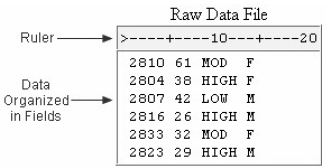
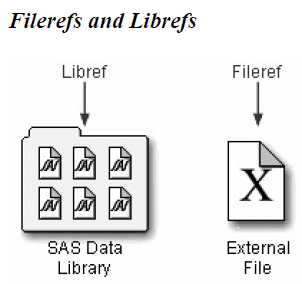
**Chapter V – Creating SAS Data Sets from External Files**

1. **INPUT FROM TEXT FILES**

Raw data file: an external text file whose records contain data values that are organized in files.

* Basic Statements for reading raw data:

|  |  |
| --- | --- |
| To do | SAS Statement |
| Reference SAS data library | **LIBNAME** statement |
| Reference external file | **FILENAME** statement |
| Name SAS data set | **DATA** statement |
| Identify external file | **INFILE** statement |
| Describe data | **INPUT** statement |
| Execute DATA step | **RUN** statement |
| Print the data set | **PROC PRINT** statement |
| Execute final program step | **RUN** statement |

* You do not need to use a **LIBNAME** statement in all situations, if you are storing the data set in a temporary SAS data set or if SAS has automatically assigned the libref for the permanent library that you are using.
* **FILENAME** statement
  + Use to point to the location of the eternal file that contain the data (Assign libref by using **LIBNAME** statement, assign fileref by using **FILENAME** statement)
  + **LIBNAME** also can associates fileref with the aggregate storage directory such as “C:\Users\Personal\Finances” (access fileref Finance: **FILENAME** Finance ' C:\Users\Personal\Finances ';)
  + Basic Code:

**FILENAME** *fileref* *'filename'*;

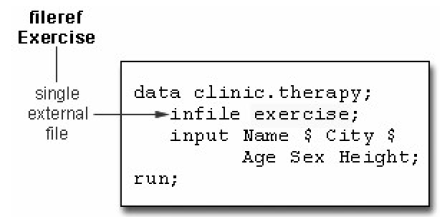
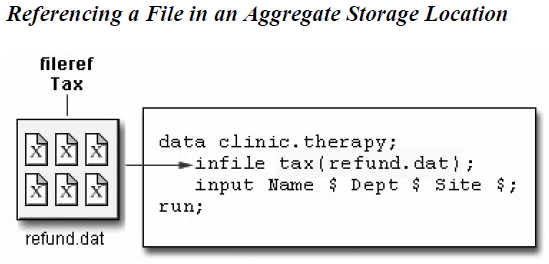
*fileref* is a name that you associate with an external file. The name must be 1 to 8 characters long, begin with a letter or underscore, and contain only letters, numerals, or underscores.

*‘FILENAME’* is the fully qualified name or location of the file.

Eg:

**FILENAME** tests 'c:\users\tmill.dat';

* Referencing a Fully Qualified Filename



In the Windows environment, you can omit the filename extension but you will need to add quotation

marks when referencing an external file, as in

Eg:

**INFILE** tax('refund');

* **DATA** step (***Chapter I, page 1***)
* Basic code:

**DATA** *SAS-data-set-1* <*...SAS-data-set-n*>;

*SAS-data-set* names (in the format *libref.filename*) the data set or data sets to be created.

* **Creating and modifying Variables** in DATA step
  + - Modify exist value or vreate new variable
    - Basic code:

*variable=expression***;**

*variable* names a new or existing variable

*expression* is any valid SAS expression

* + - Expression: sequence of operands and operators that form a set of instructions which are preformed to produce a new value
* Operands: variable names or constants (Numeric or character or both)
* Operators: special-character operators, grouping parentheses, or functions
* Operator in SAS expression

|  |  |  |  |
| --- | --- | --- | --- |
| Operator | Action | Example | Priority |
| - | negative prefix | **negative=-x** | I |
| \*\* | exponentiation | **raise=x\*\*y** | I |
| \* | multiplication | **mult=x\*y** | II |
| / | division | **divide=x/y** | II |
| + | addition | **sum=x+y** | III |
| - | subtraction | **diff=x-y** | III |

* + - Comparison Operator (Chapter 4, page 2)
* Logical Operator

|  |  |  |
| --- | --- | --- |
| Operator | Symbol | Meaning |
| And | & | If both expressions are true, then the compound expression is true. |
| Or | | | If either expression is true, then the compound expression is true. |

* Date Constants

Basic code:

‘ddmmmyy’d or “ddmmmyy”d

*dd* is a one- or two-digit value for the day

*mmm* is a three-letter abbreviation for the month (JAN, FEB, and so on)

*yy* or *yyyy* is a two- or four-digit value for the year, respectively.

