

CURSO DE INTRODUCCIÓN A LA PROGRAMACIÓN SAS[®] v8

Versión de manual 1.2

**Universidad Complutense de Madrid
Servicios Informático de Apoyo a Docencia e Investigación
Apoyo a Investigación
(Revisado febrero 2004)**

Introducción

Esta guía contiene un pequeño manual de SAS® V8 y supone una actualización de la versión anterior.

En general, esta guía está muy orientada a la sintaxis de programación de SAS®.

En la versión v8 de SAS existen utilidades que pueden realizarse con la interfaz (entorno gráfico) de usuario y que se presentan en esta guía. De todas ellas, siempre que sea posible, indicaremos cómo generar el código SAS® correspondiente. La obtención del código SAS® permite almacenarlo en un fichero y poder ejecutarlo de nuevo en otra sesión SAS.

Las utilidades de entorno gráfico que permite SAS son:

- Creación y manipulación de librerías de datos SAS (que llamaremos SAS-FILES)

- Importación/exportación de datos externos en ficheros ASCII

- Importación/exportación de datos externos en ficheros en otros formatos, por ejemplo, EXCEL.

- Exportación de gráficos a otros formatos

En el apéndice II, se detallan las novedades y utilidades de entorno gráfico que incorpora SAS v8. Las utilidades más importantes se han explicado en la parte de documento previa a los apéndices.

Dada la extensión del documento en relación con la duración del curso de introducción que se imparte, algunas utilidades u opciones más avanzadas de versiones previas de este documento se han expresado mediante notas a pie de página o mediante la indicación de AVANZADO. Por este motivo, las notas a pie de página no avanzadas se señalan con negrita.

INTRODUCCIÓN AL PAQUETE ESTADÍSTICO SAS®

INTRODUCCIÓN	2
I. GENERALIDADES DEL SISTEMA SAS®	6
I.1 VISIÓN GENERAL DEL SISTEMA SAS®	6
I.2 CLASIFICACIÓN DE LOS PROGRAMAS DE ESTADÍSTICA DE LOS MODULOS SAS/BASE® Y SAS/STAT®	6
I.3 ESTRUCTURA GENERAL DE UN PROGRAMA SAS®	9
Ejecución de SAS	9
En WINDOWS	9
Estructura de un programa SAS	10
Interfaz de usuario	12
La ejecución de SAS en el entorno de ventanas	14
I.4 NORMAS GENERALES DE SINTAXIS	14
II. ENTRADA DE DATOS EN SAS	16
II.1 CREACION DE CONJUNTOS DE DATOS SAS® (SAS-FILES).	16
II.2 LIBRERÍAS DE DATOS SAS®	17
Sentencia LIBNAME	17
Creación de librerías de datos SAS con el entorno gráfico de SAS V8	18
II.A. ENTRADA DE DATOS Y DECLARACION DE VARIABLES en el PASO DATA	19
SENTENCIA DATA.	20
SENTENCIA INFILE.	21
SENTENCIA INPUT.	23
SENTENCIA CARDS.	26
ESPECIFICACIÓN DE VALORES DE NO OBSERVACIÓN EN LOS DATOS	26
II.B. ENTRADA DE DATOS ASCII con la INTEFAZ GRÁFICA de SAS	26
II.C. IMPORTACIÓN (EXPORTACIÓN) DE DATOS EN OTROS FORMATOS	33
II. D. INTRODUCCIÓN de DATOS en SAS® con la INTERFAZ GRÁFICA de SAS V8	36
III. PASO DATA. UTILIDADES.	38
III.1 TRANSFORMACIÓN O CREACIÓN DE NUEVAS VARIABLES	38
III.2 UTILIDADES EN EL PASO DATA	39
Renombrar variables.	39
Selección de variables en un SAS-FILE.	39
Selección de observaciones en un SAS-FILE.	40
Creación de diferentes SAS-FILES en un mismo paso DATA.	41
Creación de más de una observación por línea de datos.	41
Codificación de variables.	41

III.3.A. (AVANZADO) CREACIÓN DE FICHEROS EXTERNOS DE DATOS O INFORMES	42
SENTENCIA FILE	42
SENTENCIA PUT.	44
III.3.B. EXPORTACIÓN DE DATOS A OTROS FORMATOS CON EL ENTORNO GRÁFICO DE SAS V8	45
III.4 SENTENCIAS LOGICAS Y BUCLES	45
III.5 OTRAS SENTENCIAS USADAS EN EL PASO DATA	48
III.6 ESPECIFICACIÓN DE VARIABLES EN PASOS DATA O PROC.	49
IV. MANIPULACIÓN DE CONJUNTOS DE DATOS SAS®.	50
IV.1 SENTENCIA SET. UTILIDADES	50
IV.2 SENTENCIA MERGE. UTILIDADES	52
IV.3 OPCIONES DE LOS CONJUNTOS DE DATOS SAS®	52
V. SENTENCIAS DE USO GENERAL.	54
V.1 SENTENCIAS USADAS EN PASOS DATA Y PROC	54
V.2 SENTENCIAS DE USO GENERAL	54
VI. UTILIZACIÓN DE PROCEDIMIENTOS SAS®.	56
VI.I. PROCEDIMIENTO FORMAT.	56
VI.II. OTROS PROCEDIMIENTOS	57
VI.III. GRAFICOS EN SAS	60
VII. UTILIDADES DE LA PROGRAMACIÓN SAS® E INTERACCIÓN CON OTROS PAQUETES.	62
VII.1 (AVANZADO) PROGRAMACIÓN CON MACROS	62
VII.2 INTERACCIÓN ENTRE BMDP Y SAS® Y ENTRE SPSS Y SAS®	65
VII.3 EXPORTACIÓN/IMPORTACIÓN DE FICHEROS SAS® A OTRAS PLATAFORMAS	65
APÉNDICE I (PLATÓN. UNIX)	66
Ejecución de un programa SAS®.	66
APÉNDICE II. NOVEDADES Y ENTORNO GRÁFICO DE SAS V8	67
A.II.1. Interfaz de usuario	67
A.II.2. Novedades de la versión SAS V8	67

A.II.3. Novedades estadísticas de la versión SAS V8	68
A.II.4. Documentación de SAS V8	68
A.II.5. Ayuda de SAS® V8	69
A.II.6 Ventana <i>Explorer</i>	70
A.II.7. Creación de Librerías de datos SAS®	72
A.II.8. Interfaz de ayuda para la entrada de datos de ficheros ASCII	72
A.II.9. Introducción de datos en SAS®	72
APÉNDICE III. (AVANZADO) SAS-FILES Y SAS-ENTERPRISE GUIDE	72
Nota Importante	72
SAS-Files creados en SAS®	72
Asignación de librerías en SAS-Enterprise Guide, desde el programa Administrador	73
SAS-Files creados en SAS® -Enterprise Guide	73

I. GENERALIDADES DEL SISTEMA SAS®

I.1 VISIÓN GENERAL DEL SISTEMA SAS®

El sistema SAS proporciona numerosas herramientas para el análisis estadístico de datos y la generación de informes.

Además del SAS BASE y del SAS/STAT, la Universidad Complutense de Madrid cuenta con licencia de los módulos SAS/GRAPH para la realización de gráficos, SAS/ASSIST para la construcción guiada de los procedimientos, SAS/OR de investigación operativa, SAS/IML para cálculos matriciales, SAS/FSP para construcción de paneles de entrada de datos y SAS/EIS (Executive Interactive System), SAS/CONNECT para ejecución de SAS en remoto, SAS MDDDB Server common products.

Otros módulos con licencia sólo en Windows: SAS/AF, SAS/QC, SAS/INSIGHT, SAS PC File Formats.

Existen macros de SAS que pueden encontrarse en <http://www.sas.com/> (mediante cualquier navegador de internet, por ejemplo *netscape*) o que vienen incorporadas con el producto como ficheros de ejemplo de programas SAS. Existe una macro que cabe destacar: **TREEDISC** para aplicación de la técnica de árboles de segmentación. Esta macro ya no es soportada por SAS.

Existe un documento en versión borrador que detalla la relación de SAS con el análisis estadístico de datos. Se puede solicitar a través del servicio de atención al usuario de los Servicios Informáticos (91 3 94 47 74, solicitarlo a Apoyo a Investigación, preguntando por Carmen Bravo)

I.2 CLASIFICACIÓN DE LOS PROGRAMAS DE ESTADÍSTICA DE LOS MODULOS SAS/BASE® Y SAS/STAT®

SAS permite la ejecución de algunos procedimientos no sólo en el entorno interactivo, sino que además esta ejecución sea de hecho interactiva. Es decir, se pueden ejecutar algunos comandos del procedimiento y posteriormente otros comandos del mismo. En la versión 6.12 estos procedimientos eran ANOVA, CATMOD, GLM, PLAN y REG.

Veamos sucintamente los procedimientos disponibles para el análisis de datos en la versión SAS 6.12. La versión SAS 8 incorpora nuevos procedimientos.

Las novedades estadísticas de la versión SAS V8 se encuentran en el apéndice II.

Estadística descriptiva univariante.

MEANS Descripción univariante.

UNIVARIATE Descripción univariante. Diagramas de 'tallo-hoja' y de 'caja'. Percentiles, test de normalidad.

CORR Matrices de correlación entre variables.

CHART	Gráficos sectoriales, en estrella, de barras y por bloques. Visualización gráfica de varias variables simultáneamente.
PLOT	Gráficos bidimensionales.

Regresión y análisis de la varianza y covarianza.

TTEST	Aplicación del test t de Student.
NPARIWAY	Análisis no paramétricos de comparación de grupos.
ANOVA	Análisis de la varianza en diseños balanceados. Proporciona también diversos tests de comparaciones múltiples.
GLM	Análisis de modelos lineales, usando método de mínimos cuadrados. Proporciona análisis de la varianza univariante y multivariante en diseños balanceados o no, con diversas estructuras de diseño; test de comparaciones múltiples, análisis de la covarianza, regresión simple y múltiple, correlación parcial,...
GLMMOD	Front-end para la construcción de modelos a especificar en el procedimiento GLM. Tiene por salida la matriz de diseño.
GENMOD	Modelos lineales generalizados, según la definición de Nelder y Wedderburn.
MULTTEST	Tests de comparaciones múltiples, incluyendo métodos bootstrap y permutación.
NESTED	Análisis de la varianza y covarianza en diseños anidados, especialmente cuando el número de niveles y observaciones es grande.
VARCOMP	Cálculo de componentes de la varianza en un modelo lineal general, especialmente con efectos aleatorios.
PLAN	Construcción de diseños para experimentos anidados y crossed.
MIXED	Modelos lineales mixtos.
REG	Análisis de regresión lineal general. Regresión Ridge y análisis de componentes principales incompleto, método por pasos. No posee tantos diagnósticos como el GLM.
ORTHOREG	Regresión usando el método Gentelman-Givens.
RSREG	Regresión de modelos con una superficie de respuesta cuadrática.
TRANSREG	Obtención de transformaciones lineales o no lineales, usando el método de mínimos cuadrados alternado para ajuste a modelos de regresión lineal, correlación canónica y análisis de la varianza. Incluye transformaciones logarítmicas, rangos, exponenciación y logit.
LATTICE	Análisis de la varianza en diseños Lattice.
NLIN	Regresión en modelos no lineales.
CALIS	Modelos causales. (Path analysis). Análisis de Covarianza de ecuaciones lineales estructurales, en particular, Análisis Factorial Exploratorio y Confirmatorio.

Análisis de datos cualitativos.

FREQ	Construcción de tablas de contingencia. Medidas de asociación. Análisis estratificado. Estimación de riesgo relativo.
CORRESP	Análisis de correspondencias binarias y múltiples.
PRINQUAL	Análisis de componentes principales para datos cualitativos.
CATMOD	Ajuste de modelos lineales a funciones de respuesta frecuentistas. Regresión, análisis de la varianza, modelos lineales, log-lineales, regresión logística, medidas repetidas.

LOGISTIC	Análisis de regresión logística para respuesta binaria u ordinal mediante máxima verosimilitud. Regresión por pasos. La función de respuesta puede sustituirse por las funciones 'normit' o log-log.
PROBIT	Regresión probit, logística y ordinal logística. Variable dependiente dicotómica o policotómica.

Análisis discriminante.

DISCRIM	Análisis discriminante, obteniendo funciones discriminantes lineales y cuadráticas. También, métodos no paramétricos.
CANDISC	Análisis discriminante canónico. Se asume distribución multivariante normal.
STEPDISC	Análisis discriminante por pasos, hacia delante o hacia atrás. Variables continuas.

Análisis cluster.

CLUSTER	Análisis cluster jerárquico de observaciones. Distancia euclídea o matrices de disimilaridades (similaridades).
FASTCLUS	Análisis cluster para observaciones por el método de las k-medias, usando criterio L_p . Se usa con conjuntos grandes de datos, para la determinación inicial de clusters.
VARCLUS	Análisis cluster jerárquico y disjunto de variables, mediante el método de las componentes oblicuas multigrupo.
TREE	Muestra la interpretación visual de un análisis cluster jerárquico, mediante dendogramas. Toma como entrada ficheros de salida de CLUSTER o VARCLUS. Genera fichero en el que se indica la pertenencia a cluster en cualquier nivel del árbol del cluster.
MODECLUS	Análisis cluster disjunto de observaciones. No paramétrico.
ACECLUS	Estimación de matrices de varianzas-covarianzas intra-cluster, sin previo conocimiento de los clusters. Dichas matrices serán la entrada de otros procedimientos de análisis cluster. Util para cluster elípticos alargados.

Análisis de supervivencia.

LIFEREG	Análisis de supervivencia. Métodos paramétricos.
LIFETEST	Análisis de supervivencia. Métodos no paramétricos.
PHREG	Análisis de la regresión de supervivencia, basado en el modelo de azar de Cox.

Scoring.

STANDARD	Standarización de variables.
SCORE	Construcción de nuevas variables que son combinación lineal de otras. Se utiliza junto al procedimiento FACTOR u otros.
RANK	Asignación de rangos.

Otros procedimientos multivariantes.

CANCORR	Análisis de correlación canónica.
PRINCOMP	Análisis de componentes principales.
FACTOR	Análisis factorial, incluyendo rotaciones.

MDS Escalamiento multidimensional.
INBREED Cálculo de la covarianza o coeficientes puros de un pedigree.

Otros.

OUTPUT Y TEMPLATE

I.3 ESTRUCTURA GENERAL DE UN PROGRAMA SAS®

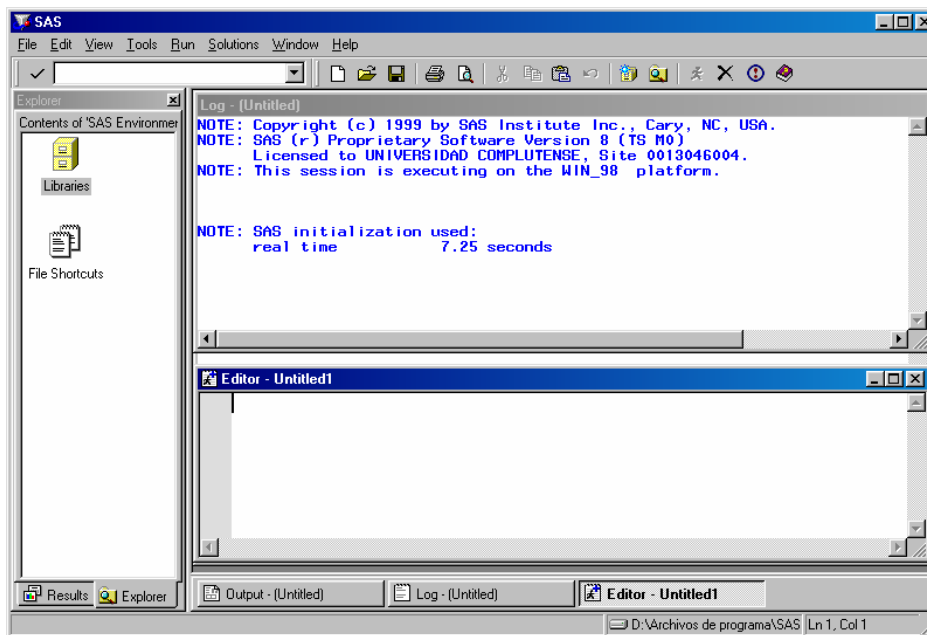
Ejecución de SAS

En WINDOWS

Por lo general realizaremos la ejecución en el entorno de ventanas.

En la ventana *editor* (o, en su defecto la ventana *program editor*), se incluirá el programa SAS. La salida de resultados, si existe, se incluye en la ventana *output* y la salida de diagnósticos en la ventana *log*. Si se desea almacenar en un fichero el contenido de una ventana, se activa la ventana y se realiza desde *File/Save* o *File/Save as*.

En general los ficheros de sentencias sas serán de tipo sas, los de salida de diagnósticos de tipo log y los de salida de resultados de tipo lst.



EN UNIX

Debido al número menor de licencias que hay en la máquina, por lo general se enviará a ejecución un fichero de sentencias sas (Véase apéndice I). SAS como resultado de la ejecución devolverá dos ficheros: el de tipo lst y el de tipo log.

La ejecución de forma no interactiva requiere la inclusión del programa SAS en un fichero. Si el segundo nombre es **sas**, la ejecución se realiza mediante: **sas primer-nombre**. De dicha

ejecución se obtienen los ficheros *primer-nombre.lst* y *primer-nombre.log*, que incluyen respectivamente la salida de resultados si existe y la salida de diagnósticos.

Si el segundo nombre del fichero es distinto de *sas*, la ejecución se realiza mediante: **sas nombre-fichero**. De dicha ejecución se obtienen los ficheros *nombre-fichero.lst* y *nombre-fichero.log*, que incluyen respectivamente la salida de resultados si existe y la salida de diagnósticos.

Estructura de un programa SAS

Conjunto de datos.

Es el conjunto de datos que se desea analizar. Puede incluirse:

- ♣¹En el fichero de sentencias SAS,
- ♣Permanecer en otro fichero externo. El fichero externo puede contener:
 - ♣Los datos en ASCII
 - ♣Los datos en una base de datos, por ejemplo EXCEL
 - Los datos en un conjunto de datos SAS o SAS-FILE permanente. Los ficheros de datos SAS son ficheros internos del entorno SAS. Son permanentes si existen en el disco de una sesión a otra de SAS. Ser introducidos directamente en SAS .

Sentencias de control o programa SAS[®].

Son las sentencias que indican al SAS la procedencia de los datos y sus características, la necesidad de realizar transformaciones a los mismos, el tipo de análisis que se desea realizar y qué resultados obtener. Un programa SAS está formado por sentencias SAS que se encuentran agrupadas en pasos DATA o pasos PROC, quedando determinada la terminación de un paso por el comienzo de otro.

En los pasos DATA se crean o modifican conjuntos de datos SAS que serán procesados por los pasos PROC. Asimismo, pueden utilizarse para crear ficheros externos de datos o generar informes mediante programación.

Los pasos PROC teniendo como entrada un conjunto de datos SAS, son los que realizan los análisis relativos a los mismos. Comienzan por la palabra PROC seguido del nombre del procedimiento.

Ejemplo de programa SAS[®]:

```
/*          Título en la salida de resultados          */
TITLE 'Este es un programa que sirve de ejemplo';

/*          Creación de formatos de escritura          */
PROC FORMAT;
VALUE sex 1='varon' 2 = 'mujer';

/*          Creación del SAS-FILE a analizar y entrada de datos          */
```

¹ El símbolo ♣ indica las opciones más usuales o las que corresponden a este manual de usuario.

```

DATA analizo;
INFILE 'morfo.datos';
INPUT id $4. (tamaño altura peso) (5.2) sexo 2.;
ARRAY algun{2} tamaño altura;

/*      Identificación de valores perdidos o no observados      */
DO I = 1 TO 2;
    IF (algun{I} = 99.99) THEN algun{I} = . ;
END;
IF (sexo = 9) THEN sexo = . ;

/*      Transformación de variables      */
bmi = (peso**2) / (altura*100);

/*      Asignación de formatos de escritura      */
FORMAT sexo sex. ;

/*      Análisis realizados sobre el SAS-FILE creado      */
PROC SORT DATA = analizo ;
BY sexo;

PROC PRINT;

PROC MEANS;
BY sexo;

PROC TTEST COCHRAN ;
CLASS sexo;
VAR tamaño--bmi ;

```

En este ejemplo, se especifica un **título** que aparecerá en la salida de resultados. Posteriormente, se crea un **formato** de escritura que podrá asignarse a una variable. En el paso **DATA**, se crea un **conjunto de datos SAS o SAS-FILE**, se especifica el **fichero externo** donde se encuentran los datos, cuáles son las **variables** de las que consta, se especifican **valores de no observación**, se **crea una variable** transformación de otras en entrada y finalmente se asigna el **formato** previamente creado, a la variable sexo para que en la salida de resultados aparezcan los descodificados de la misma. Una vez creado el **SAS-FILE**, se analizará en los **procedimientos o PROC** sucesivos. **PROC SORT** ordena el **SAS-FILE** por los valores de la variable sexo, **PROC PRINT** produce un listado de los datos, **PROC MEANS** realiza una estadística básica de los mismos categorizada por los valores de sexo, **PROC TTEST** calcula el estadístico t de Student para la comparación de sexos. Se requieren también las aproximaciones de Cochran y Cox para el caso de heterocedasticidad.

