

# **Estatística Educacional**

**Héliton Tavares**

**Cássia Barbosa Corrêa      Hugo Leonardo Faro**

2024-09-12

# **Apresentação**

Resolução das listas de exercícios

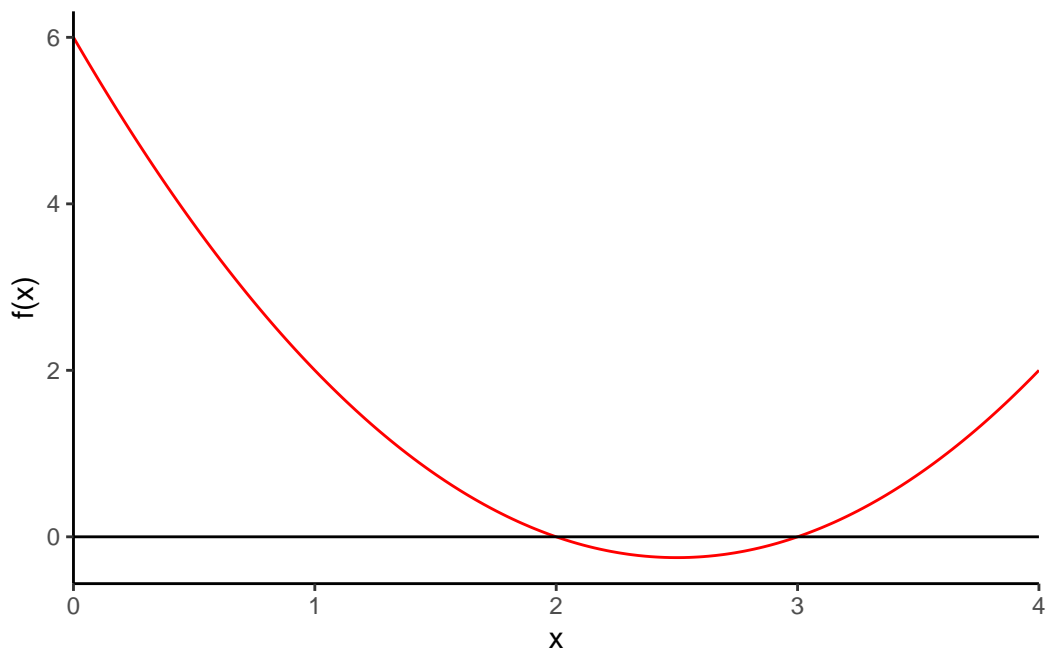
# Tarefa 1

## 1. Gráfico de Funções

**Figura 1.1** - Gráfico da função  $f(x) = x^2 - 5x + 6$ ,  $x \in [0, 4]$

```
library(ggplot2)
x <- seq(0, 4, by = 0.01)
y <- x^2 - 5*x + 6
dados <- data.frame(x = x, y = y)

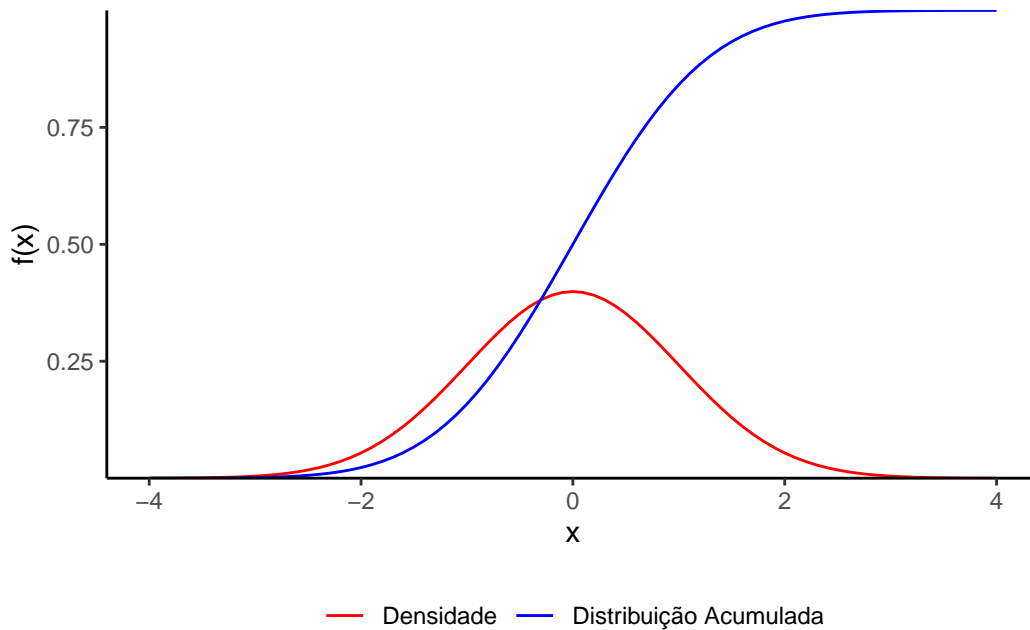
ggplot(dados, aes(x = x, y = y)) +
  geom_line(color = "red") +
  geom_hline(yintercept = 0) +
  labs(x = "x", y = "f(x)") +
  theme(panel.background = element_blank(),
        axis.line = element_line()) +
  scale_x_continuous(expand = c(0, 0))
```



**Figura 1.2** Gráfico da função  $f(x)$  = densidade da  $N(0,1)$  e Distribuição Acumulada da  $N(0,1)$

```
library(ggplot2)
x <- seq(-4, 4, by = 0.1)
densidade <- dnorm(x, mean = 0, sd = 1)
distribuicao <- pnorm(x, mean = 0, sd = 1)
dados <- data.frame(x = x, densidade = densidade, distribuicao = distribuicao)

ggplot(data = dados) +
  geom_line(aes(x = x, y = densidade, color = "Densidade")) +
  geom_line(aes(x = x, y = distribuicao, color = "Distribuição Acumulada")) +
  labs(x = "x", y = "f(x)") +
  scale_color_manual(values = c("Densidade" = "red",
                                "Distribuição Acumulada" = "blue")) +
  theme(panel.background = element_blank(),
        axis.line = element_line(),
        legend.title = element_blank(),
        legend.position = "bottom") +
  scale_y_continuous(expand = c(0, 0))
```



