

EDUARDO PAGEL FLORIANO

**SAS PARA LEIGOS
E
ENGENHEIROS FLORESTAIS**

São Gabriel
Edição do autor
2017

EDUARDO PAGEL FLORIANO

**SAS PARA LEIGOS
E
ENGENHEIROS FLORESTAIS**

1ª edição

**São Gabriel
Edição do autor
2017**

SAS para leigos e Engenheiros Florestais
©Eduardo Pagel Floriano¹
São Gabriel, 27 de julho de 2017.

F635s Floriano, Eduardo Pagel

SAS para leigos e Engenheiros Florestais/ Eduardo Floriano Pagel. São Gabriel: edição do autor, 2017.

256 p.

ISBN: 978-85-918170-2-3

1. SAS University Edition 2. Estatística 3. Experimentação
4. Análise de Regressão I. Título

CDU519.22: 630*2

APRESENTAÇÃO

Os manuais originais do SAS são escritos em língua Inglesa e somam milhares de páginas, dificultando o seu uso. O objetivo deste documento é o de servir de guia, em língua portuguesa, para as principais aplicações da linguagem SAS. Não se pretende substituir os manuais originais, mas facilitar o acesso ao que se considera o mais importante da linguagem.

Os capítulos são organizados em sequência apropriada para aprendizagem passo a passo, sem mestre. O primeiro capítulo contém um breve histórico e informações sobre o sistema SAS. O segundo capítulo dedica-se à descrição de forma abreviada do software livre SAS University Edition e sua interface SAS Studio. Os três capítulos seguintes (3, 4 e 5) explicam a estrutura da linguagem SAS tradicional. Para aprender a programar, portanto, é aconselhável ler e realizar os exercícios desta parte inicial integralmente. O Capítulo 6 refere-se à formatação de dados de entrada e de saída, o 7 aos explorers do SAS, o 8 aos formatos de saída e o Capítulo 9 diz respeito às funções pré-programadas da linguagem; Dos capítulos 6 e 9, recomenda-se a leitura da introdução e o estudo das secções sobre sintaxe, o restante pode ser utilizado somente para consulta, durante a elaboração de um programa, não sendo de leitura obrigatória. Do Capítulo 10 ao 20, são descritos resumidamente os principais procedimentos da linguagem, intitulados de "PROC". No capítulo 21 é realizada uma introdução ao procedimento IML para trabalhar com matrizes. Os capítulos 22, 23 e 24 são dedicados a exemplos de aplicações na Engenharia Florestal. E, o capítulo 25 é dedicado aos gráficos ODS (Output Delivery System), disponíveis para o SAS versão 9.2 e para o SAS University Edition.

A intenção com esta obra é a de facilitar o aprendizado do SAS e auxiliar na migração do SAS tradicional para o SAS University Edition e sua interface SAS Studio no meio acadêmico.

São Gabriel, 27 de julho de 2017.

Eduardo Pagel Floriano

Conteúdo

Capítulo 1 : SAS – O que é?	1
Capítulo 2 : SAS University Edition	3
2.1 INTRODUÇÃO AO SAS UNIVERSITY EDITION.....	3
2.2 REQUISITOS DE SISTEMA	4
2.3 OBTENÇÃO DO SAS UNIVERSITY EDITION	5
2.4 INTERFACE SAS STUDIO	15
2.4.1 <i>Server Files and Folders</i> (servidor de arquivos e pastas).....	16
2.4.2 <i>Folders Shortcuts</i> (atalhos de pastas).....	17
2.4.3 <i>My Folders</i> (minhas pastas)	17
2.4.4 <i>Tasks and Utilities</i> (tarefas e utilitários).....	19
2.4.5 <i>Snippets</i> (fragmentos)	25
2.4.6 <i>Libraries</i> (bibliotecas)	27
2.4.7 <i>File Shortcuts</i> (atalhos de arquivos).....	27
Capítulo 3 : Introdução à linguagem SAS	29
3.1 CONVENÇÕES UTILIZADAS	29
3.2 PROGRAMAS SAS: EDIÇÃO, EXECUÇÃO E COMPONENTES.....	32
3.3 PROCEDIMENTOS SAS (PROC STEP).....	38
3.4 USANDO OS EXEMPLOS DE PROGRAMAS.....	40
Capítulo 4 : Passos de programação SAS	41
4.1 DATA STEP.....	41
4.2 PROC STEP	42
4.3 JCL (<i>JOB CONTROL LANGUAGE</i>)	43
Capítulo 5 : Elementos da linguagem SAS	45
5.1 PALAVRAS	46
5.1.1 <i>Nomes</i>	46
5.1.2 <i>Literais</i>	47
5.1.3 <i>Números</i>	47
5.1.4 <i>Caracteres especiais</i>	47
5.2 CONSTANTES	48
5.2.1 <i>Constantes numéricas</i>	48
5.2.2 <i>Constantes de caracteres</i>	48
5.2.3 <i>Constantes de data, horário e data-horário</i>	48
5.3 VARIÁVEIS	49
5.4 OPERANDOS.....	49
5.5 OPERADORES.....	49
5.6 EXPRESSÕES	51
5.7 INSTRUÇÕES.....	52

5.8 DECLARAÇÕES	53
5.9 FUNÇÕES	53
Capítulo 6 : Arquivos de dados (DATA STEP)	54
6.1 SINTAXE	55
6.1.1 Criação de arquivos de dados do tipo DATA SET	55
6.1.2 Outros modelos de sintaxe para o DATA STEP	55
6.2 FORMATOS DE ENTRADA E SAÍDA (INPUT/PUT)	57
6.3 ENTRADA DE DADOS	60
6.3.1 Importando do Excel para o SAS	60
6.3.2 USANDO DADOS DE UMA PLANILHA SAS	60
6.3.3 2.3.3Incluindo os dados no programa	60
6.3.4 Criando um arquivo a partir de dados de um arquivo externo:	61
6.3.5 Importando uma planilha do excel no programa	61
6.3.6 Exportando uma planilha para o excel	61
6.3.7 Exemplo de importação e exportação de dados:	61
6.4 PROCEDIMENTOS ESSENCIAIS COM O DATA STEP	61
6.4.1 PROC APPEND	61
6.4.2 PROC SORT	63
6.5 INSTRUÇÕES ESSENCIAIS DO DATA STEP	64
6.5.1 ARRAY (instrução)	65
6.5.2 BY (instrução)	66
6.5.3 CARDS (instrução)	67
6.5.4 DATALINES (instrução)	67
6.5.5 DELETE (instrução)	67
6.5.6 DO/END (instrução)	67
6.5.7 DROP (instrução)	68
6.5.8 FILE (instrução)	69
6.5.9 FILENAME (instrução)	70
6.5.10 IF THEN (instrução)	70
6.5.11 INFILE (instrução)	71
6.5.12 INPUT (instrução)	71
6.5.13 KEEP (instrução)	73
6.5.14 MERGE (instrução)	73
6.5.15 OUTPUT (instrução)	76
6.5.16 PUT (instrução)	78
6.5.17 RETAIN (instrução)	79
6.5.18 SET (instrução)	81
6.5.19 SUM (instrução)	81
6.5.20 UPDATE (instrução)	82
6.5.21 WHERE (instrução)	82
Capítulo 7 : Explorer do SAS.....	84
7.1 PASTA WORK	84
7.2 CRIANDO PASTAS DE ARQUIVOS	84
Capítulo 8 : Formatos de saída (FORMAT) e entrada (INFORMAT)	88
8.1 FORMAT (INSTRUÇÃO DE FORMATOS DE SAÍDA)	88
8.1.1 Formatos de saída definidos pelo usuário	89
8.1.2 Formatos de saída para variáveis de caracteres	90
8.1.3 Formatos de saída para variáveis numéricas	90
8.1.4 Formatos de saída para variáveis de data_e_horário	91

8.2 INFORMAT (INSTRUÇÃO DE FORMATOS DE ENTRADA)	91
8.2.1 Formatos de entrada para variáveis de caracteres	92
8.2.2 Formatos de entrada para variáveis numéricas	93
8.2.3 Formatos de entrada para data, horário e data-horário	93
8.3 PROC FORMAT	94
8.3.1 VALUE (instrução)	95
8.3.2 INVALUE (instrução)	95
8.3.3 SELECT (instrução)	95
8.3.4 PICTURE (instrução)	96
8.3.5 EXCLUDE (instrução)	96
8.4 FORMATANDO DADOS	97
8.4.1 Comando ROUND	97
8.4.2 Comandos TITLE, Length, Format E NOOBS	97
8.5 MAIS EXEMPLOS	97
Capítulo 9 : Funções	105
9.1 RESULTADOS DE FUNÇÕES	106
9.2 CATEGORIAS DE FUNÇÕES	107
9.2.1 Funções aritméticas	107
9.2.2 Funções de caractere	108
9.2.3 Funções de data e horário	109
9.2.4 Funções financeiras	109
9.2.5 Funções matemáticas	113
9.2.6 Funções de números aleatórios	113
9.2.7 Funções de estatística básica	113
9.2.8 Funções trigonométricas	114
9.2.9 Funções de truncagem	114
9.2.10 Funções de hospedagem	114
Capítulo 10 : PROC ANOVA / PROC GLM	116
10.1 SINTAXE:	116
10.2 OPÇÕES DO PROCEDIMENTO:	117
10.3 INSTRUÇÕES DO PROCEDIMENTO:	117
Capítulo 11 : PROC CHART / PROC GCHART	122
Capítulo 12 : PROC CORR	126
12.1 SINTAXE:	126
12.2 OPÇÕES DO PROCEDIMENTO:	126
12.3 INSTRUÇÕES DO PROCEDIMENTO:	127
12.3.1 BY (instrução)	127
12.3.2 FREQ (instrução)	127
12.3.3 PARTIAL (instrução)	128
12.3.4 VAR (instrução)	128
12.3.5 WEIGHT (instrução)	128
12.3.6 WITH (instrução)	128
12.3.7 Valores perdidos (missing values)	129
12.3.8 Arquivos de saída TYPE=CORR	129
Capítulo 13 : PROC FREQ	132
13.1 SINTAXE	132
13.1.1 BY (instrução)	132

13.1.2 EXACT (instrução).....	133
13.1.3 OUTPUT (instrução).....	133
13.1.4 TABLES (instrução).....	133
13.1.5 TEST (instrução).....	134
13.1.6 WEIGHT (instrução).....	134
Capítulo 14 : PROC MEANS / PROC SUMMARY.....	141
14.1 SINTAXE.....	141
14.2 OPÇÕES DO PROCEDIMENTO.....	141
14.3 ALGUMAS ESTATÍSTICAS.....	141
Capítulo 15 : PROC MODEL.....	144
15.1 INTRODUÇÃO.....	144
15.2 VALIDAÇÃO DE MODELOS.....	146
15.2.1 Homocedasticidade da variância.....	146
15.2.2 Independência dos resíduos.....	147
15.2.3 Normalidade da distribuição dos resíduos.....	147
15.3 VALORES PERDIDOS NO PROC MODEL.....	150
Capítulo 16 : PROC PLOT / PROC GPLOT.....	155
16.1 SINTAXE.....	155
16.2 OPÇÕES DO PROCEDIMENTO PROC GPLOT.....	155
16.3 INSTRUÇÕES DO PROCEDIMENTO.....	155
16.3.1 Características dos gráficos gerados por PLOT.....	156
Capítulo 17 : PROC PRINT.....	164
17.1 SINTAXE.....	164
17.1.1 Opções do procedimento:.....	164
17.1.2 BY (instrução).....	165
17.1.3 PAGEBY (instrução).....	165
17.1.4 SUMBY (instrução).....	166
17.1.5 ID (instrução).....	166
17.1.6 SUM (instrução).....	166
17.1.7 VAR (instrução).....	166
Capítulo 18 : PROC REG.....	172
18.1 SINTAXE.....	172
18.1.1 Opções do Procedimento.....	172
18.1.2 Instruções do procedimento.....	173
Capítulo 19 : PROC NLIN.....	179
19.1 SINTAXE.....	180
19.1.1 Opções do procedimento.....	180
19.1.2 Instruções do procedimento.....	180
Capítulo 20 : PROC UNIVARIATE.....	187
20.1 SINTAXE.....	187
20.1.1 Opções do procedimento.....	187
20.1.2 Instruções do procedimento.....	188
Capítulo 21 : PROC IML (operações com matrizes).....	195
21.1 SINTAXE.....	195
21.1.1 Declarações, Módulos e Subrotinas.....	195

21.1.2 Comandos e Operadores	196
Capítulo 22 : Aplicações em Dendrometria.....	200
22.1 CÁLCULO DO VOLUME DE ÁRVORES CUBADAS	200
22.2 FATOR DE CASCA E SELEÇÃO DE VARIÁVEIS DE REGRESSÃO	204
Capítulo 23 : Aplicações em inventários florestais.....	211
23.1 PROC SURVEYMEANS	212
23.1.1 Instruções do procedimento	213
23.2 AMOSTRAGEM ESTRATIFICADA.....	217
23.3 AMOSTRAGEM ESTRATIFICADA PONDERADA	219
23.4 AMOSTRAGEM EM CONGLOMERADOS	222
Capítulo 24 : Aplicações em experimentação florestal.....	225
24.1 MODELOS MATEMÁTICOS.....	225
24.2 DELINEAMENTO INTEIRAMENTE CASUALIZADO	226
24.3 DELINEAMENTO EM BLOCOS CASUALIZADOS	228
24.4 DELINEAMENTO EM QUADRADO LATINO.....	230
Capítulo 25 : Gráficos no SAS University Edition (ODS Graphics).....	233
25.1 INTRODUÇÃO	233
25.1.1 Declarações de destino ODS	234
25.1.2 PLOTS = Opção.	235
25.1.3 Procedimentos da linguagem que suportam gráficos ODS	236
25.2 PROCEDIMENTOS PARA CONSTRUIR GRÁFICOS ODS.....	238
25.2.1 PROC SGPLOT.....	238
25.2.2 PROC SGSCATTER	240
25.2.3 PROC SGPANEL	240
Referências	242
Apêndice 1 – Planilha para coleta de dados de cubagem de árvores abatidas	243

Capítulo 1 : SAS – O que é?

O SAS foi desenvolvido na Universidade Estadual da Carolina do Norte de 1966 a 1976. Em 1976, foi fundado o SAS Institute por Anthony Barr, James Goodnight, John Sall e Jane Helwig. Inicialmente foi criado um software denominado de Statistical Analysis System com aplicações estatísticas dirigidas à análise de experimentos rurais, mas, com o passar do tempo, o Instituto SAS produziu um sistema muito complexo chamado de *SAS System*.

Em 2001, a empresa adotou a denominação atual para si e para o software: SAS. O software foi transformado num sistema integrado com uma infinidade de aplicações que permitem a análise e gerenciamento de dados e a administração de praticamente qualquer projeto ou organização.

O SAS é considerado o mais completo pacote estatístico produzido até hoje, entretanto, vai muito além disso, podendo ser visto como um software completo com aplicação em inteligência de negócios que oferece os seguintes módulos de serviços:

- Base SAS - Procedimentos básicos e gerenciamento de dados;
- SAS / STAT - Análise estatística;
- SAS / GRAPH - Gráficos e apresentação;
- SAS / OR - Pesquisa operacional;
- SAS / ETS - Econometria e análise de séries temporais;
- SAS / IML - Linguagem de matricial – cálculo com matrizes;
- SAS / AF – Módulo interativo;
- SAS / QC - Controle de qualidade;
- SAS / INSIGHT - Data mining;
- SAS / PH - Análise de ensaios clínicos;
- Enterprise Miner - Data mining;
- Enterprise Guide - Editor de código baseado em GUI e gerenciador de projetos;
- SAS EBI - Suite de aplicativos de Business Intelligence;

- SAS Grid Manager - Gerente do ambiente de computação grid SAS.

Com esta profusão de funções, os manuais do SAS acabaram por se tornar muito extensos e algo confusos, dificultando o seu uso.

Entretanto, o poder da linguagem SAS, para aplicações estatísticas em áreas como a biológica e a florestal pode ser alcançado com um mínimo de conhecimento sobre o sistema. É isso o que se pretende com este manual: descrever o essencial do SAS para aplicações estatísticas, de forma a possibilitar o uso do poder do sistema ao usuário leigo em informática, com ênfase na área da Engenharia Florestal.

Na atualidade, existe uma versão de uso livre denominada SAS University Edition que pode ser usada nos meios universitários, com todos os procedimentos e funções dos módulos básico (BASE), estatísticos (STAT e ETS) e de operações com matrizes (IML).

Capítulo 2 : SAS University Edition

O SAS University Edition é um software livre para uso não comercial, desenvolvido principalmente para uso acadêmico por professores, estudantes e pesquisadores.

2.1 INTRODUÇÃO AO SAS UNIVERSITY EDITION

A empresa produtora do software é a SAS Inc. informa o que o SAS University Edition tem a oferecer (SAS, 2017):

- “Software SAS gratuito para usar em aulas de métodos estatísticos e quantitativos em uma variedade de áreas: economia, psicologia e outras ciências sociais, ciência da computação, administração, ciências médicas e da saúde, engenharia e etc.
- Materiais gratuitos para ensino e desenvolvimento do currículo - tudo o que você precisa para dar um curso utilizando o SAS.
- Software estatístico que funciona da mesma forma em PCs, Macs ou Linux, para que as barreiras tradicionais para aos alunos, tais como custo e sistema operacional, sejam coisa do passado.
- Uma comunidade online para os usuários acadêmicos do SAS, onde você poderá interagir com outras pessoas, obter suporte, acessar os recursos do SAS e etc.
- Software e recursos para o desenvolvimento do conhecimento analítico dos alunos a fim de estimular suas habilidades analíticas e torná-los melhores solucionadores de problemas e tomadores de decisão.”

O SAS University Edition tem uma interface que permite interagir com o software de um PC, Mac ou Linux, chamada SAS Studio e é composto pelos seguintes módulos:



:

- SAS Studio - É um aplicativo web de desenvolvimento para o SAS acessível por um navegador web. Com o SAS Studio, é possível acessar arquivos de dados, bibliotecas, programas existentes e escrever novos programas. Também é possível usar as tarefas predefinidas no SAS Studio para gerar novos códigos SAS. Ao executar um programa

ou tarefa, o SAS Studio processa o código SAS em um servidor SAS. O servidor SAS pode ser um servidor em um ambiente em nuvem, um servidor em seu ambiente local ou o SAS instalado numa máquina local. Após o processamento do código, os resultados são retornados ao SAS Studio no navegador.

- BASE/SAS - linguagem de programação de quarta geração projetada como o padrão de acesso, transformação e relatórios de dados. Fornece um ambiente de software escalável e integrado especialmente concebido para acesso, transformação e relatórios de dados.
- SAS/STAT - Software de análise estatística de última geração que permite da análise tradicional de variância e regressão linear à inferência bayesiana e ferramentas de modelagem de alto desempenho para dados maciços, o SAS/STAT atende às necessidades estatísticas especializadas de qualquer empreendimento.
- SAS/IML - Software, que fornece uma linguagem de programação flexível, permitindo realizar análises de dados estatísticos, simulação, cálculos de matriz e otimização não linear; oferece uma linguagem de programação interativa com uma extensa biblioteca de sub-rotinas, permitindo criar seus próprios módulos de função personalizados; inclui a capacidade de enviar instruções SAS e chamar funções na linguagem R a partir do procedimento IML.
- SAS/ACCESS – Este módulo fornece acesso instantâneo para quase todos os formatos de arquivos de PC, possibilitando uma abordagem simplificada no acesso de dados.

2.2 REQUISITOS DE SISTEMA

A instalação do SAS University Edition tem como requisitos, por sistema operacional:

- **Windows:**

Microsoft Windows versão 7, 8, 8.1 or 10; hardware 64-bit com um mínimo de 1GB de RAM Um dos seguintes softwares de virtualização: VMware Player 7 ou posterior, Oracle VirtualBox 4.3.16 ou posterior. Um dos seguintes navegadores: Microsoft Internet Explorer 9, 10 or 11; Mozilla Firefox 21 ou posterior; Google Chrome 27 ou posterior (sugere-se o uso do VirtualBox com o Firefox, pois não têm apresentado problemas com o SAS Studio).

- **OS X:**

Mac OS X 10.8 ou posterior; hardware de 64-bit com um mínimo de 1GB de RAM. Um dos seguintes softwares de virtualização: VMware Fusion para OS X 7 ou posterior, Oracle VirtualBox 4.3.16 ou posterior. Um dos seguintes navegadores: Apple Safari 6.0 ou posterior; Mozilla Firefox 21 ou posterior; Google Chrome 27 ou posterior.

- **Linux:**

Ambiente operacional Linux x86 - hardware de 64-bit com um mínimo de 1GB de RAM. Um dos seguintes softwares de virtualização: VMware Player 7 para Linux ou posterior; Oracle VirtualBox 4.3.16 ou posterior. Um dos seguintes navegadores: Mozilla Firefox 21 ou posterior; Google Chrome 27 ou posterior.

2.3 OBTENÇÃO DO SAS UNIVERSITY EDITION

O SAS University Edition é uma aplicação virtual (vApp) e está disponível para download no site: <https://www.sas.com/pt_br/software/university-edition.html>. É necessário fazer inscrição no site do SAS para poder baixá-lo com um nome de usuário e senha previamente registrados. Você será instruído a fazer a inscrição quando for necessário. Para ser instalado e executado, antes é preciso instalar um programa de virtualização em seu computador.

Depois de baixá-lo para seu PC, Mac ou estação de trabalho Linux, e instalá-lo no ambiente virtual, o SAS funciona diretamente no computador, usando o software de virtualização e o navegador, não sendo necessário acesso à Internet após a instalação.

Para obter o SAS e o instalar, siga as etapas a seguir:

1. Baixe e instale um software de virtualização – os links estão na página de download do SAS. Verifique qual seu sistema operacional e baixe um sistema virtual compatível. Para o Windows, pode ser o software livre da Oracle: **Oracle VM VirtualBox platform packages**. O programa está disponível em: <<https://www.virtualbox.org/wiki/Downloads>>. O link para download direto é: <<http://download.virtualbox.org/virtualbox/5.1.22/VirtualBox-5.1.22-115126-Win.exe>>. Após baixar, basta executar o arquivo VirtualBox-5.1.22-115126-Win.exe para instalá-lo.
2. Baixe o Guia de Instalação do Software de Virtualização e leia-o com atenção. Todas as instruções para instalação do SAS University Edition estão descritas passo a passo neste manual. O manual do pacote VirtualBox da Oracle está disponível em: <<http://support.sas.com/software/products/university-edition/docs/en/SASUniversityEditionQuickStartVirtualBox.pdf>>
3. Baixe o SAS University Edition - Escolha o arquivo de download apropriado ao seu software de virtualização. No caso do VirtualBox, o link para download do SAS é <http://www.sas.com/store/expresscheckout.ep?item=DPUNVE001_VirtualBox>. Ao acessar a página de download haverá duas opções, uma para novos usuários, que deverão criar um perfil clicando em “Criar Perfil”, ou para usuários já inscritos que deverão fornecer seu e-mail e senha. Inicie o download e aguarde, vá tomar um café, pois o arquivo tem mais de 2 Gigabytes. O arquivo tem “.ova” como sufixo. O arquivo do SAS University Edition vApp para Windows com o VirtualBox tem o nome de: “unvbasicvapp__9411007__ova__en__sp0__1.ova”.
4. Abra o ambiente virtual Oracle VM VirtualBox Manager e Adicione o SAS University Edition vApp ao mesmo. Clique em: | Arquivo | | Importar | | Appliance... | (Figura 1).

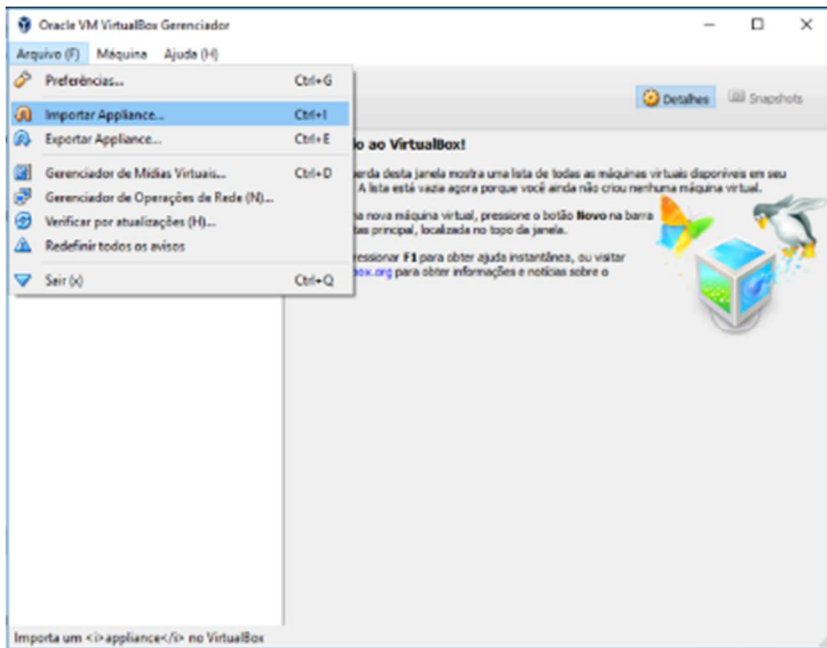


Figura 1: Janela do Oracle VM VirtualBox

Na janela que se abre a seguir (Figura 2), clique no símbolo de pasta à direita do espaço para digitação e navegue pelos menus para encontrar o arquivo do SAS onde ele foi gravado quando o download foi realizado.

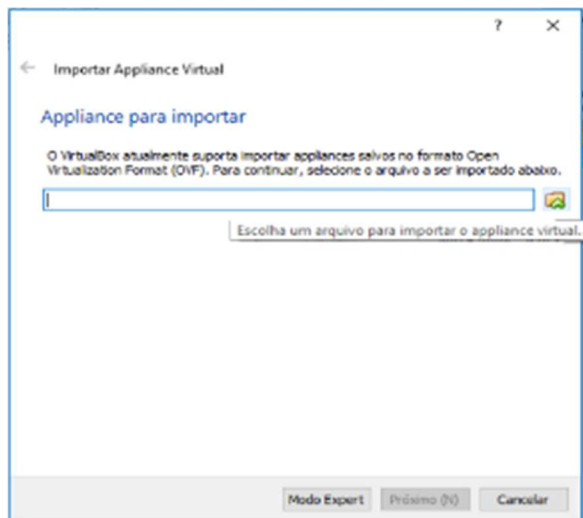


Figura 2: Janela de importação de aplicativo do VirtualBox

Ao encontrar a pasta (Figura 3), clique no arquivo do SAS denominado de “unvbasicvapp_9411007_ova_en_sp0_1.ova” e depois clique em | Abrir |.

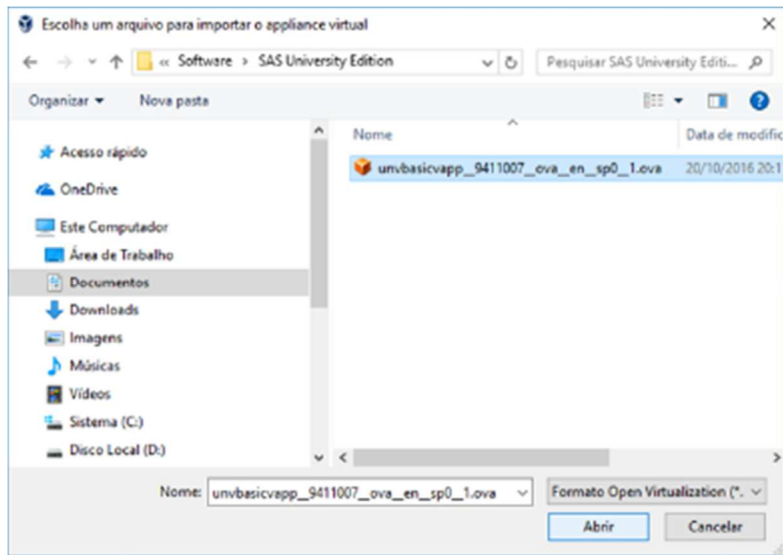


Figura 3: **Escolhendo o arquivo *.ova para importar**

Ao fazer isso, a última tela é fechada e retorna para a anterior (Figura 4). Então, clique em | Próximo |.



Figura 4: **Continuando a importação SAS vApp**

Na próxima janela (Figura 5), clique em | Importar |.

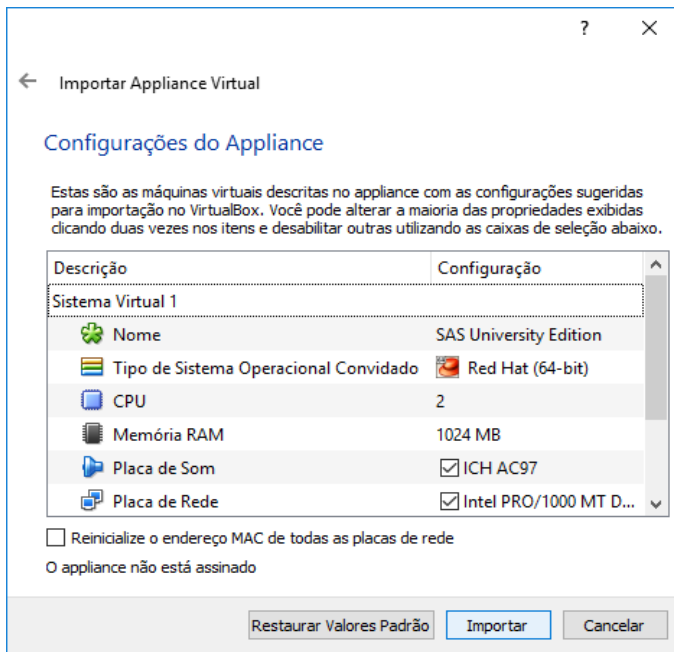


Figura 5: **Finalizando a importação do SAS University Edition vApp**

Aguarde que a importação termine (Figura 6).

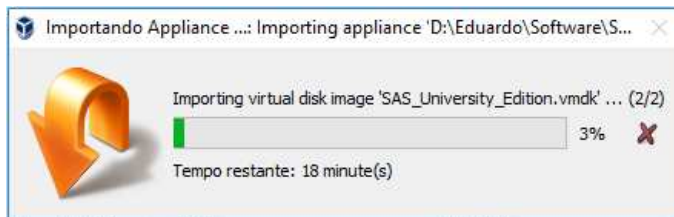


Figura 6: **Progresso da importação do SAS vApp**

Agora é preciso criar as pastas para uso com o SAS. Para isso, abra o Windows Explorer (Figura 7).

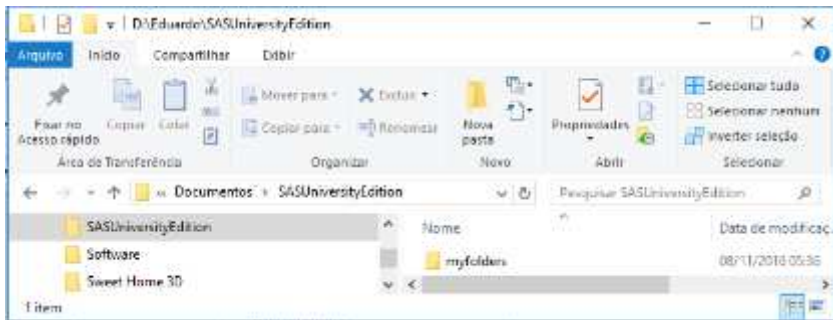


Figura 7: Criando pastas no Windows para uso com o SAS

Na pasta dos seus documentos, crie uma pasta chamada “SASUniversityEdition” e, dentro dela, crie uma subpasta denominada “myfolders”. É preciso configurar o VirtualBox para usar a pasta “myfolders”. Na janela do VirtualBox, selecione o SAS University Edition clicando nele (Figura 8).

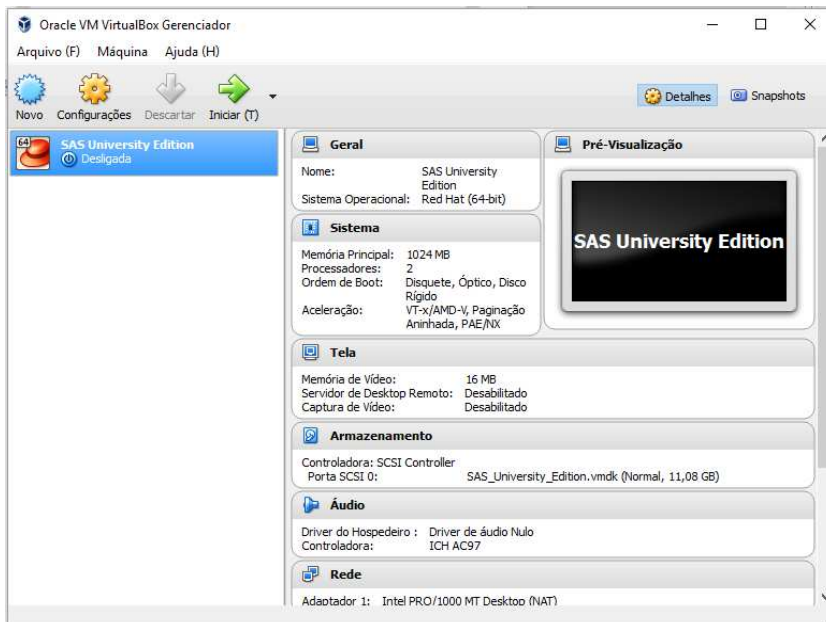


Figura 8: Selecionando o SAS no VirtualBox para adicionar as pastas

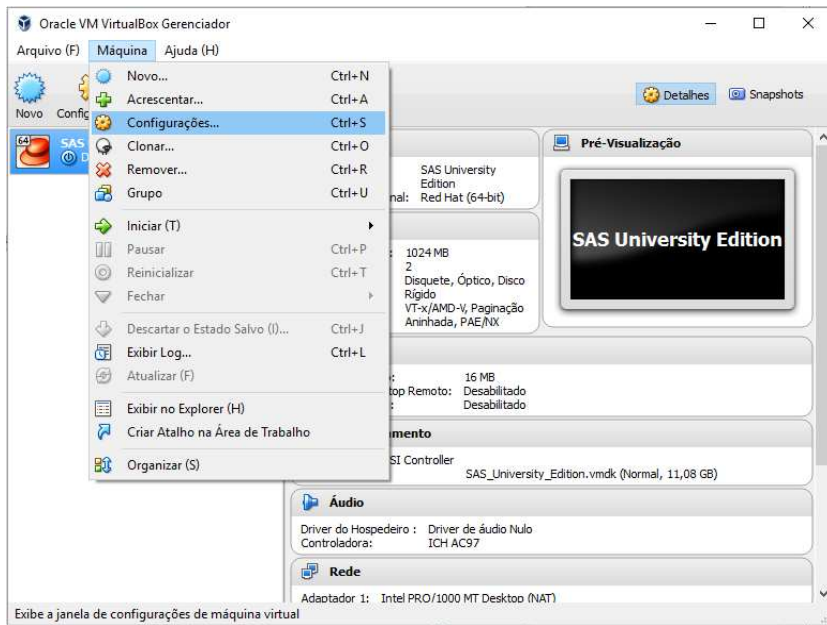


Figura 9: Configurando pastas para usar com o SAS vApp

Então, selecione | Máquina | e | Configurações | para configurar o uso das pastas criadas no Windows (Figura 9). A tela de configurações se abre. Nesta tela, selecione | Pastas Compartilhadas |. Então, clique no ícone “adicionar pastas” no canto direito da janela (Figura 10).

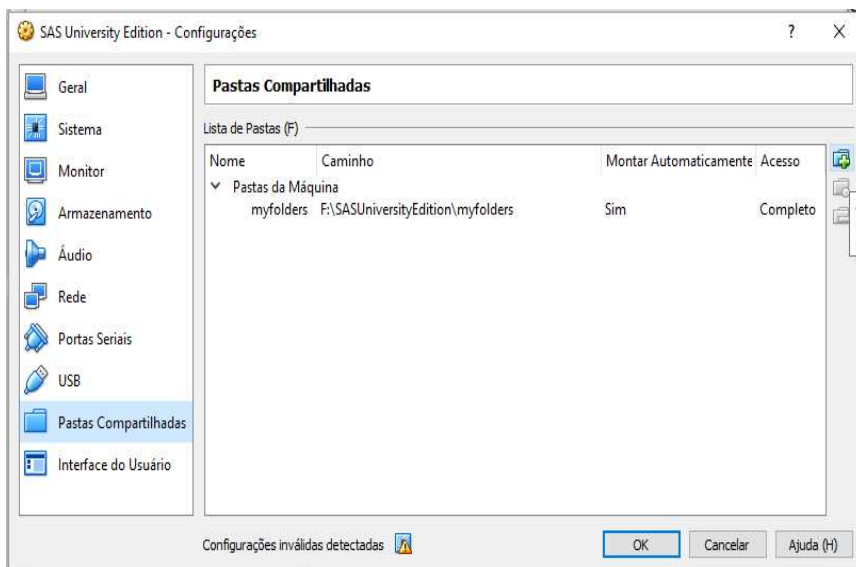


Figura 10: Janela de configurações do SAS University Edition vApp

Com isso será aberta a janela de adição de pastas. Clique no botão à direita de | Caminho da Pasta | e, depois, selecione | Other...| (Figura 11).

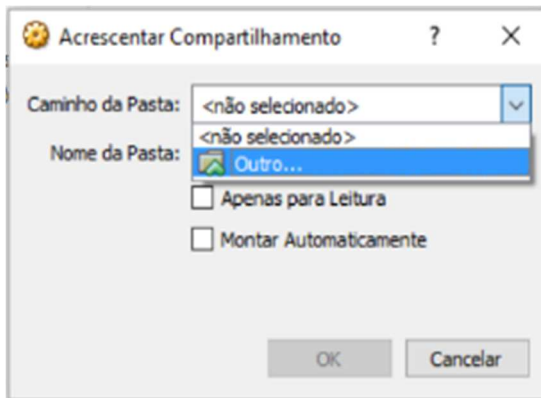


Figura 11: **Janela de adição de pastas**

Na próxima janela (Figura 12), navegue para encontrar a pasta “myfolders” dentro da pasta “SasUniversityEdition” e a selecione , depois clique em | OK |.

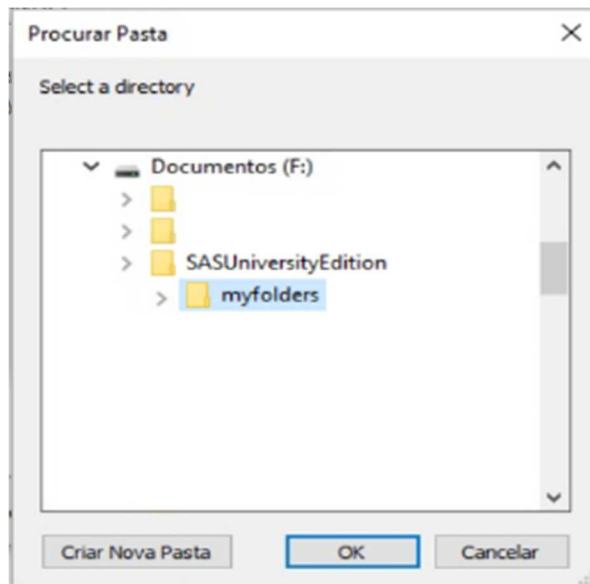


Figura 12: **Janela de navegação**

A última janela se fecha e volta-se para a anterior, onde é preciso selecionar | Montagem Automática | (Figura 13). Depois, clique em |OK |. Volta-se para a tela de configurações, onde se deve clicar em | OK | novamente para fechar a janela. A pasta “myfolders” está pronta para ser utilizada.

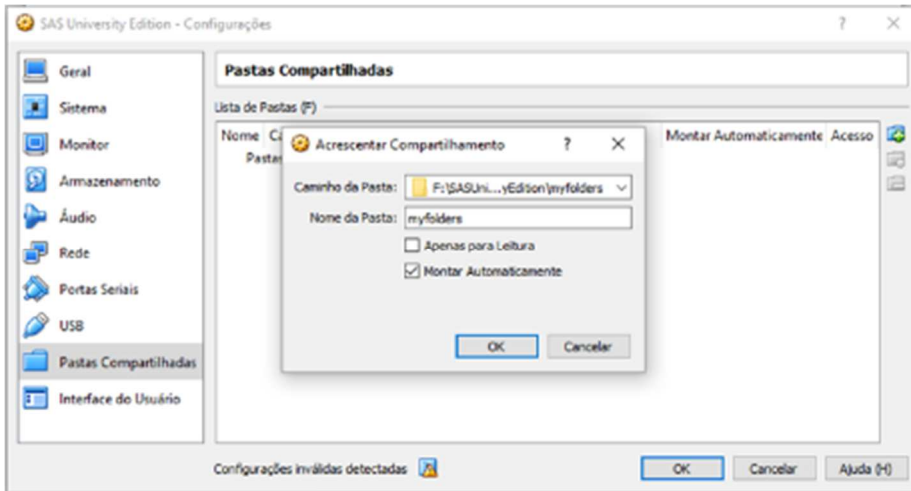


Figura 13: Terminando a configuração da subpasta "myfolders"

Nesta etapa é preciso inicializar o SAS University Edition vApp. Para isso, seleccione-o no VirtualBox novamente, clicando sobre ele. Então seleccione | Máquina | Iniciar | Iniciar normal |, como segue (Figura 14).

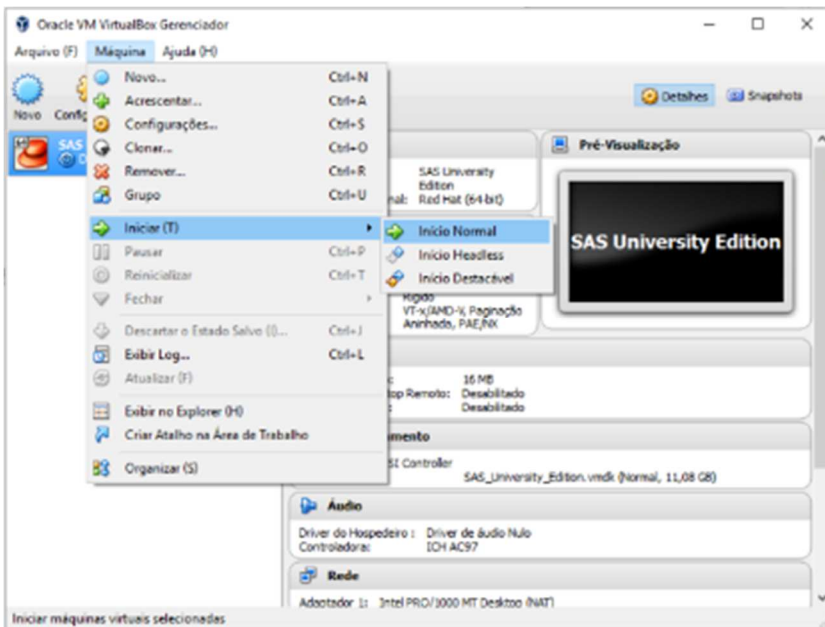


Figura 14: Inicializando o SAS University Edition vApp

A máquina virtual pode demorar para inicializar. Enquanto isso, aprecie o progresso da inicialização na janela a seguir (Figura 15).

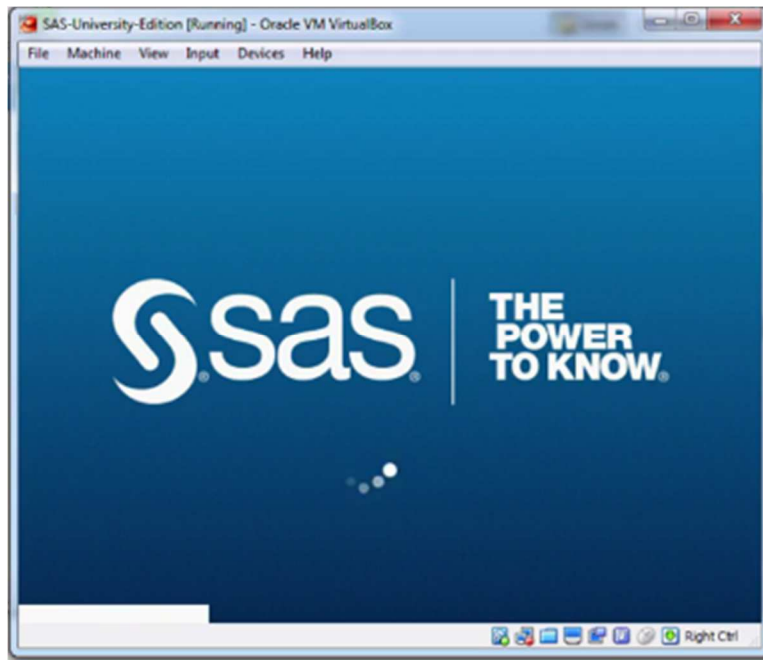


Figura 15: Progresso da inicialização do SAS University Edition

Ao encerrar, aparecerá a tela da máquina virtual do SAS University Edition (Figura 16), que não pode ser fechada para usar o SAS. Copie manualmente o endereço na área marcada em verde. É um endereço semelhante ao abaixo:

`http://localhost:10080`

Escreva este endereço no seu navegador para acessar o SAS Studio e tecele entre.

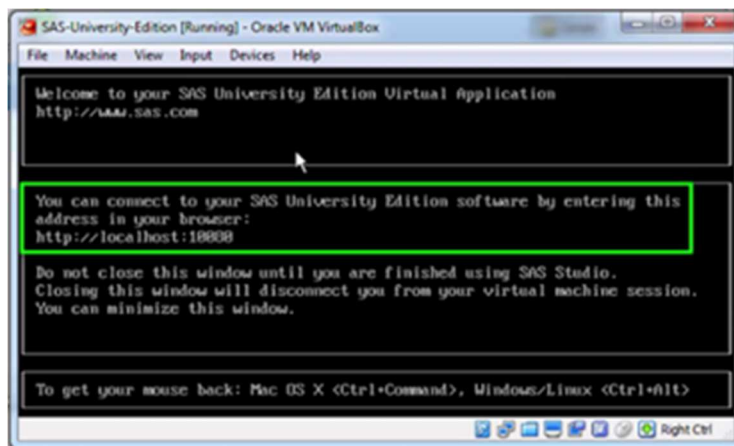


Figura 16: Tela da máquina virtual do SAS University Edition

Quando a tela de Welcome com o link do SAS Studio se abrir (Figura 17), teclae\ [Ctrl] e [D], ao mesmo tempo, para colocar o endereço nos seus preferidos, isso irá facilitar acessar o SAS Studio para ser usado na próxima vez.

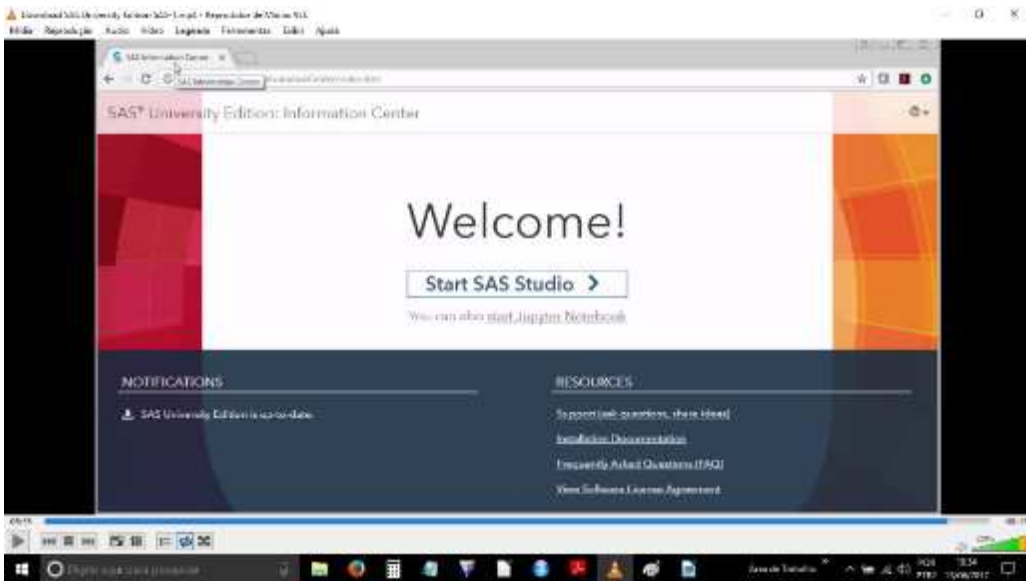


Figura 17: **Janela do SAS® University Edition - Information Center**

Na janela que se abre no navegador, clique no botão para acessar a janela de inicialização do SAS Studio. Clique no botão de inicialização do SAS Studio e aguarde até a tela da interface do SAS Studio abrir.

O SAS está pronto para ser usado pela primeira vez.

Nas próximas vezes, abra o VirtualBox, inicialize o SAS University Edition no VirtualBox e abra o navegador. Então, clique no link para o SAS Studio que você criou nos preferidos para iniciá-lo.

2.4 INTERFACE SAS STUDIO

Ao abrir o SAS Studio no navegador, nos deparamos com sua interface (Figura 18) que não é muito amigável de início; ela é um misto da interface do antigo SAS System com a do SAS Enterprise Miner. A janela divide-se em dois campos, um **painel de navegação** à esquerda e um **painel de trabalho** à direita.

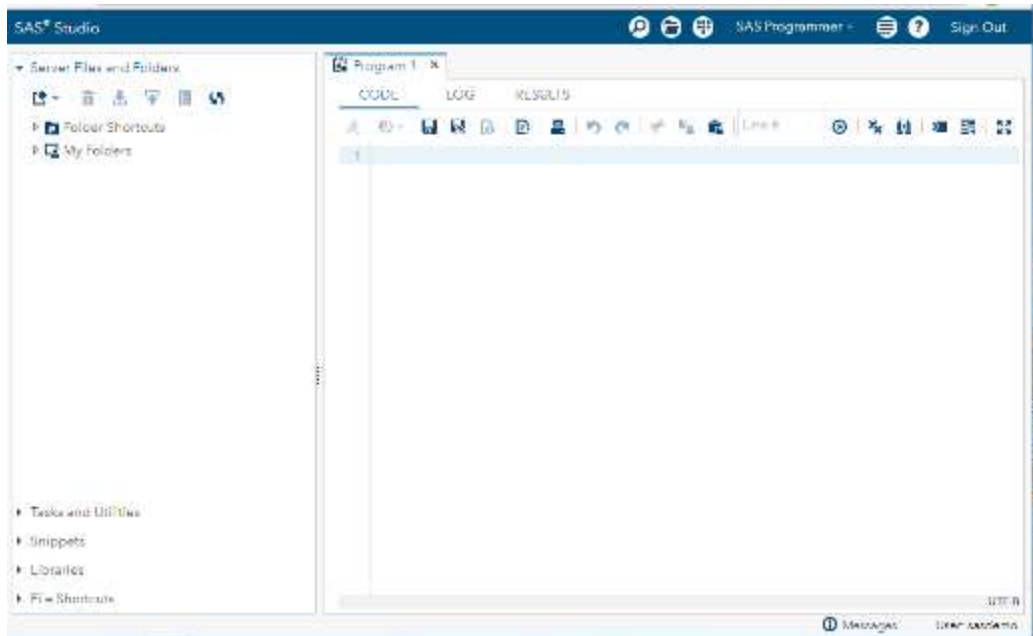


Figura 18: Tela da interface do SAS Studio

O painel de navegação é composto pelos menus a seguir:

- Server Files and Folders (servidor de arquivos e pastas)
- Folders Shortcuts (atalhos de pastas)
- My Folders (minhas pastas)
- Tasks and Utilities (tarefas e utilitários)
- Snippets (fragmentos)
- Libraries (bibliotecas)
- File Shortcuts (atalhos de arquivos)

No painel de trabalho, ao abrir o SAS Studio, encontram-se três abas na parte superior:

- CODE,
- LOG,
- RESULTS.

Sendo que acima da aba CODE está indicado o nome do programa em uso (Program 1 – enquanto não for aberto nenhum programa durante a sessão de trabalho).

2.4.1 SERVER FILES AND FOLDERS (SERVIDOR DE ARQUIVOS E PASTAS)

Abaixo de Server Files and Folders encontram-se seis ícones, sendo que o primeiro abre um menu e os demais são atalhos:



Clicando na seta do primeiro ícone abre-se o menu da Figura 19, com as seguintes funções:

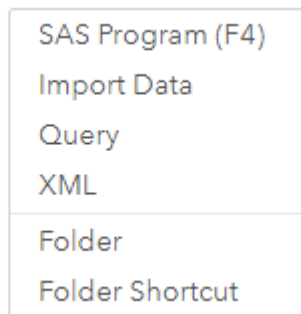







Figura 19: **Menu inicial do painel de navegação**

- O primeiro item do menu [SAS Program (F4)] inicia a criação de um novo programa,
- o segundo [Import Data] leva à importação de dados,
- o terceiro [Query] inicia a criação de buscas do tipo Query,
- o quarto [XML] inicia a criação de um arquivo XML,
- o quinto [Folder] inicia a criação de uma nova pasta,

- e, o sexto item [Folder Shortcut] inicia a criação de um atalho para uma pasta.

Os demais ícones são atalhos com as funções de:

-  Delete - excluir o que estiver selecionado
-  Download - baixar para o SAS o que for informado
-  Upload - carregar o que estiver selecionado para outro local
-  Properties - apresenta as propriedades do que estiver selecionado
-  Refresh - atualiza as pastas selecionadas

2.4.2 FOLDERS SHORTCUTS (ATALHOS DE PASTAS)

Este item do menu contém os atalhos criados pelo usuário para pastas do SAS.

2.4.3 MY FOLDERS (MINHAS PASTAS)

My folders destina-se a conter as pastas criadas pelo usuário com seus arquivos e programas.

O usuário pode adicionar outras pastas de trabalho além da pasta “myfolders” criada no início, quando da instalação do SAS no VirtualBox. Para adicionar novas pastas de trabalho, ative a interface da máquina virtual (Figura 20) e clique em | Máquina| e depois em | Configurações |.

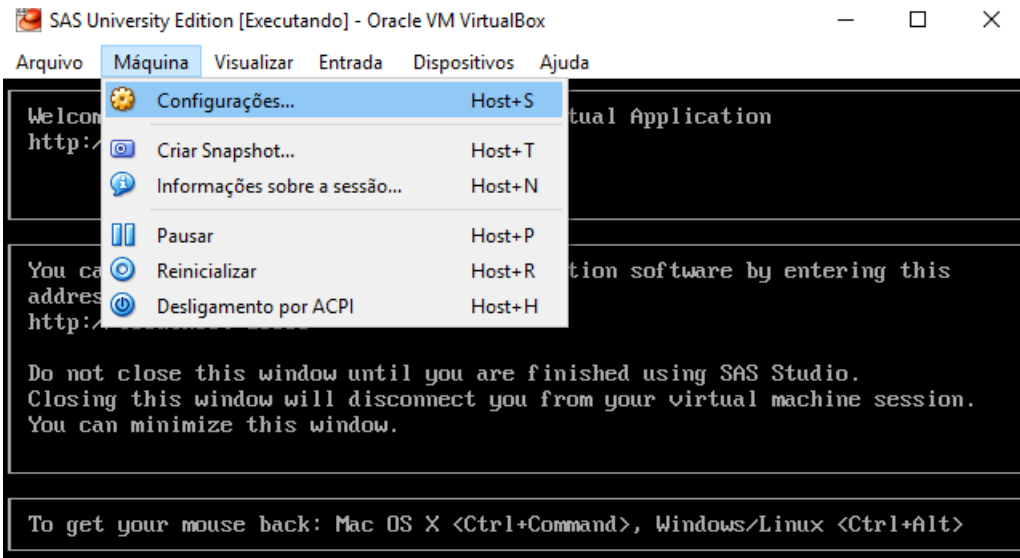



Figura 20: Acessando as configurações na interface da máquina virtual

Na janela de configurações que se abre (Figura 21), clique em | Pastas Compartilhadas | e depois no ícone  que se encontra à direita na parte superior da janela, para acrescentar pastas.

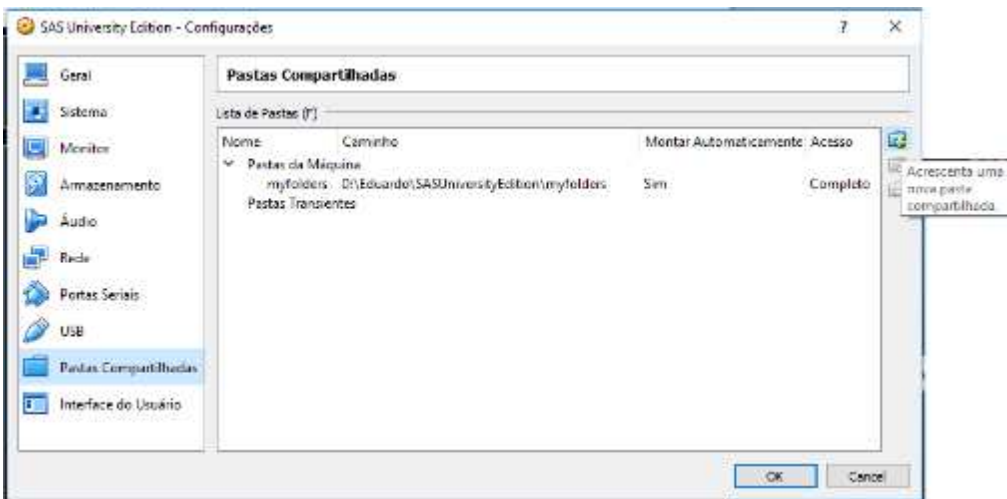


Figura 21: Janela de configurações da máquina virtual do SAS

Procure a pasta que deseja acrescentar e adicione. Depois, reinicie a máquina virtual do SAS no VirtualBox e o SAS Studio no navegador para efetivar a adição da pasta.

2.4.4 TASKS AND UTILITIES (TAREFAS E UTILITÁRIOS)

Tasks and Utilities possui um menu com três pastas: My Tasks, Tasks e Utilities. My Tasks é a pasta que se destina a conter tarefas que você irá desenvolver e gravar para uso futuro. Tasks é a pasta que contém tarefas prontas para serem usadas; você também pode copiar os códigos (programas) das tarefas prontas para usar como parte de outros programas. Utilities são utilitários (programas) para facilitar algumas ações que são frequentes no SAS.

2.4.4.1 UTILITIES

Utilities são programas utilitários para ações frequentes como a importação de arquivos.

Para aprender a usar utilitários, vamos criar a planilha Arvores do Quadro 1 no MS-Excel e salvar na pasta “myfolders” que criamos no Windows. Note que não se deve usar acentos ou caracteres especiais nos nomes dos arquivos, das colunas e nem das planilhas. Após criar e salvar a planilha, é preciso clicar na pasta myfolders no SAS e depois no ícone de atualização [Refresh] para que o arquivo do Excel apareça na pasta My Folders do SAS.

Quadro 1 - Planilha Arvores

	A	B	C	D
1	Arvore	d	h	
2	1	20,7	17,0	
3	2	20,7	15,0	
4	3	20,7	15,0	
5	4	21,0	16,0	
6	5	23,9	23,0	
7	6	23,9	22,0	
8	7	24,0	23,5	
9	8	24,2	25,9	
10	9	27,1	28,7	
11	10	26,0	26,5	
12	11	27,1	27,0	
13	12	27,4	28,2	
14	13	29,9	28,1	
15	14	29,9	29,1	
16	15	30,2	28,7	
17	16	30,6	29,0	
18	17	33,4	29,7	
19	18	33,4	30,1	
20	19	33,4	28,7	
21	20	33,4	30,0	
22	21	36,3	31,0	
23	22	36,6	30,2	
24	23	36,6	32,0	
25	24	36,9	29,8	
26	25	39,8	29,8	
27	26	40,1	33,0	
28	27	40,1	34,0	
29	28	40,1	33,0	

Arvores

2.4.4.1.1 IMPORTANDO ARQUIVOS DO EXCEL

Clique na pasta Tasks and Utilities, depois em [Utilities] e dê um clique duplo em [Import Data]. No painel de trabalho aparecerá um quadro onde se deve clicar no botão [Select File] (Ver Figura 22). Abrirá a janela OPEN na sequência (Figura 23). Então, dê um clique duplo em [My Folders] e selecione o arquivo do Excel denominado Arvores.xlsx que foi salvo aí, depois clique [Open].

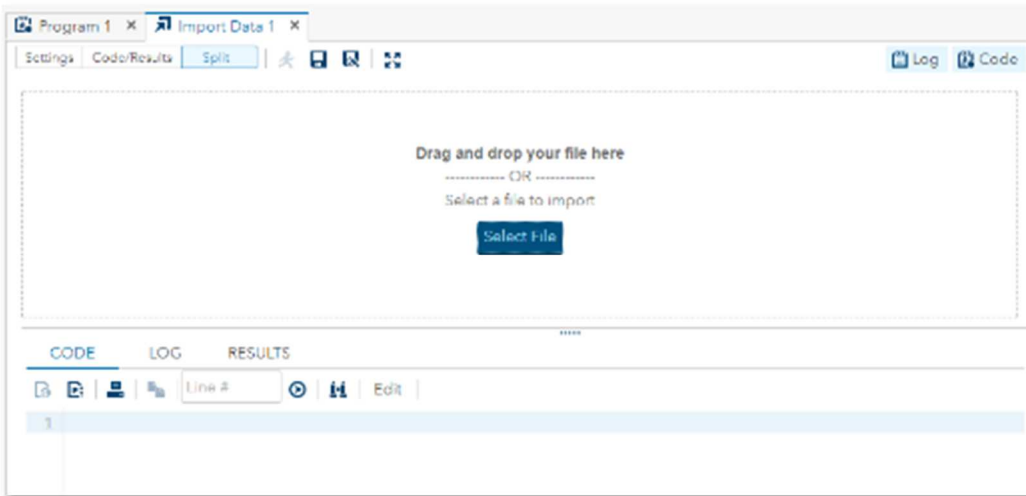


Figura 22: **Painel de trabalho com o botão |Select File|**

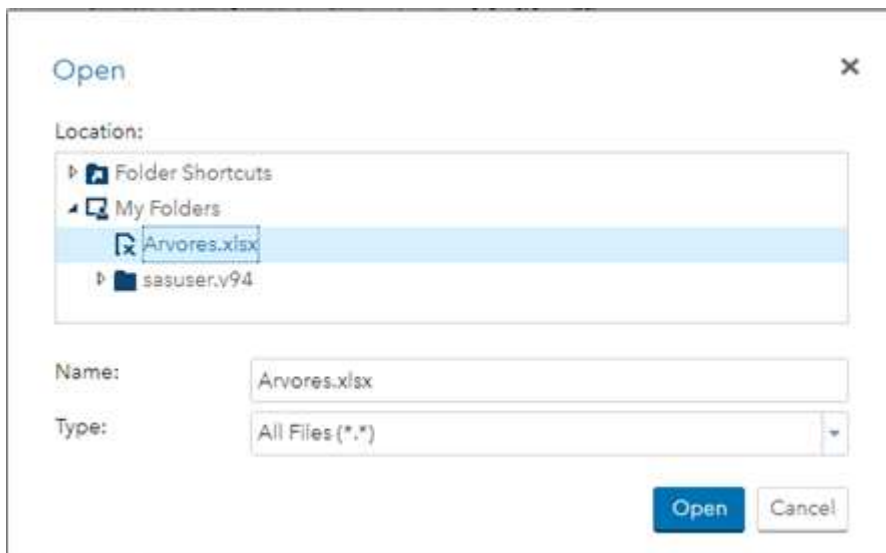


Figura 23: **Janela de importação de arquivos.**

Será criada uma nova aba com o programa para importar o arquivo. Agora, basta rodar o programa clicando em |Run| para que o arquivo do Excel seja transformado em arquivo SAS e gravado na pasta |Work| podendo ser utilizado em programas SAS.

