UNIVERSIDAD DE EL SALVADOR

FACULTAD MULDISCIPLINARIA DE OCCIDENTE

DEPARTAMENTO DE MATEMÁTICAS

LICENCIATURA EN ESTADÍSTICA

PRACTICAS REALIZADAS EN EL SOFTWARE R

DOCENTE:
LICENCIADO. JAIME ISAAC PEÑA

PRESENTADO POR:
MORIS SALVADOR HENRIQUEZ LIMA

Viernes 09 de Septiembre del 2022

Índice

1. TEOREMA DEL LÍMITE CENTRAL

Como hemos visto, R tiene algunas funciones para generar números aleatorios. Para estos números aleatorios, podemos ver la distribución usando histogramas y otras herramientas. Lo que queremos hacer ahora, es generar nuevos tipos de números aleatorios e investigar qué tipo de distribución tienen.

El Teorema del Límite Central (TLC) informa acerca de la distribución de muestreo de medias de muestras con tamaño n . Recuérdese que básicamente existen tres tipos de información que se desea conocer sobre una distribución:

1) dónde está el centro,

2) qué tanto varía, y

3) cómo está repartida.

El Teorema del Límite Central establece que sí las observaciones $X_1, X_2, X_3, .., X_n$ son variables aleatorias independientes e idénticamente distribuidas con una distribución de probabilidad cualquiera y en la cual cada una de ellas tenga la misma media μ y la misma varianza σ^2 (ambas finitas).

Entonces el promedio muestral tiene una distribución con media μ y varianza σ^2/n que tiende hacia una distribución $N(0,1)$ a medida que n tiende a ∞.

¿Cómo podemos comprobar esto? La simulación es un excelente camino.

1.1. Procesos a Realizar y Ejemplos.

1.1.1. Activa tu directorio de Trabajo

Proceso Realizado:

```
> getwd()
[1] "C:/Moris_Henriquez/Practicas_R_Sweave_2022"
```

Al realizar el comando anterior verificamos la direccion en la cual estamos trabajando, debido a que nos encontramos en nuestro escritorio de trabajo omitimos lo siguiente: setwd(C:/Curso R2022).

1.1.2. Crea un nuevo script y llámele: Sript16-Simulación del TLC.

Para este proceso creamos un nuevo scrip en R-MarkDonw con el nombre de Practica16.

1.1.3. Simular el Teorema del Límite Central con datos binomial.

Consideremos n repeticiones independientes y sea X el número de veces que ocurre un suceso A. Sea p igual a $P(A)$ y supongamos que este número es constante para todas las repeticiones consideradas.
El teorema central del límite nos indica que:

$$\frac{X - E[X]}{\sqrt{V(X)}} = \frac{X - np}{\sqrt{npq}}$$

, es aproximadamente $N(0,1)$

1.1.4. EJEMPLOS.

- **Ejemplo 1:**

Generar 100 números aleatorios de una distribución binomial con parámetros n=10 (número de ensayos o pruebas), y p=0.25 (probabilidad de éxito).

Llevamos a cabo la solucion de la siguiente manera:

Realizamos el Tamaño de la muestra, la definimos por: tm.

```
> tm = 100
> tm
```

```
[1] 100
```

```
> n <- 10
> n
```

```
[1] 10
```

```
> p <- 0.25
> p
```

```
[1] 0.25
```

Ahora generamos los 100 numeros aleatorios:

```
> S = rbinom(tm, n, p)
> S
```

```
  [1] 5 3 3 3 1 4 2 0 4 3 0 3 4 0 2 2 4 6 2 2 0 1 3 0 3 4 2 0 4 2 1 2 1 3 2 3 4
 [38] 2 4 3 2 4 5 8 2 3 2 2 4 3 1 1 2 4 2 2 2 2 1 1 4 2 1 3 1 4 2 3 5 0 1 2 3 1
 [75] 4 3 1 2 0 2 2 2 0 3 0 3 2 4 3 3 2 4 1 3 2 1 2 5 4 3
```

Ahora realizamos el proceso de estandarizar cada una de las observaciones:

