Práctica 4

Alejandro Cáceres
UPC - Statistics 2019/2020

Objetivos teóricos

Estudiar las distribuciońes de variables aleatorias discretas

- Distribuciones generales
- Ensayos de Beronulli
- Distribución Binomial

Objetivos de R

- Simular datos
- Usar funciones de R
- Hacer bucles
- Usar las funciones para cálculo de probabilidad definidas en R

Dados

La función de distribución para la suma de dos dados a y b

$$f(x) = \begin{cases} 1/36, & \text{si } x = 2(a:1, b:1), 12(a:6, b:6) \\ 2/36, & \text{si } x = 3, 11 \\ 3/36, & \text{si } x = 4, 10 \\ 4/36, & \text{si } x = 5, 9 \\ 5/36, & \text{si } x = 6, 8 \\ 6/36, & \text{si } x = 7 \end{cases}$$

Dados

definamos los vectores:

```
x <- seq(2,12)
fx <- c(1,2,3,4,5,6,5,4,3,2,1)/36
#podemos darle nombres los valores fx
names(fx) <- x
fx
```

Cuál es la probabilidad de que un lanzamiento de dos dados de un número par?

Dados

- dibuja la distribución con **plot(..., type="p", pch=16)**
- añádele la líneas con **lines()**:

Para una línea:

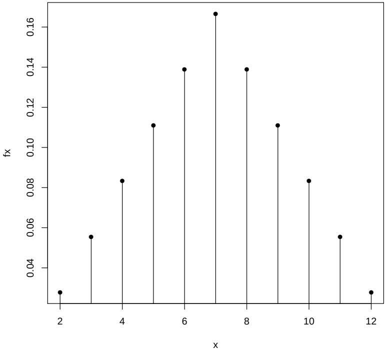
```
lines(c(x[2],x[2]),c(0,fx[2]),col="red")
```

Para todas las 11 líneas (longitud de x):

```
for(i in 1:11)
{ lines(c(x[i], x[i]), c(0, fx[i])) }
```

Dados

puedes conseguir?



Dados

- Cuál es la media de f(x)?

$$\mu = \sum_i x_i f(x_i)$$

Usa **sum** y la multipicación entre vectores, asigna el resultado a la variable `mu`

Dados

- Cuál es la varianza de f(x)?

$$\sigma^2 = \sum_i (x_i - \mu)^2 f(x_i)$$

Usa **sum** y funciones aritméticas entre vectores

Dados

- Cuál es la mediana de f(x)?

$$F(x) = Pr(x \leq mediana) = 0.5$$

Calcula la frecuencia relativa acumulada (usa **cumsum** sobre fx) y asígnala a F. Donde está el 50%?

A la frecuencia acumulada hasta 6 súmale la mitad de la probabilidad en 7. Cuánta probabilidad acumulada obtienes en total?

Dados

Cómo podemos jugar a los dados con R?

Simulemos el resultado de sumar **un** lanzamiento de dos dados

```
lanzamiento <- sample(x, size=1, prob=fx,
                      replace=TRUE)
lanzamiento
```

Qué parámetro tienes que cambiar para simular el resulatdo de 100 lanzamientos? asigna el resultado a la variable `lanzamientos`

Dados

Pinta el histograma de `lanzamientos`, usa el parámetro freq=FALSE para hacer el histograma de frecuencia relativa y `breaks=seq(1.5, 12.5)` para centrar las barras en cada valor de x.

```
hist(lanzamientos, freq=FALSE, breaks=seq(1.5, 12.5))
```

añade los puntos de la distribución con:

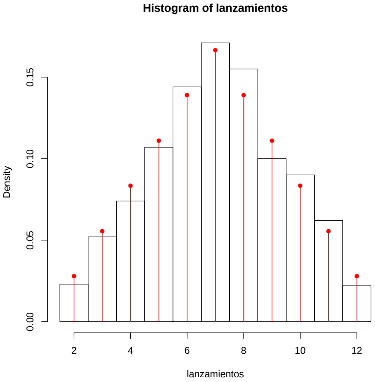
```
points(x, fx, pch=16, col="red")
```

y las líneas con:

```
for(i in 1:11)
{lines(c(x[i],x[i]), c(0,fx[i]), col="red")}
```

Dados

puedes conseguir?



