

MCMC sampling

msc2203121012

March 2023

1 Introduction

When there is a vast amount of data (population distribution) present, and to make sense of such massive data without knowing the underlying distribution one needs to study a subset of the population called a sample. Choosing a subset from the population is known as sampling and it should be done in such a way that the statistics(e.g. mean, variance, etc.) from the sample resembles that of the population. Various methods are available for sampling and Markov Chain Monte Carlo (MCMC) method is one of them.

All real-life problems are of this kind so MCMC sampling finds its application in many fields like physics, statistics, machine learning, etc.

MCMC methods are a family of algorithms that uses Markov chains to perform Monte Carlo estimate. Monte Carlo is a technique for randomly sampling a probability distribution and approximating a desired quantity. It assumes that samples are efficiently drawn from the target distribution.

MCMC can be used to sample from any probability distribution. Mostly it is used to sample from the intractable posterior distribution for the purpose of Inference.

Estimating the Posterior using Bayes can be difficult sometimes, in most cases we can find the functional form of Likelihood x Prior. However, computing marginalized probability $P(B)$ can be computationally expensive, especially when it is a continuous distribution.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where , $P(A|B)$ = posterior , $P(B|A)$ = likelihood , $P(A)$ = prior , $P(B)$ = evidence

Being difficult to evaluate, the calculation of the normalization constant is avoided.

A markov chain is initiated with a random probability distribution over states, gradually moving in the chain converging towards stationary distribution and applying condition **Detailed Balance Sheet** that ensures stationary distribution resembles desired probability distribution.

Detailed Balance Sheet Condition:

$$p(A)T(A \rightarrow B) = p(B)T(B \rightarrow A)$$

where, $p(A)$ represents the probability of being at A, and the probability $T(A \rightarrow B)$ represents the probability of moving from A to B. $p(B)$ represents the probability of being at B, and the probability $T(B \rightarrow A)$ represents the probability of moving from B to A.

Although MCMC itself is complicated, they provide a lot of flexibility. It provides efficient sampling in high-dimension. It can be used to solve problems with a large state space.

Metropolis-Hastings (MH) sampling :

MH sampling algorithms are effectively a guided random walk that preferentially samples from the high-probability regions while exploring the full range of possibilities.

Suppose there is a target posterior distribution: $\pi(x)$, where x can be any collection of parameters. In order to move around this parameter space some proposal distribution needs to be formulated.

If $p(x_{s+1} | x_s)$, specifies the probability of moving to a point x_{s+1} in parameter space, given that the current position is at x_s . The Metropolis-Hastings algorithm accepts a "jump" to x_{s+1} with the following probability,

$$P(x_{s+1}|x_s) = \min(1, \frac{\pi(x_{s+1})p(x_s|x_{s+1})}{\pi(x_s)p(x_{s+1}|x_s)}) \quad (1)$$

What the above expression means is that there is a high probability of transitioning to point x_{s+1} given the current position x_s is a function of the ratio of the value of the posterior at the new point to the old point (i.e., $\frac{\pi(x_{s+1})}{\pi(x_s)}$) and the ratio of the transition probabilities at the new point to the old point (i.e., $\frac{p(x_s|x_{s+1})}{p(x_{s+1}|x_s)}$). If this ratio is > 1 then the jump will be accepted (i.e. the chain advances to x_{s+1}).

Secondly, the ratio of the target posteriors ensures that the chain will gradually move to high probability regions. Lastly, the ratio of the transition probabilities ensures that the chain is not influenced by "favored" locations in the proposal distribution function.

Algorithm:

1. A starting point is chosen randomly (x_{s-1})
2. When the Markov chain is initiated, randomly another point is chosen from the proposal distribution. This point is called the trial point(x_{trial}).
3. If the trial point is accepted then the new point ($x_s = x_{trial}$) else old point is the new state($x_{s-1} = x_s$).
4. The probability for acceptance of trial point = minimum of $[\frac{p(x_{trial})}{p(x_{s-1})}]$ and 1.
5. If it returns 1, it means the trial values are most probable and thus are accepted else not.
6. Above steps are continued for many iterations.

Limitation — MCMC does not perform well in an approximating probability distribution that has multi modes.

Discussion of the question

The fundamental process of running an MCMC is to compare generated models against data. Those models are generated by a set of parameters, and the goal is to sample from the set of parameters that produces the models that well-fit the given data.

In the given question, values of h and Ω_m are to be estimated using MCMC sampling.

Steps:

1. First, data (31 pairs) of " z vs μ " is imported as a numpy array and is then plotted.[z = redshift($z < 1.3$) , μ = distance modulus]

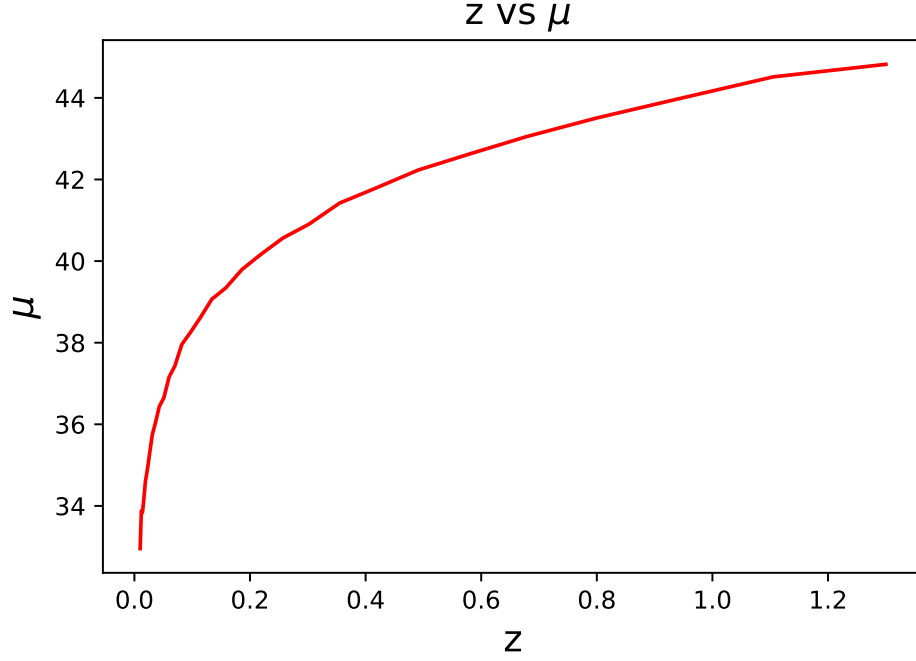


Figure 1: Data

2. Next inverse of (31×31) covariance matrix is done .
3. Then, the functions required for computation are defined in separate functions:

Hubble's constant,

$$H_o = 100 h k m s^{-1} M p c^{-1} \quad (2)$$

h = Hubble parameter

Ω_m is the density parameter for the baryonic matter.

