**Universidade de São Paulo**

**Escola Superior de Agricultura "Luiz de Queiroz"**

Seção Técnica de Informática

# O SAS Data Set – Básico

*Marcelo Corrêa Alves*

Data Step

– Piracicaba / 2016 –

# _____O SAS Data Set – Básico

## Sumário

# 1 Introdução

O SAS é um software desenvolvido com foco na tarefa de processar dados. Sendo assim, dispor de ferramentas para leitura, processamento e armazenamento de dados se torna visceral já que os dados, para sofrerem processos de análise devem estar organizados e disponíveis.

Nesse capítulo vamos dar os primeiros passos dentro da programação com a linguagem SAS, compreendendo como são usados os comandos que determinam criação de um SAS Data Set assim como, essa entidade chave, o SAS Data Set é estudado.

Em termos conceituais é importante que você tenha desenvolvido as competências tratadas no capítulo *A anatomia de um Programa SAS* e para que você possa desenvolver as atividades propostas nesse capítulo é recomendável que as habilidades inerentes ao capítulo *O Ambiente de Trabalho SAS 9.3* estejam, pelo menos, em um estado de desenvolvimento. É desejável, que o capítulo *As Janelas do SAS 9.3* também tenha sido estudado.

# 2 Objetivos

Nesse capítulo objetiva-se que você compreenda:

- O que é o SAS Data Set
- Como é organizado um SAS Data Set
- Como criar o SAS Data Set a partir de dados dentro do programa SAS
- Como disponibilizar os dados brutos dentro de um programa SAS
- Como ler os dados brutos

# 3 Os dados brutos

No ponto de vista do SAS, dados brutos são todos os dados que estão disponíveis e que podem ser lidos para popular um SAS Data Set. Os dados brutos podem estar em qualquer tipo de meio digital de armazenamento de dados.

Os dados brutos servirão, no SAS, única e exclusivamente para serem lidos de alguma forma e organizados em um arquivo com uma estrutura muito bem definida e que se chama SAS Data Set.

A construção de um SAS Data Set requererá, além dos dados brutos, regras claras e inequívocas que os organizem de forma coerente e estruturada.

## 3.1 Armazenamento de dados brutos

É muito comum termos dados, que na ótica do SAS são brutos, em planilhas eletrônicas (como o Microsoft Excel), em sistemas gerenciadores de bancos de dados (como o ACCESS, o Oracle, o Sybase) ou em arquivos texto e mesmo dentro de um programa SAS.

## 3.2 Dados brutos em SGBD

O uso de SGBD (Sistemas Gerenciadores de Bancos de Dados) requer que se faça um planejamento da estrutura de armazenamento dos dados. Os bancos de dados são organizados em Tabelas, Registros e Campos (na nomenclatura adotada pelo Microsoft Access) e alguns outros SGBD. Comumente as *Tabelas* e os *Campos* são identificados por meio de nomes definidos pelo responsável pela criação do banco de dados, e os *Registros*, comumente, são identificados por números. A criação e a administração de um banco de dados conta com profissionais específicos como o Projetista de Bancos de Dados e o DBA (Administrador de Bancos de Dados). A figura 1 ilustra a estrutura de um banco de dados do aplicativo Microsoft Access.