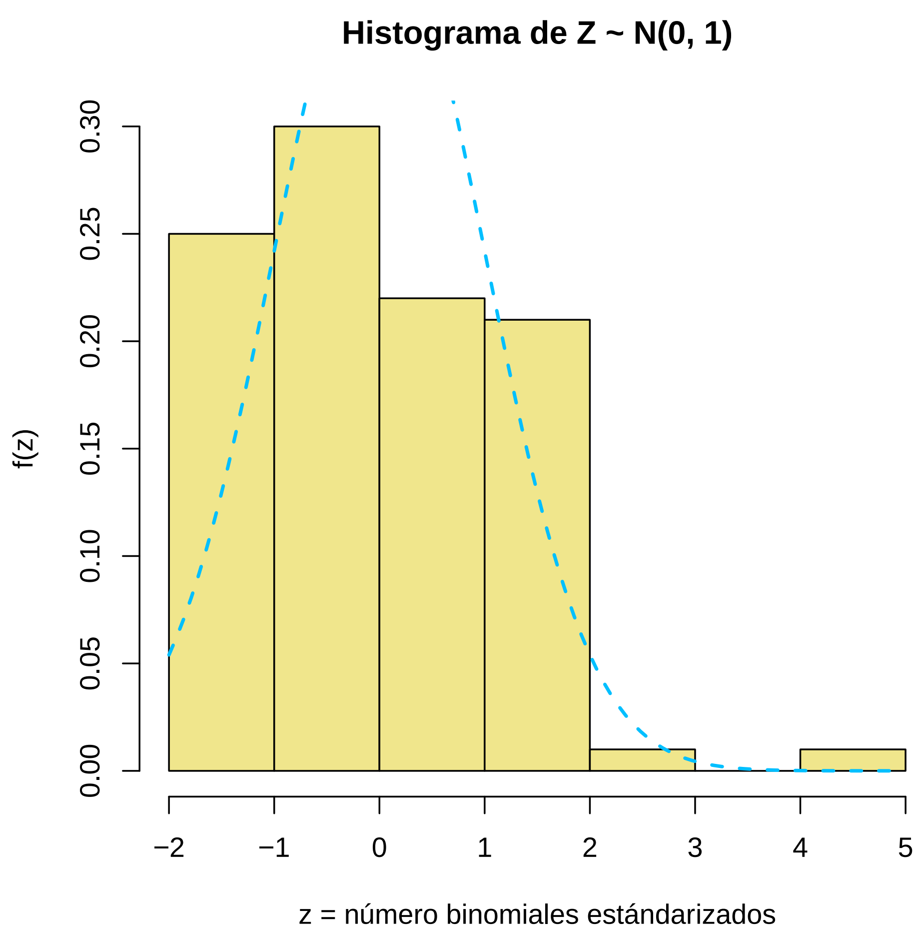
```
> Z = (S-n*p)/sqrt(n*p*(1-p))
> Z
```

```
  [1]  1.8257419  0.3651484  0.3651484  0.3651484 -1.0954451  1.0954451
  [7] -0.3651484 -1.8257419  1.0954451  0.3651484 -1.8257419  0.3651484
 [13]  1.0954451 -1.8257419 -0.3651484 -0.3651484  1.0954451  2.5560386
 [19] -0.3651484 -0.3651484 -1.8257419 -1.0954451  0.3651484 -1.8257419
 [25]  0.3651484  1.0954451 -0.3651484 -1.8257419  1.0954451 -0.3651484
 [31] -1.0954451 -0.3651484 -1.0954451  0.3651484 -0.3651484  0.3651484
 [37]  1.0954451 -0.3651484  1.0954451  0.3651484 -0.3651484  1.0954451
 [43]  1.8257419  4.0166321 -0.3651484  0.3651484 -0.3651484 -0.3651484
 [49]  1.0954451  0.3651484 -1.0954451 -1.0954451 -0.3651484  1.0954451
 [55] -0.3651484 -0.3651484 -0.3651484 -0.3651484 -1.0954451 -1.0954451
 [61]  1.0954451 -0.3651484 -1.0954451  0.3651484 -1.0954451  1.0954451
 [67] -0.3651484  0.3651484  1.8257419 -1.8257419 -1.0954451 -0.3651484
 [73]  0.3651484 -1.0954451  1.0954451  0.3651484 -1.0954451 -0.3651484
 [79] -1.8257419 -0.3651484 -0.3651484 -0.3651484 -1.8257419  0.3651484
 [85] -1.8257419  0.3651484 -0.3651484  1.0954451  0.3651484  0.3651484
 [91] -0.3651484  1.0954451 -1.0954451  0.3651484 -0.3651484 -1.0954451
 [97] -0.3651484  1.8257419  1.0954451  0.3651484
```

La variable X tiene los resultados, y podemos ver la distribución de los números aleatorios en X con un histograma.

Histograma:

```
> hist(Z, main="Histograma de Z ~ N(0, 1)", xlab="z = número binomiales estándarizados",
+ ylab="f(z)", prob=TRUE, col="khaki")
> curve(dnorm(x, 0, 1), col = "deepskyblue", lty=2, lwd=2, add=TRUE)
```



La distribución muestra un gráfico aproximadamente normal. Esto es, en forma de campana, centrada en 0 y con desviación estándar 1.

