



PUC-RIO

INF1036 - Probabilidade Computacional

Material 5 - Probabilidade Condicional
Professora - Ana Carolina Letichevsky*
2022.1

*Material Adaptado de Professor Hélio Lopes

A **probabilidade condicional** de um evento A dado o evento B , denotada por $P(A|B)$, é definida por:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \text{ com } P(B) > 0,$$

onde $P(A \cap B)$ é a probabilidade conjunta de A e B .

Similarmente,

$$P(B|A) = \frac{P(A \cap B)}{P(A)}, \text{ com } P(A) > 0,$$

é a probabilidade condicional de B dado A .

Utilizando as equações de $P(A|B)$ e de $P(B|A)$, podemos escrever:

$$P(A \cap B) = P(A|B)P(B)$$

$$P(A \cap B) = P(B|A)P(A)$$

Regra de Bayes: Se A e B são dois eventos em S , então:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Probabilidade Condicional



Exemplo 1) Considere um experimento que consiste em lançar um dado não viciado.

Sejam:

$$S = \{1, 2, 3, 4, 5, 6\}$$

$$A = \{2, 4, 6\}$$

$$B = \{1, 2, 5\}$$

Temos:

$$P(B) = \frac{n(B)}{n(S)} = \frac{3}{6} = \frac{1}{2}$$

$$P(A) = \frac{n(A)}{n(S)} = \frac{3}{6} = \frac{1}{2}$$

```
PA = probabilidade(A, S)
print(PA)
```

```
PB = probabilidade(B, S)
print(PB)
```

$P(B)$ é a probabilidade de *Ba priori*, antes que o evento se realize.

$P(A)$ é a probabilidade de *Aa priori*, antes que o evento se realize.

Exemplo 1, continuação) Considere um experimento que consiste em lançar um dado não viciado.

Temos:

Sejam:

$$S = \{1, 2, 3, 4, 5, 6\}$$

$$A = \{2, 4, 6\}$$

$$B = \{1, 2, 5\}$$

$$P(B) = \frac{n(B)}{n(S)} = \frac{3}{6} = \frac{1}{2}$$

$$P(A) = \frac{n(A)}{n(S)} = \frac{3}{6} = \frac{1}{2}$$

```
nsamples <- 1000
S <- sample(c(1,2,3,4,5,6), nsamples, replace = T)
A <- S [(S == 2) | (S == 4) | (S == 6)]
PA <- (length(A)/nsamples)
print(PA)
```

```
nsamples <- 1000
S <- sample(c(1,2,3,4,5,6), nsamples, replace = T)
B <- S [(S == 1) | (S == 2) | (S == 5)]
PB <- (length(B)/nsamples)
print(PB)
```

nsamples	P(A)	P(B)
100	0,55000	0,48000
1000	0,49600	0,50200
10000	0,50550	0,49960
100000	0,50128	0,49988

Exemplo 1, continuação) Considere um experimento que consiste em lançar um dado não viciado.

Temos:

Sejam:

$$S = \{1, 2, 3, 4, 5, 6\}$$

$$A = \{2, 4, 6\}$$

$$B = \{1, 2, 5\}$$

$$P(B) = \frac{n(B)}{n(S)} = \frac{3}{6} = \frac{1}{2}$$

$$P(A) = \frac{n(A)}{n(S)} = \frac{3}{6} = \frac{1}{2}$$

```
nsamples = 1000
S = []
for i in range(nsamples):
    S.append(random.choice([1, 2, 3, 4, 5, 6]))
A = list(filter(lambda x: (x == 2) or (x == 4) or (x == 6), S))
B = list(filter(lambda x: (x == 1) or (x == 2) or (x == 5), S))
PA = (len(A)/nsamples)
PB = (len(B)/nsamples)
print(PA)
print(PB)
```

Exemplo 1, continuação) Considere um experimento que consiste em lançar um dado não viciado.

Sejam:

$$S = \{1, 2, 3, 4, 5, 6\}$$

$$A = \{2, 4, 6\}$$

$$B = \{1, 2, 5\}$$

```
nsamples <- 100000; cont.2 <- 0; cont.a <- 0
for (i in 1:nsamples) {
  face <- sample(c(1,2,3,4,5,6), 1, replace = T)
  if ((face == 2) | (face == 4) | (face == 6)) {
    cont.a <- cont.a + 1
  }
  if (face == 2) {
    cont.2 <- cont.2 + 1
  }
}
PBdadoA <- (cont.2 / cont.a)
print(PBdadoA)
```

Suponhamos que ao realizar o experimento alguém nos informe que o resultado do mesmo é um número par, ou seja, que A ocorreu.

Com essa informação sabemos que B só pode ocorrer se o resultado tiver sido 2.

$$P(B|A) = \frac{n(B \cap A)}{n(A)} = \frac{1}{3}$$

$P(B|A)$ é a probabilidade *a posteriori* de B dado que A ocorreu.

