LIBNAME, FILENAME and INFILE; RETAIN, FIRST.ID and LAST.ID

Examples of reading an external data file (.DAT or .TXT types) using the following options:

- **LIBNAME**
- **FILENAME and INFILE**
- **LABELS**

Let me explain these concepts using EXAMPLE13A.SAS

(Note: Note that EXAMPLE13a.TXT, EXAMPLE13b.DAT files must be stored in the folder, C:\DAT_FILE.

EXAMPLE13a.TXT and EXAMPLE13b.DAT will be used in the SAS programs, EXAMPLE13A.SAS and EXAMPLE13B.SAS, respectively)

LIBNAME:

- ✓ **LIBNAME is the key word in SAS.**
- ✓ **The basic syntax: LIBNAME libref 'SAS-data-library'; where**
  - **libref is 1 to 8 characters long**
  - **SAS-data-library is the name of the directory path in which SAS files are stored**
- ✓ **In the EXAMPLE13A SAS program, LIBNAME statement is the following:**

```
libname dat_file 'c:\dat_file\';
```

- **libref is the folder called DAT_FILE in which the file will be stored**
- **SAS-data-library is C:\DAT_FILE\.**
- **First part of the "two-level" SAS data set name is DAT_FILE.**

(Note that DAT_FILE is a subfolder in C:\)

The **LIBNAME** statement points to the subdirectory from which the file can be accessed or the subdirectory in which the files are to be stored.

FILENAME & INFILE

- ✓ **FILENAME is the key word in SAS.**
- ✓ **The basic syntax: FILENAME fileref 'filename'; where**
  - **fileref is a name we associate with an external file (1 to 8 characters)**
  - **filename is the fully qualified name or location of the external file**
- ✓ **In this EXAMPLE13A SAS program,**
  - **fileref is temporary name TEMP for the external file that contains the data**
  - **The complete filename is c:\dat_file\example13a.txt**

  (Note that DAT_FILE is a subfolder in C:\)
- ✓ **When we associate a fileref with an individual external file, we specify the fileref in subsequent SAS statements and commands using the key word INFILE.**
  - **Therefore, we have:**
    Infile temp; as a subsequent SAS statement.

The **FILENAME** statement is used to link an outside file to the SAS program by giving it a SAS name. This name is like an alias, or "*fileref*".

Note:

Assign a libref using LIBNAME and assign fileref using FILENAME

Fileref performs the same function as libref: they temporarily point a storage location for data but with one difference and that is, libref refers to SAS data libraries (folders) whereas fileref refers to external files.

INFILE:

The **INFILE** statement is used in the *DATA* step before the *INPUT* Statement to let SAS know that the data will be read from an external file instead of being put with *DATALINES* statements

LABELS:

- ✓ Some variable names like *GENDER, RACE, HEIGHT* and *WEIGHT* need no explanation, but other variables like *DOB, DOV, HRATE, SY_BP*, and *DY_BP* do require explanation.

- ✓ This can be accomplished by associating each variable name with a variable *LABEL* using a ***LABEL*** statement.

- ✓ The LABEL statements can be placed anywhere in the *DATA* step. The description can contain up to ***256*** characters and must be enclosed in single or double quotes.

- ✓ These *LABELS* will be printed along with the variable names in the SAS output in *certain PROC*s.

EXAMPLE13A.SAS

```
libname dat_file 'c:\dat_file\';
```

As I have explained above, libref is the folder called DAT_FILE in which the file will be stored and the SAS-data-library is c:\dat_file\.

```
data dat_file.sasdatafile_1;
```

This statement creates a permanent SAS data set called SASDATAFILE_1 and is stored in the folder, DAT_FILE.

Note that for this statement to run correctly, you must have the LIBNAME statement created above this statement.

```
label       num = "Subjects ID"
            age = "Subjects age"
            sex = "Subjets Gender"
            race = "Subjects race"
            employ = "Does subject have employment? Yes or No"
            insuaran = "Does subject have insurance? Yes or No"
            dm = "Diabetes? Yes or No"
            ht = "Hypertension? Yes or No"
            chd ="CHD? Yes or No"
            hb = "Hemoglobin reading"
            wbc = "White Blood Cell"
            pl = "Platelet count"
            creat = "Creatinine";
```

```
filename temp 'c:\dat_file\example13a.txt';
infile temp;
```

As I have explained above, fileref is a temporary name TEMP for an external data file which is EXAMPLE13A.TXT in this program.

This external data file has an extension.TXT. That is, it is a text file.

filename is the fully qualified name or location of the external file which, in this program, is 'c:\dat_file\example13a.txt';

(Note that DAT_FILE is a subfolder in C:\. EXAMPLE13A is in an external file that is stored in DAT_FILE)

```
input Num age sex $ race $ employ $ insuaran $ DM $ HT $ CHD $ Hb WBC PL
Creat;
run;
*****************************************************************;
proc print data= dat_file.sasdatafile_1;
      title 'Tabulation of data from sasdatafile_1';
run;
```

```
proc freq data= dat_file.sasdatafile_1;
      title 'Output from Proc Freq on sasdatafile_1';
      tables sex race employ insuaran dm ht chd;
run;

proc means data= dat_file.sasdatafile_1 maxdec=2;
      title 'Output from Proc Means on sasdatafile_1';
      var age hb wbc pl creat;
run;
```

What does EXAMPLE13A do?

