

Lab4 Computational Statistics

Andreas C Charitos(*andch552*) Omkar Bhutra (*omkbh878*)

18 Feb 2019

Question 1-Computations with Metropolis-Hastings

Subquestion 1-Sample using Lognormal

We have the target probability function $f(x) = x^5 e^{-x}$, $x > 0$ and we are going to generate samples using Metropolis-Hastings Algorithm by using $LN(X_t, 1)$ as proposal distribution. So sampling from our proposal we are going to use Metropolis-Hastings to approximate our target function.

```
#set seed
set.seed(123456)
#this is our target function
target_dist<-function(x){
  return(x^5*exp(-x))
}

#initialize rejection counts
Rej<-0
#metropolis-hastings using lognormal
#input(n sample,starting point,sd given,T to return output ,T to return plot)
f.MCMC.MH<-function(nstep,X0,props,output=T,draw_plot=T){
  #initialize the first point
  X0<-X0
  #vector to use in the plot
  vN<-1:nstep
  #initialize our sample vector
  vecX<-rep(X0,nstep);
  #for loop for the algorithm
  for(i in 2:nstep){
    #take the previous point in our sample
    X<-vecX[i-1]
    #create candidate point from log-normal
    Xcand<-rlnorm(1,meanlog=log(X),sdlog=props)
    #random point
    u<-runif(1)
    #numerator of ratio
    num <- (target_dist(Xcand)*dlnorm(X,meanlog=log(Xcand),sdlog=props))
    #denominator of ratio
    den <- (target_dist(X)*dlnorm(Xcand,meanlog=log(X),sdlog=props))

    #a<-min(c(1,(target_dist(Y)*dlnorm(X,meanlog=Y,sdlog=props)/(target_dist(X)*dlnorm(Y,meanlog=X,sdlog=props)))
    a<-min(1,num/den)
    #check condition if T accept the candidate point
    if (u <=a){
      vecX[i]<-Xcand
    }
    #reject candidate stay in the current point
    else{
```

```

    vecX[i]<-X
    #rejection count
    Rej<-Rej+1
  }

}
#draw plot
if(draw_plot==T){

plot(vN,vecX,pch=19,cex=0.3,col="black",xlab="t",ylab="X(t)",
     main="Plot of sample using LN",ylim=c(1,20),type="l")
abline(h=0)
abline(h=1.96,col="red")
abline(h=15,col="red")
grid(30,30,col="lightgray")
}
#return output of sample,rejections
if(output==T){
  return(list("sample"=vecX,"rejections"=Rej))
}
}

```

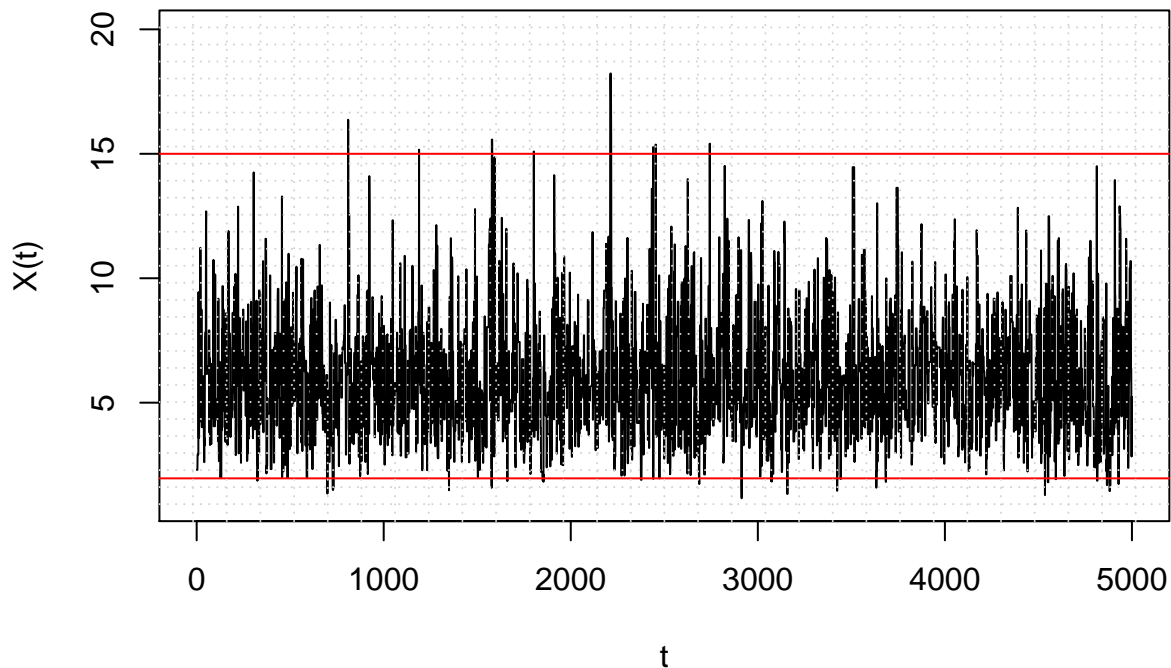
Plot of the chain using LN proposal

```

set.seed(123456)
#create a 5000 sample from our proposal
f<-f.MCMC.MH(5000,rlnorm(1,0,1),1)

