Probabilidades

Juan Manuel Morales

2015-09-05

Objetivos:

- Familiarizarse con distintas distribuciones de probabilidades y con la manera de usarlas en R.
- Simular procesos estocástcos.
- ver https://sites.google.com/site/modelosydatos/Bestiario_sp.pdf

---

Las distribuciones de probabilidad en general son ecuaciones (pueden ser tablas también) que relacionan el resultado de un experimento o procedimiento de muestreo con su probabilidad de ocurrencia. R tiene funciones para todas las distribuciones de probabilidad estándares, y para cada una de estas distribuciones tenemos funciones para:

- generar valores,
- calcular probabilidades,
- probabilidades acumuladas y
- cuantiles

Estas funciones comienzan con las letras `r`, `d` ,`p` y `q` respectivamente. Por ejemplo, para la distribución de Possion tenemos: `rpois`, `dpois`, `ppois`, y `qpois`.

En muchos casos es útil poder generar muestras de una distribución en particular. Asumimos que estas muestras generadas en la computadora son una "muestra aleatoria" pero en realidad provienen de un generador de números aleatorios por lo que es más correcto decir que son "pseudo-aleatorios". Un aspecto importante, sobre todo pensando en la reproducibilidad de nuestro trabajo, es que si iniciamos al generador de números aleatorios con un valor determinado, la secuencia de números pseudo-aleatorios se va a repetir y por lo tanto podemos reproducir exactamente una simulación estocástica. Existen muchos algoritmos para generar números pseudo-aleatorios, pero en general no nos metemos demasiado en estos detalles y confiamos en que R sabe lo que hace.

En R usamos `set.seed(12345)` para inicializar el generador de números aleatorios en 12345. El número que le ponemos a `set.seed` es arbitrario pero debe ser un número entero.

ejemplo: simulamos una muestra de 10 valores de una distribución de Poisson

```
set.seed(12345)
rpois(n=10,lambda=1.2)
```

```
##  [1] 2 2 2 3 1 0 1 1 2 4
```

Nuestros scripts pueden ser más fáciles de leer y modificar si definimos variables por fuera de las funciones. Por ejemplo:

```
n <- 100
lambda <- 1.2
```

Ahora simulamos datos y vemos la frecuencia en un histograma

```
y <- rpois(n = n, lambda = lambda)
hist(y, xlab="Valores", main="")
```

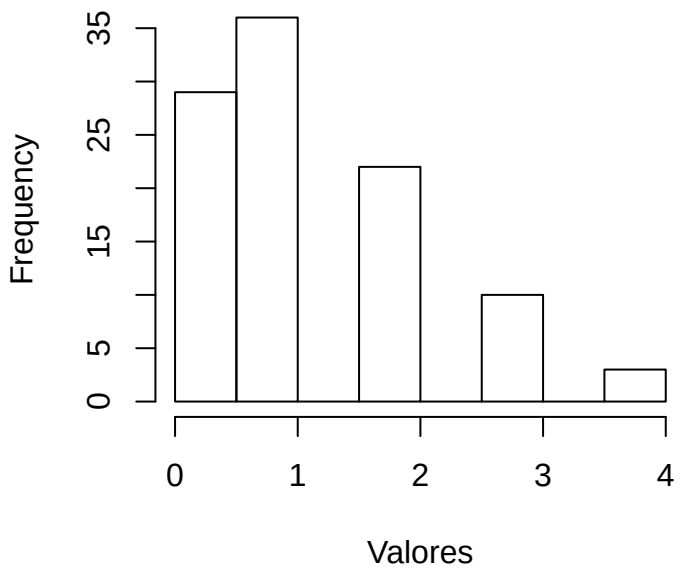

Figure 1: Histograma de datos simulados de una Poisson con $\lambda = 1.2$

Una vez que tenemos "datos" como estos, podemos ver las frecuencias relativas y compararlas con la distribución teórica usando la función `dpois`.

```
f <- factor(y,levels=0:max(y))
obsprobs <- table(f)/n
plot(obsprobs, xlab="Valores", ylab="Proporción")
tprobs <- dpois(x = 0:max(y), lambda = lambda)
points(0:max(y), tprobs, pch=16, col=2)
```