* **IF** statement

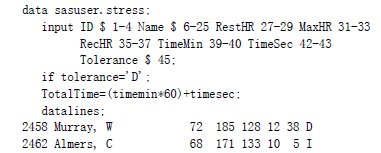
Basic code:

**IF** *expression***;**

*expression* is any valid SAS expression.

If the expression is *true*, the DATA step continues to process that observation.

If the expression is *false*, no further statements are processed for that observation, and control returns to the top of the DATA step.

* **DATALINES** statement (Reading Instream Data)
* To read instream data (直接输入文件内容without导入外部文件)
* A DATALINES statement as the last statement in the DATA step and immediately preceding the data lines
* A null statement (a single semicolon) to indicate the end of the input data.
* Basic code:

**DATALINES;**

* You can use only one DATALINES statement in a DATA step. Use separate DATA steps to enter multiple sets of data.
* You can also use LINES; or CARDS; as the last statement in a DATA step and immediately preceding the data lines. Both LINES and CARDS are aliases for the DATALINES statement. (50题, Q3)
* If your data contains semicolons, use the DATALINES4 statement plus a null statement that consists of four semicolons(;;;;).
* 注意：Null statement不需要RUN statement,因为DATALINES statement function as a step boundary
* **INFILE** statement

Indicate which file the data is in

Basic code:

**INFILE** *file-specification* <*options*>**;**

*file-specification* can take the form *fileref* to name a previously defined file reference or *'filename'* to point to the actual name and location of the file

*options* describe the input file's characteristics and specify how it is to be read with the **INFILE** statement.

Eg:

When raw data file has been assigned:

**INFILE** tests;

When raw data file has not been assigned, instead of using **FILEBANE** statement:

**INFILE** 'c:\irs\personal\refund.dat';

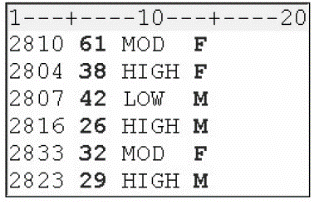
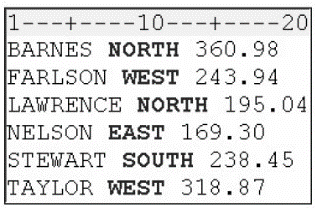
* Column Input

Specifies actual column location for value, the data must be standard character or numerical value in fixed fileds

* Fixed Field Data (Use PROC FSLIST to view the content and structure of the raw data files)

File contains data that is arranged in column, which means data begins and ends in the same columns.

Fixed Field Data Not arranged in columns



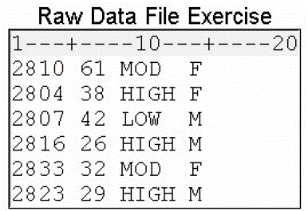
* **INPUT** statement

Describe the fields of raw data to be read and placed into SAS

Basic code:

**INPUT** *variable* <$> *startcol-endcol . . .***;**

*variable* is the SAS name that you assign to the field

the dollar sign ($) identifies the variable type as character (if the variable is numeric, then nothing appears here)

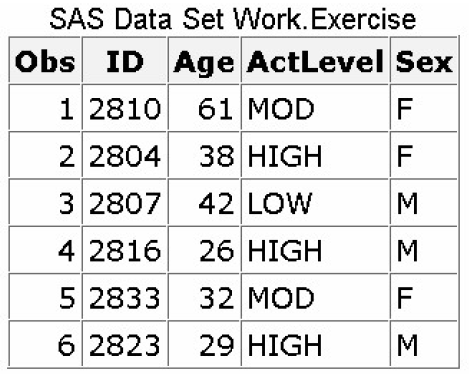
*startcol* represents the starting column for this variable

*endcol* represents the ending column for this variable

Eg:

The **INPUT** statement below assigns the character variable ID to the data in columns 1-4, the numeric variable Age to the data in columns 6-7, the character variable ActLevel to the data in columns 9-12, and the character variable Sex to the data in column 14.

