

Apresentações com xaringan

Curso: Relatórios e visualização de dados



Pacote {xaringan}

Pacote {xaringan}

- Possibilita criar apresentações **ninjas** com R Markdown
- Utiliza a biblioteca `remark.js`
- Gera resultado em HTML
- HTML pode ser convertido para PDF (estático)



Logo: Pacote xaringan

Estrutura da apresentação

- No xaringan, utilizamos o arquivo `.Rmd` para escrever o código da apresentação que é compilado para um arquivo `.html`
- Esse arquivo `.html` é interpretado pelo navegador (ex. Chrome), utilizando também os arquivos `.css` e `.js`



Estrutura da apresentação



- **HTML** (HyperText Markup Language - Linguagem de Marcação de Hipertexto): é uma linguagem de marcação, é usado para estruturar a página.
- **CSS** (Cascading Style Sheets - Folha de Estilo em Cascata): é usado para estilizar os elementos escritos no HTML.
- **Javascript**: é uma linguagem de programação, permite que as páginas sejam dinâmicas.

O que precisa saber?

- **HTML:** É gerado automaticamente ao compilar o `.Rmd`
- **CSS:** É usado para personalizar a parte visual da apresentação. Para saber mais sobre temas: [xaringan wiki](#)
- **JS:** utiliza o `Remark.js`, mas aceita outros códigos caso seja necessário.



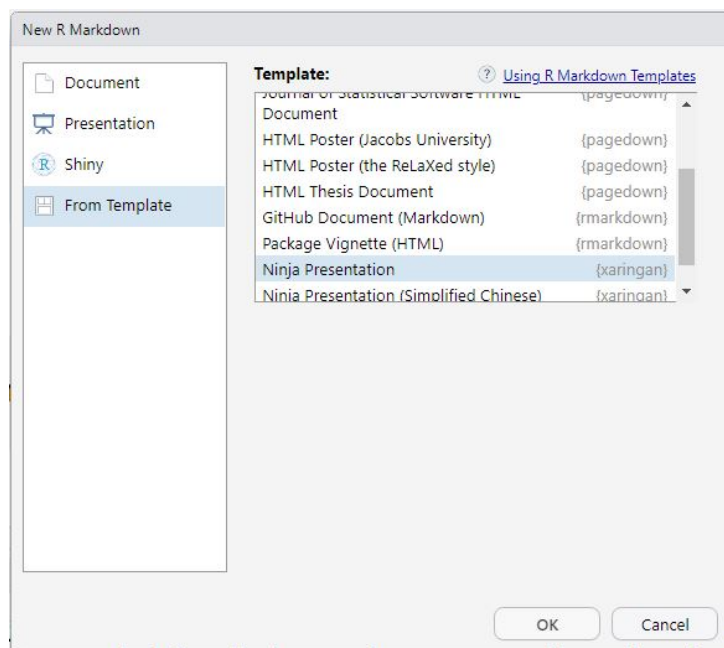
Recomendações

- Desenvolva a apresentação em um projeto no RStudio: facilita muito para organizar os arquivos e posteriormente sincronizar com o GitHub.
- Organize o seu diretório.
Por exemplo:
 - `img/` - imagens, figuras, etc.
 - `libs/` - bibliotecas.
 - `css/` - para arquivos `.css` (opcional)
- 🛎 **Importante:** Compile (knit) o arquivo com frequência!

