

Universidad de Los Andes
Facultad de Ciencia Económicas y Sociales
Escuela de Estadística

Computación Estadística con SAS.

Autor: Angel Alberto Zambrano

Presentado como trabajo de ascenso para optar a la categoría de Asociado

Octubre-2016

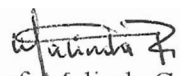
**UNIVERSIDAD DE LOS ANDES
MÉRIDA – VENEZUELA**

**FACULTAD DE CIENCIAS ECONÓMICAS Y SOCIALES
ESCUELA DE ESTADÍSTICA**

ACTA – VEREDICTO

Nosotros, Miembros del Jurado designado por el Ilustre Consejo de la Facultad de Ciencias Económicas y Sociales en su Sesión No. 24/16 de fecha 01 de noviembre de 2016 para conocer y decidir acerca de la Credencial de Mérito para el Ascenso del Profesor **ÁNGEL ZAMBRANO**, Cédula de Identidad N° 8.027.708 a la Categoría de Profesor **ASOCIADO** conforme al Artículo N° 165 del Estatuto del Personal Docente y de Investigación de la Universidad de Los Andes, reunidos en el Instituto de Estadística Aplicada y Computación de la Facultad de Ciencias Económicas y Sociales-Mérida, el día seis de diciembre de dos mil dieciséis, una vez revisado, discutido y realizada la correspondiente exposición pública del trabajo titulado: “Computación Estadística con SAS” y por cuanto consideramos que éste cumple con todos los requisitos exigidos para un Trabajo de Ascenso, acordamos impartir su **APROBACIÓN** para los fines a los cuales ha sido propuesto.


Prof. Elizabeth Torres Rivas
Coordinadora


Prof. Malinda Coa Ravelo
Miembro del Jurado


Prof. Guðberto León Rangel
Miembro del Jurado



DEDICATORIA

Dedicado a mi esposa y mis 3 Ángeles

AGRADECIMIENTOS

- A la profesora Elizabeth Torres, quien me introdujo a través de sus clases en el área de la Computación Estadística y el SAS
- A los estudiantes que tuve durante las diversas ocasiones que dicté la asignatura Computación Estadística tanto en la Licenciatura de Estadística como en la Maestría de Estadística, gracias a ellos ésta guía fue creciendo hasta ser lo que presento como Trabajo de Ascenso.

INTRODUCCIÓN

En la Licenciatura de Estadística se enseña en el área de Computación, además de la programación de computadoras mediante algún lenguaje de alto nivel, los diferentes software de procesamiento estadístico, tales como, el MS Excel, el SPSS, el lenguaje R y el SAS entre otros.

El SAS se dicta en una asignatura denominada “Computación Estadística”, en sus primeros tiempos con la figura de Tópicos Especiales y ahora como una asignatura electiva de la carrera.

El sistema SAS es un conjunto integrado de programas independientes del hardware desarrollados por el SAS Institute, para el procesamiento y análisis de las necesidades de información en los negocios, la industria, la educación y el gobierno.

El presente trabajo es el resultado de las múltiples veces que he dictado la asignatura “Computación Estadística” tanto en la Licenciatura como en la Maestría de Estadística Aplicada, y que en cada una de esas oportunidades ha ido creciendo hasta llegar a lo que hoy estoy presentando.

El trabajo está dividido en 8 partes o Capítulos, en los cuales se abarca desde el estudio del programa y su ambiente de trabajo, hasta la descripción de los principales procedimientos de los módulos SAS/BASE, SAS/STAT, SAS/GRAPH, SAS/IML y el manejo de macros.

En el primero se desarrollan los Fundamentos del SAS: el sistema SAS con sus componentes, el gestor o manejador de pantallas con la descripción de los menús y las ventanas, y por último la estructura de un programa en SAS con sus principales elementos.

En el segundo y tercero se estudian el paso de datos (DATA step) y el paso de procedimientos (PROC step). En el cuarto se estudia a profundidad el paso de datos con la manipulación y creación de los archivos de datos sas (SAS dataset),

En el quinto se estudia los procedimientos del módulo estadísticos del SAS/STAT, en el sexto se estudia los procedimientos del módulo gráfico SAS/GRAPH, en el séptimo se estudia el módulo SAS/IML (Lenguaje Interactivo de Matrices) y por último en el octavo una breve introducción al manejo de macros en SAS.

CONTENIDO

DEDICATORIA	I
AGRADECIMIENTOS	II
INTRODUCCIÓN	III
CONTENIDO	IV
INDICE DE FIGURAS.....	VIII
INDICE DE TABLAS	IX
1. FUNDAMENTOS DEL SISTEMA SAS.....	1
1.1 EL SISTEMA SAS.....	1
1.1.1 Componentes del Sistema SAS.....	2
1.1.2 Tareas del Sistema SAS.....	4
1.2 EL GESTOR O MANEJADOR DE PANTALLAS DEL SAS (SAS DISPLAY MANAGER)	4
1.2.1 Descripción del sistema de Menús del SAS 9.1	6
1.2.2 Descripción de la barra de herramientas.....	14
1.2.3 Descripción de las ventanas del área de trabajo.....	15
1.2.4 La ventana Keys (Teclas rápidas).....	17
1.3 UN PROGRAMA SAS	18
1.3.1 Estructura general de un programa SAS	19
1.3.2 Mi primer Programa SAS	20
1.4 CONJUNTOS DE DATOS SAS	21
1.5 LIBRERÍA SAS.....	23
1.6 INSTRUCCIONES GLOBALES DEL SAS.....	25
1.6.1 ENDSAS BYE	25
1.6.2 FILENAME	25
1.6.3 FOOTNOTE.....	25
1.6.4 LIBNAME.....	25
1.6.5 OPTIONS	26
1.6.6 RUN.....	28
1.6.7 TITLE	28
1.6.8 X.....	28
2. EL PASO DE DATOS (DATA STEP).....	29
2.1 INSTRUCCIONES DEL DATA STEP	30
2.1.1 La instrucción CARDS.....	30
2.1.2 La instrucción DATA.....	30
2.1.3 La Instrucción FILE	30
2.1.4 La Instrucción INFILE.....	30
2.1.5 La instrucción INPUT.....	31
2.1.6 La instrucción PUT.....	32

2.2	ENTRADA DE DATOS EN MODO COLUMNA	32
2.3	ENTRADA DE DATOS EN MODO LISTA	34
2.4	ENTRADA DE DATOS CON FORMATO	34
2.5	LISTAS DE FORMATOS DE ENTRADA Y NOMBRE DE VARIABLES NUMERADA	35
2.6	LECTURA DE REGISTROS CON MÚLTIPLES LÍNEAS	36
2.7	LEYENDO VARIOS REGISTROS POR LÍNEA DE ENTRADA	38
3.	EL PASO DE PROCEDIMIENTOS	39
3.1	INSTRUCCIONES GENERALES USADAS EN EL PASO DE PROCEDIMIENTOS	39
3.1.1	<i>Instrucción BY</i>	39
3.1.2	<i>Instrucción FOOTNOTE</i>	39
3.1.3	<i>Instrucción TITLE</i>	40
3.1.4	<i>Instrucción VARIABLES o VAR</i>	40
3.2	PROCEDIMIENTOS DEL SAS BASE	41
3.2.1	<i>Procedimiento DBF</i>	41
3.2.2	<i>El procedimiento CHART</i>	41
3.2.3	<i>El procedimiento CONTENTS</i>	42
3.2.4	<i>Procedimiento CORR</i>	43
3.2.5	<i>El procedimiento FORMAT</i>	44
3.2.6	<i>El procedimiento FREQ</i>	45
3.2.7	<i>El procedimiento MEANS</i>	46
3.2.8	<i>El procedimiento PLOT</i>	47
3.2.9	<i>El procedimiento PRINT</i>	47
3.2.10	<i>El procedimiento SORT</i>	48
	<i>El procedimiento UNIVARIATE</i>	48
3.3	OTROS PROCEDIMIENTOS DEL SAS BASE	50
4.	PROGRAMANDO EL PASO DE DATOS	52
4.1	OPCIONES DEL SAS DATA SET	52
4.2	INSTRUCCIÓN DE ASIGNACIÓN	53
4.3	INSTRUCCIÓN ATTRIB	56
4.4	INSTRUCCIÓN DELETE	56
4.5	INSTRUCCIÓN FORMAT	56
4.6	INSTRUCCIÓN IF – THEN Y IF – THEN ELSE	56
4.7	INSTRUCCIÓN INFORMAT	57
4.8	INSTRUCCIÓN LABEL	57
4.9	INSTRUCCIÓN LIST	57
4.10	INSTRUCCIÓN PUT	57
4.11	INSTRUCCIÓN RETAIN	58
4.12	INSTRUCCIONES DE REPETICIÓN	58
4.13	INSTRUCCIÓN OUTPUT	60
4.14	USO DE ARREGLOS	60
4.14.1	<i>Arreglos con subíndice explícito</i>	61
4.14.2	<i>Arreglos con subíndice implícito</i>	63
4.15	INSTRUCCIÓN SET	64
4.16	INSTRUCCIÓN MERGE	66
4.17	INSTRUCCIÓN UPDATE	69

5.	INTRODUCCIÓN AL SAS/STAT	70
5.1	PROCEDIMIENTOS DEL SAS/STAT	70
5.1.1	<i>Procedimientos de regresión:</i>	70
5.1.2	<i>Procedimientos de análisis de varianza:</i>	71
5.1.3	<i>Procedimientos para datos categóricos:</i>	71
5.1.4	<i>Procedimientos multivariantes:</i>	71
5.1.5	<i>Procedimientos de análisis cluster:</i>	72
5.1.6	<i>Procedimientos de análisis de sobrevivencia:</i>	72
5.2	PROCEDIMIENTO ANOVA	72
5.3	PROCEDIMIENTO GLM	75
5.4	PROCEDIMIENTO PRINCOMP	78
5.5	PROCEDIMIENTO REG	80
5.6	PROCEDIMIENTO TTEST	82
5.7	EJEMPLOS SAS/STAT	83
6.	INTRODUCCIÓN AL SAS/GRAPH.....	88
6.1	INSTRUCCIONES GLOBALES DEL SAS/GRAPH	88
6.1.1	<i>Instrucción AXIS</i>	88
6.1.2	<i>Instrucción BY</i>	88
6.1.3	<i>Instrucción GOPTIONS</i>	88
6.1.4	<i>Instrucción LEGEND</i>	88
6.1.5	<i>Instrucción PATTERN</i>	88
6.1.6	<i>Instrucción SYMBOL</i>	88
6.1.7	<i>Instrucción TITLE, NOTE y FOOTNOTE</i>	89
6.2	DESTINOS PARA ENVIAR LA SALIDA DEL SAS/GRAPH	89
6.2.1	<i>Enviando la salida a la ventana GRAPH</i>	89
6.2.2	<i>Enviando la salida a un archivo</i>	89
6.2.3	<i>Enviando la salida a un archivo rtf (MS Word)</i>	89
6.2.4	<i>Enviando la salida a un archivo pdf</i>	90
6.3	CAMBIANDO LA APARIENCIA DEL GRÁFICO.....	90
6.4	PROCEDIMIENTOS DEL SAS/GRAPH	92
6.4.1	<i>Procedimiento GCHART</i>	92
6.4.2	<i>Procedimiento GPLOT</i>	101
7.	INTRODUCCIÓN AL SAS/IML.....	107
7.1	EXPRESIONES	109
7.2	FUNCIÓN.....	109
7.3	OPERADORES EN IML	110
7.4	MANEJO DE SUBÍNDICES.....	116
7.5	IMPRIMIENDO MATRICES CON NOMBRES DE FILAS Y/O COLUMNAS.....	117
7.6	INSTRUCCIONES DE PROGRAMACIÓN	118
7.7	TRABAJANDO CON CONJUNTO DE DATOS SAS	122
7.8	ACCESO A ARCHIVOS EXTERNOS	125
7.9	OTRAS INSTRUCCIONES, FUNCIONES Y PROCEDIMIENTOS DEL IML	127
7.9.1	<i>Funciones que invocan matrices</i>	127
7.9.2	<i>Funciones aritméticas de matrices</i>	127
7.9.3	<i>Funciones escalares</i>	127
7.9.4	<i>Funciones de reducción</i>	128

7.9.5	<i>Funciones de manipulación de matrices.....</i>	128
7.9.6	<i>Funciones para manipulación de caracteres</i>	129
7.9.7	<i>Comandos y funciones de manejo de datos</i>	129
7.9.8	<i>Funciones algebraicas lineales y estadísticas</i>	130
8.	PROGRAMACIÓN CON MACROS EN SAS	132
8.1	MACROS	132
8.2	MACRO-VARIABLES	134
8.3	MACRO-VARIABLES DEFINIDAS POR EL SAS	136
8.4	INSTRUCCIONES MACRO	137
8.4.1	<i>Repita para</i>	137
8.4.2	<i>Repita mientras/hasta.....</i>	137
8.4.3	<i>Si condicional.....</i>	137
8.4.4	<i>Salto incondicional</i>	138
8.4.5	<i>Lectura de una macro-variable.....</i>	138
8.4.6	<i>Crear una <macro-variable> local</i>	138
8.4.7	<i>Escribir un texto.....</i>	138
	BIBLIOGRAFÍA.....	139

INDICE DE FIGURAS

FIG. 1. COMPONENTES DEL SISTEMA SAS.....	1
FIG. 2. PANTALLA TEMPORAL DE ENTRADA.	5
FIG. 3. GESTOR DE PANTALLA DEL SAS (DISPLAY MANAGER)	5
FIG. 4. BARRA DE HERRAMIENTAS	6
FIG. 5. BARRA DE ESTADO.....	6
FIG. 6. MENÚ FILE	8
FIG. 7. MENÚ EDIT	8
FIG. 8. MENÚ VIEW	9
FIG. 9. MENÚ TOOLS	9
FIG. 10. MENÚ RUN	11
FIG. 11. MENÚ SOLUTIONS	11
FIG. 12. MENÚ WINDOW	12
FIG. 13. MENÚ HELP.....	12
FIG. 14. VENTANAS ADICIONALES ANCADAS Y NO ANCLADAS.	13
FIG. 15. VENTANA DE AYUDA Y DOCUMENTACIÓN DEL SISTEMA SAS.....	13
FIG. 16. LIBRERÍAS DE EJEMPLOS CREADAS.	14
FIG. 17. EDITOR AVANZADO.	16
FIG. 18. VENTANA RESULTS.....	17
FIG. 19. VENTANA EXPLORER.....	17
FIG. 20. BARRA DE PESTAÑAS DE VENTANA.....	17
FIG. 21. VENTANA KEYS.	18
FIG. 22. ESTRUCTURA GENERAL DE UN PROGRAMA SAS.....	20
FIG. 23. FUNCIONAMIENTO DEL PASO DE DATOS.	23
FIG.24. ESTRUCTURA DE UN DATA SET.....	23
FIG.25. BARRA DE HERRAMIENTAS DE LA CARPETA LIBRARIES.	24
FIG.26. CUADRO DE DIÁLOGO NEW LIBRARY.	24
FIG. 27. EJECUCIÓN DEL PASO DE DATOS.	29
FIG 28. ÁREA DE SALIDA GRÁFICA.....	91
FIG 29. PARÁMEETROS DEL ÁREA DE SALIDA GRÁFICA	91

INDICE DE TABLAS

TABLA 1. CONTROL APUNTADOR DE COLUMNA	35
TABLA 1. FORMATOS DE ENTRADA (INFORMAT) MAS USADOS.	35
TABLA 3. RESUMEN DE LAS CARACTERÍSTICAS DE LOS PROCEDIMIENTOS DE ESTADÍSTICAS BÁSICAS.....	51
TABLA 4. OPERADORES DEL SAS.	54
TABLA 5. FUNCIONES DEL SAS/BASE.	55
TABLA 6. MACRO-VARIABLES AUTOMÁTICAS DEL SAS.....	136

1. Fundamentos del sistema SAS

1.1 El Sistema SAS

[1][2]y[3]

El sistema SAS es un conjunto integrado de programas independientes del hardware desarrollados por el SAS Institute, para el procesamiento y análisis de las necesidades de información en los negocios, la industria, la educación y el gobierno.

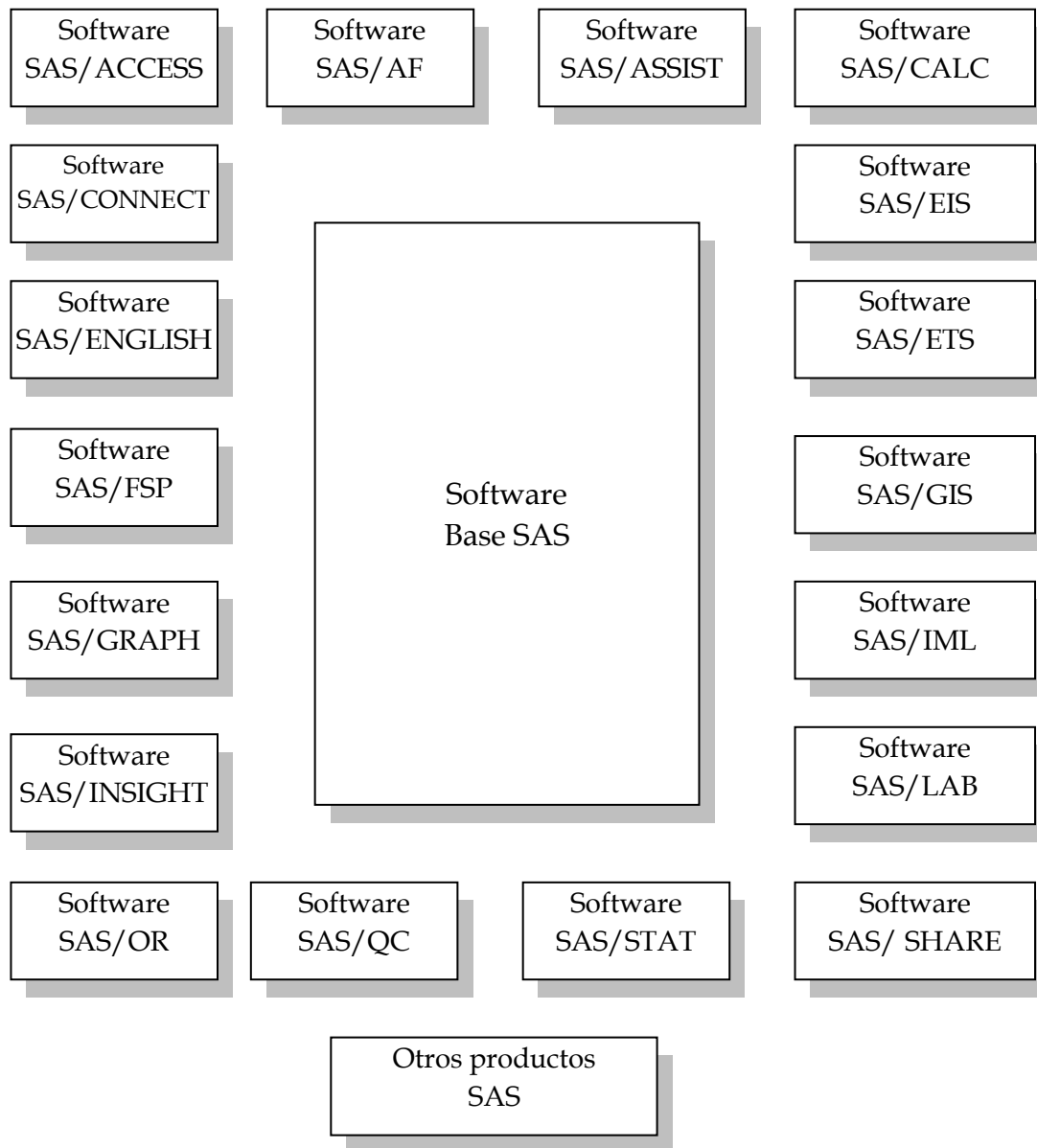


Fig. 1. Componentes del Sistema SAS.

El diseño del sistema SAS es modular y se compone de un gran número de módulos y se configura fácilmente a las necesidades de la organización. En la figura 1 se muestra los componentes del sistema SAS.

El sistema SAS es portable entre diferentes tipos de plataformas y entre diferentes Sistemas Operativos (UNIX, LINUX, WINDOWS, OS/2 y otros). Un programa puede ser ejecutado con los mismos resultados sobre un PC, estaciones de trabajo, minicomputadoras o supercomputadoras.

La arquitectura MVA (MultiVendor Architecture - Arquitectura Multi Proveedor) del SAS hace que el sistema tenga el mismo aspecto tanto en hardware como en sistemas operativos diferentes.

El sistema SAS se puede invocar y ejecutar en

- modo "batch" o por lotes (Sistema Operativo MVS para mainframes)
- en modo no interactivo (usuarios del sistema operativo CMS y PC)
- en modo de línea no interactivo
- en modo de ventanas interactivas (SAS Display Manager)
- modo dirigido por menús interactivos o aplicaciones creadas con SAS/ASSIST, SAS/AF o SAS/EIS.

Para el desarrollo de la asignatura Computación estadística y por ende del presente documento se ejecutará en modo de ventanas interactivas (SAS Display Manager).

1.1.1 Componentes del Sistema SAS

A continuación se detalla cada uno de los componentes del Sistema SAS:

- **SAS Base**, es la parte fundamental del sistema SAS; provee el acceso, manejo, análisis y presentación de datos con un poderoso ambiente de desarrollo de aplicaciones.
- **SAS/ACCESS**, provee enlace entre el sistema SAS y bases de datos como: DB2, SQL/DS, ORACLE, ADABAS, INFORMATION y otros.
- **SAS/AF**, utilidad interactiva usada crear aplicaciones de ventanas, tales como, menús y ayudas en línea.
- **SAS/ASSIST**, asistente para crear diversas tareas, tales como, acceso, manejo de datos, análisis y presentación de datos, creación de gráficos, etc.

- **SAS/CALC**, es una hoja de cálculo electrónica, con capacidad para hojas en 3D y gráficos integrados.
- **SAS/CONNECT**, permite a través de conexiones entre sesiones remotas SAS, transferir datos entre plataformas operativas diferentes que soportan el sistema SAS.
- **SAS/EIS**, para el desarrollo y mantenimiento de sistemas de información ejecutivos.
- **SAS/ENGLISH**, es una interface que permite hacer preguntas usando lenguaje natural para obtener información de alguna base de datos accesible por el sistema SAS.
- **SAS/ETS**, herramienta para el análisis de econometría, análisis y predicción de series de tiempo, modelado y simulación de sistemas, y análisis financiero.
- **SAS/FSP**, incluye procedimientos para crear aplicaciones de entrada de datos, generar cartas y reportes personalizados y pantallas de presentación de datos.
- **SAS/GIS**, permite mostrar, analizar e interactuar con datos espaciales, tales como mapas
- **SAS/GRAPH**, produce gráficos de alta calidad grafica en una gran variedad de patrones.
- **SAS/IML**, lenguaje interactivo de matrices, poderoso y flexible lenguaje de programación en un ambiente iterativo y dinámico.
- **SAS/INSIGHT**, herramienta interactiva para el análisis gráfico de datos.
- **SAS/LAB**, provee un ambiente integrado para ejecutar análisis de datos, tales como regresión y análisis de varianza. El software explica e interpreta los resultados, hace sugerencias para otros análisis y emite precauciones sobre violación de supuestos.
- **SAS/OR**, herramienta para la solución de problemas de investigación de operaciones (transporte, programación lineal, programación entera, programación y optimización no lineal, etc).
- **SAS/QC**, herramienta para la solución de problemas de control de calidad.
- **SAS/SHARE**, provee acceso concurrente a datasets.
- **SAS/STAT**, provee los análisis fundamentales estadísticos: análisis de varianza, regresión, análisis categórico, análisis multivariante, análisis cluster y análisis

no paramétrico entre otros.

- **SAS/TOOLKIT**, herramienta que le permite a los usuarios de SAS escribir sus propios procedimientos SAS, funciones, etc.

1.1.2 Tareas del Sistema SAS

El funcionamiento del SAS se centra alrededor de cuatro tareas dirigidas a datos almacenados, denominados “datasets”:

- Acceso a datos, crear y acceder a los datos que se requieren (Bases de datos, archivos de datos planos, comunicaciones entre datos).
- Manejo de datos, da a los datos el formato que la aplicación requiere (Introducción y edición de datos, recuperación de datos, formateo y conversión de datos).
- Análisis de datos, totaliza, reduce o transforma los datos planos en información útil y significativa (desde estadísticas descriptivas simples, Inferencia estadística, predicción y construcción de modelos, programación lineal hasta sofisticadas técnicas multivariantes).
- Presentación de datos, comunica información de forma que su significado se aprecia claramente (Creación de tablas e informes, gráficos de análisis, correspondencia de negocios y reportes impresos tanto en pantalla como en papel).

1.2 El Gestor o manejador de pantallas del SAS (SAS Display Manager)

Es la interfaz gráfica de ejecución del sistema SAS, que provee un control fácil y rápido de una sesión de trabajo en SAS mediante ventanas interactivas. Se utilizará la versión Portable 9.1.3 del SAS para Windows de 32bits.

Cuando se ejecuta el SAS se presenta la ventana temporal de la figura 2, en la cual se muestra información de la versión, y al finalizar la carga del programa se presenta el gestor de pantalla del SAS (Display Manager), ver figura 3.



Fig. 2. Pantalla temporal de entrada.

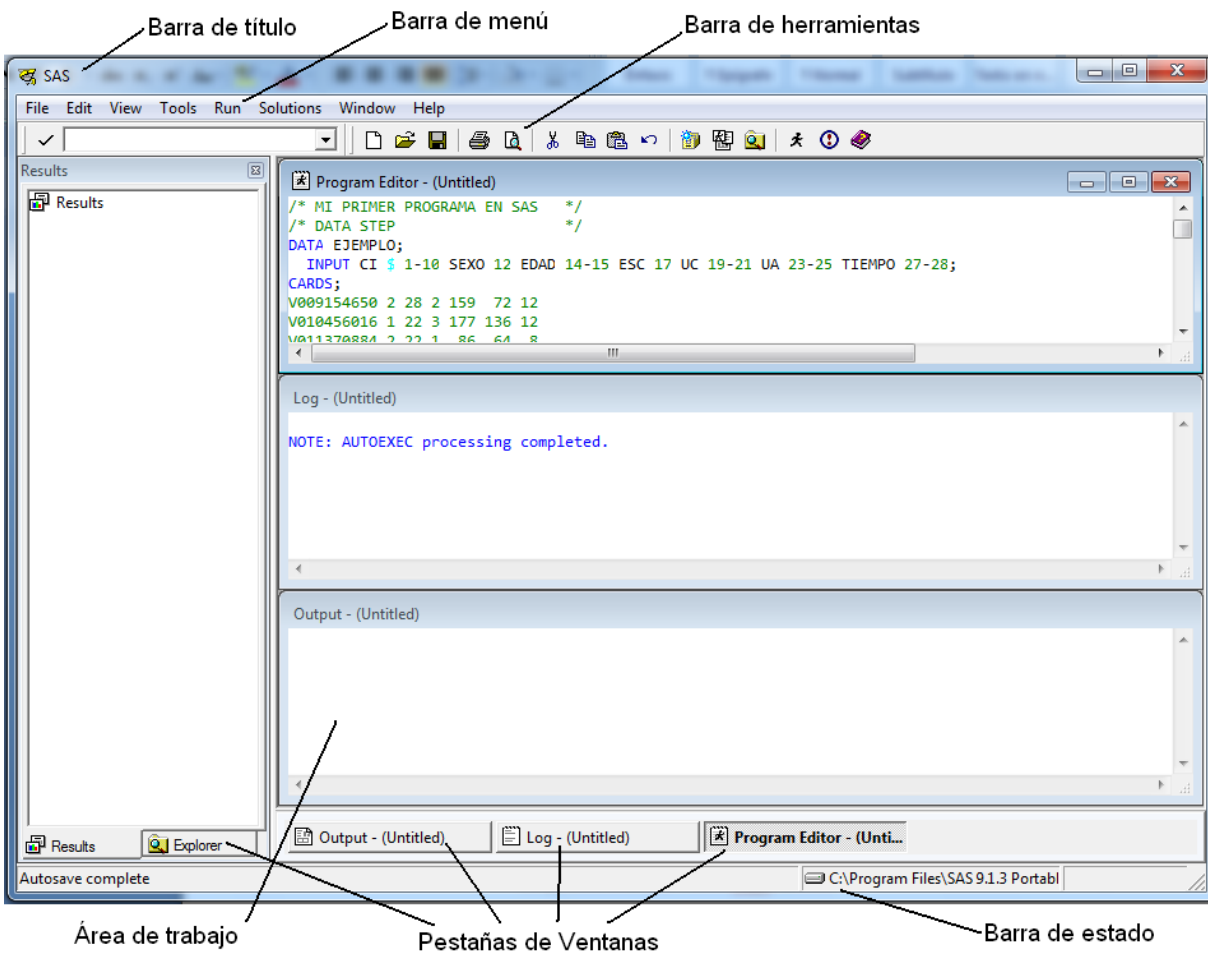


Fig. 3. Gestor de Pantalla del SAS (Display Manager)

Por ser una aplicación bajo Windows contiene los mismos elementos de una ventana windows:

- Barra de título, donde se presenta el icono del SAS y al final los botones de minimizar, maximizar y salir.
- Barra de menú, conjunto de menús desplegables con las diferentes opciones del sistema SAS (File, Edit, View, Tools, Run, Solutions, Windows y Help)
- Barra de herramientas, barra con iconos gráficos con las opciones más usadas de los menús (figura 4).
- Barra de estado, situada en la parte mas inferior de la venta, muestra información sobre la última acción realizada, de la librería actual y otros procesos. Si estamos situados en el editor nos informa también la posición del cursor: línea y columna (figura 5)
- Área de trabajo, es la parte central de la ventana y está formada por diversas ventanas de trabajo.



Fig. 4. Barra de Herramientas

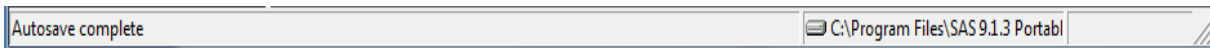


Fig. 5. Barra de Estado

1.2.1 Descripción del sistema de Menús del SAS 9.1










En la mayoría de las aplicaciones bajo Windows después de la barra de título se encuentra la barra de menú, que consiste de varios menús desplegables fijos con todas las opciones que contiene dicha aplicación.

Por lo general las aplicaciones tienen los menús de *File* (Archivo), *Edit* (Edición), *Windows* (Ventana) y *Help* (Ayuda), y dependiendo de la aplicación otros para realizar sus operaciones.

En el caso del SAS son: File, Edit, View, Tools, Run, Solutions, Windows y Help, los cuáles se explicaran con sus respectivas opciones a continuación. Algunas de las opciones de los menús desplegables tienen un icono gráfico al lado izquierdo, lo que significan que están disponibles en alguna de las barras de herramientas en algún






momento, así como también alguna de esas opciones tienen un atajo o combinación de teclas rápidas que se muestra hacia el lado derecho de la misma.


Menu File: contiene las diferentes opciones para el manejo de archivo (figura 6), si estamos sobre la ventana del editor (Edit) presenta las siguientes opciones:

-  **New**, abre archivo vacío para editar nuevo programa (atajo Ctrl+N)
-  **Open**, abre archivo existente (atajo Ctrl+O)
- **Close**, cierra el archivo actual
- **Append**, Agrega un archivo existente al final de la ventana del editor
-  **Open Object**, abre un objeto en el explorador
-  **Save**, guarda el archivo actual (atajo Ctrl+S)
- **Save as...**, guarda el archivo actual con otro nombre o formato
-  **Save as Object**, guarda el archivo actual con otro nombre o tipo de objeto
-  **Import Data...**, importa datos ASCII para el editor
-  **Export Data...**, exporta los datos del editor a un archivo ASCII
- **Page Setup...**, establece opciones de página
- **Print Setup...**, establece opciones de impresión
- **Print Preview**, muestra versión preliminar de la ventana actual
-  **Print**, imprime el contenido del editor o de la ventana actual (atajo Ctrl+P)
-  **Send Mail...**, envía el contenido de la ventana actual por correo electrónico con el manejador de correo por defecto del Windows
- **Archivos usados recientemente**, lista de los archivos abiertos recientemente
- **Exit**, salir del sistema Sas

Si estamos situados en otra ventana (Log, Output o Graph), las opciones se adaptan al contenido de la misma.

