



Servei d'Estadística
Universitat Autònoma de Barcelona

Manual de Introducción a SAS System[®] v9.4

Llorenç Badiella. Director del Servei d'Estadística Aplicada
Anabel Blasco. Asesora estadística del Servei d'Estadística Aplicada
Ester Boixadera. Asesora estadística del Servei d'Estadística Aplicada
Oliver Valero. Asesor estadístico del Servei d'Estadística Aplicada
Ana Vázquez. Asesora estadística del Servei d'Estadística Aplicada

Manual de Introducción a
SAS System® v9.4

Servei d'Estadística Aplicada
Universitat Autònoma de Barcelona

C5b/111, Facultat de Ciències, Campus UAB
08193 Cerdanyola del Vallès
(Barcelona)
Tel. 93.581.13.47
s.estadistica@uab.es
www.uab.es/s-estadistica/

Publicado por el Servei d'Estadística Aplicada de la UAB

Octubre 2022.

Este documento puede ser copiado y libremente distribuido, siempre y cuando sea preservada su integridad, referenciado su origen y comunicado su uso al Servei d'Estadística Aplicada de la UAB. No está permitido añadir, borrar o cambiar ninguna de sus partes, o extraer páginas para su uso en otros documentos.

CONTENIDOS

1	PRESENTACIÓN	5
1.1	SAS Enterprise Guide y la sintaxis de SAS System	6
2	SAS SYSTEM.....	8
2.1	Inicio de la sesión.....	8
2.2	El código SAS.....	10
2.3	Los procedimientos y el paso data.....	11
2.4	Introducción de datos	12
2.5	Importar datos.....	14
2.6	Importar datos mediante el asistente.....	16
2.7	Explorar los contenidos de un dataset.....	17
2.8	La instrucción SET del paso data	18
2.9	Almacenar datos SAS.....	19
2.10	Explorar Librerías.....	22
2.11	Explorar Datos.....	23
2.12	Ordenar datos.....	24
2.13	Fundir archivos: Añadir casos	25
2.14	Fundir archivos: Añadir variables	27
2.15	Definición de etiquetas	29
2.16	Exportar datos.....	31
2.17	Crear nuevas variables.....	32
2.18	Filtrar casos	34
2.19	Funciones de SAS.....	35
2.20	Procedimientos SAS.....	37
2.21	Resumir datos.....	40
2.22	Tablas de frecuencia: Proc Freq.....	42
2.23	Tablas de contingencia: Proc Freq.....	43
2.24	Tablas de estadísticos de resumen	46
2.25	Análisis Univariante.....	50

1 PRESENTACIÓN

Este manual de introducción a **SAS System** para Windows pretende ser una primera aproximación al uso de uno de los programas del software SAS para aquellas personas con algunos conocimientos de estadística que quieran iniciarse en la programación estadística mediante este programa. Este manual sólo trata aquellas instrucciones más básicas y sus opciones más habituales. Es altamente recomendable haber tenido contactos previos con otros programas de análisis y manipulación de datos, hojas de cálculo, bases de datos y lenguajes de programación. Por otro lado, se da por supuesto que el usuario ya tiene experiencia suficiente en el entorno Windows.

Cada sección consta de diferentes apartados: leer datos, tabular los datos, etc. pero en cada uno de ellos se introducen a modo de observaciones diversas características generales con el objetivo de remarcar otros detalles necesarios del funcionamiento del programa. Por este motivo es recomendable seguir el manual de forma lineal y consecutiva.

Para seguir los ejemplos propuestos, es necesario almacenar los ficheros del curso en el directorio de trabajo, siendo "C:\Temp\" el directorio de trabajo utilizado en el presente manual.

1.1 SAS Enterprise Guide y la sintaxis de SAS System

El usuario de **SAS Enterprise Guide**, puede emplear este manual de introducción a la sintaxis de **SAS System** para complementar y ampliar las posibilidades que ofrece el programa.

Determinadas tareas que se pueden realizar con **SAS EG** se pueden optimizar si se emplea código. Además, cuando el usuario desee reproducir una serie de análisis o pasos de manipulación, encontrará muy útil el uso de la sintaxis.

SAS EG incorpora una tarea específica llamada **New Program** donde el usuario puede escribir el código **SAS**.



Finalmente, el usuario de **SAS EG**, también se encontrará que en ocasiones necesita incorporar a una determinada tarea de **SAS EG** alguna opción muy especial. En tal caso, necesitará añadir sintaxis directamente a la tarea.

En general, las tareas tienen una casilla de **Preview Code**, y al ser activada aparece la ventana siguiente:

```

Code Preview for Task
Insert Code...

/* -----
Code generated by SAS Task

Generated on: Monday, November 09, 2015 at 6:43:27 PM
By task: Sort Data

Input Data: Local:CURSO.GALLETAS
Server: Local
-----

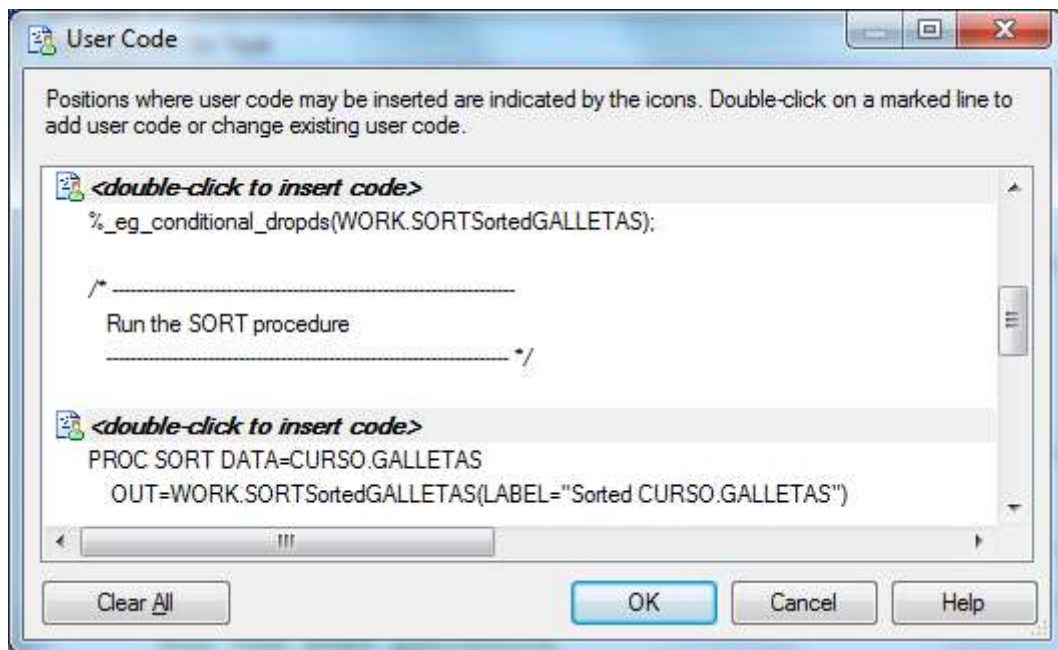
% eg_conditional_dropds (WORK.SORTSortedGALLETAS);

/* -----
Run the SORT procedure
-----
PROC SORT DATA=CURSO.GALLETAS
  OUT=WORK.SORTSortedGALLETAS (LABEL="Sorted CURSO.GALLETAS")
  ;
  BY ID TIPO MARCA ENMASCARAMIENTO;

RUN;

```

Tras pulsar **Insert Code**, aparece la ventana con el código de la tarea y diferentes posibilidades para insertar nuevas opciones de código.



2 SAS SYSTEM

2.1 Inicio de la sesión

El programa SAS básicamente se encuentra dividido en dos grandes ventanas:

Ventana izquierda “Explorer”: contiene accesos directos a los ficheros que interesen, información sobre las librerías y una ventana de resultados donde aparece de forma desglosada la información resultante de los procedimientos ejecutados.

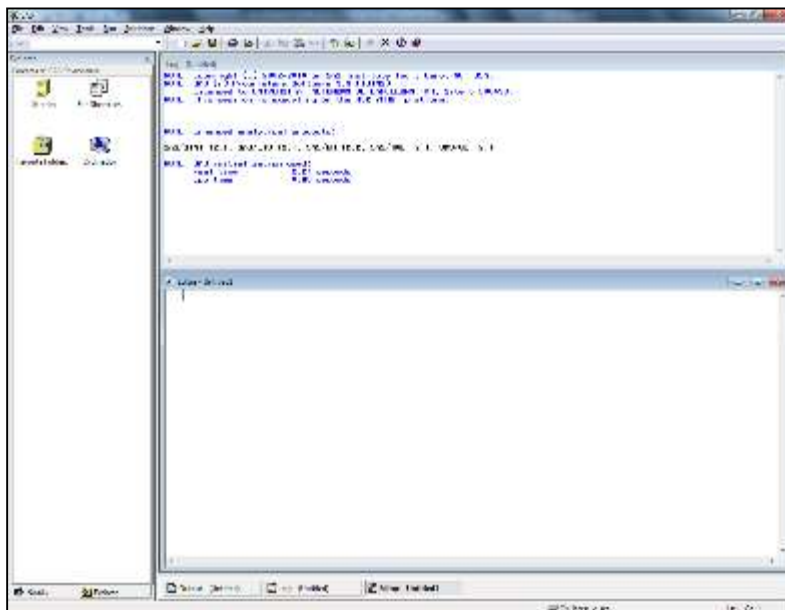
Ventana derecha: Contiene las ventanas de trabajo principales **LOG**, **OUTPUT** y **EDITOR**.

Ventanas de trabajo:


Ventana EDITOR: Esta ventana corresponde a la ventana de sintaxis, por lo tanto es editable.

Ventana LOG: En esta ventana se consulta y revisa todo lo que se ha ejecutado, aparecen mensajes de advertencia y de error en caso necesario y se informa sobre la velocidad de ejecución y recursos.

Ventana OUTPUT: Cuando se ejecutan procedimientos de SAS, en esta ventana se muestran los listados, tablas y/o resultados. En esta nueva versión SAS, se puede seleccionar que se presenten los resultados en formato .html (**RESULTS VIEWER**) en vez de la pantalla de código ASCII.



Ventana inicial de SAS.

Para borrar el contenido de la pantalla activa se puede pulsar el botón:  .

Para guardar el contenido de la pantalla activa se puede pulsar el botón:  .

Para consultar la ayuda de **SAS** se puede pulsar el botón:  .



Ventanas de diálogo desplegables: **SAS** es un programa adaptado para trabajar bajo Windows, de forma que la mayoría de ventanas desplegables tienen básicamente las mismas opciones que en cualquier otro programa para Windows:

FILE:	abrir, guardar, imprimir, ...
EDIT:	deshacer, copiar, pegar, borrar, buscar, reemplazar, ...
VIEW:	para dirigirse a determinada ventana de trabajo
TOOLS:	TABLE EDITOR (permite explorar bases de datos en formato SAS).
RUN:	SUBMIT (ejecutar), RECALL (pega la sintaxis de la última ejecución en la ventana de sintaxis).
SOLUTIONS:	ANALYSIS → GUIDE DATA ANALYSIS (aparece una ventana interactiva para el análisis de datos).
WINDOW:	permite cambiar de ventana (LOG, OUTPUT, EDITOR,...).
HELP:	V9 → SAS Help and Documentation. V8 → SAS SYSTEM HELP (ayuda de SAS), Books and training -> SAS Online Doc (Manual completo pero muy extenso de SAS).

2.2 El código SAS

Los ficheros de código SAS tienen la extensión ".SAS".

El código se puede visualizar y modificar en la ventana Editor.

Para poder ejecutar la sintaxis, se debe pulsar el botón: . Para ejecutar una parte de la sintaxis, primero se selecciona dicha parte y después se pulsa el botón. Si se desea interrumpir la ejecución del programa, se puede pulsar el botón: .

Una instrucción corresponde a una o varias líneas de código y termina con el símbolo ";". Cada instrucción puede tener varias opciones que se especifican siempre antes del símbolo ";".

