Who are the users?—An analysis of computer use in a university computer center*

by EARL HUNT, GEORGE DIEHR and DAVID GARNATZ

University of Washington
Seattle, Washington

INTRODUCTION

This is a study of how the users of the University of Washington computing center exercise its machinery. Our hope is to make an undramatic but useful contribution to knowledge. In a simpler day the distinction was made between "scientific" and "business" computing. Undoubtedly this contrast is still useful for many purposes, but finer distinctions are needed. We shall present statistics showing that, within a community which contains not a single "business" user, there are distinct groups with quite different machine requirements. Of course, nobody who is aware of modern computing would seriously dispute this. Our contribution is to provide statistics on the relative size of the different groups. We also offer this report as an example of methodology. The usefulness of our numbers to another center will depend upon the extent to which the other center is like ours. The ways in which we acquired and analyzed our statistics would be useful more generally.

From the viewpoint of the Computer Center, a knowledge of user characteristics is important in planning. In the particular center we studied, and others like it, there will probably be no major change in the types of computing done over the next five years (unless qualitatively different equipment capabilities are provided), but there will be a steady increase in the number of users. The characteristics of this increasing population must be known in order to anticipate bottlenecks and to plan for orderly expansion. Users also need to know something about themselves. Time is expensive, so computer use must be estimated as accurately as possible in budget preparation. In the days before multiprogramming, one simply rented the entire computer configuration for a few seconds, even if only half of it was used. Today charges are based upon use of memory size, processor time, and peripherals. To make accurate estimates of his needs, the user must ask "What resources do people like me actually utilize?" Consider the problem of the instructor trying to estimate the cost of a course in programming. What he knows is that he will have n students, k problems, and that the problems will take an average of m runs to solve. These runs vary greatly as the students progress from incompetence with control cards to an ability to write infinite loops. To estimate the cost of computing, the instructor needs statistics about how student jobs perform. The research scientist who has not yet settled on his batch of production programs (and who may never find them) is in a similar situation. He knows how many people he has on his project and knows how often they submit programs. He also knows that the programs vary greatly as he and his associates go through cycles of planning, debugging, production modification, and reprogramming. To estimate his budget he needs *applicable* averages.

As our final justification, we point to an application of user statistics within Computer Science. The use of models to predict system performance has become increasingly popular in system evaluation. Basically, the idea is to view a computing configuration as a job shop servicing jobs drawn at random from a population of users, and then to analyze a model of such a service. In order to make the model anything more than an exercise in mathematics, however, one must show a correspondence between it and reality. Here we present some statistics which can be appealed to in justifying a model of the user.

THE UNIVERSITY AND THE CENTER

Some words about the setting of our study are in order. The University of Washington is a large state

* This research was supported by the Institutional Research Fund of the University of Washington, Seattle, Washington.

university with about 33,000 students, 20 percent of them in graduate or professional schools, and a faculty of roughly 2,500. The University computer center provides general support for this community. Specialized research computing capabilities needed for process control or real time applications are provided by dedicated installations scattered throughout the campus. We did not study these. The University's administrative data processing is done on a dedicated Burroughs B5500 computer, and hence is also not included in this study.

The center's "scientific" computer is a Control Data 6400 system with 65,000 sixty bit words. It is used in batch mode under control of the SCOPE 3 operating system. Systems and library applications programs reside on a 132 M character CDC 6638 disk, which is also available for user temporary files. Every time that a job requires service from the operating system an appropriate message is recorded on the DAYFILE, a log maintained by the SCOPE system. To obtain our data we sampled several copies of the DAYFILE, recording the following information.

1. Job identification: The code used in job identification distinguishes between graduates, undergraduates, and faculty, and between jobs associated with classwork and jobs associated with research projects. The technique of financial control in the system discourages the use of class numbers for research jobs and vice versa.
2. Central processor time used
3. Peripheral processor time used
4. Priority of job at time it is run (0-low priority to 7-high priority)
5. Number of tape drives charged for
6. Charges assigned
7. Whether the job is a FORTRAN or non-FORTRAN
8. Number of lines printed
9. Number of cards read
10. Amount of central memory used, expressed as a percentage of 32 K words.

