Chapters 7, 17 SAS INFORMATS AND FORMATS

SAS INFORMATS

**Informats** are used to read data from external files, assigning **permanent** formats to variables. Typically, SAS informats are grouped into three categories: numeric, character, and date/time. For now, we study only numeric and character informats. The syntax of informats is as follows:

Numeric variable INFORMATw.d

Character variable $INFORMATw.

* The dollar sign indicates that it is a character type informat.
* INFORMAT refers to informat name (optional).
* The w refers to the width (the number of columns) of the variable (called a **field width**).
* The d is used for numeric type data to specifiy the number of digits to the right of the decimal place.
* Informats must contain a period (called a **period delimiter**) to differentiate informats from variable names.

SAS provides a large number of informats. A complete list may be found in SAS documentation. Some example of informats are

Numeric variable w.d COMMAw.d PERCENTw.d

Character variable $w. $CHARw.

* The numeric informat COMMAw.d instructs SAS to treat the variable as numeric and strip the values from all comma separators (and dollar signs, if there are any).
* The numeric informat PERCENTw.d instructs SAS to treat the variable with a percent sign (%) at the end as numeric and divide it by 100.
* The character informat $CHARw. instructs SAS to treat the variable as character and to store the variable with leading blanks, if they exist.

Example. Suppose the following data are stored in file ‘H:\bankdata.dat’.

ID Age Gender Income MortgageRate

355 45 F $3,578.80 3.375%

27 38 M $1,117.63 4.250%

192 32 F $2,133.12 5.500%

208 31 F $2,458.18 7.825%

11 57 M $1,995.50 4.875%

The SAS code

data bankdata;

infile ' C:\Users\mstudent\Desktop\bankdata.dat';

input @1 id $char3.

@5 age 2.0

@8 gender $1.

@10 income comma9.2

@20 mortgagerate percent6.

;

run;

proc print;

run;

produces the output

| **Obs** | **id** | **age** | **gender** | **income** | **mortgagerate** |
| --- | --- | --- | --- | --- | --- |
| **1** | 355 | 45 | F | 3578.80 | 0.03375 |
| **2** | 27 | 38 | M | 1117.63 | 0.04250 |
| **3** | 192 | 32 | F | 2133.12 | 0.05500 |
| **4** | 208 | 31 | F | 2458.18 | 0.07825 |
| **5** | 11 | 57 | M | 1995.50 | 0.04875 |
|  |  |  |  |  |  |

Exercise. Place the correct informats into the DATA step below. The real estate data contain information on house age (age), square footage (sqft), number of bedrooms (bedrms), whether house is remodeled (remodeled, yes or no), percent discount in price (discount), and listed discounted price (price). Verify that SAS reads the values properly.

data real\_estate;

input @1 age @4 sqft @10 bedrms @12 remodeled

@16 discount @20 price;

cards;

45 1,700 3 yes 10% $560,000

10 2,700 3 no 15% $670,500

12 3,450 5 no 25% $1,100,000

34 1,080 2 yes 15% $307,000

25 750 1 no 10% $195,000

;

proc print;

run;

INFORMAT STATEMENT

Informats may be placed in the INPUT statement, or, alternatively, the INFORMAT statement may be used. For example, bankdata data set introduced above may be inputted as

data bankdata;

infile ' C:\Users\mstudent\Desktop\bankdata.dat';

informat id $char3.

age 2.0

gender $1.

income comma9.2

mortgagerate percent6.

;

input @1 id

@5 age

@8 gender

@10 income

@20 mortgagerate

;

run;

* Note that the INFORMAT statement MUST precede the INPUT statement.

Exercise. In the exercise above, use the INFORMAT statement to enter the real estate data. Verify that SAS reads the data correctly.

FORMAT STATEMENT

FORMAT statement is used to assign **temporarily** formats to SAS variables to display them in PROCs. Typically, there is a need to display numeric variables in their original format with dollar signs, comma separators, or percentage signs.

