

R Markdown e Quarto

Gilberto Pereira Sassi

Departamento de Estatística
Instituto de Matemática e Estatística

Durante o curso

- Usaremos nas aulas: `posit.cloud`.
 - Recomendamos instalar e usar R com versão pelo menos 4.1: `cran.r-project.org`.
 - usaremos o *framework* `tidyverse`:
 - Instalação: `install.packages("tidyverse")`
-

Na sua casa

- **IDE** recomendadas: *RStudio* e *VSCode*.
 - Caso você queira usar o *VSCode*, instale a extensão da linguagem R: `REditorSupport`.
- Outras linguagens interessantes: `python` e `julia`.
 - `python`: linguagem interpretada de propósito geral, contemporânea do R, simples e fácil de aprender.
 - `julia`: linguagem interpretada para análise de dados, lançada em 2012, promete simplicidade e velocidade.

Este curso é apenas o começo!

Você vai ter que estudar sozinho para avançar mais...

Para usar o pacote `rmarkdown`, você precisa ter:

- conhecimento básico da linguagem R
- conhecimento básico da linguagem `latex`
- conhecimento básico da linguagem `markdown`

R

- Zen do R
- R for Datascience
- ecoR

\LaTeX

- Learn \LaTeX in 30 minutes
- Detexify
- Learn \LaTeX with Wikibooks

markdown

- Tutorial de markdown da Microsoft
- Tutorial de markdown da Mozilla
- markdown Basics

rakdown

- **rakdown**: The Definitive Guide
- **bookdown**: Authoring Books and Technical Documents with **rakdown**
- **blogdown**: Creating Websites with **rakdown**

Pacotes da linguagem R deste curso

- rmarkdown
- blogdown
- bookdown
- readxl
- writexl
- janitor
- patchwork
- prettydoc
- glue
- ggthemes
- gt
- rtables
- tidyverse

A linguagem R

A precursora da linguagem R: S.

- R é uma linguagem derivada da S.
- S foi desenvolvido em `fortran` por **John Chambers** em 1976 no **Bell Labs**.
- S foi desenvolvida para realizar análise estatística de dados.
- Filosofia do S: permitir que usuários possam analisar dados usando estatística com pouco conhecimento de programação.

História da linguagem R

- Em 1991, **Ross Ihaka** e **Robert Gentleman** criaram o R na Nova Zelândia.
- Em 1996, **Ross** e **Robert** liberam o R sob a licença “GNU General License”, o que tornou o R um software livre.
- Em 1997, The Core Group é criado para melhorar e controlar o código fonte do R.

markdown

- Criado em 2004 por **John Gruber**.
- Criado inicialmente para textos para internet.
- **Ideia:** fácil de escrever, fácil de ler e entender o código, e permitir edição em forma de prosa. Foco no conteúdo e não nos detalhes da linguagem.
- markdown foi inspirada pela formatação permitida ao escrever e-mails.
- markdown é portátil.
- Não depende de versões como Microsoft Word.
- Uso amplamente disseminado, com versões adotadas em aplicativos como: WhatsApp, Notion, GitHub, Stack Overflow, entre outros.

rmarkdown and quarto usam pandoc para converter código markdown para os formatos HTML, pdf e docx.

Seções (e subseções) são partes que dividem um texto de acordo com conteúdo afins.

Para mais detalhes, consulte [Seções e subseções de acordo com NBR 6024/2012](#).

1. SEÇÃO PRIMÁRIA

1.1 SEÇÃO SECUNDÁRIA

1.1.1 Seção Terciária

1.1.1.1 Seção quaternária

1.1.1.1.1 Seção quinária

3. MATERIAIS E MÉTODOS

3.1 ANÁLISE DESCRITIVA:
características dos grupos de
estudo

3.1.1 Brasil

3.1.1.1 Grandes Regiões

3.1.1.1.1 Norte

3.1.1.1.2 Nordeste

3.1.1.1.3 Sul

3.1.1.1.4 Sudeste

3.1.1.1.5 Centro-Oeste

3. MATERIAIS E MÉTODOS

Texto, texto, texto, texto,
texto, texto, texto, texto,
texto, texto ...

3.1 ANÁLISE DESCRITIVA:

características dos grupos de
estudo

Texto, texto, texto, texto,
texto, texto, texto, texto,
texto, texto ...

3.1.1 Brasil

Texto, texto, texto, texto,
texto, texto, texto, texto,
texto, texto ...

... continua.

Podemos definir seções e subseções com #.

O caractere # precisa estar na primeira coluna da linha.

É necessário incluir um único espaço depois de #.

código markdown	código HTML	código L ^A T _E X
# texto	<h1>texto</h1>	\section{texto}
## texto	<h2>texto</h2>	\subsection{texto}
### texto	<h3>texto</h3>	\subsubsection{texto}
#### texto	<h4>texto</h4>	\paragraph{texto}
##### texto	<h5>texto</h5>	\subparagraph{texto}
##### texto	<h6>texto</h6>	

Parágrafos

Para criar parágrafos, separa blocos de linhas com um (ou mais) linhas em branco.

código markdown	código HTML	código \LaTeX
Primeira linha.	<code><p>Primeira linha.</p></code>	Primeira linha.
Segunda linha.	<code><p>Segunda linha.</p></code>	Segunda linha.

Não inclua tabs ou espaços na primeira linha de parágrafos.

Formatação de texto

Descrição	código markdown	código HTML	código L ^A T _E X	Resultado
Itálico	<i>*Itálico*</i>	<code>Itálico</code>	<code>\textit{Itálico}</code>	<i>Itálico</i>
Negrito	**Negrito**	<code>Negrito</code>	<code>\textbf{Negrito}</code>	Negrito
Tachado	~~Tachado~~	<code><s>Tachado</s></code>		Tachado
Sobrescrito	x^2	<code>x<sup>2</sup></code>		x^2
Subscrito	t_0	<code>t<sub>0</sub></code>		t_0

Podemos ter um texto em negrito e em itálico:

******Negrito e Itálico******

Bloco de citação

Na primeira linha do parágrafo, inclua >.

