

**Instituto Tecnológico de Costa Rica**  
**Escuela de Computación**

Programa de Ciencias de los Datos  
**Curso: Estadística**

Profesor: Ph. D. Saúl Calderón Ramírez

**QUIZ 3**

Entrega: Jueves 16 de Mayo, a través del TEC digital  
Debe subir un *pdf* con la respuesta.

Valor: 100 pts.

Puntos Obtenidos: \_\_\_\_\_

Nota: \_\_\_\_\_

Nombre del (la) estudiante: Marco Ferraro Rodriguez

1. Su equipo tiene por objetivo construir un modelo Bayesiano que estime si en 1 mes será necesario que una empresa eléctrica necesite racionar el suministro a sus clientes. Para ello, dado que la empresa eléctrica está restringida a utilizar solamente fuentes de energía renovables (agua y viento), las variables de entrada para predecir si habrá cortes eléctricos en un mes ( $t = 1$ ) o no ( $t = 0$ ) serán la precipitación  $P$  promedio del año hasta ese mes (medida en  $l/m^2$ ) y la velocidad del viento  $V$  promedio (medida en  $km/h$ ). Al cabo de un histórico de 5 años, la empresa eléctrica recopiló los datos en las tablas 1 y 2.

	$p = 400$	$p = 500$	$p = 600$	$p = 700$	$p = 800$	$p = 900$	$p = 1000$	$p = 1100$	$p = 1200$	$p = 1300$
$t = 0$	5	6	9	11	12	16	18	13	5	2
$t = 1$	20	13	12	4	2	1	2	1	2	1

Cuadro 1: Datos para la variable aleatoria  $P$ .

	$v = 5$	$v = 10$	$v = 15$	$v = 20$	$v = 25$	$v = 30$	$v = 35$	$v = 40$	$v = 45$	$v = 50$
$t = 0$	2	3	5	15	6	3	1	0	0	0
$t = 1$	22	15	8	3	2	1	0	0	0	0

Cuadro 2: Datos para la variable aleatoria  $V$ .

1. **(20 puntos)** Usando pytorch, use el histograma de los datos anteriores para estimar las densidades  $p(m_1|t=0)$ ,  $p(m_1|t=1)$ ,  $p(m_2|t=0)$ ,  $p(m_2|t=1)$ .  $m_1 = p$  (primer dimension) y  $m_2 = v$  (segunda dimension)
2. **(50 puntos)** Utilizando los graficos anteriores, ajuste un modelo Gaussiano o exponencial, según sea necesario (realice la justificación según lo observado en tales graficos), a cada una

de las densidades  $p(m_1|t=0)$ ,  $p(m_1|t=1)$ ,  $p(m_2|t=0)$ ,  $p(m_2|t=1)$ . Muestre los pasos intermedios para estimar los parametros de tales modelos y documentelos. Grafique el modelo ajustado y muestre las tablas 1 y 2 con las probabilidades estimadas por estos nuevos modelos.

3. (30 puntos) Usando Bayes, estime si habra corte electrico para una entrada  $m_1 = 500$  y  $m_2 = 10$ . Muestre y explique los pasos intermedios.

## Respuesta

### Implementación de la solución Cálculo de Probabilidades Condicionales

Se calculan las probabilidades condicionales para P y V dado t (0 o 1) utilizando la función `calcular_probabilidad_condicional`.

### Graficación de Histogramas

Se grafican los histogramas de las probabilidades condicionales utilizando la función `graficar_histograma`. Los histogramas muestran la distribución de las probabilidades condicionales para cada valor de p y v dado t. Además, si se especifica, se ajusta y grafica una distribución normal o exponencial sobre el histograma.

```

1 import torch
2 import matplotlib.pyplot as plt
3 import numpy as np
4 from scipy.stats import norm
5
6
7 def calcular_probabilidad_condicional(p, P_t):
8     """
9     Calcula la probabilidad condicional P(p | t) para todos los valores de p dado t.
10
11     Args:
12     p (torch.Tensor): Tensor con los valores de p.
13     P_t (torch.Tensor): Tensor con las ocurrencias de p dado t.
14
15     Returns:
16     torch.Tensor: Tensor con las probabilidades condicionales P(p | t).
17     """
18     total_t = P_t.sum()
19     probabilidades = P_t / total_t
20     return probabilidades
21
22
23 def graficar_histograma(p, probabilidades, t_value, title, width=80, gauss=False,
24     exponencial=False, lambda_param=0.5):
25     """
26     Grafica un histograma de las probabilidades condicionales y ajusta una distribución
27     normal.
28
29     Args:
30     p (torch.Tensor): Tensor con los valores de p.
31     probabilidades (torch.Tensor): Tensor con las probabilidades condicionales.

