MONTE CARLO MARKOV CHAINS

-- Group 5

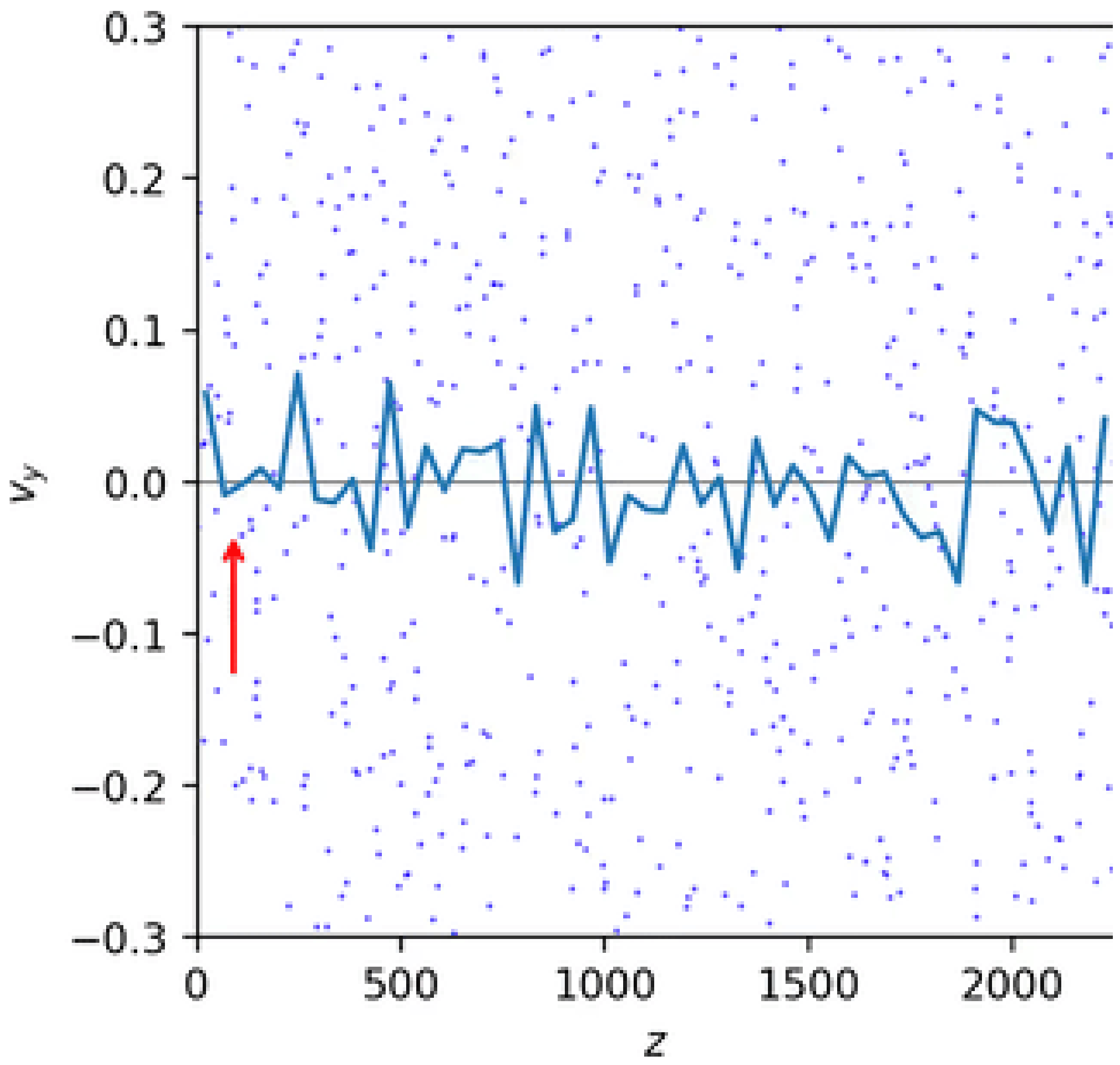
A BIT OF BACKGROUND

In probabilities and machine learning often an analytical solution is not feasible as sum of discrete or integral of continuous RV is not possible. This can be due to the following reasons:

- **Noise** in data
- Stochastic nature of process
- Large number of RVs

MONTE CARLO SAMPLING

These methods are a class of computational algorithms that rely on repeated random sampling to optimize and integrate functions or generate draws from probability distribution.**<u>The samples drawn are independent of each other</u>**. It is used in situations where deterministic methods may be difficult or impossible to apply.



https://medium.com/swlh/create-your-own-direct-simulation-monte-carlo-with-python-3b9f26fa05ab

BAYESIAN ANALYSIS

PRIOR PROBABILITY

The knowledge of the occurrence of an event before looking at the statistical/simulated data

POSTERIOR PROBABILITY

The empirical probability calculated after looking at the simulations

TERMINOLOGY The Bayes Rule

The random variable modelling prior knolwedge

Likelihood

The posterior probability (what we are trying to estimate)

$$f_{\theta|X}(\theta|x_1, x_2, ..x_n) = \frac{f_{X|\theta}(x_1, x_2, ..x_n|\theta) f_\theta(\theta)}{f_X(x_1, x_2, ..., x_n)}$$

Normalizing Factor

MONTE CARLO MARKOV CHAIN

Monte Carlo Markov Chain(MCMC) is a simple computer-driven sampling method that allows one to characterize a distribution without knowing all of the distribution’s mathematical properties by randomly sampling values out of the distribution.

- It simply combines the Monte Carlo Sampling with the Markov chain property.
- Each random sample is used as a stepping stone to generate the next random sample

