Lab 2

Martin Gustafsson (margu424) and Axel Gard (axega544)

2020-10-02

3.1.1

Likelihoodfunktioner

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

(1)

Skapa en funktion i R som du kallar llgamma och returnerar log-likelihoodvärdet för parametrarna

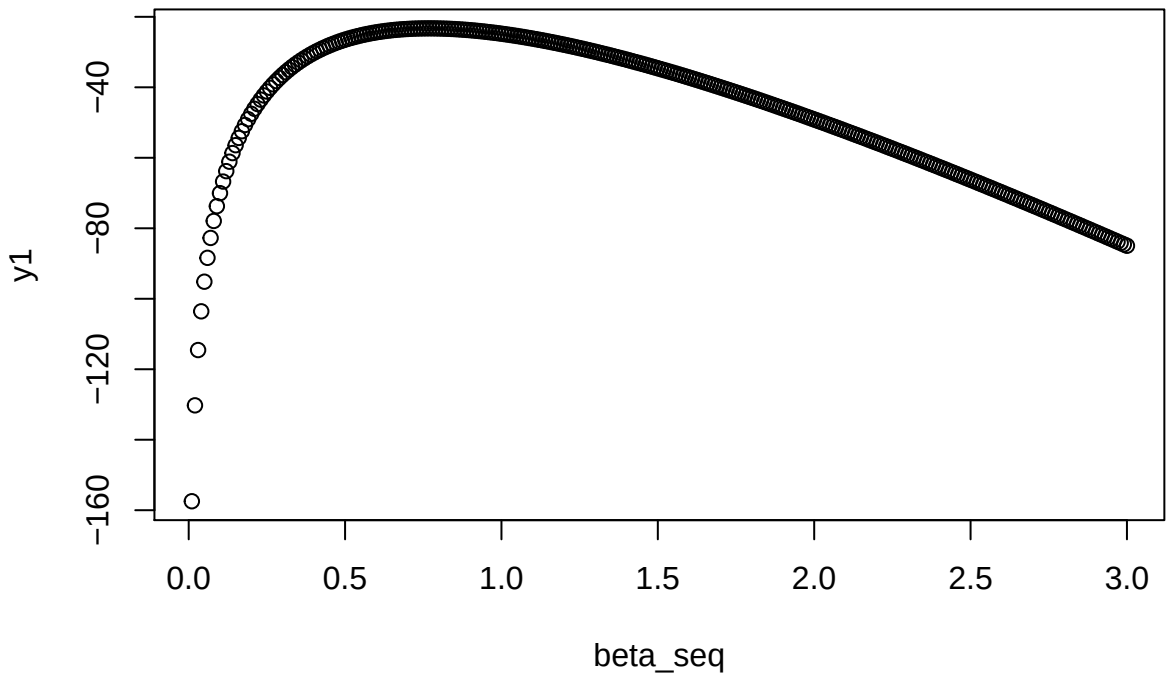
```
    llgamma <- function(x, alpha, beta) {
        return(length(x) * (alpha * log(beta) - lgamma(alpha)) + (alpha - 1) * sum(log(x)) - (beta*sum(
    }

    print(llgamma(x = x1, alpha = 2, beta = 2))
```

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

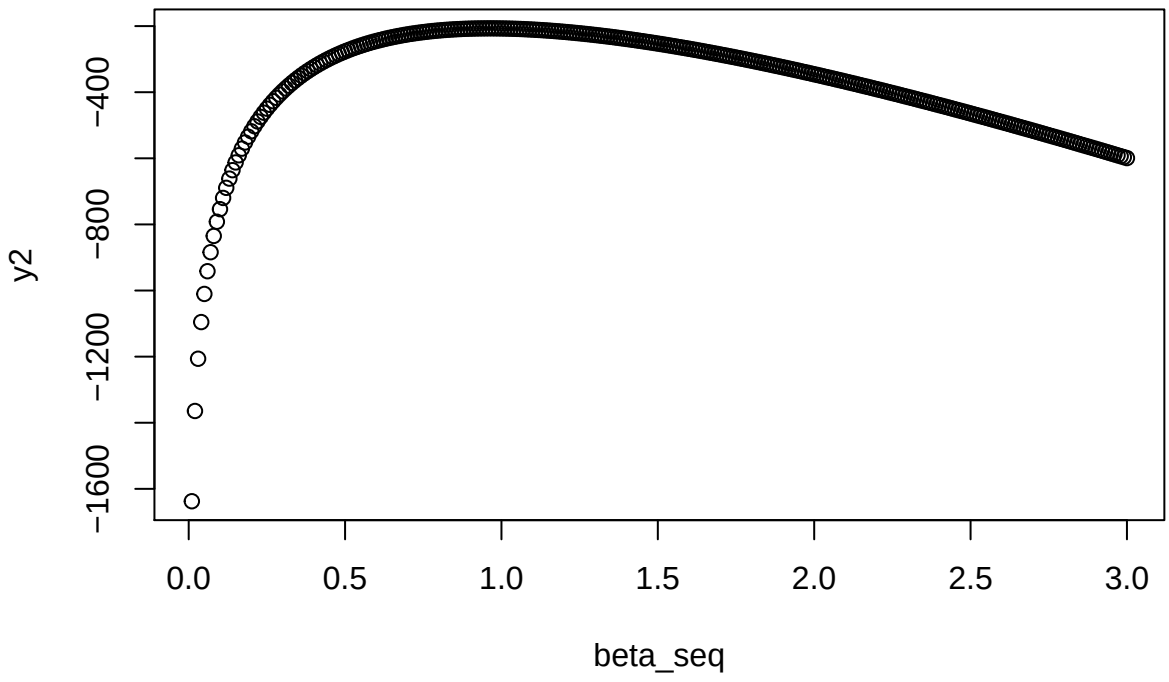
(2)

Beräkna och visualisera loglikelihood värden för x1 och x2 som simulerades då alpha = 4.

```
    alpha <- 4
    beta_seq <- seq(0.01, 3, 0.01)
    y1 <- numeric(0)
    for (beta in beta_seq) {
        y1 <- c(y1, llgamma(x = x1, alpha = alpha, beta = beta))
    }
    y2 <- numeric(0)
    for (beta in beta_seq) {
        y2 <- c(y2, llgamma(x = x2, alpha = alpha, beta = beta))
    }
```




```
## [1] "max value at beta = 0.77"
```




```
## [1] "max value at beta = 0.96"
```

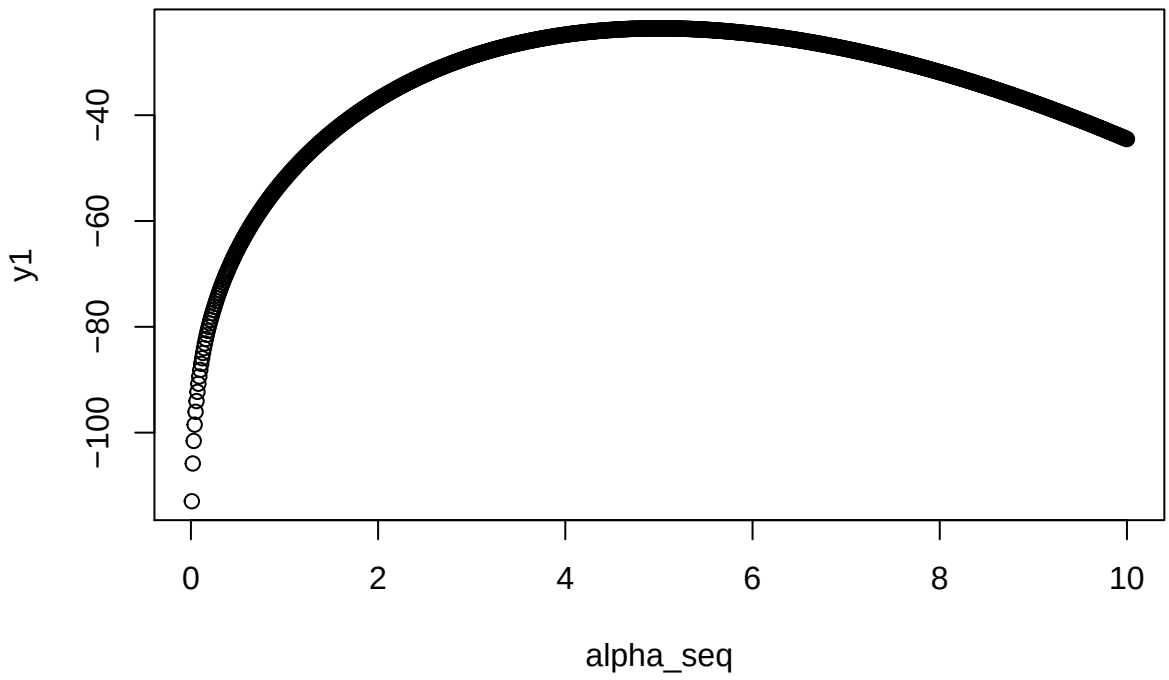
y1 har sitt max vi omkring 0.75 (0.77 exakt).

y2 har sitt max vi omkring 1.0 (0.96 exakt).

