COMPUTERS IN BEHAVIORAL SCIENCE

Publication of this department is partially supported by a grant from the National Science Foundation to the chairman of its editorial committee, Steven G. Vandenbarg.

Performance of a Reading Task by an Elementary Perceiving and Memorizing Program, Edward A. Feigenbaum, University of California, Berkeley, California, and Herbert A. Simon, Carnegie Institute of Technology, Pittsburgh, Pennsylvania.

The Elementary Perceiver and Memorizer (EPAM) is a computer program designed to simulate the processes used by human subjects to perform rote memory tasks—particularly learning nonsense syllables by the method of paired associates (Feigenbaum, 1959, 1961). In the present brief account, we wish to report some experiments which show that the mechanisms postulated in EPAM for the rote memory tasks are adequate for simulating, at least macroscopically, the processes employed by human beings in learning to read and understand printed words. First, we shall provide a summary description of the EPAM program, mentioning the main processes it uses in rote memory tasks. Then we shall describe how these processes are used by EPAM in learning to read.

The EPAM program

Since the EPAM program has been described in detail elsewhere (Feigenbaum, 1959, 1961), we shall summarize it here quite briefly. It is a computer program of about 1,000 instructions, written in an interpretative language, IPL-V, and tested under a variety of conditions in some 100 runs on the IBM 704 and IBM 7090.

The EPAM processes perform the following four principal functions:

A. recognize an external stimulus as one about which some information has already been memorized;

B. add new stimulus items to the memory by building discriminations (tests) that allow the new item to be distinguished from stimuli previously learned;

C. associate (internally) two stored items, say x and y, by storing with x some cue information about y; and

D. respond to an external stimulus X with a response, Y, by retrieving the cue to the response, and then retrieving the response using the cue.

Thus EPAM has two performance processes, enabling it to respond with material already learned: the discrimination process (A), which recognizes the stimulus, and the response process (D), which finds the appropriate response associated with the stimulus and produces it. EPAM also has two learning processes: the discrimination learning process (B), which elaborates the structure of discrimination tests it applies to stimuli, and the association learning process (C), which associates response cues with stimuli.

The central memory structure, which the performance processes use and the learning processes construct, is the discrimination net (see Figure 1). It is a tree-like nexus of associations at whose terminal nodes are stored images of encodings of external stimuli. At the nonterminal nodes of the net are stored tests which examine particular bits of the encodings. The image of a stimulus is retrieved by sorting the encoding of the stimulus down through the tests of the net to the appropriate terminal. In learning a set of stimuli, the net is expanded to a size that is just large enough (roughly) to discriminate among the different stimuli that have been presented to the system.

Association of a response, Y, to a stimulus, x, is accomplished by storing a small amount of the information about Y (an incomplete cue image of y) along with the image of x. The system determines by trial and error how much information must be stored as a cue to retrieve the response from the net when the association is made.
Computers in Behavioral Science

Fig. 1. A Typical EPAM Discrimination Net.

\[ \text{T} = \text{Discriminating test at a node} \]
\[ \text{I} = \text{Image at a terminal} \]
\[ \text{I,C} = \text{Image and cue at a terminal} \]
\[ \text{} = \text{Empty terminal} \]

EPAM responds to a stimulus (see Figure 2) by sorting it in the discrimination net, finding the associated response cue, sorting that cue in the same net, finding its image, and using the response image to produce the response.

EPAM does not simulate the initial sensory and perceptual processes (the processes in Figure 2 labeled “Perceive Features of Stimulus”) that scan an external stimulus and extract from it encoded information that is used by the memorizing and responding processes. The EPAM program takes up at the point where the initial perceptual encoding has already been accomplished. Its “stimuli” are encodings of the external stimuli. Thus, the EPAM stimuli corresponding to letters of the alphabet hold information about whether a letter contains a curved segment, a straight segment, a closed loop, and so on. Each letter is discriminated from the others, as the discrimination net grows, by its possession of a unique combination of these simple topological and metrical properties.

Different encodings are used to represent stimuli in the different sensory modes, and in certain submodes. Thus, there may be a net for discriminating among aural phonemes by encodings that represent elementary phonemic characteristics; another net for discriminating among visually presented syllables of letters; and still another net for discriminating among objects in terms of simple characteristics of shape. We will be concerned, on the sensory side, with these three specific modes: aural phonemic, visual literal, and visual object modes.

On the response side, too, several modes are represented by different encodings. In learning to read, EPAM uses three response modes, paired, respectively, with the three stimulus modes. There is an oral response

Fig. 2. EPAM Performance Process for Producing the Response Associated with a Stimulus.
mode, whose images may be interpreted as the signals that activate the muscles used to produce spoken language. There is a written response mode, representing signals that activate muscles used in printing letters.

There is a pointing response mode, which, as its name implies, signals the selection of an object from a set of objects by pointing. In the present EPAM program, the response modes are represented only in a rather rudimentary form—the system has been elaborated principally on the stimulus side.

Paired associate learning

A typical rote memory experiment, using the method of paired associates, is performed as follows:

A set of nonsense syllables (each, say, a sequence of three letters of the alphabet) is chosen, and the syllables are paired. The syllables are shown to the subject, one pair at a time. First the initial member of a pair (stimulus item) is shown. The subject tries to spell (or pronounce, depending on the instructions) the second member of the same pair (response item). After a short interval, the stimulus and response items of the pair are shown to him simultaneously. After a few seconds, the cycle is repeated with a new pair of syllables; this continues until all pairs have been presented (a trial). Trials are repeated, usually until the subject is able to give the correct response to each stimulus. In successive trials the syllable pairs are reordered randomly, so that they are not always presented in the same sequence.

In the form of the experiment we have described, both the stimuli and the responses were presented to the subject in the same mode—in this case, the visual literal mode. Moreover, the responses were to be made in the oral mode, either by pronouncing the syllables or by pronouncing the names of their letters. There is no reason why the stimulus and the response cannot be presented to the subject in quite different modes, and it is clear from the example that the subject's response can represent a sense modality (in this case oral) entirely distinct from the modality (in this case visual) of either the stimulus or the response presented to him. Thus, the stimulus items could be presented to the subject as aural phonemic syllables, the response items as visual objects (or visual pictures of objects). The subject could respond to the aural phonemic syllable by pointing to the corresponding object (or picture) selected from a set of objects. Let us call a paired associate task using this combination of modes a task of Type I.

