

Métodos Computacionais

Departamento de Estatística e Matemática Aplicada

Ronald Targino, Rafael Braz, Juvêncio Nobre e Manoel Santos-Neto

2025-08-19

Índice

Prefácio	4
1 Introdução	5
2 Motivação	6
2.1 Da teoria à simulação	6
2.2 Um atalho analítico útil	7
2.3 O papel da simulação	8
2.4 Atividade: Problema do Aniversário (22 jogadores)	9
2.5 Exercícios	9
3 Números Uniformes	11
3.1 Geração de sequências $U(0, 1)$	11
3.2 Geradores Congruenciais Lineares	12
3.2.1 Exemplo	12
3.2.2 Implementação em R	12
3.3 Geradores Congruenciais Lineares Mistos	13
3.3.1 Questão de estouro e aritmética modular	14
3.3.2 Implementação em R (com segurança de overflow)	14
3.4 Geradores Congruenciais Lineares Multiplicativos	15
3.4.1 Características e restrições	15
3.4.2 Definição de raiz primitiva	15
3.4.3 Exemplo de implementação em R	16
4 Número Pseudoaleatórios	17
4.1 Introdução	17
4.2 Métodos para Geração de Variáveis Aleatórias Discretas	17
4.2.1 Método da transformação inversa	17
4.2.2 Método da Aceitação-Rejeição	17
4.2.3 Método da Composição	17
4.3 Métodos para Geração de Variáveis Aleatórias Contínuas	17
4.3.1 Método da transformação inversa	17
4.3.2 Método da Aceitação-Rejeição	17
5 Otimização Numérica	18
5.1 Método de Newton	18

5.2	Método de Newton-Raphson	18
5.3	Método Escore de Fisher	18
5.4	Método BFGS	18
6	Métodos de Reamostragem	19
6.1	Bootstrap	19
6.1.1	Introdução	19
6.1.2	Acurária da média amostral	19
6.1.3	Estimativa bootstrap do erro padrão	20
6.1.4	Bootstrap Paramétrico	24
6.2	Jackknife	25
6.2.1	Introdução	25
6.2.2	Estimador do viés	26
6.2.3	Estimado do erro padrão	26
6.3	Intervalos de Confiança	28
6.3.1	Intervalo de Confiança Normal e t-Student	28
6.3.2	Intervalo de Confiança bootstrap-t	28
6.3.3	Intervalos de Confiança bootstrap percentil	28
6.3.4	Intervalos de Confiança bootstrap - versões aprimoradas	28
7	Métodos de Monte Carlo	29
7.1	Introdução	29
7.2	Integração de Monte Carlo	29
7.3	Erro de Monte Carlo	29
7.4	Monte Carlo via Função de Importância	29
7.5	Método de Máxima Verossimilhança	29
8	Algoritmo EM	30
9	Métodos Adicionais	31
	References	32

Prefácio

Este livro resulta de anos de experiência em sala de aula dos professores Ronald Targino, Rafael Braz, Juvêncio Nobre e Manoel Santos-Neto. Destina-se a apoiar os alunos da graduação em Estatística e do Programa de Pós-Graduação em Modelagem e Métodos Quantitativos (PPGMMQ) do Departamento de Estatística e Matemática Aplicada (DEMA) da Universidade Federal do Ceará (UFC).

Ao longo dos capítulos, abordamos a geração de números aleatórios (discretos e contínuos); métodos de suavização; simulação estocástica por inversão, rejeição e composição, bem como métodos de reamostragem; métodos de aproximação e integração; quadratura Gaussiana, integração de Monte Carlo e quadratura adaptativa; métodos de Monte Carlo em sentido amplo; amostradores MCMC, com ênfase em Gibbs e Metropolis–Hastings; otimização numérica via Newton–Raphson, Fisher scoring e quase-Newton, além do algoritmo EM; Bootstrap e Jackknife; diagnóstico de convergência; e aspectos computacionais em problemas práticos, com foco em implementação eficiente, estabilidade numérica e reprodutibilidade dos resultados.

Esperamos que este material sirva não apenas como texto-base para as disciplinas Estatística Computacional (graduação em Estatística) e Métodos Computacionais em Estatística (Mestrado-PPGMMQ), mas também como suporte para aqueles que desejam programar com qualidade na área de Estatística.

1 Introdução

A simulação tem um papel preponderante na estatística moderna, e suas vantagens no ensino de Estatística são conhecidas há muito tempo. Em um de seus primeiros números, o periódico *Teaching Statistics* publicou artigos que aludem precisamente a isso. Thomas e Moore (1980) afirmaram que “a introdução do computador na sala de aula escolar trouxe uma nova técnica para o ensino, a técnica da simulação”. Zieffler e Garfield (2007) e Tintle et al. (2015) discutem o papel e a importância da aprendizagem baseada em simulação no currículo de graduação em Estatística. No entanto, outros autores (por exemplo, Hodgson e Burke 2000) discutem alguns problemas que podem surgir ao ensinar uma disciplina por meio de simulação, a saber, o desenvolvimento de certos equívocos na mente dos estudantes (Martins 2018).

2 Motivação

A Estatística, além de lidar com modelos matemáticos rigorosos, também é permeada por situações em que a intuição humana falha de maneira sistemática. Um exemplo clássico é o **problema do aniversário**, que há décadas desperta curiosidade entre estudantes e pesquisadores.