Menu Edit: contiene las opciones de edición del Sistema SAS (figura 7):

-  **Undo**, deshace la última acción (atajo Ctrl+Z)
- **Redo**, rehace la última acción
-  **Cut**, corta la selección y lo guarda en el portapapeles (atajo Ctrl+X)
-  **Copy**, copia la selección y lo guarda en el portapapeles (atajo Ctrl+C)
-  **Paste**, pega el contenido del portapapeles en la posición actual del cursor (atajo Ctrl+V)
-  **Clear**, borra la selección (atajo Del)
- **Clear All**, borra todo el contenido de la ventana actual (atajo Ctrl+A)

- **Select All**, selecciona todo el contenido de la ventana actual
- **Collapse All**, compacta todo el contenido
- **Expand All**, expande todo el contenido
- **Deselec**, quita la selección actual
-  **Find..**, busca un contenido en la ventana actual (atajo Ctrl+F)
- **Replace...**, reemplaza un contenido por otro en la ventana actual

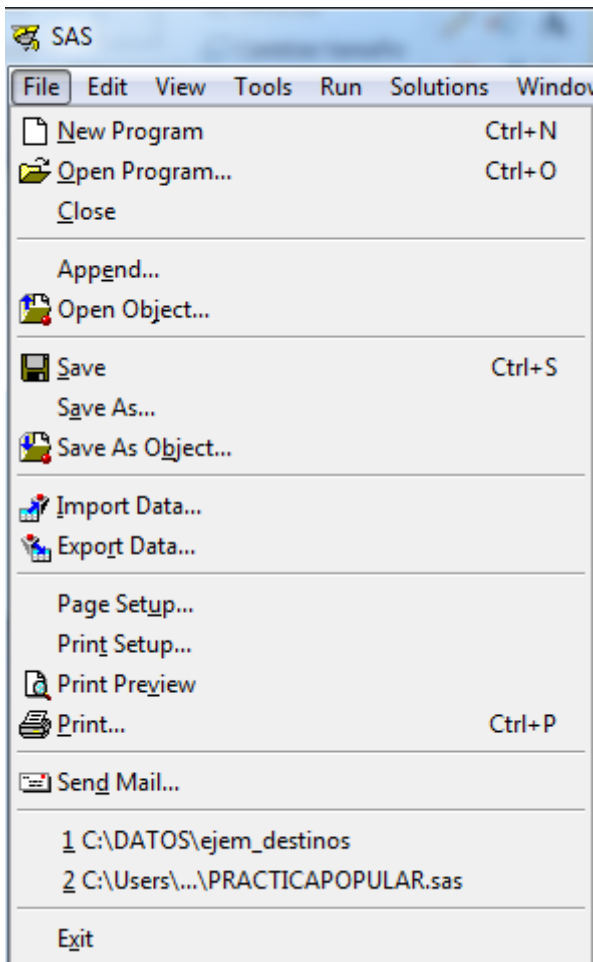


Fig. 6. Menú File

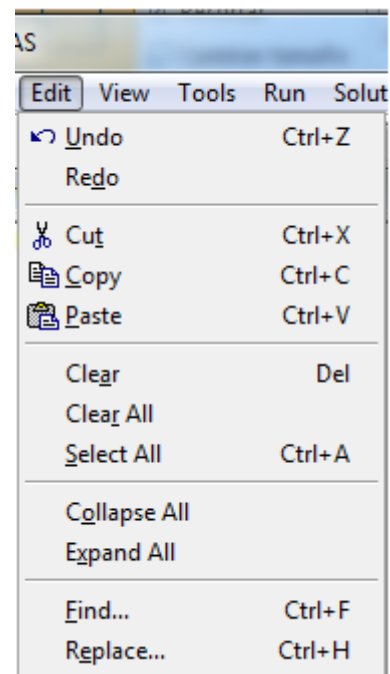











Fig. 7. Menú Edit

Menu View: contiene las opciones para visualizar las ventanas del área de trabajo del Sistema SAS (figura 8):

-  **Enhanced Editor**, abre la ventana del editor avanzado de programas
-  **Program Editor**, abre la ventana del editor de programas
-  **Log**, abre la ventana de mensajes
-  **Output**, abre la ventana de salida

-  **Graph**, abre la ventana de gráficos
-  **Results**, abre la ventana de exploración de resultados
-  **Explorer**, abre la ventana de exploración avanzada
-  **Contents Only**, abre la ventana de exploración solo elementos SAS
-  **My Favorite Folders**, abre la ventana de mis carpetas favoritas

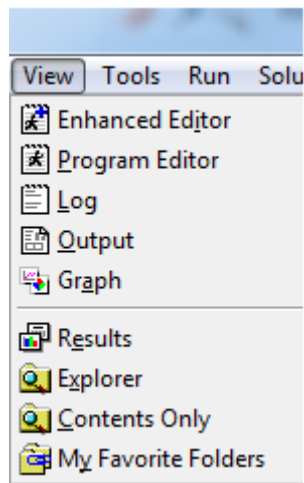


Fig. 8. Menú View

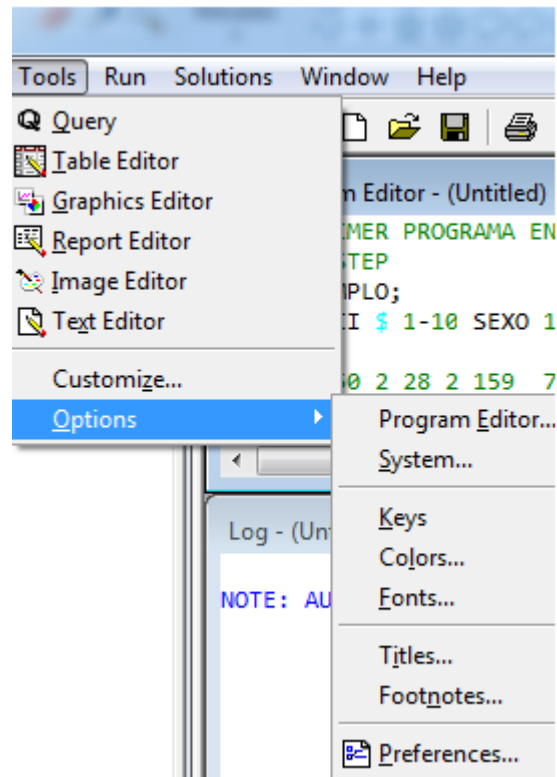








Fig. 9. Menú Tools

Menú Tools: contiene las opciones para invocar un editor específico de algún objeto del Sistema SAS (figura 9):

-  **Query**, para realizar consultas SQL sobre los “datasets”
-  **Table Editor**, para invocar el editor tabular de “datasets” nuevos o ya existentes mediante filas y columnas.
-  **Graphics Editor**, para invocar el editor de gráficos.
-  **Report Editor**, para invocar el generador de reportes.
-  **Image Editor**, para abrir el editor de imágenes
-  **Text Editor**, para abrir el editor de textos
- **Customize**, para configura los menús y barras de herramientas (se verá mas

adelante cuando se explique la barra de herramientas)

- **Options**, para configurar las diversas opciones del editor de programas, del sistema, las teclas, los colores, las fuentes, títulos, pie de páginas y otras preferencias

Menú Run: contiene las opciones para ejecutar el programa que está en la ventana del editor de programa ya sea normal o avanzado ya sea local o remoto, si esta seleccionada algunas instrucciones de la ventana sólo se ejecutan esas (figura 10):

- **Submit**, ejecuta el programa escrito actualmente e la ventana Edit o las instrucciones seleccionadas.
- **Recall Last Submit**, recupera en la ventana Edit la sintaxis del último programa ejecutado.
- **Submit Top Line**, ejecuta solo la primera línea del programa que está en la ventana Edit
- **Submit N lines...**, ejecuta las primeras N líneas del programa que está en la ventana Edit
- **Submit Clipboard**, ejecuta el programa que está actualmente en el portapapeles
- **Signon...**, realiza una conexión hacia un servidor remoto (SAS Metadata Server) mediante SAS/CONNECT
- **Remote Submit...**, hace que el programa que está en el editor se ejecute en el servidor remoto SAS, con el cual previamente se ha establecido una conexión.
- **Remote Get...**, recupera las ventanas log y output creadas por una ejecución remota asincrónica y las mezcla con las ventanas log y output locales.
- **Remote Display...**, muestra las ventanas log y output creadas por una ejecución remota asincrónica
- **Signoff...**, finaliza una conexión remota

Menú Solutions: para realizar diversos análisis estadísticas o soluciones para aquellos usuarios que le es difícil adaptarse a la programación del SAS (figura 11):

- **Analysis**, muestra una lista con todas los análisis que se pueden realiza en SAS mediante menús
- **Development and Programming**, para desarrollo y programación de Aplicaciones
- **Reporting**, para generar reportes
- **Accesories**, accesorios gráficos, de edición, juegos y otros

- **ASSIST**, invoca el asistente del SAS
- **Desktop**, muestra un escritorio propio del SAS que sustituye el menú Solutions en iconos para elegir trabajar en la gestión y acceso de datos, en reportes, en informes, en presentaciones, en análisis, en desarrollo y aplicaciones, en accesorios y otros.
- **EIS/OLAP Application Builder**, creador de aplicaciones OLAP

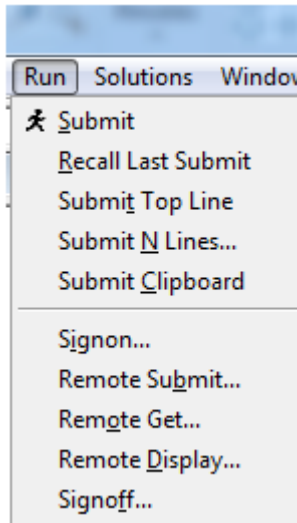


Fig. 10. Menú Run

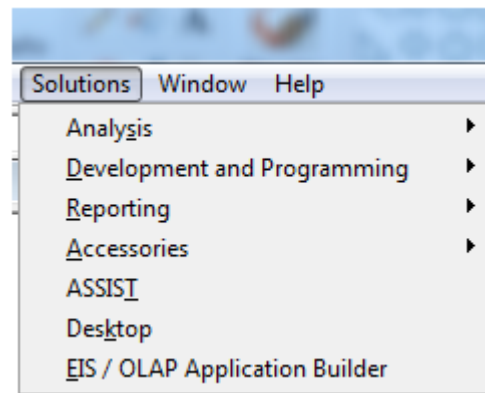


Fig. 11. Menú Solutions

Menú Window: contiene las opciones (figura 12):

- **Minimize All Windows**, minimiza todas ventanas abiertas del área de trabajo
- **Cascade**, organiza las ventanas abiertas del área de trabajo en forma de cascada (Atajo Shift+F5)
- **Tile Vertically**, organiza las ventanas abiertas del área de trabajo en forma de persianas verticales (Atajo Shift+F4)
- **Tile Horizontally**, organiza las ventanas abiertas del área de trabajo en forma de persianas horizontales (Atajo Shift+F3)
- **Resize**, recupera la distribución inicial de las ventanas
- **Size Docking View**, fijar el tamaño de las ventanas adicionales abiertas ancladas a la izquierda del área de trabajo (Resultados, Explorador y otras)
- **Docked**, activa y desactiva que se anclen las ventanas adicionales a la izquierda (figura 14)

- **Ventanas abiertas**, lista las diferentes ventanas abiertas del área de trabajo, pudiendo seleccionar alguna

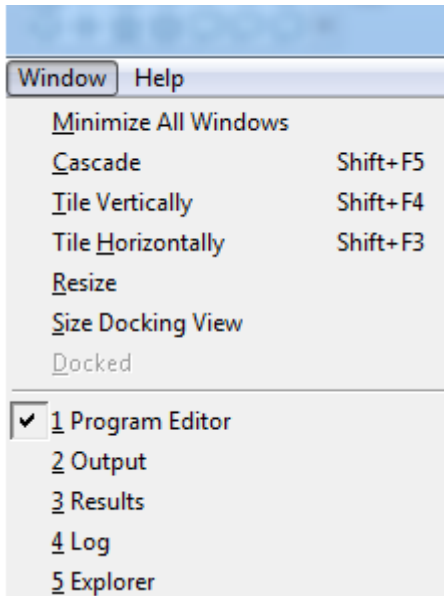


Fig. 12. Menú Window

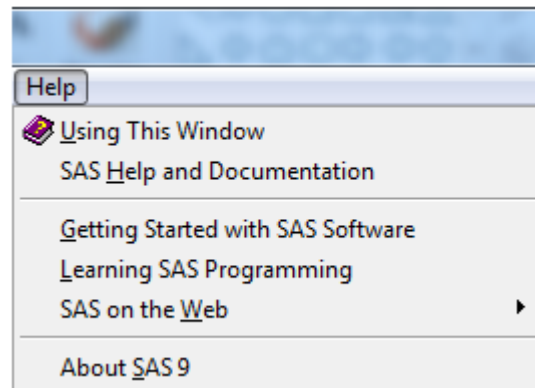


Fig. 13. Menú Help

Menú Help: contiene las opciones que proporcionan ayuda en el SAS (figura 13):

- **Use This Window**, presenta ayuda relativa a la ventana actual
- **SAS Help and Documentation**, presenta la ayuda y documentación del Sistema SAS (figura 15)
- **Getting Started with SAS Software**, primeros pasos con el Sistema SAS
- **Learning SAS Programming**, aprendiendo programación SAS, si tiene licencia SAS puede crear los ejemplos de datasets en forma de librerías, una vez creadas las puede buscar en la ventana del explorador → librerías (figura 16)
- **SAS on the Web**, permite acceso a soporte técnico por internet, a diversas FAQs y a la página principal de SAS institute.
- **About SAS 9**, presenta datos e información de la versión.

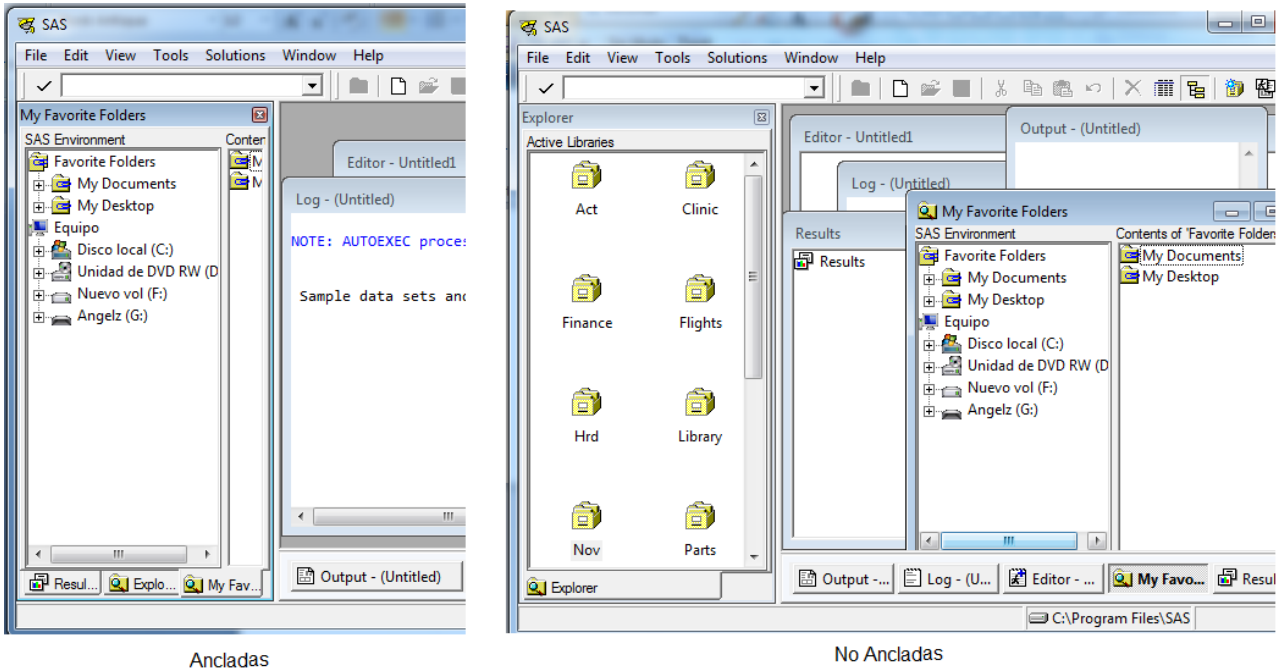


Fig. 14. Ventanas adicionales ancladas y no ancladas.

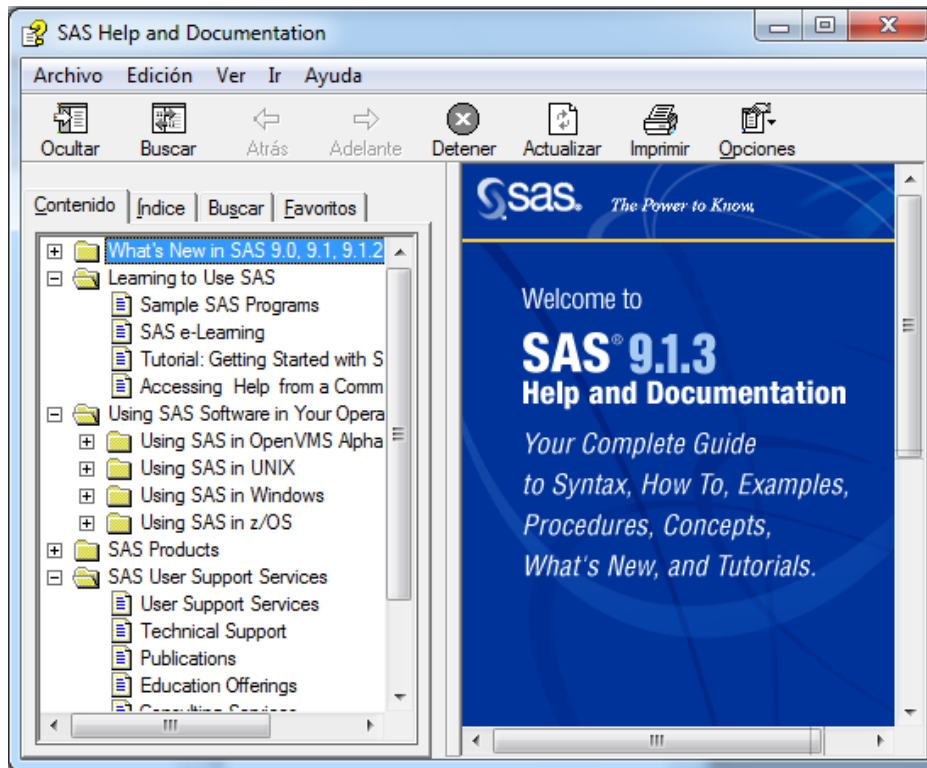


Fig. 15. Ventana de ayuda y documentación del Sistema SAS.

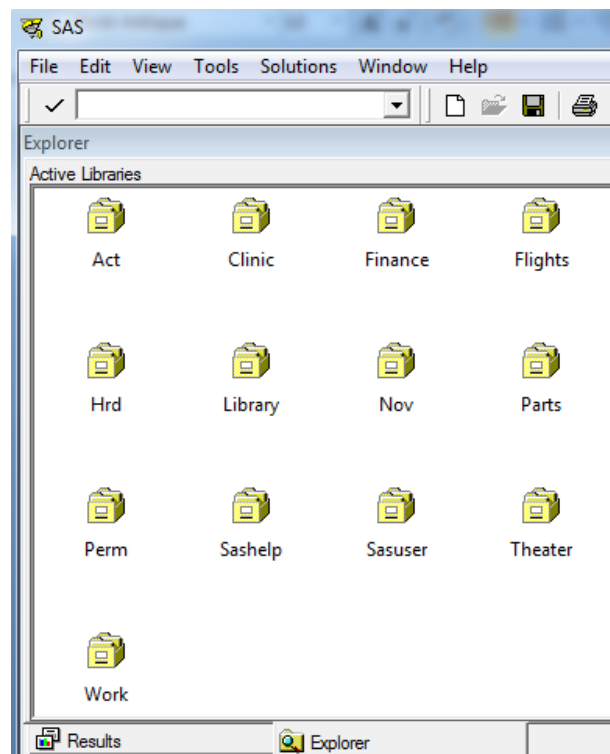


















Fig. 16. Librerías de ejemplos creadas.

1.2.2 Descripción de la barra de herramientas

Como se dijo anteriormente esta barra, que se muestra en la figura 4, contiene iconos o accesos directos gráficos que permiten acceder a las opciones del sistema de menú que mas se utilizan en el SAS sin necesidad de acudir a él. Al situar el puntero del ratón sobre un icono puede verse el significado del mismo.

De izquierda de derecha los iconos representan lo siguiente:

-  introducir comando, tocando la barra con el botón derecho puede ocultarlo
-  abrir archivo vacío para editar nuevo programa
-  abrir archivo existente
-  guardar archivo
-  imprimir archivo (ventana)
-  mostrar vista preliminar
-  cortar selección al portapapeles
-  copiar selección al portapapeles

-  pegar contenido del portapapeles
-  deshacer acción
-  crear librería nueva
-  abrir explorador de archivo (disco duro)
-  ejecutar programa del editor (submit)
-  borrar el contenido de la ventana actual
-  interrumpir ejecución del programa actual
-  ayuda

1.2.3 Descripción de las ventanas del área de trabajo

El sistema de ventanas del SAS está formado por un total de 9 ventanas, distribuidas en 2 grupos: de programación (Enhanced Editor, Program Editor, Log, Output y Graph) situadas hacia la parte derecha del área de trabajo; y de exploración (Results, Explorer, Contents Only y My Favorite Folders) ancladas hacia la parte izquierda del área de trabajo. Inicialmente se muestran 5 ventanas (figura 3), las otras se pueden activar y/o desactivar en el menú View.

A continuación se describen las mismas:

- **Ventana “Program Editor” (Editor de programas):** en esta ventana puede escribir, editar y ejecutar los programas SAS, así como también cualquier archivo de texto, por ejemplo los archivos de datos externos.
- **Ventana “Enhanced Editor” (Editor Avanzado de programas):** para escribir, editar y ejecutar los programas SAS (o archivos de datos) con mas funcionalidad que la ventana del editor de programa, como la posibilidad de señalar o marcar con distintos colores las diferentes partes y elementos del programa (figura 17), se separa los pasos de datos y/o procedimientos con líneas con la posibilidad de compactar y expandir los mismos.
- **Ventana “Log” (Registro):** en ella se muestra el registro (seguimiento) de la ejecución de los programas. En particular esta ventana interesa para ver posibles errores producidos durante la ejecución de los programas.
- **Ventana “Output” (Salida):** en la cual se presenta los resultados o salida de los diferentes procedimientos en forma de texto.
- **Ventana “Graph” (Gráficos):** en la cual se presenta los resultados o salidas gráficas de los procedimientos.

```

Editor - Untitled1 *

/* MI PRIMER PROGRAMA EN SAS */

/* Paso de Datos (DATA STEP) */
DATA EJEMPLO;
  INPUT CI $ 1-10 SEXO 12 EDAD 14-15 ESC 17 UC 19-21 UA 23-25 TIEMPO 27-28;
  CARDS;
V009154650 2 28 2 159 72 12
V010456016 1 22 3 177 136 12
V011370884 2 22 1 86 64 8
V010896200 1 24 1 189 121 13
V011090874 2 19 1 99 84 8
V010909774 2 23 1 114 75 11
V010310147 1 21 3 158 114 12
V010556950 1 24 4 181 152 15
V011383119 1 21 1 143 123 10
;
PROC PRINT;
  TITLE "DATOS ORIGINALES DE LOS ALUMNOS";
PROC MEANS;
  VAR EDAD UC UA TIEMPO;
  TITLE "ESTADISTICAS BASICAS PARA LAS VARIABLES NUMERICAS";

```

Fig. 17. Editor Avanzado.

- **Ventana “Results” (Resultados):** en la cual se presenta un resumen de toda la salida, tanto de texto como gráfica, que ha producido la ejecución de los programas, con posibilidad de navegar por todas las componentes de dicha salida. Cada vez que se ejecuta un programa se genera un componente que se va añadiendo a una especie de árbol de navegación, ver figura 18
- **Ventana Explorer (Explorador):** es una ventana de navegación o exploración sobre el sistema de archivos de los diferentes componentes de almacenamiento del computador, ver figura 19.
- **Ventana Contents Only (Contenido SAS):** es una ventana de exploración sobre los diferentes elementos u objetos del sistema SAS.
- **Ventana My Favorite Folders (Carpetas Favoritas):**

Para pasar de una ventana a otra se utiliza la barra de pestañas de ventanas (figura 20), situada en la parte inferior del área de trabajo y sobre la barra de estado; o activándola directamente con el ratón cualquiera de ellas. Otra forma es seleccionándola en la opciones Windows o View del menú principal.

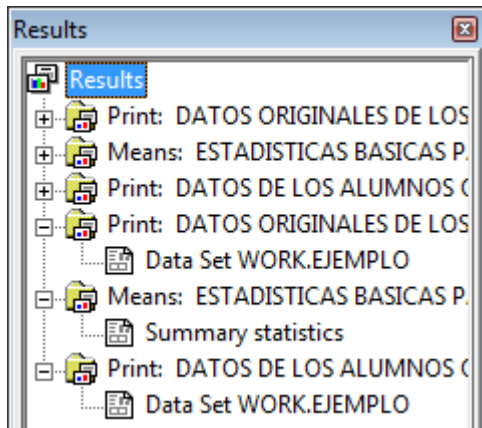


Fig. 18. Ventana Results

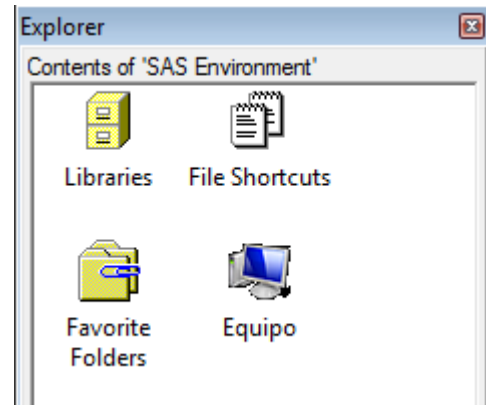


Fig. 19. Ventana Explorer

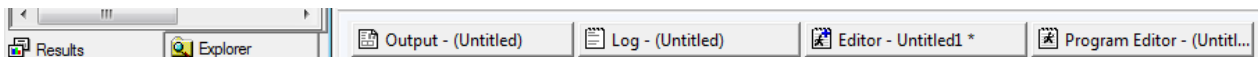




Fig. 20. Barra de pestañas de ventana.

Cualquier ventana se puede cerrar con el botón  de la respectiva ventana y ce abre seleccionándola en la opción View del menú principal.

1.2.4 La ventana Keys (Teclas rápidas)

Es una ventana especial donde se muestra la configuración de las teclas de función (F1, F2, ..., F12, Shf F1 ..., Ctr F1 ..., Alt F1..., Ctr A, Ctr B, ...), se puede acceder a ella de varias formas: a través de la opción del menú Tools->Options->Keys, o introduciendo keys en el cuadro de comandos  o presionando F9. La figura 21 muestra un extracto de ésta ventana.

Estos comandos pueden ser eliminados o modificados directamente sobre la misma ventana.

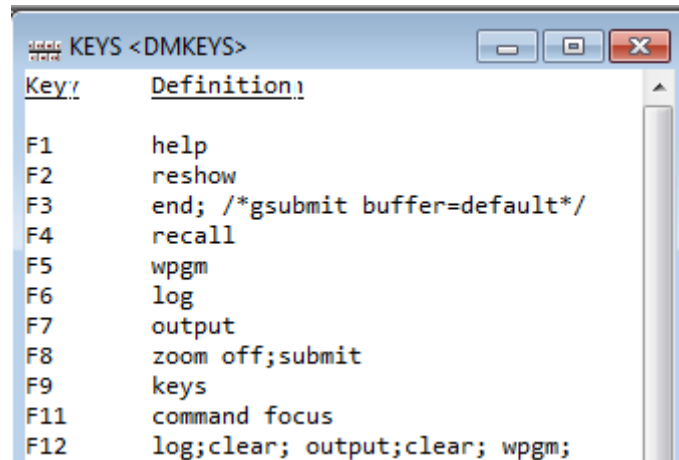


Fig. 21. Ventana Keys.

1.3 Un programa SAS

Un programa SAS consiste de instrucciones que terminan con punto y coma (;). Una instrucción es un conjunto de palabras reservadas, nombres o identificadores y operadores.

Las instrucciones en un programa SAS se escriben en formato libre; pueden empezar y finalizar en cualquier parte de una línea, pueden haber varias instrucciones en una línea, una instrucción puede continuar en varias líneas, como separador se utiliza uno o más espacios en blanco.

Una instrucción que comience con asterisco (*) es un comentario y es ignorado por el lenguaje SAS. Un comentario es un texto que sirve para documentar las instrucciones y partes del programa. Si un comentario va a ocupar mas de una línea se encierra entre /* y */.

Los caracteres ' y " se usan para en pares para delimitar las cadenas de caracteres o datos alfanuméricos.

Un Identificador o Nombre SAS es un conjunto o secuencia de hasta máximo 32 caracteres, el cual se usa para referirnos o identificar las variables, datos, "data sets", formatos, procedimientos, opciones o cualquier objeto dentro de un programa SAS. El identificador puede contener letras, dígitos o el carácter de subrayado, debe comenzar por una letra o el carácter de subrayado (_). No puede contener ni espacios en blancos ni caracteres especiales como #, %, &, etc.

Los siguientes son nombres válidos en SAS:

DATOS1	DATOS_1	CI	NROHORAS
NRO_HORAS	Curso1	EdoCivil	EDO_CIVIL
Apellidos_y_Nombres			

Los identificadores en SAS no son susceptibles a mayúsculas y minúsculas, esto significa que los siguientes ejemplos se refieren al mismo identificador:

NOMALUMNO	nomalumno	NOMalumno	NomAlumno
-----------	-----------	-----------	-----------

1.3.1 Estructura general de un programa SAS

En la figura 22 se muestra la estructura que posee un programa en SAS.

Algunas de las instrucciones son agrupadas en pasos (step) y las demás se denominan instrucciones globales. Hay dos tipos de pasos:

- Los DATA STEP o paso de datos, que permiten la creación de conjuntos de datos SAS denominados "SAS data sets", estos comienzan con la palabra reservada DATA.
- Los PROC STEP o paso de procedimientos, ejecutan un proceso o procedimiento SAS, comienzan con la palabra reservada PROC.

El fin de un paso es marcado por: la instrucción RUN; para algunos procedimientos con la instrucción QUIT; o por la palabra PROC o DATA que marque el comienzo del siguiente paso.

Se recomienda que la palabra PROC o DATA de un paso comience en la primera columna de una línea, las demás que estén indentadas o con sangría para asociarlas con el respectivo PROC o DATA.

