Binomial

Hernández Martínez Oscar Gerardo

25/6/2020

```
import os
os.environ['QT_QPA_PLATFORM_PLUGIN_PATH'] = 'C:/Users/jxsje/anaconda3/Library/plugins/platforms'
```

Función de densidad

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

Función de distribución

$$F(x) = \begin{cases} 0 & \text{si } x < 0 \\ \sum_{k=0}^{x} f(k) & \text{si } 0 \leq x < n \\ 1 & \text{si } x \geq n \end{cases}$$

Sea X

```
library(Rlab)
```

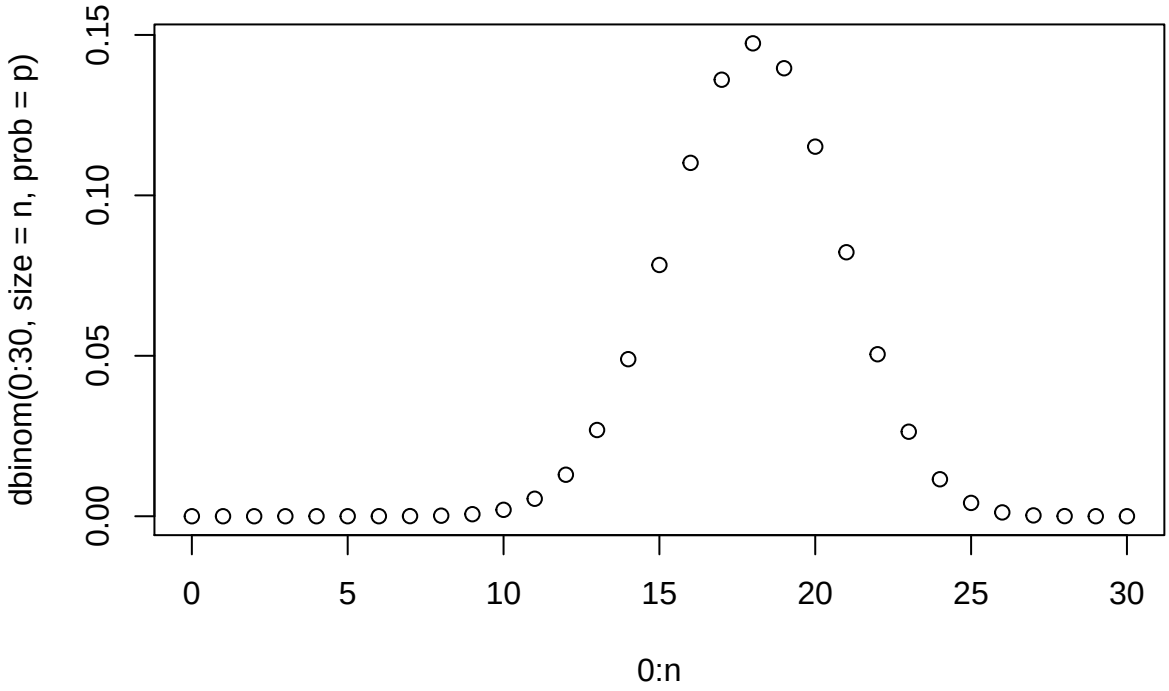
```
## Rlab 2.15.1 attached.
```

```
##
## Attaching package: 'Rlab'
```