Los comentarios son instrucciones que vienen precedidas por un signo "*".

Usualmente, las primeras instrucciones de un fichero de sintaxis son instrucciones de sistema que permiten controlar las ventanas OUTPUT o LOG y definen opciones generales:

Las instrucciones **TITLE**, **FOOTNOTE** y **OPTIONS** controlan la ventana **OUTPUT** o bien en los ficheros generados automáticamente con los resultados. Las dos primeras instrucciones permiten poner títulos y pies de página.

En la instrucción **OPTIONS**, existen diversas opciones: **LS=NUM** (o bien **LINESIZE=NUM**) define el número de caracteres por línea, **PS=NUM** (o bien **PAGESIZE=NUM**) define el número de líneas por página, **NODATE** elimina la fecha que aparece en la cabecera por defecto y **NONUMBER** elimina la numeración de las páginas.

Las instrucciones **DATA** o **PROC** definen bloques de instrucciones.

2.3 Los procedimientos y el paso data

Básicamente, todos los ficheros de sintaxis elaborados con SAS están formados por bloques de instrucciones **DATA** y **PROC**. El bloque **DATA** también recibe el nombre de **paso data** o **data step**.

Con la instrucción **DATA NOMBRE;** se inicia un bloque de instrucciones y automáticamente se crea un fichero de datos nuevo (o bien se reemplaza), siendo **NOMBRE** el correspondiente nombre asignado.

Un **dataset** es un conjunto de datos creado con SAS. Cualquier operación a realizar con un dataset se realiza dentro del bloque de instrucciones **DATA**: lectura de datos, creación de variables nuevas, recodificación, cambio de etiquetas, selección de casos,... SAS, al contrario que otros paquetes estadísticos permite trabajar simultáneamente con diversos ficheros de datos sin necesidad de abrirlos físicamente uno a uno.

Los procedimientos **PROC** se emplean para trabajar con los datos de un dataset sin modificar su estructura: análisis, tablas, listados, cálculos estadísticos,...

Los bloques de instrucciones definidos mediante las instrucciones **DATA** y **PROC** terminan con la instrucción **RUN ;** al abrir un nuevo bloque de instrucciones o bien con una instrucción de sistema.

Las instrucciones propias de los bloques de instrucciones **DATA** y **PROC** carecen de sentido y provocan errores fuera del correspondiente bloque.

2.4 Introducción de datos

La instrucción **DATA NOMBRE** inicia el bloque de instrucciones de manejo de ficheros de datos, creando un nuevo dataset llamado **NOMBRE**.

Las siguientes instrucciones forman parte del paso **DATA**, para leer datos entrados manualmente:

```
* DATOS CORRESPONDIENTES A UN ESTUDIO PARA COMPARAR DOS
TRATAMIENTOS ANTIHIPERTENSIVOS;
* SE ASIGNÓ UN TRATAMIENTO A CADA PACIENTE. SE MIDió LA
FRECUENCIA CARDÍACA ANTES DE REALIZAR UN EJERCIO FÍSICO
Y DESPUÉS;
```

```
*EN PRIMER LUGAR SE DISPONE DE LA INFORMACIÓN INICIAL
DE 3 PACIENTES;
```

```
DATA PACTIVO1;
INPUT IB SEXO $ FUMADOR EDAD FC1 FARMACO;
CARDS;
4      M      2      22.6 96      1
2      M      1      20.0 78      2
23     M      1      21.6 76      1
;
RUN;
```

En la instrucción **INPUT** se declaran las variables que van a ser leídas, ya sea mediante datos entrados manualmente o procedentes de un fichero externo: nombres de las variables y " \$ " a continuación si se trata de una variable alfanumérica.

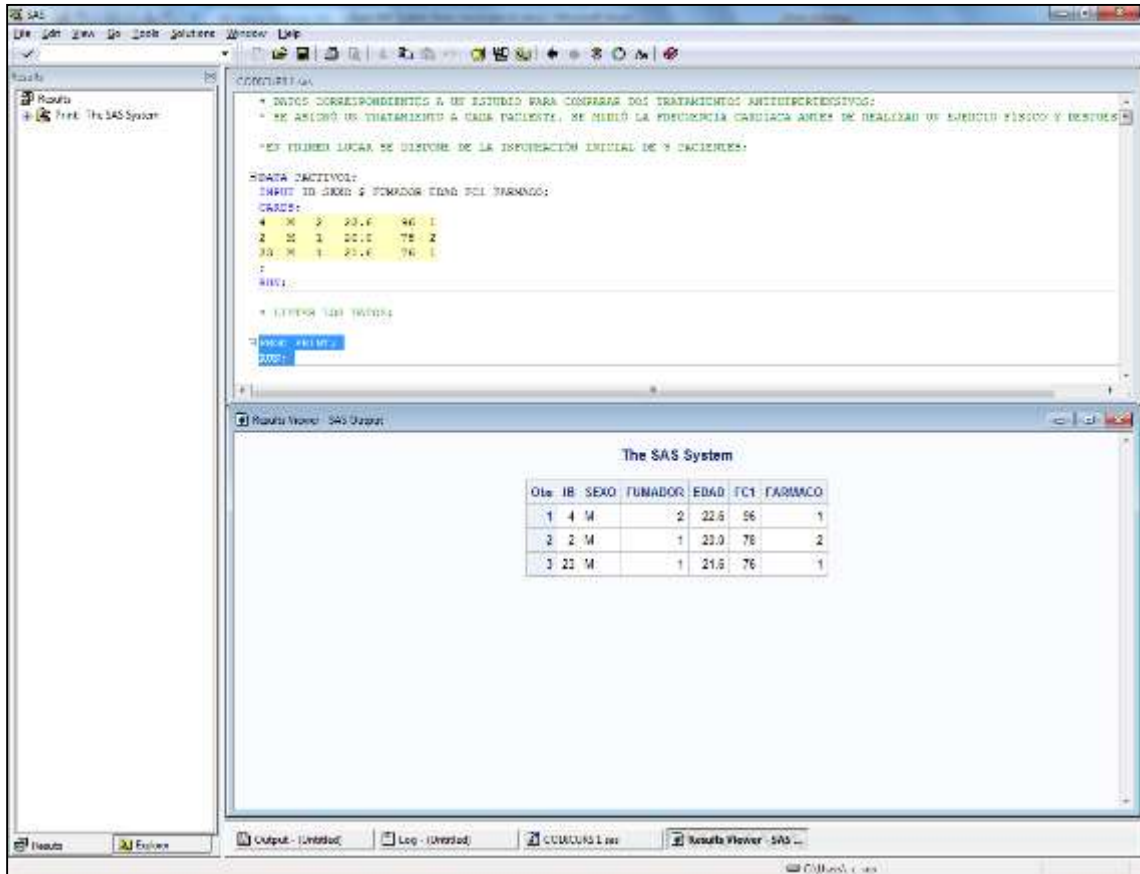
La instrucción **CARDS** sirve para iniciar la lectura de los datos internos. El símbolo "espacio" sirve para delimitar los campos de datos.

El procedimiento **PROC PRINT** es el procedimiento que lista los datos.

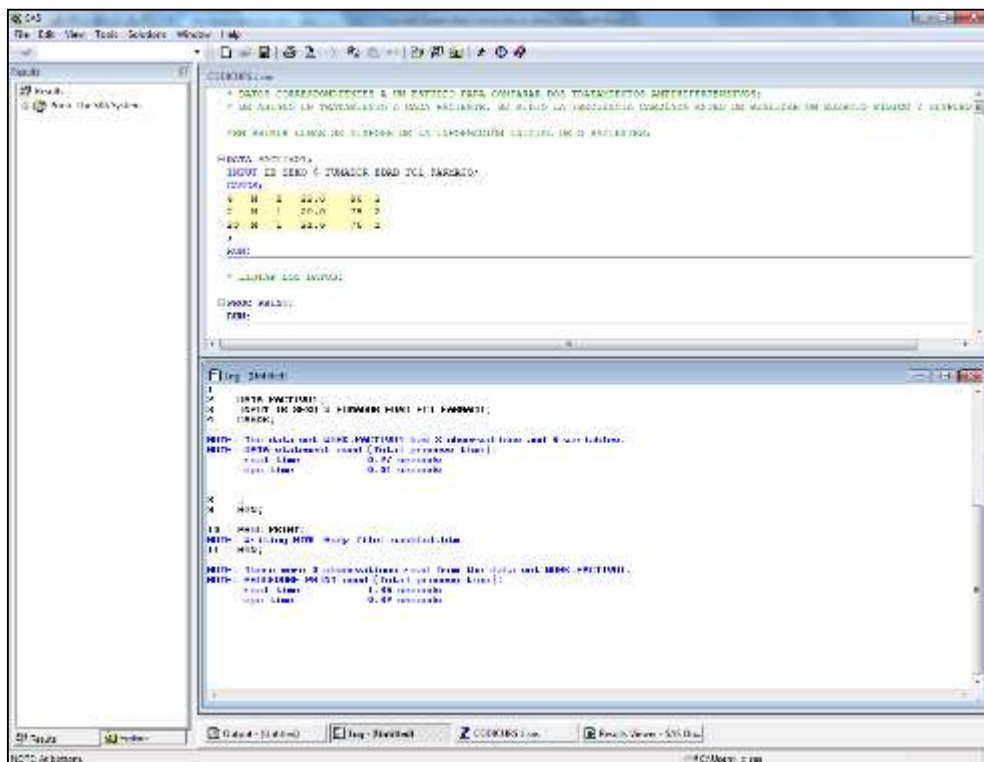
Observando la ventana **LOG**, se consultan los errores cometidos en la sintaxis del **EDITOR**.

```
* LISTAR LOS DATOS;
```

```
PROC PRINT;
RUN;
```



Ventana **SINTAX EDITOR** donde se visualiza el código ejecutado y
Ventana **RESULTS VIEWER** donde se visualiza el resultado de **PROC PRINT**.



Ventana **LOG** donde se visualiza el estado de la ejecución.

2.5 Importar datos

Dentro del paso **DATA**, la instrucción **INFILE** se utiliza para la lectura de datos externos y en ella se menciona la ruta donde se encuentra el fichero que contiene los datos. La opción **LRECL** de la instrucción **INFILE** indica la longitud máxima de cada línea (es indispensable si cada registro tiene más de 256 caracteres).

En la instrucción **INPUT** se declara las variables que se van a leer. Si el fichero es externo, éste puede tener una estructura predefinida:

Ficheros con formato fijo: A continuación del nombre de la variable se escriben las columnas que ocupa.

Ficheros delimitados: Por defecto, el separador que lee SAS es el espacio, pero con la opción **DLM** se define el delimitador que deseado, por ejemplo: **DLM='09'X** si el fichero es encuentra delimitado por tabuladores o **DLM=';'** si el fichero es encuentra delimitado por el símbolo "; ". La opción **FIRSTOBS=N** permite leer datos a partir del registro especificado, eliminando los **N-1** primeros registros del fichero.

```
* LECTURA DE DATOS EXTERNOS;  
* OTRO INVESTIGADOR HA RECOGIDO LOS DATOS INICIALES  
  PARA EL RESTO DE SUJETOS;
```

El fichero está en formato fijo. Los primeros registros del fichero son:

```
36H121.4622  
35M224.5801  
20H122.1662  
19M122.5881
```

```
DATA PACTIVO2;  
INFILE 'C:\TEMP\DATOS1.TXT' LRECL=100;  
INPUT IB 1-2 SEXO $ 3 FUMADOR 4 EDAD 5-8 FC1 9-10  
FARMACO 11;  
RUN;  
  
PROC PRINT DATA=PACTIVO2;  
RUN;
```

* EN UN FICHERO DISTINTO SE HA RECOGIDO LOS DATOS
FINALES;

El nuevo fichero está delimitado por tabuladores y contiene los nombres de las variables en la primera fila. Los primeros registros del fichero son:

```
IB FC2 STATUS
4    155    1
2    154    1
23   148    1
36   132    1
```

```
DATA PACTIVO3;
INFILE 'C:\TEMP\DATOS2.TXT' DLM='09'X FIRSTOBS=2;
INPUT IB FC2 STATUS;
RUN;

PROC PRINT DATA=PACTIVO3;
RUN;
```

Cualquier procedimiento trabaja con el dataset deseado utilizando la opción **DATA=NOMBRE_DATASET**. Por defecto, SAS utiliza el dataset creado en el paso **DATA** más reciente.