* Format **DOLLARw.d** displays the numeric value with a dollar sign and comma separators.
* Format **COMMAw.d** displays the numeric value with comma separators but **without** a dollar sign. Warning: note that COMMAw.d. used as an informat removes comma separators as well as a dollar sign, but used as a format, adds comma separators but not a dollar sign.
* Format **PERCENTw.d** displays the numeric value with a percent sign.

Example. Suppose in the bank data example, we want to output variables Income and MortgageRate in the original format. To this end, type

proc print data=bankdata;

format income dollar9.2 mortgagerate percent8.3;

run;

Exercise. Display in proper formats the variables in the real\_estate data set.

Chapter 7 PROC FORMAT

PROC FORMAT is used to assign descriptive labels to data values. The syntax is

proc format;

value *formatname* value1='label1'

value2='label2'

…

;

run;

Example. Suppose a data set contains a categorical variable Race. The original values for this variable are African-American, Asian, Caucasian, and Latino. When SAS data set is created, it is convenient to enter the categories as 1, 2, 3, or 4. Then right below the DATA step it is wise to run PROC FORMAT as follows

proc format;

value racefmt 1='African-American'

2='Asian'

3='Caucasian'

4='Latino'

;

run;

Now the FORMAT statement is used to display the variable. For example, to print the values, type

proc print;

format race racefmt.;

run;

* Note that it is convenient to give format’s names by adding the suffix “fmt” to the variable name.
* Note that in the FORMAT statement format’s name must follow by a period.
* Note that the same format may be used for other variables, if needed. Just type

proc print;

format *variablename* racefmt.;

run;

* When the specified values are **character**, the format’s name must start with a dollar sign ($), and the values must be enclosed in quotation marks. For example,

proc format;

value $gradefmt 'A'='Excellent'

'B'='Good'

'C'='Fair'

'D'-'F'='Poor'

;

run;

proc print;

format grade $gradefmt.;

run;

* When the specified values are numeric, then it is possible to specify **value ranges** by using PROC FORMAT. For example, suppose Age is a numeric variable that assumes values between 15 and 78.

proc format;

value agefmt 15-<20='Teenager'

20-<65='Adult'

65-<100='Senior Citizen'

;

run;

* Keywords **LOW** and **HIGH** may be used in place of min and max values of a numeric variable. For example, the PROC FORMAT above may be written as

proc format;

value agefmt low-<20='Teenager'

20-<65='Adult'

65-high='Senior Citizen'

;

run;

* Note that a label must be enclosed in quotation marks.
* Note that a label is limited to 256 characters.
* Note that a double quotation mark should be used if an apostrophe is to appear in the label. For example, 4="Master's degree".
* Note that multiple value statements may be specified in a single PROC FORMAT. For example,

proc format;

value educfmt 0='None'

1='grades 1-9'

2='grades 10-11'

3='HS grad'

4='some college'

5='college degree'

6='some grad school'

7='grad degree';

value educnewfmt 0-2='<HS'

3='HS grad'

4-7='HS grad+';

run;

* The keyword **OTHER** may be used in place of all values that are not otherwise specified in PROC FORMAT. For example,

proc format;

value educnewfmt 0-2='<HS'

3='HS grad'

Other='HS grad+';

run;

* The PROC FORMAT may contain a list of values separated by commas. For example,

proc format;

value educnewfmt 0,1,2='<HS'

3='HS grad'

4, 5, 6, 7='HS grad+';

run;

or

proc format;

value $gradefmt 'A'='Excellent'

'B'='Good'

'C'='Fair'

'D','F'='Poor'

;

run;

Example.

data exercise;

input id male age actlevel @@;

cards;

1 1 52 1 2 0 28 3

3 1 33 3 4 0 52 2

5 1 25 2 6 0 50 1

7 0 55 2 8 1 51 1

9 1 31 3 10 0 37 2

11 0 34 2 12 0 28 1

13 1 23 2 14 1 44 1

;

proc format;

value malefmt 1='Male' 0='Female';

value actlevelfmt 1='Low' 2='Med' 3='High';

value agefmt low-<30='20s' 30-<40='30s' 40-<50='40s'

other='other';

run;

proc print;

format male malefmt. actlevel actlevelfmt. age agefmt.;

run;