# Geométrica

Curso de Estadística Descriptiva

4/2/2019

## Función de densidad

Sea $X = Geom(p = 0.1)$ la distribución que modela la probabilidad de intentar abrir una puerta hasta conseguirlo.

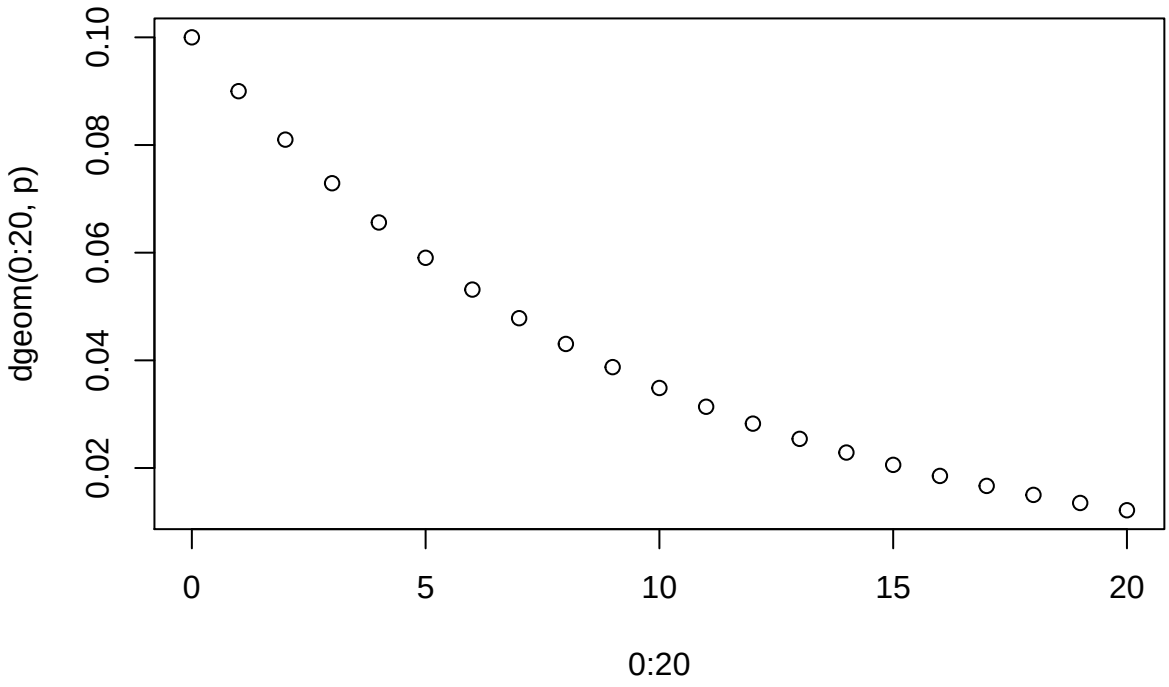
$$f(k) = (1-p)^{k-1}p$$

## En R

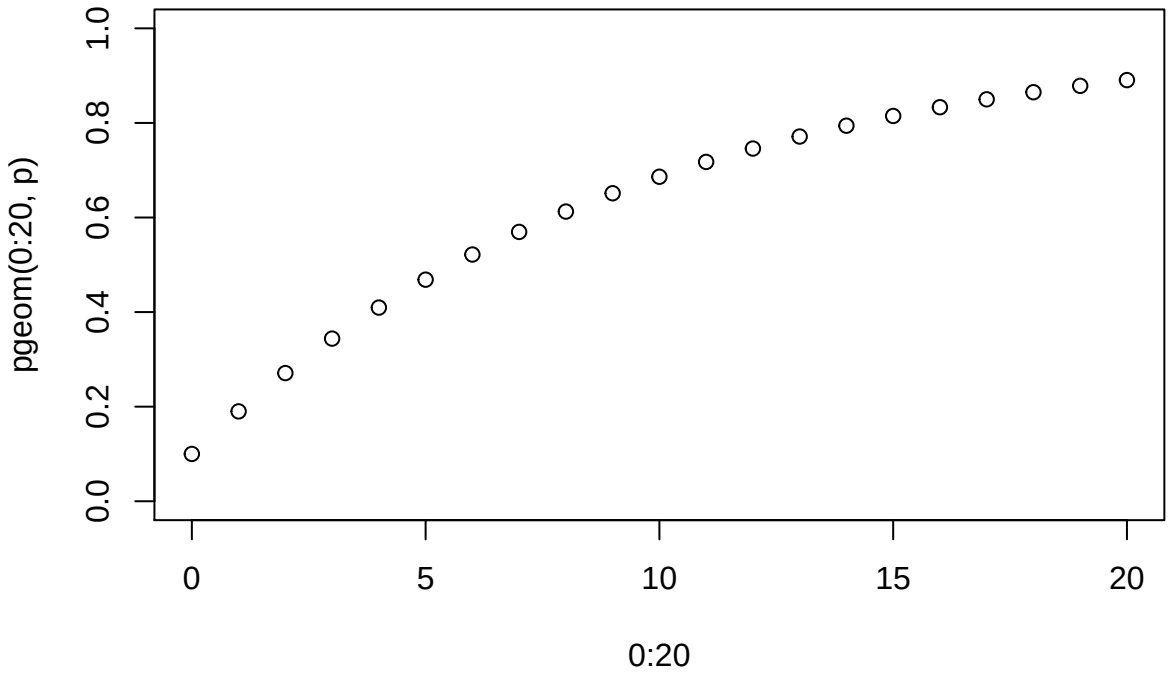
```
library(Rlab)
```

```
## 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
```

```
p = 0.1
plot(0:20, dgeom(0:20, p))
```




```
plot(0:20, pgeom(0:20, p), ylim = c(0,1))
```