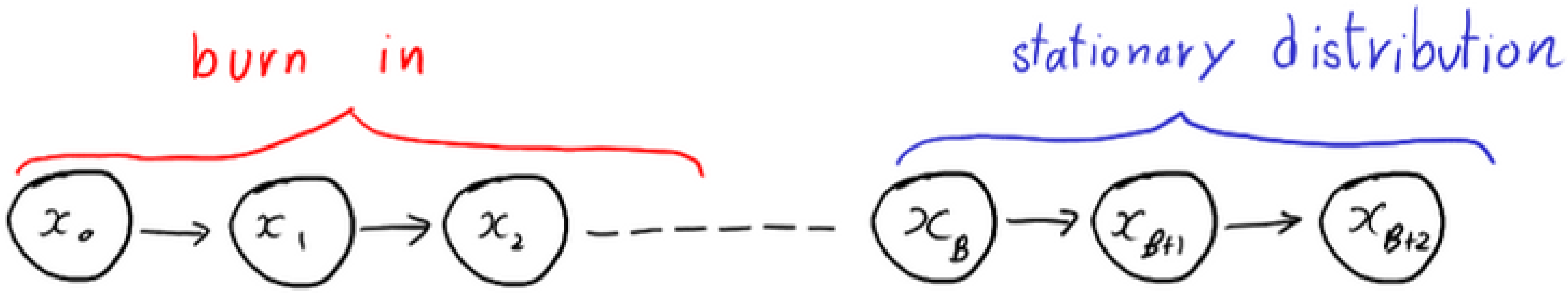
MATHS BEHIND MCMC

In the context of Monte Carlo Markov Chains (MCMC), the detailed balance equation is expressed as :

$$\pi(x)P(x \rightarrow y)=\pi(y)P(y \rightarrow x)$$

- $\pi(x)$ is the stationary probability of being in state x,
- $P(x \rightarrow y)$ is the transition probability from state x to state y,

This ensures that the Markov chain reaches a stationary distribution

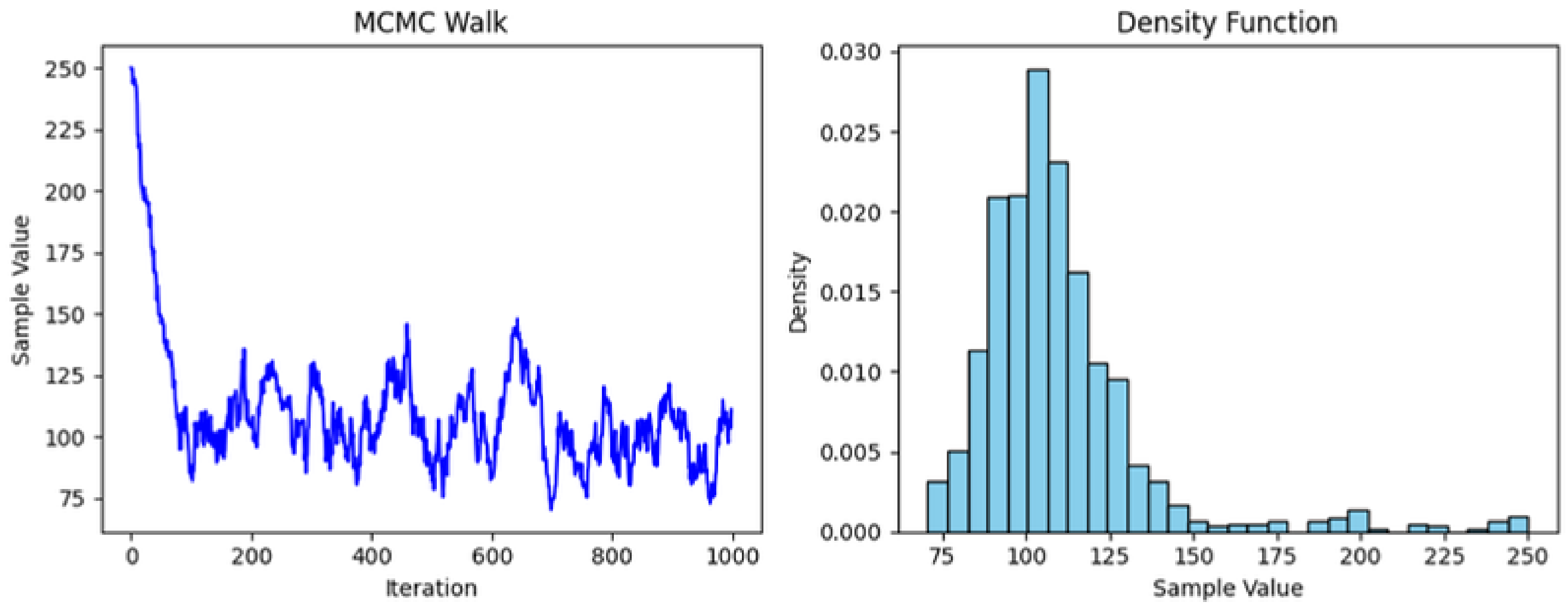


To understand this burn-in, let's look at an oversimplified example.

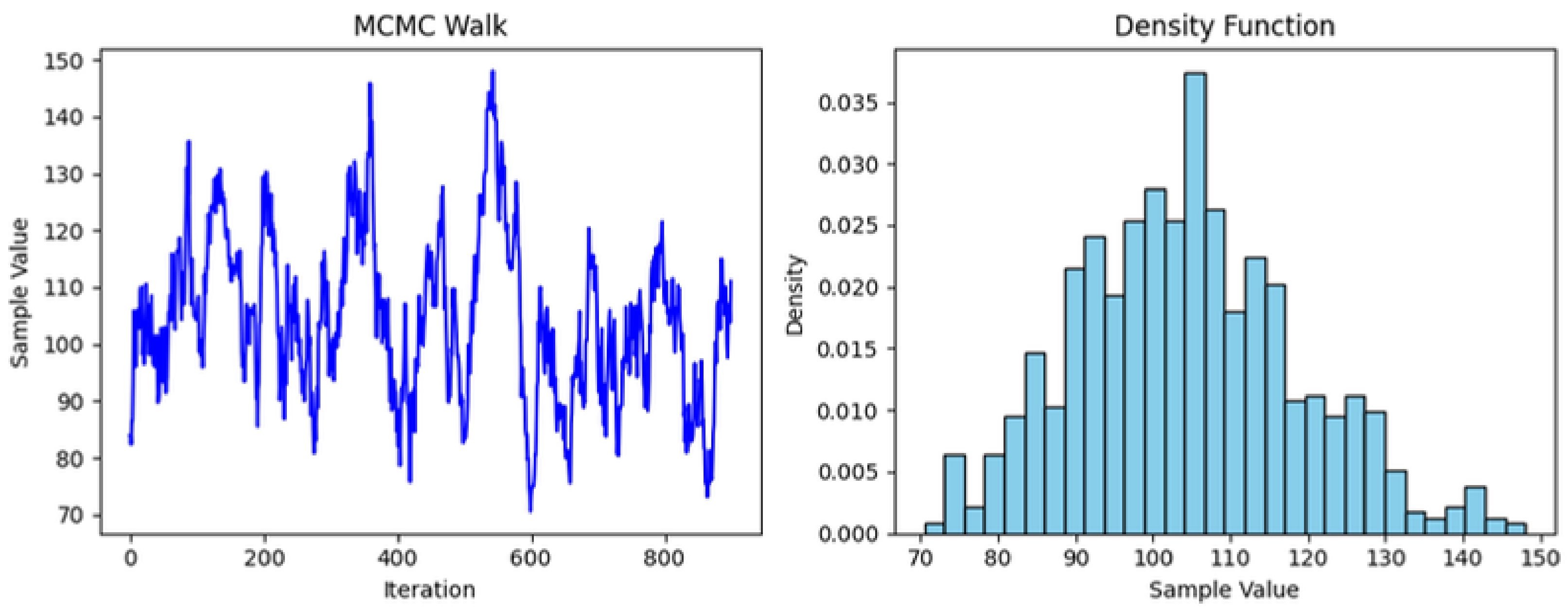
A lecturer seeks to determine the mean test score in a student population, assuming a normal distribution with a known standard deviation of 15. With only one observed test score of 100, the goal is to employ Markov Chain Monte Carlo (MCMC) to draw samples from the posterior distribution.

