### SECTION 1 DATA Step Programming

**CHAPTERS**

1. Inputting Raw Data ............................................. 3
2. Reading and Writing from External Files .................... 15
3. Using Logical Structures: IF-THEN/ELSE, WHERE, and SELECT .................................................. 17
4. Combining SAS Data Sets ........................................ 21
5. Using Numerical Functions ..................................... 31
6. Using Character Functions ..................................... 39
7. Working with Dates ............................................. 45
8. Using Arrays .................................................... 49
9. Looking Back and Ahead Across Observations ............... 57
10. Working with Longitudinal Data: Multiple Observations per Subject ................................................. 63
11. Writing Simple DATA Step Reports ............................ 69
CHAPTER 1  Inputting Raw Data

PROBLEMS

1 Reading Data Values Separated by Spaces ............................................. 4
2 Reading Data Values, Including Missing Values, Separated by Spaces ........... 4
3 Reading Data Values Separated by Spaces with a Character Value Greater than 8 Bytes Long .......................................................... 5
4 Reading Data Values Where Missing Values Are Represented by Periods except at the End of Short Data Lines ............................................. 5
5 Reading Data Values Separated by Commas Where Missing Values Are Represented by Two Adjacent Commas and Some of the Character Values Are Placed in Double Quotes ......................................................... 6
6 Reading Data Values Using Starting and Ending Column Numbers ............. 7
7 Reading Data Values with Missing Character and Numeric Values Using Starting and Ending Column Numbers ..................................... 7
8 Reading Data Values Using Pointers and Informat ................................. 8
9 Reading Mixed Record Types in One DATA Step ...................................... 8
10 Creating a Single Observation from More than One Line of Raw Data ....... 10
11 Creating More than One Observation from a Single Data Line ................ 11
12 Creating More than One Observation from a Single Data Line (Two Variables) ... 11
13 Creating More than One Observation from a Single Data Line (Two Variables: One Character, One Numeric) ........................................... 12
14 Reading “Free-form” Data, Creating More than One Observation from a Single Data Line ................................................................. 12
15 Using Variable Lists and Informat Lists to Make the INPUT Statement More Compact ................................................................. 13
16 Using Variable Lists, Informat Lists, and Relative Column Pointers to Read Data Values ................................................................. 13

INTRODUCTION

This chapter contains problems on reading raw data with an INPUT statement. All the problems read “instream” data, that is, data that you submit as part of your program. However, all the techniques used here can also be applied to reading data from external files (discussed in Chapter 2). For those “old-timers” who like to use a CARDS statement, you may still do so. The newer term DATALINES is equivalent and is used throughout this book.
PROBLEM 1
Reading Data Values Separated by Spaces

Tools
- INPUT statement
- PROC PRINT

Data
You have collected some data on a group of students. The data values, separated by one or more spaces, represent the variables F_Name (first name), ID, Gender, GPA (grade point average), Height and Weight.

Hector 123 M 3.5 59 155
Nancy 328 F 3.7 52 99
Edward 747 M 2.4 62 205
Michelle 778 F 3.0 54 115
Sampson 289 M 3.5 60 180

Directions
Create a temporary SAS data set called CLASS from these lines of data. Include the lines of data “instream” in the program. Display the contents of this data set.

Notes
1. All the first names are 8 characters or less.
2. ID is to be stored as character data.

PROBLEM 2
Reading Data Values, Including Missing Values, Separated by Spaces

Tools
- INPUT statement

Data
Raw data similar to Problem 1 with some missing values

Hector 123 M 3.5 . 155
. 328 F 3.7 52 99
Edward . . . .
Michelle 778 F 3.0 54 .
Sampson 289 M 3.5 60 180
Directions
Repeat Problem 1 with the modified data. Note that there are several missing values (both for numeric and character variables) which are represented by periods.

