

Lab 2

Björn Edblom (bjoed735) och Eskil Brännerud (eskbr129)

2023-10-11

Uppgift 3.1.1

```
set.seed(4711)
x1 <- rgamma(n=10, shape = 4, scale = 1)
x2 <- rgamma(n=100, shape = 4, scale = 1)
```

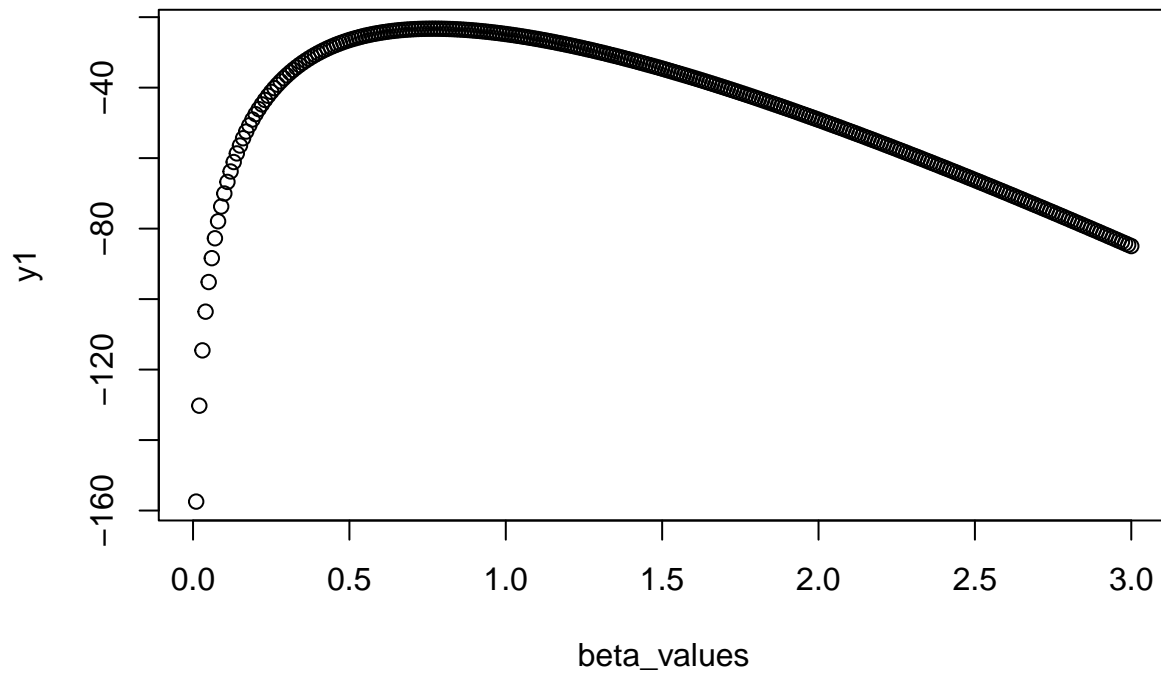
(1).

```
llgama <- function(x, alpha, beta){
  return((alpha - 1) * sum(log(x)) - (beta*sum(x)) + length(x) * (alpha * log(beta) - lgamma(alpha)))
}
print(llgama(x1,2,2))
```

```
## [1] -75.18981
```

(2).

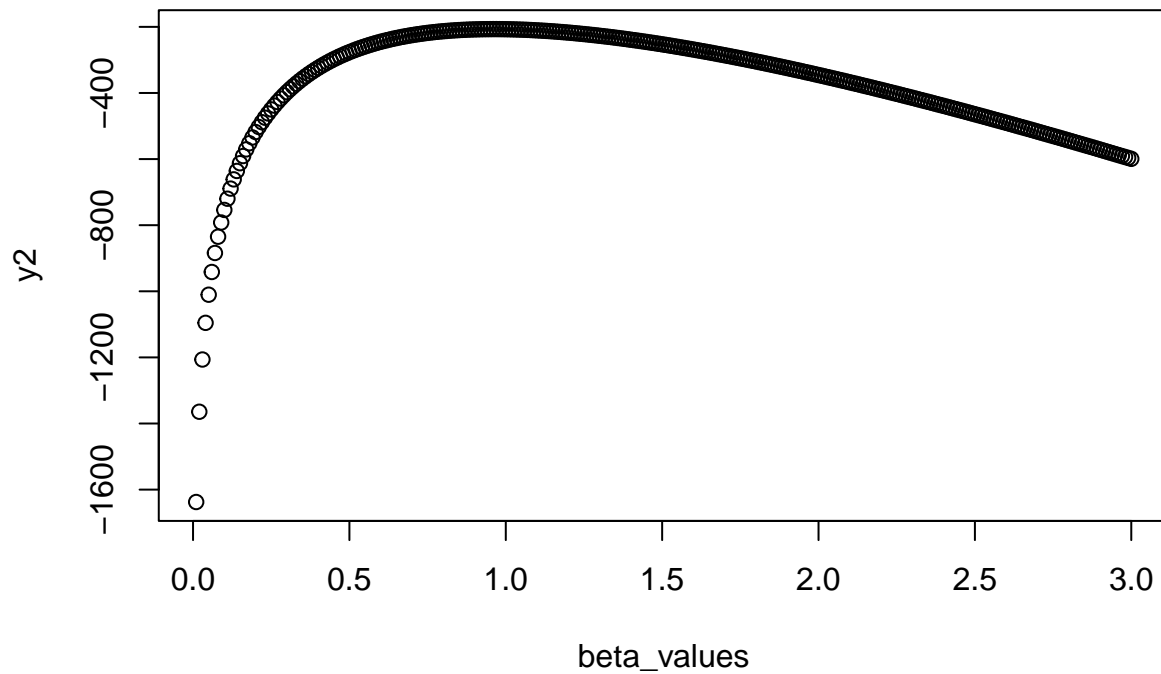
```
alpha <- 4
beta_values <- seq(0.01, 3, 0.01)
#create an array
y1 <- numeric(0)
for(beta in beta_values){
  y1 <- c(y1, llgama(x1, alpha, beta))
}
y2 <- numeric(0)
for(beta in beta_values){
  y2 <- c(y2, llgama(x2, alpha, beta))
}
plot(beta_values, y1)
```



```
b_max1 <- beta_values[which.max(y1)]
print(paste("Max värde =", b_max1))
```

```
## [1] "Max värde = 0.77"
```

```
plot(beta_values, y2)
```

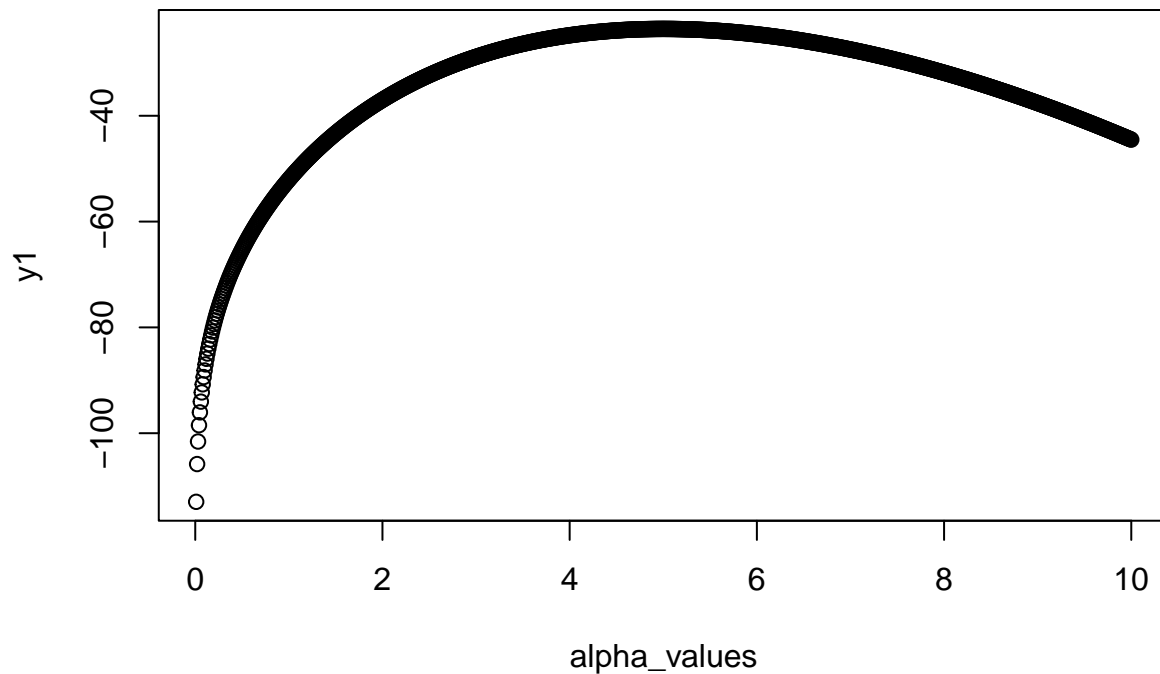


```
b_max2 <- beta_values[which.max(y2)]
print(paste("Max värde =", b_max2))
```

```
## [1] "Max värde = 0.96"
```

(3).