This, provides us with a simple posterior distribution of N(100,15)

We can see the difference between the chain with and without burn-in

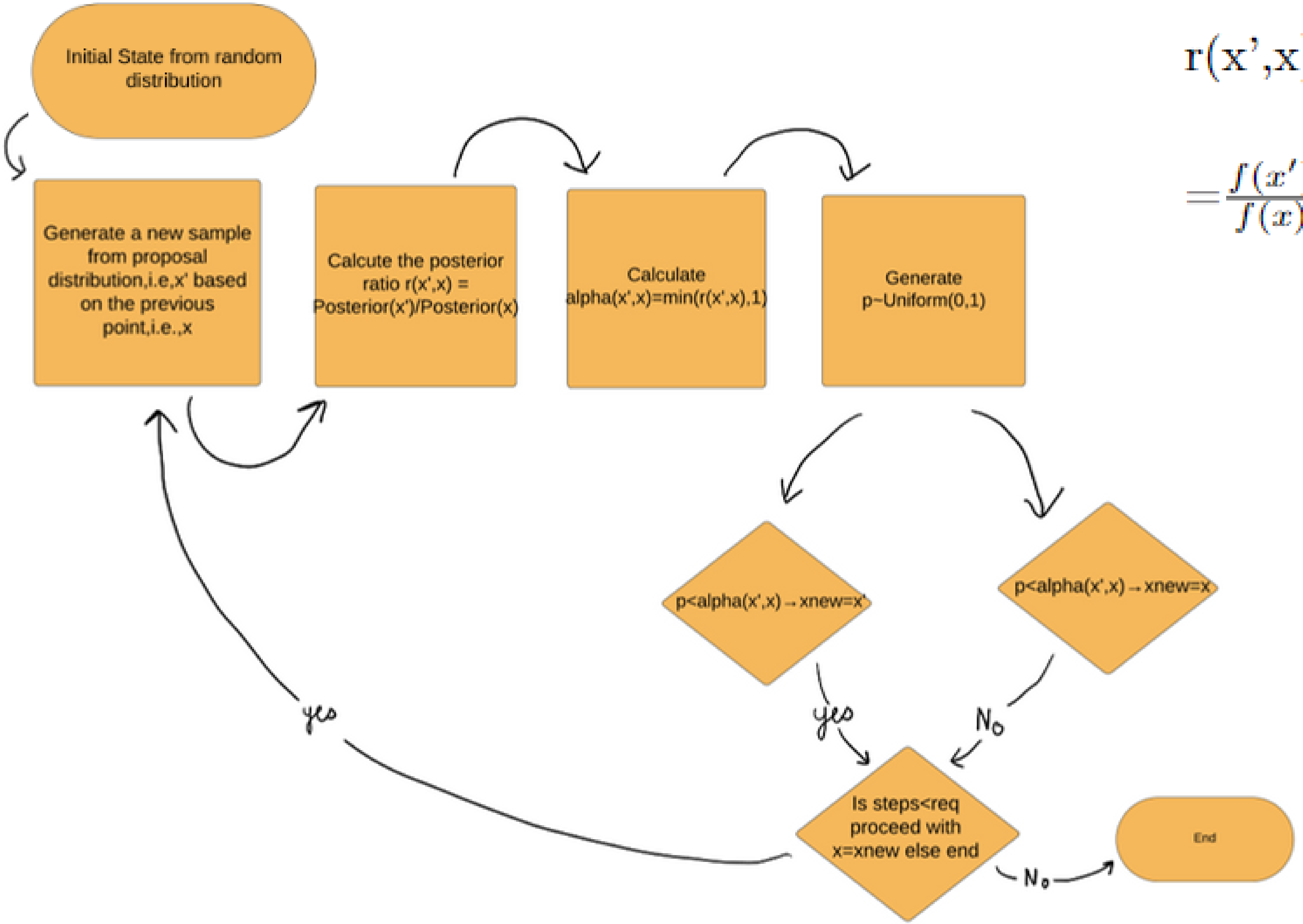


Standard Deviation: 25.91
Expectation: 109.76



Standard Deviation: 14.35
Expectation: 104.39

METROPOLIS HASTING



$$r(x',x) = \frac{Posterior(x')}{Posterior(x)}$$

$$= \frac{f(x')g(x|x')}{f(x)g(x'|x)}$$

WHY DOES METROPOLIS HASTING EVEN WORK?

Acceptance probability(A(x->x')) of moving from state x to state x' is given by :

$$r(x',x) = \frac{Posterior(x')}{Posterior(x)} = \frac{f(x')g(x|x')}{f(x)g(x'|x)}$$

Here, f(x) : prior distribution
g(x) : Likelyhood

and $A(x\text{-}{>}x') = \min(1, r(x',x))$

This easily flows into the detailed balance equation $f(x')\ g(x|x')\ A(x'\text{-}{>}x) = f(x)\ g(x'|x)\ A(x\text{-}{>}x')$

As, the probability density to move is higher in any given distribution when the points are closer to one and other, it enforces a sense of stableness. And even though the states are dependent on each other in the short run. They become independent over a long run and gain stability

FINDING POSTERIOR DISTRIBUTIONS

The Problem :

Let $P \in (0, 1)$ be a random variable modelling the probability of obtaining heads (p) such that P~Uniform(0,1) . Let X = (X1,X2,...Xn) are the results of n coin tosses, such that all Xi follow Xi ~ Bernoulli(p)

Goal: To find the posterior Distribution $P_{P|X}(P \mid X)$.

