The multiple variate counter

By Andrew Colin

The body of this paper is concerned with a description of M.V.C. Mark 5, a special-purpose language for General Survey Analysis. The paper ends with some remarks about the implementation of the language on the I.C.T. (Ferranti) Atlas.

In designing the M.V.C. language, the author has attempted to combine the flexibility and convenience usually associated with a modern programming language with the special requirements of survey analysis. The M.V.C. language is used to describe surveys to the analyzing computer. Its elements include numbers, free names for "variables," arithmetical and logical connectives, and a considerable number of special "system words," which are used to describe the various situations and operations peculiar to survey analysis.

At a superficial syntactic level, the M.V.C. language resembles ALGOL 60. For example, system words are underlined,* statements end with a semicolon, and "space" and "newline" are generally ignored. This similarity was deliberately introduced to avoid the promulgation of yet another punching convention for programs.

The description of the language contained in this paper is necessarily brief and incomplete. For a full specification, the reader is referred to the *M.V.C. Manual*, obtainable on application to the author.

Survey data

Survey data differs from other types of information usually presented to computers in that it is often somewhat ill-structured. The body of data to be analysed consists of blocks of information describing each of a number of "cases." The data structure within each case is likely to be variable; for example, certain cases may belong to a special sub-group for which additional data is supplied; or, in certain types of survey, each case may include data on a variable number of "subcases" which require some independent treatment.

The basic format in which the data is collected is also variable. Some questions in a survey questionnaire may require numerical answers; others imply a "yes/no" reply, while again others ask for the indication of exactly one out of a list of several possible answers. It frequently happens, particularly in commercial surveys, that the "respondent" fails to answer some of the questions.

Lastly, the physical presentation of the data to the computer may be made in a large number of ways. The data may be coded (in some arbitrary fashion) on to punched cards, punched on to paper tape, or recorded on magnetic tape by a different computer. All the data relevant to one case may be collected together in one block, or be scattered among several stacks of cards or tapes.

This variability in the way that data is presented is partly due to the unfortunate habit of many survey analysts (particularly in the academic field) of collecting and coding their data before deciding on the method of analysis.

Data editing

The task of a General Survey Analysis system is a two-fold one. Its first, and perhaps most important part is to absorb data in any of an almost infinite variety of forms, and to reduce it to a standard format, suitable for tabulation and other statistical purposes. The second part of the task, which will be described later on in the paper, is concerned with actually using the standardized data for the derivation of statistics.

The M.V.C. system assumes that all the data relevant to any one case is available at one time.* Translation of the data into standard form is specified in the M.V.C. language by means of *raw definitions*. Each raw definition creates a *raw variable*, assigns it an arbitrary name, and links it to some specific region of the case record. The set of raw definitions in an M.V.C. survey description is, in general, applied to each case in the survey, producing an *array* of variable values which describe that case in standard format.

In this format, there are three types of variable.

(*a*) The *numerical* variable, which takes numerical values.

(*b*) The *binary* variable, which takes the logical values T (for "true") and F (for "false").

(*c*) The *polylog* variable,† which may take one out of several possible values, which have no properties other than that of being distinct and identifiable by name.

In addition, all variables may take the special value *unknown*.

These types are used because it appears that the vast majority of questions asked in surveys correspond closely to one or other of the three. It is, of course, possible to code all survey data as numbers; but it seems unnatural to force a numerical format on to data which is essentially non-numeric in character. Why should it

* Or printed in **bold** type, for typographical convenience.—Ed.

* Survey data is generally supplied in this form. If not, it is usually possible to arrange the required presentation by pre-sorting cards, or by reading several tapes simultaneously.

† Commonly known as an *attribute*.—Ed.

be necessary to attach the integers 1, 2, 3, . . . to—say—religions or diseases?

The way in which raw data is operated on by definitions to produce values for named variables is best shown by an example. The raw definitions given below assume that the data has been punched on to 80-column cards. If the data were punched on tape, a slightly different definition format would have been used.

input cards;
raw numerical
AGE, 1. 2;
INCOME, 3, 5;
NUMBER OF CHILDREN, 8, 1;
YEARS IN PRESENT JOB, 9, 2;

raw binary
DEGREE, 11/0;
OWN CAR, 11/1;
OWN HOUSE, 11/2;

raw polylog
SEX, 12/1, 2, *MALE, FEMALE*;
MARITAL STATUS, 13/1, 4, *SINGLE, MARRIED, DIVORCED, WIDOWED*;
POLITICS, 14/1, 7 *TORY, LIB, LAB, COMM, OTHER, DONT KNOW, WONT SAY*;

The import of these definitions is this:

AGE, INCOME, NUMBER OF CHILDREN, and *YEARS IN PRESENT JOB* are names given to numerical variables. *AGE* is associated with the value punched in the two columns starting at column 1 of the punched card; similarly, *INCOME* is assigned the value punched in the five columns starting at column 3.