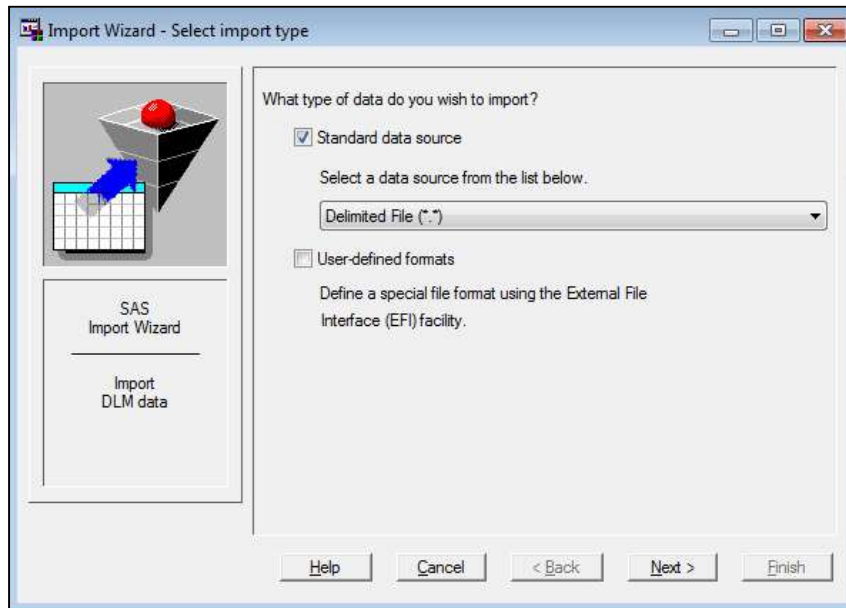
Es útil observar el listado producido por **PROC PRINT** para comprobar que efectivamente los datos se han leído perfectamente (en ocasiones no se cometen errores en la sintaxis aunque los datos no se leen adecuadamente).

2.6 Importar datos mediante el asistente

SAS dispone de un asistente para la lectura de datos externos en formato texto (entre otros), que se puede utilizar desde el menú desplegable:

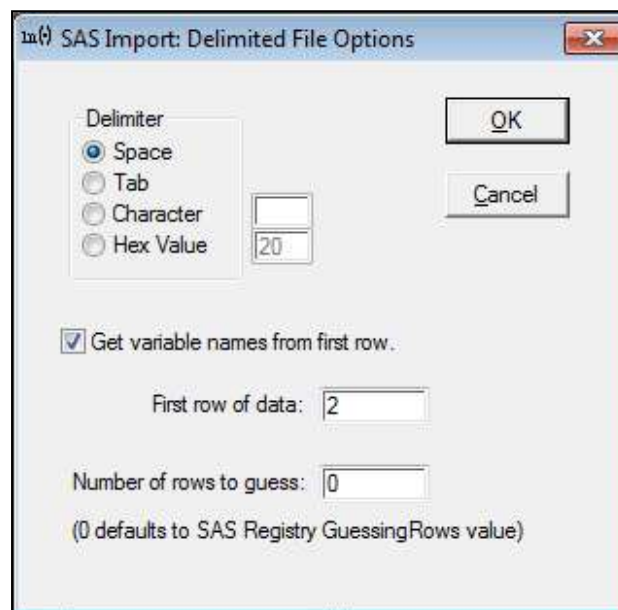
FILE → IMPORT DATA

Mediante este asistente se pueden leer de forma más automática ficheros externos, con la posibilidad de almacenar la sintaxis que genera internamente el programa.



Ventana del Asistente para la importación.

Es necesario especificar las opciones requeridas pulsando el botón **OPTIONS** :



Ventana de opciones.

2.7 Explorar los contenidos de un dataset

Con el procedimiento **CONTENTS**, se obtiene un listado con información relativa al dataset que se menciona en la opción **DATA=NOMBRE_DATASET**, así como de sus variables. Especificando la opción **VARNUM**, el listado de las variables está ordenado según la posición de cada variable en el dataset. Por defecto, sin la opción anterior, este orden es alfabético.

Este procedimiento es útil para comprobar posibles errores en la lectura de ficheros que no sean detectables a través de la ventana **LOG**.

* CONTENIDOS DE UN DATASET;

```
PROC CONTENTS DATA=PACTIVO2 VARNUM;
RUN;
```

The screenshot shows the SAS Results Viewer window. The main area displays the following information:

Member Type	DATA	Variables	6
Engine	VE	Indexes	0
Created	miércoles 24 de octubre de 2014 12:15	Observation Length	40
Last Modified	miércoles 24 de octubre de 2014 12:15	Missing Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS 32		
Executing	win32_Windows (Windows)		

Below this, there is a section titled "Engine/Host Dependent information":

Data Set Page Size	4096
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	64
Obs in First Data Page	50
Number of Data Set Segments	0
File Name	C:\Users\c...s\AppData\Local\Temp\SAS Temporary Files\TD4552\DS02CR10\pactivo2.sas7bdat
Database Created	5/22/1972
Host Created	W32_TPRO

At the bottom, there is a table titled "Variables in Creation Order":

#	Variable	Type	Len
1	ID	Num	8
2	SEXO	Char	1
3	FUMADOR	Num	8
4	EDAD	Num	8
5	FCI	Num	8
6	FARMACO	Num	8

Información parcial obtenida con **PROC CONTENTS**.

DATASET: Nombre del dataset, número de observaciones, número de variables, fecha de creación.

VARIABLES: Orden interno, nombre de la variable, tipos, longitud, **FORMAT**, **INFORMAT** y **LABEL** (estos últimos conceptos se tratarán posteriormente).

2.8 La instrucción SET del paso data

La instrucción **SET** forma parte de un bloque **DATA**, y asigna al dataset del paso **DATA** los mismos datos, variables y propiedades que el dataset especificado en la instrucción **SET**.

```
DATA A;  
SET PACTIVO3;  
RUN;
```

Con el código anterior, se crea una copia del dataset **PACTIVO3** llamado **A**. En estos momentos, los datasets **PACTIVO3** y **A** son idénticos y ambos temporales.

```
DATA A;  
SET A;  
VAR _A=1;  
RUN;
```

Con el código anterior se crea una copia del dataset **A** llamado también **A**. Simultáneamente se añade la variable **VAR_A** y se asigna para todos los casos el valor 1. De hecho se ha reemplazado el dataset **A** por otro idéntico pero con una variable más.

Siempre que sea necesaria alguna modificación, operación, recodificación,... en algún dataset se realiza mediante el paso **DATA** y la instrucción **SET**.

2.9 Almacenar datos SAS

Los ficheros de datos que utiliza SAS son siempre almacenados en alguna librería. Por defecto, esta librería es la librería temporal **WORK** que se vacía automáticamente cuando se apaga el programa.

Para definir una nueva librería se utiliza la instrucción **LIBNAME**:

```
LIBNAME NOMBRELIB "C:\RUTA DONDE SE ALMACENARÁN LOS FICHEROS\";
```

NOMBRELIB es el nombre de la nueva librería. Entre comillas se especifica la ruta física donde se desean almacenar los ficheros de datos. Esta librería será permanente en el sentido que no se vaciará al terminar el programa, y los datos estarán listos para ser utilizados de nuevo en siguientes sesiones sin ser necesaria la importación de ficheros.

Una vez definida la librería, siempre que se desea hacer referencia a un fichero de datos que físicamente se encuentre en la ruta correspondiente, se utilizará como nombre del dataset un nombre compuesto:

NOMBRELIB.NOMBRE_DATASET

Para crear un dataset temporal, el nombre del dataset tiene que ser simple: "a", "peso" y "patata" son nombres válidos. También son válidos los nombres compuestos con la librería work: "work.pera" es equivalente a "pera".

Nombres de los datasets, librerías y tipos:

PASO DATA	LIBRERIA	NOMBRE	TIPO
DATA PESO	WORK	PESO	TEMPORAL
DATA WORK.PATATA	WORK	PATATA	TEMPORAL
DATA FRUTA.KIWI	FRUTA	KIWI	PERMANENTE

Hasta ahora todos los datasets creados han sido temporales.

Es recomendable que los nombres de los datasets y de las variables no tengan más de 8 caracteres.

La instrucción **LIBNAME** es temporal en el sentido que al iniciar una nueva sesión es necesario volver a ejecutar la instrucción **LIBNAME** para recordar al programa donde se hallan nuestros ficheros de datos.

* CONVERSIÓN DE DATASETS TEMPORALES A PERMENANTES;

```
LIBNAME FC 'C:\TEMP\';
```

```
DATA FC.PACTIVO1;
SET PACTIVO1;
RUN;
```

```
DATA FC.PACTIVO2;
SET PACTIVO2;
RUN;
```

```
DATA FC.PACTIVO3;
SET PACTIVO3;
RUN;
```

```
OPTIONS LS=75 PS=50 NODATE NONUMBER;
TITLE 'DATASET FC.PACTIVO2';
FOOTNOTE 'DATASET PERMANENTE';
PROC PRINT DATA=FC.PACTIVO2;
RUN;
TITLE;
FOOTNOTE;
```

Obs	FC.PACTIVO2	FC.PACTIVO2	FC.PACTIVO2	FC.PACTIVO2
1	1	1	1	1
2	2	2	2	2
3	3	3	3	3
4	4	4	4	4
5	5	5	5	5
6	6	6	6	6
7	7	7	7	7
8	8	8	8	8
9	9	9	9	9
10	10	10	10	10
11	11	11	11	11
12	12	12	12	12
13	13	13	13	13
14	14	14	14	14
15	15	15	15	15
16	16	16	16	16
17	17	17	17	17
18	18	18	18	18
19	19	19	19	19
20	20	20	20	20
21	21	21	21	21
22	22	22	22	22
23	23	23	23	23
24	24	24	24	24
25	25	25	25	25
26	26	26	26	26
27	27	27	27	27
28	28	28	28	28
29	29	29	29	29
30	30	30	30	30

Vista parcial de la ventana **RESULTS VIEWER** después de la ejecución del código anterior.

Ahora los datasets **PACTIVO1**, **PACTIVO2** y **PACTIVO3** se encuentran en las librerías **WORK** y **FC**, es decir, en total hay seis datasets diferentes (aunque de momento iguales dos a dos). Los datasets mencionados se encuentran guardados de forma temporal en la librería **WORK** y de forma permanente en la librería **FC**, físicamente se encuentran en la ruta especificada en la instrucción **LIBNAME** (C:\TEMP\) cuando se definió la librería.

Si se desea recuperar un fichero de datos creado con SAS en una sesión anterior para utilizarlo o continuar trabajando se puede utilizar el siguiente código:

El fichero físico tiene la extensión ".SD2" en la versión 6.12 o ".SAS7BDAT" en la versión 8 y posteriores.

Supondremos que se encuentra ubicado en "C:\TEMP\" y se llama **DATASET_VIEJO**.


Físicamente, el fichero es "C:\TEMP\DATASET_VIEJO.SAS7BDAT".