SOLUTION-1

PARAMETER: P (PROBABILITY OF GETTING HEADS)
Random Variable modelling the parameter : P : $f_P(p)$~Uniform(0,1)

$$f_{X,P}(x_1, x_2 \dots .\dots . x_n, p) = f_{X|p}(x_1, x_2 \dots .\dots . x_n|p) f_p(p)$$

$$f_{X|P}(x_1, x_2 \dots \dots \dots x_n, p) = \prod_{p=1}^{n} f_{x_i|p}(x_i|p) f_p(p) \qquad \{x1, x2 \dots \dots xn \text{ are IID}\}$$

$$f_{X|P}(x_1, x_2 \dots \dots \dots x_n, p) = \prod_{i=1}^{n} p^{x_i}(1-p)^{1-x_i} * 1 \qquad \{f_p(p) = 1\}$$

$$f_{X|P}(x_1, x_2, \dots \dots \dots x_n, p) = p^{\sum_{i=1}^{n} x_i}(1-\mathrm{p})^{\sum_{i=1}^{n}(1-x_i)}$$

$$define\ s = x_1 + x_2 + \cdots + x_n$$

$$f_{X|P}(x_1, x_2, \dots \dots \dots x_n|p) = p^s(1-p)^{n-s}$$

$$f_X(x_1, x_2 \dots \dots x_n) = \int_0^1 f_{X|P}(x_1, x_2 \dots \dots \dots x_n|p) f(p) dp = B(s+1, n-s+1)$$

$$f_{P|X}(p|x_1, x_2, \dots \dots \dots x_n) = \frac{f_{X,P}(x_1, x_2 \dots .\dots . x_n, p)}{f_X(x_1, x_2 \dots \dots \dots x_n)} = \frac{p^s(1-p)^{n-s}}{B(s+1, n-s+1)}$$

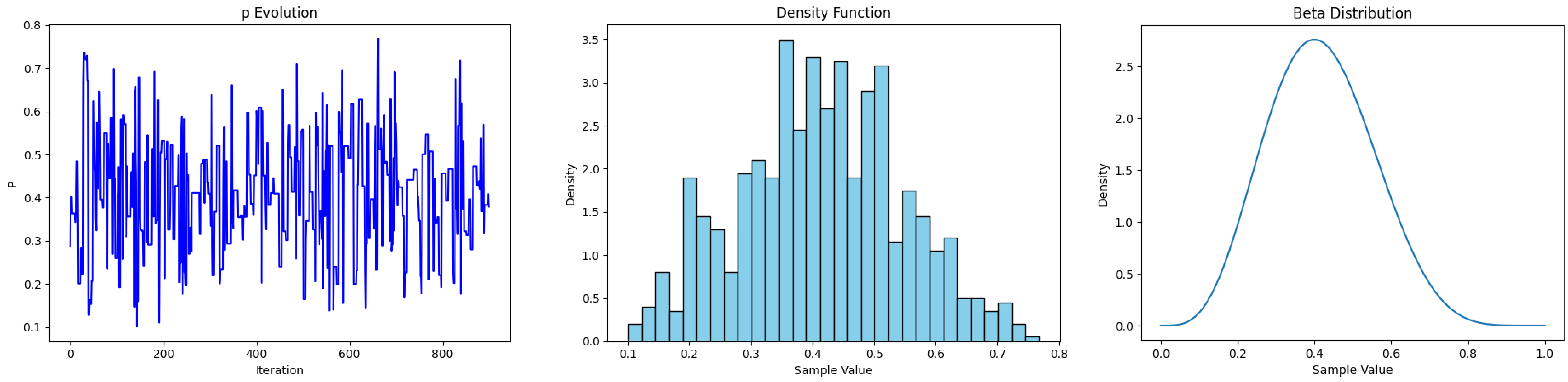
Thus we find that (ignoring the constant from the denominator)

POSTERIOR PRIOR

$$f_{P|X}(p|x_1, x_2, \dots \dots \dots x_n) \propto p^s(1-p)^{n-s} * 1$$

LIKELIHOOD

SIMULATION USING MCMC



INFERENCES

- samples mean= 0.4135
- Actual mean= 0.4
- From the simulation the estimated mean is a good approximation for the actual distribution.

ANOTHER EXAMPLE

<u>The Problem :</u>

Let $\lambda \sim$ Uniform(0,1) be the random variable modelling the parameter of an exponential random variable. Let X = (X1,X2,X3...Xn) be n successive inter-arrival times of the poisson process. Therefore each Xi ~ Exponential(λ)

Goal : To find the posterior Distribution : P(X|λ)

SOLUTION-2

Parameter: λ (Rate parameter)
Random Variable modelling the parameter : P $f_\lambda(\lambda)$~Uniform(0,1)

$$f_{x|\lambda}(x_1, x_2, \dots \dots \dots x_n, \lambda) = f_{X|\lambda}(x_1, x_2, \dots \dots \dots x_n|\lambda)$$
$$f_{X|\lambda}(x_1, x_2, \dots \dots \dots x_n, \lambda) = \prod_{i=1}^{n} f_{x_i|\lambda}(x_i|\lambda) f_\lambda(\lambda)$$
$$f_{X|\lambda}(x_1, x_2 \dots \dots \dots x_n, \lambda) = \prod_{i=1}^{n} (\lambda e^{-\lambda x_i}) * 1$$
$$f_{X|\lambda}(x_1, x_2, \dots \dots \dots . x_n, \lambda) = \lambda^n e^{-\lambda(x_1+x_2+\cdots x_n)}$$

define s = $x_1 + x_2 + x_3 + \dots x_n$

$$f_{X|\lambda}(x_1, x_2, ..x_n|\lambda) = \lambda^n e^{-\lambda s}$$
$$f_X(x_1, x_2, ...x_n) = \int_0^1 f_{X|\lambda}(x_1, x_2, ...x_n, \lambda) f(\lambda) d\lambda = s^{-(n+1)}(\Gamma(n+1) - \Gamma(n+1, s))$$
$$\frac{f_{X|\lambda}(x_1, x_2, x_3, ...x_n|\lambda) f(\lambda)}{f_X(x_1, x_2, ...x_n)} = \frac{\lambda^n e^{-\lambda s}.1}{s^{-(n+1)}(\Gamma(n+1) - \Gamma(n+1, s))}$$

