



**Cursos Integrados  
em Vigilância em Saúde**

*Curso* —

## **Produção automatizada de relatórios na vigilância em saúde**

## **UNIVERSIDADE FEDERAL DE SANTA CATARINA**

Reitor Irineu Manoel de Souza

Vice-Reitora Joana Célia dos Passos

Pró-Reitora de Pós-graduação Werner Kraus

Pró-Reitor de Pesquisa e Inovação Jacques Mick

Pró-Reitor de Extensão Olga Regina Zigelli Garcia

## **CENTRO DE CIÊNCIAS DA SAÚDE**

Diretor Fabrício de Souza Neves

Vice-Diretora Ricardo de Souza Magini

## **DEPARTAMENTO DE SAÚDE PÚBLICA**

Chefe do Departamento Rodrigo Otávio Moretti Pires

Subchefe do Departamento Sheila Rúbia Lindner

Coordenadora do Curso Alexandra Crispim Boing

## **INSTITUTO TODOS PELA SAÚDE (ITPS)**

Diretor presidente Jorge Kalil (Professor titular da Faculdade de Medicina da Universidade de São Paulo; Diretor do Laboratório de Imunologia do Incor)

## **ASSOCIAÇÃO BRASILEIRA DE SAÚDE COLETIVA (ABRASCO)**

Presidente Rosana Teresa Onocko Campos

## **EQUIPE DE PRODUÇÃO**

Denis de Oliveira Rodrigues

Kamila de Oliveira Belo

Marcelo Eduardo Borges

Oswaldo Gonçalves Cruz

Alexandra Crispim Boing

Antonio Fernando Boing



*Curso*

# Produção automatizada de relatórios na vigilância em saúde



Dados Internacionais de Catalogação-na-Publicação (CIP)

P964 Produção automatizada de relatórios na vigilância em saúde/ Denis de Oliveira Rodrigues, Kamila de Oliveira Belo, Marcelo Eduardo Borges, Oswaldo Gonçalves Cruz, Alexandra Crispim Boing, Antonio Fernando Boing . Santa Catarina ; São Paulo; Rio de Janeiro : UFSC ; ITPS ; Abrasco; 2022. 78p. (Cursos Integrados em Vigilância em Saúde).

Publicação Online

[10.52582/curso-analise-dados-vigilancia-modulo7](https://repositorio.ufsc.br/handle/10.52582/curso-analise-dados-vigilancia-modulo7)

1. Vigilância em saúde 2. Análise de dados I. Título

# Sumário

Transformando suas análises de dados em relatórios automatizados .....	06
<b>1. Como escrever um relatório automatizado? .....</b>	<b>07</b>
1.1 O que é o Rmarkdown? .....	08
1.2 Por que utilizar o Rmarkdown? .....	09
<b>2. Como criar um arquivo Rmarkdown? .....</b>	<b>10</b>
<b>3. Construindo um relatório automatizado .....</b>	<b>20</b>
3.1 Edição dos textos em markdown .....	22
3.2 Formatação de texto simples .....	23
3.3 Formatação de títulos, subtítulos e listas .....	26
3.4 Inserir links (internos e externos) .....	35
<b>4. Incluindo imagens ao relatório .....</b>	<b>38</b>
4.1 Formatando imagens .....	41
<b>5. Incluindo códigos ao relatório .....</b>	<b>46</b>
5.1 Escrevendo os códigos de um relatório .....	54
<b>6. Incluindo gráficos ao relatório .....</b>	<b>58</b>
<b>7. Incluindo tabelas ao relatório .....</b>	<b>64</b>
<b>8. Exportando o relatório .....</b>	<b>68</b>
<b>9. Outros componentes e ajustes finais .....</b>	<b>70</b>
9.1 Automatizando a atualização do texto .....	70
9.2 Quebras de linha .....	72
9.3 Espaçamento .....	73
9.4 Cabeçalho YAML .....	75

# Transformando suas análises de dados em relatórios automatizados

Já imaginou ter os informes epidemiológicos prontos em menos de 2 minutos? Se você está cansado de retrabalhos, este curso é para você!

Isso mesmo, neste curso vamos colocar em prática um dos pilares mais importantes para a vigilância em saúde: a transparência! Você publicará as análises de situação de saúde no tempo oportuno, tornando-a acessível para amplo conhecimento e sem retrabalho. Lembre-se de que a transparência é um dever das instâncias públicas e privadas de saúde, sendo elemento estruturante para a universalidade, a integralidade e a equidade na atenção à saúde.

Utilizando a linguagem de programação R, você transformará as análises realizadas diariamente no serviço de vigilância em saúde em relatórios acessíveis aos seus colegas, gestores públicos, pesquisadores, profissionais de saúde ou população em geral do município ou estado, tudo de forma automatizada! Você poderá atribuir avaliações em suas análises, personalizar relatórios, produzir informes ou boletins epidemiológicos e até mesmo painéis de indicadores ou dashboards em seu dia a dia.

## Ao final deste curso, você será capaz de:

1. utilizar o Rmarkdown para automatizar a produção de relatórios;
2. organizar suas rotinas de análises em relatórios dinâmicos (interativos);
3. criar relatórios com análises complexas;
4. salvar seus relatórios no formato .pdf, .html, e .docx de forma mais ágil.

## Atenção



Para seguir com este curso você deve conhecer as ferramentas básicas para uso da linguagem R e do RStudio. Lembre-se que você pode acessar a qualquer momento o curso **“Análise de dados para a vigilância em saúde – curso básico”** e o curso **“Visualização de dados de interesse para a Vigilância em Saúde”** obtendo os códigos desejados para a confecção do seu relatório. Caso não tenha feito os cursos, sugerimos fortemente que se inscreva neles. Maiores informações em <https://www.abrasco.org.br/site/analise-de-dados-para-a-vigilancia-em-saude/>

## *1 Como escrever um relatório automatizado?*

Na vigilância em saúde, a produção de relatórios é diária podendo ser bastante desgastante. É comum que tenhamos solicitações de análises de outras áreas técnicas, da imprensa, de nossas chefias, do secretário de saúde ou de outros secretários e até mesmo da população em geral. Quase sempre essas solicitações são imediatas, e você tem pouco tempo para disponibilizá-las.

Imaginamos que o seu trabalho sofreu o impacto desta pressão diária durante os anos epidêmicos de dengue ou na atual pandemia de covid-19. Certamente você precisou criar arquivos bases e diversos métodos para atender a tantas demandas urgentes.

Neste curso você irá aprender como é possível construir uma rotina de forma automatizada, desenvolvendo técnicas de estruturação inicial de um relatório com apoio do R. A base construída aqui poderá ser replicada para qualquer uma das suas análises diárias. Isso mesmo: qualquer uma! Incluindo as tabelas ou gráficos que você aprendeu a fazer com *scripts* no R. Eles poderão ser incorporados e reproduzidos de forma automática, de qualquer computador que possua o R e o RStudio instalados.

Vamos criar um relatório juntos em que você fará as modificações necessárias utilizando uma ferramenta poderosa do R: o Rmarkdown. Aproveite!

## 1.1 O que é o Rmarkdown?

O Rmarkdown é uma ferramenta que permite a produção de relatórios, documentos e apresentações no momento em que você está fazendo seus scripts com R. Para isto ele utiliza uma linguagem chamada de *markdown*, uma “linguagem de marcação”, utilizada para informar ao computador como ele deve interpretar e estruturar seus arquivos ou documentos.

Uma página de *internet*, por exemplo, utiliza uma linguagem de marcação que chamamos de `html`. Esta linguagem informa o navegador de *internet* quais os elementos presentes em um *site*, qual a localização de cada um e também a sua formatação. O *markdown* age de forma análoga fazendo as mesmas ações do `html`, estruturando documentos em diversos formatos. Assim, o Rmarkdown permite que possamos unir a linguagem *markdown* a pedaços de códigos que utilizam a linguagem R! Ou seja, o Rmarkdown é a junção da linguagem *Markdown* com os códigos em R.

O uso do Rmarkdown será muito útil, pois permitirá ao profissional de vigilância em saúde a reprodução das suas análises automatizando a exportação destes resultados para documentos de diferentes tipos como: `.html`, `.pdf`, `.docx`, entre outros. Com estudo esta poderosa ferramenta lhe permitirá criar os tão utilizados *dashboards*, *websites* e até mesmo livros.

## 1.2 Por que utilizar o Rmarkdown?

Imagine que semanalmente você necessita produzir informes epidemiológicos que tornem pública a situação da dengue em seu município ou estado! Você provavelmente já deve ter se deparado com o passo a passo abaixo para obter as análises de situação de dengue. Observe e perceba se há semelhança no seu dia a dia:

1. Você acessa o SINAN *online* e exporta a base de dados **DENGON.dbf** mais recente para seu computador.
2. Você abre o Excel (**.xlsx**) para fazer alguns gráficos como a série histórica de notificações de dengue.
3. Você também utiliza o Excel (**.xlsx**) para produzir tabelas com os valores de incidência e mortalidade de casos por bairro ou região de saúde.
4. Você abre um editor de documento, como o Word (**.docx**), e inicia a escrita do relatório em uma máscara pré formatada.
5. Os gráficos e tabelas produzidos são copiados e colocados em seu editor de texto.
6. E, por fim, as informações são atualizadas manualmente e o documento final é enviado.

Estas etapas podem levar horas para serem concluídas, não é mesmo? E quando temos um aumento da magnitude da doença torna-se um trabalho hercúlio reproduzir este passo a passo semanalmente. Com a utilização do **Rmarkdown**, a maioria destas tarefas rotineiras podem ser automatizadas com *scripts* escritos em **Rmarkdown**.

Com o **R** a única etapa manual a ser feita é a etapa de número 1. Após a importação do banco de dados para o diretório que você escolher em seu computador, bastará apenas clicar em *run* e rodar todo seu código escrito no **Rmarkdown**. A partir daí o **R** **processará todas as etapas necessárias (2 a 6) para seu relatório de forma automática**. Ou seja, ele atualizará de forma automática:

- a produção do gráfico com a série histórica dos casos de dengue,
- a produção das tabelas de incidência e mortalidade para cada bairro,
- a atualização do texto escrito de análise, e
- a união de todas estas informações em um documento final (relatório final) que você poderá escolher um ou vários formatos.

Parece revolucionário, não é mesmo? E é! Utilizando o **Rmarkdown** você poderá obter um informe epidemiológico realmente em 2 minutos!

## 2 Como criar um arquivo Rmarkdown?

O R é uma linguagem de programação poderosa. Com ele é possível automatizar suas análises transformando-as em relatórios. Utilizando o Rmarkdown seremos capazes de tornar os relatórios que você já sabe fazer acessíveis em pouco tempo. Mas vamos com calma. Você verá este processo todo exemplificado neste curso!

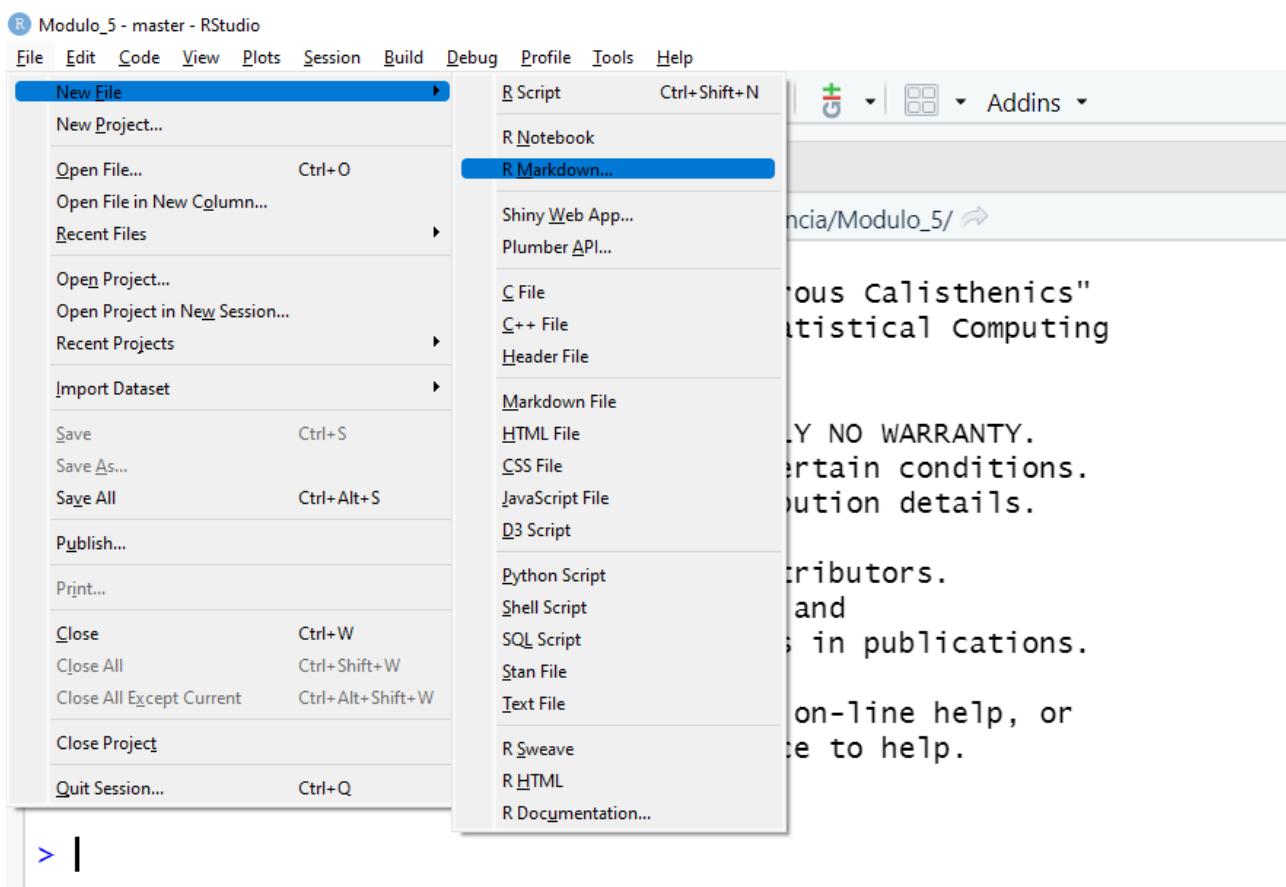
Antes de aprendermos a utilizar e personalizar relatórios, vamos aprender como criar um arquivo no formato Rmarkdown. Para este primeiro passo, abra o RStudio em seu computador seguindo o passo a passo abaixo:

1. vá até o menu do RStudio e clique em File (Arquivo),
2. em seguida em New File (Novo Arquivo), e
3. depois em Rmarkdown...

Se o seu RStudio estiver em português utilize o mesmo passo a passo, porém você verá o caminho da seguinte forma: Arquivo > Novo Arquivo > Rmarkdown....

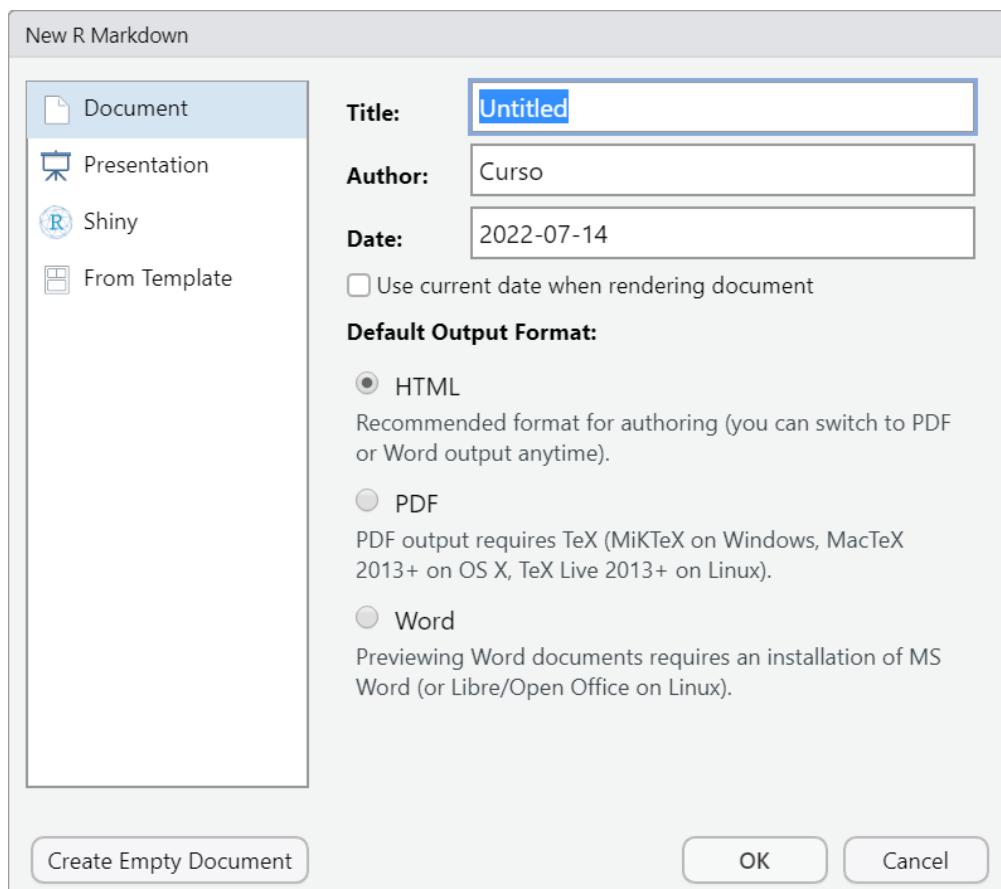
Observe que você fará acompanhando também as figuras abaixo:

**Figura 1: Tela para abrir um documento Rmarkdown.**



Após criar um documento Rmarkdown, abrirá em seu RStudio uma nova janela com opções de criação de **novos projetos**. Nesta janela são inseridos os dados básicos do documento conforme a Figura 2. Em seu computador insira o título, o nome do autor e a data do seu painel. Com isso configuraremos o nosso documento.