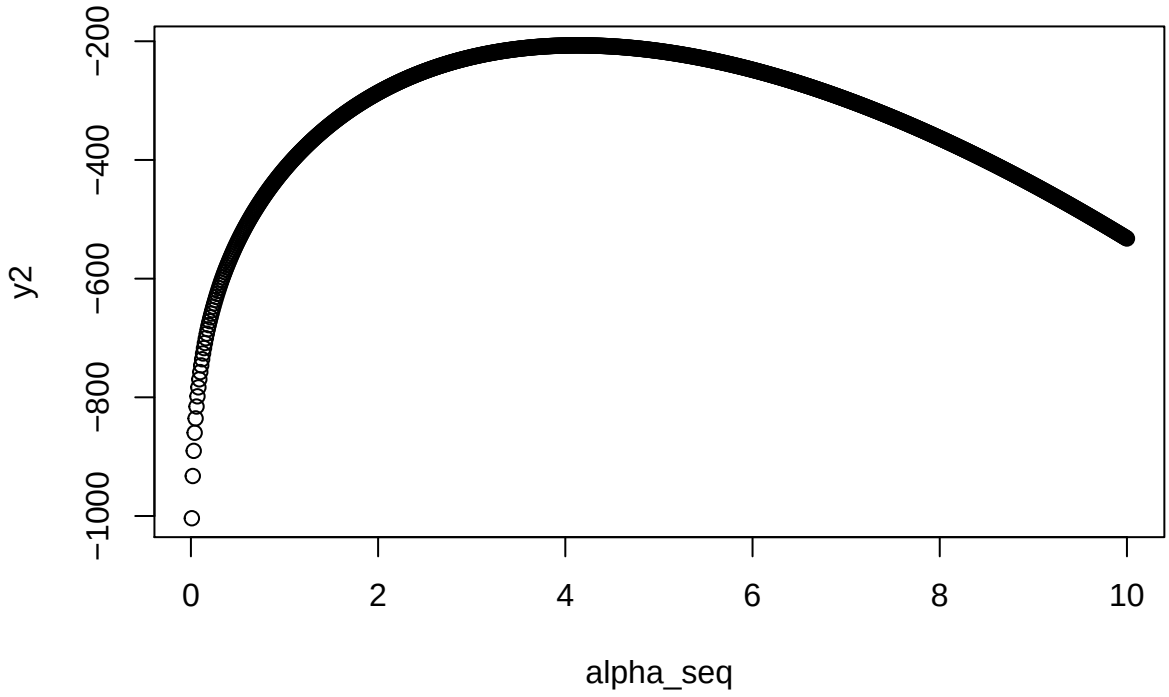
(3)

Vilket värde på alpha ger det maximala värdet för log-likelihood-funktionen?

```
beta <- 1
alpha_seq <- seq(0.01, 10, 0.01)
y1 <- numeric(0)
for (alpha in alpha_seq) {
    y1 <- c(y1, llgamma(x = x1, alpha = alpha, beta = beta))
}
y2 <- numeric(0)
for (alpha in alpha_seq) {
    y2 <- c(y2, llgamma(x = x2, alpha = alpha, beta = beta))
}
```




```
## [1] "max value at alpha = 5"
```




```
## [1] "max value at alpha = 4.13"
```

y1 har sitt max vi omkring 5 (5 exakt).

y2 har sitt max vi omkring 4 (4.13 exakt).

(4)

Täthetsfunktionen för en normalfördelning ges av:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} * e^{\frac{-1}{2}(\frac{(x-\mu)^2}{\sigma^2})}$$

Eftersom värdena är oberoende kan sannolikheterna vi får multipliceras:

$$\prod_{i=1}^{n} f(x_i) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} * e^{\frac{-1}{2}(\frac{(x_i-\mu)^2}{\sigma^2})} =$$

$$= (\frac{1}{\sqrt{2\pi\sigma^2}})^n * \prod_{i=1}^{n} e^{\frac{-1}{2}(\frac{(x_i-\mu)^2}{\sigma^2})}$$

Detta är likelyhood-funktion, så vi tar log av denna för att få log-likelihoodfunktionen:

$$\ln((\frac{1}{\sqrt{2\pi\sigma^2}})^n * \prod_{i=1}^{n} e^{\frac{-1}{2}(\frac{(x_i-\mu)^2}{\sigma^2})}) =$$

$$= \ln((\frac{1}{\sqrt{2\pi\sigma^2}})^n) + \sum_{i=1}^{n} \frac{-1}{2}(\frac{(x_i - \mu)^2}{\sigma^2}) =$$

$$= \frac{-n}{2} * \ln(2\pi\sigma^2) + \frac{-1}{2} * \frac{1}{\sigma^2} * \sum_{i=1}^{n}(x_i - \mu)^2 =$$

$$= \frac{-n}{2} * \ln(2\pi\sigma^2) + \frac{-1}{2\sigma^2} * \sum_{i=1}^{n}(x_i - \mu)^2$$

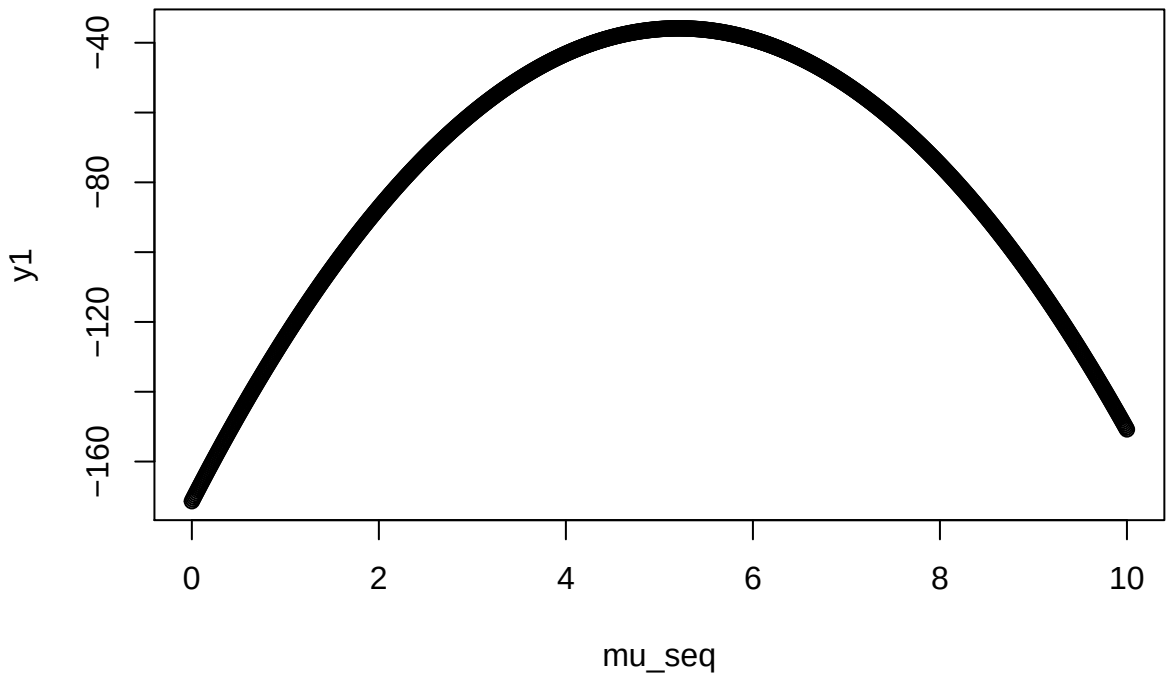
```
llnorm <- function(x, mu, sigma2){
  n <- length(x)
  return((-n/2) * log(2*pi*sigma2) + (-1/(2*sigma2)) * sum((x - mu)^2))
}
print(llnorm(x = x1, mu = 2, sigma2 = 1))
```

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

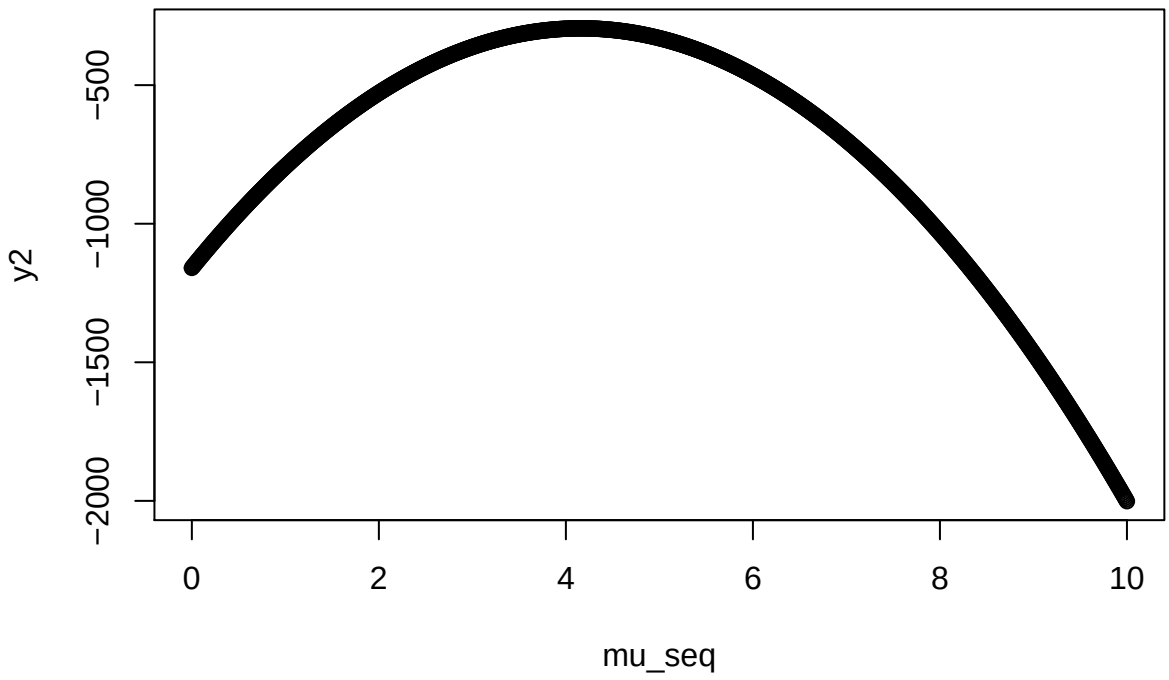
(5)

Vilken modell, normalfördelningen eller gammafördelningen, tycker du passar datamaterialet bäst?

```
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))
}
```