```

```

30 t_value (int): Valor de t (0 o 1).
31 title (str): T tulo del histograma.
32 """
33 # Crear el histograma
34 plt.bar(p.numpy(), probabilidades.numpy(), width=width,
35         align='center', alpha=0.7, color='blue', edgecolor='black')
36
37 if gauss:
38     # Ajustar una distribuci n normal
39     mu = (p * probabilidades).sum()
40     sigma = torch.sqrt((p - mu) ** 2 * probabilidades).sum())
41
42     pdf = (1 / (sigma * np.sqrt(2 * np.pi))) * \
43           np.exp(-0.5 * ((p - mu) / sigma) ** 2)
44
45     dx = p[1] - p[0]
46     pdf = pdf * dx
47
48     plt.plot(p.numpy(), pdf, color='red', linewidth=2)
49
50 if exponencial:
51     # Ajustar una distribuci n exponencial
52     min_p = p.min().item()
53     max_p = p.max().item()
54     x = np.linspace(min_p, max_p, p.numpy().size)
55
56     # Funci n de densidad de probabilidad (pdf)
57     pdf = lambda_param * np.exp(-lambda_param * x)
58
59     # Normalizar la pdf para que la suma sea 1
60     dx = x[1] - x[0] # Diferencia entre puntos consecutivos
61     pdf_normalized = pdf / (pdf.sum() * dx)
62     pdf_normalized = pdf_normalized * dx
63     plt.plot(x, pdf_normalized, color='red', linewidth=2)
64
65
66 # Aadir etiquetas y t tulo
67 plt.xlabel('P')
68 plt.ylabel(f'P(P | t = {t_value})')
69 plt.title(title)
70
71 # Mostrar el gr fico
72 plt.show()
73
74
75 # Variables de precipitaciones
76 p = torch.tensor([400, 500, 600, 700, 800, 900, 1000,
77                  1100, 1200, 1300], dtype=torch.float32)
78 v = torch.tensor([5, 10, 15, 20, 25, 30, 35, 40, 45, 50], dtype=torch.float32)
79
80 # Datos para la variable aleatoria P
81 P_t0 = torch.tensor([5, 6, 9, 11, 12, 16, 18, 13, 5, 2], dtype=torch.float32)
82 P_t1 = torch.tensor([20, 13, 12, 4, 2, 1, 2, 1, 2, 1], dtype=torch.float32)
83
84 # Datos para la variable aleatoria V
85 V_t0 = torch.tensor([2, 3, 5, 15, 6, 3, 1, 0, 0, 0], dtype=torch.float32)

```

```
86 V_t1 = torch.tensor([22, 15, 8, 3, 2, 1, 0, 0, 0, 0], dtype=torch.float32)
```

Listing 1: Implementación de Funciones

## Respuesta 1)

```
1
2 # Calcular las probabilidades condicionales
3 probabilidades_p_t0 = calcular_probabilidad_condicional(p, P_t0)
4 probabilidades_p_t1 = calcular_probabilidad_condicional(p, P_t1)
5 probabilidades_v_t0 = calcular_probabilidad_condicional(v, V_t0)
6 probabilidades_v_t1 = calcular_probabilidad_condicional(v, V_t1)
7
8 print('Probabilidades condicionales P(p | t = 0):', probabilidades_p_t0)
9 print('Probabilidades condicionales P(p | t = 1):', probabilidades_p_t1)
10 print('Probabilidades condicionales P(v | t = 0):', probabilidades_v_t0)
11 print('Probabilidades condicionales P(v | t = 1):', probabilidades_v_t1)
```