1.1.5. Simular el TLC con datos de una distribución normal.

El teorema central del límite establece que:

$$\frac{\bar{x} - \mu}{\sigma/\sqrt{n}} \sim N(0, 1)$$

- **Ejemplo 2:**

Suponga que X_i es normal con media $\mu = 5$ y desviación estándar $\sigma = 5$.

Entonces necesitamos una función para encontrar el valor de: $\frac{\bar{x}-\mu}{\sigma/\sqrt{n}}$

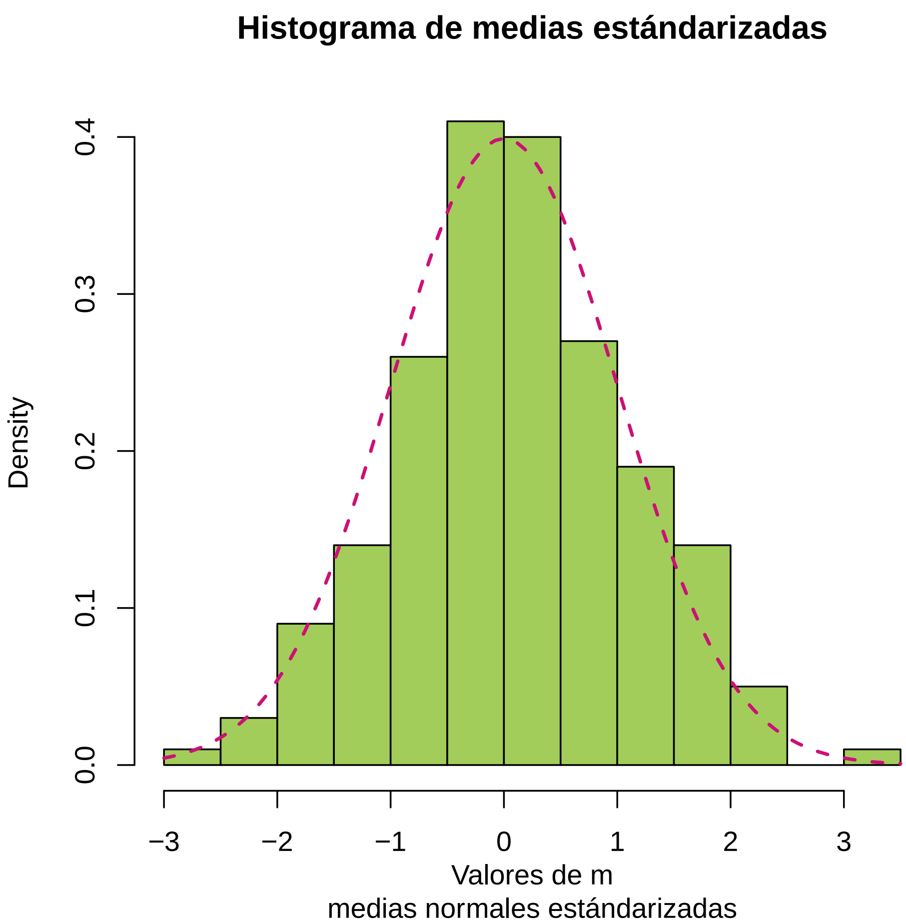
Realizamos el Proceso de creación de la Función solicitada de la siguiente manera:

```
> simulNorm <- function(mu, sigma, m=5, n=100)
+  {
+  vectMedias <<- numeric(0)
+  MediasEstand <<- numeric(0)
+  for (i in 1:m)
+ {
+  X = rnorm(n, mu, sigma)
+  # genera n valores normales
+  vectMedias[i] <<- mean(X)
+  MediasEstand[i] <<- (vectMedias[i] - mu)/(sigma/sqrt(n))
+  }
+ }
> mu = 5
> mu

[1] 5

> sigma = 5
> sigma

[1] 5

> m <- 200
> m

[1] 200
```

Utilizamos la Función Creada Para la Distribución Normal.

Numero de muestras o medias a obtener mediante la Función creada anteriormente:

```
> simulNorm(mu, sigma, m)
> hist(MediasEstand, main="Histograma de medias estándarizadas", xlab="Valores de m
+ medias normales estándarizadas", prob=TRUE, col="darkolivegreen3")
> curve(dnorm(x, 0, 1), col = "deeppink3", lty=2, lwd=2, add=TRUE)
```



