# Métodos *Bootstrap*

Lupércio França Bessegato
Dep. de Estatística/UFJF

## Roteiro Geral

1. Fundamentos de reamostragem
2. Correção *bootstrap* de viés
3. Estimação *bootstrap* de variância
4. Intervalos de confiança *bootstrap*
5. Bootstrap paramétrico
6. Bootstrap de dados com dependência
7. Redução de variância em *bootstrap* Monte Carlo
8. Referências



# Introdução ao Bootstrap

## Métodos de Reamostragem

- Métodos de permutação:
  - √ Fisher (1935); Pitman (1937, 1938)
- Jackknife
  - √ Quenouile (1949); Tukey (1958)
- Bootstrap:
  - √ Efrom (1979)

## Testes de Permutação

- Conhecido desde os anos 1930s
- Quantidade de permutações possíveis da amostra: n!
- Impedimento a seu uso
  - √ Quantidade de permutações à medida que o tamanho amostral cresce



## Bootstrap e Jackknife

- Jackknife
  - √ Em princípio útil para amostras pequenas
  - √ Pode tornar-se computacionalmente ineficiente para amostras maiores
    - (mais viável à medida que cresce a velocidade de processamento)
  - √ Efrom (1979)
    - Bootstrap construído como aproximação ao jackknife



## Bootstrap

- Amostra bootstrap:
  - √ Elementos escolhido aleatoriamente com reposição a partir da amostra original
  - √ Tem mesmo tamanho da amostra original (n)
  - √ Quantidade de reamostras possíveis: $n^n$
- Amostra aleatória do conjunto das amostras bootstrap possíveis
  - √ Maneira viável para aproximar a distribuição das amostral bootstrap



- Efrom (1979)
  - √ Conectou bootstraping com jackknife, método delta, validação cruzada e testes de permutação
- Efrom (1983)
  - √ Uso de correção de viés bootstrap com desempenho melhor que validação cruzada na estimação de taxas de erros de classificação
  - √ Variantes do bootstrap com validação cruzada e métodos de ressubstituição

- Gong (1986)
  - √ Uso de bootstrap na construção de modelo de regressão logística



## Bootstrap – Consistência

- Consistência de estimador
  - √ Aproximar-se do verdadeiro valor do parâmetro quando o tamanho amostral cresce
- Estimativa bootstrap não é consistente no sentido probabilístico
  - √ Exemplos
    - Estimação da média quando distribuição não tem variância finita
    - Estimação de máximo e mínimo



- Em geral, o bootstrap é consistente quando o Teorema Central do Limite é aplicável
- Bootstrap m-out-of-n (Bickel e Ren, 1996)
  - √ m elementos escolhidos aleatoriamente com reposição da amostra de tamanho n ($m < n$)
  - √ Extensão que supera a consistência do bootstrap



## Métodos de Reamostragem

- Objetivo:
  - √ Estimação de parâmetro populacional baseando-se apenas nos dados
- Sem suposições sobre a forma da distribuição populacional, origem dos dados

- Caso simples:
  - √ Observações independentes e identicamente distribuídas com função de distribuição acumulada F
  - √ Função de distribuição empírica ($F_n$)
    - Dá mesmo peso para cada dado (1/n)
    - Elemento básico para o bootstraping
- Interesse:
  - √ Funcionais da distribuição populacional desconhecida F
    - Maioria dos parâmetros são funcionais de F



## Exemplo

- $\mu$ e $\sigma^2$ representados como funcionais:

$$\mu = \int_{\mathrm{R}_X} x \,\mathrm{d}F(x) \qquad \sigma^2 = \int_{\mathrm{R}_X} (x-\mu)^2 \,\mathrm{d}F(x)$$

  - √ $R_X$: conjunto de valores possíveis do domínio de F
- Ideia:
  - √ Usar apenas o que é conhecido a partir dos dados
  - √ Não introduzir suposições sobre a distribuição da população



- Amostra:
  - √ F é a distribuição populacional e T(F) é o funcional que define o parâmetro
  - √ Estimação baseada em amostra iid de F, de tamanho n
    - $F_n$: função de distribuição empírica
    - $T(F_n)$: estimativa amostral do parâmetro
- Amostra bootstrap:
  - √ Amostra com reposição da amostra original
  - √ $F_n$ desempenha o papel de F
  - √ $F_n^*$: função de distribuição bootstrap
    - Desempenha o papel de $F_n$



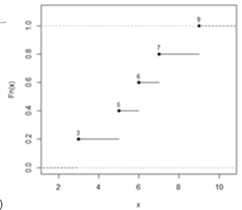
## Exemplo

- Parâmetro populacional:
  - √ $\mathrm{T}(F) = \mu$:
- Amostra original:

$$x_1 = 7; x_2 = 5; x_3 = 3; x_4 = 9; x_5 = 6$$

  - √ Estimativa parâmetro amostral: $\mathrm{T}(F_n) = \bar{x} = 6,0$
  - √ Função distribuição empírica

```
>   # Função distribuição empírica de amostra original
> original <- c(7, 5, 3, 9, 6)
> Fn <- ecdf(original)
> summary(Fn)
Empirical CDF:    5 unique values with summary
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      3       5       6       6       7       9
> knots(Fn)
[1] 3 5 6 7 9
> plot(Fn, ylim = c(0, 1.1) , main = "")
> text(knots(Fn), 1:5/5, knots(Fn), cex = 0.8, pos = 3)
```



$$x_1 = 7; x_2 = 5; x_3 = 3; x_4 = 9; x_5 = 6$$

- Amostra bootstrap:
  - √ Amostragem com reposição da amostra original

```
>   # geração de amostra bootstrap
> set.seed(666)
> amostra.boot <- sample(original, 5, replace = T)
[1] 9 7 6 5 5
> mean(amostra.boot)
[1] 6.4
```

- √ Amostra bootstrap:
  $$x_1^* = 9; x_2^* = 7; x_3^* = 6; x_4^* = 5; x_5^* = 5$$
- √ Estimativa bootstrap:
  $$\mathrm{T}(F_n^*) = \bar{x}^* = 6,4$$

18

Estatística Computacional II - 2020

$$x_1 = 7; x_2 = 5; x_3 = 3; x_4 = 9; x_5 = 6$$

- Outra amostra bootstrap:
  - √ Amostragem com reposição da amostra original

```
>   # geração de outra amostra bootstrap
> (amostra.boot <- sample(original, 5, replace = T))
[1] 9 6 3 7 5
> mean(amostra.boot)
[1] 6.0
```

- √ Amostra bootstrap:
  $$x_1^* = 9; x_2^* = 6; x_3^* = 3; x_4^* = 7; x_5^* = 5$$
- √ Estimativa bootstrap:
  $$\mathrm{T}(F_n^*) = \bar{x}^* = 6,0$$

19

Estatística Computacional II - 2020

## Distribuição Bootstrap

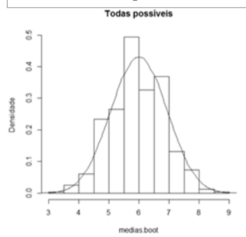
- Distribuição da estimativa do parâmetro de todas as amostras possíveis
  - √ Quantidade de amostras possíveis: $n^n$
  - √ No exemplo: $5^5 = 3.125$

20

Estatística Computacional II - 2020

√ Histograma todas as amostras com reposição

```
>   # geração de todas as amostras com reposição da original
> library(gtools)
> amostras.boot <- permutations(n = 5,  r = 5, v = original,
repeats.allowed = T)
> dim(amostras.boot)
[1] 3125    5
> medias.boot <- apply(amostras.boot, 1, mean)
> mean(medias.boot)
[1] 6
> hist(medias.boot, freq = F, ylab = "Densidade", main = "Todas possíveis")
> lines(density(medias.boot), col = "blue")
```




√ Média teórica da distribuição bootstrap é a média da amostra original.

21

Estatística Computacional II - 2020

## Distribuição Bootstrap – Aproximação Monte Carlo

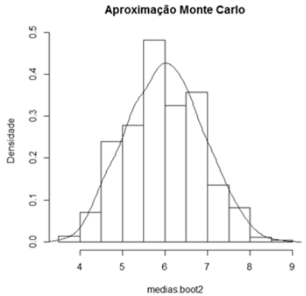
- Em geral, é inviável gerar todas as amostras com reposição possíveis
  - √ Se n = 10, $10^{10} = 10$ bilhões
- Solução:
  - √ Repetir muitas vezes o procedimento de sorteio aleatório com reposição
  - √ Construir histograma das estimativas bootstrap
  - √ Aproximação Monte Carlo da distribuição bootstrap

22

Estatística Computacional II - 2020

√ Histograma todas as amostras com reposição

```
>   # Aproximação Monte Carlo da distribuição bootstrap
> medias.boot2 <- replicate(1000, mean(sample(original, 5, replace = T)))
> mean(medias.boot2)
[1] 6.0026
> hist(medias.boot2, freq = F, ylab = "Densidade",main = "Aproximação Monte Carlo")
> lines(density(medias.boot2), col = "blue")
```




√ Média das amostras bootstrap está bem próxima de 6,0.