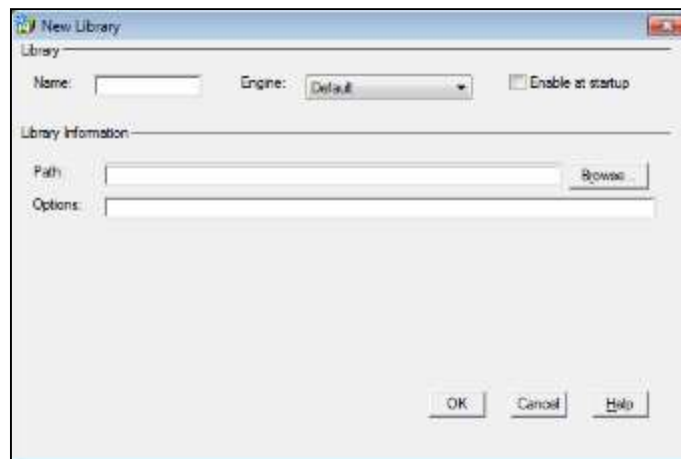
Para cargar los datos del fichero anterior, ejecutamos la siguiente sintaxis:

```
LIBNAME NOMBRE_LIBRERIA 'C:\TEMP\';

DATA NOMBRE_DATASET;
SET NOMBRE_LIBRERIA.DATASET_VIEJO;
RUN;
```

Después de ejecutar el código anterior el **DATASET_VIEJO** continúa en su ubicación original sin modificar, pero se ha creado una copia temporal llamada **NOMBRE_DATASET** en la librería **WORK**. Así mismo ambos datasets están listos para ser utilizados.

Para crear una librería auténticamente permanente se debe pulsar el botón  del menú y rellenar los campos.



Finalmente es necesario activar la pestaña: **ENABLE AT STARTUP**.

La librería creada (activando la pestaña) aparecerá siempre al iniciar una sesión de SAS.

Las librerías creadas en SAS System o SAS EG no se visualizan directamente en el otro programa.

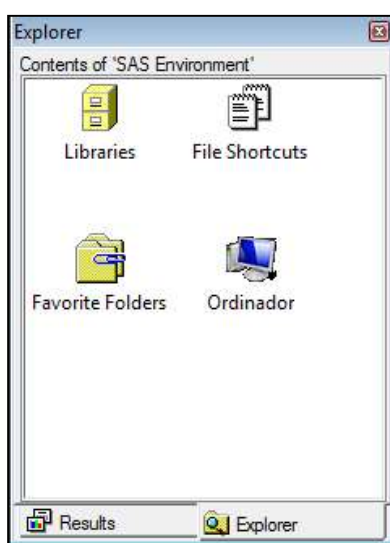
Se pueden consultar las librerías definidas utilizando la ventana **EXPLORER**.

2.10 Explorar Librerías

Utilizando la ventana **EXPLORER** es posible explorar los contenidos de las diferentes librerías, comprobando los datasets existentes así como explorar los contenidos de los mismos datasets observando la correspondiente cuadrícula de variables y registros.

Las librerías que tiene definidas SAS por defecto son:

WORK	Librería Temporal
SASHELP	Recursos de la Ayuda de SAS
SASUSER	Parámetros del usuario
MAPS	Ficheros de datos para construir mapas.

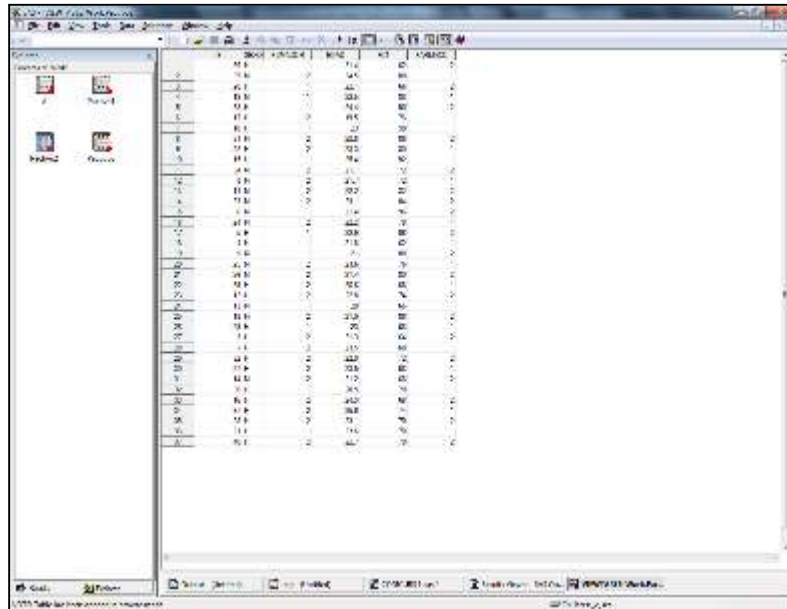


Ventana **EXPLORER**.

Al hacer doble-clic en alguna librería, aparecen listadas todas las bases de datos SAS almacenadas en la correspondiente carpeta.

2.11 Explorar Datos

Al abrir algún fichero de datos desde la ventana Explorer, se abre el visor de datos:




The screenshot shows the SAS VIEW TABLE window. The main area is a data grid with columns and rows. The columns are labeled with variable names: ID, SEX, TUMORCE, EDAD, FEU, and FARMACO. The rows contain numerical and categorical data. The window has a menu bar (File, Edit, View, Format, Tools, Help) and a toolbar with various icons for editing and viewing the data.

Visor de datos **VIEW TABLE**.

Los menús desplegables que aparecen en la barra de herramientas son diferentes a los habituales e incorporan opciones e iconos especiales para definir y manipular librerías así como editar ficheros de datos.

Haciendo doble clic en cada una de las variables podemos observar sus propiedades.

Los botones  permiten editar la tabla. Incluso es posible introducir nuevos datos visualizando la base de datos como formulario.



The screenshot shows the SAS VIEW TABLE window in form view. The main area is a form with input fields for each variable: ID, SEX, TUMORCE, EDAD, FEU, and FARMACO. The form is designed for data entry and editing. The window has a menu bar (File, Edit, View, Format, Tools, Help) and a toolbar with various icons for editing and viewing the data.

Vista de la base de datos como formulario.

2.12 Ordenar datos

Con el procedimiento **SORT** se ordena el dataset según las variables requeridas en la instrucción **BY**.

Por defecto los datos son ordenados de forma ascendente. Podemos añadir antes de la variable la opción **DESCENDING** si deseamos los datos ordenados de forma descendente.

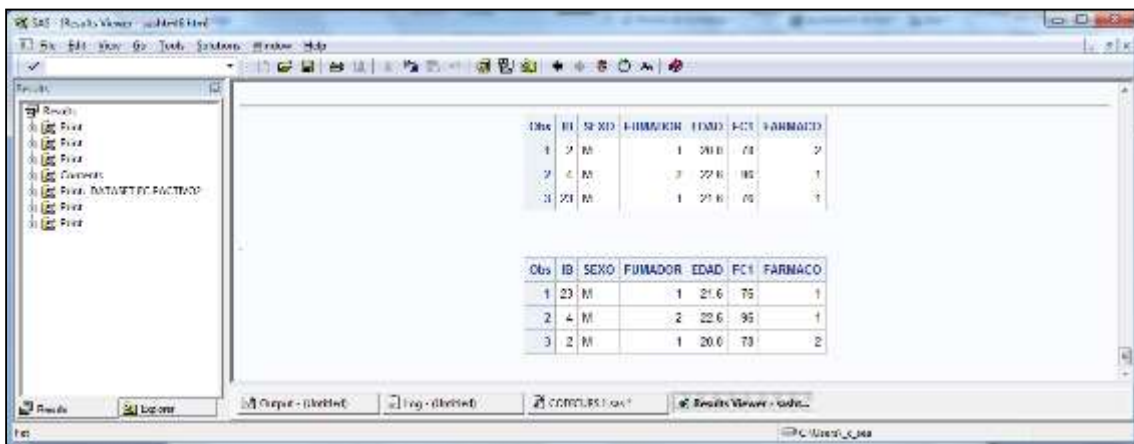
```
* ORDENAR UN DATASET;
```

```
PROC SORT DATA= PACTIVO1;  
BY IB;  
RUN;
```

```
PROC PRINT DATA= PACTIVO1;  
RUN;
```

```
PROC SORT DATA= PACTIVO1;  
BY DESCENDING IB;  
RUN;
```

```
PROC PRINT DATA= PACTIVO1;  
RUN;
```



The screenshot displays the SAS Results Viewer interface. On the left, a tree view shows the results of several PROC PRINT statements. The main window contains two data tables. The top table shows the data sorted by the variable IB, with rows 1, 2, and 3. The bottom table shows the data sorted by IB in descending order, with rows 3, 2, and 1.

Obs	IB	SEXO	FUMADOR	EDAD	FC1	FARMACO
1	3	M	1	20.0	70	2
2	4	M	2	22.6	95	1
3	2	M	1	21.6	75	1

Obs	IB	SEXO	FUMADOR	EDAD	FC1	FARMACO
1	3	M	1	21.6	75	1
2	4	M	2	22.6	95	1
3	2	M	1	20.0	70	2

2.13 Fundir archivos: Añadir casos

Para añadir casos a un dataset, se puede utilizar el procedimiento **APPEND**. El dataset resultante es el dataset especificado en la opción **BASE**, y a éste se le añaden todos los casos del dataset de la opción **DATA**. Cuando ambos datasets no tienen idéntica estructura (por ejemplo una misma variable está definida como Alfanumérica de 5 posiciones en un dataset y de 8 posiciones en otro) se requiere la opción **FORCE** al final de la instrucción.

* FUSIÓN AÑADIENDO CASOS;

```
DATA PACTIVO12;
SET PACTIVO1;
RUN;
```

```
PROC APPEND BASE= PACTIVO12 DATA= PACTIVO2 FORCE;
RUN;
```

```
PROC PRINT;
RUN;
```

Obs	SEX	PRACTICO	FUND	FE	FUNDADO
1	M	1	218	75	1
2	M	2	220	85	1
3	M	1	218	78	2
4	H	1	214	82	2
5	M	2	217	83	1
6	H	1	221	89	2
7	M	1	220	83	1
8	H	1	214	83	2
9	H	2	212	79	1
10	H	1	220	83	1
11	M	2	228	89	2
12	H	2	222	89	1
13	H	1	214	87	1
14	M	2	211	72	2
15	M	2	211	72	1
16	M	2	222	87	2
17	M	2	221	84	2
18	M	1	214	85	2
19	M	2	222	78	1
20	H	1	222	85	2
21	H	1	216	87	1
22	M	1	210	89	2
23	M	2	226	75	1
24	M	2	214	83	2
25	H	2	228	85	1
26	H	2	220	71	2
27	M	1	218	89	1
28	M	2	215	80	2
29	H	1	220	83	1
30	H	2	212	84	2

Ventana **RESULTS VIEWER** con el resultado de la unión de los datasets.

Cuando existe una variable identificativa común en ambos datasets también se usa la instrucción **SET** en un paso **DATA**, obteniendo exactamente el mismo resultado que con **PROC APPEND**, de la siguiente forma:

```
* FUSIÓN AÑADIENDO CASOS;

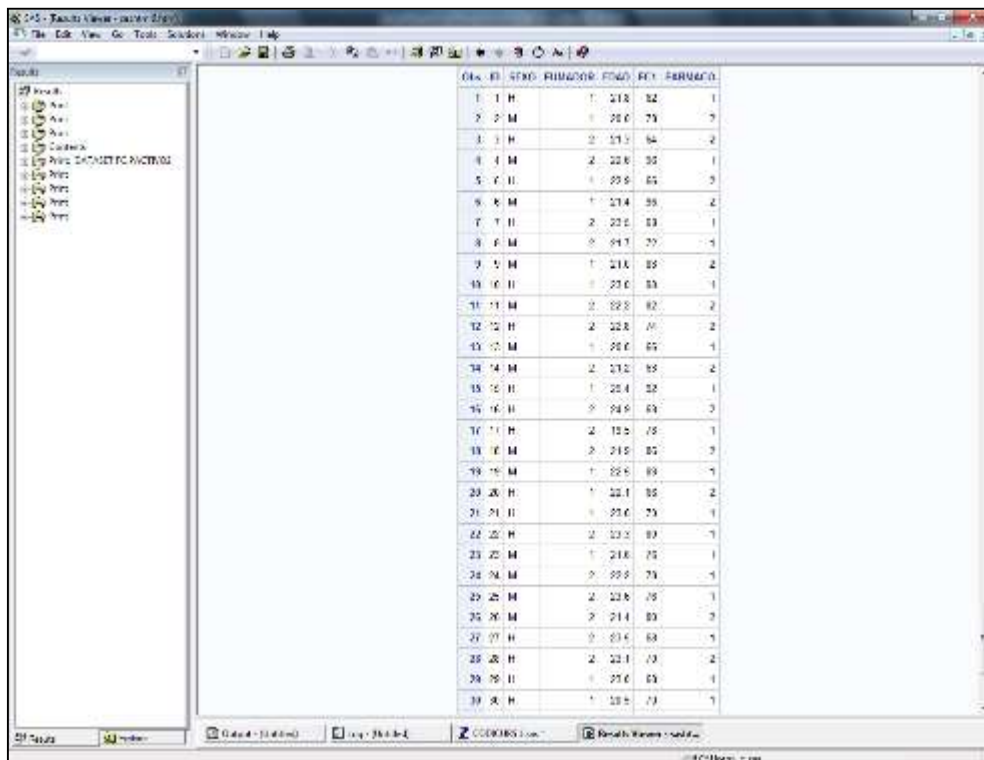
PROC SORT DATA= PACTIVO1;
BY IB;
RUN;

PROC SORT DATA= PACTIVO2;
BY IB;
RUN;

DATA PACTIVO12;
SET PACTIVO1 PACTIVO2;
BY IB;
RUN;

PROC PRINT;
RUN;
```

En general es preferible utilizar la instrucción **SET** por ser más flexible.



