INTRODUCCION AL MICRO SAS:

APLICACION AL ANALISIS DE EXPERIMENTOS AGRICOLAS

Gustavo López Pérez
Javier López Pérez

Unidad de Informática y Bioestadística

CENTRO AGRONOMICO TROPICAL DE
INVESTIGACION Y ENSEÑANZA

Turrialba, Costa Rica
Diciembre 1995

El CATIE es una institución de carácter científico y educacional cuyo propósito fundamental es la investigación y enseñanza de posgrado en el campo de las ciencias agropecuarias y de los recursos naturales renovables aplicados al trópico americano.

TABLA DE CONTENIDO

página No.

INTRODUCCION

Este documento fue diseñado como manual de referencia para quienes deseen iniciarse en el uso del sistema SAS para micro-computadoras. Uno de los problemas que tienen los usuarios de SAS es la carencia de documentos de referencia en español que sean fáciles de entender por personas que no tienen conocimientos amplios en programación y estadística. Este documento trata de llenar en parte esta necesidad.

El documento se refiere al uso del SAS a través del Manejador de Pantallas, por lo cual se explica sobre su uso; también cubre la programación del Paso DATA y algunos de los procedimientos estadísticos para análisis de datos experimentales.

Este documento se desarrolla en seis capítulos:

En el Capítulo I se explica el ambiente de ventanas que utiliza el SAS para permitir a los usuarios la interacción con el sistema. En el capítulo II se exploran las facilidades y comandos del Editor de SAS para la entrada de datos y programas, y la ejecución de programas SAS. En el Capítulo III se explican los conceptos de archivos ASCII y SAS que puede interpretar el sistema. También se estudia la estructura de un programa SAS.

En el Capítulo IV se estudian las instrucciones más importantes del SAS para la programación del Paso DATA y se incluyen algunos ejemplos de programas para manipular datos. En el Capítulo V se estudian algunos de los procedimientos para organizar archivos SAS, controlar las salidas y realizar cálculos de estadísticas descriptivas.

Finalmente, en el Capítulo VI se estudian algunos procedimientos para analizar datos de experimentos. Se dan ejemplos de programas, sus salidas y la interpretación de éstas.

Con este documento no se pretende reemplazar los manuales de documentación del Software SAS. Si algún usuario quiere profundizar más en alguno de los temas desarrollados en este documento, debe consultar los manuales SAS respectivos.

¿Qué es SAS?

Statistical Analysis System (SAS) es un conjunto de programas que proveen una variedad de capacidades para procesamiento y análisis de datos, las cuales incluyen:

- Recuperación: Técnicas flexibles de entrada de datos
- Transformaciones: Lenguaje de programación con funciones estadísticas, matemáticas, de fechas, etc.
- Mantenimiento: Almacenamiento, documentación, actualización y edición de archivos
- Manipulación: Clasificación, escogencia y concatenación de archivos
- Informes: Impresión de información utilizando instrucciones

- Gráficos: Histogramas, gráficas, cuadros de dos dimensiones, etc.
- Análisis Estadístico: Estadísticas descriptivas, inferencial, análisis multivariado, etc.

Otros productos SAS son:

- SAS/GRAPH (Gráficos de alta calidad)
- SAS/ETS (Series de tiempo y econometría)
- SAS/IML (Manipulación de matrices)

CAPITULO I. El Manejador de Pantallas

Para las versiones de Microcomputadoras, el usuario interactúa con el Sistema SAS a través del manejador de pantallas.

El manejador de pantallas consiste en un ambiente de ventanas creado con el objetivo de desarrollar interactivamente una sesión de SAS.

Para iniciar una sesión de SAS, se invoca el sistema con el comando SAS. Para proteger los archivos que componen el SAS, es recomendable ejecutar el sistema desde un directorio de trabajo. Nunca se debe trabajar SAS en la raíz del directorio que contiene el Sistema.

1.1 Partes de una ventana

Cada ventana se compone de:

- un nombre.
- un área de texto, donde se despliega el contenido de la ventana o se ingresan datos o programas.
- una línea de comandos, donde se emiten comandos del manejador de pantallas.
- una línea de mensajes, donde aparecen mensajes de error y notas del Sistema SAS.

1.2 Ventanas del manejador de Pantallas

Las siguientes son las ventanas primarias de que dispone SAS:

1.2.1 La ventana PROGRAM EDITOR

Esta ventana es la que utiliza el SAS como editor. En ella se pueden introducir y modificar archivos, que por lo general contienen datos o programas SAS.

Los programas SAS normalmente se ejecutan desde esta ventana.

1.2.2 La ventana LOG

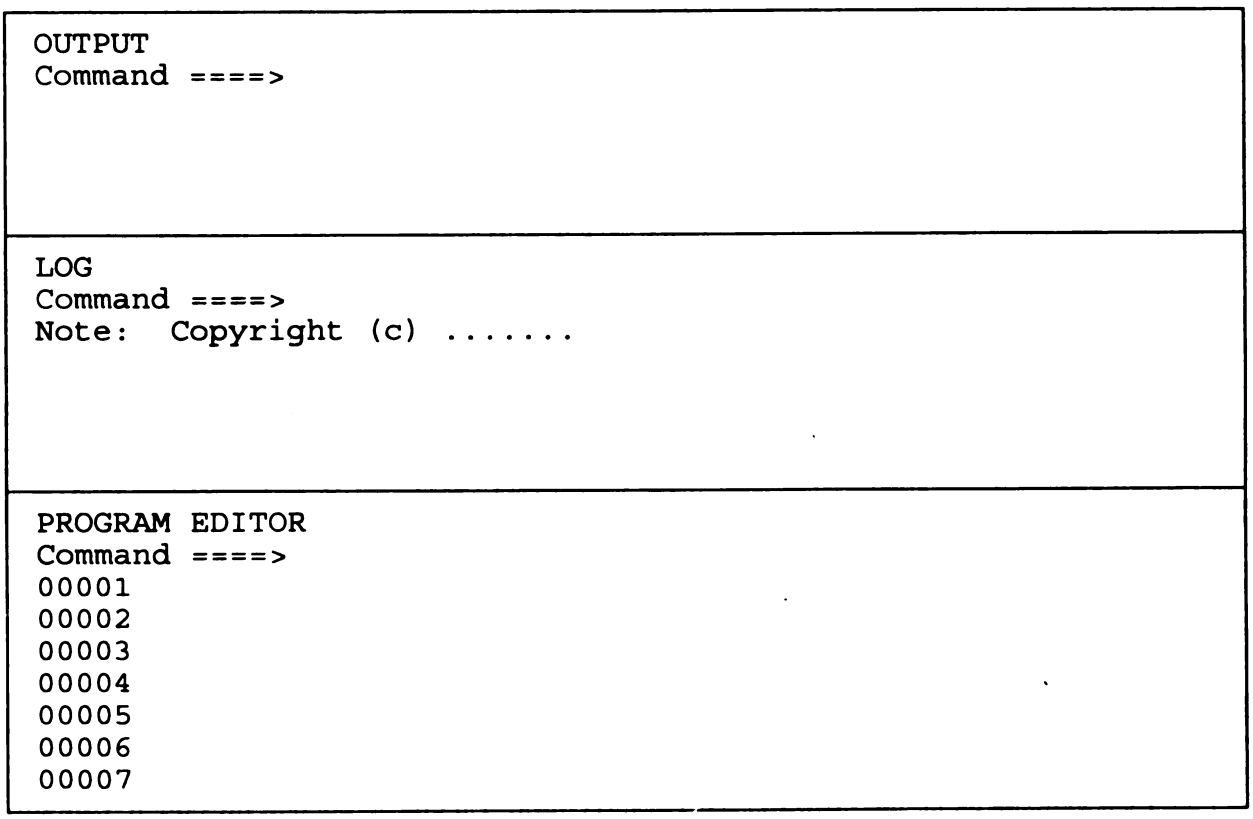
En esta ventana se despliegan todos los mensajes sobre la ejecución de los programas. La información que el sistema despliega en esta ventana incluye: mensajes de error, características de los archivos que trabaja el programa (nombre, variables, observaciones) y tiempos de duración en la ejecución de las diferentes partes del programa.

1.2.3 La ventana OUTPUT

En esta ventana se despliegan los resultados (salidas) que producen los diferentes procedimientos utilizados en el programa que se ejecuta. Si el programa consiste solo de pasos DATA, en esta ventana no se despliega nada.

Además, existen ventanas especiales (KEYS, HELP, LIB, etc) que pueden activarse con los comandos correspondientes.

Cuando se entra al sistema SAS, las ventanas PROGRAM EDITOR, LOG y OUTPUT se despliegan en la pantalla de la siguiente manera:



1.3 Activación de una ventana

La ventana activa es aquella donde está el cursor. En la ventana activa se pueden emitir comandos del sistema; en el PROGRAM EDITOR, además se puede entrar texto y ejecutar programas.

En el ambiente de ventanas, sólo una está activa a la vez. Al entrar al sistema SAS, la ventana activa es la del PROGRAM EDITOR.

Se puede activar una ventana de las siguientes formas:

1) Mover el cursor con las flechas de desplazamiento a la ventana que se desea activar.

2) Entrar el nombre de la ventana que se quiere activar en la línea de comandos de la ventana activa.

3) Usar las teclas de funciones que corresponden a cada ventana. Las teclas son las siguientes:

F3 Activa la ventana **LOG**

F4 Activa la ventana **OUTPUT**

F6 Activa la ventana **PROGRAM EDITOR**

1.4 Movimiento dentro de una ventana

Existe una serie de teclas y comandos que se utilizan para moverse dentro de una ventana. Muchos comandos tienen teclas equivalentes (estas equivalencias aparecen al abrir la ventana KEYS, y se pueden asignar nuevas teclas a comandos que no tengan tecla asignada).

Teclas:

Flechas: Mueve el cursor en cuatro direcciones (arriba, abajo, a la izquierda y a la derecha).

HOME: Regresa el cursor a la línea de comandos de la ventana activa.

PgDn: Avanza una página.

PgUp: Retrocede una página.

Shift F8: Avanza media página hacia la derecha .

Shift F7: Avanza media página hacia la izquierda .

ENTER: Si el cursor está en el área de texto del PROGRAM EDITOR, lo mueve al comienzo de la siguiente línea.

F7: Agranda la ventana activa (utiliza toda la pantalla para la ventana). También, hace regresar al ambiente original (tres ventanas en la pantalla)

F1: Abre la ventana HELP.

F2: Abre la ventana KEYS.

F10 Ejecuta un programa SAS.

F9 Recupera en la ventana PROGRAM EDITOR el último programa que se ejecutó

CTRL X Regresa al ambiente DOS sin salir de SAS

Comandos: Se digitan sólo en la línea de comandos de la ventana activa y requieren oprimir ENTER para que sean ejecutados

RI:	Avanza media página hacia la derecha (como SHIFT F8).
LEFT:	Avanza media página hacia la izquierda (como SHIFT F7).
RIn:	Avanza n columnas hacia la derecha.
LEFTn:	Avanza n columnas hacia la izquierda.
DOWN:	Avanza una página. (Como PgDn)
UP:	Retrocede una página. (Como PgUp)
DOWNn:	Avanza n líneas.
UPn:	Retrocede n líneas.
TOP:	Va al inicio de la ventana. (Como Home)
BOT:	Va al final de la ventana.
ZOOM:	Agranda la ventana. También vuelve al ambiente original de tres ventanas en la pantalla (como F7).
END:	Sale de la ventana actual y regresa a la anterior.
BYE:	Termina la sesión SAS y regresa al Sistema Operativo.
ENDSAS:	Igual a BYE.
CLEAR:	Limpia el contenido de una ventana.
SUBMIT:	Ejecuta un programa SAS (como F10).
RECALL:	Recupera en la ventana PROGRAM EDITOR el último programa que se ejecutó (como F9).
X	Regresa al ambiente DOS sin salir de SAS (como CTRL X).
EXIT	Regresa al ambiente SAS desde el DOS.

El usuario de SAS puede realizar los movimientos dentro de una ventana, utilizando cualquiera de los comandos o teclas anteriores

1.5 Otras ventanas

Además de las tres anteriores, existen las siguientes ventanas:

KEYS :	Para ver, incluir o modificar las teclas de funciones.
HELP:	Para obtener información general sobre cualquier tema del Sistema SAS.

TITLES: Para incluir títulos (hasta diez) para documentar las salidas de los diferentes procedimientos del programa que se ejecuta.

LIBNAME: Muestra las bibliotecas de SAS y su localización.

DIR: Muestra el contenido de una biblioteca.

MENU: Para trabajar con el MENU de SAS.

1.6 Almacenamiento de ventanas

El usuario puede almacenar el contenido de las ventanas primarias (PROGRAM EDITOR, LOG, OUTPUT) como archivos ASCII.(1)

Para almacenar el contenido de cualquiera de estas ventanas se emplea el comando FILE, seguido de la dirección y el nombre del archivo entre comillas. Este comando se da en la línea de comandos de la ventana que se quiere almacenar.

El nombre del archivo debe tener las características de archivos DOS, con un nombre (máximo 8 caracteres) y una extensión (máximo 3 caracteres).

Para una mejor documentación y orden con los archivos creados por las ventanas SAS se recomiendan las siguientes extensiones:

LST: Para almacenar el contenido de la ventana OUTPUT

LOG: Para almacenar el contenido de la ventana LOG.

SAS: Para almacenar el contenido de la ventana PROGRAM EDITOR, si éste es un programa SAS.

DAT: Para almacenar el contenido de la ventana PROGRAM EDITOR, si éste es un conjunto de datos.

Al almacenar los archivos con el comando FILE, éstos se graban en código ASCII.

Ejemplo: Para grabare el contenido de la ventana del editor en el archivo "EJEMPLO.SAS" en un diskette en la unidad A:

Command ===> **FILE 'A:EJEMPLO.SAS'**

1.7 Cargando los archivos en el Editor

Sólo de la ventana PROGRAM EDITOR se puede llamar archivos externos (ASCII). El comando para cargar archivos en esta ventana es INCLUDE o INC seguido de la dirección y el nombre del archivo entre comillas.

(1) Un archivo ASCII (American Standard Code Interchange International), es aquel archivo que está grabado en un código estandard, el cual puede ser interpretado por diferentes sistemas o paquetes.

Ejemplo: Para leer el archivo "EJEMPLO.SAS" de la unidad A: en la ventana PROGRAM EDITOR.

Command ====> **INCLUDE 'A:EJEMPLO.SAS'**

1.8 Impresión de ventanas

Las ventanas cuyo contenido se puede imprimir, son las primarias: PROGRAM EDITOR, LOG y OUTPUT. Desde la línea de comandos de la ventana cuyo contenido se desea imprimir se digita el comando **FILE 'PRN:'**.

1.9 Ejecución de un programa SAS

Para ejecutar un programa SAS a través de Manejador de Pantallas, éste debe estar accesible en la ventana PROGRAM EDITOR. La ejecución se puede iniciar de dos formas:

- Digitando el comando SUBMIT en la línea de comandos de la ventana PROGRAM EDITOR

- Oprimiendo la tecla F10 desde la ventana PROGRAM EDITOR.

Cuando el programa se inicia, desaparece de la ventana y aparece una R (RUN) en la esquina inferior derecha de la ventana PROGRAM EDITOR. El SAS despliega en la ventana LOG todos los mensajes sobre la ejecución del programa. Algunos de estos mensajes son: de error, contenido de los archivos, tiempo de ejecución, uso de memoria, etc.

En la ventana OUTPUT el sistema despliega todas las salidas de los diferentes procedimientos que incluye el programa. Si el programa tiene sólo pasos DATA (sin ninguna instrucción PUT OUTPUT), la ventana OUTPUT queda vacía.

1.10 Lectura de mensajes y corrección de programas

Después que un programa termina de ejecutarse, se debe activar la ventana LOG para ver los mensajes sobre la ejecución de éste. Si hay errores en el programa, se deben seguir los siguientes pasos:

- Tomar nota de los errores en la ventana LOG.

- Limpiar las ventanas LOG y OUTPUT. Para hacerlo digite el comando CLEAR en la línea de comandos de cada ventana.

- Activar la ventana PROGRAM EDITOR y recuperar el programa, digitando el comando RECALL desde la línea de comandos o presionando la tecla F9.

- Hacer las correcciones necesarias en el programa

- Ejecutar el programa corregido.

  Estos pasos deben realizarse las veces que sea necesario hasta que el programa funcione sin errores.

CAPITULO II: El Editor

Como se vió anteriormente, con el Editor se pueden crear y modificar archivos. Las principales teclas y comandos de que dispone el Editor de SAS para ayudar al usuario en este tipo de operaciones son:

2.1 Teclas:

INS	Esta tecla activa o desactiva la modalidad de inserción. En esta modalidad, se insertan caracteres en el texto, moviendo hacia la derecha el resto del texto, sin borrar el texto anterior. Cuando no se está en esta modalidad, se escribe encima del texto. Cuando está en modo INS parece una I en la esquina inferior derecha de la ventana
<-----	La tecla BACKSPACE hace retroceder el cursor borrando el texto a medida que retrocede.
DEL	La tecla DELETE borra el caracter que está en la posición del cursor.
END	La tecla END borra los caracteres de la línea a partir de la posición del cursor.

2.2 Comandos de línea

Los comandos de línea se escriben sobre los números de las líneas que aparecen en la parte izquierda de la ventana del PROGRAM EDITOR. Los principales comandos de línea son:

00001 00002 0I003	Para insertar una línea en blanco, entrar **I** sobre el número de la línea previa. **IB** para insertar una línea antes de la línea donde está el comando. **IA** para
0IB04 0IA05	insertar una línea después de la línea donde está el comando
00001 00002 0I503 0I5B4	Para insertar n líneas en blanco, entrar **In** (n = número de líneas a insertar) sobre el número de la línea previa. **InB** para insertar n líneas antes de la línea donde está el comando. **IA** para insertar una línea después de la línea donde está el comando.
00001 00002 00D53	Para eliminar una línea, entrar **D** sobre los números de la línea que se quiere eliminar. Para eliminar n líneas, entrar **Dn** (donde n= número de líneas que se quiere eliminar).

Se puede insertar o eliminar una línea presionando desde la línea donde está el cursor las teclas **ALT I** (Insertar) o **ALT D** (Eliminar).

00001 00002 00003 00A04	Para copiar una línea, entrar **C** sobre los números de la línea que se quiere copiar y entrar una **A** (después) o una **B** (antes) sobre los números de la línea donde se va a copiar.

00001
0M002
00003
00B04
00005

Para mover una línea, entrar **M** sobre los números de la línea que se quiere mover y entrar una **A** (después) o una **B** (antes) sobre los números de la línea donde se va a mover.

00001
00002
R0003

Para reproducir una línea en la posición siguiente, entrar una **R** sobre los números de la línea que se quiere reproducir.

2.3 Operaciones con bloques de texto

Para eliminar, copiar o mover bloques de texto, los comandos son los siguientes:

0dd01
00002
00003
0dd04

Para eliminar un grupo de líneas, entrar **DD** sobre los números de la primera y última líneas que se quieren eliminar

0CC01
00002
00003
0CC04
000A5

Para copiar un grupo de líneas, entrar **CC** sobre los números de la primera y la última líneas que se quieren copiar y entrar una **A** (después) o una **B** (antes) sobre los números de la línea donde se va a copiar.

0MM01
00002
00003
000MM
00b05

Para mover un grupo de líneas, entrar **MM** sobre los números de la primera y última líneas que se deseen mover y entrar una **A** (después) o una **B** (antes) sobre los números de la línea donde se quieren mover las líneas.

0rr01
00002
00003
00004
0rr05

Para duplicar un grupo de líneas inmediatamente después de la última del grupo, entrar **RR** sobre los números de la primera y la última líneas que se desean reproducir.

Todos los comandos estudiados anteriormente (de línea) se pueden escribir en mayúscula o minúscula y en cualquier posición de los números de las líneas.

2.4 Comandos del Editor

Existen comandos que son válidos sólo para el Editor, tales como:

FIND: Busca una hilera de texto indicado.

Command ====> FIND 'RAZAS'

Localiza la línea que tiene el texto 'RAZAS'

NUMS: Elimina de la pantalla los números de línea y los hace aparecer.

CHANGE: Cambia una o más ocurrencias de una hilera de texto por otra.

Command ====> CHANGE 'RASAS' 'RAZAS'

El comando anterior cambia la primera ocurrencia de "RASAS" por "RAZAS" que encuentra en el texto.

Command ====> CHANGE 'RASAS' 'RAZAS' ALL

El comando anterior utilizando ALL cambia todas las ocurrencias de 'RASAS' por 'RAZAS' que encuentra en el texto.

CAPITULO III: Archivos y estructura de un programa SAS

3.1 ARCHIVOS

Para procesar datos con SAS, es necesario que estos estén organizados y grabados como archivos. Un archivo es un conjunto de registros semejantes, conservado en dispositivos de almacenamiento de computadora.

A través de un programa SAS se puede tener acceso a dos tipos de archivos:

3.1.1 **Archivos externos** (ASCII) generados con otro tipo de software como Lotus, Word, Editores, etc. En archivos ASCII se tienen dos componentes:

-Campo: es un tipo de información que se registra sobre los objetos de un archivo

-Registro: consta de una entrada para algunos o todos los campos que se toman de un objeto.

3.1.2 **Archivos SAS** generados por el paso DATA de SAS. En los archivos SAS se tienen dos componentes:

Variable: conjunto de datos referentes a una característica

Observación: datos asociados a una entidad o individuo.

Como se vió anteriormente, los archivos SAS se componen de variables y observaciones. El nombre de una variable dentro de un programa SAS debe tener como máximo ocho caracteres (pueden ser letras o números y debe empezar siempre con una letra). Las variables pueden ser: numéricas, alfabéticas, de fecha o de tiempo.

Un conjunto de datos almacenado en un archivo SAS, puede ser interpretado y procesado por los diferentes procedimientos que contiene SAS. Es importante señalar que los archivos SAS pueden ser procesados solamente por medio de programas escritos en SAS y ejecutados bajo ambiente SAS.

3.1.2.1 **Archivos SAS Temporales**: Son aquellos creados en un paso DATA con los cuales se pueden utilizar procedimientos SAS para analizar los datos sólo durante el transcurso de la sesión SAS. Al finalizar la sesión, el archivo SAS se borrará. . Si se desea realizar análisis posteriores, será necesario volver a programar el paso DATA. (Ver figura 1)

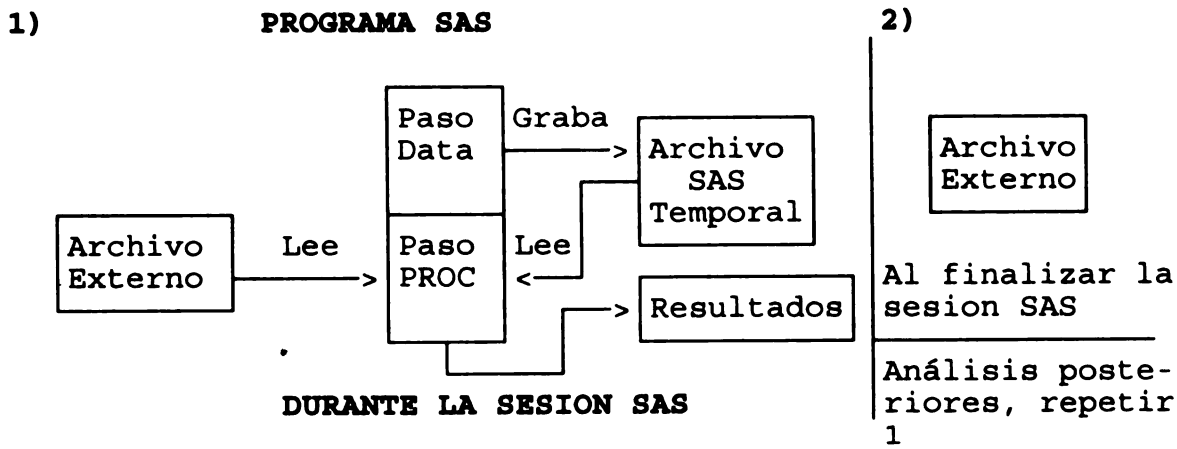


FIGURA 1

3.1.2.2. **Archivos SAS Permanentes**: Son aquellos creados en un paso DATA con los cuales se pueden utilizar procedimientos SAS para analizar los datos durante la sesión SAS. Al finalizar la sesión, el archivo SAS permanecerá grabado y posteriormente puede volver a ser utilizado sin necesidad de ejecutar de nuevo el paso DATA. Un archivo SAS permanente puede ser procesado directamente por los procedimientos SAS. (Ver figura 2)

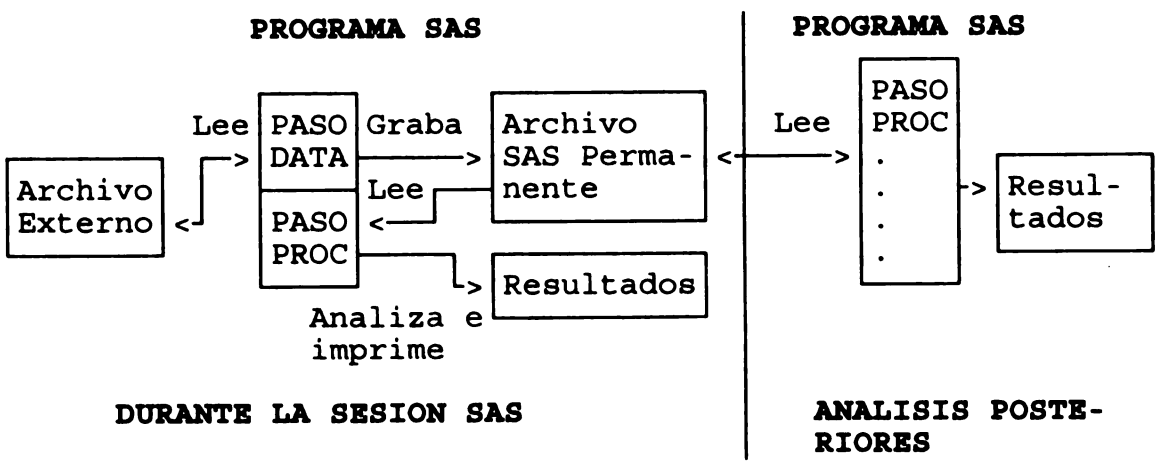


FIGURA 2

Un archivo será temporal o permanente de acuerdo a cómo se especifique su nombre en la instrucción DATA.