Salida de resultados.

Es el resultado del análisis sobre el conjunto de datos según las pautas reseñadas por las sentencias de control. Si la ejecución se realiza de forma no interactiva aparecerá en un fichero con último nombre **lst** y si es de forma interactiva en la ventana **OUTPUT**.

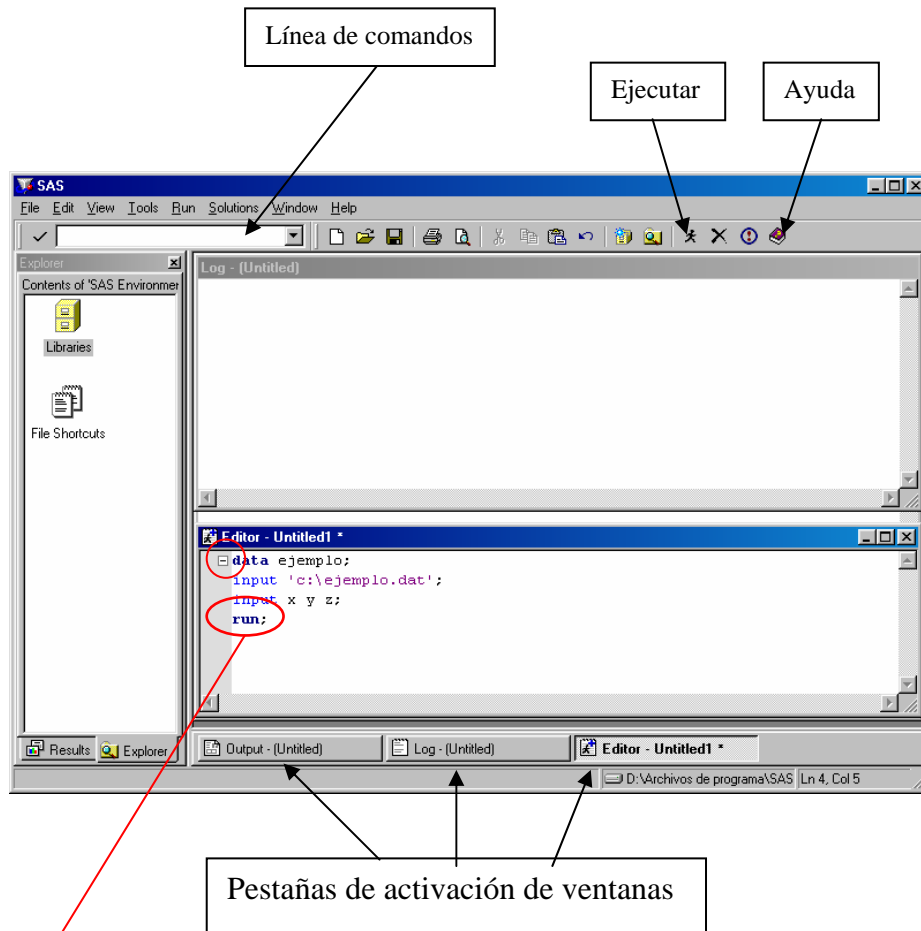
Salida de diagnósticos.

Durante la ejecución, **SAS** va realizando diagnósticos sobre los pasos que procesa. En ellos se incluyen los errores de sintaxis, el tiempo de ejecución y memoria utilizadas, las páginas de la salida de resultados en que aparecen los resultados correspondientes a cada paso, etc... Si la

ejecución se realiza de forma no interactiva aparecerá en un fichero con último nombre **log** y si es de forma interactiva en la ventana LOG.

Interfaz de usuario

Esta interfaz de usuario corresponde a la versión sas v8



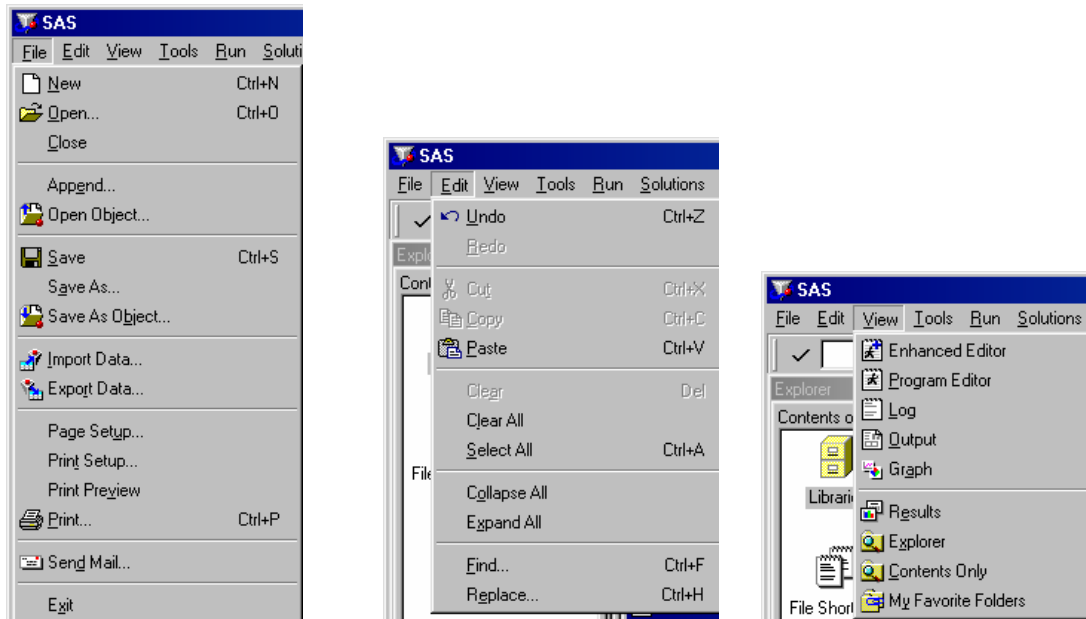
Cuando la ventana *editor* o *program editor* están activas, el icono de ejecución permite la ejecución de partes del contenido de la ventana correspondiente sin más que seleccionar esa parte con el ratón. Si no se ha seleccionado un texto, se ejecuta todo el contenido de la ventana.

La sentencia `run;` marca el final de un paso `data` o paso `proc`. El final de los pasos `data` o `proc` queda marcado igualmente por el inicio de otro paso `data` o `proc`. Es conveniente incluir la sentencia `run;` al final de un bloque que se envía a ejecución, aunque es innecesaria incluirla entre pasos `data` o `proc`.

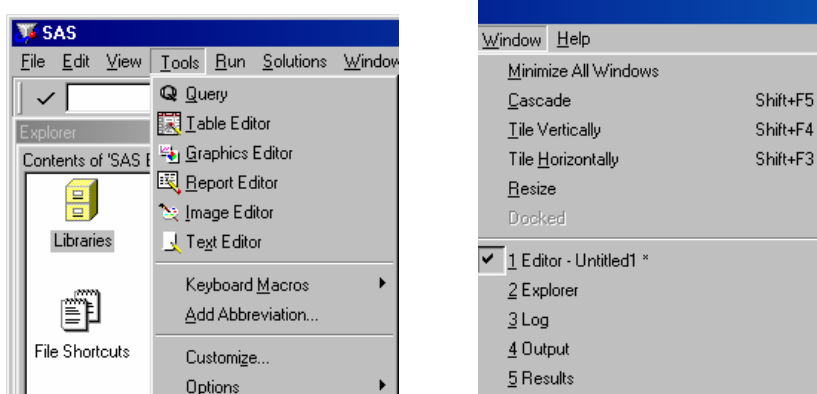
Se recomienda el uso de la ventana *editor* en lugar de la ventana *program editor* (presente en versiones anteriores de SAS como ventana PROGRAM) ya que incorpora algunas novedades:

- El contenido de la ventana *editor*² permanece en la ventana después del envío de sentencias a ejecución.
- Es sensible al lenguaje mostrando diversos colores en las palabras.
- Tiene los símbolos – y + que permiten colapsar o desplegar conjuntos de sentencias.

Menues:



El menu View permite abrir algunas ventanas si estas se han cerrado. Enhanced Editor activa la ventana *editor*. La opción Program Editor activa la ventana *program editor* de versiones anteriores a SAS V8. La opción Contents Only activa la ventana *explorer* de la forma de inicio en el entorno SAS, es decir, con las pestañas *explorer* y *results*.



² NOTA: En la ventana *program editor* si se selecciona sólo una parte con el ratón y se envía a ejecución, entonces este contenido sí permanece en la ventana.

La pestaña *explorer* (véase apéndice A.II.6, página 70) permite ver y gestionar las librerías y ficheros SAS y carpetas (shortcut) de ficheros no SAS. Permite crear librerías y ficheros sas, abrir, copiar, mover, borrar ficheros, etc... Una librería de SAS se asocia a un directorio y contiene tablas o sas-files (además de Queries y Catálogos (de formatos, gráficos, ...)).

La pestaña *results* permite navegar entre las distintas salidas de los procedimientos que se encuentran en la ventana *output*.

La ejecución de SAS en el entorno de ventanas

La ejecución se ordena desde la pantalla *editor* (o *program editor*) seleccionando con el ratón el conjunto deseado de sentencias. Si se desea ejecutar todas las sentencias de la ventana no es necesario seleccionarlas.

El final de un paso data o un procedimiento se indica por el comienzo de otro. Para indicar que el último paso data o proc (procedimiento) enviado a ejecución ha finalizado, debe indicarse el comando o sentencia *run;* .

Al enviar a ejecución código SAS, se muestran en la ventana log, el diagnóstico de la ejecución. Si se producen resultados, éstos aparecen en una ventana output que se muestra automáticamente.

I.4 NORMAS GENERALES DE SINTAXIS

-No se tiene en cuenta la posición de las palabras por columnas fijas, pudiéndose dejar tantos espacios o líneas en blanco como se quiera.

-Cada sentencia SAS comienza por una palabra clave y debe terminarse por un punto y coma (;). No existe restricción en el número de líneas a ocupar por una sentencia ni en el número de sentencias en una misma línea.

-La mayoría de las sentencias SAS tienen la posibilidad de especificar opciones, bien mediante una palabra clave, bien mediante una palabra clave seguida de = y de un valor dado por el usuario.

-La separación de elementos en una sentencia se realiza mediante uno o más espacios en blanco. Asimismo, pueden existir líneas en blanco entre elementos de una misma sentencia.

-La única restricción de posición es la de la sentencia nula (;) al finalizar las líneas de datos si éstos se encuentran en el programa SAS. Dicha sentencia nula debe encontrarse en una nueva línea.

Elementos propios de la sintaxis del SAS® .

= se utiliza en:

-sentencias de asignación.

-opciones de sentencias SAS, separando una palabra clave SAS del valor que el usuario le asigna.

' (apóstrofe) encierra:

- un título en las sentencias TITLEn
- un nombre de fichero externo en las sentencias INFILE, FILE, FILENAME, %INCLUDE.
- una descodificación en el PROC FORMAT, en la sentencia VALUE.
- un nombre de directorio en la sentencia LIBNAME.

/ (slash) en ocasiones, separa en una sentencia SAS, la lista de opciones.

() (paréntesis) agrupa términos o expresiones aritmético-lógicas o carácter.

{ } (llaves) encierra el índice de los elementos de un array.

alt1-alt3 (por ejemplo) indica las variables de nombre alt1, alt2 y alt3.

vari--varf indica todas las variables comprendidas entre vari y varf.

/* comentario */

ó *** comentario ;** incluye comentarios en el programa SAS.

\$ usado en la designación de *macro-variables*.
Indicación de variables carácter.

% usado en la designación de *macros* y en la sentencia %INCLUDE.

Características de los nombres aportados por los usuarios en SAS® .

Estos pueden ser nombres de variables, SAS-FILEs, formatos, *macros*, etc...

Los nombres aportados deben constar de 1 a 32 caracteres alfanuméricos o _, no pudiendo comenzar por un número. Los nombres de variables permiten letras mayúsculas y minúsculas.

Los nombres de formatos no pueden terminar en número y si se atribuyen a variables carácter deben comenzar por \$. Asimismo, SAS se reserva el uso de determinados nombres, como algunos que empiezan y terminan por _, así: _N_, _ERROR_, _I_,...

Convenciones de notación en esta guía.

[] (corchetes) encierra algo que es opcional.

| (barra) indica una disyunción excluyente.

... (puntos suspensivos) indican la posible repetición de lo inmediatamente anterior.

MAYUSCULA indica palabras clave de SAS.

minúscula

indica palabras que debe proporcionar el usuario.

Observación.

En las sentencias que se muestren en esta guía, se especificarán algunas opciones posibles. Debe notarse que existen para todas ellas muchas otras posibilidades.

II. ENTRADA DE DATOS EN SAS³

II.1 CREACION DE CONJUNTOS DE DATOS SAS[®] (SAS-FILES).

Como ya se ha dicho anteriormente, es necesaria la creación de un SAS-FILE para el procesamiento de cualquier conjunto de datos por los procedimientos de SAS.

Un SAS-FILE es un fichero con una estructura especial creada por SAS y almacenado en binario. Este fichero almacena, además de los datos, toda la información relativa a las variables, sus nombres, etiquetas, valores desconocidos, tipo de los datos, etc...

Los SAS-FILES pueden ser temporales a la ejecución del programa SAS o permanentes. Los SAS-FILES temporales se almacenan automáticamente en una librería de datos WORK. Los SAS-FILES permanentes se almacenan en una librería de datos que el usuario debe especificar con el comando LIBNAME o con la interfaz de usuario de SAS v8. El nombre del SAS-FILE se compone, en este caso, del nombre de la librería, seguido de un punto y del nombre del fichero donde se almacena el SAS-FILE.

Creación de un SAS-FILE

Éste puede obtenerse:

-Como salida de un procedimiento

-En un paso DATA:

- a) Por entrada de datos del usuario contenidos en un fichero externo en ASCII (Véase II.A y II.B) o insertados en el código SAS (Véase II.A).
- o bien b) Mediante programación, haciendo uso de las herramientas de SAS: Asignación, operaciones, funciones... (Véase capítulo III).
- o bien c) Por entrada de datos de otros SAS-FILES creados con anterioridad. (Véase capítulo IV).

la entrada de datos de ficheros externos en ASCII también se puede realizar con el entorno gráfico de SAS V8 (Véase II.B)

-Por importación de datos de otros formatos (Ej: Excel). Con el procedimiento import o con la interfaz de usuario de SAS V8.0 (Véase II.C)

- Mediante introducción de datos directamente en SAS (véase II.D)

³ En II.B también se incluye la exportación de datos

Como veremos más adelante y a modo de resumen, en el caso de crearse el SAS-FILE en el paso data, las sentencias necesarias en un paso DATA son:

-En todos los casos, la sentencia DATA para indicar el comienzo del paso.

-En **a)**, INFILE ó CARDS e INPUT.

-En **b)**, ninguna de las de **a)** y el resto son optativas

-En **c)**, ninguna de las de **a)**, y alguna de SET o MERGE o UPDATE, sin mezclar sentencias de distinto nombre en el mismo paso DATA.

Características tabulares de los SAS-FILE.

Los SAS-FILES son conjuntos de datos específicos de SAS que contienen, en formato binario, matrices de casos u observaciones (filas) por variables (columnas). Cada caso está compuesto por los valores de las variables que toma dicha observación.

La tabularización, sin embargo, no requiere que los ficheros de datos aportados por el usuario en entrada, sean rectangulares, ya que SAS tiene posibilidad de entrada de datos en muchas estructuras posibles.

II.2 LIBRERÍAS DE DATOS SAS®

Una librería de datos SAS es un conjunto de SAS-FILES en un directorio. Un mismo directorio puede incluir diferentes librerías SAS.

Por defecto, los SAS-FILES que se crean en una sesión se almacenan en una librería de datos temporal que al terminar la sesión desaparece.

En general, y en relación con SAS-FILES, sólo se hace referencia a una librería SAS en el caso de que se desee crear un SAS-FILE permanente o leer de un SAS-FILE permanente.

La referencia de una librería SAS se especifica en una sentencia LIBNAME (o mediante una variable de entorno). La sentencia LIBNAME es externa a un paso DATA o PROC y debe especificarse en el programa con anterioridad a su uso.

Cada librería SAS tiene asociados el directorio en el que se encuentra, una referencia (el nombre de la librería⁴).

Las librerías se pueden crear con la sentencia LIBNAME o con ayuda de la interfaz gráfica de SAS V8.

Sentencia LIBNAME

```
LIBNAME ref-librería [máquina] 'directorio';
```

⁴). Opcionalmente se puede asociar a una librería una máquina, que indica cómo son accesibles los datos de dicha librería. Si no se especifica una máquina, se toma por defecto la librería de los ficheros SAS-FILE correspondientes a la versión actual de SAS.

ref-librería: nombre que se da a la librería
directorio: directorio de la librería. Dicho directorio debe existir previamente.

NOTA IMPORTANTE. En otra sesión SAS, aunque los SAS-FILES permanentes existen en el disco, la librería de datos creada no es visible en la ventana *explorer*. Para hacerla visible (y así que sean visibles los SAS-FILES correspondientes) basta con ejecutar la sentencia LIBNAME en la nueva sesión. Si se desea que SAS acceda automáticamente a las librerías al iniciar la sesión, basta con incluir las correspondientes sentencias LIBNAME en el fichero autoexec.sas que se encuentra en el directorio donde se encuentra el ejecutable sas.exe.

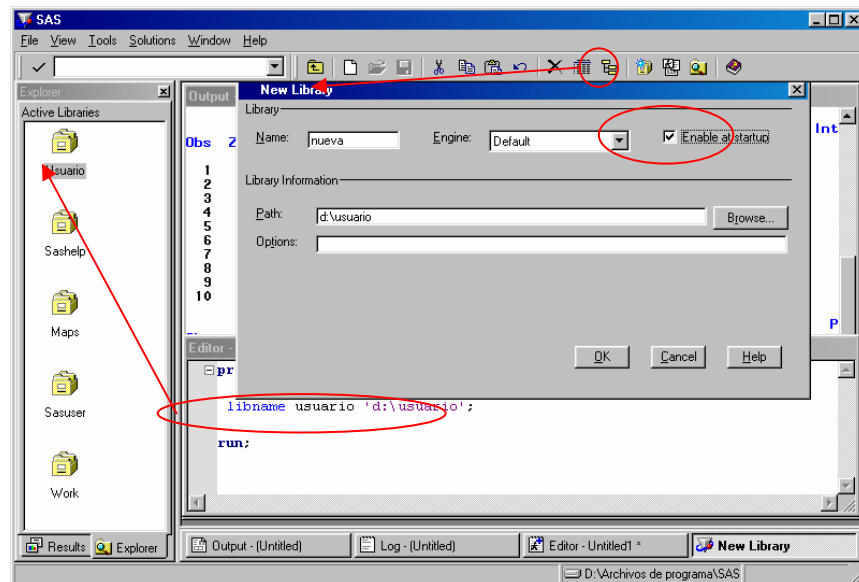
Creación de librerías de datos SAS con el entorno gráfico de SAS V8

Se pueden también crear librerías de datos SAS con el entorno gráfico de versión SAS V8 de la siguiente forma:

- Activada la ventana *explorer*⁵ y sin seleccionar una librería ni estar en el interior de ninguna de las librerías: Menu File /New/ library
- O bien con el icono que aparece en la imagen siguiente.