Listing 2: Calculo Probabilidades

```
1
2 Probabilidades condicionales P(p | t = 0): tensor([0.0515, 0.0619, 0.0928, 0.1134,
3           0.1237, 0.1649, 0.1856, 0.1340, 0.0515,
4           0.0206])
5 Probabilidades condicionales P(p | t = 1): tensor([0.3448, 0.2241, 0.2069, 0.0690,
6           0.0345, 0.0172, 0.0345, 0.0172, 0.0345,
7           0.0172])
8 Probabilidades condicionales P(v | t = 0): tensor([0.0571, 0.0857, 0.1429, 0.4286,
9           0.1714, 0.0857, 0.0286, 0.0000, 0.0000,
10          0.0000])
11 Probabilidades condicionales P(v | t = 1): tensor([0.4314, 0.2941, 0.1569, 0.0588,
12           0.0392, 0.0196, 0.0000, 0.0000, 0.0000,
13           0.0000])
```

Listing 3: Salida Probabilidades

```
1
2
3 # Graficar los histogramas
4 graficar_histograma(p, probabilidades_p_t0, 0, 'Histograma de P(P | t = 0)')
5 graficar_histograma(p, probabilidades_p_t1, 1, 'Histograma de P(P | t = 1)')
6 graficar_histograma(v, probabilidades_v_t0, 0,
7           'Histograma de P(V | t = 0)', width=2.5)
8 graficar_histograma(v, probabilidades_v_t1, 1,
9           'Histograma de P(V | t = 1)', width=2.5)
```

