Chapter 7: Reading Delimited Raw Data Files

7.1 Using Standard Delimited Data as Input

7.2 Using Nonstandard Delimited Data as Input

Objectives

- Use the DATA step to create a SAS data set from a delimited raw data file.
- Examine the compilation and execution phases of the DATA step when reading a raw data file.
- Explicitly define the length of a variable by using the LENGTH statement.

Business Scenario

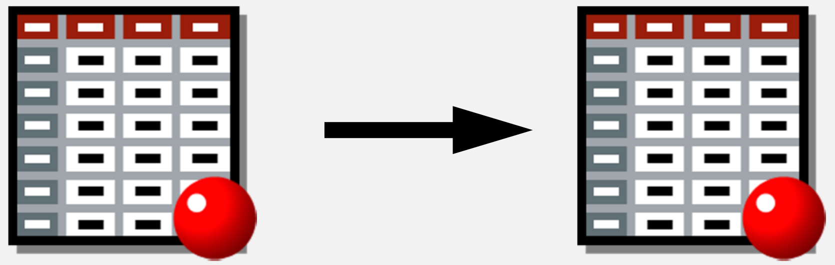
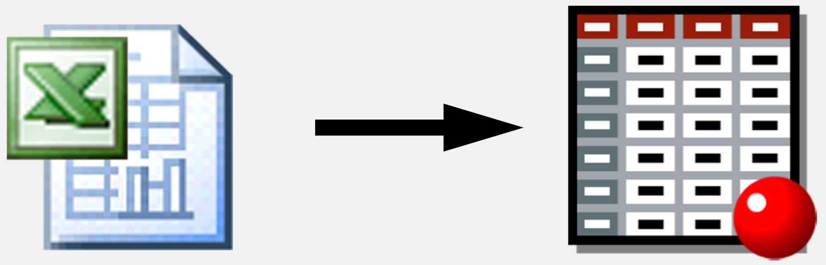
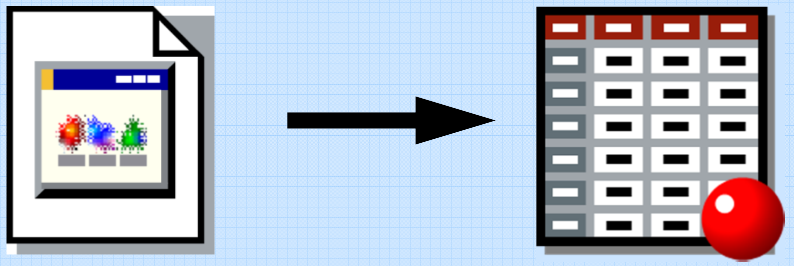
An existing data source contains information on Orion Star sales employees from Australia and the United States.

A new SAS data set needs to be created that contains a subset of this existing data source.

This new SAS data set must contain the following:

- only the employees from Australia who are Sales Representatives
- the employee's first name, last name, salary, job title, and hired date
- labels and formats in the descriptor portion

Business Scenario

Reading SAS Data Sets	
Reading Excel Worksheets	
Reading Delimited Raw Data Files	

Business Scenario

Reading SAS Data Sets	`libname ____________________;`
	`data ________________;`
	`    set ________________;`
	`    . . .`
	`run;`

Reading Excel Worksheets	`libname ____________________;`
	`data ________________;`
	`    set ________________;`
	`    . . .`
	`run;`

Reading Delimited Raw Data Files	`data ________________;`
	`    infile ________________;`
	`    input ________________;`
	`    . . .`
	`run;`

`sales.csv`

Partial `sales.csv`

comma delimited

```
120102,Tom,Zhou,M,108255,Sales Manager,AU,11AUG1969,06/01/1989
120103,Wilson,Dawes,M,87975,Sales Manager,AU,22JAN1949,01/01/1974
120121,Irenie,Elvish,F,26600,Sales Rep. II,AU,02AUG1944,01/01/1974
120122,Christina,Ngan,F,27475,Sales Rep. II,AU,27JUL1954,07/01/1978
120123,Kimiko,Hotstone,F,26190,Sales Rep. I,AU,28SEP1964,10/01/1985
120124,Lucian,Daymond,M,26480,Sales Rep. I,AU,13MAY1959,03/01/1979
120125,Fong,Hofmeister,M,32040,Sales Rep. IV,AU,06DEC1954,03/01/1979
120126,Satyakam,Denny,M,26780,Sales Rep. II,AU,20SEP1988,08/01/2006
120127,Sharryn,Clarkson,F,28100,Sales Rep. II,AU,04JAN1979,11/01/1998
120128,Monica,Kletschkus,F,30890,Sales Rep. IV,AU,14JUL1986,11/01/2006
120129,Alvin,Roebuck,M,30070,Sales Rep. III,AU,22NOV1964,10/01/1985
120130,Kevin,Lyon,M,26955,Sales Rep. I,AU,14DEC1984,05/01/2006
120131,Marinus,Surawski,M,26910,Sales Rep. I,AU,25SEP1979,01/01/2003
120132,Fancine,Kaiser,F,28525,Sales Rep. III,AU,05APR1949,10/01/1978
120133,Petrea,Soltau,F,27440,Sales Rep. II,AU,22APR1986,10/01/2006
120134,Sian,Shannan,M,28015,Sales Rep. II,AU,06JUN1949,01/01/1974
120135,Alexei,Platts,M,32490,Sales Rep. IV,AU,26JAN1969,10/01/1997
```

Business Scenario Syntax

Use the following statements to complete the scenario:

```
DATA output-SAS-data-set;
     LENGTH variable(s) $ length;
     INFILE 'raw-data-file-name';
     INPUT specifications;
     KEEP variable-list;
     LABEL variable = 'label'
           variable = 'label'
           variable = 'label';
     FORMAT variable(s) format;
RUN;
```

The DATA Statement (Review)

The *DATA statement* begins a DATA step and provides the name of the SAS data set being created.

```
DATA output-SAS-data-set;
     INFILE 'raw-data-file-name';
     INPUT specifications;
     <additional SAS statements>
RUN;
```

The DATA statement can create temporary or permanent data sets.

The INFILE Statement

The *INFILE statement* identifies the physical name of the raw data file to read with an INPUT statement.

```
DATA output-SAS-data-set;
      INFILE 'raw-data-file-name';
      INPUT specifications;
      <additional SAS statements>
RUN;
```

The physical name is the name that the operating environment uses to access the file.

