Lab 1

Martin Gustafsson (margu424) and Axel Gard (axega544)

2020-09-29

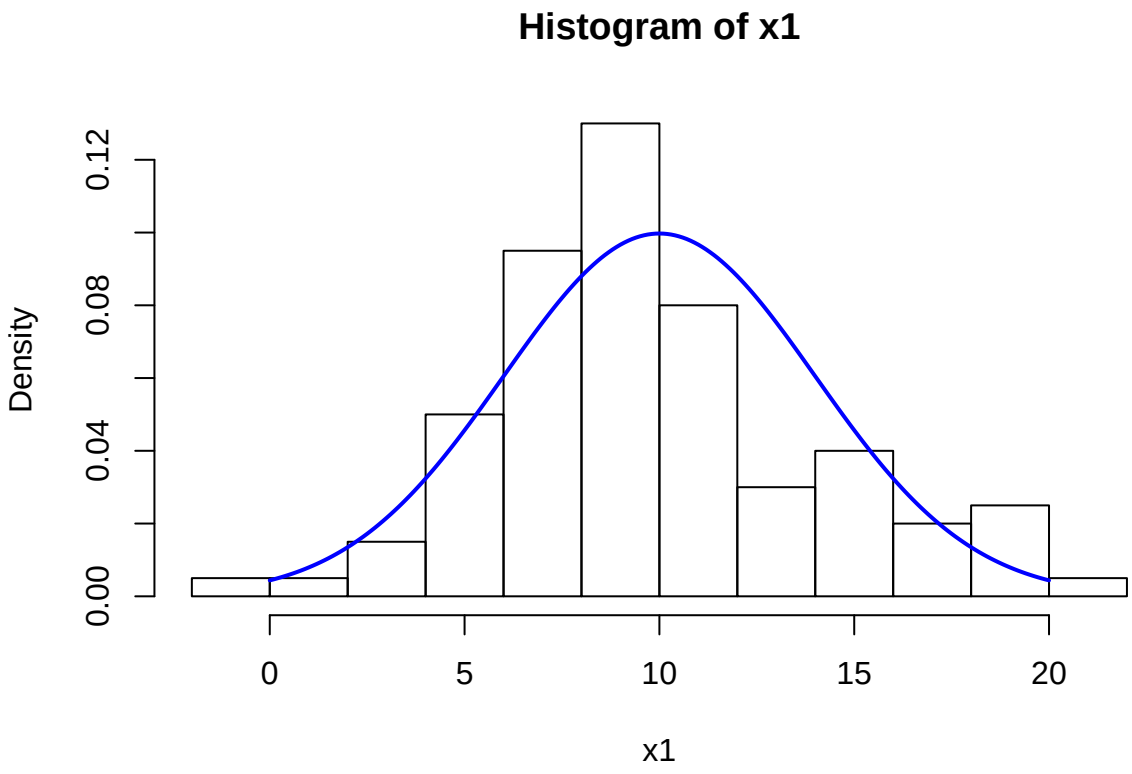
Uppgift 1 Simulering av normalfördelning

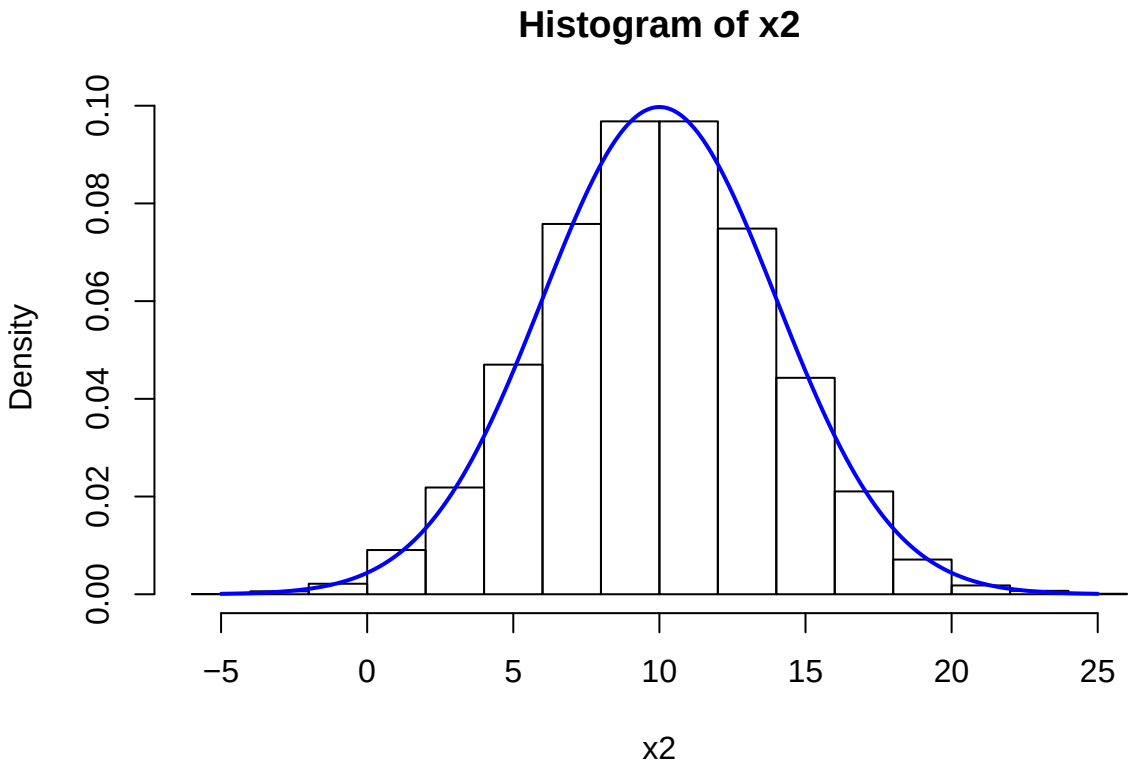
a) Visualisera fördelningarna i två histogram. Visualisera fördelningens pdf i samma graf.

Nedan simuleras normalfördelningen med olika antalet dragningar.

```
x1 <- rnorm(100, mean = 10, sd = 4)
x2 <- rnorm(10000, mean = 10, sd = 4)
```

I figurerna nedan visas resultatet av dragningarna som ett histogram tillsammans med täthetsfunktionen.



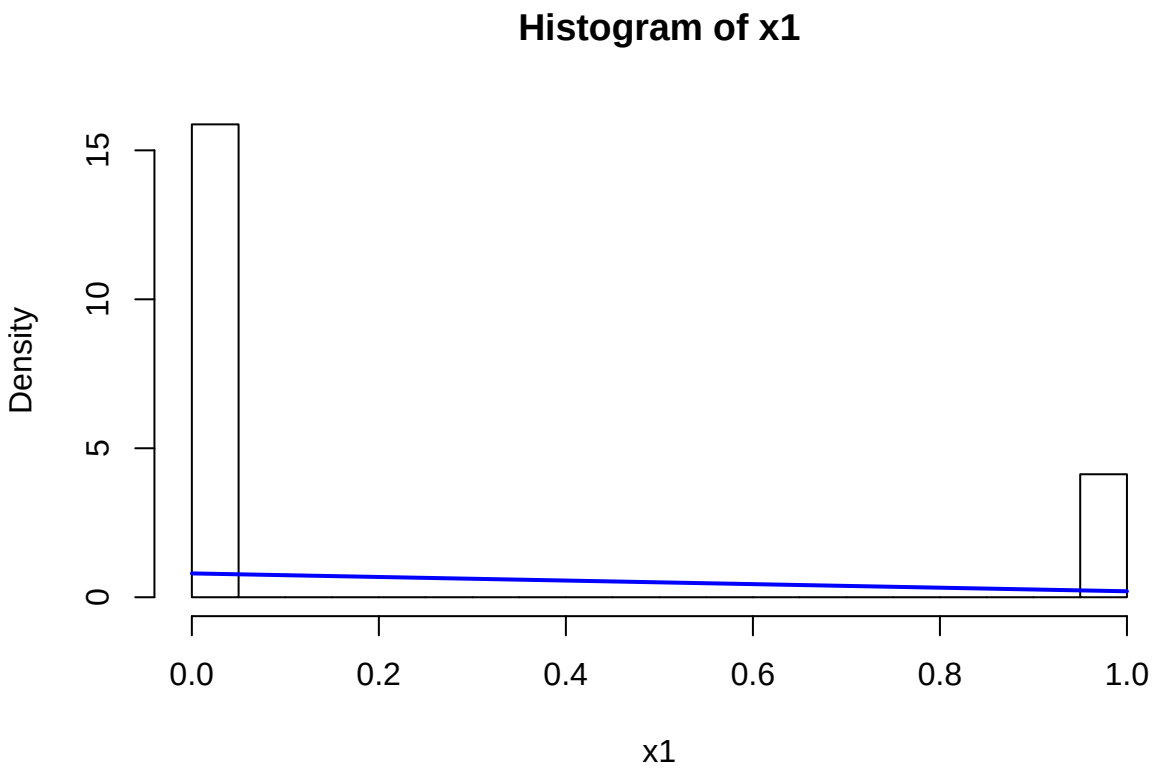


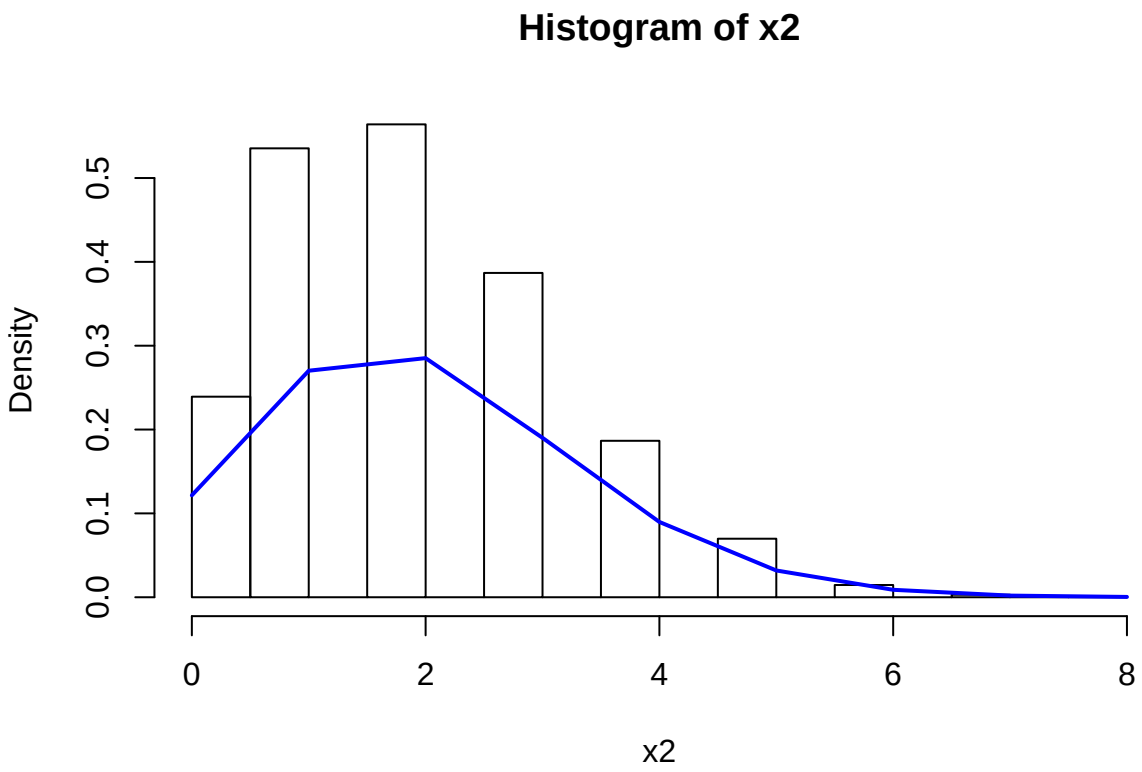
b) Beskriv skillnaden mellan de olika graferna.

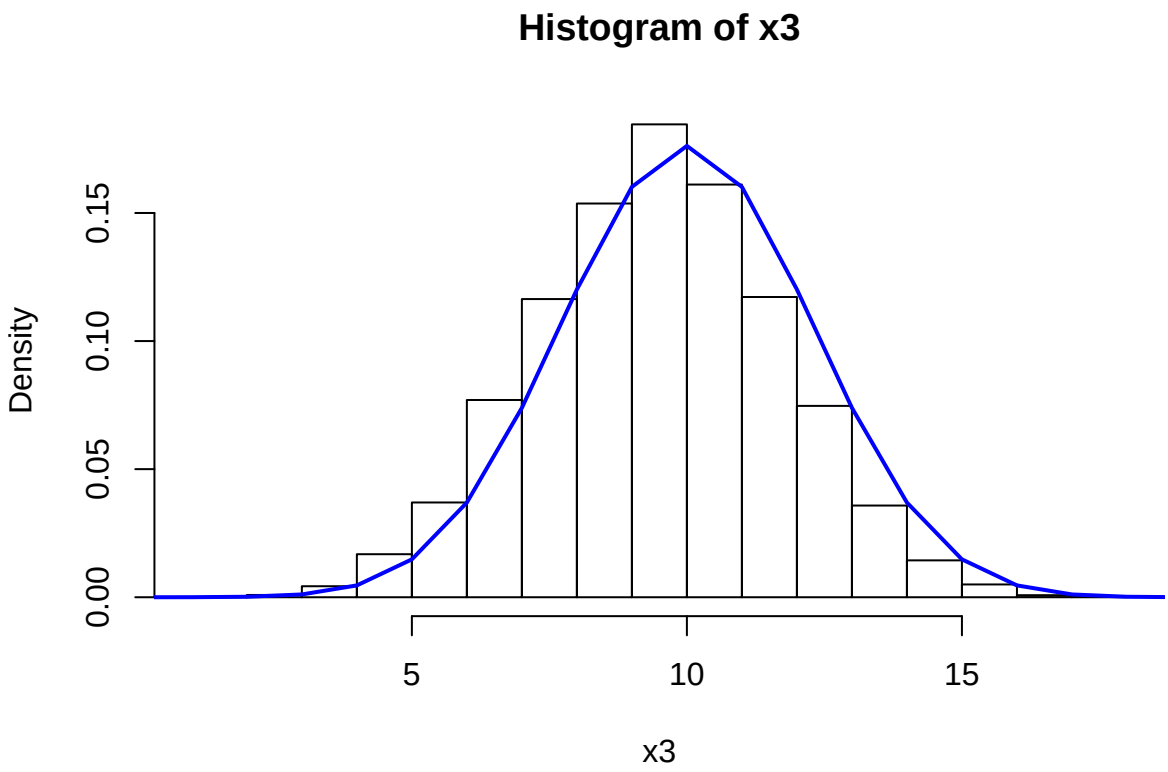
Vi kan tydligare se normalfördelningsformen om vi gör fler dragningar/simuleringar.

