

Distribucion hipergeometrica

Victor Lopez

2023-02-22

Distribución Hipergeométrica

Consideremos el experimento “extraer a la vez (o una detrás de otra, sin retornarlos) n objetos donde hay N de tipo A y M de tipo B”. Si X es variable aleatoria que mide el “número de objetos del tipo A”, diremos que X se distribuye como una Hipergeométrica con parámetros N, M, n

$$X \sim H(N, M, n)$$

- El **dominio** de X será $D_X = \{0, 1, 2, \dots, N\}$ (en general)
- La **función de probabilidad** vendrá dada por

$$f(k) = \frac{\binom{N}{K} \binom{M}{n-k}}{\binom{N+M}{n}}$$

- La **función de distribución** vendrá dada por

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

- **Esperanza** $E(X) = \frac{nN}{N+M}$
- **Varianza** $Var(X) = \frac{nNM}{(N+M)^2} \cdot \frac{N+M-n}{N+M-1}$

El código de la distribución Hipergeométrica:

- En R tenemos las funciones del paquete Rlab: `dhyper(x, m, n, k)`, `phyper(q, m, n, k)`, `qhyper(p, m, n, k)`, `rhyper(nn, m, n, k)` donde `m` es el número de objetos del primer tipo, `n` el número de objetos del segundo tipo y `k` el número de extracciones realizadas.
- En Python tenemos las funciones del paquete `scipy.stats.hypergeom`: `pmf(k, M, n, N)`, `cdf(k, M, n, N)`, `ppf(q, M, n, N)`, `rvs(M, n, N, size)` donde `M` es el número total de datos en la población, `N` es el número de extracciones realizadas y `n` el número de datos del primer tipo (el que queremos).

En R

Supongamos que tenemos 20 animales, de los cuales 7 son perros. Queremos medir la probabilidad de encontrar un número determinado de perros si elegimos $x = 12$ animales al azar

```
library(Rlab)
```

```
## Rlab 4.0 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
```

```
M = 7
```

```
N = 13
```

```
k = 12
```

```
dhyper(x = 0:12, m = M, n = N, k = k)
```

```
## [1] 0.0001031992 0.0043343653 0.0476780186 0.1986584107 0.3575851393
```

```
## [6] 0.2860681115 0.0953560372 0.0102167183 0.0000000000 0.0000000000
```

```
## [11] 0.0000000000 0.0000000000 0.0000000000
```

```
phyper(q = 0:12, m = M, n = N, k = k)
```

```
## [1] 0.0001031992 0.0044375645 0.0521155831 0.2507739938 0.6083591331
```

```
## [6] 0.8944272446 0.9897832817 1.0000000000 1.0000000000 1.0000000000
```

```
## [11] 1.0000000000 1.0000000000 1.0000000000
```

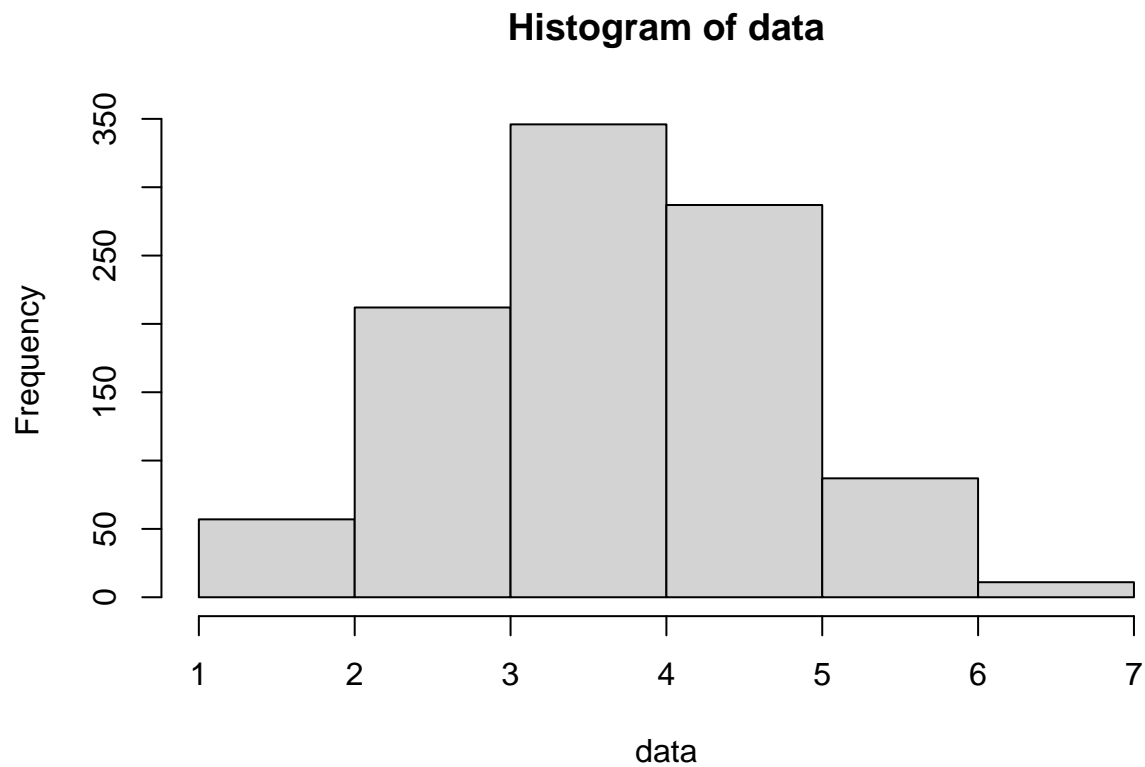
```
qhyper(p = 0.5, m = M, n = N, k = k) # Quiere decir que hay un 50% de que tenga la
```

```
## [1] 4
```

```
# cantidad devuelta o menos
```

```
rhyper(nn = 1000, m = M, n = N, k = k) -> data
```

```
hist(data, breaks = 8)
```



En Python

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

M, n, N = 20, 7, 12

rv = hypergeom(M, n, N)
x = np.arange(0, n+1)
y = rv.pmf(x)

mean, var, skew, kurt = rv.stats(moments = "mvsk")

fig = plt.figure()
ax = fig.add_subplot(111)
ax.plot(x, y, "bo")
```

