

Elementos estéticos do gráfico

Transcrição

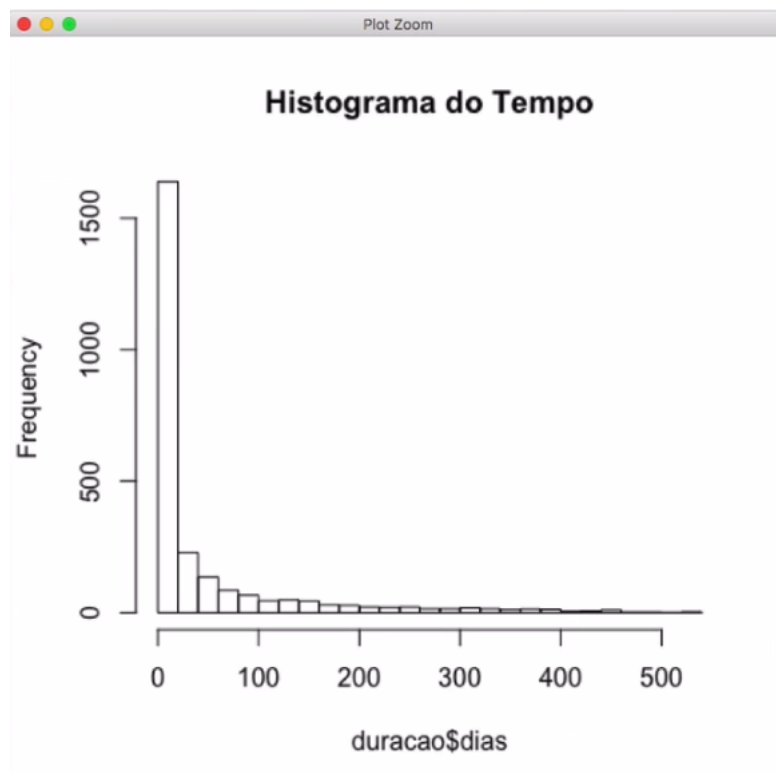
O histograma de 20 colunas melhora a interpretação do conteúdo do banco de dados, mas não é profissionalmente apresentável. Portanto, antes de apresentá-lo para a empresa que nos contratou, precisamos aprimorá-lo, alterando alguns elementos. Fecharemos a janela de exibição do histograma e continuaremos a edição do código `hist`.

Começaremos substituindo o **título** "Histogram of duracao\$dias" por "Histograma do Tempo", por meio do parâmetro `main`. Entre aspas ("), após inserirmos sinal de igual (=), digitaremos o novo título, lembrando que se quiserem podem escolher outro nome. Os nomes criados pelo programa são, por padrão, em inglês, mas optamos pelo português.

O código ficará da seguinte forma:

```
hist(duracao$dias, breaks = 20, main = "Histograma do Tempo")
```

Executaremos essa linha para conferir a mudança no título. Após aplicarmos o zoom e expandirmos a janela, a visualização do histograma fica assim:

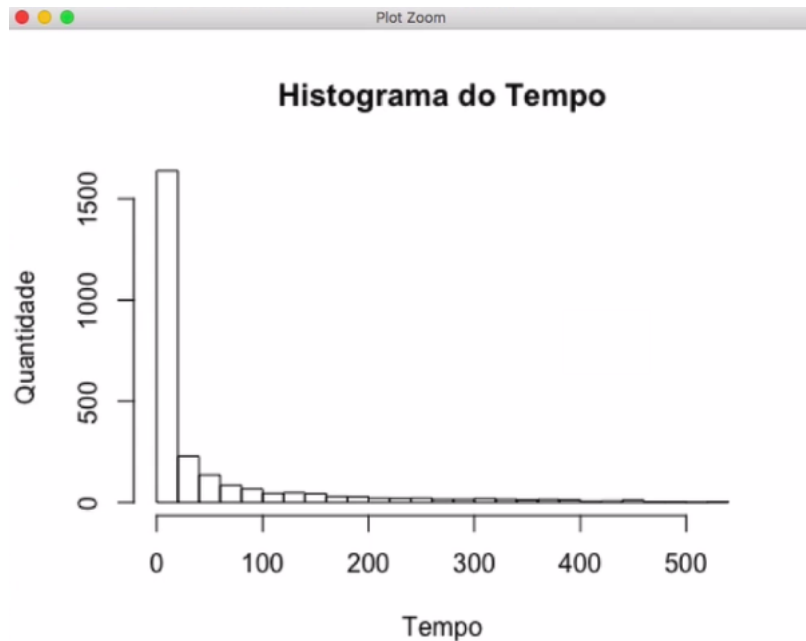


Isto é, o título foi alterado, melhorou, mas o eixo vertical ainda possui um elemento em inglês (`Frequency`), e o horizontal está com o nome da variável (`duracao$dias`). Mudaremos esses dois parâmetros também. De volta ao código `hist`, após o título do histograma, acrescentaremos vírgula (,) para trabalhar com um novo parâmetro.

No RStudio, o eixo vertical recebe o nome de `ylab`, sigla de *label* do eixo y, que em português significa "rótulo do eixo y". Sendo assim, acrescentaremos ao código `ylab = "Quantidade"`. O eixo horizontal é denominado `xlab`, e sua renomeação será estabelecida por meio de `xlab = "Tempo"`. O código ficará assim:

```
hist(duracao$dias, breaks = 20, main = "Histograma do Tempo", ylab = "Quantidade", xlab = "Tempo")
```

Ao executarmos essa linha, os títulos dos eixos do histograma serão alterados.



No lugar de "duracao\$dias" no eixo horizontal, encontramos "Tempo", e no eixo vertical, onde estava "Frequency", agora, temos "Quantidade". São essas as informações que queremos passar para a empresa. No eixo horizontal, temos o **tempo** para a conclusão de um curso, e no eixo vertical há a **quantidade** de cursos que levam esse determinado tempo para conclusão.

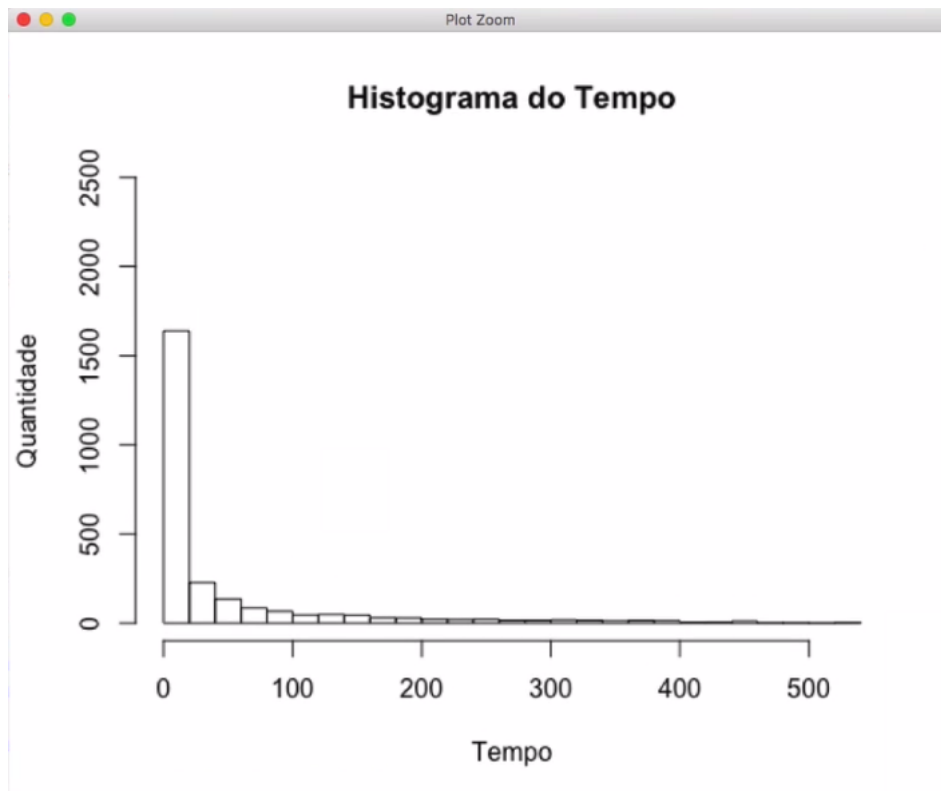
Também podemos alterar o parâmetro vertical: a primeira coluna do gráfico extrapola o limite (1500) do eixo vertical, nos deixando sem referência, o que pode prejudicar a apresentação dos dados. Para acertar isto, voltaremos a editar o código `hist`, acrescentando um limite para o eixo vertical por meio de `ylim`, seguido de sinal de igual (=) e um vetor (`c`) com dois parâmetros.

