Tarea de Distribuciones

Sea la probabilidad de tener un dia lluvioso (definido como un día con precipitación por encima de 5 mm) p = 0.12 en cierta ciudad.

```python
# Librerías
import numpy as np
import matplotlib.pyplot as plt
import scipy.special as sps
import scipy.stats as stats
import warnings
```

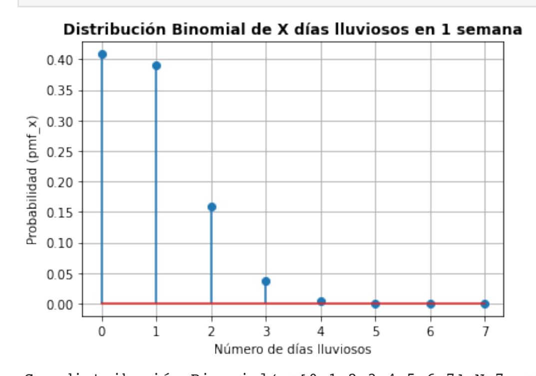
- 1.- Formula la probabilidad de tener un día lluvioso con la distribución Bernoulli.

```python
# Ejercicio 1: Distribución Bernoulli

# Función de masa de probabilidad pmf_x (Bernoulli)
def bernoulli(x, p):
    if x == 0 or x == 1:
        pmf_x = (p**x) * ((1 - p)**(1 - x))
    else:
        print("Valor de X fuera del soporte estadístico")
    return pmf_x

# Calcular probabilidad de SÍ tener un día lluvioso
p = 0.12 # Probabilidad de un día lluvioso
pmf_bernoulli = bernoulli(1,p)
print(f"\nCon distribución Bernoulli(p={p},x=1):")
print(f"P[X=1] = {pmf_bernoulli}")
```

```
Con distribución Bernoulli(p=0.12,x=1):
P[X=1] = 0.12
```

- 2.- Usa la distribución Binomial para modelar la probabilidad de tener X días lluviosos en una semana. Grafica la pmf. Además, calcula: (a) la probabilidad de tener exactamente 3 días lluviosos, (b) a lo más 3 días lluviosos, (c) al menos 2 días lluviosos.

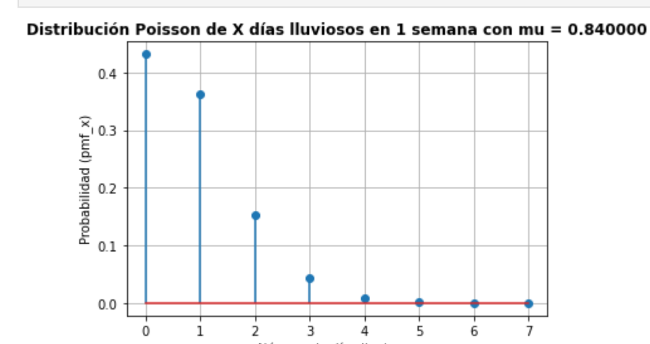
```python
# Ejercicio 2: Distribución Binomial en 1 semana

# Función de masa de probabilidad pmf_x (Binomial)
def binomial(x,N,p):
    pmf_x = sps.comb(N,x) * p**x * (1 - p)**(N - x)
    pmf_x = np.where(np.isnan(pmf_x) | np.isinf(pmf_x), 0.0, pmf_x)  # Reemplazar "nan" e "inf" por 0.0
    return pmf_x

# Arreglo de números para calcular pmf_x (Binomial) con N repeticiones
N = 7 # Número de días en 1 semana
x = np.arange(0,N+1,1)

# Calcular probabilidades para cada x en el arreglo
pmf_binomial = binomial(x,N,p)

# Graficar la pmf_x (Binomial)
plt.figure('Ejercicio 2')
plt.stem(x, pmf_binomial)
plt.xlabel('Número de días lluviosos')
plt.ylabel('Probabilidad (pmf_x)')
plt.title('Distribución Binomial de X días lluviosos en 1 semana', fontweight="bold")
plt.grid(True)
plt.show()

# Calcular la función acumulada de probabilidad cdf_x
cdf_x_binomial = np.cumsum(pmf_binomial)

# a) Probabilidad de tener exactamente 3 días lluviosos
pmf_3 = binomial(3,N,p)
print(f"\nCon distribución Binomial(x={x},N={N},p={p})")
print(f"P[X=3] = {pmf_3}")

# b) Probabilidad de tener a lo mucho 3 días lluviosos (x≤3)
print(f"P[X≤3] = {cdf_x_binomial[3]}")

# c) Probabilidad de tener al menos 2 días lluviosos (x≥2)
print(f"P[X≥2] = {1 - cdf_x_binomial[1]}")
```