En esta imagen se crean dos librerías:

- La librería usuario se ha creado con la sentencia LIBNAME;
- Y la librería nueva se crea con el icono de la ventana. La opción 'Enable at startup' hace posible que la librería creada sea visible en otra sesión de SAS.



Las librerías creadas con el entorno gráfico no generan el código de la sentencia LIBNAME.

NOTA IMPORTANTE. Las librerías creadas:

- Con la sentencia LIBNAME

⁵ La ventana *explorer* se presenta en el apéndice II

- O, Activada la ventana *explorer* Menu File /New/ library
 - O, con el icono de creación de librerías sin activar la opción 'Enable at startup'...
- ... en una sesión de SAS, no son visibles en la ventana *explorer* de otra sesión SAS. Para hacerlas visibles (y así que sean visibles los SAS-FILES correspondientes) basta con ejecutar la sentencia LIBNAME en la nueva sesión. Si se desea que SAS acceda automáticamente a las librerías al iniciar la sesión, basta con incluir las correspondientes sentencias LIBNAME en el fichero autoexec.sas que se encuentra en el directorio donde se encuentra el ejecutable sas.exe.

II.A. ENTRADA DE DATOS Y DECLARACION DE VARIABLES en el PASO DATA

En esta sección, se especifica la entrada de datos del usuario contenidos en un fichero externo en ASCII o contenidos en el programa SAS. En la sección siguiente se muestra la entrada de datos de ficheros externos en ASCII con el entorno gráfico se SAS V8.

Al crear un SAS-FILE, los datos de entrada pueden encontrarse en un fichero externo o en el fichero del programa SAS. En ambos casos son necesarias las sentencias DATA e INPUT. En el primer caso se utilizará la sentencia INFILE para indicar el fichero externo y en el segundo la sentencia CARDS, para indicar el comienzo de los datos; que una vez finalizados irán seguidos por la sentencia nula (;). Las secuencias de sentencias obligatorias, son respectivamente:

(si datos en fichero externo)

```
DATA
INFILE
INPUT
(otras sentencias SAS)
```

(si datos en programa SAS)

```
DATA
[ INFILE CARDS ; ] (optativa)
INPUT
CARDS;
(datos)
;
```

Ejemplos básicos de paso DATA.

A) Si los datos se encuentran en un fichero externo:

```
DATA ejemplo;
INFILE 'ejemplo.dat';
INPUT pac sexo altura hdl ldl htamin htamax;
```

B) Si los datos se encuentran en el programa SAS

```
DATA ejemplo;
INPUT pac sexo altura hdl ldl htamin htamax;
CARDS;
1 1 150 121 80 14.0 8.0
2 1 164 100 70 17.0 9.5
3 2 175 98 64 16.5 8.5
4 2 180 110 70 14.0 6.5
```

.....
;

Modo de ejecución de un paso DATA.

En general y en el caso que se lean observaciones, un paso DATA en el que no hay sentencias de variación de flujo, se ejecuta tantas veces como observaciones se leen. Leída una observación, se procesan sobre ella todas las sentencias ejecutables del paso de forma secuencial. Terminada la ejecución del paso DATA al llegar a la última sentencia del mismo, y una vez incluida la observación en el SAS-FILE, SAS comienza una nueva ejecución del paso DATA para la siguiente observación y así sucesivamente.

Variaciones a esto:

- No hay sentencia INPUT, es decir, los datos son creados en el paso DATA.
- Existen sentencias ejecutables anteriores a la sentencia INPUT. Dichas sentencias se ejecutarán antes de cada lectura de datos.

SENTENCIA DATA.

La sentencia DATA:

- Indica el inicio del paso DATA y da nombres a los SAS-FILES que se crean.
- Los SAS-FILES pueden ser temporales a la ejecución del programa SAS o almacenarse en un fichero de forma permanente para su uso en otros programas SAS.
- Un SAS-FILE temporal puede identificarse por un nombre aportado por el usuario o bien por un nombre dado por defecto por SAS: `_DATA1_` para el primero del programa, `_DATA2_` para el segundo y así sucesivamente.
- La creación de un SAS-FILE permanente requiere la creación precedente de una librería de datos SAS (almacén de SAS-FILES) mediante la sentencia LIBNAME o el entorno gráfico. El nombre de los SAS-FILES de una librería de datos, se identifica por el nombre de la librería, seguido de un punto (.) y seguido de otro nombre.
- Puede no crearse un SAS-FILE en la ejecución de un paso DATA si se especifica, en lugar de un nombre, la palabra `_NULL_`. Esto puede tener utilidad, por ejemplo, si el paso DATA es utilizado para salida de ficheros externos. (Véase capítulo III).

Sintaxis:

```
DATA _NULL_ ;
```

ó DATA nombre | nombre-lib.nombre [(opción-sas-file ...)] ... ;

En el caso de creación de un SAS-FILE permanente, debe especificarse *nombre-lib.nombre*. Además, previa a la sentencia DATA, debe crarse la librería *nombre-lib*, bien mediante la sentencia:

```
LIBNAME nombre-lib 'directorio-lib';
```

Bien mediante el entorno gráfico.

El SAS-FILE creado se almacena en disco en el directorio *directorio-lib*, en un fichero llamado *nombre.sas7bdat*.

Ejemplo:

```
LIBNAME censo 'c:/datos censales/sasfiles';  
  
DATA censo.año2000;  
INFILE 'c:/datos censales/censob.dat';  
INPUT id edad sexo edad profesión estudios;
```

En este ejemplo estamos leyendo datos del fichero externo censob.dat en ASCII que se encuentra en el directorio c:/datos censales. Creamos un sas-file permanente, **censo.año2000**, que se encuentra almacenado en el directorio c:/datos censales/sasfiles con el nombre de fichero año2000.sas7bdat.

Algunas opciones de SAS-FILE, a las que retornaremos una vez que el capítulo IV presente el significado de las sentencias correspondientes en el paso DATA :

FIRSTOBS = n	Indica comienzo de proceso a partir de la observación n-ésima.
OBS = n	Indica el procesamiento de n observaciones.
WHERE = (expresion)	Indica selección de observaciones que cumplen la expresión.
TYPE = tipo	Indica tipo de datos: DATA, CORR, COVA...
DROP = var ...	Exclusión de variables en el SAS-FILE.
KEEP = var ...	Mantenimiento de variables en el SAS-FILE.
RENAME = (nombreviejo = nombrenuevo ...)	Renombra variables.

SENTENCIA INFILE.

La sentencia INFILE:

-Identifica como entrada, un fichero externo de datos en ASCII. (salvo INFILE CARDS; , que, además de ser optativa sólo se especifica si los datos se encuentran en el programa SAS). Se debe especificar el nombre del fichero⁶ en disco, encerrado por comillas ('). Es conveniente especificar la ruta completa del fichero.

⁶ Se puede identificar el fichero mediante una referencia o nombre lógico que identifica el fichero. Cuando se especifica el nombre de un fichero es conveniente especificar la ruta completa del fichero. En este caso de

Ejemplo:

```
DATA salud;  
INFILE 'd:\ejemplos\salud.dat';
```

- Describe las características del fichero externo donde se encuentran los datos.
- Especifica la forma de lectura de la sentencia INPUT.
- Indica modo de actuación cuando se pretende leer más allá de la longitud del fichero.

Sintaxis:

```
INFILE CARDS | 'nombre-fichero' [ opcion ... ] ;
```

Posición:

- Debe ir precediendo la sentencia INPUT correspondiente.

Observaciones:

- Puede haber más de una sentencia INFILE si se leen datos de varios ficheros externos. Una sentencia INPUT, en este caso, leerá del fichero reseñado en el último INFILE.
- Una observación puede leerse en una o más líneas. Asimismo, en una línea puede haber una o más observaciones.
- No pueden encontrarse en un mismo paso DATA alguna sentencia INFILE con sentencias SET, MERGE o UPDATE.

Algunas opciones⁷: (NOTA: registro = línea de datos)

especificación de una referencia, debe existir antes del paso DATA una sentencia FILENAME. De esta forma queda permitida la lectura de ficheros almacenados en otros dispositivos (cinta, ...)

```
FILENAME referencia 'nombre-fichero' | especific-dispositivo ;
```

Ejemplo:

Se muestra la equivalencia con el ejemplo presentado, mediante referencia:

```
FILENAME datos 'd:\ejemlos\salud.dat';  
DATA salud;  
INFILE datos;
```

⁷ Otras opciones son:

RECFM = f	Formato de registro de entrada.
LRECL = n	Longitud del registro de entrada.
BLKSIZE = n	Tamaño de bloque del registro de entrada.

FIRSTOBS = n Primer registro a leer.
OBS = n Número de registros a leer.

Para el caso en que SAS llega al final de un registro de datos según las especificaciones de la sentencia INPUT y según esta sentencia se requieran más datos del mismo registro, pueden especificarse tres opciones:

FLOWOVER (por defecto) se continúa leyendo en la siguiente línea de datos.
MISSOVER asigna valor desconocido a los datos que faltan en la línea.
STOPOVER se para la ejecución del paso DATA.

SENTENCIA INPUT.

La sentencia INPUT:

- Describe la disposición de los datos en entrada y nombra las variables correspondientes.
- Especifica el formato en que se encuentran los datos en entrada: fijo o libre; pudiéndose entremezclar en una misma sentencia INPUT los dos tipos de formato.
- Por defecto, SAS considera que todas las variables son numéricas salvo que se lean con un formato carácter o se especifique el símbolo \$ después de su nombre⁸.
- En un mismo paso DATA, pueden existir una o más sentencias INPUT para cada fichero INFILE. Una sentencia INPUT hace referencia al último INFILE especificado.
- Cada sentencia INPUT en el paso DATA señala un salto de línea en los datos, salvo indicación contraria.

Posición:

- Debe ir detrás de alguna sentencia INFILE, salvo que los datos se encuentren en el programa SAS.

Sintaxis:

INPUT especificación [especificación ...] ;

donde especificación puede ser:

LS = n	Longitud de registro que esté accesible por el puntero.
N = n	Número de registros de entrada accesibles por el puntero en una misma ejecución del paso DATA. Por defecto es $N = \max(1, \{n / \#n \text{ aparece en INPUT}\})$.
END = var	Se crea la variable lógica var, que será verdadero si se está en la última observación. No se añade al SAS-FILE.

⁸ o bien se lo notifique una sentencia ATTRIB anterior.

- nombre de variable.
- especificación de formato: por columnas, formato SAS de entrada, formato creado por el usuario.
- control de puntero: salto de columnas, salto de línea, posicionamiento en columna o línea.
- Alguno de los siguientes caracteres: \$ & () -

Especificación de formato fijo en entrada de datos:

-En este caso, para las variables correspondientes, el formato de entrada es común a todos los casos, ocupando los datos posiciones fijas en determinadas columnas del registro de entrada de datos. El formato fijo se puede especificar por:

-Indicación de columnas. Los valores de las variables correspondientes se localizan en las mismas columnas para todos los casos.

Sintaxis: coli [- colf]

Ejemplo:

```
INPUT nombre $ 1-8 edad 11-14 ;
```

-Formateados según algún formato SAS.

Sintaxis: formato

Formatos SAS más comunes son:

\$charw. o \$w. Indica una variable carácter de w caracteres.

w.[d] Indica una variable numérica de w caracteres, d de ellos decimales.

Ejemplo:

```
INPUT nombre $char8. edad 2. ;
```

_9

Especificación de formato libre en entrada de datos:

-Los valores de las variables para un caso deben estar separados por espacios en blanco, no siendo necesario que se posicionen en columnas fijas.

Ejemplo:

⁹ Formateados según un formato propio creado previamente en un PROC FORMAT.

Sintaxis: informato.


```
INPUT nombre $ edad ;
```

Opciones¹⁰ en las especificaciones de formato:

- \$ Indica que la variable predecesora es carácter.
- & En formato de lectura libre, permite que la variable carácter predecesora de este símbolo tenga como valor válido de la misma un espacio en blanco entre dos caracteres. La separación con la siguiente variable se realiza por al menos dos espacios en blanco.
- () Agrupa varias variables en las que se repite un formato. Encierra un(os) formato(s) que se repite(n).
- Indica una secuenciación numérica de variables. (Ej: x1-x3 indica x1 x2 x3).
- / Indica un salto de línea.
- @n Indica el posicionamiento en la columna n.
- +n Indica el avance de n columnas.
- @ Mantiene el puntero de lectura en un mismo registro de una sentencia INPUT a otra sentencia INPUT del mismo paso DATA y referenciados al mismo INFILE.

Ejemplo:

El valor de la variable tipo indica si un registro de datos contiene información acerca de un profesor o acerca de un alumno.

```
DATA ;
INPUT tipo $1. @ ;
IF TIPO = 'c' THEN INPUT curso profesor $ ;
ELSE INPUT nombre $ edad ;
```

- @@ Mantiene el puntero de lectura en la misma línea de datos para la próxima ejecución del paso DATA.

Ejemplo:

Cada línea de datos contiene varios valores de la variable x.

```
DATA ;
INPUT x @@ ;
```

NOTA: Se puede especificar en la sentencia INPUT lectura en formato fijo y libre simultáneamente.

Ejemplo:

```
INPUT nombre $char20. edad 3. +3 sexo 2. @54 estatus 2.
( altura x1-x13 ) ( 5.2 ) / nota $ 16-34 puntos_a
puntos_b ;
```

¹⁰ Otras opciones son:

- #n Indica el posicionamiento en el registro n-ésimo, en la actual ejecución del paso DATA. La sentencia INFILE con la opción N = k da el rango permitido de movimiento en líneas del puntero de lectura, en una misma ejecución de paso DATA.

SENTENCIA CARDS.

La sentencia CARDS especifica el comienzo de los datos. Una vez finalizadas las líneas de datos, debe especificarse en una nueva línea la sentencia nula, es decir, un punto y coma (;). La sentencia nula debe ir en la primera columna.

Sintaxis: CARDS ;

ESPECIFICACIÓN DE VALORES DE NO OBSERVACIÓN EN LOS DATOS

-Por defecto, para variables numéricas se deberá especificar en los datos el punto (.) para indicar un valor no observado y para variables carácter el blanco (). En el caso de valores desconocidos para variables carácter y si éstos son especificados por blancos, la lectura de los datos no puede realizarse mediante formato libre, al menos en las columnas correspondientes a dichas variables carácter.

- En el caso de variables numéricas y para todas las variables simultáneamente puede ampliarse el rango de valores de no observación a valores alfabéticos y _, con la sentencia MISSING.

-Si los valores de no observación especificados en los datos fueran distintos a éstos, mediante programación en el paso DATA (Ej: mediante la sentencia IF) se asignarían los valores de no observación correspondientes: punto (.) para variables numéricas y blanco (' ') para variables carácter.

-SAS asigna valores de no observación cuando hay errores en los datos, se realizan operaciones con valores de no observación, se intentan realizar operaciones fuera de rango, etc. En todos los casos, SAS lo notifica en la salida de diagnósticos.

II.B. ENTRADA DE DATOS ASCII con la INTEFAZ GRÁFICA de SAS

Se accede a ella a partir del menu File, opción *Import Data*

- Si los datos se encuentran delimitados en columnas fijas (separadas o no por algún carácter: blanco, tabulador, ...) **se recomienda** seleccionar la opción *User-defined formats*.
- Si los datos no se encuentran delimitados en columnas fijas y están delimitados por algún carácter¹¹ (ejemplo: blanco o tabulador), se **debe seleccionar** la opción *Standard Data Source* y seleccionar en source data la opción *delimited file*.

¹¹ Esta opción también es válida para datos que se encuentran delimitados en columnas fijas y están delimitados por algún carácter: blanco, tabulador, ...

NOTA IMPORTANTE: Cuando los datos no están en columnas fijas esta opción puede ser menos sencilla que la lectura con el paso DATA ya que puede no leer correctamente los datos. Por ejemplo:

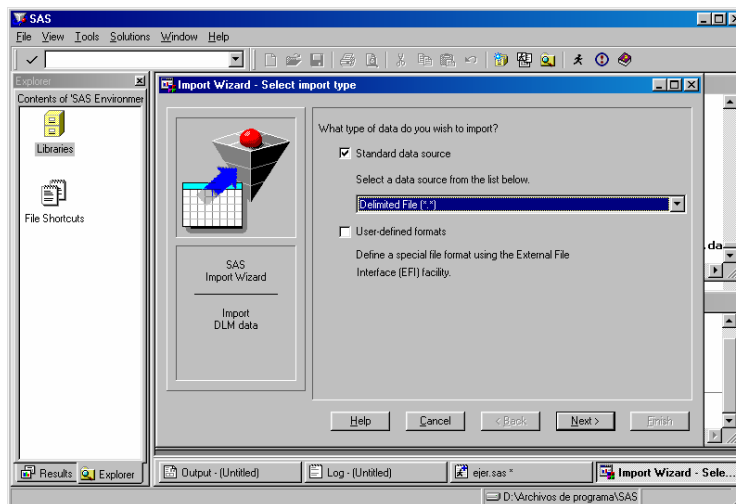
Si el separador de delimitación es un blanco y alguna línea comienza con un blanco, SAS no determina bien el campo. Una posible solución a esto es introducir un 0 en la primera columna de cada línea que comienza con blanco o llevar a la izquierda toda la línea de datos.

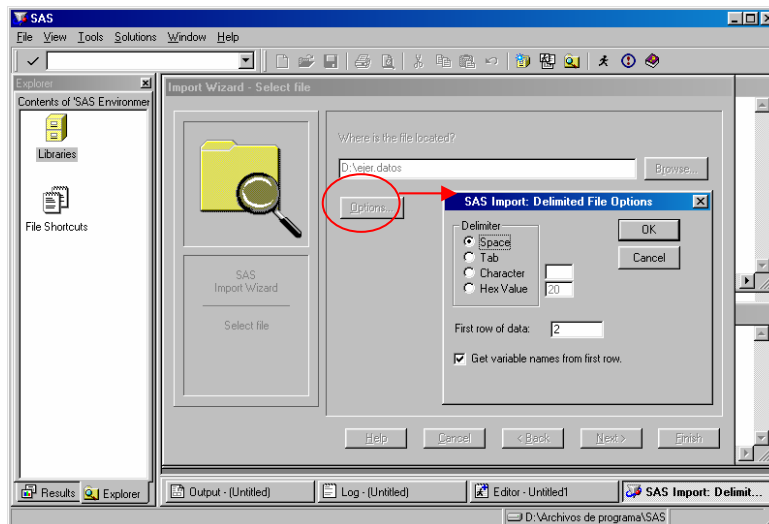
También en el caso de que el delimitador sea un blanco, si las líneas terminan en blanco, determina que hay variables a la derecha de los últimos datos y les asigna el valor missing.

- Ambas opciones pueden leer en la primera fila el nombre de las variables

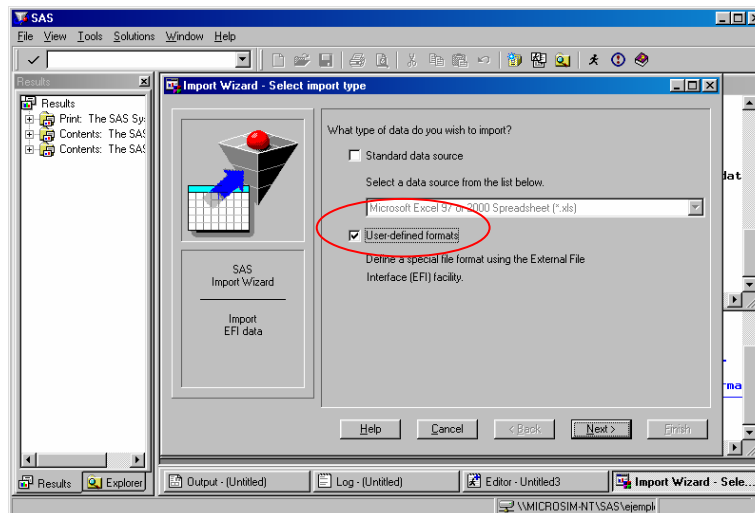
Puede generar código SAS. Una vez finalizada la lectura de los datos, se puede recuperar el código de SAS en la ventana editor o program editor, sin más que dar a la tecla F4 teniendo activa la ventana. Además, la opción Standard Data Source → delimited file permite la creación de código SAS en un fichero independiente.

