

lab2__danhe178__rical803

Daniel Herzegh & Richard Friberg

2017-10-03

Uppgift 1 Likelihoodfunktioner

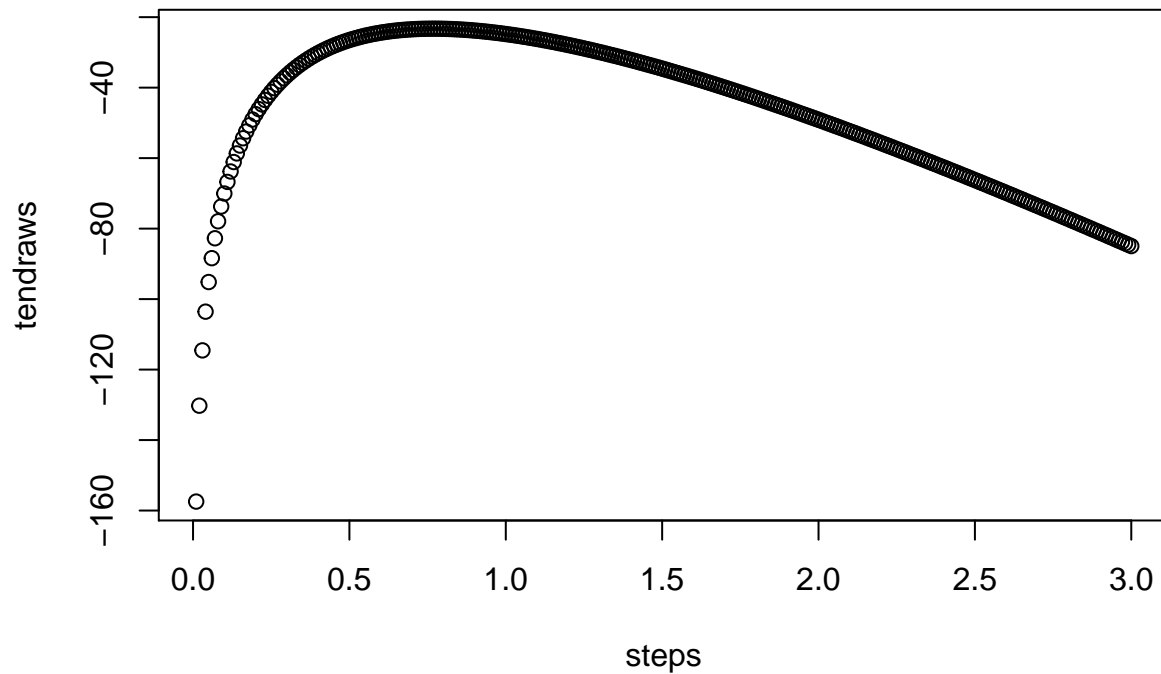
a)

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

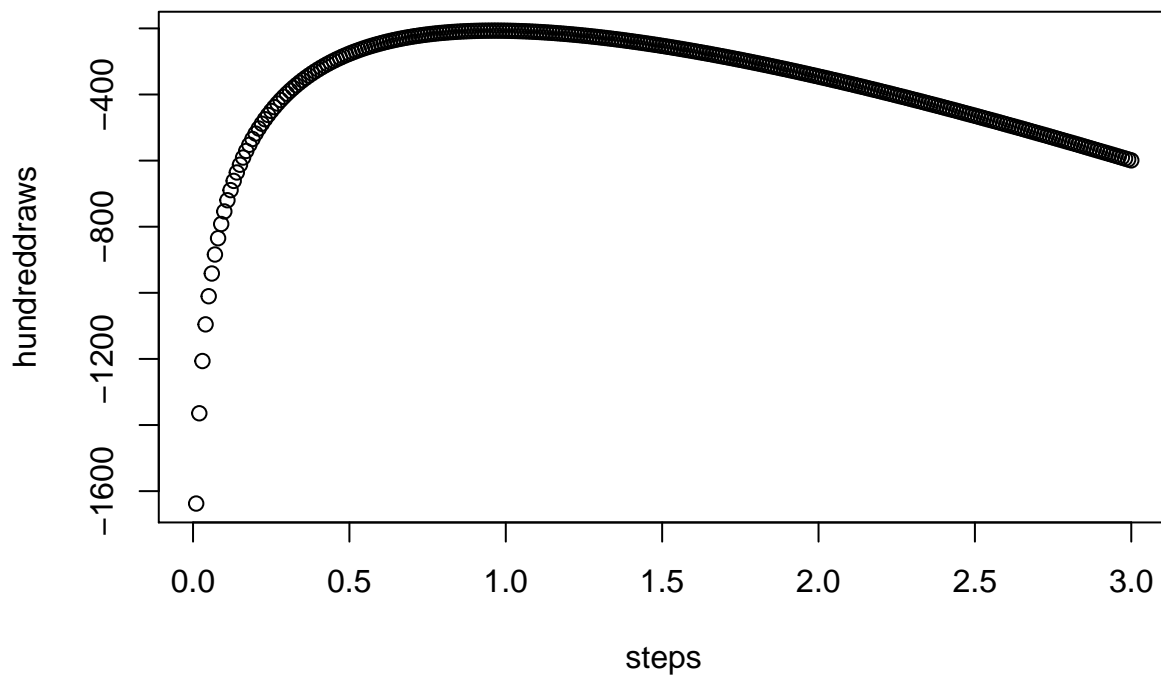
b)

```
set.seed(4711) #FRÅGA OM VI SKA ANVÄNDA SEED ELLER EJ  
x1 <- rgamma(n = 10, shape = 4, scale = 1)  
x2 <- rgamma(n = 100, shape = 4, scale = 1)  
  
tendraws <- c()  
hundreddraws <- c()  
steps <- c()  
  
i = 0.01  
while(i <= 3) {  
  tendraws <- c(tendraws, llgamma(x1, alpha = 4, beta = i))  
  hundreddraws <- c(hundreddraws, llgamma(x2, alpha = 4, beta = i))  
  steps <- c(steps, i)  
  i <- i + 0.01  
}
```

```
# plot for ten draws  
plot(steps, tendraws)
```



```
# plot for hundred draws
plot(steps, hundreddraws)
```



```
# Undersöker och returnerar vilket betavärde som loglikelihoodfunktionen får sitt maxvärde på
findMaxIndex <- function(vect) {
  i <- NULL
  currentMax <- -Inf
  x <- 1
  while (x < length(vect)) {
    if (vect[x] > currentMax) {
```

```

    currentMax <- vect[x]
    i <- x
  }
  x <- x + 1
}
return(i/100)
}

findMaxIndex(tendraws)

