

SAS Online Data Archive for Population Studies

David Barro, Leslie Benson, Steve Maczuga, Cindy Mitchell, Jeanne Spicer
Population Research Institute, Penn State University, University Park, PA

ABSTRACT

Our data archive serves up a wide selection of social science data in every flavor from EBCDIC tape to .dbf files downloaded from the web. Most of our data was kept as "flat" files so researchers could read the data into their analytical software of choice. This prevented us from having to keep a permanent SAS system file, SPSS version, Stata version, etc of the same data set. We stocked basic SAS code to input the raw data into a SAS data set. Labels and formats were generally not included unless the collecting agency supplied them. Users were on their own if they wanted the data in anything other than SAS. This resulted in confused users as well as inefficient and/or error prone data. In the end the users often created data files which violated two of the cardinal sins of programming - wasting CPU time AND disk space.

The solution:

Using skills & resources already at hand we've mixed-up a concoction called the SodaPop system. For our users it goes down smoothly; for our programmers, it's simple to prepare and for our network administrators, it's economical. Our SAS/AF experts provided a gateway to the workspace, our data management people created a combination of compressed SAS data tables & views for storage, our SAS/Intrnet gurus added a variable lookup facility and online documentation.

INTRODUCTION (THE WAY WE WERE)

We are known as the Computer Core of the Population Research Institute (PRI) at Penn State. PRI is an inter-department Institute using demographic data from various sources, to facilitate research into population trends. Our archive includes over 100 data collections from sites such as the U.S. Census Bureau, U.S. Department of Labor, National Institute of Health and the United Nations as well as data collected by Penn State researchers. The data archive includes data sets that are available for public use and others which have tight security governing their usage. The researchers are both professors and graduate students who come to Penn State with widely different computer backgrounds, knowledge and comfort. Our mission is to provide computer support to the researchers to facilitate their research. This has primarily involved offering

computer training workshops, individual consulting and data preparation and cleaning as well as data set customization for individual projects. While performing these tasks we have had the opportunity to observe our non-computer colleagues closely, to note what works and what does not as well as what they say would help them. This led to our Computer Core group design and implementation of what we fondly refer to as the SAS Online Data Archive for Population Studies -- SodaPop. [It would not be a computer application without an acronym now would it?]

The professors at PRI represent over 10 different fields of study: Demography, Sociology, Economics, Health and Human Development, statistics, to name a few. They work in several different buildings around campus. Some use PC's and some work directly on our UNIX based network (PopNet). We store copies of hundreds of data collections on our UNIX cluster. Many of our collections contain data taken from surveys. Some of these surveys may be of the "once & done" type, others may be taken annually or at other regular intervals such as the decennial census. Each data collection is stored in its own directory with each "wave" stored in a sub-directory. For example all of the Current Population Surveys are stored in the directory: /home data/census/cps. The "cps" directory is then divided into a subdirectory for each year that the CPS survey has been taken. The data itself, along with a codebook, which includes the questionnaire and the variable descriptions and a SAS program to read the raw data is stored in the appropriate subdirectory. Prior to our SodaPop system, each researcher who needed to study survey data ran the batch SAS job to create the SAS data set from the flat file. This data set was usually stored in a temporary shared directory called /sastmp which was "swept" every 3 days. We noticed that many of the researchers, particularly the students, created a data set containing all of the survey variables (sometimes numbering in the thousands!) even if they were only performing statistical analysis on a few of them. We also found multiple copies of the same data set parked around the network. This is an obvious waste of disk space as well as computer time.

Due to the time the students had to devote to their major area of study, they could not become 'computer experts'. Although we offer SAS workshops, not all students ever became truly

comfortable with computers. This led to an interesting phenomenon, the students share bad SAS code. This code typically creates the SAS data set, recodes some variables and then performs some type of analysis. Each grad student adds their own code to the program, without taking the time or having the ability to understand how it worked before. Therefore the student is afraid to remove any of the previous code. Over generations it becomes increasingly difficult for this spaghetti code to be debugged.

We felt that it would help if users did not have to work with raw data at all; instead concentrating on learning how to subset existing SAS data. We felt it would help if they had a single place to go in order to find everything they need: the data, the survey instruments and the documentation (codebooks). We also felt that we could provide some additional meta-data to help them sift through the gigabytes of data in the archive.

