

```
#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)
omega <- expand.grid(dado1= dado, dado2= dado)
omega$prob <- 1/nrow(omega)
tibble(omega)
```

```
## # A tibble: 16 x 3
##   dado1 dado2  prob
##   <dbl> <dbl> <dbl>
## 1     1     1  0.0625
## 2     2     1  0.0625
## 3     3     1  0.0625
## 4     4     1  0.0625
## 5     1     2  0.0625
## 6     2     2  0.0625
## 7     3     2  0.0625
## 8     4     2  0.0625
## 9     1     3  0.0625
## 10    2     3  0.0625
## 11    3     3  0.0625
## 12    4     3  0.0625
## 13    1     4  0.0625
## 14    2     4  0.0625
## 15    3     4  0.0625
## 16    4     4  0.0625
```

U – Soma dos valores dos dois lançamentos

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

```
## # A tibble: 16 x 4
##   dado1 dado2 prob  U
##   <dbl> <dbl> <dbl> <dbl>
## 1     1     1   1 0.0625  2
## 2     2     2   1 0.0625  3
## 3     3     3   1 0.0625  4
## 4     4     4   1 0.0625  5
## 5     1     2   2 0.0625  3
## 6     2     2   2 0.0625  4
## 7     3     2   2 0.0625  5
## 8     4     2   2 0.0625  6
## 9     1     3   3 0.0625  4
## 10    2     3   3 0.0625  5
## 11    3     3   3 0.0625  6
## 12    4     3   3 0.0625  7
## 13    1     4   4 0.0625  5
## 14    2     4   4 0.0625  6
## 15    3     4   4 0.0625  7
## 16    4     4   4 0.0625  8
```

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, $f(u)$.

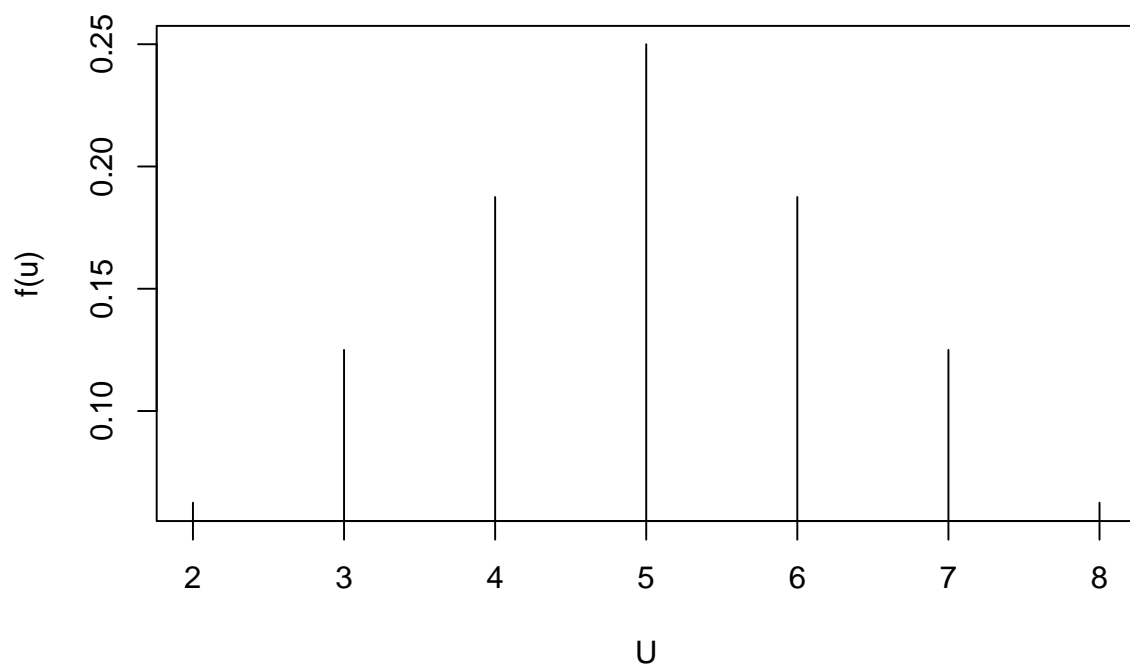
```
#fp <- omega %>% group_by(U) %>% summarise(fp = n()/36)
fp <- aggregate(prob ~ U, data = omega, FUN = sum)
tibble(fp)
```

```
## # A tibble: 7 x 2
##   U   prob
##   <dbl> <dbl>
## 1     2 0.0625
## 2     3 0.125
## 3     4 0.188
## 4     5 0.25
## 5     6 0.188
## 6     7 0.125
## 7     8 0.0625
```

