Using MCMC to estimate unknown parameters of normal mixture

Hukai Luo

26 October 2018

Abstract

In this assignment, we will use the Gibbs sampling approach to design an MCMC to estimate all the unknown parameters.Use the **arms()** function in package HI.

1 Normal mixture revisited

Consider again the normal mixture example, except that the parameters of the normal distributions are considered unknown. Suppose that prior for μ_1 and μ_2 are $N(0, 10^2)$, that the prior for $1/\sigma_1^2$ and $1/\sigma_2^2$ are $\Gamma(a, b)$ with shape $a = .5$ and scale \$ b = 10\$. Further, all the priors are independent. Design an MCMC using the Gibbs sampling approach to estimate all 5 parameters. Use the **arms()** function in package **HI**. Run your chain for sufficiently long and drop the burn-in period. Plot the histogram of the results for all the parameters.

2 Likelihood function

First, let's consider the normal mixture example, the normal mixture distribution is a mixture of two normal distribution function:

$$N(\mu_1, \sigma_1^2); N(\mu_2, \sigma_2^2)$$

and the weight for each distribution is δ and $1 - \delta$.

```r
mixture_distribution <- function(delta, x, mu1, mu2, sigma1, sigma2) {
    delta * dnorm(x, mu1, sigma1) + (1 - delta) * dnorm(x, mu2, sigma2)
}
```

However, we don't know these 5 parameters of the mixture distribution function. Further, all the priors are independent. Now, we will generate some random sample:$\delta = 0.7, \mu_1 = 7, \sigma_1^2 = 0.5^2, \mu_2 = 10, \sigma_2^2 = 0.5^2$

```r
delta <- 0.7 # true value to be estimated based on the data
n <- 100
set.seed(123)
u <- rbinom(n, prob = delta, size = 1)
x0 <- rnorm(n, ifelse(u == 1, 7, 10), 0.5)
```

By sampling $n = 100$ data from the mixture distribution, we will get the likelihood function:

$$likelihood(x; \delta, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2) = \prod_{i=1}^{n} \left[\delta N(\mu_1, \sigma_1^2) + (1 - \delta)N(\mu_2, \sigma_2^2)\right]$$

so does the loglikelihood function:

$$loglike(x; \delta, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2) = \sum_{i=1}^{n} log(\delta N(\mu_1, \sigma_1^2) + (1 - \delta)N(\mu_2, \sigma_2^2))$$

```
loglike <- function(delta, x , mu1, mu2, sigma1, sigma2){
  sum(log(delta * dnorm(x, mu1, sigma1) + (1 - delta) * dnorm(x, mu2, sigma2)))
}
```

3 Posterior density

then, let's calculate posterior density, according to Bayes theory

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

$p(\theta|x)$ is what we called posterior density, $p(x|\theta)$ is the likelihood function, $p(\theta)$ is prior probability, $p(x)$ is the normalization constant useful for Bayesian model selection. The prior for μ_1 and μ_2 are $N(0, 10^2)$, the prior for $\frac{1}{\sigma_1^2}$ and $\frac{1}{\sigma_2^2}$ are $\Gamma(0.5, 10)$, hence the prior for σ_1^2 and σ_2^2 are the inverse gamma distribution $IG(0.5, 10)$.

So we get the posterior density:

$$p(\delta, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2|x) = \prod_{i=1}^{n} \left[\delta N(\mu_1, \sigma_1^2) + (1-\delta)N(\mu_2, \sigma_2^2)\right] N(\mu_1, 0, 10)N(\mu_2, 0, 10)IG(\sigma_1^2, 0.5, 10)IG(\sigma_2^2, 0.5, 10)$$

so does the log posterior function:

$$logposterior(\delta, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2|x) = \sum_{i=1}^{n}(\delta N(\mu_1, \sigma_1^2)+(1-\delta)N(\mu_2, \sigma_2^2))+log(N(\mu_1, 0, 10))+log(N(\mu_2, 0, 10))+log(IG(\sigma_1^2, 0.5, 10)$$

```
library(invgamma)
logposterior <- function(delta, x, mu1, mu2, sigma1, sigma2){
  mu1.logprior <- dnorm(mu1, 0, 10, log = T)
  mu2.logprior <- dnorm(mu2, 0, 10, log = T)
  sigma1.logprior <- dinvgamma(sigma1^2, shape = 0.5, scale = 10, log = T)
  sigma2.logprior <- dinvgamma(sigma2^2, shape = 0.5, scale = 10, log = T)
  sum(mu1.logprior + mu2.logprior +sigma1.logprior + sigma2.logprior) + loglike(delta, x, mu1, mu2, sig
}
```