File / Import Data -> Standard data source + Delimited file





File / Import Data -> User-defined formats



Se van siguiendo los pasos correspondientes en las ventanas sucesivas. Al llegar a la ventana Import, se selecciona el botón Options... y en el conjunto de opciones se selecciona:

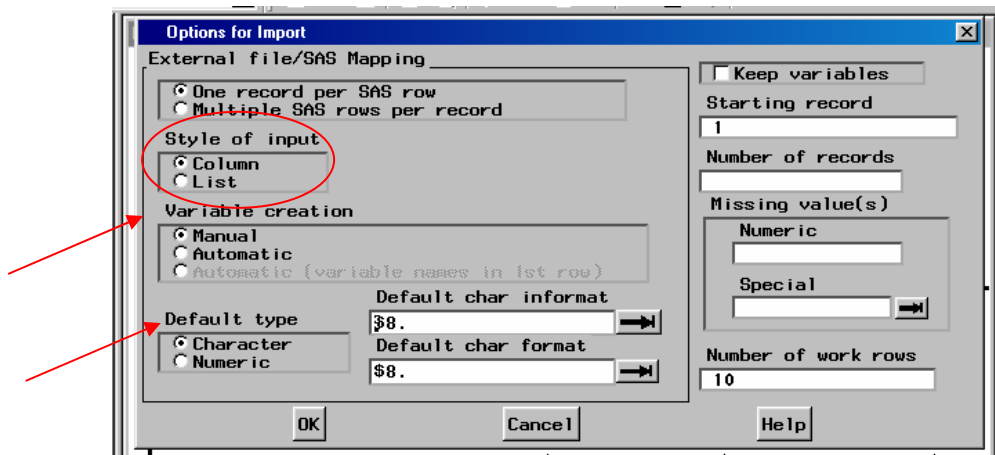
a) En el caso de que no se encuentren los nombres de variable en el fichero de datos o que no se lean de este fichero:

Style of Input → Column.

Si se desea que el tipo de las variables por defecto sean numéricas seleccionaremos la opción Default type → numeric, también debe seleccionarse esta opción.

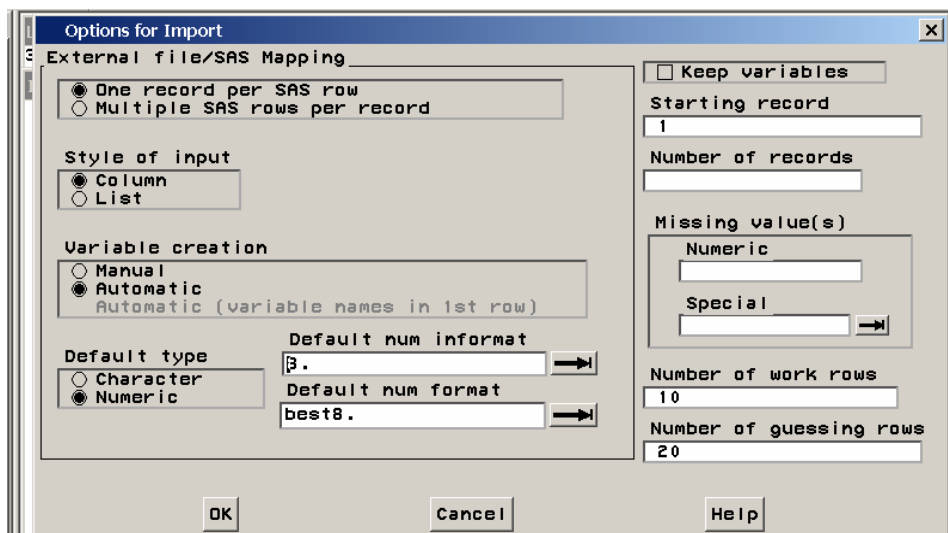
Además si deseamos que cree las variables automáticamente, seleccionaremos la opción Variable creation → Automatic

Si los nombres se encuentran en este fichero, en la primera línea de datos, debe especificarse Starting record: 2.

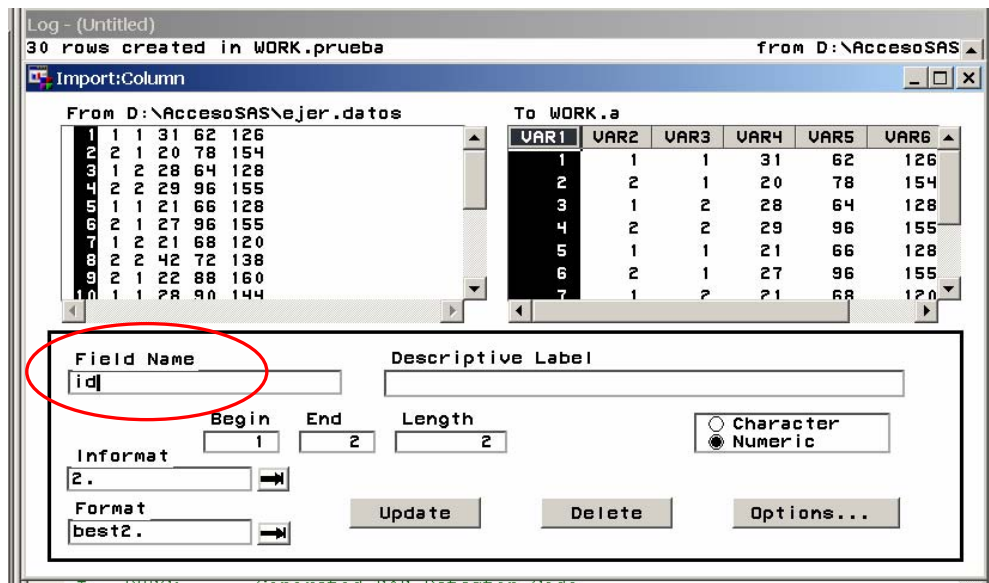


Seleccionadas las opciones correspondientes se pulsa OK y en la ventana de importación de datos a la que se vuelve, se puede cambiar para cada variable la opción de tipo de variable y se pueden nombrar las variables o borrarlas (con delete).

Las siguientes opciones:



nos devuelven el siguiente aspecto del panel de importación de datos:



donde para nombrar las variables, basta con seleccionarlas y darles un nombre y pulsar el botón update. Si deseamos borrar, pulsamos delete y si deseamos cambiar el tipo en carácter/numeric.

b) En el caso de que se encuentren los nombres de variable en el fichero de datos y se lean de este fichero:

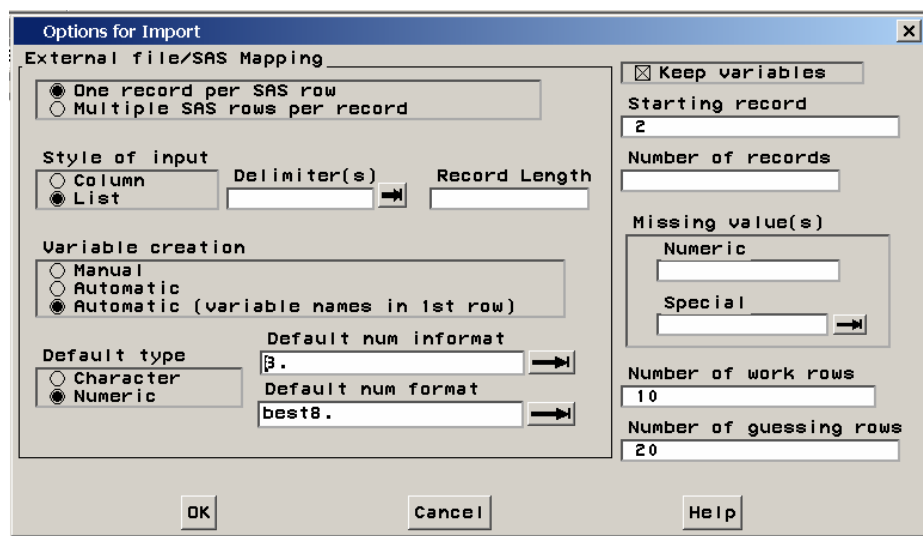
Se deben seleccionar las opciones

Style of Input → List

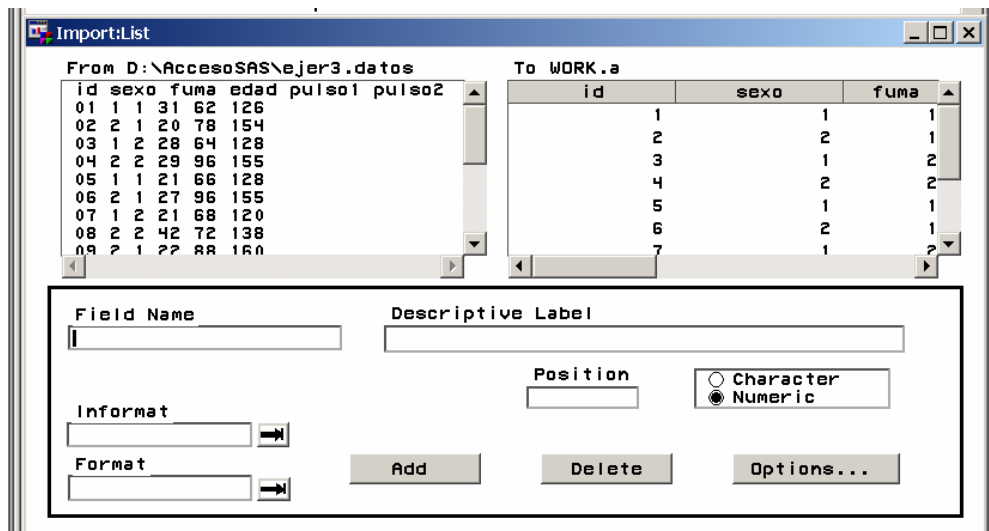
Si se desea que el tipo de las variables por defecto sean numéricas seleccionaremos la opción Default type → numeric, también debe seleccionarse esta opción.

Además si deseamos que cree las variables automáticamente y tome el nombre de la primera línea, seleccionaremos la opción Variable creation → Automatic (variable names in first row)

Seleccionar la opción sep variables y especificar en Starting record: 2.



devuelve el siguiente panel de importación de datos.



NOTA IMPORTANTE: En este modo lista si existen blancos (u otros delimitadores) en las posiciones siguientes: :

-alguna línea comienza con un delimitador

-líneas terminan en delimitador

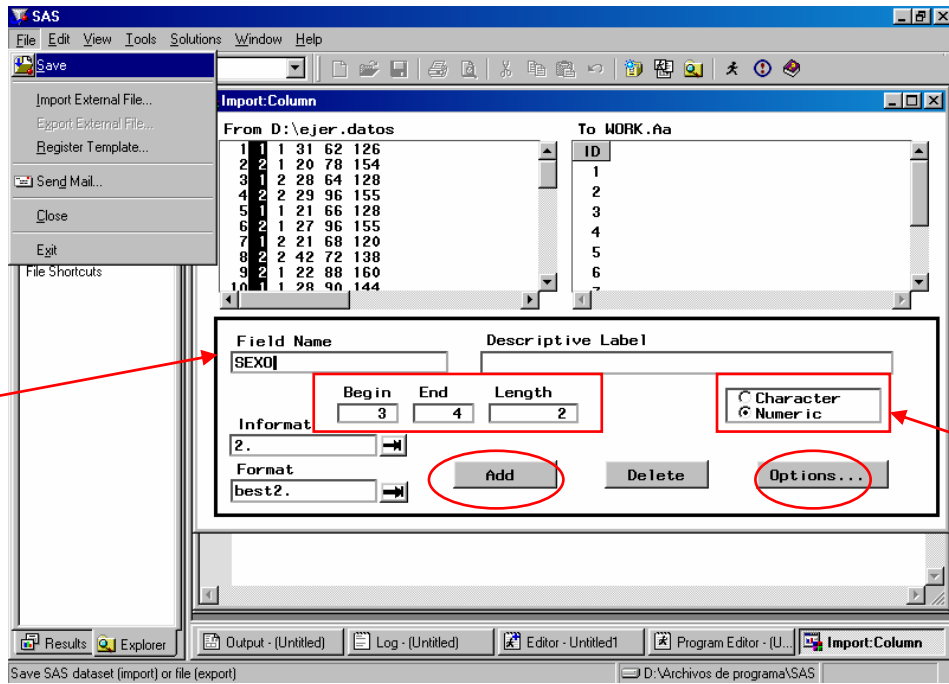
entonces SAS no determina bien el campo. Una posible solución a esto es, en el primer caso, introducir un 0 en la primera columna de cada línea de datos que comienza con delimitador o llevar a la izquierda toda la línea de datos y, en el segundo caso, determina que hay variables a la derecha de los últimos datos y les asigna el valor missing (para solucionarlo eliminar estos delimitadores).

Opcionalmente, para solucionar este problema, se puede leer como en a)

Creación manual de variables (Casos a) y b))

Se puede seleccionar en ambos casos a) y b) que la creación de variables sea manual. Al volver al panel de los datos, se deben incluir en el panel de la derecha que muestra el SAS-FILE los datos de variable en variable, indicando el tipo de variable y las columnas de inicio y final de la misma. Si la columna se selecciona con el ratón, esta información se incorpora automáticamente en el panel. Se da un nombre a la variable y se indica si es de tipo character o numeric.

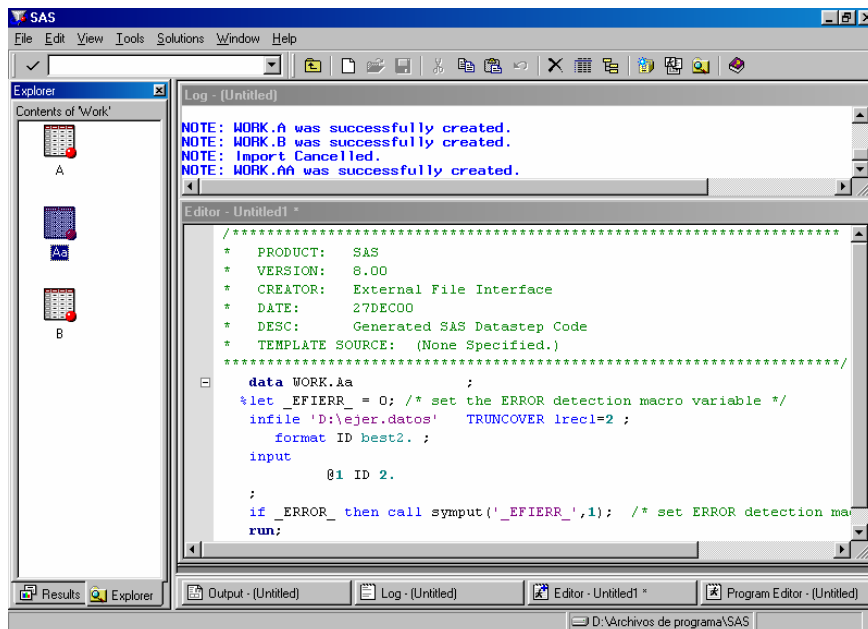
Finalmente, esta columna se incorpora al SAS-FILE con la opción ADD.



EN TODOS LOS CASOS:

Para terminar la creación del SAS-FILE, una vez creadas todas las columnas, se accede al menu File / Opcion save y se proporciona un nombre al mismo.

Código SAS que se genera al pulsar la tecla F4 cuando la ventana activa es *editor*. (En este caso el nombre proporcionado es AA y solo contiene una columna de nombre ID :



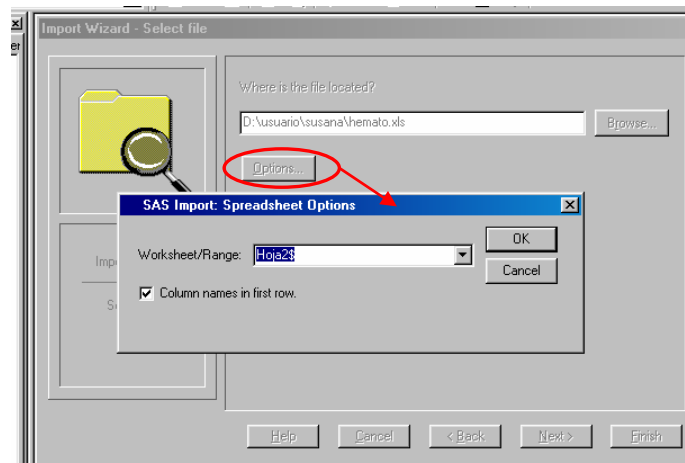
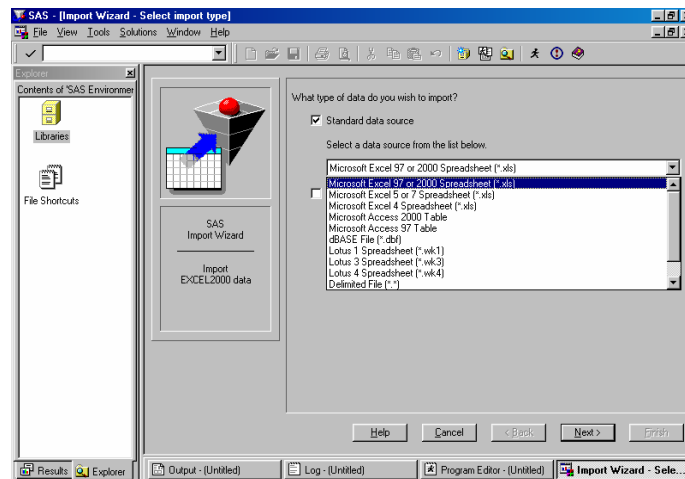
II.C. IMPORTACIÓN (EXPORTACIÓN) DE DATOS EN OTROS FORMATOS

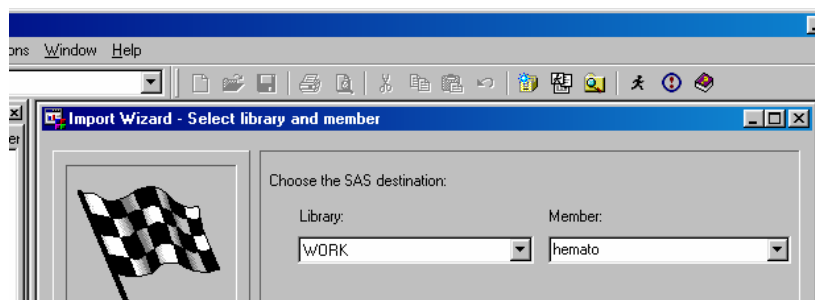
Se presenta aquí también la exportación de datos ya que es similar a la importación de los mismos.

Se puede realizar con los procedimientos Import o Export. Aquí mostramos cómo realizarlo con el entorno gráfico de SAS V8. Para ello:

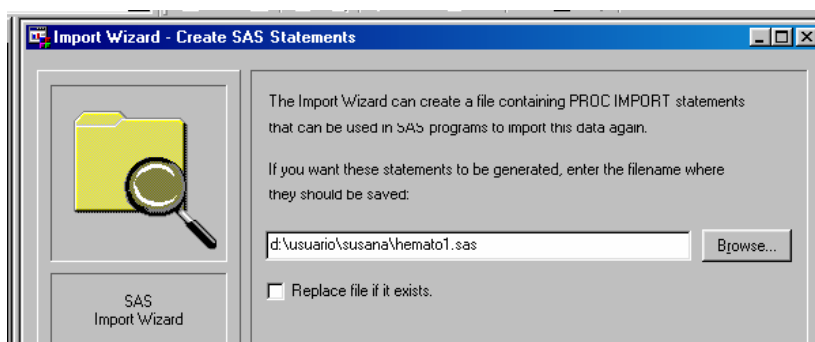
- Seleccionar en el menú **FILE** las opciones de **IMPORT DATA** o **EXPORT DATA**
- Seleccionar el tipo de formato externo de datos
- Activar Options para seleccionar las opciones correspondientes. (Por ejemplo la hoja correspondiente)
- Seleccionar el nombre del SAS-FILE que se desea crear.

En las siguientes imágenes se muestra la secuencia de ventanas correspondientes para la importación de datos de un fichero Excell.





- Opcionalmente se puede crear el código SAS que genera la importación/exportación de datos.

