

# Introdução ao R Markdown

Eduardo José de Campos Lemos Júnior, Samuel Vianna Quintanilha

2020-08-18



# Contents



# Chapter 1

## Introdução

Neste curso, iremos aprender a utilizar o R Markdown para gerar documentos.

### 1.1 O que é

Markdown é uma linguagem de marcação usada para formatar de maneira simples os textos redigidos e converte-los em HTML. John Gruber e Aaron Swartz, os criadores desse sistema, utilizaram marcadores como: `#`, `\*`, `!`, `,`, `[]` e `()`. Assim é possível inserir em nossos textos: títulos, listas, formatação de fonte, imagens e tabelas. R Markdown é um documento criado no R Studio (Ou em outras IDEs) que possibilita empregar os recursos da linguagem markdown citados acima em conjunto com a linguagem R, permitindo a melhor organização de análises, relatórios e códigos em um só documento.

### 1.2 Possíveis tipos de Outputs

O R Markdown apresenta várias possibilidades de outputs (renderizar) nos formatos de documentos, apresentações, entre outros, sendo que em cada formato há várias opções de customização. Vejamos abaixo os principais:

Documentos:

- `html_document` – documento no formato HTML;
- `pdf_document` – documento no formato PDF (via o modelo LaTeX);
- `word_document` – documento no formato do editor de texto Microsoft Word (`docx`);
- `odt_documento` – documento no formato dos editores de texto Libre Office e OpenDocument;
- `rtf_documento` – documento no formato Rich Text Format (`rtf`).

Apresentações (slides):

- `ioslides_presentation` – apresentação no formato HTML com ioslides;
- `beamer_presentation` – apresentação no formato PDF com LaTeX Beamer;
- `powerpoint_presentation` – apresentação no formato power point.

Outros:

- `flexdashboard::flex_dashboard` – apresentação interativa com dashboards;
- `htm_vignette` – R package vignette no format HTML
- `github_document` – document no format GitHub

Você pode escolher o output desejado quando for criar um documento conforme a figura 1. Para fazer deve-se clicar em `file > new file > R Markdown`. Abrirá uma aba e nela há quatro formas de output previamente estabelecidas, a saber: Document (HTML, PDF e Word), Presentation (HTML (ioslides), HTML (slidy), PDF (Beamer) e PowerPoint), Shiny (Shiny Document e Shiny Presentation) e From Template (GitHub document e Package Vignette). Escolha uma e clique em OK. Além disso, você também pode alterar o formato utilizando a função abaixo, sendo que `render` refere-se ao local que está salvo seu documento e `output_format` ao tipo de documento desejado, conforme os exemplos apontados no início.

```
render("teste.Rmd", output_format = "pdf_document")
```

O mesmo pode ser feito para outros formatos. Abaixo está presente uma lista com todos os formatos suportados por padrão com o pacote `rmarkdown`.

- `beamer_presentation`
- `context_document`
- `github_document`
- `html_document`
- `ioslides_presentation`
- `latex_document`
- `md_document`
- `odt_document`
- `pdf_document`
- `powerpoint_presentation`
- `rtf_document`
- `slidy_presentation`
- `word_document`