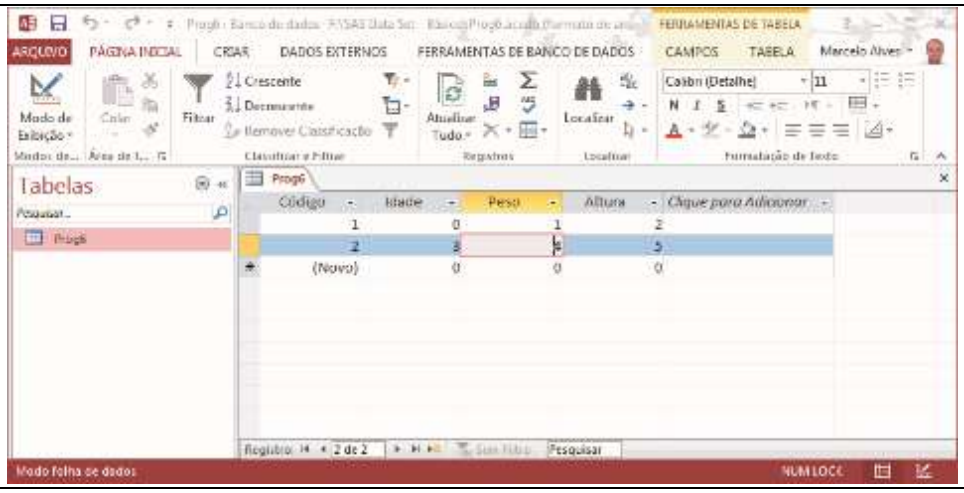


Figura 1. Imagem de um Banco de Dados do software Microsoft Access.

Observe na linha de título o nome do Banco de Dados (Prog6). À esquerda vemos a lista de tabelas desse banco de dados, nesse caso, o banco de dados conta com apenas uma tabela também chamada Prog2, cuja estrutura e dados são visualizados à direita sendo observáveis a presença de 4 campos (Código, Idade, Peso e Altura) e dois registros identificados pelos números do campo Código.

## 3.3 Dados brutos em planilhas eletrônicas

Um segundo mecanismo usado para armazenar dados são as planilhas eletrônicas, como por exemplo o Microsoft Excel. Em estágios iniciais da microinformática, as planilhas eletrônicas acumulavam a função de serem bancos de dados, por isso, podemos traçar um paralelo entre os dados armazenados nas planilhas eletrônicas e nos SGBD. O que se denomina como Tabela nos SGBD são chamados de *Planilhas* e estas também podem ser identificadas por nomes atribuídos pelo operador. Ao invés de Campos, nas planilhas encontramos as *Colunas* que diferente do que acontece no banco de dados, são identificadas por letras.

Por fim, ao invés dos *Registros* dos bancos de dados, nas planilhas existem as *Linhas*, as quais são identificadas por números. Para ilustrar, a figura 2 traz a imagem de uma Pasta de Trabalho do Excel.

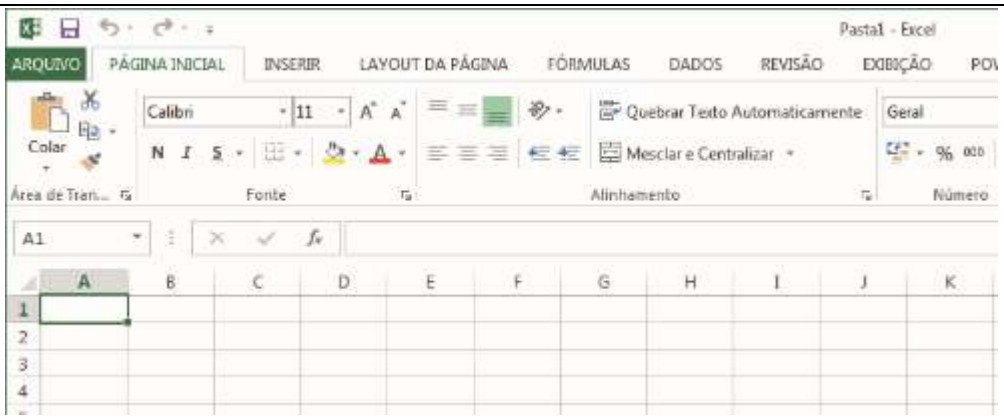


Figura 2. Abrindo o quadro de diálogo que permite acesso às preferências de trabalho do ambiente SAS.

Pode-se observar se tratar da Pasta1, nome padrão já que ainda não foi salvo, momento em que se dá nome à pasta de trabalho. Conforme pode ser identificado na parte inferior da imagem, a Pasta é composta por três planilhas e está sendo exibida a planilha que se chama Plan3, também um nome padrão e que poderá ser alterado pelo operador.