Exemplo de código

```
> Isto é uma citação.
```

Resultado

Isto é uma citação.

Bloco de citação com múltiplos parágrafos

Adicione > em cada parágrafo e nos espaços em branco entre os parágrafos.

Exemplo de código

```
> Primeira linha do bloco de citação.  
>  
> Segunda linha do bloco de citação.
```

Resultado

Primeira linha do bloco de citação.
Segunda linha do bloco de citação.

Bloco de citação pode ter todos os outros elementos markdown.

Exemplo

```
> ### Um lindo bloco de citação  
> Primeiro parágrafo tem negrito.  
>  
> Segundo parágrafo tem Itálico.  
>  
> Terceiro Párrafo tem Negrito e Itálico.
```

Resultado

*Primeiro parágrafo tem **negrito**.*

*Segundo parágrafo tem *Itálico*.*

*Terceiro Párrafo tem ***Negrito e Itálico***.*

Listas ordenadas

As listas ordenadas devem começar com o número 1 (ou i.).

código markdown	código HTML	código L ^A T _E X
1. Primeiro item	<code></code>	<code>\begin{enumerate}</code>
8. Segundo item	<code>Primeiro</code> <code>item</code>	<code>\item Primeiro</code> <code>item</code>
1. Terceiro item	<code>Segundo</code> <code>item</code> <code>Terceiro</code> <code>item</code> <code></code>	<code>\item Segundo item</code> <code>\item Terceiro</code> <code>item</code> <code>\end{enumerate}</code>

Resultado

- 1 Primeiro item
- 2 Segundo item
- 3 Terceiro item

Listas não ordenadas

As listas não ordenadas começam com: -, * ou +.

código markdown	código HTML	código L ^A T _E X
+ Primeiro item		\begin{itemize}
+ Segundo item	Primeiro item	\item Primeiro item
+ Terceiro item	Segundo item	\item Segundo item
	Terceiro item	\item Terceiro item
		\end{itemize}

Resultado

- Primeiro item
- Segundo item
- Terceiro item

Listas aninhadas

Indente as listas que estão dentro de outras listas com dois espaços.

código markdown	código HTML	código L ^A T _E X
+ Item 1 + Item 1 interno + Item 2 interno + Item 2	<pre> Item 1 Item 1 interno Item 2 interno Item 2 </pre>	<pre>\begin{itemize} \item Item 1 \begin{itemize} \item Item 1 interno \end{itemize} \item Item 2 interno \end{itemize}</pre>

Resultado

- Item 1
 - Item 1
 - Item 2
- Item 2

Listas aninhadas

Indente as listas que estão dentro de outras listas com dois espaços.

código markdown	código HTML	código L ^A T _E X
+ Item 1 1. Item 1 interno 1. Item 2 interno	<ul style="list-style-type: none">Item 1 Item 1 interno Item 2 interno Item 2 	<pre>\begin{itemize} \item Item 1 \begin{enumerate} \item Item 1 interno \end{enumerate} \item Item 2 interno \end{itemize}</pre>

Resultado

- Item 1
 - ① Item 1
 - ② Item 2
- Item 2

Código fonte (sem inclusão do resultado)

Para inclusão de código *inline* (dentro de uma frase): ``print("olá mundo!")``.

Exemplo

Um texto com código ``print(1 + 2)``.

Resultado

Um texto com código `print(1 + 2)`.

Para inclusão de código *em bloco*, use `````.

Exemplo

Podemos substituir `r` por: `python`, `html`, `julia` entre outros.

```
```r
print("Olá mundo!")
print(1 + 2)
```
```

Resultado

```
print("Olá mundo")
print(1 + 2)
```

Para ver a lista de linguagens compatíveis consulte: [linguagens compatíveis com markdown](#).

Tabelas

markdown usa tabelas conhecidas como pipe table que tem a seguinte sintaxe:

- As colunas são separadas por |.
- A primeira linha contém cabeçalho das colunas.
- A segunda linha contém o alinhamento:
 - ----- (default) valores serão alinhados à esquerda
 - -----: valores serão alinhados à direita
 - :----- valores serão alinhados à esquerda
 - :----: valores ficarão centralizados
- A partir da terceira linha, incluímos as informações
- Incluímos a legenda da tabela depois de incluímos todas as linhas

As células de uma *pipe table* **não** podem conter:

- parágrafos
- listas
- valor em múltiplas linhas

Gerador de tabelas markdown: tablesgenerator.com/markdown_tables.

Exemplo

```
default	alinhamento à esquerda	alinhamento à direita	centralizado
12	12	12	12
123	123	123	123
1	1	1	1
```

: Legenda da tabela.

Resultado

Tabela 8: Legenda da tabela.

| default | alinhamento
à esquerda | alinhamento à
direita | centralizado |
|---------|---------------------------|--------------------------|--------------|
| 12 | 12 | 12 | 12 |
| 123 | 123 | 123 | 123 |
| 1 | 1 | 1 | 1 |

Links e imagens

A sintaxe básica para links é `[texto do link](endereço do link)`, onde:

- `texto do link` é um texto descritivo para o link
- `endereço do link` é o endereço para redirecionamento

Exemplo:

```
[Google](https://www.google.com.br)
```

Resultado:

Google

Se *texto do link* é igual a *endereço do link*, você pode usar:

```
<https://www.google.com.br>
```

Links e imagens

A sintaxe básica para links é `![texto da imagem] (endereço da imagem)`, onde:

- texto da imagem é um texto descritivo da imagem
- endereço da imagem é *filename* da imagem

A imagem precisa estar dentro do mesmo diretório que o arquivo `.Rmd`.