Otra función útil es la Función de Distribución, que nos da la probabilidad de encontrar un valor menor o igual a un valor dado. Por ejemplo, la probabilidad de obtener $y <= 3$ con $\lambda = 1.2$ es:

```
ppois(3, lambda=lambda)
```

```
## [1] 0.966231
```

Si queremos saber la probabilidad de obtener un valor *mayor* a 3 hacemos

```
1 - ppois(3, lambda=lambda)
```

```
## [1] 0.03376897
```

Para ver la probabilidad de un valor en particular, por ej $y = 3$:

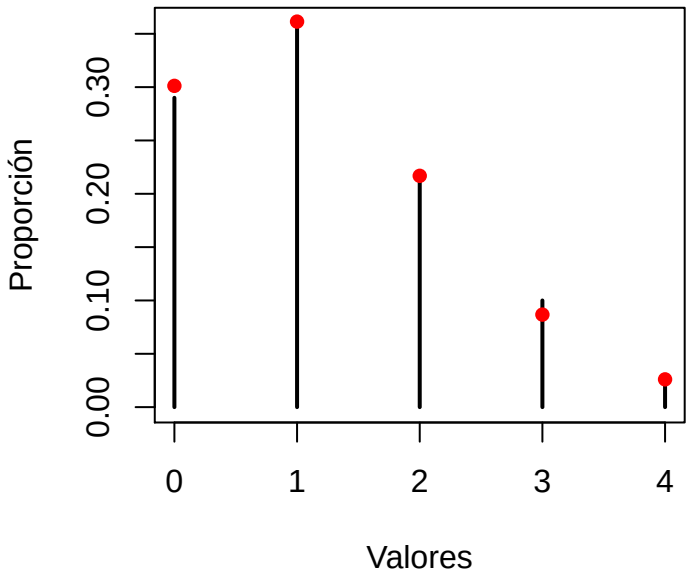

Figure 2: Distribución empírica y teórica para Poisson con $\lambda = 1.2$

```
ppois(3,lambda=lambda) - ppois(2, lambda=lambda)
```

```
## [1] 0.08674393
```

¿Por qué tiene sentido hacer lo de arriba? Pensadlo...

Podemos corroborar este resultado usando `dpois`

```
dpois(3,lambda=lambda)
```

```
## [1] 0.08674393
```

El equivalente empírico de la función acumulada es la funciónn `ecdf`:

```
eF <- ecdf(y)
plot(eF, xlab="Valores", ylab="Función de Probabilidad Empírica", main="")
lines( 0:6, ppois(0:6, lambda=1), type="s", col=2)
```

Por último, la función cuantil `qpois` es la inversa de la función de distribución y para un valor de probabilidad acumulada nos devuelve el valor de la variable

```
qpois(0.95, lambda=lambda)
```

```
## [1] 3
```

La función `qpois` sirve también para calcular intervalos que contienen un porcentaje de los valores de la distribución. Por ejemplo, el 95% de los valores están entre `qpois(c(0.025,0.975) , lambda)`. El equivalente empírico es la función `quantile`

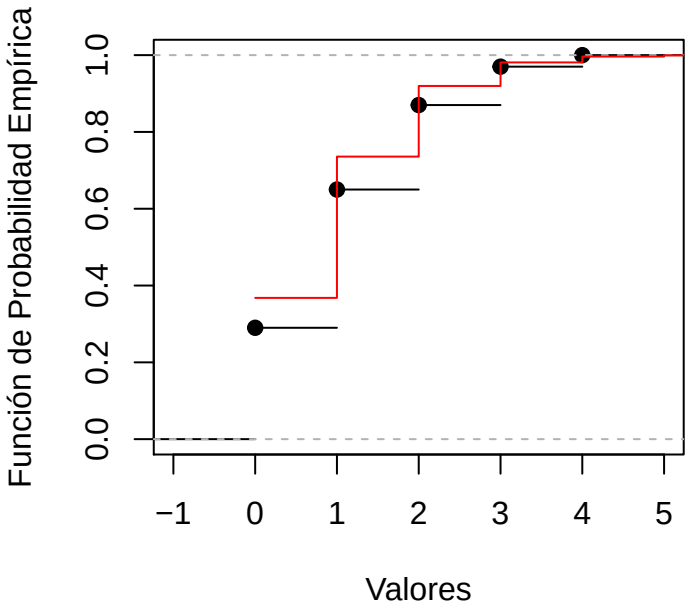

Figure 3: Acumulada empírica y teórica para Poisson con $\lambda = 1.2$

```
qpois(c(0.025,0.975) , lambda)
```

```
## [1] 0 4
```

```
quantile(y,  probs = c(0.025,0.975))
```

```
##  2.5% 97.5%
## 0.000 3.525
```

