Distribucion Geometrica

Diego Delgado Palomares

7/10/2020

Distribución Geométrica $X \sim \mathbf{Ge}(p)$

Su función de densidad viene dada por

$$\begin{array}{ll} f(k) = (1-p)^k p & \text{si empieza en } 0 \\ f(k) = (1-p)^{k-1} & \text{si empieza en } 1 \end{array}$$

Su función de ditribución es

$$F(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1-(1-p)^{k+1} & \text{si } k \leq x < k+1, k \in \mathbb{N} \end{cases}$$

con Esperanza $E(X) = \frac{1-p}{p}$

y Varianza $Var(X) = \frac{1-p}{p^2}$

Propiedad de la falta de memoria

Si X es una variable aleatoria Ge(p) , entonces:

$$p\{X \geq m+n : X \geq n\} = p\{X \geq m\} \ \forall \ m, n = \ 0, 1, ..$$

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

```
library(Rlab)
```

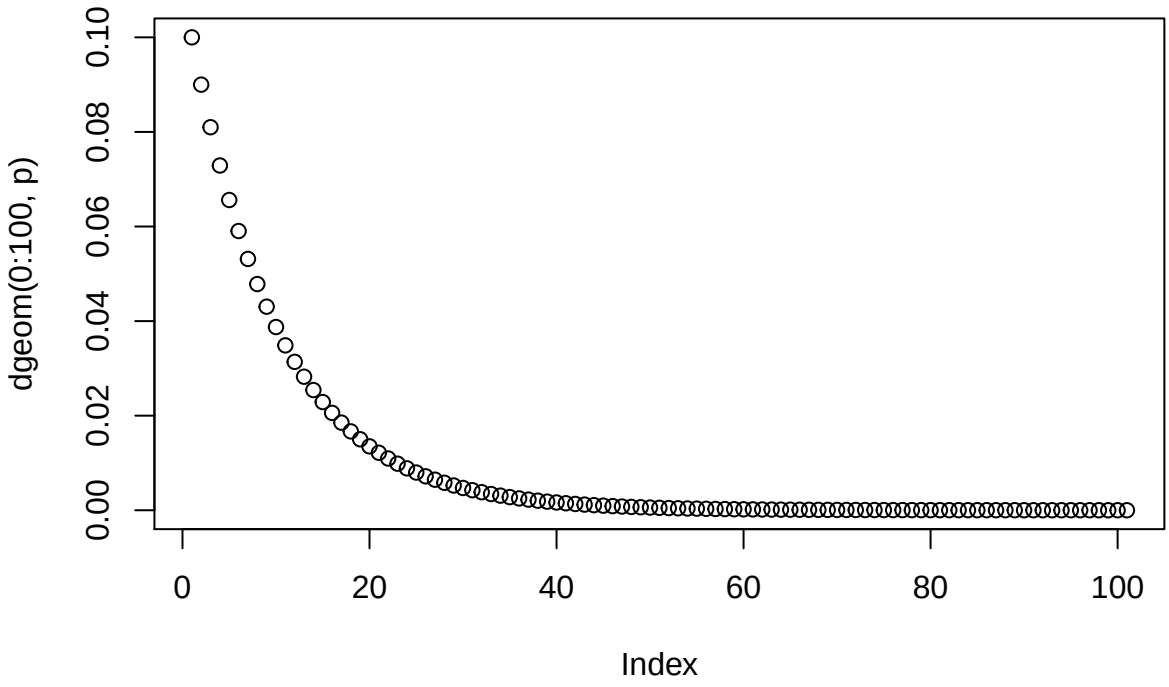
```
## Rlab 2.15.1 attached.
```

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

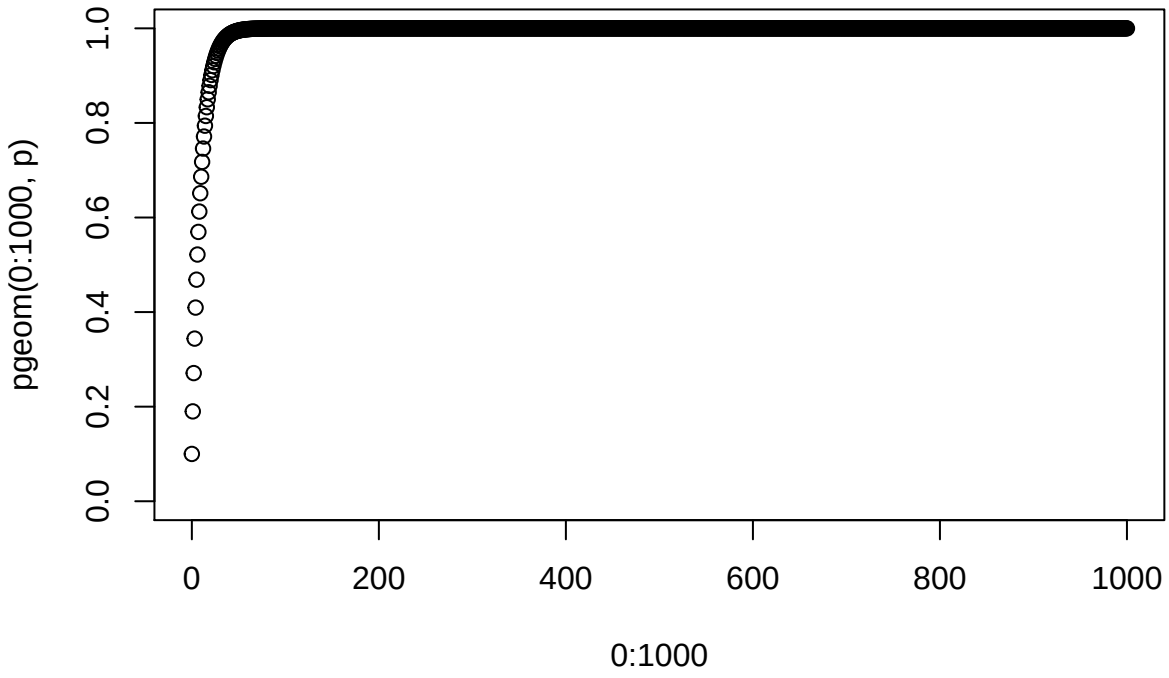
```
## The following objects are masked from 'package:stats':
##
##     dexp, dgamma, dweibull, pexp, pgamma, pweibull, qexp, qgamma,
##     qweibull, rexp, rgamma, rweibull
```

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

```
p=0.1
plot(dgeom(0:100,p))
```




```
plot(0:1000, pgeom(0:1000, p), ylim = c(0,1)  )
```



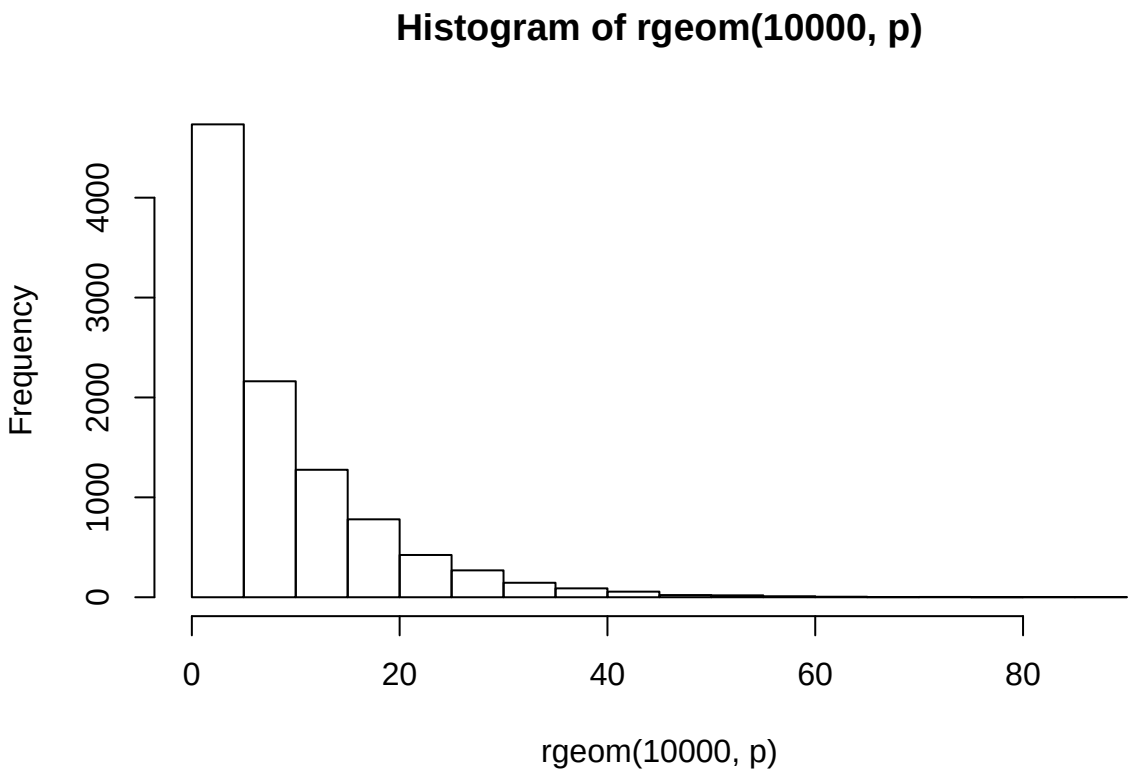

```
qgeom(0.5,p)
```

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

```
qgeom(0.7,p)
```

```
## [1] 11
```