Luminosity distance,

$$D_L(z) = \frac{c(1+z)}{H_o} \left[\eta(1, \Omega_m) - \eta\left(\frac{1}{1+z}, \Omega_m\right) \right] \quad (3)$$

where ,

$$\eta(a, \Omega_m) = 2\sqrt{s^3 + 1} \left[\frac{1}{a^4} - 0.1540 \frac{s}{a^3} + 0.4304 \frac{s^2}{a^2} + 0.19097 \frac{s^3}{a} + 0.066941 s^4 \right]^{-1/8} \quad (4)$$

and ,

$$s^3 = \frac{1 - \Omega_m}{\Omega_m} [0.2 \leq \Omega_m \leq 1] \quad (5)$$

Distance modulus :

$$\mu = 25 - 5 \log_{10} h + 5 \log_{10} \left(\frac{D_L^*}{\text{Mpc}} \right) \text{ where, } D_L^* = D_L(h = 1) \quad (6)$$

4. L =likelihood . Since there is a uniform prior posterior will be proportional to likelihood.

$$L \propto \exp \left[-\frac{1}{2} \sum_{i,j=1}^n (\mu_i - \mu_{th}(z_i)) C^{-1}_{ij} (\mu_j - \mu_{th}(z_j)) \right] \quad (7)$$

5. A function for MCMC sampling is defined where σ of the proposal distribution (Gaussian is used) and the no. of iterations are taken as arguments.

6. For initializing, values of parameters (h, Ω_m) are taken randomly from (0 - 1).

7. Next, empty arrays are created to append the accepted values of h and Ω_m .

8. Next, sampling is started and for every iteration trial values for each parameter are taken from the Gaussian distribution whose mean is centered around the previous value of parameters.

9. Next, the likelihood of those trial points is evaluated using the above formula.

10. Ratio of the likelihoods of the trial values and the previously accepted values is evaluated.

11. A new variable(f1) is assigned to store the minimum value between the ratio of likelihoods and 1.

12. Next, another variable(g) is taken which stores a random number between 0 and 1.

13. Then, comparing those two variables trial points are checked for acceptance and rejection. This is the Metropolis-Hastings (MH) condition for

acceptance of the trial values.

14. If $g < f_1$, trial values are accepted.

15. If accepted then the trial values become the new estimated values for the parameters and for the next iteration trial values are selected from the Gaussian distribution centering them.

16. If rejected then the previously accepted values of the parameters are taken as the mean values for the next iteration and again new trial values are chosen from the next iteration.

17. No. of times trial values are accepted is also evaluated.

For $\sigma = 0.05$ and for 1000 iterations:

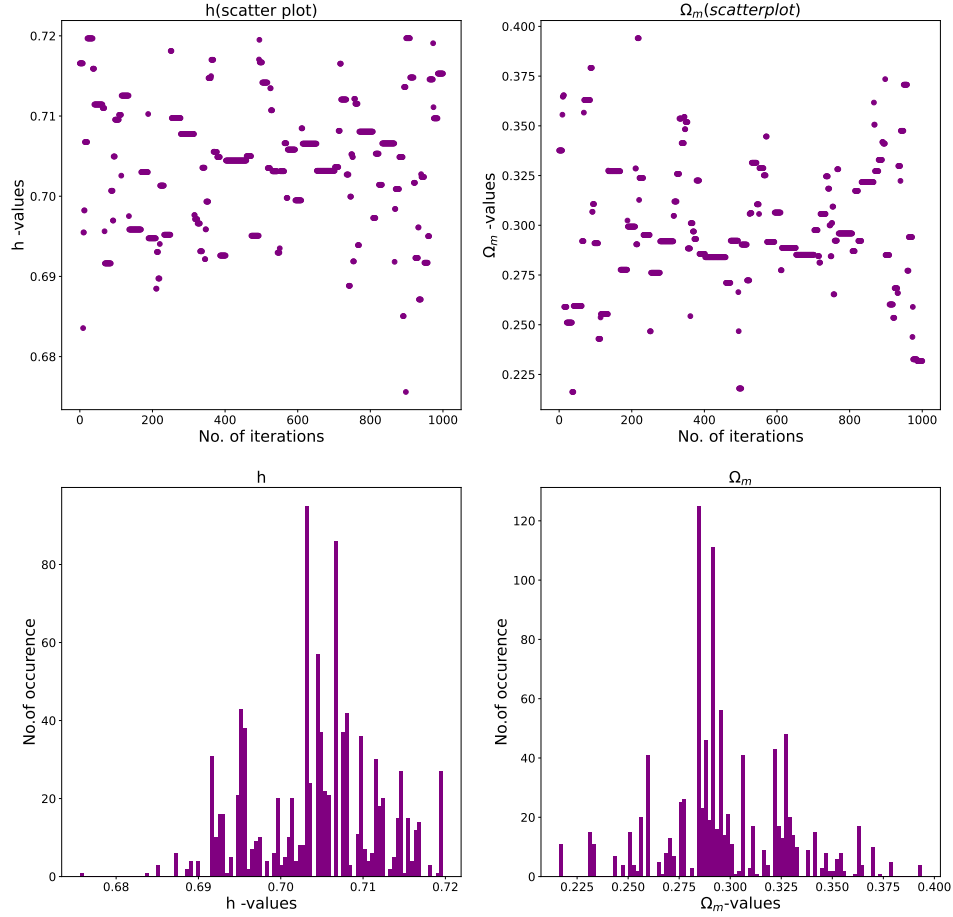


Figure 2: for $\sigma = 0.05$

From the scatter plots, it can be seen that after some iterations Markov chain is reaching the stationary distribution. The values before the stationary distribution are known as burn-in and can be ignored.

