Probability Distribution and Simulation.

Kingsuk_Jana

2023-03-28

Probability Density Function.

f(X)>=0 for all x and integral of f(x) over the range of x is equal 1.

Support: The set of values at which the PDF of PMF has positive value.

X ~ Binomial(n,p), Support(X) = {0,1,2,3..n}

X ~ Poisson(lambda), Support(X) = {0,1,2. . . .}

X ~ Normal(mu, sigma^2), Support(X) = {-Inf, +Inf}

X ~ Exponential(lambda), Support(X)= {0, +Inf}

###1.Continuous distribution: Exponential distribution

```
# How can we check whether a given exponential function is pdf
lamda = 2
Exponential_fun = function(x){
   2*exp( -2*x )
}
print(Exponential_fun)
```

```
## function(x){
##    2*exp( -2*x )
## }
```

```
# In this case the support is {0,+Inf}. We need to integrate the function over
# the range "0" to "Inf" and show that the integral is one.

out = integrate(Exponential_fun, lower = 0, upper = Inf)
print(out)
```

```
## 1 with absolute error < 5e-07
```

```r
value = out$value
print(value)
```

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

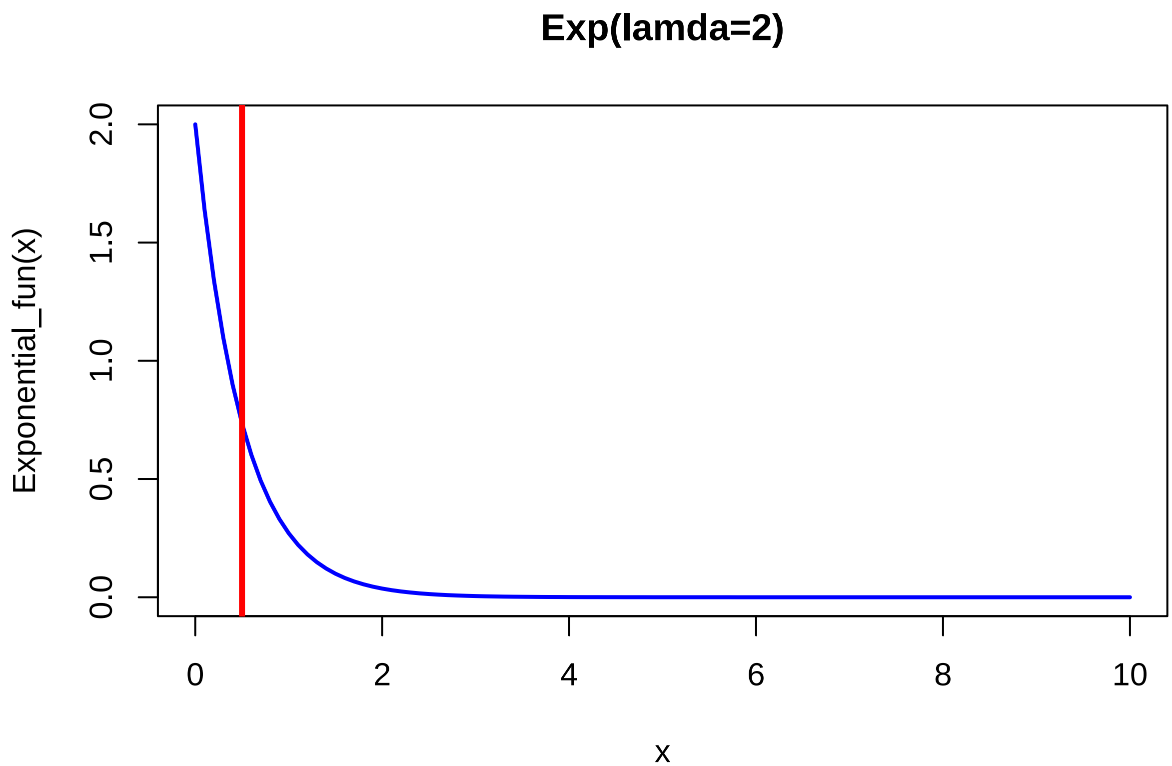
```r
# Since the value is 1, so the density is a pdf.

curve(Exponential_fun(x), col ="blue", lwd=2, main="Exp(lamda=2)", 0, 10)
# We introduced two parameters "lwd' and "col",
# "lwd" stands for line width and "col" stands for line colur.

# How can we find the population mean(E(X)) and population variance(var(X)) of a random variable x?
# E(X)= Integral x*f(x)
# Var(X)=E(X-E(X))^2= E(x^2)-(E(x))^2

## Mean and variance of exponential distribution

# a. Let's calculate the population mean= E(X), It is also called first order row moment.

mean1_f1 = function(x){
    x*2*exp(-2*x)}

print(mean1_f1)
```

```
## function(x){
##     x*2*exp(-2*x)}
```

```r
mean = integrate(mean1_f1, 0, Inf)$value
print(mean)
```