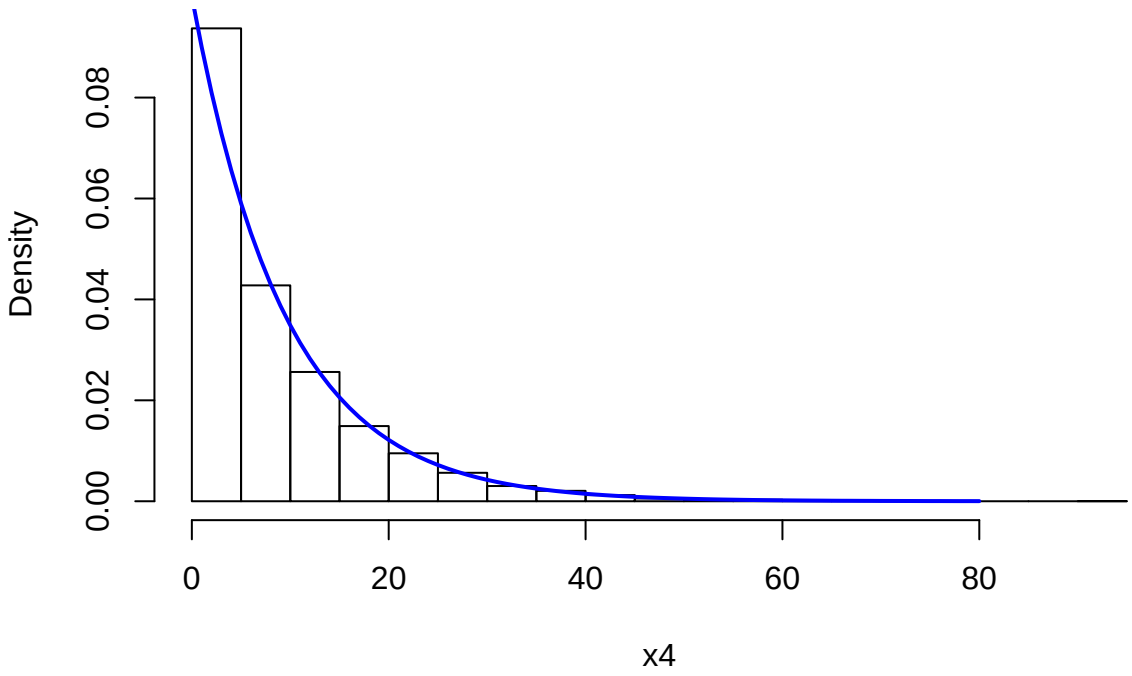
3.1.2

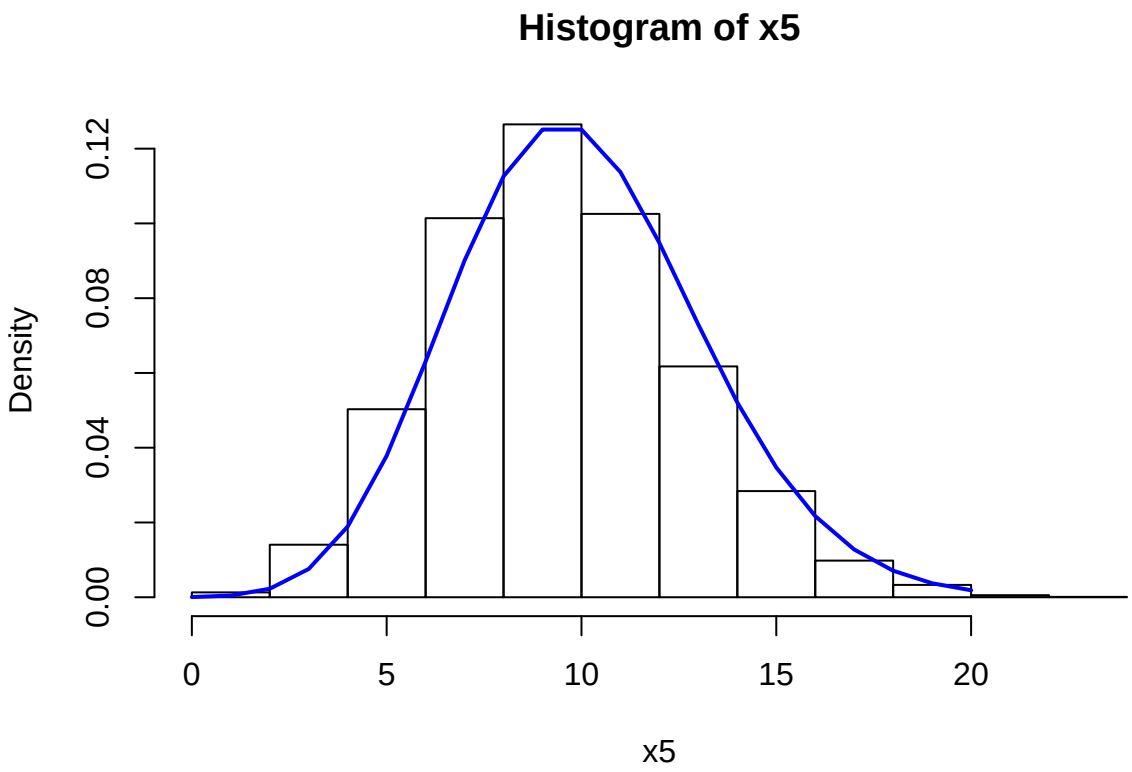
```
n <- 10000
x1 <- rbinom(n, 1, 0.2)
x2 <- rbinom(n, 20, 0.1)
x3 <- rbinom(n, 20, 0.5)
x4 <- rgeom(n, 0.1)
x5 <- rpois(n, 10)

y1 <- runif(n, min = 0, max = 1)
y2 <- rexp(n, 3)
y3 <- rgamma(n, 2, 1)
y4 <- rt(n, 3)
y5 <- rbeta(n, 0.1, 0.1)
y6 <- rbeta(n, 1, 1)
y7 <- rbeta(n, 10, 5)
```

