Estatística Computacional

Universidade Federal da Bahia

Gilberto Pereira Sassi

Tópico 7

Pacotes usados nessa semana

```
set.seed(12345)
library(readxl)
library(readODS)
library(nortest)
library(qqplotr)
library(ggthemes)
library(glue)
# instale com remotes::install_github("gilberto-sassi/statBasics")
library(statBasics)
library(tidyverse)
```

Métodos de Monte Carlo

Definição

- Simulações de Monte Carlo são usadas para realizar inferência quando métodos analíticos são complexos:
  - Não sabemos a distribuição amostral de um estimador
  - Não temos certeza que as suposições de um modelo são satisfeitas
- Pré-requisitos para realizar uma simulação de Monte de Carlo:
  - precisamos de um modelo estatístico
  - precisamos de um método para gerar números psedo-aleatórios seguindo as especificações deste modelo estatístico
- Usos mais comuns:
  - inferência quando não conhecemos a distribuição amostral de um estimador
  - estimar o desempenho do modelo quando as suposições são violadas
  - comparar modelos e avaliar a performance de métodos inferenciais

Ideia: geramos n amostras. Para cada amostra, calcular a estimativa de um estimador, e então temos n estimativas e conseguimos estudar a distribuição do estimador.

Revisão de Teste de Hipóteses

Queremos decidir, usando a evidência da amostra entre duas hipóteses:

$$H_0 : \text{hipótese nula},$$
$$H_1 : \text{hipótese alternativa}.$$

Erro possíveis:

Decisão / Situação na População	H_0	H_1
H_0	ok!	Erro tipo II
H_1	Erro tipo I	ok!

- $\alpha = P(H_1 \mid H_0)$: nível de significância
- $\beta = P(H_0 \mid H_1)$
- $1 - \beta$: poder do teste

Procedimento de Neymann-Pearson

1. Estabeleça H_0 e H_1
2. Escolha um nível de significância
3. Estabeleça a *região crítica* e a estatística do teste
4. Encontre o valor crítico
5. Rejeite H_0 se a estatística do teste pertence à *região crítica*

Exemplo procedimento de Neymann-Pearson

Suponha que $X \sim N(\mu, 28)$, e desejamos decidir entre $H_0 : \mu \neq 10$ e $H_1 : \mu = 10$.

```
amostra <- cars$speed
ht_1pop_mean(amostra, mu = 10, sd_pop = 28, alternative = "two.sided")
```

```
## # A tibble: 1 × 7
##   statistic p_value critical_value critical_region      alternative
##       <dbl>   <dbl>          <dbl> <chr>                <chr>
## 1      1.36   0.173           1.96 (-Inf,-1.960)U(1.96… two.sided
## # … with 2 more variables: mu <dbl>, sig_level <dbl>
```

valor-p

1. Estabeleça H_0 e H_1
2. Escolha um nível de significância
3. Calcule o valor-p
4. Rejeite H_0 se o valor-p for menor que o nível de significância

Note que Se H_0 é verdade na população, tomaremos a decisão errada em $100 \cdot \alpha$ das amostras.

Ilustração do valor-p: Suponha que $X \sim N(\mu, 1)$, e queremos testar $H_0 : \mu = 0$ e $H_1 : \mu \neq 0$. Imagine H_0 é verdade na população.

```
valores_p <- seq_len(1000) |>
  map_dbl(\(i) {
    amostra <- rnorm(1000, mean = 0, sd = 1)
    ht_1pop_mean(amostra, mu = 0, sd_pop = 1,
                 alternative = "two.sided")$p_value < 0.05
  })
mean(valores_p)
```

```
## [1] 0.044
```

Intervalo de Confiança

- Suponha que θ seja um parâmetro da população
- $IC(\theta, \gamma) = \left(\hat{\theta}_L; \hat{\theta}_U\right)$, em que $P(\hat{\theta}_L < \theta < \hat{\theta}_U) = 1 - \alpha$;
- Em $100 \cdot \gamma$ das amostras produzem intervalos de confiança *corretos*, como ilustrados na figura abaixo

