

# TPC\_1

Diogo Alexandre Alonso De Freitas

2022-09-13

```
##
## Attaching package: 'dplyr'

##
## The following objects are masked from 'package:stats':
##
##   filter, lag

##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

## Exercicio 1

```
# Cubo de 6 Lados
dado_6_faces <- 1:6

# Cubo de 8 Lados
dado_8_faces <- 1:8

# Criar colunas com todas as combinações possíveis
# Coluna 1 é o cubo de 6 faces, coluna 2 é o cubo de 8 faces
dados_combi <- expand.grid(dado_6_faces, dado_8_faces)
dados_combi

##   Var1 Var2
## 1     1     1
## 2     2     1
## 3     3     1
## 4     4     1
## 5     5     1
## 6     6     1
## 7     1     2
## 8     2     2
## 9     3     2
## 10    4     2
## 11    5     2
## 12    6     2
## 13     1     3
## 14     2     3
## 15     3     3
## 16     4     3
## 17     5     3
## 18     6     3
## 19     1     4
## 20     2     4
## 21     3     4
## 22     4     4
## 23     5     4
## 24     6     4
## 25     1     5
## 26     2     5
## 27     3     5
## 28     4     5
## 29     5     5
## 30     6     5
## 31     1     6
## 32     2     6
## 33     3     6
## 34     4     6
## 35     5     6
## 36     6     6
## 37     1     7
## 38     2     7
## 39     3     7
## 40     4     7
## 41     5     7
## 42     6     7
## 43     1     8
## 44     2     8
## 45     3     8
## 46     4     8
## 47     5     8
## 48     6     8

# Adicionar uma 3ª coluna com as respectivas probabilidades de cada combinação
probs_dados <- rep(1/nrow(dados_combi), times=nrow(dados_combi))
dados_combi_prob <- data.frame(dados_combi, probs_dados)
dados_combi_prob

##   Var1 Var2 probs_dados
## 1     1     1 0.02083333
## 2     2     1 0.02083333
## 3     3     1 0.02083333
## 4     4     1 0.02083333
## 5     5     1 0.02083333
## 6     6     1 0.02083333
## 7     1     2 0.02083333
## 8     2     2 0.02083333
## 9     3     2 0.02083333
## 10    4     2 0.02083333
## 11    5     2 0.02083333
## 12    6     2 0.02083333
## 13     1     3 0.02083333
## 14     2     3 0.02083333
## 15     3     3 0.02083333
## 16     4     3 0.02083333
## 17     5     3 0.02083333
## 18     6     3 0.02083333
## 19     1     4 0.02083333
## 20     2     4 0.02083333
## 21     3     4 0.02083333
## 22     4     4 0.02083333
## 23     5     4 0.02083333
## 24     6     4 0.02083333
## 25     1     5 0.02083333
## 26     2     5 0.02083333
## 27     3     5 0.02083333
## 28     4     5 0.02083333
## 29     5     5 0.02083333
## 30     6     5 0.02083333
## 31     1     6 0.02083333
## 32     2     6 0.02083333
## 33     3     6 0.02083333
## 34     4     6 0.02083333
## 35     5     6 0.02083333
## 36     6     6 0.02083333
## 37     1     7 0.02083333
## 38     2     7 0.02083333
## 39     3     7 0.02083333
## 40     4     7 0.02083333
## 41     5     7 0.02083333
## 42     6     7 0.02083333
## 43     1     8 0.02083333
## 44     2     8 0.02083333
## 45     3     8 0.02083333
## 46     4     8 0.02083333
## 47     5     8 0.02083333
## 48     6     8 0.02083333
```

## Exercicio 2

```
# Criar a condição em que a soma das faces dos dois cubos seja superior a 8
quais_maior_8 <- which(dados_combi_prob$Var1 + dados_combi_prob$Var2 > 8)
# Ver quais combinação respeitam a condição anterior e as suas probabilidades
soma_maior_8 <- dados_combi_prob[quais_maior_8,]
soma_maior_8

##   Var1 Var2 probs_dados
## 18     6     3 0.02083333
## 23     5     4 0.02083333
## 24     6     4 0.02083333
## 28     4     5 0.02083333
## 29     5     5 0.02083333
## 30     6     5 0.02083333
## 33     3     6 0.02083333
## 34     4     6 0.02083333
## 35     5     6 0.02083333
## 36     6     6 0.02083333
## 38     2     7 0.02083333
## 39     3     7 0.02083333
## 40     4     7 0.02083333
## 41     5     7 0.02083333
## 42     6     7 0.02083333
## 43     1     8 0.02083333
## 44     2     8 0.02083333
## 45     3     8 0.02083333
## 46     4     8 0.02083333
## 47     5     8 0.02083333
## 48     6     8 0.02083333

# Fazer soma de todas as probabilidades
p_de_soma_mai_8 <- sum(soma_maior_8$probs_dados)
p_de_soma_mai_8

## [1] 0.4375
```

## Exercicio 3

```
# Criar a condição em que a face do cubo 1 é igual á face do cubo 2
quais_iguais <- which(dados_combi_prob$Var1 == dados_combi_prob$Var2)
# Ver quais combinação respeitam a condição anterior e as suas probabilidades
face_igual <- dados_combi_prob[quais_iguais,]
face_igual

##   Var1 Var2 probs_dados
## 1     1     1 0.02083333
## 8     2     2 0.02083333
## 15    3     3 0.02083333
## 22    4     4 0.02083333
## 29    5     5 0.02083333
## 36    6     6 0.02083333

# Fazer soma de todas as probabilidades
p_de_face_igual <- sum(face_igual$probs_dados)
p_de_face_igual

## [1] 0.125
```

## Exercicio 4

```
# Fazer a interseção das 2 condições anteriores
quais_maior_8_igual <- intersect(soma_maior_8, face_igual)
quais_maior_8_igual

##   Var1 Var2 probs_dados
## 1     5     5 0.02083333
## 2     6     6 0.02083333

# Fazer soma de todas as probabilidades
p_quais_maior_8_igual <- sum(quais_maior_8_igual$probs_dados)
p_quais_maior_8_igual

## [1] 0.04166667
```

## Exercicio 5

```
# Fazer a interseção das 2 condições anteriores
quais_maior_8_ou_igual <- union(soma_maior_8, face_igual)
quais_maior_8_ou_igual

##   Var1 Var2 probs_dados
## 1     6     3 0.02083333
## 2     5     4 0.02083333
## 3     6     4 0.02083333
## 4     4     5 0.02083333
## 5     5     5 0.02083333
## 6     6     5 0.02083333
## 7     3     6 0.02083333
## 8     4     6 0.02083333
## 9     5     6 0.02083333
## 10    6     6 0.02083333
## 11    2     7 0.02083333
## 12    3     7 0.02083333
## 13    4     7 0.02083333
## 14    5     7 0.02083333
## 15    6     7 0.02083333
## 16    1     8 0.02083333
## 17    2     8 0.02083333
## 18    3     8 0.02083333
## 19    4     8 0.02083333
## 20    5     8 0.02083333
## 21    6     8 0.02083333
## 22    1     1 0.02083333
## 23    2     2 0.02083333
## 24    3     3 0.02083333
## 25    4     4 0.02083333

# Fazer soma de todas as probabilidades
p_quais_maior_8_ou_igual <- sum(quais_maior_8_ou_igual$probs_dados)
p_quais_maior_8_ou_igual

## [1] 0.5208333
```