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




```
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
```

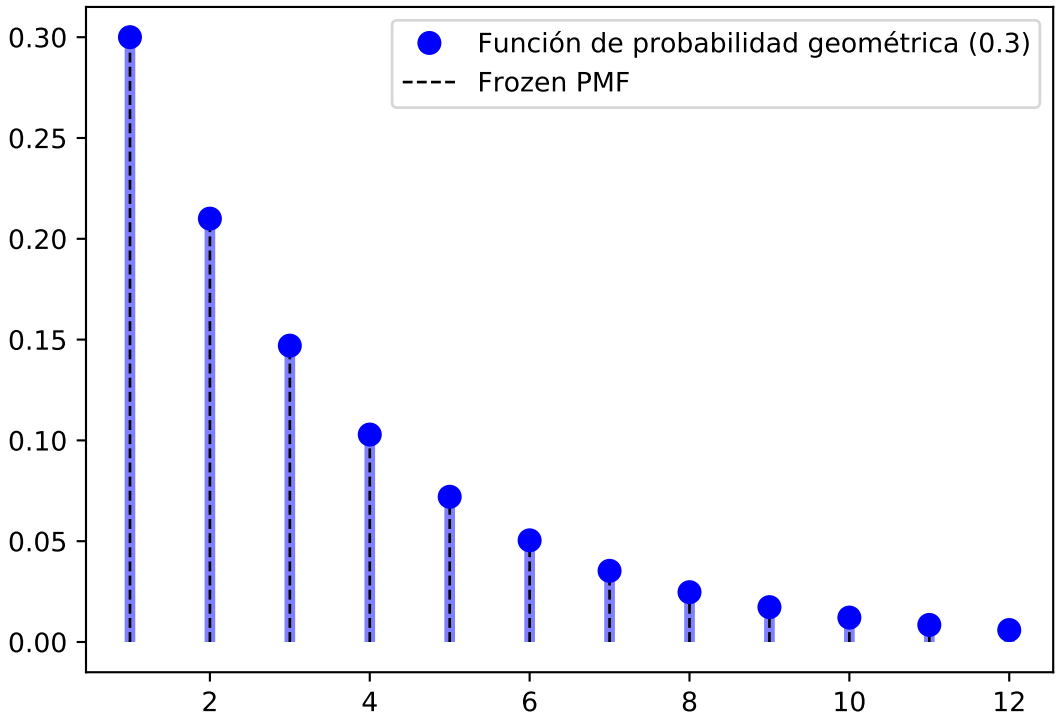
```
print("Curtosis %f"%kurt)
```

```
## Curtosis 6.128571
```

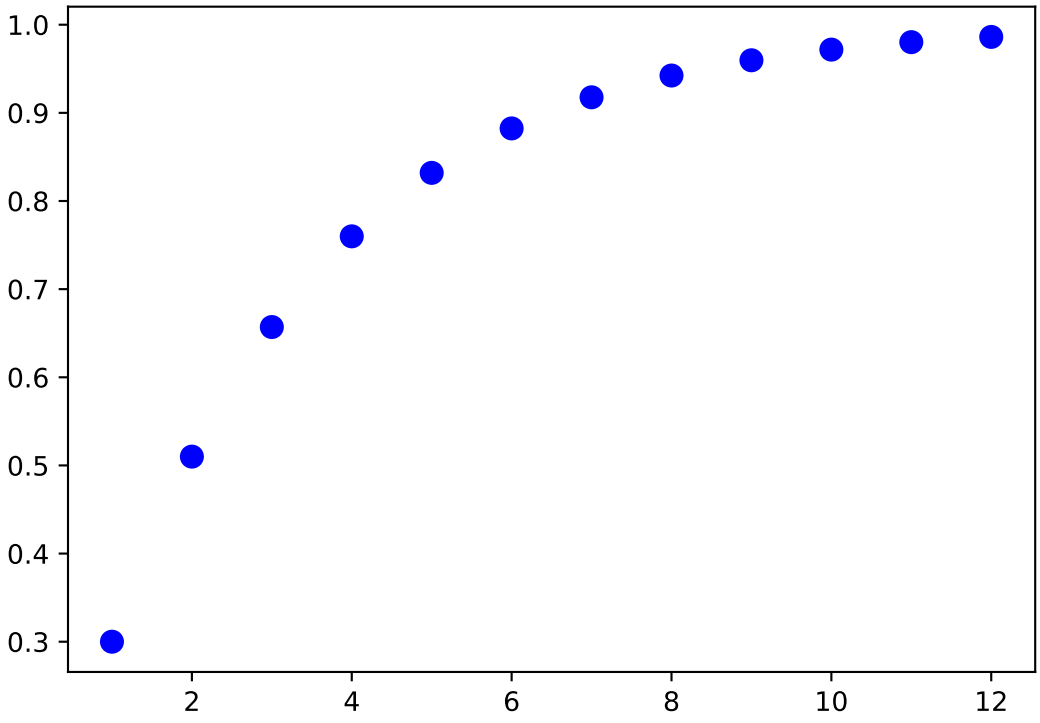
```
x = np.arange(geom.ppf(0.01,p), geom.ppf(0.99,p))
ax.plot(x,geom.pmf(x,p), "bo", ms = 8, label= "Función de probabilidad geométrica (0.3)")
ax.vlines(x,0, geom.pmf(x,p),  colors="b", lw=4, alpha = 0.5)

rv = geom(p)
ax.vlines(x,0,rv.pmf(x), colors ="k", linestyles = '--', lw = 1,
```

```
label = "Frozen PMF")
ax. legend(loc= "best")
plt.show()
```



