

Cálculo de Probabilidades II

Cuaderno de ejercicios con simulaciones

Laboratorio

Carla Reyes Hernández

5 de enero de 2026

Versión 0.46

Prefacio

Comenzando el semestre de otoño de 2025 en cada sesión del laboratorio se ha incluido al menos un ejercicio con simulación que ilustre alguno de los temas cubiertos en la sesión. Los ejercicios se han ejecutados con el lenguaje estadístico R y su código se muestra en el documento que fue generado con R Markdown.

Cualquier error que identifique, comentario y/o sugerencia serán bienvenidos. Diríjalo a Ernesto Barrios <ebarrios at itam.mx>.

Ciudad de México, 5 de enero de 2026

Lab01 — Urna 0/1: $X = \max$, $Y = \min$

2025-08-24

Ejercicio: Máximo y mínimo de dos extracciones Bernoulli

Planteamiento

Sean $A, B \stackrel{\text{i.i.d.}}{\sim} \text{Bernoulli}(p)$ con $0 < p < 1$. Definimos

$$X = \max\{A, B\}, \quad Y = \min\{A, B\}.$$

Objetivos

1. Derivar (o recordar) la **f.m.p. conjunta** $f_{X,Y}(x,y)$ y las **marginales** f_X, f_Y .

Simulación en R

Ajusta los parámetros si lo deseas y vuelve a ejecutar la celda.

```
N <- 100000      # tamaño de simulación
p <- 0.5         # parámetro Bernoulli
seed <- 123      # semilla
set.seed(seed)
```

a) F.M.P. conjunta y marginales (empíricas vs teóricas)

```
# Simulación de A,B y construcción de X,Y
a <- rbinom(N, 1, p)
b <- rbinom(N, 1, p)
X <- pmax(a, b)
Y <- pmin(a, b)

# Rejillas y fmp empírica
xs <- 0:1; ys <- 0:1
pmf_emp <- prop.table(table(factor(X, levels = xs), factor(Y, levels = ys)))

# FMP teórica (p genérico)
pmf_th <- matrix(c((1-p)^2, 0,
                  2*p*(1-p), p^2),
                nrow = 2, byrow = TRUE,
                dimnames = list(X = as.character(xs), Y = as.character(ys)))

# Marginales
margX_emp <- rowSums(pmf_emp)
margY_emp <- colSums(pmf_emp)
margX_th <- c(`0` = (1-p)^2, `1` = 1 - (1-p)^2)
margY_th <- c(`0` = 1 - p^2, `1` = p^2)
```

```

cat("FMP empírica:\n"); print(round(pmf_emp, 4))

## FMP empírica:
##
##      0      1
## 0 0.2499 0.0000
## 1 0.5027 0.2474

cat("FMP teórica:\n"); print(round(pmf_th, 4))

## FMP teórica:
##      Y
## X      0      1
## 0 0.25 0.00
## 1 0.50 0.25

cat("Marginales empíricas X, Y:\n"); print(round(margX_emp,4)); print(round(margY_emp,4))

## Marginales empíricas X, Y:
##      0      1
## 0.2499 0.7501
##      0      1
## 0.7526 0.2474

cat("Marginales teóricas X, Y:\n"); print(round(margX_th,4)); print(round(margY_th,4))

## Marginales teóricas X, Y:
##      0      1
## 0.25 0.75
##      0      1
## 0.75 0.25

```

b) Visualización rápida

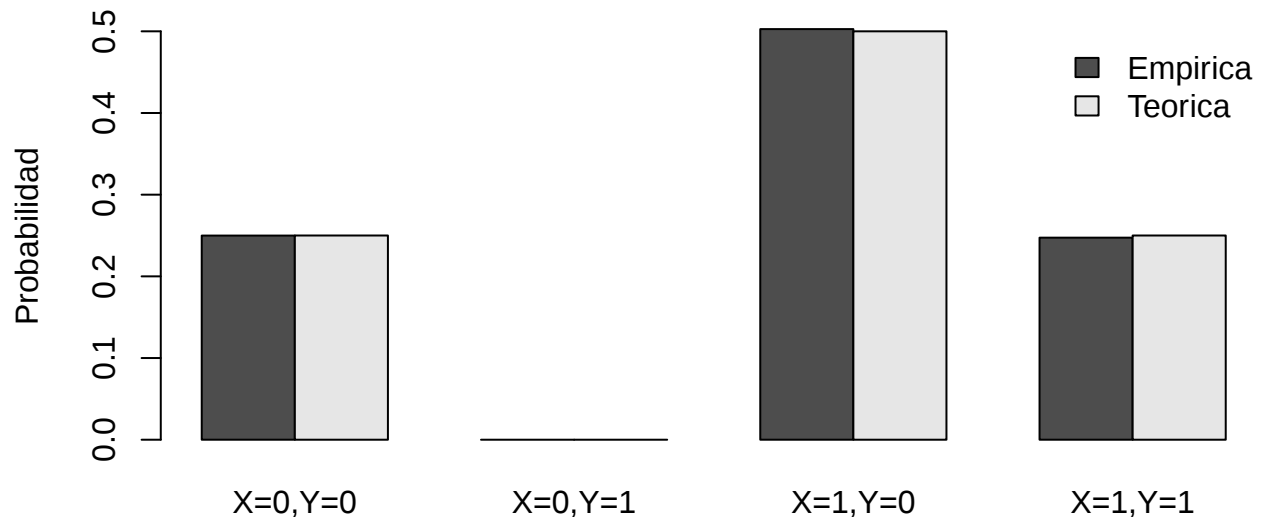
```

# Vectorizamos en el orden deseado: (X=0,Y=0), (X=0,Y=1), (X=1,Y=0), (X=1,Y=1)
labels <- c("X=0,Y=0", "X=0,Y=1", "X=1,Y=0", "X=1,Y=1")
pmf_emp_vec <- c(pmf_emp["0","0"], pmf_emp["0","1"], pmf_emp["1","0"], pmf_emp["1","1"])
pmf_th_vec <- c(pmf_th["0","0"], pmf_th["0","1"], pmf_th["1","0"], pmf_th["1","1"])

bars <- rbind(Empirica = pmf_emp_vec, Teorica = pmf_th_vec)
bp <- barplot(beside = TRUE, height = bars, names.arg = labels,
              ylab = "Probabilidad", main = "f.m.p. conjunta: empírica vs teórica")
legend("topright", legend = rownames(bars), fill = gray.colors(2), bty = "n")

```

f.m.p. conjunta: empírica vs teórica



lab02Computo — Simulación (Problema 10)

Carla Reyes

30 de agosto de 2025

Contents

Objetivo	1
Código	1
Funciones auxiliares	1
Ejecutar la simulación	2

Objetivo

Estimaremos por **simulación** la probabilidad de que **todos** los valores propios de la matriz

$$\Delta = \begin{pmatrix} a & c & 0 \\ b & a & c \\ 0 & b & a \end{pmatrix}$$

sean **reales**, cuando las entradas a, b, c se generan **independientemente** con distribución **Uniforme(-1,1)**.

Código

Funciones auxiliares

```
# Construye la matriz Delta para un triplete (a,b,c)
make_delta <- function(a, b, c) {
  matrix(c(a, c, 0,
           b, a, c,
           0, b, a), nrow = 3, byrow = TRUE)
}

# Revisa si todos los eigenvalores son reales (tolerancia numérica)
all_eigs_real <- function(M, tol = 1e-10) {
  vals <- eigen(M, only.values = TRUE)$values
  max(abs(Im(vals))) < tol
}

# Simulación directa:
# 1) Genera (a,b,c) ~ i.i.d. Unif(-1,1)
# 2) Construye Delta
# 3) Calcula eigenvalores y verifica si todos son reales
# 4) Repite N veces y devuelve la proporción (p-hat)
simulate_prob <- function(N = 10000) {
```

```

a <- runif(N, -1, 1)
b <- runif(N, -1, 1)
c <- runif(N, -1, 1)

ok <- logical(N)
for (i in seq_len(N)) {
  M <- make_delta(a[i], b[i], c[i])
  ok[i] <- all_eigs_real(M)
}

mean(ok) # devuelve solo p-hat
}

```

Ejecutar la simulación

Ajusta N para mayor precisión (más grande = más lento).

```

N <- 10000
p_hat <- simulate_prob(N)
p_hat

```

```
## [1] 0.499
```

lab03 — Problema 7: Simulación de $P(X = 1 \mid Y > 0)$

Carla Reyes

7 de septiembre de 2025

Contents

Enunciado	1
Código	1
Funciones	1
Correr la simulación	1

Enunciado

Sea $X \in \{-1, 1\}$ con $\mathbb{P}(X = 1) = 1/3$ y $Z \sim N(0, 1)$ independiente de X . Defina $Y = Z + X$. Estimar por simulación $\mathbb{P}(X = 1 \mid Y > 0)$.