Exemplo

```
![Logo da linguagem R] (figuras/r.png)
```

Resultado



Exemplo

- Vamos analisar um documento simples usando `markdown`.
- Por hora, ignore as primeiras linhas delimitadas ---.

Exemplo com sintaxe

Exercício

Use loremipsum.io para criar um texto `html`.

Inclua:

- o texto precisa ter 6 parágrafos
- uma citação de sua preferência
- texto negrito, texto itálico, texto tachado, e texto tachado e negrito
- inclua uma lista de itens que você gosta
- inclua a seguinte tabela com as colunas centralizadas
- inclua um link para o website do nosso curso rmarkdown.simple.ink
- inclua o logo rmarkdown ao final do arquivo (o arquivo na pasta `figuras`)

| Cidade | População |
|----------------|------------|
| São Paulo | 11.451.245 |
| Rio de Janeiro | 6.211.423 |
| Brasília | 2.817.068 |
| Fortaleza | 2.428.678 |
| Salvador | 2.418.005 |

Equações usando \LaTeX

`markdown` e `quarto` usam \LaTeX para composição tipográfica de equações matemáticas.

Existem dois tipos de inclusão de equações matemáticas:

- *inline*: equação é parte de um parágrafo
 - usamos `$` para equações em modo *inline*
- *display*: equação em um linha separada com texto centralizado
 - usamos `$$` para equações em modo *display*

Expressões matemáticas usando \LaTeX

Equações em modo *inline*

Exemplo:

A equação $e^{i\pi} + 1 = 0$ foi proposta por Euler.

Resultado:

A equação $e^{i\pi} + 1 = 0$ foi proposta por Euler.

Expressões matemáticas usando \LaTeX

Equações em modo *display*

Exemplo:

A seguinte equação foi proposta por Euler:

$$e^{i\pi} + 1 = 0$$

Resultado:

A seguinte equação foi proposta por Euler:

$$e^{i\pi} + 1 = 0$$

Expressões matemáticas usando \LaTeX

Símbolos e funções importantes

| Descrição | Código \LaTeX | Resultado |
|--------------------------|------------------------|-------------|
| Letra alpha | <code>\alpha</code> | α |
| Letra epsilon | <code>\epsilon</code> | ϵ |
| União de conjuntos | <code>\cup</code> | \cup |
| Intersecção de conjuntos | <code>\cap</code> | \cap |
| Menor (desigualdade) | <code><</code> | $<$ |
| Espaço simples | <code>b\ a</code> | $b\ a$ |
| Espaço duplo | <code>b\quad a</code> | $b\quad a$ |
| Espaço triplo | <code>b\qquad a</code> | $b\qquad a$ |
| Maior (desigualdade) | <code>></code> | $>$ |
| Infinito | <code>\infty</code> | ∞ |
| Logaritmo | <code>\log</code> | \log |
| Multiplicação | <code>\cdot</code> | \cdot |
| Contém | <code>\subset</code> | \subset |
| Contido | <code>\supset</code> | \supset |
| Integral | <code>\int</code> | \int |
| Somatório | <code>\sum</code> | \sum |
| Produtório | <code>\prod</code> | \prod |
| Limite | <code>\lim</code> | \lim |

Lista exaustiva de símbolos e funções.

Detexfy - aplicativo para descobrir símbolos matemáticos.

Expressões matemáticas usando \LaTeX

Sobrescrito, subscrito e fração

Sobrescrito e subscrito

Muito comum em expressões matemáticas envolvendo expoentes, índices, e em alguns proponentes especiais.

- Sobrescrito e limite superior em \int , \prod , \sum , \cap e \cup : \wedge
- Subscrito e limite inferior em \int , \prod , \sum , \cap e \cup : $_$

Se o sobrescrito e subscrito tiver mais de um caracter, envolva a expressão em chaves $\{\}$.

Fração

| Código \LaTeX | Resultado |
|-----------------------------|---------------|
| $\text{\code{\frac{a}{b}}}$ | $\frac{a}{b}$ |

Expressões matemáticas usando \LaTeX

Sobrescrito e subscrito

| Código \LaTeX | Resultado |
|--|--|
| $a^{\{n_j\}}$ | a^{n_j} |
| $a_{\{n^i\}}$ | a_{n^i} |
| $\int_{-\infty}^{\infty} \frac{1}{1+x^2} dx$ | $\int_{-\infty}^{\infty} \frac{1}{1+x^2} dx$ |
| $\sum_{i=1}^{\infty} \frac{1}{i^2}$ | $\sum_{i=1}^{\infty} \frac{1}{i^2}$ |
| $\prod_{i=1}^n i$ | $\prod_{i=1}^n i$ |
| $\cup_{i=1}^{n+12} A_i$ | $\cup_{i=1}^{n+12} A_i$ |
| $\cap_{i=-\infty}^0 B_i$ | $\cap_{i=-\infty}^0 B_i$ |

Expressões matemáticas usando \LaTeX

Parênteses, colchetes e chaves

| Descrição | Código \LaTeX | Resultado |
|-----------------------|-------------------------|-------------------------|
| Parênteses | $(x + y)$ | $(x + y)$ |
| Colchetes | $[x + y]$ | $[x + y]$ |
| Chaves | $\{x + y\}$ | $\{x + y\}$ |
| Produto interno | $\langle x, y \rangle$ | $\langle x, y \rangle$ |
| Valor absoluto | $ x + y $ | $ x + y $ |
| Norma | $\ x + y\ $ | $\ x + y\ $ |
| Arrendonda para baixo | $\lfloor x + y \rfloor$ | $\lfloor x + y \rfloor$ |
| Arrendonda para cima | $\lceil x + y \rceil$ | $\lceil x + y \rceil$ |

Para aumentar o tamanho de parênteses, colchetes e outros: `\big`, `\Big`, `\bigg`, e `\Bigg`.