Three different statistical techniques were used. One was a simple summary of the statistical characteristics of each of the nine measurements of the aggregate sample, obtained by plotting a histogram and preparing tables of measures of the central tendencies and dispersion statistics, using BMD01D program to do this. The correlations between the different measures were computed using the BMD03M program (Dixon, 1965). This program provided correlation matrices and a factor analysis using the variance maximization criterion (Harman, 1960) to define orthogonal factors. Finally, a cluster analysis (Diehr, 1969) was performed to see if the jobs analyzed fell into groups of similar jobs. The cluster analysis algorithm used grouped the observations into a fixed number of groups called *clusters*, such that the sum of squared distances from observations to their cluster means was minimized. The algorithm will be described in detail in a moment.

TABLE I—Descriptive Statistics for 1588 Jobs Submitted to CDC 6400

Measure	Mean	Standard Deviation
Cards read	224	495
Lines printed	760	1260
CPU time (sec.)	11.0	41
PPU time (sec.)	11.9	35
Central memory	55.8	25.4
Tape drives charged	.28	.55
Cost to user	1.44	4.10
Percent jobs using Fortran	.54	.50

DESCRIPTIVE STATISTICS RESULTS

Table I presents descriptive statistics for 1588 jobs selected from first shifts.* We shall discuss this sample extensively. Similar analysis of second shift data and data from a different time of the year produced very similar results. Therefore, virtually all of our remarks will be concerned with an analysis of these jobs.

Whether Table I presents a true or false picture of the user community depends on the purpose for which the examination is conducted. It shows what sort of use is made of computing by the "average" user. This hypothetical individual submits what most people intuitively familiar with the center would consider a medium-sized job, reading about 200 cards, printing 700 to 800 lines (about eight pages plus system output), and using around eleven seconds each of cpu and ppu time. Slightly more than half of the jobs execute a Fortran compilation. Like the man with 2.4 children, the average user is not the typical user! Frequency plots of the variables CARDS READ, LINES PRINTED, CP TIME, PP TIME, and COST showed that the distributions were positively skewed with means in regions of very low density, suggesting that (a) mean values were not good descriptors of the popu-

* In obtaining the 1588 teaching and research jobs we also encountered on DAYFILE logs a record of 364 miscellaneous jobs. These included jobs generated by the computer center itself, administrative work for some reason not done on the B5500, and an occasional commercial user. Because this group of jobs was so heterogeneous it was not further included in the analysis.

lation and (b) that the observations were exponentially distributed. If the second conclusion had been correct, logarithmic transformations of the indicated variables would have produced symmetric distributions. In fact, they did not. This is illustrated in Figure 1, which is a frequency histogram for the *logarithm* of CP time. (The other four variables listed above were similarly distributed, while MEMORY USE was symmetric originally and TAPE DRIVES CHARGED and FORTRAN use are discrete.) Both the mean value of the transformed CP time and the logarithmic value of the mean of the untransformed time are shown. It can be seen that neither figure is an accurate descriptor. The frequency distributions were positively skewed even after the transformation and, in some cases, appeared to be bimodal. This strongly suggests that instead of regarding jobs as being generated by a single process, the jobs should be thought of as being a mixture of two or more populations which *individually* might be satisfactorily characterized by standard descriptors of central tendency and dispersion.

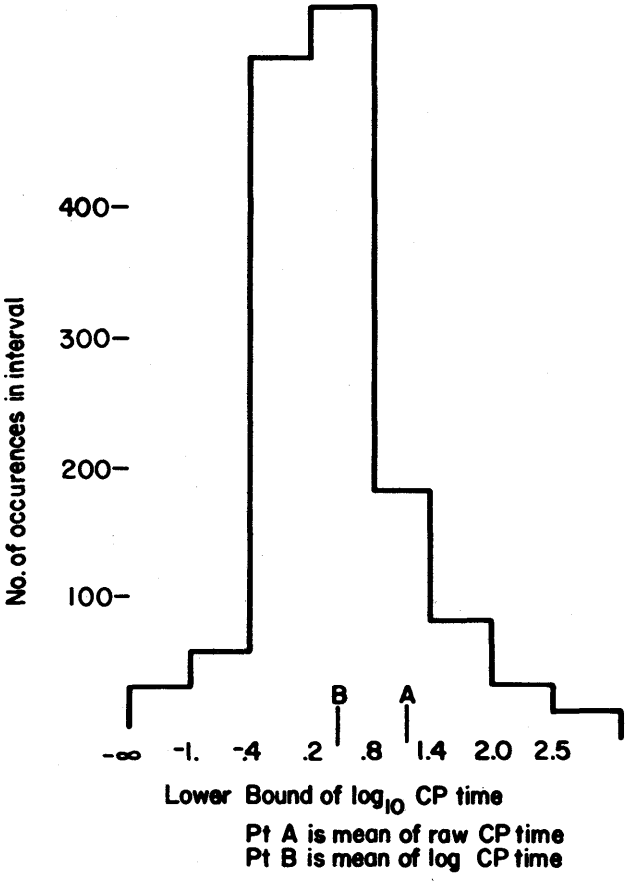

Figure 1—Frequency histogram of $\log_{10}$ CP time

TABLE II—Mean Values of Each Measurement, for Total Sample, Research, and Instructional Job Numbers