```

Plot of sample using LN



```
cat("The rejection rate is :",f$rejections/5000)
```

```
## The rejection rate is : 0.5582
```

The plot shows that we have well mixing chain and our algorithm creates candidate points that mostly get accepted and we move to other points but we have a big range of our points. When we examine the rejection rate which is 0.55 means that half of the candidate point rejected and we stay in the same point.

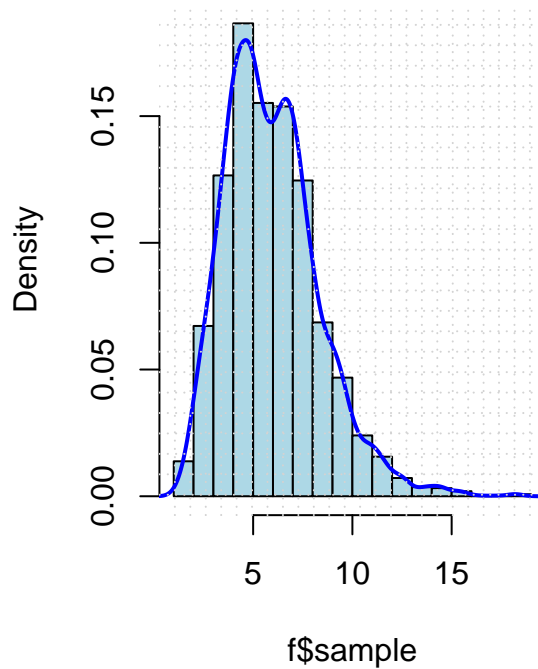
Plot of the densities

The above plots show the density from our sample and the target density we need to sample from. As we can see the lognormal is good approximation of our target density.

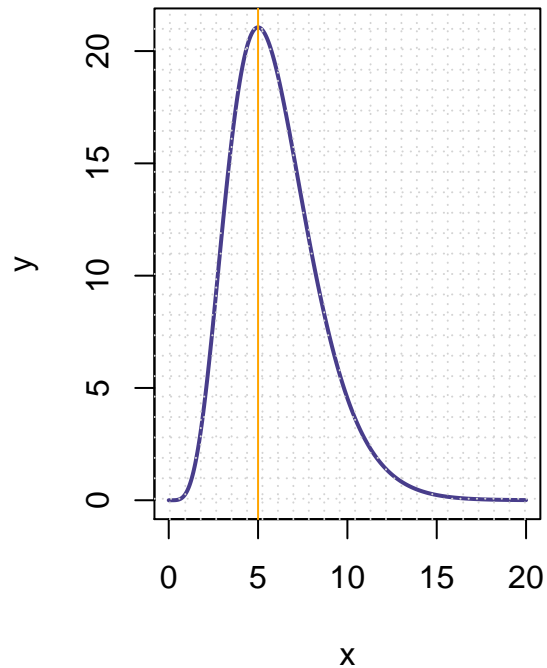
```
par(mfrow=c(1,2))
hist(f$sample,prob=T,col="lightblue",main="Histogram/Density of LN sample")
lines(density(f$sample),col="blue",lwd=2)
grid(25,25,col="lightgray")

#target
x=seq(0,20,0.01)
y=target_dist(x)
plot(x,y,type="l",main="Density of target function",
     col="darkslateblue",lwd=2)
abline(v=5,col="orange")
grid(25,25,col="lightgray")
```

