Computational Statistics lab4_group1

Vasileia Kampouraki, Weng Hang Wong, Jooyoung Lee

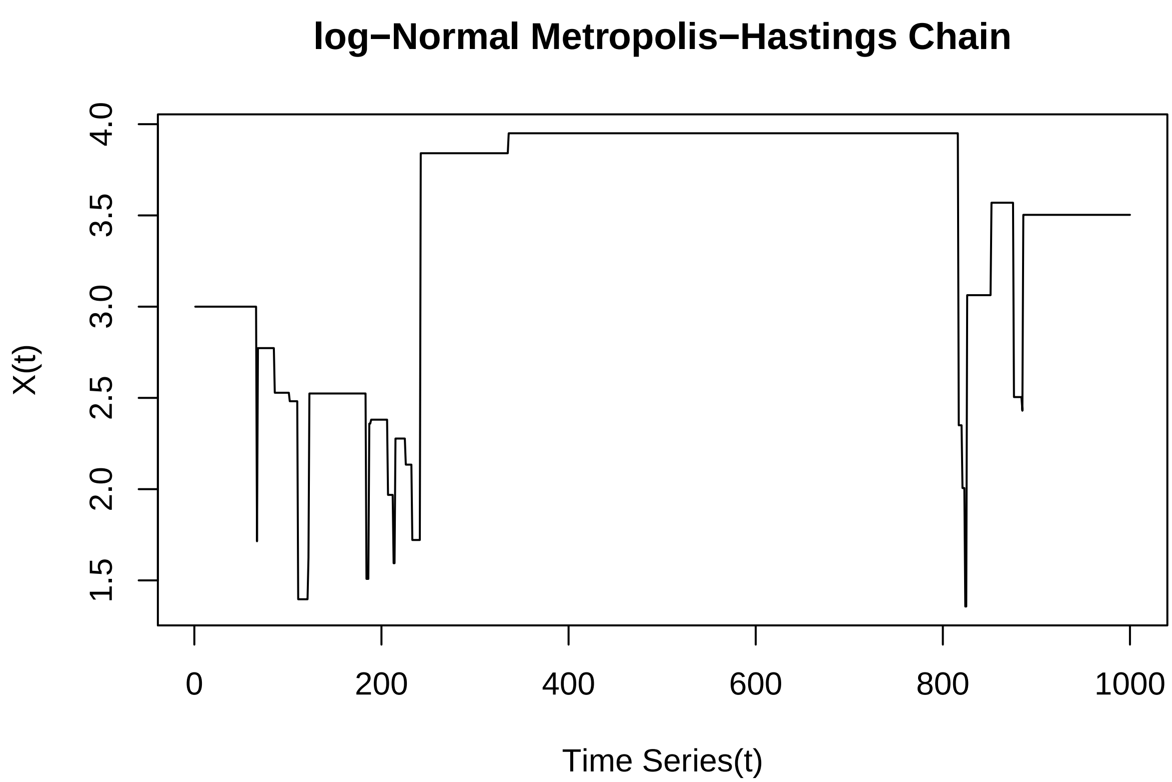
2/20/2020

Question 1: Computations with Metropolis-Hastings

1. Use Metropolis-Hastings algorithm to generate samples from this distribution by using proposal distribution as log-normal LN(Xt,1), take some starting point. Plot the chain you obtained as a time series plot. What can you guess about the convergence of the chain? If there is a burn-in period, what can be the size of this period?

Our target distribution that we want to sample from is $\pi(x) \propto x^5 e^{-x}$, by using log-Normal $LN(X_t, 1)$ as proposal distribution.

```
## Warning in RNGkind("Mersenne-Twister", "Inversion", "Rounding"): non-
## uniform 'Rounding' sampler used
```