**Figura 2: Tela de criação de um documento Rmarkdown.**



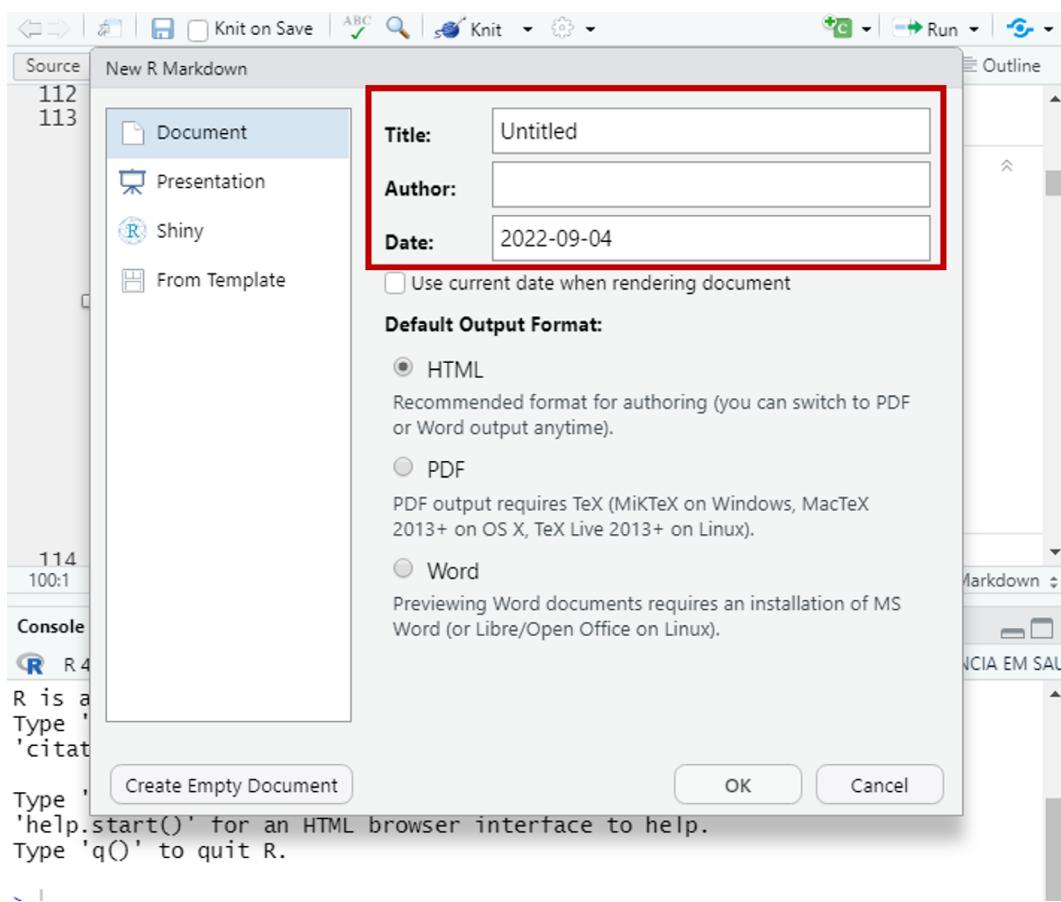
Observe que há um painel à esquerda da janela que se abriu. Nele, você terá disponível quatro opções:

- 1.** *Document* ou documento.
- 2.** *Presentation* ou apresentação.
- 3.** *Shiny* que é um pacote do R para criação de tabelas dinâmicas (animadas).
- 4.** *From Template* local em que você poderá selecionar um modelo pré-existente.

Selecione apenas a 1<sup>a</sup> opção: *Document* ou documento. Observe também que nesta janela você verá à direita três caixas (Figura 3). A primeira se refere ao título do seu novo documento, a segunda ao autor, e a terceira à data.

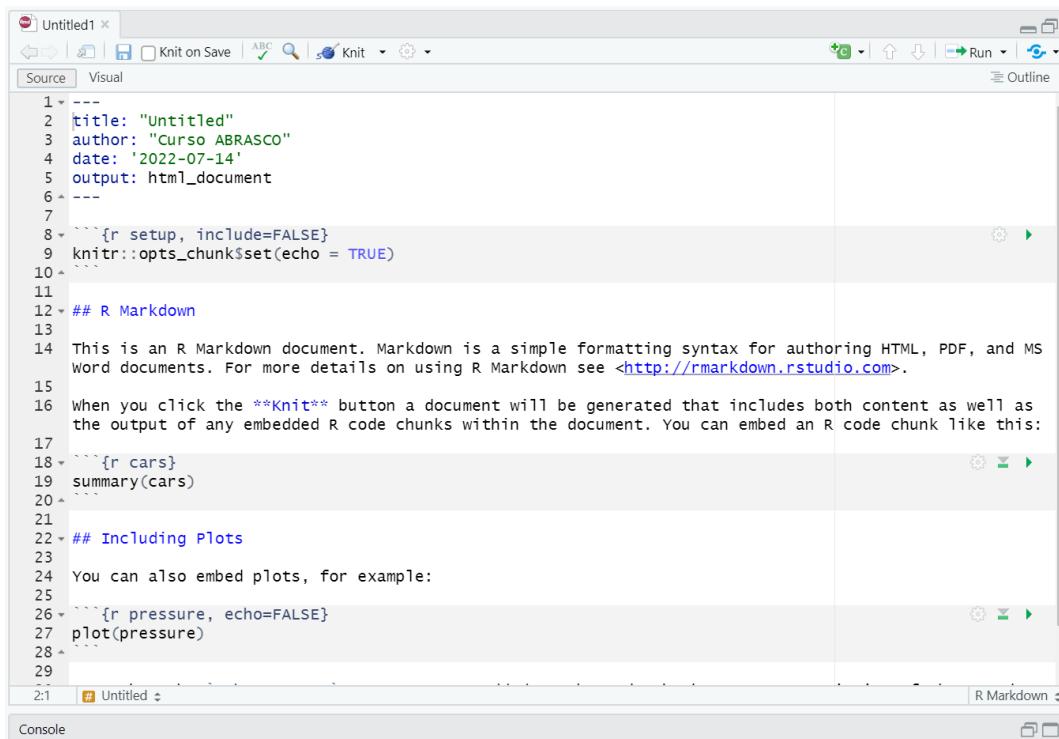


**Figura 3: Tela de configuração de um documento Rmarkdown.**



Observe que, abaixo destas caixas, existe a possibilidade de marcar uma de três alternativas, referentes ao formato do arquivo que será produzido: **HTML**, **PDF** ou **WORD**. Após escolher o tipo de arquivo, basta apenas selecionar **OK** e o programa irá abrir um novo arquivo conforme a Figura 4 com um “modelo básico de arquivo”. Você deverá obter em seu computador uma página de *script* como a apresentada na Figura 4:

**Figura 4: Tela com *script* gerado automaticamente após selecionar OK.**



```

1 ---  

2 title: "Untitled"  

3 author: "Curso ABRASCO"  

4 date: '2022-07-14'  

5 output: html_document  

6 ---  

7  

8 ````{r setup, include=FALSE}  

9 knitr::opts_chunk$set(echo = TRUE)  

10 +  

11  

12 ## R Markdown  

13  

14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS  

Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.  

15  

16 When you click the **Knit** button a document will be generated that includes both content as well as  

the output of any embedded R code chunks within the document. You can embed an R code chunk like this:  

17  

18 ````{r cars}  

19 summary(cars)  

20 +  

21  

22 ## Including Plots  

23  

24 You can also embed plots, for example:  

25  

26 ````{r pressure, echo=FALSE}  

27 plot(pressure)  

28 +  

29

```

Agora que o arquivo foi criado, nós devemos salvá-lo para dar início ao nosso relatório de análises. Para isso, digite as teclas **CTRL + S** do seu teclado ou selecione **File > Save** (Arquivo > Salvar) no menu superior do seu **RStudio**, e escolha um nome para o arquivo.

Você verá que agora o seu arquivo será salvo com uma extensão do tipo **.Rmd**, indicando por tanto que agora o *script* que você trabalhará se trata de um arquivo **Rmarkdown**. Para que fique um pouco mais claro, todo arquivo do tipo **Rmarkdown** possui quatro partes distintas: cabeçalho **YAML**, o código ou *chunk*, local para escrever o texto e mais linhas para escrita de código (*chunk*). Observe a Figura 5 e a Tabela 1 de detalhamento:



**Figura 5: Tela com *script* gerado automaticamente após selecionar OK.**

The screenshot shows the RStudio interface with an R Markdown file named "Relatorio\_novo.Rmd". The code editor displays the following R Markdown script:

```
1 ---  
2 title: "Untitled"  
3 author: "curso"  
4 date: '2022-07-14'  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ````  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS  
word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 When you click the **Knit** button a document will be generated that includes both content as well as  
the output of any embedded R code chunks within the document. You can embed an R code chunk like this:  
17  
18 ```{r cars}  
19 summary(cars)  
20 ````  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r pressure, echo=FALSE}  
27 plot(pressure)  
28 ````  
29
```

The code is annotated with four red boxes labeled 'a', 'b', 'c', and 'd' corresponding to specific sections of the R code:

- a**: Surrounds the YAML front matter (lines 1-6).
- b**: Surrounds the R code chunk setup (lines 8-10).
- c**: Surrounds the introductory text and the note about the Knit button (lines 12-16).
- d**: Surrounds the R code chunk for the 'cars' dataset (lines 18-20).

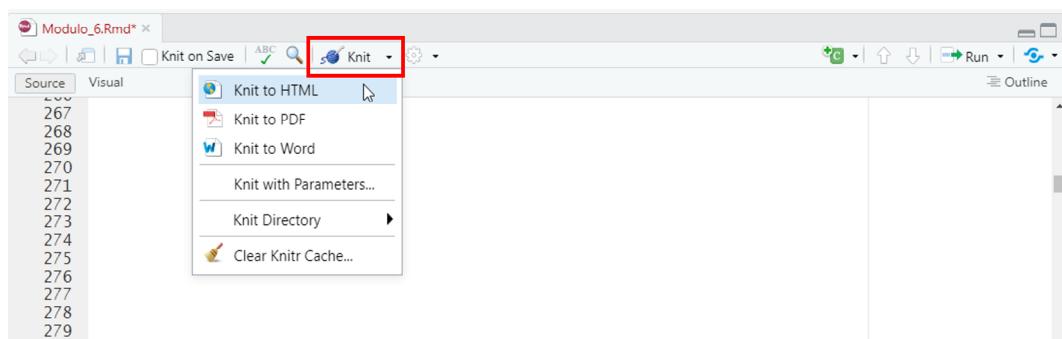
**Tabela 1: Formato básico de um arquivo Rmarkdown.**

Partes	Descrição
a	Um cabeçalho contendo os metadados do documento em formato YAML (sigla para <i>Yet Another Markup Language</i> ). Observe que esta parte do arquivo é delimitada por três hífens (---), e contém informações sobre o seu arquivo: título, autor, data, e formato de saída. Ele também poderá incluir informações de formato do arquivo final, como o tema de cores, ou a inclusão de logotipos.
b	Um trecho de código (em inglês chamado de <i>code chunk</i> ), marcado por uma região mais escura do código. Os trechos de código são delimitados sempre por três acentos graves, com a possibilidade de inclusão de uma linha de códigos na linha superior, sempre delimitados por chaves ({}). Este primeiro trecho de código, embora seja opcional, permite configurar diferentes aspectos do documento utilizando a linguagem R, conforme veremos mais adiante neste curso.
c	Um trecho contendo um título e texto. Esta parte do código segue a linguagem markdown, e permite a edição de títulos, textos e imagens que irão compor o relatório final.
d	Um outro trecho de código, também delimitado por um par de três acentos graves, e um pequeno código entre chaves após os três primeiros acentos graves. Esta parte do arquivo permite a execução de códigos em linguagem R para a realização de análises, produção de figuras e tabelas, que serão inseridos no relatório final. Conforme veremos adiante, é possível configurar estes trechos de diferentes formas: o código pode ser incluído ou oculto no momento de exportação do arquivo, assim como avisos, mensagens de erro, resultados de análises, imagens, entre outras saídas de um código em R.

Perceba que o seu arquivo `.Rmd` assim como na Figura 5 deverá possuir todas as instruções necessárias para a criação de um relatório de forma automática. A partir deste arquivo, o RStudio é capaz de **interpretar as instruções e converter** o script que você está escrevendo em um documento final. A este processo de transformação denominamos de **renderização** (*render*).

Para transformar um *script* em um arquivo `.pdf`, `.html`, ou `.docx`, ou seja, para renderizar seu código, você deverá selecionar o botão *knit* (do inglês, significa “tecer” ou “tricotar”), localizado na região superior do painel de códigos do RStudio conforme a Figura 6.

**Figura 6: Tela com a localização do botão *knit* para renderização de um *script*.**



Agora clique no botão *knit*. Assim você então renderiza seu documento criado (*script* no formato `.Rmd`) para o formato `.html`. Lembre-se que este é um formato para que seu relatório possa ser disponibilizado *online*, por isso optaremos por este formato. Ao realizar esta seleção o RStudio irá criar um novo arquivo para o formato desejado: `.html` e abrirá uma nova página dentro do próprio programa. Observe como ficaria uma página no formato para a publicação na *internet* (`.html`) na Figura 7.



