PRACTICAL 2 R

BDATC01/1717/2022

2024-11-14

```r
#1.Define the coefficients of the polynomial
#x^5+3x^4+((4x^2)/5)+3x+1
coefficients <- c(1, 3, 0, 4/5, 3, 1)  # Coefficients for x^5, x^4, x^3, x^2, x, constant term

# Find the roots using the polyroot function
roots <- polyroot(coefficients)

# Print the roots
print(roots)
```

```
## [1] -0.3344753+0.0000000i -1.2588390-0.0000000i  0.5565643-0.7954688i
## [4]  0.5565643+0.7954688i -2.5198143+0.0000000i
```

```r
library(rootSolve)
# Define the function
f<-function(x) {
  x^3 + 4*x^2 + x + 6
}
# Define the first derivative
f_prime <-function(x) {
  3*x^2 + 8*x + 1
}
# Define the second derivative
f_double_prime <-function(x) {
  6*x + 8
}
# Find the roots of the first derivative
stationary_points <-uniroot.all(f_prime, c(-10, 10))
# Determine the nature of each stationary point
nature_of_points <-sapply(stationary_points,
                          function(x) {
                            if (f_double_prime(x) > 0) {
                              return("Minimum")
                            } else if (f_double_prime(x) < 0) {
                              return("Maximum")
                            } else {
                              return("Inflection")
                            }
                          })
# Output the results
stationary_points
```

```
## [1] -2.5350850 -0.1314719
```

```r
nature_of_points
```

```
## [1] "Maximum" "Minimum"
```

```r
# Load necessary library
library(Deriv)
library(pracma)
```

```
##
## Attaching package: 'pracma'

## The following objects are masked from 'package:rootSolve':
##
##     gradient, hessian
```

