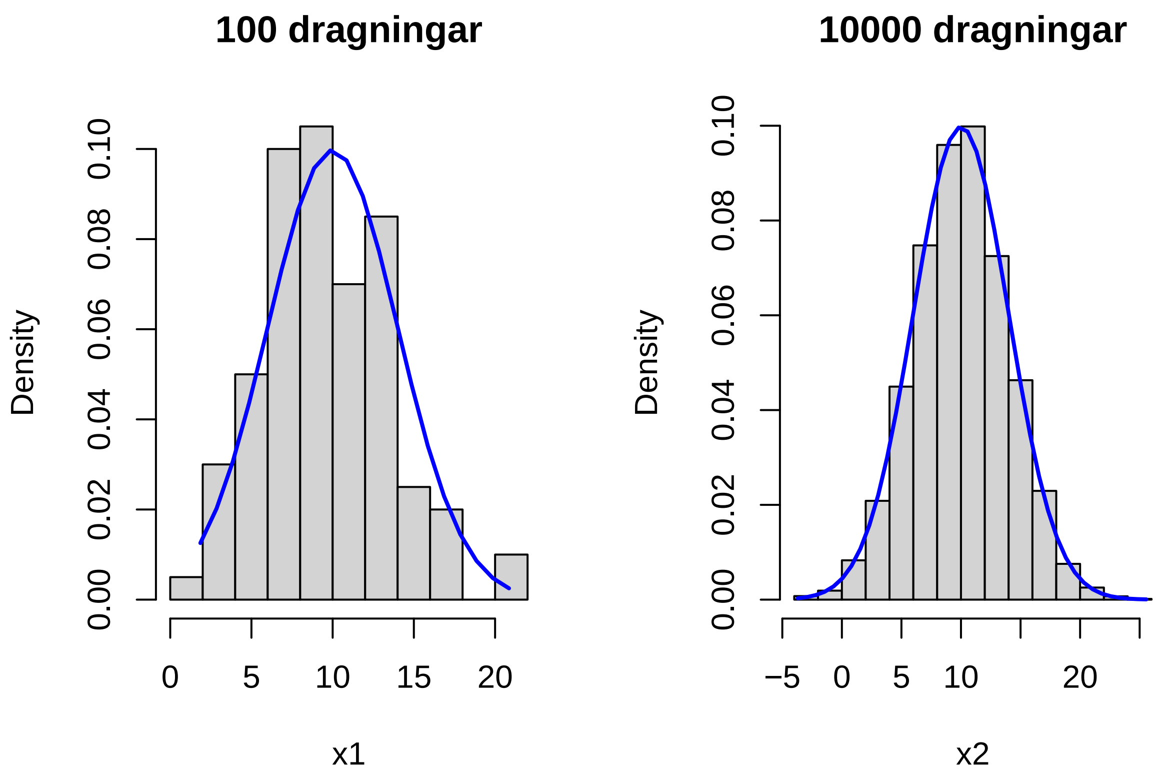
# Laboration 1

Amanda Sandberg (amasa869) & Philip Norberg (phino637)

Uppgift 3.1.1 (1)

```r
par(mfrow=c(1,2))
x.mean<-10
x.std<-4

# 100 dragningar
x1 <- rnorm(100, x.mean, x.std)
x1fit <- seq(min(x1), max(x1), 1)
pdf1 <- dnorm(x1fit, x.mean, x.std, log = FALSE)
hist(x1, probability=TRUE,breaks = 12, main = "100 dragningar")
lines(x1fit, pdf1, type = "l", col="blue", lwd=2)

# 10000 dragningar
x2 <- rnorm(10000, x.mean, x.std)
x2fit <- seq(min(x2), max(x2), length =40)
pdf2 <- dnorm(x2fit, x.mean, x.std, log = FALSE)
hist(x2, probability=TRUE, breaks = 20,  main = "10000 dragningar")
lines(x2fit, pdf2, type ="l", col="blue", lwd=2)
```



Uppgift 3.1.1 (2)
Histogrammet som visualiserar 10000 dragningar överensstämmer bättre med verkligheten, vilket beror på att urvalsstorleken är större.

Uppgift 3.1.2 (1)

```
library(Rlab)
```

```
## Rlab 4.0 attached.
```

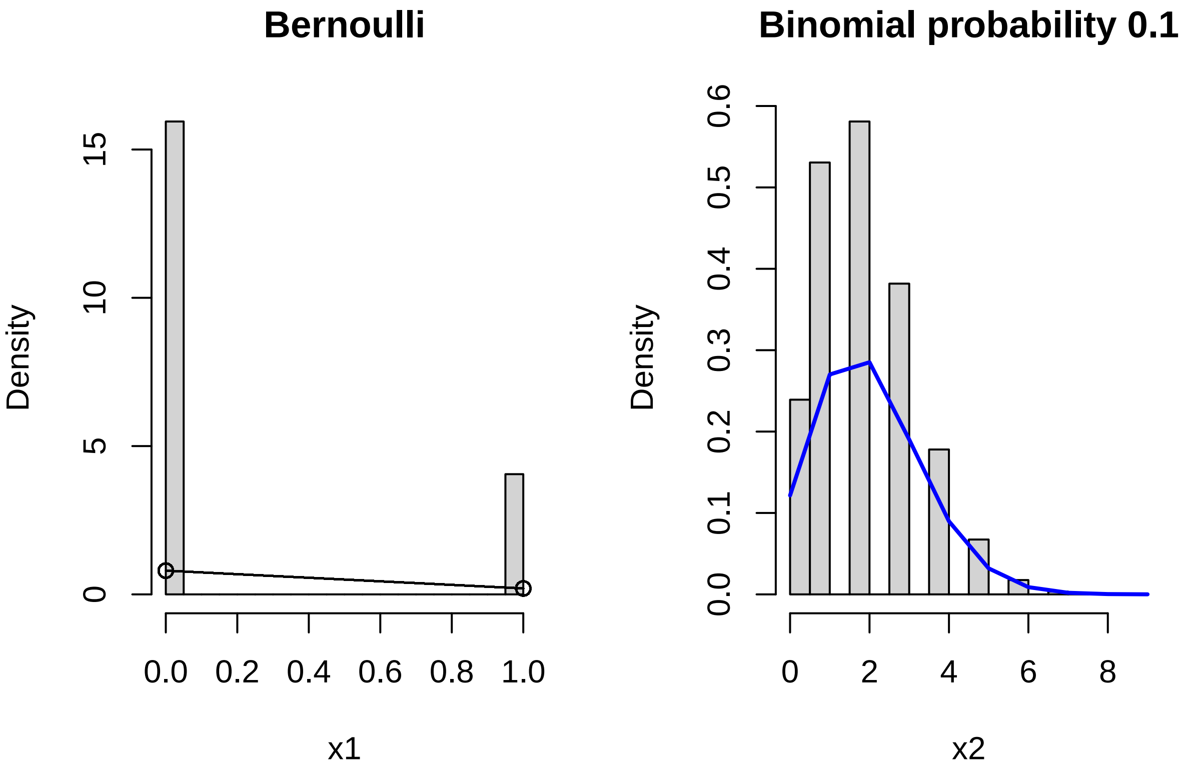
```
##
## Attaching package: 'Rlab'
```

```
## The following objects are masked from 'package:stats':
##
##     dexp, dgamma, dweibull, pexp, pgamma, pweibull, qexp, qgamma,
##     qweibull, rexp, rgamma, rweibull
```

