HW5

Haojie Liu

2023-12-04

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.3     v purrr   1.0.2
## v tibble  3.2.1     v dplyr   1.1.2
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.4     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(ggplot2)
library(diagram)
```

```
## Loading required package: shape
```

Problem 1: The weather on any given day on a tropical island could be rainy, sunny or cloudy, and the probability of tomorrow's weather only depends on today's weather and not any other previous days. Suppose that we obtained the transition probabilities as below.

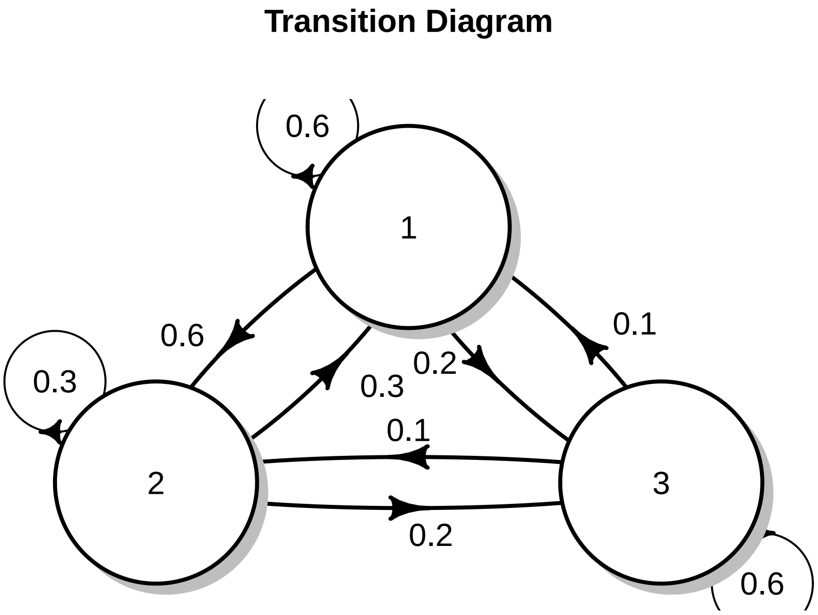
P(Rainy tomorrow — Rainy today) = 0.6
P(Rainy tomorrow — Cloudy today) = 0.6
P(Rainy tomorrow — Sunny today) = 0.2
P(Cloudy tomorrow — Rain today) = 0.3
P(Sunny tomorrow — Cloudy today) = 0.1
P(Cloudy tomorrow — Sunny today) = 0.2

(a) Please find the transition matrix and draw a transition state diagram, with state space {1: Rainy, 2: Cloudy, 3: Sunny}.

```r
M <- data.frame("1" = c(0.6, 0.6,0.2),
           "2" = c(0.3, 0.3,0.2),
           "3" = c(0.1, 0.1,0.6))
colnames(M) <- c(1,2,3)
rownames(M) <- c(1,2,3)

M
```

```
##     1   2   3
## 1 0.6 0.3 0.1
## 2 0.6 0.3 0.1
## 3 0.2 0.2 0.6
```

```
plotmat(M,
        pos = c(1,2),
        self.cex = 0.5,
        self.shiftx = c(-0.1, -0.1, 0.1),
        self.shifty = c(0.1, 0.1, -0.1),
        main = "Transition Diagram")
```



(b) Suppose today is sunny. What is the expected weather two days from now?

```
M <- matrix(unlist(M), nrow = 3)

two_step_M <- (M%*%M)[3,]
two_step_M
```

```
## [1] 0.36 0.24 0.40
```

```
print(paste("Rainy:", two_step_M[1], "Cloudy:", two_step_M[2], "Sunny:", two_step_M[3]))
```

```
## [1] "Rainy: 0.36 Cloudy: 0.24 Sunny: 0.4"
```

(c) Is this Markov chain irreducible and aperiodic? Explain. Can you find a stationary distribution? If so, please show and explain how you find it.

It is possible to reach any state from any state, which means that the weather can change from any given condition to any other condition. Therefore, it is Irreducible. Also, the greatest common divisor of the lengths of cycles in the state space is 1, indicating that the chain does not get ‘stuck’ in cycles of a fixed length. So it is Aperiodic.

```
M2 <- M%*%M

for (i in 2:10000){
  M2 <- M2%*%M
}

M2
```

```
##      [,1] [,2] [,3]
## [1,] 0.52 0.28  0.2
## [2,] 0.52 0.28  0.2
## [3,] 0.52 0.28  0.2
```

Problem 2: Two urns A and B contain a total of N balls. Assume that at time t there were exactly k balls in A. At time t + 1, an urn is selected at random in proportion to its contents (i.e., A is chosen with probability k/N and B is chosen with probability (N - k)/N). Then one of the N balls is randomly selected and placed in the chosen urn.

Let X_t denote the number of balls in urn A at time t, so $X_t, t \geq 0$ defines a Markov chain. Determine the transition matrix for this Markov chain.

Since we have that

$$P(X_{t+1} = k - 1 | X_t = k) = \frac{k}{N}$$
$$P(X_{t+1} = k + 1 | X_t = k) = \frac{N - k}{N}$$

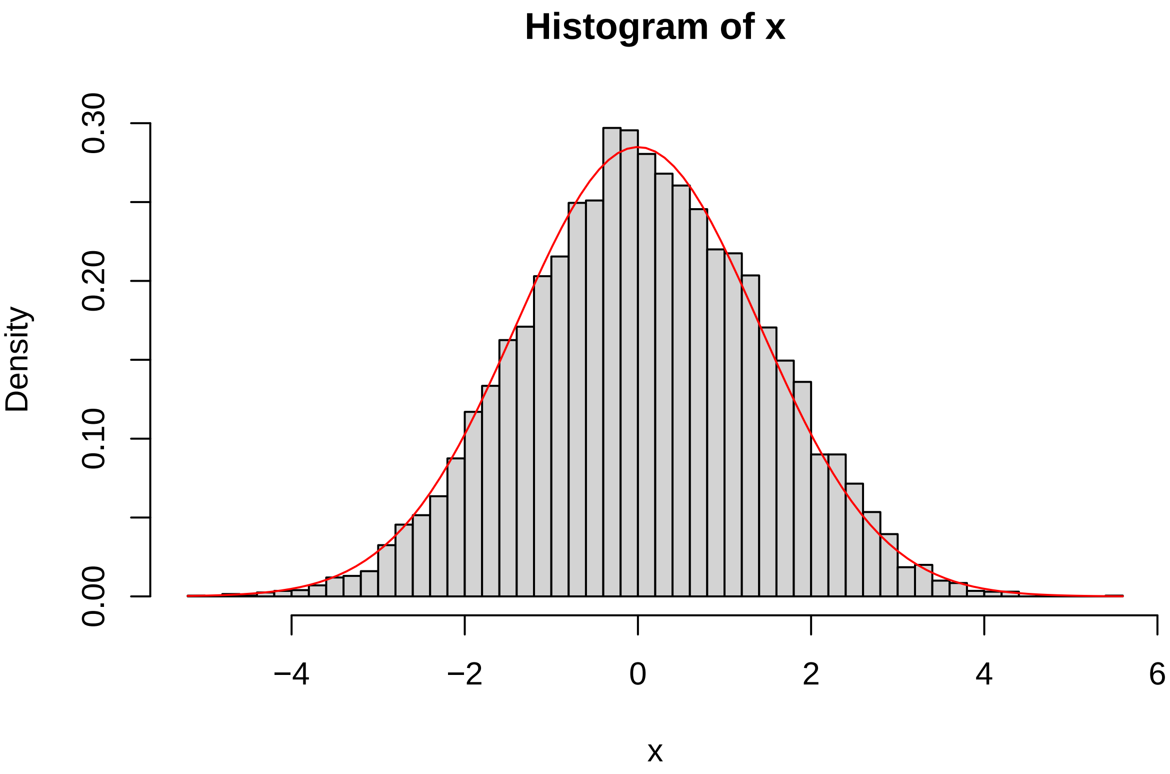
Then when $k = 0$, $P(X_{t+1} = k + 1) = 1; P(X_{t-1} = k + 1) = 0$, and when $k = 1$, $P(X_{t+1} = k + 1) = 1/N; P(X_{t-1} = k + 1) = (N - 1)/N$ So on.

Therefore, we will have the following transition matrix:

$$\begin{pmatrix} 0 & 1 & 0 & 0 & ... & 0 \\ 1/N & 0 & (N-1)/N & 0 & ... & 0 \\ 0 & 2/N & 0 & (N-2)/N & ... & 0 \\ | & & & & & | \\ 0 & 0 & 0 & ... & 1 & 0 \end{pmatrix}$$

Problem 3: Suppose the Markov chain is defined by $X^{(t+1)} = \alpha X^t + \epsilon_t$, where $\epsilon_t \sim N(0,1)$. Write R code to simulate this Markov chain with $X^{(0)} \sim N(0,1)$ for $t \leq 10^4$ and $\alpha = 0.7$. Check if your sample fits the distribution $N(mean = 0, sd = 1/\sqrt{1 - \alpha^2})$