Como vimos anteriormente, um vetor com dois parâmetros é adicionado por meio de (`c`), seguido por parênteses preenchidos com os valores que desejamos estabelecer. Esse comando recebe o nome de "vetor" porque é um conjunto de informações agregadas em uma mesma linha ou coluna. No limite do eixo y, definiremos o **limite inferior** e **superior**. Como nenhum curso leva menos de 0 dias para ser concluído, o **limite inferior**, naturalmente, será 0.

Após virgula (,) estabeleceremos o limite superior. Veremos no gráfico que o limite superior escolhido pelo programa é 1500, mas temos uma quantidade maior de cursos sendo concluídos entre 0 e 20 dias. Sendo assim, aumentaremos o **limite superior** para 2500. O código ficará assim:

```
hist(duracao$dias, breaks = 20, main = "Histograma do Tempo", ylab = "Quantidade", xlab = "Tempo", ylim = c(0, 2500))
```

Ao executarmos a linha do código, teremos como resultado o seguinte gráfico:



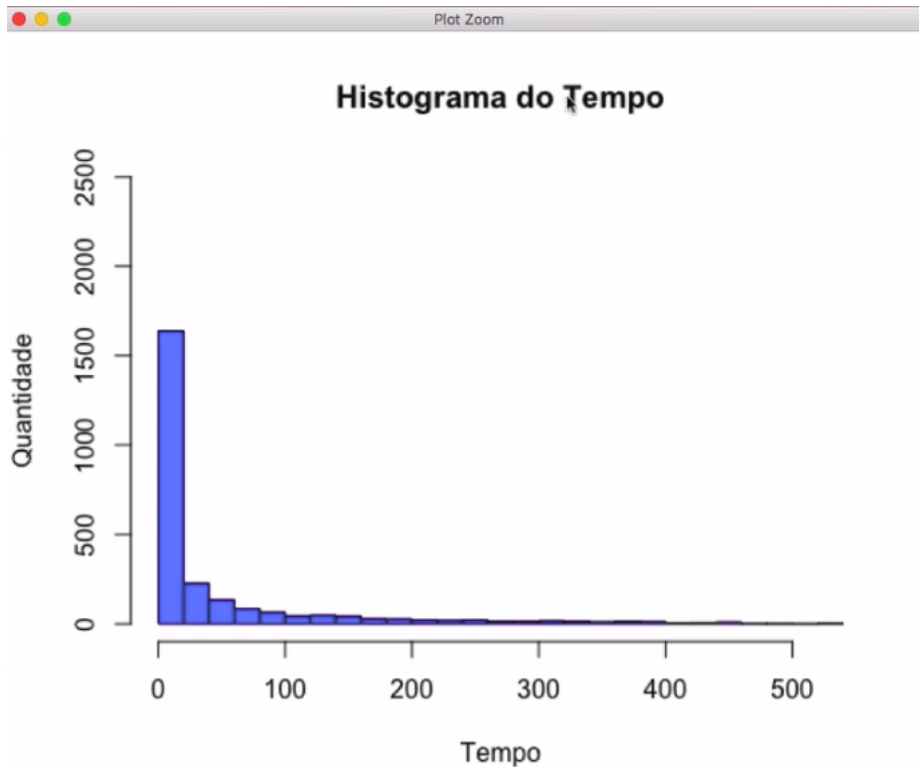
No eixo vertical, o limite foi alterado de 1500 para 2500, mantendo o topo da barra dentro do limite do gráfico.

Dependendo do software utilizado para a apresentação, você pode adaptar os limites novamente, é uma questão de adaptação estética. O importante é que a informação seja transmitida de forma rigorosa.