The INFILE Statement

Examples:

Windows	infile 's:\workshop\sales.csv';
UNIX	infile '/users/*userid*/sales.csv';
z/OS (OS/390)	infile '.workshop.rawdata(sales)';

The INPUT Statement

The *INPUT statement* describes the arrangement of values in the raw data file and assigns input values to the corresponding SAS variables.

```
DATA output-SAS-data-set;
     INFILE 'raw-data-file-name';
     INPUT specifications;
     <additional SAS statements>
RUN;
```

The following are input specifications:

- column input
- formatted input
- list input

7.01 Multiple Answer Poll

Which types of raw data files do you read?

a. delimited raw data files
b. raw data files aligned in columns
c. other
d. none
e. not sure

List Input

To read with list input, data values

- must be separated with a delimiter
- can be in standard or nonstandard form.

Partial **`sales.csv`**

```
120102,Tom,Zhou,M,108255,Sales Manager,AU,11AUG1969,06/01/1989
120103,Wilson,Dawes,M,87975,Sales Manager,AU,22JAN1949,01/01/1974
120121,Irenie,Elvish,F,26600,Sales Rep. II,AU,02AUG1944,01/01/1974
120122,Christina,Ngan,F,27475,Sales Rep. II,AU,27JUL1954,07/01/1978
120123,Kimiko,Hotstone,F,26190,Sales Rep. I,AU,28SEP1964,10/01/1985
120124,Lucian,Daymond,M,26480,Sales Rep. I,AU,13MAY1959,03/01/1979
120125,Fong,Hofmeister,M,32040,Sales Rep. IV,AU,06DEC1954,03/01/1979
120126,Satyakam,Denny,M,26780,Sales Rep. II,AU,20SEP1988,08/01/2006
120127,Sharryn,Clarkson,F,28100,Sales Rep. II,AU,04JAN1979,11/01/1998
```

Delimiter

A space (blank) is the default delimiter.

The *DLM= option* can be added to the INFILE statement to specify an alternate delimiter.

```
DATA output-SAS-data-set;
      INFILE 'raw-data-file-name' DLM='delimiter';
      INPUT specifications;
      <additional SAS statements>
RUN;
```

Standard and Nonstandard Data

- *Standard data* is data that SAS can read without any special instructions.

  Examples of standard numeric data:

  58   -23   67.23   00.99   5.67E5   1.2E-2

- *Nonstandard data* is any data that SAS cannot read without a special instruction.

  Examples of nonstandard numeric data:

  5,823   (23) $67.23   01/12/1999   12MAY2006

List Input for Standard Data

List input specification:

```
INPUT variable <$>;
```

- Variables must be specified in the order that they appear in the raw data file, left to right.
- $ indicates to store a variable value as a character value rather than as a numeric value.
- The default length for character and numeric variables is eight bytes.

List Input for Standard Data

Partial **sales.csv**

```
120102,Tom,Zhou,M,108255,Sales Manager,AU,11AUG1969,06/01/1989
120103,Wilson,Dawes,M,87975,Sales Manager,AU,22JAN1949,01/01/1974
120121,Irenie,Elvish,F,26600,Sales Rep. II,AU,02AUG1944,01/01/1974
120122,Christina,Ngan,F,27475,Sales Rep. II,AU,27JUL1954,07/01/1978
120123,Kimiko,Hotstone,F,26190,Sales Rep. I,AU,28SEP1964,10/01/1985
120124,Lucian,Daymond,M,26480,Sales Rep. I,AU,13MAY1959,03/01/1979
120125,Fong,Hofmeister,M,32040,Sales Rep. IV,AU,06DEC1954,03/01/1979
120126,Satyakam,Denny,M,26780,Sales Rep. II,AU,20SEP1988,08/01/2006
120127,Sharryn,Clarkson,F,28100,Sales Rep. II,AU,04JAN1979,11/01/1998
```

```
input Employee_ID First_Name $ Last_Name $
      Gender $ Salary Job_Title $ Country $;
```

Business Scenario

Create a temporary SAS data set named **`Work.subset3`** from the delimited raw data file named **`sales.csv`**.

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $;
run;
```

p107d01

Business Scenario

```
281  data work.subset3;
282     infile 'sales.csv' dlm=',';
283     input Employee_ID First_Name $ Last_Name $
284           Gender $ Salary Job_Title $ Country $;
285  run;

NOTE: The infile 'sales.csv' is:
      File Name=S:\Workshop\sales.csv,
      RECFM=V,LRECL=256

NOTE: 165 records were read from the infile 'sales.csv'.
      The minimum record length was 61.
      The maximum record length was 80.
NOTE: The data set WORK.SUBSET3 has 165 observations and 7 variables.
```

Business Scenario

