

Relatórios em R Markdown

O que é o R Markdown?

O R Markdown é uma ferramenta para criação de relatórios automatizados utilizando as linguagem R e Markdown.

A linguagem de marcação Markdown serve para construirmos e formatarmos diversos formatos de arquivos (PDF, HTML, Word, entre outros) a partir de um arquivo de texto com regras bem simples. O R Markdown é uma extensão de Markdown que nos permite colocar código de R.

Linguagens de marcação utilizam marcadores (símbolos, tags, funções) para formatar um arquivo de texto simples. Os exemplos mais famosos de linguagem de marcação são o HTML e LaTeX.

Após a construção do documento, para gerarmos o relatório na extensão desejada, precisamos *renderizá-lo*, isto é, transformar o arquivo R Markdown em um PDF, HTML ou Word. Isso pode ser feito no RStudio a partir do botão `knit`, que fica logo acima do script, ou pelo atalho `CTRL + SHIFT + K`.

Regras simples de formatação

Usando o R Markdown, podemos criar arquivos HTML, PDF e Word sem precisar sair do R. A grande vantagem é poder de automatização. Construindo um relatório em R Markdown, com exceção das interpretações e conclusões, só precisamos montá-lo uma vez. A partir daí, com apenas um clique podemos:

- replicar o relatório para diversas versões da base de dados (modificações, correções, processos periódicos);
- replicar o relatório para diversas variáveis.

Marcadores

A seguir, apresentamos uma lista dos principais marcadores utilizados para formatar texto:

- uma palavra entre asteriscos fica em itálico: `*texto*` é transformado em *texto*
- uma palavra entre dois asteriscos fica em negrito: `**texto**` é transformado em **texto**
- um ou mais hashtags viram títulos: `#` Título muito grande, `##` Título grande, `###` Título médio, `####` Título pequeno, `#####` Título muito pequeno
- hiperlinks podem ser criados com a estrutura `[texto] (link)`:

`[link para o site da Curso-R] (https://curso-r.com)` é transformado em link para o site da Curso-R.

- para deixar o texto com esse formato (formato de código), apenas coloque o texto entre duas crases.

Chunks: escrevendo nosso código de R

Em um arquivo R Markdown, precisamos escrever nossos códigos dentro dos *chunks*. Para inserir um chunk, utilize o atalho CTRL + ALT + I.

Dentro dos chunks você poderá escrever códigos em R como se fosse o nosso script .R tradicional. Por padrão, o código dentro do chunk será colocado no relatório, assim como o resultado da execução desse código (i.e., tudo que seria). Veja o exemplo abaixo:

```
meu_vetor <- c(1, 2, 3)
meu_vetor + 1
```

```
## [1] 2 3 4
```

Não é apenas o resultado da última linha que é colocada no relatório. Todo resultado que seria imprimido na tela (Console) também vai para o relatório. Repare que objetos criados em um chunk ficam disponíveis para todos os chunks abaixo dele.

```
meu_vetor + 1
```

```
## [1] 2 3 4
```

```
meu_vetor - 1
```

```
## [1] 0 1 2
```

```
meu_vetor * 10
```

```
## [1] 10 20 30
```

Para alterar esses comportamentos padrões, utilizamos os parâmetros do chunk. Os parâmetros são colocados dentro das chaves, na linha que define o começo do chunk. Esse `r` que aparece em todos os chunks representa que o código dentro dele é um código de R.

Para impedir que o código de um chunk apareça no relatório, basta usar o parâmetro `echo = FALSE`. As chaves neste caso ficaria `{r, echo = FALSE}`. Quando fazemos isso, apenas o resultado é mostrado no relatório.

```
## [1] 2 3 4
```

Também podemos impedir que um chunk seja avaliado, mostrando apenas o código no relatório, usando o argumento `eval = FALSE`.

```
meu_vetor + 1
```

Por fim, podemos rodar o chunk sem colocar nem o código nem o resultado no relatório usando o argumento `include = FALSE`. Isso pode ser utilizado para carregar pacotes, definir funções ou fazer qualquer tipo de operação auxiliar que o leitor do relatório não precisa saber.

Para saber mais sobre os parâmetros dos chunks, consulte este guia (inglês).

Importante dados

Você pode carregar pacotes e dados normalmente dentro de um script R Markdown.

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.2      v purrr 0.3.4
## v tibble 3.0.4       v dplyr 1.0.2
## v tidyr 1.1.2        v stringr 1.4.0
## v readr 1.4.0        v forcats 0.5.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

imdb <- read_rds("../dados/imdb.rds")
```

Veja que as mensagens e warnings dos nossos códigos também são colocadas no relatório. Para evitar isso, basta usarmos os parâmetros `message=FALSE` e `warning=FALSE`.

Você precisa carregar o pacote apenas uma vez em cada documento. Uma vez carregado um pacote, suas funções estarão disponíveis para todo código em R abaixo, no mesmo ou em outros chunks.

Incluindo tabelas

A função `knitr::kable()` é muito útil para gerar tabelas bem formatadas.

A seguir, mostramos os 10 filmes com maior lucro na base.

