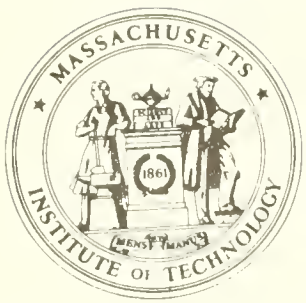

WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

APPLICATION OF A FLEXIBLE SYSTEM TO RETRIEVE, MANIPULATE, AND DISPLAY INFORMATION FROM A STABLE, QUESTIONNAIRE-ORIENTED DATA BASE TO SOCIAL SCIENCE RESEARCH

David Ness and Christopher R. Sprague*

594-72

March 1972

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

APPLICATION OF A FLEXIBLE SYSTEM TO RETRIEVE, MANIPULATE, AND DISPLAY INFORMATION FROM A STABLE, QUESTIONNAIRE-ORIENTED DATA BASE TO SOCIAL SCIENCE RESEARCH

David Ness and Christopher R. Sprague*

594-72

March 1972

Presented at the National Meeting
of the Operations Research Society of America,
Anaheim, California, 28 October 1971.

*Visiting Associate Professor of Industry
Wharton School of Finance and Commerce
University of Pennsylvania

ABSTRACT

The needs of managers making media allocation decisions have led to the establishment of a number of syndicated services selling data (demographic, media habits, and product consumption) derived from very detailed questionnaires administered periodically. Where there are enough users to support on-line storage charges, these data can be retrieved much more economically in a time-shared environment than in batch, and interactive systems with this capability are now in existence.

Such an interactive system would be of considerable utility to a researcher interested in any relatively stable questionnaire-oriented data base, the U.S. Census, for example. The syndicated data bases themselves may also be of considerable value in some research.

This paper describes such a data base, and an interactive system giving access to it. A simple study involving the relationship between total income and sources of non-employment income is described.

Introduction

This paper describes one member of a class of data bases available from syndicated sources, an interactive system giving access to many such data bases, and potential applications of the system to research in the social sciences. While the emphasis is on the system itself, the data bases themselves may well be of interest. The data bases were collected, for the most part, to facilitate the media allocation decision, but contain demographic and product consumption information as well.

The Media Allocation Problem

In making decisions about how resources should be allocated to advertising media, managers often need information about the interrelationships between the consumption of various goods and the consumers' media habits (what magazines he reads and what television shows he watches, for example). In an attempt to help answer some of these questions W. R. Simmons and Associates Research has for years conducted extensive surveys of the demographics, consumption patterns and media habits of a consumer panel, as well as some psychographic information. This data is collected once each year and the data is tabulated into a number of reports which are supplied to many advertising decision makers.

These managers also request, from time to time, that particular reports be prepared, tailor-made to their specifications, from the data collected in the survey. This service had, in the past, been supplied on 24-48 hours notice, by a batch processing computer system. The system

described in this paper discusses an implementation of such a service on an interactive basis involving turn-around times of minutes rather than hours. This allows interested researchers to access the rather extensive data base in an interactive fashion, thus making it much quicker (and easier) to test hypotheses which they generate. We call this system the "Interactive Market System" (IMS).[1]

The Panel

The specific panel referenced in this paper (surveyed for the year 1970) consists of 15,322 people. The sample is stratified to over-represent high consumers in an attempt to obtain more accurate estimates of consumption than would be possible otherwise. Approximately 15,000 bits of information are collected from each respondent. Other data bases are also accessible to the user of the system being discussed here.[2]

System Overview

In order to be able to respond to a researcher's request for information about some of the panel (and thus by implication about the population) in an interactive fashion it is clear that the data must be organized in such a way as to require only the scanning of the subset of

---

[1] Interactive Market Systems, Inc., 360 Lexington Avenue, New York City.

[2] By accessible we mean available to users who have purchased the data from the appropriate supplier.

information pertinent to answering that question. If the whole data base had to be scanned (as was the case, incidentally, with the batch system) hours would be required. This implies that the data must be stored in a directly accessible fashion, organized by questionnaire item, not by respondent. Stored in this way answering a typical request (for example, to compare the coffee consumption high income respondents with that of low income respondents) would require the retrieval of only a very small subset of all of the information available. Only in this way could the time constraints implicit in an interactive system be met at a reasonable cost.

