

Chapter 5

Creating SAS Data Sets From External Files

Topics

- Reference a raw data file
- Specify a raw data file to be read
- Read standard character and numeric values in fixed fields
- Create new variables and assign values
- Select observations based on conditions
- Read instream data
- Read a SAS data set and write the observations out to a raw data file.
- Use the DATA step to create a SAS data set from an Excel worksheet
- Use the SAS/ACCESS LIBNAME statement to read from an Excel worksheet
- Create an Excel worksheet from a SAS data set
- Use the IMPORT Wizard to read external files

Data Sources

1. Raw Data

- Internal (Instream) raw data:

- Data are typed directly into the SAS program
(Works well with small amounts of data).

- External raw data files:

- Data are external to SAS - referred to as text, ASCII, sequential or flat files.

2. Other software data files:

Microsoft Excel, Oracle, dBase, Microsoft Access, JMP files and more can be imported into SAS using the Import Wizard.

Methods for Getting Data into SAS

- Data step
- Import Wizard (Point-and-Click method)
- (Proc Import)

Reading Raw Data Files

A raw data file is an external text file. The data in the file are organized in fields. Raw data files are non-proprietary.

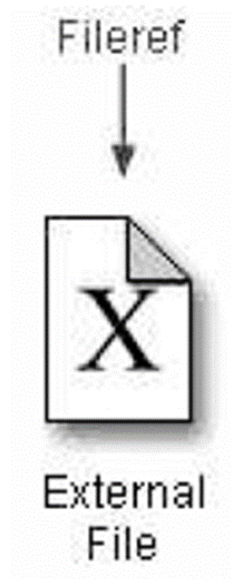
Raw Data File

Ruler →	>-----+-----10-----+-----20
Data Organized in Fields →	2810 61 MOD F
	2804 38 HIGH F
	2807 42 LOW M
	2816 26 HIGH M
	2833 32 MOD F
	2823 29 HIGH M

Basic Statements for Reading Data

To	Use
Reference a SAS data library	LIBNAME statement
Reference an external file	FILENAME statement
Name/create a SAS data set	DATA statement
Identify an external file	INFILE statement
Describe data	INPUT statement
Print the data set	PROC PRINT statement

Referencing a Fully Qualified Filename



- A fileref is global.
- The same name rule as libref applies.

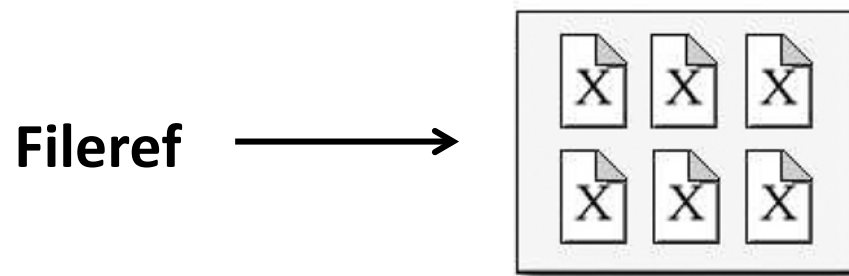
Define:

```
FILENAME test 'c: \users\tmill.dat';
```

Use:

```
INFILE test;
```

Referencing a file in an Aggregate File Storage Location (Folder)



```
FILENAME finance 'c:\users\personal\finances';
```

To reference a file called *refund* in the aggregate storage location, finance, you use:

```
INFILE finance ('refund');
```


Column Input and Fix-field Data

- **Column input:** a raw data input method that specifies actual column locations for values to read. It is appropriate only for standard numeric or character values that are in fixed fields.
- **Fixed-field data:** the data that the values for a particular field begin and end in the same columns.