Dessa planilha que está ativa, são visualizadas 4 linhas (numeradas de 1 a 4) e 11 colunas (identificadas por letras de A a K). Cada cruzamento de uma linha com uma coluna recebe o nome de célula e cada célula poderá armazenar um dado.

Uma diferença importante entre as planilhas e o banco de dados é que os bancos de dados requerem a especificação de um *Tipo*, no momento em que o *Campo* é criado enquanto que a coluna de uma planilha pode guardar dados de tipos diferentes. Ao estabelecer o tipo, incorporam-se regras aos campos do banco de dados que fazem com que a palavra “um”, por exemplo, seja rejeitada em um campo numérico.

## 3.4 Dados brutos em arquivo texto

Um terceiro, e último, tipo de arquivo usado no armazenamento de dados brutos é o arquivo texto. O arquivo texto é criado por meio de Editores de Texto tais como o software Bloco de Notas integrante do Windows, a janela Editor do SAS e até por SGBD e Planilhas Eletrônicas.

No caso dos editores de Texto, ao salvar o arquivo já criamos um arquivo texto, ao passo que nas planilhas e SGBD damos ao processo de criação de um arquivo texto que contém os dados de Exportação.

Como são criados em editores de texto, o Programa SAS é considerado como um arquivo texto para os objetivos deste capítulo serão tratados apenas os casos nos quais os dados brutos estão organizados de forma simples dentro do programa.

# 4 O SAS Data Set

O SAS Data Set é um arquivo similar a uma tabela de um banco de dados, mas há uma mudança na terminologia. Aquilo que nos bancos de dados é chamado de “Campo”, no SAS é denominado como “Variável” e os registros são denominados como “Observações”.

Cada variável, ao ser criada recebe um tipo e um nome ou herda esses atributos quando o SAS Data Set é criado a partir da leitura de bancos de dados onde tais informações já estavam definidas. A figura 3 ilustra a estrutura de dados de um SAS Data Set, quando visualizado por meio da janela *Contents Only*.


| | sigma | lambda | condicao |
|---|---|---|---|
| 1 | 1 | 2 | Boa |
| 2 | 1 | 3 | Boa |
| 3 | 1 | 25 | Ruim |
| 4 | 23 | 4 | Boa |
| 5 | 232 | 6 | Ótima |
| 6 | 32 | 245 | Ótima |

Figura 3. Janela ViewTable sendo usada para visualizar os dados armazenados no SAS Data Set que recebeu o nome Exemplo.

No decorrer do capítulo serão estudados os comandos que geraram o SAS Data Set apresentado. Na imagem, preste a atenção para o nome do SAS Data Set (em certos momentos tratado apenas como Tabela): na linha de título é especificado o nome completo do arquivo: Work.Exemplo.

Esse nome é composto por duas partes que são separadas pelo ponto ( . ), a biblioteca na qual ele está armazenado (Work) e o nome que lhe foi dado pelo programador (Exemplo).

A tabela conta com 3 colunas de dados e 6 linhas de dados o que na terminologia do SAS se traduz como 3 variáveis e 6 observações. A primeira variável se chama ***sigma*** e a segunda variável se chama ***lambda*** e a terceira se chama ***condicao*** e esses nomes foram determinados pelo programador na hora em que desenvolvia o programa.

Para que esse SAS Data Set fosse criado, um programa foi desenvolvido, salvo, submetido. Caso algum erro tenha sido cometido, o mesmo foi identificado na janela Log ou na avaliação crítica dos resultados e após corrigidos, considerou-se que a tarefa foi concluída.

Passaremos a estudar os comandos necessários para a criação do *SAS Data Set* anteriormente exibido e depois de criado, utilizaremos a janela *Contents Only* para observá-lo e a algumas de suas propriedades.

# 5 Comandos para criação de um SAS Data Set

Para criar um SAS Data Set com dados armazenados de forma simples, dentro do programa, são necessários três comandos: *data*, *input* e *datalines*.

## 5.1 Comando *data*

O comando data determina a criação de um *SAS Data Set* enquanto que os demais comandos que compõem o *data step* fornecerão instruções adicionais e que permitirão fazer a leitura dos dados brutos, operações com os dados, formatações, ...