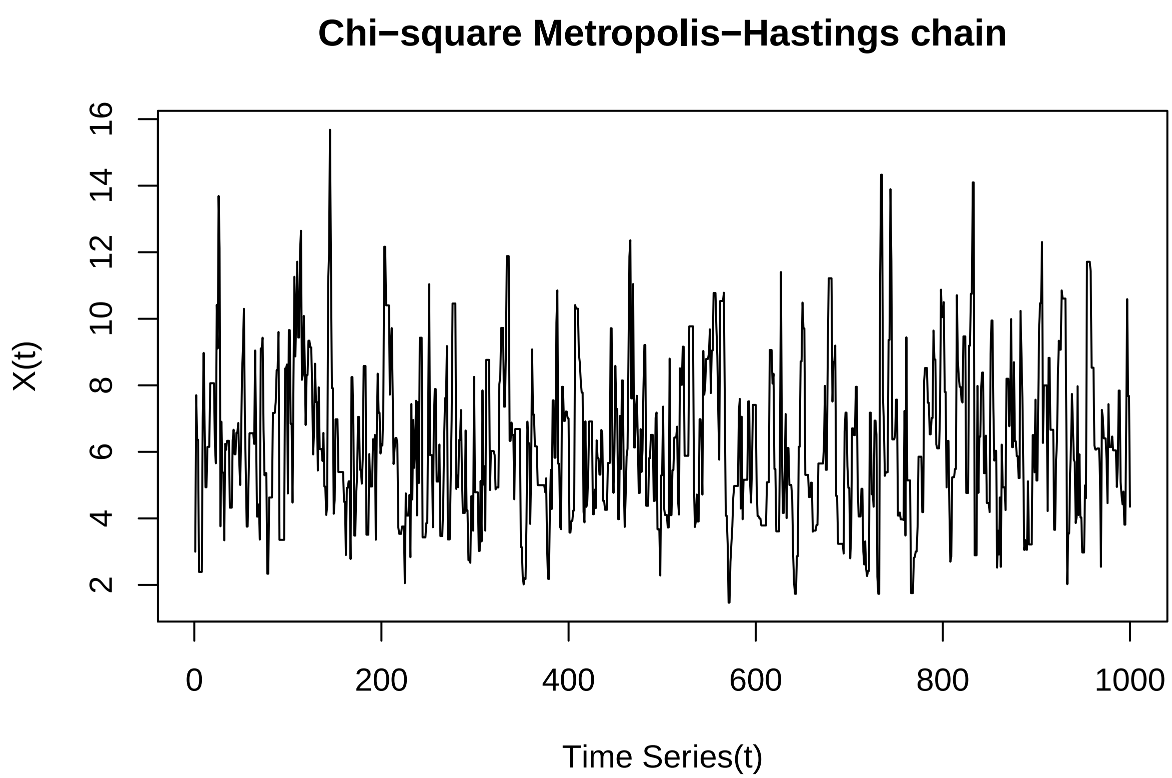
As the figure above, we run the algorithm for 1000 times and look at the plot of the chain obtained, as a time series plot. We chose $X_0 = 3$ as our, arbitrary, starting point. The basic idea of the algorithm is to simulate a Markov Chain with stationary distribution, the "target"-distribution ($\pi(x)$). However, the X_0 is chosen arbitrarily so we don't know how quickly the chain is going to stabilize.This is actually the burn-in period of the chain, and thus we can discard the samples from this period and keep those that have been stabilized.

In our case, looking at the time series plot of the chain we could say that at the beginning we have a good image of the chain and the burn-in period seems almost insignificant so we could probably say that there's no need to reject part of the sample at the beginning. However, we can observe some periods where the algorithm constantly rejects values of y and the chain remains stable at the same point, as it happens for example at around t=400 to t=800.

In the end the chain, we can see we couldn't see it converges to the target distribution ($\pi(x) \propto x^5 e^{-x}$) at t=1000

2

Repeat the Step 1 by suing the chi-square distribution $\chi^2([X_t + 1])$ as proposal distribution.



Now we changed our proposal distribution from $LN(X_t, 1)$ to $\chi^2(\lfloor X_t + 1 \rfloor)$. After plotting the chain of the target distribution we can conclude that the result is actually very good. The burn-in period is very small, we would say insignificant and the chain stabilizes quickly and stay stabilized throughout the whole process. We can see quite high oscillations in the state space but the algorithm seems to accept generally the values of Y as we can't observe periods where the chain remained stable at the same distribution (i.e. being flat for a while). Finally, the chain seems to converge to the stationary distribution π.