```
proc print data=work.subset3;
run;
```

Partial PROC PRINT Output

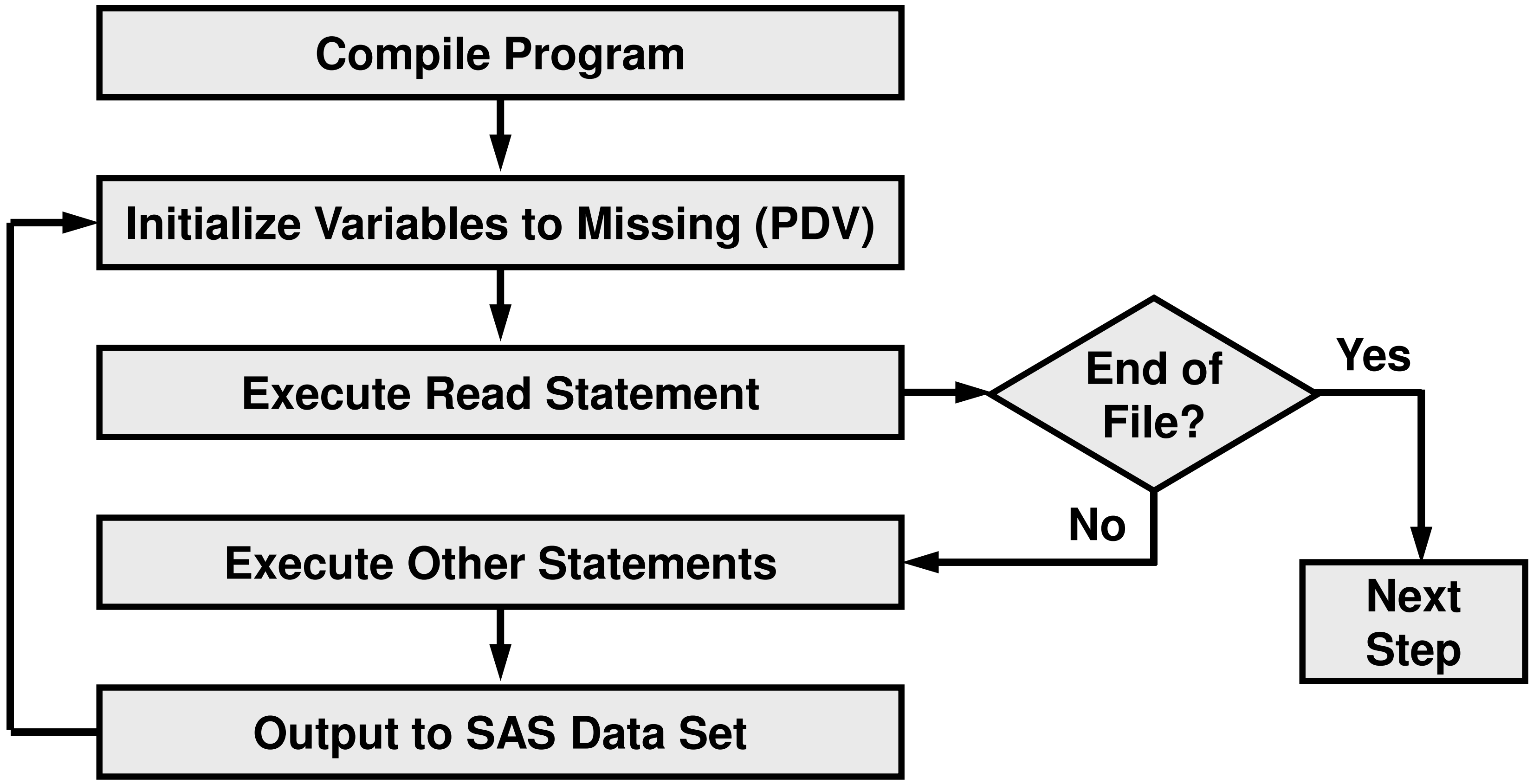
```
        Employee_     First_       Last_                                Job_
Obs        ID         Name         Name         Gender      Salary      Title       Country

  1      120102       Tom          Zhou            M        108255    Sales Ma        AU
  2      120103       Wilson       Dawes           M         87975    Sales Ma        AU
  3      120121       Irenie       Elvish          F         26600    Sales Re        AU
  4      120122       Christin     Ngan            F         27475    Sales Re        AU
  5      120123       Kimiko       Hotstone        F         26190    Sales Re        AU
  6      120124       Lucian       Daymond         M         26480    Sales Re        AU
  7      120125       Fong         Hofmeist        M         32040    Sales Re        AU
  8      120126       Satyakam     Denny           M         26780    Sales Re        AU
  9      120127       Sharryn      Clarkson        F         28100    Sales Re        AU
 10      120128       Monica       Kletschk        F         30890    Sales Re        AU
 11      120129       Alvin        Roebuck         M         30070    Sales Re        AU
 12      120130       Kevin        Lyon            M         26955    Sales Re        AU
```

p107d01

DATA Step Processing

The DATA step is processed in two phases:

- compilation
- execution



Compilation

During the compilation phase, SAS

- checks the syntax of the DATA step statements
- creates an input buffer to hold the current raw data file record that is being processed
- creates a program data vector (PDV) to hold the current SAS observation
- creates the descriptor portion of the output data set.

Compilation

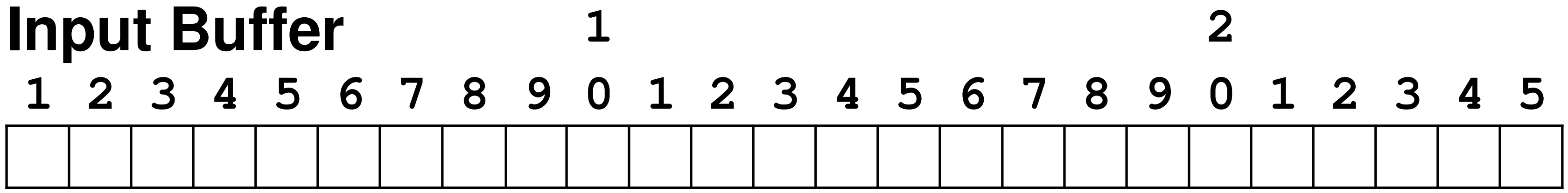
```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $;
run;
```

...

Compilation

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $;
run;
```