**FILENAME** exer 'c:\users\exer.dat'; **Output**

**DATA** exercise;

**INFILE** exer;

**INPUT** ID $ 1-4 Age 6-7 ActLevel $ 9-12 Sex $ 14;

**RUN**;

注意：在name variable的时候，rule参照Chapter I，page 3的Roles for SAS Names

* Creating a raw data file
* **\_NULL\_** statement

**DATA \_NULL\_**;不建立数据集，**DATA**;自动建立一个数据集 (goal of your SAS program is to create a raw data file and not a SAS data set, it is inefficient to print a data set name in the **DATA** statement)

* **FILE** statement

Specifies the output raw data file,

Basic code:

**FILE** *file-specification* <*options*> <*operating-environment-options*>**;**

*file-specification* can take the form *fileref* to name a previously defined file reference or *'filename'* to point to the actual name and location of the file

*options* names options that are used in creating the output file

*operating-environment-options* names options that are specific to an operating environment (for more information, see the SAS documentation for your operating environment).

Eg:

**FILE** newdat; or file 'c:\clinic\patients\stress.dat';

* **PUT** statement

Describes the lines to write to the raw data file

Basic code:

**PUT** *variable startcol-endcol...***;**

*variable* is the name of the variable whose value is written

*startcol* indicates where in the line to begin writing the value

*endcol* indicates where in the line to end the value.

Eg:

**DATA \_NULL\_**;

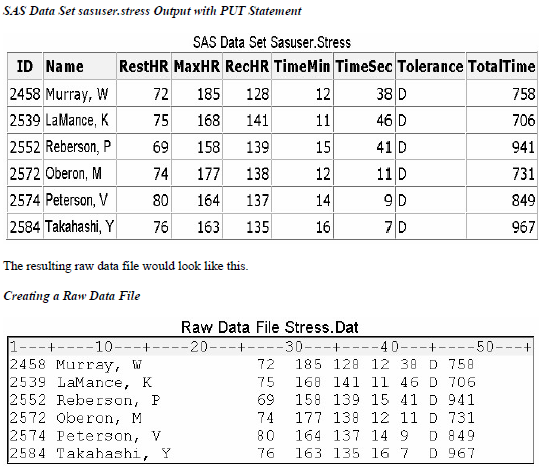
**SET** sasuser.stress;

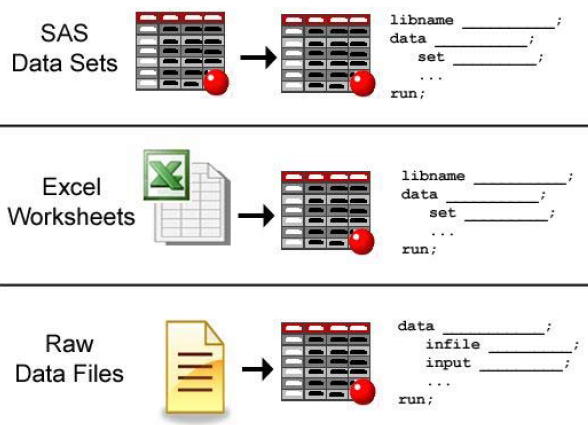
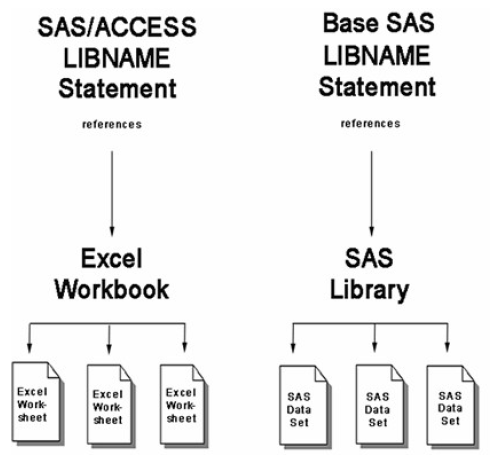
**FILE** 'c:\clinic\patients\stress.dat';

**PUT** id $ 1-4 name 6-25 resthr 27-29 maxhr 31-33 rechr 35-37 timemin 39-40 timesec 42-43 tolerance 45 totaltime 47-49;