Histogram/Density of LN sample



Density of target function



Subquestion 2-Sample using Chi-square

We are now going to use a new proposal distribution to sample from which is $x^2 = (fraction(X_t + 1))$ in order to see if we will get a better approximation for our target distribution.

```
#initialize rejection counts
Rej1<-0
#our function to sample from chi-square
f.MCMC.MH1<-function(nstep,X0,output=T,draw_plot=T){
  X0<-X0
  #vector to use for the plot
  vN<-1:nstep
  #initialize our sample vector
  vecX<-rep(X0,nstep);
  #for loop to create sample
  for (i in 2:nstep){
    #take the previous point in our sample
    Xt<-vecX[i-1]
    #create candidate using chi-square
    Xcand<-rchisq(1,df=floor(Xt+1))
    #
    u<-runif(1)
    #numerator of the ratio
    num <- target_dist(Xcand)*dchisq(Xt, df=floor(Xcand+1))
    #denominator of the ratio
    den <- target_dist(Xt)*dchisq(Xcand, df=floor(Xt+1))
```

```

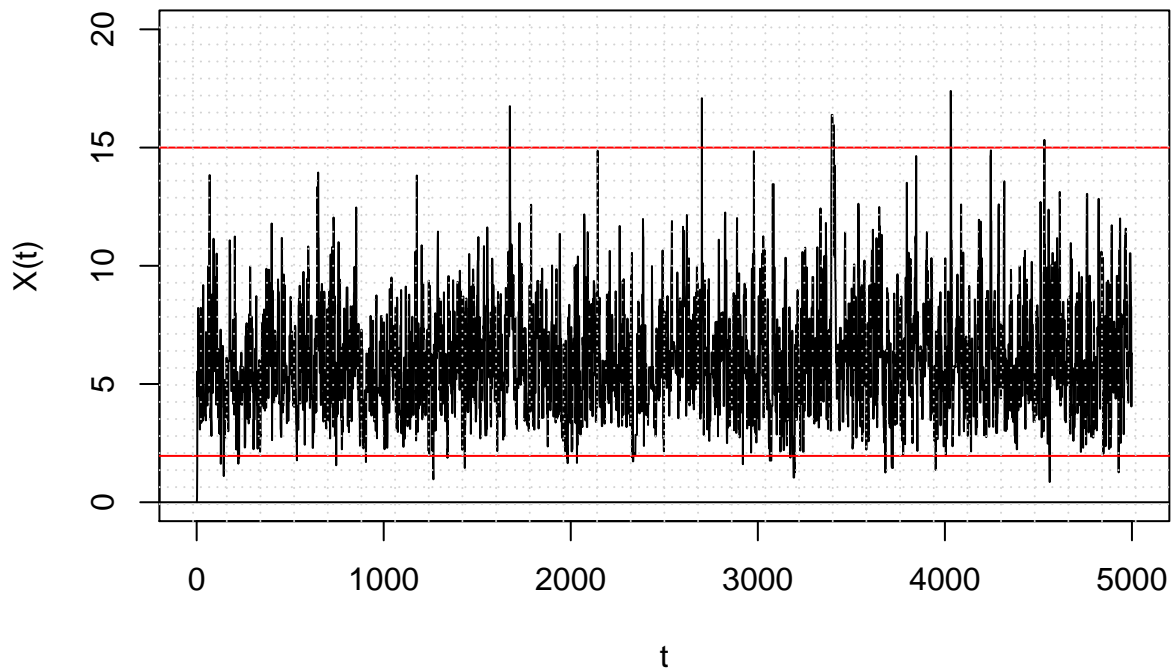
# a <- min(c(1, (target_dist(Y)*dchisq(Xt, df=floor(Y+1))/(target_dist(Xt)*dchisq(Y, df=floor(Xt+1))))))
a <- min(1, num/den)
# check condition if T accept the candidate point
if (u <= a){
  vecX[i] <- Xcand
}
# reject candidate stay on current point
else{
  vecX[i] <- Xt
  # rejection count
  Rej1 <- Rej1 + 1
}

# return plot
if (draw_plot == T){
  plot(vN, vecX, pch=19, cex=0.3, col="black",
       xlab="t", ylab="X(t)", main="", ylim=c(0, 20), type="l")
  abline(h=0)
  abline(h=1.96, col="red")
  abline(h=15, col="red")
  grid(30, 30, col="lightgray")
}
# return output of sample, rejections
if (output == T){
  return(list("sample"=vecX, "rejections"=Rej1))
}
}

```

Plot of the chain using chi-square proposal

```
f2 <- f.MCMC.MH1(5000, rchisq(1, 1), T, T)
```



```
cat("The rejection rate is :",f2$rejections/5000)#rejection rate
```

```
## The rejection rate is : 0.4062
```