From, the histograms it is observed that the most probable value of $h = 0.703$ and $\Omega_m = 0.28$.

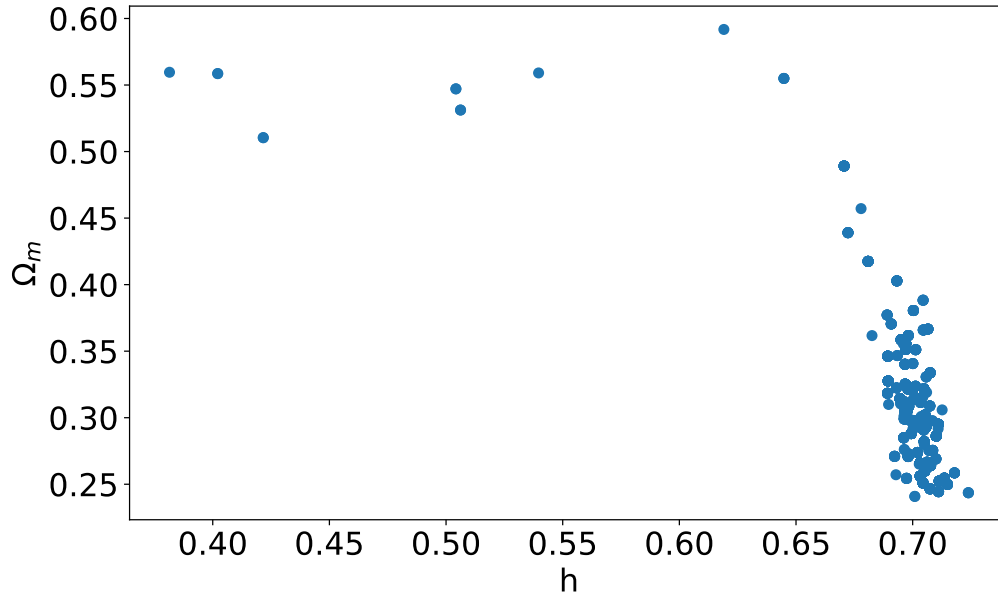


Figure 3: for $\sigma = 0.05$

Using these values, values of μ and the observed data are plotted.

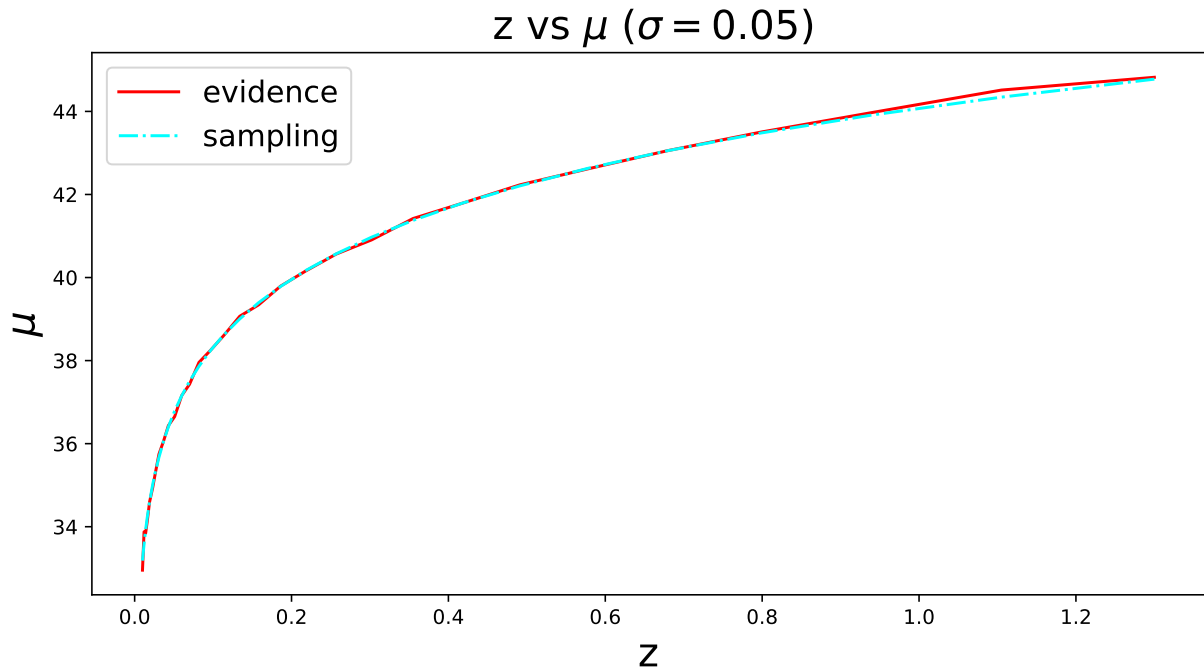


Figure 4: for $\sigma = 0.05$

Since the values are matching with the observed data it can be said that sampling was done well.

For $\sigma = 0.001$ and for 1000 iterations:

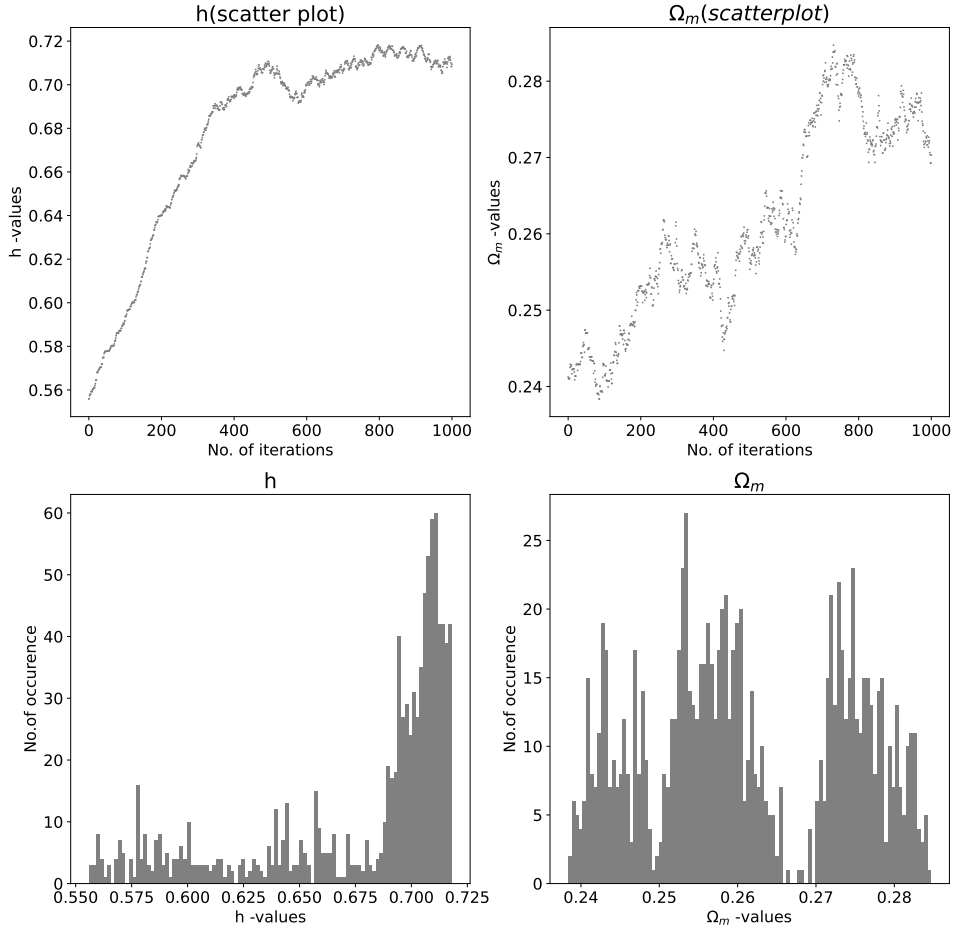


Figure 5: for $\sigma = 0.001$

From the scatter plots, it can be seen that after some iterations Markov chain has not yet reached the stationary distribution. No. of iterations need to be increased for converging into the stationary distribution. The values

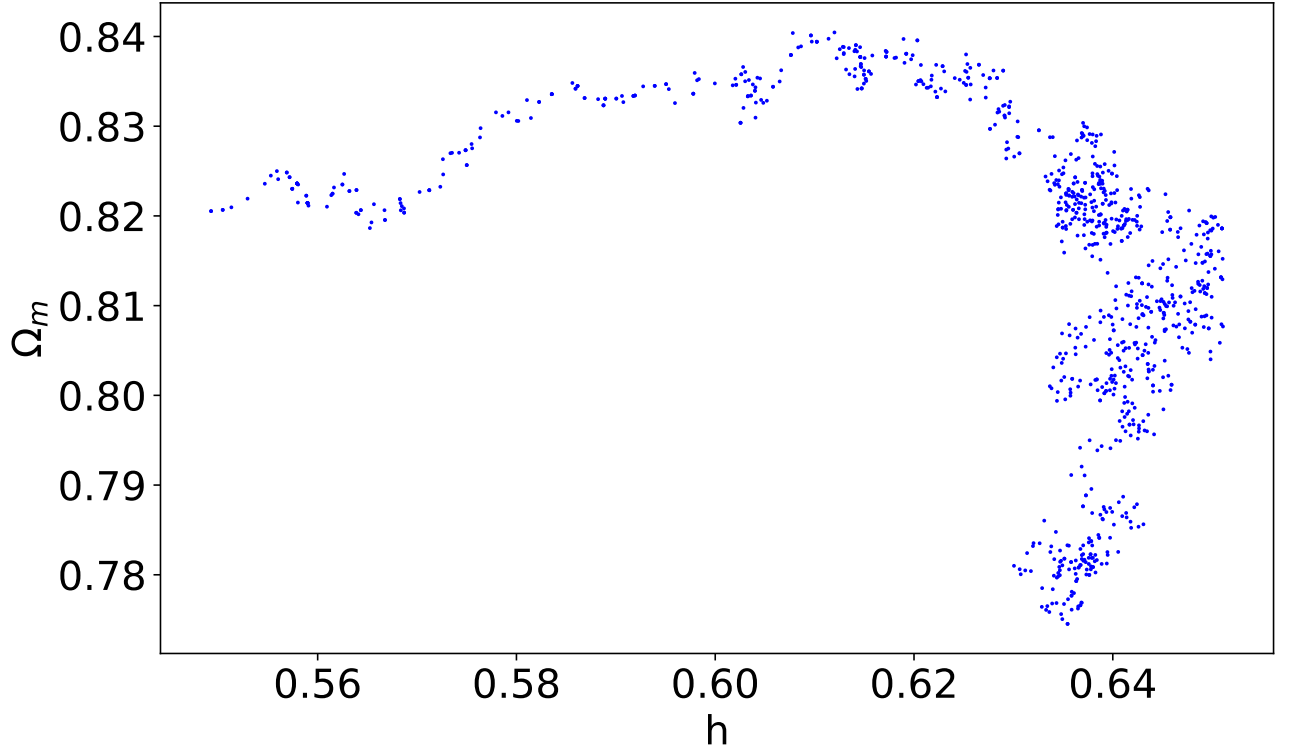


Figure 6: for $\sigma = 0.001$

before the stationary distribution are known as burn-in and can be ignored.

From, the histograms it is observed that the most probable value of $h = 0.72$ and $\Omega_m = 0.26$.

Using these values, values of μ and the observed data are plotted.