Para aumentar o tamanho de parênteses, colchetes e outros: `\left` e `\right` (na mesma linha).

Para detalhes sobre ajuste no tamanho de parênteses, colchetes e outros, consulte [parênteses, colchetes e outros](#).

Use `split` para quebrar uma equação em várias linhas, e use `&` para alinhamento.

Expressões matemáticas usando L^AT_EX

Parênteses, colchetes e chaves

Exemplo

```
$$  
\begin{split}  
a &= \sum_{i=-\infty}^{\infty} \left\{ a_i + b_i \right. \\ &= \left. c_i + \int_a^b x \cdot i \, dx \right\}  
\end{split}  
$$
```

Resultado

$$\begin{aligned} a &= \sum_{i=-\infty}^{\infty} \left\{ a_i + b_i \right. \\ &= \left. c_i + \int_a^b x \cdot i \, dx \right\} \end{aligned}$$

Expressões matemáticas usando \LaTeX

Exercício

Use a plataforma www.texrendr.com para codificar em \LaTeX as seguintes equações matemáticas:

- $\sum_{i=0}^{\infty} \frac{1}{2^i} = 2$
- $\int_{-\infty}^{\infty} \exp\left(-\frac{x^2}{2}\right) dx = \sqrt{2\pi}$
- $\prod_{i=1}^n \frac{1}{2} = \frac{1}{2^n}$
- $[1, 2] = 1$
- $\lceil 1, 2 \rceil = 2$
- $\|(1, 2)\|^2 = 5$
-

$$\begin{aligned} \sum_{i=1}^n \alpha^i &= 1 + \alpha + \cdots + \alpha^n \\ &= \frac{1 - \alpha^{n+1}}{1 - \alpha} \end{aligned}$$

Pacote rmarkdown

Documentos com extensão `.Rmd` ou `.rmd` permitem combinar:

- código fonte (code tag in html e verbatim em \LaTeX)
- resutado de computações
- texto simples em prosa

Documentos com extensão `.Rmd` ou `.rmd` são renderizados (processados) pelo pacote `rmarkdown`.

Gerador de texto aleatório para este curso: loremipsum.io.

Cheatsheet do pacote `rmarkdown`: [cheatsheet do pacote rmarkdown](#).

`rmarkdown` envia o documento `.Rmd` para o pacote `knitr` que converte o documento para `markdown`, e em seguida `pandoc` converte este arquivo `.md` para formato adequado.



- 1 Para criar um arquivo .Rmd no IDE Rstudio:
file > New File > R Markdown.
- 2 Em seguida, escolha a opção adequada para o seu texto, incluindo:
 - a. *document* (documentos em prosa).
 - b. *presentation* (apresentação).
 - c. *template* (documentos usando *templates*).

Estrutura básica de documentos .Rmd

```
---  
title: "Um belo título para o documento"  
date: 01-12-2023  
output: html_document  
---
```

```
```{r}  
#| label: setup
#| include: false
#| echo: false

library(tidyverse)
```

```

Uma seção

Eu tenho 5 irmões.

Gráfico simples usando o pacote `ggplot2` do *framework* `tidyverse`.
```

```
```{r}  
#| echo: false  
  
mtcars |>  
  ggplot(aes(x = cyl)) +  
  geom_bar()
```

```
---  
## Uma subseção  
  
Algun texto sobre alguma coisa muito importante.
```



- Criada em 2001.
- YAML (Yet Another Markup Language) é uma linguagem desenvolvida para armazenar dados.
- Fácil de ler e escrever.
- YAML é delimitada por `---` e sempre está nas primeiras linhas do documento `.Rmd`.
- YAML contrala a formatação do documento `.Rmd`.
- YAML usa dupla *chave valor*: `chave: valor`:
 - `valor` não pode ter espaço em branco. Se `valor` tiver espaço em branco, use aspas or `|`.
 - `chave`: nome do campo

```
title: "Um título que faz sentido"
```

ou

```
title: |  
  Um título que faz sentido
```



valor pode ter subcampos, e usamos indentação para neste subcampo

```
output:
  html_document:
    highlight: "haddock"
    includes:
      in_header: header.html
      before_body: before_body.html
```

Podemos usar código R, prefixando o campo com `!r`.

```
date: !r lubridate::today()
```

Campos comuns em YAML: `title`, `date` e `output`.

Citações e bibliografia

- rmarkdown usa `bibtex` para incluir referências e citações.
- **Google Scholar** e maioria das revistas científicas incluem citações `bibtex`.
- Para especificar um formato de bibliografia, use CSL (*Citation Style Language*).

```
bibliography: refs.bib  
csl: apa.csl
```

No texto, use @ + identificador da referência no arquivo .bib.

- Citação direta no texto: @wickham2023r.
- Citação entre parênteses: [@wickham2023r].
- Cite apenas o ano entre parênteses: [-@wickham2023r]
- Citação entre parênteses com comentário: [veja @wickham2023r para mais detalhes].
- Separe múltiplas citações com ;: [@wickham2023r; @wickham2019advanced; @xie2016bookdown]

Parâmetros

- Valores que passamos para os documentos.
- Útil para produzir documentos em série, onde apenas alguns valores são modificados.
- Valores ficam disponível na lista `params`.

```
params:  
  nome: "Gilberto Pereira Sassi"  
  idade: 22
```

No texto, use ``r params$nome`` e ``r params$idade``.

Geralmente `params` é usado junto com a função `render` do pacote `rmarkdown`.

Exemplo

```
---  
title: "Seu título vem aqui"  
date: 31/12/2023  
output: html_document  
params:  
  nome: "Nome da pessoa"  
  vinculo: professor  
---  
  
# Primeira seção
```

Texto da primeira seção com o nome `r params\$nome` e Vínculo `r params\$vinculo` .