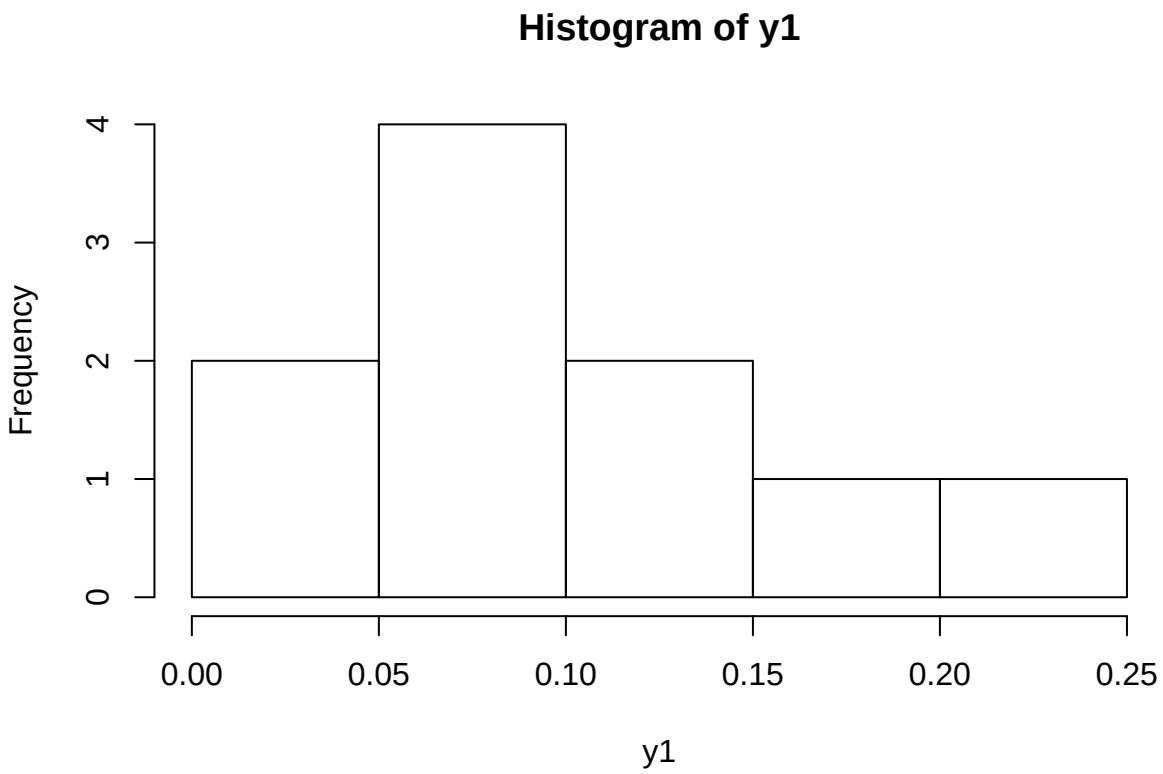



```
## [1] "max value at alpha = 5.21"
```

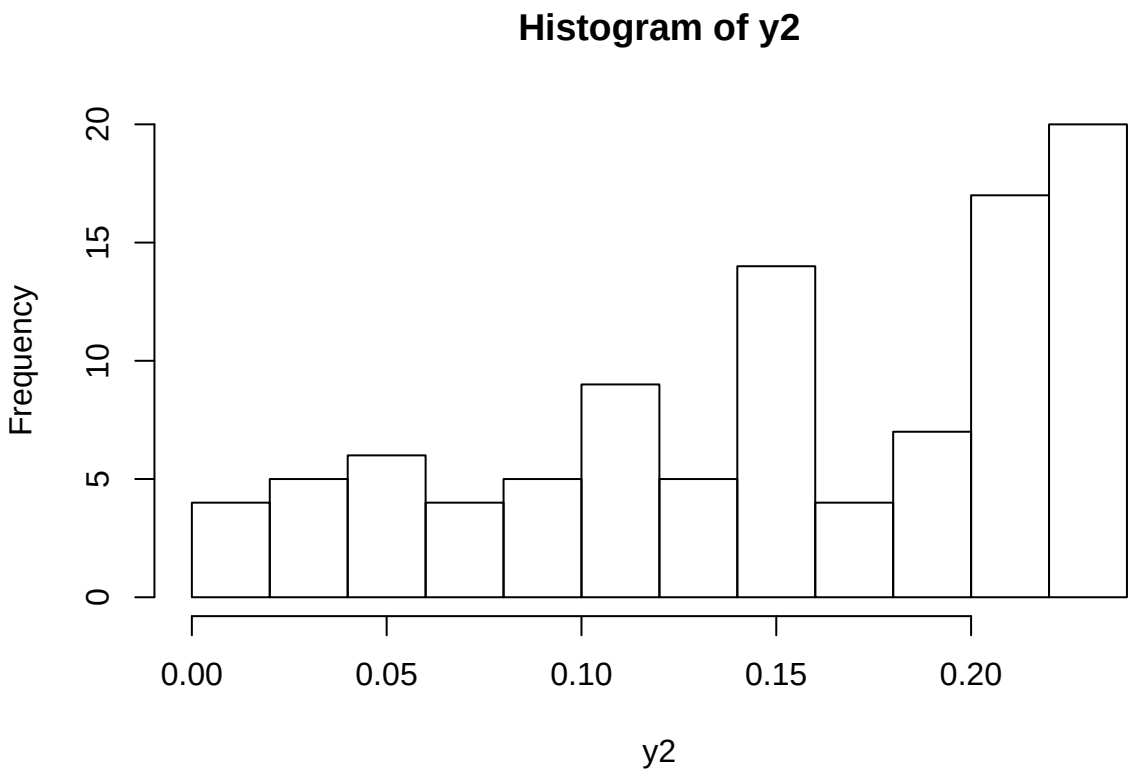



```
## [1] "max value at alpha = 4.16"
```

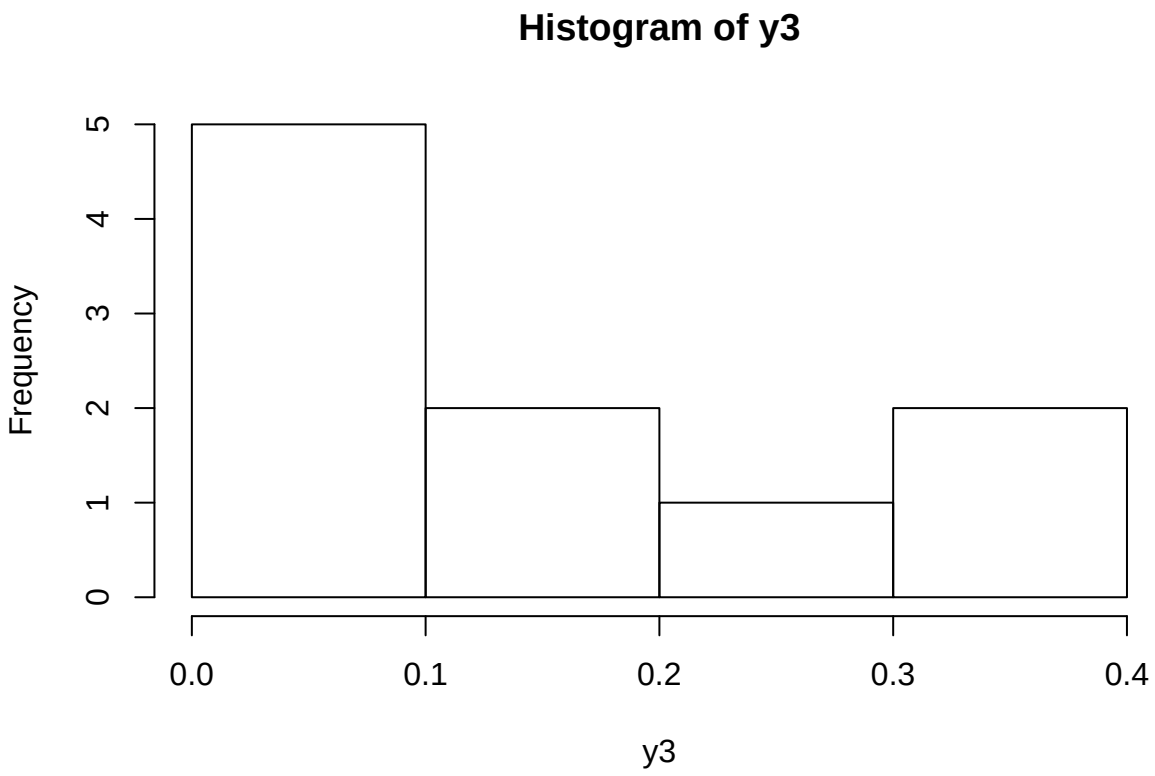
```
    y1 <- dgamma(x1, shape = alpha_max_x1, scale = beta_max_x1)
    hist(y1)
```



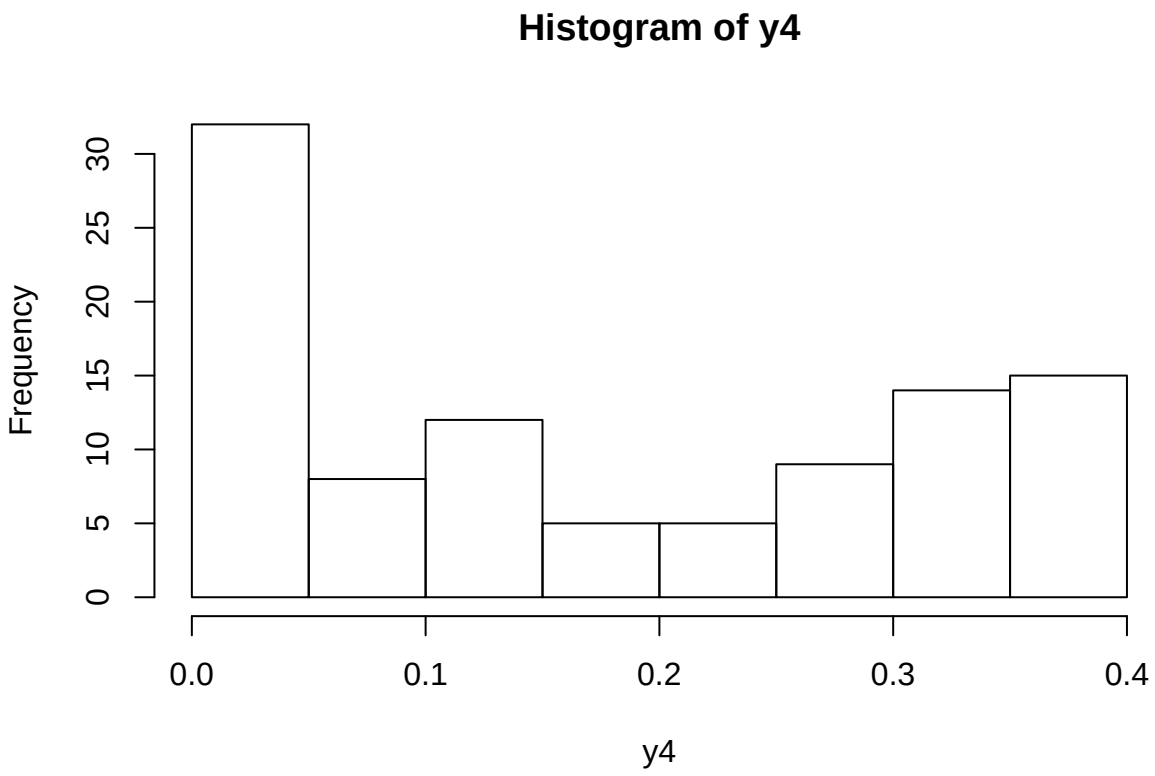

```
y2 <- dgamma(x2, shape = alpha_max_x2, scale = beta_max_x2)
hist(y2)
```