```
a <- 0.7
t <- 10^4

x <- c(rnorm(1,0,1))

for(i in 2:t){
  e <- rnorm(1,0,1)
  x[i] <- a * x[i-1] + e
}

hist(x,freq = FALSE, ylim = c(0,0.3),breaks=50)
curve(dnorm(x, mean = 0, sd = 1/sqrt(1-a^2)), add = TRUE, col = "red")
```



Problem 4: We want to model the number of siblings people have in a certain population. We can model the number of siblings a person has as a Poisson random variable Y for some unknown mean parameter λ. The probability mass function of $Y \sim Pois(\lambda)$ is

$$f(y|\lambda) = \frac{e^{-\lambda}\lambda^y}{y!},\ for\ y = 0, 1, 2, ...$$

Suppose, before observing any data, we model our prior beliefs about λ by a gamma distribution $Gamma(\alpha, \beta)$ with hyperparameters $\alpha, \beta > 0$, i.e,

$$\pi(\lambda) = \frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda},\ for\ \lambda > 0$$

where $\Gamma(z) = \int x^{z-1}e^{-x}dx$. Note that + The prior mean is $E(\lambda) = \frac{\alpha}{\beta}$ + The prior variance is $Var(\lambda) = \frac{\alpha}{\beta^2}$ + The prior mode is $mode(\lambda) = \frac{\alpha-1}{\beta}$ Suppose we observe data $y_1, y_2, ...y_n \sim^{iid} Pois(\lambda)$. Let $y = (y_1, y_2, ...y_n)$

(a) Show that the gamma distribution is a conjugate prior for the Poisson likelihood. More specifically, show that the posterior distribution $\pi(\lambda|y)$ is of the form

$$\pi(\lambda|y) \sim Gamma(\alpha + \Sigma_{i=1}^n y_i, \beta + n)$$

Proof:

$$L(\lambda|y) = \frac{e^{-n\lambda}\lambda^{\Sigma y_i}}{\prod y!}$$

$$\pi(\lambda|y) \propto \frac{e^{-n\lambda}\lambda^{\Sigma y_i}}{\prod y!}\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}$$

$$\pi(\lambda|y) \propto \lambda^{\Sigma y_i+\alpha-1}e^{-(n+\beta)\lambda}$$

$$\pi(\lambda|y) \sim Gamma(\alpha + \Sigma_{i=1}^n y_i, \beta + n)$$

(b) Show that the posterior mean $E(\lambda|y)$ is a weighted average of the prior mean and the sample mean $\bar{y} = \frac{1}{n}\Sigma_{i=1}^n y_i$. That is, find the weight w so that

$$E(\lambda|y) = w\frac{\alpha}{\beta} + (1-w)\bar{y}$$

Proof:

$$E(\lambda|y) = \frac{\alpha + \Sigma_{i=1}^n y_i}{\beta + n}$$

$$w\frac{\alpha}{\beta} + (1-w)\bar{y} = \frac{\alpha + n\bar{y}}{\beta + n}$$

$$w = \frac{\beta}{\beta + n}$$

(c) What is $lim_{n\to\infty}E(\lambda|y)$? What does this limit represent?

$$lim_{n\to\infty}E(\lambda|y) = \bar{y}$$

This limit represents that as we get more data, the posterior mean converges to the sample mean, and the influence of the prior diminishes.

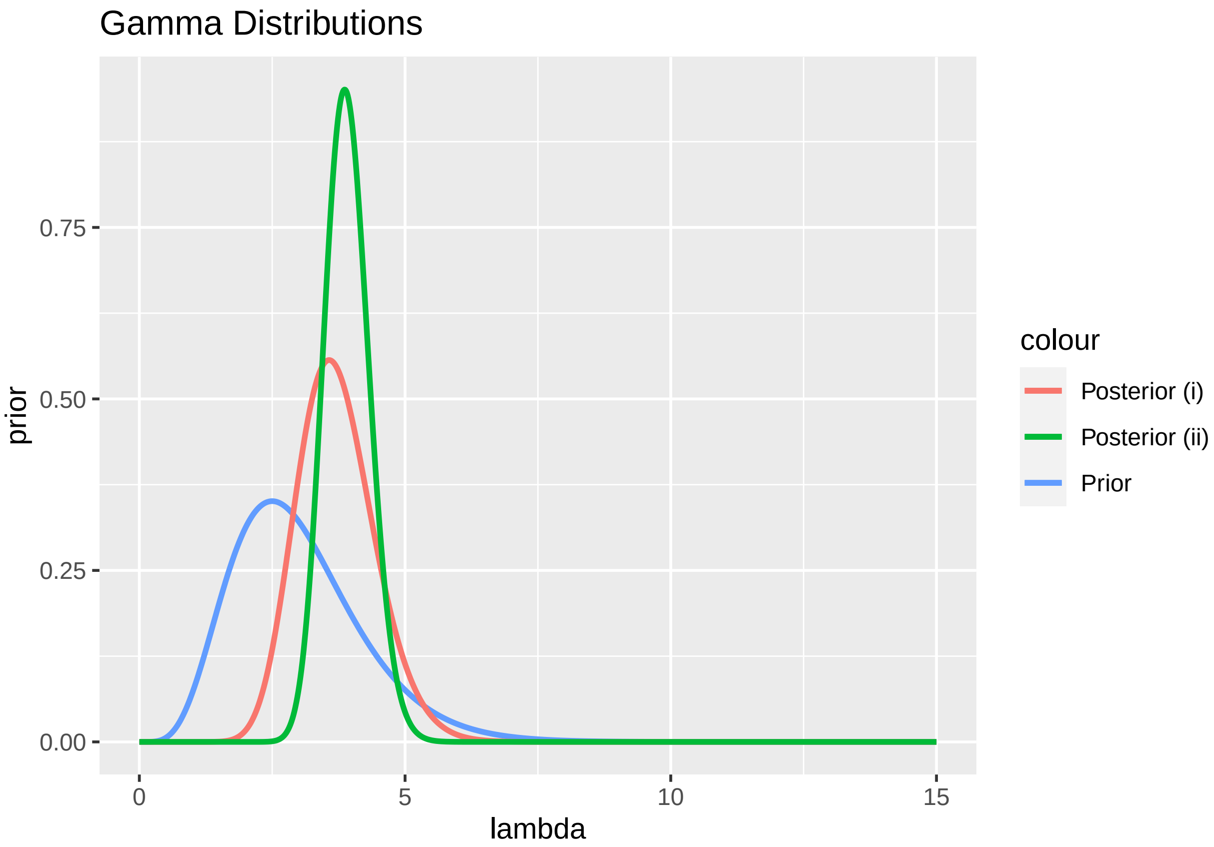
(d) Use the `dgamma()` function in R to visualize the prior and posterior densities for hyperparameters $\alpha = 6, \beta = 2$ when the data is summarized by:

- $\Sigma_{i=1}^{n} y_i = 20$ and $n = 5$
- $\Sigma_{i=1}^{n} y_i = 80$ and $n = 20$ Represent the sample mean on the plot to show how the posterior distribution is a compromise between the prior distribution and the data. Clearly indicate the different components of your plot

Hint: For a $Gamma(\alpha, \beta)$ distribution, α is the shape parameter and β is the rate parameter.

