

05-Distribucion de Bernoulli

Adrian Vitya

22/1/2022

Distribucion de Bernoulli

Si X es v.a. que mide el “numero de exitos” y se realiza un unico experimento con posibles resultados (Exito = 1, Fracaso = 0), diremos que X se distribuye como una Bernoulli con parametro p .

$$X \sim Be(p)$$

donde p es la probabilidad d exito y $q = 1 - p$ que es la probabilidad de fracaso.

- El **dominio** de X sera $X(\Omega) = \{0,1\}$
- La **funcion de probabilidad** vendra dada por

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

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

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

- **Esperanza** $E(X) = p$
- **Varianza** $Var(X) = pq$

Codigo de distribucion de Bernoulli

En **R** tenemos las funciones del paquete **Rlab**: - **dbenr(x, prob)** - **pbenr(q, prob)** - **qbenr(p, prob)** - **rbenr(n, prob)** Donde $prob$ es la probabilidad de exito

En **Python** tenemos las funciones del paquete **scipy.stats.bernoulli**: - **pmf(k,p)** - **cdf(k,p)** - **ppf(q,p)** - **rvs(p,size)** Donde p es la probabilidad de exito

Funcion de densidad

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

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

```
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
```

```
# Probabilidad de Fracasar
```

```
dbern(0, prob=0.7)
```

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

```
# Probabilidad de Exito
```

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

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

```
# La mediana
```

```
qbern(0.5, prob=0.7)
```

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

```
# El primer cuantil
```

```
qbern(0.25, prob=0.7)
```

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

```
# Experimento con 100 observaciones
```

```
rbern(100, prob = 0.7) -> data
```

```
# Mostrar un histograma con el experimento
```

```
hist(data)
```