O fato do comando *data* iniciar o *Data Step* implica que não há necessidade de nenhum comando anterior a ele, diferente dos comandos *input* e *datalines*, como veremos em seguida.

O uso do comando data segue o seguinte modelo:

*data nome_do_SAS_Data_Set_1*;

O *nome_do_SAS_Data_Set_1* é um texto qualquer especificado pelo programador e esse nome deve seguir três regras básicas:

- Deve começar com letra[1].
- Na sua composição só devem ser usadas letras, numerais e o caractere sublinhado (ou *underscore*) que, na maioria dos teclados fica junto do hífen.
- Deve ter um número máximo de 32 caracteres[2].

Da forma como foi escrito o nome especificado à direita do comando data do modelo, se trata de um nome válido já que o primeiro caractere especificado foi a letra n. Em toda a composição só temos letras, o caractere "_" aparece diversas vezes e o numeral "1". Por fim, o nome conta com apenas 22 caracteres (16 letras, 5 sublinhados e um numeral).

Além da necessidade de conhecimento dessas regras na criação do SAS data set, uma entidade de grande importância na programação SAS, as mesmas regras se aplicarão em diversas situações nas quais são especificados nomes de entidades no SAS, havendo em alguns casos, variações em relação ao número de caracteres.

Recomenda-se, portanto, especial atenção na consideração delas a cada momento que um *SAS Data Set* estiver sendo programado. Para criar o *SAS Data Set* apresentado na figura 3, devemos apenas especificar no comando *data*, o nome do desejado. O comando seria especificado da seguintes forma:

*data exemplo*;

Não há necessidade de especificar que o arquivo exemplo deva ser criada na biblioteca *Work*, pois essa é a biblioteca determinada pelo SAS por default[3].

[1] Letras modificadas por sinais gráficos (ã, õ, é, ç, ...) não são consideradas como letras para o SAS.
[2] Dependendo da versão do SAS podem ser encontrados outros limites.
[3] Informação presumida quando outra não é especificada pelo programador.

## 5.2 Comando *input*

O comando *input* somente pode ser especificado dentro de um Data Step, ou seja, após ter sido especificado um comando *data*.

No comando *input* são especificados os nomes das variáveis que serão criadas, os respectivos tipos e, em situações mais complexas serão especificadas instruções que determinam como deve ser feita a leitura dos dados brutos.

Nesse capítulo não serão especificadas instruções que permitem a adaptação do comando *input* a dados organizados de forma complexa, entretanto, é necessário estabelecer contato com o nome e com os tipos das variáveis.

Além de criar as variáveis, o comando *input* também executa a leitura dos dados brutos, uma observação por vez. Sendo assim a ordem que os nomes são especificados devem estar de acordo com a ordem em que os dados brutos são apresentados.

O uso do comando input segue o seguinte modelo:

*input nome_da_variavel_1 tipo_da_variavel_1*;

À exemplo do comando data, o *nome_da_variavel_1* é um texto estabelecido pelo programador e que deve seguir as mesmas regras especificadas para o nome do SAS Data Set começar com letra, somente usar letras números e sublinhados limitados a 32 caracteres).

Já o tipo da variável, depende da natureza do dado que será armazenado e o SAS somente conta com dois tipos de variáveis: variáveis que armazenam números e não-números. Quando o comando *input* lerá valores numéricos para armazená-los em uma variável, não há necessidade da especificação de um tipo (default: leitura de dados numéricos). Agora, se o valor que será armazenado na variável não é um número (palavras, sequência de letras, códigos, ...) então a variável deve ser sinalizada com o sinal de $ (cifrão).

