

Introdução ao Programa em R com GitHub, ChatGPT e Claude

Análise de Dados Moderna para Iniciantes de Diversas áreas

Vinicius Silva Junqueira

2025-10-06

Contents

1	Informações Gerais do Curso	3
1.1	Estrutura Pedagógica	3
1.2	Materiais Necessários	3
2	Segunda-feira Dia 1: Fundamentos e Ambiente Reprodutível	4
2.1	19h00 20h30 Ambiente e Setup Completo	4
2.2	20h30 20h50 INTERVALO	4
2.3	20h50 22h00 Fundamentos do R	4
2.4	Para Casa	5
3	Terça-feira Dia 2: Lógica, Funções e Boas Práticas	6
3.1	19h00 20h30 Programa em R	6
3.2	20h30 20h50 INTERVALO	6
3.3	20h50 22h00 Boas Práticas e IA Assistida	6
4	Quarta-feira Dia 3: Manipulação Moderna de Dados	8
4.1	19h00 20h30 Tidyverse e dplyr	8
4.2	20h30 20h50 INTERVALO	8
4.3	20h50 22h00 Transformação e Limpeza	8
5	Quinta-feira Dia 4: I/O de Dados e Visualização	10
5.1	19h00 20h30 Entrada e Saída de Dados	10
5.2	20h30 20h50 INTERVALO	10
5.3	20h50 22h00 Visualização com ggplot2	11
6	Sexta-feira Dia 5: Relatórios Reprodutíveis e Projeto Final	12
6.1	19h00 20h30 RMarkdown/Quarto	12
6.2	20h30 20h50 INTERVALO	12
6.3	20h50 22h00 Projeto Final e GitHub	13
7	Recursos Adicionais	15
7.1	Materiais Complementares	15
7.2	Usando IA de Forma Tática e Eficiente	15
7.3	Critérios de Avaliação do Projeto Final	16

7.4 Contato e Suporte Pós-Curso 16

8 Feedback dos Participantes 16

1 Informa es Gerais do Curso

Carga hor ria total: 16 horas

Per odo: Segunda a sexta-feira

Hor rio: 19h00 s 22h00 (com intervalo de 20 minutos s 20h30)

Tempo l quido por dia: 2h40 de conte do efetivo

P blico-alvo: Iniciantes de diversas reas (sa de, economia, biologia, ci ncias sociais, gest o)

Material: Reposit rio GitHub com datasets, scripts e exerc cios

Reposit rio: <https://github.com/viniciusjunqueira/curso-r-github-ia>

Pr -requisito: Computador com R, RStudio e Git instalados

1.1 Estrutura Pedag gica

- **Teoria + Pr tica:** Cada conceito seguido de exerc cio pr tico
 - **Aprendizado Incremental:** Commits no GitHub ao final de cada dia
 - **Casos Multidisciplinares:** Exemplos de sa de, economia, biodiversidade e marketing
 - **IA como Ferramenta:** Uso estrat gico de ChatGPT e Claude para aprendizado e produtividade
 - **Projeto Final:** An lise completa com dados reais e relat rio reproduz vel
-

1.2 Materiais Necess rios

1.2.1 Instala o Pr via (Obrigat rio)

- R (vers o 4.3+): <https://cran.r-project.org/>
- RStudio (vers o 2023.09+): <https://posit.co/download/rstudio-desktop/>
- Git: <https://git-scm.com/downloads>
- Conta no GitHub: <https://github.com/>

1.2.2 Pacotes R (instalados no Dia 1)

```
install.packages(c("tidyverse", "here", "janitor", "skimr",  
                  "readxl", "writexl", "rmarkdown"))
```

2 Segunda-feira Dia 1: Fundamentos e Ambiente Reprodutível

2.1 19h00 20h30 | Ambiente e Setup Completo

2.1.1 Apresentação do Curso (20min)

- Objetivos e metodologia
- Por que R, GitHub e IA?
- Panorama das áreas de aplicação
- Apresentação dos participantes e expectativas

2.1.2 Configuração do Ambiente (50min)

- Verificação das instalações (R, RStudio, Git)
- **Criação de conta GitHub** (se ainda não tiver)
- **Configuração do Git local:**
 - `git config --global user.name "Seu Nome"`
 - `git config --global user.email "seu@email.com"`
- **Autenticação do GitHub:**
 - Criação de Personal Access Token (PAT)
 - Configuração com `gitcreds::gitcreds_set()`
 - Alternativa: GitHub Desktop (interface gráfica)

