



EBook Gratis

APRENDIZAJE

sas

Free unaffiliated eBook created from
Stack Overflow contributors.

#sas

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con sas.....	2
Examples.....	2
Instalación o configuración.....	2
Descripción general de Base SAS.....	2
Hola Mundo!.....	3
Arquitectura del servidor SAS.....	3
Versiones.....	4
Capítulo 2: Copiar un archivo, byte por byte.....	6
Introducción.....	6
Examples.....	6
Copiando cualquier archivo, byte por byte.....	6
Capítulo 3: Creando Variables Macro.....	7
Introducción.....	7
Examples.....	7
Usando% LET.....	7
Utilizando PROC SQL.....	7
Usando Call Symput () en un paso de DATA.....	8
Capítulo 4: Enviando un correo electrónico con SAS.....	9
Introducción.....	9
Parámetros.....	9
Examples.....	9
Enviando un email de texto básico con SAS.....	9
Adjuntar un archivo excel a su correo electrónico SAS.....	9
Enviando un correo electrónico SAS con un cuerpo HTML.....	10
Capítulo 5: Etiquetas SAS.....	12
Observaciones.....	12
Examples.....	12
Crear etiquetas de variable permanente en el paso DATOS.....	12
Capítulo 6: Formatos SAS.....	13

Introducción.....	13
Observaciones.....	13
Examples.....	14
Usando la declaración de formato.....	14
Uso de la declaración de formato para agrupar datos.....	14
Formatos personalizados.....	15
Uso de informat para leer datos.....	16
Capítulo 7: Hacer bucle.....	18
Examples.....	18
Hacer bucle.....	18
Macro hacer bucle.....	18
Capítulo 8: Informat en SAS.....	19
Introducción.....	19
Observaciones.....	19
Examples.....	19
Importando datos de Excel en SAS.....	19
Importando personaje vs numérico.....	20
Capítulo 9: Lectura de datos.....	21
Introducción.....	21
Examples.....	21
Leer archivo de texto con delimitador de coma.....	21
Leer datos del archivo excel.....	21
PROC IMPORT para Excel, importando una hoja específica.....	21
Capítulo 10: Longitud variable.....	22
Sintaxis.....	22
Parámetros.....	22
Examples.....	22
Asignar longitud a una variable de carácter.....	22
Capítulo 11: paso de datos.....	23
Examples.....	23
obteniendo datos con datos setp.....	23

Capítulo 12: Proc SQL	24
Examples	24
Crear un conjunto de datos vacío basado en un conjunto de datos existente	24
Sintaxis SELECT	24
Capítulo 13: Resolución de variables de macro entre comillas dentro de los pasos de PROC S	26
Introducción	26
Observaciones	26
Examples	26
Paso a través con macro variable que es una fecha	26
Capítulo 14: Usando Uniones en SAS	28
Introducción	28
Parámetros	28
Observaciones	28
Examples	28
Unión vertical	28
Unir internamente	29
Unirse a la izquierda	29
Unirse a la derecha	29
Unirse completo	30
Creditos	31

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [sas](#)

It is an unofficial and free sas ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official sas.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con sas

Examples

Instalación o configuración

SAS se puede ejecutar en el modelo cliente-servidor, utilizando el cliente grueso de Enterprise Guide o el cliente ligero de SAS Studio (habilitado para la web), o en el modo de "servidor local" donde hay un sistema SAS completamente funcional en una máquina local (Windows o el escritorio o servidor Unix / Linux que se ejecuta en modo interactivo) y se ejecuta en el modo Administrador de pantalla (el cliente local) o a través de uno de los clientes cliente-servidor mencionados anteriormente (conectado al servidor instalado localmente).

Por lo general, la instalación de SAS la realiza un administrador de SAS, que instalará el software desde un almacén de software personalizado para el sitio (y que a menudo es proporcionado directamente por el Instituto SAS).

Con el propósito de aprender SAS, también está la edición gratuita de la Universidad de SAS, que se puede instalar de forma gratuita con fines educativos por cualquier persona en una computadora con Windows, Mac o Unix / Linux. Está disponible directamente desde SAS, actualmente en la [página de SAS University Edition](#), ya sea ejecutando una instancia de AWS (en el nivel gratuito) o descargando una máquina virtual localmente. Consulte [la guía de instalación en SAS.com](#) para obtener instrucciones actualizadas, o más abajo para las instrucciones actuales (julio de 2016).

Para instalarlo localmente, primero descarga e instala Oracle Virtualbox 5.0 ([Windows / Mac / Linux](#)). Luego descargue la [imagen de disco](#) más reciente de [SAS University Edition](#), que es de aproximadamente 2 GB y requiere la configuración de un perfil de SAS.com.

Una vez que lo haya hecho, debe configurar la máquina virtual en VirtualBox. Importe la máquina virtual SAS como un dispositivo ("Importar dispositivo" en VirtualBox). Cree una carpeta para que SAS la use como su almacenamiento local (para que pueda colocar los archivos en una ubicación que SAS pueda ver) y configúrela como una Carpeta compartida en el cuadro de diálogo de configuración de la máquina. Configúralo para que se monte automáticamente.

