



**SERVEI D'ESTADÍSTICA
UNIVERSITAT AUTÒNOMA DE BARCELONA**

INTRODUCCIÓN AL SAS ®

Autores: Joan Valls
Llorenç Badiella
Servei d'Estadística UAB

1. INTRODUCCIÓN.

Durante los últimos años la estadística aplicada ha ganado mucha importancia en el mundo de la investigación, probablemente debido a la gran potencia de cálculo de los ordenadores y en concreto a los diferentes paquetes estadísticos.

Paralelamente, los paquetes estadísticos también se han adaptado a las nuevas necesidades, de manera que ahora ya no se habla de "paquetes estadísticos", término demasiado concreto. Por ejemplo, SAS ® inicialmente significaba "Statistical Analysis System", pero a medida que este programa iba incorporando nuevas capacidades la mayoría de las cuales no eran propiamente estadísticas, SAS ® pasó a convertirse en una sola palabra.

Actualmente, los diferentes módulos de SAS ® hacen que éste sea un software de los que se llaman como "de inicio a fin". Permite crear gráficos, trabajar como una hoja de cálculo, compilar programas en lenguaje C, incluye herramientas para construir interfases para la WWW, herramientas para tratar el Datawarehouse o para explotar datos con la filosofía del Datamining, etc.

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

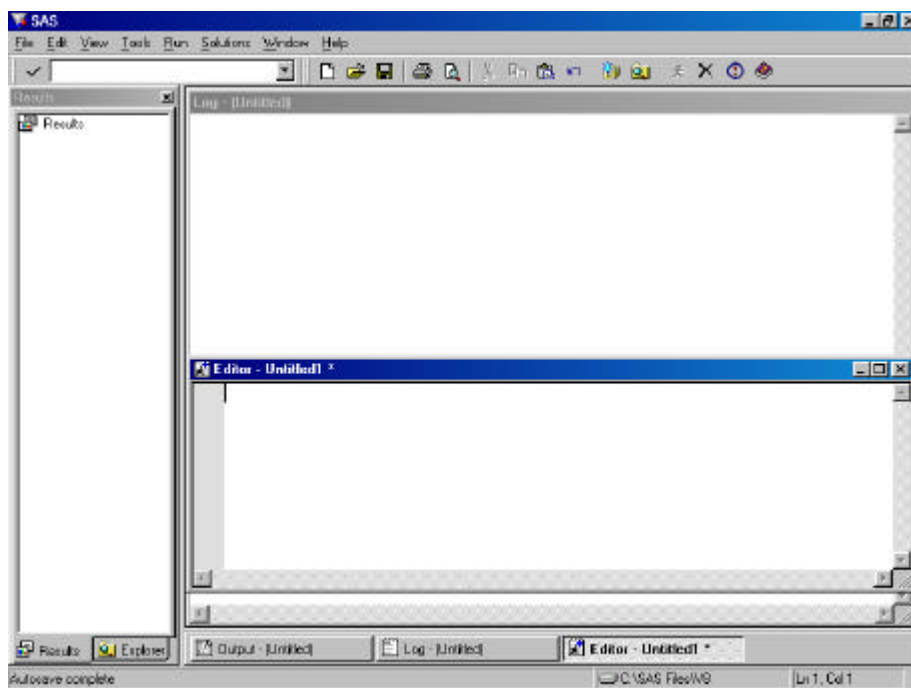
El manual está dividido en diferentes apartados bastante concretos: leer datos, tabular los datos, etc. pero en cada uno de ellos se introducen diversas características generales con el objetivo de avanzar paralelamente en otros detalles necesarios del funcionamiento del programa. Por este motivo es recomendable seguir el manual de forma lineal y consecutiva.

2. ÍNDICE.

1. INTRODUCCIÓN.....	2
2. ÍNDICE.	3
3. INICIO DE LA SESIÓN.....	5
4. SAS BASE.....	7
4.1. LOS PROCEDIMIENTOS Y EL PASO DATA.....	7
4.2. COM OBTENER DATOS CON SAS ®.	7
4.2.1. LEER DATOS INTERNOS.....	7
4.2.2. LEER DATOS EXTERNOS.....	9
4.3. LISTAR LOS CONTENIDOS DE UN DATASET.....	9
4.4. GUARDAR DATOS INTERNAMENTE: DATASETS TEMPORALES Y DATASETS PERMANENTES.	11
4.5. LA INSTRUCCIÓN SET DEL PASO DATA.	11
4.6. GUARDAR DATOS EXTERNAMENTE.	13
4.7. ORDENAR UN DATASET.....	13
4.8. MEZCLAR DATASETS.....	14
4.8.1. AÑADIR VARIABLES.....	14
4.8.2. AÑADIR CASOS.....	15
4.9. DAR ETIQUETAS A LAS VARIABLES Y A SUS VALORES.	16
4.10. ASIGNACIONES Y CONDICIONES.....	17
4.11. FUNCIONES DE SAS ®.....	18
4.12. GENERACION DE VARIABLES ALEATORIAS.....	19
4.13. LOS PROCEDIMIENTOS.....	21
5. SAS/STAT I: PRESENTACIÓN DE RESULTADOS.....	23
5.1. RESUMIR LOS DATOS.....	23
5.2. PROC TABULATE: TABLAS.....	24
6. SAS/STAT II: ANÁLISIS UNIVARIANTE.....	25
6.1. PROC UNIVARIATE.....	25
6.2. TABLAS DE FRECUENCIAS: PROC FREQ.....	28
7. SAS/STAT III: ANÁLISIS BIVARIANTE.....	29
7.1. TABLAS DE CONTINGENCIA: PROC FREQ.....	29
7.2. COMPARACIÓN DE MEDIAS.....	31
7.3. RELACIONES ENTRE VARIABLES CONTINUAS.	33
7.3.1. CORRELACIÓN.....	33
7.3.2. REGRESIÓN SIMPLE.....	35
8. PROCEDIMIENTOS GRÁFICOS (SAS/GRAPH)	39
8.1. GRÁFICAS DE BARRAS Y PASTELES.....	39
8.2. GRÁFICAS PARA DOS VARIABLES.....	40
9. ANEXO.....	44
9.1. IMPORTAR DATOS DE SPSS.....	44
9.2. DATOS DEL MANUAL.....	45
10. ANEXO 2: PROGRAMACIÓN MACRO	47
10.1. INTRODUCCIÓN.....	47
10.1.1. Variables Macro.....	47
10.1.2. Generación de Código mediante Macros.....	47
10.1.3. Macros con código SAS.....	48

10.1.4.	Macros con parámetros de entrada.....	48
10.1.5.	Comentarios en las macros.....	48
10.1.6.	Generación de código mediante condiciones.....	49
10.1.7.	Generación de código repetitivo.....	49
10.2.	VARIABLES MACRO	50
10.2.1.	Variables Macro definidas por el sistema.....	50
10.2.2.	Variables Macro definidas por el usuario.....	50
10.2.3.	Asignar valores a Variables Macro	50
10.2.4.	Referencias a variables Macro	51
10.2.5.	Visualizar Variables Macro.....	51

3. INICIO DE LA SESIÓN.





Ventana inicial.

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 (ver capítulo 6) y una ventana de resultados dónde aparece la información obtenida de las diferentes ejecuciones desglosadas.
- Ventana derecha: Contiene las ventanas principales **LOG**, **OUTPUT**, **EDITOR**.

El modo de trabajo que utiliza SAS ® se basa en éstas tres ventanas:

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

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, ...

TOOLS: table editor (permite explorar bases de datos en formato SAS ®).

RUN: run (ejecutar), recall (pega la sintaxis de la última ejecución en la ventana de sintaxis).

SOLUTIONS: Analysis -> Analyst (aparece una ventana interactiva para el análisis de datos).

WINDOW: permite cambiar de ventana (Log, Output, Editor,...).

HELP : SAS ® SYSTEM HELP (ayuda de SAS ®), Books and training -> SAS ® Online Doc (Manual completo pero muy extenso de SAS ®).

MÓDULOS DE SAS

SAS/ACCESS	Interfase a otros formatos de ficheros
SAS/AF	Lenguaje para el diseño de aplicaciones SAS
SAS/ASSIST	Asistencia para el uso de SAS
SAS/CONNECT	Conjunto de herramientas cliente/servidor
SAS/EIS	Permite construir Bases de datos multidimensionales (MDDb) y relacionales (RDBMS)
SAS/ETS	Análisis de Series Temporales
SAS/FSP	Facilidades para la entrada de datos de forma interactiva
SAS/GIS	GIS (Sistema de Información Geográfica) interactivo con SAS
SAS/GRAPH	Módulo gráfico
SAS/IML	Manipulación de matrices
SAS/INSIGHT	Herramienta para explorar y analizar datos
SAS/MDDb	Trabajo con Bases de datos Multidimensionales
SAS/ODBC	Conexiones ODBC
SAS/OR	Para resolver problemas de optimización
SAS/QC	Control de Calidad
SAS/SPECTRAVIEW	Visualización de datos cómo representaciones gráficas
SAS/STAT	Análisis Estadísticos
SAS/IntrNet	Permite a los usuarios ejecutar programas desde un navegador

Otros Productos SAS

SAS Enterprise Miner	DataMining
SAS Enterprise Guide	Facilidad para el análisis estadístico

4. SAS BASE

4.1. LOS PROCEDIMIENTOS Y EL PASO DATA.

Básicamente, todos los programas de sintaxis elaborados con SAS ® contienen tan sólo dos tipos de instrucciones: **DATA** y **PROC**.

Con la instrucción **DATA nombre** se crea un dataset nuevo (o bien se reemplaza). "**nombre**" es el nombre del nuevo dataset. Un **dataset** es un conjunto de datos creado con SAS ®. Cualquier operación a realizar con un dataset: lectura de datos, creación de variables nuevas, recodificación, cambio de etiquetas, selección de casos, ... se hace dentro de la instrucción **DATA**.

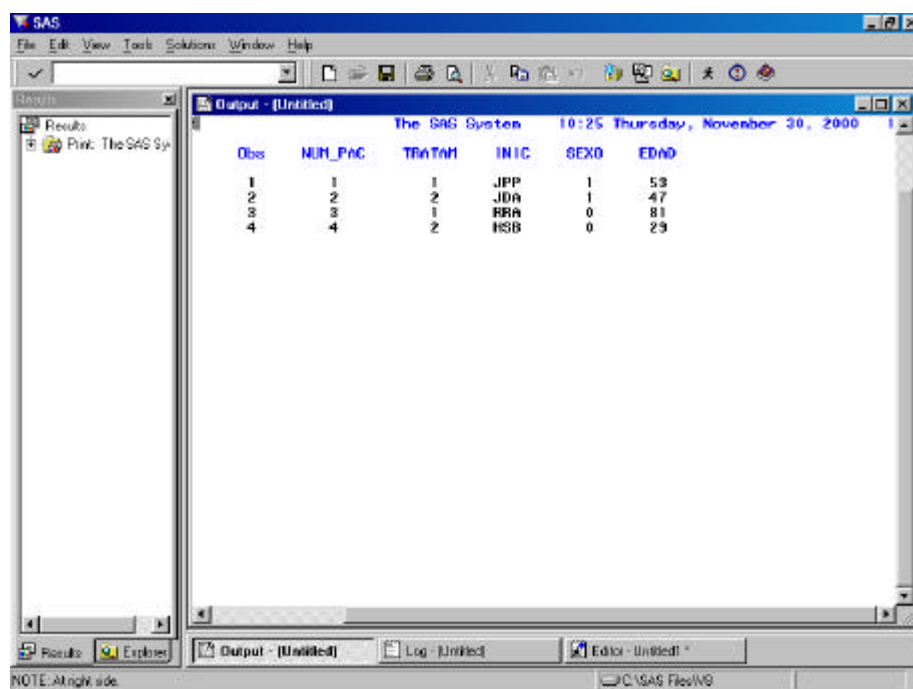
Los procedimientos **PROC** se emplean para trabajar con los datos de un dataset sin modificar su estructura: análisis, tablas, listados, cálculos, estadísticos, ...