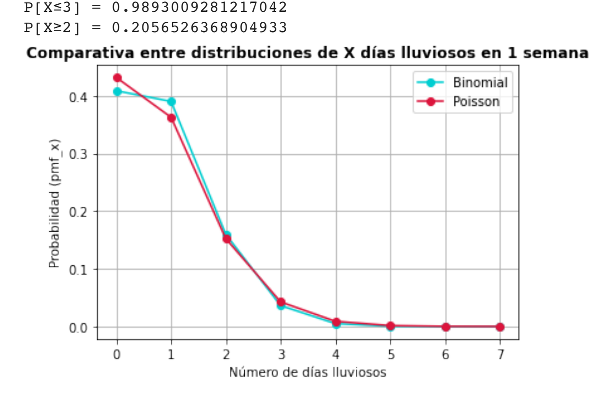

```
Con distribución Binomial(x=[0 1 2 3 4 5 6 7],N=7,p=0.12)
P[X=3] = 0.0362695753728
P[X≤3] = 0.9946307362816
P[X≥2] = 0.2012249707315199
```

- 3.- Intenta modelar la variable aleatoria del inciso anterior pero usando la distribución Poisson como aproximación. Grafica la pmf y calcula las probabilidades del inciso anterior. Compara la similitud (o diferencia) entre resultados y gráficas.

```python
# Ejercicio 3: Distribución Poisson en 1 semana

# Función de masa de probabilidad pmf_x (Poisson)
def poisson(x,mu):
    pmf_x = (mu**x * np.exp(- mu)) / sps.factorial(x)
    pmf_x = np.where(np.isnan(pmf_x) | np.isinf(pmf_x), 0.0, pmf_x)  # Reemplazar "nan" e "inf" por 0.0
    return pmf_x

# Calcular probabilidades para cada x en el arreglo y mostrar gráfico descriptivo
mu = N*p # E[X] de la Binomial
pmf_poisson = poisson(x,mu)

# Graficar la pmf_x (Poisson)
plt.figure('Ejercicio 3')
plt.stem(x, pmf_poisson, )
plt.xlabel('Número de días lluviosos')
plt.ylabel('Probabilidad (pmf_x)')
plt.title(f'Distribución Poisson de X días lluviosos en 1 semana con mu = %f' %mu, fontweight="bold")
plt.grid(True)
plt.show()

# Calcular la función acumulada de probabilidad cdf_x
cdf_x_poisson = np.cumsum(pmf_poisson)

# a) Probabilidad de tener exactamente 3 días lluviosos
pmf_3 = poisson(3,mu)
print(f"\nCon distribución Poisson(x={x},mu={mu})")
print(f"P[X=3] = {pmf_3}")

# b) Probabilidad de tener a lo mucho 3 días lluviosos (x≤3)
print(f"P[X≤3] = {cdf_x_poisson[3]}")

# c) Probabilidad de tener al menos 2 días lluviosos (x≥2)
print(f"P[X≥2] = {1 - cdf_x_poisson[1]}")

# Comparar Binomial vs Poisson
plt.figure('Ejercicio 3.2')
plt.plot(x,pmf_binomial, label = 'Binomial', marker='o', color="darkturquoise")
plt.plot(x,pmf_poisson, label = 'Poisson', marker='o', color='crimson')
plt.xlabel('Número de días lluviosos')
plt.ylabel('Probabilidad (pmf_x)')
plt.title("Comparativa entre distribuciones de X días lluviosos en 1 semana", fontweight="bold")
plt.grid(True)
plt.legend()
plt.show()
```

