

# Introdução Programação em R com GitHub, ChatGPT e Claude

Vinícius Silva Junqueira

2025-10-07

## Sumário

<b>1</b>	<b>Dia 1 — Fundamentos e Ambiente de Trabalho</b>	<b>1</b>
1.1	1. Ambientação e Setup ( 70 min) . . . . .	1
1.2	2. Fundamentos de R ( 40 min) . . . . .	3
1.3	3. Exploração inicial de dados ( 30 min) . . . . .	4
1.4	4. Entrada/Saída (I/O) e encoding ( 15 min) . . . . .	5
1.5	5. Exercícios guiados ( 20–25 min) . . . . .	5
1.6	6. Primeiro commit ( 5 min) . . . . .	6
1.7	7. Checklist de encerramento . . . . .	6
1.8	8. Referências rápidas . . . . .	6

## 1 Dia 1 — Fundamentos e Ambiente de Trabalho

### Objetivos do dia

- Preparar o ambiente (R, RStudio, Git, GitHub) e consolidar o **workflow com fork**.
- Conhecer **objetos básicos** de R: vetores, listas, data.frames, fatores.
- Explorar dados com funções descritivas e boas práticas de **portabilidade** (projetos .Rproj e here).
- Fazer o **primeiro commit** no SEU fork no GitHub.

**Tempo previsto** 19h00–22h00 (intervalo 20h30–20h50)

---

### 1.1 1. Ambientação e Setup ( 70 min)

#### 1.1.1 1.1 Verificações rápidas

```
R.version.string          # Versão do R
# RStudio.Version()$version # Versão do RStudio
# system("git --version")   # Confirma Git disponível
```

### 1.1.2 1.2 Configurar Git (uma vez só)

No **Terminal do RStudio** (funciona em Windows/macOS/Linux):

```
git config --global user.name "Seu Nome"
git config --global user.email "seu@email.com"
# Verificar
git config --global --list
```

### 1.1.3 1.3 Autenticar no GitHub (PAT recomendado)

```
# install.packages("usethis"); install.packages("gitcreds")
# usethis::create_github_token() # abre navegador
# gitcreds::gitcreds_set()      # cole o token
```

**Alternativas:** GitHub Desktop (GUI) ou SSH (avançado).

---

### 1.1.4 1.4 Workflow com Fork (obrigatório para a turma)

Original (instrutor) → FORK (sua conta) → CLONE (seu PC) → PUSH (para seu fork)

1. Abra: <https://github.com/viniciusjunqueira/curso-r-github-ia>
2. Clique **Fork** → escolha **sua conta** → *Create fork*.
3. Clone **SEU fork**:

```
git clone https://github.com/SEU-USUARIO/curso-r-github-ia.git
cd curso-r-github-ia
```

4. Abra o projeto .Rproj no RStudio.

5. **Cheque o remote:**

```
git remote -v
# Deve mostrar seu usuário em origin
```

**Por que fork?** Você controla seu repositório, faz commits/push à vontade e não altera o repo do instrutor.

---

### 1.1.5 1.5 Estrutura de projeto e portabilidade

```
curso-r-github-ia/
  curso-r-github-ia.Rproj
  data/
    raw/
    processed/
  scripts/
```

```
output/  
  figures/  
  tables/  
docs/
```

```
# Caminhos: sempre prefira here::here()  
# install.packages("here")  
# library(here)  
# caminho <- here("data", "raw", "dados.csv")  
# caminho
```

**UTF-8:** salve arquivos com File → Save with Encoding → UTF-8 (evita problemas de acentuação em todos os SOs).

---

## 1.2 2. Fundamentos de R ( 40 min)

### 1.2.1 2.1 Objetos básicos e operações

```
# Números, lógicos, strings  
x_num <- 3.14; x_log <- TRUE; x_chr <- "Olá, R!"  
class(x_num); typeof(x_num)  
class(x_log); typeof(x_log)  
class(x_chr); typeof(x_chr)  
  
# Aritmética  
10 + 2; 10 - 2; 10 * 2; 10 / 3; 2 ^ 3  
  
# Especiais  
Inf; -Inf; NaN; NA
```

### 1.2.2 2.2 Vetores e indexação

```
v <- c(10, 20, 30, 40, 50)  
length(v); mean(v); sum(v)  
  
v[1]; v[2:4]; v[-1]  
sel <- v > 25; sel; v[sel]  
  
names(v) <- letters[1:5]  
v["c"]
```