**Figura 1.3.** Gráfico da função  $f(x) = \frac{1}{1+\exp(-D \cdot a \cdot (x-b))}$ , com  $a = 1.5$ ,  $b = 1$ , para  $D = 1$  e  $D = 1.7$

```

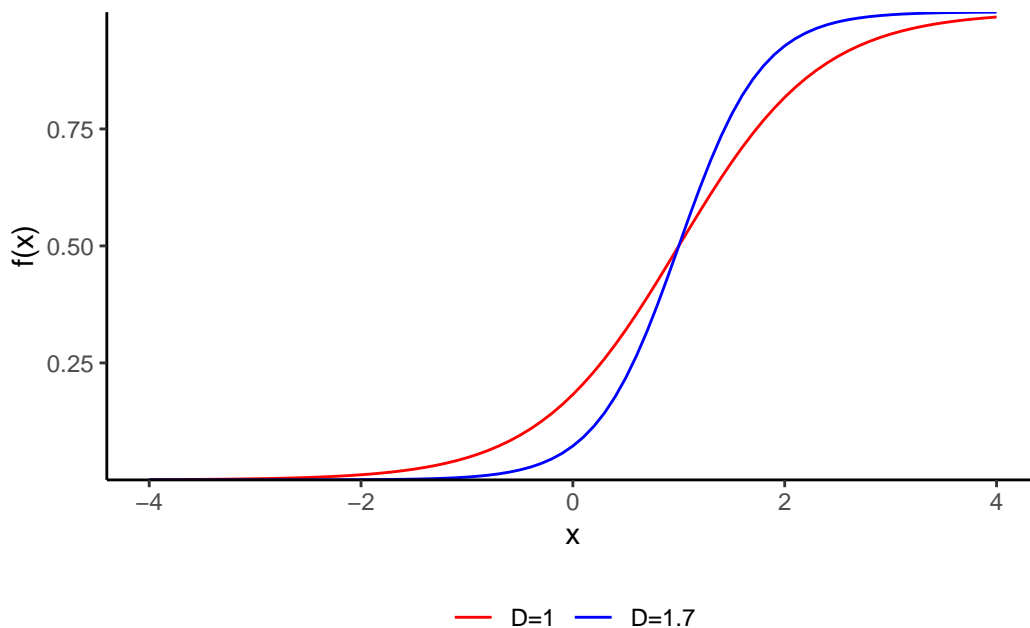
library(ggplot2)

a <- 1.5
b <- 1
x <- seq(-4, 4, by = 0.1)

f_D1 <- 1 / (1 + exp(-1 * a * (x - b)))
f_D17 <- 1 / (1 + exp(-1.7 * a * (x - b)))
dados <- data.frame(
  x = rep(x, 2),
  y = c(f_D1, f_D17),
  D = factor(rep(c("D=1", "D=1.7"), each = length(x))))

ggplot(dados, aes(x = x, y = y, color = D)) +
  geom_line() + labs(x = "x", y = "f(x)") +
  scale_color_manual(values = c("D=1" = "red", "D=1.7" = "blue")) +
  theme(panel.background = element_blank(),
        axis.line = element_line(),
        legend.title = element_blank(),
        legend.position = "bottom") +
  scale_y_continuous(expand = c(0, 0))

```



**Figura 1.4.** Comparando a função de distribuição  $N(0,1)$  com a função  $f(x) =$

$\frac{1}{1+\exp(-D \cdot a \cdot (x-b))}$ , com  $a = 1.5$ ,  $b = 1$ , para  $D = 1.7$ .

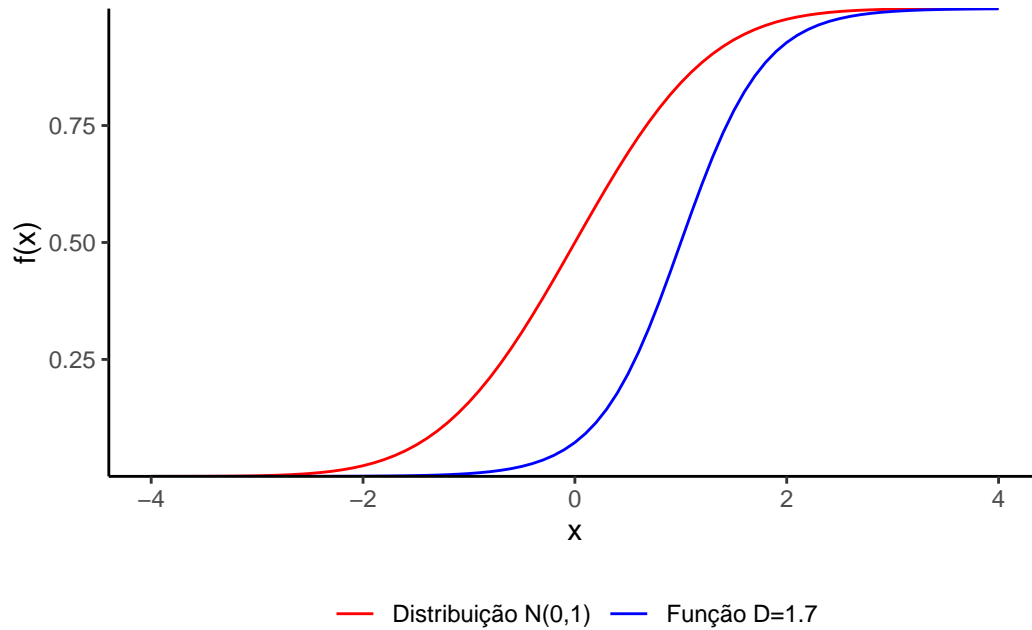
```
library(ggplot2)

a <- 1.5
b <- 1
x <- seq(-4, 4, by = 0.1)

distribuicao_normal <- pnorm(x, mean = 0, sd = 1)
f_D17 <- 1 / (1 + exp(-1.7 * a * (x - b)))

dados <- data.frame(
  x = rep(x, 2),
  y = c(distribuicao_normal, f_D17),
  Funcao = factor(rep(c("Distribuição N(0,1)", "Função D=1.7"),
    each = length(x))))

ggplot(dados, aes(x = x, y = y, color = Funcao)) +
  geom_line() + labs(x = "x", y = "f(x)") +
  scale_color_manual(values = c("Distribuição N(0,1)" = "red",
    "Função D=1.7" = "blue")) +
  theme(panel.background = element_blank(), axis.line = element_line(),
    legend.title = element_blank(), legend.position = "bottom") +
  scale_y_continuous(expand = c(0, 0))
```