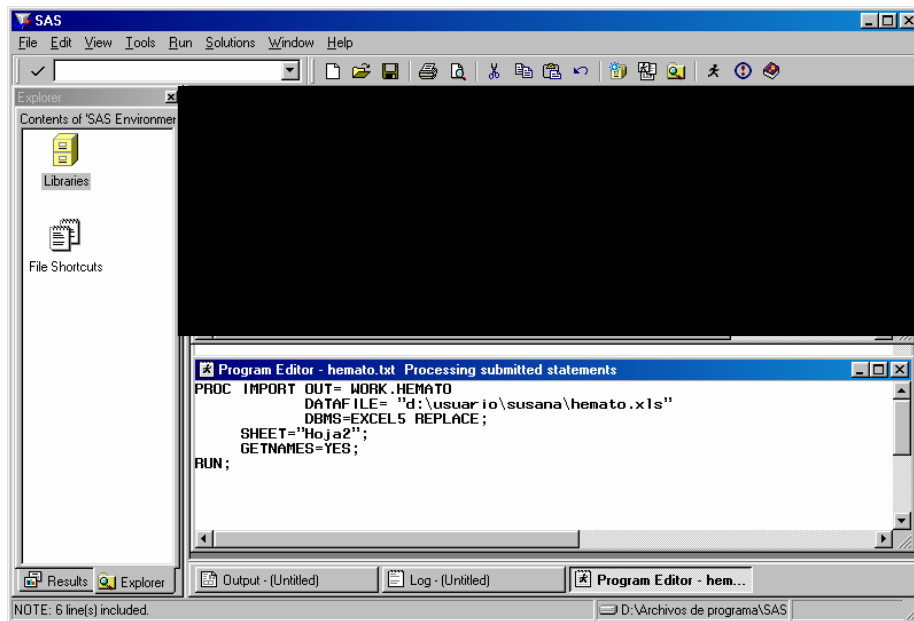


En este caso el código generado se encuentra en el fichero d:\usuario\susana\hemato1.sas y contiene:

```
PROC IMPORT OUT= WORK.HEMATO
            DATAFILE= "d:\usuario\susana\hemato.xls"
            DBMS=EXCEL5 REPLACE;
SHEET="Hoja2";
GETNAMES=YES;
RUN;
```

Se puede recuperar este código SAS¹² generado abriendo el fichero de sentencias SAS desde la ventana editor o program editor con la opción File → Open, recupera el código SAS:

¹² La tecla F4 NO facilita en la pantalla *editor* ni en la ventana *program editor* el código relativo a la importación y exportación de datos con el entorno gráfico (en el caso de EXCELL) desde la versión V8.2.



ATENCIÓN: Importar ficheros Excel en SAS puede generar datos MISSING

Este problema (los datos de algunas columnas pueden importarse como missing) puede suceder con la importación de ficheros Excel 97 o superiores si hay datos de tipos mezclados en la misma columna. SAS determinará el tipo apropiado para los datos basándose en el más común para las ocho primeras filas

Existen dos soluciones para este problema¹³:

¹³ Las soluciones a este problema son:

- A)
 1. Si queremos leer los datos como **carácter** se reformatea la columna como carácter en Excel, seleccionando la columna en cuestión, y con el botón derecho del ratón → Formato de celdas → Texto → Aceptar. Como consecuencia, la columna debería ahora estar alineada a la izquierda.
 2. Si queremos que el dato sea leído como **numérico**: En Excel, nos aseguraremos de que tenemos solamente números en cada celda. Esto incluye eliminar los espacios en blanco, que Sas tratará como carácter, seleccionando las celdas correspondientes y pulsando la tecla “supr”. Posteriormente, seleccionar toda la columna, salvo la celda con el nombre de la variable, y con el botón derecho del ratón → Formato de celdas → Numérico → Aceptar.
- B) Grabar el fichero Excel como Excel 5/95 y usar el procedimiento. Este procedimiento tiene las sentencias MIXED y TYPE, que permiten la lectura de datos de tipo mezclado. A continuación se muestra la sintaxis con un ejemplo concreto:

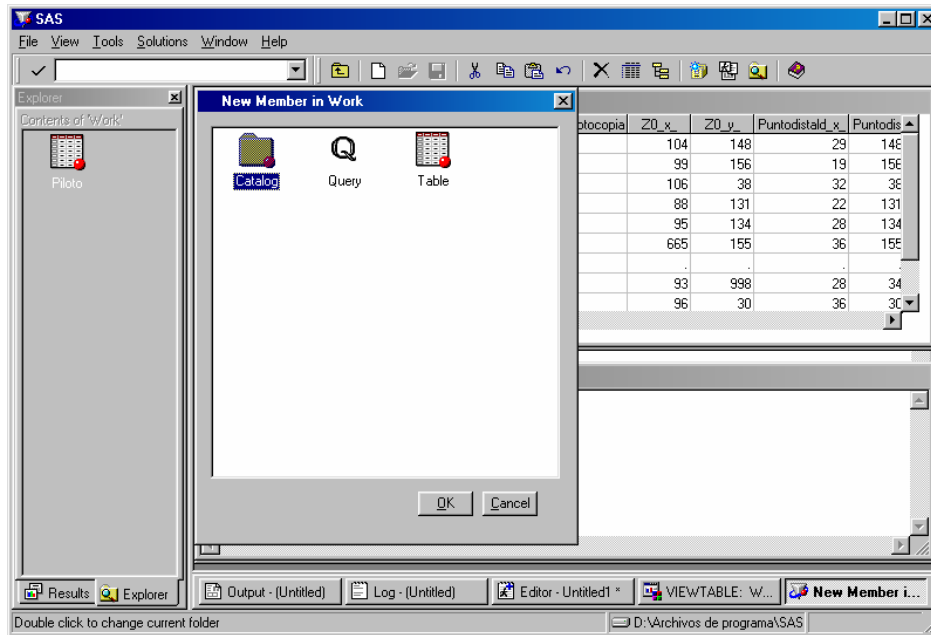
Se lee un fichero 'd:\accesoSAS\ejerb2_95.XLS'; y se crea un sas-file llamado ejercicio.

```
PROC ACCESS DBMS=XLS;
CREATE WORK.ejercicio.ACCESS;
PATH='d:\accesoSAS\ejerb2_95.XLS';
SCANTYPE=YES; /* Recorre la columna para determinar el tipo de la variable */
GETNAMES=YES; /* Lee en la primera fila los nombres de variable */
ASSIGN=YES; /* Asigna a variables SAS los nombres leídos arriba */
MIXED=YES; /* Permite la conversión de numérico a carácter de las columnas
con datos mezclados */
```

II. D. INTRODUCCIÓN de DATOS en SAS® con la INTERFAZ GRÁFICA de SAS V8

Por lo general, los datos se encontrarán en un fichero externo en ASCII u otro formato (EXCELL, dbase, etc...). Aún así, mostramos cómo podrían ser introducidos los datos en SAS con la interfaz gráfico de la versión V8, para crear SAS-files.

Activada la ventana *explorer* cuando se esta en los contenidos de una librería o seleccionada con el ratón una librería: File /New/ Table:

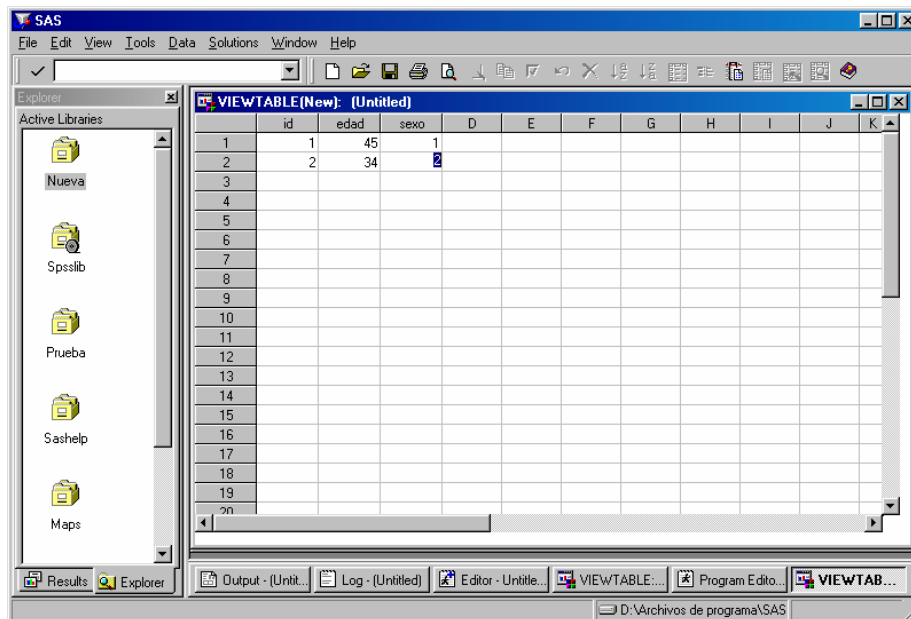


Se introducen los nombres de variable y los datos:

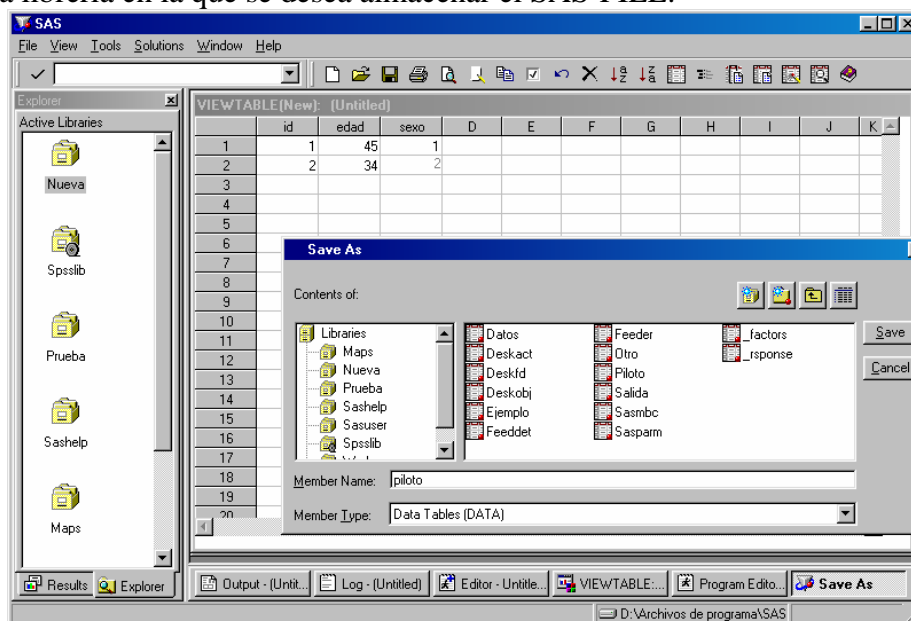
```
/* Si se desea que una variable sea de tipo carácter se debe especificar la
sentencia
TYPE nombre-variable=C;          */
CREATE WORK.ejercicio.VIEW;
SELECT ALL;
RUN;

PROC ACCESS VIEWDESC=WORK.ejercicio
OUT=WORK.ejercicio; /* nombre del sas-file generado */
RUN;
```

En este ejemplo, todas las instancias de ejercicio se pueden cambiar de nombre.



Antes de salir se nombre el SAS-File (File /save o SAS pregunta al desear salir). En el panel que aparece para nombrar y almacenar el SAS-FILE, se debe abrir con doble click de ratón izquierdo, la librería en la que se desea almacenar el SAS-FILE.



Esta utilidad no permite recuperar código SAS.

III. PASO DATA. UTILIDADES.

III.1 TRANSFORMACIÓN O CREACIÓN DE NUEVAS VARIABLES

Sentencias de asignación.

La transformación o creación de nuevas variables se realiza mediante sentencias de asignación del tipo:

$$A = B ;$$

siendo A el nombre de una variable y B una expresión aritmética o carácter.

Las expresiones aritméticas son de la forma:

$$va1 \text{ [operador-aritmético va2 ...]}$$

donde vai puede ser:

- variable
- constante
- expresión aritmética
- función aplicada a una variable o constante

Operadores aritméticos son:

- + : adición
- : sustracción
- * : multiplicación
- / : división
- ** : exponenciación

-Se puede hacer uso de paréntesis dentro de la expresión.

-En la expresión el orden de precedencia viene dado por: (), **, / y * indistintamente aquél que aparezca primero en la expresión; y - y + indistintamente también.

Ejemplo:

$$A * 7 / 4 - X * Y ** 2 + 3 \text{ equivale a } ((A * 7) / 4) - (X * (Y ** 2)) + 3$$

Funciones SAS.

-Se usan en los pasos DATA como expresiones o parte de las mismas.

-Son de muy diversas utilidades:

Aritméticas: ABS, MAX, MIN, SQRT ...

Carácter.

Financieras.
Fecha y hora.
Matemáticas: GAMMA, LOG, LOG2, LOG10, EXP, ERF ...
Probabilísticas y estadísticas: POISSON, PROBBETA, PROGCHI, PROBNORM,
PROBT, NORMAL, RANBIN, MEAN, SKEWNESS, PROBIT, FINV ...
Trigonómicas: ARCOS, SIN, TANH ...
De truncación.
Especiales.

Sintaxis:

- o nombre-función (argumento, ...)
- o CALL nombre-función (argumento, ...)

Expresiones con valores no observados.

Si en una expresión o en una función alguno de los términos o argumentos, respectivamente, tiene valor desconocido, el resultado de las mismas es desconocido.

Ejemplo:

```
logX = LOG(X) ;  
Xmean = MEAN (X1,X2,X3,X4);  
z = x**2 + y**2;
```

III.2 UTILIDADES EN EL PASO DATA

Renombrar variables.

- Se realiza mediante la sentencia RENAME.
- RENAME es también una opción de SAS-FILE.

Sintaxis:

```
RENAME nombre-viejo = nombre nuevo ... ;
```

- no se permite referenciar listas de variables en ninguno de los dos términos de la expresión.

Selección de variables en un SAS-FILE.

- Se realiza mediante las sentencias KEEP o DROP.
- La sentencia KEEP selecciona la variables a mantener en un SAS-FILE.
- La sentencia DROP selecciona las variables a eliminar de un SAS-FILE.
- Estas dos sentencias se pueden repetir en un mismo paso DATA, pero nunca deben usarse simultáneamente.
- KEEP y DROP son también opciones de SAS-FILE.

Sintaxis:

```
KEEP | DROP variable ... ;
```

Ejemplo:

Estos tres ejemplos son equivalentes:

```
DATA ejemplo;
INFILE 'ejemplo.dat';
INPUT pac sexo peso altura hdl ldl htamin htamax;
bmi = peso**2 / altura ;
drop peso altura;
```

```
DATA ejemplo;
INFILE 'ejemplo.dat';
INPUT pac sexo peso altura hdl ldl htamin htamax;
bmi = peso**2 / altura ;
keep pac sexo hdl ldl htamin htamax bmi;
```

```
DATA ejemplo;
INFILE 'ejemplo.dat';
INPUT pac sexo peso altura hdl ldl htamin htamax;
bmi = peso**2 / altura ;
keep pac sexo hdl--bmi;
```

Selección de observaciones en un SAS-FILE.

- Se realiza mediante las sentencias DELETE o OUTPUT.
- La sentencia DELETE selecciona las observaciones a eliminar de un SAS-FILE.
- La sentencia OUTPUT selecciona las observaciones a incluir en un SAS-FILE.
- Las sentencias IF o WHERE pueden utilizarse para la selección de observaciones.
- La sentencia IF se diferencia de la sentencia WHERE, en que en la primera se leen todos los valores de observación, y en la segunda se leen primero los implicados en la sentencia WHERE, y si la expresión da valor verdadero, se leen los demás.
- WHERE es también una opción de SAS-FILE.

Sintaxis:

```
DELETE ;
OUTPUT [ sas-file ... ] ;
IF | WHERE condición-lógica ;
```

(NOTA: Véase el bloque IF / THEN / ELSE para la especificación condición-lógica)

Ejemplo:

Son equivalentes:

```
DATA ;
INPUT x y ;
IF x < 5 THEN OUTPUT ;
```



```

CARDS ;

DATA ;
INPUT  x  y  ;
IF  x < 5  ;
CARDS ;

DATA ;
INPUT  x  y  ;
IF  x >= 5  THEN  DELETE  ;
CARDS ;

```

Creación de diferentes SAS-FILEs en un mismo paso DATA.

-Se puede realizar mediante la sentencia OUTPUT.

-Mediante las opciones de SAS-FILE KEEP y/o DROP, se pueden seleccionar las variables a incluir en cada uno de los SAS-FILEs creados.

Ejemplo:

```

DATA  nino  joven  ;
INFILE  'mis.datos'  ;
INPUT  nombre  $  sexo  $  edad  ;
IF  edad <= 14  THEN  OUTPUT  nino  ;
ELSE  OUTPUT  joven  ;

```

Ejemplo:

```

DATA  persona(KEEP = id nombre dir)  estad(DROP = nombre dir);
INFILE  'c:\censo.dat'  ;
INPUT  id  nombre  $char20.  dir  $char20.  x1-x20  ;

```

Creación de más de una observación por línea de datos.

-Se puede realizar mediante la sentencia OUTPUT.

Ejemplo:

```

DATA  ;
INPUT  id  x1-x3  ;
DROP  x1-x3  ;
x = x1  ;  OUTPUT  ;
x = x2  ;  OUTPUT  ;
x = x3  ;  OUTPUT  ;
CARDS  ;

```

Codificación de variables.

-Puede asignarse un descodificado a las categorías de una variable cualitativa, mediante la sentencia FORMAT, si previamente se ha creado un formato adecuado en el procedimiento FORMAT.

-Las variables continuas se pueden categorizar mediante un formato creado en PROC FORMAT, que haga asignaciones de categorías por intervalos. De esta forma, las variables a las que se asigna un formato con categorías, pueden ser consideradas categóricas (si se usan en una sentencia BY, CLASS, TABLE, TABLES) o continuas (si se usan en una sentencia VAR o no se hace mención explícita de ellas) puesto que el valor numérico original de las mismas se conserva.

-La sentencia FORMAT asocia formatos, creados en un PROC FORMAT, a variables.

N.B. Véase el procedimiento FORMAT más adelante de esta guía.

Sintaxis:

```
FORMAT especificacion-variables nombre-formato. ... ;
```

Ejemplo:

```
FORMAT edad edad x1-x10 sino.;
```

III.3.A. (AVANZADO) CREACIÓN DE FICHEROS EXTERNOS DE DATOS O INFORMES

El paso DATA puede ser de utilidad para la salida de ficheros de datos, bien por la facilidad de programación en SAS, bien por la salida de resultados interesantes de procedimientos SAS.

Asimismo, el paso DATA puede usarse para la generación de informes.

Ejemplo básico:

```
DATA _NULL_;  
INFILE 'morfo.datos';  
INPUT id $4. (tamaño altura peso) (5.2) sexo 2.;  
FILE 'morfosal.datos';  
IF (sexo = 9) THEN sexo = . ;  
bmi = (peso**2) / (altura*100);  
PUT id $4. (tamaño altura peso) (5.2) sexo 2. bmi  
7.2;
```

SENTENCIA FILE

La sentencia FILE:

-Identifica como salida, un fichero externo de datos en forma decimal. La identificación puede ser mediante el nombre del fichero en disco, encerrado por comillas ('), o bien mediante una referencia o nombre lógico que identifica el fichero. Cuando se especifica el nombre de un fichero es conveniente especificar la ruta completa del fichero.

-En el caso de especificación de una referencia, debe existir antes del paso DATA una sentencia FILENAME. La sentencia FILENAME se describe en el apartado II.3, sentencia INFILE.

-Describe las características del fichero externo.

-Especifica cabeceras que deben escribirse al principio de cada página del fichero, si el fichero es de tipo PRINT, es decir, si la referencia que identifica el fichero es PRINT o se especifica la opción PRINT.

-Indica modo de actuación cuando se pretende escribir mas allá de la longitud del fichero.

Sintaxis:

```
FILE 'nombre-fichero' | ref [ opcion ... ] ;
```

Existen algunas referencias especiales que pueden especificarse en la sentencia FILE:

-PRINT	indica el dispositivo de salida de resultados.
-LOG	indica el dispositivo de salida de diagnósticos.