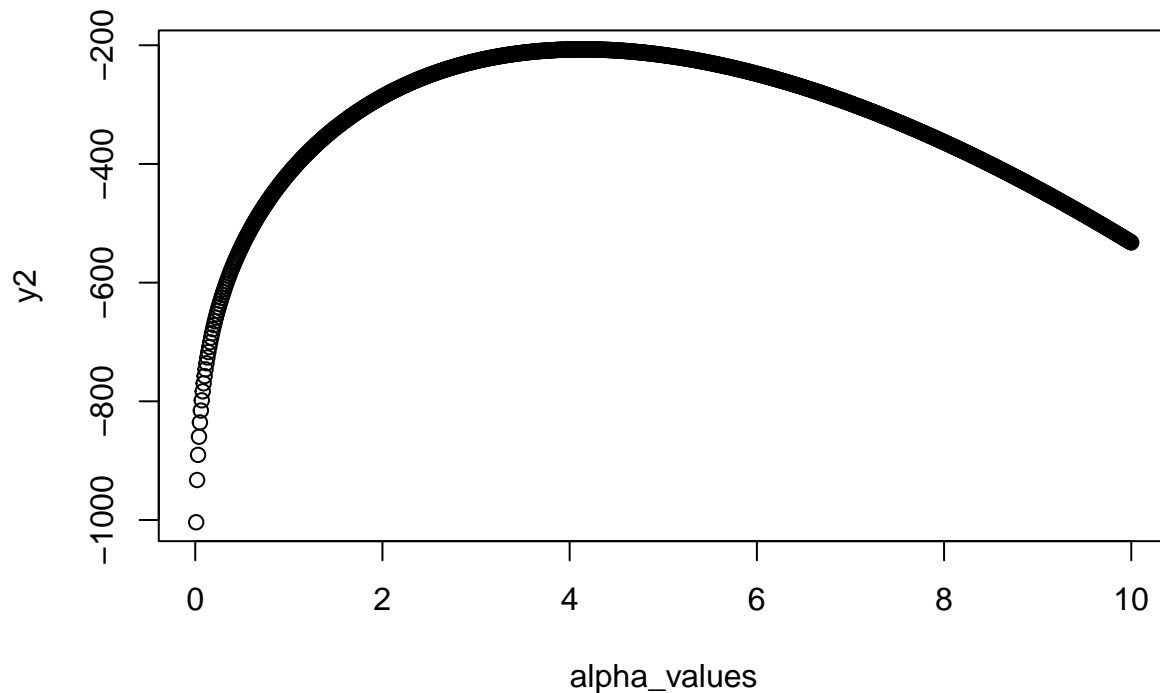
```
alpha_values <- seq(0.01, 10, 0.01)
beta <- 1
#create an array
y1 <- numeric(0)
for(alpha in alpha_values){
  y1 <- c(y1, llgama(x1, alpha, beta))
}
y2 <- numeric(0)
for(alpha in alpha_values){
  y2 <- c(y2, llgama(x2, alpha, beta))
}
plot(alpha_values, y1)
```



```
a_max1 <- alpha_values[which.max(y1)]
print(paste("Max värde =", a_max1))
```

```
## [1] "Max värde = 5"
```

```
plot(alpha_values, y2)
```



```
a_max2 <- alpha_values[which.max(y2)]
print(paste("Max värde =", a_max2))
```

```
## [1] "Max värde = 4.13"
```

(4).

```
llnorm <- function(x,mu,sigma2){
  n <- length(x)
  return((-n/2)*log(2*pi*sigma2)+(-1/(2*sigma2)*sum((x-mu)^2)))
}
print(llnorm(x=x1,mu=2,sigma2=1))
```

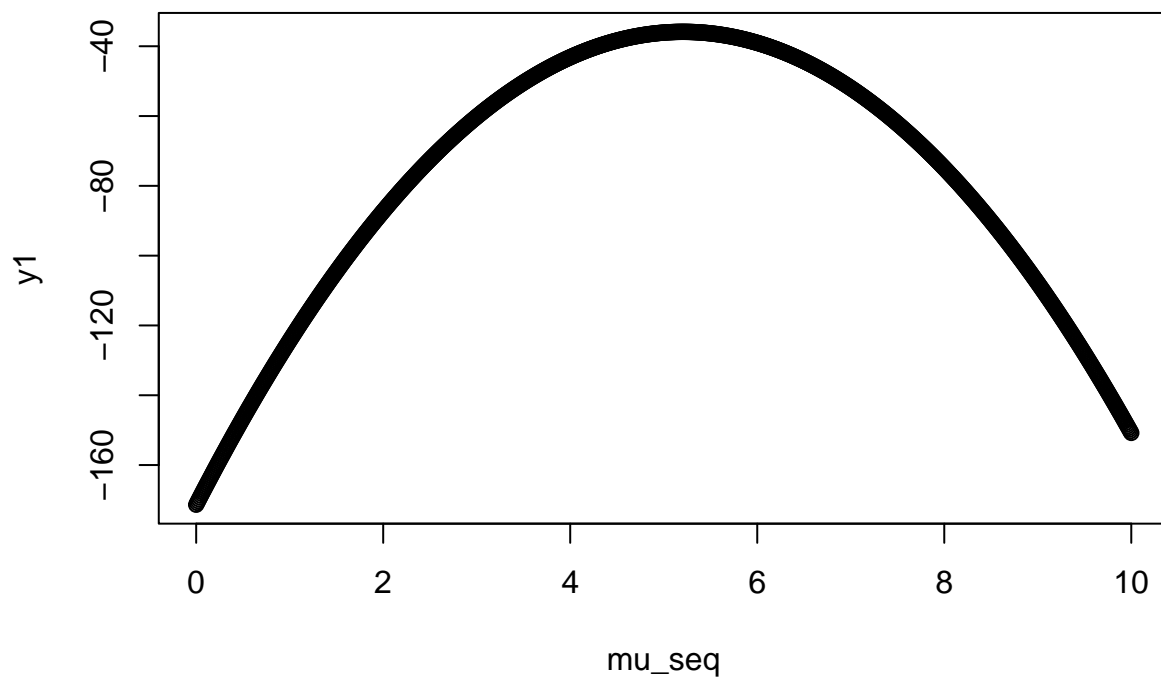
```
## [1] -87.25743
```

(5).

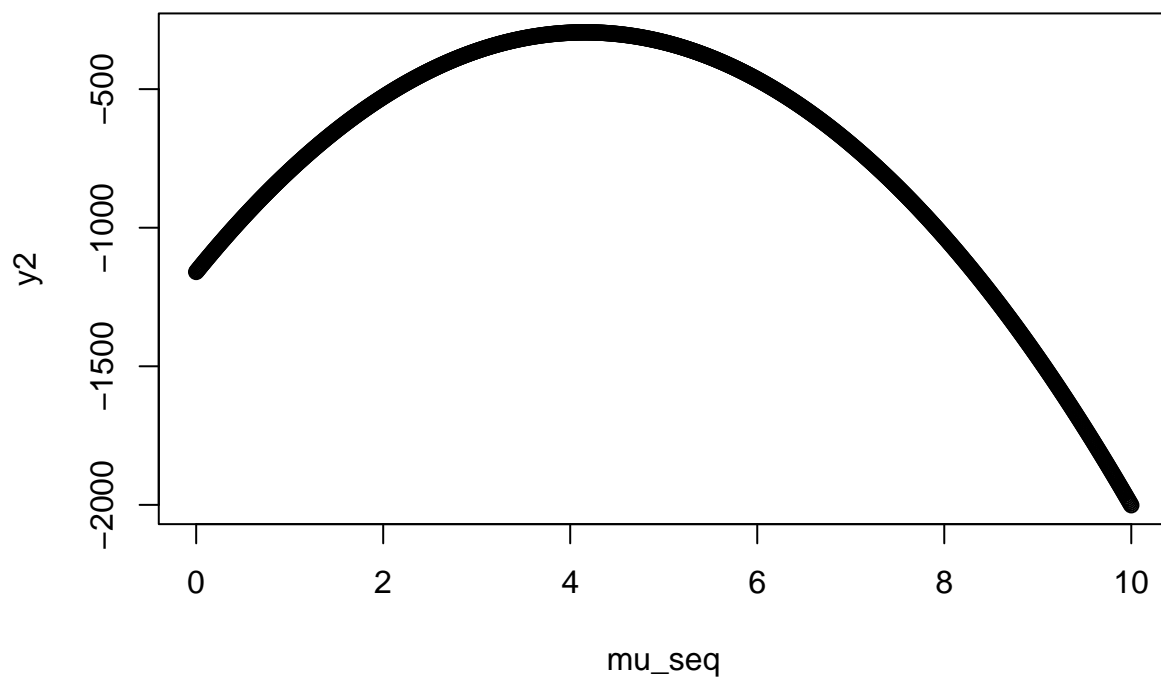
```
sigma2 <- 1
mu_seq <- seq(0, 10, 0.01)
y1 <- numeric(0)
for ( mu in mu_seq) {
  y1 <- c(y1, llnorm(x1, mu, sigma2))
}

y2 <- numeric(0)
for ( mu in mu_seq) {
  y2 <- c(y2, llnorm(x2, mu, sigma2))
}

plot(mu_seq,y1)
```



```
plot(mu_seq,y2)
```



```
#####
```

Genom att kolla på grafen ser första ut som en standardfördelning och den andra ser ut som en gammafördelning