```

```
## [1] 0.77
```

```
findMaxIndex(hundreddraws)
```

```
## [1] 0.96
```

Det varierar vilket av de upprepade värdena för beta som ger maximala värdet på loglikelihoodfunktionen, men ökar man antalet dragningar går denna siffra mot 1.0.

c)

```

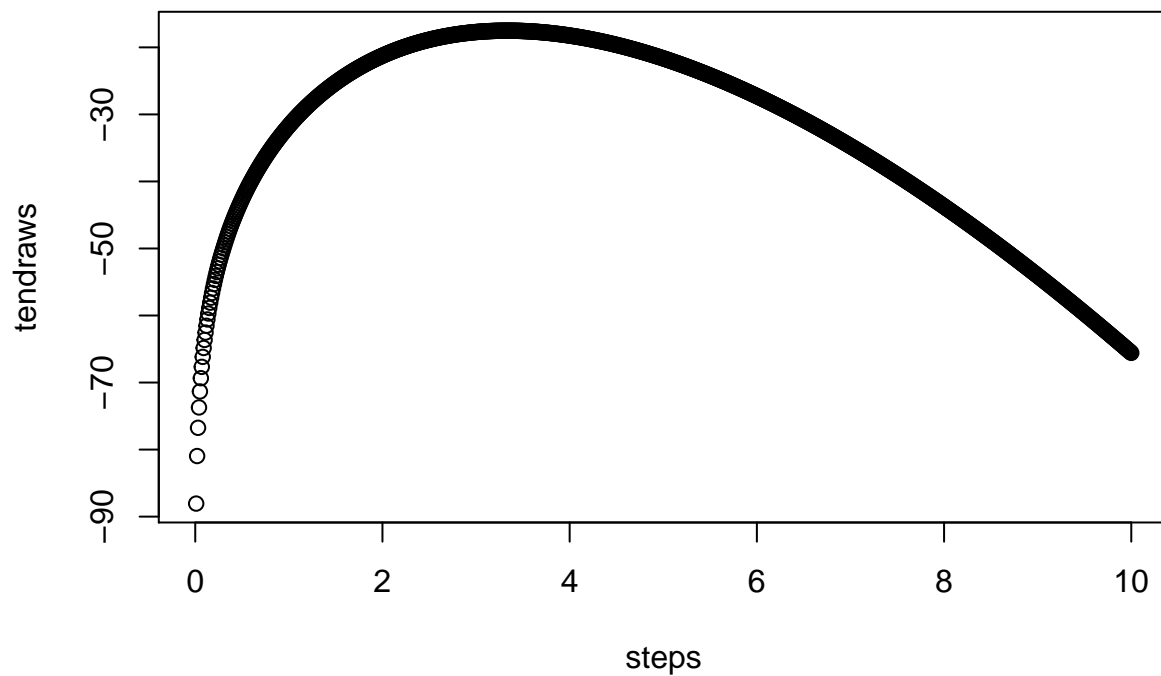
x1 <- rgamma(n = 10, shape = 4, scale = 1)
x2 <- rgamma(n = 100, shape = 4, scale = 1)

tendraws <- c()
hundreddraws <- c()
steps <- c()

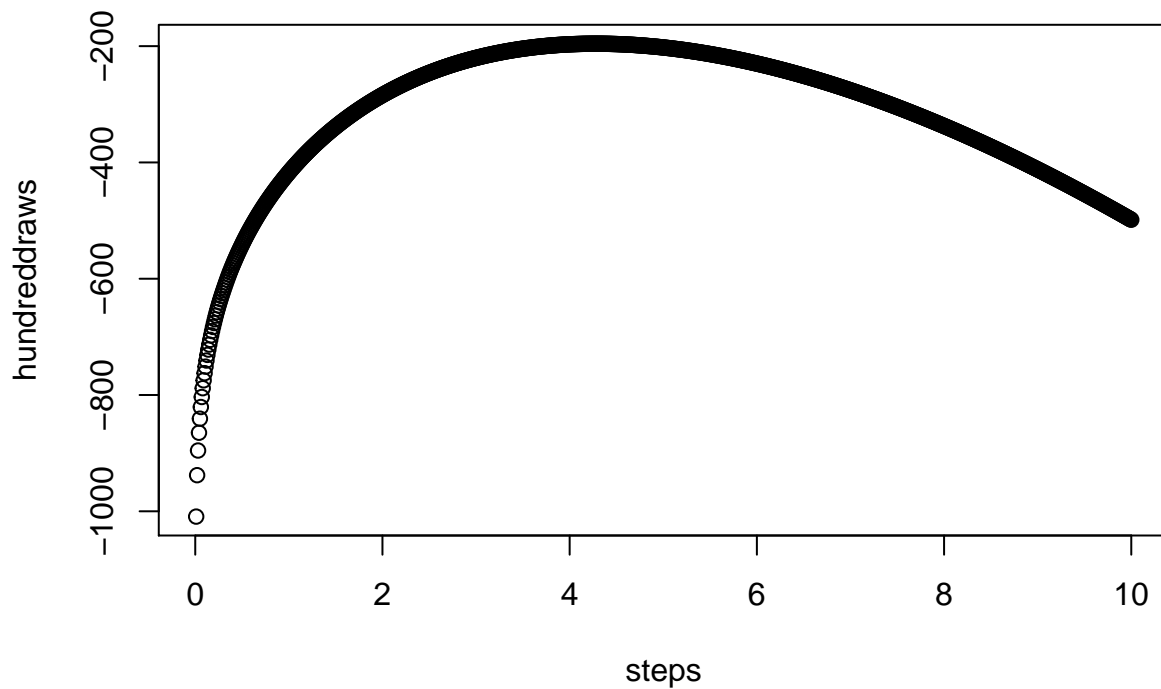
i = 0.01
while(i <= 10) {
  tendraws <- c(tendraws, llgamma(x1, alpha = i, beta = 1))
  hundreddraws <- c(hundreddraws, llgamma(x2, alpha = i, beta = 1))
  steps <- c(steps, i)
  i <- i + 0.01
}

# plot for ten draws
plot(steps, tendraws)

```



```
# plot for hundred draws
plot(steps, hundreddraws)
```



```
# Undersöker och returnerar vilket alphavärde som loglikelihoodfunktionen får sitt maxvärde på
findMaxIndex <- function(vect) {
  i <- NULL
  currentMax <- -999999
  x <- 1
  while (x < length(vect)) {
```

```

    if (vect[x] > currentMax) {
      currentMax <- vect[x]
      i <- x
    }
    x <- x + 1
  }
  return(i/100)
}

findMaxIndex(tendraws)

```

```
## [1] 3.33
```

```
findMaxIndex(hundreddraws)
```

```
## [1] 4.28
```

Det varierar vilket av de upprepade värdena för alpha som ger maximala värdet på loglikelihoodfunktionen, men ökar man antalet dragningar går denna siffra mot 4.0.

d) FRÅGA PÅ LABBEN!

```

llnormal <- function(x, mu, sigma2) {
}

```

e)

Uppgift 2 Punktskattning med MLE i en gammafördelning

```

gamma_beta_mle <- function(x, alpha) {
  return(length(x)*alpha*1/sum(x))
}
x1 <- rgamma(n = 10, shape = 4, scale = 1)
x2 <- rgamma(n = 100, shape = 4, scale = 1)
gamma_beta_mle(x1, 2)

```

```
## [1] 0.4085222
```

```
gamma_beta_mle(x2, 2)
```

```
## [1] 0.4839516
```

Vi testade att öka antalet dragningar och drar slutsatsen att estimatet går mot 0.5