Posición:

-Debe ir precediendo la sentencia PUT correspondiente.

Observación:

-Puede haber más de una sentencia FILE si se escriben datos en varios ficheros externos. Una sentencia PUT, en este caso, escribirá en el fichero reseñado en el último FILE.

Algunas opciones:

RECFM = f	Formato de registro del fichero de salida.
LRECL = n	Longitud de registro del fichero de salida.
BLKSIZE = n	Tamaño de bloque del registro del fichero de salida.
PRINT	Hace el fichero de tipo PRINT, es decir, pone control de carro. No es necesario si se especificó la referencia PRINT.
NOPRINT	No pone control de carro cuando se especifica la referencia de PRINT.
HEADER = etiqueta	Define la etiqueta de la sentencia que encabeza el grupo de sentencias que se procesan a cada nueva página. El final de dichas sentencias viene dado por RETURN; (Ficheros PRINT).
N = n PS	Número de líneas del fichero de salida, accesibles por el puntero en una iteración del paso DATA. Si PS, será toda la página. Si no se especifica, será $N = \max(1, n \text{ de } \#n)$.
PS = n	Número de líneas por página. ([20, 500]). Se incluyen las líneas ocupadas por los TITLE. (Ficheros PRINT).

MOD | OLD (por defecto) MOD indica que se escriba a continuación del fichero, OLD indica que se cree por primera vez.

Opciones al acabar una línea del fichero de salida, según las especificaciones dadas en la sentencia PUT:

FLOWOVER (por defecto) se continúa escribiendo en la siguiente línea.
DROPOVER no escribe los datos que excedan la longitud de línea.
STOPOVER se para la ejecución del paso DATA.

SENTENCIA PUT.

La sentencia PUT:

-Describe la disposición de los datos en salida y especifica el formato de escritura: fijo o libre; pudiéndose entremezclar en una misma sentencia PUT los dos tipos de formato.

-Pueden escribirse, además, cadenas de caracteres si se encierran entre comillas (') en la sentencia PUT.

-En un mismo paso DATA, pueden existir una o más sentencias PUT para cada fichero FILE. Una sentencia PUT hace referencia al último FILE especificado.

-Puede no especificarse una sentencia FILE. En este caso, se escribirá en la salida de diagnósticos.

-Cada sentencia PUT en el paso DATA señala un salto de línea en la salida, salvo indicación contraria.

Sintaxis:

PUT [especificación ...] ;

donde especificación puede ser:

-Nombre de variable [=] Si se pone el signo = aparecerá en el fichero de salida, el nombre de la variables seguido de = y seguido del valor de la variable. En otro caso, aparecerá únicamente el valor de la variable.

-Especificación de formato: por columnas, formato de salida: SAS o creado por le usuario en un PROC FORMAT.

-Control de puntero: salto de columnas, salto de línea, posicionamiento en columna o línea.

-Alguno de los siguientes caracteres: \$ () - En este caso, \$ no es necesario para indicar que una variable es carácter.

-'Cadena de caracteres'

-_INFILE_: escribe la última línea leída.

-_ALL_: escribe los valores de todas las variables incluyendo _ERROR_ y _N_. _ERROR_ indica si hay error en la observación; _N_ indica el número de veces que se ha ejecutado el paso DATA.

-_PAGE_: salta de página en el fichero de salida.

Especificación de formatos:

-Se realiza del mismo modo que en la sentencia INPUT.

- \$ () - / @n +n #n @ y @@ tienen el mismo significado que en la sentencia INPUT.

Opciones adicionales:

-@var posicionamiento en la columna indicada por la variable var.

-#var posicionamiento en la línea indicada por la variable var. Si no se especifica la opción N = n en la sentencia FILE, por defecto SAS toma el valor del mayor valor que acompaña a un #.

III.3.B. EXPORTACIÓN DE DATOS A OTROS FORMATOS CON EL ENTORNO GRÁFICO DE SAS V8

Con la opción de menú File → Export Data se pueden exportar datos a ficheros en formato ASCII de forma similar a cómo se hace con la entrada de datos en II.B. (Internamente se realiza con un paso DATA).

Se pueden exportar datos a otros formatos, como EXCEL, de forma similar a como se hace con la importación de datos en II.C. (Internamente se realiza con un proc EXPORT).

III.4 SENTENCIAS LOGICAS Y BUCLES

Bloques IF / THEN / ELSE.

Sintaxis:

```
IF expresion-logica [ THEN accion1 ; [ ELSE accion2; ] ]
```

Observaciones:

-Si la expresion-logica toma el valor verdadero se ejecutará la acción accion1 y sino se ejecutará la acción accion2.

-Cuando no aparece la cláusula THEN, debe especificarse un punto y coma (;) después de la expresión lógica. Se utiliza para seleccionar observaciones en un SAS-FILE.

-Si no aparece THEN no se puede especificar la sentencia ELSE.

-accion1 y accion2 son sentencias ejecutables, o bloques de sentencias comenzando por la sentencia DO.

Las expresiones lógicas son de la forma:

va1 operador va2 ...

donde vai puede ser:

- variable o función aplicada a una variable o constante
- constante
- expresión aritmética
- expresión lógica

Operadores son:

- | | |
|-----------------------------|-----------------------------|
| > ó GT : mayor que | >= ó GE : mayor o igual que |
| = ó EQ : igual que | ≠ ó NE : distinto a |
| <= ó LE : menor o igual que | < ó LT : menor que |
| & ó AND : y | ó OR : o |
| ¬ ó NOT : no | |

-Se puede hacer uso de paréntesis dentro de la expresión.

Valores de no observación en expresiones lógicas:

-Tienen el valor de falso en combinación con los operadores AND y OR.

Ejemplo:

```
IF ( ( a + b ) > 5 ) OR ( id = 'novale' ) THEN y = a ** 2 ;  
ELSE y = a ;
```

Bucles¹⁴. Bloques DO / END.

Sintaxis:

```
DO [ indice = n TO m [ BY k ] ] ;
```

¹⁴ Sintaxis completa de los bucles:

```
DO [ indice = n TO m [ BY k ] ] [ WHILE | UNTIL (condicion) ] ;  
(sentencias SAS)  
END ;
```

ó

```
DO OVER nombre-array ;  
(sentencias SAS)  
END ;
```

Observaciones distintivas cuando se especifican las sentencias WHILE o UNTIL:

- El grupo de sentencias entre las sentencias DO y END se ejecuta (en una misma ejecución de paso DATA) mientras el valor de la variable índice no supere m o mientras la condición especificada en WHILE o UNTIL es verdadera.
- WHILE chequea la condición antes de la ejecución de la primera sentencia del DO.
- UNTIL chequea la condición justo antes de la sentencia END. Por tanto, se ejecuta al menos una vez.

(sentencias SAS)
END ;

Observaciones:

- Si la sentencia DO aparece sin índice, se ejecutan las sentencias entre DO y END, al llegar a la sentencia DO.
- En otro caso, el grupo de sentencias entre las sentencias DO y END se ejecuta (en una misma ejecución de paso DATA) mientras el valor de la variable índice no supere m.
- n, m y k pueden ser valores numéricos o variables numéricas que tomen algún valor.
- Se puede especificar como valores posibles del índice una lista de valores:
DO índice = val1, ... valn ;
- La variable índice puede ser numérica o carácter.
- Se puede salir de un bucle con una sentencia GOTO o GO TO.
- Se pueden anidar bucles.

Ejemplo:

```
DATA ventas;  
INFILE 'fichero.ventas';  
INPUT empresa ventas1-ventas4;  
ARRAY vent{4} ventas1-ventas4;  
ventast = 0;  
DO i = 1 TO 4;  
ventast = ventast + vent{i};  
end;  
drop i;
```

Sentencia GOTO o GO TO.

Sintaxis:

GOTO etiqueta ; equivale a: GO TO etiqueta ;

Hay otra sentencia en el paso DATA que comienza por:

etiqueta: sentencia-SAS ;

La sentencia GO TO o GOTO lleva el flujo de control a la sentencia referenciada por la etiqueta.

Otros bloques de sentencias¹⁵

¹⁵ **Bloque SELECT / WHEN / OTHERWISE / END.**

Sintaxis:

```
SELECT [ ( expresion ) ] ;  
WHEN ( expresion2 ) [ accion2 ] ;  
...  
OTHERWISE [ accionn ] ;
```

III.5 OTRAS SENTENCIAS USADAS EN EL PASO DATA

Para poder referenciar variables, se pueden construir arrays¹⁶ con la sentencia:

```
ARRAY nombre-array { n1 } [ $ ] especificacion-variables ;
```

-Todas las variables deben ser del mismo tipo, numérico o carácter. Si son de tipo carácter, debe especificarse el símbolo \$.

-Posterior a esta sentencia y dentro del mismo paso DATA, ya que únicamente es una denominación de las variables, para referenciar la variable se puede hacer por el nombre de la variable o por nombre-array{i} siendo i la posición que ocupa la variable en el array.

Para poner etiquetas a variables, que aparecerán en salida:

```
LABEL variable = 'etiqueta' ... ;
```

```
END ;
```

Observaciones:

-Se evalúa la expresión de la sentencia SELECT y las expresiones de las sentencias WHEN. Sucesivamente se van comparando los valores de la expresión SELECT con los valores de las expresiones. Al coincidir la primera vez los valores, se ejecuta la acción correspondiente y finaliza la ejecución del grupo SELECT, saliendo de él.

-Las acciones pueden ser un bloque DO, SELECT o la sentencia nula (;). En ocasiones, es conveniente poner la sentencia nula en la sentencia OTHERWISE, para que no de error el SELECT si no se cumple ninguna expresión WHEN.

Ejemplo: (Son equivalentes)

```
SELECT (a) ;
    WHEN (1) x = x*10 ;
    WHEN (2) ;
    WHEN (3, 4, 5) x = x*100 ;
    OTHERWISE ;
END ;

SELECT ;
    WHEN (a = 1) x = x*10 ;
    WHEN (a = 2) ;
    WHEN (a=3|a=4|a=5) x = x*100 ;
    OTHERWISE ;
END ;
```

Sentencias LINK y RETURN.

Sintaxis:

```
LINK etiqueta;
```

Hay un bloque en el paso DATA:

```
etiqueta: sentencia-SAS ;
sentencia-SAS ; ...
RETURN ;
```

La sentencia LINK lleva el flujo de control a la sentencia referenciada por la etiqueta. Se ejecutan las sentencias SAS que se encuentran hasta una sentencia RETURN; , la cual devuelve el control a la siguiente sentencia posterior al LINK.

¹⁶ La sintaxis completa de la sentencia ARRAY es:

```
ARRAY nombre-vector { n1 [ , n2... ] } [ $ ] especificacion-variables ;
```

-En caso de especificación de más de un índice, el que se mueve más rápido es el último. En dos dimensiones se llenan primero las filas.

Para asignar formatos o informatos a variables:

```
FORMAT var ... formato. ... ;
```

Para categorizar por los valores de una o varias variables:

```
BY var ... ;
```

-Se usa en combinación con SET o MERGE.

Otras sentencias¹⁷

III.6 ESPECIFICACIÓN DE VARIABLES EN PASOS DATA O PROC.

Recapitulando, en los pasos DATA o en los pasos PROC se pueden referenciar las variables de las siguientes formas:

- Por sus nombres separados por blancos.
- Con un guión (-) entre dos variables, si existe una secuenciación numérica en el nombre de las variables. (Ej: alt14-alt23).
- Con dos guiones (--) entre dos variables, para indicar todas las variables entre la primera y la segunda. El orden en que se encuentran las variables en un SAS-FILE, viene dado por el orden de aparición de los nombres en el paso DATA.

Existen a esto algunas excepciones:

¹⁷ Para cambiar atributos de variables: ATTRIB

Para devolver el control de ejecución: RETURN ;

-Si no se encuentra asociada a una etiqueta, bien por una sentencia LINK, bien por una sentencia FILE, envía el control a la primera sentencia del paso DATA.

Para terminar la ejecución del paso DATA: STOP ;

Para asignar informatos a variables:

```
INFORMAT var ... formato. ... ;
```

Para retener valores de variables de una ejecución a otra del paso DATA: RETAIN

Para acumular valores de una ejecución a otra del paso DATA: Sentencia de suma:

```
var + expresion ;
```

-Se va asociando a la variable var el resultado de la expresión en cada una de las observaciones. var es valor acumulado de varias ejecuciones de paso DATA.

-En la opción de SAS-FILE, RENAME no se permite referenciar varias variables, antes o después del signo igual. Se debe poner una lista de tríos del tipo: nombre-viejo = nombre-nuevo .

-En la sentencia RENAME del paso DATA, tampoco está permitido.

Ejemplo:

```
DATA ejemplo;
INFILE 'ejemplo.datos';
INPUT  pac  sexo  edad  peso  altura  htamin  htamax  deporte
pulso1  pulso2;
bmi = peso**2 / altura;
```

Una vez nombradas las variables como arriba, son equivalentes:

A)

```
pac  sexo  edad  peso  altura  htamin  htamax  deporte  pulso1
pulso2  bmi
```

Y

```
pac -- bmi
```

B)

```
edad  peso  altura  htamin  htamax  pulso1  pulso2  bmi
```

Y

```
edad - htamax  pulso1 -- bmi
```

IV. MANIPULACIÓN DE CONJUNTOS DE DATOS SAS®.

Creados conjuntos de datos SAS, éstos pueden ser analizados por procedimientos SAS.

Los conjuntos de datos SAS, además, pueden ser leídos en pasos DATA para su transformación en otros conjuntos de datos SAS. Para ello se utilizan las sentencias SET y MERGE.

Entre la operaciones posibles, con SAS-FILES en los pasos DATA, se encuentran:

Modificar SAS-FILES.

Concatenar varios SAS-FILES uno a continuación del otro.

Entremezclar varios SAS-FILES por los valores de una variable.

Concatenar varios SAS-FILES uno al lado del otro

IV.1 SENTENCIA SET. UTILIDADES

La sentencia SET:

-Lee observaciones de uno o varios SAS-FILES.

-No puede combinarse en un mismo paso DATA con sentencias INFILE, MERGE o UPDATE.

-Pueden especificarse varias sentencias SET referidas o no a un mismo SAS-FILE.

-Cada sentencia SET aplicada al mismo SAS-FILE lleva su propio puntero de localización.

-Si se usa en combinación con la sentencia BY, ésta debe aparecer inmediatamente de la sentencia SET. Además los SAS-FILES, deberán estar ordenados por los valores de las variables BY, que deberán ser las mismas en todos los SAS-FILES implicados.

Sintaxis:

```
SET sas-file [ ( opcion-sas-file ... ) ] ... opcion-set ;
```

Utilidades de la sentencia SET:

-Realizar una copia de un SAS-FILE para eliminar variables u observaciones, crear nuevas variables o transformar las ya existentes.

-Concatenar varios SAS-FILES uno debajo de otro. Si el conjunto de variables no coincide, se pondrán a desconocido las variables de cada conjunto que inicialmente no existieran.

```
SET A B;
```

-Entremezclar varios SAS-FILES, por los valores de alguna variable.

```
SET A B ;  
BY var ... ;
```

-Esto es equivalente al punto anterior, sólo que se realiza por los valores de las variables especificadas en BY.

-Concatenar varios SAS-FILES uno al lado del otro. Si coincide alguna variable, en el SAS-FILE creado, ésta tomará el valor que tomara en el último SAS-FILE especificado.

```
SET A;  
SET B;
```

-Si el número de observaciones de los SAS-FILES es distinto, la ejecución del paso DATA finaliza cuando el número de ejecuciones coincida con el menor número de observaciones en un SAS-FILE.

-El caso anterior, con una sentencia BY. Es equivalente al anterior, sólo que realizado por los valores de las variables del BY.

IV.2 SENTENCIA MERGE. UTILIDADES

La sentencia MERGE:

- Combina observaciones procedentes de dos o más SAS-FILES en una nueva observación. La forma de combinarlas depende de la presencia o no de una sentencia BY.
- No puede combinarse en un mismo paso DATA con sentencias INFILE, SET o UPDATE.
- Si se usa en combinación con la sentencia BY, ésta debe aparecer inmediatamente de la sentencia MERGE. Además los SAS-FILES, deberán estar ordenados por los valores de las variables BY, que deberán ser las mismas en todos los SAS-FILES implicados.

Sintaxis:

```
MERGE sas-file [ ( opcion-sas-file ... ) ] ... opcion-merge ;
```

Utilidades:

- Combinación uno a uno: Se une la primera observación de un SAS-FILE con la primera del otro y así sucesivamente. Si coincide alguna variable, en el SAS-FILE creado, ésta tomará el valor que tomara en el último SAS-FILE especificado.

```
MERGE A B ;
```

- Si el número de observaciones de los SAS-FILES es distinto, la ejecución del paso DATA finaliza cuando el número de ejecuciones coincida con el mayor número de observaciones en un SAS-FILE.

- Si en uno de los conjuntos sólo hay una observación para los valores de las variables var ... , ésta se unirá longitudinalmente con todas las observaciones del otro conjunto con esos valores de variables var...

```
MERGE A B ;  
BY var ... ;
```

IV.3 OPCIONES DE LOS CONJUNTOS DE DATOS SAS®

Los conjuntos de datos SAS, tienen unas opciones que le son propias y que se pueden utilizar encerradas entre paréntesis al lado del SAS-FILE, siempre que éstos aparezcan, tanto en pasos DATA como en pasos PROC. Cabe destacar:

```
DATA sas-file ( opcion-sas-file ... ) ... ;  
SET sas-file ( opcion-sas-file ... ) ... ;  
MERGE sas-file ( opcion-sas-file ... ) ... ;  
PROC nombre-proc DATA = sas-file ( opcion-sas-file ... ) ... ;
```

OUTPUT = sas-file (opcion-sas-file ...);

opcion-sas-file puede ser:

KEEP = var ...

DROP = var ...

RENAME = (nombre-viejo = nombre-nuevo ...) No permitido listas de variables.

TYPE = tipo indica tipo de conjunto de datos: DATA, CORR, COVA...

usadas únicamente cuando se lee de un SAS-FILE ya existente.

WHERE = condicion

FIRSTOBS = n OBS = n no se pueden usar en combinación con la opción WHERE = o con una sentencia WHERE en el mismo paso DATA o PROC.

IN = var Sólo se puede utilizar en un paso DATA (Sentencias SET, MERGE, UPDATE). Se crea una variable lógica que toma el valor verdadero si se está procesando una observación del correspondiente SAS-FILE. Dicha variable no se añade al SAS-FILE que se crea.

Algunos ejemplos

Ejemplo A) En el fichero de entrada, el valor de la variable tipo indica si un registro de datos contiene información acerca de un profesor o acerca de un alumno.

Se crean dos SAS-FILES, uno con la información de los profesores y el otro con la información de los alumnos.

```
DATA      profesor(KEEP= curso profesor)
          alumno (DROP= tipo curso profesor);

INPUT  tipo  $1.  @ ;

IF TIPO = 'c' THEN DO;
          INPUT curso profesor $;
          OUTPUT profesor;
          END;

ELSE DO;
          INPUT nombre $ edad nota1-nota8;
          OUTPUT alumno;
          END;
```

V. SENTENCIAS DE USO GENERAL.

V.1 SENTENCIAS USADAS EN PASOS DATA Y PROC

-Algunas de las sentencias utilizadas en los pasos DATA, pueden también utilizarse en los pasos PROC.

-Las sentencias que se refieren a características de las variables u observaciones, si se especifican en un paso DATA, permanecen en los SAS-FILES creados o modificados de forma permanente. Si se especifican en un paso PROC, hacen referencia a la ejecución del paso PROC únicamente. Entre estas sentencias se encuentran:

-FORMAT, LABEL, WHERE, ...

Algunas sentencias de procedimientos.

Sentencias comunes a algunos procedimientos son:

-BY Se realiza el procedimiento para los valores de la(s) variable(s) especificadas en BY. Las observaciones del SAS-FILE que se analice deben estar ordenadas por los valores de la(s) variable(s) del BY. La ordenación de un SAS-FILE se puede realizar con el procedimiento SORT.

La sentencia BY también se puede utilizar en un paso DATA en combinación con una sentencia SET o MERGE o UPDATE.

-VAR Selecciona variables para analizar en el procedimiento.

-CLASS Indica variables de clasificación

-WEIGHT Especifica una variable de peso.