```r
cot <- function(x) 1 / tan(x)
# Define the function f(x)
f <- function(x) {
  2*x * cot(x) + x * sin(x^2) + exp(3*x^3) + x^2 + 4
}

# i) Differentiate the function with respect to x
# Use the Deriv package to differentiate the function
library(Deriv)
f_prime <- Deriv(f, "x")

print(f_prime)
```

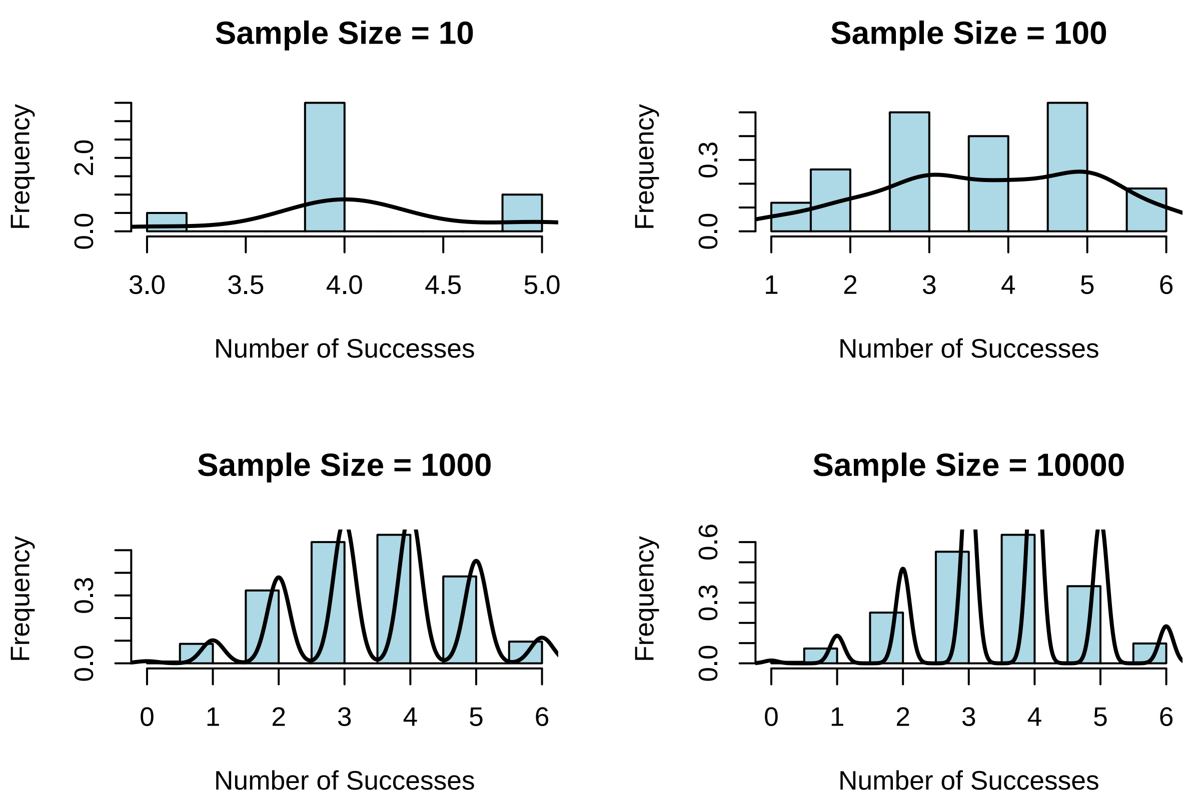
```
## function (x)
## {
##     .e1 <- x^2
##     2 * cot(x) + sin(.e1) + x * (2 + x * (2 * cos(.e1) + 9 *
##         exp(3 * x^3)) - 2/(cos(x)^2 * tan(x)^2))
## }
```

```r
# ii) Integrate the function over the range 3 < x < 6
# Use integrate function for numerical integration
integral_result <- integrate(f, lower = 3, upper = 6)
print(integral_result$value)
```

```
## [1] 8.179504e+278
```

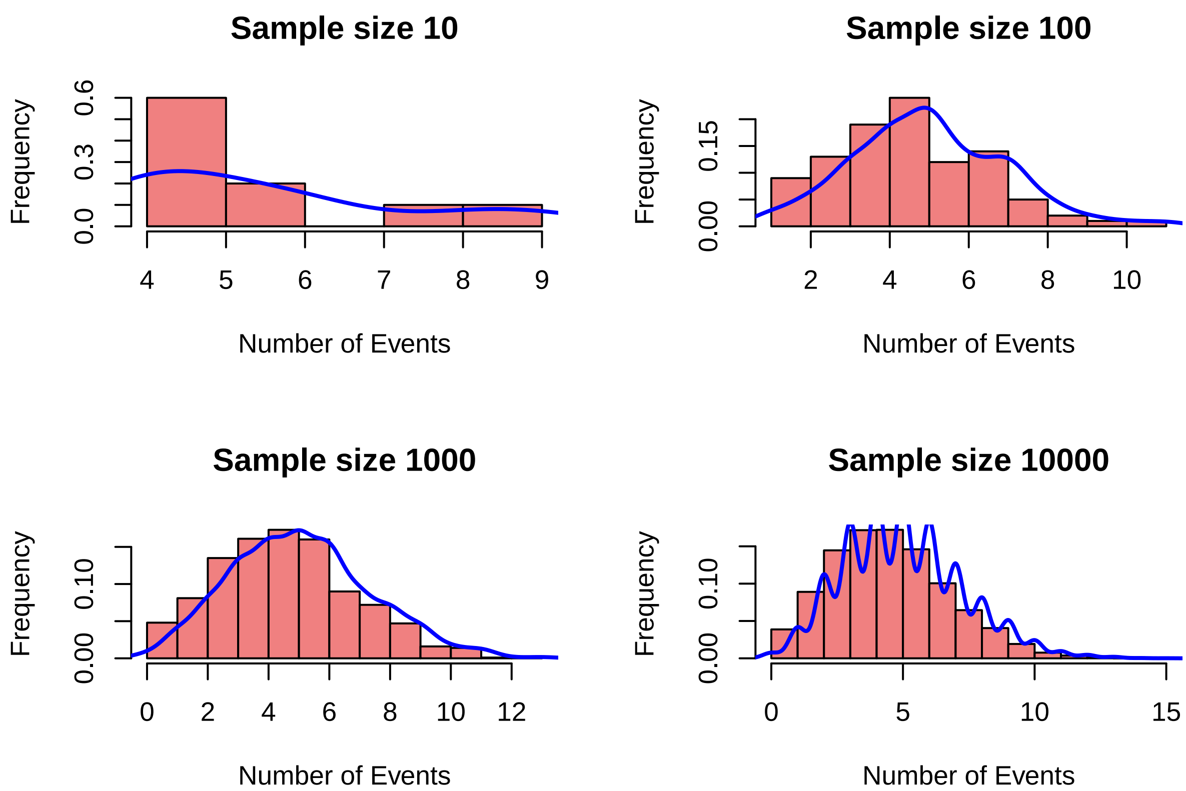
```r
#4
#Binomial distribution
# Define parameters for the binomial distribution
n<- 6        # Number of trials in the binomial distribution
p <- 0.6        # Probability of success in each trial

# Sample sizes for the simulations
sample_sizes <- c(10, 100, 1000, 10000)

# Create a 2x2 plotting grid to display all the histograms on the same plot
par(mfrow=c(2, 2))

# Loop over each sample size
for (sample_size in sample_sizes) {
```

```
  # Simulate binomial observations
  data <- rbinom(sample_size, n, p)

  # Plot histogram for the current sample size
  hist(data,
       main = paste("Sample Size =", sample_size),    # Title with sample size
       xlab = "Number of Successes",                   # X-axis label