Exemplo 1, continuação) Considere um experimento que consiste em lançar um dado não viciado.

Sejam:

$$S = \{1, 2, 3, 4, 5, 6\}$$

$$A = \{2, 4, 6\}$$

$$B = \{1, 2, 5\}$$

```
import random
nsamples = 100000
cont2 = 0
conta = 0
for i in range(nsamples):
    face = random.choice([1,2,3,4,5,6])
    if ((face == 2) or (face == 4) or (face == 6)):
        conta = conta + 1
    if (face == 2):
        cont2 = cont2 + 1
PBdadoA = (cont2 / conta)
print(PBdadoA)
```

Suponhamos que ao realizar o experimento alguém nos informe que o resultado do mesmo é um número par, ou seja, que A ocorreu.

Com essa informação sabemos que B só pode ocorrer se o resultado tiver sido 2.

$$P(B|A) = \frac{n(B \cap A)}{n(A)} = \frac{1}{3}$$

$P(B|A)$ é a probabilidade *a posteriori* de B dado que A ocorreu.

Propriedades:

- $P(\emptyset \mid A) = 0$
- $P(S \mid A) = 1$
- $0 \leq P(A \mid B) \leq 1$
- $P((A \cup C) \mid B) = P(A \mid B) + P(C \mid B)$, se $A \cap C = \emptyset$

Teorema de Bayes



Se E_1, E_2, \dots, E_n são eventos mutuamente exclusivos e exaustivos em S e B um evento qualquer em S com $P(B) > 0$, então:

$$P(E_i|B) = \frac{P(B|E_i)P(E_i)}{\sum_{i=1}^n P(B|E_i)P(E_i)} ,$$

para $i \in \{1, \dots, n\}$.

As probabilidades $P(E_i)$ são chamadas de probabilidades ***a priori***, e $P(E_i|B)$ de probabilidade ***a posteriori***.

Os eventos E_1, E_2, \dots, E_n são chamados de **eventos mutuamente exclusivos e exaustivos** se

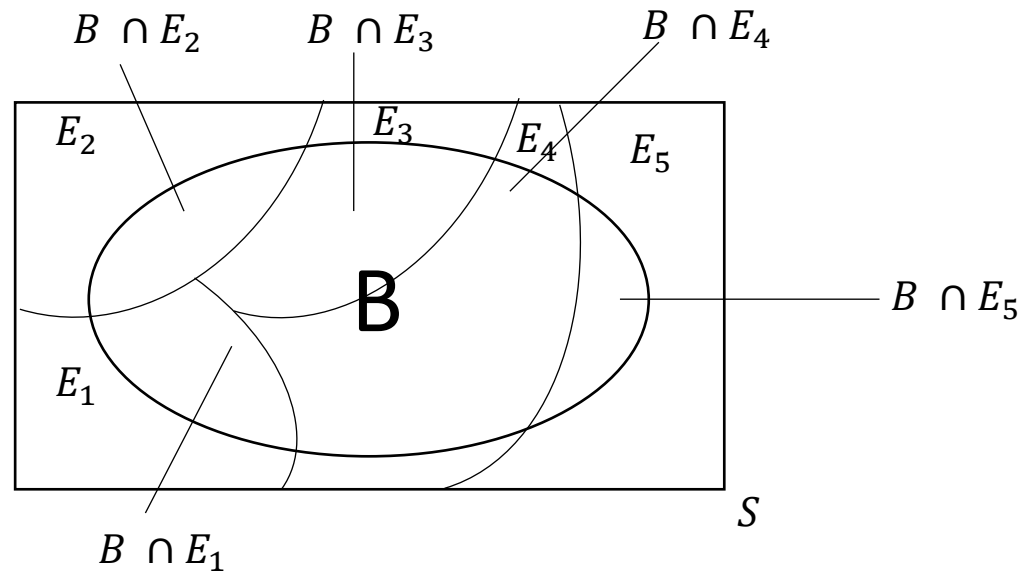
$$\bigcup_{i=1}^n E_i = S \text{ e } E_i \cap E_j = \emptyset \text{ quando } i \neq j$$

Seja B qualquer evento em S . Então

$$P(B) = \sum_{i=1}^n P(B \cap E_i) = \sum_{i=1}^n P(B \mid E_i)P(E_i),$$

que é chamada de **probabilidade total** de B .

Se $i = 5$, em termos de diagrama de Venn temos:



Nesse caso:

$$P(B) = P(B|E_1)(E_1) + P(B|E_2)(E_2) + P(B|E_3)(E_3) + P(B|E_4)(E_4) + P(B|E_5)(E_5)$$

Eventos Independentes



Dois eventos A e B são ditos **estatisticamente independentes** se e somente se

$$P(A \cap B) = P(A)P(B)$$

Imediatamente, isso implica em:

$$P(A|B) = P(A) \text{ e}$$

$$P(B|A) = P(B)$$

Os eventos A_1, A_2, \dots, A_n são ditos **estatisticamente independentes** se e somente se

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = P(A_1)P(A_2) \cdots P(A_n)$$

Finalmente, dizemos que um conjunto infinito de eventos são independentes se e somente se qualquer subconjunto finito desses eventos é independente.

