VALOR ESPERADO DE VARIABLES ALEATORIAS

Nexus-Probability

CURSO 2 (PROBABILIDAD II)

PARTE 1 / LECCIÓN 1

Valor Esperado de una Variable Aleatoria Discreta

Uno de los conceptos más importantes de la teoría de probabilidad es el valor esperado de una variable aleatoria.

Definición 1 (Valor esperado de una variable aleatoria discreta.) *Si X es una variable aleatoria discretacon función de masa de probabilidad $p(x)$, entonces el valor esperado de X, denotado por $E[X]$, se define por:*

$$E[X] = \sum_{x:p(x)} xp(x) \tag{1}$$

En otras palabras, el valor esperado de X es un promedio ponderado de los valores posibles valores que toma X, ponderados por la probabilidad de que X asuma dicho valor.

Teorema 1 *Si X es una variable aleatoria discreta que toma uno a uno los valores x_i, $i \geq 1$, con probabilidades $p(x_i)$ respectivamente, entonces, para cualquier función real g*

$$E[X] = \sum_{i} g(x_i)p(x_i) \tag{2}$$

Teorema 2 *Si a y b son constantes, entonces:*

$$E[aX + b] = aE[X] + b \tag{3}$$

Otra característica que nos aporta valiosa información sobre el comportamiento de la variable aleatoria X es la variación, aunque el valor esperado nos da un promedio ponderado de los valores que toma la variable aleatoria no nos dice nada sobre la variación o disperción de los valores que toma X.

Definición 2 (Varianza de una variable aleatoria discreta.) *Si X es una variable aleatoria con media μ, entonces la varianza de X, denotada por $Var(X)$ se define por*

$$Var(X) = E[(X - \mu)^2] \tag{4}$$

Valor Esperado de una Variable Aleatoria Continua

Al igual que en el caso discreto también tenemos la definición de valor esperado para una variable aleatoria continua. Al igual que en el caso discreto, en el caso continuo podemos interpretar el valor esperado de una variable aleatoria continua como un promedio ponderado de la siguiente manera. Si X es una variable aleatoria continua con función de densidad $f(x)$, entonces

$$f(x)dx \approx P(x \leq X \leq x + dx)$$

Definición 3 (Valor Esperado de una Variable Aleatoria Continua) *Si X es una variable aleatoria continua con función de densidad $f(x)$ se define el valor esperado de esta variable aleatoria como*

$$E[X] = \int_{-\infty}^{\infty} x f(x) dx \tag{5}$$

Teorema 3 *Si X es una variable aleatoria continua con función de densidad $f(x)$, entonces, para cualquien función real g,*

$$E[g(X)] = \int_{-\infty}^{\infty} g(x) f(x) dx \tag{6}$$

Ejercicios.

Ejercicio 1 *Supón que tienes un sistema de colas $M/M/1$ donde las llegadas siguen un proceso de Poisson con tasa $\lambda = 4$ clientes por minuto, y el tiempo de servicio sigue una distribución exponencial con parámetro $\mu = 6$ servicios por minuto. El tiempo en el sistema (espera más servicio) de un cliente se puede modelar como una variable aleatoria T. Calcula el valor esperado del tiempo total que un cliente pasa en el sistema (espera más servicio) y simula la distribución de los tiempos totales usando Python para corroborar el valor esperado teórico.*

Solución.

```
import numpy as np

# Par metros
lambda_rate = 4  # tasa de llegadas
mu = 6  # tasa de servicio

# Valor esperado te rico
expected_T = 1 / (mu - lambda_rate)
print(f"Valor esperado te rico del tiempo total en el sistema:
    {expected_T:.4f} minutos")

# Simulaci n
np.random.seed(42)
num_simulations = 10000
service_times = np.random.exponential(1/mu, num_simulations)
waiting_times = np.random.exponential(1/(mu - lambda_rate),
    num_simulations)
total_times = service_times + waiting_times

# Valor esperado emp rico
empirical_expected_T = np.mean(total_times)
print(f"Valor esperado emp rico del tiempo total en el sistema
    (simulaci n): {empirical_expected_T:.4f} minutos")