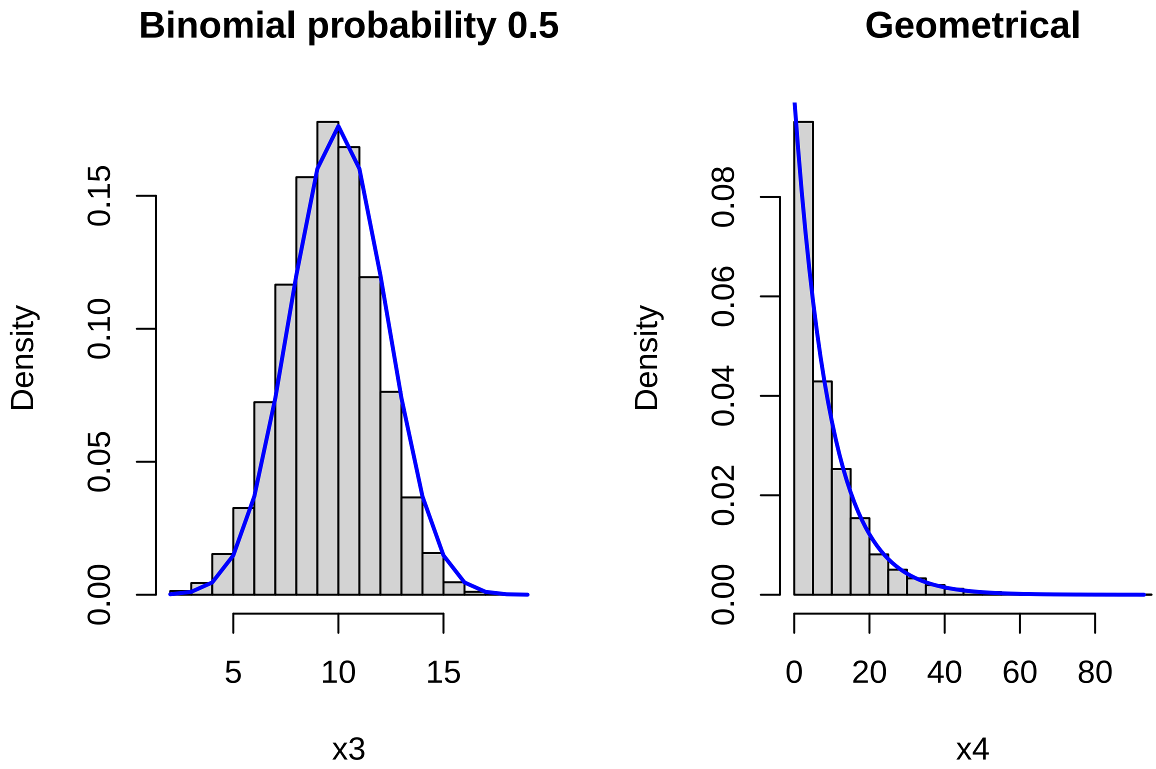
```
## The following object is masked from 'package:datasets':
##
##     precip
```

```
par(mfrow=c(1,2))
# DISCRETE DISTRIBUTIONS
pulls <- 10000
```

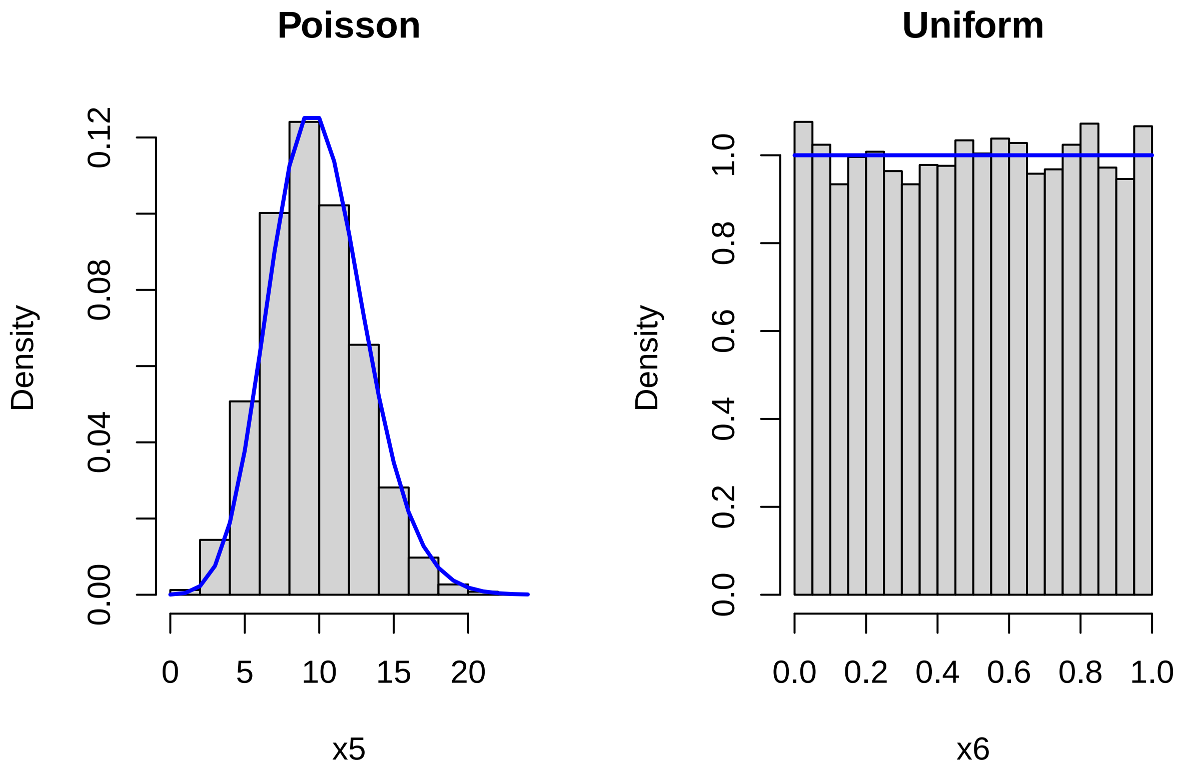
```r
x1 <- rbern(pulls, 0.2)
hist(x1, probability = TRUE, main = "Bernoulli")
y1 <- dbern(x1, 0.2, log = FALSE)
points(x1, y1, type = "o")

x2 <- rbinom(pulls, 20, 0.1)
hist(x2, probability = TRUE, main = "Binomial probability 0.1")
x2fit <- seq(min(x2), max(x2), 1)
y2 <- dbinom(x2fit, 20, 0.1)
lines(x2fit, y2, col="blue", lwd = 2)
```



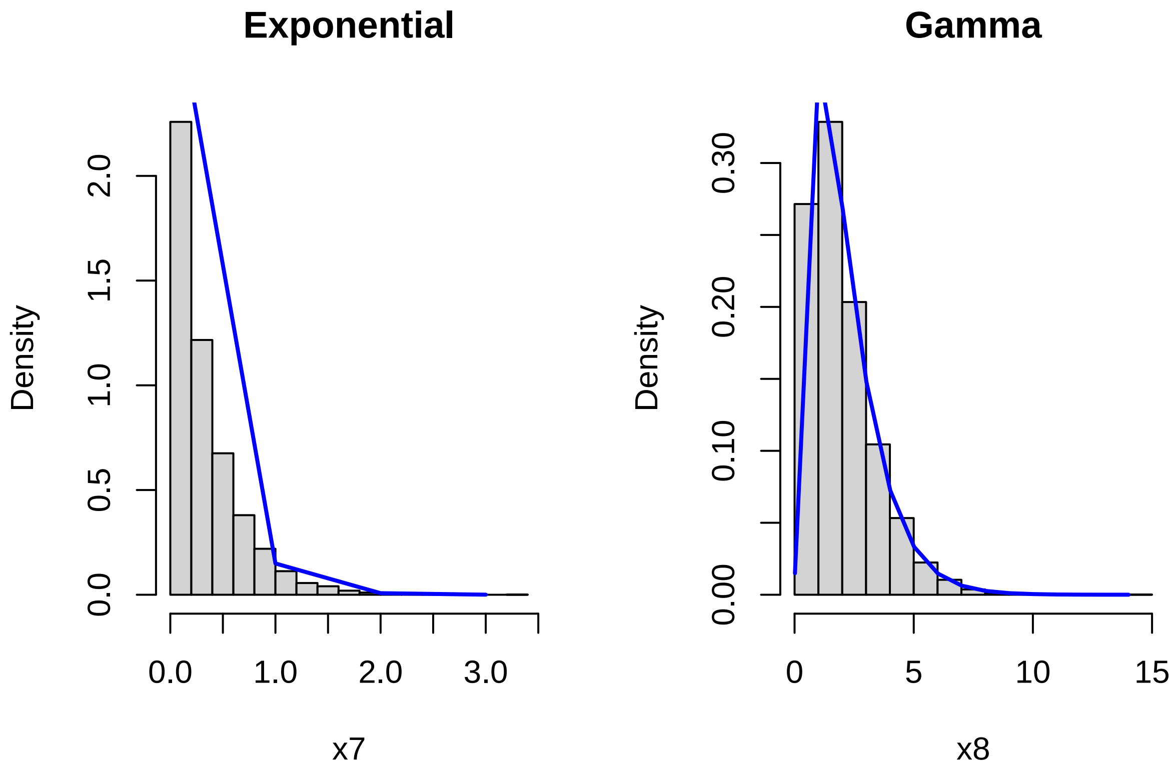
```r
x3 <-rbinom(pulls, 20, 0.5)
hist(x3, probability = TRUE, main = "Binomial probability 0.5")
x3fit <- seq(min(x3), max(x3), 1)
y3 <- dbinom(x3fit, 20, 0.5)
lines(x3fit, y3, col="blue", lwd=2)

x4 <- rgeom(pulls, 0.1)
hist(x4, probability = TRUE, main = "Geometrical")
x4fit <- seq(min(x4), max(x4), 1)
y4 <- dgeom(x4fit, 0.1, log=FALSE)
lines(x4fit, y4, col="blue", lwd=2)
```