O código do programa de importação é reproduzido a seguir (Quadro 2):

Quadro 2: Programa gerado pelo SAS para importar o arquivo Arvores.xlsx

```
/* Generated Code (IMPORT) */
/* Source File: Arvores.xlsx */
/* Source Path: /folders/myfolders */
/* Code generated on: 05/07/17 17:12 */
%web_drop_table(WORK.IMPORT);
FILENAME REFFILE '/folders/myfolders/Arvores.xlsx';
PROC IMPORT DATAFILE=REFFILE
      DBMS=XLSX
      OUT=WORK.IMPORT;
      GETNAMES=YES;
RUN;
PROC CONTENTS DATA=WORK.IMPORT; RUN;
%web_open_table(WORK.IMPORT);
```

As primeiras quatro linhas do programa iniciam por /* e finalizam com */, convertendo-as em comentários que não são processados.

Este código pode ser modificado para tornar o arquivo SAS permanente, alterando-se a pasta onde deve ser gravado, pois a pasta |Work| só mantém os arquivos temporariamente.

Para tornar o arquivo Arvores permanente e poder utilizá-lo diretamente sem necessitar importar novamente, é preciso informar a pasta em que deve ser gravado no SAS. Vamos gravá-lo na pasta My Folders, utilizando o comando LIBNAME para informar em que pasta, com a seguinte frase: LIBNAME pasta "/Folders/myfolders"; e, modificamos o programa que ficará como a seguir (Quadro 3):

Quadro 3: Programa para importar o arquivo Arvores.xlsx tornando-o permanente

```
LIBNAME pasta '/folders/myfolders';
FILENAME arquivo '/folders/myfolders/Arvores.xlsx';
PROC IMPORT DATAFILE=arquivo
      DBMS=XLSX
      OUT=pasta.Arvores;
      GETNAMES=YES;
RUN;
PROC CONTENTS DATA=pasta.Arvores;
RUN;
```

2.4.4.2 TASKS (TAREFAS)

O menu Tasks abre-se em oito submenus:

- Data: tarefas para operar com dados;

- Graph: tarefas para construir gráficos;
- Combinatorics and Probability: tarefas para análise combinatória e probabilidades;
- Statistics: tarefas para calcular estatísticas;
- Power and Sample Size: tarefas para fins diversos como estatísticas de amostragem, correlações, regressões, testes não paramétricos, etc;
- Multivariate Analysis – tarefas para análise multivariada;
- Cluster Analysis- tarefas para análise de agrupamento;
- Forecasting – tarefas para prognose e análise de séries temporais.

2.4.4.2.1 CONSTRUINDO UM GRÁFICO DE DISPERSÃO

Para aprender a utilizar Tasks (tarefas) vamos construir um gráfico de dispersão com os dados do arquivo Arvores. Para isso abra o menu [My Folders] e dê um click duplo no arquivo [Arvores] para ativá-lo. Depois, abra a pasta [Tasks and Utilities], depois [Tasks] e [Graph]. Selecione [Scatter Plot] com clique duplo; abrirá uma janela com um formulário no painel de trabalho (Figura 24), onde iremos fornecer as informações necessárias para criar um gráfico das Alturas na ordenada e dos Diâmetros na abscissa.

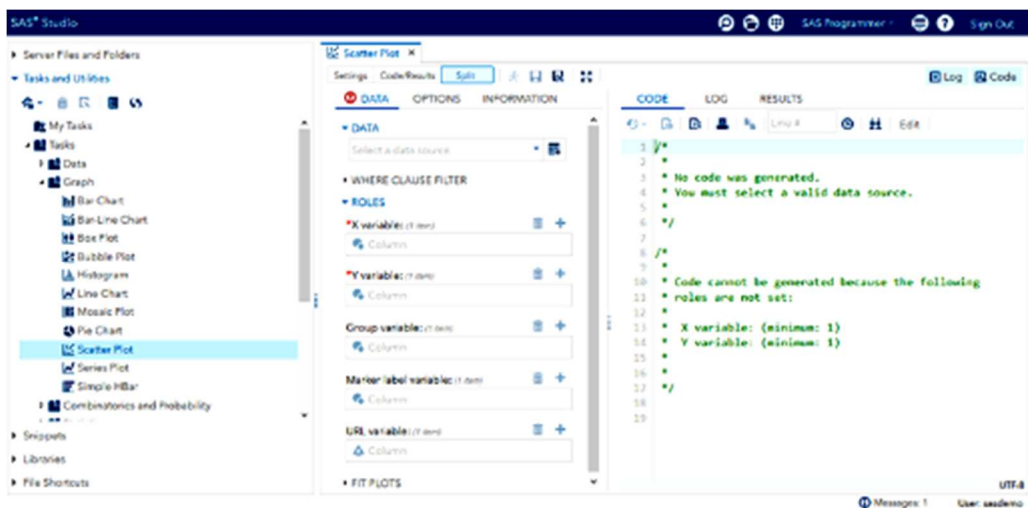


Figura 24: Criando um gráfico de dispersão

Em [DATA] no formulário, clique no botão à direita do espaço destinado ao nome do arquivo. Na janela de navegação que apareceu (Figura 25), encontre e selecione o arquivo [Arvores]; clique em [OK].

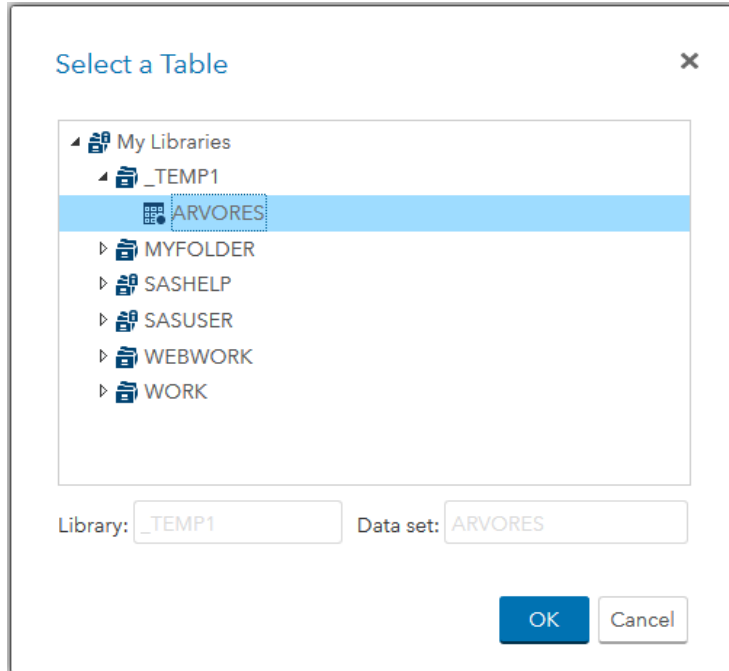


Figura 25: **Selecionando o arquivo Arvores para fazer um gráfico**

No formulário, no espaço para a variável x, clique em + e selecione a variável d do arquivo; para a variável y, selecione h (Figura 26).

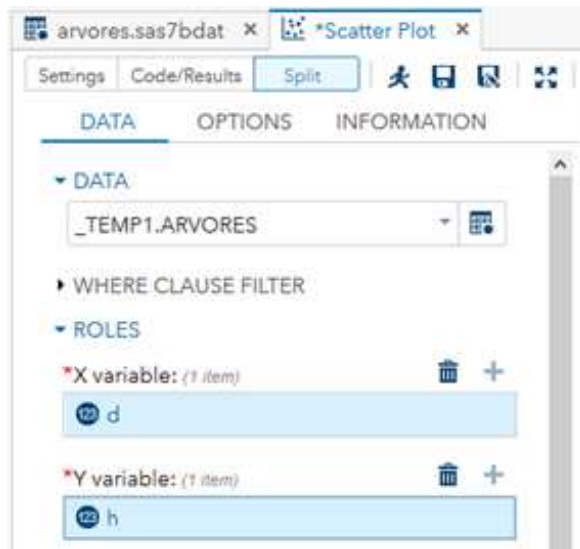



Figura 26: **Selecionando dados para criar um gráfico tipo Scatter Plot**

No painel de trabalho (Figura 27), clique na aba |CODE|, depois em |  | e aguarde o gráfico ser construído (Figura 28).

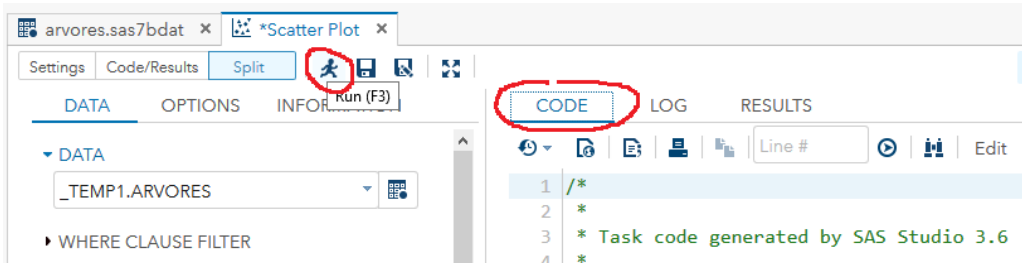


Figura 27: **Executando o programa para construir o gráfico**

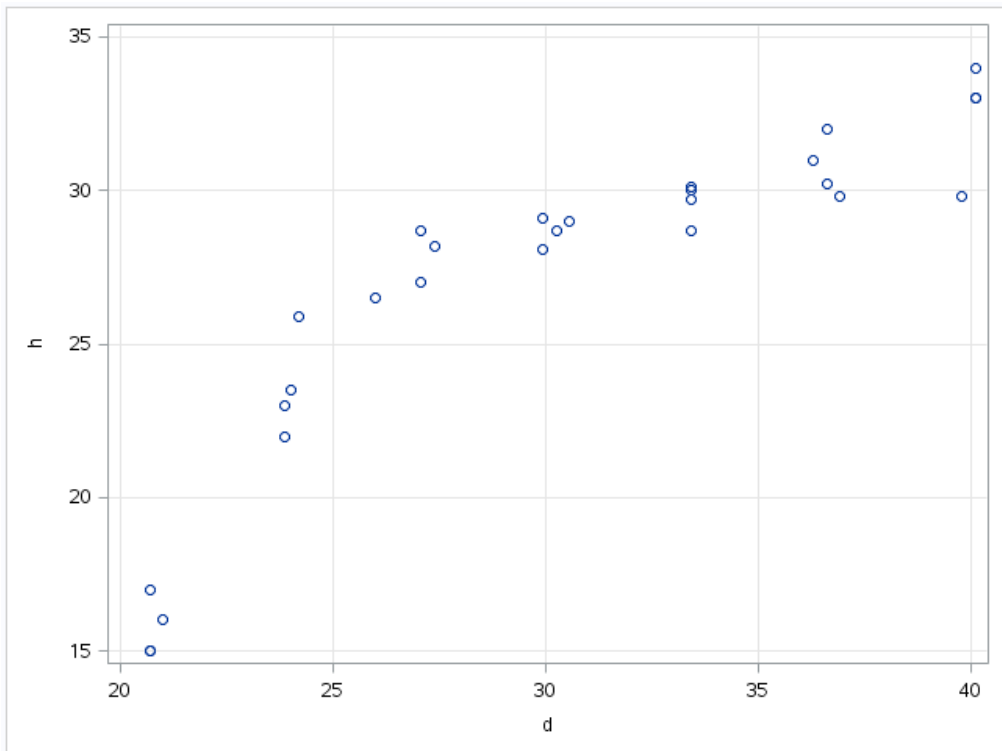


Figura 28: **Gráfico construído com as variáveis d e h**

O código do programa (Quadro 4) pode ser salvo para uso posterior e utilizado como exemplo, podendo-se alterar o arquivo a ser executado e as variáveis utilizadas.

Quadro 4: Código do programa para construir o gráfico com as variáveis x e y

```
ods graphics / reset imagemap;  
proc sgplot data=_TEMP1.ARVORES;  
    scatter x=d y=h / transparency=0.0 name='Scatter';  
    xaxis grid;  
    yaxis grid;  
run;  
ods graphics / reset;
```

2.4.5 SNIPPETS (FRAGMENTOS)

Snippets são fragmentos de códigos de programas que podem ser personalizados para as necessidades do usuário e incluídos em outros programas, facilitando o

desenvolvimento do código de algumas ações, sendo divididos nas seguintes categorias:

- Data
- Catalogs
- Descriptive
- Graph
- IML
- Macro

2.4.5.1.1 DATA (DADOS)

São trechos de programas para importação e exportação de arquivos como: CSV, XLSX e XML. Também, há outros para gerar arquivos de simulação.

2.4.5.1.2 CATALOGS (CATÁLOGOS)

Trechos para trabalhar com catálogos, SQL e criação de saídas em HTML, PDF e RTF, entre outras ações.

2.4.5.1.3 GRAPH (GRÁFICOS)

Servem para elaboração de gráficos.

2.4.5.1.4 IML (MATRIZES)

Trechos para realizar operações com matrizes.

2.4.5.1.5 MACRO (MACROS)

Trechos para criação de macros. Macros são subprogramas que podem ser chamados de qualquer parte de um programa e são úteis para ações que se repetem em várias partes de um programa, onde são passados parâmetros para o subprograma e obtida uma resposta, semelhante ao que fazem as funções.

2.4.6 LIBRARIES (BIBLIOTECAS)

São as bibliotecas do SAS. Para criar uma nova biblioteca, é preciso que a pasta já exista no Windows, então é usado o comando LIBNAME para torna-la permanente com a seguinte sintaxe: *LIBNAME nomep 'Caminho'*;

Exemplo:

```
LIBNAME pasta '/folders/myfolders/';
```

```
RUN;
```

* Este programa de duas linhas, quando executado cria uma pasta permanente denominada de *pasta* em [Libraries], [My libraries] conforme a Figura 29.

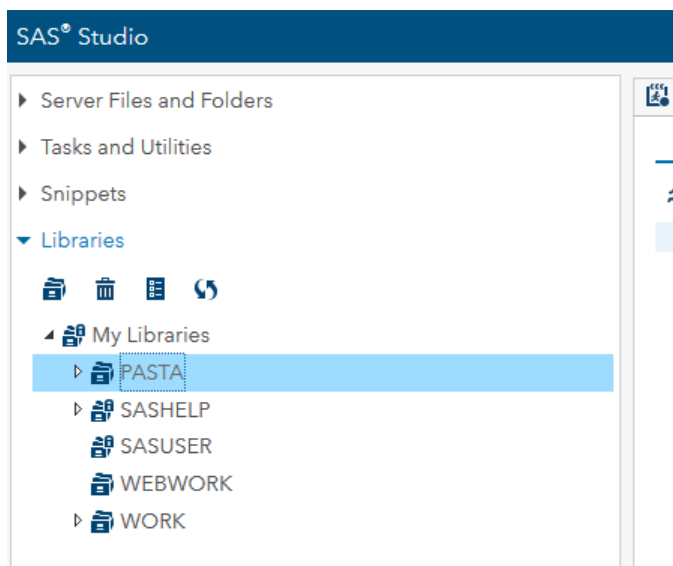



Figura 29: Pastas em [Libraries], [My Libraries]

Na pasta [SASHELP] se encontram exemplos de arquivos SAS usados em programas de exemplo da ajuda do SAS.

A pasta [WORK] guarda os arquivos temporários criados nos programas que foram executados na última sessão com o SAS Studio.

2.4.7 FILE SHORTCUTS (ATALHOS DE ARQUIVOS)

Esta pasta destina-se a conter atalhos para arquivos de dados ou de programas criados pelo usuário. Para criar um novo atalho clique [File Shortcuts] (Figura 30) e depois em  , para abrir a janela de criação de atalhos (Quadro 5).

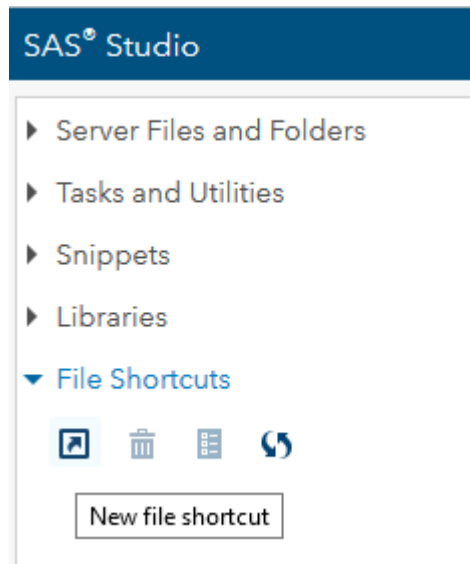
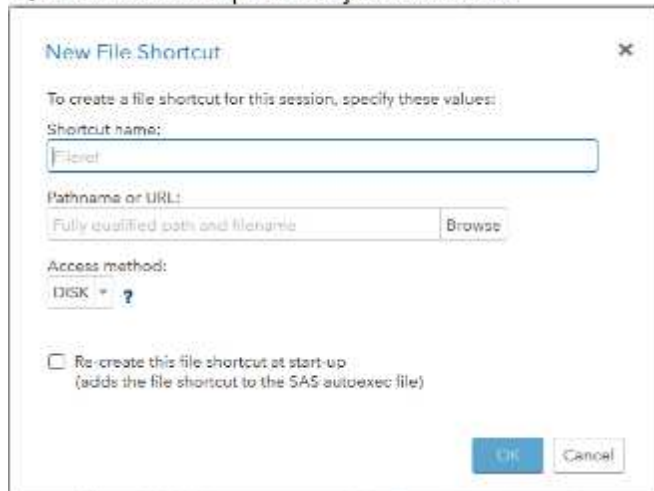


Figura 30: Criação de um novo atalho

Quando a janela de criação de atalhos abrir (Quadro 5), forneça um nome de até 8 dígitos para o atalho no espaço |Shortcut Name| e o caminho para o arquivo em |Pathname or URL|. Em |Access Method|, escolha a opção Disk se o arquivo estiver no computador, ou URL se o arquivo estiver na Web. Depois clique em |OK| que o atalho será criado.

Quadro 5: Janela para criação de atalhos



Capítulo 3 : Introdução à linguagem SAS

A programação SAS a que se refere este manual a partir deste capítulo é própria para o SAS tradicional, versões 8.1 em diante. A maioria dos programas de exemplo funcionam no SAS University Edition, mas em alguns casos são necessárias pequenas alterações. Por exemplo, o procedimento REG que faz análise de regressão linear e múltipla não aceita o comando PLOT no SAS University Edition e emitirá um aviso de erro no LOG após a execução, se houver comandos não permitidos que devem ser excluídos.

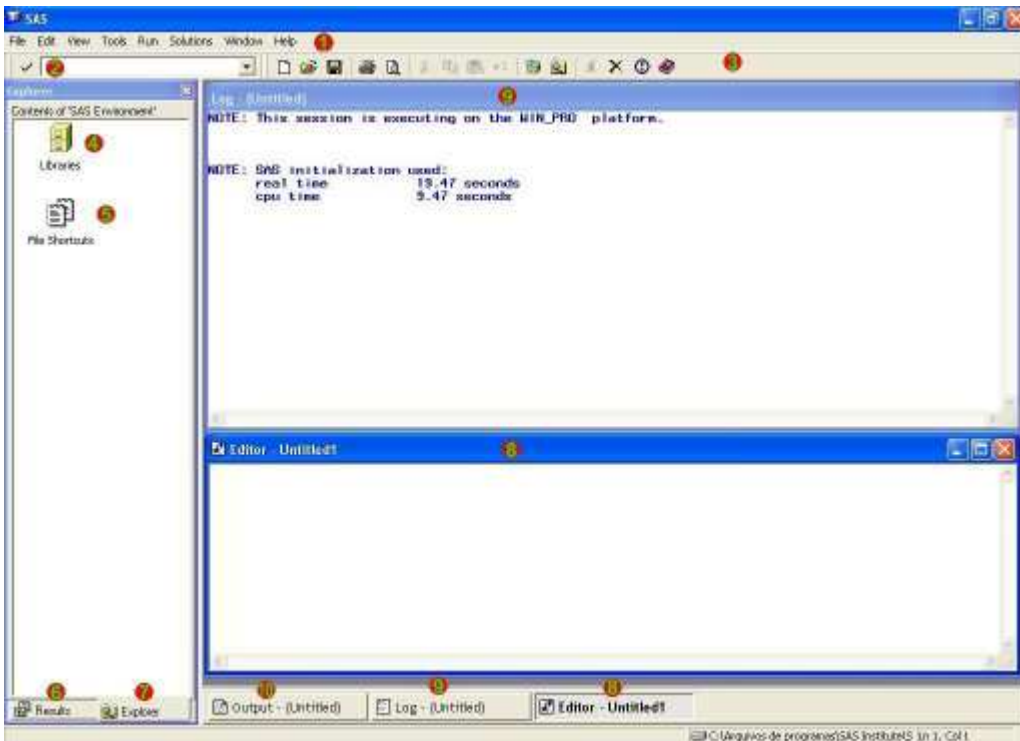
3.1 CONVENÇÕES UTILIZADAS

As principais convenções e metodologias de descrição da linguagem SAS utilizados, sejam próprias do SAS, ou deste manual, são descritas a seguir. Os manuais do SAS constituem uma enciclopédia de muitos milhares de páginas que descrevem todas as suas possibilidades. Neste manual são descritos somente o que se considerou essenciais para se usar na área biológica, com especial atenção às aplicações na Engenharia Florestal.

Muitas opções dos procedimentos foram omitidas para se evitar escrever um volume muito extenso. Para informações sobre todas as possibilidades do SAS, o leitor deve acessar seus manuais *on line*, disponíveis para *download* em PDF no site do SAS Institute <<http://support.sas.com/documentation/>> para o SAS tradicional, ou no site <<http://documentation.sas.com/>> para o SAS University Edition e SAS Studio.

Menus

As referências no texto, feitas aos menus da **barra de menus** do SAS (Figura 31), são realizadas sequencialmente, conforme sua hierarquia, entre barras verticais, como em |FILE|SAVE|, que identifica o menu |File| e seu submenu |Save|, utilizado para acessar a janela de salvamento de arquivos.



- | | |
|---------------------|---------------------------------|
| 1. Barra de menus | 6. Aba de resultados |
| 2. Endereço | 7. Aba do explorador |
| 3. Barra de Tarefas | 8. Janela do editor |
| 4. Bibliotecas | 9. Janela do arquivo de log |
| 5. Atalhos | 10. Janela do relatório (saída) |

Figura 31: Interface tradicional do SAS, versões 6 a 9

Exemplos

Neste manual, a partir deste capítulo, os exemplos e linhas de programa são escritos em **negrito tamanho 8**, limitados à esquerda por uma barra vertical cinza de 6 pontos de largura com recuo de 4 pontos à esquerda, seguidos de uma outra barra e de uma coluna com números ilustrativos à direita, os quais não fazem parte do programa. Esses servem exclusivamente para fazer referências explicativas às linhas do programa no texto, como nas duas linhas de comentário (Linhas 1 e 2) a seguir:

<p>█ * exemplo de linha de comentário no SAS;</p> <p>█ /* uma linha de comentário também pode ser assim. */</p>	<p>█ 1</p> <p>█ 2</p>
-------------------------------------------------------------------------------------------------------------------------------	-------------------------------------

Sintaxe

Sintaxe é o formato geral de um comando, instrução ou declaração na programação SAS, que deve ser seguido para que funcione.

As linhas de sintaxe são escritas em negrito tamanho 8, limitados à esquerda por uma barra vertical cinza de 6 pontos de largura, com recuo à esquerda de 1,5 cm, como a seguir:

```
■ PROC SORT <DATA=ARQUIVODEENTRADA>  
  <OUT=ARQUIVODESAIDA> <OPÇÕES>;  
  BY <DESCENDING> VARIABEL-1 ...<DESCENDING> VARIABEL-N;
```

As descrições de sintaxe são compostas por partes obrigatórias e partes opcionais.

As partes opcionais na descrição da sintaxe sempre estão entre os sinais < > (que não devem ser escritos no programa). Os nomes e outras palavras que devem ser definidos pelo usuário ou programador são especificados sem espaços, com as palavras justapostas, sendo a primeira letra de cada palavra escrita em maiúsculo e as demais em minúsculo, como no exemplo a seguir, onde a expressão '*nome do array*' está escrita como **NomeDoArray** e a palavra 'subscrito' está escrita como **Subscrito**:

```
■ NAMEDOARRAY{SUBSCRITO}<$>;
```

As palavras utilizadas com todas as letras maiúsculas são palavras do sistema e as que usam a primeira letra de cada palavra (única ou justaposta) são nomes que o usuário deve escolher, sejam de opções do sistema ou nomes definidos pelo usuário.

Valores perdidos

Valor perdido (ou faltante), é entendido como a inexistência de conteúdo em um ou mais registros de uma ou mais variáveis de um arquivo SAS. Quando não existe valor em uma variável, o caracter utilizado é o ponto (.).

Portanto, sempre que houver falta de um valor nos arquivos de dados, nos vetores, ou em matrizes, este deve ser informado como um ponto (.).

Quando o conteúdo de um arquivo é impresso e faltam valores em registros e colunas, é impresso um ponto (.) no local de cada valor que falta.

Valores perdidos são sempre os primeiros de um ordenamento ascendente e os últimos no descendente.

Em experimentos com parcelas perdidas, o seu valor deve ser informado como um ponto (.). Então, o sistema entende que aquela parcela foi perdida e não a considera. Entretanto, se o resultado nulo deve ser considerado como consequência do efeito dos

tratamentos, o valor que deve ser informado para que o sistema o considere é zero (0).
Observação: alguns procedimentos do SAS permitem que o usuário informe que os valores perdidos devem ser considerados na análise.

3.2 PROGRAMAS SAS: EDIÇÃO, EXECUÇÃO E COMPONENTES

Há duas formas para processamento de dados no SAS em ambiente WINDOWS, a interativa e a programável. Na forma interativa, não se escrevem programas, apenas se deve gerar os arquivos de dados e solicitar, através de menus e janelas, o que se deseja que o sistema faça com os dados. No modo programável, deve-se escrever os programas e depois executá-los. Este manual se restringe à forma programável.

Há, também, dois modos para execução de programas SAS no ambiente WINDOWS, o modo interativo através de janelas e o modo de lotes. Neste manual será tratado somente do modo programável interativo, ou seja, criar arquivos de dados, escrever programas e executá-los através da interface gráfica do SAS para WINDOWS.

Programas SAS em microcomputadores, aos quais se restringe este manual, são escritos na área de edição do SAS (*Editor*) e são constituídos por **passos** de programação (DATA STEP e PROC STEP). Cada passo de programa é constituído por **declarações** (frases) e cada declaração é constituída por **palavras** (Ver Capítulo 5).

É aconselhável que sejam incluídas linhas de identificação no início do programa, através de frases de comentário, iniciando-as por um asterisco seguido de um espaço branco, o que torna a frase sem efeito na programação, podendo-se escrever, na sequência, qualquer palavra, letra ou número, finalizando a linha com ponto-e-vírgula. A identificação do programa pode incluir o nome do mesmo, o que ele realiza, a data e o nome do programador. Pode-se incluir tantas frases de comentário quantas forem necessárias, iniciando cada uma por um asterisco, seguido de espaço branco e finalizando com ponto-e-vírgula.

Um programa SAS em ambiente WINDOWS deve ter, como últimas, as linhas:

RUN;	1
quit;	2

Em que: 'RUN;' é a linha que diz ao sistema operacional que execute o programa; 'QUIT|EXIT;' é a linha que instrui o sistema operacional para encerrar a execução do programa (é de uso opcional, mas faz com que o programa não fique aguardando novos passos e encerre mais rapidamente).

Um programa SAS típico para WINDOWS, portanto, é composto por:

* Linhas de comentário (identificação do programa);	1
Data Step;	2
instruções;	3
Proc Step;	4

```
instruções;  
Run;  
quit;
```

```
5  
6  
7
```

O SAS para WINDOWS possui três áreas principais (ou janelas) relacionadas à programação básica e estatística: *Editor*, *Log* e *Output*.

No *Editor* são editados os arquivos de programas e de dados, que devem ser salvos da mesma forma que os textos do MS-WORD, mas devem ter a terminações “.SAS” e “.DAT”, respectivamente. Um arquivo SAS de programa, ou de dados, pode ter um nome composto da mesma forma que qualquer outro arquivo WINDOWS.

O *Log* é um relatório com a descrição do que aconteceu durante a execução do programa.

O *Output* é o relatório de saída com os resultados do processamento.

Exemplos de nomes de programas SAS:

- Programa.sas;
- Regressão.sas;
- Programa-reg01.sas.

Exemplo de nomes de arquivos de dados:

- Dados.dat;
- Arquivo.dat;
- Arquivo_de_dados01.dat.

Ao se abrir o SAS no WINDOWS, automaticamente são geradas as três janelas: *Editor*, *Log* e *Output* (Figura 31).

Para salvar um arquivo de programa, clique no menu |File|Save| (Figura 32A) e escolha a opção salvar como tipo |SAS Files (*.sas)| (Figura 32B). Para salvar um arquivo de dados, clique no menu |File|Save| (Figura 32A) e escolha a opção salvar como tipo |DATA Files (*.dat)| (Figura 32C).

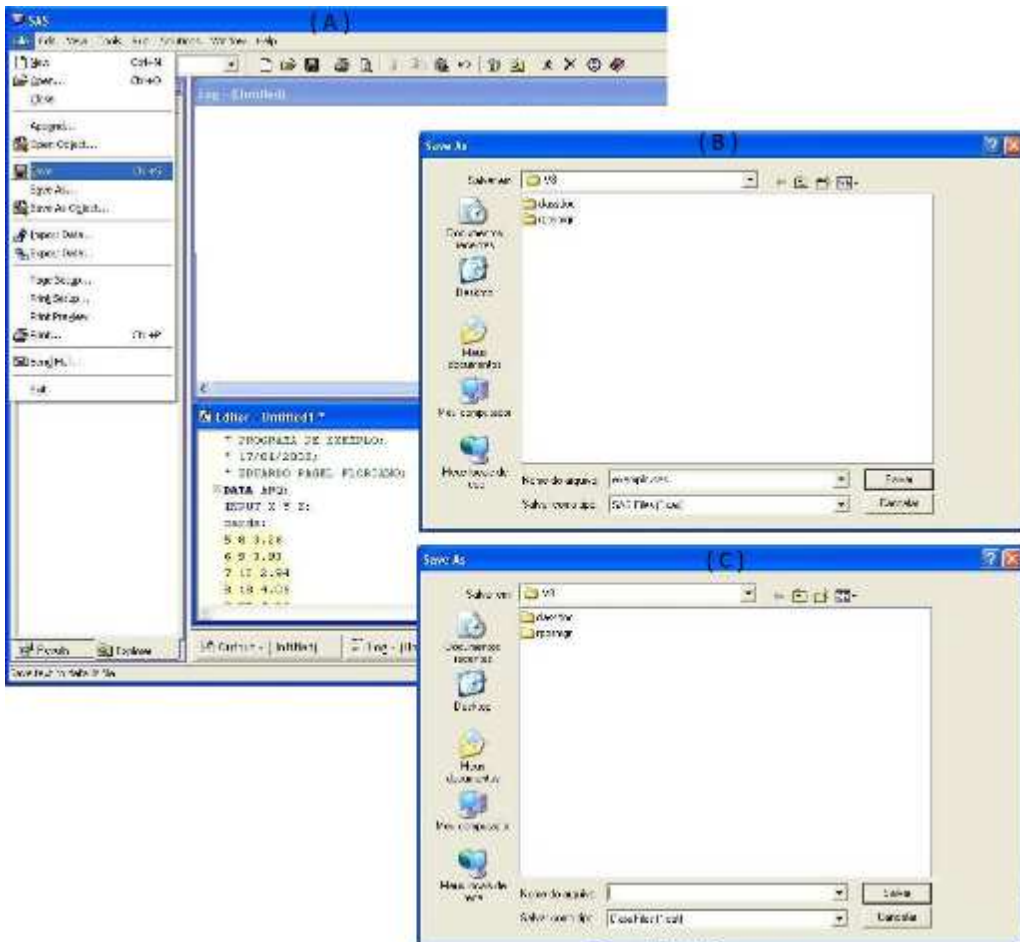


Figura 32: Salvando (2A) programas (2B) e dados (2C) em disco.

Depois de salvo, o programa pode ser executado utilizando o atalho na barra de atalhos ou o menu |Run|Submit|, como na Figura 33.



Figura 33: Execução de um programa através da barra de menus.

O Editor (Program Editor) e os relatórios podem ser acessados através do menu [View] (Figura 34).



Figura 34: **Menu View: permite acessar o Editor do SAS e os relatórios.**

Editor

Na janela do Editor pode-se escrever os programas ou dados e salvar em disco de forma semelhante a outros editores de texto. Os arquivos de programas devem ser salvos com nomes do tipo '*nomedoprograma.sas*' e arquivos de dados devem ser salvos com nomes do tipo '*arquivodedados.dat*'. O arquivo de programa '*exemplo.sas*' a seguir, contém 22 linhas de programação, sendo 3 de comentários que identificam o programa, 15 linhas de DATA STEP onde 11 são de dados, duas de PROC STEP, uma de execução e uma de finalização.

* PROGRAMA DE EXEMPLO (exemplo.sas);	1
* 17/01/2005;	2
* EDUARDO PAGEL FLORIANO;	3
DATA ARQ;	4
INPUT X Y Z;	5
cards;	6
5 8 3.26	7
6 9 3.93	8
7 12 2.94	9
8 18 4.06	10
9 25 4.35	11
10 33 4.11	12
11 42 4.64	13
12 57 4.29	14
13 70 4.50	15
14 85 4.55	16
15 101 4.97	17
;	18
PROC REG DATA=ARQ;	19
MODEL Z=X Y;	20
run;	21
QUIT EXIT;	22

Log.

Log é uma crônica ou crítica que o sistema faz, enquanto um programa está sendo executado, sobre o que foi realizado, relatando a execução e os possíveis erros.

Após a execução do programa, o SAS informa o que aconteceu durante a execução do programa na janela de Log. Os erros porventura existentes são relatados, localizados e identificados para que o programador possa corrigí-los.

No caso do programa 'exemplo.sas', não houve erros e o relatório de Log é o seguinte:

```

1  * PROGRAMA DE EXEMPLO;
2  * 17/01/2005;
3  * EDUARDO PAGEL FLORIANO;
4  DATA ARQ;
5  INPUT X Y Z;
6  cards;
NOTE: The data set WORK.ARQ has 11 observations and 3 variavels.
NOTE: DATA (instrução) used:
      real time      0.20 seconds
      cpu time       0.20 seconds
18  ;
19  PROC REG DATA=ARQ;
20  MODEL Z=X Y;
21  run;
NOTE: 11 observations read.
NOTE: 11 observations used in computations.
22  quit|exit;
NOTE: PROCEDURE REG used:
      real time      0.60 seconds
      cpu time       0.60 seconds

```

Output

O relatório de saída do programa 'exemplo.sas' é o seguinte:

```

          The SAS
          The REG Procedure
          Model: MODEL1
          Dependent Variavel: Z
          Analysis of Variance
          Sum of   Mean
Source      DF Squares  Square  F Value Pr > F
Model       2  2.50102  1.25051   9.37  0.0080
Error       8  1.06765  0.13346
Corrected Total 10  3.56867
  Root MSE    0.36532  R-Square  0.7008
  Dependent Mean 4.14545  Adj R-Sq  0.6260
  Coeff Var    8.81246
          Parameter Estimates
          Parameter  Standard
Variavel DF Estimate  Error  t Value Pr > |t|
Intercept 1  2.24246  0.84499  2.65  0.0291
X          1  0.22266  0.14120  1.58  0.1535
Y          1 -0.00774  0.01450 -0.53  0.6081

```

O SAS permite salvar este relatório no formato *rich text format* (rtf), que pode ser lido diretamente em qualquer editor de textos como o MS-Word, onde pode ser editado e modificado. Para tanto, basta clicar na aba da janela de *Output* para torná-la ativa, acessar o menu |File|, |Save as| e selecionar o formato rtf.

Existe um outro formato de saída mais elaborado que pode ser ativado através do menu |Tools|, |Opções|, |Preferences...|, na aba |Results|, marcando-se a caixa de seleção |Create HTML| (Figura 35).

O relatório HTML gerado pelo SAS apresenta os resultados no formato de tabela. As tabelas do relatório de saída podem ser copiadas e coladas em outros locais. O relatório HTML pode ser aberto no MS-Word e alterado, podendo depois ser salvo com a terminação '.doc' no formato do próprio MS-Word.



Figura 35: Selecionando relatórios no formato HTML.

3.3 PROCEDIMENTOS SAS (PROC STEP)

Os procedimentos do SAS são utilizados para processar os dados gerados no DATA STEP e abrangem muitas áreas do conhecimento.

Podem ser entendidos como rotinas de programação do sistema que realizam cálculos complexos, conforme a metodologia descrita nos manuais originais do SAS, que se baseiam em obras científicas consagradas.

Além dos procedimentos básicos, que são essenciais para se trabalhar com a linguagem, na área florestal os mais importantes são os relacionados com estatística (incluindo cálculos e produção de gráficos), programação linear e processamento de informações geográficas. Neste manual é dada ênfase aos procedimentos de estatística e de programação linear.

Os capítulos dedicados aos passos de programação de procedimentos abrangem:

- PROC ANOVA;
- PROC CHART;
- PROC CORR;
- PROC FREQ;
- PROC GLM;
- PROC MEANS;
- PROC MODEL;
- PROC PLOT;
- PROC PRINT;
- PROC REG;
- PROC UNIVARIATE.

Os procedimentos (PROC) básicos do SAS são utilizados para manipular arquivos, obter estatísticas simples uni e bivariadas, para formatar variáveis e emitir relatórios, entre outros.

Maiores detalhes sobre os procedimento básicos citados são encontrados nos manuais originais do SAS *disponíveis no site* <www.sas.com>.

Para aplicação nas ciências rurais e biológicas, considera-se essencial conhecer os procedimentos básicos relacionados a seguir:

- Procedimentos básicos estatísticos** – Geralmente não é viável medir todos os indivíduos de uma população. Então, procura-se medir alguns indivíduos que a representem. Uma coleção de valores medidos sobre uma parte dos indivíduos de uma população é chamada de amostra. O resultado de uma função matemática executada sobre valores de uma amostra é chamada de estimador (SAS Institute, 1999), enquanto que o resultado obtido através de

todos os indivíduos da população é chamado de parâmetro. É habitual representar estimadores por caracteres arábicos e parâmetros por letras gregas. Dentre todos os procedimentos estatísticos básicos do SAS há três que oferecem juntos todas as possibilidades de cálculos e relatórios dos demais, são o PROC UNIVARIATE, o PROC FREQ e o PROC CORR; além dos relatórios emitidos por eles, ainda apresentam a opção de emitir arquivos de saída com os quais se pode produzir todos os tipos de relatórios emitidos pelos demais procedimentos básicos estatísticos com auxílio do PROC PRINT. Dois procedimento que apresentam grande utilidade pela sua simplicidade são o PROC MEANS e o seu correlato PROC SUMMARY; ambos fazem os mesmos cálculos de forma semelhante, mas o primeiro apresenta relatórios mais elaborados e o segundo apresenta a possibilidade de gravar os resultados em um arquivo de saída sem emitir relatórios. Assim, este manual será restrito a esses seis procedimentos. Os demais procedimentos básicos estatísticos são: PROC TABULATE, PROC REPORT e PROC SQL.

- Procedimentos básicos de manipulação de arquivos** – Os principais são o PROC APPEND e o PROC SORT, descritos no Capítulo 6.
- Procedimento básico de formatação** – É o procedimento PROC FORMAT.
- Procedimentos de emissão de relatórios** – Os principais são o PROC PRINT, que imprime o conteúdo de arquivos em forma de lista, e o PROC TABULATE, que imprime o conteúdo de arquivos em forma de tabelas, além de calcular algumas estatísticas simples; o PROC FREQ pode ser utilizado para produzir tabelas semelhantes ao PROC TABULATE, assim, somente o primeiro foi descrito neste manual.

Neste volume são descritos os seguintes procedimentos básicos:

- PROC APPEND;
- PROC CORR;
- PROC FORMAT;
- PROC FREQ;
- PROC IMPORT;
- PROC MEANS;
- PROC PRINT;
- PROC SORT;
- PROC UNIVARIATE.

Entre os procedimentos estatísticos e de produção de gráficos, são abordados neste volume os seguintes:

- PROC ANOVA / GLM;
- PROC CHART / GCHART;
- PROC G3D;
- PROC MODEL;
- PROC PLOT / GPLOT;

❑ PROC REG.

Os procedimentos no SAS University podem ser diferentes dos descritos neste e nos próximos capítulos.

3.4 USANDO OS EXEMPLOS DE PROGRAMAS

Para usar os exemplos descritos neste manual sem alterá-los, é necessário criar uma pasta no diretório raiz (C:) do computador chamada de **EXEMPLOS**.

Para criar a pasta, abra o WINDOWS EXPLORER e clique em |Disco local (C:)| (Figura 36-A), depois clique em |Arquivo|Novo ►|Pasta| (Figura 36-B). Aparecerá uma nova pasta no diretório raiz; altere o nome padrão de **Nova pasta** para **EXEMPLOS**.



(A) Clique em |Disco local C:|

(B) Clique em |Arquivo|Novo ►|Pasta|

Figura 36: – Criando uma nova pasta no diretório raiz C:\.

Capítulo 4 : Passos de programação SAS

Os programas SAS são constituídos por dois tipos de passos de programação (STEPS): os passos de criação de arquivos de dados (DATA STEP) e os passos de execução de procedimentos (PROC STEP).

Este Capítulo é uma introdução ao DATA STEP e ao PROC STEP. O DATA STEP é descrito com detalhes no Capítulo 6 e os principais PROC STEPs são descritos nos capítulos subsequentes.

4.1 DATA STEP

O DATA STEP é o passo da programação SAS em que são criados ou modificados os **arquivos SAS** usados no programa. Pode haver tantos DATA STEP quantos necessários em um programa, que são constituídos por expressões e instruções formando frases (declarações) que iniciam pelo nome de um comando ou de uma variável e terminam por ponto-e-vírgula (;). Um DATA STEP sempre inicia por 'DATA' seguido do nome do arquivo SAS que será criado.

Um **Arquivo SAS** é um arquivo criado em um programa, com um nome iniciado por uma letra e com até 8 caracteres de comprimento, sendo padronizadamente armazenado na memória do computador até que o programa seja encerrado, opcionalmente podendo ser excluído ou modificado antes do encerramento do programa por declarações específicas escritas pelo programador, em qualquer parte do programa, após a criação do arquivo.

Exemplo de DATA STEP:

* Exemplo de data step;	1
DATA ARQ;	2
INPUT X Y @@;	3
Z=1.5+0.4*X-0.03*Y;	4
DATALINES;	5
5 8 6 9 7 12 8 18 9 25 10 33	6
11 42 12 57 13 70 14 85 15 101	7
;	8
run;	9

Neste exemplo é criado um arquivo SAS denominado de ARQ, são lidos 6 grupos de duas colunas de dados (X e Y) na primeira linha, 5 grupos na segunda linha e é

calculada uma terceira variável (Z), a partir dos valores de X e Y, para compor o conteúdo do arquivo ARQ, que resulta na seguinte tabela final:

Obs	X	Y	Z
1	5	8	3.26
2	6	9	3.63
3	7	12	3.94
4	8	18	4.16
5	9	25	4.35
6	10	33	4.51
7	11	42	4.64
8	12	57	4.59
9	13	70	4.60
10	14	85	4.55
11	15	101	4.47

4.2 PROC STEP

O PROC STEP é o passo da programação SAS em que são processados os dados de arquivos utilizando os procedimentos (PROCEDURES) do sistema. Os procedimentos são programas internos, cada um com sua sintaxe específica, que executam tarefas complexas como o desenho de gráficos e análise estatística. Pode haver quantos PROC STEPs quantos necessários em um programa. Compõem-se de comandos e operadores formando frases (declarações) que iniciam pelo nome de um comando ou de uma variável e terminam por ponto-e-vírgula (;).

Um PROC STEP sempre inicia por 'PROC' seguido do nome do procedimento que será executado.

Exemplo de PROC STEP:

```
* Exemplo de proc step;
Proc reg data=arq;
Model z=x y;
run;
```

O exemplo de PRO STEP acima calcula uma regressão com os dados do arquivo ARQ, onde a variável dependente é Z e as variáveis independentes são X e Y. O relatório de saída gerado pelo SAS em formato HTML, resultado do processamento do PROC STEP, com os dados do arquivo SAS gerado pelo exemplo de DATA STEP anterior, é apresentado a seguir no formato HTML de relatórios SAS:

The SAS

The REG Procedure

Model: MODEL1

Dependent Variavel: Z

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	2.07467	1.03734	Infy	<.0001
Error	8	0	0		
Corrected Total	10	2.07467			

Root MSE	0	R-Square	1.0000
Dependent Mean	4.24545	Adj R-Sq	1.0000
Coeff Var	0		

Parameter Estimates					
Variavel	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	1.50000	0	Infy	<.0001
X	1	0.40000	0	Infy	<.0001
Y	1	-0.03000	0	-Infy	<.0001

4.3 JCL (*JOB CONTROL LANGUAGE*)

Os programas SAS podem conter ainda, um passo inicial de instruções para integração do programa ao sistema operacional no qual está instalado, chamado de JCL. O JCL não é necessário no WINDOWS, sendo aqui referido porque podem ser encontrados programas na literatura que o incluem, embora seja possível executar programas SAS no modo de lotes (*Batch Mode*) no WINDOWS. O JCL é necessário geralmente com sistemas IBM de grande porte e UNIX.

Nos sistemas de grande porte (mainframes IBM, por exemplo) os programas são escritos na área de edição do sistema operacional, são executados através de

procedimentos próprios de cada ambiente operacional e incluem um JCL inicial que fornece informações ao sistema operacional sobre o programa a ser executado, além de linhas especiais de encerramento do programa.

Para converter programas SAS que possuem JCL, com a finalidade de executá-los em ambiente WINDOWS, geralmente é suficiente excluir o JCL, corrigir os endereços de arquivos externos referidos nos comandos INFILE do restante do programa, eliminar as linhas de encerramento do programa e incluir ao final do mesmo as linhas:

```
RUN;  
quit|exit;
```

A estrutura do JCL inclui as seguintes linhas de programação geralmente iniciadas por '//':

```
Cartão JOB  
Cartão EXEC SAS  
Cartões de identificação de arquivos de entrada e saída  
Cartão de encerramento do JCL
```

Exemplo de JCL:

```
//LOGUSUARIO JOB(R,BIB,USUARIO), 'endereço-nome',  
//          CLASS=Y,time=(5),MSGCLAS=R  
//sas      exec sas, time=5  
//nomearq1 dd dsn=enderecobib.nomearq2,disp=shr  
//sysin dd *
```

Em que: NOMEARQ1=nome do arquivo externo para chamada no programa; NOMEARQ2=nome do arquivo externo em disco.

Exemplo de linhas de encerramento de programa:

```
ENDsas;  
/*  
//
```

Capítulo 5 : Elementos da linguagem SAS

Programas na linguagem SAS são constituídos por passos de programação (DATA STEP e PROC STEP). Um passo de programa SAS, sempre inicia por uma das duas palavras: DATA ou PROC.

Cada passo de programa é constituído por declarações (frases) e cada declaração é constituída por expressões. Neste manual é feita distinção entre **declarações** (frases de programação) e **instruções** (comandos que executam ações em programas) que os manuais do SAS não distinguem e se referem a ambos como 'STATEMENTS'.

Uma declaração é uma frase completa que inicia por uma variável ou por uma instrução e sempre termina por um ponto e vírgula (;).

Uma expressão é formada por operandos e operadores escritos na forma de palavras de diferentes tipos e significados.

Operandos são constantes ou variáveis que podem ser numéricas ou de caracteres.

Operadores são símbolos que representam: uma comparação, um cálculo aritmético, ou uma operação lógica; uma função SAS; ou grupos de qualquer desses anteriores entre parênteses.

Os correspondentes em inglês dos elementos da linguagem SAS são:

- STEP – Passos de programação (DATA STEP E PROC STEP);
- STATEMENT – Declaração (frase) de programação;
- STATEMENT – Instrução de comando de uma ação;
- EXPRESSION – Expressão;
- OPERAND – Operando;
- OPERATOR – Operador;
- TOKEN | WORD – Símbolo ou palavra;
- CONSTANT – Constante;
- VARIABLE – Variável.

A forma geral dos passos de programação (DATA STEP e PROC STEP) estão descritos no Capítulo 4, funções são descritas em detalhe no Capítulo 9 e os elementos a seguir são definidos neste capítulo:

- PALAVRAS;
- OPERANDOS;
- OPERADORES;
- CONSTANTES;

- EXPRESSÕES;
- VARIÁVEIS;
- INSTRUÇÕES;
- DECLARAÇÕES;
- FUNÇÕES.

5.1 PALAVRAS

Uma palavra ou símbolo na linguagem SAS é uma coleção de caracteres, indivisíveis em unidades menores capazes de uso independente, que comunicam um significado ao SAS. Uma palavra pode conter um máximo de 32767 caracteres.

Uma palavra símbolo termina quando o SAS encontra um dos seguintes:

- O começo de um novo símbolo;
- Um espaço em branco depois de um nome ou um símbolo de número;
- A aspa de fim de um símbolo literal.

Cada palavra ou símbolo na linguagem SAS pertence a uma das quatro categorias:

- Nomes;
- Literais;
- Números;
- Caracteres especiais.

5.1.1 NOMES

Nomes de variáveis podem possuir:

- Letras (ABCDEFGHIJKLMNOPQRSTUVWXYZ);
- Algarismos (0123456789);
- Caractere de sublinhado (_).

Geralmente são limitados ao tamanho de 8 caracteres e devem iniciar por uma letra ou pelo caractere de sublinhado; não podem iniciar por um número.

Nomes de variáveis criadas pelo usuário não podem coincidir com palavras reservadas do SAS como nomes de variáveis internas, nomes de instruções ou de procedimentos e não podem ter acentos ou cedilha.

Exemplos de nomes permitidos:

- | | | |
|-----------------------------------|---------------------------------|-------------------------------------|
| <input type="checkbox"/> _DADOS | <input type="checkbox"/> ARV_03 | <input type="checkbox"/> yearcutoff |
| <input type="checkbox"/> _OBS | <input type="checkbox"/> ata | <input type="checkbox"/> year_99 |
| <input type="checkbox"/> VARIAV01 | <input type="checkbox"/> _new | <input type="checkbox"/> descending |
| <input type="checkbox"/> n_01 | <input type="checkbox"/> FREQ_ | <input type="checkbox"/> j7rk3 |

5.1.2 LITERAIS

Literais podem conter qualquer tipo de caractere, exceto caracteres de concatenação e delimitação das literais como o apóstrofo, barra vertical ou aspas (',| ou ").

Exemplos de literais:

- 'Brasil'
- "1990-91"
- 'Santa Maria'
- "It's only a literal."

O SAS não grava os delimitadores junto com o valor literal, exceto em casos especiais quando isso é determinado pelo usuário.

5.1.3 NÚMEROS

Números podem iniciar por um sinal de positivo (+) ou de negativo (-) e podem ter formato inteiro, decimal, notação científica (E-), notação hexadecimal, símbolo de valor perdido, formato de data e de data literal.

Exemplos de números:

- 5683
- 2.35
- 5
- 5.4E-1
- '24aug90'

5.1.4 CARACTERES ESPECIAIS

Caracteres especiais são todos os que fazem parte da linguagem SAS e não são utilizados para compor nomes, nem números.

A maioria das palavras (símbolos) compostas por caracteres especiais é constituída por um só caractere, mas algumas são formadas por dois caracteres.

Exemplo de palavras constituídas por caracteres especiais:

- ;
- /
- =
- '
- @
- +
- <=
- |

5.2 CONSTANTES

Uma constante SAS é um número, uma série de caracteres ou outra notação que indique um valor fixo (constante). Uma constante também é chamada de **valor literal**.

O SAS usa três tipos de constantes:

- Numéricas;
- Série de caracteres;
- Data, Horário e Data-Horário.

5.2.1 CONSTANTES NUMÉRICAS

Uma constante numérica é um número simples que é usado em uma frase de programa ou declaração SAS. Constantes numéricas podem usar ponto decimal, sinal de negativo, ou notação científica do tipo exponencial de 10.

Exemplos de exponenciais:

- $3,5 \times 10^{21} = \mathbf{3.5E21}$;
- $5 \times 10^{-3} = \mathbf{5E-3}$.

Constantes numéricas podem ser expressas como valores hexadecimais em declarações SAS. Uma constante hexadecimal inicia por um dígito numérico (usualmente zero), podendo ser seguido de um ou mais dígitos e sempre é finalizada pela letra X.

5.2.2 CONSTANTES DE CARACTERES

Uma constante de caracteres consiste de uma série de até 200 caracteres limitados por aspas simples ou duplas ('constanteliteral'). Se uma constante tiver de obrigatoriamente incluir uma aspa simples ('), ela deverá ser limitada por aspas duplas ("constante'literal"). Uma constante literal representando um valor perdido consiste de um caractere branco limitado por aspas(' ').

5.2.3 CONSTANTES DE DATA, HORÁRIO E DATA-HORÁRIO

Uma constante de data, horário, ou de data-horário é criada escrevendo-se a data, horário, ou data-horário entre aspas seguidas de, respectivamente, D para data, T para horário, ou DT para data-horário.

Veja o Capítulo 8: Formats e informats para maiores informações.

5.3 VARIÁVEIS

Variáveis são nomes de colunas utilizadas nos arquivos SAS e podem ser internas (geradas pelo sistema) ou criadas pelo usuário. Os nomes de variáveis internas são restritos; o usuário não pode criar uma nova variável com o nome de uma variável restrita do sistema. O conteúdo de uma variável interna pode ser utilizado pelo usuário como qualquer outra em cálculos ou operações relacionais.

As variáveis definidas pelo usuário podem ser números ou literais, seus nomes devem obedecer às regras para nomes e seus conteúdos devem seguir as regras para números e literais descritas anteriormente na seção 'PALAVRAS'.

5.4 OPERANDOS

Operandos são constantes ou variáveis que podem ser numéricas ou de caracteres.

Frequentemente são usados parênteses para compor grupos de operandos. Neste caso eles devem ser separados ou por espaços em branco ou por vírgulas.

Operandos são simples informações e não executam ações. O nome de uma variável colocado em uma declaração, por exemplo, não executa nenhuma ação, simplesmente diz ao SAS onde encontrar a informação.

As ações são executadas por instruções de comando ou por operadores.

Entre si e entre nomes de qualquer tipo, inclusive códigos mnemônicos, os operandos sempre devem ser separados por espaços em branco, mas não necessitam ser separados dos operadores escritos como símbolos.

5.5 OPERADORES

Os três principais tipos de operadores na linguagem SAS são os matemáticos, lógicos e relacionais. Cada operador tem uma prioridade de execução e, dependendo do grupo a que pertencem, operadores de mesmo nível de prioridade são executados em mesmo sentido.

Os operadores podem ser representados por símbolos ou por seus correspondentes mnemônicos em um programa, ambos executam as mesmas ações.

Os operadores matemáticos são semelhantes aos da linguagem matemática simples (Tabela 1).

TABELA 1 – Operadores matemáticos e sua prioridade de execução na linguagem SAS.

Grupo	Símbolo	Código Mnemônico	Descrição	Sentido	Prioridade	Exemplo
0	(Não há	Parêntese aberto	De dentro para fora	1	$X=K+Z^*(A-B)$
0)	Não há	Parêntese fechado	De dentro para fora	1	$X=K+Z^*(A-B)$
I	**	Não há	Potenciação	←	2	$Y=B^{**}X$
II	*	Não há	Multiplicação	→	4	$C=A*B$;
II	/	Não há	Divisão	→	4	$F=G/H$;
III	+	Não há	Adição	→	3	$C=A+B$;
III	-	Não há	Subtração	→	3	$F=G-H$;
VIII	=	Não há	Igualdade	←	5	$F=G-H$;

Os operadores lógicos são aqueles utilizados para formar declarações que implicam na tomada de decisão (Tabela 2).

TABELA 2 – Operadores lógicos na linguagem SAS.

Grupo	Símbolo	Código Mnemônico	Descrição	Sentido	Prioridade	Exemplo
I	^	NOT	Não lógico(*)	←	1	IF NOT Z THEN PUT X;
VII		OR	Ou lógico	→	3	IF X=1 OR Z=3 THEN DELETE;
VI	&	AND	E lógico	→	2	IF X=1 AND Z=1 THEN OUTPUT;

(*) Há três símbolos para NOT dependendo do sistema operacional: ^, ~ e ¬.

Operadores relacionais são os utilizados para comparações, principalmente do conteúdo de variáveis ou de literais (Tabela 3).

TABELA 3 – Operadores relacionais na linguagem SAS.

Grupo	Símbolo	Código Mnemônico	Descrição	Sentido	Prioridade	Exemplo
V	<	LT	Menor que	→	5	IF X LT Y THEN A=0;
V	<=	LE	Menor ou igual a	→	5	IF X LE Y THEN A=1;
V	=	EQ	Igual a	→	5	IF Y EQ (X+A) THEN DELETE;
V	^=	NE	Não igual a	→	5	IF X NE Z THEN OUTPUT;
V	>=	GE	Maior ou igual a	→	5	IF Y>=A THEN OUTPUT;
V	>	GT	Maior que	→	5	IF Z>A THEN OUTPUT;
V	Não há	IN	Igual a um da lista	→	5	IF ESTADO IN ('RS','SC','PR') THEN REGIAO='SUL';

Há outros operadores na linguagem SAS. Entre eles podem ser citados os prefixos matemáticos (positivo e negativo) utilizados para definir variáveis, os de relacionamento para definir qual o valor máximo ou mínimo entre duas variáveis e o de concatenação de literais que justapõe dois ou mais valores literais (ou o conteúdo de variáveis literais) para compor um só valor (Tabela 4).

TABELA 4 – Outros operadores na linguagem SAS.

Grupo	Símbolo	Código Mnemônico	Descrição	Sentido	Prioridade	Exemplo
I	+	Não há	Prefixo positivo	←	1	Y=+(A+B);
I	-	Não há	Prefixo negativo	←	1	Z=(A+B);
I	><	MIN	Mínimo	←	3	X=(A><B);
I	<>	MAX	Máximo	←	3	X=(A<>B);
IV		Não há	Concatenação	→	1	A='AGUA'; B='VIVA'; C=A ' ' B; Conteúdo de C: 'AGUA-VIVA'

5.6 EXPRESSÕES

Uma expressão SAS, geralmente, é uma sucessão de operandos e operadores formando um conjunto de operações que são executadas para produzir um valor resultante. Expressões em SAS são usadas para escrever declarações, para criar variáveis, nomear valores, calcular valores novos, transformar variáveis e executar processos condicionais. Expressões SAS podem resultar em valores numéricos, valores de caracteres, ou valores Booleanos.

Expressão simples – É uma expressão sem mais de um operador. Uma expressão simples pode consistir de uma única:

- Constante;
- Variável;
- Função.

Expressão combinada – É uma expressão que inclui vários operadores. Quando o SAS encontra uma expressão combinada, segue regras para determinar em que ordem deve executar cada parte da expressão.

Exemplos de expressões SAS:

Simples:

- 3;

Combinadas:

- idade<10 and peso>20;

- x;
- max;
- trim(last) || ', ' ||first;
- x=4+y.

5.7 INSTRUÇÕES

Uma instrução SAS é uma série de itens que inicia por uma palavra de comando que dá nome à instrução. Uma instrução SAS requer que o sistema execute uma ação ou forneça uma informação ao sistema.

Há três tipos principais de instruções SAS:

- As específicas do DATA STEP (passos de construção de arquivos);
- As específicas de cada PROC STEP (passos de execução de procedimentos);
- As globais que podem ser usadas em qualquer lugar de um programa de SAS.

Algumas declarações SAS específicas a cada procedimento SAS, são comuns a grupos de procedimentos afins e são executadas com sintaxe semelhante.

Instruções SAS globais podem ser colocadas fora dos passos de programa. Neste caso não serão executadas, ficarão disponíveis a partir do ponto de sua declaração no programa. Serão executadas quando um passo de programa necessitar delas, como as instruções de impressão. Por exemplo: se ao iniciar um programa for escrita uma instrução TITLE com um título qualquer, sempre que um passo de programa imprimir algo, aparecerá o título declarado.

As instruções SAS globais são classificadas nas seis categorias a seguir:

- ACESSO A DADOS - Associam nomes de referência com bibliotecas de dados SAS, catálogos SAS, arquivos externos, dispositivos de produção e arquivos remotos;
- AMBIENTE OPERACIONAL - Acessam o ambiente operacional diretamente;
- CONTROLE DE LOG - Alteram a aparência do LOG do SAS;
- CONTROLE DE PRODUÇÃO - Acrescentam títulos e notas de rodapé à produção do usuário; preparam a produção em uma variedade de formatos;
- CONTROLE DE PROGRAMA - Controlam o modo como o SAS processa os programas do usuário;
- EXIBIÇÃO DE JANELA - Exibe e personaliza janelas.

As diferentes instruções globais são descritas no manual disponível *on line*, no endereço: <support.sas.com/documentation/cdl/en/lrcon/68089/PDF/default/lrcon.pdf>.

5.8 DECLARAÇÕES

Uma declaração SAS é uma frase de programação completa, composta por uma série de itens que pode incluir palavras chave, nomes SAS, caracteres especiais, operandos e operadores, formando expressões que são reguladas por uma sintaxe rígida e específica de cada tipo de instrução ou comando envolvido na declaração. Uma declaração SAS requer que o sistema execute ação(ões), forneça informação(ões) ao sistema, ou é somente uma linha de comentário.

Geralmente uma declaração SAS inicia pelo nome de um passo (PROC ou DATA), pelo nome de uma instrução, ou pelo nome de uma variável.

Todas as declarações SAS terminam com um ponto-e-vírgula (;).

Os erros mais comuns em programas SAS ocorrem quando o programador esquece de encerrar uma declaração com ponto-e-vírgula. Então, o sistema entende que a continuação do programa é parte da declaração anterior e gera erros, obviamente.

5.9 FUNÇÕES

Funções são operações executadas pelo SAS que possuem um nome próprio. Uma função SAS, a partir do fornecimento de um argumento, retorna um valor. O SAS possui um elenco de funções em diferentes áreas, desde as mais comuns, como as funções aritméticas, até funções complexas como as econômicas e as de procura de parte de uma série de caracteres.

Por sua importância, as principais funções SAS foram descritas em mais detalhes no Capítulo 9 deste manual.

Capítulo 6 : Arquivos de dados (DATA STEP)

Os Arquivos de dados no SAS são constituídos por colunas (campos) e linhas (registros). As colunas também podem ser entendidas como as variáveis e as linhas como as observações contidas no arquivo.

Os arquivos, na programação SAS, podem ser criados a partir de:

- Arquivos externos pré-existentes;
- Arquivos criados anteriormente no mesmo programa;
- Dados fornecidos no próprio programa;
- Saída de um procedimento (PROC STEP).