Exemplo 1) Em uma instituição 20% dos funcionários tem problemas visuais, 8% tem problemas auditivos e 4% tem problemas visuais e auditivos.

- a) Os eventos “ter problemas auditivos” e “ter problemas visuais” são independentes?
- b) Ao selecionar um funcionário ao acaso qual a probabilidade dele ter problemas auditivos se ele tem problemas visuais?
- c) Ao selecionar um funcionário ao acaso qual a probabilidade dele não ter problemas visuais ou ter problemas auditivos?

Na resolução vamos considerar:

E_1 : “o funcionário tem problema visual”

E_2 : “o funcionário tem problema auditivo”

Exemplo 1, continuação) Em uma instituição 20% dos funcionários tem problemas visuais, 8% tem problemas auditivos e 4% tem problemas visuais e auditivos.

a) Os eventos “ter problemas auditivos” e “ter problemas visuais” são independentes?

$$P(E_1) = 0,20$$

$$P(E_2) = 0,08$$

$$P(E_1 \cap E_2) = 0,04$$

$$P(E_1) P(E_2) = 0,016$$

Como

$P(E_1 \cap E_2) \neq P(E_1) P(E_2)$ os eventos E_1 e E_2 não são independentes.

Exemplo 1, continuação) Em uma instituição 20% dos funcionários tem problemas visuais, 8% tem problemas auditivos e 4% tem problemas visuais e auditivos.

a) Os eventos “ter problemas auditivos” e “ter problemas visuais” são independentes?

```
nsamples <- 1000000
contvisualauditivo <- 0
for (i in 1:nsamples) {
  visual <- sample(c(T, F), 1, prob = c(0.2, 0.8))
  auditivo <- sample(c(T, F), 1, prob = c(0.08, 0.92))
  if (visual & auditivo) {
    contvisualauditivo <- contvisualauditivo + 1
  }
}
print(contvisualauditivo/nsamples)
```


Exemplo 1, continuação) Em uma instituição 20% dos funcionários tem problemas visuais, 8% tem problemas auditivos e 4% tem problemas visuais e auditivos.

b) Ao selecionar um funcionário ao acaso qual a probabilidade dele ter problemas auditivos se ele tem problemas visuais?

$$P(E_2|E_1) = \frac{P(E_2 \cap E_1)}{P(E_1)} = \frac{0,04}{0,20} = 0,20$$

Exemplo 1, continuação) Em uma instituição 20% dos funcionários tem problemas visuais, 8% tem problemas auditivos e 4% tem problemas visuais e auditivos.

b) Ao selecionar um funcionário ao acaso qual a probabilidade dele ter problemas auditivos se ele tem problemas visuais?

```
nsamples <- 1000000
contvisual <- 0
contvisualauditivo <- 0
for (i in 1:nsamples) {
  visual <- sample(c(T, F), 1, prob = c(0.2, 0.8))
  if (visual) {
    contvisual <- contvisual + 1
    visualauditivo <- sample(c(T, F), 1, prob = c(0.04, 0.96))
    if (visualauditivo) {
      contvisualauditivo <- contvisualauditivo + 1
    }
  }
}
print(contvisualauditivo/contvisual)
```

Exemplo 1, continuação) Em uma instituição 20% dos funcionários tem problemas visuais, 8% tem problemas auditivos e 4% tem problemas visuais e auditivos.

c) Ao selecionar um funcionário ao acaso qual a probabilidade dele não ter problemas visuais ou ter problemas auditivos?

$$\begin{aligned}P(\overline{E_1} \cup E_2) &= P(\overline{E_1}) + P(E_2) - P(\overline{E_1} \cap E_2) \\&= 1 - P(E_1) + P(E_2) - P(E_2)P(\overline{E_1} | E_2) \\&= 1 - P(E_1) + P(E_2) - P(E_2)(1 - P(E_1 | E_2)) \\&= 1 - P(E_1) + P(E_2) - P(E_2) \left[1 - \frac{P(E_1 \cap E_2)}{P(E_2)} \right] \\&= 1 - 0,20 + 0,08 - 0,08 \left[1 - \frac{0,04}{0,08} \right] = 0,84\end{aligned}$$

Exemplo 1, continuação) Em uma instituição 20% dos funcionários tem problemas visuais, 8% tem problemas auditivos e 4% tem problemas visuais e auditivos.