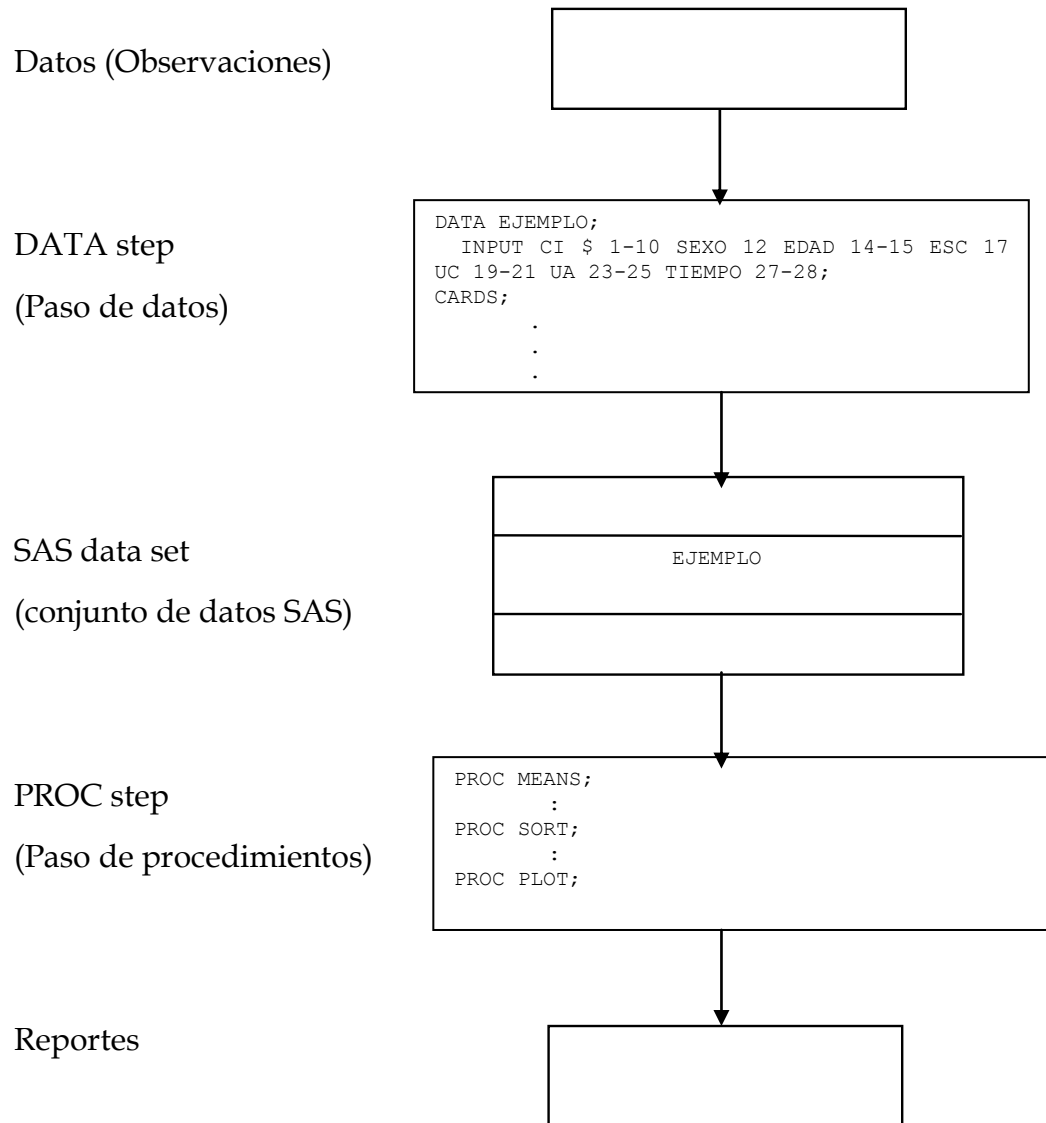


Fig. 22. Estructura general de un programa SAS.

1.3.2 Mi primer Programa SAS

El siguiente programa lee la Cédula de Identidad, el sexo, la edad, la escuela, unidades cursadas, unidades aprobadas y el tiempo en semestres de 20 alumnos de la Facultad de Ciencias Económicas y Sociales. Luego los imprime en la ventana OUTPUT, les calcula las estadísticas básicas (número, media, desviación, mínimo y máximo), los ordena por la cédula y los imprime ordenados.

```

/* MI PRIMER PROGRAMA EN SAS */

/* Paso de Datos (DATA STEP) */
DATA EJEMPLO;
  INPUT CI $ 1-10 SEXO 12 EDAD 14-15 ESC 17 UC 19-21 UA 23-25 TIEMPO 27-28;
CARDS;
V009154650 2 28 2 159 72 12
V010456016 1 22 3 177 136 12
V011370884 2 22 1 86 64 8
V010896200 1 24 1 189 121 13
V011090874 2 19 1 99 84 8
V010909774 2 23 1 114 75 11
V010310147 1 21 3 158 114 12
V005789432 1 27 3 210 107 13
V009329029 1 22 2 176 96 12
V011468864 2 20 3 96 82 9
V010897309 1 22 2 114 60 10
V011128730 1 21 3 159 142 11
V011128864 2 21 3 111 70 11
V009472815 2 24 1 107 87 11
V010556950 1 24 4 181 152 15
V011383119 1 21 1 143 123 10
V010712347 2 25 1 89 58 8
V011894350 2 24 1 104 87 11
V010714071 1 22 1 176 140 12
V010397305 1 23 3 214 152 13
;
PROC PRINT;
  TITLE "DATOS ORIGINALES DE LOS ALUMNOS";

PROC MEANS;
  VAR EDAD UC UA TIEMPO;
  TITLE "ESTADISTICAS BASICAS PARA LAS VARIABLES NUMERICAS";

PROC SORT;
  BY CI;

PROC PRINT;
  TITLE "DATOS DE LOS ALUMNOS ORDENADOS POR CEDULA";

RUN;

```

1.4 Conjuntos de datos SAS

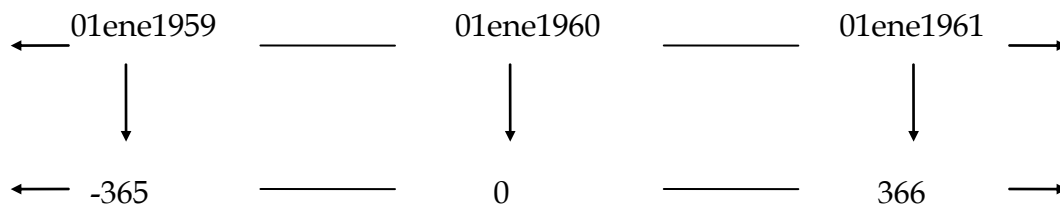
Un “data set” es un conjunto o colección de valores que toman las variables o campos de una entidad (observación) junto con la estructura descriptiva de los mismos, organizados en forma de una matriz rectangular (filas y columnas) y que puede ser reconocida y procesada por el sistema SAS.

Una variable es un dato que representa una característica de un objeto de interés para

el estudio. Las variables se representan por medio de nombres SAS y pueden ser de dos tipos:

- Variables carácter o alfanuméricas, consisten de cadenas de caracteres, los cuales pueden ser letras, dígitos y caracteres especiales. La longitud máxima de una cadena de caracteres es de 200.
- Variables numéricas, almacenan números que pueden estar precedido por un signo mas (+) o menos (-), representan cantidades numéricas. Están almacenados en formato de punto flotante en 8 bytes, lo que proporcionan espacio para 16 ó 17 dígitos.
- A los datos tipo fechas, SAS las considera variables numéricas especiales.

Un valor de fecha SAS es interpretado como el número de días entre el 1 de enero de 1960 y esa fecha.



Almacenar la fecha de esta forma permite tener la fecha y la hora al mismo tiempo en un mismo dato mediante fracciones de 1/24 para la hora, 1/1440 para los minutos y 1/86400 para los segundos. También permite hacer operaciones con fecha con una simple operación matemática.

La estructura rectangular de un conjunto de datos SAS implica que todas las variables y todas las observaciones tienen que tener un valor. Sin embargo, muchas veces existen valores ausentes. Por defecto, un valor ausente numérico se visualiza como un punto y un alfanumérico se visualiza como un espacio en blanco.

Una observación es un grupo de valores que representan los diferentes atributos o características de un objeto o entidad en estudio, una persona, una región, un país, un almacén, etc.

Un "data set" está constituido por dos áreas:

- Área de descriptores, donde se almacena la documentación del conjunto de datos, es decir, información descriptiva del conjunto de datos y sus variables. Los atributos del conjunto de datos incluyen nombre, fecha y hora de creación, etiqueta del conjunto de datos, número de observaciones, longitud del registro, fecha de la última modificación. La descripción de las variables incluyen atributos

tales como, nombre, tipo, longitud, formato, etiqueta e índice.

- Área de datos, en esta parte se almacenan los datos en forma rectangular (filas y columnas).



Fig. 23. Funcionamiento del paso de datos.

Las columnas en la tabla corresponden a las variables o campos de los datos y las filas las observaciones o registros. El número máximo de variables que se puede crear en un data set es 1024 mientras que en el número de observaciones no hay límite.

La estructura rectangular de un data set implica que toda variable debe tener valor o existir para todas y cada una de las observaciones.

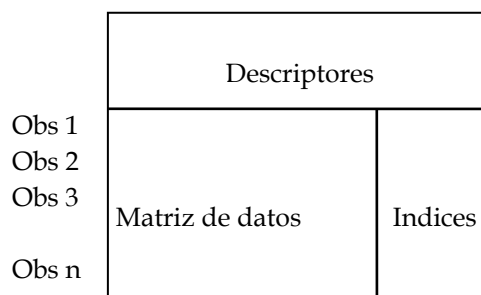


Fig.24. Estructura de un data set.

No se puede utilizar un editor de texto como el block de notas para ver o editar los “dataset” SAS (archivos de datos SAS). La extensión o tipo de archivos para los “dataset” SAS es: “.sas7bdat”.

1.5 Librería SAS

Una *librería de datos SAS* es una colección de archivos SAS que forman una unidad. En el sistema operativo Windows (basado en directorios) una librería de datos SAS es una colección de archivos SAS guardados en un directorio o carpeta específica.

Por defecto los “data set” se almacenan en la librería temporal “Work”, los cuales se borran o se pierden en el momento de cerrar una sesión de trabajo con el SAS; por lo general los almacenaremos en librerías permanentes, de esta forma los datos pueden ser utilizados en futuras aplicaciones sin necesidad de volverlos a generar.





Seleccione la carpeta “Libraries” de la Ventana “Explorer”, la barra de herramientas cambia y se ve como la mostrada en la figura 25. El icono  se utiliza para subir de nivel en el explorador, los iconos  o  para crear una nueva librería y el icono  para mostrar las librerías en forma de árbol.



Fig.25. Barra de herramientas de la carpeta Libraries.

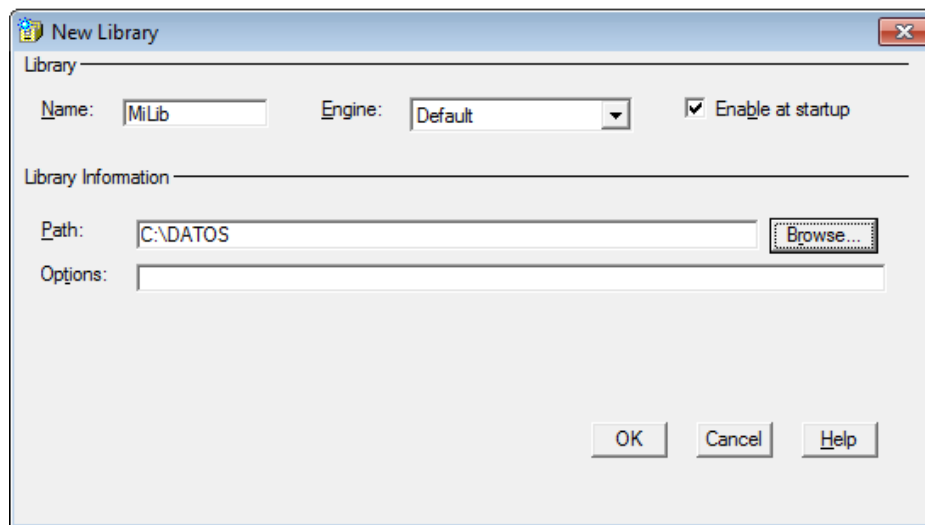



Fig.26. Cuadro de diálogo New Library.

Para crear una librería permanente pulse el botón  y se mostrará el cuadro de diálogo que se ve en la figura 26.

En el cuadro de texto “Name” introduzca el nombre de la librería (máximo 8 caracteres), en “Engine” el tipo de motor que depende del tipo de sistema operativo (Default), en “Path” coloque el camino o ruta de la carpeta o subdirectorio. Si desea que la librería esté disponible cada vez que arranque el SAS seleccione “Enable at startup”. Para finalizar y crear la librería presione el botón Ok y aparecerá la misma en la ventana “Libraries”.

1.6 Instrucciones globales del SAS

Las siguientes instrucciones se pueden usar en cualquier parte de un programa SAS.

1.6.1 ENDSAS|BYE

Causa que la sesión SAS se termine al finalizar el actual paso de datos o de procedimiento. Es el mismo efecto que seleccionar salir en el menú Archivo.

1.6.2 FILENAME

Asocia el nombre completo de un archivo externo (unidad:\camino\nombre) a un nombre o *ref-arch* (referencia de archivo). La referencia de archivo es entonces usada para referirnos o nombrar el archivo en las instrucciones de programación que accesan archivos externos (INFILE, FILE).

Formato:

```
FILENAME <ref-arch> [u:][camino]nombre.ext
```

Ejemplo para leer datos de un archivo externo:

```
FILENAME DATOS "C:\DATOS\INVIERNO";  
DATA PRUEBA;  
  INFILE DATOS;  
  INPUT X Y Z;  
  :
```

1.6.3 FOOTNOTE

Define las notas de pie de página que se imprimen en la parte inferior de las páginas. Puede definir hasta 10 líneas de pies de página.

Formato:

```
FOOTNOTE[n] ['texto']
```

1.6.4 LIBNAME

Define uno o más directorios para ser usados por los programas SAS, asocia una *ref-lib* (referencia de librería) con un nombre de directorio, el cual debe estar creado.

Formato:

```
LIBNAME <ref-lib> '[u:][camino]'
```

La asociación entre ref-lib y el directorio se mantiene hasta que se cambie por otra instrucción LIBNAME o hasta que finalice la sesión SAS.

Si se quiere leer o crear un data set permanente se crea una ref-lib y luego se nombra el archivo.

Un nombre de archivo en SAS esta formado por dos nombres SAS separados por un punto (.), el primero hace referencia a la librería y el segundo al nombre, así, si se quiere crear el data set PRUEBA dentro del directorio DATOS de su disco duro se hace:

```
LIBNAME C 'C:\DATOS';  
FILENAME F 'C:\DATOS\ANALIS.DAT';  
DATA C.PRUEBA;  
INFILE F;  
INPUT A X Z;  
      ;
```

1.6.5 OPTIONS

Esta instrucción cambian temporalmente los valores por defecto de una o más opciones del sistema SAS. Los cambios realizados por una instrucción OPTIONS tienen efecto durante la duración de la sesión SAS o hasta que sean cambiados por otra instrucción OPTIONS. Una instrucción OPTIONS puede estar en cualquier parte de un programa SAS.

Las principales opciones son:

CENTER NOCENTER	centra o no las salidas de los procedimientos a la ventana OUTPUT. Por defecto esta CENTER.
DATE NODATE	controla si la fecha actual es impresa o no en la parte superior de cada página de la ventana OUTPUT o del archivo estándar de impresión . El valor por defecto es DATE.
ERRORS =	especifica el número máximo de observaciones para el cual se imprimirá el mensaje de error. Si son detectados mas errores que el número especificado el proceso continua, pero el mensaje de error no se mostrará. Por defecto es 20.
FIRSTOBS =	especifica la primera observación a ser procesada de un data set. El valor por defecto es 1.
FORMCHAR =	especifica los caracteres de formateo de

	salida, son usados para construir las salidas tabulares de línea. El valor es una cadena de caracteres de longitud hasta 64 bytes. Los primeros 11 caracteres definen la barra vertical y horizontal, y los nueve caracteres de esquinas: superior izquierda, centro y derecha; mitad izquierda, media y derecha; e inferior izquierda, mitad y derecha. Los valores estándar de estos caracteres son --- - + ---
<code>_LAST_ =</code>	especifica el nombre del data set creado más recientemente. Después que un data set ha sido creado el valor de <code>_LAST_</code> es el nombre de ese data set.
<code>LINESIZE LS =</code>	especifica el tamaño de la línea para la ventana OUTPUT (archivo de salida estándar) usada por los procedimientos. El rango de valores permitidos es de 64 a 256, por defecto es 78.
<code>MISSING =</code>	especifica el carácter a ser usado para imprimir los datos numéricos faltantes. Por defecto es el punto (.).
<code>NOTES NONOTES</code>	especifica si se imprime o no las notas en la ventana LOG.
<code>NUMBER NONUMBER</code>	controla si se imprime o no el número de página en la primera línea de cada página impresa. Por defecto es NUMBER.
<code>OBS = n</code>	especifica la última observación que el sistema procesará de un data set, es decir, procesará las primeras n observaciones.
<code>PAGENO = n</code>	especifica el número de página de comienzo de la próxima página de salida producida por algún procedimiento.
<code>PAGESIZE PS =</code>	especifica el número de líneas por página que se imprimirán. Por defecto es 21.
<code>STIMER NOSTIMER</code>	controla si se imprime o no la cantidad de tiempo requerido para ejecutar un paso de datos de procedimiento. Por defecto es STIMER.

1.6.6 RUN

Esta instrucción causa que sean ejecutados los pasos de datos y procedimientos que previamente habían sido leídos y compilados; luego de ejecutarlos pasa a leer o entrar los siguientes para luego ejecutarlos.

1.6.7 TITLE

Es usado para especificar hasta diez líneas de título a ser impresas en cada página de salida. Puede definir hasta 10 líneas de título.

Formato:

TITLE [n] ['texto']

1.6.8 X

La instrucción X invoca a una ventana de MS-DOS, para salir de ella tipee EXIT. Si lo usa con un comando del MS-DOS ejecutará el mismo.

Formato:

X [comando]

2. El paso de Datos (DATA STEP)

[3][4] y [5]

En la figura 27 se muestra el proceso que realiza el SAS para crear los "dataset", mientras existan observaciones (CARDS o INFILE) lee una línea de datos, crea nuevas variables y escribe la observación en el "dataset".

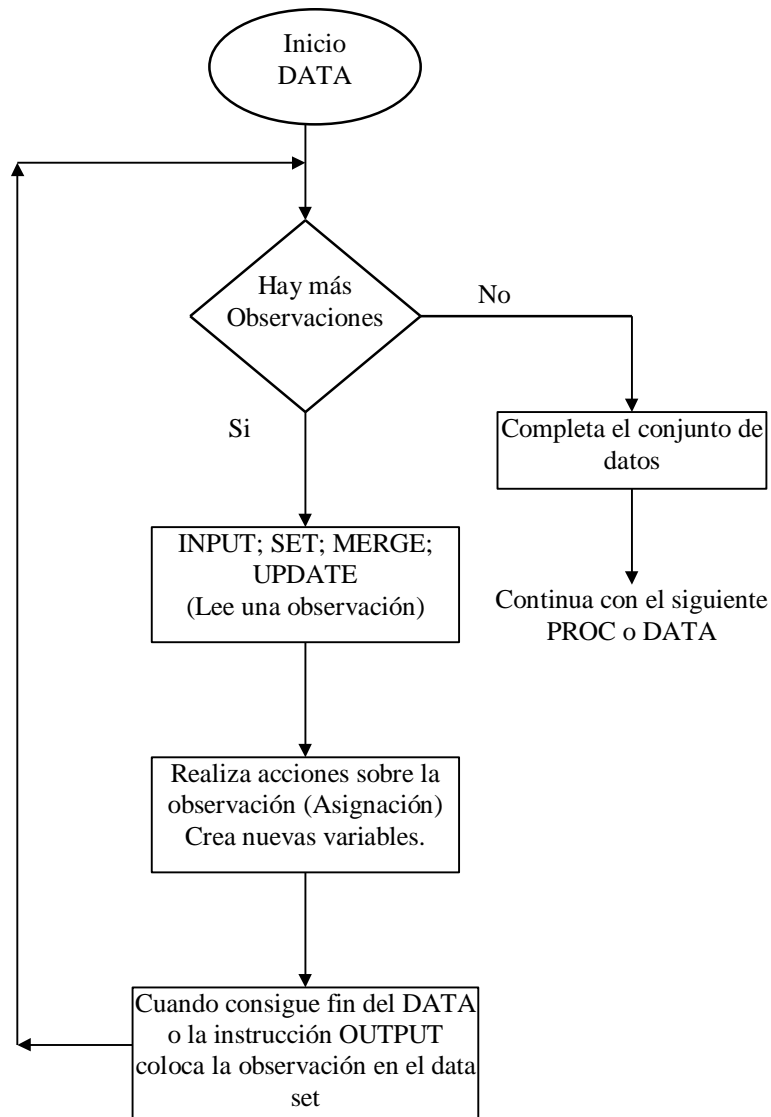


Fig. 27. Ejecución del paso de datos.

2.1 Instrucciones del DATA STEP

2.1.1 La instrucción CARDS

Indica que los datos (observaciones) que se van a leer con la instrucción INPUT vienen inmediatamente a continuación, para finalizar la entrada de datos se coloca una línea con un punto y coma (;).

```
CARDS;
V009154650 2 28 2 159 72 12
V010456016 1 22 3 177 136 12
V011370884 2 22 1 86 64 8
V010896200 1 24 1 189 121 13
;
```

2.1.2 La instrucción DATA

Se utiliza para marcar el comienzo de un paso de datos o “DATA step” y permite dar un nombre o identificador al “dataset” que se va a crear.

Formato

```
DATA [<SAS data set>] [opciones];
```

<SAS data set> es el nombre con el cual se creará y almacenará el conjunto de datos SAS.

Si este se omite el sistema SAS le asignará DATA1, DATA2,

El identificador del “dataset” (como también las variables) deben tener máximo 8 caracteres, comenzar con una letra y pueden contener de números, letras y subrayados.

Las opciones se estudiarán más adelante (sección 4.1)

2.1.3 La Instrucción FILE

Se usa para indicar la referencia a un archivo externo de salida (se usa junto con la instrucción PUT para crear un archivo de datos a partir de un “dataset SAS”).

Formato

```
FILE <ref-archivo>;
<ref-archivo>   unidad:\camino\nombreakivo.
```

2.1.4 La Instrucción INFILE

Se usa para indicar la referencia a un archivo externo de entrada que contiene los datos (observaciones).

Formato

```
INFILE <ref-archivo> <opciones>;
```

```
<ref-archivo>   unidad:\camino\nombreadarchivo.
```

```
<opciones>     una de las siguientes:
```

```
FIRSTOBS = n   indica cuál será la primera línea del archivo
                externo a ser procesada, muy útil cuando el archivo
                contiene los nombres de variables y descripción de
                los datos en las primeras líneas.
```

```
OBS = n       indica cuál será la última observación a ser
                procesada en el archivo externo.
```

En el siguiente programa los datos se leen de un archivo externo llamado `datos_programa2.dat`.

```
/* Este programa lee los datos que estaban después de CARDS utilizando
un archivo de datos externo mediante la instrucción INFILE .
Como el archivo contiene en la primera línea los nombres de variable
Lee desde la segunda línea de datos          */

/* Paso de Datos (DATA STEP)          */
DATA MiLib.EJEMPLO2;
  INFILE "C:\DATOS\datos_programa2.dat" FIRSTOBS=2;
  INPUT CI $ 1-10 SEXO 12 EDAD 14-15 ESC 17 UC 19-21 UA 23-25 TIEMPO 27-28;

PROC PRINT;
  TITLE "DATOS ORIGINALES DE LOS ALUMNOS";

PROC MEANS;
  VAR EDAD UC UA TIEMPO;
  TITLE "ESTADISTICAS BASICAS PARA LAS VARIABLES NUMERICAS";

PROC SORT;
  BY CI;

PROC PRINT;
  TITLE "DATOS DE LOS ALUMNOS ORDENADOS POR CEDULA";

RUN;
```

2.1.5 La instrucción INPUT

Permite la lectura o entrada de líneas de datos identificados con la instrucción `CARDS` o `INFILE` y a su vez asigna nombres a las variables que corresponden a cada uno de los campos.

Tiene tres modos de usarse: en modo columna, en modo de lista o modo formateada.

2.1.6 La instrucción PUT

Permite escribir datos desde un “dataset SAS” existente en un archivo de texto identificado por la instrucción FILE en formato de columna. Para evitar que se duplique el “dataset SAS” existente se utiliza DATA _NULL_;

Ejemplo:

```
DATA _NULL_;
  FILE "C:\DATOS\alumnos.dat";
  SET MiLib.Alumnos;
  PUT CI $ 1-10 SEXO 12 EDAD 14-15 ESC 17 UC 19-21 UA 23-25;
```

```
RUN;
```

2.2 Entrada de datos en modo columna

La lectura de cada variable se realiza indicando la columna donde comienza y finaliza el dato en la línea de entrada (sea CARDS o INFILE), tal y como se indica en el programa 1 de la página 20.

Formato

```
INPUT <variable1> [$] colcom-colfin [.d] <variable2> [$]
colcom-colfin [.d] ...
```

donde el símbolo \$ es opcional y significa que esa variable es alfanumérica y .d representa el número de decimales que tomará en caso de que sea numérico.

- Esta forma de lectura es apropiada cuando tenemos cada dato tabulado en la misma columna.
- Los datos faltantes se introducen dejando en blanco las columnas que corresponda en la observación.
- Se pueden leer datos carácter que contenga espacios en blanco en el medio
- Un campo o parte de él puede ser leído varias veces.

```
INPUT NOMBRE $ 1-15 APELLIDO $ 16-30 INICIAL $ 1;
CARDS;
FELIX ADALBERTOGONZALEZ PEREIRA
PEDRO LUIS      GUERRA DIAZ
:
```

- Los datos pueden ser leídos en cualquier orden y se pueden omitir algunos de ellos.

```
INPUT ESC 17 SEXO 12 EDAD 14-15 TIEMPO 27-28;
CARDS;
V009154650 2 28 2 159 72 12
V010456016 1 22 3 177 136 12
V011370884 2 22 1 86 64 8
```

- Los datos carácter son alineados a la izquierda. El número de bytes usados es determinado por el número de columnas, máximo 200. En el siguiente ejemplo el tamaño de la variable sexo es 3.

```
INPUT SEXO $ 1-3
```

1	2	3	4
M			
	M		
		M	

```
'M '
'M '
'M '
```

- Los valores numéricos pueden estar en cualquier parte del campo especificado por las columnas de comienzo y fin. Puede llevar signo, punto decimal o exponente. Los espacios en blanco no son permitidos en el medio de un campo numérico.

```
INPUT X 1-6;
```

1	2	3	4	5	6	7
				2	3	
2	3					
		2	3			
2	3	.	0			
	-	2	3			
-	2	3				
	2	3		0		
-		2	3			

```
23
23
23
23
-23
-23
NULO
NULO
```

- Con especificación de decimales

```
INPUT X 1-6 .1;
```

1	2	3	4	5	6	7
				2	3	
2	3	0				
		2	3			
2	3	.	5	0	6	
	-	2	3			

```
2.3
23.0
2.3
23.506
23.0
```

2.3 Entrada de datos en modo lista

La instrucción INPUT busca uno o más espacios en blanco hasta encontrar el primer carácter, entonces lee de la línea todos los caracteres hasta el próximo blanco o fin de línea.

Formato

```
INPUT <variable1> [$]<variable1> [$]<variable1> [$] ...
```

Donde el símbolo \$ es opcional y representa que ese campo es alfanumérico.

- Los campos deben ser especificados en el orden en que están en la línea de datos.
- Los campos en la línea de datos se separan con uno o más blancos.
- No se pueden leer a los caracteres con blancos incluidos.
- Longitud máxima de un dato carácter es 8.
- Los datos faltantes se indican con un punto.

```
INPUT CI $ NOMBRE $ SEXO EDAD ESC UC UA TIEMPO;  
CARDS;  
9154650 PEDRO 2 28 2 159 72 12  
10456016 MARÍA 1 22 3 177 136 12  
11370884 ANA 2 . 1 86 64 8  
10896200 SINFOROSA 1 24 1 189 121 13
```

Obsérvese que la última línea de datos el nombre será truncado a 8 caracteres. En la tercera línea de datos la edad se introduce como dato faltante.

2.4 Entrada de datos con formato

Se especifica la posición de comienzo y el tamaño del campo (como en Fortran y otros lenguajes). Mueve el apuntador a la posición inicial del campo, luego se especifica la variable y su formato de entrada (INFORMAT).

Formato:

```
INPUT <control apuntador> <variable> <formato de entrada>
```

<control apuntador>	Descripción
@n (absoluto)	Mueve el apuntador de columna a la posición n
+n (relativo)	Mueve el apuntador de columna n columnas de la posición actual

Tabla 1. Control apuntador de columna

<formato de entrada>	Descripción	Ancho	Decimal
w.	Numérico estándar	1-32	
w.d	Numérico estándar con decimal	1-32	1-31
\$w.	Carácter estándar	1-200	
\$CHARw.	Carácter estándar con blancos	1-200	
HEXw.	Hexadecimal numérico	1-16	
\$HEXw.	Hexadecimal carácter	1-16	
COMMAw.d	Numérico real con comas	1-32	0-31
Ew.	Notación científica	7-32	0-31
BZw.d	Blancos por ceros	1-32	1-31
DATEw.	Fecha de la forma DDMMYY	7-32	
DDMMYYw.	Fecha de la forma DDMMYY	6-32	
JULIANw.	Fecha juliana YYDDD o YYYYDDD	5-32	
MMDDYYw.	Fecha de la forma MMDDYY	6-32	
YYMMDDw.	Fecha de la forma YYMMDD	6-32	

Tabla 1. Formatos de entrada (INFORMAT) mas usados.

2.5 Listas de formatos de entrada y Nombre de variables numerada

Consideremos el siguiente ejemplo para leer el año y los niveles pluviométricos promedio de los doce meses del año:

```
DATA PRECIPI;
  INFILE 'C:\DATOS\DATOS\GUARENAS.DAT';
  INPUT ANNO 4. ENE 8.1 FEB 8.1 MAR 8.1 ABR 8.1 MAY 8.1 JUN 8.1
        JUL 8.1 AGO 8.1 SEP 8.1 OCT 8.1 NOV 8.1 DIC 8.1;
PROC PRINT;
RUN;
```

La instrucción INPUT anterior puede ser escrita de varias formas especiales:

- Variables y formatos de entrada pueden ser agrupados

```
INPUT ANNO 4. (ENE FEB MAR ABR MAY JUN JUL AGO SEP OCT NOV DIC)
(8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1 8.1);
```

- El formato de entrada es reciclado para satisfacer la lista de variables:

```
INPUT ANNO 4. (ENE FEB MAR ABR MAY JUN JUL AGO SEP OCT NOV DIC) (8.1);
```

- Para nombrar algunas variables relacionadas se pueden utilizar nombres de variables numeradas:

```
INPUT ANNO 4. (MES1 MES2 MES3 MES4 MES5 MES6 MES7 MES8 MES9 MES10 MES11
MES12) (8.1);
```

puede ser escrito de la siguiente manera:

```
INPUT ANNO 4. (MES1-MES12) (8.1);
```

2.6 Lectura de registros con múltiples líneas

En caso de que una observación este contenida en varias líneas de la entrada (CARDS o INFILE) se puede utilizar tres procesos diferentes:

- Utilizando varias instrucciones INPUT, en este caso cada línea de la entrada de datos forma un buffer de entrada. Veamos el siguiente ejemplo en el que en la primera línea esta la cédula de identidad (1-10), el sexo (12) y la escuela (15); en la segunda esta la edad (4-5) y el tiempo en la facultad((8-9); y en la tercera esta las unidades cursadas (1-3) y las unidades aprobadas (6-8)

```
DATA EJEMPLO;
  INPUT CI $ 1-10 SEXO 12 ESC 15;
  INPUT EDAD 4-5 TIEMPO 8-9;
  INPUT UC 1-3 UA 6-8;
CARDS;
V009154650 2 2
  28 12
159 72
V010456016 1 3
  22 12
177 136
V011370884 2 1
  22 8
  86 64
  :
```

- Utilizando el control apuntador / (relativo). Cada vez que un / es encontrado en una instrucción *INPUT*, una nueva línea de entrada es cargada en el buffer de entrada y coloca el apuntador de columna en la posición 1. Este apuntador puede ser utilizado en cualquier modo de lectura.

```
DATA EJEMPLO;
  INPUT CI $ 1-10 SEXO 12 ESC 15 / EDAD 4-5 TIEMPO 8-9 / UC 1-3 UA 6-8;
CARDS;
V009154650 2 2
      28 12
159    72
      :
```

- Utilizando el control apuntador # (absoluto). Este es un apuntador a la enésima línea de entrada, es decir, que cuando este apuntador es usado se avanza hasta la primera columna de la enésima de entrada.

```
DATA EJEMPLO;
  INPUT CI $ 1-10 SEXO 12 ESC 15 #3 UC 1-3 UA 6-8;
CARDS;
V009154650 2 2
      28 12
159    72
      :
```

En el ejemplo se lee de la línea 1 la cédula, el sexo y la escuela, y de la tercera las unidades cursadas y las unidades aprobadas.

El valor más grande que acompaña al apuntador # define el número de líneas en el buffer de entrada. Este apuntador puede ser utilizado en cualquier modo de lectura.