Another possible combination of modes would be to present the stimuli to the subject as visual literal (printed) symbols, the responses as aural phonemic syllables, and instruct the subject to respond to the stimulus with the appropriate oral phonemic syllable. Let us call the paired associate task using this combination of modes a task of Type II.

Learning to read

We shall now show that a system capable of performing paired-associate rote memory tasks in an appropriate range of modes is capable of reading names of objects—where "read" is understood in its usual sense of "scan visually and behave appropriately."

First, let us consider the behavior usually observed in children as they are learning spoken and written language. At an early age, a child acquires the ability to mimic simple aural phonemes. That is, he builds up associations from each aural phoneme to the oral phoneme that will produce sound corresponding, when fed back to the ear, to the aural phoneme. Also at an early stage, a child acquires the ability to "understand" simple aural phonemes that correspond to names of objects. That is, if an adult repeatedly pronounces the name of an object and then points to the object, the child gradually acquires the ability to point to the object when he hears its name. At a much later stage, the child acquires the ability to read simple printed words: that is, when he perceives a visual literal stimulus, he responds by pronouncing the appropriate word. At about the same time, he learns the "meaning" of the word: that is, when he perceives the visual literal stimulus, he can
respond by pointing to the corresponding object or picture of an object.

Let us consider the case of a child who can already produce orally words that he is familiar with aurally—who can mimic familiar speech sounds. For this child, learning the meanings of aurally presented words (i.e., so that he can respond to them by pointing) is a paired associate task of Type I. For the child then to learn to read visually presented words is a task of Type II. After the child has accomplished the Type I and Type II tasks for some set of words, he will be able to perform the third task of demonstrating that he can read “meaningfully” by pointing to the objects whose printed names he encounters.

This sequence of learning experiences was simulated successfully for the first time with EPAM in November 1960. The four words “dog,” “cat,” “car,” and “ball” were used in the experiment. Encodings were constructed corresponding to (1) characteristics of the visually-presented objects, (2) characteristics of their aurally presented names (DAWG, KAT, KAHR, BAWL), and (3) characteristics of their visually presented names (DOG, CAT, CAR, BALL). EPAM was assumed already to possess the means for producing responses associated with stimuli in a given sensory modality. That is, once EPAM had become familiar with an aural word, it could emit the corresponding oral response; once familiar with a printed word, it could emit the corresponding printed response; once familiar with an object, it could point at it. Acquiring these correlations is another learning task that was not considered in this particular experiment.

EPAM was presented with a sequence of three tasks, corresponding exactly to the sequence of learning experiences we described for the child. The first task, of Type I, presented aural phoneme syllables paired with objects, and called for a pointing response. The second task, of Type II, presented visual printed words paired with aural words, and called for an oral response. The third task presented printed words by themselves, and called for a pointing response. In the first task, EPAM learned the objects corresponding to spoken words; in the second task, it learned the spoken words corresponding to printed words. In the third task, it demonstrated reading ability, responding to printed words by pointing to the objects they named. In the actual experiment, EPAM required four trials to perform the first task correctly for all four words, four trials for the second task, and performed the third task correctly on the first trial. Because the data are not voluminous, we give, below, the actual responses EPAM made on each presentation of a stimulus on each task.

1. Presented Stimulus—aural phoneme (e.g., DAWG)  
Presented Response—visual object (e.g., dog)  
Subject’s Response—pointed at object (e.g., dog)

<table>
<thead>
<tr>
<th>Trial</th>
<th>Presented Stimulus</th>
<th>Subject’s Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>KAHR</td>
<td>car</td>
</tr>
<tr>
<td></td>
<td>DAWG</td>
<td>car</td>
</tr>
<tr>
<td></td>
<td>KAT</td>
<td>car</td>
</tr>
<tr>
<td></td>
<td>BAWL</td>
<td>car</td>
</tr>
<tr>
<td>2</td>
<td>KAHR</td>
<td>ball</td>
</tr>
<tr>
<td></td>
<td>DAWG</td>
<td>ball</td>
</tr>
<tr>
<td></td>
<td>KAT</td>
<td>cat</td>
</tr>
<tr>
<td></td>
<td>BAWL</td>
<td>ball</td>
</tr>
<tr>
<td>3</td>
<td>BAWL</td>
<td>ball</td>
</tr>
<tr>
<td></td>
<td>KAHR</td>
<td>car</td>
</tr>
<tr>
<td></td>
<td>KAT</td>
<td>cat</td>
</tr>
<tr>
<td></td>
<td>DAWG</td>
<td>cat</td>
</tr>
<tr>
<td>4</td>
<td>BAWL</td>
<td>ball</td>
</tr>
<tr>
<td></td>
<td>KAHR</td>
<td>car</td>
</tr>
<tr>
<td></td>
<td>DAWG</td>
<td>dog</td>
</tr>
<tr>
<td></td>
<td>KAT</td>
<td>cat</td>
</tr>
</tbody>
</table>

2. Presented Stimulus—visual printed word (e.g., DOG)  
Presented Response—aural phoneme (e.g., DAWG)  
Subject’s Response—oral phoneme (e.g., DAWG)

<table>
<thead>
<tr>
<th>Trial</th>
<th>Presented Stimulus</th>
<th>Subject’s Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CAR</td>
<td>KAHNR</td>
</tr>
<tr>
<td></td>
<td>DOG</td>
<td>KAHNR</td>
</tr>
<tr>
<td></td>
<td>CAT</td>
<td>KAHNR</td>
</tr>
<tr>
<td></td>
<td>BALL</td>
<td>KAHNR</td>
</tr>
<tr>
<td>2</td>
<td>DOG</td>
<td>BAWL</td>
</tr>
<tr>
<td></td>
<td>BALL</td>
<td>BAWL</td>
</tr>
<tr>
<td></td>
<td>CAT</td>
<td>BAWL</td>
</tr>
<tr>
<td></td>
<td>CAR</td>
<td>KAHNR</td>
</tr>
</tbody>
</table>
3. (Reading Test)
Presented Stimulus—visual printed word (e.g., DOG)
Subject's Response—object pointed at (e.g., dog)

