

# Universidad Politecnica Salesiana

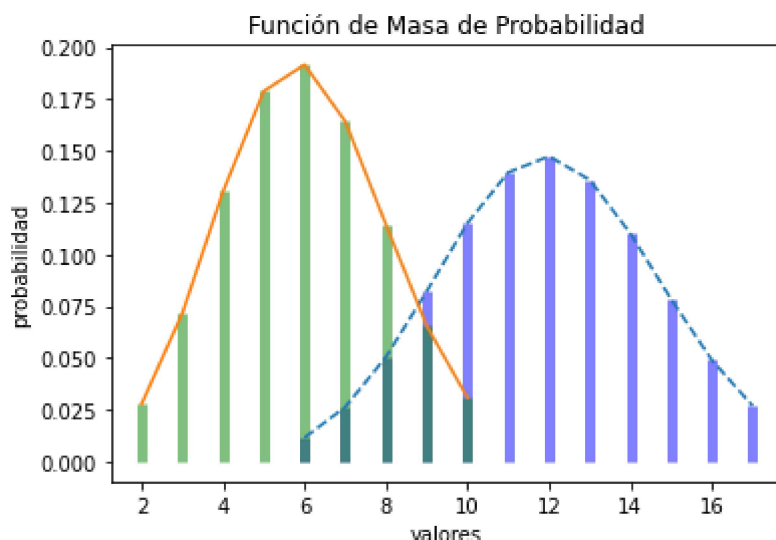
## Prueba practica 3

Realizado por: Javier Yunga

### ➤ 1. Distribucion acumulada de la binomial y la normal

```
from scipy import stats
import matplotlib.pyplot as plt
import numpy as np

n, p = 30, 0.4 # parametros de forma de la distribución binomial
n_1, p_1 = 20, 0.3 # parametros de forma de la distribución binomial
x = np.arange(stats.binom.ppf(0.01, n, p), stats.binom.ppf(0.99, n, p))
x_1 = np.arange(stats.binom.ppf(0.01, n_1, p_1), stats.binom.ppf(0.99, n_1, p_1))
fmp = stats.binom.pmf(x, n, p) # Función de Masa de Probabilidad
fmp_1 = stats.binom.pmf(x_1, n_1, p_1) # Función de Masa de Probabilidad
plt.plot(x, fmp, '--')
plt.plot(x_1, fmp_1)
plt.vlines(x, 0, fmp, colors='b', lw=5, alpha=0.5)
plt.vlines(x_1, 0, fmp_1, colors='g', lw=5, alpha=0.5)
plt.title('Función de Masa de Probabilidad')
plt.ylabel('probabilidad')
plt.xlabel('valores')
plt.show()
```



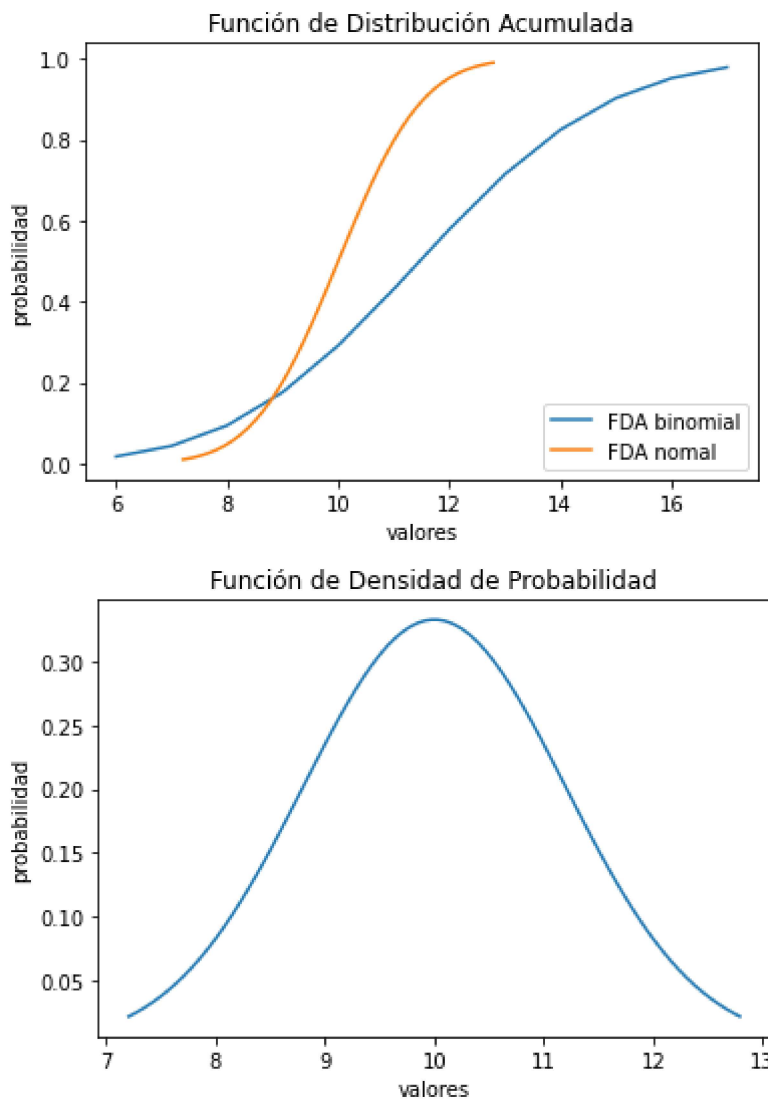
# Graficando Función de Distribución Acumulada de la normal y la binomial

```
x_1 = np.linspace(stats.norm(10, 1.2).ppf(0.01), stats.norm(10, 1.2).ppf(0.99), 100)
fda_binom = stats.binom.cdf(x, n, p) # Función de Distribución Acumulada
fda_normal = stats.norm(10, 1.2).cdf(x_1) # Función de Distribución Acumulada
plt.plot(x, fda_binom, label='FDA binomial')
```

```

plt.plot(x_1, fda_normal, label='FDA normal')
plt.title('Función de Distribución Acumulada')
plt.ylabel('probabilidad')
plt.xlabel('valores')
plt.legend(loc=4)
plt.show()
# Graficando Función de Densidad de Probabilidad con Python
FDP_normal = stats.norm(10, 1.2).pdf(x_1) # FDP
plt.plot(x_1, FDP_normal, label='FDP normal')
plt.title('Función de Densidad de Probabilidad')
plt.ylabel('probabilidad')
plt.xlabel('valores')
plt.show()

