

# E se quisermos integrar texto e código ao produzir um documento?

Introdução ao  quarto<sup>®</sup>

ESTAT0090 – Estatística Computacional

Prof. Dr. Sadraque E. F. Lucena

sadraquelucena@academico.ufs.br

# Cenário

Você realizou uma pesquisa que envolve a análise de dados e já produziu um monte de código em **R** ou **Python**. Agora você precisa escrever um relatório ou uma apresentação que contenha os resultados e sua interpretação em um texto explicativo, exibir todo o código, gráficos e tabelas de uma forma fácil de entender e que seja reproduzível. Então você precisa:

- Mostrar o código-fonte da análise sem que o documento vire uma bagunça;
- Gerar um PDF, um site interativo para seu portfólio e uma apresentação em slides;
- Fazer o trabalho parecer profissional com tabelas bem formatadas, figuras numeradas e legendas em um estilo coeso, sem passar horas ajustando detalhes.

Com o Quarto, você escreve seu texto e seu código no mesmo arquivo. O Quarto executa seu código, insere os resultados (gráficos, tabelas, valores) diretamente no texto, e formata tudo em diversos tipos de documentos (HTML, PDF, Word, apresentações, websites, etc.) com apenas um comando. Ele garante que tudo está sincronizado e é totalmente reproduzível, liberando você para focar na sua análise, não na formatação!

# Objetivo da aula

Na aula de hoje aprenderemos a:

- Instalar e configurar o Quarto no RStudio.
- Criar e organizar um novo documento Quarto (`.qmd`) com blocos de código executáveis e texto narrativo.
- Inserir e executar código R diretamente no documento, exibindo tabelas, gráficos e saídas de console de forma integrada.
- Aplicar formatação básica de texto (negrito, itálico, listas) e utilizar opções simples para personalizar a aparência dos resultados do código.
- Compilar o documento Quarto para múltiplos formatos de saída, como HTML, PDF e Word.

# Introdução ao quarto<sup>®</sup>



- O Quarto é um sistema de publicação de código aberto desenvolvido pela Posit para documentos técnicos e científicos em ciência de dados.
- Ele oferece um ambiente onde você pode integrar eficientemente seu código-fonte, os resultados gerados e texto narrativo, garantindo a reprodutibilidade de suas análises.
- Sua versatilidade permite que, a partir de um único arquivo-fonte, você gere uma ampla variedade de formatos de saída, como PDFs, documentos Word, apresentações interativas, páginas da *web*, livros e muito mais.



- Quarto não é um pacote R!
  - Quarto é uma ferramenta de linha de comando que pode ser usada no RStudio (mas também em outros ambientes de desenvolvimento integrado – IDEs, como VS Code e Jupyter).
  - Ele oferece suporte nativo para diversas linguagens de programação, como R, Python, Julia e Observable JavaScript.

# Pré-requisitos

Instalar a versão mais recente de:



- Quarto pode ser baixado em <https://quarto.org/>.
- Uma alternativa é o uso do Posit Cloud (<https://posit.cloud/>), que permite usar o RStudio *on-line*, sem a necessidade de instalações e configurações.

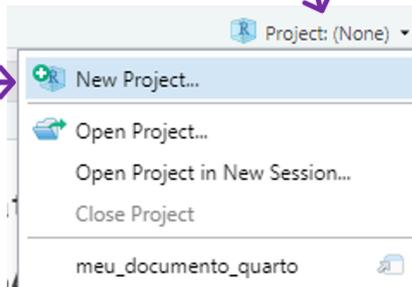


# Começando

Primeiro crie um projeto.