Ejercicios:

1. Asumiendo una distribución de Poisson:

- ¿cuál es la probabilidad teórica de obtener $y = 0$ para $\lambda = 1$?
- ¿cuál es la probabilidad teórica de $y > 2$ para $\lambda = 1$?
- comparar la probabilidad de $y = 0$ para $\lambda = 1$ de la distribución teórica con la obtenida empíricamente con muestras de 10, 100, 1000 y 10000 observaciones
- comparar las diferencias entre el valor esperado de y para la distribución teórica con los valores empíricos de muestras de 10, 100, 1000 y 10000 observaciones
- lo mismo para los intervalos de 95%

2. ¿cómo cambia la forma de la distribución de Poisson a medida que cambia λ? (pueden ver esto simulando datos y haciendo histogramas o graficando las probabilidades teóricas)

Distribuciones contínuas

Podemos hacer algo parecido a lo que hicimos con la distribución de Poisson para el arquetipo de las distribuciones contínuas:

```
set.seed(1234)
n <- 1000
mu <- 0
sigma <- 1

y <- rnorm(n, mean = mu, sd = sigma)

hist(y, breaks = 40, freq = FALSE, main="")
# podemos agregar sobre este histograma una estimación de densidades
lines(density(y), lwd = 2, lty=3, col="darkgrey")
xvec <- seq(min(y)-0.5,max(y)+0.5, by=0.1) # secuencia de valores de referencia
lines(xvec, dnorm(xvec, mean = mu, sd = sigma), lwd = 2)
```

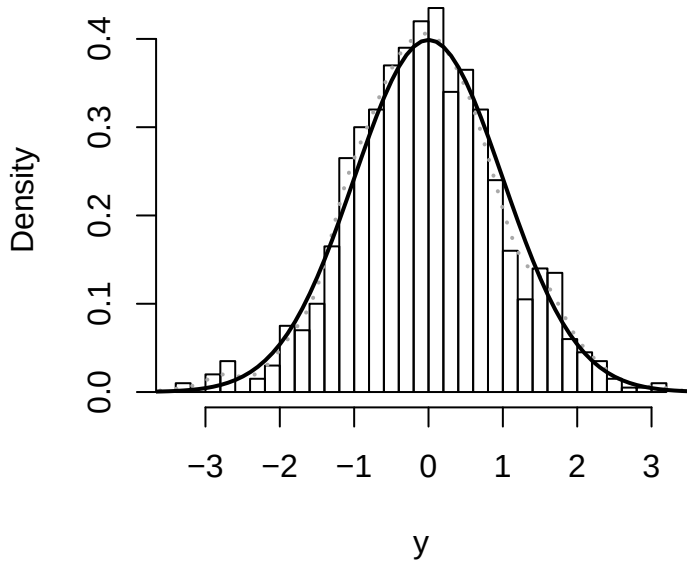

Figure 4: Histograma de datos simulados, densidad empírica (en negro) y teórica (en rojo) de una distribución Normal con $\mu = 0$ y $\sigma = 1$

¿Cuál es la probabilidad de encontrar un valor "cercano" a cero? ¿por qué "cercano" y no igual?

```
dnorm(0, mean=mu, sd=sigma)
```

```
## [1] 0.3989423
```

¿Cuál es la probabilidad de $y > 1.96$?

```
pnorm(1.96, mean = mu, sd = sigma, lower.tail = FALSE)
```

```
## [1] 0.0249979
```

Y el intervalo de 95 porciento?

```
qnorm(c(0.025, 0.975), mean = mu, sd = sigma)
```

```
## [1] -1.959964  1.959964
```

Ejercicios:

1. Estimar las mismas cantidades usando estimaciones empíricas.
2. ¿Cómo cambia la forma de la distribución cuando cambia σ?
3. Otras distribuciones. Ver otras distribuciones del bestiario para familiarizarse con las formas que éstas pueden tener y el tipo de datos que pueden llegar a representar.