Since the data, as it is originally collected, is naturally organized by respondent, not by item, an "inversion" of the file is implicit in the system we are discussing. As this data is generated only once each year, such a process need only be performed annually. The cost of the inversion can therefore be spread across many retrievals. Similarly, if there are enough users, the cost of on-line storage of the data can be spread across all of them.

We will not describe the inversion process in detail here. It suffices to note that the data collected as if on cards, many per respondent. The original data base is sequential, all information for one respondent preceding any information for the next. Classification of respondents is accomplished by presence or absence of punches in the cards. For example, the sex of a respondent is indicated on card 1, column 9, punches 1 and 2.

Reference to specific data items is handled through a hierarchy of languages and directories which allow the system to translate a user statement into a series of requests for data retrieval and computation. The languages at each level are all available to the user if he cares to make use of them but as might be expected almost all user requests are stated in the highest level language appropriate, and all of the translation is left to the system.

The File Structure

A substantial fraction of the data stored in this data base is of intrinsically binary character. Integer valued data is usually broken into ranges by the questionnaire (for example, milk consumption might be recorded as 0, 1 quart, 2-4 quarts and more than 4 quarts). This can easily be represented in binary fashion. Few real-valued variables are of concern in this process.

For this reason we adopted a binary data structure, where each basic storage block consists of a string of 15,322 bits where bit #1 represents Consumer #1, bit #2 represents Consumer #2, etc. For integer valued data (as in the milk example above) several separate strings of bits are used. We will call an integer-valued variable a "construct" because we must, using this scheme, construct a variable out of its separate bits.

The Directory Structure

There are two kinds of directories used in this system. The "logical" directories which are concerned with the problem of translating references

made in the higher level languages into statements concerning the questionnaire (for example, a reference to "smoker" might be translated into a reference stating that the answers to questions about smoking are found on card 3 in column 5). The "physical" directories are concerned with translating a statement of card number, column number (and perhaps which punches in the column) into an actual storage location and reference.

The highest level of physical notation we call the "atsign" language. In this language we refer to a card, column and punch in a six character identifier: @XXYYZ where XX is a card number, YY is a column number and Z indicates a punch position (1, 2, . . . 9, 0, X, Y). Thus @0305X refers to an X punch in column 5 of card 3. This was chosen as the highest level physical language because it was easy for an important subset of users (namely those who had previously used the existing batch processing system) to understand. Any user who is already familiar with this language (most current users are) is thus able to refer to such quantities directly without the pretranslation which would be required if a more mnemonic reference was used.

The next level of reference is called "number-sign" language. Here a quantity #XXXXX refers to the XXXXXth string of bits in the data base. This level of reference was introduced for two reasons. First, if we had used actual disk block locations instead, then any change in the size of sample (an admittedly rare event) would have implied a change in # references. Second, and more important, @ quantities do not necessarily translate directly into # quantities.

The reason for this is simple. Some @ quantities are mutually exclusive. In the milk example, as our case in point, the consumption of 0 might have been indicated by a punch in @05036, 1 quart by a punch in @05037, 2-4 quarts by @05038 and more than 4 quarts by @05039. Clearly, exactly one of these 4 possible punches must be present. While this requires 4 bits of information in the original input data structure, only two bits are really needed to represent the quantity. In this case two # quantities, say #00135 and #00135 are assigned and the elementary translation:

FROM:				TO:	
@05036	@05037	@05038	@05039	#00134	#00135
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

FIGURE 1

is performed. All information necessary to perform this task is contained in a directory called "ATSIGN." This directory is accessed by the periodic system in order to decide how to encode the information, and it is accessed by the interactive system to decide where to find something and how to decode it.

II. The Interactive System

In this section, we describe the interactive system, together with examples of its use. The primary capability of the system can be described as three-dimensional cross-tabulation. The user must specify several criteria as follows:

a) NP population "bases" (1, . . ., K, . . ., NP). A report (two-dimensional cross-tab) will be produced for each base. common examples might be, MEN, WOMEN and ALL.

b) NR "rows" (1, . . ., I, . . ., NR). A description of the composition of each row in a report.

c) NC "columns" (1, . . ., J, . . ., NC). A description of the composition of each column in a report.

