
Universidade de São Paulo
Escola Superior de Agricultura “Luiz de Queiroz”
Seção Técnica de Informática

A Anatomia de um Programa SAS

Marcelo Corrêa Alves

Fundamentos

– Piracicaba / 2016 –

A Anatomia de um Programa SAS

Sumário

1	<i>Introdução</i>	3
2	<i>Objetivos</i>	3
3	<i>Um programa SAS padrão</i>	3
4	<i>O Data Step</i>	4
5	<i>O Proc Step</i>	5
6	<i>Interpretação do programa pelo SAS</i>	5
6.1	<i>Composição do Step</i>	6
6.2	<i>O Data Step</i>	6
6.3	<i>O Proc Step</i>	7
7	<i>Resultados obtidos em cada Step</i>	7
7.1	<i>Resultado do Data Step</i>	7
7.2	<i>Resultado do primeiro Proc Step</i>	8
7.3	<i>Resultado do segundo Proc Step</i>	9

1 Introdução

Um programa de computador é uma sequência de comandos escritos em uma linguagem de programação e cada linguagem de programação tem o seu vocabulário e suas regras de redação. Antes de nos dedicarmos ao estudo de cada um dos comandos que compõem a linguagem SAS, faremos uma abordagem à longa distância, ou seja, estudaremos as grandes partes que compõem um programa SAS básico.

Nesse capítulo vamos estudar as grandes partes que compõem um programa SAS: o *Data Step* e o *Proc Step*.

Esse capítulo não pressupõe qualquer conhecimento anterior, sendo o primeiro capítulo para quem começará a programar em SAS. Caso deseje executar os programas, são necessárias habilidades para operar *O Ambiente de Trabalho SAS 9.3*.

2 Objetivos

Nesse capítulo objetiva-se que você compreenda:

- O que é o *Data Step*
- O que é o *Proc Step*
- Comentários

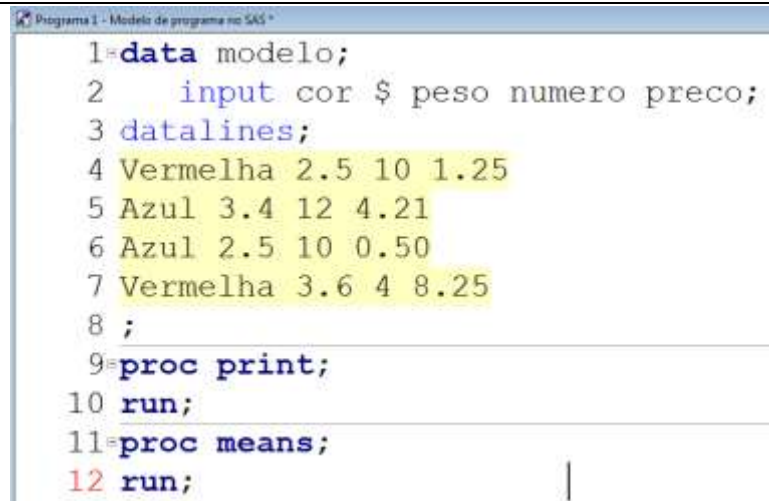
3 Um programa SAS padrão

Para estudarmos as grandes partes que compõem um programa SAS vamos partir de um programa “padrão” (programa 1).

Programa 1. Modelo de programa SAS.

```
data modelo;
  input cor $ peso numero preco;
datalines;
Vermelha 2.5 10 1.25
Azul 3.4 12 4.21
Azul 2.5 10 0.50
Vermelha 3.6 4 8.25
;
proc print;
run;
proc means;
run;
```

Quando digitado na janela Editor, o programa se parece bastante com a forma como é representado acima, exceto pelas linhas de topo e de base que delimitam o programa em relação ao texto, conforme é ilustrado na figura 1.



```
1 data modelo;
2 input cor $ peso numero preco;
3 datalines;
4 Vermelha 2.5 10 1.25
5 Azul 3.4 12 4.21
6 Azul 2.5 10 0.50
7 Vermelha 3.6 4 8.25
8 ;
9 proc print;
10 run;
11 proc means;
12 run;
```

Figura 1. Modelo de um programa do SAS digitado no Editor.

Antes de mais nada, cabe esclarecer que os números que aparecem à esquerda dos comandos, assim como os sinais de menos que aparecem à esquerda dos comandos **data** e **proc**, não fazem parte do programa, são apenas indicadores do *Editor* para facilitar a visualização e edição dos programas.

Observe que os traços horizontais separam o programa em três partes, a primeira que se inicia no comando **data** e que termina no sinal de ponto e vírgula solitário. A segunda parte é composta por duas linhas e a terceira parte, também é composta por duas linhas de comandos.