3. Compare the results of Steps 1 and 2 and make conclusions.

From the above result, we can say the result seems better when we use the chi-square $\chi^2(\lfloor X_t + 1 \rfloor$ as a proposal distribution. Since the chain is more likely to be converged and more stable, and it doesn't has some value stucked in the chain as the log-normal distribution one.

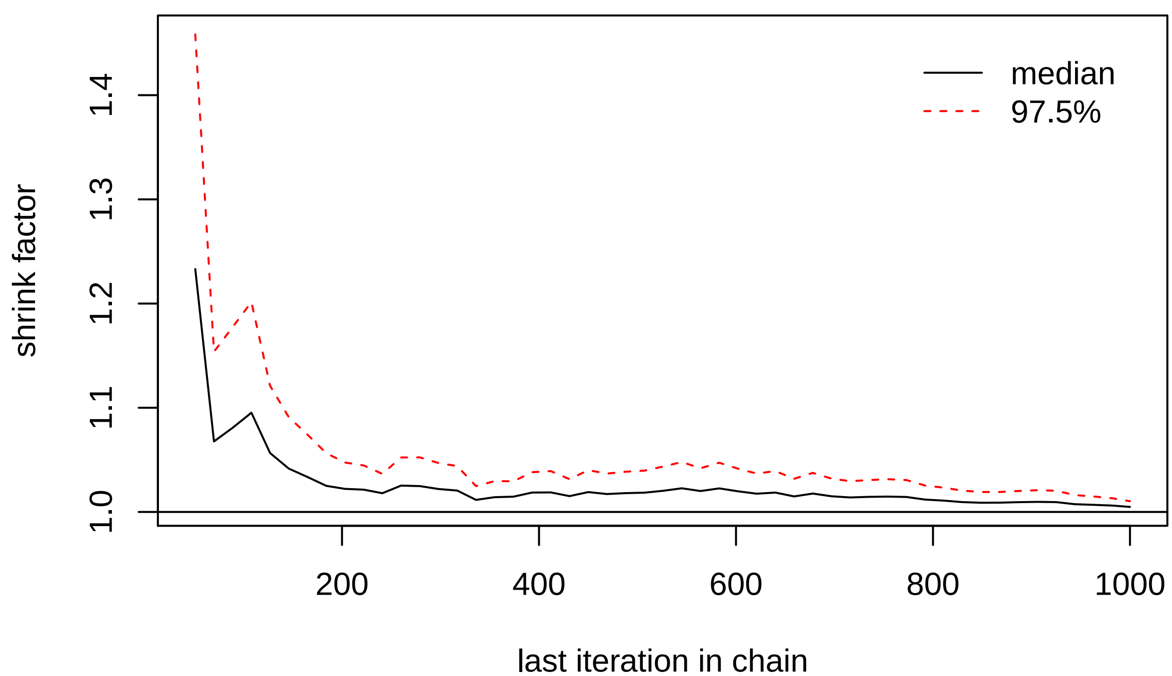
4. Use the Gelman–Rubin method to analyze convergence of these sequences.

We generate 10 MCMC sequences using the $\chi^2(\lfloor X_t + 1 \rfloor$ as proposal distribution from above. Then we use Gelman-Rubin method to analyze the convergence of the chain. *gelman.diag()* function compares the variability within the chains to the variability between the chains. If the Upper C.I is very close to 1 then we can say with 95% confidence that the chains have converged, otherwise values much larger than 1 is an indication of non-convergence.

In this case, the upper C.I has the value of 1.02 and it is very close to 1. Therefore, with using Chi-square as proposal distribution has a good convergence result that we can say we reach to the stationary distribution.

```
## The Gelman-Rubin converage anaylsis:

## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1       1.01
```



5

To estimate $\int_0^\infty x f(x) dx$, we can use the mean of the samples generated from the step1 and step2.

```
## The samples mean from Step 1 (Log-likelihood)

## [1] 3.272259

## The samples mean from Step 2 (chi-sqaure)

## [1] 5.950549
```

6.