### 1.2.3 2.3 Listas e data.frames

```
# Lista: tipos mistos  
lst <- list(id = 1, nome = "Ana", aprovado = TRUE)  
lst$nome
```

```
# Data frame
alunos <- data.frame(
  id = 1:4,
  nome = c("Ana", "Bruno", "Caio", "Dani"),
  nota = c(8.5, 7.2, 9.1, 6.8),
  ativo = c(TRUE, TRUE, FALSE, TRUE),
  stringsAsFactors = FALSE
)
str(alunos); nrow(alunos); ncol(alunos); names(alunos)
head(alunos, 2); tail(alunos, 2)

# Acesso e novas colunas
alunos$nome
alunos$aprov <- ifelse(alunos$nota >= 7, "Aprovado", "Recuperação")
```

#### 1.2.4 2.4 Fatores

```
sexo <- factor(c("F", "M", "M", "F"), levels = c("F", "M"))
levels(sexo)

conceito <- factor(c("B", "A", "C", "A"), levels = c("C", "B", "A"), ordered = TRUE)
summary(conceito)
```

---

### 1.3 3. Exploração inicial de dados ( 30 min)

Vamos usar um dataset real (palmerpenguins) para praticar **inspeção e resumo**.

```
# install.packages("palmerpenguins")
# library(palmerpenguins)
# library(dplyr)

# dplyr::glimpse(penguins)
# names(penguins)
# summary(penguins)
# colSums(is.na(penguins))

# Selecionar colunas principais
# peng_min <- penguins[, c("species", "bill_length_mm", "bill_depth_mm", "flipper_length_mm", "body_mass_g")]
# head(peng_min)

# Nova variável: razão do bico
# penguins$raz_bico <- with(penguins, bill_length_mm / bill_depth_mm)

# Estatísticas
# mean(penguins$flipper_length_mm, na.rm = TRUE)
# tapply(penguins$flipper_length_mm, penguins$species, mean, na.rm = TRUE)
```

**Dica:** quando houver NAs, use `na.rm = TRUE` nas funções de resumo.

---

## 1.4 4. Entrada/Saída (I/O) e encoding ( 15 min)

```
# library(readr)
# library(here)

# Leitura CSV em UTF-8
# dados <- readr::read_csv(here("data", "raw", "dados.csv"), locale = readr::locale(encoding = "UTF-8"))

# Para arquivos Latin1 (com acentos vindos do Windows)
# dados <- readr::read_csv(here("data", "raw", "dados.csv"), locale = readr::locale(encoding = "Latin1"))

# Excel
# library(readxl)
# plan <- readxl::read_excel(here("data", "raw", "planilha.xlsx"), sheet = 1)

# Salvando sempre em UTF-8
# readr::write_csv(plan, here("output", "tables", "planilha_export.csv"))
```

**Portabilidade:** evite caminhos absolutos; prefira `here()` dentro de um projeto `.Rproj`.

---

## 1.5 5. Exercícios guiados ( 20–25 min)

### 1.5.1 Exercício 1 — Vetores

1. Crie um vetor numérico de 8 valores.
2. Calcule média, mediana e desvio-padrão.
3. Filtre os valores **acima da média**.

*# Seu código aqui*

### 1.5.2 Exercício 2 — Data frame

1. Crie um `data.frame` com colunas: `id`, `nome`, `nota`, `ativo`.
2. Crie `situacao` com `ifelse(nota >= 7, "Aprovado", "Recuperação")`.
3. Mostre só `nome` e `situacao` das 2 primeiras linhas.

*# Seu código aqui*

### 1.5.3 Exercício 3 — Exploração palmerpenguins

1. Conte NAs por coluna.
2. Crie `massa_kg = body_mass_g/1000`.
3. Calcule a **média de flipper\_length\_mm** por espécie.

*# Seu código aqui*

---

## 1.6 6. Primeiro commit ( 5 min)

No **Terminal do RStudio**:

```
git add scripts/01_fundamentos.R
git commit -m "Dia 1: fundamentos de R e setup"
git push origin main
```

Confirme no seu repositório **forkado** no GitHub se apareceu o commit.

---

## 1.7 7. Checklist de encerramento

- ☐ R, RStudio e Git instalados.
  - ☐ Git configurado (`user.name` e `user.email`).
  - ☐ **Fork criado e clone do SEU fork.**
  - ☐ Projeto `.Rproj` aberto; `here()` funcionando.
  - ☐ Script `01_fundamentos.R` salvo em UTF-8.
  - ☐ Commit/push realizados para **SEU fork**.
- 

## 1.8 8. Referências rápidas

- **R for Data Science (2e)**: <https://r4ds.hadley.nz/>
- **Happy Git with R**: <https://happygitwithr.com/>
- **Cheatsheets Posit**: <https://posit.co/resources/cheatsheets/>
- **palmerpenguins**: <https://allisonhorst.github.io/palmerpenguins/>