

COMPUTER AIDED SOCIOLOGY

A Report on the Creation of a Computer
Program Designed to Aid Sociology Data Analysis

Tom Mackendrick
Honors Project
for Dr. Harry Nelson
May 19, 1983

Sp 201
The
LD
2487
.24
1983
M33

With the advancement of computer technology, the computer has crept into almost every aspect of society. The social sciences are no exception. Computer aided data analysis has been used by sociologists for some time now. Over the last five years, the sociology department has been using the SCSS and SPSS systems (Statistical Packages for the Social Sciences) for analyzing data from various surveys and studies performed by the department. In addition these programs have been incorporated into sociology classes by having the students run these programs using data from studies on the local and national level. Using this data and the computer, students are able to compare relationships, correlations, path equations etc. of variables and note their effects on this community and American society.

But as the computer world grows, so does the desire to use the computer more extensively. With the speed and accuracy of the computer at sometimes staggering levels, the possibilities to administer the actual survey via the computer and obtaining data analysis on the spot became a real target. This is the purpose of this project: to create a computer program that will allow the administration, data organization, and data analysis of surveys to be done in a very short span of time, possibly the same day. Combining the already in use abilities of the SCSS and SPSS programs, the availability and assistance of the SRC, and the creation of the survey program, this project has attempted to reach the above stated goal.

INITIAL STEPS: EXACT FUNCTIONS OF THE NEW PROGRAM

The broad explanation of this program is as follows: it creates a program which runs a survey on the computer, stores the individual data recorded from each survey, and creates a variable file that stores the variable names. The program also is capable of taking a survey previously stored in the computer and creating a usable program with that survey. The program created to create a usable survey has the capability of taking inputted surveys which contain headings, questions, and responses to the questions. It also has the ability to discern certain responses so that questions may be skipped if the answer to the previous question renders the following question or questions non-applicable.

In doing all of this, the program has fairly simple response prompting for entering the questions and those that are to be skipped. The program is simple enough so that one need not be skilled in computer programming to enter the survey into the program.

The program also prompts the user as to the names of the data files to be created, the variable file name and the name of the survey program that is being created. Also, the option to create the program from a previously stored survey is easily accessible. The exact procedure for entering a survey will be discussed later.

When starting this project, the sociology department had already in existence an experimental program that came close to meeting these needs. The program had more than its share of "bugs" which needed to be corrected. Furthermore, the program had been written on the sociology department's micro computer. This limited the survey administration in

the fact that the survey could only be taken on the micro computer, a single terminal device located in the Sociology Office.

In order to allow more versatility to the survey administration, it was determined that the program should be converted for use on the Ball State's new VAX-11 system. This way storage would be greater and the survey would be more accessible to the public, being that the VAX has numerous terminals throughout campus.

Conversion turned out to be no easy task, and consumed much more time than originally expected. The original program was written using Micro-soft BASIC, which had to be translated to the BASIC language used on the VAX compiler. In order to do this, many long hours were spent learning not only both the languages, since I knew either, but also both systems which I had not used or become acquainted with previously. While the languages turned out to be very similar, the file creation, device handlers, and editing programs on the two systems were quite different. Full knowledge of file manipulation was needed to start the conversion and translation. Fortunately, manuals were readily available, as was the advice and aid of many computer users in East Quad. Eventually, the conversion was completed and the program was ready to be tested.

USING THE PROGRAM TO ENTER A SURVEY

The program is stored in two different PPN numbers on the VAX. One copy is in Dr. Harry Nelson's file and one is in Tom MacKendrick's file. In order to enter a survey, the user must log into the VAX using one of these PPNs. Then one must get into BASIC mode by typing BASIC. Then RUN TOM will execute the program and the survey is ready to be entered.

Since the program was designed to be used by non-programmers, the prompting is fairly straight forward. The first questions asked are for the names of 1) survey program 2) the file name that the variable names are to be stored 3) the file the results of the survey are to be and 4) the name of the file if the survey is already stored. This whole process is demonstrated below:

```
WHAT FILE SHOULD I PUT THE ACTUAL PROGRAM IN? name.bas
WHAT FILE WILL THE VARIABLE NAMES AND LABELS GO IN? name.ext(any
extention can be used here)
WHAT FILE DO THE RESULTS GO INTO? name.ext
WHAT FILE HOLDS THE INPUT IF OTHER THAN TERMINAL? name.ext (if
none, which is usually the case, then hit the return key)
```

In the above example, the capital lines are what the computer prompts with, the small lines are what the user responds. The actual program MUST have the extention BAS, since it is a BASIC program itself. The other three are left up to any name and extention the user pleases. The last question is usually answered by merely hitting the RETURN key since the user is about to enter a survey and not use one previously stored. If one is using a survey stored already, then the name.ext must be that of the file that holds the survey. From this point the program runs in a loop in order to enter the questions, responses, and headings.