```



## ▼ 2. Distribución Uniforme acumulativa

```

from scipy.stats import uniform
import matplotlib.pyplot as plt
import numpy as np

```

```

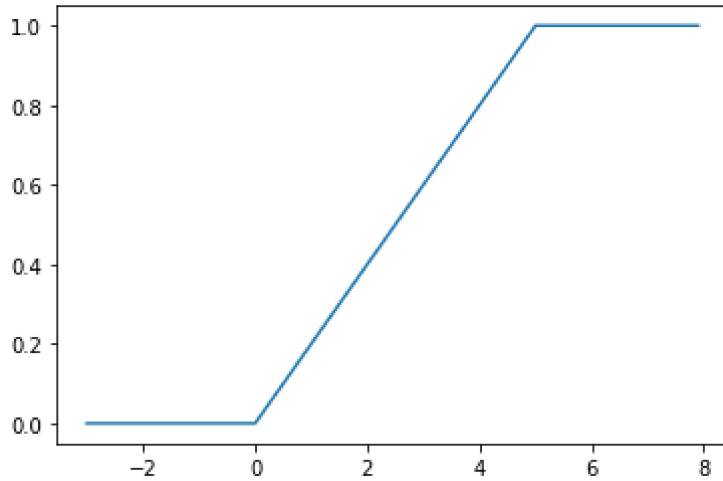
#creando una matriz de valores entre

```

```
#-3 a 8 con una diferencia de 0.1
x = np.arange(-3, 8, 0.1)

y = uniform.cdf(x, 0, 5)# aqui aplicamos la distribucion acumulativa

plt.plot(x, y)
plt.show()
```