Ilustração do intervalo de confiança: Suponha que $X \sim N(\mu, 1)$, e queremos construir o intervalo de confiança para μ. Imagine que na população temos $\mu = 0$.

```
intervalo_correto <- seq_len(1000) |>
  map_dbl(\(i) {
    amostra <- rnorm(1000, mean = 0, sd = 1)
    ci <- ci_norm(amostra, sd_pop = 1, conf_level = 0.95)
    ci$lower_ci <= 0 & 0 <= ci$upper_ci
  })
mean(intervalo_correto)
```

```
## [1] 0.946
```

Psedo-algoritmo para o procedimento – Monte Carlo

1. Escolha ou determine um modelo adequado para o seu problema
2. Retire uma amostra ~~pseudo-~~aleatória segundo este modelo
3. Compute a estatística (ou estimativa) deste modelo
4. Repita os passos **2.** e **3.** M vezes
5. Use estes M valores para estimar a Função de Distribuição Acumulada (ou alguma característica deseja) do estimador deste modelo

Procedimento de Neymann-Pearson – Monte Carlo

Ideia: Encontrar o valor crítico usando Monte Carlo.

1. Estabeleça H_0 e H_1
2. Escolha o nível de significância
3. Determine a região crítica e a estatística do teste
4. Gera uma amostra ~~pseudo-~~aleatória sob H_0
5. Compute a estatística do estatística para esta amostra
6. Repita os passos **4.** e **5.** M vezes
7. Encontre o valor crítico, que é um quantil, usando os M valores das estatísticas
8. Calcule a estatística do teste para a amostra observada. Se ela estiver na região crítica, rejeitamos H_0

Procedimento de Neymann-Pearson – Monte Carlo

Considere o conjunto de dados `ToothGrowth`, e considere a variável `len` tem distribuição normal com desvio padrão $\sigma = 7,65$. Suponha que queremos decidir entre as hipóteses $H_0 : \mu = 15$ e $H_1 : \mu > 15$ ao nível de significância $\alpha = 0,05$. Neste caso a região crítica é $RC = \{z_0 \mid z_0 > z_{1-\alpha}\}$, em que $z_0 = \frac{(\bar{x}-15)\sqrt{n}}{7,65}$.

```
# Para calcular o valo crítico assumimos que $H_0$
dados <- read_ods("data/raw/ToothGrowth.ods")
tamanho_amostra <- nrow(dados); mu_0 <- 15; sd_pop <- 7.65
estatisticas_sob_h0 <- seq_len(1000) |>
  map_dbl(\(i) {
    amostra <- rnorm(tamanho_amostra, mean = mu_0, sd = sd_pop)
    ht_1pop_mean(amostra, mu = mu_0, sd_pop = sd_pop, alternative = "greater")$statistic
  })
valor_critico <- quantile(estatisticas_sob_h0, probs = 1 - 0.05)
z_0 <- ht_1pop_mean(dados$len, mu = mu_0, sd_pop = sd_pop, alternative = "greater")$statistic
ifelse(valor_critico < z_0, "Rejeito H0", "Não rejeito H0")
```

```
##          95%
## "Rejeito H0"
```

Procedimento de valor-p – Monte Carlo

1. Estabeleça H_0 e H_1
2. Escolha o nível de significância
3. Calcule a estatística do teste para a amostra observada
4. Gere uma amostra ~~pseudo-~~aleatória sobre H_0
5. Calcule a estatística do teste para esta ~~pseudo-~~aleatória
6. Repita os passos **4.** e **5.** M vezes
7. O valor-p é a proporção de estatísticas do teste no passo **6.** que são mais extremas que a estatística de teste calculada no passo **3.**