The probability distribution function of a Gamma(a,b) is

$$f_x(x) = \frac{\beta^a}{\Gamma(a)} x^{a-1} e^{-\beta x}$$

for x>0 and $\alpha, \beta > 0$. So in our case, we have a Gamma(6,1) and thus the integral we want to compute is

$$\frac{1}{\Gamma(6)} \int_0^\infty x^6 e^{-x} dx$$

.

```
## The real value of the integral is:

## 6 with absolute error < 3.9e-05
```

Compare the value we obtained from the above step 1 and step 2, we can see that the real value only close to the value we estimated from the chi-sqaure as proposal distribution, which is 5.950549.

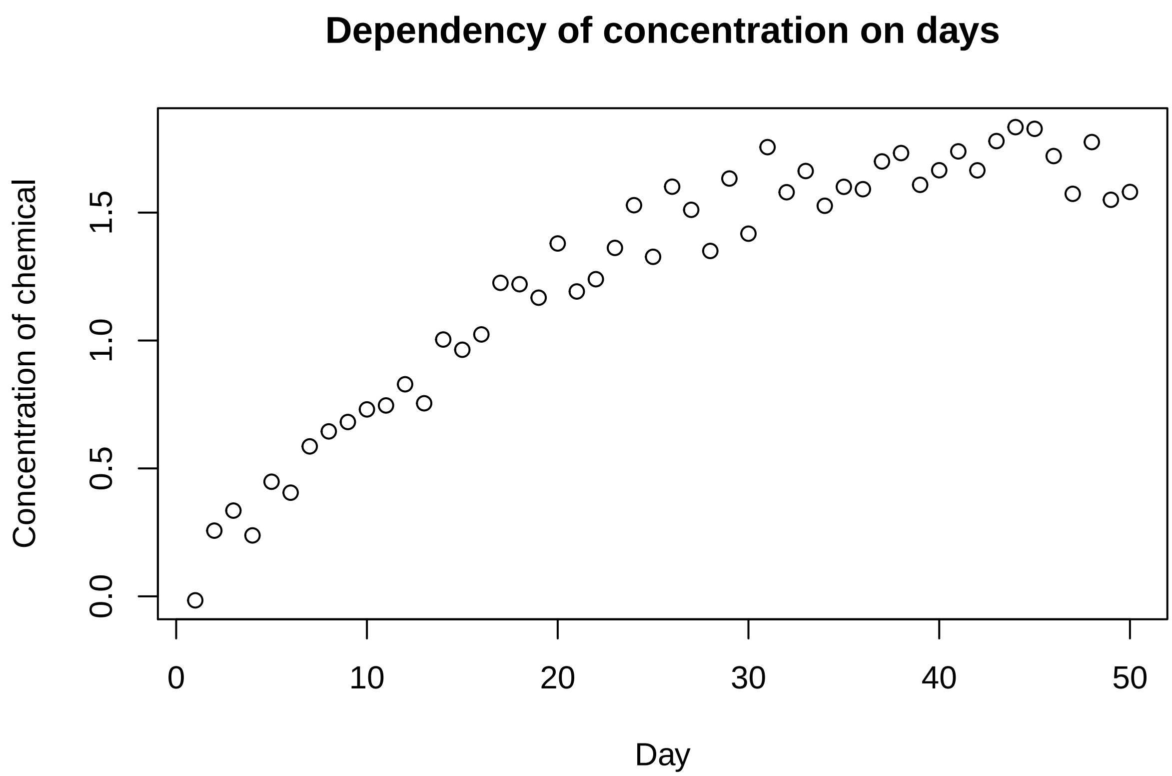
Question 2: Gibbs sampling

A concentration of a certain chemical was measured in a water sample, and the result was stored in the data *chemical.RData* having the following variables:

- X: **day of the measurement**
- Y: **measured concentration of the chemical.**

The instrument used to measure the concentration had certain accuracy; this is why the measurements can be treated as noisy. Your purpose is to restore the expected concentration values.

