta_2^{\alpha_{02}-1} e^{-\beta_{02}\theta_2}}_{f(\theta_2)} \underbrace{\prod_{i=1}^k e^{-\theta_1} \theta_1^{x_i} \prod_{i=k+1}^n e^{-\theta_2} \theta_2^{x_i}}_{L(\theta_1, \theta_2, k | \mathbf{x})}.$$

Now,

$$\begin{aligned}
f(\theta_1 \mid \theta_2, k, \mathbf{x}) &\propto \frac{f(\theta_1, \theta_2, k \mid \mathbf{x})}{f(\theta_2, k)} \\
&\propto L(\theta_1, \theta_2, k \mid \mathbf{x}) f(\theta_1) \\
&\propto \theta_1^{\alpha_{01}-1} e^{-\beta_{01}\theta_1} \prod_{i=1}^k e^{-\theta_1} \theta_1^{x_i} \prod_{i=k+1}^n e^{-\theta_2} \theta_2^{x_i} \\
&\propto \theta_1^{\alpha_{01} + \sum_{i=1}^k x_i - 1} \exp(-(\beta_{01} + k)\theta_1).
\end{aligned}$$

Consequently,

$$f(\theta_1 \mid \theta_2, k, \mathbf{x}) = f(\theta_1 \mid k, \mathbf{x}) \sim G(\alpha_{01} + \sum_{i=1}^k x_i, \beta_{01} + k).$$

Symmetrically, we obtain in the same way

$$f(\theta_2 \mid \theta_1, k, \mathbf{x}) = f(\theta_2 \mid k, \mathbf{x}) \sim G(\alpha_{02} + \sum_{i=k+1}^n x_i, \beta_{02} + k).$$

We can observe that θ_1 and θ_2 are conditionally independent given k and \mathbf{x} .

Finally, the conditional probability mass function of k is

$$\begin{aligned}
f(k \mid \theta_1, \theta_2, \mathbf{x}) &\propto \frac{f(\theta_1, \theta_2, k \mid \mathbf{x})}{f(\theta_1)f(\theta_2)} \\
&\propto L(\theta_1, \theta_2, k \mid \mathbf{x}) \\
&\propto \exp[k(\theta_2 - \theta_1)] \left(\frac{\theta_1}{\theta_2}\right)^{\sum_{i=1}^k x_i}.
\end{aligned}$$

Question c

The following function implements the Gibbs algorithm for this problem:

```

gibbs<-function(x,N,alpha10,beta10,alpha20,beta20){
  n<-length(x)
  # Initialization
  theta1 <- vector(length=N,mode="numeric")
  theta2 <- vector(length=N,mode="numeric")
  k <- vector(length=N,mode="numeric")
  p<-vector(length=n,mode="numeric")
  # First cycle
  # Sampling of k[1] from a uniform distribution
  k[1]<-sample(n,size=1)
  theta1[1]<-rgamma(1,shape=alpha10+sum(x[1:k[1]]),rate=beta10+k[1])
  theta2[1]<-rgamma(1,shape=alpha20+sum(x[(k[1]+1):n]),rate=beta20+n-k[1])
  for(t in (2:N)){
    # Conditional pmf of k
    for (j in (1:n)){
      p[j]<- (theta1[t-1]/theta2[t-1])^sum(x[1:j]) * exp(j*(theta2[t-1]-theta1[t-1]))
    }
    p<-p/sum(p)
    k[t]<- sample(n,size=1,prob=p)
    theta1[t]<-rgamma(1,shape=alpha10+sum(x[1:k[t]]),rate=beta10+k[t])
  }
}

```

```

    theta2[t]<-rgamma(1,shape=alpha20+sum(x[(k[t]+1):n]),rate=beta20+n-k[t])
  }
  return(list(k=k,theta1=theta1,theta2=theta2))
}