Luego, inicie la máquina virtual SAS y, una vez que se haya iniciado, podrá conectarse a través de su navegador web, conectándose a <http://localhost:10080/> si usó la configuración predeterminada.

Si tiene problemas, los [foros de la comunidad de SAS - Analytics U](#) son los foros de proveedores para obtener soporte, o hacer una pregunta sobre el [desbordamiento de pila](#).

Descripción general de Base SAS

SAS es un sistema integrado de soluciones de software que le permite realizar las siguientes tareas:

- Entrada, recuperación y gestión de datos.
- Redacción de informes y diseño gráfico.
- análisis estadístico y matemático
- pronóstico de negocios y apoyo a las decisiones
- Investigación de operaciones y gestión de proyectos.
- desarrollo de aplicaciones

Cómo usas SAS depende de lo que quieras lograr. Algunas personas usan muchas de las capacidades del Sistema SAS y otras usan solo unas pocas.

Hola Mundo!

Debido a la estructura de SAS, hay tres formas principales de crear "¡Hola mundo!" ejemplos:

1. Dentro de un paso de datos para colocar un mensaje en el registro de SAS (`_null_` indica que no se debe crear un conjunto de datos de salida):

```
data _null_;
  put "Hell" "o World!";
run;
```

2. Dentro de un paso de datos para almacenar "¡Hola mundo!" dentro de una variable (`foo` denota que se debe crear un conjunto de datos de salida llamado `foo` que a) contenga solo un registro yb) contenga solo una variable: `bar` , que tiene un valor de `Hello World!`):

```
data foo ;
  bar="Hello" ;
  put bar= "World!";
run ;
```

3. A través del lenguaje de macros SAS (en 'código abierto' fuera de los pasos de datos). `&` identifica una llamada a una variable macro y `.` identifica el final de la variable (si no se desea un carácter de espacio en blanco):

```
%let foo=Hello;
%put &foo.o World!;
```

4. Híbrido: usando una variable macro en un paso de datos:

```
%let foo=Hello;

data _null_ ;
  put "&foo World!";
run ;
```

Arquitectura del servidor SAS

Descripción general : normalmente hay dos tipos de implementaciones de SAS:

1. Instalación única de la Fundación SAS (BASE SAS). Esto normalmente se instala en una PC. No ejecuta ningún software de servidor.
2. Implementación planificada de SAS para su arquitectura de servidor que instalará el entorno del servidor SAS junto con posiblemente cualquier software de cliente SAS.

Cuál de estos tendrá en su correo electrónico de Orden de Software SAS indicando planificación o no planificación. Si está realizando una instalación planificada, necesitará un archivo de plan para su pedido que primero su topología.

[Nota de instalación 44320: uso de planes de implementación durante una instalación de SAS®](#)

Arquitectura del servidor SAS

El entorno del servidor SAS se divide en 3 niveles diferentes:

1. **Servidores de metadatos SAS:** el servidor de metadatos SAS es responsable de administrar el entorno del servidor SAS, incluidas las bibliotecas, los usuarios y la configuración del servidor.
2. **Servidores de aplicaciones SAS: el servidor de aplicaciones SAS** es en su mayoría un servidor de cómputo desde el cual sus clientes normalmente iniciarían trabajos.
3. **Nivel (es) intermedio (s) de SAS** = El nivel intermedio de SAS es principalmente su nivel web que ejecuta sus aplicaciones web.
4. **Nivel de cliente:** el nivel de cliente son las aplicaciones cliente de los usuarios que utilizan para conectarse al entorno, como la Guía empresarial de SAS.

[Documento 363-2011 | Comprensión de la anatomía de una implementación de SAS®: ¿Qué hay en la sopa de mi servidor? Mark Schneider, Donna Bennett y Connie Robison, SAS Institute Inc., Cary, NC](#)

Topología:

El nivel de metadatos SAS, el nivel de servidor de aplicaciones SAS y el nivel medio de SAS se pueden instalar en un solo servidor de máquina o extenderse en varios servidores. Esto está determinado por el archivo del plan que tiene, debe cumplir con la topología deseada para su implementación.

Normalmente, la mayoría, si no todos los niveles de cliente son aplicaciones basadas en Windows, por lo que el nivel de cliente estaría en las estaciones de trabajo de los usuarios de SAS. Opcionalmente, probablemente también podrían instalarse en los servidores si se basan en Windows.

[Sistemas operativos compatibles con SAS](#)

Versiones

Las principales versiones actuales de SAS son 9.4 y 9.3, estas son las versiones del motor SAS

base que se usa más comúnmente en la actualidad. El enlace a las notas de la versión para las versiones 9.1 + y otra documentación relacionada se incluyen a continuación.

Tenga en cuenta que también hay varios paquetes y funciones que amplían la funcionalidad de SAS, y estos tienen su propia documentación y funcionalidad .