```

Ejercicio 2 *Un jugador tiene M unidades monetarias y apuesta en un juego donde puede ganar una unidad con probabilidad $p = 0.4$ y perder una unidad con probabilidad $1p = 0.6$. El juego termina cuando el jugador se queda sin dinero o alcanza un capital de $N = 10$ unidades. Calcula el valor esperado del número de juegos hasta que el jugador quiebre o alcance N unidades. Simula este proceso con Python para estimar el valor esperado y compáralo con el teórico.*

Solución.

```
import numpy as np

# Par metros
M = 5  # capital inicial
N = 10  # capital objetivo
p = 0.4  # probabilidad de ganar

```

```
# Simulaci n
def simulate_game(M, N, p):
    capital = M
    games = 0
    while 0 < capital < N:
        if np.random.random() < p:
            capital += 1
        else:
            capital -= 1
        games += 1
    return games

np.random.seed(42)
num_simulations = 10000
games_until_end = [simulate_game(M, N, p) for _ in range(
    num_simulations)]

# Valor esperado emp rico
empirical_expected_games = np.mean(games_until_end)
print(f"Valor esperado emp rico del n mero de juegos: {
    empirical_expected_games:.2f}")
```

Ejercicio 3 *Una aseguradora emite pólizas que cubren pérdidas X con una distribución exponencial con parámetro $= 1/1000$. Sin embargo, el asegurado tiene un deducible de $D = 500$, lo que significa que la aseguradora solo paga la cantidad de pérdida que excede D.*
Calcula el valor esperado del pago que realiza la aseguradora en cada siniestro y simula el valor esperado con Python.

Solución.

```
import numpy as np

# Par metros
lambda_rate = 1/1000  # par metro exponencial
D = 500  # deducible

# Valor esperado te rico
expected_payment = np.exp(-lambda_rate * D) * (1/lambda_rate + D
    )
```

```
print(f"Valor esperado te rico del pago: {expected_payment:.2f}")

# Simulaci n
np.random.seed(42)
num_simulations = 10000
losses = np.random.exponential(1/lambda_rate, num_simulations)
payments = np.maximum(0, losses - D)

# Valor esperado emp rico
empirical_expected_payment = np.mean(payments)
print(f"Valor esperado emp rico del pago (simulaci n): {empirical_expected_payment:.2f}")
```

Ejercicio 4 *Una partícula realiza una caminata aleatoria en una línea con pasos de longitud 1. En cada paso, la partícula se mueve a la izquierda con probabilidad* $p = 0.5$ *y a la derecha con probabilidad* $1p = 0.5$*. Si la partícula empieza en la posición* $x = 0$*, ¿cuál es el valor esperado del tiempo hasta que la partícula regrese a la posición* $x = 0$ *por primera vez?*

Solución.

```
import numpy as np

# Simulaci n
def random_walk_return():
    position = 0
    steps = 0
    while position != 0 or steps == 0:  # no contar el inicio como regreso
        if np.random.random() < 0.5:
            position -= 1
        else:
            position += 1
        steps += 1
    return steps

np.random.seed(42)
num_simulations = 10000
return_times = [random_walk_return() for _ in range(num_simulations)]
```

```

# Valor esperado emp rico
empirical_expected_return_time = np.mean(return_times)
print(f"Valor esperado emp rico del tiempo de regreso a 0: {empirical_expected_return_time:.2f} pasos")



```

Ejercicio 5 *Se tiene una urna con 3 bolas rojas y 2 bolas azules. Se selecciona una bola al azar, se registra su color, y luego se devuelve a la urna. ¿Cuál es el valor esperado del número de extracciones necesarias para observar al menos una bola de cada color?*

Solución.

```
import numpy as np

# Simulaci n
def simulate_draws_until_both_colors():
    seen_colors = set()
    draws = 0
    while len(seen_colors) < 2:  # esperamos ver ambas (rojo y azul)
        draw = np.random.choice(['rojo', 'azul'], p=[3/5, 2/5])
        seen_colors.add(draw)
        draws += 1
    return draws

np.random.seed(42)
num_simulations = 10000
draws_until_both = [simulate_draws_until_both_colors() for _ in range(num_simulations)]

# Valor esperado emp rico
empirical_expected_draws = np.mean(draws_until_both)
print(f"Valor esperado emp rico del n mero de extracciones: {empirical_expected_draws:.2f}")



```