Each of these NP + NR + NC criteria is a logical expression having, for each respondent, either the value "true" or "false." In effect, the system must scan all respondents, preparing NP tables as shown in Figure 2.

For a given base, say K, the GRAND TOTAL is the number of respondents (or more likely, a quantity called the 'weighted sum') who satisfied the criterion specified for that base.

The COLUMN J TOTAL in table K is the number of respondents who simultaneously met base criterion K and column criterion J. Row totals are similarly generated.

$S_{I,J,K}$ is the number of respondents who simultaneously met base criterion K, column criterion J. and row criterion I.

The user describes his desired cross-tabulation in an input language which we now describe.

Standard Input Language

The most important part of a cross-tab definition is the set of criteria defining the bases, rows, and columns. In general, each is defined as a Boolean expression made up of logical variables, Boolean operators, and parentheses.

Operators

The operators available in the system are:

' or .NOT.	negation
& or .AND.	logical product
! or .OR.	inclusive or
+ or .EQ.	equivalence

Expressions are evaluated left-to-right, with an automatic operator hierarchy in the same order as shown above. Parenthesization is allowed to any depth.

Logical Variables

We have already described the two lowest level logical variables, the #-entity and the @-entity. The #-entity represents a particular bit in the data as actually stored, while the @-entity represents a particular punch position as it would appear on a card. It obviously can be either true or false.

The next level of logical variables allowed is the in-expression, which is closely related to the set-theoretic notation a A, meaning a is an element contained in the set described by A.