Ejemplo de un conjunto de datos SAS:

NOMBRE	SEXO	EDAD	ESTATURA CMS	PESO LBS
JUAN	M	33	173	170
MARIA	F	25	169	125
LUIS	M	19	180	205
CARLOS	M	27	163	140
ANA	F	23	176	136
JOSE	M	31	183	179

En la tabla anterior cada una de las columnas: sexo, edad, estatura y peso forman las variables del archivo y cada fila (una por persona) es una observación del archivo.

3.2 Estructura de un programa SAS

Todos los trabajos SAS son una secuencia de pasos SAS. Existen solamente dos clases de pasos SAS:

- Pasos DATA, los cuales graban, leen, corrigen, manipulan y transforman archivos de datos
- Pasos PROC (Procedimientos) utilizados para analizar, procesar y manipular archivos

Un programa SAS es una secuencia de instrucciones SAS. Las instrucciones SAS tienen formato libre, pueden ser ingresadas en cualquier lugar, y en la cantidad que se desee. Sin embargo, es recomendable que se digite una instrucción por línea para un mejor orden en la programación.

Las instrucciones SAS tienen dos características importantes:

- Comienzan con una palabra clave y
- Terminan con un punto y coma (;)

Ejemplo de una instrucción SAS:

- **INPUT Nombre $ Sexo $ Edad Estatura Peso;**

Un programa SAS puede estar compuesto de:

- Sólo pasos DATA. Por ejemplo, cuando se quiere leer información que se encuentra en uno o más archivos externos (ASCII) para generar uno o más archivos SAS. (Ver Figura 3)

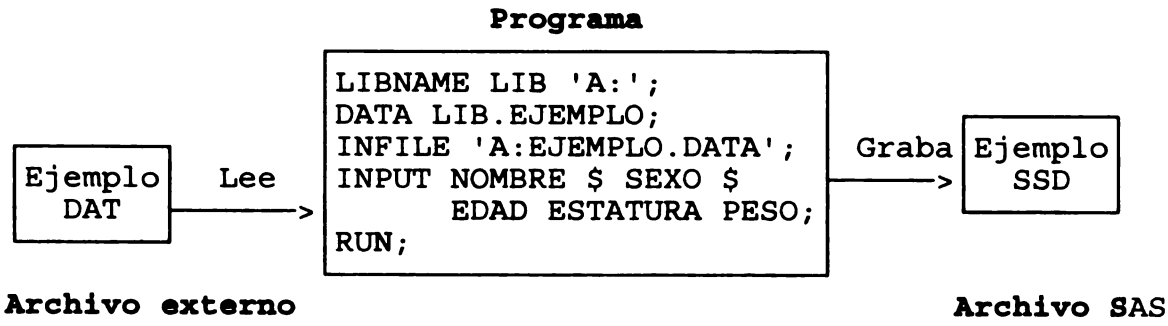


FIGURA 3

- Sólo pasos PROC. Cuando se quieren analizar datos que están grabados en archivos SAS. (Ver figura 4)

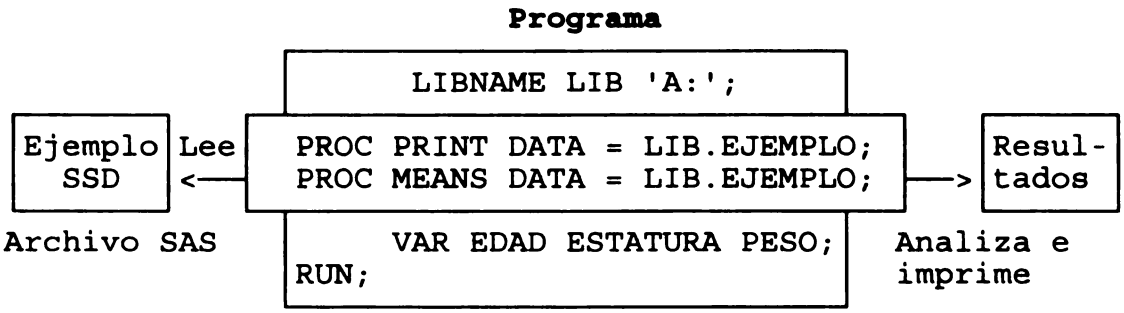


FIGURA 4

- Combinaciones de pasos DATA y PROC. Por ejemplo, cuando se leen archivos externos, se crean archivos SAS y se analizan los datos. (Ver figura 5)

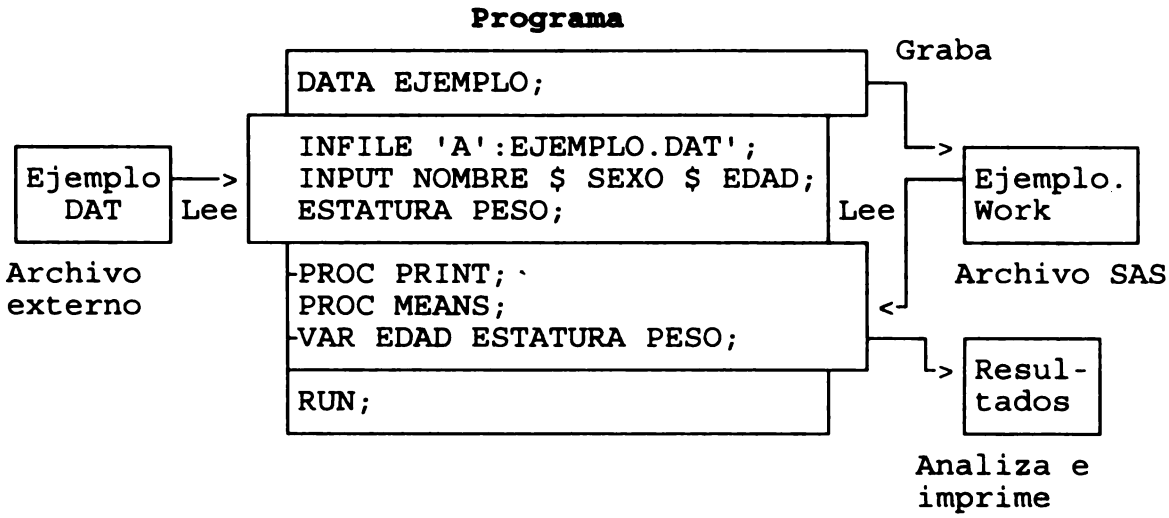

Programa
DATA EJEMPLO;
Graba
INFILE 'A':EJEMPLO.DAT';
INPUT NOMBRE $ SEXO $ EDAD;
ESTATURA PESO;
Ejemplo
DAT
Lee
Lee
Ejemplo.
Work
Archivo
externo
Archivo SAS
PROC PRINT;
PROC MEANS;
VAR EDAD ESTATURA PESO;
RUN;
Resul-
tados
Analiza e
imprime

FIGURA 5

CAPITULO IV: Programación del paso DATA

4. El paso DATA

El paso DATA es una serie ordenada de instrucciones para manejar conjuntos de datos, que comienza con la instrucción DATA. Las principales funciones de este paso son:

- Leer archivos externos y crear archivos SAS
- Actualizar archivos SAS ya existentes
- Manipular archivos SAS: concatenar horizontal o verticalmente archivos SAS
- Seleccionar o eliminar registros
- Transformar variables
- Generar nuevas variables

El conjunto de datos que se procesa en un paso DATA puede corresponder a alguna de las siguientes formas:

1) **Datos grabados en archivos externos** en disco duro o diskette; cuando se presenta esta situación, el paso DATA debe incluir al menos las siguientes instrucciones:

DATA nombre;
INFILE 'A:nombre.dat';
INPUT lista de variables;

2) **Datos incluidos en el programa**. Cuando los datos por procesar están incluidos en el programa, se debe incluir al menos las siguientes instrucciones:

DATA nombre;
INPUT lista de variables;
CARDS;
líneas de datos

3) **Datos existentes en archivos SAS**. Cuando los datos por procesar están grabados en archivos SAS, se debe incluir al menos las siguientes instrucciones:

DATA nombre;
SET-MERGE-UPDATE nombre;

La forma en que procede el paso DATA para generar el archivo SAS se ilustra en la figura 6. Este paso incluye los siguientes elementos:

1. La palabra clave DATA, indica el comienzo del paso DATA y a continuación se nombra el archivo SAS que se creará.

2. La instrucción INPUT especifica qué variables leer y cómo leerlas en cada registro del conjunto de datos que se va a procesar.

3. Con la instrucción INFILE se indica el nombre del archivo ASCII que se va a procesar, o con la instrucción CARDS se indica que seguidamente vienen los datos.

4. A cada una de las observaciones del conjunto de datos que se está procesando, SAS aplicará, en el orden en que aparecen, cada una de las instrucciones del paso DATA.

5. Graba la observación en el archivo SAS.

4.1 La instrucción DATA

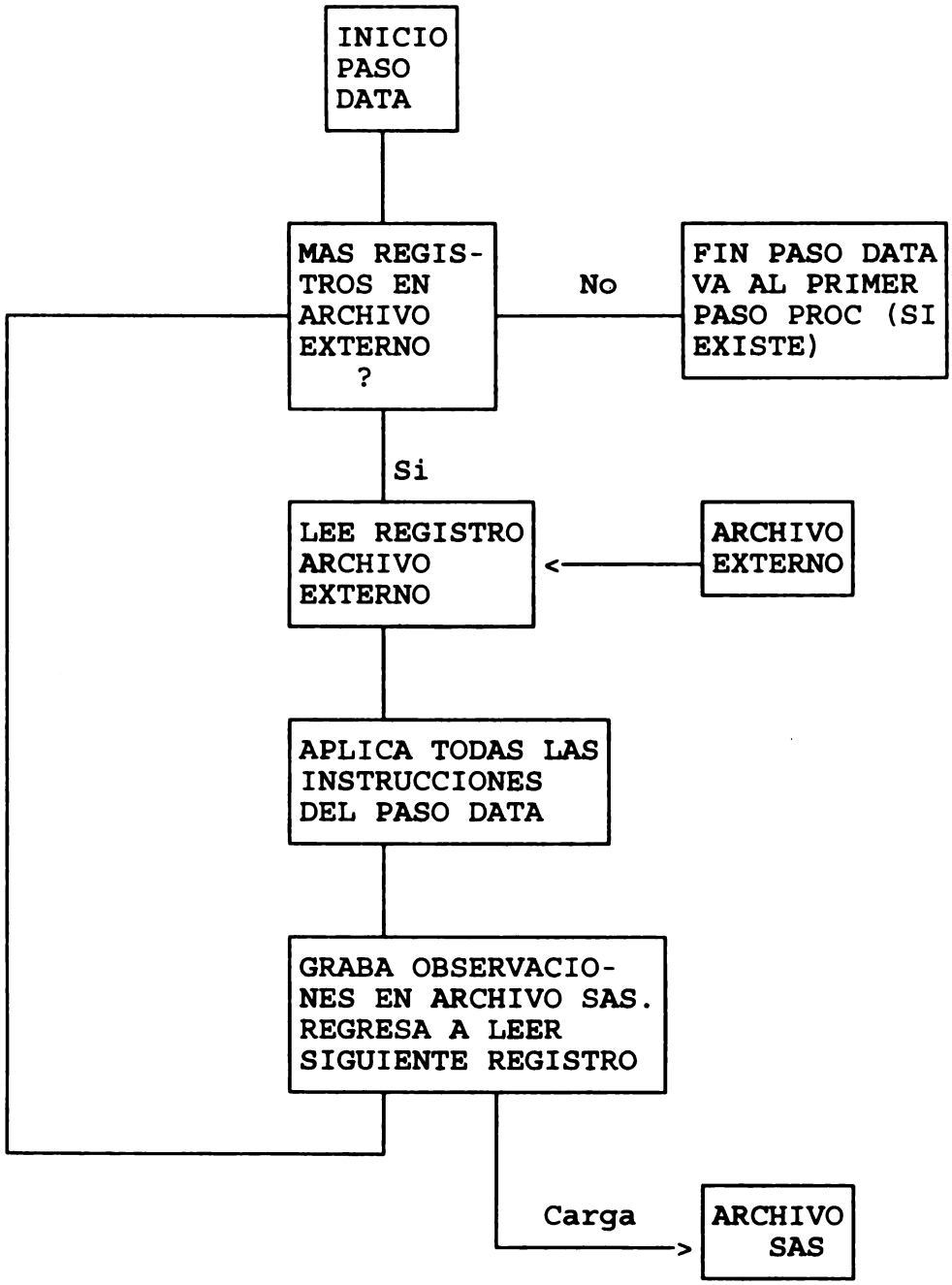
En la instrucción DATA se da el nombre del archivo SAS que se quiere generar. El nombre debe tener como máximo ocho caracteres, que pueden ser letras o números, pero el primero debe ser siempre una letra.

La instrucción DATA le indica a SAS que se inicia un paso DATA y el nombre del arhivo SAS que se generará.

Ejemplos:

DATA CURSO2; genera un archivo temporal
DATA LIB.CURSO2; genera un archivo permanente

En un programa SAS, el nombre de un archivo temporal tiene un solo nivel (una sola palabra). El nombre de un archivo permanente tiene dos niveles (dos palabras separadas por un punto). El nombre del archivo bajo el sistema operativo DOS será el segundo nivel en la instrucción DATA con la extensión SSD. Esta extensión es asignada automáticamente por el SAS para indicar al usuario que éstos son archivos de datos SAS permanentes.



FUNCIONAMIENTO DEL PASO DATA

Ejemplos de nombres de archivos temporales y permanentes:

Instrucción Data	Tipo archivo	Nombre DOS
DATA RAZAS;	Temporal	Razas.SSD
DATA LIB.RAZAS;	Permanente	Razas.SSD

4.2 La instrucción LIBNAME

El sistema SAS utiliza la instrucción LIBNAME para definir el directorio donde van a quedar almacenados los archivos SAS permanentes; es necesario incluir la instrucción LIBNAME junto al paso DATA (antes o después de la instrucción DATA) para crear el archivo permanente. En la instrucción LIBNAME se da el nombre (cualquier nombre, que sirve como sobrenombre (alias) temporal, de ocho caracteres o menos) del sub-directorio, y la dirección real donde se grabará el archivo. Después que termina la sesión SAS, el archivo permanente continúa almacenado en la dirección especificada en el LIBNAME.

Ejemplo:

```
LIBNAME LIB 'C:\DATOS';
DATA LIB.RAZAS;
```

En la instrucción LIBNAME anterior se le indica a SAS que el archivo permanente se grabe en el directorio DATOS del disco duro C:. El primer nombre del archivo en la instrucción DATA debe ser igual al nombre especificado en LIBNAME, que en este caso es "LIB". Cuando se ejecute el programa en la sesión SAS, el nombre del archivo para SAS será LIB.RAZAS. Al finalizar la sesión, el nombre del archivo bajo ambiente DOS se llamará RAZAS.SSD, que corresponde al segundo nombre en la instrucción DATA más la extensión SSD, que asigna el SAS a los archivos permanentes de datos.

Si el archivo se va a utilizar en una sesión posterior, no es necesario emplear el nombre utilizado antes en la instrucción LIBNAME. El nombre del archivo (segundo nivel en el paso DATA) sí debe ser el mismo del proceso anterior.

4.3 La instrucción INFILE

La instrucción INFILE se utiliza cuando el programa va a procesar datos que se encuentran en archivos externos (ASCII). En la instrucción se indica el nombre y la localización del archivo externo que se va a leer.

Ejemplo:

```
INFILE 'C:\CURSO\RAZAS.DAT';
```

En este ejemplo, el programa lee los datos del archivo externo RAZAS.DAT que se encuentra en el directorio **\curso** del disco C.

4.4 La instrucción CARDS

Si el conjunto de datos que se va a procesar se incluye dentro del programa, la instrucción CARDS indica que los datos se incluyen a continuación. Cuando se utiliza la instrucción CARDS, ésta debe ser la última instrucción del paso DATA y debe aparecer inmeditamente antes de lo datos.

Ejemplo:

```
DATA nombre;
INPUT lista de variables;
otras instrucciones del paso DATA
CARDS;
líneas de datos
```

4.5 La instrucción INPUT

La instrucción INPUT se utiliza para:

- Asignar nombre a las variables SAS que corresponden a los datos que se procesarán.
- Indicar el tipo de cada variable. Por ejemplo, si es numérica, de caracteres, de fecha, etc.
- Indicar la forma (formato) en que están almacenados los datos que se procesarán.
- Leer las líneas de datos del archivo externo que se grabarán en el archivo SAS.

La instrucción INPUT tiene tres formas, de acuerdo al modo como están dispuestos los datos:

- columna
- lista
- formato

En una instrucción INPUT se pueden mezclar especificaciones correspondientes a estas formas. Además, para flexibilizar aún más la entrada de datos, se pueden utilizar "controles del apuntador". Un apuntador es un indicador que se posiciona en una columna de un registro o se mueve de un registro a otro.

La instrucción INPUT permite controlar el desplazamiento del apuntador, como se verá más adelante.

4.5.1 Entrada de datos en forma de columna

Se especifica dónde encontrar los valores en el registro de entrada por medio de la posición de la columna.

Ejemplo:

Se tiene el siguiente conjunto de datos en formato de columna:

Nombre									Sexo		Edad		Altura			Peso		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
J	U	A	N		J	O	S	E		M	3	3	1	7	3	1	7	0
A	L	E	J	A	N	D	R	A		F	2	5	1	6	9	1	2	5
F	E	R	N	A	N	D	O			M	1	9	1	8	0	2	0	5
G	U	I	L	L	E	R	M	O		M	2	7	1	6	3	1	4	0
P	A	T	R	I	C	I	A			F	2	3	1	7	6	1	3	6
R	I	G	O	B	E	R	T	O		M	3	1	1	8	3	1	7	9

La tabla anterior contiene las variables: nombre, sexo, edad, altura y peso. Los datos se encuentran en formato de columna; para cada variable se reservan columnas especificadas que se mantienen constantes para todas las observaciones.

La instrucción INPUT para este conjunto de datos es de la siguiente forma:

INPUT Nombre $ 1-9 Sexo $ 11 Edad 12-13
Altura 14-16 Peso 17-19;

Note que en esta instrucción las variables Nombre y Sexo van seguidas del signo ($), para indicar que la variable es alfabética. Las demás variables: Edad, Altura y Peso son numéricas. Como se muestra en la instrucción INPUT en forma de columna, se da el nombre a la variable (máximo 8 caracteres) y se indica el tipo de la variable (si es numérica no se indica nada) además de la columna inicial y final donde se encuentra el dato.

Cuando se utiliza este formato, las posiciones de las variables deben ser fijas; es decir, si en el INPUT se indica que la variable Nombre está en las columnas 1 a 9, los datos para esa variable deben estar siempre entre esas posiciones. Este tipo de formato permite leer todo o parte del registro. En este ejemplo se lee todo el registro. Sin embargo, es posible leer sólo algunas variables; por ejemplo, si quisiéramos leer sólo el nombre y la edad, la instrucción sería:

INPUT NOMBRE $ 1-9 EDAD 12-13;

4.5.2 Entrada de datos en forma de lista

Las variables se listan en el orden en el que aparecen en el registro de entrada. Se deben tener en cuenta los siguientes aspectos en esta forma de entrada de datos.

- El orden en el que aparecen las variables en la instrucción INPUT debe coicidir exactamente con el orden de las variables en el archivo por leer.

- Los valores de las variables deben estar separados al menos por un espacio en blanco.

- En los valores para las variables alfabéticas no se permiten espacios intermedios. La longitud máxima es de ocho dígitos.

- Si se dejan espacios en blanco, cuando falta el valor de una variable en una observación, los nombres de las variables y sus valores se desfazan. Los valores faltantes en este tipo de formato se deben indicar con un punto (.).

  Las posiciones de las variables no tienen que ser fijas

Ejemplo:

Se tiene el siguiente conjunto de datos en formato de Lista:

Nombre									Sexo			Edad			Altura					Peso				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
J	U	A	N	J	O	S	E				M		3	3			1	7	3			1	7	0
A	L	E	J	A	N	D	R	A			F		2	5			1	6	9			1	2	5
F	E	R	N	A	N	D	O				M		1	9			1	8	0			2	0	5
G	U	I	L	L	E	R	M	O			M			.			1	6	3			1	4	0
P	A	T	R	I	C	I	A				F		2	3			1	7	6			1	3	6
R	I	G	O	B	E	R	T	O			M		3	1			1	8	3			1	7	9

La instrucción INPUT para este conjunto de datos es de la siguiente forma:

INPUT Nombre $ Sexo $ Edad Altura Peso;

En el cuadro anterior se observa que:

- Existe al menos un espacio en blanco para separar los valores de las variables.

- Los valores de las variables guardan un orden entre una columna inicial y una columna final. Esto no es necesario; sin embargo, para una revisión posterior de los datos, esta forma puede facilitar la tarea.

- En la línea que corresponde a GUILLERMO, no se tiene el dato para la variable Edad. Se digita un punto (.) para indicar que es un "valor faltante"[2]. Cuando se ejecuta algún procedimiento, este no considera el "valor faltante" al momento de realizar los cálculos.

- La variable nombre, en algunos de sus datos, tiene más de ocho dígitos. En estos casos solamente se registran hasta ocho caracteres, por lo cual algunos nombres quedan truncados (Ver instrucción LENGTH para manejar este problema).

(2) Un "valor faltante" es cuando en algunas de las variables no se tiene el dato para uno o más registros.
"Métodos estadísticos elementales para Técnicos Forestales-F.Freese"

4.5.3 Entrada de datos con formato

Se especifica la longitud de la variable y si tiene decimales o no. Los formatos que se tienen son los siguientes:

w. Longitud (ancho) del campo numérico, sin decimales
w.d Numérico con decimales
$w. Longitud del campo de caracteres

donde w y d simbolizan números:

Ejemplo:

Se tiene el siguiente conjunto de datos en forma de formato:

Nombre									sexo	edad		altura				peso			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
J	U	A	N		J	O	S	E	M	3	3	1	.	7	3		1	7	0
A	L	E	J	A	N	D	R	A	F	2	5	1	.	6	9		1	2	5
F	E	R	N	A	N	D	O		M	1	9	1	.	8	0		2	0	5
G	U	I	L	L	E	R	M	O	M			1	.	6	3		1	4	0
P	A	T	R	I	C	I	A		F	2	3	1	.	7	6		1	3	6
R	I	G	O	B	E	R	T	O	M	3	1	1	.	8	3		1	7	9

La siguiente instrucción INPUT lee este conjunto de datos con formato:

INPUT Nombre $ 9. Sexo $1. Edad 2. Altura 4.2 Peso 4.;

En el cuadro anterior se observa que:

- La variable Nombre es de caracteres y tiene una longitud máxima de 9 dígitos.
- La variable Sexo es de caracteres y tiene una longitud máxima de 1 dígito.
- La variable Edad es numérica y tiene una longitud máxima de 2 dígitos sin decimales.
- La variable Altura es numérica con decimales y tiene una longitud máxima de 4 dígitos. En este caso el punto se considera como un dígito más del campo. (El punto podría no estar en el archivo y el formato sería 3.2)
- La variable Peso es numérica sin decimales y tiene una longitud máxima de 3 dígitos y es numérica. Al existir un espacio a la izquierda de los valores, se dió un formato de cuatro dígitos. Este dígito en blanco no altera el contenido del valor.

4.6 Controles del apuntador

Para todas las formas del INPUT existen los siguientes controles del apuntador:

@n mover el apuntador a la columna n

+n mover el apuntador n posiciones a la derecha

@ Retener el apuntador en la línea que se está leyendo

@@ Seguir leyendo valores en el mismo registro y si se termina, ir a la línea siguiente.

Los primeros dos controles son para desplazamiento dentro del registro; los últimos dos, para el desplazamiento entre registros.

Ejemplo:

Se tiene el siguiente registro:

SITIO													RAT.		REP.							REND									% HUM	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
T	U	R	R	I	A	L	B	A					0	1		1						1	9	0	0	.	8						6	0

La instrucción INPUT se puede escribir de la siguiente manera, utilizando el apuntador de desplazamiento en el registro:

INPUT Sitio $ 1-9 @14 Trat 2. +1 Rep 1. @23 Rend 6.1 +5
Hum 2.;

En la instrucción anterior tiene el efecto siguiente:

- La variable Sitio se lee como alfabética, con un formato columnar (1-9).
- El apuntador **@14**, hace desplazarse al cursor de lectura en el registro hasta la columna 14. De este punto se comienza a leer la variable Trat (con formato 2.) con una longitud de dos dígitos, sin decimales.
- El apuntador **+1** desplaza el cursor de lectura una columna a la derecha. De esa nueva posición se comienza a leer la variable Rep (con formato 1.) con una longitud de un dígito, sin decimales.
- El apuntador **@23** desplaza el cursor de lectura hasta la columna 23. De aquí se lee la variable Rend, con formato, con una longitud de seis dígitos, incluyendo un decimal.

- El apuntador **+5** hace el cursor de lectura se mueva cinco espacios para leer la variable Hum, con formato, con una longitud de dos dígitos, sin decimales.