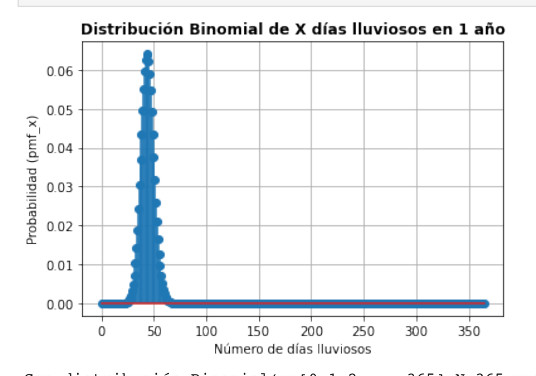



```
Con distribución Poisson(x=[0 1 2 3 4 5 6 7],mu=0.84)
P[X=3] = 0.0426460923464182
P[X≤3] = 0.9893009281217042
P[X≥2] = 0.2056526368904933
```



- 4.- Ahora considera X la variable aleatoria que mide el número de días lluviosos en un añoo (no biciesto). Grafica la pmf. Contesta: (a) la probabilidad de tener exactamente 100 días lluviosos, (b) la probabilidad de tener a lo más 200 días lluviosos, (c) la probabilidad de tener al menos 50 días lluviosos.

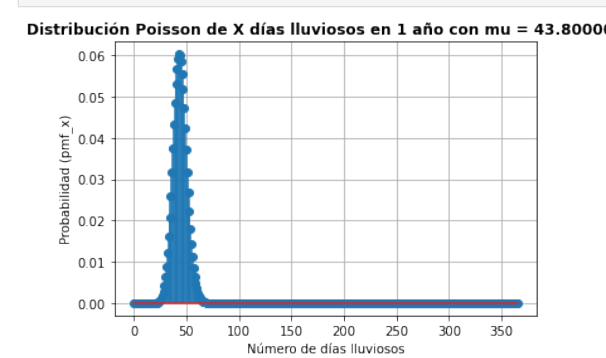
```python
# Ejercicio 4: Distribución Binomial en 1 año

# Arreglo de números para calcular pmf_x (Binomial) con N repeticiones
N = 365 # Número de días en 1 semana
x = np.arange(0,N+1,1)

# Calcular probabilidades para cada x en el arreglo
pmf_binomial_y = binomial(x,N,p)

# Graficar la pmf_x (Binomial)
plt.figure('Ejercicio 4')
plt.stem(x, pmf_binomial_y)
plt.xlabel('Número de días lluviosos')
plt.ylabel('Probabilidad (pmf_x)')
plt.title('Distribución Binomial de X días lluviosos en 1 año', fontweight="bold")
plt.grid(True)
plt.show()

# Calcular la función acumulada de probabilidad cdf_x
cdf_x_binomial_y = np.cumsum(pmf_binomial_y)

# a) Probabilidad de tener exactamente 100 días lluviosos
pmf_100 = binomial(100,N,p)
print(f"\nCon distribución Binomial(x=[0,1,2,...,365],N={N},p={p})")
print(f"P[X=100] = {pmf_100}")

# b) Probabilidad de tener a lo mucho 200 días lluviosos (x≤200)
print(f"P[X≤200] = {cdf_x_binomial_y[200]}")

# c) Probabilidad de tener al menos 50 días lluviosos (x≥50)
print(f"P[X≥50] = {1 - cdf_x_binomial_y[49]}")
```