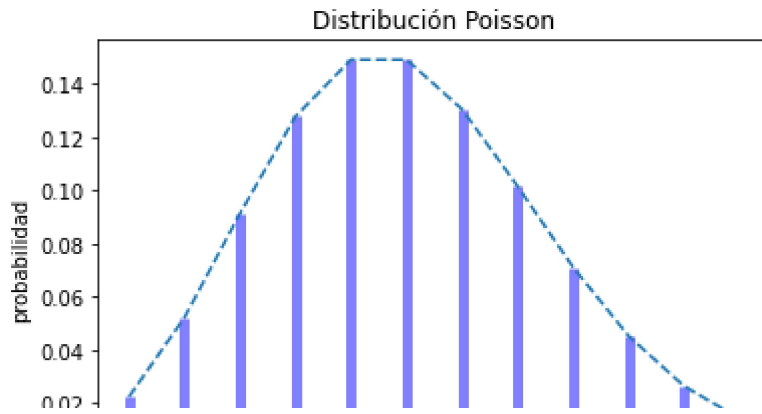


### ▼ 3. Distribucion de Poisson acumulativa

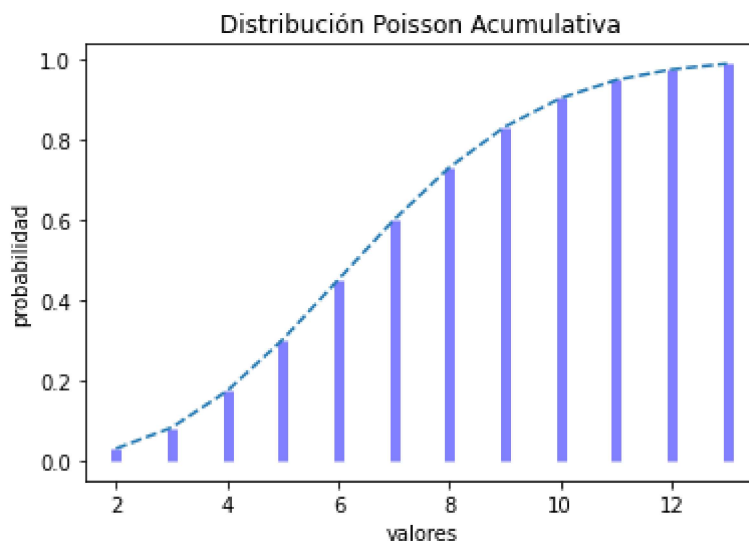
```
from scipy.stats import poisson
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import numpy

# Graficando Poisson
mu = 7 # parametro de forma
poisson = stats.poisson(mu) # Distribución
x = np.arange(poisson.ppf(0.01),
              poisson.ppf(0.99))
fmp = poisson.pmf(x) # Función de Masa de Probabilidad
plt.plot(x, fmp, '--')
plt.vlines(x, 0, fmp, colors='b', lw=5, alpha=0.5)
plt.title('Distribución Poisson')
plt.ylabel('probabilidad')
plt.xlabel('valores')
plt.show()
```





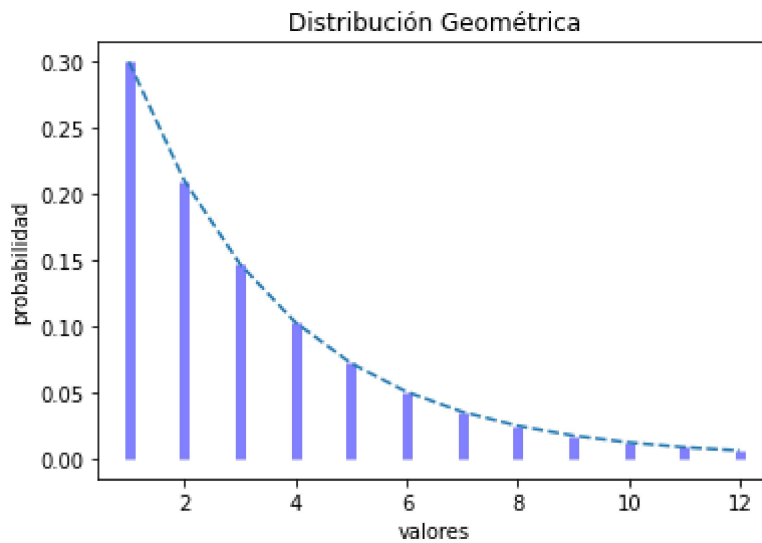
```
# Graficando Poisson acumulativa
mu = 7 # parametro de forma
poisson = stats.poisson(mu) # Distribución
x = np.arange(poisson.ppf(0.01),
              poisson.ppf(0.99))
fmp = poisson.cdf(x)
plt.plot(x, fmp, '--')
plt.vlines(x, 0, fmp, colors='b', lw=5, alpha=0.5)
plt.title('Distribución Poisson Acumulativa')
plt.ylabel('probabilidad')
plt.xlabel('valores')
plt.show()
```



## ▼ 4. Distribución Geométrica acumulativa

```
# Graficando Geométrica
p = 0.3 # parametro de forma
geometrica = stats.geom(p) # Distribución
x = np.arange(geometrica.ppf(0.01),
              geometrica.ppf(0.99))
fmp = geometrica.pmf(x) # Función de Masa de Probabilidad
plt.plot(x, fmp, '--')
plt.vlines(x, 0, fmp, colors='b', lw=5, alpha=0.5)
plt.title('Distribución Geométrica')
```

```
plt.ylabel('probabilidad')
plt.xlabel('valores')
plt.show()
```



```
# Graficando Geométrica
p = 0.3 # parametro de forma
geometrica = stats.geom(p) # Distribución
x = np.arange(geometrica.ppf(0.01),
              geometrica.ppf(0.99))
fmp = geometrica.cdf(x)
plt.plot(x, fmp, '--')
plt.vlines(x, 0, fmp, colors='b', lw=5, alpha=0.5)
plt.title('Distribución Geométrica Acumulativa')
plt.ylabel('probabilidad')
plt.xlabel('valores')
plt.show()
```