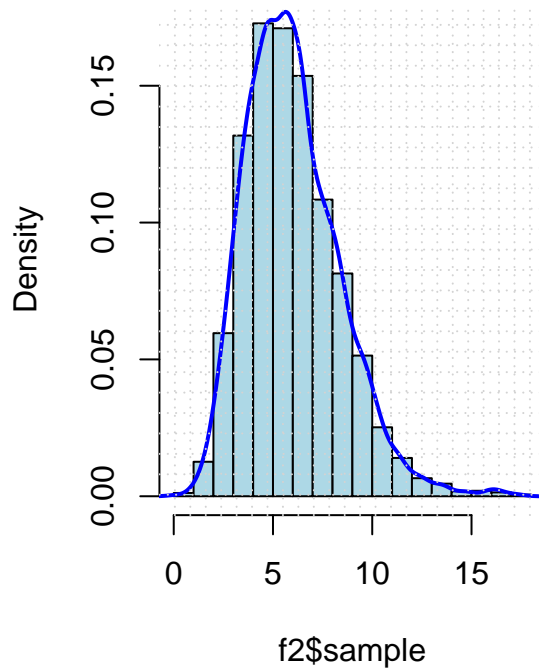
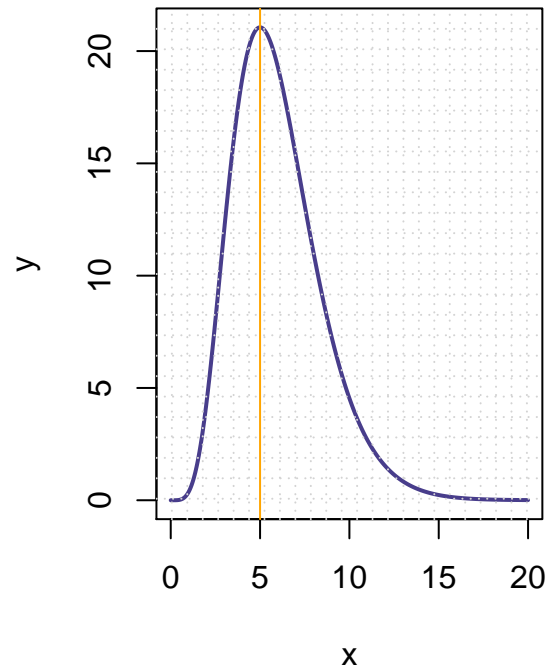
As we can see now from the plot of the chain we obtain we have similar with lognormal which means that also the function seems to approximate quite well the target function the points are changing and we move to new points but now we have smaller range of the points. Looking at the rejection rate we can see that is smaller than the previous of lognormal.

Plot of the densities

The above plots show the density from our sample and the target density we need to sample from. As we can see the chi-square is a very good approximation of our target density.

```
par(mfrow=c(1,2))
#histogram density of the sample
hist(f2$sample,prob=T,col="lightblue",main="Histogram/Density of LN sample")
lines(density(f2$sample),col="blue",lwd=2)
grid(25,25,col="lightgray")

# #target density
x=seq(0,20,0.01)
y=target_dist(x)
plot(x,y,type="l",main="Density of target function",
     col="darkslateblue",lwd=2)
abline(v=5,col="orange")
grid(25,25,col = "lightgray")
```

Histogram/Density of LN sample**Density of target function**

Subquestion 2-Sample using Chi-square

In conclusion comparing the 2 proposal distribution we can see that using Metropolis-Hastings Algorithm with lognormal we got well mixing chains meaning that the algorithm seems to explore a bigger span in the space and we don't stuck in the same point for long periods but the ranges of the points are different with the range of the chi-square being smaller. Finally, we plotted the densities obtained by the 2 proposals against the target density and confirm that chi-square is a slightly better proposal than the lognormal with seems natural because its pdf looks more like the target function.

Subquestion 2-Analyze convergence with Gelman-Rubin method

In this section we are going to analyze the convergence of the 10 sequences using the chi-square proposal with starting points from 1, 2, ..., 10 with the Gelman-Rubin method.

```
library(coda)
```

```
## Warning: package 'coda' was built under R version 3.5.2
```

```
set.seed(123456)
#number of sample
nsample<-5000
#number of starting points from 1...10
steps<-seq(1,10,1)
#initialize matrix(each row is a sample of 5000 for every starting point)
X1<-matrix(0,length(steps),nsample)
#for loop to fill the matrix
for(step in steps){
```

```

X1[steps,]<-f.MCMC.MH1(nsample,step,output=T,draw_plot=F)$sample
}
#empty list
f1<-list()
#iterate every row and make it as mcmc object
for (i in 1:dim(X1)[1]){

  f1[[i]]<-as.mcmc(X1[i,])
}

#print gelman-rubin diagnostics
print(gelman.diag(f1))

```

```

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1

```

The diagnostics from Gelman-Rubin method imply that an upper C.I. greater than 1 is an indicator that the method did not converge. In our case it seems that the method converges.

Subquestion 5-Integral estimation with sampling

we need to calculate the integral $\vartheta = \int_0^\infty xf(x)dx$.