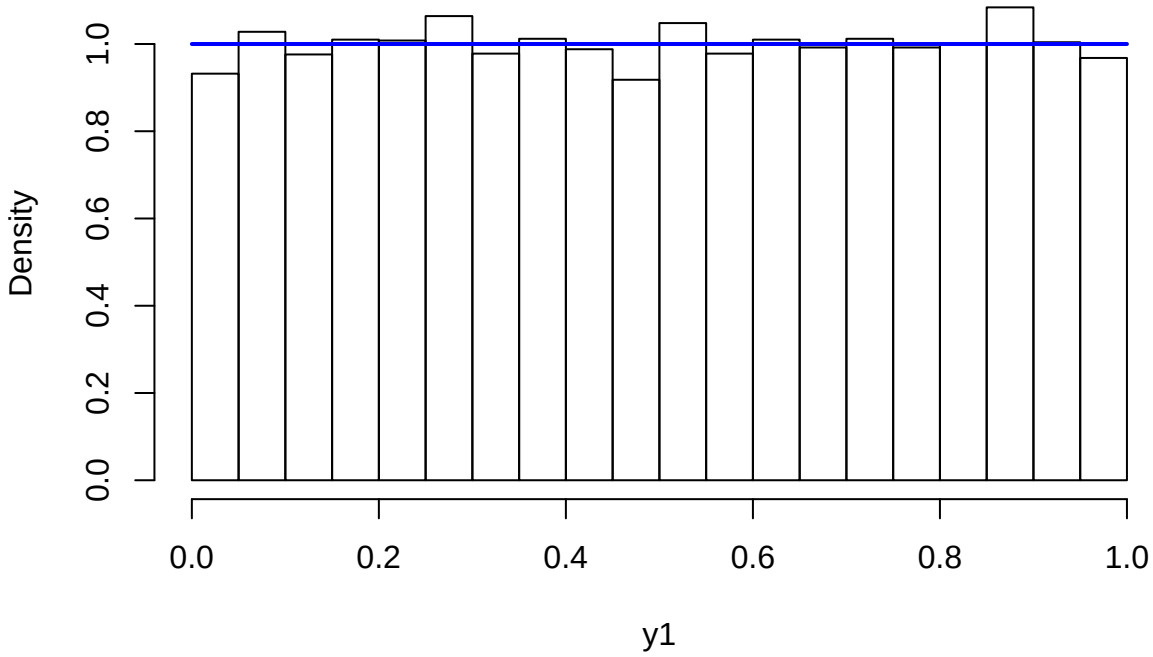

Histogram of x1
Density
0
5
10
15
0.0
0.2
0.4
0.6
0.8
1.0
x1

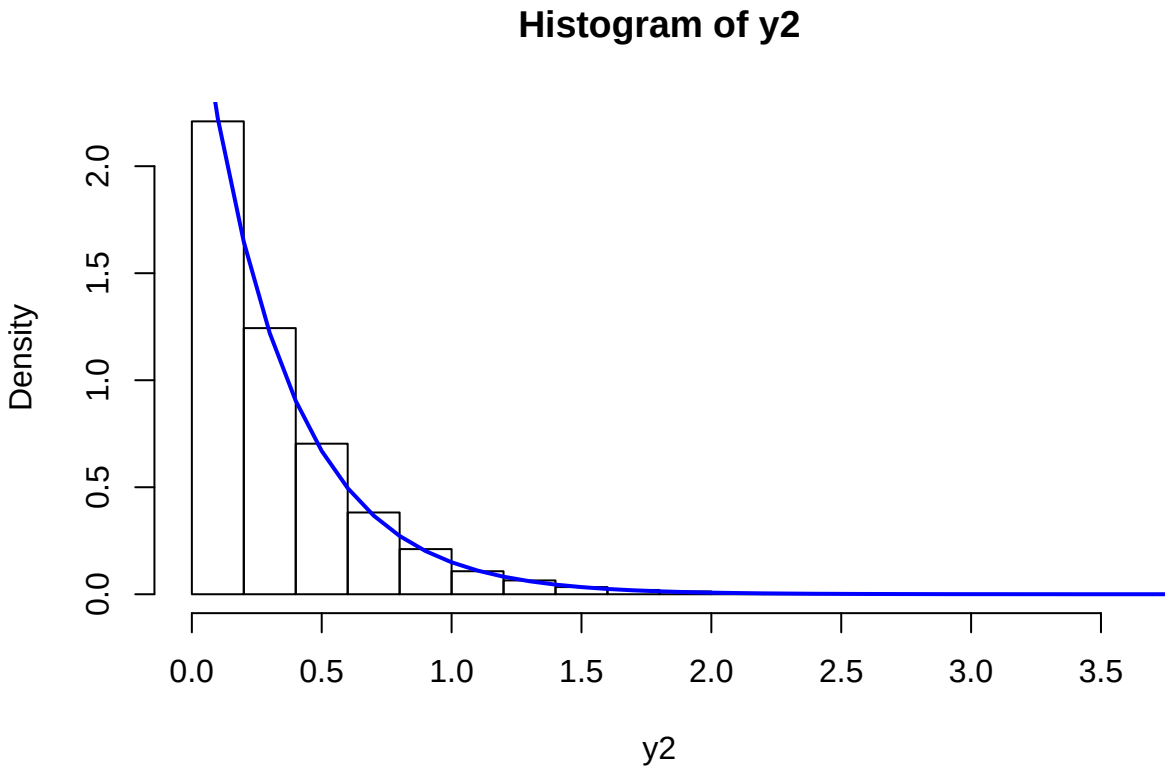

Histogram of x2
Density
0.0
0.1
0.2
0.3
0.4
0.5
0
2
4
6
8
x2

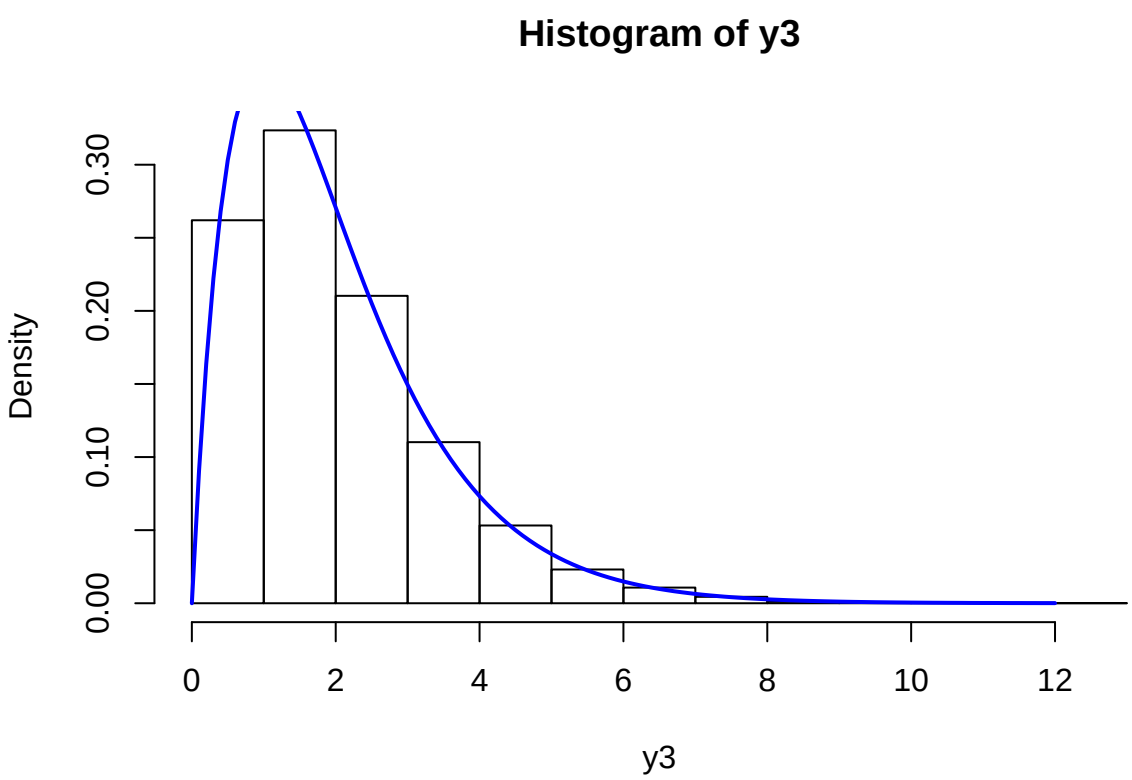

Histogram of x3
Density
0.00
0.05
0.10
0.15
5
10
15
x3

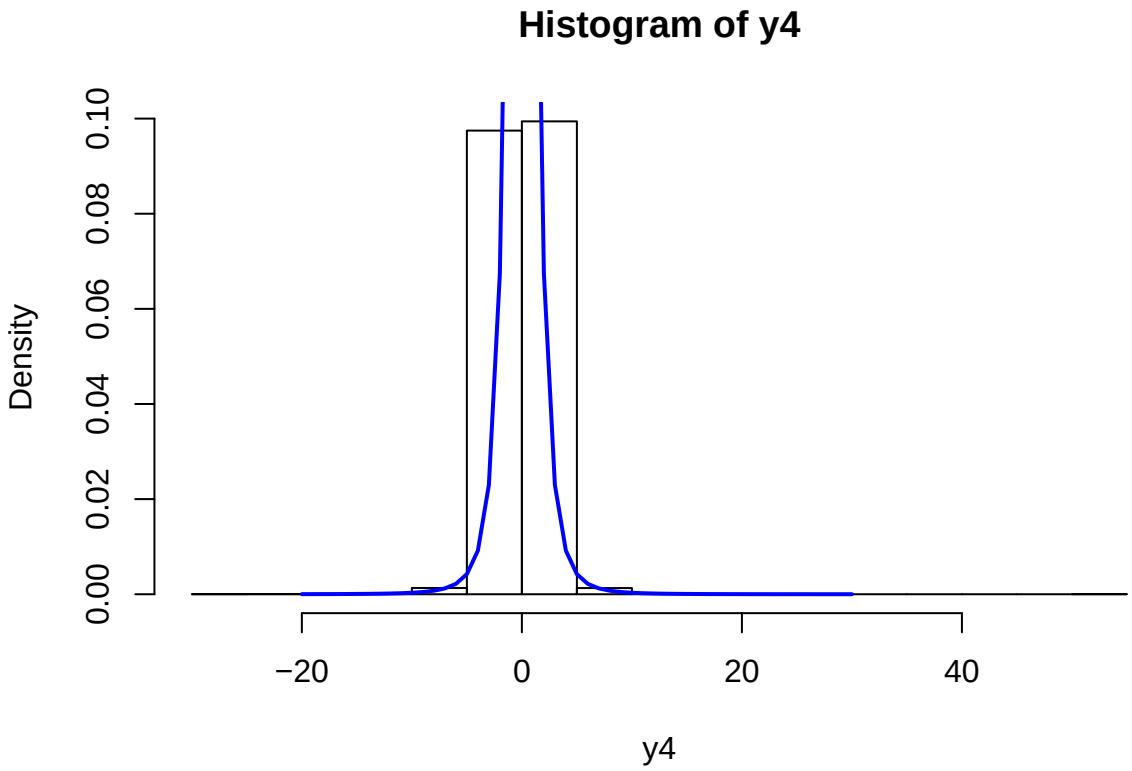

Histogram of x4
Density
0.00
0.02
0.04
0.06
0.08
0
20
40
60
80
x4

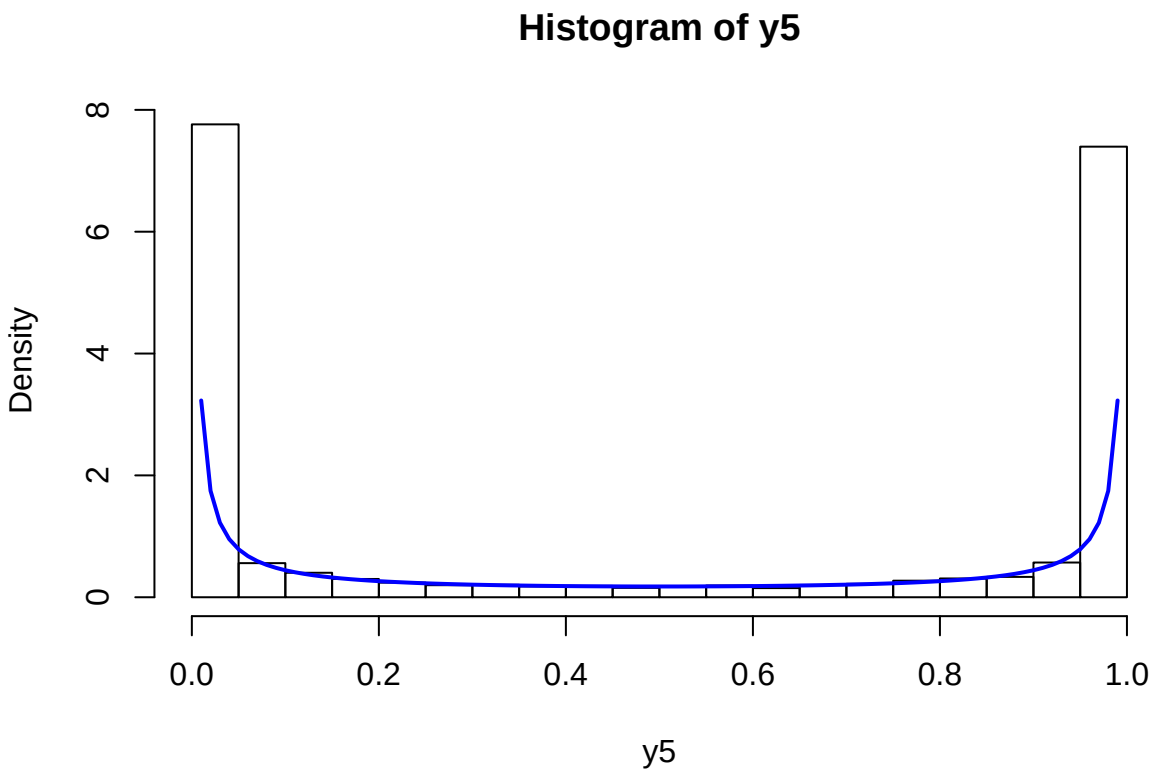

Histogram of x5
Density
0.00
0.04
0.08
0.12
0
5
10
15
20
x5

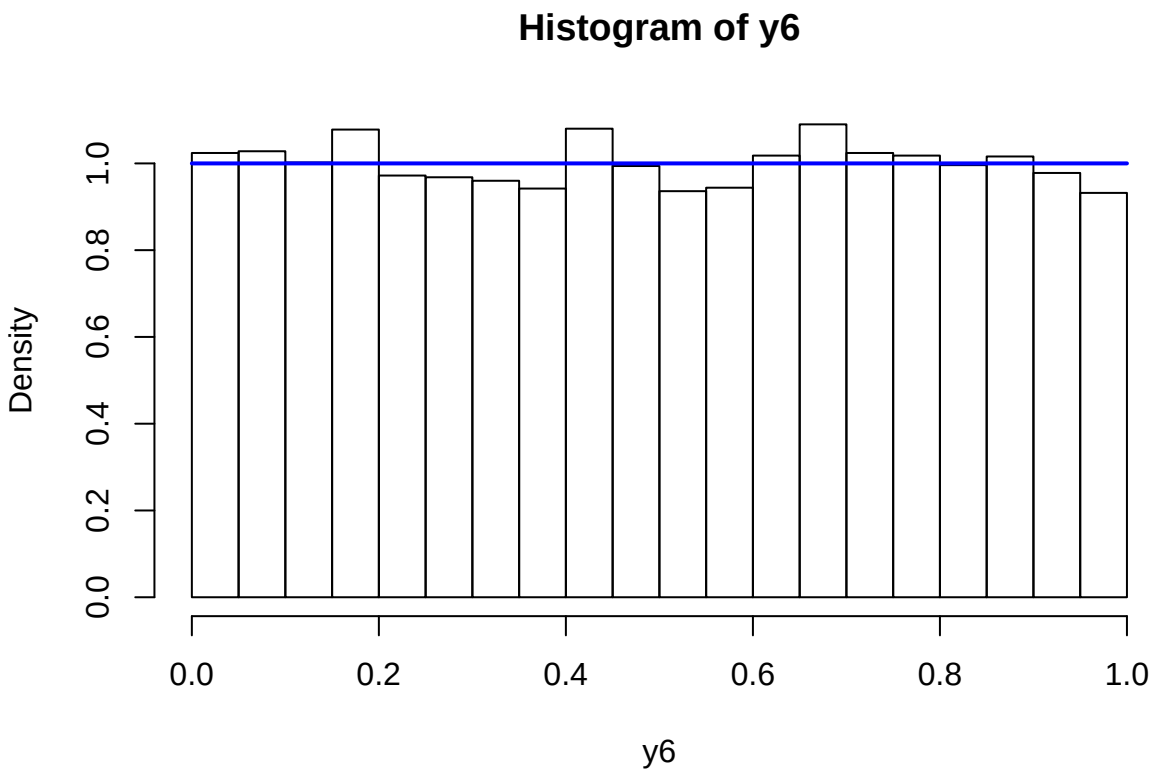

Histogram of y1
Density
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
y1

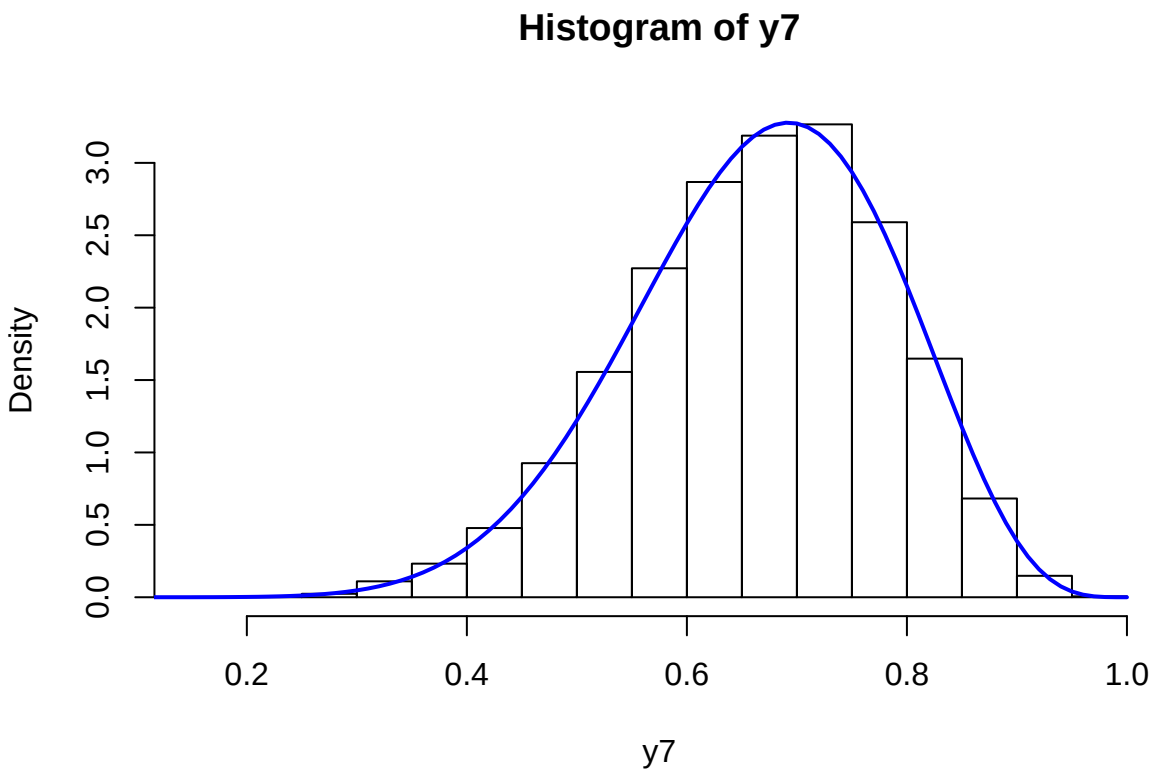

Histogram of y2
Density
0.0
0.5
1.0
1.5
2.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
y2