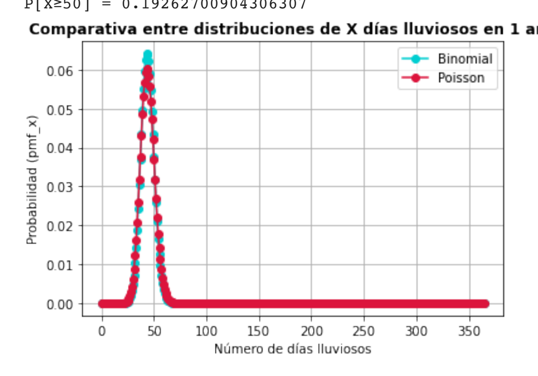

```
Con distribución Binomial(x=[0,1,2,...,365],N=365,p=0.12)
P[X=100] = 8.970277772784314e-16
P[X≤200] = 0.9999999999999669
P[X≥50] = 0.17833621000977506
```

- 5.- Repite el inciso anterior pero usando la aproximación Poisson. Genera la gráfica, calcula las probabilidades, y comenta en la similitud (o diferencia) entre los resultados.

```python
# Ejercicio 5: Distribución Poisson en 1 año
warnings.filterwarnings("ignore")

# Calcular probabilidades para cada x en el arreglo
mu = N*p # E[X] de la Binomial
pmf_poisson_y = poisson(x,mu)

# Graficar la pmf_x (Poisson)
plt.figure('Ejercicio 5')
plt.stem(x, pmf_poisson_y)
plt.xlabel('Número de días lluviosos')
plt.ylabel('Probabilidad (pmf_x)')
plt.title(f'Distribución Poisson de X días lluviosos en 1 año con mu = %f' %mu, fontweight="bold")
plt.grid(True)
plt.show()

# Calcular la función acumulada de probabilidad cdf_x
cdf_x_poisson_y = np.cumsum(pmf_poisson_y)

# a) Probabilidad de tener exactamente 100 días lluviosos
pmf_100 = poisson(100,mu)
print(f"\nCon distribución Poisson(x=[0,1,2,...,365],mu={mu})")
print(f"P[X=100] = {pmf_100}")

# b) Probabilidad de tener a lo mucho 200 días lluviosos (x≤200)
print(f"P[X≤200] = {cdf_x_poisson_y[200]}")

# c) Probabilidad de tener al menos 50 días lluviosos (x≥50)
print(f"P[X≥50] = {1 - cdf_x_poisson_y[49]}")

# Comparar Binomial vs Poisson
plt.figure('Ejercicio 5.2')
plt.plot(x,pmf_binomial_y, label = 'Binomial', marker='o', color='darkturquoise')
plt.plot(x,pmf_poisson_y, label = 'Poisson', marker='o', color='crimson')
plt.xlabel('Número de días lluviosos')
plt.ylabel('Probabilidad (pmf_x)')
plt.title("Comparativa entre distribuciones de X días lluviosos en 1 año", fontweight="bold")
plt.grid(True)
plt.legend()
plt.show()
```