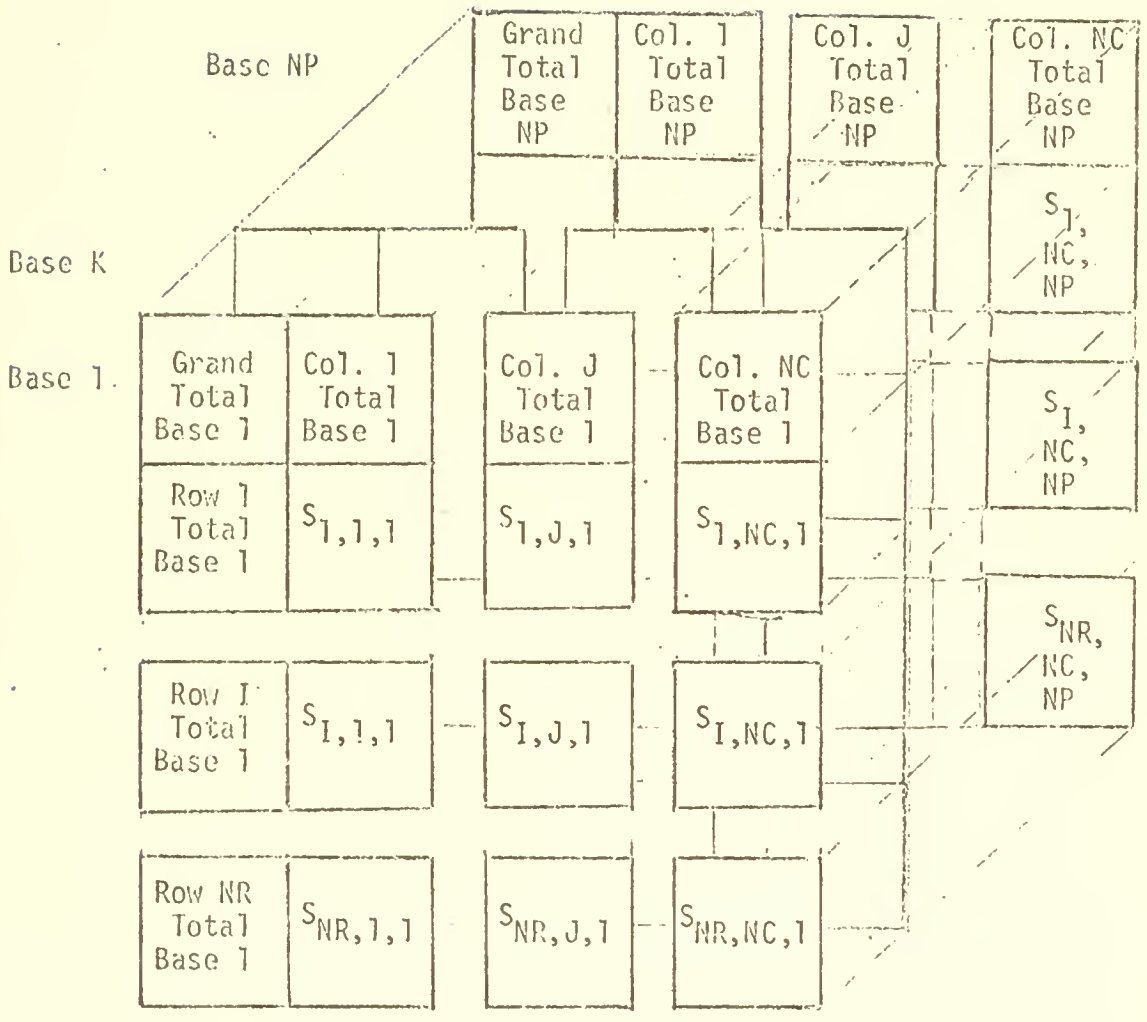

FIGURE 2

Tables for Bases 1 . . . NP

The general form of the in-expression is string IN (set), where string is any string of alphanumeric characters and set is one or more expressions of the forms:

i	meaning equal to integer i
=i	meaning equal to integer i
<=i	meaning less than or equal to i
<i	meaning less than i
>=i	meaning greater than or equal to i
>i	meaning greater than i
i:j	meaning between integers i and j (inclusive).

The expressions are set off by commas, implying inclusive or.

Thus: HH INC IN(<5000,>= 75000) is an in-expression which is true if and only if the income of the respondent's household (HH INC) is less than $5000 or greater than or equal to $75000.

Synonyms

There are two classes of synonyms available in the user language. One has the very simple form of substituting a logical expression for an arbitrary string of characters. For example, the string MEN becomes the logical expression @01091. The string RICH WOMEN might be defined as @01092 .AND. HH INC IN(> = 75000). No recursion is allowed, so the logical expressions must be in terms of #, @, and in-expressions. In other words, a synonym cannot be defined in terms of other synonyms.

The other class of synonyms is used for reference to data about media seen during the weeks preceding the two interviews which comprise

the questionnaire response. The general form of this class is

$$R_pMED_c$$

Where R is the letter R;

p is the week indicator

p = X read first week

p = Y read second week

p = 0 read neither

p = 1 read exactly one of the two weeks

p = 2 read both weeks

MED is a 3-character media code; and c is a where-read code

c = blank or null - anywhere

c = H at home

c = O at office, etc.

Thus RXTIMH is true if and only if this respondent saw TIME magazine at home (in week #2).

In order to define a table, the user engages in a dialog with the system. One such dialog is shown in Figure 3, with user responses underlined. Figure 4 shows this console session recorded in a disk file in case the user wishes to run the table again at a later date. (The sophisticated user may by-pass the console session and simply create a file of the form shown in Figure 4.)

Figure 5 shows the user request as translated into the lowest-level (#-sign) language, ready for retrieval. Only those data represented by the #-sign entities shown in Figure 5 need be retrieved. This happens to be a two-dimensional table request, since only one base is called for.

Table Generation

Actual building of a two-dimensional table is conceptually very simple. A table of NR x NC elements is set to zero. Then NR+NC+1 words are read from the intermediate file. These correspond to respondents 1-36, for the population base, NR rows, and NC columns. Starting with the left most bits of these words, we form another array of NRxNC, each element (I,J) being pop base bit .AND. row I bit .AND. column J bit. This array is then added to the current table. This proceeds through 36 cases, after which another NR+NC-1 words are read, corresponding to respondents 32-72, 73-108, etc.

This process terminates on end of the intermediate file, with the original array containing counts as described in section A of this paper.

Many users want their counts weighted to correspond to the size of the actual population, and to correct the over-representation of heavy consumers. Each respondent has several weights,* and depending on which is requested, the logically corresponding weights are, as described above, added to the array.

---

*How many people he represents, given the General Panel, Heads of Households, etc.

In fact, we build two arrays in every run. One is weighted as required by the user. The other is unweighted. This allows final reports which we generate to identify cells in the table whose low number of respondents suggests statistical unreliability.

