

Nombre: Alan Rodrigo López López

0.1 Código del dado en python

Se realizó un programa en Python para obtener un vector de 100000 elementos de números aleatorios entre 1 y n , el cuál era especificado por el usuario. Con esto, se obtienen 100000 números aleatorios los cuales se guardan en un archivo .csv para ser llevados a R. El código que realiza eso se muestra en la Figura 2.

```
import numpy as np
import pandas as pd

def n_sided_die(n, rolls = 100000):
    np.random.seed(np.random.randint(0,rolls)) # Cambia de semilla
    return np.random.randint(1,n+1,rolls) # obtiene un vector de 100000 elementos con valores aleatorios entre 1 y n

number_faces = int(input('Number of faces: '))
# convierte a dataframe el vector de números
df = pd.DataFrame(n_sided_die(number_faces), columns = ['data']).to_csv('n_sided_die.csv', index=False)
```

Figure 1: Código en Python para obtener el dado

0.2 Código para obtener el histograma en R

Una vez que se tiene el archivo .csv, se lee en R y se grafica un histograma para observar con qué frecuencia van apareciendo los números entre 1 y n . El código en R se muestra en la Figura ??.

```
table <- read.csv('n_sided_die.csv', header = TRUE, sep = ",")
head(table)
h <- hist(table$data, main = 'Histograma',
          xlab = 'Número de cara del dado',
          ylab = 'Frecuencia',
          col = '#99ccff',
          border = '#6699ff',
          label = FALSE)
text(h$mids, h$counts+ (0.02 * max(h$counts)), labels = h$counts, cex = 0.8)
```

Figure 2: Código en R para graficar un histograma de los valores obtenidos en el dado

0.3 Resultados

Se muestra el resultado para dos dados, el primero tiene $n = 5$ caras y el segundo $n = 30$. Se observa que para ambos casos tiende a una distribución uniforme.

Dado de 5 caras:

El histograma para este dado fue el siguiente.

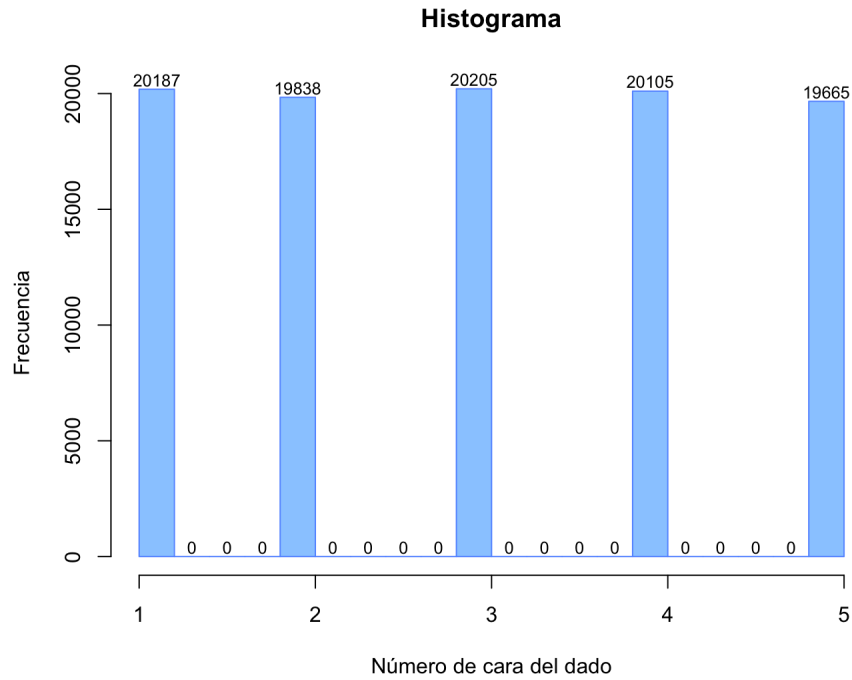


Figure 3: Histograma para el dado de 5 caras

Dado de 30 caras:

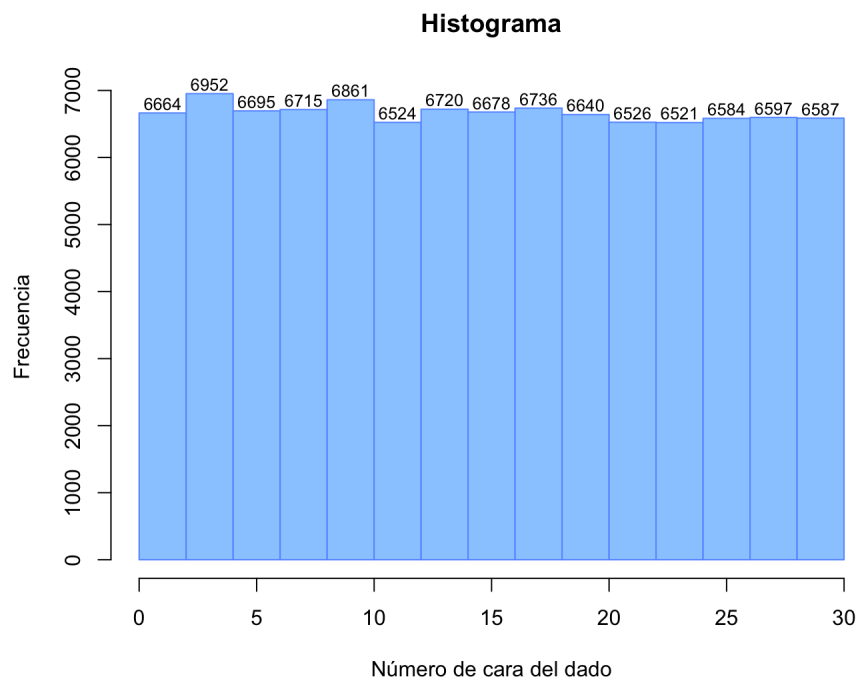


Figure 4: Histograma para el dado de 30 caras