05

Gráfico profissional

Transcrição

Produzimos um gráfico com um modelo não linear, suave. Estatisticamente, ele é muito bom e traz muitas informações, mas podemos aprimorá-lo para uma apresentação profissional. Para desenvolvermos uma estética mais apropriada, utilizaremos um comando de outra biblioteca do RStudio, a famosa `ggplot`, que vem de "*grammar of graphics*", em português, "gramática dos gráficos", que ajuda bastante na produção de gráficos estética e analiticamente. No R Script, tentaremos carregar a biblioteca por meio de:

```
library(ggplot2)
```

Ao executarmos, teremos um erro no retorno:

```
> library(ggplot2)
Error in library(ggplot2) : there is no package called 'ggplot2'
```

A mensagem diz que não há uma biblioteca chamada `'ggplot2'`. Isso acontece porque, na verdade, não a instalamos. No R Script, sua instalação é feita por meio do comando a seguir:

```
install.packages('ggplot2')
```

Executando-se o comando, a instalação é visualizada no Console. Uma vez instalada, para utilizá-la em outras ocasiões, basta carregá-la utilizando o comando `library`. Não é necessário instalar novamente. Se executarmos `library(ggplot2)` agora, a biblioteca será carregada e poderemos utilizar as funções contidas nela.

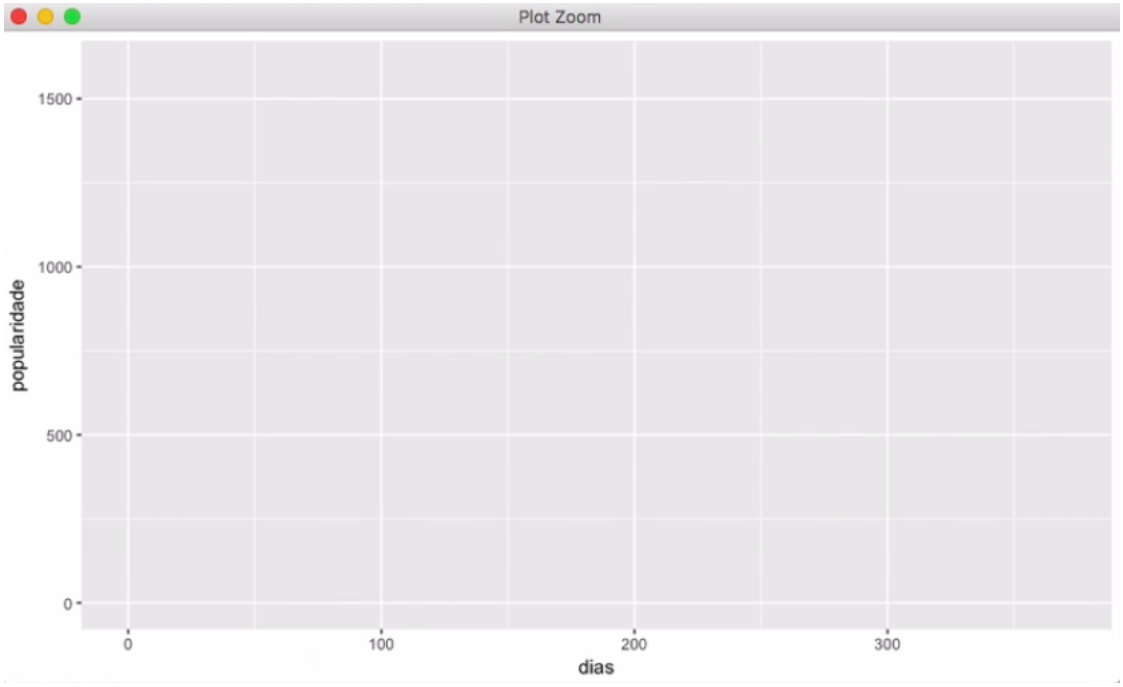
Começaremos a criar o gráfico, ao qual daremos o nome de `grafico` e atribuiremos (`<-`) a ele `ggplot`, que é o comando básico da biblioteca. Entre parênteses, especificaremos o banco de dados (`popularidade_e_duracao`), e em seguida, acrescentaremos vírgula (`,`), sendo que o parâmetro mais importante dessa função é `aes`, que vem de "*aesthetic*", em português, "estética". Nele, especificaremos as variáveis (`dias` e `popularidade`) que serão analisadas. Como já informamos o banco de dados que estamos trabalhando, não precisamos utilizar o cifrão (`$`).

```
grafico <- ggplot(popularidade_e_duracao, aes(dias, popularidade))
```