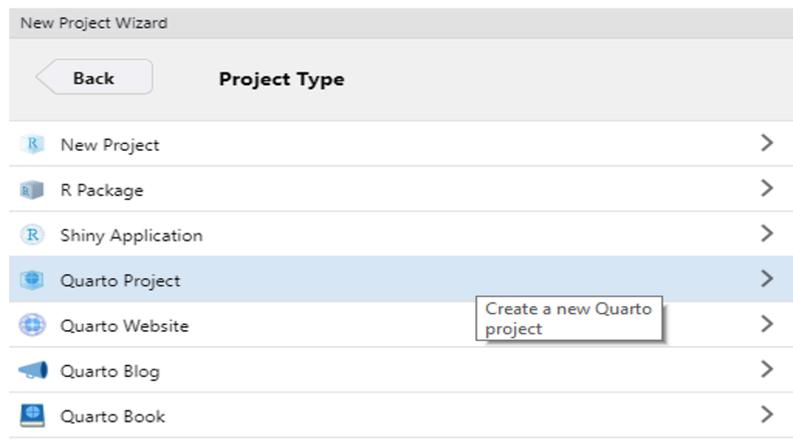
1º No canto superior direito do RStudio, clique em *Project*

2º Clicar aqui

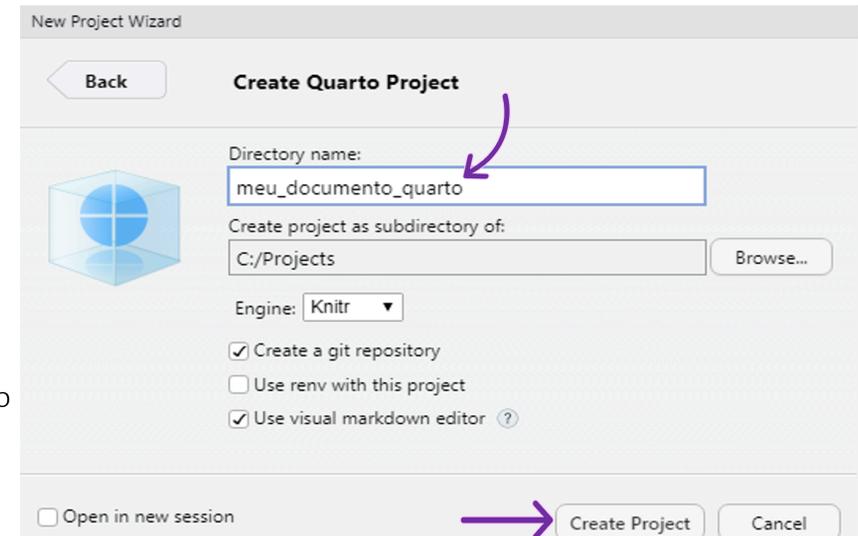


3º Escolha um novo diretório de trabalho (ou associe a algum já existente)

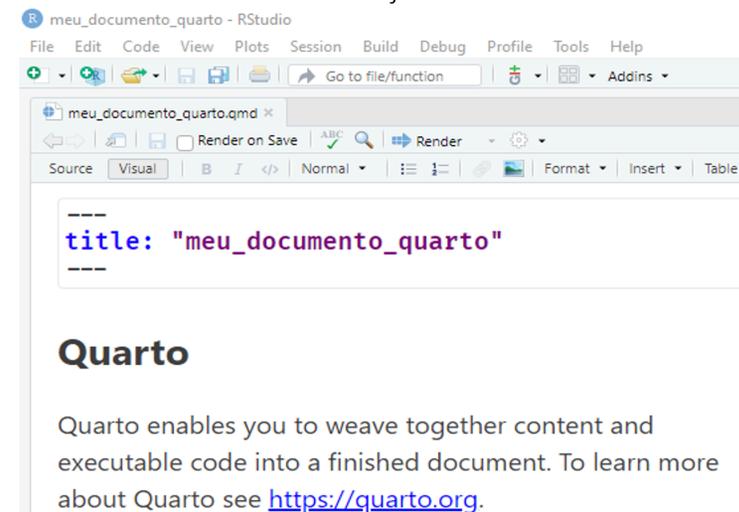
4º Selecione *Quarto Project*



5º Nomeie o diretório e clique em "Create Project"