Raw Data File Staff

1---+-----10---+-----20---+-----	
EVANS	DONNY 112 29,996.63
HELMS	LISA 105 18,567.23
HIGGINS	JOHN 111 25,309.00
LARSON	AMY 113 32,696.78
MOORE	MARY 112 28,945.89

Standard and Nonstandard Numeric Data

- **Standard numeric data** values can contain only
 - ✓ numerals
 - ✓ decimal points
 - ✓ numbers in scientific or E-notation (2.3E4, for example)
 - ✓ plus or minus signs
- **Nonstandard numeric data** includes
 - ✓ values that contain special characters, such as percent signs (%), dollar signs (\$), and commas (,)
 - ✓ date and time values
 - ✓ data in fraction, integer binary, real binary, and hexadecimal forms

You **cannot** use column input to read nonstandard numeric values.

General Form of an INPUT Statement

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

- **variable** : the SAS name that you assign to the field
- **the dollar sign (\$)**: it 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.

A DATA Step That Creates a Data Set From an External File

```
filename exer 'c:\users\exer.txt';
data exercise;
  infile exer;
  input ID $ 1-4 Age 6-7 ActLevel $ 9-12 Sex $ 14;
run;
```

Raw Data File Exercise

1	---	+	----	10	---	+	----	20
2810	61	MOD	F					
2804	38	HIGH	F					
2807	42	LOW	M					
2816	26	HIGH	M					
2833	32	MOD	F					
2823	29	HIGH	M					

SAS Data Set Work. Exercise

Obs	ID	Age	ActLevel	Sex
1	2810	61	MOD	F
2	2804	38	HIGH	F
3	2807	42	LOW	M
4	2816	26	HIGH	M
5	2833	32	MOD	F
6	2823	29	HIGH	M

Creating and Modifying Variables

- You can use SAS expressions to
 - ✓ transform variables
 - ✓ create new variables
 - ✓ conditionally process variables
 - ✓ calculate new values
 - ✓ assign new values.
- An **expression** is a sequence of operands and operators that form a set of instructions. The instructions are performed to produce a new value.
- **Operands** are variable names or constants. They can be numeric or character.
- **Operators** are special-character operators, grouping parentheses, or functions.

SAS Operators

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

Example: Creating and Modifying Variables

```
data sasuser.stress;  
  infile tests; *Assume the fileref tests was defined earlier;  
  input ID $ 1-4 Name $ 6-25 RestHR 27-28 MaxHR 30-32 RecHR  
        34-36 TimeMin 38-39 TimeSec 41-42 Tolerance $ 44;  
  TotalTime = timesec + timemin * 60 ;  
run;
```

SAS Data Set Sasuser Stress (Partial Listing)

ID	Name	RestHR	MaxHR	RecHR	TimeMin	TimeSec	Tolerance	TotalTime
2458	Murray, W	72	185	128	12	38	D	758
2462	Almers, C	68	171	133	10	5	I	605

Subsetting Data With the **IF** Statement

```
data sasuser.stress;
  infile tests;
  input ID $ 1-4 Name $ 6-25 RestHR 27-28 MaxHR 30-32
        RecHR 34-36 TimeMin 38-39 TimeSec 41-42 Tolerance $
        44;
  If Tolerance='D';
  TotalTime=timemin*60+timesec;
run;
```

SAS data set sasuser.stress (partial)

ID	Name	RestHR	MaxHR	RecHR	TimeMin	TimeSec	Tolerance	TotalTime
2458	Murray, W	72	185	128	12	38	D	758
2539	LaManc e, K	75	168	141	11	46	D	705

Reading Internal/Instream Data

- Use **DATALINES** (/ **LINES** / **CARDS**) statement as the last statement in the DATA step (except for the RUN statement if any) and immediately preceding the data lines.
- Use a null statement (a single semicolon) to indicate the end of the input data.
- Use the **DATALINES4** statement plus a null statement that consists of four semicolons (;;;;) if your data contains semicolons.

