

Lab 2

Mervan Palmér (merpa433)

2023-10-09

3.1.1

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

```
## [1] 7.732272 6.533384 6.095901 2.780186 1.356453 5.196592 1.927918 5.412191
## [9] 8.585872 6.436910
```

```
x2 <- rgamma(n = 100, shape = 4, scale = 1)
print(x2)
```

```
## [1] 4.7933145 6.4078777 1.2524996 7.0699216 2.6775015 1.7462794 5.6962765
## [8] 4.9533488 3.6483032 5.5642625 3.0268525 5.6341619 4.6954800 3.8741800
## [15] 4.7266055 4.9152438 8.2934848 6.5703531 7.1091655 4.3100000 6.8211806
## [22] 3.6603027 1.0355927 9.2742610 2.5662701 3.5025632 3.0801527 4.8644986
## [29] 4.3088394 6.7330634 2.7062688 4.5798475 2.6797855 2.2796759 2.0076845
## [36] 1.4096379 4.9922868 5.4800055 4.5163278 5.1399955 3.7729417 3.4574605
## [43] 2.8503213 2.4625271 3.3658973 3.4968829 2.4145311 5.0747220 3.8178490
## [50] 3.6556031 5.8728461 4.3116583 9.4085043 8.0308030 3.4812218 5.3774500
## [57] 7.7453305 8.5612052 0.7180357 4.3406654 3.0284858 5.7561584 1.4420738
## [64] 3.8452518 5.1088222 2.0748232 3.5825704 0.9476719 2.4432608 4.9785975
## [71] 3.5238302 1.6925612 9.0357576 3.6132114 2.4578954 2.9422182 4.9744189
## [78] 1.5697247 2.0107780 7.0152307 6.4054799 1.3905572 2.8664584 1.2083543
## [85] 1.5948936 4.0447307 7.8776784 2.7505398 5.8247179 3.3252117 2.5662169
## [92] 2.5599499 5.5872460 4.6300639 4.7439562 3.8493310 1.6461260 4.7146479
## [99] 2.2851556 3.0887078
```

Fråga 1

```
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
```

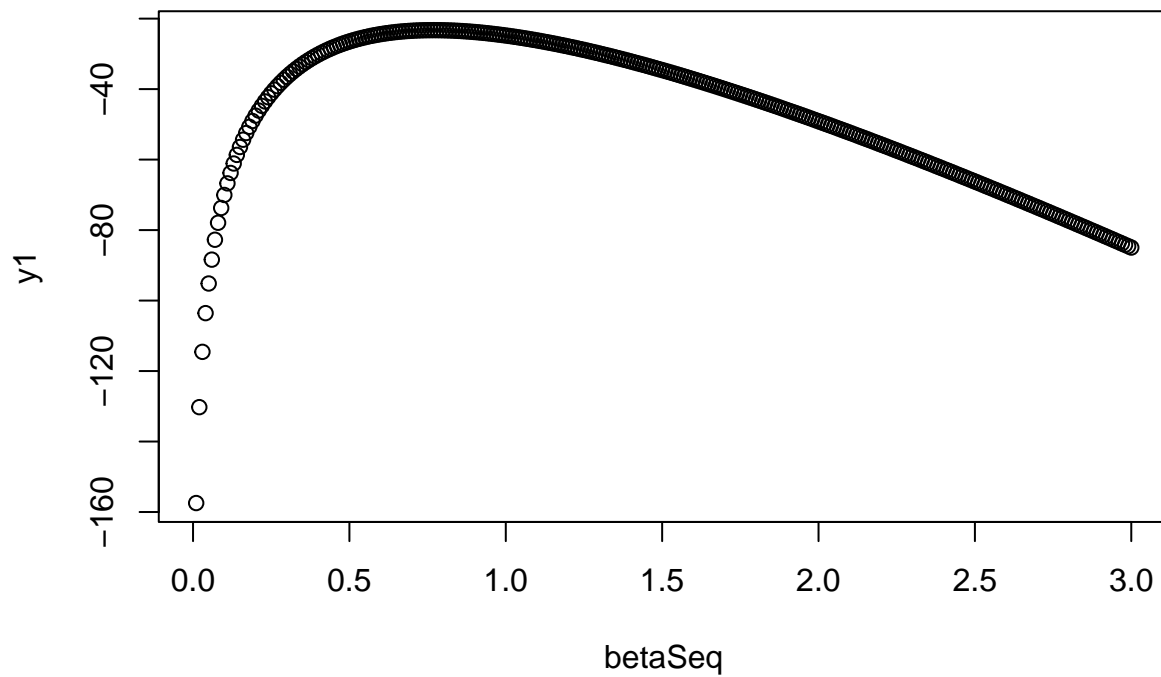
Fråga 2

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

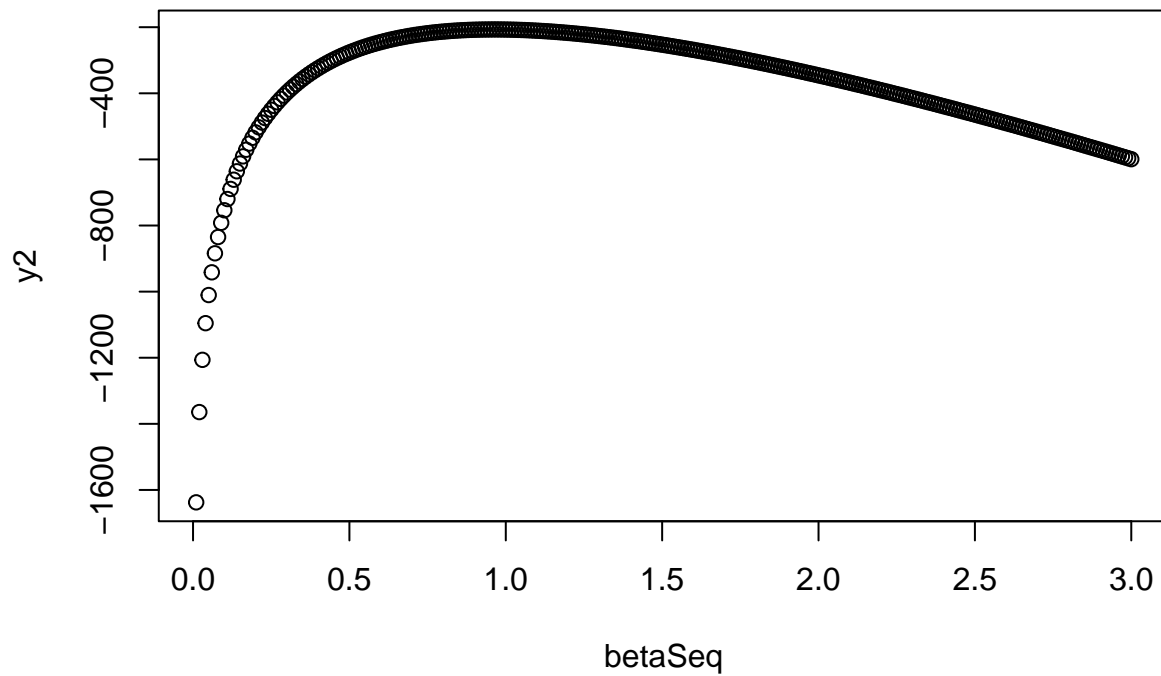
```

y2 <- numeric(0)
for (beta in betaSeq) {
  y2 <- c(y2, llgamma(x = x2, alpha = alfa, beta = beta))
}