Cada uma dessas partes recebe o nome de Passo (*Step*), podemos dizer, então que esse programa é formado por três passos ou então, por três *Steps*.

O primeiro *Step*, como principia pelo comando **data**, é chamado de um *Data Step* e os dos *Steps* seguintes, como se iniciam pelo comando **proc**, cada um deles é um *Proc Step*.

Cada *Step* é um bloco de comandos com uma finalidade específica e eles se sucedem nos programas SAS de acordo com a necessidade do programador e um programa pode ter quaisquer números de *Data* e *Proc Steps*.

4 O *Data Step*

O *Data Step* assim é chamado porque nele se faz a leitura, a transformação e a carga dos dados em uma ou mais tabelas de dados do SAS (*SAS Data Set*). O *SAS Data Set* é o dispositivo no qual os dados devem ser armazenados para que o SAS possa processá-los por meio de suas ferramentas de análise de dados. Trata-se de um arquivo que fica armazenado em algum dispositivo do computador e que

O *Data Step* é iniciado pelo comando `data` e, em seguida, um grande número de comandos podem se suceder alguns dos quais vamos estudar em outros capítulos já que nesse, não pretendemos entrar no detalhamento dos comandos.

O final de um *Data Step* ocorre quando é colocado no programa um outro comando `data` (o qual iniciará um novo *Data Step*), um comando `proc` que iniciará um *Proc Step* ou um comando `run` que não inicia nada, mas conclui¹ um *Step* qualquer (*Data* ou *Proc*).

5 O Proc Step

O *Proc Step* se caracteriza pela ativação de um procedimento (*Procedure*) e o SAS conta com um número muito grande de procedimentos. Cada procedimento é identificado por um nome e é composto por um conjunto de comandos e de opções inerentes à sua utilidade.

No programa 1 é especificada a ativação de dois procedimentos: o `proc print` e o `proc means`.

Há uma gama muito grande de procedimentos e os mais importantes são procedimentos analíticos (aqueles que executam análises de dados). Para que esses procedimentos possam ser usados, é imprescindível que os dados que serão analisados estejam em um *SAS Data Set*.

Tanto o procedimento `print` como o procedimento `means` que aparecem no programa 1, processam os dados que estão em um *SAS Data Set* denominado Modelo.

Mas existem procedimentos que não são analíticos e esses são independentes da existência prévia de um *SAS Data Set* para que possam ser usados. Existem, inclusive, procedimentos que geram o *SAS Data Set*, por exemplo, no caso de procedimentos que leem dados armazenados em bancos de dados.

6 Interpretação do programa pelo SAS

Em seguida, passamos a estudar como é compreendido o programa pelo SAS, primeiramente esclarecendo que o SAS faz a leitura da esquerda para a direita e de cima para baixo, ou seja a primeira palavra que ele lerá é a palavra `data`, depois a palavra `modelo`, depois a palavra `input` até o final do programa, na linha número 12 onde está a palavra `run`.

¹ Existem alguns procedimentos que não se encerram definitivamente no comando `run`. São os procedimentos chamados de interativos que executam alguns comandos, mas permanecem aguardando por novos comandos. Nesses casos usa-se o comando `quit` para encerrar definitivamente o Step sem iniciar outro, com um comando `data` ou `proc`, que também encerram os procedimentos interativos.

6.1 Composição do Step

Cada *Step* é formado por comandos, opções, identificadores, operadores, constantes e sinais. O Data Step, adicionalmente pode conter os dados brutos, como no programa 1. Esse programa não apresenta todos esses componentes, justamente para simplificar essa nossa primeira abordagem.

No programa em questão, cada linha (excetuando-se as linhas que contém os dados brutos e a indicação do final deles com o ponto e vírgula solitário) principia com um comando. Na linha 1: comando **data**; linha 2: comando **input**; linha 3: comando **datalines**. Linhas 9 e 11: comando **proc** e linhas 10 e 12: comando **run**.

Cada comando termina no sinal de ponto e vírgula que foi inserido ao final de cada linha. O fato de mudar de linha, não implica em acabar o comando, aliás há comandos muito longos que se estendem por muitas linhas.

Tomemos por exemplo o comando **input**; há 4 identificadores à sua direita e se fossem necessários 30 identificadores, iríamos listando o identificadores até o fim de cada linha e prosseguindo na linha inferior, até que todos fossem digitados quando informaríamos o fim do comando colocando um único sinal de ponto e vírgula após o último identificador.

6.2 O Data Step

O *Data Step* do programa 1 conta com três comandos: **data**, **input** e **datalines**. Nenhuma opção é especificada, mas são especificados 5 identificadores: modelo, cor, peso numero e preco.