Uppgift 3 Punktskattning med MLE i en normalfördelning

a)

```
norm_mu_mle <- function(x) {  
  return(1/length(x)*sum(x))  
}  
  
norm_sigma2_mle <- function(x) {  
  sumhelp <- 0  
  j <- 1  
  while(j <= length(x)) {  
    sumhelp <- sumhelp + (x[j] - norm_mu_mle(x))**2  
    j <- j + 1  
  }  
  return(1/length(x)*sumhelp)  
}
```

```
test_x <- 1:10
```

```
norm_mu_mle(x = test_x)
```

```
## [1] 5.5
```

```
norm_sigma2_mle(x = test_x)
```

```
## [1] 8.25
```

b)

```
set.seed(42)  
# Skattning med n = 10  
y1 <- rnorm(n = 10, mean = 10, sd = 2)  
  
norm_mu_mle(x = y1)
```

```
## [1] 11.09459
```

```
norm_sigma2_mle(x = y1)
```

```
## [1] 2.512709
```

```
# Skattning med n = 10000  
y2 <- rnorm(n = 10000, mean = 10, sd = 2)  
  
norm_mu_mle(x = y2)
```

```
## [1] 9.9762
```

```
norm_sigma2_mle(x = y2)
```

```
## [1] 4.048198
```

Desto större antal dragningar som görs, desto närmare kommer vi μ och σ^2 , med respektive `norm_mu_mle` och `norm_sigma2_mle`. Detta följer av centralagränsvärdessatsen som ger oss ett y som går mot normalfördelning och därmed tydligare väntevärde samt varians.