- [SAS 9.4 - Documentación clave y notas de la versión](#)
- [SAS 9.3 - Documentación clave y notas de la versión](#)
- [SAS 9.2 - Documentación clave y notas de la versión](#)
- [SAS 9.1.x - Documentación clave y notas de la versión](#)

Lea **Empezando con sas en línea**: <https://riptutorial.com/es/sas/topic/2108/empezando-con-sas>

Capítulo 2: Copiar un archivo, byte por byte

Introducción

Si está utilizando SAS para producir informes de algún tipo, se encontrará con la necesidad de copiar un archivo en algún momento. La mayoría de las veces he usado este método para copiar una plantilla de Excel y luego descargar datos a través de PROC EXPORT en el nuevo archivo que he creado.

Este es un gran ejemplo que he encontrado de Chris Hemedinger (<http://blogs.sas.com/content/sasdummy/2011/06/17/how-to-use-sas-data-step-to-copy-a-archivo-desde-cualquier-parte/>).

Examples

Copiando cualquier archivo, byte por byte

```
/* these IN and OUT filerefs can point to anything */
filename in "anyfilehere.xlsx";
filename out "anyfilehere.xlsx";

/* copy the file byte-for-byte */
data _null_;
  length filein 8 fileid 8;
  filein = fopen('in','I',1,'B');
  fileid = fopen('out','O',1,'B');
  rec = '20'x;
  do while(fread(filein)=0);
    rc = fget(filein,rec,1);
    rc = fput(fileid, rec);

    rc =fwrite(fileid);
  end;
  rc = fclose(filein);
  rc = fclose(fileid);
run;

filename in clear;
filename out clear;
```

Lea Copiar un archivo, byte por byte en línea: <https://riptutorial.com/es/sas/topic/9394/copiar-un-archivo--byte-por-byte>

Capítulo 3: Creando Variables Macro

Introducción

El uso de Variables Macro en sus programas SAS es una funcionalidad básica con la que todo programador de SAS debe estar familiarizado. El uso de variables de macro puede ayudarlo a mantener su código simple y genérico. El código genérico es un código reutilizable.

Examples

Usando %LET

Describiría a %LET como la forma más sencilla de crear una variable de macro en SAS.

```
%LET variableName = variableValue;
```

Ahora, en cualquier lugar que use &variableName , se resolverá a variableValue .

NOTA: es posible que desee considerar que variableValue por sí solo puede traerle errores de sintaxis, dependiendo de cuál es el valor y cómo se usa. Por ejemplo, si es una fecha y la está utilizando en el DÓNDE de una instrucción PROC SQL, deberá escribirse como "&variableName"d para que funcione correctamente.

Utilizando PROC SQL

El uso de PROC SQL es una buena manera de obtener resultados rápidos de una tabla y convertirlos en variables. Por lo general, encuentro que cuando quiero obtener un recuento de registros que acabo de cargar en una tabla, puedo hacer que ese recuento se convierta en una variable con una rápida llamada PROC SQL.

```
PROC SQL;  
SELECT  
    COUNT(*) INTO:aVariable  
FROM  
    MyTable  
  
;QUIT;
```

En el ejemplo anterior, aVariable representará cuántos registros existen en MyTable .

También puede utilizar PROC SQL para crear múltiples variables de macro.

```
PROC SQL;  
SELECT  
    a,  
    b,  
    c INTO:aVariable, :bVariable, :cVariable
```

```
FROM
  MyTable

;QUIT;
```

En el ejemplo anterior, las variables creadas en la instrucción INTO coincidirán con las columnas extraídas en el orden en que se devuelven desde la instrucción SELECT. Sin embargo, solo la primera fila de resultados se utilizará para llenar esas 3 variables.

Si desea almacenar más de una fila y está en la versión 6.11 o posterior, utilice el siguiente ejemplo:

```
PROC SQL;
  SELECT DISTINCT
    a,
    b,
    c INTO :aVariable1 - :aVariable5,
          :bVariable1 - :bVariable5,
          :cVariable1 - :cVariable5
  FROM
    MyTable
;QUIT;
```

Las palabras clave `THROUGH` y `THRU` pueden utilizarse en lugar del guión -

Usando Call Symput () en un paso de DATA

```
DATA _null_;
  CALL SYMPUT('testVariable','testValueText');
;RUN;
```

En el ejemplo anterior, `%PUT &testVariable;` se resolverá a `testvalueText` .

Puede encontrar la necesidad de formatear su variable dentro de la llamada SYMPUT ().

```
DATA _null_;
  CALL SYMPUT('testDate',COMPRESS(PUT(today(),date9.)));
;RUN;
```

En el ejemplo anterior, `%PUT &testDate;` se resolverá a `10MAR2017`

Lea **Creando Variables Macro en línea**: <https://riptutorial.com/es/sas/topic/9403/creando-variables-macro>

Capítulo 4: Enviando un correo electrónico con SAS

Introducción

Hay varias razones por las que podría encontrar la necesidad de capacidades de correo electrónico en SAS. Podría estar enviando un correo electrónico para notificar a alguien que un proceso pasó / falló, podría estar enviando un correo electrónico que contenga Variables de Macro que muestre cuántos registros se han cargado al final de su fuente de datos, o tal vez necesite enviar algunos archivos que contienen informes. Cualquiera que sea su necesidad, hay varias formas de enviar correos electrónicos y archivos en SAS.