Los registros en múltiples líneas pueden ser leídos en cualquier orden.

```
INPUT #2 EDAD (4-5) #1 CI $ 1-10 ESC 15 #3 UC 1-3 UA 6-8;
```


2.7 Leyendo varios registros por línea de entrada

Consideremos el siguiente ejemplo, en el cual una línea de datos contiene la información del número de llamadas a una central telefónica en 4 días continuos

```
DATA VARIAS;  
  INPUT FECHA DDMMYY6. LLAMADA @@;  
CARDS;  
010198      1897 020198      2897 030198      987 040198      1452  
050198      1113 060198      1807 070198      1071 040198      959  
:  
PROC PRINT;  
RUN;
```

Obsérvese en la salida como fue almacenada la variable fecha. @@ es utilizado para leer múltiples registro por cada línea de la entrada de datos, es decir, especifica que debe mantenerse la lectura en la misma línea de datos hasta llegar a fin de línea. No debe ser usado con la lectura en modo columna.

3. El paso de Procedimientos

[4] y [5]

La instrucción PROC se usa para invocar un procedimiento SAS. Los procedimientos SAS son programas o módulos independientes que:

- Leen data set
- Calculan estadísticas
- Realizan análisis estadístico
- Imprimen resultados
- Crean nuevos data set

Formato general de la instrucción PROC:

```
PROC programa [Opciones];
    [instrucción general;]
    [instrucción general;]
    :
```

3.1 Instrucciones generales usadas en el paso de procedimientos

3.1.1 Instrucción BY

Permite procesar por grupos, es decir, procesa por separado cada grupo que se forma con los valores de las variables. Debe ser previamente ordenado el data set por las variables de interés.

Formato:

BY <lista de variables>

Ejemplo: Calcula las estadísticas básicas para masculino y para femenino por separado

```
PROC MEANS DATA=C.ALUMNOS;
BY SEXO;
```

3.1.2 Instrucción FOOTNOTE

Define el texto a ser impreso en la parte inferior de cada página en la salida (ventana OUTPUT). Se puede especificar hasta 10 líneas de pie de página.

Formato:

FOOTNOTE n “pie de página”

Ejemplo:

```
PROC MEANS DATA=C.ALUMNOS;  
FOOTNOTE1 "REALIZADO POR: ANGEL ZAMBRANO";  
FOOTNOTE2 "Escuela de Estadística";
```

3.1.3 Instrucción TITLE

Define el texto a ser impreso en la parte superior de cada página en la salida (ventana OUTPUT). Se puede especificar hasta 10 líneas de título.

Formato:

TITLE n “título”

Ejemplo:

```
PROC MEANS DATA=C.ALUMNOS;  
TITLE1 "ESTADISTICAS BASICAS";  
TITLE2 "PROYECTO ESTUDIANTIL";
```

3.1.4 Instrucción VARIABLES o VAR

especifica las variables del data set a ser procesadas por el procedimiento.

Formato:

VARIABLES <lista de variables>

o

VAR <lista de variable>

Ejemplo:

```
PROC MEANS DATA=C.ALUMNOS;  
VAR EDAD UA UC TIEMPO;
```

3.2 Procedimientos del SAS Base

3.2.1 Procedimiento DBF

Convierte archivos de formato dBase II y dBase III a conjunto de datos SAS y viceversa.

Formato:

PROC DBF opciones;

opciones:

DB2= | DB3= especifica una referencia a un archivo dBase II o dBase III.

DATA= | OUT= Si se desea convertir de dBase a SAS especifique el nombre del archivo SAS a crear en OUT= . Si se desea convertir de SAS a dBase especifique el nombre del archivo SAS que contiene las observaciones en DATA= .

Ejemplo: convierte el "dataset" ALUMNOS a un archivo dbase III "datos.dbf"

```
PROC DBF DB3="C:\DATOS\datos.dbf" DATA=C.ALUMNOS;
```

3.2.2 El procedimiento CHART

Produce gráficos de barras verticales y horizontales (histograma), gráficos de bloques, de torta y de estrella. Estos gráficos son construidos con caracteres y por lo tanto no tienen calidad gráfica.

Formato:

PROC CHART opciones;

<instrucciones generales>

opciones

DATA = data set selecciona el data set del cual se producirá el gráfico.

LPI = p especifica las proporciones del gráfico de torta y estrella.

LPI = (líneas por pulgada / columnas por pulgada)*10 por defecto es igual a 6.

FORMCHAR define los caracteres que usaran para la construcción de los ejes vertical y horizontal, la esquina inferior izquierda y los símbolos usados para crear las barras, secciones o bloques en el gráfico.

<Instrucciones generales>

BY <Variables>

VBAR <Variables>/opciones produce barras verticales

HVAR <Variables>/opciones produce barras horizontales

BLOCK <Variables>/opciones produce gráfico de bloques

PIE <Variables>/opciones produce gráfico de torta

STAR<Variables>/opciones produce gráfico de estrella.

Opciones de las instrucciones generales

TYPE = código determina los valores que se representara por las longitudes de las barras (FREQ CFREQ PCT CPCT SUM MEAN)

SUMVAR = variable suma los valores de variable

GROUP = variable agrupa por una variable

DISCRETE agrupa las observaciones en forma discreta

MIDPOINTS = lista de valores define los puntos medios de las barras

ASCENDING/DESCENDING ordena en forma ascendente o descendente

Ejemplo:

```
PROC CHART DATA=C.EJEMPLO;
  FORMAT ESC FESC. SEXO FSEXO.;
  VBAR ESC;
  VBAR ESC/DISCRETE TYPE=PCT;
  VBAR ESC/DISCRETE GROUP=SEXO;
  HBAR TIEMPO;
  HBAR TIEMPO /MIDPOINTS = 9 11 13;
  BLOCK ESC;
  BLOCK ESC/DISCRETE GROUP=SEXO;
  PIE ESC;
  PIE ESC/DISCRETE;
```

3.2.3 El procedimiento CONTENTS

Imprime los descriptores de un data set.

Formato:

PROC CONTENTS opciones;

opciones

DATA = data set selecciona el data set del cual se imprimirá el área de descriptores.

OUT = data set crea un nuevo data set como salida

NOPRINT	Suprime la impresión de la salida del procedimiento.
POSITION	Imprime una segunda lista de las variables en el orden de la posición que ocupan en el data set. La primera la imprime en orden alfabético.
SHORT	Imprime sólo el nombre de las variables en la lista de variables.
DETAILS	
NODETAILS	Imprime o no imprime los detalles adicionales de las variables, por defecto los imprime.

3.2.4 Procedimiento CORR

Calcula coeficientes de correlación entre dos o más variables, incluyendo correlaciones de Pearson, ponderadas.

Formato general:

```
PROC CORR opciones;
    <instrucciones generales>;
```

Opciones:

DATA = data set	Selecciona el data set que se le calculará las correlaciones.
OUTH=	Crea un conjunto de datos SAS que contiene los estadísticos de Hoeffding. Requiere que esté activada la opción HOEFFDING.
OUTK=	crea un conjunto de datos SAS que contiene las correlaciones de Kendall. Requiere que esté activada la opción KENDALL.
OUTP=	Crea un conjunto de datos SAS que contiene los estadísticos de Person. Este conjunto de datos es de tipo CORR e incluye medias, desviación estándar, número de observaciones, y los coeficientes de correlación. Requiere que esté activada la opción PEARSON.
OUTS=	Crea un conjunto de datos SAS que contiene los estadísticos de Spearman. Requiere que esté activada la opción SPEARMAN.
HOEFFDING	Se calcula el estadístico D de Hoeffding.
KENDALL	Se calculan los coeficientes tao-b de Kendall.
PEARSON	Se calculan las correlaciones de Pearson.
SPEARMAN	Se calculan las correlaciones de Spearman.
NOMISS	Especifica que una observación con valor

faltante para alguna variable será extraída del análisis.

VARDEF= especifica el divisor a ser usado para calcular la varianza, los valores que toma son DF si el divisor va a ser los grados de libertad, WEIGHT usa la suma de los pesos, N el numero de observaciones o WDF la suma de los pesos menos 1 será usado. El valor por defecto es DF.

3.2.5 El procedimiento FORMAT

Es usado para crear formatos, tanto de entrada de datos como de presentación o salida de datos. Estos pueden ser usados en un paso de datos o en un paso de procedimientos.

Formato:

```
PROC FORMAT opciones;
    <instrucciones generales>
```

instrucciones generales para el procedimiento FORMAT

```
VALUE <nombre de formato> (opciones)
```

```
    rango1 = 'etiqueta1'
    rango2 = 'etiqueta2'
    :
    : ;
```

```
PICTURE <nombre de formato> (opciones)
```

```
    rango1 = 'descripción1'
    rango2 = 'descripción2'
    :
    : ;
```

Las opciones para VALUE son: DEFAULT=n MAX=n y MIN=n que especifican los tamaños por defecto, máximo y mínimo; si no se especifica DEFAULT toma el tamaño del valor mayor; el valor más grande permitido para MAX es 40.

Para la instrucción PICTURE además de las opciones de tamaño se puede especificar ROUND para decirle que redondee el número antes de formatearlo.

Los nombres de formato además de ser nombres SAS no pueden finalizar con un dígito y en caso de la instrucción VALUE debe comenzar con el signo \$ para variables carácter.

La descripción en la instrucción PICTURE es válida sólo para variables numéricas, se refiere a una secuencia de caracteres encerrados entre apóstrofes que especifica como un número será formateado, están restringidos a un máximo de 24 caracteres.

Ejemplo: en el programa del ejemplo 1 (EJEM 1.SAS)

```
PROC FORMAT;
  VALUE FSEXO 1 = 'M' 2 = 'F';
  VALUE FESC 1='ADM' 2='CON' 3='ECO' 4='EST';
  VALUE FEDAD LOW-25 = 'MENOR O IGUAL A 25' 26-HIGH = 'MAYOR DE 25';
  PICTURE FTIEMPO LOW-HIGH = '09 SEM';

PROC PRINT;
  FORMAT SEXO FSEXO. EDAD FEDAD. TIEMPO FTIEMPO.;
```

En el procedimiento PRINT, FORMAT funciona como una instrucción general del procedimiento, el identificador del formato debe terminar con un punto.

Se usa esta instrucción para cambiar el formato de presentación de las variables, en el ejemplo la variable SEXO se imprimirá una **M** para el valor 1 y una **F** para el valor 2, para la variable EDAD se imprimirá **MENOR O IGUAL A 25** o **MAYOR DE 25** y para la variable TIEMPO se imprimirá un cero al comienzo si es el número es del 0 al 9, luego un dígito y después la palabra SEM.

3.2.6 El procedimiento FREQ

Cuenta valores y genera tablas de frecuencias simples, dobles o más entradas.

Formato:

```
PROC FREQ opciones;
  <instrucciones generales>
```

opciones

DATA = data set selecciona el data set que se ordenará

Instrucciones generales del procedimiento FREQ.

```
TABLES <var1> * <var2> * ... /opciones;
```

opciones	EXPECTED	NOFREQ	NOCUM	ALL
	DEVIATION	NOPERCENT	MISSING	NOROW
	NOCOL	CELLCHI2	CHISQ	NOPRINT
	OUT=SAS data set			

Ejemplo:

```
PROC FREQ DATA = C.EJEMPLO;
  TABLES SEXO;
  TABLES SEXO*ESC;
  TABLES SEXO*ESC/ALL;
  TABLES SEXO*ESC/NOCOL NOROW;
```


3.2.7 El procedimiento MEANS

Produce estadísticas descriptivas univariantes simples para variables numéricas.

Formato:

```
PROC MEANS opciones;
    <instrucciones generales>
```

opciones

DATA = data set selecciona el data set que contiene las variables.

NOPRINT no imprime la salida

MAXDEC=n usa n cifras decimales para imprimir la salida.

<estadísticos> uno o más de los siguientes:

N	RANGE	CV	MAX
NMISS	SUM	CSS	MEAN
SKEWNESS	VAR	KURTOSIS	STDERR
STD	USS	T	MIN

Si no se especifica se calcula N MEAN STD MAX MIN

Instrucciones generales del procedimiento MEANS.

BY variables;

VAR variables;

OUTPUT OUT = SAS data set;

Ejemplo

```
PROC MEANS DATA=C.EJEMPLO;
```

```
PROC MEANS DATA=C.EJEMPLO N MEAN VAR STD SKEWNESS KURTOSIS;
```

```
    BY SEXO;
```

```
    VAR UA UC;
```

3.2.8 El procedimiento PLOT

Construye un gráfico XY (gráficos de dispersión)

Formato:

```
PROC PLOT <opciones>;
    <instrucciones generales>
```

<opciones>

DATA = "data set" selecciona el data set del cual se obtendrá el gráfico.

Instrucciones generales del procedimiento PLOT.

BY variable;

PLOT <variable X> * <variable Y> [="carácter"] / opciones;

Opciones

VAXIS = valores	coloca valores en el eje vertical
HAXIS = valores	coloca valores en el eje horizontal
VREF = valores	coloca líneas de referencias verticales
HREF = valores	coloca líneas de referencias horizontales
OVERLAY	superpone dos o más gráficos en uno

Ejemplo:

```
PROC PLOT DATA=C.EJEMPLO;
    PLOT UC*UA;
    PLOT UC*UA = "*";
    PLOT UC*UA = "*" /HREF=50 100 150 200 VREF=100 200 300;
    PLOT UC*UA = "*" /VAXIS = 0 TO 300 BY 50 HAXIS = 40 80 120 160;
```

3.2.9 El procedimiento PRINT

Imprime las observaciones de un "data set" en la ventana OUTPUT.

Formato:

```
PROC PRINT opciones;
    <instrucciones generales>
```

opciones

DATA = data set	selecciona el "data set" que se imprimirá (si no se especifica, imprime el último que se usó)
DOUBLE D	imprime a doble espacio
LABEL	imprime las etiquetas de variables (si se definieron)
NOOBS	no imprime el número de la observación
N	muestra el número de observaciones que se

imprimieron del "data set" o al llegar al final de cada grupo si se utiliza la instrucción BY.

Además de las instrucciones generales VAR, BY, TITLEn y FOOTNOTE n se puede utilizar:

SUM <lista de variables>, para totalizar las columnas o variables especificadas.

3.2.10 El procedimiento SORT

Ordena las observaciones de un "data set" por una o más variables.

Formato:

```
PROC SORT opciones;
    <instrucciones generales>
```

opciones

DATA = data set selecciona el "data set" que se ordenará
 OUT = data set crea un nuevo data set como salida (si no se especifica se reemplaza el original o fuente)

instrucción general

```
BY [DESCENDING] var1    [DESCENDING] var2 ...;
```

Se ordena por los valores de var1, si hay registros u observaciones coincidentes éstos se ordenan por los valores de var2. Por defecto se ordena en forma ascendente (de menor a mayor), si se desea ordenar en forma descendente se debe especificar la opción DESCENDING.

El procedimiento UNIVARIATE

Produce estadísticas descriptivas básicas para variables numéricas, con más detalles que otros procedimientos que calculan estadísticas básicas.

Formato:

```
PROC UNIVARIATE opciones;
    <instrucciones generales>
```

opciones

DATA = data set selecciona el data set que se ordenará
 NOPRINT no imprime la salida.
 PLOT produce un gráfico de tallo y hojas, de caja y un probabilidad normal

FREQ	produce una tabla de frecuencias con los valores de la variable, frecuencias, porcentajes y porcentajes acumulados.
NORMAL	calcula una prueba estadística para la hipótesis los datos siguen una distribución normal.
VARDEF=	especifica el divisor a ser usado para calcular la varianza, los valores que toma son DF si el divisor va a ser los grados de libertad, WEIGHT usa la suma de los pesos, N el número de observaciones o WDF la suma de los pesos menos 1 será usado. El valor por defecto es DF.

Instrucciones generales del procedimiento UNIVARIATE.

VAR variables;

BY variables;

WEIGHT variable; Usa los valores de la variable, w_i , para calcular una media y varianza ponderada, el divisor de la varianza depende de la opción VARDEF. Si el valor de la variable WEIGHT es menor que cero entonces se asume el valor cero.

ID variables; Se usa los primeros ocho caracteres de la variable para identificar la observación.

OUTPUT OUT = data set estadístico = nomvar ...

Para guardar un data set con una observación y tantas variables como estadísticos escoja, nomvar es el nombre de la variable bajo el cual se almacenará el estadístico en el data sets. Estos son: N (número de observaciones con las cuales los cálculos fueron hechos), NMISS (número de observaciones faltantes), NOBS (número de observaciones), MEAN (media), SUM (suma de los elementos), STD (desviación estándar), VAR (varianza), SKEWNESS (asimetría), KURTOSIS (kurtosis), MAX (máximo), MIN (mínimo), RANGE (rango), Q3 (cuartil superior o percentil 75), MEDIAN (mediana), Q1 (cuartil inferior o percentil 25), QRANGE (rango intercuartílico), P1, P5, P10, P90, P95, P99 (percentil 1, 5, 10, 90, 95, 99), NORMAL (prueba estadística de normalidad, si el tamaño es menor o igual a 2000 es el estadística Shapiro-Wilk en caso contrario es el de Kolmogorov)

El procedimiento UNIVARIATE provee grandes detalles sobre la distribución de una variable, tales como: valores extremos, cuantiles, percentiles, mediana, test de normalidad entre otros.

Entre los items que el procedimiento UNIVARIATE calcula e imprime están: nombre y etiqueta de la variable, el número de observaciones sobre el cual se hizo los cálculos, Suma de los pesos, media, suma, varianza, desviación estándar, asimetría, kurtosis, suma de cuadrados correlacionada, suma de cuadrados no correlacionada, coeficiente d correlación, el error estándar de la media, el valor del estadístico para la prueba de la hipótesis que la media poblacional es cero, $\text{Prob} > |T|$, máximo, mínimo, cuartil superior e inferior, mediana, percentiles 1, 5, 10, 90, 95 y 99, rango intercuartílico.

Ejemplo:

```
PROC UNIVARIATE DATA=C.EJEMPLO PLOT NORMAL;
  VAR UC UA EDAD TIEMPO;
  OUTPUT OUT = C.ESTADIS MEAN=MEDIA STD=DESV VAR=VARI SKEWNESS=ASIM
    KURTOSIS=KURTOSIS MEDIAN=MEDIANA P95=PERC95;
RUN;
```

Produce estadísticas básicas para las variables UC, UA, EDAD y TIEMPO; produce gráfico de tallo y hojas y de normalidad; calcula el estadístico para la prueba de hipótesis de que los datos siguen distribución Normal. Crea un data set con una observación con las variables MEDIA, DESV, VARI, ASIM, KURTOSIS, MEDIANA Y PERC95.

3.3 Otros procedimientos del SAS BASE

- **APPEND:** Añade observaciones de un conjunto de datos al final de otro conjunto de datos.
- **CALENDAR:** Muestra observaciones de un conjunto de datos en formato de mes por mes.
- **CATALOG:** Maneja entradas en un catalogo SAS.
- **COMPARE:** Compara los valores de las variables de dos conjunto de datos y reporta las diferencias encontradas.
- **DATASETS:** Manipula, es decir, lista, copia, renombra, inserta y borra conjunto de datos SAS y catálogos en una librería SAS.
- **DIF:** Convierte datos de formato DIF (formato de intercambio de datos) a formato SAS y viceversa.
- **ESTÁNDAR:** estandariza alguna o todas las variables de un conjunto de datos dada la media y la desviación estándar.
- **SUMMARY:** Calcula estadísticas descriptivas sobre variables numéricas de un

conjunto de datos SAS.

- **TABULATE:** Construye tablas de frecuencias de hasta tres niveles: columna, fila y página.
- **TRANSPPOSE:** Transpone un conjunto de datos SAS, es decir, convierte observaciones en variables y variables en observaciones.

Estadístico	MEANS	UNIVARIATE	SUMMARY	TABULATE	CORR
Número de valores faltantes	X	X	X	X	
Número de valores válidos	X	X	X	X	X
Número de valores	X		X		
Suma de ponderaciones	X	X	X	X	X
Media	X	X	X	X	X
Suma de valores	X	X	X	X	X
Mínimo	X	X	X	X	X
Máximo	X	X	X	X	X
Rango	X	X	X	X	
Suma corregida de cuadrados	X	X	X	X	X
Suma no corregida de cuadrados	X	X	X	X	X
Varianza	X	X	X	X	X
Desviación estándar	X	X	X	X	X
Error estándar	X	X	X	X	
Coefficiente de variación	X	X	X	X	
Asimetría		X			
Kurtosis		X			
t	X	X	X	X	
prob > t	X	X	X	X	
Mediana		X			
Cartiles		X			
Moda		X			
Correlación de Pearson					X
Imprime salida	X	X	X	X	X
Crea conjunto de datos.	X	X	X		X

Tabla 3. Resumen de las características de los procedimientos de estadísticas básicas..

4. Programando el Paso de Datos

[3] y [5]

4.1 Opciones del SAS data set

Formato de la instrucción DATA

```
DATA <nombre SAS> (opciones);
[INFILE ... ]
INPUT <variables> .....;
[<instrucciones del SAS data set>]
      :
[CARDS
      :
; ]
```

Las opciones se refieren a acciones específicas que son aplicables sólo al procesamiento de los datos. Estas son:

- DROP = variables

Causa que las variables especificadas sean omitidas del data set. Funciona de idéntica forma que la instrucción DROP del data set. En un paso de datos sólo se puede usar o la opción DROP o la KEEP, pero no ambas. Ejemplo:

```
DATA C.PROMEDIO (DROP=NOTA1 NOTA2 NOTA3);
INPUT CI NOTA1 NOTA2 NOTA3;
PROM=(NOTA1+NOTA2+NOTA3)/3;
CARDS
      :
```

En el ejemplo las variables NOTA1, NOTA2 y NOTA3 no serán almacenadas en el data set PROMEDIO.

- FIRSTOBS = n

Causa que el procesamiento comience en la enésima observación. Esta opción sólo es válida cuando se está leyendo un data set ya existente.

Ejemplo:

```
PROC PRINT DATA=ESTUDIO (FIRSTOBS=10);
```

- KEEP = variables

Sólo las variables especificadas son retenidas, es decir, almacenadas en el data set. En un paso de datos sólo se puede usar o la opción DROP o la KEEP, pero no ambas.

En el ejemplo anterior para indicar lo mismo se puede hacer:

```
DATA C.PROMEDIO (KEEP=CI PROM);
```

- LABEL = 'etiqueta'

Especifica una etiqueta para el data set, la cual es almacenada en la parte de los descriptores del data set y puede ser impresa con el procedimiento CONTENTS. La etiqueta consiste de hasta 40 caracteres encerrados entre apóstrofes.

Ejemplo:

```
DATA C.PROMEDIO (KEEP=CI PROM LABEL='PROMEDIO DE LA SECCIÓN A');
```

- OBS = n

Especifica la última observación que será procesada. Al igual que FIRSTOBS ésta opción es válida sólo para data set ya existente, es decir, con las instrucciones SET, MERGE y UPDATE, o cuando se invoca para ser el leído desde un procedimiento.

- RENAME = (nombreviejo = nombre nuevo ...)
cambia el nombre de una o más variables

```
DATA C.PROMEDIO (KEEP=CI PROM RENAME=(CI=CEDULA));
```

- TYPE = DATA | CORR

Especifica el tipo de data set para la entrada. Si se especifica CORR los datos representa una matriz de correlaciones que se usa como entrada para ciertos procedimientos del modulo STAT.

4.2 Instrucción de asignación

La instrucción de asignación es usada para crear nuevas variables y modificar los valores de las ya existentes. Cuando una variable es definida por una asignación es incluida en el vector de datos.

Formato:

```
<variable>=<expresión>;
```


La expresión es evaluada y el valor obtenido es almacenado en la variable.

Ejemplo: leer tres variables (AÑO, INGRESO y GASTO), crear un “data set” llamado BENEf. Crear una nueva variable llamada GANANCIA y convertir AÑO a 4 dígitos.

```
DATA C.BENEf;
  INPUT ANNO INGRESO GASTO;
  GANANCIA=INGRESO-GASTO;
  ANNO=ANNO+1900;
CARDS;
80 10922 7654
81 16099 10988
82 19999 13000
  :
```

Una expresión es un conjunto de operadores y operandos mediante los cuales se realizan las diferentes operaciones que se desean. Los operandos son las variables y/o constantes sobre los cuales se aplican los operadores.

Un operador es un carácter especial (o combinación de caracteres) que representan las operaciones.

Las expresiones se usan para crear nuevas variables, transformar las mismas, realizar comparaciones (condiciones), calcular nuevos valores, etc.

Una constante es un número o una cadena de caracteres entre comillas o apóstrofes que representan valores fijos, son llamadas también literales.

Los operadores pueden ser de tres tipos: aritméticos, de relación o lógicos. El tipo de los operadores definen el tipo de expresión.

Aritméticos		De relación		Lógicos	
**	potencia	= EQ	igual a	& AND	y lógico
*	multiplicación	^= NE	no igual a	OR	o lógico
+	suma	> GT	mayor que	^ NOT	no lógico
-	resta	< LT	menor que	Otros operadores	
-	cambio de signo	>= GE	mayor o igual a	><	mínimo
		<= LE	menor o igual a	<>	máximo
		IN	igual a uno de la lista		concatenación

Tabla 4. Operadores del SAS.

Una función SAS es una rutina que regresa un valor, la cual puede tener cero o más argumentos o parámetros. Toda función SAS tiene un único identificador o nombre SAS con el cual identificarla e invocarla. Para invocar una función colocamos su nombre y la lista de argumentos entre paréntesis: nombreFunción(arg1, arg2, ...)

Listado de algunas funciones SAS (ver SAS Lenguaje Guide)

Funciones aritméticas						
ABS	DIM	MAX	MIN	MOD	SIGN	SQRT
Funciones de truncamiento						
CEIL	FLOOR	INT	ROUND	TRUNC		
Funciones matemáticas						
EXP	GAMMA	LOG	LOG2	LOG10	ARCOS	ARSIN
ATAN	COS	COSH	SIN	SINH	TAN	TANH
Funciones de probabilidad						
POISSON	PROBBETA	PROBCHI	PROBF	PROBNORM	PROBT	
Funciones de estadísticos muestrales						
CSS	CV	KURTOSIS	MAX	MIN	MEAN	N
NMISS	RANGE	SKEWNESS	STD	SUM	USS	VAR
Funciones de números aleatorios						
NORMAL	RANBIN	RANCAU	RANEXP	RANGAM	RANNOR	RANPOI
RANTRI	RANUNI	UNIFORM				
Funciones para manejo de caracteres						
COMPBL	COMPRESS	DQUOTE	INDEX	INDEXC	LEFT	RIGHT
REVERSE	REPEAT	LENGTH	LOWCASE	UPERCASE	SCAM	TRIM
TRIMN	SUBSTR	BYTE	RANK			
Funciones de hora y fecha						
DATE	DATEJUL	DATETIME	DAY	DHMS	HMS	HOUR
JULDATE	MDY	MINUTE	MONTH	SECOND	TIME	TODAY
WEEKDAY	YEAR					

Tabla 5. Funciones del SAS/BASE.

Ejemplo:

```
DATA C.PRECIPI;
  INFILE 'C:\DATOS\DATOS\GUARENAS.DAT';
  INPUT ANNO 4. (MES1-MES12) (8.1);
  MEDIA=MEAN(OF MES1-MES12);
  :
```

4.3 Instrucción ATTRIB

Permite especificar formato, formato de entrada, etiqueta y longitud de una o mas variables.

Formato:

```
ATTRIB variable [FORMAT = formato] [INFORMAT = formato de
entrada]
        [LABEL = etiqueta] [LENGTH=[$]longitud] ...
```

Ejemplo:

```
ATTRIB X FORMAT=8.5 LABEL = "Var. Independiente" Y FORMAT=8.5
        LABEL = "Var. Dependiente"
```

4.4 Instrucción DELETE

Le dice al sistema SAS que pare el proceso de la observación actual, esta no es escrita en el data set y regresa al comienzo del paso de datos para otra observación.

Ejemplo:

```
DATA EJEMPLO;
    INPUT X Y Z;
    IF X>Y THEN DELETE;
    :
```

4.5 Instrucción FORMAT

Define el formato de una o mas variables.

Formato:

```
FORMAT variable(s) formato ...;
```

4.6 Instrucción IF – THEN y IF – THEN ELSE

Use esta instrucción cuando desee que SAS ejecute una o mas instrucciones para algunos pero no todos los casos (condicional).

Formato:

```
IF condición THEN instrucción;
[ELSE instrucción;]
```

Si la condición es verdadera se ejecuta la instrucción siguiente a THEN y si es falsa se ejecuta la instrucción siguiente a ELSE (si existe).

Para ejecutar más de una instrucción para una condición después de THEN o ELSE coloque DO END.

Ejemplo:

```
IF RESP="S" THEN DO
  N+1;
  Y=SQRT(X);
END;
ELSE DO
  N-1;
  Z=Y+X;
END;
```

4.7 Instrucción INFORMAT

Define el formato de entrada de una o mas variables.

Formato:

```
INFORMAT variables formato ...;
```

4.8 Instrucción LABEL

Define la etiqueta para una o mas variables. La etiqueta es almacenada junto con el nombre de la variables en el data set y es utilizada por los procedimientos SAS.

Formato:

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

4.9 Instrucción LIST

Use esta instrucción para listar o colocar sobre la ventana LOG las líneas de datos para la observación que está siendo procesada.

Ejemplo:

```
DATA EMPLEADO;
  INPUT SSN 1-9 W2 12-18
  IF W2=. THEN LIST;
  :
```

4.10 Instrucción PUT

Escribe líneas en la ventana LOG del SAS

Formato:

```
PUT [mensaje] [variable] ...
```

4.11 Instrucción *RETAIN*

Causa que el valor de una variable creada en un data set o asignada en el paso de datos sea retenido de la iteración anterior, también es usado para inicializar variables.

Formato:

```
RETAIN variables valorinicial ...
```

Todas las variables en el vector de datos del programa son normalmente colocados a valor faltante antes de la ejecución de un paso de datos.

Ejemplo:

```
DATA SUMAR;
  RETAIN CONTAR TOTAL 0;
  INPUT PUNTOS;
  CONTAR+1;
  TOTAL+PUNTOS;
CARDS;
10
5
.
6
4
```

Observe en el ejemplo anterior la instrucción `CONTAR + 1`, la cual es equivalente a decir `CONTAR=CONTAR + 1`.

4.12 Instrucciones de repetición

Al igual que otros lenguajes de programación el SAS tiene instrucciones que permiten realizar procesamiento iterativo de datos, aunque en SAS éste está asociado con el paso de datos para realizar cálculos, generar nuevas variables, leer datos, etc.

Para el procesamiento iterativo de datos los lenguajes disponen de estructuras o instrucciones de repetición que permiten ejecutar una o más instrucciones varias veces. Las estructuras de repetición pueden ser: indexadas o condicionales. En las indexadas se especifica los valores de una variable índice para la cual se repetirá el proceso (repita para); en las condicionales el proceso se repite mientras una condición sea verdadera (repita mientras) o hasta que la misma sea verdadera (repita hasta).

- Repita para variable desde valor inicial hasta valor final con un incremento

Formato:

```
DO variable = Vi TO Vf [BY incremento];
  instrucciones;
END;
```

- Repita para variable igual a algunos valores cualquiera (pueden ser numéricos o carácter)

Formato:

```
DO variable=V1, V2, . . . . , Vn
    instrucciones;
END;
```

- Repita mientras una condición sea verdadera

Formato:

```
DO WHILE (expresión);
    instrucciones;
END;
```

- Repita hasta que una condición sea verdadera

Formato:

```
DO UNTIL (condición)
    instrucciones;
END;
```

- Repita para procesamiento de arreglos (declara internamente la variable I como índice de referencia a los elementos del arreglo).

Formato:

```
DO OVER <nombrerreglo>
    instrucciones;
END;
```

Ejemplos:

```
DO Y=1 TO 10; /* Comienza en 1 hasta 1 */
```

```
DO K=2 TO 20 BY 2; /* Comienza en 2 hasta 20 de dos en dos */
```

```
DO L=1 TO 2 BY 0.1; /* Desde 1 hasta dos de 0,1 en 1,0 */
```

```
DO MES=1,3,5,7,8,10,12;
```

```
DO DIA=`LUN`, `MAR`, `MIE`, `JUE`, `VIE`;
```

```
DO WHILE (A < 20);
```

Se deposita en una cuenta el primero de Enero una cantidad de 100000 Bs. Cuál es el valor del capital al cabo de un año a una tasa de interés fija mensual del 2.5%?

```
DATA INVERSIO;
CAPITAL = 100000;
DO MES = 1 TO 12;
    CAPITAL+(CAPITAL*0.025);
END;
```

¿Por qué el data set tiene una sola instrucción? ¿Cuál es el valor de la variable MES?

4.13 Instrucción OUTPUT

Permite forzar o copiar una observación hacia el data set sin haber llegado al fin de un ciclo en el paso de datos.