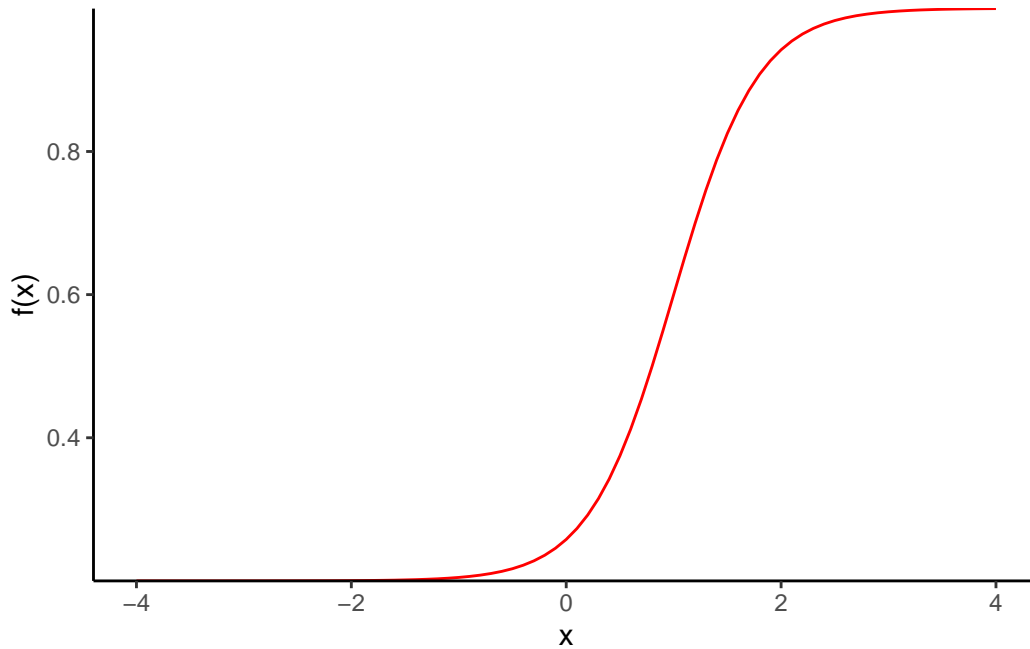
**Figura 1.5.** Gráfico da função  $f(x) = c + \frac{1-c}{1+\exp(-D \cdot a \cdot (x-b))}$ , com  $a = 1.5$ ,  $b = 1$ ,  $c = 0.2$ ,  $D = 1.7$ ,

```
library(ggplot2)

a <- 1.5
b <- 1
c <- 0.2
D <- 1.7
x <- seq(-4, 4, by = 0.1)

f_x <- c + (1 - c) / (1 + exp(-D * a * (x - b)))
dados <- data.frame(x = x, y = f_x)

ggplot(dados, aes(x = x, y = y)) +
  geom_line(color = "red") + labs(x = "x", y = "f(x)") +
  theme(panel.background = element_blank(), axis.line = element_line()) +
  scale_y_continuous(expand = c(0, 0))
```



**Figura 1.6.** Gráfico da função densidade da  $N(0,1)$  refletida ( $-f(x)$ ) junto com 3 funções logísticas com parâmetros  $\zeta = (a, b, c)$  dados por  $(1, .5, 0.2)$ ,  $(1, 1.5, 0.2)$  e  $(2, 1.5, 0.2)$

```
library(ggplot2)

x <- seq(-3, 3, by = 0.1)
densidade_refletida <- -dnorm(x, mean = 0, sd = 1)

logistica1 <- 0.2 + (1 - 0.2) / (1 + exp(-1 * (x - 0.5)))
logistica2 <- 0.2 + (1 - 0.2) / (1 + exp(-1 * (x - 1.5)))
logistica3 <- 0.2 + (1 - 0.2) / (1 + exp(-2 * (x - 1.5)))

dados <- data.frame(
  x = rep(x, 4),
  y = c(densidade_refletida, logistica1, logistica2, logistica3),
  Funcao = factor(rep(c("Densidade Refletida N(0,1)",
    "Logística 1",
    "Logística 2",
    "Logística 3"), each = length(x))))

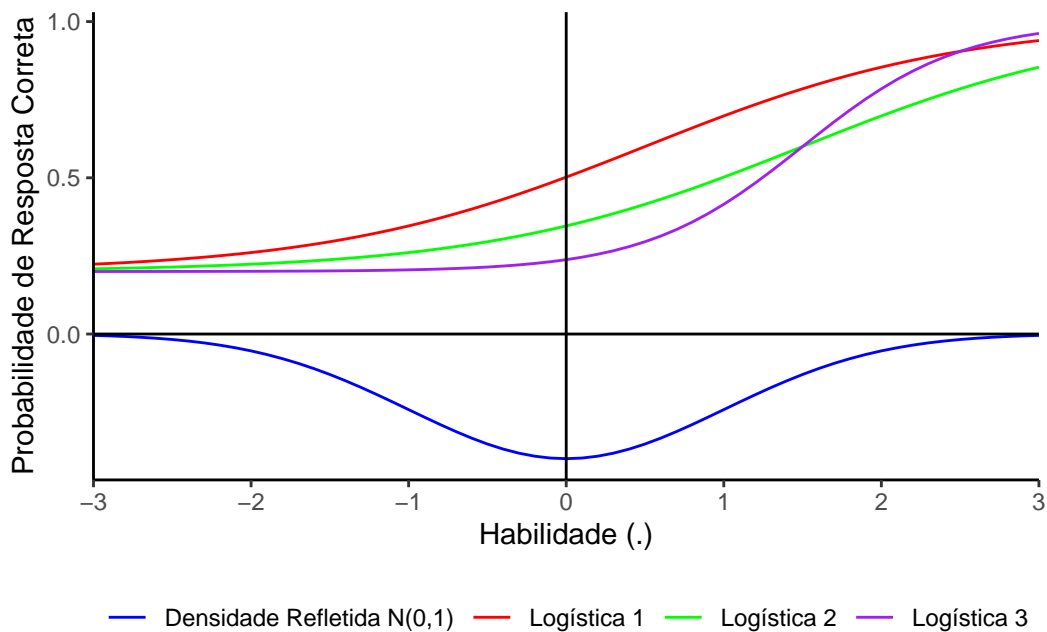
ggplot(dados, aes(x = x, y = y, color = Funcao)) +
  geom_line() +
  labs(x = "Habilidade ( )", y = "Probabilidade de Resposta Correta") +
```



```

scale_color_manual(values = c("Densidade Refletida N(0,1)" = "blue",
                              "Logística 1" = "red",
                              "Logística 2" = "green",
                              "Logística 3" = "purple")) +
theme(panel.background = element_blank(), axis.line = element_line(),
      legend.title = element_blank(), legend.position = "bottom") +
geom_hline(yintercept = 0) +
geom_vline(xintercept = 0) +
scale_x_continuous(limits = c(-3, 3), expand = c(0, 0))