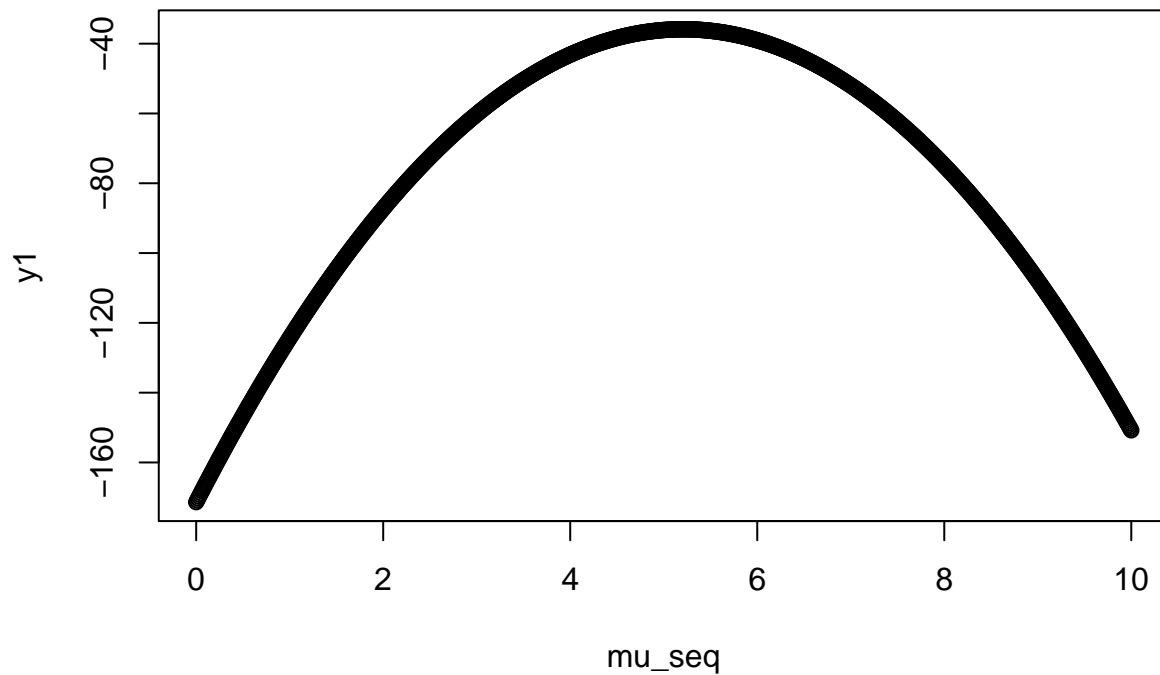
```
sigma2 <- 1
mu_seq <- seq(0, 10, 0.01)
y1 <- numeric(0)
for (mu in mu_seq) {
  y1 <- c(y1, llnorm(x = x1, mu = mu, sigma2 = sigma2))
}
```

```

}
y2 <- numeric(0)
for (mu in mu_seq) {
  y2 <- c(y2, llnorm(x = x2, mu = mu, sigma2 = sigma2))
}

```

```
plot(mu_seq, y1)
```



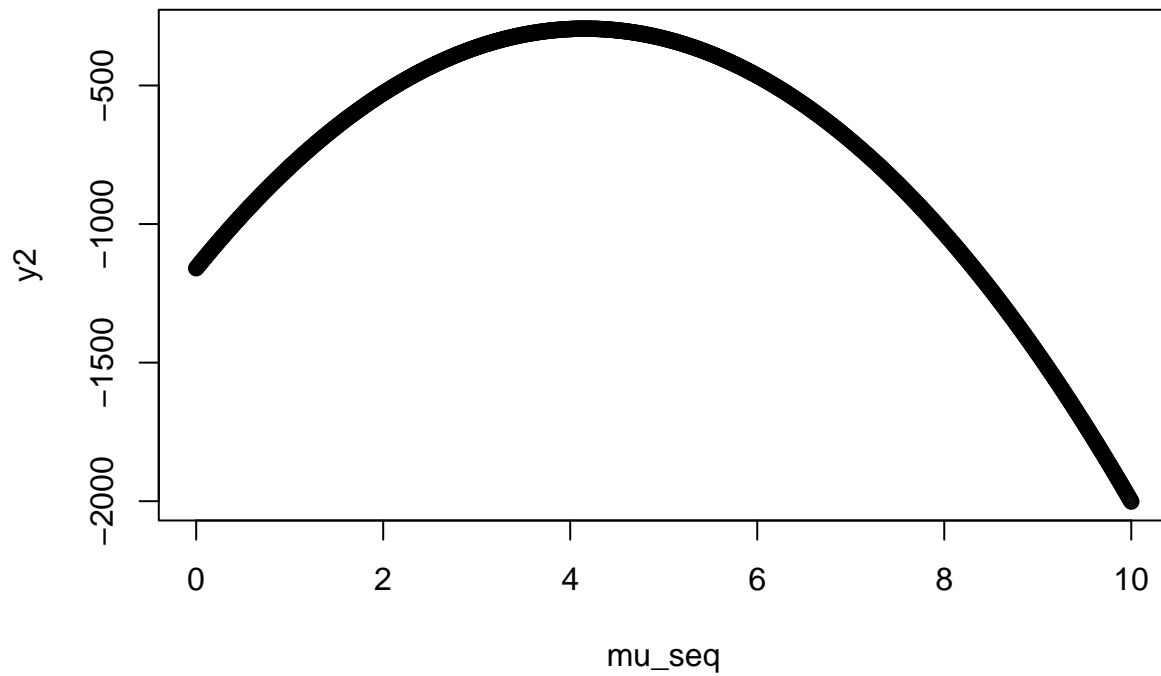
```

m_max1 <- mu_seq[which.max(y1)]
print(paste("Max värde =", m_max1))

```

```
## [1] "Max värde = 5.21"
```

```
plot(mu_seq, y2)
```

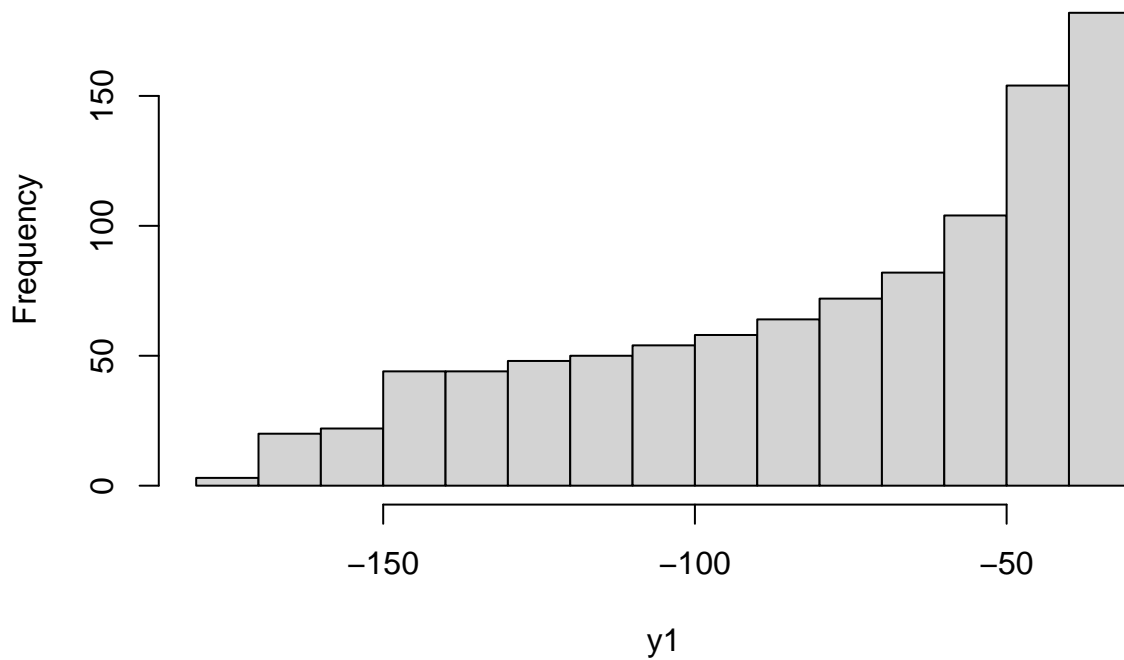


```
m_max2 <- mu_seq[which.max(y2)]
print(paste("Max värde =", m_max2))
```

```
## [1] "Max värde = 4.16"
```