Ao executarmos o comando, verificaremos no Console que não houve problemas, porém, na janela inferior direita do RStudio, o gráfico continua o mesmo que desenvolvemos anteriormente. É uma característica dessa biblioteca ou do programa, então, para visualizarmos um objeto, precisaremos escrever seu nome no R Script:

```
grafico
```

Ao executarmos, teremos um novo gráfico na janela inferior direita. Clicaremos em "Zoom" e expandiremos a janela, obtendo-se a seguinte imagem:



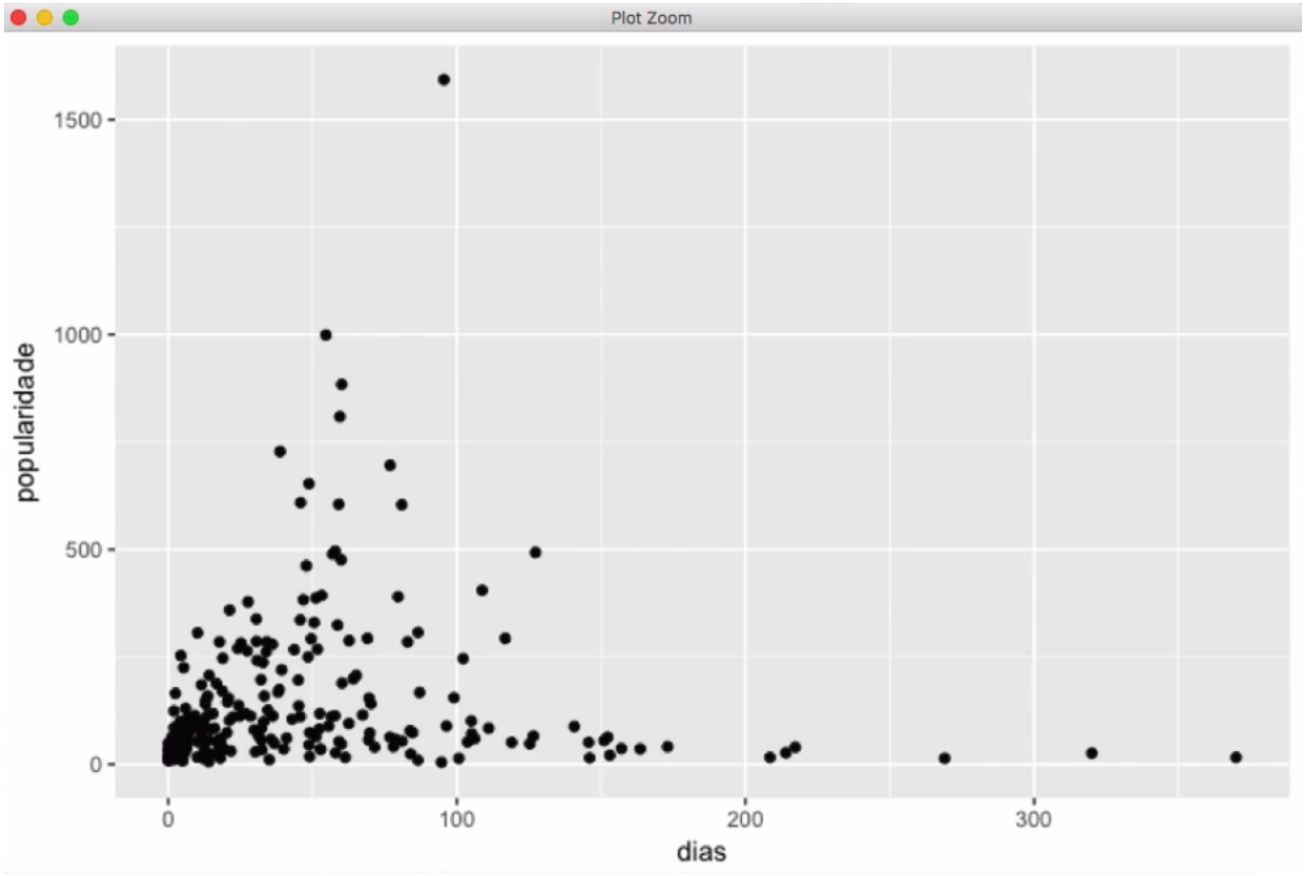
Porém, não há gráfico, e sim uma janela com as variáveis (`dias` e `popularidade`) solicitadas, e uma área em que poderemos plotar, mas nada foi plotado. Isso acontece porque a biblioteca `ggplot` funciona em etapas: primeiramente, criaremos um espaço, uma área onde colocaremos a informação. O comando `ggplot` não funciona como o `plot` das funções originais do RStudio, que procura um gráfico considerado adequado ao tipo de dados e o desenvolve quando executamos o comando.