Figure 2 shows an extra row and column labeled "total." These are obtained by defining an internal row 0 and an internal column 0, both always "true." They are not, in fact, totals, but rather counts of all these cases which meet the population and the row (column) criteria alone. In other words, the "total" for row I would be equivalent to the sum of the elements in row I if and only if the column definitions were both mutually exclusive and collectively exhaustive. These are the numbers most frequently needed by users, but sometimes summations of elements are more appropriate. In this case, the display phase (below) computes them.

The Display Phase

The display phase takes the output of the table generator and prints it out. Its tasks are simple: generate row and column totals, if required; calculate ratios of elements to row, column or grand totals if requested; formatting of output numbers, stubs and headers; and segmentation of the table as as to fit (physically) on the paper.

Figure 6 shows the output of the session begun in Figure 3.

```
RUN DSK IMS
WELCOME TO IMS AT 16:37:5 ON 11-AUGUST-1970.
ENTER YOUR JOB KEY: *CRS
PASSWORD:  *

NEW,OLD,METHR,RERUN-METHR, OR BETA? *NEW
DO YOU WISH A TITLE?
*Y
HOW MANY LINES (1-9)?
*1
1:
*TEST RUN FOR PAPER
ANY CHANGES TO THE TITLE?
*N
NUMBER OF ROWS?
*2
NUMBER OF COLUMNS?
*4
REPORT IS 2 ROWS AND 4 COLUMNS.
IS THAT CORRECT?
*Y
DO YOU HAVE A PERMANENT DEFINITION FILE TO USE?
*N
DO YOU HAVE ANY TEMPORARY DEFINITIONS?
*Y
SYMBOL: EXPRESSION;
ONE BLANK LINE TERMINATES.
*A:HH INC IN (<5000);
*
ANY CHANGES TO TEMPORARY DEFINITIONS?
*N
DEFINE THE BASE.
*RXTIM. OR:RYTIM;
CHANGE BASE?
*N
ROW DEFINITIONS.
SYMBOL=STUB: EXPRESSION;
ROWO1=
*POOR MEN; A.AND. MEN;
ROWO2=
*POOR WOMEN; A,AND,WOMEN;
ANY CHANGES?
*N
COLUMN DEFINITIONS.
SYMBOL=HEADER: EXPRESSION;
COLO1=
```

FIGURE 3

An Example of a Console Session

FIGURE 3 (CONTINUED)

```
*18 TO 19:AGE IN(18:19);
COLO2=
*20 TO 29;AGE IN(20:29);
COLO3=
*30 TO 39:AGE IN(30:39);
COLO4=
*40 UP:AGE IN( =40);
ANY CHANGES?
*N
SAMPLE AND WEIGHTING: ('HELP' FOR INSTRUCTIONS)
*POP
RECORDED AS:
WGTD: POP
ANY PERCENTAGE OPTIONS?
*Y
CELL CONTENTS:
VERTICAL PERCENT?
*N
HORIZONTAL PERCENT?
*N
PERCENT OF GRAND TOTAL?
*Y
INDEXED BY ROW TOTAL?
*N
INDEXED BY COLUMN TOTAL?
*N
SUMMATIONS (AS OPPOSED TO 'TOTALS')?
*N
ENTERING FILE STRUCTURE ANALYSIS.
WOULD YOU LIKE TO SAVE THIS FILE? *Y
ENTER FILE NAME: FIGURE 4
NO ERRORS DETECTED IN THIS PHASE.
```

```
STANDARD CROSSTAB.
01
                                        TEST RUN FOR PAPER
02
04
A:HH INC IN(<5000);

SMPOP: RXTIM.OR.RYTIM;
ROW01=POOR MEN:A.AND.MEN;
ROW02=POOR WOMEN:A.AND.WOMEN;

COL01=18 TO 19:AGE IN(18:19);
COL02=20 TO 29:AGE IN(20:29);
COL03=30 TO 39:AGE IN(30:39);
COL04=40 UP:AGE IN(>=40);

WGTD: POP
NBCT: NO
NBRT: NO
NBGT: YES
RBRT: NO
RBCT: NO
MODE: NO
END:  YES
```

