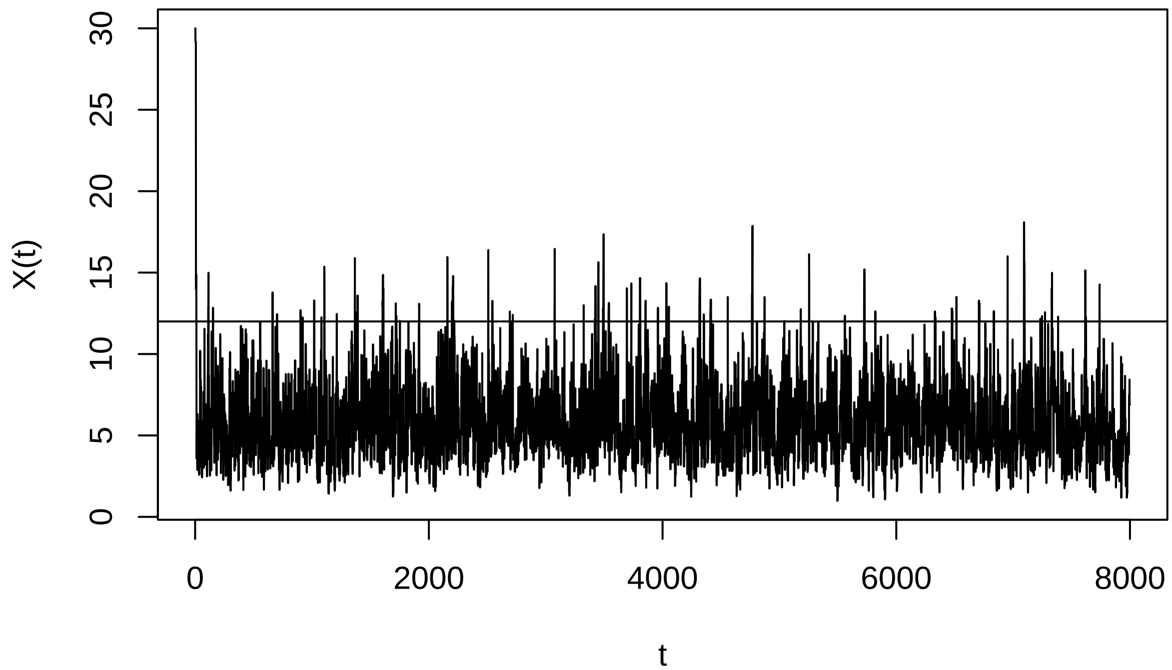
lab04

Shipeng Liu, Dongwei Ni

2022-12-06

Task 1:Metropolis–Hastings Sampler

```
mcq1 <- function(X0_ele, n, sdlog_in){

  X0 <- rep(X0_ele, n)
  pdf_pi <- \(x) if (x > 0) {
    (x^5 * exp(-x) / 120)
    } else {NA}

  for (i in 1:(n-1)) {
    X <- X0[i]
    Y <- rlnorm(1, meanlog = log(X), sdlog = sdlog_in)
    U <- runif(1)
    alpha <- min(1, pdf_pi(Y)*dlnorm(X, meanlog = log(Y), sdlog = sdlog_in) /
                   (pdf_pi(X)*dlnorm(Y, meanlog = log(X), sdlog = sdlog_in)))
    if (U < alpha) {
      X0[i+1] <- Y
    } else{
      X0[i+1] <- X0[i]
    }
  }
  return(X0)
}

X1 <- mcq1(30, 8000, 0.5)
plot(1:length(X1), X1, type = "l", xlab = "t", ylab = "X(t)")
abline(h = 12)
```

