10

Conhecendo a variável CEP

Vamos começar a trabalhar com uma nova base, a base de cadastro de clientes. Nela temos as informações das pessoas que já alugaram algum produto da *AluraPlay*.

Precisamos fazer um levantamento demográfico dos nossos clientes para saber em quais estados eles moram. Nosso cliente na *AluraPlay* pediu para primeiro priorizar os estados de São Paulo, Rio de Janeiro, Minas Gerais e Paraná. Não temos uma variável que especifique o estado em nossa base de clientes, mas temos o CEP. A partir de pesquisas no site dos Correios conseguimos descobrir que podemos saber a qual estado um determinado CEP pertence, e usar isso para resolver nosso problema.

> Para mais informações sobre como funciona a estrutura do CEP acesse este link (https://www.correios.com.br/para-voce/precisa-de-ajuda/o-que-e-cep-e-por-que-usa-lo/estrutura-do-cep). Podemos pesquisar a localidade de certas faixas de CEP por este link (http://www.buscacep.correios.com.br/sistemas/buscacep/buscaFaixaCep.cfm).

Após uma pequena pesquisa, conseguimos levantar a seguinte relação entre Estado e CEP (Conseguimos inclusive dividir o estado de SP em Grande SP e Interior.):

- Grande SP: 01000-000 a 09999-999
- Interior de SP: 10000-000 a 19999-999
- Rio de Janeiro: 20000-000 a 28999-999
- Minas Gerais: 30000-000 a 39999-999
- Paraná: 80000-000 a 87999-999

Certo, já temos aqui o que precisamos. Vamos escrever um *data step* com vários condicionais para atribuir os valores devidos à minha variável de `Estado` :

```
DATA teste1;
set alura.cadastro_cliente;

    if "01000-000" <= cep <="09999-999" then
        Estado="Grande SP";
    else if "10000-000" <= cep <="19999-999" then
        Estado="Interior SP";
    else if "20000-000" <= cep <="28999-999" then
        Estado="Rio de Janeiro";
    else if "30000-000" <= cep <="39999-999" then
        Estado="Minas Gerais";
    else if "80000-000" <= cep <="87999-999" then
        Estado="Paraná";
    else
        Estado="Demais estados";

RUN;
```

Depois de executar o código, vamos analisar o resultado que obtemos nesta nossa variável...

Estado	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Demais es	22	27.50	22	27.50
Grande SP	23	28.75	45	56.25
Interior	8	10.00	53	66.25
Minas Ger	5	6.25	58	72.50
Paraná	3	3.75	61	76.25
Rio de Ja	19	23.75	80	100.00

Não ficou bom, não é? Queríamos que o nome dos estados aparecessem por completo, não "Demais es", "Rio de Ja", etc. O que foi que aconteceu aqui?

Ao contrário de outras linguagens de programação em que as variáveis devem ser declaradas e definidas previamente para que possam ser usadas, o SAS define as características de uma variável a partir da primeiro momento em que ela é usada ou algum valor lhe é atribuído. No caso da nossa variável de estado, a primeira atribuição que fazemos a ela é `Estado="Grande SP";` . A partir deste comando o SAS define como será a variável de Estado, por exemplo, por ele vemos que ela é uma variável do tipo caractere pois ela recebe um texto, e esse texto, incluindo o espaço, possui 9 caracteres (ou 9 elementos). Ao contrário das variáveis numéricas, que por padrão são criadas já com o tamanho máximo, as variáveis de texto por padrão são criadas com o tamanho específico que lhe é atribuído. No nosso caso, 9, e assim o SAS define a variável `Estado` como uma variável caractere de tamanho 9. Mas isso não é verdade, pois os demais conteúdos que quero atribuir para minha variável são maiores que isso. Por exemplo, eu precisaria de um texto de tamanho 14 para escrever "Rio de Janeiro".

Para resolver este problema podemos usar o recurso de declarar a variável previamente, de acordo com as características que eu sei que ela precisa ter. Podemos fazer isso usando o comando `FORMAT` dentro do meu *data step*. Este comando é usado da seguinte forma:

```
format <nome da variável> <formato da variável>;
```

O nome da variável já sabemos que é `Estado` . O formato podemos saber a partir da documentação do SAS (https://v8doc.sas.com/sashtml/lgref/z1263753.htm), mas também podemos notar o padrão que os formatos de variáveis de tipo caractere seguem (olhando as demais variáveis da minha base de clientes). Vemos que elas começam com um cifrão (`$`), vem seguidas por um número e terminam com um ponto. O número é justamento o tamanho da variável, e este é um possível formato de variáveis caractere.

Sabendo isso, podemos acrescentar esse `FORMAT` ao nosso *data step*, que fica assim:

```
DATA teste1;
set alura.cadastro_cliente;

format Estado $14.;

    if "01000-000" <= cep <="09999-999" then
        Estado="Grande SP";
    else if "10000-000" <= cep <="19999-999" then
        Estado="Interior SP";
    else if "20000-000" <= cep <="28999-999" then
        Estado="Rio de Janeiro";
    else if "30000-000" <= cep <="39999-999" then
        Estado="Minas Gerais";
    else if "80000-000" <= cep <="87999-999" then
        Estado="Paraná";
    else
        Estado="Demais estados";
```

```
RUN;
```

Isso resolve o problema de "recorte" das variáveis, mas com certeza não é a melhor solução. Primeiro, o código ficou bastante extenso e restrito apenas a este *data step*. Se eu precisar determinar várias vezes o estado a partir do CEP eu terei problemas, pois precisarei repetir todos esses meus condicionais o tempo todo. E estamos falando de só quatro estados...

Segundo, precisamos trabalhar melhor com CEP. Apesar de ele estar armazenado como uma variável caractere, precisamos apenas dos dois primeiros números dele (podemos determinar qualquer um desses estados usando só a região e sub-região).

Escreva um código SAS que, usando o comando `substr` (de ***substring***, o comando de "subtexto") e o comando `input`, isola apenas os dois primeiros dígitos do CEP e o transforma em um número.