It takes an external data file called EXAMPLE13A.TXT which is stored in DAT_FILE, and converts it to a SAS data file called SASDATAFILE_1 and stores it in the folder, DAT_FILE.

Therefore, DAT_FILE.SASDATAFILE_1 is a two-level name.

Note that every SAS data set has an extension, SAS7BDAT

Note that every SAS program has an extension, SAS

The previous program used a combination of FILENAME and INFILE to read an external data file.

In the following program, we shall use just the INFILE statement

EXAMPLE13B.SAS

```
*************************************************************;
* Use of INFILE statement and an external file with extension .DAT;
*************************************************************;
*     The following segment of the program creates a PERMANENT SAS
      data set by name SASDATAFILE_2 in the subdirectory, DAT_FILE.

      This segment also uses INFILE option to read the raw data
      from an external file EXAMPLE13B.DAT;
*****************************************************************;
*********************************************************************;
libname dat_file 'c:\dat_file\';
      The explanation of this is given above.
*****************************************************************;
data dat_file.sasdatafile_2;

label       num = "Subjects ID"
            age = "Subjects age"
            sex = "Subjets Gender"
            race = "Subjects race"
            employ = "Does subject have employment? Yes or No"
            insuaran = "Does subject have insurance? Yes or No"
            dm = "Diabetes? Yes or No"
            ht = "Hypertension? Yes or No"
            chd ="CHD? Yes or No"
            hb = "Hemoglobin reading"
            wbc = "White Blood Cell"
            pl = "Platelet count"
            creat = "Creatinine";

infile 'c:\dat_file\example13b.dat';
```

The syntax is:

INFILE "file-specification"

where file-specification can take the form "filename" to point to the actual name and location of the file

In this SAS program, it refers to an external data file EXAMPLE13B.DAT that is stored in "c:\dat_file\ folder.

```
input Num age sex $ race $ employ $ insuaran $ DM $ HT $ CHD $ Hb
      WBC PL Creat;
run;
*****************************************************************;
```

```
proc print data= dat_file.sasdatafile_2;
      title 'Tabulation of data from sasdatafile_2';
run;

proc freq data= dat_file.sasdatafile_2;
      title 'Output from Proc Freq on sasdatafile_2';
      tables sex race employ insuaran dm ht chd;
run;

proc means data= dat_file.sasdatafile_2 maxdec=2;
      title 'Output from Proc Means on sasdatafile_2';
      var age hb wbc pl creat;
run;
*************************************************************;
```

What does EXAMPLE13B do?

It takes an external data file called EXAMPLE13B.DAT and converts it to a SAS data file called SASDATAFILE_2 and stores it in the folder, DAT_FILE.

Therefore, DAT_FILE.SASDATAFILE_2 is a two-level name.

The following program has the data as part of the program

EXAMPLE13C.SAS

```
*********************************************************************;
* The following program does NOT use FILENAME or INFILE statements
  to read the data. The data are entered as part of the SAS program.

* However, we use LIBNAME statement to create a permanent SAS data file
```

The following program creates a PERMANENT SAS data set by name

SASDATAFILE_3 in the subdirectory, DAT_FILE;

```
*********************************************************************;
libname dat_file 'c:\dat_file\';
*********************************************************************;
data dat_file.sasdatafile_3;

label       num = "Subjects ID"
                  age = "Subjects age"
                  sex = "Subjets Gender"
                  race = "Subjects race"
                  employ = "Does subject have employment? Yes or No"
                  insuaran = "Does subject have insurance? Yes or No"
                  dm = "Diabetes? Yes or No"
                  ht = "Hypertension? Yes or No"
                  chd ="CHD? Yes or No"
                  hb = "Hemoglobin reading"
                  wbc = "White Blood Cell"
                  pl = "Platelet count"
                  creat = "Creatinine";
input Num age sex $ race $ employ $ insuaran $ DM $ HT $ CHD $ Hb WBC PL
Creat;
datalines;
1 40  M     B     N     N     N     N     N     14.9 3.2 220 0.9
2 47  M     B     N     Y     N     Y     N     9.9 11.4 148 5.7
3 47  M     B     N     Y     N     Y     N     8.7 3.4     294   0.8
4 37  M     W     N     N     N     N     N     16.7 9 231 0.6
5 45  M     W     N     Y     N     N     N     12.5 9.5 264 0.8
6 40  F     B     N     Y     N     N     N     8.4   5.1   308   0.7
7 24  F     B     N     Y     N     N     N     7.4   10.2 212 0.5
8 60  M     W     N     N     N     N     N     9.3   6.7   211   1
9 48  M     B     N     Y     N     N     N     . . . 0.9
10 38 F     B     N     Y     N     Y     Y     11.1 11.8 43 7.4
;
*********************************************************************;
proc print data= dat_file.sasdatafile_3;
      title 'Tabulation of data from sasdatafile_3';
run;

proc freq data= dat_file.sasdatafile_3;
      title 'Output from Proc Freq on sasdatafile_3';
      tables sex race employ insuaran dm ht chd;
```

```
run;

proc means data= dat_file.sasdatafile_3 maxdec=2;
      title 'Output from Proc Means on sasdatafile_3';
      var age hb wbc pl creat;
run;
*************************************************************;
```