```r
lambda <- seq(0,15,length.out = 1000)
prior <- dgamma(lambda, 6, 2)
y1_post <- dgamma(lambda, 26, 7)
y2_post <- dgamma(lambda, 86, 22)

df <- data.frame(lambda, prior, y1_post, y2_post)

df %>%
  ggplot(aes(lambda)) +
    geom_line(aes(y = prior, color = "Prior"), size = 1) +
    geom_line(aes(y = y1_post, color = "Posterior (i)"), size = 1)+
    geom_line(aes(y = y2_post, color = "Posterior (ii)"), size = 1)+
    labs(title = "Gamma Distributions")
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



(e) For scenarios (i) and (ii) in part (d), find and interpret 95% quantile-based credible intervals.

```
y1_lower <- qgamma(0.025, shape = 26, rate = 7)
y1_upper <- qgamma(0.975, shape = 26, rate = 7)
y2_lower <- qgamma(0.025, shape = 86, rate = 22)
y2_upper <- qgamma(0.975, shape = 86, rate = 22)

paste("(i)","95% credible interval: (", y1_lower, ", ", y1_upper, ")\n")
```

```
## [1] "(i) 95% credible interval: ( 2.42629474508519 ,  5.27213309964719 )\n"
```

```
paste("(ii)","95% credible interval: (", y2_lower, ", ", y2_upper, ")\n")
```

```
## [1] "(ii) 95% credible interval: ( 3.12676814746227 ,  4.77744619631359 )\n"
```

Problem 5: The Cauchy distribution is a continuous probability distribution. It is often used in statistics as the canonical example of a "pathological" distribution, and the density is defined as:

$$f(x|\gamma, \eta) = \frac{1}{\gamma\pi[1 + (\frac{x-\eta}{\gamma})]^2}, \quad -\infty < x < \infty, \gamma > 0$$

where η is the location parameter, specifying the location of the peak of the distribution.

(a) Please write an algorithm using the Metropolis-Hastings sampler to generate samples from the Cauchy distribution with $\gamma = 1$ and $\eta = 0$. You may use $N(\mu, \sigma)$ as the proposal distribution.

1. Set the random seed for reproducibility seed <- 123
2. Define the number of samples to generate n <- 10000
3. Initialize an array X of length n to store the samples X[n] for i from 1 to n do X[i] <- 0 end for
4. Define the standard deviation for the proposal distribution C <- 1
5. Define the Cauchy distribution function function cauchy(x, gamma = 1, eta = 0) return 1 / (gamma * pi * (1 + ((x - eta) / gamma)^2)) end function
6. Set the starting value of the chain X[1] <- 0
7. Perform the Metropolis-Hastings sampling for i from 2 to n do Generate a proposed sample Y from the normal distribution with mean X[i-1] and standard deviation C Y <- normal(mean = X[i-1], sd = C)

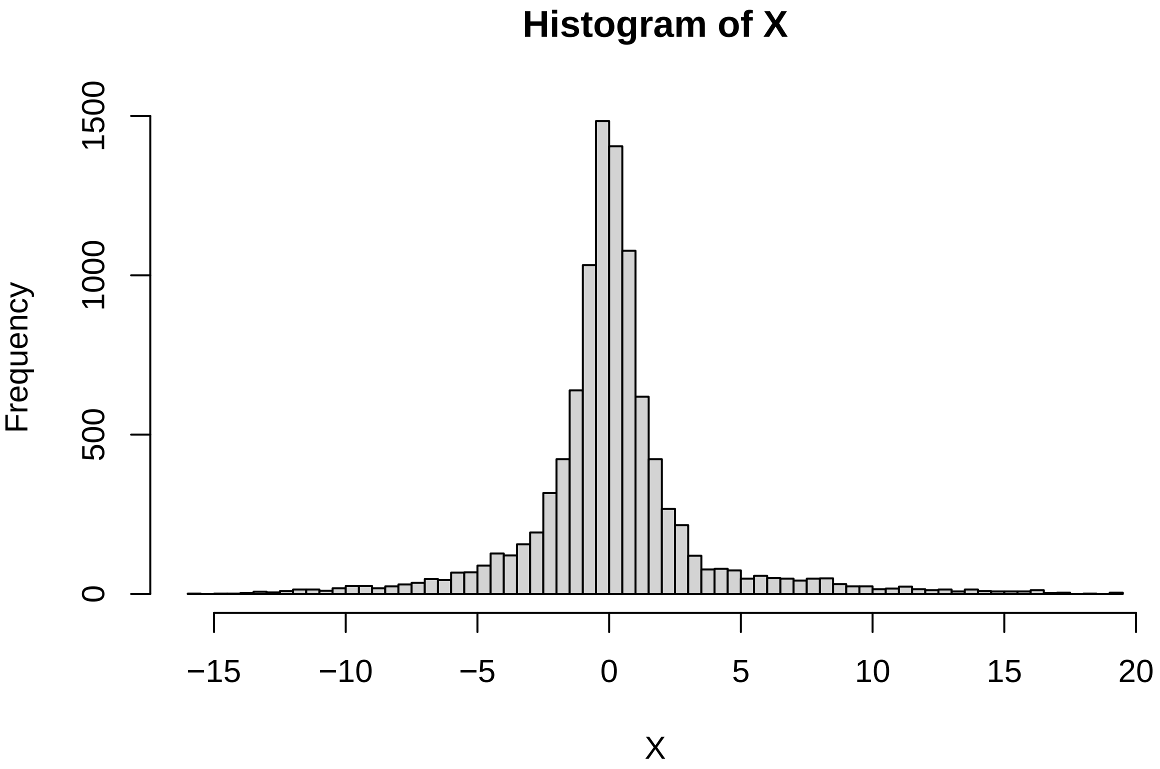
   ```
   Calculate the acceptance ratio r
   r <- minimum of 1 and (cauchy(Y) / cauchy(X[i-1]))

   Generate a uniform random number U
   U <- uniform(0, 1)

   If U is less than or equal to r, accept the proposal
   if U < r then
       X[i] <- Y
   else
       Keep the current sample
       X[i] <- X[i - 1]
   end if
   ```

   end for
8. Plot the histogram of the samples X with 50 breaks PlotHistogram(X, breaks = 50)

(b) Implement your algorithm in R to generate 10,000 samples and show the histogram.

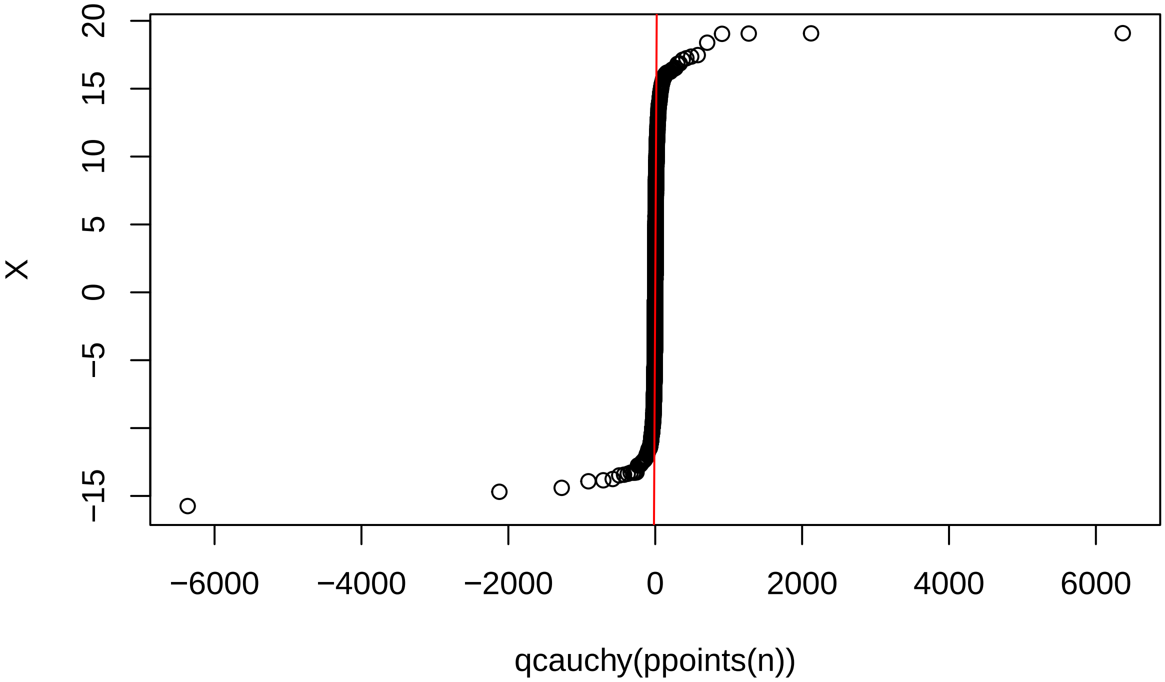
```r
set.seed(123)

n <- 10000
X <- numeric(n)
C <- 1
X[1] <- 0

cauchy <- function(x, gamma = 1, eta = 0) {
  1 / (gamma * pi * (1 + ((x - eta) / gamma)^2))
}

for(i in 2:n){

  Y <- rnorm(1, mean = X[i-1], sd = C)
  r <- min(1, cauchy(Y)/cauchy(X[i-1]))
  if (runif(1) < r) {
    X[i] <- Y
  } else {
    X[i] <- X[i - 1]
```

```
  }
}

hist(X,breaks = 50)
```