Histogram of y3
Density
0.00
0.10
0.20
0.30
0
2
4
6
8
10
12
y3


Histogram of y4
Density
0.00
0.02
0.04
0.06
0.08
0.10
−20
0
20
40
y4


Histogram of y5
Density
0
2
4
6
8
0.0
0.2
0.4
0.6
0.8
1.0
y5

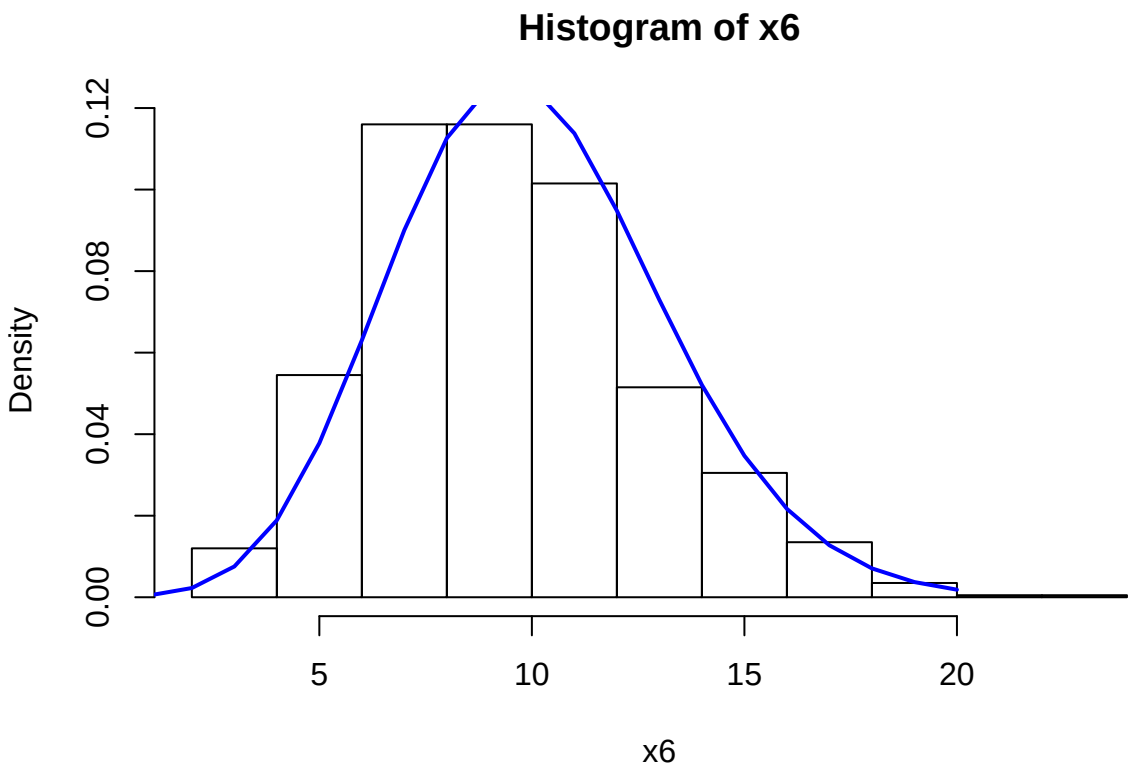

Histogram of y6
Density
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
y6

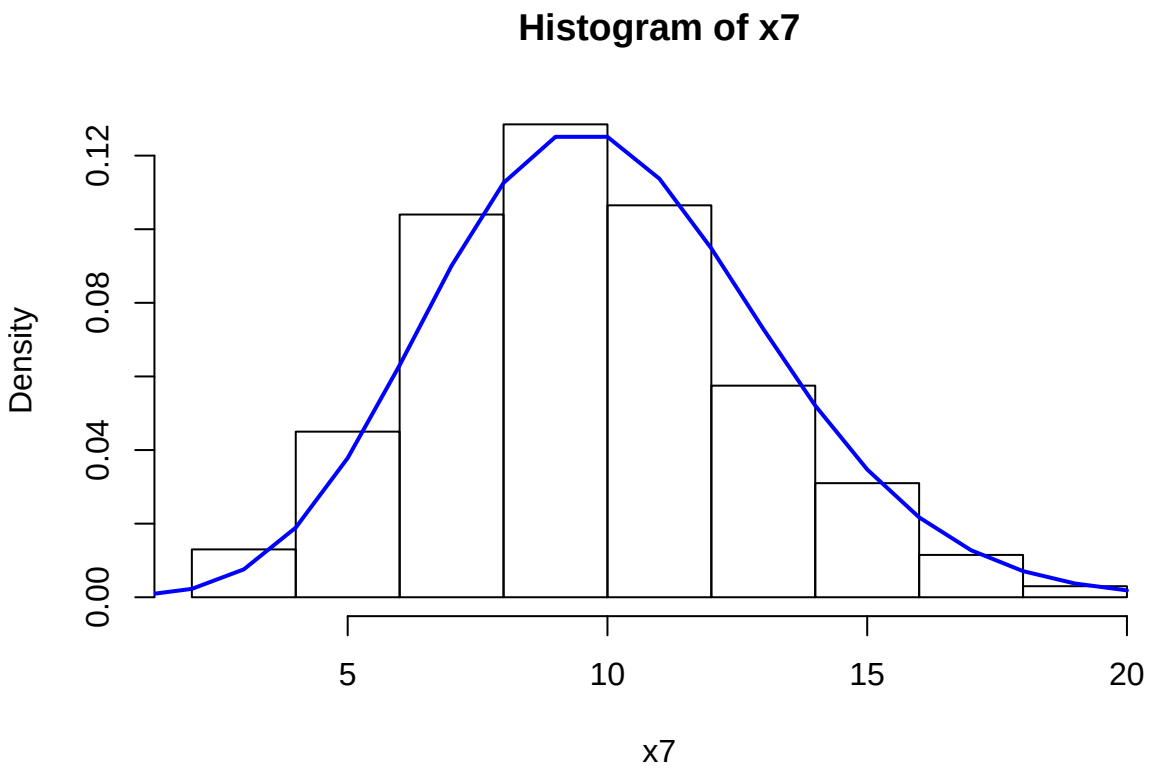


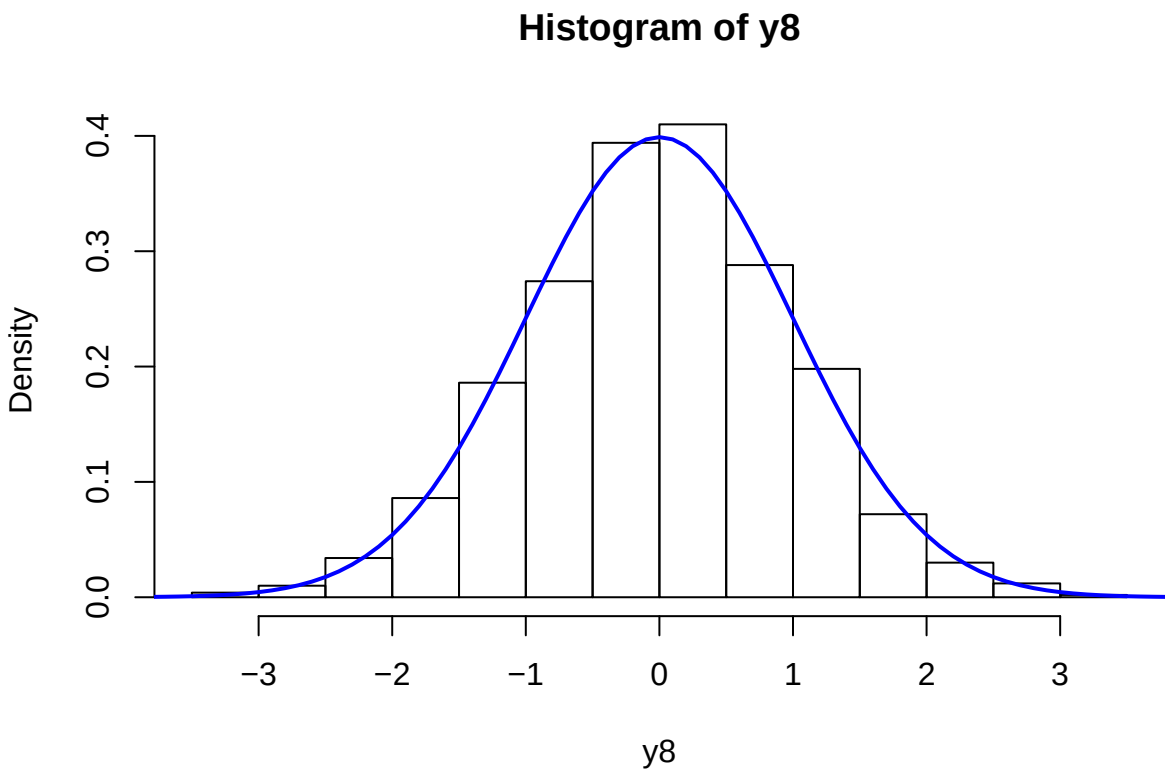
3.1.3

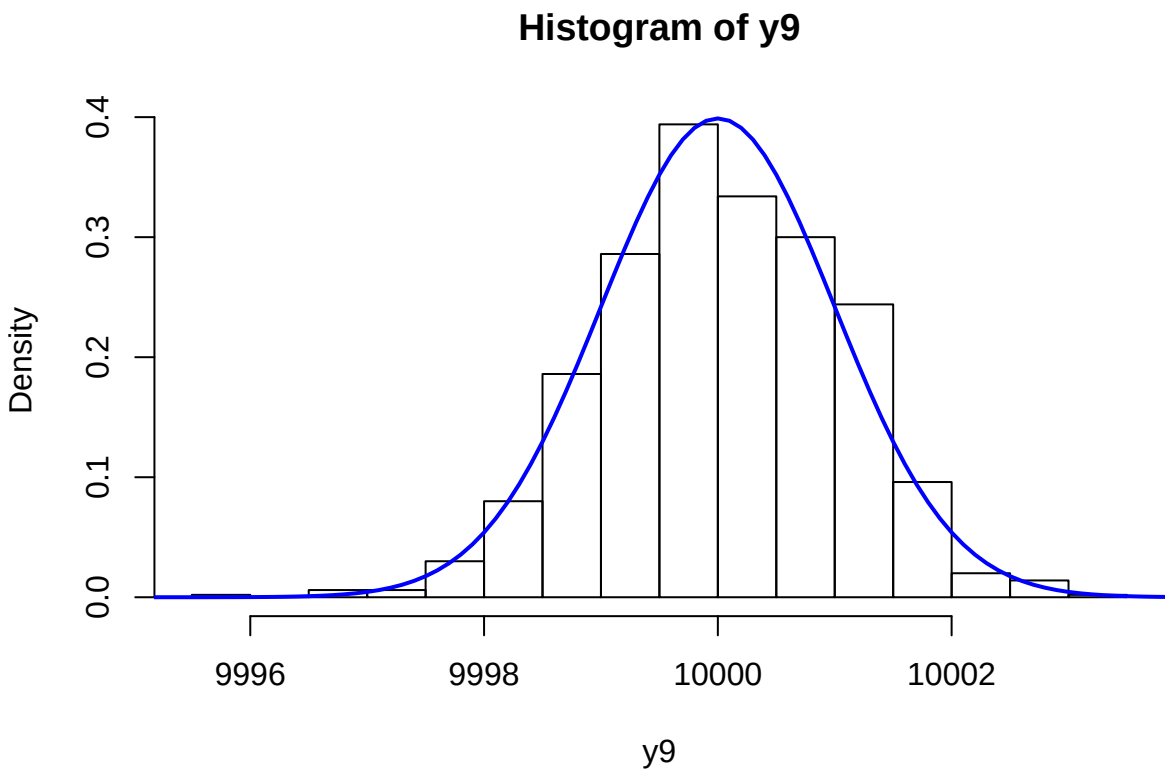
(1) Simulera 1000 dragningar från varje fördelning.

```
n <- 1000
x6 <- rbinom(n, 10000, 0.001)
y8 <- rt(n, 10000)
x7 <- rpois(n, 10000*0.001)
y9 <- rnorm(n, 10000, 1)
```