```



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



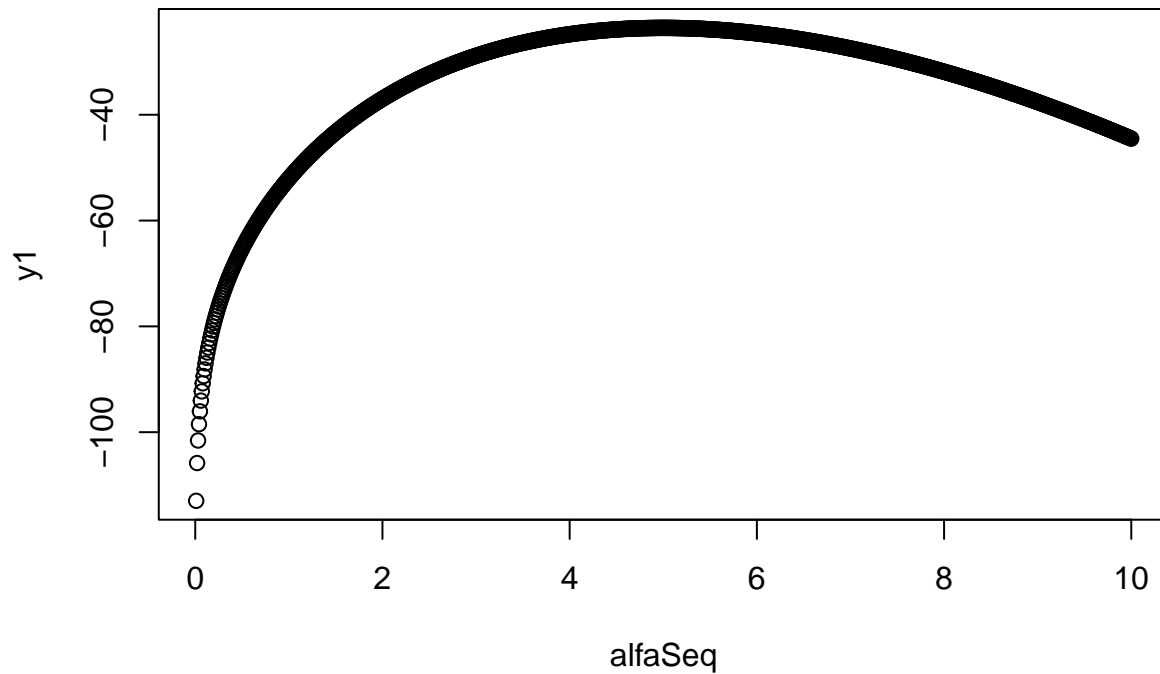
```
## [1] "max value at beta = 0.96"
```

y_1 har sitt max vid 0.77 y_2 har sitt max vid 0.96

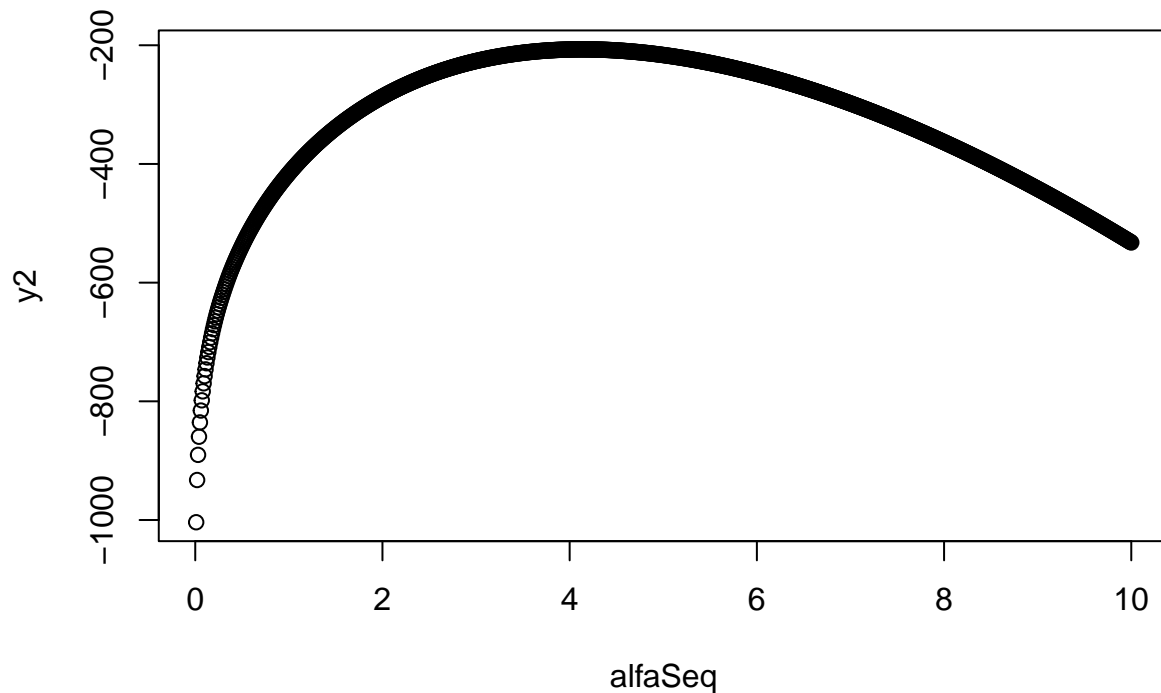
```

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

```



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



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

y1 har sitt max vid 5

y2 har sitt max vid 4.13

Fråga 4.

Täthetsfunktionen för 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 ihop med varandra:

$$\begin{aligned} \prod_{i=1}^n f(x_i) &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma^2}} * e^{\frac{-1}{2}(\frac{(x_i-\mu)^2}{\sigma^2})} = \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n * \prod_{i=1}^n e^{\frac{-1}{2}(\frac{(x_i-\mu)^2}{\sigma^2})} \end{aligned}$$

Detta är likelihood-funktionen, så vi tar log av denna för att få log-likelihoodfunktionen:

$$\begin{aligned} \ln\left(\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n * \prod_{i=1}^n e^{\frac{-1}{2}(\frac{(x_i-\mu)^2}{\sigma^2})}\right) &= \\ = \ln\left(\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n\right) + \sum_{i=1}^n \frac{-1}{2}(\frac{(x_i-\mu)^2}{\sigma^2}) &= \\ = \frac{-n}{2} * \ln(2\pi\sigma^2) + \frac{-1}{2} * \frac{1}{\sigma^2} * \sum_{i=1}^n (x_i - \mu)^2 &= \end{aligned}$$

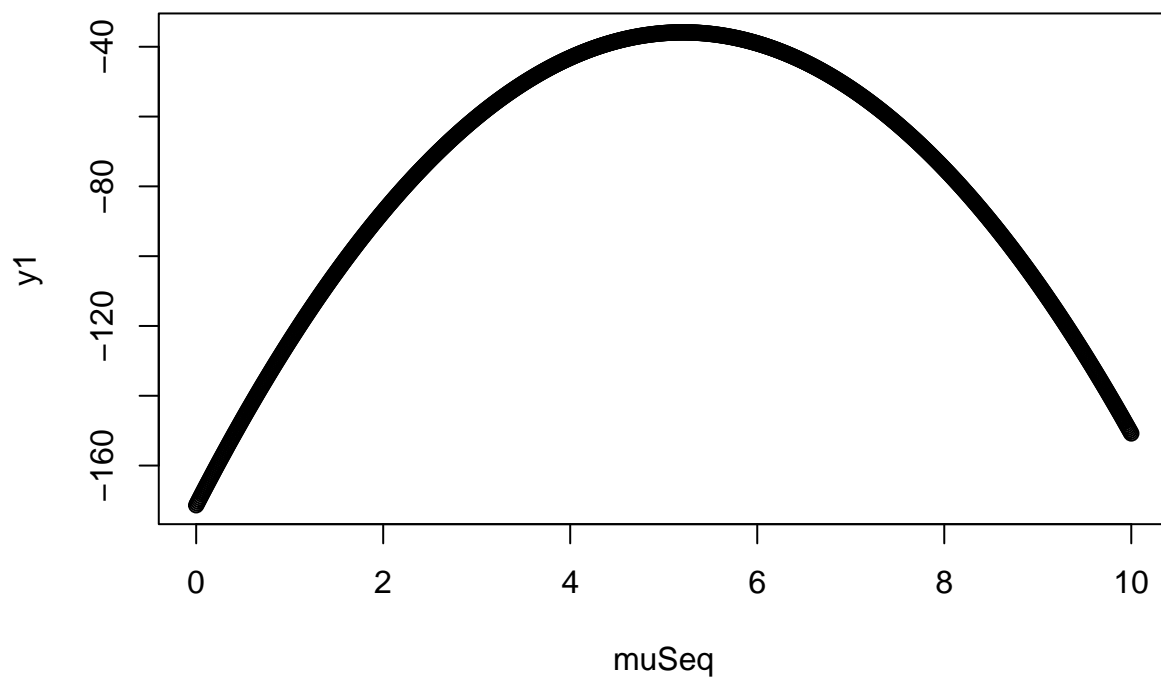
$$= \frac{-n}{2} * \ln(2\pi\sigma^2) + \frac{-1}{2\sigma^2} * \sum_{i=1}^n (x_i - \mu)^2$$

```
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))
```