Código

Funciones

```
simulate_prob7 <- function(N = 1e6) {  
  # X: +1 con prob 1/3, -1 con prob 2/3  
  X <- sample(c(1, -1), size = N, replace = TRUE, prob = c(1/3, 2/3))  
  # Z ~ N(0,1)  
  Z <- rnorm(N)  
  Y <- Z + X  
  idx <- Y > 0  
  n_pos <- sum(idx)  
  if (n_pos == 0) stop("No hubo Y>0; incrementa N.")  
  phat <- mean(X[idx] == 1)  
  list(N = N, n_pos = n_pos, p_hat = phat)  
}
```

Correr la simulación

Puedes ajustar N si quieres mayor precisión (más grande = más lento).

```
N <- 1e6  
res <- simulate_prob7(N)  
res  
  
## $N  
## [1] 1e+06  
##  
## $n_pos
```

```
## [1] 387298
##
## $p_hat
## [1] 0.7262366
```

Reporte claro

```
cat(sprintf("n total: %d | n(Y>0): %d\n", res$N, res$n_pos))
## n total: 1000000 | n(Y>0): 387298
cat(sprintf("Estimación: p_hat = %.6f\n", res$p_hat))
## Estimación: p_hat = 0.726237
```

Lab 05 Cómputo — Simulación (Problema 5)

Carla Reyes

Problema 5 del laboratorio 5

Una urna contiene tres bolas numeradas 1, 2 y 3. Se extraen **dos bolas sin reemplazo**. Sea X el número de la **primera** bola extraída y Y el número de la **segunda**. Se pide calcular $\text{Cov}(X, Y)$ y ρ_{XY} .

Simulación

```
set.seed(123456)

B <- 2e5 # número de repeticiones
# Simulamos extracciones sin reemplazo de {1,2,3}
X <- integer(B)
Y <- integer(B)

for (b in seq_len(B)) {
  s <- sample(1:3, size = 2, replace = FALSE)
  X[b] <- s[1]
  Y[b] <- s[2]
}

# Estimadores empíricos
EX <- mean(X);
EY <- mean(Y)
VarX <- var(X);
VarY <- var(Y)
CovXY <- cov(X, Y)
RhoXY <- cor(X, Y)

c(EX=EX, EY=EY, VarX=VarX, VarY=VarY, CovXY=CovXY, RhoXY=RhoXY)

##          EX          EY          VarX          VarY          CovXY          RhoXY
## 2.0016250 1.9987950 0.6658657 0.6678169 -0.3336797 -0.5003890
```

Conclusión

Tanto la derivación analítica como la simulación confirman que:

- $\text{Cov}(X, Y) = -\frac{1}{3}$
- $\rho_{XY} = -\frac{1}{2}$

La **correlación negativa** surge de que **no hay reemplazo**: si la primera bola es alta, se reduce la probabilidad de que la segunda también lo sea.

lab06Computo — Simulación y verificación (Problema 4)

Carla Reyes

2025-09-28

Contents

1	Enunciado (Problema 4)	1
2	Construcción de la distribución y simulación	1
2.1	Tabla de probabilidades	1
2.2	Simulación	2
3	Verificación analítica: medias, covarianza y dependencia	2
3.1	Demostración de dependencia (no independencia)	3
4	Comentarios finales	4

1 Enunciado (Problema 4)

Sean Y_1 y Y_2 variables aleatorias discretas con f.d.p. conjunta dada por la tabla:

	$Y_1 = -1$	$Y_1 = 0$	$Y_1 = 1$
$Y_2 = -1$	1/16	3/16	1/16
$Y_2 = 0$	3/16	0	3/16
$Y_2 = 1$	1/16	3/16	1/16

Se pide **mostrar que** Y_1 y Y_2 **son dependientes pero tienen covarianza 0**, y realizar una **simulación** que ilustre el resultado.

2 Construcción de la distribución y simulación

```
library(dplyr)
```

2.1 Tabla de probabilidades

```
# Valores posibles
valores <- expand.grid(
  Y1 = c(-1, 0, 1),
  Y2 = c(-1, 0, 1)
)

# Probabilidades según la tabla (orden: por filas de Y2: -1,0,1; y por columnas de Y1: -1,0,1)
```

```
valores$prob <- c(1,3,1, 3,0,3, 1,3,1) / 16
```

```
valores
```

```
##   Y1 Y2  prob
## 1 -1 -1 0.0625
## 2  0 -1 0.1875
## 3  1 -1 0.0625
## 4 -1  0 0.1875
## 5  0  0 0.0000
## 6  1  0 0.1875
## 7 -1  1 0.0625
## 8  0  1 0.1875
## 9  1  1 0.0625
```

Comprobamos que suma a 1:

```
sum(valores$prob)
```

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

2.2 Simulación

```
set.seed(20250928) # fija semilla para replicabilidad
```

```
n <- 20000 # tamaño de muestra simulada
```

```
# Muestreo de pares (Y1, Y2) con sus probabilidades
```

```
sim_idx <- sample.int(n = nrow(valores), size = n, replace = TRUE, prob = valores$prob)
```

```
sim <- valores[sim_idx, c("Y1", "Y2")]
```

```
head(sim)
```

```
##      Y1 Y2
## 2      0 -1
## 4     -1  0
## 2.1    0 -1
## 7     -1  1
## 3      1 -1
## 4.1   -1  0
```

3 Verificación analítica: medias, covarianza y dependencia

```
# Marginales exactos
```

```
marg_Y1 <- sim %>%
  count(Y1, name = "n") %>%
  mutate(p = n / sum(n))
```

```
marg_Y2 <- sim %>%
  count(Y2, name = "n") %>%
  mutate(p = n / sum(n))
```

```
marg_Y1; marg_Y2
```

```
##      Y1      n      p
## 1 -1 6222 0.31110
## 2  0 7527 0.37635
## 3  1 6251 0.31255

##      Y2      n      p
## 1 -1 6257 0.31285
## 2  0 7439 0.37195
## 3  1 6304 0.31520
```

```
# Medias y covarianza empíricas (se aproximan a las verdaderas)
media_Y1_emp <- mean(sim$Y1)
media_Y2_emp <- mean(sim$Y2)
cov_emp <- cov(sim$Y1, sim$Y2)

list(media_Y1_emp = media_Y1_emp,
     media_Y2_emp = media_Y2_emp,
     covarianza_empirica = cov_emp)
```

```
## $media_Y1_emp
## [1] 0.00145
##
## $media_Y2_emp
## [1] 0.00235
##
## $covarianza_empirica
## [1] 0.002296707
```

Resultado teórico esperado: $E[Y_1] = 0$, $E[Y_2] = 0$ y $\text{Cov}(Y_1, Y_2) = 0$. Abajo confirmamos de forma exacta con operaciones simbólicas sobre la distribución.

```
# Cálculo exacto con la tabla (sin depender de la simulación)
tab <- valores

EY1 <- sum(tab$Y1 * tab$prob)
EY2 <- sum(tab$Y2 * tab$prob)
EY1Y2 <- sum(tab$Y1 * tab$Y2 * tab$prob)
CovY1Y2 <- EY1Y2 - EY1 * EY2

data.frame(EY1, EY2, EY1Y2, CovY1Y2)
```

```
##      EY1 EY2 EY1Y2 CovY1Y2
## 1      0      0      0      0
```

3.1 Demostración de dependencia (no independencia)

Para independencia debería cumplirse $P(Y_1 = y_1, Y_2 = y_2) = P(Y_1 = y_1) P(Y_2 = y_2)$ para **todos** los pares (y_1, y_2) . Basta encontrar un contraejemplo:

```
# Prob conjunto de (Y1=0, Y2=0):
p_joint_00 <- tab %>% filter(Y1==0, Y2==0) %>% pull(prob)

# Prob marginales
p_Y1_0 <- tab %>% filter(Y1==0) %>% summarise(p = sum(prob)) %>% pull(p)
p_Y2_0 <- tab %>% filter(Y2==0) %>% summarise(p = sum(prob)) %>% pull(p)

data.frame(
```

```

P_Y1_eq_0 = p_Y1_0,
P_Y2_eq_0 = p_Y2_0,
P_joint_00 = p_joint_00,
Prod_marginales = p_Y1_0 * p_Y2_0
)

```

```

##   P_Y1_eq_0 P_Y2_eq_0 P_joint_00 Prod_marginales
## 1      0.375      0.375          0          0.140625

```

Observa que $P(Y_1 = 0, Y_2 = 0) = 0$ mientras que $P(Y_1 = 0)P(Y_2 = 0) > 0$. Luego **no** son independientes (aunque la covarianza sea 0).

4 Comentarios finales

- La covarianza nula **no** implica independencia.
- Este ejemplo ilustra cómo dos variables pueden ser dependientes (fallan las igualdades de independencia) pero tener **correlación cero**.

lab07Computo — Simulación del Problema 7

Carla Reyes

05 Oct 2025

Enunciado

En un centro de atención telefónica, cada agente atiende llamadas de forma continua hasta que decide tomar un descanso.

- La **duración de cada llamada** es i.i.d. Exponencial con parámetro $\lambda = 0.2$ y es independiente de las demás.
- El **número total de llamadas** N que atiende el agente es **por lo menos 5** y se distribuye **Geométrica** con parámetro $p = 0.3$, **independiente** de las duraciones.
- Interpretamos esto como $N = 5 + G$, donde $G \sim \text{Geom}(p)$ con soporte $\{0, 1, 2, \dots\}$ (número de fracasos antes del primer éxito).