<table>
<thead>
<tr>
<th>Trial</th>
<th>Presented Stimulus</th>
<th>Subject's Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CAR</td>
<td>car</td>
</tr>
<tr>
<td></td>
<td>DOG</td>
<td>dog</td>
</tr>
<tr>
<td></td>
<td>CAT</td>
<td>cat</td>
</tr>
<tr>
<td></td>
<td>BALL</td>
<td>ball</td>
</tr>
</tbody>
</table>

In performing the reading task, EPAM sorted the visual literal stimulus, finding the cue for the associated aural word. The aural cue was then sorted to find the associated visual object cue. The object cue was sorted to find the visual response image needed to activate the pointing program. Thus the association between printed word and its “meaning” was mediated via the aural mode. This mediation might explain the fact that beginning readers commonly move their lips, but it does not necessarily imply muscular oral responses, for the auditory symbols used in the mediation belong to the sensory, rather than the motor, part of the system. EPAM’s associational processes could construct new direct associations between visual literal stimuli and visually presented objects, but these additional learning processes were not postulated in the version of EPAM that was tested in the reading experiment. In our culture, oral language is almost invariably acquired first, and there is some evidence that even in most adults the aural mode performs a mediating function for word associations.

REFERENCES
(Manuscript received September 21, 1961)

Examination by Computer, Robert E. Smith, Programming Research, Control Data Corporation, Minneapolis, Minnesota.

Science is concerned with that which can be measured. As science has progressed, so has the art of measurement. The concepts to be measured and the measuring technique are closely interwoven. The electronic computer has provided scientists with an improved technique which has not only opened new doors, but has suggested another look inside doors long open. This paper suggests that the computer can make significant contributions in the general area of testing, examining, and measuring the personal characteristics of people.

Psychologists are well aware of the difficulties involved in selecting competent people for specific functions. In trying to solve this problem, many techniques have been designed and applied, including the personal interview, tests and examinations of various kinds, and the questionnaire. All of these are faced with the problem of keeping pace with increasing demands for improved selections and interpretations. The problem is further aggravated by the fact that efficient selections are more difficult to attain because of the increased number and complexity of factors involved.

What are some of the necessary elements of the selection process? First, there are the parameters, possibilities, alternatives, or hypotheses which are involved. Second, there are the methods, techniques, devices, and logic which are applied in the process. Third, there are the results or the selections themselves.

The degree of difficulty that arises in making a selection is related to the number of alternatives involved. A choice between two things is generally easier to make than between two hundred. Yet the number of alternatives has increased far beyond this figure in many of the selective processes employed in situations where large numbers of people or human factors are found. Employment interviewing, personnel studies, medical diagnoses, physical and mental fitness tests for
various occupations requiring public licenses, and psychological analyses for many other purposes are examples of areas where efficient selection has become more difficult.

Further complexity is added as the techniques through which selection is made require more and more computation and skilled interpretation. This is true in most of the areas mentioned above. In some cases, the complexity involved in the computational aspects of the problem has been such that simpler but less efficient techniques have been adopted in order to make selections.

Fortunately, man has at his disposal the electronic digital computer, which is capable of performing complicated computations at tremendous speeds. Using this machine, it is possible to examine a mass of alternatives, analyze them through complex computational techniques, and interpret the results so that selections can be made accurately and efficiently. In this paper, I want to suggest an entirely different application of the computer to the psychological selection process: namely, the use of the computer to interrogate the subject directly.

There are three problems associated with such use of the computer: cost, time and difficulty of programming, and reluctance to use the computer in certain selective areas. Taylor (1959) has discussed the first two problems; this paper is concerned with the third. Although the first two may appear to be formidable to many, there is increasing evidence that both cost and difficulties of programming are lessening. Computer costs have declined significantly and will continue to do so as more computers become available in each city. Language translation programs, known as compilers, have made programming simpler.

The third problem is more insidious. One has to understand the historical background of computer development to comprehend the reluctance to use the computer for certain applications. From the beginning, larger storage facilities and higher speed capabilities have been the twin goals of computer designers. In an effort to make the most "efficient" use of storage and speed, systems engineers have been wary of computer applications which involve mutual interaction between the machine and the operator.

In striving for more automation, systems people have generally overlooked the elegant and useful opportunities inherent in the opposite philosophy—those applications which involve co-operative interaction of computer and man. Indeed, some applications depend upon a "stimulus-response" situation in order to function properly (Vandenberg, 1960; Weinrauch & Hetherington, 1959).

This report suggests that the use of the electronic digital computer can significantly increase the efficiency of the selective process. An analysis of test characteristics, in terms of possible computer applications, indicates that such is the case.

Can the computer provide for the types of questions usually included in the ordinary test or questionnaire? It appears that it can and that certain advantages accrue in some instances. "Confession-type" questions, for example, often require a straightforward, honest response to inquiries related to the subject's personal experience or characteristics. Some subjects resent having to write down responses to such questions. Resentment may be present if the response must be made orally to another person. How will the subject feel about "confessing" to a machine? As pointed out by Vandenberg (1960, p. 172), "There is a certain fascination in speculating what the psychological effect of [direct communication between subject and computer] might be." Certainly there is reason to believe that for some it would become a more impersonal experience, and as such, responses would be made more freely. Still another advantage accrues. In writing responses to "confession-type" questions, one is often influenced to change the original response by an idea suggested in a later question. Thus the person who first admits that he "reads no books" may change this when he comes upon a later question in regard to "his reading habits." Using the computer, the earlier question would not be as readily available to the subject and thus responses not as easily changed.

