# DISTRIBUCIÓN CONDICIONAL

*Nexus-Probability*

**CURSO 3 (PROCESOS ESTOCÁSTICOS I)**

*PARTE 1 / LECCIÓN 1*

Recordemos que la probabilidad condicional de un evento $A$ dado un evento $B$ está dada por:

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}$$

Esta definición puede extenderse al caso de funciones de probabilidad o de densidad y también para el caso de funciones de distribución.

**Definición 1 (Función de Densidad de $X|Y = y$)** *Sea $(X, Y)$ un vector aleatorio discreto (o continuo) y con función de probabilidad (o de densidad) $f_{X,Y}(x, y)$. Sea $y$ un valor de la variable $Y$ tal que $f_Y(y) \neq 0$. A la función $x \mapsto f_{X|Y}(x, y)$ definida de la siguiente manera se le conoce como función de probabilidad (o densidad) de $X$ dado que $Y = y$*

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x, y)}{f_Y(y)} \tag{1}$$

Observe que la función dada por (1) se le considera como una función de $x$ y que el valor de $y$ es fijo y que puede considerarse como un parámetro de dicha función, i.e., para cada valor fijo de $y$ se tiene una función diferente. En el caso discreto de la expresión (1) es efectivamente la definición de probabilidad condicional

$$f_{X|Y}(x|y) = \frac{P(X = x, Y = y)}{P(Y = y)}$$

sin embargo, recordemos que en el caso continuo las expresiones $f_{X,Y}(x, y)$ y $f_Y(y)$ **no son probabilidades**. Sumando o integrando sobre los posibles valores $x$, es inmediato comprobar que la función dada por (1) es efectivamente un función de probabilidad o de densidad. Observe además que cuando $X$ y $Y$ son independientes,

$$f_{X|Y}(x|y) = f_X(x)$$

La formula (1) puede extenderse de manera análoga al caso de vectores de dimensión mayor. Por ejemplo, para un vector aleatorio de dimensión tres $(X, Y, Z)$ pueden calcularse funciones de densidad condicionales como $f_{X|Y,Z}(x|y, z)$ o $f_{X,Z|Y}(x, z|y)$,. Por tanto podemos dar la siguiente definición mas extendida:

**Definición 2 (Función de Distribución Condicional.)** *Sea $(X, Y)$ un vector aleatorio con función de probabilidad o densidad $f_{X,Y}(x, y)$. Sea y un valor de $Y$ tal que $f_Y(y) \neq 0$. La **función de distribución condicional** de $X$ dado $Y = y$ es la función*

$$f_{X|Y}(x|y) = \begin{cases} \sum_{u \leq x} f_{X,Y}(u, y) & \text{en el caso discreto,} \\ \int_{-\infty}^{x} f_{X,Y}(u, y) & \text{en el caso continuo.} \end{cases} \tag{2}$$

De esta forma la función de distribución condicional se calcula como la suma o integral de la correspondiente función de probabilidad o densidad condicional. Nuevamente observamos que cuando $X$ y $Y$ son independientes,

$$F_{X|Y}(x|y) = F_X(x)$$

# Ejercicios

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

**Ejercicio 1** *Sea $(X, Y)$ un vector aleatorio continuo con función de densidad*

$$f_{X,Y}(x, y) = \begin{cases} 6x^2 y & \text{si } 0 < x < 1, 0 < y < 1, \\ 0 & \text{en otro caso.} \end{cases}$$

*Calcule las siguientes funciones:*

*a)* $f_{X|Y}(x|y)$, $0 < y < 1$.

*b)* $F_{X|Y}(x|y)$, $0 < y < 1$.

## Solución.

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

# Ejercicio 1: Calculo de densidades y funciones de distribucion
```

```python

# Funcion de densidad conjunta f_{X,Y}(x, y)
def f_xy(x, y):
    if 0 < x < 1 and 0 < y < 1:
        return 6 * (x ** 2) * y
    else:
        return 0

# Funcion de densidad marginal de Y
def f_y(y):
    return quad(lambda x: f_xy(x, y), 0, 1)[0]  # Integrar respecto a x de 0 a 1

# a) Calculo de f_{X|Y}(x|y)
def f_x_given_y(x, y):
    return f_xy(x, y) / f_y(y) if f_y(y) != 0 else 0

# b) Calculo de F_{X|Y}(x|y)
def F_x_given_y(x, y):
    integral, _ = quad(lambda u: f_x_given_y(u, y), 0, x)
    return integral