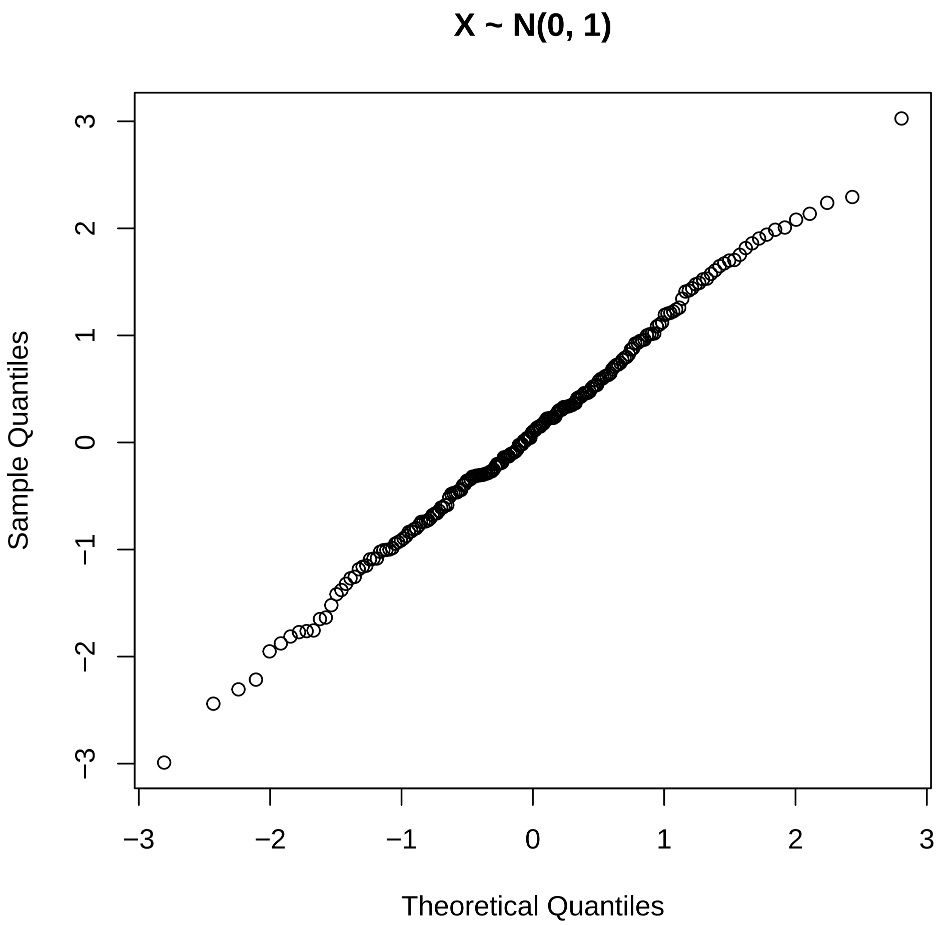
1.1.6. Grafico de Probabilidad Normal

Un mejor gráfico que el histograma para decidir si los datos aleatorios son aproximadamente normal es el llamado gráfico de probabilidad normal. La idea básica es graficar los cuantiles de sus datos contra los correspondientes cuantiles de la distribució normal. Los cuantiles de un conjunto de datos preferidos son la Mediana, Q_1 y Q_3 los más generales. El cuantil q es el valor en los datos donde. También el cuantil 0.25 es Q_1, el cuantil 0.5 es la mediana y el cuantil 0.75 es Q_3.Los cuantiles para la distribución teórica son similares, sólo cambia el número de puntos datos menores, o sea el área a la izquierda del monto especificado. Por ejemplo, la mediana parte el área por debajo de la curva de densidad en la mitad.

El gráfico de probabilidad normal es fácil de leer si conoce cómo. Esencialmente, si el gráfico parece una línea recta entonces los datos son aproximadamente normal. Está línea no es una línea de regresión. La línea es trazada a través de los puntos formados por el primer y tercer cuartil.

