La Ruina del Jugador

Rubén Martín

Resumen

Imaginemos la siguiente situación: Un jugador tiene un capital de 20 euros y cada instante de tiempo apuesta un euro al lanzamiento de una moneda, ganando un euro si sale cara y perdiendo el euro apostado si sale cruz. Consideramos la variable aleatoria discreta T que nos indica el instante de tiempo en que el jugador se arruina por primera vez. Deberemos por tanto crear una función en R que permita simular y representar gráficamente la "ruina del jugador", así como presentar el desarrollo mediante un documento de LaTeX.

Índice

1. La Teoría

Podemos extraer leyendo el ejercicio, que cada lanzamiento de la moneda se comporta como un *experimento Bernouilli*. Este tipo de variables aleatorias [1] se distribuyen de la siguiente forma:

$$P[X = 1] = p$$

$$P[X = 0] = 1 - p = q$$

Por tanto los lanzamientos consecutivos de la moneda se distribuirán según una *distribución binomial*:

$$P[X = k] = \binom{n}{k} p^k (1-p)^{n-k}$$

La esperanza de dicha variable, $E[X] = np$, nos indica que si seguimos lanzando la moneda indefinidamente ganaremos las mismas veces que perderemos. Sin embargo tenemos la posibilidad de perder todo el dinero antes de que esto ocurra.

2. Qué Ocurre

En la tabla que se muestra a continuación se puede observar el comportamiento del capital del jugador una vez que comienza a jugar.

Lanzamientos totales	Lanzamientos ganados	Capital total
1	0	19
1	1	21
2	0	18
2	1	20
2	2	22
⋮	⋮	⋮
19	0	1
20	0	0
20	20	40

Así, si el jugador tiene la "mala suerte" de perder todos los lanzamientos se quedaría sin dinero en el instante 20. Sin embargo, si en ese mismo momento hubiese ganado todos los lanzamientos el jugador tendría 40 euros en su poder.

3. El Problema

3.1. Planteamiento

Para modelizar la situación anteriormente descrita deberemos tener en cuenta:

- La función tendrá un único argumento, el capital inicial. En el caso explicado es de 20 euros, aunque admitiremos que este puede ser otro valor.
- La moneda se supone no trucada (probabilidad de cara y cruz idéntica, 1/2)
- El juego termina en el instante en que el capital de que dispone el jugador es 0 (ruina)
- La función debe devolver los resultados en un gráfico

A continuación se desarrollará paso a paso la creación de la función.

3.2. Creación de la Moneda

Para la creación de la moneda usaremos la función ya implementa en R `sample` de la siguiente forma:

```
moneda = sample(c(1,-1), 1, prob = c(0.5, 0.5))
```

Tenemos pues un "lanzamiento" con un vector de valores (1,-1) para cara y cruz respectivamente. Ambas posibilidades tienen una probabilidad de ocurrir del 50 %, tal y como se observa en el último argumento.

3.3. Creación del Bucle Principal

Una vez que tenemos la moneda pasaremos a esbozar qué es necesario en el interior de la función. Se puede suponer leyendo el enunciado del problema que el jugador sigue realizando lanzamientos de moneda hasta que se queda sin dinero, por lo que nos hace falta un bucle que repita esos lanzamientos hasta que no tenga con qué jugar.

En conclusión: la función tiene que tener un parámetro que sea el dinero con el que parte el jugador y un bucle que modifique el valor de esa cantidad dependiendo de si obtiene cara o cruz.

Decidí utilizar el bucle `while`, ya que no sé el momento exacto en el que el jugador perderá.

```
ruinaDelJugador = function(capitalInicial)
{
  while(capitalInicial !=0){
    moneda = sample(c(1,-1),1, replace = TRUE, prob = c(0.5,0.5));
    capitalInicial = capitalInicial + moneda;
  }
}
```

3.4. Dónde almacenar los datos

Para representar los datos hará falta un vector que los almacene. A ese vector lo nombraremos como `resultados`, y varía de la siguiente forma:

```
resultados <- c(resultados,capitalInicial)
```

Pero tiene que tomar un valor inicial, que es:

```
resultados <- c(capitalInicial)
```