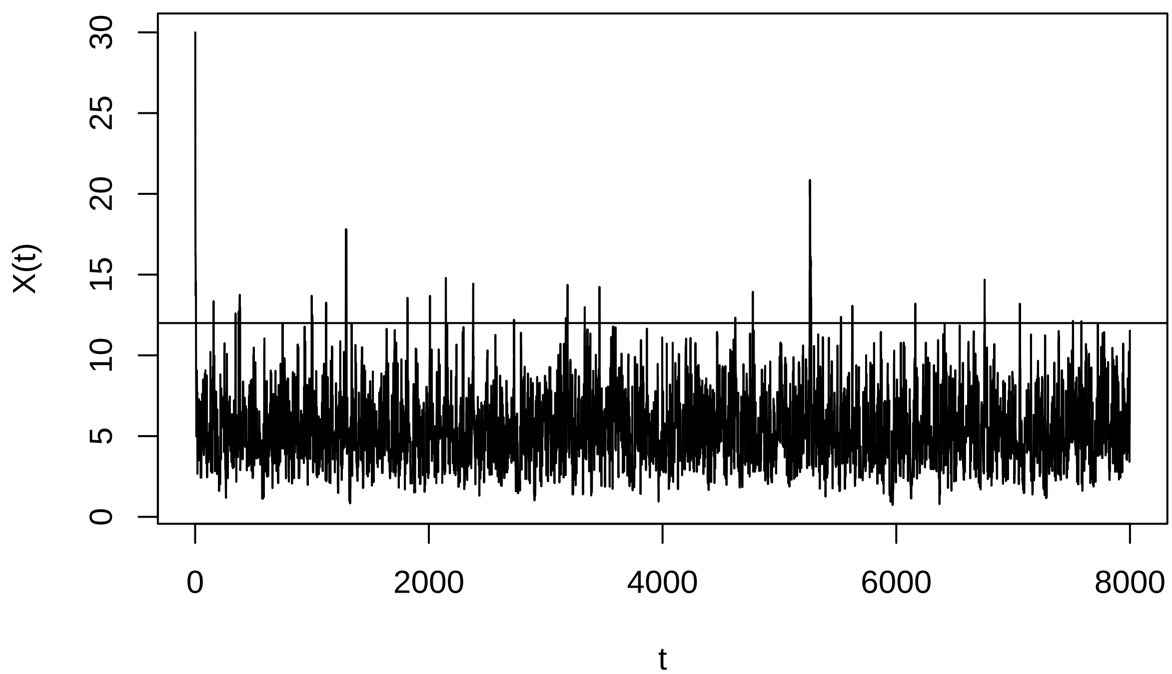


- As we can see the plot converges with only few points out of (0,12) after burn-in period. And from the plot it shows that the burn-in period can be very small because it converges rapidly.

Task 2:Sampling by chi–square distribution

```
mcq2 <- function(X0_ele, n){

  X0 <- rep(X0_ele, n)
  pdf_pi <- \(x) if (x > 0) {
    (x^5 * exp(-x) / 120)
    } else {NA}

  for (i in 1:(n-1)) {
    X <- X0[i]
    Y <- rchisq(1, df = (X) )
    U <- runif(1)
    alpha <- min(1, pdf_pi(Y)*dchisq(X, df = floor(Y+1)) /
                   (pdf_pi(X)*dchisq(Y, df =floor(X+1))))
    if (U < alpha) {
      X0[i+1] <- Y
    } else{
      X0[i+1] <- X0[i]
    }
  }
  return(X0)
}
```

```r
X2 <- mcq2(30, 8000 )
plot(1:length(X2), X2, type = "l", xlab = "t", ylab = "X(t)")
abline(h = 12)
```



Task 3:Compare the result of Step 1 and 2 and make conclusion

- From the plot we can see that both log-normal and chi-square converges. Seems there is no obvious differences between two methods. Both methods seems can be chose equivalently to use.

Task 4:Gelman–Rubin method to analyze convergence

```r
f1=mcmc.list()

for(i in 1:10){
  f1[[i]]=as.mcmc(mcq2(i, 8000 ))
}
print(gelman.diag(f1))
```

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

- The Gelman–Rubin factor is around 1,which means the chain has converged.

Task 5:Estimate Integral

We want to estimate integral$\int_0^\infty xf(x)dx$

where $\int_0^\infty f(x)dx = 1$

Then,if $X \sim f(x)$:

$$\int_0^\infty xf(x)dx = E[X]$$

Estimate:

$$\overline{E}[X] = \frac{1}{n}\sum_{i=1}^{n} x_i, \forall_i x_i \sim f(\mathring{u})$$

So,using the sampling in step 1,we get:

```
mean(X1)
```

```
## [1] 5.945337
```