Os passos da estrutura de um programa SAS, onde são criados arquivos, são chamados de DATA STEP. Num DATA STEP pode ser criado um ou mais arquivos e pode haver mais de um DATA STEP em um programa.

Deve-se evitar as palavras reservadas CON, NUL, PRN, LPT1-LPT9 e COM1-COM4 para nomes de arquivos.

O SAS utiliza dois tipos de dados: os arquivos de dados SAS típicos (DATA SET) ou dados virtuais resultantes de arquivos de lógica (DATA VIEW). Um arquivo de dados SAS (DATA SET) descreve e armazena os valores dos dados enquanto os arquivos de lógica SAS (DATA VIEW) não armazenam valores, ao invés disso, armazenam questões que criam dados virtuais que podem ser usados como se fossem simples arquivos de dados SAS; isso permite que sejam acessados dados armazenados em um ou mais arquivos de dados ao mesmo tempo e até arquivos de dados no formato de outros softwares que não o SAS. Os DATA VIEW permitem que sejam criados arquivos de lógica SAS sem usar espaço para armazenamento de dados, somente as questões lógicas são armazenadas e o que o usuário utiliza quando os acessa são os dados de resposta das questões programadas. Não se tratará dos DATA VIEW neste manual.

Um DATA STEP de um programa SAS, para criação de arquivos de dados SAS, sempre inicia por DATA e tem a seguinte forma geral:

```
DATA NOME DO ARQUIVO <OPÇÕES>;  
DECLARAÇÕES DE INICIALIZAÇÃO DE VARIÁVEIS E VETORES;  
DECLARAÇÕES PARA CÁLCULO E ATRIBUIÇÃO DE VALORES;  
DECLARAÇÕES DE ENTRADA DE DADOS;
```

6.1 SINTAXE

Há vários modelos de sintaxe para o DATA STEP, dependendo que tipo de arquivo que está sendo criado: DATA SET, DATA VIEW ou DATA PGM.

6.1.1 CRIAÇÃO DE ARQUIVOS DE DADOS DO TIPO DATA SET

```
DATA <NOMEDOARQUIVO-1 <(OPÇÕES-1)>> <NOMEDOARQUIVO--N
  <(OPÇÕES-N)>> </DEBUG>;
```

Ou

```
DATA _NULL_;
```

Nomedoarquivo – É o nome do arquivo de dados SAS a ser criado.

Opções – aparecem entre parênteses após um nome de arquivo. Especifica uma ou mais ações SAS a realizar enquanto escreve no arquivo externo. A sintaxe de opções do DATA SET tem a seguinte estrutura:

DEBUG - Permite depurar o programa, auxiliando na identificação de possíveis erros de dados ou de lógica.

NULL - Permite criar um arquivo de dados temporário que permanece na memória somente enquanto o DATA STEP está sendo executado.

6.1.2 OUTROS MODELOS DE SINTAXE PARA O DATA STEP

6.1.2.1 CRIAÇÃO DE ARQUIVOS DE LÓGICA DATA VIEW

```
DATA VIEW-NOME <NOMEDOARQUIVO-1 <(OPÇÕES-1)>>
  <...NOMEDOARQUIVO-N <(OPÇÕES-N)>> / VIEW=VIEW-NOME
  <(<SENHA-OPÇÃO>< SOURCE=FUNTE-OPÇÃO>)>;
```

ou

```
DATA VIEW=VIEW-NOME (<SENHA-OPÇÃO>);
  DESCRIBE;
  (OPÇÃO-1=VALUE-1<...OPÇÃO-N=VALUE-N)
```

6.1.2.2 CRIAÇÃO DE ARQUIVOS DE PROGRAMA DATA PGM

```
DATA NOMEDOARQUIVO / PGM=NOMEDOPROGRAMA <(<SENHA-
  OPÇÃO> <SOURCE=FUNTE-OPÇÃO>)>;
```

ou

```
DATA PGM=NOMEDOPROGRAMA (<SENHA-OPÇÃO>);
  <DESCRIBE;>
  <REDIRECTREDIRECT INPUT | OUTPUT NOMEANTIGO-1 = NOMENOVO-
  1 <... NOMEANTIGO-N = NOMENOVO-N>>;
  <EXECUTE;>
```

Exemplo - Criando um arquivo fornecendo dados no próprio programa:

```
* Exemplo - Criando um arquivo fornecendo dados no próprio
  programa;
data arq1;
  input arvore diametro altura;
  areabas1=3.1416*(diametro/100)**2/4;
  datalines;
01 10.5 11.3
02 15.0 14.1
03 12.4 11.9
```

```

;
* valores das linhas de dados nao podem ser recuados neste caso;
run;

```

10
11
12

Na linha 1 é identificado o nome do programa com uma linha de comentário.

O início do passo de dados na linha 2, segunda frase ou declaração do programa do “Exemplo - Criando um arquivo fornecendo dados no próprio programa” cria o arquivo chamado ‘ARQ1’:

```
data arq1;
```

Instrui a entrada de dados em três colunas chamadas de ‘ARVORE’, ‘DIAMETRO’ e ‘ALTURA’ através da declaração de INPUT na linha 3:

```
input arvore diametro altura;
```

Atribui o valor da área basal ‘AREABASL’ na linha 4, através da fórmula:

```
areabasl=3.1416*(diametro/100)**2/4;
```

Declara o início da entrada de dados com a declaração DATALINES linha 5:

```
datalines;
```

Dá entrada de três linhas de dados (linhas 6, 7 e 8):

```
01 10.5 11.3
02 15.0 14.1
03 12.4 11.9
```

Encerra a leitura de dados com um último ponto-e-vírgula na linha 9:

```
;
```

A linha 10 apresenta um comentário e a linha 11 instrui a execução do programa (RUN).

A partir da criação do arquivo ARQ1, este pode ser usado para a criação de outros arquivos no mesmo programa, ou pode ser chamado em um passo de procedimento.

O “Exemplo - criando um arquivo a partir de dados de um arquivo externo” cria um arquivo a partir de um arquivo externo.

Exemplo - criando um arquivo a partir de dados de um arquivo externo:

```

* Exemplo - criando um arquivo a partir de dados de um
  arquivo externo;
data arq1;
infile 'C:\EXEMPLOS\DADOSDH.dat';
input arvore diametro altura;
areabasl=3.1416*(diametro/100)**2/4;
run;

```

1
2
3
4
5
6

O passo de programação do “Exemplo - criando um arquivo a partir de dados de um arquivo externo” cria o arquivo chamado ‘ARQ1’ na linha 2:

```
data arq1;
```

Na linha 3, chama o arquivo externo ‘DADOSDH.DAT’ localizado no endereço ‘C:\EXEMPLOS\ ENDERECO\’, através da declaração de INFILE:

```
infile 'C:\EXEMPLOS\arqexterno.dat';
```

Instrui a entrada de dados em três colunas chamadas de 'ARVORE', 'DIAMETRO' e 'ALTURA' através da declaração de INPUT na linha 4:

```
input arvore diametro altura;
```

Atribui o valor da área basal 'AREABASL' na linha 5, através da fórmula:

```
areabasl=3.1416*(diametro/100)**2/4;
```

Observe que a instrução de atribuição de valores a partir de dados lidos no arquivo, em ambos os exemplos, deve ser realizada após a declaração de INPUT. Antes do INPUT podem ser incluídas declarações de vetores (ARRAY), de RETAIN, FORMAT, laços (DO;END;), atribuição de valores que não são lidos pelo INPUT, entre outras, mas não a atribuição de valores e cálculos baseados nos valores lidos pelo INPUT.

6.2 FORMATOS DE ENTRADA E SAÍDA (INPUT/PUT)

As instruções INPUT e PUT aceitam qualquer tipo de estrutura de dados em colunas e linhas. Pode-se instruir a leitura ou gravação de cada linha separadamente, de cada coluna e de diferentes maneiras. As principais são apresentadas em exemplos a seguir, iniciando com o "Exemplo – Entrada de dados em colunas sequenciais sem limites definidos, em linhas de dados sequenciais".

<pre>* Exemplo – Entrada de dados em colunas sequenciais sem limites definidos, em linhas de dados sequenciais; Data a; INPUT coluna1 coluna2 coluna3; datalines; 10.7 45.003 7. 11.0 34 71.03 10 25 93.8 14 50.3 57.8 ; PROC PRINT; run;</pre>	<pre>1 2 3 4 5 6 7 8 9 10 11</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------

Os dados devem ser colocados em duas colunas separadas por um espaço em branco. As linhas de dados são colocadas uma após a outra e o programa lê uma de cada vez. O comando DATALINES é sinônimo de CARDS na linguagem SAS, podendo-se usar um ou outro.

No Exemplo, o programa lê uma tabela de dados com três colunas chamadas de 'COLUNA1', 'COLUNA2' e 'COLUNA3' gerando um arquivo de dados SAS com o seguinte conteúdo:

```
10.7 45.003 7.00
11.0 34.000 71.03
10.0 25.000 93.80
```

■ 14.0 50.300 57.80

Após, o programa imprime o relatório a seguir:

<u>_OBS_</u>	<u>COLUNA1</u>	<u>COLUNA2</u>	<u>COLUNA3</u>
1	10.7	45.003	7.00
2	11.0	34.000	71.03
3	10.0	25.000	93.80
4	14.0	50.300	57.80

A coluna _OBS_ é inserida automaticamente no relatório SAS, contendo o número de ordem sequencial de cada observação do arquivo. Para que não seja impressa é necessário utilizar a opção NOOBS, como no exemplo abaixo:

■ PROC PRINT NOOBS;

O exemplo a seguir mostra como delimitar o número de caracteres das colunas de dados.

```
* Exemplo – Entrada de dados em colunas delimitadas, por
  grupos de linhas de dados;
data b;
input #1 @3 a 4-8 b 9-12 #2 @6 c d;
cards;
abc002009177
2.034983 9
knx009130918
40000 7 4
zjh137105937
0.330010.71000 14.8
;
run;
```

```
1
2
3
4
5
6
7
8
9
10
11
12
```

A instrução CARDS é equivalente à instrução DATALINES.

O caractere '#' seguido de um **número inteiro**, numa declaração de INPUT, indica o número de ordem de um grupo de linhas que deve ser lido.

A parte de programa do “Exemplo – Entrada de dados em colunas delimitadas, por grupos de linhas de dados” lê grupos de duas linhas, sendo a primeira linha a de número 1 e a segunda a de número 2. Inicialmente, no primeiro grupo de linhas, lê os cinco dígitos (4° até 8°) da primeira linha e os atribui à variável ‘A’ resultando em A=00200, lê os quatro dígitos de 9° até 12° da linha 1 e os atribui à variável ‘B’ resultando em B=9177; lê, a partir do 6° dígito da linha 2, até encontrar um caractere ‘branco’ e os atribui à variável ‘C’ resultando em C=983.00; em seguida lê os caracteres seguintes da linha 2 após o caráter ‘branco’ que segue a variável ‘C’ e os atribui à ‘D’, resultando em D=9.0. Depois é lido o segundo grupo e assim por diante até que o caractere ‘ponto-e-vírgula’ indique o final dos dados. O resultado da leitura é a seguinte tabela de dados:

A	B	C	D
200	9177	983.00	9.0
913	918	7.00	4.0
13710	5937	10.71	14.8

O “Exemplo– Entrada de dados em grupos de colunas sequenciais sem limites definidos, em linhas de dados sequenciais” lê grupos de dados.

* Exemplo– Entrada de dados em grupos de colunas sequenciais	1
sem limites definidos, em linhas de dados sequenciais;	2
data c;	3
INPUT coluna1 coluna2 coluna3 @@;	4
datalines;	5
10.7 45.003 7. 11.7 40.0 8.3	6
11.0 34 71.03 11.0 35 70.1	7
10 25 93.8 10 27 99.8	8
14 50.3 57.8	9
;	10
run;	

O INPUT do “Exemplo– Entrada de dados em grupos de colunas sequenciais sem limites definidos, em linhas de dados sequenciais” lê duas sequências de colunas. Poderiam ser incluídas mais sequências de colunas sem alterar o programa. São os caracteres ‘@@’ que indicam que os dados estão em grupos. Os três primeiros valores de cada linha são pertencentes à primeira sequência de três colunas e os últimos três valores de cada linha pertencem à segunda sequência, sendo cada valor separado do seguinte por um único caractere ‘branco’. O programa lê linha por linha, incluindo os dados da primeira sequência e, em seguida, os da segunda sequência da mesma linha de dados, resultando na tabela de dados a seguir:

COLUNA1	COLUNA2	COLUNA3
10.7	45.003	7.00
11.7	40.000	8.30
11.0	34.000	71.03
11.0	35.000	70.10
10.0	25.000	93.80
10.0	27.000	99.80
14.0	50.300	57.80

6.3 ENTRADA DE DADOS

6.3.1 IMPORTANDO DO EXCEL PARA O SAS

Para importar dados do Excel para o SAS, siga os passos:

- Ative o |EXPLORER| do SAS
- Ative o menu |FILE|
- Selecione |IMPORT DATA|
- Escolha |MS Excel 97 ou 2000|
- Onde o Arquivo está localizado |BROUSE| |C:\EXEMPLOS|
- Library: Curso (nome da pasta)
- Member: BlocosCasualQL (Nome do arquivo)
- Finalizar
- Abra e feche o arquivo para verificação.

6.3.2 USANDO DADOS DE UMA PLANILHA SAS

```
Libname Curso 'C:\EXEMPLOS';  
Data a;  
Set Curso.BlocosCasualQL;  
Proc Print;  
Run;  
Quit;
```

6.3.3 2.3.3 INCLUINDO OS DADOS NO PROGRAMA

```
Data a;  
Input Bloco Trat DAP Altura;  
Cards;  
1 1 10 12  
1 2 15 18  
1 3 14 16  
1 4 11 12  
1 5 10 11  
2 1 9 10  
2 2 13 15  
2 3 12 13  
2 4 10 11  
2 5 9 9  
3 1 11 14  
3 2 16 18  
3 3 12 13  
3 4 10 10  
3 5 9 10  
4 1 11 13  
4 2 16 19  
4 3 15 17  
4 4 13 14  
4 5 12 12  
;  
Proc Print;  
Run;  
Quit;
```

6.3.4 CRIANDO UM ARQUIVO A PARTIR DE DADOS DE UM ARQUIVO EXTERNO:

```
EXEMPLO:  
DATA ARQ1;  
INFILE 'C:\EXEMPLOS\DADOS1.DAT' expandtabs;  
Input Bloco Trat DAP Altura;  
AREABASAL=3.1416*(DAP/100)**2/4;  
RUN;  
QUIT;
```

6.3.5 IMPORTANDO UMA PLANILHA DO EXCEL NO PROGRAMA

```
PROC IMPORT DATAFILE='C:\EXEMPLOS\BlocosCasual.xls' OUT=Dados  
DBMS=EXCEL97 REPLACE;  
SHEET=BlocosCasualQL;
```

6.3.6 EXPORTANDO UMA PLANILHA PARA O EXCEL

```
PROC EXPORT DATA=Dados OUTFILE="C:\exemplos\Dados."  
DBMS=EXCEL97 REPLACE;
```

6.3.7 EXEMPLO DE IMPORTAÇÃO E EXPORTAÇÃO DE DADOS:

```
*Exemplo - importação e exportação de dados de e para o EXCEL;  
*Importando uma planilha do excel;  
PROC IMPORT  
/* VARIÁVEIS: Bloco Trat DAP Altura */  
DATAFILE='C:\EXEMPLOS\BlocosCasual.xls'  
OUT=Dados DBMS=EXCEL97 REPLACE;  
SHEET=BlocosCasualQL;  
DATA Dados;  
SET Dados;  
Volume=0.5*Altura*3.1416*DAP**2/40000;  
*Exportando uma planilha para o excel;  
PROC EXPORT DATA=Dados  
OUTFILE="C:\EXEMPLOS\Dados."  
DBMS=EXCEL97 REPLACE;  
RUN;  
QUIT;
```

6.4 PROCEDIMENTOS ESSENCIAIS COM O DATA STEP

Dois procedimentos associados à criação de arquivos são essenciais para trabalhar com arquivos de dados. O primeiro serve para adicionar dados de um arquivo SAS em outro (PROC APPEND) e o segundo para ordenar os dados (PROC SORT).

6.4.1 PROC APPEND

PROC APPEND adiciona observações de um arquivo SAS ao fim de outro arquivo SAS, sem processar as observações, sendo mais rápido e eficiente que o uso do DATA

STEP. O arquivo SAS original é chamado de BASE, ao qual serão adicionados os dados de outro arquivo SAS.

Sintaxe

```
PROC APPEND BASE=arquivobase DATA=arquivoadicionado <FORCE>  
<APPENDVER=V6>;
```

BASE=nomedoarquivobase – Informa o nome do arquivo BASE que deverá receber adição dos dados do outro arquivo SAS.

DATA=nomedoarquivoadicionado – Informa o nome do arquivo que será adicionado ao final do arquivo BASE.

APPENDVER=V6 – Usando a Versão 6 do SAS, deve ser especificada esta opção.

FORCE – Força a adição de dados mesmo quando o arquivo que está sendo adicionado contém variáveis que não existem no arquivo BASE.

O “Exemplo – Adicionando os dados de um arquivo SAS em outro” demonstra como adicionar dados de um arquivo a outro.

<pre>* Exemplo – Adicionando os dados de um arquivo SAS em outro; Data arq1; input parcela especie \$ dap; cards; 01 angico 35.5 01 aroeira 22.0 01 timbauva 40.5 ; data arq2; input parcela especie \$ dap; cards; 02 vassourao 19.0 02 cabreuva 37.0 ; Proc Append Base=arq1 Data=arq2; run;</pre>	<pre>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16</pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------

No “Exemplo – Adicionando os dados de um arquivo SAS em outro” são adicionados os dados do arquivo SAS denominado de ARQ2 ao arquivo SAS denominado de ARQ1. Ao final do procedimento, o arquivo ARQ2 permanece inalterado e o arquivo ARQ1 passa a ter o seguinte conteúdo:

PARCELA	ESPECIE	DAP
01	ANGICO	35.5
01	AROEIRA	22.0
01	TIMBAUVA	40.5
02	VASSOURAO	19.0
02	CABREUVA	37.0

6.4.2 PROC SORT

O procedimento SORT ordena as observações existentes em um arquivo SAS por uma ou mais variáveis de forma crescente ou decrescente, através de uma instrução BY. Pode simplesmente ser reordenado o arquivo original que passa a apresentar o ordenamento de dados requerido, ou pode ser criado um novo arquivo com os dados reordenados, permanecendo o arquivo original sem alterações. O procedimento SORT não produz impressão por si só.

Sintaxe

```
PROC SORT <data=ARQUIVODEENTRADA>  
  <OUT=ARQUIVODESAIDA> <OPÇÕES>;  
  BY <DESCENDING> VARIÁVEL-1 ...<DESCENDING> VARIÁVEL-N;
```

DATA=ArquivoDeEntrada – Especifica o arquivo de dados de entrada. Se esta opção não for utilizada, o último arquivo utilizado ou criado pelo programa será utilizado no procedimento.

OUT=ArquivoDeSaída – Esta opção especifica o arquivo de dados de saída. Se o arquivo de saída não existe, ele é criado por esta opção. Por padrão sem a opção OUT, o procedimento sobrepõe o arquivo existente com o novo, na nova ordem.

NODUPKEY – Procura e elimina observações com valores BY duplicados. Se esta opção for especificada, todos os valores BY de cada observação são comparados com a observação predecessora. Se uma observação exatamente igual é encontrada, ela é eliminada do arquivo final resultante do procedimento.

NODUPRECS – Procura e elimina observações duplicadas. Todos os valores das variáveis de cada observação são comparados com a observação prévia se esta opção for especificada. Se um valor exato é encontrado, a observação é eliminada do arquivo final resultante do procedimento. NODUPRECS confere só observações sucessivas, assim, algumas observações duplicadas não sucessivas podem permanecer arquivo final.

BY – A instrução BY é obrigatória e especifica as variáveis pelas quais o arquivo deve ser ordenado. O padrão é a ordem ascendente. Se for necessário usar ordem descendente, é necessário especificar através da opção DESCENDING.

VARIÁVEL – Podem ser relacionadas quantas variáveis forem necessárias (1 até N) para ordenar o arquivo, uma após a outra, na ordem de prioridade da esquerda para a direita.

DESCENDING – Quando for necessário ordenar de forma descendente, o termo DESCENDING deve ser escrito antes de cada variável que deve ser ordenada desta forma. Se uma variável da lista de variáveis que segue a instrução BY não for precedida pela opção DESCENDING, automaticamente será ordenada ascendentemente.

Para ordenar o dados de um arquivo, proceda como no Exemplo – Ordenando os dados de um arquivo SAS.

Utilizando os dados do arquivo de saída ARQ1 do Exemplo anterior e reordenando o arquivo pelas variáveis PARCELA e ESPECIE, tem-se o seguinte:

```
* Exemplo – Ordenando os dados de um arquivo SAS;          1
Proc sort data=arq1;                                       2
BY PARCELA ESPECIE;                                       3
Run;                                                         4
```

Como resultado o conteúdo do arquivo ARQ1 passa ser o da tabela abaixo, onde se observa que as observações 4 e 5 trocaram de posição com o re-ordenamento:

PARCELA	ESPECIE	DAP
01	ANGICO	35.5
01	AROEIRA	22.0
01	TIMBAUVA	40.5
02	CABREUVA	37.0
02	VASSOURAO	19.0

6.5 INSTRUÇÕES ESSENCIAIS DO DATA STEP

- ARRAY – Identificação de vetores;
- BY – Ordenação de variáveis;
- CARDS – Chamada de cartões de dados;
- DATALINES – Chamada de linhas de dados;
- DELETE – Instrução de exclusão de registro;
- DO END – Repetição em laço;
- DROP – Elimina variáveis de um arquivo SAS;
- FILE – Saída de arquivo externo;
- FILENAME – Identificação de arquivo externo de saída;
- IF THEN – Execução condicional.
- INFILE – Chamada de arquivo externo;
- INPUT – Chamada e formatação de dados de entrada;
- KEEP – Seleciona variáveis a permanecer num arquivo SAS;
- MERGE – Intercalação de arquivos;
- OUTPUT – Saída de dados para arquivo interno;
- PUT – Formatação de dados de saída;
- RETAIN – Retém o valor de uma variável até a próxima atribuição de valor para a mesma;
- SET – Chamada de arquivo interno;
- SUM – Efetua somatórios;
- UPDATE – Efetua atualização de dados em arquivos SAS;
- VAR – Identificação de variáveis a utilizar;
- WHERE – Impõe condições para execução de declarações.

6.5.1 ARRAY (INSTRUÇÃO)

ARRAY é um vetor de dados que pode ser de elementos literais ou números.

Um ARRAY deve ser escrito no início do passo em que será utilizado, onde se define a sua dimensão máxima e os valores iniciais. É composto por um nome seguido de um número de ordem (subscrito) entre duas chaves, assumindo o seguinte formato:

■ **NOMEDDARRAY{SUBSCRITO}<\$>**

Sintaxe

■ **ARRAY NOME{N}<\$> <COMPRIMENTO> <ELEMENTOS>
<(VALORESINICIAIS)>;
NOME{N};**

Nome – É o nome do vetor; não pode ser o mesmo de uma variável de uma janela; entretanto, variáveis de uma janela podem ser elementos de um vetor.

{n} – É a dimensão do vetor (número de elementos); se em lugar de um número, for colocado um asterisco (*), a dimensão será determinada pelo número total de elementos atribuídos ao vetor durante o processamento, ou pelo número inicial de elementos; são permitidos vetores multidimensionais.

\$ – Indica que os elementos do vetor são do tipo caractere.

Comprimento – É o comprimento máximo de elementos no vetor. Vetores de caracteres não podem ter comprimento maior que 200. O comprimento é ignorado em vetores numéricos.

Elementos – São as variáveis que constituem o vetor.

Valores iniciais – São os valores para inicializar alguns ou todos os elementos do vetor. Os valores devem ser separados por vírgula ou caracteres brancos. Como padrão todos os elementos são inicializados com valores perdidos.

Para criar vetores, siga o “Exemplo – Criando vetores (ARRAY)”.

* Exemplo – Criando vetores (ARRAY);	1
DATA arq;	2
ARRAY x{10} x1-x10 (0,0,0,0,0,0,0,0,0,0);	3
INPUT x1-x10;	4
cards;	5
14 15 12 17 19 11 17 19 13 16	6
13 19 17 13 11 14 13 15 10 11	7
;	8
run;	9

No “Exemplo – Criando vetores (ARRAY)” é criado o arquivo SAS chamado de ARQ e com a declaração ‘ARRAY X{10} X1-X10 (0,0,0,0,0,0,0,0,0,0)’ são criados 10 vetores X com 2 elementos, de X1 até X10, ou seja, X1, X2, X3, X4, X5, X6, X7, X8, X9 e X10, com todos os elementos de valor inicial zero (0); é instruída a entrada de dados em colunas sequenciais de X1 até X10 e dada a entrada de dois cartões de dados (linhas) contendo dez valores cada um.

O resultado são dez vetores com o seguinte conteúdo:

X1{1}=14, X1{2}=13;

X2{1}=15, X2{2}=19;
 X3{1}=12, X3{2}=17;
 X4{1}=17, X4{2}=13;
 X5{1}=19, X5{2}=11;
 X6{1}=11, X6{2}=14;
 X7{1}=17, X7{2}=13;
 X8{1}=19, X8{2}=15;
 X9{1}=13, X9{2}=10;
 X10{1}=16, X10{2}=11.

O conteúdo do arquivo SAS criado com o nome de ARQ é um vetor multidimensional, ou tabela, com 11 colunas e duas linhas, em que a primeira coluna contém o número de cada linha de dados que foi lida e as demais colunas contém os 10 vetores, como segue:

X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
14	15	12	17	19	11	17	19	13	16
13	19	17	13	11	14	13	15	10	11

O conjunto de vetores, neste caso, é comparável a uma matriz em que cada elemento pode ser referenciado individualmente, o que facilita a programação em alguns casos.

6.5.2 BY (INSTRUÇÃO)

A instrução BY é usada para controlar uma operação com SET, MERGE, MODIFY ou UPDATE em um DATA STEP, impondo uma condição de ordenação ou de seleção de dados.

Sintaxe

**BY <DESCENDING> VARIÁVEL-1 ...<<DESCENDING> VARIÁVEL-N>
<NOTSORTED>;**

Variável – Especifica variáveis em sequência hierárquica da esquerda para a direita que deve ser usada para ordenar ou controlar grupos. Variáveis em uma declaração de BY são chamadas de variáveis BY. Por padrão o SAS ordena as variáveis BY ascendentemente. O arquivo em uso deve ter sido previamente ordenado em um procedimento PROC SORT, ou deve-se usar a opção NOTSORTED.

DESCENDING – Quando esta opção é escrita antes de uma variável BY, indica que a ordem da variável é descendente.

NOTSORTED – Indica que o arquivo em uso não foi ordenado previamente em ordem alfabética e pode ter sido ordenado por outro critério como o cronológico.

6.5.3 CARDS (INSTRUÇÃO)

A instrução CARDS indica que a próxima linha é uma linha de dados em formato de acordo com a declaração de INPUT antecessora.

Sintaxe

■ **CARDS;**

CARDS – A instrução deve seguir uma declaração de INPUT anterior. Não há argumentos para esta instrução.

6.5.4 DATALINES (INSTRUÇÃO)

A instrução DATALINES é equivalente à instrução CARDS e indica que a próxima linha é uma linha de dados em formato de acordo com a declaração de INPUT antecessora.

Sintaxe

■ **datalines;**

DATALINES – A instrução deve seguir uma declaração de INPUT anterior. Não há argumentos para esta instrução.

6.5.5 DELETE (INSTRUÇÃO)

Elimina a linha de dados corrente do arquivo em uso.

Sintaxe

■ **delete;**

DELETE – Elimina o registro atual de dados do arquivo em uso. Não há argumentos para esta instrução.

6.5.6 DO/END (INSTRUÇÃO)

É a instrução utilizada para controle de laços de programação, semelhante às de outras linguagens como FORTRAN e BASIC.

Informa que as declarações entre DO e END devem ser executadas em grupo até que uma variável contadora de controle atinja seu o limite máximo.

O SAS permite três formas de DO:

- DO simples;
- DO UNTIL;
- DO WHILE.

Sintaxe

6.5.6.1.1 DO

```
DO VARIÁVEL= INICIAL TO FINAL <BY INCREMENTO>;  
  DECLARAÇÕES;  
END;
```

6.5.6.1.2 DO UNTIL

```
DO variavel=INICIAL TO FINAL <BY INCREMENTO> UNTIL(EXPRESSÃO);  
  DECLARAÇÕES;  
END;  
  Ou  
DO UNTIL(EXPRESSÃO);  
  DECLARAÇÕES;  
END;
```

6.5.6.1.3 DO WHILE

```
DO variavel= INICIAL TO FINAL <BY INCREMENTO> WHILE(EXPRESSÃO);  
  DECLARAÇÕES;  
END;  
  Ou  
DO WHILE(EXPRESSÃO);  
  DECLARAÇÕES;  
END;
```

DO – Inicia um laço de declarações a serem executadas em grupo.

VARIÁVEL – Nome da variável de controle do laço de declarações.

Inicial – Valor inicial da variável de controle.

Final – Valor final da variável de controle.

BY – Indica que há um valor a incrementar a cada retorno que segue a palavra BY.

Incremento – É o valor a incrementar na variável a cada passagem.

Declarações – São as declarações a executar em cada passagem.

END – Indica que se o último valor atribuído à variável de controle foi igual ao valor final, o laço deve ser encerrado e que devem ser executadas as declarações escritas após a instrução END. Enquanto o valor final não for atingido, o END faz com que o programa volte para executar todo o grupo de declarações a partir do DO.

UNTIL – Significa **até que**, ou seja, executa o grupo de declarações até que o valor da expressão entre parênteses seja encontrado.

WHILE – Significa **enquanto**, ou seja, executa o grupo de declarações enquanto o valor da expressão entre parênteses for verdadeiro.

Expressão – É qualquer expressão SAS válida.

6.5.7 DROP (INSTRUÇÃO)

Especifica os nomes das variáveis que devem ser eliminadas do arquivo SAS.

Sintaxe

■ **(DROP=LISTADEVARIABLES);**

6.5.8 FILE (INSTRUÇÃO)

Especifica o nome do arquivo SAS a ser gravado em disco conforme uma instrução FILENAME antecessora e no formato de uma declaração de PUT posterior. Deve-se evitar as palavras reservadas CON, NUL, PRN, LPT1 - LPT9 e COM1 para nomes de arquivos.

Sintaxe

■ **FILE NOMEDOARQUIVO <OPÇÕES> <HOST-OPÇÕES>;**

Nomedoarquivo – Nome do arquivo SAS a ser gravado externamente em disco;

Opções e host-opções – Há várias opções para execução desta instrução, mas não são consideradas essenciais.

A gravação de arquivos externos permanentes pode ser feita como no “Exemplo – Gravação em disco de arquivos externos permanentes”.

<pre> * Exemplo – Gravação em disco de arquivos externos permanentes; data arqsas1; input x y z; cards; 1 2 5 2 6 8 3 9 11 4 12 18 ; DATA _NULL_; filename arqsas2 "C:\EXEMPLOS\arqexterno.dat"; SET arqsas1; file arqsas2; put x y x; run; </pre>	<pre> 1 2 3 4 5 6 7 8 8 9 10 11 12 13 14 </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------

As declarações do programa do Exemplo 4.9 realizam o seguinte:

Declaração 1 (DATA ARQSAS1;) – Cria o arquivo SAS chamado de ARQSAS1;

Declaração 2 (INPUT X Y Z;) – Instrui o formato de entrada de três colunas X Y e Z;

Declaração 3 (CARDS;) – Instrui o início das linhas de dados;

Linhas 4 a 7 (valores) – Entrada de 4 linhas de dados;

Linha 8 (;) – Finaliza a entrada de dados;

Declaração 4 (DATA _NULL_;) – Cria o arquivo temporário _NULL_;

Declaração 5 (FILENAME ARQSAS2 "C:\EXEMPLOS\ARQEXTERNO.DAT;") – Instrui o programa para gravar o arquivo SAS chamado de ARQSAS2 no endereço C:\EXEMPLOS\ com o nome de ARQEXTERNO.DAT;

Declaração 6 (SET ARQSAS1;) – Chama o arquivo SAS ARQSAS1 para constituir o arquivo temporário;

Declaração 7 (FILE ARQSAS2;) – Cria o arquivo ARQSAS2 para ser gravado externamente conforme o formato da declaração de PUT a seguir;

Declaração 8 (PUT X Y X;) – Formata os dados para gravação no arquivo externo;

Declaração 9 (RUN;) – Executa o programa.

Como resultado é criado o arquivo de dados externo ARQEXTERNO.DAT com o seguinte conteúdo:

```
1 2 5
2 6 8
3 9 11
4 12 18
```

O arquivo ARQEXTERNO.DAT pode ser acessado por outros programas através de uma declaração INFILE de um DATA STEP.

6.5.9 FILENAME (INSTRUÇÃO)

Associa o nome de um arquivo SAS, referenciado em uma declaração de FILE posterior, com um arquivo externo. O nome do arquivo externo é reconhecido pelo sistema operacional e estará disponível para uso por qualquer programa no endereço especificado. Veja também a instrução FILE, anterior, com exemplo completo. Deve-se evitar as palavras reservadas CON, NUL, PRN, LPT1 - LPT9 e COM1 para nomes de arquivos.

Sintaxe

```
FILENAME NOME DO ARQUIVO
'DRIVE:\ENDEREÇO\NOME DO ARQUIVO EXTERNO.DAT' <HOST-
OPÇÕES>;
```

NomeDoArquivo – Especifica o nome de arquivo SAS a ser gravado externamente. É um apelido temporário para o arquivo. Deve ser o mesmo da declaração de FILE. Pode ser qualquer nome SAS válido para arquivos SAS.

NomeDoArquivoExterno – Especifica o endereço e o nome do arquivo a ser gravado em disco. Podem ser quaisquer endereços e nomes de arquivos válidos para o sistema operacional. Sugere-se que a extensão ou sufixo do arquivo seja '.SAS' para ser compatível com o SAS.

Host-Opções – Há várias host-opções para execução desta instrução, mas não são consideradas essenciais.

6.5.10 IF THEN (INSTRUÇÃO)

Avalia uma expressão e, condicionalmente, executa cláusula(s) ou declarações subsequentes.

Sintaxe

```
IF EXPRESSÃO THEN CLÁUSULA; <ELSE CLÁUSULA;>
```

ou

**IF EXPRESSÃO THEN DO; DECLARAÇÕES; END; <ELSE DO;
DECLARAÇÕES; END;>**

Expressão – Especifica uma ou mais expressões a serem avaliadas como condição para execução de uma instrução ou de declarações subsequentes;

Cláusula – É uma instrução a ser executada após THEN se o resultado da expressão for verdadeiro ou após ELSE se o resultado da expressão for falso;

Declarações – São quaisquer declarações SAS válidas a serem executadas entre THEN DO e END se o resultado da expressão for verdadeiro, ou entre ELSE DO e END, se o resultado for falso.

6.5.11 INFILE (INSTRUÇÃO)

Especifica o nome um arquivo fonte a ser lido por uma declaração de INPUT. Um arquivo fonte é um arquivo a ser lido; geralmente é um arquivo externo. Deve-se evitar as palavras reservadas CON, NUL, PRN, LPT1 - LPT9 e COM1 para nomes de arquivos.

Sintaxe

**INFILE NOME DO ARQUIVO <ENCODING=VALOR DE CODIGO> <OPÇÕES>
<HOST-OPÇÕES>;**

Nomedoarquivo – Identifica o nome do arquivo fonte da entrada de dados, geralmente externo.

Opções – Há algumas dezenas de opções para a instrução INFILE, mas raramente são utilizadas em programas simples. Algumas opções úteis são as de caracteres de delimitação. Exemplos: infile 'a:\arq.txt' **DELIMITER=/'**; * lê o arquivo externo arq.txt com colunas separadas por uma barra /; infile 'a:\arq.txt' **EXPANDTABS**; * lê o arquivo externo arq.txt com colunas separadas por caracteres de tabulação;. Tabelas do MS-Excel salvas como somente texto têm caracteres de tabulação como separador das colunas e podem ser lidas diretamente com a opção EXPANDTABS.

ENCODING=encoding-value – Especifica um código de caracteres para processar um arquivo externo.

Host-opções – Nomes de entrada e saída externos que são específicos para o sistema operacional.

6.5.12 INPUT (INSTRUÇÃO)

A instrução INPUT comanda a entrada de dados. É através desta instrução que se informa ao programa em que formato os dados estão, definindo nomes de variáveis e a forma de separação dos dados, seja em arquivos externos ou em dados escritos no próprio programa após a instrução CARDS ou DATALINES.

Sintaxe

**INPUT <#N> <@N>VARIÁVEL-1 <FORMATO> <LOCALIZAÇÃO> <...><#N>
<@N>VARIÁVEL-N <FORMATO> <LOCALIZAÇÃO>>
<DIRETIVAS DE REGISTROS> <@@>;**

#n – Determina que o número de ordem da linha que deve ser lida é a de número n em um grupo de linhas de dados, em que n é um número inteiro; quando se usa este tipo de controle, o SAS assume que as linhas de dados devem ser lidas por grupos com um número de linhas igual à de maior ordem informada como valor de n. Não é necessário ler todas as linhas do grupo entre a de menor e a de maior número de ordem do grupo, mas o número total de linhas de cada grupo é sempre igual ao valor da linha de maior número de ordem.

@n – Determina que a variável seja lida a partir do dígito n da linha que está sendo lida, em que n é um número inteiro; pode ser usada a opção **@(expressão)** que lê o valor da variável a partir do dígito resultante da expressão entre parênteses, em que (expressão) é uma expressão SAS que resulta em um número inteiro.

Variável – Especifica o nome da variável ou variáveis a serem lidas na posição corrente do registro atual. Cada variável pode ser seguida por uma especificação de formato de variável e/ou de posição (informat) da mesma no registro (linha de dados) que está sendo lido.

Formato – Especifica um formato de entrada de dados. Os formatos padrão de entrada de dados podem ser de caracteres, quando o nome da variável no INPUT deve ser seguido pelo caractere \$; ou numérico, quando não o caractere \$ após o nome da variável. A especificação de dados de variáveis do tipo caractere é obrigatória e se não for feita, o programa apresenta uma mensagem de erro e não processa o restante do passo de programa. Há muitos tipos de formatos de entrada possíveis com a instrução INPUT. Se um formato for especificado, os dados devem obrigatoriamente estar na forma especificada. O SAS, para dados numéricos de uma variável sem especificação de formato de entrada, assume o formato que possa servir para todos os dados encontrados para a variável. Para variáveis de caracteres sem especificação de comprimento (número de caracteres), o SAS lê dados de qualquer comprimento, mas converte-os internamente para o padrão de 8 caracteres de comprimento. Para mais informações, veja o manual on line *SAS Language Reference: Dictionary*.

m-n – Determina que a variável deva ser lida do dígito m até o n da linha ou registro corrente, sendo m e n números inteiros.

Diretivas de registros - A principal diretiva de leitura de registro é expressa pelo caractere de controle /, que determina o avanço da leitura de dados para o próximo registro.

Caractere branco – Quando valores são simplesmente escritos em linhas e em cada linha são separados por caracteres brancos, o SAS entende que a cada nova linha inicia um novo registro e que os caracteres brancos separam os valores das variáveis listadas na declaração de INPUT.

@@ - Informa que as variáveis listadas devem ser lidas em grupos consecutivos da mesma linha de dados, criando um novo registro a cada grupo completo de variáveis lidas na mesma linha, quando uma nova linha é encontrada a leitura continua a ser feita em grupos da nova linha e assim por diante.

A seguir é feito um “Exemplo de declarações de input”.

* Exemplo de declarações de input;

data a;

input @1 nome \$ @20 sexo \$ @(20+2) idade 2. peso 4.1;

cards;

nomeA67890123456789M 10 217

nomeB67890123456789F 20 593

;

proc print DATA=A;

run;

1
2
3
4
5
6
7
8
9

O programa do exemplo lê:

- A partir do dígito 1, a variável NOME, do tipo caractere;
- A partir do dígito 20, a variável SEXO, do tipo caractere;
- A partir do dígito 22, a variável IDADE, do tipo numérico, com dois dígitos significativos e nenhum após o ponto decimal;
- Após o primeiro caractere branco que segue a idade, lê a variável PESO, do tipo numérico decimal com um total de 4 dígitos, sendo dois dígitos

significativos e um após o ponto decimal, implicitamente um dígito é incluído para o ponto decimal após a leitura, por isso é necessário declarar um total de 4 dígitos. Se a variável tivesse sinal de negativo ou positivo seria necessário adicionar mais um dígito ao comprimento total.

O programa imprime o arquivo A como segue:

```

THE SAS
Obs   nome   sexo  idade  peso
1   nomeA678  M     10    21.7
2   nomeB678  F     20    59.3

```

Observa-se que o SAS converteu o conteúdo da variável NOME para o comprimento padrão de 8 dígitos, pois o comprimento da variável não foi explicitamente especificado.

6.5.13 KEEP (INSTRUÇÃO)

A instrução KEEP informa as variáveis que devem ser retidas em um arquivo SAS. As demais variáveis por ventura existentes, serão eliminadas do arquivo.

Sintaxe

■ **DATA NOME DO ARQUIVO (KEEP=LISTA DE VARIÁVEIS);**

NomeDoArquivo – É o nome do arquivo de dados (DATA SET) que está sendo criado ou modificado pelo DATA STEP.

NomeDoArquivo – São os nomes das variáveis que devem ser mantidas no arquivo de dados após a conclusão do DATA STEP.

6.5.14 MERGE (INSTRUÇÃO)

A instrução MERGE intercala dois ou mais arquivos SAS através de uma ou mais variáveis previamente ordenadas, comuns aos arquivos que estão sendo intercalados. O ordenamento pode ser feito através do procedimento PROC SORT. Todos os arquivos envolvidos devem estar ordenados pelas variáveis de controle da intercalação, na mesma sequência de ordenamento, definidas em uma declaração de BY que é escrita imediatamente após a declaração de MERGE.

Sintaxe

```

MERGE NOME DO ARQUIVO-1 <(OPÇÕES)> NOME DO ARQUIVO-2
<(OPÇÕES)>
<... NOME DO ARQUIVO-N <(OPÇÕES)>> <END=VARIÁVEL>;
BY LISTA DE VARIÁVEIS DE INTERCALAÇÃO;

```

NomeDoArquivo – Identifica cada arquivo que será intercalado para formar o arquivo resultante da intercalação. Pelo menos dois arquivos devem ser utilizados.

Opções – São as opções de ordenamento para intercalação. Se não forem especificadas, o SAS intercala o valor de uma variável em um arquivo com o valor igual da mesma variável em cada um dos demais arquivos constantes da declaração de MERGE.

BY ListaDeVariáveisDelintercalacao – São os nomes das variáveis pelas quais se deseja intercalar os arquivos.

END=Variável – Nomeia e cria uma variável temporária que contém um indicador de final de arquivo (EOF).

Um programa usando dois arquivos intercalando-os pelas variáveis PARCELA e ESPECIE, através da instrução MERGE, pode ser como no “Exemplo – Intercalando arquivos SAS” a seguir.

```

* Exemplo – Intercalando arquivos SAS;
Data arquivo1;
input parcela especie $ dap;
cards;
01 angico 35.5
01 aroeira 22.0
01 timbauva 40.5
01 cabreuva 51.5
;
data arquivo2;
input parcela especie $ DAP;
cards;
02 timbauva 11.35
02 vassourao 19.0
02 angico 45.5
03 aroeira 12.0
03 angico 59.0
02 aroeira 32.5
03 Angico 33.0
03 cabreuva 37.0
;
Proc sort data=arquivo1; by parcela especie;
Proc sort data=arquivo2; by parcela especie;
data arquivo3;
merge arquivo1 arquivo2; by parcela especie;
proc print data=arquivo3;
run;

```

Como resultado têm-se o seguinte relatório HTML impresso:

Obs	PARCELA	ESPECIE	DAP
1	1	ANGICO	35.50
2	1	AROEIRA	22.00
3	1	CABREUVA	51.50
4	1	TIMBAUVA	40.50
5	2	ANGICO	45.50
6	2	AROEIRA	32.50
7	2	TIMBAUVA	11.35
8	2	VASSOURA	19.00
9	3	ANGICO	59.00

10	3	ANGICO	33.00
11	3	AROEIRA	12.00
12	3	CABREUVA	37.00

Se, em outro caso, fosse necessário intercalar aos dois primeiros do exemplo anterior com um arquivo contendo os dados das alturas das árvores e calcular o volume das mesmas em metros cúbicos utilizando a fórmula $VOLUME=0,6.DAP^2.ALTURA/40000$, isso poderia ser feito como no programa “Exemplo – Intercalando arquivos SAS” a seguir.

```

* Exemplo – Intercalando arquivos SAS;
Data arquivo1;
input parcela especie $ dap;
cards;
01 angico 35.5
01 aroeira 22.0
01 timbauva 40.5
01 cabreuva 51.5
;
data arquivo2;
input parcela especie $ DAP;
cards;
02 timbauva 11.35
02 vassourao 19.0
02 angico 45.5
03 aroeira 12.0
03 angico 59.0
02 aroeira 32.5
03 Angico 33.0
03 cabreuva 37.0
;
DATA ARQUIVO3;
INPUT PARCELA ESPECIE $ ALTURA;
CARDS;
3 ANGICO 8.5
3 ANGICO 6.0
1 ANGICO 6.0
1 AROEIRA 8.0
1 CABREUVA 11.0
2 ANGICO 8.5
2 AROEIRA 4.5
2 TIMBAUVA 4.5
2 VASSOURA 5.0
3 AROEIRA 3.00
1 TIMBAUVA 9.5
3 CABREUVA 7.0
;
Proc sort data=arquivo1; by parcela especie;
Proc sort data=arquivo2; by parcela especie;
Proc sort data=arquivo3; by parcela especie;
data arquivo;
merge arquivo1 arquivo2 ARQUIVO3; by parcela especie;
volume=0.6*dap**2*altura/40000;
proc print data=arquivo;
run;

```

Como resultado o SAS imprime o relatório HTML com o conteúdo do arquivo de dados chamado de ARQUIVO:

Obs	PARCELA	ESPECIE	DAP	ALTURA	VOLUME
1	1	ANGICO	35.50	6.0	0.11342
2	1	AROEIRA	22.00	8.0	0.05808
3	1	CABREUVA	51.50	11.0	0.43762
4	1	TIMBAUVA	40.50	9.5	0.23374
5	2	ANGICO	45.50	8.5	0.26396
6	2	AROEIRA	32.50	4.5	0.07130
7	2	TIMBAUVA	11.35	4.5	0.00870
8	2	VASSOURA	19.00	5.0	0.02708
9	3	ANGICO	59.00	8.5	0.44383
10	3	ANGICO	33.00	6.0	0.09801
11	3	AROEIRA	12.00	3.0	0.00648
12	3	CABREUVA	37.00	7.0	0.14375

6.5.15 OUTPUT (INSTRUÇÃO)

Orienta a gravação do registro atual em um ou mais arquivos de saída.

Sintaxe

■ **OUTPUT <NOMEDOARQUIVO-1> <...NOMEDOARQUIVO-N>;**

NomeDoArquivo – É o nome do arquivo de saída em que o registro corrente (linha de dados) deve ser escrito. Quando for criado um só arquivo já referenciado na declaração de DATA, o nome do mesmo pode ser omitido na declaração de OUTPUT.

A) Criando um só arquivo a partir do ARQUIVO resultante do exemplo anterior com os dados das árvores com DAP maior do que 30 cm no “Exemplo – Criando arquivos SAS a partir do arquivo em uso, com a instrução OUTPUT”.

```

* Exemplo – A) Criando arquivos SAS a partir do arquivo em      1
  uso, com a instrução OUTPUT;                                   2
* ... incluir aqui as linhas do Exemplo – Intercalando arquivos 3
  SAS, anterior;                                             4
data maiorq30;                                               5
set arquivo;                                                 6
if dap>30 then output;                                       7
proc print data=maiorq30;
run;

```

O resultado da impressão do arquivo MAIORQ30 em formato HTML é o seguinte:

The SAS

Obs	PARCELA	ESPECIE	DAP	ALTURA	VOLUME
1	1	ANGICO	35.5	6.0	0.11342
2	1	CABREUVA	51.5	11.0	0.43762
3	1	TIMBAUVA	40.5	9.5	0.23374
4	2	ANGICO	45.5	8.5	0.26396
5	2	AROEIRA	32.5	4.5	0.07130
6	3	ANGICO	59.0	8.5	0.44383
7	3	ANGICO	33.0	6.0	0.09801
8	3	CABREUVA	37.0	7.0	0.14375

B) No “Exemplo – Criando cinco arquivos através da instrução OUTPUT”, são criados 5 novos arquivos.

<pre> * Exemplo – B) Criando cinco arquivos através da instrução OUTPUT; * ... incluir aqui as linhas do Exemplo – Intercalando arquivos SAS, anterior; DATA ANG ARO CAB TIM VAS; SET ARQUIVO; IF ESPECIE EQ 'ANGICO' THEN OUTPUT ANG; IF ESPECIE EQ 'AROEIRA' THEN OUTPUT ARO; IF ESPECIE EQ 'CABREUVA' THEN OUTPUT CAB; IF ESPECIE EQ 'TIMBAUVA' THEN OUTPUT TIM; IF ESPECIE EQ 'VASSOURA' THEN OUTPUT VAS; PROC PRINT DATA=ANG; PROC PRINT DATA=ARO; PROC PRINT DATA=CAB; PROC PRINT DATA=TIM; PROC PRINT DATA=VAS; RUN; </pre>	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------

Como resultado o SAS imprime os arquivos ANG, ARO, CAB, TIM e VAS, criados através das instruções de OUTPUT condicionadas pela instrução IF:

Obs	PARCELA	ESPECIE	DAP	ALTURA	VOLUME
1	1	ANGICO	35.5	6.0	0.11342
2	2	ANGICO	45.5	8.5	0.26396
3	3	ANGICO	59.0	8.5	0.44383
4	3	ANGICO	33.0	6.0	0.09801

Obs	PARCELA	ESPECIE	DAP	ALTURA	VOLUME
1	1	AROEIRA	22.0	8.0	0.058080
2	2	AROEIRA	32.5	4.5	0.071297
3	3	AROEIRA	12.0	3.0	0.006480

Obs	PARCELA	ESPECIE	DAP	ALTURA	VOLUME
1	1	CABREUVA	51.5	11	0.43762
2	3	CABREUVA	37.0	7	0.14375

Obs	PARCELA	ESPECIE	DAP	ALTURA	VOLUME
1	1	TIMBAUVA	40.50	9.5	0.23374
2	2	TIMBAUVA	11.35	4.5	0.00870

Obs	PARCELA	ESPECIE	DAP	ALTURA	VOLUME
1	2	VASSOURA	19	5	0.027075

6.5.16 PUT (INSTRUÇÃO)

Especifica as variáveis e formatos a serem utilizados para escrever um arquivo externo definido através da instrução FILE.

Sintaxe

■ **PUT <ESPECIFICAÇÕES> <@ | @@>;**

Especificações – Especificam o que é escrito, como é escrito e onde é escrito. Normalmente incluem os nomes das variáveis cujos valores serão escritos no arquivo externo. Algumas especificações válidas são as seguintes:

Variáveis – Nomes das variáveis que serão escritas;

\$ – Identificador de variável de caractere que deve ser escrito após o nome deste tipo de variável;

n* – Determina que devam ser repetidas n vezes o caractere subsequente;

m-n – Determina que o valor da variável que antecede deve ser escrito do dígito m até o n, em que m e n são número inteiros;

INFILE – Escreve o último registro de dados;

ALL – Escreve todos os valores de todas as variáveis no arquivo externo;

@n – Escreve o valor da variável seguinte a partir do dígito n do registro atual, em que n é um número inteiro;

@(expressão) – Escreve o valor da variável seguinte a partir do dígito resultante da expressão entre parênteses, em que (expressão) é uma expressão SAS que resulta em um número inteiro;

- +n – Move o apontador n dígitos para frente;
- +VariavelNumerica – Move o apontador para frente conforme o valor da variável;
- +(expressão) – Move o apontador um número de dígitos igual ao resultado da expressão;
- #n – moves the pointer to line n;
- #VariavelNumerica – Move o apontador para a linha indicada pelo valor da variável numérica;
- #(expressão) – move o apontador pela linha dada pela expressão, em que (expressão) é uma expressão SAS que resulta em um número inteiro;
- / – Move o apontador para o primeiro dígito da linha seguinte.

@ | @@ – Mantém a linha atual para execução da próxima declaração de PUT.

O arquivo de saída do “Exemplo – Intercalando arquivos SAS” chamado de ARQUIVO, poderia ser salvo em disco pela seguinte continuação do programa:

* continuação do Exemplo – Intercalando arquivos SAS;	1
* ... incluir aqui as linhas do Exemplo – Intercalando arquivos SAS, anterior;	2
DATA _NULL_;	3
fileName arq4 'C:\EXEMPLOS\arqext.dat';	4
SET ARQUIVO;	5
file arq4;	6
put parcela especie \$ dap 4.1;	7
run;	8

O conteúdo do arquivo externo gravado em disco com o nome de ARQEXT.DAT no endereço C:\EXEMPLOS\ é o seguinte:

```
1 ANGICO 35.5
1 AROEIRA 22.0
1 CABREUVA 51.5
1 TIMBAUVA 40.5
2 ANGICO 45.5
2 AROEIRA 32.5
2 TIMBAUVA 11.4
2 VASSOURA 19.0
3 ANGICO 59.0
3 ANGICO 33.0
3 AROEIRA 12.0
3 CABREUVA 37.0
```

6.5.17 RETAIN (INSTRUÇÃO)

A instrução RETAIN mantém o valor das variáveis declaradas na memória, até lhes seja atribuído um novo valor. É um recurso útil quando se necessita passar valores de um registro para o outro, como o saldo anterior de uma conta, por exemplo.

A instrução RETAIN é utilizada também para simplesmente declarar valores iniciais para variáveis.

As variáveis podem ser declaradas com ou sem um valor inicial, mas usualmente com variáveis numéricas se atribui o valor inicial zero. Se não for atribuído um valor inicial, a variável inicia com valor perdido.

Sintaxe

```
RETAIN NOME DA VARIÁVEL <VALOR INICIAL> <... NOME DA VARIÁVEL
<VALOR INICIAL>>;
```

NomeDaVariável – É o nome da variável cujo valor deve ser mantido;

ValorInicial – É o valor inicial da variável.

O uso da instrução retain pode ser como no “Exemplo – Acumulando valores de um registro para o outro com RETAIN”.

```
* Exemplo – Acumulando valores de um registro para o outro
  com RETAIN;
Data arq (KEEP=especie frequ frequ_ac);
retain v_retido 0;
frequ_ac=frequ+V_retido;
v_retido=frequ_AC;
Input especie $ frequ;
cards;
angico 10
cabreuva 5
timbauva 7
;
proc print data=arq;
run;
```

As declarações do “Exemplo – Acumulando valores de um registro para o outro com RETAIN” são descritas a seguir:

DATA ARQ (KEEP=ESPECIE FREQU FREQU_AC); – Cria o arquivo ARQ e mantém no arquivo ao final do DATA STEP somente as variáveis ESPECIE, FREQU e FREQU_AC.

RETAIN V_RETIDO 0; – Retém na memória o valor da variável V_RETIDO e atribui a ela o valor inicial zero.

FREQU_AC=FREQU+V_RETIDO; – Soma o valor da variável lida FREQU com o valor retido em V_RETIDO e atribui o resultado à variável FREQU_AC.

V_RETIDO=FREQU_AC; – Atribui o valor da variável FREQU_AC à variável V_RETIDO.

INPUT ESPECIE \$ FREQU; – Instrui a entrada da variável ESPECIE do tipo caracteres e da variável FREQU do tipo numérico.

CARDS; – Sinaliza o início das linhas de dados.

ANGICO 10

CABREUVA 5

TIMBAUVA 7

; – Sinaliza o final das linhas de dados.

PROC PRINT DATA=ARQ; – Imprime o arquivo ARQ.

RUN; – Executa o programa.

O relatório de impressão do arquivo ARQ pelo programa, em formato HTML, é o seguinte:

The SAS

Obs	ESPECIE	FREQU	FREQU_AC
1	ANGICO	10	10
2	CABREUVA	5	15
3	TIMBAUVA	7	22

O programa “Exemplo – Acumulando valores de um registro para o outro com RETAIN” lê as frequências de três espécies na variável FREQU e as acumula na variável FREQU_AC, utilizando a variável V_RETIDO para manter o valor da frequência acumulada de um registro para o outro, pois o SAS lê uma linha de dados, executa todas as declarações entre a instrução INPUT e a instrução CARDS e, então, lê a próxima linha de dados e, assim, consecutivamente até encontrar o final dos dados.

6.5.18 SET (INSTRUÇÃO)

Lê uma observação de um ou mais arquivos SAS. Se após a instrução SET não for informado o nome de um arquivo, o SAS assume o último arquivo em uso pelo programa.

Sintaxe

■ **SET <NOMEDOARQUIVO-1 <(OPÇÕES)>> <...NOMEDOARQUIVO-N <(OPÇÕES)>> <OPÇÕES>;**

(Opções) – Após o nome do arquivo especificam ações a realizar enquanto estiver lendo variáveis ou observações.

Opções – Ao final da declaração são opções de entrada de dados.

Para maiores informações sobre opções da instrução SET, veja o manual: SAS® Language Reference, Version 8, Chapter 6.

6.5.19 SUM (INSTRUÇÃO)

Adiciona o resultado de uma expressão a uma variável acumulativa do tipo numérico.

Sintaxe

■ **NOMEDAVARIAVEL + EXPRESSÃO;**

NomeDaVariavel – É o nome da variável acumulativa.

Expressão – É qualquer expressão SAS válida para valores numéricos.

6.5.20 UPDATE (INSTRUÇÃO)

Atualiza um arquivo mestre com os dados de um arquivo de transação. É utilizada em conjunto com a instrução BY, que informa as variáveis de controle para a atualização à semelhança da instrução MERGE. Os valores do arquivo mestre serão sobrepostos com os valores do arquivo de transação, conforme as variáveis de controle listadas na instrução BY.

Sintaxe

```
UPDATE ARQUIVOMESTRE ARQUIVODETRANSAÇÃO;
BY LISTADEVARIAVEIS;
```

ArquivoMestre – É o nome do arquivo que será atualizado.

ArquivoDeTransação – É o nome do arquivo que fornecerá os dados para atualizar o arquivo mestre.

ListaDeVariáveis – São os nomes das variáveis de controle.

6.5.21 WHERE (INSTRUÇÃO)

Define condições para a seleção de observações de um arquivo SAS.

Sintaxe

```
WHERE EXPRESSÃO;
```

Expressão – É qualquer expressão SAS que tenha um resultado que quando for verdadeiro seleciona o registro atual, ou o despreza se o resultado da expressão for falso.

Exemplo 4.16 – Selecionando registros com WHERE

Utilizando o arquivo externo gravado em disco no endereço C:\EXEMPLOS\, com o nome de ARQEXT.DAT pelo programa “continuação do Exemplo – Intercalando arquivos SAS”, poderiam ser selecionadas as árvores das espécies com nome começando pela letra A e DAP>20, através do programa “EXEMPLO – selecionando dados com WHERE”, a seguir.

```
* EXEMPLO – selecionando dados com where;          1
DATA ARQ;                                           2
  INFILE 'C:\EXEMPLOS\ARQEXT.DAT';                 3
  INPUT PARCELA ESPECIE $ DAP 4.1;                 4
  WHERE SUBSTR(ESPECIE,1,1)='A' AND DAP>20;        5
PROC PRINT DATA=ARQ;                               6
RUN;                                                7
```

As declarações do programa “EXEMPLO – SELECIONANDO DADOS COM WHERE” realizam o seguinte:

DATA ARQ; – Cria o arquivo SAS denominado ARQ.

INFILE 'C:\EXEMPLOS\ARQEXT.DAT'; – Chama o arquivo externo ARQEXT.DAT localizado no endereço C:\SAS\.

INPUT PARCELA ESPECIE \$ DAP 4.1; – Instrui a entrada das variáveis ESPECIE (de caracteres) e DAP (numérica).

WHERE SUBSTR(ESPECIE,1,1)='A' AND DAP>20; – Determina a seleção dos registros **onde** o valor de uma subsérie de caracteres, começando pela posição 1, com o comprimento de 1 caractere, do valor da variável ESPECIE é igual à letra A e o DAP tem valor maior do que 20.

PROC PRINT DATA=ARQ; – Imprime o arquivo ARQ.

RUN; – Executa o programa.

Como resultado é impressa a seguinte tabela:

The SAS

Obs	PARCELA	ESPECIE	DAP
1	1	ANGICO	35.5
2	1	AROEIRA	22.0
3	2	ANGICO	45.5
4	2	AROEIRA	32.5
5	3	ANGICO	59.0
6	3	ANGICO	33.0

Capítulo 7 : Explorer do SAS

7.1 PASTA WORK

Todos os arquivos de memória do SAS, criados em um programa, são colocados na pasta “Work” (Figura 37), dentro da pasta “Libraries” do EXPLORER.

Após executar um programa com a criação de arquivos, pode-se clicar na aba EXPLORER e dar um clique duplo na pasta “Libraries” e depois outro clique duplo na pasta “Work” e lá estarão os arquivos que foram criados pelo programa. Para ver seu conteúdo, dê um clique duplo no nome do arquivo que abrirá uma planilha com seus dados.

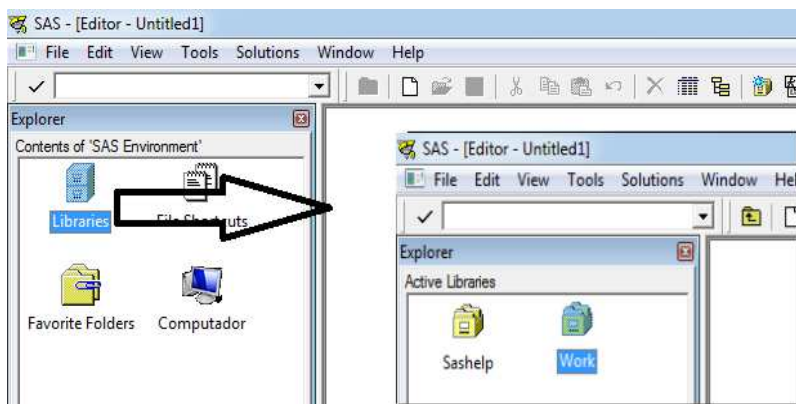


Figura 37: A pasta Work, dentro de Libraries

7.2 CRIANDO PASTAS DE ARQUIVOS

É possível criar pastas de arquivos no SAS. À semelhança do Windows, o SAS possui um sistema de arquivos, que nos permite armazenar dados e programas no próprio sistema SAS.

A criação de uma pasta de arquivos é realizada ativando-se a aba do explorador (EXPLORER) por meio do menu FILE.