Por fim, poderemos colorir o gráfico, algo que ajuda em apresentações. Sobretudo, se a apresentação tiver identidade visual entre os slides, por exemplo. É possível colorir as colunas do histograma, voltando ao código `hist` e adicionando um parâmetro `col`, de *color*, cor em inglês e, entre aspas, especificaremos o nome da cor que queremos aplicar, `blue` (azul).

```
hist(duracao$dias, breaks = 20, main = "Histograma do Tempo", ylab = "Quantidade", xlab = "Tempo", col = "blue")
```

Ao executarmos, o programa produzirá um gráfico com barras azuis, como solicitamos no código:



Ficou melhor do que o gráfico anterior. Alteramos o número de quebras, o título do gráfico, os nomes dos eixos horizontal e vertical, o limite do eixo vertical e, por fim, a cor. Assim, produzimos um gráfico profissionalmente apresentável. Porém, ainda temos um problema: ele está dentro do RStudio, não o exportamos, e dificilmente abriremos o programa para reproduzi-lo durante uma apresentação.

Para exportá-lo, fecharemos as janelas de visualização abertas no RStudio e, no R Script, informaremos ao programa que o gráfico que produziremos deve ser salvo em um arquivo `.jpg` ou `.pdf`, dependendo do software utilizado, e da sua preferência. Se quisermos salvar como `.jpg`, na linha acima de `hist`, acrescentaremos `jpeg("histograma.jpg")`. Assim, informamos que estamos criando um arquivo, dentro do qual informamos o que foi colocado nesse arquivo e, por fim, o fecharemos, dizendo ao programa para não colocar mais nada ali.

Para isso, na linha abaixo de `hist`, acrescentaremos `dev.off()`. O código ficará assim:

```
jpeg("histograma.jpg")

hist(duracao$dias, breaks = 20, main = "Histograma do Tempo", ylab = "Quantidade", xlab = "Tempo")

dev.off()
```

Dessa forma, abriremos um arquivo por meio de `jpeg("histograma.jpg")`, dentro dele colocaremos o gráfico que desenvolvemos em `hist` e avisamos o programa que não colocaremos mais nada no arquivo com `dev.off()`. Podemos selecionar esse trecho de código para executar, mas antes executaremos linha por linha para conferir se não há erros. Ao executar a primeira linha, no Console, teremos como retorno:

```
> jpeg("histograma.jpg")
```

Indicando que está correto. No R Script, o cursor vai automaticamente para a próxima linha de código. Ao clicarmos em "Run", no Console, será executado o código do gráfico.

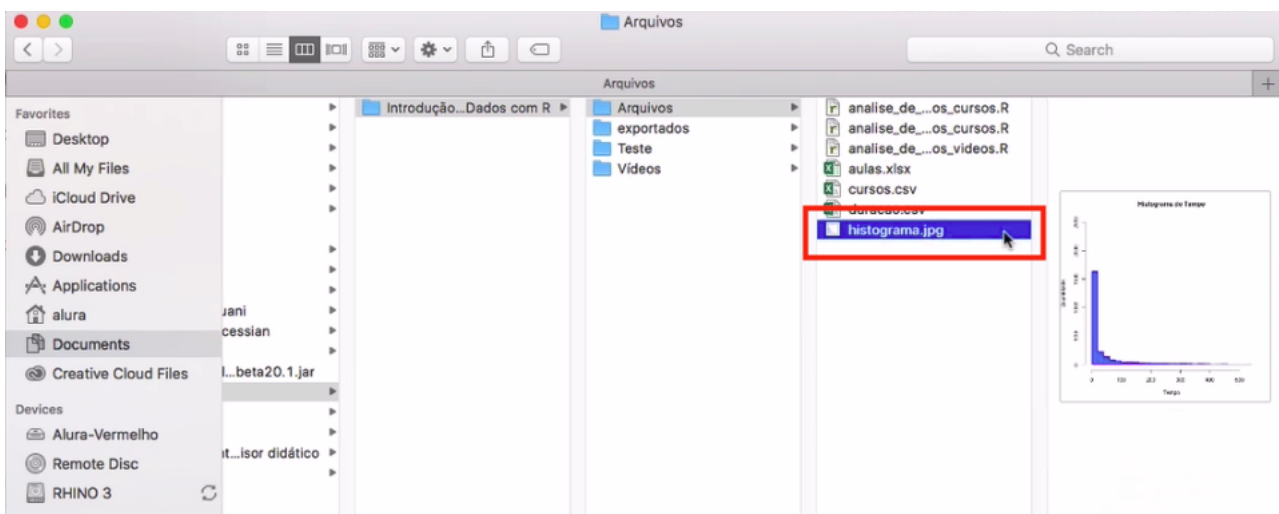
```
> hist(duracao$dias, breaks = 20, main = "Histograma do Tempo", ylab = "Quantidade", xlab = "Tempo")
```