Using the sampling in step 2,we get:

```
mean(X2)
```

```
## [1] 5.322215
```

Task 6:Actual integral of gamma distribution

From wikipedia:https://en.wikipedia.org/wiki/Gamma_distribution

The expection of gamma distribution:

$$E[X] = k\theta$$

So,

$$\int_0^\infty xf(x)dx = E[X] = k\theta = 6$$

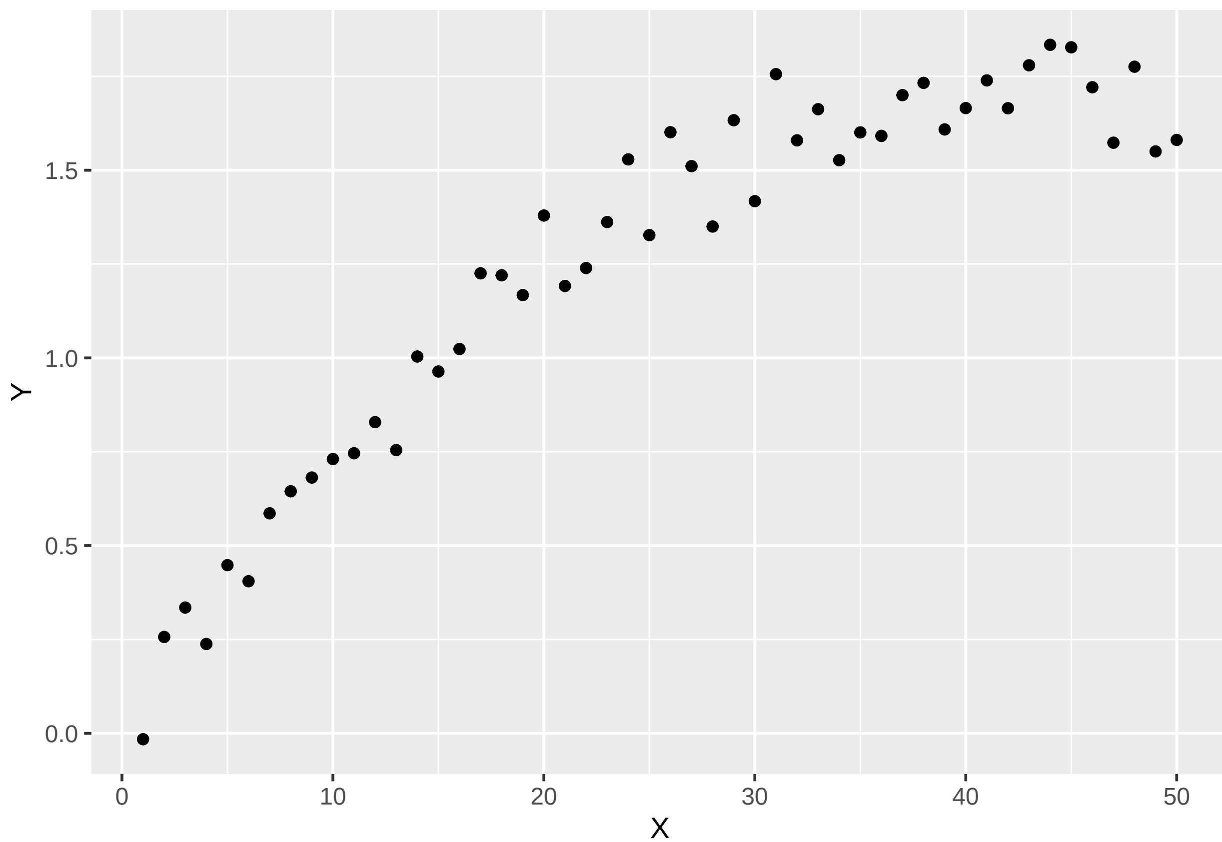
where $k = 6$ and $\theta = 1$.

Our estimate value is very close to the actual integral of gamma distribution.

Assignment 2:Gibbs Sampling

Task 1:The dependence of Y on X

```
load("chemical.RData")
chemical=data.frame("X"=X,"Y"=Y)
ggplot(data=chemical,aes(x=X,y=Y))+geom_point()
```



From the point of view of the distribution of sample points, it is more reasonable to use the logarithmic model.