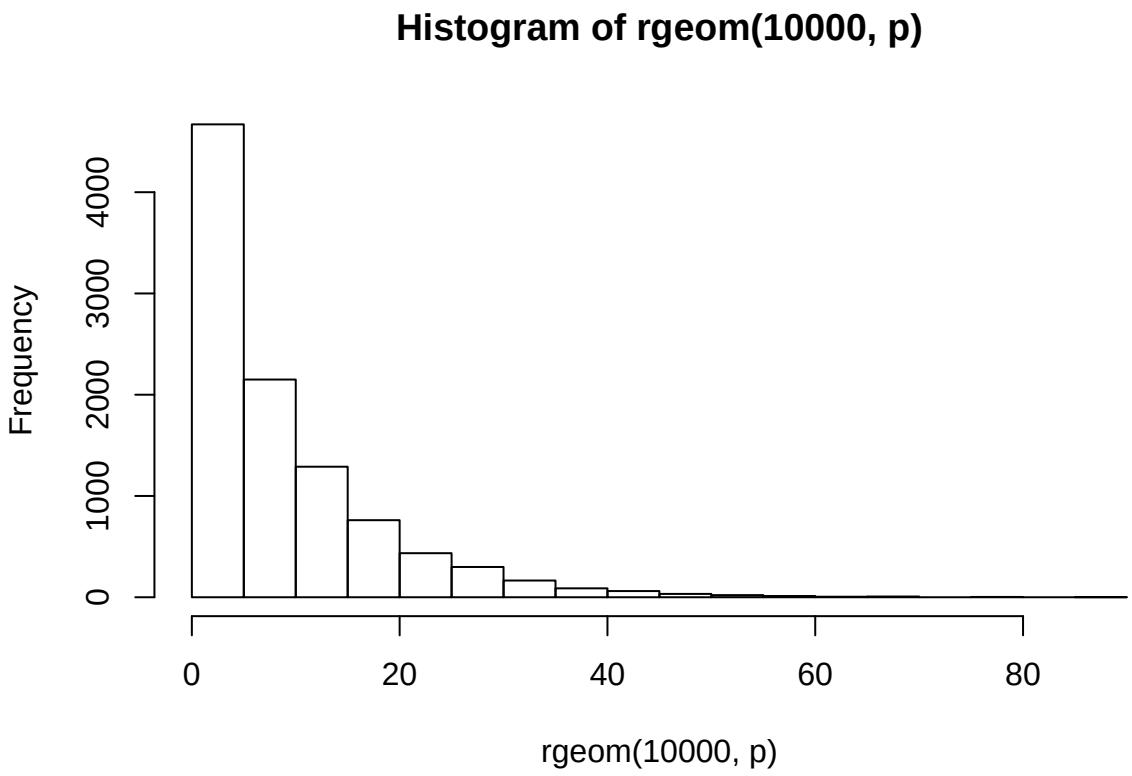

```
qgeom(0.5, p)
```

```
## [1] 6
```

```
qgeom(0.75, p)
```

```
## [1] 13
```

```
hist(rgeom(10000, p))
```




### En Python

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

fig, ax = plt.subplots(1,1)
p = 0.3
mean, var, skew, kurt = geom.stats(p, moments = 'mvsk')
print("Media %f"%mean)
```

