ESPERANZA CONDICIONAL

Nexus-Probability

CURSO 3 (PROCESOS ESTOCÁSTICOS I)

PARTE 1 / LECCIÓN 2

Ahora, vamos a definir el valor esperado de una variable aleatoria dado que un evento a ocurrido para otra variable cuando se conoce la distribución conjunta de las dos variables.

Definición 1 (Esperanza Condicional.) *Sea (X, Y) un vector aleatorio continuo con función de densidad $f_{X,Y}(x, y)$ y sea y un valor tal que $f_Y(y) \neq 0$. La esperanza condicional de X dado $Y = y$ es la esperanza de la funcion de densidad condicional $f_{X|Y}(x|y)$, cuando existe, es decir,*

$$\mathbb{E}[X|Y = y] = \int_{-\infty}^{\infty} x \cdot f_{X|Y}(x|y)dx.$$

Efectuando un cambio en el orden de las integrales es inmediato comprobar que:

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} \mathbb{E}[X|Y = y] \cdot f_Y(y)dy,$$

esta forma nos recuerda el teorema de probabilidad total, pero esta vez en términos de esperanzas. En el caso cuando el vector (X, Y) es discreto la definición es análoga:

$$\begin{aligned}\mathbb{E}[X|Y = y] &= \sum_x x \cdot f_{X|Y}(x|y) \\ &= \sum_x x \cdot P(X = x|Y = y),\end{aligned}$$

suponiendo nuevamente que $f_Y(y) \neq 0$ y que la suma indica es absolutamente convergente. Y nuevamente, efectuando un cambio en el orden de las sumas se encuentra la expresión

$$\mathbb{E}[X] = \sum_y \mathbb{E}[X|Y = y] \cdot P(Y = y).$$

En cualquier caso observe ademas que cuando X y Y son independientes,

$$\mathbb{E}(X|Y = y) = \mathbb{E}[X]$$

Ejercicios

Los siguientes ejercicios propuestos tendrán solución en **Python**, por lo que te invitamos a ejecutar el código en tu computadora.

Ejercicio 1 *Un experimento consiste en lanzar un dado equilibrado repetidas veces hasta obtener alguno de los resultados por segunda vez. Encuentre el número esperado de lanzamientos en este experimento.*

Solución.

```python
import numpy as np
from scipy.integrate import quad

# Ejercicio 1: Numero esperado de lanzamientos para obtener un resultado por segunda vez
def expected_rolls_to_repeat():
    sides = 6
    expected_value = 0
    prob_not_repeated = 1
    for i in range(1, sides + 1):
        expected_value += i * prob_not_repeated * (1 / sides)
        prob_not_repeated *= (sides - i) / sides
    return expected_value + sides  # Sumar la probabilidad de la repeticion en el ultimo lanzamiento

print("Ejercicio 1: Numero esperado de lanzamientos =", expected_rolls_to_repeat())
```

```
Ejercicio 1: Numero esperado de lanzamientos = 7.0
```

Ejercicio 2 *Se lanza un dado equilibrado dos veces consecutivas. Sea X el resultado del primer lanzamiento y sea Y el resultado del segundo lanzamiento. Calcule*

$$\mathbb{E}[X|X+Y=6].$$

Solución.

```python
# Ejercicio 2: E[X|X + Y = 6]
def conditional_expectation_x_given_x_plus_y_equals_6():
    results = []
    for x in range(1, 7):  # Valores posibles de X
        prob_x_given_sum_6 = 1 / 5  # Condici n de X + Y = 6 restringe Y
        results.append(x * prob_x_given_sum_6)
    return sum(results)

print("Ejercicio 2: E[X|X + Y = 6] =", conditional_expectation_x_given_x_plus_y_equals_6())
```

```
Ejercicio 2: E[X|X + Y = 6] = 4.2
```

Ejercicio 3 *Sea X con distribución Poisson(β) en donde β es una variable aleatoria con distribución unif(1,2). Condicionado sobre el valor de β encuentre $\mathbb{E}[X]$.*

Solución.

```python
# Ejercicio 3: E[X] con X ~ Poisson(BETA), BETA ~ Uniforme(1, 2)
def poisson_expectation():
    # Integrar E[X|BETA] = BETA con f(BETA) = 1 en [1, 2]
    result, _ = quad(lambda beta: beta * 1, 1, 2)
    return result

print("Ejercicio 3: E[X] con X ~ Poisson(beta) =", poisson_expectation())
```

```
Ejercicio 3: E[X] con X ~ Poisson(beta) = 1.5
```

Ejercicio 4 *Sea X con distribución Binomial(γ, p) en donde γ es una variable aleatoria con distribución unif(1,2). Condicionado sobre el valor de γ encuentre $\mathbb{E}[X]$.*

Solución.

```
# Ejercicio 4: E[X] con X ~ Binomial(gamma, p), gamma ~ Uniforme(1, 2)
def binomial_expectation(p=0.5):
    # Integrar E[X|gamma] = gamma * p con f(gamma) = 1 en [1, 2]
    result, _ = quad(lambda gamma: gamma * p * 1, 1, 2)
    return result

print("Ejercicio 4: E[X] con X ~ Binomial(gamma, p) =", binomial_expectation())
```

```
Ejercicio 4: E[X] con X ~ Binomial(gamma, p) = 0.75
```