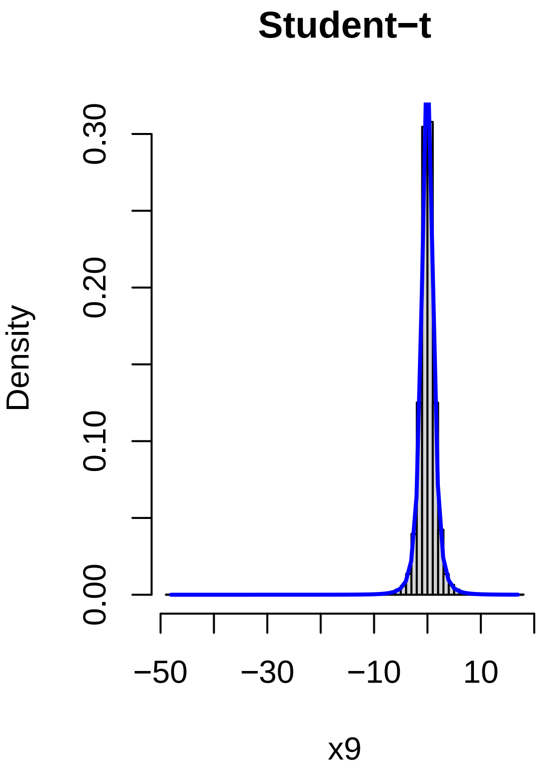
```
x5 <- rpois(pulls, 10)
hist(x5, probability = TRUE, main = "Poisson")
x5fit <- seq(min(x5), max(x5), 1)
y5 <- dpois(x5fit, 10, log=FALSE)
lines(x5fit, y5, col="blue", lwd=2)

# CONTINOUS DISTRIBUTIONS

x6 <- runif(pulls, 0, 1)
hist(x6, probability = TRUE, breaks = 20, main = "Uniform")
x6fit <- seq(0, 1, 1)
y6 <-dunif(x6fit, 0, 1, log=FALSE)
lines(x6fit, y6, col="blue", lwd=2)
```

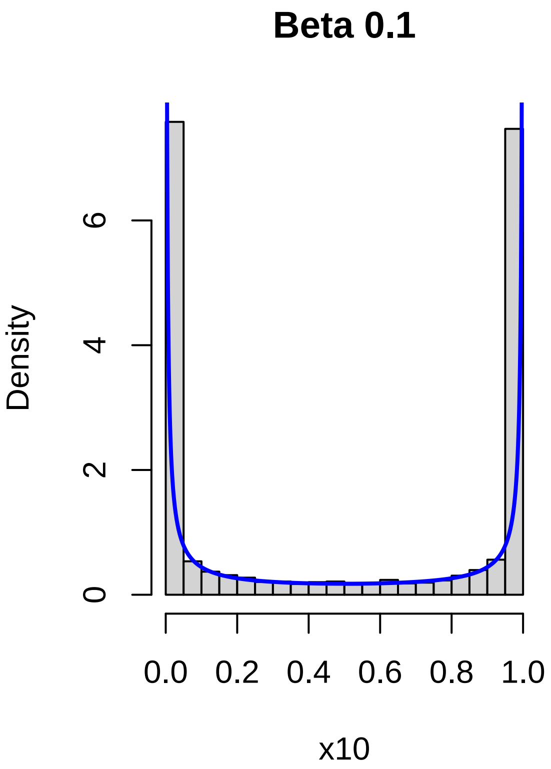


```
x7 <- rexp(pulls, 3)
hist(x7, probability = TRUE, breaks = 20, main = "Exponential")
x7fit <- seq(min(x7), max(x7), 1)
y7 <- dexp(x7fit, 3, log=FALSE)
lines(x7fit, y7, col="blue", lwd=2)

x8 <- rgamma(pulls, 2, 1)
hist(x8, probability = TRUE, main = "Gamma")
x8fit <- seq(min(x8), max(x8), 1)
y8 <- dgamma(x8fit, 2, 1)
lines(x8fit, y8, col="blue", lwd=2)
```

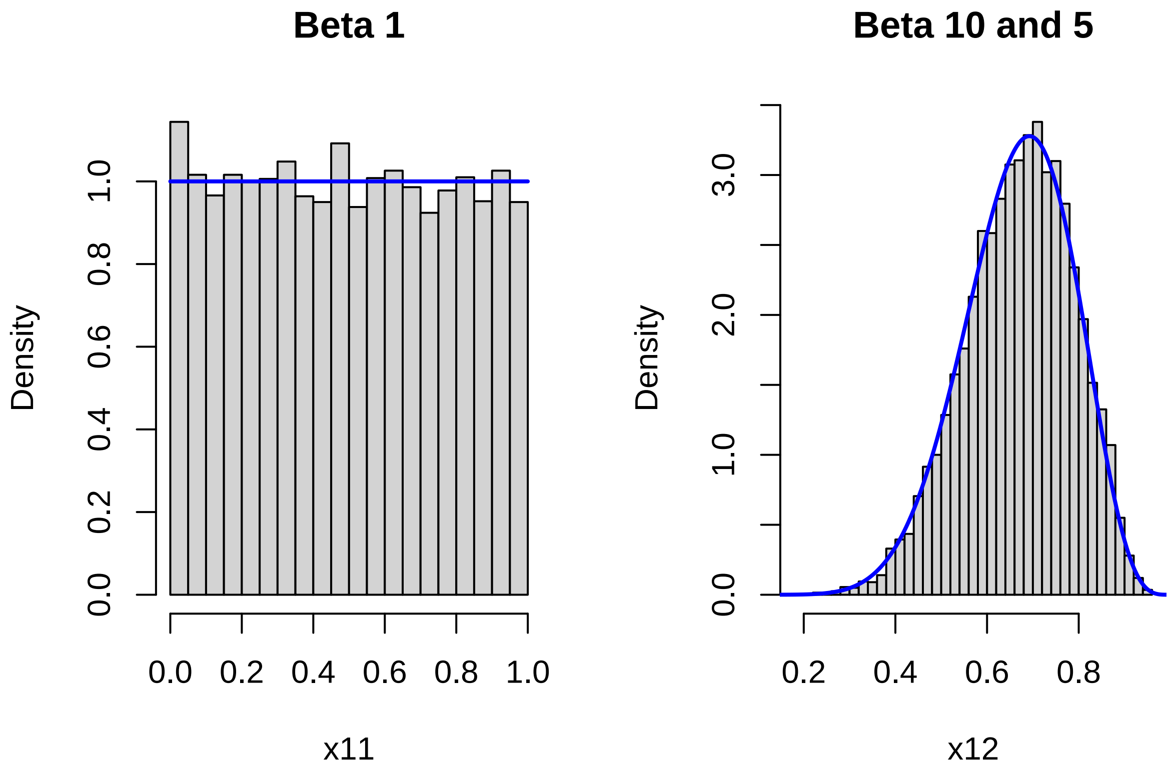


```
x9 <- rt(pulls, df=3)
hist(x9, probability = TRUE, breaks=50, main = "Student-t")
x9fit <- seq(min(x9), max(x9), 1)
y9 <- dt(x9fit, 3, log=FALSE)
lines(x9fit, y9, col="blue", lwd=2)

x10 <- rbeta(pulls, 0.1, 0.1)
hist(x10, probability = TRUE, main = "Beta 0.1")
x10fit <-seq(0, 1, 0.001)
y10 <- dbeta(x10fit, 0.1, 0.1, log=FALSE)
lines(x10fit, y10, col="blue", lwd=2)
```