FRÅGA OM MAN KAN SÄGA ATT EN VEKTOR MED TAL GÅR MOT EN VISS FÖRDELNING

Uppgift 4 Samplingfördelningen för `Bmle`, `MUmle` och `sigma2mle`

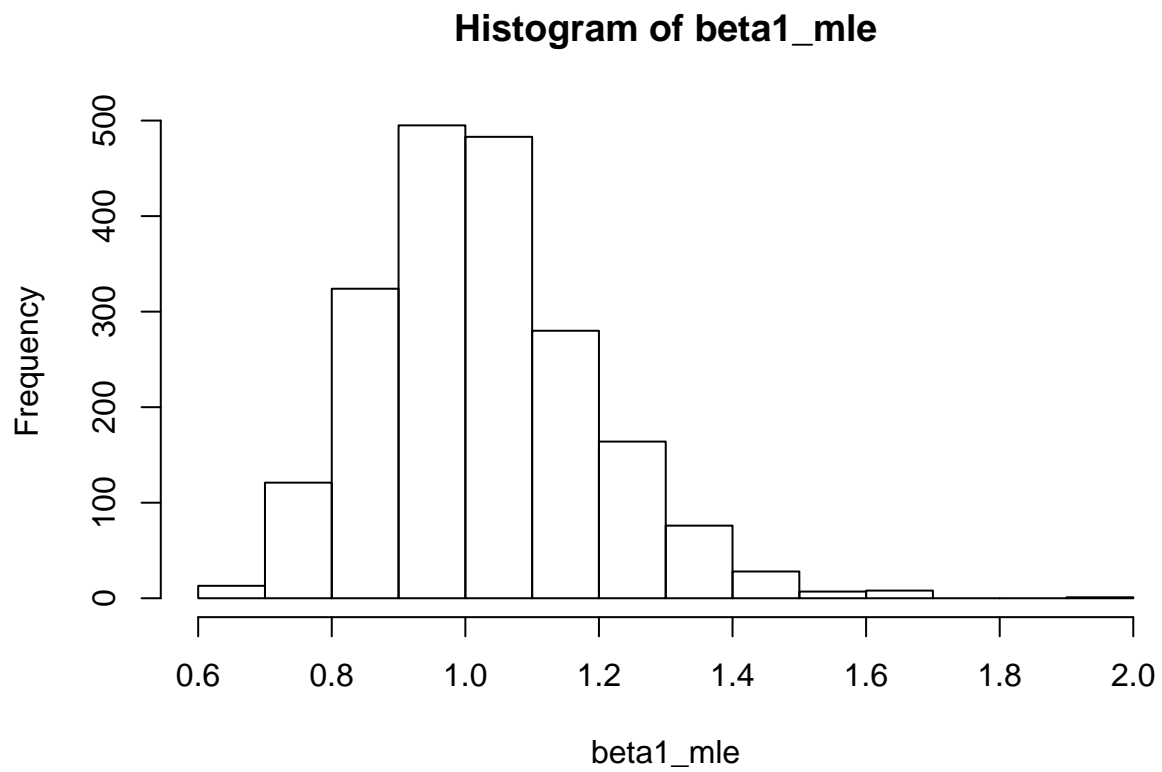
a)

```
beta1_mle <- c(1:2000)
beta2_mle <- c(1:2000)
mu1 <- c(1:2000)
mu2 <- c(1:2000)
sigma1 <- c(1:2000)
sigma2 <- c(1:2000)

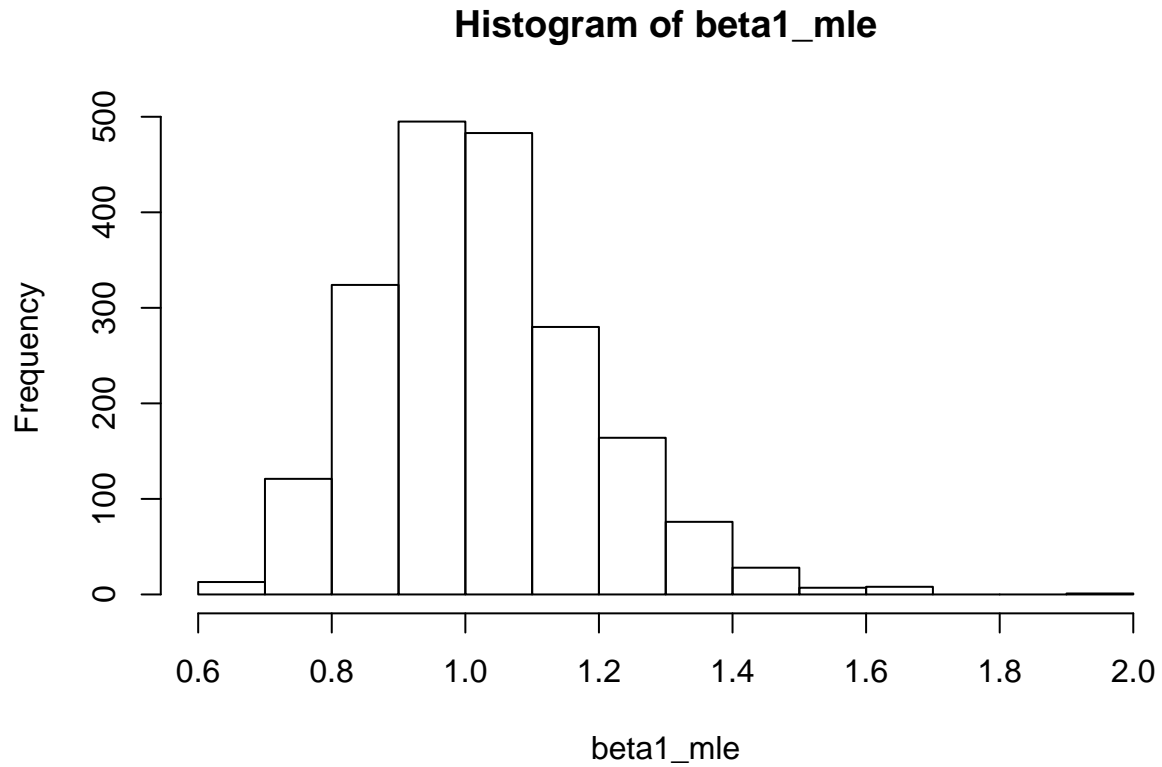
i <- 1
while (i <= 2000) {
  x1 <- rgamma(n = 10, shape = 4, rate = 1)
  x2 <- rgamma(n = 10000, shape = 4, rate = 1)