Se pide calcular el **valor esperado** y la **varianza** de la **duración total** $T = \sum_{i=1}^N X_i$; y verificar por simulación.

Derivación teórica

Sea $X \sim \text{Exp}(\lambda)$, entonces $\mathbb{E}[X] = \frac{1}{\lambda}$ y $\text{Var}(X) = \frac{1}{\lambda^2}$.

Para $G \sim \text{Geom}(p)$ en $\{0, 1, 2, \dots\}$: $\mathbb{E}[G] = \frac{1-p}{p}$ y $\text{Var}(G) = \frac{1-p}{p^2}$.

Como $N = 5 + G$,

$$\mathbb{E}[N] = 5 + \frac{1-p}{p}, \quad \text{Var}(N) = \frac{1-p}{p^2}.$$

Usando la fórmula de suma aleatoria con independencia ($T = \sum_{i=1}^N X_i$):

$$\mathbb{E}[T] = \mathbb{E}[N] \mathbb{E}[X], \quad \text{Var}(T) = \mathbb{E}[N] \text{Var}(X) + \text{Var}(N) (\mathbb{E}[X])^2.$$

Sustituyendo $\mathbb{E}[X] = 1/\lambda$ y $\text{Var}(X) = 1/\lambda^2$:

$$\mathbb{E}[T] = \frac{\mathbb{E}[N]}{\lambda}, \quad \text{Var}(T) = \frac{\mathbb{E}[N] + \text{Var}(N)}{\lambda^2}.$$

Con $p = 0.3$ y $\lambda = 0.2$:

```
p <- 0.3
lam <- 0.2
EN <- 5 + (1 - p)/p
VarN <- (1 - p)/p^2
ET_theory <- EN / lam
VarT_theory <- (EN + VarN) / lam^2
list(E_N = EN, Var_N = VarN, E_T = ET_theory, Var_T = VarT_theory,
     sd_T = sqrt(VarT_theory))
```

```
## $E_N
## [1] 7.333333
##
## $Var_N
## [1] 7.777778
##
## $E_T
## [1] 36.66667
##
## $Var_T
## [1] 377.7778
##
## $sd_T
## [1] 19.43651
```

Resultados teóricos esperados: - $\mathbb{E}[N] = 7.\overline{3}$ - $\text{Var}(N) = 7.\overline{7}$ - $\mathbb{E}[T] = 36.\overline{6}$ - $\text{Var}(T) = 377.\overline{7}$

(Unidades en minutos si la tasa λ está dada en llamadas por minuto.)

Verificación por simulación (Monte Carlo)

Simulamos B réplicas. En cada réplica: 1. Generamos $G \sim \text{Geom}(p)$ y calculamos $N = 5 + G$. 2. Sumamos N duraciones exponenciales i.i.d. $X_i \sim \text{Exp}(\lambda)$. 3. Guardamos $T = \sum_{i=1}^N X_i$.

```
set.seed(7)
B <- 1e5

# Muestras de N = 5 + G, con G ~ Geom(p) (soporte 0,1,2,...)
N <- 5 + rgeom(B, prob = p)

# Función que simula una suma de exponentiales con N_i términos
sim_T <- function(n, rate) {
  if (n <= 0) return(0)
  sum(rexp(n, rate = rate))
}

# Vector con T simulados
T_vals <- sapply(N, sim_T, rate = lam)

c(
  mean_T = mean(T_vals),
  var_T = var(T_vals),
  mean_N = mean(N),
  var_N = var(N)
)
```

```
##      mean_T      var_T      mean_N      var_N
## 36.727848 381.684951  7.327820  7.755432
```

Comparación teoría vs simulación

```
data.frame(
  Magnitud = c("E[N]", "Var[N]", "E[T]", "Var[T]"),
  Teoría = c(EN, VarN, ET_theory, VarT_theory),
```

```
Simulación = c(mean(N), var(N), mean(T_vals), var(T_vals))
)
```

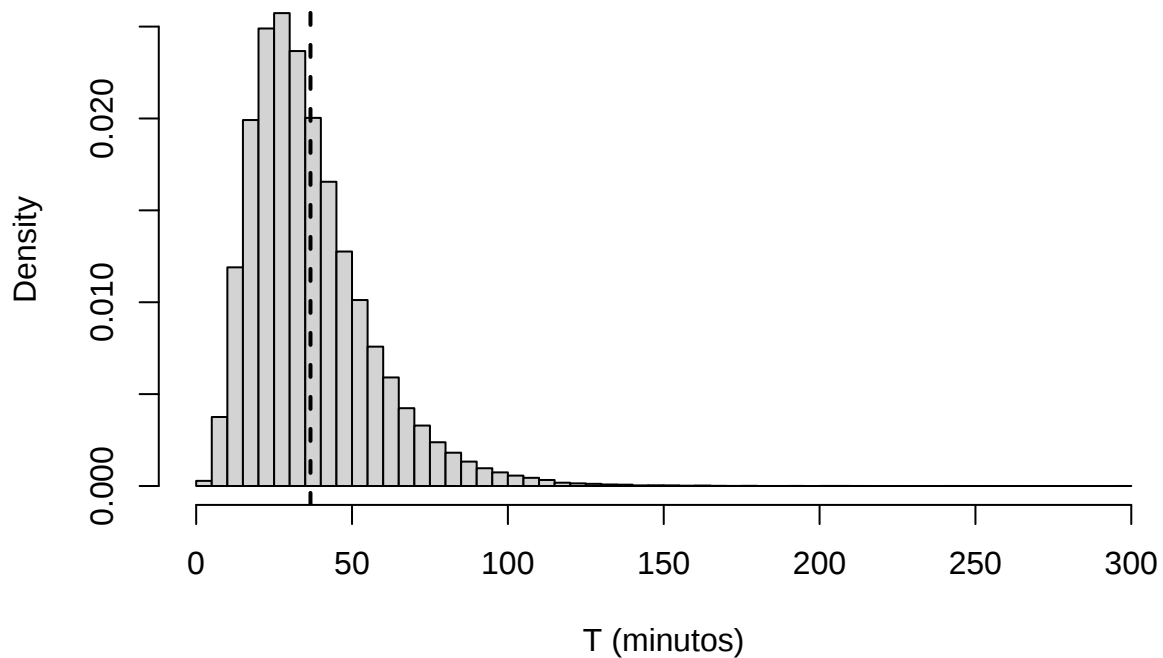
##	Magnitud	Teoría	Simulación
## 1	E[N]	7.333333	7.327820
## 2	Var[N]	7.777778	7.755432
## 3	E[T]	36.666667	36.727848
## 4	Var[T]	377.777778	381.684951

Visualización

Distribución aproximada de T (histograma) y una marca en la media teórica:

```
hist(T_vals, breaks = 60, main = "Distribución simulada de T",
     xlab = "T (minutos)", probability = TRUE)
abline(v = ET_theory, lwd = 2, lty = 2)
```

Distribución simulada de T



labComputo08

Carla Reyes

2025-10-12

Sesión 08 — Simulación del Problema 1

Considere un dado honesto lanzado 7 veces de manera independiente. Calcule la probabilidad de que **cada una de las caras 2,3,4,5,6** aparezca **exactamente una vez**, **condicionando** a que la cara **1** salió **exactamente dos veces**.

Simulación (200000 intentos)

Hacemos 200000 intentos. En cada intento simulamos 7 lanzamientos. Si en los 7 hay **exactamente dos** caras **1**, revisamos si las otras 5 caras son **2,3,4,5,6 cada una una sola vez**. La estimación es:

$$\hat{p} = \frac{\#\{\text{intentos con 2 unos y con } \{2, 3, 4, 5, 6\}\}}{\#\{\text{intentos con exactamente 2 unos}\}}.$$

```
B <- 200000

# Simulamos B intentos de 7 lanzamientos cada uno
lanz <- matrix(sample(1:6, size = 7*B, replace = TRUE), ncol = 7)

# Filtramos los intentos con exactamente dos "1"
cond_idx <- rowSums(lanz == 1) == 2
n_cond <- sum(cond_idx)

# Entre los condicionados, verificamos si las otras 5 caras son {2,3,4,5,6}
es_exacto_2a6 <- function(fila) {
  v <- fila[fila != 1] # quita los "1", quedan 5 valores
  # ¿Son exactamente 2,3,4,5,6 cada uno una sola vez?
  all(sort(v) == 2:6)
}

ok <- apply(lanz[cond_idx, , drop = FALSE], 1, es_exacto_2a6)
n_ok <- sum(ok)

p_hat <- if (n_cond > 0) n_ok / n_cond else NA_real_

data.frame(
  intentos_totales = B,
  intentos_condicionados = n_cond,
  exitos = n_ok,
  p_estimada = p_hat
)

## intentos_totales intentos_condicionados exitos p_estimada
```

## 1	2e+05	47128	1809	0.03838482
------	-------	-------	------	------------

lab09Computo

Carla Reyes

2025-10-19

Problema 1 del Laboratorio 9

Sea $(Y_1 \sim \text{Bin}(n_1, p_1=0.2))$ y $(Y_2 \sim \text{Bin}(n_2, p_2=0.8))$, independientes. Hay que estudiar la distribución de $[Z, Y_1 + n_2 - Y_2]$

Resultado teórico: Como $(n_2 - Y_2 \sim \text{Bin}(n_2, 0.2))$ y es independiente de $(Y_1 \sim \text{Bin}(n_1, 0.2))$, se tiene que $(Z = (n_2 - Y_2) + Y_1 \sim \text{Bin}(n_1 + n_2, 0.2))$.