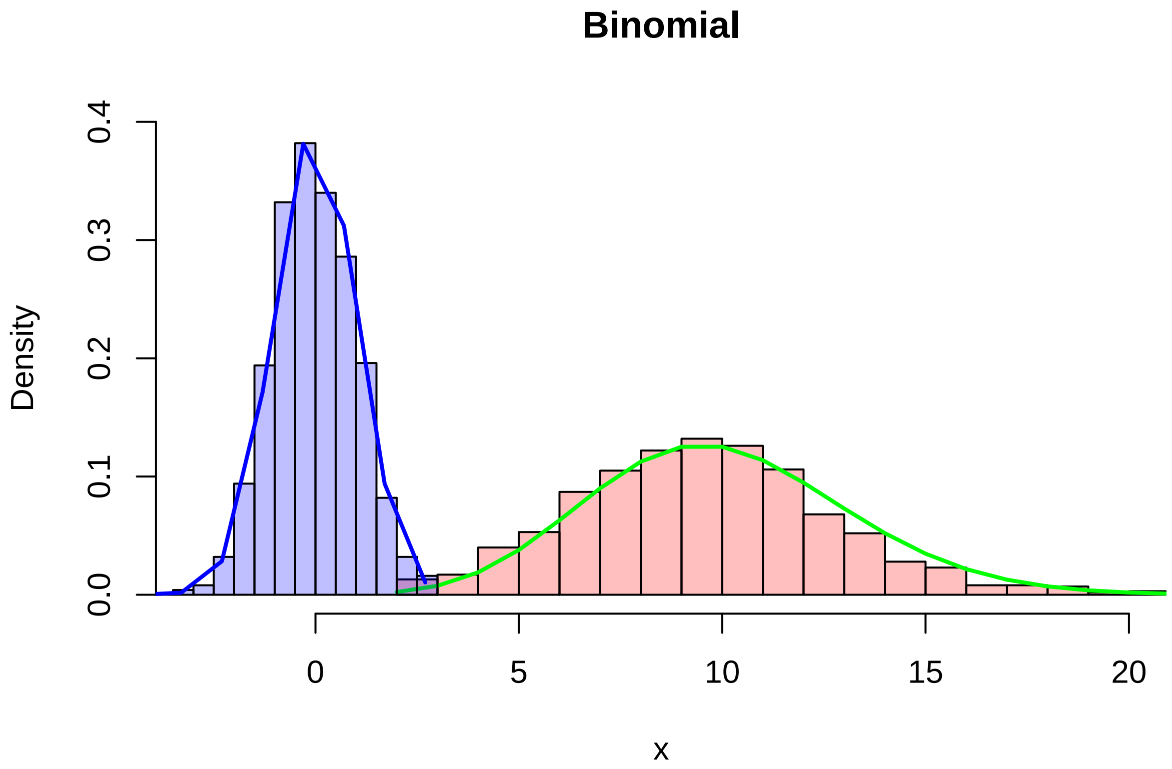


```
x11 <- rbeta(pulls, 1, 1)
hist(x11, probability = TRUE, main = "Beta 1")
x11fit <-seq(0, 1, 0.1)
y11 <- dbeta(x11fit, 1, 1, log=FALSE)
lines(x11fit, y11, col="blue", lwd=2)


x12 <- rbeta(pulls, 10, 5)
hist(x12, probability = TRUE, breaks=50, main = "Beta 10 and 5")
x12fit <-seq(0, 1, 0.001)
y12 <- dbeta(x12fit, 10, 5, log=FALSE)
lines(x12fit, y12, col="blue", lwd=2)
```



Uppgift 3.1.3 (1)

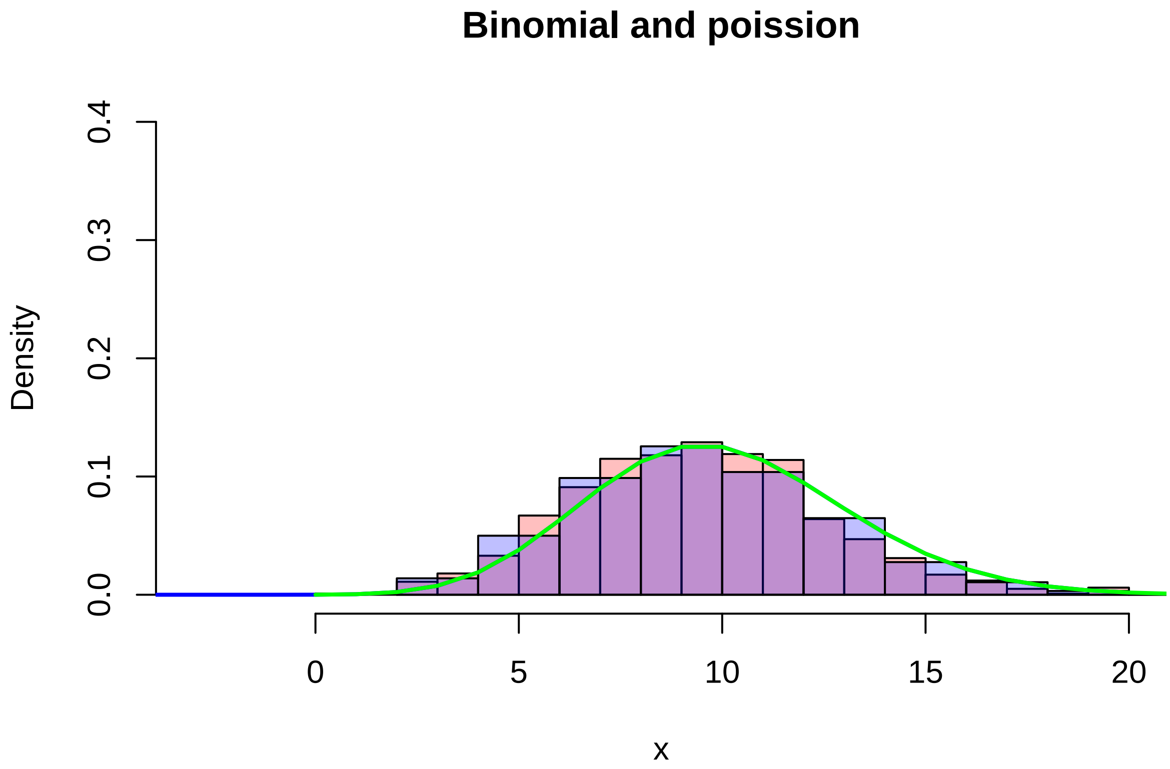
```
pulls <- 1000

x <- rbinom(pulls, 10000, 0.001)
hist(x, probability=TRUE, breaks=20, , xlim=c(-3,20), ylim=c(0,0.4), main="Binomial", col=rgb(1,0,0,1/4
xfit <- seq(min(x), max(x), 1)
y <- dbinom(xfit, 10000, 0.001)
lines(xfit, y, col="green", lwd=2)

x2 <- rt(pulls, 10000)
hist(x2, probability = TRUE, add=TRUE ,breaks=20, main = "Student-t", col=rgb(0,0,1,1/4))
x2fit <- seq(min(x2), max(x2), 1)
y2 <- dt(x2fit, 10000, log=FALSE)
lines(x2fit, y2, col="blue", lwd=2)
```



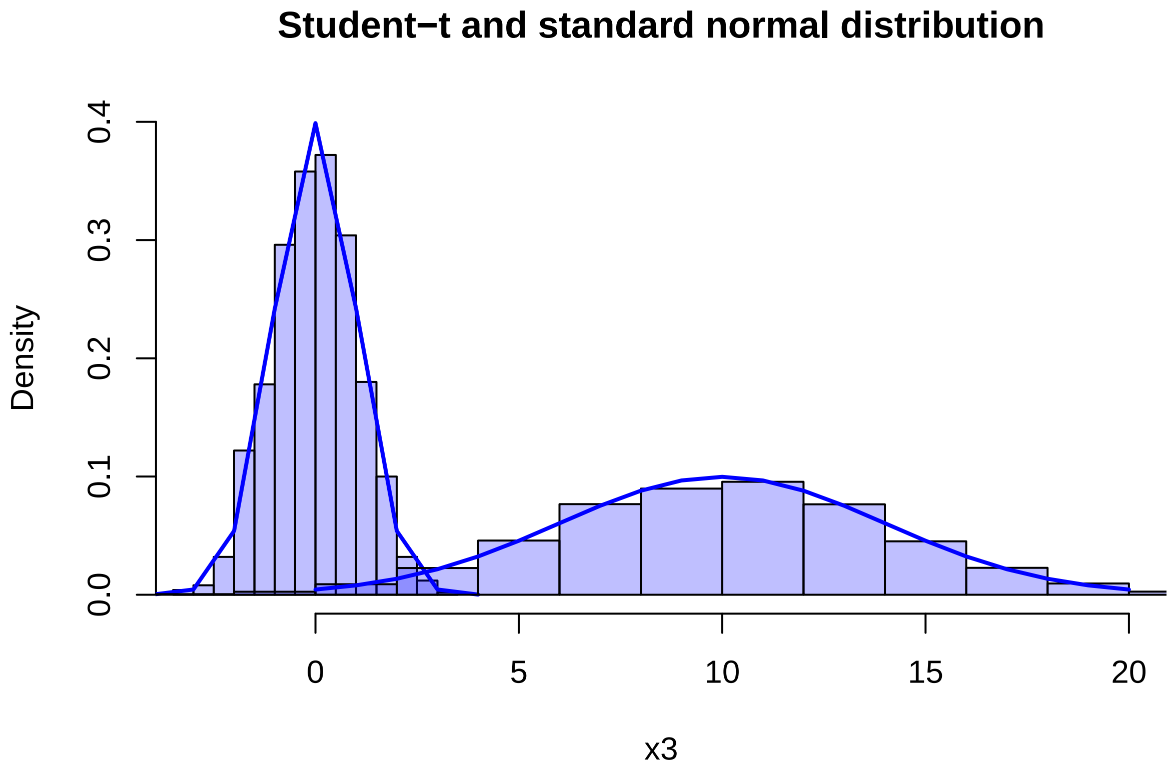
Uppgift 3.1.3 (2)
Binomial konvergerar mot poisson, student-t konvergerar mot standard normalfördelning.