```



## 2. Geração de Dados Simulados

**Figura 2.1** Histograma da Distribuição Gerada de uma  $v.a. X \sim U(0, 1)$ .

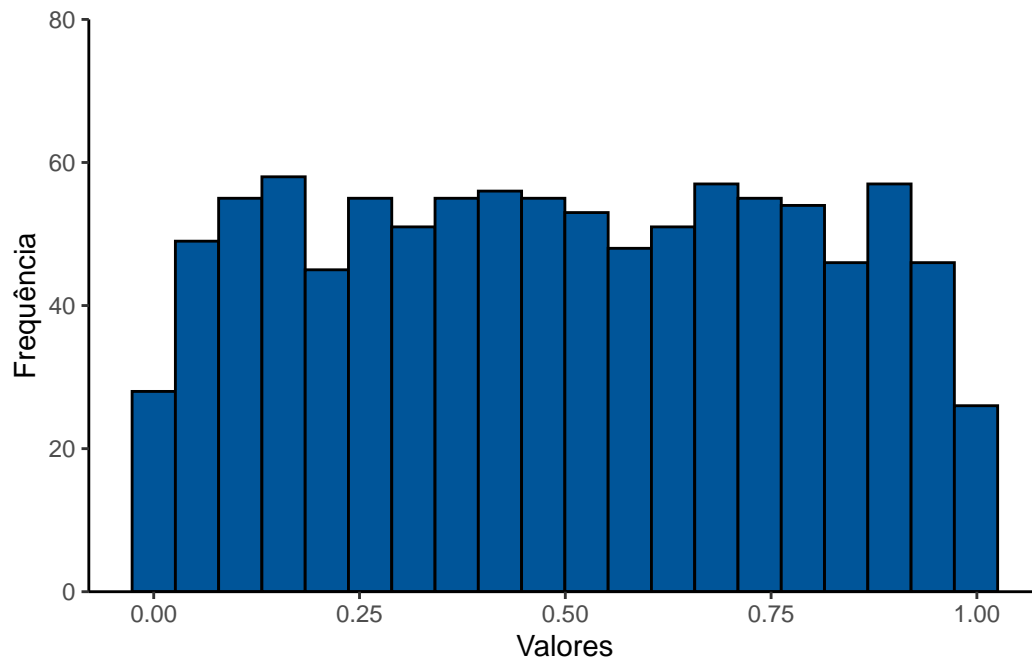
```

library(ggplot2)
set.seed(123)
valores <- runif(1000, min = 0, max = 1)
dados <- data.frame(valores = valores)

ggplot(dados, aes(x = valores)) +
  geom_histogram(bins = 20, fill = "#059", color = "#000") +
  labs(x = "Valores", y = "Frequência") +
  theme(panel.background = element_blank(),

```

```
axis.line = element_line()+
scale_y_continuous(limits = c(0, 80), expand = c(0, 0))
```



**Figura 2.2** Distribuição Gerada de uma v.a.  $X \sim \text{Bernoulli}(0.3)$ .

```
library(ggplot2)
set.seed(123)
p <- 0.3
n <- 1000
valores <- rbinom(n, size = 1, prob = p)

media_teorica <- p
variancia_teorica <- p * (1 - p)

media_empirica <- mean(valores)
variancia_empirica <- var(valores)

cat("Média teórica: ", media_teorica, "\n")
```

Média teórica: 0.3

```
cat("Média empírica: ", media_empirica, "\n")
```

Média empírica: 0.295

```
cat("Variância teórica: ", variancia_teorica, "\n")
```

Variância teórica: 0.21

```
cat("Variância empírica: ", variancia_empirica, "\n")
```

Variância empírica: 0.2081832

**Figura2.3.** Distribuição Gerada de uma v.a.  $Bin(10, 0.5)$ .

```
library(ggplot2)
set.seed(123)
n_trials <- 10
p <- 0.5
n_samples <- 1000

valores <- rbinom(n_samples, size = n_trials, prob = p)

media_empirica <- mean(valores)
variancia_empirica <- var(valores)

media_teorica <- n_trials * p
variancia_teorica <- n_trials * p * (1 - p)

cat("Média teórica: ", media_teorica, "\n")
```

Média teórica: 5

```
cat("Média empírica: ", media_empirica, "\n")
```

Média empírica: 4.975

```
cat("Variância teórica: ", variancia_teorica, "\n")
```

Variância teórica: 2.5

```
cat("Variância empírica: ", variancia_empirica, "\n")
```

Variância empírica: 2.556932

**2.4.** Distribuição Gerada de uma v.a.  $N(0, 1)$ .

```
library(ggplot2)
set.seed(123)

n_samples <- 1000
media <- 0
desvio_padrao <- 1
valores <- rnorm(n_samples, mean = media, sd = desvio_padrao)

media_teorica <- media
variancia_teorica <- desvio_padrao^2

media_empirica <- mean(valores)
variancia_empirica <- var(valores)

cat("Média teórica: ", media_teorica, "\n")
```

Média teórica: 0

```
cat("Média empírica: ", media_empirica, "\n")
```

Média empírica: 0.01612787

```
cat("Variância teórica: ", variancia_teorica, "\n")
```

Variância teórica: 1

```
cat("Variância empírica: ", variancia_empirica, "\n")
```

Variância empírica: 0.9834589

### 3. Simulando dados de avaliação

**3.1.** Gerando o  $n = 1000$  valores de uma  $N(0, 1)$  representando as habilidades dos indivíduos (x) na coluna A do Excel

```
library(openxlsx)
n <- 1000
a <- 1.5
b <- -0.5
set.seed(123)
habilidades <- rnorm(n, mean = 0, sd = 1)
probabilidades <- 1 / (1 + exp(-(a * habilidades + b)))
respostas <- rbinom(n, size = 1, prob = probabilidades)

dados <- data.frame(
  Habilidade = habilidades,
  Probabilidade = probabilidades,
  Resposta = respostas)

write.xlsx(dados, file = "C:/Users/Cassia Correa/OneDrive/Estatistica/Estatistica Educacional")
```

### 4. Otimização: obtenção do máximo de uma função

**4.1-** Valor que maximiza a densidade de uma  $N(0,1)$

---

**Microsoft Excel 16.0 Relatório de Respostas****Planilha: [Pasta1]Planilha1****Relatório Criado: 10/12/2024 10:53:37****Resultado: O Solver encontrou uma solução. Todas as Restrições e condições de adequação foram satisfeitas.****Mecanismo do Solver**

Mecanismo: GRG Não Linear

Tempo da Solução: 0,016 Segundos.