Parámetros

Etiqueta / Atributo	Valor
LRECL	Este parámetro se utiliza para definir la longitud del registro al leer y escribir archivos. He resuelto muchos problemas simplemente configurando esto en su valor máximo, que es 32767. Es muy posible que configurar algo como esto en su valor máximo sea menos eficiente, pero al final del día hace el trabajo por mí sin cualquier pérdida de rendimiento sentida. (El rango para LRECL es 1-32767)

Examples

Enviando un email de texto básico con SAS

```
Filename myEmail EMAIL
  Subject = "My Email Subject"
  From    = "myFromAddress@email.com"
  To      = 'toAddress@email.com'
  CC      = 'ccAddress@email.com'
  Type    = 'Text/Plain';

Data _null_; File myEmail;
  PUT "Email content";
  PUT "&recordsCount loaded to your favorite table today!";
RUN;
```

Adjuntar un archivo excel a su correo electrónico SAS

```
Filename myEmail EMAIL
```

```

Subject = "My Email Subject "
From    = "myFromAddress@email.com"
To      = 'toAddress@email.com'
CC      = 'ccAddress@email.com'
Type    = 'Text/Plain'
ATTACH = ("my/excel/file/path/file.extension" content_type="application/vnd.ms-excel"
LRECL= 32767);

Data _null_; File myEmail;
  PUT "Email contentent";
  PUT "&recordsCount loaded to your favorite table today!";
RUN;

```

Enviando un correo electrónico SAS con un cuerpo HTML

Tome nota del tipo de correo electrónico: Escriba = 'text / html';

```

Filename myEmail EMAIL
  Subject = "My Email Subject "
  From    = "myFromAddress@email.com"
  To      = 'toAddress@email.com'
  CC      = 'ccAddress@email.com'
  Type    = 'text/html';

Data _null_; File myEmail;
PUT "
<html>
  <head>
    <style>
      table, th, td {
        border: 1px solid black;
        border-collapse: collapse;
      }
    </style>
  </head>
  <body>
    <p>Here is your email</p>
    <p>Go ahead, organize your data within an HTML table tag here!</p>
    <table>
      <tr>
        <th>
          column 1
        </th>
        <th>
          column 2
        </th>
      </tr>
      <tr>
        <td>
          &countOfRecords1
        </td>
        <td>
          &countOfRecords2
        </td>
      </tr>
    </table>
  </body>
</html>

```

```
";  
RUN;
```

Es muy posible que después de crear un correo electrónico HTML en SAS, encuentre que el HTML está distorsionado cuando recibe el correo electrónico. Esto es el resultado de que SAS coloca saltos en la siguiente línea en el texto de su PUT. Probablemente se colocó un descanso justo en medio de uno de los textos de su etiqueta. *En caso de que esto le suceda, intente mover sus etiquetas HTML. Puede que no sea bonito, pero es posible que tenga que hacer que algunas etiquetas compartan una línea para evitar que esto suceda.* Esto me sucedió a mí, y así es exactamente como solucioné esos problemas.

Lea [Enviando un correo electrónico con SAS en línea](https://riptutorial.com/es/sas/topic/9398/enviando-un-correo-electronico-con-sas):

<https://riptutorial.com/es/sas/topic/9398/enviando-un-correo-electronico-con-sas>

Capítulo 5: Etiquetas SAS

Observaciones

Las etiquetas se pueden usar para describir una variable que ayuda a mejorar la legibilidad de sus resultados. Las etiquetas se pueden crear de forma permanente en el paso de `DATA` o se pueden crear temporalmente en un paso de `PROC` .

Examples

Crear etiquetas de variable permanente en el paso DATOS

```
data table;
  set table;
  label variable1 = 'label1'
        variable2 = 'label2'
        variable3 = 'label3';
run;
```

Lea Etiquetas SAS en línea: <https://riptutorial.com/es/sas/topic/7877/etiquetas-sas>

Capítulo 6: Formatos SAS

Introducción

Los informes y formatos se utilizan para indicar a SAS cómo leer y escribir los datos respectivamente. Los inmatos se usan comúnmente en una base de datos cuando se leen datos de un archivo externo. Los inmatos rara vez se utilizan en los procesos. Los formatos se usan comúnmente tanto en pasos de datos como en PROCs.

Observaciones

Los formatos SAS convierten los valores numéricos o de caracteres en valores de caracteres. Se puede aplicar un `format` utilizando un `format` o `put` declaración `put` , que cambia la forma en que se muestra un valor, o usando la función `put` para almacenar el valor formateado en una nueva variable.