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

```r
# Locate the mean in the density curve.
abline(v = mean, col = "red", lwd = 3, lty= 1)
```




```r
# b. Now we calculate the E(x^2), it is called second order row moment.

mean2_f1 = function(x){
  x^2*mean1_f1(x)
}
print(mean2_f1)
```

```
## function(x){
##   x^2*mean1_f1(x)
## }
```

```r
mean2 = integrate(mean2_f1, 0, Inf)$value
print(mean2)
```

```
## [1] 0.75
```

```r
#c. Population variance V(x) of the random variable X.

var = mean2 - (mean)^2  # where, var=var(x), mean1= E(x), mean2=E(x^2)
print(var)
```

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

```
sqrt(var)  # Standard deviation.
```

```
## [1] 0.7071068
```

###2. Continuous distribution: Normal distribution

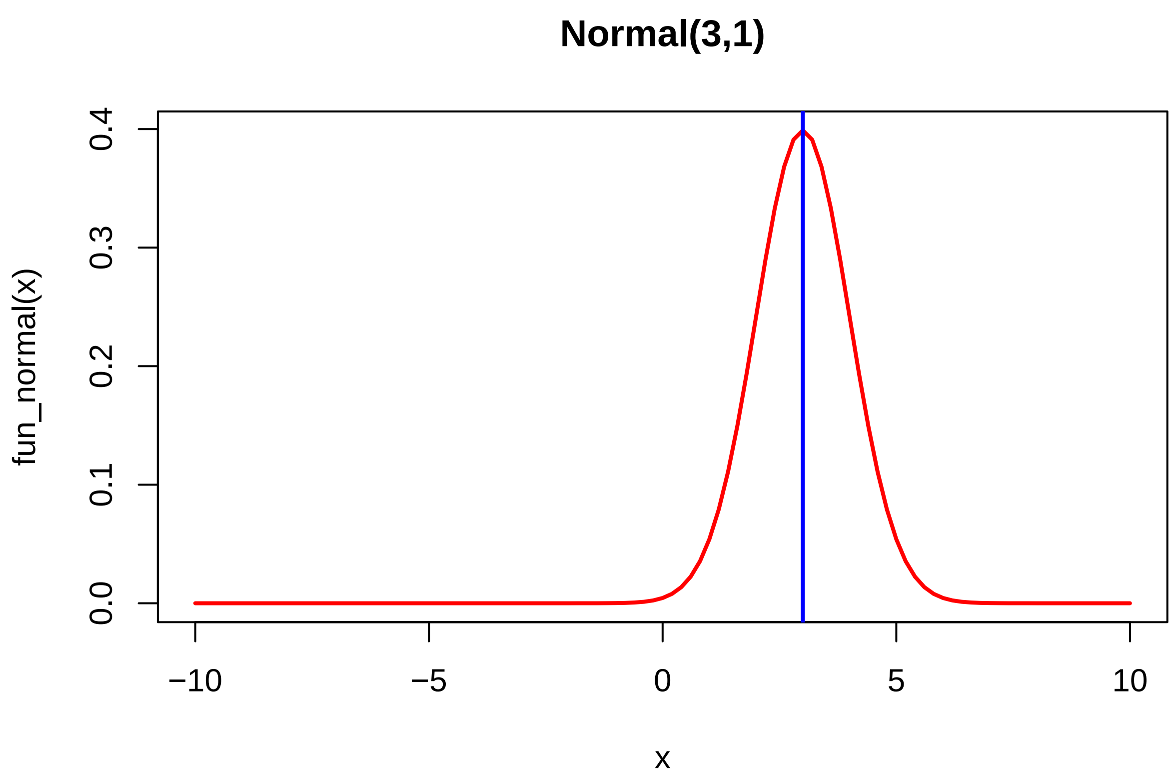
```
# Probability density function
mu=3
sigma=1
fun_normal = function(x){
    (1/(sqrt(2*pi)*sigma))*exp((-(x - mu)^2/2*sigma^2))
}
print(fun_normal)
```

```
## function(x){
##     (1/(sqrt(2*pi)*sigma))*exp((-(x - mu)^2/2*sigma^2))
## }
```

```
out_put = integrate(fun_normal, -Inf, +Inf)$value
print(out_put)
```

