

09

## Usando formatos de várias formas

Vimos alguns comandos adicionais nesta aula que nos ajudam a manipular uma base, como:

- drop : Descarta da base as variáveis declaradas.
- keep : Mantém na base *apenas* as variáveis declaradas.

Vimos que estes comandos podem ser usados de uma forma parecida com a que usamos o comando `where` na hora de chamarmos uma base de entrada (colocando entre parênteses, separados por um sinal de igual).

Vimos também o comando `obs`, muito útil para agilizar os testes em bases muito grandes, que mantém apenas o número de linhas declarado. Podemos fazer o SAS voltar a usar todas as linhas das bases usando o comando `obs=max`.

Nós também criamos o nosso próprio formato que determina o estado a partir dos dois primeiros números do CEP, usando o `PROC FORMAT`. Como já obtemos os dois primeiros dígitos do CEP em formato numérico na aula anterior (na variável `precep`), o formato que criamos foi o seguinte:

```
PROC FORMAT;
  value estados_
    low - 09 = "Grande SP"
    10 - 19 = "Interior SP"
    20 - 28 = "Rio de Janeiro"
    30 - 39 = "Minas Gerais"
    80 - 87 = "Paraná"
    OTHER     = "Demais estados";
RUN;
```

Aplicamos esse formato na variável `precep` usando o comando “irmão” do `input`, o `put`, e depois vimos que podemos aninhar todos estes comandos em uma única linha, e obtivemos o seguinte código:

```
DATA teste1;
  set alura.cadastro_cliente;

  Estado = put(input(substr(cep,1,2),best.),estados_.);

RUN;
```

Usando o `PROC FREQ` também, crie dois formatos: Um que relaciona os intervalos do CEP dos estados que desejamos estudar em formato de texto a um número (que corresponde à posição que queremos que aquele estado apareça em nossa listagem); outro que relaciona os números definidos acima aos estados correspondentes.

Depois, usando os comandos `input` e `format` faça com que o nome dos estados apareçam como uma “máscara” sobre seus números correspondentes. Lembre-se de que os formatos que possuem textos como entrada devem, no `PROC FORMAT`, serem declarados com o parâmetro “irmão” do `VALUE`, o `INVALUE`.