4.2. COM OBTENER DATOS CON SAS ®.

4.2.1. LEER DATOS INTERNOS.

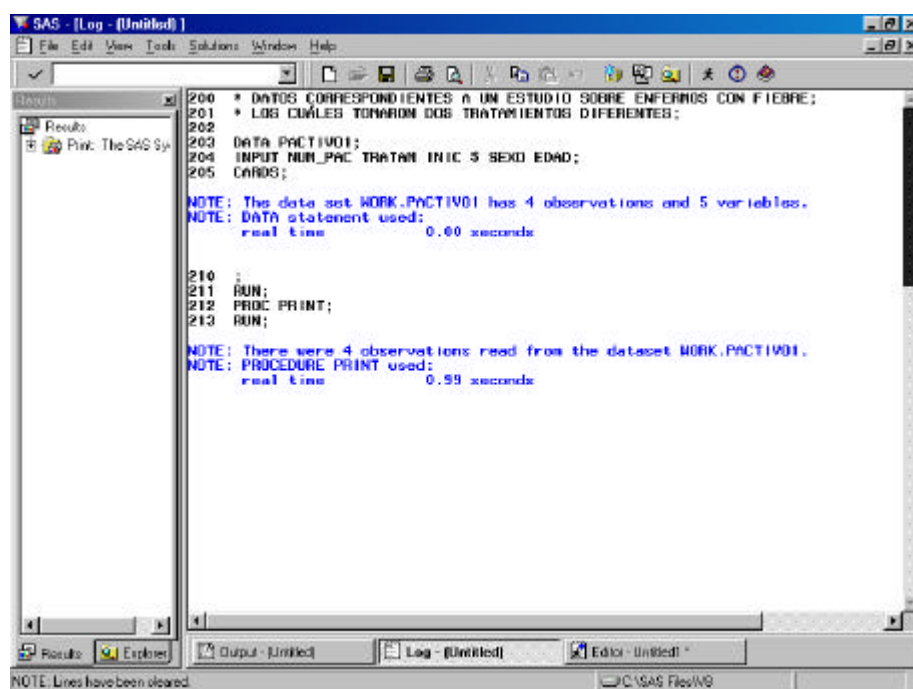
```
* DATOS CORRESPONDIENTES A UN ESTUDIO SOBRE ENFERMOS CON FIEBRE;  
* LOS CUÁLES TOMARON DOS TRATAMIENTOS DIFERENTES;
```

```
DATA PACTIVO1;  
INPUT NUM_PAC TRATAM INIC $ SEXO EDAD;  
CARDS;  
1 1 JPP 1 53  
2 2 JDA 1 47  
3 1 RRA 0 81  
4 2 HSB 0 29  
;  
RUN;  
  
PROC PRINT;  
RUN;
```



Obs	NUM_PAC	TRATAM	INIC	SEXO	EDAD
1	1	1	JPP	1	53
2	2	2	JDA	1	47
3	3	1	RRR	0	81
4	4	2	HSB	0	29

Ventana **OUTPUT** dónde se puede visualizar el resultado de **PROC PRINT**.



```

200 * DATOS CORRESPONDIENTES A UN ESTUDIO SOBRE ENFERMOS CON FIEBRE;
201 * LOS CUALES TOMARON DOS TRATAMIENTOS DIFERENTES;
202
203 DATA PACTIVO1;
204 INPUT NUM_PAC TRATAM INIC $ SEXO EDAD;
205 CARDS;

NOTE: The data set WORK.PACTIVO1 has 4 observations and 5 variables.
NOTE: DATA statement used:
      real time    0.00 seconds

210 ;
211 RUN;
212 PROC PRINT;
213 RUN;

NOTE: There were 4 observations read from the dataset WORK.PACTIVO1.
NOTE: PROCEDURE PRINT used:
      real time    0.59 seconds

```

Ventana **LOG** dónde se puede visualizar el estado de la ejecución.

Nota: Después de cada instrucción es necesario el signo ";".

Nota: Los comentarios van precedidos por un signo "*".

Nota: Después de un paso **DATA** o de un **PROC** es útil escribir "RUN;" en el caso que interese ejecutar el programa a trozos.

La instrucción **INPUT** declara como se leen las variables: 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 procedimiento **PROC PRINT** es el procedimiento que lista los datos.

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

4.2.2. LEER DATOS EXTERNOS.

```
DATA PACTIVO2;  
INFILE 'A:\DATOS.DAT' LRECL=9;  
INPUT NUM_PAC 1-2 INIC $ 4-6 SEXO 7 EDAD 8-9;  
RUN;  
  
PROC PRINT DATA=PACTIVO2;  
RUN;  
  
DATA PACTIVO3;  
INFILE 'A:\DATOS2.DAT' DLM='09'x;  
INPUT NUM_PAC TRATAM;  
RUN;  
  
PROC PRINT DATA=PACTIVO3;  
RUN;
```

La instrucción **INFILE** se utiliza para la lectura de datos externos y en ella se menciona la ruta dónde 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. Se escriben las columnas dónde se encuentran las variables si el fichero de datos externo es de formato fijo. En el caso en que el fichero de datos está delimitado, no tiene sentido especificar las columnas. 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 se encuentra delimitado por tabuladores o **DLM=';'** si el fichero se encuentra delimitado por el símbolo ";".

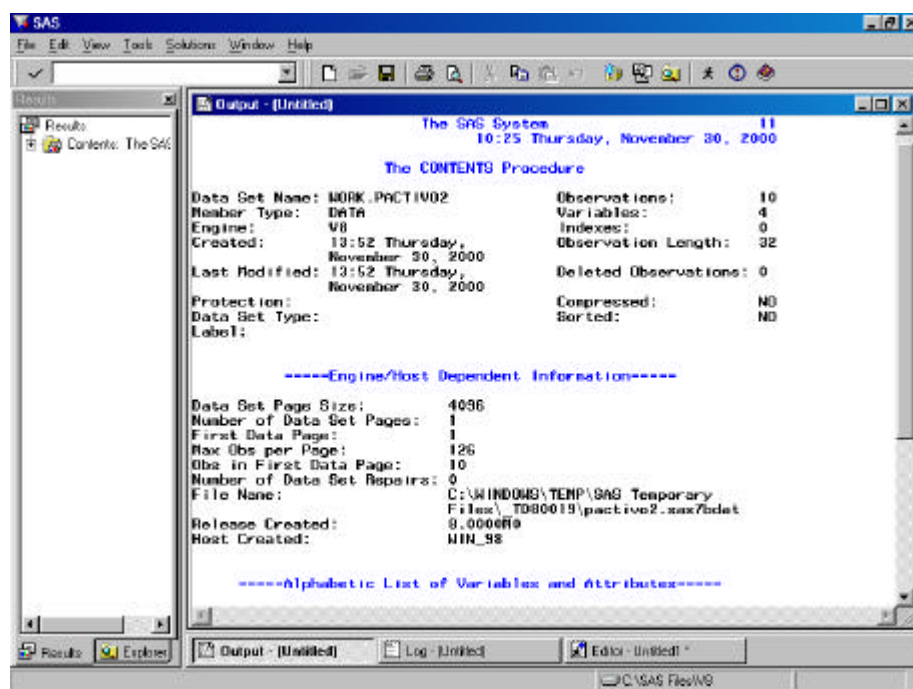
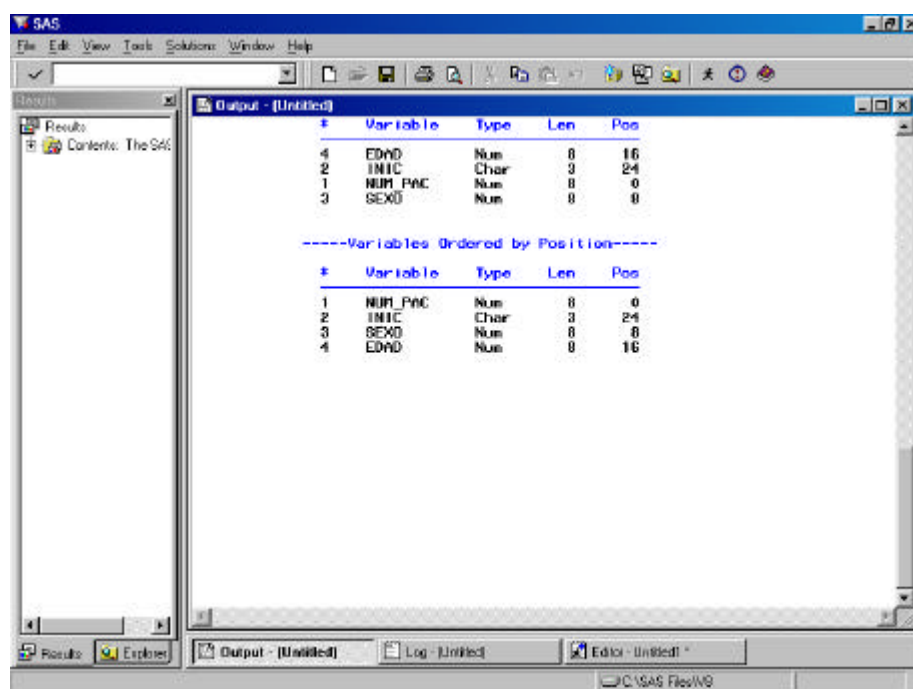
Cualquier procedimiento trabaja con el dataset deseado utilizando la opción **DATA=nombre_dataset**. Por defecto, SAS ® utiliza el dataset creado en el paso **DATA** más reciente.

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).

4.3. LISTAR LOS CONTENIDOS DE UN DATASET.

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

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 **POSITION**, 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.

Información obtenida con **PROC CONTENTS (I)**.Información obtenida con **PROC CONTENTS (II)**.

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).

4.4. GUARDAR DATOS INTERNAMENTE: DATASETS TEMPORALES Y DATASETS PERMANENTES.

Nombres de los datasets, librerías y tipos:

PASO DATA	LIBRERIA	NOMBRE	TIPO
DATA PATATA	WORK	PATATA	TEMPORAL
DATA WORK.TOMATE	WORK	TOMATE	TEMPORAL
DATA VERDURAS.CEBOLLA	VERDURAS	CEBOLLA	PERMANENTE

Nota: Hasta ahora todos los datasets creados han sido temporales.

Nota: Los nombres de los datasets y de las variables no pueden tener más de 8 caracteres.

Los datasets temporales se encuentran en una librería virtual llamada WORK que se vacía automáticamente cuando se apaga el programa. Para crear un dataset temporal, el nombre tiene que ser simple: "a", "pera" y "patata" son nombres válidos o bien compuesto con la librería work: "work.pera" es equivalente a "pera" (de hecho, son el mismo dataset).

Para crear un dataset permanente, en primer lugar se crea una librería con la instrucción LIBNAME especificando el nombre y la ruta dónde ubicarla a continuación. A partir de este momento, para crear un dataset permanente que pertenezca a esta librería se utiliza el nombre compuesto: "*nombre_librería.nombre_dataset*".

4.5. LA INSTRUCCIÓN SET DEL PASO DATA.

```
LIBNAME FIEBRE 'A:\';

DATA FIEBRE.PACTIVO1;
SET PACTIVO1;
RUN;

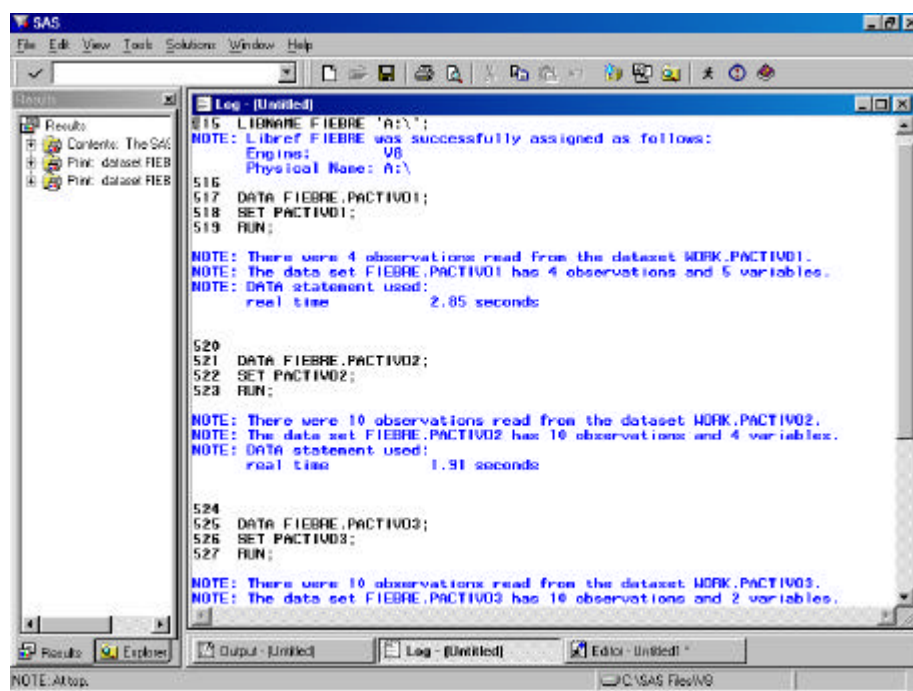
DATA FIEBRE.PACTIVO2;
SET PACTIVO2;
RUN;

DATA FIEBRE.PACTIVO3;
SET PACTIVO3;
RUN;

OPTIONS LS=75 PS=50 NODATE NONUMBER;
TITLE 'dataset FIEBRE.PACTIVO2';
FOOTNOTE 'dataset PERMANENTE';

PROC PRINT DATA=FIEBRE.PACTIVO2;
RUN;

TITLE;
FOOTNOTE;
```

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

Con las instrucciones **TITLE**, **FOOTNOTE** y **OPTIONS** se controlan la ventana **OUTPUT**. Las dos primeras instrucciones permiten poner títulos y pies de página.

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

Ahora los datasets **PACTIVO1**, **PACTIVO2** y **PACTIVO3** se encuentran en las librerías **WORK** y **FIEBRE**, 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 **FIEBRE**, físicamente se encuentran en la ruta especificada en la instrucción **LIBNAME** (A:\) cuando se creó la librería.

La instrucción **SET** asigna al dataset del paso **DATA** los mismos datos, variables y propiedades que el dataset de la instrucción **SET**.

Código ejemplo:

```

DATA B;
SET A;
RUN;

```

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

Código ejemplo:

```

DATA B;
SET B;
VAR_A=1;
RUN;

```

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

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

Si se desea recuperar un fichero de datos creado con SAS ® para utilizarlo o continuar trabajando (por lo tanto el fichero tiene la extensión **".sd2"** en la versión 6.12 o **".sas7bdat"** en la versión 8) que se encuentra ubicado por ejemplo en " A:\ " y se llama **dataset_viejo** (físicamente, el fichero es: " **A:\dataset_viejo.sd2** " o " **A:\dataset_viejo.sas7bdat** " según la versión de SAS ®) se puede utilizar el siguiente código:

Código ejemplo:

```
LIBNAME nombre_libreria 'A:\';

DATA nombre_dataset;
SET nombre_libreria.dataset_viejo;
RUN;
```

Después de ejecutar el código anterior el **dataset_viejo** continua en su ubicación original sin modificar, pero se ha creado una copia temporal llamada **nombre_dataset** en la librería **WORK**.