Listing 4: Ejecucion Graficos

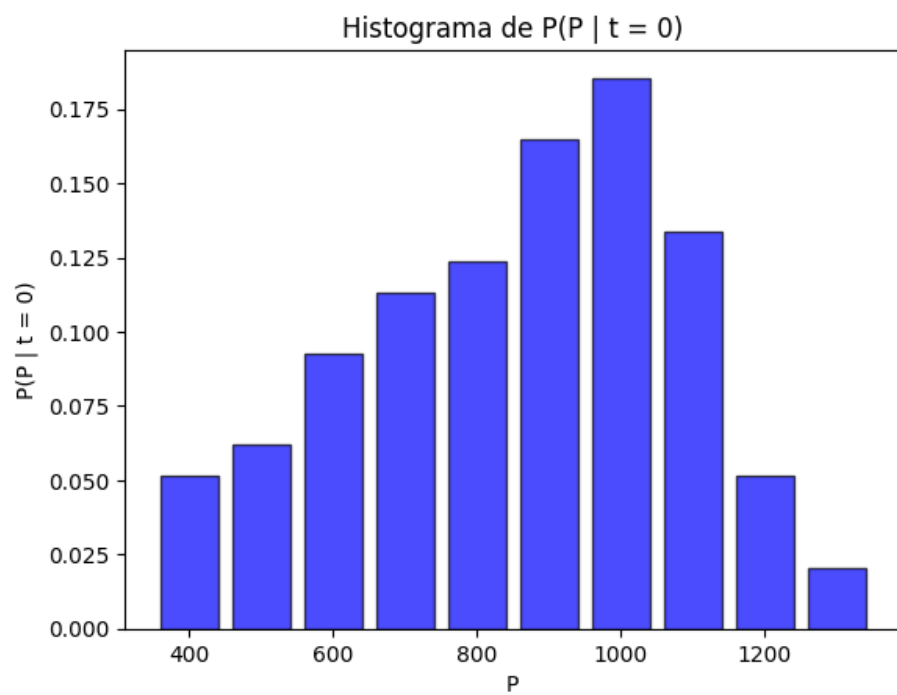


Figura 1: Histograma 1

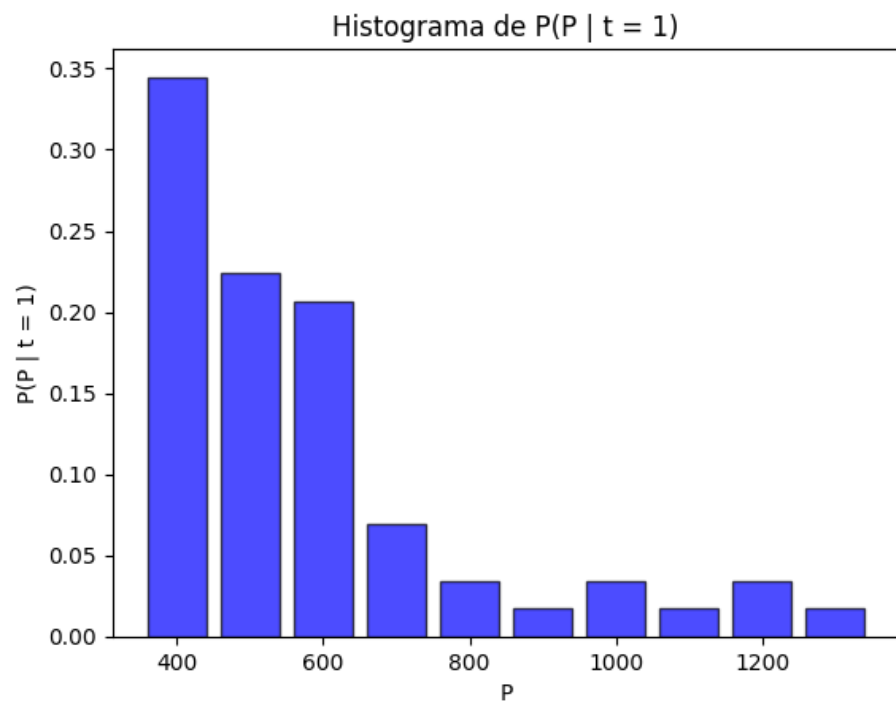


Figura 2: Histograma 2

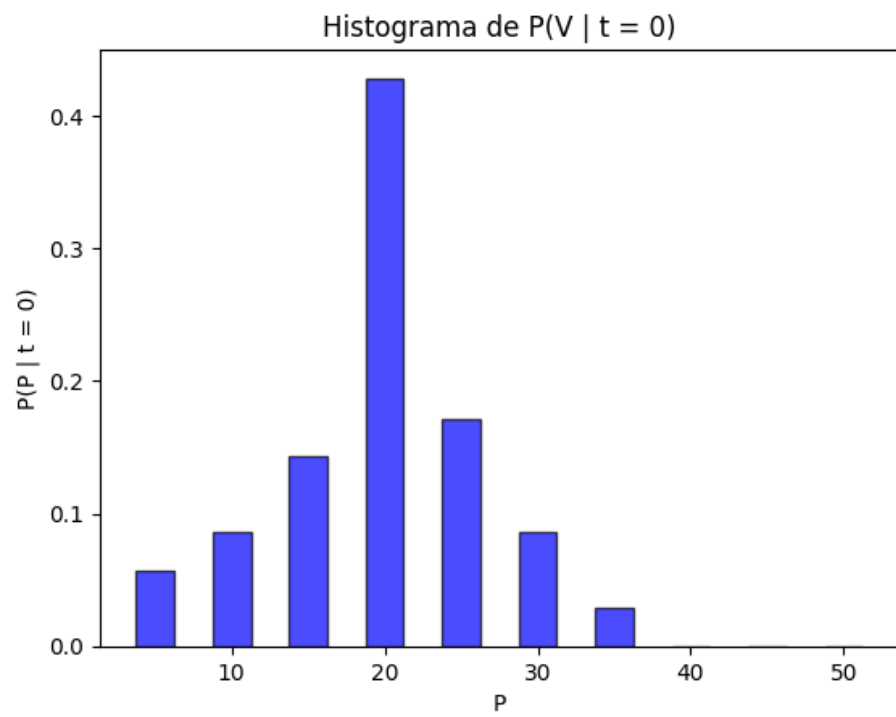


Figura 3: Histograma 3



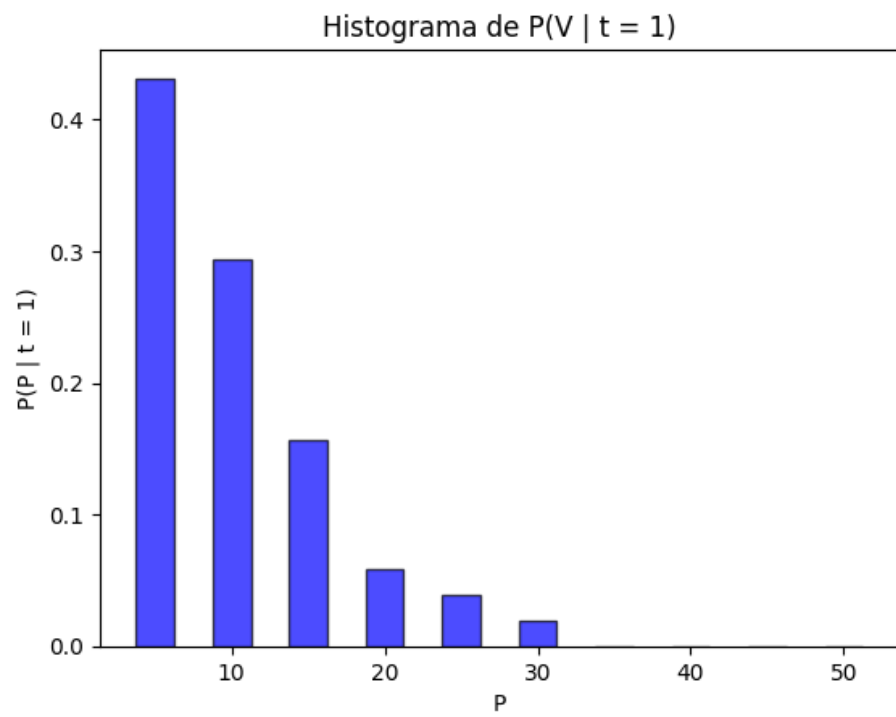


Figura 4: Histograma 4

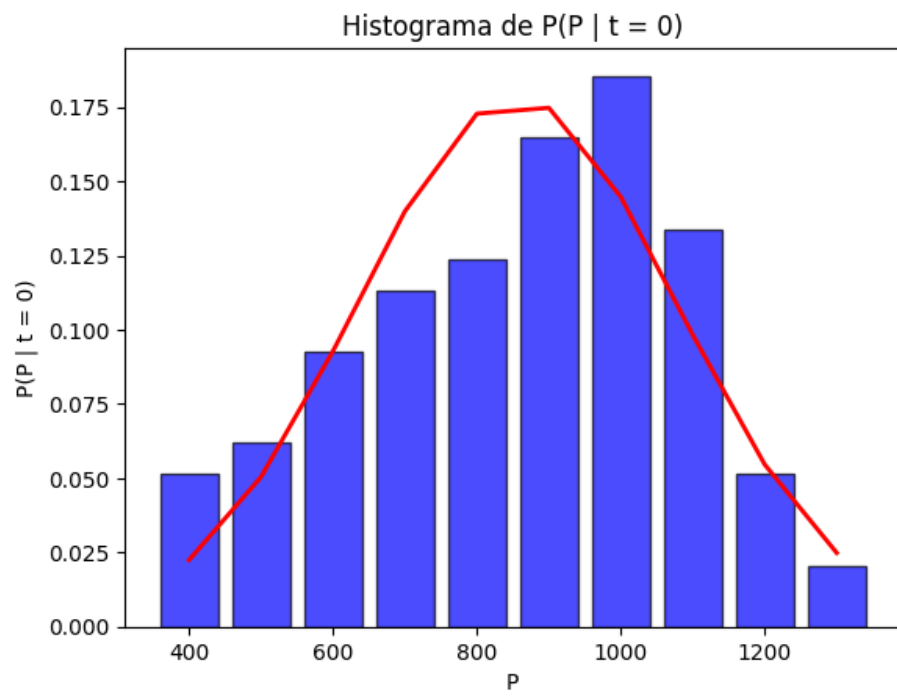


Figura 5: Histograma con Gauss 1

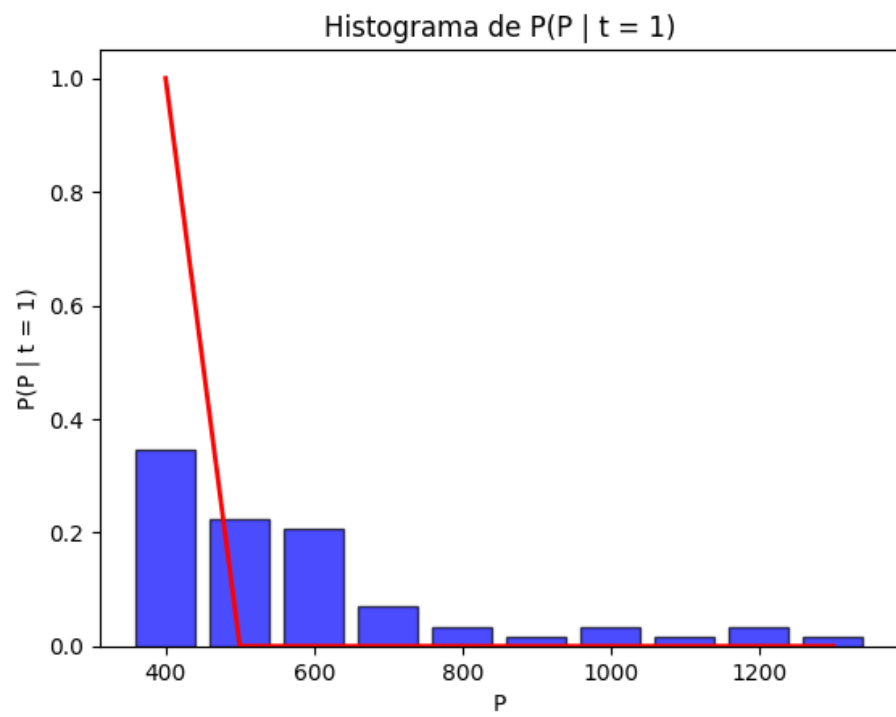


Figura 6: Histograma con Exponencial 1

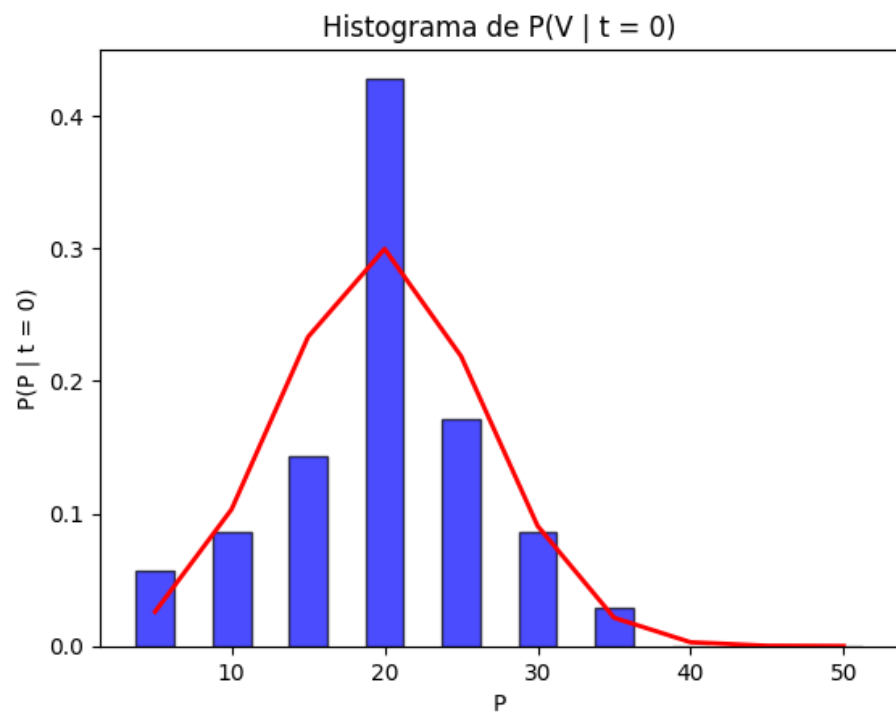


Figura 7: Histograma con Gauss 2

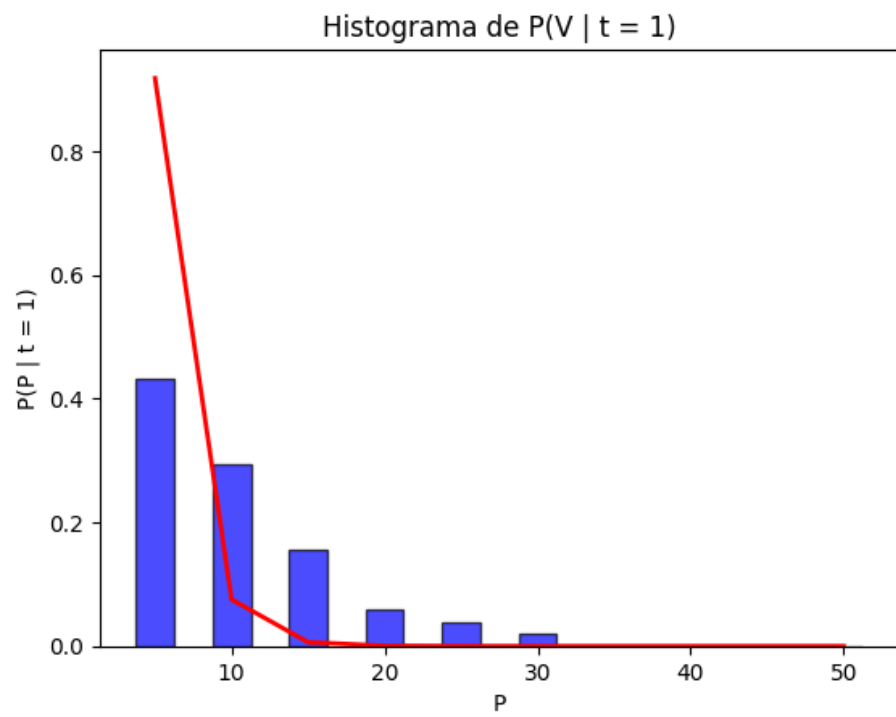


Figura 8: Histograma con Exponencial 2

## Respuesta 2)

```
1
2 # Graficar los histogramas
3 pdf_p_t0 = graficar_histograma(p, probabilidades_p_t0, 0, 'Histograma de P(P | t = 0)',
    gauss=True)
4 pdf_p_t1 = graficar_histograma(p, probabilidades_p_t1, 1, 'Histograma de P(P | t = 1)',