4 Gibbs sampling

Next we will use an MCMC based the Gibbs sampler uses the ARMS algorithm from R package HI as recommended in the question:

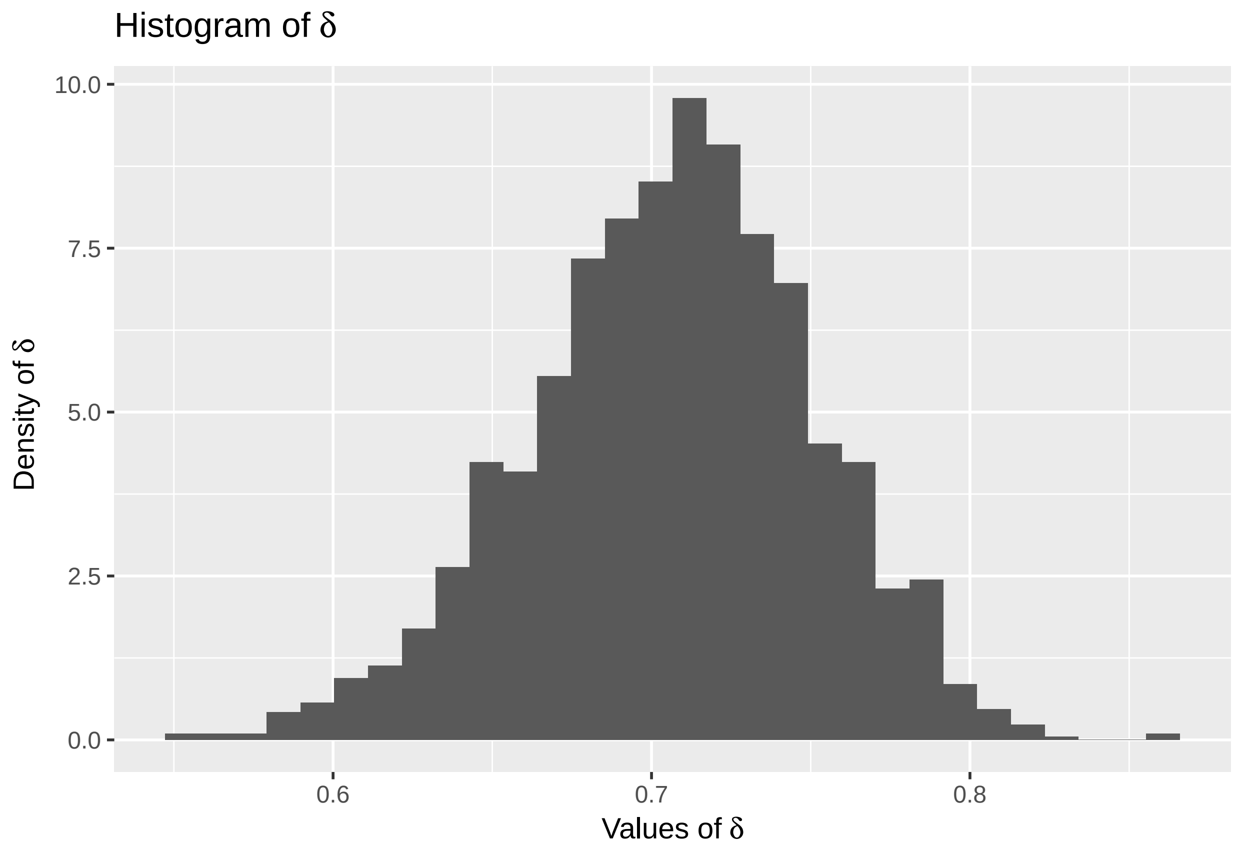
```
library(HI)
mymcmc <- function(niter, delta.init, mu1.init, mu2.init, sigma1.init, sigma2.init,x){
  data <- matrix(nrow = niter, ncol = 5)
  for(i in 1:niter){
    f1 <- function(x1)  logposterior(x1,x0,mu1.init,mu2.init,sigma1.init,sigma2.init)
    delta.init <- data[i,1] <- arms(delta.init, f1, function(x1) (x1 > 0) * (x1< 1), 1)

    f2 <- function(x2)  logposterior(delta.init,x0,x2,mu2.init,sigma1.init,sigma2.init)
    mu1.init <- data[i,2] <- arms(mu1.init, f2, function(x2) (x2 > -100) * (x2 < 100), 1)
    f3 <- function(x3)  logposterior(delta.init,x0,mu1.init,x3,sigma1.init,sigma2.init)
    mu2.init <- data[i,3] <- arms(mu2.init, f3, function(x3) (x3 > -100) * (x3 < 100), 1)

    f4 <- function(x4)  logposterior(delta.init,x0,mu1.init,mu2.init,x4,sigma2.init)
    sigma1.init <- data[i,4] <- arms(sigma1.init, f4, function(x4) (x4 > 0) * (x4 < 200), 1)

    f5 <- function(x5)  logposterior(delta.init,x0,mu1.init,mu2.init,sigma1.init,x5)
    sigma2.init <- data[i,5] <- arms(sigma2.init, f5, function(x5) (x5 > 0) * (x5 < 200), 1)
  }
  data
}
niter <- 3000
results <- mymcmc(niter, 0.5, 1,1,1,1,x0)
results <- results[-c(1:1000),]
options(warn=-1)
library(ggplot2)
plot1<- ggplot(data.frame(x = results[,1]), aes(x = x))+
  geom_histogram(aes(y=..density..))+labs(x = expression("Values of"~delta),
  y = expression(" Density of"~delta), title=expression("Histogram of"~delta))
plot1
```