Dados

Ahora estamos hablando de datos!

- Cuál es media (promedio) de lanzamientos?
- Cuál es la varianza de lanzamientos?
- Cuál es la mediana del lanzamientos?

Recuerda funciones mean, $\mathrm{sd}()^2$ y median.

Son iguales a la media, varianza y mediana de la distribución? Por qué?

Bernoulli

El caso mas sencillo de una función de distribución es un ensayo de Bernoulli donde sólo hay dos posibles eventos (A y B)

- evento A (k=1): tiene probabilidad p
- outcome B (k=0): tiene probabilidad $q = 1 - p$

La función de probabilidad de

$$f(k) = (1-p)^{1-k} p^k$$

El lanzamiendo de una moneda es un ensayo de Bernoulli con $p = 1/2$

Bernoulli

La función de probabilidad de

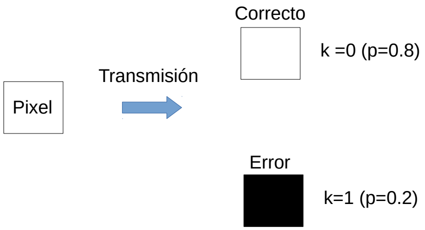
$$f(k) = (1-p)^{1-k} p^k$$

Cuya media y valor desviación estandard son

$$E(x) = \mu = p$$

$$V(X) = \sigma^2 = p(1-p)$$

Bernoulli



Bernoulli

Un pixel tiene probabilidad p=0.2 de ser transmitido incorrectamente (error:k=1) y 0.8 de ser transmitido correctamente (correcto:k=0)

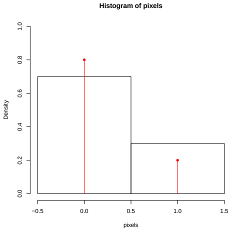
$$f(k) = \begin{cases} 0.8, & \text{si } k = 0 \\ 0.2, & \text{si } k = 1 \end{cases}$$

asigna a k el vector (0,1) y a fk el vector (0.8,0.2) y transmite una foto de 100 pixels con

```
foto <- sample(k, size=100, prob=fk,
replace=TRUE)
foto
```

Bernoulli

puedes conseguir?



- usa **hist** como antes pero añade parámetro ylim=c(0,1) como límites para el eje y; modifica el parámetro breaks de acuerdo al los nuevos valores de k.
- usa **points** como antes.
- usa **lines** con el **for** pero modifica hasta cuándo se hace el bucle.

Bernoulli

Qué tan cerca es la media y la varianza de los datos a sus valores teóricos?

Recuerda que la media del ensayo de Bernoulli es $\mu = p = 0.2$ y su varianza es $\sigma^2 = p * (1 - p) = 0.16$

Distribución Binomial

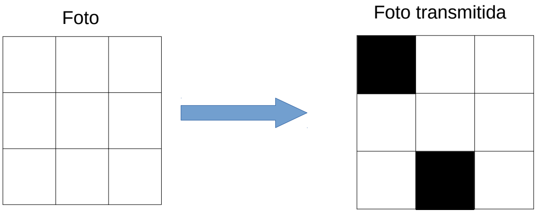
La distribución Binomial da la probabilidad de observar x eventos de un tipo (A) con probabilidad p en n ensayos de Bernoulli.

$$Bin(x; n, p) = \binom{n}{x} p^x (1-p)^{n-x}$$

x=0,1,...n

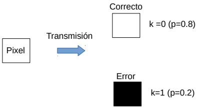
- $E(X) = \mu = np$
- $V(X) = \sigma^2 = np(1-p)$

Distribución Binomial



(Pixel 1, Pixel 2, …. Pixel 9) = (1, 0, 0, 0, 0, 0, 0, 1, 0)

X=2, errores



Distribución Binomial

Enviemos una foto de 100 pixels y contemos cuantas veces aparece el 1 (x=cuantos errores hay).

```
> foto <- sample(x=c(0,1),size=100, prob=fk, replace=TRUE)
> foto
  [1] 0 0 1 0 1  ...
> sum(foto)
[1] 12
```

esta foto tuvo x=12 errores.