c) Ao selecionar um funcionário ao acaso qual a probabilidade dele não ter problemas visuais ou ter problemas auditivos?

```
nsamples <- 1000000; contnvisual <- 0; contauditivo <- 0; contvisualeauditivo <- 0
for (i in 1:nsamples) {
  visual <- sample(c(T, F), 1, prob = c(0.2, 0.8))
  auditivo <- sample(c(T, F), 1, prob = c(0.08, 0.92))
  visualeauditivo <- sample(c(T, F), 1, prob = c(0.04, 0.96))
  if (visual == F) {
    contnvisual <- contnvisual + 1
  }
  if (auditivo) {
    contauditivo <- contauditivo + 1
  }
  if (visualeauditivo) {
    contvisualeauditivo <- contvisualeauditivo + 1
  }
}
print((contnvisual + contauditivo - contvisualeauditivo)/nsamples)
```

1. Se $\{E_i, i = 1, 2, \dots, n\}$ é uma sequência de eventos mutuamente exclusivos, então:

$$P\left(\bigcup_{i=1}^n E_i\right) = \sum_{i=1}^n P(E_i)$$

2. Se $\{E_i, i = 1, 2, \dots, n\}$ é uma sequência de eventos independentes, então:

$$P\left(\bigcap_{i=1}^n E_i\right) = \prod_{i=1}^n P(E_i)$$

Exemplo 2) Para uma empresa, um grupo de funcionários está classificado da seguinte forma:

Tempo na Empresa	fala inglês	fala alemão	fala francês
até 5 anos	82	30	37
5 anos ou mais	101	23	27

Escolhe-se um funcionário ao acaso. Sabendo-se que este funcionário fala francês, qual é a probabilidade de que ele tenha até 5 anos na empresa?

Sejam A e B os eventos:

A: “o funcionário escolhido fala francês”; e

B: “o funcionário escolhido tem até 5 anos de empresa”.

$$P(A) = \frac{37+27}{300} = \frac{64}{300}, \quad P(A \cap B) = \frac{37}{300}, \quad P(B|A) = \frac{P(A \cap B)}{P(A)} = \frac{37}{64}$$

Como todos os pontos do espaço amostral são igualmente prováveis também poderíamos fazer.

$$P(B|A) = \frac{n(A \cap B)}{n(A)} = \frac{37}{64}$$

Exemplo 2, continuação) Para uma empresa, um grupo de funcionários está classificado da seguinte forma:

```
nsamples <- 10000
ate5falamingles <- 82; ate5falamalemao <- 30; ate5falamfrances <- 37
mais5falamingles <- 101; mais5falamalemao <- 23; mais5falamfrances <- 27
probfalamingles <- (ate5falamingles + mais5falamingles) / totalfuncionario
probfalamalemao <- (ate5falamalemao + mais5falamalemao) / totalfuncionario
probfalamfrances <- (ate5falamfrances + mais5falamfrances) / totalfuncionario
probfalamfranceate5 <- ate5falamfrances / (ate5falamfrances + mais5falamfrances)
probfalamfrancemais5 <- mais5falamfrances / (ate5falamfrances + mais5falamfrances)
contfala <- 0; contate5 <- 0
for (i in 1:nsamples) {
  fala <- sample(c("I", "A", "F"), 1, prob = c(probfalamingles, probfalamalemao, probfalamfrances))
  if (fala == "F") {
    contfala <- contfala + 1
    ate5 <- sample(c(T, F), 1, prob = c(probfalamfranceate5, probfalamfrancemais5))
    if (ate5) {
      contate5 <- contate5 + 1
    }
  }
}
prob <- (contate5/contfala)
print(prob)
```

Exemplo 3) Um aluno realiza uma prova com 10 questões do tipo verdadeiro-falso. Calcule a probabilidade dele:

- a) acertar todas as questões ao escolher aleatoriamente as respostas;
- b) acertar todas as questões sabendo que há mais respostas “verdadeiras” do que “falsas”;
- c) errar pelo menos uma questão;
- d) errar pelo menos uma questão sabendo que há mais respostas “verdadeiras” do que “falsas”.

Exemplo 3, continuação) Um aluno realiza uma prova com 10 questões do tipo verdadeiro-falso. Calcule a probabilidade dele:

a) acertar todas as questões ao escolher aleatoriamente as respostas,

Há $2^{10} = 1024$ formas possíveis de escolher as respostas, logo $P[\text{acertar as 10 perguntas}] = \frac{1}{1024}$.

```
nsamples <- 1000000; cont.acerto <- 0; cont.gabaritou <- 0; nquestao <- 10
gabarito <- c("V","F","V","V","F","V","F","V","F","F") # um gabarito qualquer
for (i in 1:nsamples) {
  cont.acerto <- 0
  for (questao in 1:nquestao) { # escolhe as 10 respostas
    resposta <- sample(c("V","F"), 1, replace = T)
    if (resposta == gabarito[questao]) { # verifica se acertou a resposta
      cont.acerto <- cont.acerto + 1
    }
  }
  if (cont.acerto == nquestao) { # verifica a quantidade de acertos
    cont.gabaritou <- cont.gabaritou + 1
  }
}
prob <- (cont.gabaritou/nsamples)
print(prob)
```