O enunciado é simples: *em uma sala com r pessoas, qual a probabilidade de que pelo menos duas delas compartilhem o mesmo aniversário?*

Um resultado surpreendente é: com apenas **23 pessoas** em uma sala, a probabilidade de que haja pelo menos uma coincidência de aniversários já é **superior a 50%**. Esse resultado é tão interessante que pode ser uma porta de entrada natural para discutir a diferença entre **probabilidade teórica** e **evidência empírica obtida por simulação**.

2.1 Da teoria à simulação

Do ponto de vista teórico, a probabilidade de que todos os aniversários sejam distintos entre r pessoas é

$$\Pr(\text{todos distintos}) = \prod_{i=1}^{r-1} \frac{365-i}{365} = \left(1 - \frac{1}{365}\right) \left(1 - \frac{2}{365}\right) \cdots \left(1 - \frac{r-1}{365}\right).$$

Logo, a probabilidade de pelo menos uma coincidência é

$$p_r = 1 - \Pr(\text{todos distintos}).$$

Esse produto é conceitualmente claro, mas fica pouco manejável mentalmente para k moderados. É aqui que a **simulação computacional** pode entrar como aliada didática e científica.

2.2 Um atalho analítico útil

O produto acima admite uma **aproximação exponencial simples e acurada**, obtida tomando logaritmo e usando a expansão para argumentos pequenos:

$$\ln(1-x) = -x + o(x), \quad (x \rightarrow 0).$$

Aplicando ao produto,

$$\begin{aligned} \ln(1-p_r) &= \sum_{i=1}^{r-1} \ln\left(1 - \frac{i}{365}\right) \\ &\approx - \sum_{i=1}^{r-1} \frac{i}{365} = - \frac{1+2+\dots+(r-1)}{365} = - \frac{r(r-1)}{2 \cdot 365}. \end{aligned}$$

Exponentiando e isolando p_r , obtemos a aproximação

$$p_r \approx 1 - \exp\left\{-\frac{r(r-1)}{730}\right\}.$$

Essa fórmula tem três virtudes didáticas:

- 1) **Clareza**: exhibe explicitamente o papel do número de pares $\binom{r}{2}$.
- 2) **Rapidez**: permite cálculos mentais aproximados para valores de r de interesse.
- 3) **Boas aproximações** já para r na casa de dezenas.

Exemplo rápido:

- Para **23** pessoas:

$$p_{23}^{(\text{aprox})} = 1 - \exp\left\{-\frac{23 \cdot 22}{730}\right\} = 1 - \exp\{-0.69315\} \approx 0.500,$$

alinhando-se ao resultado clássico de que **23** pessoas já superam 50% de chance de coincidência.

2.3 O papel da simulação

A simulação estatística permite reproduzir o experimento de forma empírica: sorteamos aleatoriamente dias de aniversário para os indivíduos e verificamos se há repetições. Repetindo o processo milhares de vezes, obtemos uma estimativa para a probabilidade de coincidência.

Por exemplo, em **R**:

```
k <- 23
birthdays <- sample(1:365, k, replace = TRUE)
any(duplicated(birthdays))
```

```
[1] TRUE
```

Ao repetir esse procedimento muitas vezes (por exemplo, 10.000 simulações), podemos estimar a proporção de conjuntos com coincidência. Pela Lei dos Grandes Números, essa estimativa converge para o valor teórico de aproximadamente 0,507 quando $k = 23$.

```
set.seed(123) #reprodutibilidade

k <- 23
B <- 10000

acertos <- 0L
i <- 0L

repeat {
  i <- i + 1L
  bdays <- sample(1:365, k, replace = TRUE)
  acertos <- acertos + as.integer(any(duplicated(bdays)))
  if (i >= B) break
}

p_hat <- acertos / B
p_hat
```

```
[1] 0.5073
```


2.4 Atividade: Problema do Aniversário (22 jogadores)

Nesta motivação consideramos um exemplo discutido em Martins (2018) que é o conhecido e amplamente divulgado problema do aniversário (ver, por exemplo, Falk 2014). Martins (2018) segue o exemplo de Matthews e Stones (1998), considerando duas equipes de futebol e, portanto, coincidências de aniversário entre 22 jogadores. Martins (2018) afirma que um resultado positivo importante dessa atividade é a discussão que surgirá naturalmente entre os estudantes, com o professor atuando como mediador. Além disso, os estudantes adoram jogos e a descoberta prática, e a simulação facilita o engajamento nessas atividades, ao mesmo tempo que ilustra resultados que podem ser não intuitivos, bem como teoria geral, como a **Lei dos Grandes Números**.

Agora iremos considerar o seguinte problema:

O problema: Em uma partida de futebol, qual é a probabilidade de que pelo menos dois dos 22 jogadores façam aniversário no mesmo dia?

Em um país chamado de país do futebol, o contexto é proposital: o futebol é popular e as probabilidades resultantes são contraintuitivas. Antes de qualquer cálculo, considere as hipóteses: (i) todos os 365 dias do ano são igualmente prováveis para qualquer aniversário; (ii) as datas de aniversário dos jogadores são independentes entre si.