PROBLEM 3
Reading Data Values Separated by Spaces with a Character Value Greater than 8 Bytes Long

Tools
INPUT statement
Colon format modifier (:) or LENGTH statement or INFORMAT statement

Data
Raw data similar to Problem 1 with one name greater than 8 characters in length

| Name    | Sex | Age | Height | Weight | Weight
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hector</td>
<td>M</td>
<td>3.5</td>
<td>M</td>
<td>59</td>
<td>155</td>
</tr>
<tr>
<td>Nancy</td>
<td>F</td>
<td>3.7</td>
<td>F</td>
<td>52</td>
<td>99</td>
</tr>
<tr>
<td>Edward</td>
<td>M</td>
<td>2.4</td>
<td>M</td>
<td>62</td>
<td>205</td>
</tr>
<tr>
<td>Michelle</td>
<td>F</td>
<td>3.0</td>
<td>F</td>
<td>54</td>
<td>115</td>
</tr>
<tr>
<td>Washington</td>
<td>M</td>
<td>3.5</td>
<td>M</td>
<td>60</td>
<td>180</td>
</tr>
</tbody>
</table>

Directions
Repeat Problem 1 with the new data. Note that one of the names is now longer than 8 characters long.

PROBLEM 4
Reading Data Values Where Missing Values Are Represented by Periods except at the End of Short Data Lines

Tools
INFILE statement
MISSOVER option
INPUT statement
Data
Raw data similar to Problem 1 with some missing values and short records

<table>
<thead>
<tr>
<th>Hector</th>
<th>123</th>
<th>M</th>
<th>3.5</th>
<th>.</th>
<th>155</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>328</td>
<td>F</td>
<td>3.7</td>
<td>52</td>
<td>99</td>
</tr>
<tr>
<td>Edward</td>
<td>747</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Michelle</td>
<td>778</td>
<td>F</td>
<td>3.0</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>Sampson</td>
<td>289</td>
<td>M</td>
<td>3.5</td>
<td>60</td>
<td>180</td>
</tr>
</tbody>
</table>

Directions
Repeat Problem 1 with the new data. Note that there are several missing values (both for numeric and character variables) and there are some short records.

PROBLEM 5
Reading Data Values Separated by Commas Where Missing Values Are Represented by Two Adjacent Commas and Some of the Character Values Are Placed in Double Quotes

Tools
INFILE statement
DSD option
INPUT statement

Data
Raw data similar to Problem 1 with comma delimiters

| Hector, 123, M, 3.5, ., 155 |
|---|---|---|---|---|---|
| ., 328, "F", 3.7, 52, 99 |
| 'Edward', 747, ., ., . |
| Michelle, 778, F, 3.0, 54, . |
| Sampson, 289, M, 3.5, 60, 180 |

Directions
Repeat Problem 1 using comma-delimited data. Note that some of the character values are enclosed in double quotes and that consecutive commas (without any space between) represent missing values.
PROBLEM 6
Reading Data Values Using Starting and Ending Column Numbers

Tools
INPUT statement, with column numbers

Data
Raw data similar to Problem 1 lined up in columns

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>12345678901234567890123456789012345678901234567890123456789012345678901</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hector</td>
<td>123</td>
<td>M</td>
<td>3.5</td>
<td>59</td>
</tr>
<tr>
<td>Nancy</td>
<td>328</td>
<td>F</td>
<td>3.7</td>
<td>52</td>
</tr>
<tr>
<td>Edward</td>
<td>747</td>
<td>M</td>
<td>2.4</td>
<td>62</td>
</tr>
<tr>
<td>Michelle</td>
<td>778</td>
<td>F</td>
<td>3.0</td>
<td>54</td>
</tr>
<tr>
<td>Sampson</td>
<td>289</td>
<td>M</td>
<td>3.5</td>
<td>60</td>
</tr>
</tbody>
</table>

Directions
Use the same data as in Problem 1, but this time write the INPUT statement using starting and ending column numbers. The data are listed with column numbers displayed.