Examples: Reading Internal/Instream Data

```
data sasuser.stress;  
  input ID $ 1-4 Name $ 6-25 RestHR 27-28 MaxHR 30- 32  
        RecHR 34-36 TimeMin 38-38 TimeSec 41-42 Tolerance  
        $ 44;  
  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  
;
```

You **do not need a RUN statement** following the null statement (the semicolon after the data lines). The DATALINES statement functions as a step boundary, so the DATA step is executed as soon as SAS encounters it.

Examples: Reading Internal/Instream Data

```
data biblio;  
  input number citation $50.;  
  datalines4;  
1 KIRK, 1988  
2 LIN ET AL., 1995; BRADY, 1993  
3 BERG, 1990; ROA, 1994; WILLIAMS, 1992  
;;;
```

Creating a Raw Data File With the FILE and PUT Statements

```
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;
```

- The **_NULL_** keyword: enables you to use the DATA step without actually creating a SAS data set.
- A **SET statement** specifies the SAS data set that you want to read from.
- The **FILE statement** is to specify the output file.
- The **PUT statement** mirrors the capabilities of the INPUT statement. In this case you are working with column output.
- Because you are creating raw data, you don't need to follow character variable names with a dollar sign (\$).

Creating a Raw Data File With the FILE and PUT Statements

SAS Data Set Sasuser.Stress

ID	Name	RestHR	MaxHR	RecHR	TimeMin	TimeSec	Tolerance	TotalTime
2458	Murray, W	72	185	128	12	38	D	758
2539	LaMance, K	75	168	141	11	46	D	706
2552	Reberson, P	69	158	139	15	41	D	941
2572	Oberon, M	74	177	138	12	11	D	731
2574	Peterson, V	80	164	137	14	9	D	849
2584	Takahashi, Y	76	163	135	16	7	D	967



Raw Data File Stress.Dat

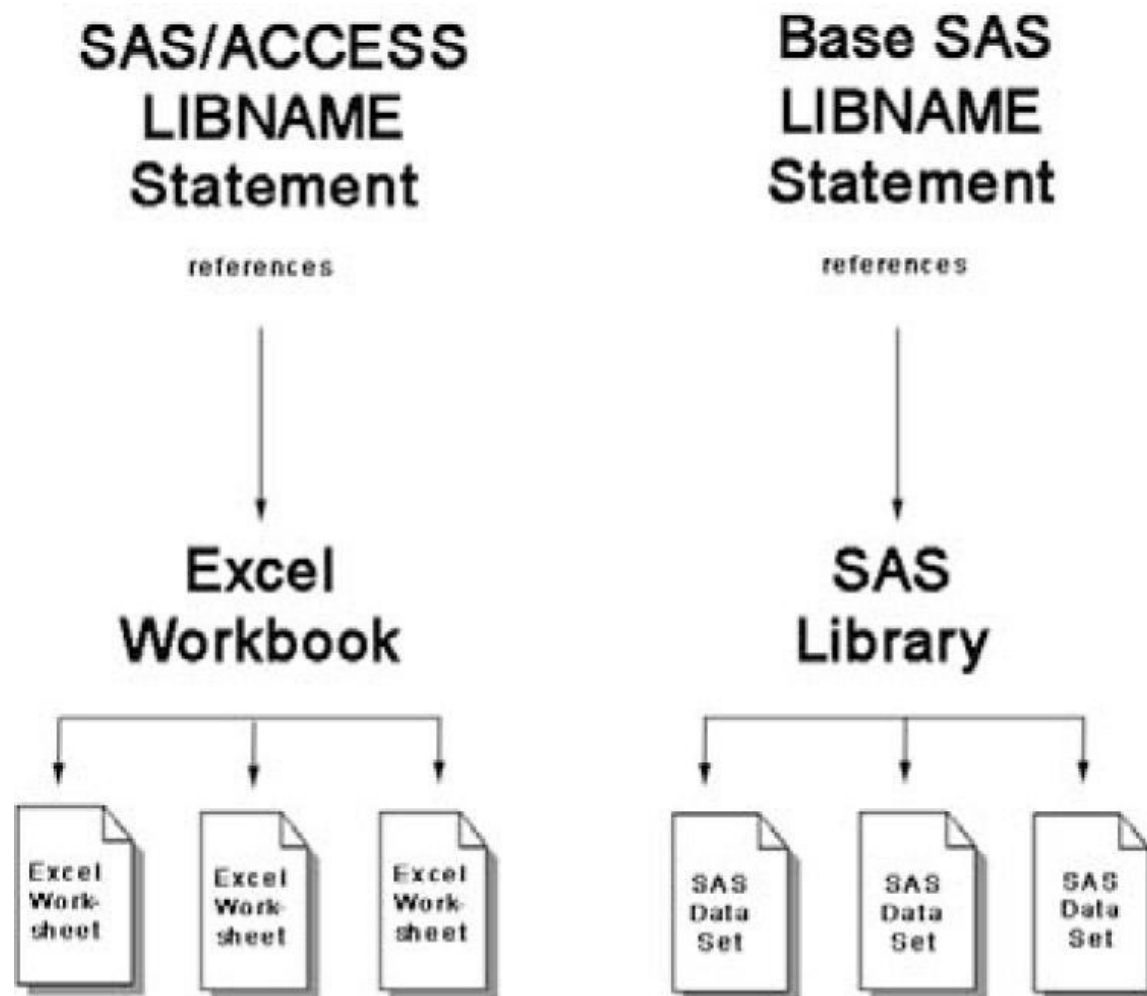
1	----	10	----	20	----	30	----	40	----	50	----
2458	Murray, W	72	185	128	12	38	D	758			
2539	LaMance, K	75	168	141	11	46	D	706			
2552	Reberson, P	69	158	139	15	41	D	941			
2572	Oberon, M	74	177	138	12	11	D	731			
2574	Peterson, V	80	164	137	14	9	D	849			
2584	Takahashi, Y	76	163	135	16	7	D	967			

