

Distribucion Binomial

Victor Lopez

2023-02-19

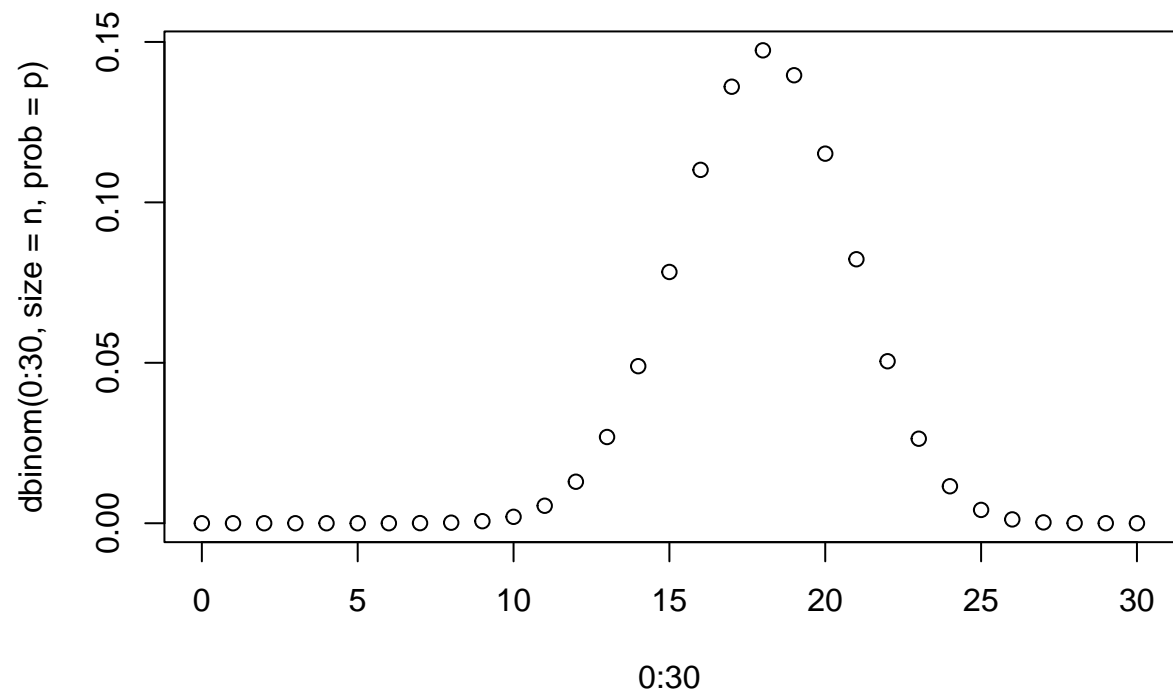
Distribucion Binomial

- En R tenemos las funciones del paquete Rlab: `dbinom(x, size, prob)`, `pbinom(x, size, prob)`, `qbinom(p, size, prob)`, `rbinom(n, size, prob)` donde `prob` es la probabilidad de éxito y `n` el número de ensayos del experimento. `size` en R = `n` en estadística
- En Python tenemos las funciones del paquete `scipy.stats.binom`: `pmf(x,n,p)`, `cdf(x,n,p)`, `ppf(q,n,p)`, `rvs(n, p, size)` donde `p` es la probabilidad de éxito y `n` el número de ensayos del experimento.