Objetivos

- Estimar, via simulação, a probabilidade de coincidência de aniversários.
- Relacionar frequência relativa, Lei dos Grandes Números e variação amostral.
- Comparar o resultado exato e aproximado.

Hipóteses

- 365 dias equiprováveis, datas independentes, ignorar bissexto/gêmeos.

Materiais

- R (ou Posit Cloud), roteiro com comandos `sample()`, `table()`, `mean()`.

2.5 Exercícios

- 1) Determinar o menor número de pessoas que deve estar em uma sala para que se possa apostar, com mais de 50% de chance de ganhar, que entre elas existam pelo menos duas com o mesmo aniversário.

- 2) Determinar o menor número de outras pessoas que deve estar em uma sala com você para que se possa apostar, com mais de 50% de chance de ganhar, que pelo menos uma delas tenha o mesmo aniversário que o seu.

3 Números Uniformes

As simulações, de modo geral, requerem uma base inicial formada por números aleatórios. Diz-se que uma sequência R_1, R_2, \dots é composta por números aleatórios quando cada termo segue a distribuição uniforme $U(0, 1)$ e R_i é independente de R_j para todo $i \neq j$. Embora alguns autores utilizem o termo “números aleatórios” para se referir a variáveis amostradas de qualquer distribuição, aqui ele será usado exclusivamente para variáveis com distribuição $U(0, 1)$.

3.1 Geração de sequências $U(0, 1)$

Uma abordagem é utilizar dispositivos físicos aleatorizadores, como máquinas que sorteiam números de loteria, roletas ou circuitos eletrônicos que produzem “ruído aleatório”. Contudo, tais dispositivos apresentam desvantagens:

1. **Baixa velocidade** e dificuldade de integração direta com computadores.
2. **Necessidade de reprodutibilidade** da sequência. Por exemplo, para verificação de código ou comparação de políticas em um modelo de simulação, usando a mesma sequência para reduzir a variância da diferença entre resultados.

Uma forma simples de obter reprodutibilidade é armazenar a sequência em um dispositivo de memória (HD, CD-ROM, livro). De fato, a RAND Corporation publicou *A Million Random Digits with 100 000 Random Normal Deviates* (1955). Entretanto, acessar armazenamento externo milhares ou milhões de vezes torna a simulação lenta.

Assim, a abordagem preferida é **gerar números pseudoaleatórios em tempo de execução**, via recorrências determinísticas sobre inteiros. Isso permite:

- Geração rápida;
- Eliminação do problema de armazenamento;
- Reprodutibilidade controlada.

Entretanto, a escolha inadequada da recorrência pode gerar sequências com baixa qualidade estatística.

3.2 Geradores Congruenciais Lineares

Um **Gerador Congruencial Linear (LGC)** produz uma sequência de inteiros não negativos X_i , $i = 1, 2, \dots$, por meio da relação de recorrência:

$$X_i = (aX_{i-1} + c) \bmod m, \quad i = 1, 2, \dots,$$

em que $a > 0$ é o multiplicador, $X_0 \geq 0$ é a *semente* (*seed*), $c \geq 0$ é o incremento e $m > 0$ é o módulo.

Os valores a, c, X_0 estão no intervalo $[0, m - 1]$. O número pseudoaleatório R_i é obtido por:

$$R_i = \frac{X_i}{m}, \quad R_i \in (0, 1).$$

Se m for suficientemente grande, os valores discretos $0/m, 1/m, \dots, (m-1)/m$ são tão próximos que R_i pode ser tratado como variável contínua.

3.2.1 Exemplo

Seja o gerador:

$$X_i = (9X_{i-1} + 3) \bmod 24, \quad i \geq 1.$$

Escolhendo $X_0 = 3$:

$$X_1 = (9 \times 3 + 3) \bmod 24 = 14$$

$$X_2 = (9 \times 14 + 3) \bmod 24 = 1$$

e assim por diante.

A sequência $R_i = X_i/16$ gerada terá período $\ell = 16$.

3.2.2 Implementação em R

```
# Função LCG genérica
lcg <- function(a, c, m, seed, n) {
  x <- numeric(n)
  x[1] <- seed
  for (i in 2:n) {
    x[i] <- (a * x[i-1] + c) %% m
  }
  r <- x / m
  return(list(X = x, R = r))
}

# Exemplo com a = 9, c = 3, m = 24, seed = 3
resultado <- lcg(a = 9, c = 3, m = 24, seed = 3, n = 20)
resultado$X
```

```
[1] 3 6 9 12 15 18 21 0 3 6 9 12 15 18 21 0 3 6 9 12
```

```
resultado$R
```

```
[1] 0.125 0.250 0.375 0.500 0.625 0.750 0.875 0.000 0.125 0.250 0.375 0.500
[13] 0.625 0.750 0.875 0.000 0.125 0.250 0.375 0.500
```

3.3 Geradores Congruenciais Lineares Mistos

Nos LCGs **mistos** temos $c > 0$. Uma escolha prática é $m = 2^b$, onde b é o número de bits utilizável para inteiros positivos na arquitetura/linguagem. Em muitos ambientes, inteiros usam 32 bits (um para o sinal), implicando $b = 31$ e intervalo $[-2^{31}, 2^{31} - 1]$.