Bootstrap

Objetivo: Revisar los conceptos básicos de remuestreo de datos

Ejemplo básico:

Estimación de la media e Intervalo de Confianza de datos de una distribucion Gamma. Para remuestrear datos usamos la función `sample`.

```
datos <- rgamma(50,shape = 0.8,rate=1)   # simulo 50 datos
n <- length(datos)                       # número de observaciones
B <- 2000                                # número de muestras de bootstrap
theta_b <- numeric(B)                    # vector donde se van a guardar los resultados

for (i in 1:B){
    y <- sample(datos,size=n,replace=T) # bootstrap sample
    theta_b[i] <- mean(y)               # en gral hacemos algo más interesante aquí...
}

mb <- mean(theta_b)
mb
```

```
## [1] 0.5273573
```

```
se <- sqrt(sum((theta_b-mb)^2)/ (B-1)) # esto es el desvío estándar de un estadistico muestreal
                                       # lo podemos calcular directamente como sd(theta_b)
se
```

```
## [1] 0.07641065
```

```
ci = quantile(theta_b,probs=c(0.025,0.975)) # intervalo que contiene el 95% de los valores
ci
```

```
##      2.5%     97.5%
## 0.3822975 0.6835077
```

Ejemplo: Poner a prueba un "Correlated Random Walk"

Cargamos datos de localizaciones de elk reintroducidos y graficamos trayectorias de movimiento y el desplazamiento cuadrado

```
setwd("~/Dropbox/ME")
elk <- read.table("elkGPS1.txt", header = T)
op<-par(mfrow=c(1,2))
plot(elk$Easting, elk$Northing, type="o", asp=1, xlab="easting", ylab="northing")
R2 <- (elk$Easting/1000 - elk$Easting[1]/1000)^2 + (elk$Northing/1000 - elk$Northing[1]/1000)^2
plot(R2, type="l", xlab="t (días)")
```