Parámetros

```
set.seed(1704)
n1  <- 5
n2  <- 8
p1  <- 0.2
p2  <- 0.8
reps <- 1e5 # número de muestras
```

Simulación

```
# Genera muestras
Y1 <- rbinom(reps, size = n1, prob = p1)
Y2 <- rbinom(reps, size = n2, prob = p2)

Z  <- Y1 + n2 - Y2
```

Resumen empírico vs teórico

```
# Soporte teórico de Z
k <- 0:(n1 + n2)

# Probabilidades teóricas si  $Z \sim \text{Bin}(n_1+n_2, 0.2)$ 
theo_probs <- dbinom(k, size = n1 + n2, prob = 0.2)

# Probabilidades empíricas por simulación
obs_tab  <- table(factor(Z, levels = k))
obs_probs <- as.numeric(obs_tab) / sum(obs_tab)

# Medias y varianzas
emp_mean <- mean(Z)
emp_var  <- var(Z)
```

```

th_mean <- (n1 + n2) * 0.2
th_var <- (n1 + n2) * 0.2 * 0.8

list(
  media_empirica = emp_mean,
  var_empirica = emp_var,
  media_teorica = th_mean,
  var_teorica = th_var
)

## $media_empirica
## [1] 2.60503
##
## $var_empirica
## [1] 2.08307
##
## $media_teorica
## [1] 2.6
##
## $var_teorica
## [1] 2.08

```

Comparación visual (barras)

```

library(ggplot2)

df_plot <- data.frame(
  k = k,
  Empirica = obs_probs,
  Teorica = theo_probs
)

ggplot(df_plot, aes(x = k)) +
  geom_col(aes(y = Empirica), width = 0.7, alpha = 0.6) +
  geom_point(aes(y = Teorica)) +
  labs(
    title = "Distribución de  $Z = Y_1 + n_2 - Y_2$ ",
    subtitle = paste0("n1=", n1, ", n2=", n2, ", p1=0.2, p2=0.8; réplicas=", reps),
    x = "k",
    y = "Probabilidad"
  ) +
  theme_minimal()

```

Distribución de $Z = Y1 + n2 - Y2$

$n1=5, n2=8, p1=0.2, p2=0.8$; réplicas= $1e+05$

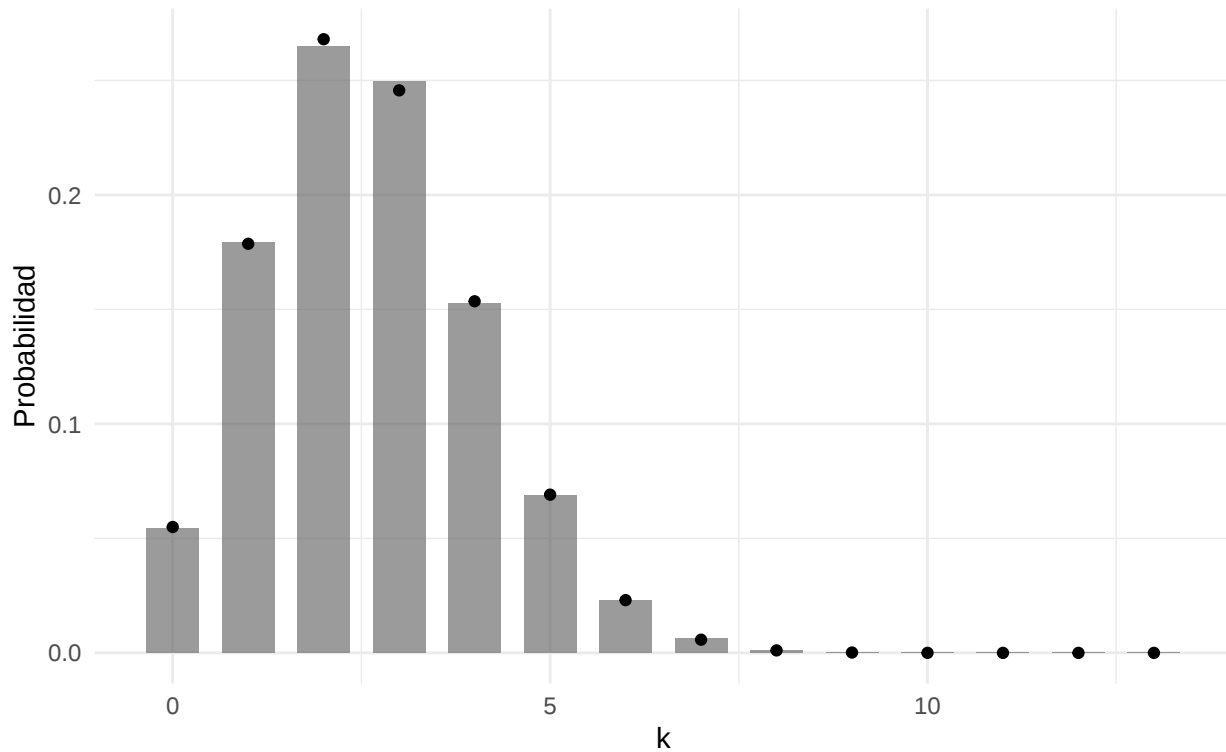


Tabla compacta

```
head(data.frame(
  k = k,
  Prob_Empirica = round(obs_probs, 5),
  Prob_Teorica = round(theo_probs, 5)
), 15)
```

##	k	Prob_Empirica	Prob_Teorica
## 1	0	0.05425	0.05498
## 2	1	0.17909	0.17867
## 3	2	0.26501	0.26801
## 4	3	0.24962	0.24567
## 5	4	0.15264	0.15355
## 6	5	0.06888	0.06910
## 7	6	0.02300	0.02303
## 8	7	0.00614	0.00576
## 9	8	0.00122	0.00108
## 10	9	0.00014	0.00015
## 11	10	0.00001	0.00001
## 12	11	0.00000	0.00000
## 13	12	0.00000	0.00000
## 14	13	0.00000	0.00000

lab10Computo— Transformación Integral de Probabilidad

Carla Reyes

2025-10-29

Objetivo

Simular y verificar empíricamente el **Teorema de la Transformación Integral de Probabilidad** usando una **Exponencial** con parámetro $\lambda > 0$: 1) Si $U \sim \text{Unif}(0, 1)$ y definimos $X = F^{-1}(U)$, entonces $X \sim F$. 2) Si $X \sim F$ y definimos $U = F(X)$, entonces $U \sim \text{Unif}(0, 1)$.

Trabajaremos con la distribución **Exponencial**(λ) cuya función de distribución acumulada (FDA) es:

$$F(x) = 1 - e^{-\lambda x}, \quad x \geq 0, \lambda > 0.$$

La inversa generalizada de F es:

$$F^{-1}(u) = -\frac{1}{\lambda} \log(1 - u), \quad u \in (0, 1).$$

Parámetros de simulación

```
n <- 100000      # tamaño de muestra
lambda <- 5      # parámetro de la Exponencial
```

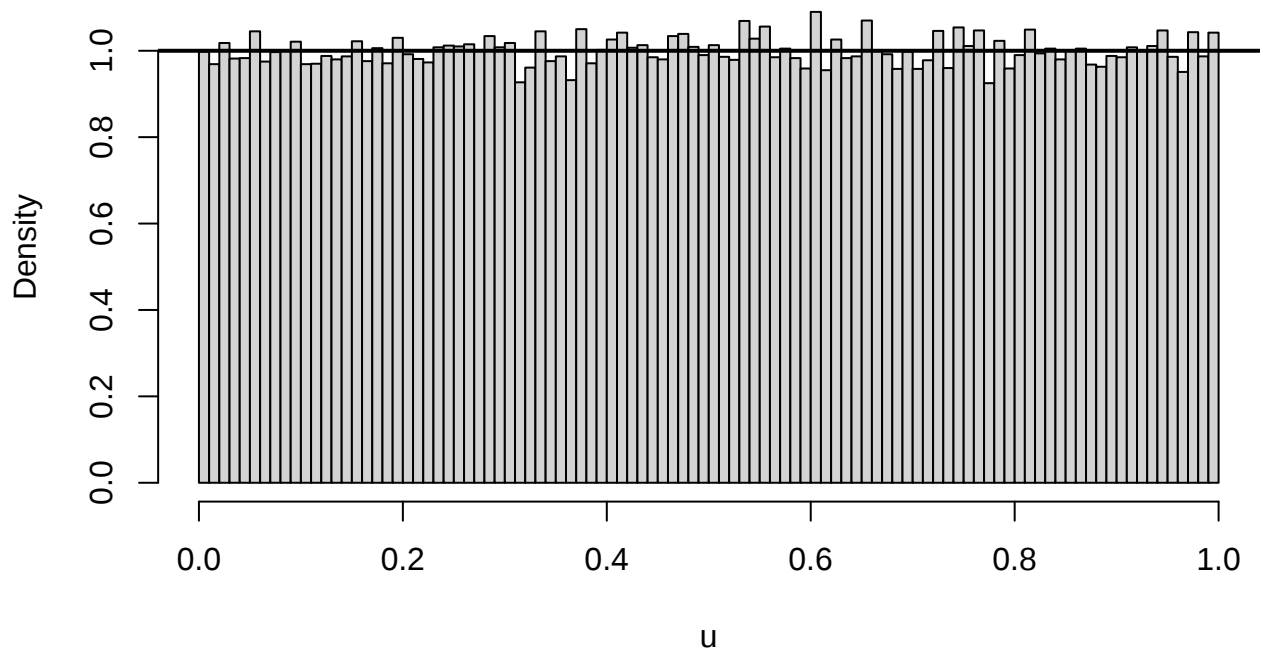
Parte A: De Uniforme(0,1) a Exponencial(λ)