Quando $m = 2^b$, obtemos **período completo** ($\ell = m$) se:

- 1) c é **ímpar** (garante $\gcd(c, m) = 1$);
- 2) $a - 1$ é múltiplo de todos os fatores primos de m e também de 4 (como m é potência de 2).

Essa é a razão de geradores simples com $m = 2^b$, c ímpar e $a \equiv 1 \pmod{4}$ atingirem $\ell = m$.

3.3.1 Questão de estouro e aritmética modular

Em linguagens com inteiros limitados, calcular $aX_{i-1} + c$ pode **transbordar**. Soluções comuns:

- usar precisão estendida (64 bits) ou bibliotecas de inteiros grandes;
- empregar **truques de aritmética modular** (como o método de Schrage) para evitar overflow;
- trabalhar com módulo $m = 2^b$ e aproveitar o “wrap” de bits.

A seguir, implementamos LCG misto com $m = 2^{31}$, $a = 906185749$, $c = 1$. Parâmetros com boas propriedades estatísticas relatadas na literatura.

3.3.2 Implementação em R (com segurança de overflow)

Para garantir a correção do módulo com inteiros grandes, usaremos `bit64` (inteiros de 64 bits) e normalizaremos para $(0, 1)$.

```
#if (!requireNamespace("bit64", quietly = TRUE)) {  
#  install.packages("bit64")  
#}  
  
library(bit64)  
  
lcg_misto <- function(n, seed = 3456L,  
                      a = 906185749L,  
                      c = 1L,  
                      m = bit64::as.integer64(2)^31) {  
  # Trabalha em integer64 para evitar perda de precisão  
  x <- bit64::as.integer64(seed)  
  outX <- bit64::integer64(n)  
  outR <- numeric(n)  
  outX[1] <- x  
  outR[1] <- as.double(x) / as.double(m)  
  for (i in 2:n) {  
    x <- (bit64::as.integer64(a) * x + bit64::as.integer64(c)) %% m  
    outX[i] <- x  
    outR[i] <- as.double(x) / as.double(m)  
  }  
  list(X = outX, R = outR)  
}
```

```
# Exemplo: primeiros 5 números com seed = 3456
set.seed(NULL)
g1 <- lcg_misto(n = 5, seed = 3456L)
g1$X
```

```
integer64
[1] 3456          746789761  460230038  1591485775 1024426876
```

```
g1$R
```

```
[1] 1.609325e-06 3.477511e-01 2.143113e-01 7.410933e-01 4.770359e-01
```

3.4 Geradores Congruenciais Lineares Multiplicativos

No caso **multiplicativo**, temos $c = 0$, e a recorrência fica:

$$X_i = (aX_{i-1}) \bmod m$$

3.4.1 Características e restrições

- Se $X_i = 0$ em algum passo, toda a sequência futura será zero — portanto $X_0 \neq 0$.
- Se $a = 1$, a sequência é constante — também deve ser evitado.
- O **período máximo** possível é $m - 1$, e ele só é atingido quando:
 1. m é primo;
 2. a é uma **raiz primitiva** módulo m .

3.4.2 Definição de raiz primitiva

Um número a é raiz primitiva módulo m se seus poderes geram todos os inteiros não nulos módulo m .

Matematicamente, a satisfaz:

$$m \nmid a^{(m-1)/q} - 1, \quad \forall q \text{ primo que divide } m - 1$$

Esse tipo de gerador é chamado **Gerador de Módulo Primo e Período Máximo**.

3.4.3 Exemplo de implementação em R

A seguir, implementamos um gerador multiplicativo com módulo primo $m = 2^{31} - 1$ (primo de Mersenne) e multiplicador $a = 630360016$, conhecido por apresentar boas propriedades estatísticas.

```
if (!requireNamespace("gmp", quietly = TRUE)) {
  install.packages("gmp")
}
library(gmp)

lcg_mult_primo <- function(n, seed, a = 630360016, m = 2147483647) {
  A <- as.bigz(a); M <- as.bigz(m)
  x <- as.bigz(seed)
  X <- integer(n); R <- numeric(n)
  for (i in seq_len(n)) {
    X[i] <- as.integer(x)
    R[i] <- as.numeric(x) / m
    x <- (A * x) %% M
  }
  list(X = X, R = R)
}

# Exemplo: gerar 10 valores
g2 <- lcg_mult_primo(n = 10, seed = 12345L)
g2$X
```

```
[1]      12345 1461144439 1646755962 423395703 2041926374 720397004
[7] 140279311 597861375 629442282 759842328
```

```
g2$R
```

```
[1] 5.748589e-06 6.803984e-01 7.668305e-01 1.971590e-01 9.508461e-01
[6] 3.354610e-01 6.532264e-02 2.784009e-01 2.931069e-01 3.538292e-01
```