Iterações: 0 Subproblemas: 0

**Opções do Solver**

Tempo Máx. Ilimitado, Iterações Ilimitado, Precision 0,000001

Convergência 0,0001, Tamanho da População 0, Propagação Aleatória 0, Encaminhar Derivativos

Subproblemas Máx. Ilimitado, Soluç. Máx. Núm. Inteiro Ilimitado, Tolerância de Número Inteiro 1%, Assumir Não Negativo

**Célula do Objetivo (Máx.)**

Célula	Nome	Valor Original	Valor Final	
\$B\$2	f(x)	0,39894228	0,39894228	

**Células Variáveis**

Célula	Nome	Valor Original	Valor Final	Número Inteiro
\$A\$2	x	0	0	Conting.

**Restrições**

---

**NENHUM**

---

**4.2-**  $P(U = x) = p^x(1 - p)^{1-x}$  e uma amostra  $(x_1, \dots, x_{10}) = (0, 1, 0, 0, 1, 0, 0, 0, 1, 0)$  após a construção da função de verossimilhança (produto das probabilidades). Usado Suplemento Solver para obter o valor de  $p$  que maximiza a verossimilhança.

---

**Microsoft Excel 16.0 Relatório de Respostas****Planilha: [Pasta1]Planilha1****Relatório Criado: 10/12/2024 11:26:51****Resultado: O Solver fez uma convergência para a solução atual. Todas as Restrições foram satisfeitas.****Mecanismo do Solver**

Mecanismo: GRG Não Linear

Tempo da Solução: 0,062 Segundos.

Iterações: 7 Subproblemas: 0

**Opções do Solver**

Tempo Máx. Ilimitado, Iterações Ilimitado, Precision 0,000001

Convergência 0,0001, Tamanho da População 0, Propagação Aleatória 0, Encaminhar Derivativos

Subproblemas Máx. Ilimitado, Soluç. Máx. Núm. Inteiro Ilimitado, Tolerância de Número Inteiro 1%, Assumir Não Negativo

**Célula do Objetivo (Máx.)**

Célula	Nome	Valor Original	Valor Final
\$C\$7	Questão 4.1	-6,238324625	-5,72862751

**Células Variáveis**

Célula	Nome	Valor Original	Valor Final	Número Inteiro
\$B\$7	f(x)	0,5	0,333333333	Conting.

**Restrições**

Célula	Nome	Valor da Célula	Fórmula	Status	Margem de Atraso
\$A\$7	x	0	\$A\$7<=0	Associação	0
\$A\$7	x	0	\$A\$7>=0	Associação	0

## 5. Gerenciamento de Bases de Dados no EXCEL

Realizado o gerenciamento de um banco de dados no Excel utilizando macro VBA. Durante esse processo, geramos um código que contém as respostas correspondentes às alternativas escolhidas pelos participantes do ENEM 2017, no caderno azul, referentes às questões de matemática. O arquivo está no formato de texto formatado, com os valores separados por espaços



macro.prn	Questao 5.prn	×	+
Arquivo	Editar	Exibir	
170000007404	000001100000101000100111000010100000000100000		
170005101416	0000100000000000010010010001000100100000100010		
170000098218	101011000011101011011010001111001000011100100		
170005205041	100011011111001011010111111111000011011101111		
170005203958	100111000100100000001111000010000010100110010		
170000162158	111111110111001110111100111111011101101100110		
170000147521	101011110111101111111111001100010111110111111		
170000182413	100011000010000000000011110010100001001000000		
170001874145	100011000001101001110011001010101010011100110		
170003535712	111001111011000111111111001010100000000100000		
170001904052	101101000101000110000000001101000101010000100		
170003745189	100011000001000111000010111000000110100110001		
170000318552	0000010000000000010000010001010100000000100001		
170002007985	100001100111000001111011001010111100001000100		
170003680349	100010000001001100010111010010100110001111010		
170000378571	100011001001000001100010000010100000000001000		
170000409170	100001000000100001111010011010000100000000000		
170002199980	101011110011101010111111011000001100001010000		
170005814538	010000000001010011110010001010100001001001011		
170000773620	111011110011001101111110001010101001101110100		
170000782944	000010000001000001111010011010000111000000100		
170004174002	100011000001000000011011011100000000001100000		
170001071403	100001010001000011011111000110100000001111000		
170002572804	101011010001001011101110001010100011001101100		
170004277067	101001010011000011010110001010100000110010010		
170001012829	0001010000001010000010110000000000000001000000		
170005984750	1000010000000000110010011000010000001001100000		
170001057683	000000111010000100000101010000010000000000001		
170002724838	000011000111100000011111000010101100000000000		
170004473601	0001000001110100010000110000110000100000000000		
170004448890	000011010011100001101010101100101011000010001		
170006156563	101111110101001001110111000010101010011110100		
170004602121	100001010000001000110111010110000001110000000		
170006267075	100010000011000000010011001010000010101000001		
170001294810	000001000001000001010011001110000001110000000		
170001460046	1010110000000000000100001000000000000101011000		
170005148831	110111111001001001011011010010100000101100000		
170002458242	00000100001000000000000100010110000110101000100		