  beta1_mle[i] <- gamma_beta_mle(x = x1, alpha = 4)
  beta2_mle[i] <- gamma_beta_mle(x = x2, alpha = 4)

  y1 <- rnorm(n = 10, mean = 10, sd = 2)
  y2 <- rnorm(n = 10000, mean = 10, sd = 2)
  mu1[i] <- norm_mu_mle(x = y1)
  mu2[i] <- norm_mu_mle(x = y2)
  sigma1[i] <- norm_sigma2_mle(x = y1)
  #sigma2[i] <- norm_sigma2_mle(x = y2) #FRÅGA: GÅR LÅNGSAMT. HOW TO MAKE IT GO FASTER??? CAN WE DO THIS?
  i <- i + 1
}

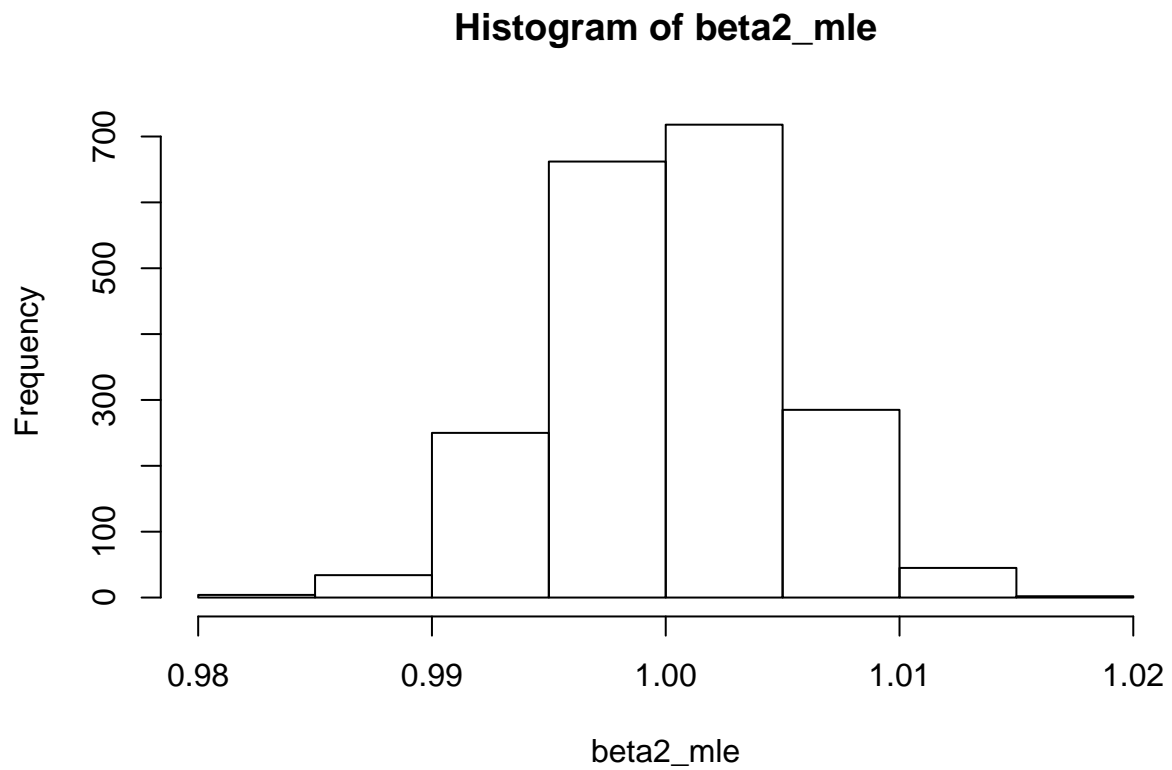
hist(beta1_mle)
```



