

UNIVERSIDAD AUTÓNOMA DE GUERRERO  
FACULTAD DE CIENCIAS Y TECNOLOGÍAS DE LA  
INFORMACIÓN

CURSO PROPEDEÚTICO 3ERA GENERACIÓN MAESTRÍA EN CIENCIA DE  
DATOS Y SISTEMAS INTELIGENTES

FUNDAMENTOS DE PROBABILIDAD

Nubia Montserrat Pineda Galeana

2025-11-19

## Contents

<b>Librerías</b>	<b>1</b>
<b>CONJUNTOS</b>	<b>2</b>
Definir Conjuntos . . . . .	2
Operaciones con Conjuntos . . . . .	2
<b>ESPACIO MUESTRAL Y ESPACIO DE EVENTOS</b>	<b>5</b>
Lanzamiento de 2 monedas . . . . .	5
Lanzamiento de un dado . . . . .	7
<b>CÁLCULO DE PROBABILIDADES</b>	<b>8</b>
Suma de Probabilidades . . . . .	8
Probabilidad Condicional . . . . .	10

## Librerías

```
library(tinytex)      # Compilación de documentos
library(tidyverse)    # Manipulación y análisis de datos
library(dplyr)        # Manipulación de datos
library(fBasics)      # Para variables aleatorias
library(prob)         # Cálculo de probabilidades
```

# CONJUNTOS

## Definir Conjuntos

```
# Los conjuntos A, B, Y C se definirán mediante vectores
```

```
A <- c(1, 2, 3, 4, 5) # Conjunto A  
B <- c(4, 5, 6, 7, 8) # Conjunto B  
C <- c(3, 5, 8, 9, 10) # Conjunto C
```

```
cat("Conjuntos \n")
```

Conjuntos

```
cat("A = {", A, "}", "\n") # Muestra el Conjunto A
```

A = { 1 2 3 4 5 }

```
cat("B = {", B, "}", "\n") # Muestra el Conjunto B
```

B = { 4 5 6 7 8 }

```
cat("C = {", C, "}", "\n") # Muestra el Conjunto C
```

C = { 3 5 8 9 10 }

## Operaciones con Conjuntos

Funciones union - Unión

intersect - Intersección

setdiff - Diferencia

### Unión de Conjuntos

```
# A unión B
```

```
A_union_B <- union(A, B)
```

```
cat("A u B = {", A_union_B, "}", "\n\n") # cat: imprime una leyenda. #\n: genera salto de línea
```

A u B = { 1 2 3 4 5 6 7 8 }

```
# A unión C
```

```
A_union_C <- union(A, C)
```

```
cat("A u C = {", A_union_C, "}", "\n\n")
```

A u C = { 1 2 3 4 5 8 9 10 }

```
# B unión C
B_union_C <- union(B, C)
cat("B u C = {", B_union_C, "}", "\n\n")
```

B u C = { 4 5 6 7 8 3 9 10 }

```
# Unión de los 3 conjuntos A B C
A_union_B_union_C <- union(union(A, B), C)
cat("A u B u C = {", A_union_B_union_C, "}", "\n")
```

A u B u C = { 1 2 3 4 5 6 7 8 9 10 }

## Intersección de Conjuntos

```
# A intersección B
A_inter_B <- intersect(A,B)
cat("A n B = {", A_inter_B , "}", "\n\n")
```

A n B = { 4 5 }

```
# A intersección C
A_inter_C <- intersect(A,C)
cat("A n C = {", A_inter_C , "}", "\n\n")
```

A n C = { 3 5 }

```
# B intersección C
B_inter_C <- intersect(B,C)
cat("B n C = {", B_inter_C , "}", "\n\n")
```

B n C = { 5 8 }

```
# Intersección de los 3 conjuntos A B C
A_inter_B_inter_C <- intersect(intersect(A, B), C)
cat("A n B n C = {", A_inter_B_inter_C, "}", "\n")
```

A n B n C = { 5 }

## Diferencia de Conjuntos

```
# A - B
A_menos_B <- setdiff(A,B)
cat("A - B = {", A_menos_B , "}", "\n\n")
```

A - B = { 1 2 3 }

```
# B - C
B_menos_C <- setdiff(B,C)
cat("B - C = {", B_menos_C , "}", "\n\n")
```

B - C = { 4 6 7 }

```
# C - A
C_menos_A <- setdiff(C,A)
cat("C - A = {", C_menos_A , "}", "\n\n")
```

C - A = { 8 9 10 }

## Complemento de Conjuntos

```
# Paso 1. Definir el Conjunto Universo
# Son todos los elementos de los 3 conjuntos, sin duplicados
U <- sort(unique(c(A, B, C))) # sort: ordena el resultado
cat("Universo = {", U , "}", "\n\n")
```

Universo = { 1 2 3 4 5 6 7 8 9 10 }

```
# Paso 2. Visualizar Conjuntos A, B y C
cat("A = {", A, "}", "\n") # Muestra el Conjunto A
```