```
## Media 3.333333
```

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

```
## Varianza 7.777778
```

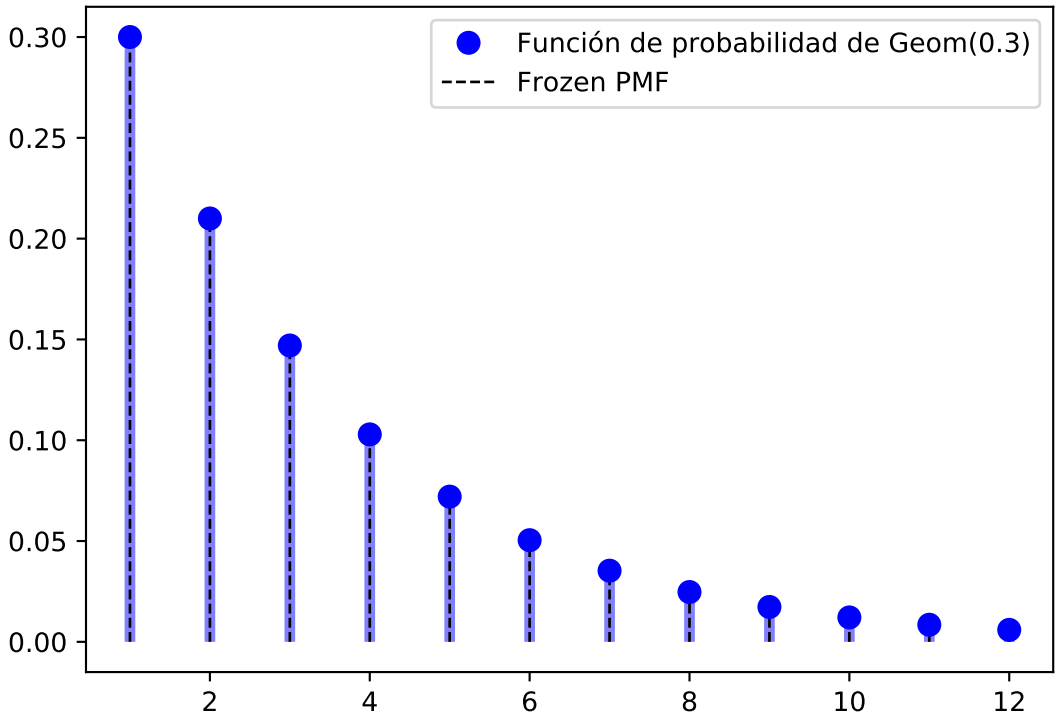
```
print("Sesgo %f"%skew)
```

```
## Sesgo 2.031889
```

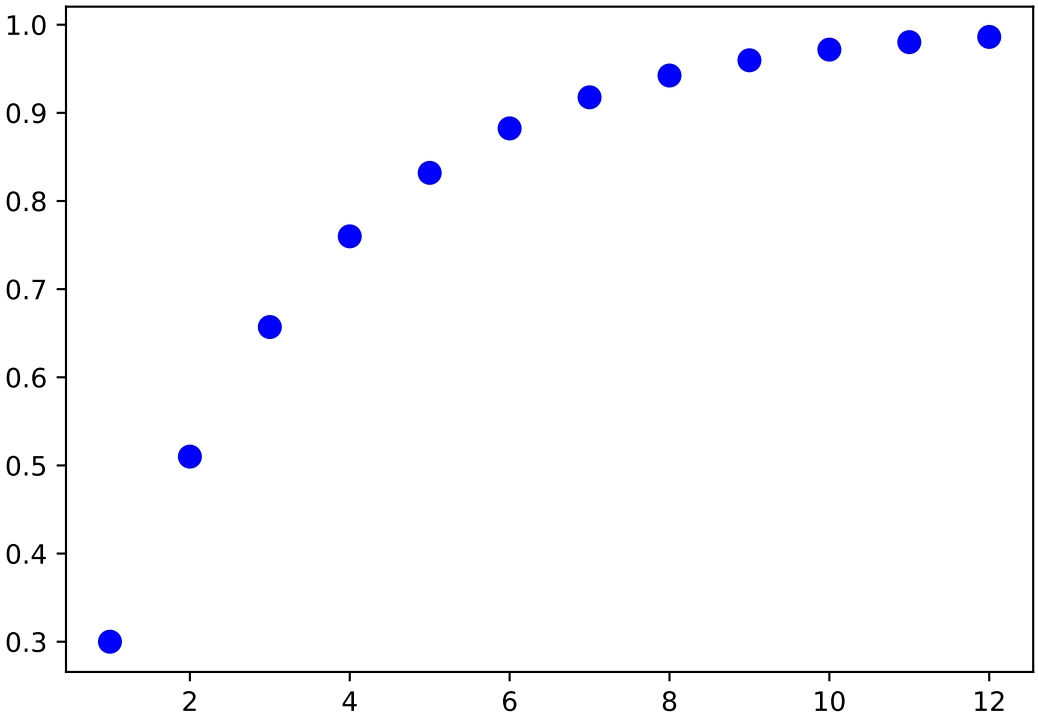
```
print("Curtosis %f"%kurt)
```