4.6.1 Uso del apuntador @

Se tiene el siguiente conjunto de datos:

1	2	3	4	5	6	7	8	9	10	11	12	13
I	D	0	0	2			M		0	8		
M	A	0	0	1			H		0	3		0
X	T	1	9	0			H		0	6		
M	A	0	0	2			M		1	5		
I	D	0	0	3			M		1			

```
Variables              Columnas
Ident. animal             1-5
Sexo                      8
Si el sexo= 'M' (macho)
# montas                 10-11
Si el sexo= 'H' (Hembra)
# partos                 10-11
pérdidas en parto         13
```

Este conjunto de datos tiene las siguientes características:

- El conjunto de variables es diferente dependiendo del sexo del animal.
- Las variables # montas (para machos) y # partos (para hembras), comparten las mismas columnas (10 y 11).

La instrucción INPUT para leer este archivo es:

```
INPUT ident $ 1-5 Sexo $ 8 @;
IF sexo = 'M' THEN INPUT montas;
ELSE INPUT partos perdidas;
```

En el primer INPUT se leen, sin condiciones, las variables **ident** y **sexo**, que son aplicables para ambos sexos.

Se utiliza el apuntador @ para retener el cursor de registro y leer las siguientes variables de acuerdo al sexo del animal. Es necesario utilizar este apuntador, ya que cada vez que SAS encuentra un INPUT, desplaza el cursor al siguiente registro. Las variables **montas**, **partos** y **perdidas** están en el mismo registro que **ident** y **sexo**.

Las instrucciones alternativas (IF-THEN y ELSE), que aparecen en el ejemplo se estudiarán posteriormente.

El archivo SAS generado por los INPUT anteriores queda de la siguiente forma:

VARIABLES

IDENT	SEXO	MONTAS	PARTOS	PERDIDAS
ID002	M	08	.	.
MA001	H	.	3	0
XT190	H	.	6	1
MA002	M	15	.	.
ID003	M	1	.	.

4.7 Lectura de varias observaciones por línea de datos

En los siguientes ejemplos vamos a considerar como observación al elemento del archivo SAS que contiene los valores de todas las variables medidas a un individuo. Una línea de datos puede contener los datos de una o varias observaciones del archivo externo (o datos incluídos) que se lee para generar el archivo SAS.

Ejemplo: Se realizó un experimento donde se toman datos de volumen de copa y altura para 13 árboles. Los datos fueron:

	Arbol												
	1	2	3	4	5	6	7	8	9	10	11	12	13
volumen	22	6	93	62	84	14	52	69	99	98	41	85	90
altura	36	9	67	44	72	24	33	61	64	65	47	60	51

El siguiente programa SAS genera un archivo SAS con los datos anteriores. Estos se incluyen dentro del programa y se digitan las observaciones de varios árboles en un mismo registro.

```
DATA EJEMPLO;
INPUT Volumen Altura @@;
CARDS;
22 36 6 9 93 67 62 44 84 72 14 24 52 33
69 61 99 64 98 65 41 47 85 60 90 51
PROC PRINT;
RUN;
```

En este ejemplo, el archivo SAS tendrá 13 observaciones con dos variables. El archivo quedará de la siguiente forma:

Volumen	Altura
22	36
6	9
93	67
62	44
84	72
14	24
52	33
69	61
99	64
98	65
41	47
85	60
90	51

La instrucción INPUT, utilizando el apuntador @@, graba una observación para cada par de datos (volumen, altura) y recorre todo el registro (línea) de entrada localizando los valores. Al llegar al final del registro, pasa al siguiente registro a leer más datos y así sucesivamente hasta llegar al último registro.

4.8 Lectura de una observación en varios registros

En algunos casos es más fácil digitar los datos de cada observación en varios registros del archivo externo. Cuando el número de variables es grande, y los datos no caben en la pantalla, es más cómodo digitar las variables en varias líneas. En este caso se puede utilizar el ancho de la pantalla (80 columnas) como ancho máximo del registro.

Las siguientes son las formas de leer varios registros para una observación SAS:

Cada instrucción **INPUT** avanza hacia el siguiente registro en el archivo externo.

DATA ejemplo;
INPUT Nombre $ 1-8 Sexo $ 11;
INPUT Edad 4-5;

Se utiliza el caracter / para avanzar al próximo registro.

DATA ejemplo;
INPUT Nombre $1-8 Sexo $ 11 / Edad 4-5;

Se utiliza el signo # para avanzar hacia la primera columna del enésimo registro en el archivo externo.

DATA ejemplo;
INPUT #1 Nombre $ 1-8 Sexo $11 #2 Edad 4-5;

Ejemplo: En el siguiente archivo externo, para cada observación, hay dos registros (líneas).

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
J	U	A	N		C	A	R	L	O	S		M	O	R	A	L	E	S		C	O	M	P	U	T	O			
3	5	1	2	8	-	4	1	5		3	5		M			8	5	0	0	0									
A	N	A		M	A	R	I	A		C	O	N	T	R	E	R	A	S		A	D	M	I	N	I	S	T	.	
1	-	4	1	3	-	0	1	0		2	5		F			4	9	5	0	0	0								
A	L	E	X	I	X		B	R	E	N	E	S		T	O	R	R	E	S	A	D	M	I	N	I	S	T	.	
3	-	2	4	0	-	7	2	9		3	1		M		1	1	0	6	7	0									
R	O	C	I	0		F	E	R	N	A	N	D	E	Z						G	A	N	A	D	E	R	I	A	
2	-	3	2	9	-	5	1	2		2	1		F			9	3	8	0	0									
M	I	G	U	E	L		R	O	D	R	I	G	U	E	Z					C	O	M	P	U	T	O			
3	-	6	8	0	-	0	0	6		2	3		M			4	5	6	0	0									

	Variables	Columnas
En el registro 1		
	Nombre	(1-20)
	Departamento	(21-30)
En el registro 2		
	Cédula	(1-9)
	Edad	(11-12)
	Sexo	(14)
	Salario	(16-21)

El siguiente programa SAS lee el conjunto de datos de la tabla anterior y genera un archivo permanente SAS LIB.Ejemplo en la unidad A:

```
Libname Lib 'A:';
Data Lib.ejemplo;
Infile 'A: ejemplo.dat';
Input nombre $ 1-20 depart $ 21-30;
Input cédula $ 1-9 edad 11-12 sexo $ 14 salario 16-21;
PROC print;
RUN;
```

El archivo SAS queda de la siguiente forma:

observación	Nombre	Depart	Cédula	Edad	Sexo	Salario
1	Juan Carlos Morales	Cómputo	3-128-415	35	M	85000
2	Ana María Contreras	Administ.	1-413-010	25	F	495000
3	Alexis Brenes Torres	Administ.	3-240-729	31	M	110670
4	Rocío Fernández	Ganadería	2-329-512	21	F	93800
5	Miguel Rodríguez	Cómputo	3-680-006	23	M	45600

En la instrucción INPUT se debe considerar lo siguiente:

- Cada vez que aparece un INPUT en el programa, SAS lee un nuevo registro del archivo externo, excepto cuando se utiliza el apuntador @.

- La observación en el archivo SAS se graba cuando se terminan de ejecutar todas las instrucciones del paso DATA, excepto si aparece la instrucción OUTPUT. (Esta instrucción se estudiará más adelante).

4.9 Generación o modificación de nuevas variables

Las formas más comunes de agregar, generar o modificar variables al conjunto de valores de datos en un archivo SAS son:

- Por medio de fórmulas aritméticas.

- De acuerdo a una(s) característica(s) de una(s) variable(s) ya existente(s).

- Modificando una variable ya existente.

- Utilizando las diferentes funciones de que dispone SAS (se anexa una tabla al final del documento con las funciones más utilizadas).

Para generar una nueva variable se debe hacer lo siguiente:

- Escojer un nombre para la variable nueva. Este nombre debe ser diferente de los nombres de las variables que ya existen. Si el nombre de la variable es igual al nombre de una variable ya definida, los valores de la última serán sustituídos por

los valores de la expresión que modifican la variable.

- Escojer la fórmula para generar o modificar la variable.

- Codificar la fórmula como una instrucción SAS. El nombre de la nueva variable se coloca a la izquierda del signo "=". Si utiliza un nombre de variable existente, la misma se modificará y la nueva variable reemplazará a la existente.

- Si es una variable que se genera o se modifica de acuerdo a una característica de otra variable, se debe programar la instrucción IF necesaria.

- Si se necesita hacer uso de una función (raíz cuadrada, logaritmo, arcoseno, etc.) se debe escoger la función adecuada.

Ejemplo: Se tiene el siguiente programa SAS:

```
DATA PROMEDIO;
INPUT  Carnet $ Nota1 Nota2 Nota3;
Promedio = (Nota1 + Nota2 + Nota3)/3;
IF Promedio = 70 THEN
        Resultad = 'GANA';
ELSE Resultad = 'PIERDE';
CARDS;
M001 60 50 70
M002 80 70 90
M003 60 90 75
M004 90 95 93
M005 73 45 62
PROC PRINT;
RUN;
```

Al ejecutar el programa, el archivo SAS contiene la siguiente información:

OBSERVACION	CARNET	NOTA1	NOTA2	NOTA3	PROMEDIO	RESULTADO
1	M001	60	50	70	60.00	PIERDE
2	M002	80	70	90	80.00	GANA
3	M003	60	90	75	75.00	GANA
4	M004	90	95	93	92.67	GANA
5	M005	73	45	62	60.00	PIERDE

En el ejemplo anterior vemos que:

- La variable **Promedio** se generó a partir de **Nota1, Nota2, Nota3**. Se aplica la fórmula aritmética para obtener el promedio de las tres variables.

- La variable **Resultad** se generó a partir de la variable **Promedio**. En este caso, esta nueva variable se generó con base en una característica de la variable **Promedio**. Si esta variable es mayor o igual a 70, la nueva variable asumió el valor "GANA". En caso contrario, (ELSE), la variable asumió el valor "PIERDE".

- A cada registro del conjunto de datos se agregan las nuevas variables.

4.10 Símbolos utilizados en expresiones aritméticas

La siguiente tabla contiene los símbolos utilizados en expresiones aritméticas. El orden de estos representa la jerarquía del orden de ejecución de los mismos. Para alterar este orden se deben utilizar los paréntesis.

SIMBOLO	OPERACION	EJEMPLO	INSTRUCCION SAS
**	exponenciación	$Y = X^2$	Y = X**2;
*	multiplicación	A = BxC	A = B*C;
/	división	A = B/C	A = B/C;
+	suma	A = B+C	A = B+C;
-	resta	A = B-C	A = B-C;

4.11 La instrucción LABEL

Al nombrar una variable SAS, se puede utilizar como máximo 8 dígitos. Esto algunas veces no es suficiente para que el nombre de la variable identifique claramente su contenido. La instrucción LABEL se puede utilizar en un paso DATA para dar una mejor identificación de las variables.

La forma de la instrucción LABEL es:

LABEL variable = 'etiqueta';

donde variable = nombre de la variable SAS

etiqueta = descripción del contenido de la variable (máximo 40 caracteres).

Cuando se utilizan procedimientos (Paso PROC), el nombre de la variable en los listados será sustituido por la etiqueta de la instrucción LABEL, (excepto el PRINT) dando una mejor documentación a los resultados.

Ejemplo:

```
LABEL Rep   = 'Repetición'
Trat        = 'Tratamiento'
nplaha      = 'Número de plantas por hectárea'
rha12h      = 'Rendimiento por ha al 12% hum';
```

4.12 La instrucción DELETE

Esta instrucción es utilizada cuando queremos eliminar observaciones del archivo SAS que cumplan cierta condición. La instrucción aparece siempre como una parte de la instrucción IF

Ejemplo:

```
IF nota >= 70 THEN DELETE;
```

En este ejemplo, todos los casos de las observaciones que cumplan la condición de que la nota "es mayor o igual" a 70 serán eliminados del archivo SAS.

4.13 La instrucción OUTPUT

La instrucción OUTPUT se utiliza cuando queremos que alguna observación se grabe en el archivo SAS. La instrucción casi siempre aparece como una parte de la instrucción IF.

Si la instrucción aparece sola, el paso DATA grabará la observación en el archivo SAS inmediatamente después de leer OUTPUT.

Ejemplo:

IF nota < 70;

En este ejemplo, todas los casos de las observaciones que tienen valor "menor que" 70 en la variable nota serán grabados en el archivo SAS.

Si se tiene un IF sin THEN; el sistema lo interpreta como: THEN OUTPUT.

Se tiene el siguiente conjunto de datos

Estudiante														Nota					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
C	A	R	M	E	N		S	A	N	C	H	E	Z					6	8
C	A	R	L	O	S		T	O	R	R	E	S						9	0
J	O	S	E		M	.		B	R	E	N	E	S				1	0	0
M	A	R	I	A		A	S	T	U	A								7	0
F	R	A	N	C	I	S	C	O		L	E	O	N					5	3
S	A	N	D	R	A		C	A	M	P	O	S						8	1
A	L	V	A	L	O	O	F	U	E	N	T	E	S					7	5
R	O	D	O	C	F			P	E	R	E	Z						8	8
E	M	I	L	C	E		S	A	L	A	Z	A	R					5	7

El siguiente programa SAS lee el conjunto de datos anterior y genera un archivo SAS sólo para los registros donde la nota es mayor o igual a 70. Se supone que la tabla anterior está almacenada en un archivo externo con nombre **notas.dat**

```
DATA Notas;
INFILE 'A:Notas·Dat';
INPUT Nombre $ 1-15 Nota 18-20;
IF Nota > 70 THEN OUTPUT;
PROC PRINT;
RUN;
```

El archivo SAS resultante al ejecutar el programa es:

Observación	Estudiante	Nota
1	Carlos Torres	90
2	José M. Brenes	100
3	María Astúa	70
4	Sandra Campos	81
5	Alvaro Fuentes	75
6	Daniel Pérez	88

4.14 La Instrucción IF

En algunas situaciones se requiere que el Sistema SAS realice ciertas acciones para las observaciones en el conjunto de datos cuando se cumplen algunas condiciones.

Esto puede lograrse en SAS mediante el empleo de la instrucción IF en alguna de sus modalidades. El sistema SAS dispone de los siguientes tipos de IF:

a) **IF-THEN**: Sin acción cuando la expresión es falsa. Una sola acción si la expresión es verdadera. En este caso, si la condición se cumple se ejecuta solo una instrucción.

Sintaxis:

IF condición **THEN** instrucción;

donde instrucción = lo que se ejecuta si la condición se cumple.

Ejemplo:

```
IF nota > 70 THEN DELETE;
```

Esta instrucción elimina todas las observaciones del archivo SAS cuando se cumple la condición que el valor que contiene la variable **nota** sea mayor que 70. Se programó una sola acción cuando la condición es verdadera.

a) **IF-THEN**: Con diferentes acciones de acuerdo a la condición.
ELSE Una acción cuando la condición es verdadera y otra acción cuando la condición es falsa.

Sintaxis:

IF condición **THEN** instrucción1;
ELSE
instrucción2;

donde instrucción1 = Lo que se ejecuta si la condición se cumple
instrucción2 = Lo que se ejecuta si la condición no se cumple.

Ejemplo:

```
IF nota > 70 THEN
  resultad = 'GANA';
ELSE
  resultad = 'PIERDE';
```

Con estas instrucciones se está generando una nueva variable de acuerdo al valor de otra. Para todas las observaciones del archivo SAS cuyo valor en la variable nota sea mayor que 70, el valor que asume la variable **resultad** será "GANA". En caso contrario (condición no se cumple) esta variable asumirá el valor de "PIERDE". Se ejecuta una sola acción con cualquiera de los dos posibles resultados de la condición.

c) **IF-THEN DO**: Sin acciones cuando la condición es falsa.

Varias acciones cuando la expresión es verdadera.

Sintaxis:

```
IF condición THEN DO;
     instrucción1;
     .....
     .....
     .....
     instrucciónn;
END; (FIN del IF)
```

Ejemplo:

```
IF Sitio = 'CATIE' THEN DO;
  canton = 'Turrialba';
  provinc = 'Cartago';
END;
```

Con estas instrucciones se están generando dos nuevas variables. Los valores que tendrán éstas son "Turrialba" para la variable **canton** y "Cartago" para la variable **provinc**. Cuando la condición no se cumple, las nuevas variables no tendrán valor. Siempre que se va a realizar más de una acción de acuerdo a una condición, es necesario que la instrucción THEN vaya acompañada de la instrucción DO, para indicar al SAS que se realizará más de una acción después del THEN.

Toda instrucción DO debe ser cerrada con la instrucción END. La instrucción END le indica a SAS hasta dónde llegan las instrucciones que se deben ejecutar si la condición se cumplió.

IF-THEN DO: Con diferentes acciones de acuerdo a la condición.
ELSE DO Varias acciones cuando la condición es verdadera y varias acciones cuando la expresión es falsa

Sintaxis:

```
IF expresión THEN DO;
     instrucción 1.1;
     .....
     .....
     instrucción 1.n;
  END;
  ELSE DO;
     instrucción 2.1;
     .....
     .....
     instrucción 2.n;
  END;
```

Ejemplo:

```
IF sitio = 'Turrialba' THEN DO;
    rendha = (rend * 10000)/12;
    mazorcha = (mazorc * 10000)/12;
  END; (Fin del ciclo cuando la condición del IF se cumple)
  ELSE DO;
    rendha = (rend * 10000)/8;
    mazorcha = (mazorc* 10000)/8;
  END; (Fin del ciclo cuando la condición no se cumple)
```

Con las instrucciones anteriores, el sistema SAS genera dos nuevas variables de acuerdo al valor de la variable **sitio**. Note que la fórmula aritmética es la misma. Sin embargo, la división por la constante es diferente de acuerdo al valor de la variable sitio.

4.15 Operadores lógicos y de comparación en la instrucción IF

La condición de la alternativa en la instrucción IF casi siempre contiene operadores lógicos o de comparación. Estos operadores que utiliza el SAS se presentan a continuacion.

Símbolo	Abreviatura	Comparación
<	LT	Menor que
< =	LE	Menor o igual que
>	GT	Mayor que
> =	GE	Mayor o igual que
=	EQ	Igual que
^ =	NE	No igual a

Nota: En la instrucción IF se puede usar el símbolo o la abreviatura de los operadores.

Operadores lógicos en la instrucción IF

Algunas veces, para que determinada acción se ejecute, se tiene que cumplir más de una condición.

Ejemplos:

```
IF lugar = 'CATIE' AND variedad = 'CATURRA'
  THEN factor = 1.078;
```

En la anterior instrucción SAS, la variable **factor** asumirá un valor de 1.078 para todas las observaciones del archivo SAS que cumplan las dos condiciones conectadas por el operador lógico AND. Si una de ellas no se cumple, la condición es falsa.

IF (lugar = 'GUAYABO' OR lugar = 'GUAPILES') AND variedad = 'ARABIGO' THEN factor = 0.887;

En la anterior instrucción SAS, la variable **factor** asumirá un valor de 0.887 para todas las observaciones del archivo SAS que cumpla la condición de que en su variable **lugar**, los valores sean 'Guayabo' o 'GUAPILES', y además el valor de la variable **variedad** sea 'ARABIGO'.

OPERADORES LOGICOS EN LA INSTRUCCION IF

OPERADOR	COMPARACION
AND	Y
OR	O
NOT	NO

4.16 Los ciclos DO

Cuando se requiere ejecutar una o algunas instrucciones repetitivamente (ciclos), SAS dispone de la instrucción DO, que puede adoptar las siguientes formas.

DO
DO OVER;
DO WHILE;
DO UNTIL;

El número de veces que se ejecuta el ciclo DO depende de la forma de utilizar la instrucción. Todo ciclo DO termina con la instrucción END.

4.16.1 Ciclos controlados (Do iterativo)

Se puede ejecutar un ciclo controlado, es decir, se indica el inicio y final del ciclo. Para este tipo de ciclo se requiere de una variable de control. La forma de la instrucción es:

DO I = inicio **TO** Final **BY** increm;

donde:

I = Variable de control

Inicio = Valor inicial de la variable de control

Final = Valor final de la variable de control

Increm = Magnitud en que se incrementa el valor de la variable de control (opcional)

Ejemplo:

Se tiene el siguiente conjunto de datos:

TEMPERATURAS						
Número Planta	Alta			Baja		
	Variedad 1	2	3	Variedad 1	2	3
1	3.5	2.5	3.0	8.5	6.5	7.0
2	4.0	4.5	3.0	6.0	7.0	7.5
3	3.0	5.5	2.5	9.0	8.0	7.0

El cuadro anterior contiene datos de altura de plantas de menta. Se tienen dos temperaturas, tres variedades por temperatura y tres plantas por variedad.

Note que este conjunto de datos contiene tres niveles jerárquicos bien definidos. El primer nivel corresponde a las temperaturas, el segundo nivel es dado por las variedades y el tercero corresponde a las plantas.

Utilizando la instrucción DO se puede generar un archivo SAS de estos datos, leyendo únicamente los valores para la variable altura. Las otras tres variables (temperatura, variedad y planta) se pueden generar automáticamente con instrucciones DO.

El siguiente programa crea el archivo SAS para la tabla anterior.

```
DATA CICLOS;
    DO Temp = `alta`, `baja`;
      DO Varied = 1 to 3;
           DO Planta = 1 to 3;
                INPUT Altura @@;
                OUTPUT;
           END;
      END;
    END;
CARDS;
3.5  4.0  3.0  2.5  4.5  5.5  3.0  3.0  2.5
8.5  6.0  9.0  6.5  7.0  8.0  7.0  7.5  7.0
RUN;
```

El archivo SAS que se crea tiene la siguiente forma:

TEMP	VARIED	PLANTA	ALTURA
Alta	1	1	3.5
Alta	1	2	4.0
Alta	1	3	3.0
Alta	2	1	2.5
Alta	2	2	4.5
Alta	2	3	5.5
Alta	3	1	3.0
Alta	3	2	3.0
Alta	3	3	1.5
Baja	1	1	8.5
Baja	1	2	6.0
Baja	1	3	9.0
Baja	2	1	6.5
Baja	2	2	7.0
Baja	2	3	8.0
Baja	3	1	7.0
Baja	3	2	7.5
Baja	3	3	7.0

En el ejemplo anterior se tienen ciclos anidados. En estos casos el ciclo más interno es el que cambia más rápido. Observando el resultado del archivo generado, vemos cómo el número de planta es el que varía más rápido (columna 3); el número de variedad (columna 2) cambia después que se ha completado un ciclo para planta y el nivel de temperatura (columna 1) cambia después que el ciclo de variedad ha terminado.

4.16.2 Ciclos para recorrer arreglos (DO OVER)

Cuando a un conjunto de variables se les va a aplicar una misma acción, se puede definir un arreglo (matriz unidimensional) y recorrer éste con la instrucción DO OVER para realizar la acción a cada una de las variables.

La forma de la instrucción DO OVER es la siguiente:

DO OVER Nombre;
  instrucción1;
  .....
  instrucciónn;
END;

donde:
Nombre = arreglo que se va a recorrer

Note que el **DO OVER** termina con la instrucción **END**

Ejemplo: Se quiere calcular la raíz cuadrada de los valores de 50 variables, de un archivo ASCII. Esto se puede lograr con el siguiente programa

```
DATA CICLO;
INFILE 'A:ejemplo.dat';
INPUT X1-X50;
ARRAY raizcuad X1-X50;
DO OVER raizcuad;
   raizcuad = SQRT(raizcuad);
END;
```

El programa lee un archivo externo que contiene las 50 variables (denominadas X1, X2,...X50). En la instrucción ARRAY se define un arreglo de nombre **raizcuad** cuyos elementos son las cincuenta variables. Con la instrucción DO OVER se recorre uno a uno los elementos del arreglo y se calcula la raíz cuadrada para cada uno de ellos. Si no se utilizara un ciclo DO OVER, sería necesario escribir 50 instrucciones (una para cada variable) para transformar los datos.

La forma de la instrucción ARRAY es la siguiente:

ARRAY nombre lista;

donde:

Nombre = un nombre para el arreglo, no más de ocho caracteres

lista = cada uno de los elementos (variables) que tiene el arreglo

4.16.3 Ciclos condicionados (DO WHILE y DO UNTIL)

En algunas oportunidades se requiere realizar ciclos cuyo final está condicionado al cumplimiento de alguna condición.

Los siguientes son los ciclos DO condicionados del SAS.

DO WHILE (expresión);

donde:

expresión = cualquier expresión. Esta es evaluada cada vez que se ejecuta el ciclo. Mientras la expresión sea verdadera, el ciclo se ejecutará. Cuando la expresión sea falsa, el ciclo terminará.

Ejemplo:

```
N = 0;
DO WHILE (N LE 5);
      N = N + 1;
(otras instrucciones del ciclo)
END;
```

En las instrucciones SAS anteriores se tiene un ciclo condicionado. Se inicializa la variable N con valor cero. La primera vez que entra al DO WHILE, la expresión se cumple (note que la expresión se cumple mientras el valor de N sea menor o igual que 5). Se entra a ejecutar las instrucciones del ciclo. En este caso, la única instrucción es incrementar el valor de N en una unidad. La expresión **N = N + 1** hace que el valor de N se incremente una unidad cada vez que se ejecuta el conjunto de instrucciones correspondiente a un ciclo.

DO UNTIL (expresión)

donde:

expresión= cualquier expresión. Esta es evaluada cada vez que se ejecuta el ciclo. Hasta que la expresión sea FALSA la instrucción se ejecutará, y la primera vez que la expresión sea verdadera, la instrucción dejará de ejecutarse.

Ejemplo:

```
N = 0;
DO UNTIL (N GT 5);
    N = N+1;
(otras instrucciones del ciclo)
END;
```

En las instrucciones SAS anteriores se tiene un ciclo condicionado. Se inicializa una variable con valor cero. La primera vez que entra el DO UNTIL la expresión no se cumple (note que la expresión se cumple cuando la variable N sea mayor que 5). Se entra a ejecutar las instrucciones del ciclo.