2.1.3 Clonagem do Repositório do Curso (20min)

- Clone via RStudio ou linha de comando
- Estrutura de pastas: `data/`, `scripts/`, `output/`, `docs/`
- **Criação do primeiro Projeto RStudio (.Rproj)**
- Importância do working directory e `here::here()`

2.2 20h30 20h50 | INTERVALO

2.3 20h50 22h00 | Fundamentos do R

2.3.1 Objetos e Estruturas de Dados (40min)

- Vetores: numérico, lógico, string
- Listas e `data.frames`
- Fatores: variáveis categóricas
- Funções básicas: `c()`, `length()`, `class()`, `typeof()`

2.3.2 Exploração Inicial de Dados (30min)

- `str()`, `head()`, `tail()`, `names()`
- `dplyr::glance()` (visão moderna)
- `summary()` para estatísticas descritivas
- **Indexação:** por posição `[1]`, por nome `$coluna`, por condição

2.3.3 Prática Guiada (20min)

- Criar vetores e data.frames
- Manipular objetos básicos
- **Primeiro script:** salvar como `01_fundamentos.R`
- **Primeiro commit:** “Fundamentos do R - Dia 1”

2.4 Para Casa

- Revisar conceitos no material do repositório
 - Experimentar com seus próprios dados (se tiver)
-

3 Ter a-feira Dia 2: Lógica, Funções e Boas Práticas

3.1 19h00 20h30 | Programa o em R

3.1.1 Operadores e Condicionais (35min)

- **Operadores lógicos:** `&`, `|`, `!`, `%in%`
- **Operadores relacionais:** `==`, `!=`, `>`, `<`, `>=`, `<=`
- **Condicionais:**
 - `if` e `else` (estrutura completa)
 - `ifelse()` vetorizado
 - `dplyr::case_when()` para mltiplas condi es

3.1.2 Loops e Itera o (25min)

- Loop `for`: sintaxe e casos de uso
- Vetoriza o vs. loops (efici ncia)
- Breve men o a `while` e `repeat`
- Fam lia `apply()` (conceitual)

3.1.3 Fun es Customizadas (30min)

- Sintaxe: `function(argumentos) { corpo }`
 - Argumentos obrigat rios e opcionais
 - Valor de retorno: `return()` vs. ltima linha
 - Documenta o interna com coment rios
 - **Exemplo pr tico:** fun o para calcular IMC por categoria
-

3.2 20h30 20h50 | INTERVALO

3.3 20h50 22h00 | Boas Práticas e IA Assistida

3.3.1 Boas Práticas de C digo (25min)

- **Naming conventions:** `snake_case` para objetos e fun es
- **Coment rios eficientes:** o qu vs. por qu
- **Organiza o:** uma opera o por linha
- **Estilo:** espa amento, indenta o
- Pacotes teis: `styler::style_file()`, `lintr`

3.3.2 Debugging e Gest o de Erros (25min)

- **Lendo mensagens de erro:** anatomy of an error
- **Estrat gias de debugging:**
 - `print()` para check points
 - `traceback()` para rastrear erros
 - `debug()` e `undebug()`

- Breakpoints no RStudio
- **Tipos comuns de erro:** syntax, object not found, type mismatch

3.3.3 IA como Assistente de Código (20min)

- **ChatGPT vs Claude:** quando usar cada um
 - ChatGPT: debugging rápido, sugestões de código
 - Claude: revisão detalhada, explicações conceituais, documentação
- **Boas práticas:**
 - Fornecer contexto adequado
 - Verificar e testar código gerado
 - Aprender com as sugestões (não apenas copiar)
 - Prática: sempre entender o código que você usa

3.3.4 Prática: Depuração Assistida (20min)

- Corrigir scripts com erros intencionais
 - Usar ChatGPT/Claude para explicar problemas
 - Criar função complexa com assistência de IA
 - **Commit:** “Lógica, funções e debugging - Dia 2”
-

4 Quarta-feira Dia 3: Manipula o Moderna de Dados

4.1 19h00 20h30 | Tidyverse e dplyr

4.1.1 Filosofia Tidyverse (15min)

- Princípios: tidy data, pipe, verbos consistentes
- Carregar o tidyverse: `library(tidyverse)`
- Estrutura de dados tidy: uma observação por linha