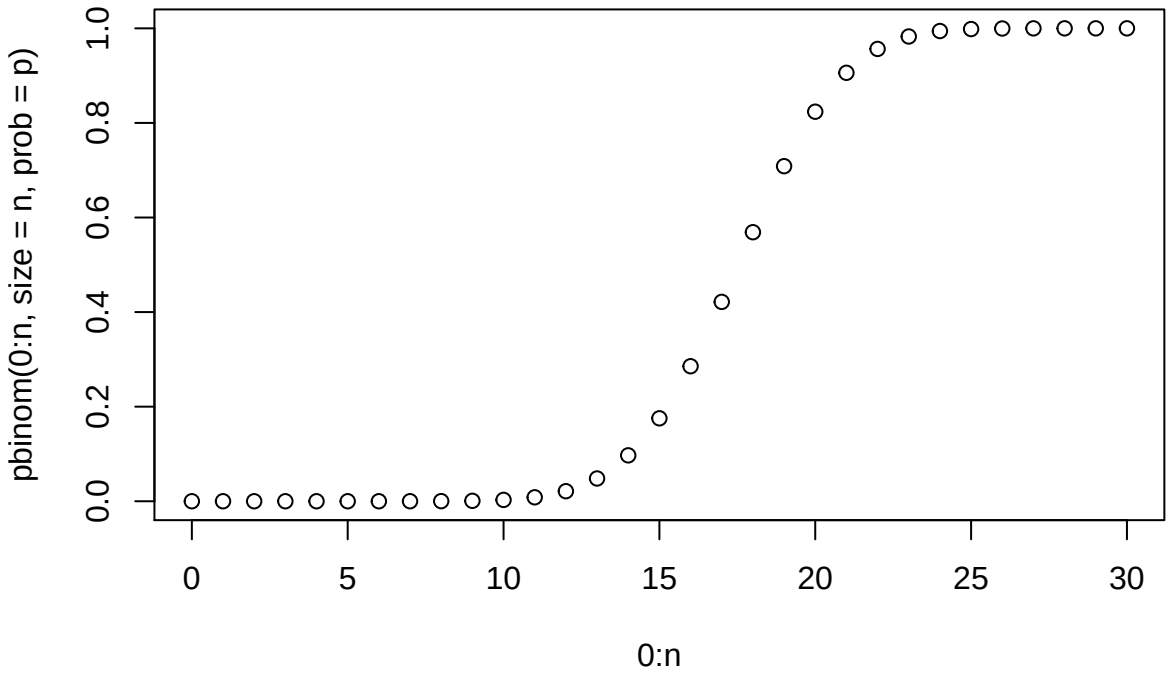
```
## The following objects are masked from 'package:stats':
##
##     dexp, dgamma, dweibull, pexp, pgamma, pweibull, qexp, qgamma,
##     qweibull, rexp, rgamma, rweibull
```

```
## The following object is masked from 'package:datasets':
##
##     precip
```

```
n = 30
p = 0.6
plot(0:n, dbinom(0:30, size = n, prob = p))
```




```
plot(0:n, pbinom(0:n, size = n, prob = p))
```