Formato:

```
OUTPUT <data set>;
```

Ejemplo:

```
DATA INVERSIO;
CAPITAL=100000;
DO MES = 1 TO 12;
    CAPITAL+(CAPITAL*0.025);
    OUTPUT;
END;
```

Al comienzo de enero se deposita 100000 Bs. Y al comienzo de cada mes siguiente 20000 Bs., calcular los meses necesarios para que el capital sea de 500000 Bs.

```
DATA INVERSIO;
CAPITAL = 80000;
DO WHILE (CAPITAL<500000);
    MES+1;
    CAPITAL+20000;
    CAPITAL+(CAPITAL*0.025);
    OUTPUT;
END;
```

4.14 Uso de arreglos

Un arreglo es un conjunto de variables del mismo tipo, que son agrupadas bajo un único identificador. Cada una de las variables o elementos del arreglo se referencia mediante un índice que representa la posición del mismo dentro del arreglo.

Un arreglo se utiliza para procesar variables dentro de una observación del paso de datos. Se usan para realizar cálculos o comparaciones repetitivos sobre variables que se procesaran o manipularan de forma similar.

Los arreglos en SAS se clasifican de acuerdo a la forma de acceso en arreglos con subíndice explícito y con subíndice implícito.

4.14.1 Arreglos con subíndice explícito

Para definir un arreglo con subíndice explícito se utiliza la siguiente instrucción:

```
ARRAY <nomarreglo> [{n}] [$] [longitud] elementos [(valores
iniciales)];
```

Donde:

`n` indica el número de elementos del arreglo, se puede omitir o colocar asterisco para eliminar la enumeración en los elementos.

`$` se coloca si los elementos son de tipo carácter.

`longitud` define una longitud común para todos los elementos.

`elementos` son las variables del data set declaradas que sustituirán los elementos del arreglo.

`valores iniciales` para crear las variables o elementos con valores iniciales.

Ejemplos:

```
INPUT ANNO 4. (ENE FEB MAR ABR MAY JUN JUL AGO SEP OCT NOV DIC) (8.1);
ARRAY A{12} ENE--DIC;
```

Se crea un arreglo de 12 elementos donde A[1] corresponde a ENE, A[2] corresponde a FEB, etc.

```
INPUT ANNO 4. (MES1-MES12) (8.1);
ARRAY A MES1-MES12;
```

Si se omite `n`, el número de elementos del arreglo corresponde con el número de variables definidas, en el ejemplo desde MES1 A MES12 hay 12 variables.

```
INPUT ANNO 4. (MES1-MES12) (8.1);
ARRAY LLUVIA[12];
```

Si se omite los elementos el sistema SAS crea o referencia variables numeradas que corresponde con el nombre del arreglo, en el ejemplo estas son LLUVIA1, LLUVIA2, LLUVIA3, ..., LLUVIA12.

```
INPUT ANNO 4. (EN FE MA AB MA JN JL AG SE OC NO DI) (8.1);
ARRAY LLUVIA[*]EN, FE, MA, AB, MA, JN, JL, AG, SE, OC, NO, DI;
```

Si se coloca `*` en el número de elementos del arreglo el sistema SAS elimina las variables numeradas en los elementos y se debe incluir estos.

La instrucción ARRAY debe usarse para definir (declarar) el arreglo antes que éste sea referenciado. Si las variables (elementos) no han sido definidas las crea. No es una instrucción ejecutable.

Para procesar un arreglo con subíndice explícito generalmente se utiliza el lazo de repetición DO, aunque puede utilizarse también el DO WHILE y DO UNTIL. En forma general se usa el siguiente formato:

```
ARRAY ... (* Se define el arreglo *)
DO indice=1 TO n;
  instrucciones SAS que usen el arreglo mediante la variable índice
END;
```

El siguiente ejemplo lee el año y el nivel pluviométrico de los doce meses y calcula el total por trimestre:

```
DATA LLUVIA (DROP=I, J);
  INFILE 'C:\DATOS\LLUVIA.DAT';
  INPUT ANNO 4. (MES1-MES12) (8.1);
  ARRAY MES {12};
  ARRAY TRI{4};
  DO I=1 TO 4;
    TRI{I}=0;
    DO J=1 TO 3;
      TRI{I}+MES{(I-1)*3+J};
    END;
  END;
```

Obsérvese que en las declaraciones de los arreglos no se colocó los elementos y por lo tanto el sistema los asumió como variables numeradas (MES1, MES2, ..., MES12; TRI1, TRI2, TRI3 y TRI4).

El ejemplo anterior puede realizarse declarando a MES como un arreglo de dos dimensiones, tal y como se muestra a continuación:

```

:
INPUT ANNO 4. (MES1-MES12) (8.1);
  ARRAY MES {4, 3} MES1-MES12;
  ARRAY TRI{4};
  DO I=1 TO 4;
    TRI{I}=0;
    DO J=1 TO 3;
      TRI{I}+MES{I, J};
    END;
  END;
```

4.14.2 Arreglos con subíndice implícito

Para definir un arreglo con subíndice implícito se utiliza la siguiente instrucción:

```
ARRAY <nomarreglo>{índice} [$] [longitud] elementos;
```

`índice` es el nombre de la variable índice que se usará para definir el actual elemento que se está procesando. Si no se especifica el sistema define automáticamente la variable `_I_` y la coloca en el vector de datos pero no la escribe en el archivo que se está creando.

`$` se coloca si los elementos son de tipo carácter.

`longitud` define una longitud común para todos los elementos.

`elementos` son las variables declaradas del "data set" que sustituirán los elementos del arreglo.

`valores iniciales` para crear las variables o elementos con valores iniciales.

Para procesar los arreglos con subíndice implícito generalmente se utilizan los lazos de repetición `DO OVER`. La forma general de esta instrucción es:

```
DO OVER <nomarreglo>;
  instrucciones;
END;
```

El siguiente ejemplo muestra el uso de los arreglos con subíndice implícito. Se dan cuatro valores que representan la estatura en pulgadas y cuatro que representan el peso en libras y se desea guardarlos convertidos a centímetros y kilogramos respectivamente.

```
DATA CONVER;
  INPUT CODIGO $ ALT1 PE1 ALT2 PE2 ALT3 PE3 ALT4 PE4;
  ARRAY ALTURA ALT1-ALT4;
  ARRAY PESO PE1-PE4;
  DO OVER ALTURA;
    ALTURA=ALTURA*2.54;
    PESO=PESO*0.454;
  END;
CARDS;
A 65 166 67 175 55 110 69 178
B 66 133 67 161 75 200 72 180
C 72 182 63 115 50 105 67 190
;
```

Estos arreglos se pueden usar también con DO, DO UNTIL y DO WHILE pero se debe tener cuidado en usar variable índice que se definió, o si se omitió se usa la variable `_I_`. En el ejemplo anterior la instrucción DO OVER se puede sustituir por DO `_I_=1 TO 4`.

4.15 Instrucción SET

Se utiliza para leer, extraer o concatenar observaciones de uno o más conjunto de datos SAS existentes y crear uno nuevo. En tiempo de compilación el sistema SAS lee el descriptor de cada uno de los "data set" y coloca todas las variables en el descriptor de datos del "data set" que se creará. El formato general de la instrucción es la siguiente:

```
SET [<dataset> [(opciones)] <dataset>[(opciones)] ...] [opciones]
```

Donde:

dataset	El nombre o nombres de los conjunto de datos SAS
opciones	Especifica las opciones de cada conjunto de datos SAS a ser leído. Son las mismas opciones de un conjunto de datos SAS vistas anteriormente.
opciones SET	Especifica las opciones de la instrucción SET. Estas son:
	POINT = nombre crea una variable numérica cuyo valor es el número de la observación en el conjunto de datos de entrada que se quiere procesar. Es como un apuntador a un registro.
	NOBS = nombre crea una variable cuyo valor es el número total de observaciones en el conjunto de datos de entrada.
	END = nombre crea una variable que contiene una indicación de fin de archivo. Esta variable es inicializada en 0 y se coloca en 1 cuando se lee la última observación del conjunto de datos de entrada o a concatenar.

Ejemplos:

```
DATA NUEVO;
  SET VIEJO;
```

Hace una copia del conjunto de datos VIEJO con el nombre NUEVO.

```
DATA NUEVO;
```

```
SET VIEJO (DROP= X Y);
```

Hace una copia de un subconjunto de variables del conjunto de datos VIEJO.

```
DATA HOMBRES MUJERES;
  SET TODOS;
  IF SEX="M" THEN OUTPUT HOMBRES;
  ELSE
  OUTPUT MUJERES;
```

Crea conjunto de datos SAS formado por un subconjuntos de observaciones de otro conjunto de datos SAS, separa las mujeres y los hombres.

```
DATA LLUTRI;
  SET LLUVIA (KEEP=ANNO TRI1-TRI4);
  ARRAY MES {4,3} MES1-MES12;
  ARRAY TRI{4};
  DO I=1 TO 4;
    TRI{I}=0;
    DO J=1 TO 3;
      TRI{I}+MES{I,J};
    END;
  END;
```

Crea un conjunto de datos SAS formado por una variable de un conjunto de datos y crea nuevas variables. Para calcular los totales trimestrales (LLUTRI) se usa los datos de los meses (LLUVIA).

La instrucción SET se puede usar para concatenar archivos (conjuntos de datos). La concatenación consiste en formar (crear) un conjunto de datos a partir de dos o más .

Cuando los conjunto de datos iniciales tienen las mismas variables el nuevo conjunto de datos contendrá las observaciones del primer conjunto de datos listado y a continuación las observaciones del segundo, etc., esta concatenación se llama unión de archivos vertical.

```
DATA A95AL97;
  SET A95 A96 A97;
```

Suponiendo que los conjunto de datos A95, A96 y A97 contiene las mismas variables el nuevo conjunto de datos A95AL97 contendrá todas las observaciones de los tres archivos.

Si los archivos originales en la instrucción SET contiene diferentes variables, las observaciones obtenidas en el nuevo conjunto de datos tiene valores faltantes para las variables que no son comunes, o que están definidas en un sólo conjunto de datos.

Los conjunto de datos listados en la instrucción SET puede contener las mismas variables pero diferente atributo o formato. En este caso si se puede tomará el formato o atributo del primer conjunto de datos listado:

```
DATA A;
  NOMBRE="MARY";
DATA B;
  NOMBRE="SUSANA";
DATA C;
  SET A B;
```

La variable NOMBRE en el conjunto de datos A tiene tamaño 4 y en el B tiene tamaño 6; la variable NOMBRE en C tendrá tamaño 4 puesto que A está listado primero.

Se puede unir o mezclar dos archivos intercalando por algún orden, para lo cual se deben tener los archivos ordenados.

```
PROC SORT DATA=HOMBRES;
  BY APELLI;
PROC SORT DATA=MUJERES;
  BY APELLI;
DATA ORDEN;
  SET HOMBRES MUJERES;
  BY APELLI;
```

4.16 Instrucción MERGE

Se utiliza para unir observaciones de uno o más conjunto de datos SAS existentes en una observación simple de un conjunto de datos SAS nuevo. En tiempo de compilación el sistema SAS lee el descriptor de cada uno de los conjuntos de datos SAS y coloca todas las variables en el descriptor de datos del conjunto de datos que se creará. El formato general de la instrucción es la siguiente:

```
MERGE [<dataset> [(opciones)] <dataset>[(opciones)] ...]
[END=nombre]
```

Donde:

dataset	Los nombres de dos o más conjunto de datos SAS
opciones	Especifica las opciones de cada conjunto de datos SAS a ser leído. Son las mismas opciones de un conjunto de datos SAS vistas anteriormente.
END = nombre	crea una variable que contiene una indicación de fin de archivo. Esta variable es inicializada en 0 y se coloca en 1 cuando se lee la última observación del conjunto de datos de entrada o a cocatenar.

Ejemplos:

```

DATA PERSONA;
  INPUT NOMBRE $ CIUDAD $;
CARDS;
ANA MERIDA
SUSANA BARINAS
NANCY CARACAS
;
DATA CARRO;
  INPUT ANNO MARCA $;
CARDS;
1980 JEEP
1985 CHEVETE
1994 NEON
;
DATA MEZCLA;
  MERGE PERSONA CARRO;

```

En el ejemplo anterior se tienen dos archivos con el mismo número de observaciones, en este caso el resultado de la instrucción MERGE es un conjunto de datos que contiene tres observaciones con las variables NOMBRE CIUDAD ANNO y MARCA. A continuación se muestra como se verán los datos:

NOMBRE	CIUDAD	ANNO	MARCA
ANA	MERIDA	1980	JEEP
SUSANA	BARINAS	1985	CHEVETE
NANCY	CARACAS	1994	NEON

Esta mezcla se denomina uno a uno porque toma la primera observación de cada conjunto de datos y crea la primera observación del Conjunto de datos resultado, luego toma la segunda de cada conjunto de datos y así sucesivamente. En caso de que alguno de los archivos tenga mas datos que el otro, el resultado tendrá el número de observaciones del que tiene más, con valores faltantes en las variables del que tiene menos, esto se puede controlar usando marcas de fin de datos como se muestra:

```

DATA PERSONA;
  INPUT NOMBRE $ CIUDAD $;
CARDS;
ANA MERIDA
SUSANA BARINAS
NANCY CARACAS
GENRY MERIDA
;
DATA CARRO;
  INPUT ANNO MARCA $;
CARDS;
1980 JEEP
1985 CHEVETE
1994 NEON

```

```
;
DATA MEZCLA;
  MERGE PERSONA CARRO END=FIN;;
  IF (FIN) THEN DELETE;
```

Si se tiene dos o más conjunto de datos que tiene una o más variables en común la mezcla se realiza utilizando la instrucción BY después de la instrucción MERGE. Las variables nombradas en el BY deben ser del mismo tipo y con el mismo nombre en cada uno de los conjuntos de datos SAS a unir. En la unión de archivos con una variable en común, pueda ocurrir que existe en ambos archivos varias observaciones en común y otras no, en cuyo caso las variables leídas de los conjuntos de datos son inicializadas a faltantes cuando el valor de la variable BY no es común.

En caso de utilizar este tipo de mezcla hay que ordenar todos los archivos por las variables BY.

```
DATA PERSONA;
  INPUT NOMBRE $ CIUDAD $;
CARDS;
ANA MERIDA
SUSANA BARINAS
NANCY CARACAS
GENRY MERIDA
;
PROC SORT;
  BY NOMBRE;
DATA CARRO;
  INPUT NOMBRE $ MARCA $;
CARDS;
ANA JEEP
ANA CHEVETE
NANCY NEON
ANGEL FIAT
GENRY FORD
;
PROC SORT;
  BY NOMBRE;
DATA MEZCLA;
  MERGE PERSONA CARRO;
  BY NOMBRE;
```

El resultado del ejemplo anterior se muestra a continuación, observe que hay tres registros con observaciones faltantes ¿Por qué?.

NOMBRE	CIUDAD	MARCA
ANA	MERIDA	JEEP
ANA	MERIDA	CHEVETE
ANGEL		FIAT
GENRY	MERIDA	TOYOTA
NANCY		NEON
SUSANA	BARINAS	

4.17 Instrucción UPDATE

Esta instrucción combina las observaciones de dos archivos actualizando los de un archivo denominado maestro con las observaciones de otro denominado transacciones. Esta instrucción debe estar acompañada de la instrucción BY y por lo tanto los dos archivos deben ser ordenados por estas variables.

El archivo maestro no debe contener más de una observación para un mismo valor de la(s) variable(s) BY.

Formato general de la instrucción:

```
UPDATE maestro [(opciones)] transac [(opciones)] [END =
nombre];
```

maestro	Nombre del conjunto de datos maestro, debe tener una sola observación para cada valor de la(s) variable(s) BY.
transac	Nombre del conjunto de datos que contiene las transacciones a realizar sobre el maestro, puede contener más de una observación para un valor de la(s) variable(s) BY.
opciones	Especifica las opciones de cada conjunto de datos SAS a ser leído. Son las mismas opciones de un conjunto de datos SAS vistas anteriormente.
END = nombre	crea una variable que contiene una indicación de fin de archivo. Esta variable es inicializada en 0 y se coloca en 1 cuando se lee la última observación del conjunto de datos de entrada o a cocatenar.

En el archivo de transacciones se coloca para cada observación sólo los valores de las variables que van a ser actualizadas.

5. Introducción al SAS/STAT

[6]

Los procedimientos estadísticos vistos hasta el momento corresponden al módulo base del SAS y permiten realizar estadística descriptiva de datos. El SAS/STAT es una herramienta de análisis estadístico que ofrece una amplia gama de capacidades incluyendo análisis de varianza, de regresión, categórico, multivariante, de sobrevivencia, cluster y no paramétrico. En primer lugar se presentará una lista resumen de los procedimientos del SAS/STAT y después se estudiará en detalle algunos de ellos.

5.1 Procedimientos del SAS/STAT

5.1.1 Procedimientos de regresión:

CADMOD	análisis de regresión lineal, log-lineal, regresión logística para datos categóricos.
GLM	usa el método de los mínimos cuadrados para estimar el modelo lineal general.
GLMMOD	produce un conjunto de datos SAS que contiene la matriz de diseño para un modelo usando los efectos facilitados por el procedimiento GLM.
LOGISTIC	estima el modelo de regresión logístico (datos dicotómicos) por el método de máximo verosimilitud.
NLIN	produce estimadores mínimos cuadrados y mínimos cuadrados ponderados de modelos no lineales.
ORTHOREG	produce regresión usando el método de Gentleman-Givens (datos imperfectos).
PHREG	análisis de regresión basado en el modelo de azar proporcional (Regresión de Cox).
PLS	modelo de regresión parcial por el método de mínimos cuadrados.
REG	estima modelos de regresión lineal por el método de mínimos cuadrados.
RSREG	estima los parámetros de una superficie de respuesta cuadrática completa.

TRANSREG encuentra transformaciones lineales y no lineales de variables usando el método de mínimos cuadrados alternativos para optimizar la estimación de los modelos de análisis de varianza, regresión lineal y correlación canónica.

5.1.2 Procedimientos de análisis de varianza:

ANOVA análisis de varianza para datos balanceados de una amplia variedad de diseño de experimentos.

CATMOD análisis de varianza para datos categóricos.

GENMOD produce estimaciones del modelo lineal generalizado.

GLM análisis de varianza para el modelo lineal general.

MIXED estima modelos general mixto (de efectos fijos y aleatorios)

NESTED análisis de varianza con efectos aleatorios.

NPAR1WAY análisis de varianza en una vía.

LATTICE produce análisis de varianza y análisis de covarianza simple para un experimento con diseño lattice.

TTEST calcula el estadístico t de la prueba de igualdad de dos medias.

VARCOMP estimadores de componentes de varianza en el modelo lineal general.

5.1.3 Procedimientos para datos categóricos:

FREQ produce tablas de frecuencias y tablas cruzadas.

CATMOD regresión logística y análisis de varianza para datos categóricos.

5.1.4 Procedimientos multivariantes:

CALIS análisis de estructuras de covarianzas.

CANCORR análisis de correlación canónica.

CANDISC análisis discriminante canónico.

CORRESP análisis de correspondencias simples y compuestas.

DISCRIM calcula funciones discriminantes para clasificar observaciones.

FACTOR análisis factorial.

PRINCOMP análisis de componentes principales.

- PRINTQUAL encuentra transformaciones lineales y no lineales de variables usando el método de mínimos cuadrados alternativos para optimizar propiedades de la matriz de correlación o covarianza de las variables transformadas.
- STEPDISC análisis discriminante paso a paso.

5.1.5 Procedimientos de análisis cluster:

- ACECLUS obtiene estimaciones aproximadas de la matriz de covarianza para el análisis cluster.
- CLUSTE análisis cluster (clasificación)
- FASTCLUS análisis cluster disjunto para conjunto de datos muy grandes.
- VARCLUS realiza o análisis cluster disjunto o jerárquico basado sobre la matriz de correlación o covarianza.
- TREE produce (imprime) el diagrama de árbol del análisis cluster, conocido como dendograma.

5.1.6 Procedimientos de análisis de sobrevivencia:

- LIFEREG ajusta modelos de sobrevivencia con datos censurados, as como también modelos de probabilidad tobit.
- LIFETEST estima unciones de sobrevivencia y tablas de vida.

5.2 Procedimiento ANOVA

Este es uno de los procedimientos disponibles en SAS/STAT para análisis de varianza para datos balanceados con la excepción de diseño de cuadrados latinos y ciertos diseños de bloques balanceados incompletos. Este procedimiento verifica si el diseño es balanceado.

Formato:

```
PROC ANOVA <opciones>;
  <instrucciones ANOVA>;
```

<opciones>

- DATA= nombre del conjunto de datos que contiene las observaciones a analizar.
- MANOVA hace que ANOVA use el modo multivariante de eliminación de observaciones con valores faltantes. Esta opción es útil en modo iterativo.

MULTIPASS hace que ANOVA relea los datos de entrada.

OUTSTAT= nombre de un conjunto de datos SAS que contendrá las sumas de cuadrados, el estadístico F y los niveles de probabilidad de cada efecto en el modelo.

<instrucciones ANOVA>

CLASS variables; declara las variables de clasificación que identifican los grupos en el análisis. Debe preceder a la instrucción MODEL y es obligatoria.

MODEL dependientes = efectos /opciones;

indica las variables dependientes y los efectos independientes. La sintaxis de los efectos se indicará más abajo. Si no se especifica efectos ANOVA estima solamente el intercepto, con la prueba de hipótesis que la media de la variable independiente es cero. Las opciones después de la / son las siguientes:

INT hace que ANOVA imprima la prueba de hipótesis asociada con el intercepto como un efecto en el modelo.

NOUNI hace que ANOVA no imprima el análisis univariado que son producido por defecto.

ABSORB variables; absorbe los efectos de clasificación en el modelo.

BY variables; procesa el análisis de varianza por grupos, especificados en los niveles de las variables BY.

FERQ variable; especifica que la variable es de tipo frecuencia.

MEANS efectos / opciones;

ANOVA calcula las medias de las variables dependientes para algún efecto que aparezca en el lado derecho del modelo. Las opciones que pueden aparecer después del / son entre otras:

BON realiza la prueba t de Bonferoni de diferencias entre medias para todos los efectos principales dados en la instrucción MEANS.

DUNCAN realiza la prueba de rangos múltiples de Duncan sobre todos los efectos principales dados en la instrucción MEANS.

GABRIEL realiza las comparaciones múltiples de Gabriel sobre todos los efectos principales dados en la instrucción MEANS.

LSD

SCHEFFE realiza las comparaciones múltiples de Scheffe

sobre todos los efectos principales dados en la instrucción MEANS.

TUKEY realiza la prueba de rangos de Tukey sobre todos los efectos principales dados en la instrucción MEANS.

ALPHA= especifica el nivel de significancia para las comparaciones sobre las medias, por defecto el valor es de 0.05. Con la opción DUNCAN sólo puede especificar 0.01, 0.05 y 0.1; para las demás opciones puede utilizar valores entre 0.0001 y 0.9999.

REPEATED nombrefactor niveles(valores) / opciones;
cuando los valores de las variables dependientes representan medidas repetidas sobre las mismas unidades experimentales, por ejemplo crecimiento de unas plantas tomadas cada semana, esta instrucción permite probar la hipótesis acerca de los factores medidos, así como también las interacciones entre los individuos.

Para especificar los efectos tomemos como ejemplo que se tiene un conjunto de datos PRUEBA con la variable respuesta Y, y con tres variables factores A, B y C. Así que para realizar un análisis de varianza se escribe:

```
PROC ANOVA DATA=PRUEBA;
  CLASS A B C;
  MODEL Y = A B C A*B A*C B*C A*B*C;
```

Los efectos principales se especifica con A B C, los efectos cruzados con A*B A*C B*C A*B*C. Se puede especificar modelo factorial completo, es decir, efectos principales y efectos cruzados, con la notación de barras:

```
MODEL Y = A|B|C;
```

El procedimiento ANOVA puede utilizarse en forma iterativa, después de especificar el modelo y correr el programa con la instrucción RUN, las instrucciones: MEANS, MANOVA, TEST y REPEATED puede ser utilizadas sin necesidad de especificar nuevamente el modelo, ni recalculan las sumas cuadradas del modelo. Para salirse del modo iterativo basta ejecutar un paso de datos, otro procedimiento, salirse del SAS o con la instrucción QUIT.

Ejemplos:

Análisis de varianza en una vía con comparación de medias (análisis a posteriori).

```
PROC ANOVA;
  CLASS X;
  MODEL Y=X;
  MEANS Y /DUNCAN TUKEY;
```

Análisis de varianza para probar si hay diferencias en la lluvia media de los cuatro primeros meses.

```
DATA PRUEBA;
  SET C.GUARENAS (KEEP=LLUVIA MES);
  ARRAY MESES {4} MES1-MES4;
  DO I=1 TO 4;
    LLUVIA=MESES{I};
    MES=I;
    OUTPUT;
  END;
PROC ANOVA DATA=PRUEBA;
  CLASS MES;
  MODEL LLUVIA=MES;
  MEANS MES /DUNCAN;
```

Bloques completamente aleatorizados:

```
DATA PRUEBA;
  INPUT BLOQ TRAT $ DEVOL VALOR @@;
CARDS;
1 A 32.6 112 1 B 36.4 130 1 C 29.5 106
2 A 42.7 139 2 B 47.1 143 2 C 32.9 112
3 A 35.3 124 3 B 40.1 134 3 C 33.6 116
;
PROC ANOVA;
  CLASS BLOQ TRAT;
  MODEL DEVOL VALOR=BLOQ TRAT;
  MEANS BLOQ TRAT / DUNCAN;
```

5.3 Procedimiento GLM

Este procedimiento usa el método de los mínimos cuadrados para estimar el modelo lineal general. Con GLM se puede realizar regresión (simple, múltiple, ponderada y polinómica), análisis de varianza (especialmente para datos no balanceados), análisis de covarianza, análisis de varianza multivariante (MANOVA), análisis de varianza con medidas repetidas y correlación parcial.

Formato:

```
PROC GLM <opciones>;
  <instrucciones ANOVA>;
```

<opciones>

DATA= nombre del conjunto de datos que contiene las observaciones a analizar.

MANOVA hace que GLM use el modo multivariante de eliminación de observaciones con valores faltantes. Esta opción es útil en modo interactivo.

MULTIPASS hace que GLM relea los datos de entrada cuando sea necesario.

NOPRINT suprime la salida normal de resultados

OUTSTAT= nombre de un conjunto de datos SAS que contendrá las sumas de cuadrados, el estadístico F y los niveles de probabilidad de cada efecto en el modelo.

<instrucciones GLM>

CLASS variables; declara las variables de clasificación que identifican los grupos en el análisis. Debe preceder a la instrucción MODEL y es obligatoria.

MODEL dependientes = independientes /opciones;

indica las variables dependientes y los efectos independientes. La sintaxis de los efectos se indicará más abajo. Si no se especifica efectos GLM estima solamente el intercepto. Las opciones después de la / son las siguientes:

INT hace que GLM imprima la prueba de hipótesis asociada con el intercepto como un efecto en el modelo.

NOINT hace que el parámetro intercepto no sea incluido en el modelo.

NOUNI hace que GLM no imprima el análisis univariado que son producido por defecto.

ALPHA= especifica el nivel para los intervalos de confianza, sólo acepta 0.1, 0.01 y 0.05. Por defecto toma 0.05.

P imprime los valores observados, predichos y residuales para cada observación que no contenga valores faltantes para las variables independientes.

XPX imprime la matriz de productos cruzados $X'X$

INVERSE imprime la matriz inversa o la inversa generalizada $X'X$.

ABSORB variables; absorbe los efectos de clasificación en el

modelo.

BY variables; procesa el análisis de varianza por grupos, especificados en los niveles de las variables BY.

FERQ variable; especifica que la variable es de tipo frecuencia.

ID variable; especifica variable para identificar las observaciones en la salida.

WEIGHT variable; especifica una variable para ponderar las observaciones.

MEANS efectos / opciones;

GLM puede calcular las medias de las variables dependientes para algún efecto que aparezca en el lado derecho del modelo. Las opciones después del / son entre otras:

BON realiza la prueba t de Bonferoni de diferencias entre medias para todos los efectos principales dados en la instrucción MEANS.

DUNCAN realiza la prueba de rangos múltiples de Duncan sobre todos los efectos principales dados en la instrucción MEANS.

GABRIEL realiza las comparaciones múltiples de Gabriel sobre todos los efectos principales dados en la instrucción MEANS.

SCHEFFE realiza las comparaciones múltiples de Scheffe sobre todos los efectos principales dados en la instrucción MEANS.

TUKEY realiza la prueba de rangos de Tukey sobre todos los efectos principales dados en la instrucción MEANS.

ALPHA= especifica el nivel de significancia para las comparaciones sobre las medias, por defecto el valor es de 0.05. Con la opción DUNCAN sólo puede especificar 0.01, 0.05 y 0.1; puede utilizar valores entre 0.0001 y 0.9999.

REPEATED nombrefactor niveles(valores) / opciones;

cuando los valores de las variables dependientes representan medidas repetidas sobre las mismas unidades experimentales, por ejemplo crecimiento de unas plantas tomadas cada semana, esta instrucción permite probar la hipótesis acerca de los factores medidos, así como también las interacciones entre los individuos.

Ejemplo:

A continuación se muestra la instrucción MODEL de acuerdo a la clase de modelo que se desea analizar, A, B, C representa variables CLASS y X, X1, X2, X3, Y, Y1, Y2 y Y3 representan variables continuas.

MODEL Y=X1;	regresión simple
MODEL Y=X1 X2 X2 ...;	regresión múltiple
MODEL Y=X1 X1*X1;	regresión polinómica
MODEL Y1 Y2 ...=X1 X2 ...;	regresión multivariante
MODEL Y=A;	análisis de varianza en una vía
MODEL Y=A B C;	de efectos principales
MODEL Y=A B A*B;	factorial (interacción)
MODEL Y=A B(A) C(B A);	anidado
MODEL Y1 Y2=A B;	MANOVA
MODEL Y=A X1;	análisis de covarianza

5.4 Procedimiento PRINCOMP

El procedimiento PRINCOMP realiza el análisis de componentes principales. Como entrada se pueden usar datos en bruto, una matriz de correlación, una matriz de covarianza, o una matriz de sumas de cuadrados y producto cruzados (SSCP). Se pueden crear como resultado del análisis dataset que contengan autovalores, autovectores, y puntuaciones, estandarizadas o no, de los componentes principales.

El análisis de componentes principales transforma a un conjunto de variables correlacionadas en un nuevo conjunto de variables incorrelacionadas llamadas componentes principales (Dallas, 2000). Dichas componentes son combinaciones lineales derivados en forma decreciente de importancia de tal manera que el primer componente es el que explica la mayor variabilidad posible de la variación total contenida en los datos originales. La segunda componente principal se elige de forma que explique la mayor cantidad posible de variación que resta sin explicar por la primera componente principal. La tercera componente no está correlacionada ni con la primera ni con la segunda y tiene la tercera varianza más grande, y así sucesivamente.

Formato:

```
PROC PRINCOMP < opciones >;
  BY variables;
  FREQ variable;
  ID variables;
  PARTIAL variables;
  VAR variables;
  WEIGHT variable;
```

<opciones>

DATA= nombre del dataset que contiene las observaciones a usar. Este conjunto de datos puede ser de tipo DATA, CORR, COV, ACE, FACTOR o SSCP.

OUT= crea un dataset que contiene todos los datos originales así como también las puntuaciones de los componentes principales.

OUTSTAT crea un dataset que contiene los siguientes estadísticos medias, desviaciones estándar, número de observaciones, correlaciones o covarianzas, autovalores y autovectores.

COV Calcula la matriz de covarianza para los componentes principales

N= Número de componentes principales a ser calculados

NOINT Omite el intercepto para el modelo

PREFIX= Prefijo para nombrar los componentes principales

PARPREFIX= Prefijo para nombrar las variables residuales

SINGULAR= Especifica el criterio de singularidad

STD Estandariza las puntuaciones de los componentes principales

VARDEF= Especifica el divisor usado en el cálculo de varianzas y desviaciones estándar

NOPRINT Suprime mostrar todas las salidas

PLOTS= Especifica las opciones con los detalles de los gráficos

Instrucciones del procedimiento PRINCOMP

BY variables; se pueden obtener análisis por separado de observaciones en grupos que son definidos por las variables especificadas en BY.