4.1.2 Verbos Essenciais do dplyr (45min)

- `filter()`: filtrar linhas por condição
- `select()`: selecionar colunas
- `mutate()`: criar/modificar colunas
- `arrange()`: ordenar linhas
- `summarize()`: resumir dados
- `group_by()`: agrupar para operações
- Operador pipe: `%>%` ou `|>` (native pipe)

4.1.3 Exemplo Integrado (30min)

- Dataset: dados de saúde pública (ex: vacinação por estado)
 - Pipeline completo com todos os verbos
 - Casos multidisciplinares:
 - Saúde: taxa de vacinação por região
 - Economia: PIB per capita por país
 - Biologia: abundância de espécies por bioma
-

4.2 20h30 20h50 | INTERVALO

4.3 20h50 22h00 | Transformação e Limpeza

4.3.1 Tidy para Reshape (30min)

- `pivot_longer()`: wide to long
- `pivot_wider()`: long to wide
- Casos de uso: dados de painel, séries temporais
- `separate()` e `unite()`: manipular strings em colunas

4.3.2 Tratamento de Valores Ausentes (20min)

- Identificar: `is.na()`, `anyNA()`, `sum(is.na())`
- Remover: `na.omit()`, `drop_na()`, `complete.cases()`
- Imputar: `replace_na()`, `fill()`
- Decisões: quando remover vs. imputar

4.3.3 Ferramentas Modernas de Exploração (20min)

- `janitor::clean_names()`: padronizar nomes de colunas
- `janitor::tabyl()`: tabelas de frequência
- `skimr::skim()`: sumário estatístico completo
- `dplyr::count()` e `add_count()`

4.3.4 Prática com Claude (20min)

- Criar pipeline complexo
- Usar Claude para:
 - Otimizar código
 - Sugerir alternativas
 - Explicar cada etapa
 - Documentar o pipeline
- **Commit:** “Manipulação e limpeza com tidyverse - Dia 3”

4.3.5 Exercício Para Casa

- Pipeline de análise com dataset de sua área
 - Documentar cada etapa com comentários
-

5 Quinta-feira Dia 4: I/O de Dados e Visualiza o

5.1 19h00 20h30 | Entrada e Sa da de Dados

5.1.1 Leitura de Arquivos (40min)

- **CSV:** `readr::read_csv()` vs. `read.csv()`
 - Vantagens do `readr`: velocidade, consist ncia, encoding
 - Par metros: `col_types`, `na`, `skip`, `locale`
- **Excel:** `readxl::read_excel()`
 - Especificar sheets: `sheet = "nome"` ou `sheet = 2`
 - Ranges: `range = "A1:D50"`
- **Outros formatos:** `.rds`, `.txt`, `.tsv`
- **Encoding:** lidar com acentua o (UTF-8)

5.1.2 Escrita de Arquivos (20min)

- **CSV:** `readr::write_csv()`
- **Excel:** `writexl::write_xlsx()`
- **RDS:** `saveRDS()` e `readRDS()` (formato R nativo)
- Boas pr ticas: datas no nome de arquivo, versionamento

5.1.3 Organiza o de Projetos (30min)

- **Estrutura recomendada:**

```
meu-projeto/  
  meu-projeto.Rproj  
  README.md  
  data/  
    raw/          # dados originais (nunca modificar!)  
    processed/    # dados limpos  
  scripts/  
    01_import.R  
    02_clean.R  
    03_analyze.R  
  output/  
    figures/  
    tables/  
  docs/           # relat rios
```

- **here::here()** para caminhos relativos
 - Portabilidade entre sistemas
 - Exemplo: `read_csv(here("data", "raw", "dados.csv"))`
- **Projetos RStudio:** sempre usar!