Comandos e as opções fazem parte do vocabulário do SAS, ou seja, são criadas pelos desenvolvedores do Sistema SAS e não podem ser modificados. Já os identificadores são criados pelos programadores, ou seja, os termos modelo, cor, peso, numero e preco foram estabelecidos por mim, no momento em que desenvolvia o programa.

Dois sinais foram usados: o sinal de cifrão (\$) e o sinal de ponto e vírgula (;).

Ainda compondo o *Data Step*, há uma região nesse programa que vai desde a linha seguinte à do comando **datalines** até a linha que contém um “;” solitário, logo na primeira coluna e que aparece com fundo amarelado no editor do SAS. Essa região contém os dados brutos, por isso, não contém comandos ou símbolos, mas apenas os dados.

Os comandos instruem o SAS em relação ao que ele deve fazer, é uma ordem que será executada. As opções, os identificadores fornecem informações necessárias para que os comandos sejam executados de acordo com interesses específicos do programador.

Os símbolos tem uma gama muito grande de utilidades e eles tem o uso determinado pelas regras da linguagem de programação (sintaxe).

Uma característica importante do Data Step é a estrutura repetitiva que está implícita na interpretação do comando SAS. O comando **data** é executado uma única vez, o comando **datalines** identifica a região do programa onde estão os dados brutos, não sendo um comando que se repete, o comando **input** e futuramente outros comandos que são usados no *Data Step* são executados mais de uma vez.

Como os dados brutos comportam a criação de 4 observações no *SAS Data Set*, o comando **input** é executado 4 vezes, a cada execução uma observação é lida e armazenada em uma estrutura auxiliar, o *Data Vector*. Ao final do *Data Step* os dados do vetor são transferidos à tabela de dados (*SAS Data Set*).

Esse conhecimento será especialmente útil quando outros comandos forem incorporados ao programa e não apenas o comando **input**, mas todos os comandos do *Data Step* serão executados uma vez para cada observação.

6.3 O Proc Step

Os dois procedimentos utilizados no programa 1 são analíticos, por isso, ambos exigem que exista um *SAS Data Set* disponível para que eles possam processar os dados nele armazenado.

É pelo motivo acima descrito que o *Data Step* foi escrito, no programa, antes do *Proc Step*.

Cada procedimento se associa a um tipo específico de processamento, por isso, toda vez que usamos o procedimento `print` obteremos uma listagem dos dados armazenados em um *SAS Data Set* e toda vez que usamos o procedimento `means`, obtemos como resultado, estatísticas básicas (média, desvio padrão, ...).

De maneira similar ao *Data Step*, o *Proc Step* também se encerra² em um comando **data**, em outro comando **proc** ou no comando **run**.

7 Resultados obtidos em cada Step

Se imaginarmos executar cada passo do programa, primeiro o *Data Step*, e depois cada um dos procedimentos (*Proc Step*), obteremos resultados distintos oriundos de cada um deles.

7.1 Resultado do Data Step

O *Data Step* criará um *SAS Data Set* o que é visualizado na janela *Log* e na janela *Contents Only* apresentada na figura 2.

² Existem alguns procedimentos que ficam em estado latente mesmo após terem sido executados.

```

Log - Output
101 data modelo;
102   input cor $ peso numero preco;
103   datalines;

NOTE: The data set WORK.MODELO has 4 observations and 4 variables.
NOTE: DATA statement used (Total process time):
      real time           0.03 seconds
      cpu time            0.00 seconds

108 ;

```

Figura 2. Representação do resultado obtido do processamento do *Data Step* do programa 1.

A janela *Log* mostra notas evidenciando que um data set denominado WORK.MODELO tem 4 observações e 4 variáveis ao mesmo tempo em que surgiu um ícone chamado Modelo na janela do Explorer, entretanto, nenhum resultado é exibido na janela Output em função dos comandos que aparecem listados na janela *Log*.

Note que por padrão, os dados brutos não são exibidos na janela *Log*.

7.2 Resultado do primeiro Proc Step

O primeiro *Proc Step* se refere à ativação do procedimento **print** e o resultado obtido desse processamento é ilustrado na figura 3.

```

Log - Output
109 proc print;
NOTE: Writing HTML Body file: sashtml.htm
110 run;

NOTE: There were 4 observations read from the data set WORK.MODELO.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           2.15 seconds
      cpu time            0.20 seconds

```

Obs	cor	peso	numero	preco
1	Vermelha	2.5	10	1.25
2	Azul	3.4	10	4.21
3	Azul	2.5	10	8.58
4	Vermelha	3.5	6	8.25

Figura 3. Representação do resultado obtido do processamento do primeiro *Proc Step* do programa 1 (**proc print**).