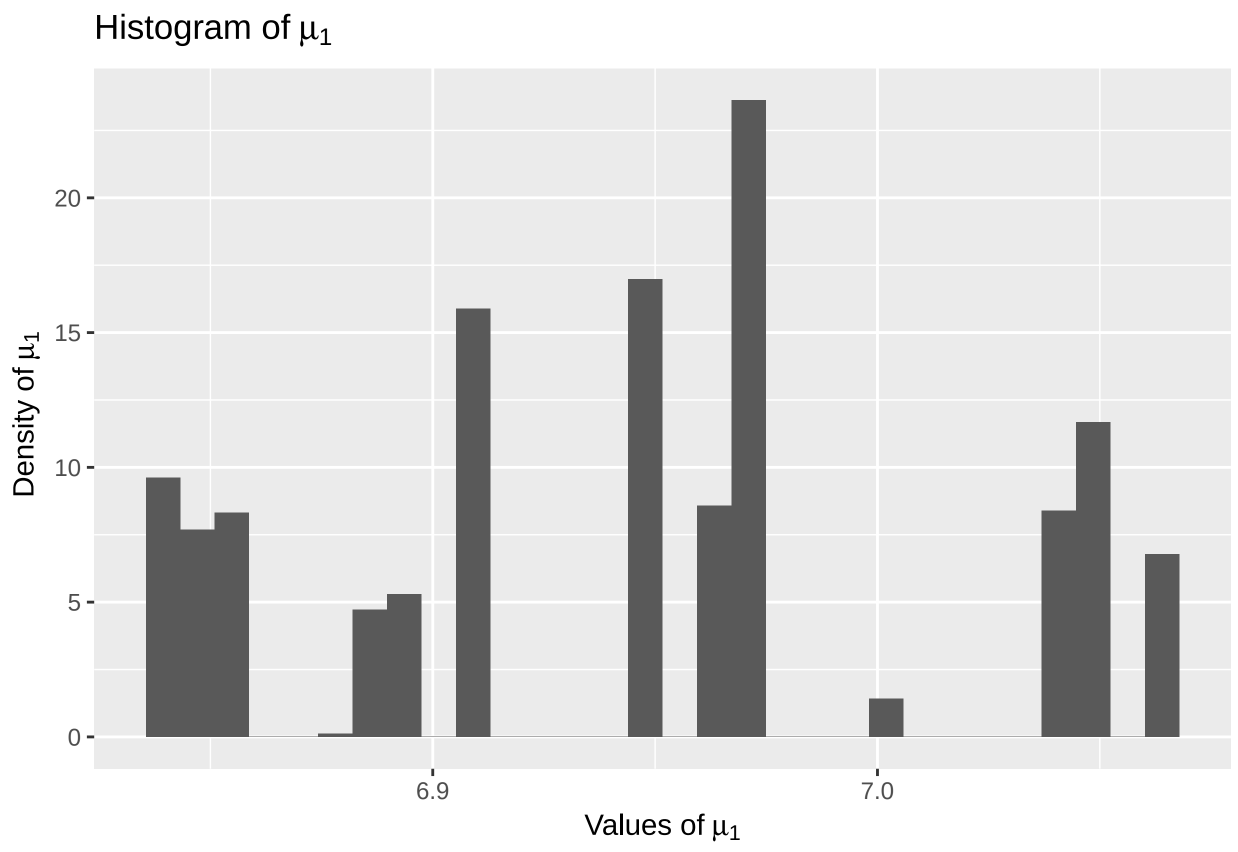
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



