TPC 3: Variáveis Aleatórias - Turma 1

André Filipe Gomes Silvestre Nº104532 CDB1

Exercício

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

U – Soma dos valores dos dois lançamentos

1. Construir um dataframe que contenha, na primeira coluna, os valores possíveis para esta variável e, na segunda, as respetivas probabilidades de ocorrência, ou seja a função de probabilidade, $f(u)$.
2. Representar graficamente a função de probabilidade.
3. Obter a função de distribuição, $F(u)$, nos pontos de probabilidade não nula de U.
4. Escrever a função de distribuição (com todos os seus ramos).
5. Representar graficamente a função de distribuição.
6. Qual a probabilidade de obter uma soma de pelo menos 7 e não mais do que 10, no lançamento de dois dados equilibrados? Escrever a probabilidade pretendida em termos da variável aleatória definida, e calculá-la:

- Com recurso à função de probabilidade
- Com recurso à função de distribuição.

7. Repetir 6, para um resultado maior que 7 e não mais do que 10.
8. Repetir 6, para um resultado maior que 7 e menor que 10.
9. Calcular a média, a variância e o desvio-padrão de U, pelas fórmulas gerais do valor esperado e da variância.

---

1. **Resolução**

(Vamos em primeiro lugar construir o espaço de resultados para o lançamento dos dois dados)

Os resultados possíveis para o lançamento do um dado de 6 faces são 6: 1,2,3,4,5,ou 6.

Logo, o espaço de resultados (Ω) correspondente ao lançamento do dado 2 vezes contém 36 resultados elementares.

```
dado<-1:6
dado
```

```
## [1] 1 2 3 4 5 6
```

```
esp_res_dados <- expand.grid("dado1"=dado, "dado2"=dado)
```

Uma vez que U corresponde à *soma dos valores dos dois lançamentos* e há vários somas que resultam num mesmo valor, agrega-se para obter a **função de probabilidade** de **U** ($f(u)$), obtendo um dataframe final simplificado.

```
esp_res_dados$soma <- esp_res_dados$dado1 + esp_res_dados$dado2
# OU esp_res_dados$soma <-rowSums(esp_res_dados[,1:2])

esp_res_dados$prob <- rep(1/nrow(esp_res_dados),times=nrow(esp_res_dados))

U <- aggregate(prob~soma,    # agregar a coluna prob, de acordo com a soma das faces
            esp_res_dados,  # no dataframe criado
            sum)            # através da soma
U
```

```
##    soma       prob
## 1     2 0.02777778
## 2     3 0.05555556
## 3     4 0.08333333
## 4     5 0.11111111
## 5     6 0.13888889
## 6     7 0.16666667
## 7     8 0.13888889
## 8     9 0.11111111
## 9    10 0.08333333
## 10   11 0.05555556
## 11   12 0.02777778
```