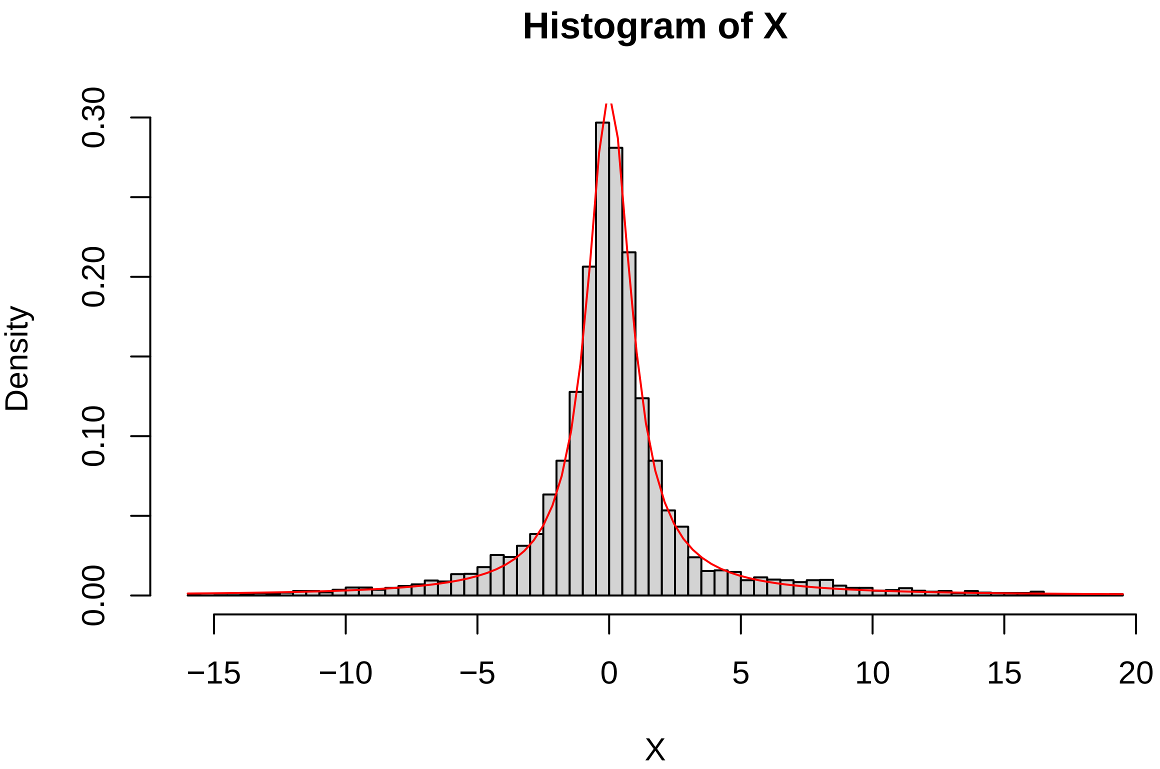


(c) Compare the generated samples with `qcauchy` in R. Please conduct a exploratory analysis to determine if your samples agree with the output of `qcauchy`.

```
qqplot(qcauchy(ppoints(n)),X)
abline(0, 1, col = "red")
```



```
hist(X,breaks = 50,freq=FALSE)
curve(dcauchy(x), col = "red", add = TRUE)
```



Problem 6: The Rayleigh distribution is a continuous probability density function, and is widely used for modeling the lifetime of an object depending on its age. The Rayleigh density is defined as:

$$f(x) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)}, x \geq 0$$

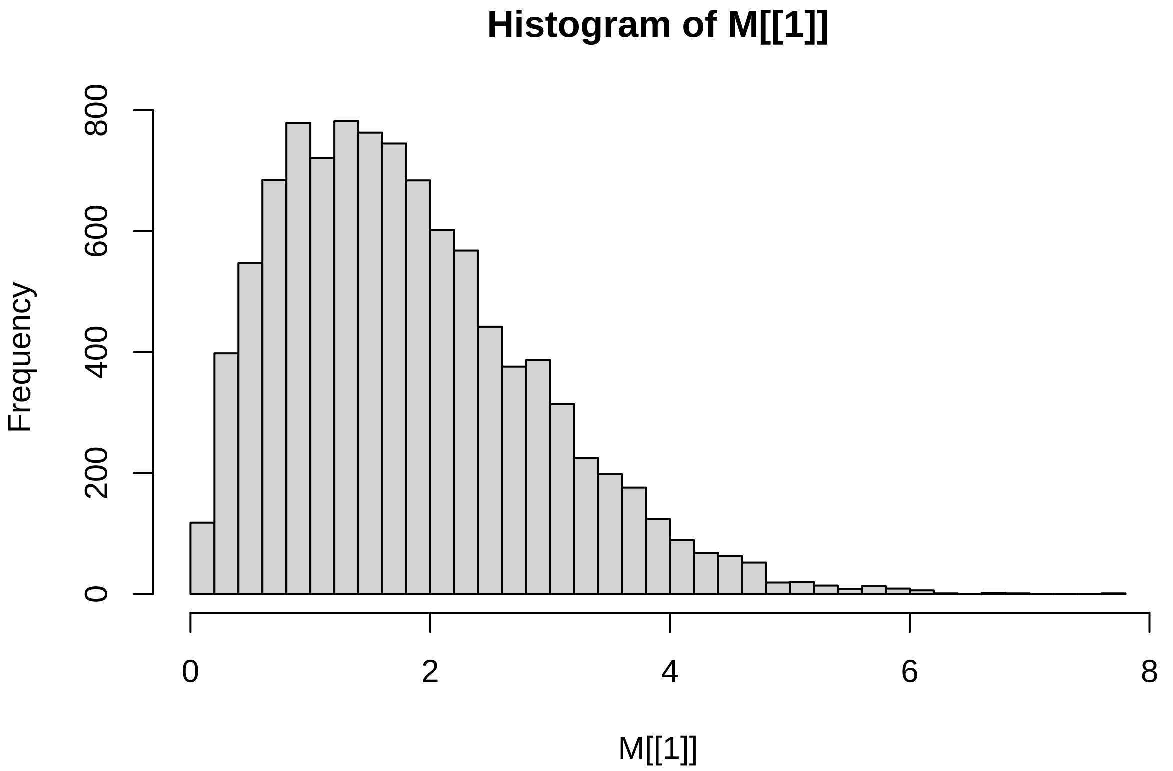
, where σ is the scale parameter of the distribution.

(a) Design a Metropolis algorithm to sample from the Rayleigh distribution with the χ^2 distribution as the proposal distribution. Implement this algorithm with $\sigma = 2$, 4, and 6, respectively. Compare the performance of these three cases. Please give the acceptance rate, and plot the sample vs. the time index and the histograms.