Uppgift 3.1.3 (3)

```
#Binomial and poisson
x <- rbinom(n=1000, size=10000, prob=0.001)
hist(x, probability=TRUE, breaks=20, , xlim=c(-3,20), ylim=c(0,0.4), main="Binomial and poission", col=
xfit <- seq(-4, 20, 1)
y <- dbinom(xfit, size = 10000, prob=0.001)
lines(xfit, y, col="blue", lwd=2)

x2 <- rpois(10000, lambda=10)
hist(x2, probability = TRUE, add=TRUE, col=rgb(0,0,1,1/4))
x2fit <- seq(0, 30, 1)
y2 <- dpois(x2fit, lambda=10, log=FALSE)
lines(x2fit, y2, col="green", lwd=2)
```



```
#Student-t and standard normal distribution
x3 <- rt(1000, df=10000)
hist(x3, probability = TRUE ,breaks=20, xlim=c(-3,20), ylim=c(0,0.4), main = "Student-t and standard no
x3fit <- seq(-4, 4, 1)
y3 <- dt(x3fit, df=10000, log=FALSE)
lines(x3fit, y3, col="blue", lwd=2)

x4 <- rnorm(10000, 10, 4)
hist(x4, probability=TRUE, add=TRUE, col=rgb(0,0,1,1/4))
x4fit <- seq(0, 20, 1)
y4 <- dnorm(x4fit, 10, 4, log=FALSE)
lines(x4fit, y4, col="blue", lwd=2)
```



Uppgift 3.1.4 (1)

```
pulls <- 10000
PY0 <- dbinom(0,10, 0.1, log = FALSE)
x <- rbinom(pulls, 10, 0.1)

zeros=0
for (i in x){
  if (i==0)
    zeros = zeros+1
}
print(zeros)
```

```
## [1] 3562
```

Uppgift 3.1.4 (2)

```
#P(X < 0)
x <- pnorm(0)
x
```

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

```
#P(-1 < X < 1)
x2 <- pnorm(1) - pnorm(-1)
x2
```

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

```
#P(1.96 < X)
x3 <- pnorm(1.96, lower.tail=FALSE)
x3
```

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

```
#P(0 < Y < 10) pbinom first parameter "find prob of x or fewer"
x4 <- pbinom(9,10,0.1)- pbinom(0,10,0.1)
x4
```

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

```
#P(Y=0)
x5 <- pbinom(0, 10, 0.1)
x5
```

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

```
#P(0<= Y <= 10)
x6<-pbinom(10,10,0.1)
x6
```

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

Uppgift 3.1.4 (3)

```
#P(X<0)
x <- rnorm(10000, 0, 1)
length(x[x < 0])
```

```
## [1] 4990
```

```
#P(-1 X < 1)
x2 <- rnorm(10000, 0, 1)
length(x2[x2 < 1]) - length(x2[x2 < -1])
```

```
## [1] 6813
```

```
#P(1.96 < X)
x3 <- rnorm(10000, 0, 1)
10000 - length(x3[x3 < 1.96])
```

```
## [1] 253
```

```
#P(0 < Y < 10)
x4 <- rbinom(n=10000, size=10, prob=0.1)
count = 0
for ( i in x4){
    if( i != 0 && i != 10)
    count = count + 1
}
print(count)
```

```
## [1] 6583
```

```
#P(Y=0)
x5 <- rbinom(n=10000, size=10, prob=0.1)
count = 0
for ( i in x5){
    if( i == 0)
    count = count + 1
}
print(count)
```

```
## [1] 3484
```

```
#P(0 <= Y <= 10)
x <- rbinom(n=10000, size=10, prob=0.1)
count = 0
for ( i in x5){
    if(i >= 0 && i <= 10)
    count = count + 1
}
print(count)
```

```
## [1] 10000
```

Uppgift 3.1.5 (1, 2, 3)

```
#(1)
#Simulations
simold <- rbinom(10000, 337, 0.1)
simnew <- runif(10000, 0.02, 0.16)
n <- 337

#Calculate expected values
expectedold <- mean(simold)
meannew <- mean(simnew)
expectednew <- meannew * n

expectedold
```

```
## [1] 33.6718
```

```
expectednew
```

```
## [1] 30.39802
```

```
#(2)
new <- rbinom(10000, 337, meannew)
result<-(mean(new < simold))
result
```

```
## [1] 0.6385
```

```
#(3)
# Old
count = 0
for (i in simold) {
  if (i > 50){
    count = count + 1
  }
}
result = count / 10000
result
```

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

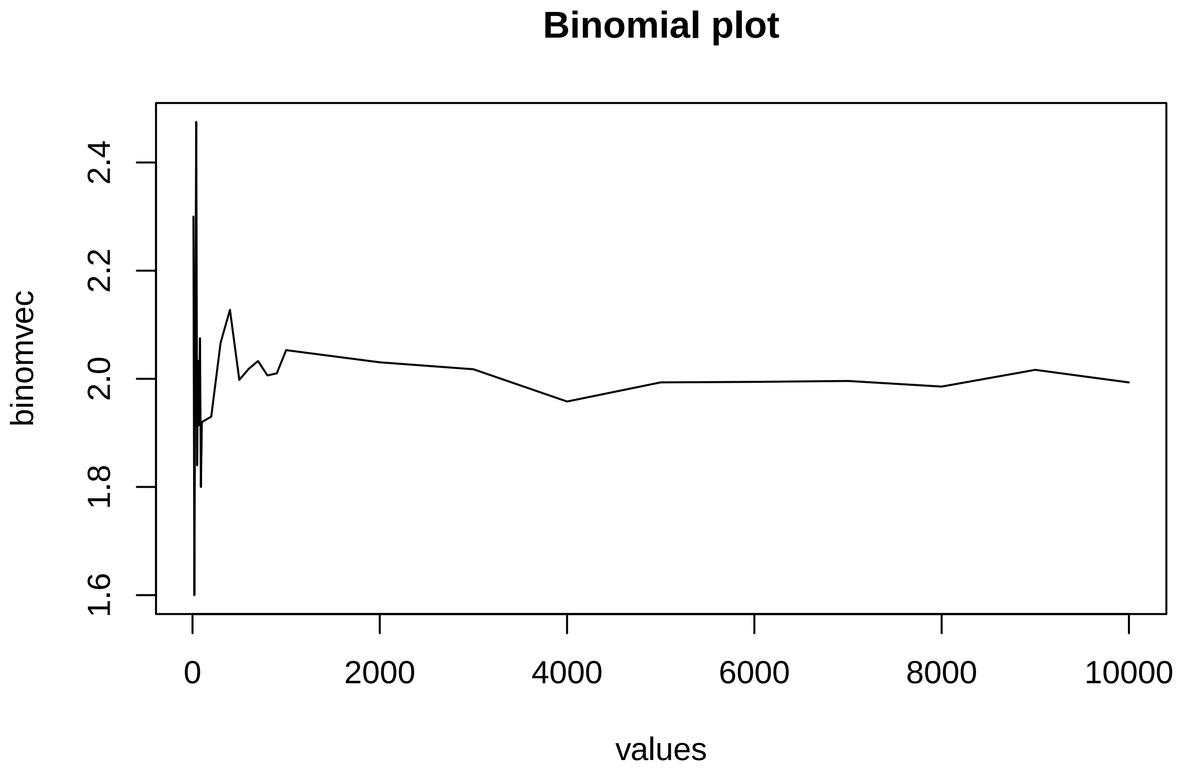
```
#New
count = 0
temp = 0
for (i in simnew){
  temp = i * 337
  if (temp > 50){
    count = count +1
  }
  temp = 0
}
result = count / 10000
result
```

```
## [1] 0.0867
```