PROBLEM 7
Reading Data Values with Missing Character and Numeric Values Using Starting and Ending Column Numbers

Tools
INFILE statement
PAD option
INPUT statement, with column numbers

Data
Raw data similar to Problem 1 lined up in columns with some missing values and short records

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>12345678901234567890123456789012345678901234567890123456789012345678901</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hector</td>
<td>123</td>
<td>M</td>
<td>3.5</td>
<td>155</td>
</tr>
<tr>
<td>Nancy</td>
<td>328</td>
<td>F</td>
<td>3.7</td>
<td>52</td>
</tr>
<tr>
<td>Edward</td>
<td>747</td>
<td></td>
<td>2.4</td>
<td>62</td>
</tr>
<tr>
<td>Michelle</td>
<td>778</td>
<td>F</td>
<td>3.0</td>
<td>54</td>
</tr>
<tr>
<td>Sampson</td>
<td>289</td>
<td>M</td>
<td>3.5</td>
<td>60</td>
</tr>
</tbody>
</table>
Directions
Repeat Problem 6 with the modified data. Note that there are now blanks representing both character and numeric missing values. Notice that lines 3 and 4 are short records that are not padded with blanks.

PROBLEM 8
Reading Data Values Using Pointers and Informats

Tools
INPUT statement, with column pointers (@) and informats

Data
The same raw data as in Problem 6

Directions
Instead of using starting and ending column numbers for the data values (as in Problem 6), use column pointers and informats to read the same data values and create a SAS data set called CLASS. Use the data description that follows to help you write the INPUT statement:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Starting Column</th>
<th>Length</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>F_NAME</td>
<td>1</td>
<td>8</td>
<td>Char</td>
</tr>
<tr>
<td>ID</td>
<td>13</td>
<td>3</td>
<td>Char</td>
</tr>
<tr>
<td>GENDER</td>
<td>22</td>
<td>1</td>
<td>Char</td>
</tr>
<tr>
<td>GPA</td>
<td>31</td>
<td>3</td>
<td>Numeric</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>39</td>
<td>2</td>
<td>Numeric</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>49</td>
<td>3</td>
<td>Numeric</td>
</tr>
</tbody>
</table>

PROBLEM 9
Reading Mixed Record Types in One DATA Step

Tools
INPUT statement, single trailing at sign (@)
Data
Survey data as shown

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>12345678901234567890</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>001MRY 3 1994</td>
<td></td>
</tr>
<tr>
<td>00923FDY 1 1995</td>
<td></td>
</tr>
<tr>
<td>012FDN 2 1994</td>
<td></td>
</tr>
<tr>
<td>00518MRN 2 1995</td>
<td></td>
</tr>
<tr>
<td>003MDY 4 1994</td>
<td></td>
</tr>
</tbody>
</table>

Directions
You have data from a survey administered in 1994 and 1995. In 1995, it was decided to record the age of each person (which was not done in 1994). Age information was not added at the end of the raw data. Instead, it was placed right after the subject ID, creating two completely different data layouts for the two years. Fortunately, the year the survey was administered was also entered into the data file. Create a temporary SAS data set called SURVEY which correctly reads these mixed records. The data description follows:

### 1994 Data Description

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Starting Column</th>
<th>Ending Column</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Subject ID</td>
<td>1</td>
<td>3</td>
<td>Char</td>
</tr>
<tr>
<td>GENDER</td>
<td>Subject Gender</td>
<td>4</td>
<td>4</td>
<td>Char</td>
</tr>
<tr>
<td>PARTY</td>
<td>Political Party</td>
<td>5</td>
<td>5</td>
<td>Char</td>
</tr>
<tr>
<td>VOTE</td>
<td>Did you vote in the last election?</td>
<td>6</td>
<td>6</td>
<td>Char</td>
</tr>
<tr>
<td>NUM_TV</td>
<td>Number of TV's</td>
<td>7</td>
<td>8</td>
<td>Numeric</td>
</tr>
<tr>
<td>YEAR</td>
<td>Survey year</td>
<td>15</td>
<td>18</td>
<td>Char</td>
</tr>
</tbody>
</table>