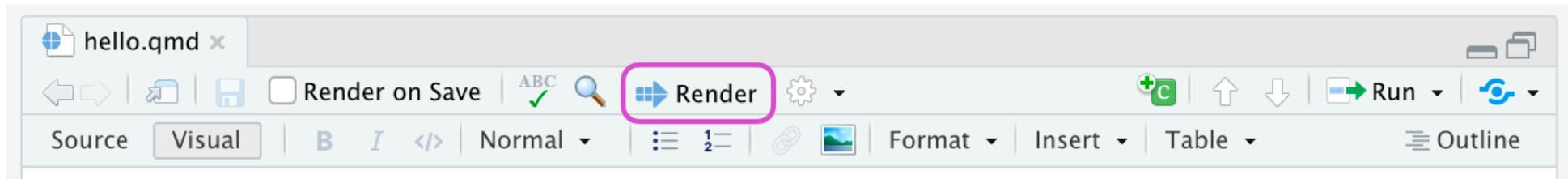
6º Pronto! Projeto criado! :D



Fonte: <https://rladies-sp.org/>

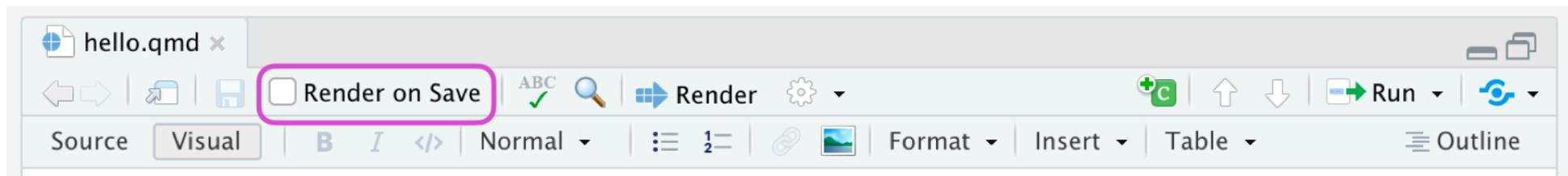
# Começando

- O arquivo editável tem extensão `.qmd`. Para renderizá-lo clique em render ou utilize o atalho do teclado **CTRL + SHIFT + K**.



Fonte: <https://quarto.org/docs/get-started/hello/rstudio.html>

- Você pode ativar a opção **Render on Save** na barra do editor para atualizar automaticamente a visualização ao salvar o documento. A prévia lado a lado funciona para HTML e PDF.



Fonte: <https://quarto.org/docs/get-started/hello/rstudio.html>

# Entendendo a estrutura de um documento em Quarto

Um documento em quarto é composto por três partes:

1. **YAML (Yet Another Markup Language, Mais uma linguagem de marcação):** a seção inicial do documento onde são especificadas configurações como formatação, data, título, autor, e outros detalhes. Essa seção é sempre posicionada no início do documento e é delimitada por três traços (---).
2. **Texto:** utiliza a sintaxe do Markdown como seu principal formato de documento;
3. **Código:** é compatível com várias linguagens, o que permite a inserção de blocos de código em R, Python, Julia e outras.

# YAML

Exemplo de cabeçalho YAML com as configurações iniciais e diretrizes para criação do arquivo:

```
---  
title: "Meu primeiro documento"  
format: html  
lang: pt  
---
```

- A sintaxe básica do YAML usa o par `chave: valor`.
- Ao ser renderizado, o título “Meu primeiro documento” aparecerá no topo do documento com um tamanho de fonte maior que o restante do texto.
- Os outros dois campos YAML indicam que a saída deve ser no formato HTML e que o documento deve estar em português.

