

Estatística Computacional I

Lupércio França Bessegato
Dep. de Estatística/UFJF



Roteiro Geral



1. Programando em R
2. Preparação, limpeza e manipulação de dados
3. Gráficos em R
4. Tópicos especiais
5. Referências

Estatística Computacional I - 2020

2

Programando em R

Simulação

Comando replicate

- Atua com a função `sapply`, para avaliação repetida de expressão
 - ✓ Em geral, com geração de números aleatórios
 - ✓ Saída é um vetor ou matriz
- Syntaxe:

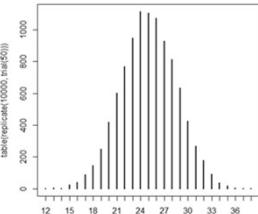

```
replicate(n, expr, ...)
```

 - ✓ `n`: quantidade de replicações
 - ✓ `expr`: expressão que será avaliada repetidamente
 - ✓ `...` : argumentos opcionais da função

275

✓ Lançamento de uma moeda:

```
> # Monte Carlo
> # função para contar a qte de caras (ou coroas, tanto faz)
> lancamentos <- function(n) sum(runif(n) < 0.5)
> # lançamento de uma moeda 10 vezes
> trial(10)
[1] 6
> # 5 repetições do lançamento de uma moeda 10 vezes
> replicate(5, trial(10))
[1] 6 6 4 3 5
> # 1000 repetições do lançamento de uma moeda 50 vezes
> plot(table(replicate(10000, trial(50))))
```



✓ Se a moeda for viciada?

- A probabilidade de cara ser, por exemplo, 0.3.

277

✓ Exemplo – geração de números aleatórios: ✓ Amostras de população normal:

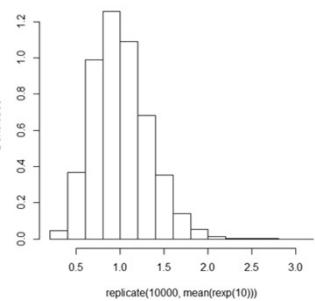
```
> # Geração de 10 amostras normais padrão de tamanho 5
> (amostras <- replicate(10, rnorm(5)))
[1,] -1.67930536 -1.0707589 0.0981208 0.03765597 0.44122502 0.9692450
[2,] -0.04670559 0.7698282 -1.5940969 -0.97425884 -0.06288204 -1.9789664
[3,] 0.73822989 0.7350203 -0.3226240 2.23524896 -0.10750389 -1.0891118
[4,] 1.43657754 0.3473096 -0.7004251 -1.61056486 0.54328757 1.2078708
[5,] -0.75926441 -1.0941653 0.2063968 -1.52482166 1.09996974 0.8460089
[6,] 0.51027178 2.0998434 0.7435842 0.7179811
[7,] 0.90178654 1.0320483 2.4015640 0.6981859
[8,] -1.06158664 -1.2170829 -1.3185953 0.5272686
[9,] -0.54925963 0.3804547 0.4507574 1.5388225
[10,] 0.04192024 -1.4635302 0.6330648 1.0232150
> is.matrix(amostras)
[1] TRUE
> dim(amostras)
[1] 5 10
> # medias das amostras
> apply(amostras, 2, mean)
[1] -0.062093585 -0.062553226 -0.462525674 -0.367348086 0.382819279
[6] -0.008990712 -0.031373542 0.166346674 0.582075045 0.901094621
```

276

✓ Distribuição da média amostral (n = 10):

```
> # histograma das médias de 100 amostras exponenciais padrão de tamanho 10
> hist(replicate(10000, mean(rexp(10))), freq = F, ylab = "Densidade")
```

Histogram of `replicate(10000, mean(rexp(10)))`



✓ Qual o formato do gráfico?
 ✓ O que você espera que aconteça se amostra aumentar?
 - Digamos, de 10 para 100

278

✓ Distribuição da média amostral (n = 100):

```
> # histograma das médias de 10000 amostras exponenciais padrão de tamanho 100
> hist(replicate(10000, mean(rexp(100))), freq = F, ylab = "Densidade")
```

Histogram of replicate(10000, mean(rexp(100)))

✓ Compare os gráficos?
 – O formato mudou?
 – O que mais aconteceu?