Para produzir documento, use o seguinte código R

```
render(  
  "filename.Rmd",  
  render_options = list(  
    params = list(nome = "Um nome", data = "01/01/1900")  
  )  
)
```



`rmarkdown`

`ioslides_presentation`

Apresentação em formato .html.

Para criar, especifique output: `ioslides_presentation` no cabeçalho YAML.

```
---  
title: "Um lindo slide"  
author: "Fulano de Tal"  
date: 01/01/1900  
output: ioslides_presentation  
---
```

Neste caso, você pode usar código html no texto.

rmarkdown ioslides_presentation

Para adicionar um novo slide use ##:

```
## Um novo slide
```

Conteúdo do slide

Você pode adicionar um subtítulo do slide com |:

```
## Título do slide | Subtítulo do slide
```

Conteúdo do slide

Se o slide estiver muito grande, você pode usar --- para quebrar o conteúdo:

```
## Título do slide | Subtítulo do slide
```

Conteúdo do slide

```
---
```

Essa parte vai para um novo slide

O ioslides tem os seguintes modos de exibição:

- f: modo tela cheia
- w: modo tela ampla
- o: modo *overview*
- h: modo de destaque do código
- p: modo de apresentação para o palestrante
- esc: volta ao modo normal de apresentação

Algumas configurações disponíveis no cabeçalho YAML:

- incremental: itens das listas são apresentadas um por um
 - Campo booleano (true - os itens das listas são apresentados um por um)
 - Valor default: false
- widescreen: apresentação em modo de tela ampla
 - Campo booleano (true - a apresentação fica em tela ampla)
 - Valor default: apresentação em modo de tela normal
- smaller: tamanho da fonte menor
 - Campo booleano: (true - tamanho da fonte fica um pouco menor)
 - Valor default: tamanho normal (12pt)
- transition: velocidade de transição entre os slides
 - Valores possíveis: default, slower, faster, ou número real que indica os segundos gastos na transição de slides:
 - Valor default: default
- logo: adiciona a logo ao slide
 - Nome do arquivo. Por exemplo, logo.png

Podemos incluir código .html para customização adicional da apresentação com includes:

- in_header: inclusão de código entre <header> e </header>
- before_body: inclusão de código imediatamente depois <body>
- after_body: inclusão de código imediatamente antes de </body>

output:

```
pdf_document:
```

```
  includes:
```

```
    in_header: in_header.html
```

```
    before_body: before_body.html
```

```
    after_body: after_body.html
```

Podemos fazer algumas configurações específicas para um slide.

Implementação da opção `incremental`: `true` para um único slide.

```
## Título do slide {.build}
```

Implementação da opção `smaller`: `true` para um único slide.

```
## Título do slide {.build .smaller}
```

Para destacar uma parte do código dentro de um *chunk* usamos `## ` e `## `.

Exemplo:

```
```{r}
```

```
library(tidyverse)
```

```

```

```
ggplot(airquality, aes(Ozone)) + geom_histogram(bins = 8)
```

```

```

```
summarise(
```

```
 media_ozone = mean(Ozone), dp_ozone = sd(Ozone),
```

```
 q1 = quantile(Ozone), mediana = median(Ozone),
```

```
 q3 = quantile(Ozone), cv_ozone = dp_ozone / media_ozone
```

```
)
```

```
```
```



Podemos fazer algumas configurações específicas para um slide.

Implementação da opção incremental: true para um único slide.

```
## Título do slide {.build}
```

Implementação da opção smaller: true para um único slide.

```
## Título do slide {.build .smaller}
```

Para destacar uma parte do código dentro de um *chunk* usamos `## ` e `## `.

Exemplo:

```
```{r}
```

```
library(tidyverse)
```

```

```

```
ggplot(airquality, aes(Ozone)) + geom_histogram(bins = 8)
```

```

```

```
summarise(
```

```
 media_ozone = mean(Ozone), dp_ozone = sd(Ozone),
```

```
 q1 = quantile(Ozone), mediana = median(Ozone),
```

```
 q3 = quantile(Ozone), cv_ozone = dp_ozone / media_ozone
```

```
)
```

```
```
```



Centralização do conteúdo do slide use `<div class="centered">` e `</div>`.

```
<div class="centered">  
  Texto centralizado no slide.  
</div>
```

Conteúdo em duas colunas: `<div class="columns">` e `</div>`.

```
<div class="columns">  
  
  
+ Primeiro item  
+ Segundo item  
+ Terceiro item  
</div>
```

Colorindo texto: `<div class="cor"> e </div>`.

Onde cor é uma das opções: red, blue, green, yellow, and gray.

Exemplo:

Isto é muito importante: `<div class="red2">coisa importante</div>`.

Notas para apresentador (apenas você vai ver) em reuniões on-line: `<div class="notes"> e </div>`.

```
<div class="notes">
```

Texto que apenas o apresentador vai ver!

```
</div>
```

Acrescente `?presentme=true` no endereço do arquivo `.html` para ativar o as notas do apresentador.

rmarkdown ioslides_presentation

Use o seguinte comando dentro da linguagem R (no console):
`pagedown::chrome_print('arquivo.Rmd')`.

Necessário ter o pacote pagedown instalado.

Exemplo:

```
pagedown::chrome_print("slides/apresentacao.Rmd")
```

Você pode carregar esta apresentação nas plataformas netlify.com e github.com.

Netlify

- 1 Faça login em netlify.com.
- 2 Clique em *sites* na barra lateral.
- 3 Arraste o arquivo `.html` até o quadrado em destaque como apresentado na figura abaixo.