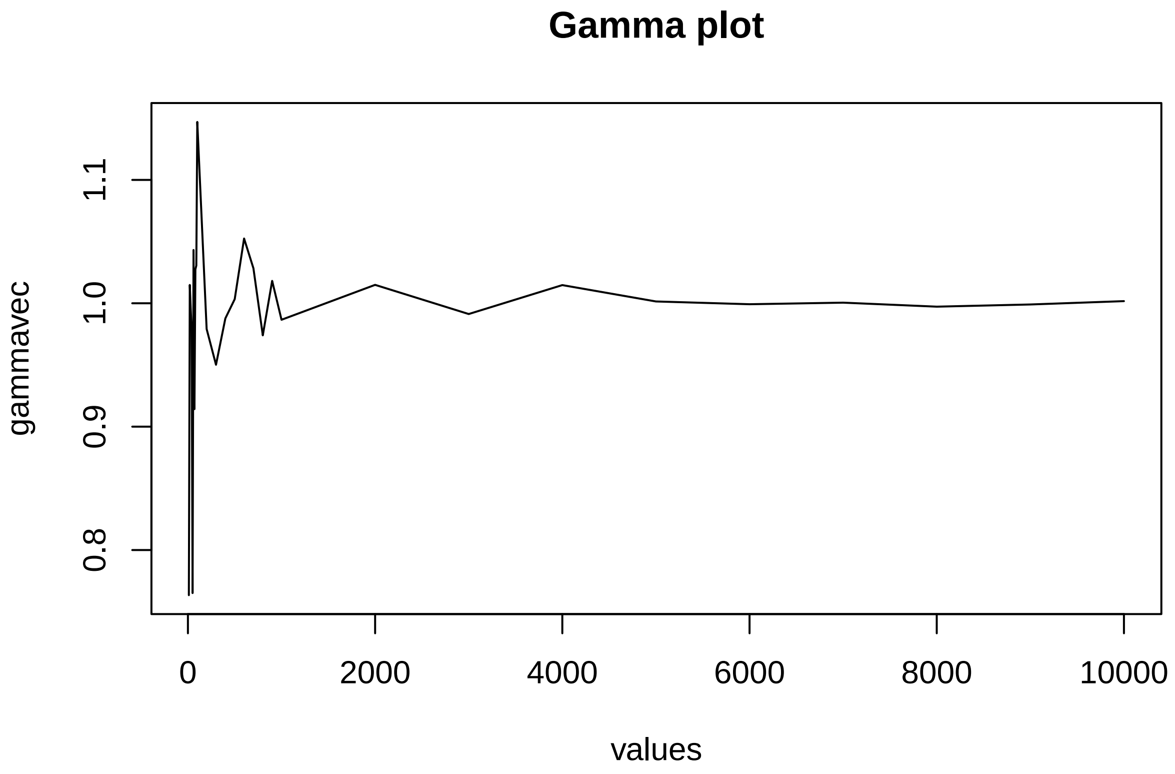
Uppgift 3.2.1 (1)
De teoretiska väntevärdena är:
Binomialfördelning: E(X) = n * p = 10 * 0.2 = 2
Gammafördelning: E(Y) = alpha / beta = 2 / 2 = 1

Uppgift 3.2.1 (2)

```
values = c()
for (i in 1:3) {
  for (j in 1:10) {
    values <- append(values, (j * 10^i))
  }
}
values <- values[!duplicated(values)]

binomvec = c()
for(value in values){
  x<-rbinom(value, 10, 0.2)
  mean<-mean(x)
  binomvec<-append(binomvec, mean)
}

gammavec = c()
for(value in values){
    y<-rgamma(value,2,2)
    mean<-mean(y)
    gammavec<-append(gammavec, mean)
}


plot(values, binomvec, type="l", main="Binomial plot")
```



```
plot(values, gammavec, type="l", main="Gamma plot")
```



Uppgift 3.3.1 (1)
De teoretiska väntevärdena är:
Exponentialfördelning: 1 / theta = 1 / 10 = 0.1
Poissionfördelning = lambda = 3

De teoretiska variansvärdena är:
Expontialfördelning: 1 / lambda² = 1 / 100 = 0.01
Poissingfördelning = E(Y) = lambda = 3

Uppgift 3.3.1 (2)

```
pulls <- 10000
exponential <- rexp(pulls,10)
expmean <- mean(exponential)
expvar <- var(exponential)

poisson <- rpois(pulls, 3)
poismean <- mean(poisson)
poisvar <- var(poisson)

cat("Expected mean of exponential distribution: ", expmean, sep="\n")
```

```
## Expected mean of exponential distribution:
```

```
## 0.10014
```

```
cat("Expected variance of exponential distribution: ", expvar, sep="\n")
```

```
## Expected variance of exponential distribution:
## 0.01005128
```

```
cat("Expected mean of poisson distribution: ", poismean, sep="\n")
```

```
## Expected mean of poisson distribution:
## 3.012
```

```
cat("Expected variance of poission distribution: ", poisvar, sep="\n")
```

```
## Expected variance of poission distribution:
## 3.03796
```

Uppgift 3.3.1 (3)

E(c) = c ger att:
E(3) = 3

E(aX + b) = a * E(X) + b, för konstant a, b, ger att:
E(3 * X + 2) = 3 * 0.1 + 2 = 2.3

E(X + Y) = E(X) + E(Y) ger att:
E(X + Y) = 0.1 + 3 = 3.1

E(X * Y) = E(X) * E(Y) (där X och Y är oberoende) ger att:
E(X * Y) = 0.1 * 3 = 0.3

E(aX + bY + c) = a * E(X) + b * E(Y) + c, för konstant a, b, c, ger att:
E(3 * X + 2 * Y - 3) = 3 * 0.1 + 2 * 3 - 3 = 3.3

Var(aX + b) = $a^2$ * Var(X) ger att:
Var(2 * x - 5) = $2^2$ * 0.01 = 0.04

Var(X + Y) = Var(X) + Var(Y) ger att:
Var(X + Y) = 0.01 + 3 = 3.01

Uppgift 3.3.1 (4)

```
x <- rexp(10000, 10)
y <- rpois(10000, 3)

#E(3)
mean(3)
```

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

```r
#E(3 * X + 2)
mean(3*x+2)
```

```
## [1] 2.299609
```

```r
#E(X+Y)
mean(x+y)
```

```
## [1] 3.09307
```

```r
#E(X*Y)
mean(x*y)
```

```
## [1] 0.299522
```

```r
#E(3*x +2Y -3)
mean(3*x+2*y-3)
```

```
## [1] 3.286009
```

```r
#Var(2*X -5)
var(2*x-5)
```

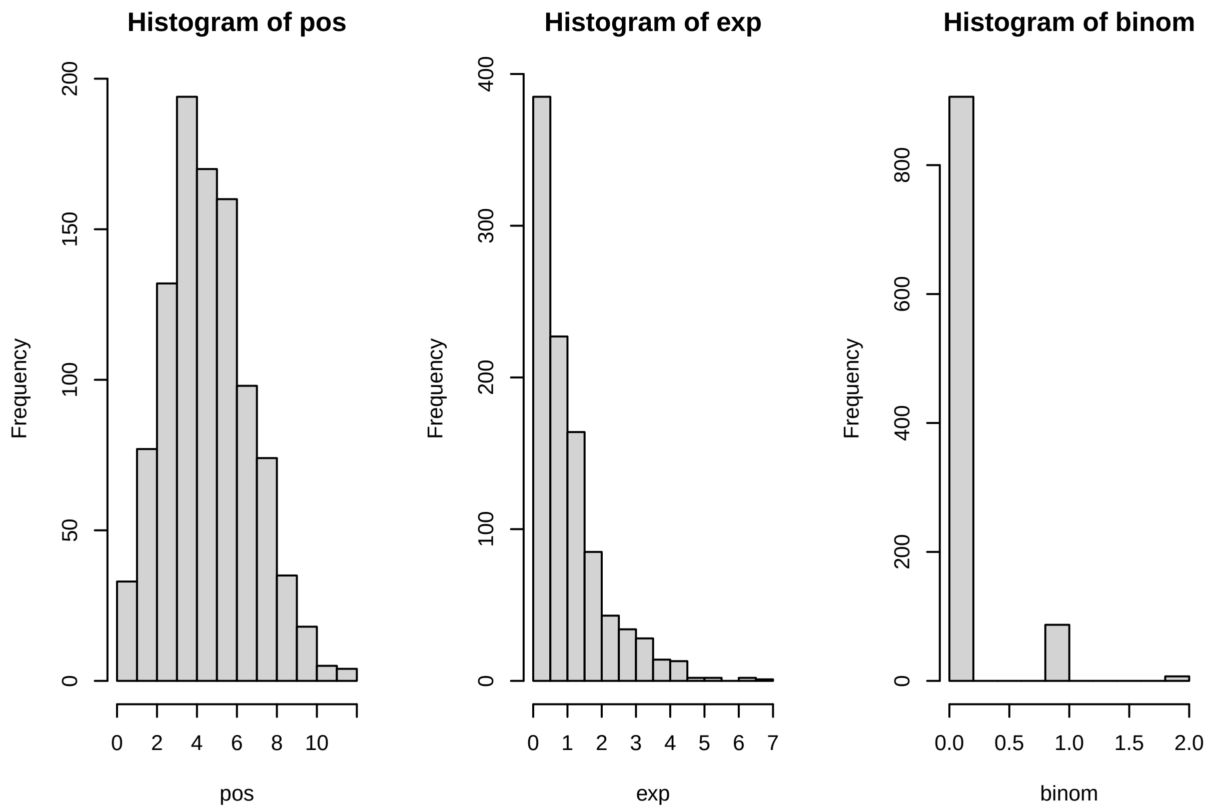
```
## [1] 0.03953831
```

```r
#Var(X+Y)
var(x+y)
```

```
## [1] 2.949317
```