### 1.3 Criando o primeiro documento

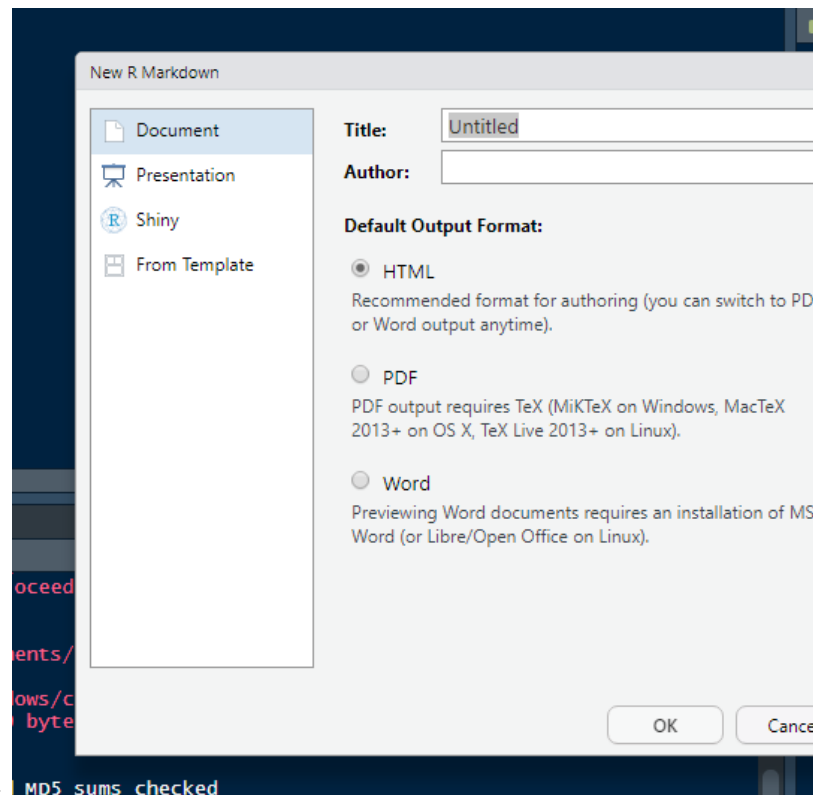
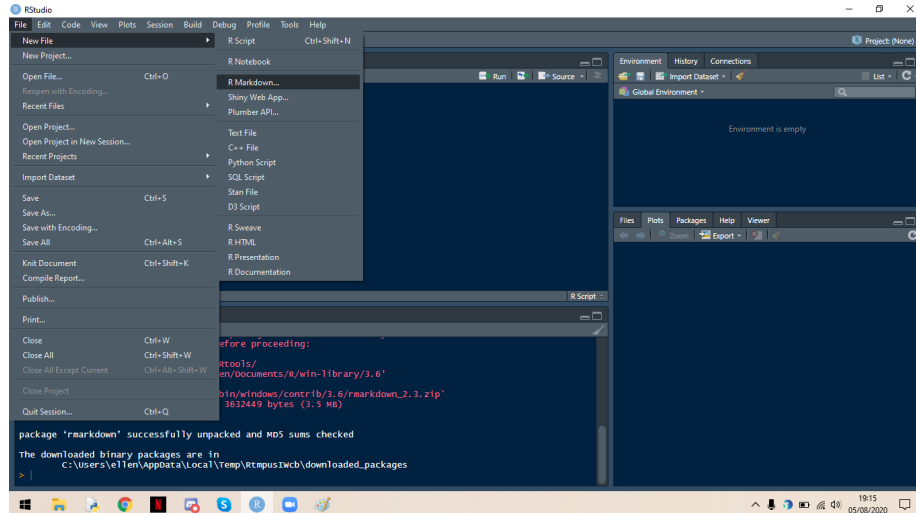
Para gerar um arquivo em R Markdown é necessário abrir o programa R Studio, e instalar o pacote `rmarkdown` :

```
install.packages("rmarkdown")
```

### 1.3. CRIANDO O PRIMEIRO DOCUMENTO

7

Após a instalação do pacote no R Studio, siga os seguintes passos:



Em seguida, escolha o tipo de arquivo desejado:

Obs: Para gerar documentos em PDF, é necessário ter instalado em seu computador o programa Latex