Idea: Generar $U_1, \dots, U_n \stackrel{iid}{\sim} \text{Unif}(0, 1)$ y transformar $X_i = F^{-1}(U_i) = -\log(1 - U_i)/\lambda$. Se espera $X \sim \text{Exp}(\lambda)$.

```
# Muestras uniformes
U <- runif(n)

# Muestras visualización
hist(U, breaks = 80, probability = TRUE,
     main = "Parte A (previa): U ~ Unif(0,1) antes de transformar",
     xlab = "u", xlim = c(0, 1))
abline(h = 1, lwd = 2) # densidad de Unif(0,1)
```

Parte A (previa): $U \sim \text{Unif}(0,1)$ antes de transformar



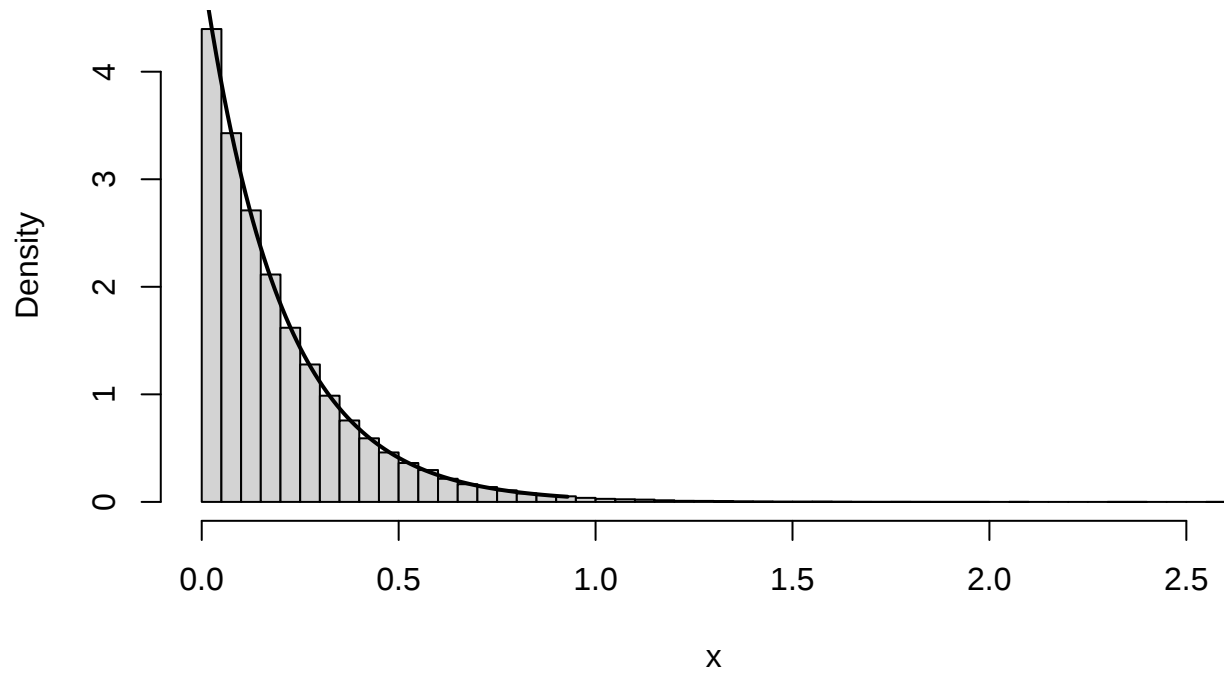
```
c(mean = mean(U), var = var(U),
  expected_mean = 0.5, expected_var = 1/12)

##          mean          var expected_mean expected_var
## 0.50052451 0.08313908 0.50000000 0.08333333

# Transformación inversa: Exp(lambda)
X_from_U <- -log(1 - U) / lambda

# Comprobamos
# 1) Histograma con densidad teórica
hist(X_from_U, breaks = 80, probability = TRUE, main = "Parte A: X desde U vía F^{-1}",
  xlab = "x")
curve(lambda * exp(-lambda * x), from = 0, to = quantile(X_from_U, 0.99),
  add = TRUE, lwd = 2)
```

Parte A: X desde U vía F^{-1}



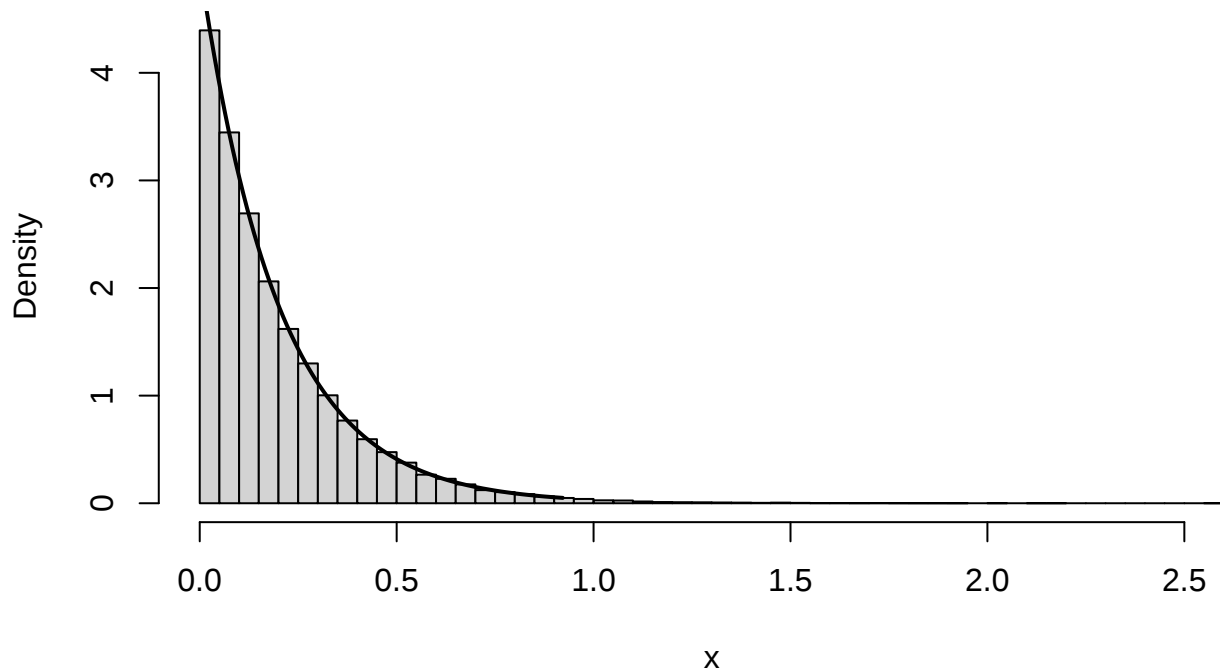
Parte B: De Exponencial(λ) a Uniforme(0,1)

Idea: Generar $X_1, \dots, X_n \stackrel{iid}{\sim} \text{Exp}(\lambda)$ y transformar $U_i = F(X_i) = 1 - e^{-\lambda X_i}$. Se espera $U \sim \text{Unif}(0, 1)$.

```
# Muestras exponenciales directas
X <- rexp(n, rate = lambda)

# Muestras visualización
hist(X, breaks = 80, probability = TRUE,
     main = "Parte B (previa): X ~ Exp(lambda) antes de transformar",
     xlab = "x")
curve(lambda * exp(-lambda * x), from = 0, to = quantile(X, 0.99),
     add = TRUE, lwd = 2)
```