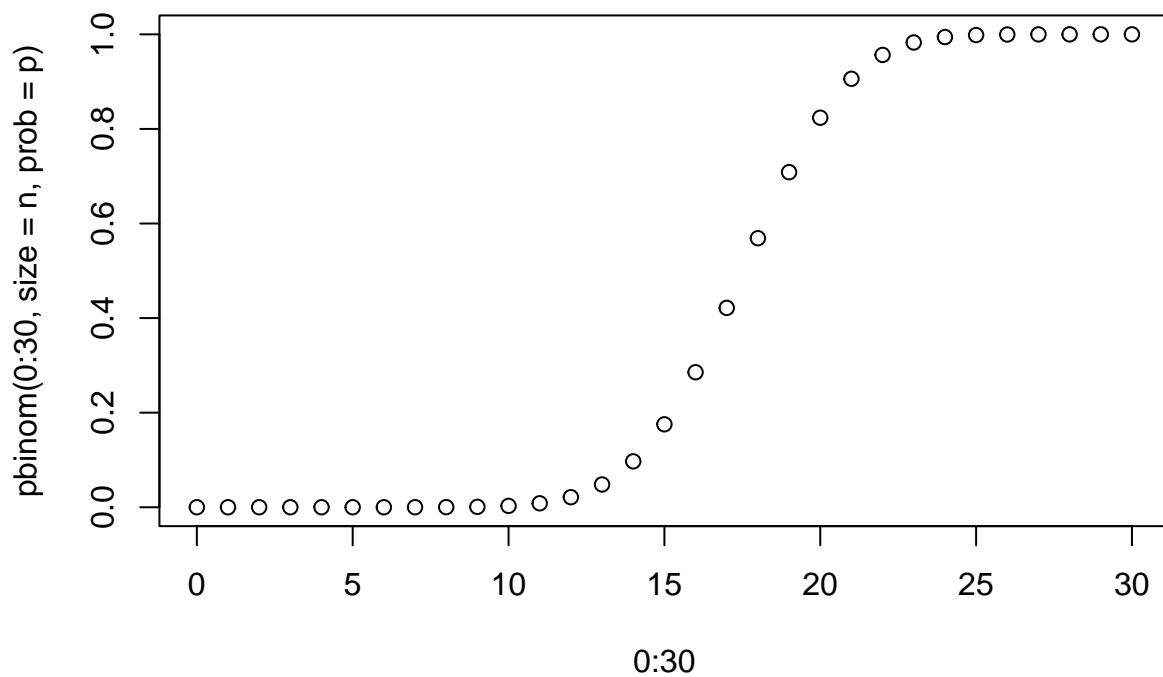
En R

```
n = 30
p = 0.6

plot(0:30, dbinom(0:30, size = n, prob = p)) # Recuerda que dbinom regresara un vector
```



```
# con la probabilidad de cada uno de los valores de n ya que en vez de pasar un solo
# valor en x, le pasamos un vector
plot(0:30, pbinom(0:30, size = n, prob = p)) # Funcion de distribucion graficada
```



```
qbinom(0.5, n, p) # El 50% de numero de exitos, esta por debajo de la mediana 18,
```

```
## [1] 18
```

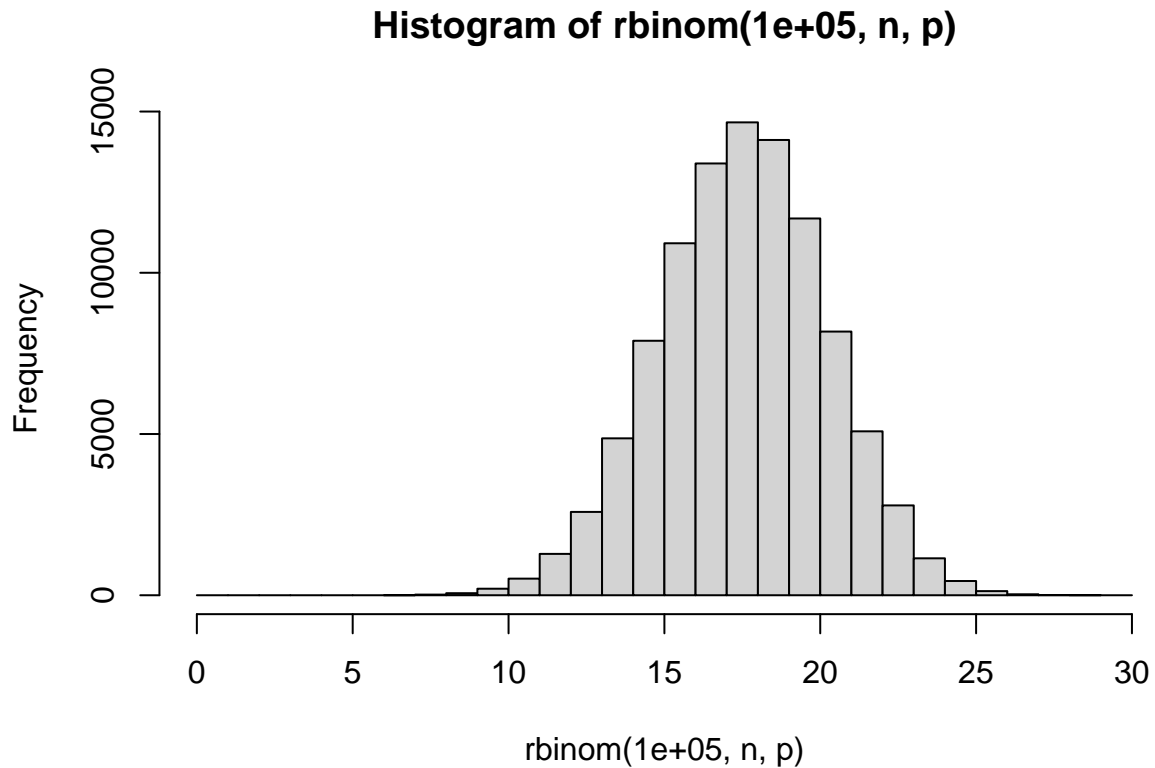
```
qbinom(0.25, n, p) # El 25% de numero de exitos, esta por debajo del 16 en este caso.
```

```
## [1] 16
```

```
# Debido a que estan muy juntos, quiere que es una distribucion muy concentrada
```

```
# Generacion de numeros random de 0 hasta n donde cada numero tiene la probabilidad de  
# salir que tiene una variable binomimial n, p
```

```
hist(rbinom(100000, n, p), breaks = 0:30)
```