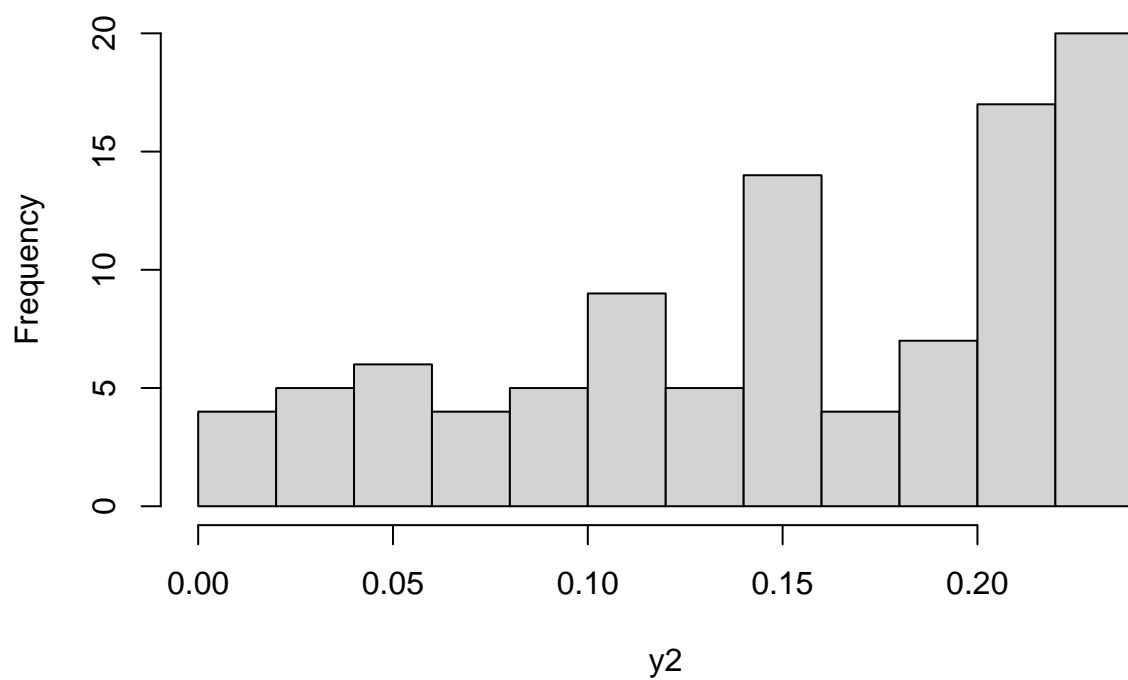
```
hist(y1)
```

Histogram of y1



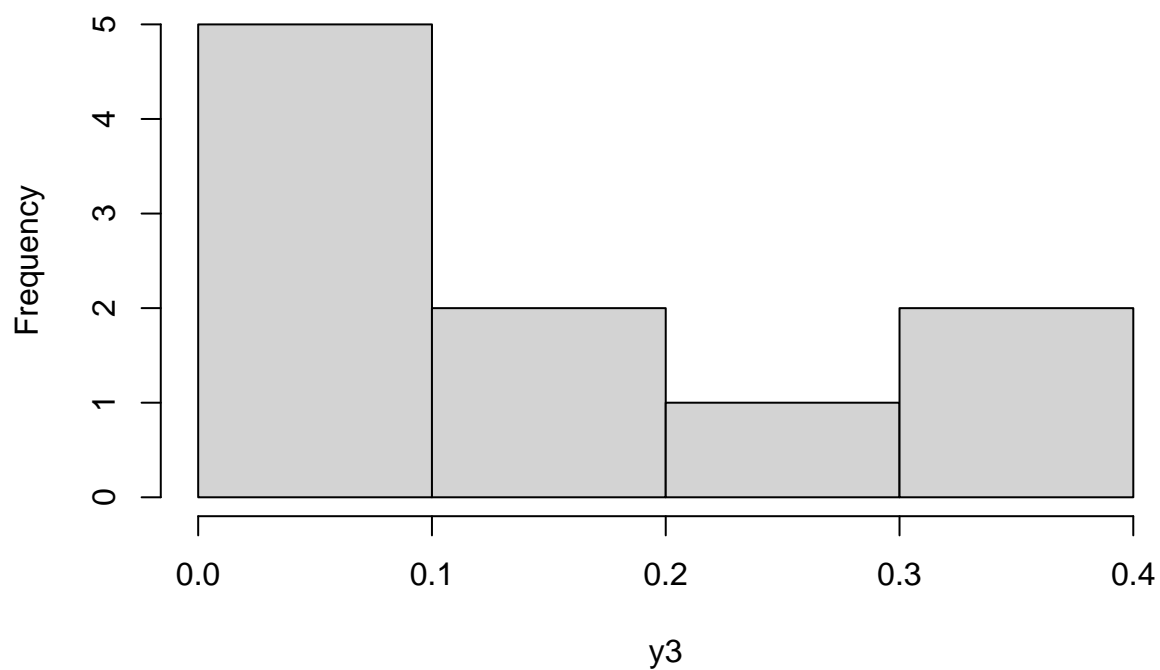
```
y2 <- dgamma(x2, shape = a_max2, scale = b_max2)
hist(y2)
```

Histogram of y2



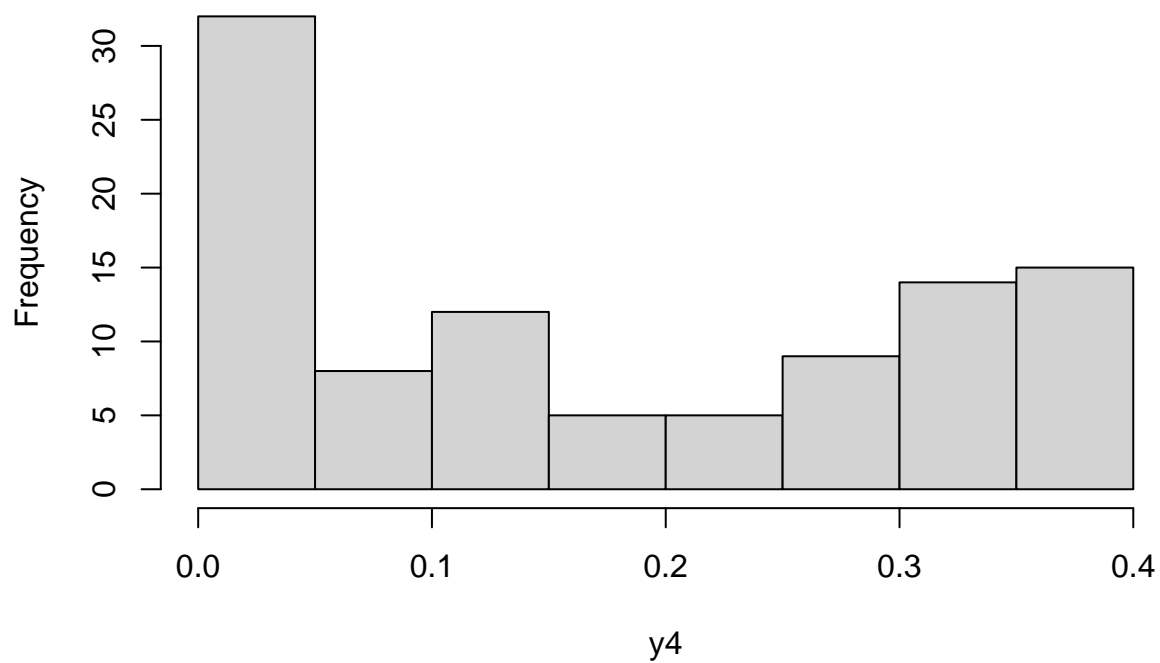
```
y3 <- dnorm(x1, mean = m_max1, sd = 1)
hist(y3)
```

Histogram of y3



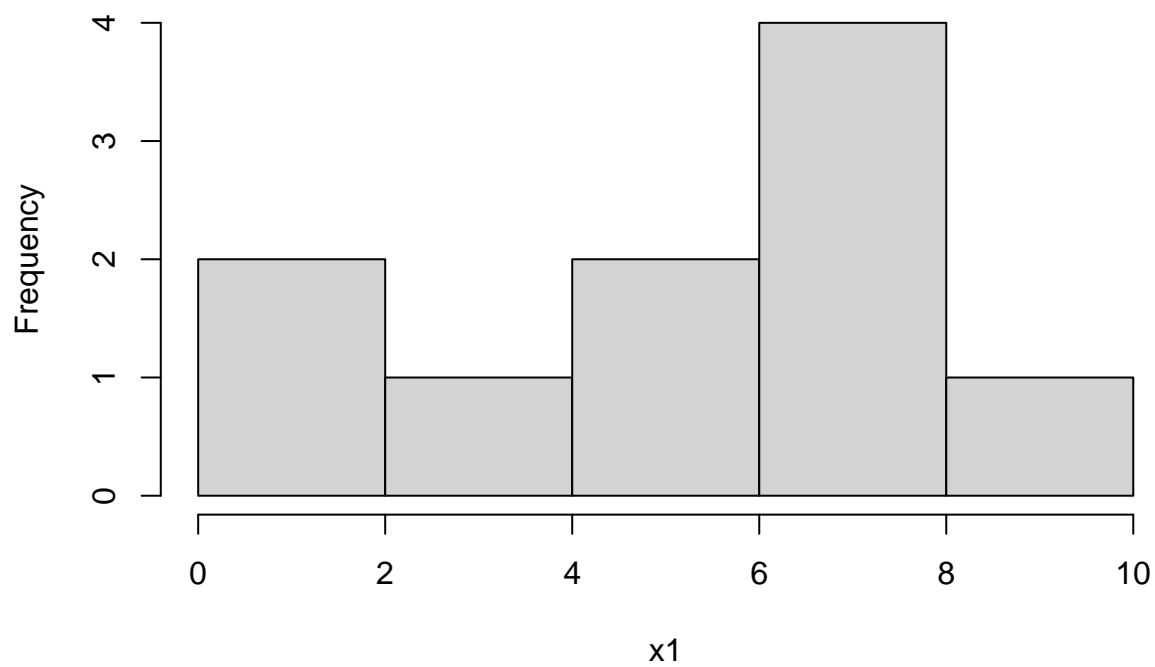
```
y4 <- dnorm(x2, mean = m_max2, sd = 1)
hist(y4)
```