23

Estatística Computacional II - 2020

- Aproximação Monte Carlo da distribuição bootstrap
  - √ Permite observação da variabilidade das estimativas
  - √ Pode-se estimar
    - Assimetria, curtose, erro padrão, intervalos de confiança
  - √ Na prática usa-se aproximação Monte Carlo
  - √ Geração B = 10.000 (ou 100.000) reamostras
    - Distribuição se aproxima da distribuição bootstrap

24

Estatística Computacional II - 2020

## Procedimento

1. Geração de amostras bootstrap (com reposição) a partir da distribuição empírica dos dados originais
2. Cálculo de $T(F_n^*)$
   - √ Estimativa bootstrap de $T(F)$
3. Repetem-se os passos anteriores B vezes
   - √ B grande

25

Estatística Computacional II - 2020

## Fontes de Erro

- Aproximação Monte Carlo da distribuição bootstrap
  - √ Diminui à medida que B é grande
- Aproximação da distribuição bootstrap ($F_n^*$) à distribuição populacional F
  - √ O bootstrap funciona se $T(F_n^*) \rightarrow T(F)$, quando $n \rightarrow \infty$.
  - √ Ocorre com frequência mas não é garantido



- Em muitos casos, está demonstrada a consistência do bootstrap.
  - √ Há exemplos em que o bootstrap não é consistente
  - √ Há casos que nem a consistência nem a inconsistência estão provadas
- Usam-se simulações para confirmar ou negar a utilidade do bootstrap em casos especiais



## Aplicações

- É tentador usar o bootstrap em uma grande variedade de aplicações
  - √ Às vezes ele não funciona bem
- Solução:
  - √ Provar a consistência de acordo a um conjunto de suposições
  - √ Verificar comportamento por meio de simulações



- Algumas aplicações mais usuais:
  - √ Construção de intervalos de confiança
  - √ Estimação de parâmetros
  - √ Estimação em modelos de regressão
    - Bootstrap dos resíduos
    - Bootstrap dos vetores (pares)
    - Seleção de variáveis
  - √ Estimação de taxas de erros com ajuste de viés em problemas de classificação

- Simplicidade:
  - √ Para quase todo problema há uma maneira de gerar amostras bootstrap
- Deve-se tomar cuidados
  - √ Nem sempre percebe-se quando o bootstrap irá falhar
- Há extensões do bootstrap com modificações para contornar problemas conhecidos na estimação



## Bootstrap e a Linguagem R

```
>    # Bootsrap no R
> ??bootstrap
> help.search("bootstrap")
```



## Estimação Pontual

## Estimação Pontual

- Estimação de viés com bootstrap
  - √ Correção de viés para aprimorar estimativa
- Bootstrap foi proposto inicialmente para estimar erro padrão, sendo usado posteriormente na correção de viés
  - √ Jacknife é também usado para corrigir viés

## Estimação de Viés

- Seja $\hat{\theta}$ um estimador do parâmetro $\theta$.

$$\text{vicio}(\hat{\theta}) = \mathrm{E}(\hat{\theta} - \theta)$$

- Exemplo:
  - √ Estimador de máxima verossimilhança de $\sigma^2$ de uma variável aleatória normal univariada

$$\mathrm{S}_n^2 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n} \qquad \text{vicio}(\mathrm{S}_n^2) = -\frac{\sigma^2}{n}$$

39

Estatística Computacional II - 2020

- Estimação bootstrap para o viés:
  - √ $\hat{\theta} = S_n^2$.

$$B^* = \mathrm{E}(\theta^* - \hat{\theta}) \qquad \theta^* = \frac{\sum_{i=1}^{n}(X_i^* - \bar{X}^*)^2}{n} \qquad \bar{X}^* = \frac{\sum_{i=1}^{n} X_i^*}{n}$$

  - √ Aproximação Monte Carlo para $B^*$:

$$B_{\text{Monte}} = \frac{\sum_{i=1}^{N} B_j^*}{N}$$

  - $B_j^*$: estimativa do viés para a $j$-ésima reamostra
  - N: quantidade de amostras bootstrap

$$B_j^* = \theta_j^* - \hat{\theta} \qquad \theta_j^* = \frac{\sum_{i=1}^{n}(X_{ij}^* - \bar{X}_j^*)^2}{n}$$

40

Estatística Computacional II - 2020

- Comentários:
  - √ Geralmente o objetivo de estimar o viés é corrigir uma estimativa
    - Subtração do valor estimado de seu viés
  - √ Correção funciona quando a redução do quadrado do vício é maior que o aumento da variância
    - De outra maneira a estimativa corrigida pode ser menos precisa que a original
  - √ Correção de viés tem de ser executada com cuidado

41

Estatística Computacional II - 2020

## Exemplo

- Amostra oriunda de população normal

```
> #Correção de Viés
> set.seed(666)
>   # amostra pequena
> n <- 25
>   # amostra original
> x <- rnorm(n)
> head(x)
[1]  0.7533110  2.0143547 -0.3551345  2.0281678 -2.2168745  0.758396
>   # verdadeiro valor do parâmetro
> theta <- 1
>   # EMV da variância (variância amostral não corrigida)
> theta.hat <- function(x) var(x) * (n - 1)/n
>   # estimativa amostral do parâmetro
> (sigma2.hat <- theta.hat(x)
[1] 1.47103
>   # estimativa parâmetro, erro amostral, vies esperado
> c(sig2.amost = theta.hat(x), erro.amost = theta.hat(x) - theta, bias.esp = - 1/n)
sig2.amost erro.amost   bias.esp
 1.4710295  0.4710295 -0.0400000
```

42

Estatística Computacional II - 2020

- Aproximação Monte Carlo para o viés

```
>   # Aproximação Monte Carlo para o viés
>
>   # quantidade de reamostras bootstrap
> N <- 5000
>   # vetor com as estimativas bootstrap de EMV de sigma2
> vetor <- replicate(N, theta.hat(sample(x, n, replace = TRUE)))
>   # estimação de theta.estrela
> theta.star <- mean(vetor)
>   # estimativa bootstrap da variância e estimativa do viés
> c(theta.star, theta.star - theta.hat(x))
[1]  1.41455434 -0.05647518
```



## Estimação de Locação

- Médias amostrais
  - √ Estáveis se o 4º momento existir
  - √ Distribuições simétricas unimodais
    - Ex.: normal, t com pelo menos 3 gl
    - Média amostral é boa medida de tendência central
  - √ Estimador de máxima verossimilhança da média de algumas populações
    - Estimador consistente e de mínima variância na classe dos não viciados
    - Caso da normal e exponencial (assimetria forte)



- O que o bootstrap pode oferecer?
  - √ Desnecessária a utilização de bootstrap
    (principalmente a aproximação Monte Carlo)
  - √ A média de todas as médias bootstrap é a média amostral



## Mediana Amostral

- Populações fortemente assimétricas ou com média não definida
  - √ Mediana e moda (no caso de distribuição unimodal) representam melhor o centro da distribuição
- Cauchy:
  - √ Mediana populacional é bem definida e a mediana amostral é estimador consistente

- Mediana bootstrap é estimador consistente da mediana populacional
  - √ Mas não traz vantagem em relação à mediana amostral
- Bootstrap pode ser útil para estimar erro padrão da média e da mediana



## Estimação de Dispersão

- Desvio padrão pode ser estimado se o 2º momento existe
- Distribuições normais (ou com formato de sino)
  - √ Regra empírica baseia-se na quantidade de desvios padrão de afastamento da média



- Classe das distribuições com 2º momento definido
  - √ Desigualdade de Chebyshev:

$$\mathrm{P}\{|X-\mu|\geq k\sigma\} \leq \frac{1}{k^2}$$

  - √ Para $k = \sqrt{2}$

$$\mathrm{P}\{\mu-\sqrt{2}\sigma < X < \mu+\sqrt{2}\sigma\} > \frac{1}{2}$$

  - √ Limite na probabilidade de a observação estar k desvios padrão afastada da média (k > 1)
  - √ Ajuda na compreensão da dispersão dos dados



- Cauchy
  - √ Caudas são tão pesadas que não existe média (nem variância)
  - √ Não se aplica a desigualdade de Chebyshev
  - √ Nesses casos, pode-se usar o intervalo interquartílico como medida de variabilidade

## Estimação Bootstrap de Erro Padrão

- Seja $\widehat{\theta}$ um estimador do parâmetro $\theta$ e $\widehat{\theta}_i^*$ a estimativa bootstrap baseada na i-ésima amostra bootstrap
  - √ $\theta^*$: média dos $\hat{\theta}_i^*$s
- Estimativa boostrap do erro padrão do estimador $\widehat{\theta}$ é dada por:

$$\mathrm{SE}_b = \left\{ \frac{1}{N-1} \sum_{i=1}^{N} (\theta_i^* - \theta^*)^2 \right\}^2$$



## Estimação do Intervalo Interquartílico

- Estimador natural:
  - √ Diferença entre:
    - 75º percentil da distribuição boostrap
    - 25º percentil da distribuição bootstrap
- Usar aproximação bootstrap caso não seja possível cálculo exato.



