

10

Corrigindo nossa variável de lançamento

Encontrar bases de dados com algum tipo de problema não é algo incomum, por isso precisamos estar sempre atentos com as informações que estamos usando, pois se elas forem ruins, nossas conclusões podem estar erradas.

Na nossa base de cadastro de produtos usamos a variável `data` para construir outra variável, a `flag_lancamento`, por isso é importante dar uma olhada com mais cuidado nela. Descobrimos que em algumas observações eu não tenho a data preenchida. Essa ausência de informação que uma variável pode ter nós chamamos de ***missing*** (“ausência” ou “falta”, em inglês). Aprendemos uma nova opção do `PROC FREQ`, a `missing`, que faz com que as observações que não estão preenchidas apareçam dentro da minha listagem de categorias como uma categoria própria, e assim usamos o código abaixo para ver que temos 3 observações *missings* na minha variável `data`:

```
PROC FREQ
  data = alura.cadastro_produto_v2;
  table data / missing;
RUN;
```

Data de lançamento					
Data	Frequency	Percent	Cumulative Frequency	Cumulative Percent	
-	3	0.60	3	0.60	
200501	14	2.80	17	3.40	
200708	7	1.40	24	4.80	

Vimos que podemos usar o comando `WHERE` em um *data step* para realizar uma operação conhecida como **filtro**, ou seja, para *filtrar* nossa base de dados e assim manter apenas as observações que atendam as condições do nosso filtro (o que vem depois do `WHERE`). Utilizamos este comando para construir uma base intermediária onde conseguimos manter nela apenas as minhas linhas onde a variável `Data` é missing:

```
DATA teste1;
set alura.cadastro_produto_v2;

where data = .;

RUN;
```

Assim conseguimos descobrir mais informações sobre estas datas não preenchidas, como, por exemplo, o nome de cada um dos jogos onde isso acontece. Como todas as cópias que temos com o mesmo nome (ou *título*) são cópias do mesmo jogo, todas elas deveriam ter a mesma data de lançamento. Assim, se encontrarmos outra cópia do mesmo jogo com a data de lançamento preenchida saberemos de fato a data que deveria estar preenchida nestes meus missings.

Usamos o comando `WHERE` novamente para filtrar as linhas onde a minha variável `Nome` é preenchida com um dos 3 nomes que preciso examinar:

```
DATA teste1;
set alura.cadastro_produto_v2;

where nome = "Soccer" OR
  nome = "Forgotten Echo" OR
  nome = "Fireshock";
```

```
RUN;
```

Usamos o operador lógico **Ou** (“*Or*” em inglês), para declararmos que desejamos as observações tenham o nome igual a “*Soccer*” *OU* “*Forgotten Echo*” *OU* “*Fireshock*” .

Mas aqui estamos buscando um conjunto de valores possíveis de uma única variável, assim não precisamos ficar repetindo a expressão `nome = <...>`, podemos simplesmente dizer ao SAS que queremos um conjunto de valores que esteja *dentro* de uma variável usando a expressão **IN**, da seguinte forma:

```
DATA teste1;
set alura.cadastro_produto_v2;

where nome in ("Soccer","Forgotten Echo","Fireshock");

RUN;
```

Obs: Podemos separar os termos dentro dos parênteses do `in` por vírgulas ou espaços.

Um `PROC FREQ` pode ser usado para checar as datas de cada título nesta base temporária, mas vimos que nem precisamos criar essa base, pois este filtro pode ser aplicado diretamente quando declaramos qual base desejamos usar como entrada do nosso comando! O comando é parecido com o que usamos no *data step*, mas ele fica dentro de parênteses, e colocamos um “igual” depois do `where` :

```
PROC FREQ
  data=alura.cadastro_produto_v2
    (where=(nome in ("Soccer" "Forgotten Echo" "Fireshock")));
  table nome*data
    /list missing;
RUN;
```

Note que também usamos as opções `list` e `missing` para gerar nossa listagem de frequências. Usando o comando dessa forma nos ajuda a obter a informação que desejamos sem precisar criar uma base intermediária adicional.

Falta agora consertar a nossa variável de data e, consequentemente, nossa variável `flag_lancamento` . Use os comandos `IF ... THEN DO` e `SELECT` para preencher os *missings* na variável de data e recalcule a variável de `flag_lancamento` . Para relembrar:

- O jogo ***Fireshock*** foi lançado em **201706**,
- O jogo ***Forgotten Echo*** foi lançado em **201411** e
- O jogo ***Soccer*** foi lançado em **201709**.

Não se esqueça de salvar a base, comentar e salvar o código.