    exponencial=True)
5 pdf_v_t0 = graficar_histograma(v, probabilidades_v_t0, 0,
6     'Histograma de P(V | t = 0)', width=2.5, gauss=True)
7 pdf_v_t1 = graficar_histograma(v, probabilidades_v_t1, 1,
8     'Histograma de P(V | t = 1)', width=2.5, exponencial=True)
9
10 print('pdf_p_t0:', pdf_p_t0)
11 print('pdf_p_t1:', pdf_p_t1)
12 print('pdf_v_t0:', pdf_v_t0)
13 print('pdf_v_t1:', pdf_v_t1)
```

Listing 5: Calculo de Histogramas

```
1
2 pdf_p_t0: tensor([0.0224, 0.0504, 0.0928, 0.1399, 0.1728, 0.1748, 0.1448, 0.0982, 0.0546,
3     0.0248])
4 pdf_p_t1: [1.00000000e+000 1.92874985e-022 3.72007598e-044 7.17509597e-066
5     1.38389653e-087 2.66919022e-109 5.14820022e-131 9.92959040e-153
6     1.91516960e-174 3.69388307e-196]
7 pdf_v_t0: tensor([2.5916e-02, 1.0318e-01, 2.3333e-01, 2.9974e-01, 2.1873e-01, 9.0665e-02,
8     2.1348e-02, 2.8553e-03, 2.1694e-04, 9.3625e-06])