## 6. Gerenciamento de Bases de Dados no R

```
library(readxl)
library(ggplot2)

respostas <- read_excel("ENEM2017MT.xlsx")
gabarito <- read_csv("ENEM2017MT_infoitem.csv",
                    sep = ";", stringsAsFactors = FALSE)

gabarito_MT <- gabarito$GABARITO[1:45]

respostas_split <- data.frame(
  do.call(rbind, strsplit(substr(respostas$TX_RESPOSTAS_MT, 1, 45), "")))
colnames(respostas_split) <- paste0("Q", 1:45)

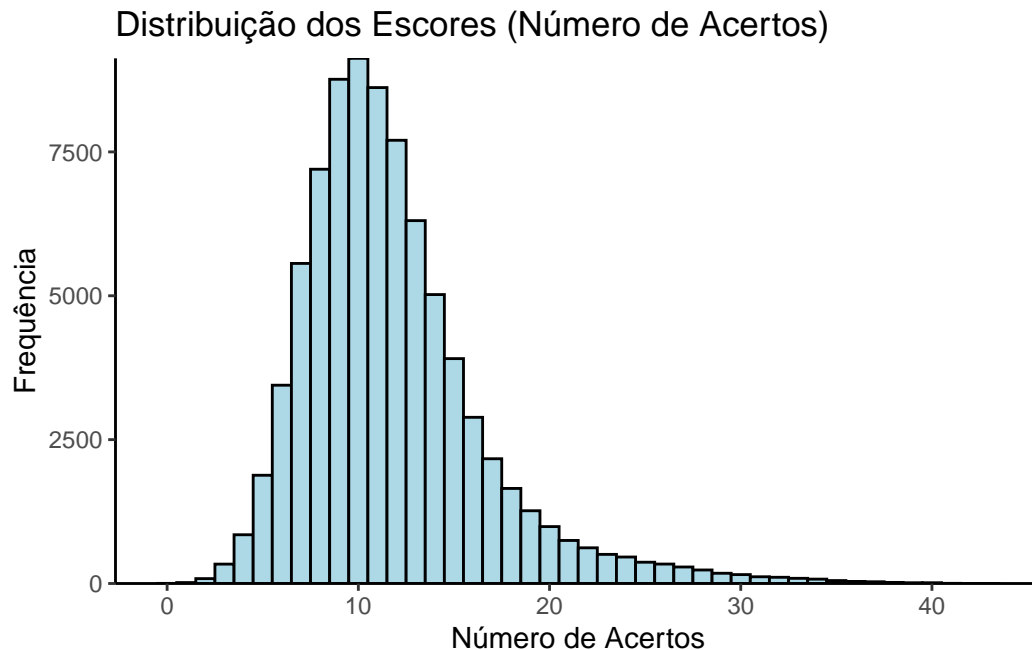
respostas_dicotomizadas <- respostas_split
for (i in 1:45) {
  respostas_dicotomizadas[[i]] <- ifelse(
    respostas_split[[i]] == gabarito_MT[i], 1, 0)
}

respostas_dicotomizadas <- cbind(NU_INSCRICAO = respostas$NU_INSCRICAO, respostas_dicotomizadas)
respostas_dicotomizadas$SCORE <- rowSums(respostas_dicotomizadas[2:46])

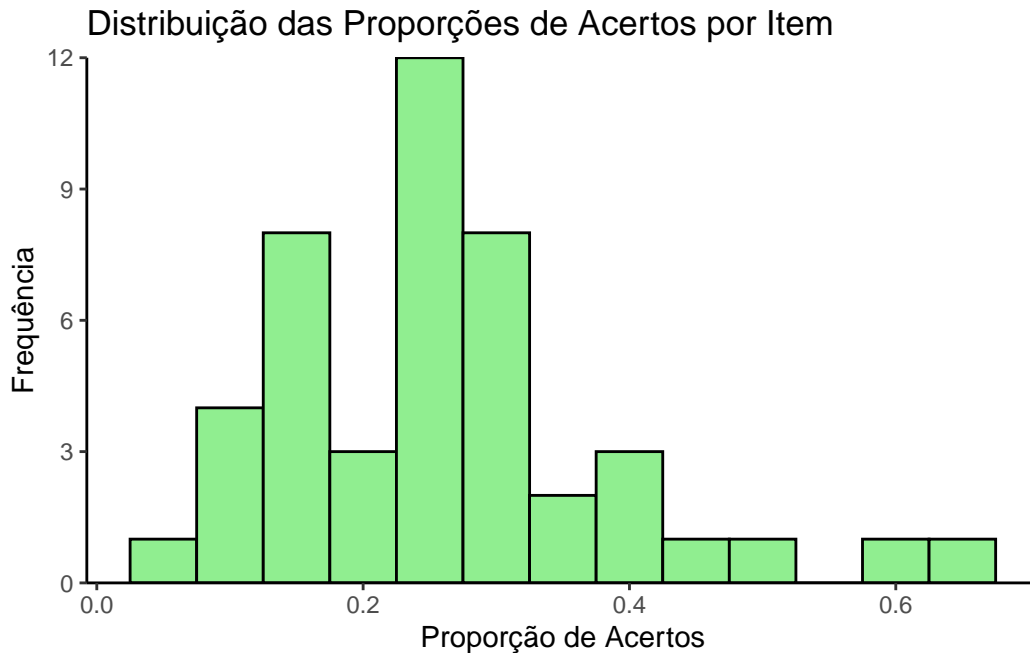
respostas_formatadas <- apply(
  respostas_dicotomizadas[, 1:46], 1, function(row) {
    paste(substr(row[1], 1, 12), paste(row[-1], collapse = ""), sep = " ")
  })
cat(head(respostas_formatadas, 12), sep = "\n")
```

```
170000007404 0000011000001010001001110000101000000000100000
170005101416 0000100000000000010010010001000100100000100010
170000098218 101011000011101011011010001111001000011100100
170005205041 100011011111001011010111111111000011011101111
170005203958 10011100010010000000011110000100000010100110010
170000162158 111111111011100111011110011111101110101100110
170000147521 101011110111101111111110011000101111101111111
170000182413 1000110000100000000000011110010100001001000000
170001874145 100011000001101001110011001010101010011100110
170003535712 111001111011000111111111001010100000000100000
170001904052 1011010001010001100000000001101000101010000100
170003745189 100011000001000111000010111000000110100110001
```

```
ggplot(respostas_dicotomizadas, aes(x = SCORE)) +
  geom_histogram(binwidth = 1, color = "black", fill = "lightblue") +
  labs(title = "Distribuição dos Escores (Número de Acertos)",
       x = "Número de Acertos",
       y = "Frequência") +
  theme(panel.background = element_blank(),
        axis.line = element_line()) +
  scale_y_continuous(expand = c(0, 0))
```



```
proporcao_acertos <- colMeans(respostas_dicotomizadas[2:46])
ggplot(data.frame(Proporcao = proporcao_acertos), aes(x = Proporcao)) +
  geom_histogram(binwidth = 0.05, color = "black", fill = "lightgreen") +
  labs(title = "Distribuição das Proporções de Acertos por Item",
       x = "Proporção de Acertos",
       y = "Frequência") +
  theme(panel.background = element_blank(),
        axis.line = element_line()) +
  scale_y_continuous(expand = c(0, 0))
```