Obs	IB	SEXO	PRACTICACION	EDAD	ECI	FARMACIA
1	1	H	1	21.8	32	1
2	2	M	1	28.0	73	2
3	1	H	2	21.3	34	2
4	4	M	2	32.8	35	1
5	2	H	1	22.5	65	2
6	8	M	1	21.4	35	2
7	1	H	2	22.0	33	1
8	4	M	2	24.1	29	1
9	5	M	1	21.6	35	2
10	6	H	1	23.0	33	1
11	1	M	2	22.8	32	2
12	12	H	2	32.8	36	2
13	7	M	1	28.0	35	1
14	4	M	2	21.2	35	2
15	15	H	1	22.4	32	1
16	16	H	2	24.9	33	2
17	1	H	2	12.5	35	1
18	8	M	2	21.9	35	2
19	16	M	1	22.5	33	1
20	20	H	1	22.1	35	2
21	21	H	1	22.0	29	1
22	22	H	2	22.2	33	1
23	23	M	1	21.8	25	1
24	24	M	2	22.2	23	1
25	25	M	2	22.8	25	1
26	26	M	2	21.4	33	2
27	27	H	2	23.5	33	1
28	28	H	2	22.1	33	2
29	29	H	1	22.0	33	1
30	30	H	1	22.5	23	1

Ventana **RESULTS VIEWER** con el merge de los datasets.

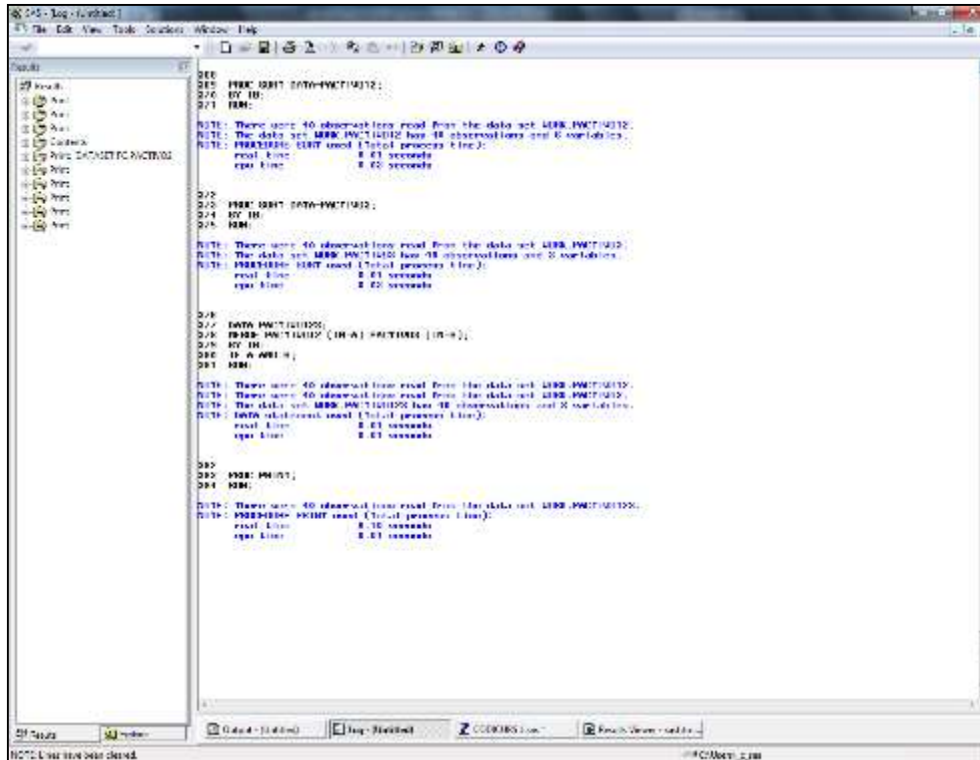
2.14 Fundir archivos: Añadir variables

Dentro del paso **DATA**, la instrucción **MERGE** permite añadir variables.

Si los datasets que contienen las diferentes variables no tienen exactamente los mismos registros, es recomendable utilizar la instrucción **BY VARIABLES**. Las variables referidas en la instrucción **BY**, son tomadas como claves de referencia, tienen que estar presentes en ambos datasets y permiten identificar los diferentes casos dentro de cada dataset y asignarles la correspondencia adecuada. En caso de utilizar esta última opción, los dos datasets deben estar ordenados por las variables mencionadas en **BY** antes de realizar la fusión.

```
* FUSIÓN AÑADIENDO VARIABLES;  
  
PROC SORT DATA= PACTIVO12;  
BY IB;  
RUN;  
  
PROC SORT DATA= PACTIVO3;  
BY IB;  
RUN;  
  
DATA PACTIVO123;  
MERGE PACTIVO12 (IN=A) PACTIVO3 (IN=B);  
BY IB;  
IF A AND B;  
RUN;  
  
PROC PRINT;  
RUN;
```

Las variables A y B son temporales y se definen en la instrucción **MERGE**. Indican la procedencia de cada observación tras la FUSIÓN con los valores 0 (NO) y 1 (SÍ). Cuando un caso está presente en ambas bases de datos, las variables A y B valdrán 1. Sin embargo, si un caso no está presente en la segunda base de datos, la variable B valdrá 0.



```
225 PROC SORT DATA=WORK12;  
226 BY IN;  
227 RUN;  
NOTE: There were 40 observations read from the data set WORK12.  
NOTE: The data set WORK12 has 40 observations and 0 variables.  
NOTE: PROC SORT used Total processor time:  
real time 0.01 seconds  
cpu time 0.00 seconds  
228  
229 PROC SORT DATA=WORK12;  
230 BY IN;  
231 RUN;  
NOTE: There were 40 observations read from the data set WORK12.  
NOTE: The data set WORK12 has 40 observations and 0 variables.  
NOTE: PROC SORT used Total processor time:  
real time 0.01 seconds  
cpu time 0.00 seconds  
232  
233 DATA WORK12;  
234 IN=WORK12 (IN=1) *WORK12 (IN=2);  
235 BY IN;  
236 RUN;  
NOTE: There were 40 observations read from the data set WORK12.  
NOTE: There were 40 observations read from the data set WORK12.  
NOTE: The data set WORK12 has 40 observations and 0 variables.  
NOTE: DATA used Total processor time:  
real time 0.01 seconds  
cpu time 0.01 seconds  
237  
238 PROC PRINT;  
239 RUN;  
NOTE: There were 40 observations read from the data set WORK12.  
NOTE: PROC PRINT used Total processor time:  
real time 0.10 seconds  
cpu time 0.01 seconds
```

Ventana **LOG** después de la ejecución del código anterior.

2.15 Definición de etiquetas

SAS nos permite dar formato a los valores que son listados o visualizados en la ventana **OUTPUT**.

Cualquiera de estos "formatos" se puede crear mediante el procedimiento **FORMAT** y una vez creado, se puede utilizar para tantas variables como nos interese.

La instrucción **VALUE** indica un nuevo formato:

```
VALUE NOMBRE_FORMATO
  VALOR1="ETIQUETA_DE_VALOR1"
  VALOR2="ETIQUETA_DE_VALOR2"
  ...
  OTHER="ETIQUETA_POR_DEFECTO" ;
```

Para formatos alfanuméricos:

```
VALUE $ NOMBRE_FORMATO
  "VALOR1"="ETIQUETA_DE_VALOR1"
  "VALOR2"="ETIQUETA_DE_VALOR2"
  ...
  OTHER="ETIQUETA_POR_DEFECTO" ;
```

```
* DEFINICIÓN DE ETIQUETAS DE VALORES;
```

```
PROC FORMAT;
VALUE VFUM 1="SI" 2="NO";
VALUE $ VSEXO "H"='HOMBRE' "M"='MUJER';
RUN;
```

Con la instrucción **LABEL**, se dan etiquetas a las variables.

La operación de asignar un *formato* a una variable se realiza mediante la instrucción **FORMAT** en un paso **DATA** o en un **PROC**. La diferencia reside en que la especificación de estas instrucciones en un **PROC** tiene carácter temporal, mientras que las definiciones en un paso **DATA** pasan a ser permanentes. Si finalmente se definen en un paso **DATA**, al iniciar la sesión será imprescindible ejecutar de nuevo las definiciones de los formatos (**PROC FORMAT**).

```
FORMAT VARIABLENUMÉRICA NOMBRE_FORMATO.
  VARIABLEALFANUMÉRICA NOMBRE_FORMATO$ . ;
```

siendo indispensable el punto después del nombre de formato.

```
LABEL VARIABLE = "ETIQUETA";
```

* DEFINICIÓN DEL FORMATO DE LAS VARIABLES;

```
DATA FINAL;
SET PACTIVO123;
LABEL IB='NUMERO PACIENTE';
LABEL FARMACO='TRATAMIENTO ADMINISTRADO';
FORMAT SEXO $VSEXO. EDAD 2.;
RUN;
```

* GUARDAR LA BBDD DEFINITIVA;

```
DATA FC.FINAL;
SET FINAL;
RUN;
```

* IMPRIMIR LA BASE DE DATOS;

```
PROC PRINT DATA=FC.FINAL LABEL NOOBS;
VAR IB FARMACO FUMADOR EDAD SEXO FC1 FC2;
LABEL FC1='LABEL TEMPORAL' FC2='LABEL TEMPORAL';
RUN;
```

NUMERO PACIENTE	TRATAMIENTO ADMINISTRADO	FUMADOR	EDAD	SEXO	LABEL TEMPORAL	LABEL TEMPORAL
1		1	22	HOMBRE	82	126
2		1	20	MUJER	79	124
3		2	21	HOMBRE	84	125
4		1	22	MUJER	82	126
5		2	21	HOMBRE	84	125
6		1	21	MUJER	80	124
7		1	24	HOMBRE	83	127
8		1	22	MUJER	72	118
9		2	21	MUJER	80	124
10		1	22	HOMBRE	82	126
11		2	22	MUJER	82	126
12		2	22	HOMBRE	79	124
13		1	20	MUJER	80	124
14		2	21	MUJER	80	124
15		1	22	HOMBRE	82	126
16		2	20	HOMBRE	80	124
17		1	22	HOMBRE	79	124
18		2	22	MUJER	82	126
19		1	22	HOMBRE	82	126
20		2	22	HOMBRE	82	126
21		1	24	HOMBRE	79	124
22		1	22	HOMBRE	82	126
23		1	22	MUJER	79	124
24		1	22	MUJER	79	124
25		1	24	MUJER	79	124
26		2	21	MUJER	80	124
27		2	21	MUJER	80	124
28		2	22	HOMBRE	79	124
29		1	21	HOMBRE	80	124
30		1	21	HOMBRE	79	124

Ventana **RESULTS VIEWER** con el resultado de **PROC PRINT** con las etiquetas.