```
## Curtosis 6.128571
```

```
x = np.arange(geom.ppf(0.01,p), geom.ppf(0.99, p))
ax.plot(x, geom.pmf(x, p), 'bo', ms = 8, label = "Función de probabilidad de Geom(0.3)")
ax.vlines(x,0,geom.pmf(x,p),  colors = 'b', lw = 4, alpha = 0.5)

rv = geom(p)
ax.vlines(x,0,rv.pmf(x), colors = 'k', linestyles = '--', lw = 1, label = "Frozen PMF")
ax.legend(loc = 'best')
plt.show()
```



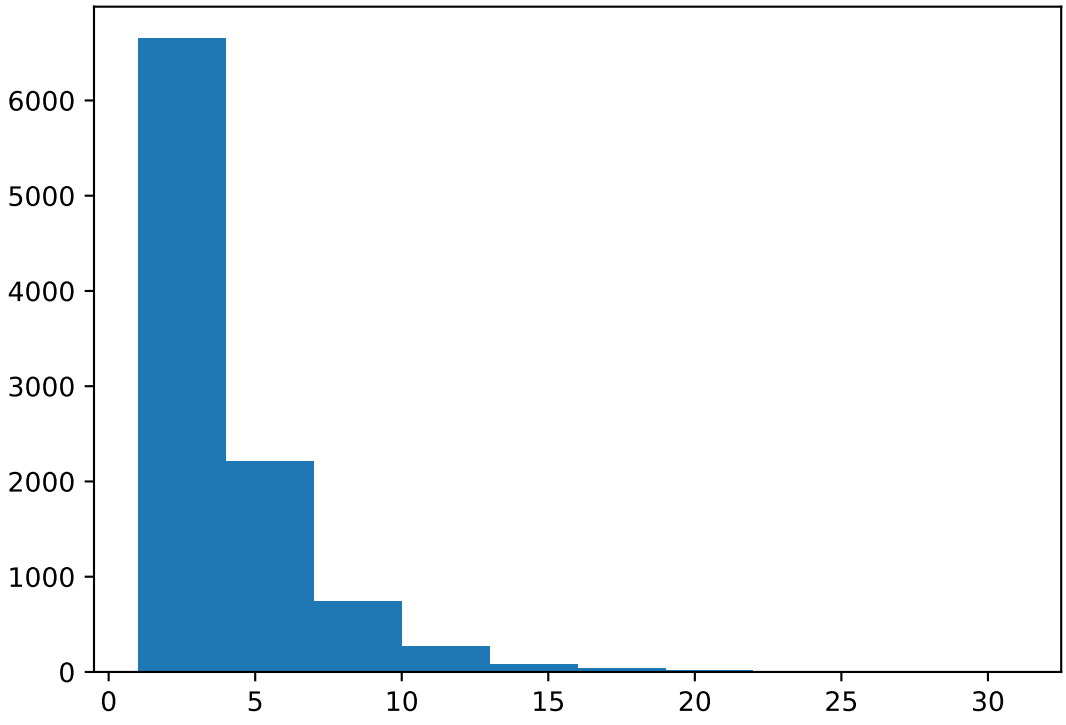

```
fig, ax = plt.subplots(1,1)
prob = geom.cdf(x,p)
ax.plot(x, prob, 'bo', ms = 8, label = "Función de distribución acumulada")
plt.show()
```




```
fig, ax = plt.subplots(1,1)
r = geom.rvs(p, size = 10000)
plt.hist(r)
```

```
## (array([6.652e+03, 2.205e+03, 7.410e+02, 2.690e+02, 7.700e+01, 3.600e+01,
##         1.400e+01, 4.000e+00, 1.000e+00, 1.000e+00]), array([ 1.,  4.,  7., 10., 13., 16., 19., 22., 
```

```
plt.show()
```


6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
30