Resumiendo lo que llevamos hasta ahora:

```
ruinaDelJugador = function(capitalInicial)
{
  resultados <- c(capitalInicial);
  while(capitalInicial !=0){
    moneda = sample(c(1,-1),1, replace = TRUE, prob = c(0.5,0.5));
    capitalInicial = capitalInicial + moneda;
    resultados <- c(resultados,capitalInicial);
  }
}
```

3.5. Representación gráfica

Por último se pide que representemos esos valores. Esto es bastante sencillo puesto que solo es necesario utilizar la función `plot`.

```
plot(resultados,type = "l",xlab = "Intentos",ylab = "Capital")
```

Que añadiéndolo a la función quedaría:

```
ruinaDelJugador = function(capitalInicial)
{
  resultados <- c(capitalInicial);
  while(capitalInicial !=0){
    moneda = sample(c(1,-1),1, replace = TRUE, prob = c(0.5,0.5));
    capitalInicial = capitalInicial + moneda;
    resultados <- c(resultados,capitalInicial);
 }
  plot(resultados,type = "l",xlab = "Intentos",ylab = "Capital")
}
```

3.6. Últimos ajustes

Para clarificar aún más la gráfica le añadiremos dos líneas horizontales que señalen el valor del dinero con el que parte el jugador y el valor 0. Para ello utilizaremos la función `abline` de la siguiente forma:

```
capitalIn <- capitalInicial;

...

abline(h = capitalIn, lty = 3, col = 4);
abline(h = 0, lty = 3, col = 2);
```

El valor de `capitalIn` indica el valor que ha introducido el jugador, y debe de escribirse antes de modificar el valor de `capitalInicial` en el bucle `while`.

Del mismo modo añadimos un mensaje que indique en qué momento exacto el jugador pierde todo su dinero utilizando el siguiente código:

```
print("El jugador se arruina en el momento: ");
print(length(resultados));
```

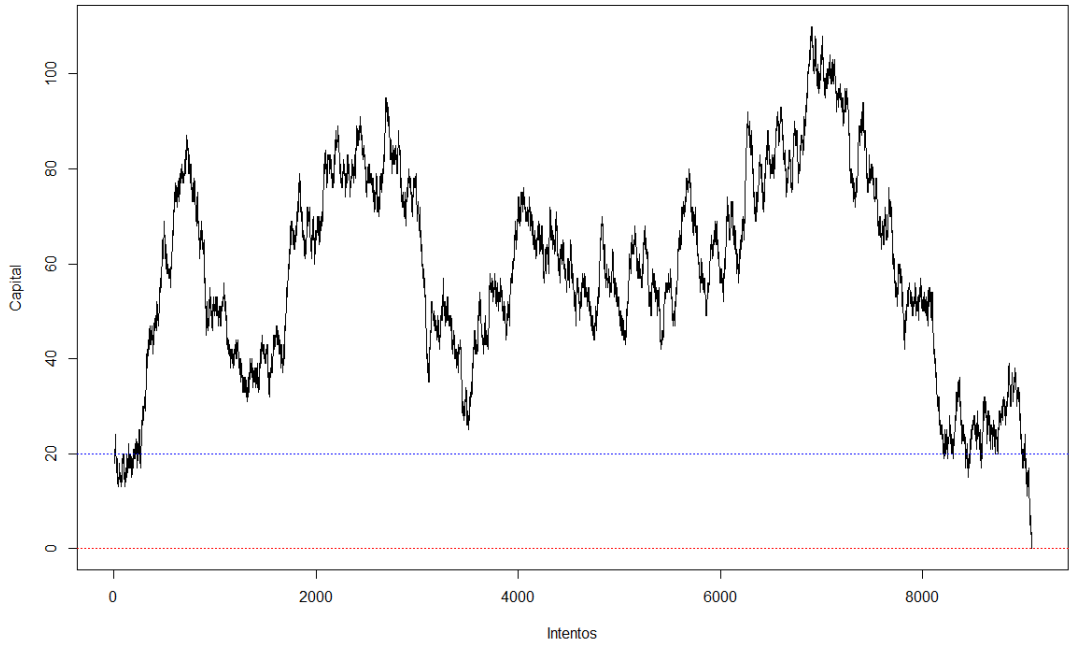
3.7. Conclusiones

Para terminar uniremos todo el código utilizado y añadiremos un par de ejemplos con sus respectivas gráficas:

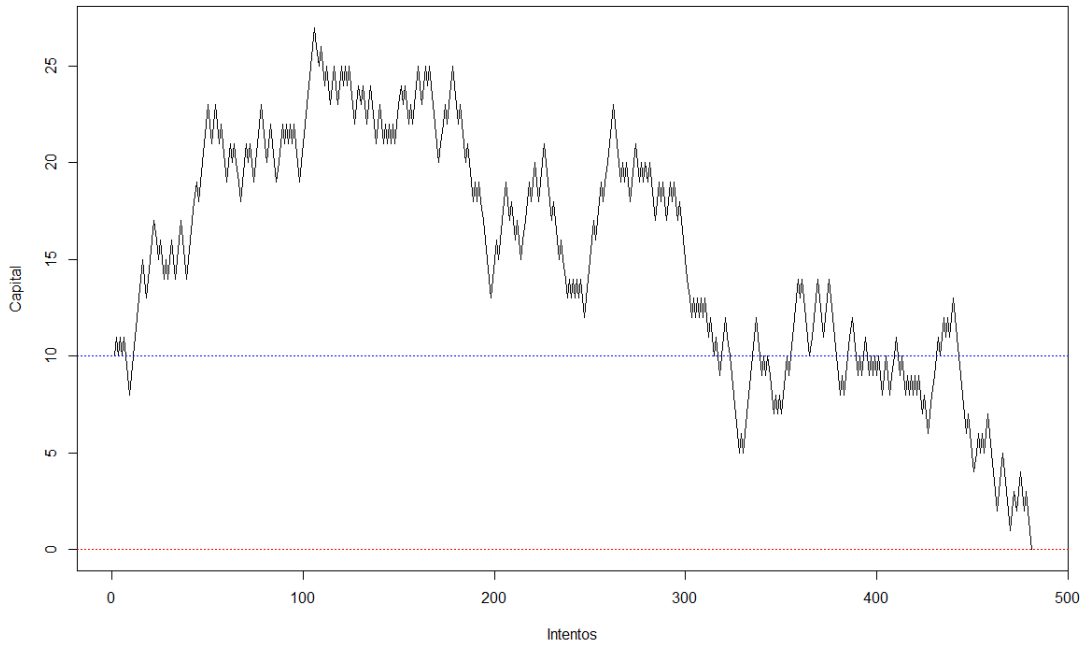
```
ruinaDelJugador = function(capitalInicial)
{
 capitalIn <- capitalInicial;
 resultados <- c(capitalInicial);

 while(capitalInicial !=0){
 moneda = sample(c(1,-1),1, replace = TRUE, prob = c(0.5,0.5));
 capitalInicial = capitalInicial + moneda;
 resultados <- c(resultados,capitalInicial);
 }

 plot(resultados,type = "l",xlab = "Intentos",ylab = "Capital")
 abline(h = capitalIn, lty = 3, col = 4)
 abline(h = 0, lty = 3, col = 2)

 print("El jugador se arruina en el momento: ");
 print(length(resultados));
}
```

3.8. Ejemplos

```
> ruinaDelJugador(20)
[1] "El jugador se arruina en el momento: "
[1] 8682.
```




```
> ruinaDelJugador(10)
[1] "El jugador se arruina en el momento: "
[1] 482.
```



4. ANEXO

Para poder utilizar Latex [2] correctamente fue necesario realizar un curso con Darwin Eventur, en el que participaron Ángel Pablo Hinojosa y Renato Ramírez Rivero. En dicho curso se expusieron las ideas principales del funcionamiento de LaTeX y Git.

Ambas partes fueron muy interesantes y entretenidas, pues se intentó que tanto la gente que tuviera conocimientos como la que no aprendiera lo máximo posible.

Referencias

[1] del Pino, A. M. A. *Curso y Ejercicios de Cálculo de Probabilidades*. Copias Coca, 2000.

[2] Tobias Oetiker, Hubert Partl, I. H. E. S. *La introducción no-tan-corta a LaTeX 2E*. Free Software Foundation, 2014.