**Figura 7: Tela com script renderizado para o formato .html.**

exemplo\_1.html | Open in Browser | Find | Publish ▾

# Untitled

Curso

2022-07-11

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

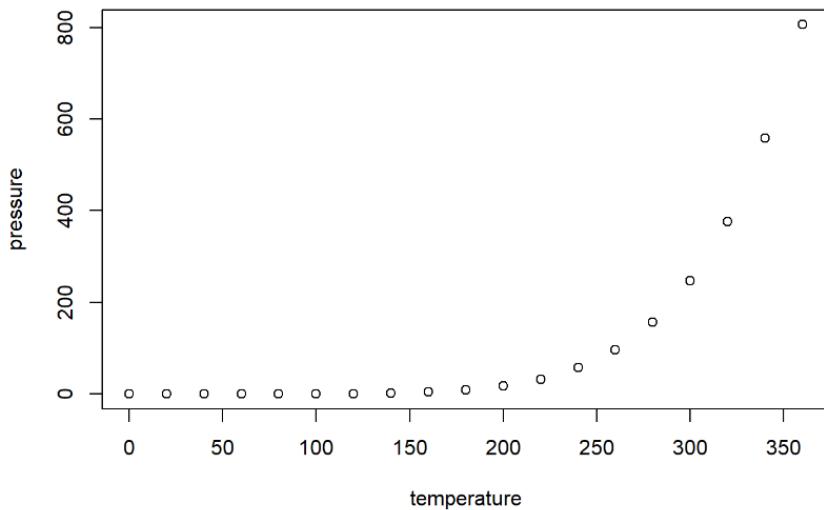
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed          dist
## Min.   : 4.0   Min.   :  2.00
## 1st Qu.:12.0   1st Qu.: 26.00
## Median :15.0   Median : 36.00
## Mean   :15.4   Mean   : 42.98
## 3rd Qu.:19.0   3rd Qu.: 56.00
## Max.   :25.0   Max.   :120.00
```

## Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Neste curso destacaremos o formato `.html`, pois ele será fundamental para a transformação do seu relatório em um painel ou *dashboard*. Você poderá a qualquer momento gerar o relatório no formato `.docx` ou `.pdf`, teste e experimente clicando nas opções apresentadas junto ao botão *knit* em seu *RStudio*.

Observe que apresentamos um modelo básico de relatório gerado utilizando o *Rmarkdown*. Agora é a hora de praticar!



Para saber mais sobre como construir um painel de indicadores de saúde você poderá consultar o curso “Construção de painéis (dashboards) para monitoramento de indicadores de saúde”. Caso deseje fazer a inscrição, clique [aqui](#).

### Atenção



Caso você tenha encontrado dificuldade de chegar a um arquivo com os *scripts* que utilizamos, não se preocupe e continue no curso!

Deixamos pronto para você um arquivo de estudo com todos os elementos que aplicamos nesta subseção: o `exemplo1.Rmd`. Você poderá encontrá-lo acessando o menu lateral “Arquivos” do Ambiente Virtual do curso e fazer o *download*.

## *3 Construindo um relatório automatizado*

Imagine que você é um profissional da vigilância epidemiológica do Estado de Rosas. Foi solicitado a você pelo secretário de saúde a atual situação da dengue no estado para uma entrevista com a imprensa em 4 horas. Você precisará criar com todas as análises necessárias **um relatório de avaliação de dengue no Estado de Rosas** (fictício) rapidamente. Com ele você dará informações da situação epidemiológica da dengue para o secretário de saúde e a sua avaliação da série histórica de casos. Para isso, você necessitará seguir alguns passos:

1. a instalação de pacotes específicos para a criação de um arquivo do tipo **.Rmd** (relatório),
2. a escrita de um *script* definindo as bases de dados que serão utilizadas,
3. os objetos e componentes que serão inclusos no relatório,
4. a estruturação do conteúdo a ser apresentado,
5. a padronização de um layout e
6. a transformação do *script* em uma página no formato **.html** (renderização).

Vamos lá! Até aqui já cumprimos a etapa de número 1. A partir do arquivo **.Rmd** que criamos, iremos selecionar todo o conteúdo a partir da linha 12 de seu *script* (Figura 8), e então deletá-lo para utilizarmos o relatório criado.

**Figura 8: Tela com textos que devem ser excluídos.**

```

1  ---
2  title: "Untitled"
3  author: "Curso"
4  date: '2022-07-11'
5  output: html_document
6  ---
7
8  ````{r setup, include=FALSE}
9  knitr::opts_chunk$set(echo = TRUE)
10
11
12 ## R Markdown
13
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more
15 details on using R Markdown see <http://rmarkdown.rstudio.com>.
16 When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R
17 code chunks within the document. You can embed an R code chunk like this:
18 ````{r cars}
19 summary(cars)
20
21
22 ## Including Plots
23
24 You can also embed plots, for example:
25
26 ````{r pressure, echo=FALSE}
27 plot(pressure)
28
29
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.
31

```

Conseguiu excluir? Se sim, agora poderemos utilizar este arquivo de base ou *template* original. Vamos deixá-lo apenas com o cabeçalho em YAML e o primeiro trecho de código (*chunk*). Em seguida, iremos personalizar as informações de “título” e “autor” do arquivo (Figura 9). Para isso, vá até as linhas 2 e 3 do *script* em seu computador e substitua o texto anterior pelas seguintes informações:

```
title: "Relatório Informativo sobre Dengue"  
author: "Vigilância Epidemiológica do Estado de Rosas"
```

Após estas mudanças, o seu arquivo `.Rmd` deve ficar semelhante como observamos na Figura 9 abaixo:

**Figura 9: Configurando o cabeçalho do relatório.**

```
1 -  
2 title: "Relatório Informativo sobre Dengue"  
3 author: "Vigilância Epidemiológica do Estado de Rosas"  
4 date: '2022-07-11'  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ``-  
11  
12
```

Pronto! Agora vamos adiante personalizando o nosso relatório que será automático.

Calma que as coisas vão seclareando aos poucos. Sigamos em frente.

### 3.1 Edição dos textos em markdown

Ao utilizarmos a linguagem `markdown` podemos simplesmente inserir parágrafos de texto em nosso `script`. Para isso, iremos sempre utilizar o espaço do documento `.Rmd` que está **fora dos trechos de códigos**, conforme apresentado na Figura 5. Nele poderemos personalizar o texto do relatório que criaremos, facilitando a organização e a leitura do conteúdo que será feita pelo secretário estadual de saúde de Rosas.

Vamos lá! Localize no seu `script` estes trechos e inclua algumas informações de introdução sobre o seu relatório. Neste momento precisamos incluir uma breve explicação sobre as análises que serão apresentadas e guiar o leitor quanto ao entendimento das informações. Copie e replique o trecho de código a seguir em seu `RStudio`:

O Departamento de Vigilância Epidemiológica do Estado de Rosas, por meio deste boletim epidemiológico (referência 1), apresenta informações gerais sobre a dengue (referência 2), assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou N casos confirmados de dengue e M óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentado na Figura 1. O número de casos por classificação final são apresentados na Tabela 1.

Pronto, você deverá obter em seu computador um arquivo (`.Rmd`) semelhante à Figura 10 que você visualiza abaixo:

**Figura 10: Tela com texto que introdutório do relatório informativo da dengue no arquivo `.Rmd`.**

```

1 ---
2 title: "Relatório Informativo sobre Dengue"
3 author: "Vigilância Epidemiológica do Estado de Rosas"
4 date: '2022-07-11'
5 output: html_document
6 ---
7
8 ````{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10 ````

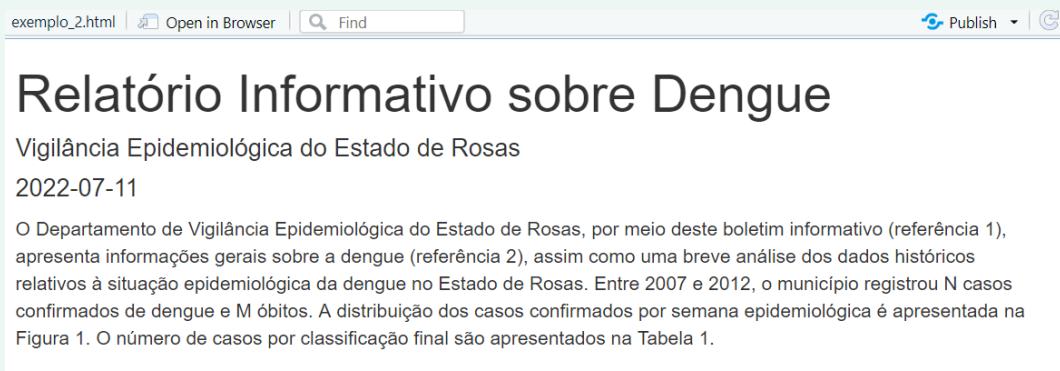
11
12 O Departamento de Vigilância Epidemiológica do Estado de Rosas, por meio deste boletim informativo (referência 1), apresenta informações gerais sobre a dengue (referência 2), assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou N casos confirmados de dengue e M óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na Figura 1. O número de casos por classificação final são apresentados na Tabela 1.
13
14
15

```



Para ver como fica o relatório com a inclusão deste parágrafo no formato *online*, aperte o botão *Knit*, no menu logo acima, e espere o **RStudio** renderizar o documento `*.Rmd`. Você verá uma janela com a seguinte apresentação:

**Figura 11: Tela com texto que introduz o  
relatório informativo da dengue no arquivo `.html`.**



The screenshot shows a browser window with the title 'exemplo\_2.html'. Below the title are buttons for 'Open in Browser' and 'Find'. On the right, there are 'Publish' and 'Print' options. The main content is a report titled 'Relatório Informativo sobre Dengue' from 'Vigilância Epidemiológica do Estado de Rosas' dated '2022-07-11'. It discusses the epidemiological situation of dengue in the state of Rosas from 2007 to 2012, mentioning N confirmed cases and M deaths. A summary sentence at the end refers to Figure 1 and Table 1.

Agora vamos personalizar o texto e torná-lo mais intuitivo.

## 3.2 Formatação de texto simples

A linguagem `markdown` possui uma maneira própria e, por isso, teremos que escrever de um jeito que o **R** nos entenda. Nós a utilizaremos como um editor de texto, assim como fazemos com o *libreoffice* ou *word*. Utilizando a linguagem `markdown` vamos incluir pequenas edições no texto para destacar pontos importantes. Para isso utilizaremos textos em **negrito**, em *italíco*, com ~~rashurado~~ ou outras formas de edição de texto que couberem melhor as necessidades do relatório que está sendo criado.

Veja na tabela abaixo como você poderá indicar estas edições em seu texto no arquivo `.Rmd` que estamos criando:

**Tabela 2: Formatação básica de texto para arquivo Rmarkdown.**

Formato	Descrição
Negrito	Definidos por um par de asteriscos (` <b>**Negrito**</b> `: <b>**Negrito**</b> ).
Itálico	Definidos pelo uso de asteriscos ao redor de um texto (` <i>*Itálico*</i> `: <i>*Itálico*</i> ) ou uma linha sublinha (` <u>_Itálico_</u> `: <u>_Itálico_</u> ).
Rasurado	Definidos pelo uso de um par de acentos til ao redor do texto (` <del>~~rasurado~~</del> `: <del>~~rasurado~~</del> )
Texto sobrescrito	Definidos por um par de acentos circunflexos (`X^3^`: X^3^).
Texto subescrito	definidos por um par de acentos til (`H~2~O` : H~2~O).
Código no texto	a marcação de códigos ao longo do texto é definida pelo uso de um par de acentos graves (` ` ` `). Veja o exemplo: <b>**`summary()`**:</b> `summary()`

Agora, na mesma região onde incluímos inicialmente a introdução do relatório, você irá destacar alguns trechos deste parágrafo utilizando alguns formatos da Tabela 2. Observe o texto a seguir e tente encontrar quais mudanças foram feitas:

O **\*\*Departamento de Vigilância Epidemiológica do Estado de Rosas\*\***, por meio deste boletim informativo<sup>1</sup>, apresenta informações gerais sobre a dengue<sup>2</sup>, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou N casos confirmados de dengue e M óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na **\*Figura 1\***. O número de casos por classificação final é apresentado na **\*Tabela 1\***.

Agora, vamos praticar. Inclua estas modificações (ou copie o código acima) e substitua o parágrafo que havíamos inseridos anteriormente com as mudanças necessárias. Você deverá obter um arquivo em seu computador semelhante ao da Figura 12.



**Figura 12: Script com texto em destaque no arquivo Rmarkdown.**

```
1 ---  
2 title: "Relatório Informativo sobre Dengue"  
3 author: "Vigilância Epidemiológica do Estado de Rosas"  
4 date: '2022-07-11'  
5 output: html_document  
6 ---  
7  
8 ````{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ````  
11  
12 O **Departamento de Vigilância Epidemiológica do Estado de Rosas**, por meio deste boletim informativo1,  
apresenta informações gerais sobre a dengue2, assim como uma breve análise dos dados históricos relativos à  
situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou N casos  
confirmados de dengue e M óbitos. A distribuição dos casos confirmados por semana epidemiológica é  
apresentada na *Figura 1*. O número de casos por classificação final são apresentados na *Tabela 1*.  
13
```

Note que algumas das alterações agora aparecem destacadas em seu código *markdown*! Mas atenção, você deve ter percebido que trocamos os termos (*referência 1*) por ^1^, e (*referência 2*) por ^2^. Estas marcações não aparecem destacados no nosso código, mas irão se refletir no texto renderizado para o formato `.html` na forma de números sobrescritos. Para visualizar como esta etapa se dá na prática, aperte *knit* no menu superior do seu *RStudio* e renderize o arquivo para o formato `.html` como a Figura 13.

**Figura 13: Renderização do script com texto em destaque.**

The screenshot shows a web browser window with the following content:

exemplo\_2.html | Open in Browser | Find | Publish | C

# Relatório Informativo sobre Dengue

Vigilância Epidemiológica do Estado de Rosas

2022-07-11

O Departamento de Vigilância Epidemiológica do Estado de Rosas, por meio deste boletim informativo<sup>1</sup>, apresenta informações gerais sobre a dengue<sup>2</sup>, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou N casos confirmados de dengue e M óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na *Figura 1*. O número de casos por classificação final são apresentados na *Tabela 1*.

Ficou bacana, não é mesmo?

### Atenção



Caso tenha encontrado dificuldade de chegar a um arquivo com os *scripts* que utilizamos, não se preocupe e continue no curso!

Deixamos pronto para você um arquivo de estudo com todos os elementos que aplicamos nesta subseção: o exemplo2.Rmd. Você poderá encontrá-lo acessando o menu lateral “Arquivos” do Ambiente Virtual do curso e fazer o *download*.

### 3.3 Formatação de títulos, subtítulos e listas

Quase sempre precisamos alterar a formatação de títulos, subtítulos, e até mesmo de lista de itens quando estavam fazendo relatórios. Você também conseguirá fazer estas edições com o R utilizando a linguagem `Rmarkdown`. Para criar títulos e subtítulos, basta você utilizar o símbolo `# (hashtag)` seguido do texto que nomeará o título. Todos os títulos subsequentes, com tamanhos de fonte menores, podem ser criados pelo uso progressivo de mais `#`.

É bem simples, vamos exemplificar. No espaço para `markdown`, você pode incluir títulos da seguinte forma seguindo um padrão progressivo das `# (hashtags)`:

```
# Título de nível 1  
  
## Título de nível 2  
  
### Título de nível 3  
  
#### Título de nível 4
```

Ficou mais claro, certo? Agora copie os códigos apresentados aqui em seu `RStudio` e veja como ficaria uma sequência de títulos no `Rmarkdown`. Você deve visualizar seu `chunk` como o apresentado na Figura 14.



**Figura 14:** Script com título editado em destaque no arquivo Rmarkdown.

```
1 ---  
2 title: "Tutorial R Markdown"  
3 author: "Curso ABRASCO"  
4 date: '2022-07-14'  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10  
11  
12 # Título de nível 1  
13  
14 ## Título de nível 2  
15  
16 ### Título de nível 3  
17  
18 #### Título de nível 4  
19
```

Em seguida, renderize o script clicando no botão *knit* e observe na Figura 15 como ficaria cada título em um arquivo do tipo `.html`.

**Figura 15:** Renderização do script com título editado.

# Título de nível 1

## Título de nível 2

### Título de nível 3

#### Título de nível 4

Além de títulos, é possível criar listas ordenadas de itens. Cada item é listado por um número seguido de ponto (ex: 1.), enquanto subitens são listados por números romanos seguidos de ponto (ex: i.) após uma *identação* (o pequeno espaço - recuo - que você observa entre o início da linha e os números romanos deste exemplo). A *identação* pode ser incluída ao se digitar a tecla TAB.

Observe abaixo como será o padrão que utilizaremos para construir o código para o nosso relatório sobre dengue em Rosas:

1. Primeiro item
2. Segundo item
  - i. Subitem 1
  - ii. Subitem 2
  - iii. Subitem 3

Agora, copie e replique o código em seu RStudio e observe a Figura 16. Veja o exemplo deste código na linguagem markdown:

**Figura 16: Script com itens no arquivo Rmarkdown.**

```
11
12 # Lista ordenada
13
14 1. Primeiro item
15
16 2. Segundo item
17
18   i. Subitem 1
19
20   ii. Subitem 2
21
22   iii. Subitem 3
23
```

Agora clique no botão *knit* e renderize o *script* para o formato .html. Os itens irão aparecer de forma semelhante ao apresentado na Figura 17.

**Figura 17:** Renderização do *script* com itens.

## Lista ordenada

1. Primeiro item
2. Segundo item
  - i. Subitem 1
  - ii. Subitem 2
  - iii. Subitem 3

Você também poderá criar itens de listas não-ordenadas (apenas com marcadores). Para isso, você deve utilizar o símbolo `*` (asterisco) para o item principal. Para adicionar subitens utilizar o símbolo `+` (mais) após dois espaços. Observe o código abaixo:

```
* Primeiro item

* Segundo item

  + Subitem 1

  + Subitem 2

  + Subitem 3
```

Agora pratique em seu arquivo `.Rmd`. Inclua estes itens no seu trecho de código *markdown*. É esperado que você visualize seu documento conforme a Figura 18.

**Figura 18:** Script com itens de listas não-ordenadas (apenas com marcadores).