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

```
curve(fun_normal(x), -10, 10, col = "red", lwd=2, main = "Normal(3,1)")
# Plot the normal density function over the range -10 to 10.
abline(v = mu, col = "blue", lwd = 2 )
```



```r
##Mean and variance of normal distribution:

#a. Population mean=E(X), it is also called first order row moment.

fun_mean1_normal = function(x){
  x*fun_normal(x)
}
print(fun_mean1_normal)
```

```
## function(x){
##    x*fun_normal(x)
## }
```

```r
mean1_normal = integrate(fun_mean1_normal, -Inf, +Inf)$value
print(mean1_normal)
```

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

```r
#b.  E(X^2), second order row moment.

fun_mean2_normal = function(x){
  x^2*fun_normal(x)
}
print(fun_mean2_normal)
```

```
## function(x){
##    x^2*fun_normal(x)
## }
```

```r
mean2_normal = integrate(fun_mean2_normal, -Inf, +Inf)$value
print(mean2_normal)
```

```
## [1] 10
```

```r
#c. Population variance of X,  Var(x)=E(x^2)-(E(x))^2.

var=mean2_normal- (mean1_normal)^2
print(var)
```

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

```r
sqrt(var)  # Standard deviation.
```

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

###3. Discrete distribution: Binomial distribution

```r
# Binomial distribution has two parameters(n, p).
n = 10 # Parameter 1
p = 0.5 # Parameter 2
Binom_fun = function(x){
  choose(10, x)*p^(x)*(1 - p)^(10 - x) }
print(Binom_fun)
```

```
## function(x){
##   choose(10, x)*p^(x)*(1 - p)^(10 - x) }
```

```r
x=1:n
Binom_prob = Binom_fun(x)
Binom_prob
```

```
##  [1] 0.0097656250 0.0439453125 0.1171875000 0.2050781250 0.2460937500
##  [6] 0.2050781250 0.1171875000 0.0439453125 0.0097656250 0.0009765625
```