Histogram of x6
Density
0.00
0.04
0.08
0.12
5
10
15
20
x6


Histogram of x7
Density
0.00
0.04
0.08
0.12
5
10
15
20
x7


Histogram of y8
Density
0.0
0.1
0.2
0.3
0.4
−3
−2
−1
0
1
2
3
y8

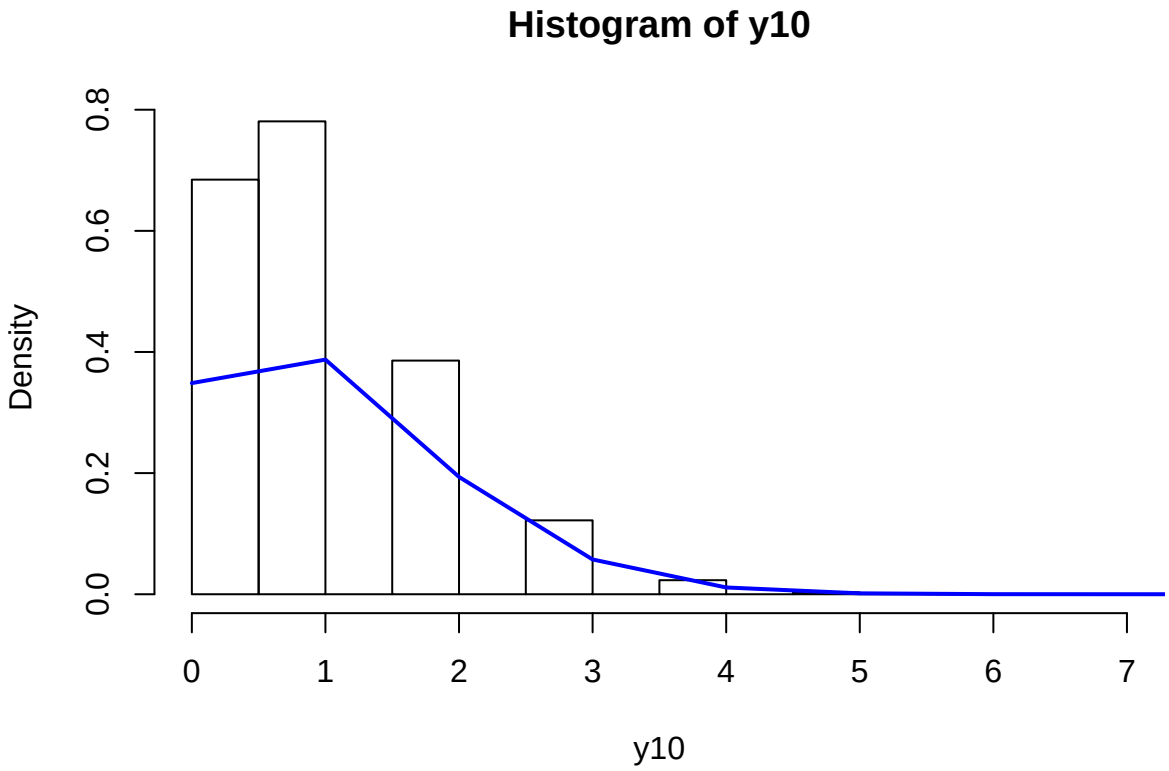


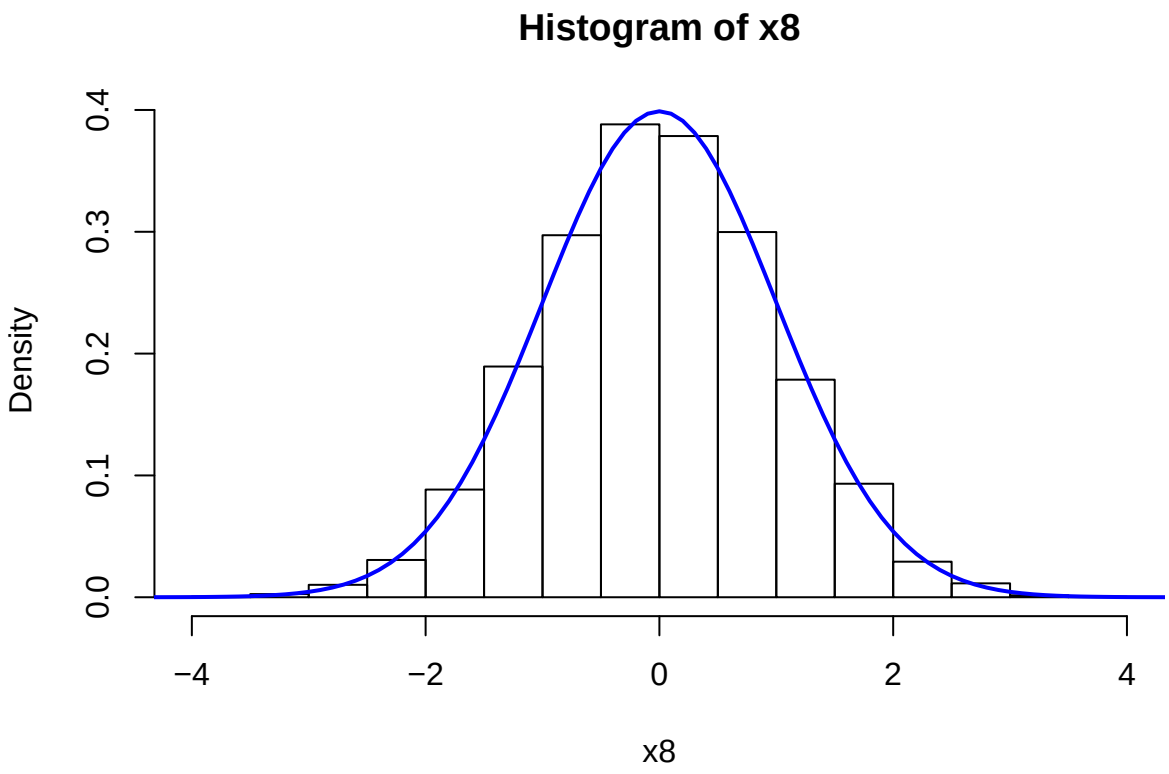
3.1.4

(1) simulera 10000 dragningar från varje fördelning och skriv ut sannolikheten att en dragning är lika med 0

```
y10 <- rbinom(10000, 10, 0.1)
x8 <- rnorm(10000, 0, 1)

p <- dbinom(0, 10, 0.1)
print(p)
```

```
## [1] 0.3486784
```


Histogram of y10
Density
0.0
0.2
0.4
0.6
0.8
0
1
2
3
4
5
6
7
y10



(2) Använd den kumulativa fördelningsfunktionen för att beräkna sannolikheterna

```
p1 <- pnorm(0, 0, 1)
print(p1)
```

```
## [1] 0.5
```

```
p2 <- pnorm(1, 0, 1) - pnorm(-1, 0, 1)
print(p2)
```

```
## [1] 0.6826895
```

```
p3 <- 1 - pnorm(1.96, 0, 1)
print(p3)
```

```
## [1] 0.0249979
```

```
p4 <- pbinom(10, 10, 0.1) - pbinom(0, 10, 0.1)
print(p4)
```

```
## [1] 0.6513216
```

```
eps <- 0.0001
p5 <- pbinom(0 + eps, 10, 0.1) - pbinom(0 - eps, 10, 0.1)
print(p5)
```

```
## [1] 0.3486784
```

```
p6 <- p4 + p5
print(p6)
```

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

(3) Simuleringar för att beräkna samma sannolikheten som i (2)

```
p1s <- sum(x8 < 0) / 10000
print(p1s)
```

```
## [1] 0.5036
```

```
p2s <- (sum(x8 < 1) - sum(x8 <= -1)) / 10000
print(p2s)
```

```
## [1] 0.6819
```

```
p3s <- sum(x8 > 1.96) / 10000
print(p3s)
```

```
## [1] 0.0245
```

```
p4s <- (sum(y10 < 10) - sum(y10 <= 0)) / 10000
print(p4s)
```

```
## [1] 0.6577
```

```
p5s <- sum(y10 == 0) / 10000
print(p5s)
```

```
## [1] 0.3423
```

```
p6s <- (sum(y10 <= 10) - sum(y10 < 0)) / 10000
print(p6s)
```

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

3.1.5

(1) Beräkna antalet förväntade fel.

```
old <- rbinom(n = 10000, size = 337, p = 0.1)
print(sum(old)/10000)
```

```
## [1] 33.6597
```

```
prob <- sum(runif(n = 10000, min = 0.02, max = 0.16))/10000
new <- rbinom(n = 10000, size = 337, p = prob)
print(sum(new)/10000)
```

```
## [1] 30.09
```

(2) Beräkna sannolikheten att vi får mindre fel i den nya jämfört med den gamla.

```
print(sum(new < old)/10000)
```

```
## [1] 0.6577
```

(3) Beräkna sannolikheten att vi får mer än 50 fel.

```
sum(old > 50)/10000
```

```
## [1] 0.0018
```

```
    sum(new > 50)/10000
```

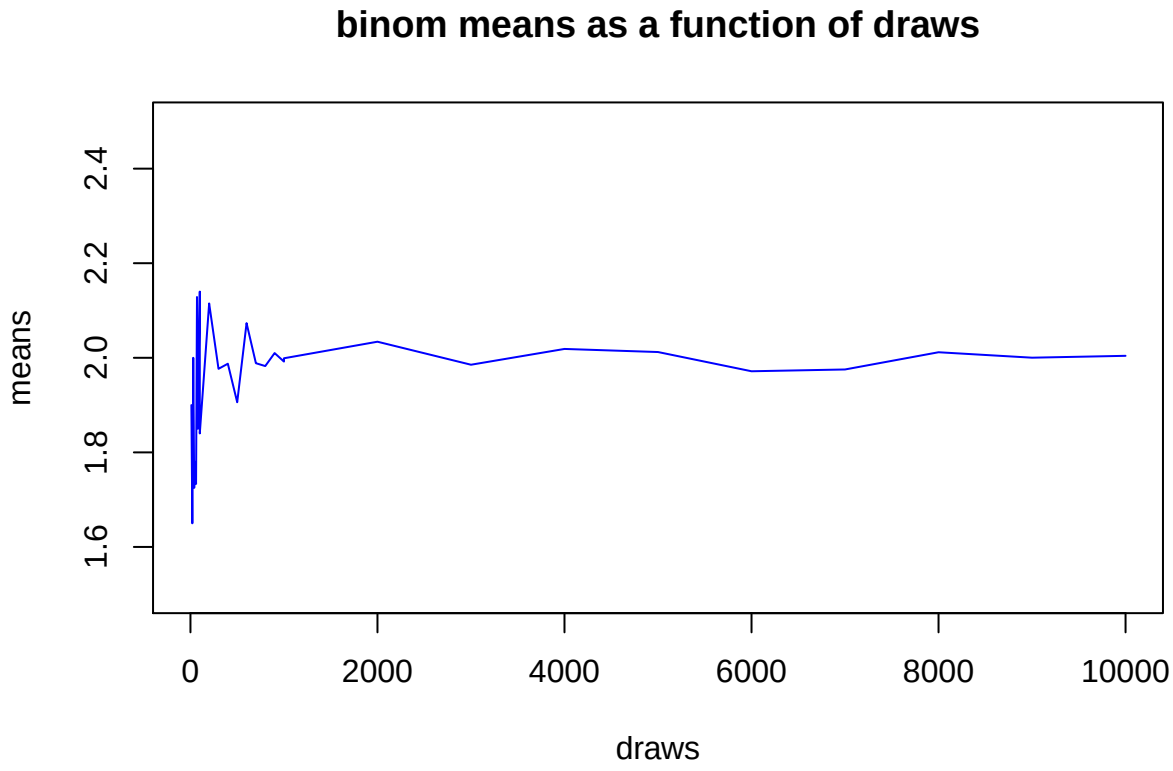
```
## [1] 3e-04
```

3.2.1

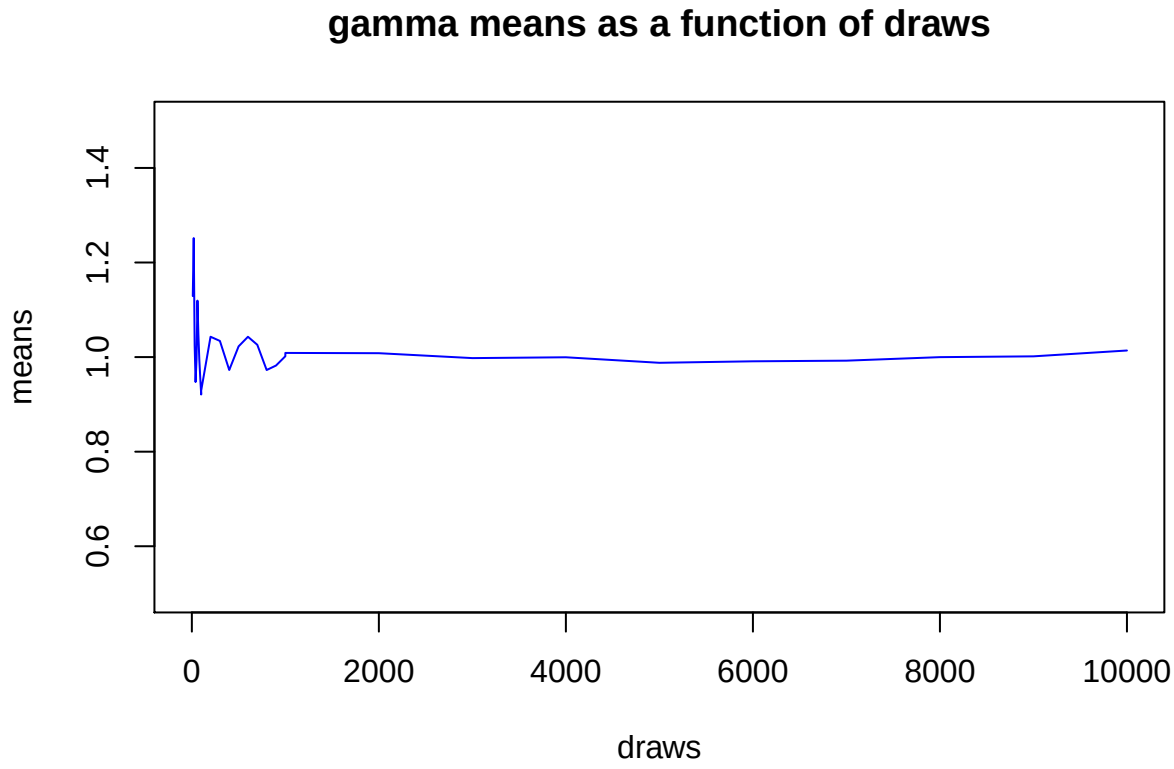
(1) $\mathrm{E(X)} = 10 * 0.2 = 2$

$\mathrm{E(Y)} = 2 \ / \ 2 = 1$

```
    draws <- c(seq(10, 100, 10), seq(100, 1000, 100), seq(1000, 10000, 1000))
    binomMeans <- numeric(length(draws))
    gammaMeans <- numeric(length(draws))
    for (i in 1:length(draws)) {
        n <- draws[i]
        binomMeans[i] <- mean(rbinom(n, 10, 0.2))
        gammaMeans[i] <- mean(rgamma(n, 2, 2))
    }
```


binom means as a function of draws
means
1.6
1.8
2.0
2.2
2.4
0
2000
4000
6000
8000
10000
draws

(2)


gamma means as a function of draws
means
0.6
0.8
1.0
1.2
1.4
0
2000
4000
6000
8000
10000
draws

3.3.1

(1) E(X) = 1 / 10 = 0.1

Var(X) = 1 / (10^2) = 0.01

E(Y) = 3

Var(Y) = 3

(2) Använd simuleringarna för att beräkna medelvärdet och variansen.

```
    x <- rexp(10000, 10)
    print(mean(x))
```

```
## [1] 0.1008502
```

```
    print(var(x))
```

```
## [1] 0.01000535
```

```
    y <- rpois(10000, 3)
    print(mean(y))
```

```
## [1] 2.9795
```

```
    print(var(y))
```

```
## [1] 3.009381
```

(3) E(3) = 3

E(3X + 2) = E(3X) + E(2) = 3 * E(X) + 2 = 0.3 + 2 = 2.3

E(X + Y) = E(X) + E(Y) = 0.1 + 3 = 3.1

E(X * Y) = E(X) * E(Y) = 0.1 * 3 = 0.3

E(3X + 2Y - 3) = 3E(X) + 2E(Y) - 3 = 0.3 + 6 - 3 = 3.3

Var(2X - 5) = 2^2 * Var(X) = 4 * 0.01 = 0.04

Var(X + Y) = Var(X) + Var(Y) = 0.01 + 3 = 3.01

(4) Använd simulering för att beräkna värdena i 3.

```
    print(mean(3))
```

```
## [1] 3
```

```
    print(mean(3*x + 2))
```

```
## [1] 2.302551
```

```
    print(mean(x + y))
```

```
## [1] 3.08035
```

```
    print(mean(x * y))
```

```
## [1] 0.3022441
```

```
    print(mean(3*x + 2*y - 3))
```

```
## [1] 3.261551
```

```
    print(var(2*x - 5))
```

```
## [1] 0.0400214
```