Histogram of y4

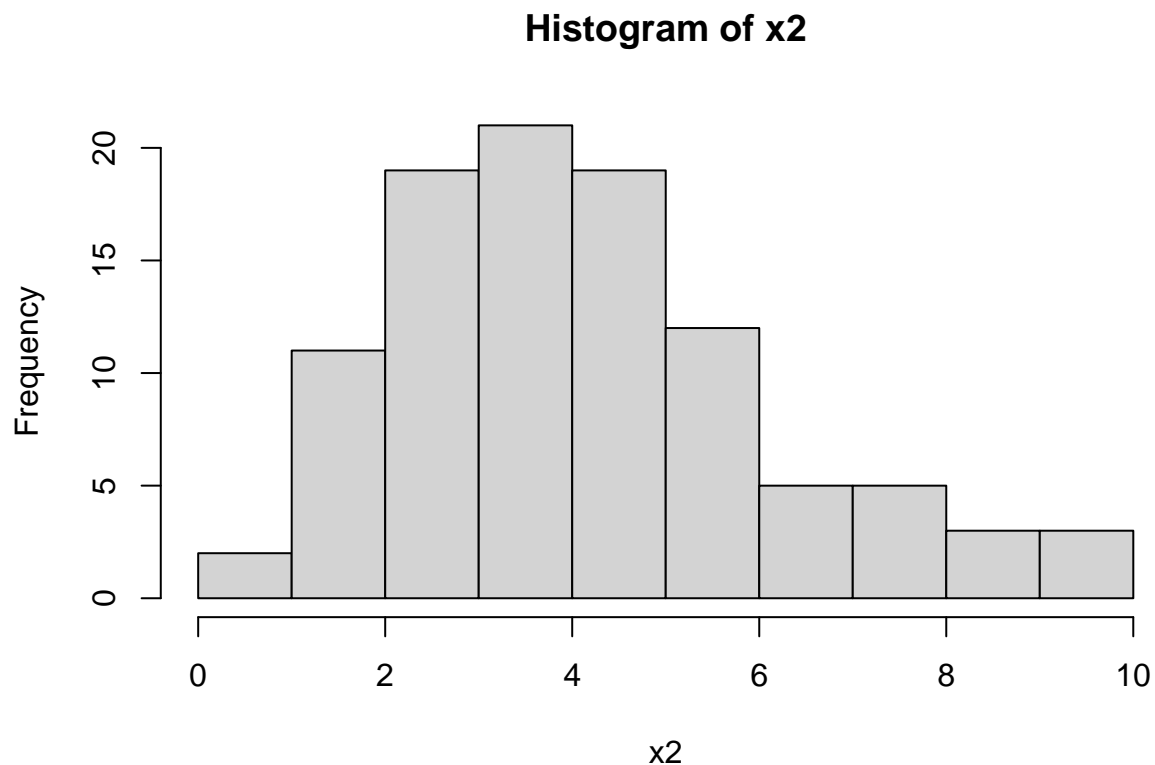


```
hist(x1)
```

Histogram of x1



```
hist(x2)
```



3.2.1

```
gamma_beta_mle <- function(x,alpha){
  n<-length(x)
  a<-alpha
  coolSum <- n*a*(sum(unlist(x)))^-1;

  return(coolSum)
}
print(gamma_beta_mle(x1,2))
```

```
## [1] 0.3841892
```

För x1 så maximeras chansen för att få dessa värden då beta är 0.768 För x2 så maximeras chansen för att få dessa värden då beta är 0.961

```
#1
test_x <- 1:10
norm_mu_mle <- function(x){
  n<-length(x)
  coolSum <- (1/n)*sum(unlist(x));
  return(coolSum)
}

norm_sigma2_mle <- function(x){
  xMean <- mean(unlist(x))
  coolSum = 0
  for(i in x){
```

```

    coolSum = coolSum + ((i-xMean)^2)
  }
  return((1/length(x))*coolSum)

}

print(norm_mu_mle(test_x))

## [1] 5.5
print(norm_sigma2_mle(test_x))

## [1] 8.25
#2
set.seed(42)
y10 <- rnorm(10,10,2)
y10000 <- rnorm(10000,10,2)

print("Uppgift 2")

## [1] "Uppgift 2"
print(norm_mu_mle(y10))

## [1] 11.09459
print(norm_sigma2_mle(y10))

## [1] 2.512709
print(norm_mu_mle(y10000))

## [1] 9.9762
print(norm_sigma2_mle(y10000))

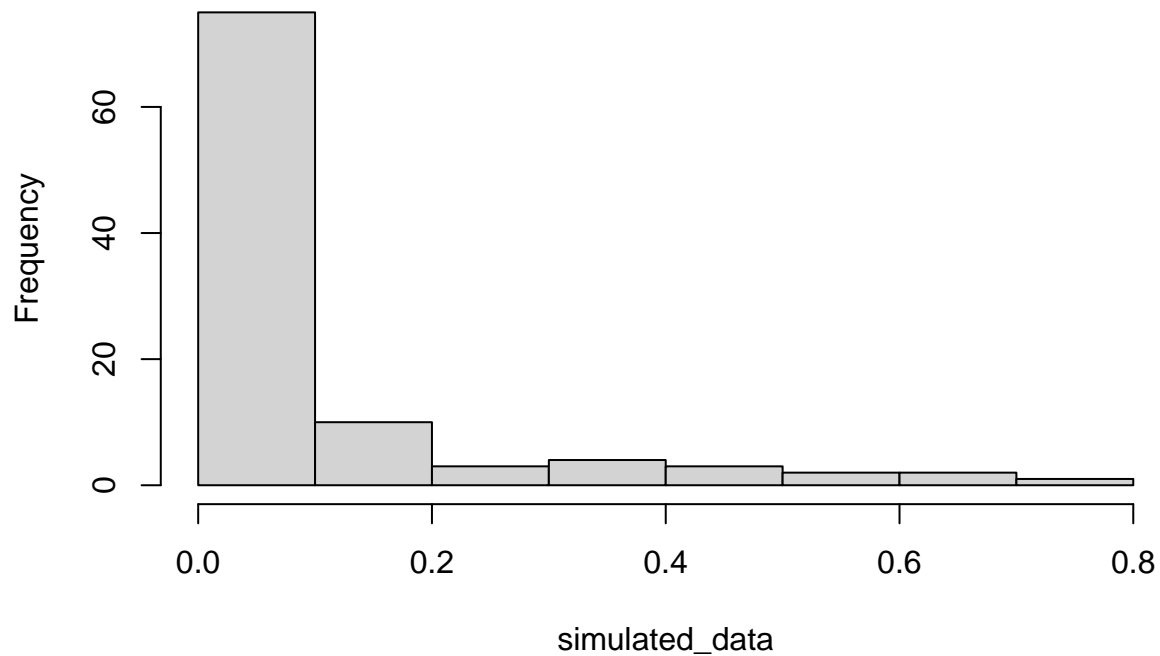
## [1] 4.048198
#Visar att fler dragningar i norm_mu_mle(x) gör att värdet närmar sig medelvärde/variansen

##3.3.1 ###1
log_likelihood_beta <- function(params, data) {
  alpha <- params[1]
  beta <- params[2]
  log_likelihood <- sum(dbeta(data, alpha, beta, log = TRUE))
  return(-log_likelihood) # Returnera negativt värde eftersom optim() minimiserar
}

###2
simulated_data <- rbeta(100, shape1 = 0.2, shape2 = 2)
hist(simulated_data)

```

Histogram of simulated_data



##3.4.1 ###1

```
n <- 2000
beta1 <- numeric(n)
beta2 <- numeric(n)
m1 <- numeric(n)
m2 <- numeric(n)
sigma1 <- numeric(n)
sigma2 <- numeric(n)

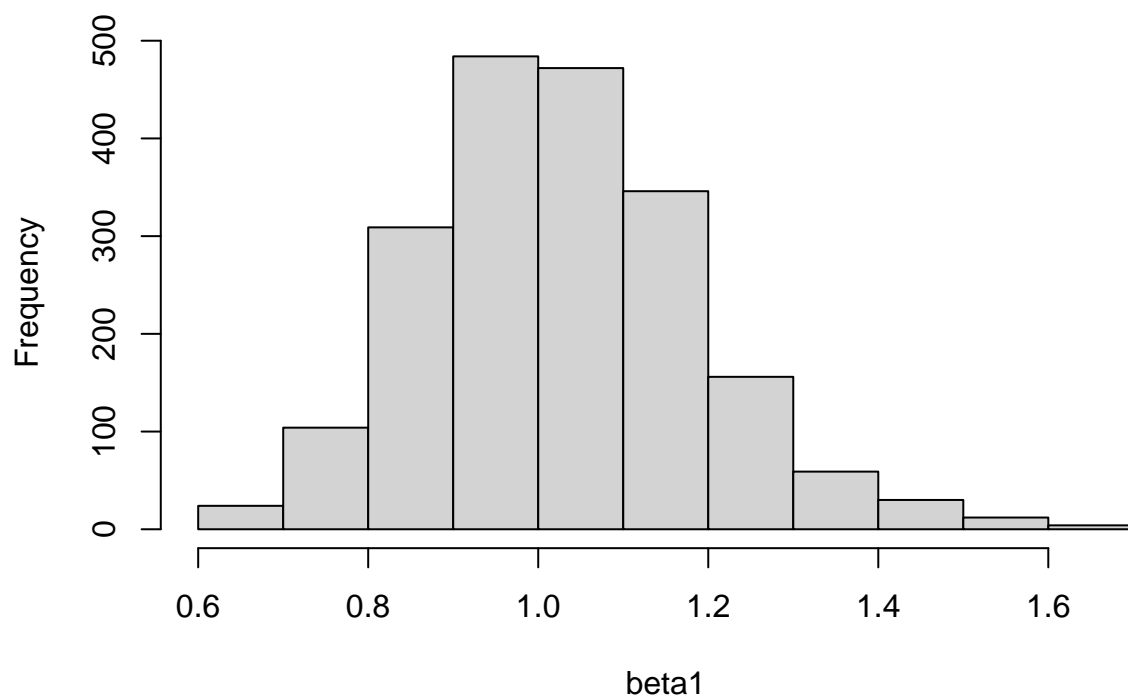
for (i in 1:n) {

  y1 <- rnorm(n = 10, mean = 10, sd = 2)
  y2 <- rnorm(n = 10000, mean = 10, sd = 2)
  m1[i] <- norm_mu_mle(x = y1)
  m2[i] <- norm_mu_mle(x = y2)
  sigma1[i] <- norm_sigma2_mle(x = y1)
  sigma2[i] <- norm_sigma2_mle(x = y2)
  x1 <- rgamma(n = 10, shape = 4, rate = 1)
  x2 <- rgamma(n = 10000, shape = 4, rate = 1)
  beta1[i] <- gamma_beta_mle(x = x1, alpha = 4)
  beta2[i] <- gamma_beta_mle(x = x2, alpha = 4)

}

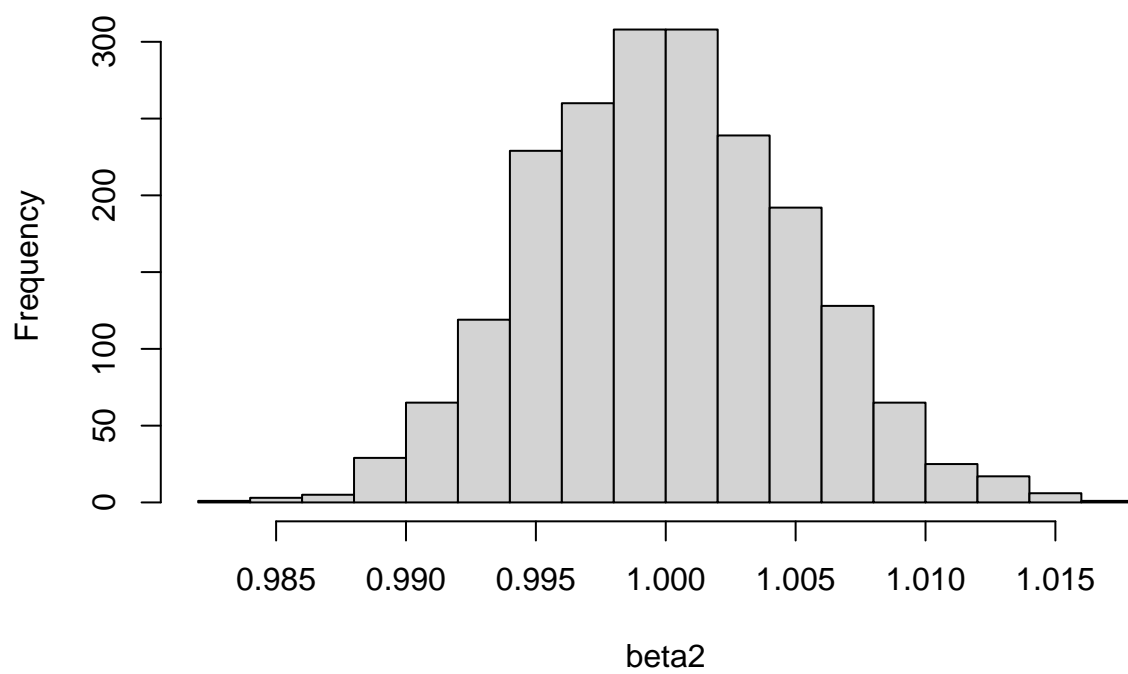
hist(beta1)
```