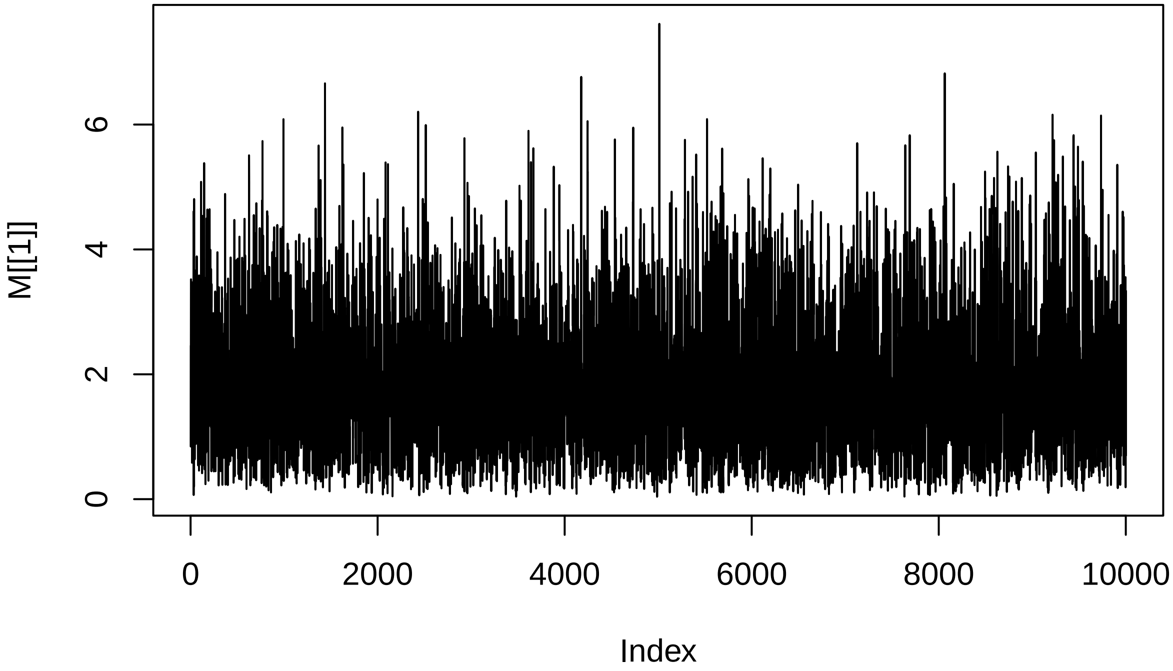
1. Initialization: Set up parameters for the Monte Carlo Markov Chain (MCMC) simulation, including the number of iterations, initial values, sigma values for the Rayleigh distribution, and counters for tracking accepted values in each chain.
2. Rayleigh Distribution Function: Implement a function to calculate the Rayleigh distribution given a value and a sigma parameter.
3. MCMC Simulation For each sigma value in the set [2, 4, 6] Run an MCMC simulation for a specified number of iterations. In each iteration, propose a new value from a chi-squared distribution. Decide whether to accept this new value based on the acceptance ratio calculated from the Rayleigh distribution. Keep track of accepted values and store the resulting chain.

```r
set.seed(123)

n <- 10000
X <- numeric(n)
C <- c(2,4,6)
X[1] <- rexp(1, 1)
M <- list(3)
count <- c(0,0,0)

Rayleigh <- function(x, sigma) {
  if (x < 0) return(0)
  x / sigma^2 * exp(-x^2 / (2 * sigma^2))
}

for (j in 1:3) {
  for(i in 2:n){
  Y <- rchisq(1, 2)
  r <- min(1, Rayleigh(Y,C[j])/Rayleigh(X[i-1],C[j]))
  if (runif(1) < r) {
    X[i] <- Y
    count[j] <- count[j] + 1
  } else {
    X[i] <- X[i - 1]
  }
  }

  M[[j]] <- X
  X <- numeric(n)
  X[1] <- rexp(1, 1)
}
```