Lets set $g(x) = x$ and $f(x) = x^5e^{-(x)}$ which is target function from before. The integral $\int_0^\infty f(x)dx = 1$ because it's the CDF of the $f(x)$. So can now write that if X is a random variable that follows $f(x)$ then:

$$\vartheta = \int_0^\infty g(x)f(x)dx = E[g(X)]$$

and we know that expected value is the mean of the sample thus :

$$\hat{\vartheta} = \frac{1}{n} \sum_{i=1}^n g(x_i) = \frac{1}{n} \sum_{i=1}^n x_i, \quad \forall x_i \sim f(x)$$

What this formula tells us is that if we take a sample from our $f(x)$ and transform it using $g(x)$ and take the mean of the sample we can calculate the integral of $f(x)$ which in our case is the CDF of $f(x)$.

```

set.seed(123456)
#calculate mean sample using the lognormal sample
s1<-f.MCMC.MH(5000,rlnorm(1,0,1),1,output=T,draw_plot=F)
#calculate mean using the chi-square sample
s2<-f.MCMC.MH1(5000,rchisq(1,floor(1)),T,F)

cat("The integral estimation with a sample using log normal is :",mean(s1$sample))

## The integral estimation with a sample using log normal is : 5.952158
cat("\n-----")

##
## -----
cat("\nThe integral estimation with a sample using chi-square is :",mean(s2$sample))

##
## The integral estimation with a sample using chi-square is : 6.004897

```


Subquestion 6-Gamma distribution and its integral

The probability density for gamma distribution is given by the formula :

$$f(x; \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)} \quad \text{for } x > 0, \alpha, \beta > 0$$

Where

$$\Gamma(a) = (a-1)!$$

if we set $\alpha = 1$ and $\beta = 6$ we have:

$$f(x; \alpha = 6, \beta = 1) = \frac{x^5 e^{-x}}{120} (1)$$

we need to calculate

$$\int_{-\infty}^0 x f(x) dx (2)$$

if we assume that $f(x)$ is the gamma distribution (1) the requested interval (2) is the CDF Of the gamma distribution which is the expected value of X

$$\int_{-\infty}^0 x f(x) dx = E[X]$$

the expected value of the gamma distribution is then given by $E(X) = \frac{\alpha}{\beta} = \frac{6}{1} = 6$

Finally, the requested integral is 6.

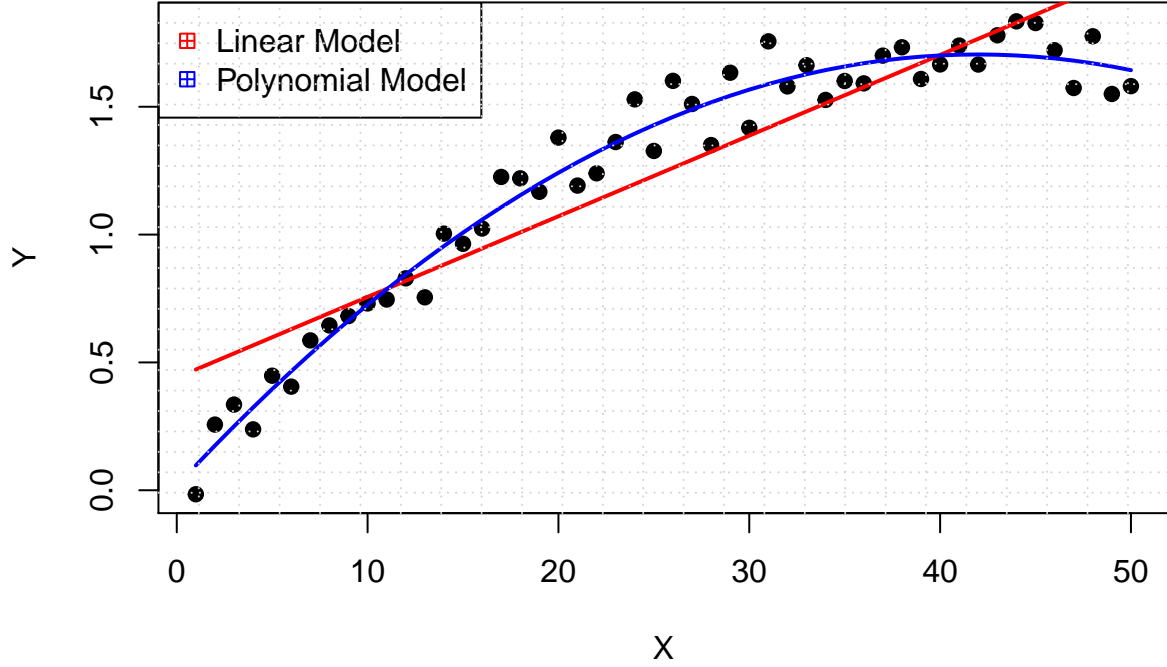
Question 2-Gibbs Sampling

Subquestion 1-Plot Dependence on chemical data

```
load("chemical.Rdata")
```

The above plot shows the dependence of the Y (measured concentration of the chemical) vs X (day of the measurement) for the chemical data given. We have plotted a linear model and a 2nd order polynomial model which seems to be better model for the given data.

```
#linear model
mod1<-lm(Y~X)
#second order polynomial model
mod2<-lm(Y~poly(X,2))
#preds for linear model
Y_lm=predict(mod1)
#preds for poly model2
Y_lm2=predict(mod2)
#plot
plot(X,Y,col="black",pch=19)
lines(X,Y_lm,col="red",lwd=2)
lines(X,Y_lm2,col="blue",lwd=2)
legend("topleft" ,pch=12, col=c("red", "blue"), c("Linear Model", "Polynomial Model"))
grid(25,25,col="lightgray")
```