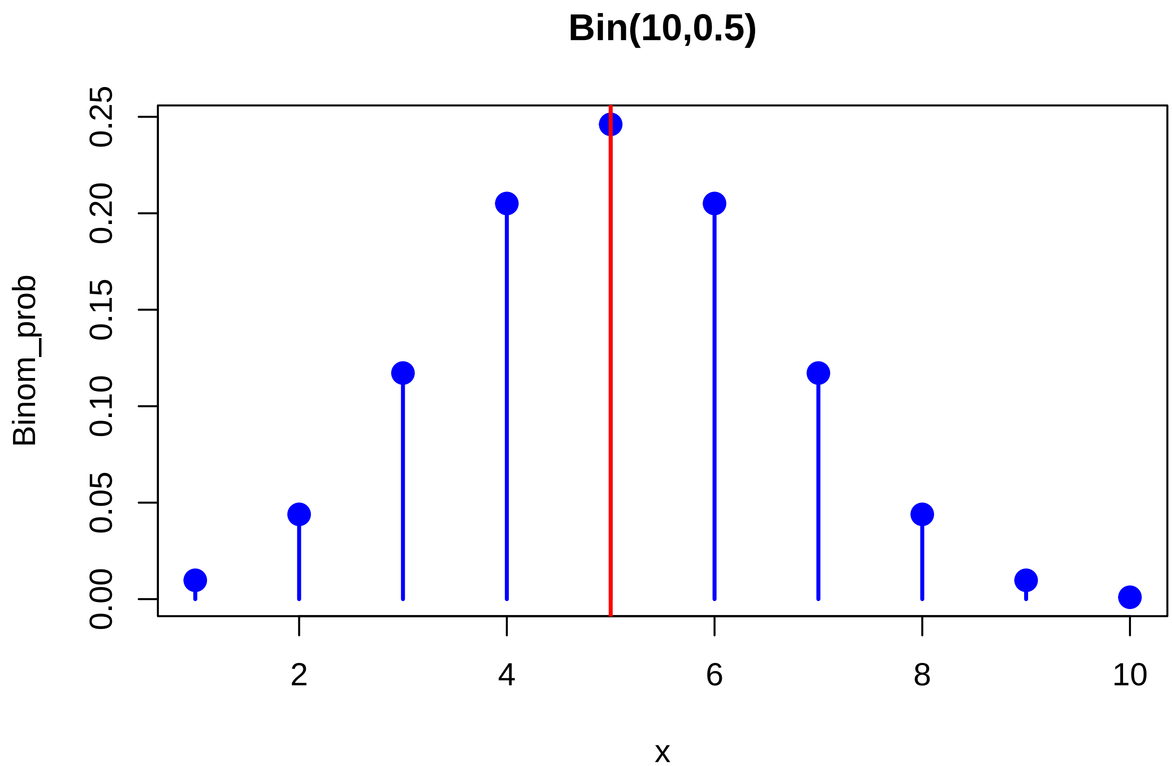
```r
plot(x,Binom_prob, type="h", col="blue", main = "Bin(10,0.5)", lwd=2)
points(x,Binom_prob, pch=19, cex=1.5, col="blue")


# Expectation, E(x)
Binom_Expec = function(x){
  x*choose(10, x)*p^(x)*(1 - p)^(10 - x)}
print(Binom_Expec)
```

```
## function(x){
##   x*choose(10, x)*p^(x)*(1 - p)^(10 - x)}
```

```r
y=1:n
Binom_Expec_cal=Binom_Expec(y)
sum(Binom_Expec_cal) # Value of the expectation.
```

```
## [1] 5
```

```r
abline(v=5, col="red", lwd=2)
```



###4.Discrete distribution: Poisson Distribution

```
Poisson_lambda = 10
Poisson_fun = function(x){
  exp( - Poisson_lambda )*( Poisson_lambda )^x/factorial(x)
}
Poisson_prob=Poisson_fun(x)
print(Poisson_prob)
```

```
##  [1] 0.0004539993 0.0022699965 0.0075666550 0.0189166374 0.0378332748
##  [6] 0.0630554580 0.0900792257 0.1125990321 0.1251100357 0.1251100357
```