## Intervalos de Confiança

## Intervalos de Confiança

- IC bootstrap não são exatos
  - √ Nível de confiança < nível de confiança nominal $(1 - \alpha)$
- Se o estimador bootstrap for consistente o IC bootstrap também é consistente
  - √ Nível de confiança se aproxima de $1 - \alpha$ quando $n$ cresce

## Método do Percentil de Efrom

- $\hat{\theta}_i^*$ : i-ésima estimativa bootstrap baseada na i-ésima amostra bootstrap, de tamanho n
- Procedimento:
  - √ Ordenar os dados
  - √ Identificar o centro
  - √ Tomar o $\left(1 - \frac{\alpha}{2}\right) \times 100\%$ menor valor e o $\frac{\alpha}{2} \times 100\%$ maior valor



- O método percentil não é bom para amostras pequenas e moderadas, para distribuições assimétricas ou de cauda pesada
  - √ Necessárias modificações (bootstrap de ordem superior)



## Exemplo

- Surimi
  - √ Proteína de peixe purificada usada na indústria alimentícia
  - √ Resistência do gel de surimi é fator crítico na produção
  - √ Amostra com 40 porções de surimi



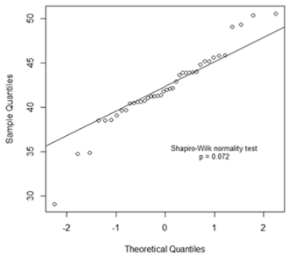
- Exploração dos dados

```
>   # amostra de 40 observações de resistência à deformação
> surimi <- c(41.28, 45.16, 34.75, 40.76, 43.61, 39.05, 41.20, 41.02, 41.33,
+   40.61, 40.49, 41.77, 42.07, 44.83, 29.12, 45.59, 41.95, 45.78,
+   42.89, 40.42, 49.31, 44.01, 34.87, 38.60, 39.63, 38.52, 38.52,
+   43.95, 49.08, 50.52, 43.85, 40.64, 45.86, 41.25, 50.35, 45.18,
+   39.67, 43.89, 43.89, 42.16)
> summary(surimi)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  29.12   40.47   41.86   42.19   44.22   50.52
> c(variancia = var(surimi), desvio = sd(surimi))
variancia    desvio
17.297605  4.159039
```

  - √ Distribuição dos dados aparenta ser normal
  - √ Leve assimetria à esquerda
    - – Distâncias assimétricas da mediana para Q1 e Q3

√ Verificação normalidade dos dados:

```
>   # teste de normalidade dos dados
> (surimi.test <- shapiro.test(surimi))
        Shapiro-Wilk normality test
data:  surimi
W = 0.94942, p-value = 0.07241
>   # qq-plot
> qqnorm(surimi); qqline(surimi)
> text(1, 35, cex = 0.8,
+ paste0(surimi.test$method,"\np = ", round(surimi.test$p.value, 3)))
```




√ Aparentemente há ocorrência de valores atípicos nas extremidades.

62

Estatística Computacional II - 2020

- Intervalo de confiança t:

```
>   # IC t com 95% - assumindo normalidade (tamanho amostral)
> t.test(surimi)$conf.int[1:2]
[1] 40.85562 43.51588
```

√ Assumindo normalidade e tamanho amostral

√ Espera-se que seja uma boa aproximação

63

Estatística Computacional II - 2020

**Método Percentil t de Efrom**

- Suponha um parâmetro $\theta$ e uma estimativa $\theta_h$ para ele, obtida por amostra original de tamanho n
- Seja $\theta^*$ a estimativa bootstrap baseada na amostra original
- Suponha que haja um estimativa $S_h$ do desvio padrão de $\theta_h$ e uma estimativa bootstrap $S^*$ para $S_h$
  √ $S^*$ é específica à uma amostra bootstrap

64

Estatística Computacional II - 2020

- Seja a estatística bootstrap $T^*$:

$$\mathrm{T}^* = \frac{\theta^* - \theta_h}{S^*}$$

√ Versão bootstrap padronizada e centrada

√ Análoga a $\mathrm{T} = \dfrac{\theta_h - \theta}{S_h}$

√ Se $\theta$ é a média populacional e $\theta_h$, a média amostral

– T é uma quantidade pivotal se a amostra é normalmente distribuída

$(T \sim t_{n-1})$

√ Quantidade pivotal: quantidade aleatória que não depende de parâmetro desconhecido

65

Estatística Computacional II - 2020

- Pivoteamento:

$$\begin{aligned}P\{-c < T_{n-1} < c\} &= P\left\{-c < \frac{\theta_h - \theta}{S_h} < c\right\}\\ &= P\{-\theta_h - cS_h < -\theta < -\theta_h + cS_h\}\\ &= P\{\theta_h - cS_h < \theta < \theta_h + cS_h\}\end{aligned}$$

- No caso da estatística bootstrap $T^*$ para a média populacional:
  - √ $T^*$ é assintoticamente pivotal
    - A distribuição se torna independente dos parâmetros e seus percentis convergem para os percentis da distribuição t
  - √ Construção de intervalos bootstrap mais precisos que os obtidos pelo método percentil

66

Estatística Computacional II - 2020

- Caso mais geral:
  - √ Seja $\theta$ um parâmetro mais complicado que a média
  - √ $\theta^*$ : estimativa bootstrap que necessita aproximação Monte Carlo para gerar o intervalo de confiança

67

Estatística Computacional II - 2020

- Procedimento:
  - √ Para cada uma de B amostras bootstrap, há uma estimativa $\theta^*$ e pode-se calcular sua estatística $T^*$
  - √ Ordenam-se os B valores de $T^*$
  - √ Intervalo aproximado com $100(1-2\alpha)\%$ de confiança é obtido por

$$(\theta_h - \mathrm{T}^*_{(1-\alpha)}\, S_h; \theta_h + \mathrm{T}^*_{\alpha}\, S_h)$$

    - $t^*$: $100(1-2\alpha)$ percentil de uma $t_{n-1}$
    - $S^*$: estimativa bootstrap do desvio padrão de $\theta$

68

Estatística Computacional II - 2020

## Intervalo de Confiança Boostrap t

- Hesterberg et al. (2003)
  - √ Usa o bootstrap para estimar o erro padrão
  - √ Recomendado apenas se a distribuição bootstrap for aproximadamente normal
  - √ É menos geral que o procedimento percentil de Efrom

69

Estatística Computacional II - 2020

- Procedimento para intervalo com $100(1-2\alpha)\%$ de confiança:

$$(\theta_h - t^* S^*; \theta_h + t^* S^*)$$

  - $t^*_{1-\alpha}$: percentil $100\alpha$ dos $T^*$s
  - $S^*$: estimador bootstrap do desvio padrão de $\theta$
- Limitação:
  - √ São necessários $S_h$ e a versão bootstrap $S^*$
- Solução:
  - √ Quando $\theta$ é um parâmetro complicado usar double bootstrap (ou nested ou iterated)



## Exemplo

- Intervalo de confiança bootstrap para $\mu$:
  - √ Conjunto de dados `surimi`:
  - √ Erro padrão de $\bar{x}$: $S_h = \dfrac{S}{\sqrt{n}}$
    - Aceitável $S_h = \frac{S}{\sqrt{n}}$ para amostras moderadas ou grandes (n ≥ 30)

```
>   # intervalo de confiança bootstrap
> source("surimi.R")
> set.seed(666)
> n <- length(surimi)
> (u0 <- mean(surimi) )
[1] 42.18575
> (Sh <- sd(surimi)/sqrt(n))
[1] 0.6576018
```



