

Introducción a distribuciones de probabilidad

Ramon Ceballos

7/2/2021

Distribución de Bernoulli

1. Conceptos teóricos y matemáticos

Si X es variable aleatoria que mide el “número de éxitos” y se realiza un único experimento con dos posibles resultados (éxito, que toma valor 1, o fracaso, que toma valor 0), diremos que X se distribuye como una Bernoulli con parámetro p , del siguiente modo:

$$X \sim \text{Be}(p)$$

En dicha expresión, p es la probabilidad de éxito y $q = 1 - p$ es la probabilidad de fracaso.

Vamos a definir algunas de las propiedades que tendrá una distribución de Bernoulli.

- El **dominio** de X será $D_X = \{0, 1\}$. Debido a que solo se pueden tomar los valores 0 y 1.
- La **función de probabilidad** vendrá dada por:

$$f(k) = p^k(1-p)^{1-k} = \begin{cases} p & \text{si } k = 1 \\ 1-p & \text{si } k = 0 \\ 0 & \text{en cualquier otro caso} \end{cases}$$

En esta expresión k puede tomar los valores del D_X (0 y 1).

- La **función de distribución** (va acumulando los valores que se va encontrando hasta un momento dado) vendrá dada por:

$$F(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1-p & \text{si } 0 \leq x < 1 \\ 1 & \text{si } x \geq 1 \end{cases}$$

En esta expresión para todo valor de k negativo, valdrá 0, porque no se ha podido dar un valor presente en D_X . Para un valor entre (0,1), el valor será $q = 1 - p$. Para valores de k superiores a 1, el valor de $F(k)$ será de $q + p = 1$.

- **Esperanza.** La definición global es $E(X) = p$. Proviene de:

$$E(X) = \sum_{x \in D_X} x \cdot f(x) = (0 \cdot (1-p)) + (1 \cdot p) = p$$

- **Varianza.** $\text{Var}(X) = pq$. Proviene de:

$$Var(X) = E(X^2) - (E(X))^2 = p - p^2 = p \cdot (1 - p) = p \cdot q$$

En lenguaje cotidiano, la variable aleatoria de Bernoulli puede referir a la probabilidad de aprobar un examen, la probabilidad de sacar cara en un lanzamiento de una moneda...

2. Distribución de Bernoulli en R y Python

El código de la distribución de Bernoulli en estos dos lenguajes de programación, vendrá dado por los siguiente:

- En **R** tenemos las funciones del paquete **Rlab**: `dbenr(x,prob)` para la función de probabilidad, `pbenr(q,prob)` para la función de distribución, `qbenr(p,prob)` para obtener un cuantil determinado, `rbenr(n, prob)` para generar los n^{os} aleatorios. En estas funciones, el parámetro **prob** es la probabilidad de éxito.
- En **Python** tenemos las funciones del paquete **scipy.stats.bernoulli**: `pmf(k,p)` para la función de probabilidad, `cdf(k,p)` para la función de distribución, `ppf(q,p)` para obtener un cuantil determinado, `rvs(p, size)` para generar los n^{os} aleatorios. En estas funciones el parámetro **p** es la probabilidad de éxito.

3. Ejemplos en código para la distribución de Bernoulli (R y Python)

Función de probabilidad

Sea $X = Be(p = 0.7)$, la distribución que modela la probabilidad de obtener una cara usando una moneda trucada.

$$f(k) = p^k(1 - p)^{1-p}, \quad k \in \{0, 1\}$$

Ejemplo en R Necesitamos llamar al paquete de R conocido como Rlab.

```
#Importamos Rlab
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
```

```
#Obtener la probabilidad de Bernoulli de que sea un fracaso  
dbern(0, prob= 0.7)
```

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

```
#Obtener la probabilidad de Bernoulli de que sea un éxito  
dbern(1, prob = 0.7)
```

```
## [1] 0.7
```

```
#Obtener la función de distribución (acumulada) de Bernoulli para fracaso  
pbern(0, prob = 0.7)
```

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

```
#Obtener la función de distribución (acumulada) de Bernoulli para éxito  
pbern(1, prob = 0.7)
```

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

```
#Obtener la mediana a través de la fórmula de cuantiles  
qbern(0.5, prob = 0.7)
```

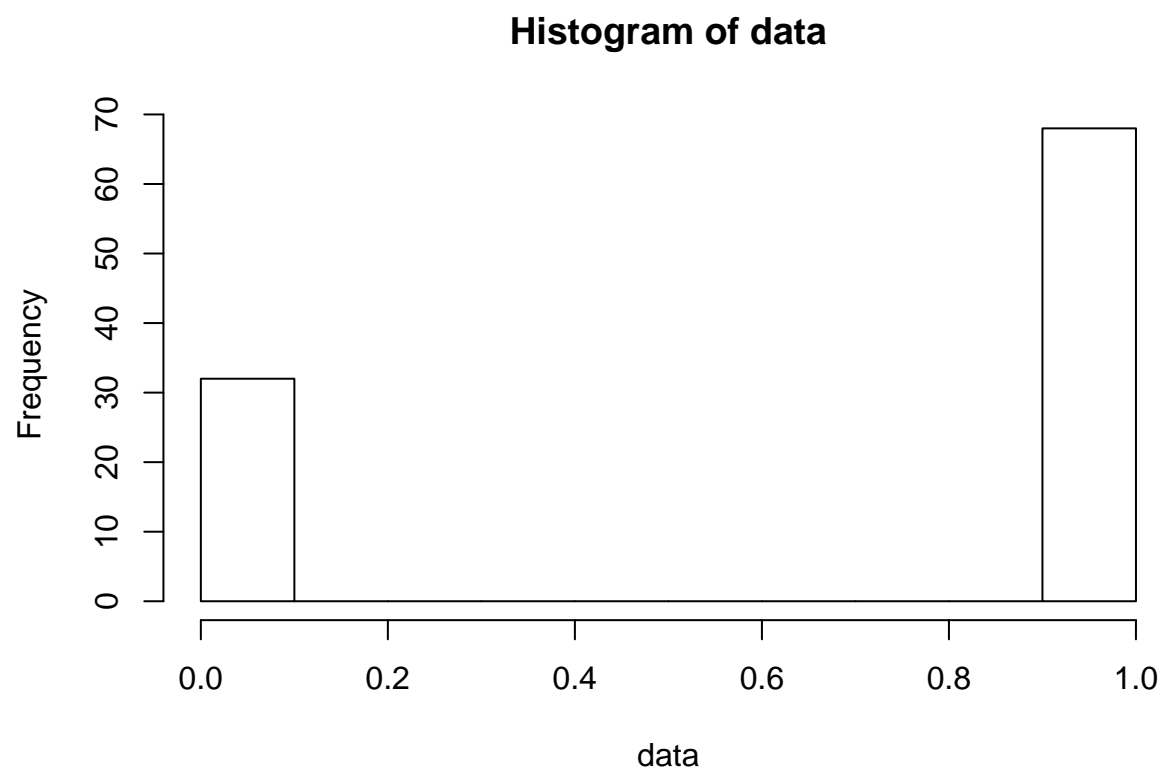
```
## [1] 1
```

```
##Obtener Q(0.25) a través de la fórmula de cuantiles  
qbern(0.25, prob = 0.7)
```

```
## [1] 0
```

```
#Generamos 100 n°s aleatorios para probabilidad de éxito de 0.7  
#Según distribución de Bernoulli  
#Lo guardamos en data  
rbern(100, prob = 0.7) -> data
```

```
#Representamos el histograma de data  
hist(data)
```



Ejemplo en Python Vete al script de Python del presente tema 5.