Como vimos anteriormente, às vezes isso pode causar problemas, pois os gráficos não transmitem as informações de forma eficiente. O `ggplot` cria gráficos **somente após especificarmos o tipo que queremos**. Faremos isso agora, trabalhando em cima do mesmo objeto. Em uma nova linha do R Script, digitaremos:

```
grafico <- grafico + geom_point()
```

Notem que **adicionamos** (`+`) elementos ao gráfico, transformando o objeto (`grafico`) inicial, somando a ele `grafico` e `geom_point` . Esse último comando é utilizado para informar ao programa que estamos trabalhando com um modelo suave e não linear. Em seguida, abrimos e fechamos parênteses. Há uma etimologia do termo, que expressa "geometria de ponto", mas não nos aprofundaremos nisso.

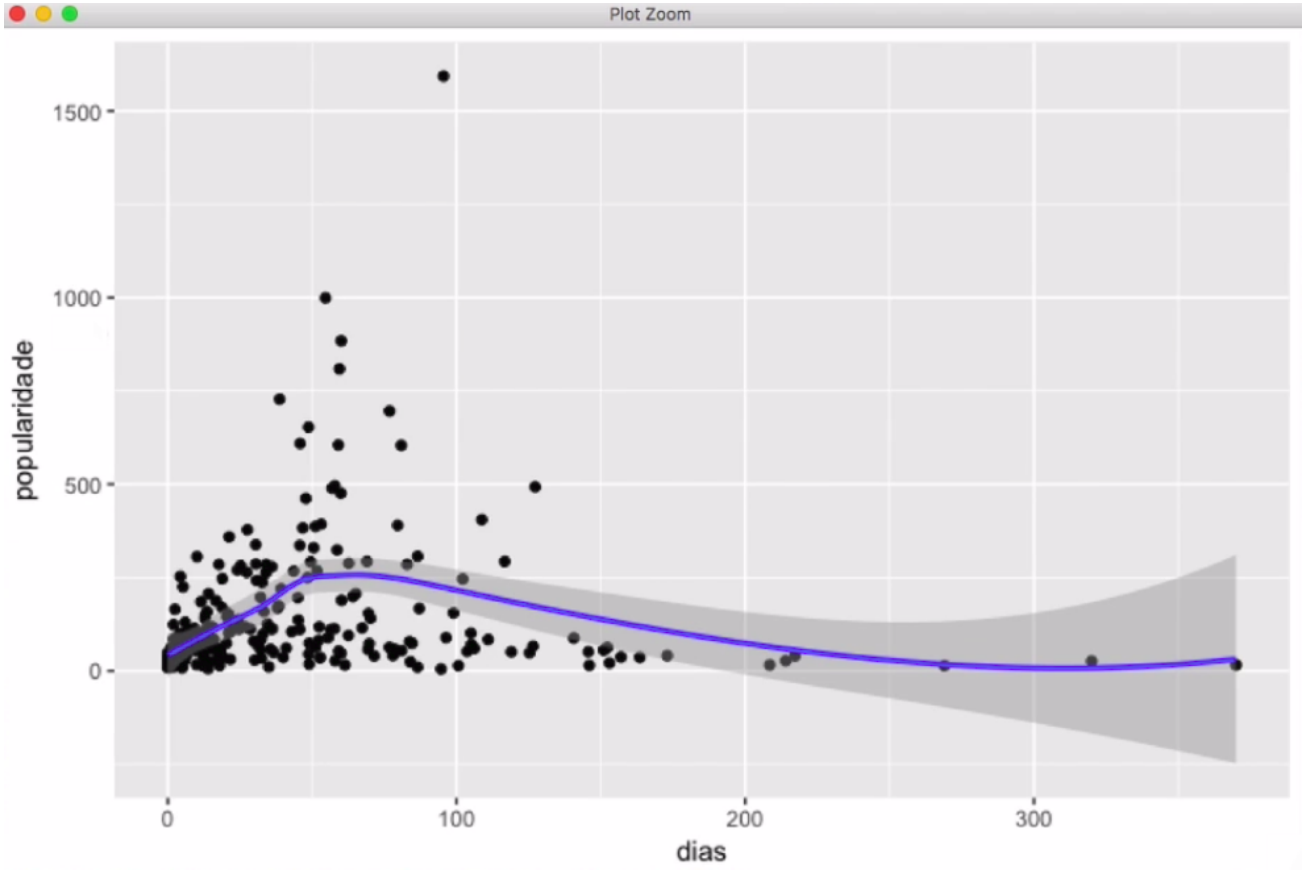
Executaremos o comando, e não teremos problemas no Console, aparentemente. O gráfico foi alterado e, para visualizá-lo, precisaremos digitar `grafico` no R Script e executá-lo novamente. Feito isso, clicaremos no "Zoom" na janela inferior direita do RStudio para obtermos a seguinte imagem:



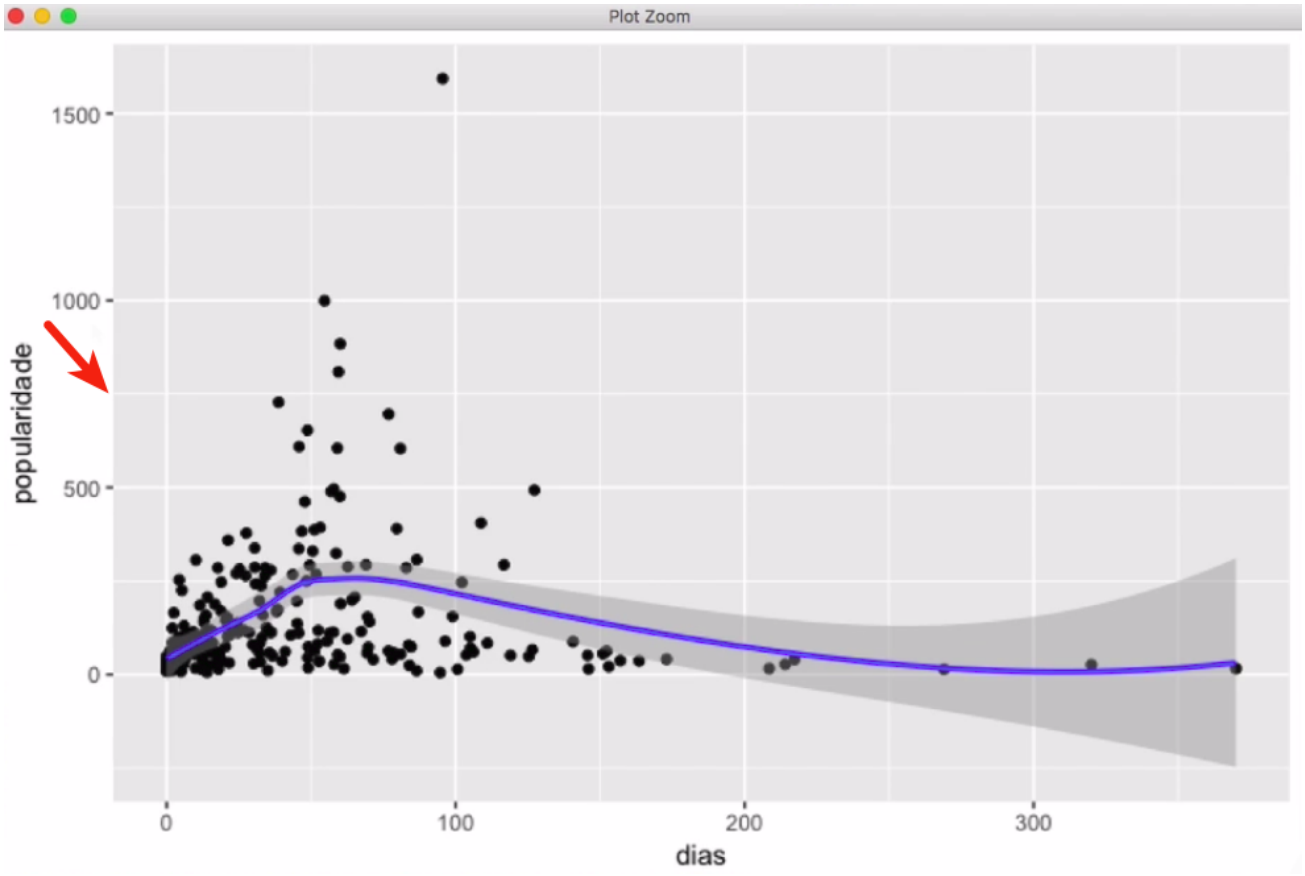
É o gráfico que fizemos anteriormente, visualmente mais agradável. Reparem como a sobreposição dos valores iniciais é menos problemática do que na solução original do RStudio. Neste momento, já vimos como salvá-lo e exportá-lo como imagem `.jpg` ou em `.pdf`, para acrescentá-lo a uma apresentação. Falta a curva, que não está no gráfico. Para inseri-la, repetiremos o procedimento anterior, adicionando (`+`) a curva ao gráfico em uma nova linha do R Script:

```
grafico <- grafico + geom_smooth()
```