Con los ciclos condicionados se debe tener el cuidado de que la expresión condicionante se cumpla en determinado momento. De lo contrario se entraría en un ciclo infinito.

4.17 Las instrucciones KEEP y DROP

Estas instrucciones se utilizan cuando se va a trabajar sólo con una parte de variables de un archivo y son útiles para economizar recursos del sistema de cómputo (disco, memoria).

4.17.1 La instrucción KEEP

Esta instrucción mantiene en el archivo SAS, sólo las variables que son listadas en la instrucción.

La forma de la instrucción es la siguiente:

KEEP lista de variables;

4.17.2 La instrucción DROP

Esta instrucción no incluirá en el archivo SAS aquellas variables que sean listadas en la instrucción.

La forma de la instrucción es la siguiente:

DROP lista de variables;

Ejemplo: Se tiene un archivo externo que contiene 50 variables (X1,...,X50). Se van a calcular las estadísticas descriptivas sólo de algunas variables (X10...X20). En este caso no es necesario grabar en el archivo SAS todas las variables. Se grabará solo aquéllas que van a ser procesadas.

Para este caso se puede utilizar el siguiente programa:

```
DATA EJEMPLO; KEEP X10-X20;
INFILE 'a:ejemplo.dat';
INPUT X1-X50;
PROC MEANS;
RUN;
```

4.18 La instrucción RENAME

Se utiliza para cambiar nombres a variables de archivos SAS.
La forma de la instrucción es la siguiente:

RENAME nombre actual = nombre nuevo;

donde:

nombre actual = nombre actual de la variable

nombre nuevo = nombre que sustituye al nombre actual

4.19 La instrucción TITLE

Se utiliza para añadir líneas de título a las salidas de los procedimientos. Se pueden colocar hasta diez títulos.

La forma de la instrucción es la siguiente:

TITLEn 'título';

donde:

TITLEn = el título enésimo; n es el número de la . línea de título

'título' = el contenido del título enésimo

Ejemplo:

TITULO1 'Datos agronómicos de Turrialba';

TITULO2 'Estadísticas descriptivas';

Estos títulos aparecerán como encabezamiento de todas las páginas que se produzcan hasta el fin de la sesión, a menos que se reemplacen por otros. Si se quiere suprimir una línea de título que se ha estado imprimiendo en una sesión se emplea la instrucción TITLE sin contenido (TITLEn;).

4.20 Concatenación de archivos SAS

El sistema SAS permite generar un archivo resultante de concatenar dos o más archivos SAS.

Los archivos se pueden concatenar agregando variables (horizontalmente) mediante la instrucción MERGE o agregando registros (verticalmente) mediante la instrucciones SET o APPEND.

4.20.1 La instrucción MERGE

Esta instrucción se utiliza para concatenar dos o más archivos SAS horizontalmente, es decir, para juntar valores de diferentes variables correspondientes a los mismos individuos.

Características del MERGE

- Con una instrucción MERGE se pueden concatenar hasta 50 archivos.
- Se pueden concatenar archivos observación por observación. La primera observación de un archivo con la primera del otro, la segunda de uno con la segunda del otro, etc.
- Se pueden concatenar archivos por una o más variables de clasificación, si están ordenados por esas variables, de acuerdo a los valores de esas variables.
- El archivo resultante contendrá las variables de los diferentes archivos concatenados.
- Los archivos que se concatenan tienen que estar ordenados ya sea por posición o por las variables de clasificación.

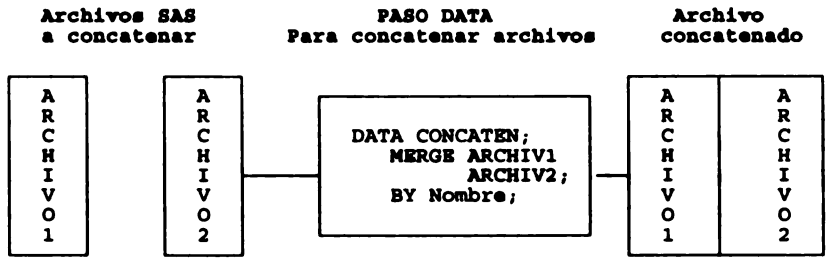
La forma de la instrucción es la siguiente:

```
DATA nuevo;
  MERGE arch1...archn;
    BY var1...varn;
```

donde:

nuevo = el archivo que contiene la concatenación
Arch1...archn = los archivos que se concatenen
Var1...varn = las variables de clasificación. En este caso, todos los archivos que se concatenan deben tener estas variables.

Ilustración gráfica del MERGE



Ejemplo:

Se tienen los siguientes archivos:

Archivo 1

Nombre								Identificación					Fecha Ingreso						Edad			Sexo		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
C	A	R	L	O	S			I	D	0	0	1	0	1	0	1	7	9		4	5			M
A	N	A						I	D	0	0	8	1	6	0	5	8	3		2	8			F
M	A	R	I	O				I	D	0	0	3	2	5	0	9	8	2		3	3			M
M	A	R	I	C	E	L		I	D	1	0	1	0	1	1	1	8	5		2	6			F
B	E	R	T	H	A			I	D	0	9	8	3	1	0	8	8	0		5	2			F
L	U	1	S					I	D	0	7	3	1	5	1	2	9	0		2	1			M
A	N	T	O	N	I	O		I	D	1	2	5	0	1	0	1	8	5		3	7			M
J	O	S	E					I	D	1	4	0	1	5	1	0	9	0		4	0			M
M	A	N	U	E	L			I	D	0	0	4	0	1	0	1	8	5		2	9			M
C	A	R	M	E	N			I	D	1	3	3	0	1	0	1	9	0		1	9			F

Archivo 2

Identificación					Departamento					Salario					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I	D	I	3	3		D	0	0	1		4	5	0	0	0
I	D	0	0	8		D	0	0	3	1	1	5	0	0	0
I	D	0	0	1		D	0	0	2		9	7	6	0	0
I	D	0	7	3		D	0	0	1		3	9	7	5	5
I	D	0	9	8		D	0	0	1	1	4	8	7	4	0
I	D	0	0	3		D	0	0	2		8	3	0	0	0
I	D	0	0	4		D	0	0	3	1	2	3	5	5	0
I	D	I	0	1		D	0	0	1		8	6	6	7	5
I	D	I	2	5		D	0	0	2	1	6	8	4	0	0

El siguiente programa concatena horizontalmente los dos archivos anteriores utilizando la identificación como variable de clasificación.

```
DATA ARCHIVO1;
INFILE 'A:archivo1.dat';
INPUT Nombre $ 1-7 Ident $ 9-13 Fecing $ 14-19
    Edad 21-22 Sexo $ 25;
PROC SORT; BY Ident;

DATA ARCHIVO2;
INFILE 'A:archivo2.dat';
INPUT Ident $ 1-5 Depart $ 7-10 Salario 11-16;
PROC SORT; BY Ident;
DATA CONCATEN;
      MERGE ARCHIVO1 ARCHIVO2;
      BY Ident;
PROC PRINT;
RUN;
```

El archivo SAS concatenado queda de la siguiente forma:

OBSERV.	NOMBRE	IDENT	FECING	EDAD	SEXO	DEPART	SALARIO
1	Carlos	ID001	010179	45	M	D002	97600
2	Mario	ID003	250982	33	M	D002	83000
3	Manuel	ID004	010185	29	M	D003	123550
4	Ana	ID008	160583	28	F	D003	115000
5	Luis	ID073	151290	21	M	D001	39755
6	Bertha	ID098	310880	52	F	D001	148740
7	Maricel	ID101	011185	26	F	D001	86675
8	Antonio	ID125	010185	37	M	D002	168400
9	Carmen	ID133	010190	19	F	D001	45000
10	José	ID140	151090	40	M		.

En el ejemplo anterior note lo siguiente:

- Se utiliza la variable Ident para concatenar los archivos. Esta variable está presente en ambos archivos.
- Los archivos no están ordenados por la variable Ident.
- Se clasifican (ordenan) en cada paso DATA los archivos por la variable Ident. El procedimiento SAS para ordenar archivos es el SORT. (ver sección 5.2.1)
- Los archivos tienen diferente número de observaciones. En el archivo2 no existe la información para Ident = ID140.
- La observación en archivo CONCATEN en Ident = ID140 no contiene los datos para las variables Depart y Salario (SAS las graba como valores faltantes).

4.20.2 La instrucción SET

Esta instrucción lee las observaciones de uno o más archivos SAS. Esta instrucción se utiliza cuando se quiere leer, hacer subconjuntos o concatenar verticalmente archivos SAS existentes para generar un nuevo archivo SAS. Concatenar verticalmente significa, juntar valores de diferentes individuos sobre algunas variables en común.

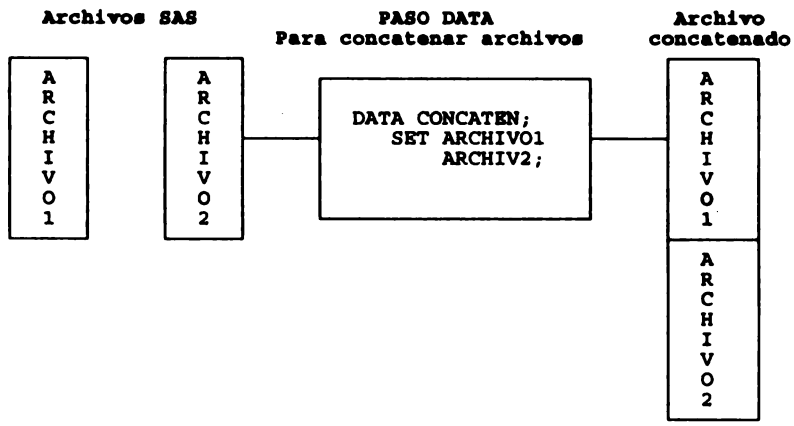
La forma de la instrucción es la siguiente.

DATA nuevo; **SET** arch1...archn;

donde nuevo = El nuevo archivo que se genera

arch1...arch2 = los archivos que contendrá el nuevo archivo

Forma gráfica de concatenación vertical.



Creación de un subconjunto de datos con la instrucción SET

Si se tiene un archivo SAS y no se requiere procesar todo el conjunto de datos, la instrucción SET permite crear un subconjunto de datos de un archivo SAS existente.

Ejemplo:

Se tiene el siguiente conjunto de datos almacenados en un archivo SAS (nombre archivo = ejemplo)

SITIO .	REPETIC	TRAT	PLANTAS	MAZORCAS	RENDIM
1	1	1	128	140	1870.5
1	1	2	110	105	1420.0
1	2	1	153	168	2654.0
1	2	2	148	147	1970.0
2	1	1	105	110	1240.0
2	1	2	97	90	970.5
2	2	1	84	75	1100.0
2	2	2	79	82	843.5
3	1	1	220	233	3205.0
3	1	2	210	206	2870.0
3	2	1	248	267	3723.0
3	2	2	235	232	3028.0

Del archivo anterior se va a generar un nuevo archivo que contenga sólo las observaciones donde el valor de sitio = 3.

El siguiente programa crea el nuevo archivo SAS:

```
DATA NUEVO;
     SET EJEMPLO;
IF Sitio = 3 THEN OUTPUT;
```

La instrucción IF en el programa anterior hace que se graben en el archivo NUEVO sólo aquellas observaciones del archivo EJEMPLO que cumpla con la condición de que el sitio = 3.

El archivo NUEVO resultante es:

SITIO .	REPETIC	TRAT	PLANTAS	MAZORCAS	RENDIM
3	1	1	220	233	3205.0
3	1	2	210	206	2870.0
3	2	1	248	267	3723.0
3	2	2	235	232	3028.0

4.21 Las variables FIRST y LAST

Se puede procesar un conjunto de datos por grupos si se incluye la instrucción BY después de la instrucción SET o MERGE en el paso DATA. Para esto, los datos deben estar ordenados por la(s) variable(s) que identifica(n) al grupo.

Cuando se procesan archivos por grupos, se puede identificar cual es la primera y la última observación de cada grupo existente en el archivo, haciendo uso de ciertas variables automáticamente creadas por SAS, que se llaman FIRST.variable y LAST.variable

4.21.1 La variable FIRST

Esta variable identifica cuál es la primera observación de cada grupo existente en un archivo SAS, tomando el valor 0 si no es la primera observación y 1 si es la primera observación de un determinado grupo.

La forma de aplicar una instrucción a la primera observación de cada grupo es:

IF FIRST.variable **THEN** instrucción;

donde:

variable	=	variable que identifica el grupo
instrucción	=	lo que se ejecuta si el FIRST vale 1

4.21.2 La variable LAST

Esta variable identifica cuál es la última observación de cada grupo existente en un archivo SAS. Toma valor 0 si no es la última observación y 1 si es la última observación de un determinado grupo.

La forma de aplicar una instrucción a la última observación de cada grupo es:

IF LAST.variable **THEN** instrucción;

donde:

variable	=	variable que identifica el grupo
instrucción	=	lo que se ejecuta si el LAST vale 1 (última observación del grupo).

Ejemplo:

El siguiente programa lee el archivo externo, lo ordena por las variables identificación y número de parto y crea un archivo SAS que contiene todos los registros que cumplan con la condición que sea la primera observación (FIRST=1) de cada grupo contenido en la variable identificación de la vaca.

Identif. de la vaca					No. de parto	Fecha de Parto								sexocr	Peso de la cría			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	M	0	0	1		1	0	8	/	0	3	/	7	9	H	2	8	.	5
A	M	0	0	1		2	1	7	/	0	5	/	8	0	M	2	9	.	7
A	M	0	0	1		3	2	5	/	0	2	/	8	2	M	3	2	.	0
A	M	0	0	1		4	1	3	/	0	6	/	8	3	H	3	1	.	5
A	M	0	0	1		5	0	5	/	1	1	/	8	4	H	2	7	.	0
D	R	0	0	2		1	1	9	/	1	2	/	8	7	M	3	3	.	5
D	R	0	0	2		2	3	0	/	0	3	/	8	8	M	2	9	.	6
D	R	0	0	2		3	2	7	/	0	6	/	8	9	H	3	2	.	5
C	L	1	4	0		1	1	5	/	1	0	/	8	9	H	2	7	.	8
C	L	1	4	0		2	2	1	/	1	2	/	9	0	M	3	3	.	0
M	R	1	3	3		1	1	8	/	0	8	/	9	1	H	3	1	.	5

```
DATA VACAS;
INFILE 'A:\VACAS.DAT';
INPUT IDENT $ 1-5 PARTO 7 FECHAPAR $ 8-15 SEXOCRIA $ 16
      PESOCRIA 17-20;

PROC SORT;
      BY IDENT PARTO;
DATA PRIMERO; SET VACAS;
      BY IDENT;
IF FIRST.IDENT THEN OUTPUT;
TITLE 'archivo con primer parto';
PROC PRINT;
RUN;
```

Nota: Cuando se utiliza el BY con las instrucciones FIRST o LAST es necesario ordenar el archivo de acuerdo a las variables que identifican los diferentes grupos.

El archivo SAS resultante al ejecutar el programa queda de la siguiente forma:

IDENT	PARTO	FECHAPAR	SEXO CRIA	PESOCRIA
AM001	1	08/03/79	H	28.5
CL140	1	15/10/89	H	27.8
DR002	1	19/12/87	M	33.5
MR133	1	18/08/91	H	31.5

CAPITULO V. Procedimientos para salidas y estadísticas descriptivas

El paso DATA tiene que ver con el manejo de archivos externos y su incorporación como archivos SAS; además, trabaja con conjuntos de datos y archivos SAS en lo relacionado a su edición, transformación y actualización. Todo esto casi siempre es una preparación de los datos para su manejo posterior por medio del paso PROC.

En este capítulo se explicará sobre las opciones y procedimientos para la elaboración y presentación de informes, en pantalla o impresos, resultantes del manejo y análisis de los datos. Se explica también sobre procedimientos para obtener estadísticas descriptivas.

5.1. Opciones y procedimientos para el control de salida

La mayoría de los procedimientos SAS producen "**salidas**" (resultados o informes que se despliegan en pantalla y que pueden imprimirse o grabarse). La forma de estas salidas pueden ser controladas por el usuario de varias maneras. Algunas de las más utilizadas se describen a continuación.

5.1.1. Cambiando las opciones del sistema.

Para cambiar las opciones de las salidas de los procedimientos existen dos formas:

- Se activa la ventana OPTIONS y se modifican las opciones deseadas.
- Utilizando la instrucción OPTIONS en el programa. Esta instrucción pude ir en cualquier parte del programa, pero es muy común por razones prácticas que aparezca como la primera instrucción.

Las opciones más importantes a considerar en el control de salidas son:

- **LINESIZE =n o LS =n** Determina el número de columnas por página.
- **PAGESIZE=n o PS =n** Determina el número de líneas que se imprimen por página. Para aprovechar al máximo las líneas por página se puede emplear un tamaño de página = 56.
- **DATE** Hace que la fecha se imprima en las páginas.
- **NODATE** Hace que la fecha no se imprima en las páginas.

Ejemplo:

```
OPTIONS LS = 80   PS = 56   DATE;
```

La longitud de la línea generalmente se hace igual o menor que el número de columnas que muestra una pantalla (p. ej. 80 columnas); esto coincide con el número de columnas que caben en una hoja de papel de computadora cuando se emplea el tamaño de letra corriente (10 ó 12 puntos). Sin embargo, si se emplea, por ejemplo un tamaño 8 (condensado) caben hasta 132 caracteres por línea.

En forma correspondiente, se puede indicar al SAS una longitud de línea superior a 80, en cuyo caso, las salidas en la pantalla aparecen truncadas hacia la derecha. Para ver la parte oculta se requiere dar el comando RI (Shift+F8). Sin embargo estas salidas se pueden imprimir en papel empleando la modalidad "condensado" (p. ej. tamaño 8, o 15 caracteres por pulgada).

5.1.2. Definiendo formatos para los valores de variables.

Con el PROC FORMAT se pueden definir formatos para variables numéricas o alfanuméricas. Estos formatos se pueden asociar después dentro de otro procedimiento para cambiar los valores de las variables.

La forma del PROC FORMAT es la siguiente:

PROC FORMAT;

VALUE nombre valoract = 'valornue';

donde	nombre	=	nombre del formato
	valoract	=	valor actual que contiene la variable
	valornue	=	valor que sustituye al actual cuando se imprima

En los procedimientos que se utilicen posteriormente, se asocia(n) la(s) variable(s) con el formato deseado.

Ejemplo: Se tiene el siguiente programa:

```
Data Ejemplo;

INPUT nombre $ sexo estcivil comunid;
CARDS;
María      1    1    1
Carlos     2    2    2
José       2    3    3
Ana        1    1    2
Sonia      1    2    4

Juan       2    1    3
Mario      2    2    1
Miguel     2    3    2
```

```
PROC FORMAT;
      VALUE      genero      1      = 'Femenino'
                             2      = 'Masculino';
      VALUE      civil       1      = 'Soltero'
                             2      = 'Casado'
                             3      = 'Otros';
      VALUE      comunid     1-2    = 'Rural'
                             3-4    = 'Urbana';

PROC PRINT;
Format sexo genero. estcivil civil. comunid comunid.;
RUN;
```

Al ejecutar el programa anterior se obtiene el siguiente resultado, donde los códigos han sido reemplazados, para su presentación, por los nombres más descriptivos de los valores de las variables.

Nombre	Sexo	Estcivil	comunidad
María	Femenino	Soltero	Rural
Carlos	Masculino	Casado	Rural
José	Masculino	Otros	Urbana
Ana	Femenino	Soltero	Rural
Sonia	Femenino	Casado	Urbana
Juan	Masculino	Soltero	Urbana
Mario	Masculino	Casado	Rural
Miguel	Masculino	Otros	Rural

En el programa anterior note que:

- Se definió un formato para cada variable codificada (sexo, estcivil, comunidad). El nombre del formato debe tener como máximo 8 caracteres y deben ser sólo letras.

- Al PROC PRINT se le agrega la instrucción FORMAT donde asociamos los formatos para cada variable a la cual se le definió un formato. El nombre del formato termina con un punto (.), de modo que si el nombre del formato es igual al nombre de la variable no habrá confusión.

5.2. Procedimientos para reorganizar archivos SAS

Existen procedimientos para reorganizar archivos SAS. Los más importantes son los siguientes:

5.2.1. PROC SORT

Este procedimiento se utiliza para ordenar las observaciones de archivos SAS de acuerdo a una o algunas variables. Con este procedimiento se debe utilizar la instrucción BY, seguido por el nombre de la(s) variable(s) por la(s) que se va a ordenar.

La forma del procedimiento es:

PROC SORT opciones ;

BY opción variable opción variable...;

donde las opciones que se pueden utilizar en la instrucción PROC SORT son:

DATA = nombre del conjunto SAS que se va a ordenar. Si se omite esta opción el procedimiento se aplica al último conjunto creado.

OUT = nombre del conjunto ordenado. Si se omite esta opción el conjunto ordenado reemplaza al conjunto sin ordenar.

NODUPKEY evita que en el nuevo archivo se incluyan registros para los cuales se repiten los valores de las variables por las cuales se ordena.

Las opción de la instrucción BY es:

DESCENDING cuando se quiere ordenar una variable en forma descendente. Por defecto el ordenamiento es ascendente, por lo cual no es necesario especificarlo.

Ejemplos:

a)
PROC SORT;
BY edad;

b)

PROC SORT DATA = vacas;
BY ident DESCENDING pesocria;

En el ejemplo a) se formará un archivo ordenado del último archivo creado, en forma ascendente por edad, y el resultado se almacenará en el mismo archivo. En el ejemplo b) se especifica que se ordenará el archivo **vacas** en forma ascendente por **ident** y en forma descendente por **pesocria** dentro de **ident**; ya que no se especifica otra cosa, el archivo ordenado reemplaza al archivo **vacas**.

5.2.2. PROC TRANSPOSE

Este procedimiento se utiliza para transponer archivos SAS, cambiando las observaciones por variables. Si no se utiliza la instrucción BY en el procedimiento, la transposición será para todo el archivo. Si se desea transponer por grupos dentro de un archivo SAS, se utiliza la instrucción BY.

Ejemplo:

Se tiene el siguiente programa SAS que transpone el archivo por una variable de clasificación (TRAT).

```
DATA ejemplo;
INPUT trat rep1 rep2 rep3;
CARDS;
1       10.2    8.9     12.5
2        6.3    4.5     5.2
3       11.2    13.4    12.7
PROC TRANSPOSE OUT = SALE PREFIX = rend;
        VAR rep1 rep2 rep3;
        By Trat;
PROC PRINT;
RUN;
```

El archivo SAS transpuesto queda de la siguiente forma:

TRAT	-NAME-	REND1
1	REP1	10.2
1	REP2	8.9
1	REP3	12.5
2	REP1	6.3
2	REP2	4.5
2	REP3	5.2
3	REP1	11.2
3	REP2	13.4
3	REP3	12.7

En el programa anterior note que:

- El archivo SAS se transpone por una variable que identifica un grupo (en este caso **trat**).
- Las variables **rep1**, **rep2** y **rep3** pasan a ser observaciones para cada grupo identificado en la variable **trat**.
- Se utiliza la instrucción OUT= **sale**. Con esta instrucción se genera un nuevo archivo SAS (en este caso **sale**) el cual contiene la transposición.
- Se utiliza la instrucción PREFIX para darle un nombre a la nueva variable que se genera. Si no se utiliza esta instrucción el SAS dará el prefijo COL a las variables.

5.3. Procedimiento para listar archivos SAS

El procedimiento PRINT imprime archivos SAS. Se puede utilizar la instrucción BY para listar el archivo por grupos (de acuerdo a una o algunas variables de clasificación).

La forma del procedimiento es:

```
PROC PRINT;
  VAR V1..Vn;
  BY G1..Gn;
```

donde:

V1..Vn = las variables que se desea imprimir. Si no se utiliza la instrucción VAR el procedimiento imprime todas las variables que tiene el archivo.

G1..Gn = Las variables que identifican los diferentes grupos por los que se quiere imprimir el archivo. Si no se utiliza la instrucción BY el listado será uno solo para todo el archivo.

Si se ha utilizado la instrucción LABEL y se desea que el procedimiento imprima las etiquetas, la instrucción sería de la siguiente manera:

PROC PRINT LABEL;

5.4. Procedimientos para obtener gráficas.

El SAS cuenta con un subsistema para elaboración de gráficas de alta calidad. Sin embargo, su empleo adecuado requiere el manejo de varias instrucciones y muchas opciones que no pueden cubrirse en este manual. Aquí se explicará sobre el uso de unos procedimientos para elaboración de gráficas que ayudan en el análisis de la información, pero que no son adecuadas para documentos en su presentación final.

5.4.1. PROC PLOT

Con este procedimiento se grafica una variable contra otra. Las siguientes son las características del PLOT:

- Se puede controlar la escala de las variables y el tamaño del gráfico.
- Si no se especifica el tamaño del gráfico, el procedimiento produce un gráfico cuadrado que ocupa el ancho de la página.
- Se puede controlar el símbolo utilizado. Si no se especifica, SAS utiliza letras: 'A' para una observación, 'B' para dos observaciones, etc..
- Se puede usar el valor de una variable como símbolo.
- Se puede sobreponer más de una gráfica en la misma figura.
- Se pueden hacer gráficos por variables que identifiquen grupos

La forma del procedimiento es la siguiente:

PROC PLOT;
PLOT Y*X opciones;

donde:

Y = variable en el eje vertical
X = variable en el eje horizontal
opciones = Si se desean utilizar las opciones del procedimiento

Las opciones del procedimiento son:

OVERLAY = Si se desea sobreponer más de un gráfico

HPOS = El tamaño del eje horizontal (de 1 a 80)

VPOS = El tamaño del eje vertical (de 1 a 80)

VAXIS = Si se controla la escala en el eje vertical

HAXIS = Si se controla la escala en el eje horizontal

Si se desea hacer gráficos por alguna(s) variable(s) que identifique grupos, se debe utilizar la instrucción BY.