**RUN**;

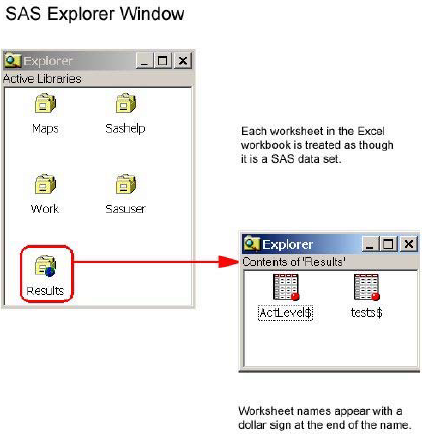
注意：If you do not execute a FILE statement before a PUT statement in the current iteration of the DATA step, SAS writes the lines to the SAS log. If you specify the PRINT fileref in the FILE statement, before the PUT statement, SAS writes the lines to the procedure output file.



1. **INPUT FROM EXCEL FILES**

* Basic Statements for reading Excel files:

|  |  |  |
| --- | --- | --- |
| To do | SAS Statement | Example |
| Reference Excel workbook file | **SAS/ACCESS LIBNAME** statement | **LIBNAME** results **'c:\users\exercise.xlsx';** |
| Output the contents of the SAS library | **PROC CONTENTS** statement | **PROC CONTENTS** data=results.\_all\_; |
| Execute **PROC CONTENTS** statement | **RUN** statement | **RUN;** |
| Name and create a new SAS data set | **DATA** statement | **DATA** work.stress |
| Read in an Excel worksheet | **SET** statement | **SET** results.’ActLevel&;n |
| Execute DATA step | **RUN** statement | **RUN;** |
| Print the data set | **PROC PRINT** statement | **PROC PRINT** data=stress; |
| Execute final program step | **RUN** statement | **RUN;** |

* **SAS/ACCESS LIBNAME** statement

Basic code:

**LIBNAME** *libref 'location-of-Excel-workbook '* <*options*>**;**

*libref* is a name that you associate with an Excel workbook.

*'location-of-Excel-workbook’* is the physical location of the Excel workbook.

Eg:

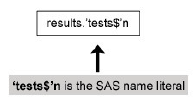
**LIBNAME** results **'c:\users\exercise.xlsx';**

* Referencing an Excel Workbook
* The **LIBNAME** statement creates the libref **results**, which points to the Excel workbook exercise.xlsx with two worksheets, **test** and **ActLevel.**

After the **LIBNAME** statement, check the Explorer Window:

* Name Literals

A name token that is expressed as a string within special character ($), followed by the upper or lowercase letter n. All Excel worksheets are designated this way.



* Name Ranges
* A named range is a range of cells within a worksheet that you define in Excel and assign a name to.
* Data set created from the named range will have no dollar sign ($) appended to its name
* **PROC CONTENTS** statement (Chapter II, page 2)

To using the SAS explorer window to view library data, you can also use the CONTENTS procedure with the \_ALL\_ keyword to produce information about a data library and its contents

Eg:

**PROC** **CONTENTS** data=results.\_**ALL**\_;

**RUN**;



* **DATA** statement (Same as import raw data file)

Eg:

**DATA** work.stress;

**SET** results.'ActLevel$'n;

**WHERE** ActLevel='HIGH';

**RUN**;

* **PROC PRINT** statement
* Disassociating a Libref

**LIBNAME** *libref CLEAR*

Eg:

**LIBNAME** results clear

* **LIBNAME** options

Basic code:

**LIBNAME** libref 'location-of-Excel-workbook' <options>;

Eg:

**LIBNAME** doctors 'c:\clinicNotes\addresses.xlsx' mixed=yes;

* **DBMAX\_TEXT=n**

Indicates the length of the longest character string where n is any integer between 256 and 32,767 inclusive. Any character string with a length greater than this value is truncated. The default is 1024.

* **GETNAMES=YES|NO**

Determines whether SAS will use the first row of data in an Excel worksheet or range as column names.

**YES** specifies to use the first row of data in an Excel worksheet or range as column names.

**NO** specifies not to use the first row of data in an Excel worksheet or range as column names. SAS generates and uses the variable names F1, F2, F3, and so on.

**The default is YES.**

* **MIXED=YES|NO**

Specifies whether to import data with both character and numeric values and convert all data to character.

**YES** specifies that all data values will be converted to character.

**NO** specifies that numeric data will be missing when a character type is assigned. Character data will be missing when a numeric data type is assigned.

**The default is NO.**

* **SCANTEXT=YES|NO**

Specifies whether to read the entire data column and use the length of the longest string found as the SAS column width.