δ is about 0.7

```r
plot2<- ggplot(data.frame(x = results[,2]), aes(x = x))+
  geom_histogram(aes(y=..density..))+labs(x = expression("Values of"~mu[1]),
  y = expression(" Density of"~mu[1]), title=expression("Histogram of"~mu[1]))
plot2
```

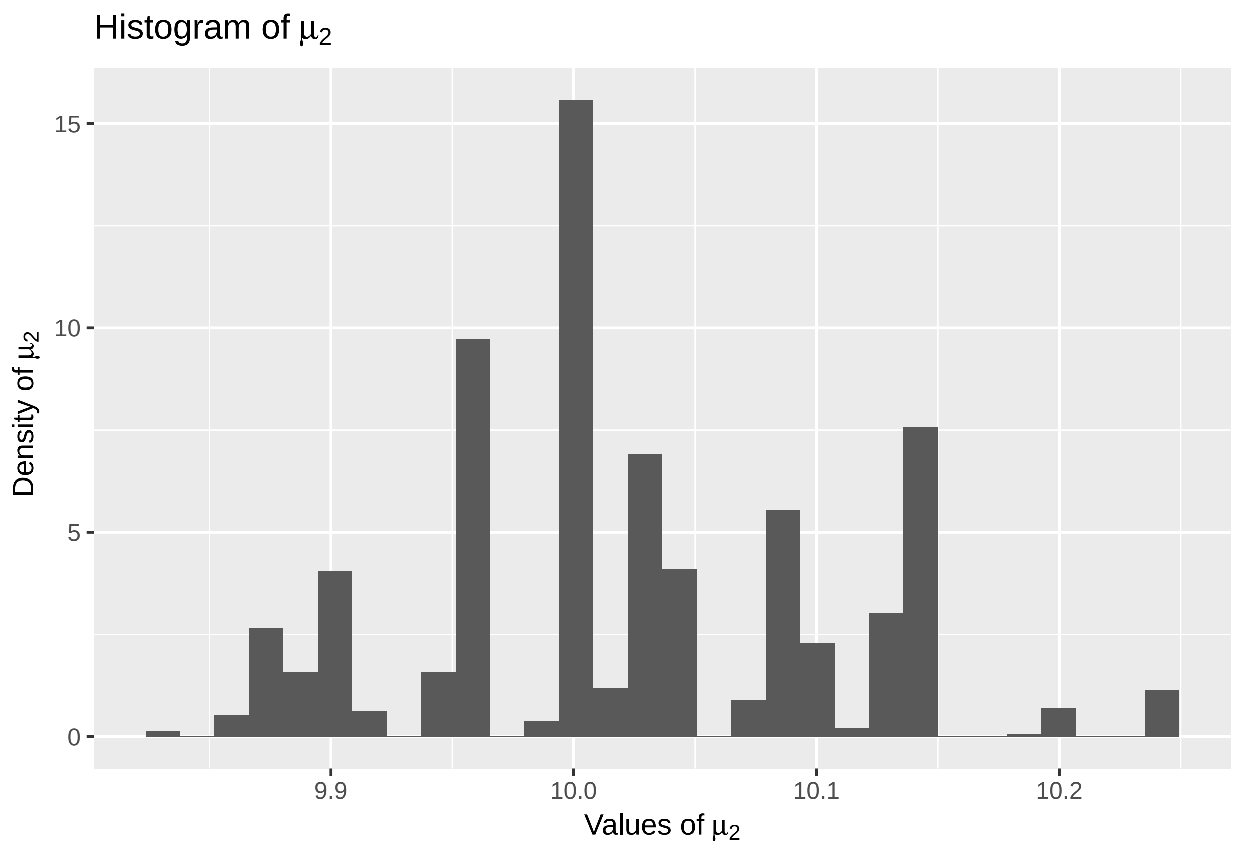
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