Caso nosso arquivo de exemplo tivesse exclusivamente a primeira variável (***sigma***) e considerando que a mesma é numérica[4] (figura 3), então o comando input ficaria da seguinte forma:

*input sigma*;

Como em nosso exemplo temos 3 variáveis (***sigma***, ***lambda*** e ***condicao***) o comando input deve ser modificado para que também essas outras variáveis sejam criadas e seus valores lidos e armazenados. E para executar essa modificação, podemos incluir quantas variáveis forem necessárias à direita do comando input e a direita de cada variável, quando necessário o símbolo que identifica se tratar de uma variável alfanumérica, conforme representado em seguida:

*input sigma lambda condicao $*;

Note que o nome de cada variável segue as regras, inclusive a de não ser permitido o uso de sinais modificados graficamente com cedilha e til e que apenas a variável condição recebeu o sinal de cifrão indicando se tratar de uma variável cujos valores não são numéricos.

[4] Na figura 3, os dados alinhados à direita na célula são numéricos ao passo que os dados não numéricos (tecnicamente chamados de alfanuméricos) são alinhados à esquerda na célula.

## 5.3 Comando *datalines*[5]

O comando *datalines* também só pode ser usado no Data Step e em geral ele é o último comando desta parte do programa.

Esse comando sempre usado da seguinte forma:

*datalines*;

;

Note que há um ponto e vírgula especificado em linha inferior à do comando *datalines*. É nesse espaço entre o comando e o ponto e vírgula que serão listados os dados brutos. Cada observação (um valor para cada variável deverá ocupar uma linha) abaixo do *datalines* dentro da observação os valores das diferentes variáveis devem ser separados por, pelo menos, um espaço em branco.

Os dados do exemplo são inseridos entre o comando e o ponte e vírgula final:

```
datalines;
1 2 Boa
1 3 Boa
1 25 Ruim
23 4 Boa
232 6 Ótima
32 245 Ótima
;
```

É importante salientar que após a lista de dados deve ser colocado um ponto e vírgula que informa ao comando input o fim da leitura dos dados. Esse ponto e vírgula deve ser colocado na primeira coluna da linha seguinte à última observação.

## 5.4 O programa completo

Os três comandos foram tratados separadamente, entretanto, isoladamente cada um deles é insuficiente para efetivar a criação do SAS Data Set, por isso, os três são incorporados e representados no programa 1.

Programa 1. Programa para exercitar a operação do ambiente SAS e a exibição de um gráfico na janela *Graph*.

```
data exemplo;
   input sigma lambda condicao $;
datalines;
1 2 Boa
1 3 Boa
1 25 Ruim
23 4 Boa
232 6 Ótima
32 245 Ótima
;
```

O comando *data* cria a tabela ao passo que o comando *input* cria as variáveis e faz a leitura dos dados brutos especificados abaixo do comando *datalines*.

[5] Programas antigos usam o comando *cards* e não há nenhuma diferença entre eles.

Note que deve existir uma compatibilidade estrita entre as informações do comando *input* com a forma de digitação dos dados no comando *datalines*. O primeiro dado especificado no comando *datalines* (1) será armazenado na primeira variável especificada no comando input (***sigma***). Como um espaço em branco foi encontrado após o número 1, o input vai alocar o segundo dado (2) na segunda variável (***lambda***). Feito isso, o terceiro dado será lido e o input já espera que se trate de um valor não numérico pois a terceira variável (***condicao***) tinha um cifrão à sua direita, por isso ele faz a leitura da palavra "Boa" que é armazenada na variável.

Tendo sido constituída a primeira observação, um valor para cada variável, ela é armazenada no *SAS Data Set* e o processo de leitura e armazenamento se repete nas linhas subsequentes até que o sinal de ponto e vírgula ( ; ) na primeira coluna da linha seguinte aos dados encerra a leitura.

Vamos digitar, salvar, submeter e avaliar a janela *Log* do programa 1. O resultado desse processamento é ilustrado na figura 4.