Reading MS Excel Data

To read Excel data, you could use

- SAS/ACCESS LIBNAME statement
- Import Wizard (**Point-and-Click** method)

Use SAS/ACCESS LIBNAME Statement To Read Excel Data



Use SAS/ACCESS LIBNAME To Read Excel Data

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

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

OR **LIBNAME** results pcfiles path='c:\users\exercise.xlsx';

exercise.xlsx - Microsoft Excel

Home Insert Page Layout Formulas Data Review View Developer Add-Ins Acrobat

A1 ID

	A	B	C	D	E	F	G	H	I
1	ID	Name	RestHR	MaxHR	RecHR	TimeMin	TimeSec	Tolerance	TestDate
2	2458	Murray, W	72	185	128	12	38	D	8/25/2008
3	2462	Almers, C	68	171	133	10	5	I	6/26/2008
4	2501	Bonaventure, T	90	177	139	11	13	I	6/26/2008
5	2523	Johnson, R	69	162	114	9	42	S	7/14/2008
6	2539	LaMance, K	75	168	141	11	46	D	8/25/2008
7	2544	Jones, M	79	187	136	12	26	N	7/14/2008
8	2552	Reberson, P	69	158	139	15	41	D	8/25/2008
9	2555	King, E	70	167	122	13	13	I	7/14/2008
10	2563	Pitts, D	71	159	116	10	22	S	8/25/2008
11	2568	Eberhardt, S	72	182	122	16	49	N	6/26/2008

