

```
#libraries
library(tibble)
library(psych)
library(dplyr)
library(MASS)
```

Experiência: Lançamento de um dado de 6 faces e equilibrado, duas vezes

```
dado <- c(1,2,3,4,5,6)
omega <- expand.grid(dado1= dado, dado2= dado)
omega$prob <- 1/nrow(omega)
tibble(omega)
```

```
## # A tibble: 36 x 3
##   dado1 dado2  prob
##   <dbl> <dbl> <dbl>
## 1     1     1  0.0278
## 2     2     1  0.0278
## 3     3     1  0.0278
## 4     4     1  0.0278
## 5     5     1  0.0278
## 6     6     1  0.0278
## 7     1     2  0.0278
## 8     2     2  0.0278
## 9     3     2  0.0278
## 10    4     2  0.0278
## # ... with 26 more rows
```

U – Soma dos valores dos dois lançamentos

```
omega$U <- omega$dado1 + omega$dado2
tibble(omega)
```

```
## # A tibble: 36 x 4
##   dado1 dado2 prob  U
##   <dbl> <dbl> <dbl> <dbl>
## 1     1     1  1 0.0278  2
## 2     2     2  1 0.0278  3
## 3     3     3  1 0.0278  4
## 4     4     4  1 0.0278  5
## 5     5     5  1 0.0278  6
## 6     6     6  1 0.0278  7
## 7     1     2  2 0.0278  3
## 8     2     2  2 0.0278  4
## 9     3     2  2 0.0278  5
## 10    4     2  2 0.0278  6
## # ... with 26 more rows
```

1

Construir um dataframe que contenha, na primeira coluna, os valores possíveis para esta variável e, na segunda, as respectivas probabilidades de ocorrência, ou seja a função de probabilidade, $f(u)$.

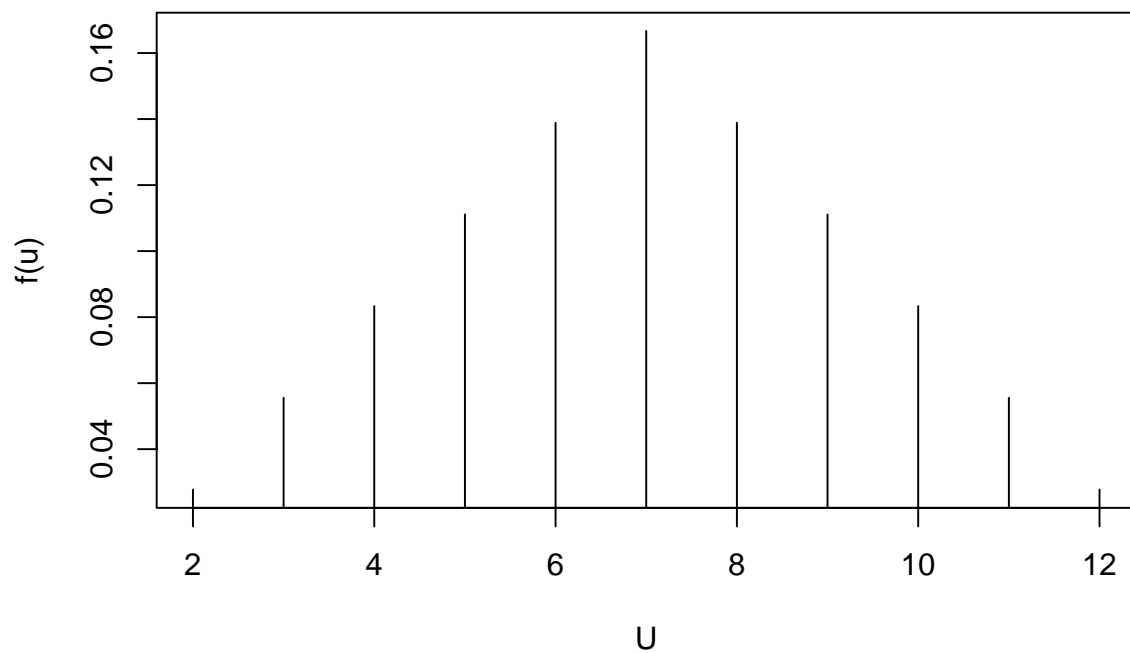
```
#fp <- omega %>% group_by(U) %>% summarise(fp = n()/36)
fp <- aggregate(prob ~ U, data = omega, FUN = sum)
tibble(fp)
```

```
## # A tibble: 11 x 2
##   U prob
##   <dbl> <dbl>
## 1     2 0.0278
## 2     3 0.0556
## 3     4 0.0833
## 4     5 0.111
## 5     6 0.139
## 6     7 0.167
## 7     8 0.139
## 8     9 0.111
## 9    10 0.0833
## 10    11 0.0556
## 11    12 0.0278
```