#### 7. Gráfico da Proporção de acerto por escore para os 5 primeiros itens.

```
library(readxl)
library(ggplot2)
respostas <- read_excel("ENEM2017MT.xlsx")
gabarito <- read_csv("ENEM2017MT_infoitem.csv",
                    sep = ";", stringsAsFactors = FALSE)
gabarito_MT <- gabarito$GABARITO[1:45]
respostas_split <- data.frame(
  do.call(rbind, strsplit(substr(respostas$TX_RESPOSTAS_MT, 1, 45), "")))
colnames(respostas_split) <- paste0("Q", 1:45)
respostas_dicotomizadas <- respostas_split
for (i in 1:45) {
  respostas_dicotomizadas[[i]] <- ifelse(
    respostas_split[[i]] == gabarito_MT[i], 1, 0)
}

respostas_dicotomizadas$SCORE <- rowSums(respostas_dicotomizadas)

proporcao_por_escore <- lapply(1:5, function(i) {
  tapply(respostas_dicotomizadas[[i]], respostas_dicotomizadas$SCORE, mean)})

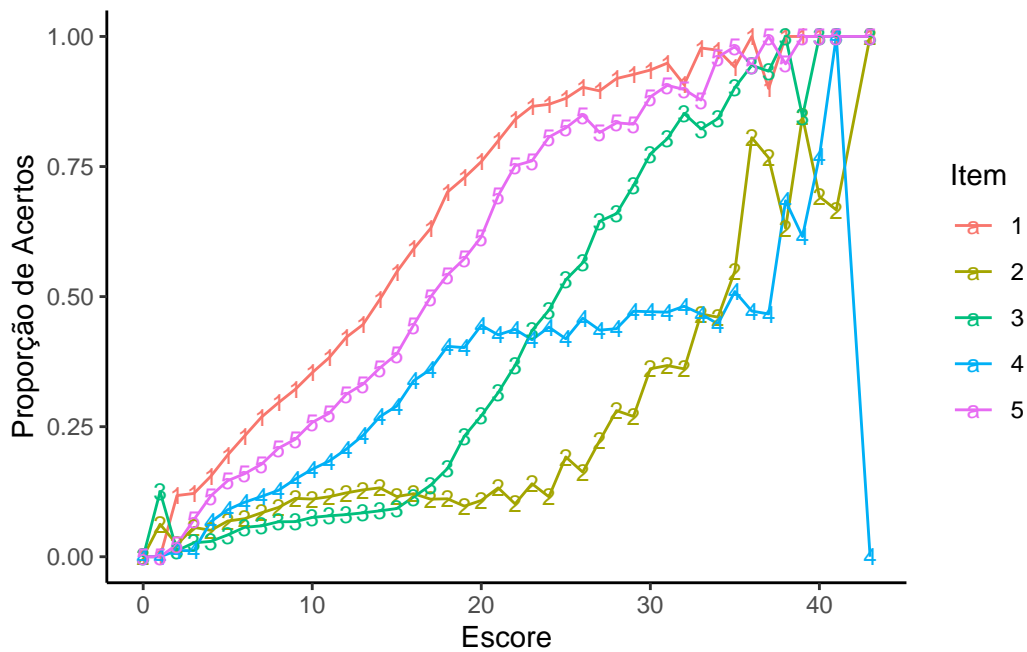
df_plot <- data.frame(
  Escore = rep(as.numeric(names(proporcao_por_escore[[1]])), 5),
```

```

Proporcao = unlist(proporcao_por_escore),
Item = rep(1:5, each = length(proporcao_por_escore[[1]])))

ggplot(df_plot, aes(
  x = Escore, y = Proporcao, color = factor(Item), label = Item)) +
  geom_line() + geom_text(size = 3) +
  labs(x = "Escore", y = "Proporção de Acertos", color = "Item") +
  theme(panel.background = element_blank(),
        axis.line = element_line()) +
  scale_y_continuous(limits = c(0, 1))