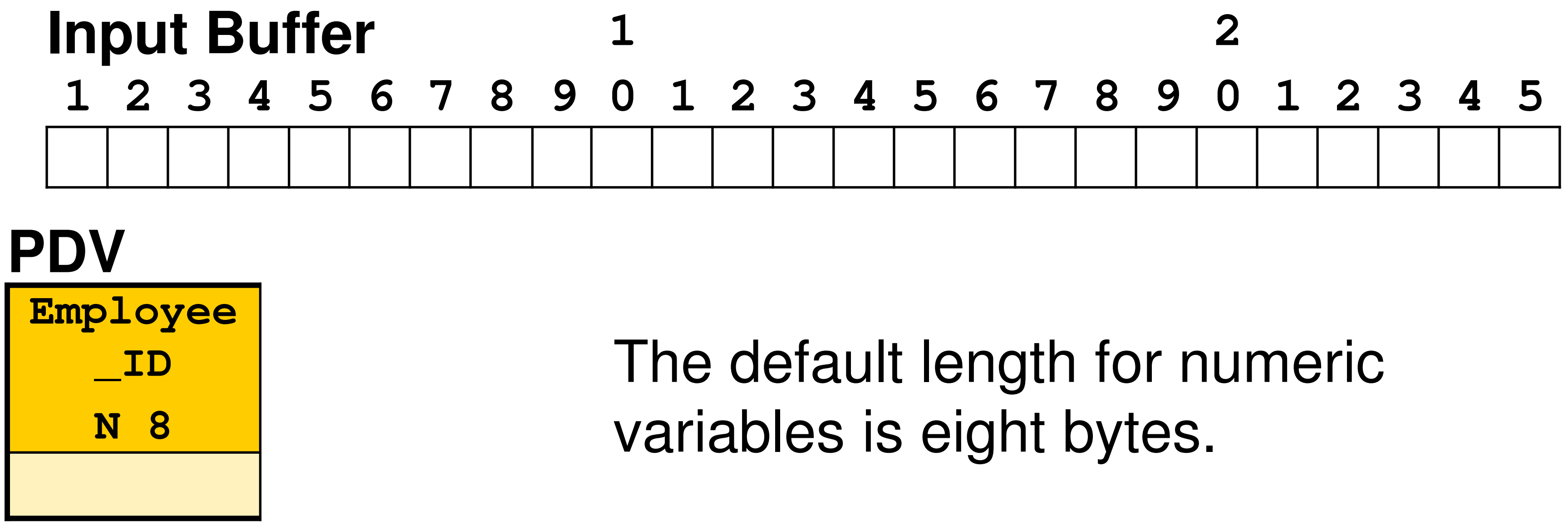
Input Buffer



...

Compilation

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $;
run;
```



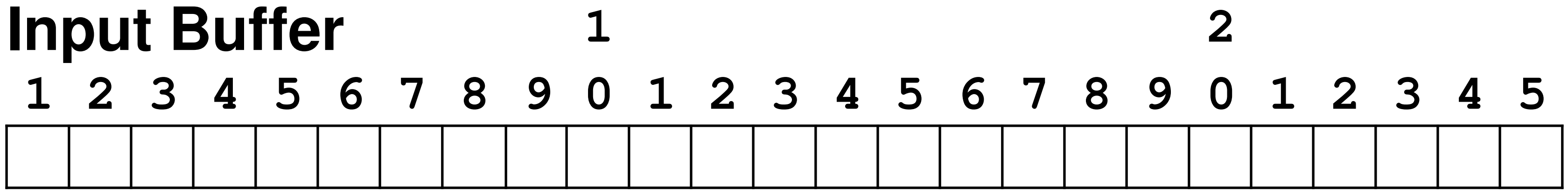
The default length for numeric variables is eight bytes.

...

Compilation

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $;
run;
```

Input Buffer



PDV

Employee _ID N 8	First_ Name $ 8

For list input, the default length for character variables is eight bytes.

...

Compilation

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $;
run;
```

Input Buffer

									1										2					
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

PDV

Employee _ID N 8	First_ Name $ 8	Last _Name $ 8	Gender $ 8	Salary N 8	Job_ Title $ 8	Country $ 8

...

Compilation

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $;
run;
```

Descriptor Portion `Work.subset3`

Employee _ID N 8	First_ Name $ 8	Last _Name $ 8	Gender $ 8	Salary N 8	Job_ Title $ 8	Country $ 8

...

7.02 Multiple Choice Poll

Which statement is true?

a. An input buffer is only created if you are reading data from a raw data file.

b. The PDV at compile time holds the variable name, type, byte size, and initial value.

c. The descriptor portion is the first item that is created at compile time.

Execution

Partial `sales.csv`

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

```
data work.subset3;
   infile 'sa                    ',';
   input Empl                   Name $
         Last                  $
         Salary Job_Title $
         Country $;
run;
```

Initialize PDV

Input Buffer

									1										2					
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

PDV

Employee_ID N 8	First_Name $ 8	Last_Name $ 8	Gender $ 8	Salary N 8	Job_Title $ 8	Country $ 8
.				.		

...

Execution

Partial sales.csv

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $
         Last_Name $ Gender $
         Salary Job_Title $
         Country $;
run;
```

Input Buffer

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

PDV

Employee_ID N 8	First_Name $ 8	Last_Name $ 8	Gender $ 8	Salary N 8	Job_Title $ 8	Country $ 8
.				.		

...

Execution

Partial `sales.csv`

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $
         Last_Name $ Gender $
         Salary Job_Title $
         Country $;
run;
```

Input Buffer

									1										2					
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1	2	0	1	0	2	,	T	o	m	,	Z	h	o	u	,	M	,	1	0	8	2	5	5	,

PDV

Employee _ID N 8	First_ Name $ 8	Last _Name $ 8	Gender $ 8	Salary N 8	Job_ Title $ 8	Country $ 8
.				.		

...

Execution

Partial sales.csv

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $
         Last_Name $ Gender $
         Salary Job_Title $
         Country $;
run;
```

Input Buffer

									1										2					
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1	2	0	1	0	2	,	T	o	m	,	Z	h	o	u	,	M	,	1	0	8	2	5	5	,

PDV

Employee _ID N 8	First_ Name $ 8	Last _Name $ 8	Gender $ 8	Salary N 8	Job_ Title $ 8	Country $ 8
120102	Tom	Zhou	M	108255	Sales Ma	AU

...

Execution

Partial sales.csv

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $
         Last_Name $ Gender $
         Salary Job_Title $
         Country $;
run;
```

Implicit OUTPUT;
Implicit RETURN;

Input Buffer

```
                  1
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
1 2 0 1 0 2 , T o m , Z h o u , M , 1 0 8 2 5 5 ,
```

PDV

Employee _ID N 8	First_ Name $ 8	Last _Name $ 8	Gender $ 8	Salary N 8	Job_ Title $ 8	Country $ 8
120102	Tom	Zhou	M	108255	Sales Ma	AU

...

Execution

Output SAS Data Set after First Iteration of DATA Step

Work.subset3

Employee_ID	First_Name	Last_Name	Gender	Salary	Job_Title	Country
120102	Tom	Zhou	M	108255	Sales Ma	AU

...

Execution

Partial sales.csv

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

```
data work.subset3;
   infile 'sa                     ;
   input Empl                     me $
         Last                     
         Salary Job_Title $
         Country $;
run;
```

Reinitialize PDV

Input Buffer

									1										2					
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1	2	0	1	0	2	,	T	o	m	,	Z	h	o	u	,	M	,	1	0	8	2	5	5	,

PDV

Employee_ID N 8	First_Name $ 8	Last_Name $ 8	Gender $ 8	Salary N 8	Job_Title $ 8	Country $ 8
.				.		

...