```
11
12 # Lista não-ordenada
13
14 * Primeiro item
15
16 * Segundo item
17
18 + Subitem 1
19
20 + Subitem 2
21
22 + Subitem 3
23
```

Deu certo? Agora clique no botão *knit* no menu superior do seu RStudio e renderize seu documento para o formato `.html`. Você deverá obter um arquivo semelhante ao da Figura 19.

**Figura 19:** Renderização do script itens de listas  
não-ordenadas (apenas com marcadores).

## Lista não-ordenada

- Primeiro item
- Segundo item
  - Subitem 1
  - Subitem 2
  - Subitem 3

### Atenção



Caso tenha encontrado dificuldade de chegar a um arquivo com os *scripts* que utilizamos, não se preocupe e continue no curso!

Deixamos pronto para você um arquivo de estudo com todos os elementos que aplicamos nesta subseção: o `exemplo_títulos.Rmd`. Você poderá encontrá-lo acessando o menu lateral “Arquivos” do Ambiente Virtual do curso e fazer o *download*.



Certo, agora que você já conhece as diferentes formas de incluir títulos e listas com a linguagem *markdown*, vamos aplicar esse conhecimento em nosso boletim de vigilância no Estado de Rosas. Vamos lá!

1. Primeiro estruture o relatório inserindo seus títulos e subtítulos. Para isso, adicione as seguintes linhas de código após o parágrafo de introdução. Copie e replique em seu RStudio:

```
## Introdução

## Análises

### 1. Distribuição de casos por semana epidemiológica

### 2. Número de casos por classificação final
```

É esperado que você obtenha em sua tela códigos como os da Figura 20. Perceba que os títulos e subtítulos também aparecem com uma cor destacada em relação ao restante do texto:

**Figura 20: Script com título e itens editados do relatório de Rosas.**

```
1 ---  
2 title: "Relatório Informativo sobre Dengue"  
3 author: "Vigilância Epidemiológica do Estado de Rosas"  
4 date: '2022-07-11'  
5 output: html_document  
6 ---  
7  
8 ````{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ...  
11  
12 O **Departamento de Vigilância Epidemiológica do Estado de Rosas**, por meio deste boletim informativo^1^, apresenta informações gerais sobre a dengue^2^, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou N casos confirmados de dengue e M óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na *Figura 1*. O número de casos por classificação final são apresentados na *Tabela 1*.  
13  
14 ## Introdução  
15  
16 ## Análises  
17  
18 ### 1. Distribuição de casos por semana epidemiológica  
19  
20 ### 2. Número de casos por classificação final  
21
```

Agora renderize o seu documento e transforme-o em um arquivo do tipo `.html`. Observe se está semelhante ao apresentado aqui na Figura 21.

**Figura 21: Renderização do script com título e itens editados do relatório de Rosas.**

# Relatório Informativo sobre Dengue

Vigilância Epidemiológica do Estado de Rosas

2022-07-11

O Departamento de Vigilância Epidemiológica do Estado de Rosas, por meio deste boletim informativo<sup>1</sup>, apresenta informações gerais sobre a dengue<sup>2</sup>, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou  $N$  casos confirmados de dengue e  $M$  óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na *Figura 1*. O número de casos por classificação final são apresentados na *Tabela 1*.

## Introdução

## Análises

1. Distribuição de casos por semana epidemiológica

2. Número de casos por classificação final

2. Agora iremos incluir algumas listas ordenadas e não-ordenadas. Incluiremos no relatório sobre dengue um pequeno resumo de informações sobre a doença utilizando como referência o conteúdo técnico disponibilizado pelo [Ministério da Saúde](#).

Observe o *script* abaixo e replique-o em seu RStudio:

### 1. O que é dengue

A dengue é a arbovirose urbana mais prevalente nas Américas, principalmente no Brasil. É uma doença febril que tem se mostrado de grande importância em saúde pública nos últimos anos. O vírus dengue (DENV) é um arbovírus transmitido pela picada da fêmea do mosquito *Aedes aegypti* e possui quatro sorotipos diferentes (DENV-1, DENV-2, DENV-3 e DENV-4).

#### i. Principais Sintomas

- \* Febre alta > 38°C.
- \* Dor no corpo e articulações
- \* Dor atrás dos olhos.
- \* Mal estar.
- \* Falta de apetite.
- \* Dor de cabeça.
- \* Manchas vermelhas no corpo.

#### ii. Transmissão

O vírus da dengue (DENV) pode ser transmitido ao homem principalmente por via vetorial, pela picada de fêmeas de *Aedes aegypti* infectadas, no ciclo urbano humano-vetor-humano. Os relatos de transmissão por via vertical (de mãe para filho durante a gestação) e transfusional são raros.

#### iii. Diagnóstico

##### \* Métodos diretos

- + Pesquisa de vírus (isolamento viral por inoculação em células);
- + Pesquisa de genoma do vírus da dengue por transcrição reversa seguida de reação em cadeia da polimerase (RT-PCR);

##### \* Métodos indiretos

- + Pesquisa de anticorpos IgM por testes sorológicos (ensaio imunoenzimático - ELISA)
- + Teste de neutralização por redução de placas (PRNT);
- + Inibição da hemaglutinação (IH);
- + Pesquisa de antígeno NS1 (ensaio imunoenzimático - ELISA);
- + Patologia: estudo anatomo-patológico seguido de pesquisa de antígenos virais por imuno-histoquímica (IHQ).

Observe que você deverá incluir símbolos que destaquem tanto as listas ordenadas quanto as não-ordenadas. Além disso, o nome científico do mosquito da dengue (*Aedes aegypti*) será marcado com a grafia para itálico (\_Aedes aegypti\_) quando avaliarmos necessário. Percebeu a diferença?

3. Em um terceiro passo para que estas informações apareçam no boletim que estamos produzindo, copie o texto deste arquivo para o seu RStudio incluindo-o logo **após o título de Introdução**.
4. Por fim, peça para que o RStudio gere o boletim no formato .html. Para isso clique no botão *knit* e gere o documento. Você deverá visualizar um arquivo como o da Figura 22.

### **Figura 22: Renderização do script com conteúdo de dengue.**

## Introdução

### 1. O que é Dengue

A dengue é a arbovirose urbana mais prevalente nas Américas, principalmente no Brasil. É uma doença febril que tem se mostrado de grande importância em saúde pública nos últimos anos. O vírus dengue (DENV) é um arbovírus transmitido pela picada da fêmea do mosquito *Aedes aegypti* e possui quatro sorotipos diferentes (DENV-1, DENV-2, DENV-3 e DENV-4).

#### i. Principais Sintomas

- Febre alta > 38°C.
- Dor no corpo e articulações
- Dor atrás dos olhos.
- Mal estar.
- Falta de apetite.
- Dor de cabeça.
- Manchas vermelhas no corpo.

#### ii. Transmissão

O vírus da dengue (DENV) pode ser transmitido ao homem principalmente por via vetorial, pela picada de fêmeas de *Aedes aegypti* infectadas, no ciclo urbano humano–vetor–humano. Os relatos de transmissão por via vertical (de mãe para filho durante a gestação) e transfusional são raros.

#### iii. Diagnóstico

- Métodos diretos
  - Pesquisa de vírus (isolamento viral por inoculação em células);
  - Pesquisa de genoma do vírus da dengue por transcrição reversa seguida de reação em cadeia da polimerase (RT-PCR);
- Métodos indiretos
  - Pesquisa de anticorpos IgM por testes sorológicos (ensaio imunoenzimático – ELISA)
  - Teste de neutralização por redução de placas (PRNT);
  - Inibição da hemaglutinação (IH);
  - Pesquisa de antígeno NS1 (ensaio imunoenzimático – ELISA);
  - Patologia: estudo anatomo-patológico seguido de pesquisa de antígenos virais por imuno-histoquímica (IHQ).

### Atenção



Caso tenha encontrado dificuldade de chegar a um arquivo com os *scripts* que utilizamos, não se preocupe e continue no curso!

Deixamos pronto para você um arquivo de estudo com todos os elementos que aplicamos nesta subseção: o `exemplo3.Rmd`. Você poderá encontrá-lo acessando o menu lateral “Arquivos” do Ambiente Virtual do curso e fazer o *download*.

## 3.4 Inserir links (internos e externos)

Utilizando a linguagem *markdown* no R você também poderá incluir *links* ou *hiperlinks* no próprio documento. Esses *links* serão clicados e irão abrir o endereço de *internet* inserido em um navegador do computador. Estes *hyperlinks* podem ser inseridos no documento pelo uso do texto entre `[]` (colchetes) seguido do link entre `()` (parênteses) assim: `[texto](link)`.

Observe na Figura 23 como ficaria a escrita incluindo um *link* no seu arquivo do tipo `.Rmd`.

**Figura 23: Tela de visualização de script com hyperlink.**

```
11
12 site de [R Markdown] (https://rmarkdown.rstudio.com/)
13
```

Você poderá escrever qualquer texto gerando acesso a arquivos externos em seu documento. Acompanhe mais um exemplo:

site de [Rmarkdown](#)

Agora vamos praticar! Em nosso relatório de dengue poderemos incluir um *link* indicando um acesso externo que contém o material que utilizamos de referência. Observe o código abaixo e copie em seu RStudio:

<sup>^1</sup> Este relatório foi produzido utilizando a linguagem ``Rmarkdown``.

<sup>^2</sup> As informações sobre esta doença foram baseadas em conteúdo disponibilizado pelo Ministério da Saúde. Para obter mais informações, acesse [este link](<https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/dengue>).

Observe que neste trecho colocamos o número da referência em sobrescrito (com uso dos marcadores ^), e também deixamos sobrescrito o *link* para o site do Ministério da Saúde. Perceba que também utilizamos `` (aspas) para colocar o termo “Rmarkdown” em formato de código.

Faça o teste, em seu arquivo .Rmd. Copie o código acima e o inclua **antes do título “## Introdução”**, conforme a Figura 24.

**Figura 24: Script com códigos contendo *link*.**

```

1  ---
2  title: "Relatório Informativo sobre Dengue"
3  author: "Vigilância Epidemiológica do Estado de Rosas"
4  date: '2022-07-11'
5  output: html_document
6  ---
7
8  ```{r setup, include=FALSE}
9  knitr::opts_chunk$set(echo = TRUE)
10 ```

11 O **Departamento de Vigilância Epidemiológica do Estado de Rosas**1, por meio deste boletim informativo1, apresenta
12 informações gerais sobre a dengue2, assim como uma breve análise dos dados históricos relativos à situação
13 epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou N casos confirmados de dengue e
14 M óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na *Figura 1*. O número de casos
15 por classificação final são apresentados na *Tabela 1*.
16 1 Este relatório foi produzido utilizando a linguagem `RMarkdown`.
17 2 As informações sobre esta doença foram baseadas em conteúdo disponibilizado pelo Ministério da Saúde. Para obter mais
18 informações, acesse [este link](https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/dengue).
19 ## Introdução
20 1. O que é Dengue
21

```

Você percebeu como foi simples? Agora, renderize seu script clicando no botão *knit* e gere um arquivo em formato .html como o apresentado na Figura 25.

### Figura 25: Renderização do script com link.

## Relatório Informativo sobre Dengue

Vigilância Epidemiológica do Estado de Rosas

2022-07-11

O Departamento de Vigilância Epidemiológica do Estado de Rosas, por meio deste boletim informativo<sup>1</sup>, apresenta informações gerais sobre a dengue<sup>2</sup>, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou  $N$  casos confirmados de dengue e  $M$  óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na Figura 1. O número de casos por classificação final são apresentados na Tabela 1.

<sup>1</sup> Este relatório foi produzido utilizando a linguagem RMarkdown.

<sup>2</sup> As informações sobre esta doença foram baseadas em conteúdo disponibilizado pelo Ministério da Saúde. Para obter mais informações, acesse [este link](#).

### Introdução

1. O que é Dengue

Clicando sobre o trecho [este link](#), você verá que seu navegador padrão irá abrir o *link* que incluímos no relatório de dengue de Rosas.

#### Atenção



Caso tenha encontrado dificuldade de chegar a um arquivo com os *scripts* que utilizamos, não se preocupe e continue no curso!

Deixamos pronto para você um arquivo de estudo com todos os elementos que aplicamos nesta subseção: o exemplo4.Rmd. Você poderá encontrá-lo acessando o menu lateral “Arquivos” do Ambiente Virtual do curso e fazer o *download*.

## 4 Incluindo imagens ao relatório

Além de editar os textos e tornar seu relatório bonito e bem estruturado, é possível incluir imagens junto aos *scripts* do R. O Rmarkdown permite que figuras prontas possam ser inseridas em meio ao texto do documento. Estas imagens podem vir de duas fontes: a partir de um arquivo local, ou a partir de um endereço da internet (*online*): a URL.

Todas as vezes em que houver a necessidade de incluir **uma imagem local**, você deverá escrevê-la conforme apresentado na Figura 26.

**Figura 26: Script indicando uma inserção de imagem local.**

```
11
12 ! [texto opcional] (pasta/imagem.jpg)
13
```

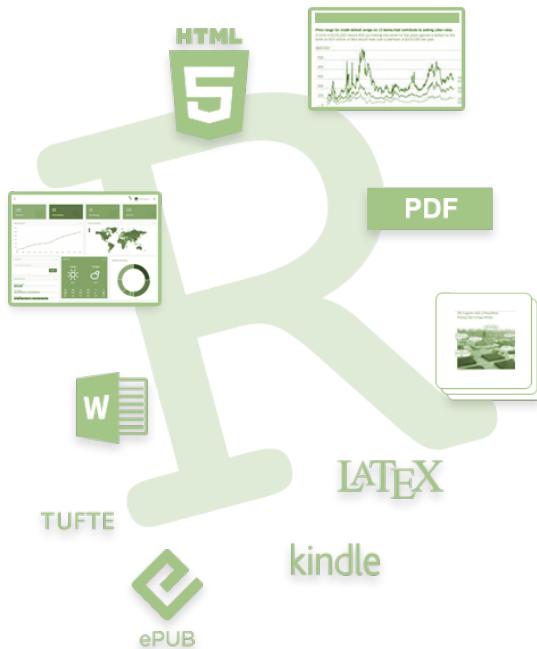
Como você pode visualizar na Figura 26, para incluir uma imagem primeiramente você deverá conhecer o caminho (diretório ou pasta) onde o seu arquivo .Rmd está localizado e indicar para o R onde a imagem que você deseja incluir está localizada também.

Como boa prática de organização, nós recomendamos a criação de uma pasta exclusiva para armazenar as imagens que serão inseridas no seu documento. No exemplo a seguir, criamos uma pasta denominada **Imagens**. Nela inserimos um arquivo nomeado como **mod\_7\_exemploimagem.png**. Copie o código abaixo em seu arquivo .Rmd e observe o que acontece:

```
![Primeiro exemplo](Imagens/mod_7_exemploimagem.png)
```

Em seu RStudio você deverá obter uma visualização como o da Figura 27.

**Figura 27: Imagem local adicionada ao script.**



**Figura 27: Primeiro exemplo**

Agora, quando queremos importar **uma imagem da internet** devemos inserir o endereço da imagem, e utilizar uma sintaxe como a apresentada na Figura 28.

### Figura 28: Script indicando uma inserção de imagem local.