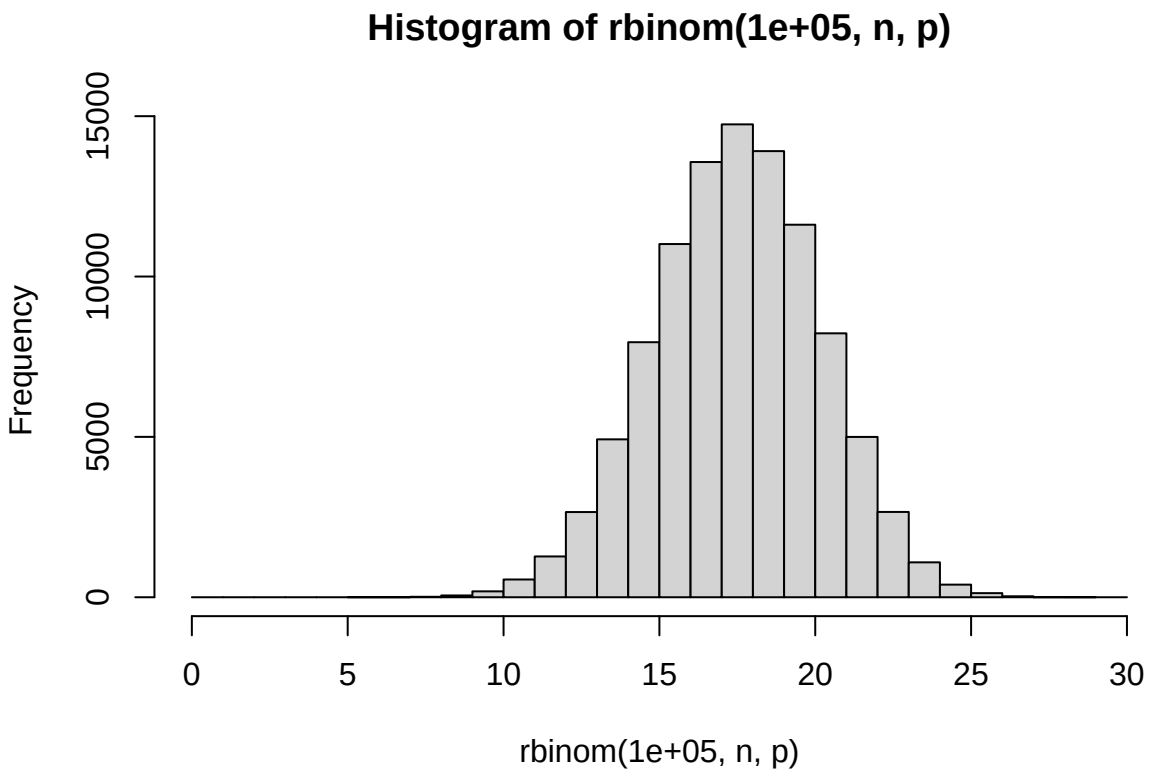

```
qbinom(0.5, n, p)
```

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

```
qbinom(0.25, n, p)
```

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

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



#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("Media %f"%mean)
```

```
## Media 2.800000
```

```
print("Varianza %f"%var)
```

```
## Varianza 1.680000
```

```
print("Sesgo %f"%skew)
```

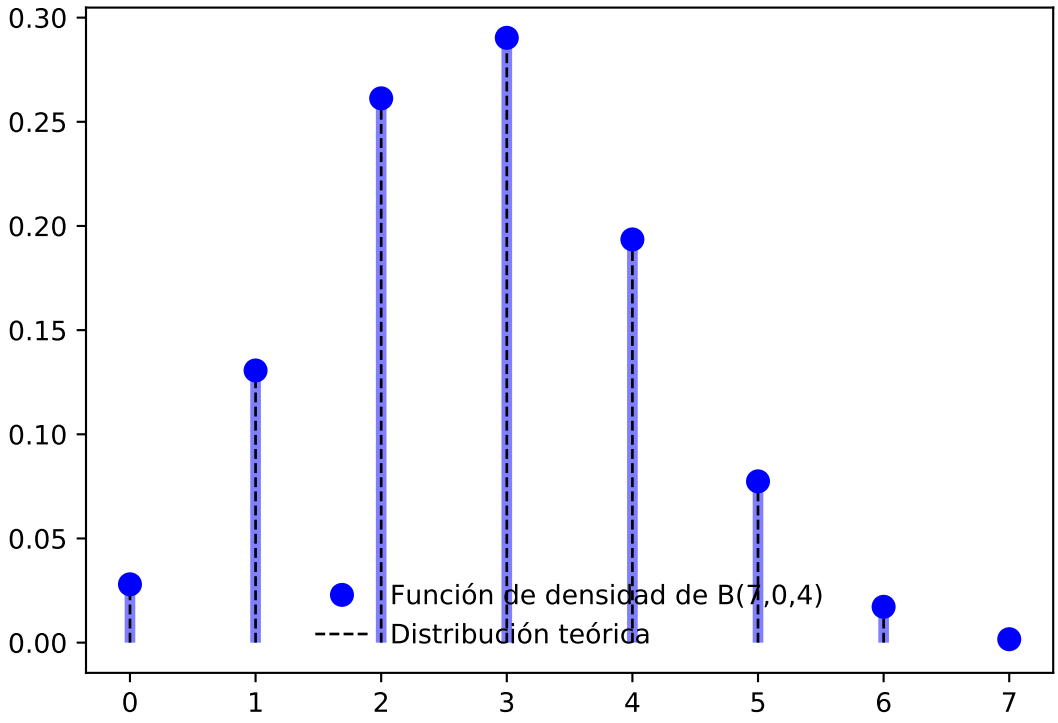
```
## Sesgo 0.154303
```