Next, the computer will prompt the user for the variable name of the question to be entered and a description of the question. An example follows:

```
WHAT IS THE VARIABLE NAME AND LABEL? name description (RETURN)
```

Here, a one line brief description is allowed. Once the return is hit, the computer will prompt the user for the text of the question as follows:

ENTER THE EXACT TEXT OF THE QUESTION, ENDING WITH A BLANK LINE:
text of question.....(RETURN)
continued.....

Here, as many lines as needed can be used for the question. When finished with the text of the question, the user must hit the RETURN key twice.

Next, the program gives the user two options for how the question is to be answered. The program has the capability to use questions that are answered in a multiple choice format or an open-ended format. In this way, questions such as age, occupation, or opinion questions can be used as well as the more standard multiple choice. To distinguish between the two, the computer prompts:

HOW IS THIS TO BE ANSWERED? (N OR A) n for multiple choice types
a for open-ended types

When N is entered the computer asks for the responses,

WHAT ARE THE POSSIBLE ANSWERS TO THIS QUESTION?

Here, one types in the responses, each response being limited to one line only. When finished with all the responses, one simply hits the return twice. The program automatically assigns numbers to each one line response so the user need not do so himself.

It is in this part of the entering process that the information is relayed as to whether the questions that follow the one being entered are to be skipped during the survey or not. After each response, the user MUST type in the number of questions to be skipped. If none are to be skipped, as in the majority of the cases, the user MUST type in a zero(0) at the very end of the response. If he/she wishes to skip

N number of question, on must type in what number at the end of the response. An example follows:

```
WHAT ARE THE POSSIBLE ANSWERS TO THIS QUESTION?  
male0 (RETURN)    no skipped questions  
female2 (RETURN) skip the next 2 questions
```

To start the question loop again and enter the next question, hit the return key twice.

If A is entered for how the question is to be answered, the program simply returns with the prompt for the next question.

To enter headings for questions or sections of questions, one must do so in the question loop described above. When the prompt for the variable name and label appears, simply type in the variable name and label for the first question to follow the heading. Then when asked for the exact text of the question, type in the heading. When finished with the heading, type in the next question. To space lines between the heading and the question, type in at least one blank or other character on each line. DO NOT hit the return to enter blank lines! There MUST be at least one character entered or the program will move to the next prompt before the question is entered. When finished entering the heading and the question, hit the return twice.

When the last question has been entered and the user wants to terminate the program, he/she has two options. If at anytime during the entering process one wants to start over or create another program with another survey, all the user has to do is type in a Q to answer the prompt for the variable name and label question. In order to simply terminate the entering of the survey, the user must hit the CONTROL

key and the C key (control C) at the same time. This will cause the program to be exited. The user will now be in BASIC mode. The EXIT command will put the user back in monitor mode and all files created will be available for execution by the user.

Currently, there exists a survey program created by this program in storage. The survey program has been debugged for "typos" that occurred when the user inputted the survey. At the present moment, editing must be done by a user with computer programming skills and an active knowledge of both the BASIC programs. This is because the actual program must be edited, not just the survey. Plans for the future consist of improving this system so that editing can be done by a non-computer user on the survey itself.

The program that was created (the survey program) follows, as does the file containing the variable names and labels. Also, an example of how the results are stored is included.

ADMINISTRATION OF THE SURVEY

An outline for the administration of the survey program follows:

- 1) Distribute the PPN that contains the survey program to the respondents.
- 2) Once on the PPN, they will be "switched" automatically to the survey.
- 3) Prompt respondents to log off when finished with the survey.
- 4) Monitor the activities.

The PPN number can be given to respondents by distributing it in classes or can be placed as a bulletin in the computer rooms to be used for the survey. This procedure will have to be modified since the computer center frowns upon such access. Since there will be public access to the file, no other programs should be in the PPN. Also, a

subdirectory should be utilized for the program itself so that the respondents can not change or tamper with the survey program.

Once the user has logged on into the terminal, the switch (a command program) will automatically take the user through the survey. Then the program will ask the respondent to logoff when through taking the survey.

The survey program is very straight forward and can actually be done with very little monitoring. The program only allows numbers to be entered for responses on the multiple choice questions and will tell the respondent if he/she makes an error. The computer will continue to prompt for the correct type of answer until it receives it. During monitoring stage, the result files must be appended together every so often since most PPNs are limited to the number of files they can use. Once the survey has been taken by as many respondents as needed and the results appended into one file, they are ready for the SPSS or SCSS data analysis programs.

CONCLUSION

These new programs will aid the sociology department in the speed and ease of which a survey can be given. Not only can they be used in the way described above, but phone surveys can be used by having the interviewers type the responses directly into the computer. With copies of the survey program in each interviewers own PPN, many interviews can be given at the same time.

Sociology has moved into the world of the computer by being able to utilize the capability to generate large amounts of data, analyze it, and give the results all in the same day if need be. With the aid of the programs described in this report, the sociology department will

now be able to produce data information at the speed and accuracy expected in today's high-tech world.