Variable	Overall Mean	Research Job Mean	Instructional Job Mean
Cards read	224	490	95
Lines printed	760	1430	442
CPU time	11.0	26	3.8
PPU time	11.9	22	7.1
Central memory	55.8	66.0	51.0
Tape drives	.28	.4	.22
Cost	1.44	3.40	.48
Percent jobs using Fortran	.54	.73	.44
Priority of run	.016	.04	.004
Number of jobs	1588	527	1061

To investigate this hypothesis, we first divided the sample into two groups, jobs associated with research projects and jobs associated with instruction. It was immediately clear that this was, indeed, a reasonable distinction. Table II shows the means on each measure for the sample as a whole and by subgroups. On the average, the difference between subgroup *means* exceeds one standard deviation about the sample mean, thus clearly supporting the hypothesis that there are two distinct subgroups.

One is tempted to say, "Of course, why bother to measure such an obvious thing?" We would expect to find differences between instructional and research work, although our intuition is not very good at predicting the fine detail of these differences. We also found, however, that this simple division is not enough—averages do not describe the typical research or instructional job either! Examination of the histograms within classes based on the research-instruction distinction again showed distributions similar to Figure 1. We therefore eschewed our intuition and turned to an "automatic" method of dividing jobs into homogeneous groups, using cluster analysis.

CLUSTER ANALYSIS RESULTS

The purpose of a cluster analysis is to group observations into k subclasses such that, in some sense, the differences between members of the same class is small relative to the differences between members of different classes. The particular cluster analysis technique we used regards each observation as a point in n dimensional Euclidean space. Observations are assigned to a predetermined number of groups (*clusters*) in such a way that the sum of squares of the distances of points to their cluster mean point is minimized. Thus the cluster analysis is bound to produce groups for which

TABLE III—Mean Values for Measures—Two Clusters Compared to Research and Instructional Jobs

Variable	Group			
	Cluster 1	Instruction	Cluster 2	Research
Log cards read	3.9	3.8	5.4	5.6
Log lines printed	5.6	5.5	6.3	6.5
Log CP time	—.09	—.5	1.9	2.3
Log PP time	1.6	1.5	2.4	2.5
Memory use	51	48	67	70
Tape drives	.22	.19	.40	.44
Log cost	−2.0	−2.1	.53	.50
Percent Fortran use	.44	.44	.74	.72

central tendency measures are reasonable descriptors, while the standard deviation within a cluster indicates how much variation there is about the mean point. The algorithm begins with all observations in a single cluster around the population mean point. A second cluster is initiated whose first member is the observation furthest away from the center point. An iteration phase follows, in which each observation is assigned to one or the other cluster by making the choice which minimizes the sum of squares about cluster points. The cluster mean point is adjusted as the observation is grouped. The iteration is continued until no further adjustments are made. A new cluster is then initiated by choosing as its first member that observation which is furthest from the mean point of the cluster to which it is now assigned. The iteration is then repeated. The entire process is continued until the predetermined number of clusters is obtained.

While a stable partition represents a local minimum by the sum of squares criterion, it is not necessarily a global minimum. Extensive experimentation with this algorithm in comparison to several other clustering methods has indicated that it consistently finds good clusters (Diehr, 1969). Our only reservation is that because a minimum variance criterion is being used, one wants to avoid situations in which the means and variances of the partitions are correlated. Fortunately, this can be achieved by using logarithmic transformations of highly skewed variables (in this case CARDS READ, LINES PRINTED, CP TIME, PP TIME, and COST). Accordingly these variables were included after a logarithmic transformation. The variables MEMORY USE, TAPE USE, and FORTRAN USE were included but not transformed.

If the research-instructional distinction is a valid one, then a clustering into two classes should recreate it. This is, indeed, what happens. Table III shows the means and standard deviations for two clusters, compared to the breakdown of jobs by research or instructional sources. Table IV shows a cross classification of jobs both by their origin and the cluster into which they fall. Almost 90 percent of the instructional jobs fall into the first cluster, while about 75 percent of the research jobs fall into the second cluster.