Thus we find that (ignoring the constant from the denominator)

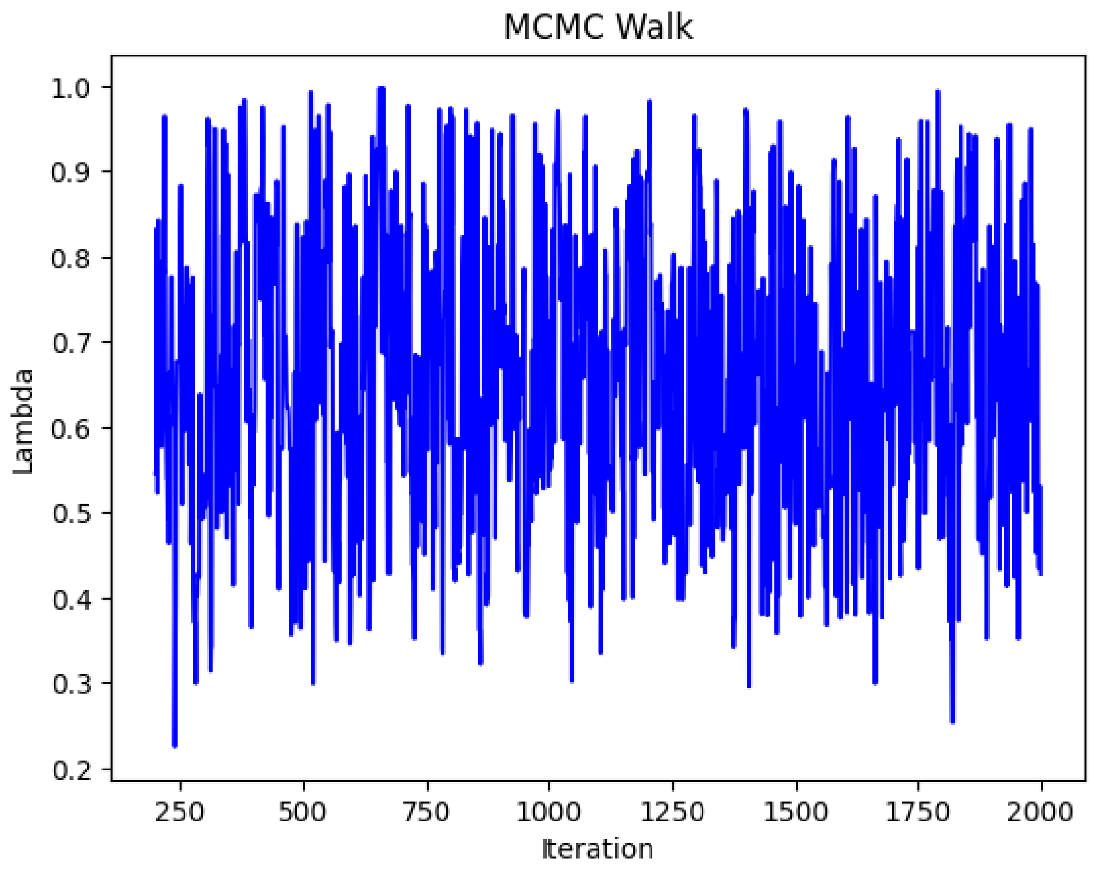
POSTERIOR

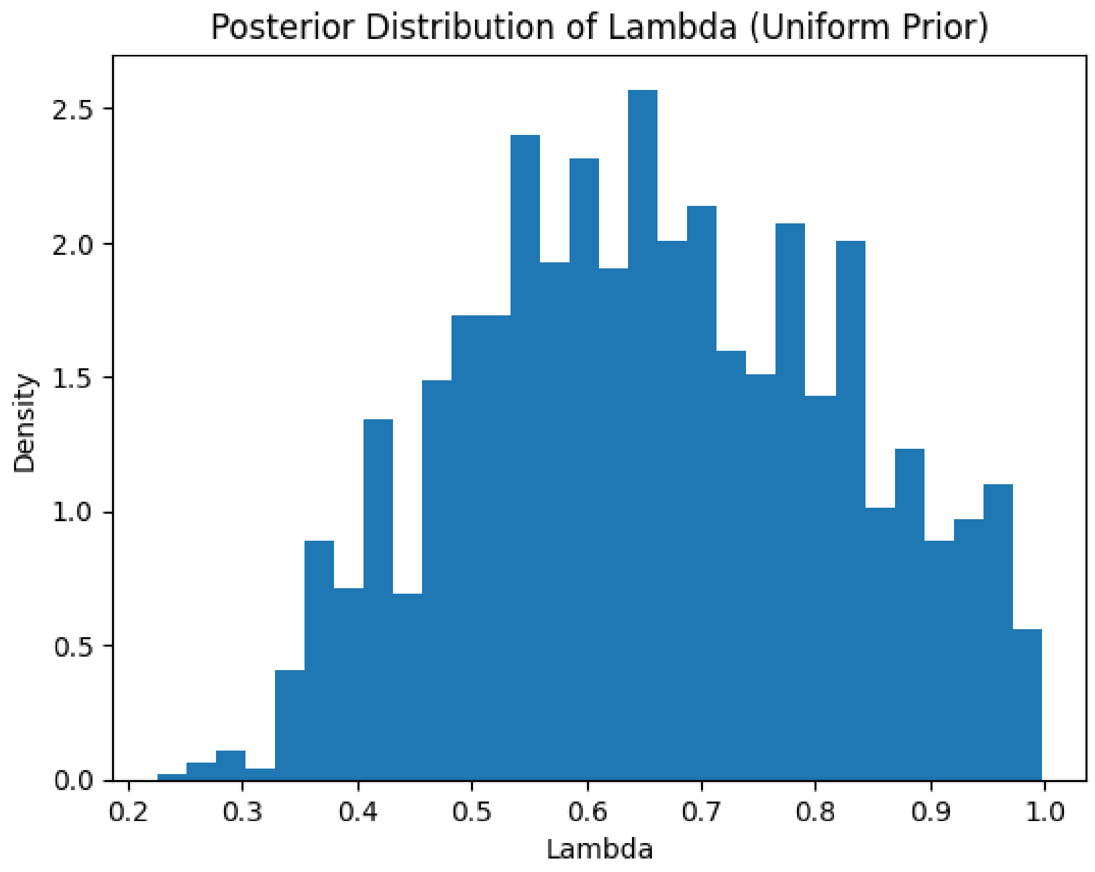
PRIOR

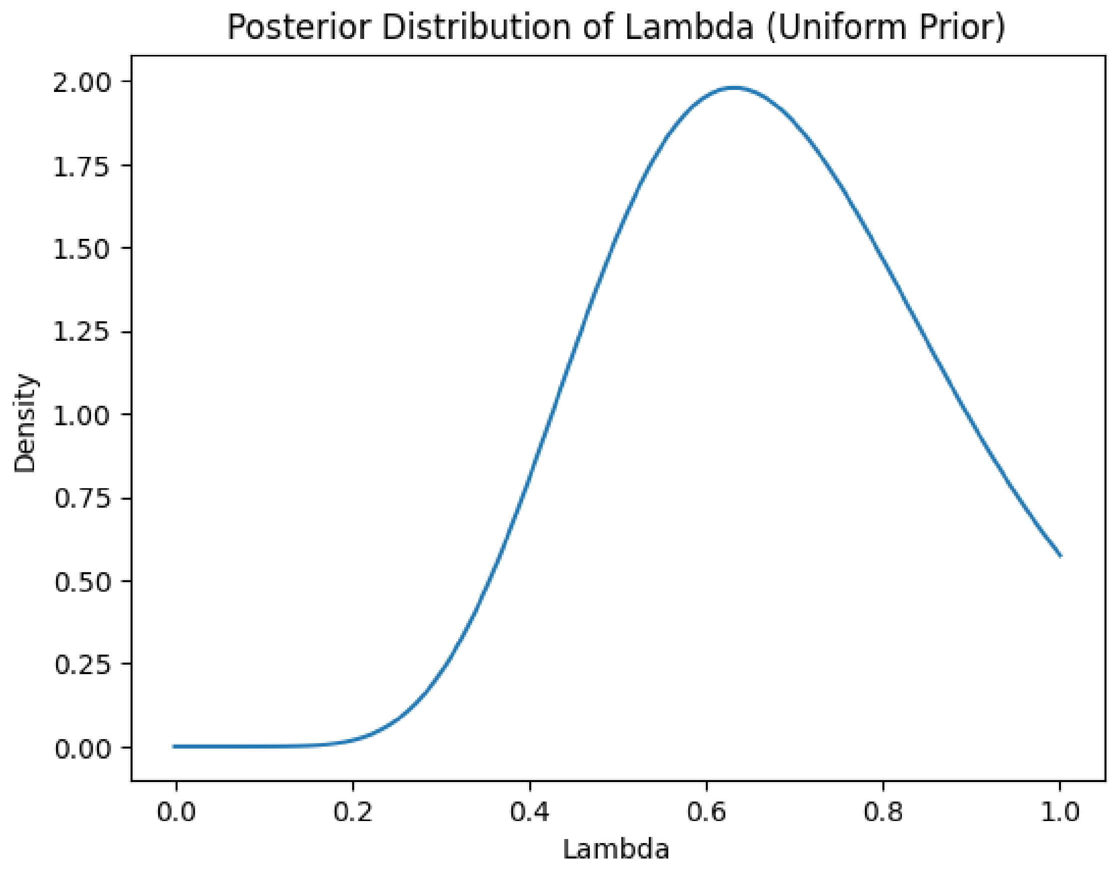
$$f_{\lambda|X}(\lambda|x_1, x_2, ..., x_n) \; \alpha \; \lambda^n e^{-\lambda s}.1$$

LIKELIHOOD

SIMULATION USING MCMC







INFERENCES

- sample mean 0.6605733757631754
- actual mean 0.9102556273667239
- There is not always the guarantee that Markov Chain converges to a good approximation.

PROBLEMS WITH POSTERIOR ANALYSIS

TAKE A CLOSER LOOK AT THE NORMALIZING FACTORS

BERNOULLI DISTRIBUTION POSTERIOR

$$f_{p|X}(p|x_1, x_2, \dots \dots \dots x_n) = \frac{p^s(1-p)^{n-s}.1}{B(s+1, n-s+1)}$$

Very Difficult to calculate

EXPONENTIAL DISTRIBUTION POSTERIOR

$$f_{\lambda|X}(\lambda|x_1, x_2, \dots, x_n) = \frac{\lambda^n e^{-\lambda s}.1}{s^{-(n+1)}(\Gamma(n+1)-\Gamma(n+1,s))}$$

Very Difficult to calculate

APPLICATION IN LINEAR REGRESSION

Problem - Let the input feature is x and target variable is y. The parameters are w and b, such that prediction is $\hat{y} = wx + b$.

Aim - Given data samples (x, y), find parameters w and b which fit the data optimally.

Let's try to come up with a probabilistic model for this problem statement.

Assumptions for developing a probabilistic model:

- Errors in data samples follow normal distribution N(0, σ2)
- Data samples are IID