FIGURE 4

Intermediate Text File Generated
by the Session Shown in Figure 3

```
02
04
((#00453)).OR.((#00454));
(('#00174&'#00175&'#00176&#00177)!('#00174&'#00175&#00176&'#00177)) .AND
. (('#00059&#00060));
(('#00174&'#00175&'#00176&#00177)!)'#00174&'#00175&#00176&'#00177)) .AND
.((#00059&'#00060));
(('#00182&'#00183&'#00184&#00185));
(('#00182&'#00183&#00184&'#00185)!('#00182&'#00183&#00184&#00185));
(('#00182&#00183&'#00184&'#00185)!('#00182&#00183&'#00184&#00185));
(('#00182&#00183&#00184&'#00185)!('#00182&#00183&#00184&#00185)!(#00182&
'#00183&'#00184&'00185)!(#00182&'#00183&'#00184&#00185)!(#00182&'#00183&
#00184&'#00185)!(#00182&'#00183&#00184&#00185)) ;


EXIT
'C
```

FIGURE 5

Intermediate File Generated by the System Describing the Population, Rows and Columns in Full #-expressions. This is the result of the phase described in Section E operating on the input shown in Figure 4.

TABLE GENERATION IS COMPLETE.

TEST RUN FOR PAPER

'TOTALS'	'TOTALS'	18 TO 19	20 TO 29	30 TO 39	40 UP
	26449	2607	6421	4905	12517
GRTO%	100.00	9.86	24.28	18.55	47.33
POOR MEN		**	*	**	
	1474	134	230	179	931
GRTO%	5.57	0.51	0.87	0.68	3.52
POOR WOMEN		**	**	**	
	1854	83	140	253	1377
GRTO%	7.01	0.31	0.53	0.96	5.21

FIGURE 6

Application of Interactive System to Social Science Research

The major advantage of this and similar systems to research using similar data bases is one of cost. Response time is also a factor, but the cost advantage is very large. Since the researcher typically works with very few facts about many respondents, the marginal cost of a cross-tabulation can be reduced by a large factor (typically at least 10, in our experience). What is needed is a way of paying the fixed costs of inversion and on-line storage of data. The retrieval technology is available today.

We illustrate the use of the system to retrieve a cross-tabulation involving total household income and source of other income in Figures 7 (table definition) and 8 (results). These data were drawn from the 1971 panel offered by W. R. Simmons and Associates Research, Inc.

FIGURE 7

```
STANDARD CROSSTAB.
02
07
TOTAL HOUSEHOLD INCOME VERSUS SOURCES OF NON-EMPLOYMENT INCOME
FOR MALE RESPONDENTS 71 SIMMONS FULL SAMPLE. POPULATION WEIGHTS
PREPARED FOR:
"APPLICATION OF A FLEXIBLE SYSTEM TO RETRIEVE, MANIPULATE, AND
DISPLAY INFORMATION FROM A STABLE, QUESTIONNAIRE-ORIENTED DATA
BASE TO SOCIAL SCIENCE RESEARCH"
BY CHRISTOPHER R. SPRAGUE AND DAVID NESS
01
TABLE AS ABOVE FOR FEMALE RESPONDENTS
14
04

BAS01:MEN;
BAS02:WOMEN;

ROW01=NONE:00150Y;
ROW02=SOC SEC:0 01491;
ROW03=UNEMP:001492;
ROW04=WELFARE:001493;
ROW05=PENSION:001494;
ROW06=SAV ACCT:001495;
ROW07=DIVDS:001496;
ROW08=RENT:001497;
ROW09=MTGS:001498;
ROW10=INHERIT:0001499;
ROW11=BONDS:00149Ø;
ROW12=ANNUITY:00149X;
ROW13=STK MKT:00149Y;
ROW14=OTHER:001501;

COL01=<5:HH INC IN(<5000);
COL02=5-10:HH INC IN (5000:9999);
COL03=10-20:HH INC IN (10000:19999);
COL04=20 UP:HH INC IN (>=20000);

WGTD: 71SIMPOP
NBCT: YES
NBRT: NO
NBGT: YES
RBRT: NO
RBCT: NO
MODE: NO
END:  YES

EXIT
'C
```