# Resultados para el Ejercicio 1
y_val = 0.5  # Ejemplo para y = 0.5
print("Ejercicio 1:")
print(f"f_X|Y(x|{y_val}) = ", [f_x_given_y(x, y_val) for x in np.linspace(0, 1, 5)])
print(f"F_X|Y(x|{y_val}) = ", [F_x_given_y(x, y_val) for x in np.linspace(0, 1, 5)])

```

```
Ejercicio 1:
f_X|Y(x|0.5) =  [0.0, 0.1875, 0.75, 1.6875, 0.0]
F_X|Y(x|0.5) =  [0.0, 0.015625, 0.125, 0.421875, 1.0]
```

**Ejercicio 2** *Sea $(X, Y)$ un vector discreto con función de probabilidad dada por la siguiente tabla:*

| x/y | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0.1 | 0.05 | 0.1 |
| 1 | 0.05 | 0.2 | 0.1 |
| 2 | 0.05 | 0.05 | 0.3 |

*Calcule las siguientes funciones:*

*a)* $f_{X|Y}(x|0)$

*b)* $f_{X|Y}(x|1)$

*c)* $F_{X|Y}(x|1)$

## Solución.

```
# Ejercicio 2: Calculo de funciones discretas

# Matriz de probabilidades P(X, Y)
P_XY = np.array([[0.1, 0.05, 0.1],
                 [0.05, 0.2, 0.1],
                 [0.05, 0.05, 0.3]])

# a) Calculo de f_X|Y(x|0)
f_X_given_Y_0 = P_XY[:, 0] / np.sum(P_XY[:, 0])

# b) Calculo de f_X|Y(x|1)
f_X_given_Y_1 = P_XY[:, 1] / np.sum(P_XY[:, 1])

# c) Calculo de F_X|Y(x|1)
F_X_given_Y_1 = np.cumsum(f_X_given_Y_1)

# Resultados para el Ejercicio 2
print("\nEjercicio 2:")
print("f_X|Y(x|0) =", f_X_given_Y_0)
print("f_X|Y(x|1) =", f_X_given_Y_1)
print("F_X|Y(x|1) =", F_X_given_Y_1)
```

```
Ejercicio 2:
f_X|Y(x|0) = [0.5  0.25 0.25]
f_X|Y(x|1) = [0.16666667 0.66666667 0.16666667]
```

```
F_X|Y(x|1) = [0.16666667 0.83333333 1.        ]
```

**Ejercicio 3** *Se lanza un dado equilibrado dos veces. Sea $X$ el resultado del primer lanzamiento y sea $Y$ el mayor de los dos resultados.*

*a) Encuentre la función de probabilidad conjunta de $X$ y $Y$.*

*b) Calcule las funciones $f_{Y|X}(y|x = 3)$ y $f_{X|Y}(x|y = 3)$*

## Solución.

```
# Ejercicio 3: Lanzamiento de dado
from itertools import product
from collections import Counter

# Calcular la funcion de probabilidad conjunta P(X, Y)
lanzamientos = list(product(range(1, 7), repeat=2))
xy_pairs = [(x, max(x, y)) for x, y in lanzamientos]
contador = Counter(xy_pairs)

# a) Funcion de probabilidad conjunta
P_XY_dado = {k: v / len(lanzamientos) for k, v in contador.items()}

# b) Calcular f_Y|X(y|x=3) y f_X|Y(x|y=3)
def f_y_given_x(y, x):
    prob_x = sum(v for (i, j), v in P_XY_dado.items() if i == x)
    return P_XY_dado.get((x, y), 0) / prob_x if prob_x != 0 else 0
def f_x_given_y(x, y):
    prob_y = sum(v for (i, j), v in P_XY_dado.items() if j == y)
    return P_XY_dado.get((x, y), 0) / prob_y if prob_y != 0 else 0

 # Resultados para el Ejercicio 3
 print("\nEjercicio 3:")
 print("P(X, Y) =", P_XY_dado)
 print("f_Y|X(y|x=3) =", {y: f_y_given_x(y, 3) for y in range(1, 7)})
 print("f_X|Y(x|y=3) =", {x: f_x_given_y(x, 3) for x in range(1, 7)})
```