Read in Thousands of Datasets with One Click!

Baibai Chen, ICF International, Inc., Rockville, MD
Joe Singh, ICF International, Inc., Rockville, MD

ABSTRACT

Got thousands of datasets to read in and process on a tight schedule? The Florida Youth Tobacco Survey (FYTS) has provided data for monitoring and evaluating tobacco use among youth for the Florida Department of Health's Bureau of Tobacco Prevention and Control since 1998. In 2015, 35 counties submitted middle school data while 33 counties submitted high school data. The maximum school or class size within one county is over 130. In each dataset scanned, there are three different surveys. Over time, we have read in thousands of datasets on two different platforms, Linux SAS and PC SAS. The final output will be three combined data sets by survey type. Our approach is to use "PIPE" to get the list of county and school, and use PROC SQL and SAS macros to automatically read in all the datasets and process them according to the survey type.

INTRODUCTION

The Florida Youth Tobacco Survey were distributed within classrooms with the intention of having every third student complete the FYTS at both the high school and middle school level. The sampling frame covers all public high schools and middle schools in Florida, among whom 174 schools were finally selected to participate. Over 12,000 completed survey questionnaires were collected in 2015. Several additional steps involving data cleaning, editing and aggregating were then taken in order to prepare the data for further analytical use.

The previous approach involved using simple SAS@ MACRO to read in all the data sets. That approach worked, but required considerable time and attention from the programmer and was complex and error prone.

In engineering a new approach, we sought to overcome four challenges: (1) handling three different types of surveys for middle schools, high schools and tobacco use, in the same file at the same time; (2) creating a unique key variable, school ID, based on the information embedded in the dataset name; (3) streamlining the SAS code so that it could be run on different platforms; (4) shortening the processing time .

In this paper, we would show how to develop a program to cope with the various challenges and deliver the final data sets with high quality.

READ IN DATA WITH THREE DIFFERENT INPUTS

Since each file contains three different surveys, we have to read in the data from three different inputs.

Frist, we will identify which survey of each record.

```
data y&sch._&cls._&typ;
infile "Z:\Florida\2015 FYS\Scanned Data\Middle School
Data\&_county\&filein"  dsd missover lrecl=10000;
input
                @21     SURVEYTYP  ??   1.
                @129    Litho   ??      8.@;
```

Second, we will include different input statement for each record according to which type of survey that record is.

```
if surveytyp = 1;
%include "Z:\Florida\2015 FYS\SAS Programs\SAS Codes
Include\MS_FYSAS_INPUT.sas";
```

The macro variable “&sch” is the school’s ID number and the macro variable “&cls” is the class number. The macro variable “&typ” is the survey type. Variable SURVEYTYP has values 1, 2, or 3 to represent three different type of survey: Florida high school health behavior, middle school health behavior, or Florida Youth Tobacco Survey. We will repeat the code above to read in one data file the time for a specific county, school, and class and output three data sets for each type of survey for later to compile the national data.

USE “PIPE” COMMAND TO GET A LIST OF COUNTY NAMES AND SCHOOL IDS

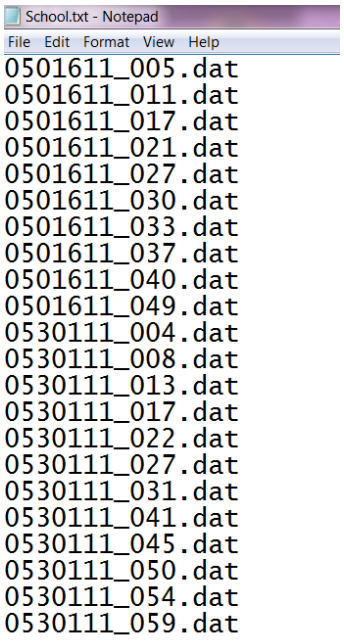
Previously, people use simple SAS@MACRO to read in the data. Such as

```
%readin(srvy = YRBS, cnty = BAY, cntyno = 03, sch = 0300611, cls =
047, level = 1);
%readin(srvy = YRBS, cnty = BREVARD, cntyno = 05, sch = 0520111, cls =
010, level = 1);
```

The shortcomings of this method were error-prone and time-consuming. There are over 1300 lines of code to read in over one thousand data files for just one type of survey.

Now, we will get a list of school data file names first. On LINUX SAS, we just used the UNIX shell command line to get the list of file names in the directory of interest and redirecting the output to a separate text file, called “school.txt”.

```
>ls *.dat/b > school.txt;
```



Next, we would read “school.txt” into SAS and use the file names as a list for picking up school IDs in a macro variable.

Now, with our new, streamlined approach, we achieve the same results by using the SAS@ command “PIPE”, which acts like the UNIX/Linux/DOS pipe operator “|” for commands or the redirection operator “>”. The advantages of this new way are no opportunity for type error, better time performance.

```
filename school pipe 'dir Z:\Florida\"2015 FYS"\"Scanned Data"\*.dat
/b';

data school_list;
length fname1 $20;
infile school truncover;
input fname1 $20.;
call symput ('sch',substr(fname1,1,7));
call symput ('cls',substr(fname1,9,3));
call symput ('num_files',_n_);
run;
%macro readfile;
%do i = 1 %to &num_files;

data _null_;
set school_list;
if _n_ = &i;
call symput ('filein',fname);
run;
/* here add code to read each type of survey data */
%mend readfile;
%readfile
```

TWO BIRDS WITH ONE STONE—WE HAD SCHOOL ID, CLASS ID, AND COUNTY NAME AS THE DATA FILE NAME AND THE VALUE OF THE VARIBLES IN EACH DATA SET

When we use PIPE command, we also create the macro variables of school ID and class ID. So we will add those variables in each of the output data sets.

```
/*adding variables SCHOOLID, CLASSID, and COUNTY to data set*/
data y&sch._&cls._&typ;
  set y&sch._&cls._&typ;
  COUNTY = substr(&sch.,6,2);
  SCHOOLID = &sch.;
  CLASSID= &cls.;
Run;
```

Finally, we will compile all the data sets by the survey type. We use “PROC SQL” to get all the data sets name.

```
PROC SQL NOPRINT;
     SELECT QUOTE(TRIM(FNAME)) INTO :SCHLIST SEPARATED BY '  '
     FROM SCHOOL_LIST;
QUIT;

data y._&typ;
set &SCHLIST;
Run;
```

There are only 7 lines of code to compile a national data for one type of survey. To read thousand data and create three different survey data, we only have less than 150 lines of code.

CONCLUSION

Using the powerful **SAS**@ MACRO, PROC SQL and PIPE commands to handle the multiple unites of processing (county, school, and class) and multiple types of surveys, we not only save a great deal of time, but also avoid all the typos in the data processing.

ACKNOWLEDGMENTS

Specially thanks to Tonja Kyle (Principle, ICF International, Inc.), Brenda Clark (Principle, ICF International, Inc.), Wen Song (Technical Specialist, ICF International, Inc.), and Lee Harding (Technical Specialist, ICF International, Inc.) for helps and support.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact us at:

Baibai Chen
Joe Singh
ICF International, Inc.

530 Gaither Road,
Rockville, MD 25850
Work Phone: 301-407-6500
Fax: 301-407-6501
E-mail: bchen@icfi.com
jsingh@icfi.com