While this confirms our faith in the research-instruction distinction, it still leaves us with too gross an analysis. Clusterings into from two to six groups provided a significant insight into the data. Let us describe the results of these successive clusterings briefly.

Three groups: The data was partitioned into small, medium, and large resource use groups. The small job group is largely classwork jobs, the large usage group largely research jobs, and the medium usage group made up of half research-half classwork jobs. There is no indication of sub-populations which have heavy I/O use but light processor use (i.e., no "scientific business" breakdown).

Four groups: The data was partitioned into two groups of jobs with small resource use; differentiated only by use or non-use of the FORTRAN compiler. The other two groups were jobs with medium to large system resource use and "aborted" jobs. The medium-large usage group is similar to the medium-large usage group found for two clusters. The group of aborted jobs tends to be small in terms of I/O requirements, and had virtually no CP use.

Five groups: This clustering separated a group of large jobs using tape drives from the four groups described above.

Six groups: This is perhaps the most interesting clustering. Two levels of system resource use were uncovered, with three types of jobs within each level. There were three types of small jobs; 408 FORTRAN and 472 non-FORTRAN jobs, and 89 aborted jobs. The small job groups were primarily instructional, and included jobs using a BASIC interpreter. The aborted jobs were almost all terminated due to control card errors. It is interesting to note that such errors apparently occur on about 5 percent of the jobs submitted.

The medium to large job groups included 181 medium-sized jobs using tape drives, 293 medium to large jobs which did not use tapes, and 148 very large jobs.

TABLE IV—Cross Classification of Jobs by Cluster and Administrative Source

		Administrative Source	
		Instruction	Research
Cluster	1	884	144
	2	187	373

We feel that the most interesting contrasts are between (a) the population statistics, (b) the statistics for the two-cluster (research-instruction) partition, and (c) the finer data of the six group clustering. Figure 2 is a graphic summary of what one sees if jobs are regarded as coming from one, two, or six populations. In this figure each cluster is represented as a rectangle. The following information is coded in the figure:

1. The area of the rectangle drawn for the group is proportional to the number of jobs within it.
2. The shading indicates the number of research jobs—i.e., a completely shaded rectangle would represent a group containing only research jobs, while an unshaded rectangle would represent a group of instructional jobs.
3. The horizontal axis shows the average number of standard deviations between a group mean and the population mean on each of the resource variables. Thus the "partition" consisting of all 1588 jobs has its rectangle centered at 0.0 on the horizontal axis, while clusters containing large resource use jobs are centered to the right of this point, and those containing small jobs are centered to the left.
4. The vertical axis indicates the number of groups (1, 2, and 6) on which the partition is based and, within the region for a given number of groups, the fraction of FORTRAN jobs. Thus one can determine that the 1028 "small" jobs in the two groups clustering contained approximately 45 percent FORTRAN jobs, while the "medium-large" jobs were 75 percent FORTRAN by examining the vertical position of the appropriate rectangles.

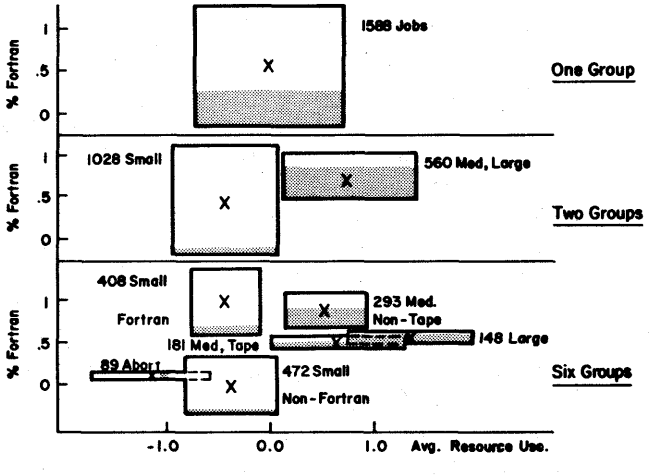

Figure 2—Graphic summary of six cluster result—see text for explanation of code

TABLE V—Correlations Between Variables Based on 1588 Cases

Variable	1	2	3	4	5	6	7
1. Log cards read	1.00	.42	.51	.39	.36	.02	.62
2. Log lines printed		1.00	.46	.39	.22	.10	.50
3. Log CP time			1.00	.53	.46	.16	.75
4. Log PP time				1.00	.18	.41	.71
5. Memory use					1.00	.10	.43
6. Tape drives						1.00	.31
7. Log cost							1.00

5. The length of the rectangle indicates the average variation on the system use variables, with 0.8 std. dev. used as a basis. Thus, it is evident that for six groups the "small-non-FORTRAN" jobs had a slightly greater variation on the average than the "small-FORTRAN" jobs. The length of the rectangles also shows that the "med-non-tape" jobs are better defined than either the "med-tape" jobs or the "large" jobs.