EXAMPLE13D.SAS

Data Management of longitudinal Study data

Sometimes we may have to create data set that contains the data for the LATEST VISIT (or for the FIRST VISIT) for each patient.

```
*****************************************************************
1. Name of the program is EXAMPLE13D.SAS
2. Sometimes we may have to create data set that contains the data
   for the LATEST VISIT (or for the FIRST VISIT) for each patient
*****************************************************************;
* The following program creates a SAS data set, EXAMPLE13D.SAS7BDAT
  with all records;
data example13d;
input id name $ visit_dt tchol sysbp age;
informat visit_dt mmddyy10.;
format visit_dt mmddyy10.;
datalines;
1 John 12/16/1999 270 180 69
1 John 4/17/2000 220 160 69
1 John 2/4/2013 198 140 69
2 Amy 10/30/2001 218 150 21
2 Amy 12/4/2001 205 145 21
2 Amy 4/2/2002 185 130 21
2 Amy 1/23/2003 160 110 21
3 Paul 5/1/2000 240 145 47
3 Paul 5/22/2000 230 140 47
3 Paul 9/19/2000 215 130 47
3 Paul 8/14/2003 187 117 47
;
run;
*************************************************************************;
* First SORT the dataset by ID and VISIT_DT;
proc sort data=example13d;
      by id visit_dt;
run;
*********************************************************************
To create a data set by name LATESTRECORD.SAS7BDAT that contains
the records for each person by their latest VISIT DATE (VISIT_DT);
data latestrecord;
      set example13d;
by id;
if last.id;     *We can also use if last.id=1;
run;
proc print data=latestrecord;
      title 'Tabulation of the data set that contains the latest';
      title2 'record for each patient';
run;
*********************************************************************
```

EXAMPLE13E.SAS

Data Management of longitudinal Study data

Suppose we want to find, for example, the difference in Total Cholesterol (TCHOL) between the first visit and the last visit for each person.

```
*****************************************************************
1. Name of the program is EXAMPLE13E.SAS

2. Suppose we want to find, for example, the difference in Total Cholesterol
   (TCHOL) between the first visit and the last visit for each person.

3. We want to do the same thing for SYSTOLIC BLOOD PRESSURE (SYSBP)

4. Name of the program is EXAMPLE13E.SAS
*****************************************************************;
* The following program creates a SAS data set, EXAMPLE13E.SAS7BDAT
  with all records;

data example13e;
      input id name $ visit_dt tchol sysbp age;
      informat visit_dt mmddyy10.;
      format visit_dt mmddyy10.;
datalines;
1 John 12/16/1999 270 180 69
1 John 4/17/2000 220 160 69
1 John 2/4/2013 198 140 69
2 Amy 10/30/2001 218 150 21
2 Amy 12/4/2001 205 145 21
2 Amy 4/2/2002 185 130 21
2 Amy 1/23/2003 160 110 21
3 Paul 5/1/2000 240 145 47
3 Paul 5/22/2000 230 140 47
3 Paul 9/19/2000 215 130 47
3 Paul 8/14/2003 187 117 47
;
run;
*********************************************************************;
* First SORT the dataset by ID and VISIT_DT;
proc sort data=example13e;
      by id visit_dt;
run;
*****************************************************************
Note that the original data set EXAMPLE13E.SAS7BDAT has already been
sorted by ID and VISIT_DT;

data difference1;
      set example13e;
by id;
retain first_tchol first_sysbp;
```

LIBNAME, FILENAME and INFILE; RETAIN, FIRST.ID and LAST.ID

```
****************************************;
* If there is only 1 visit for any person, then that observation
  has to be deleted;
if first.id and last.id then delete;
****************************************;
if first.id then
            do;
                  first_tchol = tchol;
                  first_sysbp = sysbp;
            end;

if last.id then
            do;
                  diff_tchol = tchol - first_tchol;
                  diff_sysbp = sysbp - first_sysbp;
                  output;
            end;
run;

proc print data=difference1;
      title 'Tabulation of the data set that contains the latest records for
             each patient';
      title2 'as well as the NEWLY CREATED variables, DIFF_TCHOL and
              DIFF_SYSBP';
    title3 'That is, the difference in TCHOL (and SYSBP) between the FIRST
            and the LAST';
    title4 'visit for each person';
run;
****************************************************************;
```