```

We can run this algorithm on the data:

```

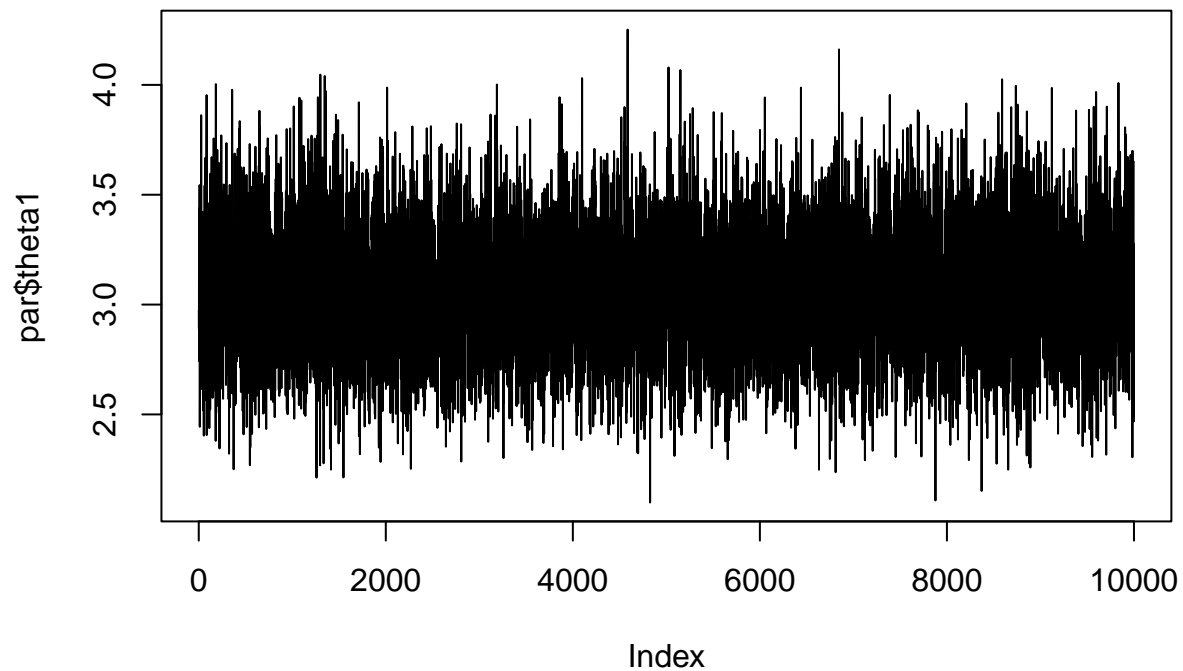
N<-10000
alpha10<-0.5
alpha20<-0.5
beta10<-1
beta20<-1
par<-gibbs(x=coal$disasters,N,alpha10,beta10,alpha20,beta20)

```

Question d

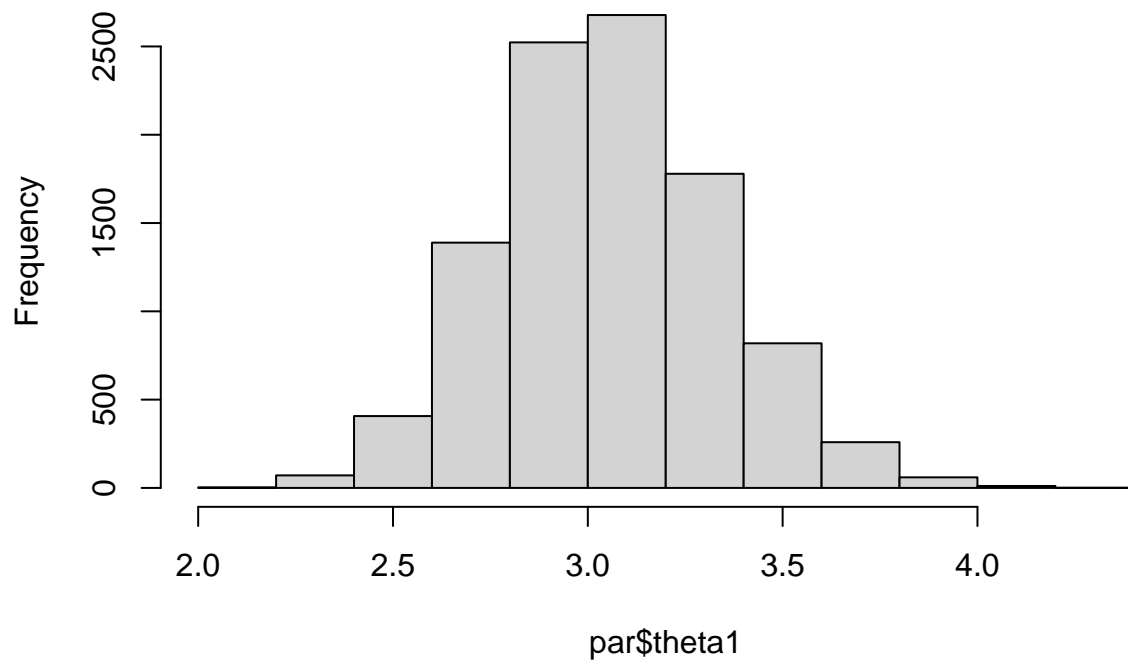
Plots for θ_1 :

```
plot(par$theta1,type="l")
```