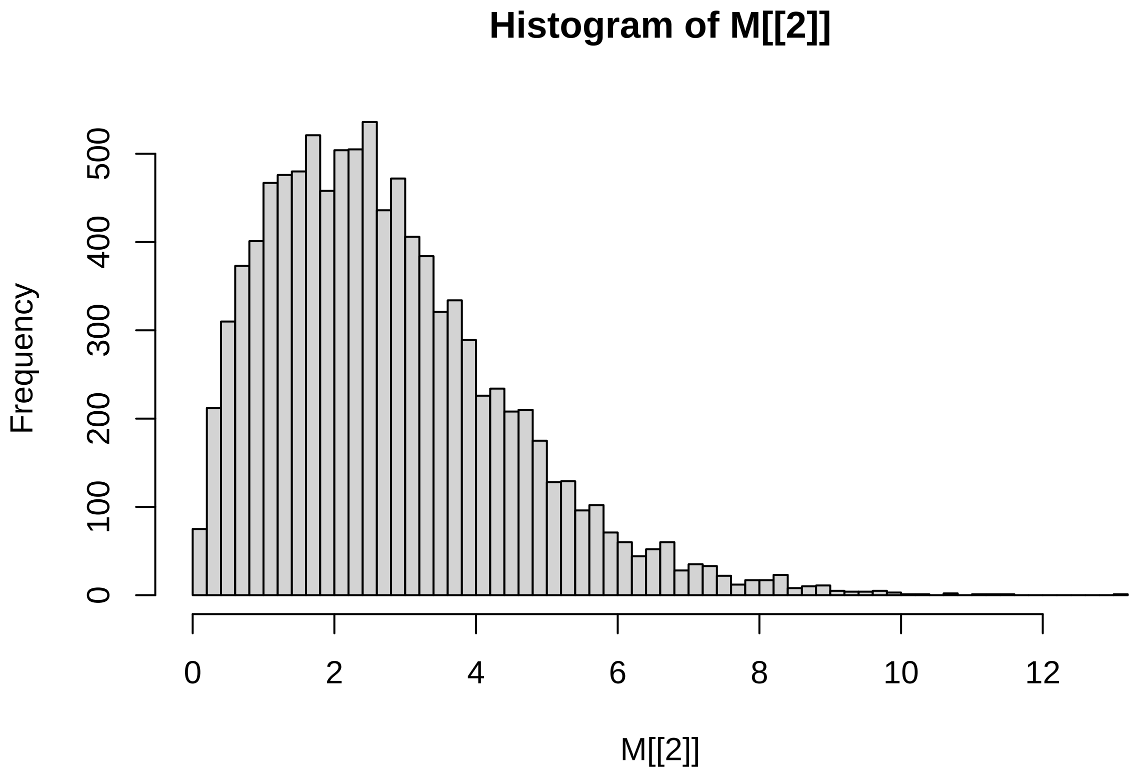
```r
hist(M[[1]],breaks = 50)
```



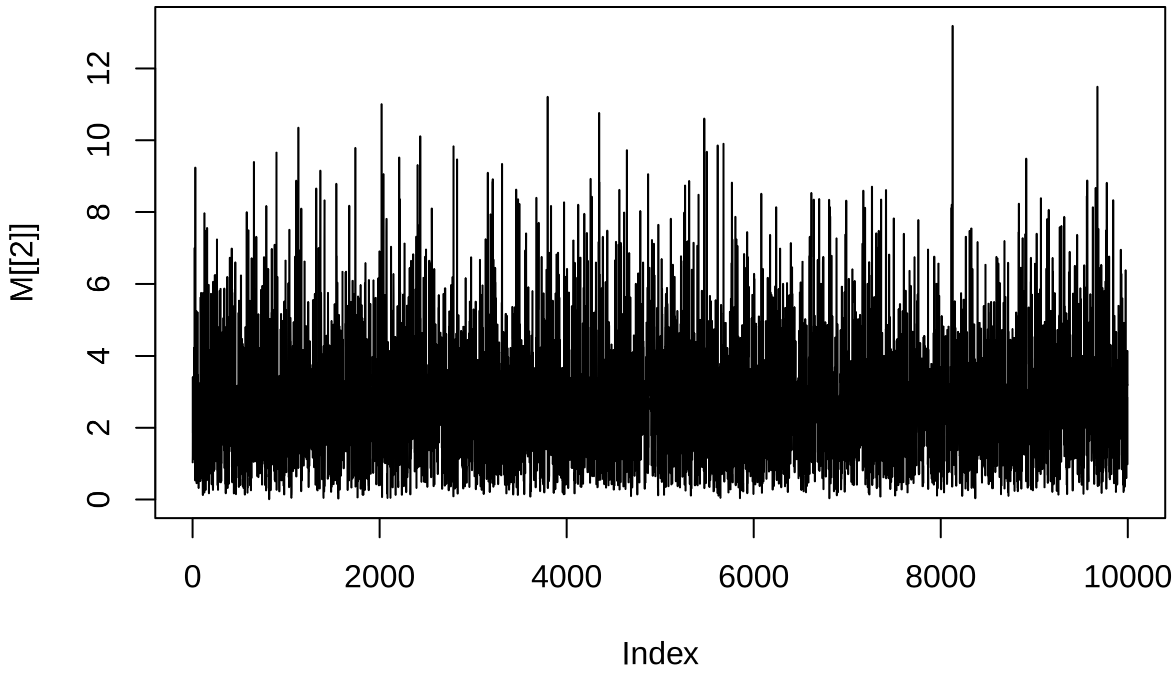
```r
plot(M[[1]], type = "l")
```



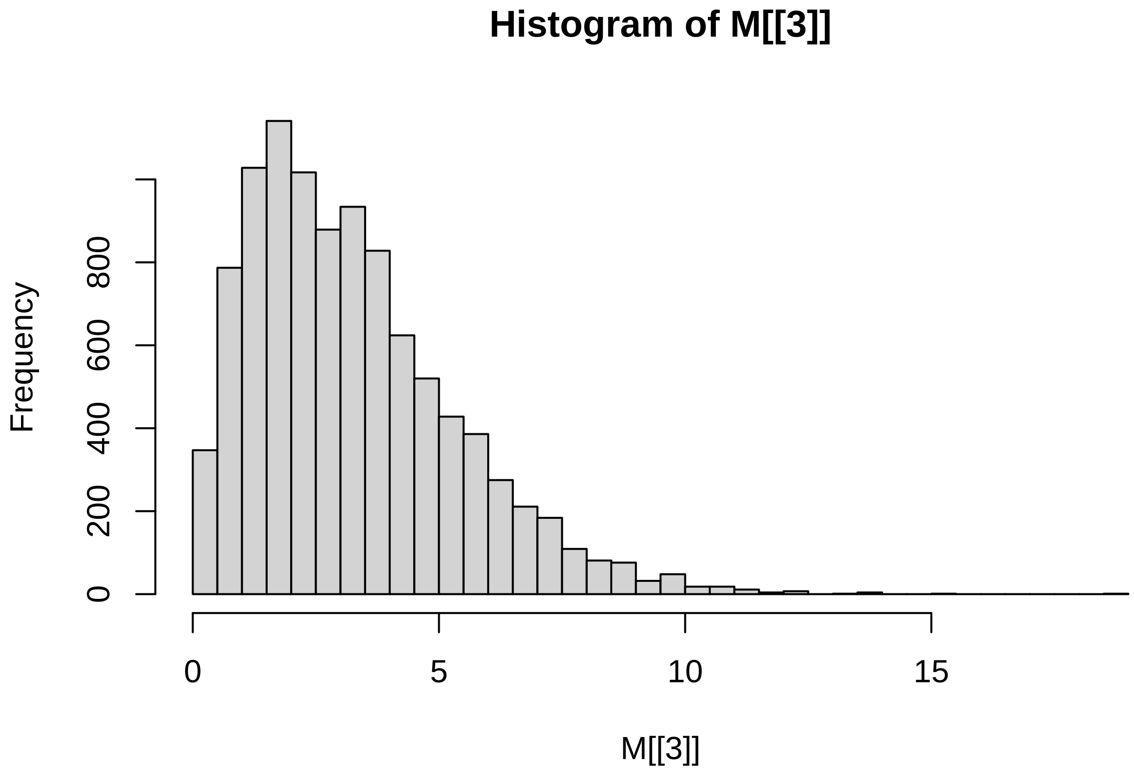
```
hist(M[[2]],breaks = 50)
```



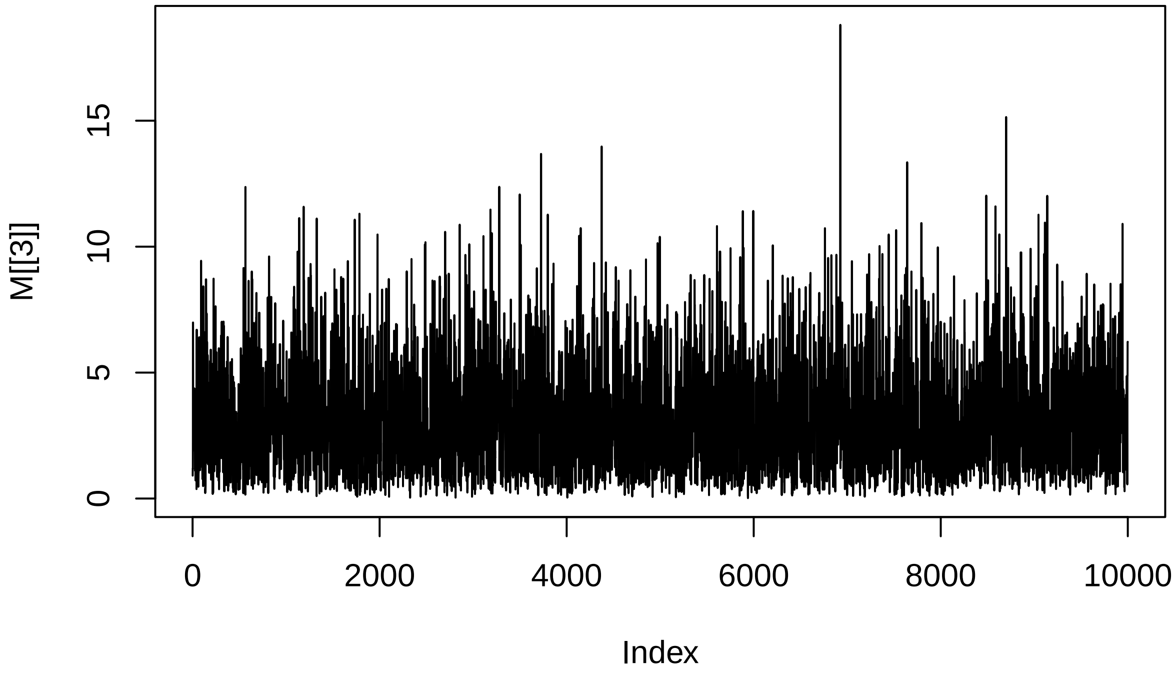
```
plot(M[[2]], type = "l")
```



```
hist(M[[3]],breaks = 50)
```



```
plot(M[[3]], type = "l")
```



```
paste("Sigma = 2, Accept rate:", count[1]/(n-1))
```

```
## [1] "Sigma = 2, Accept rate: 0.664866486648665"
```

```r
paste("Sigma = 4, Accept rate:", count[2]/(n-1))
```

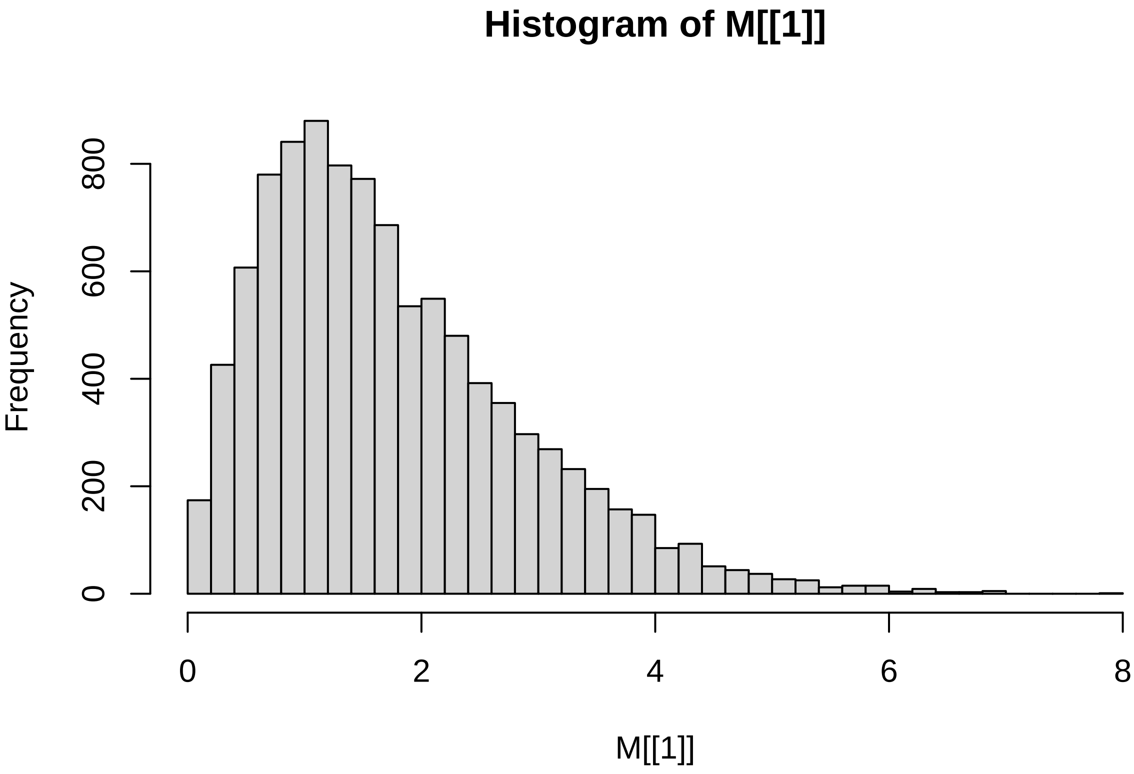
```
## [1] "Sigma = 4, Accept rate: 0.64026402640264"
```

```r
paste("Sigma = 6, Accept rate:", count[3]/(n-1))
```