4.6. GUARDAR DATOS EXTERNAMENTE.

```
DATA AUXILIAR;
SET FIEBRE.PACTIVO1;
FILE 'A:\DATOS1.DAT' ;
PUT NUM_PAC 1-2 TRATAM 3 INIC $ 4-6 SEXO 7 EDAD 8-9;
RUN;
```

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).

4.7. ORDENAR UN DATASET.

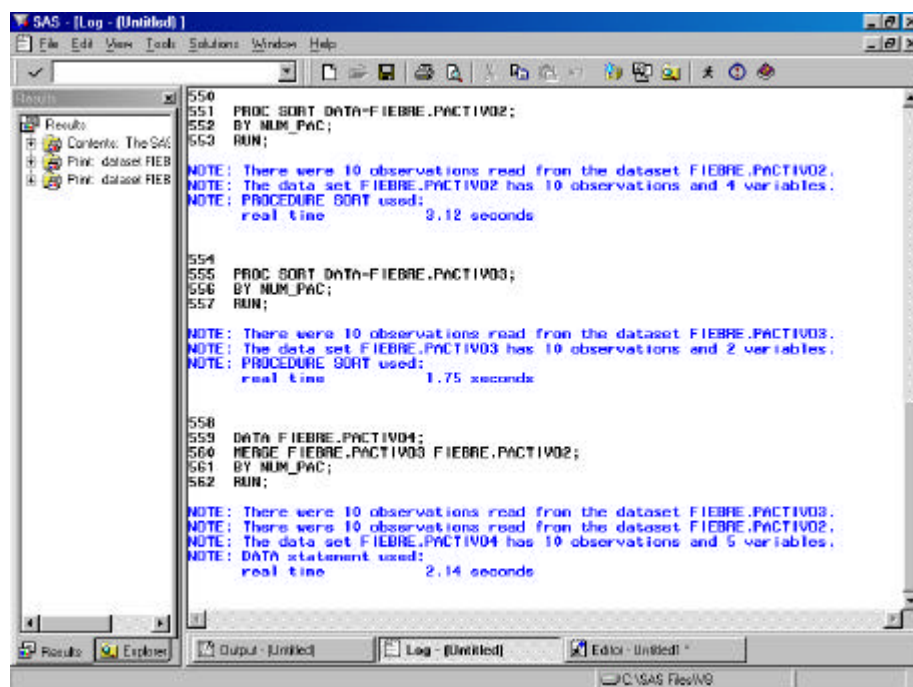
```
PROC SORT DATA=FIEBRE.PACTIVO1;
BY NUM_PAC;
RUN;
```

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

4.8. MEZCLAR DATASETS.

4.8.1. AÑADIR VARIABLES.

```
PROC SORT DATA=FIEBRE.PACTIVO2;  
BY NUM_PAC;  
RUN;  
  
PROC SORT DATA=FIEBRE.PACTIVO3;  
BY NUM_PAC;  
RUN;  
  
DATA FIEBRE.PACTIVO4;  
MERGE FIEBRE.PACTIVO3 FIEBRE.PACTIVO2;  
BY NUM_PAC;  
RUN;  
  
PROC PRINT;  
RUN;
```



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

La instrucción **MERGE** permite añadir variables. Si los datasets que contienen las diferentes variables no tienen el mismo número de observaciones, es recomendable utilizar la instrucción **BY variables**. Las variables referidas en la instrucción **BY**, tienen que estar presentes en ambos datasets, permitiendo identificar los diferentes casos dentro de cada dataset y asignarles una cierta correspondencia. En caso de utilizar esta última opción, los dos datasets deben estar ordenados por las variables mencionadas en **BY** antes de realizar la fusión.

4.8.2. AÑADIR CASOS.

```

DATA FIEBRE.PACTIVO5;
SET FIEBRE.PACTIVO4;
RUN;

PROC APPEND BASE=FIEBRE.PACTIVO5 DATA=FIEBRE.PACTIVO1 FORCE;
RUN;

PROC PRINT;
RUN;

```

Ventana **OUTPUT** con el resultado de la unión de los datasets.

Para añadir casos a un dataset, se utiliza 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.

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:

```

PROC SORT DATA=FIEBRE.PACTIVO4;
BY NUM_PAC;
RUN;

PROC SORT DATA=FIEBRE.PACTIVO1;
BY NUM_PAC;
RUN;

DATA FIEBRE.PACTIVO5;
SET FIEBRE.PACTIVO4 PACTIVO1;
BY NUM_PAC;
RUN;

PROC PRINT;
RUN;

```

4.9. DAR ETIQUETAS A LAS VARIABLES Y A SUS VALORES.

```
PROC FORMAT;
VALUE VSEXO 1='HOMBRE'
           0='MUJER';
RUN;

DATA FIEBRE.FINAL;
SET FIEBRE.PACTIVO5;
LABEL NUM_PAC='NUMERO PACIENTE';
LABEL TRATAM='TRATAMIENTO ADMINISTRADO';
FORMAT SEXO VSEXO. EDAD 2.;
RUN;
```

SAS ® nos permite definir la forma en qué son listados los valores de las variables. Esta operación se realiza mediante la instrucción **FORMAT** del paso **DATA**. Una de estas " formas " se crea mediante el procedimiento **FORMAT**. Una vez creada, se puede utilizar para cualquier otra variable. Con la instrucción **LABEL**, se dan etiquetas a las variables.

```
PROC PRINT DATA=FIEBRE.FINAL LABEL NOOBS;
VAR NUM_PAC INIC TRATAM SEXO;
LABEL INIC='LABEL TEMPORAL';
RUN;
```

NUMERO PACIENTE	LABEL TEMPORAL	TRATAMIENTO ADMINISTRADO	SEXO
1	JPP	1	HOMBRE
2	JDA	2	HOMBRE
3	ARA	1	MUJER
4	ASB	2	MUJER
5	ABC	1	HOMBRE
6	FRE	2	HOMBRE
7	SAD	1	HOMBRE
8	GAC	2	MUJER
9	PPC	1	MUJER
10	LBB	2	MUJER
11	JVM	1	MUJER
12	ARA	2	MUJER
13	XBG	1	HOMBRE
14	ASD	2	HOMBRE

Ventana **OUTPUT** con el resultado de **PROC PRINT** con las etiquetas.

Cuando 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 (per defecto se listan todas).

Si se incluyen las instrucciones **LABEL** y/o **FORMAT** en **PROC PRINT**, es una acción temporal; utilizando **LABEL** y/o **FORMAT** en un paso **DATA**, se definen las etiquetas de las variables o valores en el correspondiente dataset.

4.10. ASIGNACIONES Y CONDICIONES.

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

```
DATA FINAL_S;
SET FIEBRE.FINAL;
CENTRE=1;
IF EDAD < 50 THEN EDAD2=1;
ELSE EDAD2=2;
IF NUM_PAC=5 THEN DO;
INIC='ABC';
SEXO=1;
END;
RUN;

PROC PRINT;
RUN;
```

Obs	NUM_PAC	TRATAR	INIC	SEXO	EDAD	CENTRE	EDAD2
1	1	1	JPP	HOMBRE	53	1	2
2	2	2	JDA	HOMBRE	47	1	1
3	3	1	RFA	MUJER	81	1	2
4	4	2	HSB	MUJER	29	1	1
5	5	1	ABC	HOMBRE	43	1	1
6	6	2	FRE	HOMBRE	73	1	2
7	7	1	SAD	HOMBRE	22	1	1
8	8	2	GAC	MUJER	53	1	2
9	9	1	PAC	MUJER	35	1	1
10	10	2	LBB	MUJER	35	1	1
11	11	1	JVA	MUJER	44	1	1
12	12	2	RFA	MUJER	15	1	1
13	13	1	XBD	HOMBRE	75	1	2
14	14	2	MSD	HOMBRE	64	1	2

Ventana **OUTPUT** con el resultado del **PROC PRINT** ejecutado (no hay **LABELS**).

En el caso anterior se crea un dataset temporal con las modificaciones. 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**.

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

```

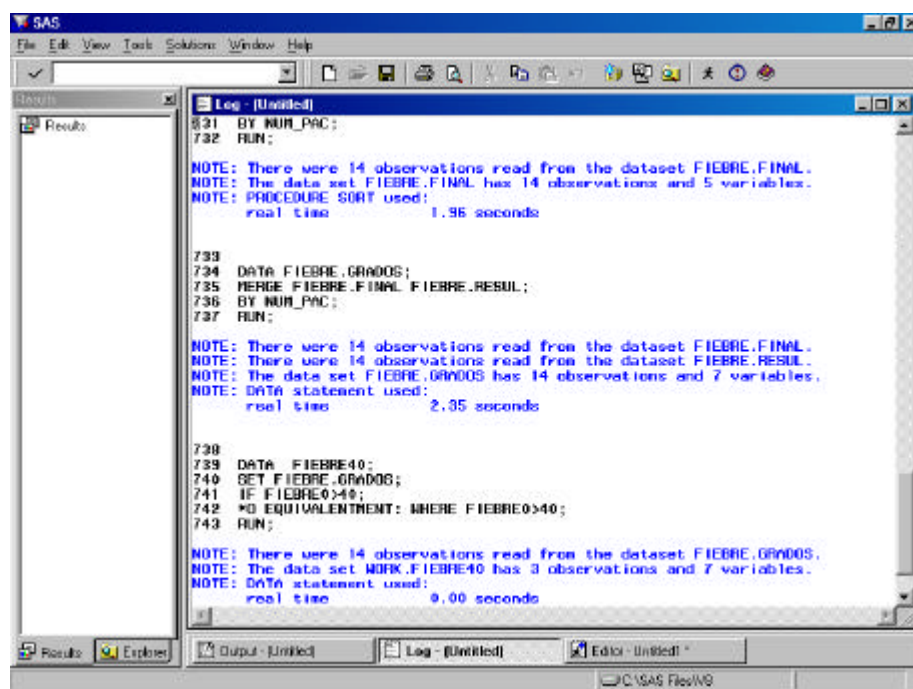
PROC SORT DATA=FIEBRE.RESUL;
BY NUM_PAC;
RUN;

PROC SORT DATA=FIEBRE.FINAL;
BY NUM_PAC;
RUN;

DATA FIEBRE.GRADOS;
MERGE FIEBRE.FINAL FIEBRE.RESUL;
BY NUM_PAC;
RUN;

DATA FIEBRE40;
SET FIEBRE.GRADOS;
IF FIEBRE0>40;
*O EQUIVALENTMENT: WHERE FIEBRE0>40;
RUN;

```



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

4.11. 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: =, <, >, <=, >=, ^=, 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

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	Extraes una subcadena de un argumento
TRANSLATE	Reemplaza caracteres específicos de un argumento

4.12. GENERACION DE VARIABLES ALEATORIAS.

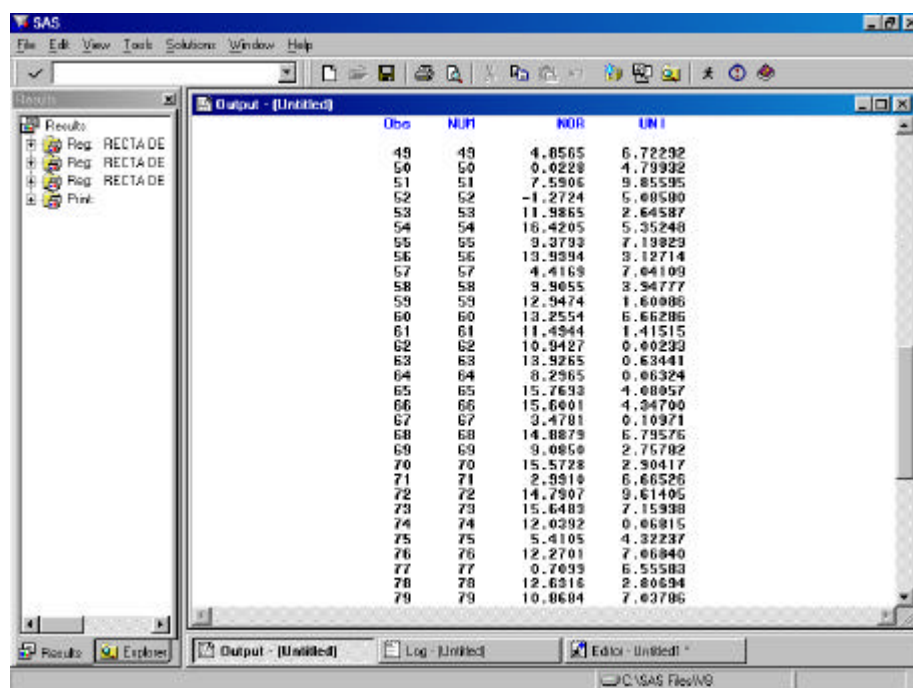
Para generar una variable aleatoria, creándola dentro de un determinado dataset ya existente, existen las funciones aleatorias definidas en el apartado anterior:

```
DATA COMP2 ;
SET FIEBRE.GRADOS ;
NOR=RANNOR ( 0 ) *5+10 ;
UNI=RANUNI ( 0 ) *10 ;
KEEP NOR UNI ;
RUN ;
```

La instrucción **KEEP** permite guardar en el dataset creado sólo las variables allí mencionadas. Las demás variables son eliminadas del dataset resultante.