2

Representar graficamente a função de probabilidade.

```
plot(fp$U, fp$prob, type = "h", xlab = "U", ylab = "f(u)")
```



3

Obter a função de distribuição, $F(u)$, nos pontos de probabilidade não nula de U.

```
Fu <- cumsum(fp$prob)  
tibble(U = fp$U, Fu = Fu)
```

```
## # A tibble: 11 x 2
##       U      Fu
##   <dbl> <dbl>
## 1     2 0.0278
## 2     3 0.0833
## 3     4 0.167
## 4     5 0.278
## 5     6 0.417
## 6     7 0.583
## 7     8 0.722
## 8     9 0.833
## 9    10 0.917
## 10   11 0.972
## 11   12 1
```

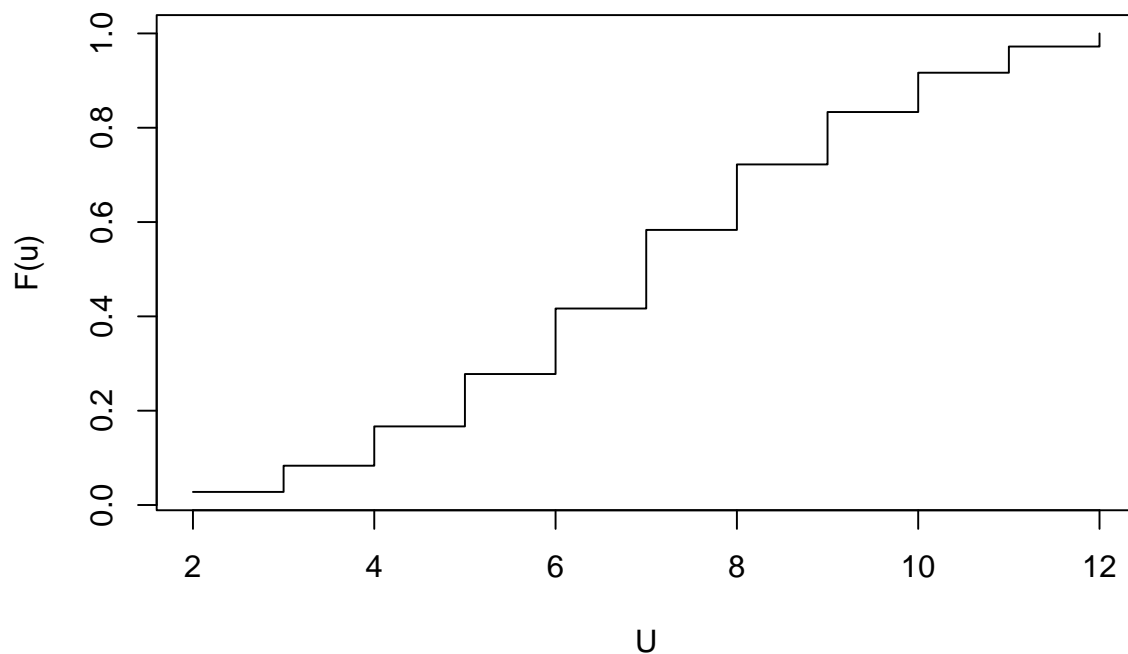
4

TODO

5

Representar graficamente a função de distribuição.

```
plot(fp$U, Fu, type = "s", xlab = "U", ylab = "F(u)")
```



6 > Qual a probabilidade de obter uma soma de pelo menos 7 e não mais do que 10, no lançamento de dois dados equilibrados?

```
# P(7<=U<=10)
# Com recurso a funcao de probabilidade
print(fractions(sum(fp$prob[fp$U >= 7 & fp$U <= 10])))
```

```
## [1] 1/2
```

```
# Com recurso a funcao de distribuicao
print(fractions(Fu[which(fp$U == 10)] - Fu[which(fp$U == 7-1)]))
```

```
## [1] 1/2
```

7

Repetir 6, para um resultado maior que 7 e não mais do que 10

```
#P(7<U<=10)
#Com recurso a funcao de probabilidade
print(fractions(sum(fp$prob[fp$U > 7 & fp$U <= 10])))
```

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

```
#Com recurso a funcao de distribuicao
print(fractions(Fu[which(fp$U == 10)] - Fu[which(fp$U == 7)]))
```

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

8

Repetir 6, para um resultado maior que 7 e menor que 10

```
#P(7<U<10)
#Com recurso a funcao de probabilidade
print(fractions(sum(fp$prob[fp$U > 7 & fp$U < 10])))
```

```
## [1] 1/4
```

```
#Com recurso a funcao de distribuicao
print(fractions(Fu[which(fp$U == 10-1)] - Fu[which(fp$U == 7)]))
```

```
## [1] 1/4
```