**YES** scans the entire data column and uses the longest string value to determine the SAS column width.

**NO** does not scan the column and defaults to a width of 255.

**The default is YES.**

* **SCANTIME=YES|NO**

Specifies whether to scan all row values in a date/time column and automatically determine the TIME. format if only time values exist.

YES specifies that a column with only time values be assigned the TIME8. format.

NO specifies that a column with only time values be assigned the DATE9. format.

The default is NO.

* **USEDATE=YES|NO**

Specifies whether to use the DATE9. format for date/time values in Excel workbooks.

**YES** specifies that date/time values be assigned the DATE9. format.

**NO** specifies that date/time values be assigned the DATETIME. format.

**The default is YES.**

1. Points to Remember

* **LIBNAME** and **FILENAME** statements are global. Librefs and filerefs remain in effect until you change them, cancel them, or end your SAS session.
* For each field of raw data that you read into your SAS data set, you *must* specify the following in the **INPUT** statement: a valid SAS variable name, a type (character or numeric), a starting column, and if necessary, an ending column.
* When you use column input, you can read any or all fields from the raw data file, read the fields in any order, and specify only the starting column for variables whose values occupy only one column.
* Column input is appropriate only in some situations. When you use column input, your data *must* be standard character and numeric values, and these values *must* be in fixed fields. That is, values for a particular variable must be in the same location in all records.

1. Sample Program

* Reading Data from an External File

**LIBNAME** clinic 'c:\bethesda\patients\admit';

**FILENAME** admit 'c:\clinic\patients\admit.dat';

**DATA** clinic.admittan;

**INFILE** admit OBS=5;

**INPUT** ID $ 1-4 Name $ 6-25 RestHR 27-29 MaxHR 31-33 RecHR 35-37 TimeMin 39-40 TimeSec 42-43

Tolerance $ 45;

**IF** tolerance='D';

TotalTime=(timemin\*60)+timesec;

**RUN**;

**PROC** **PRINT** data=clinic.admittan;

**RUN**;

* Reading Instream Data

**LIBNAME** clinic 'c:\bethesda\patients\admit';

**DATA** clinic.group1;

**INPUT** ID $ 1-4 Name $ 6-25 RestHR 27-29 MaxHR 31-33 RecHR 35-37

TimeMin 39-40 TimeSec 42-43

Tolerance $ 45;

**IF** tolerance='D';

TotalTime=(timemin\*60)+timesec;

**DATALINES**;

2458 Murray, W 72 185 128 12 38 D

2462 Almers, C 68 171 133 10 5 I

2501 Bonaventure, T 78 177 139 11 13 I

2523 Johnson, R 69 162 114 9 42 S

2539 LaMance, K 75 168 141 11 46 D

2544 Jones, M 79 187 136 12 26 N

2595 Warren, C 77 170 136 12 10 S;

**PROC** **PRINT** data=clinic.group1;

**RUN**;

* Reading Excel Data

**LIBNAME** sasuser 'c:\users\admit.xlsx' **MIXED**=yes;

**PROC CONTESNTS** data=sasuser.\_**ALL**\_;

**RUN**;

**PROC** **PRINT** data=sasuser.'worksheet1$'n;

**RUN**;

* Creating an Excel Worksheet

**LIBNAME** clinic 'c:\Users\mylaptop\admitxl.xlsx' **MIXED** =yes;

**DATA** clinic.admit;

**SET** work.admit;

**RUN**;

练习

1. Which statement identifies the name of a raw data file to be read with the fileref Products and specifies that the **DATA** step read only records 1-15?
2. **INFILE** products **OBS** 15;
3. **INFILE** products **OBS** =15;
4. **INFILE** products **OBS** =15;
5. **INFILE** products 1-15;
6. Which of the following programs correctly writes the observations from the data set below to a raw data file?