Para criar uma biblioteca SAS, antes é preciso criar uma pasta no Windows. Por exemplo crie a pasta “C:\Exemplos” na pasta raiz do computador. Então, abra o SAS e siga as instruções a seguir:

- Clique na aba |EXPLORER|;
- Dê um clique duplo na pasta |Libraries| dentro do EXPLORER;
- Clique no menu |FILE|, |New| (Figura 37);
- Na tela de criação de nova biblioteca (New Librarie), escreva o nome da nova pasta SAS que denominamos de “Exemplos”, no campo Name; deixe a opção de motor (Engine) como “default”, ou seja: padrão; e preencha o caminho para a pasta do Windows “C:\Exemplos” no campo *Path* (Figura 38);
- Marque a opção “*Enable at Startup*” para que a pasta seja acessada sempre que abrir o SAS;
- Clique em |OK|;
- No explorer do SAS aparecerá a pasta recém criada (Figura 39).

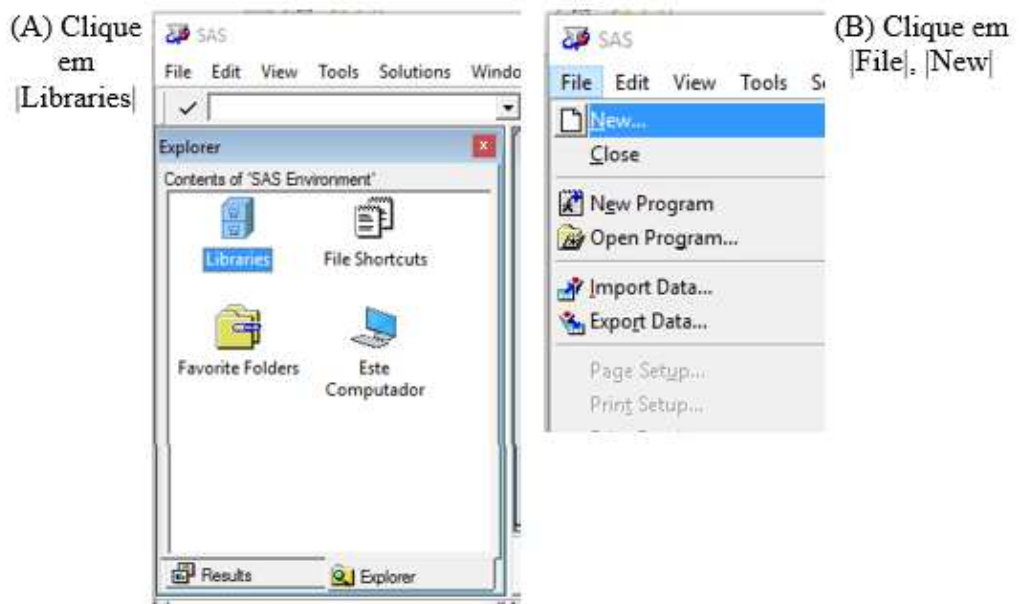


Figura 38: Criando uma nova pasta no SAS

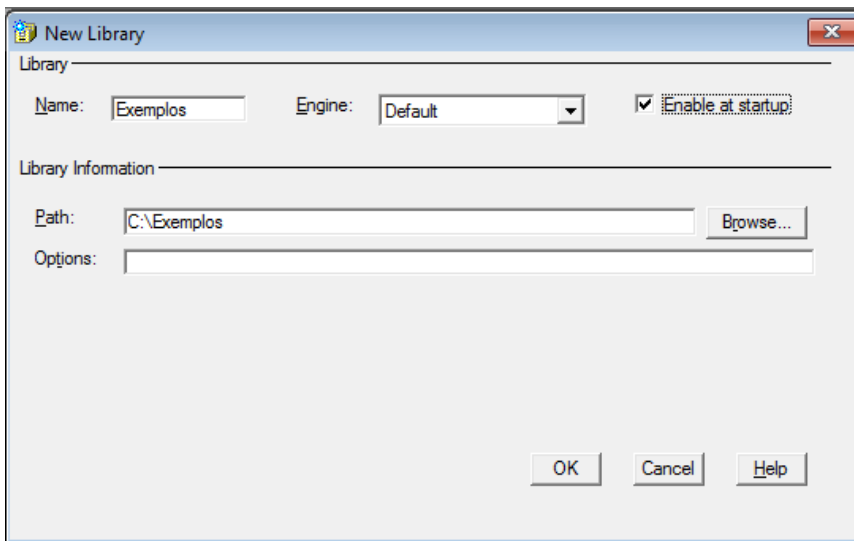


Figura 39: Criando uma nova pasta no SAS

A partir da criação da pasta (Figura 40), é possível importar arquivos do EXCEL, por exemplo, para dentro dela, abrindo-a e depois importando a planilha do EXCEL. Os programas SAS podem acessar diretamente os arquivos colocados na pasta [EXEMPLOS], usando o comando LIBNAME. Se criarmos um arquivo chamado DADOS1 na pasta, pode-se acessá-lo como no “Exemplo - usando o comando LIBNAME”.

```
* EXEMPLO - usando o comando LIBNAME;  
LIBNAME alias "c:\EXEMPLOS\";  
DATA DADOS;  
  SET Alias.dados1;  
RUN;
```

Onde “ALIAS” é o apelido que queremos atribuir à pasta, que pode ser qualquer nome SAS válido. O comando SET chama o arquivo DADOS1 que está na pasta ALIAS, e o coloca no arquivo de memória “DADOS”, deixando-o pronto para uso no programa SAS.

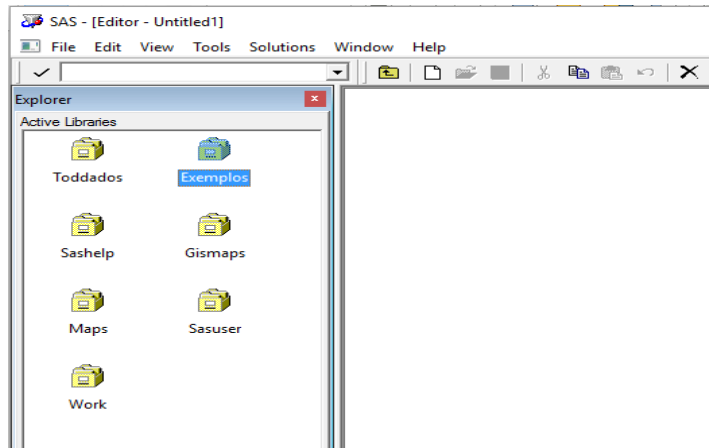


Figura 40: Pasta “Exemplos”, criada no EXPLORER do SAS

Capítulo 8 : Formatos de saída (FORMAT) e entrada (INFORMAT)

As instruções de formato são importantes porque fixam os formatos para entrada e saída de dados.

O SAS utiliza o formato interno padrão para **variáveis numéricas** que lhes dá a maior precisão. Quando se deseja que o formato seja diferente do padrão (maior precisão), é preciso especificar o formato na construção do arquivo de dados após a instrução DATA, num DATA STEP, o que geralmente é feito, ou através de uma instrução INPUT, ou de uma instrução FORMAT.

Para **variáveis de caracteres**, o padrão é o comprimento de oito (8) dígitos para impressão, embora o SAS guarde na memória o valor da variável com o comprimento em que foi gerado. Quando se deseja que as variáveis sejam impressas em maior comprimento, é necessário declarar o comprimento na construção do arquivo de dados, após a instrução DATA, num DATA STEP, através de uma instrução LENGTH, por exemplo.

O SAS tem muitos padrões prontos para serem usados com variáveis de caracteres, numéricas, data e horário, mas o programador pode criar seus próprios formatos através do procedimento PROC FORMAT, que não deve ser confundido com a instrução FORMAT que é utilizada com o DATA STEP.

8.1 FORMAT (INSTRUÇÃO DE FORMATOS DE SAÍDA)

FORMAT é uma instrução do SAS usada com o DATA STEP para fixar o formato de impressão ou de gravação dos valores de variáveis em relatórios ou arquivos de saída, respectivamente.

Formatos podem ser especificados através de:

- Instruções PUT, ATTRIB e FORMAT;
- Funções PUT, PUTC, PUTN e %SYSFUNC;
- Procedimento PROC FORMAT.

A instrução PUT com um formato após o nome da variável, usa o formato especificado para escrever o valor da variável em um arquivo de dados.

A função PUT escreve uma variável, numérica ou de caracteres, ou uma constante, com qualquer formato válido e retorna o resultado do valor em caracteres.

A função macro %SYSFUNC (ou %QSYSFUNC) executa funções SAS internas ou funções SAS definidas pelo usuário e aplica um formato opcional ao resultado da função na saída de um DATA STEP.

A instrução FORMAT, a partir da linha de declaração de formato, associa um formato permanentemente à uma variável em um programa.

A instrução ATTRIB associa permanentemente atributos a uma ou mais variáveis, incluindo formatos.

8.1.1 FORMATOS DE SAÍDA DEFINIDOS PELO USUÁRIO

Além dos formatos internos do sistema, o usuário pode criar seus próprios formatos utilizando o procedimento PROC FORMAT.

Há duas formas de disponibilizar os formatos definidos pelo usuário:

- Criando formatos permanentes usando PROC FORMAT em um programa;
- Gravando o código fonte que cria o formato em um passo PROC FORMAT com o programa que os usa.

Para informações adicionais sobre formatos de saída, veja:

- Procedimento FORMAT no *SAS Procedures Guide*;
- Instrução PUT no *SAS Language Reference: Dictionary*;
- Função PUT no *SAS Language Reference: Dictionary*;
- Função Macro %SYSFUNC no *SAS Macro Language: Reference*;
- Instrução FORMAT no *SAS Language Reference: Dictionary*;
- Instrução ATTRIB no *SAS Language Reference: Dictionary*.

Sintaxe

Geral:

■ <\$>NOMEDOFORMATO<W>.<D>

\$ - Indica um formato de caractere. Quando ausente, indica que o formato é numérico.

NomeDoFormato – Nome do formato atribuído pelo usuário, ou o nome de um formato SAS. Formatos definidos pelo usuário devem ser previamente declarados através de uma instrução VALUE em um procedimento PROC FORMAT.

w – Especifica o comprimento, que para muitos formatos é o número de dígitos no arquivo de saída.

d – Especifica o número de casas decimais para formatos numéricos.

FORMAT (instrução):

■ **FORMAT VARIÁVEIS** <FORMATO> <DEFAULT=FORMATOPADRÃO>;

Variáveis - Nomes de variável(is) a serem associadas a um formato. Para desassociar um formato de uma variável usa-se a instrução FORMAT seguida do nome da variável sem nenhum formato.

Formato – Formato especificado para a impressão ou gravação da variável.

8.1.2 FORMATOS DE SAÍDA PARA VARIÁVEIS DE CARACTERES

Formato	Descrição	Amplitude	Comprimento	Alinhamento
\$w.	Padrão geral para caracteres	1-32767	1 ou o compr. da variável	esquerdo
\$ASCIIw.	Converte caracteres para o padrão ASCII	1-32767	1	esquerdo
\$CHARw.	Padrão geral para caracteres	1-32767	1 ou comprim.	esquerdo
\$EBCDICw.	Converte caracteres para o padrão EBCDIC	1-200	1	
\$HEXw.	Converte caracteres para o padrão hexadecimal	1-32767	comprim.	esquerdo
\$MSGCASEw.	Converte caracteres para maiúsculo quando com opção MSGCASE ativa	1-200	8 ou comprim.	esquerdo
\$QUOTEw.	Escreve valores entre aspas	1-200	8 ou comprim.	esquerdo
\$REVERJw.	Escreve na ordem inversa preservando brancos	1-32767	1 se sem especificação	direito
\$REVERSw.	Escreve na ordem inversa e alinha a esquerda	1-32767	1 se sem especificação	esquerdo
\$UCS2Xw.	Escreve em 16-bit UCS2 Unicode sem marca de ordem de byte.	1-32767	8	esquerdo
\$UCS2Bw.	Escreve um valor SAS value em 16-bit UCS2 Unicode indiano grande sem marca de ordem de byte.	1-32767	8	esquerdo
\$UCS2Lw.	Escreve um valor SAS value em 16-bit UCS2 Unicode indiano pequeno.	1-32767	8	esquerdo
\$UTF8Xw.	Escreve em código Unicode Transformation Format (UTF-8).	1-32767	8	esquerdo
\$UPCASEw.	Converte caracteres para maiúsculo	1-200	8 ou comprim.	esquerdo
\$VARYINGw.	Escreve dados de caráter de comprimento variado	1-200	8 ou comprim.	esquerdo

Obs.: Outros formatos estão disponíveis – ver manuais do SAS.

8.1.3 FORMATOS DE SAÍDA PARA VARIÁVEIS NUMÉRICAS

Formato	Descrição	Amplitude	Amplitude Decimal	Comprimento	Alinhamento
w.d	Padrão numérico	1-32	d<w		direito
BESTw.	O SAS escolhe a melhor anotação	1-32		12	direito
BINARYw.	Converte valores numéricos para representação binária	1-64		8	esquerdo
COMMAw.d	Escreve valores numéricos com vírgulas e pontos decimais	2-32	0 or 2	6	direito
COMMAXw.d	Escreve valor numérico com vírgulas e períodos	2-32	0 or 2	6	direito
DOLLARw.d	Escreve valores numéricos com sinais de cifrão, vírgulas e pontos decimais	2-32	0 or 2	6	direito
DOLLARXw.d	Escreve valores numéricos com sinais de cifrão, períodos e vírgulas	2-32	0 or 2	6	direito
Dw.s	Escreve dígitos significantes			12	direito
Ew.	Escreve valores numéricos em notação científica	7-32		12	direito
FLOATw.d	Gera um valor nativo de ponto flutuante de precisão simples, multiplicando um número elevado a 10ª potencia	4		4	esquerdo
FRACTw.	Converte valores numéricos para frações	4-32		10	direito
ROMANw.	Escreve valores numéricos como numerais romanos	2-32		6	esquerdo
Zw.d	Impressão de zeros principais	1-32		1	direito

Formato	Descrição	Amplitude	Amplitude Decimal	Comprimento	Alinhamento
S370FFw.d	Escreve para nativo dados numéricos padrão mainframe IBM	1-32	0-10	12	
WORDFw.	Escreve valores numéricos como palavras, com frações mostradas numericamente	5-32767		10	
WORDSw.	Escreve valores numéricos como palavras	5-32767		10	
YENw.d	Escreve valores numéricos com sinais de iene, vírgulas e pontos decimais	1-32		1	direito
ZDw.d	Escreve dados em formato decimal dividido em zonas	1-32		1	esquerdo

Obs.: Outros formatos estão disponíveis – ver manuais do SAS.

8.1.4 FORMATOS DE SAÍDA PARA VARIÁVEIS DE DATA_E_HORÁRIO

Formato	Descrição	Amplitude	Comprimento
DATEw.	Escreve valores de data na forma ddmmaa ou ddmmaaaa	5-9	7
DATEAMPWw.d	Escreve valores de data-horário na forma ddmmaa:hh:mm:ss com É ou PM	7-40	19
DATETIMEw.d	Escreve valores de data-horário na forma ddmmaa:hh:mm:ss.ss	7-40	16
DAYw.	Escreve dia do mês	2-32	2
DDMMYYw.	Escreve valores de data na forma ddmmaa ou ddmmaaaa	2-10	8
DDMMYYxw.	Escreve valores de data na forma ddmmaa ou ddmmaaaa com um separador especificado	2-10	8
DOWNAMEw.	Escreve valores de dados como o nome do dia da semana	1-32	9
DTDATEw.	Espera um valor de data-horário como contribuição e escreve valores de data na forma ddmmaa ou ddmmaaaa	5-9	7
DTMONYYw.	Escreve a parte de data de um valor de data-horário como o mês e ano no mmaaaa na forma ou mmaaaa	5-7	5
WORDDATEw.	Escreve valores de data como o nome do mês, o dia e o ano na forma mês-nome dd, aaaa	3-32	18
WORDDATXw.	Escreve valores de data como o dia, nome do mês, e o ano na forma dd na forma mês-nome aaaa	3-32	18
DTWKDATXw.	Escreve a parte de data de um valor de data-horário como o dia da semana e a data pelo dia-da-semana na forma, dd mês-nome aa (ou aaaa)	3-37	29
DTYEARw.	Escreve a parte de data de um valor de data-horário como o ano na forma aa ou aaaa	1-4	4
YEARw.	Escreve valores de data como o ano	2-32	4
YYMMxw.	Escreve valores de data como o ano e mês e os separa por um caractere	5-32	6
YYMMDDw.	Escreve valores de data na forma aammdd ou aaaammdd	2-8	8
YYMMDDxw.	Escreve valores de data na forma aammdd ou aaaammdd com um separador especificado	2-10	8
YYMONw.	Escreve valores de data como o ano e abreviação de mês	5-32	7
YYQw.	Escreve valores de data como o ano e quarto de ano	4-32	6
YYQxw.	Valores de data na formatos como o ano e quarto, separados por um caractere	4-32	6
YYQRxw.	Escreve valores de data como o ano e o quarto em numeral romano e os separa com caractere	6-32	8

Obs.: Outros formatos estão disponíveis – ver manuais do SAS.

8.2 INFORMAT (INSTRUÇÃO DE FORMATOS DE ENTRADA)

INFORMAT é uma instrução usada com o SAS para ler valores de variáveis.

Formatos podem ser especificados através de:

- Instruções INPUT, ATTRIB e INFORMAT;
- Funções INPUT, INPUTC e INPUTN;
- Procedimento PROC INFORMAT.

A instrução INPUT com um formato após o nome da variável, usa o formato especificado para escrever o valor da variável em um arquivo de dados.

A função INPUT escreve uma variável, numérica ou de caracteres, ou uma constante, com qualquer formato válido e retorna o resultado do valor em caracteres.

A instrução INFORMAT, a partir da linha de declaração de formato, associa um formato de entrada a uma variável de forma permanente em um programa.

A instrução ATTRIB associa permanentemente atributos a uma ou mais variáveis, incluindo formatos.

Sintaxe

Geral:

■ `<$>NOMEDOFORMATO<W>.<D>`

\$ - Indica um formato de caractere. Quando ausente, indica que o formato é numérico.

NomeDoFormato – Nome do formato atribuído pelo usuário, ou o nome de um formato SAS. Formatos definidos pelo usuário devem ser previamente declarados através de uma instrução VALUE em um procedimento PROC FORMAT.

w – Especifica o comprimento, que para muitos formatos é o número de dígitos no arquivo de saída.

d – Especifica o número de casas decimais para formatos numéricos.

INFORMAT (instrução):

■ `INFORMAT VARIÁVEIS<INFORMAT> <DEFAULT=INFORMAT-PADRÃO>;`

Variáveis - Nomes de variável(is) a associar com um INFORMAT. Para desassociar um informat de uma variável, usa-se o nome da variável após a instrução INFORMAT sem nenhum informat.

Informat – Especificação de um formato de entrada de dados (informat) para leitura de valores de variável(is). Informats associados com variáveis usando uma instrução INFORMAT que se comporta como informat usado com sinal modificador de dois pontos. O SAS lê as variáveis usando lista de input, mas com um informat.

8.2.1 FORMATOS DE ENTRADA PARA VARIÁVEIS DE CARACTERES

Informat	Descrição	Amplitude	Comprimento
\$w.	Caractere padrão	1-3276	1
\$CHARw.	Lê dados de caractere com espaços em branco	1-3276	1
\$EBCDICw.	Converte dados EBCDIC a formato nativo de caractere	1-3276	1
\$REVERJw.	Lê dados de caractere da direita para esquerda e preserva espaços em branco	1-32767	1 se w não é especificado
\$REVERSw.	Lê dados de caractere da direita para esquerda e alinha a esquerda	1-32767	1 se w não é especificado
\$UPCASEw.	Converte caracteres para maiúsculo	1-3276	8

Informat	Descrição	Amplitude	Comprimento
\$VARYINGw.	Lê dados de caractere de comprimento variado	1-3276	8

Obs.: Outros formatos estão disponíveis – ver manuais do SAS

8.2.2 FORMATOS DE ENTRADA PARA VARIÁVEIS NUMÉRICAS

Formato	Descrição	Amplitude	Comprimento	Alinhamento
w.d	Lê dados em padrão numérico	1-32	0-31	
BZw.d	Converte espaços em branco a zeros	1-32	0-31	1
COMMAw.d	Remove caracteres embutidos	1-32	0-31	1
COMMAXw.d	Remove caracteres embutidos	1-32	0-31	1
Ew.d	Lê valores numéricos que são armazenados em notação científica e precisão dupla	7-32	0-31	12
NUMXw.d	Lê valores numéricos com uma vírgula para o ponto decimal	1-32		12
PERCENTw.d	Converte porcentagens em valores numéricos	1-32		6
S370FF	Lê dados numéricos para EBCDIC	1-32	0-31	12
S370FPDw.d	Lê dados empacotados em formato de mainframe IBM	1-16	0-10	1
ZDw.d	Leituras divididas em zonas de dados decimais	1-32		1
ZDBw.d	Leituras divididas em zonas de dados decimais nos quais os zeros foram espaço em branco esquerdo	1-32	1-10	1
ZDVw.d	Lê e valida dados decimais divididos em zonas	1-32		1

Obs.: Outros formatos estão disponíveis – ver manuais do SAS

8.2.3 FORMATOS DE ENTRADA PARA DATA, HORÁRIO E DATA-HORÁRIO

Informat	Descrição	Amplitude	Comprimento
DATEw.	Lê valores de data na forma DDMMYY ou DDMMYYYY	7-32	7
DATETIMEw. de	Lê valores data-horário na forma ddmmmyy e hh:mm:ss.ss ou ddmmmyyyy hh:mm:ss.ss	13-40	18
DDMMYYw.	Lê valores de data na forma DDMMYY ou DDMMYYYY	6-32	8
EURDFEw.	Lê valores de data internacional	7-32	7
EURDFTw.	Lê data-horário internacional na forma ddmmmyy e hh:mm:ss.ss ou ddmmmyyyy hh:mm:ss.ss	13-40	18
EURDFYw.	Lê valores de data de mês e de ano na forma mmmyy ou mmmyyyy	5-32	5
JULIANw.	Lê datas Juliano na forma (YYDDD ou YYYYDDD)	5-32	5
MMDDYw.	Lê valores de data na forma mmddy ou mmddyyy	6-32	6
MONYYw.	Lê data de mês e ano na forma mmmyy ou mmmyyyy	5-32	5
TIMEw.d	Lê hora minuto e segundos na forma hh:mm:ss.ss	5-32	8
TODSTAMPw.	Lê 8-byte de stamp de horário-de-dia		8
YYMMDDw.	Lê valores de data na forma yymmdd ou yyyyymmdd	6-32	8
YYMMNw.	Lê valores de data na forma yyyymm ou yymm	4-6	4
YYQw.	Lê quartos de ano	4-32	4

Obs.: Outros formatos estão disponíveis – ver manuais do SAS

8.3 PROC FORMAT

O procedimento PROC FORMAT é um método simples de criar formatos personalizados.

Sintaxe

```
PROC FORMAT <OPÇÕES>;
EXCLUDE ENTRADAS(S);
INVALUE <$>NOME <(OPÇÕES-DE-INFORMAT)> AMPLITUDE(S);
PICTURE NOME <(OPÇÕES)> AMPLITUDE-1
<(PICTURE_1-OPÇÕES)> <...AMPLITUDE-N <(PICTURE_N-OPÇÕES)>>;

SELECT ENTRADA(S);

VALUE <$>NOME <(OPÇÕES)> AMPLITUDE(S);
```

Opções do procedimento PROC FORMAT:

CNTLIN= – Especifica um arquivo SAS do qual PROC FORMAT constrói um formato de entrada ou de saída.

CNTLOUT= – Cria um arquivo SAS que armazena informação sobre formatos de entrada e saída.

FMTLIB – Imprime informação sobre formatos.

LIBRARY= – Especifica um catálogo SAS que conterá formatos criados pelo usuário com um passo de PROC FORMAT.

MAXLABELN= – Especifica o número de caracteres do valor formatado que aparecerá no output do PROC FORMAT.

MAXSELEN= – Especifica o número de caracteres de valor e final que aparece no output do PROC FORMAT.

NOREPLACE – Impede um novo formato de repassar um existente com mesmo nome.

PAGE – Imprime informação sobre cada formato em uma página separada.

Opções comuns das instruções de Informat e de Format:

A seguir são descritas as opções válidas com as instruções INVALUE, PICTURE e VALUE, que são colocadas entre parênteses após um nome de informat ou format.

DEFAULT=comprimento – Especifica um comprimento padrão para o formato.

FUZZ=fatorfrisado – Especifica um fator frisado para emparelhar valores a uma amplitude.

MAX=comprimento – Especifica um comprimento máximo para o formato. Padrão=40.

MIN=comprimento – Especifica um comprimento mínimo para o formato. Padrão=1.

NOTSORTED – Armazena os valores na ordem especificada pelo usuário, sem tentar ordená-los alfabeticamente ou numericamente.

MULTILABEL - Permite múltiplos valores para uma dada amplitude ou para sobrepor amplitudes.

Especificando valores e amplitudes

A especificação de amplitudes de formatos é obrigatória.

As instruções INVALUE, PICTURE, and VALUE acitam valores numéricos do lado esquerdo do sinal de igualdade. As instruções INVALUE e VALUE aceitam caracteres também.

Pode haver múltiplas amplitudes ou valores em cada informação de amplitude separadas por vírgulas. Cada valor ou amplitude é um dos seguintes:

- Valor – um valor simples como 12 ou 'BR';
- Amplitude - é uma lista de valores como 10-20 ou 'A'-'Z'.

É possível utilizar as cláusulas LOW ou HIGH como um valor em uma amplitude ou o símbolo < para excluir valores menores ou maiores que o especificado. Exemplos: a amplitude 0<-100 não inclui o valor zero, enquanto 0-<100 não inclui o valor 100; em 'AA'-'<AJ'=1 'AJ'-'AZ'=2 AJ não pertence à primeira amplitude (1), mas pertence à amplitude 2.

8.3.1 VALUE (INSTRUÇÃO)

Cria um formato que especifica cordões de caracteres a usar para imprimir valores de variáveis.

Sintaxe

■ **VALUE** <\$>**NOME** <(FORMAT-OPÇÕES)> <AMPLITUDE(S)>;

Opções - FUZZ= fuzz-factor, MAX=length, MIN=length, NOTSORTED, MULTILABEL.

8.3.2 INVALUE (INSTRUÇÃO)

Cria um formato de entrada para leitura e conversão de linhas de valores de dados.

Sintaxe

■ **INVALUE** <\$>**NOME** <(INFORMAT-OPÇÕES)> <AMPLITUDE(S)>;

Nome – Denominação do formato que está sendo criado.

Amplitude(s) – Especifica uma linha de dados e valores em que a linha de dados irá se transformar. Pode ser um ou mais dos seguintes: valor-ou-amplitude-1 <..., valor-ou-amplitude-n>= valor-não-formatado [[informat-existente]. O informat converte a linha de dados para os valores-não-formatados à direita do sinal de igualdade.

Valor-não-formatado – É o valor em que se deseja que a linha de valor-ou-amplitude se transforme. Usa-se uma das seguintes formas para valores não formatados: 'Série-de-caracteres', número.

ERROR – Cria valores designados como inválidos.

SAME – Impede que o informat de converter a linha de dados como qualquer outro valor.

Informat-existente – É um formato suprido pelo SAS ou previamente definido pelo usuário.

Opções - DEFAULT= , FUZZ= , MAX= , MIN= , NOTSORTED , JUST , UPCASE.

JUST – Alinha à esquerda todos os cordões de caracteres antes de os comparar às amplitudes.

UPCASE – Converte todas as linhas de valores para maiúsculo antes de os comparar com possíveis amplitudes.

8.3.3 SELECT (INSTRUÇÃO)

Seleciona entradas de processamento pelas opções FMTLIB e CNTLOUT=.

Sintaxe

■ SELECT ENTRADA(S);

Entrada(s) – Especifica um ou mais entradas para processar.

São usadas as seguintes regras especificando entradas com a instrução SELECT:

- Nomes de entradas que contém caracteres são precedidos por um cifrão (\$).
- Nomes de entradas que contém caracteres são precedidos por um arroba (@).
- Nomes de entradas que contém informats de caracteres são precedidos por um arroba e um cifrão (@\$).

8.3.4 PICTURE (INSTRUÇÃO)

Cria um modelo para imprimir números.

Sintaxe

■ PICTURE NOME <(FORMAT-OPÇÕES)> <AMPLITUDE-1 <(PICTURE-1-OPÇÕES)> <...AMPLITUDE-N <(PICTURE-N-OPÇÕES)>>>;

Nome – É o nome do formato que está sendo criado.

Opções – DEFAULT= , FUZZ= , MAX= , MIN= , MULTILABEL, NOTSORTED, ROUND, FILL= , MULTIPLIER= , NOEDIT, PREFIX= .

ROUND – Arredonda o valor para o inteiro mais próximo antes de formatar.

FILL= – Especifica um caractere que completa o valor formatado.

MULTIPLIER= – Especifica um número para multiplicar o valor da variável antes de o formatar.

NOEDIT – Especifica que números são mensagens em vez de dígitos seletores.

PREFIX= – Especifica um caractere de prefixo para o valor formatado.

Outros argumentos da instrução - DATATYPE=DATE | TIME | DATETIME , DECSEP='character' , DIG3SEP='character' .

8.3.5 EXCLUDE (INSTRUÇÃO)

Exclui entradas do processamento pelas opções FMMLIB e CNTLOUT=.

Sintaxe

■ EXCLUDE ENTRADA(S);

Entrada(s) – Especifica um ou mais catálogos de entrada para excluir do processamento.

São usadas as seguintes regras para especificar entradas na instrução EXCLUDE:

- Nomes de entradas que contém caracteres são precedidos por um cifrão (\$).
- Nomes de entradas que contém caracteres são precedidos por um arroba (@).
- Nomes de entradas que contém informats de caracteres são precedidos por um arroba e um cifrão (@\$).

8.4 FORMATANDO DADOS

8.4.1 COMANDO ROUND

Usado como declaração do PROC PRINT, força a impressão de duas casas decimais para todas as variáveis numéricas com mais de duas casas decimais.

8.4.2 COMANDOS TITLE, LENGTH, FORMAT E NOOBS

Devem estar após ao comando Data e antes da entrada ou leitura de dados na primeira vez que o arquivo de dados que contém a variável é declarado.

Exemplo:

```
* EXEMPLO - COMANDOS TITLE, LENGTH, FORMAT E NOOBS;
DATA ARQ;
LENGTH ESPECIE $14.;
FORMAT D 6.2;
INPUT BLOCO 1-2 ESPECIE $ 4-17 D 19-25;
CARDS;
01 CANELA PRETA 19.8347
01 CANELA AMARELA 12.7775
02 CANELA PRETA 29.3098
02 CANELA AMARELA 16.5734
;
PROC PRINT DATA=ARQ NOOBS ROUND;
TITLE1 "EXEMPLO DE FORMATAÇÃO DE DADOS";
TITLE2 "COMANDOS TITLE, LENGTH, FORMAT E NOOBS";
RUN;
QUIT;
```

8.5 MAIS EXEMPLOS

```
/* SAS SAMPLE LIBRARY */
/* NAME: BPG18R01 */
/* REF: SAS Procedures Guide, CHAPTER 18 */
options ls=132;
proc format;
picture phonenum other='000/000-0000';
picture fax other='0999)999-9999' (prefix='(');
run;

data a;
input phone fx;
put phone phonenum;
format fx fax.;
```

```
cards;
9196778000 3332211111
9195551212 5556677777
;
run;

proc format;
  picture phonenum other='000/000-0000';
  picture fax other='0999)999-9999' (prefix='(');
data a;
  input phone fx;
  put phone phonenum;
  format fx fax.;
  cards;
9196778000 3332211111
9195551212 5556677777
;

proc print data=a;
  format phone phonenum;
  title "Formatting in the PROC PRINT Step";
run;

proc format;
  invalue grade 'A'=4 'B'=3 'C'=2 'D'=1 'F'=0;
data grades;
  input nome $ (course1-course3) (: grade.);
  gpa = mean(of course1-course3);
  cards;
BILL  A B A
JIM   A B B
RICK  B C D
ROBERT D . F
;

proc print data=grades;
  title "Reading Data with an Informat";
run;

proc format;
  picture miles 1-99='000000'
  100-high='>100 MILES' (noedit);
data temp;
  input nome $ distance 3.;
  cards;
JOHN 300
MARY 600
DAVID 27
ANN 2
;

proc print data=temp;
  format distance miles.;
  title "NOEDIT Option";
run;

data test;
a=123.45;
b=123.45;
c=12345;
```

```

    d=12345;
run;

proc format;
    picture mwdec low-high='00000.0' (mult=.1);
    picture mnodec low-high='00000.0' (mult=10);
run;

data temp;
    input cents;
    cards;
4123
2130
7250
5379
;

proc format;
    picture dolls low-high='009' (mult=.01);
run;

proc print;
    format cents dolls.;
run;

proc format;
    picture feet other='000000009' (mult=5280);
run;

data feet;
    input miles @@;
    format miles feet.;
    cards;
1 1.5 2
;
run;

proc print;
run;

proc format fmlib;
title "FMTLIB Option";
    value zipne
    01000-02799='Massachusetts'
    02800-02999='Rhode Island' 03000-03899='New Hampshire'
    03900-04999='Maine' 05000-05999='Vermont'
    06000-06999='Connecticut' 07000-08999='New Jersey'
    09000-14999='New York' 15000-19699='Pennsylvania'
    19700-19999='Delaware'
;
title "PAGE Option with Picture Formats";

proc format page;
    picture prot low- -1E5='-OVERFLOW' (noedit)
        -99999.99-<0='000,009.99' (prefix='- ' fill= '*')
        0-999999.99='000,009.99' (fill= '*')
        1E6-high='OVERFLOW' (noedit);
    picture europe low-0='00.009,00' (prefix='- ' mult=100)
        0-high='00.009,00' (mult=100);

proc format;
    /* ZIPST: converts ZIP codes to state nomes */
    value zipst

```

```

00600-00999='Puerto Rico'   01000-02799='Massachusetts'
02800-02999='Rhode Island'   03000-03899='New Hampshire'
03900-04999='Maine'         05000-05999='Vermont'
06000-06999='Connecticut'   07000-08999='New Jersey'
09000-14999='New York'      15000-19699='Pennsylvania'
19700-19999='Delaware'      20000-20599='District of Columbia'
20600-21999='Maryland'      22000-24699='Virginia'
24700-26899='West Virginia' 27000-28999='North Carolina'
29000-29999='South Carolina' 30000-31999='Georgia'
32000-34999='Florida'       35000-36999='Alabama'
37000-38599='Tennessee'     38600-39799='Mississippi'
40000-42799='Kentucky'      43000-45899='Ohio'
46000-47999='Indiana'       48000-49999='Michigan'
50000-52899='Iowa'          53000-54999='Wisconsin'
55000-56799='Minnesota'     57000-57799='South Dakota'
58000-58899='North Dakota'  59000-59999='Montana'
60000-62999='Illinois'      63000-65899='Missouri'
66000-67999='Kansas'        68000-69399='Nebraska'
70000-71499='Louisiana'     71600-72999='Arkansas'
73000-74999='Oklahoma'      75000-79999='Texas'
80000-81699='Colorado'      82000-83199='Wyoming'
83200-83899='Idaho'         84000-84799='Utah'
85000-86599='Arizona'       87000-88499='New Mexico'
89000-89899='Nevada'        90000-96699='California'
96700-96899='Hawaii'        96900-96999='Guam'
97000-97999='Oregon'        98000-99499='Washington'
99500-99999='Alaska'

```

```

;
/* STATE: converts state abbrev. to state nomes */
value $state
'AL'='Alabama'           'NB'='Nebraska'
'AK'='Alaska'           'NV'='Nevada'
'AZ'='Arizona'          'NH'='New Hampshire'
'AR'='Arkansas'         'NJ'='New Jersey'
'CA'='California'       'NM'='New Mexico'
'CO'='Colorado'         'NY'='New York'
'CT'='Connecticut'     'NC'='North Carolina'
'DE'='Delaware'        'ND'='North Dakota'
'DC'='District of Columbia' 'OH'='Ohio'
'FL'='Florida'          'OK'='Oklahoma'
'GA'='Georgia'          'OR'='Oregon'
'HI'='Hawaii'           'PA'='Pennsylvania'
'ID'='Idaho'            'RI'='Rhode Island'
'IL'='Illinois'         'SC'='South Carolina'
'IN'='Indiana'          'SD'='South Dakota'
'IA'='Iowa'             'TN'='Tennessee'
'KS'='Kansas'           'TX'='Texas'
'KY'='Kentucky'        'UT'='Utah'
'LA'='Louisiana'       'VT'='Vermont'
'ME'='Maine'           'VA'='Virginia'
'MD'='Maryland'        'WA'='Washington'
'MA'='Massachusetts'   'WV'='West Virginia'
'MI'='Michigan'         'WI'='Wisconsin'
'MN'='Minnesota'       'WY'='Wyoming'
'MS'='Mississippi'     'RQ'='Puerto Rico'
'MO'='Missouri'        'GQ'='Guam'
'MT'='Montana'         '99'='Foreign'

```

```

;
/* STATE: official census bureau codes for state */
value state
01='Alabama'           30='Montana'
02='Alaska'            31='Nebraska'
04='Arizona'           32='Nevada'

```

```

05='Arkansas'    33='New Hampshire'
06='California'  34='New Jersey'
08='Colorado'    35='New Mexico'
09='Connecticut' 36='New York'
10='Delaware'    37='North Carolina'
11='D.C.'        38='North Dakota'
12='Florida'     39='Ohio'
13='Georgia'     40='Oklahoma'
15='Hawaii'      41='Oregon'
16='Idaho'       42='Pennsylvania'
17='Illinois'    44='Rhode Island'
18='Indiana'     45='South Carolina'
19='Iowa'        46='South Dakota'
20='Kansas'      47='Tennessee'
21='Kentucky'   48='Texas'
22='Louisiana'  49='Utah'
23='Maine'       50='Vermont'
24='Maryland'   51='Virginia'
25='Massachusetts' 53='Washington'
26='Michigan'   54='West Virginia'
27='Minnesota' 55='Wisconsin'
28='Mississippi' 56='Wyoming'
29='Missouri'
;

data new;
input zip abbrev $ censusbu;
zipst=put(zip,zipst.);
cstate=put(abbrev,$state.);
nstate=put(censusbu,state.);
cards;
27512 NC 37
;

proc print;
var zip zipst abbrev cstate censusbu nstate;
title "Formatted State Codes and Abbreviations";
run;

proc format;
picture acct low-<0='000,009.99)' (prefix='(')
0-high='000,009.99 ' ;
picture prot low- -1E5='-OVERFLOW' (noedit)
-99999.99-<0='000,009.99' (prefix='-' fill='*')
0-999999.99='000,009.99' (fill='*')
1E6-high='OVERFLOW' (noedit);
picture dol low-<0='000,009.99' (prefix='$-')
0-high='000,009.99' (prefix='$');
picture rsign low-<0='000,009.99-'
0-high='000,009.00+';
picture credit low-<0='00,009.99DR'
0-high='00,009.99CR';
picture europe low-<0='00.009,00' (prefix='- ' mult=100)
0-high='00.009,00' (mult=100);
picture blank low-<0='000 009.99' (prefix='- ')
0-high='000 009.99';
picture thous 0-high='00,009K' (mult=.001);
picture phone other='000/000-0000';
run;

data a;
input x phone;
acct=x; prot=x; dol=x; rsign=x; credit=x; europe=x; blank=x; thou=x;

```

```

format acct acct. prot prot. dol dol. rsign rsign.
credit credit. europe europe. blank blank. thou thous.
phone phone. X 12.2;
cards;
12345 9196778000
0 6778000
-12345 .
-187.65 9196778000
187.65 .
.23 .
101.23 .
1.1E6 .
;
proc print data=a;
  id x;
  var acct prot dol rsign credit europe blank thou phone;
  title 'Print the Formats Created with PROC FORMAT';
run;

/*-----Create informats for grade scales and for sex-----*/
proc format;
  invalue grade 'A+'=4.0 'A'=3.5 'A-'=3.2
               'B+'=3.0 'B'=2.5 'B-'=2.2
               'C+'=2.0 'C'=1.5 'C-'=1.2
               'D+'=1.0 'D'=0.5 'D-'=0.2
               'E'=0 'I'=. ;
  invalue sex 'M'=1 'F'=2;
run;

/*-----Read in the student nome data-----*/
data students;
  input id sex : sex. nome $;
  cards;
003 F Jane
005 F Mary
001 M John
002 F Robin
004 M Rick
;
run;

/*-----Sort to be in order by ID-----*/
proc sort;
  by id;
run;

/*-----Input the grade information-----*/
data grades;
  input id nclasses @;
  do i=1 to nclasses;
    input grade: grade. @;
    output;
  end;
  keep id grade;
  cards;
003 5 A B+ B C+ A-
002 5 B+ I C C E
001 4 A B C- B-
005 5 A A A A+ A-
004 4 B- D E C
;
run;

```

```

/*-----Sort to be in order by ID-----*/
proc sort data=grades;
  by id;
run;

/*-----Determine the GPAs for the students-----*/
proc means data=grades noprint;
  by id;
  var grade;
  output out=final(drop=_type_) mean=gpa;
run;

/*-----Merge GPA into student info-----*/
data students;
  merge students final;
  by id;
run;

/*-----Sort to print by SEX-----*/
proc sort data=students;
  by sex;

/*-----Print the final results-----*/
proc print data=students;
  by sex;
  title 'Students' Nomes and GPAs';
run;

data acctinfo;
  input acctnum nome $15. @26 opendate date7.;
  cards;
5008074 John Smith    01JAN87
5008075 Bill Jones    10JAN87
5009766 Benjamin Estes 18OCT86
7089477 Mary Wilson   27NOV86
;

/*-----First create an input control data set-----*/
/*-----First create an input control data set-----*/
data cntlacct(rename=(acctnum=start nome=label));
  set acctinfo(keep=acctnum nome);
  fmtnome='account';
run;

proc print data=cntlacct;
  title "Input Control Data Set";
run;

/*-----Read the control data set into PROC FORMAT-----*/
proc format cntlin=cntlacct;
run;

/*-----The format is now created and ready to use-----*/
data charges;
  input acctnum transamt;
  put acctnum account. ' account charged ' transamt dollar10.2;
  cards;
5008074 127.86
7089477 100.00
5009766 50.00
;

proc format cntlout=outdata;

```

```
picture phonenum other='000/000-0000';  
invalue grade 'a'=4 'b'=3 'c'=2 'd'=1 'e'=0;  
run;  
  
proc print data=outdata;  
title 'Output Control Data Set';  
run;
```

Capítulo 9 : Funções

Este capítulo é uma versão resumida dos capítulos sobre funções dos manuais disponíveis no site de documentação do software em <<http://support.sas.com/documentation/>>.

Funções são operações executadas pelo SAS que possuem um nome próprio. Uma função SAS, a partir do fornecimento de um argumento, retorna um valor. O SAS possui um elenco de funções em diferentes áreas, desde as mais comuns, como as funções aritméticas, até funções complexas como as econômicas e as de procura de parte de uma série de caracteres. Muitas funções necessitam de argumentos fornecidos pelo usuário, enquanto algumas tem argumentos fornecidos pelo ambiente operacional.

Sintaxe

Há três formas de sintaxe de funções, conforme segue:

```

FUNÇÃO NOME ( ARGUMENTO-1 <... , ARGUMENTO-N > )
FUNÇÃO NOME ( OF LISTA DE VARIÁVEIS )
FUNÇÃO NOME ( OF ARRAY-NOME { * } )

```

Função-nome – É o nome da função.

Argumento – É o argumento repassado pelo usuário ou pelo ambiente operacional. Pode ser o nome de uma variável, uma constante, ou qualquer expressão SAS, inclusive outra função. Argumentos múltiplos são separados por vírgulas. Exemplos:

```

x=max(avista,crédito);
x=sqrt(1500);
novacidade=left(uppercase(cidade));
x=min(temperatura-jul,temperatura-dez);
s=repeat('----+',16);
x=min((registrar-baixa),(registrar-falha));
valor=int(avista);
if sum(avista,crédito)>1000 then Put 'meta atingida';

```

OF listadevariáveis – Lista de quaisquer tipos de variáveis SAS separadas por espaços em branco.

Exemplos:

```

a=sum(of x y z);
z=sum(of y1-y10);
* os dois exemplos a seguir são equivalentes;
a=sum(of x1-x10 y1-y10 z1-z10);
A=sum(of x1-x10, of y1-y10, of z1-z10);

```

OF array-nome{*} – Nome de um vetor definido (ARRAY) que é tratado como uma lista de variáveis em vez de processar só um elemento do vetor de cada vez. Exemplo:

```

data arq; * cria o arquivo de dados arq;

```

1
2

```

array y{10} y1-y10; * declaração do vetor y com 10 elementos      3
    iniciando com y1 e terminando com y10;                        4
input y1-y10; * formato de entrada de dados;                      5
X=sum(of y{*}); * função que soma os elementos do vetor em x;    6
datalines; * chama as linhas de dados;                            7
3 6 2 6 5 2 4 7 3 6                                             8
2 4 7 9 4 2 5 6 8 3                                             9
;                                                                    10
proc print; * imprime o arquivo resultante;                       11
run; * executa o programa;

```

O programa de exemplo de função de soma com um vetor imprime o seguinte relatório:

<i>The SAS System</i>											
Obs	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	X
1	3	6	2	6	5	2	4	7	3	6	44
2	2	4	7	9	4	2	5	6	8	3	50

9.1 RESULTADOS DE FUNÇÕES

Os resultados ou variáveis alvo de funções SAS são usualmente de caracteres se os argumentos são de caracteres, ou numéricos se os argumentos são numéricos. A exceção a esta regra é a função PUT, porque seus resultados são de caracteres independentemente do tipo de argumento.

O comprimento padrão para a maioria dos alvos numéricos é de 8 dígitos e para as variáveis de caracteres é de 200. Há exceções. Este comprimento padrão não se aplica, por exemplo, às funções de comprimento: BYTE (1), COMPRESS (comprimento do primeiro argumento), INPUT (largura do informante numérico 8), LEFT (comprimento do argumento), PUT (largura do formato), REVERSE (comprimento do argumento), RIGHT (comprimento do argumento), SCAN (200), SUBSTR (comprimento do primeiro argumento), TRANSLATE (comprimento do primeiro argumento), TRIM (comprimento do argumento), UPCASE (comprimento do argumento).

9.2 CATEGORIAS DE FUNÇÕES

As funções SAS são organizadas nas seguintes categorias:

• Medida da Resposta da aplicação (ARM)	Application response measurement (ARM)
• Funções aritméticas	Arithmetic functions
• Funções lógicas bitwise	Bitwise logical functions
• Funções de caractere	Character functions
• Série de caracteres que emparelham funções	Character string matching functions
• Conversão de moeda	Currency conversion
• Funções de data_e_horário	Date and Time Functions
• Funções DBCS	DBCS functions
• Biblioteca de ligação dinâmica	Dynamic link library
• Funções de arquivo externo	External file functions
• Funções financeiras	Financial functions
• Biblioteca e funções de catálogo	Library and Catalog Functions
• Funções matemáticas	Mathematical functions
• Funções de Probabilidade e Densidade	Probability and Density Functions
• Funções quantís	Quantile functions
• Funções de números aleatórios	Random number functions
• Funções estatísticas amostrais	Sample statistic functions
• Funções de entrada e saída de arquivos SAS	SAS file i/o functions
• Funções especiais	Special functions
• Funções de Estado e CEP	State and Zip Code Functions
• Funções trigonométricas	Trigonometric functions
• Funções de redução por corte	Truncation functions
• Funções de informação de variável	Variável information functions
• Funções de não-centralidade	Noncentrality functions
• Ferramentas da Web	Web tools

9.2.1 FUNÇÕES ARITMÉTICAS

ABS(argumento) – retorna o valor absoluto.

DIM<n>(array-nome) – retorna o número de elementos de um vetor unidimensional ou o número de elementos de uma dimensão especificada de um vetor multidimensional.

n – Especifica a dimensão em um vetor multidimensional do qual se deseja saber o número de elementos.

DIM(array-nome,bound-n) – Retorna o número de elementos em um vetor unidimensional, ou o número de elementos de uma dimensão especificada de um vetor multidimensional.

bound-n – Especifica a dimensão de um vetor multidimensional do qual se deseja conhecer o número de elementos.

HBOUND<n>(array-nome) – retorna o limite superior de um vetor.

HBOUND(array-nome,bound-n) – retorna o limite superior de um vetor.

LBOUND<n>(array-nome) – retorna o limite inferior de um vetor.

LBOUND(array-nome,bound-n) – retorna o limite inferior de um vetor.

MAX(argumento,argumento, ...) – Retorna o maior valor de argumentos numéricos.

MIN(argumento,argumento, ...) – Retorna o menor valor de argumentos numéricos.

MOD(argumento-1, argumento-2) – Retorna o restante.

SIGN(argumento) – Retorna o sinal do valor ou 0 para valores nulos.

SQRT(argumento) – Retorna a raiz quadrada.

9.2.2 FUNÇÕES DE CARACTERE

BYTE(n) – Retorna um caractere em ASCII or EBCDIC onde n é um inteiro representando um caractere ASCII ou EBCDIC.

COLLATE(start-position<,end-position>) | (start-position<,,length>) – Retorna uma sequência de caracteres ASCII or EBCDIC .

COMPBL(source) – Remove múltiplos brancos entre palavras de uma série de caracteres.

COMPRESS(source<,caracteres-to-remove>) – Remove caracteres específicos de uma série de caracteres.

DEQUOTE(argumento) – Remove aspas de um valor de caractere.

INDEX(source,excerpt) – Procura a fonte de uma série de caracteres especificada por excerpt .

INDEXC(source,excerpt-1<, ... excerpt-n>) – Procura a fonte para qualquer caractere presente no excerpt .

INDEXW(source,excerpt) – Procura a fonte para um padrão especificado como uma palavra.

LEFT(argumento) – Alinha uma série de caracteres à esquerda.

LENGTH(argumento) – Retorna o comprimento de um argumento.

LOWCASE(argumento) – Converte todas as letras do argumento em minúsculas.

QUOTE(argumento) – Adiciona aspas duplas a um valor de caractere.

RANK(x) – Retorna a posição de uma sequência de caracteres ASCII ou EBCDIC.

REPEAT(argumento,n) – Repete uma expressão de caracteres.

REVERSE(argumento) – Reverte uma expressão de caracteres.

RIGHT(argumento) – Alinha à direita uma expressão de caracteres.

SCAN(argumento,n<,delimiters>) – Retorna uma dada palavra de uma expressão de caracteres.

SOUNDEX(argumento) – Codifica uma série de caracteres para facilitar a busca.

SUBSTR(argumento,position<,n>)=caracteres-a-repassar – Repassa o valor do conteúdo na posição n indicada.

var=SUBSTR(argumento,position<,n>) – Extrai uma subsérie de caracteres de um argumento. (var é qualquer nome de variável SAS válido.)

TRANSLATE(source,to-1,from-1<,,...to-n,from-n>) – Repassa caracteres específicos em uma expressão de caracteres.

TRANWRD(source,target,replacement) – Repassa ou remove todas as ocorrências de uma palavra em uma série de caracteres.

TRIM(argumento) – Remove espaços brancos de uma expressão de caracteres e retorna um branco se o valor é perdido.

TRIMN(argumento) – Remove espaços brancos de expressões de caracteres e retorna uma série de caracteres nulos se a expressão é perdida.

UPCASE(argumento) – Converte todas as letras do argumento em maiúsculas.

VERIFY(source,excerpt-1<,...excerpt-n) – Retorna a posição do primeiro caractere único para uma expressão.

9.2.3 FUNÇÕES DE DATA E HORÁRIO

DATDIF(sdata,edata,basis) – Retorna o número de dias entre duas datas.

DATA() – Retorna a data corrente como um valor de data SAS.

DATAJUL(julian-data) – Converte um valor de data SAS para o calendário Juliano.

DATAPART(data-horário) – Extrai a data de um valor de data-horário.

DATETIME() – Retorna a data e o horário atuais.

DAY(data) – Retorna o dia do mês de um valor de data SAS.

DHMS(data,hora,minuto,segundo) – Retorna um valor de data-horário SAS de data, horas, minutos, segundos.

HMS(hora,minuto,segundo) – Retorna um valor de horário SAS de horas, minutos, segundos.

HOUR(<horário | data-horário>) – Retorna a hora de um valor de data-horário SAS.

INTCK('interval',from,to) – Retorna o número de intervalos de tempo em um dado espaço de tempo.

INTNX('interval',start-from,increment<,'Alinhamento'>) – Avança um valor de data, horário, ou data-horário por um dado intervalo e retorna a data, horário ou data-horário ajustado.

JULDATE(data) – Retorna a data Juliana de um valor de data SAS.

MDY(mês,dia,ano) – Retorna um valor de data SAS de valores de mês, dia, ano.

MINUTE(horário | data-horário) – Retorna o minuto de um valor de horário ou data-horário SAS.

MONTH(data) – Retorna o mês de um valor de data SAS.

QTR(data) – Retorna o quarto do ano de um valor de data SAS

SECOND(horário | data-horário) – Retorna o segundo de um valor de horário ou data-horário SAS.

HORÁRIO() – Retorna a hora corrente do dia.

HORÁRIOPART(data-horário) – Extrai um valor de horário de um valor de data-horário SAS.

TODIA() – Retorna a data corrente como um valor de data SAS .

WEEKDIA(data) – Retorna o dia da semana de um valor de data SAS.

ANO(data) – Retorna o ano de um valor de data SAS.

YRDIF(sdata,edata,basis) – Retorna a diferença em anos entre duas datas.

YYQ(ano,quarter) – Retorna o valor de data SAS de ano e quartos.

9.2.4 FUNÇÕES FINANCEIRAS

As funções financeiras são descritas, a seguir, por tipo:

FUNÇÕES DE FLUXO DE CAIXA:

CONVX, CONVXP – Calcula convexidade para fluxos de caixa.

DUR, DURP – Calcula a duração modificada para fluxos de caixa.

PVP, YIELDP – Calcula valor presente e rendimento-para-maturidade para um fluxo de caixa periódico.

9.2.4.1 FUNÇÕES DE CÁLCULOS DE PARÂMETROS:

COMPOUND – Calcula combinação de parâmetros de interesse.

MORT – Calcula parâmetros de amortização.

Funções de taxa interna de retorno:

INTRR, IRR – Calcula a taxa interna de retorno,

Funções de valor líquido presente e futuro:

NETPV, NPV – Calcula valores líquidos presente e futuro.

SAVING – Calcula o valor futuro de poupança periódica.

Funções de depreciação:

DACCxx – Calcula a depreciação acumulada para cima para o período especificado.

DEPxxx – Calcula a depreciação para um único período.

O argumento de período para funções de depreciação pode ser fracionário para todas as funções, exceto DEPDBSL e DACCDBSL. Para argumentos fracionários, a depreciação é rateada entre os dois períodos de tempo sucessivos que precede e que segue o período fracionário. PRECAUÇÃO: Verifique o método de depreciação para períodos fracionários. É necessário verificar se o método é apropriado para usar com períodos fracionários porque muitos planos de depreciação, especificados como tabelas, têm regras especiais para períodos fracionários.

Sintaxe e descrição das funções financeiras:

COMPOUND(montante,futuro,taxa,número) – Retorna a combinação de parâmetros de interesse.

CONVX(y,f,c(1),...,C(k)) – Retorna a convexidade para um determinado fluxo de caixa.

CONVXP(A,C,n,K,k0,y) – Retorna a convexidade para uma sequência de fluxo de caixa, como um bônus.

DACCDB(período,valor,anos,taxa) – Retorna a amortização de equilíbrio da depreciação acumulada.

DACCDBSL(período, valor,anos,taxa) – Retorna a amortização de equilíbrio da depreciação com conversão para uma depreciação em linha reta.

DACCSSL(período, valor,anos) – Retorna a linha reta de depreciação acumulada.

DACCSD(período, valor,anos) – Retorna a depreciação da soma acumulada dos dígitos dos anos.

DACCTAB(período,valor,t1,...,tn) – Retorna a depreciação acumulada de tabelas especificadas, em que t1, ...tn são frações numéricas de depreciação para cada período de tempo.

DEPDB(período,valor,ano,taxa) – Retorna a amortização de equilíbrio da depreciação.

DEPDBSL(período,valor,anos,taxa) – Retorna amortização de equilíbrio com conversão para uma linha reta de depreciação.

DEPSL(período,valor,anos) – Retorna a linha reta de depreciação.

DEPSYD(período,valor,anos) – Retorna a depreciação da soma dos dígitos dos anos.

DEPTAB(período,valor,t1,...,tn) – Retorna a depreciação de tabelas especificadas, t1,...,tn.

DUR(y,f,c(1),...c(k)) – Retorna a duração modificada para um fluxo de caixa enumetaxado.

DURP(A,c,n,K,k0,y) – Retorna a duração modificada para um fluxo de caixa periódico, como um bônus.

INTRR(frequência,c0,c1,...,cn) – Retorna a taxa interna de retorno como uma fração.

IRR(frequência,c0,c1,...,cn) – Retorna a taxa interna de retorno como uma porcentagem. Frequência é o número de pagamentos sobre um período base especificado que é associado a uma taxa de retorno desejada, com amplitude maior que zero, exceto no caso em que freq = 0 é uma bandeira para permitir composição contínua. c0,c1, ..., cn são os pagamentos opcionais. A função INTRR retorna a taxa interna de retorno sobre um período base especificado para o conjunto de pagamentos c0, c1, ..., cn. O intervalo

de tempo entre dois pagamentos é assumido pelo SAS como sendo igual. O argumento freq>0 descreve o número de pagamentos que ocorrem sobre o período base especificado. A taxa interna de retorno é a taxa de interesse sobre a sequência de pagamentos que tem um Valor Líquido Presente (VPL) igual a zero (ver a função NETPV). A taxa interna de retorno é dada por:

$$r = \begin{cases} \frac{1}{x^{freq}} & freq > 0 \\ -\log_e(x) & freq = 0 \end{cases}$$

Em que: x é a raiz real, infinitamente próxima de 1, da polinomial: $\sum_{i=0}^n c_i \cdot x^i = 0$.

A rotina de IRR usa o método de Newton para encontrar a taxa interna de retorno infinitamente próxima de 0. Dependendo do valor dos pagamentos, pode não existir uma raiz para a equação; neste caso, retorna um valor perdido (.). Valores perdidos nos valores de pagamentos são tratados como zero(0). Quando freq>0, a taxa de retorno computada é a taxa efetiva sobre o período base especificado. Para computar trimestralmente a taxa interna de retorno com pagamentos mensais faça freq igual a 3. Se freq é 0, é assumida composição contínua e o período base é o intervalo de tempo entre dois pagamentos consecutivos. A taxa interna de retorno computada é a taxa nominal de retorno sobre o período base. Para computar com composição contínua e pagamentos mensais, faça freq igual a zero (0). A taxa interna de retorno será uma taxa mensal. Exemplo: Para um gasto inicial de R\$400,00 e expectativas de pagamentos de R\$100,00, R\$200,00 e R\$300,00 sobre os três anos seguintes, A taxa interna de retorno pode ser expressa em taxa=intrr(1,-400,100,200,300); o valor retornado é 0,19438.

MORT(montante,pagamento,taxa,número) – Retorna parâmetros de amortização.

NETPV(r,freq,c0,c1,...,cn) - Retorna o valor presente líquido como uma fração. NPV é semelhante à NETPV, exceto que r é provido como percentagem na NPV.

Sintaxe:

■ **NETPV(R,FREQ,C0,C1, . . . ,CN)**

r – É a taxa de interesse sobre um período base específico expresso como uma fração.

freq – É o número de pagamentos sobre o período base que é especificado com a taxa r. Amplitude: freq > 0. Excessão: o caso em que freq = 0 é uma marca para permitir desconto contínuo.

c0,c1, . . . ,cn – São fluxos de caixa que representam pagamentos ou receitas durante os momentos 0, 1, ...n. Os momentos são assumidos como igualmente espaçados, no instante inicial do período. Valores negativos representam pagamentos, valores positivos representam receitas e valores 0 (zero) representam nenhum movimento de valor num dado momento. Os argumentos c0 e c1 são requeridos. A função NETPV retorna o valor presente líquido no momento 0 para a série de pagamentos c0,c1, . . . ,cn, com uma taxa r sobre um especificado período de tempo. O argumento freq>0 descreve o número de pagamentos que ocorre sobre o especificado período base de tempo. O Valor Presente Líquido (NETPV) é dado por:

$$NETPV(r, \text{freq}, c_0, c_1, \dots, c_n) = \sum_{i=0}^n c_i \cdot x^i$$

$$\text{Em que: } \frac{1}{(1+r)^{(1/\text{freq})}} \text{ freq} > 0$$

$$e^{-r} \text{ freq} = 0$$

$$x =$$

Valores perdidos em pagamentos são tratados como zero (0). Quando $\text{freq} > 0$, a taxa r é a taxa efetiva sobre o período base especificado. Para especificar uma taxa trimestral de 4% com pagamentos mensais, faça freq igual a 3 e r a .04. Se freq é 0, é assumido desconto contínuo. O período básico é o intervalo de tempo entre dois pagamentos sucessivos e a taxa r é uma taxa nominal. Para computar com uma taxa de interesse anual nominal de 11 por cento descontada continuamente com pagamentos mensais, atribua 0 para freq e 11/12 para r . Exemplo: Para um investimento inicial de \$500 que retorna pagamentos bianuais de \$200, \$300, e \$400 sobre os sucessores 6 anos e uma taxa de desconto anual de 10 por cento, o valor presente líquido do investimento pode ser expresso como $\text{valor} = \text{netpv}(.10, .5, -500, 200, 300, 400)$; o valor resultante é de 95.98.

$NPV(r, \text{freq}, c_0, c_1, \dots, c_n)$ - Retorna o valor presente líquido com uma taxa expressa como uma porcentagem. NPV é semelhante à $NETPV$, exceto que r é provido como porcentagem na NPV .

Sintaxe

■ $NPV(R, \text{FREQ}, C_0, C_1, \dots, C_N)$.

r - É a taxa de interesse, sobre um especificado período básico de tempo, expressa como uma porcentagem.
 freq - É o número de pagamentos durante o especificado período básico de tempo com a taxa r . Amplitude: $\text{freq} > 0$. Exceção: O caso em que $\text{freq} = 0$ é usado como uma bandeira para permitir desconto contínuo.
 c_0, c_1, \dots, c_n - Fluxo de caixa que representa pagamentos ou rendimentos ocorrendo nos tempos 0, 1, ... n. É assumido os períodos do fluxo de caixa são igualmente espaçados. Valores negativos representam pagamentos, valores positivos representam renda, e valores 0 não representam nenhum valor de fluxo monetário em um determinado momento. São requeridos como obrigatórios os argumentos c_0 e c_1 .

$PVP(A, c, n, K, k_0, y)$ - Retorna o valor presente para um fluxo de caixa periódico, como um bônus. Em que: A é o valor de paridade com amplitude > 0 ; c é a taxa de cupom nominal por-período, expressa como uma fração com amplitude de $0 < c < 1$; n é o número de o número de cupons por período com amplitude $n > 0$ e é sempre inteiro; K é o número de cupons remanescentes com amplitude $K > 0$ e é sempre inteiro; k_0 é o período de tempo entre a data presente e a data do primeiro cupom, expressa em termos de número de períodos, com amplitude de $0 < k_0 < 1/n$; y é o por-período nominal de rendimento-para-maturidade, expresso como uma fração, com amplitude $y > 0$. A função PVP é baseada na relação:

$$P = \sum_{k=1}^K c(k) \frac{1}{(1 + \frac{y}{n})^{t_k}}$$

Em que: $t_k = k - (1 - nk_0)$; $c(k) = A(c/n)$ para $k = 1, \dots, K-1$; $c(K) = A(1 + (c/n))$.