Para generar un número determinado de valores de una cierta variable aleatoria se crea un dataset nuevo definiendo alguna de las variables de este dataset utilizando las funciones aleatorias mencionadas anteriormente:

```
DATA COMP3;  
DO NUM=1 TO 100;  
NOR=RANNOR(0)*5+10;  
UNI=RANUNI(0)*10;  
OUTPUT;  
END;  
RUN;  
  
PROC PRINT;  
RUN;
```



Obs	NUM	NOR	UNI
49	49	4.8565	6.72292
50	50	0.0228	4.79932
51	51	7.5906	9.85595
52	52	-1.2724	5.08500
53	53	11.9865	2.64587
54	54	16.4205	5.35248
55	55	9.3792	7.19929
56	56	13.9594	9.12714
57	57	4.4169	7.64109
58	58	3.9655	3.94777
59	59	12.9474	1.60086
60	60	13.2554	6.66286
61	61	11.4944	1.41515
62	62	10.9427	0.00233
63	63	13.9265	0.63441
64	64	8.2965	0.66324
65	65	15.7693	4.08957
66	66	15.6001	4.34700
67	67	3.4781	0.10971
68	68	14.8879	6.79576
69	69	9.0850	2.76782
70	70	15.5728	2.30417
71	71	2.9510	6.66526
72	72	14.7907	9.61496
73	73	15.6483	7.15938
74	74	12.0292	0.66815
75	75	5.4105	4.32237
76	76	12.2701	7.66840
77	77	0.7093	6.55583
78	78	12.6316	2.86694
79	79	10.8684	7.63786

Ventana **OUTPUT** con parte del resultado de **PROC PRINT** del dataset COMP3.

A partir del bucle **DO - END**, cada vez que se ejecuta la instrucción **OUTPUT**, los valores de las variables **NOR**, **UNI** y **NUM** son añadidas al dataset. Así pues, el número de iteraciones del bucle da el número total de observaciones del dataset final.

4.13. LOS PROCEDIMIENTOS

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:

```
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 para 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** y **OUTPUT** no sirven para los procedimientos no estadísticos.

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

PROCEDIMIENTOS DE SAS/BASE

APPEND Procedure	Añadir datos a un dataset
CALENDAR Procedure	Calendario con fechas y citas
CATALOG Procedure	Manipula los catálogos de SAS
CHART Procedure	Gráficos de barras sencillos
CIMPORT Procedure	Importación de datos de otras versiones de SAS
COMPARE Procedure	Comparación de Bases de datos
CONTENTS Procedure	Contenidos de un dataset
COPY Procedure	Realiza copias de un dataset
CORR Procedure	Correlación entre variables
CPORT Procedure	Exportación de datos de otras versiones de SAS
DATASETS Procedure	Manipulación de datasets (eliminar).
DBCSTAB Procedure	Produce tablas de conversión a caracteres de doble-byte.
DISPLAY Procedure	Visualizar títulos
EXPLODE Procedure	Títulos grandes, vía explotar caracteres.
EXPORT Procedure	Exportar datasets a texto (p. ej.)
FORMAT Procedure	Dar etiquetas a los valores de las variables
FORMS Procedure	Para crear etiquetas adhesivas
FREQ Procedure	Tablas de frecuencias
FSLIST Procedure	Examinar ficheros externos al SAS
IMPORT Procedure	Importar datos (p. ej. en formato texto)
MEANS Procedure	Resumir los datos
OPTIONS Procedure	Opciones de la ventana OUTPUT
PLOT Procedure	Diagramas de dispersión sencillos
PMENU Procedure	Prepara Menus para ser utilizados por otros módulos de SAS (AF, FSP)
PRINT Procedure	Listar datasets
PRINTTO Procedure	Defini las rutas dónde almacenar las ventanas LOG y OUTPUT
RANK Procedure	Crea Rangos a partir de variables.
REGISTRY Procedure	Mantiene el registro de SAS.
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
TIMEPLOT Procedure	Diagramas de dispersión respecto el tiempo
TRANPOSE Procedure	Transpone datasets
TRANTAB Procedure	Produce y mantiene tablas de traducción de códigos
UNIVARIATE Procedure	Análisis univariados

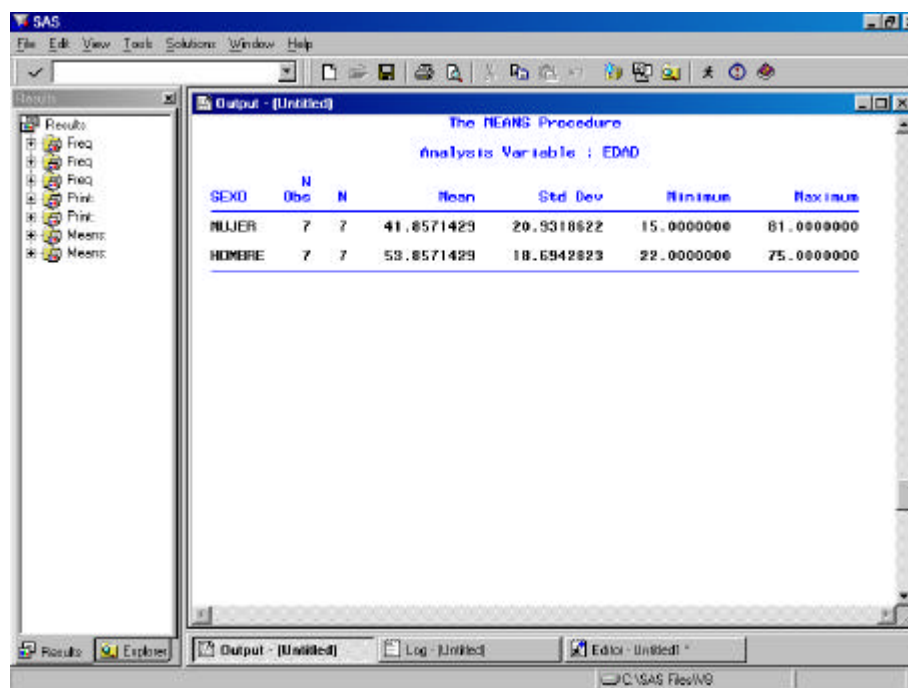
5. SAS/STAT I: Presentación de Resultados.

5.1. RESUMIR LOS DATOS.

Una vez leída la matriz de datos es usual observar algunos de los estadísticos más sencillos y tener así una idea aproximada de cómo 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**.

```
PROC MEANS DATA=FIEBRE.GRADOS ;
VAR EDAD ;
CLASS SEXO ;
RUN ;
```

```
PROC MEANS DATA=FIEBRE.GRADOS ;
VAR FIEBRE0 FIEBRE1 ;
OUTPUT OUT=M_FIEBRE MEAN(FIEBRE0 FIEBRE1)= M0 M1 STD(FIEBRE0
FIEBRE1)=STD0 STD1 ;
RUN ;
```



SEXO	Obs	N	Mean	Std Dev	Minimum	Maximum
MUJER	7	7	41.8571429	20.9318622	15.0000000	81.0000000
HOMBRE	7	7	53.8571429	18.6942823	22.0000000	75.0000000

Ventana **OUTPUT** 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 qué 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. 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	Skewness.
KURTOSIS	Kurtosis.
T	t-Student para la hipótesis nula que la media de la variable es igual a cero.
PRT	Probabilidad para un t-valor superior.

5.2. PROC TABULATE: TABLAS

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

```
PROC TABULATE DATA=FIEBRE.GRADOS;
CLASS TRATAM SEXO;
TABLE TRATAM, (N*SEXO ALL);
RUN;
```

```
PROC TABULATE DATA=FIEBRE.GRADOS;
CLASS TRATAM;
VAR FIEBRE0 FIEBRE1;
TABLE TRATAM, (MEAN*FIEBRE0 MEAN*FIEBRE1);
RUN;
```

	N		All
	SEXO		
		MUJER	HOMBRE
TRATAMIENTO ADMINISTRADO			
1	3.00	4.00	7.00
2	4.00	3.00	7.00

Ventana **OUTPUT** 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 el resto de variables a utilizar.

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.

6. SAS/STAT II: Análisis Univariante

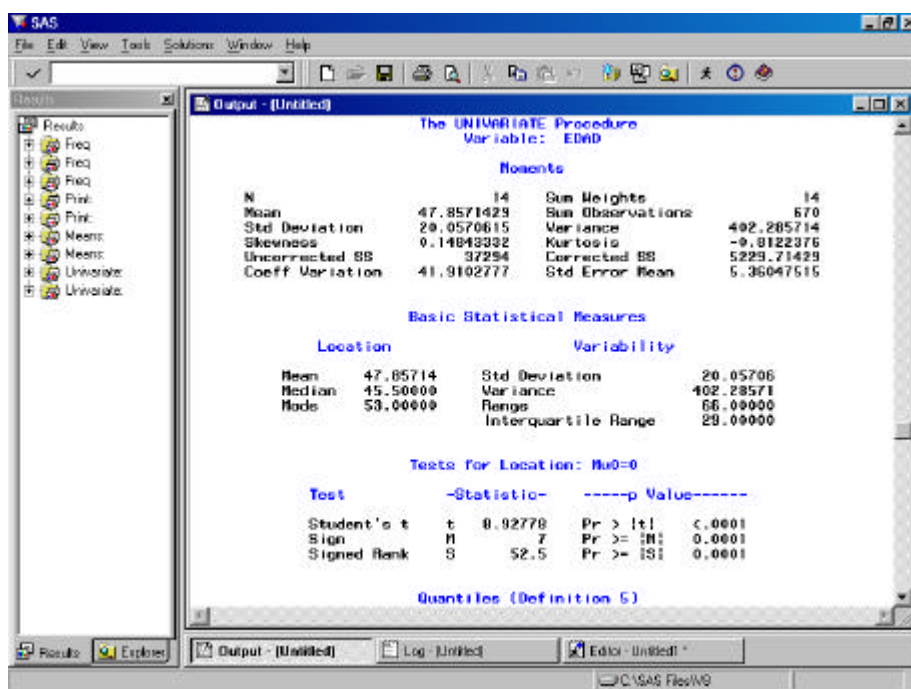
6.1. PROC UNIVARIATE

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**.

```
PROC UNIVARIATE DATA=FIEBRE.GRADOS ;  
VAR EDAD ;  
OUTPUT OUT=U_EDAD MEAN=MITJANA MEDIAN=MEDIANA MODE=MODA ;  
RUN ;
```

```
PROC UNIVARIATE DATA=FIEBRE.GRADOS PLOT NORMAL ;  
VAR EDAD ;  
RUN ;
```

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



The UNIVARIATE Procedure
Variable: EDAD

Moments

	N	Sum Weights
Mean	47.8571429	14
Std Deviation	20.0570615	Sum Observations
Skewness	0.1484332	Variance
Uncorrected SS	37254	Kurtosis
Coeff Variation	41.9102777	Corrected SS
		Std Error Mean

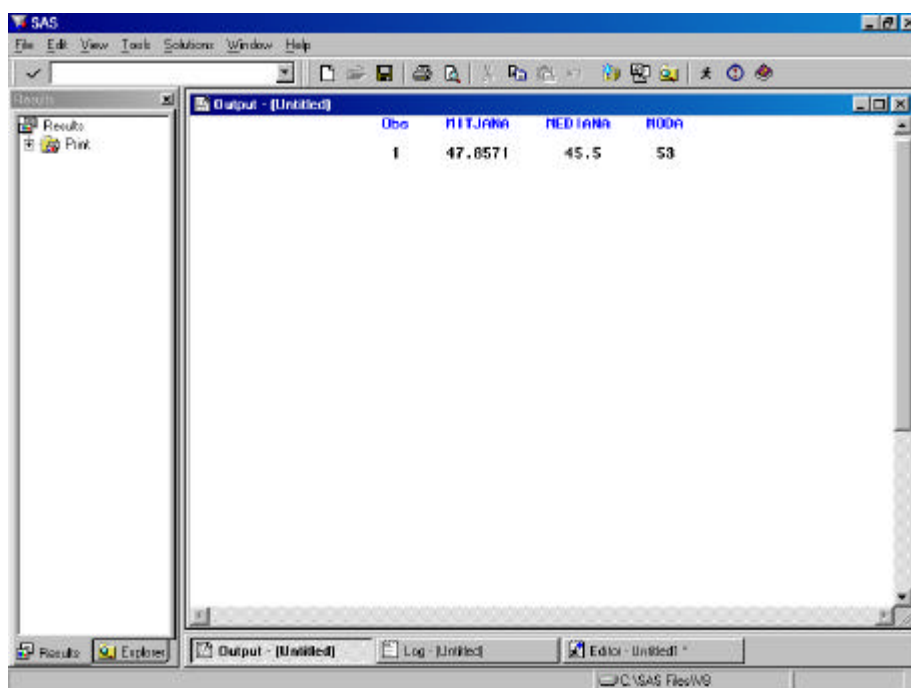
Basic Statistical Measures

Location		Variability	
Mean	47.85714	Std Deviation	20.05706
Median	45.50000	Variance	402.28571
Mode	53.00000	Range	66.00000
		Interquartile Range	29.00000

Tests for Location: Mu0=0

Test	-Statistic-	-----p Value-----
Student's t	t 0.92778	Pr > t < .0001
Sign	M 7	Pr >= M 0.0001
Signed Rank	S 52.5	Pr >= S 0.0001

Quantiles (Definition 5)

Ventana **OUTPUT** con parte del resultado de **PROC UNIVARIATE**.


Obs	MEAN	MEDIAN	MODA
1	47.8571	45.5	53

Ventana **OUTPUT** con el resultado de **PROC PRINT** del dataset de salida U_EDAD.

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 puede obtener un histograma mediante la instrucción:

```
PROC UNIVARIATE DATA=FIEBRE.GRADOS ;  
VAR EDAD ;  
HISTOGRAM ;  
RUN ;
```

6.2. TABLAS DE FRECUENCIAS: 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).