Parte B (previa): $X \sim \text{Exp}(\lambda)$ antes de transformar



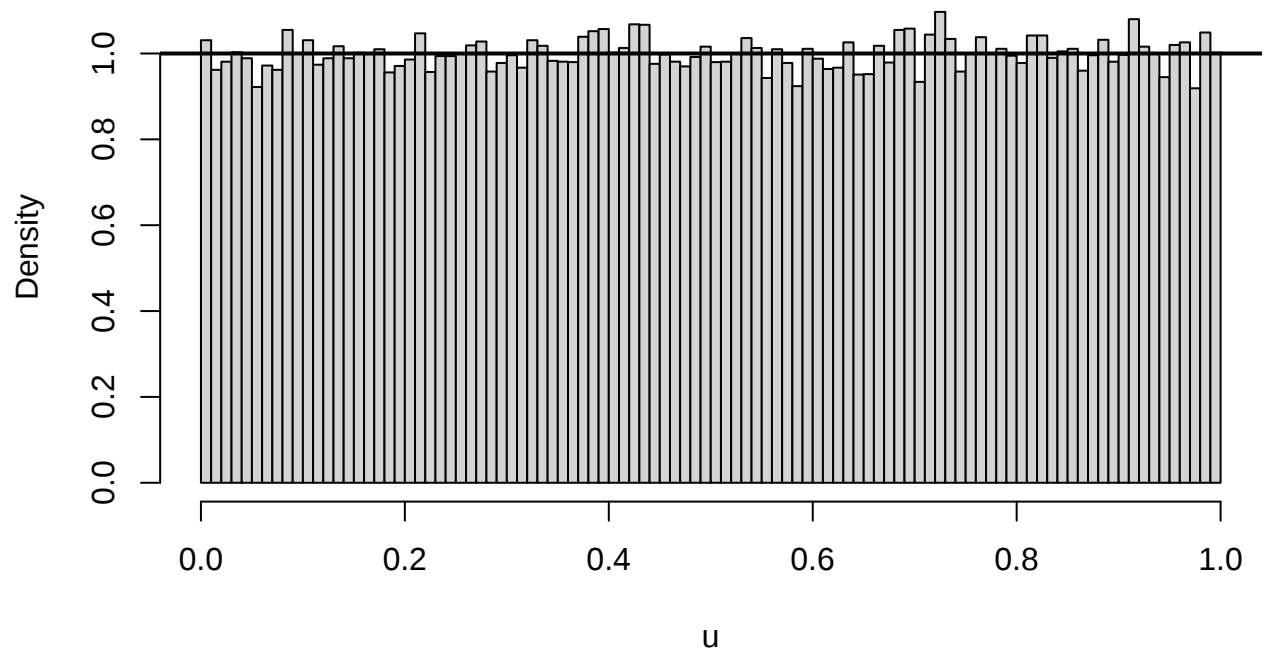
```
c(mean = mean(X), var = var(X),
  expected_mean = 1/lambda, expected_var = 1/(lambda^2))

##          mean          var expected_mean expected_var
## 0.20058818 0.04011564 0.20000000 0.04000000

# Transformación directa:  $U = F(X)$ 
U_from_X <- 1 - exp(-lambda * X)

# Comprobamos
# 1) Histograma con densidad teórica
hist(U_from_X, breaks = 80, probability = TRUE, main = "Parte B:  $U = F(X)$  desde  $X \sim \text{Exp}(\lambda)$ ",
  xlab = "u", xlim = c(0,1))
abline(h = 1, lwd = 2) # densidad de  $\text{Unif}(0,1)$ 
```

Parte B: $U = F(X)$ desde $X \sim \text{Exp}(\text{lambda})$



Anexo: fórmulas clave

- Exponencial(λ): $f(x) = \lambda e^{-\lambda x}$, $F(x) = 1 - e^{-\lambda x}$, $x \geq 0$.
- Inversa: $F^{-1}(u) = -\log(1 - u)/\lambda$, $u \in (0, 1)$.
- Transformación directa: $U = F(X) \Rightarrow U \sim \text{Unif}(0, 1)$ si $X \sim F$.

lab11Computo — Distribución de $U = e^{-(X+Y)}$

Carla Reyes

2025-11-02

Objetivo

En este laboratorio estudiamos la variable

$$U = e^{-(X+Y)},$$

donde (X, Y) tiene densidad conjunta

$$f_{X,Y}(x,y) = e^{-(x+y)} \mathbf{1}_{(0,\infty)}(x) \mathbf{1}_{(0,\infty)}(y).$$

Esto corresponde a dos variables independientes $X \sim \text{Exp}(1)$ y $Y \sim \text{Exp}(1)$. Vamos a:

1. Simular muchas realizaciones de U .
2. Graficar el histograma empírico de U .
3. Dibujar encima la densidad teórica de U .
4. Comparar la esperanza empírica con el valor teórico.

Paso 1: generar muestras de X, Y y luego U

Simulamos N pares (X, Y) con distribuciones exponenciales de tasa 1, luego calculamos $U = \exp(-(X + Y))$.

```
N <- 10^5 # tamaño de la simulación
```

```
# X y Y ~ Exp(1)
```

```
X <- rexp(N, rate = 1)
```

```
Y <- rexp(N, rate = 1)
```

```
# Definimos U = exp(-(X+Y))
```

```
U <- exp(-(X + Y))
```

```
summary(U)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## 6.200e-07 6.787e-02 1.874e-01 2.510e-01 3.827e-01 9.962e-01
```

```
range(U) # debería estar dentro de (0,1)
```

```
## [1] 6.158473e-07 9.962246e-01
```

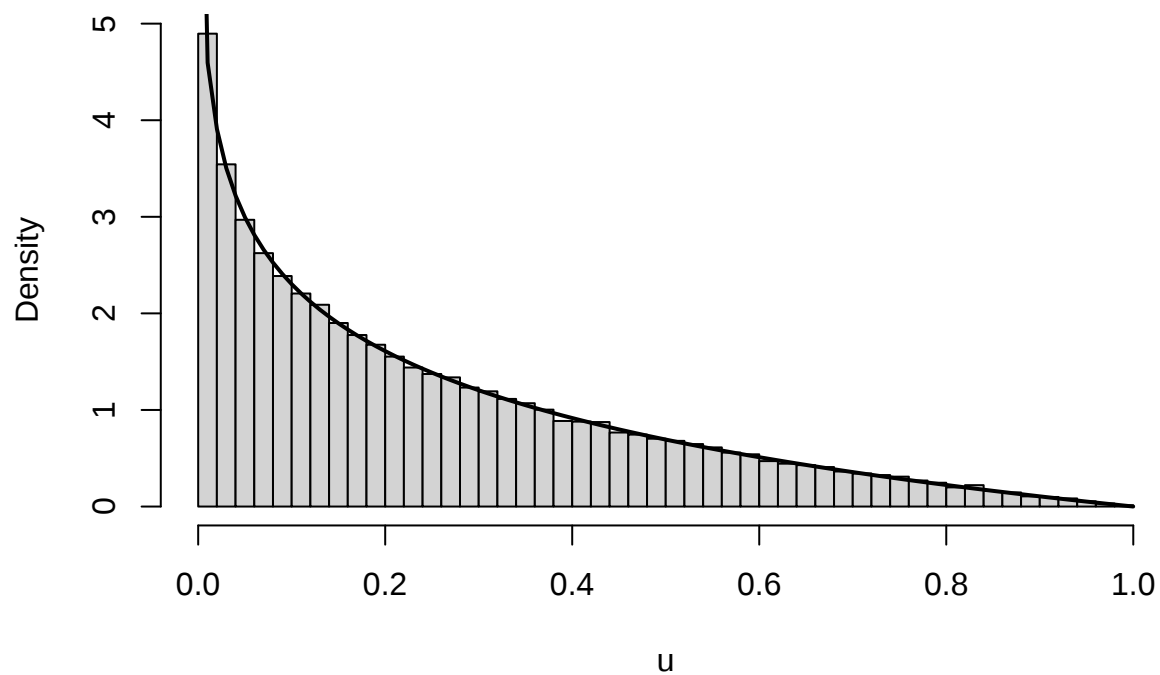
Paso 2: histograma empírico de U

Graficamos el histograma de U normalizado como densidad (`probability=TRUE`) y encima la densidad teórica $f_U(u) = -\log(u)$ en $(0,1)$.

```
hist(U,
     probability = TRUE,
     breaks = 60,
     main = "Distribución de U = exp(-(X+Y))",
     xlab = "u")

curve(-log(x),
      from = 0.0001, # evitar log(0)
      to = 0.9999,
      add = TRUE,
      lwd = 2)
```

Distribución de $U = \exp(-(X+Y))$



Paso 3: Esperanza empírica vs teórica

Calculamos el promedio muestral de U y lo comparamos con $1/4$.

```
EU_muestral <- mean(U)
EU_teorico  <- 1/4
```

```
EU_muestral
```

```
## [1] 0.2509841
```

```
EU_teorico
```

```
## [1] 0.25
```

lab12Computo — Simulación de Rayleigh

Carla Reyes

2025-11-10

Objetivo

Partiendo de dos normales independientes con media cero y varianza diferentes de cero, definimos la distancia al origen $R = \sqrt{X_1^2 + X_2^2}$. Se sabe que R sigue una **distribución de Rayleigh** con parámetro σ .