Hay cuatro categorías de formatos:

- **Carácter:** indica a SAS que escriba valores de datos de caracteres a partir de variables de caracteres.
- **Fecha y hora:** indica a SAS que escriba valores de datos a partir de variables que representan fechas, horas y tiempos de datos.
- **ISO 8601:** indica a SAS que escriba los valores de fecha, hora y fecha y hora utilizando el estándar ISO 8601.
- **Numérico:** indica a SAS que escriba valores de datos numéricos a partir de variables numéricas.

Los formatos usualmente toman la forma `<formatname><w>.<d>` , siendo `w` el ancho (incluidos los decimales y el punto), `d` siendo el número de decimales.

Formatos de fecha comunes (aplicados a valores de fecha SAS):

- `date9.` por ejemplo, 02AUG2016
- `ddmmyyn8.` por ejemplo, 02082016
- `ddmmyy8.` por ejemplo 02/08/16
- `yymmdd10.` ej. 20160802
- `year4.` por ejemplo, 2016

Formatos numéricos comunes (aplicados a los números):

- `comma11.0` por ejemplo 1,234,567
- `comma12.2` por ejemplo 1,234,567.00
- `dollar11.2` eg \$ 5,789.12
- `nlmnlgbp11.2`

por ejemplo £ 2,468.02

Otros formatos:

- `$hex8.` , convierte cadena a hexadecimal
- `$upcase.` , convierte la cadena a mayúsculas
- `$quote.` , encierre una cadena entre comillas

Puede encontrar una lista completa de formatos aquí>

<https://support.sas.com/documentation/cdl/en/lrdict/64316/HTML/default/viewer.htm#a001263753.htm>

Examples

Usando la declaración de formato

La declaración de `format` aplica el formato dado a la variable especificada solo *para fines de visualización* , es decir, el valor subyacente no cambia.

```
data example1 ;
  Date = '02AUG2016'd ; /* stored as a SAS date, i.e. a number */
  Date2 = '31AUG2016'd ;
  format Date monyy7. Date2 yymmddn8. ;
run ;
```

Fecha	Fecha 2
AUG2016	20160831

Uso de la declaración de formato para agrupar datos

Puede aplicar formatos dentro de un procedimiento, por ejemplo, para cambiar las agrupaciones dentro de un `proc summary` o una `proc freq` .

Agrupar fechas de SAS

```
data example2 ;
  do Date = '01JUN2016'dt to '31AUG2016'dt ;
    Days = 1 ;
    output ;
  end ;
run ;

/* Summarise by year & month */
proc summary data=example2 nway ;
  class Date ;
  var Days ;
  output out=example2_sum (drop=_TYPE_ _FREQ_) sum= ;
  format Date yymmn6. ; /* e.g. 201606 */
run ;
```

Fecha	Días
201606	30
201607	31
201608	31

```

/* Summarise by month & year */
proc summary data=example2 nway ;
  class Date ;
  var Days ;
  output out=example2_sum2 (drop=_TYPE_ _FREQ_) sum= ;
  format Date monyy7. ; /* e.g. JUN2016 */
run ;

```

Fecha	Días
JUN2016	30
JUL2016	31
AUG2016	31

La ventaja de utilizar un formato es que se conserva el orden natural.

Usando `sashelp.class` como ejemplo, digamos que desea comparar la frecuencia de la primera letra de cada nombre. Podría usar la función `substr()` para encontrar la primera letra y ejecutar una `proc freq` en la nueva variable. Alternativamente, puede aplicar los `$1.` Formato a la variable

Name :

```

proc freq data=sashelp.class ;
  table Name ;
  format Name $1. ;
run ;

```

Nombre	CONTAR
UNA	7
segundo	4
do	2
etc.	

Formatos personalizados

Los formatos personalizados, también conocidos como formatos definidos por el usuario, se

pueden crear y utilizar como cualquier otro formato predeterminado.

```
/*Create new character format for state variables*/
PROC FORMAT;
VALUE $statef          'CA' = 'California'
                      'MA' = 'Massachusetts'
                      'NY' = 'New York';

/*Once created, you can use your custom format in PROC and DATA steps*/
PROC PRINT DATA=table;
FORMAT state-var $statef.;
RUN;
```

La variable `state-var` se imprimirá de acuerdo con el nuevo formato. Por ejemplo, el valor 'CA' se imprimirá como 'California' . Si un valor no fue formateado, como 'CT' , ese valor se imprimirá tal como aparece en el conjunto de datos.

Uso de informat para leer datos.