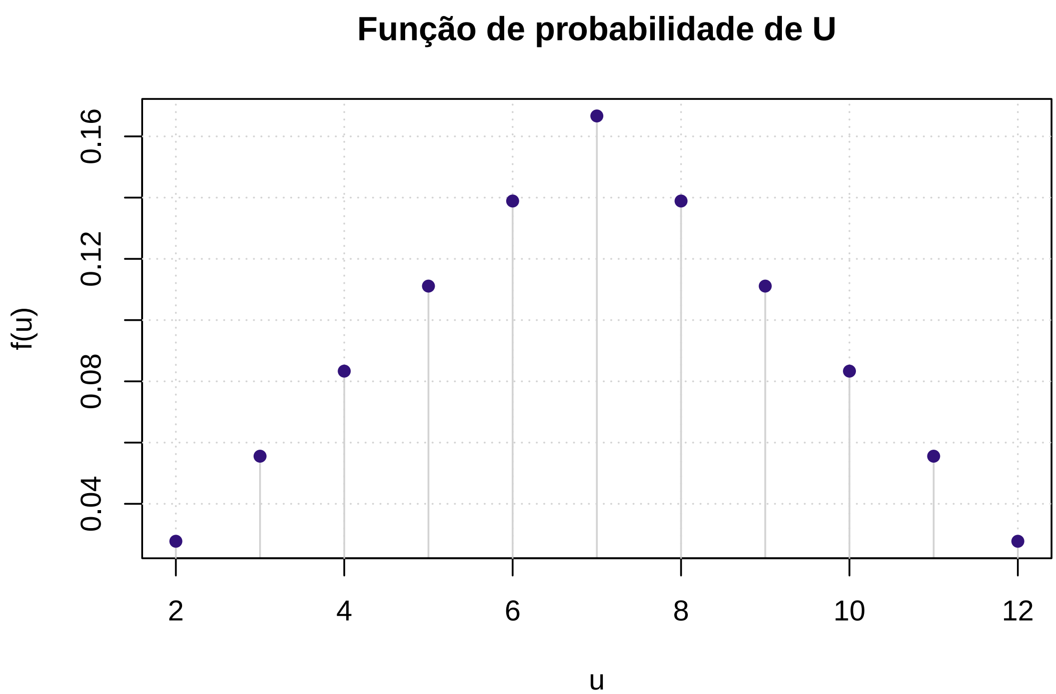
2. Resolução

Graficamente a função de probabilidade $f(u)$ representa-se:

```
plot (x=U$soma,                        # os valores da variável U
      y=U$prob ,                       # os valores da função de probabilidade
      type="h",                        # linhas verticais entre x e y
      main="Função de probabilidade de U",
      xlab="u",                        # id do eixo x
      ylab="f(u)",                     # id do eixo y
      col="light grey")                 # cor das linhas

grid()

#Adicionar pontos ao gráfico
points(x=U$soma,
       y=U$prob,
       pch=16,                         # símbolo usado, 16 é círculo
       col="#32127a")
```



3. Resolução

As imagens da função de distribuição, $F(u)$, nos pontos de probabilidade não nula de U, são dadas por:

```
F_U_pontos<-cumsum(U$prob)
F_U_pontos
```

```
##  [1] 0.02777778 0.08333333 0.16666667 0.27777778 0.41666667 0.58333333
##  [7] 0.72222222 0.83333333 0.91666667 0.97222222 1.00000000
```

```
U$F_U_pontos <- F_U_pontos
U
```

```
##    soma       prob F_U_pontos
## 1     2 0.02777778 0.02777778
## 2     3 0.05555556 0.08333333
## 3     4 0.08333333 0.16666667
## 4     5 0.11111111 0.27777778
## 5     6 0.13888889 0.41666667
## 6     7 0.16666667 0.58333333
## 7     8 0.13888889 0.72222222
## 8     9 0.11111111 0.83333333
## 9    10 0.08333333 0.91666667
## 10   11 0.05555556 0.97222222
## 11   12 0.02777778 1.00000000
```

---

4. Resolução

- Em $\mathbb{R}$, a Função de distribuição $F(u)$ será então:

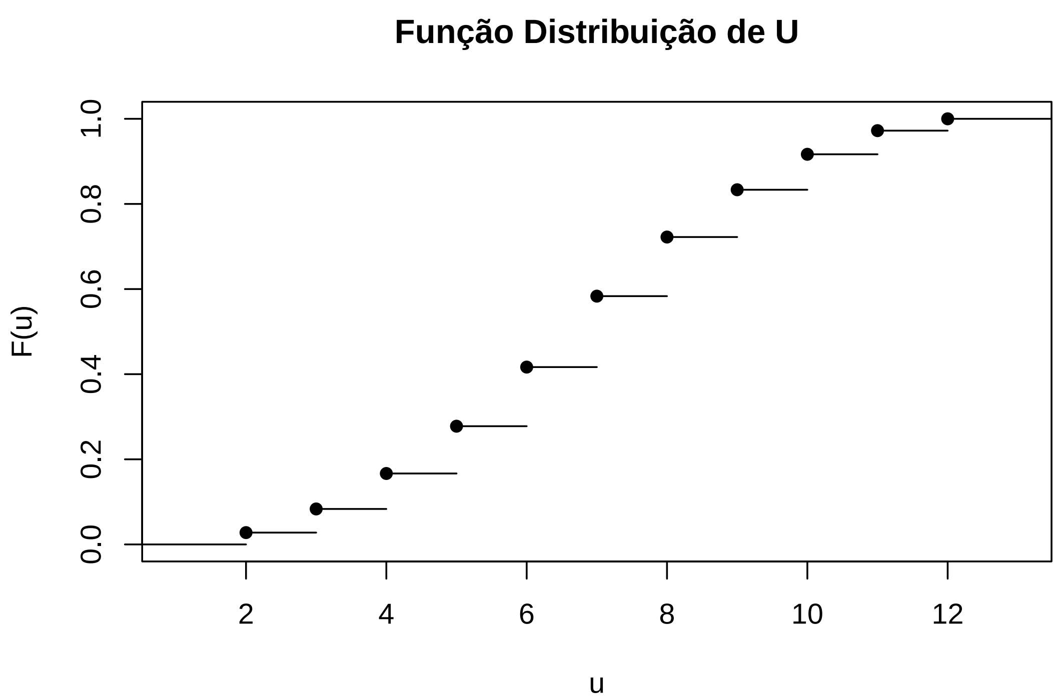
$$F(u) = \begin{cases} 0 & u < 2 \\ 0.028 & 2 \leq u < 3 \\ 0.083 & 3 \leq u < 4 \\ 0.167 & 4 \leq u < 5 \\ 0.278 & 5 \leq u < 6 \\ 0.417 & 6 \leq u < 7 \\ 0.583 & 7 \leq u < 8 \\ 0.722 & 8 \leq u < 9 \\ 0.833 & 9 \leq u < 10 \\ 0.917 & 10 \leq u < 11 \\ 0.972 & 11 \leq u < 12 \\ 1 & u \geq 12 \end{cases}$$