FIGURE 8

```
.RU IMS
WELCOME TO IMS AT 13:25:34 ON 18-OCTOBER-1971.
ENTER YOUR JOB KEY: *CRS
PASSWORD: *

9/23: DATA OFF-LINE: 70SIM, 70BRI CARDS E,R,H,J-M.

ENTER TABLE TYPE*OLD
FILE: PAPER.TST

INPUT ANALYSIS PHASES
1971 SIMMONS FULL SAMPLE, WEIGHTED BY POP

PHASES 20, 2A, 2B, 2C, 2D, 2Y, 2E, 3 COMPLETE
TABLE GENERATION IS .. COMPLETE.
```

TOTAL HOUSEHOLD INCOME VERSUS SOURCES OF NON-EMPLOYMENT INCOME
FOR MALE RESPONDENTS 71 SIMMONS FULL SAMPLE, POPULATION WEIGHTS
PREPARED FOR:
"APPLICATION OF A FLEXIBLE SYSTEM TO RETRIEVE, MANIPULATE, AND
DISPLAY INFORMATION FROM A STABLE, QUESTIONNAIRE-ORIENTED DATA
BASE TO SOCIAL SCIENCE RESEARCH"
BY CHRISTOPHER R. SPRAGUE AND DAVID NESS

	'TOTALS'	<5	5-10	10-20	20 UP
'TOTALS'					
	60744	11681	19563	23072	6428
VERT%	100.00	100.00	100.00	100.00	100.00
GRTO%	100.00	19.23	32.21	37.98	10.58

NONE	2425Ø	3773	1Ø233	8797	1448
VERT%	39.92	32.3Ø	52.31	38.13	22.53
GRTO%	39.92	6.21	16.85	14.48	2.38
SOC SEC					
	11631	5616	3391	2Ø99	525
VERT%	19.15	48.Ø8	17.33	9.1Ø	8.17
GRTO%	19.15	9.25	5.58	3.46	Ø.86
UNEMP		**	*	*	**
	584	212	289	377	6
VERT%	1.46	1.81	1.48	1.63	Ø.Ø9
GRTO%	1.46	Ø.35	Ø.48	Ø.62	Ø.Ø1
WELFARE			**	**	**
	1855	1334	3Ø6	194	2Ø
VERT%	3.Ø5	11.42	1.56	Ø.84	Ø.31
GRTO%	3.Ø5	2.2Ø	Ø.5Ø	Ø.32	Ø.Ø3
PENSION					
	5845	2163	1832	1333	516
VERT%	9.62	18.52	9.36	5.78	8.Ø3
GRTO%	9.62	3.56	3.Ø2	2.19	Ø.85
SAV ACCT					
	2Ø735	1518	5Ø41	1Ø3Ø5	3871
VERT%	34.14	13.ØØ	25.77	44.66	6Ø.22
GRTO%	34.14	2.5Ø	8.3Ø	16.96	6.37
DIVDS		*			
	91Ø2	42Ø	1293	4649	274Ø
VERT%	14.98	3.6Ø	6.61	2Ø.15	42.63
GRTO%	14.98	Ø.69	2.13	7.65	4.51
RENT		*			
	48Ø1	595	998	2123	1Ø85
VERT%	7.9Ø	5.Ø9	5.1Ø	9.2Ø	16.88
GRTO%	7.9Ø	Ø.98	1.64	3.49	1.79
MTGS		**	**		
	1118	68	16Ø	489	4Ø1
VERT%	1.84	Ø.58	Ø.82	2.12	6.24
GRTO%	1.84	Ø.11	Ø.26	Ø.81	Ø.66
INHERIT		**	**	*	
	679	58	66	258	299
VERT%	1.12	Ø.5Ø	Ø.34	1.12	4.65
GRTO%	1.12	Ø.1Ø	Ø.11	Ø.42	Ø.49
BONDS		**	**		
	1525	47	219	7Ø3	557
VERT%	2.51	Ø.4Ø	1.12	3.Ø5	8.67
GRTO%	2.51	Ø.Ø8	Ø.36	1.16	Ø.92
ANNUITY		**	*		
	1827	258	338	7Ø8	523