Subquestion 2-Posterior Prior Bayesian Model

we are given that $Y_i \sim N(\mu_i, \sigma^2)$ $i = 1, 2, \dots, n$ $\vec{\mu} = (\mu_1, \mu_2, \dots, \mu_n)$ $p(\mu_1) = 1$ $p(\mu_{i+1}, \mu_i \sim N(\mu_i, \sigma^2))$ $i = 1, 2, 3, \dots, n-1$

The pdf of the Y_i is given by the formula :

$$f(Y_i/\mu_i, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\left(\frac{(Y_i - \mu_i)^2}{2\sigma^2}\right)}$$

Then the Likelihood is given by :

$$L(Y_1, Y_2, \dots, Y_n/\mu_i, \sigma^2) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\left(\frac{(Y_i - \mu_i)^2}{\sigma^2}\right)}$$

$$p(\vec{Y}/\vec{\mu}) \propto e^{-\left(\sum_{i=1}^n \frac{(Y_i - \mu_i)^2}{\sigma^2}\right)} (1)$$

We now need to derive the prior which is given by

$$p(\vec{\mu}) = p(\mu_1)p(\mu_2/\mu_1)\dots p(\mu_n/\mu_{n-1}) =$$

$$\prod_{i=1}^{n-1} \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\left(\frac{(\mu_{i+1} - \mu_i)^2}{\sigma^2}\right)}$$

$$p(\vec{\mu}) \propto e^{-\left(\sum_{i=1}^{n-1} \frac{(\mu_{i+1} - \mu_i)^2}{\sigma^2}\right)} (2)$$

we know that

$$\text{Posterior} \propto \text{Prior} \cdot \text{Likelihood}$$

or

$$p(\vec{\mu}/\vec{Y}) \propto p(\vec{Y}/\vec{\mu}) \cdot p(\vec{\mu})$$

so we have

$$(1) * (2) = e^{(-\sum_{i=1}^{n-1} \frac{(\mu_{i+1}-\mu_i)^2}{2\sigma^2} - \sum_{i=1}^n \frac{(Y_i-\mu_i)^2}{2\sigma^2})}$$

Subquestion 3-Posterior up to a constant proportionality and conditional probabilities

We know that

$$p(x_j/x_1, x_2, x_3, \dots, x_n) \propto \frac{p(x_1, x_2, x_3, \dots, x_n)}{p(x_1, x_2, \dots, x_{j-1}, x_{j+1}, \dots, x_n)}$$

using the previous formula we are going to find :

$$p(\mu_n/\vec{\mu}_{-n}/\vec{Y}) \propto \frac{p(\vec{\mu}/\vec{Y})}{p(\vec{\mu}_{-n}/\vec{Y})}$$

so we have

$$\begin{aligned} & \frac{e^{(-\sum_{i=1}^{n-1} \frac{(\mu_{i+1}-\mu_i)^2}{2\sigma^2} - \sum_{i=1}^n \frac{(Y_i-\mu_i)^2}{2\sigma^2})}}{e^{(-\sum_{i=1}^{n-2} \frac{(\mu_{i+1}-\mu_i)^2}{2\sigma^2} - \sum_{i=1}^{n-1} \frac{(Y_i-\mu_i)^2}{2\sigma^2})}} = \\ & e^{-\frac{1}{2\sigma^2}[(\mu_n-\mu_{n-1})^2 + (Y_n-\mu_n)^2]} \stackrel{(HINTB)}{=} \\ & e^{-\frac{(\mu_n - (\mu_{n-1} + Y_n)/2)^2}{2\sigma^2/2}} \sim N(\frac{\mu_{n-1}+Y_n}{2}, \frac{\sigma^2}{2}) \end{aligned}$$