A = { 1 2 3 4 5 }

```
cat("B = {", B, "}", "\n") # Muestra el Conjunto B
```

B = { 4 5 6 7 8 }

```
cat("C = {", C, "}", "\n\n") # Muestra el Conjunto C
```

C = { 3 5 8 9 10 }

```
# Paso 3. Calcular los Conjuntos Complementos
# Complemento de A
# Todos los elementos de Universo que no están en A
A_compl <- setdiff(U,A)
cat("A Complemento = {", A_compl , "}", "\n\n")
```

A Complemento = { 6 7 8 9 10 }

```
# Complemento de B
B_compl <- setdiff(U,B)
cat("B Complemento = {", B_compl , "}", "\n\n")
```

B Complemento = { 1 2 3 9 10 }

```
# Complemento de C
C_compl <- setdiff(U,C)
cat("C Complemento = {", C_compl , "}", "\n\n")
```

C Complemento = { 1 2 4 6 7 }

## Cardinalidad de un Conjunto

La cardinalidad de un conjunto es el número de elementos que lo componen. Para determinar la cardinalidad, se utiliza la función "length"

```
card_A <- length(A)
cat("Cardinalidad A = {", card_A , "}", "\n\n")
```

Cardinalidad A = { 5 }

```
card_B <- length(B)
cat("Cardinalidad B = {", card_B , "}", "\n\n")
```

Cardinalidad B = { 5 }

```
card_C <- length(C)
cat("Cardinalidad C = {", card_C , "}", "\n\n")
```

Cardinalidad C = { 5 }

```
card_U <- length(U)
cat("Cardinalidad U = {", card_U , "}", "\n\n")
```

Cardinalidad U = { 10 }

# ESPACIO MUESTRAL Y ESPACIO DE EVENTOS

## Lanzamiento de 2 monedas

### Espacio Muestral

Determinar el Espacio Muestral  $\Omega$  del experimento al lanzar 2 monedas

Función: tosscoin.

Establece un espacio muestral para el experimento de lanzar una moneda repetidamente con los resultados "H" (cara) o "T" (cruz).

```
Omega <- tosscoin(2)
Omega
```

	toss1	toss2
1	H	H
2	T	H
3	H	T
4	T	T

```
Card_Omega <- nrow(Omega)
cat("Cardinalidad Omega = ", Card_Omega , "\n\n")
```

Cardinalidad Omega = 4

```
cat("El espacio muestral contiene", Card_Omega , "resultados posibles al lanzar 2 monedas")
```

El espacio muestral contiene 4 resultados posibles al lanzar 2 monedas

## Espacio de Eventos

Un evento es cualquier subconjunto del espacio muestral  $\Omega$

Determinar el total de eventos de  $\Omega$

Se calcula la potencia del conjunto

$2^{(\text{Num elementos de } \Omega)}$

```
Eventos_Tot <- 2^4
Eventos_Tot
```

[1] 16

```
cat("El espacio de eventos contiene", Eventos_Tot , "subconjuntos posibles del experimento de lanzar 2 monedas")
```

El espacio de eventos contiene 16 subconjuntos posibles del experimento de lanzar 2 monedas

Evento vacío

1.  $\{ \}$

Eventos con un solo resultado

2.  $\{HH\}$

3.  $\{HT\}$

4.  $\{TH\}$

5.  $\{TT\}$

Eventos con dos resultados

6.  $\{HH, HT\}$

7.  $\{HH, TH\}$

8. {HH, TT}
9. {HT, TH}
10. {HT, TT}
11. {TH, TT}

Eventos con tres resultados

12. {HH, HT, TH}
13. {HH, HT, TT}
14. {HH, TH, TT}
15. {HT, TH, TT}

Evento con cuatro resultados (el total)

16. {HH, HT, TH, TT} =  $\Omega$

## Lanzamiento de un dado

### Espacio Muestral

```
Omega.Dado <- 1:6
Card_Omega.D <- length(Omega.Dado)
cat("El espacio muestral  $\Omega = \{$ ", Omega.Dado, " $\}$ ", "contiene", Card_Omega.D, "resultados posibles al lanzar 1 dado")
```

El espacio muestral  $\Omega = \{ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \}$  contiene 6 resultados posibles al lanzar 1 dado

### Espacio de Eventos

```
Eventos_Tot_dado <- 2^6
Eventos_Tot_dado

[1] 64

cat("El espacio de eventos contiene", Eventos_Tot_dado, "subconjuntos posibles del experimento de lanzar 1 dado")
```

El espacio de eventos contiene 64 subconjuntos posibles del experimento de lanzar 1 dado

### Probabilidades de Eventos

```
# Probabilidad de A: número par
probA <- length(c(2,4,6)) / length(Omega.Dado)
cat("Probabilidad de A: número par = ", probA , "\n\n")
```