       ylab = "Frequency",                             # Y-axis label
       col = "lightblue",                             # Histogram color
       border = "black",                              # Border color
       probability = TRUE,                            # Normalize the histogram to show density
       breaks = 10)                                   # Number of bins

  # Add density line to the histogram
  lines(density(data), col = "black", lwd = 2)        # Density line in red with thicker line width
}
```



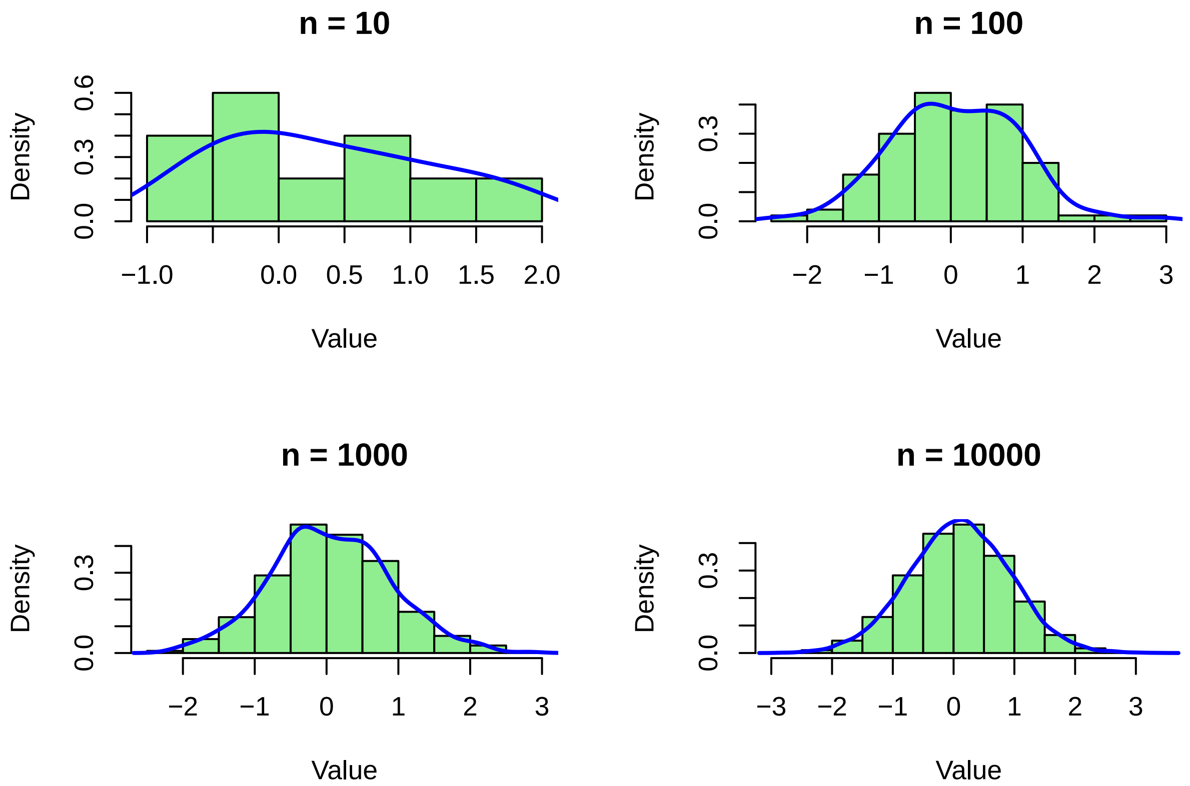

```
#Poisson distribution
# Simulate data from a Poisson distribution with lambda = 5 for different sample sizes
p1 <- rpois(n = 10, lambda = 5)      # Sample of 10 from Poisson distribution
p2 <- rpois(n = 100, lambda = 5)     # Sample of 100 from Poisson distribution
p3 <- rpois(n = 1000, lambda = 5)    # Sample of 1000 from Poisson distribution
p4 <- rpois(n = 10000, lambda = 5)   # Sample of 10000 from Poisson distribution

# Create a 2x2 plotting grid to display all the histograms on the same plot
par(mfrow=c(2,2))
```

```
# Plot for sample size = 10
hist(p1, main = "Sample size 10",
     xlab = "Number of Events", ylab = "Frequency",
     col = "lightcoral", border = "black",
     probability = TRUE, breaks = 5)
lines(density(p1), col = "blue", lwd = 2)  # Density line in blue
# Comment: "For n = 10, the histogram is very sparse, with many zero counts. The density line is rough 

# Plot for sample size = 100
hist(p2, main = "Sample size 100",
     xlab = "Number of Events", ylab = "Frequency",