1. **DATA** \_**NULL**\_;

**SET** work.patients;

**INFILE** 'c:\clinic\patients\referals.dat';

**INPUT** id $ 1-4 sex 6 $ age 8-9 height 11-12 weight 14-16 pulse 18-20;

**RUN**;

1. **DATA** referals.dat;

**SET** work.patients;

**INPUT** id $ 1-4 sex $ 6 age 8-9 height 11-12 weight 14-16 pulse 18-20;

**RUN**;

1. **DATA** \_ **NULL** \_;

**SET** work.patients;

**FILE**c:\clinic\patients\referals.dat;

**PUT**id $ 1-4 sex 6 $ age 8-9 height 11-12 weight 14-16 pulse 18-20;

**RUN**;

1. **DATA** \_ **NULL** \_;

**SET** work.patients;

**FILE**'c:\clinic\patients\referals.dat';

**PUT**id $ 1-4 sex 6 $ age 8-9 height 11-12 weight 14-16 pulse 18-20;

**RUN**;

1. Which program creates the output shown below?

***Raw Data and SAS Output Data Set***



1. **DATA** work.salesrep;

**INFILE** empdata;

**INPUT** ID $ 1-4 LastName $ 6-12 FirstName $ 14-18 City $ 20-29;

**RUN**;

**PROC** **PRINT** data=work.salesrep;

**RUN**;

1. **DATA** work.salesrep;

**INFILE** empdata;

**INPUT** ID $ 1-4 Name $ 6-12 FirstName $ 14-18 City $ 20-29;

**RUN**;

**PROC** **INPUT** data=work.salesrep;

**RUN**;

1. **DATA** work.salesrep;

**INFILE** empdata;

**INPUT** ID $ 1-4 name1 $ 6-12 name2 $ 14-18 City $ 20-29;

**RUN**;

**PROC** **INPUT** data=work.salesrep;

**RUN**;

1. all of the above.
2. Which statement correctly reads the fields in the following order: StockNumber, Price, Item, Finish, Style?

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Start Column** | **End Column** | **Data Type** |
| StockNumber | 1 | 3 | character |
| Finish | 5 | 9 | character |
| Style | 11 | 18 | character |
| Item | 20 | 24 | character |
| Price | 27 | 32 | numeric |



1. **INPUT** StockNumber $ 1-3 Finish $ 5-9 Style $ 11-18 Item $ 20-24 Price 27-32;
2. **INPUT** StockNumber $ 1-3 Price 27-32 Item $ 20-24 Finish $ 5-9 Style $ 11-18;
3. **INPUT** $ StockNumber 1-3 Price 27-32 $ Item 20-24 $ Finish 5-9 $ Style 11-18;
4. **INPUT** StockNumber $ 1-3 Price $ 27-32 Item $ 20-24 Finish $ 5-9 Style $ 11-18;
5. Which program correctly reads instream data?
6. **DATA** finance.newloan;

**INPUT** datalines;

**IF** country='JAPAN';

MonthAvg=amount/12;

1998 US CARS 194324.12

1998 US TRUCKS 142290.30

1998 CANADA CARS 10483.44

1998 CANADA TRUCKS 93543.64

1998 MEXICO CARS 22500.57

1998 MEXICO TRUCKS 10098.88

1998 JAPAN CARS 15066.43

1998 JAPAN TRUCKS 40700.34

;

1. **DATA** finance.newloan;

**INPUT** Year 1-4 Country $ 6-11 Vehicle $ 13-18 Amount 20-28;

**IF** country='JAPAN';

MonthAvg=amount/12;

**DATALINES**;

**RUN**;

1. **DATA** finance.newloan;

**INPUT** Year 1-4 Country 6-11 Vehicle 13-18 Amount 20-28;

**IF** country='JAPAN';

MonthAvg=amount/12;

**DATALINES**;

1998 US CARS 194324.12

1998 US TRUCKS 142290.30

1998 CANADA CARS 10483.44

1998 CANADA TRUCKS 93543.64

1998 MEXICO CARS 22500.57

1998 MEXICO TRUCKS 10098.88

1998 JAPAN CARS 15066.43

1998 JAPAN TRUCKS 40700.34

;

1. DATA finance.newloan;

**INPUT** Year 1-4 Country $ 6-11 Vehicle $ 13-18 Amount 20-28;

**IF** country='JAPAN';

MonthAvg=amount/12;

**DATALINES**;

1998 US CARS 194324.12

1998 US TRUCKS 142290.30

1998 CANADA CARS 10483.44

1998 CANADA TRUCKS 93543.64

1998 MEXICO CARS 22500.57

1998 MEXICO TRUCKS 10098.88

1998 JAPAN CARS 15066.43

1. AN TRUCKS 40700.34

;

1. Which SAS statement subsets the raw data shown below so that only the observations in which Sex (in the second field) has a value of F are processed?



1. **IF** sex=f;
2. **IF** sex=F;
3. **IF** sex='F';
4. a or b