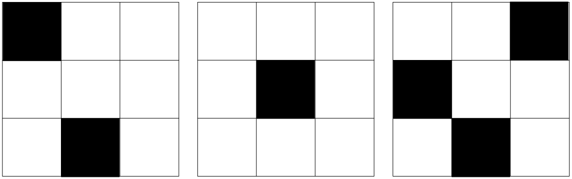
Distribución Binomial

Pero cuál es la probabilidad de observar fotos con 1 error, 2 errores, ... 12 errores, ... 100 errores? Pr(x=2), Pr(x=3), ... Pr(x=100)?

Tenemos que enviar muchas fotos de 100 pixeles (p.ej. 1000) y contar cuantas fotos tienen 1 error, 2 errores, ... 100 errores (fr(x=2), fr(x=3), ... fr(x=100))

Bernoulli

Foto transmitida 1 Foto transmitida 2 Foto transmitida 3



Foto

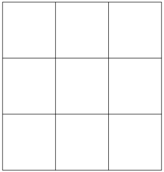
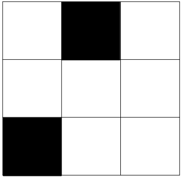


Foto transmitida N



....

Contamos el número total de errores en cada foto.

X = (x1, x2, ….xN) = (2, 1, 3, …. 0, 2) : muestra de “Nfotos”

Distribución Binomial

Hagamos una función para que simule los errores en eviar fotos de 100 pixels (cuando la probabilidad de error por pixel (k=1) es 0.2 -Bernoulli)

```
enviar <- function(...)
{
   foto <- sample(c(0,1),size=100,
                  replace=TRUE,prob=fk)
   sum(foto)
}
```

los tres puntos indican que no importa el argumento en la función, esta siempre hara lo mismo.

Distribución Binomial

```
enviar()
enviar(1)
enviar(1)
enviar("hola")
```

cada vez que llamamos a **enviar** con cualquier argumento, envía una foto de 100 pixels y cuenta los errores.

Distribución Binomial

Podemos simular los errores producidos en el envío de tres fotos de 100 pixeles

```
> c(enviar(1), enviar(1), enviar(1))
[1] 21 14 22
```

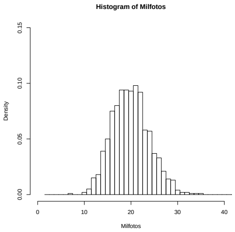
en R esto se pueden hacer con la función **sapply**

```
> Tresfotos <- sapply(rep(1,3), enviar)
> Tresfotos
  [1] 21 19 27
```

Simula los errores al enviar mil fotos, asígnalo a **Milfotos** y pinta su histograma

Distribución Binomial

puedes conseguir?



- usa **hist** como antes pero añade parámetro ylim=c(0,0.15) como límites para el eje y; modifica el parámetro breaks de acuerdo al los nuevos valores de x, breaks=seq(1.5, 100.5); usa xlim=c(0,40) como límites para el eje x.

Distribución Binomial

Cuales son la media (mean) y varianza ($sd()^2$) de Milfotos?

Coinciden con los valores de la media y varianza de una distribución binomial?

- $E(X) = \mu = np = 100 * 0.2 = 20$
- $V(X) = \sigma = np(1 - p) = 10 * 0.3 * 0.7 = 16$

Distribución Binomial

Qué tanto coincide el histograma con la distribución binomial?