FREQ variable; provee frecuencias para cada observación en DATA= data set. Específicamente, si n es el valor de la variable FREQ para una observación dada, entonces la observación es usada n veces.

ID var; etiqueta las observaciones usando valores de la primera variable ID en el gráfico de puntuaciones de los componentes principales.

PARTIAL var; Se utiliza para analizar parcialmente la matriz de correlación (o covarianza), especificando en la declaración PARTIAL las variables

cuantitativas que serán incluidas.

VAR var; lista las variables numéricas a ser analizadas. Si se omite, todas las variables numéricas no especificadas en otras declaraciones serán analizadas.

WEIGHT var; Para usar pesos relativos para cada observación en el conjunto de datos de entrada, se coloca el peso en una variable del conjunto de datos y se especifica su nombre en la declaración WEIGHT.

5.5 Procedimiento REG

Este procedimiento estima modelos de regresión lineal por el método de mínimos cuadrados. Puede ser determinados subconjunto de variables independientes que mejor predicen a la variable dependiente o respuesta. REG es uno de los procedimientos de regresión con que cuenta es SAS/STAT, otros son: CATMOD, GLM, NLIN, ORTHOREG y RSREG.

Formato:

```
PROC REG <opciones>;
    <instrucciones>;
```

<opciones>

CORR imprime la matriz de correlación para todas las variables listadas en las instrucciones MODEL y VAR.

COVOUT muestra las matrices de covarianzas para los parámetros estimados. Es válida sólo si esta la opción OUTEST=.

DATA= nombre del conjunto de datos que contiene las observaciones a usar. Este conjunto de datos puede ser de tipo DATA, CORR o COV.

NOPRINT suprime la impresión de la salida.

OUTEST= hace que los parámetros estimados y los estadísticos opcionales sean almacenados en este conjunto de datos.

SIMPLE imprime la suma, media, varianza, desviación estándar y la suma de cuadrados no corregida para cada variable usada en REG.

<Instrucciones>

```
MODEL dependientes = regresores /opciones;
    especifica el modelo a usar en la regresión, antes
```

del signo igual se escribe las variables dependientes (respuestas) y después del igual las independientes. Las variables especificadas en el modelo deben ser numéricas. Las opciones de ésta instrucción son:

SELECTION=nombre se refiere al método usado para seleccionar el modelo, donde nombre puede ser: FORWARD (o F), BACKWARD (o B), STEPWISE, MAXR, MINR, RSQUARE, ADJRSQ, CP o NONE.

GROUPNAMES='nombre1' 'nombre2' ...
provee nombres para las variables grupos.

NOINT suprime el término intercepto del modelo.

BY var; para obtener análisis separados sobre las observaciones en grupos definidos por estas variables.

FREQ var; especifica variables tipo frecuencias.

ID var; nombre de una variable para identificar las observaciones en la salida.

VAR var; lista de las variables para las cuales los productos cruzados serán calculados, o que pueden ser añadidas en forma iterativa al modelo o a ser usadas gráficos.

WEIGHT var; declara una variable para ponderar las observaciones.

ADD var; añade variables independientes al modelo de regresión (en forma iterativa).

DELETE var; elimina variables independientes del modelo de regresión (en forma iterativa).

OUTPOUT OUT= crea un conjunto de datos con los valores predichos, residuales y otros estadísticos.

PLOT varY1*varX1=<simbolo> varY2*varX2=<simbolo> ...
genera gráficos XY

Ejemplo:

```
DATA PRUEBA;
  INPUT NOMBRE $ PESO ESTAT;
CARDS;
ALFREDO 72 185
CARLOS 67 172
ALICIA 55 161
:
PROC REG;
  MODEL ESTAT=PESO;
  PLOT ESTAT*PESO;
```

5.6 Procedimiento TTEST

Este procedimiento calcula el estadístico t para probar la hipótesis que la media de dos grupos de observaciones son iguales.

El estadístico t calculado es basado en el supuesto de que las varianzas de los dos grupos son iguales, y calcula una aproximación al estadístico t basado en el supuesto de que las varianzas no son iguales (Behrens-Fisher). Para cada t, los grados de libertad y los niveles de probabilidad son dados

Formato:

```
PROC TTEST <opciones>;  
<Instrucciones del TTEST>;
```

<opciones>

DATA = Especifica el conjunto de datos SAS que contiene las observaciones a comparar.

COCHRAN Especifica que se calcule la aproximación de Cochran y Cox del nivel probabilidad del estadístico t para varianzas desiguales.

<Instrucciones del TTEST>

CLASS variable; especifica la variable de clasificación, es decir, la variable que contiene los dos grupos que se compararan. Esta variable debe tener sólo dos niveles (valores). Es obligatoria;

VAR variables; Representa los nombres de las variables dependientes cuyas medias serán comparadas. Si se omite se tomarán todas las variables numéricas se incluirán en el análisis (excepto la variable de clasificación).

BY variables; Se utiliza para obtener análisis separados sobre las observaciones en grupos definidos por los diferentes niveles de las variables BY.

Ejemplo:

```
DATA PRUEBA (KEEP=LLUVIA MES);  
  SET C.GUARENAS;  
  LLUVIA=MES1;  
  MES=1;  
  OUTPUT;  
  LLUVIA=MES7;
```

```
MES=2;
OUTPUT;
PROC TTEST DATA=PRUEBA;
  CLASS MES;
  VAR LLUVIA;
```

En el paso de datos se crea el conjunto de datos que contiene la variable que se va a comparar (LLUVIA) y la variable MES que representa los dos grupos a comparar, 1 corresponde a enero y 2 a julio; luego se compara si la lluvia media del mes de enero es igual a la lluvia media del mes de julio.

Este procedimiento no sirve para realizar comparaciones apareadas (generalmente se refiere a mediciones antes y después de aplicar un tratamiento sobre el mismo individuo), en el caso de necesitar este tipo de análisis se debe realizar mediante el procedimiento MEANS, creando una variable extra que contenga la resta de las dos observaciones apareadas.

5.7 Ejemplos SAS/STAT

Ejemplo 1 (ejem1_stat.sas):

```
/* SE SOSPECHA QUE 5 MAQUINAS ALIMENTADORAS EN CIERTA PLANTA ESTAN
LLENNADO LATAS A DIFERENTES NIVELES. SE TOMARON MUESTRAS AL AZAR DE
LA PRODUCCION DE CADA MAQUINA, CON LOS SIGUIENTES RESULTADOS:
```

MAQUINA				
1	2	3	4	5
11.95	12.18	12.16	12.25	12.10
12.00	12.11	12.15	12.30	12.04
12.25		12.08	12.10	12.02
12.10				12.02

```
ANALIZAR LOS DATOS.
```

```
GRAFICAR LOS RESIDUOS VS. OBSERVACIONES.
```

```
RESIDUOS VS. PREDICCIONES.
```

```
RESIDUOS VS. TIEMPO.
```

```
(REALIZAR ANALISIS DE RESIDUOS.)
```

```
*/
```

```
OPTIONS LS=72 PS=55;
```

```
DATA LLENADO;
```

```
INPUT MAQUINA $ LLENADO @@;
```

```
TIEMPO + 1;
```

```
OUTPUT;
```

```
CARDS;
```

```
A 11.95 A 12.00 A 12.25 A 12.10
B 12.18 B 12.11
C 12.16 C 12.15 C 12.08
D 12.25 D 12.30 D 12.10
E 12.10 E 12.04 E 12.02 E 12.02
```

```

;

/* IMPRIMIR DATOS INICIALES */
PROC PRINT;
RUN;

/* GENERAR EL MODELO */
PROC GLM;
  CLASS MAQUINA;
  MODEL LLENADO = MAQUINA;
  OUTPUT OUT=NU_DATA P=VAL_EST R=RESIDUO;
RUN;

/* IMPRIMIR DATOS NUEVOS (RESIDUOS Y ESTIMADOS) */
PROC PRINT DATA=NU_DATA;
RUN;

/* ANALISIS DE RESIDUALES */
PROC PLOT VPERCENT =33;
  PLOT RESIDUO*VAL_EST='p';
  PLOT RESIDUO*LLENADO;
  PLOT RESIDUO*TIEMPO='p';
RUN;

```

Ejemplo 2 (ejem2_stat.sas):

```

/* UNA COMPAÑIA DE SERVICIOS PUBLICOS DESEA CONSTATAR SI LOS VOLTIMETROS
  UTILIZADOS SON HOMOGENEOS. SE TRATA COMO UN DISEÑO DE EFECTOS
  ALEATORIOS EN EL MODELO COMPLETAMENTE ALEATORIZADO. */

DATA EJEMPLO1;
DO I=1 TO 3;
  DO TRAT=1 TO 6;
    INPUT Y @;
    OUTPUT;
  END;
END;
CARDS;
0.95 0.33 -2.15 -1.20 1.80 -1.05
1.06 -1.46 1.70 0.62 0.88 -0.65
1.96 0.20 0.48 1.50 0.20 0.80
;

PROC PRINT;
RUN;

PROC GLM DATA=EJEMPLO1;
  CLASS TRAT;
  MODEL Y = TRAT;
  RANDOM TRAT;
RUN;

```

Ejemplo 3 (ejem3_stat.sas):

```

/* TRES SOLUCIONES DE LAVADO DIFERENTES ESTAN SIENDO COMPARADAS PARA
ESTUDIAR SU EFECTIVIDAD EN EL RETARDO DEL CRECIMIENTO DE BACTERIAS
CONTENIDAS EN 5 GALONES DE LECHE. EL ANALISIS ES HECHO EN UN
LABORATORIO, Y SOLO TRES INTENTOS PUEDEN SER EJECUTADOS EN UN SOLO
DIA. DEBIDO A QUE LOS DIAS PUEDEN REPRESENTAR UNA FUENTE POTENCIAL
DE VARIABILIDAD,EL EXPERIMENTADOR DECIDE USAR DISEÑO DE BLOQUES
ALEATORIZADOS.
LAS OBSERVACIONES SON TOMADAS PARA CUATRO DIAS. ANALICE LOS
DATOS Y OBTENGA CONCLUSIONES. */

OPTIONS LS=72 PS=55;

DATA BACTE;
INPUT SOLUCION $ @;
DO DIAS = 1 TO 4;
    INPUT EFECTIVI @;
    TIEMPO + 1;
    OUTPUT;
END;
CARDS;
1 13 22 18 39
2 16 24 17 44
3 5 4 1 22
;

/* IMPRIMIR DATOS INICIALES */
PROC PRINT;
RUN;

/* GENERAR EL MODELO */
PROC GLM;
    CLASS SOLUCION DIAS;
    MODEL EFECTIVI=SOLUCION DIAS;
    MEANS SOLUCION / DUNCAN;
    OUTPUT OUT=NU_DATA P=VAL_EST R=RESIDUO;

/* IMPRIMIR DATOS NUEVOS (RESIDUOS Y ESTIMADOS) */
PROC PRINT DATA=NU_DATA;

/* ANALISIS DE RESIDUALES */
PROC PLOT VPERCENT =50;
    PLOT RESIDUO*VAL_EST='p';
    PLOT RESIDUO*EFECTIVI='p';
    PLOT RESIDUO*TIEMPO='p';
RUN;

```

Ejemplo 4 (ejem4_stat):

```

/* EN EL PROBLEMA SE SUPONE QUE LA CANTIDAD DE CARBON USADA EN LA
PRODUCCION DE ACERO TIENE UN EFECTO SOBRE SU RESISTENCIA SE APLICAN
TRATAMIENTOS CUANTITATIVOS IGUALMENTE ESPACIADOS */

DATA PROB1;

```



```

INPUT CARBON RESIS @@;
CARDS;
.10 23 .10 36 .10 31 .10 33 .10 31 .10 31
.20 42 .20 26 .20 47 .20 34 .20 37 .20 31
.30 47 .30 43 .30 43 .30 39 .30 42 .30 35
;

PROC PRINT;

PROC GLM ;
  CLASS CARBON;
  MODEL RESIS=CARBON / P ;
  MEANS CARBON / TUKEY;
  CONTRAST 'LINEAL' CARBON -1 0 1;
  CONTRAST 'CUADRATICO' CARBON 1 -2 1;
  OUTPUT OUT=NUEVO P=RESIS_ES R=RESIDUOS;

PROC PLOT data = nuevo;
  PLOT RESIDUOS*RESIS;

PROC PLOT data = nuevo;
  PLOT RESIDUOS*RESIS_ES;

RUN;

```

Ejemplo 5 (Componentes principales, tomado del Ejemplo 58.2 del SAS/STAT 9.1 User's Guide, pag 3619):

Tasas de crimen por cada 100.000 personas en siete categorías, por estados, para Estados Unidos en el año 1977.

```

title 'Crime Rates per 100,000 Population by State';
data Crime;
input State $1-15 Murder Rape Robbery Assault
Burglary Larceny Auto_Theft;
cards;
Alabama          14.2 25.2 96.8 278.3 1135.5 1881.9 280.7
Alaska           10.8 51.6 96.8 284.0 1331.7 3369.8 753.3
Arizona          9.5 34.2 138.2 312.3 2346.1 4467.4 439.5
Arkansas         8.8 27.6 83.2 203.4 972.6 1862.1 183.4
California       11.5 49.4 287.0 358.0 2139.4 3499.8 663.5
Colorado         6.3 42.0 170.7 292.9 1935.2 3903.2 477.1
Connecticut      4.2 16.8 129.5 131.8 1346.0 2620.7 593.2
Delaware         6.0 24.9 157.0 194.2 1682.6 3678.4 467.0
Florida          10.2 39.6 187.9 449.1 1859.9 3840.5 351.4
Georgia          11.7 31.1 140.5 256.5 1351.1 2170.2 297.9
Hawaii           7.2 25.5 128.0 64.1 1911.5 3920.4 489.4
Idaho            5.5 19.4 39.6 172.5 1050.8 2599.6 237.6
Illinois         9.9 21.8 211.3 209.0 1085.0 2828.5 528.6
Indiana          7.4 26.5 123.2 153.5 1086.2 2498.7 377.4
Iowa             2.3 10.6 41.2 89.8 812.5 2685.1 219.9

```

```

Kansas          6.6 22.0 100.7 180.5 1270.4 2739.3 244.3
Kentucky       10.1 19.1 81.1 123.3 872.2 1662.1 245.4
Louisiana     15.5 30.9 142.9 335.5 1165.5 2469.9 337.7
Maine         2.4 13.5 38.7 170.0 1253.1 2350.7 246.9
Maryland      8.0 34.8 292.1 358.9 1400.0 3177.7 428.5
Massachusetts 3.1 20.8 169.1 231.6 1532.2 2311.3 1140.1
Michigan      9.3 38.9 261.9 274.6 1522.7 3159.0 545.5
Minnesota    2.7 19.5 85.9 85.8 1134.7 2559.3 343.1
Mississippi   14.3 19.6 65.7 189.1 915.6 1239.9 144.4
Missouri     9.6 28.3 189.0 233.5 1318.3 2424.2 378.4
Montana      5.4 16.7 39.2 156.8 804.9 2773.2 309.2
Nebraska     3.9 18.1 64.7 112.7 760.0 2316.1 249.1
Nevada       15.8 49.1 323.1 355.0 2453.1 4212.6 559.2
New Hampshire 3.2 10.7 23.2 76.0 1041.7 2343.9 293.4
New Jersey   5.6 21.0 180.4 185.1 1435.8 2774.5 511.5
New Mexico   8.8 39.1 109.6 343.4 1418.7 3008.6 259.5
New York     10.7 29.4 472.6 319.1 1728.0 2782.0 745.8
North Carolina 10.6 17.0 61.3 318.3 1154.1 2037.8 192.1
North Dakota 0.9 9.0 13.3 43.8 446.1 1843.0 144.7
Ohio         7.8 27.3 190.5 181.1 1216.0 2696.8 400.4
Oklahoma     8.6 29.2 73.8 205.0 1288.2 2228.1 326.8
Oregon       4.9 39.9 124.1 286.9 1636.4 3506.1 388.9
Pennsylvania 5.6 19.0 130.3 128.0 877.5 1624.1 333.2
Rhode Island 3.6 10.5 86.5 201.0 1489.5 2844.1 791.4
South Carolina 11.9 33.0 105.9 485.3 1613.6 2342.4 245.1
South Dakota 2.0 13.5 17.9 155.7 570.5 1704.4 147.5
Tennessee    10.1 29.7 145.8 203.9 1259.7 1776.5 314.0
Texas        13.3 33.8 152.4 208.2 1603.1 2988.7 397.6
Utah         3.5 20.3 68.8 147.3 1171.6 3004.6 334.5
Vermont      1.4 15.9 30.8 101.2 1348.2 2201.0 265.2
Virginia     9.0 23.3 92.1 165.7 986.2 2521.2 226.7
Washington   4.3 39.6 106.2 224.8 1605.6 3386.9 360.3
West Virginia 6.0 13.2 42.2 90.9 597.4 1341.7 163.3
Wisconsin    2.8 12.9 52.2 63.7 846.9 2614.2 220.7
Wyoming     5.4 21.9 39.7 173.9 811.6 2772.2 282.0
;
ods graphics on;
proc princomp out=Crime_Components plots= score(ellipse ncomp=3);
id State;
run;

```

6. Introducción al SAS/GRAPH

[7] El SAS/GRAPH es el componente o módulo del sistema SAS para la visualización y presentación gráfica de los datos. SAS/GRAPH produce una gran cantidad de gráficos, diagramas y mapas de 2 y 3 dimensiones.

6.1 Instrucciones Globales del SAS/GRAPH

6.1.1 Instrucción AXIS

Modifica la apariencia, position y rango de valores de los ejes en los gráficos y diagramas.

6.1.2 Instrucción BY

Procesa los datos y ordena la salida de acuerdo a los valores de clasificación (igual como funciona en el SAS Base y SAS/STAT).

6.1.3 Instrucción GOPTIONS

Especifica las opciones del gráfico que controla la apariencia de los elementos del gráfico, tales como: colores por defecto, patrones de relleno, fuentes o tamaño del texto.

6.1.4 Instrucción LEGEND

Modifica la apariencia y posición y posición de la leyenda generada por los procedimientos que producen los gráficos, diagramas y mapas.

6.1.5 Instrucción PATTERN

Define las características de los patrones usados usados en los gráficos creados por los procedimientos GAREABAR, GBARLINE, GCHART, GCONTOUR, GMAP y GPLOT.

6.1.6 Instrucción SYMBOL

Define las características de los símbolos que muestran los datos colocados por la instrucción PLOT usada por los procedimientos GBARLINE, GCONTOUR y GPLOT. La instrucción SYMBOL también controla la apariencia de las líneas en el contorno de los gráficos.

6.1.7 Instrucción TITLE, NOTE y FOOTNOTE

Añade texto a los mapas, gráficos, diagramas y diapositivas. Controlan el contenido, apariencia y lugar del título, pie de página y notas del gráfico.

6.2 Destinos para enviar la salida del SAS/GRAPH

La salida de los procedimientos del sistema SAS es controlada por ODS (Output Delivery System). ODS controla el destino donde es enviada la salida, la cual puede ser un archivo, una ventana gráfica, directamente a la impresora y otros.

Por defecto ODS aplica un estilo a la salida. Los estilos controlan la apariencia de la salida incluyendo los colores y fuente que se usan.

Cada destino está asociado a un controlador de dispositivo por defecto para generar la salida gráfica. Los controladores de dispositivos determinan la forma que toman los gráficos.

6.2.1 Enviando la salida a la ventana GRAPH

Cuando trabajamos en ambiente interactivo como el Windows, el destino por defecto es la ventana GRAPH. El siguiente ejemplo envía la salida a la ventana GRAPH.

```
goptions reset=all border;
title "DISTRIBUCIÓN DE LOS ESTUDIANTES POR EDAD";

proc gchart data=C.ALUMNOS ;
    vbar EDAD / subgroup=SEXO;
run;
quit;
```

6.2.2 Enviando la salida a un archivo

```
filename gout "c:\datos\GraEdad.gif";
goptions reset=all device=gif gsfname=gout border;

title "DISTRIBUCION DE LOS ESTUDIANTES POR EDAD";

proc gchart data=C.ALUMNOS ;
    vbar EDAD / subgroup=SEXO;
run;
```

6.2.3 Enviando la salida a un archivo rtf (MS Word)

```
ods listing close;
ods rtf;
goptions reset=all border;
```

```
title "DISTRIBUCION DE LOS ESTUDIANTES POR EDAD";

proc gchart data=C.ALUMNOS ;
    vbar EDAD / subgroup=SEXO;
run;
quit;
ods rtf close;
ods listing;
```

6.2.4 Enviando la salida a un archivo pdf

```
ods listing close;
ods pdf;
goptions reset=all border;
title "DISTRIBUCION DE LOS ESTUDIANTES POR EDAD";

proc gchart data=C.ALUMNOS ;
    vbar EDAD / subgroup=SEXO;
run;
quit;
ods pdf close;
ods listing;
```

6.3 Cambiando la apariencia del gráfico

SAS/GRAPH produce la salida gráfica en un área denominada “área de salida gráfica”. Esta área está contenida dentro del área del dispositivo. Las características de ambas áreas son determinadas por los valores de los parámetros del dispositivo específico. En muchos casos las dimensiones del área de salida gráfica son iguales a los del área del dispositivo, en particular los monitores y terminales. Los dispositivos como las impresoras crean un margen entre desde las dimensiones del área de salida gráfica hasta el área del dispositivo gráfico.

Algunas de las características de ambas áreas se pueden modificar usando las opciones gráficas.

Las dimensiones del área del dispositivo gráfico son controladas por XMAX y YMAX en pulgadas, a estos valores están asociados XPIXELS y YPIXELS.

Las dimensiones del área de salida gráfica están controladas por los parámetros HSIZE y VSIZE del dispositivo. En el caso de dispositivos como el monitor los valores de HSIZE y VSIZE son cero.

En la siguiente figura se ve la relación entre estos parámetros en caso de dispositivos como la impresora.

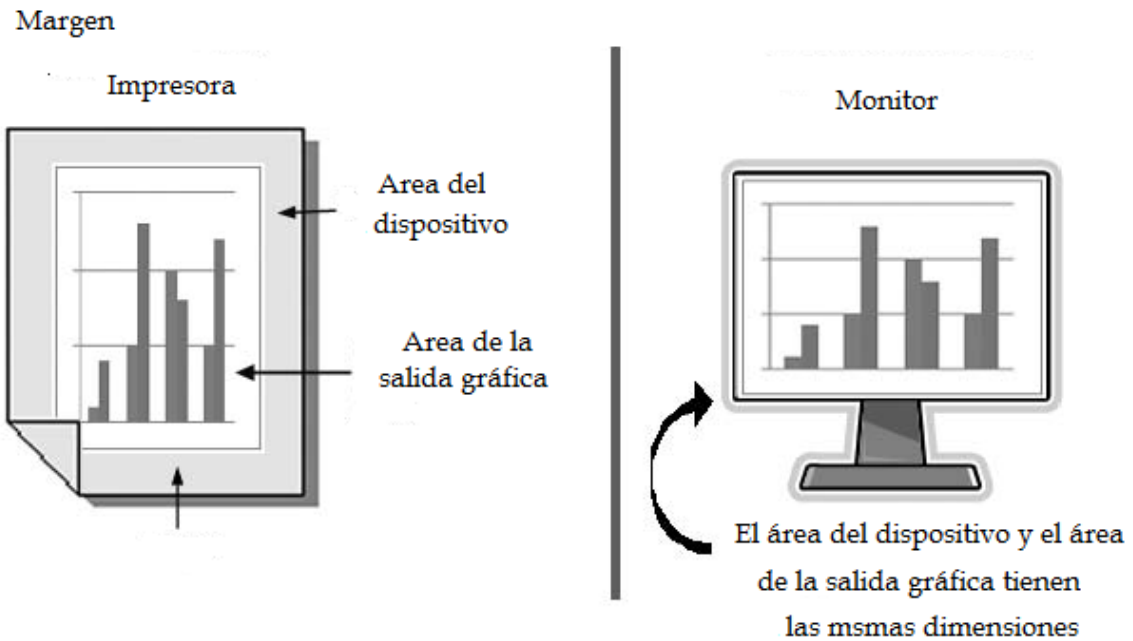


Fig 28. Área de salida gráfica

HORIGIN y VORIGIN definen los márgenes izquierdo e inferior respectivamente. Los márgenes derecho y superior son calculados por el dispositivo como sigue:
 $\text{Margen_derecho} = \text{XMAX} - (\text{HSIZE} + \text{HORIGIN})$
 $\text{Margen_superior} = \text{YMAX} - (\text{VSIZE} + \text{VORIGIN})$

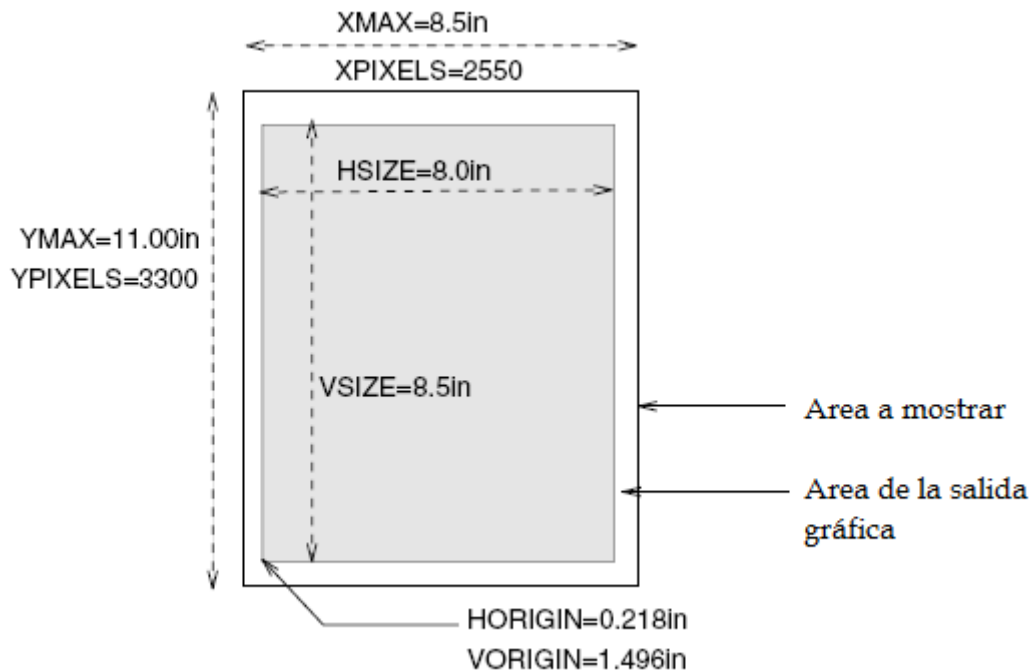


Fig 29. Parámetros del área de salida gráfica

La resolución de una imagen es el número de pixel por pulgada, es determinada por los valores de los parámetros XMAX, YMAX, XPIXELS y YPIXELS del dispositivo, como sigue: $\text{Resolución}_X = \text{XPIXELS} / \text{XMAX}$.

Use la instrucción GOPTIONS para cambiar los valores de XPIXELS= y YPIXELS=, o los valores de XMAX= y YMAX=.

6.4 Procedimientos del SAS/GRAPH

6.4.1 Procedimiento GCHART

Con el procedimiento GCHART se pueden hacer:

- Gráficos de bloques (BLOCK)
- Gráficos de barras horizontales y verticales (HBAR | HBAR3D | VBAR | VBAR3D)
- Gráficos de sectores o torta, de sectores detallados, de sectores 3D, de donas (PIE | PIE3D | DONUT)
- Gráficos de estrellas (STAR).

Estos gráficos representan el valor de un estadístico para una o más variables en un dataset. Estas variables pueden ser cualitativas o cuantitativas.

El procedimiento calcula estas estadísticas: frecuencia absoluta, frecuencia absoluta acumulada, porcentaje, porcentaje acumulado, totales y medias.