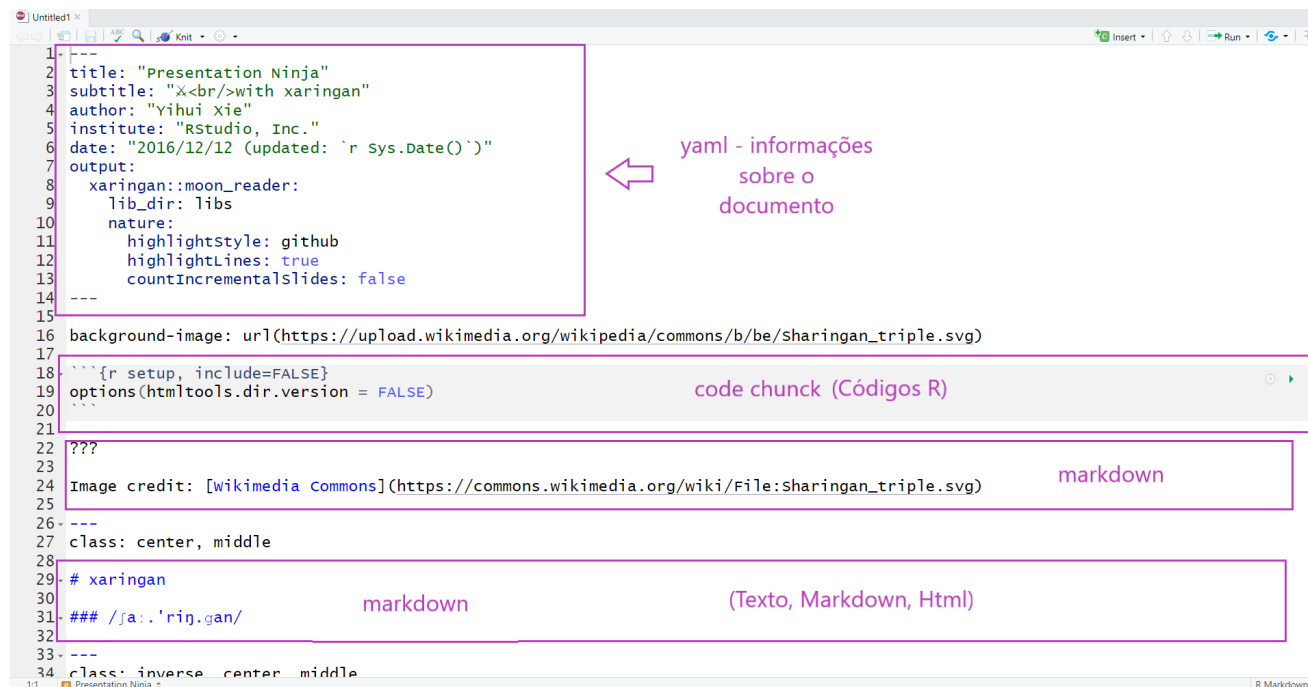
Criar um arquivo R Markdown

Com template Ninja Presentation

- File > New File > R Markdown...
- Na janela New R Markdown:
From Template > Ninja Presentation



Estrutura arquivo R Markdown



The image shows a screenshot of an R Markdown file in a text editor. The file is divided into several sections, each highlighted with a pink box and labeled with a pink arrow and text:

- YAML front-matter:** Lines 1-14, containing document metadata like title, subtitle, author, institute, date, and output options. Labeled "yaml - informações sobre o documento".
- Code chunk:** Lines 18-19, containing R code for setting up the presentation. Labeled "code chunk (Códigos R)".
- Image credit:** Line 24, containing a link to the image source. Labeled "markdown".
- Class and style:** Lines 26-34, containing CSS class and style definitions. Labeled "markdown" and "(Texto, Markdown, Html)".

```
1 ---
2 title: "Presentation Ninja"
3 subtitle: "X<br/>with xaringan"
4 author: "Yihui Xie"
5 institute: "RStudio, Inc."
6 date: "2016/12/12 (updated: `r Sys.Date()`)"
7 output:
8   xaringan::moon_reader:
9     lib_dir: libs
10     nature:
11       highlightstyle: github
12       highlightLines: true
13       countIncrementalSlides: false
14 ---
15
16 background-image: url(https://upload.wikimedia.org/wikipedia/commons/b/be/sharingan_triple.svg)
17
18 {r setup, include=FALSE}
19 options(htmltools.dir.version = FALSE)
20
21
22 ???
23
24 Image credit: [wikimedia commons](https://commons.wikimedia.org/wiki/File:Sharingan_triple.svg)
25
26 ---
27 class: center, middle
28
29 # xaringan
30
31 ## /ja:.'rin.gan/
32
33 ---
34 class: inverse_center_middle
```

Como delimitar slides?