DEGREE, OWN CAR, and *OWN HOUSE* are all binary variables. *DEGREE* takes the value **T** if hole site 0 in column 11 is punched; otherwise it takes the value **F**. The values of the other binary variables are similarly associated with the presence or absence of holes at the specified hole sites.

SEX, MARITAL STATUS, and *POLITICS* are the names assigned to the polylog variables. Each variable is connected to a field of consecutive hole sites on the card, only one of which may be punched; and the value given to the polylog in any one case corresponds to the position of the punched hole. The definition of a raw polylog variable includes the variable name, the position of the beginning of the field, the number of possible answers (or the number of hole sites in the field) and a list of names for the answers. For example, if hole site 4 in column 14 is punched, the variable *POLITICS* is assigned the value *COMM*.

If punching is in any sense ambiguous, the corresponding variable is given the value "unknown."

To give an example, Table 1 below is a list of the holes punched in two possible case-cards to which these definitions are applied, and Table 2 shows the values to which these case-cards give rise.

Table 1

HOLE SITE POSITIONS		
COLUMN NUMBER	CASE 1	CASE 2
1	2	4
2	9	7
3	0	0
4	0	2
5	9	5
6	8	0
7	5	0
8	0	4
9	0	2
10	4	1
11	0 and 1	1 and 2
12	2	1
13	1	2
14	1	2

Table 2

VARIABLES	VALUES	
	CARD 1	CARD 2
AGE	29	47
INCOME	985	2,500
NUMBER OF CHILDREN	0	4
YEARS IN PRESENT JOB	4	21
DEGREE	**T**	**F**
OWN CAR	**T**	**T**
OWN HOUSE	**F**	**T**
SEX	*FEMALE*	*MALE*
MARITAL STATUS	*SINGLE*	*MARRIED*
POLITICS	*TORY*	*LIB*

It will be seen that each of the raw variables described above corresponds to one question in a questionnaire. Frequently, questionnaires contain blocks of closely related questions; such as, for example, a student's marks for each week of a twelve-week term, or a consumer's reaction to each of a number of similar products. The M.V.C. system allows such groups to be handled as one-dimensional vectors, with a common vector name. For instance, the twelve weekly scores could be known as *MARKS* [1], *MARKS* [2], . . . *MARKS* [12], the general vector name being *MARKS*.

To simplify the handling of vectors, special indices are provided. There are 26 in the system, known by the names *A*, *B*, *C*, . . ., *Z*. (These identifiers are reserved for this purpose, and may not be used as variable names.)

The values of indices are controlled by *index setting statements*, which take the form:

$A := 1, 3, 14, 97;$

or $B := 5[12]17;$

or $C := 8[7];$

or $D := 1, 3(3)15, 14[9], 8, 5, 1(1)7;$

Each index setting statement defines a sequence of values which the index named on the left is to take. The construction $a(b)c$ defines the arithmetical progression $a, a + b, a + 2b, \ldots c - b, c$, and $d[e]$ means d repeated e times.

Index statements are used in *blocks* of one or more. They act simultaneously, thus defining a sequence of values for a vector of indices. Naturally, the number of values defined by each index setting statement in a block must be the same.

Index setting statements are used in the following construction:

for < block of index setting statements > **begin** < portion of M.V.C. text > **end**

The portion of M.V.C. text uses indices instead of integers at various points. The import of the entire construction is that the text is "lexicographically" repeated, once for each value of the index vector.

To illustrate the use of the vector facility, suppose that part of the data in a survey are a student's marks in each of six subjects. The maximum mark in each subject is variable, and is listed below, together with a possible card-coding convention for punching the actual marks gained.