THE SODAPOP SYSTEM (AS IT IS NOW)

The creation and implementation of the SodaPop system had to take place during our "free time", always taking a back seat to project deadlines, teaching and consulting schedules. We all had a notion of what needed to be done and we let our individual skills and experience determine a workable solution. The person that knew a little SAS/AF said "I can build a point & click interface for SAS users"; the person who knew a little SAS/Intrnet came up with a nifty variable lookup routine; etc.

The system uses several SAS components and SAS/Intrnet powered web pages to aid in the use of SAS datasets and/or SAS libraries for our students and researchers. The user logs in to the network and launches a Version 8 interactive SAS session. The application is launched with an AF command: "af c= archive.startup.mainmenu.frame". To make that a little easier, the user is encouraged to add a custom icon to their SAS Toolbar to issue the command "on-click". Our icon is our unit's logo. [This is a one-time process, but it would be nice to be able to do this for all users globally.]

When the application begins, a SAS/AF window and a web-browser window appear (see figure 1 below). The tabbed layout object allows the user to select one of the data collections commonly used at the PRI (i.e. Summary Tape File (STF), part of the US Census). A data collection may contain SAS datasets from one year or many years.

Once a checkbox is activated and the OK button "clicked", the SAS/AF program issues libname statements for the SAS library associated with the data collection. For instance, part 3C of the STF includes 50 SAS datasets. The libname statement issued by the SAS/AF application for the entire STF collection for 1990 would be:

```
libname stf90 '/home/sas_data/stf90';
```

Our users have always been baffled by "libname" statements. But SodaPop issues those statements behind the scenes and all the user needs to do is look in the SAS Explorer window for the folder (library) and file (dataset). The data are ready for use immediately.

The SAS/AF application also activates a web browser window that offers some general help on our system. For example, the web page contains links to a SQL Query Window on-line help article, a link to a STAT/Transfer on-line help screen, and a link to enable users to e-mail the computer staff at the Population Research Institute to ask for help or to provide comments.

After the user activates a SAS Library, this browser window will change to a help screen for the particular data collection. This page is labeled "Programmer's Notes". The "Programmer's Notes" page provides the names of all datasets in the collection, the location of a batch program to extract the data, links to related web sites, online codebooks and lastly, links to our SAS/Intrnet powered help pages. So all the documentation is right in front of them as they work with the dataset.

A "click" on the SAS/Intrnet powered help pages link allows the user to search the metadata for a particular data collection to find information on the variables. The search tool uses a SAS dataset made from the combined PROC CONTENTS for each dataset in the collection. Information returned to the user for each survey is the variable length, label, variable name, and dataset name. For instance, many questions about "INCOME" are asked in the Panel Study of Income Dynamics 1968-1995 (see figure 2). A search against our metadata would produce a list of over 130 variables pertaining to income from all the files in the collection. This is particularly helpful for building longitudinal files. These help pages are output using the %OUT2HTM macro in SAS/Intrnet. The searches can be performed on SAS labels, variable name or SAS dataset.

MAINTAINING THE SYSTEM

The first step is to write the code that creates the SAS data set from the raw data. In many cases a bare-bones version of an input statement is supplied with the data, but often without the niceties of variable labels or formats. Other times we have to write code from scratch or convert the data from another file format into SAS format. We put formats on hold if they are not provided, but the labels serve a purpose beyond the labeling of output, they form the basis of our descriptive information (or Metadata). So we take the time to add labels for all datasets.

Now that the data is in a dataset, it needs to be made available to the SAS/AF program. This is accomplished by updating index tables in the SodaPop library that contain information used to build the libname statements and load the radio button objects for the tabbed layout object which serves as the main menu. The tables are relatively simple. They consist of the library name, path or directory, the text or label describing the study, and the tab name. There is one table per tab and one tab for each category of data.

Now that the data is in place, and the users can see it on the SodaPop menu, the associated documentation needs to be created. If the new data is a continuation of a previous collection, all that may be required is to update or create hypertext links to any new documentation, or update the programmer's comments. If not, then a "Programmers Notes" html page should be generated.

The "Programmers Notes" pages are written using a standard format. The background color, logos, borders, and general format are the same for every page using a cascading style sheet. A program was written to help with the standardization of the html files. Using a NULL data step, the program PUTS lines of html code ensuring the same items are included in each file. The program uses macro variables to plug in the library name and the dataset names & labels into the html file.