     col = "lightcoral", border = "black",
     probability = TRUE, breaks = 10)
lines(density(p2), col = "blue", lwd = 2)  # Density line in blue
# Comment: "At n = 100, the histogram starts to resemble the Poisson distribution more closely. The den

# Plot for sample size = 1000
hist(p3, main = "Sample size 1000",
     xlab = "Number of Events", ylab = "Frequency",
     col = "lightcoral", border = "black",
     probability = TRUE, breaks = 15)
lines(density(p3), col = "blue", lwd = 2)  # Density line in blue
# Comment: "With n = 1000, the histogram becomes much smoother and is more representative of the Poisso

# Plot for sample size = 10000
hist(p4, main = "Sample size 10000",
     xlab = "Number of Events", ylab = "Frequency",
     col = "lightcoral", border = "black",
     probability = TRUE, breaks = 20)
lines(density(p4), col = "blue", lwd = 2)  # Density line in blue
```



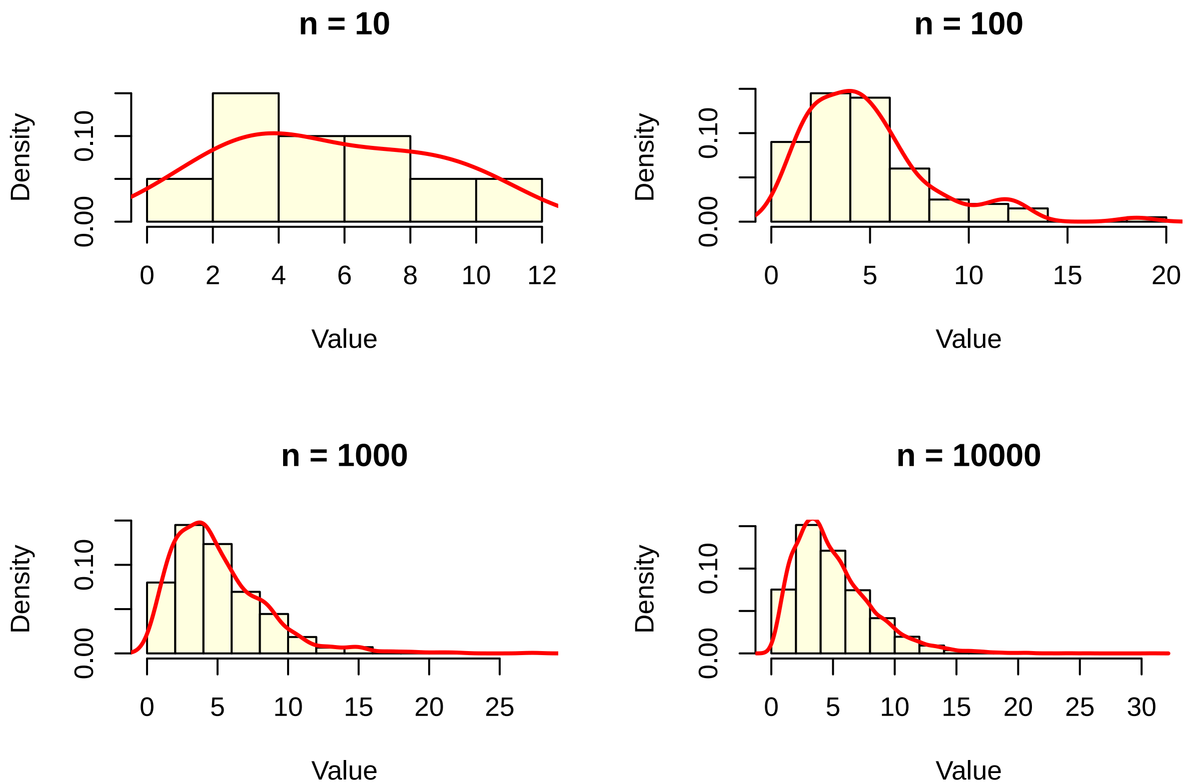

```
# Comment: "At n = 10000, the histogram is very smooth and closely approximates the theoretical Poisson


#Normal distribution
# Simulate data from a normal distribution with mean = 0.1 and sd = 0.84
norm <- rnorm(10, mean = 0.1, sd = 0.84)       # Sample size = 10
norm1 <- rnorm(100, mean = 0.1, sd = 0.84)      # Sample size = 100
norm2 <- rnorm(1000, mean = 0.1, sd = 0.84)     # Sample size = 1000
norm3 <- rnorm(10000, mean = 0.1, sd = 0.84)    # Sample size = 10000

# Create a 2x2 plotting grid to display all the histograms on the same plot
par(mfrow=c(2,2))

# Plot for sample size = 10
hist(norm, probability = TRUE, main = "n = 10",