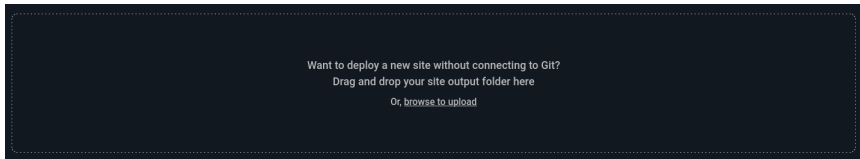


Figura 2: Colocando sua página on-line usando netlify.

GitHub Pages

- ① Faça login em github.com.
- ② Crie um repositório público chamado `<username>.github.io`.
 - Se o *username* é fulano-tal, você criará o repositório `fulano-tal.github.io`.
- ① Adicione o arquivo `arquivo.html` neste repositório.
- ② Depois de alguns minutos ~~poucos~~, o arquivo está disponível em `https://<username>.github.io/arquivo.html`.
- Se o *username* é fulano-tal e o arquivo tem nome `arquivo.html`, o arquivo está on-line em `https://fulano-tal.github.io/arquivo.html`.

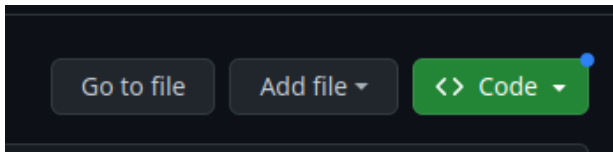


Figura 3: Colocando sua página on-line usando GitHub Pages.

Exemplo

Vamos analisar um exemplo!

exemplo-6

Crie um documento chamado `apresentacao.html`, e inclua

- `title` - inclua o seguinte título: **Apresentação ioslides**.
- `date` - data de nascimento.
- `author` - inclua o seu nome.
- logo do pacote `rmarkdown`.
- garanta que as figuras tenham legendas.
- três slides com subtítulo.
- centralize o texto de um dos slides.
- coloque o texto em duas colunas em um dos slides.
- deixe alguma parte do texto azul.
- notas para o apresentador.
- inclua um slide com um chunk com o seguinte código

```
library(ggthemes)
library(tidyverse)
ggplot(mtcars) +
  geom_histogram(aes(mpg), bins = 7, fill = "blue") +
  theme_calc()
```

Use loremipsum.io para gerar o texto dos slides.

Gere o arquivo .pdf da apresentação *ioslides*.

`rmarkdown`

`beamer_presentation`

Apresentação em formato .pdf.

Para criar, especifique output: beamer_presentation no cabeçalho YAML.

```
---  
title: "Apresentação beamer"  
author: "Gilberto P. Sassi"  
date: 01/01/1900  
output: beamer_presentation  
---
```

Neste caso, você pode usar \LaTeX no texto.

Algumas configurações disponíveis no cabeçalho YAML:

- `theme`: estilo dos slides
 - Veja as opções em [Matrix de temas Beamer](#)
- `colortheme`: coloração dos slides
 - Veja as opções em [Matrix de temas Beamer](#)
- `fonttheme`: fonte da apresentação
 - Opções: `default`, `professionalfonts`, `serif`, `structurebold`, `structureitalicserif`, `structuresmallcapsserif`
 - Veja as fontes em [Fontes disponíveis para Beamer](#)
- `highlight`: formatação do código incluído no texto
 - Valores possíveis: `default`, `tango`, `pygments`, `kate`, `monochrome`, `espresso`, `zenburn`, `haddock`, `breezedark`, `arrow`, e `rstudio`
 - Valor default: `default`
- `fig_caption`: inclusão de legenda na figura
 - Campo booleano (apenas `true` e `false`)
 - Valor default: `true`

- `lang`: especificação da linguagem do documento
 - para português brasileiro use `pt-br`
- `fontsize`: tamanho de fonte
 - Valores possíveis: 10pt, 11pt, 12pt, e outros
 - Valor default: 12pt
- `linkcolor`: cor para links internos dentro do documento
 - Valores possíveis: consulte `pacote xcolor`
- `urlcolor`: cor para link externo dentro do documento
 - Valores possíveis: consulte `pacote xcolor`
- `citecolor`: cor para citações dentro do texto
 - Valores possíveis: consulte `pacote xcolor`
- `citation_package`: processamento das citações dentro do documento
 - Valores possíveis: `pandoc-citeproc`, `natbib`, e `biblatex`
 - Valor default: `pandoc-citeproc`
- `keep_tex`: mantenha código fonte \LaTeX ?
 - Campo booleano (apenas `true` e `false`. Se `true`, mantenha o código fonte \LaTeX)
 - Valor default: `false`

Podemos incluir código \LaTeX para customização adicional do documento com `includes`:

- `in_header`: inclusão de código no preâmbulo (entre `\documentclass{article}` e `\begin{document}`)
- `before_body`: inclusão de código imediatamente depois `\begin{document}`
- `after_body`: inclusão de código imediatamente antes de `\end{document}`

output:

`pdf_document:`

`includes:`

`in_header: preambulo.tex`

`before_body: prefixo.tex`

`after_body: sufixo.tex`

Vamos analisar um exemplo!

exemplo-7

Exercício

Crie um documento chamado `apresentacao.pdf`, e inclua

- `title` - inclua o seguinte título: “Apresentação em pdf”.
- `date` - data de nascimento.
- `author` - inclua seu nome.
- três slides com texto dummy.
- mude o tamanho da fonte para 10pt
- especifique o idioma para português do Brasil (pt-br)
- inclua o seguinte código entre `\documentclass` e `\begin{document}`

```
\usepackage{xcolor}  
\usepackage{tikz}  
\usepackage{bm}
```

- inclua o código imediatamente depois de `\begin{document}`

`\doublespacing`

- escolha um tema, a cor do tema e a fonte dos slides (consulte [galeria de temas](#))

Use [loremipsum.io](#) para gerar o texto para colocar nos slides.