Execution

Partial `sales.csv`

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $
         Last_Name $ Gender $
         Salary Job_Title $
         Country $;
run;
```

Input Buffer

									1										2					
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1	2	0	1	0	2	,	T	o	m	,	Z	h	o	u	,	M	,	1	0	8	2	5	5	,

PDV

Employee_ID N 8	First_Name $ 8	Last_Name $ 8	Gender $ 8	Salary N 8	Job_Title $ 8	Country $ 8
.				.		

...

Execution

Partial `sales.csv`

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $
         Last_Name $ Gender $
         Salary Job_Title $
         Country $;
run;
```

Input Buffer

									1										2					
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1	2	0	1	0	3	,	W	i	l	s	o	n	,	D	a	w	e	s	,	M	,	8	7	9

PDV

Employee_ID N 8	First_Name $ 8	Last_Name $ 8	Gender $ 8	Salary N 8	Job_Title $ 8	Country $ 8
.				.		

...

Execution

Partial sales.csv

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $
         Last_Name $ Gender $
         Salary Job_Title $
         Country $;
run;
```

Input Buffer

									1										2					
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1	2	0	1	0	3	,	W	i	l	s	o	n	,	D	a	w	e	s	,	M	,	8	7	9

PDV

Employee _ID N 8	First_ Name $ 8	Last _Name $ 8	Gender $ 8	Salary N 8	Job_ Title $ 8	Country $ 8
120103	Wilson	Dawes	M	87975	Sales Ma	AU

...

Execution

Partial sales.csv

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

```
data work.subset3;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $
         Last_Name $ Gender $
         Salary Job_Title $
         Country $;
run;
```

Implicit OUTPUT;
Implicit RETURN;

Input Buffer

									1															
1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1	2	0	1	0	3	,	W	i	l	s	o	n	,	D	a	w	e	s	,	M	,	8	7	9

PDV

Employee _ID N 8	First_ Name $ 8	Last _Name $ 8	Gender $ 8	Salary N 8	Job_ Title $ 8	Country $ 8
120103	Wilson	Dawes	M	87975	Sales Ma	AU

...

Execution

Output SAS Data Set after Second Iteration of DATA Step

Work.subset3

Employee _ID	First_ Name	Last _Name	Gender	Salary	Job_ Title	Country
120102	Tom	Zhou	M	108255	Sales Ma	AU
120103	Wilson	Dawes	M	87975	Sales Ma	AU

...

Execution

Partial `sales.csv`

```
120102,Tom,Zhou, ...
120103,Wilson,Dawes, ...
120121,Irenie,Elvish, ...
120122,Christina,Ngan, ...
120123,Kimiko,Hotstone, ...
120124,Lucian,Daymond, ...
120125,Fong,Hofmeister, ...
```

Continue until EOF

```
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $
         Last_Name $ Gender $
         Salary Job_Title $
         Country $;
run;
```

Input Buffer

```
                         1                             2
 1  2  3  4  5  6  7  8  9  0  1  2  3  4  5  6  7  8  9  0  1  2  3  4  5
|1 |2 |0 |1 |0 |3 |, |W |i |l |s |o |n |, |D |a |w |e |s |, |M |, |8 |7 |9 |
```

PDV

Employee _ID N 8	First_ Name $ 8	Last _Name $ 8	Gender $ 8	Salary N 8	Job_ Title $ 8	Country $ 8
120103	Wilson	Dawes	M	87975	Sales Ma	AU

7.03 Multiple Choice Poll

Which statement is true?

a. Data is read directly from the raw data file to the PDV.
b. At the bottom of the DATA step, the contents of the PDV are output to the output SAS data set.
c. When SAS returns to the top of the DATA step, any variable coming from a SAS data set is set to missing.

The LENGTH Statement

The *LENGTH statement* defines the length of a variable explicitly.

General form of the LENGTH statement:

LENGTH *variable(s)* $ *length*;

Example:

```
length First_Name Last_Name $ 12
       Gender $ 1;
```

Business Scenario

Create a temporary SAS data set named **`Work.subset3`** from the delimited raw data file named **`sales.csv`**.

```
data work.subset3;
   length First_Name $ 12 Last_Name $ 18
          Gender $ 1 Job_Title $ 25
          Country $ 2;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $;
run;
```

p107d02

Business Scenario

```
proc print data=work.subset3;
run;
```

Partial PROC PRINT Output

```
         First_                                                                  Employee_
Obs      Name          Last_Name      Gender       Job_Title          Country       ID          Salary

  1      Tom           Zhou              M       Sales Manager          AU        120102        108255
  2      Wilson        Dawes             M       Sales Manager          AU        120103         87975
  3      Irenie        Elvish            F       Sales Rep. II          AU        120121         26600
  4      Christina     Ngan              F       Sales Rep. II          AU        120122         27475
  5      Kimiko        Hotstone          F       Sales Rep. I           AU        120123         26190
  6      Lucian        Daymond           M       Sales Rep. I           AU        120124         26480
  7      Fong          Hofmeister        M       Sales Rep. IV          AU        120125         32040
  8      Satyakam      Denny             M       Sales Rep. II          AU        120126         26780
  9      Sharryn       Clarkson          F       Sales Rep. II          AU        120127         28100
 10      Monica        Kletschkus        F       Sales Rep. IV          AU        120128         30890
 11      Alvin         Roebuck           M       Sales Rep. III         AU        120129         30070
 12      Kevin         Lyon              M       Sales Rep. I           AU        120130         26955
```

p107d02

Compilation