```
PROC FREQ DATA=FIEBRE.GRADOS ;  
TABLES SEXO /OUT=F_SEXO ;  
RUN ;  
  
PROC FREQ DATA=FIEBRE.GRADOS ;  
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.

7. SAS/STAT III: Análisis Bivariante

7.1. TABLAS DE CONTINGENCIA: PROC FREQ

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

```
PROC FREQ DATA=FIEBRE.GRADOS;
TABLES TRATAM*SEXO /CHISQ;
OUTPUT OUT=X_TRATAM CHISQ;
RUN;
```

```
PROC FREQ DATA=FIEBRE.FINAL;
TABLES TRATAM*SEXO /CHISQ OUT=F_TRATAM;
RUN;
```

```
PROC PRINT DATA=F_TRATAM;
RUN;
```

```
PROC PRINT DATA=X_TRATAM;
RUN;
```

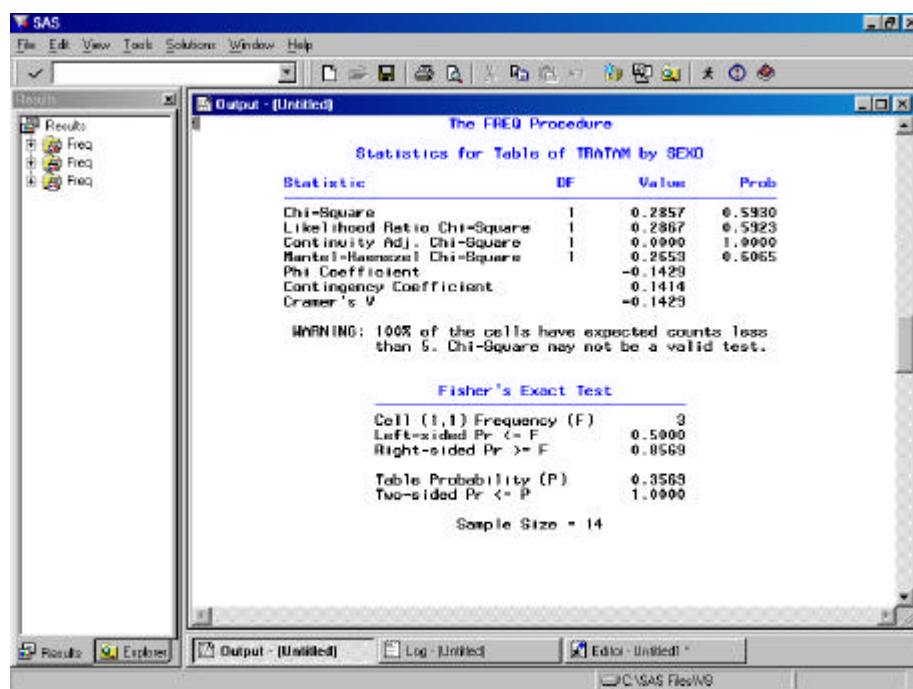
Tal y como se ha comentado en el apartado anterior, escribiendo la opción **OUTPUT OUT=dataset_salida** se obtienen los estadísticos calculados (siempre y cuando hayan sido requeridos y especificados a continuación).

The FREQ Procedure
Table of TRATAM by SEXO

TRATAM(TRATAMIENTO ADMINISTRADO)

		SEXO		
		MUJER	HOMBRE	Total
Frequency				
Percent				
Row Pct				
Col Pct				
1		3	4	7
		21.43	28.57	50.00
		42.86	57.14	
		42.86	57.14	
2		4	3	7
		28.57	21.43	50.00
		57.14	42.86	
		57.14	42.86	
Total		7	7	14
		50.00	50.00	100.00

Ventana **OUTPUT** con la tabla de contingencia TRATAM vs SEXO



The FREQ Procedure

Statistics for Table of TRATAM by SEXO

Statistic	DF	Value	Prob
Chi-Square	1	0.2857	0.5930
Likelihood Ratio Chi-Square	1	0.2867	0.5923
Continuity Adj. Chi-Square	1	0.0900	1.0000
Natural Inhomog. Chi-Square	1	0.2553	0.6065
Phi Coefficient		-0.1429	
Contingency Coefficient		0.1414	
Cramer's V		-0.1429	

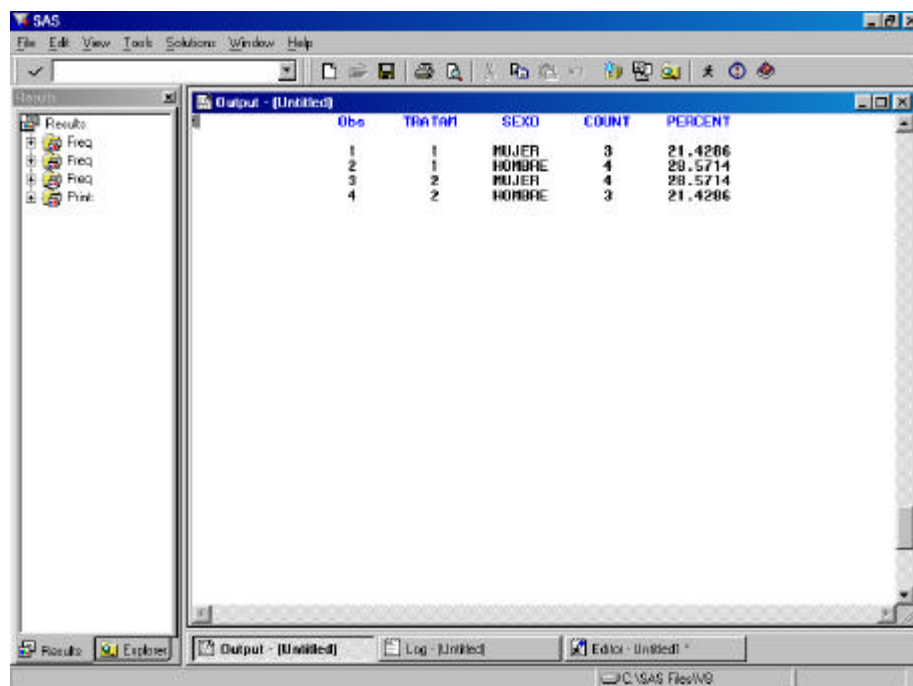
WARNING: 100% of the cells have expected counts less than 5. Chi-Square may not be a valid test.

Fisher's Exact Test

Cell (1,1) Frequency (F)	3
Left-sided Pr <= F	0.5000
Right-sided Pr >= F	0.0569
Table Probability (P)	0.3569
Two-sided Pr <= P	1.0000

Sample Size = 14

Ventana **OUTPUT** con los estadísticos correspondientes a / **CHISQ** de la tabla TRATAM vs SEXO.



Obs	TRATAM	SEXO	COUNT	PERCENT
1	1	MUJER	3	21.4286
2	1	HOMBRE	4	28.5714
3	2	MUJER	4	28.5714
4	2	HOMBRE	3	21.4286

Ventana **OUTPUT** con el resultado del **PROC PRINT** del dataset de salida F_TRATAM.

D	F	P	L	R	C	H	I
I	14	0.28571	0.59298	0.28669	0.59235	0	1
0.26531							

D	F	P	X	P	L	X	P	R	X	P	2	C	N	T	G	Y	C	R	A	N	V
I	1	0.60650	0.5	0.85693	1	-0.14286	0.14142	-0.14286													

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

En la instrucción **TABLES** es dónde 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**.

7.2. COMPARACIÓN DE MEDIAS.

Uno de los procedimientos que se emplean para comparar medias cuando hay dos únicos grupos es el procedimiento **TTEST**. Este procedimiento realiza el t-Test bajo la hipótesis nula que las medias de los dos grupos son iguales.

```
DATA COMP;
SET FIEBRE.GRADOS;
FEB=FIEBRE1-FIEBRE0;
RUN;

PROC TTEST DATA=COMP;
CLASS TRATAM;
VAR FIEBRE0;
RUN;

PROC TTEST DATA=COMP;
CLASS TRATAM;
VAR FEB;
RUN;
```

The screenshot shows the SAS Output window with the following data:

Variable	Class	N	Mean	Mean	Mean	Std Dev	Std Dev
FEB	1	7	-6.743	6.2857	1.3148	0.717	1.1127
FEB	2	7	-4.387	-3.143	-1.899	0.8668	1.3452
FEB	Diff (1-2)		1.9909	3.4286	4.8662	0.8852	1.2944

Variable	Class	Upper CL	Std Dev	Std Err	Minimum	Maximum
FEB	1	2.4502	0.4206		-1	2
FEB	2	2.9622	0.5004		-5	-1
FEB	Diff (1-2)	2.0377	0.6598			

Variable	Method	Variances	DF	t Value	Pr > t
FEB	Pooled	Equal	12	5.20	0.0002
FEB	Satterthwaite	Unequal	11.6	5.20	0.0002

Variable	Method	Nus DF	Den DF	F Value	Pr > F
FEB	Folded F	6	6	1.46	0.6566

Ventana **OUTPUT** con el resultado de **PROC TTEST**.

La variable especificada en la instrucción **CLASS** es la variable que clasifica los datos en dos grupos (sólo puede tener dos categorías), mientras que la variable que se especifica en **VAR** es la variable a estudiar.

Después de ejecutar el procedimiento **TTEST**, aparecen en la ventana **OUTPUT** algunos estadísticos correspondientes a cada una de las dos clases: el número de observaciones, la media, la desviación estándar, error estándar, el valor mínimo, el valor máximo, el resultado del test (tanto bajo la suposición de varianzas iguales cómo bajo la suposición que las varianzas son diferentes) y finalmente el resultado para un test de homogeneidad de varianzas (bajo la hipótesis nula que las varianzas son iguales).

Nota: En este procedimiento no existe ninguna opción para guardar los estadísticos en un dataset de salida.

7.3. RELACIONES ENTRE VARIABLES CONTINUAS.

7.3.1. CORRELACIÓN.

Para estudiar la relación entre dos o diversas variables, es práctico comenzar por analizar la correlación entre éstas utilizando el procedimiento **CORR**.

Este procedimiento proporciona algunos estadísticos básicos de cada una de las variables estudiadas: el número de observaciones, la media, la desviación estándar, la suma de todos los valores, el valor mínimo y el valor máximo.

Por otra parte, también se muestra una tabla con la correlación entre las variables especificadas en la instrucción **VAR**. Esta tabla contiene la correlación de Pearson (por defecto éstos son los estadísticos calculados) y la significación asociada a la hipótesis nula que el coeficiente de correlación es cero para cada posible combinación de las variables especificadas.

```
PROC CORR DATA=FIEBRE.GRADOS ;  
VAR FIEBRE0 FIEBRE1 ;  
WITH EDAD ;  
RUN ;
```

Con la instrucción **WITH**, sólo se calcula la correlación de las variables especificadas en **VAR** con las variables especificadas en **WITH**.

Con este procedimiento también existe la opción de calcular estadísticos no paramétricos para los coeficientes de correlación, utilizando **SPEARMAN** después de la palabra **CORR** (también hay otros estadísticos no paramétricos disponibles).

```
PROC CORR DATA=FIEBRE.GRADOS OUTF=C_FIEBRE ;  
VAR FIEBRE0 FIEBRE1 ;  
RUN ;  
  
PROC PRINT DATA=C_FIEBRE ;  
RUN ;  
  
PROC CORR DATA=FIEBRE.GRADOS COV OUTF=C_FIEBRE2 ;  
VAR FIEBRE0 FIEBRE1 ;  
RUN ;  
  
PROC PRINT DATA=C_FIEBRE2 ;  
RUN ;
```

Covariance Matrix, DF = 13

	FIEBRE0	FIEBRE1
FIEBRE0	1.807692308	0.500000000
FIEBRE1	0.500000000	3.763736264

Simple Statistics

Variable	N	Mean	Std Dev	Sum
FIEBRE0	14	39.50000	1.34450	553.00000
FIEBRE1	14	38.07143	1.94004	533.00000

Simple Statistics

Variable	Minimum	Maximum
FIEBRE0	37.00000	42.00000
FIEBRE1	36.00000	42.00000

**Pearson Correlation Coefficients, N = 14
Prob > |r| under H0: Rho=0**

	FIEBRE0	FIEBRE1
FIEBRE0	1.00000	0.19169
FIEBRE1	0.19169	1.00000

Ventana **OUTPUT** con parte del resultado de **PROC CORR** con la opción **COV**.

Obs	_TYPE_	_NAME_	FIEBRE0	FIEBRE1
1	COV	FIEBRE0	1.8077	0.5000
2	COV	FIEBRE1	0.5000	3.7637
3	MEAN		39.5000	38.0714
4	STD		1.3445	1.9400
5	N		14.0000	14.0000
6	CORR	FIEBRE0	1.0000	0.1917
7	CORR	FIEBRE1	0.1917	1.0000

Ventana **OUTPUT** con el resultado de **PROC PRINT** del dataset de salida C_FIEBRE2 (obtenido a partir de **PROC CORR** con la opción **COV**).