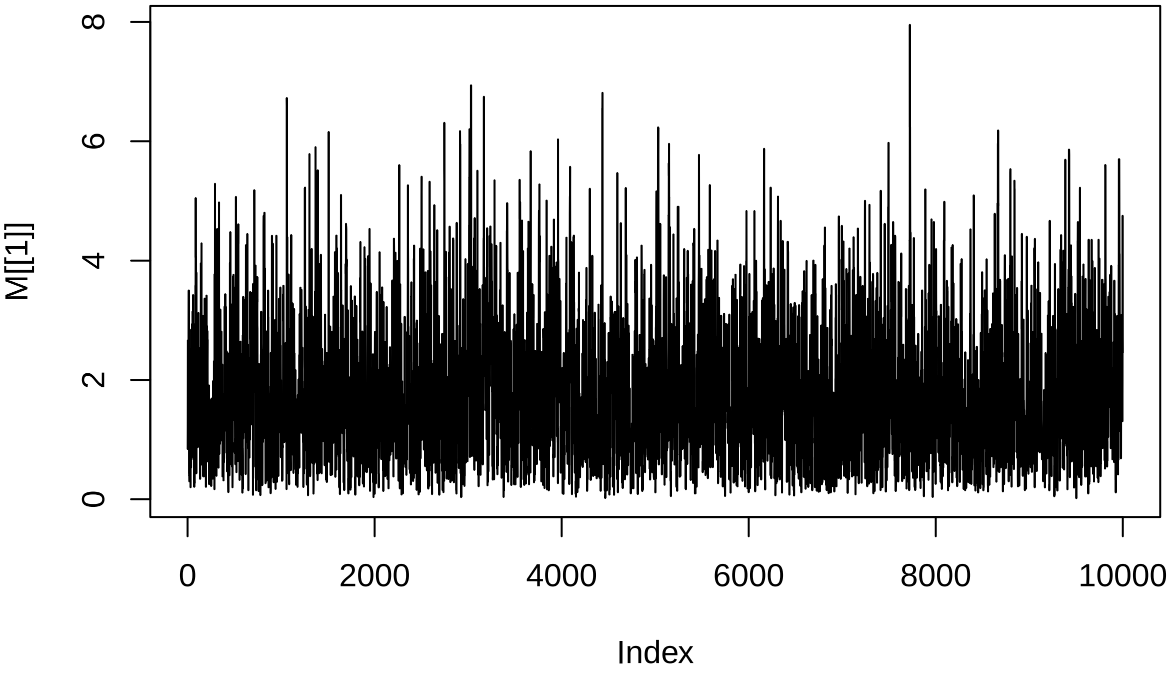
```
## [1] "Sigma = 6, Accept rate: 0.593859385938594"
```

(b) Repeat the above problem using the Gamma distribution as the proposal distribution with shape parameter as X_t and rate parameter as 1.

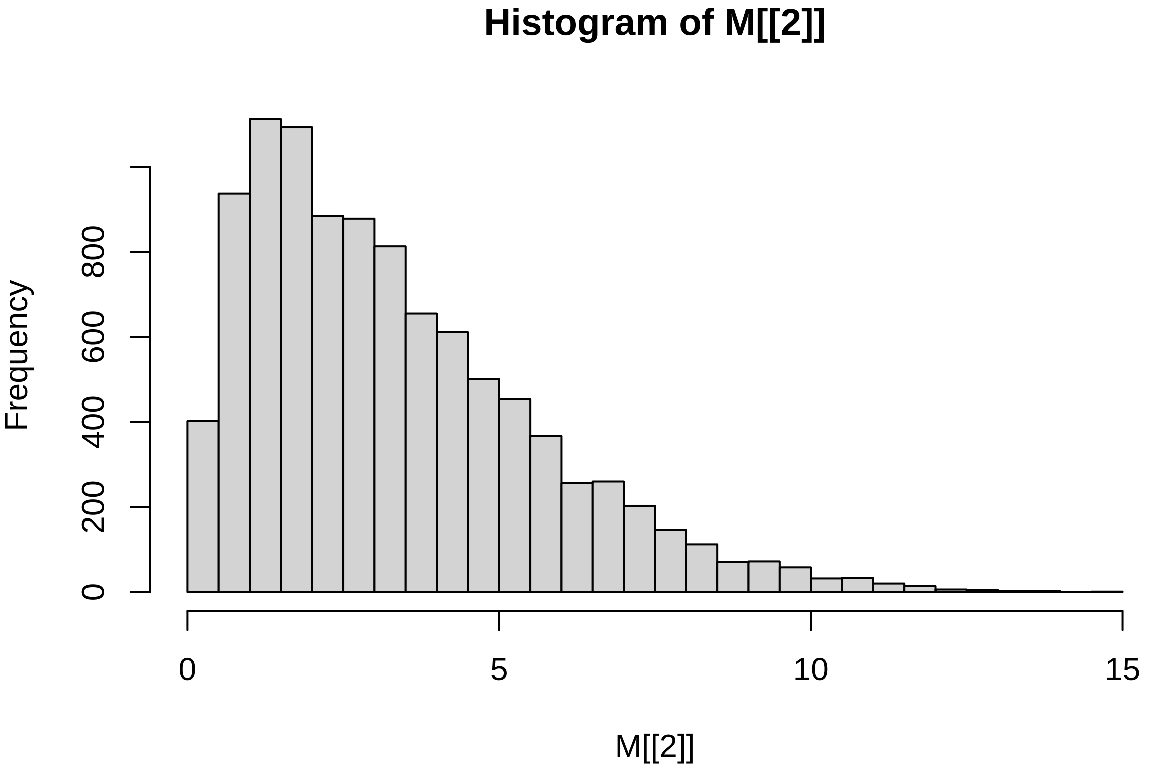
```r
set.seed(123)

n <- 10000
X <- numeric(n)
C <- c(2,4,6)
X[1] <- rexp(1, 1)
M <- list(3)
count <- c(0,0,0)

for (j in 1:3) {
  for(i in 2:n){
  Y <- rgamma(1, shape = max(1, X[i - 1]), rate = 1)
  r <- min(1, Rayleigh(Y,C[j])/Rayleigh(X[i-1],C[j]))
  if (runif(1) < r) {
    X[i] <- Y
    count[j] <- count[j] + 1
  } else {
    X[i] <- X[i - 1]
  }
  }

  M[[j]] <- X
  X <- numeric(n)
  X[1] <- rexp(1, 1)
}

hist(M[[1]],breaks = 50)
```