Também está correto. Por fim, executaremos a última linha do trecho. O retorno no Console será o seguinte:

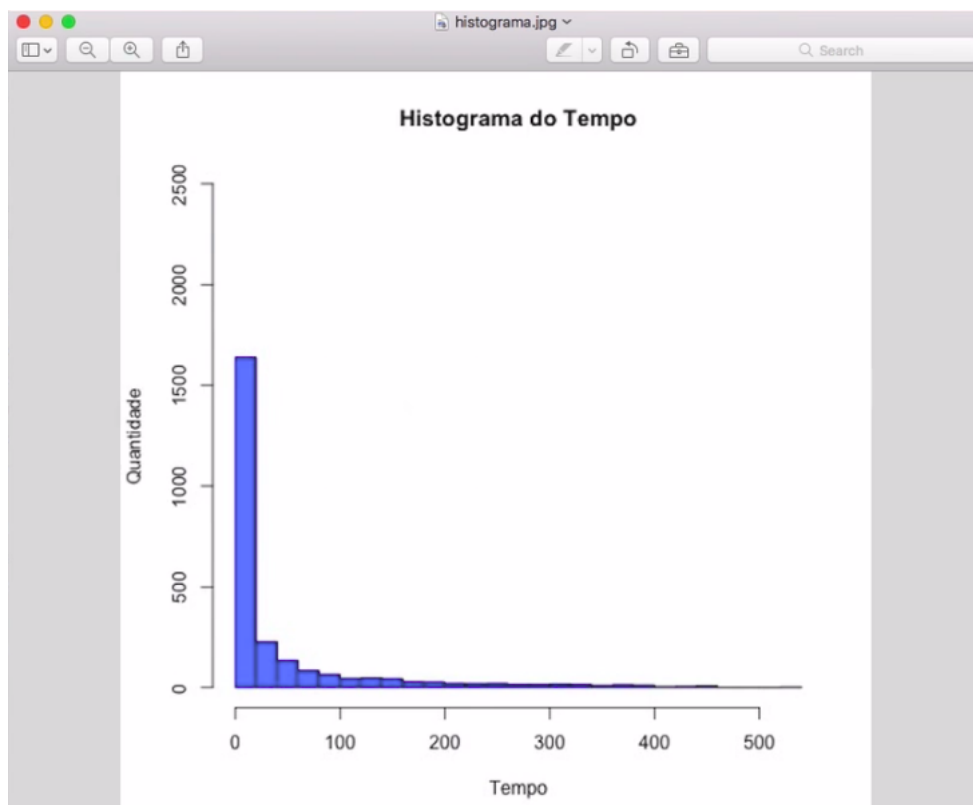
```
> dev.off()  
RStudioGD
```

2

Aparece uma mensagem informando que o arquivo foi fechado. Confirmamos que as linhas estão funcionando corretamente. Então, verificaremos se o arquivo foi salvo no diretório e tentaremos abri-lo. Sempre temos que conferir o trabalho. Localizamos o arquivo `histograma.jpg`.



Ao clicar para abrir, teremos exatamente o histograma que produzimos no RStudio:



Os parâmetros que inserimos permaneceram, assim como as cores. É necessário verificar sempre, pois o gráfico pode sofrer algumas alterações do RStudio para o programa de visualização, e a cor pode variar um pouco, entre outros. Não foi o caso do gráfico que desenvolvemos agora, a visualização dele em `.jpg` ficou igual a do RStudio.

Pronto! Agora, temos um histograma apresentável ao cliente.