La única forma de guardar los estadísticos calculados en un dataset nuevo, es especificar **OUTP=dataset_salida** o bien **OUTS=dataset_salida** después de la palabra **CORR**, según se deseen los coeficientes de correlación de Pearson o los de Spearman.

Finalmente, en el caso que interese obtener un listado de la matriz de covarianzas, se utiliza la opción **COV**, escribiendo esta palabra después de **CORR** en la primera instrucción. Si la opción **OUTP=dataset_salida** es especificada, entonces el dataset de salida también contiene la matriz de covarianzas.

7.3.2. REGRESIÓN SIMPLE.

Para estudiar la recta de regresión entre dos variables, es útil el procedimiento **REG**. El procedimiento **REG** ajusta modelos de regresión lineal por el método de los mínimos cuadrados. Aunque sólo se comenta el **PROC REG**, para el análisis de modelos lineales simples, SAS ® dispone de diversos procedimientos que pueden realizar análisis de modelos de todo tipo.

El procedimiento REG requiere la instrucción MODEL, en la cual se especifica el modelo y las variables que se van a utilizar, escribiendo la variable dependiente seguida de un signo igual y la variable independiente.

```
LIBNAME PESO 'A:\';

PROC PRINT DATA=PESO.FRUTA;
TITLE1 ' RECTA DE REGRESION ENTRE LAS VARIABLES: ';
TITLE2 ' PESO DE LAS MANZANAS y ZUMO EXPRIMIDO';
RUN;

PROC REG DATA=PESO.FRUTA;
MODEL ZUMO=PESO;
RUN;
```

RECTA DE REGRESION ENTRE LAS VARIABLES:
PESO DE LAS MANZANAS y ZUMO EXPRIMIDO

The REG Procedure
Model: MODEL1
Dependent Variable: ZUMO

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	0.04286	0.04286	13.08	0.0035
Error	12	0.03931	0.00328		
Corrected Total	13	0.08217			

Root MSE 0.05724
Dependent Mean 0.25857
Coeff Var 22.13615

R-Square 0.5216
Adj R-Sq 0.4817

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	-0.02404	0.07962	-0.30	0.7679
PESO	1	0.56201	0.15533	3.62	0.0035

Ventana **OUTPUT** con el resultado de **PROC REG**.

La salida en la ventana **OUTPUT** del procedimiento **REG** tiene dos partes.

La primera es la sección de análisis de la varianza, y muestra información del ajuste de los datos al modelo:

SOURCE	Fuente de variación.
DF	Grados de libertad asociados a la fuente.
Sum of Squares	Suma de cuadrados de cada término.
Mean Square	Suma de cuadrados dividido por los grados de libertad.
F value	Valor F para testar la hipótesis nula que todos los parámetros son cero excepto INTERCEPT (la constante).
Prob>F	Significación del test.
Root MSE	Raíz de error medio cuadrático.
Dep Mean	Media de la variable dependiente.
C.V.	Coefficiente de variación.
R-Square	Valor de R^2 .
Adj R-sq	Valor de R^2 ajustado a los grados de libertad.

La segunda sección está dedicada a los estimadores de los parámetros. La información que proporciona es la siguiente:

Variable	Variable regresora y término constante (INTERCEPT).
DF	Grados de libertad para la variable.
Parameter estimate	Estimación del parámetro.
Standard Error	Error estándar
T for H0	t Test para la hipótesis nula que el estimador del parámetro es igual a cero.
Prob > T	Significación de dos colas del test.

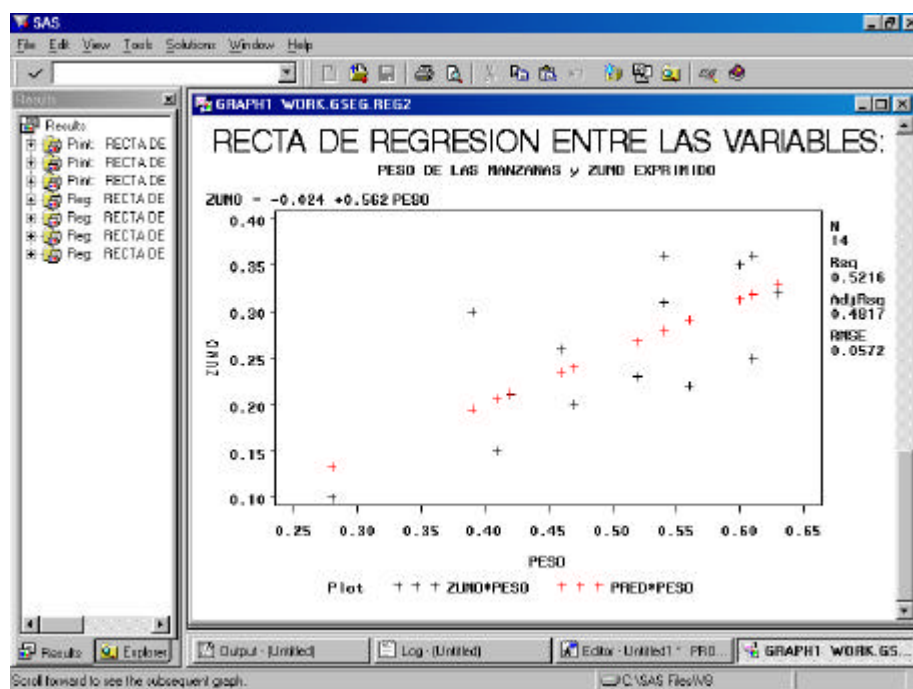
El procedimiento **REG** también permite obtener representaciones de los datos en una gráfica con la instrucción **PLOT**.

```
PROC REG DATA=PESO.FRUTA;
MODEL ZUMO=PESO;
PLOT ZUMO * PESO;
RUN;
```

Para solicitar una gráfica se escribe las dos variables intercaladas por el símbolo ' * ', teniendo en cuenta que la variable de la izquierda es la variable del eje vertical y que la de la derecha es la variable del eje horizontal.

También es posible escoger el carácter con el que se representan los datos escribiendo un símbolo ' = ' seguido del símbolo deseado entre comillas. Si no se especifica ningún símbolo, aparecen los números correspondientes a la frecuencia con que es repite aquel punto de la gráfica.

```
PROC REG DATA=PESO.FRUTA;
MODEL ZUMO=PESO;
PLOT ZUMO * PESO P. * PESO = ' * ' / OVERLAY;
RUN;
```



Ventana de gráficos con el resultado del gráfico creado por **PROC REG**.

Es posible representar diversas gráficas al mismo tiempo, poniéndolas en la instrucción **PLOT** separadas por espacios. También es posible visualizar estadísticos si son escritos seguidos de un punto (cómo si se trataran de otras variables).

Algunos de los estadísticos representables son:

P	Valores predichos.
R	Residuos.
STUDENT	Residuos estudentizados.
U95	Cota superior para el intervalo de confianza del 95% para predicciones Individuales.
L95	Cota inferior para el intervalo de confianza del 95% para predicciones Individuales.

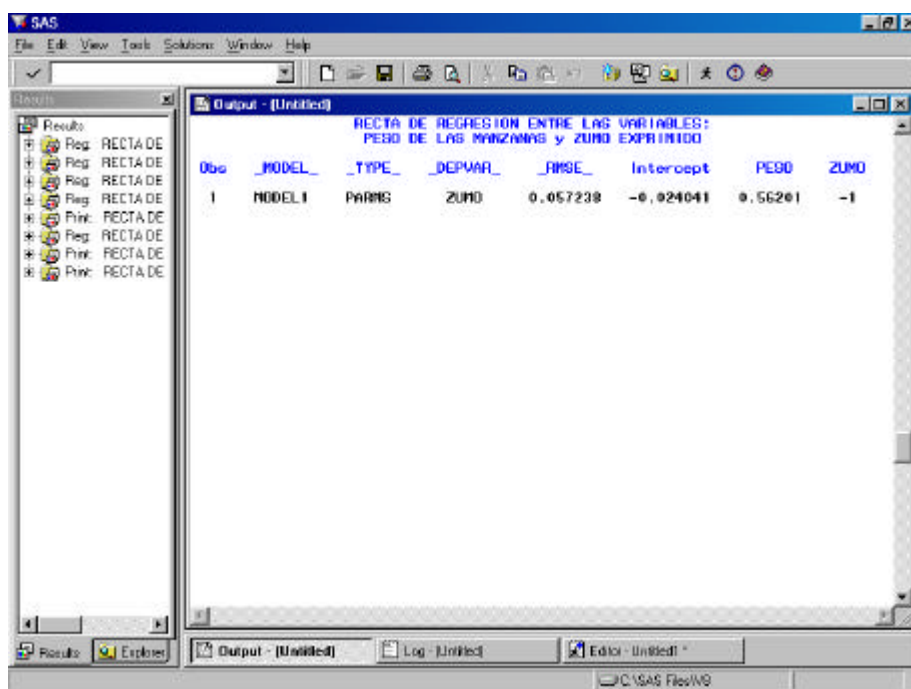
La opción '**/ OVERLAY**' se emplea para visualizar diversas gráficas superpuestas. Cuando se sobreponen los valores de las dos gráficas, aparece el símbolo '? '.

```
PROC REG DATA=PESO.FRUTA OUTEST=T_PESO;
MODEL ZUMO=PESO;
OUTPUT OUT=O_PESO P=PREDI;
RUN;

PROC PRINT DATA=T_PESO;
RUN;

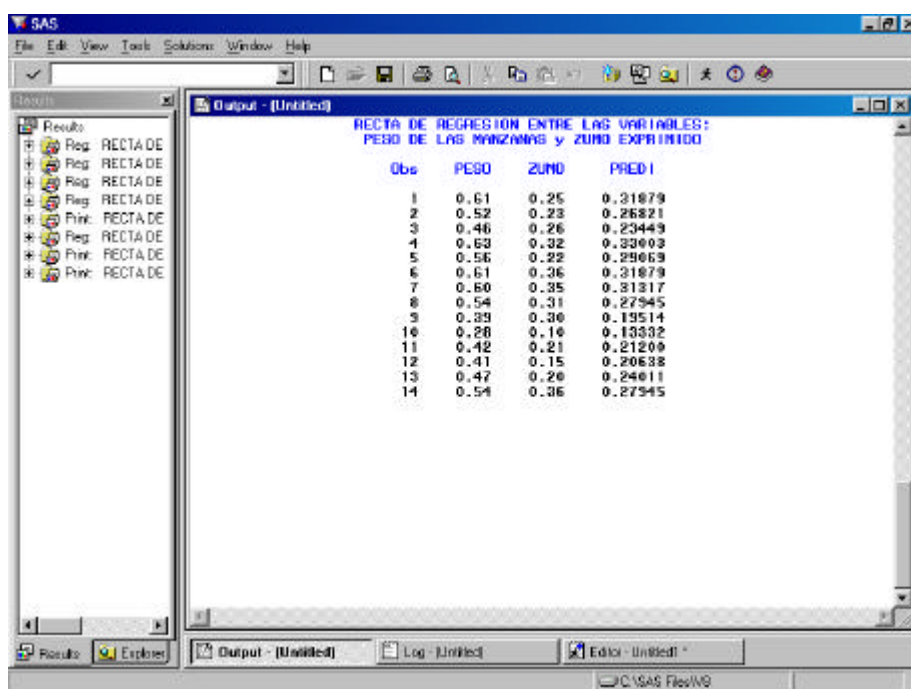
PROC PRINT DATA=O_PESO;
RUN;

TITLE;
```



Obs	_MODEL_	_TYPE_	_DEPVAR_	_RMSE_	Intercept	PESO	ZUMO
1	MODEL1	PARMS	ZUMO	0.057238	-0.024041	0.56261	-1

Ventana **OUTPUT** con el resultado de **PROC PRINT** del dataset de salida T_PESO (con los estadísticos de **PROC REG**).



Obs	PESO	ZUMO	PRED1
1	0.61	0.25	0.31879
2	0.52	0.23	0.26821
3	0.46	0.26	0.23449
4	0.63	0.32	0.33003
5	0.56	0.22	0.29053
6	0.61	0.36	0.31879
7	0.60	0.35	0.31317
8	0.54	0.31	0.27945
9	0.39	0.30	0.19514
10	0.26	0.19	0.10332
11	0.42	0.21	0.21209
12	0.41	0.15	0.20638
13	0.47	0.20	0.24611
14	0.54	0.36	0.27945

Ventana **OUTPUT** con el resultado de **PROC PRINT** del dataset de salida O_PESO (con los valores predichos por **PROC REG**).

Si es necesario guardar los estimadores de los parámetros en un dataset, se utiliza **OUTEST=dataset_salida** después de la palabra **REG** en la primera instrucción del procedimiento. Para guardar en un dataset los estadísticos mencionados anteriormente (**P**, **R**, **Student**, **U95** o **L95**) es útil la instrucción **OUTPUT OUT=dataset_salida**, especificando a continuación los nombres deseados para las variables con dichos estadísticos (este nuevo dataset también contiene los valores de las variables del modelo utilizado).