     xlab = "Value", ylab = "Density",
     col = "lightgreen", border = "black",
     breaks = 5)
lines(density(norm), col = "blue", lwd = 2)   # Density line in blue
# Comment: "For n = 10, the histogram is very jagged and sparse, with a limited representation of the n

# Plot for sample size = 100
hist(norm1, probability = TRUE, main = "n = 100",
     xlab = "Value", ylab = "Density",
     col = "lightgreen", border = "black",
     breaks = 10)
```

```
lines(density(norm1), col = "blue", lwd = 2)  # Density line in blue
# Comment: "At n = 100, the histogram becomes smoother and starts to show the bell-shaped curve typical

# Plot for sample size = 1000
hist(norm2, probability = TRUE, main = "n = 1000",
     xlab = "Value", ylab = "Density",
     col = "lightgreen", border = "black",
     breaks = 15)
lines(density(norm2), col = "blue", lwd = 2)  # Density line in blue
# Comment: "With n = 1000, the histogram closely resembles the true normal distribution. The density li

# Plot for sample size = 10000
hist(norm3, probability = TRUE, main = "n = 10000",
     xlab = "Value", ylab = "Density",
     col = "lightgreen", border = "black",
     breaks = 20)
lines(density(norm3), col = "blue", lwd = 2)  # Density line in blue
```



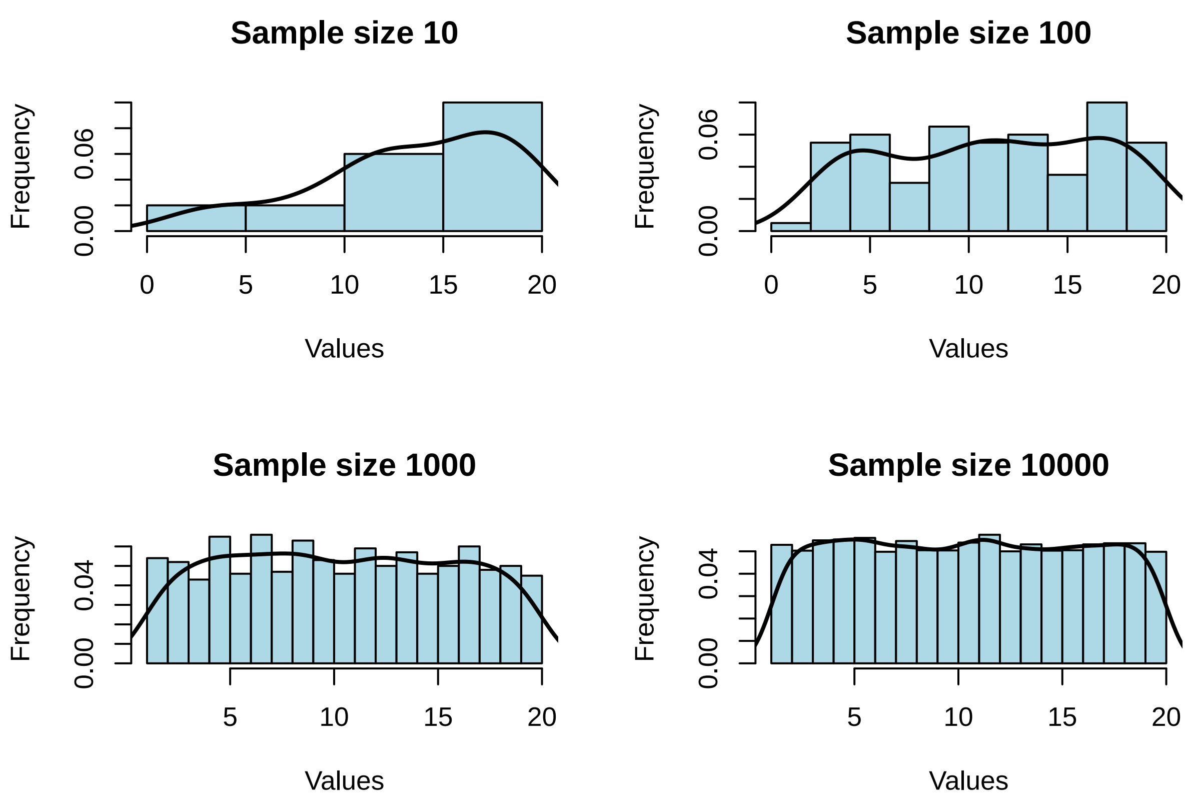

```
# Comment: "At n = 10000, the histogram is very smooth, and the density line fits the data perfectly. T



#Chi-square distribution
# Simulate data from a chi-square distribution with df = 5
ha <- rchisq(10, df = 5)       # Sample size = 10