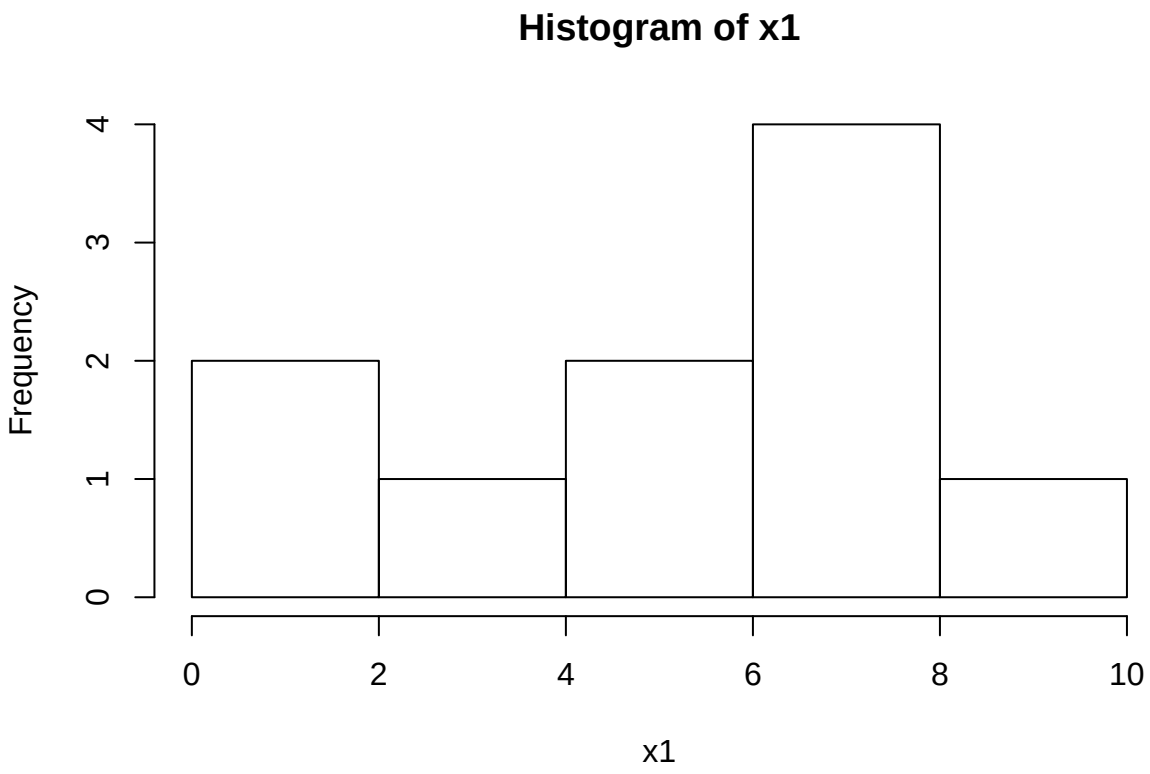

```
y3 <- dnorm(x1, mean = mu_max_x1, sd = 1)
hist(y3)
```



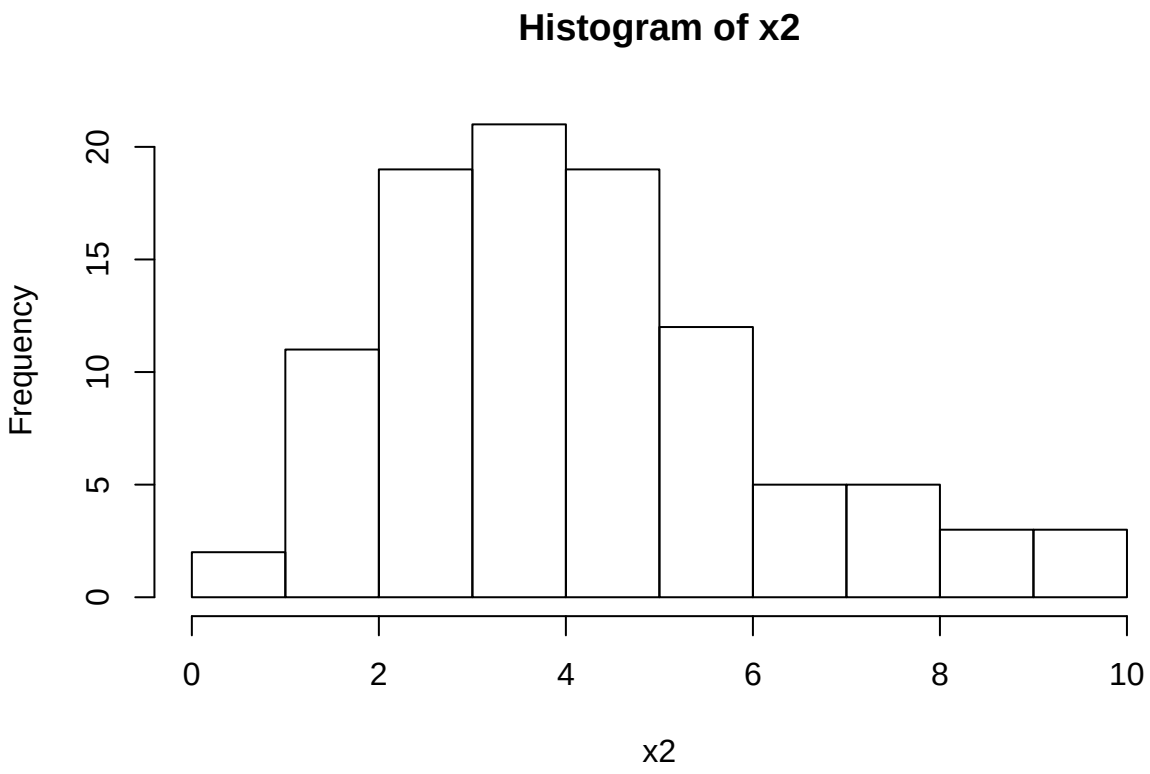

```
y4 <- dnorm(x2, mean = mu_max_x2, sd = 1)
hist(y4)
```




```
hist(x1)
```




```
hist(x2)
```



Om vi tittar på y3 och y4 så ser vi att dessa har en stor del av värdena hamnar kring 0 vilket då säger oss att en normalfördelning troligtvis inte skapade x1 och x2.

Om vi sedan tittar på y1 och y2 så ser vi att fler värden hamnar kring högre sannolikheter vilket leder oss till att tro att en gammafördelning har skapat x1 och x2.

3.2.1

Implementera estimatorn ovan somen funktion du kallar gamma_beta_mle(x, alpha) med parametrarna x (data) och alpha.

```
    gamma_beta_mle <- function(x, alpha) {
        return(length(x) * alpha / sum(x))
    }
    print(gamma_beta_mle(x = x1, alpha = 4))
```

```
## [1] 0.7683785
```

```
    print(gamma_beta_mle(x = x2, alpha = 4))
```

```
## [1] 0.9619473
```

För x1 så maximeras sannolikheterna för att få dessa värden då beta = 0.768 då alpha är 4 För x2 så maximeras sannolikheterna för att få dessa värden då beta = 0.961 då alpha är 4

3.2.2

Punktskattning med MLE i en normalfördelning.

(1)

MLE-estimatorn för mu och sigma2 i en normalfördelning

```
    norm_mu_mle <- function(x) {
        return(sum(x) / length(x))
    }

    norm_sigma2_mle <- function(x) {
        xhat <- norm_mu_mle(x)
        return(sum((x - xhat)^2) / length(x))
    }
    test_x <- 1:10
    print(norm_mu_mle(x = test_x))
```

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

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

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

(2)

Använd dina två estimatorer för att först skatta mu och sigma2. Vad är skillnaden mellan 10 och 10000 dragningar? Vad beror detta på?

```
    set.seed(42)
    y10 <- rnorm(n = 10, mean = 10, sd = 2) # sqrt(4) = 2
    y10000 <- rnorm(n = 10000, mean = 10, sd = 2)
    print(norm_mu_mle(x = y10))
```

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

```
    print(norm_sigma2_mle(x = y10))
```

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

```
    print(norm_mu_mle(x = y10000))
```

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

```
    print(norm_sigma2_mle(x = y10000))
```

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

Skillnaden blir att för den med 10000 dragningar så kommer vi närmare mu = 10 och sigma2 = 4 vilket är det vi skickat in till rnorm. så när antal dragningar går mot oändligheten så kommer mu gå mot 10 och sigma2 mot 4

3.3.1

Log-likelihoodfunktionen for betafördelningen

(1)