- IC bootstrap – método percentil

```
>   # qute de amostras bootstrap
> B <- 1000
>   # cálculo média e estatística t por amostra bootstrap
> est.boot <- function(x) {list(med = mean(x), te = sqrt(n)*(mean(x)-u0)/sd(x))}
> matriz <- replicate(B, est.boot(sample(surimi, n, replace = T)))
> amostras.boot <- matrix(unlist(matriz),ncol = 2, byrow=T)
> colnames(amostras.boot) <- c("theta", "t")
> theta.star <- amostras.boot[, "theta"]
> t.star <- amostras.boot[, "t"]
>   # compara média amostral com as médias bootstrap
> c(u0, mean(theta.star))
[1] 42.18575 42.17262
>   # comapara erro padrão da amostra com erro padrão bootstrap
> c(Sh, sd(theta.star))
[1] 0.6576018 0.6455912
>   # aparenta simetria e normalidade?
> summary(t.star)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-2.85347 -0.64650  0.02182  0.01559  0.64791  3.57593
```

$\theta_h = 42,186$

$\theta^* = 42,173$

$S_h = 0,658$

$S^* = 0,646$

- √ $\theta_h$ é bastante próxima da média bootstrap
- √ Erros padrão não tão próximos
- √ $t^*$ aparentemente simétrico



- IC bootstrap – método percentil

```
>   # quantis dos percentis t bootstrap 2.5% 97.5%
> quantile(t.star, probs = c(0.025,0.975))
     2.5%     97.5%
-1.954212  2.214949
>   # quantis de t com n-1 graus de liberdade
> qt(c(0.025,0.975), n - 1)
[1] -2.022691  2.022691
>   # intervalo de confiança bootstrap percentil t
> u0 + quantile(t.star, probs = c(0.025,0.975)) * Sh
    2.5%    97.5%
40.90066 43.64230
>   # intervalo de confiança t de student
> u0 + qt(c(0.025,0.975), n - 1) * Sh
[1] 40.85562 43.51588
```

$\text{quantis}_{t^*} = (-1,95; 2,21)$

$\text{quantis}_{t_{39}} = (-2,02; 2,02)$

$\text{IC}_b = (40,90; 43,64)$

$\text{IC}_t = (40,86; 43,52)$

- √ Quantis da $t^*$ são diferentes de quantis da $t_{39}$
  - Refletem assimetria da distribuição subjacente.
- √ Proximidade dos dois intervalos sugere precisão de ambos

- IC bootstrap – pacote “`bootstrap`”

```
> # usando o pacote bootstrap
>
> set.seed(666)
> library("bootstrap")
>   # função para erro padrão da amostra bootstrap
> sdmean <- function(x, ...) {sqrt(var(x)/length(x))}
>   # função para cálculo IC bootstrap
> boott(surimi, theta = mean, sdfun = sdmean, nboott = 1000,
+ perc = c(0.025,0.975) #bootstrap percentile t
+ )
$`confpoints`
        0.025    0.975
[1,] 40.90306 43.48429
```

$IC_b = (40,90; 43,64)$
$IC_{bp} = (40,90; 43,48$
$IC_t = (40,86; 43,52)$

√ Resultado próximo dos resultados anteriores

74

Estatística Computacional II - 2020

## Bootstrap Iterado

- Há numerosos procedimentos para iteração bootstrap
- Intervalos de confiança aproximados com precisão de 1ª ordem:
  - √ Diferença entre verdadeira probabilidade de cobertura e probabilidade de cobertura limite tende a zero a uma taxa $n^{-1/2}$.
  - √ Alguns procedimentos:
    - Padrão
    - Percentil bootstrap

75

Estatística Computacional II - 2020

- Intervalos de confiança aproximados com precisão de 2ª ordem:
  - √ Diferença entre verdadeira probabilidade de cobertura e probabilidade de cobertura limite tende a zero a uma taxa $n^{-1}$.
  - √ Alguns procedimentos:
    - Percentil t
    - BCa
    - Bootstrp duplo
  - √ Dados 2 Ics de precisão de 2ª ordem. Qual a melhor escolha
    - Menor comprimento esperado

76

Estatística Computacional II - 2020

## Exemplo

- Intervalo de confiança bootstrap duplo para $\mu$:
  - √ Conjunto de dados `surimi`:

```
> # double bootstrap - "bootstrap" package
> set.seed(666)
>   # IC bootstrap 95% - percentile t c/ bootstrap aninhado
> boott(surimi, theta = mean, nbootsd = 100, nboott = 1000,
+ perc = c(0.025,0.975))
$`confpoints`
        0.025    0.975
[1,] 40.78245 43.52531
```

$IC_b = (40,90; 43,64)$
$IC_{bp} = (40,90; 43,48$
$IC_t = (40,86; 43,52)$

  - √ Estimativa boostrap do erro padrão
    - Bootstrap interno de 100 reamostras
  - √ Estimativa bootstrap do IC
    - Bootstratp externo com om 1.000 reamostras

78

Estatística Computacional II - 2020

- Comentários:

$$\mathrm{IC}_b = (40,90;43,64)$$
$$\mathrm{IC}_{bp} = (40,90;43,48$$
$$\mathrm{IC}_{bd} = (40,78;43,52$$
$$\mathrm{IC}_t = (40,86;43,52)$$

√ Intervalo com 95% de confiança bastante próximo dos anteriores

√ Não exige fórmula para o erro padrão do estimador

√ Pode ser usado com qualquer estatística

- Custo computacional pode ser alto



## Intervalo de Confiança Bootstrap com Correção de Viés

- Incorporam procedimento para correção de vício do bootstrap
- Alguns procedimentos:

√ Bootstrap BC (*Bias correction*)

- Trabalha bem com coeficiente de correlação bivariado
  - Nesse caso o método percentil não trabalha bem
- Não tem precisão de 2ª ordem



√ Bootstrap BCa:

- Incorpora constante de aceleração na correção de viés
- Baseado no 3º momento
  - Corrige assimetria

√ Percentil ajustado (Davison e Hinkely, 1997)

- Tem precisão de 2ª ordem

√ ABC

- Muito próxima da precisão do BCa
- Embora tenha um parâmetro a mais, é mais simples e mais rápido que o método BCa.



## Exemplo

- Intervalo de confiança bootstrap BCa:

√ Pacote `boot`:

```
 library("boot")
> # correção de viés - bootstrap BCa e ABC
>
>
> library("boot")
> set.seed(666)
> # IC bootstrap BCa
>   # estimação dados os dados x e o conjunto de índices i
> fboot <- function(x, i) mean(x[i])
>   # gera as estimativas bootstrap
> bs <- boot(surimi, fboot, R = 1000)
>   # IC 95% bootstrap BCa
> boot.ci(bs, type = "bca" , conf = 0.95)
Intervals :
Level       BCa
95%   (40.73, 43.33 )
```

$$\mathrm{IC}_b = (40,90;43,64)$$
$$\mathrm{IC}_{bp} = (40,90;43,48$$
$$\mathrm{IC}_{bd} = (40,78;43,52$$
$$\mathrm{IC}_t = (40,86;43,52)$$

√ Estimativa comparável com as anteriores

- Intervalo de confiança bootstrap ABC:
  - √ Pacote "`boot`"

```
> # IC bootstrap ABC
>
>   # usa média pondereada
> fabc <- function(x, w) w %*% x
>   # IC 95% bootstrap ABC
> abc.ci(surimi, fabc, conf = 0.95)
[1]  0.95000 40.84506 43.39569
```

$$\mathrm{IC}_b = (40,90;43,64)$$
$$\mathrm{IC}_{bp} = (40,90;43,48$$
$$\mathrm{IC}_{bd} = (40,78;43,52$$
$$\mathrm{IC}_{BCa} = (40,73;43,33)$$
$$\mathrm{IC}_{ABC} = (40,85;43,40)$$
$$\mathrm{IC}_t = (40,86;43,52)$$

  - √ Resultado próximo dos anteriores



## Bootstrap Paramétrico

## Bootstrap Paramétrico

- Assume-se que F pertence a uma família paramétrica
- Amostragem com reposição a partir dessa distribuição
  - √ Se é usado o método de máxima verossimilhança para estimar os parâmetros de F, a abordagem é essencialmente a mesma que a de Máxima Verossimilhança

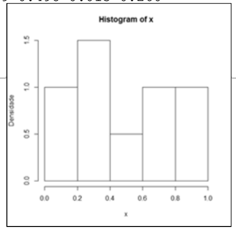


- Em geral, executar o bootstrap acrescenta pouco nos problemas paramétricos
- Em problemas complexos, pode ser útil pelo menos uma parametrização parcial
  - √ Modelo de riscos proporcionais de Cox
- Comparação entre bootstrap paramétrico e não paramétrico pode auxiliar na verificação das suposições paramétricas

√ Exemplo:

– Qual é o valor esperado da mediana de uma amostra aleatória de tamanho n = 51, de uma população Exponencial (1)? E sua variância?

```
> # geração de uma amostra uniforme
> set.seed(666)
> options(digits = 2)
> (x <- runif(10))
 [1] 0.774 0.197 0.978 0.201 0.361 0.743 0.979 0.498 0.013 0.260
> hist(x, freq = F, ylab = "Densidade")
> mean(x)
[1] 0.5
```