For the "Do you know" type of questions,
the use of a computer holds interesting possibilities. There is no reason why the questions presently appearing on tests can not be used. Directions for choosing the correct response would require some revision in order that representative codes might be entered into the computer. For example, a certain question now requiring a pencil mark opposite "Box C" for the correct answer might require that the code of "C" be entered as input in the computer. There are many ways in which this could be done; for example, the subject could be directed to depress the input button C, or to type a C on a typewriter. Even more important and attractive is the possibility of being able to vary the sequence and the contents of test questions for different subjects. Under present testing conditions, all subjects are usually required to answer the same questions presented in the same sequential order. However, as in some teaching machines, a computer program can be written so that the subject's answer determines the next question to ask. Thus the whole philosophy of "pertinent questioning" (so interestingly exemplified in the popular book on personality analysis by Loewenstein & Gerhardi, 1942) is made possible.

For other types of questions—dexterity, timed tests, agree-disagree items, etc.—the use of the computer holds intriguing capabilities. It is true that problems must first be solved by anyone contemplating such use of computers but evidence at hand indicates that the gains are worth the effort required. As one writer recently claims (Coulson, 1961, p. 252), "Computers may present complex stimulus patterns to an individual or group, and may also analyze the humans' response behavior." This area of using the computer to stimulate and interact with human subjects is one that has been too often neglected. White (1961) has given some examples of this in the field of research in perception.

Can the computer improve test administration? Regardless of the care taken to standardize administration procedures inequities often result. This is largely due to the fact that organizational procedures are controlled by humans who try to follow fixed regulations concerned with administering the test. Different interpretations of these regulations often lead to inequities in administration of the tests. If the administrative process is controlled by a fixed computer program, it would seem that more equal procedures for each test would result. With the computer, it is also possible to read the test questions to the subject who can then respond by entering a code into the computer. The reader can be in another room. Thus, the subject hears the question, rather than reading it, and responds in privacy through the computer. The fact that some tests must be read by the subject is in itself a disadvantage, since results are often influenced by one's reading ability rather than by the ability being tested. The use of a dictaphone or other sound reproducer can eliminate the need for another person to read the questions. In this connection, it is worth noting that the time may not be long until the computer can phonetically read the questions from its own memory (Dersch, 1960; Davis, 1960).

How well can the computer score the tests and interpret the results? Here indeed, the computer can make a tremendous contribution. Designed to operate at fantastic speeds, the computer can score and interpret results as fast as the subject can respond to the questions. This means that upon completion of the test, the scored and interpreted results can be available immediately. It is of interest to note, in this respect, that most, or all, of the computer time allotted to administering the test can be recovered in the time saved in scoring and interpreting the results.

Of importance in regard to computer interpretation is the fact that many selective processes require analyses of multinomial distributions. In particular, these seem to originate from questionnaire or survey type data. As one recent writer (Farmer, 1961) indicates, "The use of electronic computers for obtaining distributions provides three very desirable advantages over other methods: (1) accuracy, (2) greatly shortened analysis time, and (3) the results are in suitable form (punched cards or tape) for further analysis by automatic methods."
If computer usage in a particular testing system appears to be conducive to good management then it is important that new principles be established so that the selective process can be built around the computer. This will require extensive study and research by those trained to design testing instruments. It will be necessary for these experts to become acquainted with computers or to work closely with those who have been so trained. The over-all goals of the psychological tests now being used need not change significantly, yet the means to these goals may require extensive modifications.

There is evidence on all hands to indicate an increasing need for better measuring instruments in key selective areas. Some of these areas, like personnel analyses, salary administration, and employee selection, lie at the very heart of management. Others, like medical diagnosis, decision making, and the selection of competent automobile drivers, are of importance to every citizen. Perhaps hope lies in computer programmed interactions between subject and testing instrument.

REFERENCES

Coulson, J. E. Recent areas of application. Behav. Sci., 1961, 6, 252.


White, B. J. The computer as a pattern generator for perceptual research. Behav. Sci., 1961, 6, 252-259.

(Manuscript received December 15, 1961)

BOOK REVIEW


This is a manual and primer from which one can teach himself "IPL-V"—the language used for a number of the recent simulation and artificial intelligence programs for digital computers. Any intelligent person who is not upset by numbers or precise detailed specifications should be able, with a fair amount of application, to sit down with this manual, start writing very elementary program examples (following the first 44 pages of text) in an hour or two, start writing (poorly) a simulation problem of fair complexity in a week or two, and, given several months of experience (during which he may have completed a program which he almost certainly won't like) be fairly proficient and comfortable with a tool that can serve as a relatively effective way to formulate and explore his ideas, as well as, via the finished program, examine their consequences. In other words, programming is a relatively simple and homely skill, similar to carpentry or bookkeeping. A crooked nail can be driven almost immediately; almost-unconscious practice and application lead to slow and steady progress; a finished chair can be a work of art.

This manual seems adequate to its task. It is really two manuals—a first section, for the layman, that introduces programming ideas and problems in easy steps, and a second section that summarizes the system. The reader with some familiarity with programming may prefer to start with the second section, and any reader will probably find it helpful to skip to the second section, and to skip over difficult parts (like "generators") in the first section. This is material to be absorbed through practice, and only with practice will the power of the tool being learned become apparent.

Pages 84 through 133 pose, analyze, and give the program for a simple adaptive system that builds up a memory network of associations as a function of training. This is
quite a suggestive problem, one that seems well worth trying to work through and then to extend beyond the present simple form. It is rather different from the complex strategy-manipulating models of problem-solving and theorem-proving behavior that Newell and the Carnegie group are commonly associated with, and gives a glimpse of the great potential flexibility of the computer as a vehicle for building models.

IPL-V is probably not as easy a programming language to learn as FORTRAN and some of the other high-level compilers. But for certain types of problems it has a good bit of mechanism already built into it, to make programming easier. (There is really no issue as to whether a program can be written in a less exotic language than IPL-V. If a program can be written, then it can [in theory] be written in any language. Nor does it seem fair to say that IPL-V is so much easier than any other language that it is obviously the language of choice. Interesting simulation programs that have to handle the same messy types of structures have been written in all types of languages.) The chief virtue of IPL-V is that it gives the programmer great flexibility in handling complicated data structures—lists, trees, interrelated collections of objects. Another virtue of IPL-V is its availability in at least a dozen universities and laboratories. Thus it is possible to get help, and to trade programs.