5. Resolução

Representação gráfica da função de distribuição $F(u)$

```
#Representação gráfica da Função de Distribuição de U

plot.stepfun(
  stepfun(                                  # definir a função em patamares
          U$soma,                           # os valores de u a considerar
          c(0,F_U_pontos),                  # os patamares a considerar,
                                            # ponto adicional inicial 0
          right=FALSE                       # intervalos fechados à esquerda
          ),
  verticals=FALSE,                          # não colocar traços verticais nos pontos de salto
  pch = 16,                                 # tipo de símbolo
  main="Função Distribuição de U",          # título e identificação dos eixos
  xlab="u",
  ylab="F(u)")
```



6. Resolução

Seja $P[7 \leq U \leq 10]$ a *probabilidade de obter uma soma de pelo menos 7 e não mais do que 10, no lançamento de dois dados equilibrados*, o valor pode ser obtido de duas formas:

1. Com recurso à Função de Probabilidade

```
sum(U$prob[U$soma %in% 7:10])
```

```
## [1] 0.5
```

```
# --------- ALTERNATIVA ---------
# quais<-which(U$U>=7 & U$U<=10)
# sum(U$prob[quais])
```

2. Com recurso à Função de Distribuição

- Temos de notar que, como U é discreta, $P[7 \leq U \leq 10] = P[6 < U \leq 10]$ e portanto pode ser calculada como $F(10) - F(6)$.

```
F_U_pontos[which(U$soma == 10)] - F_U_pontos[which(U$soma == 6)]
```

```
## [1] 0.5
```

---

7. Resolução

Seja $P[7 < U \leq 10]$ a *probabilidade de obter uma soma maior que 7 e não mais do que 10, no lançamento de dois dados equilibrados*, o valor pode ser obtido de duas formas:

1. Recorrendo a Função de Probabilidade

```
sum(U$prob[U$soma %in% 8:10])
```

```
## [1] 0.3333333
```

```
# --------- ALTERNATIVA ---------
# quais<-which(U$U>7 & U$U<=10)
# sum(U$prob[quais])
```

2. Recorrendo à Função Distribuição

- Basta fazer $F(10) - F(7)$.

```
F_U_pontos[which(U$soma == 10)] - F_U_pontos[which(U$soma == 7)]
```

```
## [1] 0.3333333
```

---

8. Resolução

Seja $P[7 < U < 10]$ a *probabilidade de obter uma soma maior que 7 e menor que 10, no lançamento de dois dados equilibrados*, o valor pode ser obtido de duas formas:

1. Com recurso à Função de Probabilidade

```
sum(U$prob[U$soma %in% 8:9])
```

```
## [1] 0.25
```

```
# --------- ALTERNATIVA ---------
# quais<-which(U$U>7 & U$U<10)
# sum(U$prob[quais])
```

2. Com recurso à Função de Distribuição

- Temos de notar que, como U é discreta,$P[7 < U < 10] = P[7 < U \leq 9]$ e portanto pode ser calculada como $F(9) - F(7)$.

```
F_U_pontos[which(U$soma == 9)] - F_U_pontos[which(U$soma == 7)]
```

```
## [1] 0.25
```

---

9. Resolução

Seguindo a fórmula genérica

$$E[X] = \sum_{1}^{n} x_i f(x_i)$$

, basta calcular a soma dos produtos da primeira com a segunda colunas definidas em U:

```
# Miu de U
miu_u<-sum(U$soma*U$prob)
miu_u
```

Média de U

```
## [1] 7
```

Variância($VAR[U]$) Para a variância, podemos aplicar a fórmula simplificada (média dos quadrados menos o quadrado da média) ou a definição formal (média dos quadrados dos desvios face à média).

```
# Média dos Quadrados menos o Quadrado da Média
var_u<- sum((U$soma^2)*U$prob)-miu_u^2
round(var_u,4)
```

```
## [1] 5.8333
```

```
# Média dos Quadrados dos Desvios face à Média
round(sum((U$soma-miu_u)^2*U$prob),4)
```

```
## [1] 5.8333
```

Desvio-Padrão (σ_U)

$\therefore$ Assim o desvio-padrão será $\sigma_U = 2.42$