Probabilidad de A: número par = 0.5

```
# Probabilidad de B: número > 4
probB <- length(c(5,6)) / length(Omega.Dado)
cat("Probabilidad de B: número > 4 = ", probB , "\n\n")
```

Probabilidad de B: número > 4 = 0.3333333

```
# Probabilidad de C: número 1 o 6
probC <- length(c(1,6)) / length(Omega.Dado)
cat("Probabilidad de C: número 1 o 6 = ", probC , "\n\n")
```

Probabilidad de C: número 1 o 6 = 0.3333333

## CÁLCULO DE PROBABILIDADES

### Suma de Probabilidades

Experimento: Lanzar un dado justo de 6 caras

Conocer la probabilidad de obtener un número par o un número mayor que 4

Definir los eventos:

Evento A = Obtener un número par = {2,4,6}

Evento B = Obtener un número mayor que 4 = {5,6}

Probabilidad que se desea encontrar:

$A \cup B$

```
# Crear un data frame con columna R para los resultados del lanzamiento de 1 dado
Omega.D <- data.frame(R = 1:6)
Omega.D
```

```
  R
1 1
2 2
3 3
4 4
5 5
6 6
```

```
# Evento A: Números pares
eA <- subset(Omega.D, R %% 2 == 0)
eA
```



```
R
2 2
4 4
6 6
```

```
# Evento B: Números mayores que 4
eB <- subset(Omega.D, R > 4)
eB
```

```
R
5 5
6 6
```

## Forma 1 de resolución

Probabilidad  $A \cup B$

```
# A unión B
eA_union_eB <- union(eA, eB)
cat("A u B = {", eA_union_eB, "}", "\n\n")
```

```
A u B = { 2 4 5 6 }
```

```
Prob_AuB <- length(eA_union_eB)/nrow(Omega.D)
Prob_AuB
```

```
[1] 0.6666667
```

## Forma 2 de resolución

Sabemos que

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

```
# P(A) - Probabilidad del Evento A
Prob_A <- nrow(eA)/nrow(Omega.D)
cat("P(A) =", Prob_A, "\n\n")
```

```
P(A) = 0.5
```

```
# P(B) - Probabilidad del Evento B
Prob_B <- nrow(eB)/nrow(Omega.D)
cat("P(B) =", Prob_B, "\n\n")
```

```
P(B) = 0.3333333
```

```
# P(A n B) - Probabilidad de A intersección B
# (A n B) = {6}, solo consta de 1 resultado, entonces:
Prob_A_inter_B = 1/nrow(Omega.D)
cat("P(A n B) =", Prob_A_inter_B, "\n\n")
```

$$P(A \cap B) = 0.1666667$$

```
# Calculamos la Probabilidad de la unión de eventos A con B, P(A u B)
Prob_A_union_B <- Prob_A + Prob_B - Prob_A_inter_B
cat("P(A u B) =", Prob_A_union_B, "es la probabilidad de obtener un número par o un número mayor que 4")
```

$P(A \cup B) = 0.6666667$  es la probabilidad de obtener un número par o un número mayor que 4

## Probabilidad Condicional

Situación:

En una clase de 30 estudiantes:

18 son mujeres, de las cuales 6 usan lentes.

12 son hombres, de los cuales 4 usan lentes.

Definimos los eventos:

L = “Estudiante que usa lentes”

M = “Estudiante que es mujer”

Deseamos conocer:

¿Cuál es la probabilidad de que un estudiante use lentes, dado que sabemos que es mujer?

$P(L | M)$

Sabemos que:

$$P(L | M) = \frac{P(L \cap M)}{P(M)}$$

donde:  $(L \cap M)$  = Estudiante mujer que usa lentes

Cálculo de Probabilidades

```
# Definir valores
Universo <- 30 # Total de estudiantes

Mujeres <- 18

Mujer_Lentes <- 6

# Probabilidad de ser estudiante mujer
Prob_M <- Mujeres/Universo
cat("La probabilidad de ser estudiante mujer, P(M) =", Prob_M, "\n\n")
```

La probabilidad de ser estudiante mujer,  $P(M) = 0.6$

```
# Probabilidad de ser mujer que usa lentes
Prob_M.Lentes <- Mujer_Lentes/Universo
cat("La probabilidad de ser estudiante mujer que usa lentes, P(L n M) =", Prob_M.Lentes, "\n\n")
```

La probabilidad de ser estudiante mujer que usa lentes,  $P(L \cap M) = 0.2$

```
# Probabilidad Condicional: Ser estudiante que usa lentes dado que sea mujer
Prob.Cond_LM <- round((Prob_M.Lentes / Prob_M),2)
cat("La probabilidad de ser estudiante que usa lentes dado que es mujer,  $P(L|M)$  =", Prob.Cond_LM)
```

La probabilidad de ser estudiante que usa lentes dado que es mujer,  $P(L|M) = 0.33$