Histogram of beta1



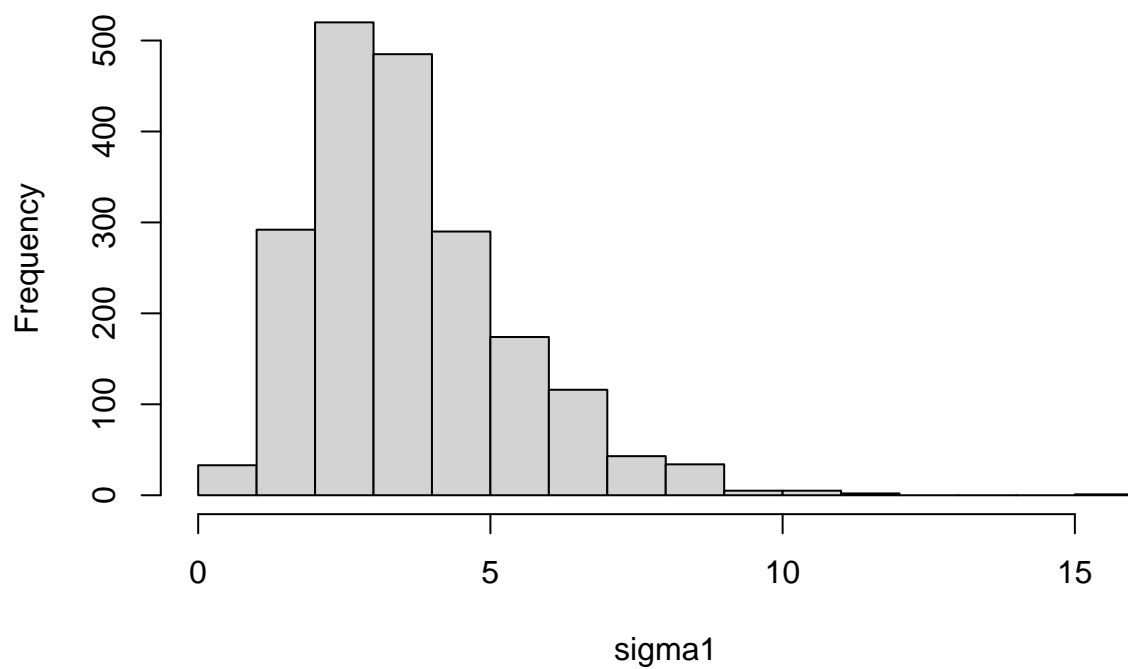
```
hist(beta2)
```

Histogram of beta2



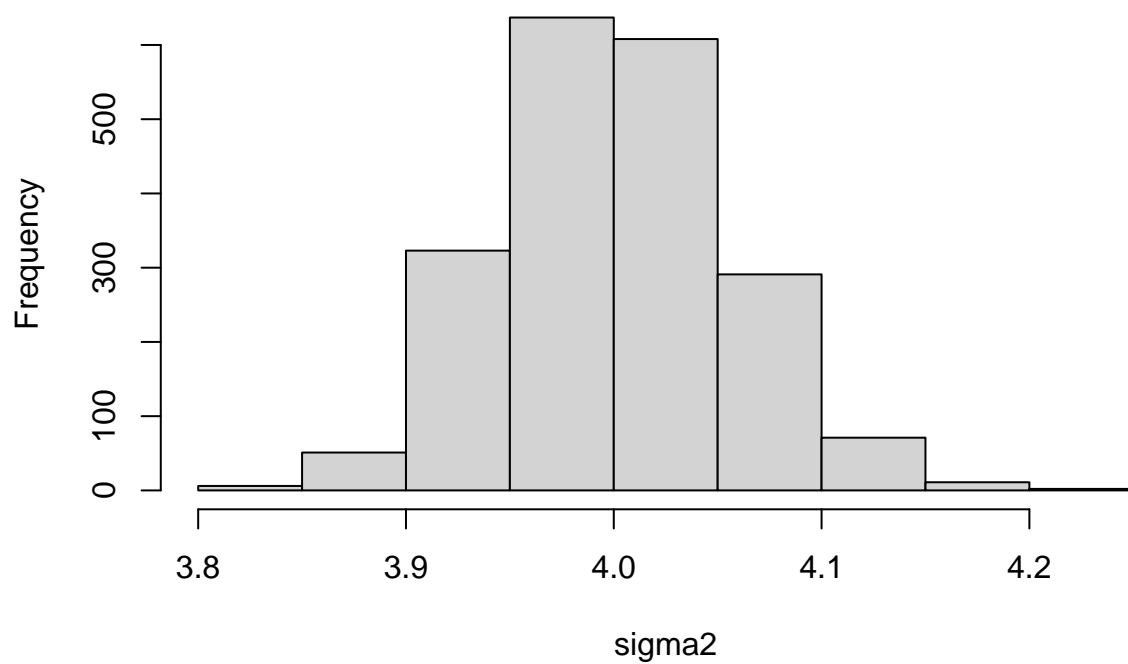
```
hist(sigma1)
```

Histogram of sigma1



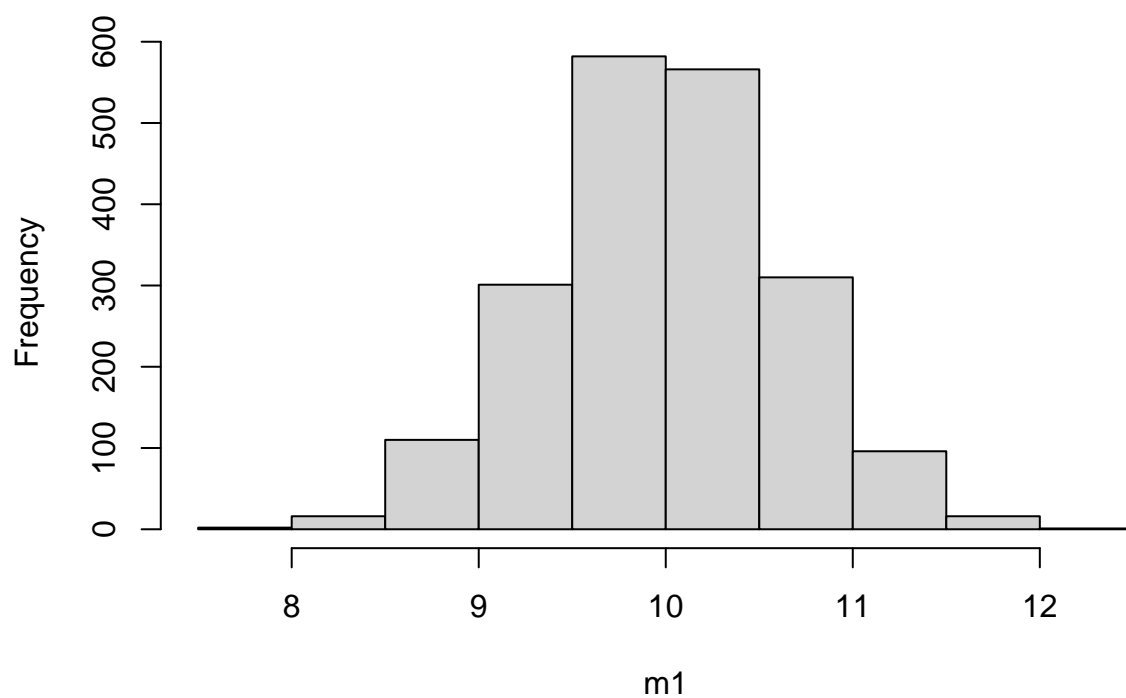
```
hist(sigma2)
```

Histogram of sigma2



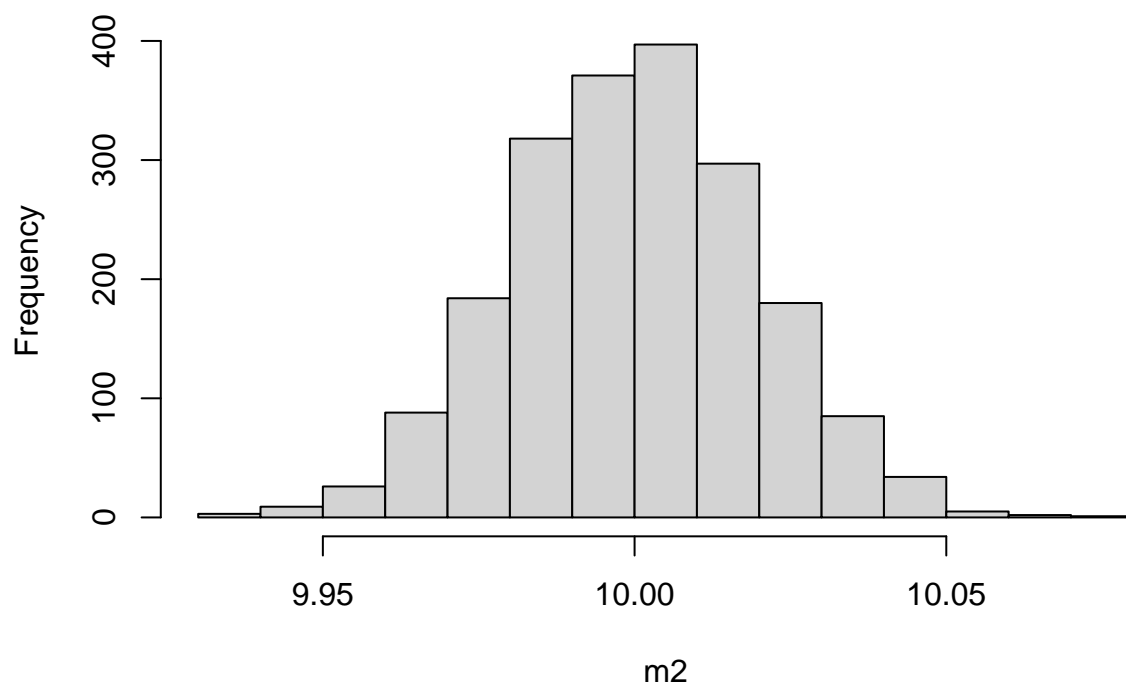
```
hist(m1)
```

Histogram of m1



```
hist(m2)
```

Histogram of m2



slutsats är att när vi höjer värdet på n så minskar variansen. ###2

Vår

```
n <- 2000
```

```
m1 <- numeric(n)
```

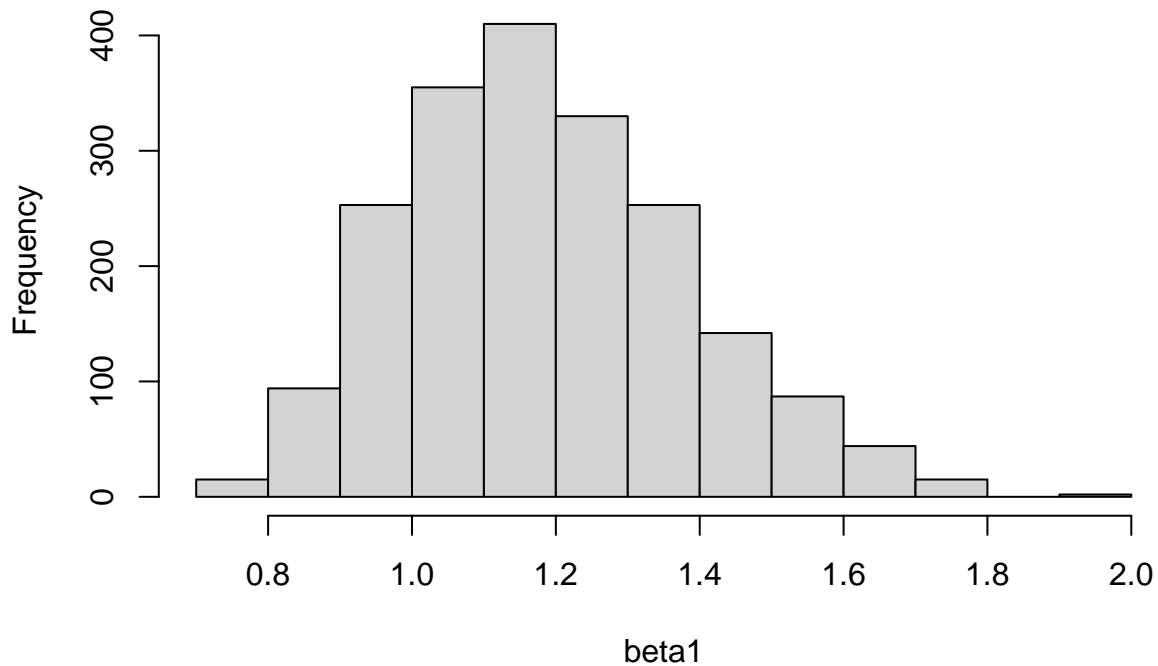
```

m2 <- numeric(n)
beta1 <- numeric(n)
beta2 <- numeric(n)
sigma1 <- numeric(n)
sigma2 <- numeric(n)