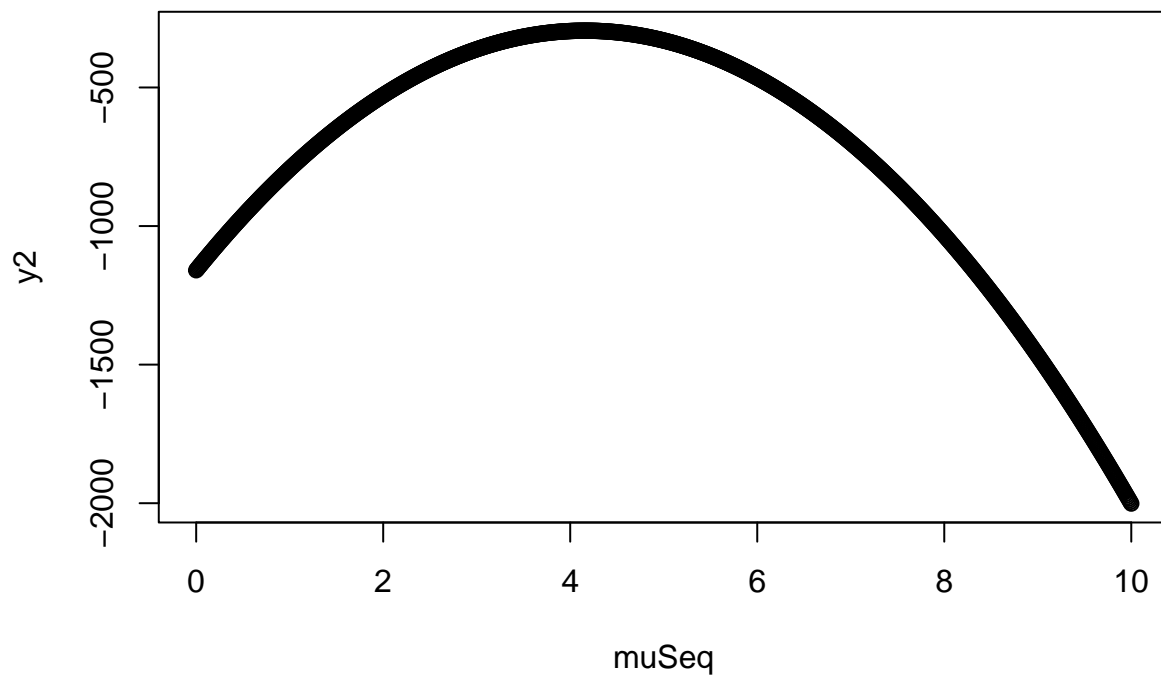
```
## [1] -87.25743
```

Fråga 5.

```
sigma2 <- 1
y1 <- numeric(0)
y2 <- numeric(0)
muSeq <- seq(0, 10, 0.01)
for (mu in muSeq) {
  y1 <- c(y1, llnorm(x = x1, mu = mu, sigma2 = sigma2))
}
for (mu in muSeq) {
  y2 <- c(y2, llnorm(x = x2, mu = mu, sigma2 = sigma2))
}
```



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

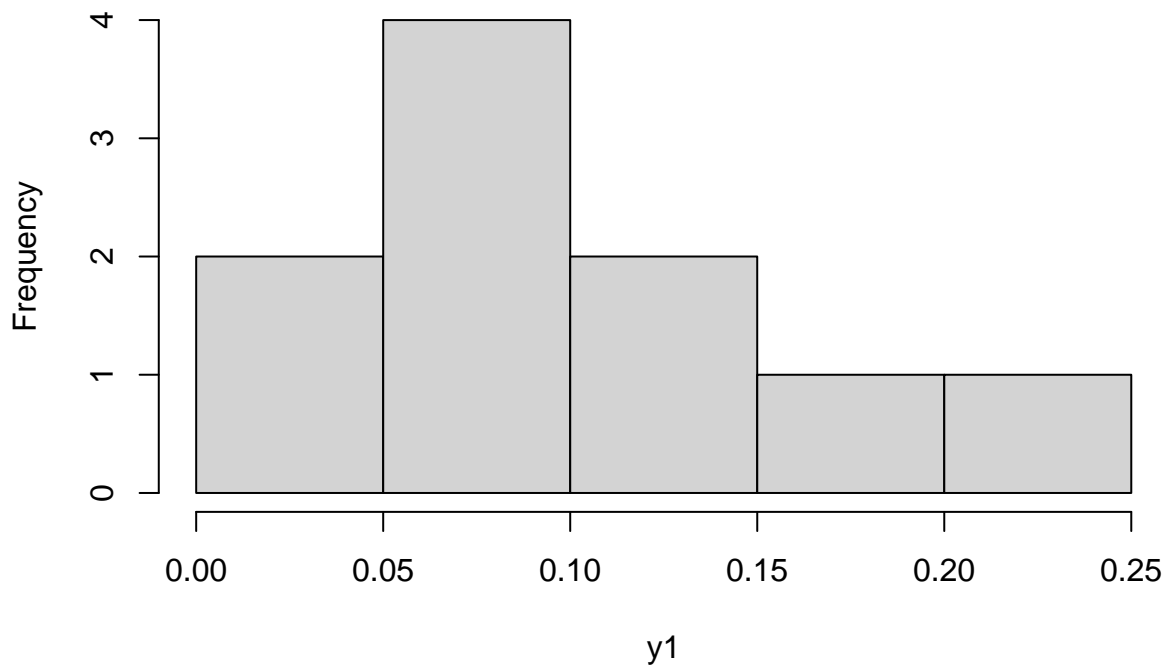


```
## [1] "max value at alpha = 4.16"
```

```
y1 <- dgamma(x1, shape = alfaMaxX1, scale = betaMaxX1)
y2 <- dgamma(x2, shape = alfaMaxX2, scale = betaMaxX2)
y3 <- dnorm(x1, mean = muMaxX1, sd = 1)
y4 <- dnorm(x2, mean = muMaxX2, sd = 1)

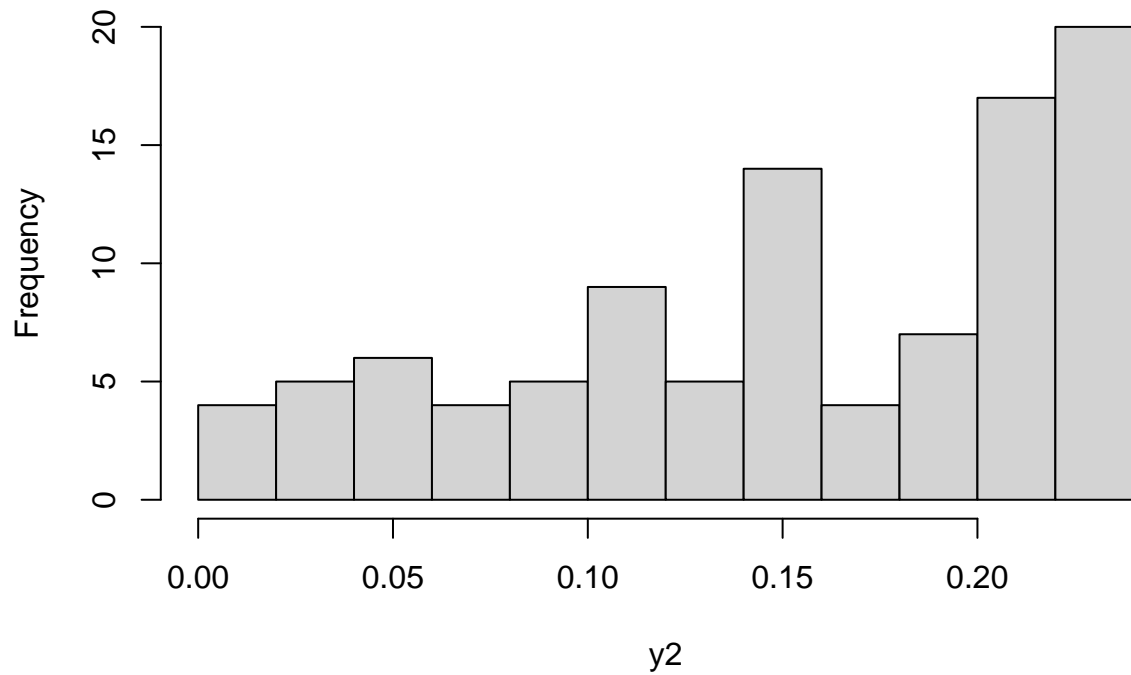
hist(y1)
```