For some people the task of writing a program is great fun, much like a good game or puzzle. For others it is a minor frustration and bore. But having to write one’s own program is probably a smaller problem than having to communicate to a programmer who will write it for you. And it leads to a much better understanding of the potentialities and the problem itself—what can and cannot reasonably be done. The behavioral scientist who would like to develop more complex models should realize that computer programming, rather than a barrier, is a minor technical hurdle similar to the clinician’s statistics or the morphologist’s microscopy.

Leonard Uhr
Mental Health Research Institute

COMPUTER PROGRAM ABSTRACTS

CDC 1604 Program for Product-Moment Intercorrelation, Tetrachoric Approximations, and Elementary Linkage Analysis, Donald J. Veldman, University of Texas. (CPA 101)

Description: The program computes means, standard deviation parameters, standard deviation statistics, and Pearson product-moment intercorrelations for a maximum of 100 variables. Intercorrelations are printed as a square matrix. If dichotomous (0,1) data are used, the resulting phi-coefficients are used to obtain tetrachoric approximations (optional) by means of the formula \( r_t = \sin(\phi) \). At option, the program will also construct a list containing column numbers and the row number for each column which determines the highest (absolute) correlation. An elementary linkage analysis (McQuitty, 1957) may be completed in a few minutes from this list. Another option allows inversion of the data matrix at read-in. The program is coded in FORTRAN and is available from the Department of Educational Psychology, The University of Texas (refer to CORMAT).

Reference:

CDC 1604 Program for Intercorrelation, Principal Axis Factor Analysis, Varimax Rotation, and Factor Score Computation, Donald J. Veldman and Earl Jennings, The University of Texas. (CPA 102)

Description: The program computes means, standard deviations, and intercorrelations for a maximum of 100 variables. Eigenvalues and vectors are extracted by Jacobi’s method and the most significant factors are retained for rotation by the normalized varimax procedure. In order to obtain weights for factor score computation, the original intercorrelation matrix is augmented by the rotated factor matrix and each column is predicted in turn by all of the original variables utilizing an iterative regression routine. The weights are adjusted for heterogeneity among the original-variable variances and factor scores are computed for all subjects in the original data matrix. These scores are standard-
Computers in Behavioral Science

IBM 704 Program for Forecasting by Exponential Smoothing, J. N. Whelan, E. I. du Pont de Nemours and Co., Inc., Wilmington 98, Delaware. (CPA 104)

Description: The program tests a forecasting model based on exponential smoothing on a time series such as monthly sales data. The model considers both seasonal and trend effects. In order to use the program, actual monthly sales data for several years are required. The first few years of these data are used to calculate starting values for the deseasonalized average monthly sales, the monthly seasonal factors, and the trend of sales. The program then proceeds one month at a time through the remainder of the actual sales data, each month forecasting the sales for the desired number of months in the future. At the end of each month the average sales, seasonal factors, and trends are updated based on actual sales for that month and a new forecast is made. Each forecast is compared with actual sales and this comparison is used to calculate a standard deviation of the forecast errors, the coefficient of variation of the forecast errors, a weighted sum of the squared errors, and an average fractional error. This provides a method for determining the combination of exponential smoothing constants that result in the best forecasts (Winters, 1960).

Computer: IBM 704 8K Core, 1 Tape. Program language: Fortran II.

Running time: Varies depending upon the number of items of data, number of periods projected, the period of time over which forecasts are made and the printout options selected. In a typical case involving 6\frac{1}{2} months of monthly data, of which the first half was used to calculate initial values, and the second half to forecast one and two months ahead, 27 combinations of smoothing constants were run in 1.2 minutes including card loading.

Comments: Printout options allow the selection of a minimum amount of printout when various combinations of smoothing constants are being investigated and a more complete printout when the best combination has been determined. The program has been successfully run many times.

Availability: A manual for this program will be published by the American Institute of Chemical Engineers, 25 West 45th Street, New York 36, N.Y., if sufficient interest develops. Reprinted from Chem. Eng. Prog., 1962, 58, 2, 96, where it was program 088.

The program is coded in FORTRAN and is available from the Department of Educational Psychology, The University of Texas (refer to ABSTRAC).

CDC 1604 Program for Intercorrelation and Multiple Regression Analysis, Earl Jennings, The University of Texas, and Joe H. Ward, Jr., Air Force Systems Command. (CPA 103)

Description: For n observations on k variables (k \leq 160), the program produces a vector of means of dimension k, a vector of standard deviations of dimension k, and a matrix of intercorrelations of dimension k by k. Any number of regression equations may be computed based on the intercorrelation matrix. For a particular equation any one of the k variables may be selected as the criterion (dependent) variable, and any subset of the remaining k–1 variables may be specified as predictor (independent) variables. In order to circumvent the difficulties encountered when the predictor matrix is singular and to speed processing time, an iterative technique which guarantees a solution for the least squares weights is utilized. For each equation the program produces an iteration sequence, the squared multiple correlation at each iteration, a vector of standard partial regression weights of dimension k, a vector of partial regression weights of dimension k, and the regression constant.

The program is coded in FORTRAN and is available from the Department of Educational Psychology, The University of Texas (refer to REGRESS).
Reference:


An IBM 1620 Program for a Regression Transformation of Psychophysiologic Data, Britain J. Williams and Herbert Zimmer, University of Georgia. (CPA 105)

Description: Lacey's autonomic lability score probably is the most widely used transformation of psychophysiologic data at the present time. Presented here is a program to compute Lacey's autonomic lability score from a series of paired baseline (or resting) and reaction (or stress) scores, collected either on a number of different experimental subjects, or on a single experimental subject over time. The correlation between baseline and reaction levels, the means and standard deviations for baseline and reaction levels, and the standard scores can be obtained as intermediate results.

This program is coded in FORTRAN for an IBM 1620 data processing system with card input-output. With minor changes, it can be adapted for use with any compatible system.