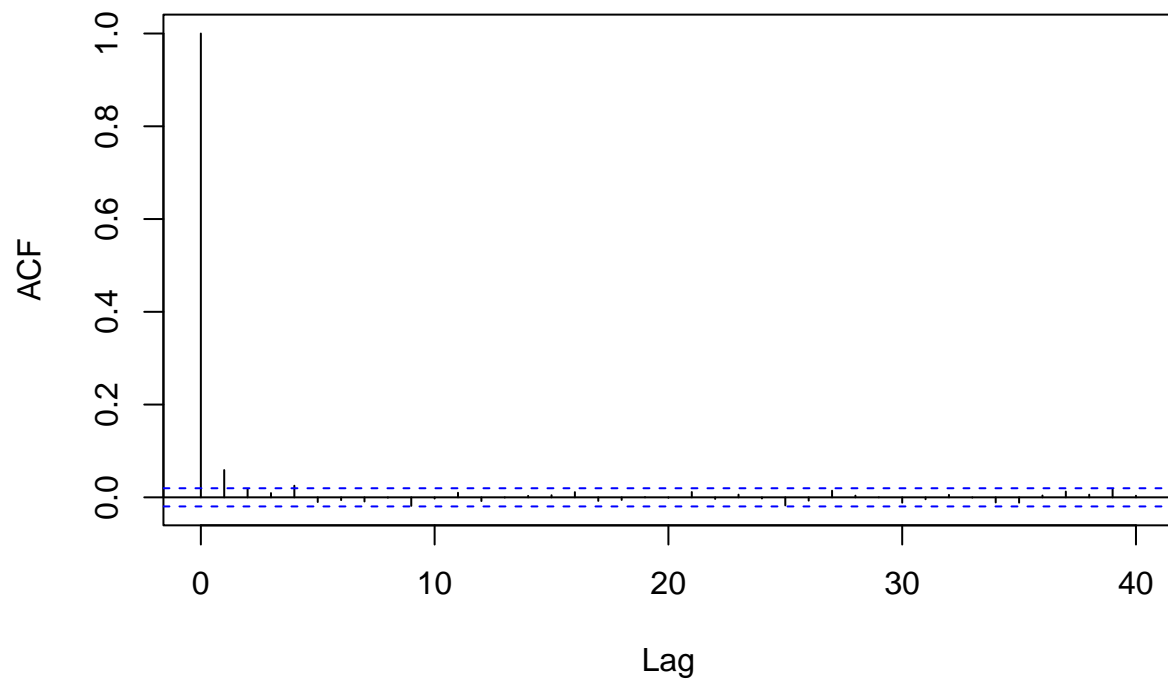
```
hist(par$theta1)
```

Histogram of par\$theta1



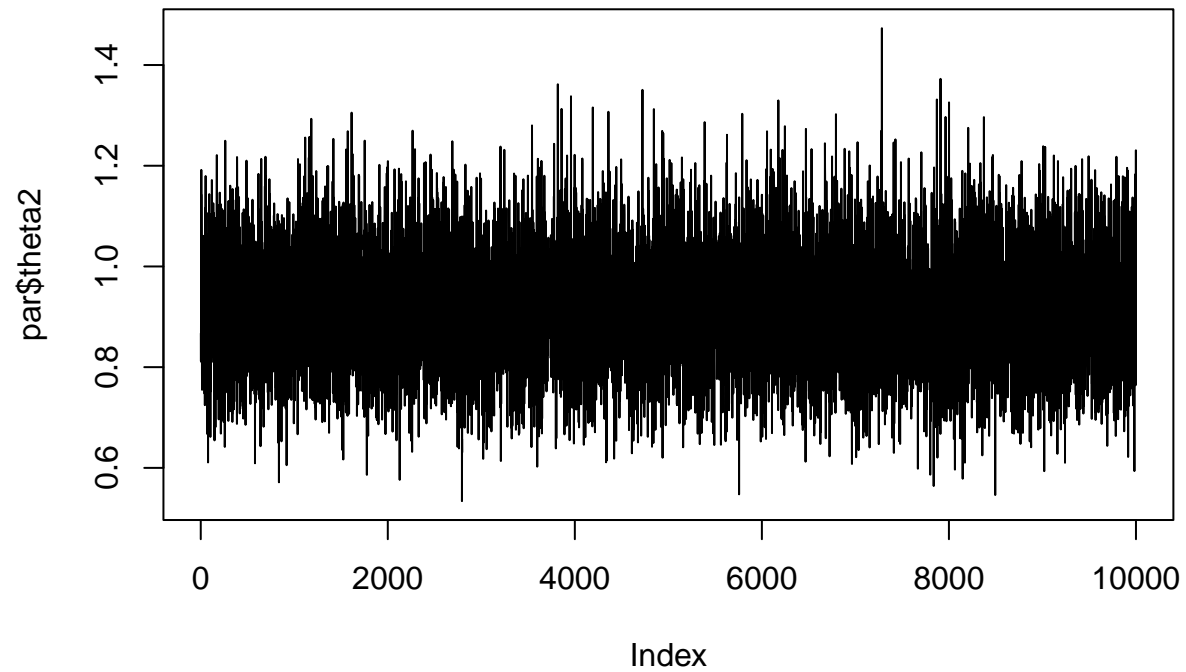
```