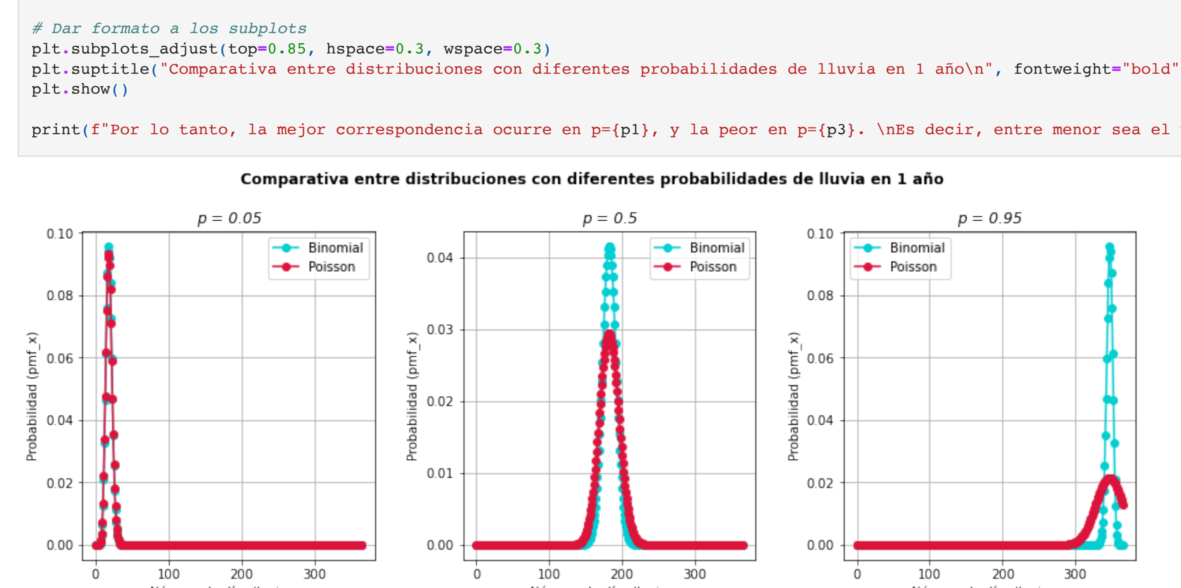



```
Con distribución Poisson(x=[0,1,2,...,365],mu=43.8)
P[X=100] = 1.4299112674311552e-13
P[X≤200] = 0.9999999999999997
P[X≥50] = 0.19262700904306307
```



- 6.- Grafica las pmfs para X el número de días lluviosos en 1 añoo cuando p = 0.05 y p = 0.5 y p = 0.95 usando la binomial y la Poisson. Comenta cuando es mejor la correspondencia.

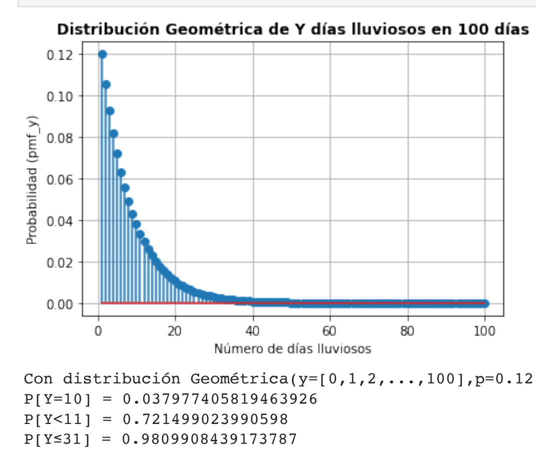
```python
# Ejercicio 6: Distribuciónes con diferentes valores de probabilidad en 1 año

# Definir nuestras probabiliadades y calcular cada pmf Binomial con ellas
p1, p2, p3 = 0.05, 0.5, 0.95
pmf_binomial_p1 = binomial(x,N,p1)
pmf_binomial_p2 = binomial(x,N,p2)
pmf_binomial_p3 = binomial(x,N,p3)

# Definir mu para las pmf Binomial, y calcular cada pmf Poisson con estas
mu1, mu2, mu3 = N*p1, N*p2, N*p3 # E[X] de cada distribución binomial
pmf_poisson_p1 = stats.poisson.pmf(x,mu1)
pmf_poisson_p2 = stats.poisson.pmf(x,mu2)
pmf_poisson_p3 = stats.poisson.pmf(x,mu3)

# Crear una figura con tres subplots
fig, axs = plt.subplots(1, 3, figsize=(15, 5))
p_values = [0.05, 0.5, 0.95]
pmfs_binomial = [pmf_binomial_p1, pmf_binomial_p2, pmf_binomial_p3]
pmfs_poisson = [pmf_poisson_p1, pmf_poisson_p2, pmf_poisson_p3]

# Ciclo para graficar las pmf_x y comparar Binomial vs Poisson con cada valor de p
for i in range(3):
    axs[i].plot(x, pmfs_binomial[i], label='Binomial', marker='o', color='darkturquoise')
    axs[i].plot(x, pmfs_poisson[i], label='Poisson', marker='o', color='crimson')
    axs[i].set_xlabel('Número de días lluviosos')
    axs[i].set_ylabel('Probabilidad (pmf_x)')
    axs[i].set_title(f"p = {p_values[i]}", fontstyle="italic")
    axs[i].grid(True)
    axs[i].legend()

# Dar formato a los subplots
plt.subplots_adjust(top=0.85, hspace=0.3, wspace=0.3)
plt.suptitle("Comparativa entre distribuciones con diferentes probabilidades de lluvia en 1 año\n", fontweight="bold")
plt.show()

print(f"Por lo tanto, la mejor correspondencia ocurre en p={p1}, y la peor en p={p3}. \nEs decir, entre menor sea el valor de p, mejor es la correspondencia.")
```




```
Por lo tanto, la mejor correspondencia ocurre en p=0.05, y la peor en p=0.95.
Es decir, entre menor sea el valor de p, mejor es la correspondencia.
```