Exemplo: $p = \text{pvp}(1000, 1/100, 4, 14, .33/2, .10)$ retorna o valor 743.168.

SAVING(futuro,pagamento,taxa,número) – Retorna o valor futuro de uma poupança periódica.

YIELDP(A,c,n,K,k0,p) – Retorna o rendimento-para-maturidade para um fluxo de caixa periódico, como um bônus.

9.2.5 FUNÇÕES MATEMÁTICAS

AIRY(x) – Retorna o valor da função AIRY.

DAIRY(x) – Retorna a derivada da função AIRY.

DIGAMMA(argumento) – Retorna o valor da função de DIGAMMA

ERF(argumento) – Retorna o valor do (normal) de função de erro.

ERFC(argumento) – Retorna o valor do (normal) de função de erro.

EXP(argumento) – Retorna o valor da função exponencial.

GAMMA(argumento) – Retorna o valor da função GAMA.

IBESSEL(nu,x,kode) – Retorna o valor da função Bessel modificada.

JBESSEL(nu,x) – Retorna o valor da função Bessel.

LGAMMA(argumento) – Retorna o logaritmo natural da função GAMA.

LOG(argumento) – Retorna o logaritmo natural (de base e).

LOG2(argumento) – Retorna o logaritmo de base 2.

LOG10(argumento) – Retorna o logaritmo de base 10.

TRIGAMMA(argumento) – Retorna o valor da função TRIGAMMA.

9.2.6 FUNÇÕES DE NÚMEROS ALEATÓRIOS

NORMAL(seed) – Retorna uma variação aleatória de uma distribuição normal².

RANBIN(seed,n,p) – Retorna uma variação aleatória de uma distribuição binomial.

RANCAU(seed) – Retorna uma variação aleatória de uma distribuição Cauchy.

RAND('dist', parm-1, ..., parm-k) (EXPERIMENTAL) – Retorna uma variação aleatória de uma distribuição especificada. Nota: esta é uma função experimental.

RANEXP(seed) – Retorna uma variação aleatória de uma distribuição exponencial.

RANGAM(seed,a) – Retorna uma variação aleatória de uma distribuição Gamma.

RANNOR(seed) – Retorna uma variação aleatória de uma distribuição normal.

RANPOI(seed,m) – Retorna uma variação aleatória de uma distribuição Poisson.

RANTBL(seed,p1,..pi,..pn) – Retorna uma variação aleatória de uma probabilidade tabelada.

RANTRI(seed,h) – Retorna uma variação aleatória de uma distribuição triangular.

RANUNI(seed) – Retorna uma variação aleatória de uma distribuição uniforme.

UNIFORM(seed) – Retorna um valor pseudo-randômico de uma distribuição uniforme.

9.2.7 FUNÇÕES DE ESTATÍSTICA BÁSICA

CSS(argumento,argumento,...) – Retorna a soma de quadrados corrigidos.

CV(argumento,argumento,...) – Retorna o coeficiente de variação.

² seed=semente, ou valor inicial.

KURTOSIS(argumento,argumento,...) – Retorna a curtose (ou 4º momento).
MAX(argumento,argumento, ...) – Retorna o maior valor.
MIN(argumento,argumento, ...) – Retorna o menor valor.
MEAN(argumento,argumento, ...) – Retorna a média aritmética (média).
MISSING(numeric-expressão | caractere-expressão) – Retorna um resultado numérico que indica se o argumento contém um valor perdido.
N(argumento,argumento, ...) – Retorna o número de valores de não perdidos.
NMIS(argumento,argumento, ...) – Retorna o número de valores perdidos.
ORDINAL(count,argumento,argumento,...) – Retorna o valor mais alto de uma parte de uma lista.
RANGE(argumento,argumento,...) – Retorna a amplitude de valores.
SKEWNESS(argumento,argumento,argumento,...) – Retorna a tendência.
STD(argumento,argumento,...) – Retorna o erro padrão.
STDERR(argumento,argumento,...) – Retorna o erro padrão da média.
SUM(argumento,argumento,...) – Retorna a soma.
USS(argumento,argumento,...) – Retorna a soma de quadrados não corrigidos.
VAR(argumento,argumento,...) – Retorna a variância.

9.2.8 FUNÇÕES TRIGONOMÉTRICAS

ARCOS(argumento) – Retorna o arcocosseno.
ARSIN(argumento) – Retorna o arcosseno.
ATAN(argumento) – Retorna o arcotangente.
COS(argumento) – Retorna o cosseno.
COSH(argumento) – Retorna o cosseno hiperbólico.
SIN(argumento) – Retorna o seno.
SINH(argumento) – Retorna o seno hiperbólico.
TAN(argumento) – Retorna a tangente.
TANH(argumento) – Retorna a tangente hiperbólica.

9.2.9 FUNÇÕES DE TRUNCAGEM

CEIL(argumento) – Retorna o menor inteiro que é maior ou igual ao argumento.
FLOOR(argumento) – Retorna o maior inteiro menor ou igual ao argumento.
FUZZ(argumento) – Retorna inteiro mais próximo se o argumento está dentro de 1E-12.
INT(argumento) – Retorna o valor inteiro do argumento.
ROUND(argumento,round-off-unit) – Arredonda para a unidade mais próxima.
TRUNC(number, length) – Trunca um valor numérico num comprimento especificado.

9.2.10 FUNÇÕES DE HOSPEDAGEM

As funções de hospedagem dizem respeito ao ambiente operacional em que o SAS está instalado.

As funções SAS de hospedagem retornam um valor do ambiente operacional, seja do computador ou sistema operacional. A maioria destas funções utilizam argumentos fornecidos pelo usuário.

A maior parte destas funções são descritas no manual 'SAS Language Reference: Dictionary', disponível em <<https://support.sas.com/documentation/cdl/en/Irdict/64316/PDF/default/Irdict.pdf>>; as principais são as seguintes: BYTE MODULE, COLLATE PEEK, DMYTECHC RANK, DMYTECWD SLEEP, DMYTERVC TRANSLATE, MCIPISLP WAKEUP, MCIPISTR.

Capítulo 10 : PROC ANOVA / PROC GLM

Os dois procedimentos realizam análise da variância. O procedimento PROC ANOVA realiza análise da variância somente de experimentos em delineamentos balanceados, ou seja, que possuam números iguais de observações para toda combinação dos fatores de classificação do delineamento. O procedimento PROC GLM (General Linear Models) efetua análise da variância de modelos lineares em geral, podendo ser utilizado para análise de dados de experimentos, inventários florestais e regressão linear sem a necessidade de haver balanceamento dos dados.

Os procedimentos estatísticos disponíveis são descritos no manual “SAS/STAT User’s Guide” disponível em <<http://support.sas.com/documentation/onlinedoc/stat/142/statug.pdf>>.

O uso do PROC ANOVA é bastante fácil. Desde que os dados sejam balanceados, é suficiente informar as classes em que o experimento é organizado e o modelo a ser utilizado na análise. Entretanto, se for um experimento não balanceado, onde se perdeu parcelas, ou um delineamento de blocos incompletos, o procedimento a ser utilizado é o GLM, que faz o balanceamento automaticamente.

O uso do PROC GLM é mais complexo. O procedimento é mais poderoso que seu correlato, mas exige cuidados especiais na interpretação dos resultados e com as informações sobre o erro a utilizar na análise de diferenças entre os efeitos e quanto ao tipo de erro.

Ambos tem instruções embutidas para testes de médias como os testes de Tukey, Duncan, Scheffe, entre outros, podendo também realizar análise multivariada.

A sintaxe geral de ambos é semelhante, sendo que o GLM tem maior número de opções do que o ANOVA.

10.1 SINTAXE:

```
PROC ANOVA OPÇÕES;  
CLASS VARIÁVEIS;  
MODEL VARIÁVEISDEPENDENTES = EFEITOS / OPÇÕES;  
MEANS EFEITOS </ <LINES> <TESTESDEMÉDIAS> <OPÇÕES>>;  
TEST < H=EFEITOS > E=TERMODEERRO ;  
FREQ VARIÁVEL ;  
REPEATED ESPECIFICAÇÃODEFATOR </ OPÇÕES> ;  
ABSORB VARIÁVEIS;
```

**MANOVA <TESTES> </ OPÇÕES> ;
BY VARIÁVEIS;**

10.2 OPÇÕES DO PROCEDIMENTO:

DATA=ArquivoDeDados – Define o arquivo de dados a utilizar com o procedimento.

MANOVA – Realiza análise multivariada com mais de uma variável dependente eliminando valores perdidos.

NOPRINT – Impede a impressão do relatório; é útil quando se quer somente um arquivo de saída.

ORDER=opção – Determina como os dados estão ordenados; a opção pode ser uma das seguintes: DATA | FORMATTED | FREQ | INTERNAL.

OUTSTAT=NomeDoArquivoDeSaída – Grava um arquivo SAS com os resultados do processamento.

10.3 INSTRUÇÕES DO PROCEDIMENTO:

CLASS Variáveis - Indica as classes de efeitos que devem ser considerados.

MODEL VariáveisDependentes = Efeitos / opções – Informa o modelo a ser utilizado na análise.

Means Efeitos </ <LINES> <TestesDeMédias> <Opções>> - Informa os efeitos sobre os quais devem ser realizados os testes de médias e que testes devem ser realizados.

TEST < H=Efeitos > E=TermoDeErro – Informa um ou mais efeitos que devem ser analisados com um termo de erro que não é o geral do modelo;

FREQ Variável – Informa a variável que deve ser utilizada como frequência das observações, se houver.

REPEATED EspecificaçãoDeFator </ opções> - Informa um fator que deve ser repetido.

ABSORB Variáveis – Informa variáveis de absorção.

MANOVA <Testes> </ Opções> - Informa os testes multivariados que devem ser realizados.

BY Variáveis – Informa variáveis que devem ser usadas para separar a análise por grupos.

O “EXEMPLO - ANOVA DE BLOCOS CASUALIZADOS” é típico para análise da variância de blocos ao acaso.

* EXEMPLO - ANOVA de blocos casualizados; TITLE;	1
DATA ARQUIVO;	2
 INPUT BLOCO TRATAMENTO \$ PRODUCAO VALOR @@;	3
 DATALINES;	4
1 A 32.6 112 1 B 36.4 130 1 C 29.5 106	5
2 A 42.7 139 2 B 47.1 143 2 C 32.9 112	6
3 A 35.3 124 3 B 40.1 134 3 C 33.6 116	7
;	8
PROC ANOVA DATA=ARQUIVO;	9
 CLASS BLOCO TRATAMENTO;	10
 MODEL PRODUCAO VALOR=BLOCO TRATAMENTO;	11
 MEANS TRATAMENTO / LINES TUKEY;	12
RUN;	13
QUIT;	14

Neste exemplo é criado um arquivo com as colunas Bloco, Tratamento, Produção e Valor (Linhas 2 até 8). Em seguida os dados são processados com o procedimento ANOVA (Linha 9), onde as classes são Bloco e Tratamento (Linha 10); as variáveis

independentes (efeitos) do modelo, são as mesmas informadas como classes; e, as variáveis dependentes são “Produção e Valor” (Linha 11). É solicitado um teste de médias (reduzido pela opção lines) com o teste de Tukey (Linha 12). Os dados foram obtidos de exemplos da ajuda do SAS 8.2.

O modelo, neste caso, pode ser representado simplificadaamente como:

$$Y = \text{média} + \text{bloco} + \text{tratamento} + \text{erro},$$

onde: Y é a variável dependente dos efeitos dos Blocos e Tratamentos, caracterizados pelos resultados medidos da Produção e do seu Valor (Linha 11).

O resultado do processamento é o relatório a seguir:

The ANOVA Procedure

Class Level Information		
Class	Levels	Values
Bloco	3	1 2 3
Tratamento	3	A B C

Number of observations	9
------------------------	---

The ANOVA Procedure
Dependent Variable: Producao

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	225.2777778	56.3194444	8.94	0.0283
Error	4	25.1911111	6.2977778		
Corrected Total	8	250.4688889			

R-Square	Coeff Var	Root MSE	Producao Mean
0.899424	6.840047	2.509537	36.68889

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Bloco	2	98.1755556	49.0877778	7.79	0.0417
Tratamento	2	127.1022222	63.5511111	10.09	0.0274

The ANOVA Procedure
Dependent Variable: Valor

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	1247.333333	311.833333	8.28	0.0323
Error	4	150.666667	37.666667		
Corrected Total	8	1398.000000			

R-Square	Coeff Var	Root MSE	Valor Mean
0.892227	4.949450	6.137318	124.0000

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Bloco	2	354.6666667	177.3333333	4.71	0.0889
Tratamento	2	892.6666667	446.3333333	11.85	0.0209

The ANOVA Procedure

Tukey's Studentized Range (HSD) Test for Producao

This test controls the Type I experimentwise error rate, but it generally has a higher Type II error rate than REGWQ.

Alpha	0.05
Error Degrees of Freedom	4
Error Mean Square	6.297778
Critical Value of Studentized Range	5.04024
Minimum Significant Difference	7.3027

Means with the same letter are not significantly different.				
Tukey Grouping		Mean	N	Tratamento
	A	41.200	3	B
	A			
B	A	36.867	3	A
B				
B		32.000	3	C

The ANOVA Procedure

Tukey's Studentized Range (HSD) Test for Valor

This test controls the Type I experimentwise error rate, but it generally has a higher Type II error rate than REGWQ.

Alpha	0.05
Error Degrees of Freedom	4
Error Mean Square	37.66667
Critical Value of Studentized Range	5.04024
Minimum Significant Difference	17.859


Means with the same letter are not significantly different.				
Tukey Grouping		Mean	N	Tratamento
	A	135.667	3	B
	A			
B	A	125.000	3	A
B				

Means with the same letter are not significantly different.				
Tukey Grouping		Mean	N	Tratamento
B		111.333	3	C

Capítulo 11 : PROC CHART / PROC GCHART

Os procedimentos CHART e GCHART diferem principalmente na qualidade dos gráficos produzidos. O primeiro é mais antigo e emite gráficos de baixa resolução que não são mais utilizados, mantido somente para compatibilizar programas antigos com o sistema atual. Estes procedimentos não estão disponíveis no SAS University Edition.

O PROC GCHART produz gráficos de barra horizontal e vertical, blocos, bolos e estrela, podendo dispor estatísticas associadas aos seus valores. As variáveis delineadas podem ser numéricas ou de caractere.

É possível editar os gráficos com o editor de gráficos do próprio SAS para, por exemplo, traduzir as palavras internas em inglês, para o português. Depois de rodar o programa, acesse os resultados e clique sobre o gráfico a editar; depois acesse o editor de gráficos através do menu [Tools|Graphics Editor]. Para editar textos, clique no cursor de seleção  do editor e depois sobre o texto a alterar. Com a caixa de texto selecionada, clique sobre ela com o botão direito e depois, na janelinha flutuante que é ativada, clique em [View|Object Properties|More-], então é só alterar o texto e fechar as janelas clicando [End], [End] e salvando o gráfico.

Sintaxe:

```
PROC CHART <OPÇÃO(ÕES)> BLOCK VARIÁVEL(EIS) < / OPÇÃO(ÕES)>;  
BY<VARIÁVEL(EIS)>;  
HBAR VARIÁVEL(EIS) < / OPTION(S)>;  
PIE VARIÁVEL(EIS) < / OPTION(S)>;  
STAR VARIÁVEL(EIS) < / OPTION(S)>;  
VBAR VARIÁVEL(EIS) < / OPTION(S)>;
```

Opções do procedimento:

DATA=ArquivoDeDados – Identifica o arquivo de dados a ser usado com o procedimento.

FORMCHAR <(posição(ões))>='CaracterDeFormatação(ões)' – Define o caractere a ser utilizado na construção do gráfico.

Posição(ões) – identifica a posição de um ou mais caracteres na corrente de caracteres de formatação SAS. Um espaço ou uma vírgula separa as posições. O procedimento pode usar até 6 dos 20 caracteres de formatação disponíveis no SAS.

Caracteres de formatação – São utilizados na ordem em que são grafados na corrente de caracteres. Por exemplo, a opção a seguir assinala o asterisco como segundo caractere de formatação e o sinal de cancela como o sétimo e não altera os caracteres remanescentes: `formchar(2,7)='*#'`

As demais opções são descritas no manual “Introduction to SAS/Graph Software – Chapter 13: The GCHART Procedure”.

Exemplo 9.1 – Exemplo de gráficos de blocos, de barras verticais e horizontais

```

* EXEMPLO 9.1 - GRÁFICOS DE BLOCOS E DE BARRAS          1
  HORIZONTAIS;                                          2
DATA TOTAIS;                                           3
LENGTH ESPECIE $ 11 SITIO $ 5;                         4
INPUT ESPECIE SITIO PARCELA MORTAS;                   5
DATALINES;                                             6
P.ELLIOTTII BOM 1 15                                   7
P.TAEDA BOM 2 12                                       8
P.CARIBEA BOM 3 15                                     9
P.ELLIOTTII MEDIO 4 15                                10
P.TAEDA MEDIO 5 14                                    11
P.CARIBEA MEDIO 6 16                                  12
P.ELLIOTTII RUIM 7 17                                 13
P.TAEDA RUIM 8 30                                     14
P.CARIBEA RUIM 9 47                                   15
;                                                       16
PROC GCHART DATA=TOTAIS;                               17
TITLE 'TOTAL DE ARVORES MORTAS por sitio';             18
  FOOTNOTE J=R 'GRAFICO 1';                             19
  FORMAT MORTAS 8.;                                     20
  BLOCK SITIO / SUMVAR=MORTAS;                          21
PROC GCHART DATA=TOTAIS;                               22
TITLE 'Média DE ARVORES MORTAS por especie';          23
  FOOTNOTE J=R 'GRAFICO 2';                             24
  FORMAT MORTAS 8.;                                     25
  VBAR ESPECIE / SUMVAR=MORTAS TYPE=MEAN;              26
PROC GCHART DATA=TOTAIS;                               27
TITLE 'frequencia DE ARVORES MORTAS por especie em cada  28
  sitio';                                              29
  FOOTNOTE J=R 'GRAFICO 3';                             30
  FORMAT MORTAS 8.;                                     31
  HBAR ESPECIE / FREQ=MORTAS GROUP=SITIO;              32
RUN;                                                    33
QUIT;

```

Neste exemplo são produzidos três gráficos com os dados de mortalidade de árvores em nove parcelas, sendo uma parcela de cada uma das três espécies plantadas em três diferentes sítios.

O primeiro gráfico, de blocos, representa o total de árvores mortas em cada sítio.

O segundo gráfico, de barras verticais, representa a média por espécie. Na linha 25 é informada a variável a ser utilizada através da instrução SUMVAR= e o tipo de estatística é informado através da opção TYPE=MEAN; quando esta opção não é usada o gráfico é feito com os totais, como no gráfico anterior.

O último gráfico, de barras horizontais, representa a frequência de árvores mortas por espécie em cada sítio e é apresentada a percentagem do total de árvores mortas em cada classe; a classificação das espécies por sítio é realizada através da opção GROUP=SITIO (Linha 30).

São usadas notas de rodapé para identificar os gráficos através da declaração FOOTNOTE J=R 'GRAFICO n' ; em que J= é utilizado para indicar o alinhamento da nota em relação à página, neste caso R de *Right* (à direita).

Como resultado do processamento do programa, no exemplo 8.1, são emitidos os gráficos a seguir (os gráficos foram editados):

TOTAL DE ARVORES MORTAS POR SITIO
 GRAFICO DE BLOCOS DAS SOMAS

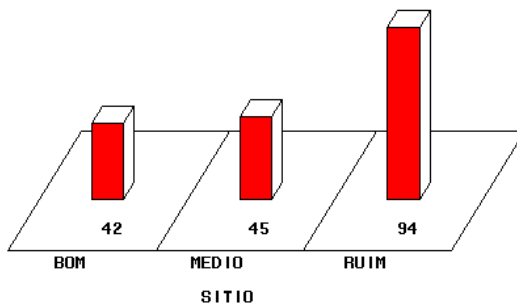


GRAFICO 1

MEDIA DE ARVORES MORTAS POR ESPECIE

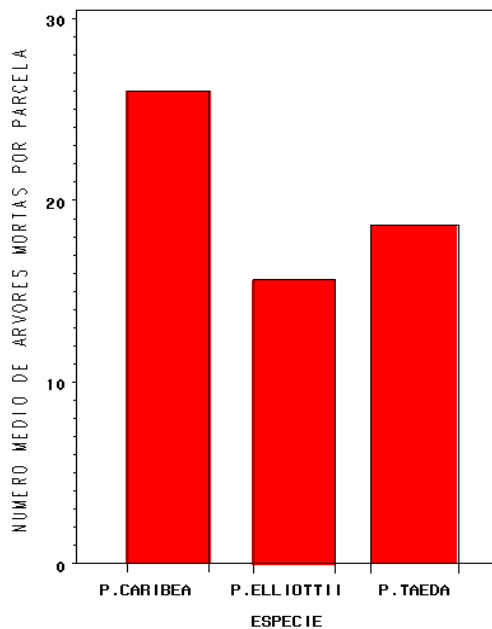
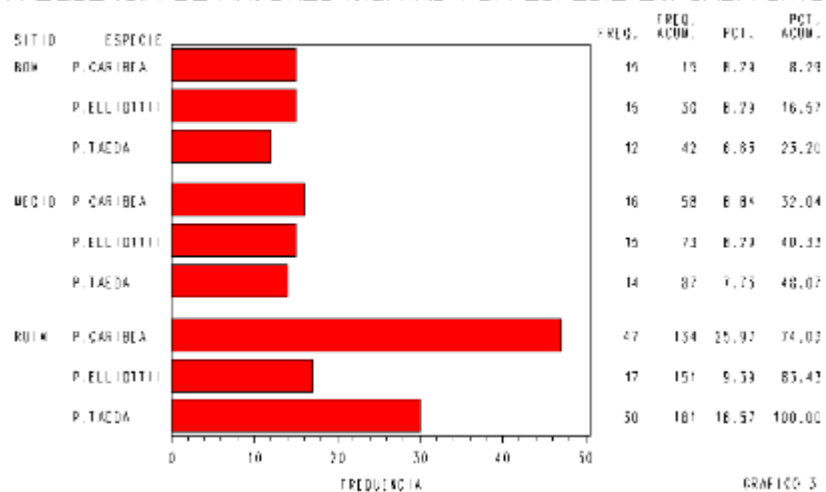


GRAFICO 2

FREQUENCIA DE ARVORES MORTAS POR ESPECIE EM CADA SITIO



Capítulo 12 : PROC CORR

O procedimento PROC CORR analisa variáveis aleatórias, computando coeficientes de correlação de Pearson, três medições de associação não-paramétrica e probabilidades associadas com estas estatísticas e calcula estatísticas descritivas.

Os cálculos de correlação incluem: produto-momento de Pearson e correlação ponderada (weighted) de produto-momento, correlação de ordem de amplitude de Spearman, tau-b de Kendall, medida de dependência D de Hoeffding, correlação parcial de Pearson, de Spearman, e de Kendall; coeficiente alfa de Cronbach para dependência.

12.1 SINTAXE:

```
PROC CORR <OPÇÕES>;
BY <DESCENDING> VARIÁVEL-1
  <...<DESCENDING> VARIÁVEL-N> <NOTSORTED>;
FREQ FREQUENCY-VARIABLE;
PARTIAL VARIÁVEIS;
VAR VARIÁVEIS;
WEIGHT VARIÁVEL;
WITH VARIÁVEIS;
```

12.2 OPÇÕES DO PROCEDIMENTO:

ALPHA – Calcula e imprime o coeficiente alfa de Cronbach.

BEST=n – Imprime n coeficientes de correlação para cada variável.

COV – Calcula e imprime as covariâncias.

CSSCP – Imprime as somas de quadrados corrigidos e os produtos cruzados.

DATA=ArquivoDeDados – Especifica o arquivo de dados para processamento.

EXCLNPWGT – Exclui observações com pesos nulos e negativos da análise.

HOEFFDING – Calcula e imprime a estatística D de Hoeffding.

KENDALL – Calcula e imprime os coeficientes tau-b de Kendall com base no número de pares de observações concordantes e discordantes. Não funciona com a instrução PARTIAL.

NOCORR – Suspende o cálculo dos coeficientes de correlação de Pearson.

NOMISS – Exclui as observações com valores perdidos.

NOPRINT – Suspende a impressão de relatório do procedimento.

NOPROB – Suspende a impressão de probabilidades associadas a cada coeficiente de correlação.

NOSIMPLE – Suspende a impressão de estatísticas descritivas simples para cada variável.

OUTH=ArquivoDeSaída – Cria um arquivo de saída contendo a estatística D de Hoeffding.

OUTK=ArquivoDeSaída – Cria um arquivo de saída contendo as estatísticas para a correlação de Kendall.

OUTP=ArquivoDeSaída – Cria um arquivo de saída contendo as estatísticas da correlação de Pearson, médias, desvios-padrão e número de observações. O valor da variável `_TYPE_` é CORR. Se a opção ALPHA for usada, o arquivo conterá 6 observações com os coeficientes alpha de Cronbach.

OUTS=ArquivoDeSaída – Cria um arquivo de saída contendo as estatísticas da correlação de Spearman.

PEARSON – Calcula e imprime as correlações de produto-momento de Pearson product-moment quando é utilizada a opção HOEFFDING, KENDALL, ou SPEARMAN. Se a opção for omitida, o procedimento calcula automaticamente os coeficientes de correlação de Pearson.

RANK – Imprime os coeficientes de correlação para cada variável na ordem do mais alto para o mais baixo. O procedimento imprime a estatística D se for utilizada a opção HOEFFDING.

SINGULAR=p – Especifica o critério para determinação de singularidade quando é utilizada a instrução PARCIAL. Uma variável é considerada singular se a diagonal principal após a decomposição de Cholesky tiver valor menor que p vezes a original soma de quadrados não parcializada da variável. O padrão é 1E-8 e a amplitude de 0 até 1.

SPEARMAN – Calcula e imprime os coeficientes de correlação de Spearman com base nos graus das variáveis. Não é válido com a instrução WEIGHT.

SSCP – Imprime as somas de quadrados dos produtos cruzados. Invoca a correlação de PEARSON. Usada com a instrução PARTIAL, é impressa a matriz não parcial com a opção SSCP.

VARDEF=divisor – Especifica o divisor a usar no cálculo de variâncias, desvios-padrão e covariâncias. O padrão é DF.

12.3 INSTRUÇÕES DO PROCEDIMENTO:

12.3.1 BY (INSTRUÇÃO)

Produz análises de correlação separadamente para cada grupo de variável especificada.

Sintaxe:

```
■ BY <DESCENDING> VARIÁVEL-1 <...<DESCENDING>  
  VARIÁVEL-N> <NOTSORTED>;
```

Variável(1 a n) - São as variáveis pelas quais devem ser separados os grupos; devem ser escritas em ordem de hierarquia.

DESCENDING – Indica que a ordem das observações da variável especificada é descendente.

NOTSORTED – Indica que as observações não estão em ordem alfabética, mas em uma ordem determinada pelo usuário.

12.3.2 FREQ (INSTRUÇÃO)

Identifica a variável que representa a frequência para cada observação.

Sintaxe:

```
■ FREQ VARIÁVEL;
```

Variável – É o nome da variável que possui a frequência de cada observação do arquivo de dados.

12.3.3 PARTIAL (INSTRUÇÃO)

Identifica as variáveis de controle para computar os coeficientes de correlação parcial de Pearson, Spearman, ou Kendall.

Sintaxe:

■ **PARTIAL VARIÁVEIS;**

Variáveis – Identifica uma ou mais variáveis para computar correlações parciais.

12.3.4 VAR (INSTRUÇÃO)

Identifica as variáveis a considerar na análise de correlação e a sua ordem.

Sintaxe:

■ **VAR VARIÁVEIS;**

Variáveis – Identifica uma ou mais variáveis para usar no cálculo dos coeficientes de correlação.

12.3.5 WEIGHT (INSTRUÇÃO)

Identifica a variável cujos valores representam o peso de cada observação para calcular o coeficiente de correlação ponderada de produto-momento de Pearson.

Sintaxe:

■ **WEIGHT VARIÁVEL;**

Variável – Especifica uma variável numérica que contém o peso para cálculo dos coeficientes de correlação ponderados de produto-momento. Não é necessário que os valores sejam inteiros. Se o valor é 0 o número de observações é contado no número total de observações; menor que 0, converte o valor para zero e conta as observações no número total de observações; valor perdido exclui a observação.

12.3.6 WITH (INSTRUÇÃO)

Calcula correlações para as combinações especificadas de variáveis. Deve ser usada em conjunto com a instrução VAR.

Sintaxe:

■ **WITH VARIÁVEIS;**

Variáveis – Lista de uma ou mais variáveis para obter coeficientes de correlação de combinações específicas. Exemplo: * se forem declaradas as instruções; VAR A B; WITH X Y Z; * serão computadas as combinações X e A, X e B, Y e A, Y e B, Z e A, Z e B;

12.3.7 VALORES PERDIDOS (MISSING VALUES)

O padrão do sistema usa todos os pares de valores não perdidos das variáveis listadas nas instruções VAR e WITH; entretanto, valores perdidos podem causar problemas de cálculos e criar tendenciosidades, para evitar isso deve ser utilizada a opção NOMISS sempre que se souber da existência de valores perdidos entre os dados das variáveis, ou se deve transformar os valores perdidos em zeros antes do procedimento ser invocado.

12.3.8 ARQUIVOS DE SAÍDA TYPE=CORR

Arquivos de saída são requisitados pelas opções OUTP=, OUTS=, OUTK= e OUTH= entre as opções do procedimento. Podem ser produzidos arquivos SAS com as estatísticas de Pearson, Spearman, Kendall e Hoeffding. Estes arquivos de saída têm uma especificação própria denominada arquivos TYPE=CORR. Este tipo de arquivo é reconhecido por muitos outros procedimentos estatísticos do SAS, incluindo PROC REG e PROC FACTOR. Um arquivo TYPE=CORR contém uma matriz de correlação, as médias das variáveis, desvios-padrão, n número de observações no arquivo original SAS de onde a matriz foi computada e outras estatísticas.

No “Exemplo – Correlação entre variáveis dendrométricas” são correlacionadas as variáveis H, HC, DCC e CASCA.

```

* EXEMPLO - Correlacao entre variaveis dendrometricas; Title;
DATA arvores;
INPUT Arvore h hc dcc casca @@;
DATALINES;
1 21.05 16 26.1 2.9 13 16.8 12 22 3.5 25 19.3 13 16.6 2.3 37 20.1 14
  20.4 1.6
2 18.8 12 16.6 1.6 14 20.5 15 23.6 3.2 26 21 14 22.6 3.2 38 21.5 15 25.1
  3.1
3 21 15 23.6 2.9 15 16.7 11 16.9 2.9 27 20.9 15 23.2 3.5 39 17.4 12 17.2
  2.2
4 19.8 13 17.8 2.2 16 21.5 15 22.6 3.2 28 20.7 15 22.9 2.8 40 18 12 18.1
  2.5
5 19.3 13 21 1.9 17 20.3 15 24.8 3.5 29 17.2 11 15.6 1.9 41 21 15 23.6
  3.9
6 20.8 15 25.8 3.2 18 19.3 13 17.8 2.5 30 18.1 11 13.1 1.3 42 20.7 14 21
  2.2
7 17 11 13.6 0.4 19 19.6 13 15.9 2.2 31 20.5 14 21.3 3.2 43 15.6 10 15.6
  3.8
8 21.8 15 23.6 3.2 20 19.7 13 21 3.5 32 20.1 14 25.1 4.1 44 20.3 13 17.5
  1.6
9 22.1 15 20.4 3.2 21 19.5 14 23.9 3.5 33 18 12 16.2 2.2 45 21 14 21.3
  2.5
10 21.6 15 23.2 3.5 22 21.5 14 21.3 2.8 34 17.3 11 15.3 2.6 46 17.7 11
  15 1.9
11 17.3 11 14.3 1.9 23 19.9 14 22.9 2.5 35 18.6 13 18.8 2.6 47 19.1 13
  18.5 2.6

```

```

12 17.2 12 18.5 1.9 24 17.7 11 14.6 1.2 36 21.1 15 23.2 2.5 48 20 15
24.8 2.5
;
PROC CORR data=arvores;
VAR h hc dcc casca;
RUN;
QUIT;
    
```

Neste exemplo são usados os dados de altura (h), altura comercial (hc), diâmetro com casca (dcc) e dupla espessura de casca (casca) de 48 árvores cubadas, em análise de correlação simples. Os resultados são apresentados no relatório a seguir:

The CORR Procedure

4 Variables:	H HC DCC CASCA
--------------	-------------------

Simple Statistics						
Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
H	48	19.49896	1.65852	935.95000	15.60000	22.10000
HC	48	13.31250	1.57313	639.00000	10.00000	16.00000
DCC	48	20.07917	3.69727	963.80000	13.10000	26.10000
CASCA	48	2.62292	0.77494	125.90000	0.40000	4.10000

Pearson Correlation Coefficients, N = 48 Prob > r under H0: Rho=0				
	H	HC	DCC	CASCA
H	1.00000	0.92611 <.0001	0.74732 <.0001	0.40486 0.0043
HC	0.92611 <.0001	1.00000	0.89811 <.0001	0.52108 0.0001
DCC	0.74732 <.0001	0.89811 <.0001	1.00000	0.68737 <.0001
CASCA	0.40486 0.0043	0.52108 0.0001	0.68737 <.0001	1.00000

O primeiro quadro apresenta, para cada variável analisada, as estatísticas: frequência, média, desvio padrão, soma, mínimo e máximo valor observado.

No segundo quadro é apresentado o índice de correlação de Pearson (R), que são impressos na parte superior de cada célula, observa-se que há correlação altamente significativa entre todas as variáveis, pois todos os valores de probabilidade de

significância, que são os valores que aparecem na parte de baixo de cada célula, são inferiores a 1%, ou 0,01 em termos absolutos, como é impresso pelo SAS.

Capítulo 13 :

PROC FREQ

Este procedimento analisa frequências e proporções com a possibilidade de gerar tabelas de frequência e arquivos de saída com os resultados do processamento.

13.1 SINTAXE

```
PROC FREQ <OPÇÕESDOPROCEDIMENTO> ;
BY VARIÁVEIS ;
EXACT OPÇÕESESTATÍSTICAS </ OPÇÕESDEPROCESSAMENTO> ;
OUTPUT <OUT=ARQUIVODESAÍDA> OPÇÕESDESAÍDA;
TABLES REQUESTS </ OPÇÕESDETABELAS > ;
TEST OPÇÕESDETESTE ;
WEIGHT VARIÁVEL ;
```

Opções do procedimento:

COMPRESS - A opção COMPRESS começa a exibição da próxima tabela de frequência de um sentido na mesma página que a tabela de um sentido precedente, se houver bastante espaço para iniciá-la. Como padrão, a próxima tabela de um sentido começa só na página atual se a tabela inteira couber nela. A opção COMPRESS não é válida com a opção PAGE.

DATA=ArquivoDeEntrada - ArquivoDeEntrada é o nome do arquivo que contém as informações a serem analisadas pelo procedimento.

FORMCHAR(1,2,7) ='Caracterese separador' - A opção FORMCHAR permite definir o caractere a ser utilizado para construir as linhas e divisores das células das tabelas de contingências. Há 3 posições diferentes de separação que podem ser definidos usando 20 tipos de caracteres. É necessário definir todas as três posições: (1) separador vertical, (2) separador horizontal e (7) intersecção vertical-horizontal. Exemplos: FORMCHAR(1,2,7)=' ' define todos os separadores como invisíveis (brancos); FORMCHAR(1,2,7)='|'+', desenha o caractere | como separador vertical, - como separador horizontal e + nas intersecções entre linhas verticais e horizontais.

NOPRINT – Suspende a impressão de relatório pelo procedimento. Desabilita também o sistema de saída do procedimento (Output Delivery System-ODS).

ORDER= DATA | FORMATTED | FREQ | INTERNAL - A opção ORDER= especifica a ordem os valores de frequência e das variáveis de tabulação cruzada devem ser apresentadas. As sub-opções da opção ORDER são: DATA - ordena os valores de acordo com o arquivo de entrada; FORMATTED – ordena os valores conforme sua formatação - é dependente do sistema operacional; FREQ – ordena os valores por ordem descendente de frequência; INTERNAL – ordena os valores pela ordem de seus valores não formatados de forma semelhante à PROC SORT - é o padrão e dependente do sistema operacional.

PAGE – Esta opção obriga à apresentação de só uma tabela por página. Não é válida com a opção COMPRESS.

13.1.1 BY (INSTRUÇÃO)

Separa a análise por grupos de observações. É necessário ordenar o arquivo antes.

Sintaxe:

■ **BY VARIÁVEIS;**

Variáveis – São as variáveis pelas quais deve ser separada a análise formando grupos.

13.1.2 EXACT (INSTRUÇÃO)

Requisita testes exatos ou dentro de intervalos de confiança. Às vezes necessita muita memória e trava o computador.

Sintaxe:

■ **EXACT OPÇÕES ESTATÍSTICAS </ OPÇÕES COMPUTACIONAIS>;**

Opções Estatísticas – Especifica as estatísticas para realizar os testes.

Opções Computacionais – Especifica as opções para computar as estatísticas exatas.

13.1.3 OUTPUT (INSTRUÇÃO)

Define um arquivo de saída e as estatísticas que serão gravadas nele. É obrigatório especificar uma só instrução TABLES para gerar o arquivo.

Sintaxe:

■ **OUTPUT < OUT=ARQUIVODESAÍDA> OPÇÕES;**

Opções – São as estatísticas desejadas no arquivo de saída, sempre referentes ao especificado na instrução TABLES como: AGREE, ALL, CHISQ, CMH e MEASURES.

13.1.4 TABLES (INSTRUÇÃO)

Solicita frequências de único sentido, de n sentidos, tabelas de tabulação cruzada e computa as estatísticas para as mesmas.

■ **TABLES REQUISIÇÕES DE TABELAS / <OUT=ARQUIVODESAÍDA> OPÇÕES;**

Requisições De Tabelas – Especifica as tabelas de frequência e de tabulação cruzada a produzir. As tabelas podem ser dos tipos a seguir:

- TABLES A*(B C); = TABLES A*B A*C;
- TABLES (A B)*(C D); = TABLES A*C B*C A*D B*D;
- TABLES (A B C)*D; = TABLES A*D B*D C*D;
- TABLES A -C; = TABLES A B C;
- TABLES (A -C)*D; = TABLES A*D B*D C*D.

Sem opções – Tabelas de frequência de um sentido para uma variável sem opções especificadas produz frequências simples e acumuladas, percentagens das frequências simples e acumuladas para cada valor da variável em análise. Se for solicitada uma tabela de duplo sentido ou uma tabela de tabulação cruzada de n sentidos sem especificar opções, o PROC FREQ produz tabelas de tabulação cruzada que incluem frequências de célula, percentagens de célula da frequência total, percentagens de célula de frequências

de linha e percentagens de célula de frequências de coluna. O procedimento exclui observações com valores perdidos da tabela mas exhibe a frequência total de observações perdidas sob cada tabela.

Opções – São relacionadas a seguir: AGREE < (WT=FC) >; ALL; ALPHA= ; BINOMIAL < (P= valor) | (LEVEL= NúmeroDeNíveis | ValorDoNível) >; BINOMIALC < (P= valor) | (LEVEL= NúmeroDeNíveis | ValorDoNível) >; CELLCHI2; CHISQ; CL; CMH; CMH1; CMH2; CONTENTS=TextoDeLigação; CONVERGE=Valor; CUMCOL; DEVIATION; FISHER | EXACT; EXPECTED; FORMATO=Formato-
nome; JT; LIST; MAXITER=Número; MEASURES; MISSING; MISSPRINT; NOCOL; NOCUM; NOFREQ; NOPERCENT; NOPRINT; NOROW; OUT=ArquivoSAS; OUTCUM; OUTEXPECT; OUTPCT; PLCORR; PRINTKWT; RELRISK; RISKDIFF; RISKDIFFC; SCORES=Tipo; SCOROUT; SPARSE; TESTF=(Valores); TESTP=(Valores); TOTPCT; TREND; OUT=ArquivoSAS – Esta opção quando usada com a instrução permite gerar um arquivo de saída com os valores e frequências das variáveis do arquivo de entrada em que a variável COUNT contém as frequências e a variável PERCENT contém as percentagens. Podem ser especificadas também as opções OUTCUM (frequências acumuladas), OUTEXPECT (frequências esperadas) e OUTPCT (frequência em percentagem). A opção SPARSE lista todas as possibilidades para tabelas de n sentidos quando n>1.

13.1.5 TEST (INSTRUÇÃO)

A instrução TEST requisita testes assintóticos para associações de medições especificadas e medições de conformidade. É necessário utilizar a instrução TABLES com a instrução TEST.

Sintaxe:

■ TEST ESTATÍSTICA(S);

Estatística(s) – Especifica as estatísticas que devem ser fornecidas com os testes assintóticos.

13.1.6 WEIGHT (INSTRUÇÃO)

Especificação de uma variável numérica que possui os valores para usar como frequência das observações do arquivo de dados de entrada.

Sintaxe:

■ WEIGHT VARIÁVEL;

Variável – É o nome da variável que possui os valores para usar como frequência das observações do arquivo de dados de entrada.

O “EXEMPLO - FREQUENCIA DE ARVORES POR SITUACAO” a seguir demonstra o uso do procedimento e da elaboração de tabelas de frequência no SAS.

```
* EXEMPLO - FREQUENCIA DE ARVORES POR SITUACAO; 1
DATA ARVORES; 2
INPUT PARCELA ARVORE DAP SITUACAO SITIO @@; 3
CLASSE=INT(DAP/5); 4
***** 5
SITUACOES DAS ARVORES: 6
0=NORMAL 7
1=MORTA 8
```

2=BIFURDADA	9
3=QUEBRADA	10
4=DOENTE	11
5=ATACADA POR PRAGA	12
*****	13
;	14
DATALINES;	15
1 1 19.5 0 1 3 1 12.8 0 1 5 1 14.5 0 2	16
1 2 17.3 0 1 3 2 17.2 2 1 5 2 20.8 0 2	17
1 3 19.9 0 1 3 3 18.6 0 1 5 3 11.1 0 2	18
1 4 27.9 0 1 3 4 19.3 0 1 5 4 13.0 2 2	19
1 5 30.9 0 1 3 5 25.5 0 1 5 5 14.9 0 2	20
1 6 14.4 0 1 3 6 0.0 1 1 5 6 15.4 0 2	21
1 7 0.0 1 1 3 7 20.8 0 1 5 7 14.1 0 2	22
1 8 12.6 0 1 3 8 16.8 0 1 5 8 19.7 0 2	23
1 9 22.0 2 1 3 9 0.0 1 1 5 9 19.7 0 2	24
1 10 20.3 0 1 3 10 15.6 0 1 5 10 15.0 0 2	25
1 11 19.6 0 1 3 11 9.6 0 1 5 11 17.5 0 2	26
1 12 19.8 0 1 3 12 14.0 0 1 5 12 23.3 0 2	27
1 13 26.1 0 1 3 13 22.5 0 1 5 13 8.3 0 2	28
1 14 17.0 0 1 3 14 0.0 1 1 5 14 15.3 4 2	29
1 15 19.5 0 1 3 15 13.8 0 1 5 15 15.1 0 2	30
1 16 14.1 0 1 3 16 19.5 0 1 5 16 17.0 0 2	31
1 17 22.3 0 1 3 17 18.5 0 1 5 17 17.4 0 2	32
1 18 13.5 0 1 3 18 20.6 0 1 5 18 13.9 0 2	33
1 19 20.5 0 1 3 19 19.5 0 1 5 19 0.0 1 2	34
1 20 20.5 0 1 3 20 19.5 3 1 5 20 15.7 0 2	35
1 21 16.5 0 1 3 21 15.8 0 1 5 21 16.1 0 2	36
1 22 23.0 0 1 3 22 21.1 0 1 5 22 16.7 0 2	37
1 23 14.7 0 1 3 23 12.3 0 1 5 23 18.6 0 2	38
1 24 19.9 0 1 3 24 30.8 0 1 5 24 13.2 2 2	39
1 25 14.9 0 1 3 25 20.3 0 1 5 25 21.9 0 2	40
1 26 17.1 0 1 3 26 22.2 2 1 5 26 19.7 0 2	41
1 27 22.5 2 1 3 27 11.1 0 1 5 27 11.1 0 2	42
1 28 19.9 0 1 3 28 19.3 0 1 5 28 23.3 0 2	43
1 29 13.5 0 1 3 29 21.1 0 1 5 29 22.7 0 2	44
1 30 13.1 0 1 3 30 9.7 0 1 5 30 21.8 0 2	45
1 31 19.4 0 1 3 31 15.2 0 1 5 31 10.8 0 2	46
1 32 15.0 0 1 3 32 26.5 0 1 5 32 11.3 0 2	47
1 33 17.0 0 1 3 33 20.6 0 1 5 33 19.9 0 2	48
1 34 20.0 0 1 3 34 12.3 0 1 5 34 9.0 0 2	49
1 35 11.8 0 1 3 35 9.9 0 1 5 35 17.0 2 2	50
1 36 18.1 0 1 3 36 15.0 0 1 5 36 19.7 0 2	51
1 37 20.0 0 1 3 37 15.6 0 1 5 37 22.0 0 2	52
1 38 20.3 0 1 3 38 17.5 0 1 5 38 17.9 0 2	53
1 39 15.9 0 1 3 39 22.1 0 1 5 39 14.6 0 2	54
1 40 16.2 0 1 3 40 24.1 0 1 5 40 17.5 0 2	55
1 41 16.4 0 1 3 41 20.8 0 1 5 41 12.6 0 2	56
1 42 14.9 0 1 3 42 19.0 0 1 5 42 21.1 0 2	57
1 43 11.4 0 1 3 43 14.4 0 1 5 43 26.4 0 2	58
1 44 17.6 0 1 3 44 12.8 0 1 5 44 18.2 0 2	59
1 45 12.7 0 1 3 45 13.5 0 1 5 45 16.6 2 2	60
1 46 19.5 0 1 3 46 15.1 0 1 5 46 20.0 0 2	61
1 47 11.8 0 1 3 47 19.8 0 1 5 47 14.6 0 2	62
1 48 26.3 0 1 3 48 25.5 0 1 5 48 19.7 0 2	63
1 49 17.5 0 1 3 49 0.0 1 1 5 49 19.9 0 2	64
1 50 16.7 0 1 3 50 19.3 0 1 5 50 9.0 0 2	65
2 1 13.7 0 1 4 1 28.4 2 2 6 1 18.9 0 2	66
2 2 16.2 0 1 4 2 23.1 0 2 6 2 19.5 2 2	67
2 3 15.2 0 1 4 3 30.1 0 2 6 3 18.8 0 2	68
2 4 25.9 0 1 4 4 14.1 0 2 6 4 12.0 0 2	69
2 5 17.6 0 1 4 5 21.9 0 2 6 5 11.7 0 2	70
2 6 19.8 0 1 4 6 21.3 0 2 6 6 0.0 1 2	71

```

2 7 16.0 2 1 4 7 10.8 0 2 6 7 22.8 0 2 72
2 8 24.9 0 1 4 8 16.1 0 2 6 8 21.5 0 2 73
2 9 10.9 0 1 4 9 17.3 0 2 6 9 10.2 0 2 74
2 10 17.7 0 1 4 10 14.2 0 2 6 10 15.3 0 2 75
2 11 18.8 0 1 4 11 14.1 0 2 6 11 21.9 0 2 76
2 12 21.1 0 1 4 12 24.2 0 2 6 12 11.7 0 2 77
2 13 23.0 0 1 4 13 17.8 0 2 6 13 14.5 0 2 78
2 14 18.5 0 1 4 14 11.9 0 2 6 14 20.8 0 2 79
2 15 14.0 0 1 4 15 16.9 0 2 6 15 11.1 2 2 80
2 16 16.7 0 1 4 16 15.1 0 2 6 16 19.5 0 2 81
2 17 10.5 0 1 4 17 22.0 0 2 6 17 19.5 0 2 82
2 18 11.5 0 1 4 18 0.0 1 2 6 18 15.8 0 2 83
2 19 11.4 0 1 4 19 18.5 2 2 6 19 21.1 0 2 84
2 20 16.8 0 1 4 20 12.1 0 2 6 20 12.3 0 2 85
2 21 19.2 0 1 4 21 16.0 0 2 6 21 30.8 0 2 86
2 22 21.3 0 1 4 22 13.1 0 2 6 22 20.3 0 2 87
2 23 17.2 0 1 4 23 20.1 0 2 6 23 22.2 0 2 88
2 24 18.1 0 1 4 24 8.6 0 2 6 24 11.1 0 2 89
2 25 0.0 1 1 4 25 19.3 0 2 6 25 19.3 2 2 100
2 26 18.4 0 1 4 26 11.5 0 2 6 26 21.8 4 2 101
2 27 15.4 0 1 4 27 19.9 0 2 6 27 17.2 0 2 102
2 28 0.0 1 1 4 28 0.0 1 2 6 28 18.0 0 2 103
2 29 15.9 0 1 4 29 19.4 0 2 6 29 20.2 0 2 104
2 30 25.4 0 1 4 30 13.6 2 2 6 30 12.8 0 2 105
2 31 16.5 5 1 4 31 22.3 0 2 6 31 19.3 0 2 016
2 32 17.2 0 1 4 32 21.9 0 2 6 32 25.5 0 2 107
2 33 28.0 0 1 4 33 18.1 0 2 6 33 11.4 0 2 108
2 34 10.8 0 1 4 34 17.3 0 2 6 34 16.8 0 2 110
2 35 15.4 0 1 4 35 18.0 0 2 6 35 13.8 0 2 111
2 36 12.3 0 1 4 36 15.7 0 2 6 36 17.1 0 2 112
2 37 21.8 2 1 4 37 22.2 0 2 6 37 22.5 0 2 113
2 38 17.2 0 1 4 38 20.9 0 2 6 38 19.9 2 2 114
2 39 18.0 0 1 4 39 18.9 2 2 6 39 13.5 0 2 115
2 40 20.2 0 1 4 40 19.5 0 2 6 40 13.1 0 2 116
2 41 12.8 0 1 4 41 18.8 0 2 6 41 19.4 0 2 117
2 42 19.3 0 1 4 42 12.0 0 2 6 42 15.0 0 2 118
2 43 25.5 0 1 4 43 11.7 0 2 6 43 17.0 5 2 119
2 44 11.4 0 1 4 44 0.0 1 2 6 44 20.0 0 2 120
2 45 16.8 0 1 4 45 22.8 0 2 6 45 11.8 0 2 121
2 46 13.8 0 1 4 46 21.5 0 2 6 46 18.1 0 2 122
2 47 19.5 0 1 4 47 10.2 0 2 6 47 20.0 0 2 123
2 48 14.4 0 1 4 48 15.3 0 2 6 48 20.3 0 2 124
2 49 12.8 0 1 4 49 21.9 0 2 6 49 11.4 0 2 125
2 50 20.6 0 1 4 50 11.7 0 2 6 50 16.8 0 2 126
; 127
PROC SORT DATA=ARVORES; BY SITIO CLASSE 128
SITUACAO; 129
PROC FREQ DATA=ARVORES; 130
TITLE 'FREQUENCIA DE ARVORES EM CADA SITIO POR 131
SITUACAO'; 132
TABLES SITIO * SITUACAO; 133
TABLES CLASSE * SITUACAO; 134
TABLES SITIO * CLASSE * SITUACAO; 135
RUN;
QUIT;

```

São criadas classes de DAP com intervalo de 5 cm através da Linha 4 do programa.

São dados três comandos para elaboração de tabelas (TABLES), sendo que o último gera duas tabelas, uma para cada sítio (Linhas 131 a 133).

O resultado do processamento são as tabelas de referência cruzada com as situações das árvores localizadas nas colunas.

Interpretando os resultados da primeira tabela, observa-se que no sítio 1, 90% das árvores são normais, 4,67% são mortas, 4% são bifurcadas, 0,67% são quebradas, 0% são doentes e 0,67% são atacadas por pragas; no sítio 2, 86,67% são normais, 3,33% são mortas, 8% são bifurcadas, 0% são quebradas, 1,33% são doentes e 0,67% são atacadas por pragas.

A interpretação das demais tabelas segue a mesma lógica.

FREQUENCIA DE ARVORES EM CADA SITIO POR SITUACAO
The FREQ Procedure

Table of SITIO by SITUACAO							
SITIO	SITUACAO						Total
	0	1	2	3	4	5	
1	135	7	6	1	0	1	150
	45.00	2.33	2.00	0.33	0.00	0.33	50.00
	90.00	4.67	4.00	0.67	0.00	0.67	
	50.94	58.33	33.33	100.00	0.00	50.00	
2	130	5	12	0	2	1	150
	43.33	1.67	4.00	0.00	0.67	0.33	50.00
	86.67	3.33	8.00	0.00	1.33	0.67	
	49.06	41.67	66.67	0.00	100.00	50.00	
Total	265	12	18	1	2	2	300
	88.33	4.00	6.00	0.33	0.67	0.67	100.00

Table of CLASSE by SITUACAO							
CLASSE	SITUACAO						Total
	0	1	2	3	4	5	
0	0 0.00 0.00 0.00	12 4.00 100.00 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	12 4.00
1	7 2.33 100.00 2.64	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	7 2.33
2	72 24.00 94.74 27.17	0 0.00 0.00 0.00	4 1.33 5.26 22.22	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	76 25.33
3	112 37.33 89.60 42.26	0 0.00 0.00 0.00	9 3.00 7.20 50.00	1 0.33 0.80 100.00	1 0.33 0.80 50.00	2 0.67 1.60 100.00	125 41.67
4	58 19.33 92.06 21.89	0 0.00 0.00 0.00	4 1.33 6.35 22.22	0 0.00 0.00 0.00	1 0.33 1.59 50.00	0 0.00 0.00 0.00	63 21.00
5	12 4.00 92.31 4.53	0 0.00 0.00 0.00	1 0.33 7.69 5.56	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	13 4.33
6	4 1.33 100.00 1.51	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	4 1.33
Total	265 88.33	12 4.00	18 6.00	1 0.33	2 0.67	2 0.67	300 100.00

Frequency
Percent
Row Pct
Col Pct

Frequency
Percent
Row Pct
Col Pct

Table 1 of CLASSE by SITUACAO							
Controlling for SITIO=1							
CLASSE	SITUACAO						Total
	0	1	2	3	4	5	
0	0 0.00 0.00 0.00	7 4.67 100.00 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	7 4.67
1	3 2.00 100.00 2.22	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	3 2.00
2	35 23.33 100.00 25.93	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	35 23.33
3	61 40.67 93.85 45.19	0 0.00 0.00 0.00	2 1.33 3.08 33.33	1 0.67 1.54 100.00	0 0.00 0.00 0.00	1 0.67 1.54 100.00	65 43.33
4	24 16.00 85.71 17.78	0 0.00 0.00 0.00	4 2.67 14.29 66.67	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	28 18.67
5	10 6.67 100.00 7.41	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	10 6.67
6	2 1.33 100.00 1.48	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	2 1.33
Total	135 90.00	7 4.67	6 4.00	1 0.67	0 0.00	1 0.67	150 100.00

Frequency
Percent
Row Pct
Col Pct

Table 2 of CLASSE by SITUACAO							
Controlling for SITIO=2							
CLASSE	SITUACAO						Total
	0	1	2	3	4	5	
0	0 0.00 0.00 0.00	5 3.33 100.00 100.00	0 0.00 0.00 0.00	0 0.00 0.00 .	0 0.00 0.00 0.00	0 0.00 0.00 0.00	5 3.33
1	4 2.67 100.00 3.08	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 .	0 0.00 0.00 0.00	0 0.00 0.00 0.00	4 2.67
2	37 24.67 90.24 28.46	0 0.00 0.00 0.00	4 2.67 9.76 33.33	0 0.00 0.00 .	0 0.00 0.00 0.00	0 0.00 0.00 0.00	41 27.33
3	51 34.00 85.00 39.23	0 0.00 0.00 0.00	7 4.67 11.67 58.33	0 0.00 0.00 .	1 0.67 1.67 50.00	1 0.67 1.67 100.00	60 40.00
4	34 22.67 97.14 26.15	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 .	1 0.67 2.86 50.00	0 0.00 0.00 0.00	35 23.33
5	2 1.33 66.67 1.54	0 0.00 0.00 0.00	1 0.67 33.33 8.33	0 0.00 0.00 .	0 0.00 0.00 0.00	0 0.00 0.00 0.00	3 2.00
6	2 1.33 100.00 1.54	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 .	0 0.00 0.00 0.00	0 0.00 0.00 0.00	2 1.33
Total	130 86.67	5 3.33	12 8.00	0 0.00	2 1.33	1 0.67	150 100.00

Capítulo 14 : PROC MEANS / PROC SUMMARY

O procedimento MEANS calcula estatísticas simples para variáveis numéricas. O procedimento MEANS com a opção NOPRINT realiza o mesmo que o SUMMARY e não emite relatório. Os dois procedimentos são equivalentes, mas o PROC SUMMARY não emite relatório automático.

14.1 SINTAXE

```
PROC MEANS <OPÇÕES> <ESTATÍSTICAS>;  
VAR <VARIABLE LIST>;  
BY <VARIABLE LIST>;  
OUTPUT <OUT= > <OUTPUT STATISTICS LIST>
```

14.2 OPÇÕES DO PROCEDIMENTO

Algumas opções do procedimento são:

DATA= - Nome do arquivo com dados a serem analisados.

NOPRINT- suprime a impressão do PROC MEANS. Usa-se quando um arquivo de saída é criado PROC MEANS.

14.3 ALGUMAS ESTATÍSTICAS

Algumas estatísticas providas pelo procedimento são:

CSS – Soma de quadrados corrigidos.

CV – Coeficiente de variação.

MEAN – Média.

N – Número de observações.

NMISS – Número de observações perdidas.

STD – Desvio padrão.

SUM – Soma.

USS – Soma de quadrados não-corrigidos.

VAR - Variância (Não confundir com a declaração VAR).

A seguir, demonstra-se o cálculo de médias com o “Exemplo – Médias por parcela, por sítio”.

Neste exemplo foram utilizados os dados do “EXEMPLO - FREQUENCIA DE ARVORES POR SITUACAO” do capítulo anterior.

As alturas foram calculadas pela equação $H=23,5-1047,3/DAP^2$ (Linha 6).

As árvores mortas são deletadas antes do cálculo das alturas (Linha 5), para evitar erro de divisão por zero, pois as árvores mortas têm $DAP=0$, o que iria interromper o processamento.

```

* EXEMPLO - Médias por parcela, por sitio;
DATA ARVORES;
INPUT PARCELA ARVORE DAP SITUACAO SITIO @@;
CLASSE=INT(DAP/5);
if situacao=1 then delete;
H=23.5-1047.3/DAP**2;
*****
SITUACOES DAS ARVORES:
0=NORMAL
1=MORTA
2=BIFURDADA
3=QUEBRADA
4=DOENTE
5=ATACADA POR PRAGA
*****
;
DATALINES;
1 1 19.5 0 1 3 1 12.8 0 1 5 1 14.5 0 2
1 2 17.3 0 1 3 2 17.2 2 1 5 2 20.8 0 2
1 3 19.9 0 1 3 3 18.6 0 1 5 3 11.1 0 2
...
...
...
;
PROC MEANS DATA=ARVORES MEAN STD;
TITLE1 'MEDIA E DESVIO PADRAO DO DIAMETRO E
        ALTURA';
TITLE2 'POR CLASSE DE DAP EM CADA SITIO';
CLASS CLASSE;
VAR DAP H;
BY SITIO;
RUN;
QUIT;

```

Como resultado do processamento é emitido o seguinte relatório:

MEDIA E DESVIO PADRAO DO DIAMETRO E ALTURA
POR CLASSE DE DAP EM CADA SITIO

The MEANS Procedure

SITIO=1

CLASSE	N Obs	Variable	Mean	Std Dev
1	3	DAP	9.7333333	0.1527525
		H	12.4398637	0.3446260
2	35	DAP	12.8771429	1.2718874
		H	16.9952459	1.3369736
3	65	DAP	17.6738462	1.5691635
		H	20.0664777	0.6231953
4	28	DAP	21.4464286	1.2621211
		H	21.2016162	0.2506761
5	10	DAP	26.2600000	0.9663218
		H	21.9759446	0.1066607
6	2	DAP	30.8500000	0.0707107
		H	22.3995655	0.0050446

SITIO=2

CLASSE	N Obs	Variable	Mean	Std Dev
1	4	DAP	8.7250000	0.3403430
		H	9.6944718	1.0972422
2	41	DAP	12.5146341	1.3576747
		H	16.5803611	1.4867026
3	60	DAP	17.7400000	1.5974768
		H	20.0884458	0.6395992
4	35	DAP	21.6685714	1.0742361
		H	21.2534504	0.2225752
5	3	DAP	26.7666667	1.4843629
		H	22.0294142	0.1585209
6	2	DAP	30.4500000	0.4949747
		H	22.3700255	0.0367314

Capítulo 15 : PROC MODEL

15.1 INTRODUÇÃO

O procedimento MODEL analisa modelos nos quais as relações entre as variáveis incluem um sistema de uma ou mais equações não-lineares. Usos primários do procedimento MODEL são estimação, simulação, e prognose de modelos simultâneos de equações não-lineares. Este procedimento não está disponível no SAS University Edition. No SAS University Edition está disponível o procedimento NLIN que é equivalente ao procedimento MODEL no ajuste de equações, mas com funcionalidades diferentes e menor número de estatísticas disponíveis.

Um sistema de equações pode ser não-linear quanto aos parâmetros, quanto às variáveis, ou ambos, quanto aos parâmetros e variáveis.

Um sistema não-linear quanto aos parâmetros significa que não é requerido que a relação matemática entre as variáveis e os parâmetros tenha forma linear. Um modelo linear pode ser considerado como um caso especial de modelo não-linear. Portanto, este procedimento pode ser utilizado para ajustar equações lineares.

Um modelo simples não-linear tem a forma:

$$y = f(x,\theta) + \varepsilon$$

Em que: y é a variável dependente, x é um vetor de variáveis exógenas, θ é um vetor de parâmetros correspondentes a x e ε é o erro.

Para estimar parâmetros desconhecidos de um modelo não-linear utilizando o procedimento MODEL, faça o seguinte:

- Use o procedimento MODEL com a opção DATA= para especificar o arquivo de dados contendo os pares de valores de y e x ;
- Escreva a equação para o modelo usando declarações SAS, incluindo todos os parâmetros, variáveis e operadores aritméticos, deixando de fora o termo de erro;
- Use a declaração FIT seguida do nome da variável dependente para ajustar o modelo de equação para os dados de entrada para determinar os parâmetros θ .