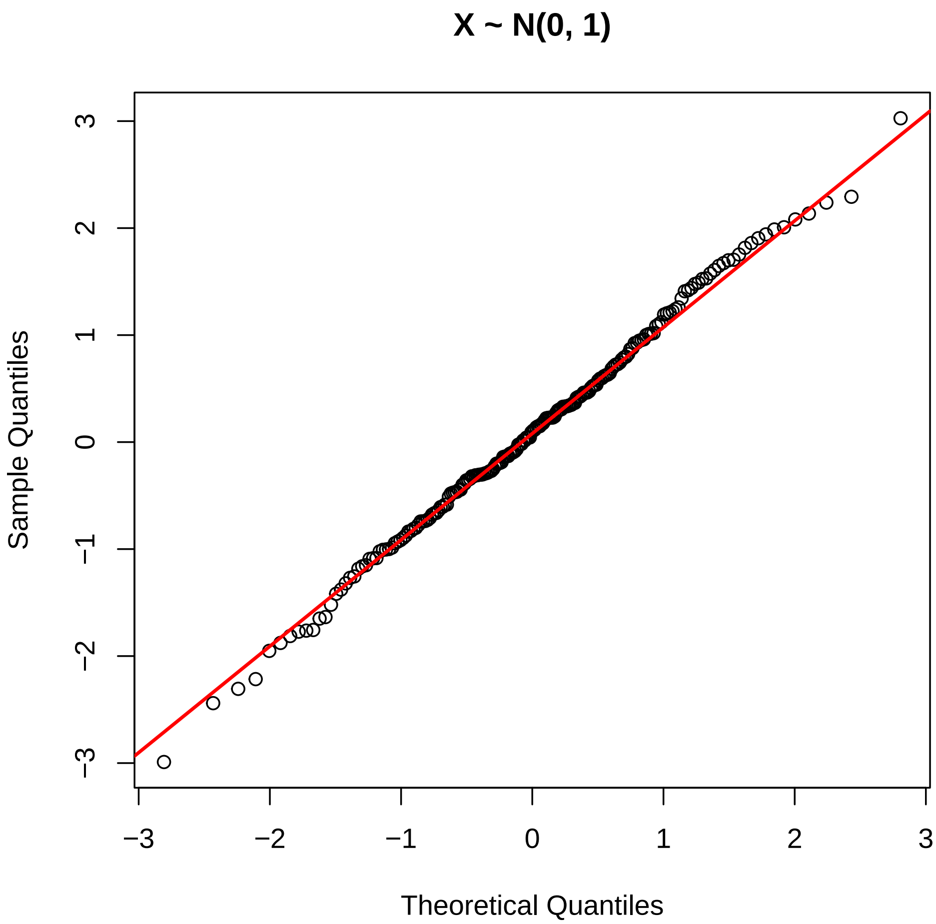
Primer Proceso a realizar:

```
> qqnorm(MediasEstand, main="X ~ N(0, 1)")
```



Proceso de mostrar linea de normalidad:

```
> qqnorm(MediasEstand, main="X ~ N(0, 1)")
> qqline(MediasEstand, lty=1, lwd=2, col="red")
```



1.1.7. Simular el Teorema del Límite Central con datos exponencial.

Un ejemplo de una distribución sesgada es la exponencial. Necesitamos conocer que sí tiene media $\mu = 10$, entonces la desviación estándar σ es también 10, por eso sólo necesitamos especificar la media.

Vamos a simular para varios valores de n.

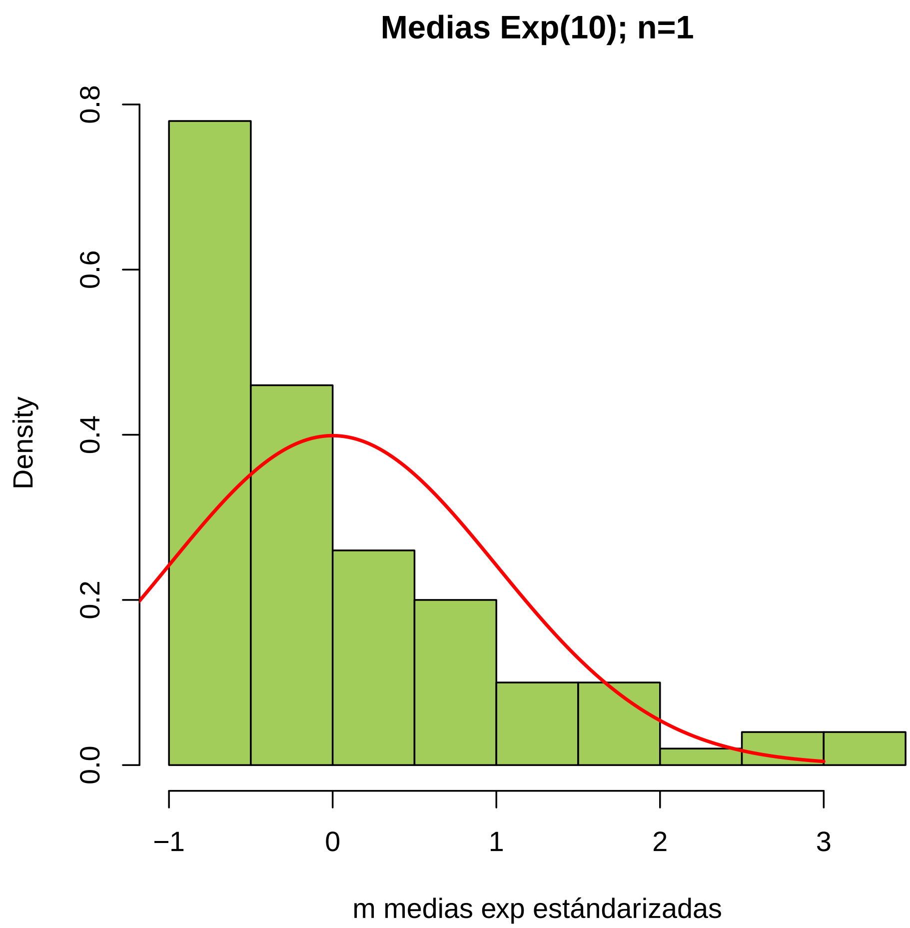
Para cada una de las m =100 muestras, n será, 5, 15, 50 (el número de valores aleatorios en cada uno de los promedios).