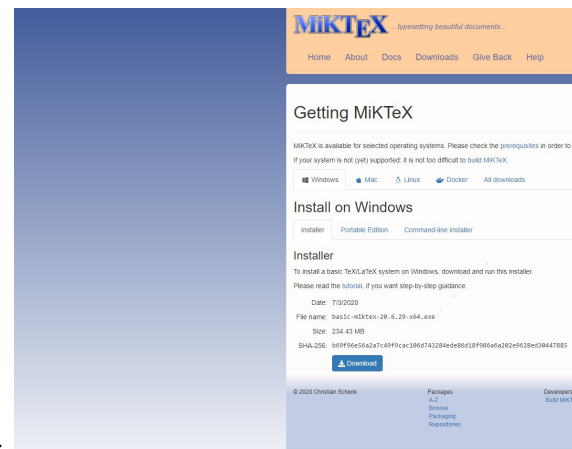
Seguindo os passos acima, você terá criado o seu primeiro documento em R Markdown.

Vale ressaltar que é possível utilizar o R Markdown sem que tenha instalado o R Studio, porém, é necessário ter instalado o programa Pandoc

## 1.4 Download MiKTeX

Para exportar um arquivo PDF utilizando o R Markdown é necessário um motor LaTeX pois é ele que irá converter o arquivo .tex em PDF. Então é necessário que tenha instalado em seu computador o programa MiKTeX,. Para fazer o download é só acessar o link: <https://miktex.org/download>.

### 1.4.1 Windows



Selecione a aba Windows e clique no botão de download:

### 1.4.2 Linux

Selecione a aba Linux em seguida a aba de sua distribuição Linux para receber as

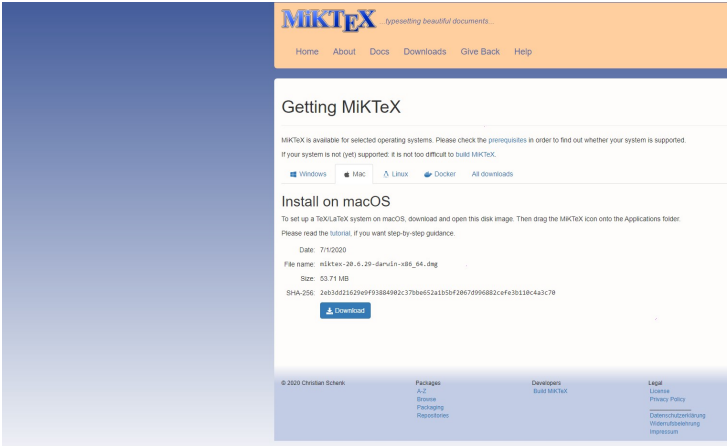


instruções de instalação:



1.4.3 MacOS

Selecione a aba macOS e clique no botão de download:





## Chapter 2

# Sintaxe

### 2.1 Prêambulo

No início de um documento R Markdown, é utilizada a linguagem `yaml` para definir as configurações do seu arquivo. As configurações disponíveis no preâmbulo do seu documento R Markdown são variadas, podendo inclusive ter diferentes opções para diferentes tipos de arquivos. Uma das configurações mais importantes para se definir no seu preambulo é o tipo de documento a ser gerado.

#### 2.1.1 Definindo tipo de output

A opção de tipo de documento é definida da seguinte forma:

```
---  
output: pdf_document  
---
```

No código acima, `output` foi definido como `pdf_document()`. `Output` é o formato final do seu documento, e `pdf_document()` é, dentro do pacote `rmarkdown`, o formato pdf. Para gerar arquivos de formatos diferentes, é necessário somente que seja modificado a opção de `output` no preâmbulo. Note que a opção definida no preâmbulo foi escrita entre duas linhas tracejadas, não é possível definir suas configurações fora dessas linhas tracejadas, da mesma forma, não é possível escrever partes do seu documento ou rodar códigos de R dentro das linhas. Por padrão no `yaml` utilizado no preâmbulo de R Markdown, a opção a ser definida é escrita sem espaços, seguida de dois pontos, espaço e então a definição da opção. Existem opções dentro de outras opções, como por exemplo `pdf_document()`, podemos definir configurações se o documento terá ou não sumário, quantos níveis de cabeçalho serão utilizados no sumário, etc.

```
---
```

```
output: pdf_document:
  toc: true
  toc_depth: 3
  latex_engine: xelatex
---
```

No exemplo acima, definimos `toc` (table of contents) como `true`, o que vai fazer com que seja gerado um sumário no documento final, perceba que em `yaml` o valor lógico de verdadeiro é escrito com todas as letras minúsculas. Definindo `toc_depth` como 3, quando o sumário for gerado, até três níveis de cabeçalho apareceram no sumário.