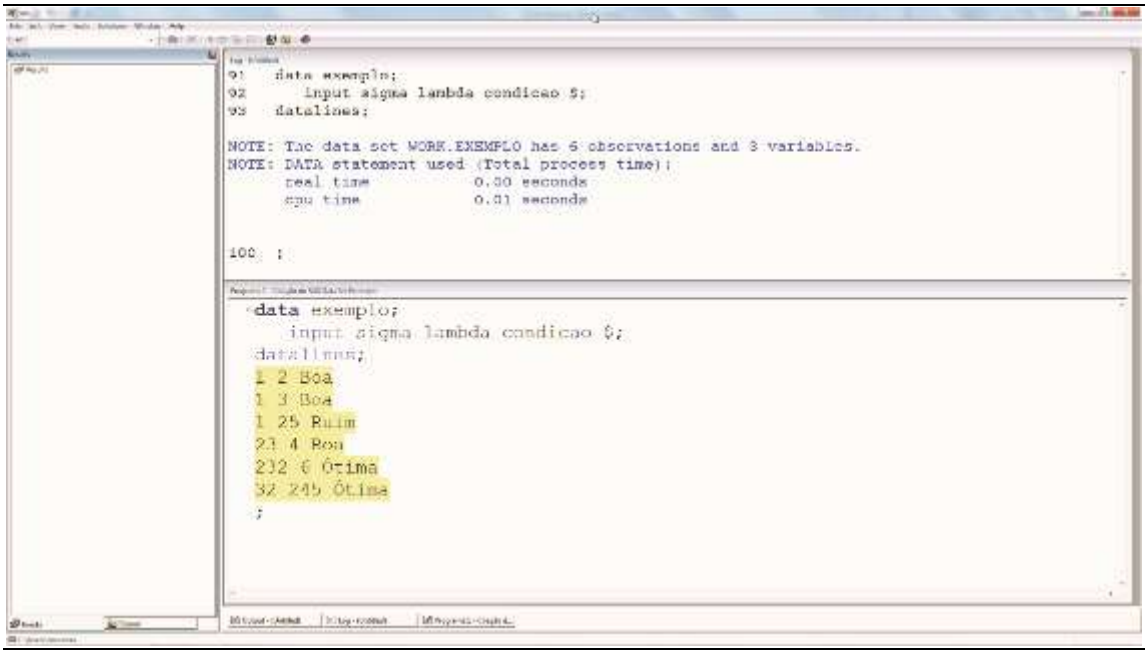


Figura 4. Janela *Editor* com Programa 1 – Criação do *SAS Data Set* Exemplo, a janela *Log* resultante do processamento e a janela *Results*.

Ao avaliarmos a janela *Log*, vemos em primeiro lugar que não foram apontados erros ou mensagens de alerta. Observamos as linhas de programa submetidas e à esquerda de cada uma delas, um número de linha que depende de quantos comandos já foram executados nesta seção[6] do SAS.

Se fosse o primeiro programa executado na seção, ao invés do número 81 à esquerda do comando *data*, teríamos o número 1.

Por fim, o mais importante: a nota abaixo destacada:

NOTE: The data set WORK.EXEMPLO has 6 observations and 3 variables.

[6] Tempo decorrido entre a ativação do SAS e seu fechamento.

A nota nos informa que o *SAS Data Set* foi criado e que ele conta com 3 variáveis e 6 observações. E esse era o objetivo do programa, criar o *SAS Data Set*.

Note que nenhum resultado é exibido já que o *Data Step* não gera relatórios, o que exigiria algum *Proc Step*, conforme visto no capítulo *A anatomia de um programa SAS*.

Para ter certeza de que o arquivo foi corretamente criado, todavia, há necessidade de analisarmos o seu conteúdo o que pode ser feito por meio da programação, incluindo os seguintes comandos no programa, por exemplo:

```
proc print;
run;
```

Ou então podemos usar a janela *Contents Only* para observar os dados armazenados no *SAS Data Set*.

# 6 Visualizando os dados por meio da janela Contents Only

Em primeiro lugar, precisamos ativar a janela Contents Only e para isso usamos a guia rotulada como Explorer no painel de navegação (lado esquerdo da tela na figura 4) e obtemos a imagem representada na figura 5.