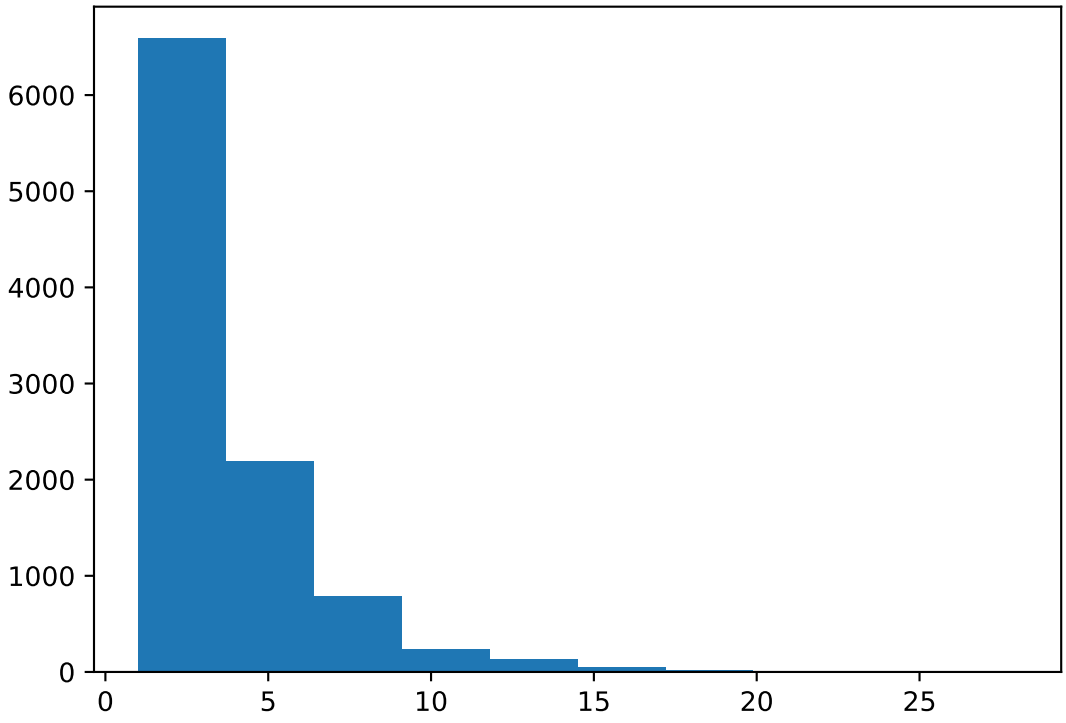

```
fig, ax = plt.subplots(1,1)
prob = geom.cdf(x,p)
ax.plot(x, prob, "bo", ms = 8, label = "Función de acumulación")
plt.show()
```




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

```
## (array([6.590e+03, 2.187e+03, 7.850e+02, 2.300e+02, 1.340e+02, 4.900e+01,
##         1.500e+01, 7.000e+00, 2.000e+00, 1.000e+00]), array([ 1. ,  3.7,  6.4,  9.1, 11.8, 14.5, 17.2
```

```
plt.show()
```


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