- 7.- Ahora considera la variable aleatoria Y : primer día lluvioso a partir de cierto instante. Por ejemplo y = 3 significa que el tuvimos 2 días secos y el 3 fue el primero lluvioso. ¿Qué distribución describe esta variable aleatoria? Escribe su pmf, y grafícala. Calcula: (a) la probabilidad de que el primer dia lluvioso sea el 10, (b) la probabilidad de que el primer día lluvioso no sea igual o mayor al 11, (c) la probabilidad de tener que esperar a lo más 31 días (inclusivo) para tener un día lluvioso.

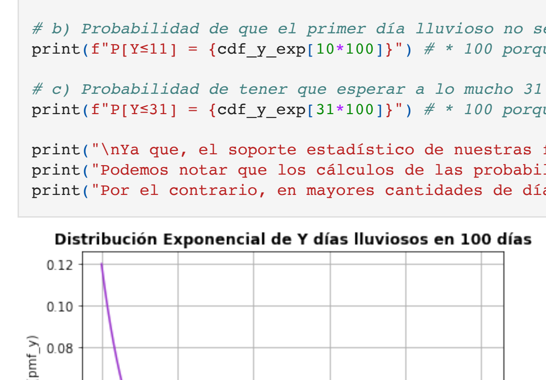
```python
# Ejercicio 7: Distribución Geométrica

# Función de masa de probabilidad pmf_y (Geométrica)
def geometric(y, p):
    pmf_y = p * ((1 - p)**(y - 1))
    return pmf_y

# Arreglo de números para calcular pmf_y (Geométrica) con N repeticiones
N = 100 # Número arbitrario de días lluviosos
y = np.arange(1,N+1,1)

# Calcular probabilidades para cada y en el arreglo
pmf_geometric = geometric(y,p) # Recordando que p=0.12

# Graficar la pmf_x (Geométrica)
plt.figure('Ejercicio 7')
plt.stem(y, pmf_geometric)
plt.xlabel('Número de días lluviosos')
plt.ylabel('Probabilidad (pmf_y)')
plt.title('Distribución Geométrica de Y días lluviosos en 100 días', fontweight="bold")
plt.grid(True)
plt.show()

# Calcular la función acumulada de probabilidad cdf_y
cdf_y_geometric = np.cumsum(pmf_geometric)

# a) Probabilidad de que el primer día lluvioso sea el 10
pmf_10 = geometric(10,p)
print(f"\nCon distribución Geométrica(y=[0,1,2,...,100],p={p})")
print(f"P[Y=10] = {pmf_10}")

# b) Probabilidad de que el primer día lluvioso no sea igual o mayor al 11: [1 - (y≥11)] = (y<11) = y(≤10)
print(f"P[Y<11] = {cdf_y_geometric[10-1]}") # - 1 porque nuestro arreglo no empieza en 0

# c) Probabilidad de tener que esperar a lo mucho 31 días (inclusivo) para tener un día lluvioso p(≤31)
print(f"P[Y≤31] = {cdf_y_geometric[31-1]}") # - 1 porque nuestro arreglo no empieza en 0
```




```
Con distribución Geométrica(y=[0,1,2,...,100],p=0.12)
P[Y=10] = 0.037977405819463926
P[Y<11] = 0.721499023990598
P[Y≤31] = 0.9809908439173787
```

