

DEFINICIÓN DE VARIABLES ALEATORIAS

Nexus-Probability

CURSO 1 (PROBABILIDAD I)

PARTE 3 / LECCIÓN 2

Definición 1 (Funciones de Distribución) La **función de distribución acumulada (FDA)** $F(\cdot)$ de la variable aleatoria X está definida para cualquier número real x :

$$F_X(x) = P(X \leq x).$$

La FDA tiene las siguientes propiedades:

- (i) $F(x)$ es una función no decreciente de x ,
- (ii) $\lim_{x \rightarrow \infty} F(x) = F(\infty) = 1$,
- (iii) $\lim_{x \rightarrow -\infty} F(x) = F(-\infty) = 0$.

En palabras, $F(x)$ denota la probabilidad de que la variable aleatoria X tome un valor que sea menor o igual que x .

La FDA describe completamente la distribución de una variable aleatoria. Si la FDA es diferenciable, su derivada es la **función de densidad de probabilidad (PDF)** para variables continuas o la **función de probabilidad (PMF)** para discretas.

Ejercicios

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

Ejercicio 1 Simula valores de una distribución uniforme continua en el intervalo $[0, 1]$ y calcula su FDA.

Solución.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  # Simular valores de una distribución uniforme entre 0 y 1
5  n_simulaciones = 1000
6  resultados = np.random.uniform(0, 1, n_simulaciones)
7
8  # Calcular la FDA
9  valores = np.sort(resultados)
10 cdf = np.arange(1, n_simulaciones + 1) / n_simulaciones
11
12 # Graficar la FDA
13 plt.plot(valores, cdf, label="CDF Uniforme (0, 1)")
14 plt.title("Funcion de Distribucion Acumulativa (FDA) para
una Uniforme [0, 1]")
15 plt.xlabel("X")
16 plt.ylabel("F(X)")
17 plt.grid(True)
18 plt.legend()
19 plt.show()
20

```

Ejercicio 2 Supón que tienes un conjunto de datos específico:

$X = 3, 6, 7, 8, 8, 10, 15, 20, 20, 20$

Escribe un programa en Python que calcule la CDF para este conjunto de datos y grafique los resultados.

Solución.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  # Conjunto de datos dado
5  datos = np.array([3, 6, 7, 8, 8, 10, 15, 20, 20, 20])
6
7  # Ordenar los datos
8  datos_ordenados = np.sort(datos)
9
10 # Calcular la CDF
11 y_values = np.arange(1, len(datos_ordenados)+1) / len(
datos_ordenados)
12
13 # Graficar la CDF

```

```

14     plt.step(datos_ordenados, y_values, where="post", label="CDF
15 ")
16     plt.title("Funcion de Distribucion Acumulativa (CDF)")
17     plt.xlabel("X")
18     plt.ylabel("F(X)")
19     plt.grid(True)
20     plt.legend()
21     plt.show()

```

Ejercicio 3 Define una función de distribución acumulativa $F(x)$ que esté definida por partes de la siguiente manera:

$$F(x) = \begin{cases} 0, & \text{si } x < 0, \\ \frac{x}{2}, & \text{si } 0 \leq x < 2, \\ 1, & \text{si } x \geq 2. \end{cases}$$

Implementa esta función $F(x)$ en Python, y calcula su valor para:

$$x = -1, 0, 1, 2, 3$$

Asegúrate de que los resultados sean consistentes con las propiedades de una función de distribución.

Solución.

```

1     import numpy as np
2
3     # Definir la funcion de distribucion acumulativa por partes
4     def F(x):
5         if x < 0:
6             return 0
7         elif 0 <= x < 2:
8             return x / 2
9         else:
10            return 1
11
12    # Calcular la FDA para varios valores de x
13    x_values = [-1, 0, 1, 2, 3]
14    for x in x_values:
15        print(f"F({x}) = {F(x)}")
16

```

Ejercicio 3: $F(-1) = 0$, $F(0) = 0.0$, $F(1) = 0.5$, $F(2) = 1$, $F(3) = 1$

Ejercicio 4 Definir una función de distribución acumulativa $F(x)$ de la siguiente manera:

$$F(x) = \begin{cases} 0, & \text{si } x < -1, \\ \frac{x+1}{3}, & \text{si } -1 \leq x < 2, \\ 1, & \text{si } x \geq 2. \end{cases}$$

Implementa esta función $F(x)$ en Python, y calcula su valor para:

$$x = -2, -1, 0, 1, 2, 3$$

Solución

```
1     def F1(x):
2     if x < -1:
3         return 0
4     elif -1 <= x < 2:
5         return (x + 1) / 3
6     else:
7         return 1
8
9     valores_x1 = [-2, -1, 0, 1, 2, 3]
10    resultados1 = [F1(x) for x in valores_x1]
11
12    print("Resultados Ejercicio 4:", resultados1)
13
```

Resultados Ejercicio 4: $[0, 0.0, 0.3333333333333333, 0.6666666666666666, 1, 1]$