Method: Let \( X_i \) and \( Y_i \) denote the \( i \)th baseline and reaction scores of a series. The program reads a set of \( N \) such pairs of scores and computes the baseline mean and standard deviation, \( \bar{X} \) and \( S_x \); the reaction mean and standard deviation, \( \bar{Y} \) and \( S_y \); and the correlation, \( r_{xy} \), between baseline and reaction levels. The data are then transformed to standard normal deviates, \( X_{st} \) and \( Y_{st} \). In this program McCall's normalized \( T \)-scores, \( x_{st} \) and \( y_{st} \), are calculated directly from the data, rather than looked up in a table which has been read into memory, since this appears to be a much more efficient operation. These results may be punched out, if desired, under the control of a console sense switch.

The autonomic lability scores, \( A_i \), are then computed (and punched) as follows:

\[
X_{st} = \frac{1}{S_x} [X_i - \bar{X}],
\]

\[
Y_{st} = \frac{1}{S_y} [Y_i - \bar{Y}],
\]

\[
r_{xy} = \frac{\sum_{i=1}^{N} X_i Y_i}{N S_x S_y} - \frac{\left( \sum_{i=1}^{N} X_i \right) \left( \sum_{i=1}^{N} Y_i \right)}{N S_x S_y},
\]

\[
A_i = 50 + 10 \left( \frac{Y_{st} - X_{st} r_{xy}}{\sqrt{1 - r_{xy}^2}} \right).
\]

Since one set of paired scores is processed before the next is read, there is no limit to the number of sets which can be processed. The number of pairs in a serial set of scores is limited by the size of the equipment available; however, a basic 1620 system with 20K (20,000 digits) of storage can process a set consisting of 800 pairs. Each additional 20K of storage increases this number by 1,000.

Computer: IBM 1620, with 60K core storage, automatic division, indirect addressing, and IBM 1622 read-punch unit. Program language: FORTRAN. Running time: For a set of 100 paired scores, the running time is estimated at two minutes.

Reference:

A Bendix G-15D Program for Symmetric and Nonsymmetric Correlation Matrices of Data with Missing Values, Rhea S. Das, Case Institute of Technology, Cleveland, Ohio, and James R. Wall and S. J. Singer, Aurora Gasoline Co., Detroit, Michigan. (CPA 106)

Description: Given a data matrix with some values missing, this program computes the product-moment correlation between each pair of variables using only the data for those subjects for whom both values on the pair of variables are present. For each correlation coefficient, the program also computes the associated pair of means and standard deviations for the subjects included in the computation of the correlation, and gives the number of subjects included in the computation. The formulas used are:

\[ r = \frac{\sum x y}{\sqrt{\sum x^2 \sum y^2}} \], \[ \bar{x} = \frac{\sum x}{N} \], and

\[ \sigma = \sqrt{\frac{\sum (x - \bar{x})^2}{N}} \]

for the correlation coefficient, mean, and standard deviation, respectively. At the option of the user, the program will compute the entire intercorrelation matrix, referred to as a symmetric matrix, or only a specified portion of the correlation matrix, referred to as a nonsymmetric matrix.

The data are to be punched as integers on an offline Flexowriter paper tape, after which they are converted to binary form and written on magnetic tape by the first part of the program. The correlations, means, and standard deviations are computed and typed out by the second part of the program. The number of variables must not exceed 106; the number of subjects is limited by the capacity of a G-15D word, such that the raw sums of squares or crossproducts must not exceed 268, 435, 456; any one value must range between one and three integers, and the maximum value which may be entered for any subject on any variable is 999.

Computer: Bendix G-15D, Alphanumeric Typewriter, Magnetic Tape Unit (MTA-2), Flexowriter. Program language: G-15D machine language. Running time: For a data matrix with \( N \) subjects and \( K \) variables, the time required to write the data on magnetic tape in binary form can be estimated in seconds by the equation:

\[ t_{sec} = 19.74 + 3.27 N + 8.60 K \]

To compute the symmetric intercorrelation matrix, the time required can be estimated in seconds \( \times 10^2 \) by the equation:

\[ t_{sec} \times 10^2 = -39.02 + .31 N + 4.84 K \]

Comments: The program uses fixed-point single precision arithmetic. Correlation coefficients are typed out to three decimal places, and means and standard deviations to two decimal places.

The program writeup, including operating instructions and the program tapes, can be obtained from the Bendix Users' Exchange, Bendix Computer Division, 5630 Arbor Vitae Street, Los Angeles 45, California, by requesting Users' Project 648-2.

IBM 7070 Item Analysis Program, A. W. Bendig, University of Pittsburgh. (CPA 107)

Description: The program reads in single-digit item responses and (optionally) criterion measures for each subject. The item responses may be left unkeyed or weighted by a key also read into storage. Printed output gives the mean and standard deviation of the total score and of each item, the correlation of the item with the total score, and the reliability (Kuder-Richardson Formula 20) of the total score. If criterion measures are read in, the means and standard deviations of the criteria and intercorrelations among the total score and criteria are also printed out along with the correlations of each item with the criteria. Optional output includes printing and/or punching the total score for each subject.

Computer: IBM 7070 with 10K storage, floating-decimal, and tape units or online card reader, printer, and punch. Program language: Modified Autocoder. Limitation: Single-digit (0-9) item responses, 2 to 600 items, 0 to 3 criterion measures, 3 to 10K subjects. If tape input-output option is used, cards will be read from tape unit 5 on channel 1 and printed and punched output will be written on unit 2 through channel 2.

An extended description of the program and symbolic and object decks are available from IBM Corporation, Data Processing Library Service, 590 Madison Avenue, New York 22, New York.

A Univac 1103 Program for Monte Carlo Trials on a Linear Learning Model in a 2-Person Situation, David P. Campbell and Robert L. Hall, University of Minnesota. (CPA 108)

Description: Burke (1959) has extended a
linear model of learning to analyze certain 2-person experiments. Theoretical means and variances of asymptotic response probabilities may be obtained analytically when the diagonal-sums restriction, described by Burke, is satisfied. When that restriction is not satisfied, the same theoretical values may be estimated by Monte Carlo methods. This program provides a flexible means of obtaining Monte Carlo estimates for various input values. Inputs that may be varied include the two matrices of conditional probabilities of reinforcing events, values of the learning-rate parameters, number of subjects and number of trials, and the initial response probabilities. The outputs include mean and variance of response probabilities after the specified number of Monte Carlo trials, and mean and variance of the probability of joint occurrence of a pair of responses by the two subjects.