4 Número Pseudoaleatórios

4.1 Introdução

4.2 Métodos para Geração de Variáveis Aleatórias Discretas

4.2.1 Método da transformação inversa

4.2.2 Método da Aceitação-Rejeição

4.2.3 Método da Composição

4.3 Métodos para Geração de Variáveis Aleatórias Contínuas

4.3.1 Método da transformação inversa

4.3.2 Método da Aceitação-Rejeição

5 Otimização Numérica

5.1 Método de Newton

5.2 Método de Newton-Raphson

5.3 Método Escore de Fisher

5.4 Método BFGS

6 Métodos de Reamostragem

6.1 Bootstrap

6.1.1 Introdução

Bootstrap é um método (computacional) de reamostragem baseado em subamostras de uma amostra observada, sendo introduzido por Efron (1979). Pode ser utilizado com o propósito de estimar erros padrão, viés de estimadores, construir intervalos de confiança, testes de hipóteses, entre outros. Pode ser utilizando sob duas abordagens: **paramétrica** e **não paramétrica**. A abordagem paramétrica exige um modelo estatística, enquanto que na abordagem não paramétrica não há suposição de modelo estatístico; toma-se por base uma distribuição empírica que atribui probabilidade $1/n$ para cada um dos n elementos da amostra. Algumas referências importantes neste tema são: Efron (1979), Wu (1986), Fisher & Hall (1989), Fredman (1986), Efron & Tibshirani (1993), Horowitz (1997), Davison & Hinkley (1997).

6.1.2 Acurária da média amostral

Experimento com 16 ratos, divididos em dois grupos: um grupo recebeu o tratamento e um outro grupo não recebeu o tratamento (controle). O tempo de sobrevivência (dias) é apresentado para cada um dos ratos. O tratamento prolonga a vida?

grupo	tempo 1	tempo 2	tempo 3	n	média	$\widehat{ep}(\text{média})$
tratamento (X)	94	197	16	7	86,86	25,24
	38	99	141			
	23					
controle (Y)	52	104	146	9	56,22	14,14
	10	51	30			
	40	27	46			
diferença					30,63	28,93

Alguns pontos importantes são:

- A resposta à pergunta dependerá de quão acurado(s) é(são) o(s) estimador(es).

- O erro padrão é uma medida (muito usual) de acurácia de estimador.
- erro padrão estimado para a média amostral \bar{X}

$$\widehat{ep}(\bar{X}) = \sqrt{\frac{s^2}{n}},$$

em que $s^2 = \sum_{i=1}^n (x_i - \bar{x})^2 / (n - 1)$.

- erro padrão de qualquer estimador é definido pela raiz quadrada de sua variância

Além disso, temos que:

- erro padrão pequeno \rightarrow acurácia alta
- erro padrão grande \rightarrow acurácia baixa
- acurácia alta (baixa) indica que o estimador apresenta valores próximos (distantes) ao seu valor esperado
- espera-se que 68% dos valores do estimador estejam a menos de um erro padrão do seu valor esperado, e 95%, a menos de dois erros padrões
- erro padrão da diferença ($\bar{X} - \bar{Y}$):

$$28.93 = \sqrt{25.24^2 + 14.14^2}.$$

Resposta à pergunta: a diferença observada 30.63 é somente 30.63/28.93=1.05 erros padrões (estimados) maior que zero, indicando um resultado não significativo, ou seja, o tratamento não aumenta o tempo médio de vida (considerando a teoria dos testes de hipóteses).

O erro padrão para o estimador média amostral apresenta fórmula conhecida, mas há casos em que não dispomos de fórmulas. Suponha que haja interesse em comparar os dois grupos de ratos em relação aos tempos medianos. Temos: md(X)=94 e md(Y)=46. A diferença é 48, maior que a diferença para as médias. Com base na mediana, o tratamento prolonga a vida?

6.1.3 Estimativa bootstrap do erro padrão

Considerações:

- $\tilde{x} = (x_1, x_2, \dots, x_n)$: vetor de dados observado (amostra original de tamanho n)
- $s(\tilde{x})$: estatística de interesse (por exemplo, média amostral)
- Uma amostra bootstrap $\tilde{x}^* = (x_1^*, x_2^*, \dots, x_n^*)$ é obtida pela amostragem aleatória de tamanho n , com reposição, de \tilde{x} . Por exemplo, com $n = 7$, poderíamos obter $\tilde{x}^* = (x_1^*, x_2^*, \dots, x_n^*) = (x_5, x_7, x_5, x_4, x_7, x_3, x_1)$.

O algoritmo bootstrap

- gerar B amostras bootstrap independentes: $x_{\sim}^{*1}, x_{\sim}^{*2}, \dots, x_{\sim}^{*B}$, cada uma de tamanho n e $50 \leq B \leq 200$.
- calcular $s(x_{\sim}^{*b})$, $b = 1, 2, \dots, B$. $s(x_{\sim}^{*b})$ é denominada réplica bootstrap de $s(x)$
- calcular $\hat{e}p_{boot} = \hat{e}p_B = \sqrt{\frac{\sum_{b=1}^B [s(x_{\sim}^{*b}) - s(\cdot)]^2}{B-1}}$, em que $s(\cdot) = \frac{\sum_{b=1}^B s(x_{\sim}^{*b})}{B}$

Sintaxe do R para calcular a estimativa bootstrap do erro padrão da média do tempo de sobrevida dos ratos do grupo tratamento.

```
set.seed(1234)
# grupo tratamento (amostra original)
x <- c(94, 197, 16, 38, 99, 141, 23)
n <- length(x)
s <- 0 # estatística de interesse
B <- 50 # no. de amostras bootstrap

for(i in 1:B){
  # réplica bootstrap para o estimador média
  s[i] <- mean(sample(x,n,replace=TRUE))
}
# estimativa bootstrap para o erro padrão da média
ep <- sd(s); ep
```

[1] 27.41873

Estimativas bootstrap do erro padrão da média e da mediana do tempo de sobrevivência dos ratos do grupo tratamento. A mediana é menos acurada (erros padrões maiores) que a média para esse conjunto de dados.