`rmarkdown`

`prettydoc`

Objetivo: Criar documentos html com formatação *bonita* rapidamente.

Para criar, especifique output: prettydoc::html_pretty no cabeçalho YAML.

```
---  
title: "Um lindo título"  
author: "Fulano de tal"  
date: 01/01/1900  
output: prettydoc::html_pretty  
---
```

Para mais informações, consulte prettydoc.statr.me.

Configurações disponíveis no cabeçalho YAML.

- `theme`: estilo dos slides
 - Opções disponíveis: `cayman`, `tactile`, `architect`, `leonids`, `hpstr`
- `highlight`: formatação do código no texto
 - Opções disponíveis: `github`, `vignette`

Vamos analisar um exemplo!

exemplo-8

Exercício

Crie um documento chamado `documento.html`, e inclua

- `title` - inclua o seguinte título: “um lindo documento”
- `date` - inclua a sua data de nascimento
- `author` - inclua o seu nome
- três slides com texto *dummy*
- inclua o tema `leonids`
- inclua a formatação de código `github`

Use loremipsum.io para gerar o texto para colocar nos slides.

`rmarkdown`

`rticles`

Objetivo: simplificar a produção de textos acadêmicos para revistas científicas.

Para criar um documento, use o seguinte código:

```
rmarkdown::draft(  
  "manuscrito.Rmd",  
  template = "elsevier",  
  package = "rticles"  
)
```

Para consultar as revistas disponíveis use o código `rticles::journals()`.

Vamos analisar um exemplo!

```
rmarkdown::draft(  
  "paper/manuscrito.Rmd",  
  template = "acm",  
  package = "rticles"  
)
```

Crie um manuscrito usando para publicar em uma revista da **springer** dentro da pasta `scripts`.

Algumas pacotes úteis que usam rmarkdown:

- pacote para automatização de envio de emails: `blastula`
- pacote para criação de blogs e websites usando o gerador de site estatístico – Hugo: `blogdown`
- pacote para redação de livros: `bookdown`

quarto

- Próxima geração do pacote rmarkdown desenvolvida por `posit` (sucessor de `rstudio`).
- quarto usa markdown e \LaTeX para produzir documentos.
- quarto permite que usemos código das seguintes linguagens:
 - `python`
 - `R`
 - `julia`
 - `Observable JS`
- Se você sabe usar rmarkdown, você sabe usar quarto.
- Para a linguagem R, quarto usa `knitr`.
- De forma semelhante ao pacote rmarkdown, quarto usa `pandoc`.
- Os arquivos tem extensão `.qmd`.
- Para criar documento `.pdf`, é necessário tem \LaTeX instalada (por exemplo, `miktex` e `texlive`).

Para detalhes, consulte `quarto`.

quarto

html

Documentos em formato html.

Para criar, especifique format: html no cabeçalho YAML.

```
---  
title: "Título magnífico"  
author: "Fulano de Tal"  
date: 01/01/1900  
format: html  
---
```

Neste caso, podemos usamos código html no texto.

Algumas configurações disponíveis no cabeçalho YAML.

- `toc`: inclusão de sumário
 - Campo booleano (`true` inclui o sumário)
 - Valor default: `false`
- `toc-depth`: nível de seção para inclusão no sumário
 - Valores possíveis: números inteiros de 1 a 6
 - Valor default: 6
- `toc-location`: localização do sumário. Essa opção pode estar desabilitada dependendo do tema.
 - Valores possíveis: `left`, `right` e `body`
 - Valor default: `right`

- `number-sections`: numeração das seções
 - Campo booleano (`true` - as seções serão numeradas)
 - Valor default: `false`
- `number-depth`: nível máximo de seção para numeração
 - Valores possíveis: números inteiros entre 1 a 6
 - Valor default: 6

Para retirar a numeração de uma seção, use o seguinte `{.unnumbered}`.

```
## Seção {.unnumbered}
```

Você pode esconder todos os códigos (`echo: false`) com a seguintes opções:

```
execute:  
  echo: true # inclusão do código de todos os chunks
```

Valores possíveis: `false`, `true`, e `fenced`. `fenced` mostra o *chunk* completamente incluindo ````${r}```` e `````.

-
- `embed-resources`: inclusão de figuras, css, js e outros elementos diretamente no código do arquivo html
 - Campo booleano (`true` - inclusão das figuras e outros elementos no código)
 - Valor default: `false`
 - `anchor-sections`: link para seções do documento
 - Campo booleano (`true` - inclusão de links para as seções)
 - Valor default: `false`
 - `link-external-icon`: mostre um ícone para indicar que o link é externo.
 - Campos booleano (`true` - inclusão do ícone nos links externos)
 - Valor default: `false`

- `link-external-newwindow`: abrir o link externo em uma nova aba?
 - Campo booleano (`true` - o link é aberto em nova aba)
 - Valor default: `false`
- `include-in-header`: inclusão de código html imediatamente antes de `</header>`
 - Valor possíveis: *filename* indicando a localização do arquivo dentro da pasta
- `include-before-body`: inclusão de código html imediatamente depois de `<body>`
 - Valor possíveis: *filename* indicando a localização do arquivo dentro da pasta
- `include-after-body`: inclusão de código html imediatamente antes de `</body>`
 - Valor possíveis: *filename* indicando a localização do arquivo dentro da pasta
- `code-fold`: botão para mostrar/esconder o código
 - Valores possíveis:
 - `false`: não inclui o botão
 - `true`: inclui o botão com o código escondido
 - `show`: inclui o botão com o código a mostra
 - Valor default: `false`
- `code-summary`: nome para inclusão no botão
 - Valor especial: texto entre aspas


```
code-tools:  
  source: true  
  toggle: true  
  caption: "Nome do botão"
```

or

```
code-tools: true
```

code-tools inclui o botão para esconder ou mostrar o código de todos os *chunks*.