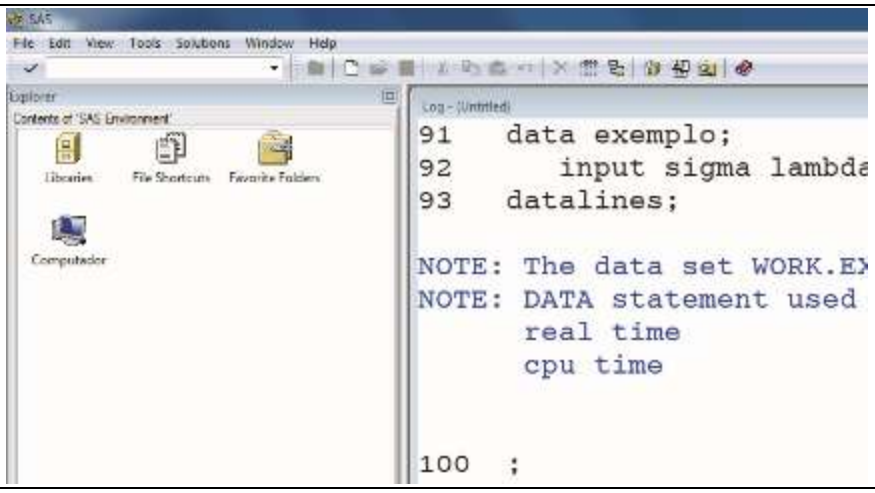


Figura 5. Janela *Contents Only* no início de sua operação.

Feita esta configuração, já receberemos os relatórios também em HTML e para isso, a cada vez que um programa que gere relatórios for submetido, uma nova janela (Internal Browser) será exibida com o resultado no formato HTML.

Os SAS Data Sets são armazenados em bibliotecas, por isso, para observarmos o arquivo, devemos clicar no ícone *Libraries*, no canto superior direito da janela. Ao darmos um clique duplo (botão esquerdo do mouse) podemos visualizar as bibliotecas disponíveis nessa instalação do SAS, conforme ilustra a figura 6.

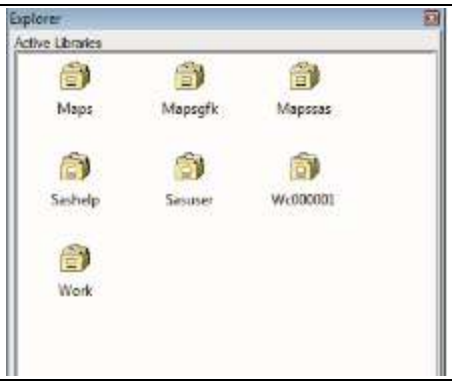


Figura 6. Bibliotecas ativas (Active Libraries) mostradas na janela *Contents Only*.

Podemos observar a presença de 7 bibliotecas ativas (*Active Libraries*) dentre as quais observamos uma chamada *Work*. Quando não se especifica no programa qual biblioteca deve ser usada para armazenar o *SAS Data Set*, o mesmo é armazenado na biblioteca *Work*. Por isso, o SAS Data Set que foi criado anteriormente está dentro dessa janela, conforme ilustra a figura 7.

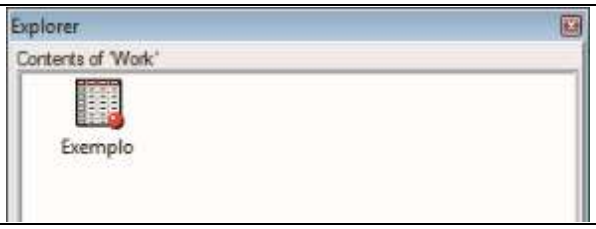


Figura 7. Conteúdo da biblioteca Work.

Se clicarmos com o botão direito do mouse, surge um menu Pop Up que exibe as operações que podemos executar com o SAS Data Set exemplo, conforme ilustra a figura 8.