En Python

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

fig, ax = plt.subplots(1,1)

n = 7
p = 0.4

mean, var, skew, kurt = binom.stats(n,p, moments = "mvsk")

print(mean, var, skew, kurt)

## 2.8000000000000003 1.6800000000000002 0.15430334996209186 -0.261904761904762

x = np.arange(0, n + 1)

ax.plot(x, binom.pmf(x, n, p), "bo", ms = 8, label = "Funcion de densidad de B(7, 0.4)")
ax.vlines(x, 0, binom.pmf(x, n, p), colors = "b", lw = 4, alpha = 0.5)
```

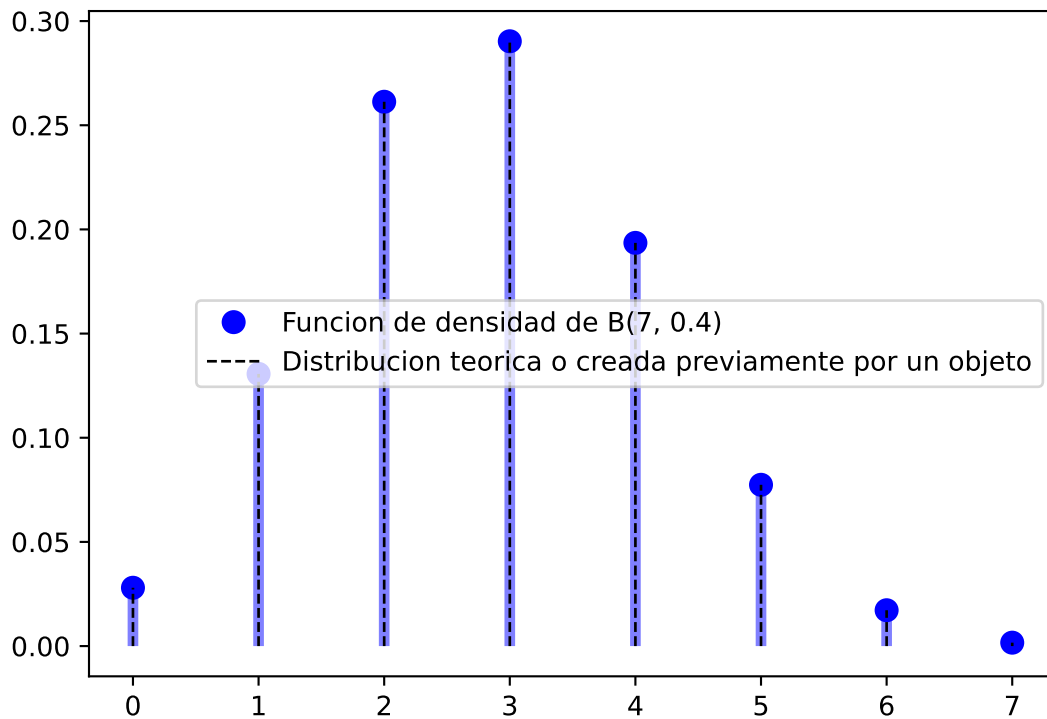
```

