

Problemas comuns encontrados nas visualizações em R

Os erros de programação são inevitáveis ao se escrever um código – especialmente quando se está começando a aprender uma nova linguagem de programação. Nesta leitura, você aprenderá como reconhecer erros comuns de programação ao criar visualizações usando **ggplot2**. Além disso, você encontrará links para recursos que podem ser usados para ajudar a lidar com problemas de programação com os quais poderá se deparar mais adiante.

Erros de programação comuns em ggplot2

Ao trabalhar com o código R em ggplot2, muitos dos erros comuns de programação envolvem questões com sintaxe, tais como caracteres nos lugares errados. Por isso é muito importante prestar atenção aos detalhes quando se está escrevendo um código de programação. Quando ocorre um erro no código que pode ser detectado por R, ele irá gerar uma mensagem de erro. As mensagens de erro ajudam a mostrar qual é o caminho correto, mas nem sempre ajudam a descobrir precisamente qual é o problema.

Vamos explorar alguns dos erros de programação mais comuns que podem ser encontrados em ggplot2.

Detecção de maiúsculas e minúsculas

O código R é sensível a letras maiúsculas e minúsculas. Caso você insira uma letra maiúscula no início da palavra de uma determinada função, sem querer, isso pode afetar o seu código. Aqui está um exemplo:

```
Glimpse(penguins)
```

A mensagem de erro permite que você saiba que R não consegue encontrar uma função chamada “Glimpse”.

```
Error in Glimpse(penguins) : could not find function "Glimpse"
```

Mas você sabe que existe uma função “glimpse” (com “g” minúsculo). Note que a mensagem de erro não explica exatamente o que está errado, mas aponta para um caminho genérico.

Com base nisso, é possível descobrir que esse é o código correto:

```
glimpse(penguins)
```

Equilíbrio entre parênteses e aspas

Outro erro de programação comum em R envolve parênteses e aspas. Em R, você precisa se certificar de que cada parêntese inicial em sua função tenha um parêntese final, e que cada aspa inicial tenha uma aspa final. Por exemplo, se você executar o seguinte código, nada acontecerá. R não criará um gráfico. Isso ocorre porque estão faltando dois parênteses finais na segunda linha do código:

```
ggplot(data = penguins) +  
  geom_point(mapping = aes(x = flipper_length_mm, y = body_mass_g
```

RStudio irá alertar sobre o problema. No lado esquerdo da linha do código em seu editor de código fonte de RStudio, você irá notar um círculo vermelho com um “X” branco no meio. Se você passar o cursor sobre o círculo, verá a seguinte mensagem:

RStudio avisa que existe um parêntese inicial sem um parêntese final. Por isso, para corrigir o código, você sabe que basta adicionar um parêntese final para cada parêntese inicial.

Este é o código correto:

```
ggplot(data = penguins) +  
  geom_point(mapping = aes(x = flipper_length_mm, y = body_mass_g))
```

Uso do sinal de adição para acrescentar camadas

Em ggplot2, você precisa acrescentar um sinal de adição (“+”) ao seu código quando quiser acrescentar uma nova camada ao seu gráfico. Colocar o sinal de adição no lugar errado é um erro comum. O sinal de adição deve sempre ser colocado ao final de uma linha de código, e não no início de uma linha.

Veja um exemplo de código que inclui o uso incorreto do sinal de adição:

```
ggplot(data = penguins)  
  + geom_point(mapping = aes(x = flipper_length_mm, y = body_mass_g))
```

Neste caso, a mensagem de erro de R identifica o problema e avisa que você deve corrigi-lo:

Error: Cannot use `+.gg()` with a single argument. Você inseriu um + acidentalmente em uma linha nova?

Este é o código correto:

```
ggplot(data = penguins) +  
  geom_point(mapping = aes(x = flipper_length_mm, y = body_mass_g))
```

Pode ser também que você tenha usado um símbolo de pipe em vez de um sinal de adição para acrescentar uma nova camada ao gráfico, da seguinte maneira:

```
ggplot(data = penguins)%>%  
  geom_point(mapping = aes(x = flipper_length_mm, y = body_mass_g))
```

Você receberá a seguinte mensagem de erro:

```
Error: `data` must be a data frame, or other object coercible by  
`fortify()`, not an S3 object with class gg/ggplot
```

Este é o código correto:

```
ggplot(data = penguins) +  
  geom_point(mapping = aes(x = flipper_length_mm, y = body_mass_g))
```

Ter em mente essas questões e prestar atenção aos detalhes enquanto escreve o código o ajudará a reduzir os erros e poupar tempo, para que você possa se concentrar em sua análise.

Recursos de ajuda

Todos cometem erros ao escrever um código, pois isso faz parte do processo de aprendizagem. Felizmente, existem muitos recursos úteis disponíveis tanto em RStudio quanto online.

- Documentação R
 - R possui uma documentação interna sobre todas as funções e todos os pacotes. Para saber mais sobre qualquer função de R, basta executar o código **?function_name**. Por exemplo, se quiser saber mais sobre a função `geom_bar`, digite:
`?geom_bar`
Ao executar o código, aparecerá uma entrada sobre “geom_bar” na aba de Ajuda (Help), no canto inferior direito de seu espaço de trabalho de RStudio. A entrada começa com uma seção de descrição, que fala sobre gráficos de barras:

O [site RDocumentation](#) disponibiliza boa parte desse mesmo conteúdo, em um formato ligeiramente diferente, com exemplos e links extras.

- Site ggplot2
 - O [site ggplot2](#), que faz parte da documentação oficial de tidyverse, é um ótimo recurso para tudo que está relacionado ao ggplot2. Ela inclui informações sobre os principais tópicos, exemplos úteis de códigos e links para outros recursos úteis.
- Pesquisa online
 - Outra opção é fazer uma pesquisa online com a mensagem de erro encontrada (inclua “R” e o nome da função ou do pacote nos termos de busca). É bem provável que alguém já tenha encontrado o mesmo erro e postado sobre ele online.
- A comunidade de R
 - Se os outros recursos não forem úteis, experimente procurar a comunidade de R online. Existem diversos fóruns e sites online muito úteis, onde as pessoas fazem perguntas e recebem ajuda, incluindo os seguintes:
 - [R for Data Science Online Learning Community](#)
 - [RStudio Community](#)
 - [Stackoverflow](#)
 - [Twitter \(#rstats\)](#)