Workshop-2.R

jsegu

2025-09-06

```
# MARIA PAULA BONILLA MARTINEZ, JUAN SEBASTIÁN GUAYAZÁN CLAVIJO, JERONIMO ESTEBAN QUILAGUY TORRES
# Aprendizaje estadístico 2 (MATE APE2-1)
# Coordinación Ingeniería Estadística
# Ingeniería Estadística
# Escuela Colombiana de Ingeniería Julio Garavito
# 2025-1

# EJERCICIO 1 (Frutas)

# a) Distribución conjunta

X <- c(0,0,0,1,1,1,2,2,2,3,3,3)
Y <- c(0,1,2,0,1,2,0,1,2,0,1,2)

dist_conjunta <- c(3/70, 6/70, 3/70, 9/70, 18/70, 9/70, 9/70, 18/70, 9/70, 3/70, 6/70, 3/70)
data.frame(X, Y, dist_conjunta)
```

```
##    X Y dist_conjunta
## 1  0 0    0.04285714
## 2  0 1    0.08571429
## 3  0 2    0.04285714
## 4  1 0    0.12857143
## 5  1 1    0.25714286
## 6  1 2    0.12857143
## 7  2 0    0.12857143
## 8  2 1    0.25714286
## 9  2 2    0.12857143
## 10 3 0    0.04285714
## 11 3 1    0.08571429
## 12 3 2    0.04285714
```

```
# b) P((X,Y) E A) donde A = {(x,y): x + y <= 2}
prob_a <- sum(dist_conjunta[X + Y <= 2])
prob_a
```

```
## [1] 0.6857143
```

```
# c) Verificar independencia
independientes <- (3/70) == (12/70)*(24/70)
independientes
```

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

```r
#NO SON INDEPENDIENTES, porque P(0,0) != P(X=0) × P(Y=0)



# EJERCICIO 2 (Fast-food Restaurant)

# a) f_X(x) = (2/3)(x + 1)
x<- 0.5 #Ejemplo para comprobar (puede tomar cualquier valor entre 0<=x<=1)
f_x <- (2/3)*(x + 1)
f_x
```

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

```r
# b) f_Y(y) = (1 + 4y)/3
y <- 0.5 #Ejemplo para comprobar (puede tomar cualquier valor entre 0<=y<=1)
f_y <- (1 + 4*y)/3
f_y
```

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

```r
# c) P(X < 1/2)
p_c <- (2/3) * ((0.5^2)/2 + 0.5)
p_c
```

```
## [1] 0.4166667
```

```r
# d) P(Y > 1/2)
p_d <- (1 + 2*1^2)/3 - (0.5 + 2*(0.5^2))/3
p_d
```

```
## [1] 0.6666667
```

```r
# e) Verificar independencia en un punto
x = 0
y = 0
fxy <- (2/3)*(x + 2*y)
marginal <- ((2/3)*(x+1)) * ((1+4*y)/3)
fxy == marginal
```

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

```r
#NO SON INDEPENDIENTES



# EJERCICIO 3
# a) Evaluar K
ix2 <- (50^3 - 30^3)/3
int_total <- (40/3) * (50^3 - 30^3)
k <- 1 / int_total
k
```

```
## [1] 7.653061e-07
```

```r
# Solución integrales
x2_30_50 <- (50^3 - 30^3)/3
x2_30_40 <- (40^3 - 30^3)/3
y2_40_50 <- (50^3 - 40^3)/3

# b) P(30<=X<=40 , 40<=Y<=50)
t1 <- x2_30_40 * 10
t2 <- y2_40_50 * 10
pb <- k * (t1 + t2)
pb
```

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

```r
# c) P(X<40 , Y<40)
ta <- x2_30_40 * 10
tb <- x2_30_40 * 10
pc <- k * (ta + tb)
pc
```

```
## [1] 0.1887755
```

```r
# d) Independencia
fxy <- k * (35^2 + 45^2)

fx <- k * (20*35^2 + (50^3 - 30^3)/3)
fy <- k * (20*45^2 + (50^3 - 30^3)/3)

ind <- fx * fy

fxy == ind
```

