PEC 2024-2025 de Estadística

Alejandro Muñoz Fernández

27/04/2025

Como no he estudiado la unidad 4 todavía, mi única opción es el ejercicio 2, con enunciado:

> *En este ejercicio se propone simular 50000 observaciones de una variable aleatoria Z con distribución N (0, 1), a partir de los valores simulados de una variable uniforme en (0, 1), utilizando el método de BOX-MULLER (capítulo 3 del texto base).*
>
> 1. *Describir los pasos del procedimiento que le permitiría obtener los valores simulados de la N (0, 1) e implementar el procedimiento en un algoritmo para la simulación.*
> 2. *Explicar la manera en que un histograma de frecuencias nos ayuda a representar la función de densidad de una variable continua; para ello puede consultar cualquier manual introductorio de Estadística. Representar el histograma de los valores simulados. ¿Qué se puede decir de la forma del histograma obtenido? (máximo 1 página).*
> 3. *Utilizar los resultados de la simulación para aproximar la probabilidad P (Z > 1,645).*

Apartado 1

El algoritmo de Box-Müller genera dos variables normales independientes (X e Y), de media 0 y desviación típica 1, o sea, con distribución normal típica N(0,1). Lo usaremos para simular las 50000 observaciones.

Los pasos de este algoritmo para obtener las observaciones simuladas son:

1. Se generan dos variables aleatorias independientes (U_1 y U_2) con densidad uniforme entre 0 y 1, con 25000 observaciones cada una
2. Se aplica la transformación Box-Müller a U_1 y U_2 para obtener X e Y con estas fórmulas:
   - Devuelve X: $X = \sqrt{-2lnU_2}cos\pi U_1$
   - Devuelve Y: $Y = \sqrt{-2lnU_2}sen\pi U_1$
3. Se combinan X e Y en Z, con 50000 observaciones en total y con distribución normal N(0,1)

He implementado el algoritmo de Box-Müller con este código en R:

```r
# Número de observaciones; piden 50000
n <- 50000

# Genera n/2 pares de variables uniformes U₁ y U₂, cada una con 25000 variables
set.seed(123)  # Reproduce una secuencia de números aleatorios
U1 <- runif(n / 2)
U2 <- runif(n / 2)

# Aplica la transformación de Box-Müller (conforme a la página 199 del libro)
# Parte de la raíz cuadrada
raiz <- sqrt(-2 * log(U2))
# Parte del ángulo 0 (zeta, teta o theta o como se llame)
angulo <- 2 * pi * U1
X <- raiz * cos(angulo)
Y <- raiz * sin(angulo)

# Combina X e Y para obtener las 50000 observaciones
Z <- c(X, Y)
```

Apartado 2

El histograma representa la función de densidad de probabilidad en barras:

- la *anchura* se corresponde con el intervalo en que se dividen los datos
- la *altura* representa la frecuencia de observaciones de Z en ese intervalo

La suma de todas las áreas de todas las barras del histograma debe dar uno, lo que se asemeja al área bajo la curva de la función de densidad de una variable aleatoria continua, que también da uno.

El histograma no representa la función de densidad con exactitud, porque se divide en intervalos, pero se aproxima bastante si los intervalos son bastante pequeños (barras estrechas).

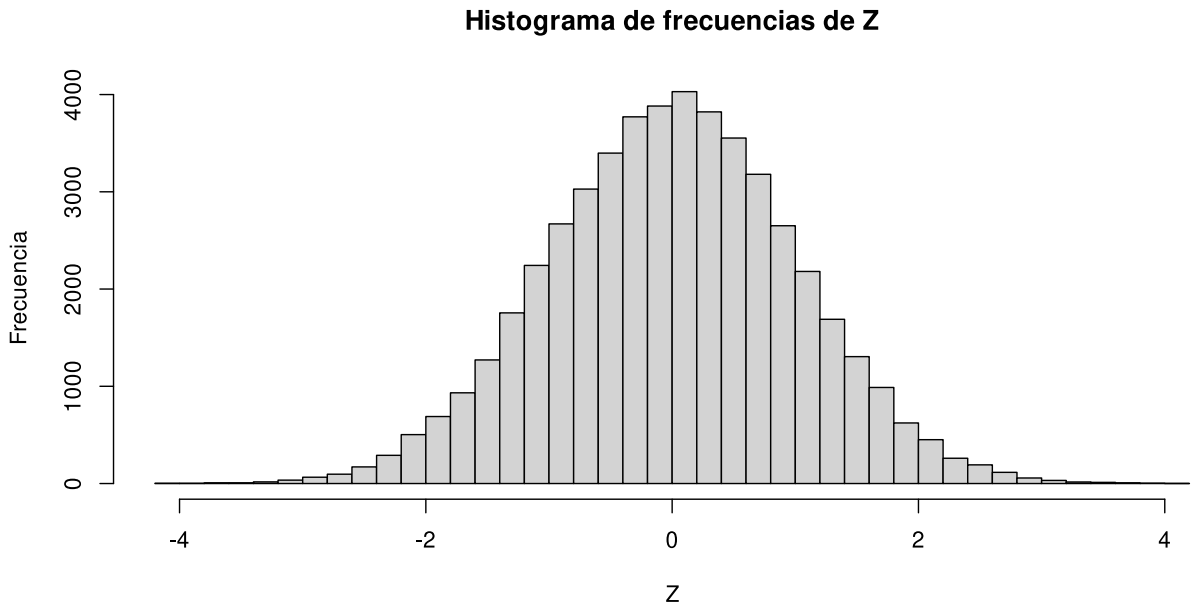
En un histograma se pueden apreciar aproximadamente:

1. La simetría de la distribución: es simétrica si tiene forma de campana de Gauss centrada en cero
2. La media en torno a cero y la altura de las barras cae hacia los extremos si la distribución es N(0,1)
3. Las modas si hay
4. La dispersión de los datos
5. La varianza se aproxima a 1 (68 % de los valores entre -1 y 1), pero esto es más difícil de apreciar.

He usado este código para simular el histograma en R:

```r
# Dibuja el histograma de los valores simulados (Z)
hist(
  Z,
  breaks = 50,
  main = "Histograma de frecuencias de Z",
  xlab = "Z",
  ylab = "Frecuencia"
)
```

Con él se genera un histograma con 50 intervalos de observaciones (de los valores de Z):



De la forma del histograma de mi simulación podemos decir que:

1. hay una moda en el intervalo (0, 0.2)
2. es más densa alrededor de cero, donde está la media, con forma de campana
3. la función es más o menos simétrica
4. la varianza se aproxima a uno

5. la media se aproxima a cero
6. la distribución es normal en apariencia (simétrica y en forma de campana)

Apartado 3

El ejercicio pide calcular P(Z > 1.645), es decir, la proporción de valores de Z de la simulación que son mayores que 1.645.

El código `R` para calcularla es:

```r
# Calcula la P(Z > 1.645)
probabilidad <- sum(Z > 1.645) / length(Z)
# Muestra la P(Z > 1.645)
probabilidad
```

Los resultados cambian al ejecutar el código en ordenadores distintos o cambiando el nombre del archivo; he obtenido los resultados 0.05124 y 0.04988 en las dos últimas simulaciones. Ambos resultados de P(Z > 1.645) son muy próximos a 0.05, el resultado teórico esperado, es decir, parece que está correcto.

Código fuente

Para la simulación he usado el programa sugerido por el equipo docente, *Rstudio*, entorno de desarrollo integrado (IDE) para el lenguaje de programación R.

El código completo que he usado para esta PEC, en lenguaje de programación `R`, es:

```r
# PEC 2024-2025
# Alejandro Muñoz Fernández
# Versión: 202504251218

## Código del apartado 1
#
# Número de observaciones; piden 50000
n <- 50000

# Genera n/2 pares de variables uniformes U₁ y U₂, cada una con 25000 variables
set.seed(123)  # Reproduce una secuencia de números aleatorios
U1 <- runif(n / 2)
U2 <- runif(n / 2)

# Aplica la transformación de Box-Müller (conforme a la página 199 del libro)
# Parte de la raíz cuadrada
raiz <- sqrt(-2 * log(U2))
# Parte del ángulo 0 (zeta, teta o theta o como se llame)
angulo <- 2 * pi * U1
X <- raiz * cos(angulo)
Y <- raiz * sin(angulo)

# Combina X e Y para obtener las 50000 observaciones
Z <- c(X, Y)

## Código del apartado 2
#
# Dibuja el histograma de los valores simulados (Z)
hist(
  Z,
  breaks = 50,
  main = "Histograma de frecuencias de Z",
  xlab = "Z",
  ylab = "Frecuencia"
)
```

```r
## Código del apartado 3
#
# Calcula la P(Z > 1.645)
probabilidad <- sum(Z > 1.645) / length(Z)
# Muestra la P(Z > 1.645)
probabilidad

# Muestra las estadísticas descriptivas de la simulación;
# que no pide el ejercicio, pero me vale para comprobar
# que los datos son coherentes
summary(Z)
# varianza
var(Z)
```