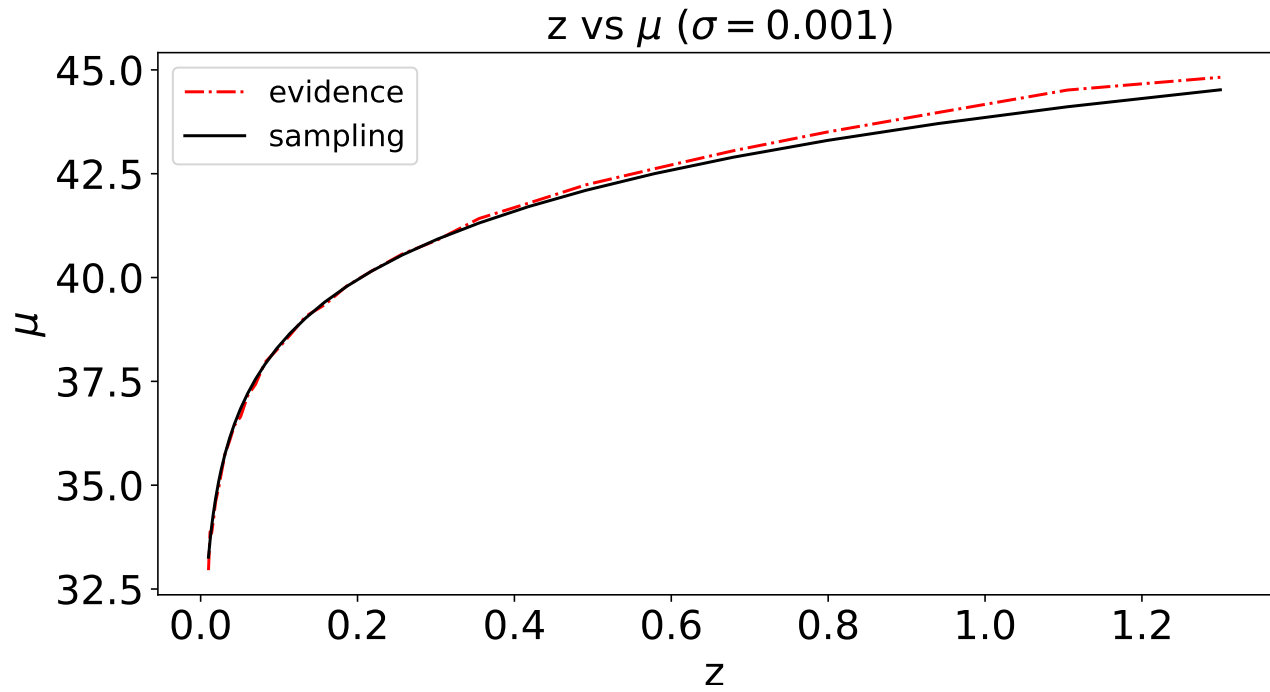


Figure 7: for $\sigma = 0.001$

Since the values are deviating from the observed data for higher values of z it can be said that sampling was not done well.

For $\sigma = 1$ and for 1000 iterations:

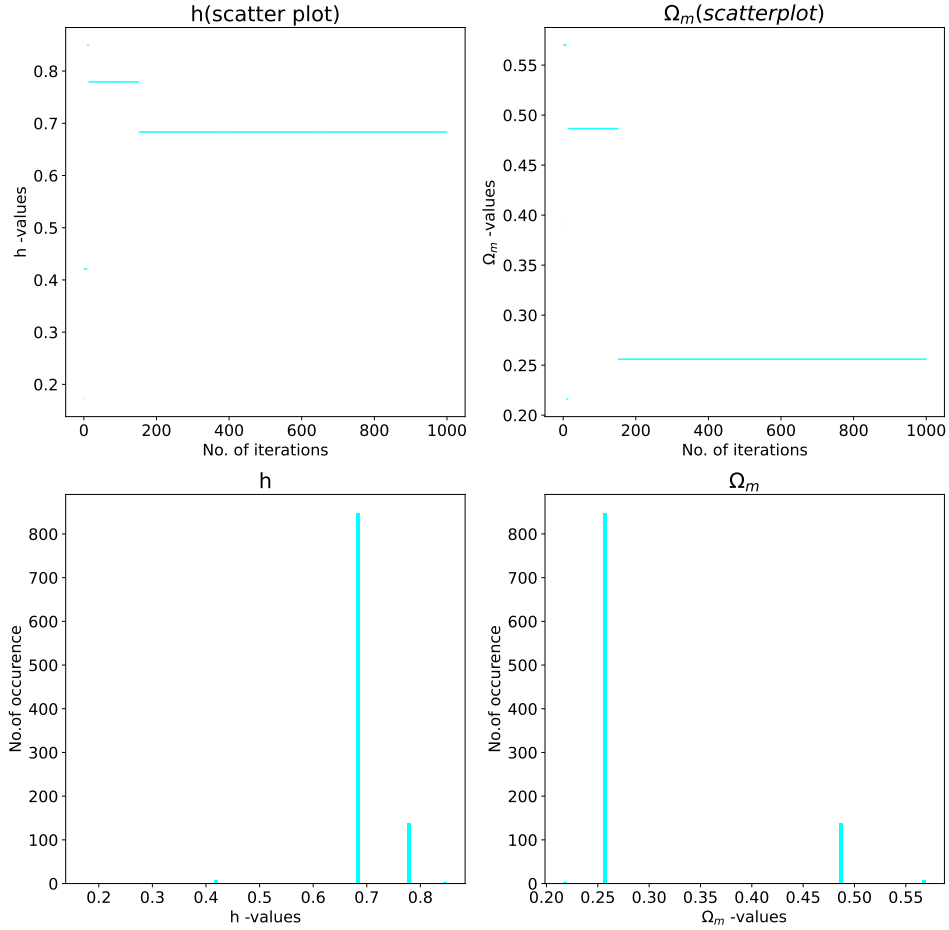


Figure 8: for $\sigma = 1$

From the plots, it is seen that the sampling is very poor. From, the histograms it is observed that the most probable value of $h = 0.68$ and $\Omega_m = 0.26$.