hb <- rchisq(100, df = 5)      # Sample size = 100
```

```r
hc <- rchisq(1000, df = 5)    # Sample size = 1000
hd <- rchisq(10000, df = 5)   # Sample size = 10000

# Create a 2x2 plotting grid to display all the histograms on the same plot
par(mfrow=c(2,2))

# Plot for sample size = 10
hist(ha, probability = TRUE, main = "n = 10",
     xlab = "Value", ylab = "Density",
     col = "lightyellow", border = "black",
     breaks = 5)
lines(density(ha), col = "red", lwd = 2)  # Density line in red
# Comment: "For n = 10, the histogram is very irregular and does not resemble the typical Chi-square di

# Plot for sample size = 100
hist(hb, probability = TRUE, main = "n = 100",
     xlab = "Value", ylab = "Density",
     col = "lightyellow", border = "black",
     breaks = 10)
lines(density(hb), col = "red", lwd = 2)  # Density line in red
# Comment: "At n = 100, the histogram begins to take a more recognizable shape of the Chi-square distri

# Plot for sample size = 1000
hist(hc, probability = TRUE, main = "n = 1000",
     xlab = "Value", ylab = "Density",
     col = "lightyellow", border = "black",
     breaks = 15)
lines(density(hc), col = "red", lwd = 2)  # Density line in red
# Comment: "With n = 1000, the histogram closely matches the expected Chi-square distribution. The dens

# Plot for sample size = 10000
hist(hd, probability = TRUE, main = "n = 10000",
     xlab = "Value", ylab = "Density",
     col = "lightyellow", border = "black",
     breaks = 20)
lines(density(hd), col = "red", lwd = 2)  # Density line in red
```



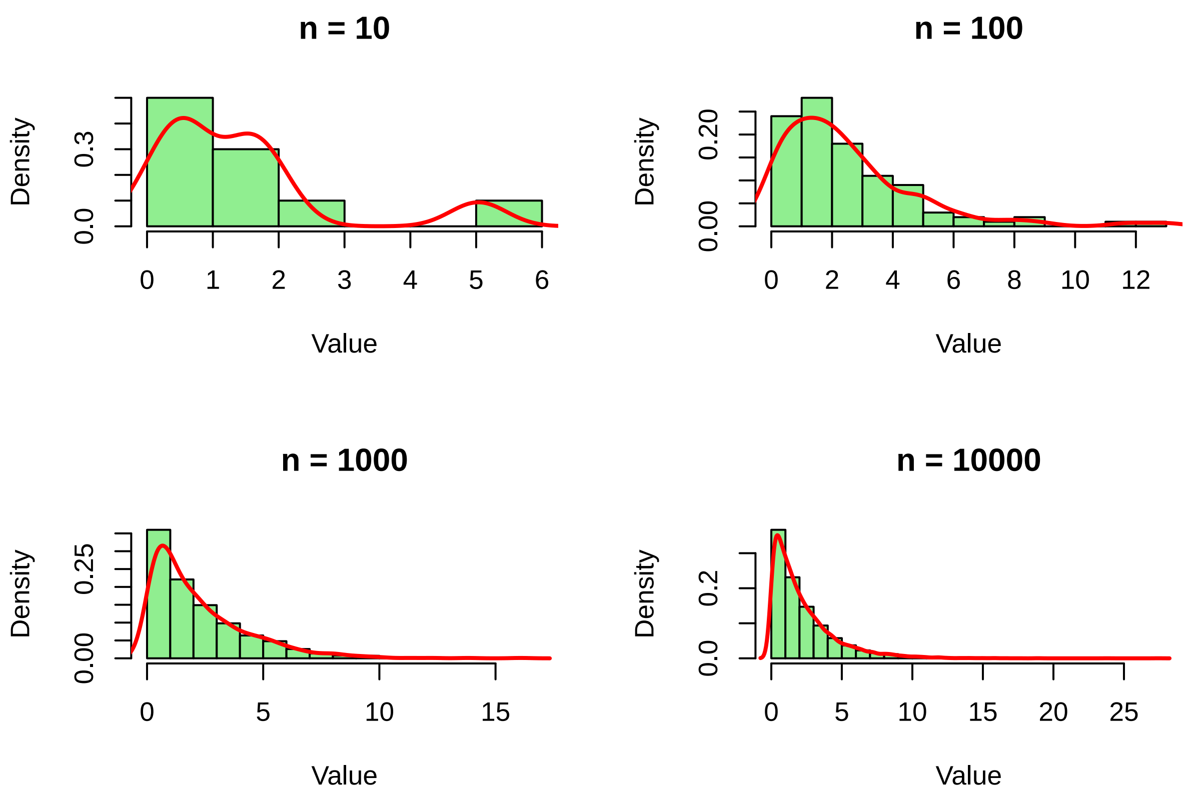