After the basic shell of the "Programmers Notes" file has been constructed, links to codebooks, data directories, and/or other documentation along with information on running batch jobs against the data are added. Especially useful is a bibliographic citation identifying the source of the data. Any other documentation that the programmer feels may be important or of interest are also included at this time. The hypertext links to the documentation and notes must be manually included in the file.

Finally, a file to run batch jobs is constructed. [We still have a few users with the good sense to run an extract against a 10 million record file in batch mode while they run over to the local coffee house.] If the dataset can use formats created by PROC FORMAT, then two files need to be placed in the directory. The first contains the PROC FORMAT code that generates the formats, and the second the FORMAT statements used in the DATA and PROC steps. The "standard" names for these files are "procformat.sas" and "formatstmts.sas", respectively.

The composition of the batch file is simple. At the beginning a DATA statement and a SET statement with the KEEP= option followed by the list of all variables in the dataset and their labels (as comments). At end of the file are "%include" statements for a PROC FORMAT program, a PROC MEANS, and another include for the FORMAT statements.

```
DATA datasetname;
  SET SODA_POP_dataset (KEEP=
    variable1 /* label for variable1 */
    variable2 /* label for variable2 */
    .
    .
    last_variable /* label for last variable
*/
  run;
%include '/pathname/procformat.sas';
  proc means;
%include '/pathname/formatstmts.sas';
run;
```

To create an extract or subset of the data, the user can copy this file, and delete the lines to remove the variables that they do not want to keep. If they decide that they don't want to use the formats, they can comment out the include statements.

To aid in the creation of the batch file, a program similar to the one that creates the "Programmers Notes" shell was written. It takes macro variables for the library and dataset, runs PROC SQL code to create a table containing the variable names and labels from the sashelp.vcolumn table, and then runs a data step to create the ascii file containing the SAS code.

Once the bugs were worked out all the code was combined into one Metadata macro which outputs 1) programmers notes, 2) batch code and 3) html documentation using just a few macro parameters as input. A priority list to load data based upon the frequency of use or critical need of our researchers was developed and we just started loading 'em in.

THE FUTURE OF THE INTERFACE

The SAS/AF application requires a login to the UNIX network. As more and more users move from the workstations to an NT desktop, they are faced with the problem of emulating the graphical environment for an online SAS/UNIX session on their PC. This is forcing us to look into two options; porting the interface to the NT server (integrated into our UNIX network) or moving to the web.

The code could fairly easily be ported to an NT machine leaving the data on the UNIX platform file server (now that SAS v8 datatables are platform independent!) and the html pages could remain on the UNIX web server.

But in order to satisfy today's user's, "it's no good if it ain't on the web" mentality. We will eventually need to develop a web-based interface to all or part of the system. The problem is that some of our data sets are over a gigabyte in size. The future may bring a web-based version of the extract capabilities of the application whereby a user can select a dataset from the collection and choose the variables they need and specify the output format. The request would launch a batch job using the Application Dispatcher. Output would be deposited to temporary disk space or ftp server for the user to pick up.

CONCLUSION

Although we have seen many fancy web-based query & extract applications like StEPS from the U.S. Bureau of the Census or IPUMS from the University of Minnesota, PRI's data archive contains hundreds of data sets, each one containing hundreds or even thousands of variables. We simply do not have the staff or the time to devote to the construction of a tailor-made application that would be able to incorporate all of our varied data.

Our researchers have strong statistical backgrounds but their programming and data management skills run the gamut from excellent to those of the pointy-haired boss from the Dilbert cartoon. They want access to the data but they don't want to become programmers. They want to run fancy Procs but don't have a clue how to read in a variable from raw data described as 9(5)v99 in an old codebook. Many of our researchers and the vast majority of our graduate students have grown up in the point-and-click world of programming and data management. Negotiating UNIX paths and actually writing code is a foreign and painful experience.

The SodaPop System is our attempt to provide easy, reliable data access to researchers at PRI. Giving researchers the tools to find relevant variables, assign libnames and formats, access the "Programmer's Notes" and extract their own data using a GUI interface greatly expands the capabilities of our research associates and frees up the programming staff for more challenging research programming tasks.