O PROC MODEL inclui os seguintes métodos para estimativa de parâmetros e as seguintes possibilidades e ferramentas:

- Ordinary Least Squares* (OLS) – Mínimos quadrados ordinários;

- ❑ *Two-Stage Least Squares (2SLS)* – Mínimos quadrados em dois estágios;
- ❑ *Seemingly Unrelated Regression (SUR) and iterative SUR (ITSUR)* – Regressão aparentemente não relacionada (SUR) e SUR interativa;
- ❑ *Three-Stage Least Squares (3SLS) and iterative 3SLS (IT3SLS)* - Mínimos quadrados em três estágios (3SLS) e 3SLS interativa;
- ❑ *Generalized Method of Moments (GMM)* – Método de momentos generalizado;
- ❑ *Full Information Maximum Likelihood (FIML)* - Informação completa de Máxima Verossimilhança;
- ❑ *Simulation and forecasting capabilities* – Capacidades de simulação e prognoses;
- ❑ *Monte Carlo simulation* – Simulação Monte Carlo;
- ❑ *Goal seeking solutions* – Busca de solução de metas;
- ❑ *Programação de declarações SAS para definir sistemas de equações não-lineares simultâneas;*
- ❑ *Ferramentas para análise da estrutura de sistemas de equações não lineares simultâneas;*
- ❑ *ARIMA, PDL e outras capacidades de modelagem dinâmica;*
- ❑ *Ferramentas para estimar e especificar a estrutura do erro de covariância;*
- ❑ *Ferramentas para estimar e solucionar equações diferenciais ordinárias.*

Um sistema geral de equações não-lineares pode ser escrito como:

$$\begin{aligned}
 & q_1 (y_{1,t} + y_{2,t}, \dots, y_{g,t}, x_{1,t}, x_{2,t}, \dots, x_{m,t}, \theta_1, \theta_2, \dots, \theta_p) \\
 & = \varepsilon_{1,t} q_2 (y_{1,t} + y_{2,t}, \dots, y_{g,t}, x_{1,t}, x_{2,t}, \dots, x_{m,t}, \theta_1, \theta_2, \dots, \theta_p) = \varepsilon_{2,t} : \\
 & q_g (y_{1,t} + y_{2,t}, \dots, y_{g,t}, x_{1,t}, x_{2,t}, \dots, x_{m,t}, \theta_1, \theta_2, \dots, \theta_p) = \varepsilon_{g,t}
 \end{aligned}$$

Em que: where $y_{i,t}$ é uma variável endógena, $x_{i,t}$ é uma variável exógena, θ_i é um parâmetro e ε é o erro desconhecido. O subscripto t representa o tempo ou outro subscripto para os dados.

O modelo pode ser escrito na forma de vetor como:

$$q(y_t, x_t, \theta) = \varepsilon_t$$

O procedimento Model permite as duas formas: matricial e vetorial.

Neste procedimento é possível o uso de muitas declarações do tipo que se usa nos passos de programação de manipulação e criação de arquivos, os DATA STEP. Podem ser utilizados laços com DO / END, declarações de atribuição com expressões matemáticas, declarações de tomada de decisão com IF / THEN / ELSE, etc.

No caso de dados em série temporal com erros correlacionados, o erro referente aos parâmetros do sistema é sobre-estimado.

Outras instruções para uso do procedimento e para ajuste de equações não-lineares podem ser encontradas no manual “SAS/ETS Procedures User’s Guide – The

Model Procedure", disponível em < <https://support.sas.com/documentation/onlinedoc/ets/142/model.pdf> >.

15.2 VALIDAÇÃO DE MODELOS

O procedimento MODEL é particularmente útil para realizar a validação de modelos, verificando a homocedasticidade da variância, a independência dos resíduos e a normalidade na distribuição dos resíduos.

15.2.1 HOMOCEASTICIDADE DA VARIÂNCIA

Uma das principais pressuposições para os mínimos quadrados da regressão usual é a homogeneidade da variância (homocedasticidade). Se o modelo for bem-ajustado, não deveria haver nenhum padrão para os resíduos delineados (plotados) contra os valores ajustados. Se a variância dos resíduos não é constante, então é dito que a variância residual é "heterocedástica". Há métodos gráficos e não-gráficos para detectar heterocedasticidade. Um método gráfico geralmente usado é delinear (plotar) os resíduos contra valores ajustados. O SAS calcula os resíduos e valores ajustados através dos procedimentos GLM, REG e NLIN, que podem ser apresentados em um gráfico. Quando os resíduos são distribuídos sem nenhum padrão, não há heterocedasticidade.

Um método matemático para determinar se há homogeneidade de variância dos resíduos e que é possível de ser executado através do SAS é o teste de White (SAS Institute, 2004). O teste de White é computado achando nR^2 de uma regressão de e_i^2 sobre todas as variáveis distintas em $X \times X$, onde X é o vetor de variáveis dependentes incluindo uma constante. Esta estatística é distribuída assintoticamente como Qui-quadrado (χ^2) com $k-1$ graus de liberdade, onde k é o número de regressores.

O método testa a hipótese nula de que a variância residual é homogênea. Então, se o valor "p" for muito pequeno, a hipótese é rejeitada e aceita-se a hipótese alternativa de que a variância não é homogênea. Para isso é utilizada a opção "SPEC" na declaração do modelo conforme o exemplo a seguir:

```
PROC REG;  
MODEL Y = X / SPEC;
```

O teste também pode ser executado através da opção WHITE da declaração FIT do procedimento MODEL no SAS, como no exemplo a seguir:

```
PROC MODEL;  
PARMS A B C;
```

```

Y = A + B * X1 + C * X2;
FIT Y / WHITE;

```

15.2.2 INDEPENDÊNCIA DOS RESÍDUOS

O valor da estatística “d” de Durbin-Watson (SAS Institute, 2004) é obtido através da opção CLM da declaração MODEL do procedimento GLM do SAS, ou ainda da opção DWPROB da declaração FIT do procedimento MODEL, ou ainda da opção DW da declaração MODEL do procedimento REG, conforme os exemplos abaixo:

```

PROC REG;
MODEL Y=X1 X2 / dw;
ou

```

```

PROC MODEL;
PARMS A B C;
Y = A + B * X1 + C * X2;
FIT Y / DWPROB;

```

É esperado que a estatística “d” seja aproximadamente igual a 2, se os resíduos forem independentes. Caso contrário, se os resíduos forem correlacionados positivamente, tenderá a ser próxima de 0 (zero), ou próxima de 4, se os resíduos forem correlacionados negativamente (Nemec, 1996).

O valor de d é dado por:

$$d = \frac{\sum_{i=2}^n (E_i - E_{i-1})^2}{\sum_{i=1}^n E_i^2}$$

Em que: d = estatística “d” de Durbin-Watson; E_i = erro estocástico = ; n = número de observações; \hat{Y}_i = valor estimado.

15.2.3 NORMALIDADE DA DISTRIBUIÇÃO DOS RESÍDUOS

O princípio deste teste baseia-se na comparação da curva da frequência cumulativa dos dados com a função de distribuição teórica em hipótese. Quando as duas curvas se sobrepõem a estatística de teste é calculada através da máxima diferença entre ambas. A magnitude da diferença é estabelecida segundo a distribuição de probabilidade dessa estatística, que se encontra tabelada. Se os dados experimentais se afastam significativamente do que é esperado da distribuição em hipótese, então as curvas obtidas devem encontrar-se igualmente afastadas e, por um raciocínio análogo, se o ajustamento ao modelo hipotético é admissível, então as curvas têm um desenvolvimento próximo.

A estatística (D) de Kolmogorov-Smirnov (SAS Institute, 2004) é uma estatística do tipo Função de Distribuição Empírica (EDF). A Função de Distribuição Empírica (EDF)

é definida para um conjunto de n observações independentes X_1, \dots, X_n com uma função de distribuição comum $F(x)$. Sob a hipótese de nulidade, $F(x)$ é a distribuição normal. As observações são ordenadas da menor para a maior como $X(1), \dots, X(n)$.

A função de distribuição empírica $F_n(x)$, é definida como:

$$\begin{aligned} F_n(x) &= 0, x < X(1) \\ F_n(x) &= i/n, X(i) \leq x < X(i+1), \quad i = 1, 2, \dots, n-1 \\ F_n(x) &= 1, x(n) \leq x \end{aligned}$$

Note-se que $F_n(x)$ é uma função sequencial que avança em $[1/n]$ a cada observação. Esta função calcula a função de distribuição $F(x)$. A qualquer valor x , $F_n(x)$ é a proporção de observações menor ou igual a x , enquanto $F(x)$ é a probabilidade de uma observação ser menor ou igual a x . Estatísticas de EDF medem a discrepância entre $F_n(x)$ e $F(x)$. As fórmulas computacionais para as estatísticas de EDF fazem uso da transformação da integral de probabilidade $U=F(X)$. Se $F(X)$ é a função de distribuição de X , a variável aleatória U é distribuída uniformemente entre 0 e 1.

Dadas n observações de $X(1), \dots, X(n)$, os valores $U(i)=F(X(i))$ são computados como mostrado a seguir. A estatística (D) de Kolmogorov-Smirnov é baseada na maior diferença vertical entre $F(x)$ e $F_n(x)$, sendo definida como:

$$D = \sup_x |F_n(x) - F(x)|$$

A estatística de Kolmogorov-Smirnov é computada como o máximo de D^+ e D^- , onde D^+ é a maior distância vertical entre o EDF e a função de distribuição quando o EDF é maior que a função de distribuição e D^- é a maior distância vertical quando o EDF for menor que a função de distribuição.

$$\begin{aligned} D^+ &= \max_i (i/n) - U(i) \\ D^- &= \max_i (U(i) - (i-1)/n) \\ D &= \max(D^+, D^-) \end{aligned}$$

O procedimento CAPABILITY do SAS usa a estatística D de **Kolmogorov** modificada para testar os dados contra a distribuição normal com média e variância igual à média e variância da amostra. No procedimento MODEL a estatística é utilizada somente para amostras acima de 2000 indivíduos. No caso de pequenas amostras é utilizado o teste de **Shapiro-Wilk** descrito a seguir, em substituição ao teste de Kolmogorov-Smirnov. No procedimento MODEL o teste de normalidade é obtido pela opção NORMAL da declaração FIT como no exemplo:

```
PROC MODEL;
  PARMs A B C;
  Y=A+B*x1+C*x2;
  FIT Y / NORMAL;
```

A estatística W de Shapiro-Wilk (IU, 2004) é a razão entre o melhor estimador da variância e a soma dos quadrados corrigidos do estimador da variância dos dados coletados. O valor é positivo e menor do que 1, sendo tanto mais próximo da normalidade quanto mais próximo de 1. A estatística W requer valores entre 7 e 2000 unidades, sendo o padrão para pequenas amostras ($n < 2000$) através do procedimento UNIVARIATE do SAS, que usa o teste de Kolmogorov-Smirnov para amostras com tamanho superior a 2000 unidades. Um valor significativo para o valor de W indica falta de normalidade para a variável analisada (Anjos, 2003). O valor de W é calculado como segue (IU, 2004):

$$W = \frac{\sum (a_i x_{(i)})^2}{\sum (x_i - \bar{x})^2}$$

Onde $a_i = (a_1, a_2, \dots, a_n) = m' \cdot V^{-1} [m' \cdot V^{-1} \cdot V^{-1} \cdot m']^{-1/2}$; $m' = (m_1, m_2, \dots, m_n)$ é o vetor de valores esperados da estatística de ordem normal; V é a matriz de covariância n por n ; $x' = (x_1, x_2, \dots, x_n)$ é uma amostra aleatória e $x(1) < x(2) < \dots < x(n)$.

Sintaxe

```

PROC MODEL OPÇÕES;
ABORT ;
ARRAY ARRAYNOME VARIÁVEIS ... ;
ATTRIB LISTADEVARIÁVEIS LISTADEATRIBUTOS
  [LISTADEVARIÁVEIS LISTADEATRIBUTOS];
BOUNDS LIMITE1, LIMITE2 ... ;
BY VARIÁVEIS;
CALL NOME [( EXPRESSÃO [, EXPRESSÃO ... ] ) ] ;
CONTROL VARIÁVEL [ VALOR ] ... ;
DELETE ;
DO [VARIÁVEL = EXPRESSÃO [ TO EXPRESSÃO ] [ BY EXPRESSÃO ]
  [, EXPRESSÃO [ TO EXPRESSÃO ] [ BY EXPRESSÃO ] ... ]
  [ WHILE EXPRESSÃO ] [ UNTIL EXPRESSÃO ] ;
END ;
DROP VARIÁVEL ... ;
ENDOGENOUS VARIÁVEL [ VALORESINICIAIS ] ... ;
ERRORMODEL NOME DA EQUAÇÃO DISTRIBUIÇÃO
  [ CDF=(CDF(OPÇÕES)) ];
ESTIMATE ITEM [ , ITEM ... ] [ / OPÇÕES ] ;
EXOGENOUS VARIÁVEL [ VALORESINICIAIS ] ... ;
FIT EQUATIONS [ PARMS=(VALORES DOS PARÂMETROS ... ) ]
  START=(VALORES DOS PARÂMETROS... )
  [ DROP=(PARÂMETROS) ] [ / OPÇÕES ] ;
FORMAT VARIÁVEL [ FORMATO ] [ DEFAULT = FORMATOPADRÃO ] ;
GOTO RÓTULO DE DECLARAÇÃO ;
ID VARIÁVEL ;
IF EXPRESSÃO ;
IF EXPRESSÃO THEN PROGRAMMING_STATEMENT ;
  ELSE PROGRAMMING_STATEMENT ;
VARIABLE = EXPRESSÃO ;
VARIABLE + EXPRESSÃO ;
INCLUDE ARQUIVOS DE MODELOS ... ;
INSTRUMENTS [ INSTRUMENTOS ] [_ EXOG_ ]
  [ EXCLUDE=(PARÂMETROS) ] [ / OPÇÕES ] ;
KEEP VARIÁVEL ... ;

```

```

 LABEL VARIÁVEL ='LABEL' ... ;
 LENGTH VARIÁVEIS [ $ ]
 COMPRIMENTO ...[DEFAULT=COMPRIMENTO ];
 LINK DECLARAÇÃO_LABEL ;
 OUTVARS VARIÁVEL ... ;
 PARAMETERS VARIÁVEL [ VALOR ] VARIÁVEL [ VALOR ] ... ;
 PUT IMPRIMIR_ITEM ... [ @ ] [ @@ ] ;
 RANGE VARIÁVEL [ = FIRST ] [TO LAST ];
 RENAME NOMEANTIGO =NOMENOVO ... [ NOMEANTIGO =NOMENOVO ];
 RESET OPÇÕES;
 RESTRICT RESTRIÇÃO1 [ , RESTRIÇÃO2 ... ];
 RETAIN VARIÁVEIS VALORES [ VARIÁVEIS VALORES... ] ;
 RETURN ;
 SOLVE VARIÁVEIS [SATISFY=(EQUAÇÕES) ] [ / OPÇÕES ] ;
 SUBSTR( VARIÁVEL, ÍNDICE, COMPRIMENTO ) = EXPRESSÃO ;
 SELECT [ ( EXPRESSÃO ) ] ;
 OTHERWISE PROGAMAÇÃO DA DECLARAÇÃO ;
 STOP ;
 TEST [ "NOME" ] TESTE1 [ , TESTE2 ... ] [ / OPÇÕES ] ;
 VAR VARIÁVEL [ VALORES INICIAIS ] ... ;
 WEIGHT VARIÁVEL;
 WHEN ( EXPRESSÃO ) PROGAMAÇÃO DA DECLARAÇÃO ;

```

As funções a seguir utilizam a palavra chave em inglês: *Lag*, que significa anterior em português.

As funções do PROC MODEL, para tratamento de dados de séries temporais e expressões em períodos anteriores, são as seguintes:

LAGn(i , x) – Retorna o i-ésimo anterior de x, onde n é o máximo anterior (último antes do atual);

DIFn(x) – Diferença de x no anterior n;

ZLAGn(i , x) - Retorna o i-ésimo anterior de x, onde n é o máximo anterior (último antes do atual), com anteriores perdidos substituídos por zero.

ZDIFn(x) - Diferença com o comprimento do anterior truncado e valores perdidos convertidos para zero.

MOVAVGn(x) – A amplitude da média móvel é n e x é uma expressão. O argumento i é uma variável ou expressão para computar a média móvel de. Os valores perdidos de x são omitidos no cálculo da média.

Em todas as funções, n representa o número de períodos e x é uma expressão. O argumento i é uma variável ou expressão com o comprimento dos anteriores ($0 \leq i \leq n$), se o valor do índice i for omitido, o máximo comprimento anterior n é usado.

15.3 VALORES PERDIDOS NO PROC MODEL

Uma observação é excluída das estimativas se qualquer variável usada para ajustar a equação apresentar valor perdido.

O uso do procedimento MODEL é demonstrado no “Exemplo – Ajuste de uma função linear e estatísticas de validação do modelo.

```

* EXEMPLO - Ajuste de uma funcao linear e estatísticas de
  validacao do modelo;
DATA dados;
INPUT Bloco TratAMENTO ARVORE H DBase DAP Volume;

```

```

1
2
3
4

```

```

D2H=DAP**2*h;
DATALINES;
1 0 14 21.05 31.5 26.1 0.53171
1 0 56 18.80 21.6 16.6 0.24366
1 0 65 21.00 29.3 23.6 0.51740
1 40 3 19.80 23.2 17.8 0.29079
1 40 15 19.30 26.7 21.0 0.37148
1 40 50 20.08 30.2 25.8 0.59485
1 60 42 17.00 18.1 13.4 0.12305
1 60 55 21.80 28.3 23.6 0.61318
1 60 60 22.10 24.8 20.4 0.40231
1 80 51 21.60 29.0 23.2 0.50209
1 80 62 17.30 18.5 14.3 0.16202
1 80 67 17.50 22.0 18.5 0.28919
2 0 43 17.80 25.5 22.0 0.30988
2 0 55 20.50 29.0 23.6 0.53559
2 0 62 16.70 19.7 16.9 0.17384
2 40 47 21.50 29.0 22.6 0.50604
2 40 56 20.30 29.3 24.8 0.57305
2 40 65 19.30 21.6 17.8 0.26287
2 60 18 19.60 21.0 15.9 0.23197
2 60 33 19.70 26.1 21.0 0.39317
2 60 58 19.50 30.2 23.9 0.52865
2 80 21 21.50 26.1 21.3 0.42406
2 80 50 17.70 28.6 22.9 0.50028
2 80 67 19.90 18.5 14.6 0.18820
3 0 27 19.30 19.7 16.6 0.21111
3 0 41 21.00 28.3 22.6 0.45431
3 0 47 20.90 29.6 23.2 0.46010
3 40 25 20.70 29.9 22.9 0.48164
3 40 40 17.20 21.6 15.6 0.18800
3 40 42 18.10 17.2 13.1 0.14907
3 60 53 20.50 27.7 21.3 0.45549
3 60 58 20.10 29.3 25.1 0.55054
3 60 59 18.00 19.1 16.2 0.19369
3 80 19 17.30 19.4 15.3 0.15639
3 80 35 18.60 22.6 18.8 0.29944
3 80 50 21.10 28.3 23.2 0.52996
4 0 20 20.10 30.2 20.4 0.42882
4 0 46 21.50 34.7 25.1 0.61833
4 0 69 17.40 20.7 17.2 0.21127
4 40 19 18.00 22.9 18.1 0.25424
4 40 25 21.00 29.9 23.6 0.55125
4 40 42 20.70 24.8 21.0 0.39309
4 60 7 15.60 18.8 15.6 0.13469
4 60 18 20.30 22.3 17.5 0.31869
4 60 28 21.00 28.3 21.3 0.36157
4 80 5 17.70 19.1 15.0 0.19278
4 80 27 19.10 24.5 18.5 0.32424
4 80 48 20.00 31.5 24.8 0.64111
;
PROC MODEL DATA=DADOS;
  TITLE 'AJUSTE DA EQUACAO DE SPURR PARA VOLUME';
  TITLE2 'E VALIDACAO DO MODELO';
  PARS B0 B1;
  VOLUME=B0+B1*D2H;
  FIT VOLUME / WHITE DW NORMAL;
RUN;
QUIT;

```

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

Como resultado é impresso o relatório a seguir:

AJUSTE DA EQUACAO DE SPURR PARA VOLUME
E VALIDACAO DO MODELO
The MODEL Procedure

Model Summary	
Model Variables	1
Parameters	2
Equations	1
Number of Statements	1

Model Variables	VOLUME
Parameters	B0 B1
Equations	VOLUME

The Equation to Estimate is	
VOLUME =	F(B0(1), B1(D2H))

NOTE: At OLS Iteration 1 CONVERGE=0.001 Criteria Met.

AJUSTE DA EQUACAO DE SPURR PARA VOLUME
E VALIDACAO DO MODELO
The MODEL Procedure
OLS Estimation Summary

Data Set Options	
DATA=	DADOS

Minimization Summary	
Parameters Estimated	2
Method	Gauss
Iterations	1

Final Convergence Criteria	
R	0
PPC	0
RPC(B0)	53.08961
Object	0.994606
Trace(S)	0.001362
Objective Value	0.001305

Observations Processed	
Read	48
Solved	48

AJUSTE DA EQUACAO DE SPURR PARA VOLUME
E VALIDACAO DO MODELO
The MODEL Procedure

Nonlinear OLS Summary of Residual Errors								
Equation	DF Model	DF Error	SSE	MSE	Root MSE	R-Square	Adj R-Sq	Durbin Watson
VOLUME	2	46	0.0627	0.00136	0.0369	0.9446	0.9434	1.8840

Nonlinear OLS Parameter Estimates				
Parameter	Estimate	Approx Std Err	t Value	Approx Pr > t
B0	-0.00526	0.0145	-0.36	0.7177
B1	0.000045	1.623E-6	28.01	<.0001

Number of Observations		Statistics for System	
Used	48	Objective	0.001305
Missing	0	Objective*N	0.0627

AJUSTE DA EQUACAO DE SPURR PARA VOLUME
E VALIDACAO DO MODELO
The MODEL Procedure

Heteroscedasticity Test					
Equation	Test	Statistic	DF	Pr > ChiSq	Variables
VOLUME	White's Test	7.11	2	0.0285	Cross of all vars

Normality Test			
Equation	Test Statistic	Value	Prob
VOLUME	Shapiro-Wilk W	0.96	0.1855
System	Mardia Skewness	0.70	0.4032
	Mardia Kurtosis	2.34	0.0194
	Henze-Zirkler T	1.80	0.0726

Capítulo 16 : PROC PLOT / PROC GPLOT

Os procedimentos PLOT e GPLOT produzem gráficos de uma variável contra outra. Cada ponto no gráfico corresponde a valores de duas variáveis. A descrição completa do procedimento é realizada no manual “*SAS/GRAPH software User’s Guide*”. Estes procedimentos não estão disponíveis no SAS University Edition.

Os dois procedimentos são semelhantes. PLOT produz gráficos de baixa resolução e GPLOT, mais atual, produz gráficos de alta resolução e possui mais opções que seu correlato, que é mantido somente para compatibilizar programas antigos.

16.1 SINTAXE

```
PROC GPLOT <DATA=INPUT-DATA-SET>
  <ANNOTATE=ANNOTATE-DATA-SET>
  <GOUT=<LIBREF.>OUTPUT-CATALOG>
  <IMAGEMAP=OUTPUT-DATA-SET>
  <UNIFORM>;
  BUBBLE PLOT-REQUEST(S) </OPTION(S)>;
  BUBBLE2 PLOT-REQUEST(S) </OPTION(S)>;
  PLOT PLOT-REQUEST(S) </OPTION(S)>;
  PLOT2 PLOT-REQUEST(S) </OPTION(S)>;
```

16.2 OPÇÕES DO PROCEDIMENTO PROC GPLOT

DATA= especifica o arquivo de entrada.

FORMCHAR = especifica os caracteres para usar na construção do eixo

HPERCENT = especifica a % da página horizontal a usar para cada gráfico.

VPERCENT = especifica a % da página vertical a usar para cada gráfico.

Por exemplo, a seguinte declaração significa que os eixos dos gráficos serão linhas e que cada gráfico usará somente 50% da página.

```
PROC PLOT FORMCHAR='|---|+|---' VPERCENT=50;
```

16.3 INSTRUÇÕES DO PROCEDIMENTO

BUBBLE - Cria gráficos de bolha nos quais uma terceira variável é delineada contra duas variáveis representadas pelos eixos horizontais e verticais; o valor da terceira variável controla o tamanho da bolha. Exigências: Pelo menos uma solicitação de gráfico é requerida. Instruções globais: AXIS, FOOTNOTE, TITLE (EIXO, NOTA DE RODAPÉ, TÍTULO).

PLOT – Cria gráficos em que uma variável independente é delineada no eixo horizontal e uma variável dependente no eixo vertical. Exigências: Pelo menos uma solicitação de gráfico é requerida. Instruções globais: AXIS, FOOTNOTE, LEGEND, PATTERN, SYMBOL, TITLE (EIXO, NOTA DE RODAPÉ, LEGENDA, PADRÃO, SÍMBOLO, TÍTULO). Suporta: funcionalidade sequencial inferior.

Opções da instrução PLOT

■ PLOT GRÁFICOREQUERIDO </ OPÇÕES>;

Nota: Não confundir as opções da instrução com as opções do procedimento PLOT/ GPLOT.

GráficoRequerido - Define o cruzamento das variáveis $y * x$, ou $y * x = \text{'Símbolo'}$, ou $y * x = z$, em que x é a variável independente, y é a variável dependente e z é uma terceira variável com resultados de uma função do tipo $y=f(x)$.

CAXIS=CorDoEixo – Define a cor dos eixos cartesianos.

HAXIS=ListaDeValores – Especifica os valores para a marcação do eixo horizontal.

VAXIS=ListaDeValores – Especifica os valores para a marcação do eixo vertical.

OVERLAY – Permite que um gráfico seja desenhado sobre o outro sem apagar o que fica embaixo.

AREAS= n – Preenche a área entre duas linhas com um padrão. AREAS=1 preenche a primeira área.

AREAS=2 preenche a primeira e segunda áreas e assim por diante. Funciona somente se a opção INTERPOL= for utilizada.

GRID – Desenha uma grade com os pontos principais dos dois eixos vertical e horizontal.

INTERPOL=JOIN – Transforma o gráfico em linhas. Não se aplica automaticamente para gráficos com múltiplas linhas que devem ser desenhadas com formas diferentes. Neste caso, quando se quer que as linhas sejam diferentes, é necessário utilizar uma instrução $SYMBOL_n=$ para definir que símbolo deve ser utilizado para cada linha, onde n é o número da linha (Exemplo: `symbol4 value=star cv=blue interpol=join;`). Quando as linhas podem ser iguais, não é necessário utilizar a instrução SYMBOL.

LEGEND | LEGEND=LEGEND<1...99> - Acrescenta legendas ao gráfico com os símbolos utilizados.

NOLEGEND - Suprime a legenda do gráfico.

HAXIS=ListaDeValores | AXIS<1...99> – Especifica os valores das marcas de divisão do eixo horizontal.

HREF=ListaDeValores – Especifica a localização de linhas de referência perpendicular ao eixo horizontal.

HZERO – Especifica que a primeira marca sobre a linha horizontal inicia em zero.

HAXIS=ListaDeValores | AXIS<1...99> – Especifica os valores das marcas de divisão do eixo horizontal.

VREF=ListaDeValores – Especifica a localização de linhas de referência perpendicular ao eixo vertical.

VZERO – Especifica que a primeira marca sobre a linha vertical inicia em zero.

Outras opções estão disponíveis para esta instrução.

16.3.1 CARACTERÍSTICAS DOS GRÁFICOS GERADOS POR PLOT

As características dos gráficos gerados pela instrução PLOT são relacionadas a seguir:

- Area 1, area 2, ... - Área abaixo da linha 1, área abaixo da linha 2 e acima da 1, etc.
- Axis area – Área dentro do espaço de delineamento do gráfico.
- Frame - Quadro.
- Horizontal axis label – Rótulo do eixo horizontal.
- Major ticks marks – Marcas de divisão principal do eixo.
- Minor ticks marks – Marcas de divisão secundária do eixo.
- Offset – Área fora da área de delineamento do gráfico e dentro do quadro.
- Plot line – Linha desenhada.

- Plot symbol – Símbolo de pontos de delineamento dos cruzamentos y * x.
- Reference line – Linha de referência.
- Vertical axis label – Rótulo do eixo vertical.

As Figuras 7 e 8 apresentam as características dos gráficos de linha e área desenhados pelo procedimento com a instrução PLOT.

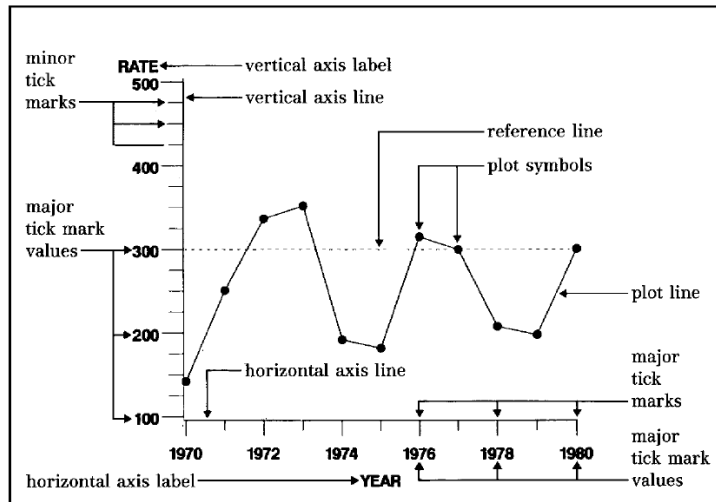


Figura 41: - Características dos gráficos gerados por PLOT.

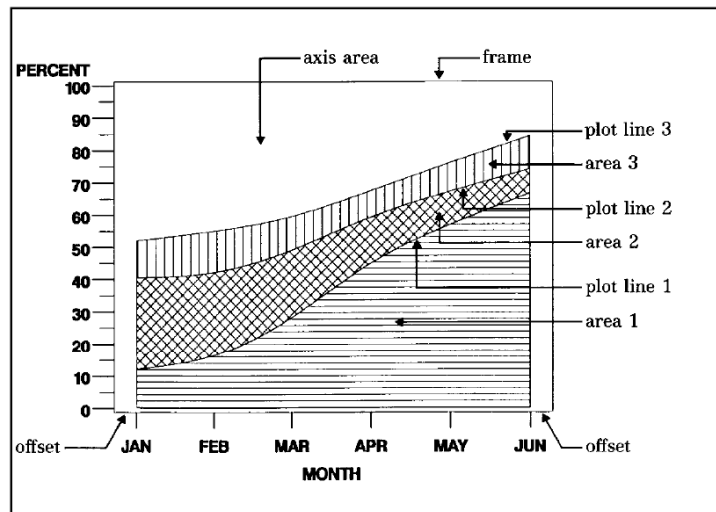


Figura 42: - Características adicionais dos gráficos gerados por PLOT.

No “Exemplo – Gráficos resultantes de produto cartesiano e gráficos de linhas” são utilizados os dados de 48 árvores cubadas, provenientes de um experimento de

desrama de *Pinus elliottii* Engelman. com quatro tratamentos correspondentes a 0%, 40%, 60% e 80% da altura total das árvores desramadas.

```

* EXEMPLO - Graficos resultantes de produto cartesiano e          1
  graficos de linha;                                           2
DATA dados;                                                     3
INPUT Bloco TratAMENTO ARVORE H DBase DAP VolUME;              4
CLASSE=INT(DAP/5);                                             5
DATALINES;                                                     6
1 0 14 21.05 31.5 26.1 0.53171                                7
1 0 56 18.80 21.6 16.6 0.24366                                8
...                                                             9
...                                                            10
...                                                            11
4 80 48 20.00 31.5 24.8 0.64111                                12
;                                                                13
PROC PRINT;                                                    14
*****;                                                        15
TITLE '1 - ALTURA * DAP, DE TODAS AS ARVORES';              16
PROC GPLOT DATA=DADOS;                                        17
  PLOT H * DAP = "*";                                         18
RUN;                                                            19
*****;                                                        20
TITLE '2 - ALTURA * CLASSE DAP = VOLUME';                    21
PROC SORT DATA=DADOS; BY CLASSE;                              22
PROC MEANS DATA=DADOS NOPRINT;                                23
  VAR DAP H VOLUME;                                           24
  BY CLASSE;                                                  25
  OUTPUT OUT=MCLASSE MEAN(DAP)=DAP MEAN(H)=H                 26
  MEAN(VOLUME)=VOLUME;                                        27
PROC PRINT;                                                    28
PROC GPLOT DATA=MCLASSE;                                       29
  PLOT H * CLASSE = VOLUME;                                    30
RUN;                                                            31
*****;                                                        32
TITLE '3 - DAP * TRATAMENTO';                                  33
PROC SORT DATA=DADOS; BY TRATAMENTO;                          34
PROC MEANS DATA=DADOS NOPRINT;                                35
  VAR DAP;                                                    36
  BY TRATAMENTO;                                              37
  OUTPUT OUT=MTRAT MEAN(DAP)=DAP;                              38
PROC PRINT;                                                    39
PROC GPLOT DATA=MTRAT;                                         40
  SYMBOL VALUE=STAR CV=BLUE INTERPOL=JOIN;                    41
  PLOT DAP * TRATAMENTO;                                       42
RUN;                                                            43
*****;                                                        44
TITLE '4 - ALTURA * DAP POR TRATAMENTO';                      45
PROC SORT DATA=DADOS; BY TRATAMENTO CLASSE;                  46
PROC MEANS DATA=DADOS NOPRINT;                                47
  VAR DAP H;                                                  48
  BY TRATAMENTO CLASSE;                                       49
  OUTPUT OUT=MTRATCLS MEAN(DAP)=DAP MEAN(H)=H;                50
PROC PRINT;                                                    51
PROC GPLOT DATA=MTRATCLS;                                       52
  SYMBOL1 VALUE=STAR CV=BLUE INTERPOL=JOIN;                    53
  SYMBOL2 VALUE=TRIANGLE CV=RED INTERPOL=JOIN;                 54
  SYMBOL3 VALUE=SQUARE CV=GREEN INTERPOL=JOIN;                 55
  SYMBOL4 VALUE=CIRCLE CV=BLACK INTERPOL=JOIN;                 56
  PLOT H * DAP = TRATAMENTO / OVERLAY;                          57

```

RUN;

58

*****;

59

QUIT;

60

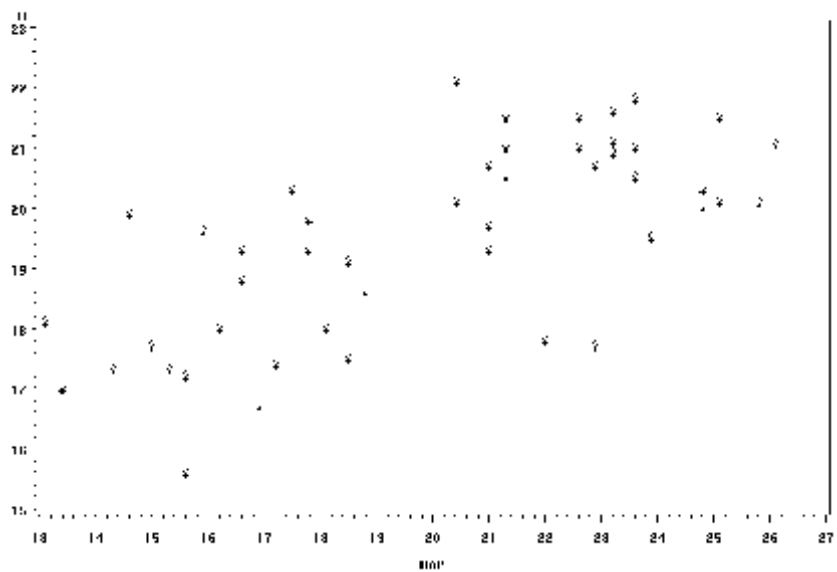
Os relatórios emitidos pelo SAS, com os dados e respectivos gráficos, em sequência, são os seguintes:

1 - ALTURA * DAP, DE TODAS AS ARVORES

Obs	BLOCO	TRATAMENTO	ARVORE	H	DBASE	DAP	VOLUME	CLASSE
1	1	0	14	21.05	31.5	26.1	0.53171	5
2	1	0	56	18.80	21.6	16.6	0.24366	3
3	1	0	65	21.00	29.3	23.6	0.51740	4
4	1	40	3	19.80	23.2	17.8	0.29079	3
5	1	40	15	19.30	26.7	21.0	0.37148	4
6	1	40	50	20.08	30.2	25.8	0.59485	5
7	1	60	42	17.00	18.1	13.4	0.12305	2
8	1	60	55	21.80	28.3	23.6	0.61318	4
9	1	60	60	22.10	24.8	20.4	0.40231	4
10	1	80	51	21.60	29.0	23.2	0.50209	4
11	1	80	62	17.30	18.5	14.3	0.16202	2
12	1	80	67	17.50	22.0	18.5	0.28919	3
13	2	0	43	17.80	25.5	22.0	0.30988	4
14	2	0	55	20.50	29.0	23.6	0.53559	4
15	2	0	62	16.70	19.7	16.9	0.17384	3
16	2	40	47	21.50	29.0	22.6	0.50604	4
17	2	40	56	20.30	29.3	24.8	0.57305	4
18	2	40	65	19.30	21.6	17.8	0.26287	3
19	2	60	18	19.60	21.0	15.9	0.23197	3
20	2	60	33	19.70	26.1	21.0	0.39317	4
21	2	60	58	19.50	30.2	23.9	0.52865	4
22	2	80	21	21.50	26.1	21.3	0.42406	4
23	2	80	50	17.70	28.6	22.9	0.50028	4
24	2	80	67	19.90	18.5	14.6	0.18820	2
25	3	0	27	19.30	19.7	16.6	0.21111	3
26	3	0	41	21.00	28.3	22.6	0.45431	4
27	3	0	47	20.90	29.6	23.2	0.46010	4
28	3	40	25	20.70	29.9	22.9	0.48164	4
29	3	40	40	17.20	21.6	15.6	0.18800	3
30	3	40	42	18.10	17.2	13.1	0.14907	2
31	3	60	53	20.50	27.7	21.3	0.45549	4

Obs	BLOCO	TRATAMENTO	ARVORE	H	DBASE	DAP	VOLUME	CLASSE
32	3	60	58	20.10	29.3	25.1	0.55054	5
33	3	60	59	18.00	19.1	16.2	0.19369	3
34	3	80	19	17.30	19.4	15.3	0.15639	3
35	3	80	35	18.60	22.6	18.8	0.29944	3
36	3	80	50	21.10	28.3	23.2	0.52996	4
37	4	0	20	20.10	30.2	20.4	0.42882	4
38	4	0	46	21.50	34.7	25.1	0.61833	5
39	4	0	69	17.40	20.7	17.2	0.21127	3
40	4	40	19	18.00	22.9	18.1	0.25424	3
41	4	40	25	21.00	29.9	23.6	0.55125	4
42	4	40	42	20.70	24.8	21.0	0.39309	4
43	4	60	7	15.60	18.8	15.6	0.13469	3
44	4	60	18	20.30	22.3	17.5	0.31869	3
45	4	60	28	21.00	28.3	21.3	0.36157	4
46	4	80	5	17.70	19.1	15.0	0.19278	3
47	4	80	27	19.10	24.5	18.5	0.32424	3
48	4	80	48	20.00	31.5	24.8	0.64111	4

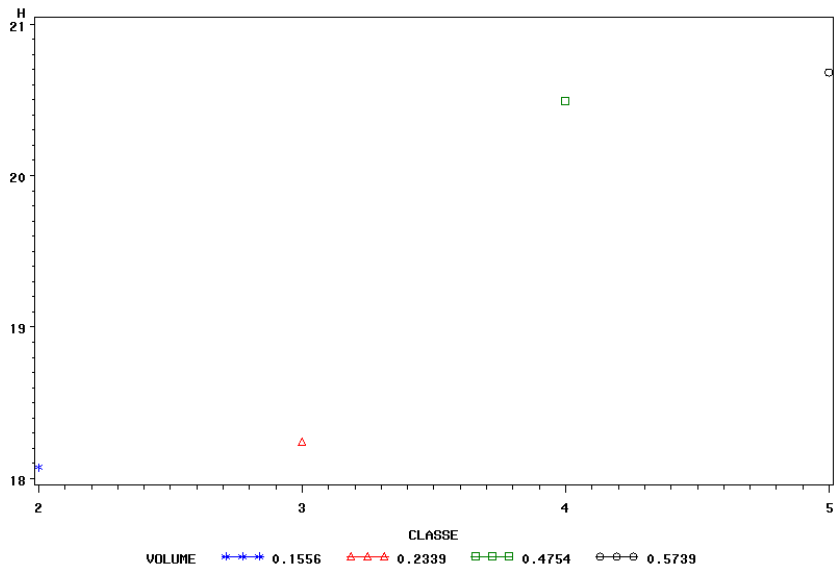
1 — ALTURA * DAP, DE TODAS AS ARVORES



2 - ALTURA * CLASSE DAP = VOLUME

Obs	CLASSE	_TYPE_	_FREQ_	DAP	H	VOLUME
1	2	0	4	13.8500	18.0750	0.15559
2	3	0	17	16.9353	18.2471	0.23393
3	4	0	23	22.5304	20.4913	0.47541
4	5	0	4	25.5250	20.6825	0.57386

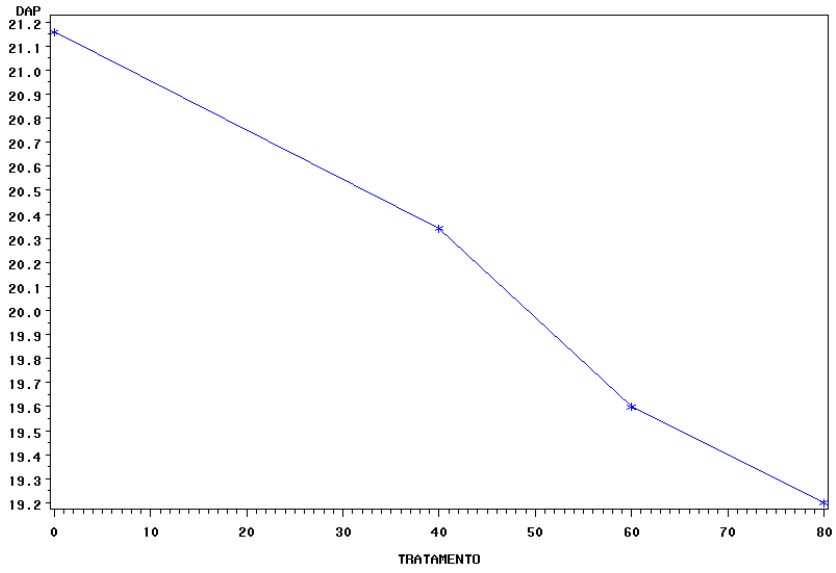
2 - ALTURA * CLASSE DAP = VOLUME



3 - DAP * TRATAMENTO

Obs	TRATAMENTO	_TYPE_	_FREQ_	DAP
1	0	0	12	21.1583
2	40	0	12	20.3417
3	60	0	12	19.6000
4	80	0	12	19.2000

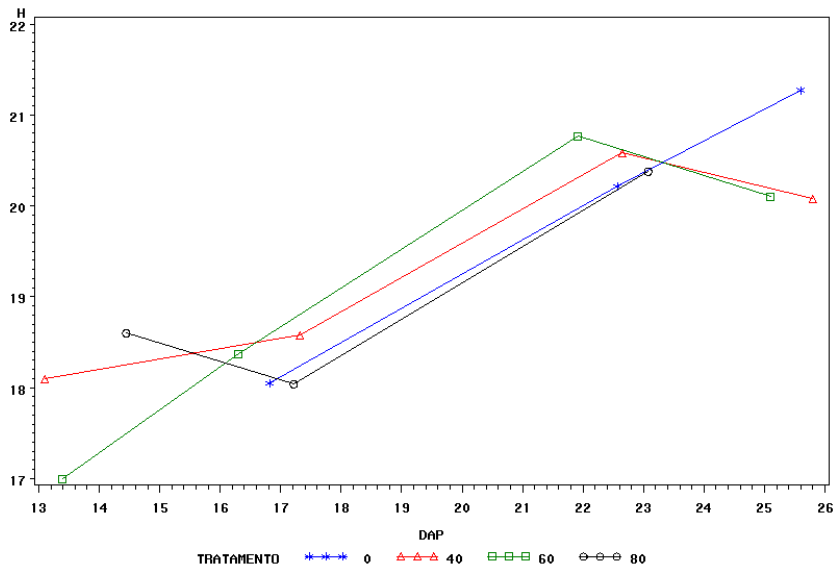
3 – DAP * TRATAMENTO



4 - ALTURA * DAP POR TRATAMENTO

Obs	TRATAMENTO	CLASSE	_TYPE_	_FREQ_	DAP	H
1	0	3	0	4	16.8250	18.0500
2	0	4	0	6	22.5667	20.2167
3	0	5	0	2	25.6000	21.2750
4	40	2	0	1	13.1000	18.1000
5	40	3	0	4	17.3250	18.5750
6	40	4	0	6	22.6500	20.5833
7	40	5	0	1	25.8000	20.0800
8	60	2	0	1	13.4000	17.0000
9	60	3	0	4	16.3000	18.3750
10	60	4	0	6	21.9167	20.7667
11	60	5	0	1	25.1000	20.1000
12	80	2	0	2	14.4500	18.6000
13	80	3	0	5	17.2200	18.0400
14	80	4	0	5	23.0800	20.3800

4 - ALTURA * DAP POR TRATAMENTO



Capítulo 17 : PROC PRINT

O procedimento PROC PRINT imprime as observações contidas em um arquivo SAS, usando todas ou algumas das variáveis. É possível criar relatórios que variam de uma listagem simples a um relatório altamente personalizado que agrupa os dados e calcula totais e subtotais para variáveis numéricas.

As instruções que podem ser utilizadas com o procedimento são as seguintes:

- BY – Produz uma seção separada do relatório para cada variável listada;
- ID – Identifica observações pelos valores formatados das variáveis listadas em vez de através dos números das observação;
- PAGEBY – Controla a ejeção de página pelas variáveis listadas antes de uma página estar completa;
- SUMBY – Limita o número de somas que aparecem no relatório;
- SUM – Apresenta os valores totais de variáveis numéricas;
- VAR – Seleciona variáveis que aparecem no relatório e determina a sua ordem.

17.1 SINTAXE

```
PROC PRINT <OPÇÕES DO PROCEDIMENTO>;
  BY <DESCENDING> VARIÁVEL-1 <...<DESCENDING>
  VARIÁVEL-N><NOTSORTED>;
  PAGEBY BY-VARIÁVEL;
  SUMBY BY-VARIÁVEL;
  ID VARIÁVEL(EIS);
  SUM VARIÁVEL(EIS);
  VAR VARIÁVEL(EIS);
```

17.1.1 OPÇÕES DO PROCEDIMENTO:

Opções de controle dados:

CONTENTS=TextoDeLigação – Especifica texto ligado aos conteúdos de HTML com os dados do arquivo de saída;

DATA=ArquivoDeEntrada – Especifica o arquivo de entrada;

Opções de controle geral de formato:

DOUBLE – Escreve uma linha em branco entre observações;

N="string-1" <"string-2">> – Imprime o número de observações existente no arquivo de dados, ou em grupos BY, ou ambos, e especifica texto explicativo para imprimir com o número;

NOOBS – Suprime a coluna no arquivo de saída que identifica cada observação através de número;

OBS="Cabeçalho" – Especifica um cabeçalho de coluna para a coluna que identifica cada observação através de número;

ROUND – Arredonda valores numéricos não formatados para duas casas decimais; não use formatos PICTURE com esta opção;

Opções de controle do formato de página:

ROWS=FormatoDePágina – Formata as linhas em uma página;

WIDTH=UNIFORM – Usa a largura formatada de cada variável como sua largura de coluna em todas as páginas;

Opções de controle do formato de coluna:

HEADING=Direção – Controla a orientação dos títulos de coluna (HORIZONTAL|VERTICAL); se for usada a opção LABEL, todos os cabeçalhos serão horizontais;

LABEL ou SPLIT= – Usa rótulos (label) de variáveis como títulos de coluna; se não for especificada esta opção, serão impressos os nomes das colunas, mesmo se existir uma instrução LABEL; o sistema necessita ser habilitado pela opção LABEL para que a mesma opção possa ser utilizada por qualquer procedimento;

SPLIT='Caractereseparador' – Especifica o caractere separador que controla quebras de linha em títulos de coluna; se for usado o caractere ponto (.) como separador, não é necessário especificar SPLIT=;

WIDTH= – Determina a largura da coluna para cada variável;

STYLE = <(Localização(ões))> = <ElementoDeEstilo> <[EspecificaçãoDeEstilo(s)]> – Especifica um ou mais elementos de estilo (para o ODS) para usar com diferentes partes do relatório; afeta as partes especificadas do relatório.

17.1.2 BY (INSTRUÇÃO)

Separa a análise por uma ou mais variáveis especificadas.

Sintaxe:

```
■ BY <DESCENDING> VARIÁVEL-1 <...<DESCENDING> VARIÁVEL-N>  
<NOTSORTED>;
```

Variável-1 a n – Especifica as variáveis para formar os grupos.

DESCENDING – Identifica a ordem descendente para a variável subsequente à opção.

NOTSORTED – Especifica que as observações no arquivo não estão em ordem alfabética, mas em uma outra ordem determinada pelo programador.

17.1.3 PAGEBY (INSTRUÇÃO)

Imprime separadamente uma página para cada valor diferente da variável especificada.

Sintaxe:

```
■ PAGEBY VARIÁVEL;
```

Variável - Nome da variável que deve ser indicador de separação de páginas; a variável deve aparecer antes na instrução BY.

17.1.4 SUMBY (INSTRUÇÃO)

Determina uma variável para formar grupos ao apresentar totais.

Sintaxe:

■ **SUMBY VARIÁVEL;**

Variável – Variável de controle de apresentação de totais; a variável deve aparecer antes na instrução BY; é usada em conjunto com a instrução SUM.

17.1.5 ID (INSTRUÇÃO)

Identifica observações pelo uso de valores formatados das variáveis, em vez de usar os números das observações.

Sintaxe:

■ **ID VARIÁVEL(EIS) </STYLE <(LOCALIZAÇÃO(ÕES))> = <ELEMENTODEESTILO> <[ESPECIFICAÇÃODEESTILO(S)]>;**

Variável(eis) – Especifica uma ou mais variáveis a imprimir em vez do número de observações no início de cada linha. Restrição: se o valor da variável for muito longo para caber as colunas subsequentes, o sistema emite uma mensagem de advertência e não trata todas as variáveis da instrução ID como variáveis ID. Se a variável for relacionada também na instrução VAR, o sistema imprime duas colunas com a variável.

STYLE <(Localização(ões))> = <ElementoDeEstilo> <[EspecificaçãoDeEstilo(s)]> - Especifica o elemento de estilo a usar com as colunas ID. Para mais de um estilo, declare outras instruções ID.

17.1.6 SUM (INSTRUÇÃO)

Apresenta os totais para as variáveis numéricas especificadas.

Sintaxe:

■ **SUM VARIÁVEL(EIS) </STYLE <(LOCALIZAÇÃO(ÕES))> = <ELEMENTODEESTILO> <[ESPECIFICAÇÃODEESTILO(S)]>;**

Variável(eis) – Identifica as variáveis numéricas que devem ter totais no relatório.

STYLE <(Localização(ões))> = <ElementoDeEstilo> <[EspecificaçãoDeEstilo(s)]> - Especifica o elemento de estilo a utilizar com as células que contém totais.

17.1.7 VAR (INSTRUÇÃO)

Seleciona as variáveis que serão impressas.

Sintaxe:

■ **VAR VARIÁVEL(EIS) </STYLE <(LOCALIZAÇÃO(ÕES))> = <ELEMENTODEESTILO><[ESPECIFICAÇÃODEESTILO(S)]>;**

Variável(eis) – Nome das variáveis a imprimir. Interage com a função ID.

STYLE <(Localização(ões))> = <ElementoDeEstilo> <[EspecificaçãoDeEstilo(s)]> - Identifica o estilo a utilizar com as variáveis selecionadas pela instrução VAR.

No “Exemplo – Relatório de despesas anuais”, considere-se os custos (fictícios), de uma floresta cultivada com eucalipto, listados na Tabela 5.

TABELA 5 - Custos de uma floresta de eucalipto

Cento de Custos	Valor	Ano da despesa
Infra-estrutura	110,00	1
Combate_a_formiga	22,95	1
Formicida	37,80	1
Limpeza_do_terreno	270,00	1
Preparo_do_solo	127,50	1
Adubacao_manual	20,40	1
Adubos	343,40	1
Plantio_e_replatio	153,00	1
Mudas	219,60	1
Estradas_e_aceiros	50,00	1
Projeto_e_Assist._Técnica	81,28	1
Infra-estrutura	30,00	2
Combate_a_formiga	22,95	2
Formicida	37,80	2
Manutencao	63,75	3
Manutencao	63,75	4
Manutencao	63,75	5
Manutencao	63,75	6
Manutencao	63,75	7
Combate_a_formiga	22,95	3
Formicida	37,80	3
Combate_a_formiga	22,95	4
Formicida	37,80	4
Combate_a_formiga	22,95	5
Formicida	37,80	5
Combate_a_formiga	22,95	6
Formicida	37,80	6
Combate_a_formiga	22,95	7
Formicida	37,80	7

O programa, para ler os dados e criar um arquivo, imprimir os dados classificando-os por ano, com as somas anuais de despesas, pode ser escrito como segue:

```

* EXEMPLO - Relatorio de despesas anuais;
DATA dados;
  length Cento_de_Custos $25.;
  INPUT Cento_de_Custos $ valor ano;
DATALINES;
Infra-estrutura 110.00 1
Combate_a_formiga 22.95 1
Formicida 37.80 1
Limpeza_do_terreno 270.00 1
Preparo_do_solo 127.50 1
Adubacao_manual 20.40 1
Adubos 343.40 1
Plantio_e_replanteio 153.00 1
Mudas 219.60 1
Estradas_e_aceiros 50.00 1
Projeto_e_Assist._Tecnica 81.28 1
Infra-estrutura 30.00 2
Combate_a_formiga 22.95 2
Formicida 37.80 2
Manutencao 63.75 3
Manutencao 63.75 4
Manutencao 63.75 5
Manutencao 63.75 6
Manutencao 63.75 7
Combate_a_formiga 22.95 3
Formicida 37.80 3
Combate_a_formiga 22.95 4
Formicida 37.80 4
Combate_a_formiga 22.95 5
Formicida 37.80 5
Combate_a_formiga 22.95 6
Formicida 37.80 6
Combate_a_formiga 22.95 7
Formicida 37.80 7
;
PROC SORT DATA=DADOS; BY ANO CENTO_DE_CUSTOS;
PROC PRINT DATA=DADOS NOOBS;
  TITLE1 'FLORESTA DE EUCALIPTO';
  TITLE2 'RELATORIO DE DESPESAS ANUAIS';
  VAR ANO CENTO_DE_CUSTOS VALOR;
  SUMBY ANO;
  BY ANO;
*****;
RUN;
QUIT;

```

O seguinte relatório é impresso pelo programa:

FLORESTA DE EUCALIPTO
RELATORIO DE DESPESAS ANUAIS
ANO=1

CENTO_DE_CUSTOS	VALOR
ADUBACAO_MANUAL	20.40
ADUBOS	343.40
COMBATE_A_FORMIGA	22.95
ESTRADAS_E_ACEIROS	50.00

CENTO_DE_CUSTOS	VALOR
FORMICIDA	37.80
INFRA-ESTRUTURA	110.00
LIMPEZA_DO_TERRENO	270.00
MUDAS	219.60
PLANTIO_E_REPLANTIO	153.00
PREPARO_DO_SOLO	127.50
PROJETO_E_ASSIST._TECNICA	81.28
	1435.93

ANO=2

CENTO_DE_CUSTOS	VALOR
COMBATE_A_FORMIGA	22.95
FORMICIDA	37.80
INFRA-ESTRUTURA	30.00
	90.75

ANO=3

CENTO_DE_CUSTOS	VALOR
COMBATE_A_FORMIGA	22.95
FORMICIDA	37.80
MANUTENCAO	63.75
	124.50

ANO=4

CENTO_DE_CUSTOS	VALOR
COMBATE_A_FORMIGA	22.95
FORMICIDA	37.80
MANUTENCAO	63.75
	124.50

ANO=5

CENTO_DE_CUSTOS	VALOR
COMBATE_A_FORMIGA	22.95
FORMICIDA	37.80

CENTO_DE_CUSTOS	VALOR
MANUTENCAO	63.75
	124.50

ANO=6

CENTO_DE_CUSTOS	VALOR
COMBATE_A_FORMIGA	22.95
FORMICIDA	37.80
MANUTENCAO	63.75
	124.50

ANO=7

CENTO_DE_CUSTOS	VALOR
COMBATE_A_FORMIGA	22.95
FORMICIDA	37.80
MANUTENCAO	63.75
	124.50
	2149.18

Capítulo 18 :

PROC REG

O procedimento REG ajusta modelos de regressão linear usando o método dos mínimos quadrados. O procedimento é descrito com detalhes no manual “SAS / STAT Software”.

Algumas declarações deste procedimento não estão disponíveis no SAS University Edition, como a declaração PLOT, mas o SAS University Edition gera gráficos ODS de resíduos automaticamente e pode gerar outros gráficos ODS habilitando-se os mesmos antes do procedimento e usando-se, em substituição à PLOT, a declaração “PLOTS=opções;”.

18.1 SINTAXE

```
PROC REG < OPÇÕES > ;
  < RÓTULO: > MODEL VARIÁVEISDEPENDENTES= <REGRESSORES>
    < / OPÇÕES > ;
  BY VARIÁVEIS ;
  FREQ VARIÁVEL ;
  ID VARIÁVEIS ;
  VAR VARIÁVEIS ;
  WEIGHT VARIÁVEL ;
  ADD VARIÁVEIS ;
  DELETE VARIÁVEIS ;
  < RÓTULO: > MTEST
    <EQUAÇÃON, : : : ,EQUAÇÃON>
    < / OPÇÕES > ;
  OUTPUT < OUT=ARQUIVODESAÍDA > PALAVRASCHAVES=NOMES
    < : : : PALAVRASCHAVES=NOMES > ;
  PAINT <CONDIÇÃO J ALLOBS>
    < / OPÇÕES > J < STATUS | UNDO > ;
  PLOT <VARIÁVELY*VARIÁVELX> <=SÍMBOLO>
    < : : : VARIÁVELY*VARIÁVELX > <=SÍMBOLO>
    < / OPÇÕES > ;
  PRINT < OPÇÕES > < ANOVA > < MODELDATA > ;
  REFIT;
  RESTRICT EQUAÇÃON, : : : ,EQUAÇÃON ;
  REWEIGHT <CONDIÇÃO J ALLOBS>
    < / OPÇÕES > J < STATUS | UNDO > ;
  < RÓTULO: > TEST EQUAÇÃON,< : : : ,EQUAÇÃON>
    < / OPÇÃO > ;
```

18.1.1 OPÇÕES DO PROCEDIMENTO

DATA=ArquivoDeDados - Nome do arquivo de dados a usar com o procedimento.

OUTEST=ArquivoDeSaída - Cria um arquivo de saída que contém os parâmetros estimados e outras estatísticas.

OUTSSCP=ArquivoDeSaída - Cria um arquivo de saída que contém as somas de quadrados e produtos cruzados.

COVOUT – Imprime a matriz de covariância para os parâmetros estimados. Só funciona em conjunto com OUTEST.

CORR – Imprime a matrix de correlações das variáveis listadas nas instruções MODEL e VAR.

SIMPLE – Imprime estatísticas simples para as variáveis listadas nas instruções MODEL e VAR.

COLLIN – Imprime a análise de colinearidade.

USSCP – Imprime a matriz de somas de quadrados e produtos cruzados não-corrigidos.

ALL – Imprime todas as estatísticas (CORR, SIMPLE e USSCP).

NOPRINT – Impede a impressão de relatório automático.

ALPHA=Valor – Atribui um valor de significância para os testes do procedimento (Padrão=0.05).

SINGULAR=Critério – Atribui um critério para verificação de singularidade.

Observação: há outras opções para o procedimento.

18.1.2 INSTRUÇÕES DO PROCEDIMENTO

ADD Variáveis; – Adiciona variáveis independentes ao modelo de regressão.

BY Variáveis; – Separa a análise por grupos conforme as variáveis listadas.

DELETE – Elimina variáveis independentes ao modelo de regressão.

FREQ Variável; - Especifica uma variável com a frequência das observações.

MODEL Dependente=Independentes/ Opções; Especifica as variáveis dependentes, independentes e opções do modelo de regressão.

MTEST – Realiza testes multivariados.

OUTPUT < OUT=ArquivoDeSaída P=<NomeDasEstimativas> R=<NomeDosResíduos> Estatísticas=Nomes > ; - Cria um arquivo de saída com as estatísticas relacionadas; o nome para as estimativas e para os resíduos pode ser qualquer nome de variável válido, escolhido pelo usuário, exceto palavras restritas do sistema.

PLOT – Imprime um gráfico nos moldes do PROC PLOT com cruzamentos Y*X; Exemplo: PLOT R.*P.; - Gera um gráfico dos resíduos (ordenadas), distribuídos ao longo do eixo das abscissas que tem os valores estimados para a variável dependente.

TEST – Realiza um teste F sobre as funções lineares dos parâmetros.

VAR Variáveis; - Lista de variáveis que devem ser computadas para os produtos cruzados, variáveis que podem ser adicionadas interativamente ao modelo ou variáveis a serem utilizadas na elaboração de gráficos e que não estão no modelo.

WEIGHT Variável; - Declara uma variável com o peso para as observações.

Observação: há outras instruções disponíveis para o procedimento.

18.1.2.1 OPÇÕES DA INSTRUÇÃO MODEL

■ <RÓTULO>: MODEL DEPENDENTES= <REGRESSORES> < / OPÇÕES > ;
ACOV – Imprime a matriz de covariância assintótica estimada sob a hipótese de heterocedasticidade.

- COLLIN – Imprime a análise de colinearidade.
- COVB – Imprime a matriz de covariâncias estimada para as estimativas: $s^2(X'X)^{-1}$.
- DW – Calcula a estatística DW de Durbin-Watson para teste de autocorrelação de primeira ordem dos erros.
- I – Imprime a matriz $(X'X)^{-1}$.
- NOINT – Ajusta um modelo sem intercepto.
- SPEC – Realiza um teste de verificação se o primeiro e segundo momentos do modelo são corretamente especificados.
- STB – Imprime os parâmetros estimados padronizados.
- OUTPUT DATA=regout P=yhat E=error – Salva os valores estimados (yhat) e resíduos (error) do modelo no arquivo regout.
- SELECTION=MétodoDeSeleção – Especifica um método para seleção das variáveis independentes mais significativas (Métodos possíveis: FORWARD (ou F), BACKWARD (ou B), STEPWISE, MAXR, MINR, RSQUARE, ADJRSQ, CP ou NONE (usa o modelo completo)).
- BEST=Número – Especifica um número máximo de subconjunto de modelos a apresentar nos resultados ou para saída no arquivo definido por OUTEST=ArquivoDeSaída.
- MAXSTEP=Número – Especifica um número máximo de passos que devem ser realizados.
- NOINT – Ajusta o modelo sem intercepto.
- OUTEST=ArquivoDeSaída - Cria um arquivo de saída que contém os parâmetros estimados e outras estatísticas para o modelo.
- Observação: há outras opções para a instrução MODEL.

Exemplo 14.1 – Ajuste de um modelo para estimar valores perdidos e substituir no arquivo de dados iniciais.