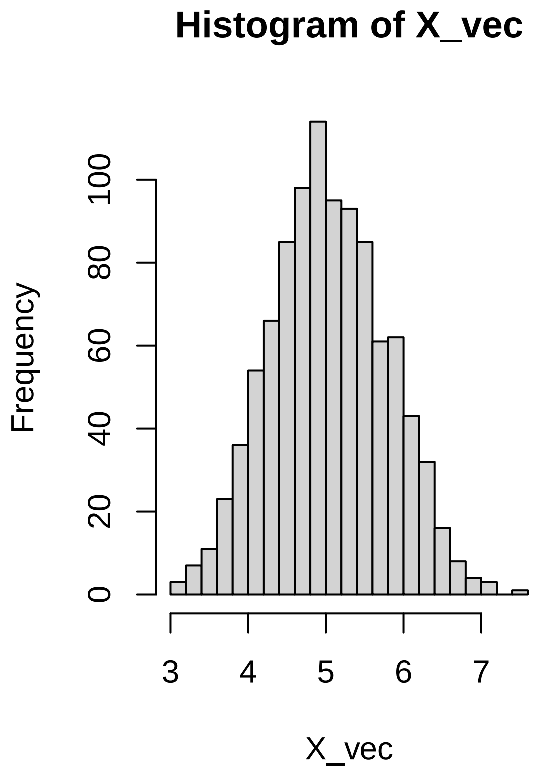
Uppgift 3.4.1 (1)

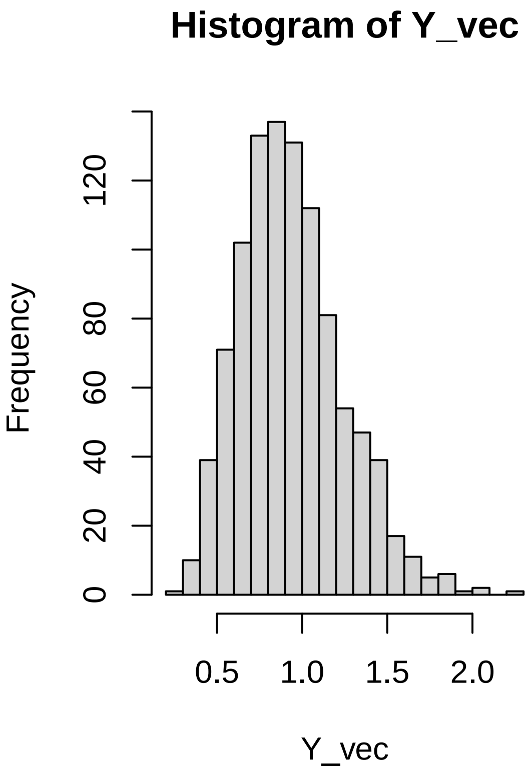
```r
par(mfrow=c(1,3))

pos <- rpois(1000, 5)
exp <- rexp(1000, 1)
binom <- rbinom(1000, 10, 0.01)

hist(pos)
hist(exp)
hist(binom)
```



Uppgift 3.4.1 (2)

```r
par(mfrow=c(1,2))

X <- rpois(1000, 5)
Y <- rexp(1000, 1)

X_vec <- c()
Y_vec <- c()

for(i in 1:1000) {
  x_sample <- sample(X, 10)
  y_sample <- sample(Y, 10)

  x_mean <- mean(x_sample)
  y_mean <- mean(y_sample)

  X_vec <- append(X_vec, x_mean)
  Y_vec <- append(Y_vec, y_mean)
}

hist(X_vec, breaks = 20)
hist(Y_vec, breaks = 20)
```

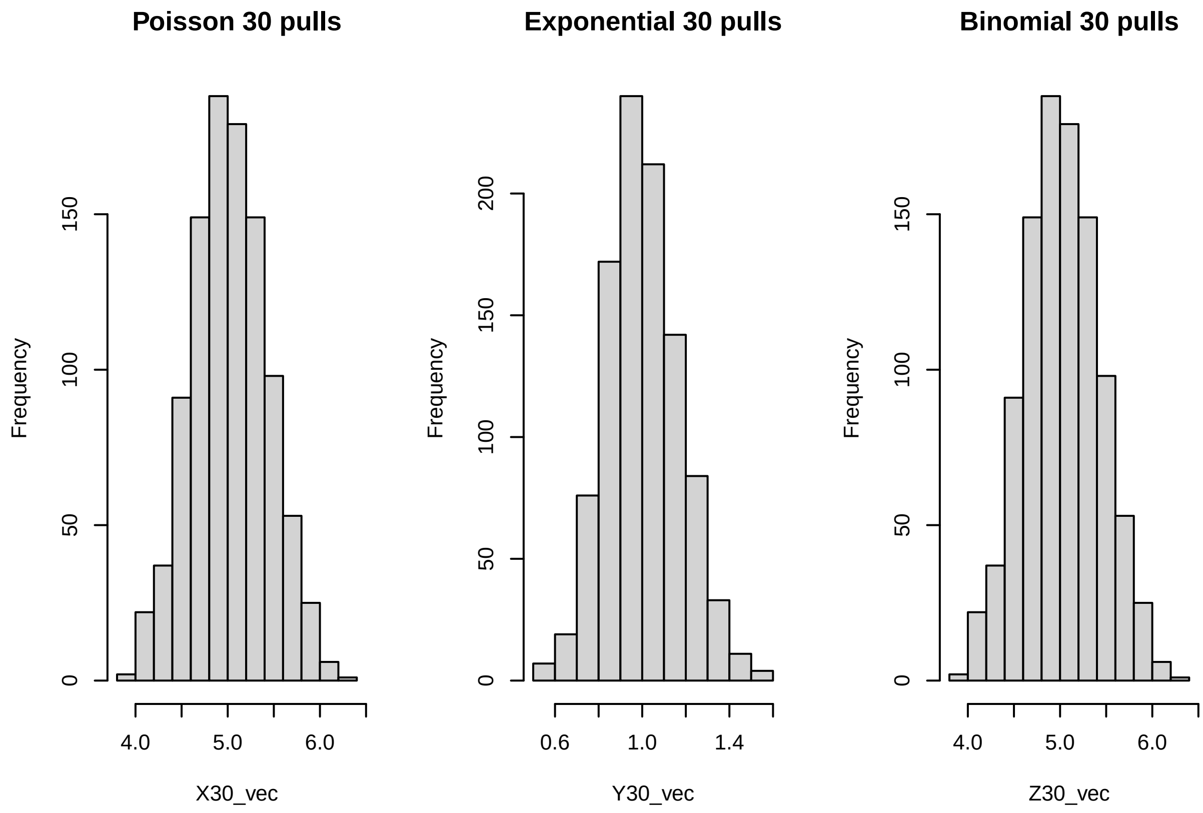




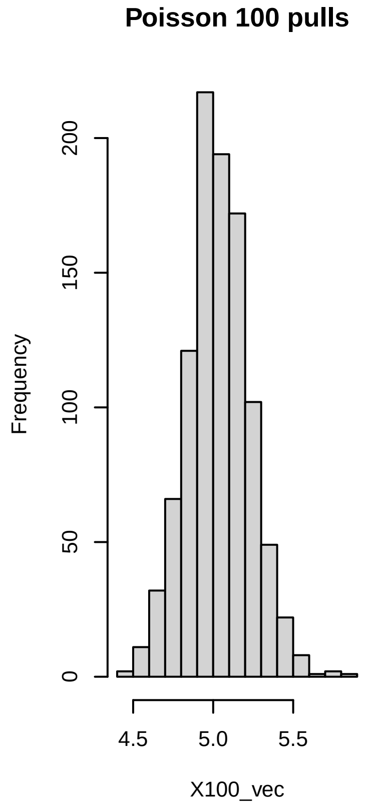
Uppgift 3.4.1 (3)