Ejemplo utilizando opciones:

```
PROC PLOT; BY v1;
        PLOT y1*x = '*' y2*x = '·'/OVERLAY
          VPOS = 20 HPOS = 40  VAXIS = 1 to 10 BY 1
                               HAXIS = 10 to 50 BY 5;
```

5.4.2. PROC CHART

Este procedimiento produce diagramas de barras, histogramas, diagramas de bloques y diagramas circulares.

El tipo de gráfica se define con la instrucción:

;

VBAR histograma vertical

HBAR histograma horizontal

BLOCK diagrama de bloques

PIE diagrama circular

El tipo de estadística se define con la opción TYPE = .Las siguientes son las estadísticas disponibles:

FREQ Frecuencia

PCT Porcentajes

CFREQ Frecuencias acumuladas

CPCT Porcentajes acumulados

SUM Totales

MEAN Promedios

La forma de agrupar los valores se define según el tipo de variable graficada, sea numérica o de caracteres, y por opciones:

DISCRETE	para agrupamiento categórico (variables discretas)
GROUP=	para agrupar según otra variable
MIDPOINTS=	para definir puntos medios de intervalos en variables contínuas

Ejemplo: Graficar un diagrama vertical, horizontal y de bloque para agruparla según variable COM (comunidad), utilizando porcentajes.

PROC CHART; (Produce un diagrama vertical)
VBAR OCUP/DISCRETE TYPE=PCT GROUP=COM;

PROC CHART; (Produce un diagrama horizontal)
HBAR OCUP/DISCRETE TYPE=PCT GROUP=COM;

PROC CHART; (Produce un diagrama de bloques)
BLOCK OCUP/DISCRETE TYPE=PCT GROUP=COM;

5.5. Procedimientos para estadística descriptiva

El módulo BASE del SAS, además del paso DATA y de los procedimientos anteriores, tiene procedimientos para calcular estadísticas descriptivas. Los principales de estos procedimientos son MEANS y FREQ, que se verán a continuación.

5.5.1. PROC MEANS

Este procedimiento produce estadísticas descriptivas univariadas simples para variables numéricas. Las siguientes son las características del MEANS:

- Se pueden seleccionar las estadísticas por calcular.
- Util para calcular estadísticas descriptivas por grupos: utilizando la instrucción BY, el MEANS calculará las estadísticas de acuerdo a la(s) variable(s) del BY (el archivo debe estar ordenado por esta(s) variable(s)).
- La impresión de las salidas es opcional.
- Se puede crear un archivo SAS de las estadísticas solicitadas.

Forma del MEANS:

PROC MEANS opciones:

Las opciones que tiene el MEANS son las siguientes:

- NOPRINT No imprime las salidas
- MAXDEC=n Utiliza n decimales en las salidas

Las estadíticas que calcula el procedimiento son:

Estadística	Descripción
N	Número de observaciones
NMISS	Número de observaciones con valores faltantes
MEAN	La media aritmética
STD	La desviación típica
MIN	El valor mínimo
MAX	El valor máximo
RANGE	La amplitud (máximo-mínimo)
SUM	La suma
VAR	La varianza
USS	Suma de cuadrados no corregidos
CV	El coeficiente de variación
SKEWNESS	El valor de SKEWNESS
KURTOSIS	El valor de KURTOSIS
T	El valor de T
CSS	La suma de cuadrados corregidos
STDERR	El error estándar

Las instrucciones que se pueden utilizar en el MEANS son las siguientes:

- **BY** variables ; La lista de variables para calcular las estadísticas por grupo.
- **VAR** variables; La lista de las variables a las que se le calcularán las estadísticas.
- **OUTPUT OUT=** archivo ; El nombre del archivo SAS que tendrá las estadísticas calculadas (opcional).

Ejemplo: Se tiene el siguiente programa SAS:

```
DATA ejemplo;
INPUT nitrogen rendim @@;
CARDS;
0    8.5   100   10.2   0    6.9   200   14.3   200   15.5
100  9.3   300   19.5   300  21.3  100   8.5    0     4.3
400  12.5  400   11.3   400  10.5
PROC SORT; BY nitrogen;
PROC MEANS MEAN STD SUM NOPRINT;
   BY nitrogen; var rendim;
OUTPUT OUT = ejemplo   MEAN = MEDREND
                       STD  =   STDREND
                       SUM  =   SUMREND;
RUN;
```

Al ejecutar el programa, el archivo SAS queda de la siguiente manera:

NITROGEN	MEDREND	STDREND	SUMREND
0	6.57	2.12	19.7
100	9.33	0.85	28.0
200	14.90	0.85	29.8
300	20.40	1.27	40.8
400	11.433	1.01	34.3

5.5.2 PROC FREQ

El procedimiento FREQ produce tablas de frecuencias en una o más dimensiones. Las siguientes son las características del procedimiento.

- Produce frecuencias absolutas y relativas, simples y acumuladas.
- Calcula medidas de asociación y pruebas estadísticas para tablas de dos dimensiones o entradas.
- Produce salidas opcionales a un archivo SAS.

Forma del procedimiento FREQ:

PROC FREQ opciones;
TABLES tablas solicitadas /opciones;
WEIGHT variables de ponderación ;
BY para construir las tablas por grupos;

Ejemplo: Se tiene el siguiente programa:

```
Data ejemplo;
INPUT sexo niveledu @@;
CARDS;
1 1  2  3  1  3  2 2  2  4 1  3  1 2  1  3  2  4  1  5  2  5
1 2  1  3  2  4  2 3  1  1 2  1  2 4  1  1  2  5
PROC FORMAT;
      VALUE sexo        1 = 'Femenino'
                        2 = 'Masculino'; (Cont. en la sig. pág.)
      VALUE niveledu    1 = 'Sin escolaridad'
                        2 = 'Primaria'
                        3 = 'Secundaria'
                        4 = 'Universitaria'
                        5 = 'Otros';
PROC FREQ;
      TABLES sexo niveledu; (Produce listas)
      FORMAT sexo sexo. niveledu niveledu.;
      TABLES sexo * niveledu ; (Produce tabla de dos dimensiones)
      FORMAT sexo sexo. niveledu niveledu.;
RUN;
```

La siguiente es la salida que produce el programa:

SEXO	FREQUENCY	PERCENT	CUMULATIVE FREQUENCY	CUMULATIVE
Femenino	10	50.0	10	50.0
Masculino	10	50.0	20	100.0

NIVELEDU	FREQUENCY	PERCENT	CUMULATIVE FREQUENCY	CUMULATIVE PERCENT
Sin escolaridad	4	20.0	4	20.0
Primaria	3	15.0	7	35.0
Secundaria	6	30.0	13	65.0
Universitaria	4	20.0	17	85.0
Otros	3	15.0	20	100.0

TABLE OF SEXO BY NIVELEDU

SEXO FREQUENCY PERCENT ROW PCT COL PCT	NIVELEDU: SIN ESCOLARIDAD	PRIMARIA	SECUNDARIA	UNIVERSITARIA	OTROS	TOTAL
Femenino	3 15.00 30.00 75.00	2 20.00 20.00 66.67	4 20.00 40.00 66.67	0 0.00 0.00 0.00	1 5.00 10.00 33.33	10 50.00
Masculino	1 5.00 10.00 25.00	1 5.00 10.00 33.33	2 10.00 20.00 33.33	4 20.00 40.00 100.00	2 10.00 20.00 66.67	10 50.00
TOTAL	4 20.00	3 15.00	6 30.00	4 20.00	3 15.00	20 100.00

Las siguientes son las opciones del procedimietno FREQ:

EXPECTED	NOROW	NOPRINT
DEVIATION	NOCOL	NOFREQ
CELLCHI2	CUMCOL	MISSING
CHISQ	LIST	NOPERCENT
ALL	NOCUM	SPARSE

CAPITULO VI. ALGUNOS PROCEDIMIENTOS PARA ANALISIS ESTADISTICOS DE DATOS EXPERIMENTALES

En este capítulo se estudian algunos de los procedimientos estadísticos más utilizados en el análisis de datos experimentales, con ejemplos del campo agropecuario y forestal. Se explica el uso del procedimiento, la manera de programarlo en SAS y la interpretación de los resultados que se despliegan en la pantalla.

6.1. Correlación

El científico a veces está interesado en la relación que existe entre dos o más variables. Por ejemplo, se quiere saber la interdependencia que existe entre la edad de un animal y su peso. Supongamos que se tienen los siguientes datos:

edad (años)	1	3	2	4	2	3	2	1	2	3	4	5
peso (kgs)	20	40	32	48	21	39	34	17	31	37	51	55

El programa SAS sería:

```
DATA pesos;
INPUT edad peso;
CARDS;
1       20
3       40
.               (resto de los datos)
.
5       55
PROC CORR; VAR edad peso;
RUN;
```

Si el conjunto de datos tiene más variables, pueden incluirse todas en la lista de variables. La salida de este programa es:

```
VARIABLE      N       MEAN      STD DEV       SUM       MINIMUM       MAXIMUM

EDAD         12      2.667       1.2309     32.00        1.000         5.000
PESO         12     35.417      12.2062    425.00       17.000        55.000

PEARSON CORRELATION COEFICIENTS / PROB > |R| UNDER HO:RHO=0 / N = 12

                  EDAD         PESO
      EDAD        1.000        0.9600 (1)
                  0.000        0.001  (2)

      PESO        0.9600       1.0000
                  0.0001       0.0000
```

Como se puede ver, SAS produce una matriz de correlación con el coeficiente de correlación (1) entre EDAD y PESO de 0.96. El valor que se encuentra debajo del coeficiente de correlación es la probabilidad (2) de error cuando se rechaza la hipótesis de que el coeficiente de correlación es cero. En el ejemplo vemos que la probabilidad de que la correlación sea cero es muy pequeña (0.001). Por esto se acepta que la correlación es diferente de cero, con una probabilidad de error de 0.001, o un "nivel de significancia" de 1 - 0.001 = 0.999.

6.2. Análisis de regresión

6.2.1. Regresión lineal simple

Supongamos que un técnico forestal está interesado en determinar el crecimiento en diámetro de un pino a partir del volumen de la copa. Mediante un análisis de regresión se puede averiguar si existe una relación significativa entre las dos variables y expresar, por medio de una ecuación, la relación entre el crecimiento del árbol y el volumen. Esta ecuación permite predecir cuál sería el crecimiento del árbol para determinado volumen de copa.

Supongamos que el técnico obtuvo los siguientes datos con las mediciones:*

Volúmen de la copa (x)	Crecimiento (Y)
22	.36
6	.09
93	.67
62	.44
84	.72
14	.24
52	.33
69	.61
99	.64
98	.65
41	.47
85	.60
90	.51

Si la relación entre las dos variables fuese lineal, el modelo que predeciría el crecimiento sería:

Crecimiento = a + b*volumen + e

Se pueden utilizar varios procedimeintos SAS para ajustar un modelo de regresión. Uno de esos es el procedimiento REG, que permite ajustar modelos lineales, utilizando diferentes estrategias para seleccionar los modelos. Otro procedimiento es RSREG que permite ajustar modelos de superficies de respuesta; otro procedimeitno es NLIN que se emplea para modelos no lineales. En este manual se va a emplear el procedimiento GLM (generalizad linear models), que también se empleará para análisis de varianza de diseños estadísticos.

El programa SAS para ajustar este modelo de regresión, utilizando el procedimiento GLM, sería:

```
DATA regre;
INPUT crecim volumen;
CARDS;
22  0.36
 6  0.09
 .  .   (resto de los datos)
 .  .
90  0.51
PROC GLM;
      MODEL crecim = volumen;
RUN;
```

* "Métodos estadísticos elementales para Técnicos Forestales F.Freese"

Note que es necesario escribir la palabra **model** y luego Y = X. No deben incluirse las constantes **a** y **b** del modelo.

La salida de SAS incluye un análisis de varianza, que nos dice si la regresión es significativa, así como los valores calculados: **a** (intercepto) y **b** (coeficiente de regresión). En el cuadro que se presenta a continuación aprecen unos números entre paréntesis, escritos en negrilla, a la derecha de algunos valores, a los que se hará referencia más adelante. La salida corespondiente al ejemplo es la siguiente:

GENERAL LINEAR MODELS PROCEDURE

DEPENDENT VARIABLE: CRECIM

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR > F	R-SQUARE	C.V.
MODEL	1	0.34951099	0.34951099	48.93	0.0001**(1)**	0.81647**(2)**	17.36
ERROR	11	0.07856593	0.00714236	ROOT MSE			Y MEAN
TOTAL	12	0.42807692		0.0845235			0.48692

SOURCE	DF	TYPE I SS	F VALUE	PR > F	DF	TYPE III SS	F VALUE	PR > F
VOLUMEN	1	0.34951099	48.93	0.0001	1	0.34951099	48.93	0.0001

PARAMETER	ESTIMATE	T FOR HO: PARAMETER=0	PR > |T|	STD ERROR OF ESTIMATE
INTERCEPT	0.162439**(3)**	3.13	0.0097	0.051197153
VOLUMEN	0.005176**(4)**	7.00	0.0001**(5)**	0.00073989

Lo que interesa de esta salida es:

- Ver si la regresión es significativa, es decir si hay relación entre crecimiento y volumen. El análisis de varianza en este caso prueba la hipótesis H_0:ß=0 contra H1:ß≠0. El valor 0.0001 que se observa debajo de PROB > F (1), significa que se puede rechazar la hipótesis de que ß=0 con una probabilidad de error de 0.0001 o menor. Esto lleva a aceptar que el valor de ß es diferente de cero y se puede representar satisfactoriamente con el valor de b calculado.

- El R^2 (R Square). El R cuadrado (2) indica qué proporción de la variación en Y se encontró asociada con X. En este caso el R^2 fue 0.81. Esto quiere decir que el modelo lineal explica el 81% de la variación.

- Los valores estimados para "**a**" (intercepto) y para "**b**" (coeficiente de regresión). En este caso "a" (**3**) es el valor que aparece a la par de INTERCEPT (0.16244), y el de "b" (**4**) el que aparece a la par de VOLUMEN (0.00518) en la última línea. Entonces la ecuación para predecir el crecimiento sería:

$$\text{Crecimiento} = 0.16244 + 0.00518(\text{Volumen})$$

- El nivel de significancia para el cual rechazaríamos la hipótesis que el parámetro ß (**4**) sea 0 (es decir, que no haya regresión). En este caso rechazamos la hipótesis con nivel 0.0001 (0.01%). Nótese que la prueba (**5**) coincide con la (**1**) en este caso particular.

Si se desea hacer un gráfico de los datos superponiendo la línea de regresión, simplemente se añaden las siguientes líneas al programa:

```
Output Predicted = py;
Proc Plot;
Plot crecim*volumen= '*' py*volumen= '+' / overlay;
```

En el gráfico los datos serán denotados con '*' y la línea de regresión (datos predichos) con '+'.

6.2.2. Ajuste de modelos cuadráticos y cúbicos

Algunas veces la relación entre dos variables es cuadrática o cúbica. Una relación típica es, por ejemplo, cuando la variable independiente (X) es un fertilizante aplicado en varias dosis. Al aumentar la dosis, aumenta el rendimiento, pero el incremento se hace cada vez menor hasta que se llega a un punto en que el rendimiento empieza a bajar (incrementos negativos) debido a que dosis altas del fertilizante producen efectos dañinos.

Para ajustar un modelo cuadrático (usando los datos del ejemplo anterior) simplemente escribimos las siguientes líneas:

```
PROC GLM;
Model Crecim = volumen volumen*volumen;
```

Estas dos líneas de programa ajustan el siguiente modelo:

$$\text{Crecimiento} = a + b*\text{volumen} + c*\text{volumen}^2 + e$$

La salida de SAS es similar a la mostrada en la sección 6.2.1, excepto que ahora se tiene un parámetro más (c), con su probabilidad de que sea significativo. Para saber si el modelo cuadrático se ajusta mejor que el lineal, debemos fijarnos en la reducción producida en el error. Si la reducción es grande entonces quiere decir que el modelo cuadrático es mejor. También podemos ver si el parámetro c es estadísticamente significativo, lo mismo que si el R cuadrado aumentó considerablemente.

El R cuadrado del modelo cuadrático siempre será más alto que el del modelo lineal (lo mismo aplica para el modelo cúbico), pero se debe considerar si este aumento en el R cuadrado es suficiente como para complicar el modelo con un factor más.

Si deseamos ajustar un modelo cúbico, es decir,

Crecimiento = a + b*volumen + c*volumen^2 + d*volumen^3 + e

simplemente escribimos en nuestro programa lo siguiente.

```
PROC GLM;
MODEL crecim=volumen volumen*volumen volumen*volumen*volumen;
```

La salida incluirá el parámetro adicional (d), así como la probabilidad de que sea significativo. Las mismas observaciones anteriores aplican aquí para saber si el modelo cúbico es mejor que el cuadrático.

6.2.3. Regresión Múltiple

Algunas veces tenemos un modelo de regresión en el cual la variable depende de varias variables independientes (X). Podemos entonces ajustar un modelo de regresión múltiple. Supongamos que creemos que el rendimiento, en kgs, de parcelas de maíz depende de la cantidad de nitrógeno aplicado al suelo, la altura promedio de las plantas (cm) y el número de mazorcas.

En este caso tendríamos un modelo de regresión múltiple:

Rendimiento = a + b_1*N +b_2*Altura + b_3*Mazorcas + e

Para ajustar este modelo en SAS necesitamos hacer el siguiente programa:

```
DATA regres;
INPUT  rend m altura mazorcas;
cards;
18.56   120   120   54
21.34   126   124   68
25.58   128   138   75
33.28   133   140   88
13.23   110   120   28
12.28   112   135   27
11.19   115   129   29
12.12   119   127   45
17.28   112   127   37
17.12   110   149   31
17.28   122   158   51
18.28   121   148   67
10.20    99   128   38
PROC GLM;
MODEL rend = m altura mazorcas;
RUN;
```

Los resultados son los siguientes:

```
                    GENERAL LINEAR MODELS PROCEDURE

DEPENDENT VARIABLE: REND

SOURCE      DF     SUM OF SQUARES      MEAN SQUARE     F VALUE(1)    PR>F     R-SQUARE     C.V.

MODEL        3      393.70738          131.23579          11.21     0.0021    0.789719   19.48198

ERROR        9      104.83355           11.64818                   ROOT MSE          REND MEAN

TOTAL       12      498.54097                                       3.4129          17.124615

SOURCE      DF    TYPE I SS    F VALUE    PR > F    DF    TYPE III SS    F VALUE    PR > F

M            1   334.43746      28.71     0.0005     1     9.905102        0.85      0.0404
ALTURA       1     2.38746       0.20     0.6615     1     1.351637        0.12      0.5281
MAZORCAS     1    56.88246       4.88     0.0545     1    56.88246         4.88      0.0545

PARAMETER       ESTIMATE        T FOR HO:          PR > |T|        STD ERROR OF
                                PARAMETER = 0                       ESTIMATE

INTERCEPT    -18.67629(2)       -0.84          0.42421(3)        22.30819
M              0.1896 (2)        0.92          0.3805 (3)         0.20556
ALTURA         0.0291 (2)        0.34          0.7412 (3)         0.08535
MAZORCAS       0.20439(2)        2.21          0.0845 (3)         0.09249
```

La salida es similar a la de la sección 2.1. En este caso tenemos que:

(1) El valor de F de 11.21 se usa para probar la hipótesis nula de que
HO: $b_1 = b_2 = b_3 = 0$ (o sea que no hay regresión)

(2) Los parámetros de este modelo serían:

rendimiento = -18.68 + 0.19(M) + 0.0291*altura + 0.2044*mazorcas

(3) El nivel de significación para probar que cada parámetro es diferente de cero, ó sea, si está verdaderamente contribuyendo al modelo. En este caso podemos ver que el único parámetro significativo es el número de mazorcas. Por lo tanto, este modelo necesitaría incluir solo la variable mazorcas porque las otras no explican la variación en el rendimiento.

Como se puede notar en la salida, tenemos dos tipos de sumas de cuadrados. Vea la sección 6.7 para una explicación sobre este tema.

6.3. Análisis de Medias

Los siguientes son algunos de los procedimientos SAS para comparar medias:

6.3.1. Comparación de dos medias muestrales

Supongamos que se hace un experimento para probar dos tratamientos (por ejemplo: dos variedades de arroz, dos procedencias de pino hondureño, dos razas de cerdos, etc.).

La prueba "t" de Student proporciona el método para comparar las medidas de dos muestras. Considere, por ejemplo, que se tienen datos de dos procedencias de pino hondureño, las que van a compararse sobre la base de su producción en volumen durante cierto período. Los volúmenes (en m^3) de 11 árboles de cada procedencia fueron los siguientes:

Procedencia 1:	11	5	9	8	10	11	10	8	11	8	8
Procedencia 2:	9	6	9	9	13	8	6	5	6	10	7

Para probar la hipótesis de que no hay diferencia entre las medias de las procedencias (i.e. hipótesis nula, $H_0: \mu_1 = \mu_2$), SAS realiza la prueba de "t" bajo las alternativas: a) varianzas iguales; b) varianzas diferentes. Además SAS hace una prueba de igualdad de varianzas para que, con base en el resultado de esa prueba, el usuario escoja la prueba de "t" correspondiente.

El programa SAS para analizar estos datos sería como sigue:

```
DATA medias;
IMPUT proced volumen;
Cards;
1       11
1        5
.
.                       (resto de las observaciones)
2       9
2       6
.
.                       (resto de las observaciones)
PROC TTEST;
CLASS proced;
VAR volumen;
RUN;
```

La instrucción CLASS sirve para indicar la variable que identifica los dos grupos que se van a comparar. En este ejemplo se trata de dos procedencias, que se especifican en la variable **proced**. Esta variable toma valores 1 ó 2 unicamente. El procedimiento TTEST hace una prueba de "t" para la(s) variable(s) indicada(s) en la instrucción VAR, usando la variable indicadora incluída en la instrucción CLASS como criterio de agrupamiento.

La salida que produce SAS es la siguiente:

TTEST PROCEDURE

VARIABLE: VOLUMEN

PROCED	M	MEAN	STD DEV	STD ERROR	MINIMUM	MAXIMUM	VARIANCES	T	DF	PROB>|T|
1	11	9.00	1.844	0.5559	5.00	11.00	UNEQUAL	1.11	19	0.2775
1	11	8.00	1.323	0.7000	5.00	13.00	EQUAL	1.11	20	0.2768 **(2)**

FOR HO: VARIANCES ARE EQUAL, F'= 1.59 WITH 10 AND 10 DF PROB > F'= 0.4775 **(1)**

Para interpretar esta salida lo primero que se debe examinar es la prueba de varianzas (última línea). La probabilidad de obtener un valor mayor que F = 1.59 es muy alta (0.4775)(1), lo que indica que no hay diferencias significativas entre las varianzas. Para poder rechazar H_0 y aceptar H_1, el valor calculado debe ser menor o igual que la probabilidad de error establecida (por ejemplo, 0.01).

Al aceptar que las varianzas son iguales, se usa la prueba de "t" para varianzas iguales (equal). El valor obtenido para t es 1.11 y el valor para la probabilidad de error es de 0.2768(2) (valor prob > _T_). En este caso la probabilidad de error es alta (mayor que 0.05), por lo tanto no se puede rechazar la hipótesis de igualdad de promedios.

6.3.2 Comparación de medias de datos apareados

Considere un experimento donde se aplican dos drogas somníferas (A y B) a un grupo de individuos y se midió el número de horas extra de sueño. La droga A se aplicó a cada individuo durante los primeros ocho días y la droga B los siguientes ocho días. Los siguientes datos corresponden al promedio de horas extra de sueño para cada individuo para cada droga:

Individuo	1	2	3	4	5	6	7	8	9	10	11	12
droga A	2	1	0	2	1	2	1.3	2	3	2	1	0
droga B	3	2	1	3	3	4	3.8	2	4	4	3	3

Se quiere saber si la droga B produce efectos diferentes que la droga A. El método para comparar medias apareadas se basa en calcular la diferencia dentro de cada par y hacer una prueba de t usando la varianza de las diferencias. Entre más cerca de cero esté la media de las diferencias más parecidas son las drogas en los efectos que se evalúan. El valor de t, que relaciona el promedio de las diferencias con su desviación estándar, es el criterio adecuado para someter a prueba la igualdad de los promedios (diferencia = 0).

Para resolver esto se recurre al procedimiento MEANS:

```
DATA drogas;
INPUT drogaa drogab;
diferen = drogaa - drogab;
CARDS;
2       3
1       2
.               (resto de las observaciones)
.
2       4
1       32 ;
PROC MEANS MEAN T PRT;
        VAR diferen;
RUN;
```

Note que cada columna de datos corresponde a una de las drogas. También hay que notar que en el programa se creó la variable **diferen** para calcular la diferencia entre las dos drogas. Las opciones usadas en el PROC MEANS son: mean (media), T (el valor de "t") y PRT (la probabilidad de que la media sea cero).

La salida de este programa es la siguiente:

SAS

VARIABLE	MEAN	T	PROB > _T_
DIFF	-1.54166	-6.37	0.001

La salida nos muestra la media de las diferencias, el valor de lo calculado y el nivel de significancia. La probabilidad de obtener un valor mayor que T = -6.37 es muy baja (0.001), lo cual indica que la diferencia es altamente significativa. Se puede concluir que la efectividad de las drogas es diferente con 99.9% de seguridad.