```
data work.subset3;
   length First_Name $ 12 Last_Name $ 18
          Gender $ 1 Job_Title $ 25
          Country $ 2;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $;
run;
```

PDV

First _Name $ 12	Last _Name $ 18	Gender $ 1	Job_Title $ 25	Country $ 2

...

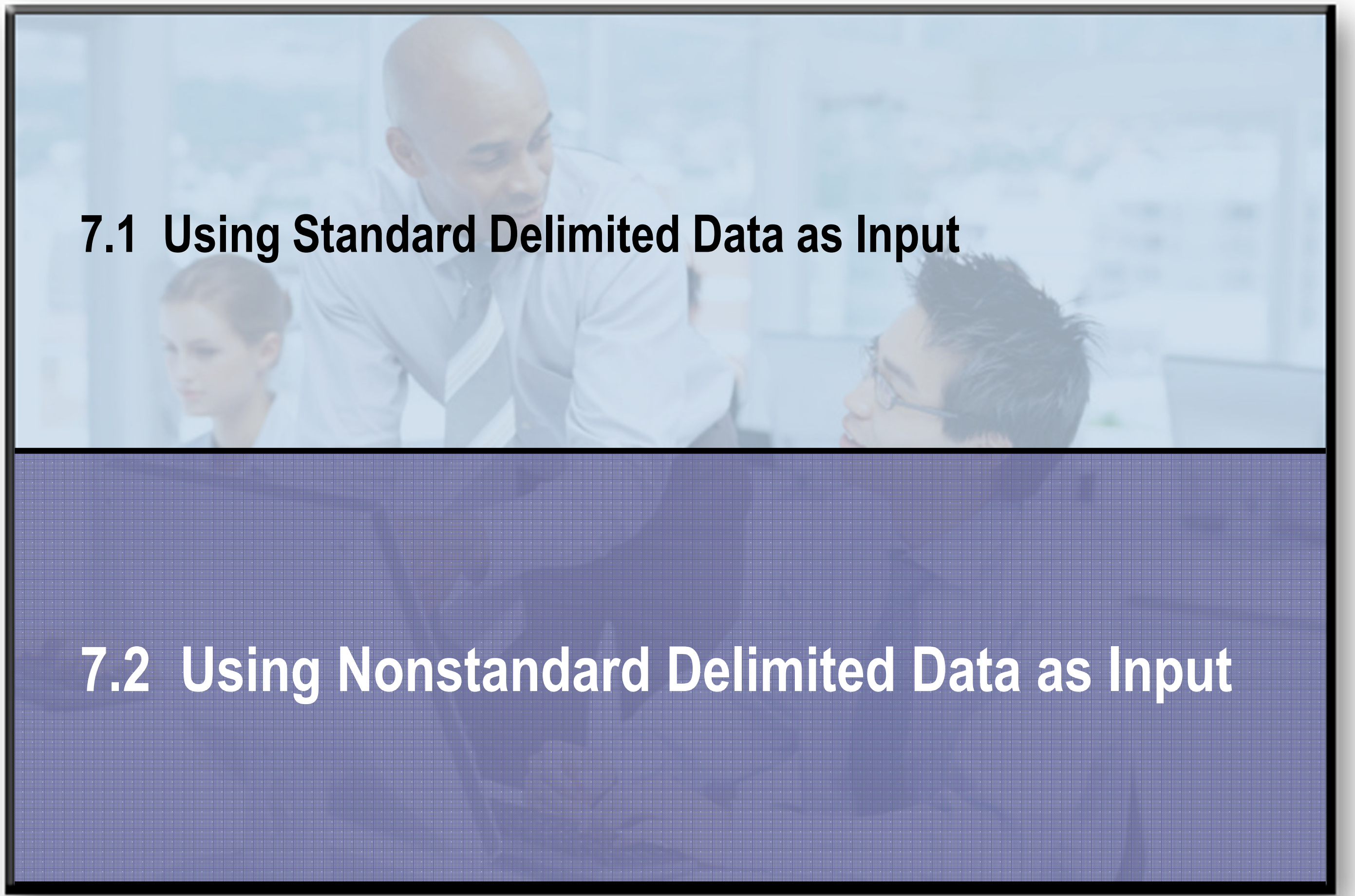
Compilation

```
data work.subset3;
   length First_Name $ 12 Last_Name $ 18
          Gender $ 1 Job_Title $ 25
          Country $ 2;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $;
run;
```

PDV

First _Name $ 12	Last _Name $ 18	Gender $ 1	Job_Title $ 25	Country $ 2	Employee _ID N 8	Salary N 8

Chapter 7: Reading Delimited Raw Data Files



Objectives

- Use informats to read nonstandard data.
- Add additional SAS statements to perform further processing in the DATA step.
- Use the DSD option with list input to read consecutive delimiters as missing values.
- Use the MISSOVER option to recognize missing values at the end of a record (Self-Study).

Standard and Nonstandard Data

- *Standard data* is data that SAS can read without any special instructions.

  Examples of standard numeric data:

  58   -23   67.23   00.99   5.67E5   1.2E-2

- ➡ *Nonstandard data* is any data that SAS cannot read without a special instruction.

  Examples of nonstandard numeric data:

  5,823   (23) $67.23   01/12/1999   12MAY2006

List Input for Nonstandard Data

List input specification:

INPUT *variable* <$> *variable* < *:informat* >**;**

- The : format modifier enables you to use an informat to read nonstandard delimited data.
- An *informat* is an instruction that SAS uses to read data values into a variable.
- The width of the informat can be eliminated.
- For character variables, the width of the informat determines the variable length, if it has not been previously defined.

SAS Informats

SAS informats have the following form:

$\boxed{\text{<\$>}\textit{informat}\text{<}\textit{w}\text{>.<}\textit{d}\text{>}}$

$	indicates a character informat.
informat	names the SAS informat or user-defined informat.
w	specifies the number of columns to read in the input data.
.	is a required delimiter.
d	specifies an optional decimal scaling factor in the numeric informats.

SAS Informats

Selected SAS Informats:

Informat	Definition
$*w.*	reads standard character data.
w.d	reads standard numeric data.
COMMA*w.d* DOLLAR*w.d*	reads nonstandard numeric data and removes embedded commas, blanks, dollar signs, percent signs, and dashes.
COMMAX*w.d* DOLLARX*w.d*	reads nonstandard numeric data and removes embedded periods, blanks, dollar signs, percent signs, and dashes.
EUROX*w.d*	reads nonstandard numeric data and removes embedded characters in European currency.

SAS Informats

In list input, informats are used to convert nonstandard numeric data to SAS numeric values.

Informat	Raw Data Value	SAS Data Value
COMMA. DOLLAR.	`$12,345`	`12345`
COMMAX. DOLLARX.	`$12.345`	`12345`
EUROX.	`€12.345`	`12345`

SAS Informats

SAS uses date informats to read and convert dates to SAS date values.

Informat	Raw Data Value	SAS Data Value

MMDDYY.	`010160`
	`01/01/60`
	`01/01/1960`
		`0`

DDMMYY.	`311260`
	`31/12/60`
	`31/12/1960`
		`365`

DATE.	`31DEC59`
	`31DEC1959`
		`-1`

List Input for Nonstandard Data

Partial **`sales.csv`**