Task 2:Likelihood and Prior

Likelihood $p(\vec{Y}|\vec{\mu})$**:**

$$
\begin{aligned}
p(\vec{Y}|\vec{\mu}) &= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i-\mu_i)^2}{2\sigma^2}} \\
&= (\frac{1}{\sigma\sqrt{2\pi}})^n e^{-\sum_{i=1}^{n} \frac{(y_i-\mu_i)^2}{2\sigma^2}}
\end{aligned}
$$

where $\sigma^2 = 0.2$

Prior$p(\vec{\mu})$**:**

$$
\begin{aligned}
p(\vec{\mu}) &= p(\mu_1)\mathring{u}p(\mu_2|\mu_1)\mathring{u}\ldots\mathring{u}p(\mu_n|\mu_{n-1}) \\
&= 1\mathring{u}\prod_{i=1}^{n-1} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\mu_{i+1}-\mu_i)^2}{2\sigma^2}} \\
&= (\frac{1}{\sigma\sqrt{2\pi}})^{n-1} e^{-\sum_{i=1}^{n-1} \frac{(\mu_{i+1}-\mu_i)^2}{2\sigma^2}}
\end{aligned}
$$

where $\sigma^2 = 0.2$

Task 3:Posterior,conditional marginal distribution

Posterior$p(\vec{\mu}|\vec{Y})$:

$$
\begin{aligned}
p(\vec{\mu}|\vec{Y}) &\propto P(\vec{Y}|\vec{\mu})P(\vec{\mu}) \\
&\propto exp\left[-\sum_{i=1}^{n}\frac{(y_i-\mu_i)^2}{2\sigma^2}\right]exp\left[-\sum_{i=1}^{n-1}\frac{(\mu_{i+1}-\mu_i)^2}{2\sigma^2}\right] \\
&\propto exp\left[-\frac{1}{2\sigma^2}\left(\sum_{i=1}^{n-1}\left[(\mu_i-\mu_{i+1})^2+(\mu_i-y_i)^2\right]+(\mu_n-y_n)^2\right)\right]
\end{aligned}
$$

Consider $p(\mu_1|\vec{\mu_{-1}},\vec{Y})$:

$$
p(\mu_1|\vec{\mu_{-1}},\vec{Y}) \propto exp\left[-\frac{1}{2\sigma^2}\Big((\mu_1-\mu_2)^2+(\mu_1-y_1)^2\Big)\right]
$$

From Hint B we get:

$$
p(\mu_1|\vec{\mu_{-1}},\vec{Y}) \propto exp\left[-\frac{1}{\sigma^2}\left(\mu_1-\frac{(\mu_2+y_1)}{2}\right)^2\right] \sim N\left(\frac{\mu_2+y_1}{2},\frac{\sigma^2}{2}\right)
$$

Consider $p(\mu_n|\vec{\mu_{-n}},\vec{Y})$:

$$
p(\mu_n|\vec{\mu_{-n}},\vec{Y}) \propto exp\left[-\frac{1}{2\sigma^2}\Big((\mu_{n-1}-\mu_n)^2+(\mu_n-y_n)^2\Big)\right]
$$

From Hint B we get:

$$
p(\mu_n|\vec{\mu_{-n}},\vec{Y}) \propto exp\left[-\frac{1}{\sigma^2}\left(\mu_n-\frac{(\mu_{n-1}+y_n)}{2}\right)^2\right] \sim N\left(\frac{\mu_{n-1}+y_n}{2},\frac{\sigma^2}{2}\right)
$$

Finally we consider $p(\mu_i|\vec{\mu_{-i}},\vec{Y})$:

$$
p(\mu_i|\vec{\mu_{-i}},\vec{Y}) \propto exp\left[-\frac{1}{2\sigma^2}\Big((\mu_{i-1}-\mu_i)^2+(\mu_i-\mu_{i+1})^2+(\mu_i-y_i)^2\Big)\right]
$$

From Hint C we get:

$$
\begin{aligned}
p(\mu_i|\vec{\mu_{-i}},\vec{Y}) &\propto exp\left[-\frac{1}{2\sigma^2}\Big((\mu_i-\mu_{i-1})^2+(\mu_i-\mu_{i+1})^2+(\mu_i-y_i)^2\Big)\right] \\
&\propto exp\left[-\frac{2\sigma^2}{3}\left(\mu_i-\frac{(\mu_{i-1}+\mu_{i+1}+y_i)}{3}\right)^2\right] \sim N\left(\frac{\mu_{i-1}+\mu_{i+1}+y_i}{3},\frac{\sigma^2}{3}\right)
\end{aligned}
$$

Task 4:Implement a Gibbs sampler

```
Gibbs=function(data,t_max){
  n=nrow(data)  #
  record=matrix(0,nrow=t_max,ncol = n)
  t=1
  while(t<=t_max){
    for(i in 1:n){
      if(i==1){
```