tests ActLevel

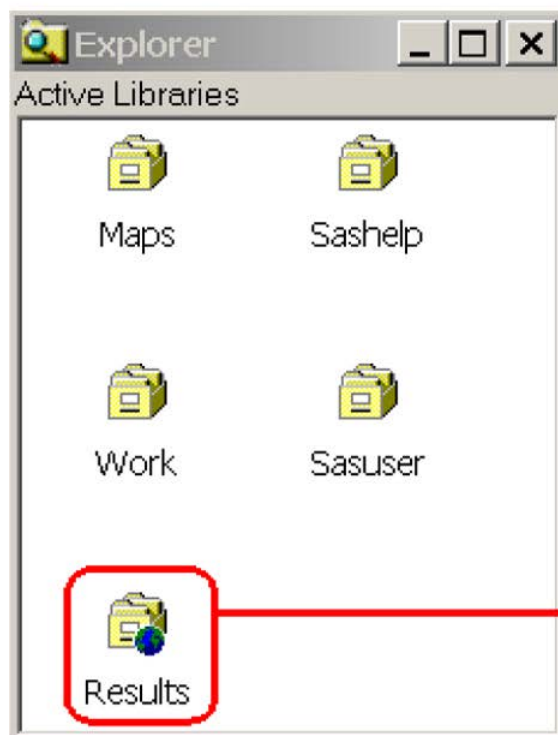
Ready 100%

Work-sheets

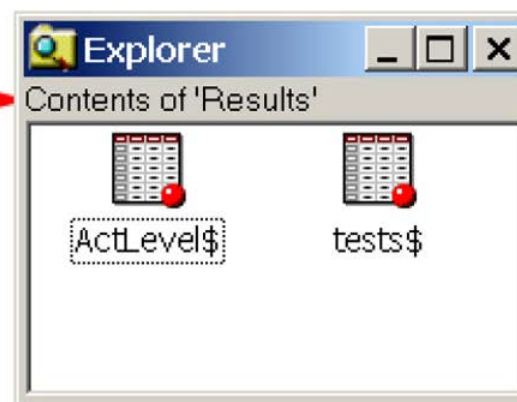
Cells formatted as dates

Use SAS/ACCESS LIBNAME To Read Excel Data

SAS Explorer Window



Each worksheet in the Excel workbook is treated as though it is a SAS data set.



Worksheet names appear with a dollar sign at the end of the name.

Use SAS/ACCESS LIBNAME To Read Excel Named Ranges—a named range of cells within a worksheet

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

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

OR **LIBNAME** results pcfiles path='c:\users\exercise.xlsx';

exercise.xlsx - Microsoft Excel

Home Insert Page Layout Formulas Data Review View Developer Add-Ins Acrobat

A1 ID

	A	B	C	D	E	F	G	H	I
1	ID	Name	RestHR	MaxHR	RecHR	TimeMin	TimeSec	Tolerance	TestDate
2	2458	Murray, W	72	185	128	12	38	D	8/25/2008
3	2462	Almers, C	68	171	133	10	5	I	6/26/2008
4	2501	Bonaventure, T	90	177	139	11	13	I	6/26/2008
5	2523	Johnson, R	69	162	114	9	42	S	7/14/2008
6	2539	LaMance, K	75	168	141	11	46	D	8/25/2008
7	2544	Jones, M	79	187	136	12	26	N	7/14/2008
8	2552	Reberson, P	69	158	139	15	41	D	8/25/2008
9	2555	King, E	70	167	122	13	13	I	7/14/2008
10	2563	Pitts, D	71	159	116	10	22	S	8/25/2008
11	2568	Eberhardt, S	72	182	122	16	49	N	6/26/2008