Realizamos el Proceso de creación de la Función solicitada:

```
> simulExp <- function(mu, m=5, n=100)
+ {
+  razon <- 1/mu
+  vectMedias <<- numeric(0)
+  MediasEstand <<- numeric(0)
+  for (i in 1:m)
+ {
+  X = rexp(n, razon)
+ # genera n valores exponenciales
+  vectMedias[i] <<- mean(X)
+  MediasEstand[i] <<- (vectMedias[i] - mu)/(mu/sqrt(n))
+  }
+ }
>
```

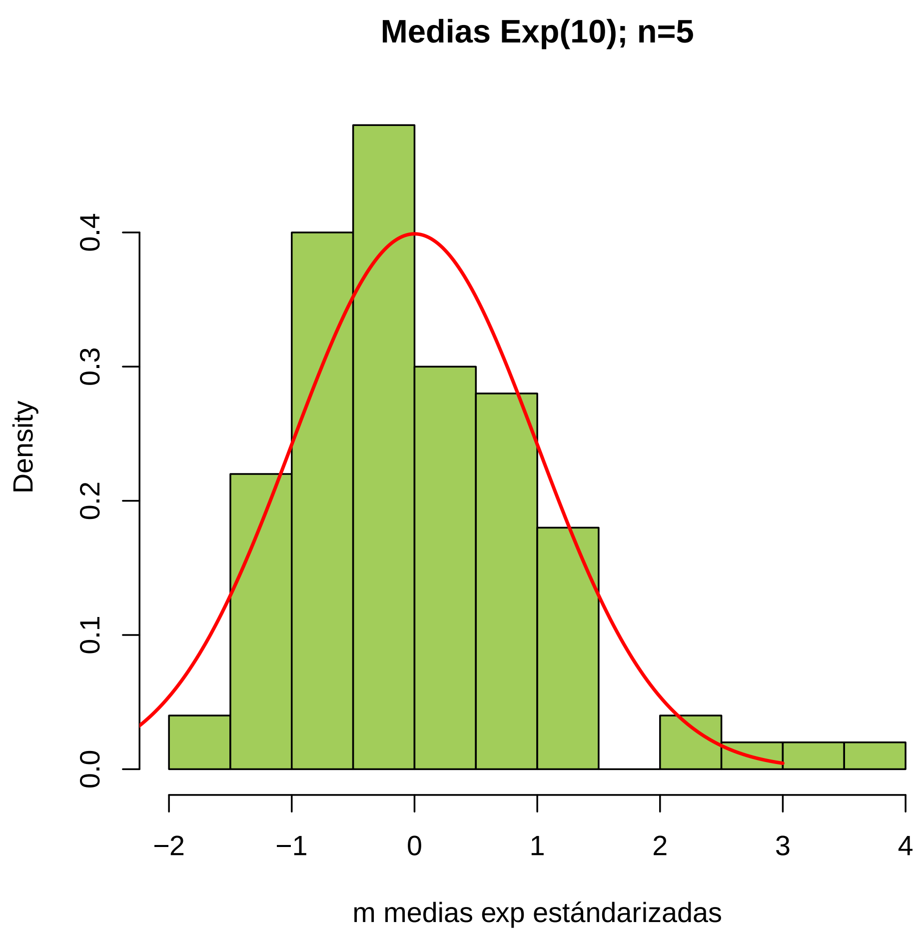
Llevamos a cabo la realización de los siguientes ejemplos:
Para n=1

```
>  mu=10
>  m <- 100
>  n <- 1
>  simulExp(mu, m, n)
>  hist(MediasEstand, main="Medias Exp(10); n=1", xlab="m medias exp estándarizadas",
+ prob=TRUE, col="darkolivegreen3")
>  xvals = seq(from=-3, to=3, by=0.01)
>  points(xvals, dnorm(xvals, 0, 1), col = "red", type="l", lty=1, lwd=2)
```