5.2 20h30 20h50 | INTERVALO

5.3 20h50 22h00 | Visualiza o com ggplot2

5.3.1 Gram tica de Gr ficos (20min)

- Filosofia: camadas sequenciais
- Estrutura b sica: `ggplot(data, aes()) + geom_*()`
- **Aesthetics:** `x`, `y`, `color`, `fill`, `size`, `alpha`
- **Geometries:** pontos, linhas, barras, boxplots

5.3.2 Tipos de Gr ficos (50min)

- **Dispers o:** `geom_point()` + `geom_smooth()`
 - Caso: rela o entre vari veis cont nuas
- **Barras:** `geom_col()` vs. `geom_bar()`
 - Caso: compara o entre categorias
- **Boxplot:** `geom_boxplot()` + `geom_jitter()`
 - Caso: distribui o e outliers
- **Linhas:** `geom_line()` + `geom_ribbon()`
 - Caso: s ries temporais
- **Histograma:** `geom_histogram()` + `geom_density()`
 - Caso: distribui o de frequ ncias

5.3.3 Personaliza o (20min)

- **Temas:** `theme_minimal()`, `theme_bw()`, `theme_classic()`
- **T tulos e labels:** `labs(title, x, y, caption)`
- **Cores:** `scale_color_brewer()`, `scale_fill_viridis_d()`
- **Facetas:** `facet_wrap()`, `facet_grid()`
- **Salvamento:** `ggsave("figura.png", width = 8, height = 6, dpi = 300)`

5.3.4 Pr tica com IA (10min)

- Usar ChatGPT/Claude para:
 - Sugerir tipo de gr fico adequado aos dados
 - Melhorar est tica
 - Corrigir erros de sintaxe
 - **Commit:** “I/O e visualiza o - Dia 4”
-

6 Sexta-feira Dia 5: Relat rios Reprodut veis e Projeto Final

6.1 19h00 20h30 | RMarkdown/Quarto

6.1.1 Introdu o Programa o Liter ria (20min)

- **Conceito:** c digo + narrativa + resultados
- **Reprodutibilidade:** garantir que an lise possa ser replicada
- RMarkdown vs. Quarto (breve compara o)
- Cria o de novo documento: File New File R Markdown

6.1.2 Estrutura de Documentos (40min)

- **YAML header:**

```
---  
title: "Minha An lise"  
author: "Seu Nome"  
date: "2025-10-06"  
output: html_document  
---
```

- **Markdown b sico:**

- Headers: #, ##, ###
- Listas: - ou 1.
- nfase: **it lico**, ****negrito****
- Links: [texto](url)

- **Chunks de c digo:**

```
``` r  
seu c digo aqui
```
```

- **Op es teis:** echo, eval, warning, message, fig.width, fig.height

6.1.3 Renderiza o (20min)

- Knit para HTML, PDF (requer LaTeX), Word
- Inline code: usar acento grave + r + c digo entre acentos graves
- Tabelas formatadas: `knitr::kable()`, `gt::gt()`
- **Pr tica:** criar relat rio simples com an lise do Dia 3

6.2 20h30 20h50 | INTERVALO

6.3 20h50 22h00 | Projeto Final e GitHub

6.3.1 Desenvolvimento do Projeto (40min)

- **Escolha do dataset:**
 - Op o 1: Dataset fornecido (dados p blicos interessantes)
 - Op o 2: Dados pr prios da rea do aluno
- **Roteiro do projeto:**
 1. Importar e explorar dados (`skimr::skim()`)
 2. Limpar e transformar (`dplyr + tidyr`)
 3. Criar 3-4 visualiza es relevantes
 4. Gerar insights e interpreta es
 5. Compilar em relat rio RMarkdown
- **Trabalho individual ou em duplas**
- Suporte do instrutor e uso estrat gico de IA

6.3.2 GitHub: Versionamento Final (25min)

- **Conceitos essenciais:**
 - `git add . staging`
 - `git commit -m "mensagem" snapshot`
 - `git push` enviar para GitHub
- **Arquivo .gitignore:**
 - Ignorar arquivos grandes, tempor rios, sens veis
 - Exemplo:

```
.Rproj.user
.Rhistory
.RData
*.html
data/raw/* # se dados forem muito grandes
```
- **README.md:**
 - Descri o do projeto
 - Estrutura de arquivos
 - Como reproduzir a an lise
 - Depend ncias (pacotes necess rios)
- **Push do projeto final**

6.3.3 Uso de Claude para Documenta o (15min)

- Gerar README.md profissional
- Revisar c digo para clareza
- Sugerir melhorias na narrativa do relat rio
- Checar reprodutibilidade

6.3.4 Apresenta es R pidas (25min)

- 3-5 minutos por pessoa/dupla
- Principais achados
- Desafios enfrentados
- Feedback coletivo