tests ActLevel

Ready

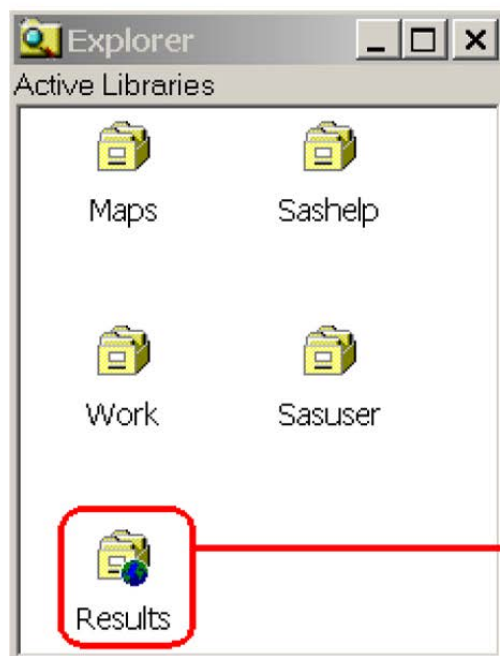
Week1

Work-sheets

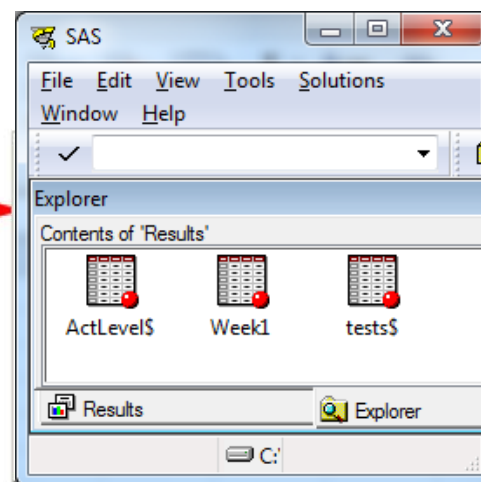
Cells formatted as dates

Use SAS/ACCESS LIBNAME To Read Excel Data

SAS Explorer Window



Each worksheet and named range is treated as separate worksheets



Worksheet names appear with a dollar sign at the end of the name but the data set created from the named range has no dollar sign appended to its name

Use SAS/ACCESS LIBNAME To Read Excel Data

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

```
DATA class.stress;
```

```
    SET results.'tests$'n;
```

```
RUN;
```

```
PROC CONTENTS data=results._all_;
```

```
RUN;
```

```
PROC PRINT data=results.'tests$'n;
```

```
RUN;
```



SAS name literal

Disassociate a Libref

An Excel file cannot be opened if there is a libref assigned to it. You use a LIBNAME statement to disassociate it.

```
LIBNAME results 'c:\users\exercise.xlsx';  
proc print data=results.'tests$'n;  
RUN;  
  
libname results clear;
```

Creating Excel Worksheet From a SAS Dataset

```
LIBNAME mydata 'D:\STSCI 5010 fall 2018\data\try.xlsx';
DATA mydata.ad;
    SET sasuser.admit;
RUN;
```

try [Read-Only] - Microsoft Excel

	A	B	C	D	E	F	G	H	I
1	ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee
2	2458	Murray, W	M	27	1	72	168	HIGH	85.2
3	2462	Almers, C	F	34	3	66	152	HIGH	124.8
4	2501	Bonavent	F	31	17	61	123	LOW	149.75
5	2523	Johnson, J	F	43	31	63	137	MOD	149.75
6	2539	LaMance, M		51	4	71	158	LOW	124.8
7	2544	Jones, M	M	29	6	76	193	HIGH	124.8
8	2552	Reberson, F		32	9	67	151	MOD	149.75
9	2555	King, E	M	35	13	70	173	MOD	149.75
10	2563	Pitts, D	M	34	22	73	154	LOW	124.8
11	2568	Eberhardt	F	49	27	64	172	LOW	124.8
12	2571	Nunnelly, F		44	19	66	140	HIGH	149.75
13	2572	Oberon, M	F	28	17	62	118	LOW	85.2

ad

Ready 100%

Use the IMPORT Wizard

- The Import Wizard enables you to create a SAS dataset from different types of external files, such as, dBase files, Excel files, MS Access tables, CSV files, etc.
- To activate the Import Wizard, select **File → Import Data** from the menu. After the following windows opens, follow the instructions.