B	50	100	250	500	1000	∞
média	19.72	23.63	22.32	23.79	23.02	23.36
mediana	32.21	36.35	34.46	36.72	36.48	37.83

Formalização:

- $X_{\sim} = X_1, X_2, \dots, X_n$: amostra aleatória de uma f.d.a. F
- $x_{\sim} = (x_1, x_2, \dots, x_n)$: amostra aleatória observada de F

- $\Theta = t(F)$: parâmetro (Θ uma função de F)
- $\hat{\Theta} = s(\tilde{X})$: estimador para Θ
- $\hat{\theta} = s(\tilde{x})$: estimativa para Θ
- Quão acurado é o estimador $\hat{\Theta}$?
- Seja \hat{F} a distribuição empírica que atribui a probabilidade $1/n$ para cada valor observado x_i , $i = 1, 2, \dots, n$. A amostra bootstrap \tilde{x}^* é definida como a amostra aleatória com reposição de tamanho n extraída de \hat{F} .

$$\tilde{x}^* = (x_1^*, x_2^* \dots, x_n^*)$$

$$\hat{F} \longrightarrow (x_1^*, x_2^* \dots, x_n^*)$$

Nota:

- x^* : o símbolo $*$ indica que a amostra não é a original \tilde{x} , mas uma versão aleatorizada(reamostrada) de \tilde{x}
- a cada amostra bootstrap corresponde uma réplica bootstrap de $\hat{\theta}$, $\hat{\theta}^* = s(\tilde{x}^*)$
- $ep_F(\hat{\Theta})$ é estimado por $ep_{\hat{F}}(\hat{\Theta}^*)$, chamado **estimador bootstrap ideal** para o erro padrão $\hat{\Theta}$
- Raramente faz-se necessário $B \geq 200$ para estimar erro padrão; valores (muito) maiores são necessários, por exemplo, para IC bootstrap.
- $\lim_{B \rightarrow \infty} \hat{ep}_B = ep_{\hat{F}} = ep_{\hat{F}}(\hat{\Theta}^*)$
- O estimador bootstrap ideal $ep_{\hat{F}}(\hat{\Theta}^*)$ e sua aproximação \hat{ep}_B são chamados **estimadores bootstrap não-paramétricos**, pois baseiam-se em \hat{F} , o estimador não-paramétrico de F .
- total de amostras bootstrap distintas (combinação com repetição):

$$\binom{2n-1}{n}.$$

- No R: ver as funções `factorial()`, `choose()`, `combn()`, `combinations()`.

6.1.3.1 Exemplo

Dados de faculdades americanas de direito. População: $N = 82$ faculdades. Amostra aleatória: $n = 15$ faculdades. Variáveis analisadas: LSAT (escore médio em um teste), GPA (pontuação média na faculdade).

escola	LSAT	GPA	escola	LSAT	GPA
1	576	3,39	9	651	3,36
2	635	3,30	10	605	3,13
3	558	2,81	11	653	3,12
4	578	3,03	12	575	2,74
5	666	3,44	13	545	2,76
6	580	3,07	14	572	2,88
7	555	3,00	15	594	2,96
8	661	3,43			

- Façamos $Y=LSAT$ e $Z=GPA$. A estatística (estimador) de interesse é o coeficiente de correlação amostral entre as variáveis Y e Z :

$$\hat{\Theta} = corr(Y, Z) = \frac{Cov(Y, Z)}{DP(Y).DP(Z)} = \frac{\sum_{i=1}^n (Y_i - \bar{Y})(Z_i - \bar{Z})/n}{DP(Y).DP(Z)}$$

- Para os dados observados, a estimativa do coeficiente de correlação amostral é 0.776. Quão acurado é o estimador?

Tabela 6.4: **Estimativas bootstrap do erro padrão para $\hat{\Theta} = corr(Y, Z)$**

B	25	50	100	200	400	800	1600	3200
$\hat{e}p_B$	0,140	0,142	0,151	0,143	0,141	0,137	0,133	0,132

6.1.3.2 Exercícios:

- Considere $B=3200$. Para cada uma das 3200 amostras bootstrap, obtenha a réplica bootstrap $\hat{\theta}^* = corr(y^*, z^*)$. Faça o histograma das réplicas.
- A Tabela 3.2, p. 21 do livro texto, apresenta os dados populacionais das 82 faculdades. Selecione 3200 amostras aleatórias de tamanho $n=15$. Para cada uma dessas amostra calcule o coeficiente de correlação e faça o histograma.

No caso de valores extremos inflacionarem fortemente $\hat{e}p_B$, uma medida mais robusta para o estimador bootstrap do erro padrão é desejável. (ver Efron & Tibshirani (1993)).

Inferências baseadas na distribuição normal são “questionáveis” quando o histograma das réplicas bootstrap indica forte assimetria.

6.1.4 Bootstrap Paramétrico

O estimador bootstrap paramétrico do erro padrão é definido por

$$ep_{\hat{F}_{par}}(\hat{\Theta}^*),$$

em que \hat{F}_{par} é um estimador de F derivado do modelo paramétrico para os dados.