# YAML

- Outros campos YAML comumente encontrados nos cabeçalhos dos documentos incluem metadados como
  - `author`
  - `theme`
  - `subtitle`
  - `fontcolor`
  - `date`
  - `fig-width`, etc.
- Todos os campos YAML disponíveis para documentos HTML encontram-se em <https://quarto.org/docs/reference/formats/html.html>.
- Os campos YAML disponíveis variam conforme o formato do documento.
  - Campos YAML para PDF: <https://quarto.org/docs/reference/formats/pdf.html>
  - Campos YAML MS Word: <https://quarto.org/docs/reference/formats/docx.html>

# Blocos de código (*Code chunks*)

- Os blocos de código iniciam com três crases seguidos do nome da linguagem utilizada.
- Em seguida você pode definir configurações ou parâmetros para o bloco de código.
- Exemplo:

```
```${r}  
#| label: load-packages  
#| include: false  
  
library(tidyverse)  
library(palmerpenguins)  
```
```

# Blocos de código (*Code chunks*)

Algumas das principais opções para blocos de código são:

- `#| echo`: controla se o código é exibido no documento.
- `#| eval`: determina se o código é executado e a saída apresentada.
- `#| warning`: controla a exibição de mensagens de aviso geradas pelo código.
- `#| error`: determina se os erros devem ser incluídos na saída.
- `#| include`: controla se o trecho de código e a saída são incluídos no documento renderizado.
- `#| label`: rótulo dado para fazer referência. Uma vez definido um rótulo, você pode referenciá-lo usando `@nome_do_rotulo`.
- `#| fig-cap`: adiciona uma legenda a gráficos gerados pelo código.

# Blocos de código (*Code chunks*)

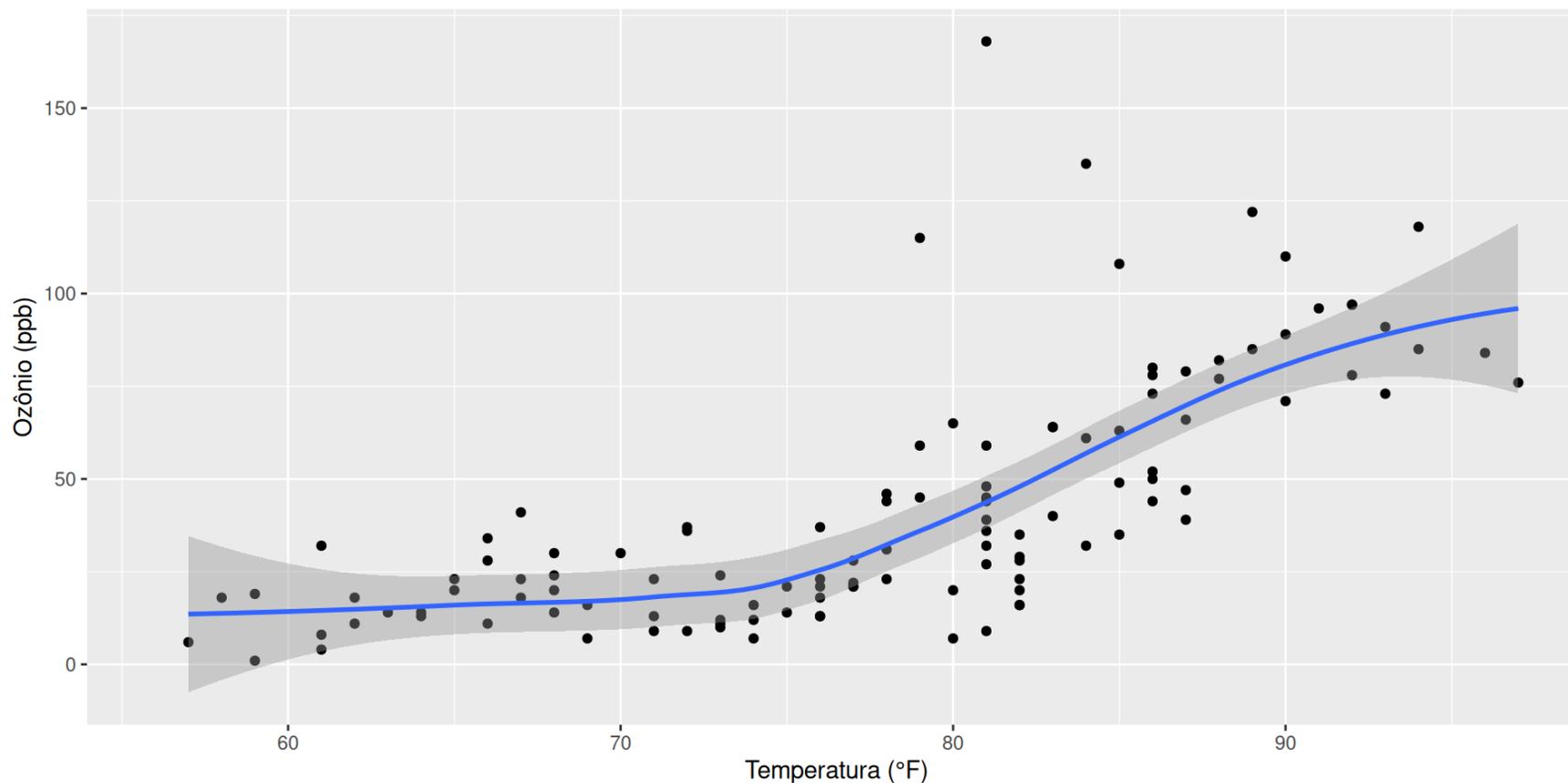
Exemplo:

```
```\r}\n#| echo: true\n#| eval: true\n\nlibrary(ggplot2)\nggplot(airquality, aes(Temp, Ozone)) +\n  geom_point() +\n  geom_smooth(method = "loess") +\n  labs(x = "Temperatura (°F)", y = "Ozônio (ppb)")\n```\n
```

# Blocos de código (*Code chunks*)

Produz:

```
library(ggplot2)
ggplot(airquality, aes(Temp, Ozone)) +
  geom_point() +
  geom_smooth(method = "loess") +
  labs(x = "Temperatura (°F)", y = "Ozônio (ppb)")
```



# Formatação de texto

Algumas opções são

| Sintaxe                                 | Saída                                 |
|-----------------------------------------|---------------------------------------|
| <code>*itálico*</code>                  | <i>itálico</i>                        |
| <code>**negrito**</code>                | <b>negrito</b>                        |
| <code>***negrito itálico***</code>      | <b><i>negrito itálico</i></b>         |
| <code>sobrescrito^2^</code>             | sobrescrito <sup>2</sup>              |
| <code>subscrito~2~</code>               | subscrito <sub>2</sub>                |
| <code>~~riscado~~</code>                | <del>riscado</del>                    |
| <code>`código não interpretável`</code> | <code>código não interpretável</code> |

# Seções

A seção e as subseções são definidas pela quantidade de #. Exemplos:

| Sintaxe               | Saída                   |
|-----------------------|-------------------------|
| # Seção 1             | <b>Seção 1</b>          |
| ## Subseção 2         | <b>Subseção 2</b>       |
| ### Subsubseção 3     | <b>Subsubseção 3</b>    |
| #### Subsubsubseção 4 | <b>Subsubsubseção 4</b> |

# Links e Imagens

| Sintaxe                                             | Saída                                                                                                                                                                                                                                                      |
|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="https://quarto.org">https://quarto.org</a> | <a href="https://quarto.org">https://quarto.org</a>                                                                                                                                                                                                        |
| Quarto                                              | Quarto                                                                                                                                                                                                                                                     |
| ![Legenda](ufs_logotipo.png)                        | <p>The logo of the Universidade Federal de Sergipe (UFS) is displayed. It features the letters 'ufs' in a stylized, blue, outlined font. Below the letters, the text 'UNIVERSIDADE FEDERAL DE SERGIPE' is written in a smaller, blue, sans-serif font.</p> |
| #### Subsubsubseção 4                               | <b>Subsubsubseção 4</b>                                                                                                                                                                                                                                    |

# Listas

```
* Lista sem ordem
+ subitem 1
+ subitem 2
  - subsubitem 1
```

Produz:

- Lista sem ordem
  - subitem 1
  - subitem 2
    - subsubitem 1