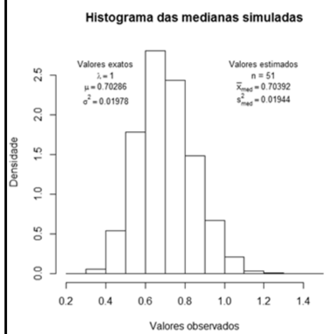

√ Mediana amostral

– Valor estimado da esperança e variância

```
> # valor estimado da esperança e da variância da mediana amostral
> set.seed(666)
> medianas <- replicate(10000, median(rexp(n = ene, rate = taxa)))
> (media <- mean(medianas))
 [1] 0.7039184
> (variancia <- var(medianas))
[1] 0.01943892
> # histograma das 10.000 réplicas
> hist(medianas, freq = F, xlab = "Valores observados", ylab =
"Densidade",
+ main = "Histograma das medianas simuladas")
> text(0.4, 2.65, "Valores exatos", cex = 0.8)
> text(0.4, 2.50, expression(lambda == 1), cex = 0.8)
> text(0.4, 2.35, expression(mu == 0.70286), cex = 0.8)
> text(0.4, 2.20, expression(sigma^2 == 0.01978), cex = 0.8)
> text(1.2, 2.65, "Valores estimados", cex = 0.8)
> text(1.2, 2.5, paste("n =", ene), cex = 0.8)
> text(1.2, 2.35, bquote(bar(x)[med] == .(round(media, 5))), cex = 0.8)
> text(1.2, 2.20, bquote(s[med]^2 == .(round(variancia, 5))), cex = 0.8)
```



√ Histograma dos valores observados




√ Construção de intervalo de confiança para a mediana.

√ Poderia ser aproximada pela normal?



**Modelos de Regressão por Bootstrap**

## Estimação Bivariada

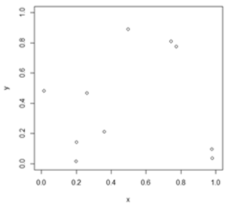
- Reamostragem de mais de uma variável:
  - √ Medidas de várias variáveis por indivíduo
    - Reamostrados os indivíduos (linhas)



## Exemplo

- Estimação de correlação

```
> # reamostragem de mais de uma variável
> set.seed(666)
> (xy <- data.frame(x = runif(10), y = runif(10)))
       x     y
1  0.774 0.776
2  0.197 0.016
3  0.978 0.096
4  0.201 0.142
5  0.361 0.211
6  0.743 0.811
7  0.979 0.037
8  0.498 0.892
9  0.013 0.483
10 0.260 0.467
> plot(y ~x, data = xy, xlim = c(0, 1), ylim = c(0, 1))
> # correlação original
> cor(xy$x, xy$y)
[1] 0.043
>   # IC paramétrico para correlação
> cor.test(xy$x, xy$y)$conf.int[1:2]
[1] -0.6030740  0.6547849
```





√ Reamostragem de uma amostra:

```
> # uma amostra bootstrap
> (xy.boot <- xy[sample(1:nrow(xy), replace = TRUE),])
        x     y
10   0.260 0.467
7    0.979 0.037
1    0.774 0.776
2    0.197 0.016
9    0.013 0.483
3    0.978 0.096
1.1  0.774 0.776
7.1  0.979 0.037
6    0.743 0.811
9.1  0.013 0.483
> # estimativa bootstrap da correlação
> cor(xy.boot$x, xy.boot$y)
[1] -0.14
```

```
> xy
   x     y
1  0.774 0.776
2  0.197 0.016
3  0.978 0.096
4  0.201 0.142
5  0.361 0.211
6  0.743 0.811
7  0.979 0.037
8  0.498 0.892
9  0.013 0.483
10 0.260 0.467
```



√ Reamostragem de duas amostras:

```
> # duas amostras bootstrap
> (xy.boot2 <- replicate(2, xy[sample(1:nrow(xy),replace=T),], simplify = F))
  [[1]]
        x     y
5    0.361 0.211
4    0.201 0.142
9    0.013 0.483
6    0.743 0.811
7    0.979 0.037
5.1  0.361 0.211
3    0.978 0.096
10   0.260 0.467
8    0.498 0.892
6.1  0.743 0.811
```

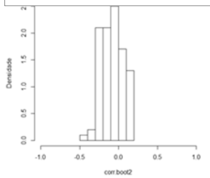
```
[[2]]
        x     y
3    0.978 0.096
6    0.743 0.811
2    0.197 0.016
10   0.260 0.467
9    0.013 0.483
8    0.498 0.892
4    0.201 0.142
7    0.979 0.037
6.1  0.743 0.811
10.1 0.260 0.467
```

```
> is.list(xy.boot2)
[1] TRUE
> sapply(xy.boot2, function(mat) cor(mat$x, mat$y))
[1] -0.0515  0.0025
```

```
> xy
   x     y
1  0.774 0.776
2  0.197 0.016
3  0.978 0.096
4  0.201 0.142
5  0.361 0.211
6  0.743 0.811
7  0.979 0.037
8  0.498 0.892
9  0.013 0.483
10 0.260 0.467
```

√ Reamostragem com 100 réplicas:

```
> # 100 amostras bootstrap, com cálculo da correlação de cada reamostra
> xy.boot <- replicate(100, xy[sample(1:nrow(xy),replace=T),], simplify
= F)
> corr.boot <- sapply(xy.boot, function(mat) cor(mat$x, mat$y))
> hist(corr.boot, freq = F, ylab = "Densidade")
```




√ A precisão da estimativa do coeficiente de correlação amostral melhorou?

– Estimação de intervalo com 95% de confiança:

```
> # intervalo com 95% de confiança aproximado para a média
> quantile(corr.boot2, probs = c(0.025, 0.975))
 2.5%   98%
-0.31  0.17
```

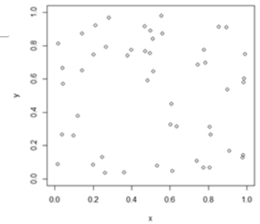
$\mathrm{IC}_{\text{par.}} = (-0,60; 0,65)$



√ Reamostragem de mais de uma variável:

– Amostra bivariada original de tamanho 50

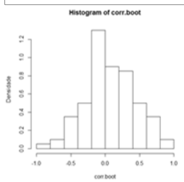
```
> # amostra bivariada de tamanho 50
> set.seed(666)
> options(digits = 2)
> xy2 <- data.frame(x = runif(50), y = runif(50))
> head(xy2)
     x     y
1 0.77 0.069
2 0.20 0.085
3 0.98 0.130
4 0.20 0.746
5 0.36 0.039
6 0.74 0.686
> plot(y ~ x, data = xy2, xlim = c(0, 1), ylim = c(0, 1))
> # correlação original
> cor(xy2$x, xy2$y)
[1] -0.077
```





– Reamostragem com 100 réplicas:

```
> # 100 amostras bootstrap, com cálculo da correlação de cada reamostra
> xy2.boot <- replicate(100, xy2[sample(1:nrow(xy2),replace = T),],
simplify = F)
> corr.boot2 <- sapply(xy2.boot, function(mat) cor(mat$x, mat$y))
> hist(corr.boot2, freq = F, ylab = "Densidade")
```




√ Como seria uma estimativa bootstrap da correlação se amostra fosse maior?

– Estimação de intervalo com 95% de confiança:

```
> # intervalo com 95% de confiança aproximado para a média
> quantile(corr.boot, probs = c(0.025, 0.975))
 2.5%   98%
-0.61  0.77
```



## Bootsrapping em Regressão

- Estimação bootstrap de modelo de regressão:
  - √ Bootstrap de vetores (ou pares)
  - √ Bootstrap de resíduos
- Ambas abordagens podem ser usadas tanto em regressão linear quanto não linear

## Regressão Linear

- Estimadores de mínimos quadrados:
  - √ Modelo é razoável se o termo de erro pode ser considerado iid, com média zero e variância constante $\sigma^2$
  - √ Estimação bootstrap não agregará nada
- Teorema de Gauss-Markov
  - √ EMQ dos parâmetros de regressão são não viciadas, com a menor variância na classe dos estimadores lineares não viciados



- Matriz de covariâncias de $\widehat{\boldsymbol{\beta}}$:
  - √ $\widehat{\boldsymbol{\beta}}$: EMQ de $\boldsymbol{\beta}$.

$$\boldsymbol{\Sigma} = \sigma^2(\mathbf{XX})^{-1}, \text{ se } (\mathbf{XX})^{-1} \text{ existir.}$$

  - √ Se $\hat{\sigma}^2$ é o EMQ da variância residual, o estimador usual de $\boldsymbol{\Sigma}$ é: $\hat{\Sigma} = \hat{\sigma}^2(\mathbf{XX})^{-1}$.
- Caso o erro possa ser considerado normal
  - √ EMQ tem a propriedade adicional de ser estimador de mínimos quadrados
  - √ Estimador mais eficiente



## Violações do Modelo Normal

- As estimativas não são robustas se as hipóteses do modelo forem violadas
  - √ Erros com cauda pesada ou com outliers
    - EMQ darão muito peso aos outliers, tentando ajustá-los em detrimento do restantes dos dados
    - Outliers têm grande influência nos parâmetros da regressão quando sua remoção acarreta mudança importante dos parâmetros
  - √ Procedimentos robustos a outliers:
    - Desvios absolutos mínimos, Estimação-M, medianas repetidas



- Independente do procedimento de estimação dos parâmetros da regressão
  - √ Se interesse é construção de IC's para parâmetros e IP's para observações futuras
    - Necessário conhecimento sobre distribuição de $\boldsymbol{\epsilon}$
  - √ Caso distribuição do erros seja normal
    - IC's e IP's são calculados diretamente
- Bootstrap é útil na construção de IC's e IP's em modelos de regressão.