Informats se utiliza para decirle a SAS cómo leer los datos y se identifica con una declaración

`informat .`

```
data test;
  infile test.csv;
  informat      id $6.
               date mmddyy10.
               cost comma10.2
;
input @1 id
      @7 date
      @20 cost
;
run;
```

Informats y Formats también se pueden usar juntos para leer los datos y escribirlos en un formato diferente, como con la siguiente variable salarial:

```
DATA workers;
  informat first last $16.;
  informat salary 12.1;
  informat birthdate 8.;
  input
    first $
    last $
    birthdate
    salary;
  format salary dollar10.;
datalines;
John Smith 19810505 54998.5
Jane Doe 19950925 45884.5
Frank James 19600222 70000.5
Jamie Love 19630530 292000.5
;
run;
```

Lea Formatos SAS en línea: <https://riptutorial.com/es/sas/topic/5010/formatos-sas>

Capítulo 7: Hacer bucle

Examples

Hacer bucle

```
DATA salary;
  /*define variables*/
  raise=0.1;
  salary=50000;
  year=1;
  /*do loop*/
  DO year=1 to 20 by 2;
    salary + salary*raise;
    output; /*generates an observation for each iteration of the do loop, optional*/
  END;
RUN;
```

Macro hacer bucle

```
%macro doloop;
  %do age=11 %to 15 %by 2;
    title Age=&age.;
    proc print data=sashelp.class(where=(age=&age.));
    run;
  %end;
%mend;
%doloop;
```

Lea Hacer bucle en línea: <https://riptutorial.com/es/sas/topic/7919/hacer-bucle>

Capítulo 8: Informat en SAS

Introducción

Los `informat`s SAS instruyen a SAS sobre cómo leer datos desde cualquier ubicación de entrada (como un archivo, una hoja de cálculo de Excel, una canalización con nombre o incluso otra variable de SAS, etc.) en una variable.

SAS tiene solo dos tipos de datos: carácter y numérico, y cada información es específica para almacenar el valor en un carácter o variable numérica. Si la variable de destino es un carácter, entonces el `informat` comenzará con un símbolo `$`, cualquier otra cosa será un `informat` numérico.

Observaciones

Los informes son muy importantes, especialmente cuando importamos datos de otros conjuntos de datos. Por ejemplo, la mayoría de las veces, mientras trabajamos en datos en tiempo real, extraemos datos de varias fuentes de datos (Oracle, Mysql, Teradata, etc.). Cada vez que importamos datos necesitamos especificar la declaración de información para que SAS pueda leer los datos correctamente.

Examples

Importando datos de Excel en SAS

Por ejemplo, a continuación se muestran los datos de muestra en una "prueba" de Excel,

Purchase_Date	Customer_Name	Price
05-05-2017	Adam	1075
06-05-2017	Noah	1093
07-05-2017	Peter	1072
08-05-2017	Louis	1101
09-05-2017	Zoe	1248
10-05-2017	Kevin	1045
11-05-2017	Messiah	1072
12-05-2017	John	1046
13-05-2017	Stephen	1043
14-05-2017	Solly	1113
15-05-2017	Jeevan	1137

Debe usar el siguiente código para importar esto con éxito,

```
Data Test;  
Infile 'D:\Test.csv';  
Delimiter=', ' Missover DSD Getnames=Yes;  
Informat Purchase_Date date9.;  
Informat Price dollarx10.2;  
Format Purchase_Date date9.;  
Format Price dollarx10.2;
```

```
run;
```

Informat in the above code helps SAS to read the data from Excel.
Format in the above code helps to write the data properly into SAS Data set.

Importando personaje vs numérico

El siguiente ejemplo utiliza la declaración de entrada para leer un valor de una fuente (en este caso, la cadena 123) en un destino de carácter y un destino numérico.

```
data test;  
  source = '123';  
  numeric_destination = input(source, best.);  
  character_destination = input(source, $3.);  
run;
```

Lea Informat en SAS en línea: <https://riptutorial.com/es/sas/topic/9888/informat-en-sas>

Capítulo 9: Lectura de datos

Introducción

La lectura de los datos en un archivo SAS se puede lograr usando varios enfoques, incluyendo la `datalines` comunicado, desde un archivo externo usando un `infile` declaración en el paso de datos, o leer datos de un archivo externo usando `proc import` . Además, puede leer datos de fuentes externas compatibles con odbc (por ejemplo, bases de datos SQL) utilizando los controladores odbc.

Examples

Leer archivo de texto con delimitador de coma.

```
DATA table-name;
  INFILE "file-path/file-name.csv" dsd;
  INPUT Name $ City $ Age;
RUN;
```

Leer datos del archivo excel

```
PROC IMPORT DATAFILE = "file-path/file-name.xlsx" OUT=data_set DBMS=XLSX REPLACE;
```

PROC IMPORT para Excel, importando una hoja específica

Habrà ocasiones en las que solo desee importar una hoja específica de un archivo de Excel con varias hojas. Para hacer eso, usaremos " **SHEET =** ".

```
PROC IMPORT
  OUT= YourNewTable
  DATAFILE= "myfolder/excelfilename.xlsx"
  DBMS=xlsx
  REPLACE;
  SHEET="Sheet1";
  GETNAMES=YES;
RUN;
```

También tome nota de la capacidad de especificar si la fila superior importada contiene o no nombres de columnas (**GETNAMES = YES** (o NO)).