```
11  
12 ! [texto opcional] (link para imagem)  
13
```

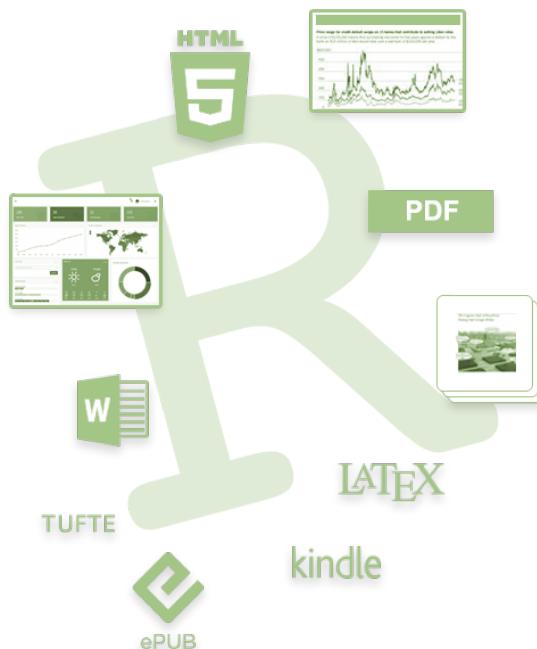
Agora inclua em seu documento Rmarkdown uma imagem que pode ser acessada pelo url <https://tinyurl.com/rmarkdownimagem>. Para isso digite o seguinte código em seu RStudio:

```

```

Você deverá obter uma visualização como a apresentada na Figura 29.

### Figura 29: Imagem da internet adicionada ao script.



## 4.1 Formatando imagens

Agora que você já consegue inserir imagens que estão em seu computador e outras que encontrou na *internet*, você deve editar o seu alinhamento no arquivo. É possível alterar a posição dessas imagens no texto e centralizá-las ou alterar o seu tamanho.

Por padrão qualquer imagem inserida em um documento .Rmd é automaticamente alinhada à esquerda. Para centralizar a imagem devemos incluir os códigos <center> antes da imagem e o </center> após o código da imagem. Observe na Figura 30 a seguir um exemplo e replique-o em seu Rstudio:

**Figura 30: Script com imagem centralizada no texto.**

```
12 <center>
13 ! [] (Imagens/bases.png)
14 </center>
```

Agora, observe na Figura 31 como a imagem fica quando o script é renderizado.

**Figura 31: Renderização do script com imagem centralizada no texto.**



Mas se você desejar alterar o tamanho de uma imagem? É possível especificar a sua largura no arquivo Rmarkdown. Para isso, você precisará inserir o comando {width= "10%"}, sendo que o valor indicado por porcentagem determinará o ajuste de tamanho da imagem. Assim, você deve sempre ajustar o valor de porcentagem desejado. Teste e experimente, pois não há um padrão pré-determinado.

Observe na Figura 32 como seria escrever a configuração de tamanho da imagem em seu documento Rmarkdown.

### **Figura 32: Script com inserção de imagem e edição da proporção de tamanho.**

```
11
12 ! [](Imagens/bases.png){width="10%"}  
13
```

Copie o código que você visualizou na Figura 32 em seu RStudio e renderize o arquivo. Observe na Figura 33 como deve ficar o arquivo final (.html).

### **Figura 33: Renderização do script com inserção de imagem e edição da proporção de tamanho.**



Para fazer o ajuste específico da altura da imagem, basta trocar o termo `width` (largura) por `height` (altura). Experimente

Outra configuração possível é especificar o tamanho da imagem em unidade pixels (px). Você poderá definir o tamanho pela seguinte nomenclatura: `{width= "100px"}`, conforme na Figura 34. Também será importante que você teste e experimente diversos tamanhos, pois não há um padrão pré-determinado.

### Figura 34: Script com inserção de imagem e edição de tamanho em pixel.

```
11
12 ! [](Imagens/bases.png){width="100px"}
13
```

Copie e cole o código da Figura 34 em seu computador e renderize o código. Você deverá obter uma imagem em seu documento final semelhante ao apresentado na Figura 35.

### Figura 35: Renderização do script com inserção de imagem e edição de tamanho em pixel.



Faça este exercício: crie uma pasta com o nome “Imagens” em seu computador e inclua nela o arquivo “dengue.png” disponibilizado junto ao menu lateral “Arquivos” deste curso.

Pronto. Já poderemos inserir uma imagem de ilustração em nosso relatório de vigilância em saúde. Para isso você deverá criar uma pasta dentro do diretório onde está localizado o arquivo `.Rmd` que estamos trabalhando e nomeá-la de “**Imagens**”. Como boa prática de organização, o uso desse nome facilita que os arquivos de um mesmo tipo sejam facilmente encontrados. Caso você opte em outros projetos de utilizar um nome diferente para o diretório de imagens, lembre-se de adequar o caminho correto das pastas na hora de escrever o seu código. Caso contrário, a imagem não será visualizada.

Agora, após o texto do parágrafo inicial de introdução, vamos incluir o código abaixo junto ao *script* do relatório de Rosas. Copie-o e execute-o em seu arquivo `.Rmd`:

```
! [](Imagens/dengue.png){width="100%"}  
[43]
```

Seguindo estas instruções, você deverá obter um *script* conforme a presentado na Figura 36.

**Figura 36:** Script incluindo imagem do mosquito da dengue.

```
1 ---  
2 title: "Relatório Informativo sobre Dengue"  
3 author: "Vigilância Epidemiológica do Estado de Rosas"  
4 date: '2022-07-11'  
5 output: html_document  
6 ---  
7  
8 ``{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ``  
11  
12 O **Departamento de Vigilância Epidemiológica do Estado de Rosas**1, por meio deste boletim informativo1, apresenta informações gerais sobre a dengue2, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou N casos confirmados de dengue e M óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na *Figura 1*. O número de casos por classificação final são apresentados na *Tabela 1*.  
13  
14 {width="100%"}  
15  
16 ^1 Este relatório foi produzido utilizando a linguagem `RMarkdown`.  
17  
18 ^2 As informações sobre esta doença foram baseadas em conteúdo disponibilizado pelo Ministério da Saúde. Para obter mais informações, acesse [este link](https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/d/dengue).  
19  
20 ## Introdução
```

Observe que o código indicando a Figura 36 foi inserido na linha 14. Agora vamos renderizar o documento clicando em *knit* para que possamos obter um arquivo como o apresentado na Figura 37.

**Figura 37: Renderização do *script* incluindo imagem do mosquito da dengue.**

## Relatório Informativo sobre Dengue

Vigilância Epidemiológica do Estado de Rosas

2022-07-11

O Departamento de Vigilância Epidemiológica do Estado de Rosas, por meio deste boletim informativo<sup>1</sup>, apresenta informações gerais sobre a dengue<sup>2</sup>, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou  $N$  casos confirmados de dengue e  $M$  óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na *Figura 1*. O número de casos por classificação final são apresentados na *Tabela 1*.



<sup>1</sup> Este relatório foi produzido utilizando a linguagem [RMarkdown](#).

<sup>2</sup> As informações sobre esta doença foram baseadas em conteúdo disponibilizado pelo Ministério da Saúde. Para obter mais informações, acesse [este link](#).

### Introdução

1. O que é Dengue



#### Atenção

Caso tenha encontrado dificuldade de chegar a um arquivo com os *scripts* que utilizamos, não se preocupe e continue no curso!

Deixamos pronto para você um arquivo de estudo com todos os elementos que aplicamos nesta subseção: o exemplo5.Rmd. Você poderá encontrá-lo acessando o menu lateral “Arquivos” do Ambiente Virtual do curso e fazer o *download*.

## 5 Incluindo códigos ao relatório

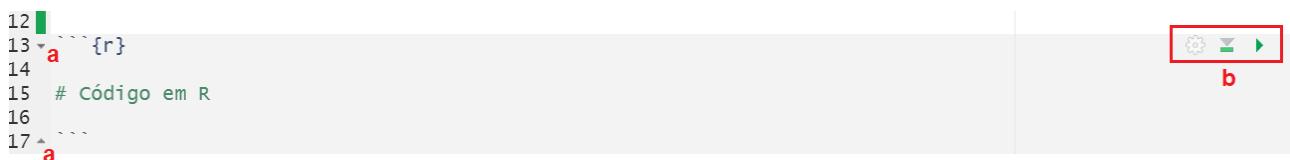
O grande diferencial dos arquivos escritos em Rmarkdown é a possibilidade de incluir trechos de códigos (em inglês: *code chunks*) no texto. Esta etapa é fundamental para que nosso relatório possua dados, que podem ser reproduzíveis e sempre atualizados. Dentro dos *chunks* iremos inserir códigos que produzam análises, tabelas e figuras, tornando nosso documento de texto mais interessante e complexo. Nesta subseção, iremos aprender a mesclar esses trechos de códigos em nossos documentos, personalizá-los. Assim, iremos transformar nosso arquivo .Rmd em um arquivo que poderá ser enviado para o secretário de saúde do Estado de Rosas, para a imprensa e até mesmo disponibilizado de forma *online* em um *website*.

Vamos lá!

Até aqui vimos que os *chunks* são trechos para inclusão de código em R. Esses códigos deverão ser delimitados no início da sua escrita por três acentos graves ( ` ` ` ) seguido do código {r} (uma letra "r" minúscula entre colchetes). Para encerrar a inserção dos códigos basta novamente você incluir os três acentos graves ( ` ` ` ). Elas irão indicar a finalização do trecho de código.

Pareceu difícil? Calma, iremos detalhar passo a passo nesta seção do curso. Lembre-se que no seu texto o *chunk* será indicado por uma área sombreada (Figura 38). Essa área avisa que o código a ser interpretado para o relatório está na linguagem R.

**Figura 38: Área do script para inserção de códigos**



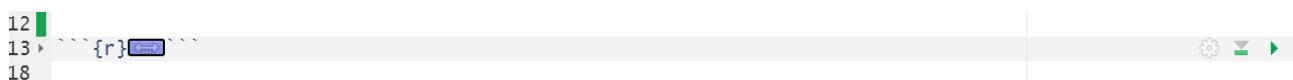
A screenshot of an RStudio code editor. The code in the script pane is:

```
12 
13 a`{r}
14 # Código em R
15 
16 a`
```

The line 'a`{r}' is highlighted with a red box and labeled 'a'. The line '# Código em R' is also highlighted with a red box and labeled 'b'. The status bar at the bottom right shows the number 46.

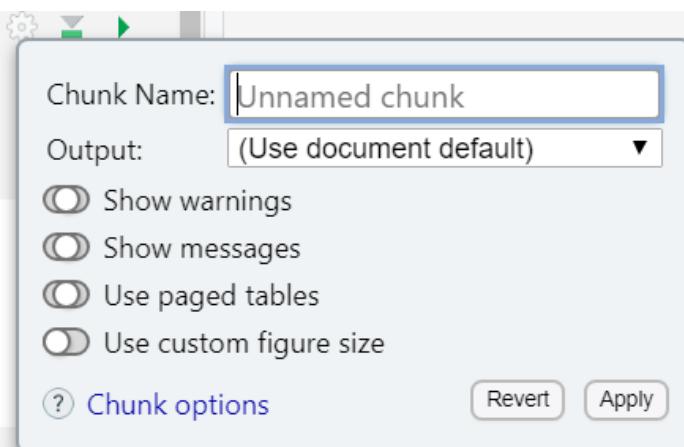
Tente se lembrar da Figura 5 no início deste curso ou volte a unidade dois do curso e observe o *chunk* destacado em **a**. Ao clicar nas pequenas setas ao lado do número das linhas no RStudio (item **b** do desenho), você ocultará todo o código neste trecho. Veja na Figura 39 como podemos facilitar a escrita do documento como um todo quando estamos escrevendo um arquivo **.Rmd**. A possibilidade de ocultar o *chunk* é um recurso interessante que pode facilitar a visualização do documento.

**Figura 39: Ocultando o *chunk* do script Rmarkdown.**



Agora observe também que no item **b** da Figura 38 você visualiza três símbolos diferentes. Ao clicar na roda dentada (Figura 40), o RStudio irá abrir um pequeno menu com opções para este código. Observe e replique esta etapa em seu computador.

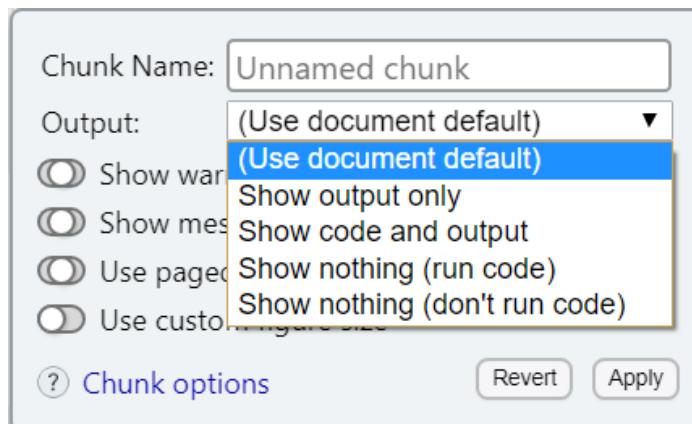
**Figura 40: Configurando o *chunk* em um script Rmarkdown.**



Neste menu é possível nomear o trecho de código em *Chunk Name*, se assim desejar. Caso decida nomear estes trechos de códigos, faça com atenção pois para que o *chunk* funcione sem erros não poderá haver nomes repetidos. Se houver um nome repetido o R irá apresentar erro no momento de renderização do arquivo, lhe impedindo de gerar o documento final.

Agora observe a caixa de nome *output* na Figura 40. Ela permitirá que você escolha cinco possibilidades para configuração de exibição do código no momento da renderização para o documento final. Observe esta etapa na Figura 41 e a descrição das opções na Tabela 3 abaixo.

**Figura 41: Configurando o output do *chunk* em um *script Rmarkdown*.**



**Tabela 3: Configurando o output do *chunk* em um *script Rmarkdown*.**

Opção	Descrição
Use documento default	Irá utilizar as mesmas configurações padrões de todo o documento, definida no primeiro trecho de código do arquivo.
Show output only	O código não será mostrado no documento final, apenas o seu <i>output</i> , equivalente ao que aparecia no console do RStudio.
Show code and output	O documento final irá mostrar tanto o código quanto o seu <i>output</i> .
Show nothing (run code)	Apenas executa o código, sem que ele ou o seu <i>output</i> apareçam no documento final.
Show nothing (don't run code)	Não executa o código, e o código também não irá ser incluído no documento final.

Ao selecionar cada uma dessas opções, você irá observar que um novo **argumento** irá aparecer na primeira linha do seu trecho de código (ao lado do trecho entre colchetes `{r}`). Por exemplo, após incluirmos o nome para este trecho de código como “**código 1**”, e selecionar “**Show output only**”, obteremos um *chunk* semelhante à Figura 42. Repita esta etapa em seu RStudio e visualize em seu computador.



### **Figura 42: Configurando "Show output only"\*\* do *chunk* em um *script Rmarkdown*.\*\***

```
11 ````{r código 1, echo=FALSE}
12 # Código em R
13
14 ...
15
16 ````
```

Agora vamos utilizar esses argumentos para customizar como nossos trechos de *chunks* serão inseridos no documento final (renderizado). Lembre-se, que a estrutura entre as chaves deverá seguir a seguinte lógica: `{r nome do trecho de código, argumento 1, argumento 2, ...}`.

- O **echo**: Indica se o código irá aparecer no documento final, enquanto sempre mostra o *output*. O valor do **echo**, por padrão, permanece como sendo igual a `TRUE`. Neste caso, o código e o *output* são apresentados no documento final. Observe o código no arquivo `.Rmd` conforme Figura 43:

### **Figura 43: Configurando echo\*\* escolhida para o *chunk* em um *script Rmarkdown*.\*\***

```
12 ````{r, echo = TRUE}
13
14 print(1 + 1)
15
16 ````
```

Agora veja como visualizamos em um arquivo renderizado conforme Figura 44. Replique esta etapa em seu `RStudio` e renderize seu código.

### **Figura 44: Renderizando a configuração echo = TRUE\*\* escolhida para o *chunk* em um *script Rmarkdown*.\*\***

```
print(1 + 1)
```

```
## [1] 2
```

Fica bem interessante, não é mesmo? Agora, se você optar para o valor de `echo = FALSE`, apenas o *output* é mostrado no documento final, e o código em `R` fica oculto, conforme a Figura 45.



**Figura 45: Renderizando a configuração echo = FALSE\*\*  
escolhida para o *chunk* em um script Rmarkdown.\*\***

```
## [1] 2
```

O **include**: Indica se o código irá aparecer no documento final. Por padrão, o seu valor é igual a `TRUE`. Neste caso, o código será mostrado como o exemplo na Figura 46. Veja esta configuração no arquivo `.Rmd`:

**Figura 46: Configurando include = TRUE\*\*  
escolhida para o *chunk* em um script Rmarkdown.\*\***

```
12 ~~~{r, include = TRUE}
13
14 print(1 + 1)
15
16 ~~~
```

Agora veja como visualizamos em um arquivo renderizado (Figura 47). Replique o código desta etapa em seu RStudio e renderize-o.

**Figura 47: Renderizando a configuração include = TRUE\*\*  
escolhida para o *chunk* em um script Rmarkdown.\*\***

```
print(1 + 1)
```

```
## [1] 2
```

Quando definido o argumento **include** igual a `FALSE`, ele irá ocultar do documento final **o código em R e o seu output**. Note que este comportamento se diferencia do comando `echo`, que quando definido como `FALSE`, omite apenas o código, mas apresenta o `output` no documento final.



Embora, tenha sido possível definir se o código ou *output* poderiam ser ocultados ou não utilizando os comandos anteriores. É importante que você saiba que os códigos dentro do trecho sempre são executados. O que muda é se iremos ou não visualizá-lo quando o documento é renderizado.

- O **eval**: Indica se o código deve ser executado (Figura 48). Com o comando **eval** é possível escolher se o código deverá ser executado no momento em que o documento é renderizado (gerado) ou não. Por padrão, o seu valor é igual a `TRUE`. Veja como fica o código no arquivo `.Rmd` visualizado na Figura 48:

**Figura 48: Configurando eval = TRUE\*\* escolhida para o chunk em um script Rmarkdown..\*\***