acf(par$theta1)
```

Series par\$theta1



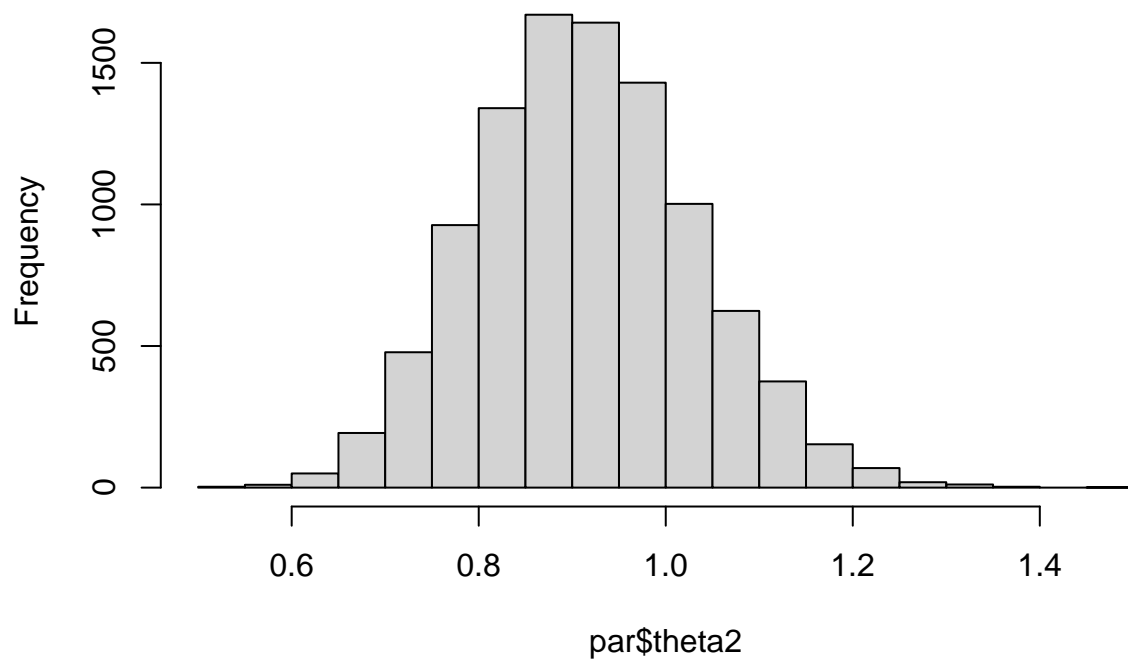
Plots for θ_2 :

```
plot(par$theta2,type="l")
```