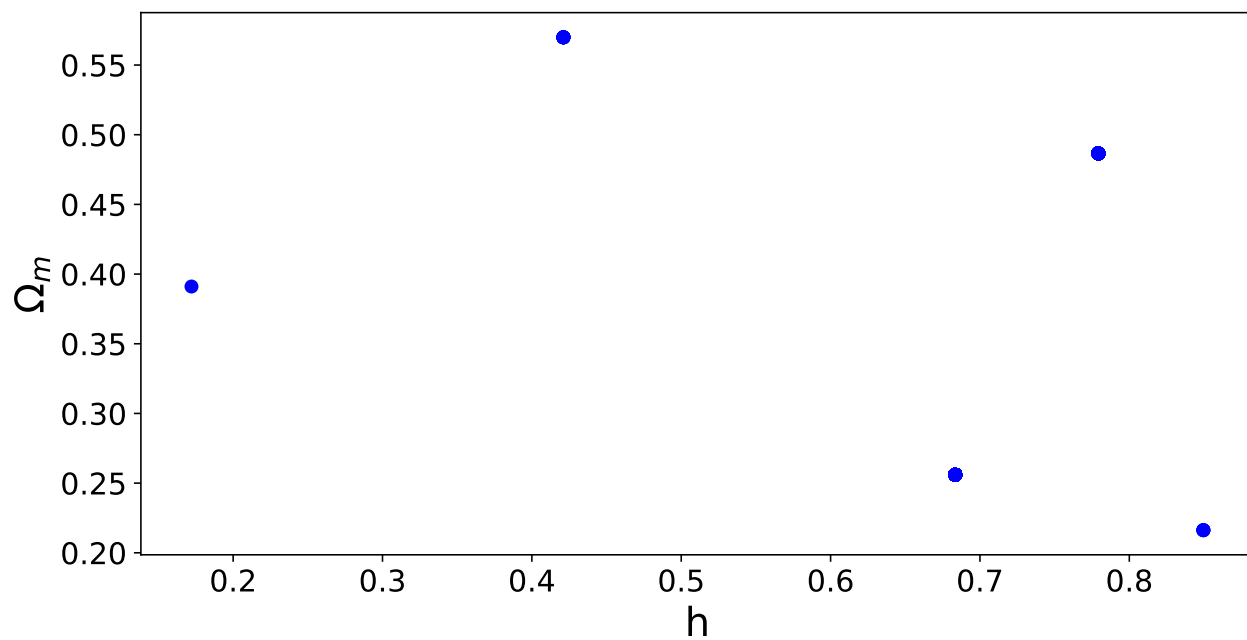


Figure 9: for $\sigma = 1$

Using these values, values of μ and the observed data are plotted.

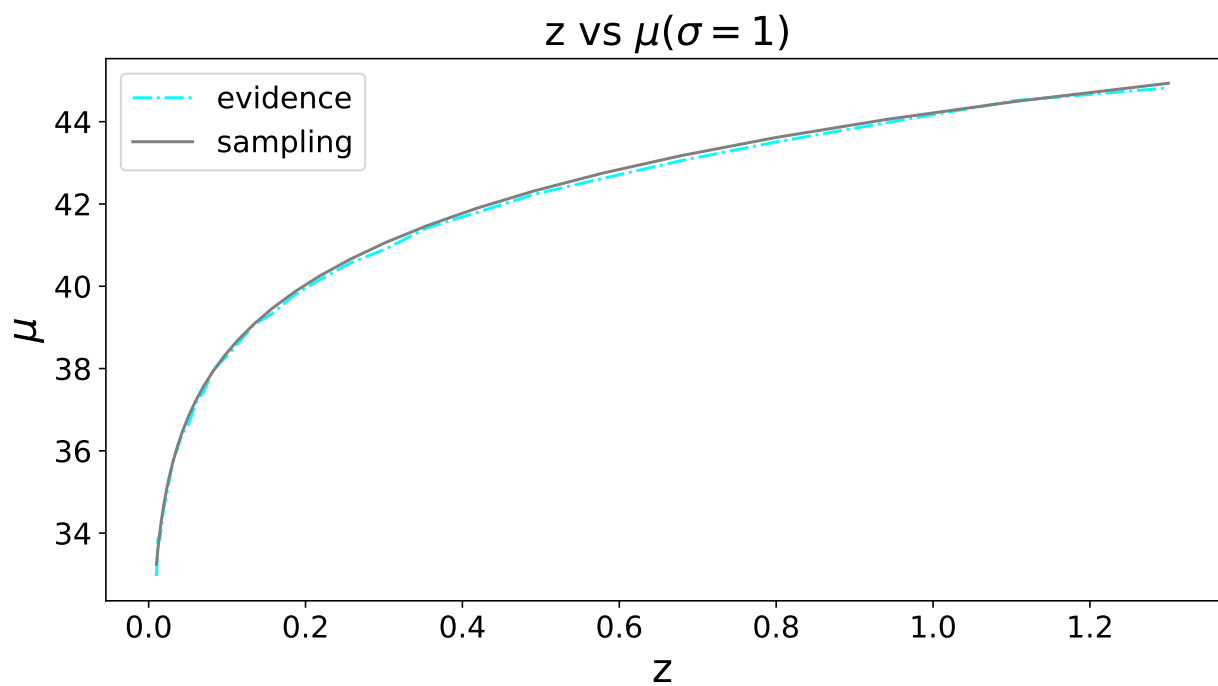


Figure 10: for $\sigma = 1$

Though the values are very near to the observed data it cannot be said that sampling was done well.

Acceptance ratio

: While using the Metropolis-MH algorithm, the acceptance rate needs to be monitored so that it stays within the optimal range. If the trial value is accepted almost every time, it states that each time the chain only jumps a very small step (so that the acceptance ratio is close to 1 every time), which will make the algorithm slow in converging to the stationary distribution. On the other hand, if the acceptance rate is very low, then the chain will get stuck to just a few locations and it takes hundreds of iterations for it to make one jump. For the Metropolis/MH algorithm, an optimal acceptance rate would be something between 10% to 60%.