Quando se incluye la opción **LABEL** en el procedimiento **PRINT**, aparecen las etiquetas de las variables, con la opción **NOOBS** no aparece el número de observación correspondiente a cada caso. Utilizando **VAR**, se consigue que sólo se listen las variables mencionadas y en el orden deseado (por defecto se listan todas).

2.16 Exportar datos

Con un nuevo paso **DATA** podemos exportar los datos en formato de texto.

La instrucción **FILE** es necesaria para exportar los datos de un dataset a la ruta y formato deseados, pero sólo sirve para ficheros de tipo texto.

En la instrucción **PUT** se mencionan las posiciones de cada variable (en el caso que el fichero de datos sea de formato fijo).

Utilizando como nombre de un dataset la palabra **_NULL_**, no se creará ningún dataset nuevo.

```
* EXPORTAR DATOS;  
  
DATA _NULL_ ;  
SET FC.FINAL;  
FILE 'C:\TEMP\FINAL.TXT' ;  
PUT IB 1-2 FARMACO 3 SEXO 7;  
RUN;
```

Al igual que para importar, también existe la posibilidad de exportar datos utilizando el asistente proporcionado por SAS.

FILE → EXPORT DATA

2.17 Crear nuevas variables

Para trabajar con los datos, muchas veces es necesario recodificar y generar nuevas variables:

Estas modificaciones se realizan en el paso **DATA**.

VARIABLES nuevas:

VARIABLE = VALOR O FÓRMULA

La fórmula puede incluir otras variables, obteniendo el resultado de la fórmula registro a registro.

Recodificación:

Cuando la condición que hay después de **IF** es cierta, entonces se aplica la transformación que aparece después de **THEN**. Cuando después de un **IF** hay la instrucción **ELSE**, entonces se aplica la transformación que se menciona, sólo cuando la condición especificada en **IF** es falsa.

Al hacer diversas transformaciones en una sola condición **IF**, deben estar todas ellas entre un **DO** y un **END**.

* RECODIFICACIÓN DE VALORES;
* CREAR VARIABLES NUEVAS;

```
DATA FINAL_S;  
SET FC.FINAL;  
DIF=FC2-FC1;  
IF EDAD < 23 THEN EDAD2=1;  
ELSE EDAD2=2;  
IF IB=5 THEN DO;  
EDAD=26;  
SEXO="M";  
END;  
RUN;  
  
PROC PRINT;  
RUN;
```


Se ha creado un dataset temporal con las modificaciones.

Obs	ID	SEXO	FUMADOR	EDAD	FC1	FARMACO	PC2	STATUS	DIF	EDAD2
1	1	HOMBRE	1	22	62	1	126	2	64	1
2	2	MUJER	1	23	70	2	154	1	76	1
3	3	HOMBRE	2	21	64	2	120	3	64	1
4	4	MUJER	2	23	96	1	166	1	68	1
5	5	MUJER	1	25	66	2	120	2	62	1
6	6	MUJER	1	21	96	2	164	2	124	1
7	7	HOMBRE	2	24	80	1	120	3	102	2
8	8	MUJER	2	23	12	1	128	1	106	1
9	9	MUJER	1	21	88	2	160	2	72	1
10	10	HOMBRE	1	23	50	1	142	1	54	2
11	11	MUJER	2	22	82	2	140	1	58	1
12	12	HOMBRE	2	23	74	2	134	2	80	1
13	13	MUJER	1	23	66	1	148	2	82	1
14	14	MUJER	2	21	68	2	142	3	74	1
15	15	HOMBRE	1	25	92	1	134	1	42	2
16	16	HOMBRE	2	25	60	2	112	3	44	2
17	17	HOMBRE	2	23	76	1	158	1	82	1
18	18	MUJER	2	22	66	2	146	2	60	1
19	19	MUJER	1	23	60	1	166	1	60	1
20	20	HOMBRE	1	22	66	2	132	1	66	1
21	21	HOMBRE	1	24	60	1	122	3	62	2
22	22	HOMBRE	2	23	80	1	120	1	106	2
23	23	MUJER	1	22	66	1	140	1	72	1
24	24	MUJER	2	23	66	1	140	2	66	1
25	25	MUJER	2	24	76	1	126	2	80	2
26	26	MUJER	2	21	80	2	158	2	78	1
27	27	HOMBRE	2	24	68	1	116	3	48	2
28	28	HOMBRE	2	23	70	2	120	3	50	2
29	29	HOMBRE	1	22	68	1	126	3	58	2
30	30	HOMBRE	1	21	70	1	144	3	74	1

Ventana **RESULTS VIEWER** con el resultado del **PROC PRINT** ejecutado.

2.18 Filtrar casos

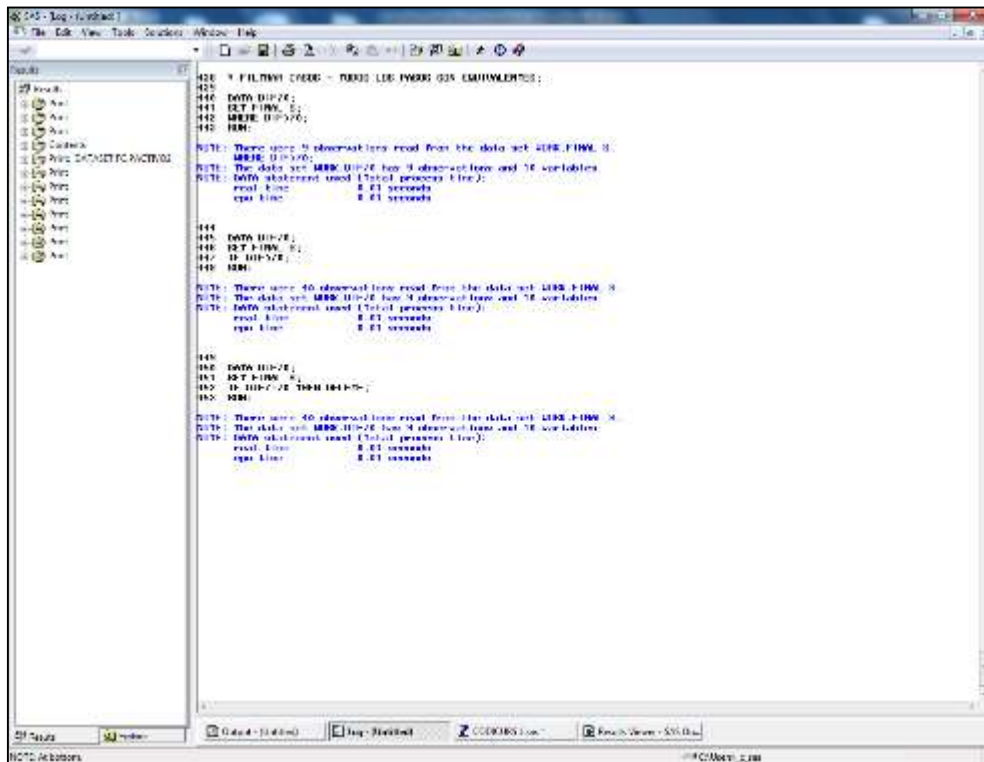
Se pueden seleccionar casos o datos que cumplan cierta condición, a partir de la instrucción **IF** o equivalentemente la instrucción **WHERE**:

* FILTRAR CASOS – TODOS LOS PASOS SON EQUIVALENTES;

```
DATA DIF70;
SET FINAL_S;
WHERE DIF>70;
RUN;
```

```
DATA DIF70;
SET FINAL_S;
IF DIF>70;
RUN;
```

```
DATA DIF70;
SET FINAL_S;
IF DIF<=70 THEN DELETE;
RUN;
```



Ventana **LOG** donde se puede ver la selección de casos realizada.

El filtro con **WHERE** se ejecuta **antes** de la lectura, mientras que **IF** se ejecuta **después**.

2.19 Funciones de SAS

Cuando se trabaja con las variables de la base de datos, es usual generar nuevas variables a partir de aquellas que ya existían utilizando funciones y operadores. Igualmente se emplean funciones y operadores para escribir condiciones utilizando, además, símbolos para comparar expresiones. Aunque hay una gran cantidad de funciones sólo se describen aquellas que son más usuales:

OPERADORES : + , - , * , / , ** .

COMPARADORES : = , < , > , <= , >= , ^= , & , | , EQ , LT , LG , LE , GE , NE , AND , OR .

FUNCIONES NUMÉRICAS:

ABS(EXPRESIÓN)	Valor Absoluto.
SQRT(EXPRESIÓN)	Raíz Cuadrada.
ROUND(EXPRESIÓN)	Redondear.
ROUND(EXPRESIÓN, PRECISIÓN)	Redondear con cierta precisión (la precisión es una potencia de 10, por ejemplo: 10, 1, 0.1, 0.01, etc.).
EXP(EXPRESIÓN)	Exponencial.
LOG(EXPRESIÓN)	Logaritmo.
LOG2(EXPRESIÓN)	Logaritmo con base 2.
LOG10(EXPRESIÓN)	Logaritmo con base 10.
COS(EXPRESIÓN)	Coseno.
SIN(EXPRESIÓN)	Seno.
TAN(EXPRESIÓN)	Tangente.
LAG(VARIABLE)	Copia la variable en cuestión pero con retraso de una observación.

FUNCIONES ALEATORIAS:

RANBIN(SEMILLA, n, p)	Binomial de parámetros "n" y "p" generada a partir de cierta semilla.
RANNOR(SEMILLA)	Normal de media "0" y desviación estándar "1" generada a partir de cierta semilla.
RANPOI(SEMILLA, a)	Poisson de parámetro "a" generada a partir de cierta semilla.
RANUNI(SEMILLA)	Uniforme con parámetros "0" y "1" generada a partir de cierta semilla.

Cuando la semilla es cero, este valor se toma en realidad aleatoriamente.

FUNCIONES ALFANUMÉRICAS:

INDEX	Busca una expresión de caracteres dentro de una cadena.
COMPRESS	Elimina caracteres específicos de una cadena.
LOWCASE	Convierte todas las letras del argumento a minúsculas.
UPCASE	Convierte todas las letras del argumento a mayúsculas.
LENGTH	Retorna la longitud del argumento.
LEFT	Alinea a la izquierda una expresión de caracteres.
REVERSE	Da la vuelta a una cadena.
SCAN	Selecciona una palabra en particular de una expresión de caracteres.
SOUNDEX	Codifica una cadena a sonidos para facilitar comparaciones.
SPEDIS	Determina la similitud entre dos palabras expresada como una distancia.
SUBSTR	Extrae una subcadena de un argumento.
TRANSLATE	Reemplaza caracteres específicos de un argumento.

2.20 Procedimientos SAS

La sintaxis de los diferentes procedimientos suele ser muy similar. La mayoría de opciones sirven para casi todos los procedimientos, pero con prudencia, ya que cada procedimiento tiene sus particularidades y no siempre estas opciones tienen sentido en cualquier **PROC**. En líneas generales, la estructura de un procedimiento puede ser la siguiente:

Código ejemplo:

```
* USO GENERAL DE LOS PROCEDIMIENTOS;  
  
PROC NOM_PROC DATA=NOMBRE_DATASET OPCIONES  
  ESPECÍFICAS;  
  
* INSTRUCCIONES ESPECÍFICAS DEL PROC;  
* INSTRUCCIONES OPTATIVAS DEL PROC;  
  
WHERE CONDICIÓN;  
BY VARIABLES;  
WEIGHT VARIABLE_PESO;  
OUTPUT OUT=DATASET_SALIDA;  
RUN;
```

Después del nombre del procedimiento se pueden escribir las opciones específicas que usualmente son diferentes para cada procedimiento, excepto la opción **DATA=NOMBRE_DATASET** que ya ha sido comentada anteriormente. Esta opción permite especificar cuál es el dataset de trabajo.