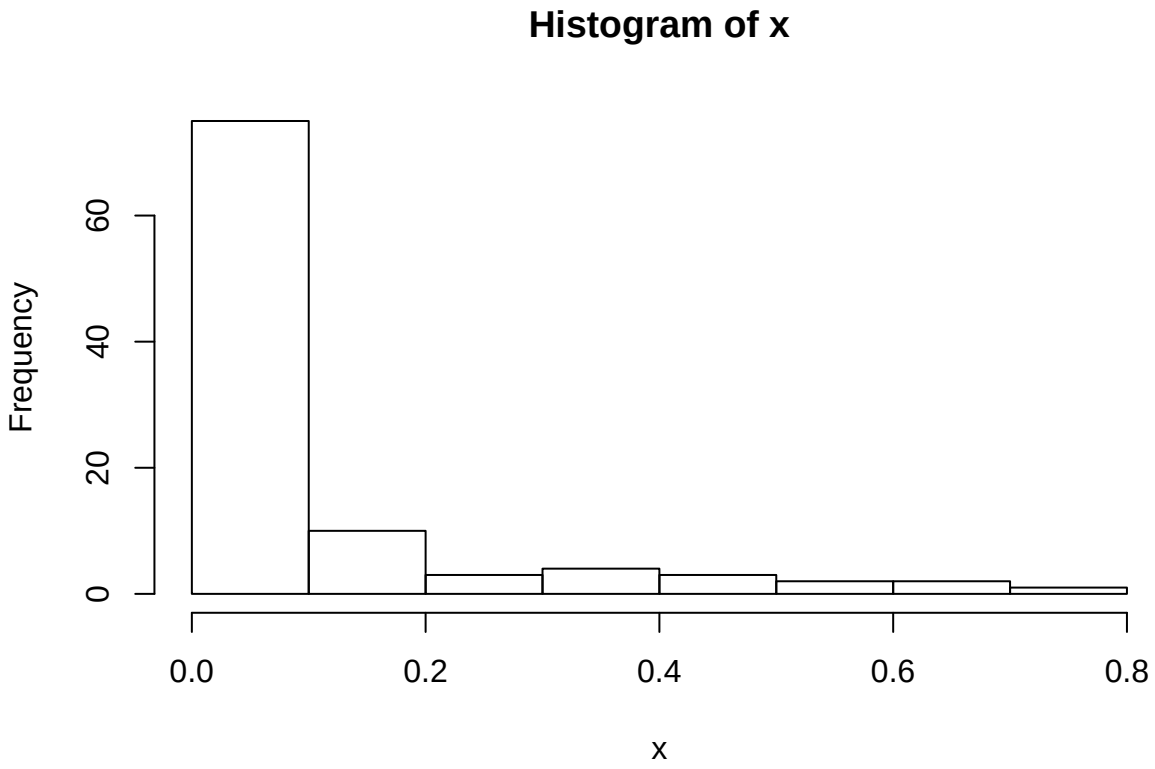
leta reda på log-likelihoodfunktionen för betafördelningen log-likelihoodfunktionen fås genom att ta log av produkten av Täthetsfunktionen, denna multipliceras sedan med -1.

```
    llbeta <- function(par, x) {
      return(-sum(dbeta(x, par[1], par[2], log = TRUE)))
    }
```

(2)

Simulera 100 dragningar från Beta och visa i histogram.

```
x <- rbeta(n = 100, shape1 = 0.2, shape2 = 2)
hist(x)
```



(3)

Använd optim() för att baserat på dessa dragningar och log-likelihoodfunktionen uppskatta parametrarna alpha och beta i betafördelningen.

```
opt_res <- optim(par = c(1, 1), fn = llbeta, x = x, method = "L-BFGS-B", lower = 0.000001)
print(opt_res$par)
```

```
## [1] 0.2211375 2.1439056
```

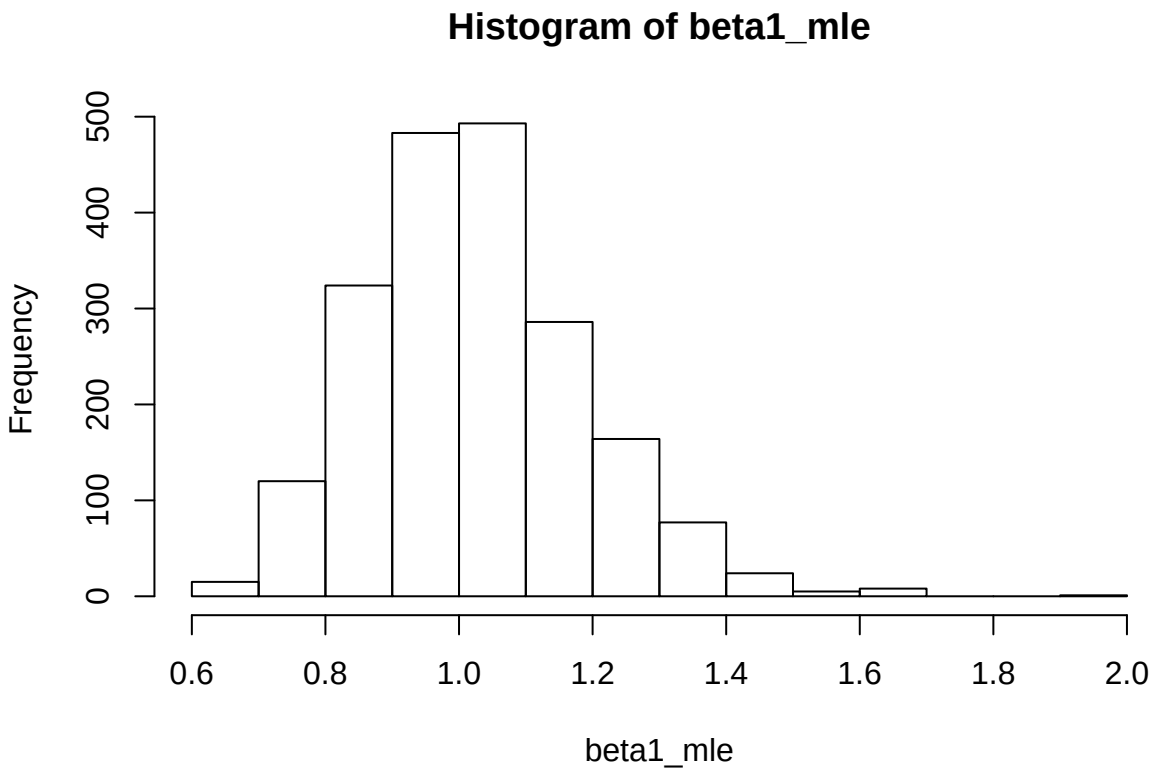
3.4.1

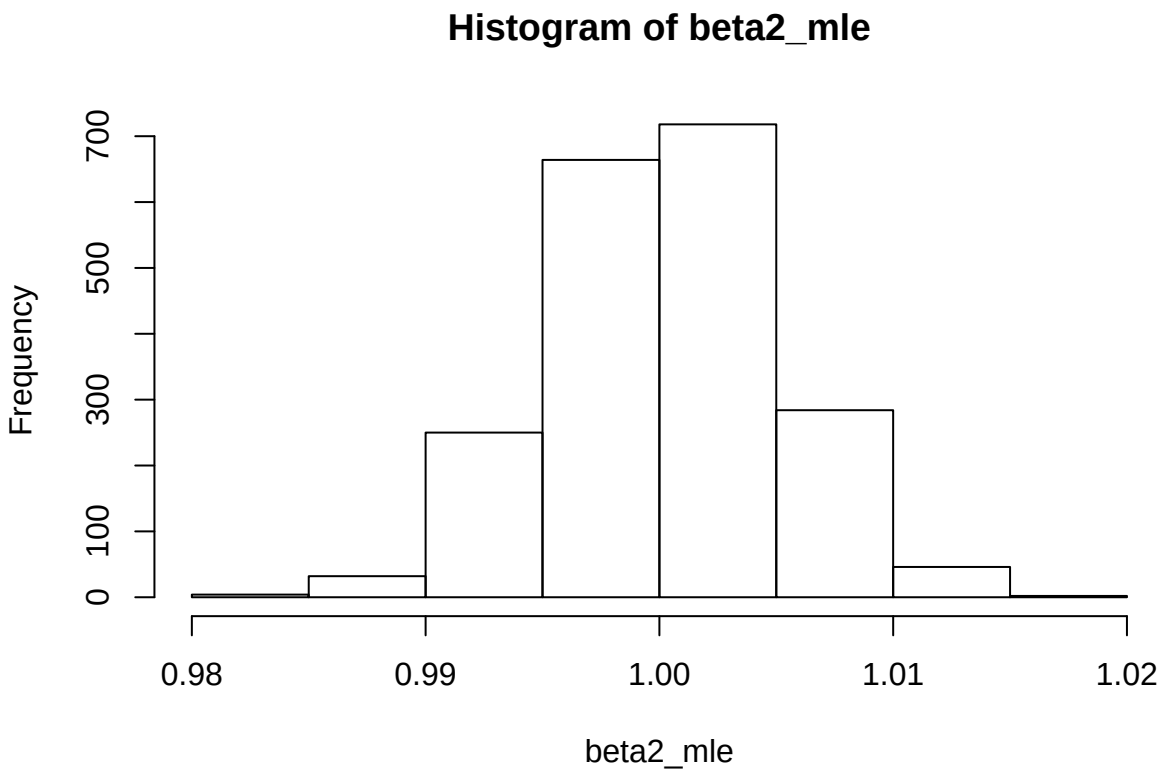
(1)

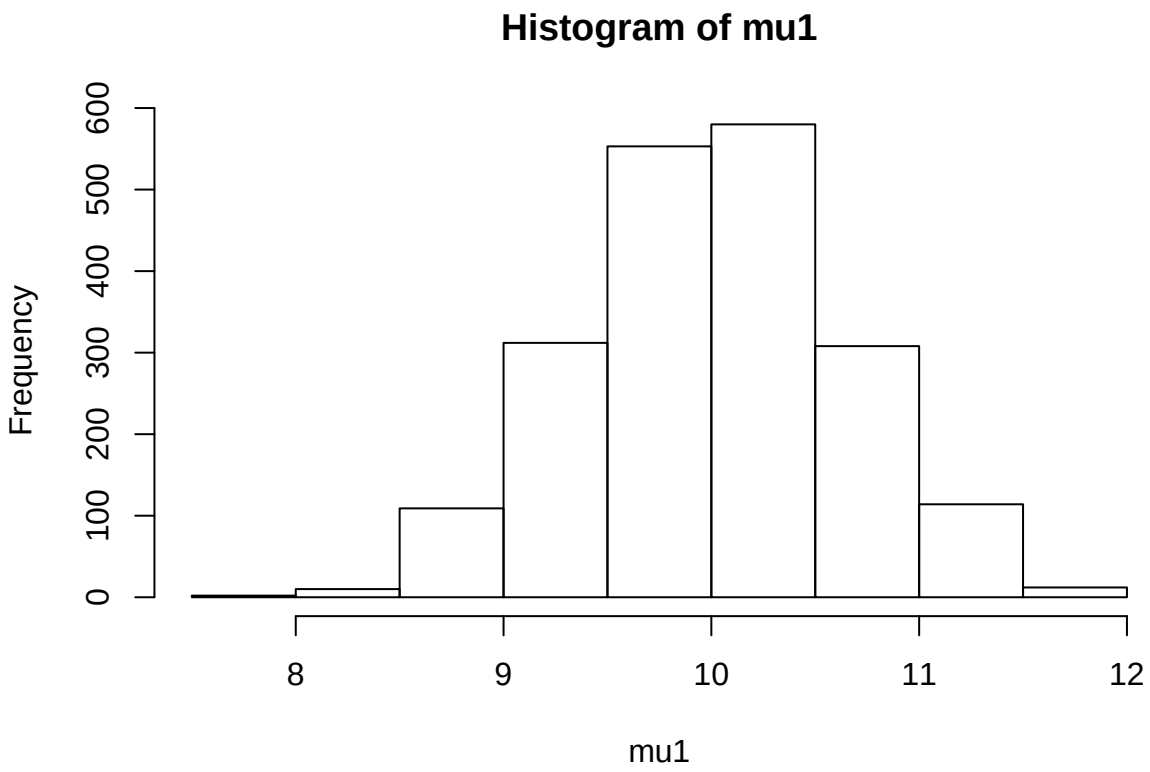
Visualisera samplingfördelningarna för gamma_beta_mle, norm_mu_mle och norm_sigma2_mle då n=10 och då n=10000 i ett histogram. Vad är dina slutsatser ?