- No xaringan, os slides são delimitados por `---` no início da linha.
- Para fazer slides que aparecem aos poucos (*incremental slides*), utilize `--` no início da linha.

Por exemplo:

O conteúdo aparece aos poucos!

Chunk de código

Para inserir um chunk, ou campo de código

- O chunk deve *iniciar e terminar* por uma série de 3 crases `````
- O chunk deve iniciar fornecendo os parâmetro entre `{}`
Se quiser ver o código e o resultado use `echo = TRUE`,
se quiser ver apenas o resultado use `echo = FALSE`.

```
14 ---
15
16 background-image: url(https://upload.wikimedia.org/wikipedia/commons/b/be/sharingan_triple.svg)
17
18 {r|setup, include=FALSE}
19 options(htmltools.dir.version = FALSE)
20
21
22 ???
23
24 Image credit: [wikimedia commons](https://commons.wikimedia.org/wiki/File:Sharingan_triple.svg)
25
26 --- sinal para próximo slide
27
```

Chunk options - opções
do campo de código

1) Linguagem de programação; 2) Nome do campo de
código (opcional); 3) Chunk options (opcional)

Executa o código que está
no chunk

Metadados da apresentação: YAML

Alterar YAML

- O YAML possui informações importantes para a apresentação como:
 - título;
 - subtítulo;
 - autor;
 - instituição;
 - data;
 - etc.
- **output:** indica o formato, por exemplo `html_document`, `pdf_document`, etc. No caso do `xaringan`, o output é `xaringan::moon_reader`
- **libs_dir:** pasta onde colocará bibliotecas necessárias. Por exemplo: `lib_dir: libs` indica que a biblioteca que usamos `remark-latest.min.js` será salva na pasta `libs`.

Alterar `YAML`

- Cuidado com a indentação!



Exemplo do código Y A M L:

```
---  
title: "Título da apresentação"  
subtitle: "Subtítulo da apresentação"  
author: "Autor"  
institute: "Instituição"  
date: "07/09/2019"  
output:  
  xaringan::moon_reader:  
    lib_dir: libs  
    nature:  
      highlightStyle: github  
      highlightLines: true  
      countIncrementalSlides: false  
---
```

- Aperte H ou ? para conferir a lista de atalhos do xaringan

Encoding

- Adicionar no `YAML`:

```
encoding: "UTF-8"
```

"Uma **codificação de caracteres** é um padrão de relacionamento entre um conjunto de caracteres(...) com um conjunto de outra coisa, como por exemplo números ou pulsos elétricos com o objetivo de **facilitar o armazenamento de texto em computadores e sua transmissão** através de redes de telecomunicação."

[Wikipedia](#)

Biblioteca `remark.js`

- Seu uso é opcional, mas recomendado!
Possibilita que sua apresentação seja **visualizada offline**!
- Duas opções para fazer download da biblioteca `remark.js`:
 - A função `xaringan::summon_remark()` faz o download da versão mais atual do `remark.js` e salva na pasta `/libs`
 - Usando a função `download.file` indicando como primeiro parâmetro a url de origem, bem como o destino como parâmetro do *destfile*

Biblioteca `remark.js`

```
xaringan::summon_remark()
```

```
# Ou:
```

```
download.file("https://remarkjs.com/downloads/remark-latest.min.js"  
              destfile = "libs/remark-latest.min.js")
```

- Adicionar no `YAML`:

```
output:  
  xaringan::moon_reader:  
    chakra: libs/remark-latest.min.js  
    lib_dir: libs
```

Adicionando conteúdo na apresentação

- Atualiza a apresentação enquanto é editada:

```
xaringan::inf_mr()
```

Estilo (.css)

Estilo (`.css`)

- Possibilidades:
 - Temas do xaringan
 - Temas com `{xaringanthemer}`
 - Alterando o `.css`: arquivo `custom.css`