Las instrucciones específicas de cada procedimiento son un requisito indispensable para el funcionamiento del mismo, y suelen estar relacionadas con las variables que se analizan. Generalmente, después de escribir la lista de variables, se escribe el símbolo "/" seguido de otras opciones más concretas que debe realizar el procedimiento.

La instrucción **WHERE** permite estudiar sólo los datos del dataset que cumplen una cierta condición.

La instrucción **BY** se utiliza para partir los datos del dataset en diferentes grupos según los valores que toma la variable allí mencionada. El procedimiento repite su análisis para cada uno de estos subgrupos. **BY** es una opción imprescindible en el procedimiento **SORT** aunque no tiene exactamente este sentido.

En los procedimientos estadísticos se puede incluir la instrucción **WEIGHT**, donde se especifica qué variable da el peso de los datos, es decir, qué variable contiene las frecuencias (datos ponderados).

La instrucción **OUTPUT OUT=DATASET_SALIDA** es una opción muy útil que permite guardar los estadísticos calculados por el procedimiento en un dataset que se llama **DATASET_SALIDA**. En algunos casos, también se pueden almacenar los datos que ha

generado el procedimiento para calcular los estadísticos, pero este paso se menciona dentro de las instrucciones específicas del procedimiento después del símbolo "/" por ejemplo.

Las instrucciones **WEIGHT** (a veces se denomina **FREQ**) y **OUTPUT** no sirven para los procedimientos no estadísticos.

Para cualquier duda en la sintaxis de un procedimiento, consultar el **HELP**. Allá se describen todas las instrucciones que son imprescindibles y todas las opciones que permite el procedimiento.

Procedimientos habituales de SAS/BASE

APPEND Procedure	Añadir datos a un dataset.
CHART Procedure	Gráficos de barras sencillos.
COMPARE Procedure	Comparación de Bases de datos.
CONTENTS Procedure	Contenidos de un dataset.
CORR Procedure	Correlación entre variables.
DATASETS Procedure	Gestión de datasets (eliminar).
EXPORT Procedure	Exportar datasets a texto.
FORMAT Procedure	Dar etiquetas a los valores de las variables.
FREQ Procedure	Tablas de frecuencias.
IMPORT Procedure	Importar datos (p. ej. en formato texto).
MEANS Procedure	Resumir los datos.
PLOT Procedure	Diagramas de dispersión sencillos.
PRINT Procedure	Listar datasets.
PRINTTO Procedure	Definir las rutas donde almacenar las ventanas LOG y OUTPUT .
RANK Procedure	Crea Rangos a partir de variables.
REPORT Procedure	Para realizar informes.
SORT Procedure	Ordenar un dataset.
SQL Procedure	Consultas a datasets mediante instrucciones SQL.
STANDARD Procedure	Produce variables estandarizadas.
SUMMARY Procedure	Estadísticos de Resumen.
TABULATE Procedure	Tablas.
TRANSPOSE Procedure	Transpone datasets.
UNIVARIATE Procedure	Análisis univariados.

Procedimientos habituales de SAS/STAT

ANOVA Procedure	Análisis de la Varianza.
BOXPLOT Procedure	Diagramas de Caja.
CALIS Procedure	Modelos de ecuaciones estructurales utilizando Análisis de la Covarianza.
CANCORR Procedure	Correlación Canónica.
CANDISC Procedure	Análisis Discriminate Canónico.
CATMOD Procedure	Modelos para datos categóricos.
CLUSTER Procedure	Análisis de Conglomerados Jerárquicos.
CORRESP Procedure	Análisis de Correspondencias.

DISCRIM Procedure	Análisis Discriminante.
FACTOR Procedure	Análisis Factorial.
FASTCLUS Procedure	Análisis de Conglomerados no Jerárquicos.
FREQ Procedure	Tablas de Contingencia.
GENMOD Procedure	Modelos Lineales Generalizados.
GLM Procedure	Modelos Lineales Generales.
LIFEREG Procedure	Análisis de Supervivencia Paramétrico.
LIFETEST Procedure	Análisis de Supervivencia no Paramétrico.
LOGISTIC Procedure	Regresión Logística.
MDS Procedure	Escalamiento Multidimensional.
MIXED Procedure	Modelos Mixtos.
MODECLUS Procedure	Análisis de Conglomerados no Paramétrico.
NLIN Procedure	Modelos no Lineales.
NLMIXED Procedure	Modelos Mixtos no Lineales.
NPAR1WAY Procedure	Análisis no Paramétrico de 1 Factor.
PHREG Procedure	Análisis de Supervivencia Semiparamétrico.
PLAN Procedure	Construcción de Diseños y Aleatorización.
PLS Procedure	Regresión mediante <i>Partial Least Squares</i> .
POWER Procedure	Calcular tamaños de muestra.
PRINCOMP Procedure	Componentes Principales.
PRINQUAL Procedure	Componentes Principales para datos cualitativos.
PROBIT Procedure	Análisis de Regresión PROBIT.
REG Procedure	Análisis de Regresión.
RSREG Procedure	Análisis de Superficie-Respuesta.
SIM2D Procedure	Simulaciones bivariadas de variables gaussianas.
STDIZE Procedure	Produce Variables Estandarizadas.
STEPDISC Procedure	Análisis Discriminante por pasos.
TREE Procedure	Dendogramas.
TTEST Procedure	T-Test.
VARCOMP Procedure	Componentes de la Varianza.
VARIOGRAM Procedure	Variogramas.

Procedimientos habituales de SAS/GRAPH

GCHART Procedure	Diagramas de Barras.
GMAP Procedure	Representación de Mapas.
GLOT Procedure	Diagramas de dispersión.
GSLIDE Procedure	Presentaciones y Diapositivas.
G3D Procedure	Superficies.

2.21 Resumir datos

Una vez leída la matriz de datos es usual observar algunos de los estadísticos más sencillos y tener así una idea aproximada de como son las variables analizadas. Esta información también puede ayudar a detectar posibles errores de los datos. El procedimiento que permite obtener estos estadísticos es **PROC MEANS**.

```
* RESUMIR DATOS;
```

```
PROC MEANS DATA=FC.FINAL;
```

```
VAR EDAD;
```

```
CLASS SEXO;
```

```
RUN;
```

```
PROC MEANS DATA=FC.FINAL MAXDEC=2;
```

```
VAR FC1 FC2;
```

```
OUTPUT OUT=M_FC MEAN(FC1 FC2)= M1 M2 STD(FC1 FC2)=STD1  
STD2;
```

```
RUN;
```

En este último caso, se obtiene el fichero M_FC con las medias y las desviaciones estándares de las variables FC1 y FC2.

The screenshot shows the SAS Results Viewer window with the following data tables:

The MEANS Procedure

SEXO	N Obs	N	Mean	Std Dev	Minimum	Maximum
MUMRF	20	20	20.000000	1.414214	18.000000	25.000000
FUJUEH	10	10	21.000000	1.105455	20.000000	24.000000

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
FC1	40	70.50	8.54	50.00	80.00
FC2	40	137.70	11.70	110.00	165.00

Ventana **RESULTS VIEWER** con el resultado de **PROC MEANS**.

Con la instrucción **VAR** se define qué variables son estudiadas. La instrucción **CLASS** es optativa y es muy similar a la opción **BY**, pero la diferencia es que con la primera los resultados en la ventana **OUTPUT** son mucho más compactos. Para la instrucción **OUTPUT OUT=dataset_salida** en este caso es necesario incorporar la lista de estadísticos y los nombres de las variables con que se van a guardar. Cuando se exploran diversas variables al mismo tiempo, es posible incluir estadísticos para ambas variables en el dataset de salida, especificando cada uno de los nombres. **MAXDEC=N** indica la precisión deseada. Los posibles estadísticos a incluir son:

N	Número de observaciones no faltantes.
NMISS	Número de observaciones faltantes.
MIN	Valor mínimo.
MAX	Valor máximo.
RANGE	Rango de valores.
SUM	Suma de los valores.
SUMWGT	Suma de los pesos de los valores.
MEAN	Media.
USS	Suma de cuadrados no corregida.
VAR	Varianza.
STD	Desviación estándar.
STDERR	Error estándar de la media.
CV	Coficiente de variación.
SKEWNESS	Asimetría.
KURTOSIS	Curtosis.
T	t-Student para la hipótesis nula que la media de la variable es igual a cero.
PRT	Probabilidad asociada al valor observado del Estadístico t-Student.

2.22 Tablas de frecuencia: Proc Freq

El procedimiento **FREQ** genera tablas de frecuencias. Este procedimiento, también puede ser útil para observar la calidad de los datos, comprobando si hay valores extraños o faltantes (y poder proceder con su recodificación o corrección).

```
* TABLAS DE FRECUENCIA ;

PROC FREQ DATA=FC.FINAL;
TABLES SEXO /OUT=F_SEXO;
RUN;

PROC FREQ DATA=FC.FINAL ORDER=DATA;
TABLES SEXO /TESTP=(0.4 0.6);
RUN;
```

Para obtener los datos referentes a la tabla de contingencia o la tabla de frecuencias correspondiente en un dataset nuevo es necesaria la opción **OUT=DATASET_SALIDA** (si se realizan diversas tablas al mismo tiempo, sólo se guardan los datos referentes a la última tabla especificada).

Las opciones TESTP o TESTF permiten contrastar si la distribución de los datos es homogénea con otra distribución poblacional definida por el usuario.

TESTP=(p1 p2 p3 ... pn) representan las probabilidades de los n posibles valores de la variable analizada.

TESTF=(f1 f2 f3 ... fn) representan las frecuencias relativas de los n posibles valores de la variable analizada.

2.23 Tablas de contingencia: Proc Freq

Por otra parte este procedimiento también permite hacer tablas de contingencia a partir de diversas variables discretas.

* TABLAS DE CONTINGENCIA;

```
PROC FREQ DATA=FC.FINAL;
TABLES FARMACO*SEXO /CHISQ;
OUTPUT OUT=X_FARMACO CHISQ;
RUN;
```

```
PROC FREQ DATA=FC.FINAL;
TABLES FARMACO*SEXO /CHISQ OUT=F_FARMACO;
RUN;
```

```
PROC PRINT DATA=F_FARMACO;RUN;
PROC PRINT DATA=X_FARMACO;RUN;
```

Tal y como se ha comentado en el apartado anterior, escribiendo la opción **OUTPUT OUT=DATASET_SALIDA** se obtienen los estadísticos calculados (siempre y cuando hayan sido requeridos y especificados a continuación).

The FREQ Procedure				
Table of FARMACO by SEXO				
Frequency	FARMACO (TRATAMIENTO ADMINISTRADO)		SEXO	
Percent	HOMBRE	MUJER	Total	
Row Pct				
Col Pct				
	1	12	9	21
		30.00	20.00	50.00
		60.00	40.00	
		54.55	44.44	
	2	10	13	23
		25.00	25.00	50.00
		60.00	50.00	
		45.45	55.56	
Total		22	19	41
		56.00	45.00	101.00

Ventana **RESULTS VIEWER** con la tabla de contingencia FARMACO vs SEXO.

Statistics for Table of FARMACO by SEXO

Statistic	DF	Value	P-value
Chi-Square	1	0.4043	0.5253
Likelihood Ratio Chi-Square	1	0.4048	0.5245
Continuity Adj. Chi-Square	1	0.1913	0.7538
Mantel-Haenszel Chi-Square	1	0.3933	0.5322
Phi Coefficient		0.1925	
Contingency Coefficient		0.1923	
Cramer's V		0.1925	

Fisher's exact test	
Cell (1,1) Frequency (N)	12
Left-sided Pr >= F	0.8257
Right-sided Pr <= F	0.3756
Table Probability (P)	0.2253
Two-sided Pr <= P	0.7542

Sample Size = 40

Ventana **RESULTS VIEWER** con los estadísticos correspondientes a / **CHISQ** de la tabla FARMACO vs SEXO.

Obs	FARMACO	SEXO	COUNT	PERCENT
1	1	HOMBRE	12	30
2	1	MUJER	8	20
3	2	HOMBRE	10	25
4	2	MUJER	10	25

Ventana **RESULTS VIEWER** con el resultado del **PROC PRINT** del dataset de salida F_FARMACO.