```
12  ``{r eval = TRUE}
13
14  print(1 + 1)
15
16  ...
```

Por sua vez, é possível também configurar o **eval** e definir que o código apareça no documento final, mas sem que seja executado conforme a Figura 49 dando valor `FALSE` a ele.

**Figura 49: Renderizando eval = FALSE\*\* escolhida para o chunk em um script Rmarkdown..\*\***

```
12 Instruções de como somar valores no R:
13
14  ``{r eval = FALSE}
15
16  print(1 + 1)
17
18  ...
```

Agora veja como visualizamos em um arquivo renderizado conforme Figura 50. Replique o código desta etapa em seu **RStudio** e renderize-o.

**Figura 50: Renderização Renderizando eval = FALSE\*\*  
escolhida para o *chunk* em um *script R* markdown.\*\***

Instruções de como somar valores no R:

```
print(1 + 1)
```

Além destes comandos, é possível definir se mensagens do console irão ser mostradas no documento final, conforme a Tabela 4:

**Tabela 4: Argumentos de configuração das mensagens  
do console no *script* em um *script R* markdown.**

Argumento	Padrão	Descrição
warning	TRUE	Define se avisos ( <i>warnings</i> ) serão exibidos
message	TRUE	Define se mensagens geradas pelo código serão exibidas
error	TRUE	Define se mensagens de erro serão exibidas

Ainda temos alguns trechos de código para detalhar no arquivo `.Rmd` então vamos lá. Continuaremos aprendendo sobre os componentes presentes no destaque **b** da Figura 38. Você verá o ícone **RUN**: o símbolo .

Este símbolo irá executar na própria sessão do `RStudio` o código presente no trecho de código escolhido. Esta função facilita com que o *script* em R digitado possa ser testado antes da geração do documento final. Veja na Figura 51 como seria executar um código no *chunk*.

**Figura 51: Tela de executando do *chunk* em um *script R* markdown.**



```

23 ~~~{r}
24
25 print(1 + 2)
26 ~~~
27 ~~~
28 [1] 3
29

```

The screenshot shows an RStudio interface. A code chunk is running, and its output, '[1] 3', is displayed in a separate window below the code editor. The window has a red border around its title bar and close button.

Perceba que um novo quadro aparece abaixo do trecho de código. Nas Figuras 52 A, 52 B e 52 C, você verá três ícones em vermelho com algumas opções, veja:

**Figura 52 A: Ícone do *chunk* em uma nova janela.**



: abre o *output* em uma nova janela.

**Figura 52 B: Ícone do *chunk* oculta os detalhes do quadro de *output*.**



: oculta os detalhes do quadro de *output*.

**Figura 52 C: Ícone do *chunk* fecha o quadro *output*.**



: fecha o quadro *output*.

Por fim, o símbolo  executa todos os trechos de códigos anteriores ao *chunk* atual. Esta função pode ser interessante caso você deseje iniciar a execução do código a partir de um determinado ponto do documento, sem a necessidade de rodar os trechos de códigos anteriores um a um. Você os rodará em bloco.

## 5.1 Escrevendo os códigos de um relatório

Agora que entendemos como editar os *chunks*, vamos preparar nosso relatório para gerar gráficos e tabelas a partir de códigos em R. O primeiro passo é organizarmos o nosso ambiente de análise nos seguintes passos:

1. Carregar os pacotes que serão utilizados.
2. Carregar as bases de dados.
3. Armazenar os dados carregados em objetos que serão utilizados posteriormente.



### Atenção

Lembre-se que você pode acessar a qualquer momento o curso “**Análise de dados para a vigilância em saúde – curso básico**” e o curso “**Visualização de dados de interesse para a Vigilância em Saúde**”, obtendo os códigos desejados para a confecção do seu relatório. Caso não tenha feito os cursos, sugerimos que se inscreva neles. Maiores informações em <https://www.abrasco.org.br/site/analise-de-dados-para-a-vigilancia-em-saude/>

Observe como você deverá preencher o *chunk* para executar o primeiro passo da nossa análise (carregar os pacotes que serão utilizados):

```
# Carregando os pacotes necessários
require(foreign)
require(tidyverse)
require(lubridate)
require(kableExtra)
```

Vamos lá! Estamos construindo um relatório que irá apresentar a situação da dengue em Rosas para o secretário de Estado. Para isso, precisaremos utilizar dados de dengue para o Estado de Rosas. Aqui iremos utilizar como base de dados o arquivo exportado do SINAN Net chamado de **NINDINET.dbf**. Este arquivo está disponível no menu lateral “Arquivos” do Ambiente Virtual do curso.

Também iremos filtrar os registros de agravos relacionados à dengue e, em seguida, criaremos algumas colunas com informações-chave sobre esse agravio (**semana epidemiológica, mês, e ano da data de primeiros sintomas**) desses registros. Todas estas rotinas (passos 2 e 3) serão acrescidas no primeiro passo da nossa análise. Veja os comandos abaixo:

```
# Carregando os pacotes necessários
require(foreign)
require(tidyverse)
require(lubridate)
require(kableExtra)

# Importando o banco de dados com a função `read.dbf()`
nindi <- read.dbf(file = 'Dados/NINDINET.dbf')

# Armazenando apenas dados para dengue
dengue <- nindi |>
  # Filtrando apenas casos de dengue com a função filter()
  filter(ID_AGRAVO == 'A90') |>
  # Criando novas colunas com a função mutate()
  # Convertendo a coluna DT_SIN_PRI (data de primeiros sintomas) para o formato de date (Date)
  mutate(DT_SIN_PRI = ymd(DT_SIN_PRI),
        # Criando nova coluna com a semana epidemiológica com o uso da função epiweek()
        sem_epi = epiweek(DT_SIN_PRI),
        # Criando nova coluna com o ano epidemiológico com o uso da função epiyear()
        ano_epi = epiyear(DT_SIN_PRI),
        # Criando nova coluna com o mês de primeiros sintomas
        mes = month(DT_SIN_PRI))
```

Agora replique estes códigos no *chunk* do seu **RStudio**. Recomendamos que esta parte do código seja inserida no primeiro trecho de código do seu arquivo **.Rmd**, pois desta forma, todos os códigos executados posteriormente já contarão com os pacotes e objetos necessários para as análises que faremos. Lembre-se que você deverá sempre verificar se todos os pacotes necessários foram instalados e carregados no **R**, caso encontre algum aviso (*warning*) ou erro (*error*), revise seu código e instale os pacotes necessários.



Muitas vezes os códigos que geram as tabelas e os objetos que serão utilizados demoram um longo tempo para serem processados. Porém, com experiência você aprenderá a fazer pequenos ajustes que tornarão o processo mais rápido. A repetição de códigos, por exemplo, é um dos grandes motivos que tornam o processamento de relatórios **.Rmd** bastante demorados.

Uma alternativa que permite otimizar o tempo é fazer a preparação das tabelas e dos objetos que serão utilizados nos relatórios automáticos em outro arquivo de *script* em **.R**, de forma separada, e salvá-los como objetos do tipo tabelas (**.csv**) ou em objetos do tipo dados do **R** (**.Rdata**). Depois basta mencionar esses objetos em seu arquivo **.Rmd** utilizando a função **source()** no primeiro trecho de código do seu relatório em *RMarkdown*. Assim, o **R** irá buscar a versão “pronta” de sua tabela ou objeto, reduzindo o tempo de produção do seu relatório.

Você terá como resultado um *script* mais sucinto e fácil de compreender por qualquer pessoa que o manusear.

Precisaremos agora configurar o *chunk*. Copie o trecho de código que treinamos anteriormente e o insira trechos na linha 12 do seu arquivo **.Rmd**. A linha 12 é a que contém o seguinte código:

```
knitr::opts_chunk$set(echo = TRUE)
```

Pronto, agora verifique se o seu arquivo **.Rmd** ficou semelhante a Figura 53. Veja:

### **Figura 53: Construção do *chunk* para o *script* de análise da dengue do Estado de Rosas.**

```

6 ^ ---
7
8 ````{r setup, include=FALSE}
9 knitr::opts_chunk$set(echo = TRUE)
10
11 # Carregando os pacotes necessários
12 require(foreign)
13 require(tidyverse)
14 require(lubridate)
15 require(kableExtra)
16
17 # Importando o banco de dados com a função `read.dbf()`
18 nindi <- read.dbf(file = 'Dados/NINDINET.dbf')
19
20 # Armazenando apenas dados para dengue
21 dengue <- nindi |>
22   # Filtrando apenas casos de dengue com a função filter()
23   filter(ID_AGRAVO == 'A90') |>
24   # Criando novas colunas com a função mutate()
25     # Convertendo a coluna DT_SIN_PRI (data de primeiros sintomas) para o formato de date (Date)
26   mutate(DT_SIN_PRI = ymd(DT_SIN_PRI),
27     # Criando nova coluna com a semana epidemiológica com o uso da função epiweek()
28     sem_epi = epiweek(DT_SIN_PRI),
29     # Criando nova coluna com o ano epidemiológico com o uso da função epiyear()
30     ano_epi = epiyear(DT_SIN_PRI),
31     # Criando nova coluna com o mês de primeiros sintomas
32     mes = month(DT_SIN_PRI))
33 ````
```

Note que na Figura 53 este trecho inicial de código possui o argumento `include=FALSE`, ou seja, configuramos o *chunk* para que o código seja executado apenas quando renderizarmos o documento. Além disso, não deverá exibir nenhum *output*.

#### **Atenção**



Caso tenha encontrado dificuldade de chegar a um arquivo com os *scripts* que utilizamos, não se preocupe e continue no curso!

Deixamos pronto para você um arquivo de estudo com todos os elementos que aplicamos nesta subseção: o `exemplo6.Rmd`. Você poderá encontrá-lo acessando o menu lateral “Arquivos” do Ambiente Virtual do curso e fazer o *download*.

## 6 Incluindo gráficos ao relatório

Agora que temos nosso ambiente configurado e já sabemos editar os *chunks* (trechos de códigos), poderemos utilizar a tabela com casos notificados de dengue para produzir gráficos e gerar outras tabelas.

Para incluir gráficos que serão gerados de forma automática pelo R em seu documento, você deverá inserir o *script* abaixo, após o subtítulo **### 1. Distribuição de casos por semana epidemiológica**, com os comandos que produzirão o gráfico. Tudo isso deverá estar dentro do trecho de código (*chunk*) do seu documento, aquele delimitado por três acentos graves. Veja o código que iremos utilizar para produzir o gráfico e cole em seu arquivo **.Rmd**:

```
grafico_1 <- dengue |>

# Contando número de casos por ano e semana epidemiológica
count(ano_epi, sem_epi) |>

# Plotando visualização de gráfico por ano e semana epidemiológica
ggplot(aes(x = sem_epi, y = n, color = factor(ano_epi) )) +

# Adicionando linhas
geom_line() +

# Adicionando pontos
geom_point() +

# Aplicando novo tema para o gráfico
theme_minimal() +

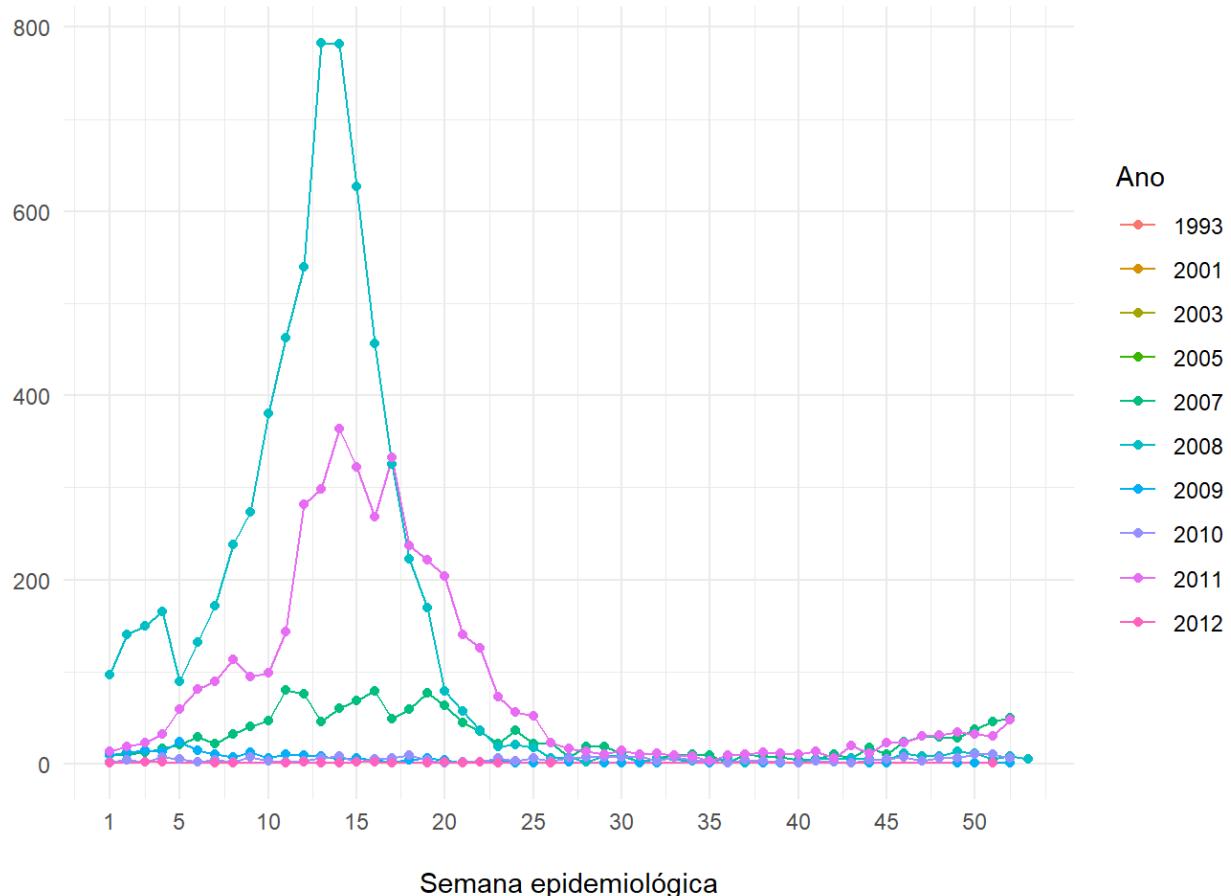
# Adicionando rótulo para o eixo x
xlab("\nSemana epidemiológica") +

# Adicionando rótulo para o eixo y
ylab("") +

# Definindo o título da legenda
scale_color_discrete("Ano") +

# Definindo o intervalo de valores dos rótulos do eixo x
scale_x_continuous(breaks = c(1, seq(5, 50, 5))) 

grafico_1
```



Pronto, agora verifique se seu arquivo `.Rmd` está semelhante ao apresentado na Figura 54. Nela você poderá observar que inserimos estes códigos após o subtítulo `## 1. Distribuição de casos por semana epidemiológica`. Veja a seguir:

**Figura 54: Incluindo gráfico *script* após o subtítulo do relatório.**

```

82 ## Análises
83
84 ### 1. Distribuição de casos por semana epidemiológica
85
86 ````{r}
87 grafico_1 <- dengue |>
88
89 # Contando número de casos por ano e semana epidemiológica
90 count(ano_epi, sem_epi) |>
91
92 # Plotando visualização de gráfico por ano e semana epidemiológica
93 ggplot(aes(x = sem_epi, y = n, color = factor(ano_epi) )) +
94
95 # Adicionando linhas
96 geom_line() +
97
98 # Adicionando pontos
99 geom_point() +
100
101 # Aplicando novo tema para o gráfico
102 theme_minimal() +
103
104 # Adicionando rótulo para o eixo x
105 xlab("\nSemana epidemiológica") +
106
107 # Adicionando rótulo para o eixo y
108 ylab("") +
109
110 # Definindo o título da legenda
111 scale_color_discrete("Ano") +
112
113 # Definindo o intervalo de valores dos rótulos do eixo x
114 scale_x_continuous(breaks = c(1, seq(5, 50, 5)))
115
116 grafico_1
117 ````{r}
118

```

Agora renderize o seu documento: clique no botão *knit* e gere um documento final do tipo `.html` que será semelhante ao apresentado na Figura 55.

## Figura 55: Renderização do script com inclusão de gráfico.

### Análises

#### 1. Distribuição de casos por semana epidemiológica