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

```r
# NO SON INDEPENDIENTES



# EJERCICIO 4
p <- c(0.05,0.05,0.10,
       0.05,0.10,0.35,
       0.00,0.20,0.10)

# a) Marginal de X
px1 <- sum(p * c(1,0,0, 1,0,0, 1,0,0))
px2 <- sum(p * c(0,1,0, 0,1,0, 0,1,0))
px3 <- sum(p * c(0,0,1, 0,0,1, 0,0,1))
px  <- c(px1, px2, px3)
px
```

```
## [1] 0.10 0.35 0.55
```

```r
# b) Marginal de Y
py1 <- sum(p * c(1,1,1, 0,0,0, 0,0,0))
py2 <- sum(p * c(0,0,0, 1,1,1, 0,0,0))
py3 <- sum(p * c(0,0,0, 0,0,0, 1,1,1))
py  <- c(py1, py2, py3)
py
```

```
## [1] 0.2 0.5 0.3
```

```r
# c) Probabilidad condicional
num <- sum(p * c(0,0,0, 0,0,0, 0,1,0))
pbb_c  <- num / px2
pbb_c
```

```
## [1] 0.5714286
```

```r
# # EJERCICIO 5

# a) distribución conjunta
p <- c(0.36,0.24,0, 0,0.24,0.16)

# b) marginal de W
p_w0 <- sum(p * c(1,0,0, 0,0,0))
p_w1 <- sum(p * c(0,1,0, 0,1,0))
p_w2 <- sum(p * c(0,0,0, 0,0,1))
p_w  <- c(p_w0,p_w1,p_w2)
p_w
```

```
## [1] 0.36 0.48 0.16
```

```r
# c) marginal de Z
p_z0 <- sum(p * c(1,1,1, 0,0,0))
p_z1 <- sum(p * c(0,0,0, 1,1,1))
p_z  <- c(p_z0,p_z1)
p_z
```

```
## [1] 0.6 0.4
```

```r
# d) probabilidad al menos una cara
pbb_1 <- 1 - p_w0
pbb_1
```

```
## [1] 0.64
```

```r
# e) prueba de independencia (ejemplo con W=1,Z=0)
ind <- (0.24 == p_w1 * p_z0)
ind
```