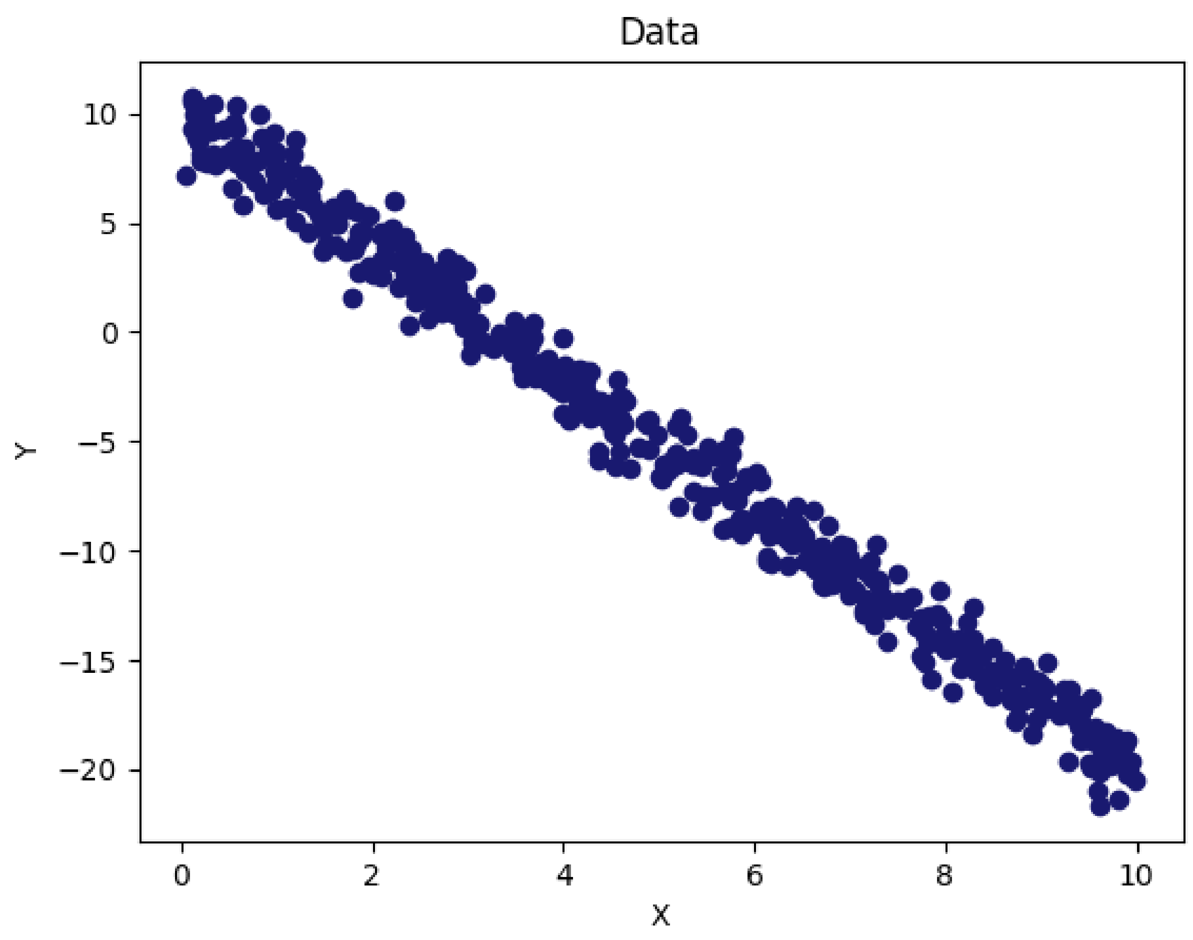


Fig → data samples generated

Approach:

- Start with arbitrary initial values for w and b. Denote vector of parameters as θ.
- Sample a new value, say θ* using the proposal function. We assume that it follows a normal distribution centered around mean = θ (current value) and diagonal covariance matrix with values σ2.
- Each new sample of θ depends on the current sample alone. Hence, they act as states of the Monte Carlo Markov chain.

- Now, we need to compute the posterior probabilities of θ and θ*. As per Bayes rule, it is proportional to the product of Likelihood of data given the parameter and the prior probability of the parameter.

- As a prior belief, we assume that w and b follow normal distribution N(0.5, 0.5). The likelihood function, for observations given the parameter will be the joint probability density function (PDF) of samples. Since they are independent, it is equal to the product of the individual PDFs.

$$For\ parameter\ \theta = [w, b],\ define\ f_\theta(x) = wx\ +\ b.$$

$$Given\ a\ data\ sample\ (x, y),\ P(y|x, \theta, \sigma^2) = N(f_\theta(x), \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y - f_\theta(x))^2}{2\sigma^2}}$$

$$For\ the\ entire\ dataset\ (X, Y)\ where\ i^{th}\ sample\ is\ (x_i, y_i),\ Likelihood\ is$$

$$L(Y|X, \theta, \sigma^2) = P(y_1|x_1, \theta, \sigma^2) \cdot P(y_2|x_2, \theta, \sigma^2) \cdot \ldots \cdot P(y_n|x_n, \theta, \sigma^2)$$

$$= \prod_{i=1}^{n} N(f_\theta(x_i), \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_i - f_\theta(x_i))^2}{2\sigma^2}}$$

- Using the Metropolis-Hastings algorithm, we calculate an acceptance ratio which can be used to accept/reject the proposed sample. The acceptance ratio is posterior probability of θ* divided by posterior probability of θ.

- Finally after n iterations, the samples in the Monte carlo Markov chain can be used to infer the value of θ (parameters w and b). Note that we remove the first quarter of states as the Burn-in period.

- On plotting a histogram, we observe that the value of parameters is roughly distributed as a normal distribution with mean close to the true value.

$$r(\theta^*, \theta) = \frac{L(Y|X, \theta^*) * P(\theta^*)}{L(Y|X, \theta) * P(\theta)}$$

An example of code simulation is demonstrated. Data containing 500 samples is generated from the line y = -3x + 10 by adding small perturbations. After 50,000 iterations following observations were made.

