

ACTIVIDAD N°6

Evaluar empíricamente las propiedades del generador Mersenne twister y compararlo con otros generadores.

Generar una librería de números aleatorios en R

Nombre Completo: David León Callohuanca Condori

Curso: Estadística Computacional

Carrera: Ingeniería Estadística e informática

Universidad: Universidad Nacional del Altiplano

Fecha: 28 de Abril 2025

En la siguiente actividad se buscará comparar 2 generadores diferentes:

- Mersenne twister
- LCG

El cual busca evaluar empíricamente que tan "aleatorios" que son los números que generan ambos, usando:

- Cálculo de media y varianza.
- Histograma para visualizar la distribución.
- Prueba de Kolmogorov-Smirnov (KS) para medir cuán uniforme es la distribución generada.

Finalmente, se muestran los resultados lado a lado para compararlos fácilmente.

FUNCIONALIDADES

1. FUNCION `generador_lgs()`: Esta función simula un LCG para generar números pseudoaleatorios normalizados entre:

- `semi`: semilla inicial.
- `a,b,c`: constantes del generador LCG (multiplicador, incremento, módulo).
- `d`: cuántos números generar.

Luego, dependiendo del 'contexto' (talla o peso)

- Si es "talla", los números aleatorios se asignan a tallas: XS, X, M, L, XL.
- Si es "peso", transforma los números a valores entre 50 y 100 kilogramos

2. FUNCION `generador_mersenne()`: Esta función usa el generador Mersenne Twister (`runif()`) que es estándar en R.

- También usa una semilla (`set.seed(semi)`) para reproducibilidad.
- Genera `d` números aleatorios entre 0 y 1.
- Transforma esos números igual que la función LCG: según talla o peso.

3. FUNCION `evaluar_generador()`: Esta función evalúa empíricamente los datos generados:

- Imprime la media.
- Imprime la varianza.
- Dibuja un histograma para ver la distribución de los datos.

- Realiza una prueba de Kolmogorov-Smirnov (KS test) para comparar contra una distribución uniforme [0,1].

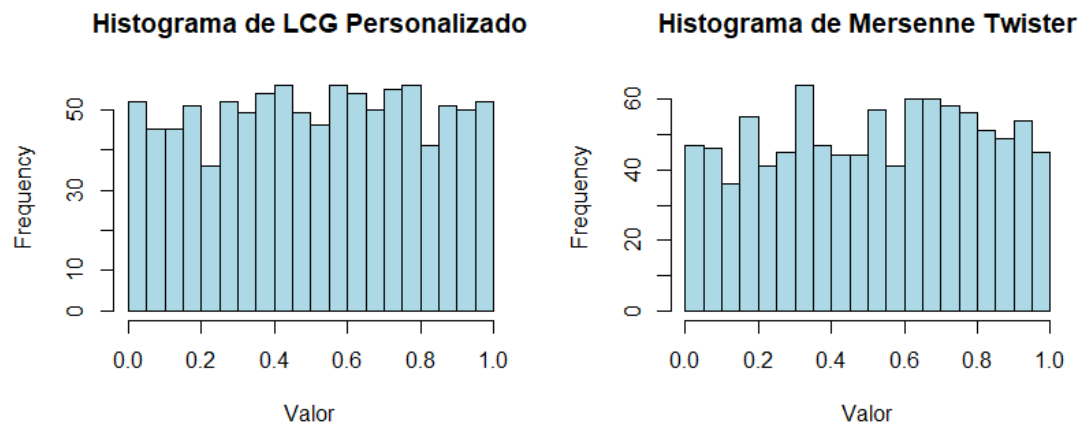
CODIGO:

```
1      #' @param contexto "talla" o "peso"
2      #' @param semi Semilla inicial
3      #' @param a Multiplicador LCG
4      #' @param b Incremento LCG
5      #' @param c Modulo LCG
6      #' @param d Numero de datos
7      #'
8      #' @return Vector de tallas o pesos
9      #' @export
10     generador_lgs <- function(contexto = "talla", semi, a, b, c, d) {
11       contexto <- tolower(contexto)
12       resultados_raw <- numeric(d)
13       x <- semi
14       for (i in 1:d) {
15         x <- (a * x + b) %% c
16         resultados_raw[i] <- x / c
17       }
18       if (contexto == "talla") {
19         niveles <- c("XS", "S", "M", "L", "XL")
20         indices <- floor(resultados_raw * length(niveles)) + 1
21         indices[indices > length(niveles)] <- length(niveles)
22         resultados <- niveles[indices]
23         print("Talla")
24       } else if (contexto == "peso") {
25         resultados <- round(50 + resultados_raw * 50, 1)
26         print("Valores de peso generados")
27       } else {
28         stop("Contexto no reconocido. Usa 'talla' o 'peso'.")
29       }
30       return(resultados)
31     }
32
33
34     generador_mersenne <- function(contexto = "talla", semi, d) {
35       contexto <- tolower(contexto)
36       set.seed(semi)
37       resultados_raw <- runif(d)
38
39       if (contexto == "talla") {
40         niveles <- c("XS", "S", "M", "L", "XL")
41         indices <- floor(resultados_raw * length(niveles)) + 1
42         indices[indices > length(niveles)] <- length(niveles)
43         resultados <- niveles[indices]
44         print("Talla")
45       } else if (contexto == "peso") {
46         resultados <- round(50 + resultados_raw * 50, 1)
47         print("Valores de peso generados")
48       } else {
49         stop("Contexto no reconocido. Usa 'talla' o 'peso'.")
50       }
51       return(resultados)
52     }
53
54
55     evaluar_generador <- function(datos, nombre_generador) {
56       cat("\nResultados para", nombre_generador, "\n")
57       cat("Media:", mean(datos), "\n")
58       cat("Varianza:", var(datos), "\n")
59     }
```

```
60 hist(datos, main=paste("Histograma de", nombre_generador),
61      xlab="Valor", col="lightblue", breaks=20)
62
63
64 ks <- ks.test(datos, "punif", 0, 1)
65 print(ks)
66 }
```

Listing 1: Libreria R.

RESPUESTA:

GIT HUB: <https://github.com/austraraptor/R2.git>