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: 7 x 2
##       U      Fu
##   <dbl> <dbl>
## 1     2 0.0625
## 2     3 0.188
## 3     4 0.375
## 4     5 0.625
## 5     6 0.812
## 6     7 0.938
## 7     8 1
```

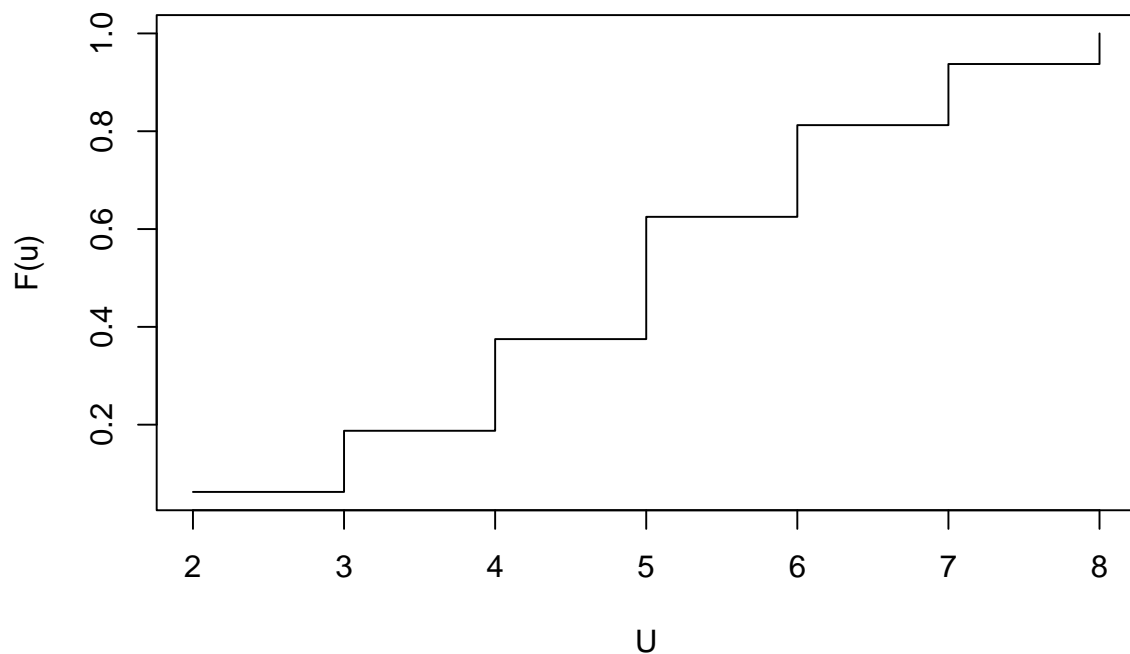
4

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 4 e não mais do que 7, no lançamento de dois dados equilibrados?

```
# P(4<=U<=7)
# Com recurso a funcao de probabilidade
print(sum(fp$prob[fp$U >= 4 & fp$U <= 7]))
```

```
## [1] 0.75
```

```
# Com recurso a funcao de distribuicao
print(Fu[which(fp$U == 7)] - Fu[which(fp$U == 4-1)])
```

```
## [1] 0.75
```

7

Repetir 6, para um resultado maior que 4 e não mais do que 7

```
#P(4<U<=7)
#Com recurso a funcao de probabilidade
print(sum(fp$prob[fp$U > 4 & fp$U <= 7]))
```

```
## [1] 0.5625
```

```
#Com recurso a funcao de distribuicao
print(Fu[which(fp$U == 7)] - Fu[which(fp$U == 4)])
```

```
## [1] 0.5625
```

8

Repetir 6, para um resultado maior que 4 e menor que 7

```
#P(4<U<7)
#Com recurso a funcao de probabilidade
print(sum(fp$prob[fp$U > 4 & fp$U < 7]))
```

```
## [1] 0.4375
```

```
#Com recurso a funcao de distribuicao
print(Fu[which(fp$U == 7-1)] - Fu[which(fp$U == 4)])
```

```
## [1] 0.4375
```

9

Calcular a média, a variância e o desvio-padrão de U, pelas fórmulas gerais do valor esperado e da variância.

```
#media
media <- sum(fp$U * fp$prob)
#variância
variância <- sum((fp$U - media)^2 * fp$prob)
#desvio padrao
desvio_padrao <- sqrt(variância)
tibble(media, variância, desvio_padrao)
```

```
## # A tibble: 1 x 3
##   media variancia desvio_padrao
##   <dbl>      <dbl>      <dbl>
## 1      5        2.5        1.58
```