```r
# Comment: "For n = 10000, the histogram is very smooth, and the density line fits the data perfectly.


#uniform distribution
# Sample sizes for the simulations
hh <- runif(10, min = 1, max = 20)      # Sample of 10 from Uniform distribution
hh1 <- runif(100, min = 1, max = 20)    # Sample of 100 from Uniform distribution
hh2 <- runif(1000, min = 1, max = 20)   # Sample of 1000 from Uniform distribution
hh3 <- runif(10000, min = 1, max = 20) # Sample of 10000 from Uniform distribution

# Create a 2x2 plotting grid to display all the histograms on the same plot
par(mfrow=c(2,2))

# Plot for sample size = 10
hist(hh, main = "Sample size 10",
     xlab = "Values", ylab = "Frequency",
     col = "lightblue", border = "black",
     probability = TRUE, breaks = 5)
lines(density(hh), col = "black", lwd = 2)  # Density line in blue
# Comment: "For n = 10, the histogram is discrete with some irregularity. The density line is rough, re

# Plot for sample size = 100
hist(hh1, main = "Sample size 100",
     xlab = "Values", ylab = "Frequency",
     col = "lightblue", border = "black",
     probability = TRUE, breaks = 10)
```

```r
lines(density(hh1), col = "black", lwd = 2)  # Density line in blue
# Comment: "At n = 100, the histogram begins to smooth out. The density line is still somewhat jagged d

# Plot for sample size = 1000
hist(hh2, main = "Sample size 1000",
     xlab = "Values", ylab = "Frequency",
     col = "lightblue", border = "black",
     probability = TRUE, breaks = 20)
lines(density(hh2), col = "black", lwd = 2)  # Density line in blue
# Comment: "With n = 1000, the histogram is much smoother and the density line is a much better approxi

# Plot for sample size = 10000
hist(hh3, main = "Sample size 10000",
     xlab = "Values", ylab = "Frequency",
     col = "lightblue", border = "black",
     probability = TRUE, breaks = 25)
lines(density(hh3), col = "black", lwd = 2)  # Density line in blue
```



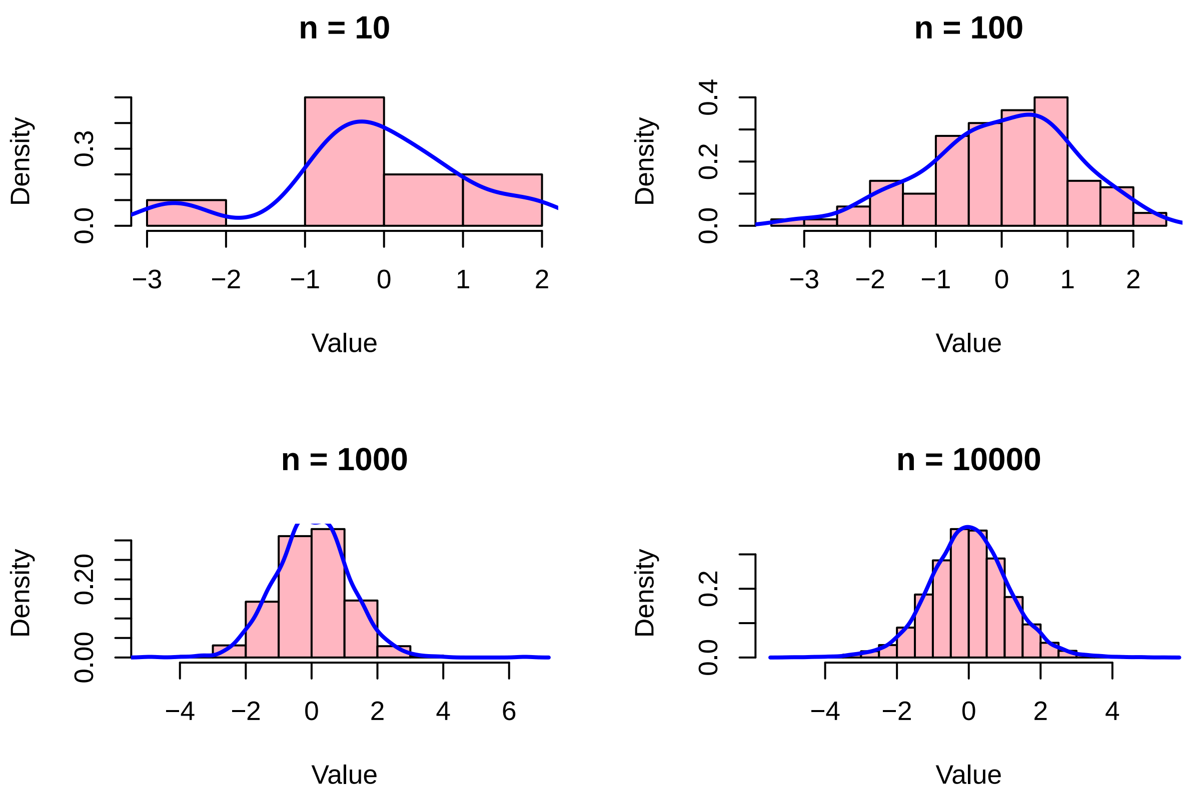

```r
# Comment: "At n = 10000, the histogram is very close to a uniform distribution. The density line is sm




#Exponential distribution
# Simulate data from an exponential distribution with rate = 0.45
exp1 <- rexp(10, rate = 0.45)      # Sample size = 10
```

```r
exp2 <- rexp(100, rate = 0.45)     # Sample size = 100
exp3 <- rexp(1000, rate = 0.45)    # Sample size = 1000
exp4 <- rexp(10000, rate = 0.45)   # Sample size = 10000

# Create a 2x2 plotting grid to display all the histograms on the same plot
par(mfrow=c(2,2))

# Plot for sample size = 10
hist(exp1, probability = TRUE, main = "n = 10",
     xlab = "Value", ylab = "Density",
     col = "lightgreen", border = "black",
     breaks = 5)
lines(density(exp1), col = "red", lwd = 2)  # Density line in red
# Comment: "For n = 10, the histogram is highly irregular and does not closely follow the expected expo

# Plot for sample size = 100
hist(exp2, probability = TRUE, main = "n = 100",