Formato general del procedimiento GCHART (para mayor información ver el Manual de Referencia del SAS/GRAPH):

```
PROC GCHART <DATA=dataset> <ANNOTATE=dataset>;
BLOCK variables</ opciones>;
HBAR | HBAR3D | VBAR | VBAR3D variable(s) </ opciones >;
PIE | PIE3D | DONUT variables </ opciones >;
STAR variables </ opciones >;
```

Formato para los gráficos de barras horizontales y verticales con sus respectivas opciones (para mayor información ver el Manual de Referencia del SAS/GRAPH):

```
HBAR | HBAR3D | VBAR | VBAR3D variable(s) </ opciones >;
```

Opciones

se refiere a una o más opciones de algún o todas de las siguientes categorías:

➤ **opciones de apariencia**

ANNOTATE=dataset
 CAUTOREF=color-línea-ref
 CAXIS=color-eje
 CFRAME= color-fondo
 COUTLINE= color-borde-barras | SAME
 CREF= color-línea-ref |(color-línea-ref)|lista color-línea-ref
 CTEXT=color-texto
 FRAME | NOFRAME
 IMAGESTYLE = TILE | FIT
 LEGEND=LEGEND<1...99>
 NOLEGEND
 NOPLANE
 PATTERNID=BY | GROUP | MIDPOINT | SUBGROUP
 SHAPE=figura-barra-3D (solo HBAR3D y VBAR3D)
 WIDTH=ancho-barra
 WOUTLINE=ancho-borde

➤ **Opciones de estadísticos**

CFREQ
 CFREQLABEL='etiqueta-columna' | NONE (solo HBAR y HBAR3D)
 CPERCENT
 CPERCENTLABEL='etiqueta-columna' | NONE (solo HBAR y HBAR3D)
 ERRORBAR=BARS | BOTH | TOP
 FREQ
 FREQLABEL='etiqueta-columna' | NONE (solo HBAR y HBAR3D)
 FREQ=variable-numérica
 G100
 INSIDE=estadístico
 MEAN
 MEANLABEL='etiqueta-columna' | NONE (solo HBAR y HBAR3D)
 NOSTATS (solo HBAR y HBAR3D)
 OUTSIDE= estadístico
 PERCENT
 PERCENTLABEL='etiqueta-columna' | NONE (solo HBAR y HBAR3D)
 PERCENTSUM
 SUM
 SUMLABEL='etiqueta-columna' | NONE (solo HBAR y HBAR3D)
 SUMVAR=variable
 TYPE= estadístico

➤ Opciones de puntos medios

```
DISCRETE  
GROUP=variable-grupo  
LEVELS=número-niveles|ALL  
MIDPOINTS=lista-valores  
MIDPOINTS=OLD  
MISSING  
RANGE  
SUBGROUP=variable-subgrupo
```

➤ Opciones de ejes

```
ASCENDING  
AUTOREF  
AXIS=AXIS<1...99>  
DESCENDING  
FRONTREF (solo HBAR3D y VBAR3D)  
GAXIS=AXIS<1...99>  
MAXIS=AXIS<1...99>  
MINOR=número-menores  
NOAXIS  
NOBASEREF  
NOZERO  
RANGE  
RAXIS=lista-valores | AXIS<1...99>
```

Ejemplo 1:

```
/* Barras Horizontales simple */  
data midata;  
input ITEM $ 1-6 cantidad;  
cards;  
Art. A 10.8  
Art. B 10.5  
Art. C 8.8  
Art. D 4.8  
Art. E 6.2  
Art. F 11.3;  
run;  
  
goptions device=png;  
goptions noborder;  
  
goptions gunit=pct htitle=6 ftitle="arial/bold" htext=4 ftext="arial";  
  
axis1 label=none;
```

```

axis2 label=('Cantidad') order=(0 to 12 by 2) minor=(number=1)
offset=(0,0);

/* pattern v=solid color=red; */
/* pattern v=solid color=green; */
pattern v=solid color=cx43a2ca; /*hex rgb color para azul cielo */

title1 ls=1.5 "Gráfico de Barra Simple";
proc gchart data=midata;
hbar item / discrete type=sum sumvar=cantidad nostats
  maxis=axis1 raxis=axis2
  cref=graycc coutline=black;
run;

quit;

```

Ejemplo 2:

```

/*Barras Horizontales Apareadas */
proc format; picture posval low-high='000,009'; run;

data midata;
input carrera $ 1-14 p_masc p_femen;
format promedio posval.;
grupo='Masculino'; promedio=-1*p_masc; output;
grupo='Femenino'; promedio= p_femen; output;
cards;
Administración 14.1 15.8
Contaduría 13.3 15.1
Economía 13.2 12.8
Estadística 13.8 13.2
;
run;

goptions device=png;
goptions noborder;

goptions gunit=pct htitle=6 ftitle="arial/bold" htext=4.25
ftext="arial/bold";

axis1 label=none;
axis2 label=none order=(-20 to 20 by 5) minor=none offset=(0,0)
value=(h=3pct);

pattern1 v=solid color=blue;
pattern2 v=solid color=red;

legend1 label=none position=(bottom) cframe=white
shape=bar(3,3) cborder=white across=2;

title1 ls=1.5 "Gráfico de Barras Apareados";

proc gchart data=midata;

```

```

hbar carrera / discrete type=sum sumvar=promedio nostats
  subgroup=grupo
  maxis=axis1 raxis=axis2
  autoref clipref cref=graycc
  legend=legend1 coutline=same;
run;

quit;

```

Ejemplo 3:

```

/* Barras Horizontales 3D */
data midata;
input ITEM $ 1-9 AMOUNT;
cards;
Lunes      10.8
Martes     10.5
Miercoles  18.8
Jueves     14.8
Viernes    9.2
Sábado     11.3
Domingo    7.1
;
run;

goptions device=png;
goptions noborder;

goptions gunit=pct htitle=6 ftitle="arial/bold" htext=4 ftext="arial";

axis1 label=none;
axis2 label=('Cantidad') order=(0 to 20 by 2) minor=(number=1)
offset=(0,0);

pattern v=solid color=green;

title1 ls=1.5 "Gráfico de Barra Horizontales 3D";
proc gchart data=midata;
hbar3d item / discrete type=sum sumvar=amount nostats
  maxis=axis1 raxis=axis2
  cref=graycc coutline=black;
run;

quit;

```

Ejemplo 4:

```

/* Grafico de columna simple */

libname c "c:\datos";
filename arch_tex "c:\datos\guarenas.dat";

```

```

data c.lluvia (drop= anno);
infile arch_tex;
input anno 4. (mes1-mes12) (8.1);

proc means;
output out=prom (keep=pmes1-pmes12) mean=pmes1-pmes12;

data grafico (keep = mes Prom_lluvia);
set prom;
array p_mes pmes1-pmes12;
do i=1 to 12;
    mes = i;
    Prom_lluvia= p_mes{i};
    output;
end;

goptions device=png;
goptions noborder;

goptions      gunit=pct      htitle=6      ftitle="arial/bold"      htext=4.25
ftext="arial/bold";

axis1 label=('Mes');
axis2 label=(a=90 'mm3') order=(0 to 150 by 20) minor=none offset=(0,0);

pattern v=solid color=yellow;

title1 ls=1.5 "Lluvia promedio mensual";
title2 ls=1.5 "Gráfico de Columna Simple";

proc gchart data=grafico;
    vbar mes / discrete type=sum sumvar=Prom_lluvia
        maxis=axis1 raxis=axis2
        autoref clipref cref=graycc
        coutline=black width=6 space=2;
/*si se coloca space=0 las barras son conectadas como para histogramas */
run;

quit;

```

Ejemplo 5:

```

/* Gráfico de barras Verticales Agrupado */

libname c "c:\datos";
filename arch_tex "c:\datos\hemoglobina.dat";

data c.hemoglobina;
    infile arch_tex;
    input escuela $ 8 caract $ 12 genero $ 16 n_hemog;

/* Se genera la tabla resumen con el procedimiento FREQ que se utilizará
para generar el gráfico de barras verticales agrupado */

```

```

proc freq data=c.hemoglobina noprint;
  tables escuela * genero/ list noprint out = salida;

goptions device=png;
goptions noborder;

goptions      gunit=pct      htitle=6      ftitle="arial/bold"      htext=4.25
ftext="arial/bold";

axis1 label=none value=none;
axis2 label=(a=90 'fi') order=(0 to 12 by 2) minor=(number=1) offset=(0,0);
axis3 label=('Escuela') offset=(4,4);

legend1 label=none position=(top left inside) cframe=white mode=protect
  shape=bar(3,3) cborder=black across=1;

pattern1 v=solid color=cxbd0026; /* color rojo */
pattern2 v=solid color=cx43a2ca; /* color hex para el azul */

title1 ls=1.5 "Distribución por Escuela y Genero";
title2 ls=1 "Gráfico de Columna Agrupado";

proc gchart data=salida;
vbar genero / discrete type=sum sumvar=count
  group=escuela subgroup=genero
  space=0 /* espacio entre sub-grupo (genero) */
  gspace=4 /* espacio entre grupo (escuela) */
  maxis=axis1 raxis=axis2 gaxis=axis3
  autoref clipref cref=graycc
  legend=legend1 coutline=black;

run;

```

Ejemplo 6:

```

/* Gráfico de barras Verticales Apilado */

axis1 label=('Escuela');
axis2 label=(a=90 'fi') order=(0 to 25 by 5) minor=(number=1) offset=(0,0);

legend1 label=none position=(top left inside) cframe=white mode=protect
  shape=bar(3,3) cborder=black across=1;

pattern1 v=solid color=cxbd0026; /* color rojo */
pattern2 v=solid color=cx43a2ca; /* color hex para el azul */

title1 ls=1.5 "Distribución por Escuela y Genero";
title2 ls=1 "Gráfico de Columna Apilado";

proc gchart data=salida;
vbar escuela / discrete type=sum sumvar=count

```

```

subgroup=genero
maxis=axis1 raxis=axis2
autoref clipref cref=graycc
legend=legend1 coutline=black width=7.5;

```

```
run;
```

Para generar los gráficos de barras en 3 dimensiones cambie “hbar” por “hbar3d” y “vbar” por “vbar3d”.

Formato para los gráficos de torta o sectores y de donas con sus respectivas opciones (para mayor información ver el Manual de Referencia del SAS/GRAPH):

```
PIE | PIE3D | DONUT variable(s) </ opciones>;
```

opciones se refiere a una o más opciones de algunas o todas las siguientes categorías:

➤ **opciones de apariencia:**

```

ANNOTATE=dataset
CFILL=color-relleno
COUTLINE= color-borde | SAME
DETAIL_RADIUS=porcentaje (PIE and DONUT only)
EXPLODE=lista-valores
FILL=SOLID | X
INVISIBLE= lista-valores
NOHEADING
RADIUS=
WOUTLINE=tamaño-borde

```

➤ **opciones de estadísticos**

```

FREQ=variable-numerica
SUMVAR=variable-totalización
TYPE=estadístico

```

➤ **opciones de puntos medios**

```

DISCRETE
LEVELS=número-puntos|ALL
MIDPOINTS= lista-valores
MIDPOINTS=OLD
MISSING

```

➤ **opciones de detalles de sectores (solo PIE y DONUT)**

```

DETAIL=variable
DETAIL_THRESHOLD=porcentaje

```

➤ **opciones de grupo y subgrupo**

ACROSS=número-columnas
 DOWN=número-filas
 GROUP=variable-grupo
 NOGROUPHEADING
 SUBGROUP= variable-subgrupo

➤ **opciones de ordenar los sectores**

ANGLE=grados con respecto a la hora 3
 ASCENDING
 CLOCKWISE
 DESCENDING
 JSTYLE

➤ **opciones de las etiquetas de los sectores**

CTEXT=color-texto
 LEGEND | LEGEND=LEGEND<1...99>
 MATCHCOLOR
 NOLEGEND
 OTHERLABEL='cadena'
 PERCENT=ARROW | INSIDE | NONE | OUTSIDE
 PLABEL=(texto)
 SLICE=ARROW | INSIDE | NONE | OUTSIDE
 VALUE=ARROW | INSIDE | NONE | OUTSIDE

opciones de detalles de las etiquetas de sectores (solo PIE y DONUT)

DETAIL_PERCENT=BEST|NONE
 DETAIL_SLICE=BEST|NONE
 DETAIL_VALUE=BEST|NONE

➤ **opciones de las etiquetas de las donas (solo DONUT):**

DONUTPCT=porcentaje
 LABEL=textos

Ejemplo 7:

```
/* Gráfico de torta 3D Simple */
libname c "c:\datos";
filename arch_tex "c:\datos\hemoglobina.dat";

data c.hemoglobina;
  infile arch_tex;
```

```

input escuela $ 8 caract $ 12 genero $ 16 n_hemog;

/* Se genera la tabla resumen con el procedimiento FREQ que se utilizará
para generar el gráfico de barras verticales agrupado */

proc freq data=c.hemoglobina noprint;
  tables escuela/ list noprint out = salida;

goptions device=png;
goptions noborder;
goptions gunit=pct htitle=6 ftitle="arial/bold" htext=4.25
ftext="arial/bold";

pattern1 v=psolid color=cx9999ff; /* azul claro */
pattern2 v=psolid color=cx993366; /* purpura */
pattern3 v=psolid color=red; /* rojo */
pattern4 v=psolid color=green; /* verde */
pattern5 v=psolid color=cxffffcc; /* amarillo claro */

title1 ls=1.5 "Distribución por Escuela";
title2 "Gráfico de sectores 3D";

proc gchart data=salida;
pie3d escuela / type=sum sumvar=count
  explode="A" /* Resaltar éste sector */
  angle=90 /* 0 = a la hora 3, 90 a las 12, -90 a las 6, 180 a las 9 */
  clockwise /* en sentido agujas del reloj, por defecto anti-horario */
  noheading /* para utilizar los encabezados definidos por el usuario */
  value=none /* donde se coloca el valor ARROW | INSIDE | NONE | OUTSIDE*/
  slice=inside /* donde se coloca la etiqueta ARROW | INSIDE | NONE |
OUTSIDE*/
  percent=arrow /* donde se coloca el porcentaje ARROW | INSIDE | NONE |
OUTSIDE*/
  coutline=black; /* color del borde */
run;

```

6.4.2 Procedimiento GPLOT

Con el procedimiento GPLOT se pueden hacer: diagramas de los valores de dos o mas variables sobre un plano de coordenadas X-Y. Las coordenadas de cada punto sobre el diagrama corresponden a los valores de 2 variables de una observación del "dataset" de entrada. El procedimiento también puede generar un gráfico separado para cada valor de una tercera variable de clasificación, o generar una burbuja cuyo diámetro es proporcional a los valores de una tercera variable.

El procedimiento produce una variedad de diagramas o gráficos de 2 dimensiones:

- Diagrama de dispersión simple (PLOT)
- Diagramas sobre-puestos, en el cual multiples conjuntos de datos se muestran sobre un conjunto de ejes.
- Diagramas sobre un segundo eje vertical (PLOT2)

- Diagramas de burbujas (BUBBLE)
- Diagramas logarítmicos (controlado con la instrucción AXIS)

Formato general del procedimiento GPLOT (para mayor información ver el Manual de Referencia del SAS/GRAPH):

```
PROC GPLOT <DATA=dataset> <ANNOTATE=dataset> <UNIFORM>;
BUBBLE requerimiento-diagrama </ opciones>;
BUBBLE2 requerimiento-diagrama </ opciones>;
PLOT requerimiento-diagrama </ opciones>;
PLOT2 requerimiento-diagrama </ opciones >;
```

Formato para los diagramas de dispersión simple con sus respectivas opciones (para mayor información ver el Manual de Referencia del SAS/GRAPH):

```
PLOT requerimiento-diagrama </opciones>;
```

Opciones se refiere a una o mas opciones de algunas o todas de las siguientes:

➤ **opciones del diagrama:**

```
AREAS=n
GRID
LEGEND | LEGEND=LEGEND<1...99>
NOLEGEND
OVERLAY
REGEQN
SKIPMISS
```

➤ **opciones de apariencia:**

```
CAXIS=color-eje
CFRAME=color-fondo
COUTLINE=color-borde
CTEXT=color-texto
FRAME | NOFRAME
FRONTREF
IFRAME= ref-archivo-externo
IMAGESTYLE = TILE | FIT
NOAXIS | NOAXES
```

➤ **opciones eje horizontal:**

```
AUTOHREF
CAUTOHREF= color-línea-ref
CHREF= color-línea-ref | (color-línea-ref) | lista-color-
```

```

linea-ref
HAXIS=lista-valores | AXIS<1...99>
HMINOR=número-div-menores
HREF= lista-valores
HREVERSE
HZERO
LAUTOHREF=tipo-linea-ref
LHREF= tipo-linea-ref | (tipo-linea-ref) | lista-tipo-linea-
ref

```

➤ **opciones eje vertical:**

```

AUTOVREF
CAUTOVREF= color-linea-ref
CVREF= color-linea-ref | (color-linea-ref) | lista-tipo-
linea-ref LAUTOVREF= tipo-linea-ref
LVREF= tipo-linea-ref | (tipo-linea-ref) | lista-tipo-linea-
ref VAXIS=value-list | AXIS<1...99>
VMINOR= número-div-menores
VREF= lista-valores
VREVERSE
VZERO
WAUTOVREF
WVREF

```

Ejemplo 8: Tomado del SAS/GRAPH® 9.2 Reference, pagina 1375.

```

/* Gráfico de líneas simple */

/* Activa el ambiente gráfico para la pantalla */
options reset=all border;

/* Crea el conjunto de datos del Índice Industrial Dow Jones alto y
   bajo para cuatro décadas, desde 1956 hasta 1995 */

data stocks;
input year high low @@;
cards;
1956 521.05 462.35 1957 520.77 419.79
1958 583.65 436.89 1959 679.36 574.46
1960 685.47 568.05 1961 734.91 610.25
1962 726.01 535.76 1963 767.21 646.79
1964 891.71 768.08 1965 969.26 840.59
1966 995.15 744.32 1967 943.08 786.41
1968 985.21 825.13 1969 968.85 769.93
1970 842.00 631.16 1971 950.82 797.97
1972 1036.27 889.15 1973 1051.70 788.31
1974 891.66 577.60 1975 881.81 632.04
1976 1014.79 858.71 1977 999.75 800.85

```

```

1978 907.74 742.12 1979 897.61 796.67
1980 1000.17 759.13 1981 1024.05 824.01
1982 1070.55 776.92 1983 1287.20 1027.04
1984 1286.64 1086.57 1985 1553.10 1184.96
1986 1955.57 1502.29 1987 2722.42 1738.74
1988 2183.50 1879.14 1989 2791.41 2144.64
1990 2999.75 2365.10 1991 3168.83 2470.30
1992 3413.21 3136.58 1993 3794.33 3241.95
1994 3978.36 3593.35 1995 5216.47 3832.08
;

/* Define títulos, pie de páginas Y simbolos */
title1 "Dow Jones Yearly Highs";
title2 "1956 - 1995";
footnote1 j=1 "Source: 1997 World Almanac"
           j=r " GPLDTPT1 ";
symbol1 interpol=join
value=dot;

/* Genera el diagrama (gráfico), modifica el eje horizontal y define 3
líneas de referencia */

proc gplot data=stocks;
  plot high*year / haxis=1955 to 1995 by 5
  vaxis=0 to 6000 by 1000
  hminor=3
  vminor=1
  vref=1000 3000 5000
  lvref=2;

run;
quit;

```

Ejemplo 9: Tomado del SAS/GRAPH® 9.2 Reference, pagina 1377.

```

/* Gráfico de líneas superpuesto en otro */
/* Usa los datos anteriores */

/* Define títulos, pie de páginas, simbolos, ejes y leyendas */
title1 "Dow Jones Yearly Highs and Lows";
title2 "1956 - 1995";
footnote1 j=1 "Source: 1997 World Almanac";

symbol1 interpol=join
  value=dot
  color=red;

symbol2 interpol=join
  value=C
  font=marker
  color=blue ;

```

```

axis1 order=(1955 to 1995 by 5) offset=(2,2)
      label=none
      major=(height=2)
      minor=(height=1);

axis2 order=(0 to 6000 by 1000) offset=(0,0)
      label=none
      major=(height=2)
      minor=(height=1);

legend1 label=none
      position=(top center inside)
      mode=share;

/* Genera dos diagramas (gráfico) y los muestra sobre el mismo eje */
proc gplot data=stocks;
plot high*year low*year / overlay legend=legend1
      vref=1000 to 5000 by 1000
      lvref=2
      haxis=axis1 hminor=4
      vaxis=axis2 vminor=1;
run;
quit;

```

Ejemplo 10: Tomado del SAS/GRAPH® 9.2 Reference, pagina 1383.

```

/* 3 Gráfico de líneas simultáneos */
/* Coloca las opciones gráficas */

goptions reset=all border;

goptions reset=global gunit=pct border ftext=swissb htitle=6 htext=3;

/* crea el dataset CITYTEMP */
data citytemp;
input date date7. Month season
      f1      /* Raleigh, North Carolina */
      f2      /* Minneapolis, Minnesota */
      f3;     /* Phoenix, Arizona */

/* reestructura los datos de tal forma que haya
una observación para cada ciudad */
drop date season f1-f3;
faren=f1; city='Raleigh'; output;
faren=f2; city='Minn'; output;
faren=f3; city='Phoenix'; output;
cards;
01JAN83 1 1 40.5 12.2 52.1
01FEB83 2 1 42.2 16.5 55.1
01MAR83 3 2 49.2 28.3 59.7
01APR83 4 2 59.5 45.1 67.7
01MAY83 5 2 67.4 57.1 76.3

```

```
01JUN83 6 3 74.4 66.9 84.6
01JUL83 7 3 77.5 71.9 91.2
01AUG83 8 3 76.5 70.2 89.1
01SEP83 9 4 70.6 60.0 83.8
01OCT83 10 4 60.2 50.0 72.2
01NOV83 11 4 50.0 32.4 59.8
01DEC83 12 1 41.2 18.6 52.5
;
run;

/* define títulos y pie de páginas */
title1 'Average Monthly Temperature';
title2 h=4 'Minneapolis, Phoenix, and Raleigh';
footnote1 j=1 ' Source: 1984 American Express';
footnote2 j=1 ' Appointment Book';

/* define las características de los símbolos */
symbol1 color=red interpol=spline width=2 value=triangle height=3;
symbol2 color=blue interpol=spline width=2 value=circle height=3;
symbol3 color=green interpol=spline width=2 value=square height=3;

/* define las características de los ejes */
axis1 label=none
value=('JAN' 'FEB' 'MAR' 'APR' 'MAY' 'JUN' 'JUL' 'AUG' 'SEP' 'OCT'
'NOV' 'DEC')
offset=(2) width=3;

axis2 label=('Degrees' justify=right 'Fahrenheit')
order=(0 to 100 by 10) width=3;

/* genera un gráfico para cada una de las tres ciudades */
/* que produce una leyenda */
proc gplot data=citytemp ;
plot faren*month=city / haxis=axis1
vaxis=axis2
hminor=0
vminor=1
frame;
run;
```

7. Introducción al SAS/IML

[8] El SAS/IML provee acceso a un poderoso y flexible lenguaje de programación en un ambiente iterativo y dinámico. El SAS/IML se puede usar tanto en modo iterativo en el cual se van obteniendo los resultados y respuestas de las instrucciones en el momento o por lote en el cual se almacenan las instrucciones en un programa, el cual se ejecutan más adelante.

Un programa en SAS/IML tiene la siguiente estructura:

```
PROC IML;
  instrucciones SAS/IML;
  :
  :
QUIT;
```

Para invocar al procedimiento SAS/IML en modo iterativo se escribe

```
PROC IML;
```

Y en la ventana LOG aparecerá la siguiente respuesta indicando que IML está listo y esperando instrucciones:

```
IML Ready
```

Los resultados de las instrucciones que se ejecutan en IML aparecerán en la ventana OUTPUT y las respuestas a las mismas aparecerán en la ventana LOG.

El objeto fundamental de este lenguaje es un dato denominado matriz. Una matriz es un arreglo de dos dimensiones (filas y columnas) cuyos elementos pueden ser numéricos o carácter, pero todos del mismo tipo. Estas matrices tienen las siguientes propiedades:

- Los elementos de una matriz numérica son almacenados en doble precisión; los elementos de una matriz carácter son cadenas de caracteres de la misma longitud, esta longitud puede ser de 1 hasta 32676.
- Las matrices son referidas o nombradas por nombres o identificadores válidos SAS. Estos nombres pueden tener de 1 a 8 caracteres de longitud, comienzan con una letra o el carácter de subrayado y el resto pueden ser letras, dígitos o el carácter de subrayado.
- Las matrices pueden contener elementos que representan valores faltantes.
- La dimensión de una matriz está definida por el número de filas y de columnas. Una matriz $Anxm$ tiene nxm elementos arreglados en n filas y m columnas. Los

siguientes son matrices especiales:

- Las matrices $1 \times n$ son llamadas vector fila.
- Las matrices $m \times 1$ son llamadas vector columna.
- Las matrices 1×1 son llamados escalares.

Una matriz literal es una matriz representada por sus valores. Para representar una matriz simplemente se especifican los valores de cada elemento de la matriz. Una matriz literal puede ser un elemento simple (escalar) o varios elementos arreglados en forma rectangular (filas por columnas). Use un par de llaves ({ }) para encerrar todos los elementos de la matriz y comas (,) para separar las filas. La dimensión de la matriz es determinada automáticamente por la forma en que se ingresan los valores.

Los siguientes son ejemplos de matrices introducidas en forma literal:

```
Matriz escalar A1x1
A=2;
```

```
A
2
```

```
Vector fila B1x4
B={1 2 3 4};
```

```
B
1 2 3 4
```

```
Vector columna C4x1
C={8, 1, 3, 9};
```

```
C
8
1
3
9
```

```
Matriz de 4 filas y 4 columnas D4x4
D={8 1 6 0, 1 2 3 4, 1 0 4 2, 7 1 3 4};
```

```
D
8 1 6 0
1 2 3 4
1 0 4 2
7 1 3 4
```

```
Valor faltante
A1=.;
```

```
A1
.
```

```
Matriz de tipo carácter C14x1
C1={"Rancho","Casa","Apartamento","Quinta"};
```

```
C1
Rancho
Casa
Apartamento
Quinta
```

```
Matriz de tipo carácter C21x3
C2={"Pedro" "Luis" "Marcos"};
```

```
C2
Pedro Luis Marcos
```

7.1 Expresiones

Como en todos los lenguajes de programación la manipulación de datos se realiza mediante la asignación. La asignación es el proceso de evaluar una expresión y el resultado de la misma guardarla almacenarla en memoria en forma de datos (matrices en IML).

Las expresiones se forman como la combinación de uno o más operadores con uno o más operandos. Existen tres de tipos de operadores, a saber:

- Operadores *prefijo*, los cuales son colocados antes del operando, por ejemplo el cambio de signo (-A).
- Operadores *infijo*, los cuales son colocados en medio de dos operandos, por ejemplo la suma (A+B).
- Operadores *postfijo*, los cuales son colocados después del operando, por ejemplo la traspuesta (A').

7.2 Función.

Para manipular y/o crear matrices existen en IML funciones.

Una función es un procedimiento o subrutina que regresa un único valor a través del nombre de la misma. En IML las funciones se clasifican en 6 categorías: funciones que dan información de las matrices, funciones escalares, funciones totalizadoras, funciones de aritméticas de matrices, funciones que transforman matrices y funciones de álgebra lineal y estadísticas.

Algunas usan escalares de parámetro, es decir, operan sobre cada elemento de la matriz como la raíz cuadrada o el logaritmo; otros operan sobre la matriz completa como es el caso de la inversa.

```
R=sqrt (B) ;
```



```

R
1  1.4142136  1.7320508          2

X=log(C);

X
2.0794415
0
1.0986123
2.1972246

Y=inv(D);

Y
0.0457516  -0.078431  -0.065359  0.1111111
0.2026144  0.5098039  -0.575163  -0.2222222
0.0718954  0.0196078  0.1830065  -0.1111111
-0.1666667          0  0.1666667  0.1666667

```

Las instrucciones del IML se clasifican en 3 grupos:

- Comandos, son usados para realizar acciones específicas del sistema, tales como almacenar y cargar matrices y módulos o acciones especiales de procesamiento de datos. Entre los comandos tenemos FREE, LOAD, MATTRIB, PRINT, RESET, REMOVE, SHOW y STORE.
- Instrucciones de control, es un conjunto de instrucciones que controlan y llevan el flujo de ejecución. Las instrucciones más usadas en los programas son DO ... END, GOTO, LINK, IF THEN ELSE, PAUSE, QUIT, RESUME, RETURN, RUN, START, FINISH, STOP y ABORT.
- Llamados a subrutinas se realiza con la instrucción CALL.

7.3 Operadores en IML

- + **suma los elementos de matrices**

Formato :

matriz1+matriz2 suma el elemento (i,j) de la matriz1 con el elemento (i,j) de la matriz2 (deben tener el mismo número de filas y columnas).

matriz+escalar suma el escalar a cada elemento de la matriz.

- < > = <= >= ^= **comparación de matrices**

Formato :

matriz1>matriz2 matriz1<matriz2

matriz1=matriz2 matriz1>=matriz2

matriz1<=matriz2 matriz1^=matriz2

compara el elemento (i,j) de la matriz1 con elemento (i,j) de la matriz1 y produce una nueva matriz que contiene 0's donde la comparación de los elemento es falsa y 1's donde la comparación es verdadera.

- **|| concatenación horizontal de matrices**

Formato:

matriz1||matriz2 produce una nueva matriz formada por la unión horizontal de las dos matrices (deben tener el mismo número de filas).

```
A={1 1 1,7 7 7} ;
B={0 0 0,8 8 8} ;
C=A||B ;
```

el resultado es una matriz de 2 filas y 6 columnas :

```
C
1 1 1 0 0 0
7 7 7 8 8 8
```

- **// concatenación vertical de matrices**

Formato:

matriz1//matriz2 produce una nueva matriz formada por la unión vertical de las dos matrices (deben tener el mismo número de columnas).

```
C=A||B ;
```

el resultado es una matriz de 4 filas y 3 columnas :

```
C
1 1 1
7 7 7
0 0 0
8 8 8
```

- **@ producto directo de dos matrices (producto Kronecker)**

Formato:

```
matriz1@matriz2
```

```
A={1 2,3 4};
B={1 1 1,1 1 1};
C=A@B ;
```

el resultado es una matriz cuyo número de filas es igual al producto del número de filas de ambas matrices y el número de columnas igual al producto del número de columnas. Esta nueva matriz es de la siguiente forma:

$$\begin{array}{cccc}
 a_{1,1}B & a_{1,2}B & \dots & a_{1,m}B \\
 a_{2,1}B & a_{2,2}B & \dots & a_{2,m}B \\
 : & : & \dots & : \\
 a_{n,1}B & a_{n,2}B & \dots & a_{n,m}B
 \end{array}$$

```

C
1 1 1 2 2 2
1 1 1 2 2 2
3 3 3 4 4 4
3 3 3 4 4 4

```

- **/ divide los elementos de matrices**

Formato:

matriz1/matriz2 divide el elemento (i,j) de la matriz1 con el elemento (i,j) de la matriz2 (deben tener el mismo número de filas y columnas).

matriz/escalar divide cada elemento de la matriz por el escalar.

- **<> máximo de los elementos de matrices**

Formato :

matriz1<>matriz2 compara cada elemento de la matriz1 con el elemento correspondiente de la matriz2 y produce como resultado el mayor elemento de ambos (deben tener el mismo número de filas y columnas).

- **>< mínimo de los elementos de matrices**

Formato :

matriz1><matriz2 compara cada elemento de la matriz1 con el elemento correspondiente de la matriz2 y produce como resultado el menor elemento de ambos (deben tener el mismo número de filas y columnas).

- **: crea un vector índice**

Formato :

valor1:valor2 crea un vector cuyo primer elemento es el valor1, el segundo elemento es el valor1+1, el tercero valor1+2, etc. El valor1 puede ser menor que el valor2 en cuyo caso el índice es inverso. También trabaja con cadenas de caracteres con sufijo numérico.

```

I=3:8;
R=8:5;
LISTA="var1":"var5";

I
3 4 5 6 7 8

R
8 7 6 5

LISTA
var1 var2 var3 var4 var5

```

- **& | ^ comparación lógica de los elementos de matrices**

Formato:

`matriz1&matriz2` compara el elemento (i,j) de la matriz1 con el elemento (i,j) de la matriz2 (Y lógico ó AND), cada elemento de la nueva matriz es 1 si ambos elementos son diferente de cero y 0 en caso contrario.

`matriz1|matriz2` compara el elemento (i,j) de la matriz1 con el elemento (i,j) de la matriz2 (O lógico ó OR), cada elemento de la nueva matriz es 1 si uno de ellos es diferente de cero y 0 si ambos son cero.

`^matriz2` examina cada elemento de la matriz y produce una nueva matriz con un 0 donde el elemento es diferente de cero y un 1 si es igual a cero.

- **# multiplicación de elementos de matrices**

Formato:

`matriz1#matriz2` produce una nueva matriz formada por la multiplicación de los correspondientes elementos de las matrices).

`matriz#escalar` multiplica cada elemento de la matriz por el escalar.

`Matriz#vector` multiplica los elementos de cada fila de la matriz por el correspondiente elemento del vector (el número de filas de la matriz debe ser igual al número de columnas del vector).

```

A={1 1 1,7 6 5} ;
B={1 2 3,3 5 8} ;
E={5,10} ;
C=A#B ;
D=B#5 ;
F=A#E ;

```

```

C
1 2 3
21 30 40

D
5 5 5
35 30 25

F
5 5 5
70 60 50

```

- *** multiplicación de matrices**

Formato :

matriz1*matriz2 halla la multiplicación de matrices, que consiste en cada fila de una matriz por cada columna de la otra, por lo que el número de columnas de la primera debe ser igual al número de filas de la segunda.

$$A_{n \times p} * B_{p \times m} = C_{n \times m} \text{ produce: } C_{i,j} = \sum_{k=1,p} a_{i,k} * b_{k,j}$$

- **## potencia de los elementos de matrices**

Formato :

matriz1##matriz2 eleva el elemento (i,j) de la matriz1 al elemento (i,j) de la matriz2 (deben ser del mismo orden).

Matriz##escalar eleva cada elemento de la matriz al escalar.

- **** potencia de una matriz**

Formato :

matriz**escalar encuentra la potencia de una matriz es decir : $A^{**4}=A*A*A*A$, la matriz debe ser cuadrada, es decir, igual número de filas y columnas.

- **[] selecciona submatrices**

Formato :

```
matriz[filas, columnas]
matriz[elementos]
```

donde filas y columnas son expresiones que puede ser escalares o vectores, estas expresiones contiene valores válidos de índices de filas y columnas. Más información en el siguiente apartado, *manejo de subíndices*.

```
X={1 2 3, 4 5 6, 7 8 9} ;
Y=X[2,3] ;
Z=X[1:2,2:3] ;
```

esto produce :

```
X
1 2 3
4 5 6
7 8 9
```

```
Y
6
```

```
Z
2 3
5 6
```

- - **cambio de signo de matriz**

Formato :

-matriz produce una matriz donde todos sus elementos tienen el signo cambiado con respecto a la matriz original.

- - **resta los elementos de matrices**

Formato :

matriz1-matriz2 resta al elemento (i,j) de la matriz1 el elemento (i,j) de la matriz2 (deben tener el mismo número de filas y columnas).

Matriz - escalar resta cada elemento de la matriz menos el escalar.

- ` **transpuesta de una matriz**

Formato :

matriz` la transpuesta es una matriz que contiene como filas las columnas de la matriz original y como columnas las filas.

```
A={1 2, 3 4, 5 6};
B=A`;
```

produce:

```
B
1 3 5
2 4 6
```

7.4 Manejo de subíndices

Los subíndices son operadores postfijo especiales, representados por los paréntesis cuadrados ([]) después de una matriz. La forma general es :

matriz[filas, columnas]

donde filas se refiere a una expresión, escalar o vector, para seleccionar una o más filas de una matriz y columnas una expresión, escalar o vector, para seleccionar una o más columnas de una matriz.

Los subíndices se utilizan para :

- extraer un elemento simple de una matriz

`A[2,3]` selecciona el elemento de la fila 2 y columna 3 de la matriz A.

- extraer fila o columna completa de una matriz

`A[3,]` selecciona toda la fila 3 de la matriz A
`A[,2]` selecciona toda la columna 2 de la matriz A

- extraer una submatriz rectangular de una matriz

`A[{1 3},{1 2 5}]` selecciona las filas 1 y 3, y las columnas 1, 2 y 5 de la matriz A.
`A[1:3,2:5]` selecciona las filas de la 1 a la 3 y las columnas de la 2 a la 5 de la matriz A.

- asignar un valor a un elemento de una matriz

`A[3,4]=6` asigna al elemento de la fila 3 y columna 4 el valor 6.