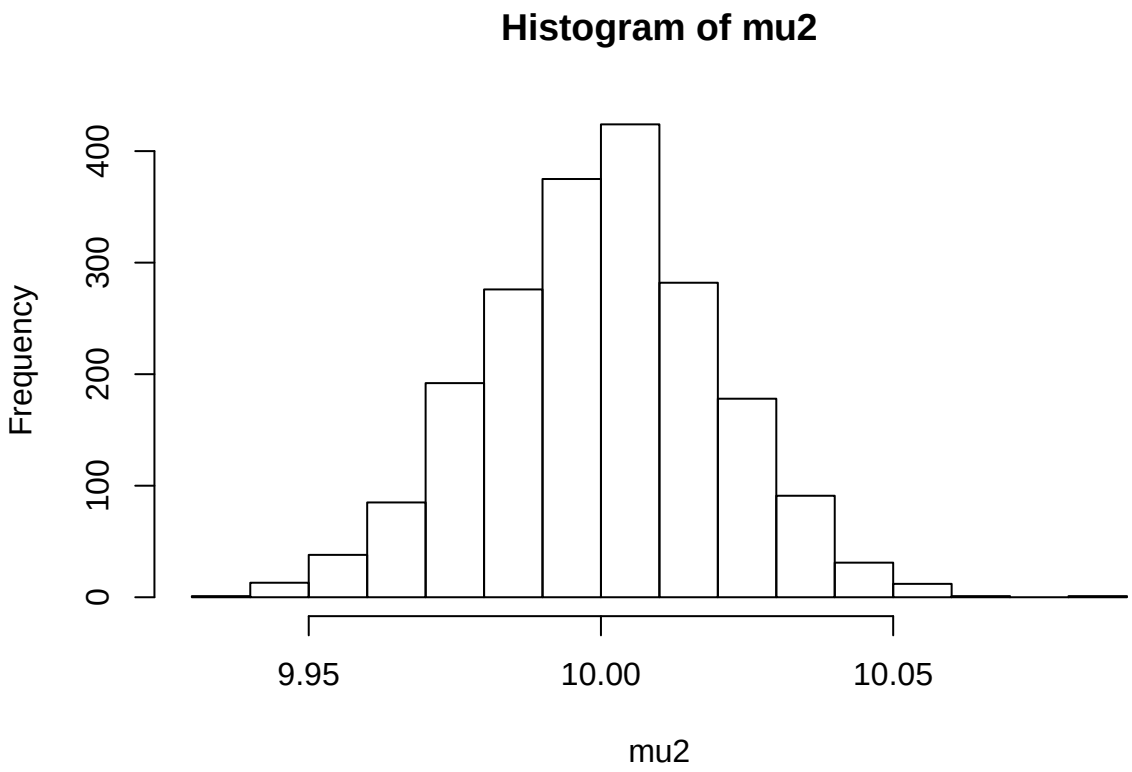
```
n <- 2000
beta1_mle <- numeric(n)
beta2_mle <- numeric(n)
mu1 <- numeric(n)
mu2 <- numeric(n)
```

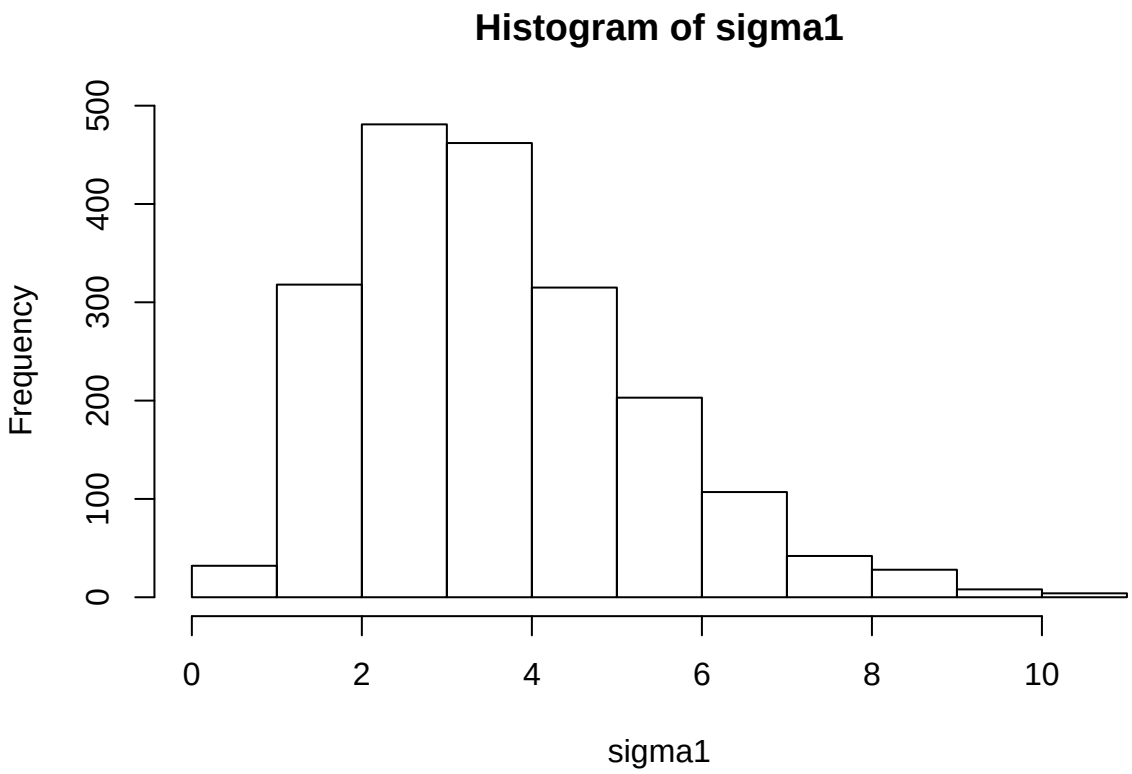
```r
sigma1 <- numeric(n)
sigma2 <- numeric(n)

for (i in 1:n) {
    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)
}
```

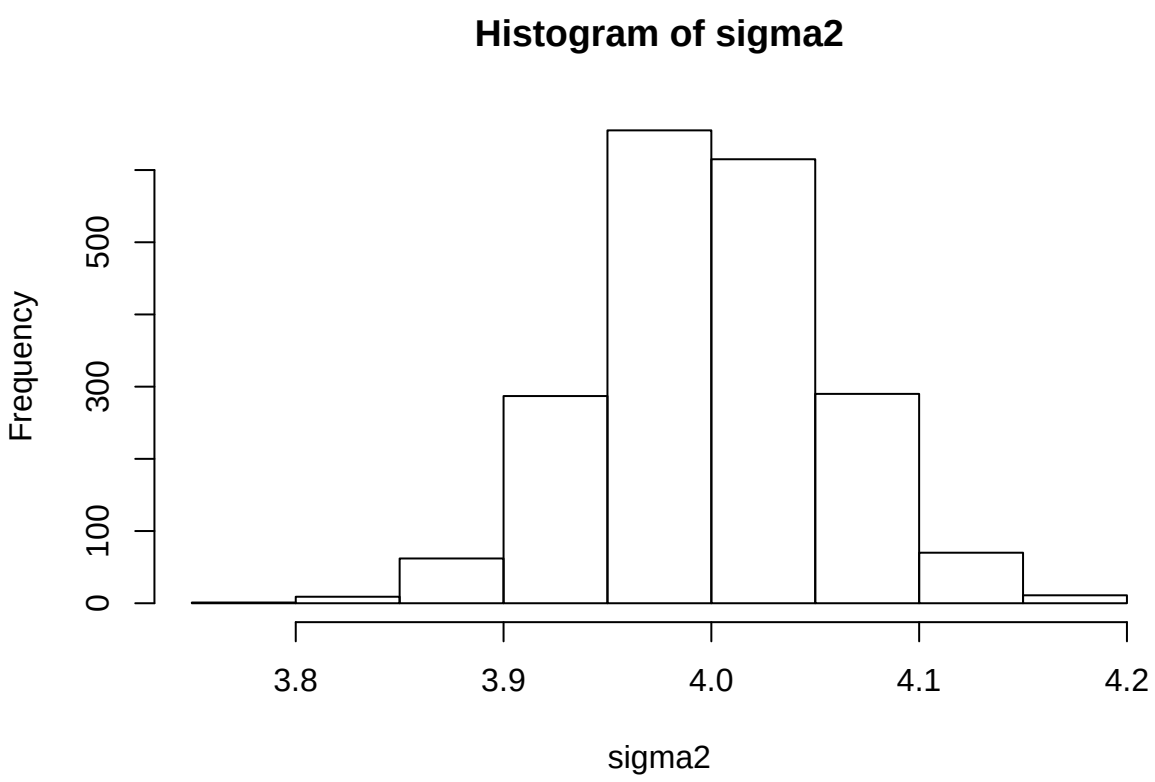



Histogram of beta2_mle
Frequency
0
100
300
500
700
0.98
0.99
1.00
1.01
1.02
beta2_mle