- Outras complicações que podem ser solucionadas pelo bootstrap
  - √ Heterocedasticidade
  - √ Não linearidade dos parâmetros do modelo
  - √ Viés devido a transformações



## Bootstrap de Vetores (ou Pares)

- Amostra original:
  - √ n vetores com dimensão p+1:
    $(y_i, x_{1i}, x_{2i}, \ldots, x_{pi}),\ i = 1, 2, \ldots, n.$
- Amostra bootstrap:
  - √ Reamostragem com reposição de n vetores de dimensão p+1
  - √ Ajuste de modelo em cada amostra bootstrap



- Considera que as observações são iid, possivelmente com estrutura de correlação
- Método é mais robusto a desvios de normalidade e/ou na presença de erro de especificação dos termos do modelo



## Bootstrap dos Resíduos

- Amostra original:
  - √ Ajusta-se um modelo aos dados, calculando-se os resíduos do modelo
- Amostra bootstrap:
  - √ Reamostragem com reposição dos resíduos
  - √ Obtenção de pseudo amostra com estimação do resíduo bootstrap às estimativas
  - √ Ajuste de modelo à pseudo amostra

- Modelo:

$$y_i = g_i(\boldsymbol{\beta}) + \epsilon_i,\ i = 1, 2, \ldots, n.$$

  √ $g_i$: função de forma conhecida para um conjunto de covariáveis ($X_1$, $X_2$, ..., $X_p$)
  - Para regressão linear, é a mesma função para cada i e é linear nos componentes de $\boldsymbol{\beta}$.

  √ $\boldsymbol{\beta}$ : vetor de dimensão p associados às covariáveis

  √ $\epsilon_i$ : erro iid de uma distribuição F desconhecida

  √ Assume-se que F está centrada em zero
  - Em geral, exige-se que a mediana seja zero



- Estimação de $\boldsymbol{\beta}$:

  √ Determinar valores que minimizam uma distância de $\lambda(\boldsymbol{\beta})$ até os dados observados $(y_1, y_2, \ldots, y_n)$

$$\lambda(\boldsymbol{\beta}) = (g_1(\boldsymbol{\beta}), g_2(\boldsymbol{\beta}), \ldots, g_n(\boldsymbol{\beta}))$$

  √ $D(y, \lambda(\boldsymbol{\beta}))$: medida de distância:
  - Critério dos mínimos quadrados:

$$\mathrm{D}(y, \lambda(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - g_i(\boldsymbol{\beta})^2.$$

  - Critério dos mínimos desvios absolutos

$$\mathrm{D}(y, \lambda(\boldsymbol{\beta}) = \sum_{i=1}^{n} |y_i - g_i(\boldsymbol{\beta}|.$$



- Estimativa $\widehat{\boldsymbol{\beta}}$ são os valores de $\boldsymbol{\beta}$ tais que:

$$\mathrm{D}(y, \lambda(\hat{\boldsymbol{\beta}}) = \min_{\boldsymbol{\beta}} \mathrm{D}(y, \lambda(\boldsymbol{\beta}).$$

- Resíduos são definidos como:

$$\hat{\epsilon}_i = y_i - g_i(\hat{\boldsymbol{\beta}}),\ \text{para}\ i = 1, 2, \ldots, n.$$



- Geração dos resíduos bootstrap:

  √ Utiliza-se a função de distribuição empírica para os resíduos
  - Probabilidade de 1/n para cada resíduo $\epsilon_i$
  - Escolha com reposição de amostra com n resíduos

$$(\epsilon_1^*, \epsilon_2^*, \ldots, \epsilon_n^*).$$

  - Geração de amostra bootstrap de observações

$$y_i^* = g_i(\hat{\boldsymbol{\beta}}) + \epsilon_i^*,\ i = 1, 2, \ldots, n.$$

  - Para cada amostra bootstrap estima-se $\boldsymbol{\beta}^*$ tal que:

$$\mathrm{D}(y^*, \lambda(\boldsymbol{\beta}^*) = \min_{\boldsymbol{\beta}} \mathrm{D}(y^*, \lambda(\boldsymbol{\beta}).$$

- Aproximação Monte Carlo:
  - √ Repetição do processo B vezes
- Estimativa bootstrap para a matriz de covariâncias de $\widehat{\boldsymbol{\beta}}$:

$$\boldsymbol{\Sigma}^* = \frac{1}{B-1}\sum_{j=1}^{B}(\boldsymbol{\beta}_j^* - \boldsymbol{\beta}^*)(\boldsymbol{\beta}_j^* - \boldsymbol{\beta}^*)'$$

  - √ onde:
    - $\boldsymbol{\beta}_j^*$: estimativa de $\boldsymbol{\beta}$ da j-ésima amostra bootstrap

$$\boldsymbol{\beta}^* = \frac{1}{B}\sum_{j=1}^{B}\boldsymbol{\beta}_j^*.$$



- Bootstrapping de resíduos:
  - √ Aplicável quando for assumido modelo paramétrico de regressão
  - √ Funciona bem quando os termos dos resíduos são normais
  - √ Na prática podemos não estar seguros de que a forma paramétrica está correta
    - Nesse caso, melhor usar bootstrap de vetores



## Bootstrap Vetores e Resíduos

- Comparação dos métodos:
  - √ São assintoticamente equivalentes se modelo está correto
  - √ Desempenho pode ser diferente em amostras pequenas
  - √ Bootstrap por vetores é menos sensível às hipóteses do modelo
    - Pode funcionar razoavelmente quando hipóteses são violadas
    - Método usa o modelo explicitamente o modelo



## Heterocedasticidade

- Métodos que funcionam bem quando a variância residual é heterocedástica
  - √ Bootstrap por vetores
    - Funciona melhor que o bootstrap de resíduos
  - √ Resíduos bootstrap modificados
    - *Wild bootstrap*

## Regressão Não Linear

- Modelos não-lineares
  - √ Permitem aproximações locais lineares por meio de expansões por séries de Taylor
  - √ Classe de modelos altamente não lineares em que as aproximações lineares não funcionam
- Efrom (1982):
  - √ Bootstrap pode ser aplicado a quase todo problema não-linear
    - Não necessitam ter formas não-lineares diferenciáveis



- Exemplos:
  - √ Modelo linear: $f(x,\theta) = \theta_1 + \theta_2\, \mathrm{e}^{x} + \epsilon.$
  - √ Modelo não-linear: $g(x,\theta) = \theta_1 + \theta_2\, \mathrm{e}^{\theta_3 x} + \epsilon.$
    - $e^{\theta_3 x}$ não é função linear de $\theta_3$.