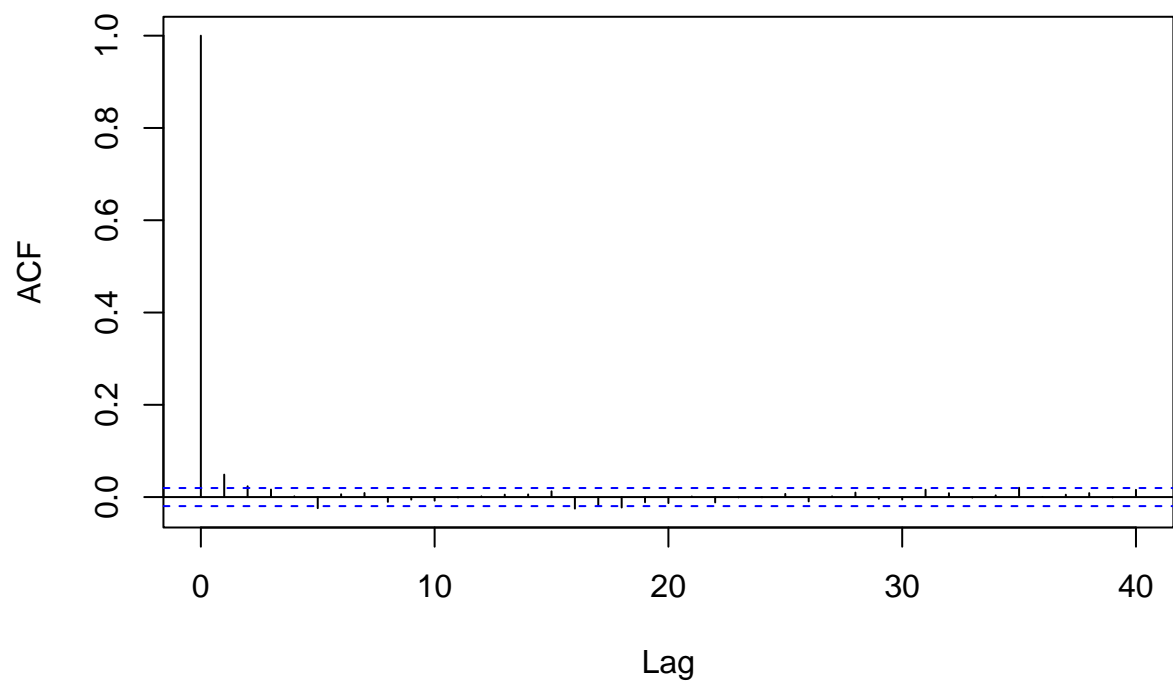
```
hist(par$theta2)
```

Histogram of par\$theta2



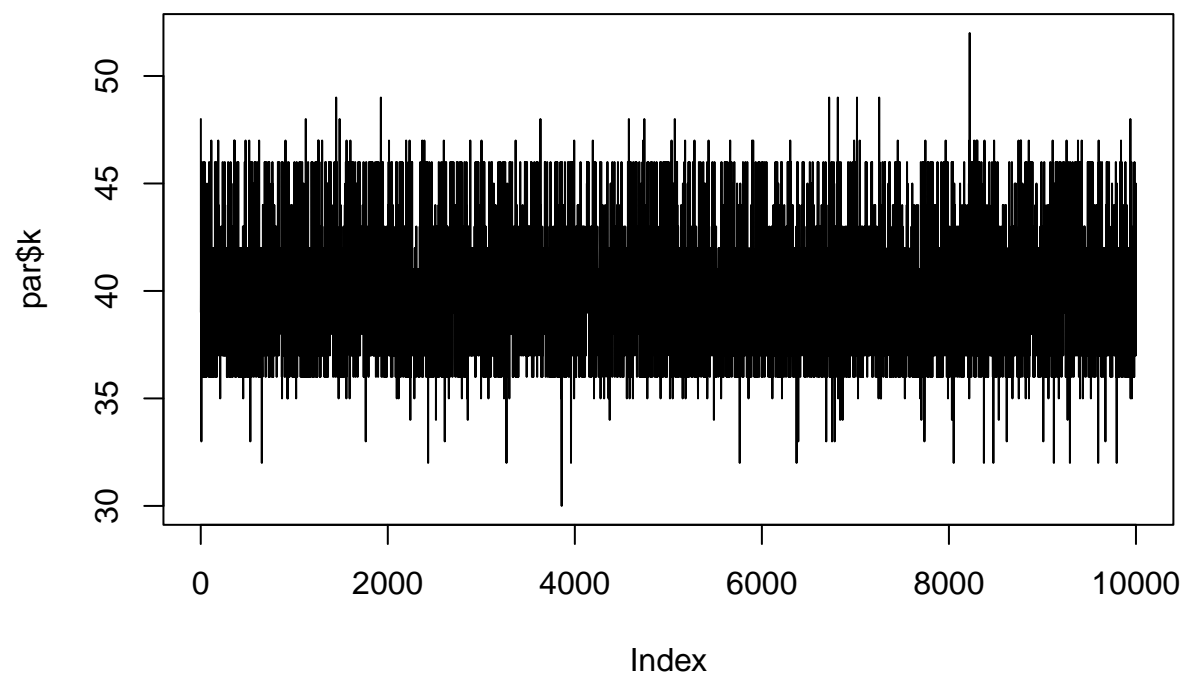