8. PROCEDIMIENTOS GRÁFICOS (SAS/GRAPH)

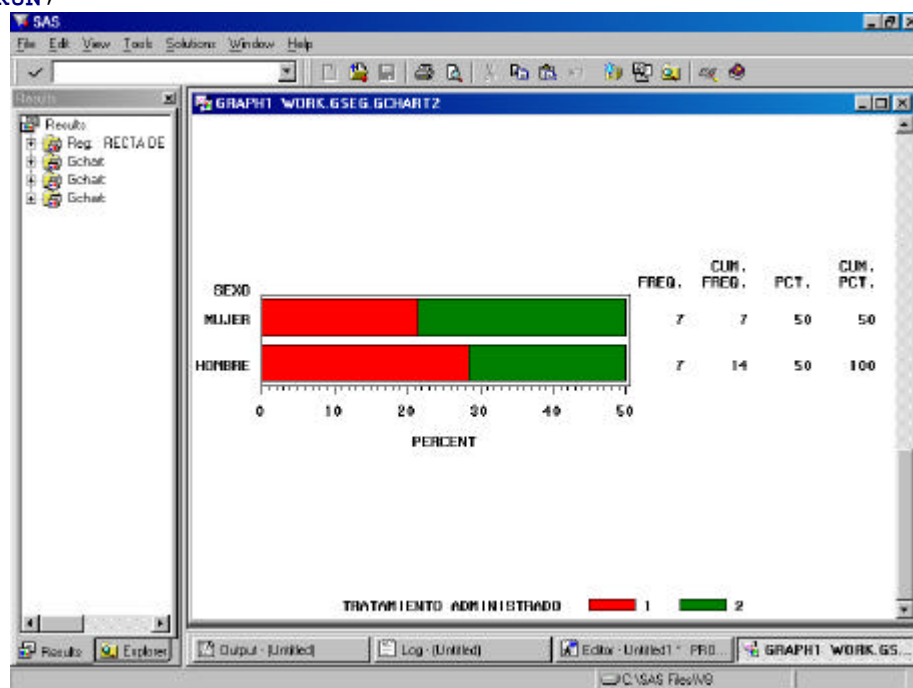
8.1. GRÁFICAS DE BARRAS Y PASTELES.

El procedimiento empleado para crear gráficas de barras verticales, horizontales o gráficos de tipo pastel es el procedimiento **GCHART**. Este procedimiento forma parte del módulo **GRAPH**, un módulo muy completo que se gestiona a parte del resto de módulos (por este motivo no todos los usuarios de SAS ® disponen de él). Éstas son algunas de las posibilidades de este procedimiento:

```
PROC GCHART DATA=FIEBRE.GRADOS ;
VBAR SEXO / DISCRETE GROUP=TRATAM;
RUN;
```

Esta primera gráfica de barras es vertical ya que utiliza la instrucción **VBAR** y representa las frecuencias de la variable **SEXO**. Especificando **DISCRETE**, el procedimiento reconoce que la variable a mostrar es discreta y respeta los valores de la variable (en caso contrario el resultado es un histograma). Especificando la opción **GROUP=nombre_variable** se muestra un diagrama de comparación respecto la variable que se indica.

```
PROC GCHART DATA=FIEBRE.GRADOS ;
HBAR SEXO / DISCRETE SUBGROUP=TRATAM TYPE=PERCENT ;
RUN;
```



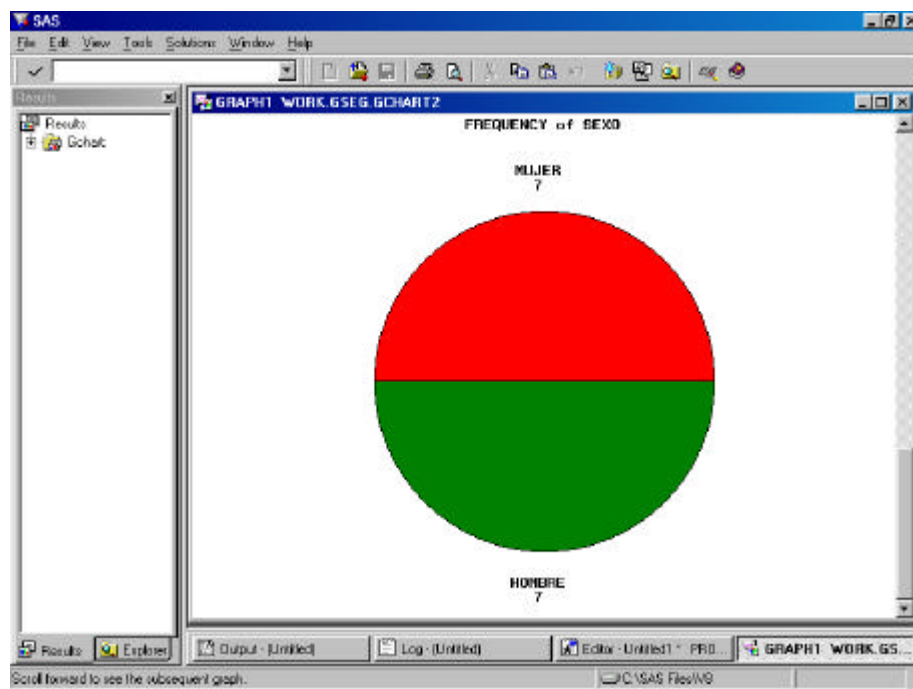
Ventana de gráficos con el resultado del gráfico creado por **PROC GCHART**.

La segunda gráfica de barras representa la variable **SEXO** pero de forma horizontal a causa de la instrucción **HBAR**. La opción **SUBGROUP=nombre_variable** realiza un diagrama apilado respecto la variable que se indica. La opción **TYPE=PERCENT** sirve para representar los porcentajes y no las frecuencias que son representadas por defecto. Las posibilidades de la opción **TYPE=** son **FREQ**, **PERCENT**, **SUM** y **MEAN** (teniendo que especificar **SUMVAR=nombre_variable** en las dos últimas posibilidades).


```
PROC GCHART DATA=FIEBRE.GRADOS ;  
VBAR TRATAM /DISCRETE SUMVAR=FIEBRE1 TYPE=MEAN ;  
RUN ;
```

El código anterior representa en forma vertical los valores medios de la variable FIEBRE1 para los diferentes tratamientos.

```
PROC GCHART DATA=FIEBRE.GRADOS ;  
PIE SEXO / DISCRETE ;  
RUN ;
```



Ventana de gráficos con el resultado del gráfico creado por PROC GCHART.

El código anterior representa en forma de pastel la variable SEXO.

8.2. GRÁFICAS PARA DOS VARIABLES.

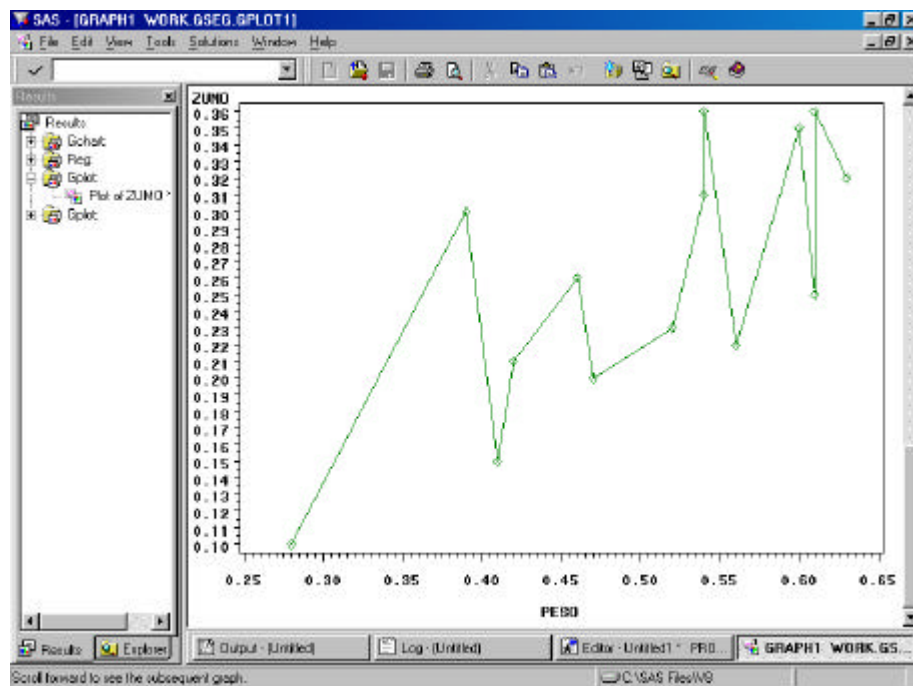
El procedimiento a utilizar para realizar gráficas de dos variables es el procedimiento **GPLOT**, que también forma parte del módulo **GRAPH** del SAS ®. Este procedimiento permite visualizar una variable respecto otra variable punto a punto. Las coordenadas de cada punto de la representación corresponden a los valores de ambas variables en una o más observaciones de los datos.

En la instrucción **PLOT** se menciona qué variables aparecen en la gráfica intercalándolas por un asterisco. Se pueden representar diversas gráficas al mismo tiempo separando las combinaciones por espacios.

Para cambiar la representación de los puntos que aparece por se emplea la instrucción **SYMBOL1** o **SYMBOLn**. Dado que es posible representar varias gráficas al mismo tiempo, cuando 'n' es 1, se refiere a la primera gráfica, cuando 'n' es 2, la segunda, y así sucesivamente.

Especificando a la instrucción **SYMBOL**, **INTERPOL=JOIN**, se unen los puntos representados mediante rectas, mientras que especificando **VALUE=DIAMOND** o **VALUE=STAR**, los puntos aparecen representados con un pequeño diamante o bien un asterisco.

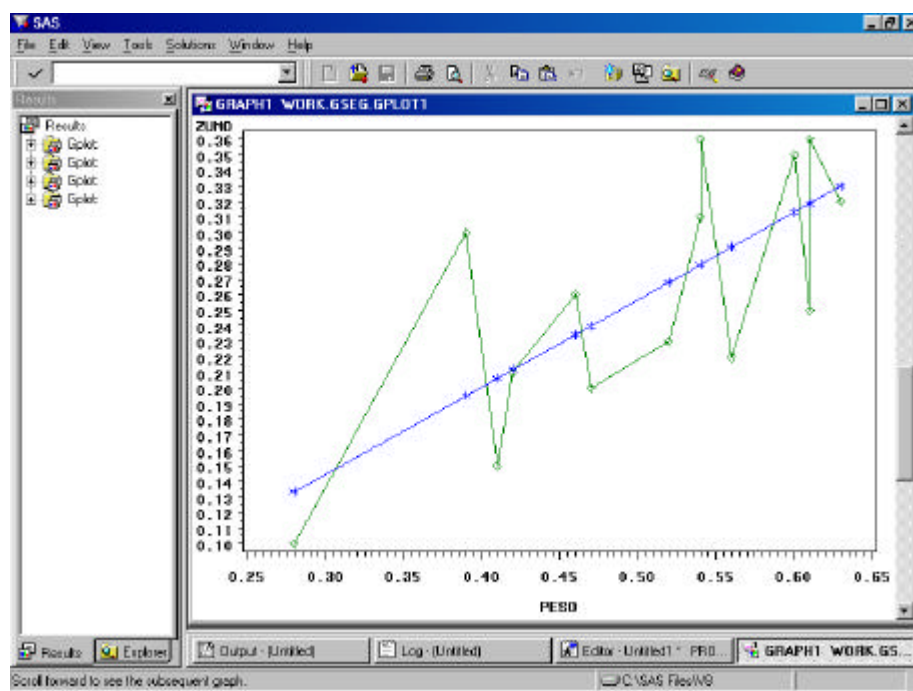
```
PROC REG DATA=PESO.FRUTA;  
MODEL ZUMO=PESO;  
OUTPUT OUT=O_PESO P=PREDI;  
RUN;  
  
PROC SORT DATA=O_PESO;  
BY PESO;  
RUN;  
  
PROC Gplot DATA=O_PESO;  
PLOT ZUMO*PESO;  
SYMBOL1 INTERPOL=JOIN VALUE=DIAMOND COLOR=GREEN;  
RUN;
```



Ventana de gráficos con el resultado del gráfico creado por **PROC Gplot**.

Para representar diversas gráficas superpuestas, se indica mediante la opción **/OVERLAY**.

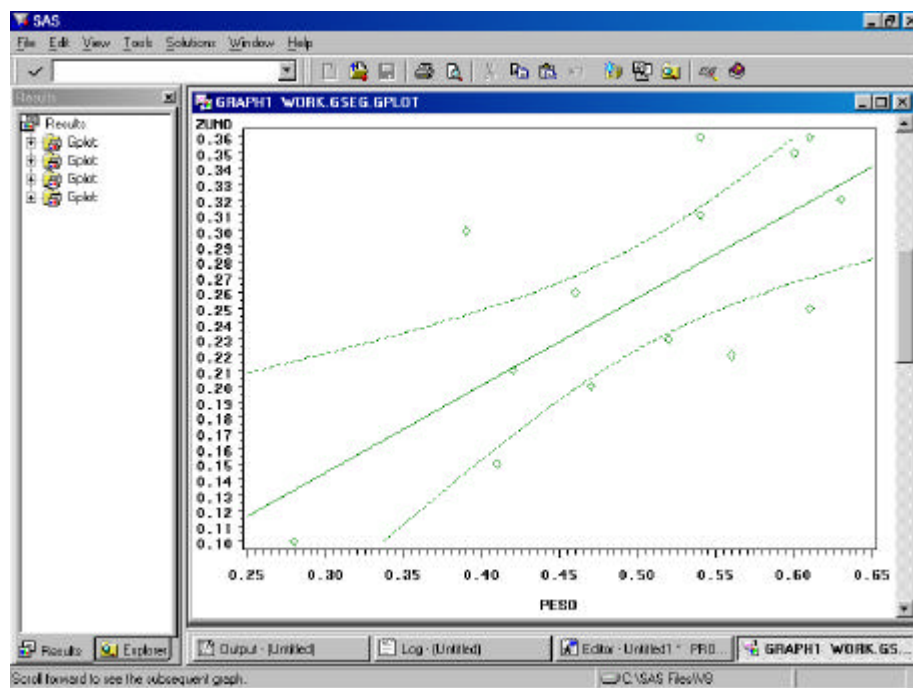
```
PROC Gplot DATA=O_PESO;  
PLOT ZUMO*PESO PREDI*PESO /OVERLAY;  
SYMBOL1 INTERPOL=JOIN VALUE=DIAMOND COLOR=GREEN;  
SYMBOL2 INTERPOL=JOIN VALUE=STAR COLOR=BLUE;  
RUN;
```



Ventana de gráficos con el resultado del gráfico creado por **PROC GLOT**.