CORRELATION ANALYSIS

The cluster and descriptive analyses dealt with the relations between jobs. Another way to analyze our statistics is to look at the relationship between variables. The table of correlation coefficients for all variables was computed and factor analyzed. The analysis was performed separately for the different classes of user and for all cases together. Since there was no substantial difference in either the correlation or factor matrixes, only the overall picture will be discussed.

Before performing the correlation analysis a certain amount of data editing was done. The distinction between FORTRAN and non-FORTRAN jobs and the priority measures were dropped, and a logarithmic transformation was performed on all other variables. The logarithmic transformation was used because all variables were either exponentially distributed or had a number of cases with extreme values. High or low correlation coefficients based on untransformed data, then, might be produced by only a few cases. The use of the logarithmic transformation greatly reduces the chance of this occurring.

The correlations between the variables are shown in Table V. The table of untransformed variables presents substantially the same appearance except that the extreme values are somewhat higher. The picture of correlations is not immediately clear. It becomes so, however, when one looks at Table VI, which shows the

TABLE VI—Factor Loadings for Variables on First 3 Factors

Variable	Factor 1	Factor 2	Factor 3
Log cards read	.72	.35	.12
Log lines printed	.65	.14	.45
Log CP time	.84	.13	−.07
Log PP time	.76	−.40	.18
Memory use	.55	.34	−.71
Tape drives	.35	−.83	−.26
Log cost	.92	−.05	.02
Cumulative % variance	50	66	77

factor loadings for each of the variables on each possible factor*. Only the first three factors will be discussed. These account for better than 75 percent of the total variance. The first, and by far the largest (50 percent) factor can be thought of as a "standard job" factor. It accounts for half or more of the variance in cards read, lines printed, and central and peripheral processor time. Our interpretation is that this factor is produced by the correlated variation in the measures used by most jobs. The second factor is essentially a "tape request" factor (note the high loading of "requests"), and reflects a difference between jobs that do or do not use tapes. The third factor has its heaviest loading on memory use. It reflects variation in memory use by some jobs that lie outside of the normal spectrum of computing (i.e., outside the range covered by factor 1). This is probably caused by (a) jobs that have control card errors and hence use little memory and (b) a few research jobs that utilize memory heavily.

* Since factor analysis may not be familiar to all readers, we shall explain a way to interpret its results. For further details see Harman (1960) or Morrison (1965).

Suppose each job were plotted as a point in 7 dimensional space. Since most measures are exponential, conversion to a logarithmic scale ensures that the swarm of points will be roughly a hyperellipse. The factors can be thought of as the axes of the hyperellipse. The first factor is the major axis, the second factor the next longest axis, etc. The 'percent variance extracted' by each factor is the percent of variance in distances from the centroid of the ellipse associated with projections on the factor in question. The loading of a variable on a factor can be interpreted in the following way. Suppose each point is plotted on a chart of variables against factor. Note that these will not generally be orthogonal axes. The square of the loading of the variable on the factor is the fraction of variance in the variable associated with variance in the factor. Alternately, the loading can be thought of as the correlation between the variable and a hypothetical pure test of the factor.

In general, the factor analysis supports the other statistics we have gathered. An interesting point is the low loading for memory use on factor 1, which indicates that most jobs have a uniform memory requirement. This could be quite important in designing memory allocation algorithms in multi-programming systems.

SPECIFIC QUESTIONS

The statistical analysis raised a number of non-statistical questions about jobs, and particularly about jobs that were not typical of their administrative category. To answer these a special cross tabulation program was written, modeled after a more extensive information retrieval system designed by Finke (1970) and used to sort jobs in various ways. Some of the specific questions and their answers were as follows:

Q.1. How does the use of FORTRAN or BASIC affect instructional jobs?

A. BASIC jobs use less processor time than FORTRAN but, on the average, much more memory than the average for *instructional* jobs. Research jobs virtually never use BASIC except for relatively small jobs.

Q.2. What percent of memory is used by the "average" job?