```
acf(par$theta2)
```

Series par\$theta2

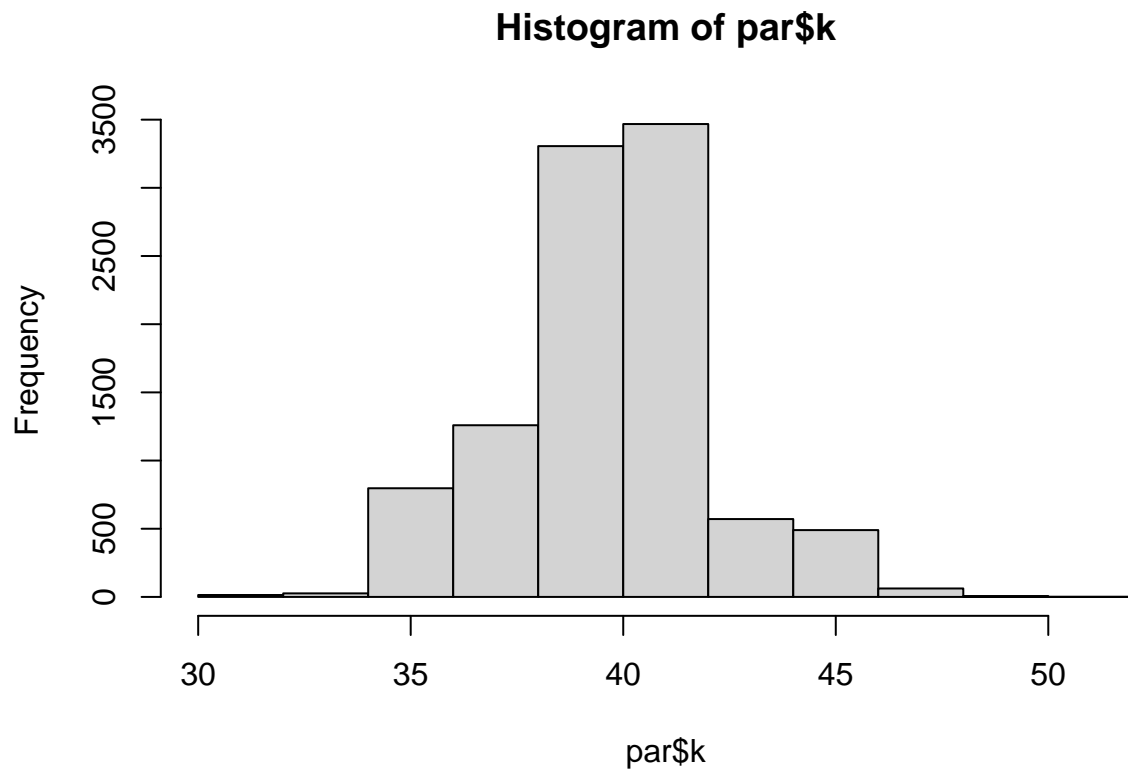


Plots for k :