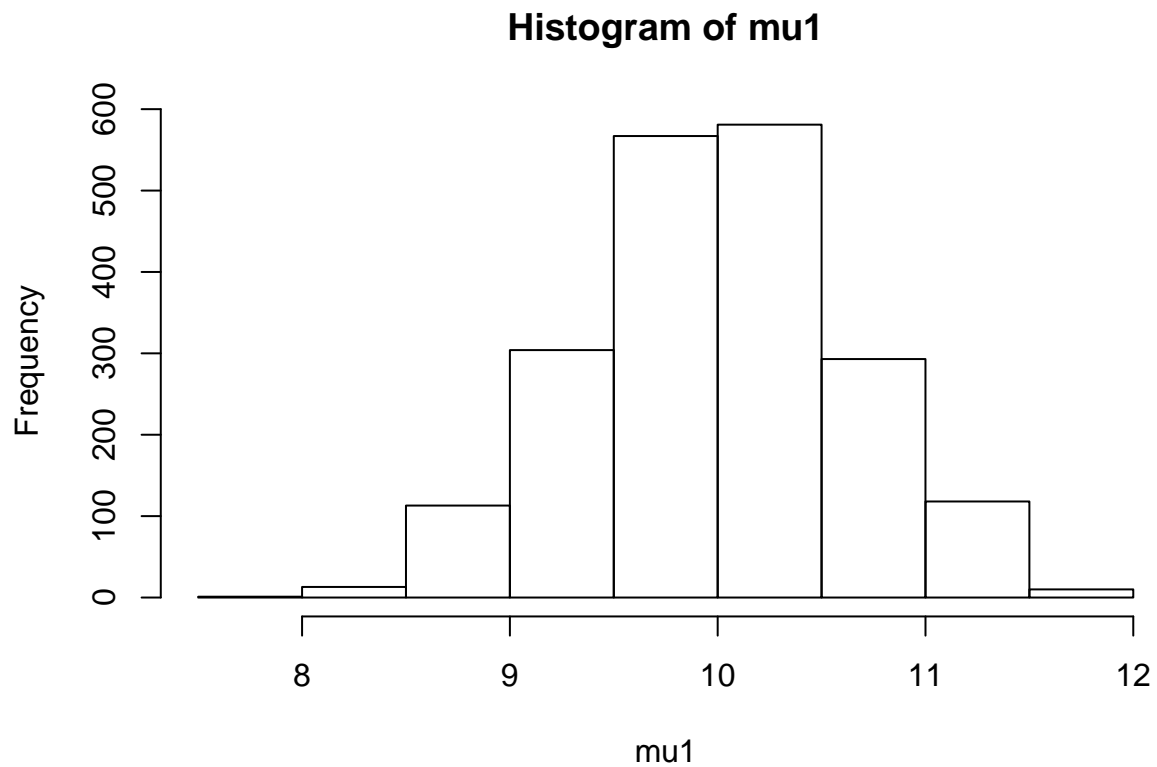
```
hist(beta1_mle)
```



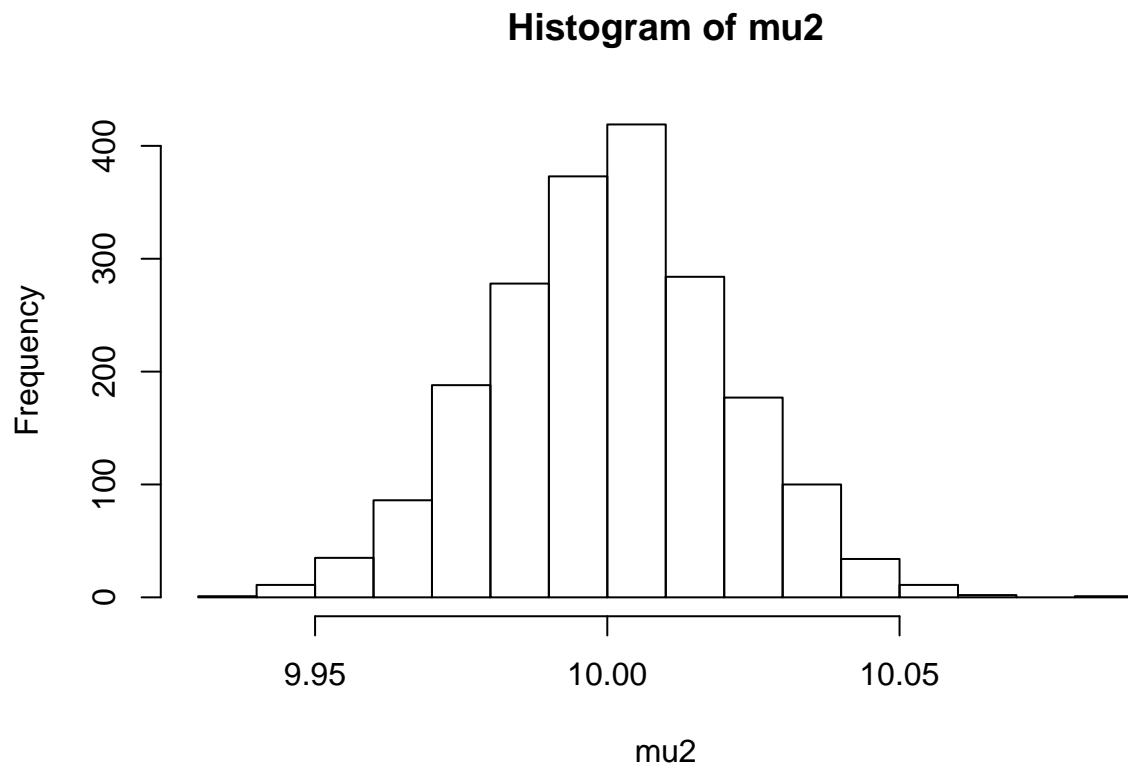

```
hist(beta2_mle)
```



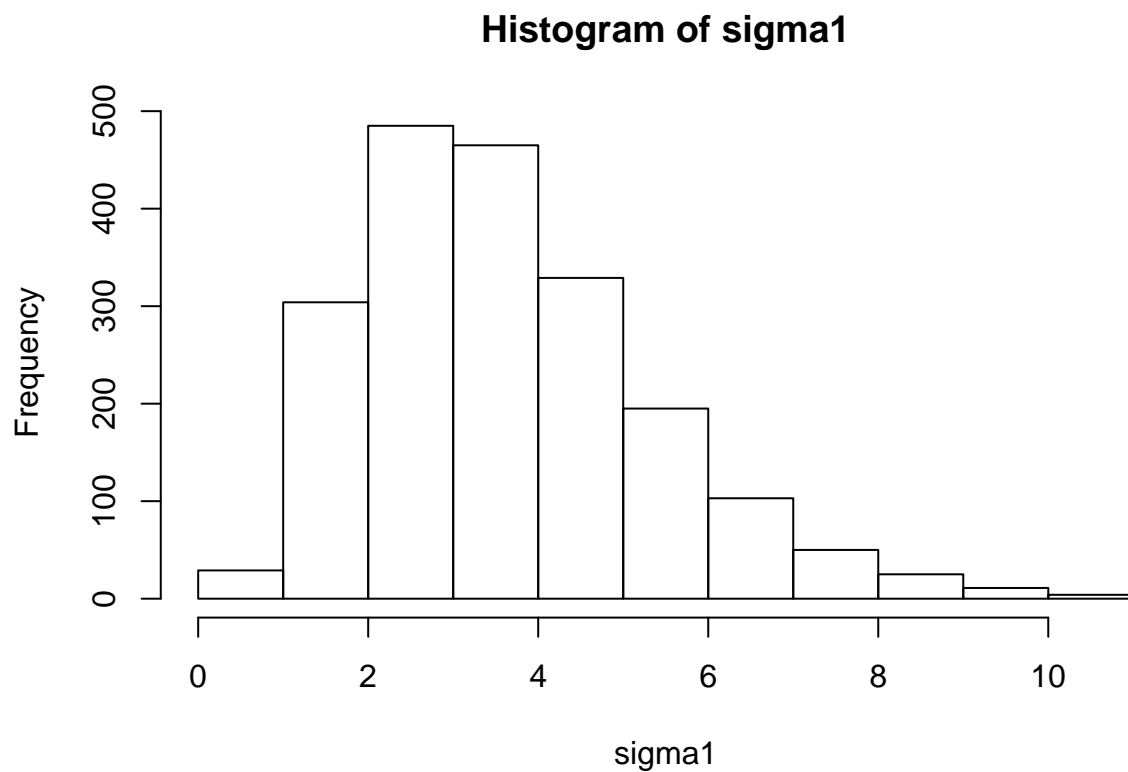
```
hist(mu1)
```



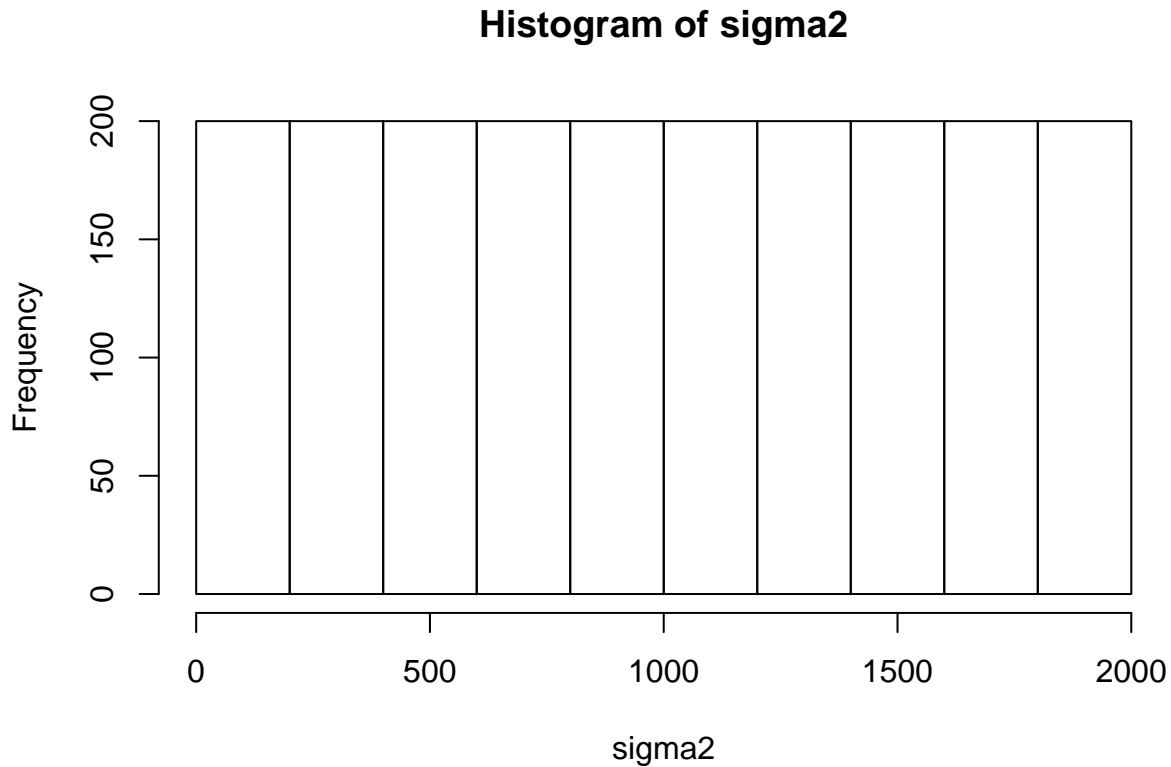
```
hist(mu2)
```



```
hist(sigma1)
```



```
hist(sigma2)
```



Precis som tidigare ser vi att ju fler dragningar så närmar sig histogrammen en normalfördelning vilket följer av den centrala gränsvärdessatsen.

Uppgift 5 Log-likelihoodfunktionen för betafördelning

a)

```
llbeta <- function(par, x){  
  helpsum <- 0  
  i <- 1  
  while(i <= length(x)) {  
    helpsum <- helpsum + logb(par[1] + 1, x[i] + par[2])  
    i <- i + 1  
  }  
  return (helpsum + length(x)*logb(par[1], par[2]))  
}  
llbeta(par = c(2, 2), x = c(0.01, 0.5, 0.99))
```

```
## [1] 6.775666
```

FRÅGA: OPTIM()?, MULTIPLICERAD MED -1???

relevant länk (med vår formel): <https://stats.stackexchange.com/questions/137989/how-do-you-work-out-the-likelihood-function-for-the-beta>