ACKNOWLEDGEMENTS

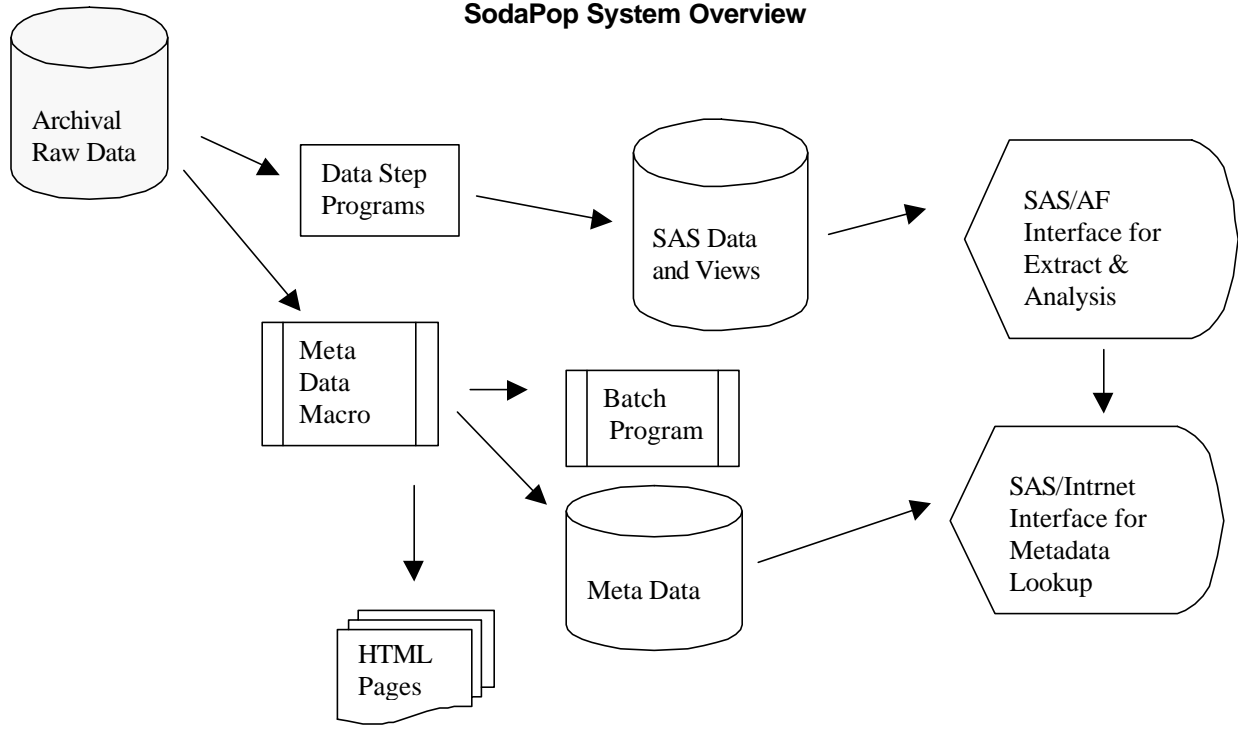
The authors would like to thank our resident "Eeyore" for his help & guidance in the writing of this paper.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors at:

PRI - Programming Team
Penn State University
813 Oswald Tower
University Park, PA 16802
spicer@pop.psu.edu