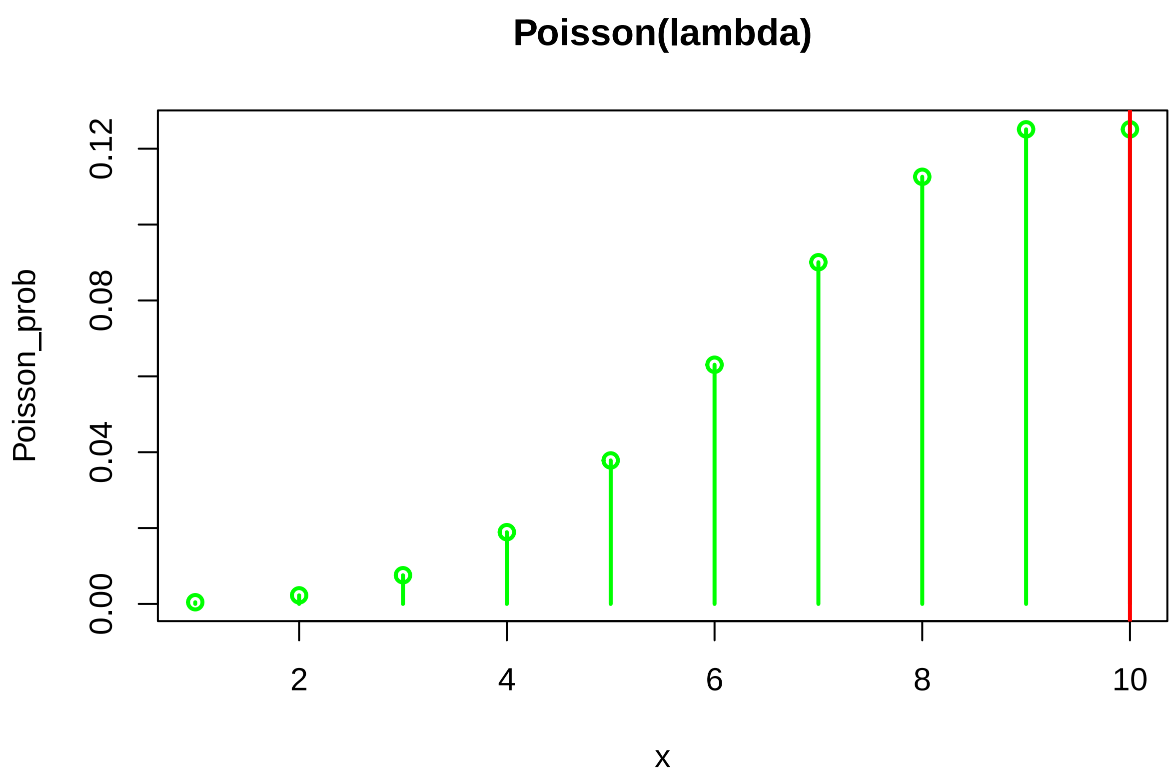
```
plot(x,Poisson_prob, type="h",col="green", main = "Poisson(lambda)", lwd=2)
points(x,Poisson_prob,col = "green", pch=1.9,lwd=2 )

# Expectation, E(x)
Poisson_Expec = function(x){
x*exp( - Poisson_lambda )*( Poisson_lambda )^x/factorial(x)}
print(Poisson_Expec)
```

```
## function(x){
## x*exp( - Poisson_lambda )*( Poisson_lambda )^x/factorial(x)}
```

```r
y=1:20
Poisson_Expec_cal = Poisson_Expec(y)
sum(Poisson_Expec_cal) # Value of the expectation.
```

```
## [1] 9.965457
```

```r
abline(v=10, col="red", lwd=2)
```