x1 <- rgamma(n = 10, shape = 4, rate = 1)
x2 <- rgamma(n = 10000, shape = 4, rate = 1)
y1 <- rnorm(n = 10, mean = 10, sd = 2)
y2 <- rnorm(n = 10000, mean = 10, sd = 2)

for (i in 1:n) {
  sigma1[i] <- norm_sigma2_mle(x = sample(y1, 10, replace = TRUE))
  sigma2[i] <- norm_sigma2_mle(x = sample(y2, 10000, replace = TRUE))
  beta1[i] <- gamma_beta_mle(x = sample(x1, 10, replace = TRUE), alpha = 4)
  beta2[i] <- gamma_beta_mle(x = sample(x2, 10000, replace = TRUE), alpha = 4)
  m1[i] <- norm_mu_mle(x = sample(y1, 10, replace = TRUE))
  m2[i] <- norm_mu_mle(x = sample(y2, 10000, replace = TRUE))
}
hist(beta1)

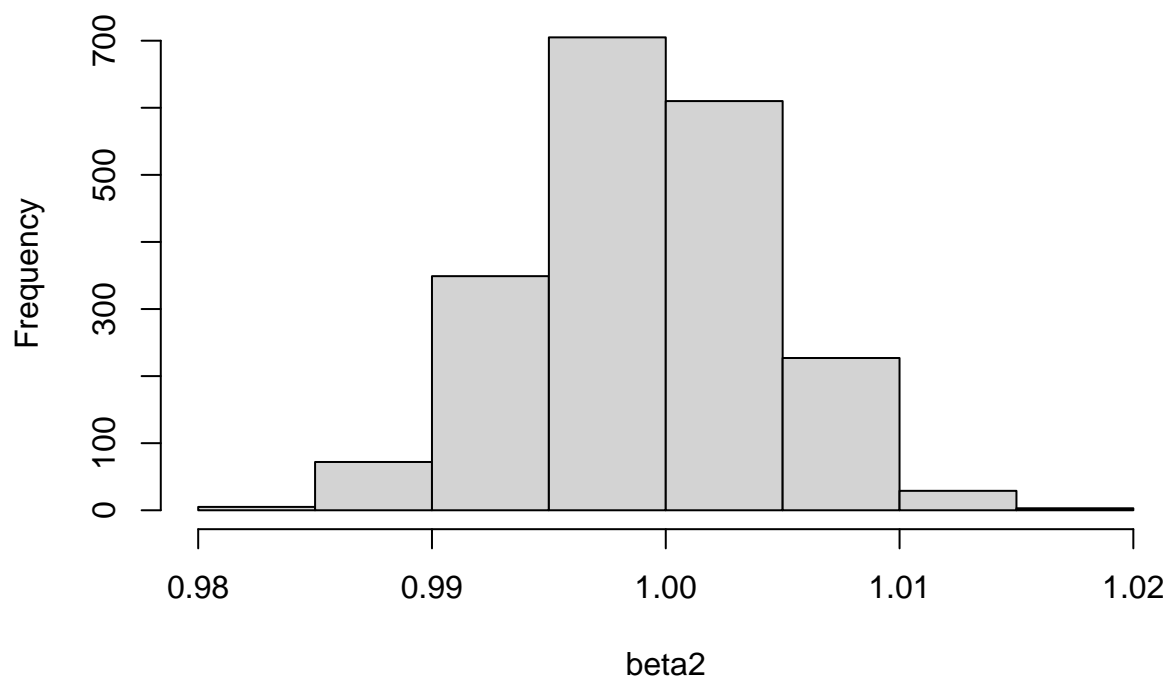
```