SodaPop System Overview



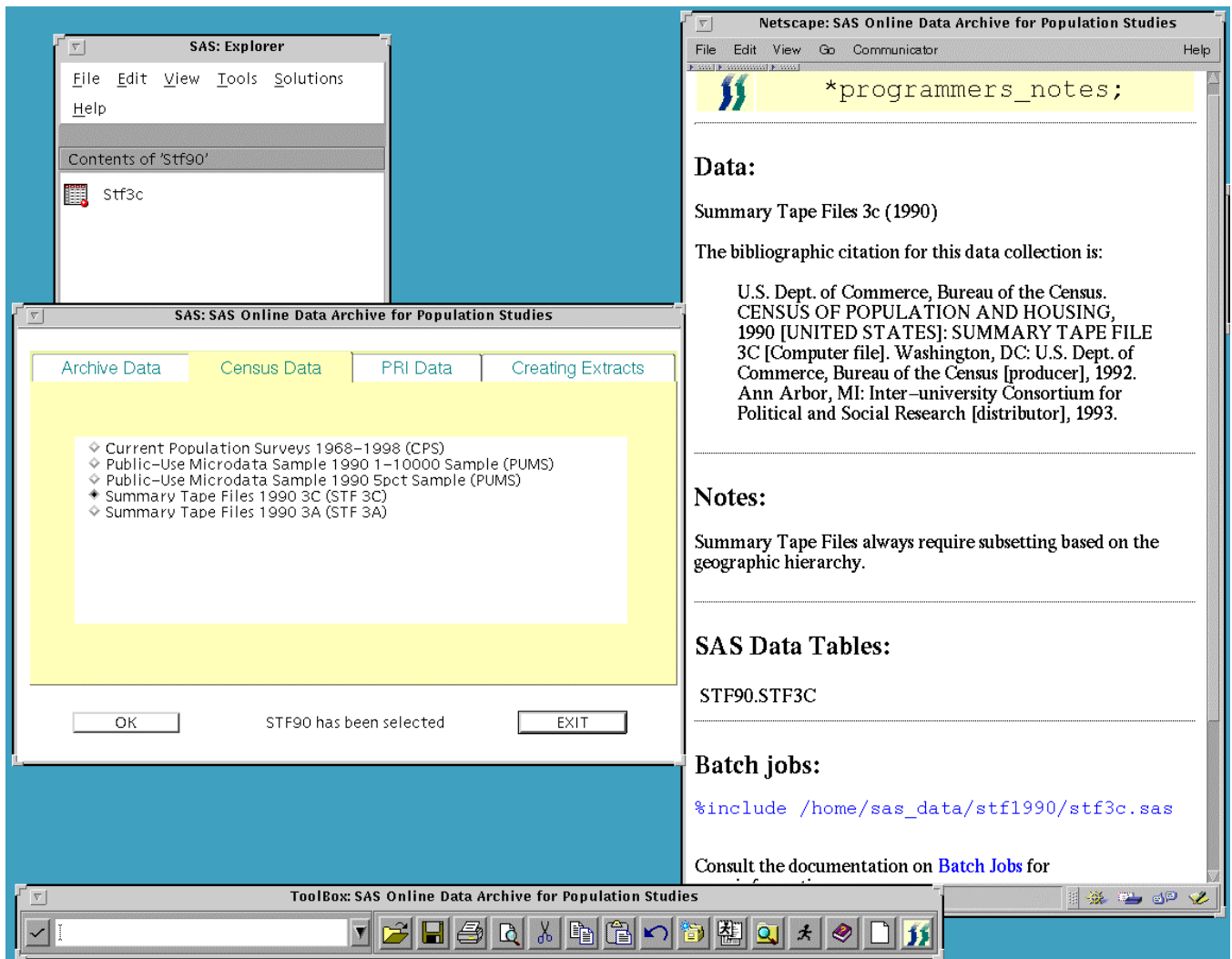


Figure 1

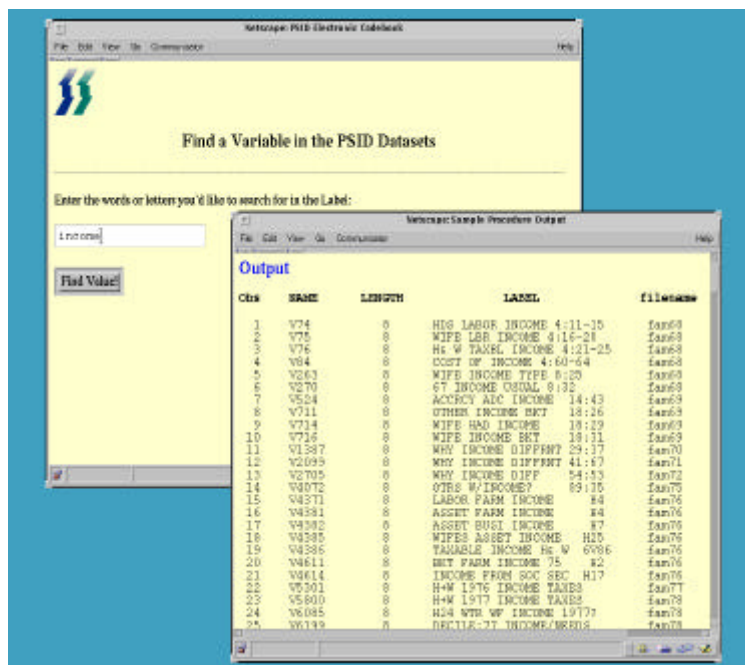


Figure 2