SUBJECT NUMBER	MAX. MARKS	PUNCHING CONVENTION
Subject 1	150	Cols. 8, 9, 10
Subject 2	500	Cols. 11, 12, 13
Subject 3	1,000	Cols. 14, 15, 16, 17
Subject 4	5	Col. 18
Subject 5	25	Cols. 19, 20
Subject 6	200	Cols. 21, 22, 23

If vectors were not available, we would define variables for the student's marks as follows:

SCORE [1], 8, 3; (i.e. 3 cols. starting at col. 8)
SCORE [2], 11, 3;
SCORE [3], 14, 4;
SCORE [4], 18, 1;
SCORE [5], 19, 2;
SCORE [6], 21, 3;

With the aid of vectors, however, this can be shortened to:

vector *SCORE* [1, 6];
for $X := 1(1)6;$
$Y := 8, 11, 14, 18, 19, 21;$
$Z := 3, 3, 4, 1, 2, 3;$ **begin**
SCORE [*X*], *Y*, *Z*; **end** (i.e. *Z* columns starting at column *Y*).

Data transformations

Many survey analyses require that various transformations be applied to raw data before it is used for analysis. For example, numerical values may need to be grouped, or various combinations of binary and polylog answers collected together. These transformations, together with certain other functions to be described below, are carried out with the aid of *expressions*.

Expressions are of two types: *arithmetical*, and *Boolean*. Arithmetical expressions are constructed in a manner roughly similar to that encountered in ALGOL 60. Operands may be numbers, numerical variables, or indices. As there is nothing comparable to the ALGOL procedure mechanism, a short selection of "built-in" functions is provided. These are indicated by suitable system words.

Boolean expressions, on the other hand, are constructed with the operators **and, or,** and **not**, and *Boolean Particles*. Each particle may take the value **T** or **F**, and may be one of three types of entity:

(*a*) A binary variable.
(*b*) A polylog answer name. Such a particle takes the value **T** if the name corresponds to the current value of the polylog variable, and **F** otherwise. (For example, if *POLITICS* = *TORY*, then *TORY* = **T**, and *LIB* = **F**).
(*c*) A relation, such as >, =, or ≠, between two arithmetical expressions. (For example, if *HEIGHT* = 69, then (*HEIGHT* < 71) = **T**, and (*HEIGHT* ≠ 69) = **F**.)

Transformed survey data is expressed in terms of *derived variables*. These are also of the three basic types. Numerical and Binary derived variables are defined in terms of arithmetical and Boolean expressions, respectively. Derived polylog variables are defined by sequences of Boolean expressions. The value assigned to a derived polylog for any one case corresponds to the first of the sequence of expressions found, on evaluation, to have the value **T**. Naturally, expressions used in defining derived variables must only include constants and variables whose values are already known.

Some examples of derived definitions are:

derived numerical
AGE OF STARTING JOB := *AGE* − *TIME IN PRESENT JOB*;
SKIN AREA := 71·84 × **exp**(0·425 × **log**(*WEIGHT*) + 0·725 × **log**(*HEIGHT*));

derived binary

FAMILY MAN := *MALE* **and** (*MARRIED* **or** *WIDOWED*) **and** (*NUMBER OF CHILDREN* $\neq 0$);

derived polylog

AGE GROUP := (*AGE* $<$ 30), (*AGE* $<$ 60), (*AGE* $\geqslant$ 60), *YOUNG*, *MIDDLEAGED*, *OLD*;

In this group, the second definition is a coding of Duboys' formula $S = 71{\cdot}84\ W^{0{\cdot}425} \times H^{0{\cdot}725}$. The fourth definition assigns the polylog variable *AGE GROUP* one of the three values *YOUNG*, *MIDDLEAGED*, or *OLD*, according to the value of the numerical variable *AGE*.

The treatment of variable data formats

The problem of analyzing surveys in which the amount of data per case is variable is solved by the simple device of defining sufficient variables to cover every possibility, and then assigning the value 'unknown" to variables which are superfluous in any particular case. When a block of data is known (through the values of previous variables) to be absent, its definitions can be omitted by means of *definition skipping statements*. These consist of the system word **unless,** a Boolean expression, and the name of the definition to which the system is to jump. For example, consider the following group of questions.

(1) Do you play chess?
  If so, how many games per week do you play?
  Do you consider yourself (*a*) a good player?
  or (*b*) a medium player?
  or (*c*) a bad player?
  At what age did you learn the game?

(2) Do you play bridge?
  Etc.