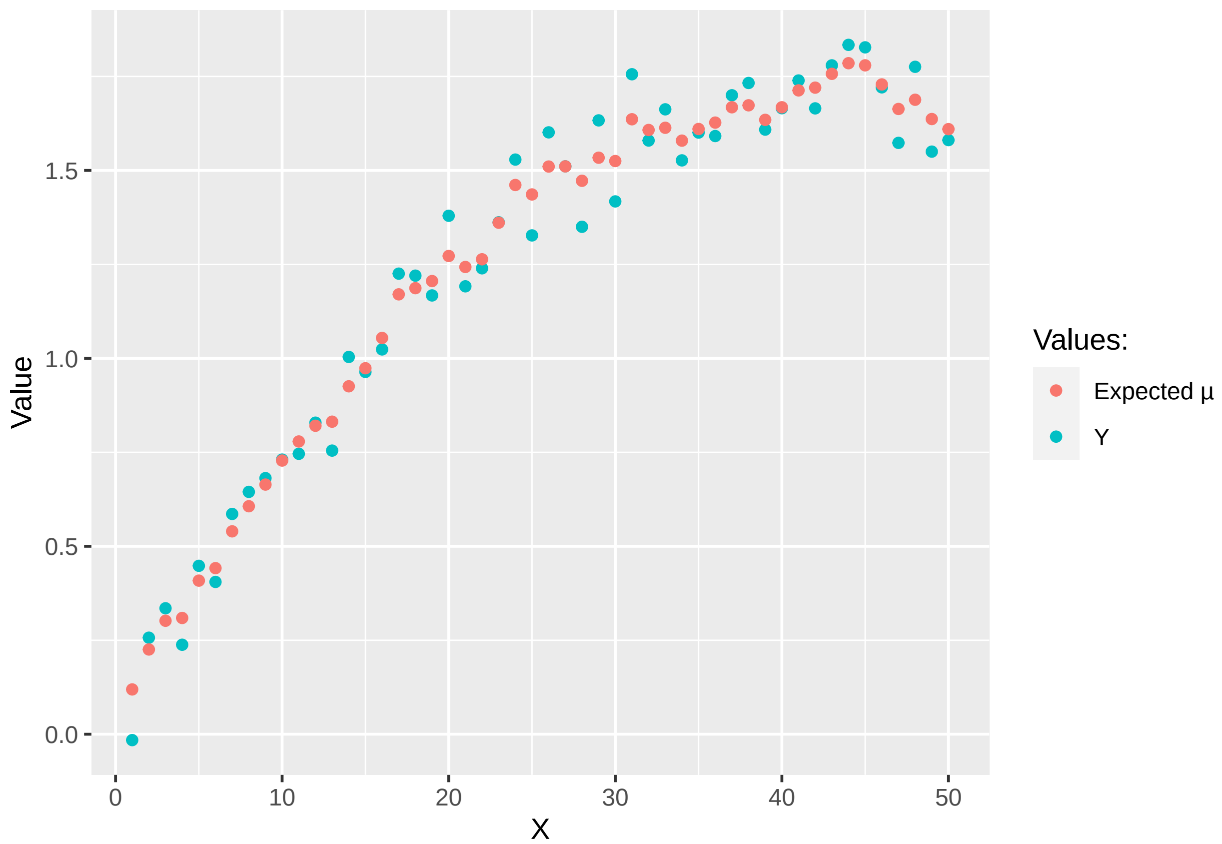
```
        record[t,i]=rnorm(1,(record[t,2]+data[1,2])/2,sqrt(0.2/2))
      }else if(i>1 && i<n){
        record[t,i]=rnorm(1,(record[t,i-1]+
                               record[t,i+1]+data[i,2])/3,sqrt(0.2/3))
      }else if(i==n){
        record[t,i]=rnorm(1,(record[t,n-1]+data[n,2])/2,sqrt(0.2/2))
      }
    }
    if(t<t_max){
      record[t+1,]=record[t,]
    }
    t=t+1
  }
  return(record)
}

samp=Gibbs(chemical,1000)
```

The expected value of µ by Monte Carlo approach .