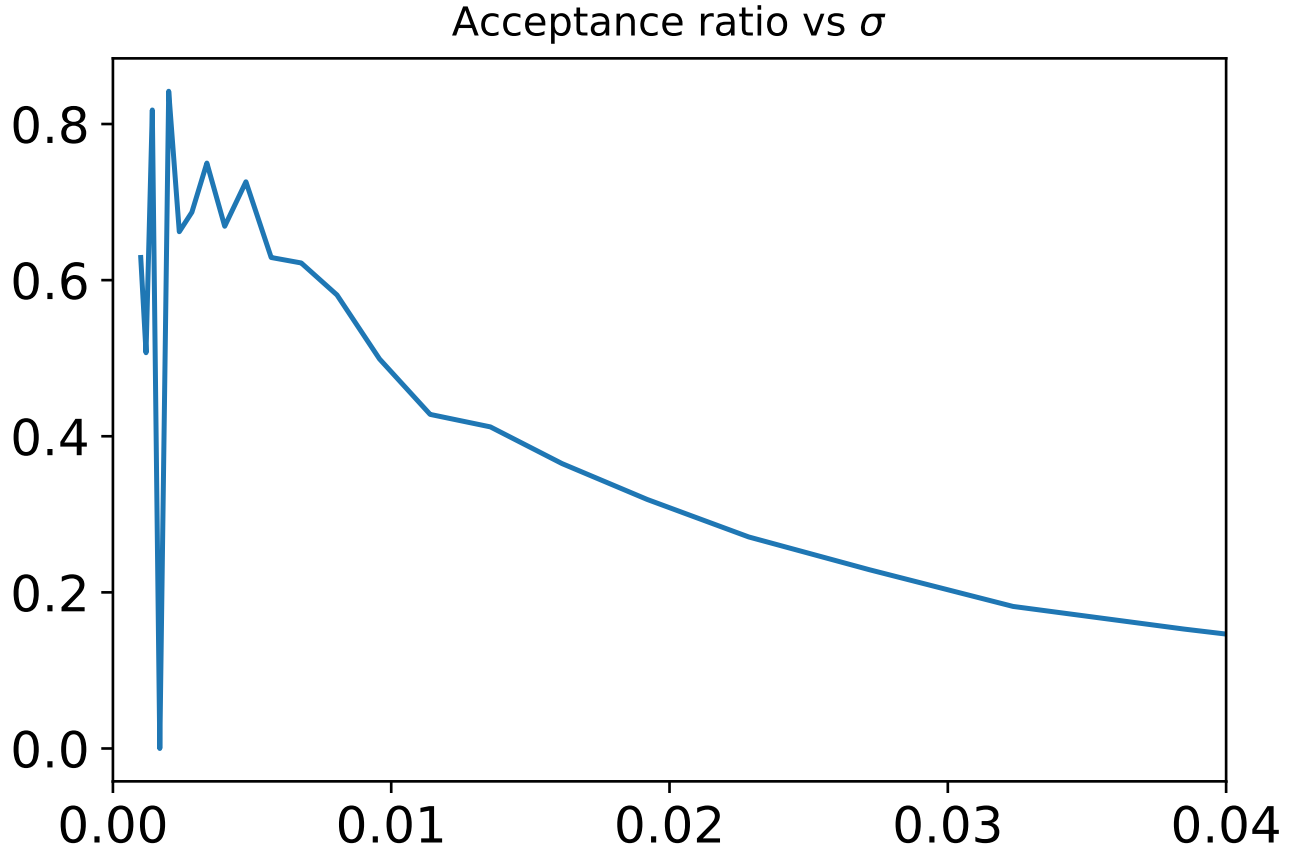


Figure 11:

For optimum σ value and for 10000 iterations:

From the plot of the "acceptance ratio vs standard deviation", $\sigma = 0.04$ value seems to be a good value for sampling. From, the histograms it is observed that the most probable value of $h = 0.70$ and $\Omega_m = 0.30$.

The burn-in is estimated from the plot and is marked.

Even the values from the sample :

sample mean(h): $0.703016662149768 \approx 0.7$

sample mean(Ω_m): $0.29548839863811871 \approx 0.3$

sample variance(h): $5.7724906292772076 \times 10^{-05}$

sample variance(Ω_m): 0.0010402468258699502
covariance between h and Ω_m :-1.59753359 x 10⁻⁰⁴
Using these values, values of μ and the observed data are plotted. As expected, from the above plot μ values match well with the observed data, and the sampling is done well.

