

Apresentação RMarkdown

Wilson Tarantin Junior

23/05/2023

R Markdown

O R Markdown é um ambiente onde é possível escrever os scripts em R e, caso necessário, exportá-los para alguns formatos como PDF, Word ou HTML. Em geral, melhora a organização das informações e a torna mais fácil. O compartilhamento do código também é facilitado.

Em um projeto, poderíamos escrever diversas linhas explicativas ou nossos comentários sobre determinada etapa da análise.

Logo que aberto um novo documento, o R Markdown é um grande campo em branco onde podemos escrever os textos. Um novo arquivo vem com algumas informações padrões, mas podem ser apagadas sem problemas (mantendo o cabeçalho, como acima).

Então, primeiramente, o R Markdown funciona como um editor de textos. Note que não é necessário começar o texto com o #, como em um R Script comum.

Também é possível formatar os textos, mas as formatações aparecem no documento quando for publicado:

1. *Texto em itálico*: com um * no início e no fim.
2. **Texto em negrito**: com dois ** no início e no fim.
3. A seguir, com um * no início do texto, são gerados tópicos (bullets)
 - Tópico 1
 - Tópico 2
 - Subtópico 2.1 (aqui tem um tab no início)
4. Texto Subscrito: Texto_{abcd123}
5. Texto Sobrescrito: Texto^{efgh456}
6. Equações podem ser escritas na sintaxe LaTeX. A seguir, algumas informações básicas:

Para escrever uma equação na linha, coloque-a entre cifrão: $a = b^2$ No output, existiria a equação $a = b^2$ e depois o texto continuaria.

Para escrever uma equação que fica fora do texto, mas em outra linha, utilize dois cifrões:

$$a = \sqrt{b} + c^2$$

$$a = (b \times c) + \left(\frac{d}{e}\right)$$

$$\alpha_1 = \beta_1 + \beta_2 - \beta_3^2$$

No R Markdown, além dos textos e equações, podemos inserir os códigos do R para que sejam executados. Para tanto, devemos indicar um campo específico que contém o código. O campo é chamado de chunk. O campo fica indicado conforme o seguinte:

```
vetor_num <- c(1,2,3,4,5)
```

O chunk fica destacado em cinza e fica indicado que trata-se de um código em R. Para inserir este campo, basta ir até o botão +c acima e selecionar “R”.

Na engrenagem que fica dentro de cada chunk, é possível atribuir nome ao código daquele chunk e escolher como ele será reportado no documento final.

Para executar o código do chunk, basta clicar no “play” em verde.

Vamos abrir um dataset e ver como os outputs ficam no documento final:

```
library(readxl)
dados <- read_excel(file.path("../", "data", "(1.2) Dataset Aula Data Wrangling.xls"))
```

Primeiro, vamos carregar a biblioteca:

```
library(tidyverse)
library(knitr)
```

Neste caso das bibliotecas, na engrenagem, é possível desativar as mensagens sobre pacotes carregados para o documento final.

Vamos gerar a média das variáveis “tempo” e “distância”, depois de alterar os nomes do dataset original. Por fim, vamos utilizar a função kable() para criar uma tabela com as informações das médias:

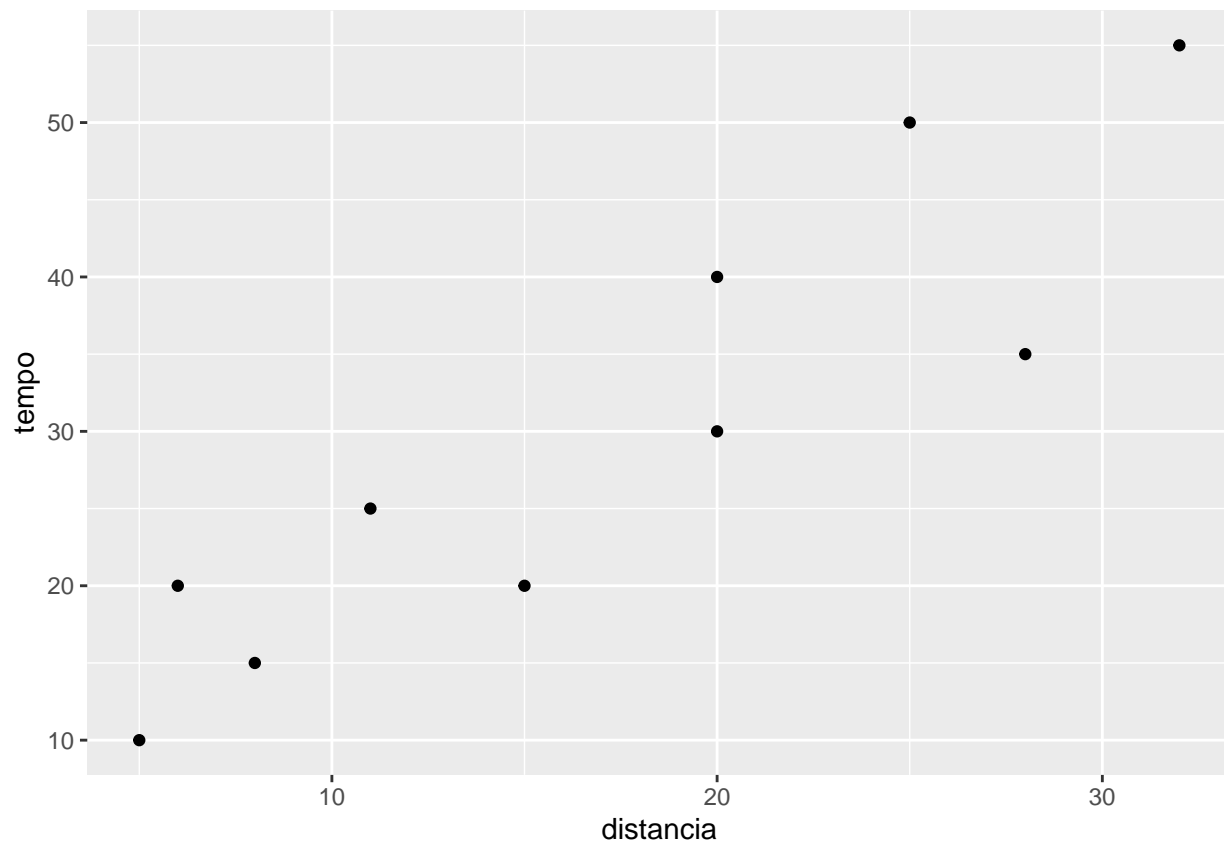
```
dados %>%
  rename(tempo="Tempo para chegar à escola (minutos)",
          distancia="Distância percorrida até a escola (quilômetros)") %>%
  summarise(média_tempo=mean(tempo),
            média_distância=mean(distancia)) %>% kable()
```

média_tempo	média_distância
30	17

Na engrenagem do chunk, existem algumas opções para reportar o conteúdo no documento final. Por exemplo, é possível escolher somente o output, código + output ou mesmo não reportar aquele chunk.

Para gerar gráficos, o procedimento é semelhante. Como exemplo, vamos gerar um gráfico de dispersão entre tempo (eixo Y) e distância (eixo X).

```
dados %>%
  rename(tempo="Tempo para chegar à escola (minutos)",
          distancia="Distância percorrida até a escola (quilômetros)") %>%
  ggplot() + geom_point(aes(x=distancia, y=tempo))
```



Para exportar o documento criado, basta clicar na flecha ao lado do “Knit” e selecionar o tipo de documento que deseja.

Os formatos podem ser Word, PDF e HTML.

Atenção: antes de gerar documento em PDF, é necessário instalar a seguinte função no console do R:

```
tinytex::install_tinytex()
```

FIM!