```
120102,Tom,Zhou,M,108255,Sales Manager,AU,11AUG1969,06/01/1989
120103,Wilson,Dawes,M,87975,Sales Manager,AU,22JAN1949,01/01/1974
120121,Irenie,Elvish,F,26600,Sales Rep. II,AU,02AUG1944,01/01/1974
120122,Christina,Ngan,F,27475,Sales Rep. II,AU,27JUL1954,07/01/1978
120123,Kimiko,Hotstone,F,26190,Sales Rep. I,AU,28SEP1964,10/01/1985
120124,Lucian,Daymond,M,26480,Sales Rep. I,AU,13MAY1959,03/01/1979
120125,Fong,Hofmeister,M,32040,Sales Rep. IV,AU,06DEC1954,03/01/1979
120126,Satyakam,Denny,M,26780,Sales Rep. II,AU,20SEP1988,08/01/2006
120127,Sharryn,Clarkson,F,28100,Sales Rep. II,AU,04JAN1979,11/01/1998
```

```
input Employee_ID First_Name $ Last_Name $
      Gender $ Salary Job_Title $ Country $
      Birth_Date :date.
      Hire_Date :mmddyy.;
```

7.04 Quiz

Which INPUT statement correctly uses list input to read the space-delimited raw data file?

Raw Data

```
Donny 5MAY2008 25 FL $43,132.50
Margaret 20FEB2008 43 NC 65,150
```

a.
```
input name $ hired date. age
      state $ salary comma.;
```

b.
```
input name $ hired :date. age
      state $ salary :comma.;
```

Business Scenario

Create a temporary SAS data set named **`Work.subset3`** from the delimited raw data file named **`sales.csv`**.

```
data work.subset3;
   length First_Name $ 12 Last_Name $ 18
          Gender $ 1 Job_Title $ 25
          Country $ 2;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $
         Birth_Date :date.
         Hire_Date :mmddyy.;
run;
```

p107d03

Business Scenario

```
proc print data=work.subset3;
run;
```

Partial PROC PRINT Output

```
       First_                                                           Employee_
 Obs   Name         Last_Name      Gender     Job_Title       Country       ID      Salary   Birth_   Hire_
                                                                                              Date     Date

   1   Tom          Zhou             M      Sales Manager       AU       120102   108255     3510   10744
   2   Wilson       Dawes            M      Sales Manager       AU       120103    87975    -3996    5114
   3   Irenie       Elvish           F      Sales Rep. II       AU       120121    26600    -5630    5114
   4   Christina    Ngan             F      Sales Rep. II       AU       120122    27475    -1984    6756
   5   Kimiko       Hotstone         F      Sales Rep. I        AU       120123    26190     1732    9405
   6   Lucian       Daymond          M      Sales Rep. I        AU       120124    26480     -233    6999
   7   Fong         Hofmeister       M      Sales Rep. IV       AU       120125    32040    -1852    6999
   8   Satyakam     Denny            M      Sales Rep. II       AU       120126    26780    10490   17014
   9   Sharryn      Clarkson         F      Sales Rep. II       AU       120127    28100     6943   14184
  10   Monica       Kletschkus       F      Sales Rep. IV       AU       120128    30890     9691   17106
  11   Alvin        Roebuck          M      Sales Rep. III      AU       120129    30070     1787    9405
  12   Kevin        Lyon             M      Sales Rep. I        AU       120130    26955     9114   16922
  13   Marinus      Surawski         M      Sales Rep. I        AU       120131    26910     7207   15706
  14   Fancine      Kaiser           F      Sales Rep. III      AU       120132    28525    -3923    6848
```

p107d03

Additional SAS Statements

Additional SAS statements can be added to perform further processing in the DATA step.

```
data work.subset3;
   length First_Name $ 12 Last_Name $ 18
          Gender $ 1 Job_Title $ 25
          Country $ 2;
   infile 'sales.csv' dlm=',';
   input Employee_ID First_Name $ Last_Name $
         Gender $ Salary Job_Title $ Country $
         Birth_Date :date.
         Hire_Date :mmddyy.;
   keep First_Name Last_Name Salary
        Job_Title Hire_Date;
   label Job_Title='Sales Title'
         Hire_Date='Date Hired';
   format Salary dollar12. Hire_Date monyy7.;
run;
```

p107d04

Additional SAS Statements

```
proc print data=work.subset3 label;
run;
```

Partial PROC PRINT Output

```
        First_                                                 Date
 Obs    Name         Last_Name        Sales Title        Salary    Hired

   1    Tom          Zhou            Sales Manager     $108,255   JUN1989
   2    Wilson       Dawes           Sales Manager      $87,975   JAN1974
   3    Irenie       Elvish          Sales Rep. II      $26,600   JAN1974
   4    Christina    Ngan            Sales Rep. II      $27,475   JUL1978
   5    Kimiko       Hotstone        Sales Rep. I       $26,190   OCT1985
   6    Lucian       Daymond         Sales Rep. I       $26,480   MAR1979
   7    Fong         Hofmeister      Sales Rep. IV      $32,040   MAR1979
   8    Satyakam     Denny           Sales Rep. II      $26,780   AUG2006
   9    Sharryn      Clarkson        Sales Rep. II      $28,100   NOV1998
  10    Monica       Kletschkus      Sales Rep. IV      $30,890   NOV2006
```

p107d04

Additional SAS Statements

- The WHERE statement is used to obtain a subset of observations from an input data set.
- The WHERE statement cannot be used to select records from a raw data file.

The subsetting IF can subset data that is in the PDV.

Missing Values in the Middle of the Record

Each record in **`phone2.csv`** has a contact name, phone number, and a mobile number. The phone number is missing from some of the records.

Missing data is indicated by two consecutive delimiters.

`phone2.csv`

```
          1    1    2    2    3    3    4    4
1---5----0----5----0----5----0----5----0----5
James Kvarniq,(704) 293-8126,(701) 281-8923
Sandrina Stephano,,(919) 271-4592
Cornelia Krahl,(212) 891-3241,(212) 233-5413
Karen Ballinger,,(714) 644-9090
Elke Wallstab,(910) 763-5561,(910) 545-3421
```

7.05 Quiz

- Open and submit **p107a01**.
- Examine the SAS log.
- How many input records were read and how many observations were created?

