

Distribución de Bernoulli

Imanol

3/3/2021

Función de densidad

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

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

EN R

```
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
```

```
dbern(0, prob = 0.7) # Calcula la probabilidad de sacar 0
```

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

```
dbern(1, prob = 0.7) # Calcula la probabilidad de sacar 1
```

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

```

pbern(0, prob = 0.7) # Función acumulada en 0

## [1] 0.3

pbern(1, prob = 0.7) # Función acumulada en 1

## [1] 1

qbern(0.5, prob = 0.7) # Calcular la mediana dándole la distribución

## [1] 1

qbern(0.25, prob = 0.7) # Calcular la Q1 dándole la distribución

## [1] 0

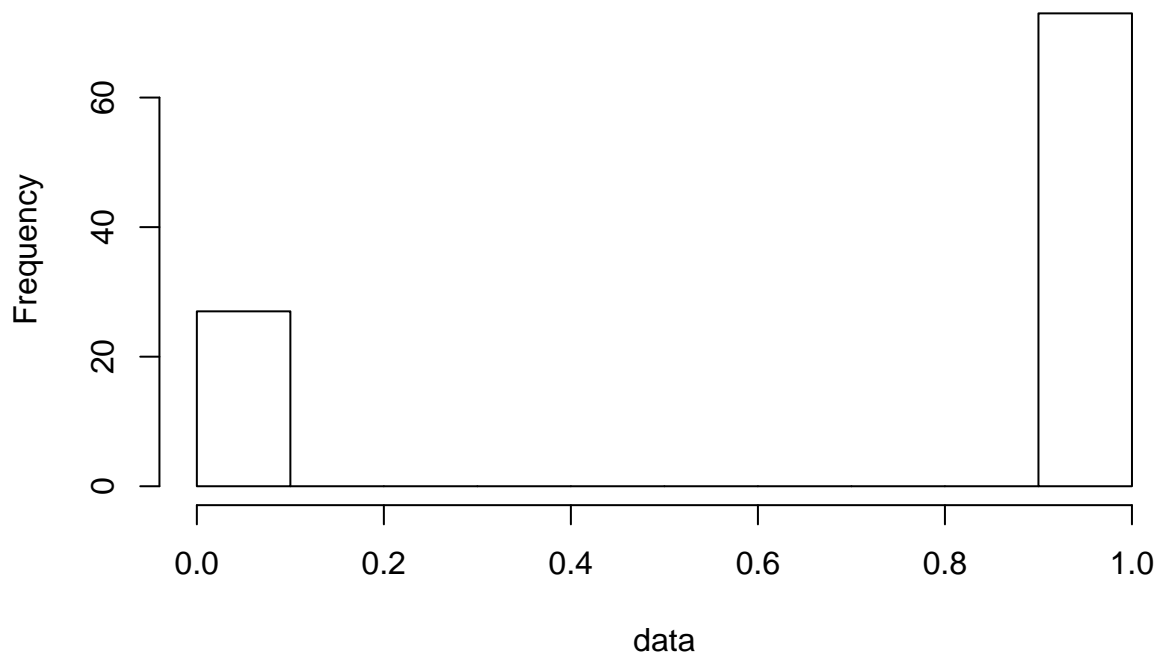
rbern(100, prob = 0.7) # Generar numeros aleatorios con esta distribución

##      [1] 1 1 1 1 1 1 0 0 1 1 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 0 1 0 0 0 1 1 1 1 0 0 1
##     [38] 0 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 0 1 1 1 1 1 1 1
##    [75] 0 1 1 1 1 1 1 0 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 0

rbern(100, prob = 0.7) -> data
hist(data) # Creo un histograma con esos datos aleatorios creados

```

Histogram of data



EN PYTHON

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

p = 0.7

# Cada una de las letras es cada uno de los parametros
#Ejemplo: mean = m
mean, var, skew, kurt = bernoulli.stats(p, moments = 'mvsk')
print("Media %f"%mean)

## Media 0.700000

print("Varianza %f"%var)

## Varianza 0.210000

print("Sesgo %f"%skew)

## Sesgo -0.872872

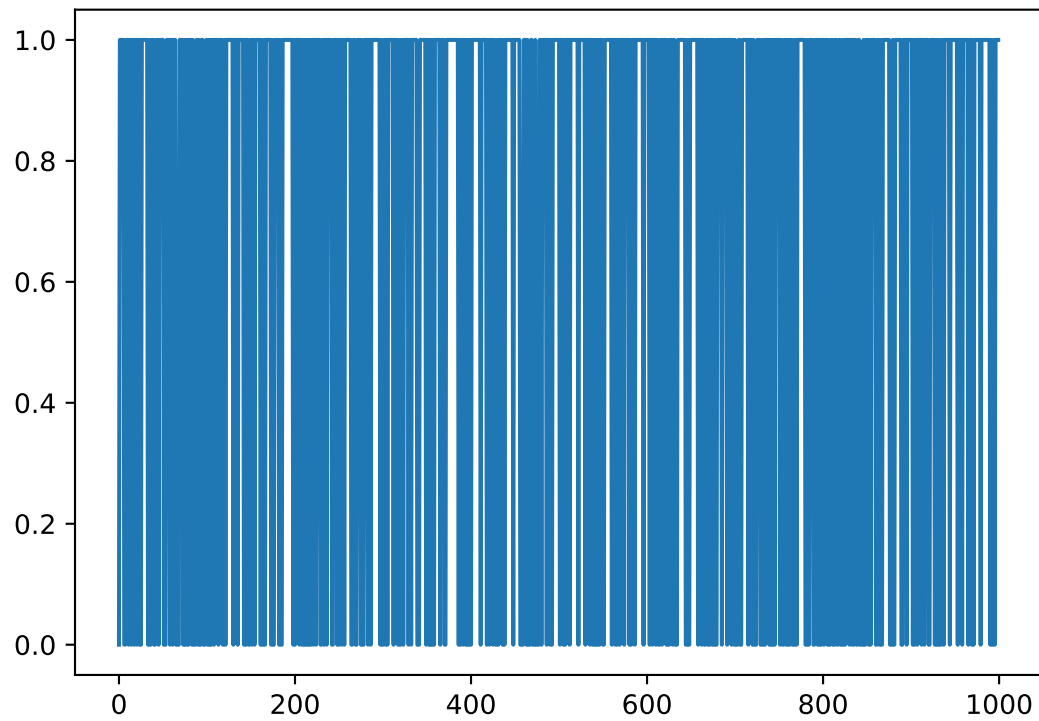
print("Curtosis %f"%kurt)

# Función de probabilidad

## Curtosis -1.238095

fix, ax = plt.subplots(1,1)
x = bernoulli.rvs(p, size = 1000)
ax.plot(x) # raya vertical azul si es un exito
           # raya vertical blanca si es un fracaso
plt.show()

# Como historgrama se ve mejor
```



```
fix, ax = plt.subplots(1,1)
x = bernoulli.rvs(p, size = 1000)
ax.hist(x) # raya vertical azul si es un exito
           # raya vertical blanca si es un fracaso
```

```
## (array([284.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0., 716.]), array([0. , 0.1, 0.2, 0.3, 0.4
```

```
plt.show()
```