9 pdf_v_t1: [9.17915001e-01 7.53470516e-02 6.18486263e-03 5.07684440e-04
10    4.16732766e-05 3.42075085e-06 2.80792329e-07 2.30488379e-08
11    1.89196383e-09 1.55301848e-10]
```

Listing 6: Salida

En el código, se están graficando histogramas de probabilidades condicionales y ajustando curvas de distribuciones teóricas (normal y exponencial) sobre estos histogramas. Esto se hace con el fin de visualizar si los datos empíricos se asemejan a alguna distribución conocida.

Además, dependiendo de los parámetros 'gauss' y 'exponencial', la función ajusta una curva de distribución normal o exponencial, respectivamente, sobre el histograma. Esto se logra calculando los parámetros de la distribución teórica (media y desviación estándar para la normal, y el parámetro lambda para la exponencial) a partir de los datos empíricos.

Para la distribución normal, se calcula la media 'mu' como la suma ponderada de los valores 'p' por sus probabilidades, y la desviación estándar 'sigma' como la raíz cuadrada de la suma ponderada de los cuadrados de las desviaciones de 'p' respecto a 'mu'. Luego, se evalúa la función de densidad de probabilidad (pdf) de la distribución normal con estos parámetros y se grafica sobre el histograma.

Para la distribución exponencial, se calcula el parámetro 'lambda\_param' y se evalúa la pdf de la distribución exponencial en un rango de valores 'x' que cubre los datos empíricos. Luego, se normaliza la pdf para que su suma sea 1 (propiedad de una función de densidad de probabilidad válida) y se grafica sobre el histograma.

Al ajustar estas curvas teóricas sobre los histogramas empíricos, se puede visualizar si los datos se asemejan a alguna de estas distribuciones conocidas.