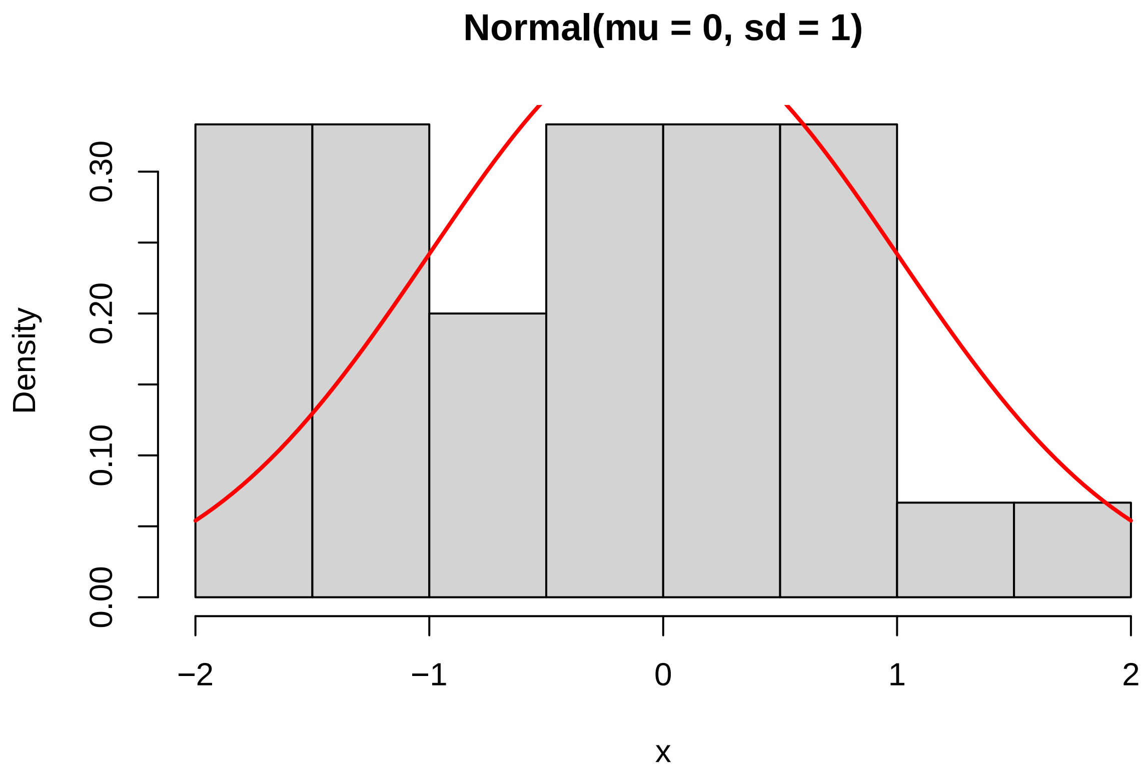


Some inbulit function in R

```r
# The four important symbols  are 'r', 'd', 'p', 'p'.
x = round(rnorm(n = 30, mean=0, sd = 1), 2)
print(x)
```

```
##  [1]  0.53 -0.24  0.53  1.48 -1.83 -1.78 -0.79 -0.53 -1.47 -1.21 -0.10 -1.57
## [13] -0.60 -1.92  0.17 -1.29  0.94  0.38 -0.04  1.95 -0.16 -1.05  0.21 -1.24
## [25]  0.66  0.94 -1.86  0.22 -0.02  0.01
```

```r
hist(x, main = "Normal(mu = 0, sd = 1)", probability = TRUE)
curve(dnorm(x), add = TRUE, col="red", lwd = 2)
```




```r
pnorm(2, mean = 0, sd = 1, lower.tail = TRUE)
```

```
## [1] 0.9772499
```

```r
qnorm(0.9772, mean = 0, sd = 1)
```