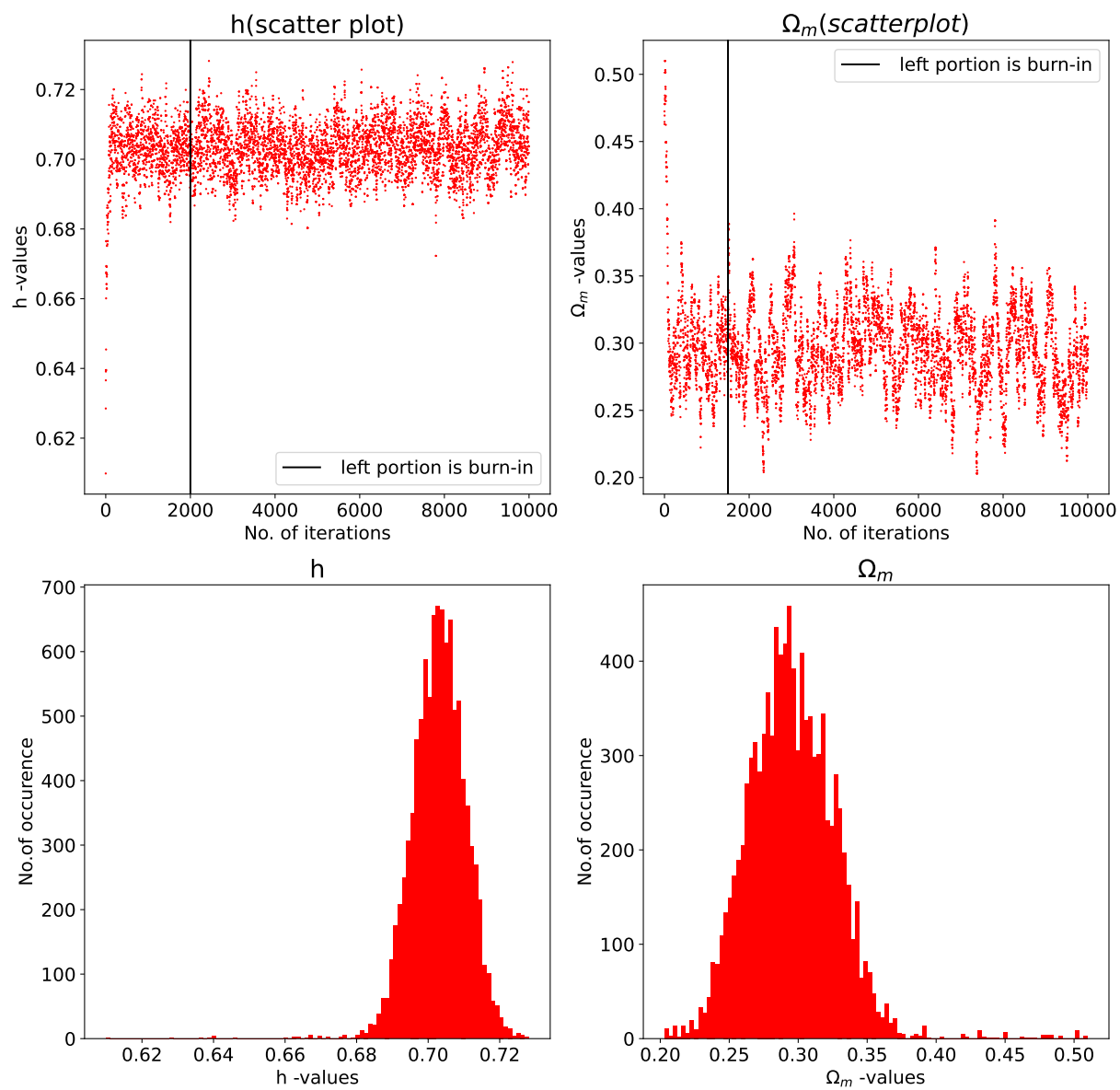


Figure 12:
19

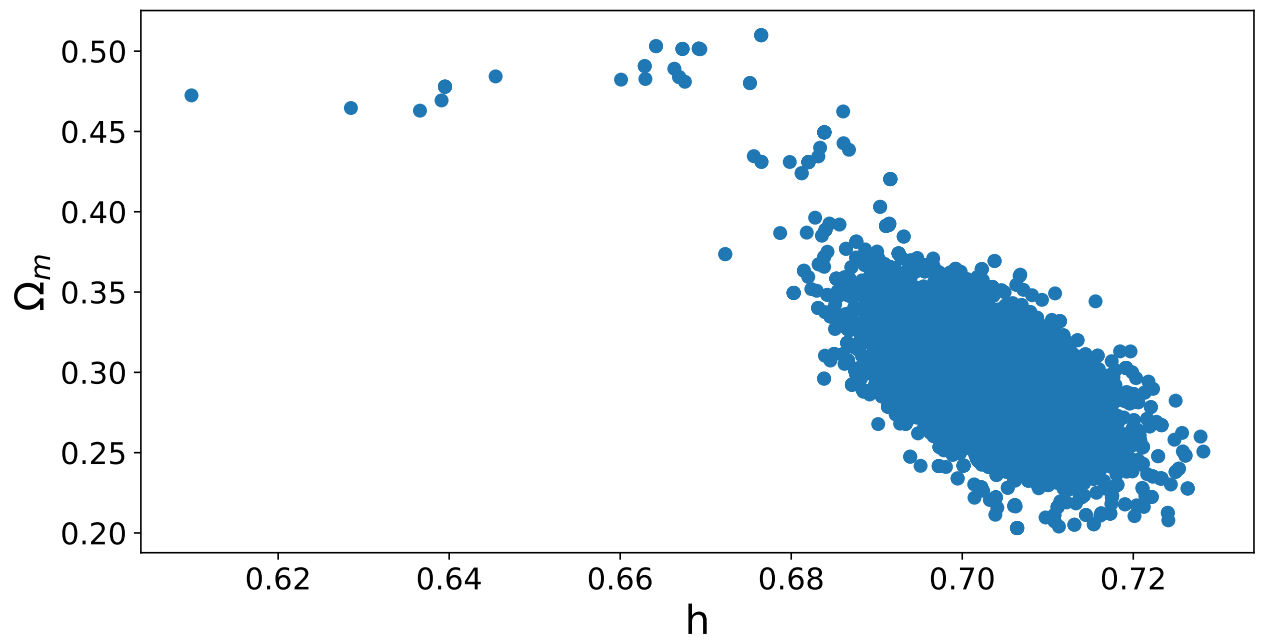


Figure 13: for optimal value

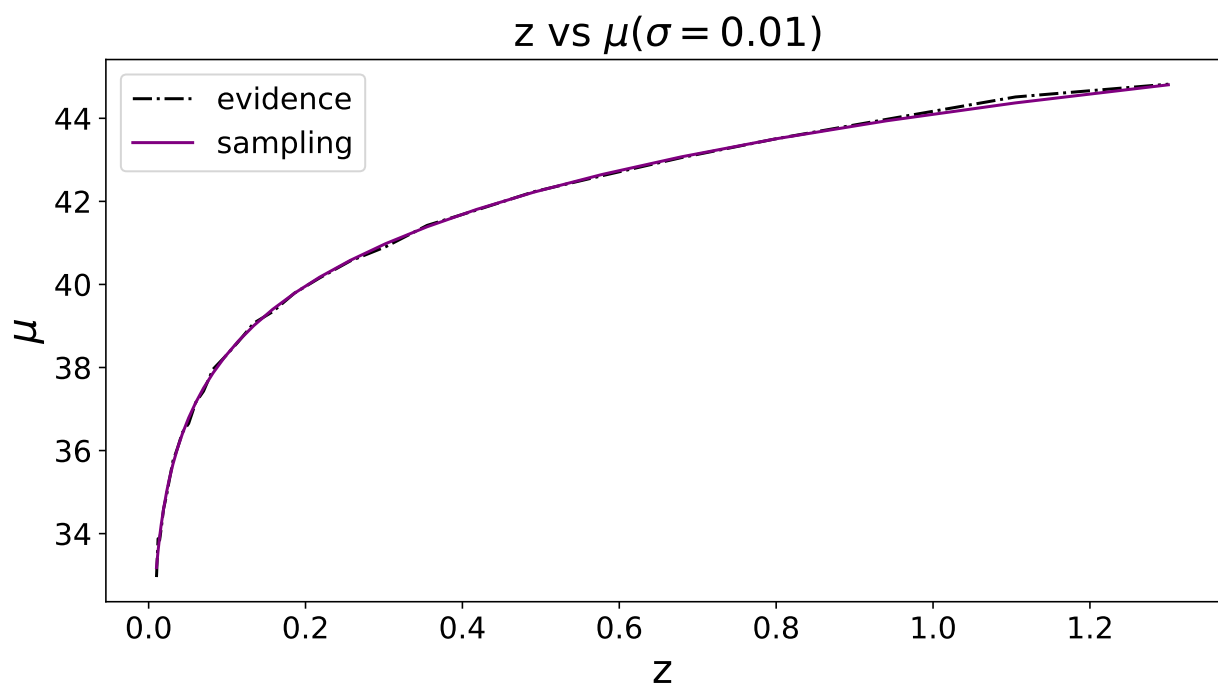


Figure 14: for optimal value