Histogram of beta1



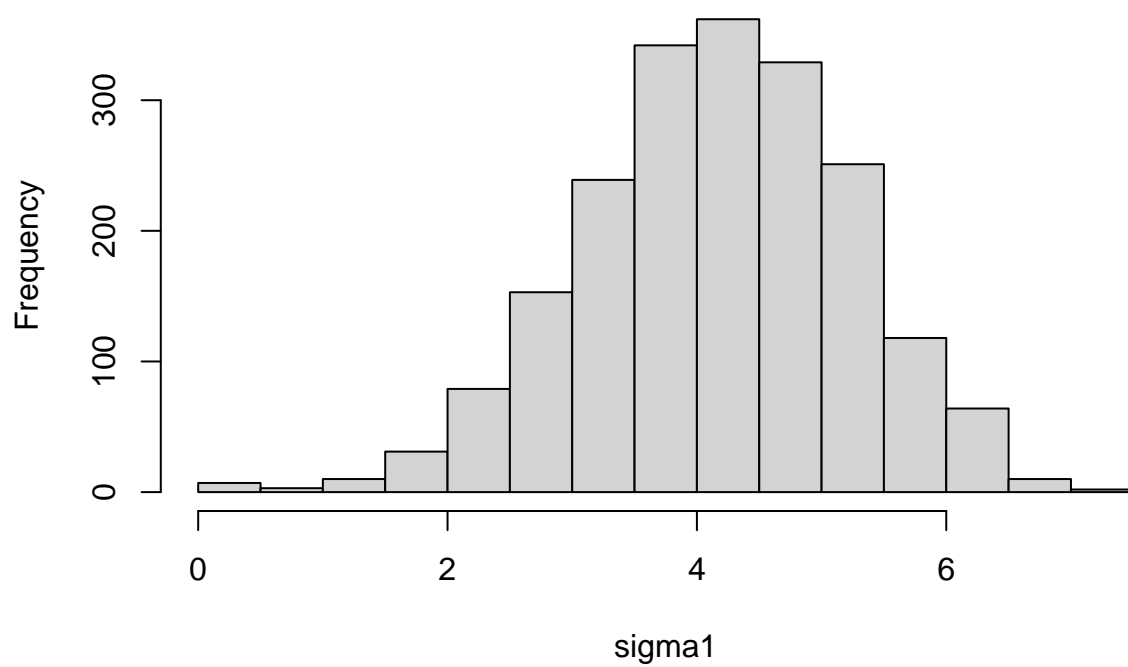
```
hist(beta2)
```


Histogram of beta2



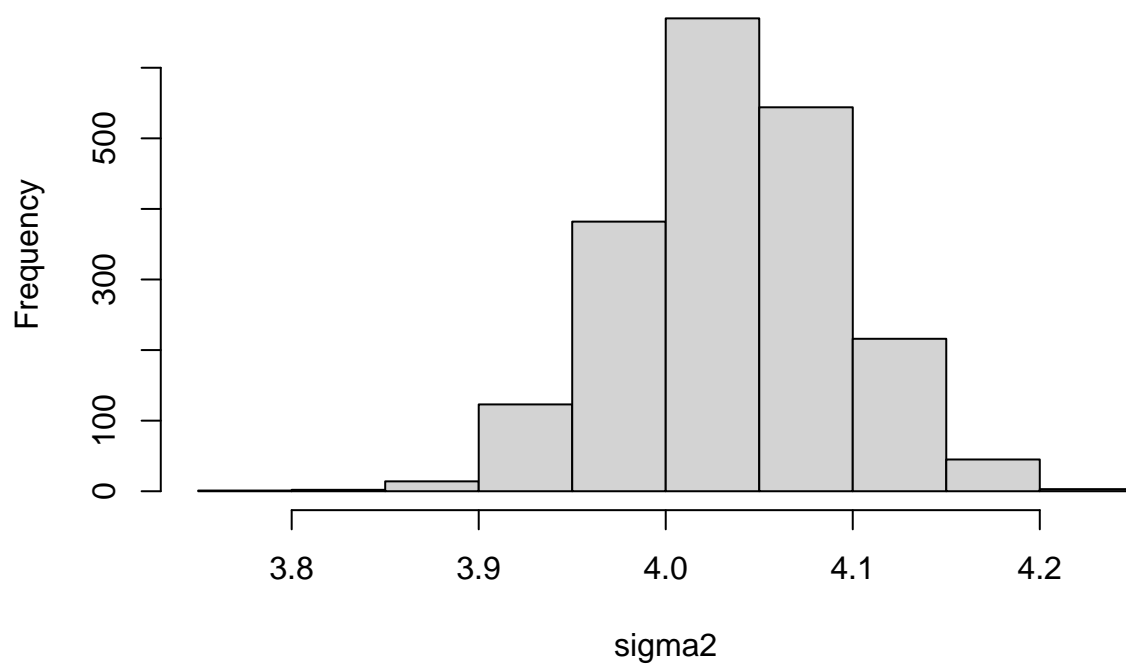
```
hist(sigma1)
```

Histogram of sigma1



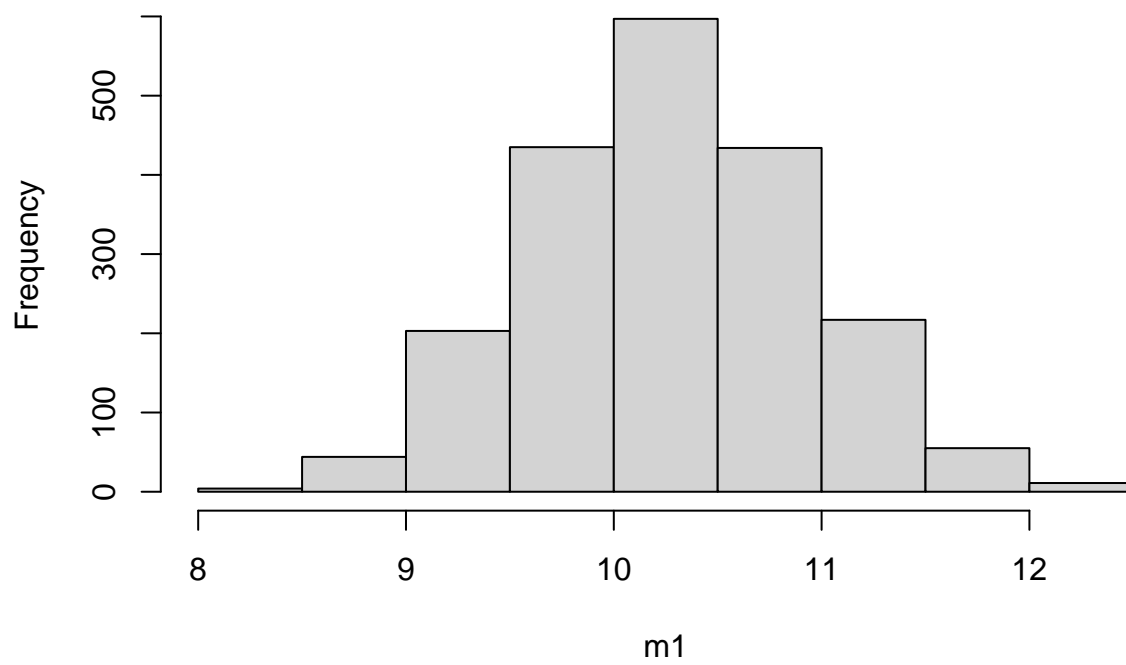
```
hist(sigma2)
```

Histogram of sigma2

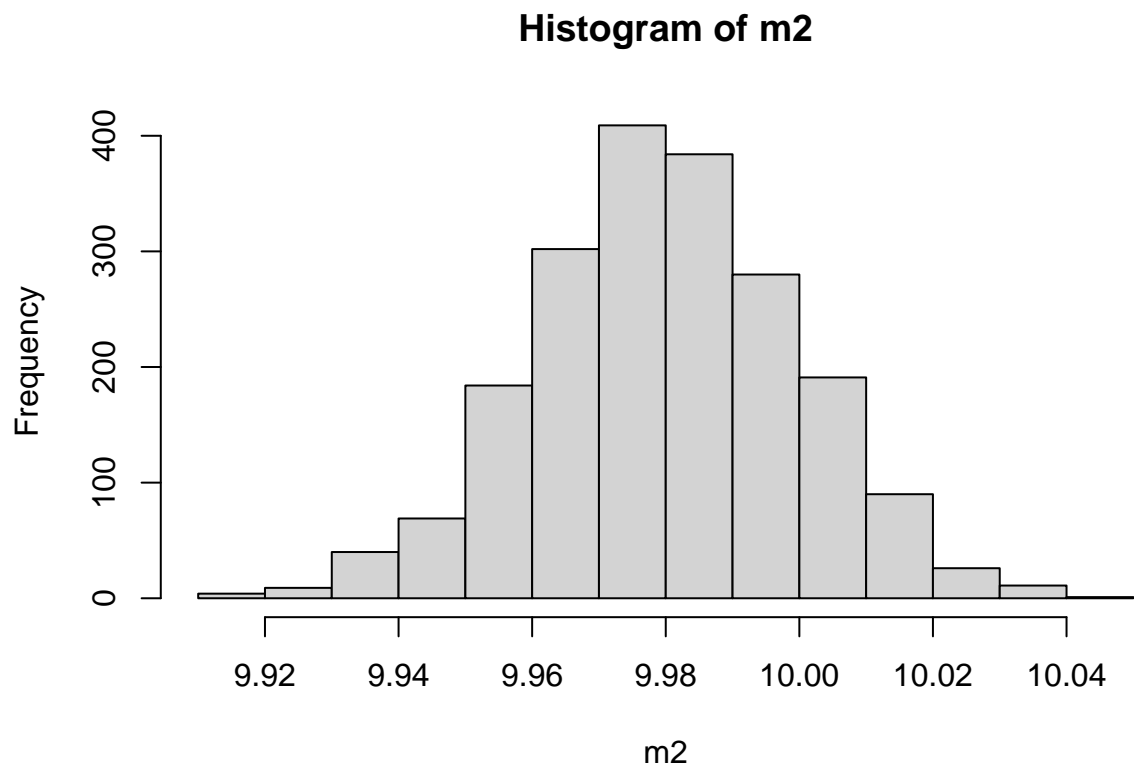


```
hist(m1)
```

Histogram of m1



```
hist(m2)
```



När vi inte längre använder oss a populationen utan istället kör på stickprov så kommer vår data ej bli lika exakt.