Exemplo 3, continuação) Um aluno realiza uma prova com 10 questões do tipo verdadeiro-falso. Calcule a probabilidade dele:

a) acertar todas as questões ao escolher aleatoriamente as respostas,

Há $2^{10} = 1024$ formas possíveis de escolher as respostas, logo $P[\text{acertar as 10 perguntas}] = \frac{1}{1024}$.

```
nsamples <- 1000000
cont.gabaritou <- 0
nquestao <- 10
gabarito <- c("V", "F", "V", "V", "F", "V", "F", "V", "F", "F") # um gabarito qualquer
for (i in 1:nsamples) {
  respostas <- sample(c("V", "F"), nquestao, replace = T) # monta vetor de respostas
  acertos <- (respostas == gabarito) # retorna um vetor com T onde respostas[i] é igual ao gabarito[i]
  cont.acerto <- length(acertos[acertos == T]) # filtra elementos com T e pega o tamanho do vetor
  if (cont.acerto == nquestao) { # verifica a quantidade de acertos
    cont.gabaritou <- cont.gabaritou + 1
  }
}
prob <- (cont.gabaritou/nsamples)
print(prob)
```

Exemplo 3, continuação) Um aluno realiza uma prova com 10 questões do tipo verdadeiro-falso. Calcule a probabilidade dele:

a) acertar todas as questões ao escolher aleatoriamente as respostas,

Há $2^{10} = 1024$ formas possíveis de escolher as respostas, logo $P[\text{acertar as 10 perguntas}] = \frac{1}{1024}$.

```
import random
nsamples = 1000000
contacerto = 0
contgabaritou = 0
nquestao = 10
gabarito = ["V","F","V","V","F","V","F","V","F","F"] # um gabarito qualquer
resposta = ""
for i in range(nsamples):
    contacerto = 0
    for questao in range(nquestao): # escolhe as 10 respostas
        resposta = random.choice(["V","F"])
        if (resposta == gabarito[questao]): # verifica se acertou a resposta
            contacerto = contacerto + 1
    if (contacerto == nquestao): # verifica a quantidade de acertos
        contgabaritou = contgabaritou + 1
prob = (contgabaritou/nsamples)
print(prob)
```

Exemplo 3, continuação) Um aluno realiza uma prova com 10 questões do tipo verdadeiro-falso. Calcule a probabilidade dele:

b) acertar todas as questões sabendo que há mais respostas “verdadeiras” do que “falsas”.

Seja C o conjunto de todas as opções possíveis de respostas com mais “V” do que “F”.

$$n(A) = \binom{10}{6} + \binom{10}{7} + \binom{10}{8} + \binom{10}{9} + \binom{10}{10}$$

$$= 210 + 120 + 45 + 10 + 1 = 386$$

$$P[\text{acertar todas as 10 questões} \mid \text{há mais "V" do que "F"}] = \frac{1}{386}$$

Probabilidade Condicional



```
nsamples <- 1000000
cont.gabarito.v <- 0
cont.gabaritou <- 0
nquestao <- 10
for (i in 1:nsamples) {
  ngabarito.v <- 0
  while (ngabarito.v <= 5) { # gabarito com mais "V" do que "F"
    gabarito <- sample(c("V", "F"), nquestao, replace = T)
    ngabarito.v <- length(gabarito[(gabarito == "V")])
  }
  nresposta.v <- 0
  while (nresposta.v <= 5) { # devido à regra do gabarito, as respostas possuem também mais "V" do que "F"
    resposta <- sample(c("V", "F"), nquestao, replace = T)
    nresposta.v <- length(resposta[(resposta == "V")])
  }
  acertos <- (resposta == gabarito) # retorna um vetor com T onde a resposta[i] é igual ao gabarito[i]
  nacerto <- length(acertos[(acertos == T)]) # filtra T e pega o tamanho do vetor
  if (nacerto == nquestao) { # verifica a quantidade de acertos
    cont.gabaritou <- cont.gabaritou + 1
  }
}
print(cont.gabaritou/nsamples)
```

Probabilidade Condicional



```
import random

nsamples = 1000000
gabarito = []
contgabaritou = 0
nquestao = 10
for i in range(nsamples):
    ngabaritov = 0
    while (ngabaritov <= 5): # gabarito com mais "V" do que "F"
        gabarito = random.choices(["V","F"], k = nquestao)
        ngabaritov = len(list(filter(lambda x: (x == "V"), gabarito)))
    nrespostav = 0
    while (nrespostav <= 5): # devido à regra do gabarito, as respostas possuem também mais "V" do que "F"
        resposta = random.choices(["V","F"], k = nquestao)
        nrespostav = len(list(filter(lambda x: (x == "V"), resposta)))
    if (resposta == gabarito): # compara os dois vetores elemento por elemento e retorna T se são iguais
        contgabaritou = contgabaritou + 1

print(contgabaritou/nsamples)
```