Distribución Binomial

Necesitamos los valores de la distribucioń binomial para n=100 y p=0.2. Esto se hace con la función **dbinom**

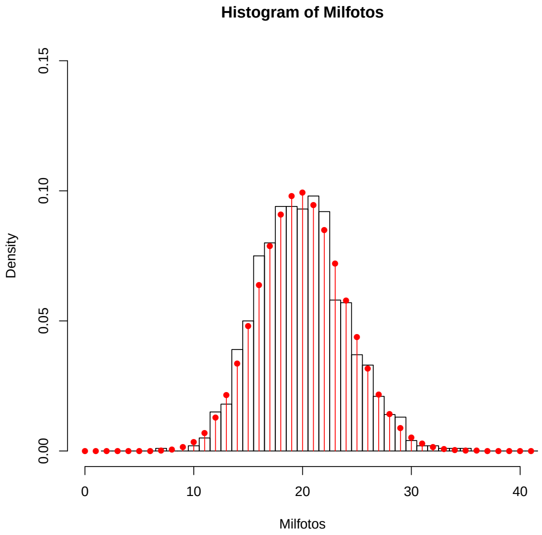
Bin(x; n,p) en R es `dbinom(x, size=n, prob=p)`

```
> x <- 0:100
> binomial <- dbinom(x, size=100, prob=0.2)
> names(binomial) <- x
> head(binomial)
           0            1            2            3
2.037036e-10 5.092590e-09 6.302080e-08 5.146699e-07
```

Distribución Binomial

```
> hist(Milfotos,breaks=seq(1.5,100.5),
freq=FALSE, ylim=c(0,0.15), xlim=c(0,40))

> points(x, binomial, pch= 16, col="red")

> for(i in 1:101)
{lines(c(x[i], x[i]), c(0, binomial[i]),
  col="red")}
```

Distribución Binomial



Distribución Binomial

Usando la función **dbinom** responde:

- Cuál es la probabilidad de observar **exactamente** 5 errores en la tranmisión de una foto de 50 pixeles cuando la probabilidad de error es 0.1? (R/ 0.1849246)
- Cuál es la probabilidad de que haya **exactamente** un error en una foto de 3.1 mega pixels cuando el error en un pixel es de 1e-6? (R/ 0.1396525)

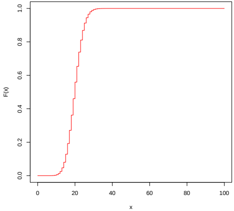
Distribución Binomial

En R está la función **pbinom** que es la función de acumulación de probabilidad para una distribución binomial $F_{bin}(x; n, p)$

$F_{bin}(x; n, p)$ en R es `pbinom(x, size=n, prob=p)`

Distribución Binomial

```
#funcion de acumulacion para n=100 y p=0.2
> binCum <- pbinom(x, size=100, prob=0.2)
> binCum
  [1] 2.037036e-10 5.296294e-09 6.831709e-08 5
> plot(x, binCum, ylim=c(0,1), type="s",
       col="red", ylab="F(x)", xlab="x")
```



Distribución Binomial

Cuál es la probabilidad de que haya **al menos** un error en una foto de 3,1 mega pixels cuando el error en un pixel es de 1e-6? (R/ 0.1847016)

Otras distribuciones

Material adicional:

Distribución Geométrica

La distribución geométrica da la probabilidad de en número de eventos (x) de un tipo (B) que hay que esperar hasta obtener un evento del otro tipo (A) de probabilidad p

$$P(X = x) = f(x) = (1 - p)^x p,$$

$x = 0, 1, 2, \ldots$
Su media y varianza muestral son
$E(X) = \mu = \frac{1-p}{p}$ y $V(X) = \sigma^2 = \frac{1-p}{p^2}$

Distribución Geométrica

Cuál es la probabilidad de observar fotos con 1 pixel correcto antes del primer error, 2 pixels correctos antes del primer error, ... 100 pixeles correctos (sin error)? Pr(x=2), Pr(x=3), ... Pr(x=100)?

Tenemos que enviar muchas fotos de 100 pixeles (p.ej. 1000) y contar cuantos ceros hay antes del primer 1.