VERT%	3.Ø1	2.21	1.73	3.Ø7	8.14
GRTO%	3.Ø1	Ø.42	Ø.56	1.17	Ø.86
STK MKT		**	*		
	196Ø	38	2Ø6	925	791
VERT%	3.23	Ø.33	1.Ø5	4.Ø1	12.31
GRTO%	3.23	Ø.Ø6	Ø.34	1.52	1.3Ø
OTHER					
	3623	66Ø	124Ø	1349	375
VERT%	5.96	5.65	6.34	5.85	5.83
GRTO%	5.96	1.Ø9	2.Ø4	2.22	Ø.62

TABEL AS ABOVE FOR FEMALE RESPONDENTS

	'TOTALS'	5	5-1Ø	1Ø-2Ø	2Ø UP
'TOTALS'					
	66751	16962	22Ø47	22757	4987
VERT%	1ØØ.ØØ	1ØØ.ØØ	1ØØ.ØØ	1ØØ.ØØ	1ØØ.ØØ
GRTO%	1ØØ.ØØ	25.41	33.Ø3	34.Ø9	7.47
NONE					
	24593	3842	1Ø736	89Ø4	1111
VERT%	36.84	22.65	48.7Ø	39.13	22.28
GRTO%	36.84	5.76	16.Ø8	13.34	1.66
SOC SEC					
	16323	9Ø2Ø	4Ø5Ø	2481	772
VERT%	24.45	53.18	18.37	1Ø.9Ø	15.48
GRTO%	24.45	13.51	6.Ø7	3.72	1.16
UNEMP		*	*	*	**
	946	246	465	198	36
VERT%	1.42	1.45	2.11	Ø.87	Ø.72
GRTO%	1.42	Ø.37	Ø.7Ø	Ø.3Ø	Ø.Ø5
WELFARE			*	**	**
	3881	3Ø2Ø	6Ø2	158	1Ø2
VERT%	5.81	17.8Ø	2.73	Ø.69	2.Ø5
GRTO%	5.81	4.52	Ø.9Ø	Ø.24	Ø.15
PENSION					
	5907	2193	22Ø6	1Ø9	4ØØ
VERT%	8.85	12.93	1Ø.Ø1	4.87	8.Ø2
GRTO%	8.85	3.29	3.3Ø	1.66	Ø.6Ø
SAV ACCT					
	223Ø6	2944	6Ø2Ø	1Ø178	3164
VERT%	33.42	17.36	27.31	44.72	63.44
GRTO%	33.42	4.41	9.Ø2	15.25	4.74

DIVDS					
	8969	759	2Ø2Ø	3873	2318
VERT%	13.44	4.47	9.16	17.Ø2	46.48
GRTO%	13.44	1.14	3.Ø3	5.8Ø	3.47
RENT					
	5996	1147	1374	2469	1ØØ6
VERT%	8.98	6.76	6.23	1Ø.85	2Ø.17
GRTO%	8.98	1.72	2.Ø6	3.7Ø	1.51
MTGS		**	**		
	1246	132	175	6Ø4	335
VERT%	1.87	Ø.78	Ø.79	2.65	6.72
GRTO%	1.87	Ø.2Ø	Ø.26	Ø.9Ø	Ø.5Ø
INHERIT		**	*		
	1115	9Ø	2Ø3	5Ø4	317
VERT%	1.67	Ø.53	Ø.92	2.21	6.36
GRTO%	1.67	Ø.13	Ø.3Ø	Ø.76	Ø.47
BONDS		**	*		
	12Ø5	2Ø4	183	5Ø8	31Ø
VERT%	1.81	1.2Ø	Ø.83	2.23	6.22
GRTO%	1.81	Ø.31	Ø.27	Ø.76	Ø.46
ANNUITY		*			
	2495	356	549	111Ø	481
VERT%	3.74	2.1Ø	2.49	4.88	9.65
GRTO%	3.74	Ø.53	Ø.82	1.66	Ø.72
STK MKT		**	*		
	1578	61	335	543	639
VERT%	2.36	Ø.36	1.52	2.39	12.81
GRTO%	2.36	Ø.Ø9	Ø.5Ø	Ø.81	Ø.96
OTHER					
	3362	719	1156	1236	251
VERT%	5.Ø4	4.24	5.24	5.43	5.Ø3
GRTO%	5.Ø4	1.Ø8	1.73	1.85	Ø.38