Exemplo 3, continuação) Um aluno realiza uma prova com 10 questões do tipo verdadeiro-falso. Calcule a probabilidade dele:

c) errar pelo menos uma questão.

$$1 - P[\text{acertar todas as questões}] = 1 - \frac{1}{1024} = \frac{1023}{1024}$$

```
nsamples <- 1000000
cont.ngabaritou <- 0
nquestao <- 10
gabarito <- c("V", "F", "V", "V", "F", "V", "F", "V", "F", "F")
for (i in 1:nsamples) {
  respostas <- sample(c("V", "F"), nquestao, replace = T)
  acertos <- (respostas == gabarito) # retorna um vetor com T onde são iguais
  cont.acerto <- length(acertos[(acertos == T)]) # filtra T e pega o tamanho do vetor
  if (cont.acerto != nquestao) { # diferença na lógica para o item a
    cont.ngabaritou <- cont.ngabaritou + 1
  }
}
prob <- (cont.ngabaritou/nsamples)
print(prob)
```

Exemplo 3, continuação) Um aluno realiza uma prova com 10 questões do tipo verdadeiro-falso. Calcule a probabilidade dele:

c) errar pelo menos uma questão.

$$1 - P[\text{acertar todas as questões}] = 1 - \frac{1}{1024} = \frac{1023}{1024}$$

```
import random
nsamples = 1000000
contacerto = 0
contngabaritou = 0
nquestao = 10
gabarito = ["V","F","V","V","F","V","F","V","F","F"] # um gabarito qualquer
resposta = ""
for i in range(nsamples):
    contacerto = 0
    for questao in range(nquestao): # monta as 10 respostas de uma prova
        resposta = random.choice(["V","F"])
        if (resposta == gabarito[questao]):
            contacerto = contacerto + 1
    if (contacerto != nquestao): # diferença na lógica para o item a
        contngabaritou = contngabaritou + 1
prob = (contngabaritou/nsamples)
print(prob)
```


Exemplo 3, continuação) Um aluno realiza uma prova com 10 questões do tipo verdadeiro-falso. Calcule a probabilidade dele:

d) errar pelo menos uma questão sabendo que há mais respostas “verdadeiras” do que “falsas”.

$$1 - P[\text{acertar todas as 10 questões} \mid \text{há mais "V" do que "F"}] = 1 - \frac{1}{386} = \frac{385}{386}$$

Probabilidade Condicional



```
nsamples <- 1000000
cont.gabarito.v <- 0
cont.gabaritou <- 0
nquestao <- 10
for (i in 1:nsamples) {
  ngabarito.v <- 0
  while (ngabarito.v <= 5) { # gabarito com mais "V" do que "F"
    gabarito <- sample(c("V", "F"), nquestao, replace = T)
    ngabarito.v <- length(gabarito[(gabarito == "V")])
  }
  nresposta.v <- 0
  while (nresposta.v <= 5) { # devido à regra do gabarito, as respostas possuem também mais "V" do que "F"
    resposta <- sample(c("V", "F"), nquestao, replace = T)
    nresposta.v <- length(resposta[(resposta == "V")])
  }
  acertos <- (resposta == gabarito) # retorna um vetor com T onde são iguais
  nacerto <- length(acertos[(acertos == T)]) # filtra T e pega o tamanho do vetor
  if (nacerto != nquestao) { # diferença na lógica para o item b
    cont.gabaritou <- cont.gabaritou + 1
  }
}
print(cont.gabaritou/nsamples)
```

Probabilidade Condicional



```
import random

nsamples = 1000000
gabarito = []
contgabaritou = 0
nquestao = 10
for i in range(nsamples):
    ngabaritov = 0
    while (ngabaritov <= 5): # gabarito com mais "V" do que "F"
        gabarito = random.choices(["V","F"], k = nquestao)
        ngabaritov = len(list(filter(lambda x: (x == "V"), gabarito)))
    nrespostav = 0
    while (nrespostav <= 5): # devido à regra do gabarito, as respostas possuem também mais "V" do que "F"
        resposta = random.choices(["V","F"], k = nquestao)
        nrespostav = len(list(filter(lambda x: (x == "V"), resposta)))
    if (resposta != gabarito): # diferença na lógica para o item b
        contgabaritou = contgabaritou + 1

print(contgabaritou/nsamples)
```

Exemplo 4) Uma gincana possui 100 participantes, cada um dos participantes recebe uma identificação diferente, composta por 2 dígitos entre 0 e 9. Dois participantes da gincana, P1 e P2 são escolhidos ao acaso, qual a probabilidade de que:

- a) suas identificações contenham exatamente os mesmos 2 dígitos, com diferença apenas na ordem?
- b) não haja nenhuma coincidência entre os 2 dígitos da identificação de ambos?
- c) o primeiro dígito das duas identificações sejam iguais?
- d) o primeiro dígito das suas identificações sejam diferentes?