## Exemplo

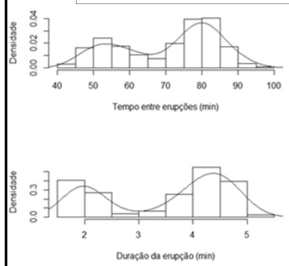
- Conjunto de dados `faithful`:
  - √ Tempo entre erupções e duração de erupções de gêiser denominado Old Faithful

```
>   # carregamento do conjunto de dados
> help(faithful)
> str(faithful)
'data.frame':   272 obs. of  2 variables:
 $ eruptions: num  3.6 1.8 3.33 2.28 4.53 ...
 $ waiting  : num  79 54 74 62 85 55 88 85 51 85 ...
> dim(faithful)
[1] 272   2
> head(faithful)
eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
4     2.283      62
```



√ Histograma das variáveis:

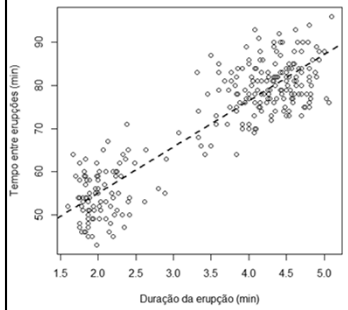
```
> par(mfrow = c(2, 1))
>   # eruptions
> hist(faithful$waiting, freq = F, ylab = "Densidade", main = "",
+ xlab = "Tempo entre erupções (min)")
> lines(density(faithful$waiting), col = "blue")
>   # durations
> hist(faithful$eruption, freq = F, ylab = "Densidade", main = "",
+ xlab = "Duração da erupção (min)")
> lines(density(faithful$eruption), col = "blue")
> par(mfrow = c(1, 1))
```




√ Variáveis com bimodalidade

√ Plot das variáveis:

```
>   # plot dos dados
> plot(waiting ~ eruptions, data = faithful, ylab = "Tempo entre erupções (min)",
+ xlab = "Duração da erupção (min)")
>   # modelo linear
> ml.1 <- lm(waiting ~ eruptions, data = faithful)
> abline(ml.1, col = "blue", lty = 2, lwd = 2)
```




- √ Apresenta dois clusters
  - Regiões de bimodalidade



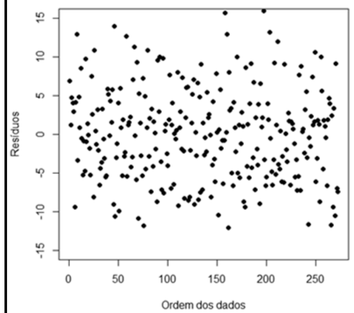
√ Ajuste do modelo linear

```
>   # modelo linear
> ml.1 <- lm(waiting ~ eruptions, data = faithful)
> summary(ml.1)
Residuals:
     Min       1Q   Median       3Q      Max
-12.0796  -4.4831   0.2122   3.9246  15.9719

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.4744     1.1549   28.98   <2e-16 ***
eruptions    10.7296     0.3148   34.09   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.914 on 270 degrees of freedom
Multiple R-squared:  0.8115,    Adjusted R-squared:  0.8108
F-statistic:  1162 on 1 and 270 DF,  p-value: < 2.2e-16
>   # intervalos de confiança parâmetros
> confint(ml.1)
               2.5 %   97.5 %
(Intercept) 31.20069 35.74810
eruptions   10.10996 11.34932
```



√ Verificação do modelo:
- Independência dos resíduos

```
>   # Independência dos resíduos
> plot(ml.1$residuals, pch = 16, ylim = c(-15, 15), ylab = "Resíduos",
+ xlab = "Ordem dos dados")
```

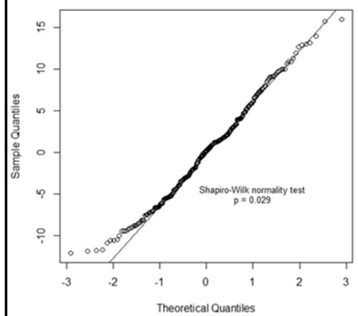



- √ Desvio em relação aos valores positivos dos resíduos



√ Verificação do modelo:
- Normalidade

```
>   # normalidade dos resíduos
> qqnorm(ml.1$residuals); qqline(ml.1$residuals)
```

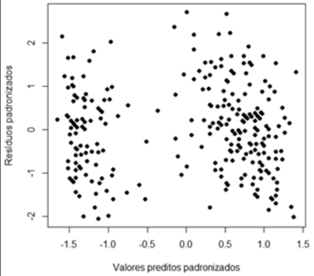



- √ Desvios acentuados na extremidade inferior
- √ Rejeita-se a hipótese de normalidade

√ Verificação do modelo:

– Homogeneidade da variância

```
>   # Homogeneidade da variância
> preditos.1 <- fitted(ml.1)
> PadPred.1 <- scale(preditos.1)
> PadRes.1 <- scale(ml.1$res)
> plot(PadPred.1, PadRes.1, pch = 16, xlab = "Valores preditos padronizados",
+ ylab = "Resíduos padronizados")
```




√ Clusters no gráfico

134

Estatística Computacional II - 2020

- Intervalos de confiança bootstrap

√ Método dos vetores

```
> # bootstrap dos vetores
> B <- 10000
>   # geração das amostras bootstrap
> Tboot <- matrix(0, nrow = B,ncol = 2)
> for(i in 1:B){
+ s <- 1:272
+ u <- sample(s, 272, replace = T)
+ f <- faithful[u, ]
+ x <- f[, 1]
+ y <- f[, 2]
+ ajuste <- lm(y ~ x)
+ Tboot[i, ] <- summary(ajuste)$coefficients[, 1]
+ }
```

135

Estatística Computacional II - 2020

√ Estimação bootstrap dos parâmetros

```
>   # estimativas bootstrap pontuais
> beta0 <- mean(Tboot[, 1])
> beta1 <- mean(Tboot[,2])
> beta0
[1] 33.47715
> beta1
[1] 10.73111
>
>   # estimativas bootstrap intervalares
> beta0.IC <- quantile(Tboot[,1], prob = c(0.025, 0.975))
> beta1.IC <- quantile(Tboot[,2], prob = c(0.025, 0.975))
> beta0.IC
    2.5%    97.5%
31.31975 35.65300
> beta1.IC
    2.5%    97.5%
10.14290 11.31889
```

$\hat{\beta}_0 = 33,474$

$\hat{\beta}_1 = 10,730$

$\text{IC}_{\beta_0} = (31,20; 35,75)$

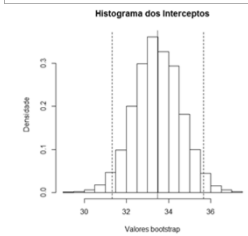
$\text{IC}_{\beta_1} = (10,11; 11,35)$

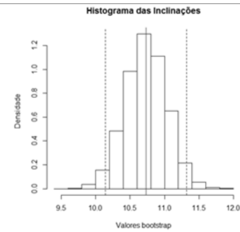
136

Estatística Computacional II - 2020

√ Distribuição dos parâmetros bootstrap:

```
>   # histograma dos interceptos
> hist(Tboot[, 1], main = "Histograma dos Interceptos", freq = F,
+ xlab = "Valores bootstrap", ylab = "Densidade")
> abline(v = beta0, col = "red")
> abline(v = quantile(Tboot[, 1], prob = c(0.025)), col = "blue", lty = 2)
> abline(v = quantile(Tboot[, 1], prob = c(0.975)), col = "blue", lty = 2)
>   # histograma das inclinações
> hist(Tboot[, 2], main = "Histograma das Inclinações", freq = F,
+ xlab = "Valores bootstrap", ylab = "Densidade")
> abline(v = beta1, col = "red")
> abline(v = quantile(Tboot[, 2], prob = c(0.025)), col = "blue", lty = 2)
> abline(v = quantile(Tboot[, 2], prob = c(0.975)), col = "blue", lty = 2)
```




137

Estatística Computacional II - 2020

- Intervalos de confiança bootstrap
  - √ Método dos resíduos

```
> # bootstrap dos resíduos
>
> x <- faithful$eruptions
> f <- fitted(ml.1)
> e <- residuals(ml.1)
>  # quantidade de amostras bootstrap
> B <- 10000
>  # Geração das amostras bootstrap
> Tboot <- matrix(0, nrow = B, ncol = 2)
> for(i in 1:B){
+ e.star <- sample(e, 272, replace = T)
+ y.star <- f + e.star
+ Tboot[i, ] <- summary(lm(y.star ~ x))$coefficients[, 1]
+ }
```

138

Estatística Computacional II - 2020

√ Estimação bootstrap dos parâmetros

```
>   # estimativas bootstrap pontuais
>   # Estimação bootstrap pontual
> beta0 <- mean(Tboot[, 1])
> beta1 <- mean(Tboot[, 2])
> beta0
[1] 33.49229
> beta1
[1] 10.72536
>
>   # estimativas bootstrap intervalares
> beta0.IC <- quantile(Tboot[,1], prob = c(0.025, 0.975))
> beta1.IC <- quantile(Tboot[,2], prob = c(0.025, 0.975))
> beta0.IC
    2.5%    97.5%
31.24205 35.80458
> beta1.IC
    2.5%    97.5%
10.09827 11.32948
```

$\hat{\beta}_0 = 33,474$ $\quad$ $\hat{\beta}_0^* = 33,472$

$\hat{\beta}_1 = 10,730$ $\quad$ $\hat{\beta}_1^* = 10,731$

$\mathrm{IC}_{\beta_0} = (31,20;35,75)$ $\quad$ $\mathrm{IC}^*_{\beta_0} = (31,32;35,65)$