Tema

- Depende da versão do `xaringan` que está usando.
- Para consultar os temas disponíveis:

```
names(xaringan::list_css())
```

```
## [1] "chocolate-fonts" "chocolate"      "default-fonts"  "default"
## [5] "duke-blue"        "fc-fonts"        "fc"             "hygge-duke"
## [9] "hygge"            "ki-fonts"        "ki"             "kunoichi"
## [13] "lucy-fonts"       "lucy"            "metropolis-fonts" "metropolis"
## [17] "middlebury-fonts" "middlebury"      "nhsr-fonts"     "nhsr"
## [21] "ninjutsu"         "rladies-fonts"   "rladies"        "robot-fonts"
## [25] "robot"            "rutgers-fonts"   "rutgers"        "shinobi"
## [29] "tamu-fonts"       "tamu"            "uio-fonts"      "uio"
## [33] "uo-fonts"         "uo"              "uol-fonts"      "uol"
## [37] "useR-fonts"       "useR"
```

Tema

- 🛎 Dica:
Usar em dupla: `nomedotema` e `nomedotema-fonts`
- Adicionar no `YAML`:

```
output:  
  xaringan::moon_reader:  
    css: ["metropolis", "metropolis-fonts"]
```

Exemplo do código YAML:

```
---
title: "Título da apresentação"
subtitle: "Subtítulo da apresentação"
author: "Autor"
institute: "Instituição"
date: "07/09/2019"
encoding: "UTF-8"
output:
  xaringan::moon_reader:
    chakra: libs/remark-latest.min.js
    css: ["rladies", "rladies-fonts"]
    lib_dir: libs
    nature:
      highlightStyle: github
      highlightLines: true
      countIncrementalSlides: false
---
```


xaringanthemer

- Cria o arquivo `.css`!
- Duas possibilidades para usar:
 - Criar um arquivo com template:
| Ninja **themed** presentation
- Usar em uma apresentação que já existe: [veja aqui](#)
- No YAML:

```
output:  
  xaringan::moon_reader:  
    lib_dir: libs  
    chakra: libs/remark-latest.min.js  
    css: [xaringan-themer.css]
```

Customizando o CSS

- Podemos alterar coisas específicas no CSS.
- Dica: usar o inspetor do navegador para descobrir o que mudar. Firefox é `CTRL + SHIFT + C`
- Crie um arquivo CSS no projeto (Ex: `custom.css`), e adicione no YAML.
- Dica 1: usar `{xaringan-themer}`, e arquivos CSS customizados.

```
output:
  xaringan::moon_reader:
    css: [xaringan-themer.css, custom.css]
```

- Dica 2: usar temas do xaringan e arquivos CSS customizados.

```
output:
  xaringan::moon_reader:
    css: ["rladies", "rladies-fonts", custom.css]
```

Compartilhando a apresentação

PDF

- Para compartilhar em PDF, é possível converter o HTML em PDF com a função `pagedown::chrome_print()`:

```
remotes::install_github('rstudio/pagedown')  
pagedown::chrome_print("index.html")
```

Página web

- HTML - Pode disponibilizar online :)
 - Em seu site;
 - A partir de um repositório do GitHub:
 - [GitHub Pages](#)
 - [Netlify](#) - Faça login com a sua conta no GitHub!
 - Arrastando para enviar:
 - [Netlify Drop](#)



- Dica: você pode encurtar o link com o [bit.ly](#)

Pacote {xaringanExtra}

- O pacote `xaringanExtra` oferece extensões para utilizar nas apresentações que fazemos com `xaringan`.
- O pacote deve ser instalado através do GitHub:

```
# install.packages("devtools")  
remotes::install_github("gadenbuie/xaringanExtra")
```

- O melhor lugar para saber mais é [a documentação](#)

Outros pacotes

Aumentando as possibilidades!

- [pagedown](#) - ✓ CRAN, ✓ GitHub
- [xaringanthemer](#) - ✓ CRAN, ✓ GitHub
- [giphyr](#) - ✓ CRAN, ✓ GitHub
- [metathis](#) - ✓ CRAN, ✓ GitHub
- [xaringanExtra](#) - ✗ CRAN, ✓ GitHub
- [countdown](#) - ✗ CRAN, ✓ GitHub
- [emo](#) + [eremoji](#) - ✗ CRAN, ✓ GitHub