1. Import the data to R and plot the dependence of Y on X. What kind of model is reasonable to use here?



Looking at the way the data are on the plot we could assume that a logarithmic model could probably fit well those data.

2. A researcher has decided to use the following (random-walk) Bayesian model (n=number of observations,$\vec{\mu} = \mu_1, \mu_2, ..., \mu_n$ are unknown parameters):

Yi $\sim N(\mu_i, variance = 0.2)$ **i = 1,. . . ,n**

where the prior is

$$p(\mu_1) = 1$$

$$p(\mu_{i+1}|\mu_i) = N(\mu_i, 0.2)$$

Present the formulae showing the likelihood $P(\vec{Y}|\vec{\mu})$ and the prior $p(\vec{\mu})$. Hint: a chain rule can be used here $p(\vec{\mu}) = p(\mu_1)p(\mu_2|\mu_1)p(\mu_3|\mu_2)...p(\mu_n|\mu_{n_1})$.

Yi $\sim N(\mu_i, variance = 0.2)$, so the likelihood will be:

Likelihood:

$$P(\vec{Y}|\vec{\mu}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi 0.2}} e^{-\frac{1}{2\cdot 0.2}(y-\mu)^2} \propto e^{-\frac{1}{0.4}\sum_{i=1}^{n}(y_i-\mu_i)^2}$$

For the **prior** probability $p(\vec{\mu})$ we will use the chain rule.

$$p(\mu_1) = 1$$

$$p(\mu_2|\mu_1) \sim N(\mu_1, 0.2) = \frac{1}{\sqrt{2\pi 0.2}} e^{-\frac{1}{2\cdot 0.2}(\mu_2-\mu_1)^2}$$

$$p(\mu_3|\mu_2) \sim N(\mu_2, 0.2) = \frac{1}{\sqrt{2\pi 0.2}} e^{-\frac{1}{2\cdot 0.2}(\mu_2-\mu_3)^2} \vdots$$

$$p(\mu_n|\mu_{n_1}) \sim N(\mu_{n_1}, 0.2) = \frac{1}{\sqrt{2\pi 0.2}} e^{-\frac{1}{2\cdot 0.2}(\mu_{n_1}-\mu_n)^2}$$

Thus, using the chain rule and combining all those equations we get that the prior is:

$$p(\vec{\mu}) = p(\mu_1)p(\mu_2|\mu_1)p(\mu_3|\mu_2)...p(\mu_n|\mu_{n_1}) \propto e^{-\frac{\sum_{i=2}^{n}(\mu_i-\mu_{i-1})^2}{0.4}}$$

3. Use Bayes' Theorem to get the posterior up to a constant proportionality, and then find out the distributions of $(\mu_i|\vec{\mu_{-i}}$, where $\vec{\mu_{-i}}$ is a vector containing all μ values except of μ_i.

Based on the Bayes' Theorem **Posterior probability $\propto$ Likelihood x Prior probability**

From the previous question we have already computed the Likelihood $P(\vec{Y}|\vec{\mu})$ and the prior probability p($\vec{\mu}$).

We are asked to find the distributions of $(\mu_i|\vec{\mu_{-i}}, \vec{Y})$ where $\vec{\mu_{-i}}$ is a vector containing all μ values except of μ_i.

So, to find the distributions we first have to find the posterior probabilities and then look at their formula and see from which distribution they come from.

Posterior probability

$$P(\mu_i|\vec{\mu_{-i}}, \vec{Y}) \propto P(\vec{Y}|\vec{\mu})P(\vec{\mu})$$

For μ_1**:**

$$P(\mu_1|\vec{\mu_{-1}}, \vec{Y}) \propto e^{-\frac{1}{2\cdot 0.2}\{(y_1-\mu_1)^2+(\mu_2-\mu_1)^2\}} = e^{-\frac{1}{\frac{0.4}{2}}\{\mu_1-\frac{y_1+\mu_2}{2}\}^2}$$