```
1. Lista ordenada
2. item 2
  i) subitem 1
    A. subsubitem 1
```

Produz:

1. Lista ordenada
2. item 2
  - i. subitem 1
    - a. subsubitem 1

# Tabelas

| Default | Esquerda | Direita | Centro |
|---------|----------|---------|--------|
| 12      | 12       | 12      | 12     |
| 123     | 123      | 123     | 123    |
| 1       | 1        | 1       | 1      |

Produz:

| Default | Esquerda | Direita | Centro |
|---------|----------|---------|--------|
| 12      | 12       | 12      | 12     |
| 123     | 123      | 123     | 123    |
| 1       | 1        | 1       | 1      |

# Equações

- Funcionam exatamente como no LaTeX. Use  $\$$  para fórmulas e símbolos matemáticos embutidos no texto e  $\$\$$  fórmulas e símbolos matemáticos destacados.
- Exemplo:

Seja  $X \sim N(\mu, \sigma^2)$  uma variável aleatória. Sua função de densidade é dada por

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

em que  $-\infty < x < \infty$ ,  $-\infty < \mu < \infty$  e  $\sigma^2 > 0$ .

Produz:

Seja  $X \sim N(\mu, \sigma^2)$  uma variável aleatória. Sua função de densidade é dada por

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

em que  $-\infty < x < \infty$ ,  $-\infty < \mu < \infty$  e  $\sigma^2 > 0$ .

# Citação

- Existem várias formas de fazer citação em Quarto. Vamos ver a citação em BibLaTeX (arquivos com extensão `.bib`).
- Primeiro você precisa informar o arquivo com as referências no YAML. Por padrão é usado o formato *Chicago Manual of Style* com autor e data, mas você pode especificar uma formatação personalizada usando o CSL (*Citation Style Language*).
- Você pode encontrar arquivos CSL ou saber mais sobre como usar estilos no Projeto CSL. Você pode navegar na lista de mais de 8.500 definições CSL Creative Commons no [repositório central do Projeto CSL](#) ou no [repositório de estilos do Zotero](#).

# Citação

- Exemplo:

```
---  
title: Título  
bibliography: references.bib  
csl: abnt.csl  
---
```

Os tipos de citação mais comuns são:

- `@Wickham2023` produz WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND (2023)
- `[@Wickham2023]` produz (WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND, 2023)
- O arquivo csl com a formatação da ABNT pode ser encontrada [aqui](#).

# Publicação

- Com o documento renderizado para HTML, você pode publicá-lo no RPubS (um serviço gratuito da RStudio para compartilhamento de documentos na *web*) simplesmente clicando no botão de publicar (  Publish) na barra de ferramentas do editor ou na janela de visualização.
- Outras opções possíveis de publicação incluem o RStudio Connect, o ShinyApps, GitHub Pages, Netlify, entre outros.
- O artigo [Publishing HTML](#) oferece mais detalhes sobre as opções de publicação.

Esse conteúdo foi baseado no:

- Livro *R for Data Science, 2nd Edition* de WICKHAM; ÇETINKAYA-RUNDEL; GROLEMUND (2023).
- Tutorial *E aí, vamos falar de Quarto?*, disponível em <https://rladies-sp.org/posts/2023-02-tutorial-quarto/>
- Guia do Quarto disponível em <https://quarto.org/docs/guide/>.

# Outros formatos

- **Apresentações:** Crie apresentações em PowerPoint, Beamer e Revealjs usando a mesma sintaxe que você aprendeu para criar documentos.
- **Sites:** Publique coleções de documentos como um site. Os sites oferecem diversos tipos de navegação e suporte a busca em texto completo.
- **Blogs:** Crie um blog com uma página “sobre”, listagens flexíveis de posts, categorias, feeds RSS e mais de vinte temas disponíveis.
- **Livros:** Produza livros e manuscritos em formatos para impressão (PDF, MS Word) e online (HTML, ePub).
- **Interatividade:** Inclua componentes interativos para ajudar leitoras e leitores a explorar mais profundamente os conceitos e os dados que você está apresentando.

# Fim

# Referências

WICKHAM, Hadley; ÇETINKAYA-RUNDEL, Mine; GROLEMUND, Garrett. **R for data science: Import, tidy, transform, visualize, and model data**. Sebastopol, CA: O'Reilly, 2023.