Histogram of mu1
Frequency
0
100
200
300
400
500
600
8
9
10
11
12
mu1


Histogram of mu2
Frequency
0
100
200
300
400
9.95
10.00
10.05
mu2


Histogram of sigma1
Frequency
0
100
200
300
400
500
0
2
4
6
8
10
sigma1



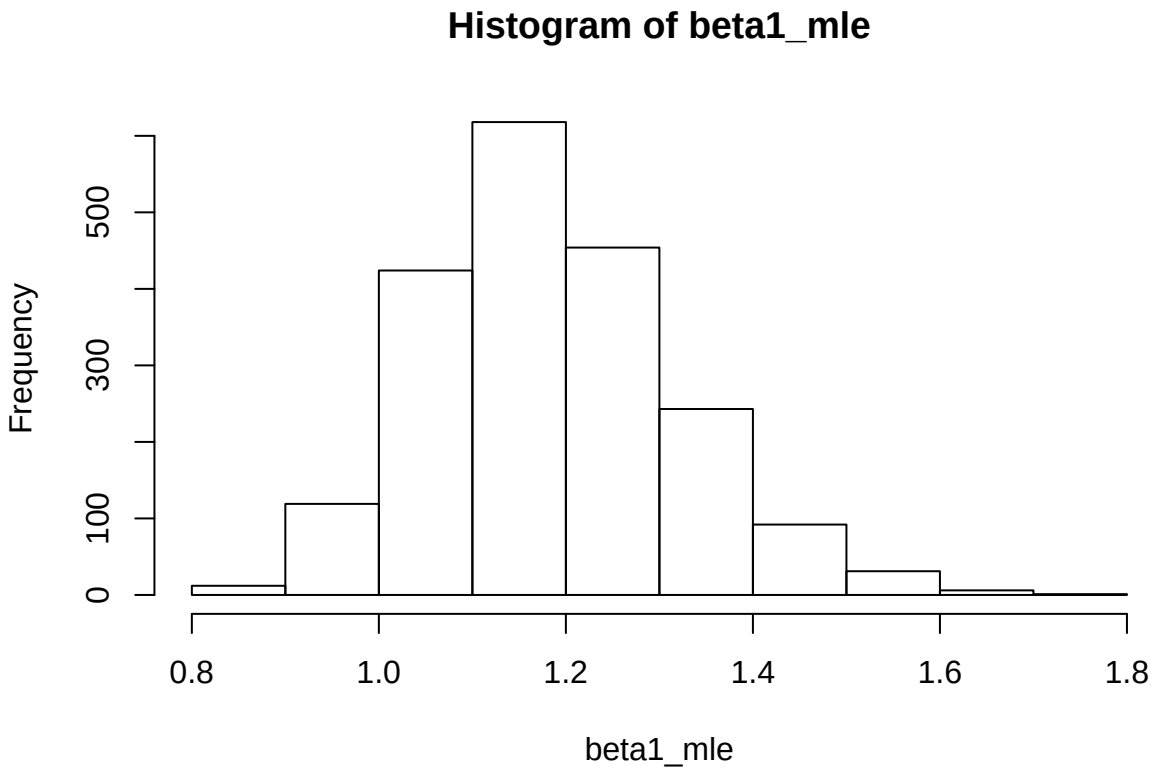
Våra slutsatser är att nära vi sätter n = 10000 så blir variansen mycket mindre än i fallet när vi har n = 10. sigma1 blir lite svår att tolka eftersom det är så få dragningar förhållande till sigma och alla "fel" kvadreras så dessa verkar större än de är.

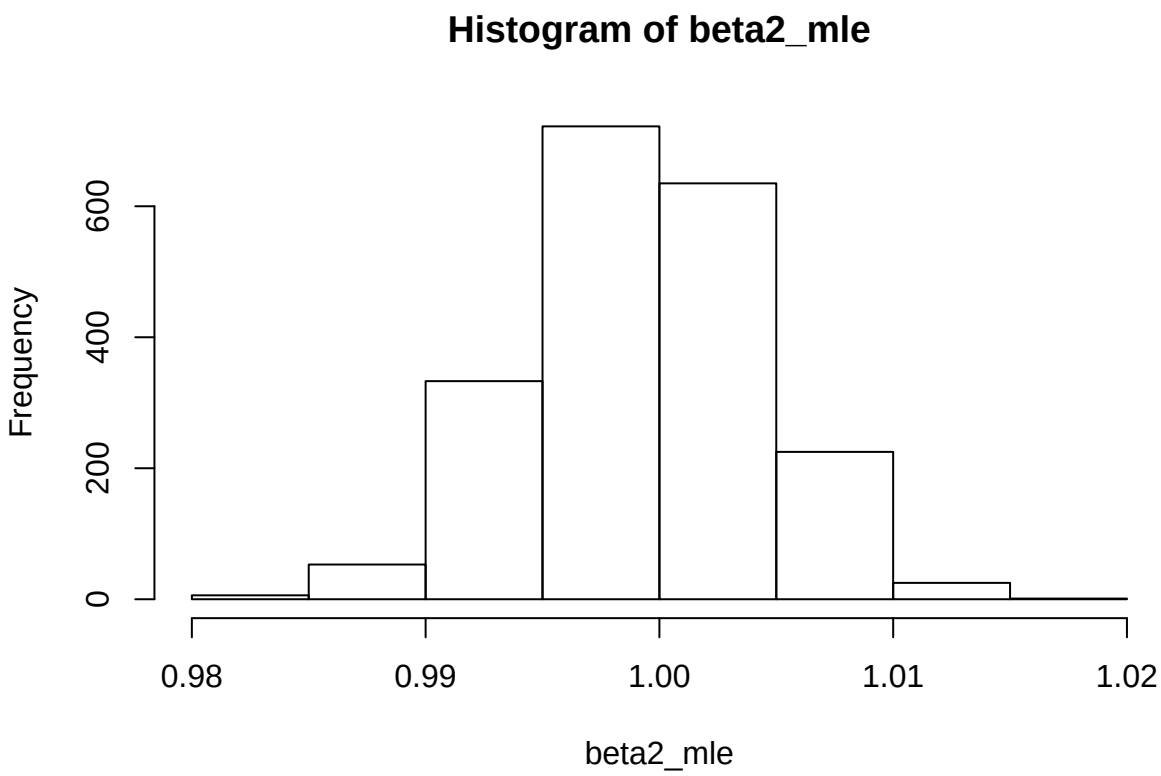
(2)

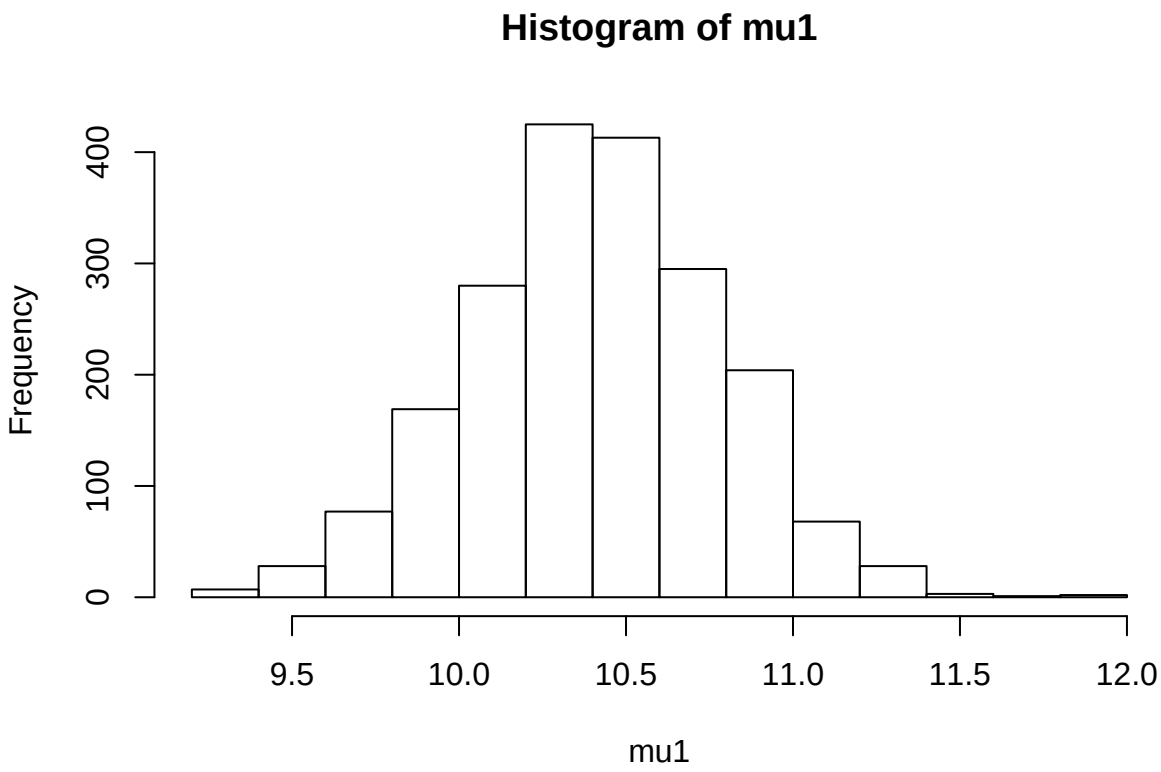
Visualisera boostrapfordelningarna för gamma_beta_mle, norm_mu_mle och norm_sigma2_mle då n=10 och då n=10000 i ett histogram. Jämfor dem med samplingfördelningarna i (1). Vad är dina slutsatser?