```
imdb %>%
  mutate(lucro = receita - orcamento) %>%
  top_n(10, lucro) %>%
  arrange(lucro) %>%
  mutate(
    pos = 10:1,
    lucro = scales::dollar(lucro)
  ) %>%
  select(`Posição` = pos, Filme = titulo, Lucro = lucro) %>%
  knitr::kable()
```

Posição	Filme	Lucro
10	The Hunger Games	\$329,999,255
9	The Dark Knight	\$348,316,061
8	Star Wars: Episode I - The Phantom Menace	\$359,544,677
7	The Lion King	\$377,783,777
6	The Avengers	\$403,279,547
5	E.T. the Extra-Terrestrial	\$424,449,459
4	Star Wars: Episode IV - A New Hope	\$449,935,665
3	Titanic	\$458,672,302
2	Jurassic World	\$502,177,271
1	Avatar	\$523,505,847

Incluindo gráficos

Para construir gráficos, não há segredos. O mesmo gráfico que apareceria na aba **Plots** do RStudio aparecerá no relatório.

```
imdb %>%  
  group_by(ano) %>%  
  summarise(nota_media = mean(nota_imdb, na.rm = TRUE)) %>%  
  ggplot() +  
  geom_line(aes(x = ano, y = nota_media)) +  
  theme_minimal()
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

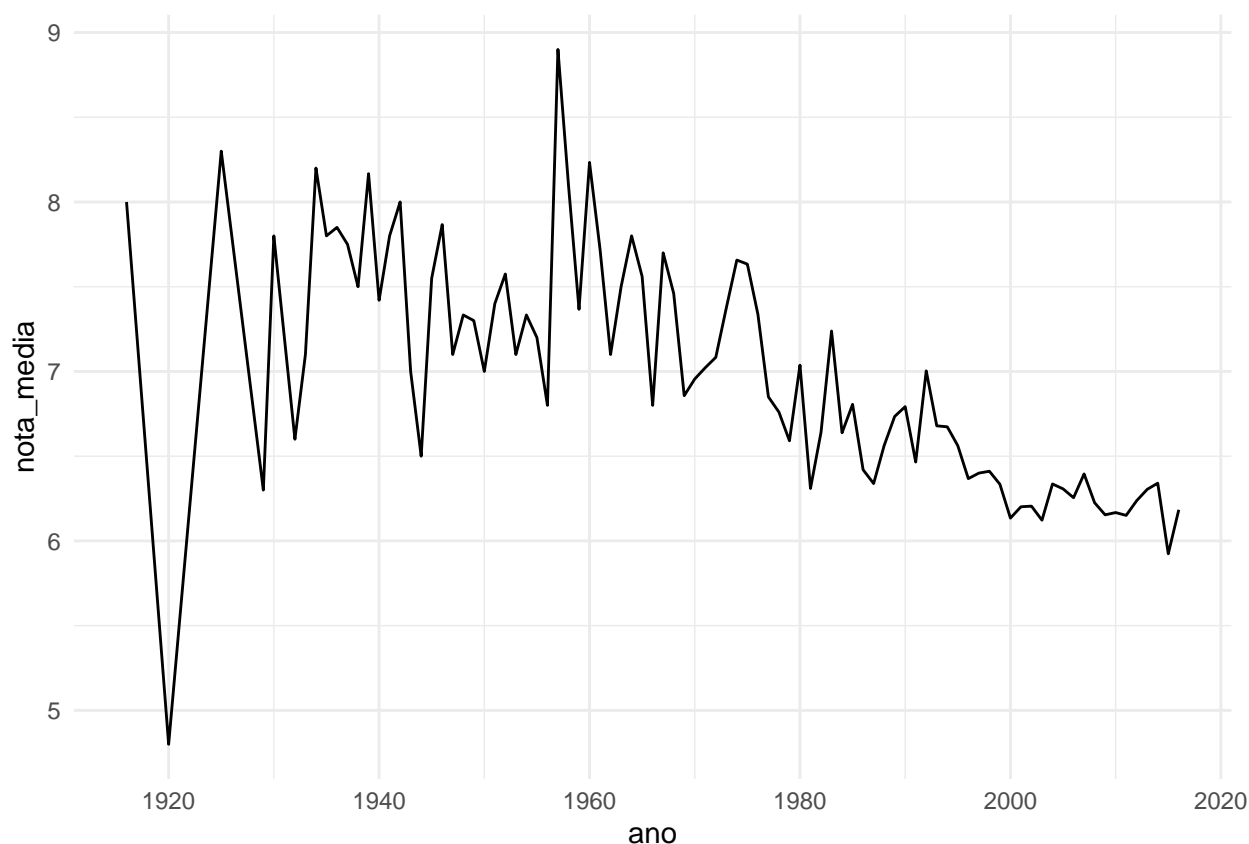


Figure 1: Figura 1. Nota média ao longo dos anos.

Para centralizar o gráfico no documento, você pode usar o parâmetro `fig.align = "center"` no chunk. Para alterar o tamanho da figura, existem os parâmetros `fig.width` (comprimento) e `fig.height` (altura). O parâmetro `fig.cap` coloca legendas.