Para os dados das faculdades: Vamos supor que a população (LSAT, GPA) possa ser descrita por um modelo paramétrico normal bivariado F . Estimamos F por \hat{F}_{normal} , que denota a f.d.a. de uma normal bivariada com vetor de médias e matriz de covariâncias (\bar{y}, \bar{z}) e $\frac{1}{14} \begin{pmatrix} \sum (y_i - \bar{y})^2 & \sum (y_i - \bar{y})(z_i - \bar{z}) \\ \sum (y_i - \bar{y})(z_i - \bar{z}) & \sum (z_i - \bar{z})^2 \end{pmatrix}$.

O estimador bootstrap paramétrico do erro padrão da correlação $\hat{\Theta}$ será dado por $ep_{\hat{F}_{normal}}(\hat{\Theta}^*)$. Esse estimador bootstrap ideal será aproximado por \hat{ep}_B (conforme algoritmo a seguir).

O algoritmo bootstrap:

- extrair B amostras de tamanho n de \hat{F}_{par} : $\tilde{x}^{*1}, \tilde{x}^{*2}, \dots, \tilde{x}^{*B}$
- calcular $s(\tilde{x}^{*b})$, $b = 1, 2, \dots, B$. $s(\tilde{x}^{*b})$ é a réplica bootstrap de $s(x)$
- calcular $\hat{ep}_{boot} = \hat{ep}_B = \sqrt{\frac{\sum_{b=1}^B [s(\tilde{x}^{*b}) - s(\cdot)]^2}{B - 1}}$, em que $s(\cdot) = \frac{\sum_{b=1}^B s(\tilde{x}^{*b})}{B}$

No exemplo das faculdades, assumindo o modelo normal bivariado, extraímos B amostras de tamanho $n = 15$ de \hat{F}_{normal} , calculamos o coeficiente de correlação para cada amostra e, por fim, calculamos o desvio padrão desses coeficientes de correlação. Usando $B = 3200$ encontramos $\hat{ep}_B = 0.124$, que é próximo ao valor 0.131 obtido com o bootstrap não-paramétrico.

A fórmula teórica para o erro padrão do coeficiente de correlação é $\frac{1 - \hat{\Theta}^2}{\sqrt{n - 3}}$, com $\hat{\Theta} = corr(Y, Z)$. Vimos que $\hat{\theta} = 0.776$, o que resulta a estimativa 0.115 para o erro padrão do coeficiente de correlação entre as variáveis $Y=LSAT$ e $Z=GPA$.

Transformação de Fisher para o coeficiente de correlação $\hat{\Theta}$: $\hat{\zeta} = 0.5 \log \left(\frac{1 + \hat{\Theta}}{1 - \hat{\Theta}} \right)$. Assim, $\hat{\zeta}$ tem distribuição aproximadamente normal com média $0.5 \log \left(\frac{1 + \Theta}{1 - \Theta} \right)$ e variância $\frac{1}{n-3}$. O erro padrão para $\hat{\zeta}$ é $\sqrt{\frac{1}{n-3}}$. Para o exemplo das faculdades, o valor é $\frac{1}{\sqrt{12}} = 0.289$.

A título de comparação com o procedimento bootstrap, a estatística (estimador) $\hat{\zeta}$ foi estimado em cada uma das $B = 3200$ amostras bootstrap. O desvio padrão das réplicas bootstrap resultou 0.290 (muito próximo ao valor teórico 0.289). Histogramas para as correlações $\hat{\theta}^*$ e para os $\hat{\zeta}^*$.

Muitas fórmulas para os erros padrões são aproximações baseadas na teoria normal e isso “explica” os resultados próximos obtidos com o uso do bootstrap paramétrico que extrai amostras a partir da distribuição normal.

Vantagens do bootstrap sobre os métodos tradicionais:

- **bootstrap não-paramétrico:** não é necessário fazer suposições de modelos paramétricos para a população;
- **bootstrap paramétrico:** possibilita estimar erros padrões em problemas para os quais não há fórmulas para os erros padrões.

6.2 Jackknife

6.2.1 Introdução

Jackknife é uma técnica para estimar viés e erro padrão de estimadores. É uma técnica que antecede o bootstrap e foi proposta no trabalho pioneiro Quenouille (1949) para reduzir viés do estimador da correlação serial.

Formalização:

- $\tilde{X} = (X_1, X_2, \dots, X_n)$: amostra aleatória de uma f.d.a. F
- $\tilde{x} = (x_1, x_2, \dots, x_n)$: amostra aleatória observada de F
- $\Theta = t(F)$: parâmetro (Θ uma função de F)
- $\hat{\Theta} = s(\tilde{X})$: estimador para Θ
- $\hat{\theta} = s(\tilde{x})$: estimativa para Θ

O jackknife toma como base n amostras de tamanho $n - 1$ selecionadas da amostra aleatória observada. A i -ésima amostra jackknife consiste da amostra observada com a i -ésima observação removida, $i = 1, 2, \dots, n$:

$$x_{\sim(i)} = (x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n).$$

6.2.2 Estimador do viés

Para cada amostra i jackknife, é obtida a réplica jackknife $\hat{\theta}_{(i)} = s(x_{\sim(i)})$. O estimador jackknife do viés de $\hat{\Theta}$ é dado por

$$\widehat{\text{viés}}_{jack} = (n - 1)(\hat{\Theta}_{(\cdot)} - \hat{\Theta}),$$

em que $\hat{\Theta}_{(\cdot)} = \sum_{i=1}^n \frac{\hat{\Theta}_{(i)}}{n}$.