The expected µ value:

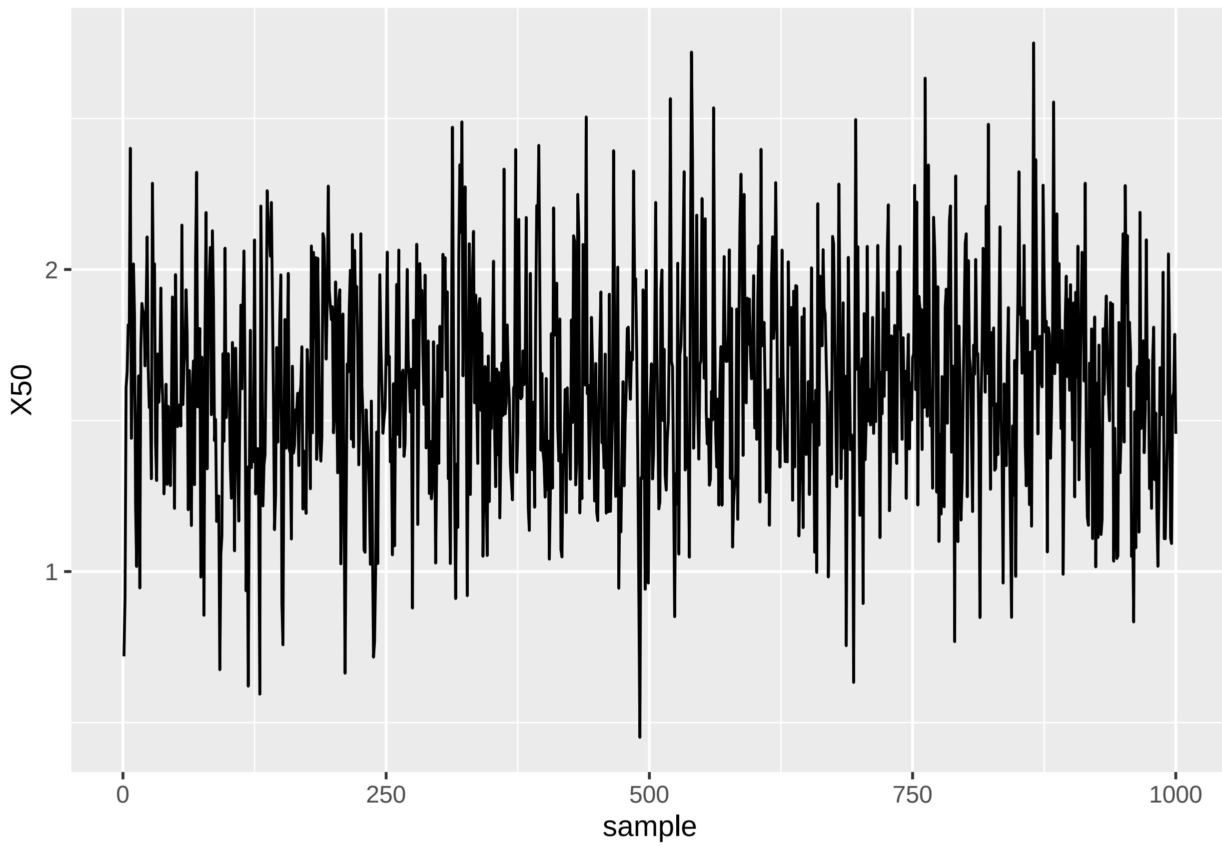
```
##  [1] 0.1191981 0.2254603 0.3018659 0.3093051 0.4086567 0.4418913 0.5397863
##  [8] 0.6065880 0.6642520 0.7281348 0.7788594 0.8206546 0.8315125 0.9256974
## [15] 0.9737241 1.0542855 1.1703122 1.1867794 1.2056360 1.2723502 1.2431176
## [22] 1.2636001 1.3608769 1.4612306 1.4360739 1.5103229 1.5107746 1.4723904
## [29] 1.5338236 1.5249507 1.6361363 1.6075854 1.6134896 1.5792957 1.6102311
## [36] 1.6272486 1.6679692 1.6732342 1.6345874 1.6680121 1.7129322 1.7202624
## [43] 1.7569764 1.7852377 1.7796140 1.7287580 1.6634911 1.6880269 1.6367250
## [50] 1.6098756
```

Plot the expected value of µ versus X and Y versus X in the same graph .



It seems some of the noise have been removed,and the expected value of μ can catch the true underlying dependence between Y and X.

Task 5:Converge



Trace plot for μ_n .

From the first sampling to the last time, μ_n always oscillates in a range, which means that this Markov chain reaches a stable condition and convergence at the beginning, and it has no Burn-in period