```
grafico_1 <- dengue |>

# Contando número de casos por ano e semana epidemiológica
count(ano_epi, sem_epi) |>

# Plotando visualização de gráfico por ano e semana epidemiológica
ggplot(aes(x = sem_epi, y = n, color = factor(ano_epi) )) +

# Adicionando linhas
geom_line() +

# Adicionando pontos
geom_point() +

# Aplicando novo tema para o gráfico
theme_minimal() +

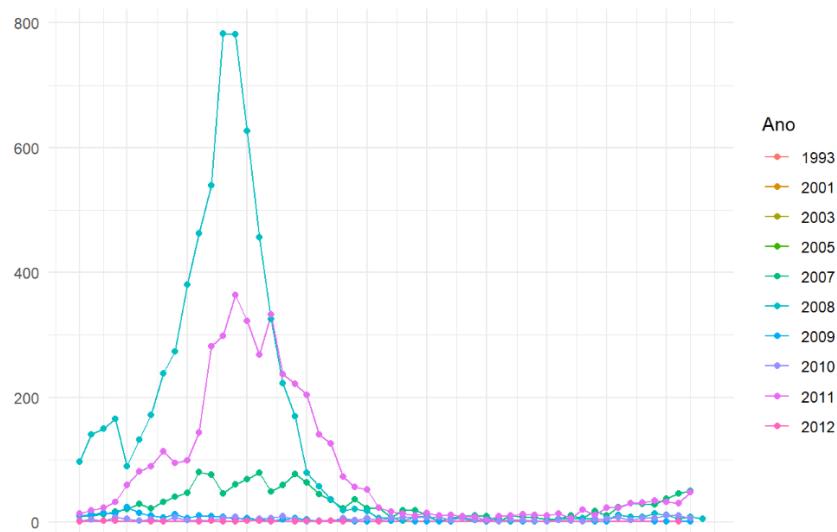
# Adicionando rótulo para o eixo x
xlab("\nSemana epidemiológica") +

# Adicionando rótulo para o eixo y
ylab("") +

# Definindo o título da legenda
scale_color_discrete("Ano") +

# Definindo o intervalo de valores dos rótulos do eixo x
scale_x_continuous(breaks = c(1, seq(5, 50, 5)))
```

grafico\_1



Observe na Figura 55 que o documento apresentará tanto o código quanto o *output* deste código, ou seja, o gráfico. Conforme vimos anteriormente, devemos incluir o argumento `echo = FALSE` (Figura 56) no início do trecho de código para que ele apresente somente o gráfico, dando uma estrutura de relatório a ele. Observe a Figura 56 abaixo:



**Figura 56: Renderização do script com a inclusão do argumento echo = FALSE na configuração do chunk.**

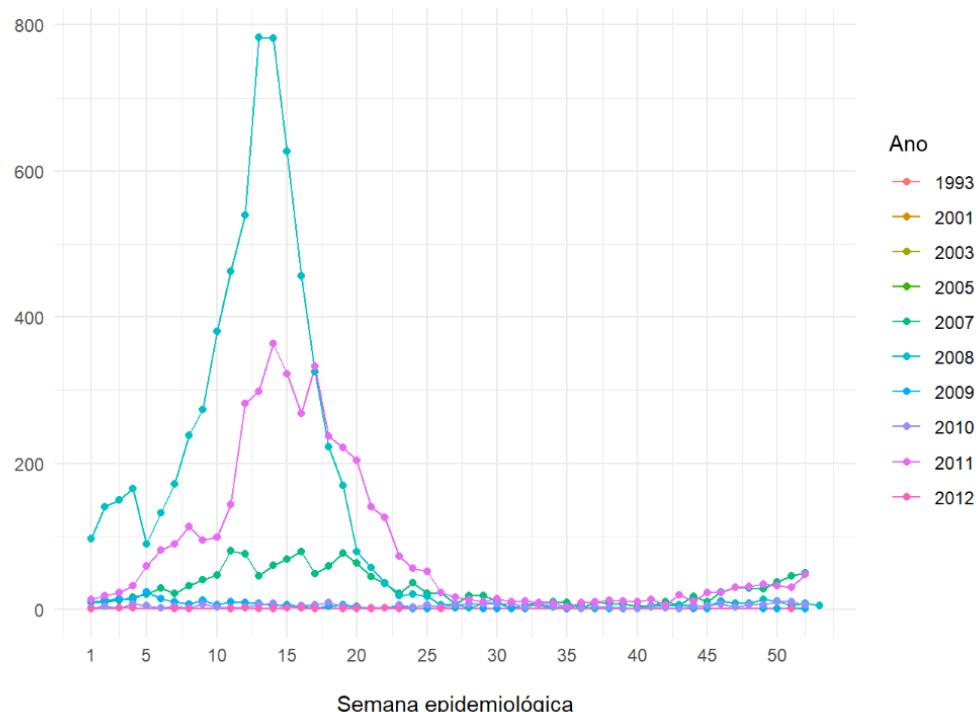
```
82 ## Análises
83
84 ### 1. Distribuição de casos por semana epidemiológica
85
86 ````{r, echo = FALSE}
87
88 grafico_1 <- dengue |>
89
90 # Contando número de casos por ano e semana epidemiológica
91 count(ano_epi, sem_epi) |>
92
```

Repita o processo e gere um novo arquivo do tipo `.html` clicando no botão `knit` no menu superior do R. Perceba que você obterá como resultado um documento semelhante ao apresentado na Figura 57, ou seja, você verá que o código foi omitido e que agora só conseguimos visualizar o gráfico que criamos.

**Figura 57: Renderização do script com a inclusão do argumento echo = FALSE para produção gráfica.**

## Análises

### 1. Distribuição de casos por semana epidemiológica





## Atenção

Caso tenha encontrado dificuldade de chegar a um arquivo com os *scripts* que utilizamos, não se preocupe e continue no curso!

Deixamos pronto para você um arquivo de estudo com todos os elementos que aplicamos nesta subseção: o `exemplo7.Rmd`. Você poderá encontrá-lo acessando o menu lateral “Arquivos” do Ambiente Virtual do curso e fazer o *download*.



É possível ainda customizar o gráfico de saída com os seguintes argumentos:

- `fig.cap`: título do gráfico
- `fig.height`: altura do gráfico, em polegadas
- `fig.width`: largura do gráfico, em polegadas
- `fig.align`: alinhamento do gráfico. Você pode escolher as seguintes alternativas:
  - `right`: alinhado à direita.
  - `left`: alinhado à esquerda
  - `center`: alinhado ao centro.

Para saber mais, acesse o curso **“Visualização de dados de interesse para a Vigilância em Saúde”**. Caso ainda não tenha feito o curso, sugerimos que se inscreva em  
<https://www.abrasco.org.br/site/analise-de-dados-para-a-vigilancia-em-saude/>.

## **7 Incluindo tabelas ao relatório**

Também será possível incluir e visualizar tabelas geradas pelo código em R no seu arquivo **.Rmd**, para que elas possam ser exibidas em seu documento final que poderá ser um PDF (**.pdf**), uma página da *internet* (**.html**) ou um documento para edição no Microsoft Word (**.pdf**).

Vamos lá! Estamos construindo o relatório epidemiológico sobre a dengue no Estado de Rosas, e agora geraremos uma tabela demonstrando a contagem de casos por classificação final para cada um dos anos epidemiológicos presentes em nossa base de dados **{NINDINET.dbf}**.

Observe a seguir como você poderá escrever seu *script*:

```
# Criando nova tabela com informações sobre dengue
tabela_dengue <- dengue |>

# Criando nova coluna com os nomes da Classificação Final utilizando a função mutate()
mutate(
  Classificacao = case_when(
    CLASSI_FIN == 1 ~ "Cura",
    CLASSI_FIN == 2 ~ "Óbito",
    CLASSI_FIN >= 3 ~ "Outro",
    is.na(CLASSI_FIN) ~ "Ignorado"
  )
) |>

# Contando o número de casos por ano epidemiológico e classificação final com a função count()
count(ano_epi, Classificacao) |>

# Pivotando a tabela para o formato "largo"
pivot_wider(names_from = Classificacao,
            values_from = n,
            values_fill = 0) |>

# Selecionando apenas as colunas de interesse
select(ano_epi, Cura, Óbito, Outro, `Ignorado`) |>

# Renomeando a coluna de ano epidemiológico ("ano_epi") para "Ano"
rename(Ano = ano_epi)
```

Agora copie o código apresentado acima e inclua-o em seu arquivo `.Rmd` no seu computador. Para que o relatório fique intuitivo, você deverá incluir este *chunk* após o título `## 2. Número de casos por classificação final`. Ele ficará semelhante ao apresentado na Figura 58.

**Figura 58: Script com inclusão da tabela de casos do Estado de Rosas (fictício).**

```

120 ## 2. Número de casos por classificação final
121
122 ```{r, echo = FALSE}
123
124 # Criando nova tabela com informações sobre dengue
125 tabela_dengue <- dengue |>
126
127 # Criando nova coluna com os nomes da Classificação Final utilizando a função mutate()
128 mutate(
129   Classificacao = case_when(
130     CLASSI_FIN == 1 ~ "Cura",
131     CLASSI_FIN == 2 ~ "Óbito",
132     CLASSI_FIN >= 3 ~ "Outro",
133     is.na(CLASSI_FIN) ~ "Ignorado"
134   )
135 ) |>
136
137 # Contando o número de casos por ano epidemiológico e classificação final com a função count()
138 count(ano_epi, Classificacao) |>
139
140 # Pivotando a tabela para o formato "largo"
141 pivot_wider(names_from = Classificacao,
142   values_from = n,
143   values_fill = 0) |>
144
145 # Selecionando apenas as colunas de interesse
146 select(ano_epi, Cura, Óbito, Outro, `Ignorado`) |>
147
148 # Renomeando a coluna de ano epidemiológico ("ano_epi") para "Ano"
149 rename(Ano = ano_epi)
150 ```
151

```

Note, na Figura 58, a inclusão do argumento `echo = FALSE` no início do *chunk*.

Pronto, já temos nossa tabela. Agora precisaremos inserir um comando para que ela fique visível em nosso relatório. Para isso, iremos utilizar algumas funções do pacote `kableExtra` e `kable()` que permitirão que criemos a tabela e o pacote `kable_styling()` para adicionar alguns detalhes à visualização da tabela criada. Observe o trecho de código a seguir e inclua-o logo abaixo dos demais códigos do *chunk* com os quais criamos a tabela.

```

kable(tabela_dengue) |>
  kable_styling()

```

Agora, vamos renderizar o arquivo utilizando o botão `knit` do menu superior do R para obter um resultado semelhante ao observado na Figura 59.

**Figura 59: Renderização do script com tabela  
com o número de casos notificados de dengue.**

## 2. Número de casos por classificação final

Ano	Cura	Óbito	Outro	Ignorado
1993	0	0	1	0
2001	0	0	1	0
2003	1	0	0	0
2005	1	0	0	0
2007	347	27	1074	50
2008	1164	453	4751	219
2009	29	7	166	8
2010	86	30	105	13
2011	3584	52	541	45
2012	18	1	2	5



Caso o documento final produzido seja um arquivo .pdf é recomendado inserir o argumento `latex_options = "HOLD_position"` na função `kable_styling`, conforme exemplo abaixo:

```
kable(tabela_dengue) |>
  kable_styling(latex_options = "HOLD_position")
```

Isto irá garantir que a tabela gerada seja posicionada de forma correta no arquivo final.



### Atenção

Caso tenha encontrado dificuldade de chegar a um arquivo com os *scripts* que utilizamos, não se preocupe e continue no curso!

Deixamos pronto para você um arquivo de estudo com todos os elementos que aplicamos nesta subseção: o `exemplo8.Rmd`. Você poderá encontrá-lo acessando o menu lateral “Arquivos” do Ambiente Virtual do curso e fazer o *download*.

Atualmente, existem diversos pacotes que permitem personalizar o formato de tabelas geradas pelo R para documentos. Um guia detalhado de cada um desses pacotes tornaria este curso muito extenso. No entanto, é muito interessante conhecer essas possibilidades adicionais. Assim, a seguir indicamos alguns dos principais pacotes utilizados e a descrição dos seus principais usos com um guia de como aprender a utilizá-los:

Pacote	Descrição	Tutorial
Kableextra	Personalizar tabelas para <code>html</code>	<a href="#">link</a>
gtsummary	Personalizar tabelas de resultados de análises	<a href="#">link</a>
flextable	Personaliza tabelas <code>data.frame</code> para diferentes formatos	<a href="#">link</a>
huxtable	Personaliza tabelas para formato <code>html</code> e <code>LaTeX</code>	<a href="#">link</a>

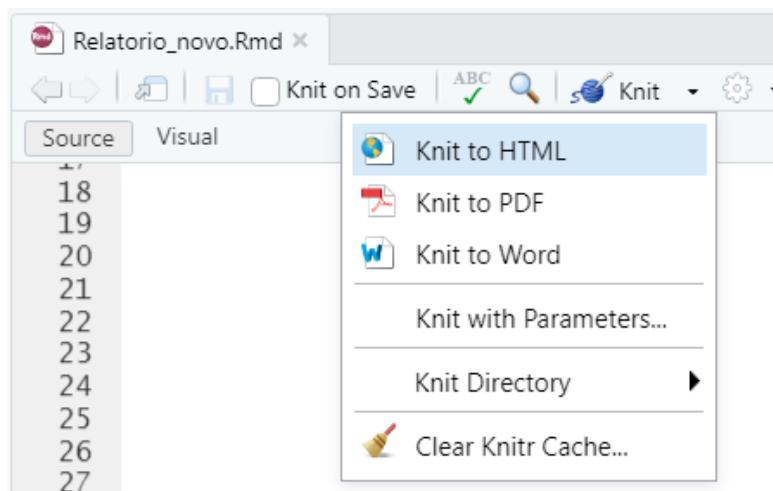
A partir de agora você pode incluir os gráficos, tabelas e textos que desejar para construir o seu modelo de relatório. Vá recheando-o com informações conforme suas necessidades utilizando os códigos de análise do R no Rmarkdown.

## 8 Exportando o relatório

Pronto, até aqui conseguimos criar um relatório com informações básicas sobre a situação epidemiológica da dengue no Estado de Rosas. Agora precisamos encaminhá-lo ao secretário de saúde do estado para que ele participe da coletiva à imprensa. É provável que ele também queira imprimir o mesmo relatório para disponibilizar à imprensa logo após a coletiva. Ou seja, chegou a hora de exportar um documento final que possa ser compartilhado ou impresso.

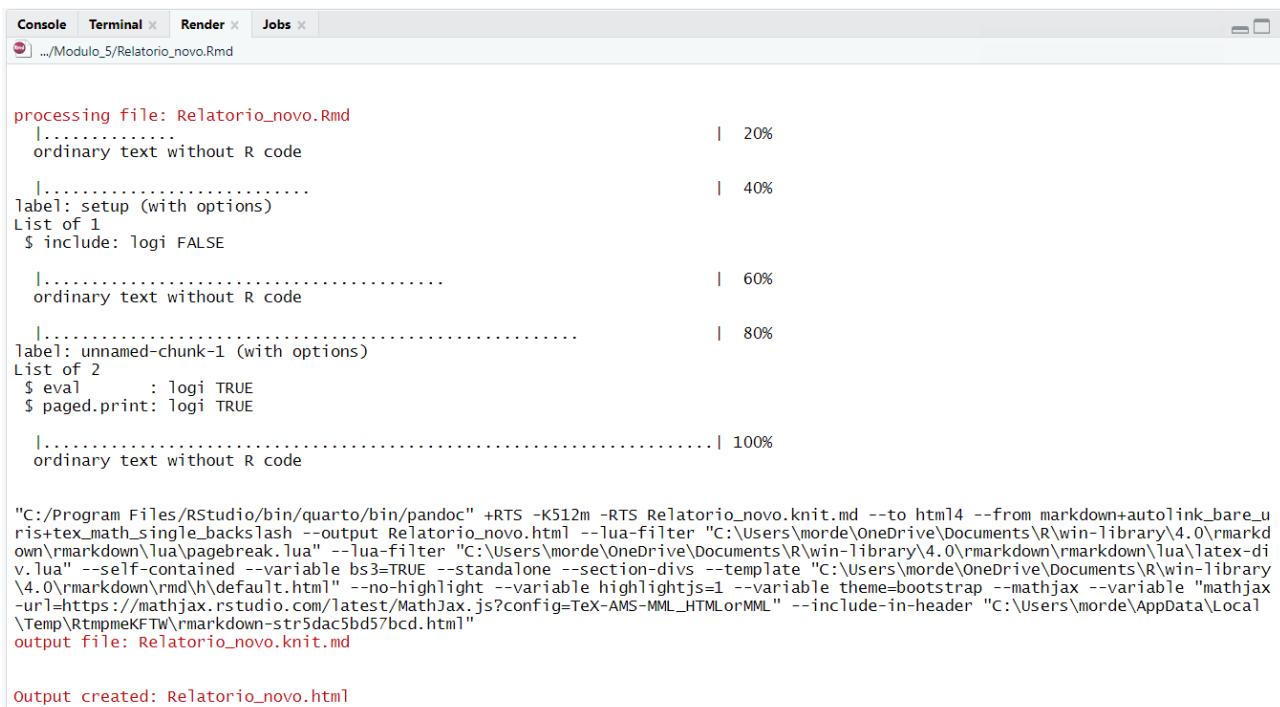
O `Rmarkdown` oferece três opções de arquivos para exportação: `.html`, `.pdf`, e o `.docx`. Para iniciar a exportação, basta renderizar o arquivo no formato que desejar selecionando o botão `knit` na parte superior da aba do seu `script`. Ao clicar neste ícone, você verá uma janela semelhante a visualizada na Figura 60 a seguir. Veja:

**Figura 60: Renderizando o script com o botão `knit`.**



Na sequência, basta selecionar o formato do arquivo que mais se adequará à sua necessidade e clique nele para exportar o seu relatório. O `RStudio` irá compilar e renderizar o seu documento no formato escolhido. Você poderá acompanhar as mensagens sobre o andamento deste processo no painel console. Para isso observe a aba inferior à esquerda do `RStudio` denominada **Render** (Figura 61).

**Figura 61: Tela de visualização da aba *render* do painel console.**



The screenshot shows the RStudio interface with the 'Render' tab selected. A progress bar indicates the status of the compilation process:

- processing file: Relatorio\_novo.Rmd
- ordinary text without R code (20%)
- label: setup (with options) (40%)
- List of 1
- \$ include: logi FALSE
- ordinary text without R code (60%)
- label: unnamed-chunk-1 (with options) (80%)
- List of 2
- \$ eval : logi TRUE
- \$ paged.print: logi TRUE
- ordinary text without R code (100%)

Below the progress bar, the command run is shown:

```
"C:/Program Files/RStudio/bin/quarto/bin/pandoc" +RTS -K512m -RTS Relatorio_novo.knit.md --to html4 --from markdown+autolink_bare_underscores+tex_math_single_backslash --output Relatorio_novo.html --lua-filter "C:\Users\morde\OneDrive\Documents\R\win-library\4.0\rmarkdown\lua\pagebreak.lua" --lua-filter "C:\Users\morde\OneDrive\Documents\R\win-library\4.0\rmarkdown\lua\latext-dv.lua" --self-contained --variable bs3=TRUE --standalone --section-divs --template "C:\Users\morde\OneDrive\Documents\R\win-library\4.0\rmarkdown\rmd\h\default.html" --no-highlight --variable highlightjs=1 --variable theme=bootstrap --mathjax --variable "mathjax-url=https://mathjax.rstudio.com/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML" --include-in-header "C:\Users\morde\AppData\Local\Temp\RtmpmeKFTw\rmarkdown-str5dac5bd57bcd.html"
```

Output created: Relatorio\_novo.html

Caso exista algum erro na hora de gerar o seu documento, uma mensagem de aviso *warning* será informado. Nele você terá detalhes sobre o erro e poderá editar o *script*. Esses avisos também são apresentados no painel console na aba *render*. Após esse passo ser concluído com sucesso, o **RStudio** irá abrir uma nova janela mostrando o seu documento final. Você também pode abrir o arquivo gerado no seu programa de preferência no mesmo diretório em que o arquivo **.Rmd** está localizado.

Pronto, agora você já possui um arquivo com as informações básicas para entregar ao secretário de saúde de Rosas. Lembre-se que se ele te ligar novamente nesta semana, com este *script* pronto bastará apenas atualizar a fonte de dados e clicar em **RUN** para gerar um novo relatório de forma automatizada.

## 9 Outros componentes e ajustes finais

Ao produzirmos um relatório buscamos sempre que ele seja apresentado com o melhor *layout* possível. Sempre será necessário fazer ajustes nos textos, figuras ou tabelas. Muitas vezes não contamos com uma equipe de comunicação que poderá nos auxiliar nesta etapa e o profissional de vigilância deverá desenvolver um documento claro, intuitivo e detalhado de forma autônoma.

Nesta seção, veja abaixo algumas dicas para tornar seu relatório automatizado ainda melhor!

### 9.1 Automatizando a atualização do texto

Em relatórios automatizados, algumas vezes pode ser interessante inserir valores em meio aos textos para que estes sejam atualizados automaticamente (a todo momento que modificarmos a fonte de dados). Você se lembra do conteúdo que criamos para construir a introdução do nosso relatório? Nele o texto inicial trazia trechos sem o seu preenchimento, referindo que havia **N** casos de dengue e **M** óbitos (Figura 11):

“O Departamento de Vigilância Epidemiológica do Estado de Rosas, por meio deste boletim epidemiológico (referência 1), apresenta informações gerais sobre a dengue (referência 2), assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou **N** casos confirmados de dengue e **M** óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na Figura 1. O número de casos por classificação final são apresentado na Tabela 1.”

Agora, precisamos programar no R um script para que estes valores sejam calculados e inseridos no texto de forma automática, sendo modificados todas as vezes em que a base de dados é alterada.

Vamos lá! Para isso, adicione o código abaixo no primeiro trecho do *chunk* do arquivo .Rmd em seu computador:

```
n_casos = nrow(dengue)
n_obitos = sum(dengue$CLASSTI_FIN == 2, na.rm = TRUE)
```

Observe na Figura 62 como deverá ficar o início de seu arquivo .Rmd (linhas 36 e 37 do script).

**Figura 62: Tela com a inserção de valores no texto do script.**

```
9
10 ````{r setup, include=FALSE}
11 knitr::opts_chunk$set(echo = TRUE)
12
13 # Carregando os pacotes necessários
14 require(foreign)
15 require(tidyverse)
16 require(lubridate)
17 require(kableExtra)
18
19 # Importando o banco de dados com a função `read.dbf()`
20 nindi <- read.dbf(file = '../Dados/NINDINET.dbf')
21
22 # Armazenando apenas dados para dengue
23 dengue <- nindi |>
24   # Filtrando apenas casos de dengue com a função filter()
25   filter(ID_AGRAVO == 'A90') |>
26   # Criando novas colunas com a função mutate()
27   # Convertendo a coluna DT_SIN_PRI (data de primeiros sintomas) para o formato de date (Date)
28   mutate(DT_SIN_PRI = ymd(DT_SIN_PRI),
29         # Criando nova coluna com a semana epidemiológica com o uso da função epiweek()
30         sem_epi = epiweek(DT_SIN_PRI),
31         # Criando nova coluna com o ano epidemiológico com o uso da função epiyear()
32         ano_epi = epiyear(DT_SIN_PRI),
33         # Criando nova coluna com o mês de primeiros sintomas
34         mes = month(DT_SIN_PRI))
35
36 n_casos = nrow(dengue)
37 n_obitos = sum(dengue$CLASSTI_FIN == 2, na.rm = TRUE)
38
39
```

Bom, agora vamos detalhar o que aconteceu! Você poderá inserir a qualquer momento os valores definidos, por estes objetos, em textos que são fixos utilizando a seguinte escrita: `aspas + r + nome_do_objeto + aspas`. Observe abaixo como ficou o texto final com estas inclusões:

O \*\*Departamento de Vigilância Epidemiológica do Estado de Rosas\*\*, por meio deste boletim informativo<sup>1</sup>, apresenta informações gerais sobre a dengue<sup>2</sup>, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou `r n\_casos` casos confirmados de dengue e `r n\_obitos` óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na \*Figura 1\*. O número de casos por classificação final é apresentado na \*Figura 1\*.

Digite os códigos apresentado acima no seu RStudio para que você possa obter um documento final como o apresentado na Figura 63.

### **Figura 63: Renderização do script com valores substituídos no texto.**

#### Sobre

O Departamento de Vigilância Epidemiológica do Estado de Rosas, por meio deste boletim informativo<sup>1</sup>, apresenta informações gerais sobre a dengue<sup>2</sup>, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou 12781 casos confirmados de dengue e 570 óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na *Figura 1*. O número de casos por classificação final são apresentados na *Tabela 1*.

Foi fácil, não é mesmo? Daremos mais dicas de aprimoramento mais adiante nesta seção.

## ***9.2 Quebras de linha***

Ao exportar arquivos para formatos PDF (.pdf) ou Word (.docx) você poderá determinar o local de quebra de linha entre diferentes pontos do texto. Esta etapa facilitará para que o leitor não se sinta perdido no texto ou ache que você esqueceu conteúdos que ficarão na página seguinte. Para isso você pode incluir a expressão `\newpage` ao longo dos trechos de texto em seu arquivo Rmarkdown.

Para praticar, inclua esta expressão antes do início de cada título de seu documento .Rmd. Em seguida, selecione a seta ao lado do botão *knit*, e escolha a opção de renderizar seu documento no formato .pdf. Observe um exemplo no código abaixo:

```
# Título 1  
  
'\newpage'  
  
# Título 2
```

Agora avalie o resultado em seu RStudio. O que achou desta configuração?

## 9.3 Espaçamento

Por padrão, o R ignora todos os espaços entre parágrafos e palavras no arquivo `.Rmd`. Mas se você julgar necessário pode inclui-los explicitamente. Para isso basta escrever a sintaxe: `</br>` para que todos os arquivos que você renderizar (`.html`, `.pdf` e `.docx`). Para praticar, insira o texto  entre o primeiro parágrafo do documento e a primeira imagem, conforme o código abaixo:

O \*\*Departamento de Vigilância Epidemiológica do Estado de Rosas\*\*, por meio deste boletim informativo<sup>1</sup>, apresenta informações gerais sobre a dengue<sup>2</sup>, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou `r n\_casos` casos confirmados de dengue e `r n\_obitos` óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na \*Figura 1\*. O número de casos por classificação final é apresentado na \*Figura 1\*.

```
</br>  
&nbsp  
</br>  
&nbsp  
{width="100%"}  
[73]
```

Veja como ficaria o seu código no Rstudio (Figura 64):

### Figura 64: Espaçamento do script.

```
40
41 ## Sobre
42
43 O Departamento de Vigilância Epidemiológica do Estado de Rosas, por meio deste boletim informativo1, apresenta informações gerais sobre a dengue2, assim como uma breve análise dos dados históricos relativos à situação epidemiológica da dengue no Estado de Rosas. Entre 2007 e 2012, o município registrou "r_n_casos" casos confirmados de dengue e "r_n_obitos" óbitos. A distribuição dos casos confirmados por semana epidemiológica é apresentada na *Figura 1*. O número de casos por classificação final são apresentados na *Tabela 1*.
44 &nbsp;
45 &nbsp;
46 &nbsp;
47 &nbsp;
48
49 ! [] (Imagens/dengue.png) {width="100%"}
```

Agora renderize o seu documento novamente no formato .pdf e verifique a diferença!

### Atenção



Caso tenha encontrado dificuldade de chegar a um arquivo com os *scripts* que utilizamos, não se preocupe e continue no curso!

Deixamos pronto para você um arquivo de estudo com todos os elementos que aplicamos nesta subseção: o `exemplo_1_final.Rmd`. Você poderá encontrá-lo acessando o menu lateral “Arquivos” do Ambiente Virtual do curso e fazer o *download*.

## 9.4 Cabeçalho YAML

Conforme vimos no início deste curso, todo arquivo `.Rmd` tem em seu início um cabeçalho com **metadados** sobre o documento. Lembre-se que os limites são indicados por três hífens (---) no início e final desta parte do documento.

Este cabeçalho é escrito em um formato que inclui informações gerais sobre o documento, assim como definições indicando como o documento final (*output*) será renderizado. As informações incluídas no arquivo irão aparecer no início dele. Veja abaixo algumas informações mais comuns que poderão ser configuradas:

- título: `title`
- subtítulo: `subtitle`
- autor: `author`
- data: `1º de janeiro de 2022`

Veja um exemplo de um cabeçalho YAML no arquivo `.Rmd`:

**Figura 65: Cabeçalho YAML do script.**

```
1 ---  
2 title: "Relatório Informativo sobre Dengue"  
3 author: "Vigilância Epidemiológica do Estado de Rosas"  
4 date: '2022-07-11'  
5 output: html_document  
6 ---  
7
```

Agora veja como fica a renderização do arquivo na Figura 66 após as informações que inserimos no cabeçalho `YAML` do *script*.



**Figura 66:** Renderização do *script* com as informações do cabeçalho YAML.

## Relatório Informativo sobre Dengue

Vigilância Epidemiológica do Estado de Rosas

2022-07-11

Como vimos durante este curso, os códigos em R ou *chunks* podem ser incluídos pelo uso da seguinte expressão `{```r`}`. Podemos configurar apenas uma única vez no cabeçalho YAML, por exemplo, para que todos os blocos de conteúdo *chunks* utilizem a data em que o documento foi gerado. Desta forma, o valor a ser colocado após o argumento `date:` deve ser: "2022-10-27". Note que além dos acentos graves, ainda é necessário usar aspas ao redor do valor. Assim, você deverá escrever o código final para incluir esta data como o apresentado na Figura 67.

**Figura 67:** Cabeçalho YAML do *script*.

```
1 ---  
2 title: "Relatório Informativo sobre Dengue"  
3 author: "Vigilância Epidemiológica do Estado de Rosas"  
4 date: "`r Sys.Date()`"  
5 output: html_document  
6 ---  
7
```

Agora observe como ficará o início do documento que criamos para o Estado de Rosas ao renderizá-lo (Figura 68).

**Figura 68:** Renderização do arquivo com o cabeçalho YAML do *script* editado.

## Relatório Informativo sobre Dengue

Vigilância Epidemiológica do Estado de Rosas

2022-10-04

Também é possível personalizar o formato em que as datas são exibidas pelo R, utilizando a função `format()`. Veja alguns exemplos mais utilizados:

```
format(Sys.time(), '%d %B, %Y'): "19 julho, 2022" format(Sys.time(), '%B de %Y'):  
"julho de 2022" format(Sys.time(), '%d/%m/%Y'): "19/07/2022"
```

Outro argumento que pode ser configurado é o tipo de documento que será gerado, ao se renderizar o documento final. Este argumento é especificado como `output`; no cabeçalho YAML. Observe como podemos especificar três formatos mais utilizados de arquivos:

- Arquivo no formato PDF (`.pdf`): `output: "pdf_document"`
- Página da *internet* (`.html`): `output: "pdf_document"`
- Documento do Microsoft Word (`.docx`): `output: "word_document"`



Para documentos criados no formato Microsoft Word é possível especificar uma referência de estilo utilizando um **documento modelo**. Estas referências de estilo incluem características como margens do documento, estilo de títulos, fontes, entre diversas outras. Estas configurações são incluídas pelo uso de um documento de word (`.docx`) na mesma pasta em que está o seu arquivo `Rmd`. Achou complexo? Vamos visualizar a Figura 69 para te apoiar.

Observe na Figura 69 o código em seu cabeçalho YAML (linhas 5 a 7 do exemplo):

**Figura 69: Script com cabeçalho YAML (linhas 5 a 7 do exemplo).**

```
1 ---  
2 title: "Tutorial para R Markdown"  
3 subtitle: "Módulo 5"  
4 author: "Curso ABRASCO"  
5 output:  
6   word_document:  
7     reference_docx: documento_de_referencia.docx  
8 ---
```

Note que o nome `documento_de_referência.docx` pode ser substituído pelo nome do arquivo `.docx` que será utilizado como modelo de estilo de formatação.

Para um aprofundamento deste tema, você também pode consultar este [tutorial em inglês](#).



## Nossos cursos

Pronto, chegamos ao final deste curso! Agora você já conhece as principais ações para construir um *relatório* com o apoio da linguagem de programação R. Quer seguir a diante no aprendizado? Você encontrará outras etapas para aprofundamento das análises de dados em vigilância em saúde nos outros cursos. Aproveite e já faça sua inscrição nos cursos abaixo clicando nos *links*:

- [Análises de dados para Vigilância em Saúde - curso básico.](#)
- [Visualização de dados de interesse para a vigilância em saúde.](#)
- [Construção de diagramas de controle na vigilância em saúde.](#)
- [Linkage de bases de dados de saúde.](#)
- [Análise espacial de dados para a vigilância em saúde.](#)
- [Construção de painéis \(dashboards\) para monitoramento de indicadores de saúde.](#)

Aproveite!