Next we need to find

$$p(\mu_i/\vec{\mu}_{-i}/\vec{Y}) \propto \frac{p(\vec{\mu}/\vec{Y})}{p(\vec{\mu}_{-i}/\vec{Y})}$$

so now we have

$$\begin{aligned} & \frac{e^{(-\sum_{i=1}^{n-1} \frac{(\mu_{i+1}-\mu_i)^2}{2\sigma^2} - \sum_{i=1}^n \frac{(Y_i-\mu_i)^2}{2\sigma^2})}}{e^{(-\sum_{j=i}^{i-2} \frac{(\mu_{j+1}-\mu_j)^2}{2\sigma^2} - \sum_{j=i+1}^{n-1} \frac{(\mu_{j+1}-\mu_j)^2}{2\sigma^2} - \sum_{j=1}^{i-1} \frac{(Y_j-\mu_j)^2}{2\sigma^2} - \sum_{j=i+1}^n \frac{(Y_j-\mu_j)^2}{2\sigma^2})}} = \\ & e^{-\frac{1}{2\sigma^2}[(\mu_{i+1}-\mu_i)^2 + (\mu_i-\mu_{i-1})^2 + (Y_i-\mu_i)^2]} \stackrel{(HINTC)}{=} \\ & e^{-\frac{(\mu_i - (\mu_{i-1} + Y_i + \mu_{i+1})/3)^2}{2\sigma^2/3}} \end{aligned}$$

Using the same steps we obtain also that

$$p(\mu_1/\vec{\mu}_{-1}/\vec{Y}) \propto e^{-\frac{1}{2\sigma^2}[(\mu_2-\mu_1)^2 + (Y_1-\mu_1)^2]} \propto e^{-\frac{(\mu_1 - (\mu_2 + Y_1)/2)^2}{2\sigma^2/2}}$$

Subquestion 4-Sampling with Gibbs

```
#gibbs sampling function
f.MCMC.Gibbs <- function(nstep,X0,sd){

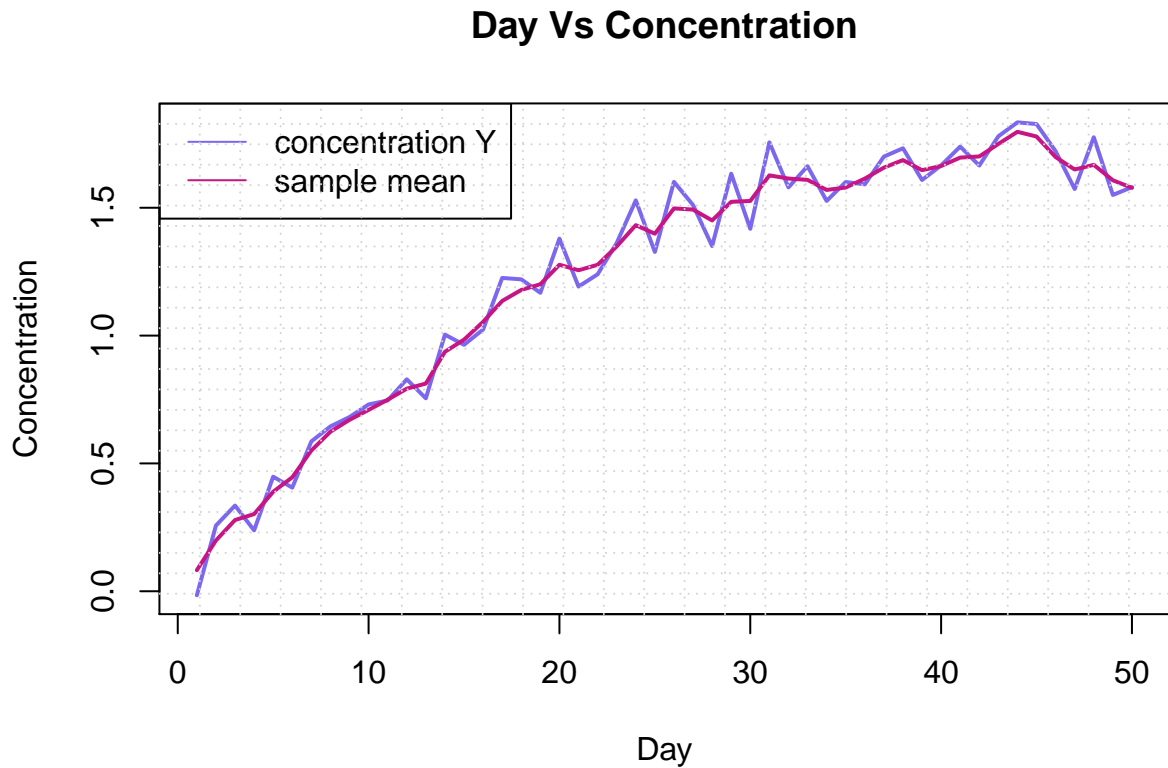
  mX<-matrix(0,nrow=length(X0),ncol=nstep)

  for (t in 1:(nstep-1)){
    for (i in 1:length(X0)){
      if (i==1){
        y <- (mX[i+1,t]+X0[i])/2
        mX[i,t+1] <- rnorm(1,y,sd/sqrt(2))
      }
      else if (i ==50){
        y <- (mX[i-1,t+1]+X0[i])/2
        mX[i,t+1] <- rnorm(1, y, sd/sqrt(2))
      }
      else{
        y <- (mX[i+1,t]+ mX[i-1,t+1] +X0[i])/3
        mX[i,t+1] <- rnorm(1,y,sd/sqrt(3))
      }
    }
  }
  return(mX)
}
```