-FREQ "

La sintaxis de todas ellas es:

PALABRA especificacion-variable ... ;

V.2 SENTENCIAS DE USO GENERAL

Algunas sentencias son de uso general y no es necesario que se encuentren en un paso DATA o PROC. Pueden encontrarse en cualquier lugar del programa y tienen efecto hasta que aparezca otra del mismo tipo.

Especificación de títulos en la salida de resultados:

TITLE[n] 'título n-ésimo'; n <= 10

Especificación de notas a pies de página en la salida de resultados:

FOOTNOTE[n] 'nota a pie de página n-ésima'; n <= 10

Deasignación de referencias a librerías SAS:

LIBNAME ref CLEAR; (Elimina la identificación de la referencia ref)

Especificación de valores de no observación en los datos, correspondientes a variables numéricas:

MISSING carácter ... ;

carácter puede ser un valor alfabético o _

Especificación de **opciones SAS**, en general dependientes del sistema operativo:

OPTIONS opcion ... ;

Algunas opciones son:

NODATE no salida de la fecha en la salida de resultados

LS = n longitud de línea n en la salida de resultados

NOCENTER No centrar los títulos en la salida de resultados

ALTLOG = otro-salida-log Copia de la salida de diagnósticos a otra-salida-log,
que puede ser TERM, para terminal

FULLSTIMER Recolección de datos estadísticos de rendimiento del sistema.
Existen varias opciones de este tipo

Inclusión de parte de programa procedente de otro fichero:

%INCLUDE¹⁸ 'nombre-de-fichero' ;

Sentencia: RUN ;

Indica el fin de un paso DATA o un paso PROC. Se suele usar en el entorno interactivo o de ventanas.

Sentencia: QUIT ;

Se usa en el entorno interactivo o de ventanas, para indicar que las sentencias de un procedimiento que es interactivo, han finalizado.

¹⁸ Otra forma de uso de esta sentencia es:

%INCLUDE ref ... ;

ref puede ser:

-el primer nombre de un fichero externo de segundo nombre **sas**.

-una referencia a la que hay que asignar un fichero externo con la sentencia FILENAME.

También se usa para indicar el fin de un procedimiento gráfico.

Otras sentencias¹⁹

VI. UTILIZACIÓN DE PROCEDIMIENTOS SAS®.

VI.I. PROCEDIMIENTO FORMAT.

El procedimiento FORMAT:

- Crea formatos para que luego puedan ser asociados a variables, asociación realizada en un paso DATA o en un paso PROC, con la sentencia FORMAT.
- La creación de los formatos se puede realizar de forma temporal a la ejecución de un programa SAS, o de forma permanente, para luego ser utilizados por otros programas (sin necesidad de que se creen los formatos otra vez).
- Para crear un formato SAS de forma permanente basta especificar la opción LIB = LIBRARY en la sentencia PROC FORMAT. En este caso, debe crearse previamente la librería LIBRARY.
- El uso de un formato SAS permanente en otra sesión SAS, requiere que esté accedida la librería LIBRARY.

Sintaxis:

```
PROC FORMAT [ LIB = LIBRARY ] ;  
    VALUE [$]nombre especificacion-valor = 'descodificacion' ... ; ...
```

¹⁹ Identificación de referencias a ficheros externos a SAS:

```
FILENAME ref 'nombre-fichero' ;
```

Deasignación de referencias a ficheros:

```
FILENAME ref CLEAR ;(Elimina la identificación de la referencia ref)
```

Ejecución de comandos del sistema operativo:

```
X [ ] comando-sistema-operativo [ ] ;
```

Ir al entorno del sistema operativo:

```
X;
```

Para volver a la continuación de ejecución del programa SAS basta con pulsar ^D.

-Si el formato se asociará a una variable carácter, deben especificarse el dolar (\$) justo antes del nombre del formato y las comillas para encerrar especificaciones de valor.

-especificacion-valor puede ser:

val	indica el valor val.
vali-valf	indica desde valor vali hasta valor varf, ambos incluidos. vali puede ser LOW para indicar el mínimo de los valores posibles y vali puede ser HIGH, para indicar el máximo.
vali<-valf	... excluyendo vali. Valf puede ser HIGH.
vali-<valf	... o excluyendo valf. Vali puede ser LOW.
varias combinaciones anteriores, separadas por comas (,).	
en el caso de formatos para variables carácter, se han de especificar los valores encerrados en comillas (').	
other	indica cualquier otro valor no especificado en las especificaciones anteriores.

Ejemplo:

```
PROC FORMAT;  
  VALUE  sino  1 = 'si'  2 = 'no';  
  VALUE  $sexo 'm' = 'mujer' 'v' = 'varon';  
  VALUE  nota  LOW-5 = 'suspense'  5<-7 = 'aprobado'  
          7<-HIGH = 'excelente';
```

VI.II. OTROS PROCEDIMIENTOS

PROCEDIMIENTO PRINT.

```
PROC PRINT [ DATA = sas-file ] ;  
[ VAR especificacion-variable ... ; ]  
[ BY especificacion-variable ... ; ]  
[ SUM especificacion-variable ... ; ]
```

PROCEDIMIENTO SORT.

```
PROC SORT [ DATA = sas-file ] ;  
BY [ DESCENDING ] especificacion-variable ... ;
```

En general, orden ascendente de SAS: _, ., a-z, -n, 0, +n siendo n un número.

PROCEDIMIENTO CONTENTS.

```
PROC CONTENTS [ DATA = sas-file ] [ POSITION ] ;
```

Muestra información sobre el sas-file. La opción POSITION hace que las variables aparezcan en el orden del sas-file y no en orden alfabético.

PROCEDIMIENTO SQL.

Procedimiento que utiliza el lenguaje SQL para la creación o manipulación de tablas, vistas, etc...

PROCEDIMIENTO MEANS.

```
PROC MEANS [ DATA = sas-file ] opcion... ;  
[ VAR especificacion-variable ... ; ]  
[ CLASS especificacion-variable ... ; ]  
[ WEIGHT especificacion-variable ... ; ]  
[ FREQ especificacion-variable ... ; ]  
[ BY especificacion-variable ... ; ]  
[ OUTPUT OUT = sas-file opcion ... ; ]
```

PROCEDIMIENTO TTEST (V8).

```
PROC TTEST [ DATA = sas-file ] opcion... ;  
[ VAR especificacion-variables ... ; ]  
[ CLASS especificacion-variable ... ; ]  
[ PAIRED especificacion-variables ... ; ]  
[ WEIGHT especificacion-variable ... ; ]  
[ FREQ especificacion-variable ... ; ]  
[ BY especificacion-variables ... ; ]
```

PROC TTEST < options > ;

The following options can appear in the PROC TTEST statement.

- [ALPHA=p](#) /* Intervalos confidenciales 100(1-p)% ; Por defecto p = 0.05 */
- [CI=EQUAL](#)
- [COCHRAN](#)
- [DATA=SAS-data-set](#)
- [H0=m](#)

CI=EQUAL

CI=UMPU

CI=NONE

The CI= option specifies whether a confidence interval is displayed for and, if so, what kind. The CI=EQUAL option specifies an equal tailed confidence interval, and it is the default. The CI=UMPU option specifies an interval based on the uniformly most powerful unbiased test of . The CI=NONE option requests that no confidence interval be displayed for . The values EQUAL and UMPU together request that both types of confidence intervals be displayed. If the value NONE is specified with one or both of the values EQUAL and UMPU, NONE takes precedence

Ejemplo:

```
proc ttest;  
var v1 ;  
class SEXO;
```

The TTEST Procedure

Statistics

Variable	Class	N	Lower CL		Upper CL		Lower CL		Upper CL	
			Mean	Mean	Mean	Std Dev	Std Dev	Std Dev	Std Err	
V1	hombre	138	182.06	183.85	185.65	9.5293	10.655	12.086	0.907	
V1	mujer	92	173.11	175.58	178.06	10.44	11.953	13.983	1.2462	
V1	Diff (1-2)		5.3044	8.2725	11.241	10.252	11.191	12.322	1.5063	

T-Tests

Variable	Method	Variances	DF	t Value	Pr > t
V1	Pooled	Equal	228	5.49	<.0001
V1	Satterthwaite	Unequal	179	5.37	<.0001

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
V1	Folded F	91	137	1.26	0.2224

```
proc ttest CI= NONE;  
var v1 ;  
class SEXO;
```

No muestra en salida los intervalos de confianza.

```
data varon;  
set duda;  
if sexo='hombre';  
proc ttest;  
var v1 ;  
run;
```

Variable	N	Lower CL		Upper CL		Lower CL		Upper CL	
		Mean	Mean	Mean	Std Dev	Std Dev	Std Dev	Std Err	
V1	138	182.06	183.85	185.65	9.5293	10.655	12.086	0.907	

T-Tests

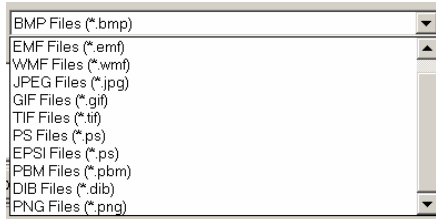
Variable	DF	t Value	Pr > t
V1	137	202.69	<.0001

PROCEDIMIENTOS FREQ, PLOT, CHART, UNIVARIATE, TRANSPOSE

VII.III. GRAFICOS EN SAS

Los gráficos producidos en sas, por defecto se muestran en el dispositivo de salida de SAS en la ventana Graph.

Se pueden almacenar gráficos en ficheros con otros formatos. Esto se puede realizar desde la ventana Graph mediante la opción de menú File → Export as Image y seleccionar alguno de los formatos que aparecen en la figura.



o bien mediante código sas con la sentencia `goptions` especificando el dispositivo gráfico y el fichero antes del procedimiento que produce el gráfico.

```
FILENAME referencia-fichero 'unidad-directorio-fichero';
```

```
GOPTIONS DEVICE=dispositivo GFSNAME=referencia-fichero GFSMODE=replace  
[otras-opciones];
```

Dispositivos pueden ser: `psepsf` para encapsulado postcript, `cljpsa4` para postcript, `gif` para formato gif, , etc, etc...

Otras-opciones incluye:

```
ROTATE=LANDSCAPE
```

La calidad de los gráficos de los dispositivos de sas no es óptima salvo para el caso del propio dispositivo de sas y los formatos postcript y encapsulado postcript.

Ejemplo:

```
FILENAME file1 'c:\graficos\ejemplo.eps';  
GOPTIONS DEVICE=psepsf GFSNAME=file1 GFSMODE=replace ;  
  
PROC GPLOT DATA=ejer;  
plot pulso2*pulso1=sexo;
```

En el ejemplo, el gráfico se muestra en la ventana graph y se almacena con formato encapsulado postcript en el fichero `c:\graficos\ejemplo.eps`.

Antes de obtener un nuevo gráfico se deben especificar de nuevo las sentencias `filename` y `goptions` correspondientes. Si se desea eliminar la información de un dispositivo específico, se debe especificar la sentencia:

```
GOPTIONS RESET=ALL;
```

Es importante, para evitar problemas de que sas se quede enganchado con un procedimiento gráfico, que después de cada procedimiento que realiza un gráfico se añadan las dos sentencias:

```
RUN;  
QUIT;
```

Algunos procedimientos estadísticos pueden producir gráficos. Para ello hay que consultar la sintaxis de los procedimientos y/o la documentación de los mismos.

Ejemplo

```
proc reg data= ejer ;  
model pulso2=pulso1;  
plot pulso2*pulso1 / pred ; * plot pulso2*pulso1;  
plot r.*p.;  
run; quit;
```

SAS/GRAPH consta de procedimientos gráficos.

Ejemplo

```
proc gplot data=ejer;  
plot pulso2*pulso1=sexo;  
run; quit;  
  
proc gchart data=ejer;  
hbar pulso1;  
vbar pulso1 /group=sexo;  
run; quit;
```

También existen macros que vienen con el producto que se pueden utilizar para producir gráficos, por ejemplo, la macro plotit realiza gráficos de puntos en un plan X, Y. Permite etiquetar los puntos, dar diferentes tamaños / colores dependiendo de una tercera variable, etc... Uso particular de la misma es la representación de los factores en las técnicas multivariantes factoriales.

Ejemplo

```
*--- Analysis de Correspondencias Simples ---;  
proc corresp all data=cars outc=coor;  
tables marital, origin;  
  
%plotit(data=coor, datatype=corresp)
```

Ejemplo

```
%plotit(data=FACT2, datatype=, plotvars=factor2 factor1,  
LABELVAR=IDENT, symvar=., symsize=0.4,PAINT=crel_eje1_punto )
```

En este ejemplo, el sas-file FACT2 no es un sasfile producido por un procedimiento anterior.

VII. UTILIDADES DE LA PROGRAMACIÓN SAS® E INTERACCIÓN CON OTROS PAQUETES.

VII.1 (AVANZADO) PROGRAMACIÓN CON MACROS

Una *macro* es un conjunto de sentencias *macro* y/o sentencias SAS, que puede ser referenciada por otras partes del programa. Se permite, asimismo, la especificación de argumentos. Dichos argumentos se llaman *macro-variables*.

Veamos un escueto resumen:

Creación de *macros*.

Se realiza mediante:

```
%MACRO nombre-macro [( par1 [, par2 ... ] )] ;  
(Sentencias macro y/o sentencias SAS)  
%MEND [ nombre-macro ] ;
```

-Los parámetros de la *macro* son *macro variables locales*, que tienen validez únicamente en esa *macro*.

-La referencia dentro de la *macro* a parámetros de la misma, requiere un ampersand (&) justo delante de los mismos.

***Macro variables*.**

Las *macro variables* son variables que transfieren su valor a lo largo de varios pasos DATA o pasos PROC.

Existen *macro variables globales* y *macro variables locales*. Las *macro variables globales* tienen valor en un programa desde su creación. Las *macro variables locales* tienen valor únicamente durante la ejecución de una *macro*.

Creación de *macro variables*

Se pueden crear en cualquier lugar de un programa SAS con la sentencia:

```
%LET macro-variable [= valor-inicial ] ;
```

Si una *macro variable* se crea dentro de una *macro*, es una *macro variable local* a esa *macro*.

Si el valor que se quiere asignar a una *macro variable* es un conjunto de sentencias SAS, esto se realiza mediante el uso de %STR:

```
%LET macro-variable = %STR ( sentencias SAS );
```

Referencia a *macro variables*.

La referencia de *macro variables* se hace siempre con un & delante de su nombre.

Llamada de *macros*.

```
%nombre-macro [ ( val1 [, val2...] ) ] ;
```

La llamada a una *macro* debe encontrarse después de su creación.

Se pueden hacer llamadas a *macros* dentro de la definición de una *macro*.

Ejemplo.

```
%MACRO unavar (datos, x) ;  
PROC GLM DATA = &datos ;  
CLASS sexo ;  
MODEL &x = sexo ;  
%MEND unavar ;  
  
%unavar (ejercici, pulso) ;
```

Sentencias *macro* en la definición de una *macro*.

En la definición de una *macro* se pueden especificar las siguientes sentencias *macro*:

```
%DO macro-variable = num1 %TO num2 [ %BY num3 ] ;  
(Sentencias macro y/o sentencias SAS)  
%END ;
```

En la sentencia %DO, la *macro-variable*, que se crea, no lleva &. Dentro del bucle, para referirse a la *macro-variable*, debe especificarse *¯o-variable*.

```
%DO [ %WHILE | %UNTIL ] condición ;  
(Sentencias macro y/o sentencias SAS)  
%END ;
```

```
%IF condición %THEN accion; [ %ELSE accion2 ; ]
```

```
%GOTO etiqueta ;
```

```
%etiqueta : [ sentencia ] ;
```

```
%STR ( parte de sentencias SAS y/o sentencias SAS ) ;
```

Para especificar una cadena de caracteres.

```
%INPUT macro-variable ;
```

Para entrada de datos del exterior.

Sentencias *macro* en la definición de una *macro* o externas a la definición.

```
%LET macro-variable [ = valor-inicial ] ;
```

Crea una *macro variable* a la que da un determinado valor inicial. Si se especifica la sentencia fuera de cualquier *macro*, tiene validez para todo el programa SAS. En caso, contrario, para la *macro* en que se ha definido.

```
%PUT texto;
```

Permite la escritura de un texto.

Ejemplo.

```
%LET num = 30;
%MACRO TOTAL ;
%DO i = 1 %TO &num ;
    %INPUT cen ;
    PROC TABULATE DATA = &cen ;
        (Sentencias SAS que terminen de especificar un programa)
%END ;
%MEND ;

%TOTAL ; /* se realiza la ejecución de la macro */
```

Combinación de *macro variables*:

Combinación de los valores de dos *macro variables*:

```
&macro-variable1&macro-variable2
```

Combinación de los valores de dos *macro variables* con intercalación de otro nombre, digamos X:

```
&macro-variable1.X&macro-variable2
```

Creación de *macro variables* con valores de un paso data:

```
Call SYMPUT ('macro-variable', variable);
```

Se crea la *macro-variable*, si no existía, y se le da el valor de *variable*
Esta sentencia no es necesario referenciarla en la definición de una *macro*.

Asignación de valores de una *macro variable* a variables de un paso DATA:

Se realiza mediante la llamada a la función:

SYMGET('macro-variable');

Esta sentencia no es necesario referenciarla en la definición de una *macro*.

VII.2 INTERACCIÓN ENTRE BMDP Y SAS® Y ENTRE SPSS Y SAS®

Esta interacción no está permitida en todos los entornos. A modo de muestra presentamos la conversión de un SPSS-FILE a un SAS-FILE.

Conversión de un BMDP-FILE / SPSS-FILE a un SAS-FILE. Procedimiento CONVERT.

- Existe la posibilidad de analizar conjuntos de datos BMDP (BMDP-FILES) o SPSS (SPSS-FILES) en SAS, previa conversión de los mismos a SAS-FILES.
- Esta conversión se realiza mediante el procedimiento CONVERT.
- Previamente se debe asociar una referencia con el nombre del fichero que contiene el BMDP-FILE o el SPSS-FILE con la sentencia FILENAME.
- Para un fichero spss (que debe exportarse en formato portable (.por) en SPSS), se deben incluir las siguientes sentencias:
 FILENAME referencia 'fichero-que-contiene-el-fichero SPSS';
 PROC CONVERT SPSS = referencia [OUT = fich2];
- De igual forma para un fichero bmdp.
- Alternativamente²⁰, también se puede realizar, especificando la sentencia LIBNAME y asociando una librería (nombre-librería) de una *engine* SPSS ó BMDP con el fichero que contiene el BMDP-FILE o el SPSS-FILE y el directorio en que se encuentra. Esta referencia hace que se pueda utilizar el fichero sas-file *nombre-librería._first_*.

Ejemplo:

```
LIBNAME nombre-libreria BMDP 'camino completo fichero';  
PROC PRINT DATA = nombre-libreria._first_;
```

VII.3 EXPORTACIÓN/IMPORTACIÓN DE FICHEROS SAS® A OTRAS PLATAFORMAS

Esto se realiza con los procedimientos **CPORT** y **CIMPORT**.

²⁰ A partir de la version 8.0, se recomienda la forma alternativa anterior

APÉNDICE I (Platón. Unix)

Ejecución de un programa SAS®.

Ejecución en modo no interactivo (opción recomendada):

La ejecución de forma no interactiva requiere la inclusión del programa SAS en un fichero. Si el segundo nombre es **sas**, la ejecución se realiza mediante: **sas** primer-nombre. De dicha ejecución se obtienen los ficheros *primer-nombre.lst* y *primer-nombre.log*, que incluyen respectivamente la salida de resultados si existe y la salida de diagnósticos.

Si el segundo nombre del fichero es distinto de *sas*, la ejecución se realiza mediante: **sas** nombre-fichero. De dicha ejecución se obtienen los ficheros *nombre-fichero.lst* y *nombre-fichero.log*, que incluyen respectivamente la salida de resultados si existe y la salida de diagnósticos.

Ejecución interactiva o en modo ventanas:

Sesión X: **sas**

 en el caso de que no sea una terminal X, debe habilitarse la sesión para habilitarla. Esto se hace con el comando Unix: `setenv DISPLAY`

Modo texto: **sas -fsdevice ascii**

Si la ejecución es de forma interactiva, se incluirá el programa en la ventana *editor*. La salida de resultados, si existe, se incluye en la ventana OUTPUT y la salida de diagnósticos en la ventana LOG.