Subopções de code-tools:

- source: mostrar o código da página? true - inclui o botão para mostrar o código da página.
- toggle: mostrar o botão de mostrar/esconder o código de todos os *chunk*? true - mostra o botão para mostrar o botão.
- caption: nome do botão. Valores possíveis: none ou texto.

- `code-copy`: inclusão de botão de copiar.
 - Valores possíveis:
 - `hover` - inclusão do botão de copiar ao passar o mouse em cima - default
 - `true` - inclusão do botão de copiar
 - `false` - nunca inclua o botão de copiar
- `code-line-number`: numeração nos blocos de código.
 - Campo booleano (`true` - inclua o numeração nos blocos de código)
 - Valor default: `false`
- `theme`: tema da página. Por padrão, quarto usa *Bootstrap 5*.
 - Valores possíveis: temas disponíveis para documentos html
- `highlight-style`: tema para formatação dos blocos de código.
 - Valores possíveis: temas disponíveis para formatação dos blocos de código

Tabela 14: Algumas opções de formatação.

Código	Descrição	Exemplo
max-width	Largura máxima da página	max-width: 1400px
mainfont	Fonte do documento	mainfont: monospace
fontsize	Tamanho da fonte	fontsize: 12px
fontcolor	Cor da fonte (letras)	fontcolor: #ff9522
monofont	Fonte das linhas de códigos	monofont: math
linestretch	Correponde a propriedade CSS line-height	linestretch: 1.7
backgroundcolor	Cor do fundo do documento	background: "#9596b4"
margin-*	Corresponde a propriedade CSS margin	margin-top: 1em

* pode ser: left, right, top, e bottom.

Vamos analisar um exemplo!

exemplo-10

Crie um documento chamado `documento.html`, e inclua:

- `title` - inclua o seguinte título: Documento `html`
- `date` - inclua a sua data de nascimento
- inclua três seções com texto *dummy*
- inclua o sumário e deixe ele a esquerda
- numere as seções
- inclua figuras, css, js e outros elementos diretamente no código
- mude o tema do documento `html`
- muda o estilo de formatação dos blocos de código
- mude o tamanho da fonte
- mude a largura da linha
- mude a fonte do documento

quarto

pdf

Necessário ter \LaTeX instalado (por exemplo, `miktex` e `texlive`).

Para criar, especifique `format: pdf` no cabeçalho YAML.

```
---  
title: "Título do documento .pdf"  
author: "Fulano de Tal"  
date: 01/01/1900  
format: pdf  
---
```

Neste caso, podemos usar código \LaTeX no texto.

Opções semelhantes ao formato output: `pdf_document` do pacote `rmarkdown`.

Verifique as opções em: `opções para format: pdf`.

- `cite-method`: método de produção de citação
 - Valor possíveis: `biblatex`, `natbib`, e `citeproc`
 - Valor default: `citeproc`
- `biblio-title`: texto da seção de bibliografia
- `include-in-header`: inclusão de código latex imediatamente antes de `\begin{document}`
 - Valor possíveis: *filename* indicando a localização do arquivo dentro da pasta
- `include-before-body`: inclusão de código latex imediatamente depois de `\begin{document}`
 - Valor possíveis: *filename* indicando a localização do arquivo dentro da pasta
- `include-after-body`: inclusão de código latex imediatamente antes de `\end{document}`
 - Valor possíveis: *filename* indicando a localização do arquivo dentro da pasta

- pdf-engine: aplicativo usado para gerar os documentos
 - Valores possíveis: xelatex, pdflatex, lualatex, tectonic, latexmk, context, wkhtmltopdf, prince, weasyprint, e pdffroff
 - Valor default: xelatex
- lof: inclua a lista de figuras
 - Campo booleano (true - inclusão da lista de figuras)
 - Valor default: false
- lot: inclua a lista de tabelas
 - Campos booleano (true - inclusão da lista de tabelas)
 - Valor default: false

Vamos analisar um exemplo!

exemplo-11

Exercício

Crie um documento chamado `documento.pdf`, e inclua os seguintes campos no cabeçalho YAML:

- `title` - inclua o seguinte título: “Documento pdf gerado com quarto”
- `date` - inclua sua data de nascimento
- `lang` - inclua o idioma português brasileiro - pt-br
- `fontsize` - formate o texto para 12pt
- inclua 3 cm de margem superior e a esquerda, e 2 cm para margem inferior e a direita
- inclua o pacote `enumerate` como preâmbulo
- Inclua duas seções, três parágrafos e o seguinte código (com seu resultado) em seu documento:

```
summary(iris)
ggplot(iris) + geom_bar(aes(x = Species))
```



quarto

docx

Use `format: pdf` e `format: html` sempre que possível.

`format: docx` é útil por causa do monopólio da Microsoft em produtos de processamento de texto (e planilha).

Para customizar customizações, use `template`:

- Crie um documento simples `template.docx`:

```
quarto pandoc -o relatorio/template.docx \  
  --print-default-data-file reference.docx
```

- Modifique o estilo e as configurações de margens deste documento
- Modifique o cabeçalho YAML para:

```
---  
title: "Um título lindo"  
author: "fulnao de tal"  
date: 01/01/1900  
output:  
  docx:  
    reference-doc: template.docx  
---
```

Vamos analisar um exemplo!

exemplo-12

Crie um documento chamado `documento.docx`, e inclua os seguinte campos no cabeçalho YAML:

- `title` - inclua o seguinte título: Documento word usando quarto
- `date` - inclua sua data de nascimento
- `author` - inclua seu nome
- Inclua um documento de template:
 - Modifique o estilo de título do documento:
 - fonte: times new roman
 - Negrito e itálico
 - tamanho: 30pt
- Inclua três seções com um parágrafos cada.