```



## 8. Histograma Por Área do Conhecimento.

```

library(readr)
library(ggplot2)
library(rlang)

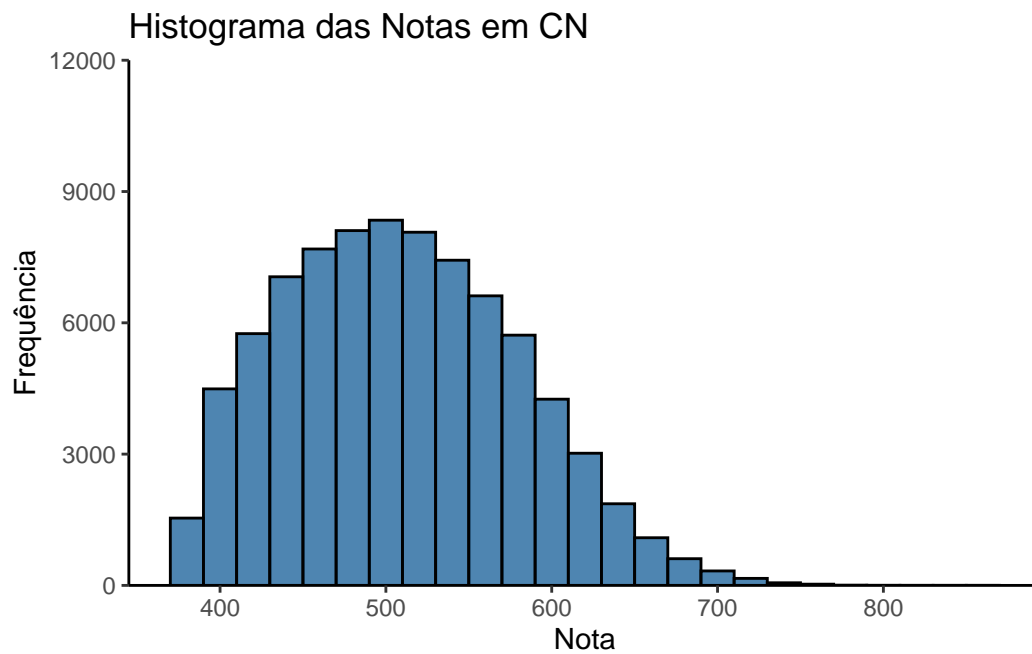
itens_prova <- read.csv("ITENS_PROVA_2017.csv", sep = ";", encoding = "latin1")
enem_data <- read.csv("ENEM2017.csv", sep = ",", encoding = "latin1")
lc_data <- enem_data[, c("NU_NOTA_LC", "TX_RESPOSTAS_LC", "TX_GABARITO_LC")]

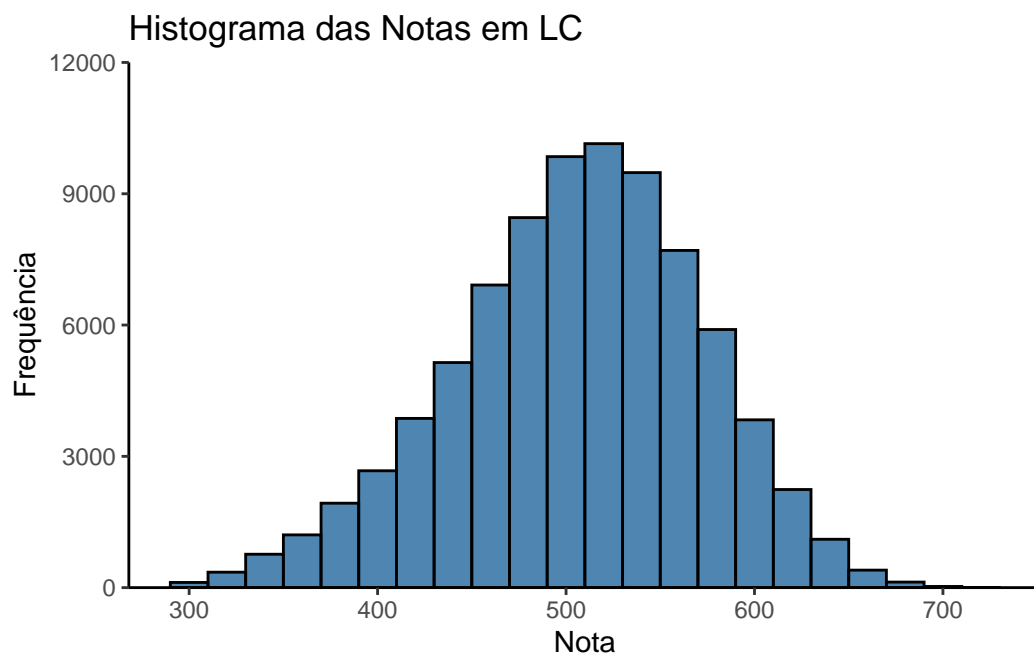
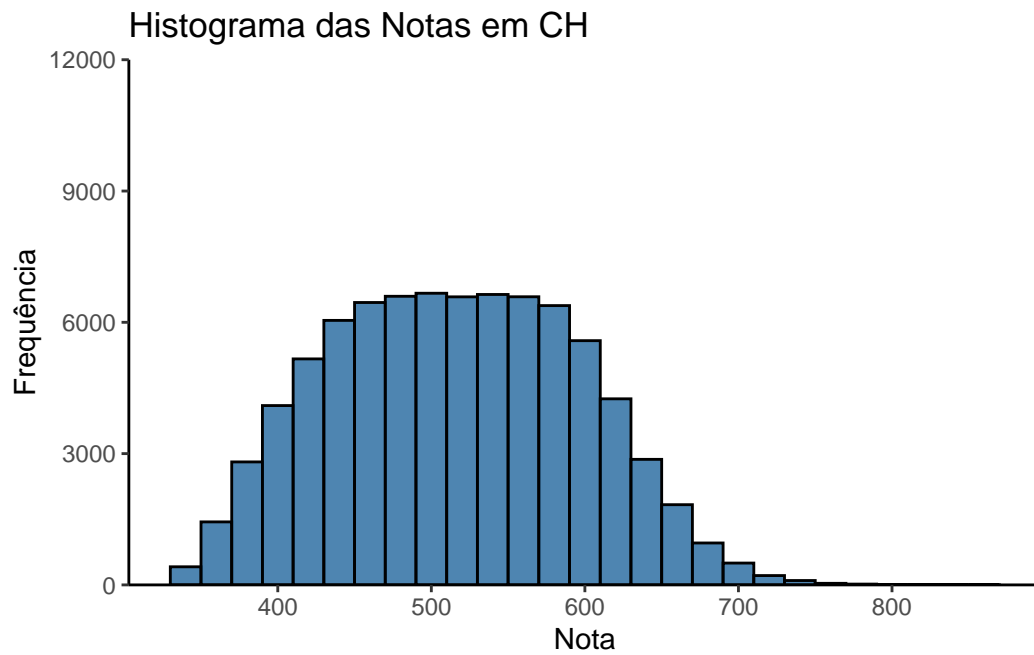
areas <- c("NU_NOTA_CN", "NU_NOTA_CH", "NU_NOTA_LC", "NU_NOTA_MT")

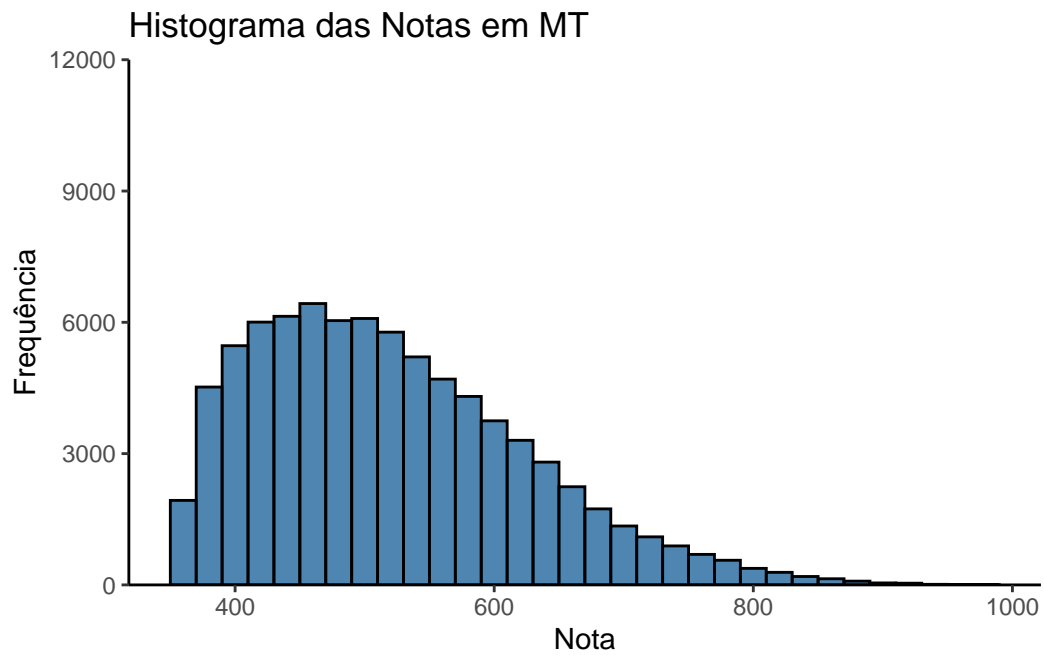
```

```
for (area in areas) {
  p <- ggplot(enem_data, aes(x = !!sym(area))) +
    geom_histogram(binwidth = 20, fill = "#005090",
                   color = "black", alpha = 0.7) +
    labs(title = paste("Histograma das Notas em", gsub("NU_NOTA_", "", area)),
         x = "Nota", y = "Frequência") +
    theme(panel.background = element_blank(), axis.line = element_line()) +
    scale_y_continuous(limits = c(0, 12000), expand = c(0, 0))
  print(p)}

```







## Tarefa 2

```
1 + 1
```

```
[1] 2
```