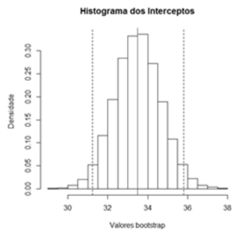
$\mathrm{IC}_{\beta_1} = (10,11;11,35)$ $\quad$ $\mathrm{IC}^*_{\beta_1} = (10,13;11,32)$

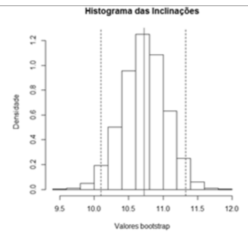
139

Estatística Computacional II - 2020

√ Distribuição dos parâmetros bootstrap:

```
>   # histograma dos interceptos
> hist(Tboot[, 1], main = "Histograma dos Interceptos", freq = F,
+ xlab = "Valores bootstrap", ylab = "Densidade")
> abline(v = beta0, col = "red")
> abline(v = quantile(Tboot[, 1], prob = c(0.025)), col = "blue", lty = 2)
> abline(v = quantile(Tboot[, 1], prob = c(0.975)), col = "blue", lty = 2)
>   # histograma das inclinações
> hist(Tboot[, 2], main = "Histograma das Inclinações", freq = F,
+ xlab = "Valores bootstrap", ylab = "Densidade")
> abline(v = beta1, col = "red")
> abline(v = quantile(Tboot[, 2], prob = c(0.025)), col = "blue", lty = 2)
> abline(v = quantile(Tboot[, 2], prob = c(0.975)), col = "blue", lty = 2)
```




140

Estatística Computacional II - 2020

**Seleção de Variáveis**

- Há muitas maneiras de conduzir seleção de variáveis em problemas de regressão
  - √ Critérios de escolha:
    - $R^2$
    - AIC – *Akaike Information Criterion*
    - BIC – *Bayesian Information Criterion*
    - Teste F *stepwise*
    - Complexidade estocástica
  - √ Algumas dessas abordagens podem ser aplicadas em regressão não linear

141

Estatística Computacional II - 2020

## Exemplo

- Considere o modelo:

$$y = 0,5 + 0,5x + 0,2x^2 + z + 0,1xz + \epsilon$$

√ com:

$$x = t$$
$$z = \sin(t)$$
$$t \in [-5, 5]$$
$$\epsilon \sim \mathrm{N}(\mu = 0, \sigma = 0.25)$$

√ Objetivo:

- Determinar modelo subjacente em contexto mais geral de especificação

$$y \sim x + z + x^2 + z^2 + x^3 + z^3 + xz + x^2z + xz^2$$

√ Usada regressão linear stepwise com AIC

- B = 1.000 reamostras de 51 dados



## √ Construção de conjunto de dados

```
> #   Seleção de Variáveis
> set.seed(666)
> t <- seq(-5, 5, 0.2)
>   # qte de dados
> (n <- length(t) )
[1] 51
>   # construção do conjunto de dados
> x <- t
> z <- sin(t)
> y.verd <- 0.5 + 0.5*x + 0.2*x^2 + z + 0.1*x*z
> epsilon <- rnorm(n, 0, 0.5)
> dados <- data.frame(x = x, x2 = x^2, z = z, z2 = z^2, xz = x*z, x3 = x^3,
+ z3 = z^3, x2z = x^2*z, xz2 = x*z^2, y = y.verd + epsilon)
> head(dados)
     x    x2         z        z2        xz       x3        z3      x2z
1 -5.0 25.00 0.9589243 0.9195358 -4.794621 -125.000 0.8817652 23.97311
2 -4.8 23.04 0.9961646 0.9923439 -4.781590 -110.592 0.9885379 22.95163
3 -4.6 21.16 0.9936910 0.9874218 -4.570979  -97.336 0.9811922 21.02650
4 -4.4 19.36 0.9516021 0.9055465 -4.187049  -85.184 0.8617199 18.42302
        xz2        y
1 -4.597679 3.856118
2 -4.763251 4.233183
3 -4.542140 2.791026
4 -3.984405 3.718981
```



## √ Ajuste do modelo

```
> library("MASS")
>   # ajuste do modelo
> library("MASS")
>   # inciando a modelagem
> mod.1 <- lm(y ~ x + z, data = dados)
> amod.l <- stepAIC(mod.1, scope = list(upper = ~ x + z + x2 + z2 + xz + x3 +
+ z3 + x2z + xz2, lower = ~ 1), trace = FALSE)
> summary(amod.l)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.15488    0.17896   0.865 0.391389
x            0.57449    0.04582  12.539 2.76e-16 ***
z            0.72747    0.19557   3.720 0.000551 ***
x2           0.24315    0.02317  10.493 1.13e-13 ***
xz           0.20480    0.07966   2.571 0.013520 *
x2z          0.03416    0.01995   1.712 0.093742 .
---
Residual standard error: 0.5485 on 45 degrees of freedom
Multiple R-squared:  0.9435,    Adjusted R-squared:  0.9373
F-statistic: 150.4 on 5 and 45 DF,  p-value: < 2.2e-16
```

√ Ajuste do modelo simulado a partir da especificação mais geral



## √ Bootstrap para encontrar modelos comuns:

```
>   # bootstrap para encontrar modelos comuns
> nReal = 1000
>   # proporção de ocorrência de cada variável
> p <- rep(0, 9)
> iReal <-1
> for (iReal in 1:nReal){
+   # bootstrap por indices
+ ind <- sample(1:n, n, replace = TRUE)
+   # seleção das linhas da reamostra
+ bdados <- dados[ind,]
+   # inicialização do modelo
+ mod.2 <- lm(y ~ x + z, data = bdados)
+ amod.2 <- stepAIC(mod.1, scope = list(upper = ~ x + z + x2 + z2 + xz +
+ x3 + z3 + x2z + xz2, lower = ~ 1), trace = FALSE)
+   # variáveis ajustadas
+ s <- names(coef(amod.2))
+ m <- length(s)

... # o código continua no próximo slide
```

√ Código continua no próximo slide!

√ Código para bootstraping (continuação):

```
>   # bootstrap para encontrar modelos comuns
+   # verifica os termos das variáveis
+ for (j in 2:m){
+ if (s[j] == "x"){
+ p[1] <- p[1] + 1
+ }else if (s[j] == "x2"){
+ p[2] <- p[2] + 1
+ }else if (s[j] == "z"){
+ p[3] <- p[3] + 1
+ }else if (s[j] == "z2"){
+ p[4] <- p[4] + 1
+ }else if (s[j] == "xz"){
+ p[5] <- p[5] + 1
+ }else if (s[j] == "x3"){
+ p[6] <- p[6] + 1
+ }else if (s[j] == "z3"){
+ p[7] <- p[7] + 1
+ }else if (s[j] == "x2z"){
+ p[8] <- p[8] + 1
+ }else if (s[j] == "xz2"){
+ p[9] <- p[9] + 1
+ }else{
+ cat("Erro!", sprintf("%5d", m), sprintf("%5d", j), "\n")
+ cat(s)
+ }
+ }
+ } # final do código
```

146



√ Análise do resultado do bootstrap:

```
>   # Análise dos resultados
> print("Variáveis: x, x^2, z, z^2, x*z, x^3, z^3, x^2*z, x*z^2")
[1] "Variáveis: x, x^2, z, z^2, x*z, x^3, z^3, x^2*z, x*z^2"
> print("Verdadeiro:: 1 1 1 0 1 0 0 0 0")
[1] "Verdadeiro:: 1 1 1 0 1 0 0 0 0”
>   # proporções associadas com os termos das variáveis
> (prop <- p/nReal)
[[1] 1 1 1 0 1 0 0 1 0
>   # tabela de resultados
> tabela <- rbind(c(1, 1, 1, 0, 1, 0, 0, 0, 0), p/nReal)
> rownames(tabela) <- c("Verdadeiro", "Bootstrap")
> colnames(tabela) <- c("x", "x^2", "z", "z^2", "x*z", "x^3", "z^3", "x^2*z",
+ "x*z^2")
> tabela
           x x^2 z z^2 x*z x^3 z^3 x^2*z x*z^2
Verdadeiro 1   1 1   0   1   0   0     0     0
Bootstrap  1   1 1   0   1   0   0     1     0
```

√ Encontrados todos os termos do modelo

– Junto com o termo $x^2z$ espúrio (p > 0.05)

√ Refazer com `set.seed(1)`.

147



**Referências**

## Bibliografia Recomendada


- CHERNICK, M.; LABUDDE, R. *An introduction to bootstrap methods with applications to R*. Wiley, 2014.
- DAVISON, A.; HINKLEY, D. *Bootstrap methods and their application*. Cambridge University Press, 1997.
- EFRON, B.; TIBSHIRANI, R. J. *An introduction to the bootstrap*. Chapman & Hall, 1994


150