### 1995 Data Description

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Starting Column</th>
<th>Ending Column</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Subject ID</td>
<td>1</td>
<td>3</td>
<td>Char</td>
</tr>
<tr>
<td>AGE</td>
<td>Subject's age</td>
<td>4</td>
<td>5</td>
<td>Numeric</td>
</tr>
<tr>
<td>GENDER</td>
<td>Subject Gender</td>
<td>6</td>
<td>6</td>
<td>Char</td>
</tr>
<tr>
<td>PARTY</td>
<td>Political Party</td>
<td>7</td>
<td>7</td>
<td>Char</td>
</tr>
<tr>
<td>VOTE</td>
<td>Did you vote in the last election?</td>
<td>8</td>
<td>8</td>
<td>Char</td>
</tr>
<tr>
<td>NUM_TV</td>
<td>Number of TV's</td>
<td>9</td>
<td>10</td>
<td>Numeric</td>
</tr>
<tr>
<td>YEAR</td>
<td>Survey year</td>
<td>15</td>
<td>18</td>
<td>Char</td>
</tr>
</tbody>
</table>
PROBLEM 10
Creating a Single Observation from More than One Line of Raw Data

Tools
INPUT statement, line and column pointers (# and @), informats

Data
Car survey data as follows

<table>
<thead>
<tr>
<th>Line</th>
<th>SUBJECT</th>
<th>DOB</th>
<th>STATE</th>
<th>ZIP_CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>12310/21/46</td>
<td>NJ</td>
<td>1234567890</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>23711/01/55</td>
<td>NY</td>
<td>11518</td>
</tr>
</tbody>
</table>

Directions
A survey was conducted and the data were recorded using two lines of data for each subject. Use the data description and sample lines of data to create a SAS data set called SURVEY, with one observation per subject. The file description follows:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Starting Column</th>
<th>Ending Column</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBJECT</td>
<td>Subject number</td>
<td>1</td>
<td>3</td>
<td>Char</td>
</tr>
<tr>
<td>DOB</td>
<td>Date of Birth</td>
<td>4</td>
<td>11</td>
<td>MM/DD/YY</td>
</tr>
<tr>
<td>STATE</td>
<td>State where living</td>
<td>25</td>
<td>26</td>
<td>Char</td>
</tr>
<tr>
<td>ZIP_CODE</td>
<td>Zip Code</td>
<td>40</td>
<td>44</td>
<td>Character</td>
</tr>
<tr>
<td>NUMBER</td>
<td>Number of cars</td>
<td>5</td>
<td>5</td>
<td>Numeric</td>
</tr>
<tr>
<td>CAR1</td>
<td>Make of Car 1</td>
<td>11</td>
<td>20</td>
<td>Char</td>
</tr>
<tr>
<td>CAR2</td>
<td>Make of Car 2</td>
<td>21</td>
<td>30</td>
<td>Char</td>
</tr>
</tbody>
</table>

Notes
1. You may use either pointers and informats for all of your variables, or you may use column specifications for all of your variables except for DOB for which you will need either a pointer and an informat or an INFORMAT statement.

2. For this problem, do not read the SUBJECT value in line 2 of the data. In a more sophisticated program, you might want to read this as a different variable and check that it is the same as the subject number in line 1.
PROBLEM 11
Creating More than One Observation from a Single Data Line

Tools
- INPUT statement, double trailing at sign (@@)

Data
Twenty temperatures as shown

\[
\begin{align*}
21 & \quad 23 & \quad 29 & \quad 33 & \quad 19 & \quad 28 \\
33 & \quad 39 & \quad 43 & \quad 44 & \quad 28 & \quad 21 & \quad 24 \\
37 & \quad 32 & \quad 31 & \quad 33 & \quad 29 \\
\end{align*}
\]

Directions
A researcher collected 20 temperatures (one for each day) and entered them on several lines as shown in the data section.

As you can see, there are several temperatures per line (not always the same number) and there are one or more spaces between each number. Create a temporary SAS data set called TEMPER from these data. Print out the contents of this data set.