Assim, atribui-se ao `grafico` o próprio `grafico` e o novo elemento `geom_smooth`, referente à geometria suave. Em seguida, digitaremos `grafico` novamente no R Script, para executá-lo, e conferiremos na janela inferior direita do RStudio que ele foi criado. Aplicaremos "Zoom" e expandiremos a janela para obtermos essa visualização:



Bacana! Criamos um gráfico com as mesmas informações de antes, ou seja, estatisticamente rigoroso e com as mesmas informações. No entanto, além disso o visual está muito bacana e profissional, com escala e fundo quadriculado que auxilia em inferências estatísticas. Por exemplo, no eixo vertical temos `500`, e dois quadrados acima temos `1000`. Pelos quadrados sabemos que `750` está exatamente no meio deles.



Ele trabalha automaticamente com cores, então os pontos estão pretos e a curva está azul. Há também uma **área sombreada** que é novidade. Não solicitamos, mas o programa a inseriu automaticamente. Ela é a **margem de confiança**, na qual esperamos que os pontos estejam distribuídos. Funciona como uma **medida de erro** da curva, e poderíamos explicá-la para a empresa da seguinte forma:

> "Por segurança, além da linha azul, considere a área sombreada na cor cinza. Ela indica a região em que se espera encontrar valores. A interpretação correta inclui os valores da área sombreada, da margem de erro".

Ela é menor no início do gráfico, entre `0` e `100` dias, no eixo horizontal. Depois de `100` dias, a área sombreada aumenta. Isso significa que no final do gráfico a chance de erro é grande e que pode ser arriscado fazermos alguma inferência sobre a popularidade dos cursos que demoram mais tempo até a conclusão. A segurança para afirmar algo é menor, porque são pouquíssimos cursos que duraram mais de `300` dias, e a margem de erro é maior nessa região pois há menos informações nela.

Assim, ela aumenta bastante a partir de `300` do eixo horizontal do gráfico. Temos mais segurança no início do gráfico, embora muitos pontos nessa região estejam fora da área sombreada, indicando que o modelo suave ajusta os pontos. Há uma margem de erro, mas ainda assim encontramos pontos fora da curva e da área sombreada na amostra.

Tudo isso é absolutamente esperado em uma análise de dados, como a que trabalhamos. Dificilmente os dados do mundo real são comportados e permitem uma descrição com uma curva ou margem de erro. Geralmente, encontraremos pontos fora da expectativa, da margem de segurança. Sendo assim, levaremos à empresa:

- a média e o pico da popularidade;
- a localização da margem de erro e como ela se comporta do começo ao fim da distribuição;
- quais são os casos discrepantes.

São valores a serem agregados à análise, e informações muito relevantes para a empresa que nos contratou.