Exemplo 4, continuação) a) suas identificações contenham exatamente os mesmos 2 dígitos, com diferença apenas na ordem?

De quantas formas podemos escolher as identificações de P1 e P2?

- Podemos escolher a identificação de P1 de 10×10 formas.
- Dado que escolhemos a identificação de P1 podemos escolher a de P2 de $100 - 1$ formas.
- Podemos escolher as identificações de P1 e de P2 de 100×99 formas.

Quantos pares de identificações atendem ao requisito?

- Não podemos ter dígitos repetidos.
- Dado que uma identificação foi escolhida há apenas uma que atende o requisito.
- Formas de escolher as identificações de P1 e P2 atendendo ao requisito: $9 \times 10 \times 1$: 90.

Seja A o evento: “o par de identificações contem exatamente os mesmos 2 dígitos”

$$P(A) = 90/9900$$

Exemplo 4, continuação) a) suas identificações contenham exatamente os mesmos 2 dígitos, com diferença apenas na ordem?

```
nsamples <- 100000 # número de simulações
letra.a <- numeric(nsamples) # inicializados com 0

for (i in 1:nsamples) {
  P1 <- sample(0:9, 2, replace = T) # identificação de P1
  P2 <- sample(0:9, 2, replace = T) # identificação de P2
  while ((P1[1] == P2[1]) & (P1[2] == P2[2])) { # while que garante P1 diferente de P2
    P2 <- sample(0:9, 2, replace = T)
  }
  if ((P1[1] == P2[2]) & (P1[2] == P2[1])) {
    letra.a[i] <- 1 # como atende o item a) coloco 1
  }
}
Prob.a <- (sum(letra.a)/nsamples)
print(Prob.a)
```

Exemplo 4, continuação) a) suas identificações contenham exatamente os mesmos 2 dígitos, com diferença apenas na ordem?

```
import random
nsamples = 100000 # número de simulações
letraa = []

for i in range(nsamples):
    P1 = random.choices(list(range(10)), k = 2) # identificação de P1
    P2 = random.choices(list(range(10)), k = 2) # identificação de P2
    while ((P1[0] == P2[0]) and (P1[1] == P2[1])): # while que garante P1 diferente de P2
        P2 = random.choices(list(range(10)), k = 2)
    if ((P1[0] == P2[1]) and (P1[1] == P2[0])):
        letraa.append(1) # como atende o item a) coloco 1

Proba = (sum(letraa)/nsamples)
print(Proba)
```

Exemplo 4, continuação) b) não haja coincidência entre os 2 dígitos da identificação de ambos?

Pares de identificações que atendem ao requisito:

- Caso 1: identificação de P1 possui dois números iguais.
Identificações possíveis de P1: $10 \times 1 = 10$; Identificações possíveis de P2: $9 \times 9 = 81$.
- Caso 2: identificação de P1 possui dois números diferentes.
Identificações possíveis de P1: $10 \times 9 = 90$; Identificações possíveis de P2: $8 \times 8 = 64$.
- Pares de identificação atendem ao requisito: $(81 \times 10) + (90 \times 64) = 6570$.

Pares de identificação possíveis de P1 e de P2 foi vista no item a):

- Formas de escolher as identificações de P1 e P2: 9900.

Seja B o evento: “não há coincidência entre os 2 dígitos da identificação de P1 e P2”

$$P(B) = 6570/9900$$

Exemplo 4, continuação) b) não haja coincidência entre os 2 dígitos da identificação de ambos?

```
nsamples <- 100000 # número de simulações
letra.a <- numeric(nsamples) # inicializados com 0

for (i in 1:nsamples) {
  P1 <- sample(0:9, 2, replace = T) # identificação de P1
  P2 <- sample(0:9, 2, replace = T) # identificação de P2
  while ((P1[1] == P2[1]) & (P1[2] == P2[2])) {# while que garante P1 diferente de P2
    P2 <- sample(0:9, 2, replace = T)
  }
  if ((P1[1] != P2[1]) & (P1[1] != P2[2]) & (P1[2] != P2[1]) & (P1[2] != P2[2])) {
    letra.b[i] <- 1 # como atende o item b) coloco 1
  }
}
prob.b <- (sum(letra.b)/nsamples)
print(prob.b)
```

Exemplo 4, continuação) b) não haja coincidência entre os 2 dígitos da identificação de ambos?

```
import random
nsamples = 100000 # número de simulações
letrab = []

for i in range(nsamples):
    P1 = random.choices(list(range(10)), k = 2) # identificação de P1
    P2 = random.choices(list(range(10)), k = 2) # identificação de P2
    while ((P1[0] == P2[0]) and (P1[1] == P2[1])): # while que garante P1 diferente de P2
        P2 = random.choices(list(range(10)), k = 2)
    if ((P1[0] != P2[0]) and (P1[0] != P2[1]) and (P1[1] != P2[0]) and (P1[1] != P2[1])):
        letrab.append(1) # como atende o item b) coloco 1

Probb = (sum(letrab)/nsamples)
print(Probb)
```

Exemplo 4, continuação) c) o primeiro dígito das duas identificações sejam iguais?