- realizar una reducción de filas o columnas de una matriz, producir una matriz de dimensión reducida. Se utiliza en lugar de subíndices uno de los siguientes operadores de reducción

+ suma
multiplicación
<> máximo
>< mínimo
<:> índice del máximo

>:< índice del mínimo
 : media
 ## suma de cuadrados

```
A={0 1 2, 5 4 3, 7 6 8};
B=A[+,];
C=A[:];
D=A[,##];
```

produce :

A	B	C	D
0 1 2	12 11 13	4	5
5 4 3			50
7 6 8			149

7.5 Imprimiendo matrices con nombres de filas y/o columnas

Se usan las opciones ROWNAME= y COLNAME=, ejemplo:

```
nombres={"Pedro" "Juan" "María" "Luis" "Alberto"};
parcial={"P1" "P2" "P3" "T1" "T2"};
notas={10 12 13 17 14,9 12 17 12 19,12 5 14 17 10,15 16 17 18 16,10 10 11
12 13};

print notas[rowname=nombres colname=parcial];
```

produce la siguiente salida :

NOTAS	P1	P2	P3	T1	T2
Pedro	10	12	13	17	14
Juan	9	12	17	12	19
María	12	5	14	17	10
Luis	15	16	17	18	16
Alberto	10	10	11	12	13

También se puede realizar con la instrucción MATTRIB que permite asociar atributos de impresión a las matrices. El formato general es :

```
MATTRIB        nommatriz        [ROWNAME=vecf]        [COLNAME=vecc]        [LABEL=etiq]
[FORMAT=formato] ;
```

donde *nommatriz* es el nombre de la matriz a imprimir, *vecf* y *vecc* son los vectores que contienen las etiquetas de las filas y columnas, *etiq* es la etiqueta de la matriz y *formato* es el formato de impresión de los elementos de la matriz.

Ejemplo :

```
mattrib notas ROWNAME=nombres COLNAME=parcial LABEL="Notas A-98" FORMAT=5.1;
print notas;
```

produce :

Notas A-98	P1	P2	P3	T1	T2
Pedro	10.0	12.0	13.0	17.0	14.0
Juan	9.0	12.0	17.0	12.0	19.0
María	12.0	5.0	14.0	17.0	10.0
Luis	15.0	16.0	17.0	18.0	16.0
Alberto	10.0	10.0	11.0	12.0	13.0

7.6 Instrucciones de programación

IML es un lenguaje de programación, por lo tanto tiene facilidades que permiten controlar el camino de ejecución mediante instrucciones. Las instrucciones de control tienen en IML la función a las correspondientes instrucciones en el paso de datos SAS. Estas instrucciones son las siguientes :

- Instrucción IF THEN/ELSE

Permite ejecutar una o más instrucciones dependiendo de una condición. La forma general es la siguiente :

```
IF expresión lógica THEN instrucción1 ;
ELSE instrucción2 ;
```

Primero se evalúa la expresión, si esta es verdadera el flujo de ejecución pasa a la instrucción1. Si la expresión es falsa y está presente ELSE se ejecuta la instrucción2, si no está presente pasa a la siguiente instrucción. Ejemplo :

```
if max(a)<20 then p=0 ;
else p=1 ;
```

- Agrupamiento de instrucciones (DO)

Hay ocasiones en que un conjunto de instrucciones debe ser tratadas como si fuera una sola es decir en forma de un bloque, para que esto ocurra marcamos el comienzo del bloque con la instrucción DO y el fin del mismo con la instrucción END. Por ejemplo si en una instrucción IF THEN en caso de que la expresión sea verdadera se debe ejecutar varias instrucciones, éstas se marcan con el DO ... END.

Ejemplo :

```

if x<y then
  do ;
    z1=abs(x+y) ;
    z2=abs(x-y) ;
  end ;
else
  do ;
    z1=abs(x-y) ;
    z2=abs(x+y) ;
  end ;

```

- Instrucciones de repetición

Estas se dividen en tres : DO DATA, DO iterativo, DO WHILE y DO UNTIL.

DO DATA especifica que la iteración se detiene cuando la condición de fin de archivo ocurre, es decir, cuando finalicen los datos. Se puede usar DO DATA para leer de un archivo externo o para procesar los datos de un conjunto de datos SAS.

Formato :

```

DO DATA ;
  :
  :
end ;

```

DO iterativo

Formato :

```
DO variable = comienzo TO final [BY incremento] ;
```

La variable comienza en el valor *comienzo*, éste valor se va incrementando por el valor *incremento* hasta que este sea mayor o igual que *final*.

DO WHILE causa que la expresión sea evaluada antes de cada iteración, si es verdadera continua la ejecución del lazo hasta que sea falsa. Si la expresión es falsa en la primera iteración el lazo no se ejecuta.

Formato :

```

DO WHILE (expresión) ;
  :
  :
end ;

```

DO UNTIL funciona de la misma forma que WHILE solo que la expresión es evaluada al final de la iteración y por lo tanto el lazo es ejecutado al menos una

vez. Al evaluar la expresión si es falsa se vuelve a ejecutar el lazo.

Formato :

```
DO UNTIL (condición) ;
  :
  :
end ;
```

- Salto incondicional

Durante una ejecución normal las instrucciones son ejecutadas una después de la otra. Existen instrucciones en IML que causan que salten inmediatamente a otra instrucción. El lugar al cual la ejecución salta es identificada con una etiqueta, la cual consiste de un nombre seguido por dos puntos colocada antes de la instrucción. Estas instrucciones de salto son GOTO Y LINK.

Formato :

```
GOTO etiqueta ;
LINK etiqueta ;
```

- Definición y ejecución de módulos

Los módulos son usados para crear grupos de instrucciones que pueden ser invocados como una unidad desde cualquier parte de un programa, esto es, un procedimiento o función.

Un módulo siempre comienza con la palabra reservada START y finaliza con FINISH.

```
START nombre [(argumentos)] [global(argumentos)];
  instrucciones ;
  :
  :
FINISH ;
```

Los argumentos son los datos que necesita el módulo y deben ser suministrados cuando se llama o invoca el mismo. Estos deben ir separados por comas.

La cláusula GLOBAL causa que el modulo tome estos datos del programa o módulo principal.

Hay dos formas de ejecutar un módulo :

```
RUN nombre[(argumentos)] ;
```

```
CALL nombre[(argumentos)] ;
```

Los argumentos deben corresponder con los argumentos definidos en el módulo.

Las funciones son módulos especiales que regresan un único dato a través del nombre del mismo. Para crear un módulo tipo función debemos incluir entre las instrucciones del mismo por lo menos una instrucción

```
RETURN (valor) ;
```

Para llamar o invocar una función se realiza mediante una expresión, por ejemplo :

```
start suma(x,y);
  S=x+y ;
  return (s);
finish ;
a={9 4} ;
b={1 8} ;
c=suma(a+b);
d=suma(suma(2,3),suma(5,6));
print c d;
```

esto produce lo siguiente:

C		D
10	12	16

Los módulos creados pueden ser almacenados en un catalogo especial y luego ser recargados a un programa, esto se realiza con las siguientes instrucciones :

```
STORE MODULE = nombre ;
LOAD MODULE = nombre ;
```

Para ver los módulos almacenados utilice la siguiente instrucción :

```
SHOW storage ;
```

- Fin de ejecución

La ejecución de un programa se puede realizar mediante las instrucciones PAUSE, STOP y ABORT. Recordemos que con la instrucción QUIT también podemos parar la ejecución de un programa, pero también causa que el IML se remueva de memoria.

- PAUSE [mensaje] [*];

Esta instrucción para la ejecución de un modulo, imprime el mensaje especificado. La instrucción RESUME permite continuar la ejecución del modulo en el mismo lugar donde se detuvo. Si no se coloca mensaje alguno en la instrucción el sistema envía el siguiente mensaje : *paused in module XXX* . Para

evitar éste mensaje use la opción *.

- STOP ;
Para la ejecución y regresa al modo donde se pueden ingresar nuevas instrucciones.
- ABORT ;
Para la ejecución y se sale del IML, esta acción es parecida a la instrucción QUIT, con la excepción que ABORT es ejecutable y programable.

7.7 Trabajando con conjunto de datos SAS

El IML tiene algunas instrucciones que permiten pasar datos de un conjunto de datos SAS a matrices y viceversa. Se pueden crear matrices de las variables y observaciones de un conjunto de datos SAS ; se pueden crear vectores columna para cada variable de un conjunto de datos SAS. Se pueden usar todas las observaciones o un subconjunto de ellas. Se puede crear un conjunto de datos SAS con los elementos de una matriz, en este caso las columnas corresponden a variables y las filas a observaciones.

Para acceder un conjunto de datos SAS se debe primero abrir el mismo. En IML existe tres formas de abrir un conjunto de datos :

- USE nombre [VAR {variables}] [WHERE(expresión)];
 nombre es el conjunto de datos SAS que se desea leer, puede ser de dos niveles, es decir, librería.nombre.
 VAR especifica las variables que se van a leer del conjunto de datos SAS, si no se especifica se toman todas.
 WHERE funciona como un filtro para leer observaciones, es decir, se procesaran las observaciones que cumplan con la condición.
 Esta instrucción se usa simplemente para leer un conjunto de datos ya existente, con alguna de las siguientes instrucciones FIND, INDEX, LIST y READ.
- EDIT nombre [VAR {variables}] [WHERE(expresión)];
 nombre es el conjunto de datos SAS.
 VAR especifica las variables que se van a procesar del conjunto de datos SAS, si no se especifica se toman todas.

WHERE funciona como un filtro para procesar observaciones.

Esta instrucción se usa para leer y escribir de un conjunto de datos ya existente, con alguna de las siguientes instrucciones FIND, INDEX, LIST y READ para leer ; y para escribir con REPLACE, APPEND, DELETE y PURGE.

- CREATE nombre [VAR variables] ;
 CREATE nombre FROM matriz [COLNAME= ROWNAME=];
 nombre es el conjunto de datos SAS.
 VAR especifica las variables IML existentes con las cuales se creará el conjunto de datos SAS.
 FROM se especifica una matriz que contiene los datos.
 COLNAME= un vector carácter que contiene los nombres de las variables que se crearán en el conjunto de datos SAS.
 ROWNAME= un vector carácter que contiene los nombres de las observaciones.

Use la instrucción APPEND para colocar los datos en las matrices en el conjunto de datos recientemente creado.

Una vez abierto el conjunto de datos a procesar, dependiendo del caso (leer o escribir observaciones) se utiliza uno de las siguientes instrucciones :

- APPEND [VAR variables] ;
 APPEND [FROM matriz] ;
 Use esta instrucción para añadir datos al final del conjunto de datos SAS actual. Las observaciones añadidas provienen o de las variables especificadas en la cláusula VAR o variables creadas de las columnas de la matriz de la cláusula FROM (se debe especificar sólo una de las cláusulas VAR o FROM).
 Si en la instrucción EDIT o CREATE se especifica o la cláusula VAR o la FROM en la instrucción APPEND no es necesario colocarlas.

Ejemplo :

```
X={1 2 3, 4 5 6, 7 8 9} ;
CREATE MISDATOS FROM X [COLNAME={X1 X2 X2}] ;
APPEND FROM X;
```

Se crea un conjunto de datos con tres variables (X1, X2 y X3) y tres observaciones.

- CLOSE [conjunto de datos] ;
 Es usada para cerrar uno o más conjunto de datos SAS abiertos con USE, EDIT o CREATE. Si no se especifica el

conjunto de datos se cierra el actual.

- **DELETE** [rango] [WHERE (condición)];
 Use esta instrucción para marcar registros a ser borrados en el conjunto de datos actual de salida (abierto con EDIT). Para borrar los registros y reenumerar el resto de las observaciones use la instrucción PURGE.
 Rango especifica el rango de las observaciones a borrar, puede ser :
 ALL todas las observaciones.
 CURRENT la observación actual (valor por defecto).
 NEXT [n] la próxima observación o las proximas n observaciones.
 AFTER todas las observaciones después de la actual.
 POINT n especifica un apuntador a uno o más registros.
 WHERE especifica un filtro sobre las observaciones a marcar como para eliminar.
- **FIND** [rango] [WHERE (condición)] INTO matriz ;
 Encuentra los números de observaciones de los registros especificados en el rango y que cumple con la cláusula WHERE y los coloca en la matriz numérica.
- **LIST** [rango] [VAR variables] [WHERE (condición)] ;
 Muestra observaciones de un conjunto de datos SAS.
- **PURGE** ;
 Elimina las observaciones marcadas para borrar y reenumera los registros restantes.
- **READ** [rango] [VAR variables] [WHERE (condición)] INTO matriz.
 Lee variables o registros del actual conjunto de datos SAS abierto con las instrucciones USE o EDIT en forma de matrices columnas especificadas en la cláusula VAR o en la matriz especificada en la cláusula INTO.
- **REPLACE** [rango] [VAR variables] [WHERE (condición)] ;
 Reemplaza los valores de las observaciones en el conjunto de datos SAS con los valores actuales de las matrices IML con el mismo nombre. Use el rango, las cláusulas VAR y WHERE para limitar los reemplazos de las variables y/o observaciones.
- **INDEX** variables
 Crea un índice para las variables nombradas en el conjunto de datos SAS actual de salida (abierto con las instrucciones USE y EDIT)

7.8 Acceso a archivos externos

Los datos que se usan en el IML pueden ser leídos desde archivos externos en forma de texto o binarios, también pueden escribirse datos en archivos externos en forma de texto. Para realizar esto se debe declarar los archivos externo con las instrucciones INFILE o FILE y luego se accesan los datos (leer o escribir), usando las instrucciones INPUT o PUT.

- `INFILE` archivo [opciones] ;
 Abre un archivo externo para entrada de datos.
 archivo se refiere o al nombre de un archivo con nombre y extensión encerrados entre apóstrofes o a una referencia previa a un archivo definida con la instrucción `FILENAME`.
 Las opciones disponibles para ésta instrucción son :
 LENGTH= especifica una variable, en la cual IML almacenará la longitud del registro leído.
 RECFM=N especifica que el archivo a ser leído como binario puro.
 FLOWOVER permite a la instrucción `INPUT` ir al próximo registro para obtener valores para las variables.
 MISSOVER asigna valores faltantes a las variables leídas después del fin del registro.
- `FILE` archivo [recfm=N] [LRECL=] ;
 Abre un archivo externo para salida.
 archivo se refiere o al nombre de un archivo con nombre y extensión encerrados entre apóstrofes o a una referencia previa a un archivo definida con la instrucción `FILENAME`.
 RECFM=N especifica que el archivo a ser leído como binario puro.
 LRECL= especifica la longitud del registro del archivo de salida, por defecto la longitud el registro es 512.
- `INPUT` [variables] [formato] [directivas del registro] [posición] ...
 Esta instrucción permite leer registros del actual archivo de entrada definido con la instrucción `INFILE`, colocando los valores dentro de variables IML.
 variables especifica la variables o variables que se leeran en la posición actual del registro. Cada variable puede ser seguida de una

- especificación de formato.
- formato especifica el formato de entrada de una variable, estas son las mismas que se usaron en el paso de datos.
- directivas especifica la forma de avanzar a un nuevo registro (ver @ y / en el paso de datos).
- posición para ir a una columna especificada en el registro (ver @ y + en el paso de datos).
- PUT dato [directiva del registro] [posición] [formato] ...
 - dato especifica el valor a imprimir en la posición actual del registro. Puede ser el nombre de una variable (escalar), un valor literal (constante) o una expresión entre paréntesis.
 - formato especifica el formato de entrada de una variable, estas son las mismas que se usaron en el paso de datos.
 - directivas especifica la forma de avanzar a un nuevo registro (ver @ y / en el paso de datos).
 - posición para ir a una columna especificada en el registro (ver @ y + en el paso de datos).
 - CLOSEFILE archivos.
 - Es usado para cerrar los archivos abiertos por las instrucciones INFILE y FILE.
 - SHOW FILES.
 - Muestra todos los archivos abiertos.

Ejemplos:

Imprime al archivo c:\datos\salida.dat los valores entre 1 y 10 con sus valores respectivos de la raíz cuadrada, cuadrado y el logaritmo natural.

```
proc iml;
X=repeat(0,10,4);
Mattrib X format=10.5;
do i=1 to 10;
  X[i,1]=i;
  x[i,2]=sqrt(i);
  x[i,3]=i**2;
  x[i,4]=log(i);
end;
filename archivo 'c:\datos\salida.dat';
file archivo;
do i=1 to nrow(x);
  put @1 (x[i,1]) +5 (x[i,2]) +5 (x[i,3]) +5 (x[i,4]);
```

```
end;
closefile archivo;
```

Crea un conjunto de datos SAS llamado clase con los datos que están en un archivo externo denominado c:\datos\salida.dat.

```
Filename nclase 'c:\datos\clase.dat' ;
infile clase misover ;
create c.clase var{nombre sexo edad peso estat} ;
do data ;
  input nombre $ sexo $ edad peso estat ;
  append ;
end ;
close c.clase ;
closefile nclase ;
```

7.9 Otras instrucciones, funciones y procedimientos del IML

7.9.1 Funciones que invocan matrices

ALL	chequea para todos los elementos no ceros de la matriz
ANY	chequea para algún elemento no cero de la matriz
LOC	encuentra elementos no cero de una matriz
NCOL	número de columnas de una matriz
NLENG	tamaño de un elemento carácter
NROW	número de filas de una matriz
SHOW ALLNAMES	muestra todos los atributos de todos los nombres con o sin valores
SHOW NAMES	muestra los atributos de los nombres que tienen valores
SHOW name	muestra los atributos de una matriz
TYPE	determina el tipo de una matriz

7.9.2 Funciones aritméticas de matrices

CUSUM	suma acumulada de una matriz
HDIR	producto horizontal directo
TRACE	traza de una matriz (suma de los elementos de la diagonal)

7.9.3 Funciones escalares

ABS	valor absoluto
ARCOS	arco coseno

ARSIN	arco seno
ATAN	arco tangente
COS	coseno
EXP	calcula el exponencial e^x
INT	parte entera (trunca)
LOG	logaritmo natural o neperiano
MOD	modulo o resto de una división entera
NORMAL	número aleatorio normal estándar.
SIN	seno
SQRT	raíz cuadrada
TAN	tangente
UNIFORM	número aleatorio uniforme entre 0 y 1

7.9.4 Funciones de reducción

MAX	máximo valor de una matriz
MIN	mínimo valor de una matriz
SSQ	suma de cuadrados de los elementos de una matriz
SUM	suma de los elementos de una matriz

7.9.5 Funciones de manipulación de matrices

BLOCK	crea una matriz bloque diagonal
BTRAN	transpuesta por bloques
DIAG	crea una matriz diagonal
I	crea una matriz identidad
INSERT	inserta una matriz dentro de otra
J	Crea una matriz de valores identicos
REMOVE	elimina elementos de una matriz
REPEAT	crea una matriz de valores repetidos
SHAPE	reforma y repite valores
SQRSYM	convierte una matriz simétrica a una cuadrada
SIMSQR	convierte una matriz cuadrada a una simétrica
T	transpuesta de una matriz
VECDIAG	crea un vector columna de elementos diagonales

7.9.6 Funciones para manipulación de caracteres

BYTE	el carácter ASCII de un número
CHANGE	busca y reemplaza un texto en una cadena
CHAR	produce la representación carácter de una expresión numérica
CONCAT	concatenación de cadenas de caracteres
CSHAPE	reforma y repite valores carácter
LENGTH	encuentra la longitud de los elementos de una matriz carácter
NAME	lista los nombres de argumentos
NUM	produce una representación numérica de una matriz carácter
SUBSTR	toma subcadena de elementos carácter

7.9.7 Comandos y funciones de manejo de datos

APPEND	añade registros al final de un conjunto de datos SAS
CLOSE	cierra un conjunto de datos SAS
CLOSEFILE	cierra un archivo externo
CONTENTS	obtiene las variables de un conjunto de datos SAS
CREATE	crea un conjunto de datos SAS
DATASETS	obtiene los nombres de los dataset SAS en una librería
DELETE (proc)	borra un conjunto de datos SAS
DELETE	marca registros como borrados en un conjunto de datos SAS
EDIT	abre un conjunto de datos SAS para acceso de entrada/salida
FILE	abre un archivo externo para salida
FIND	encuentra registros en un conjunto de datos SAS
INFILE	abre un archivo externo (texto) para entrada
INPUT	lee datos de un archivo externo
LIST	muestra registros
PURGE	elimina registros de un dataset SAS previamente marcados
PUT	escribe datos a un archivo externo
READ	lee datos
RENAME	renombra un conjunto de datos SAS
REPLACE	reemplaza datos en un conjunto de datos SAS
SETIN	selecciona un conjunto de datos SAS abierto, para entrada
SETOUT	selecciona un conjunto de datos SAS abierto, para salida
SHOW CONTENTS	muestra el contenido del conjunto de datos SAS activo
SHOW DATASETS	muestra los conjunto de datos SAS activos

STORAGE	regresa el nombre de las matrices abiertas en el área de almacenamientos
USE	abre un conjunto de datos SAS para lectura

7.9.8 Funciones algebraicas lineales y estadísticas

DESIGN	crea matrices de diseño
DESIGNF	crea matrices de diseño de rango completo
DET	determinante de una matriz cuadrada
ECHELON	reduce una matriz a la forma normal fila-echelon
EIGEN	calcula y autovectores
EIGVAL	calcula autovalores
EIGVEC	crea autovectores
GINV	inversa generalizada
GSORTH	factorización ortonormal de Gram-Schmidt
HALF	descomposición de Cholesky
HERMITE	reduce una matriz a la forma normal Hermite
HOMOGEN	resuelve un problema lineal homogéneo
INV	inversa de una matriz
INVUPDT	actualiza una matriz inversa
IPF	ejecuta un ajuste proporcional iterativo
LCP	resuelve un problema lineal complementario
LP	resuelve un problema de programación lineal
MARG	evalúa totales marginales en una tabla de contingencia multivía
ORPOL	genera polinomios ortogonales
POLYROOT	encuentra ceros de un polinomio real
PRODUCT	multiplica matrices de polinomios
RANK	matriz con el orden de los elementos
RADIO	divide matrices de polinomios
ROOT	descomposición de Cholesky
SOLVE	resuelve un problema de ecuaciones lineales

Ejemplo: Regresión múltiple.

```
options nocenter ;

/* Se crean los datos un dataset */
data test ;
  input y x0 x1 x2 ;
cards;
7 1 2 1
15 1 4 4
20 1 3 7
15 1 6 3
12 1 5 2
8 1 1 2
;
run;

proc iml;
  use test;
  read all ;      /* Se leen los datos del dataset a y, x0, x1 y x2 */

  x = x0 || x1 || x2;
  print x;
  b = INV(x`*x)*(x`*y);
  yp = x * b ;
  ye = y - yp ;
  print "variables independientes";
  print x;
  print "variable dependiente";
  print y;
  print "estimación de parámetros";
  print b;
  print "predicción del valor de Y";
  print yp ;
  print "error en la predicción de Y";
  print ye ;
  create c var {parameter} ;
  append from b;
  quit;
proc print data = c;
run;
```

8. Programación con macros en SAS

[8] y [9] La facilidad macro es una herramienta para extender y simplificar la programación SAS, reduciendo la cantidad de texto para realizar tareas comunes. Permite asignar un nombre (<nombre-macro> o <macro-variable>) a una cadena de caracteres o un conjunto de instrucciones sas, de tal forma que puede usar el texto como si fuesen las mismas instrucciones sas.

La facilidad macro tiene dos componentes:

- El macro procesador, que es la parte del sas que realiza las tareas
- El macro lenguaje, que es la sintaxis que se usa para comunicarse con el macro procesador.

8.1 Macros

Una macro es un conjunto de <instrucciones y comandos sas> y de <instrucciones macro>, que permite ejecutar una tarea o parte del programa varias veces, siendo referenciadas desde cualquier parte del programa. Es mucho más que los módulos, subprogramas o procedimientos en otros lenguajes de programación. Las macros permiten sustituir texto en un programa y hacer muchas otras cosas.

Las <instrucciones macro> comienzan con los símbolos % y adicionalmente se usa el símbolo & para definir variables macro y referenciar los parámetros dentro de la macro.

Una macro se crea de la siguiente forma:

```
%MACRO <nombre-macro> [(par1, par2, ...)];
    <sentencia SAS> y/o <sentencia macro>
    <sentencia SAS> y/o <sentencia macro>
    :
    :
%MEND <nombre-macro>;
```

El <nombre-macro> de la definición (%MACRO) debe coincidir con el de la instrucción de fin de macro (%MEND)

Los corchetes [] significa que lo está dentro es opcional, es decir, que una macro puede o no tener parámetros.

Los parámetros son macro-variables locales que tienen validez solo en la macro, se referencian anteponiendo el símbolo &.

La llamada a una macro se hace desde cualquier parte del programa pero después que se haya declarado la misma de la siguiente forma:

```
%<nombre-macro> [(valor1, valor2, ...)];
```

Las macros al igual que los programas sas pueden contener comentarios, estos pueden ser de dos formas:

- Encerrados entre `/* y */`

```
/* esto es un comentario que puede
   Ser de varias líneas                */
```

- A continuación de `%*` y finalizados con un `;`

```
%* esto es un comentario de una línea ;
```

Ejemplo: la siguiente macro ordena el dataset que viene en el parámetro “datos” por la variable que viene en el parámetro “x” y luego calcula las estadísticas básicas para los diversos valores de “x”.

```
%MACRO est_bas(datos, x);
  Proc sort data=&datos;
  By &x;

  Proc means data=&datos;
  By &x;

  Run;
%MEND est_bas;
```

La llamada sería de la siguiente forma:

```
/* Calcula las estadísticas básicas por sexo */
%est_bas(alumnos, sexo);

/* Calcula las estadísticas básicas por escuela */
%est_bas(alumnos, escuela);
```


Al ejecutarse el código anterior, se produce el siguiente programa:

```
/* Calcula las estadísticas básicas por sexo */
Proc sort data=alumnos;
By sexo;

Proc means data=alumnos;
By sexo;

Run;

/* Calcula las estadísticas básicas por escuela */
Proc sort data=alumnos;
By escuela;

Proc means data=alumnos;
By escuela;

Run;
```

8.2 Macro-variables

Una macro-variable es una variable que tiene validez o ámbito durante cualquier paso de datos (DATA) o paso de procedimientos (PROC), es una herramienta que le permite modificar dinámicamente un texto en un programa SAS a través de sustitución simbólica. Se puede asignar grandes o pequeñas cantidades de texto a las macro-variables y después ese texto se usa por una referencia simple de la variable que lo contiene.

Las macro-variables pueden ser globales o locales, las globales tienen validez en todo el programa desde su creación, y las locales son las que se crean dentro de una macro y tienen validez solo durante la ejecución de la macro.

Las macro-variables se crean de la siguiente forma:

```
%let <macro-variable> [= valor inicial];
```

Una macro-variable lo que hace es sustituir el valor que tiene asignado en el lugar en que se hace referencia a la misma como si fuera un literal.

A una macro-variable se le puede asignar varias instrucciones sas, lo cual se realiza con la instrucción %str de la siguiente forma:

```
%let <macro-variable> = %str( sentencias sas );
```

El valor que toma una macro-variable es de tipo carácter y la longitud puede ser hasta 65534 caracteres entre letras, dígitos y caracteres especiales a excepción del % y &. Si se omite el valor inicial la longitud será cero (vacío). Si una macro-variable ya existe, el valor asignado reemplaza al valor anterior.

Para almacenar un valor a una macro-variable no es necesario colocar delimitadores, por lo tanto, en forma directa solo se puede almacenar una cadena simple. Si se colocan los delimitadores, estos formarían parte del valor de la macro-variable. En el siguiente ejemplo los tres casos producen el mismo valor *Gabriel*, los espacios en blanco no son almacenados.

```
%let nombre = Gabriel    ;
%let nombre =           Gabriel;
%let nombre =      Gabriel  ;
```

Si se desea una cadena con espacios y otros símbolos se usa la función %str y %nrstr, ésta última permite que los caracteres % y & formen parte del valor de la macro-variable.

```
%let state=%str( North Carolina);
%let town=%str(Taylor%'s Pond);
%let store=%nrstr(Smith&Jones);
%let plotit=%str(proc plot;
                  plot income*age;
                  run;
                  );
```

Para usar un delimitador como parte del valor de una macro-variable se debe usar anteponiendo el carácter %, como en la variable town del ejemplo anterior.

Si se asigna puros dígitos estos no son considerados como número sino caracteres. En caso que se intente almacenar una expresión ésta no es evaluada. Utilice las funciones %eval y %sysevalf para evaluar expresiones, la segunda para evaluar punto flotante.

```
%let num=123;
%let total=100+200;          /* produce 100+200   */
%let total=%eval(100+200);  /* produce 300   */
%let total=%sysevalf(100.2+200.125); /* produce 300.325 */
```

Se puede utilizar una referencia a una macro-variable:

```
%let n1=123;
%let n2=202;
%let n3=&n1 + &n2 = %eval(&n1+&n2); /* Produce 123 + 202 = 325
*/
```

8.3 Macro-variables definidas por el SAS

Cuando se invoca el SAS, el macro procesador crea automáticamente macro-variables que suplen información relacionada con la sesión de trabajo. Para usar una de estas macro-variables simplemente se le antepone el símbolo “&” al nombre de la misma. El siguiente ejemplo muestra un pie de página y utiliza las macro-variables SYSDATE y SYSDATE9:

```
Footnote "Reporte realizado el &sysdate, &sysdate9";
```

Si la sesión actual fue invocada el sábado 16 de julio del 2016, producirá la siguiente instrucción:

```
FOOTNOTE "Reporte realizado el Saturday, 16JUL2016";
```

Macro-variable del SAS	Contenido
SYSCMD	El último comando reconocido de la línea de comando o de la ventana macro
SYSDEVICE	Nombre del actual dispositivo gráfico
SYSDSN	Nombre del dataset más reciente en dos campos
SYSLAST	Nombre del dataset más reciente en un campo
SYSDATE	La fecha de la sesión de SAS en formato de 2 dígitos para el año
SYSDATE9	La fecha de la sesión de SAS en formato de 4 dígitos para el año
SYSDAY	El día de la semana de la sesión de SAS
SYSTIME	La hora de la sesión de SAS

Tabla 6. Macro-variables automáticas del SAS

8.4 Instrucciones macro

8.4.1 Repita para

```
%DO <macro-variable>= N1 %TO N2 [%BY N3];  
    <sentencia SAS> y/o <sentencia macro>  
    <sentencia SAS> y/o <sentencia macro>  
    :  
%END
```

En la definición del repita o bucle %DO la macro-variable no lleva el símbolo &, pero dentro del %DO ... %END si se le antepone el ¯o-variable.

8.4.2 Repita mientras/hasta

```
%DO &WHILE | &UNTIL <condición>;  
    <sentencia SAS> y/o <sentencia macro>  
    <sentencia SAS> y/o <sentencia macro>  
    :  
%END
```

8.4.3 Si condicional

```
%IF <condición> acción1;  
  
[%ELSE acción2;]
```

Cuando acción1 o acción2 se trata de mas de una sentencia se utiliza %DO y %END para marcar el comienzo y fin del mismo.

```
%DO;  
    <sentencia SAS> y/o <sentencia macro>  
    <sentencia SAS> y/o <sentencia macro>  
    :  
%END;
```

8.4.4 Salto incondicional

```
%GOTO <etiqueta>;
```

En cualquier parte de la macro debe estar definida la etiqueta:

```
%<etiqueta>: sentencia;
```

8.4.5 Lectura de una macro-variable

```
%INPUT <macro-variale>;
```

8.4.6 Crear una <macro-variable> local

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

8.4.7 Escribir un texto

```
%PUT texto;
```

BIBLIOGRAFÍA

- [1] Pérez López, C (2001). El Sistema Estadístico SAS. España. Prentice Hall.
- [2] Rebolledo Robles, H. H. (2002) Manual SAS por computadora. Editorial Trillas.
- [3] SAS 9.1.3 Language Reference: Concepts
http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_913/base_1rconcept_9196.pdf
- [4] Torres Rivas, E (2014). Computación Estadística. IEAC - FACES - ULA.
- [5] Base SAS 9.1.3 Procedures Guide, Second Edition
http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_913/base_proc_8977.pdf
- [6] SAS/STAT User's Guide
http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_91/stat_ug_7313.pdf
- [7] SAS/GRAPH 9.2: Reference, Second Edition
<http://support.sas.com/documentation/cdl/en/graphref/63022/PDF/default/graphref.pdf>
- [8] SAS/IML 9.1: User's Guide
http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_91/iml_ug_7306.pdf
- [9] SAS 9.1 Macro Language: Reference
http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_91/base_macro_6997.pdf