**Respuesta 3)** Para estimar si habrá un corte eléctrico usando el teorema de Bayes con las entradas  $m1 = 500$  y  $m2 = 10$ , seguimos los siguientes pasos: Primero, definimos las variables y los datos:  $P$  representa la precipitación,  $V$  representa la velocidad del viento, y  $t$  es la variable que indica si hay corte eléctrico ( $t = 1$ ) o no ( $t = 0$ ). Los datos de las ocurrencias de  $P$  y  $V$  dado  $t$  están en los tensores  $P\_t0$ ,  $P\_t1$ ,  $V\_t0$ , y  $V\_t1$ . Luego, calculamos las probabilidades condicionales  $P(P|t = 0)$  y  $P(P|t = 1)$  para la precipitación, y  $P(V|t = 0)$  y  $P(V|t = 1)$  para la velocidad del viento. Aplicamos el teorema de Bayes para calcular  $P(t = 1|P = m1, V = m2)$  usando la fórmula:

$$P(t = 1|P = m1, V = m2) = \frac{P(P = m1|t = 1) \cdot P(V = m2|t = 1) \cdot P(t = 1)}{P(P = m1) \cdot P(V = m2)}$$

y de manera similar para  $P(t = 0|P = m1, V = m2)$ . Finalmente, comparamos las probabilidades  $P(t = 1|P = m1, V = m2)$  y  $P(t = 0|P = m1, V = m2)$  para determinar si es más probable que haya un corte eléctrico. Este enfoque permite estimar la probabilidad de un corte eléctrico dado los valores de precipitación y velocidad del viento utilizando el teorema de Bayes.

```

1
2
3 probabilidades_p_t0 = calcular_probabilidad_condicional(p, P_t0)
4 probabilidades_p_t1 = calcular_probabilidad_condicional(p, P_t1)
5 probabilidades_v_t0 = calcular_probabilidad_condicional(v, V_t0)
6 probabilidades_v_t1 = calcular_probabilidad_condicional(v, V_t1)
7
8 # Definir las entradas
9 m1 = 500
10 m2 = 10
11
12 # Encontrar los ndices correspondientes a m1 y m2
13 index_p = (p == m1).nonzero(as_tuple=True)[0].item()
14 index_v = (v == m2).nonzero(as_tuple=True)[0].item()
15
16 # Calcular las probabilidades a priori
17 P_t1_prior = 0.5 # Asumimos que P(t=1) = P(t=0) = 0.5
18 P_t0_prior = 0.5
19
20 # Calcular las probabilidades condicionales
21 P_m1_given_t1 = probabilidades_p_t1[index_p].item()
22 P_m2_given_t1 = probabilidades_v_t1[index_v].item()
23 P_m1_given_t0 = probabilidades_p_t0[index_p].item()
24 P_m2_given_t0 = probabilidades_v_t0[index_v].item()
25
26 # Calcular las probabilidades marginales
27 P_m1 = P_m1_given_t1 * P_t1_prior + P_m1_given_t0 * P_t0_prior
28 P_m2 = P_m2_given_t1 * P_t1_prior + P_m2_given_t0 * P_t0_prior
29
30 # Aplicar el teorema de Bayes
31 P_t1_given_m1_m2 = (P_m1_given_t1 * P_m2_given_t1 * P_t1_prior) / (P_m1 * P_m2)
32 P_t0_given_m1_m2 = (P_m1_given_t0 * P_m2_given_t0 * P_t0_prior) / (P_m1 * P_m2)
33
34 # Mostrar los resultados
35 print(f"P(t=1 | P={m1}, V={m2}) = {P_t1_given_m1_m2}")
36 print(f"P(t=0 | P={m1}, V={m2}) = {P_t0_given_m1_m2}")
37
38 # Determinar si habra corte el ctrico
39 if P_t1_given_m1_m2 > P_t0_given_m1_m2:

```

```
40     print("Es probable que haya un corte el ctrico.")
41 else:
42     print("Es probable que no haya un corte el ctrico.")
```

Listing 7: Calculo Bayes

```
1
2 P(t=1 | P=500, V=10) = 1.2137203944943804
3 P(t=0 | P=500, V=10) = 0.09761463300808174
4 Es probable que haya un corte el ctrico.
```

Listing 8: Salida Bayes