Para que aparezca la recta de regresión y el intervalo de confianza del 95 %, se debe utilizar en la instrucción **SYMBOL**, la opción **INTERPOL=RLCLM95**.

```
PROC GLOT DATA=O_PESO;  
PLOT ZUMD*PESO;  
SYMBOL1 INTERPOL=RLCLM95;  
RUN;
```



Ventana de gráficos con el resultado del gráfico creado por **PROC GLOT**.

En la última versión de SAS se incorpora el procedimiento BOXPLOT, que permite representar diagramas de caja.

```
PROC BOXPLOT DATA=dataset;  
PLOT var1*var2;  
RUN;
```

Siendo var1 la variable continua y var2 la variable discreta.

9. ANEXO

9.1. IMPORTAR DATOS DE SPSS

El objetivo de este anexo es presentar una de las posibles formas de transferir datos entre los paquetes estadísticos SPSS y MS Excel paquete estadístico SAS ® y viceversa.

Un de los formatos de fichero que pueden leer y guardar es el llamado "**DELIMITADO POR TABULADORES**".

En este sentido, para exportar un fichero de datos entre dos de estos paquetes, en primer lugar es necesario almacenar dicho fichero en este formato. Sin embargo, los diferentes programas tienen también sus particularidades:

- SPSS lee o guarda los ficheros separados por tabuladores sólo en la extensión **".dat"**.
- Al abrir o guardar un fichero, SPSS pregunta si en el primer registro se encuentran los nombres de las variables.
- Al abrir o guardar un fichero, MS Excel no pregunta si en el primer registro hay los nombres, simplemente lee o almacena el fichero entero.
- Aunque utilizando SAS ® existe la posibilidad de leer o exportar datos con los nombres de las variables en la primera línea utilizando el paso **DATA**, los pasos a seguir no son elementales. Por ejemplo, al leer datos externos es preferible declarar los nombres en la instrucción INPUT en lugar de leerlos directamente del fichero de datos.
- La versión v.8 de SAS ® tiene incorporado un desplegable que permite importar o exportar datos en formato de texto, incluido el "separado para tabuladores", con la posibilidad de incluir los nombres de las variables en el primer registro del fichero. El desplegable utiliza unos procedimientos llamados **PROC IMPORT** y **PROC EXPORT**.

También existen otras formas de importar datos al SAS ® menos costosas, pero es imprescindible poseer módulos complementarios especiales para la importación/exportación de datos.

El siguiente código también lee ficheros de SPSS si estos están guardados previamente en la extensión **".por"** correspondiente al formato llamado **portable**:

```
LIBNAME nombre_libreria SPSS 'nombre_fichero.por';  
DATA nombre_dataset_final;  
SET nombre_libreria._FIRST_;  
RUN;
```

9.2. DATOS DEL MANUAL.

A lo largo del manual se hace referencia a ficheros de datos y a su ubicación. Para simplificar muchas veces se utilizan datos que han de estar guardados en el disco A:\ .

Los ficheros utilizados se pueden obtener ejecutando el siguiente código:

Nota: Es imprescindible tener colocado un disquete en "A:\", dado que allí es dónde se almacenarán los datos. Si se desea, se puede cambiar el código para que el almacenamiento se realice en otra ubicación.

*Datos.dat;

```
DATA A;
FILE "A:\DATOS.DAT";
INPUT NUM_PAC 1-2 TRATAM 3 INIC $ 4-6 SEXO 7 EDAD 8-9;
PUT NUM_PAC 1-2 TRATAM 3 INIC $ 4-6 SEXO 7 EDAD 8-9;
CARDS;
  51ABC143
  62FRE173
  71SAD122
  82GRC053
  91PPC035
 102LBB036
 111JVM044
 122RAR015
 131XBG175
 142MSD164
;
RUN;
```

*Datos2.dat;

```
DATA A;
FILE "A:\DATOS2.DAT" DLM="09"X;
INPUT NUM_PAC 1-2 TRATAM 4;
PUT NUM_PAC TRATAM;
CARDS;
  5 1
  6 2
  7 1
  8 2
  9 1
 10 2
 11 1
 12 2
 13 1
 14 2
;
RUN;
```

*Resul.sd2;

```
LIBNAME FIEBRE "A:\";  
DATA FIEBRE.RESUL;  
INPUT NUM_PAC FIEBRE0 FIEBRE1;  
CARDS;  
1 38 37  
2 39 37  
3 39 38  
4 40 37  
5 41 42  
6 40 36  
7 38 38  
8 39 36  
9 40 42  
10 41 37  
11 40 40  
12 42 37  
13 37 38  
14 39 38  
;  
RUN;
```

*Fruta.sd2;

```
LIBNAME PESO "A:\";  
DATA PESO.FRUTA;  
INPUT PESO ZUMO;  
CARDS;  
0.61 0.25  
0.52 0.23  
0.46 0.26  
0.63 0.32  
0.56 0.22  
0.61 0.36  
0.6 0.35  
0.54 0.31  
0.39 0.3  
0.28 0.1  
0.42 0.21  
0.41 0.15  
0.47 0.2  
0.54 0.36  
;  
RUN;
```

10. ANEXO 2: PROGRAMACIÓN MACRO

10.1. Introducción

La programación Macro es una herramienta muy útil para estandarizar, extender y la programación clásica en código SAS. Entre otras virtudes, permite reducir en gran medida la cantidad de código necesario para las tareas más frecuentes. En resumen, se podría decir que la programación Macro permite asignar a un conjunto de caracteres o de sentencias de código un nombre clave. Esta palabra clave será sustituida por el valor asignado al ser ejecutado y compilado el correspondiente código. La ventana de programación para crear Macros, es la ventana habitual de sintaxis.

La programación Macro tiene dos componentes principales:

- El procesador Macro, encargado de “traducir”.
- El lenguaje Macro, la sintaxis que se debe utilizar para comunicarse con el procesador Macro.

Al compilar un texto, existen dos caracteres que activan al procesador Macro:

- & : &nombre , que designará a las variables Macro.
- % : %nombre , que designará a las Macros creadas por nosotros o ya residentes en el sistema.

10.1.1. Variables Macro

Ejemplo:

```
%let city = Barcelona;  
Title "Ciudad: &city";    ->    Title "Ciudad: Barcelona";
```

La instrucción %let permite asignar a un nombre un conjunto de caracteres utilizando la sintaxis:

```
% let nombre = literal ;
```

Las referencias a variables Macro ya creadas siempre serán precedidas por el símbolo: “&”.

Los nombres de variables Macro no pueden contener caracteres extraños como:

; , espacio “ ” AND OR , etc.

Al utilizar referencias de variables Macro en expresiones entre comillas es necesario utilizar las comillas dobles: “ ” para obtener la asignación deseada. En caso de comillas simples: ‘ ’ no se obtiene el valor correcto.

10.1.2. Generación de Código mediante Macros

Ejemplo:

```
%macro b;  
a  
%mend b;
```

```
proc print data=%b;          ->    proc print data=a;  
run;                          run;
```

Para invocar una macro ya creada, es necesario escribir el prefijo: "%". Aunque invocar una macro parece una instrucción de SAS, no es necesario acabar la instrucción con ";" como es habitual.

10.1.3. Macros con código SAS

Ejemplo:

```
%macro pl;  
proc plot;  
plot y*x;  
run;  
%mend;
```

```
%pl;                ->      proc plot;  
                        plot y*x;  
                        run;
```

10.1.4. Macros con parámetros de entrada

```
%macro pl (data=,yvar=,xvar=);  
proc plot data = &data ;  
plot &yvar * &xvar ;  
run;  
%mend;
```

```
%pl (data=a, yvar=y, xvar=x);  ->      proc plot data=a;  
                                    plot y*x;  
                                    run;
```

Los parámetros se pueden definir de varias maneras:

<u>Definición de la Macro:</u>	<u>Llamada a la Macro</u>
--------------------------------	---------------------------

%macro pl (data=,yvar=,xvar=);	%pl (data=a, yvar=y, xvar=x);
%macro pl (data,yvar,xvar);	%pl (a,y,x);
%macro pl (data=a,yvar=y,xvar=x);	%pl (xvar=x);
(en este último caso se definen valores por defecto).	

10.1.5. Comentarios en las macros

Dentro de una macro se pueden insertar comentarios utilizando:

```
%* comentario ;
```


10.1.6. Generación de código mediante condiciones

Ejemplo:

```
%macro quehago(info=);  
%if &info=a %then %do;  
    proc print;  
    run;  
%else %if &info=b %then %do;  
    proc print noobs label;  
    run;  
%else %put "NO SE QUE HACER" &INFO "NO ME SIRVE";  
%mend;
```

La instrucción %IF ... %THEN ... ; %ELSE ... ; es la correspondiente instrucción condicional del código Macro. Cuando las transformaciones a realizar son varias, éstas deben de escribirse entre un %DO; y un %END;

Para optimizar el código anterior se podría utilizar la función %UPCASE y así evitar posibles confusiones entre mayúsculas y minúsculas.

La instrucción %PUT escribe un literal o el valor de una variable Macro en la ventana LOG.

10.1.7. Generación de código repetitivo

Ejemplo:

```
%macro a (num);  
%do i=1 %to &num;  
    %put &num "Me portaré bien en clase";  
%end;  
%mend;  
  
%macro a (num,data);  
%do i=1 %to &num;  
    proc print data=&data&num;  
    run;  
%end;  
%mend;  
  
%a(2,b)      ->      proc print data=b1;  
                      run;  
                      proc print data=b2;  
                      run;
```

Mediante el uso de Macros es posible y muy recomendable la generación de código repetitivo.

10.2. VARIABLES MACRO

Las variables Macro tienen una longitud máxima de 32K caracteres. Su longitud es determinada por la longitud del texto asociado.

Las variables Macro contienen tan sólo caracteres, aunque existe la posibilidad de realizar operaciones entre los valores numéricos de variables Macro.

10.2.1. Variables Macro definidas por el sistema

Algunas de tales variables son:

Sysdate	Fecha
Sysday	Día
Syslast	El último dataset analizado
Syserr	Resultado del paso data.

10.2.2. Variables Macro definidas por el usuario

Tales variables deben tener un nombre que empiece por una letra o “_” aunque no es recomendable que empiecen por SYS, AF, DMS, SQL debido a que son nombres utilizados frecuentemente por el sistema.

Se puede escribir %put _all_; para ver todas las variables Macro creadas por el usuario.

10.2.3. Asignar valores a Variables Macro

La forma más simple de asignar un valor a una variable Macro es mediante la instrucción %let.

Posibles maneras de crear variables Macro:

%let	Asignación simple
%do iterativo	El contador
%global	Definen variables Macros que serán utilizadas en varias Macros y código SAS
%local	Definen variables Macro que serán utilizadas sólo en una Macro
%input	Permite asignar valores a variables Macros dentro del código Macro
%macro	Dentro de una macro
Symput	Dentro de un paso DATA
Into	Dentro del Proc SQL
%window	Dentro de una ventana Macro

Ejemplos de asignaciones con %let:

```
%let calle=Diagonal;
%let calle=
    Diagonal;
%let calle=    Diagonal ;
(todas producen el mismo resultado: Diagonal).
```

```
%let num=123+123
(num tiene el valor 123+123).
```

```
%let num=%eval(123+123);
(num tiene el valor 246).
```

```
%let num=%sysevalf(123.5+122.5);
```

(La función %sysevalf realiza operaciones en punto flotante, num tiene el valor 246).

```
%let casa=;
```

(casa tiene valor nulo).

```
%let nombre=luis;  
%let nick=&nombre;
```

(&nombre se convierte en luis y este valor es asignado a la variable nick).

```
%let mac=%pl;
```

(se asigna a mac el valor de la ejecución de la macro pl).

```
%let mac=%nrstr(%pl);
```

(se asigna a mac el literal "%pl").

```
%let plo = %str(proc print;run;);
```

(se asigna proc print;run; evitando así los ";" intermedios).

```
Data a;  
Set b;  
If _n_ = _last_ then CALL SYMPUT ('num', left(x));  
Run;
```

(se asigna a la variable Macro num el valor de la última observación de la variable "x" del dataset "a").

10.2.4. Referencias a variables Macro

Ejemplos

```
%let name=Juan;  
%let name1=Pepe;  
%let i=1;  
%let Juan1=María
```

&name1	-> Pepe
&name&i	-> Juan1
&&name&i	-> Pepe
&name.1	-> Juan1
hola&name	-> holaJuan
&name.&name	-> JuanJuan
&name..&name	-> Juan.Juan
&&&name&i	-> María

10.2.5. Visualizar Variables Macro

%put	Escribe en el Log.
Options symbolgen;	Escribe en el Log todas las asignaciones que vamos realizando.
Options Mprint	Escribe el código generado por las Macros en el Log.