Neste exemplo, há 12 observações com valores de Y e X, sendo que dois valores de Y estão faltando, são valores perdidos. O programa do “Exemplo – Estimando dados perdidos” foi escrito para ajustar uma equação que estima os valores perdidos de Y e os inclui no arquivo.

```

* EXEMPLO - Estimando dados perdidos;
OPTIONS LS=80 PS=54 NODATE NOSTIMER; TITLE;
DATA ARQREG ;
INPUT Y X @@;
  X1=X;
  X2=X**2;
  VAR_AUX=0;
CARDS;
12 18 13 20 14 22
. 22 16 23 17 24
18 25 19 26 . 26
21 26 22 27 23 27
;
PROC SORT DATA=ARQREG; BY VAR_AUX;
PROC PRINT DATA=ARQREG;
  TITLE 'DADOS INICIAIS COM 2 VALORES DE Y PERDIDOS';
*****;
PROC REG DATA=ARQREG OUTEST=PARAMETROS;
  TITLE 'AJUSTE DA EQUACAO Y=B0 + B1*X + B2*X**2';
  MODEL Y= X1 X2;
  PLOT R.*P.;
DATA PARAMETROS (KEEP=B0 B1 B2 VAR_AUX) NOOBS;

```

```

SET PARAMETROS;
B0=INTERCEPT;
B1=X1;
B2=X2;
VAR_AUX=0;
PROC PRINT DATA=PARAMETROS NOOBS;
TITLE 'PARAMETROS ESTIMADOS';
*****;
PROC SORT DATA=PARAMETROS; BY VAR_AUX;
DATA ARQREG (KEEP=Y X);
MERGE ARQREG PARAMETROS; BY VAR_AUX;
IF Y LE 0 THEN Y= B0 + B1*X + B2*X**2;
PROC PRINT DATA=ARQREG;
TITLE 'DADOS DE SAIDA COM 2 VALORES DE Y
ESTIMADOS';
*****;
RUN;
QUIT;

```

23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39

O relatório impresso pelo programa é o seguinte:

DADOS INICIAIS COM 2 VALORES DE Y PERDIDOS

Obs	Y	X	X1	X2	VAR_AUX
1	12	18	18	324	0
2	13	20	20	400	0
3	14	22	22	484	0
4	.	22	22	484	0
5	16	23	23	529	0
6	17	24	24	576	0
7	18	25	25	625	0
8	19	26	26	676	0
9	.	26	26	676	0
10	21	26	26	676	0
11	22	27	27	729	0
12	23	27	27	729	0

AJUSTE DA EQUACAO $Y=B_0 + B_1*X + B_2*X**2$

The REG Procedure

Model: MODEL1

Dependent Variable: Y

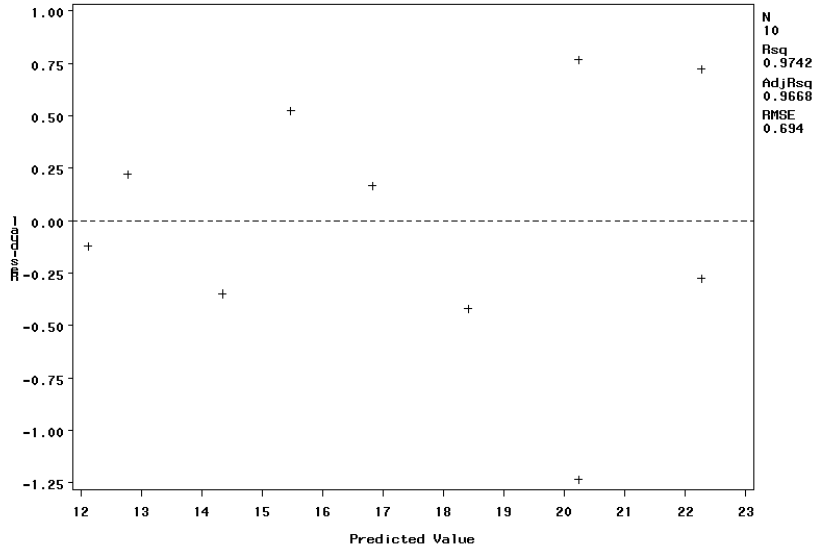
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	127.12809	63.56404	131.96	<.0001
Error	7	3.37191	0.48170		
Corrected Total	9	130.50000			

Root MSE	0.69405	R-Square	0.9742
Dependent Mean	17.50000	Adj R-Sq	0.9668
Coeff Var	3.96598		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	47.39804	14.89286	3.18	0.0154
X1	1	-4.01847	1.32277	-3.04	0.0189
X2	1	0.11437	0.02896	3.95	0.0055

The REG Procedure
 AJUSTE DA EQUACAO $Y = B_0 + B_1 \cdot X + B_2 \cdot X^2$

$Y = 47.398 - 4.0185 X_1 + 0.1144 X_2$



PARAMETROS ESTIMADOS

B0	B1	B2	VAR_AUX
47.3980	-4.01847	0.11437	0

DADOS DE SAIDA COM 2 VALORES DE Y ESTIMADOS

Obs	Y	X
1	12.0000	18
2	13.0000	20
3	14.0000	22
4	14.3482	22
5	16.0000	23
6	17.0000	24
7	18.0000	25
8	19.0000	26
9	20.2339	26
10	21.0000	26
11	22.0000	27
12	23.0000	27

Capítulo 19 : PROC NLIN

O procedimento NLIN ajusta modelos não-lineares através dos métodos: Gauss-Newton, Marquardt, Newton, Gradient e DUD (Doesn't Use Derivatives). O padrão, quando se fornece as derivadas parciais dos parâmetros é o método de Gauss-Newton; quando não se fornece as derivadas, o padrão é o método DUD, que se baseia no método da secante multivariada. O procedimento NLIN pode gerar gráficos ODS, desde que seja usado o comando "ODS Graphics ON;" antes do procedimento NLIN.

O uso do PROC NLIN é bastante semelhante ao procedimento REG quanto a especificação do modelo.

Entretanto, no PROC NLIN é possível fornecer as derivadas parciais dos parâmetros do modelo para facilitar o cálculo; isso é obrigatório para todos os métodos, exceto com o DUD.

Pode-se fornecer os valores iniciais (sementes) para os parâmetros e até mesmo uma amplitude para cada um deles. O procedimento permite, ainda, determinar limites para os parâmetros, reduzindo as iterações possíveis e a possibilidade de erro na determinação da melhor iteração quando se conhece as amplitudes possíveis. Isso é útil com funções como a equação de crescimento de Chapman-Richards, em que se sabe antecipadamente que valores os parâmetros podem assumir, conforme o "Exemplo - Ajuste da equação de Chapman-Richards".

Da mesma forma que no procedimento REG, é possível criar um arquivo de saída com os resultados e com os parâmetros estimados, para uso posterior com outros procedimentos do programa.

O procedimento NLIN não calcula todas as estatísticas possíveis para modelos não-lineares, como o R^2 , e, em alguns casos, é necessário utilizar o PROC MODEL para obter estatísticas complementares. Mas, o PROC MODEL não gera arquivos de saída, então, é possível dizer que são procedimentos que se complementam.

Para maior poder do procedimento, praticamente todas as instruções de atribuição de valor do DATA STEP foram incrementadas no PROC NLIN. Assim, após declarar o modelo e os parâmetros, é possível inserir instruções de cálculo intermediário.

Uma obra extremamente útil que orienta o ajuste das funções não-lineares mais conhecidas com o SAS é o seguinte: "SIT, V. *Catalog of curves for curve fitting*."

Victoria, Canada, B.C.: Ministry of Forests, Forest Science Research Branch, Biometrics information handbook series, n.4, 1994.", disponível em: <www.for.gov.bc.ca/hfd/pubs/docs/bio/Bio04.pdf>.

Mais informações sobre o uso do PROC NLIN podem ser encontrados no manual: "SAS-STAT User's Guide."

19.1 SINTAXE

```
PROC NLIN < OPÇÕES > ;
MODEL DEPENDENTE=EXPRESSÕES ;
PARAMETERS PARÂMETRO=VALORES
    <,..., PARÂMETRO=VALORES>;
* OUTRAS INSTRUÇÕES DO PROCEDIMENTO;
BOUNDS DESIGUALDADE
    <,..., DESIGUALDADE > ;
BY VARIÁVEIS ;
DER.PARÂMETRO=EXPRESSÕES ;
DER.PARÂMETRO.PARÂMETRO=EXPRESSÕES ;
ID VARIÁVEIS ;
OUTPUT OUT=ARQUIVODESAÍDA PALAVRACHAVE=NOME
    <,..., PALAVRACHAVE=NOME> ;
CONTROL VARIÁVEL < =VALORES >
    <... VARIÁVEL < =VALORES > > ;
```

19.1.1 OPÇÕES DO PROCEDIMENTO

BEST=n – Apresetar somente as n melhores iterações de acordo com a soma de quadrados do resíduo.

METHOD=NomeDoMétodo – Especifica o método a ser utilizado (GAUSS, ou MARQUARDT, ou NEWTON, ou GRADIENT, ou DUD).

MAXITER=i – Limita o número de iterações.

NOITPRINT – Suprime a impressão das iterações realizadas, mostrando somente os resultados para a melhor.

NOPRINT – Suprime a impressão; é útil quando se quer somente gerar um arquivo de resultados para uso posterior.

OUTEST=ArquivoDeSaída – Especifica um arquivo onde são gravados os parâmetros estimados para cada iteração; quando se deseja somente a melhor, deve-se usar juntamente com a opção BEST=1.

PRINT – Imprime todas as estatísticas do procedimento (são muitas páginas de informação; use somente se necessário).

Observação: Há muitas outras opções para este procedimento.

19.1.2 INSTRUÇÕES DO PROCEDIMENTO

BOUNDS Limites; – Limita a amplitude para cálculo dos parâmetros; exemplos: bounds a1-a10<=20; bounds c>30; bounds a b c > 0; bounds 0<=B<=10; bounds 15<x1<=30; bounds r <= s <= p < q;

BY Variáveis; – Especifica variáveis para formar subgrupos na análise.

DER.Parâmetro<.Parâmetro>; - Especifica derivadas parciais dos parâmetros.

ID Variáveis; - Especifica variáveis a incluir no arquivo de saída.

MODEL Dependentes=Expressão; - Define a equação do modelo de regressão.

OUTPUT OUT=ArquivoDeSaída Estatísticas=NomesDeSaída; - Cria um arquivo de saída com as estatísticas especificadas em que os nomes das colunas (NomesDeSaída) são atribuídos pelo usuário.

PARMS NomesDosParâmetros=Amplitudes; - Informa os nomes dos parâmetros utilizados no modelo.

Observação: Há outras instruções possíveis com o procedimento, por exemplo: instruções ARRAY , DO e instruções de controle de fluxo.

A função de crescimento de Chapman-Richards é aplicável quando a curva de crescimento da produção tem forma sigmoideal, apresentando um ponto de inflexão onde a curva muda de direção e é dada pela equação:

$$Y = A [1 - e^{-k \cdot t}]^r$$

Em que: Y = variável dependente; A = assíntota superior (ou máximo valor de Y); k = velocidade de crescimento que pode variar de 0 até 1 (geralmente próximo de 0,2 para altura de árvores); r = ponto de inflexão entre 0 e a assíntota, geralmente < A/2.

No “Exemplo - Ajuste da equação de Chapman-Richards” a seguir, são usados os dados de 18 árvores de uma mesma espécie com idades entre 3 e 30 anos. Os dados são delineados em um gráfico com as alturas na ordenada e as idades na abscissa, para se determinar aproximadamente onde se encontram a assíntota superior e o ponto de inflexão. Com isso é possível estimar os valores iniciais para os parâmetros e os seus limites, que são escritos no programa do exemplo 15.1. Portanto, é necessário executar o programa duas vezes. Na primeira vez, não se sabe onde se encontra a assíntota nem o ponto de inflexão e é atribuído um valor válido para árvores em situação normal. Com isso, nem sempre o programa consegue uma iteração onde são satisfeitos os critérios de convergência. Mas, com os dados plotados no primeiro gráfico, é verificado em que nível se encontra a assíntota e o ponto de inflexão para que se possa atribuir estes valores para as sementes (faixas de valores iniciais) dos parâmetros.

O programa do “Exemplo - Ajuste da equação de Chapman-Richards” cria um arquivo de saída com os resultados e imprime o gráfico dos resíduos em relação às estimativas e outro dos valores observados e das estimativas em relação à idade.

```
* EXEMPLO - Ajuste da equação de Chapman-Richards;
OPTIONS LS=80 PS=54 NODATE NOSTIMER; TITLE;
data A;
input idade h @@;
Y=h;
X=idade;
cards;
05 04 05 04 12 18 14 22 07 07 07 06 25 28 25 26 03 01
10 12 08 10 15 23 16 25 20 27 22 27 30 30 27 29 03 02
;
```

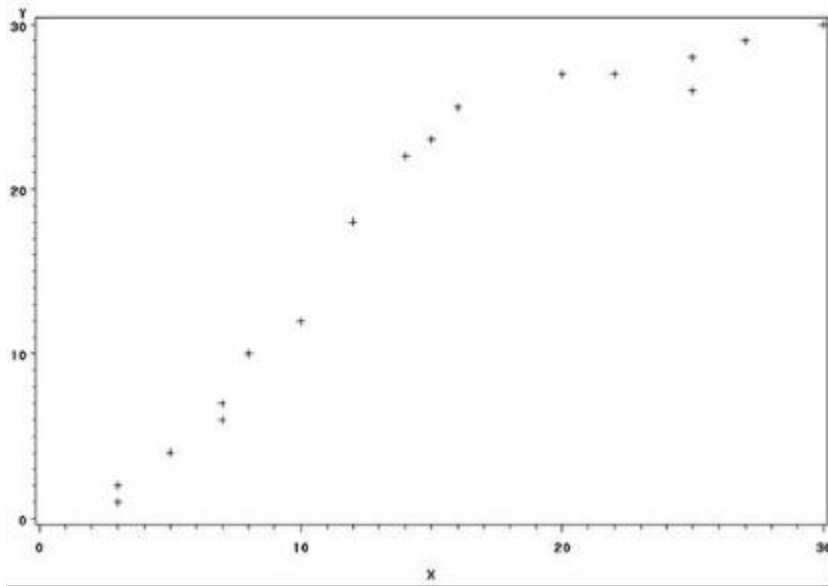
1
2
3
4
5
6
7
8
9
10

```

PROC SORT DATA=A; BY X Y;          11
PROC GPLOT;                          12
  PLOT Y*X;                          13
PROC NLIN DATA=A BEST=1;          14
  PARAMETERS A=20 TO 30 BY 5.0 B=0.2 C=0.1 TO 9.9 BY 1.0; 15
  BOUNDS 19<A<31, 0<B<1, 0<C<10; 16
  EBX = EXP(-B*X);                  17
  EBX1 = 1 - EBX;                   18
  EBXC = (EBX1)**C;                 19
  MODEL Y = A*EBXC;                 20
  DER.A = EBXC;                     21
  DER.B = A*X*C*EBX*EBX1**(C-1);   22
  DER.C = A*EBXC*LOG(EBX1);        23
  OUTPUT OUT=RESULTADO              24
    P=ESTIMATIVAS R=RESIDUOS PARMS=A B C; 25
proc print data=resultado;          26
PROC GPLOT DATA=RESULTADO;        27
  SYMBOL1 VALUE=STAR CV=BLUE;      28
  PLOT RESIDUOS*ESTIMATIVAS / VREF=0; 29
  SYMBOL1 VALUE=STAR CV=BLUE;      30
  SYMBOL2 VALUE=TRIANGLE CV=RED INTERPOL=JOIN; 31
  PLOT H*IDADE ESTIMATIVAS*IDADE / OVERLAY; 32
  *****;                          33
RUN;                                34
QUIT;                                35
* Observação: Deve-se atribuir valores tais que as sementes não
  ultrapassem os limites;

```

O resultado do processamento, com os parâmetros tendo seus valores de partida e limites atribuídos com base no exame do primeiro gráfico, é o relatório a seguir, onde se observa no gráfico de Y*X que a assíntota deve ficar entre 20 m e 30 m e o ponto de inflexão está abaixo de 10 m. Estes valores são substituídos no programa quanto às sementes dos parâmetros e seus limites (bounds).



The NLIN Procedure
Dependent Variable Y

Grid Search			
A	B	C	Sum of Squares
30.0000	0.2000	5.1000	23.9411

The NLIN Procedure
Dependent Variable Y
Method: Gauss-Newton

Iterative Phase				
Iter	A	B	C	Sum of Squares
0	30.0000	0.2000	5.1000	23.9411
1	29.3852	0.1929	4.6831	19.4509
2	29.4098	0.1916	4.6359	19.4430
3	29.4097	0.1916	4.6335	19.4429
4	29.4099	0.1916	4.6331	19.4429

Iterative Phase				
Iter	A	B	C	Sum of Squares
5	29.4099	0.1916	4.6331	19.4429

NOTE: Convergence criterion met.

Estimation Summary	
Method	Gauss-Newton
Iterations	5
R	2.327E-6
PPC(C)	1.425E-6
RPC(C)	0.000011
Object	3.65E-10
Objective	19.44293
Observations Read	18
Observations Used	18
Observations Missing	0

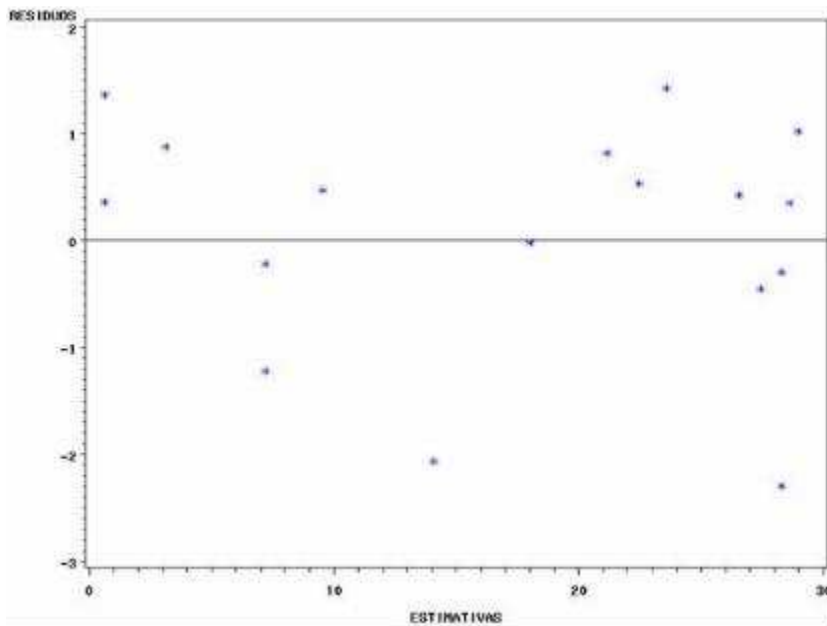
An intercept was not specified for this model.

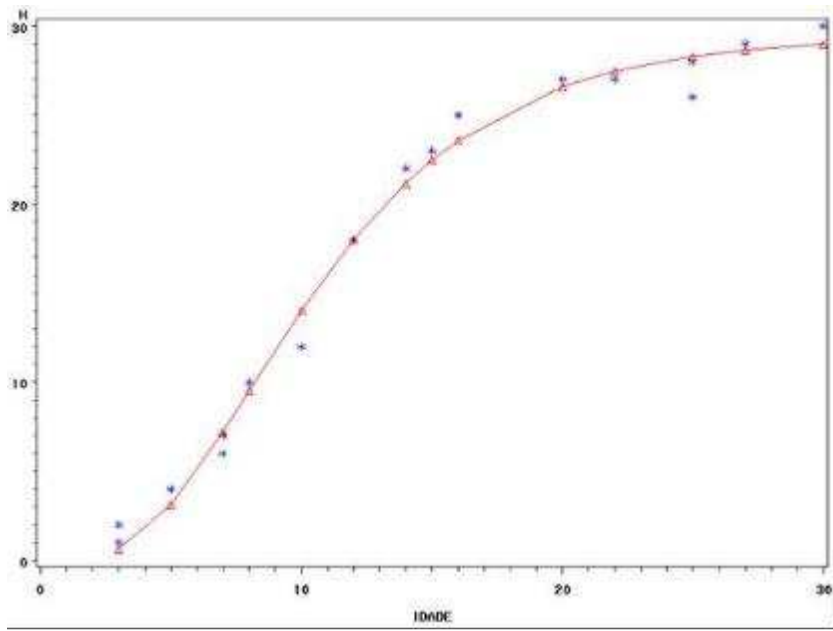
Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Regression	3	6967.6	2322.5	1791.80	<.0001
Residual	15	19.4429	1.2962		
Uncorrected Total	18	6987.0			
Corrected Total	17	1953.6			

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
A	29.4099	0.7348	27.8436	30.9761

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
B	0.1916	0.0194	0.1503	0.2329
C	4.6331	0.8048	2.9177	6.3485

Approximate Correlation Matrix			
	A	B	C
A	1.0000000	-0.7615412	-0.6300812
B	-0.7615412	1.0000000	0.9583696
C	-0.6300812	0.9583696	1.0000000





Capítulo 20 : PROC UNIVARIATE

O procedimento UNIVARIATE calcula estatísticas univariadas para grupos de dados.

20.1 SINTAXE

```
PROC UNIVARIATE <OPÇÕES>;
BY <DESCENDING> VARIÁVEL-1 <...<DESCENDING> VARIÁVEL-
N><NOTSORTED>;
CLASS VARIÁVEL-1<(VARIÁVEL-OPÇÕES)>
    <VARIÁVEL-2<(VARIÁVEL-OPÇÕES)>>
    </ KEYLEVEL='VALOR1'|('VALOR1' 'VALOR2')>;
FREQ VARIÁVEL;
HISTOGRAM <VARIÁVEIS> </ OPÇÕES>;
ID VARIÁVEIS;
INSET <PALAVRASCHAVE> </ OPÇÕES>;
OUTPUT <OUT=ARQUIVODESAÍDA>
    ESTATÍSTICAS=NOMES
    <ESPECIFICAÇÃODEPERCENTÍS>;
PROBPLOT <VARIÁVEIS> </ OPÇÕES>;
QQPLOT <VARIÁVEIS> </ OPÇÕES>;
VAR VARIÁVEIS;
WEIGHT VARIÁVEL;
```

20.1.1 OPÇÕES DO PROCEDIMENTO

ALL – Faz com que o procedimento imprima, quando adequado, as seguintes estatísticas e tabelas: FREQ, MODES, NEXTRVAL=5, PLOT e CIBASIC, CIPCTLD, CIPCTLNORMAL, LOCCOUNT, NORMAL, ROBUSTCALE, TRIMMED=.25 e WINSORIZED=.25.

ALPHA=value – Informa o nível de significância para os testes (Padrão=0,05).

CIBASIC(<TYPE=keyword> <ALPHA=value>)> - Requisita limites de confiança para média e variação assumindo que os dados tem distribuição normal.

FREQ – Requisita uma tabela de frequência dos dados.

MODES – Requisita todos os tipos de tabelas de frequência possíveis.

MU0=Valor(es) – Informa um valor para a média para os testes de hipóteses. Se não for fornecido, o padrão será zero.

NOPRINT – Impede a impressão de relatório automático.

NORMAL – Requisita que seja realizado o teste de normalidade dos dados. Não é realizado quando é utilizada a instrução WEIGHT.

PLOTS – Produz um gráfico de barras horizontais.

Observação: Há várias outras opções possíveis para este procedimento.

20.1.2 INSTRUÇÕES DO PROCEDIMENTO

BY Variáveis; - Separa as estatísticas por grupos de variáveis.

CLASS Variáveis-opções; - Classifica as estatísticas por grupos de variáveis. Algumas das opções são: MISSING; Determina que os valores perdidos devem entrar nas estatísticas. Se não for usado, os valores perdidos serão excluídos das análises; FREQ – Ordena os valores de acordo com a sua frequência.

FREQ Variável; - Especifica uma variável que contém as frequências das observações.

HISTOGRAM – Cria um histograma com os resultados.

ID Variáveis; – Especifica variáveis para incluir na tabela de valores extremos.

OUTPUT <OUT=ArquivoDeSaída> Palavras-chave=Nomes <EspecificaçõesDePercentís> ; - Cria um arquivo de saída. Palavras-chave das estatísticas disponíveis: CSS,CV, KURTOSIS, MAX, MEAN, N, MIN, MODE, RANGE, NMISS, NOBS, STDMEAN, SKEWNESS, STD, USS, SUM, SUMWGT, VAR, MEDIAN, P1, P5, P10, P90, P95, P99, Q1, Q3, QRANGE, GINI, MAD, QN, SN, STD_GINI, STD_MAD, STD_QN, STD_QRANGE, STD_SN, NORMAL, PROBN, MSIGN, PROBM, SIGNRANK, PROBS, T, PROBT.

PROBPLOT – Cria um gráfico de probabilidades.

VAR Variáveis; – Informa quais variáveis devem ser incluídas nas estatísticas.

WEIGHT variable; – Indica uma variável que contém o peso das observações.

Observações: Há outras instruções disponíveis com este procedimento.

Exemplo 18.1 - Normalidade de variáveis dendrométricas

```

* EXEMPLO 18.1 - Normalidade de variáveis dendrométricas;
  Title;
DATA arvores;
INPUT Arvore h hc dcc casca @@;
DATALINES;
1 21.05 16 26.1 2.9 13 16.8 12 22 3.5 25 19.3 13 16.6 2.3 37 20.1 14
  20.4 1.6
2 18.8 12 16.6 1.6 14 20.5 15 23.6 3.2 26 21 14 22.6 3.2 38 21.5 15 25.1
  3.1
3 21 15 23.6 2.9 15 16.7 11 16.9 2.9 27 20.9 15 23.2 3.5 39 17.4 12 17.2
  2.2
4 19.8 13 17.8 2.2 16 21.5 15 22.6 3.2 28 20.7 15 22.9 2.8 40 18 12 18.1
  2.5
5 19.3 13 21 1.9 17 20.3 15 24.8 3.5 29 17.2 11 15.6 1.9 41 21 15 23.6
  3.9
6 20.8 15 25.8 3.2 18 19.3 13 17.8 2.5 30 18.1 11 13.1 1.3 42 20.7 14 21
  2.2
7 17 11 13.6 0.4 19 19.6 13 15.9 2.2 31 20.5 14 21.3 3.2 43 15.6 10 15.6
  3.8
8 21.8 15 23.6 3.2 20 19.7 13 21 3.5 32 20.1 14 25.1 4.1 44 20.3 13 17.5
  1.6
9 22.1 15 20.4 3.2 21 19.5 14 23.9 3.5 33 18 12 16.2 2.2 45 21 14 21.3
  2.5
10 21.6 15 23.2 3.5 22 21.5 14 21.3 2.8 34 17.3 11 15.3 2.6 46 17.7 11
  15 1.9
11 17.3 11 14.3 1.9 23 19.9 14 22.9 2.5 35 18.6 13 18.8 2.6 47 19.1 13
  18.5 2.6
12 17.2 12 18.5 1.9 24 17.7 11 14.6 1.2 36 21.1 15 23.2 2.5 48 20 15
  24.8 2.5
;
PROC UNIVARIATE DATA=ARVORES NORMAL;
  VAR H DCC CASCA;
  HISTOGRAM;
RUN;

```

■ QUIT;

Resultados:

The UNIVARIATE Procedure
Variable: H

Moments			
N	48	Sum Weights	48
Mean	19.4989583	Sum Observations	935.95
Std Deviation	1.65852054	Variance	2.75069038
Skewness	-0.4719703	Kurtosis	-0.8674416
Uncorrected SS	18379.3325	Corrected SS	129.282448
Coeff Variation	8.5056879	Std Error Mean	0.23938682

Basic Statistical Measures			
Location		Variability	
Mean	19.49896	Std Deviation	1.65852
Median	19.85000	Variance	2.75069
Mode	21.00000	Range	6.50000
		Interquartile Range	2.95000

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	81.45377	Pr > t	<.0001
Sign	M	24	Pr >= M	<.0001
Signed Rank	S	588	Pr >= S	<.0001

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.941115	Pr < W	0.0179

Tests for Normality				
Test	Statistic		p Value	
Kolmogorov-Smirnov	D	0.102114	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.173283	Pr > W-Sq	0.0115
Anderson-Darling	A-Sq	1.044287	Pr > A-Sq	0.0089

Quantiles (Definition 5)	
Quantile	Estimate
100% Max	22.10
99%	22.10
95%	21.60
90%	21.50
75% Q3	20.95
50% Median	19.85
25% Q1	18.00
10%	17.20
5%	16.80
1%	15.60
0% Min	15.60

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
15.6	28	21.5	14
16.7	10	21.5	38
16.8	2	21.6	37
17.0	25	21.8	29
17.2	45	22.1	33

The UNIVARIATE Procedure
Variable: H

The UNIVARIATE Procedure
Variable: DCC

Moments			
N	48	Sum Weights	48
Mean	20.0791667	Sum Observations	963.8
Std Deviation	3.69726514	Variance	13.6697695
Skewness	-0.1919842	Kurtosis	-1.2272458
Uncorrected SS	19994.78	Corrected SS	642.479167
Coeff Variation	18.4134392	Std Error Mean	0.53365426

Basic Statistical Measures			
Location		Variability	
Mean	20.07917	Std Deviation	3.69727
Median	21.00000	Variance	13.66977
Mode	23.60000	Range	13.00000
		Interquartile Range	6.45000

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	37.6258	Pr > t	<.0001
Sign	M	24	Pr >= M	<.0001
Signed Rank	S	588	Pr >= S	<.0001

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.945318	Pr < W	0.0261
Kolmogorov-Smirnov	D	0.127321	Pr > D	0.0494
Cramer-von Mises	W-Sq	0.165404	Pr > W-Sq	0.0154
Anderson-Darling	A-Sq	0.933504	Pr > A-Sq	0.0180

Quantiles (Definition 5)	
Quantile	Estimate
100% Max	26.10
99%	26.10
95%	25.10
90%	24.80
75% Q3	23.20
50% Median	21.00
25% Q1	16.75
10%	15.00
5%	14.30
1%	13.10
0% Min	13.10

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
13.1	23	24.8	48
13.6	25	25.1	8
14.3	41	25.1	31
14.6	46	25.8	21
15.0	40	26.1	1

The UNIVARIATE Procedure
Variable: DCC

The UNIVARIATE Procedure
Variable: CASCA

Moments			
N	48	Sum Weights	48
Mean	2.62291667	Sum Observations	125.9
Std Deviation	0.77493708	Variance	0.60052748
Skewness	-0.438433	Kurtosis	0.1596969
Uncorrected SS	358.45	Corrected SS	28.2247917
Coeff Variation	29.544861	Std Error Mean	0.11185253

Basic Statistical Measures			
Location		Variability	
Mean	2.622917	Std Deviation	0.77494
Median	2.600000	Variance	0.60053
Mode	3.200000	Range	3.70000
		Interquartile Range	1.00000

Tests for Location: Mu0=0				
Test	Statistic		p Value	
Student's t	t	23.44977	Pr > t	<.0001
Sign	M	24	Pr >= M	<.0001
Signed Rank	S	588	Pr >= S	<.0001

Tests for Normality				
Test	Statistic		p Value	
Shapiro-Wilk	W	0.974645	Pr < W	0.3797
Kolmogorov-Smirnov	D	0.105102	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.062539	Pr > W-Sq	>0.2500
Anderson-Darling	A-Sq	0.403456	Pr > A-Sq	>0.2500

Quantiles (Definition 5)	
Quantile	Estimate
100% Max	4.1
99%	4.1
95%	3.8
90%	3.5
75% Q3	3.2
50% Median	2.6
25% Q1	2.2
10%	1.6
5%	1.3
1%	0.4
0% Min	0.4

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs
0.4	25	3.5	34
1.2	46	3.5	37
1.3	23	3.8	28
1.6	32	3.9	20
1.6	5	4.1	31

Capítulo 21 : PROC IML (operações com matrizes)

O procedimento PROC IML (Interactive Matrix Language) realiza operações com matrizes. Originalmente foi desenvolvido para ser interativo, mas passou a ser programável e permite todo tipo de operações com matrizes, tendo muitas funções internas que abrangem praticamente toda a álgebra matricial e muitas funções estatísticas. Este capítulo deve ser entendido como uma simples introdução ao poder do PROC IML que, na verdade, é uma linguagem separada e de características próprias, mas com interfaces com a linguagem principal do SAS e de sintaxe semelhante. Mais informações podem ser obtidas no manual "SAS Institute. SAS/IML User's Guide, v. 8. Cary, NC, USA: SAS Institute Inc., 1999. 846 p."

21.1 SINTAXE

```
PROC IML;
  DECLARAÇÕES;
  ....COMANDOS;
  FUNÇÕESIML;
  MÓDULOS;
  SUBROTINAS;
RUN;
QUIT;
```

21.1.1 DECLARAÇÕES, MÓDULOS E SUBROTINAS

São quaisquer declarações válidas SAS, como atribuição de valores à vetores e matrizes, comandos, laços de DO END e condições de IF, etc, além das declarações próprias da linguagem IML.

Com algumas exceções, são aceitas no dentro da PROC IML todas as declarações desenvolvidas para o DATA STEP.

A Atribuição de valores à vetores e matrizes é realizada através de uma expressão de atribuição de valor como:

```
NomeDaMatriz={a11 a12 ... a1m,
              a21 a22 ... a2m,
              ... .. ...,
              an1 an2 ... anm};
```

Em que: NomeDaMatriz=Nome da matriz podendo ser qualquer nome válido na linguagem SAS; a

Subrotinas são subprogramas escritos na linguagem IML (não são descritas aqui).

21.1.2 COMANDOS E OPERADORES

Os principais comandos da linguagem IML são os seguintes:

COMANDO	AÇÃO
FREE	libera uma matriz de seus valores aumentando o espaço disponível
LOAD	carrega uma matriz de uma biblioteca de armazenamento
MATTRIB	associa atributos de impressão com uma matriz
PRINT	imprime uma matriz ou uma mensagem
RESET	acessa várias opções de sistema
REMOVE	remove uma matriz ou módulo de uma biblioteca
SHOW	requisita que as informações de sistema sejam mostradas
STORE	armazena uma matriz em uma biblioteca

Os operadores da linguagem IML são relacionados a seguir:

OPERADOR	AÇÃO
+	adição
-	subtração
*	multiplicação
/	divisão
#	multiplicação de elementos
##	potenciação de elementos
<>	maior elemento
><	menor elemento
	OU lógico (OR)
&	E lógico (AND)
<	menor que
<=	menor ou igual a
>	maior que
>=	maior ou igual a
^=	não igual a
=	igual a

No “Exemplo - PROC IML - Interactive Matrix Language”, a seguir, demonstra-se o uso do procedimento.

```

* Exemplo - PROC IML - Interactive Matrix Language;          1
proc iml;                                                    2
reset print; * LIMPA A Memória de impressão;                3
Print "Exemplo de modulo";                                   4
start add(x,y);                                              5
  sum=x+y;                                                   6
  return(sum);                                               7
finish;                                                      8
a={9 2,5 7};                                                9
b={1 6,8 10};                                               10
c=add(a,b);                                                  11
d=add(add(6,3),add(5,5));                                    12
print d;                                                      13

```

```

PROC IML;
print "Exemplos de operacoes com matrizes";
E={9 2 5 7,
  7 5 6 3,
  5 3 4 4,
  8 3 7 6};
F={9 2 3 2,
  7 5 3 3,
  5 3 2 2,
  8 3 1 1};
Q={9 2 3 2,
  7 5 3 3};
Print 'Primeira Matriz' E;
Print 'Segunda Matriz' F;
Print 'G = Inverso de E';
G=INV(E);
Print G;
H=F#G;* Multiplica cada elemento de F por
      cada de elemento de G;
Print 'H = Multiplicacao de F por G', H;
L=Q*G; *Multiplica a matriz Q pela matriz G;
Print 'L = Multiplicacao de Q por G', L;
Media=H[:];
Print 'Media dos elementos de H', Media;
Transposta=H`;
Print 'Transposta de H', Transposta;
run;
quit;

```

Resultados:

EXEMPLO DE MODULO

A 2 rows 2 cols (numeric)

9	2
5	7

B 2 rows 2 cols (numeric)

1	6
8	10

SUM 2 rows 2 cols (numeric)

10	8
13	17

C 2 rows 2 cols (numeric)

10	8
13	17

SUM 1 row 1 col (numeric)

9

SUM 1 row 1 col (numeric)

10

SUM 1 row 1 col (numeric)

19

D 1 row 1 col (numeric)

19

D
19

EXEMPLOS DE OPERACOES COM MATRIZES

	E			
PRIMEIRA MATRIZ	9	2	5	7
	7	5	6	3
	5	3	4	4
	8	3	7	6

	F			
SEGUNDA MATRIZ	9	2	3	2
	7	5	3	3
	5	3	2	2
	8	3	1	1

G = INVERSO DE E

G			
0.3625	0.3125	-0.55	-0.2125
-0.0875	0.0625	0.65	-0.3625
-0.3125	-0.0625	-0.25	0.5625
-0.075	-0.375	0.7	-0.025

H = MULTIPLICACAO DE F POR G

H			
3.2625	0.625	-1.65	-0.425
-0.6125	0.3125	1.95	-1.0875
-1.5625	-0.1875	-0.5	1.125
-0.6	-1.125	0.7	-0.025

L = MULTIPLICACAO DE Q POR G

L			
2	2	-3	-1
0.9375	1.1875	0.75	-1.6875

MEDIA DOS ELEMENTOS DE H

MEDIA
0.0125

TRANSPOSTA DE H

TRANSPOSTA			
3.2625	-0.6125	-1.5625	-0.6
0.625	0.3125	-0.1875	-1.125
-1.65	1.95	-0.5	0.7
-0.425	-1.0875	1.125	-0.025

Capítulo 22 : Aplicações em Dendrometria

22.1 CÁLCULO DO VOLUME DE ÁRVORES CUBADAS

O presente exemplo aplica-se a árvores cubadas pelo método de Smalian.

A planilha para coleta de dados é apresentada no Apêndice 1.

Para processar os dados é necessário criar uma pasta chamada **CUBAGEM** dentro da pasta C:\EXEMPLOS\ do computador. Para tanto, abra o WINDOWS EXPLORER e clique em C:\EXEMPLOS\, depois em |Arquivo|Novo ►|Pasta| e altere o nome da nova pasta para CUBAGEM, onde serão armazenados os dados e arquivos deste exemplo.

Os dados e programas podem ser criados e salvos com qualquer programa de edição de textos ou planilhas, ou com o SAS System. Aqui será demonstrado como criar e editar os dados e programas utilizando o programa BLOCO DE NOTAS que vem com o WINDOWS e é acessível pelo menu |Iniciar|Programas|Acessórios|Bloco de notas|. Se o WINDOWS estiver em língua inglesa, procure o programa NOTE PAD.

Abra o BLOCO DE NOTAS e clique em |Arquivo|Salvar como...| encontre a pasta C:\EXEMPLOS\CUBAGEM\ e escolha |Salvar como tipo:|Todos os Arquivos|. Clique na área para |Nome do Arquivo:| e digite 'identif.dat', depois clique no botão |Salvar|.

Digite os dados do arquivo identif.dat, ou copie e cole no Bloco de Notas:

```
Piratini
RS702-KM30
fazenda guanabara
projeto de pinus
t02
```

Clique em |Arquivo|Salvar|.

Clique em |Arquivo|Novo|, depois em |Arquivo|Salvar como...| e escolha |Salvar como tipo:|Todos os Arquivos|. Clique na área para |Nome do Arquivo:| e digite 'arvores.dat', depois clique no botão |Salvar|.

Digite os dados do arquivo arvores.dat, ou copie e cole no Bloco de Notas:

```
1 21.05 16 26.1 2.9
2 18.8 12 16.6 1.6
3 21 15 23.6 2.9
4 19.8 13 17.8 2.2
5 19.3 13 21 1.9
6 20.8 15 25.8 3.2
7 17 11 13.6 0.4
8 21.8 15 23.6 3.2
```

Clique em |Arquivo|Salvar|.

Clique em |Arquivo|Novo|, depois em |Arquivo|Salvar como...| e escolha |Salvar como tipo:|Todos os Arquivos|. Clique na área para |Nome do Arquivo:| e digite 'medicoes.dat', depois clique no botão |Salvar|.

Digite os dados do arquivo medicoes.dat, ou copie e cole no Bloco de Notas:

```
1 0.2 31.5 27.1
1 1.3 26.1 23.2
1 3.0 25.1 22.0
1 5.8 20.1 18.1
1 8.6 19.1 17.8
1 11.4 16.6 14.6
1 14.2 13.7 13.1
1 17.0 7.6 7.0
1 19.9 4.8 4.1
1 21.05 0 0
2 0.2 21.6 18.1
2 1.3 16.6 15.0
2 3.0 16.2 14.6
2 5.8 15.0 13.4
2 8.6 13.4 12.7
2 11.4 11.8 10.8
2 14.2 9.2 8.6
2 17.0 3.8 3.2
2 18.8 0.0 0.0
3 0.2 29.3 25.5
3 1.3 23.6 20.7
3 3.0 22.0 19.7
3 5.8 21.6 19.1
3 8.6 20.4 18.1
3 11.4 17.2 15.6
3 14.2 14.3 12.7
3 17.0 8.6 8.0
3 19.9 3.7 3.1
3 21.00 0 0
4 0.2 23.2 19.7
4 1.3 17.8 15.6
4 3.0 16.9 15.0
4 5.8 15.9 14.6
4 8.6 15.0 13.4
4 11.4 12.7 11.8
4 14.2 10.5 9.5
4 17.0 6.4 5.7
4 19.80 0.0 0.0
5 0.2 26.7 23.6
5 1.3 21.0 19.1
5 3.0 19.4 17.5
5 5.8 18.5 17.2
5 8.6 17.2 15.0
5 11.4 14.0 13.4
5 14.2 10.8 10.2
5 17.0 7.0 6.0
5 19.30 0.0 0.0
6 0.2 30.2 26.4
6 1.3 25.8 22.6
6 3.0 23.9 21.3
6 5.8 23.9 21.3
6 8.6 20.7 19.7
```

```

6 11.4 18.5 17.5
6 14.2 15.3 14.3
6 17.0 9.9 9.2
6 19.9 3.8 3.2
6 20.80 0 0
7 0.2 18.1 16.2
7 1.3 13.4 13.4
7 3.0 12.4 12.1
7 5.8 10.2 10.2
7 8.6 8.9 8.3
7 11.4 7.0 6.7
7 14.2 5.1 4.8
7 17.00 0.0 0.0
8 0.2 28.3 24.5
8 1.3 23.6 20.4
8 3.0 22.6 19.4
8 5.8 21.3 18.8
8 8.6 19.7 18.1
8 11.4 19.1 16.6
8 14.2 17.5 13.7
8 17.0 16.2 9.2
8 19.9 11.1 8.6
8 21.80 0 0

```

Clique em |Arquivo|Salvar|.

Clique em |Arquivo|Novo|, depois em |Arquivo|Salvar como...| e escolha |Salvar como tipo:|Todos os Arquivos|. Clique na área para |Nome do Arquivo:| e digite 'pgm-cubagem.sas', depois clique no botão |Salvar|.

Digite o programa PGM-CUBAGEM.SAS, ou copie e cole:

```

OPTIONS LS=80 PS=40 NODATE NOSTIMER; title;
DATA IDENTIF;
  INFILE 'C:\EXEMPLOS\CUBAGEM\IDENTIF.DAT';
  INPUT #1 MUNIC $
        #2 LOCAL $
        #3 DIFTAL $
        #4 UNFTAL $
        #5 TALHAO $;
DATA ARVORES;
  INFILE 'C:\EXEMPLOS\CUBAGEM\ARVORES.DAT';
  INPUT ARVORE HTOTAL HCOM DAPCC CASCA;
  DAPSC=DAPCC-CASCA;
DATA MEDICOES;
  INFILE 'C:\EXEMPLOS\CUBAGEM\MEDICOES.DAT';
  INPUT ARVORE SECCAO ALTSEC DMTSEC DSCSEC;
PROC SORT DATA=MEDICOES; BY ARVORE SECCAO;
DATA MEDICOES (DROP=ARV_ANT SEC_ANT ALT_ANT DMT_ANT CAS_ANT
  AB_ANT VOL_ANT);
  RETAIN ARV_ANT 0 SEC_ANT 0 ALT_ANT 0 DMT_ANT 0 CAS_ANT 0 AB_ANT 0
  VOL_ANT 0;
  SET MEDICOES;
  ABSEC=3.1416*DMTSEC**2/40000;
  IF SECCAO=1 THEN DO; AB_ANT=ABSEC; SEC_ANT=0; ALT_ANT=0;
  DMT_ANT=0; CAS_ANT=0; VOL_ANT=0; END;
  ABMSEC=(ABSEC+AB_ANT)/2;
  COMPSEC=ALTSEC-ALT_ANT;

```

```

VOLSEC=COMPSEC*ABMSEC;
VOLSECAC=VOLSEC+VOL_ANT;
OUTPUT;
VOL_ANT=VOLSECAC;
ARV_ANT=ARVORE;
SEC_ANT=SECCAO;
ALT_ANT=ALTSEC;
DMT_ANT=DMTSEC;
CAS_ANT=DSCSEC;
AB_ANT=ABSEC;
PROC SORT DATA=MEDICOES; BY ARVORE SECCAO;
DATA MEDICOES;
  RETAIN ARV_ANT 0 SEC_ANT 0 ALT_ANT 0 DMT_ANT 0 CAS_ANT 0 AB_ANT 0
  VOL_ANT 0;
  SET MEDICOES;
  ABSECS=3.1416*DSCSEC**2/40000;
  IF SECCAO=1 THEN DO; AB_ANT=ABSECS; SEC_ANT=0; ALT_ANT=0;
  DMT_ANT=0; CAS_ANT=0; VOL_ANT=0; END;
  ABMSECS=(ABSECS+AB_ANT)/2;
  COMPSEC=ALTSEC-ALT_ANT;
  VOLSECS=COMPSEC*ABMSECS;
  VOLSECAS=VOLSECS+VOL_ANT;
  OUTPUT;
  VOL_ANT=VOLSECAS;
  ARV_ANT=ARVORE;
  SEC_ANT=SECCAO;
  ALT_ANT=ALTSEC;
  DMT_ANT=DMTSEC;
  CAS_ANT=DSCSEC;
  AB_ANT=ABSECS;
PROC SORT DATA=ARVORES; BY ARVORE;
PROC SORT DATA=MEDICOES; BY ARVORE;
DATA ARVOMEDI;
  MERGE ARVORES MEDICOES; BY ARVORE;
  IF DMTSEC=0 THEN OUTPUT;
DATA _NULL_;
  SET ARVOMEDI;
  FILENAME ARQSAIDA 'C:\EXEMPLOS\CUBAGEM\VARIAVEIS.DAT';
  FILE ARQSAIDA;
  PUT ARVORE HTOTAL HCOM DAPCC CASCA VOLSECAC VOLSECAS;
PROC PRINT data=ARVOMEDI;
  var ARVORE HTOTAL HCOM DAPCC CASCA VOLSECAC VOLSECAS;
RUN;

```

Clique em |Arquivo|Salvar|.

Para executar o programa, abra o SAS System. Ao abrir o SAS está selecionado o Editor, por padrão. Mantenha-o selecionado e clique em |File|Open|. Na tela de abertura de arquivos que se abre na área |Examinar:|, encontre e selecione a pasta |C:\EXEMPLOS\CUBAGEM|. Clique no programa |PGM-CUBAGEM.SAS| para o selecionar e depois clique no botão |Abrir|. Com o programa aberto, clique no menu |Run|Submit|. O programa é executado, aguarde-o encerrar o processamento.

O resultado do processamento do programa é a gravação do arquivo externo VARIAVEIS.DAT na pasta C:\EXEMPLOS\CUBAGEM\ e a impressão do relatório com as variáveis gravadas no arquivo. O conteúdo do arquivo VARIAVEIS.DAT é o seguinte:

```

1 21.05 16 26.1 2.9 0.5322241109 0.4279526594
2 18.8 12 16.6 1.6 0.2449692445 0.2004285429
3 21 15 23.6 2.9 0.5201953565 0.4110934397
4 19.8 13 17.8 2.2 0.2908137101 0.233356595
5 19.3 13 21 1.9 0.3758823476 0.3111557272
6 20.8 15 25.8 3.2 0.5985395955 0.4952985692
7 17 11 13.6 0.4 0.1230077379 0.114043575
8 21.8 15 23.6 3.2 0.613337042 0.4321576713
    
```

O arquivo VARIÁVEIS.DAT pode ser usado por outros programas SAS.

O relatório emitido pelo programa, com as variáveis selecionadas, que foram gravadas no arquivo VARIÁVEIS.DAT, impresso no modo HTML, fica como segue:

Obs	ARVORE	HTOTAL	HCOM	DAPCC	CASCA	VOLSECAC	VOLSECAS
1	1	21.05	16	26.1	2.9	0.53222	0.42795
2	2	18.80	12	16.6	1.6	0.24497	0.20043
3	3	21.00	15	23.6	2.9	0.52020	0.41109
4	4	19.80	13	17.8	2.2	0.29081	0.23336
5	5	19.30	13	21.0	1.9	0.37588	0.31116
6	6	20.80	15	25.8	3.2	0.59854	0.49530
7	7	17.00	11	13.6	0.4	0.12301	0.11404
8	8	21.80	15	23.6	3.2	0.61334	0.43216

22.2 FATOR DE CASCA E SELEÇÃO DE VARIÁVEIS DE REGRESSÃO

Esta aplicação demonstra a modelagem de equações e usa os dados do arquivo C:\EXEMPLOS\ANALISEDETRONCO\DADOS-VARIAV.DAT, a seguir:

DADOS-VARIAV.DAT

```

1 1 0 14 0 0 1.5100633914 0.0000379254 .
1 1 0 56 0 0 1.3828077314 0.0000173611 .
1 1 0 65 0 0 2.4 0.0000834392 .
2 1 0 14 . 0.8585 3.85 0.0002363281 1.06043541
2 1 0 56 . 0.569 2.0457979954 0.0000897156 .
2 1 0 65 . 1.226 3.85 0.000371948 0.8183717836
3 1 0 14 . 1.4175 3.7310594627 0.0009015986 1.5312463388
3 1 0 56 . 0.9145 2.4986507325 0.0002880265 .
3 1 0 65 . 1.7355 3.8927130232 0.0007667866 0.8326893662
4 1 0 14 . 2.501 4.3847872019 0.0022542778 1.0465057983
4 1 0 56 . 1.5445 3.3244024672 0.0008152588 1.3089292399
4 1 0 65 . 2.613 4.9142904786 0.0019908392 0.7554516295
5 1 0 14 . 3.681 5.3829157863 0.0045743188 0.7985228062
    
```

```

5 1 0 56 . 2.4555 5.3178277956 0.0023579792 0.9363450918
5 1 0 65 . 4.461 9.6775956284 0.0061170959 0.4044114949
6 1 0 14 7.8 5.894 8.3849436148 0.0116387236 0.5087390369
6 1 0 56 4.5 3.3035 7.569785124 0.0045996314 0.7089261093
6 1 0 65 6 5.192 10.52240576 0.0106971929 0.4801705653
7 1 0 14 11.5 8.914 10.688871656 0.0237832135 0.3565361176
7 1 0 56 7.2 5.6275 8.4019173554 0.0093860934 0.4491439739
7 1 0 65 10 8.3205 11.858958069 0.022889854 0.3549831653
8 1 0 14 14.5 11.084 12.197983715 0.0412072111 0.3501081797
8 1 0 56 9 7.1415 11.8230869 0.0174386608 0.3682251819
8 1 0 65 13 10.25 12.197073495 0.0385645775 0.3831733759
9 1 0 14 18.3 13.426 12.541178457 0.0655534768 0.3692106463
9 1 0 56 11.2 8.7935 12.098451817 0.0296392895 0.4033894837
9 1 0 65 15.5 12.5505 12.752045228 0.0642778446 0.407445498
10 1 0 14 19.5 14.929 13.363261861 0.0907861999 0.3881110803
10 1 0 56 12.5 9.8445 12.715513428 0.0424577598 0.4386781136
10 1 0 65 17.2 13.9975 13.715796475 0.0893998754 0.4235698585
11 1 0 14 21 16.6965 14.033597839 0.1228212144 0.3997271023
11 1 0 56 13.5 10.8285 13.557725118 0.0577335083 0.4623966104
11 1 0 65 19.5 15.338 14.485400732 0.1205058805 0.4502465337
12 1 0 14 22.6 17.6885 15.06025125 0.1494975381 0.4039521559
12 1 0 56 14 11.3475 14.212827804 0.0695467358 0.4838447049
12 1 0 65 19.5 16.1795 15.202304501 0.1435814547 0.4593765776
13 1 0 14 24 19.084 15.957726552 0.1925839079 0.4219104471
13 1 0 56 14.5 11.979 14.875934492 0.0870377655 0.5191495291
13 1 0 65 20.5 17.258 16.122445844 0.1848462425 0.4901272955
14 1 0 14 25 20.0365 16.659177949 0.2293871125 0.4366986451
14 1 0 56 15 12.5035 15.642180273 0.1031372672 0.5369888334
14 1 0 65 20.5 17.988 18.365735476 0.2232828519 0.4784002596
15 1 0 14 26.5 20.6775 19.377908996 0.260157928 0.3998013099
15 1 0 56 16 12.9405 16.432579072 0.1174887755 0.5436240369
15 1 0 65 22 18.353 19.633868974 0.2529028207 0.4869044841
16 1 0 14 . 21.244 20.182820227 0.3047529585 0.4259941269
16 1 0 56 . 13.535 17.737741935 0.1398814766 0.5480945382
16 1 0 65 . 19.0235 20.195197256 0.2912044583 0.5073167405
17 1 0 14 . 21.6295 20.600105094 0.3459902665 0.4571004024
17 1 0 56 . 14.06 18.233225806 0.1633310227 0.5769585959
17 1 0 65 . 19.5485 20.549022298 0.3270390058 0.5302633752
18 1 0 14 27.5 22.2505 21.05 0.394519536 0.4819999385
18 1 0 56 17 14.6825 18.8 0.1879647836 0.5905120161
18 1 0 65 23.5 20.273 21 0.3753504867 0.5537219842

```

O programa inicial de processamento dos dados, PGM-FATORDECASCA.SAS, cria as variáveis independentes X0 a X7 a partir de transformações das variáveis originais d, d_{sc}, h, v e f_{1,3} (=FF). A seleção de variáveis significativas é realizada pelo coeficiente de determinação, como segue:

PGM-FATORDECASCA.SAS

```

OPTIONS LS=120 PS=40 NODATE NOSTIMER;          1
DATA A;                                          2
INFILE 'C:\EXEMPLOS\ANALISEDETRONCO\DADOS-      3
  VARIIV.DAT';                                  4
INPUT IDADE BLOCO TRAT ARVORE D DSC H V FF;    5
IF D<5 THEN DELETE;                             6
DATA A; SET A;                                   7
FCASCA=D**2/DSC**2;                             8
Y=FCASCA;                                        9
X0=IDADE;                                       10
X1=D;                                           11

```

```

X2=D**2;
D2=X2;
X3=H/D;
X4=H;
X5=H**2;
X6=1/D;
X7=1/H;
U_H=X7;
PROC REG;
MODEL Y=X0 X1 X2 X3 X4 X5 X6 X7
/ SELECTION=RSQUARE MSE JP GMSEP CP AIC BIC
SBC B BEST=1;
PROC REG;
MODEL Y=D D2 U_H;
RUN;

```

12
13
14
15
16
17
18
19
20
21
22
23
24

O relatório de resultados do processamento é o seguinte:

The REG Procedure
Model: MODEL1
Dependent Variable: Y
R-Square Selection Method

Number in Model	R- Square	C(p)	AIC	BIC	Estimated MSE of Prediction	J(p)	MSE	SBC	Parameter Estimates								
									Intercept	X0	X1	X2	X3	X4	X5	X6	X7
1	0,25	48,89	-137,49	-138,50	0,01	0,01	0,01	-134,55	1,83	,	,	,	,	-0,02	,	,	,
2	0,63	12,10	-157,96	-156,97	0,01	0,01	0,01	-153,56	1,11	,	,	,	,	,	,	-4,83	10,64
3	0,67	10,20	-159,42	-157,94	0,01	0,01	0,01	-153,55	3,26	,	-0,09	0,00	-0,85	,	,	,	,
4	0,69	9,48	-160,07	-157,90	0,01	0,01	0,01	-152,74	4,11	,	-0,13	0,00	,	-0,05	,	-9,22	,
5	0,77	4,02	-166,72	-160,96	0,01	0,01	0,00	-157,93	-2,49	,	,	0,00	4,07	-0,06	,	-45,73	51,55
6	0,77	5,23	-165,79	-158,79	0,01	0,01	0,00	-155,53	-1,11	,	-0,05	0,00	3,42	-0,06	,	-41,43	43,66
7	0,77	7,14	-163,92	-156,08	0,01	0,01	0,00	-152,19	-0,27	,	-0,04	0,00	3,31	-0,12	0,00	-40,14	39,26
8	0,78	9,00	-162,11	-153,37	0,01	0,01	0,01	-148,92	0,76	0,01	-0,04	0,00	3,10	-0,18	0,00	-37,90	33,40

The REG Procedure
 Model: MODEL1
 Dependent Variable: Y

Analysis of Variance					
Source	DF	207umo f Squares	Mean Square	F Value	Pr > F
Model	3	0.28599	0.09533	11.73	<.0001
Error	28	0.22760	0.00813		
Corrected Total	31	0.51359			

Root MSE	0.09016	R-Square	0.5568
Dependent Mean	1.54708	Adj R-Sq	0.5094
Coeff Var	5.82773		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	0.14604	0.27711	0.53	0.6023
D	1	0.05767	0.01891	3.05	0.0050
D2	1	-0.00101	0.00049707	-2.04	0.0508
U_H	1	10.35827	1.76410	5.87	<.0001

A equação gerada para função de casca foi a seguinte:

$$FCASCA = -2.48752 + 0.00187 * D^{**2} + 4.07332 * H/D - 0.06254 * H - 45.73107/D + 51.55283/H$$

Entretanto, o intercepto não apresenta significância ao nível de 5% de probabilidade. Deve-se, portanto eliminá-lo do modelo. Também, deve-se verificar se não há inflação de variância entre as variáveis. Isso pode ser realizado acrescentando-se a opção VIF ao modelo. Assim, teríamos o seguinte programa:

```

OPTIONS LS=120 os=40 NODATE NOSTIMER;
DATA A;
INFILE 'C:\EXEMPLOS\ANALISEDETRONCO\DADOS-VARIAV.DAT';
INPUT IDADE BLOCO TRAT ARVORE D DSC H V FF;
IF D<5 THEN DELETE;
DATA A; SET A;
FCASCA=D**2/DSC**2;
Y=FCASCA;
X0=IDADE;
X1=D;
X2=D**2;
D2=X2;
X3=H/D;
X4=H;
X5=H**2;
X6=1/D;
X7=1/H;
U_H=X7;
PROC REG;
MODEL Y=X0 X1 X2 X3 X4 X5 X6 X7

```

```

/ NOINT VIF SELECTION=RSQUARE MSE JP GMSEP CP AIC BIC SBC B
BEST=1;
PLOT R.*P.;
RUN;
    
```

Os resultados são apresentados a seguir.

The SAS System
The REG Procedure
Model: MODEL1
Dependent Variable: Y
R-Square Selection Method

Number of Observations Read 32
Number of Observations Used 32

Note: No intercept in model. R-Square is redefined.

Number in Model	R-Square	C(p)	AIC	BIC	Estimated MSE of Prediction	J(p)	MSE	SBC	Parameter Estimates								
									X0	X1	X2	X3	X4	X5	X6	X7	
1	0.958	649.1	-71.1	-72.8	0.108	0.108	0.105	-69.6	20.3366
2	0.996	39.8	-142.8	-143.5	0.012	0.012	0.011	-139.9	.	0.0336	13.5000
3	0.997	21.9	-152.0	-152.2	0.009	0.009	0.008	-147.6	.	0.0662	-0.0012	11.1888
4	0.998	11.6	-159.4	-158.3	0.007	0.007	0.006	-153.6	.	.	0.0004	0.8237	.	.	-13.0831	21.4498	
5	0.998	7.0	-164.0	-161.0	0.006	0.006	0.005	-156.7	.	.	0.0010	1.7888	-0.0494	.	-21.8127	24.7836	
6	0.999	4.5	-167.4	-161.9	0.006	0.005	0.005	-158.6	.	-0.0686	0.0028	2.6945	-0.0558	.	-35.0412	35.0220	
7	0.999	6.1	-165.9	-159.5	0.006	0.006	0.005	-155.7	.	-0.0437	0.0025	3.2220	-0.1292	0.0017	-39.3198	37.5861	
8	0.999	8.0	-164.1	-157.0	0.007	0.006	0.005	-152.3	0.0046	0.0405	0.0025	3.3042	-0.1424	0.0019	-39.9212	37.9158	

The SAS System
The REG Procedure
Model: MODEL1
Dependent Variable: Y

Number of Observations Read 32
Number of Observations Used 32

Note: No intercept in model. R-Square is redefined.

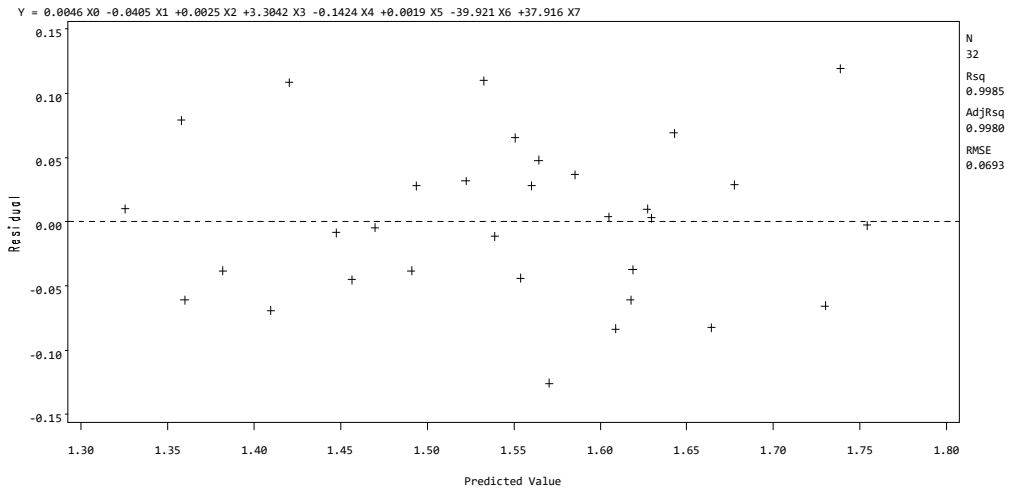
Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	8	76.989	9.6236	2005.4	<.0001
Error	24	0.1152	0.0048		
Uncorrected Total	32	77.104			

Root MSE	0.06927	R-Square	0.9985
Dependent Mean	1.54708	Adj R-Sq	0.998
Coeff Var	4.47772		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t	Variance Inflation
X0	1	0.0046	0.0133	0.34	0.7358	166.4
X1	1	-0.04	0.0513	-0.79	0.4382	5479
X2	1	0.0025	0.0011	2.35	0.0272	983.1
X3	1	3.3042	1.0317	3.2	0.0038	6467
X4	1	-0.142	0.1222	-1.16	0.2555	21962
X5	1	0.0019	0.0027	0.69	0.4966	2817
X6	1	-39.92	10	-3.99	0.0005	3813
X7	1	37.916	6.5515	5.79	<.0001	1597

The REG Procedure
Plot of RESIDUAL vs PRED



No teste t, as variáveis significativas, por ordem de maior significância, foram X7 (1/h), X6 (1/d), X3 (h/d) e X2 (d²).