Obs	R	PC28	DI	PCH	P	PCH	LIQCH	DI	LIQCH	P	LIQCH	ALQCH	DI	ALQCH	P	ALQCH	BIQCH	DI	BIQCH	P	BIQCH	SIS	TISS	OSIS	LI	
1	46	0.46404		0.1354	0.0076		1	0.12405	0.10101		0.75357	0.3434		0.5782	0.4980	0.775										

Ventana **OUTPUT** con parte del resultado del **PROC PRINT** del dataset de salida X_
FARMACO.

En la instrucción **TABLES** es donde se hace referencia a las variables con las cuales se trabaja (es posible solicitar diversas tablas al mismo tiempo). Si se desea una tabla de contingencia es imprescindible escribir el símbolo ' * ' entre las variables a representar. En este último caso, es útil solicitar también el cálculo del estadístico Chi-Cuadrado escribiendo ' / **CHISQ** ' al final de la instrucción **TABLES**.

2.24 Tablas de estadísticos de resumen

El procedimiento **TABULATE** diseña tablas al gusto del usuario. Este procedimiento dispone de muchas opciones, pero sólo se describen algunos casos simples:

* PRESENTACIÓN DE RESULTADOS EN TABLAS;

```
PROC TABULATE DATA=FC.FINAL;
CLASS FARMACO SEXO;
TABLE FARMACO, (N*SEXO ALL);
RUN;
```

```
PROC TABULATE DATA=FC.FINAL;
CLASS FARMACO;
VAR FC1 FC2;
TABLE FARMACO, MEAN*(FC1 FC2);
RUN;
```

FARMACO ADMINISTRADO	N		N
	FEMEAS	MASCOS	
1	12	8	20
2	10	10	20

Ventana **RESULTS VIEWER** con el resultado del primer **PROC TABULATE**.

La instrucción **CLASS** indica cuáles de las variables de la tabla clasifican los datos.

La instrucción **VAR** indica las variables numéricas que se quiere describir.

En la instrucción **TABLE** se indica la expresión de las variables que generan la tabla, introduciendo las variables representadas como filas y las representadas en columnas separadas por una coma. Estas expresiones pueden estar formadas por variables **CLASS**, variables **VAR**, la palabra **ALL** (que se refiere al total) y estadísticos, poniendo el símbolo "*" para combinar y paréntesis para encadenar.

Algunos de los estadísticos disponibles son:

MAX	Valor máximo.
MIN	Valor mínimo.
MEAN	Media.
N	Número de observaciones.
PCTN	Porcentaje de observaciones.
SUM	Suma de las observaciones.
PCTSUM	Porcentaje de la suma de las observaciones.
RANGE	Rango.
STD	Desviación estándar.
STDERR	Error estándar.
VAR	Varianza.

Algunas de las opciones disponibles son:

NOSEPS	Opción de la sentencia PROC TABULATE, elimina las líneas horizontales entre filas y el cuerpo de la tabla.
MISSING	Opción de la sentencia PROC TABULATE, considera los valores MISSING como valores válidos en las variables CLASS.
OUT	Opción de la sentencia PROC TABULATE, guarda los estadísticos de salida de la tabla en un dataset.
BOX	Opción de la sentencia TABLE, escribe un texto en el rectángulo vacío que hay encima de las variables fila.
INDENT	Opción de la sentencia TABLE, especifica un número de espacios para indexar variables fila anidadas.
RTS	Opción de la sentencia TABLE, especifica en número de espacios TOTAL para las variables fila.

*PRESENTACIÓN DE RESULTADOS EN TABLAS CON DIFERENTES OPCIONES;

```

PROC FORMAT;
VALUE FARMACOF
    1='TRACTAM. A'
    2='TRACTAM. B';
RUN;

PROC TABULATE DATA=FC.FINAL NOSEPS;
CLASS FARMACO SEXO;
VAR FC1 FC2;
TABLE (FARMACO *SEXO=' ') ALL="TOTAL",
    (FC1="FC INICIAL" FC2="FC FINAL") *
    ((MEAN="MITJANA" STD="DESV.TIPICA") *F=11.2) / INDENT=10;
FORMAT FARMACO FARMACOF.;
RUN;

```

En esta primera tabla, se han utilizado las siguientes opciones:

- Se ha eliminado las líneas horizontales del cuerpo de la tabla (opción NOSEPS).
- Se han puesto etiquetas a las variables FC1 y FC2 , y a los estadísticos media y desviación.
- Se ha dado formato a la variable FARMACO (formato FARMACOF), y a los estadísticos media y desviación (longitud 11 con dos decimales).
- Se ha indexado la variable anidada por filas, SEXO (con la opción INDENT).

Y se obtiene el siguiente resultado:

FARMACO	SEXO	FC INICIAL		FC FINAL	
		MEAN	STD	MEAN	STD
TRACTAM. A	INICIAL	70.67	0.07	82.00	12.30
	FINAL	75.63	0.20	81.00	16.00
TRACTAM. B	INICIAL	75.53	0.06	83.00	11.50
	FINAL	82.63	0.05	84.00	8.00
TOTAL		76.55	0.21	83.00	12.00


```

PROC TABULATE DATA=FC.FINAL NOSEPS ;
CLASS FARMACO /ORDER=FREQ;
CLASS SEXO;
TABLE (FARMACO='TRATAMIENTO ADMINISTRADO' ),
      (SEXO=' ' ALL="TOTAL") *
      ( (N="N"*F=6.0 PCTN<SEXO ALL>="%"*F=6.2) )
      /RTS=30 BOX="TRATAMIENTO SEGUN SEXO";
FORMAT FARMACO FARMACOF.;
RUN;

```

En esta segunda tabla, se han utilizado las siguientes opciones:

- Se han eliminado las líneas horizontales del cuerpo de la tabla (opción NOSEPS).
- Se ha puesto etiqueta a la variable TRATAM , y a los estadísticos número de casos y porcentaje.
- Se ha dado formato a la variable TRATAM (formato TRATAMF), y a los estadísticos número de casos y porcentaje (longitud 6 sin decimales, y longitud 6 con dos decimales, respectivamente).
- Se ha calculado el total de porcentajes por tratamiento (la opción que hay entre “<>” después del estadístico PCTN).
- Se ha escrito un texto que describe la tabla, en el rectángulo superior izquierdo (opción BOX).
- Se ha definido el número de caracteres para variables fila de 30 (opción RTS).

Y se obtiene el siguiente resultado:

TRATAMIENTO SEGUN SEXO					
TRATAMIENTO ADMINISTRADO	HOMBRE		MUJER		TOTAL
	N	%	N	%	N
TRACTAM A	12	60.00	8	40.00	20
TRACTAM B	10	50.00	10	50.00	20

2.25 Análisis Univariante

Cuando el objetivo del análisis es por ejemplo testar una hipótesis o bien cualquier otro análisis más profundo de los datos, es interesante explorar más detalladamente las variables y obtener otros estadísticos descriptivos así como tests relativos a la distribución de los datos o bien representaciones gráficas. El procedimiento que se utiliza para este tipo de exploración es el **PROC UNIVARIATE**.

* ANÁLISIS UNIVARIANTE ;

```
PROC UNIVARIATE DATA=FC.FINAL;
VAR EDAD;
OUTPUT OUT=U_EDAD MEAN=MEDIA MEDIAN=MEDIANA MODE=MODA;
RUN;
```

The screenshot shows the SAS Results Viewer window with the following output for the variable EDAD:

Moments			
N	40	Sum Weights	40
Mean	22.415	Sum Observations	896.6
Std Deviation	1.43934533	Variance	2.06797436
Skewness	0.10543212	Kurtosis	2.15144474
Unadjusted SS	27177.54	Corrected SS	60.651
Coeff Variation	6.41554806	Std Error Mean	0.22737450

Basic Statistical Measures			
Location		Variability	
Mean	22.41500	Std Deviation	1.43805
Median	22.55000	Variance	2.06797
Mode	21.42000	Range	6.10000
Interquartile Range: 1.80000			

Tests for Location: Shift=0			
Test	Statistic	p-Value	
Student's t	t	98.58187	P <= t <.0001
Sign	M	22	P <= M <.0001
Signed Rank	S	412	P <= S <.0001

Quantiles (Definition 5)	
Quantile	Estimate
100% Max	26.80
95%	26.80
90%	25.15
75% Q3	23.20
50% Median	22.55
25% Q1	21.40

Ventana **RESULTS VIEWER** con parte del resultado de **PROC UNIVARIATE**.

```
PROC PRINT DATA=U_EDAD;
RUN;
```

```
PROC UNIVARIATE DATA=FC.FINAL PLOT NORMAL;
VAR EDAD;
RUN;
```

Muchos de los estadísticos que produce el procedimiento **UNIVARIATE** también se obtienen con el procedimiento **MEANS**. El procedimiento **UNIVARIATE** incorpora estadísticos tales como los resultados de diferentes tests no paramétricos, tests de normalidad, y estadísticos descriptivos como percentiles, moda y valores más extremos.

Los estadísticos a incluir en el ***DATASET_SALIDA*** mediante la instrucción **OUTPUT** son:

N	Número de observaciones no faltantes.
NMISS	Número de observaciones faltantes.
NOBS	Número de observaciones.
MEAN	Media.
STDMEAN	Desviación estándar de la media.
SUM	Suma de los valores.
STD	Desviación estándar.
VAR	Varianza.
CV	Coefficiente de variación.
USS	Suma de cuadrados no corregida.
CSS	Suma de cuadrados corregida.
SKEWNESS	Skewness.
KURTOSIS	Kurtosis.
SUMWGT	Suma de los pesos de los valores.
MAX	Valor máximo.
MIN	Valor mínimo.
RANGE	Rango de valores.
Q3	Percentil del 75% (3er Cuartil).
MEDIAN	Mediana.
Q1	Percentil del 25% (1er Cuartil).
QRANGE	Diferencia entre Q3 y Q1.
P1	Percentil 1.
P5	Percentil 5.
P10	Percentil 10.
P90	Percentil 90.
P95	Percentil 95.
P99	Percentil 99.
MODE	Moda.
T	t-Student para la hipótesis nula que la media de la variable es igual a cero.
PROBT	Probabilidad que el valor absoluto de t sea superior a T.
MSIGN	Estadístico de signo.
PROBM	Probabilidad que el valor absoluto del estadístico de signo sea superior a MSIGN.
SIGNRANK	Estadístico de signo del rango.
PROBS	Probabilidad que el valor absoluto del estadístico de signo del rango sea superior a SIGNRANK centrado.

El procedimiento **UNIVARIATE** realiza tres tipos diferentes de representaciones de los datos si se especifica la opción **PLOT**: un diagrama de tallo y hojas, un diagrama de cajas y una gráfica para comparar la función de probabilidades empírica de los datos con la función de probabilidades de una variable aleatoria con distribución Normal con la misma media y desviación que la variable descrita.

Especificando la opción **NORMAL** en la instrucción **PROC UNIVARIATE**, se presentan dos estadísticos nuevos para testar si los datos provienen o no de una distribución normal, y en este caso también es posible incluirlos en el **DATASET_SALIDA**:

W Estadístico para testar si los datos provienen de una variable Normal.
PROBW Significación del test de normalidad bajo la hipótesis nula que los datos provienen de una variable Normal.

El procedimiento **UNIVARIATE** posibilita el contraste de diferencias entre medias de dos variables relacionadas (datos apareados o medidas repetidas) o bien el contraste para ver si la media de una cierta variable es igual a una determinada constante (en estos casos sólo existe un grupo de observaciones y no dos) estudiando la variable generada a partir de la diferencia (de las dos variables de medidas repetidas o bien de la variable y el valor de contraste) y observando el resultado del t-Test que proporciona. También se permite obtener un histograma mediante la instrucción:

* HISTOGRAMA ;

```
PROC UNIVARIATE DATA=FC.FINAL;
CLASS SEXO;
VAR FC2;
HISTOGRAM / NORMAL CFILL=GRAY;
RUN;
```