Parámetros y utilidades

```
n      <- 200000    # número de simulaciones
sigma  <- 2.0       # parámetro de escala de las normales y de Rayleigh

# densidad de Rayleigh
dRayleigh <- function(r, sigma) (r/sigma^2) * exp(-(r^2)/(2*sigma^2))

ER_theo  <- sigma * sqrt(pi/2)
Var_theo <- (2 - pi/2) * sigma^2
c(ER_theo = ER_theo, Var_theo = Var_theo)

##  ER_theo Var_theo
## 2.506628 1.716815
```

Simulación desde normales y transformación radial

```
Z1 <- rnorm(n)
Z2 <- rnorm(n)
X1 <- sigma * Z1
X2 <- sigma * Z2
R  <- sqrt(X1^2 + X2^2)

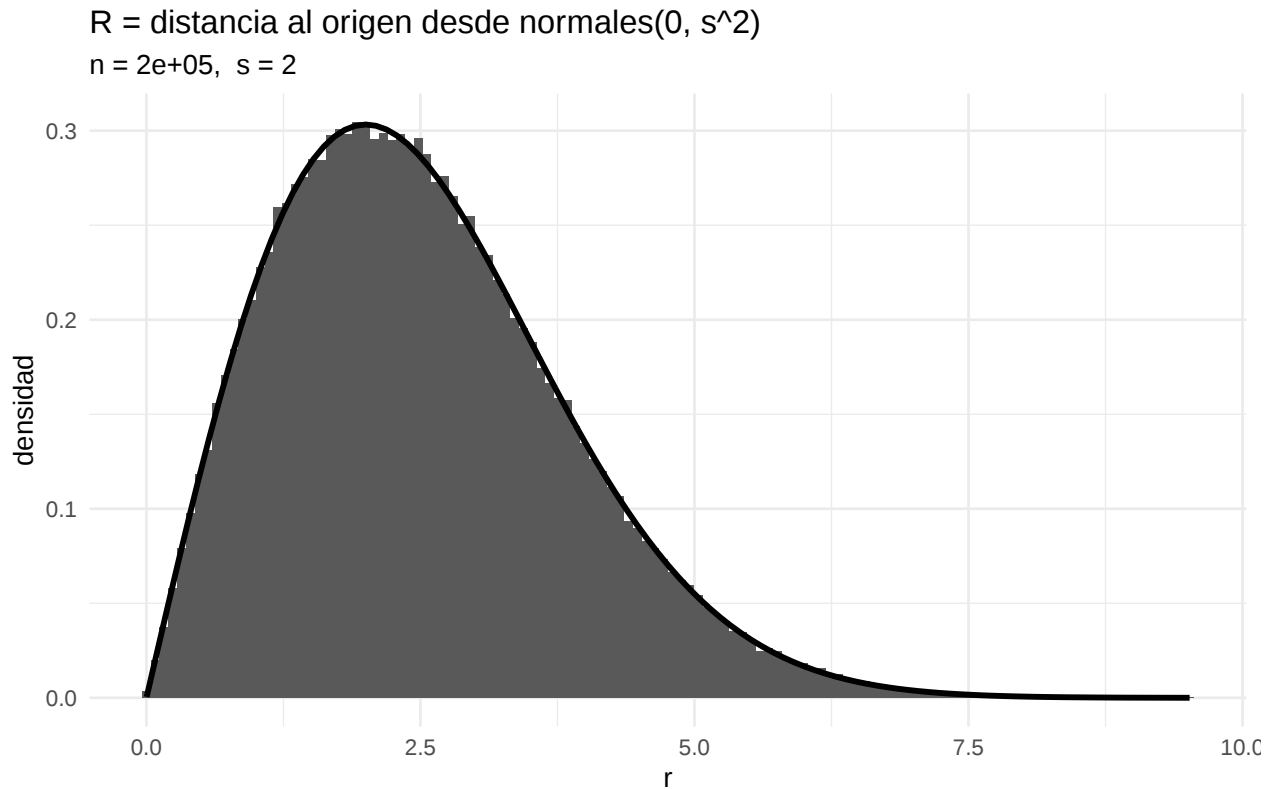
ER_hat <- mean(R)
VR_hat <- var(R)
c(ER_hat = ER_hat, VR_hat = VR_hat)

##  ER_hat  VR_hat
## 2.507424 1.722702
```

Histograma vs. densidad teórica

```
library(ggplot2)
```

```
df <- data.frame(R = R)
g <- ggplot(df, aes(x = R)) +
  geom_histogram(aes(y = ..density..), bins = 120, boundary = 0, closed = "left") +
  stat_function(fun = function(x) dRayleigh(x, sigma), linewidth = 1.1) +
  labs(title = "R = distancia al origen desde normales(0, s^2)",
        subtitle = paste0("n = ", format(n, big.mark=","), ", s = ", sigma),
        x = "r", y = "densidad") +
  theme_minimal()
g
```



Verificación de momentos

```
data.frame(
  cantidad = c("E[R]", "Var(R)"),
  teorico = c(ER_theo, Var_theo),
  empirico = c(ER_hat, VR_hat)
)
```

```
## cantidad teorico empirico
## 1 E[R] 2.506628 2.507424
## 2 Var(R) 1.716815 1.722702
```

Lab 13 Cómputo - Estadísticos de orden

Carla Reyes

19 de noviembre de 2025

Ejercicio 1: comparación de dos estimadores de θ

Consideramos una muestra aleatoria $X_1, \dots, X_n$ de una distribución $X_i \sim \text{Unif}(0, \theta)$. Estudiaremos por simulación los siguientes estimadores:

$$\tilde{\theta} = 2\bar{X}, \quad \hat{\theta} = \frac{n+1}{n}X_{(n)}.$$

Parámetros de simulación

```
set.seed(12345)
theta_verdadera <- 10
n_vec <- c(5, 10, 30)
B <- 10000
```

```
theta_verdadera
```

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

```
n_vec
```

```
## [1] 5 10 30
```

```
B
```

```
## [1] 10000
```

Función de simulación

```
sim_estimadores <- function(n, theta, B = 10000) {
  muestras <- matrix(runif(B * n, 0, theta), nrow = B)

  medias <- rowMeans(muestras)
  maximos <- apply(muestras, 1, max)

  data.frame(
    n = n,
    theta_tilde = 2 * medias,
    theta_hat = (n + 1)/n * maximos
  )
}
```

Correr simulaciones

```
resultados_lista <- lapply(n_vec, sim_estimadores,  
                           theta = theta_verdadera, B = B)  
  
resultados <- do.call(rbind, resultados_lista)  
  
head(resultados)  
  
##   n theta_tilde theta_hat  
## 1 5   10.446557  8.650847  
## 2 5   16.629186 11.968946  
## 3 5   14.389521 10.435692  
## 4 5   14.444601 11.774803  
## 5 5    7.545054  7.477126  
## 6 5    8.237339 11.357798
```

Resumen

```
library(dplyr)  
  
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union  
  
theta0 <- theta_verdadera  
  
resumen <- resultados |>  
  tidyr::pivot_longer(  
    cols = c(theta_tilde, theta_hat),  
    names_to = "estimador",  
    values_to = "valor"  
  ) |>  
  group_by(n, estimador) |>  
  summarise(  
    media_sim = mean(valor),  
    sesgo_sim = mean(valor) - theta0,  
    var_sim   = var(valor),  
    mse_sim   = mean((valor - theta0)^2),  
    .groups = "drop"  
  )  
  
resumen
```

```
## # A tibble: 6 × 6
##       n estimador   media_sim sesgo_sim var_sim mse_sim
##   <dbl> <chr>       <dbl>     <dbl>   <dbl>   <dbl>
## 1     5 theta_hat     9.99  -0.0136   2.89   2.89
## 2     5 theta_tilde    9.98  -0.0238   6.63   6.63
## 3    10 theta_hat     9.98  -0.0154   0.858  0.858
## 4    10 theta_tilde   10.0    0.0123   3.37   3.37
## 5    30 theta_hat     10.0    0.00331  0.103  0.103
## 6    30 theta_tilde   10.0    0.0105   1.09   1.09
```