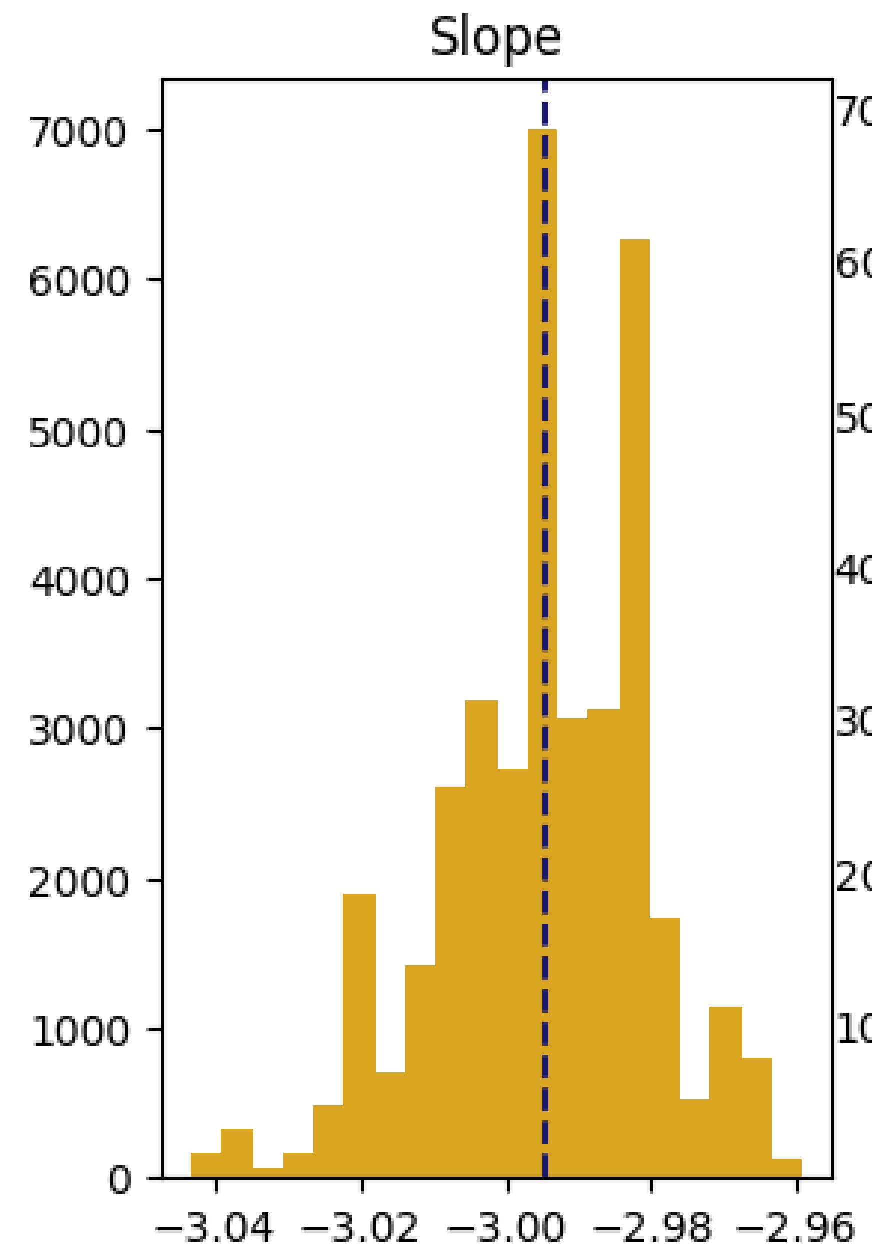


Fig → distribution and mean of parameters w and b

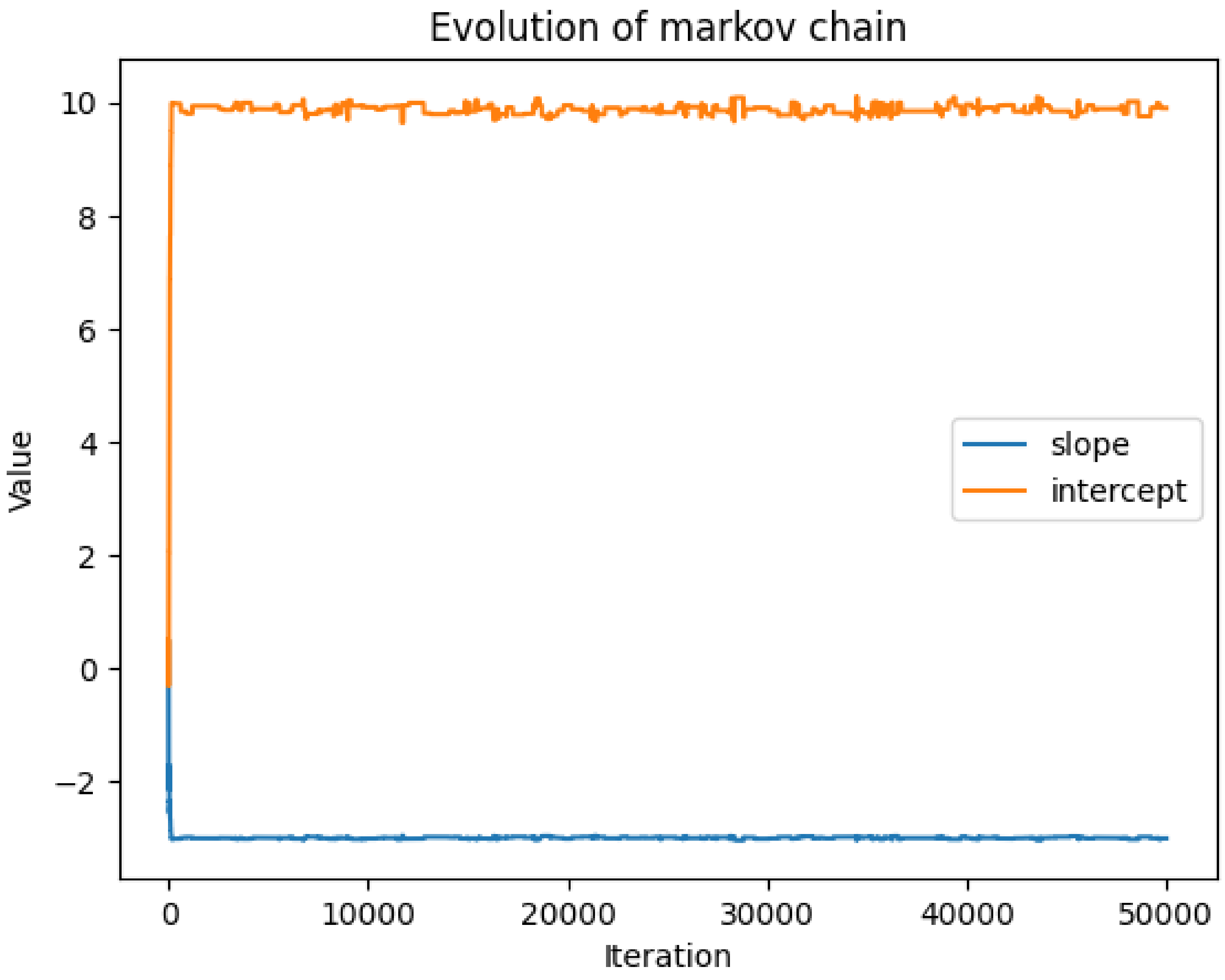


Fig → random walk of parameters w and b

THANK YOU

Members:

Arnav Gupta (2021236)

Chaitanya Garg (2021248)

Karan Gupta (2021258)

Raghav Sakuja (2021274)

Rudra Jyotirmay (2021280)

Shivesh Gulati (2021286)

Tanishq Jain (2021294)

https://github.com/RaghavSakhuja/EMCEE.git

REFERNCES

- https://web.stanford.edu/class/stats200/Lecture20.pdf
- https://www.researchgate.net/publication/297897462_A_simple_introduction_to_Markov_Chain_Monte-Carlo_sampling
- https://www.youtube.com/watch?v=yApmR-c_hKU
- https://www.geeksforgeeks.org/implementation-of-bayesian-regression/
- https://medium.com/@tinonucera/bayesian-linear-regression-from-scratch-a-metropolis-hastings-implementation-63526857f191