Histogram of y1



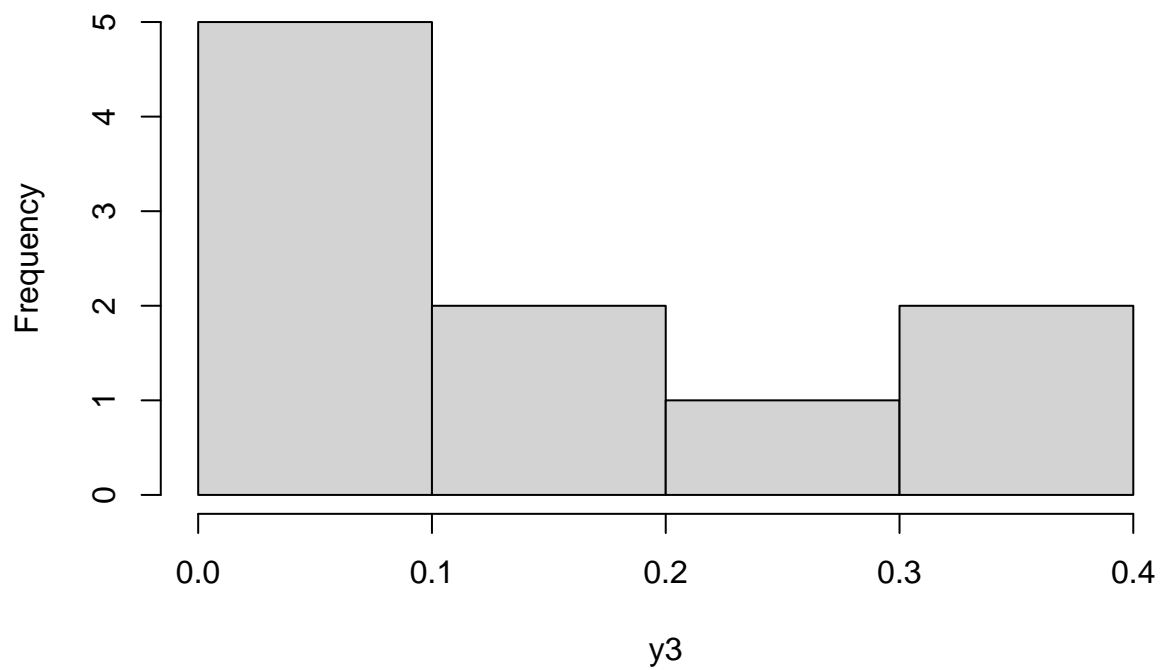
```
hist(y2)
```

Histogram of y2



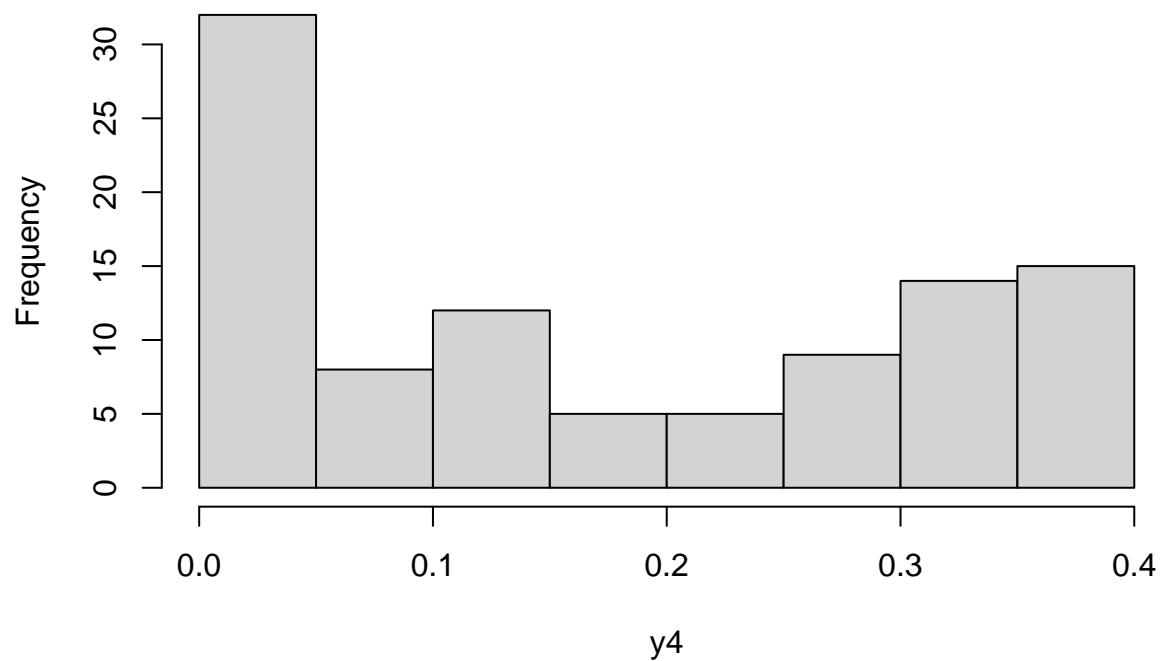
```
hist(y3)
```

Histogram of y3



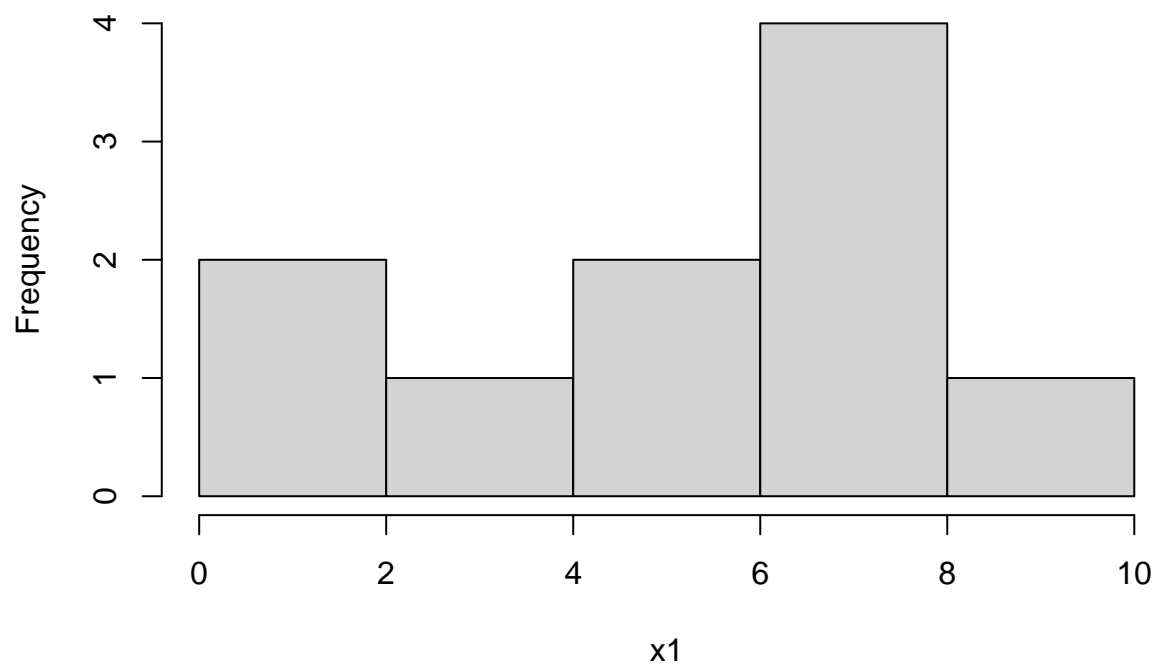
```
hist(y4)
```

Histogram of y4

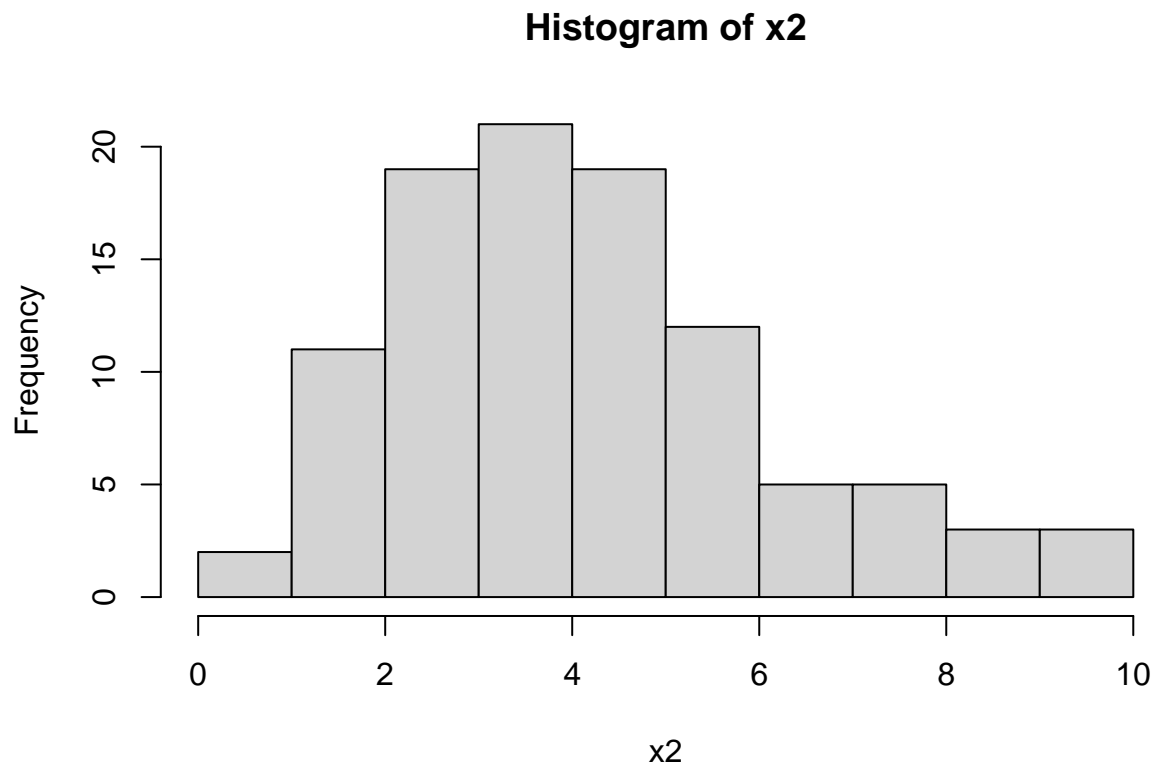


```
hist(x1)
```

Histogram of x1




```
hist(x2)
```



Y1 och Y2 har fler värden kring högre sannolikhetsgrader vilket får mig att tro att det är en gammafördelning
Y3 och Y4 har en stor del av värdena kring 0 vilket visar att en normalfördelning inte skapade x1 och x2

3.2.1

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

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

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

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

För x1 så maximeras sannolikheten för dessa värden då $\beta = 0.7683785$ när α är 4

För x2 så maximeras sannolikheten för dessa värden då $\beta = 0.9619473$ när α är 4

3.2.2

Fråga 1

```