Thus,

$$\mu_1|\vec{\mu_{-1}} \sim N(\frac{y_1+\mu_2}{2}, 0.1)$$

For μ_i ,i=2,...,n-1:

$$P(\mu_i|\vec{\mu_{-i}}, \vec{Y}) \propto e^{-\frac{1}{2\cdot 0.2}\{(y_i-\mu_i)^2+(\mu_i-\mu_{i-1})^2+(\mu_{i+1}-\mu_i)^2\}} = e^{-\frac{1}{\frac{0.4}{3}}\{\mu_i-\frac{y_i+\mu_{i-1}+\mu_{i+1}}{3}\}^2}$$

Thus,

$$\mu_i|\vec{\mu_{-i}} \sim N(\frac{y_i + \mu_{i-1} + \mu_{i+1}}{3}, 0.06)$$

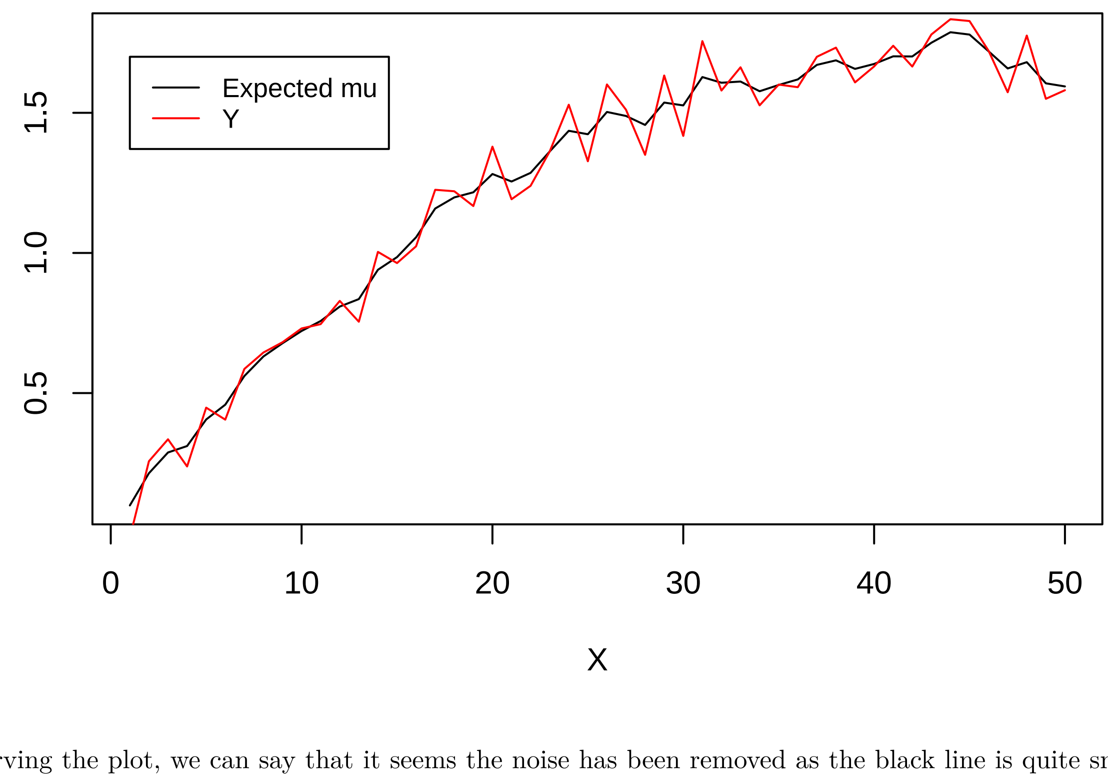
For μ_n:

$$P(\mu_n|\vec{\mu_{-n}}, \vec{Y}) \propto e^{-\frac{1}{2 \cdot 0.2}\{(y_n - \mu_n)^2 + (\mu_n - \mu_{n-1})^2\}} = e^{-\frac{1}{\frac{0.4}{2}}\{\mu_n - \frac{y_n + \mu_{n-1}}{2}\}^2}$$