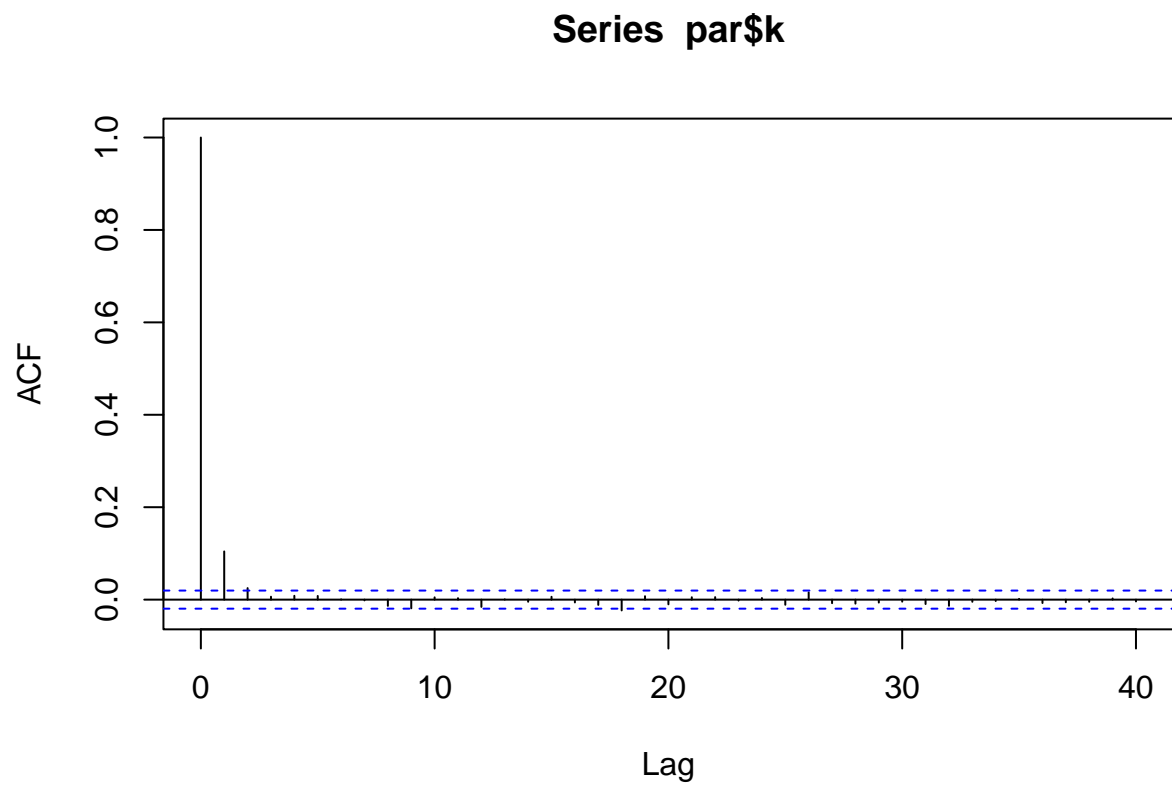
```
plot(par$k,type="l")
```



```
hist(par$k)
```



```
acf(par$k)
```



Question e

We set the lag to $k_0 = 10$ and the burn-in period to $D = 1000$, and we compute the number B of batches:

```
k0 <- 10 # lag
D <- 1000 # burn in
B<-floor((N-D)/k0) # number of batches
```

Estimated conditional expectation and simulated standard error for θ_1 :

```
Z<-vector(B,mode="numeric")
for(b in (1:B)) Z[b]<-mean(par$theta1[(D+(b-1)*k0+1):(D+b*k0)])
se <- sd(Z)/sqrt(B)
cat("mean=", mean(par$theta1[(D+1):N]), "se =", se)
```

```
## mean= 3.051553 se = 0.003175602
```

Estimated conditional expectation and simulated standard error for θ_2 :

```
for(b in (1:B)) Z[b]<-mean(par$theta2[(D+(b-1)*k0+1):(D+b*k0)])
se <- sd(Z)/sqrt(B)
cat("mean=", mean(par$theta2[(D+1):N]), "se =", se)
```

```
## mean= 0.9144713 se = 0.001345767
```

Estimated conditional expectation and simulated standard error for k :

```
for(b in (1:B)) Z[b]<-mean(par$k[(D+(b-1)*k0+1):(D+b*k0)])
se <- sd(Z)/sqrt(B)
cat("mean=", mean(par$k[(D+1):N]), "se =", se)
```

```
## mean= 40.15444 se = 0.02908424
```

The change is estimated to have taken place in the year

```
cat(coal$year[round(mean(par$k[(D+1):N]))])
```

```
## 1890
```

Exercise 4

Question a

Let us first read and transform the data:

```
data <- read.table("/Users/Thierry/Documents/R/Data/Compstat/investment.txt",header=TRUE)
attach(data)
y <- Invest/(CPI*10)
time <- 1:15
GNP1<-data$GNP/(CPI*10)
X <- cbind(rep(1,15),time,GNP1,Interest,Inflation)
data1 <- data.frame(Invest=y,time, GNP=GNP1,Interest,Inflation)
```

Let us plot the data:

```
plot(data1)
```