Capítulo 23 : Aplicações em inventários florestais

Os procedimentos relacionados à amostragem são relacionados na Tabela 6, mas outros procedimentos do SAS também são úteis para esta finalidade, como: PROC NESTED, PROC ANOVA, PROC GLM e PROC MEANS.

TABELA 6 : Procedimentos para amostragem no SAS/STAT e suas funcionalidades

PROCEDIMENTO	CARACTERÍSTICAS
PROC SURVEYFREQ	<ul style="list-style-type: none"> • tabelas de frequência unidirecional • tabelas cruzadas de duas vias e multidões • totais da população e proporções • erros padrão • limites de confiança, coeficientes de variação e efeitos de design • testes de qualidade de ajuste • testes de independência • riscos e diferenças de risco • razões odds e riscos relativos • frequência ponderada e parcelas percentuais • Mosaicos • razão odds, risco relativo e parcelas de diferença de risco
PROC SURVEYIMPUTE	<ul style="list-style-type: none"> • imputação de hot-deck com os seguintes métodos de seleção de doadores: <ul style="list-style-type: none"> - bootstrap bayesiano aproximado - amostras aleatórias simples sem substituição - Amostras aleatórias simples com substituição - probabilidade proporcional aos pesos dos - entrevistados com substituição • imputação fraccional fraca e eficiente do baralho - Pesos de repetição ajustados por imputação - Pesos de replicação repetida equilibrada ajustada por imputação (BRR) - Pesos de Jackknife ajustados à imputação - imputar variáveis de forma conjunta ou independente
PROC SURVEYLOGISTIC	<ul style="list-style-type: none"> • ajuste cumulativo da regressão de logit • logit, log-log complementares e funções de link probit • Modelo modelo de regressão logística • estimativas de coeficientes de regressão • estimativas de matrizes de covariância • testes de hipóteses • diagnóstico de modelos • estimativas de odds ratios • limites de confiança • funções estimáveis • estimativas e erros padrão para contrastes • análise de domínio

PROCEDIMENTO	CARACTERÍSTICAS
PROC SURVEYMEANS	<ul style="list-style-type: none"> • estimativas de médias e totais da população • estimativas de proporções populacionais • erros padrão • limites de confiança • testes de hipóteses (testes t) • análise de domínio • estimativas de razão
PROC SURVEYPHREG	<ul style="list-style-type: none"> • análise de regressão baseada no modelo de risco proporcional de Cox • estimativas da razão de risco • valores previstos e seus erros padrão • resíduos de martingale, Schoenfeld, pontuação e desvio • testes de significância • limites de confiança • funções estimáveis • análise de domínio
PROC SURVEYREG	<ul style="list-style-type: none"> • ajuste do modelo de regressão linear • estimativas de coeficientes de regressão • estimativas de matrizes de covariância • testes de significância • limites de confiança • funções estimáveis • estimativas e erros padrão para contrastes • análise de domínio
PROC SURVEYSELECT	<ul style="list-style-type: none"> • amostragem aleatória simples • amostragem aleatória sem restrições (com substituição) • amostragem sistemática • amostragem sequencial • Probabilidade de seleção proporcional ao tamanho (PPS) com e sem substituição • Amostragem sistemática PPS • PPS para duas unidades por estrato • PPS sequencial com substituição mínima

Neste manual, descreve-se o procedimento SURVEYMEANS e são dados alguns exemplos de uso dos procedimentos GLM, MEANS e NESTED.

23.1 PROC SURVEYMEANS

Este procedimento pode ser considerado o mais importante para amostragem, pois é com ele que se calcula os resultados estatísticos da amostragem como média, desvio padrão, limites de confiança e erro amostral.

TABELA 7 : Dados da amostragem

Estrato	UA	Arvore	d	h	S	Estrato	UA	Arvore	d	h	S	Estrato	UA	Arvore	d	h	S	Estrato	UA	Arvore	d	h	S
1	1	1	40.7	30.0	0	1	2	25	29.6	29.0	0	1	4	21	33.4	28.7	0	2	6	16	35.0	28.1	2
1	1	2	33.1	29.0	0	1	2	26	34.6	30.2	0	1	4	22	30.8	27.1	0	2	6	17	31.5	31.0	2
1	1	3	26.4	29.1	0	1	2	27	35.3	31.6	0	1	4	23	32.1	27.6	0	2	6	18	27.6	28.0	0
1	1	4	35.9	28.2	0	1	2	28	20.6	20.7	0	1	4	24	29.2	26.6	0	2	6	19	26.7	28.1	0
1	1	5	32.1	28.6	0	1	3	1	33.7	28.4	2	1	4	25	32.4	26.1	0	2	6	20	27.3	31.0	0
1	1	6	31.8	28.7	0	1	3	2	25.7	29.2	0	2	5	1	30.5	28.7	0	2	6	21	32.4	28.0	0
1	1	7	37.8	30.3	0	1	3	3	27.6	27.6	0	2	5	2	29.6	27.6	0	2	6	22	29.2	29.3	0
1	1	8	27.6	25.4	0	1	3	4	27.0	28.7	0	2	5	3	28.9	29.5	0	2	6	23	31.5	29.9	0
1	1	9	35.9	29.1	0	1	3	5	23.2	25.5	0	2	5	4	25.7	25.8	0	2	6	24	28.0	27.6	0
1	1	10	29.9	27.5	0	1	3	6	28.0	25.7	0	2	5	5	41.3	29.2	0	2	6	25	34.0	26.4	0
1	1	12	28.3	27.3	0	1	3	7	36.2	26.6	0	2	5	6	35.0	29.3	0	2	6	26	23.5	27.2	0
1	1	13	36.2	29.7	0	1	3	8	31.5	28.7	0	2	5	7	31.8	29.5	0	2	6	27	25.7	30.7	0
1	1	14	30.8	27.5	0	1	3	9	27.3	27.8	0	2	5	8	25.1	28.5	0	2	6	28	26.1	30.8	0
1	1	15	38.5	28.4	0	1	3	10	28.6	25.0	0	2	5	9	36.6	30.2	0	2	6	29	35.0	30.9	0
1	1	16	26.1	28.7	0	1	3	11	31.5	28.2	0	2	5	10	34.3	28.2	0	2	6	30	29.9	29.1	0
1	1	17	31.1	27.0	0	1	3	12	29.6	26.6	0	2	5	11	31.8	27.2	0	2	6	31	30.5	29.5	0
1	1	18	27.6	25.1	0	1	3	13	25.1	26.4	0	2	5	12	33.4	28.7	0	2	6	32	19.4	22.9	0
1	1	19	33.4	28.0	0	1	3	14	25.7	25.5	0	2	5	13	31.1	28.3	0	2	6	33	32.1	30.1	0
1	1	20	34.6	26.2	0	1	3	15	24.1	25.9	0	2	5	14	34.3	30.5	0	2	6	34	24.5	28.4	0
1	1	21	33.1	28.1	0	1	3	16	40.1	30.2	0	2	5	15	26.4	28.3	0	2	6	35	28.3	30.3	0
1	1	22	25.7	26.7	0	1	3	17	27.6	26.9	0	2	5	16	39.7	29.8	0	2	7	1	33.4	28.7	0
1	1	23	30.8	28.5	0	1	3	18	26.7	28.8	0	2	5	17	31.5	30.7	0	2	7	2	31.1	28.3	0
1	1	24	31.8	25.5	0	1	3	19	30.8	26.2	2	2	5	18	38.1	29.7	0	2	7	3	34.3	30.5	0
1	1	25	30.8	26.1	0	1	3	20	28.3	26.1	0	2	5	19	26.7	30.2	2	2	7	4	26.4	28.3	0
1	1	26	28.6	24.2	0	1	3	21	35.9	29.3	0	2	5	20	42.3	30.7	0	2	7	5	39.7	29.8	0
1	2	1	36.9	29.8	0	1	3	22	45.8	29.8	0	2	5	21	28.3	28.4	0	2	7	6	31.5	30.7	0
1	2	2	27.6	26.5	0	1	3	23	32.1	30.7	0	2	5	22	29.6	28.7	2	2	7	7	38.1	29.7	0
1	2	3	35.6	28.3	2	1	3	24	23.8	29.3	0	2	5	23	33.1	19.5	4	2	7	8	26.7	30.2	2
1	2	4	29.9	27.1	0	1	3	25	33.4	28.2	3	2	5	24	35.3	28.4	0	2	7	9	42.3	30.7	0
1	2	5	30.2	27.6	2	1	4	1	32.7	28.5	0	2	5	25	27.6	27.4	0	2	7	10	28.3	28.4	0
1	2	6	33.1	29.2	0	1	4	2	30.8	30.3	0	2	5	26	30.5	28.2	0	2	7	11	29.6	28.7	2
1	2	7	21.0	13.6	4	1	4	3	29.9	28.3	0	2	5	27	25.1	27.1	0	2	7	12	33.1	19.5	4
1	2	8	29.2	27.1	0	1	4	4	30.8	29.8	0	2	5	28	27.6	29.8	2	2	7	13	35.3	28.4	0
1	2	9	33.7	29.1	0	1	4	5	32.1	29.3	0	2	5	29	32.7	29.7	0	2	7	14	27.6	27.4	0
1	2	10	27.3	28.2	0	1	4	6	30.5	30.1	0	2	6	1	33.4	26.9	0	2	7	15	30.5	28.2	0
1	2	11	26.1	25.0	0	1	4	7	33.7	30.3	0	2	6	2	32.4	26.9	0	2	7	16	25.1	27.1	0
1	2	12	24.5	24.1	0	1	4	8	23.2	25.7	2	2	6	3	29.2	26.7	0	2	7	17	27.6	29.8	2
1	2	13	29.2	27.7	0	1	4	9	27.0	29.0	0	2	6	4	23.8	25.8	0	2	7	18	32.7	29.7	0
1	2	14	35.0	30.5	0	1	4	10	33.7	28.9	0	2	6	5	33.1	28.7	0	2	7	19	33.4	26.9	0
1	2	15	32.1	29.8	0	1	4	11	33.7	27.5	0	2	6	6	32.1	27.7	0	2	7	20	32.4	26.9	0
1	2	16	40.1	30.8	0	1	4	12	29.6	27.8	0	2	6	7	29.6	28.2	2	2	7	21	29.2	26.7	0
1	2	17	47.1	28.7	0	1	4	13	28.0	27.3	2	2	6	8	32.1	28.0	0	2	7	22	23.8	25.8	0
1	2	18	34.3	30.8	0	1	4	14	32.1	29.8	0	2	6	9	30.5	27.7	0	2	7	23	33.1	28.7	0
1	2	19	24.8	24.7	0	1	4	15	31.8	28.0	0	2	6	10	36.6	29.4	0	2	7	24	32.1	27.7	0
1	2	20	25.7	27.6	0	1	4	16	31.1	26.6	2	2	6	11	27.0	28.0	0	2	7	25	29.6	28.2	2
1	2	21	24.5	27.7	0	1	4	17	28.3	25.9	0	2	6	12	26.1	26.6	2	2	7	26	32.1	28.0	0
1	2	22	32.7	26.9	0	1	4	18	32.4	27.6	0	2	6	13	29.9	28.1	0	2	7	27	30.5	27.7	0
1	2	23	38.1	28.3	0	1	4	19	33.1	27.4	0	2	6	14	22.9	27.2	0	2	7	28	36.6	29.4	0
1	2	24	31.1	27.4	0	1	4	20	32.1	28.4	0	2	6	15	30.5	28.7	0	2	7	29	27.0	28.0	0

Onde: Estrato=número do estrato amostral; UA=número da unidade amostral; Arvore=número da árvore na parcela; d=diâmetro à 1,3m de altura em centímetros; h=altura da árvore em metros (usado ponto decimal no lugar de vírgula); S=situação da árvore (0=norma;1=morta;2=bifurcada;3=atacada por praga ou doença; 4=quebrada).

O arquivo de dados deve ser gravado em uma única sequência no formato xlsx do MS-Excel na pasta [myfolders] e depois importado com o programa “Exemplo – Importando arquivo do excel para My Folders”.

```

*Exemplo – Importando arquivo do excel para My Folders;
LIBNAME myfolder '/folders/myfolders/';
FILENAME arquivo '/folders/myfolders/Inventario.xlsx';
PROC IMPORT DATAFILE=arquivo
  DBMS=XLSX
  OUT=myfolder.Inventario;
  GETNAMES=YES;
RUN;
PROC CONTENTS DATA=myfolder.Inventario;
RUN;

```

Após a importação do arquivo, o processamento pode ser realizado com o “Exemplo – Estatísticas de amostragem com o PROC SURVEYMEANS”. As áreas dos dois estratos são de 400 e 100 ha e a áreas da unidade amostral (UA) é de 400 m². As médias gerais são ponderadas pelas áreas dos estratos com a instrução WEIGHT. As variáveis analisadas foram o volume/ha (Volume) e o número de árvores/ha (Arvores).

```

*Exemplo – Estatísticas de amostragem com o PROC SURVEYMEANS;
LIBNAME myfolder '/folders/myfolders/';
DATA dados; set myfolder.Inventario;
  v=0.45*h*3.1416*d**2/40000;
  If Estrato=1 then Area=400;
  If Estrato=2 then Area=100;
  AreaDaUA=400;
  Arvores=10000/AreaDaUA;
  Volume=Arvores*v;
PROC SORT data=dados; by Estrato UA;
PROC SUMMARY DATA=dados;
  Var Volume Arvores;
  By Estrato UA Area;
  OUTPUT OUT=medias (drop=_TYPE_)
    sum(Volume)=Volume sum(Arvores)=Arvores max(Area)=Area;
PROC SORT data=medias; by Estrato UA;
PROC SURVEYMEANS DATA=medias;
  VAR Volume Arvores;
  DOMAIN Estrato;
  STRATA Estrato;
  WEIGHT Area;
RUN;

```

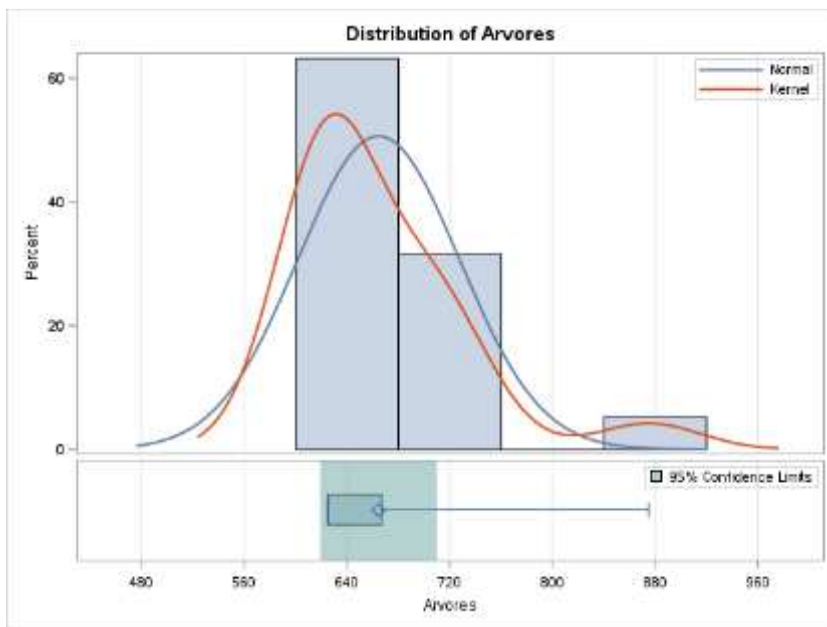
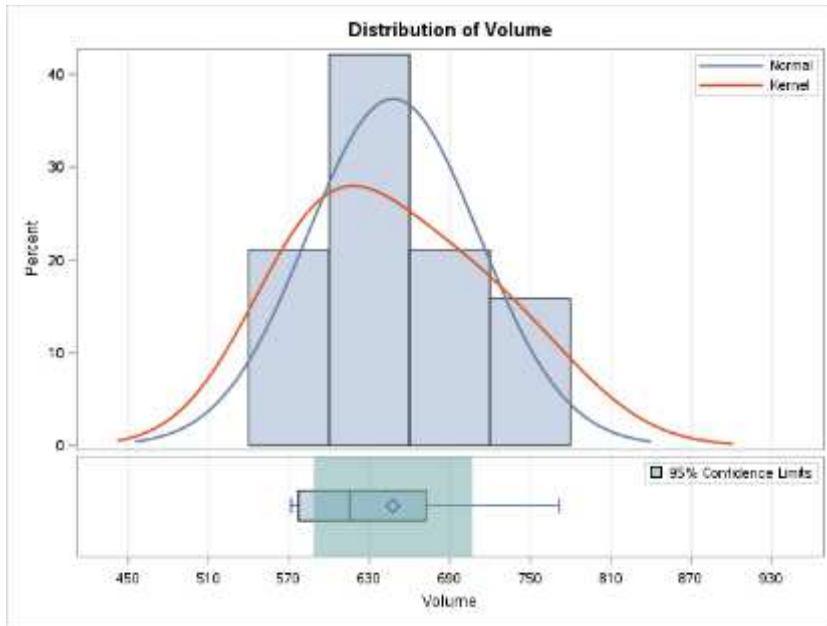
Os resultados são apresentados no relatório a seguir.

The SURVEYMEANS Procedure

Data Summary	
Number of Strata	2
Number of Observations	7
Sum of Weights	1900

Statistics					
Variable	N	Mean	Std Error of Mean	95% CL for Mean	
Volume	7	647.704451	23.036654	588.486846	706.922056
Arvores	7	664.473684	17.653168	619.094771	709.852598

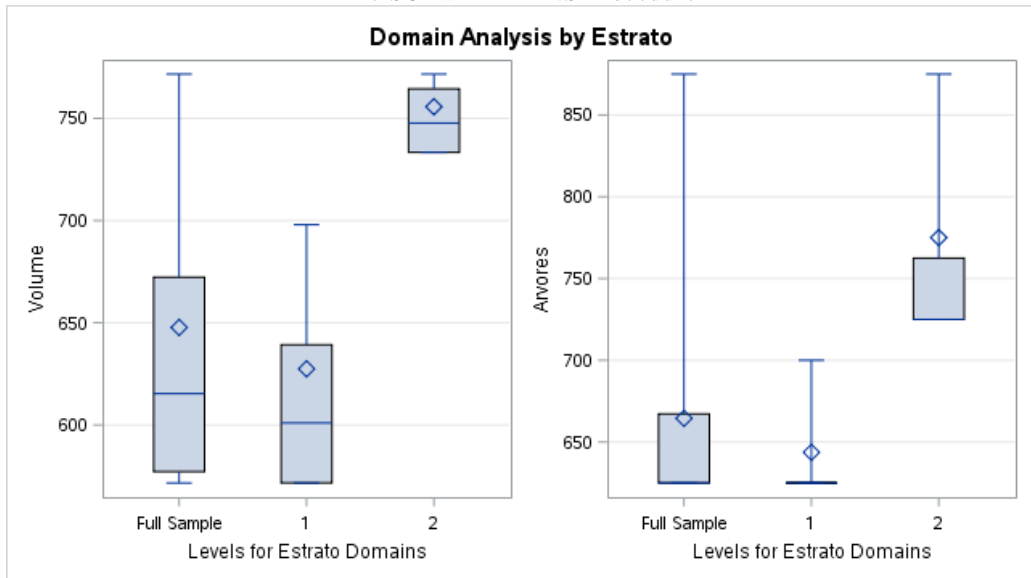
The SURVEYMEANS Procedure



The SURVEYMEANS Procedure

Statistics for Estrato Domains						
Estrato	Variable	N	Mean	Std Error of Mean	95% CL for Mean	
1	Volume	4	627.456571	27.270985	540.668126	714.245015
	Arvores	4	643.750000	18.750000	584.079132	703.420868
2	Volume	3	755.693146	11.495288	706.232914	805.153379
	Arvores	3	775.000000	50.000000	559.867364	990.132636

The SURVEYMEANS Procedure



23.2 AMOSTRAGEM ESTRATIFICADA

A amostragem estratificada pode ser considerada uma amostragem com mais de um efeito aleatório, o efeito dos estratos e das parcelas em cada estrato. Assim, o modelo é linear e pode-se utilizar ou o PROC NESTED ou o PROC ANOVA ou o PROC GLM, sendo o PROC GLM preferível, pois a amostragem em diferentes estratos quase sempre resulta em um número diferente de parcelas para cada estrato, constituindo-se em uma série de dados não balanceados que o PROC ANOVA não analisa corretamente e o PROC NESTED emite aviso de que há problema de balanceamento dos dados para realizar os testes estatísticos. Quando os dados são balanceados, deve-se dar preferência ao PROC NESTED, pois este separa os componentes da variância.

O “Exemplo – Amostragem estratificada”, demonstra o uso dos procedimentos MEANS e NESTED com amostragem estratificada.

```

OPTIONS LS=80 PS=58 NODATE NOSTIMER;
TITLE;
DATA ARQUIVO;
  INPUT ESTRATO PARCELA Y @@;
  Y=Y/10;
DATALINES;
2 1 204 1 1 158 3 1 213
2 2 305 1 2 076 3 2 243
2 3 268 1 3 088 3 3 292
2 4 272 1 4 125 3 4 218
2 5 284 1 5 111 3 5 331
2 6 197 1 6 162 3 6 358
2 7 204 1 7 122 3 7 267
2 8 231
;
PROC SORT DATA=ARQUIVO;
  BY ESTRATO PARCELA;
TITLE1 'INVENTARIO FLORESTAL - AMOSTRAGEM
  ESTRATIFICADA';
TITLE2 'MEDIAS POR ESTRATO';
PROC MEANS DATA=ARQUIVO;
  CLASS ESTRATO;
  VAR Y;
PROC NESTED DATA=ARQUIVO;
TITLE2 'ANALISE DA VARIANCIA';
  CLASS ESTRATO;
  VAR Y;
RUN;

```

Neste exemplo, os dados não são balanceados, portanto o teste F não é válido. Para validar o teste F é necessário ponderar os dados e fazer os cálculos como no exemplo de amostragem estratificada ponderada na próxima secção.

O sistema não calcula o teste F e emite o seguinte aviso através do LOG:

"WARNING: The design is not balanced--F tests cannot be computed."

O relatório emitido pelo programa é o seguinte:

INVENTARIO FLORESTAL - AMOSTRAGEM ESTRATIFICADA
 MEDIAS POR ESTRATO
 The MEANS Procedure
 Analysis Variable : Y

ESTRATO	N Obs	N	Mean	Std Dev	Minimum	Maximum
1	7	7	12.0285714	3.2345604	7.6000000	16.2000000
2	8	8	24.5625000	4.1812293	19.7000000	30.5000000
3	7	7	27.4571429	5.5518337	21.3000000	35.8000000

INVENTARIO FLORESTAL - AMOSTRAGEM ESTRATIFICADA
ANALISE DA VARIANCIA
The NESTED Procedure

Coefficients of Expected Mean Squares		
Source	ESTRATO	Error
ESTRATO	7.31818182	1.00000000
Error	0.00000000	1.00000000

Nested Random Effects Analysis of Variance for Variable Y								
Variance Source	DF	Sum of Squares	F Value	Pr > F	Error Term	Mean Square	Variance Component	Percent of Total
Total	21	1321.4895				62.9280	81.819	100.000
ESTRATO	2	951.3993				475.6996	62.340	76.193
Error	19	370.0901				19.4784	19.478	23.806
Y Mean						21.49545455		
Standard Error of Y Mean						4.66385207		

23.3 AMOSTRAGEM ESTRATIFICADA PONDERADA

Utilizando-se os mesmos dados do exemplo anterior, realizando-se a análise da variância com o PROC GLM com os dados ponderados através do peso de cada estrato com a opção WEIGHT, obtém-se os testes estatísticos. Neste exemplo foi utilizado o PROC SURVEYMEANS para obter as médias por estrato, pois este permite o uso da opção WEIGHT que não existe com o PROC MEANS. O PROC SURVEYMEANS é o mais indicado para obter médias e estatísticas de dados amostrados onde os efeitos são aleatórios.

No “Exemplo – Amostragem estratificada ponderada”, demonstra-se a forma de ponderação com WEIGHT.

```

Exemplo – Amostragem estratificada ponderada;
OPTIONS LS=80 PS=58 NODATE NOSTIMER;
TITLE;
* ----- ATENCAO -----;
* SEM PONDERACAO O TESTE F NAO EH VALIDO, ENTAO FOI USADO
  O PROC SURVEYMEANS PARA CALCULAR MEDIAS E O PROC GLM PARA
  FAZER A ANALISE DA VARIANCIA, AMBOS COM PONDERACAO, PARA
  OBTER AS ESTATISTICAS CORRETAS E VALIDAR O TESTE F DO GLM;
* -----;
DATA ARQUIVO;
* USOU-SE A OPCAO EXPANDTABS COM A INSTRUCAO INFILE PORQUE
  O ARQUIVO EH SEPARADO POR TABULACOES - ISSO OCORRE QUANDO

```

```

UMA PLANILHA DO MS-EXCEL EH GRAVADA COMO SOMENTE TEXTO;
INFILE 'C:\EXEMPLOS\INVENTARIO\DAADOS-ESTRATIFICADA.SAS'
EXPANDTABS;
INPUT ESTRATO PARCELA Y NJ @@ ;
Y=Y/10;
* USOU-SE, PARA PONDERAR, O NUMERO DE ESTRATOS MULTIPLICADO
PELO NUMERO POTENCIAL DE PARCELAS DO ESTRATO, DIVIDIDOS
PELO NUMERO POTENCIAL TOTAL DE OBSERVACOES (M*Nj/N);
PESO=3*NJ/450;
PROC SORT DATA=ARQUIVO;
BY ESTRATO PARCELA;
TITLE1 'INVENTARIO FLORESTAL - AMOSTRAGEM ESTRATIFICADA';
TITLE2 'MEDIAS POR ESTRATO - VALORES PONDERADOS';
PROC SURVEYMEANS DATA=ARQUIVO MEAN NOBS STD STDERR CLM;
CLASS ESTRATO;
DOMAIN ESTRATO;
VAR Y;
WEIGHT PESO;
PROC GLM DATA=ARQUIVO;
TITLE2 'ANALISE DA VARIANCIA - VALORES PONDERADOS';
CLASS ESTRATO PARCELA;
MODEL Y=ESTRATO / SS1;
WEIGHT PESO;
RUN;

```

O Relatório emitido pelo programa é o seguinte:

INVENTARIO FLORESTAL - AMOSTRAGEM ESTRATIFICADA
 MEDIAS POR ESTRATO - VALORES PONDERADOS
 The SURVEYMEANS Procedure

Data Summary

Number of Observations	22
Sum of Weights	22.0933333

Statistics

Variable	N	Mean	Std Error of Mean	Lower 95% CL for Mean	Upper 95% CL for Mean	Std Dev
Y	22	21.618346	1.644412	18.198605	25.038088	38.745621

Domain Analysis: ESTRATO

ESTRATO	Variable	N	Mean	Std Error of Mean	Lower 95% CL for Mean	Upper 95% CL for Mean	Std Dev
1	Y	7	12.028571	1.158496	9.619347	14.437796	26.968930
2	Y	8	24.562500	1.415353	21.619113	27.505887	63.242447
3	Y	7	27.457143	1.988455	23.321925	31.592361	59.596536

INVENTARIO FLORESTAL - AMOSTRAGEM ESTRATIFICADA
 ANALISE DA VARIANCIA - VALORES PONDERADOS
 The GLM Procedure

Class Level Information

Class	Levels	Values
ESTRATO	3	1 2 3
PARCELA	8	1 2 3 4 5 6 7 8

Number of observations 22

INVENTARIO FLORESTAL - AMOSTRAGEM ESTRATIFICADA
 ANALISE DA VARIANCIA - VALORES PONDERADOS
 The GLM Procedure
 Dependent Variable: Y
 Weight: PESO

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	919.726387	459.863194	23.67	<.0001
Error	19	369.137910	19.428311		
Corrected Total	21	1288.864297			

R-Square	Coeff Var	Root MSE	Y Mean
0.713594	20.38896	4.407756	21.61835

Source	DF	Type I SS	Mean Square	F Value	Pr > F
ESTRATO	2	919.7263874	459.8631937	23.67	<.0001

23.4 AMOSTRAGEM EM CONGLOMERADOS

A amostragem em conglomerados é um modelo aninhado, onde um grupo de subunidades está dentro das unidades de maior grau. Para efeitos aleatórios aninhados, como é o caso dos inventários em conglomerados, utiliza-se o PROC NESTED para análise da variância. Os dados utilizados no exemplo a seguir são os volumes de madeira com casca encontrados em um inventário na mata atlântica em Blumenau, SC. As unidades eram conglomerados quadrados com 300 metros de lado (9 ha cada) constituídos por quatro subunidades, distribuídas em cruz grega, com 20 metros de largura por 100 metros de comprimento cada (2000m²), afastadas 50 metros do centro do conglomerado.

O PROC MEANS é utilizado para calcular as médias por conglomerado e o PROC NESTED é usado para realizar a análise da variância. No PROC NESTED é necessário somente informar as classes em que os grupos são classificados, do de maior ordem (unidade maior) para o de menor ordem, e a variável dependente a analisar com a instrução VAR. O nível mais baixo, que neste exemplo é a classificação de parcela, não é informado instrução CLASS, somente os níveis mais altos, que neste caso é somente o nível de conglomerado.

Exemplo 9.3 – Amostragem em conglomerados

```

OPTIONS LS=80 PS=58 NODATE NOSTIMER;
TITLE;
DATA ARQUIVO;
INPUT CONGL PARC Y @@;
CARDS;
1 1 091.10 4 1 300.17 7 1 095.59
1 2 109.15 4 2 072.96 7 2 075.69
1 3 063.05 4 3 148.23 7 3 162.50
1 4 080.75 4 4 107.47 7 4 143.98
2 1 090.07 5 1 160.80 8 1 054.10
2 2 074.31 5 2 127.36 8 2 125.43
2 3 191.88 5 3 101.37 8 3 098.74
2 4 044.45 5 4 124.32 8 4 144.87
3 1 079.49 6 1 060.33 9 1 210.27
3 2 079.46 6 2 136.09 9 2 126.68
3 3 144.95 6 3 117.50 9 3 185.08
3 4 071.39 6 4 079.83 9 4 072.57
;
PROC SORT DATA=ARQUIVO; BY CONGL PARC;
TITLE1 'INVENTARIO FLORESTAL';

```

```

TITLE2 'MEDIAS POR CONGLOMERADO';
PROC MEANS DATA=ARQUIVO;
CLASS CONGL;
VAR Y;
TITLE2 'AMOSTRAGEM EM CONGLOMERADOS';
PROC NESTED DATA=ARQUIVO;
CLASS CONGL;
VAR Y;
RUN;
    
```

Como resultado do processamento tem-se o relatório a seguir:

INVENTARIO FLORESTAL
 AMOSTRAGEM EM CONGLOMERADOS
 The MEANS Procedure
 Analysis Variable : Y

CONGL	N Obs	N	Mean	Std Dev	Minimum	Maximum
1	4	4	86.0125000	19.2890027	63.0500000	109.1500000
2	4	4	100.1775000	63.9952912	44.4500000	191.8800000
3	4	4	93.8225000	34.2974259	71.3900000	144.9500000
4	4	4	157.2075000	100.1504403	72.9600000	300.1700000
5	4	4	128.4625000	24.4819204	101.3700000	160.8000000
6	4	4	98.4375000	34.5422942	60.3300000	136.0900000
7	4	4	119.4400000	40.5761022	75.6900000	162.5000000
8	4	4	105.7850000	39.3045142	54.1000000	144.8700000
9	4	4	148.6500000	61.6306650	72.5700000	210.2700000

Continua...

INVENTARIO FLORESTAL
 AMOSTRAGEM EM CONGLOMERADOS
 The NESTED Procedure

Coefficients of Expected Mean Squares		
Source	CONGL	Error
CONGL	4	1
Error	0	1

Nested Random Effects Analysis of Variance for Variable Y

Variance Source	DF	Sum of Squares	F Value	Pr > F	Error Term	Mean Square	Variance Component	Percent of Total
Total	35	93294				2665.5395	2717.3361	100.000
CONGL	8	19926	0.92	0.5180	Error	2490.7260	-56.6525	0.000
Error	27	73368				2717.3361	2717.3361	100.000

Y Mean	115.33277778
Standard Error of Y Mean	8.31786247

Capítulo 24 : Aplicações em experimentação florestal

A análise de experimentos florestais é extremamente facilitada pelo uso do SAS System.

Há vários procedimentos úteis para a análise de experimentos, entre os principais estão o PROC ANOVA e PROC GLM; os dois são muito semelhantes, diferenciando-se principalmente pelo primeiro ser específico para dados balanceados, enquanto o PROC GLM pode ser usado para dados não balanceados, ser mais flexível nas análises e realizar maior número de cálculos e funções.

Embora o PROC GLM seja mais flexível e apresentar um número maior de opções, por essa mesma característica existe maior risco de o procedimento emitir relatórios equivocados quando o usuário não está perfeitamente familiarizado com as estatísticas que está procurando determinar e com os cálculos realizados pelo SAS System.

Portanto, quando se possui dados de um experimento implantado em um delineamento bem conhecido e sem apresentar parcelas perdidas ou outra forma de desbalanceamento dos dados que possam comprometer a análise, é preferível utilizar o PROC ANOVA.

Para quem possui um bom conhecimento de estatística e familiaridade com o PROC GLM, este se torna um poderoso instrumento para a análise de experimentos em qualquer tipo de delineamento experimental, exceto em Látice, para o qual o sistema oferece um procedimento específico chamado de PROC LATTICE.

24.1 MODELOS MATEMÁTICOS

Os tipos de delineamentos experimentais mais comuns, seus modelos matemáticos e a transcrição do modelo para a linguagem SAS são apresentados na Tabela 8.

Na sequência são apresentados alguns modelos de programas para análise de experimentos utilizando as seguintes convenções: Y=variável dependente; B=blocos; R=repetições; A=tratamento a; C=tratamento c; D=tratamento d; AC=interação entre a e c; AD=interação entre a e d; CD=interação entre c e d; ACD=interação entre a, c e d; TB=interação entre tratamentos e blocos; I=idade.

TABELA 8 : Modelos matemáticos de delineamentos experimentais e transcrição para a linguagem SAS

Delineamento	Modelo Matemático	Modelo transcrito para a linguagem SAS
Inteiramente casualizado	$Y_{ij} = m + t_i + e_{ij}$	MODEL y = t;
Blocos casualizados	$Y_{bij} = m + t_i + b_j + e_{ij}$	MODEL y = t b;
Quadrado latino	$Y(i)j k = m + t(i)j k + C_j + F_k + e(i)j k$	MODEL y = t c f;
Blocos casualizados com parcelas subdivididas	$Y_{i j k} = m + b_k + a_i + (ab)_{i k} + d_j + (ad)_{i j} + e_{i j k}$	ba=b*a; ad=a*d; MODEL y= b a ba d ad;
Bifatorial inteiramente casualizado	$Y_{i j} = m + a_i + d_j + (ad)_{i j} + e_{i j k}$	ad=a*d; MODEL y = a d ad;
Bifatorial em blocos casualizados	$Y_{i j k} = m + a_i + d_j + (ad)_{i j} + b_k + e_{i j k}$	ad=a*d; MODEL y = a d ad b;
Bifatorial em blocos casualizados com parcelas subdivididas	$Y_{i j k} = m + b_k + a_i + (ba)_{k j} + d_i + (ad)_{i k} + e_{i j k}$	ba=b*a; ad=a*d; MODEL y = b a ba d ad;
Fatorial 3 ³ em blocos casualizados	$Y_{i j k l} = m + b_l + a_i + c_j + d_k + (ac)_{i j} + (ad)_{i k} + (cd)_{j k} + (acd)_{j k l} + (tb)_{(i j k)l} + e_{i j k l}$	* t=todas as combinações de a-c-d; * erro=b*t; ac=a*c; ad=a*d; cd=c*d; bt=b*t; MODEL y = b a c d ac ad cd acd bt;

Em que: Y=variável dependente; m=média; b=blocos ou repetições; a=tratamento a; c=tratamento c; d=tratamento d; ac=interação entre a e c; ad=interação entre a e d; cd=interação entre c e d; acd=interação entre a, c e d; tb=interação entre tratamentos e blocos; e=erro experimental; i j k l = índices.

24.2 DELINEAMENTO INTEIRAMENTE CASUALIZADO

Exemplo – Delineamento inteiramente casualizado, extraído de Stork e Lopes (1998), página 24.

```

OPTIONS LS=80 PS=58 NODATE NOSTIMER;
TITLE;
DATA ARQUIVO;
DO T=1 TO 6; * TRATAMENTOS;
DO R=1 TO 4; * REPETIÇÕES;
INPUT Y @;
OUTPUT;
END;
END;
DATALINES; * LINHAS=TRATAMENTOS E
COLUNAS=REPETICOES;
50 50 38 38
44 44 44 47
38 47 47 47
49 49 47 47
43 43 43 40
29 35 35 39
;
PROC SORT DATA=ARQUIVO;
BY T;

```

```

PROC GLM DATA=ARQUIVO;          21
CLASS T;                          22
MODEL Y=T;                        23
MEANS T / LINES TUKEY;           24
RUN;                               25
    
```

O resultado do processamento é o relatório a seguir:

The GLM Procedure

Class Level Information		
Class	Levels	Values
T	6	1 2 3 4 5 6

Number of observations	24
------------------------	----

The GLM Procedure
Dependent Variable: Y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	419.7083333	83.9416667	5.53	0.0029
Error	18	273.2500000	15.1805556		
Corrected Total	23	692.9583333			

R-Square	Coeff Var	Root MSE	Y Mean
0.605676	9.052213	3.896223	43.04167

Source	DF	Type I SS	Mean Square	F Value	Pr > F
T	5	419.7083333	83.9416667	5.53	0.0029

Source	DF	Type III SS	Mean Square	F Value	Pr > F
T	5	419.7083333	83.9416667	5.53	0.0029

The GLM Procedure

Tukey's Studentized Range (HSD) Test for Y

NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error rate than REGWQ.

Alpha	0.05
Error Degrees of Freedom	18
Error Mean Square	15.18056
Critical Value of Studentized Range	4.49442
Minimum Significant Difference	8.7556

Means with the same letter are not significantly different.				
Tukey Grouping		Mean	N	T
	A	48.000	4	4
	A	44.750	4	3
	A	44.750	4	2
	A	44.000	4	1
B	A	42.250	4	5
B		34.500	4	6

24.3 DELINEAMENTO EM BLOCOS CASUALIZADOS

Exemplo 11.2 – Delineamento em blocos casualizados.

```

OPTIONS LS=80 PS=58 NODATE NOSTIMER;
TITLE;
DATA ARQUIVO;
INPUT T @;
DO B=1 TO 4;
  INPUT Y 5.1 @;
  OUTPUT;
END;
DATALINES;
00 20.1 19.8 19.8 20.3
40 19.5 20.3 19.2 19.4
60 18.8 18.3 18.6 18.0
80 17.4 17.8 17.2 16.9
;
PROC SORT DATA=ARQUIVO;
  BY B T;
PROC GLM DATA=ARQUIVO;
  CLASS B T;
  MODEL Y=B T;
  MEANS B T / LINES TUKEY;
RUN;
    
```

O resultado do processamento é o relatório a seguir:

The GLM Procedure

Class Level Information		
Class	Levels	Values
B	4	1 2 3 4
T	4	0 40 60 80

Number of observations	16
------------------------	----

The GLM Procedure
Dependent Variable: Y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	18.0100000	3.00166667	22.01	<.0001
Error	9	1.22750000	0.13638889		
Corrected Total	15	19.23750000			

R-Square	Coeff Var	Root MSE	Y Mean
0.936192	1.960497	0.369309	18.83750

Source	DF	Type I SS	Mean Square	F Value	Pr > F
B	3	0.44750000	0.14916667	1.09	0.4006
T	3	17.56250000	5.85416667	42.92	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
B	3	0.44750000	0.14916667	1.09	0.4006
T	3	17.56250000	5.85416667	42.92	<.0001

The GLM Procedure
Tukey's Studentized Range (HSD) Test for Y
NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error rate than REGWQ.

Alpha	0.05
Error Degrees of Freedom	9
Error Mean Square	0.136389
Critical Value of Studentized Range	4.41490
Minimum Significant Difference	0.8152

Means with the same letter are not significantly different.			
Tukey Grouping	Mean	N	B
A	19.0500	4	2
A	18.9500	4	1
A	18.7000	4	3
A	18.6500	4	4

The GLM Procedure
 Tukey's Studentized Range (HSD) Test for Y

NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error rate than REGWQ.

Alpha	0.05
Error Degrees of Freedom	9
Error Mean Square	0.136389
Critical Value of Studentized Range	4.41490
Minimum Significant Difference	0.8152

Means with the same letter are not significantly different.			
Tukey Grouping	Mean	N	T
A	20.0000	4	0
A	19.6000	4	40
B	18.4250	4	60
C	17.3250	4	80

24.4 DELINEAMENTO EM QUADRADO LATINO

Exemplo 11.3 – Delineamento em quadrado latino – extraído de Gomes (1982) – página 119.

```

OPTIONS LS=80 PS=58 NODATE NOSTIMER;
TITLE;
DATA ARQUIVO;
    INPUT linha coluna t y @@;
datalines;
1 1 4 432 1 2 1 518 1 3 2 458 1 4 3 583 1 5 5 331
2 1 3 724 2 2 5 478 2 3 1 524 2 4 2 550 2 5 4 400
3 1 5 489 3 2 2 384 3 3 3 556 3 4 4 297 3 5 1 420
4 1 2 494 4 2 4 500 4 3 5 313 4 4 1 486 4 5 3 501
5 1 1 515 5 2 3 660 5 3 4 438 5 4 5 394 5 5 2 318
;
proc sort data=arquivo;
    by t colunas linhas;
MODEL y = t colunas linhas / ss1;
    MEANS T / LINES TUKEY;
RUN;
    
```

O resultado do processamento é o seguinte relatório:

The GLM Procedure

Class Level Information		
Class	Levels	Values
T	5	1 2 3 4 5
COLUNA	5	1 2 3 4 5
LINHA	5	1 2 3 4 5

Number of observations	25
------------------------	----

The GLM Procedure
Dependent Variable: Y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	12	223609.5200	18634.1267	6.55	0.0014
Error	12	34114.7200	2842.8933		
Corrected Total	24	257724.2400			

R-Square	Coeff Var	Root MSE	Y Mean
0.867631	11.33189	53.31879	470.5200

Source	DF	Type I SS	Mean Square	F Value	Pr > F
T	4	137488.2400	34372.0600	12.09	0.0004
COLUNA	4	55640.6400	13910.1600	4.89	0.0142
LINHA	4	30480.6400	7620.1600	2.68	0.0831

The GLM Procedure
Tukey's Studentized Range (HSD) Test for Y

NOTE: This test controls the Type I experimentwise error rate, but it generally has a higher Type II error rate than REGWQ.

Alpha	0.05
Error Degrees of Freedom	12
Error Mean Square	2842.893
Critical Value of Studentized Range	4.50760
Minimum Significant Difference	107.48

Means with the same letter are not significantly different.			
Tukey Grouping	Mean	N	T
A	604.80	5	3
B	492.60	5	1
B	440.80	5	2
B	413.40	5	4
B	401.00	5	5

Capítulo 25 : Gráficos no SAS University Edition (ODS Graphics)

25.1 INTRODUÇÃO

No SAS University Edition existem os procedimentos para gráficos ODS, que têm uma sintaxe diferente dos procedimentos existentes no SAS tradicional. Para construir gráficos no SAS University Edition é aconselhável usar o modo interativo em “Tasks and Utilities”, ou consultar o manual “Basic ODS Graphics – Examples”, disponível em <http://support.sas.com/documentation/prod-p/grstat/9.4/en/PDF/odsbasicg.pdf>; veja o exemplo na secção “2.4.4.2.1 CONSTRUINDO UM GRÁFICO DE DISPERSÃO”.

Sintaxe:

```
■ ODS GRAPHICS <OFF | ON> </ OPTIONS> ;
```

Antes de criar gráficos ODS, eles devem ser habilitados antes do procedimento onde serão utilizados, com a expressão:

ODS graphics on;

Após o procedimento onde se criou os gráficos desejados é preciso desabilitar usando a expressão:

ODS graphics off;

Opções do procedimento:

ANTIALIAS = ON | OFF - Controla o uso de antialiasing para alisar os componentes de um gráfico.

ANTIALIASMAX = n - Especifica o número máximo de marcadores ou linhas a serem anti-polarizados antes que o anti-alinhamento seja desabilitado.

BORDER = ON | OFF - Especifica se deseja desenhar o gráfico com uma borda.

HEIGHT = *dimensão* - Especifica a altura do gráfico. O padrão é HEIGHT = 480PX (480 pixels).

IMAGEFMT = <image-file-type | ESTÁTICA> - Especifica o formato da imagem para gráficos. Por padrão, IMAGEFMT = STATIC e ODS usa dinamicamente o formato de imagem estática de melhor qualidade para o destino de saída ativo.

IMAGEMAP = ON | OFF - Controla a geração de dicas de ferramentas no destino HTML. O padrão é IMAGEMAP = OFF.

IMAGENAME = <base-file-name> - Especifica o nome do arquivo da imagem base. O padrão é o nome do objeto de saída.

LABELMAX = n - Especifica o número máximo de áreas rotuladas antes da rotulação estar desabilitada.

MAXLEGENDAREA = n - Especifica a porcentagem máxima da área de gráfico geral que uma legenda pode ocupar. O padrão é MAXLEGENDAREA = 20.

RESET <= opção> - Redefine uma ou mais opções GRAPHICAS ODS para suas configurações padrão. As opções RESET e RESET = ALL são equivalentes. Se você quiser redefinir mais de uma opção, mas não todas as opções, então você deve especificar RESET = separadamente para cada opção que você redefinir (por exemplo, ods graphics on / reset = antialias reset = index;). As opções RESET = incluem o seguinte: TODOS, ANTIALIAS, ANTIALIASMAX, BORDER, ÍNDICE, ALTURA, IMAGEMAP, LABELMAX, SCALE, TIPMAX, WIDTH.

SCALE=ON | OFF - Especifica se as fontes e os marcadores de símbolos são dimensionados proporcionalmente com o tamanho do gráfico. O padrão é SCALE = ON.

TIPMAX = n - Especifica o número máximo de dicas de ferramentas distintas permitidas antes que as dicas de ferramentas estejam desabilitadas. O padrão é TIPMAX = 500.

WIDTH = dimension -

Especifica a largura do gráfico. O padrão é WIDTH = 640PX (640 pixels). Você também pode especificar larguras em polegadas (por exemplo, WIDTH = 5in) ou centímetros (por exemplo, WIDTH = 12cm).

25.1.1 DECLARAÇÕES DE DESTINO ODS

ODS tem uma série de declarações que controlam o destino da saída ODS, como: ODS DOCUMENT, ODS HTML, ODS LATEX, ODS LISTING, ODS PCL, ODS PDF, ODS PS e ODS RTF. Especificar uma declaração abre um destino, a menos que a opção CLOSE seja especificada. Cada uma das seguintes instruções abre um destino ODS:

- ods html;
- ods rtf;
- ods html image_dpi=300;
- ods listing style=Statistical;

Cada uma das seguintes declarações encerra um destino ODS:

- ods html close;
- ods rtf close;
- ods listing close;

Existem duas opções que são comumente usadas nas instruções de destino ODS para controlar aspectos de ODS Graphics:

- IMAGE_DPI = dpi - Especifica os pontos por polegada (DPI). O padrão varia dependendo do destino. Por exemplo, o padrão é 100 para HTML e 200 para RTF.
- STYLE = nome do estilo - Especifica um estilo. Os estilos comumente usados incluem DEFAULT, LISTING, STATISTICAL, JOURNAL, JOURNAL2, RTF e ANALYSIS.

25.1.2 PLOTS = OPÇÃO.

Cada procedimento estatístico que produz ODS Graphics tem uma opção PLOTS = que é usada para selecionar gráficos e especificar algumas opções. A sintaxe da opção PLOTS = é a seguinte:

```
PLOTS <(GLOBAL-PLOT-OPTIONS)> <= PLOT-REQUEST <(OPÇÕES) >>  
PLOTS <(GLOBAL-PLOT-OPTIONS)> <= (PLOT-REQUEST <(OPTIONS)>  
<... PLOT-REQUEST <(OPTIONS) >>>
```

A opção PLOTS = tem uma sintaxe geral comum para todos os procedimentos estatísticos, mas as opções de plotagem global específicas, solicitações de plotagem e opções de plotagem variam de acordo com os procedimentos. Esta seção discute apenas algumas das opções disponíveis na opção PLOTS =. Para obter mais informações sobre a opção PLOTS =, consulte a seção "Sintaxe" para cada procedimento que produz o ODS Graphics. Há apenas um número limitado de coisas

possíveis de controlar com a opção PLOTS =. A maioria dos detalhes gráficos são controlados por modelos de gráfico ou por estilos.

Esta opção está disponível no SAS tradicional, mas não está no SAS Studio.

A opção PLOTS = geralmente aparece na instrução PROC. No entanto, para alguns procedimentos, certas análises e, portanto, certos gráficos podem aparecer somente se for especificada uma declaração adicional. Esses procedimentos geralmente possuem uma opção PLOTS = nessa outra declaração. Por exemplo, o procedimento PHREG possui uma opção PLOTS = na instrução BAYES, que é usada para realizar uma análise bayesiana. Consulte a seção "Sintaxe" de cada capítulo do procedimento para obter mais informações. Os exemplos a seguir ilustram a sintaxe da opção PLOTS =:

- plots=all
- plots=none
- plots=residuals
- plots=residuals(smooth)
- plots=(trace autocorr)
- plots(unpack)
- plots(unpack)=diagnostics
- plots=diagnostics(unpack)
- plots(only)=freqplot
- plots=(scree(unpack) loadings(plotref) preloadings(flip))
- plots(unpack maxparmlabel=0 stepaxis=number)=coefficients
- plots(sigonly)=(rawprob adjusted(unpack))

25.1.3 PROCEDIMENTOS DA LINGUAGEM QUE SUPORTAM GRÁFICOS ODS

OS procedimentos estatísticos que suportam ODS Graphics têm uma opção PLOTS= que pode ser usada para selecionar gráficos e especificar opções.

Sintaxe

A sintaxe geral para isso inclui a habilitação de gráficos ODS e tem a seguinte forma:

```

ODS GRAPHICS ON;
PROC NOMEDOPROCEDIMENTO <OPÇÕESDOPROCEDIMENTO>
PLOTS=<OPÇÕES>;
DECLARAÇÕESDOPROCEDIMENTO;
ODS GRAPHICS OFF;
    
```

Os procedimentos do SAS em que é possível criar gráficos ODS são listados na Tabela 9, por módulo do SAS. Dos módulos listados, somente Base e STAT estão disponíveis no SAS University Edition. As instruções ODS no ambiente SAS Studio podem desativar alguns recursos de saída. O comando “PLOTS=” pode não funcionar no SAS Studio.

TABELA 9 – Procedimentos que suportam gráficos ODS, por módulo do SAS

SAS/STAT		SAS/QC	SAS/ETS	Base
ANOVA	MI	ANOM	ARIMA	CORR
BOXPLOT	MIXED	CAPABILITY	AUTOREG	FREQ
CALIS	MULTTEST	CUSUM	ENTROPY	UNIVARIATE
CLUSTER	NPAR1WAY	MACONTROL	ESM	
CORRESP	PHREG	PARETO	EXPAND	
FACTOR	PLS	RELIABILITY	MODEL	
FREQ	PRINCOMP	SHEWHART	PANEL	
GAM	PRINQUAL		SIMILARITY	
GENMOD	PROBIT		SYSLIN	
GLIMMIX	QUANTREG		TIMESERIES	
GLM	REG		UCM	
GLMSELECT	ROBUSTREG		VARMAX	
KDE	RSREG		X12	
KRIGE2D	SEQDESIGN			
LIFEREG	SEQTEST			
LIFETEST	SIM2D			
LOESS	TCALIS			
LOGISTIC	TRANSREG			
MCMC	TTEST			
MDS	VARIogram			

O “Exemplo – PROC NLIN com gráficos ODS” demonstra o uso deste tipo de gráficos com o procedimento NLIN, a seguir.

```
* EXEMPLO - PROC NLIN com gráficos ODS;
data A;
input idade h @@;
Y=h;
X=idade;
cards;
05 04 05 04 12 18 14 22 07 07 07 06 25 28 25 26 03 01
10 12 08 10 15 23 16 25 20 27 22 27 30 30 27 29 03 02
;
ODS Graphics on;
```

```

PROC NLIN DATA=A BEST=1 PLOTS=Residuals;
PARAMETERS A=20 TO 30 BY 5.0 B=0.2 C=0.1 TO 9.9 BY 1.0;
BOUNDS 19<A<31, 0<B<1, 0<C<10;
EBX = EXP(-B*X);
EBX1 = 1 - EBX;
EBXC = (EBX1)**C;
MODEL Y = A*EBXC;
DER.A = EBXC;
DER.B = A*X*C*EBX*EBX1**(C-1);
DER.C = A*EBXC*LOG(EBX1);
ODS Graphics OFF;
RUN;

```

O resultado do processamento é o que segue.

25.2 PROCEDIMENTOS PARA CONSTRUIR GRÁFICOS ODS

Os gráficos ODS (Output Delivery System) incluem os procedimentos:

- SGSCATTER - Cria gráficos de dispersão de células únicas e células múltiplas e matrizes de dispersão com ajustes opcionais e elipses.
- SGPLOT - Cria gráficos de uma única célula com uma variedade de gráficos.
- SGPANEL - Cria painéis de páginas únicas ou múltiplas páginas de gráficos e gráficos condicionados à variáveis de classificação.
- SGRENDER – Serve para criar gráficos a partir de modelos de gráficos modificados ou criados pelo usuário (não faz parte deste manual).

25.2.1 PROC SGPLOT

Sintaxe:

```

PROC SGPLOT <OPTIONS>;
BAND X=X-VAR UPPER=U-VAR LOWER=L-VAR /
<OPTIONS>;
BLOCK X=X-VAR BLOCK=BLOCK-VAR / <OPTIONS>;
BUBBLE Y=Y-VAR X=X-VAR BLOCK=B-VAR / <OPTIONS>;
DENSITY VAR / <TYPE=TYPE> <OPTIONS>;
DOT VAR / <RESPONSE=R-VAR> <STAT=STAT>
<OPTIONS>;
DROPLINE Y=Y-VAR | X=X-VAR / <OPTIONS>;
ELLIPSE Y=Y-VAR X=X-VAR / <OPTIONS>;
FRINGE VAR / <OPTIONS>;
GRADLEGEND <NAME> / <OPTIONS>;
[HV]BAR VAR / <RESPONSE=R-VAR> <STAT=STAT>
<OPTIONS>;
[HV]BARBASIC VAR / <OPTIONS>;
[HV]BARPARM <CATEGORY=C-VAR>

```

```

<RESPONSE=R-VAR> / <OPTIONS>;
[HV]BOX VAR / <CATEGORY=C-VAR> <OPTIONS>;
HEATMAP Y=Y-VAR X=X-VAR / <OPTIONS>;
HEATMAPPARM Y=Y-VAR X=X-VAR
COLORGROUP=C-VAR / <OPTIONS>;
HIGHLOW Y=Y-VAR X=X-VAR
HIGH=H-VAR LOW=L-VAR / <OPTIONS>;
HISTOGRAM VAR / <OPTIONS>;
[HV]LINE VAR / <RESPONSE=R-VAR> <STAT=STAT>
<OPTIONS>;
INSET INSET-TEXT / <OPTIONS> ;
KEYLEGEND NAMES / <OPTIONS>;
LOESS Y=Y-VAR X=X-VAR / <OPTIONS>;
NEEDLE Y=Y-VAR X=X-VAR / <OPTIONS>;
PBSPLINE Y=Y-VAR X=X-VAR / <OPTIONS>;
REFLINE VALUE / AXIS=X|Y <OPTIONS>;
REG Y=Y-VAR X=X-VAR / <GROUP=G-VAR>
<DEGREE=D> <OPTIONS>;
SCATTER Y=Y-VAR X=X-VAR / <OPTIONS>;
SERIES Y=Y-VAR X=X-VAR / <OPTIONS>;
STEP Y=Y-VAR X=X-VAR / <OPTIONS>;
STYLEATTRS OPTIONS;
SYMBOLCHAR NAME=IDENTIFIER
CHAR='HEX-STRING'X | KEYWORD / <OPTIONS>;
SYMBOLIMAGE NAME=IDENTIFIER
IMAGE='FILE' / <OPTIONS>;
TEXT Y=Y-VAR X=X-VAR TEXT=T-VAR / <OPTIONS>;
VECTOR Y=Y-VAR X=X-VAR / <OPTIONS>;
WATERFALL CATEGORY=C-VAR
RESPONSE=R-VAR / <OPTIONS>;
[XY]AXIS<2> OPTIONS;
RUN;

```

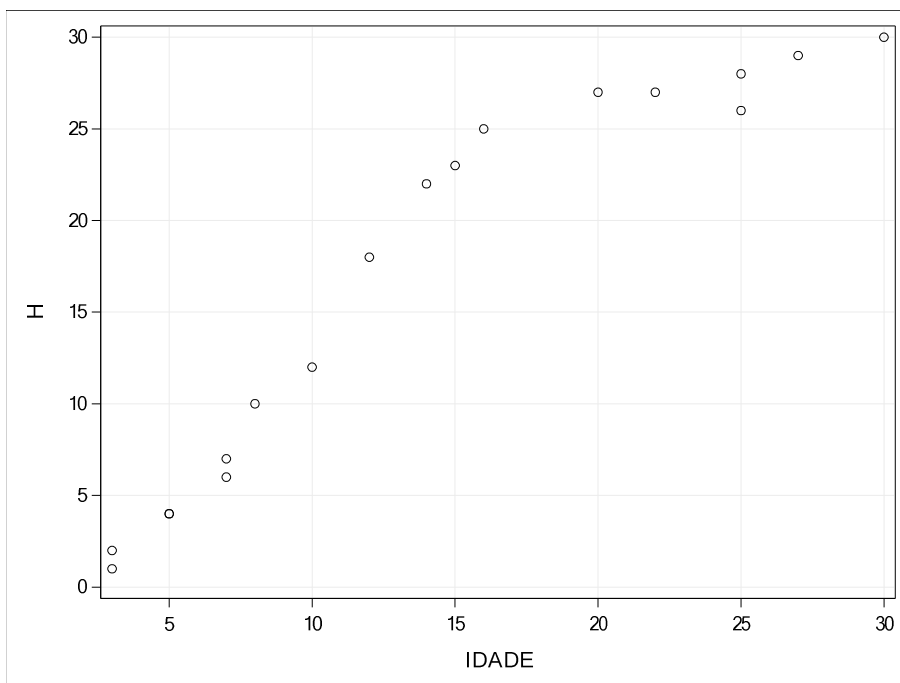
O “Exemplo – Gráfico de dispersão com SGPLOT” mostra como construir este tipo de gráfico.

```

* EXEMPLO - Gráfico de dispersão com SGPLOT;
data A;
input idade h @@;
DATALINES;
05 04 05 04 12 18 14 22 07 07 07 06 25 28 25 26 03 01
10 12 08 10 15 23 16 25 20 27 22 27 30 30 27 29 03 02
;
run;
ods graphics / reset imagemap;
proc sgplot data=WORK.A;
scatter x=IDADE y=H / transparency=0.0 name='Scatter';
xaxis grid;
yaxis grid;
run;
ods graphics / reset;

```

Como resultado tem-se o gráfico a seguir.



25.2.2 PROC SGSCATTER

Sintaxe:

```
PROC SGSCATTER <OPTIONS>;
COMPARE Y=Y-VAR-LIST X=X-VAR-LIST / <OPTIONS>;
MATRIX VAR-LIST / <OPTIONS>;
PLOT Y-VAR * X-VAR / <OPTIONS>;
RUN;
```

O uso e exemplos deste procedimento são descritos no manual “SAS GRAPH Statistical Graphics Procedures Guide”, disponível em <http://support.sas.com/documentation/cdl/en/grstatproc/62603/HTML/default/viewer.htm#titlepage.htm> .

25.2.3 PROC SGPANEL

Sintaxe:

```
PROC SGPANEL <OPTIONS>;
PANELBY VAR-LIST / <OPTIONS>;
```

**MANY-PROC-SGPLOT-STATEMENTS
RUN;**

O uso e exemplos deste procedimento são descritos no manual “SAS GRAPH: Statistical Graphics Procedures Guide”, disponível em <http://support.sas.com/documentation/cdl/en/grstatproc/62603/HTML/default/viewer.htm#titlepage.htm> >.

Referências

ANJOS, Adilson dos. **Planejamento de experimentos I - teste de Shapiro-Wilk para normalidade**. [Curitiba]: UFPR, Departamento de Estatística, 2003. Disponível em: <<http://www.est.ufpr.br/planexp/planexp.pdf>>. Acesso em: 15/maio/2004.

IU. **Testing normality using SAS, STATA, and SPSS**. Indiana: Indiana University – IU, UITS - Center for Statistical and Mathematical Computing, p 94-73. Disponível em: <<http://www.indiana.edu/statmath>>. Acesso em: 15/maio/2004. 26p.

SAS. **SAS Product Documentation**. Disponível em: <<https://support.sas.com/documentation/>>. Acesso em: 10/07/2017.

SAS. **SAS University Edition**. Disponível em: < https://www.sas.com/en_us/software/university-edition.html>. Acesso em: 10/07/2017.

SAS Institute. **SAS language reference**, v.8. Cary, NC: SAS Institute, 1999. 1242p.

_____. **SAS language reference: Concepts**, v.8. Cary, NC: SAS Institute, 1999. 554p.

_____. **SAS procedures guide**, v.8. Cary, NC: SAS Institute, 1999. 1729p.

_____. **SAS/graph user's guide**, v.8. Cary, NC: SAS Institute, 1999. 1220p.

_____. **SAS/STAT user's guide**, v.8. Cary, NC: SAS Institute, 1999. 3365p.

Apêndice 1 – Planilha para coleta de dados de cubagem de árvores abatidas

Identificação do local da amostra (Arquivar em: identif.dat)									
Município:									
Local:									
Distrito Florestal:									
Unidade Florestal:									
Talhão:									
Identificação da árvore e dimensões (Arquivar em: arvores.dat)									
Árvore n°:	Altura total (m)	Altura comercial (M)	DAP cc (cm)	Casca (cm)					
Medições das secções da árvore (Arquivar em: medicoes.dat)									
Secção	Árvore n°:	h(i) (m)	Dcc(i) (cm)	Casca (i) (cm)	Secção	Árvore n°:	h(i) (m)	Dcc(i) (cm)	Casca (i) (cm)
1					26				
2					27				
3					28				
4					29				
5					30				
6					31				
7					32				
8					33				
9					34				
10					35				
11					36				
12					37				
13					38				
14					39				
15					40				
16					41				
17					42				
18					43				
19					44				
20					45				
21					46				
22					47				
23					48				
24					49				
25					50				

Obs: Casca refere-se à dupla espessura de casca.