Procedimento de valor-p – Monte Carlo

Considere o conjunto de dados `ToothGrowth`, e considere a variável `len` tem distribuição normal com desvio padrão $\sigma = 7,65$. Suponha que queremos decidir entre as hipóteses $H_0 : \mu = 15$ e $H_1 : \mu > 15$ ao nível de significância $\alpha = 0,05$. Neste caso a estatística do teste é $z_0 = \frac{(\bar{x}-15)\sqrt{n}}{7,65}$.

```
dados <- read_ods("data/raw/ToothGrowth.ods")
tamanho_amostra <- nrow(dados); mu_0 <- 15; sd_pop <- 7.65
z_0 <- ht_1pop_mean(dados$len, mu = mu_0, sd_pop = sd_pop, alternative = "greater")$statistic
estatistica_extrema <- seq_len(1000) |>
  map_dbl(\(i) {
    amostra <- rnorm(tamanho_amostra, mean = mu_0, sd = sd_pop)
    ht_1pop_mean(amostra, mu = mu_0, sd_pop = sd_pop, alternative = "greater")$statistic > z_0
  })
glue("Valor-p: {mean(estatistica_extrema)}")
```

```
## Valor-p: 0
```

```
ifelse(mean(estatistica_extrema) < 0.05, "Rejeito H0", "Não rejeito H0")
```

```
## [1] "Rejeito H0"
```

Cálculo do poder de teste

Ideia: determinar o nível de significância para o teste de hipóteses.

1. Estabeleça H_0 e H_1
2. Determine a região crítica e o valor crítico
3. Determine a distribuição da estatística do teste quando a hipótese alternativa é verdadeira
4. Gere uma amostra ~~pseudo-~~aleatória usando a especificação do passo 3)
5. Defina $Erro = 1$ se não rejeitamos H_0, e $Erro = 0$ se rejeitamos H_0
6. Repita os passos **3.** e **4.** M vezes
7. O nível de significância é $\frac{\sum_{j=1}^{M} Erro_j}{M}$

Cálculo do poder de teste

Considere o conjunto de dados `ToothGrowth`, e considere a variável `len` tem distribuição normal com desvio padrão $\sigma = 7,65$. Suponha que queremos decidir entre as hipóteses $H_0 : \mu = 15$ e $H_1 : \mu > 15$ ao nível de significância $\alpha = 0,05$. Assuma a hipótese alternativa é verdadeira, mais especificamente $\mu = 20$.

```
dados <- read_ods("data/raw/ToothGrowth.ods")
tamanho_amostra <- nrow(dados); mu_1 <- 100; sd_pop <- 7.65; mu_0 <- 15
erro_tipo_ii <- seq_len(1000) |>
  map_dbl(\(i) {
    amostra <- rnorm(1000, mean = mu_1, sd = sd_pop)
    teste <- ht_1pop_mean(amostra, mu = mu_0, sd_pop = sd_pop, alternative = "greater")
    teste$statistic <= teste$critical_value
  })
glue("Poder do teste: {1 - mean(erro_tipo_ii)}")
```

```
## Poder do teste: 1
```

Aproximando integrais – Monte Carlo

Se $X_1, \ldots, X_m \overset{iid}{\sim} g$. Se $E|h(X_1)| < \infty$, então pela lei fraca dos grandes números temos que $\frac{\sum_{k=1}^{m} h(X_k)}{m} \to \int h(x)g(x)d(x)$.

Suponh que temos uma $X_1, \ldots, X_m \overset{iid}{\sim} g$, então

$$\int f(x)dx = \int \frac{f(x)}{g(x)} g(x)dx \approx \frac{1}{m} \sum_{k=1}^{m} \frac{f(x_k)}{g(x_k)}.$$

Exemplo

Suponha que queremos $\int_0^3 log(\sum_{k=1}^4 x^k) \cdot \sum_{k=1}^4 x^k dx$.

```
amostra <- runif(50000, min = 0, max = 3)
integracao  <- amostra |>
  map_dbl(\(x) {
    soma <- map_dbl(1:4, \(k) x^k) |> sum()
    soma * log(soma) / dunif(x,  min = 0, max = 3)
  }) |>
  mean()
integracao
```

```
## [1] 329.7158
```