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

```
# NO SON INDEPENDIENTES

# # Ejercicio 6
# f_xy <- function(x, y) {
#   ifelse(x > 0 & x < 2 & y > 2 & y < 4, (6 - x - y)/8, 0)
# }
#
# # a
# f_x <- function(x) {
#   # Integrar respecto a y de 2 a 4
#   integrate(function(y) f_xy(x, y), lower=2, upper=4)$value
# }
#
# # Función para calcular la densidad marginal de Y
# f_y <- function(y) {
#   # Integrar respecto a x de 0 a 2
#   integrate(function(x) f_xy(x, y), lower=0, upper=2)$value
# }
#
# # Calcular f_X(1)
# fx_1 <- f_x(1)
#
# # Definir la densidad condicional de Y dado X=1
# f_y_given_x1 <- function(y) {
#   f_xy(1, y) / fx_1
# }
#
# # Calcular P(1 < Y < 3 | X=1)
# p <- integrate(f_y_given_x1, lower=1, upper=3)$value
#
# cat("a) P(1 < Y < 3 | X=1) =", p, "\n")
#
# # Verificar independencia
# x_vals <- seq(0.1, 1.9, length.out=5)
# y_vals <- seq(2.1, 3.9, length.out=5)
#
# independent <- TRUE
#
# for (x in x_vals) {
#   for (y in y_vals) {
#     f_xy_val <- f_xy(x, y)
#     prod_marginals <- f_x(x) * f_y(y)
#     if (abs(f_xy_val - prod_marginals) > 1e-4) {
#       independent <- FALSE
#       break
#     }
#   }
#   if (!independent) break
# }
#
# if (independent) {
#   cat("X y Y son independientes.\n")
# } else {
```

```
#   cat("X y Y NO son independientes.\n")
# }
# #Ejercicio 7
#
# # Definir la función de densidad conjunta
# f <- function(x, y, z) {
#   4 * x * y * z^2 / 9
# }
#
# # a) Función marginal conjunta de Y y Z (integrar sobre x en (0,1))
# f_YZ <- function(y, z) {
#   integrate(function(x) f(x, y, z), lower=0, upper=1)$value
# }
#
# # b) Función marginal de Y (integrar f_YZ sobre z en (0,3))
# f_Y <- function(y) {
#   integrate(function(z) f_YZ(y, z), lower=0, upper=3)$value
# }
#
# # c) Probabilidad P(1/4 < X < 1/2, Y > 1/3, 1 < Z < 2)
# p_c <- integrate(function(y) {
#   sapply(y, function(yy) {
#     integrate(function(z) {
#       integrate(function(x) f(x, yy, z), lower=1/4, upper=1/2)$value
#     }, lower=1, upper=2)$value
#   })
# }, lower=1/3, upper=1)$value
#
# # d) Probabilidad condicional P(0 < X < 1/2 | Y=1/3, Z=2)
# numerador_d <- integrate(function(x) f(x, 1/3, 2), lower=0, upper=1/2)$value
# denominador_d <- f_YZ(1/3, 2)
# p_d <- numerador_d / denominador_d
#
# # Imprimir resultados
# cat("a) f_YZ(y,z) para y=0.5,z=1.5: ", f_YZ(0.5, 1.5), "\n")
# cat("b) f_Y(y) para y=0.5: ", f_Y(0.5), "\n")
# cat("c) P(1/4 < X < 1/2, Y > 1/3, 1 < Z < 2): ", p_c, "\n")
# cat("d) P(0 < X < 1/2 | Y=1/3, Z=2): ", p_d, "\n")
#
# # Ejercicio 9A
# # Definir la función de densidad conjunta
# f_xy <- function(x, y) {
#   ifelse(x >= 0 & x <= 1 & y >= 0 & y <= 1, x + y, 0)
# }
#
# # Distribución marginal de X: f_X(x) =  f(x,y) dy desde 0 a 1
# f_x <- function(x) {
#   sapply(x, function(xi) {
#     integrate(function(y) f_xy(xi, y), lower = 0, upper = 1)$value
#   })
# }
#
# # Distribución marginal de Y: f_Y(y) =  f(x,y) dx desde 0 a 1
```

```
# f_y <- function(y) {
#   sapply(y, function(yi) {
#     integrate(function(x) f_xy(x, yi), lower = 0, upper = 1)$value
#   })
# }
#
# # Calcular P(X > 0.5, Y > 0.5)
# prob <- integrate(function(y) {
#   sapply(y, function(yi) {
#     integrate(function(x) f_xy(x, yi), lower = 0.5, upper = 1)$value
#   })
# }, lower = 0.5, upper = 1)$value
#
# # Mostrar resultados
# cat("Marginal de X en x = 0.5:", f_x(0.5), "\n")
# cat("Marginal de Y en y = 0.5:", f_y(0.5), "\n")
# cat("P(X > 0.5, Y > 0.5) =", prob, "\n")
#
# # Ejercicio 9B
#
# # Definición de la función de densidad conjunta
# f <- function(x, y) {
#   ifelse(x > 1 & x < 3 & y > 1 & y < 2, (3*x - y)/9, 0)
# }
#
# # Marginal de X
# fx <- function(x) {
#   sapply(x, function(xi) {
#     integrate(function(y) f(xi, y), lower = 1, upper = 2)$value
#   })
# }
#
# # Marginal de Y
# fy <- function(y) {
#   sapply(y, function(yi) {
#     integrate(function(x) f(x, yi), lower = 1, upper = 3)$value
#   })
# }
#
# # Chequear independencia
# # Si f(x,y) = fX(x)*fY(y) para todo (x,y) en el dominio, son independientes
# check_independence <- function(x, y) {
#   abs(f(x, y) - fx(x)*fy(y)) < 1e-6
# }
#
# # Calcular P(X > 2)
# p_x_greater_2 <- integrate(function(x) {
#   fx(x)
# }, lower = 2, upper = 3)$value

#Punto 10