- 8.- Resuelve el ejercicio anterior considerando que el tiempo es continuo. Utilizarás la distribución exponencial. Calcula la pdf y grafíacala (ten cuidado al calcular la θ). Calcula las probabilidades pedidas en el inciso anterior (ojo con la primera...). Compara los resultados y comenta la similitud o diferencia entre los resultados.

```python
# Ejercicio 8: Distribución Exponencial

# Función de densidad de probabilidad pdf_y (Exponencial)
def exponential(y, theta):
    pmf_y = 1 / theta * np.exp(- y / theta)
    return pmf_y

# Función de acumulada de probabilidad cdf_y (Exponencial)
def cdf_exponential(y,theta):
    cdf_y = 1 - np.exp(- y / theta)
    return cdf_y

# Definir θ y crear un arreglo de números para calcular pdf_y (Exponencial) con N repeticiones
mu = p # E[Y] de la Binomial con N=1
theta = 1 / mu  # Relación Binomial - Exponencial
y = np.arange(0,N+1,.01) # N = 100 suficiente para modelar el comportamiento en infinito

# Calcular las densidades para cada y en el arreglo
pdf_y = exponential(y,theta)

# Graficar la pdf_y (Exponencial)
plt.figure('Ejercicio 8')
plt.plot(y, pdf_y, color='darkorchid')
plt.xlabel('Número de días lluviosos')
plt.ylabel('Densidad (pmf_y)')
plt.title('Distribución Exponencial de Y días lluviosos en 100 días', fontweight="bold")
plt.grid(True)
plt.show()

# Calcular la función acumulada de probabilidad cdf_y
cdf_y_exp = cdf_exponential(y,theta)

# a) Probabilidad de que el primer día lluvioso sea el 10
print(f"\nCon distribución Exponencial(y=[0,1,2,...,100],p={p})")
print(f"P[Y=10] = {0}. Puesto que las probabilidades puntuales no existen.")

# b) Probabilidad de que el primer día lluvioso no sea igual o mayor al 11: [1 - (y≥11)] = (y<11) = y(≤11)
print(f"P[Y≤11] = {cdf_y_exp[10*100]}") # * 100 porque el paso del arreglo es 0.01

# c) Probabilidad de tener que esperar a lo mucho 31 días (inclusivo) para tener un día lluvioso p(≤31)
print(f"P[Y≤31] = {cdf_y_exp[31*100]}") # * 100 porque el paso del arreglo es 0.01

print("\nYa que, el soporte estadístico de nuestras funciones, Geométrica y Exponencial, difieren. Pues, la primera empieza en 1 y la segunda en 0.")
print("Podemos notar que los cálculos de las probabilidades en menores cantidades de días lluviosos, cercanos al límite inferior, difieren en mayor medida.")
print("Por el contrario, en mayores cantidades de días lluviosos, estas diferencias se reducen, por lo que las aproximaciones son prácticamente iguales.")
```




```
Con distribución Exponencial(y=[0,1,2,...,100],p=0.12)
P[Y=10] = 0. Puesto que las probabilidades puntuales no existen.
P[Y≤11] = 0.6988057880877978
P[Y≤31] = 0.9757660321543089

Ya que, el soporte estadístico de nuestras funciones, Geométrica y Exponencial, difieren. Pues, la primera empieza en 1 y la segunda en 0.
Podemos notar que los cálculos de las probabilidades en menores cantidades de días lluviosos, cercanos al límite inferior, difieren en mayor medida.
Por el contrario, en mayores cantidades de días lluviosos, estas diferencias se reducen, por lo que las aproximaciones son prácticamente iguales.
```