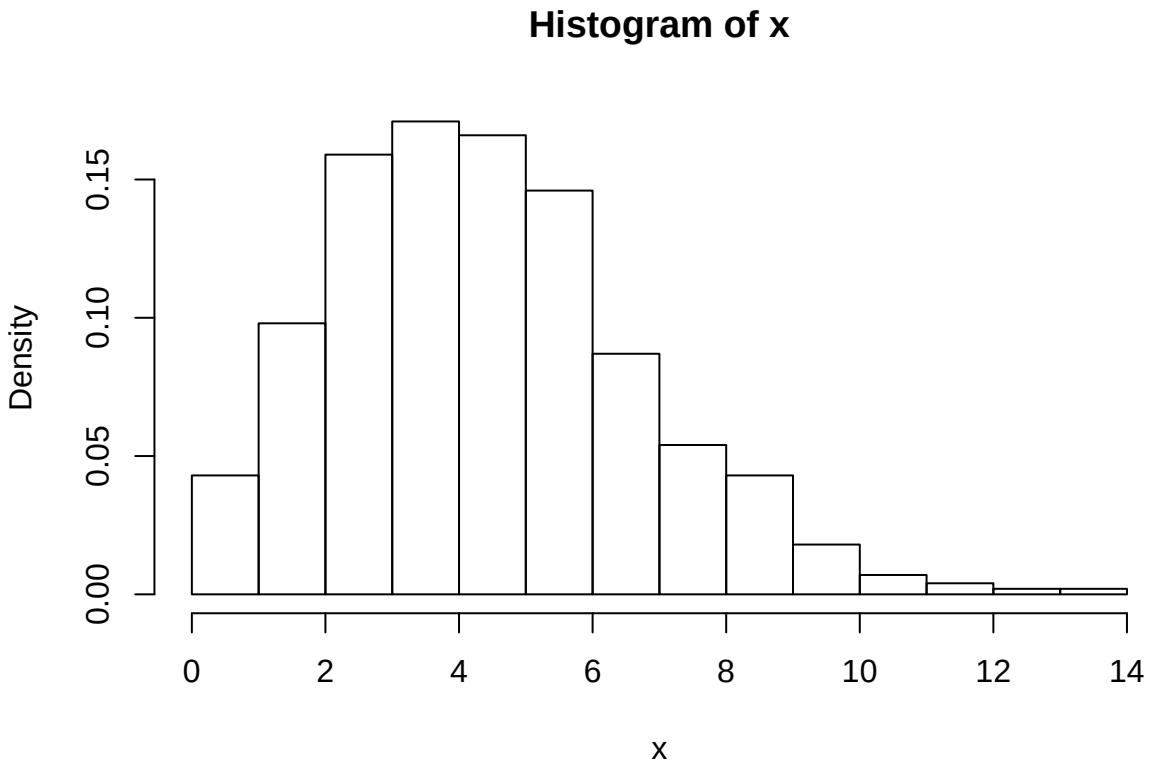
```
    print(var(x + y))
```

```
## [1] 3.022908
```

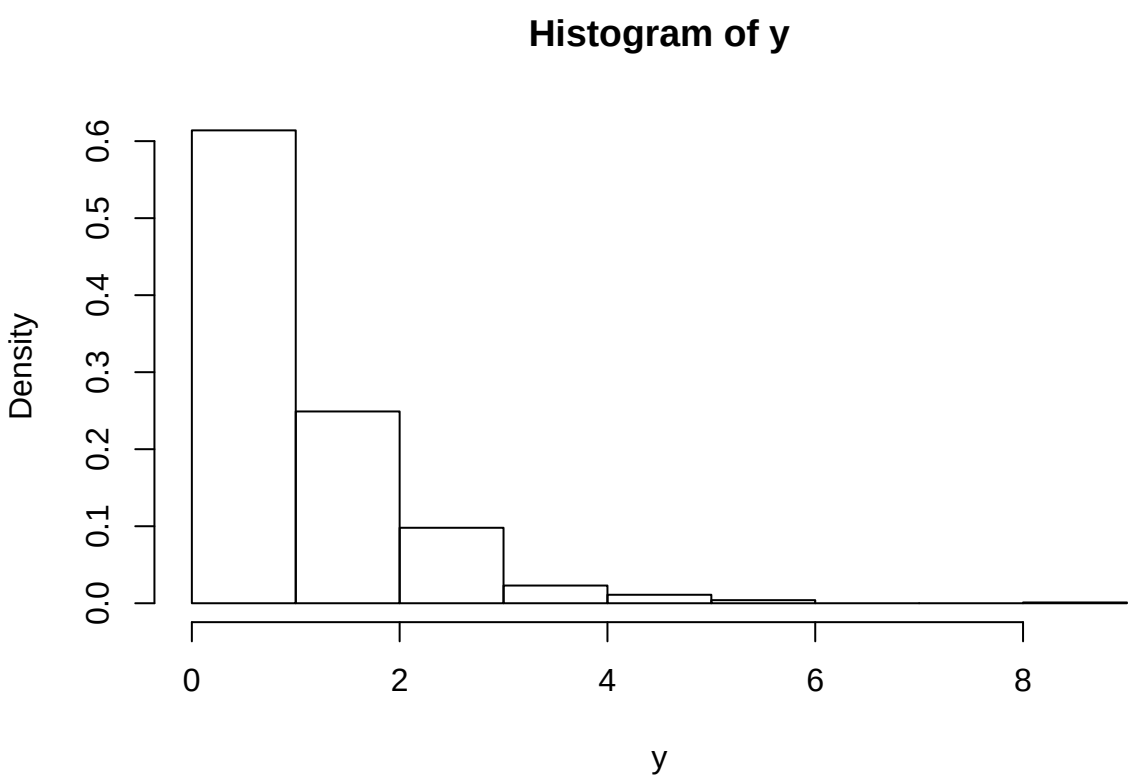
3.4.1

(1) Kör 1000 dragningar och visualisera med histogram.

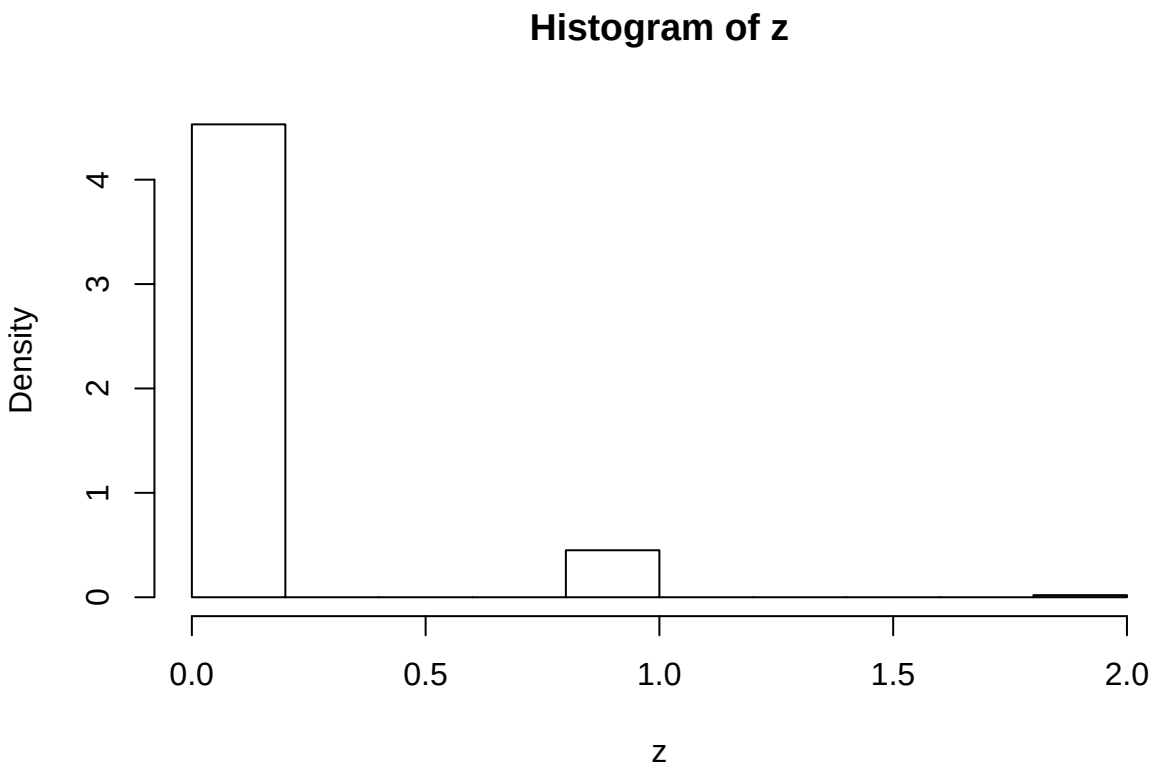
```
    x <- rpois(1000, 5)
    y <- rexp(1000, 1)
    z <- rbinom(1000, 10, 0.01)

    hist(x, probability = TRUE)
```




```
    hist(y, probability = TRUE)
```



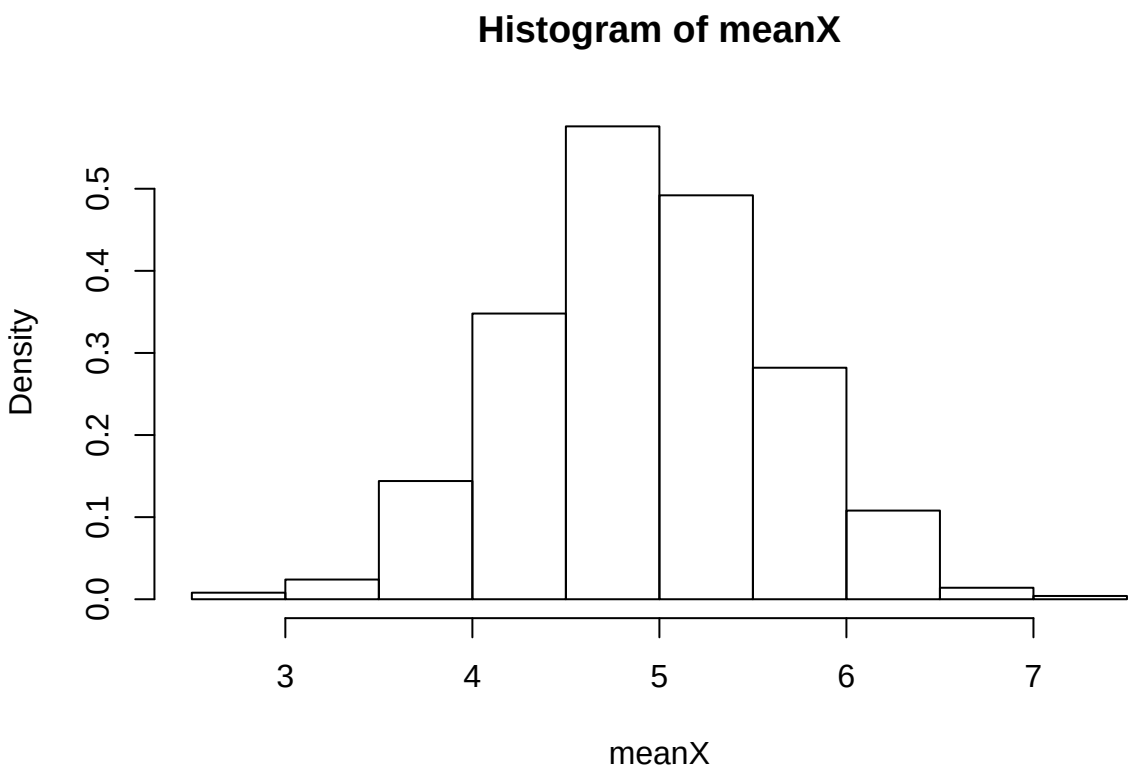

```
    hist(z, probability = TRUE)
```

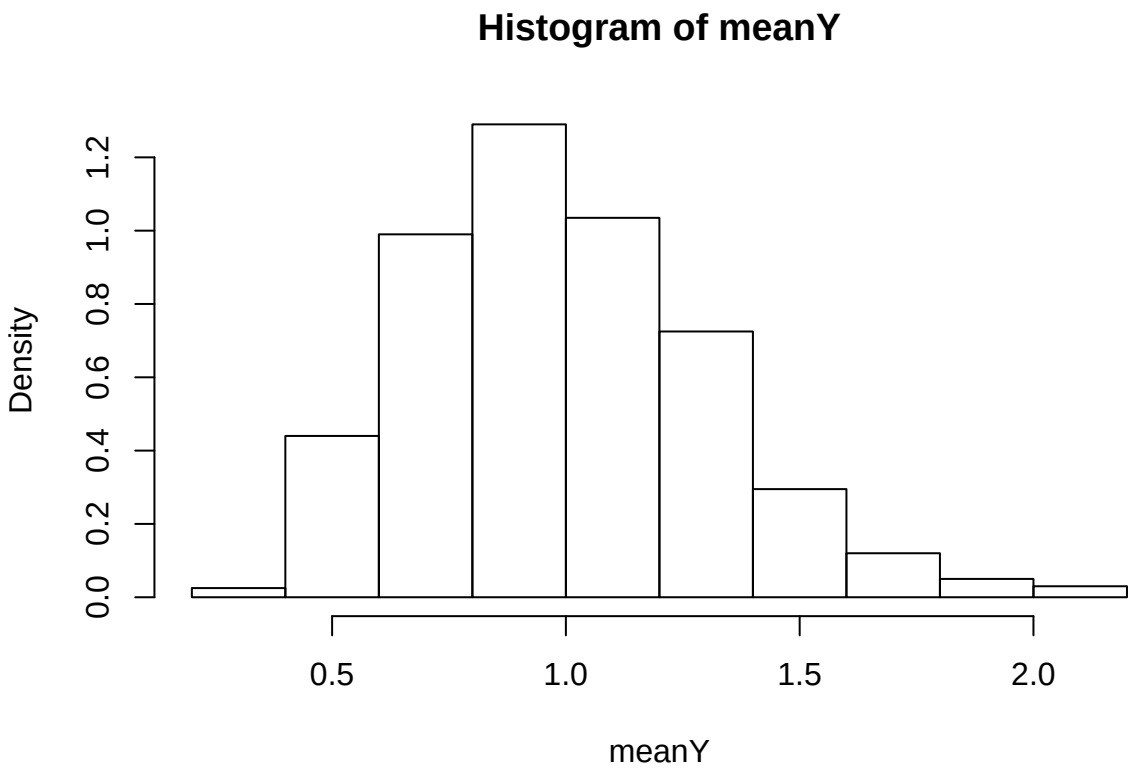


(2) kör 10 dragningar 1000 gånger och beräkna medelvärdet för varje 10 dragningar.

Sen visas medelvärdena som histogram.