     xlab = "Value", ylab = "Density",
     col = "lightgreen", border = "black",
     breaks = 10)
lines(density(exp2), col = "red", lwd = 2)  # Density line in red
# Comment: "At n = 100, the histogram starts to approximate the expected exponential distribution. The

# Plot for sample size = 1000
hist(exp3, probability = TRUE, main = "n = 1000",
     xlab = "Value", ylab = "Density",
     col = "lightgreen", border = "black",
     breaks = 15)
lines(density(exp3), col = "red", lwd = 2)  # Density line in red
# Comment: "With n = 1000, the histogram more closely matches the shape of the exponential distribution

# Plot for sample size = 10000
hist(exp4, probability = TRUE, main = "n = 10000",
     xlab = "Value", ylab = "Density",
     col = "lightgreen", border = "black",
     breaks = 20)
lines(density(exp4), col = "red", lwd = 2)  # Density line in red
```




```
# Comment: "For n = 10000, the histogram is smooth, and the density line fits the data perfectly. This


#t-distribution
# Simulate data from a t-distribution with 10 degrees of freedom
b1 <- rt(10, df = 10)       # Sample size = 10
b2 <- rt(100, df = 10)      # Sample size = 100
b3 <- rt(1000, df = 10)     # Sample size = 1000
b4 <- rt(10000, df = 10)    # Sample size = 10000

# Create a 2x2 plotting grid to display all the histograms on the same plot
par(mfrow=c(2,2))

# Plot for sample size = 10
hist(b1, probability = TRUE, main = "n = 10",
     xlab = "Value", ylab = "Density",
     col = "lightpink", border = "black",
     breaks = 5)
lines(density(b1), col = "blue", lwd = 2)  # Density line in blue
# Comment: "For n = 10, the histogram is very jagged and has heavy tails. The density line is not a goo

# Plot for sample size = 100
hist(b2, probability = TRUE, main = "n = 100",
     xlab = "Value", ylab = "Density",
     col = "lightpink", border = "black",
     breaks = 10)
```

```r
lines(density(b2), col = "blue", lwd = 2)  # Density line in blue
# Comment: "At n = 100, the histogram is smoother than for n = 10, but the tails of the distribution ar

# Plot for sample size = 1000
hist(b3, probability = TRUE, main = "n = 1000",
     xlab = "Value", ylab = "Density",
     col = "lightpink", border = "black",
     breaks = 15)
lines(density(b3), col = "blue", lwd = 2)  # Density line in blue
# Comment: "For n = 1000, the histogram is much smoother and starts to resemble the true t-distribution

# Plot for sample size = 10000
hist(b4, probability = TRUE, main = "n = 10000",
     xlab = "Value", ylab = "Density",
     col = "lightpink", border = "black",
     breaks = 20)
lines(density(b4), col = "blue", lwd = 2)  # Density line in blue
```




```r
# Comment: "At n = 10000, the histogram closely matches the true t-distribution. The density line fits
```