The raw definitions associated with these questions might run as follows:

raw binary
CHESS, 56/0;
unless *CHESS* **go to** *BRIDGE*; (i.e. skip certain definitions unless *CHESS* = **T**)
raw numerical
GAMES PER WEEK, 57, 2;
AGE LEARNED CHESS, 60, 2;
raw polylog
CHESS PROFICIENCY, 59/1, 3, *GOOD*, *MEDIUM*, *POOR*;
raw binary
BRIDGE, 62/0;
. . . . . .

Checks on data

Survey data is intrinsically liable to errors in compilation and blunders in coding, so it is usually advisable to apply some kind of check to it before using it for statistical purposes. Verifications of this type can be specified by *checking statements*. Each statement consists of the word **accept** followed by a list of Boolean expressions which must all have the value **T** if that case is to be used in the analysis. An alternative form, consisting of the word **reject** followed by a list of expressions which must all have the value **F**, is also available.

At this stage in analysis, cases are not automatically rejected if they contain unknown values; this may be quite normal in certain types of survey. If it is essential, in the succeeding analysis, for certain values to be known, rejection may be arranged for by using the special "existence" function, **E**(*x*). This function has the property that

E (any known variable) = **T**,
and **E** (any variable whose value is unknown) = **F**.

Some examples of checking statements are:

accept (*AGE* $>$ 20);
reject (*TIME IN PRESENT JOB* $>$ *AGE* $-$ 13);
accept **E** (*MARITAL STATUS*), **E** (*POLITICS*), **E** (*NUMBER OF CHILDREN*);

These checking statements ensure that all persons whose cases are actually analyzed are over 20 years of age, did not start their present jobs under the age of 13, and have replied to the questions relating to their marital status, political views, and number of children. Rejection of cases will be caused by punching errors, or lack of response or frivolous answers given by the persons concerned.

When a case is rejected, full details of the reason are supplied to the user.

Basic tabulation

As mentioned above, the final part of automatic survey analysis is the derivation of statistics from standardized input data. The simplest and most widely used method of analysis is that of sorting the cases presented into a number of groups or classes, counting the number in each class, and arranging the results in a "frequency table."

In the M.V.C. system, the generation of such tables is controlled by *table specifications*. Here, the classes to be counted are defined by Boolean expressions, called *table entries*. They are preceded by a *table heading*, which contains a title for the tabulation.

A simple table specification, and the table to which it might give rise, are shown below.

title *TABULATION OF CAR OWNERSHIP*;
count *MALE*, *FEMALE*,
  MALE **and** *OWN CAR*,
  MALE **and not** *OWN CAR*,
  FEMALE **and** *OWN CAR*,
  FEMALE **and not** *OWN CAR*;

TABULATION OF CAR OWNERSHIP	
TOTAL	1,798
MALE	1,001
FEMALE	797
MALE and OWN CAR	578
MALE and not OWN CAR	423
FEMALE and OWN CAR	312
FEMALE and not OWN CAR	485

In most surveys, frequency counts are required for a large number of simply-related classes. In order to avoid writing out an explicit Boolean expression for each class, the user may employ a number of shorthand devices. The simplest of these is the *table entry criterion*. When all the entries in a table have some common property, this property may be removed from the individual entries and put in the table heading preceded by the word **include**. For example, if we were interested in the car ownership of university graduates only, we could put:

title *CAR OWNERSHIP OF GRADUATES*;
include *DEGREE*;
count *MALE*, *FEMALE*, etc.

Another shorthand device is the use of a polylog **question** name as a table entry. Here, it is exactly synonymous to the list of its possible answers. For example, the specification:

title *POLITICS OF MARRIED MEN*;
include *MALE* **and** *MARRIED*;
count *POLITICS*;

would produce the table:

POLITICS OF MARRIED MEN	
MALE and MARRIED	
TOTAL	890
TORY	460
LIB	120
LAB	245
COMM	3
OTHER	5
DONTKNOW	49
WONTSAY	8

Perhaps the most powerful method for shortening table specifications is the *table split*, which allows multi-dimensional tables to be generated. Table splits are included in the table heading. They consist of the words **tabulate by,** followed by the name of a polylog question. The result is that the cases entering into that tabulation are divided into several groups according to the actual value of the polylog in the table split, and each group is tabulated independently. Thus, for example, we may put:

title *DISTRIBUTION OF HOUSE OWNERSHIP BY AGE*;
tabulate by *AGE GROUP*;
count *OWN HOUSE*;

This will generate the table:

DISTRIBUTION OF HOUSE OWNERSHIP BY AGE				
	YOUNG	MIDDLEAGED	OLD	TOTAL
TOTAL	973	819	705	2,497
OWN HOUSE	147	346	477	970
not OWN HOUSE	826	473	228	1,527

Using a table split, the first table specification given above would be condensed to

title *CONDENSED TABULATION OF CAR OWNERSHIP*;
tabulate by *SEX*;
count *OWN CAR*, **not** *OWN CAR*;

The result of this specification would be:

CONDENSED TABULATION OF CAR OWNERSHIP			
	MALE	FEMALE	TOTAL
TOTAL	1,001	797	1,798
OWN CAR	578	312	890
not OWN CAR	423	485	908

The table splits illustrated above both produced two-dimensional tables. The splits can be extended to generate tables of any dimension; for example, we may write:

tabulate by *AGEGROUP* **by** *SEX* **by** *POLITICS*;

This table split will generate a separate table for each possible combination of answers to the named polylogs. One such table, for example, refers to persons who are

YOUNG **and** *MALE* **and** *TORY*

A complete example of a multi-dimensional table-split would be too bulky to quote in this paper.

An M.V.C. survey description may include any number of different, independent table specifications. The tables or groups of tables corresponding to each one are generated simultaneously.

The ultimate fate of cases which contain unknown variables can now be considered. Whenever a case satisfies the entry criterion of a table specification, all the values of variables actually required are first collected, and examined. If these values include any which are unknown, the incrementation of that table only is skipped. This ensures that the maximum amount of information is extracted from cases with missing or wrongly punched answers. It means, however, that sometimes, in tables which should have the same number of entries, totals differ slightly. In commercial survey analysis, it is common practice either to "invent" values for unknown variables before tabulation, or to "adjust" tables with different totals by multiplying one of them by a number differing slightly from unity. In the author's opinion, both these practices are objectionable, because they generate no new information, but merely conceal from the user the poor quality of his data. Consequently, the M.V.C. system contains no mechanisms for this type of "correction."

In many cases, the automatic rejection of cases with unknown values serves a useful purpose. Thus, if in the course of a survey on spare-time activities, we wish to form a table about the habits of—say—chessplayers, the corresponding variables for those who do not play chess will be unknown, and will be rejected from the table.

More advanced statistics

As well as producing the basic frequency tables described in the previous section, the M.V.C. system has provision for generating more sophisticated statistics. In this direction the system is open-ended; although the only techniques to be incorporated in the immediate future are the simple ones described below, provision is made for the eventual inclusion of any standard process for which there is a demand.

The operations which it is proposed to include from the start fall into three groups:

(*a*) Summation of numerical data.
(*b*) Measures of association between pairs of variables.
(*c*) Operations on entire tables.

Summation of numerical data

The facilities available for this purpose are similar to those described in the section on 'basic' tabulation. They are intended for use on continuously variable data, as expressed by numerical, rather than binary or polylog variables, and generate sums, averages, and standard deviations.

Table specifications for such numerical statistics are closely allied to those used for frequency counts. The only difference is that instead of the word **count** followed by a list of Boolean expressions, they contain the words **sum, mean,** or **stdev,** followed by lists of arithmetical expressions. Table heading items, such as titles, table splits, and entry criteria remain unchanged. It is, in fact, permissible to mix frequency counts and numerical statistics in one table specification.

These facilities are illustrated in this table specification, and in the table which follows:

title *FAMILY SIZE OF GRADUATES AND NON-GRADUATES*;
include (*MARRIED* **or** *WIDOWED*);
tabulate by *DEGREE*;
mean *NUMBER OF CHILDREN*;
stdev *NUMBER OF CHILDREN*;

FAMILY SIZE OF GRADUATES AND NON-GRADUATES

	DEGREE	not DEGREE	TOTAL
TOTAL	117	1,389	1,506
mean NUMBER OF CHILDREN	3·72	3·51	3·53
stdev NUMBER OF CHILDREN	1·73	2·50	2·44

In this table, the "total" column contains weighted means of the other entries in the same line.

Measures of association

M.V.C. allows for simple tests of association between pairs of variables. For discrete variables (e.g. polylogs) the χ^2 test is used; for continuous variables, product moment correlation is available.