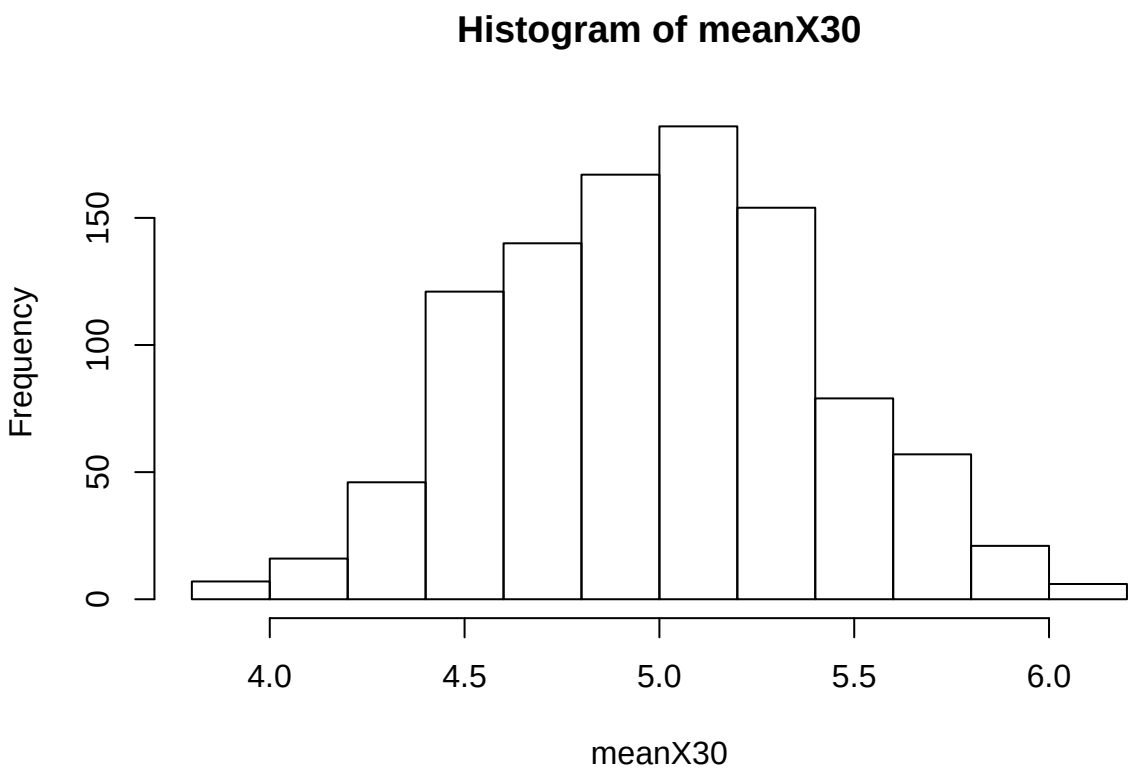
```
meanX <- numeric(0)
meanY <- numeric(0)
for (i in 1:1000) {
    meanX <- c(meanX, mean(rpois(10, 5)))
    meanY <- c(meanY, mean(rexp(10, 1)))
}
```

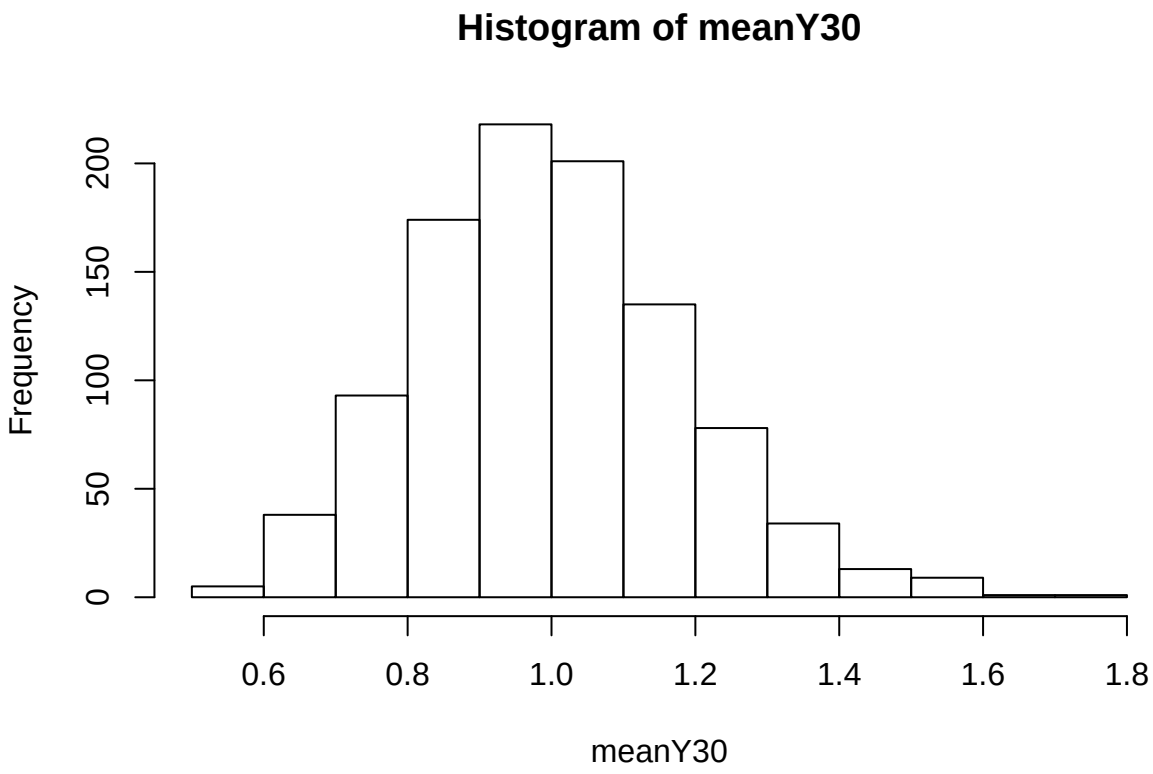

Histogram of meanX
Density
0.0
0.1
0.2
0.3
0.4
0.5
3
4
5
6
7
meanX

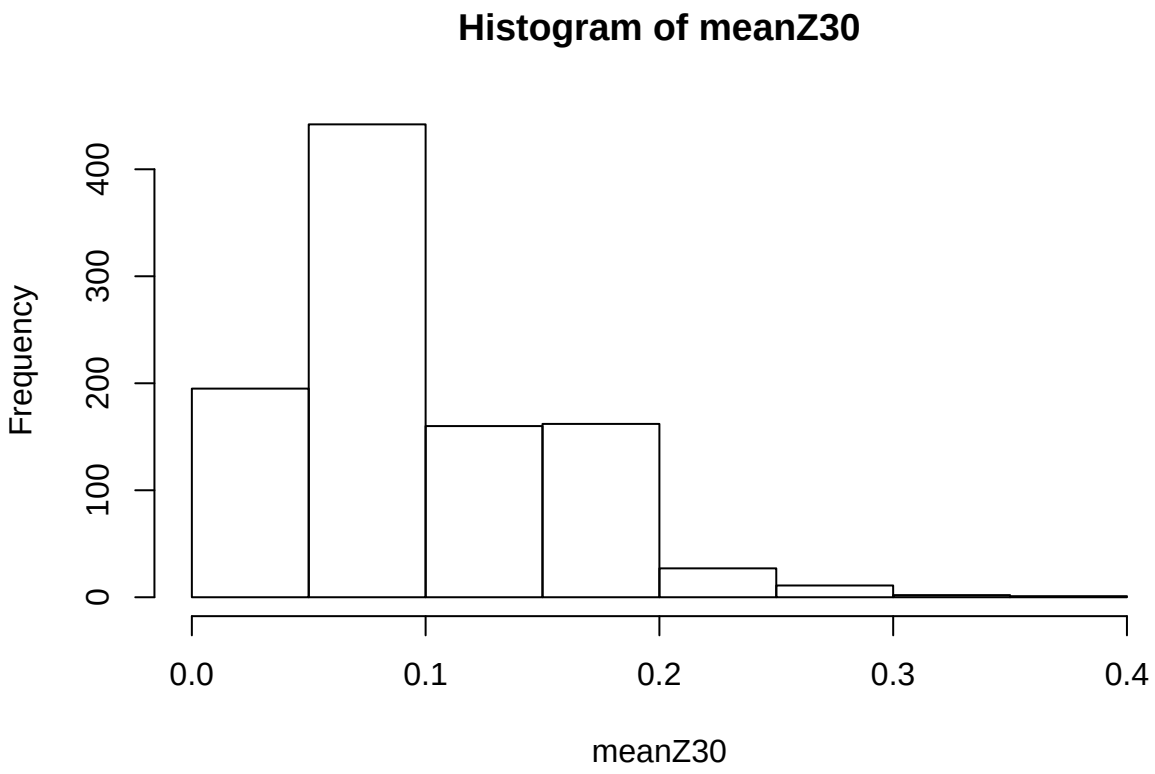


(3) kör 30, 100 och 1000 dragningar 1000 gånger och beräkna medelvärdet för varje 30, 100, 1000 dragningar. Sen visas medelvärdena som histogram.

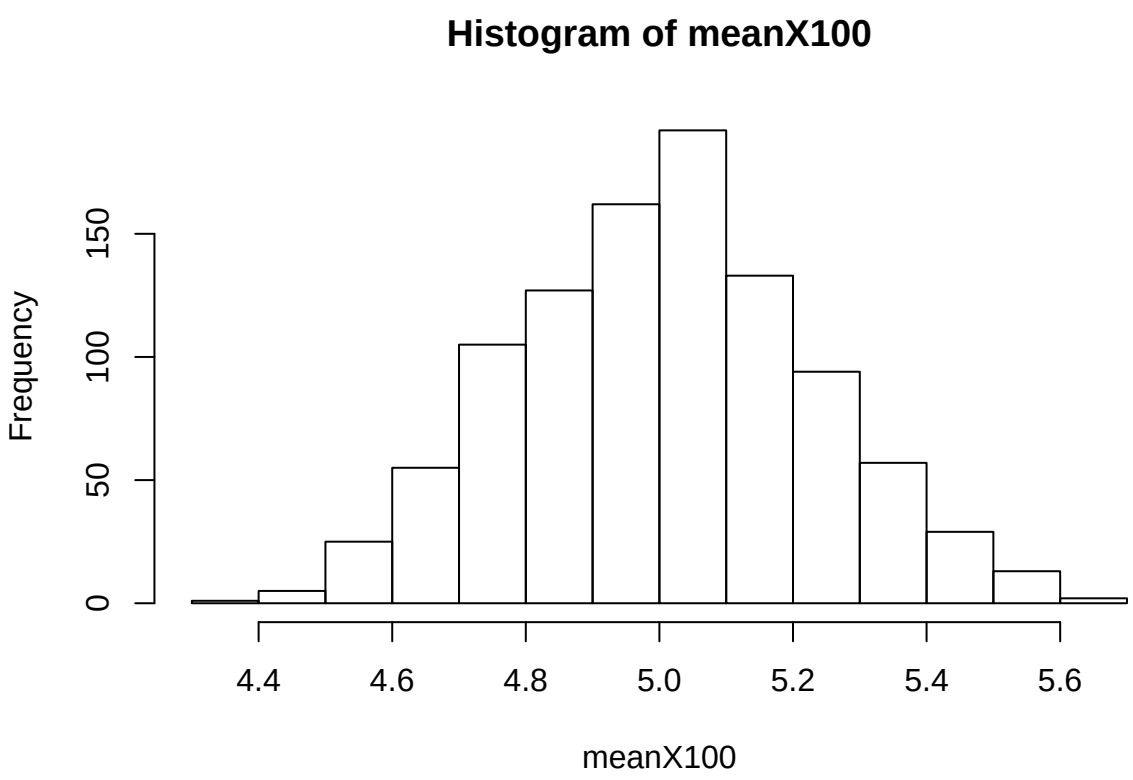
```
meanX30 <- numeric(0)
meanY30 <- numeric(0)
meanZ30 <- numeric(0)
for (i in 1:1000) {
    meanX30 <- c(meanX30, mean(rpois(30, 5)))
    meanY30 <- c(meanY30, mean(rexp(30, 1)))
    meanZ30 <- c(meanZ30, mean(rbinom(30, 10, 0.01)))
}
```