**Computer:** Univac 1103 with RECO II, Flexie Load, and Supersampler routines. **Running time:** About ten minutes for one set of inputs, and six to eight minutes for each additional set.

**Availability:** A write-up of the program and sample of inputs and outputs may be obtained by writing to Dr. David Campbell, Student Counseling Bureau, 101 Eddy Hall, University of Minnesota, Minneapolis 14, Minnesota.

**Reference:**

**IBM 7070 Single Digit Correlation Program, A. W. Bendig, University of Pittsburgh.** (CPA 109)

**Description:** Assuming that alphanumeric data cards have been placed on tape by a previous card-to-tape operation, this program reads in from tape single-digit scores packed 75 per card. The variables (maximum number is 130) are intercorrelated and vectors of means and standard deviations and the triangular matrix of product-moment correlation coefficients (all rounded to three decimal places) are written on tape for later listing. The vectors and matrix, in floating-decimal form, remain in core storage at the end of the program and may be further analyzed by available multiple regression (Lotto, 1962) or factor analysis (Bendig, 1962a) programs. Data card input is identical to that required for rater reliability and item analysis programs (Bendig, 1962b, 1963).

**Computer:** IBM 7070 with 10K core storage, a floating-decimal, and tape units. **Program language:** Modified Autocoder 74.

**Limitations:** Input data must be single-digit scores (stanines, deciles, stens, item responses). Can be adapted to a 5K core storage machine which would limit the matrix to 85 variables. No limit on the number of data cases. Can be adapted to machines with online card reader and printer.

An extended description of the program and symbolic and object decks are available from IBM Corporation, Data Processing Library Service, White Plains, N. Y.

**References:**
Bendig, A. W. IBM 7070 program for principal axis factor analysis. *Behav. Sci.*, 1962, 7, 126-127. (a)
Bendig, A. W. IBM 7070 rater reliability program. *Behav. Sci.*, 1962, 7, 500. (b)
Bendig, A. W. IBM 7070 item analysis program. *Behav. Sci.*, 1963, 8, 83.

**IBM 7070 Item Analysis Program for the Strong Vocational Interest Blank, A. W. Bendig, University of Pittsburgh.** (CPA 110)

**Description:** This program correlates the three possible response options of each of the 400 SVIB items with one to three external criterion measures. Punched card input: 7 cards per subject including a card containing the criterion measure(s) and six cards with the 400-item responses (70 items per card). Printed output gives (for each item) the percentage of Ss selecting each response, zero order correlations (point-biserial) between the responses and the criteria, and first and second order partial correlations between the responses and each criterion partialing out the other criteria. A total of 14,400 correlations (three-decimal-place accuracy) are printed out if three criteria are available.

**Computer:** IBM 7070 with 10K core storage, floating-point, on-line card reader and printer. **Program language:** modified Four Tape Autocoder.

Object deck and description of program available from the University of Pittsburgh Computation and Data Processing Center.
Variance of Mean of a Sample Subject to Natural Clustering: Intra-Class Correlation.

FORTRAN Program for IBM 709-7090,
Carl E. Hopkins, School of Public Health, University of California, Los Angeles. (CPA 111)

Description: For a sample of observations, subsets of which are intra-class correlated, the program (a) sorts the elements into clusters and forms a frequency distribution of the number of clusters of each size, (b) computes the analysis of variance for each cluster size, arriving at the Fisher coefficient of intra-class correlation \( R = (MSB - MSW)/(MSB + (M - 1)MSW) \), (c) computes the grand mean across all clusters, (d) computes the ordinary variance of the grand mean, treating the whole sample as a simple random sample (ignoring clustering of elements), and (e) computes the variance of the grand mean corrected for the intra-class correlations observed in the sample. \( \text{Var.} (y) = (EM)^2EV^2 - 2EMEYEMY + (EV)^2(EM^2)/EM^3 (EM - 1) \). All of the above are printed out. The Finite Population Correction is not included.

Input is by card. Program will handle any number of variables, one at a time, on up to 20 elements in each of 20 cluster sizes.

Reference:

IBM 7070 Item Analysis Program II, A. W. Bendig, University of Pittsburgh. (CPA 112)

Description: This program permits the assigning of 2 to 600 test items to 1 to 10 subscales so that the program may find subscale scores for each subject. The output includes the mean, standard deviation, and reliability of each subscale, subscale intercorrelations, and subscale correlations with 1 to 5 optional criterion measures. The printed output also lists for each item its mean, standard deviation, correlation with its subscale total score, and correlations with the optional criteria. Subscale scores for individual subjects may be printed and/or punched on cards. Item responses may be scored or weighted if desired. Input-output may be online reader-printer-punch or by tape units.

Computer: IBM 7070 with 10K core storage, floating-point hardware, and either online card reader, printer, and card punch or two online tape units. The program is designed for 7070-1401 combination, but can be adapted to other configurations.

Program language: Locally modified Auto-coder. Symbolic and object decks plus program listing are available from IBM Corporation, Data Processing Division, Program Information Department, 112 East Post Road, White Plains, New York.

IBM 709 program for analysis of variance with repeated measures, D. L. Hartford and F. J. King, Florida State University. (CPA 113)

Description: The purpose of this program is to compute factorial analyses of variance in which all categories or levels of one of the factors contain the same subjects. For example, the levels of one of the factors could be the before-and-after-treatment testings, or even several different tests. The method of analysis is outlined by Lindquist (1953). The total variation is analyzed into between-subjects and within-subjects sources. The between-subjects source of variation is then broken into the main effects and interactions of nonrepeated factors. A between-subjects error term is used to compute the required \( F \) ratios. The within-subjects variation is analyzed into variation due to the repeated factor and its interactions with the other factors. A within-subjects error term is used to compute \( F \)-ratios for these sources of variations.

This program computes: (1) \( F \)-ratios, (2) mean squares, (3) sums of squares, (4) degrees of freedom, and (5) tables of means for up to five levels of as many as four factors including one repeated measure.

