

Métodos Quantitativos IV - Lista 3

Professor Manoel Galdino e Monitor Davi Veronese

September 14, 2023

Nesta lista, trabalharemos com simulação estatística. Explique detalhadamente suas simulações, comentando o código etapa por etapa.

1

Rode `?rnorm` no R e leia o help. Se preciso, consulte outras fontes. Explique o que a função faz. Verifique que essa função gera uma simulação de uma distribuição normal, em que você pode especificar a média e o desvio-padrão. Verifique que entendeu rodando função e gerando valores simulados da função `rnorm`. Pode usar média 0 e desvio-padrão 1, que são o default da função.

```
# ?rnorm
```

2

Rode `x <- rnorm(100, mean = 2, sd = 1)`. Por que a média da distribuição é diferente da média que você obtém rodando `mean(x)`?

```
# x <- rnorm(100, mean=2, sd=1)
# mean(x)
```

Os parâmetros estabelecidos definem a distribuição a partir da qual serão retirados valores aleatórios. A aleatoriedade faz com que a média desses valores não seja exatamente igual à média da distribuição.

3

(OPCIONAL PARA A GRADUAÇÃO)

Se você rodar de novo $x <- rnorm(100, mean = 2, sd = 1)$ e calcular a média de x , obterá um valor um pouco diferente da primeira vez. Por quê?

A aleatoriedade dos valores retirados faz com que a média desses valores não seja exatamente igual à média da distribuição. Como os valores retirados são aleatórios, há variabilidade entre simulações.

4

Uma forma de você armazenar as duas médias que você computou em um vetor é do seguinte modo:

```
vetor_medias <- numeric()
vetor_medias[1] <- mean(rnorm(100, mean=2, sd=1))
vetor_medias[2] <- mean(rnorm(100, mean=2, sd=1))
```

Imprima o conteúdo de "vetor_medias" e verifique que de fato armazenou duas médias.

5

(OPCIONAL PARA A GRADUAÇÃO)

Repita esse procedimento 30 vezes no total, armazenando as 30 médias no vetor `vetor_medias`. Se possível, use um loop (laço) para fazer isso.

```
vetor_medias <- numeric()

for (i in 1:30) {

  vetor_medias[i] <- mean(rnorm(100, mean=2, sd=1))

}

print(vetor_medias)

## [1] 1.962704 2.011136 2.000846 2.102798 1.888707 1.963051 2.034009 2.008199
## [9] 2.103619 1.933343 2.130895 1.853933 1.972068 1.983080 2.030385 2.068222
## [17] 1.988013 1.979212 1.990660 1.956685 2.012626 1.946750 2.066790 2.151209
## [25] 2.169443 1.920292 1.998892 1.928670 2.058198 1.904865
```

6

Plote o histograma (use a função `geom_histogram` no `ggplot`) das médias. Para isso, crie um banco de dados (`ggplot` só aceita plotar variáveis de banco de dados) do seguinte modo:

```
#df <- data.frame(medias = vetor_medias, sim_id = 1:30)
#hist(vetor_medias)
#mean(vetor_medias)
#var(vetor_medias)
```

Você reconhece a distribuição apresentada pelo histograma? Se sim, qual é ela? Consegue advinhar a média e desvio padrão da distribuição ou calculá-la?

Distribuição normal com média μ e variância $\frac{\sigma^2}{n}$.

7

(OPCIONAL PARA A GRADUAÇÃO)

Realize uma simulação estatística para verificar a distribuição de probabilidade dos resultados do lançamento de uma moeda.

```
library(dplyr)
library(tidylog)
library(ggplot2)
library(tidyverse)

# Defina o número de lançamentos
n <- 1000000

# Crie um vetor para armazenar os resultados das jogadas
X <- numeric()

# Suponha que "Cara" == "sair 1" & "Coroa" == "sair 2"

# Lance a moeda n vezes e armazene os resultados
set.seed(13492)
for (i in 1:n) {
  X[i] <- sample(1:2, size=1, replace = TRUE)
}
```

```

X <- ifelse(X == 1, "Cara", "Coroa")

# Calcule as probabilidades
## Uma possibilidade
table(X)

## X
##   Cara   Coroa
## 500690 499310

## Outra possibilidade:
sum(X=="Cara")/n # 50,069%

## [1] 0.50069

sum(X=="Coroa")/n # 49,931%

## [1] 0.49931

# Outra possibilidade
sum(rbinom(1000, 1, 0.5))

## [1] 527

```

8

(OPCIONAL PARA A GRADUAÇÃO)

Retire 10, 100, 1000 e 10000 valores de uma distribuição normal padrão e de uma distribuição binomial ($n = 20$, $p = 0.7$). Apresente os histogramas.

$$Z \sim \mathcal{N}(0, 1)$$

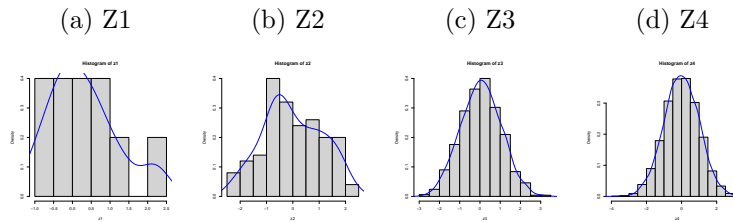
$$B \sim \text{Bin}(20, 0.7)$$

```

# Defina o número de repetições
n_1 <- 10
n_2 <- 100

```

Figure 1: Histogramas Z



```
n_3 <- 1000
n_4 <- 10000

# Retiradas da normal
z1 <- rnorm(n_1, 0, 1)
z2 <- rnorm(n_2, 0, 1)
z3 <- rnorm(n_3, 0, 1)
z4 <- rnorm(n_4, 0, 1)

# Retiradas da binomial
b1 <- rbinom(n_1, 20, 0.7)
b2 <- rbinom(n_2, 20, 0.7)
b3 <- rbinom(n_3, 20, 0.7)
b4 <- rbinom(n_4, 20, 0.7)

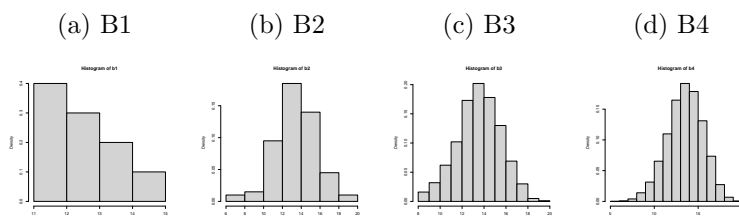
# Apresente os histogramas
hist(z1, probability = TRUE)
lines(density(z1), col = 'blue')

hist(z2, probability = TRUE)
lines(density(z2), col = 'blue')

hist(z3, probability = TRUE)
lines(density(z3), col = 'blue')

hist(z4, probability = TRUE)
lines(density(z4), col = 'blue')
```

Figure 2: Histogramas B



```
hist(b1, probability = TRUE)
```

```
hist(b2, probability = TRUE)
```

```
hist(b3, probability = TRUE)
```

```
hist(b4, probability = TRUE)
```

```
# Limpe seu environment
```

```
rm(list=ls())
```

9

Apresente seus resultados em um arquivo PDF. Garanta que seu arquivo esteja limpo, contendo as respostas, os gráficos e as tabelas, mas não eventuais mensagens e erros. O arquivo PDF pode ser gerado diretamente a partir do R por meio do RMarkdown ou do RSweave. **Para os alunos de graduação, isso é recomendado, mas não obrigatório.** Adicionalmente, forneça o script para replicação.