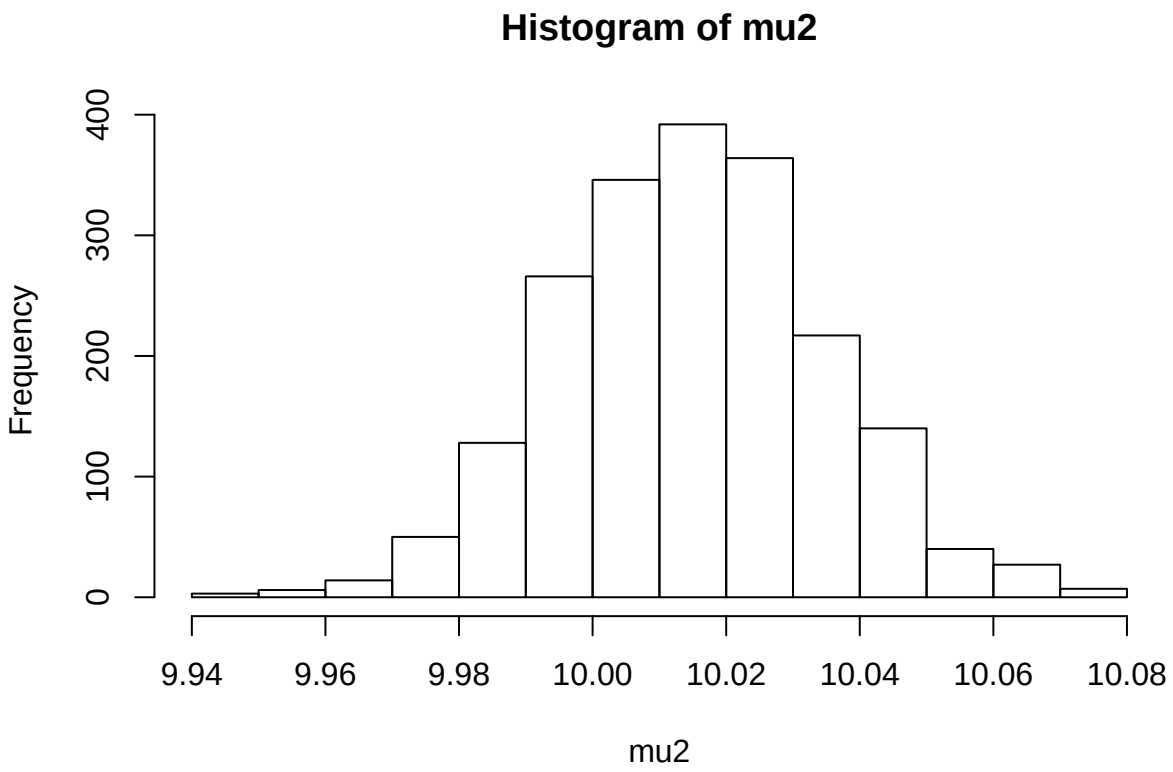
```
n <- 2000
beta1_mle <- numeric(n)
beta2_mle <- numeric(n)
mu1 <- numeric(n)
mu2 <- numeric(n)
sigma1 <- numeric(n)
sigma2 <- numeric(n)
x1 <- rgamma(n = 10, shape = 4, rate = 1)
x2 <- rgamma(n = 10000, shape = 4, rate = 1)
y1 <- rnorm(n = 10, mean = 10, sd = 2)
y2 <- rnorm(n = 10000, mean = 10, sd = 2)

for (i in 1:n) {
    beta1_mle[i] <- gamma_beta_mle(x = sample(x1, 10, replace = TRUE), alpha = 4)
    beta2_mle[i] <- gamma_beta_mle(x = sample(x2, 10000, replace = TRUE), alpha = 4)
    mu1[i] <- norm_mu_mle(x = sample(y1, 10, replace = TRUE))
    mu2[i] <- norm_mu_mle(x = sample(y2, 10000, replace = TRUE))
    sigma1[i] <- norm_sigma2_mle(x = sample(y1, 10, replace = TRUE))
```

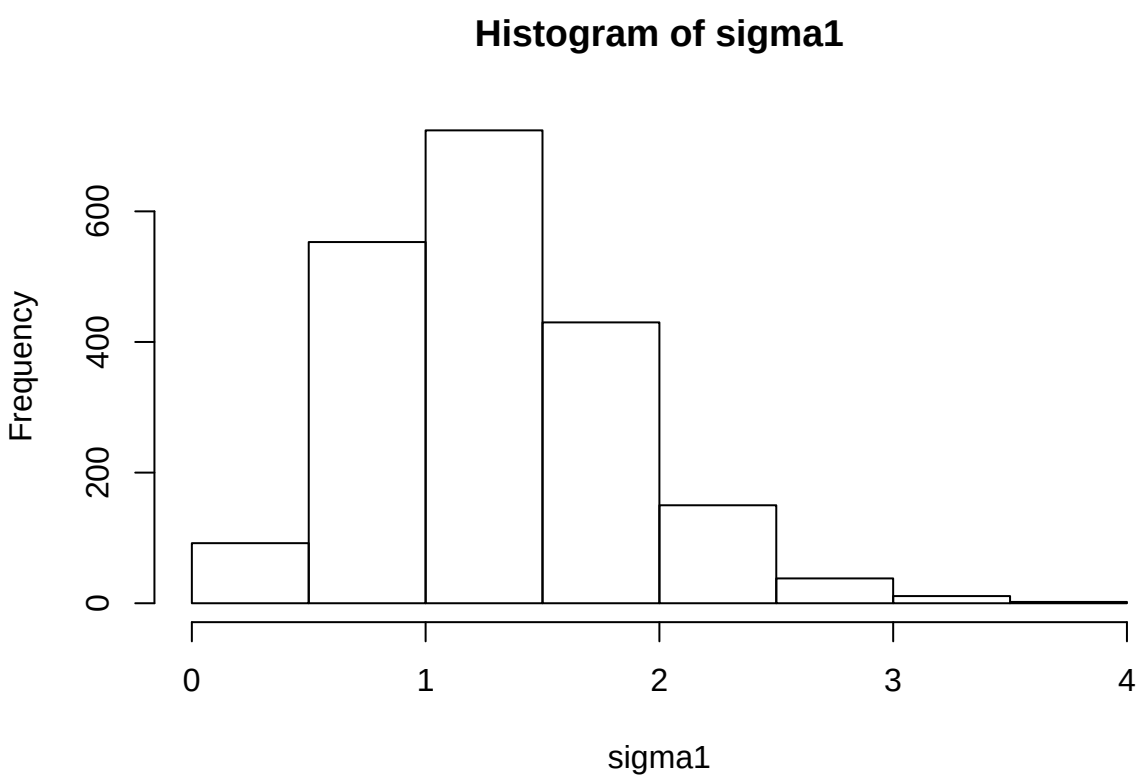
```
    sigma2[i] <- norm_sigma2_mle(x = sample(y2, 10000, replace = TRUE))
}
```

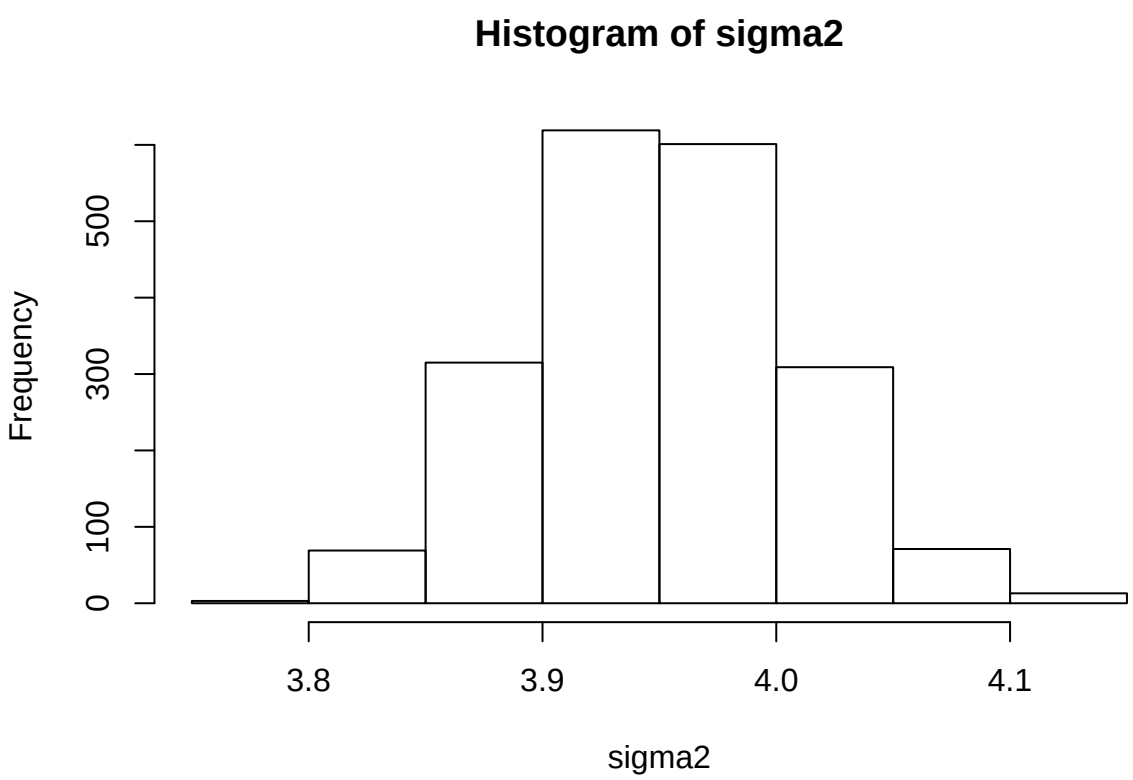



Histogram of beta2_mle
Frequency
0
200
400
600
0.98
0.99
1.00
1.01
1.02
beta2_mle


Histogram of mu1
Frequency
0
100
200
300
400
9.5
10.0
10.5
11.0
11.5
12.0
mu1


Histogram of mu2
Frequency
0
100
200
300
400
9.94
9.96
9.98
10.00
10.02
10.04
10.06
10.08
mu2


Histogram of sigma1
Frequency
0
200
400
600
0
1
2
3
4
sigma1



Våra slutsatser är att till skillnad från punkten ovan så blir uppskattningarna inte lika bra då vår indata är begränsad. Detta ger att "fel" i själva dragningarna har en större inverkan på de uppskattningar vi gör. Detta kan tydligt ses i de dragningar där n = 10 då de inte har väntevärde kring det faktiskta värdet utan lite vid sidan om.