Plot sampling with Gibbs

```
#create df to store the loaded data
chemical<- data.frame(X,Y)
X0<-chemical$Y
#call function to get sample
gib <- f.MCMC.Gibbs(1000, X0, sqrt(0.2))

#plot of expected value gib vs X and Y vs X
plot(chemical$X, chemical$Y, main="Day Vs Concentration", col="mediumslateblue",
     xlab="Day", ylab="Concentration", type="l",lwd=2)
points(chemical$X,rowMeans(gib), col="mediumvioletred", type="l",lwd=2)
legend("topleft", legend=c("concentration Y","sample mean"), col=c("mediumslateblue","mediumvioletred")
grid(25,25)
```



From the above plot we can conclude that the observed value of $\vec{\mu}$ seems to follow the pattern of dependence between Y and X.

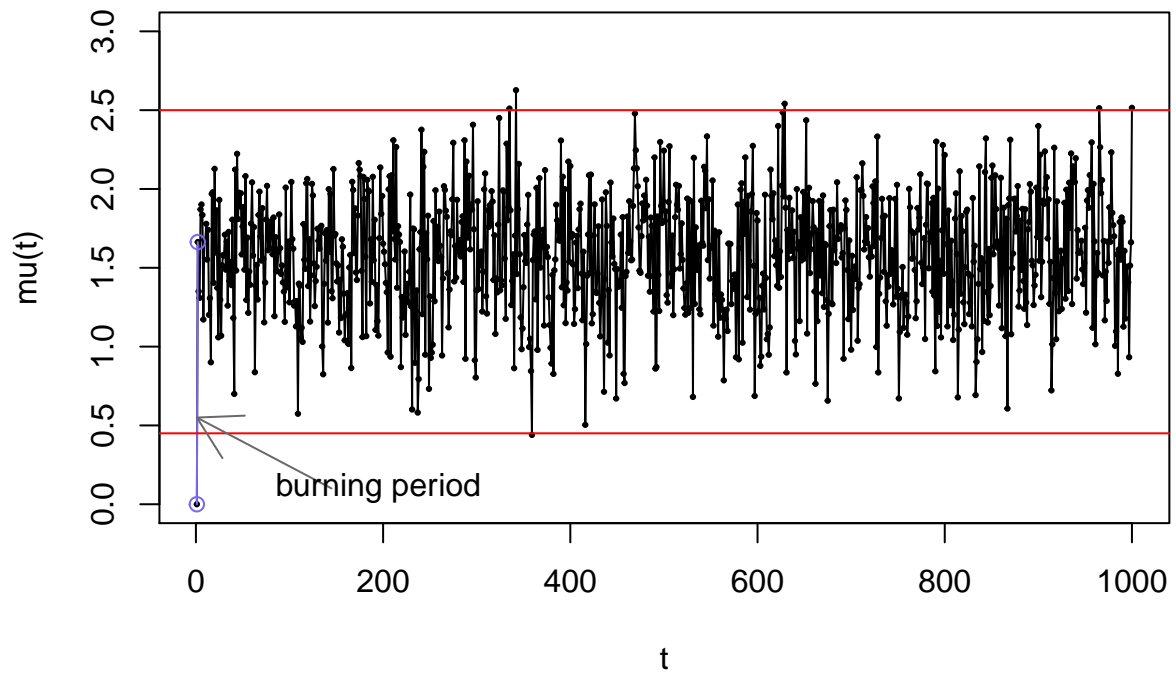
Subquestion 5-Trace Plot

Trace plot

```
vN<-1:length(gib[50,])
plot(vN,gib[50,],pch=19,cex=0.3,col="black",xlab="t",ylab="mu(t)",
     main="Trace Plot",ylim=c(0,3),type="o")
points(gib[50,1:2],col="mediumslateblue",type="o")
abline(h=0.45,col="red")
abline(h=2.5,col="red")

arrows(145,0.1,1,0.55,col="dimgrey")
text(195,0.1,c("burning period"))
```

Trace Plot



The plot shows the chain for the last μ_{50} with blue we have 2 observations that seem not follow the overall pattern and we may consider them as burning period and we can discard them from overall plot.