6.3.3. Comparación de Varias Medias: Análisis de Varianza

Si se desea hacer inferencia estadística de más de dos medias, se debe utilizar el análisis de varianza. El sistema SAS ofrece dos formas para hacer análisis de varianza: PROC ANOVA para diseños ortogonales balanceados y PROC GLM para diseños no balanceados.

El análisis de varianza parte la variación total dentro de las observaciones en porciones asociadas con ciertos factores que están definidas por el esquema de clasificación de los datos. Estos factores son las fuentes de variación. Por ejemplo, la variación en la producción de leche en vacas puede ser fraccionada en porciones asociadas con diferencias entre hatos, diferencias genéticas y otras diferencias. Estas divisiones se hacen en términos de sumas de cuadrados asociadas a los grados de libertad. A continuación se presentan diferentes diseños experimentales y la forma en que se analizan en SAS.

6.3.3.1. Diseño Completamente al Azar

Este tipo de diseño experimental asume que las unidades experimentales de una población se asignan al azar a grupos que generalmente son llamados tratamientos. La hipótesis nula es que las poblaciones estudiadas tienen la misma media.

Por ejemplo, se tienen datos de cuatro variedades de arroz y se quiere saber si se debe aceptar que los promedios de ellas (en la población) no difieren entre si. Los rendimientos (en kgs) observados de cuatro parcelas de cada variedad son:

Variedad	Rendimiento			
1	23.5	31.2	18.2	27.8
2	24.4	26.3	28.2	28.3
3	32.3	28.3	33.0	34.5
4	38.9	42.1	32.3	38.2

El modelo ajustado a estos datos sería:

$$\text{Rendimiento} = \mu + t_i + e_{ij}$$

donde:

μ	es el promedio de las medias de las variedades
t_i	es el efecto del tratamiento i
e_{ij}	es el error experimental

La tabla del análisis de varianza para este experimento es:

Fuente de variación	**gl**
Variedad	3
Error	12
Total	15

Este análisis de varianza puede hacerse, lo mismo que comparaciones múltiples de las medias de las variedades, usando PROC GLM con la opción para la prueba de DUNCAN. Esto se haría con el siguiente programa:

```
DATA varianza;
INPUT variedad rend;
CARDS;
1       23.5
1       31.2
..                  (más datos)
2       24.4
2       26.3
..                  (más datos)
3       32.3
3       28.3
..                  (más datos)
4       38.9
4       42.1
..                  (resto de los datos)
;
PROC GLM;
CLASS variedad;
MODEL rend = variedad;
MEANS variedad/DUNCAN;
RUN;
```

Los datos están clasificados sólo de acuerdo a los valores de variedad; por lo tanto **variedad** es la única variable que debe aparecer en la instrucción CLASS. La variable rendimiento es la variable de respuesta que va a analizarse; por lo tanto, **rend** aparece al lado izquierdo del signo '=' en la lína MODEL. La única fuente de variación además del ERROR (residuo), es VARIEDADES; por lo tanto, se digita **variedad** a la derecha del signo "=". La salida producida por la instrucción MODEL es la siguiente:

GENERAL LINEAR MODELS PROCEDURE

DEPENDENT VARIABLE: REND

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR>F	R-SQUARE	C.V.
MODEL	3	681.7818	227.2606	26.48	0.868773	0.89973	9.59
ERROR	12	102.9825	8.5818767		ROOT MSE		REND
MEAN							
TOTAL	15	784.76937			2.9295		
29.587500							

SOURCE	DF	TYPE I SS	F VALUE	PR>F	DF	TYPE III	F VALUE	PR>F
VARIEDAD	3	681.7818	26.48	0.0001	3	681.7818	26.48	.0001

Note que la suma de cuadrados del MODEL es la misma que la suma de cuadrados de VARIEDAD. Esto se debe a que la variedad es la única fuente de variación en el modelo, aparte del ERROR. Recuerde que la suma de las sumas de cuadrados de las fuentes de variación es igual a la suma de cuadrados total. El valor de p de 0.0001 indica que hay diferencia altamente significativas entre las medias de las variedades.

Los resultados del PROC GLM se pueden resumir en la siguiente tabla:

FUENTE	**GL**	**SC**	**CM**	**F**	**P**
Variedades	3	681.78	227.26	26.5	0.0001
Error	12	102.88	8.58		
Total	15	784.76			

La instrucción MEANS (para realizar prueba de Duncan a las medias) produce la siguiente salida:

MEANS WITH THE SAME LETTER ARE NOT SIGNIFICANTLY DIFFERENT

DUNCAN	**GROUPING**	**MEAN**	**N**	**VARIEDAD**
	A	37.875	4	4
	B	32.025	4	3
	C	26.800	4	2
	D	20.175	4	1

N indica el número de observaciones usadas para calcular cada media. Como lo dice la salida de SAS, las medias que tengan la misma letra en la columna GROUPING no son significativamente diferentes. En este caso todas las medias son diferentes ya que ninguna tiene la misma letra. Las medias de cada variedad aparecen en la columna MEAN.

Es posible hacer otro tipo de comparación múltiple, escribiendo el nombre de la prueba después del "/" en la línea MEANS (p.ej. DUNNETT, TUKEY, SCHEFFE).

Algo importante que se debe tener en cuenta es que la mayoría de las pruebas de comparación múltiple requieren que los tratamientos tengan el mismo número de observaciones.

Con frecuencia es más adecuado utilizar contrastes en vez de pruebas de comparaciones múltiples. Además, si los tratamientos son de tipo cuantitativo en lugar de cualitativo (ej. niveles de algún fertilizante), en vez de usar una prueba de comparaciones múltiples es preferible ajustar algún modelo de regresión a los datos (Ver sección 6.4. "Análisis de varianza con tratamientos cuantitativos").

6.3.3.2. Diseño de Bloques Completos al Azar

Este diseño se emplea cuando se supone que las unidades experimentales pueden formar grupos relativamente homogéneos, generalmente llamados bloques. Los tratamientos se asignan al azar a las unidades experimentales dentro de los bloques.

Supongamos que en un experimento se sembraron cinco variedades de maíz en tres bloques. Se tienen los siguientes datos, en kg/parcela.

Bloque	Tratamiento				
	1	2	3	4	5
1	25.4	34.3	23.4	28.3	12.2
2	34.2	38.8	28.2	28.2	14.2
3	21.2	32.1	21.2	26.2	14.5
4	39.9	45.8	34.3	32.3	19.3

Se emplea el siguiente modelo para cualquier rendimiento Yij perteneciente al bloque i y al tratamiento j:

$$Y_{ij} = \mu + \beta_i + t_j + e_{ij}$$

donde:

μ es la media general de los tratamientos
β_i es el efecto del bloque i
t_j es el efecto del tratamiento j
e_{ij} es el error experimental

La tabla del análisis de varianza para este experimento es:

Fuente de variación	gl
Bloques	3
Tratamientos	4
Error	12
Total	19

El programa en SAS sería:

```
DATA varianza;
INPUT bloque trat rend;
CARDS;
1    1    25.4
1    2    34.3
1    3    33.4
1    4    28.3
1    5    12.2
2    1    34.2
.    .  .        (resto de los datos)
.
;
PROC GLM;
CLASS bloque trat;
MODEL rend = bloque trat;
MEANS trat/DUNCAN;
```

La instrucción CLASS especifica el nombre de las variables que identifican los criterios de clasificación que corresponden al diseño. En este caso los datos están clasificados de acuerdo al bloque y el tratamiento al que pertenecen. La variable de respuesta es el rendimiento (**rend**), la cual debe ir a la izquierda del signo "=". En este caso, las dos fuentes de variación además del Error son **bloque** y **trat**, por lo cual deben aparecer a la derecha del signo "=".

Los resultados del análisis son los siguientes:

GENERAL LINEAR MODELS PROCEDURE

DEPENDENT VARIABLE: REND

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR>F	R-SQUARE	C.V.
MODEL	7	1453.7390	207.6770000	26.37	0.0001	0.93896	10.13
ERROR	12	94.5010	7.87515000		ROOT MSE		REND MEAN
TOTAL	19	1548.2400			2.8063		27.70

SOURCE	DF	TYPE I SS	F VALUE	PR>F	DF	TYPE III	F VALUE	FR>F
BLOQUE	3	377.304	15.97	0.0020	3	337.304	15.97	0.0021
TRAT	4	1076.435	34.17	0.0001	4	1076.435	34.17	0.0001

Los resultados del PROC GLM se pueden resumir en la siguiente tabla:

FUENTE	**GL**	**SC**	**CM**	**F**	**P**
Bloque	3	377.30	125.8	15.97	0.0021
Trat	4	1076.44	269.11	34.17	0.0001
Error	12	94.50	7.88		
Total	19	1548.24			

De estos resultados se concluye que los bloques son significativamente diferentes; esto indica que la formación de bloques surtió efecto. También se puede concluir que hay diferencias reales entre tratamientos.

El resultado de la instrucción MEANS es el siguiente:

DUNCAN GROUPING	MEAN	N	TRAT
A	37.75	4	2
B	30.175	4	1
B B	28.755	4	4
B	26.775	4	3
C	15.050	4	5

En este caso la prueba de Duncan indica que la media del tratamiento 2 es diferente a las demás. En el extremo inferior, la media del tratamiento 5 también es diferente del resto. Las medias de los tratamientos 1, 3 y 4 no son diferentes entre sí, pero sí diferentes a la media del tratamiento 2 y del 5.

Si el experimento consta de tratamientos cuantitativos (ej. niveles de algún fertilizante) ver "Curvas de respuesta", en la Sección 6.4.

6.3.3.3. Diseño de Cuadrado Latino

En el diseño de bloques al azar se busca aislar una fuente de variación extraña reconocible usando bloques. Si el bloqueo funciona, el cuadrado medio del error se reduce dando una prueba más sensible que la obtenida con el diseño completamente al azar.

No obstante, algunas veces hay gradientes en dos sentidos que deben ser aisladas formando bloques en dos direcciones. Por ejemplo, en un campo, los gradientes de fertilidad pueden existir en dos sentidos: paralelamente y en ángulo recto a los surcos del arado. El empleo del diseño de Bloques al Azar aísla solamente una de estas fuentes de variación, mientras la otra forma parte del término de error, lo cual reduce la precisión de la prueba.

Supongamos que tenemos cinco tratamientos (A,B,C,D,E) en un diseño de Cuadrado Latino, distribuídos de la siguiente manera:

		Columnas				
		1	**2**	**3**	**4**	**5**
Filas	1	C(32)	A(23)	B(24)	E(34)	D(28)
	2	A(28)	C(36)	D(33)	B(32)	E(38)
	3	E(40)	D(37)	A(31)	C(39)	B(26)
	4	D(39)	B(31)	E(43)	A(33)	C(41)
	5	B(32)	E(45)	C(43)	D(40)	A(35)

Los valores a la derecha de cada letra (tratamientos) son los rendimientos de maíz por parcela.

Se escribe el siguiente programa:

```
DATA latino;
INPUT col fila trat $ rend;
CARDS;

1     1     C     32
1     2     A     28
1     3     E     40
1     4     D     39
1     5     B     32
2     1     A     23
2     2     C     36
.     .     .     .
.     .     .     .
5     4     C     41
5     5     A     35
;
PROC GLM;
CLASS col fila trat;
MODEL rend = col fila trat;
MEANS trat/DUNCAN;
RUN;
```

La salida de SAS es la siguiente:

GENERAL LINEAR MODELS PROCEDURE

DEPENDENT VARIABLE: REND

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR>F	R-SQUARE	C.V.
MODEL	12	835.12	69.59333	26.84	0.0001	0.964075	4.665
ERROR	12	31.1200	2.59333		ROOT MSE		REND MEAN
TOTAL	24	866.24			1.6104		34.52

SOURCE	DF	TYPE I SS	F VALUE	PR>F	DF	TYPE III	F VALUE	FR>F
COL	4	11.04	1.06	0.4159	4	11.04	1.06	0.4159
FILA	4	347.84	33.53	0.0001(1)	4	347.84	33.53	0.0001
TRAT	4	476.24	45.91	0.0001(2)	4	476.24	45.91	0.0001

Los resultados del PROC GLM se pueden resumir en la siguiente tabla:

Fuente	gl	SC	F	P
Col	4	11.04	1.06	0.4159 (1)
Fila	4	347.84	33.53	0.0001 (2)
Trat	4	476.24	45.91	0.0001
Error	12	78.32		
Total	24	866.24		

Como se puede ver, el efecto de fila es altamente signifi-cativo **(1)**, lo mismo que los tratamientos **(2)**. En este caso no se ganó precisión al emplear columnas como fuente de variación.

La salida de la prueba de Duncan es la siguiente:

DUNCAN	GROUPING	MEAN	N	TRAT
	A	40.00	5	E
	A			
	A	38.20	5	C
	A			
	A	35.40	5	D
	B	30.00	5	A
	B			
	B	29.00	5	B

De la prueba de Duncan vemos que las medias de los trata-mientos E, C y D no difieren entre sí. Las medias de A y B son iguales entre sí pero diferentes a las medias de E, C y D.

6.3.3.4. Arreglos Factoriales

Los arreglos factoriales nos permiten estudiar varios fac-tores simultáneamente con muy poco trabajo adicional; aumentan la precisión, la cobertura y utilidad de los resultados al proveer información sobre las interacciones entre los factores en prueba.

Como ilustración consideremos un arreglo factorial que envuelve tres variedades de caña de azúcar (V) y tres niveles de nitrógeno (N), conducido usando un diseño de bloques completos al azar con dos repeticiones.

Cuando analizamos los resultados del experimento podemos hacer las siguientes comparaciones:

i) comparaciones entre variedades
ii) comparaciones entre niveles de nitrógeno
iii) la interacción entre variedades y nitrógeno

Las comparaciones **i** y **ii** son entre efectos principales. La presencia o ausencia de efectos principales no dice nada acerca de la presencia o ausencia de la interacción o vice-versa; por tanto, debemos considerarlas separadamente.

Una interacción significativa implica que los efectos de los factores no son independientes entre sí. En este caso, no podemos concluir que el mejor tratamiento corresponde a la combinación de la variedad con el mayor promedio y el nivel con el promedio más alto. Es necesario estudiar más a fondo cómo se comporta cada variedad con los diferentes niveles de fertilización o los nive-les de fertilización con cada variedad.

Supongamos que el experimento dió los siguientes resultados de rendimiento de caña en toneladas por hectárea.

		Variedad		
Fertilización		V_0	V_1	V_2
Bloque I	f_0	66.52	61.45	68.60
	f_1	68.98	62.55	64.54
	f_2	75.95	57.90	68.09
Bloque II	f_0	56.50	53.45	58.50
	f_1	58.95	51.55	54.54
	f_2	66.95	47.90	58.19

Para represenar el rendimiento, y, se emplea el siguiente modelo:

$$Y_{ijk} = \mu + \beta_i + V_j + f_k + Vf_{jk} + \varepsilon_{ijk}$$

donde:

μ	es la media general
β_i	es el efecto del bloque i
V_j	es el efecto principal de la variedad j
f_k	es el efecto principal del nivel k de fertiliz.
Vf_{jk}	es la interacción de la variedad j por el nivel k en el bloque i
ε_{ijk}	es el error experimental

La tabla del análisis de varianza en este caso sería:

Fuente de Variación	gl
Bloque	1
Variedad	2
fertiliz	2
Variedad*fertiliz	4
Error	8
Total	17

Puede emplearse al siguiente programa SAS:

```
DATA factoria;
INPUT bloque var fert rend;
CARDS;
1    1    0     66.52
1    1    1     68.98
1    1    2     75.95
1    2    0     61.45
.    .    .     .        (Resto de los datos)
.    .    .     .
2    3    0     58.50
2    3    1     54.54
2    3    2     58.19
;
PROC GLM;
CLASS bloque variedad fert;
MODEL rend = bloque variedad fert variedad*fert;
RUN;
```

Los resultados del programa son:

GENERAL LINEAR MODELS PROCEDURE

DEPENDENT VARIABLE: REND

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR>F	R-SQUARE	C.V.
MODEL	9	880.73896111	97.8598846	279.66	0.0001	0.996832	0.97
ERROR	8	2.7994000	0.3499250		ROOT MSE		REND MEAN
TOTAL	17	883.53836111			0.59154459		61.172778

SOURCE	DF	TYPE I SS	F VALUE	PR>F	DF	TYPE III	F VALUE	PR>F
BLOQUE	1	430.71125	1230.87	0.0001	1	430.71125	1230.87	0.0001
VAR	2	297.92834	435.70	0.0001	2	297.92834	425.70	0.0001
FERT	2	17.04814	24.36	0.0004	2	17.04814	24.36	0.0004
VAR*FERT	4	135.05122	96.49	0.0001	4	135.05122	96.49	0.0001

Aunque los efectos principales, variedad y fertilizante son significativos, no podemos estudiarlos por separado, ya que la mejor dosis de fertilizante depende de la variedad que estemos investigando, según debe inferirse del hecho de que la interacción también es significativa. Lo más recomendable en este caso sería hacer un análisis, estudiando cuál es la tendencia de la respuesta de cada variedad a la fertilización. Para ello, se calculan los componentes lineal y cuadrático para cada variedad, empleando dos grados de libertad por variedad. Los seis grados de liber-

tad que se ocupan corresponden a los de FERT mas los de VAR*FERT; en forma correspondiente, la suma de las sumas de cuadrados de los componentes es igual a la suma de las sumas de cuadrados de FERT y VAR*FERT. (ver "Curvas de respuesta" en la Sección 6.4)

Si la interacción no hubiera sido significativa, se hubiera podido hacer comparaciones de las medias de las variedades y ajustar curvas de respuesta para los niveles de fertilización del promedio de las variedades (ver "Curvas de respuesta" en la Sección 6.4).

Para hacer la comparación entre las medias de las variedades se agrega la siguiente línea al programa SAS:

Means var/Duncan;

6.3.3.5. Diseño de parcelas divididas

La necesidad de utilizar el diseño de Parcelas divididas surge cuando se aplican dos o más tipos de tratamientos en arreglos factoriales, si los niveles de un factor puede aplicarse a parcelas relativamente pequeñas, mientras que los del otro sea más conveniente aplicarlos a parcelas más grandes.

Un ejemplo del uso del diseño de parcelas divididas es cuando se prueban diferentes niveles de irrigación en las parcelas grandes y factores tal como variedades o fertilizantes son aplicados a las parcelas pequeñas. Supóngase que tenemos un experimento con dos niveles de irrigación (alta y moderada) y cuatro variedades de caña en cuatro bloques. Los datos de los rendimientos de la caña son:

		Variedad			
	Irrigación	1	2	3	4
Bloque I	alta	123.2	132.3	123.2	128.8
	moderada	118.2	123.2	115.2	116.3
Bloque II	alta	128.2	138.3	128.2	125.8
	moderada	119.2	120.2	117.2	121.3
Bloque III	alta	118.2	122.3	121.2	124.8
	moderada	111.2	117.2	113.2	113.3
Bloque IV	alta	128.2	123.3	128.2	132.8
	moderada	113.2	122.2	114.2	116.3

El modelo para este experimento es:

$$Y_{ijk} = \mu + \beta_i + I_j + \varepsilon_{ij} + V_i + IV_{ij} + \varepsilon_{ijk}$$

donde:

μ	es la media general
β_i	es el efecto del bloque i
I_j	es el efecto del factor de la parcela principal (Irrigación)
ε_{ij}	es el error experimental asociado a la parcela grande (Irrigación)
V_i	es el efecto del factor de la subparcela (Variedad)
IV_{ij}	es la interacción entre método de irrigación j y variedad k
ε_{ijk}	es el error experimental asociado a las subparcelas

La tabla del análisis de varianza para este experimento sería:

Fuente de variación	**gl**
Bloque	3
Irrigación	1
Error A (bloque*Irr)	3
Variedad	3
Variedad*Irrigación	3
Error B (bloque*irr + var*irr*bloque)	18
Total	31

El Error A es usado para probar las diferencias entre bloques, irrigaciones y el error B es usado para probar las diferencias entre variedades y la interacción entre variedades e irrigaciones.

El programa en SAS sería el siguiente:

```
DATA dividida;
INPUT bloque irr $ var rend;
CARDS;
1    A    1    123.2
1    A    2    132.3
1    A    3    123.2
1    A    4    128.8
1    M    1    118.2
1    M    2    123.2
1    M    3    115.2
1    M    4    132.3
.    .    .  . (resto de los datos)
.    .    .  .
PROC GLM;
CLASSES bloque irr var;
MODEL rend = bloque irr bloque*irr var var*irr;
TEST H=bloque irr E=bloque*irr;
RUN;
```

La instrucción TEST (última línea) indica que bloque e **irr** (**T=bloque irr**) deben probarse usando bloque*irr como error (**E=bloque*irr**). Si no se especifica esto, el programa ejecuta la prueba de F para Bloques y para Irrigación usando el error B (ERROR), lo cual es incorrecto.

La salida de este programa sería la siguiente:

```
                    GENERAL LINEAR MODELS PROCEDURE

DEPENDENT VARIABLE:  REND

SOURCE       DF      SUM OF SQUARES      MEAN SQUARE     F VALUE     PR>F      R-SQUARE      C.V.

MODEL        13     1127.30625000      86.7158656          8.00     0.0001    0.852490      2.70
ERROR        18      195.06250000      10.8368056                  ROOT MSE              REND MEAN

TOTAL        31     1322.36875000                                  3.29193037         121.8312500


SOURCE       DF    TYPE I SS     F VALUE     PR>F     DF     TYPE III     F VALUE     FR>F

BLOQUE        3   214.59375         6.60     0.0033    1    214.59375       6.60     0.0033
IRR           1   754.66125        69.64     0.0001    2    754.66125      69.54     0.0001
BLOQUE*IRR    3    18.09375         0.56     0.6504    2     18.09375       0.56     0.6504
VAR           3   129.92375         4.00     0.0241    4    129.92375       4.00     0.0241
VAR*IRR       3    10.03370         0.31     0.8188    4     10.03370       0.31     0.8188

TEST OF HYPOTHESES USING TYPE III MS FOR BLOQUE*IRR AS AM ERROR TERM

SOURCE              DF         TYPE III SS         F VALUE      PF > F

BLOQUE               3         214.5937500           11.86      0.0359
IRR                  1         754.6612500          125.13      0.0015
```

La salida de SAS puede resumirse en la siguiente tabla:

Fuente	DF	SS	MS	F	P
Bloque	3	214.59	71.53	11.86	0.0359
Irri	1	714.66	714.66	125.13	0.0015
Bloque*Irri	3	18.09	6.03		
Variedad	3	129.92	43.31	4.00	0.0241
Irri*Var	3	10.03	3.34	0.31	0.8188
Error B	18	195.06	10.83		
Total	31	1322.37			

Como podemos ver en esta tabla de ANDEVA, el error que se usa para la prueba de F para probar Bloques e Irrigación es el error A (es decir la interacción Bloque*Irrigación). Note que SAS produce dos tablas de ANDEVA, una donde prueba todas las fuentes de variación contra el Cuadrado Medio del Error y otra donde prueba la parcela grande contra el error A. Por lo tanto, el usuario debe modificar la tabla del ANDEVA como se muestra en la última tabla.

En este ejemplo hubo diferencias significativas entre irrigaciones y entre variedades, pero no hubo significancias en la interacción de irrigación * variedad. Por lo tanto, podemos hacer una prueba de Duncan para las medias de variedades y para las medias de irrigaciones. Esto lo hacemos agregando las siguientes líneas al programa SAS:

```
MEANS irr / DUNCAN E=bloque*irr;
MEANS var / DUNCAN;
```

6.4. Análisis de varianza con tratamientos cuantitativos

Cuando los tratamientos usados en un experimento son cuantitativos, por ejemplo diferentes niveles de un fertilizante aplicados a varias parcelas con algún cultivo, no tiene mucho sentido hacer comparaciones múltiples para comparar las medias de los tratamientos.

En este caso puede ser más útil averiguar cual es la tendencia de la respuesta y no simplemente si hay diferencias entre los diferentes niveles o cual de los niveles empleados es el que produce el mayor rendimiento. Para hacer esto, ajustamos curvas de respuesta para ver si la relación entre el nivel de fertilización y el rendimiento es lineal, cuadrática, cúbica, etc.

Tenemos los siguientes datos de rendimientos en Kg obtenidos al aplicar cuatro niveles de nitrógeno (gramos por parcela) a varias parcelas de maíz, en cuatro bloques

Bloque	Nivel de Nitrógeno			
	0	50	100	150
1	25.4	34.3	36.4	28.3
2	34.2	42.8	43.2	36.2
3	21.2	32.1	34.2	23.2
4	39.9	45.8	43.3	41.3

Debido a que tenemos cuatro niveles de nitrógeno, sólo podemos ajustar polinomios hasta tercer grado (efecto cúbico). Para hacer esto en SAS sería como sigue:

```
DATA a;
INPUT bloque nit rend;
CARDS;
1       0       25.0
2       0       34.2
3       0       21.2
4        0      39.9
1       50      34.3
1       50      42.8
1       50      32.1
1       50      45.8
.       .       . (resto de los datos)
.       .       .
PROC GLM;
CLASS bloque nit;
MODEL rend = bloque nit;
CONTRAST 'Lineal' nit  -3  -1   1   3;
CONTRAST 'Cuadrat' nit  1  -1  -1   1;
CONTRAST 'Cubico' nit  -1   3  -3   1;
RUN;
```

En el caso de este experimento los niveles de nitrógeno están igualmente espaciados. Para este caso se puede recurrir a "tablas de polinomios ortogonales" donde se presentan los coeficientes que deben utilizarse para multiplicar por cada nivel del factor en estudio. Estas tablas nos dan los coeficientes de acuerdo al número de tratamientos que tengamos; en este caso buscamos los coeficientes para cuatro tratamientos. La salida del programa es la siguiente:

GENERAL LINEAR MODELS PROCEDURE

DEPENDENT VARIABLE: REND

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR>F	R-SQUARE	C.V.
MODEL	6	824.545	137.42417	37.62	0.0001	0.6166	5.44
ERROR	9	32.8725	3.6525		ROOT MSE		REND MEAN
TOTAL	15	857.4175			1.9112		35.1125

SOURCE	DF	TYPE I SS	F VALUE	PR>F	DF	TYPE III	F VALUE	FR>F
BLOQUE	3	572.0225	52.20	0.0001	3	572.0225	52.20	0.0001
NIT	3	252.5225	23.05	0.0001	3	252.5225	23.05	0.0002

CONTRAST	DF	SS	F VALUE	PR>F
LINEAL	1	9.1125	2.49	0.1487
CUADRAT	1	243.36	66.63	0.0001
CUBICO	1	0.0500	0.01	0.9094

Del resultado de las pruebas de los contrastes observamos que el polinomio cuadrático es el que explica la relación entre rendimiento y nitrogéno. Los polinomios lineal y cúbico no son significativos. En el caso de que todos los polinomios fueran significativos se debe escoger el polinomio de mayor grado.