There is no limit to the number of values which may be summed into each cell. Variation of the input format is possible and output is self-explanatory. The user has the option of supplying his own wording for the output titles.

Computer: IBM 709 with 32K storage.

Availability: A program writeup, program cards (binary or FORTRAN), and a test problem may be obtained from the Department of Research and Testing, School of Education, Florida State University, Tallahassee, Florida.

Reference:
IBM 7090 program for 4-fold tables, Curtis R. Miller, Pacific State Hospital, Pomona, California. (CPA 114)

Description: This program was expressly written for "bad data," such as input with missing values, odd values used to indicate unknowns, or values that were not intended to be included. The program builds up the 4-fold tables based on cutting points specified by the user. Blanks and/or selected code values (up to nine different ones) may be eliminated from the counts.

Output includes any or all of the following 4-fold statistics desired: (1) chi squares; (2) chi squares with Yates correction; (3) phi coefficients; (4) phi/phi max coefficients; (5) Fisher's exact test (sample size must be under 100); (6) contingency coefficients; (7) coefficients of concordance (Merrifield, 1960); and (8) cosine-pi approximation to tetrachoric r. The 4-fold tables with marginals are an additional option.

Some of these statistics require that certain cells of the ABCD table not be equal to zero. The program checks for this and prints 999.0000 for such cases. It also checks the sample size if Fisher's exact test is requested.

The program was written so that any other 4-fold statistics a user might desire may easily be incorporated without the necessity of extensive recompilation. Only one subroutine need be altered and the new routine for the statistic inserted.

The program will handle 50 variables and 99,999 cases. Because of blank elimination feature, the number of cases for each variable need not be the same.

Computer: IBM 7090 with 32K storage.
Program language: FORTRAN.

Reference:

IBM 7090 Procrustes Program, Curtis R. Miller, Pacific State Hospital, Pomona, California. (CPA 115)

Description: This program was written to handle a problem previously programmed on the ILLIAC by Hurley and Cattell (1962). The basic problem was to find the transformation matrix between any two factor matrices of the same size. This program will accept 100-variable, 30-factor matrices as written.

The output includes the transformation matrix, correlations between the reference vectors, and correlations between the primary factors. An optional matrix to check the calculations is also provided.

Computer: IBM 7090 with 32K storage.
Program language: FORTRAN. Acknowledgement: The programmer is indebted to Dr. Kern Dickman, University of Illinois, for helpful comments on Procrustes.

Reference:

IBM 7090 Factor Score Program, Curtis R. Miller, Pacific State Hospital, Pomona, California. (CPA 116)

Description: This program performs a wide variety of operations having to do with the calculation of factor scores. Any factor matrix up to 20 x 20, orthogonal or oblique, may be used for the calculations. The sample size is limited to 1000 cases; however, options in the program may be used to eliminate this restriction for practical purposes. The method of calculation of factor scores is that given in Thurstone (1947).

Multiple problems may be processed. For any given problem, the means and standard deviations may either be calculated, read in, or retained from the previous problem. The factor matrix may also be read in or retained. If the factor matrix is not orthogonal, the matrix of factor intercorrelations is required as additional input.

The output includes standard scores, factor scores, means and standard deviations of input, and means and standard deviations of factor scores. The latter allows comparisons between subsets of the original sample.

Computer: IBM 7090 with 32K storage.
Program language: FORTRAN and FAP. Acknowledgement: This program incorporates BIMED X33, an inversion subroutine written in FAP. The programmer is indebted to the Biostatistics Unit, University of California, Los Angeles, for making this routine available.

Reference:

An IBM 650 Program written in SOAP for the computation of speech disturbances per
time, speaker, and group,1 Stanley Feldstein and Joseph Jaffe, William Alanson White Institute, New York City. (CPA 117)

The purpose of this specialized program is to compute Ah and Non-ah ratios (two speech disturbance ratios) for any desired unit of time, for each of n interviews, and for each of n groups. Essentially designed to handle 2-person, or dyadic, verbal interaction, the program computes, in addition to the above, separate ratios for each speaker in the dyad. It also calculates the number of words per the chosen time unit. Finally, it yields the dyadic Ah and Non-ah ratios per time unit, interview, and group.

A more detailed description of this program and a list of the instructions are deposited as Document Number 7445 with the ADI Auxiliary Publications Project, Photoduplication Service, Library of Congress, Washington 25, D.C. Advance payment of $3.75 for photoprints, or $2.00 for 35 mm microfilm, should be made in ordering from Chief, Photoduplication Service, Library of Congress.

Frequency Tabulator Program for the IBM 650, Donald L. Hartford, Florida State University. (CPA 118)

This program (entitled FRETAB II) accumulates frequency counts on the occurrence of data values within specified class intervals. It calculates cumulative frequencies, relative frequencies, and percentiles for each class interval, and the mean and standard deviation for each variable. There may be as many as 96 class intervals per variable and up to 8 variables may be tabulated in one run. The data values must be five digits or less in size.

Computer: FRETAB II requires an IBM 650 equipped with 60 words of core storage and three index registers. Provision is made for selecting an online printer (407) for output if one is available. Output tables are labeled and the class boundaries are printed to the left of the data for the class. The user may provide a title for each study processed, and multile processing is possible.

Running time: A test deck of 96 intervals with seven variables and identification was run. The program takes approximately 15 seconds per card to tabulate seven variables on the 96th interval. Tabulating seven variables on the 21st step interval takes about five seconds per card. Lower values run more quickly. Over 450 studies have been run using this program including some distributions in which all the data were missing.

Operating write-ups (including I/O wiring specifications), SOAP listings, and 7-per-card program decks are available from the author. Write to the Department of Research and Testing, Florida State University, 426 Education Building, Tallahassee, Florida.

1 Supported by USPHS Grants M-4548 and M-4571.

Scientific questions have their origin deep in human consciousness, often below the analytical level. They constitute specialized restatements of large questions that philosophers formulated long before scientists began to work on their determinism, questions which have preoccupied men ever since they began to think—even before the beginnings of formal philosophy.

RENÉ DUBOS, The Dreams of Reason