Lea [Lectura de datos en línea](https://riptutorial.com/es/sas/topic/7989/lectura-de-datos): <https://riptutorial.com/es/sas/topic/7989/lectura-de-datos>

Capítulo 10: Longitud variable

Sintaxis

- LONGITUD variable (s) <\$> longitud;

Parámetros

Parámetro	Detalles
variable (s)	variable (s) que desea asignar una longitud a
PS	parámetro opcional que especifica si su variable es una variable de carácter
longitud	entero que especifica la longitud de la variable

Examples

Asignar longitud a una variable de carácter

```
data table;  
set table;  
length state_full $8;  
if state = 'KS' then state_full = 'Kansas';  
else if state = 'CO' then state_full = 'Colorado';  
else state_full = 'Other';  
run;
```

Lea Longitud variable en línea: <https://riptutorial.com/es/sas/topic/7883/longitud-variable>

Capítulo 11: paso de datos

Examples

obteniendo datos con datos setp

```
data newclass(keep=first_name sex weight yearborn);  
  set sashelp.class(drop=height rename=(name=first_name));  
  yearborn=year(date())-age;  
  if yearborn >2002;  
run;
```

Datos especifica el conjunto de datos de destino. Mantener la opción especifica columnas para imprimir a destino.

El conjunto especifica el conjunto de datos de origen. Drop especifica columnas para no tomar. Renombra el nombre de nombre a primer nombre.

El año es una variable numérica implícita calculada (columna).

Filtrar y datos de salida implícitos con `if` para alumnos nacidos después de 2002.

Lea paso de datos en línea: <https://riptutorial.com/es/sas/topic/10673/paso-de-datos>

Capítulo 12: Proc SQL

Examples

Crear un conjunto de datos vacío basado en un conjunto de datos existente

Método 1:

```
proc sql;
  create table foo like sashelp.class;
quit;
```

Método 2:

```
proc sql;
  create table bar as
    select * from sashelp.class (obs=0);
quit;
```

El método 1 debería ser la opción preferida.

Sintaxis SELECT

```
PROC SQL options;
  SELECT column(s)
FROM table-name | view-name
  WHERE expression
  GROUP BY column(s)
  HAVING expression
ORDER BY column(s);
QUIT;
```

Ejemplo 1:

```
proc sql;
  select name
         ,sex
  from sashelp.class ;
quit;
```

La instrucción SELECT se especifica en este orden:

```
1.select;
2.from;
3.where;
4.group by;
5.having;
6.order by.
```

Se requieren "seleccionar" y "desde". Las demás cláusulas son opcionales.

Lea Proc SQL en línea: <https://riptutorial.com/es/sas/topic/5870/proc-sql>

Capítulo 13: Resolución de variables de macro entre comillas dentro de los pasos de PROC SQL

Introducción

Uno de los desafíos que enfrenté cuando comencé a usar SAS no era solo pasar los datos de Variables de Macro a un paso de PROC SQL, sino también resolverlos correctamente si era necesario hacer citas. Cuando se pasa una cadena como valor o fecha / fecha y hora a un paso de PROC SQL, lo más probable es que tenga comillas simples a su alrededor cuando se resuelva.

He encontrado los mejores resultados al usar la función % BQUOTE para lograr esto.

Observaciones

Puede encontrar más información sobre la función % BQUOTE aquí:

<https://v8doc.sas.com/sashtml/macro/z4bquote.htm>

Examples

Paso a través con macro variable que es una fecha

Primero, colocaré mi fecha en una Variable Macro.

NOTA: Encuentro esa fecha9. funciona muy bien con IBM® Netezza® SQL y Transact-SQL. Utilice el formato que funcione para el tipo de SQL que está ejecutando.

```
data _null_;
    call symput ('testDate', COMPRESS (put (today (), date9.)));
;RUN;
%PUT &testDate;
```

Mi declaración % PUT se resuelve en: 10MAR2017

A continuación, deseo ejecutar un paso de PROC SQL y resolver esa Variable de macro en el interior para especificar una fecha.

```
PROC SQL;
CONNECT TO odbc AS alias (dsn=myServer user=userName password= pass);
CREATE TABLE TableName AS
SELECT *
FROM connection to alias
```

```
(
  SELECT *
  FROM
    Database.schema.MyTable
  WHERE
    DateColumn = %bquote('&testDate')
);
QUIT;
```

% bquote ('& testDate') se resolverá a '10MAR2017' cuando se ejecute el código.

Lea [Resolución de variables de macro entre comillas dentro de los pasos de PROC SQL en línea](https://riptutorial.com/es/sas/topic/9396/resolucion-de-variables-de-macro-entre-comillas-dentro-de-los-pasos-de-proc-sql):
<https://riptutorial.com/es/sas/topic/9396/resolucion-de-variables-de-macro-entre-comillas-dentro-de-los-pasos-de-proc-sql>

Capítulo 14: Usando Uniones en SAS

Introducción

Cada base de datos es una colección de tablas diferentes y cada tabla contiene datos diferentes de forma organizada. Mientras trabajamos con datos, la mayoría de las veces la información que necesitamos está dispersa en más de una tabla. Necesitamos unir / fusionar para obtener el resultado deseado.

En SAS utilizamos uniones mientras trabajamos con `Proc SQL` y usamos la combinación mientras trabajamos con el `Data step`. Ahora hablaremos solo de las uniones dentro de `Proc SQL`.