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
```

Fråga 2

```

set.seed(42)

y10 <- rnorm(n = 10, mean = 10, sd = 2) # Roten ur 4 är 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
```

Den stora skillnaden är att den med 10 000 dragningar kommer värdena närmare $\mu = 10$ och $\sigma^2 = 4$ vilket är det vi skickar in i `rnorm` funktionen. Baserat på detta skulle vi förmodligen få väldigt nära $\mu = 10$ och $\sigma^2 = 4$ om vi går mot oändligheten

3.3.1

Fråga 1

```

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

```

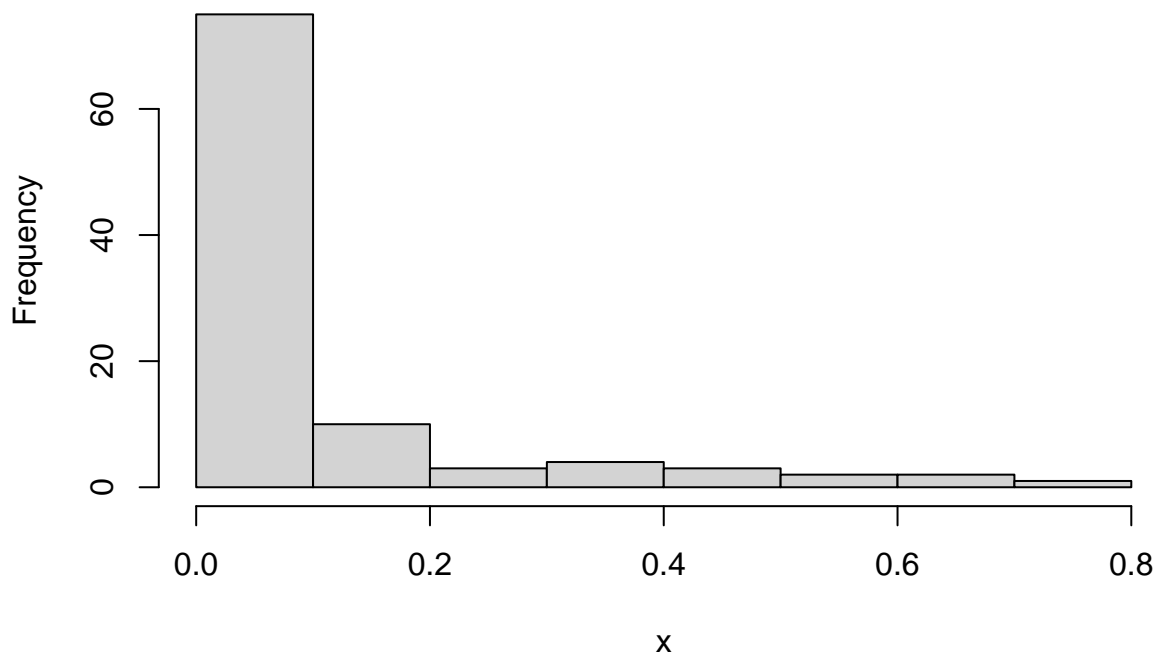
Fråga 2

```

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

```

Histogram of x



Fråga 3 Method “L-BFGS-B” is that of Byrd et. al. (1995) which allows box constraints, that is each variable can be given a lower and/or upper bound. The initial value must satisfy the constraints. This uses a limited-memory modification of the BFGS quasi-Newton method. If non-trivial bounds are supplied, this method will be selected, with a warning.