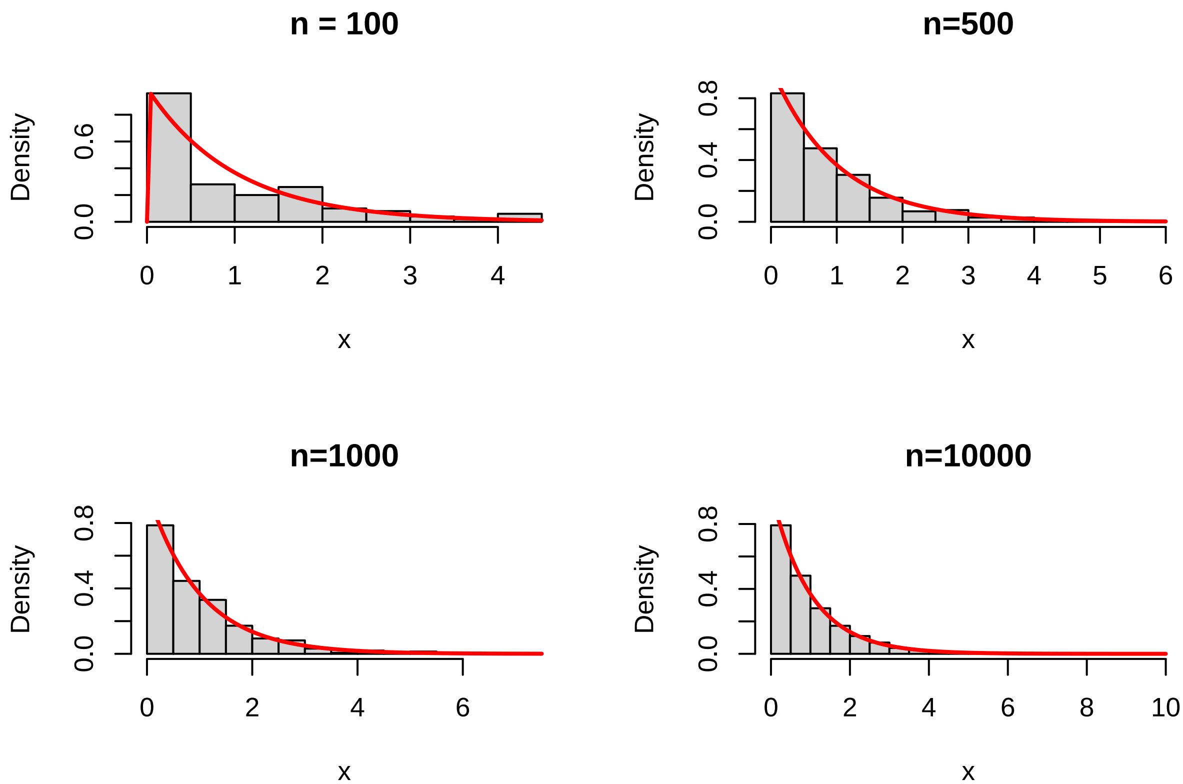
```
## [1] 1.999077
```

```r
#  "rnorm" function is used to simulate random sample from normal distribution with
# mu = 0 and standard deviation = 1.
# "dnorm" function for probability density function.
# "pnorm" for cumulative distribution function.
# "qnorm" quantile function.
```