Quantos pares de identificações atendem ao requisito?

- Podemos escolher a identificação de P1 de 10×10 formas.
- Dado que escolhemos a identificação de P1 podemos escolher a de P2 de 9 formas.
- Formas de escolher as identificações de P1 e P2 atendendo ao requisito: $100 \times 9 = 900$.

Pares de identificação possíveis de P1 e de P2 foi vista no item a):

- Formas de escolher as identificações de P1 e P2: 9900.

Seja C o evento: “o primeiro dígito das duas identificações é igual”

$$P(C) = 900/9900$$

Exemplo 4, continuação) c) o primeiro dígito das duas identificações sejam iguais?

```
nsamples <- 100000 # número de simulações
letra.c <- numeric(nsamples) # inicializados com 0
for (i in 1:nsamples) {
  P1 <- sample(0:9, 2, replace = T) # identificação de P1
  P2 <- sample(0:9, 2, replace = T) # identificação de P2
  while ((P1[1] == P2[1]) & (P1[2] == P2[2])) { # while que garante P1 diferente de P2
    P2 <- sample(0:9, 2, replace = T)
  }
  if (P1[1] == P2[1]) {
    letra.c[i] <- 1
  }
}
prob.c <- (sum(letra.c)/nsamples)
print(prob.c)
```

Exemplo 4, continuação) c) o primeiro dígito das duas identificações sejam iguais?

```
import random
nsamples = 100000 # número de simulações
letrac = []

for i in range(nsamples):
    P1 = random.choices(list(range(10)), k = 2) # identificação de P1
    P2 = random.choices(list(range(10)), k = 2) # identificação de P2
    while ((P1[0] == P2[0]) and (P1[1] == P2[1])): # while que garante P1 diferente de P2
        P2 = random.choices(list(range(10)), k = 2)
    if (P1[0] == P2[0]):
        letrac.append(1)

Probc = (sum(letrac)/nsamples)
print(Probc)
```

Exemplo 4, continuação) d) o primeiro dígito das suas identificações sejam diferentes?

Pares de identificações que atendem ao requisito:

- Caso 1: identificação de P1 possui dois números iguais.
Identificações possíveis de P1: $10 \times 1 = 10$; Identificações possíveis de P2: $9 \times 10 = 90$.
- Caso 2: identificação de P1 possui dois números diferentes.
Identificações possíveis de P1: $10 \times 9 = 90$; Identificações possíveis de P2: $9 \times 10 = 90$.
- Pares de identificação atendem ao requisito: $(90 \times 10) + (90 \times 90) = 9000$.

Pares de identificações possíveis de P1 e de P2 foi vista no item a):

- Formas de escolher as identificações de P1 e P2: 9900.

Seja C o evento: “o primeiro dígito das duas identificações é diferente”

$$P(D) = 9000/9900 \quad \text{ou} \quad P(D) = 1 - P(C) = 1 - 900/9900$$

Exemplo 4, continuação) d) o primeiro dígito das suas identificações sejam diferentes?

```
nsamples <- 100000 # número de simulações
letra.d <- numeric(nsamples) # inicializados com 0
for (i in 1:nsamples) {
  P1 <- sample(0:9, 2, replace = T) # identificação de P1
  P2 <- sample(0:9, 2, replace = T) # identificação de P2
  while ((P1[1] == P2[1]) & (P1[2] == P2[2])) { # while que garante P1 diferente de P2
    P2 <- sample(0:9, 2, replace = T)
  }
  if (P1[1] != P2[1]) {
    letra.d[i] <- 1
  }
}
prob.d <- (sum(letra.d)/nsamples)
print(prob.d)
```

Exemplo 4, continuação) d) o primeiro dígito das suas identificações sejam diferentes?

```
import random
nsamples = 100000 # número de simulações
letrad = []

for i in range(nsamples):
    P1 = random.choices(list(range(10)), k = 2) # identificação de P1
    P2 = random.choices(list(range(10)), k = 2) # identificação de P2
    while ((P1[0] == P2[0]) and (P1[1] == P2[1])): # while que garante P1 diferente de P2
        P2 = random.choices(list(range(10)), k = 2)
    if (P1[0] != P2[0]):
        letrad.append(1)

Probd = (sum(letrad)/nsamples)
print(Probd)
```