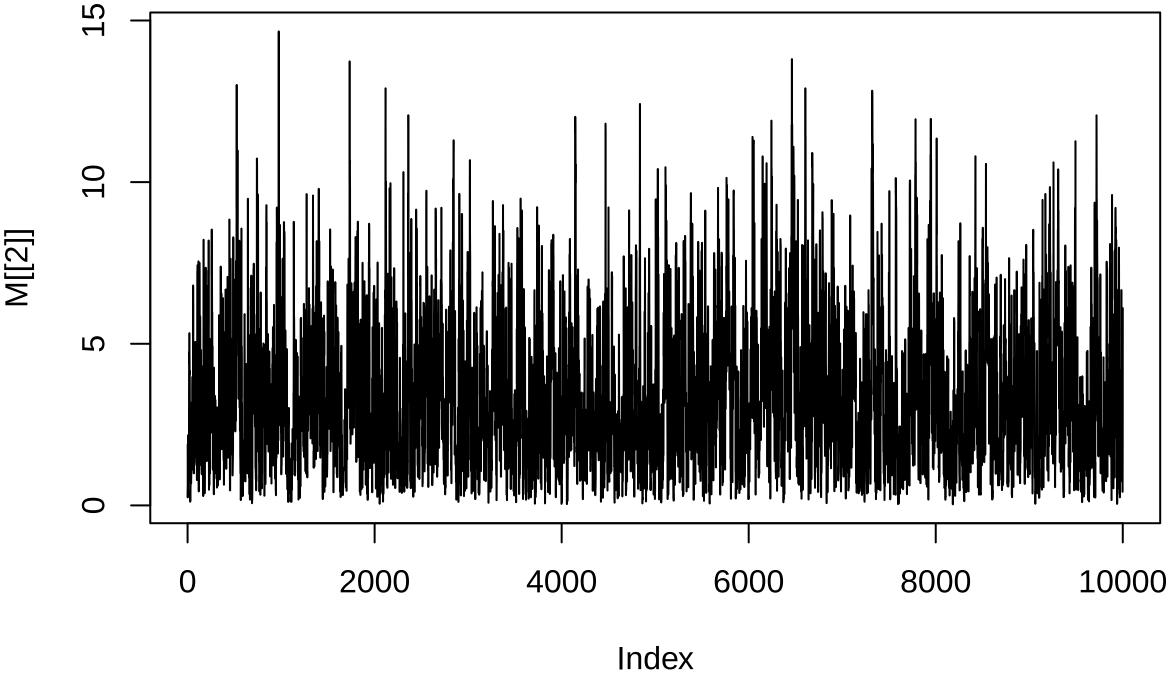
```
plot(M[[1]], type = "l")
```



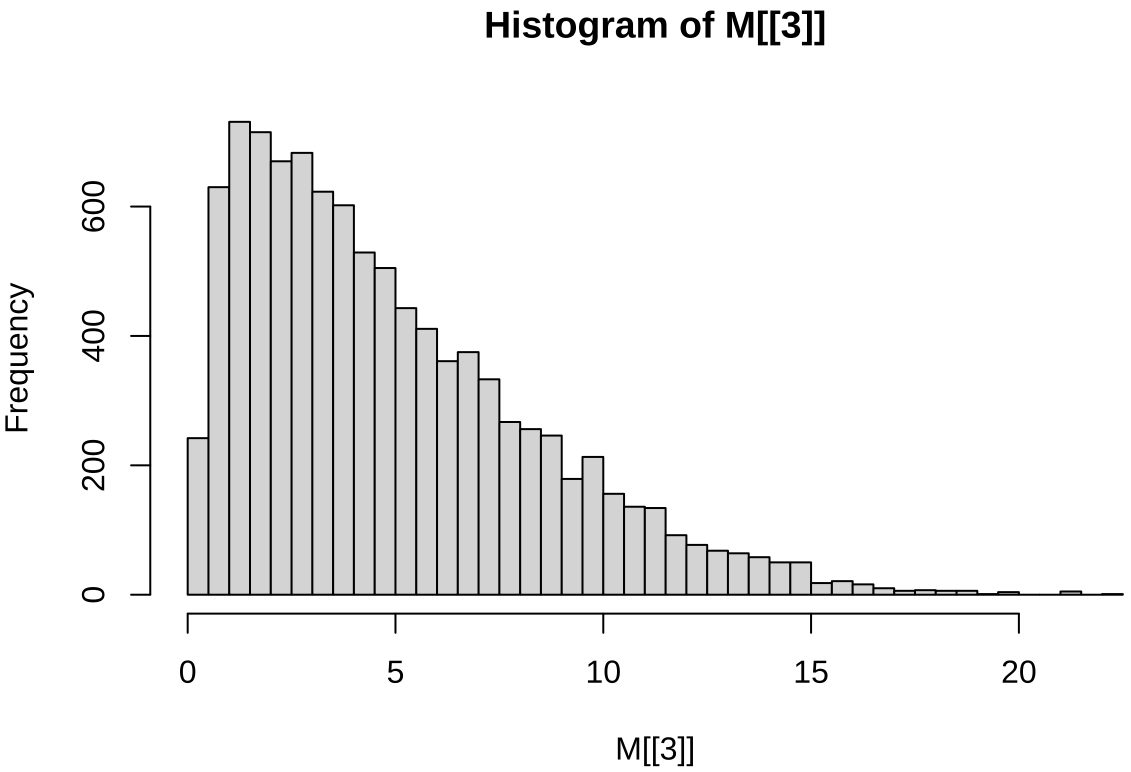
```
hist(M[[2]],breaks = 50)
```



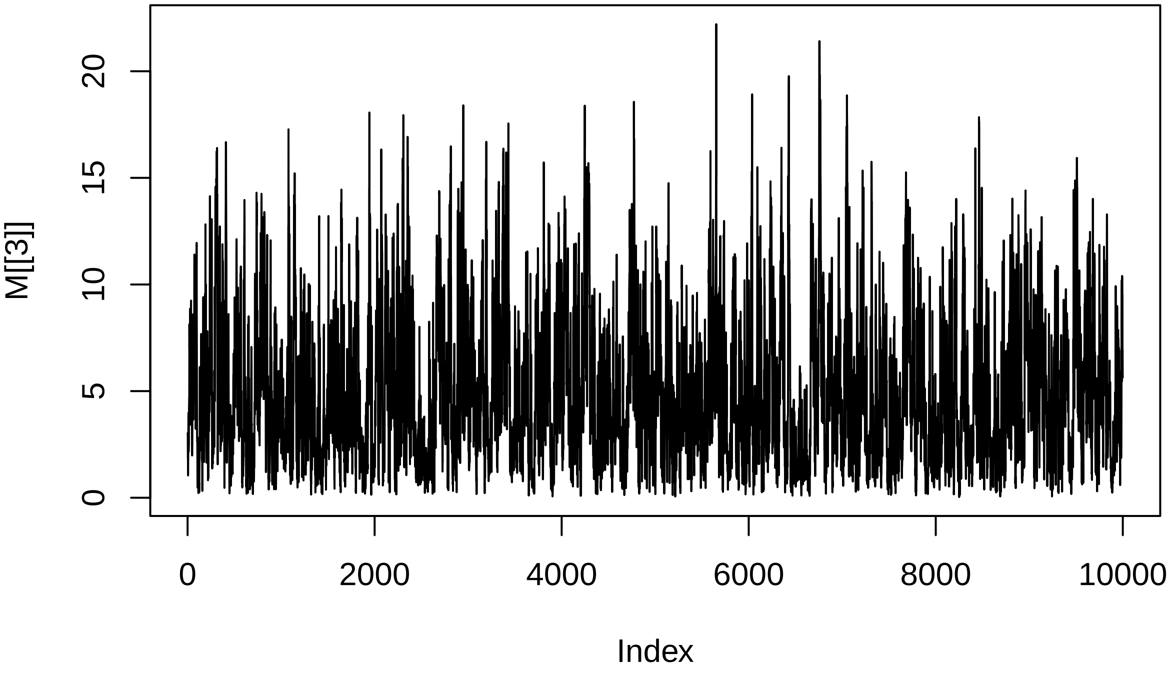
```
plot(M[[2]], type = "l")
```



```r
hist(M[[3]],breaks = 50)
```



```r
plot(M[[3]], type = "l")
```



```r
paste("Sigma = 2, Accept rate:", count[1]/(n-1))
```

```
## [1] "Sigma = 2, Accept rate: 0.713271327132713"
```

```r
paste("Sigma = 4, Accept rate:", count[2]/(n-1))
```

```
## [1] "Sigma = 4, Accept rate: 0.769076907690769"
```

```r
paste("Sigma = 6, Accept rate:", count[3]/(n-1))
```

```
## [1] "Sigma = 6, Accept rate: 0.79037903790379"
```