# Valores posibles de X
x <- c(1, 3, 5, 7)
```

```r
# Probabilidades obtenidas de la cdf
p <- c(0.4, 0.2, 0.2, 0.2)
# Construir la pmf en forma de tabla
pmf <- data.frame(X = x, P = p)
pmf
```

```
##   X   P
## 1 1 0.4
## 2 3 0.2
## 3 5 0.2
## 4 7 0.2
```

```r
# (b) Calcular P(4 < X <= 7)
prob <- sum(p[x > 4 & x <= 7])
prob
```

```
## [1] 0.4
```

```r
#Punto 11
x <- 2
y <- 2

funcionxy <- (9/16) * (1/4)^(x + y)

mx <- (3/4) * (1/4)^x
my <- (3/4) * (1/4)^y
mxmy <- mx * my

funcionxy
```

```
## [1] 0.002197266
```

```r
mxmy
```

```
## [1] 0.002197266
```

```r
# Como la respuesta en ambos casos es igual, quiere decir que son independientes.

#Punto 12
#A coin is biased so that the probability of a head is three times the probability of a tail.
#Compute the expected number of tails if this coin is tossed twice.
#Let  =  (tail) and  (head) = 3, with  + 3 = 1. Find [ ] where  is the number of tails
#in two tosses.
# Se resolverá el ejercicio con una distribución binomial
# Parámetros
# Como se menciona en el enunciado, al ser cara 3 veces mayor que sello y al despejar p en la función q
n <- 2          # número de lanzamientos
p <- 1/4        # probabilidad de obtener sello

# Distribución de T ~ Binomial(n=2, p=0.25)
valores <- 0:n
```

```r
probabilidades <- dbinom(0:n,n, p)

# Mostrar distribución
data.frame(0:n, probabilidades)
```

```
##   X0.n probabilidades
## 1    0         0.5625
## 2    1         0.3750
## 3    2         0.0625
```

```r
# Valor esperado
Ex <- sum(valores * probabilidades)
Ex
```

```
## [1] 0.5
```

```r
#El número esperado de sellos al lanzar la moneda dos veces es del 50%

# Punto 13
p_j = 4/52
p_q = 4/52
p_k = 4/52
p_a = 4/52

p_jq <- p_j + p_q
p_ka <- p_k + p_a
p_otra <- 1 - p_jq - p_ka

pago_jq <- 3
pago_ka <- 5
pago_otra <- 0

Ex <- pago_jq * p_jq + pago_ka * p_ka + pago_otra * p_otra
Ex
```

```
## [1] 1.230769
```

```r
# El precio justo de entrada es de $ 1.23

#Punto 14

# Valores posibles de X
x <- c(-3, 6, 9)

# Probabilidades
p <- c(1/6, 1/2, 1/3)

# Definimos g(X) = (2X + 1)^2
g <- (2*x + 1)^2
#De esta manera se calcula g(x) cuando x valen -3,6 y 9, las convertimos en un vector para que sea más 
g
```

```
## [1]  25 169 361
```

```r
# Valor esperado E[g(X)]
Eg <- sum(g * p)
Eg
```

```
## [1] 209
```

```r
#Punto 15
#A large industrial company buys several new word processors at the end of each year;
#the exact number depends on the repair frequency of the previous year.
#Suppose the number of word processors, , purchased each year has the following probability
#distribution:
#If the cost of the desired model is $1200 per unit,
#and at the end of the year the company obtains a discount of 50 2 dollars,
#how much does the company expect to spend on new word processors during this year?

x <- c(0,1,2,3)
p <- c(1/10, 3/10, 2/5, 1/5)
#Se calcula primero el valor esperado de las variables, dejando todo como vectores para facilitar las o
EX  <- sum(x * p)
EX
```

```
## [1] 1.7
```

```r
#Luego, se calcula el valor esperado de las variables elevadas al cuadrado, dejando todo como vectores
EX2 <- sum(x^2 * p)
EX2
```

```
## [1] 3.7
```

```r
#Por último, se reemplaza en la ecuación y se obtiene el resultado
Ecost <- 1200 * EX - 50 * EX2
data.frame(x = x, p = p, x2 = x^2, contrib = (1200*x - 50*x^2) * p)
```

```
##   x   p x2 contrib
## 1 0 0.1  0       0
## 2 1 0.3  1     345
## 3 2 0.4  4     880
## 4 3 0.2  9     630
```

```r
Ecost
```

```
## [1] 1855
```

```r
#La compañía espera gastar $1,855 en procesadores ese año.
```