El siguiente paso a seguir sería ajustar una regresión cuadrática a las medias de cada nivel de nitrógeno. Esto se haría agregando las siguientes líneas al programa anterior:

```
PROC SORT; BY nit;
PROC MEANS MEAN NOPRINT; BY nit; VAR rend;
OUTPUT OUT=b MEAN = rend;
PROC GLM;
MODEL rend = nit nit*nit;
OUTPUT PREDICTED = py;
PROC PLOT;
PLOT rend*nit= '*' py*nit='+' / OVERLAY;
RUN;
```

Las dos últimas líneas ordenan hacer un gráfico de los datos observados superponiendo la línea de regresión del modelo ajustado (cuadrático).

El ajuste de la regresión cuadrática nos da la ecuación que explica el rendimiento. Para obtener el nivel de nitrógeno que produce el máximo rendimiento, simplemente derivamos la ecuación.

Para hacer el análisis de varianza si los niveles no son igualmente espaciados, se excluye la variable **nit** de **classes** y se modifica el modelo como sigue:

MODEL = bloque nit nit*nit nit*nit*nit;

6.5. Comparación de medias usando contrastes ortogonales

Algunas veces, debido a la naturaleza de los tratamientos, necesitamos hacer comparaciones de grupos de tratamientos, en vez de comparaciones entre promedios de tratamientos individuales.

Consideremos un experimento para estudiar el efecto de fertilizantes e irrigación en el crecimiento de zanahorias en el que se usaron dos fertilizantes: sulfato de amonio (SA) y fosfato monocálcico (FM), lo mismo que un control (sin fertilizante). También se les dió a las parcelas irrigación fuerte e irrigación moderada durante el crecimiento.

Seis tratamientos fueron aplicados al azar a 12 parcelas de zanahoria y se midió el promedio del peso de la zanahoria por parcela.

	Irrigación fuerte			Irrigación moderada		
	SA	FM	Control	SA	FM	Control
	89	84	72	56	85	61
	110	89	80	54	81	40
Trat	1	2	3	4	5	6

En este caso sería interesante hacer las siguientes comparaciones:

1) Irrigación fuerte contra irrigación moderada

2) SA contra FM

3) SA vs FM en irrigación fuerte y SA vs FM en irrigación moderada

4) Fertilizado contra no fertilizado (control vs SA y MP)

5) Fertilizado vs no fertilizando en irrigación fuerte y fertilizado vs no fertilizado en irrigación moderada.

Aquí no podemos utilizar pruebas de comparación múltiple (ej. Duncan, Tukey). Debemos usar la instrucción contrast en el procedimiento GLM.

El programa en SAS para analizar este experimento sería:

```
DATA a;
INPUT trat rend;
CARDS;
1       89
1       110
2       84
2       89
3       72
3       80
.       .       (resto de los datos)
.       .
6       61
6       40
;
PROC GLM;
```

```
CLASS trat;
MODEL rend = trat;
CONTRAST 'IF vs IM' trat 1  1  1  -1  -1  -1;
CONTRAST 'SA vs FM' trat 1 -1  0   1  -1   0;
CONTRAST 'SA vs FM dentro irrig' trat 1  -1  0  -1  1  0;
CONTRAST 'Fert vs No fert' trat 1  1  -2  1  1  -2;
CONTRAST 'Fert vs No fert en irri' trat 1  1  -2  -1  -1  2;
RUN;
```

Los coeficientes para cada contraste deben sumar cero. Para obtener los coeficientes debe tener en cuenta el orden en que están los tratamientos y el peso que hay que darle a cada media. Por ejemplo, para el primer contraste, los tratamientos con irrigación fuerte son los tres primeros y los tratamientos con irrigación moderada son los tres últimos (4,5,6) de acuerdo al orden en que fueron puestos en el programa. Los coeficientes serían: 1,1,1,-1,-1,-1. Esto le dice a SAS que compare la media de las medias de los tratamientos con coeficientes positivos, contra la medias de las medias de los tratamientos con coeficientes negativos.

Las comparaciones hechas en este ejemplo tienen una propiedad importante. Al considerar cualquier par de contrastes, la suma de los productos de los coeficientes que ocupan igual posición es cero. Esto se cumple para cualquier par de contrastes. Las comparaciones con esta propiedad son llamadas comparaciones ortogonales.

La salida de SAS es la siguiente:

```
                    GENERAL LINEAR MODELS PROCEDURE

DEPENDENT VARIABLE:  REND

SOURCE       DF      SUM OF SQUARES      MEAN SQUARE      F VAl       PR>F      R-SQUARE       C.V.

MODEL         5      3595.41666667      719.08333333        8.71      0.0101     0.878878      12.1033

ERROR         6       495.50000000       82.58333333                ROOT MSE              REND MEAN

TOTAL        11      4090.91666667                                   9.08753725            75.08333333


SOURCE       DF    TYPE I SS      F VALUE      PR>F      DF      TYPE III      F VALUE       FR>F

TRAT          5  3595.416667         8.71      0.0101     5   3595.4166667      8.71        0.0101


CONTRAST                DF       SS           F VALUE       PR>F

IF VS IM                 1    1800.750         21.81       0.0034
SA VS FM                 1     112.500          1.36       0.2874
SA VS FM DENTRO IRR      1     840.500         10.18       0.0188
FERT VS MO FERT          1     840.167         10.17       0.0188
FERT VS NO FERT EN IRR   1       1.500          0.02       0.8972
```

Como se ve en la salida del SAS, encontramos diferencias significativas entre irrigación fuerte y moderada, también entre fertilizado y no fertilizado, por último entre SA vs FM dentro de irrigaciones. Los otros dos contrastes no son significativos. En este ejemplo, los cinco grados de

libertad de los tratamientos fueron divididos en cinco contrastes independientes, cada uno con un grado de libertad. La suma de las sumas de cuadrados de los contrastes debe ser igual a la suma de cuadrados de los tratamientos.

6.6. Análisis de Covarianza

Se hizo una prueba del efecto de tres tratamientos al suelo sobre crecimiento en altura de arbolitos de dos años. Los tratamientos se asignaron al azar a las tres parcelas dentro de cada uno de los 10 bloques. Cada parcela incluía 50 arbolitos. La media del crecimiento de cinco años fue el criterio para evaluar los tratamientos. Las alturas iniciales y los crecimientos de cinco años, todos ellos medidos en pies, fueron:

	Tratamiento					
	A		B		C	
Bloque	Altura	Crecim	Altura	Crecim	Altura	Crecim
1	3.6	8.9	3.1	10.7	4.7	12.4
2	4.7	10.1	4.9	14.2	2.6	9.0
3	2.6	6.3	0.8	5.9	1.5	7.4
4	5.3	14.0	4.6	12.6	4.3	10.1
5	3.1	9.6	3.9	12.5	3.3	6.8
6	1.8	6.4	1.7	9.6	3.6	10.0
7	5.8	12.3	5.5	12.8	5.8	11.9
8	3.8	10.8	2.6	8.0	2.0	7.5
9	2.4	8.0	1.1	7.5	1.6	5.2
10	5.3	12.6	4.4	8.4	4.8	10.7

Al hacer un análisis de varianza para el crecimiento, encontramos que no hay pruebas de una diferencia real en el crecimiento debido a tratamientos.

Fuente	gl	SC	CM	
Bloques	9	110.39	12.26	
Tratamientos	2	6.65	3.33	No sig
Error	18	67.58	3.75	
Total	29	184.61		

Sin embargo, existen razones para creer que, en el caso de arbolitos jóvenes, el crecimiento está afectado por la altura inicial. La posibilidad de que los efectos de los tratamientos estén encubiertos por diferencias en las alturas iniciales plantea la duda de cómo se hubieran comportado los tratamientos si su altura inicial hubiera sido igual.

El análisis de covarianza usando la variable altura inicial como covariable, ajustaría los datos de tal manera que se probarían los tratamientos como si todas las alturas iniciales fueran las mismas. Para hacer ésto en SAS, necesitamos el siguiente programa:

```
DATA a;
INPUT bloque trat $ altura crecim;
CARDS;
 1    A    3.6   8.9
 2    A    4.7   10.1
 3    A    2.6   6.3      (Continúa en la siguiente página)
 .    .    .     .        (resto de los datos)
 .    .    .     .
 8    C    2.0   7.5
 9    C    1.6   5.2
10    C    4.8   10.7 ;
```

```
PROC GLM;
CLASS bloque trat;
MODEL crecim = bloque trat altura;
RUN;
```

En la instrucción model, se incluyó la variable altura. Debido a que altura no está definida como una clase (en la línea class), SAS la considera como una covariable. Si se tienen más covariables en el experimento, estas se incluyen a la derecha del signo "=", después de las fuentes de variación (en este ejemplo bloque y tratamiento). La salida sería:

GENERAL LINEAR MODELS PROCEDURE

DEPENDENT VARIABLE: REND

SOURCE	DF	SUM OF SQUARES	MEAN SQUARE	F VALUE	PR>F	R-SQUARE	C.V.
MODEL	12	149.7824	12.4819	6.09	0.0004	0.851182	14.692
ERROR	17	34.8296	2.0488	ROOT MSE		REND MEAN	
TOTAL	29	184.612		1.4314		9.74	

SOURCE	DF	TYPE I SS	F VALUE	PR>F	DF	TYPE III	F VALUE	FR>F
BLOQUE	9	110.379	5.99	0.0008	9	10.70	0.58	0.7954
TRAT	2	6.656	1.62	0.2262	2	11.68	2.85	0.0855
ALT	1	32.748	15.98	0.0009	1	32.798	15.98	0.0009

De la salida podemos ver que la variable altura inicial explica mucho de la variación total y es altamente significativa. Para ver si los tratamientos son significativamente diferentes examinamos las sumas de cuadrados tipo III (a la derecha) y vemos que el valor de F ajustado para tratamientos es 2.85 (altamente significativo). Este valor es mucho más alto que el F sin ajustar (1.62, no significativo). Podemos entonces ver que el análisis de covarianza nos sirvió para encontrar diferencias reales de los tratamientos.

Si deseamos calcular las medias ajustadas de cada tratamiento y al mismo tiempo ver cuáles medias son diferentes, agregamos la siguiente línea al programa:

```
LSMEANS TRAT / PDIFF;
```

Esta línea produce lo siguiente:

LEAST SQUARES MEANS

TRAT	CRECIM LSMEAN	PROB > |T| HO: I/J	1	LSMEAN(I)=LSMEAN(J) 2	3
A	9.3143	1	.	0.0069	0.9270
B	10.6534	2	0.0069	.	0.0440
C	9.2522	3	0.9270	0.0440	.

Las medias ajustadas de cada tratamiento aparecen debajo de la columna Crecim/Lsmean. A la derecha aparece una matriz de probabilidades apoyando la hipótesis nula, Ho: media ajustada i = media ajustada j. En este ejemplo la media del tratamiento A es diferente a la del tratamiento B (la probabilidad obtenida, 0.0069, es menor que 0.05). También la media B es diferente a la media C (0.0080), mientras que la media A y C no son diferentes (0.8548).

6.7. Tipos de sumas de cuadrados calculados por el PROC GLM

El Proc Glm, produce cuatro diferentes tipos de sumas de cuadrados relacionadas con diferentes interpretaciones teóricas. Estos cuatro tipos de sumas de cuadrados son llamados Tipo I, Tipo II, Tipo III y Tipo IV. Para nuestros propósitos sólo nos interesan las sumas de cuadrados Tipo I y Tipo III.

Las sumas de cuadrados Tipo I corresponden a añadir cada fuente (factor) secuencialmente al modelo en el orden dado. Este tipo de sumas de cuadrados no es útil para una estructura de varias entradas con datos no balanceados, pero sirve para diseños ortogonales (p. ej. bloques al azar, parcelas divididas, etc. sin datos perdidos), diseños anidados y modelos polinomiales.

Las sumas de cuadrados Tipo III son sumas de cuadrados parciales. Su uso principal es en situaciones que requieren una comparación de efectos principales aún bajo la presencia de interacción. Cada efecto es ajustado después de los otros efectos. Deben usarse cuando tenemos datos no balanceados y en análisis de covarianza, pero no en diseños o situaciones donde el orden de los factores no puede ser cambiado (diseño de parcelas divididas, ajustes de modelos polinomiales).

Supongamos que ajustamos el siguiente modelo:

MODEL Y = A B A*B;

Las sumas de cuadrados Tipo I y Tipo III ajustarían los factores de la siguiente manera:

Efecto	Tipo I	Tipo III
A	$R(\alpha \mid \mu)$	$R(\alpha \mid \mu, \beta, \alpha\beta)$
B	$R(\beta \mid \mu, \alpha)$	$R(\beta \mid \mu, \alpha, \alpha\beta)$
A*B	$R(\alpha\beta \mid \mu, \alpha, \beta)$	$R(\alpha\beta \mid \mu, \alpha, \beta)$

Como se puede ver la sumas de cuadrado Tipo I ajustan los efectos secuencialmente, primero el factor A, luego el B y por último la interacción. Las sumas de cuadrados Tipo III ajustan cada factor después de haber ajustado los otros factores.

LISTA DE FUNCIONES MAS UTILIZADAS EN SAS

CATEGORIA	FUNCION	Descripción	Ejemplo
Aritméticas			
	ABS	El valor absoluto	Y = ABS (X);
	MAX	El valor máximo	Y = MAX (7,6,4,5,2);
	MIN	El valor mínimo	Y = MIN (7,6,4,5,2);
	SQRT	La raíz cuadrada	Y = SQRT (X);
Matemáticas			
	EXP	Exponencial	Y = EXP (X);
	LOG	Logaritmo natural (base e)	Y = LOG (X);
	LOG2	Logaritmo en base 2	Y = LOG2 (X);
	LOG10	Logaritmo en base 10	Y = LOG10 (X);
Trigonométricas			
	ARCOS	El arco coseno	Y = ARCOS (X);
	ARSIN	El arco seno	Y = ARSIN (X);
	ATAN	El arco tangente	Y = ATAN (X);
	SIN	El seno	Y = SIN (X);
	COS	El coseno	Y = COS (X);
Estadística descriptiva			
	CV	El coeficiente de variación	Y = CV (1,3,5,1,7,6);
	MEAN	La media aritmética	Y = MEAN (1,3,5,1,7,6);
	RANGE	El rango	Y = RANGE (1,3,5,1,7,6);
	STD	La desviación estándar	Y = STD (1,3,5,1,7,6)
	SUM	La suma	Y = SUM (1,3,5,1,7,6);
	VAR	La varianza	Y = VAR (1,3,5,1,7,6);
De fecha y tiempo			
	DATEJUL	Convierte fecha juliana a un valor	Y = DATEJUL(X);
	DATE	La fecha actual	Y = DATE ( );
	DAY	El día de una fecha	Y = DAY (Fecha);
	HMS	Las horas, minutos y segundos	Y = HMS (Hora, minutos, seg.);
	JULDATE	Convierte un valor a fecha juliana	Y = JULDATE (X);
	MDY	Convierte una fecha a un valor	Y = MDY (mes, día, año);
	YEAR	Toma el año de una variable de fecha y se lo asigna a una variable	Y = YEAR (Fecha);

ANEXO

EJERCICIOS PRACTICOS

EJERCICIO No. 1

Se tienen los siguientes datos de ejemplares bovinos representativos de 12 fincas ganaderas. Los ejemplares corresponden a diferentes propósitos (leche, carne y doble). Los datos son los siguientes:

FINCA	PESO	RAZA	PROPOSITO
Santa Clara	300	Holstein	leche
Mercedes	350	Holstein	leche
Los llanos	500	Brown siwss	doble
El rio	510	Brown swiss	doble
San José	370	Jersey	leche
San Martín	670	Cebu	carne
El paso	470	Jersey	leche
Los toros	510	Pardo suizo	doble
El Jefe	380	Santa gertrudis	doble
El Chaparral	610	Brahman	carne
Tres ases	350	Jersey	leche
San José	580	Cebu	carne

Hacer lo siguiente:

a. Crear un archivo SAS temporal con los datos
b. Imprimir los datos
c. Obtener las medias para la variable peso
d. Realizar un análisis de frecuencias para las variables raza y proposito.
e. Obtener las medias de peso para cada raza.

EJERCICIO No. 2

Se dispone de los datos promedio de temperatura y precipitacion pluvial de la década de 1981-1990 de la Ciudad de San clemente.

MES	TEMPERATURA	PRECIPITACION
A (ENE)	20.2	800
B (FEB)	22.0	750
C (MAR)	23.5	600
D (ABR)	24.0	550
E (MAY)	25.3	650
F (JUN)	24.8	800
G (JUL)	26.0	850
H (AGO)	24.2	1000
I (SEP)	26.0	1125
J (OCT)	25.4	1400
K (NOV)	25.0	1350
L (DIC)	22.0	1000

Utilizando los datos hacer lo siguiente:

a. Crear un archivo SAS temporal
b. Imprimir los datos
c. Hacer un gráfico de la precipitación vs. mes
d. Hacer un gráfico de la temperatura vs. mes

Nota: Para ambos gráficos utilice un tamaño de 10 en el eje vertical y 30 en el horizontal.

EJERCICIO No. 3

Los siguientes datos corresponden al peso, en miligramos, de 25 semillas.

96	101	102	100	98
104	105	102	103	107
103	98	108	105	99
108	97	94	96	110
101	104	99	105	102

Hacer lo siguiente:

a. Crear un archivo SAS temporal
b. Construir el histograma y la curva de frecuencias relativas acumuladas.
c. Calcular la media, varianza y recorrido (rango).
d. Probar si estos datos provienen de una distribución normal.

EJERCICIO No. 4

Los siguientes datos corresponden a la altura de 23 plantas (cm) de 3 variedades de maiz a los treinta días después de la siembra.

var 1			var 2				var 3				
20	21	31	23	25	27	28	30	29	36	33	30
31	23	30	28	30	22	27	39	39	27	35	

Resolver los siguientes aspectos:

a. Crear un archivo SAS temporal
b. Calcular el promedio, varianza, desvío estándar y coeficiente de variación por variedad.
c. Elaborar un gráfico de barras en el cual aparezca la altura media en el eje vertical y la variedad en el horizontal.

EJERCICIO No. 5

En un muestreo realizado en un bosque natural, se estudiaron 126 parcelas de 1/2 hectárea cada una contándose en ellas el número de plantas de Ryania sp. Se obtuvieron los resultados siguientes:

No. de plantas	No. de parcelas
0	100
1	9
2	6
3	8
4	1
5	0
6	2
	126

Resolver los siguientes aspectos:

a. Crear un archivo SAS temporal
b. Construir un diagrama de barras de la variable número de plantas/parcela.
c. Calcular la media, mediana, varianza y desviación estándar del número de plantas/parcela.
d. Calcular la media y varianza del número de plantas por hectárea.

EJERCICIO No.6

Con el propósito de evaluar la ganancia de peso en cabritos se probaron dos dietas diferentes. Los resultados se presentan a continuación:

dieta 1		dieta 2	
1.2	2.0	0.8	1.2
0.8	1.7	1.1	0.9
1.3	1.4	0.7	0.7
0.7	1.8	0.6	0.4
		0.8	0.7

Resolver los siguientes aspectos:

a. Crear un archivo SAS temporal
b. Realizar una prueba para determinar si las dietas producen efectos iguales en la ganancia de peso en los cabritos.
c. Imprimir los resultados.

EJERCICIO No. 7

Se realizó un experimento con el fin de evaluar la respuesta de 4 niveles de nitrógeno en el rendimiento del cultivo de sorgo. Cada nivel tuvo 5 repeticiones. El rendimiento obtenido (kg/parcela) aparece en el cuadro siguiente:

REPETICION	NIVEL DE NIT.	RENDIMIENTO (kg/parc.)
1	0	25.4
2	0	34.2
3	0	21.2
4	0	36.9
1	50	34.3
2	50	36.8
3	50	32.1
4	50	40.6
1	150	40.2
2	150	47.5
3	150	39.3
4	150	49.7
1	100	36.4
2	100	43.2
3	100	34.2
4	100	43.3

Resolver los siguientes aspectos:

a. Crear un archivo ASCII en la unidad "A" y nombrarlo EJEMPLO.DAT
b. Hacer un programa SAS que lea el archivo creado
c. Ordenar los datos por la variable nitrógeno y dar un listado de ellos.
d. Obtener las promedios de rendimiento para cada nivel de nitrógeno.
e. Realizar un análisis de regresión lineal simple de las variables rendimiento y niveles de nitrógeno, utilizando los promedios.
f. Elaborar un análisis gráfico de los valores observados y predichos.

EJERCICIO No. 8

El rendimiento promedio de vainas de frijol (kg/parcela), se analizó en 4 sitios diferentes, cada uno con 5 parcelas. Los datos de campo son los siguientes:

SITIO	PARCELA	REND.	SITIO	PARCELA	REND.
A	1	7.8	C	1	15.3
A	2	6.3	C	2	16.9
A	3	5.4	C	3	11.7
A	4	4.2	C	4	9.7
A	5	9.8	C	5	12.7
B	1	3.2	D	1	9.6
B	2	4.2	D	2	8.9
B	3	3.9	D	3	7.6
B	4	2.8	D	4	8.3
B	5	4.7	D	5	10.5

Resolver los siguientes aspectos:

a. Crear en el directorio CURSO un archivo ASCII con los datos de los sitios A y B denominado DATOS1.DAT y otro con los datos de los sitios C y D denominado DATOS2.DAT.
b. Crear archivos SAS temporales
c. Concatenar archivos SAS
d. Imprimir los datos
e. Crear un subconjunto con los sitios "A" y "C" e imprimir los datos. Realizar la prueba "T" para comparar el rendimiento en ambos sitios.
f. Crear un subconjunto con los sitios "B" y "D" e imprimir los datos. Realizar la prueba "T" para comparar el rendimiento en ambos sitios.
g. Obtener estadísticas por sitio para la variable rendimiento, considerando el archivo concatenado.

EJERCICIO No. 9

En tres sitios forestales se evaluó el volumen de especies latifoliadas y coníferas. Con los datos de campo obtenidos se determinó un volumen aparente (m^3), el cual debe ser multiplicado por un factor de forma de acuerdo a la especie y el sitio para obtener el volumen real (m^3). Las tablas de datos de campo y de factores de forma son:

SITIO	ESPECIE	VOL.APAR.
A	LATIF	2.22
A	CONIF	1.80
B	LATIF	1.89
B	CONIF	1.68
C	LATIF	2.10
C	CONIF	2.30

REFERENCIAS

SITIOS	ESPECIE	FACTOR
A	LATIF	0.7
	CONIF	0.7
B	LATIF	0.6
	CONIF	0.5
C	LATIF	0.68
	CONIF	0.73

Hacer lo siguiente:

a. Crear un archivo ASCII con los datos de campo
b. Crear un programa SAS temporal que lea el archivo ASCII creado.
c. Generar una nueva variable que contenga el volumen real a partir del volumen aparente y el factor de forma.
d. Imprimir los datos.
e. Obtener promedios de las variables volumen real y volumen aparente.

EJERCICIO No. 10

Se cuenta con información acerca de un grupo de técnicos que participaron en un taller de capacitación sobre sistemas agroforestales durante 15 días. Al finalizar el curso los datos de asistencia y calificaciones fueron los siguientes:

NOMBRE	COMUNIDAD	SEXO	NOTA	ASISTENCIA
López	El Naranjo	M	83	10
Pérez	San José	M	93	9
Rosales	El Remate	F	78	13
Pineda	Macanche	M	88	10
Avalos	Macanche	M	88	11
Contreras	El Remate	F	76	7
Ramos	El Naranjo	F	76	10
Mejia	Flores	F	81	9
Hernández	San Andres	M	90	15
Esquivel	Poptun	M	92	14
Rodríguez	Poptun	M	94	13
Montero	Flores	F	82	8
Guerra	San José	F	92	15
Vargas	Flores	F	92	15
Soto	El Remate	M	94	15

Se bonificara la nota según la asistencia al curso, de la siguiente manera:

- de 8-10 días se bonificara con 1.5%
- de 11-12 días se bonificara con un 3%
- de 13-15 días se bonificara con un 5%

Hacer lo siguiente:

a. Generar una variable para la nota bonificada
b. Crear una variable cuyos valores sean 'gana' o 'pierde' que indica la situación del estudiante en el curso. Para el efecto, un estudiante gana el curso si la nota bonificada es mayor o igual a 80.
c. imprimir los datos
d. Calcular las estadísticas descriptivas del grupo para la nota y nota bonificada
e. Calcular las estadísticas descriptivas por comunidad para la variable nota y nota bonificada
f. Calcular las estadísticas descriptivas por sexo y resultado ('gana' 'pierde'), utilizando la nota bonificada.