Directorio de ejemplos:

`/usr/local/SAS_8.2/samples`

APÉNDICE II. NOVEDADES Y ENTORNO GRÁFICO DE SAS V8

A.II.1. Interfaz de usuario

Véase Capítulo I

A.II.2. Novedades de la versión SAS V8

- Interfaz de usuario similar a la interfaz general de usuario Windows, con barra de ventanas e iconos de edición similares a los de los programas Windows.
- Se permiten nombres más largos, hasta de 32 caracteres.
- Los nombres de variables permiten mayúsculas y minúsculas.
- Se permite mayor longitud para variables de tipo carácter (32K).
- Ventana *Explorer* y *Results*
- Generación de resultados en formato HTML
- Ayuda en formato HTML
- Ventana *Editor*: Editor mejorado:
 - El código permanece en el editor después de ser ejecutado.
 - Las secciones de código se pueden expandir y colapsar.
- Las extensiones de los ficheros han sido modificadas:

	Version 6 Unix	Version 6 Windows	Versión 8
DATA	.ssd01	.sd2	.sas7bdat
VIEW	.ssv01	.sv2	.sas7bview
CATALOG	.sct01	.sc2	.sas7bcatalog
PROGRAM	.ssp01	.sp2	.sas7bpqm
MDDB	.ssm01	.sm2	.sas7bmdb

- El procedimiento **FORMAT**:
DECSEP='Separador' permite especificar el carácter para separar decimales.
En la sentencia VALUE, MULTILABEL permite múltiples valores para un rango dado y permite que se solapen rangos.
- El procedimiento **MEANS**:
Nuevas estadísticas: Mediana, P1, P5, P10, P90, P95, P99, Q1, Q3, Qrange
La sentencia TYPE permite especificar las combinaciones de las variables de clase
La sentencia WAYS: especifica uno o más enteros que define el número de variables de clasificación a combinar para formar todas las combinaciones posibles de esas variables de clasificación
Permite especificar múltiples sentencias CLASS
Nuevas opciones de salida: AUTONAME, INHERIT, IDGROUP MIN|MAX

La sentencia VAR soporta la opción WEIGHT

Ejemplo

```
proc means data=sasuser.class
  class Sex Age;
  type Sex*Age;

proc means data=airline.fltaten;
  class State JobClass GenderCode;
  ways 1 2;
run;
```

- El procedimiento **TABULATE**:
Permite como nueva opción OUT='conjunto de datos SAS'.
Admite múltiples instrucciones VAR y múltiples instrucciones CLASS.
Admite las nuevas estadísticas del procedimiento MEANS.
Permite nuevos comandos a para crear porcentajes: de informe, de columnas, de filas y de página.
- Nuevos procedimientos

El procedimiento **IMPORT**: Genera un conjunto de datos SAS a partir de:
Fichero plano con delimitadores.
Hoja de cálculo o tabla de Microsoft En este caso, se requiere el módulo SAS /ACCESS PCFile Formats.

El procedimiento **EXPORT**:
Exporta un conjunto de datos SAS a las fuentes anteriores

NOTA: La importación/exportación a otros formatos de datos puede realizarse fácilmente desde la interfaz de usuario (véase **II.5**).

El procedimiento **REGISTRY**:
Gestiona el entorno de la sesión SAS

- El módulo **SAS/GRAPH** incorpora las siguientes novedades:

Crear gráficos en tres dimensiones.
Etiqueta estadísticos en gráficos de barras horizontales y verticales.

A.II.3. Novedades estadísticas de la versión SAS V8

A.II.4. Documentación de SAS V8

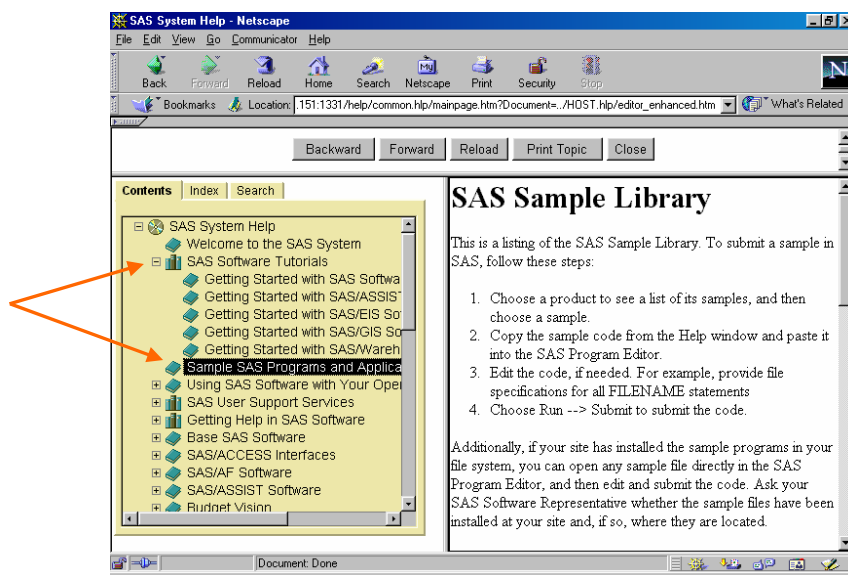
Los manuales de SAS se encuentran a disposición de los usuarios de la UCM, en formato PDF, en la página web del software científico de la Universidad Complutense de Madrid.

<http://www.sisoft.ucm.es/>

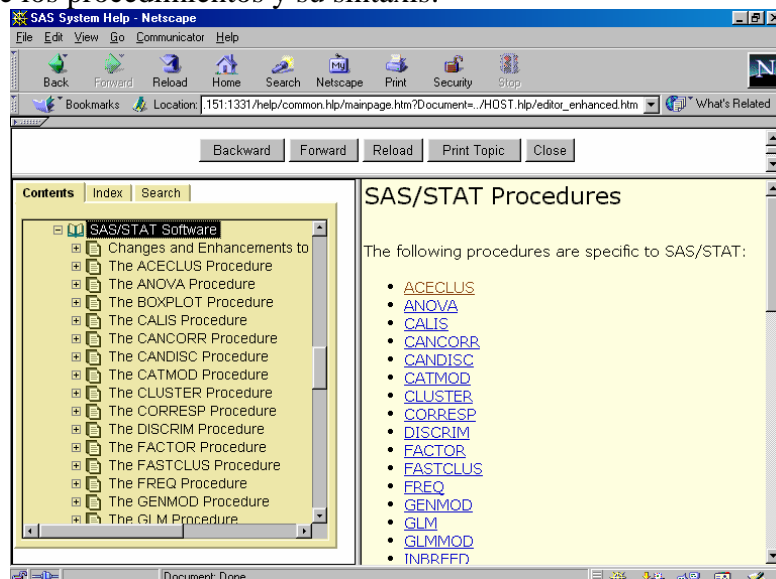
Y seguidamente activar la pestaña: documentación

A.II.5. Ayuda de SAS® V8

En el programa SAS existe una ayuda bastante extensa. Para acceder a la ayuda hay que teclear el icono de ayuda que está simbolizado por un libro. La ayuda incluye información acerca de la sintaxis de SAS e información acerca de los procedimientos y su sintaxis. La ayuda incluye además ejemplos de programas en SAS.



Ayudas acerca de los procedimientos y su sintaxis:



A.II.6 Ventana Explorer

La ventana *explorer* permite ver y gestionar las librerías y ficheros SAS y carpetas (shortcut) de ficheros no SAS. Permite crear librerías y ficheros sas, abrir, copiar, mover, borrar ficheros, etc... Para visualizar la pantalla *explorer* si se ha cerrado previamente: Menu View/ Opción Contents only.

Una librería de SAS se asocia a un directorio y contiene tablas o sas-files (además de Queries y Catálogos que exceden el contenido de esta guía).

Las librerías inicialmente contenidas en SAS son:

SASHELP Contiene SAS-FILES de ejemplo que son utilizados en los ejemplos de ayuda del programa SAS.

MAPS Contiene SAS-FILES de ejemplo acerca de países

SASUSER

WORK Se incluyen los SAS-FILES temporales a la ejecución del programa

SAS.

Las operaciones realizadas desde la ventana *explorer* no generan código SAS.

Algunas utilidades de la ventana *explorer*:

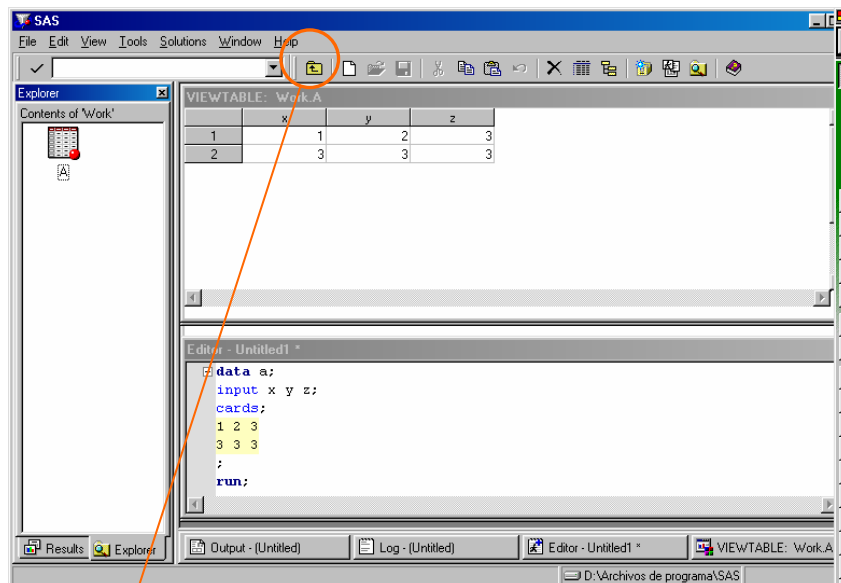
Navegar por las distintas librerías y los distintos sas-files

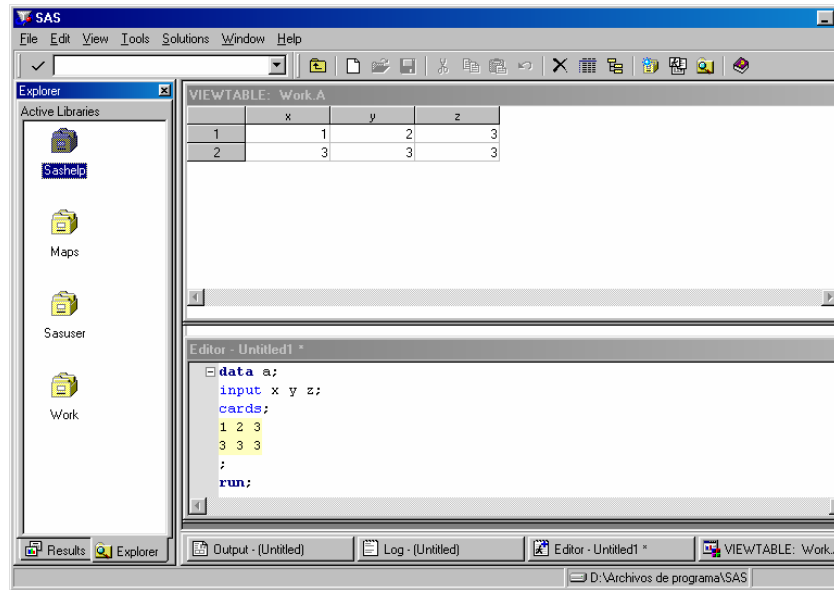
Ver el contenido de los SAS-files y en modo edición cambiar su contenido

Crear nuevas librerías y nuevos SAS-Files y catálogos de estas librerías.

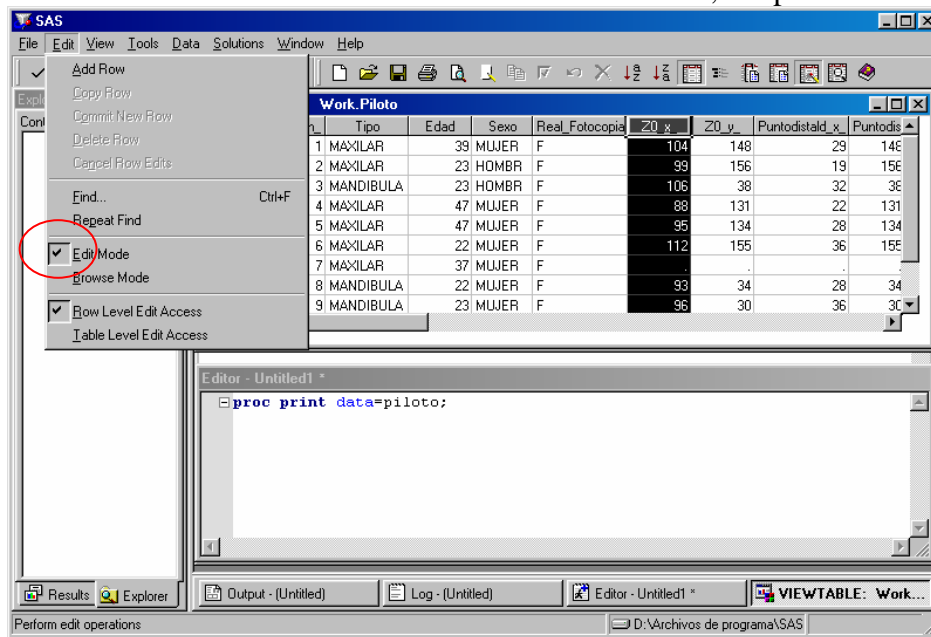
Copiar, mover, borrar librerías, sas-files o ficheros, etc...

Se puede navegar por el conjunto de librerías y SAS-FILES:

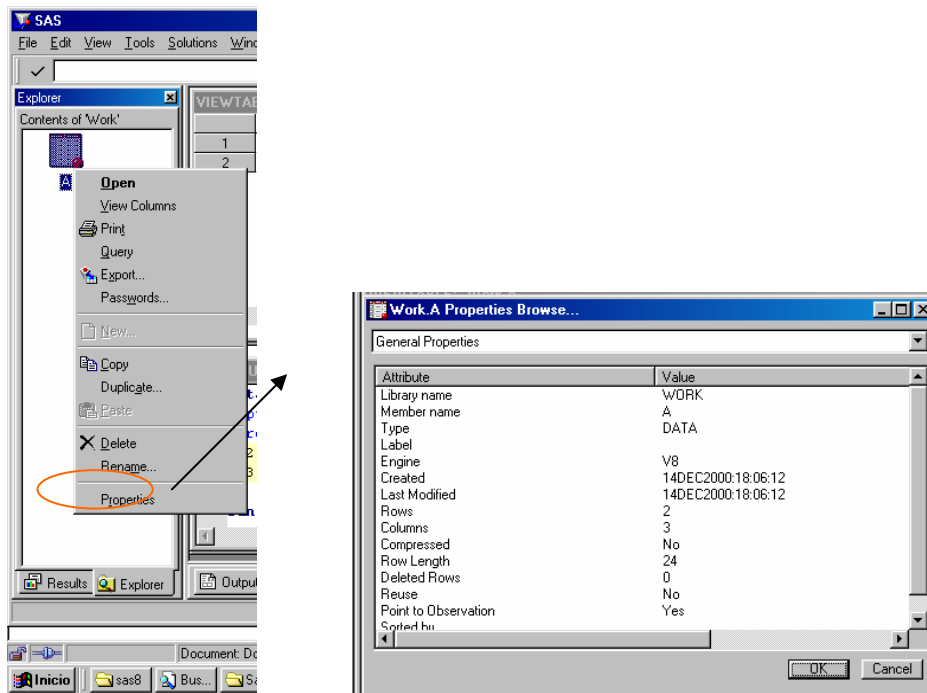




Se puede ver el contenido de un SAS-FILE (con la utilidad VIEWTABLE) dando doble click sobre él. Para modificar datos se debe seleccionar en el menu Edit, la opcion Edit Mode:



Se pueden ver las propiedades de los SAS-FILES:



A.II.7. Creación de Librerías de datos SAS®

Véase II.2

A.II.8. Interfaz de ayuda para la entrada de datos de ficheros ASCII

Véase II.B

A.II.9. Introducción de datos en SAS®

Véase II.D

APÉNDICE III. (AVANZADO) SAS-FILES Y SAS-ENTERPRISE GUIDE

Nota Importante

Si no se desean perder los accesos a los perfiles de usuario de SAS y SAS-Enterprise Guide, no deben tenerse abiertas dos sesiones simultáneamente de estos dos programas. El perfil de usuario contiene información acerca de los accesos a librerías permanentes.

SAS-Files creados en SAS®

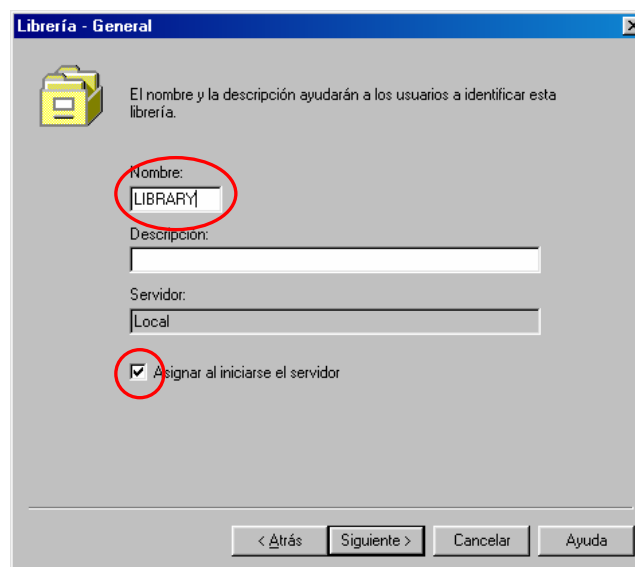
Los SAS-files permanentes creados con SAS pueden leerse en SAS-Enterprise Guide. Si los SAS-Files no tienen asociados formatos (permanentes), el acceso es directo. Si las variables

tienen formatos permanentes éstos se encuentran en la librería LIBRARY²¹ (véase creación de formatos en esta guía). Esta librería no está por defecto accesible en SAS-Enterprise Guide. Se pueden hacer dos cosas alternativas:

- Copiar los formatos en Windows desde el directorio asociado a la librería LIBRARY hasta el directorio asociado a la librería SASUSER (por defecto²², es el directorio \Mis documentos\My SAS Files\V8 en la unidad donde se encuentra instalado SAS).
- Asociar la librería LIBRARY al directorio correspondiente con el programa administrador asociado a Enterprise Guide. Esto se realiza de la siguiente forma:

Asignación de librerías en SAS-Enterprise Guide, desde el programa Administrador

Para crear una librería en SAS-Enterprise Guide, se puede hacer desde el programa Administrador asociado al Eguide, con la opción de menú Archivo → Nuevo → Librería



Si se selecciona la opción Asignar al iniciarse el servidor, esta librería quedará accesible siempre que se ejecute SAS-Enterprise Guide.

SAS-Files creados en SAS[®] -Enterprise Guide

Los SAS-files permanentes creados con SAS-Enterprise Guide pueden leerse en SAS. Si los SAS-Files no tienen asociados formatos (permanentes), el acceso es directo. Si las variables tienen formatos permanentes, éstos se encuentran en la librería SASUSER²³. Para que SAS

²¹ Según se explica en este manual. Se pueden estructurar los formatos en otras librerías.

²² Se puede averiguar el directorio asociado con la librería SASUSER, pulsando el botón derecho del ratón y seleccionando propiedades.

²³ Por defecto, la librería SASUSER se asocia al directorio \Mis documentos\My SAS Files\V8 en la unidad donde se encuentra instalado SAS. Se puede averiguar el directorio asociado con la librería SASUSER, pulsando el botón derecho del ratón y seleccionando propiedades.

encuentre los formatos de esta librería se debe especificar la sentencia `OPTIONS` en el programa SAS antes de acceder a los SAS-files. Por ejemplo:

```
OPTIONS FMTSEARCH = ( WORK LIBRARY SASUSER);
```

La opción `FMTSEARCH` especifica las librerías donde se buscan los formatos permanentes de las variables de los SAS-files.

NOTA: por defecto el valor de `FMTSEARCH` es `(WORK LIBRARY)`