Para n=5

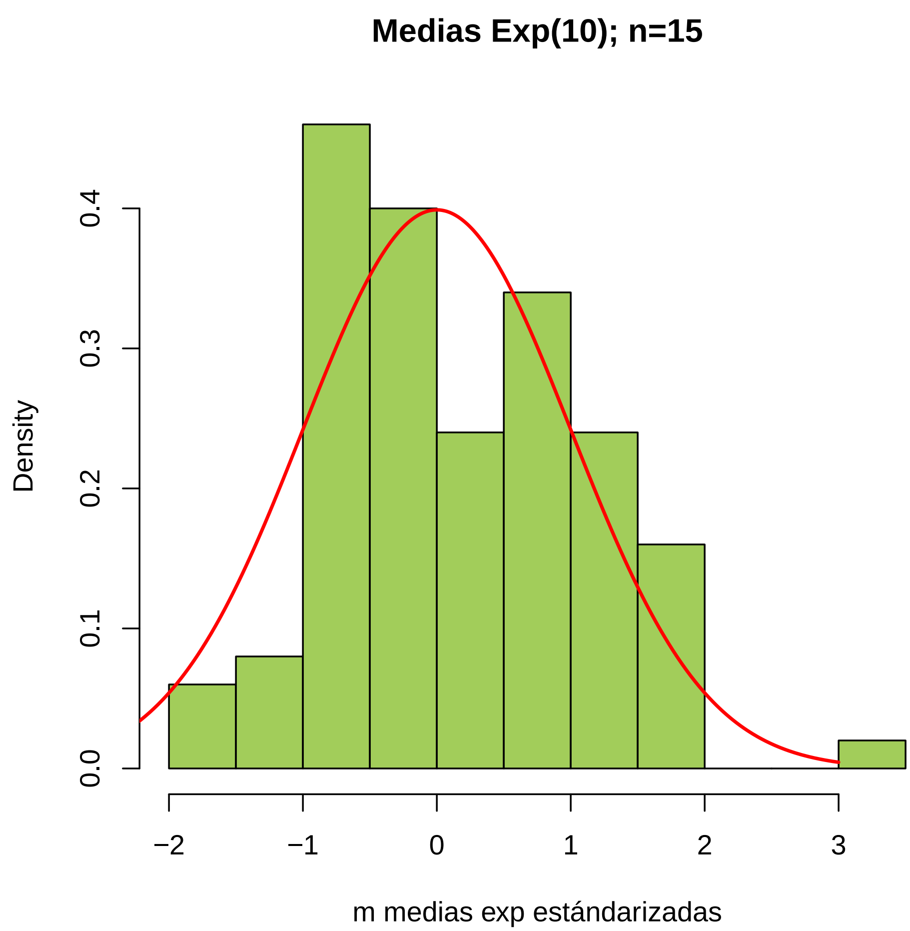
```
> mu=10
> m <- 100
> n <- 5
>  simulExp(mu, m, n)
>  hist(MediasEstand, main="Medias Exp(10); n=5", xlab="m medias exp estándarizadas",
+ prob=TRUE, col="darkolivegreen3")
>  xvals = seq(from=-3, to=3, by=0.01)
>  points(xvals, dnorm(xvals, 0, 1), col = "red", type="l", lty=1, lwd=2)
>
```



Repita este proceso para n=15 y n=50

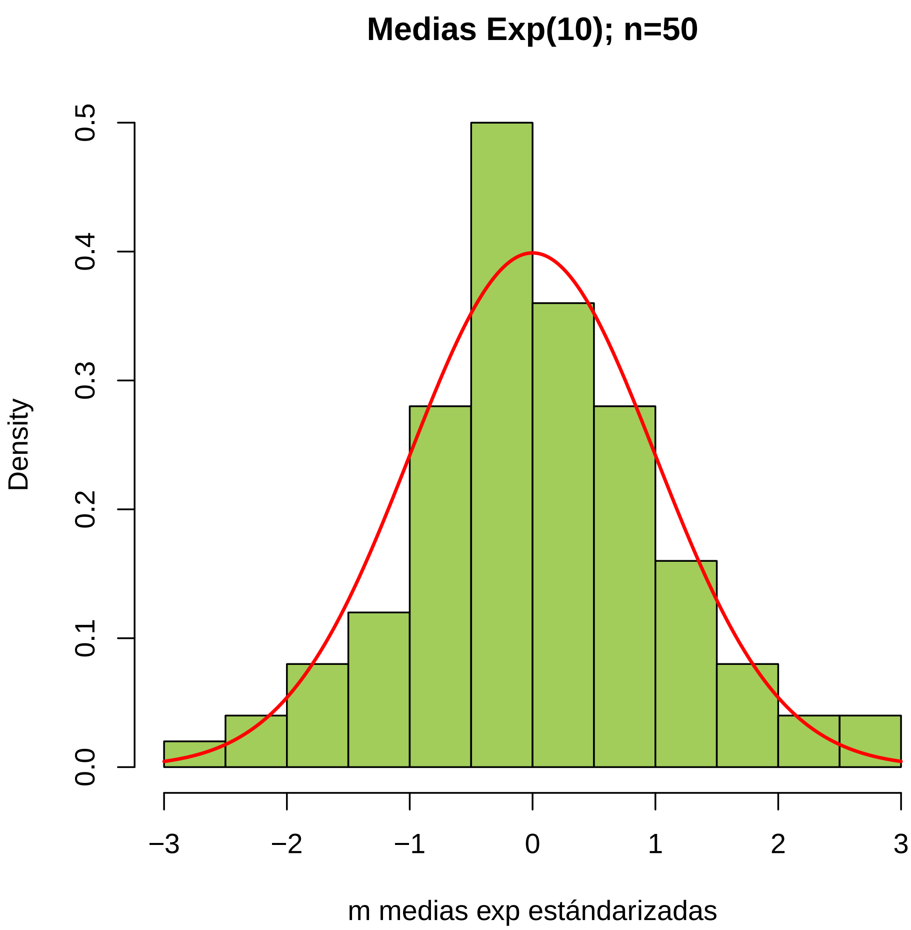
Para n=15

```
> mu=10
> m <- 100
> n <- 15
> simulExp(mu, m, n)
> hist(MediasEstand, main="Medias Exp(10); n=15", xlab="m medias exp estándarizadas",
+ prob=TRUE, col="darkolivegreen3")
> xvals = seq(from=-3, to=3, by=0.01)
> points(xvals, dnorm(xvals, 0, 1), col = "red", type="l", lty=1, lwd=2)
```