To compute the association between two polylog variables, we put, for example:

chisquare *AGEGROUP, MARITAL STATUS*;

Such a statement in a table specification causes the system to construct, internally, a suitable contingency table. When the table is complete, the "expected" frequency for each cell, and hence the value of χ^2, is computed. The "null hypothesis" used is that there is no association between the variables. The system has an internal table of significance levels, which is entered for the appropriate number of degrees of freedom, to obtain the probability p of the null hypothesis being true. Finally, the value of χ^2 is printed, together with one asterisk for $p < 0{\cdot}05$, two for $p < 0{\cdot}01$, and three for $p < 0{\cdot}001$.

In reality, every **chisquare** statement defines an array of χ^2 values. (The example given above is a degenerate case in which there is only one element.) If more than two polylog variables are specified, then association is measured for every pair, and the results arranged in a matrix of upper triangular form. When the **chisquare**

statement includes one, two, or three asterisks, the actual contingency table itself will be printed if $p < 0{\cdot}05$, $0{\cdot}01$ or $0{\cdot}001$, respectively. For example, the statement

chisquare *SEX, AGEGROUP, MARITAL STATUS, POLITICS****

will produce the χ^2 table:

	AGEGROUP	MARITAL STATUS	POLITICS
SEX	3·24	8·17*	11·91
AGE GROUP		645·11***	47·19**
MARITAL STATUS			12·30*

and the contingency table:

	SINGLE	MARRIED	WIDOWED	DIVORCED	TOTAL
YOUNG	497	318	4	25	844
MIDDLE-AGED	314	837	65	157	1,373
OLD	219	316	185	42	762
TOTAL	1,030	1,471	254	224	2,979

When association tests between every possible pair of polylog variables are not required, it is often more convenient for the results to be in the form of a rectangular matrix. This can be specified by including the word **by** in the **chisquare** statement. For example:

chisquare *FAMILY BACKGROUND, SEX, RACE* **by** *ACADEMIC SUCCESS, SPORTS RESULTS, BEHAVIOUR*;

would produce:

	ACADEMIC SUCCESS	SPORTS RESULTS	BEHAVIOUR
FAMILY BACK-GROUND	13·37*	22·70**	7·91
SEX	7·24	14·00**	8·99
RACE	5·74	35·89***	12·57*

In computing the value of χ^2, Yates' correction is automatically applied. If the "expected" value for any cell is less than 5, the calculated value of χ^2 for that table is unreliable, and a warning is printed. In many cases, however, it will be possible to avoid this difficulty by "lumping together" small classes by a suitable derived polylog definition. For example:

LUMPED RELIGION := (*PROTESTANT* **or** *CATHOLIC*), (*JEWISH* **or** *MUSLIM* **or** *HINDU* **or** *BUDDHIST*), *CHRISTIAN, NONCHRISTIAN*;

For continuous data, correlation matrices may be produced by a similar manner. For example, we may put

correlation *AGE, NUMBER OF CHILDREN, INCOME;*

From this statement we obtain

	NUMBER OF CHILDREN	INCOME
AGE	0·7835	0·6128
NUMBER OF CHILDREN		−0·3204

chisquare and **correlation** statements may be included in table specifications with normal table headings.

Operations on entire tables

The only facility involving the manipulation of entire tables and planned for immediate implementation is that of *percentaging*. By including the word **percent,** and a suitable qualifier in a table heading, a table may be percentaged with respect to its row totals, its column totals, its grand total, or the frequency of any named class.

In the initial version of M.V.C., percentages in tables are printed just below the raw totals used in computing them. It is, however, intended that one of the first additions to the system will be a facility for varying the output format of results in any desired way. This will enable percentages to be separated from their parent frequency tables.

The implementation of M.V.C. Mark 5 on Atlas

Atlas is a large time-sharing computer. Its operation is governed by a supervisor program which organizes the time-sharing of users' programs and the operation of peripheral devices so as to use the system to its best advantage. In particular, the input and output phases of any program, although effected by the computer itself, are effectively off-line operations. Data is automatically read from the input medium and stored in an "input well," which is mainly on magnetic tape. Unless special arrangements are made, it is impossible for a program to refer to any of this data unless all of it has been read and recorded by the supervisor. In a similar manner, output from a program is stored in an "output well," and is not actually printed until a suitable peripheral device becomes free. This may be a considerable time after the output is generated. Inside the computer itself, the supervisor chooses suitable combinations of programs, and runs them simultaneously. It gives no external indication of the identity of the programs actually being processed at any time.

It will be seen, therefore, that once a program has been supplied to the machine, it is virtually impossible for the user to observe its progress, or to influence the course of events in any way.