EJERCICIO No. 11

Se cuenta con información acerca de los alumnos que participaron en el curso de capacitación sobre educación ambiental. Para cada uno se incluyen los años de educación universitaria.

NOMBRE	SEXO	NOTA	AÑOS ED. UN.
Gamboa	m	82	4
Pérez	f	93	5
Mejia	m	78	3
Chinchilla	f	81	4
Soto	m	90	5
Araujo	m	77	3
López	m	76	4
Ching	f	54	1
Campos	f	92	5
Aguilar	m	62	2
Rodríguez	m	41	1
Kim	f	99	5

Analizar los datos resolviendo los siguientes puntos:

a. Crear un archivo ASCII denominado CURSO.DAT
b. Crear un programa SAS temporal que lea el archivo ASCII creado
c. Generar una variable que valga 1 o 0 de acuerdo a que el estudiante haya aprobado (nota mayor o igual a 70) o reprobado el curso. Imprima una tabla que contenga los nombres en la primera columna y la palabra 'aprobado' o 'reprobado' en la segunda.
d. Calcular las estadísticas descriptivas para cada sexo
e. Calcular el porcentaje de aprobados para cada sexo
f. Ajuste un modelo de regresión lineal que explique la variable nota en función de los años de educación. Haga un gráfico que contenga los puntos observados y las predicciones.
g. Realizar un análisis de residuos para verificar si se cumplen los supuestos del modelo.

EJERCICIO No. 12

Se evaluaron tres dietas para alimentación de cerdos. Las ganancias de peso (kg) para los animales observados fueron los siguientes:

DIETA 1	DIETA 2	DIETA 3
1.2	0.8	1.1
0.8	0.4	1.2
1.4	0.3	1.5
1.7	0.6	0.9
2.3	0.9	1.2

Resolver los siguientes aspectos:

a. Crear un archivo SAS temporal
b. Imprimir los datos
c. Realizar el análisis de varianza para un DCA
d. Realizar una prueba de Tukey para la comparación de las dietas.

EJERCICIO No. 13

Para el grupo de estudiantes del curso de estadística de la promoción 1990, se tienen las calificaciones obtenidas en los quices, trabajos domiciliarios y exámenes parcial y final. La información se tiene almacenada en dos archivos ASCII. Tal como se muestra en el cuadro, el primero denominado HORIZ1.DAT tiene la información sobre quices y trabajos domiciliarios y el segundo denominado HORIZ2.DAT tiene la información sobre exámenes parcial y final.

	HORIZ1.DAT					HORIZ2.DAT	
NOMBRE	QUIZ1	QUIZ2	QUIZ3	DOMIC1	DOMIC2	EXAMEN1	EXAMEN2
ana	77	60	67	80	85	67	82
carlos	80	90	87	90	93	82	90
ines	30	50	50	70	80	60	55
josé	100	95	90	100	100	100	98
juan	75	80	77	85	90	75	80
lorena	60	70	80	83	90	50	70
luis	93	100	95	100	95	90	100
maria	92	89	80	90	90	75	80
mario	90	85	80	87	89	70	90
marta	73	80	80	90	85	78	82
miguel	91	85	90	85	86	67	70
ramon	60	65	70	90	95	83	87

Para la elaboracion de un informe:

a. Generar archivos SAS temporales para cada archivo ASCII
b. Ordenar cada archivo SAS por nombre
c. Crear un archivo de trabajo SAS temporal con la concatenación horizontal de los archivos SAS creados anteriormente. Para la concatenación usar la variable nombre

d. Generar la variable nota final "notafin" de acuerdo a la siguiente ponderación:
   - promedio de quices = 20%
   - promedio de domiciliarios = 30%
   - examen 1 = 30%
   - examen 2 = 20%

e. Generar una variable denominada "result" de acuerdo a la variable nota final utilizando el criterio: menor a 70 'perdio' y mayor o igual a 70 'gano'

f. Imprimir los datos

EJERCICIO No. 14

Un experimento consistió en la evaluación del rendimiento de maiz en respuesta al fosfato. En el cuadro siguiente se presentan los rendimientos (lb/grano/unidad experimental) del control (X1); las diferencias entre el tratamiento con fosfato y el control (Y) y además se presentan datos acerca de la saturación de bases (X2) y el pH (X3) en las diferentes unidades experimentales.

(Y)	(X1)	(X2)	(X3)
88	844	67	5.75
80	1687	57	6.05
42	1573	39	5.45
37	3025	54	5.70
37	653	46	5.55
20	1991	62	5.00
20	2187	69	6.40
18	1262	74	6.10
18	4624	69	6.05
4	5249	76	6.15
2	4258	80	5.55
2	2943	79	6.40
-2	5092	82	6.55
-7	4496	85	6.50

Analizar los datos haciendo lo siguiente:

a. Crear y ejecutar un programa SAS que lea el archivo ASCII generado y que cree un archivo permanente

b. Imprimir los datos

c. Obtener la matriz de correlación para todas las variables

d. Obtener las siguientes regresiones lineales
   - Y en función de X1
   - Y en función de X2
   - Y en función de X3

EJERCICIO No. 15

Se realizo un ensayo para evaluar el comportamiento de 6 cultivares de trébol rojo inoculadas con combinaciones de cultivos de cepas de Rhizobium troffoli y Rhizobium meliloti. Para el efecto se empleó un diseño completamente al azar balanceado con 30 unidades experimentales. La variable de respuesta fue el contenido de nitrógeno en las plantas. Los resultados se presentan en el cuadro siguiente:

CULTIVAR	NITROGENO	CULTIVAR	NITROGENO
3d0k1	19.4	3d0k7	20.7
3d0k1	32.6	3d0k7	21.0
3d0k1	27.0	3d0k7	20.5
3d0k1	32.1	3d0k7	18.8
3d0k1	33.0	3d0k7	18.6
3d0k5	17.7	3d0k9	14.3
3d0k5	24.8	3d0k9	14.4
3d0k5	27.9	3d0k9	11.8
3d0k5	25.2	3d0k9	11.6
3d0k5	24.3	3d0k9	14.2
3d0k4	17.0	3d0k8	17.3
3d0k4	19.4	3d0k8	19.4
3d0k4	9.1	3d0k8	19.1
3d0k4	11.9	3d0k8	16.9
3d0k4	15.8	3d0k8	20.8

a. Realizar el análisis de varianza para un DCA
b. Realizar una prueba de Tukey y una de Duncan para la comparación de los cultivares.

EJERCICIO No. 16

Un experimento con 6 tratamientos (substrato edáfico definido por la textura y la fertilidad) y tres repeticiones, arrojó los siguientes resultados de rendimiento de soya, en Kg/parcela.

PARCELA	RENDIMIENTO	PARCELA	RENDIMIENTO
1	7.21	4	5.62
2	5.08	5	7.67
3	6.94	6	5.94
4	6.25	1	7.55
5	7.31	2	5.57
6	6.14	3	6.54
1	7.98	4	4.46
2	4.32	5	5.63
3	6.01	6	5.90

La identificación del substrato edafico debe hacerse en función del la textura y la fertilidad de acuerdo a las siguiente tabla:

PARCELA	TEXTURA	FERTILIDAD
1	1	1
2	1	2
3	2	1
4	2	2
5	3	1
6	3	2

Resolver los siguientes aspectos:

a. Crear un archivo ASCII con los datos de rendimiento por parcela
b. Crear un programa SAS que lea el archivo ASCII anterior y genere un archivo SAS permanente
c. Generar dos nuevas variables para caracterizar el substrato edafico (textura = text y fertilidad = fert) a partir de la variable parcela
d. Imprimir los datos
e. Obtener las estadísticas descriptivas para la variable rendimiento de acuerdo a los diferentes tipos de textura
f. Realizar una prueba de medias para la variable rendimiento de acuerdo a los dos tipos de fertilidad.

EJERCICIO No. 17

Se recolectaron muestras de tabaco en diferentes campos de cultivo y se realizo un análisis de la combustión foliar y los contenidos de nitrogeno, cloro y potasio. Los resultados se presentan en el cuadro siguiente:

A	B	C	D
3.05	1.45	5.67	0.34
4.22	1.35	4.86	0.11
3.34	0.26	4.19	0.38
3.77	0.23	4.42	0.68
3.52	1.10	3.17	0.18
3.54	0.76	2.76	0.00
3.74	1.59	3.81	0.08
3.78	0.39	3.23	0.11
2.92	0.39	5.44	1.53
3.10	0.64	6.16	0.77
2.86	0.82	5.48	1.17
2.78	0.64	4.62	1.01
2.22	0.85	4.49	0.89
2.67	0.90	5.59	1.40
3.12	0.92	5.86	1.05
3.03	0.97	6.60	1.15
2.45	0.18	4.51	1.49
4.12	0.62	5.31	0.51
4.61	0.51	5.16	0.18
3.94	0.45	4.45	0.34
4.12	1.79	6.17	0.36
2.93	0.25	3.38	0.89
2.66	0.31	3.51	0.91
3.17	0.20	3.08	0.92
2.79	0.24	3.98	1.35
2.61	0.20	3.64	1.33
3.74	2.27	6.50	0.23
3.13	1.48	4.28	0.26
3.49	0.25	4.71	0.73
2.94	2.22	4.58	0.23

REFERENCIAS

A: NITROGENO
B: CLORO
C: POTASIO
D: COMBUSTION FOLIAR (LOG)

a. Obtener el modelo de regresión lineal múltiple para analizar la relación entre la combustión foliar y los contenidos de nitrogeno, cloro y potasio.
b. Utilice un procedimiento que discrimine variables

EJERCICIO No. 18

En el cuadro adjunto se presentan datos de DAP y volumen total (M^3/ha) de 15 árboles medidos en una parcela de bosque natural.

No. ARBOL	DAP	VOLUMEN/HA (M^3)
1	10	22
2	12	25
3	30	51
4	33	50
5	17	33
6	20	31
7	25	36
8	27	33
9	13	40
10	15	39
11	16	32
12	48	45
13	27	42
14	59	70
15	52	65

b. Obtener la distribución de los árboles por clases diametricas y por clases de volumen. Utilizar un ancho de clase de 10 para el DAP y de 8 para el volumen
c. Imprimir los datos

EJERCICIO No. 19

Dos muestras de tamaño fueron tomadas al azar de una población de cerdos. Luego se formaron pares de pesos similares y se les proporcionó la dieta A a un individuo y la dieta B al otro. Las ganancias de peso vivo después de la prueba de engorde se presentan a continuación

PAREJA	DIETA A	DIETA B
1	24.3	24.2
2	27.2	28.1
3	28.6	25.9
4	26.7	27.0
5	25.3	25.2
6	28.5	26.9
7	23.7	23.6
8	27.0	28.3
9	26.1	26.1

a. Realizar una prueba para evaluar si las dietas producen efectos diferentes sobre el incremento de peso de los cerdos
b. imprimir los datos

EJERCICIO No. 20

Con el propósito de evaluar el comportamiento de cinco variedades de frijol (A, B, C, D y E), se instaló un experimento bajo condiciones de invernadero utilizando un diseño completamente al azar en el cual los tratamientos se repitieron 6 veces. La variable respuesta fue el peso de materia seca (g) 30 días después de la siembra. Los resultados se presentan en el cuadro siguiente:

VARIEDADES Y PESOS SECOS

A 2.9	B 3.3	C 3.1	D 4.5	E 6.6
A 3.5	B 3.8	C 3.8	D 4.4	E 6.8
A 4.1	B 3.7	C 4.2	D 4.8	E 7.4
A 3.0	B 3.6	C 3.1	D 4.7	E 7.2
A 3.0	B 3.1	C 3.5	D 4.5	E 7.0
A 3.6	B 3.3	C 3.2	D 5.0	E 7.6

a. Realizar un análisis de varianza
b. Realizar las pruebas de Duncan y Tukey para las medias de peso seco de las variedades a niveles de significancia de 1% y 5%
c. Imprimir los datos

RESPUESTAS

Respuesta ejercicio No 1:

```
data a;
input finca $ 1-14  peso 15-18 raza $ 19-34 propos $;
cards;
Santa Clara     300  Holstein            leche
Mercedes        350  Holstein            leche
Los llanos      500  Brown siwss         doble
El rio          510  Brown swiss         doble
San José        370  Jersey              leche
San Martín      670  Cebu                carne
El paso         470  Jersey              leche
Los toros       510  Pardo suizo         doble
El Jefe         380  Santa gertrudis     doble
El Chaparral    610  Brahman             carne
Tres ases       350  Jersey              leche
San José        580  Cebu                carne
```

Proc print;
title 'informacion de ejemplares bovinos en 12 fincas';
Proc means;
var peso;
proc freq;
tables raza propos;
proc sort; by raza;
proc means mean; var peso; by raza;
run;

Respuesta ejercicio No 2:

```
data a;
input mes $ temp precipit;
cards;
A   20.2    800
B   22.0    750
C   23.5    600
D   24.0    550
E   25.3    650
F   24.8    800
G   26.0    850
H   24.2   1000
I   26.0   1125
J   25.4   1400
K   25.0   1350
L   22.0   1000
```

proc print;
title 1 'Datos climaticos de San Clemente';
proc plot;
plot precipit*mes = '*'/Vpos=10 Hpos=30;
title 'Distribución anual de la lluvia en San Clemente';
proc plot;
plot temp*mes = '*'/ Vpos=10 Hpos=30;
title 'Temperatura mensual en San Clemente';
run;

Respuesta ejercicio No.3 :

```
data a;
input pesosem @@;
cards;
 96          101          102          100           98
104          105          102          103          107
103           98          108          105           99
108           97           94           96          110
101          104           99          105          102
proc chart;
vbar pesosem/discrete type=freq;
vbar pesosem/discrete type=cfreq;
proc means mean var range;
proc univariate;
run;
```

Respuesta ejercicio No. 4:

```
data a;
input varie alt @@;
cards;
1  20  2  23  3  30
1  21  2  25  3  29
1  21  2  27  3  36
1  31  2  28  3  33
1  23  2  28  3  30
1  30  2  30  3  39
       2  22  3  39
       2  37  3  27
                 3  35
proc sort; by varie;
proc means mean var std cv; var alt; by varie;
proc chart;
vbar varie/discrete type=mean sumvar=alt;
run;
```

Respuesta ejercicio No. 5:

```
data a;
input Nplan Nparc;
      Nplanha=Nplan*2;
cards;
0   100
1     9
2     6
3     8
4     1
5     0
6     2
proc chart;
vbar Nplan/discrete freq=Nparc;
proc univariate  vardef=wdf; var Nplan; weight Nparc;
```

```
proc univariate  vardef=wdf; var Nplanha; weight Nparc;
run;
```

Respuesta ejercicio No 6:

```
data dieta;
input dieta peso @@;
cards;
1  1.2  2  0.8
1  0.8  2  1.1
1  1.3  2  0.7
1  0.7  2  0.6
1  2.0  2  0.8
1  1.7  2  1.2
1  1.4  2  0.9
1  1.8  2  0.7
       2  0.4
       2  0.7
proc print;
proc ttest;
class dieta;
var peso;
run;
```

Respuesta ejercicio No 7:

```
data a;
infile 'a:\ejemplo.dat';
input repet nitro rend;
proc sort;
by nitro;
proc print;
proc means mean; var rend; by nitro;
output out = medias mean = medrend;
proc print;
proc glm;
model medrend = nitro;
output predicted = Prend;
proc plot;
plot medrend * nitro = 'o' Prend * nitro = 'p'/overlay
Vpos = 15 Hpos = 30;
run;
```

Respuesta ejercicio No. 8:

```
data a;
infile 'c:\curso\datos1.dat';
input sitio $ parc rend;
data b;
infile 'c:\curso\datos2.dat';
input sitio $ parc rend;
data c; set a b;
proc print;
```

```
data D; set c;
if sitio= 'A' or sitio='C'
   then output;
proc print;
proc ttest; class sitio; var rend;
data E; set c;
if sitio= 'A' or sitio='C'
   then delete;
proc print;
proc ttest; class sitio; var rend;
proc means data=c; by sitio; var rend;
run;
```

Respuesta ejercicio No. 9:

```
data a;
infile 'a:\ejemplo.dat';
input sitio $ especie $ volapar;
if sitio='A' then volreal = volapar * 0.7;
if sitio='B' and especie = 'LATIF' then volreal = volapar * 0.6;
if sitio='B' and especie = 'CONIF' then volreal = volapar * 0.5;
if sitio='C' and especie = 'LATIF' then volreal = volapar * 0.68;
if sitio='C' and especie = 'CONIF' then volreal = volapar * 0.73;
proc print;
proc means mean;
var volreal volapar;
run;
```

Respuesta ejercicio No. 10:

```
data a;
input nombre $ 1-9 comuni $ 11-20 sexo $ 22 nota asist;
if asist <8 then notabon = nota;
if asist >=8 and asist <=10 then notabon = nota * 1.015;
if asist >10 and asist <=12 then notabon = nota * 1.03;
if asist >12 then notabon = nota * 1.05;
if notabon >=80 then result = 'gano';
else result = 'perdio';
cards;
López     El Naranjo M     83   10
Pérez     San José   M     93    9
Rosales   El Remate  F     78   13
Pineda    Macanche   M     88   10
Avalos    Macanche   M     88   11
Contreras El Remate  F     76    7
Ramos     El Naranjo F     76   10
Mejia     Flores     F     81    9
Hernández San Andres M     90   15
Esquivel  Poptun     M     92   14
Rodríguez Poptun     M     94   13
Montero   Flores     F     82    8
Guerra    San José   F     92   15
Vargas    Flores     F     92   15
Soto      El Remate  M     94   15
```

```
Proc print;
Pros means; var nota notabon;
proc sort; by comuni;
proc means; by comuni; var nota notabon;
data ganaron;
set a;
if result = 'gano' then output;
proc sort; by sexo;
proc means; by sexo;
var notabon;
run;
```

Respuesta ejercicio No. 11:

Incisos de a - f:

```
data a;
infile 'a:curso.dat';
input nombre $ 1-11 sexo $ nota estud;
If nota >=70 then result = 1;
Else result = 0;
If nota >=70 then total='aprobado';
Else total='reprobado';
proc print;
proc print; var nombre total;
proc sort; by sexo;
Proc freq; by sexo;
     tables total;
proc sort;
     by sexo;
proc means; by sexo;
     var nota;
Proc Glm;
     model nota=estud;
output predicted=Pnota;
proc plot;
   plot nota*estud='*' Pnota*estud='.'/overlay Vpos=15 hpos=30;
run;
```

inciso g:

```
data a;
infile 'a:curso.dat';
input nombre $ 1-11 sexo $ nota estud;
proc reg; model nota = estud;
output out=datares p=Pnota R=residuos;
proc print; var nota estud Pnota residuos;
proc univariate plot normal; var residuos;
proc plot; plot residuos*Pnota residuos*estud/vpos=15 hpos=30;
data a;
set datares;
resi_1=LAG1 (residuos);
proc print; var residuos resi_1;
```

```
proc plot;
plot residuos*resi_1 /vpos=15 hpos=30;
run;
```

Respuesta ejercicio No. 12:

```
data a;
input dietas pesokg @@;
cards;
1 1.2 1 0.8 1 1.4 1 1.7 1 2.3 2 0.8 2 0.4 2 0.3
2 0.6 2 0.9 3 1.1 3 1.2 3 1.5 3 0.9 3 1.2
proc print;
proc glm;
class dietas;
model pesokg = dietas;
means dietas/tukey;
run;
```

Respuesta ejercicio No. 13:

```
data a;
infile 'c:\horiz1.dat';
input nombre $ quiz1 quiz2 quiz3 domic1 domic2;
proc sort; by nombre;
proc print;
data b;
infile 'c:\horiz2.dat';
input nombre $ examen1 examen2;
proc sort; by nombre;
proc print;
data c;
merge a b;
      by nombre;
proc print;
pq = mean (of quiz1-quiz3) * 0.2;
pd = mean (of domic1-domic2) * 0.3;
ex1 = examen1 * 0.3;
ex2 = examen2 * 0.2;
notafin = pq+pd+ex1+ex2;
if notafin >= 70 then result='gano';
else result='perdio';
proc print;
run;
```

Respuesta ejercicio No. 14:

Incisos a y b:

```
libname maiz 'a:';
data maiz.maiz;
infile 'a:maiz.dat';
input y x1 x2 x3;
proc print;
run;
```

Incisos c, d y e:

```
libname maiz 'a:';
proc print data= maiz.maiz;
proc corr data= maiz.maiz;
var y x1 x2 x3;
proc glm data= maiz.maiz;
model y = x1;
proc glm data= maiz.maiz;
model y = x2;
proc glm data= maiz.maiz;
model y = x3;
run;
```

Respuesta ejercicio No. 15:

```
data a;
input cult $ nitrog;
cards;
3d0k1 19.4
3d0k1 32.6
3d0k1 27.0
3d0k1 32.1
3d0k1 33.0
3d0k5 17.7
3d0k5 24.8
3d0k5 27.9
3d0k5 25.2
3d0k5 24.3
3d0k4 17.0
3d0k4 19.4
3d0k4  9.1
3d0k4 11.9
3d0k4 15.8
3d0k7 20.7
3d0k7 21.0
3d0k7 20.5
3d0k7 18.8
3d0k7 18.6
3d0k9 14.3
3d0k9 14.4
3d0k9 11.8
3d0k9 11.6
3d0k9 14.2
3d0k8 17.3
3d0k8 19.4
3d0k8 19.1
3d0k8 16.9
3d0k8 20.8
title 'Contenido de nitrogeno de plantas de trébol rojo';
title2 'inoculadas con combinaciones de cultivos de cepas';
title3 'Rhizobium troffoli y Rhizobium meliloti en miligramos';
proc print;
```

```
proc glm;
    classes cult;
    model nitrog=cult;
    means cult / duncan tukey;
run;
```

Respuesta ejercicio No. 16:

```
libname soya 'c:';
data soya.soya;
infile 'c:\soya.dat';
input parce rend;
if parce=1 then do; text=1; fert=1; end;
if parce=2 then do; text=1; fert=2; end;
if parce=3 then do; text=2; fert=1; end;
if parce=4 then do; text=2; fert=2; end;
if parce=5 then do; text=3; fert=1; end;
if parce=6 then do; text=3; fert=2; end;
proc print;
proc sort; by text;
proc means; var rend; by text;
proc ttest;
classes fert;
var rend;
run;
```

Respuesta ejercicio No. 17:

```
data a;
input nitrog cloro potasio combfol;
cards;
3.05   1.45   5.67 0.34
4.22   1.35   4.86 0.11
3.34   0.26   4.19 0.38
3.77   0.23   4.42 0.68
3.52   1.10   3.17 0.18
3.54   0.76   2.76 0.00
3.74   1.59   3.81 0.08
3.78   0.39   3.23 0.11
2.92   0.39   5.44 1.53
3.10   0.64   6.16 0.77
2.86   0.82   5.48 1.17
2.78   0.64   4.62 1.01
2.22   0.85   4.49 0.89
2.67   0.90   5.59 1.40
3.12   0.92   5.86 1.05
3.03   0.97   6.60 1.15
2.45   0.18   4.51 1.49
4.12   0.62   5.31 0.51
4.61   0.51   5.16 0.18
3.94   0.45   4.45 0.34
4.12   1.79   6.17 0.36
```

```
2.93   0.25   3.38 0.89
2.66   0.31   3.51 0.91
3.17   0.20   3.08 0.92
2.79   0.24   3.98 1.35
2.61   0.20   3.64 1.33
3.74   2.27   6.50 0.23
3.13   1.48   4.28 0.26
3.49   0.25   4.71 0.73
2.94   2.22   4.58 0.23
title 'Porcentajes de nitrogeno, cloro, potasio y ';
title2 'combustion foliar en segundos (log)';
title3 'en muestras de tabaco de campos de agricultores';
proc print;
proc glm;
   model combfol=nitrog cloro potasio;
proc stepwise;
   model combfol=nitrog cloro potasio;
run;
```

Respuesta ejercicio No. 18:

```
data a;
input DAP VOL;
cards;
10 22
30 51
17 33
25 36
13 40
48 45
59 29
37 48
proc format;
value DAP       10-20 = '10-20'
                21-30 = '21-30'
                31-40 = '31-40'
                41-50 = '41-50'
                51-60 = '51-60';
value VOL       22-30 = '22-30'
                31-39 = '31-39'
                40-48 = '40-48'
                49-57 = '49-57'
                58-66 = '58-66'
                67-75 = '67-75';
proc print;
var DAP VOL;
format DAP DAP. VOL VOL.;
proc freq;
```

```
tables DAP VOL;
format DAP DAP. VOL VOL.;
run;
```

Respuesta ejercicio No. 19:

```
data a;
input parce dietaA dietaB;
diferen = dietaA-dietaB;
cards;
1   24.3   24.2
2   27.2   28.1
3   28.6   25.9
4   26.7   27.0
5   25.3   25.2
6   28.5   26.9
7   23.7   23.6
8   27.0   28.3
9   26.1   26.1
proc print;
proc means mean t prt;
var diferen;
run;
```

Respuesta ejercicio No. 20:

```
data a;
input varied $ peso @@;
cards;
A 2.9 B 3.3 C 3.1 D 4.5 E 6.6
A 3.5 B 3.8 C 3.8 D 4.4 E 6.8
A 4.1 B 3.7 C 4.2 D 4.8 E 7.4
A 3.0 B 3.6 C 3.1 D 4.7 E 7.2
A 3.0 B 3.1 C 3.5 D 4.5 E 7.0
A 3.6 B 3.3 C 3.2 D 5.0 E 7.6
proc print;
proc GLM;
class varied;
model peso=varied;
means varied/duncan tukey;
means varied/duncan tukey
alpha = 0.01;
run;
```

/DM-PROSAS3.SAS/