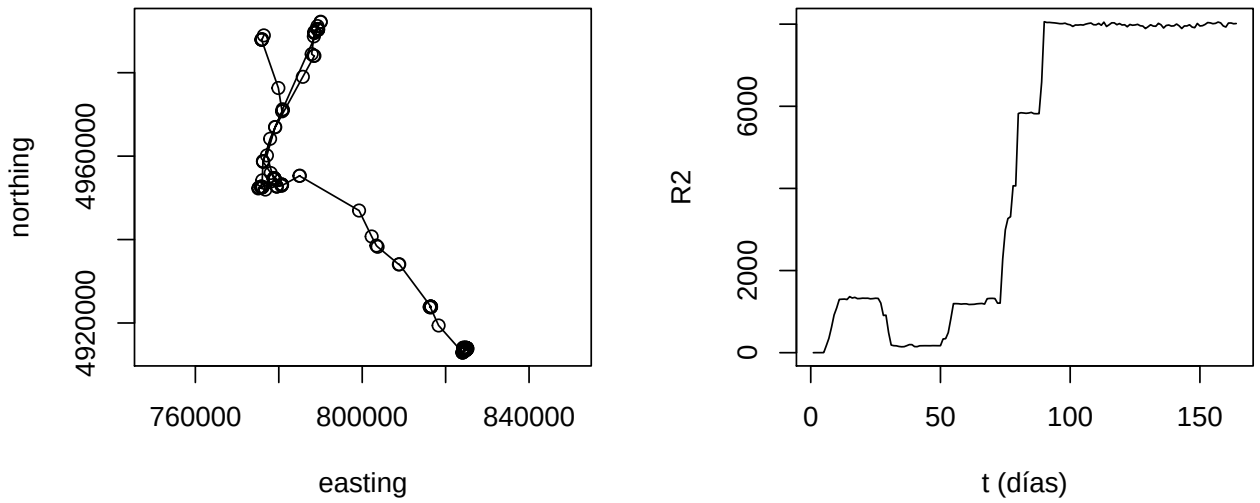

Figure 5: trayectoria y desplazamiento cuadrado

```
par(op)
```

¿Les parece que esto podría haber sido generado por un CRW? Para responder a esta pregunta, vamos a obtener los percentiles de R2 bajo el supuesto de que la trayectoria se genera con un CRW

```
# definimos los "pasos" y "giros" que ya están calculados en el set de datos
steps <- elk$km_day
turns <- elk$turns
sts <- steps[2:(length(steps)-1)]  #elimino el primer y ultimo dato que son NA´s
trns <- turns[2:(length(steps)-1)] #elimino el primer y ultimo dato que son NA´s

# simulamos una cantidad de CRWs
B <- 1000
nobs <- length(sts)
R2B <- matrix(NA,B,nobs)

for(j in 1:B){
  x <- numeric(nobs)
  y <- numeric(nobs)
  compass <- numeric(nobs)

  # simulamos CRW
```

```r
  for(i in 2:length(sts)){
    compass[i] <- compass[i-1] + sample(trns, size=1, replace=T)
    steplength <- sample(sts, size=1)
    x[i] <- x[i-1] + steplength*cos(compass[i])
    y[i] <- y[i-1] + steplength*sin(compass[i])
  }
  R2B[j,] <- (x-x[1])^2 + (y-y[1])^2
}

# calcular percentiles
per <- matrix(NA, nobs,2)
for(i in 1:nobs){
per[i,] <- quantile(R2B[,i], probs=c(0.025,0.975))
}

plot(R2, type="o", ylim=c(0, max(R2B)), pch=16, xlab="t (días)", ylab="Desplazamiento al Cuadrado")
for(i in 1:B) lines(R2B[i,], col="gray")
lines(per[,1], lwd=2)
lines(per[,2], lwd=2)
lines(R2, type="o", pch=16)
```

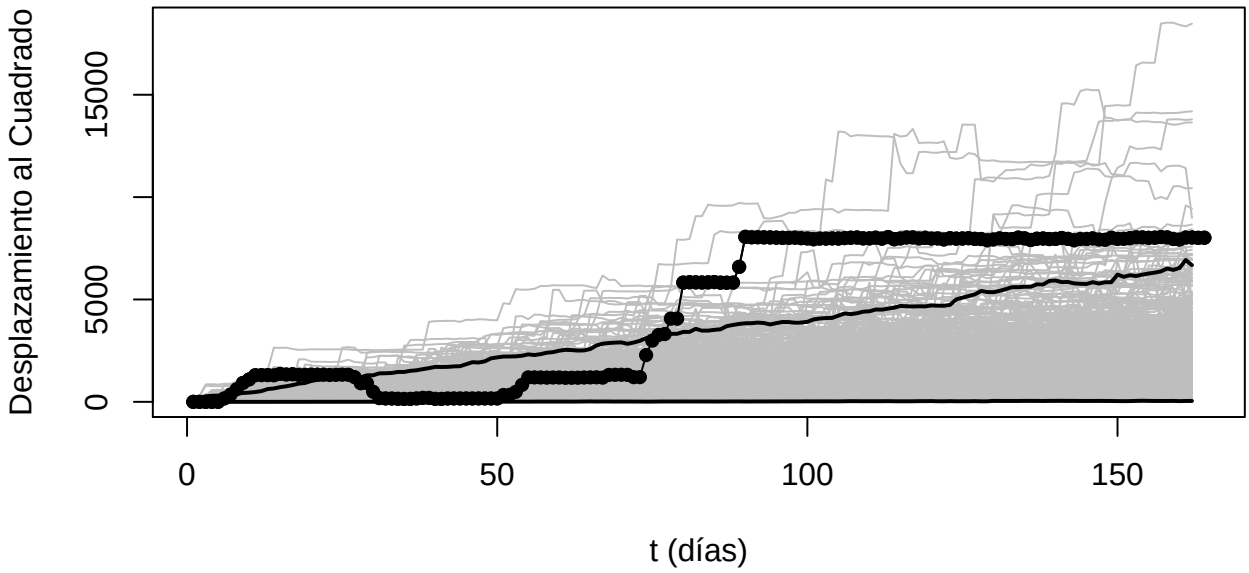

Figure 6: Desplazamiento cuadrado observado (línea con puntos) y simulado (líneas grises). Las líneas sólidas marcan los percentiles de 95%.