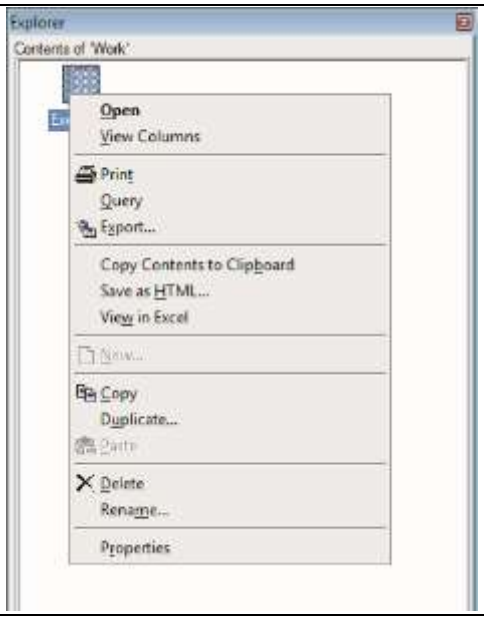


Figura 8. Opções de operação do SAS Data Set Exemplo, por meio da janela *Contents Only*.

Note que podemos abrir a tabela para visualizá-la (***Open***), observar informações sobre as colunas (*View Columns*), imprimir a tabela (*Print*), selecionar parte dos dados (*Query*), exportas os dados para outros aplicativos (*Export*), copiar o conteúdo para a área de transferência, salvar os dados no formato HTML, abrir a tabela SAS no software Microsoft Excel,

copiar, criar outra tabela de nome diferente mais com o mesmo conteúdo (*Duplicate*), excluir a tabela (*Delete*), renomear a tabela e observar propriedades da tabela (*Properties*).

Em primeiro lugar, vamos observar a tabela, para isso, clicamos na opção Open. O efeito seria o mesmo se tivéssemos simplesmente dado um clique duplo sobre a tabela. O resultado é apresentado na figura 3 já que o conteúdo da tabela já havia sido apresentado antes da execução do programa.

Após avaliar que tudo está conforme o esperado, devemos fechar a janela e vamos avaliar as colunas, desta vez clicando com o botão direito do mouse e selecionando a opção *View Colums*, conforme ilustrado na figura 9.

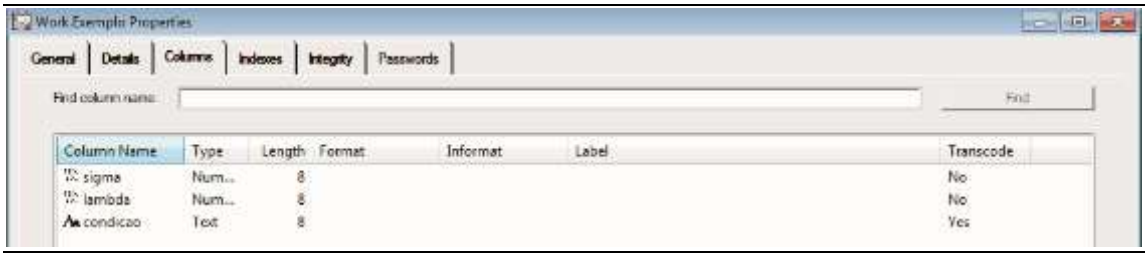


Figura 9. Informações sobre as colunas (variáveis) da tabela Exemplo.

Observe os nomes e tipos das variáveis e note, também que o SAS começa a tratar os termos ***variável*** e ***coluna*** como sinônimos.

# 7 Exercícios

Desenvolva programas que criem SAS Data Sets e avalie o sucesso da programação por meio das janelas *Log* e *Contents Only*.

## 7.1 Nomes e idades de 5 pessoas.

Tabela 1. Nomes e idades de 4 pessoas.

| Nome | Idade |
|---|---|
| João | 31 |
| Maria | 45 |
| Joana | 32 |
| Aderbal | 12 |

## 7.2 Pesos e estaturas de 3 crianças

Tabela 2. Pesos e estaturas de 3 crianças.

| Nome | Peso (kg) | Estatura (m) |
|---|---|---|
| Ana | 30.30[7] | 1.20 |
| Cecília | 31.20 | 1.30 |
| Hommer | 36.80 | 1.28 |

[7] O SAS somente aceita ponto como indicador de casa decimal. Não use vírgulas.