# Esta es otra forma equivalente de mostrarla en el histograma
# Primero declaramos el objeto, y despues llamamos a sus propiedades
rv = binom(n, p) # Creamos un objeto congelado de la Binomial o distribucion teorica
ax.vlines(x, 0, rv.pmf(x), colors = "k", linestyle = "--", lw = 1, label = "Distribucion teorica o creada previamente por un objeto")

ax.legend(loc = "best")

plt.show()

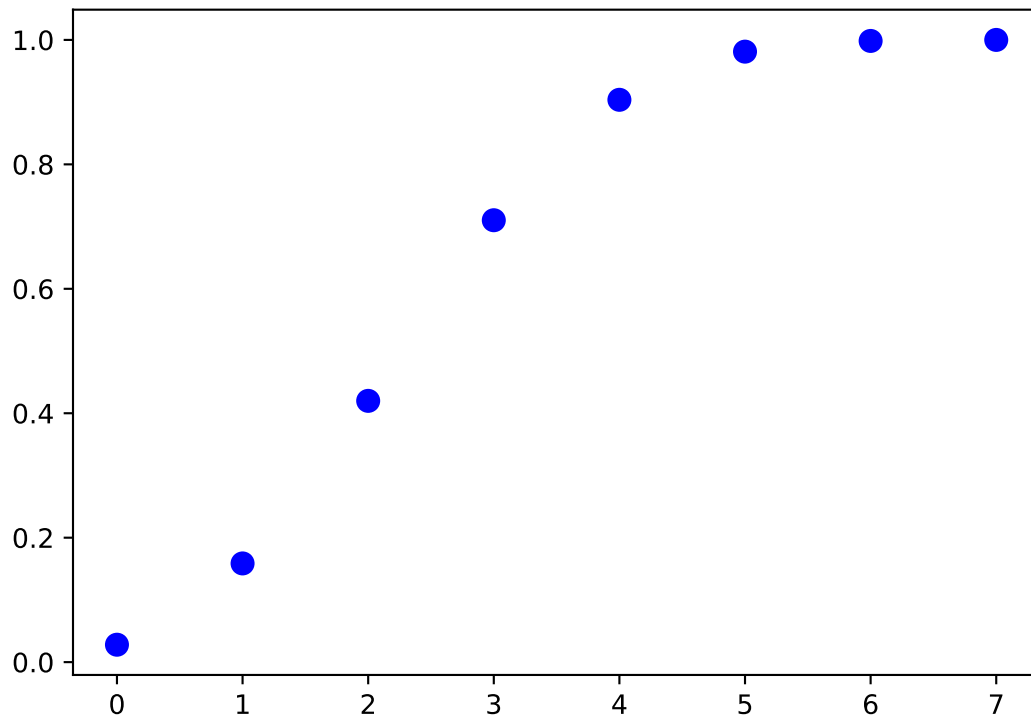
```



```

# Funcion de distribucion
fig, ax = plt.subplots(1,1)
ax.plot(x, binom.cdf(x, n, p), "bo", ms = 8, label = "Funcion de densidad de B(7, 0.4)")
plt.show()

```



```
# Histograma con valores random
```

```
fig, ax = plt.subplots(1,1)
```

```
ax.hist(binom.rvs(n, p, size = 10000), bins = 7)
```

```
## (array([ 280., 1318., 2618., 2913., 1894.,  794.,  183.]), array([0., 1., 2., 3., 4., 5., 6., 7.]),
```

```
plt.show()
```