Survey analysis, on the other hand, must be carried out as a "joint operation" of machine and user. At an early stage, the user may wish to make corrections to either specification or data; and, in later stages, he may wish to use preliminary results as a guide in drawing up further table specifications intended to explore a section of the same data in greater detail.

If the entire body of data, and the specification, had to be supplied to the machine afresh after every correction or addition made by the user, the overall operation of the system would be intolerably inefficient. The actual operating system for M.V.C. programs is therefore a compromise between the "closed-shop" requirements of the computer, and the "open-shop" needs of the user.

A survey analysis on Atlas runs in three independent stages. Each stage is, from the machine's point of view, a complete program. The only communication between stages is provided by a private magnetic tape, allotted exclusively to the user.

Stage one is concerned with reading M.V.C. survey descriptions, and translating them into a form suitable for use by the subsequent stages. The compiler-type program used for this phase is carefully constructed so as to detect all syntactical errors, and to query unusual constructions. It also provides a check on the amount of output requested; this was judged necessary because experience with an earlier survey analysis program (M.V.C. Mark 3) showed that inexperienced users are liable to make such free use of the available short-hand devices that they unwittingly specify many thousands, or even millions, of independent frequency counts.

This stage may be used either to translate "brand-new" specifications, or to make additions, corrections, and deletions to previously supplied ones.

All the input text is stored, in untranslated form, on the user's private magnetic tape, against the possibility of future modifications. The translation is also stored if the specification is successfully compiled without detection of any errors.

Stage two is devoted to reading and editing raw data. It reads card images or strings of characters, evaluates all the defined variables, and carries out the specified checks. Successful cases are stored on the private tape, and details of unsuccessful cases are printed. Full particulars are only supplied for the first 100 cases to fail; thereafter, the program is stopped. This is a necessary precaution against errors in the specification which would cause every case to fail; this is the only likely reason for such a high failure rate, and the precise cause can usually be discovered from an examination of the first few cases.

All this is carried out by routines for which parameters are provided by stage one. No tabulations are made at this stage. Stage two can be used to read either new or additional and corrected data. For small and medium sized surveys, reading will be via the input well; but when the amount of data exceeds the capacity of the well, it may be necessary to use an on-line device.

The products of stage two are, therefore,

(1) a magnetic tape containing edited and transformed data; and

(2) a full description of up to 100 rejected cases.

Stage three is used to make the actual tabulations. It draws its information entirely from the private magnetic tape, using specifications recorded during stage one to analyze data read during stage two.

The output from the programs can be directed to any type of peripheral device, such as a line printer, a Flexowriter, or a teleprinter, at the user's convenience. It can also be stored on the private magnetic tape for eventual use by a different program.

These three phases are arranged to be subroutines of an M.V.C. operating system. The user communicates with this system by means of an "M.V.C. metalanguage," which allows the stages to be used in a flexible manner.

We shall illustrate the use of the system by a simple example. Let us suppose that a user wishes to make an exploratory survey of a body of data. He has written a survey specification, which he believes to be correct. His initial message to the operating system is:*

COMPILE NEW SPECIFICATION
(i.e. carry out stage 1)
COMPILE NEW DATA (i.e. carry out stage 2)
STOP IF ERRORS EXCEED 3
TABULATE

Suppose that the specification is in fact correct, but that the data contains 5 cases with errors. The system completes stage one, and starts on stage two. When it has examined all the data, and recorded details of the correct cases on the private tape, it stops, because the number of errors (5) exceeds 3.

Next, the user produces corrected versions of the rejected cases, and supplies them to the system, with the following message:

COMPILE ADDITIONAL DATA
TABULATE

The system enters stage two, and compiles the corrected cases. If they are indeed correct, it passes on to stage three, generating the required tables.

Suppose now that the user has examined his results (taking maybe several weeks or months to do so), and has drawn up more table specifications to be applied to the same data. He gives them to the machine, with his private tape and the following message:

COMPILE ADDITIONAL SPECIFICATION
TABULATE

In this way, the flexibility required for survey analysis can be achieved without gross inefficiency in the overall operation of the computer.

* Naturally, all messages and data supplied to the machine must be cast in the form of Atlas "Documents", with suitable headings and endings. These are not shown in this paper.

This operating system is a tentative one, and only experience can show whether it is really suitable for its purpose.