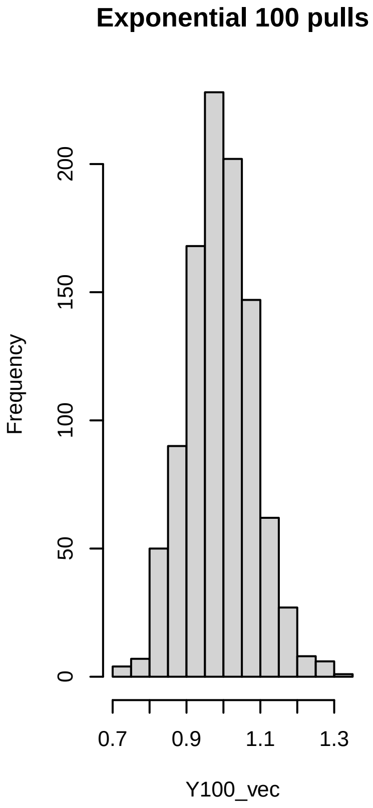
```
par(mfrow=c(1, 3))

X <- rpois(1000, 5)
Y <- rexp(1000, 1)
Z <- rbinom(1000, 10, 0.01)

# 30 pulls
X30_vec <- c()
Y30_vec <- c()
Z30_vec <- c()

for(i in 1:1000) {
  x30_sample <- sample(X, 30)
  y30_sample <- sample(Y, 30)
  z30_sample <- sample(Z, 20)

  x30_mean <- mean(x30_sample)
  y30_mean <- mean(y30_sample)
  z30_mean <- mean(x30_sample)

  X30_vec <- append(X30_vec, x30_mean)
  Y30_vec <- append(Y30_vec, y30_mean)
  Z30_vec <- append(Z30_vec, z30_mean)
}
```

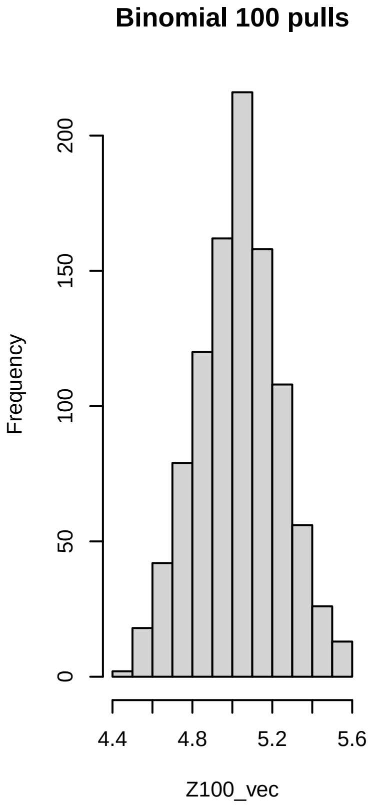
```r
hist(X30_vec, main = "Poisson 30 pulls")
hist(Y30_vec, main = "Exponential 30 pulls")
hist(Z30_vec, main = "Binomial 30 pulls")
```



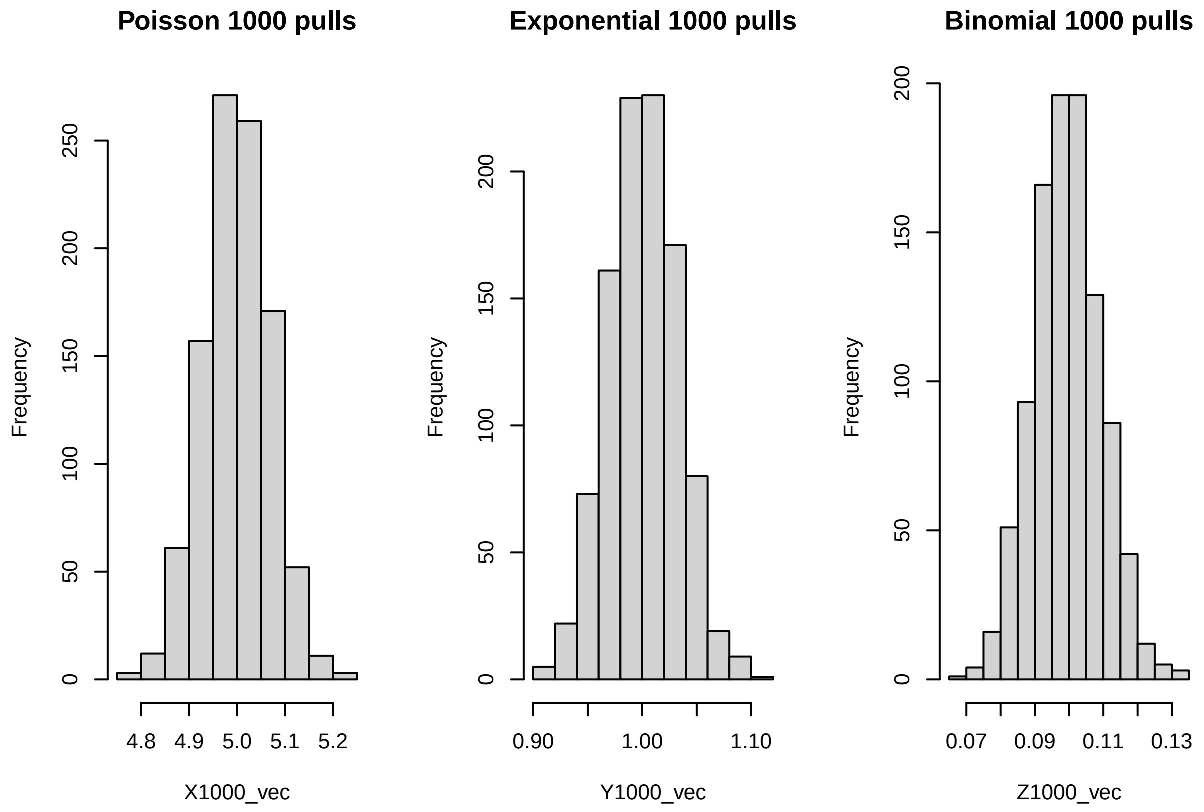
```r
# 100 pulls
X100_vec <- c()
Y100_vec <- c()
Z100_vec <- c()

for(i in 1:1000) {
  x100_sample <- sample(X, 100)
  y100_sample <- sample(Y, 100)
  z100_sample <- sample(X, 100)


  x100_mean <- mean(x100_sample)
  y100_mean <- mean(y100_sample)
  z100_mean <- mean(z100_sample)

  X100_vec <- append(X100_vec, x100_mean)
  Y100_vec <- append(Y100_vec, y100_mean)
  Z100_vec <- append(Z100_vec, z100_mean)
}

hist(X100_vec, main = "Poisson 100 pulls")
hist(Y100_vec, main = "Exponential 100 pulls")
hist(Z100_vec, main = "Binomial 100 pulls")
```







```
# 1000 pulls

X1000_vec <- c()
Y1000_vec <- c()
Z1000_vec <- c()

for(i in 1:1000) {
  X <- rpois(1000, 5)
  Y <- rexp(1000, 1)
  Z <- rbinom(1000, 10, 0.01)

  x1000_mean <- mean(X)
  y1000_mean <- mean(Y)
  z1000_mean <- mean(Z)

  X1000_vec <- append(X1000_vec, x1000_mean)
  Y1000_vec <- append(Y1000_vec, y1000_mean)
  Z1000_vec <- append(Z1000_vec, z1000_mean)
}

hist(X1000_vec, main = "Poisson 1000 pulls")
hist(Y1000_vec, main = "Exponential 1000 pulls")
hist(Z1000_vec, main = "Binomial 1000 pulls")
```



Medelvärdena konvergerar mot en normalfördelning.
Z konvergerar snabbare mot normalfördelningen för att binomialfördelningen har lägst standardavvikelse (av de tre fördelningarna).