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

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

Fråga 4

För att vara en uppskattnings funktion blir de väldigt nära då alfa är 0,2 men uppskattnings ger oss 0,2211375 och att vi skickar in beta 2 men får 2.1439056. Men det är svårt med tanke på att det är just en uppskattning som görs att få exakt, lite som minsta kvadratmetoden i linjär algebra

3.4.1

Fråga 1.

```
n <- 2000
betaMle1 <- numeric(n)
betaMle2 <- numeric(n)
my1 <- numeric(n)
my2 <- numeric(n)
sigma1 <- numeric(n)
sigma2 <- numeric(n)

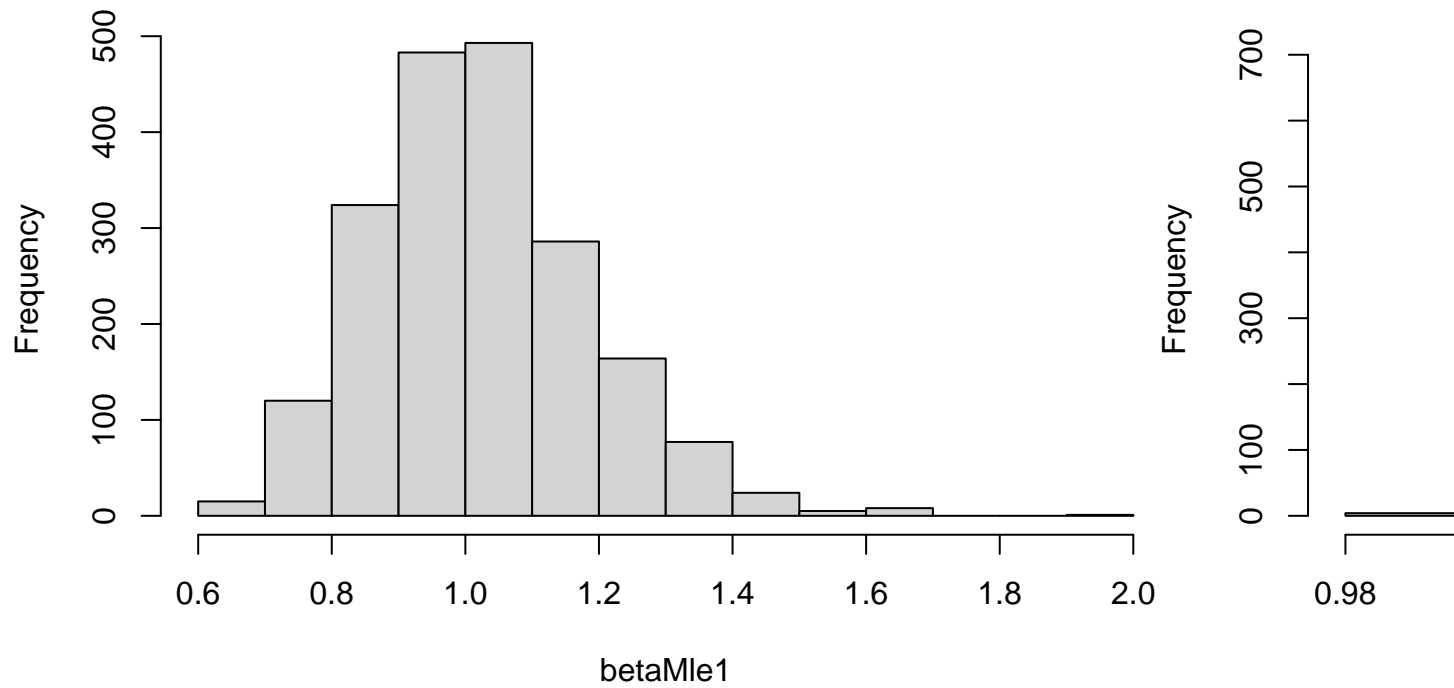
for (i in 1:n) {
  x1 <- rgamma(n = 10, shape = 4, rate = 1)
  x2 <- rgamma(n = 10000, shape = 4, rate = 1)
  betaMle1[i] <- gammaBetaMle(x = x1, alpha = 4)
  betaMle2[i] <- gammaBetaMle(x = x2, alpha = 4)
```

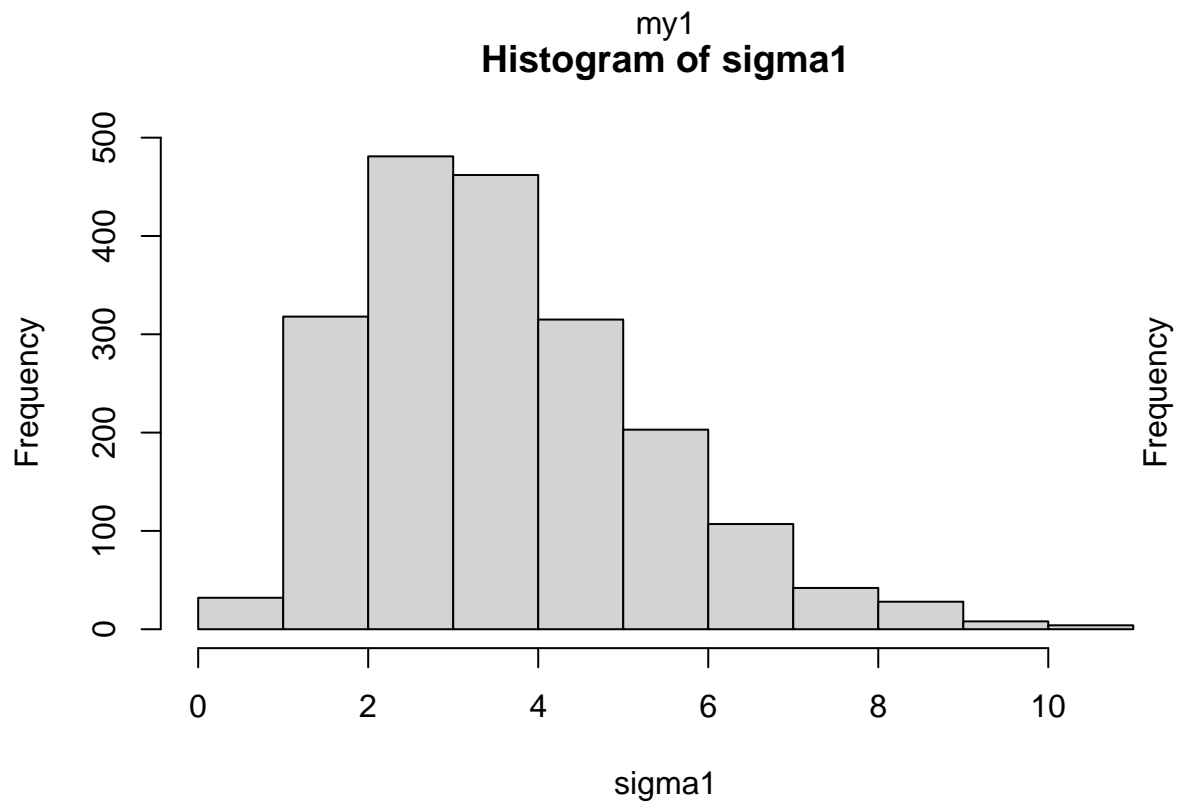
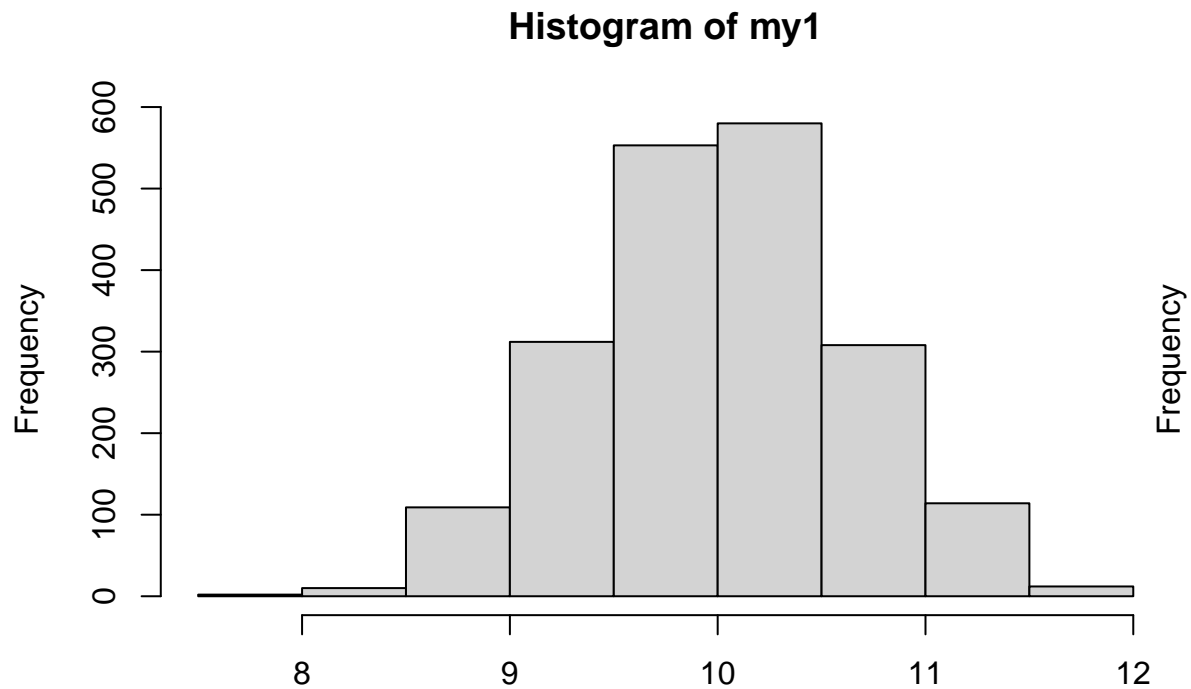
```

y1 <- rnorm(n = 10, mean = 10, sd = 2)
y2 <- rnorm(n = 10000, mean = 10, sd = 2)
my1[i] <- norm_mu_mle(x = y1)
my2[i] <- norm_mu_mle(x = y2)
sigma1[i] <- norm_sigma2_mle(x = y1)
sigma2[i] <- norm_sigma2_mle(x = y2)
}