Gráficas de las distribuciones simuladas

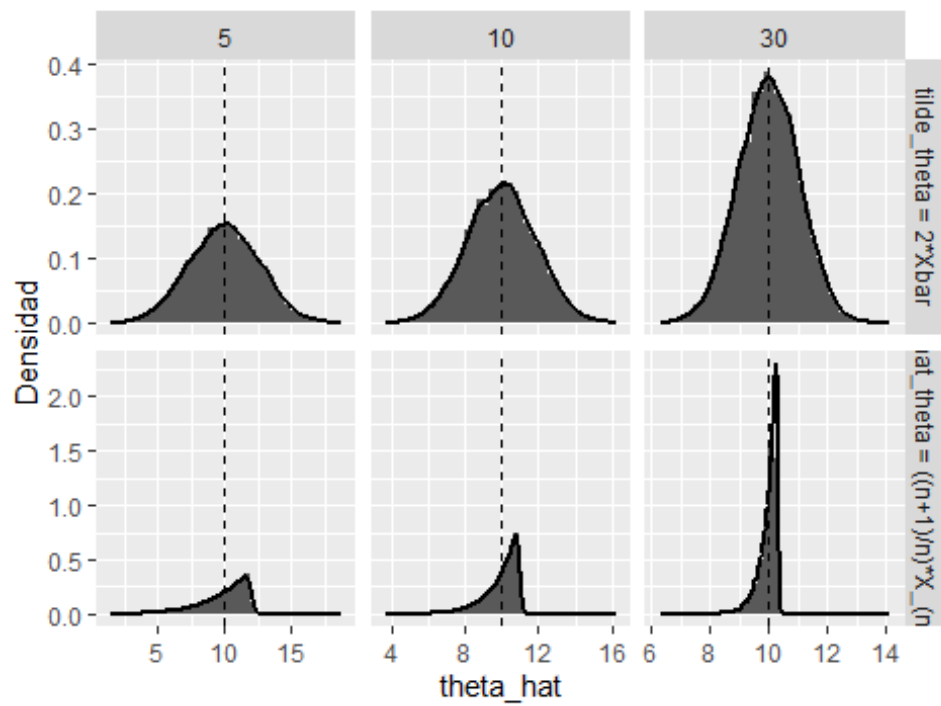
```
library(ggplot2)

datos_largos <- resultados |>
  tidyr::pivot_longer(
    cols = c(theta_tilde, theta_hat),
    names_to = "estimador",
    values_to = "valor"
  )

datos_largos$estimador <- factor(
  datos_largos$estimador,
  levels = c("theta_tilde", "theta_hat"),
  labels = c("tilde_theta = 2*Xbar", "hat_theta = ((n+1)/n)*X_(n)")
)

ggplot(datos_largos, aes(x = valor)) +
  geom_histogram(aes(y = after_stat(density)), bins = 40) +
  geom_density(linewidth = 1) +
  geom_vline(xintercept = theta_verdadera, linetype = "dashed") +
  facet_grid(estimador ~ n, scales = "free") +
  labs(
    title = "Distribuciones simuladas de los estimadores de  $\theta$ ",
    x = expression(theta_hat),
    y = "Densidad"
  )
```

Distribuciones simuladas de los estimadores de θ



Lab14Computo - Simulación del TCL y la LGN

Carla Reyes

Teorema Central del Límite (TCL)

En este apartado simularemos el TCL con variables aleatorias Uniforme(0,1).

```
set.seed(123)

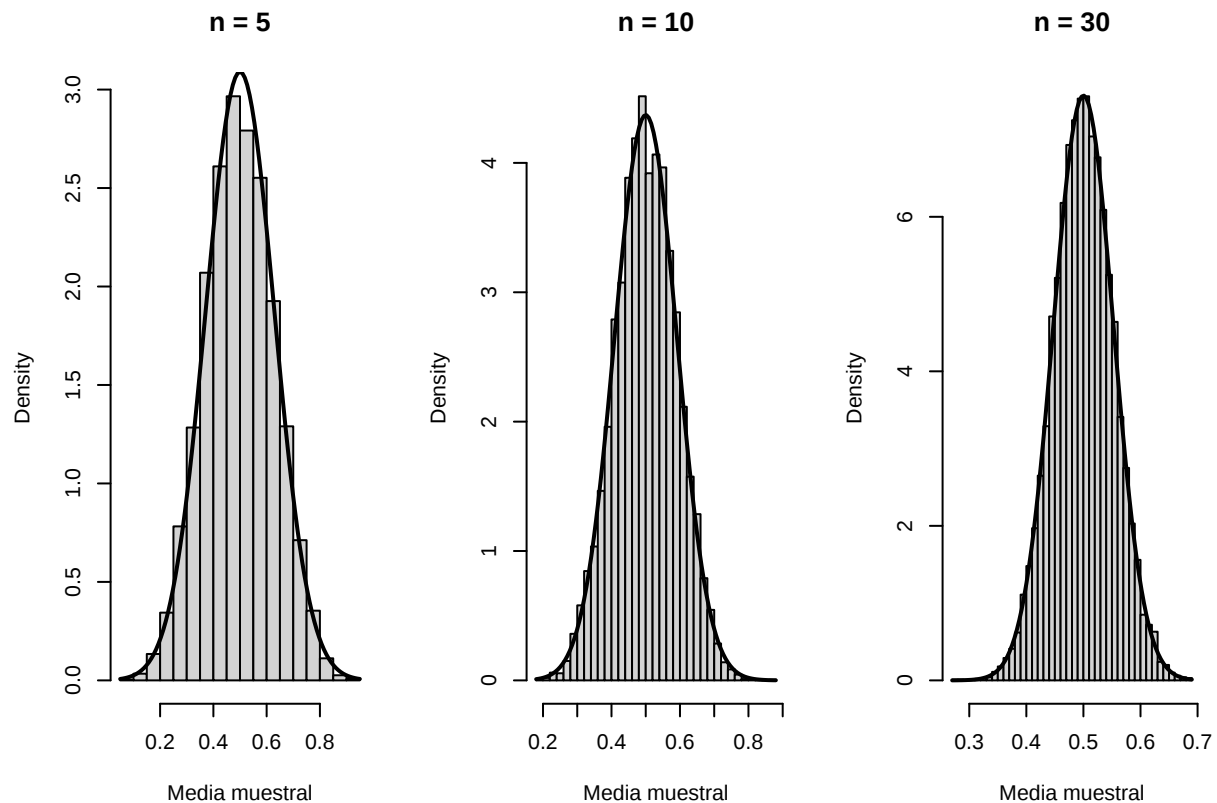
B <- 10000          # número de simulaciones
n_vec <- c(5, 10, 30) # tamaños de muestra para ilustrar el TCL

par(mfrow=c(1,3))

for (n in n_vec) {
  medias <- replicate(B, mean(runif(n, 0, 1)))

  hist(medias, freq = FALSE, main = paste("n =", n),
       xlab = "Media muestral", breaks = 30)

  curve(dnorm(x, mean = 0.5, sd = 1/sqrt(12*n)),
        add = TRUE, lwd = 2)
}
```



Observamos que conforme n aumenta, la distribución de la media muestral se acerca más a una normal.

Ley de los Grandes Números (LGN)

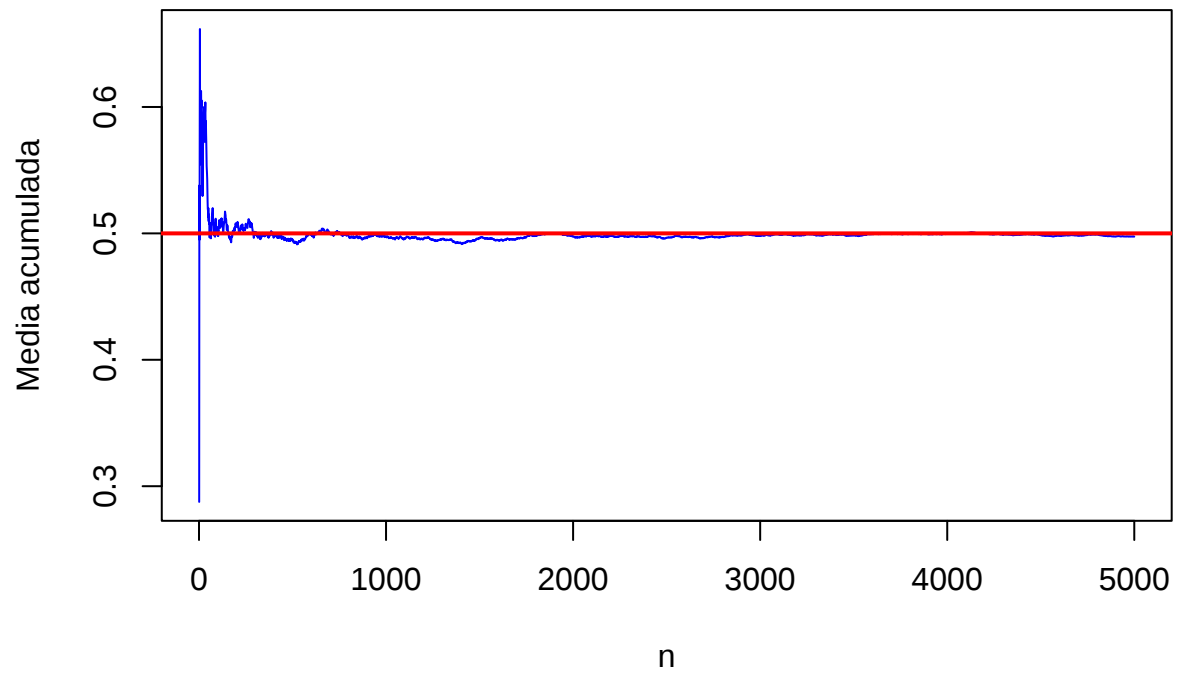
Simulamos cómo la media muestral converge al valor esperado de una $\text{Uniforme}(0,1)$, que es 0.5.

```
set.seed(123)

N <- 5000
x <- runif(N)
media_acumulada <- cumsum(x) / (1:N)

plot(media_acumulada, type="l", col="blue",
      xlab="n", ylab="Media acumulada",
      main="Ley de los Grandes Números")
abline(h = 0.5, col="red", lwd=2)
```

Ley de los Grandes Números



La media acumulada converge hacia 0.5, lo que demuestra la LGN.