

Distribucion de Bernoulli

Victor Lopez

2023-02-19

Distribucion de Bernoulli

El código de la distribución de Bernoulli:

- En R tenemos las funciones del paquete Rlab: `dbern(x,prob)`, `pbern(x,prob)`, `qbern(p,prob)`, `rbern(n, prob)` donde `prob` es la probabilidad de éxito.
- En Python tenemos las funciones del paquete `scipy.stats.bernoulli`: `pmf(x,p)`, `cdf(x,p)`, `ppf(q,p)`, `rvs(p, size)` donde `p` es la probabilidad de éxito.

En R

`dbern(x, p)`. `x` es alguno de los valores que puede tomar `X`. `p` probabilidad de éxito

```
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

# Funcion de densidad
dbern(0, prob = 0.7) # Probabilidad de que en una variable que sigue distribucion de bernoulli

## [1] 0.3

# sea igual a 0
dbern(1, prob = 0.7)

## [1] 0.7
```

```
# Funcion de distribucion acumulada  
pbern(0, prob = 0.7)
```

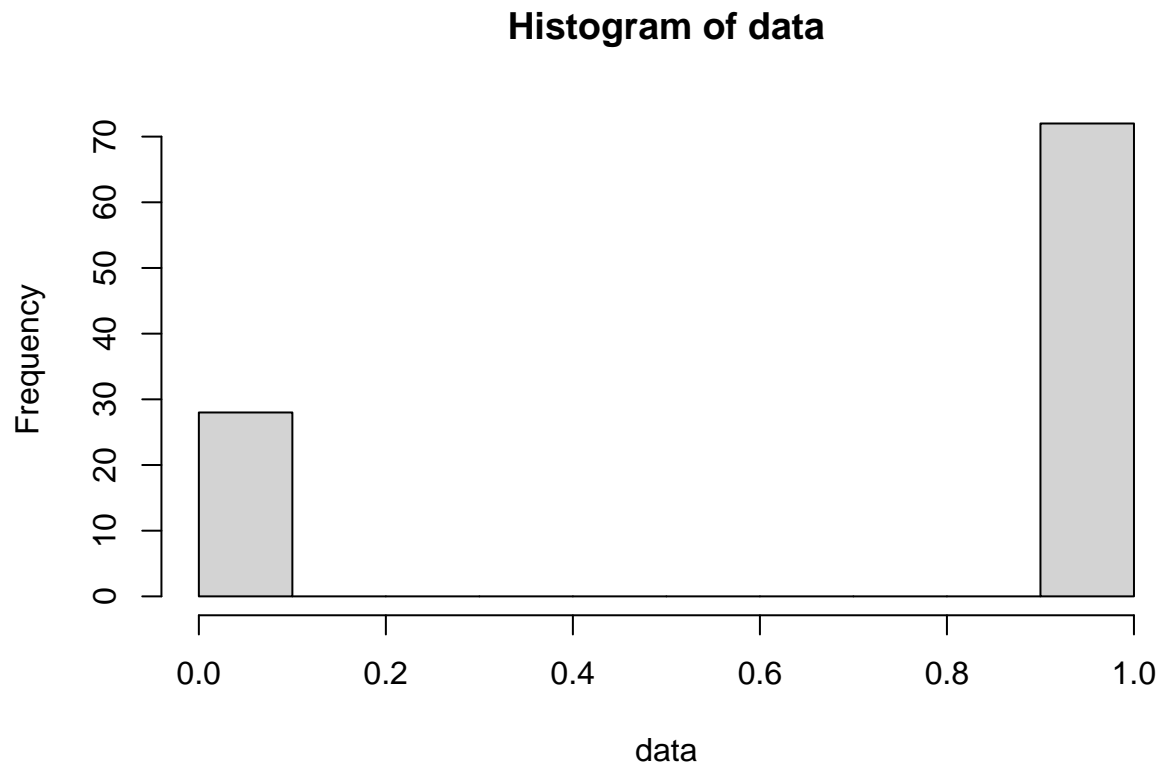
```
## [1] 0.3
```

```
pbern(1, prob = 0.7)
```

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

```
# Vector de numeros random con distribucion Bernoulli  
rbern(100, prob = 0.7) -> data # Creamos un vector de numeros random de 0 y 1's en el  
# que el 1 tiene una probabilidad de 0.7 de salir y 0 tiene probabilidad de 0.3
```

```
hist(data)
```



En Python

```
from scipy.stats import bernoulli  
import matplotlib.pyplot as plt
```

```
p = 0.7
```

```
mean, var, skew, kurt = bernoulli.stats(p, moments = "mvsk" ) # mvsk => mean, var,  
# skewness, kurtosis
```

```
print(mean, var, skew, kurt)
```

```
## 0.7 0.21000000000000002 -0.8728715609439702 -1.2380952380952361
```

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

```
x = bernoulli.rvs(p, size = 1000)
```

```
ax.hist(x)
```

```
## (array([303.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0., 697.]), array([0. , 0.1, 0.2, 0.3, 0.4
```

```
plt.show())
```