Parámetros

Tipo de unión	Salida
Proc Sql	Procedimiento SQL dentro de SAS
Crear mesa	Crea un conjunto de datos SAS
Seleccionar	Selecciona las variables requeridas de los respectivos conjuntos de datos
Dónde	Especifica condición particular
Dejar	Terminar el procedimiento

Observaciones

Como se mencionó en la introducción, también podemos usar `Merge` dentro de un `data step` que se tratará en un tema aparte. Las uniones desempeñan un papel muy importante para combinar y unificar los datos de acuerdo con el requisito.

Examples

Unión vertical

La unión vertical agrega el conjunto de datos B al conjunto de datos A, siempre que ambos tengan variables similares. Por ejemplo, tenemos ventas para el mes del 17 de enero en el conjunto de datos A y ventas para el 17 de febrero en el conjunto de datos B. Para crear un conjunto de datos C que tenga ventas tanto de enero como de febrero, usamos Vertical Join.

```
PROC SQL;  
CREATE TABLE C AS  
SELECT *
```

```
FROM A
UNION
SELECT *
FROM B;
QUIT;
```

Ahora el conjunto de datos C tiene observaciones tanto de A como de B y se adjunta verticalmente.

Unir internamente

La unión interna crea un conjunto de datos que contiene registros que tienen valores coincidentes de ambas tablas. Por ejemplo, tenemos un conjunto de datos A que contiene información del cliente y un conjunto de datos B que contiene los detalles de la tarjeta de crédito. Para obtener los detalles de la tarjeta de crédito de los clientes en el conjunto de datos A, creamos el conjunto de datos C

```
PROC SQL;
CREATE TABLE C AS
SELECT A.*, B.CC_NUM
FROM CUSTOMER A, CC_DETAILS B
WHERE A.CUSTOMERID=B.CUSTOMERID
QUIT;
```

El conjunto de datos C solo tendrá observaciones coincidentes de ambos conjuntos de datos.

Unirse a la izquierda

La combinación izquierda devuelve todas las observaciones en el conjunto de datos izquierdo independientemente de sus valores clave, pero solo las observaciones con valores clave coincidentes del conjunto de datos correcto. Teniendo en cuenta el mismo ejemplo que el anterior,

```
PROC SQL;
CREATE TABLE C AS
SELECT A.*, B.CC_NUMBER, B.START_DATE
FROM CUSTOMER A LEFT JOIN CC_DETAILS B
ON A.CUSTOMERID=B.CUSTOMERID
QUIT;
```

El conjunto de datos C contiene todos los valores de la tabla de la izquierda, más los valores coincidentes de la tabla de la derecha o los valores faltantes en el caso de que no haya coincidencia.

Unirse a la derecha

Al igual que la combinación izquierda, la combinación derecha selecciona todas las observaciones del conjunto de datos derecho y los registros coincidentes de la tabla izquierda.

```
PROC SQL;
```

```
CREATE TABLE C AS
SELECT A.*, B.CC_NUMBER, B.START_DATE
FROM CUSTOMER A RIGHT JOIN CC_DETAILS B
ON A.CUSTOMERID=B.CUSTOMERID
QUIT;
```

El conjunto de datos C contiene todos los valores de la tabla derecha, más los valores coincidentes de la tabla izquierda o los valores faltantes en el caso de que no haya coincidencia.

Unirse completo

La combinación completa selecciona todas las observaciones de ambos conjuntos de datos, pero faltan valores en los que el valor clave de cada observación se encuentra en una sola tabla.

```
PROC SQL;
CREATE TABLE C AS
SELECT A.*, B.CC_NUMBER, B.START_DATE
FROM CUSTOMER A FULL JOIN CC_DETAILS B
ON A.CUSTOMERID=B.CUSTOMERID
QUIT;
```

El conjunto de datos C contendrá todos los registros de ambas tablas y se completará . por partidos perdidos en ambos lados.

Lea Usando Uniones en SAS en línea: <https://riptutorial.com/es/sas/topic/9900/usando-uniones-en-sas>

Creditos

S. No	Capítulos	Contributors
1	Empezando con sas	Bendy , brusso , Chris J , Community , dcudonk , fl0r3k , Jay Stevens , Joe
2	Copiar un archivo, byte por byte	Joshua Schlichting
3	Creando Variables Macro	Joshua Schlichting
4	Enviando un correo electrónico con SAS	Joshua Schlichting
5	Etiquetas SAS	heydrien
6	Formatos SAS	Chris J , GForce , heydrien , Robert Penridge
7	Hacer bucle	heydrien , zuluk
8	Informat en SAS	Praneeth Rachumallu , Robert Penridge
9	Lectura de datos	GForce , heydrien , Joshua Schlichting
10	Longitud variable	heydrien
11	paso de datos	zuluk
12	Proc SQL	Altons , D. O. , Jay Stevens
13	Resolución de variables de macro entre comillas dentro de los pasos de PROC SQL	Joshua Schlichting
14	Usando Uniones en SAS	Praneeth Rachumallu