

# Simulación

Ximena Fernández

21 de abril de 2020

## 1 Simulaciones en R

Calcular la probabilidad de al tirar dos dados la suma me de par

$$\Omega = \{(d_1, d_2) : d_i \in \{1, 2, 3, 4, 5, 6\}\}$$

$$E = \{(d_1, d_2) \in \Omega : d_1 + d_2 \equiv 0 \pmod{2}\}$$

Simular es repetir el experimento muchas veces y calcular la frecuencia relativa del evento.

```
[61]: set.seed(1234)
      n_rep = 10000
      mean(replicate(n_rep, sum(sample(6, 2, replace = TRUE))%%2==0))
```

0.4962

Simular NO es calcular con la computadora por fuerza bruta casos favorables/casos totales

```
[43]: casos_totales = 0
      casos_favorables = 0

      for (dado1 in 1:6)
      {
        for (dado2 in 1:6)
        {
          casos_totales = casos_totales + 1
          if ((dado1 + dado2)%2==0)
          {
            casos_favorables = casos_favorables + 1
          }
        }
      }
      }
```

```
[44]: casos_favorables/casos_totales
```

0.5

## 1.1 Derangements

```
[62]: n<-30
      sample(1:n)
```

```
1. 2 2. 26 3. 21 4. 17 5. 19 6. 15 7. 14 8. 13 9. 28 10. 6 11. 7 12. 9 13. 1 14. 30 15. 8 16. 16 17. 5 18. 23
19. 20 20. 18 21. 27 22. 24 23. 12 24. 25 25. 22 26. 10 27. 29 28. 4 29. 11 30. 3
```

```
[63]: fixed_points <- function(vector)
      {
        fixed_points = 0
        for (i in 1:length(vector))
        {
          if (vector[i]==i)
          {
            fixed_points <- fixed_points + 1
          }
        }
        return (fixed_points)
      }
```

```
[64]: set.seed(1234)

      for (n_rep in c(10, 100, 500, 1000, 5000, 10000, 50000, 100000, 500000, 1000000,
        ↪5000000))
      {
        print(mean(replicate(n_rep, fixed_points(sample(1:n))==0)))
      }
```

```
[1] 0.1
[1] 0.35
[1] 0.318
[1] 0.36
[1] 0.3674
[1] 0.3661
[1] 0.368
[1] 0.37037
[1] 0.367832
[1] 0.367247
[1] 0.3679582
```

Mejor guardamos las distintas simulaciones en un vector.

```
[65]: set.seed(1234)
      n_rep = c(10, 100, 500, 1000, 5000, 10000, 50000, 100000, 500000, 1000000,
        ↪5000000)

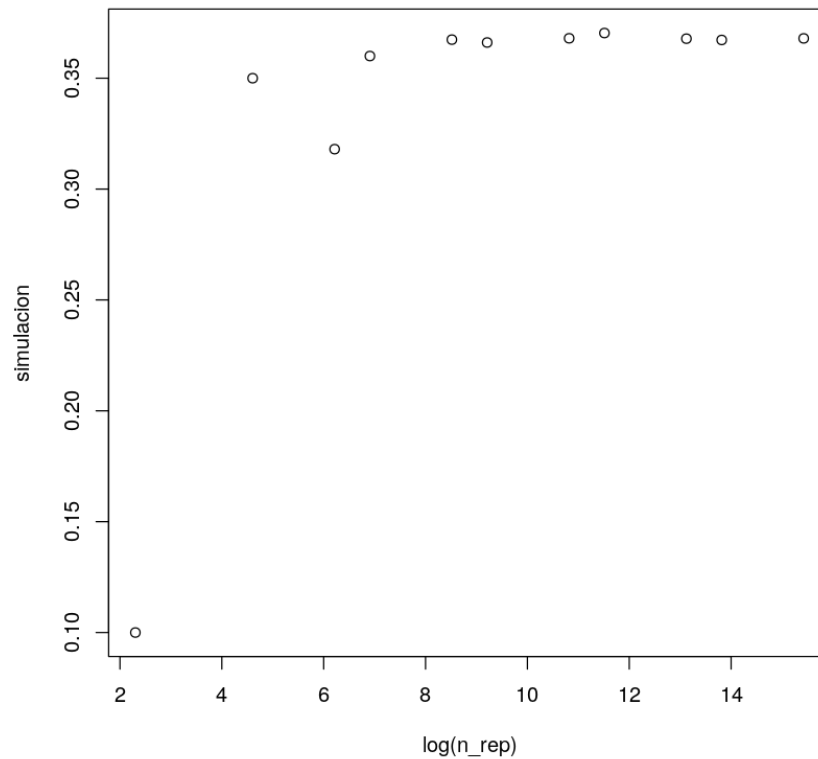
      simulacion <- rep(NA, length(n_rep))
```

```
for (i in 1:length(n_rep))
{
  simulacion[i] <- (mean(replicate(n_rep[i],fixed_points(sample(1:n))==0)))
}
```

```
[66]: simulacion
```

```
1. 0.1 2. 0.35 3. 0.318 4. 0.36 5. 0.3674 6. 0.3661 7. 0.368 8. 0.37037 9. 0.367832 10. 0.367247
11. 0.3679582
```

```
[67]: plot(log(n_rep), simulacion)
```



La cantidad de **derangements** de  $n$  elementos (denotado por  $!n$ ) es sospechosamente parecido para cualquier valor de  $n$ :

$$!n \simeq \frac{1}{e}$$

```
[137]: 1/exp(1)
```

```
0.367879441171442
```

Acá pueden encontrar una explicación divertida de la respuesta, por el genio de **Numberphile**

<https://www.youtube.com/watch?v=pbXg5EI5t4c>

<https://www.youtube.com/watch?v=qYAWjIVY7Zw>