μ_1 is about 7

```
plot3<- ggplot(data.frame(x = results[,3]), aes(x = x))+
  geom_histogram(aes(y=..density..))+labs(x = expression("Values of"~mu[2]),
  y = expression(" Density of"~mu[2]), title=expression("Histogram of"~mu[2]))
plot3
```

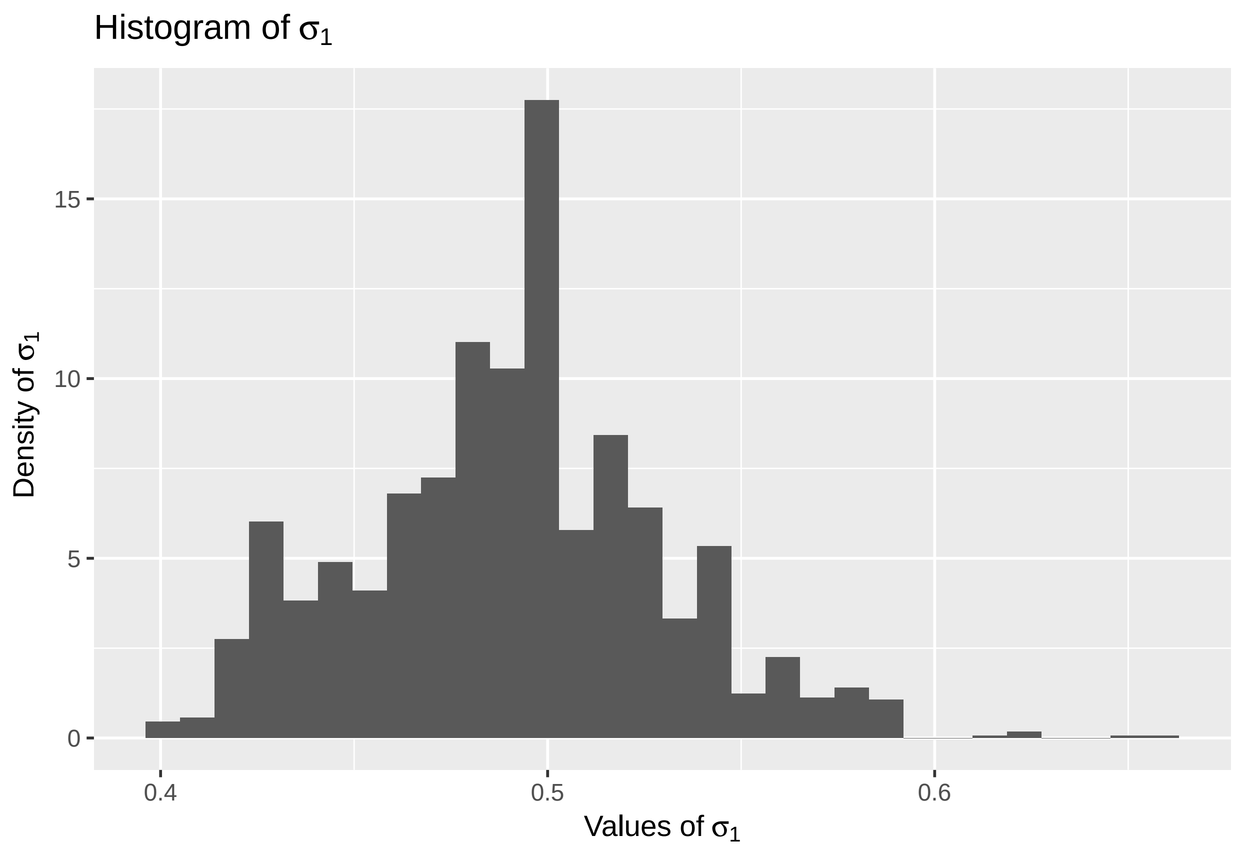
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