Distribución Geométrica

Foto

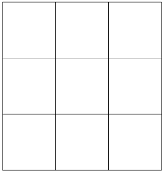


Foto transmitida 1 Foto transmitida 2 Foto transmitida 3

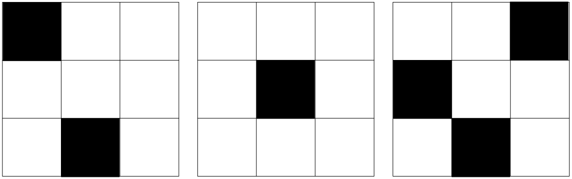
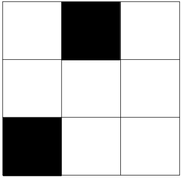


Foto transmitida N



……………..

Contamos el número total de pixeles correctos antes primer error

X = (x1, x2, ….xN) = (0, 4, 2, …. , 1) : muestra de “Nfotos”

Distribución Geométrica

Hagamos una foto:

- hagamos 100 ensayos de Bernoulli (p=0.2)

```
> foto <- sample(c(0,1),100, replace=TRUE, prob=fk)
> foto
  [1] 0 0 1 0 1  ...
```

Ahora identificemos cuantos ceros aparecieron antes del primer 1 (en mi caso hubo 2)

Distribución Geométrica

Con la función **which** tenemos las posiciones de un vector que satisfacen una condición (Qué elemento en foto es igual a 1? usa ==)

```
> which(foto==1)
 [1]  3  5  ...
```

Si asignamos la variable **ones** al resultado de **which**, Cuál sería la posición del último cero?

Distribución Geométrica

Pongámolo todo junto.
Esta es uan foto con su número de ceros antes del primer 1, asignado a la variable **lastzero**

```
> foto <- sample(c(0,1),100, replace=TRUE, prob=fk)
> ones <- which(foto==1)
> firstone <- ones[1]
> lastzero <- firstone-1
> lastzero
[1] 5
```

Esta es sólo una foto, queremos enviar 1000 fotos y hacer un histograma?

Distribución Geométrica

1. Hay que hacer una función llamémosla (**enviar**)

```
> enviar <- function(x)
          {
             foto <- sample(c(0,1),100,
                       replace=TRUE, prob=fk)
             ones <- which(foto==1)
             firstone <- ones[1]
             lastzero <- firstone-1
             lastzero
          }

> enviar()
[1] 1
```

esta función crea una foto la enva y cuenta el número de pixels bien trasmitidos hasta el primer error

Distribución Geométrica

2. Hagamos un bucle de 1000 iteraciones de la función (**sapply**)

```
> Milfotos <- sapply(rep(1,1000), enviar)
> head(Milfotos)
>  head(Milfotos)
[1]  2 16  2  0  1  5
> mean(Milfotos)
[1] 4.136
> sd(Milfotos)^2
[1] 19.31482
```

La media y la varianza de la distribución geométrica son

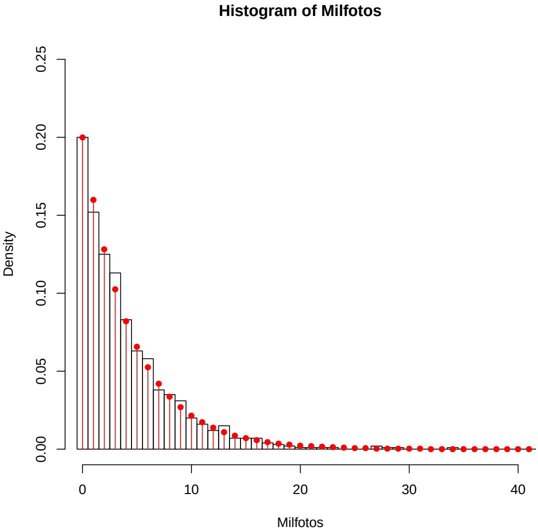
$E(X) = \frac{1-p}{p} = 4$ y $V(X) = \frac{1-p}{p^2} = 20$

Distribución Geométrica

```
> hist(Milfotos,breaks=seq(-0.5,100.5),
freq=FALSE, ylim=c(0,0.25), xlim=c(0,40))

> geom <- dgeom(x, prob=0.2)

> points(x, geom, pch= 16, col="red")

> for(i in 1:101)
{lines(c(x[i], x[i]), c(0, geom[i]),
  col="red")}
```

Distribución Geométrica

puedes conseguir?