Thus,

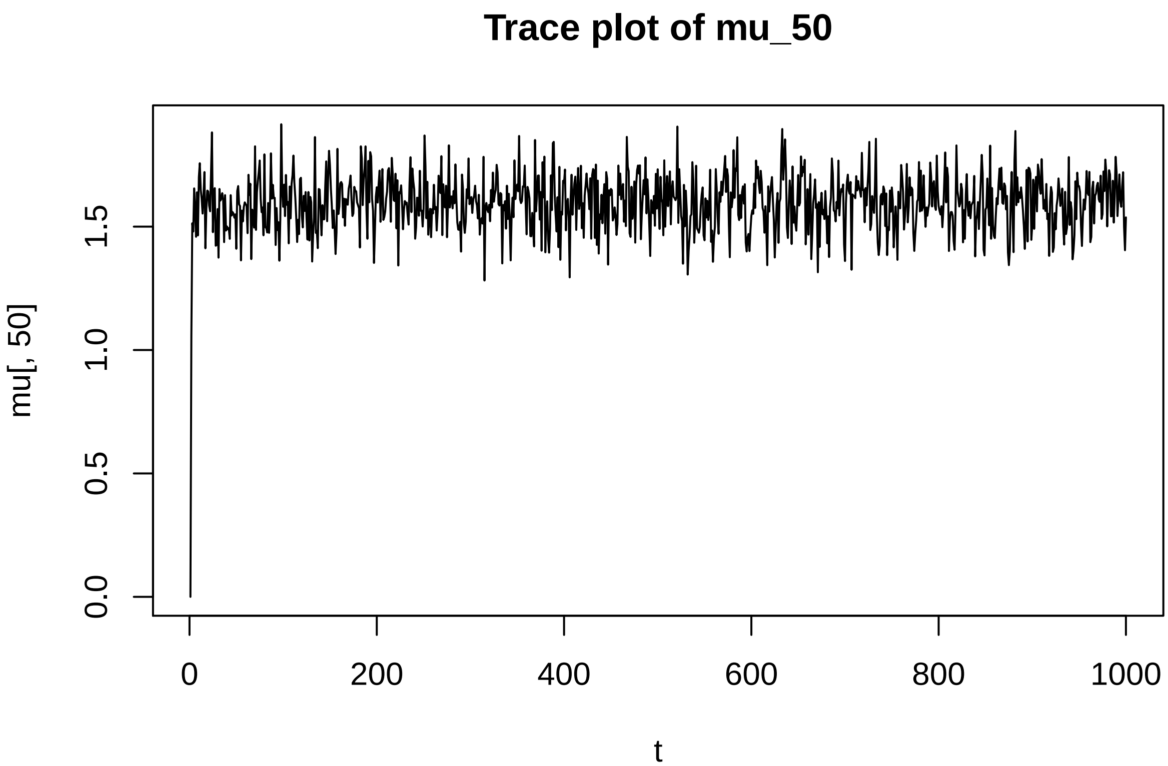
$$\mu_n|\vec{\mu_{-n}} \sim N(\frac{y_n + \mu_{n-1}}{2}, 0.1)$$

4. Use the distributions derived in Step 3 to implement a Gibbs sampler that uses $\vec{\mu}^0 =$ (0,...,0) as a starting point. Run the Gibbs sampler to obtain 1000 values of $\vec{\mu}$ and then compute the expected value of $\vec{\mu}$ by using a Monte Carlo approach. Plot the expected value of $\vec{\mu}$ versus X and Y versus X in the same graph. Does it seem that you have managed to remove the noise? Does it seem that the expected value of $\vec{\mu}$ can catch the true underlying dependence between Y and X?



Observing the plot, we can say that it seems the noise has been removed as the black line is quite smooth, compared ,at least, to the red line which represents Y. Also, from the form of the line of expected μ we can say that the expected value of μ can catch the true underlying dependence between Y and X as it follows well the red line.

5. Make a trace plot for μ_n and comment on the burn-in period and convergence.



We can see that the burn-in beriod of μ_{50} is very small, basically it takes around 10 or less,probably, iterations until it stabilizes so, this is a small burn-in period,we could say. Finally, it seems that that the chain has converged to the true posterior.