Para n=50

```
>  mu=10
>  m <-100
>  n <- 50
>  simulExp(mu, m, n)
>  hist(MediasEstand, main="Medias Exp(10); n=50", xlab="m medias exp estándarizadas",
+ prob=TRUE, col="darkolivegreen3")
>  xvals = seq(from=-3, to=3, by=0.01)
>  points(xvals, dnorm(xvals, 0, 1), col = "red", type="l", lty=1, lwd=2)
```



Observe que el histograma tiene una forma muy acampanada entre n=15 y n=50, aunque justo en n=50 parece todavía ser un poco sesgada.

2. Ejercicio Propuesto sobre la Practica.

1. Simular el Teorema del Límite Central para una variable aleatoria que tiene distribución Poisson con lambda o media 4. Considerar 100 muestras aleatorias de tamaño 1, 10, 30, 50 valores de la distribución. Los gráficos deben estar en una misma ventana.

Solución:

Realizamos el Proceso de creación de la Función solicitada de la siguiente manera:

```
> simulPois <- function(mu, m=5, n=100)
+  {
+  vectMedias <<- numeric(0)
+  MediasEstand <<- numeric(0)
+  for (i in 1:m)
+ {
+  X = rpois(n,mu)
+  # genera n valores de la distribucion de Poisson:
+  vectMedias[i] <<- mean(X)
+  MediasEstand[i] <<- (vectMedias[i] - mu)/(mu/sqrt(n))
+  }
+ }
> mu = 4
> mu

[1] 4

> m <- 100
> m

[1] 100
```

Proceso de Simulacón mediante la Distribucíon de Poisson para n = 1, 10, 30, 50

```
>  par(mfrow = c(2,2))
>  # Para n = 1
>  mu= 4
>  m <-100
>  n <- 1
>  simulPois(mu, m, n)
>  hist(MediasEstand, main="Histograma Estandarizado ; n=1", xlab="m medias estándarizadas"
+  , prob=TRUE, col="darkolivegreen3")
>  xvals = seq(from=-3, to=3, by=0.01)
>  points(xvals, dnorm(xvals, 0, 1), col = "red", type="l", lty=1, lwd=2)
>  # Para n = 10
>  mu= 4
>  m <-100
>  n <- 10
>  simulPois(mu, m, n)
>  hist(MediasEstand, main="Histograma Estandarizad0 ; n=10", xlab="m medias
+  estándarizadas" , prob=TRUE, col="darkolivegreen3")
>  xvals = seq(from=-3, to=3, by=0.01)
```

```
> points(xvals, dnorm(xvals, 0, 1), col = "red", type="l", lty=1, lwd=2)
> # Para n = 30
> mu= 4
> m <-100
> n <- 30
> simulPois(mu, m, n)
> hist(MediasEstand, main="Histograma Estandarizado ; n=30", xlab="m medias
+ estándarizadas", prob=TRUE, col="darkolivegreen3")
> xvals = seq(from=-3, to=3, by=0.01)
> points(xvals, dnorm(xvals, 0, 1), col = "red", type="l", lty=1, lwd=2)
> # Para n = 50
> mu= 4
> m <-100
> n <- 50
> simulPois(mu, m, n)
> hist(MediasEstand, main="Histograma Estandarizado ; n=50", xlab="m medias
+ estándarizadas", prob=TRUE, col="darkolivegreen3")
> xvals = seq(from=-3, to=3, by=0.01)
> points(xvals, dnorm(xvals, 0, 1), col = "red", type="l", lty=1, lwd=2)
>
```