```
data contacts;
   length Name $ 20 Phone Mobile $ 14;
   infile 'phone2.csv' dlm=',';
   input Name $ Phone $ Mobile $;
run;

proc print data=contacts noobs;
run;
```

Unexpected Results

The missing phone numbers caused unexpected results in the output.

PROC PRINT Output

```
       Name                   Phone              Mobile

   James Kvarniq         (704) 293-8126     (701) 281-8923
   Sandrina Stephano     (919) 871-7830     Cornelia Krahl
   Karen Ballinger       (714) 344-4321     Elke Wallstab
```

Partial SAS Log

```
NOTE: 5 records were read from the infile 'phone2.csv'.
      The minimum record length was 31.
      The maximum record length was 44.
NOTE: SAS went to a new line when INPUT statement reached
past the end of a line.
NOTE: The data set WORK.CONTACTS has 3 observations and 3
variables.
```

Consecutive Delimiters in List Input

By default, list input treats two or more consecutive delimiters as a single delimiter and not treated as a missing value.

The two consecutive commas are not being read as a missing value.

phone2.csv

```
         1    1    2    2    3    3    4    4
1---5----0----5----0----5----0----5----0----5
James Kvarniq,(704) 293-8126,(701) 281-8923
Sandrina Stephano,,(919) 271-4592
Cornelia Krahl,(212) 891-3241,(212) 233-5413
Karen Ballinger,,(714) 644-9090
Elke Wallstab,(910) 763-5561,(910) 545-3421
```

The DSD Option

The DSD option for the INFILE statement

- sets the default delimiter to a comma
- treats consecutive delimiters as missing values
- enables SAS to read values with embedded delimiters if the value is surrounded by quotation marks.

General form of a DSD option in an INFILE statement:

INFILE '*raw-data-file-name*' DSD**;**

Using the DSD Option

Adding the DSD option will correctly read the **`phone2.csv`** data file.

```
data contacts;
   length Name $ 20 Phone Mobile $ 14;
   infile 'phone2.csv' dsd;
   input Name $ Phone $ Mobile $;
run;

proc print data=contacts noobs;
run;
```

✎ The DLM=',' option is no longer needed in the INFILE statement because the DSD option sets the default delimiter to a comma.

p107d05

Results

Adding the DSD option gives the expected results.

PROC PRINT Output

```
      Name                       Phone                Mobile

James Kvarniq              (704) 293-8126     (701) 281-8923
Sandrina Stephano                             (919) 271-4592
Cornelia Krahl             (212) 891-3241     (212) 233-5413
Karen Ballinger                               (714) 644-9090
Elke Wallstab              (910) 763-5561     (910) 545-3421
```

Partial SAS Log

```
NOTE: 5 records were read from the infile 'phone2.csv'.
      The minimum record length was 31.
      The maximum record length was 44.
NOTE: The data set WORK.CONTACTS has 5 observations and
3 variables.
```

Missing Values at the End of a Record (Self-Study)

The data values in **phone.csv** are separated by commas. Each record has a contact name, and then a phone number, and finally a mobile number.

phone.csv

The mobile number and comma delimiter are missing from some of the lines of data.

```
          1                                            4
1---5----0----                                       ---5
James Kvarniq,(704) 293-8126,(701) 281-8923
Sandrina Stephano,(919) 871-7830
Cornelia Krahl,(212) 891-3241,(212) 233-5413
Karen Ballinger,(714) 344-4321
Elke Wallstab,(910) 763-5561,(910) 545-3421
```

7.06 Quiz (Self-Study)

Open and submit **p107a02**. Examine the SAS log. How many input records were read and how many observations were created?

```
data contacts;
   length Name $ 20 Phone Mobile $ 14;
   infile 'phone.csv' dsd;
   input Name $ Phone $ Mobile $;
run;

proc print data=contacts noobs;
run;
```

Unexpected Results (Self-Study)

The missing mobile phone numbers caused unexpected results in the output.

PROC PRINT Output

```
       Name                    Phone              Mobile

  James Kvarniq          (704) 293-8126     (701) 281-8923
  Sandrina Stephano      (919) 871-7830     Cornelia Krahl
  Karen Ballinger        (714) 344-4321     Elke Wallstab
```

Partial SAS Log

```
NOTE: 5 records were read from the infile 'phone.csv'.
      The minimum record length was 31.
      The maximum record length was 44.
NOTE: SAS went to a new line when INPUT statement
reached past the end of a line.
NOTE: The data set WORK.CONTACTS has 3 observations and
3 variables.
```

Missing Values at the End of a Record (Self-Study)

By default, when there is missing data at the end of a row, SAS does the following:

- loads the next record to finish the observation
- writes a note to the log

The MISSOVER Option (Self-Study)

The MISSOVER option prevents SAS from loading a new record when the end of the current record is reached.

General form of an INFILE statement with a MISSOVER option:

INFILE '*raw-data-file-name*' MISSOVER**;**

If SAS reaches the end of the row without finding values for all fields, variables without values are set to missing.

7.07 Quiz (Self-Study)

Open **p107a03** and add the MISSOVER option to the INFILE statement. Submit the program and examine the SAS log. How many input records were read and how many observations were created?

```
data contacts;
   length Name $ 20 Phone Mobile $ 14;
   infile 'phone.csv' dsd;
   input Name $ Phone $ Mobile $;
run;

proc print data=contacts noobs;
run;
```

Results (Self-Study)

Adding the MISSOVER option gives the expected results.

PROC PRINT Output

```
      Name                   Phone              Mobile

  James Kvarniq          (704) 293-8126    (701) 281-8923
  Sandrina Stephano      (919) 871-7830
  Cornelia Krahl         (212) 891-3241    (212) 233-5413
  Karen Ballinger        (714) 344-4321
  Elke Wallstab          (910) 763-5561    (910) 545-3421
```

Partial SAS Log

```
NOTE: 5 records were read from the infile 'phone.csv'.
      The minimum record length was 31.
      The maximum record length was 44.
NOTE: The data set WORK.CONTACTS has 5 observations and
3 variables.
```

Chapter Review

1. What statement identifies the physical filename of the raw data file to read?

2. What statement describes the arrangement of values in the raw data file?

3. What is the default delimiter when the DLM= option is used?

4. What are the two phases of DATA step processing?

5. What is a program data vector (PDV)?

continued...

Chapter Review

6. Why would you use a LENGTH statement?

7. What is an instruction that SAS uses to read data values into a variable?

8. When would you use a : modifier?