Más material adicional:

Distribución Binomial Negativa

Foto transmitida 1

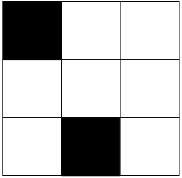


Foto transmitida 2

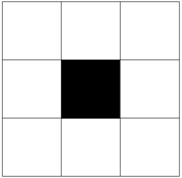
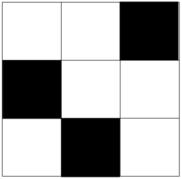


Foto transmitida 3



Foto

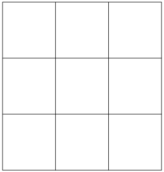
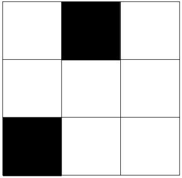


Foto transmitida N



………………..

Contamos el número total de pixeles correctos antes segundo error

X = (x1, x2, ….xN) = (0, 6, 2, …. , 5) : muestra de “Nfotos”

Distribución Binomial Negativa

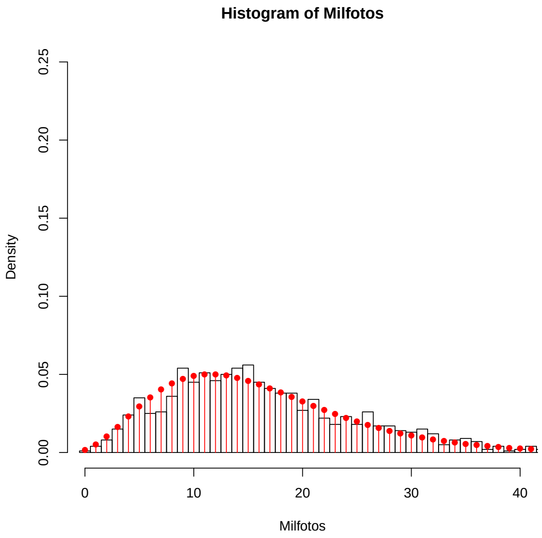
Enviemos una foto y contemos en qué pixel aparece antes del cuarto error (r=4). Imagina que no queremos fotos con mas de 4 errores. El número de pixels antes del cuarto error se distribuye bajo una binomial gegativa NegBin(x; r,p) o en R dnbinom(x, size=r, prob=p).

Enviemos una foto y veamos que pixel tiene el cuarto error

```
> fotos <- sample(c(0,1),100, replace=TRUE,prob=fk)
> ones <- which(fotos==1)
> fourthone <- ones[4]
> lastzero <- fourthone-1
#le quitamos los primeros tres errores
> goodpixels <- lastzero-3
[1] 13
```

Distribución Geométrica

puedes conseguir?



Distribución Binomial Negativa

```
enviar <- function(x)
 {
 foto <- sample(c(0,1),100,
 replace=TRUE, prob=fk)
 ones <- which(foto==1)
 firstone <- ones[4]
 lastzero <- firstone-1
 goodpixels <- lastzero-3
 goodpixels
 }

Milfotos <- sapply(rep(1,1000), enviar)

hist(Milfotos,breaks=seq(-0.5,100.5),
 freq=FALSE, ylim=c(0,0.25), xlim=c(0,40))

nb <- dnbinom(x, size=4, prob=0.2)

#continua...
```

Distribución Binomial Negativa

```
points(x, nb, pch= 16, col="red")

for(i in 1:101)
 {lines(c(x[i], x[i]), c(0, nb[i]),
   col="red")}
```