PROBLEM 12
Creating More than One Observation from a Single Data Line (Two Variables)

Tools
- INPUT statement, double trailing at sign (@@)

Data
Twenty pairs of days of the month and temperatures as shown

\[
\begin{align*}
5 & \quad 21 & \quad 6 & \quad 23 & \quad 7 & \quad 29 & \quad 8 & \quad 33 & \quad 9 & \quad 19 & \quad 10 & \quad 28 \\
11 & \quad 33 & \quad 12 & \quad 39 & \quad 13 & \quad 43 & \quad 14 & \quad 44 & \quad 15 & \quad 28 \\
20 & \quad 37 & \quad 21 & \quad 32 & \quad 22 & \quad 31 & \quad 23 & \quad 33 & \quad 24 & \quad 29 \\
\end{align*}
\]

Directions
This time the researcher recorded the day of the month along with each of the temperatures and entered them in pairs (day temperature). Create a temporary SAS data set called TEMP_DAY containing two variables (DAY and TEMP), using these data. List the contents of this data set.
PROBLEM 13
Creating More than One Observation from a Single Data Line (Two Variables: One Character, One Numeric)

Tools
INPUT statement, double trailing at sign (@@)

Data
Rat data (GROUP and WEIGHT) in pairs with a varying number of pairs per line as shown
A 34  B 58  A 28  C 55
C 56  A 27  B 52  C 58  A 21  B 62

Directions
A researcher treated three groups of rats (Groups A, B, and C) and recorded the weight of each rat after one week. The data were arranged with each GROUP and WEIGHT in pairs, with a varying number of pairs on each line.

Write a SAS DATA step to read these data and create a temporary data set called RATS. Print out the contents of this data set.

PROBLEM 14
Reading “Free-form” Data, Creating More than One Observation from a Single Data Line

Tools
RETAIN statement
INPUT statement, double trailing at sign (@@)

Data
Rat data similar to Problem 13
A 34  28  B 50  52  62  C 55  A 27  21  C 56  58

Directions
The same data values as found in Problem 13 were entered differently. This time, the researcher entered a GROUP value (A, B, or C) followed by one or more weights as shown.

Write a SAS DATA step to read these data. Print out the contents of the data set. Again, call the data set RATS.
Hint
Look carefully at the tools needed for this problem.

PROBLEM 15
Using Variable Lists and Informat Lists to Make the INPUT Statement More Compact

Tools
INPUT statement, variable lists, and informat lists

Data
Instream data lines in the program to be rewritten

Directions
Rewrite the DATA step below, substituting an INPUT statement which uses a variable list and an informat list to make it more compact. Print out the contents of the data set.

```
DATA VARLIST;
  INPUT @1 Q1 2. @3 Q2 2. @5 Q3 2. @7 Q4 2. @9 Q5 2. @15 DATE1 MMDDYY8.
  @23 DATE2 MMDDYY8.
  @31 DATE3 MMDDYY8.
  @50 X1 $1. @51 X2 $1. @52 X3 $1.
  @53 Y1 $1. @54 Y2 $1. @55 Y3 $1.;
DATALINES;
1122334455    10/21/4611/13/4206/05/48           123456
9672347656    01/01/9501/02/9501/03/95           987654
;```

PROBLEM 16
Using Variable Lists, Informat Lists, and Relative Column Pointers to Read Data Values

Tools
INPUT statement, variable lists, and informat lists; relative column pointers (+)

Data
Instream data lines in the program to be rewritten

Directions
Rewrite the DATA step below, substituting an INPUT statement which uses a variable list, an informat list, and relative column pointers (+). Print out the contents of the data set.

```
DATA VARLIST;
  INPUT @1 Q1 2. @3 Q2 2. @5 Q3 2. @7 Q4 2. @9 Q5 2. @15 DATE1 MMDDYY8.
  @23 DATE2 MMDDYY8.
  @31 DATE3 MMDDYY8.
  @50 X1 $1. @51 X2 $1. @52 X3 $1.
  @53 Y1 $1. @54 Y2 $1. @55 Y3 $1.;
DATALINES;
1122334455    10/21/4611/13/4206/05/48           123456
9672347656    01/01/9501/02/9501/03/95           987654
;```
Hint
Read all the X’s first, then the Y’s, and finally, the Z’s.

DATA POINTER;
   INPUT @1  X1 2.
       @3  Y1 2.
       @5  Z1 $3.
       @8  X2 2.
       @10 Y2 2.
       @12 Z2 $3.
       @15 X3 2.
       @17 Y3 2.
       @19 Z3 $3.;
DATA LINES;
   0102AAA0304BBB0506CCC
   2837ABC9676DEF8765GHI
;