```

Histogram of betaMle1





Slutsaten är att när vi tar $n=10\,000$ så blir variansen mycket mindre än när vi har $n = 10$. Det är sigma1 är svårtolkad då den har få dragningar jämfört med sigma

Fråga 2

```
n <- 2000
beta1_mle <- numeric(n)
```

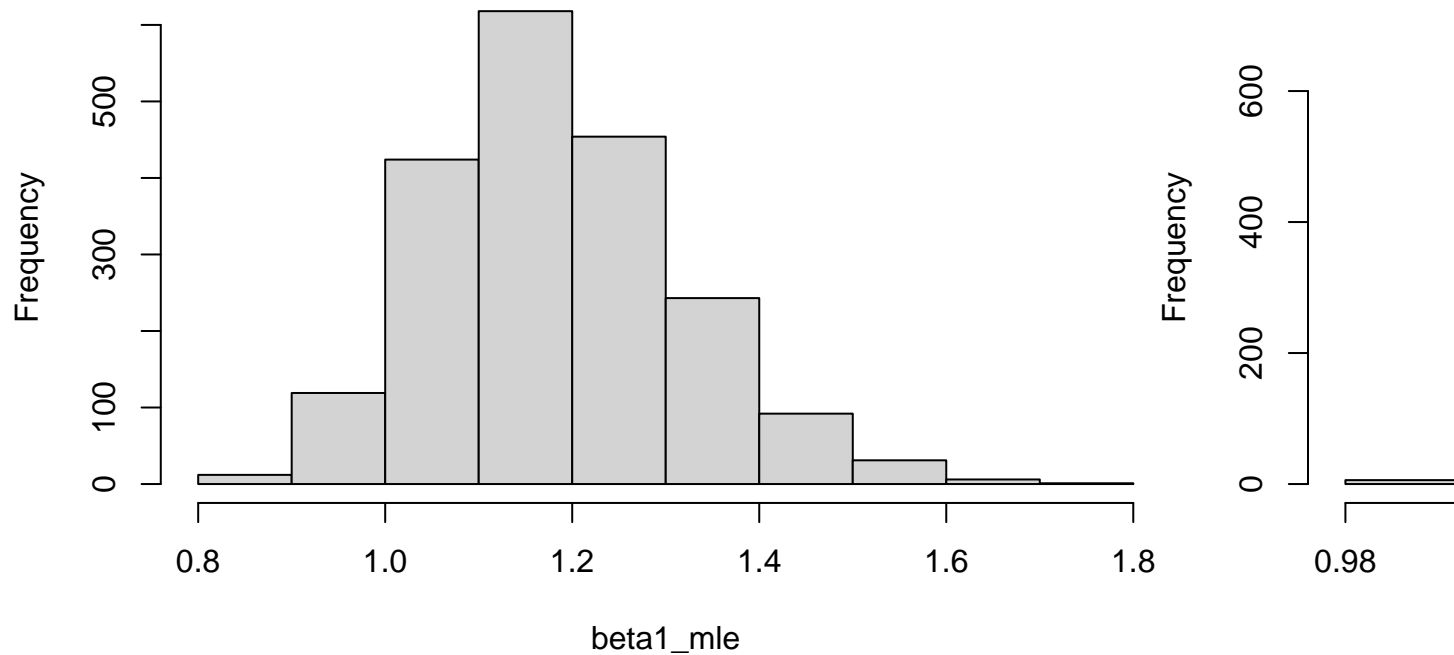
```

beta2_mle <- numeric(n)
my1 <- numeric(n)
my2 <- 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)

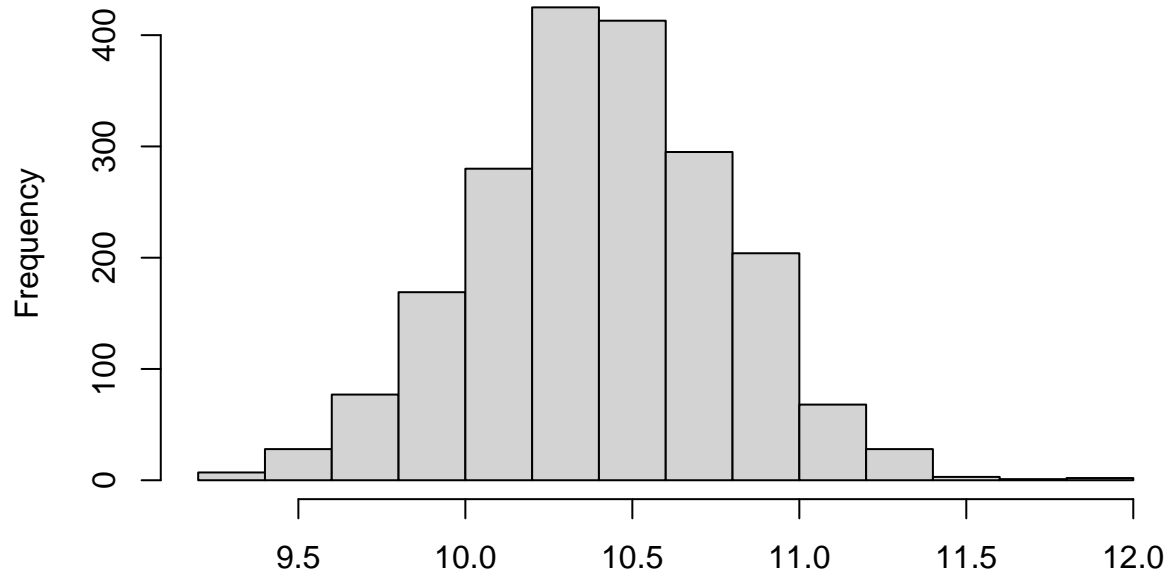
for (i in 1:n) {
  beta1_mle[i] <- gammaBetaMle(x = sample(x1, 10, replace = TRUE), alpha = 4)
  beta2_mle[i] <- gammaBetaMle(x = sample(x2, 10000, replace = TRUE), alpha = 4)
  my1[i] <- norm_mu_mle(x = sample(y1, 10, replace = TRUE))
  my2[i] <- norm_mu_mle(x = sample(y2, 10000, replace = TRUE))
  sigma1[i] <- norm_sigma2_mle(x = sample(y1, 10, replace = TRUE))
  sigma2[i] <- norm_sigma2_mle(x = sample(y2, 10000, replace = TRUE))
}