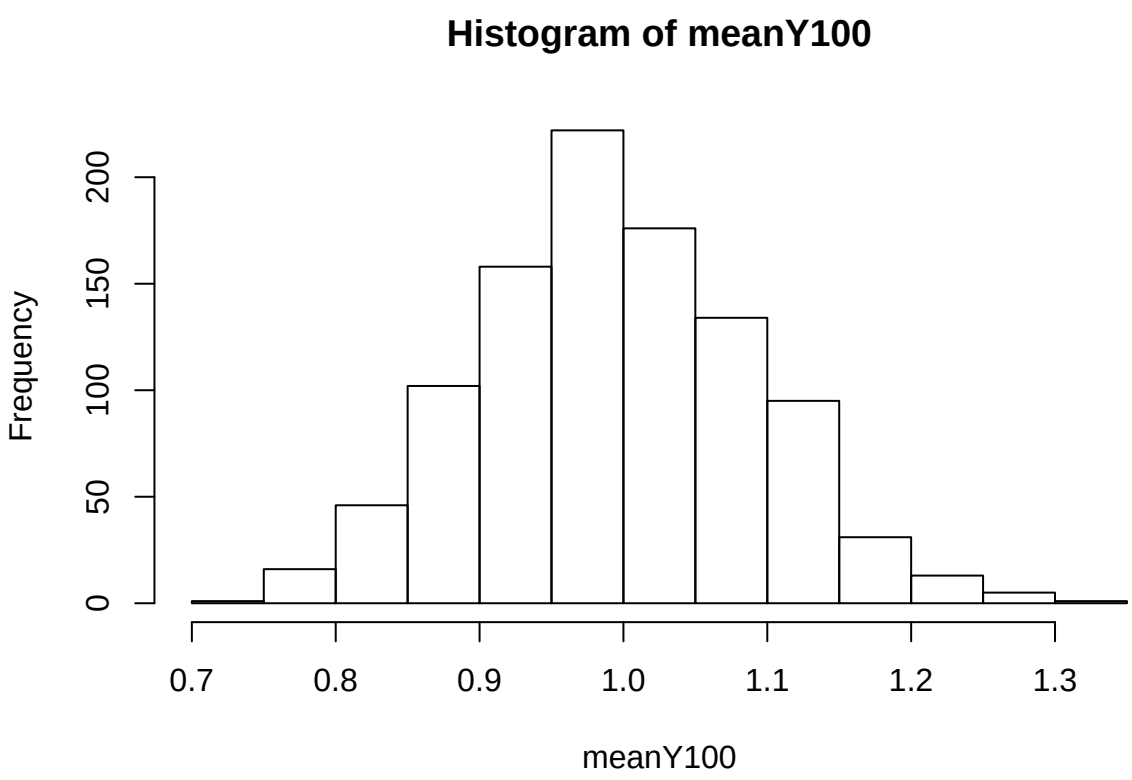

Histogram of meanX30
Frequency
0
50
100
150
4.0
4.5
5.0
5.5
6.0
meanX30

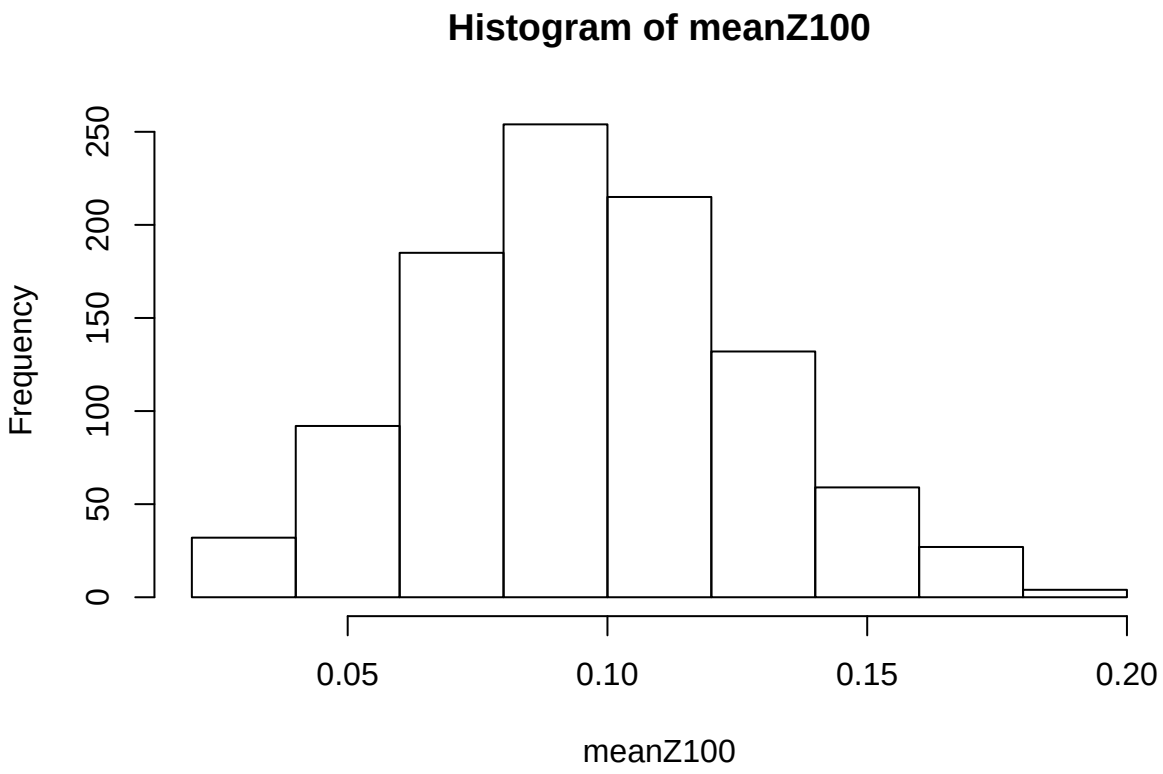

Histogram of meanY30
Frequency
0
50
100
150
200
0.6
0.8
1.0
1.2
1.4
1.6
1.8
meanY30



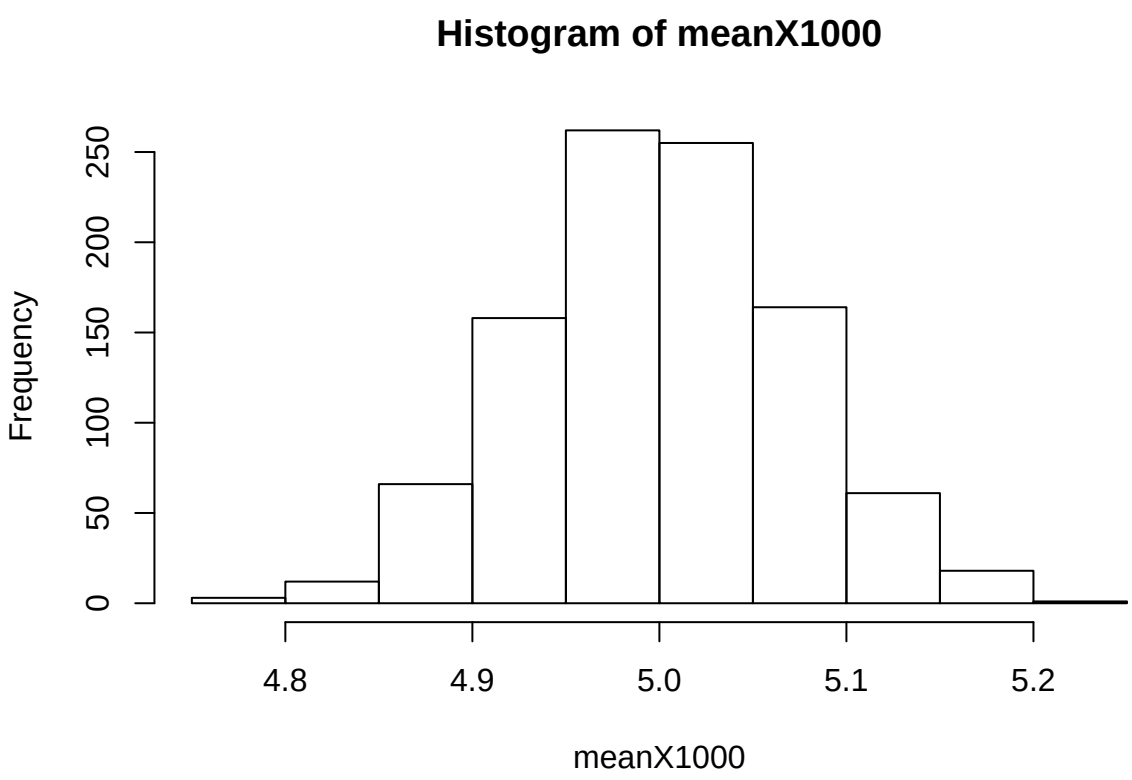

```
meanX100 <- numeric(0)
meanY100 <- numeric(0)
meanZ100 <- numeric(0)
for (i in 1:1000) {
    meanX100 <- c(meanX100, mean(rpois(100, 5)))
    meanY100 <- c(meanY100, mean(rexp(100, 1)))
    meanZ100 <- c(meanZ100, mean(rbinom(100, 10, 0.01)))
}
```

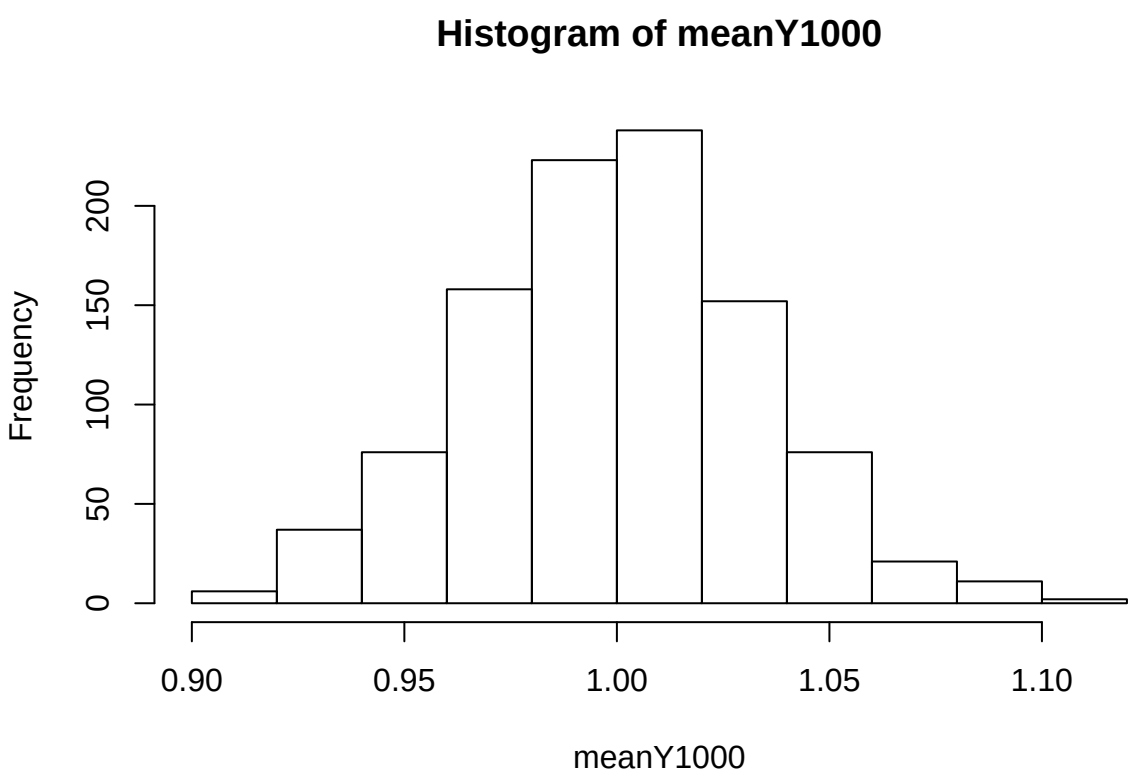

Histogram of meanX100
Frequency
0
50
100
150
4.4
4.6
4.8
5.0
5.2
5.4
5.6
meanX100

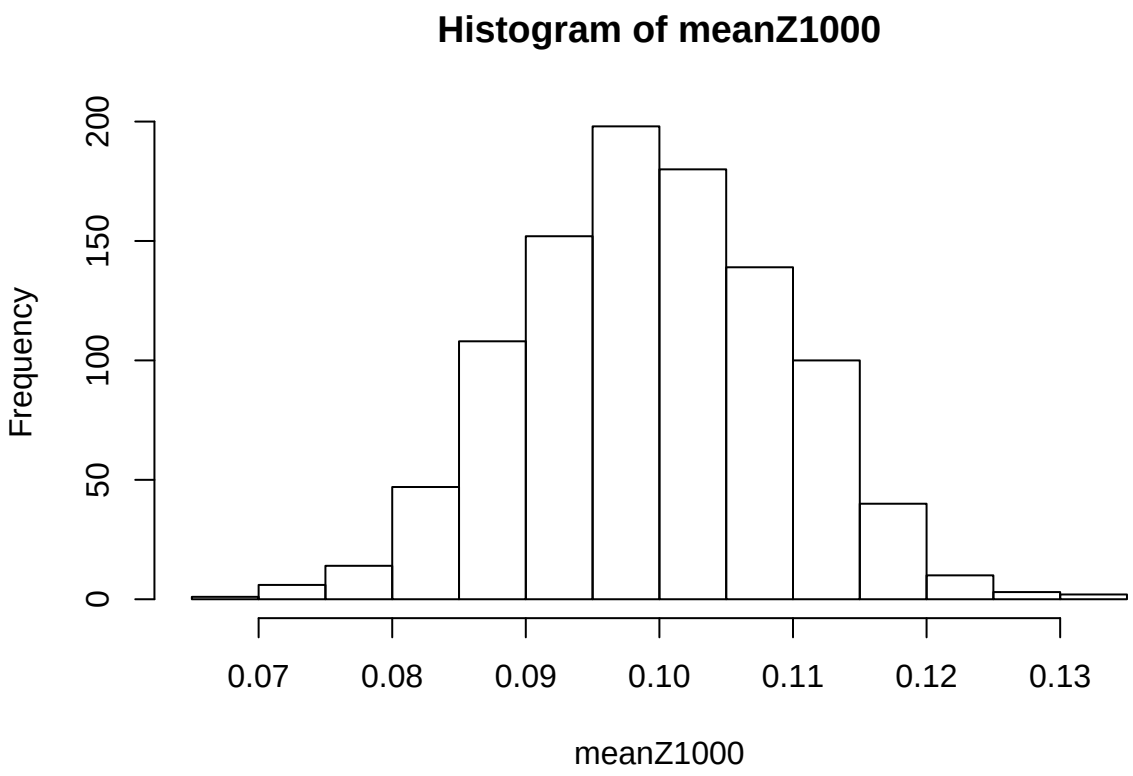

Histogram of meanY100
Frequency
0
50
100
150
200
0.7
0.8
0.9
1.0
1.1
1.2
1.3
meanY100




```
meanX1000 <- numeric(0)
meanY1000 <- numeric(0)
meanZ1000 <- numeric(0)
for (i in 1:1000) {
    meanX1000 <- c(meanX1000, mean(rpois(1000, 5)))
    meanY1000 <- c(meanY1000, mean(rexp(1000, 1)))
    meanZ1000 <- c(meanZ1000, mean(rbinom(1000, 10, 0.01)))
}
```


Histogram of meanX1000
Frequency
0
50
100
150
200
250
4.8
4.9
5.0
5.1
5.2
meanX1000


Histogram of meanY1000
Frequency
0
50
100
150
200
0.90
0.95
1.00
1.05
1.10
meanY1000



Centrala gränsvärdessatsen säger att medelvärdena konvergerar till en normalfördelning vilket de verkar göra.

Vid låg varians så minskar spridningen på medelvärdena vilket borde leda till att den konvergerar snabbare till en normalfördelning.

Var(X) = 5

Var(Y) = 1/(1^2) = 1

Var(Z) = 10 * 0.01 * 0.99 = 0.099

Detta borde då leda till att Z konvergerar snabbast mot en normalfördelning.