Conclusion

It need hardly be remarked that the notion of a general program for survey analysis is not even remotely original. In designing M.V.C. Mark 5, the author has attempted to blend the useful features of earlier survey analysis programs, to make use of practical experience gained by extensive use of one of these, and to produce a system which would be properly matched to the vastly increased power of a computer the size of Atlas.

One of his chief aims was to make the program easy to use by persons totally unacquainted with computers. With one exception known to the author, earlier systems do not fulfil this requirement. Where they are more than just a collection of subroutines, they employ special-purpose autocodes of such abstract symbolism that their use is virtually confined to professional programmers. The exception is AUTOSTAT (Douglas and Mitchell, 1960), which employs a simple and easily grasped notation. When AUTOSTAT is extended in certain directions, it forms a system comparable in flexibility and power to the one outlined in this paper.

In spite of this fact, the decision to design a completely new system was taken because it was felt that a system developed with a large computer directly in mind would be more generally acceptable than one which was originally intended for a small machine, and subsequently extended.

When M.V.C. Mark 5 comes into operation, it will be interesting to compare its practical merits with those of the OPAL system (an extended version of AUTOSTAT) available on the IBM 7090.

References

DOUGLAS, A. S., and MITCHELL, A. J. (1960). "AUTOSTAT: a Language for Statistical Programming," *The Computer Journal*, Vol. 3, p. 61.

YATES, F., and SIMPSON, H. R. (1960). "A General Program for the Analysis of Surveys," Pt. 1: *The Computer Journal*, Vol. 3, p. 136. Pt. 2: *The Computer Journal*, Vol. 4, p. 20.

COLIN, A. J. T. (1961). "M.V.C. Mark 3: A General Survey Analysis Program for the FERRANTI Mercury." Internal report, University of London Computer Unit.*

COLIN, A. J. T. (1963). "The M.V.C. Manual." Internal report, University of London Computer Unit.*

NAUR, P. (Ed.) (1960). "Report on the algorithmic language ALGOL 60," *Comm. ACM*, Vol. 3, p. 299, and *Numer. Math.*, Vol. 2, p. 106.

KILBURN, T., HOWARTH, D. J., PAYNE, R. B., and SUMNER, F. H. (1961). "The Manchester University Atlas Operating System," Pt. 1: *The Computer Journal*, Vol. 4, p. 222; Pt. 2: *The Computer Journal*, Vol. 4, p. 226.

KILBURN, T., PAYNE, R. B., and HOWARTH, D. J. (1961). "The Atlas supervisor," *Proc. E. J. C. C.*, December, 1961.

HOWARTH, D. J., JONES, P. D., and WYLD, M. T. (1962). "The Atlas scheduling system," *The Computer Journal*, Vol. 5, p. 238.

PILLING, DIANA (1963). "C-E-I-R OPAL Language," Internal report, C-E-I-R (U.K.) Ltd.

* Obtainable on request to the author.

Book Review

Computer Applications in the Behavioral Sciences, Edited by HAROLD BORKO, 1962; 633 pages. (Englewood Cliffs, N.J., and London: *Prentice-Hall*, 57s. 6d.)

This book is in three very unequal parts. Parts I and II, which occupy the first 140 pages, are by the editor himself, who is a psychologist on the research staff of the Systems Development Corporation; they comprise an introduction to computers for the behavioral scientists to whom the book is addressed. Dr. Borko's sketch of the potential applications of computers in the behavioral sciences and his discussion of the question "Do Computers Think?" account for 30 or so of the 140 pages, and will certainly interest and stimulate his intended readers. The remainder of Parts I and II comprise a conventional and pedestrian account of the history, design, construction, function and programming of the automatic calculating engine.

The rest of the book, another 500 pages, consists of 17 invited contributions describing specific applications, and covers a pretty wide range: thus Feldman on his simulation of human binary-choice behaviour; Hiller and Baker on computer analysis and synthesis of music; Ross Ashby on brain simulation; Ledley on computers in medicine; Culbertson on nerve net theory; Benson on simulation of international relations and diplomacy. There are three contributions on certain kinds of multi-variate statistical analysis favoured by certain kinds of behavioral scientist—factor analysis, for example. No service will be done to psychology if these chapters contribute still further to the computer-assisted application of methodological incompetence to larger and larger masses of data—which in the view of their superficial and uncritical approach is all too likely.

In view of the moderate price, this book can be recommended on the curate's egg principle, although there may well be disagreement on the distribution of goodness among its parts.

W. L. B. NIXON.