```

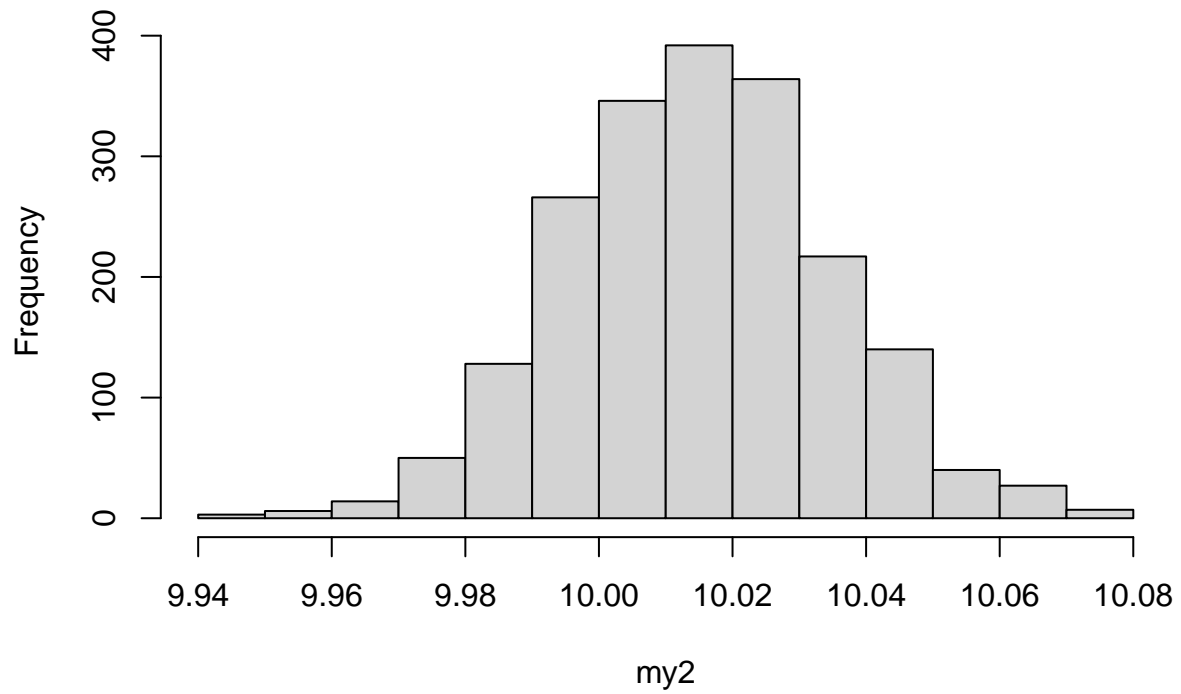
Histogram of beta1_mle

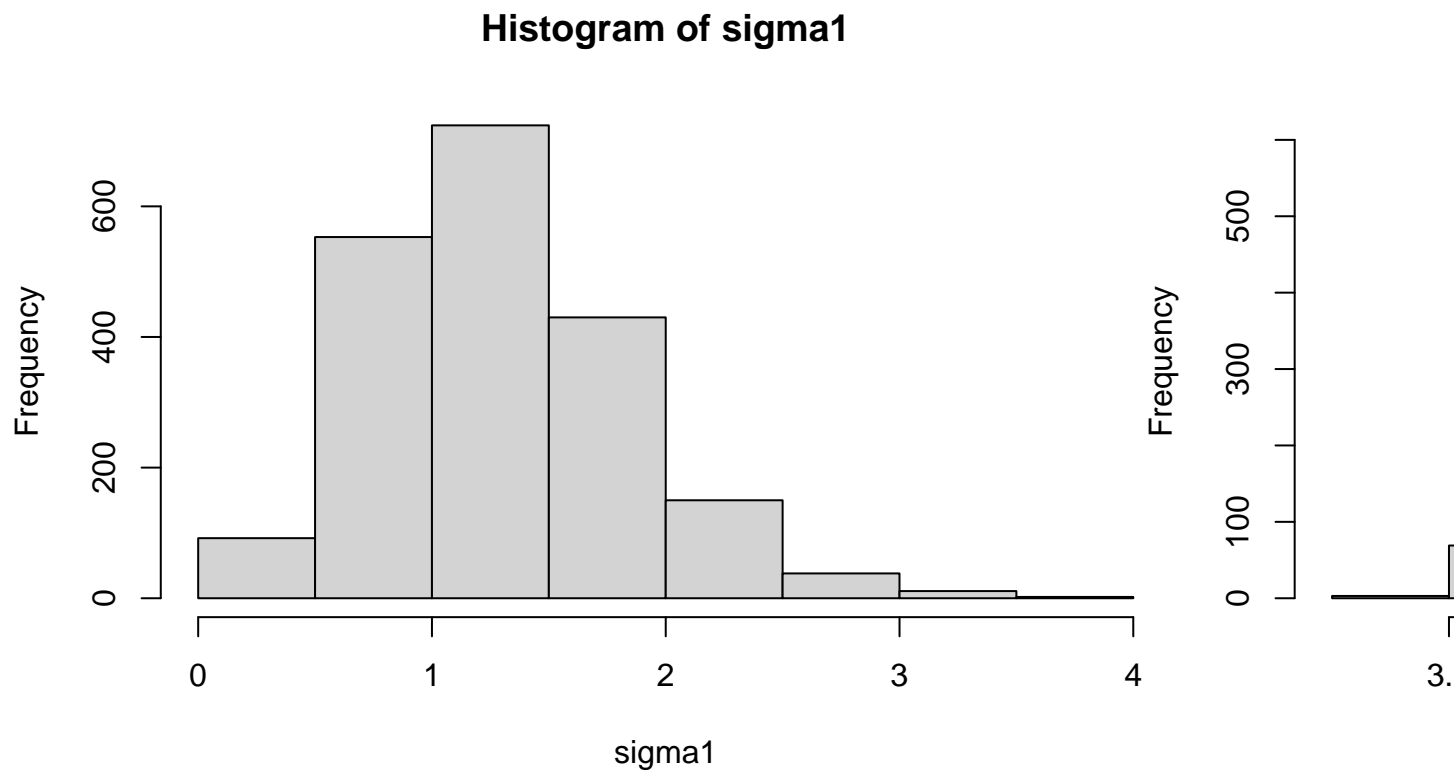


Histogram of my1



my1
Histogram of my2





Till skillnad från ovan blir dessa uppskattningar inte lika bra då vi har en mycket mer begränsad indata. Det ger att fel i dragningarna får en större inverkan på uppskattningarna som görs. Det syns tydligt på dragningar där $n = 10$.