6.2.3 Estimado do erro padrão

O estimador jackknife do erro padrão de $\hat{\Theta}$ é dado por

$$\widehat{ep}_{jack} = \left[\frac{n-1}{n} \sum_{i=1}^n (\hat{\Theta}_{(i)} - \hat{\Theta}_{(\cdot)})^2 \right]^{\frac{1}{2}},$$

em que $\hat{\Theta}_{(\cdot)} = \sum_{i=1}^n \frac{\hat{\Theta}_{(i)}}{n}$.

Algumas considerações importantes:

- Bootstrap: amostragem aleatória com reposição
- Jackknife: amostras fixas
- Jackknife requer o cálculo do estimador apenas para n amostras
- A acurácia do estimador jackknife do erro padrão depende de quão próximo o estimador é da linearidade. Para funções fortemente não lineares, o jackknife pode ser ineficiente.
- Estimador linear: $\hat{\Theta} = s(X) = \mu + \frac{1}{n} \sum_{i=1}^n \alpha(X_i)$

6.2.3.1 Exemplo

Considere a amostra observada: $x = (10, 26, 30, 40, 48)$ e estimador: $\text{mediana}(\hat{\Theta})$. As amostras e réplicas jackknife são das por:

1. $x_{\sim(1)} = (26, 30, 40, 48); \hat{\theta}_{(1)} = 35$
2. $x_{\sim(2)} = (10, 30, 40, 48); \hat{\theta}_{(2)} = 35$
3. $x_{\sim(3)} = (10, 26, 40, 48); \hat{\theta}_{(3)} = 33$
4. $x_{\sim(4)} = (10, 26, 30, 48); \hat{\theta}_{(4)} = 28$
5. $x_{\sim(5)} = (10, 26, 30, 40); \hat{\theta}_{(5)} = 28$

Desta forma, a estimativa jackknife do erro padrão de $\hat{\Theta}$:

$$\widehat{ep}_{jack} = \left[\frac{4}{5} \sum_{i=1}^5 (\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)})^2 \right]^{\frac{1}{2}} \approx 6.38,$$

$$\text{com } \hat{\theta}_{(\cdot)} = \sum_{i=1}^5 \frac{\hat{\theta}_{(i)}}{5} = 31.8$$

A Sintaxe do R para calcular a estimativa jackknife do erro padrão para a mediana.

```
x <- c(10,26,30,40,48)
n <- length(x)

est.jack <- 0

for(i in 1:n){
  est.jack[i]=median(x[-i])
}

ep.jack <- sqrt(((n-1)^2/n)*var(est.jack))
ep.jack
```

```
[1] 6.374951
```

6.3 Intervalos de Confiança

6.3.1 Intervalo de Confiança Normal e t-Student

6.3.2 Intervalo de Confiança bootstrap-t

6.3.3 Intervalos de Confiança bootstrap percentil

6.3.4 Intervalos de Confiança bootstrap - versões aprimoradas

7 Métodos de Monte Carlo

7.1 Introdução

7.2 Integração de Monte Carlo

7.3 Erro de Monte Carlo

7.4 Monte Carlo via Função de Importância

7.5 Método de Máxima Verossimilhança

8 Algoritmo EM

9 Métodos Adicionais

References

- Falk, Ruma. 2014. “A Closer Look at the Notorious Birthday Coincidences”. *Teaching Statistics* 36 (2): 41–46. <https://doi.org/10.1111/test.12014>.
- Hodgson, Ted, e Maurice Burke. 2000. “On Simulation and the Teaching of Statistics”. *Teaching Statistics* 22 (3): 91–96. <https://doi.org/10.1111/1467-9639.00033>.
- Martins, Rui Manuel Da Costa. 2018. “Learning the Principles of Simulation Using the Birthday Problem”. *Teaching Statistics* 40 (3): 108–11. <https://doi.org/10.1111/test.12164>.
- Matthews, Robert, e Fiona Stones. 1998. “Coincidences: the truth is out there”. *Teaching Statistics* 20 (1): 17–19. <https://doi.org/https://doi.org/10.1111/j.1467-9639.1998.tb00752.x>.
- Thomas, F. H., e J. L. Moore. 1980. “CUSUM: Computer Simulation for Statistics Teaching”. *Teaching Statistics* 2 (1): 23–28. <https://doi.org/10.1111/j.1467-9639.1980.tb00374.x>.
- Tintle, Nathan, Beth Chance, George Cobb, Soma Roy, Todd Swanson, e Jill VanderStoep. 2015. “Combating Anti-Statistical Thinking Using Simulation-Based Methods Throughout the Undergraduate Curriculum”. *The American Statistician* 69 (4): 362–70. <https://doi.org/10.1080/00031305.2015.1081619>.
- Zieffler, Andrew, e Joan B. Garfield. 2007. “Studying the Role of Simulation in Developing Students’ Statistical Reasoning”. Em *Proceedings of the 56th Session of the International Statistical Institute (ISI)*. International Statistical Institute.