6.3.5 Encerramento (5min)

- Recursos para continuar aprendendo
 - Comunidades: R-Ladies, Meetups, Posit Community
 - Cursos recomendados: R for Data Science (r4ds.hadley.nz)
 - Certificado de participação
-

7 Recursos Adicionais

7.1 Materiais Complementares

7.1.1 Livros Online (Gratuitos)

- **R for Data Science (2e):** <https://r4ds.hadley.nz/>
- **Advanced R:** <https://adv-r.hadley.nz/>
- **R Graphics Cookbook:** <https://r-graphics.org/>
- **Happy Git with R:** <https://happygitwithr.com/>

7.1.2 Cheat Sheets

- RStudio IDE: <https://posit.co/resources/cheatsheets/>
- dplyr, ggplot2, RMarkdown dispon veis no menu Help Cheat Sheets

7.1.3 Comunidades

- **Posit Community:** <https://community.rstudio.com/>
- **Stack Overflow:** tag [r]
- **R-Ladies:** cap tulos locais e globais
- **LinkedIn/Twitter:** #RStats

7.1.4 Datasets para Pr tica

- **datasets::** (pacote nativo do R)
- **tidytuesday:** <https://github.com/rfordatascience/tidytuesday>
- **Kaggle Datasets:** <https://www.kaggle.com/datasets>
- **Brasil.io:** <https://brasil.io/datasets/> (dados brasileiros)
- **IBGE, DataSUS, INEP:** dados p blicos brasileiros

7.2 Usando IA de Forma tica e Eficiente

7.2.1 Boas Pr ticas

[OK] **Fa a:** - Use IA para explicar conceitos que n o entendeu - Pe a sugest es de otimizao de c digo que voc j escreveu - Use para debugging (mas sempre entenda o erro primeiro) - Solicite revis o de documentao e coment rios - Aprenda com as explica es fornecidas

[X] **Evite:** - Copiar c digo sem entender o que ele faz - Usar IA para fazer TODO o trabalho (voc n o aprende) - Confiar cegamente no c digo gerado (sempre teste!) - Compartilhar dados sens veis com ferramentas de IA

7.2.2 Prompts Eficientes

Ruim: > “Como fa o um gr fico no R?”

Bom: > “Tenho um data.frame com colunas ‘ano’, ‘pais’ e ‘pib’. Como criar um gr fico de linhas no ggplot2 mostrando a evolu o do PIB de cada pa s ao longo dos anos, com cada pa s em uma cor diferente?”

Excelente: > “Estou aprendendo ggplot2 e tenho este código [cole o código]. Ele gera o gráfico, mas as legendas estão sobrepostas e os labels do eixo x estão ilegíveis. Como posso melhorar a legibilidade sem perder informação?”

7.3 Critérios de Avaliação do Projeto Final

7.3.1 Análise de Dados (40%)

- Limpeza e transformação adequadas
- Uso correto de dplyr/tidyr
- Tratamento de valores ausentes

7.3.2 Visualizações (30%)

- Escolha apropriada de gráficos
- Clareza e estética
- Interpretação correta

7.3.3 Reprodutibilidade (20%)

- Código organizado e comentado
- Estrutura de projeto adequada
- README.md completo
- Relatório RMarkdown funcional

7.3.4 GitHub (10%)

- Commits incrementais
 - Mensagens de commit claras
 - .gitignore apropriado
-

7.4 Contato e Suporte Pós-Curso

Instrutor: Vinicius Silva Junqueira

Email: junqueiravinicius@hotmail.com

GitHub do curso: <https://github.com/viniciusjunqueira/curso-r-github-ia>

Lattes: <http://lattes.cnpq.br/4686677580216927>

LinkedIn: www.linkedin.com/in/junqueiravinicius

Horário de dúvidas: Agendar via email

Suporte após o curso: - Dúvidas via Issues no GitHub do curso (resposta em até 48h) - Grupo no Slack/Discord (se aplicável) - Sessão de follow-up opcional após 2 semanas

8 Feedback dos Participantes

Por favor, preencha o formulário de avaliação ao final do curso: [Link para formulário]

Sua opinião é fundamental para melhorarmos continuamente!

**** Última atualização: **** 2025-10-06

Versão: 2.0