Sample to population

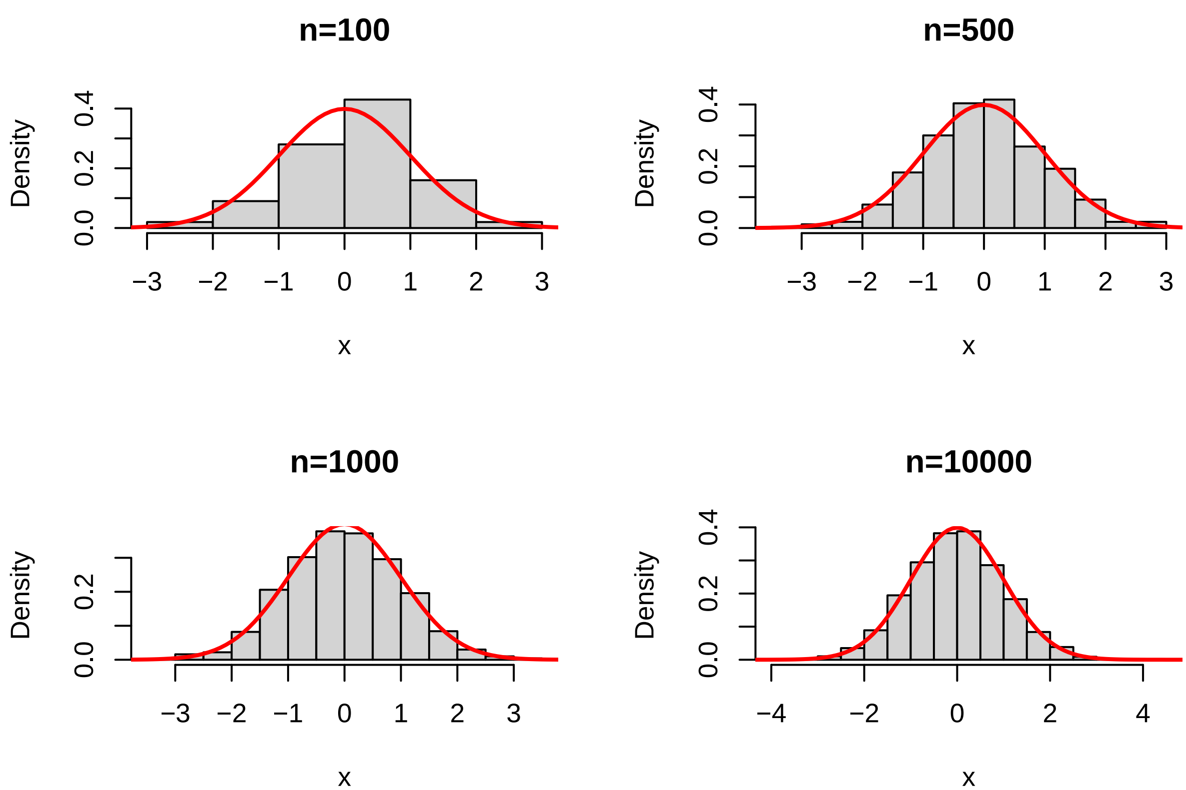
Understanding distribution of data (histogram) and its connection to the underlying population density function.As the sample size increases the histograms are getting closer to the population density function. More and more you are collecting data from a particular population more and more accurate your histogram will be to the population density function.

```r
par(mfrow = c(2, 2))
# The function "Par(mfrow)" is used to arranging the multiple plot in one plotting space.

## Exponential distribution:

# Histogram of sample of size(n) =100

x = rexp(n = 100, rate = 1)  # Draw sample from the exponential distribution, size=100
print(x)
hist(x, probability = TRUE, main = "n = 100")
curve(dexp(x, rate = 1), add = TRUE, col ="red", lwd = 2)

# Histogram of sample of size (n)=500

x = rexp(n = 500, rate = 1)   # Draw sample from the exponential distribution, size=500
print(x)
hist(x, probability = TRUE, main = "n=500")
curve(dexp(x, rate = 1),add = TRUE, col="red", lwd = 2)

#Histogram of sample of size (n) = 1000

x= rexp(n = 1000, rate = 1) # Draw sample from the exponential distribution, size=1000
print(x)
hist(x, probability = TRUE, main = "n=1000")
curve(dexp(x, rate = 1), add = TRUE, col="red", lwd=2)

#Histogram of sample of size (n)=10000

x = rexp(n = 10000, rate = 1) # Draw sample from the exponential distribution, size=10000
print(x)
hist(x, probability = TRUE, main = "n=10000")
curve(dexp(x, rate = 1), add = TRUE, col = "red", lwd = 2)
```




```
# From above four histograms we can conclude that as the sample size increases the histogram are gettin
# probability density curve of exponential distribution.


##Normal distribution(Standard normal distribution)

par(mfrow = c(2, 2))

# Histogram of sample of size(n)=100

x = rnorm(n = 100, mean = 0, sd = 1)    # Draw sample from the normal distribution, size=100
hist(x, probability = TRUE, main= "n=100")
curve(dnorm(x, mean = 0, sd = 1), add = TRUE, col = "red", lwd = 2, -5, 5)



# Histogram of sample of size(n)=500

x = rnorm(n = 500, mean = 0, sd = 1)    # Draw sample from the normal distribution, size=200
hist(x, probability = TRUE, main = "n=500")
curve(dnorm(x, mean= 0, sd = 1), add = TRUE, col = "red", lwd = 2, -5 ,5 )

#Histogram of sample of size(n)=1000

x = rnorm(n = 1000, mean = 0, sd = 1)    # Draw sample from the normal distribution, size=500
hist(x, probability = TRUE, main = "n=1000")
```

```r
curve(dnorm(x, mean = 0, sd = 1), add = TRUE, col = "red", lwd = 2, -5, 5)

# Histogram of sample of size(n)=10000

x = rnorm(n = 10000, mean = 0, sd = 1)  # Draw sample from the normal distribution, size=10000
hist(x, probability = TRUE, main = "n=10000")
curve(dnorm(x, mean = 0, sd = 1), add = TRUE, col = "red", lwd = 2, -5, 5)
```




```r
# From the above four histogram we can conclude as the sample size increases the histogram getting clos
#the normal distribution.
```