279

Estatística Computacional I - 2020

✓ Comparação com a distribuição normal:

```
> # comparando com a distribuição normal
> normal <- function(x) dnorm(x, mean= 1, sd = 1/sqrt(100))
> curve(normal, from = 0, to = 2, lty = 2, lwd = 2, col = "blue", add = T)
```

Histogram of replicate(10000, mean(rexp(100)))

✓ O histograma se aproxima da curva?
 – Aumente a quantidade de replicações

✓ O que acontece quando o tamanho amostral aumenta?
 – Use n = 500, por exemplo?

280

Estatística Computacional I - 2020

✓ Geração de 10.000 amostras de tamanho 10:

```
> # tamanho amostral: 10
> n <- 10
> medias.n <- replicate(run, mean(rfx(n)))
> hist(medias.n, freq = F, xlim = c(0, 1), ylim = c(0, maximo),
+ ylab = "Densidade", main = "Histograma de Médias Amostrais")
> text(0.75, 5, bquote(bar(x)[bar(x)] == .(mean(medias.n))), cex = 0.8)
> text(0.75, 4.75, bquote(s[bar(x)] == .(sd(medias.n))), cex = 0.8)
> mtext(paste("n = ", n), side = 3, line = 0.5)
```

Histograma de Médias Amostrais
n = 10

✓ Houve modificação na distribuição das médias amostrais?
 – Quais?

281

Estatística Computacional I - 2020

✓ Geração de 10.000 amostras de tamanho 20:

```
> # tamanho amostral: 20
> n <- 20
> medias.n <- replicate(run, mean(rfx(n)))
> hist(medias.n, freq = F, xlim = c(0, 1), ylim = c(0, maximo),
+ ylab = "Densidade", main = "Histograma de Médias Amostrais")
> text(0.75, 5, bquote(bar(x)[bar(x)] == .(mean(medias.n))), cex = 0.8)
> text(0.75, 4.75, bquote(s[bar(x)] == .(sd(medias.n))), cex = 0.8)
> mtext(paste("n = ", n), side = 3, line = 0.5)
> curve(dnorm(x, mean = 0.5, sd = 0.387/sqrt(n)), from = 0, to = 1,
+ col = "blue", add = TRUE)
```

Histograma de Médias Amostrais
n = 20

✓ O que aconteceu com a distribuição das médias amostrais?

282

Estatística Computacional I - 2020

✓ Gráficos no mesmo frame:

```

> # parametros dos gráficos
> abs.x <- 0.8
> ord.y <- 5
> maximo <- 5.5
> for(n in c(2, 4, 10, 20)){
+ hist(replicate(run, mean(rfx(n))), freq = F, xlim = c(0, 1),
+ ylim = c(0, maximo), ylab = "Densidade", xlab = expression(X[user]),
+ main = paste("n = ", n), cex.main = 1)
+ mtext("Histograma de médias amostrais", side = 3, line = -18, font = 2,
+ outer = TRUE) }
```

✓ Quais cuidados devem ser tomados para construção dos gráficos?

283

Estatística Computacional I - 2020

Referências

✓ Bibliografia Recomendada

- ALBERT, J.; RIZZO, M. *R by Example*. Springer, 2012.
- CHRISTIAN, N. *Basic Programming*, Lecture Notes
- DALGAARD, P. *Introductory statistics with R*. Springer, 2008.
- KLEIBER, C.; ZEILEIS, A. *Applied econometrics with R*. Springer, 2008.
- GARDENER, M. *Beginning R: The statistical programming language*. John Wiley & Sons, 2012.

287

Estatística Computacional I - 2020