μ_2 is about 10

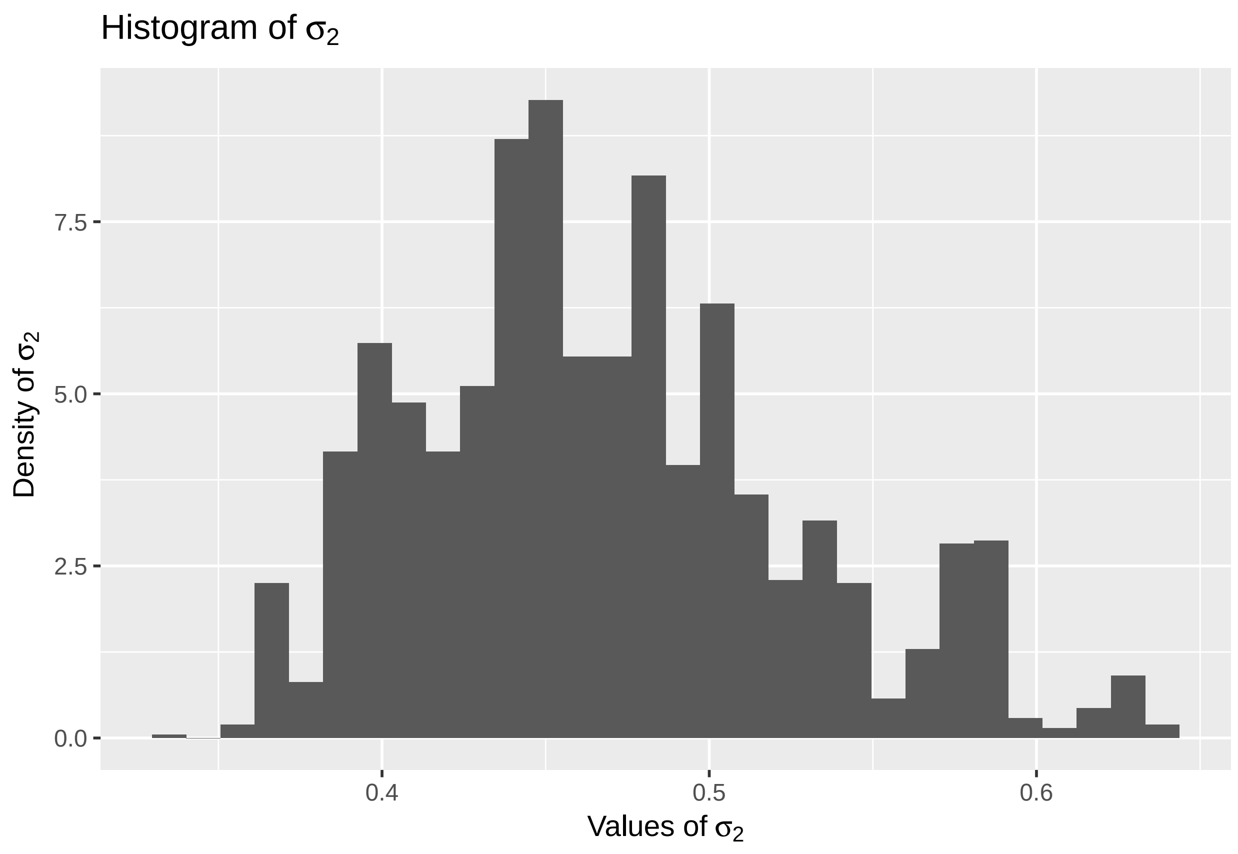
```r
plot4<- ggplot(data.frame(x = results[,4]), aes(x = x))+
  geom_histogram(aes(y=..density..))+labs(x = expression("Values of"~sigma[1]),
  y = expression(" Density of"~sigma[1]), title=expression("Histogram of"~sigma[1]))
plot4
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
plot5<- ggplot(data.frame(x = results[,5]), aes(x = x))+
  geom_histogram(aes(y=..density..))+labs(x = expression("Values of"~sigma[2]),
  y = expression(" Density of"~sigma[2]), title=expression("Histogram of"~sigma[2]))
plot5
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Histogram of σ_2
Density of σ_2
Values of σ_2