Imaginando que o programa fosse apenas o *proc print*, observaríamos uma listagem que mostra os dados que foram armazenados no SAS Data Set denominado modelo que está sendo exibida na janela *Results Viewer*.

Na janela *Log* vemos uma indicação de que um arquivo sashtml.htm foi criado para conter o relatório que está sendo exibido e que quais foram os dados lidos (4 observações lidas do data set WORK.MODELO) para a confecção da listagem.

Na janela *Results*, à direita da tela, observamos um índice mostrando que há um resultado disponível e que é identificado pelo nome do procedimento que o gerou *Print* seguido do nome do título que aparece na listagem *The SAS System (Print:The SAS System)*.

7.3 Resultado do segundo Proc Step

Cada procedimento é executado a seu tempo, ou seja, depois que o primeiro procedimento é executado pelo SAS e a listagem dos dados é exibida nas áreas de resultados, o segundo procedimento será interpretado e executado, gerando um segundo processamento.

Em outras palavras, concluída a execução do **proc print**, o SAS executa o **proc means** e este procedimento é especificado no programa quando se deseja obter estatísticas básicas tais como o número de observações processadas, a média, o desvio padrão, o menor valor e o maior valor variáveis numéricas. Outras estatísticas podem ser especificadas nesse mesmo procedimento, mas como não foram especificadas quaisquer estatísticas o SAS exibe as estatísticas default, conforme pode ser observado na figura 4.

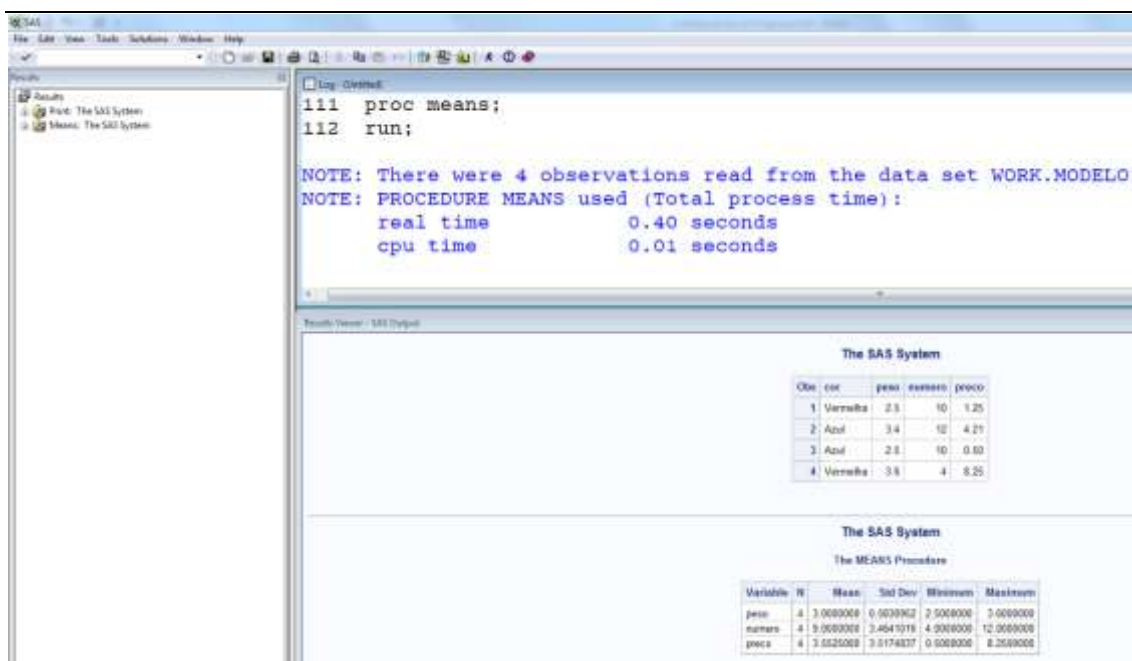


Figura 4. Representação do resultado obtido do processamento do segundo *Proc Step* do programa 1 (**proc means**).

Note que as estatísticas padrão (default) são exibidas para todas as variáveis numéricas do arquivo (peso, numero e preco).

O conhecimento em programação do *Proc Step* depende do estudo das opções e comandos que modificam o comportamento de cada procedimento. Nesse exemplo específico do **proc means**, opções podem ser usadas para determinar o cálculo de outras estatísticas e de mudanças na forma de apresentação dos resultados e outros comandos determinam uma ampla gama de modificações na forma do cálculo e exibição dos resultados.

Cada procedimento traz resultados coerentes com o objetivo de sua criação e a adoção de opções e a especificação de comandos próprios de cada procedimento é vital para que se aplique a técnica desejada de forma correta.