### 2.1.2 Informações Gerais

Além do tipo de documento, e das opções de cada tipo de documento, podemos definir algumas opções gerais, como autor, título do documento, e data.

```
---
author: "Fulano"
output: pdf_document:
  toc: true
  toc_depth: 3
  latex_engine: xelatex
---
```

Seguindo com o preâmbulo já feito anteriormente, foi adicionada a opção `author`, que irá definir o autor do seu documento. Por padrão, o autor irá aparecer na página inicial de diversos tipos de documento.

```
---
date: 1 de janeiro de 1970
output: pdf_document:
  toc: true
  toc_depth: 3
  latex_engine: xelatex
author: Fulano
---
```

Utilizando a opção `date` podemos definir uma data para o documento, semelhante ao autor, a data aparece por padrão no início dos documentos. Perceba que no exemplo acima, o nome do autor e a data foram escritas sem aspas, mas irão funcionar da mesma forma, perceba também que a opção do autor foi trocada de ordem, as opções do preâmbulo não necessitam de uma ordem específica, mas as opções de dentro de outras opções devem sempre estar abaixo da opção mãe (sei lá como chamar isso) e com uma indentação a mais.

```
---
author: Fulano
date: 1 de janeiro de 1970
```

```

title: Título
output: pdf_document:
  toc: true
  toc_depth: 3
  latex_engine: xelatex
---
```

Por fim foi definida também a opção `title` que irá definir o título do documento final. Para maior customização do seu documento pelo preâmbulo veja o capítulo 4.

### 2.1.3 Sumario

## 2.2 Textos

### 2.2.1 Títulos

```

# Título Nível 1

## Título Nível 2

### Título Nível 3

#### Título Nível 4

##### Título Nível 5

##### Título Nível 6
```

### 2.2.2 Formatação de textos (negrito, itálico, sobrescrito, tachado e código)

```

*Negrito* _Negrito_

Negrito Negrito

**Italico** __italico__

Italico Italico

textosobrescrito

textosobrescrito

~~~tachado~~

~tachado
```

### 2.2.3 Links

Para um link devemos utilizar a seguinte sintaxe:

```
[nome do link](url do link)
[Curso de RMarkdown](https://estatsej.github.io/curso_rmarkdown)
```

Existem também outras variações para que utilizemos os links em nosso material.

**1 - Nome do link seja ele próprio:**

```
https://estatsej.github.io/curso_rmarkdown
```

**2- Link contendo um título, que aparece ao deixar o cursor do mouse em cima do link:**

```
[Curso de RMarkdown](https://estatsej.github.io/curso_rmarkdown "Aqui está o nosso curso")
```

Podemos observar: Curso de RMarkdown

### 2.2.4 Listas

**Lista Ordenada**

1. Primeiro item
2. Segundo item
3. Terceiro item

**Lista Não-Ordenada**

- Primeiro item
- Segundo item
- Terceiro item

**Lista com Sublista**

1. Item
  - Um sub-item
  - Outro sub-item

## 2.3 Inserindo Imagens

Antes de inserir a imagem escolhida podemos definir a configuração global para todas as imagens, lembrando que a configuração feita diretamente na imagem vai sobrepor a configuração global.

```
```{r setup, include=FALSE}
library(knitr)
opts_chunk$set(echo = FALSE,
               out.width = "10%",
               fig.align = "center")
...```
```