```
print("Curtosis %f"%kurt)
```

```
## Curtosis -0.261905
```

```
x = np.arange(0, n+1)
ax.plot(x, binom.pmf(x,n,p), 'bo', ms = 8, label = "Función de densidad de B(7,0,4)")
ax.vlines(x, 0, binom.pmf(x,n,p), colors = 'b', lw = 4, alpha = 0.5)
```

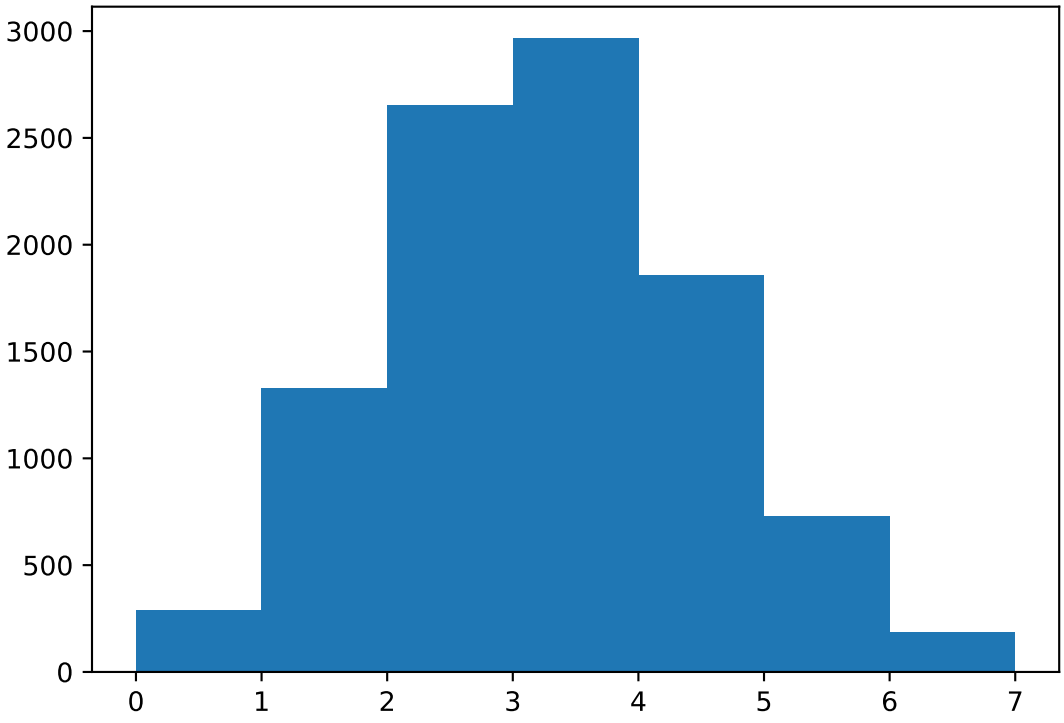
```
rv = binom(n,p)

ax.vlines(x,0,rv.pmf(x), colors = 'k', linestyles='--', lw = 1, label = "Distribución teórica")
ax.legend(loc = 'best', frameon = False)

plt.show()
```




```
fix, ax = plt.subplots(1,1)
r = binom.rvs(n,p,size=10000)
ax.hist(r, bins=7)
```

```
## (array([ 287., 1328., 2651., 2966., 1858.,  726.,  184.]), array([0., 1., 2., 3., 4., 5., 6., 7.]),
```

```
plt.show()
```


3000
2500
2000
1500
1000
500
0
0
1
2
3
4
5
6
7