A. Better than half the teaching jobs use less than 16K words. The comparable "break even" point for research jobs is 24K. Twelve percent of the research jobs use more than 32K words, while less than two percent of instructional jobs do. Furthermore, most of the long instructional jobs are generated by a few individuals (i.e., are multiple jobs with the same user I.D.).

Q.3. How many runs are compiler runs of any sort? What compilers were used?

A. About two-thirds of the jobs call for at least one compilation. In 1588 runs the FORTRAN compiler was called 849 times, BASIC 249 times, SNOBOL once, SIMSCRIPT 13 times, the COMPASS assembler *twice* (by the same job number) and COBOL and ALGOL never. (Excellent COBOL and ALGOL systems are available on the University's B5500, so this may be misleading.) Only three center supported "packages" were used: the BMD statistical programs, the SMIS package, and a SORT-MERGE system, for a total of 69 runs. One wonders two things: how much effort are users devoting to duplicating library programs and how much effort should a computer center devote to maintaining such programs?

CONCLUSIONS

Our specific conclusions are that the University of Washington Computer Center users create jobs that fall into four groups, some with important subgroups, producing the six groups graphed in Figure 2. The four major groups are aborted jobs, small jobs (with subgroups FORTRAN and non-FORTRAN), medium-sized jobs (with subgroups tape and non-tape jobs), and large jobs. Small jobs are primarily due to classroom work, middle and large jobs are associated with research work. The principal ways in which jobs differ from each other is in the amount of processor time used and the amount of input. These statistics, which are not terribly startling, are of direct use to the University of Washington and are of indirect use to any institution which is willing to assume it is like Washington.

Should our analysis be used generally even though our particular results are not general? To answer this, we will point out two courses of action which are available to the University of Washington now that it has these statistics, but which might not have been available (or at least, would have been available only by trusting the Computer Center Director's intuition) without the analysis.

At most universities computer use of education is supported by intramural funds, while a substantial part of the research computing support is extramural. Understandably, granting agencies (notably the United States Government) insist that the same algorithm be used to allot charges to all users. The argument is that the cost of a computation should be determined by the nature of the computation and not by who makes it. While seemingly fair, this can be frustrating to an institution which wishes to encourage educational use of computing, but needs to capture all funds that are available for research computing. More generally, there are a number of situations in which a computer center may wish to encourage or discourage certain classes of user, while still retaining the principle that the same charge will be levied for the same service. The solution proposed is to establish a charging algorithm which is sensitive to the varying characteristics of jobs from different user sources. For example, if the University of Washington were to place a very low charge for the first 200 cards read and the first 10 seconds each of CP and PP time, and charge considerably for system utilization beyond these limits, then the educational users would pay proportionately less and the research users proportionately more of the total bill. Charges would still be non-discriminatory in the sense that identical jobs receive identical bills. Note also, how our statistical analysis dictates the type of charging algorithm. From the correlational analysis we know that the only way of differentially effecting user charges is to manipulate the number of cards read and the processor time charges. From the descriptive statistics and the cluster analysis we can predict how a given manipulation will affect different sections of the user community.

Very much the same reasoning can be used in planning for new equipment acquisition. Obviously equipment additions aid in computing either because they facilitate the running of all jobs equally (in which case the aim is to increase throughput uniformly) or because they aid in processing of certain types of jobs. The computer center director rightly looks at equipment in terms of how it affects bottlenecks in his throughput or in his capability to do certain types of computation. From the University administrators' view, however money into the computing center is a means toward the end of achieving some educational or scholarly goal, such as increased production of engineering B.S.'s or support of a Geophysics research program. We can use a statistical analysis of user characteristics to reconcile these points of view. Taking an obvious example from our data, if the University of Washington decides to put x dollars into support of computing for education, the money should not be spent buying tape drives. To take a more subtle case, suppose we were faced with a choice of obtaining a medium-sized computer or expanding the CDC 6400 system to a CDC 6500 or CDC 6600 computer. The appropriate course of action might be determined by the purpose for which the money is intended, to facilitate educational or research use. Without these statistics, we do not see how the management goals of the institution and the technical goals of the Computer Center can be coordinated.

Our results also are of interest to two groups of people outside of our own institution; those interested in research on computing systems and those involved in selling computers to universities. We feel that we have clearly shown that a simple model of a single process for generating statistical characteristics of user jobs—such as those assumed to estimate the performance of system algorithms—is not appropriate. A model of a university computing community must be based on sub-models for quite different populations. The business-scientific distinction is decidedly not appropriate.

We close with some remarks on the methods we have used. Two of our techniques, descriptive statistics and correlational analysis, are conventional statistical methods. Indeed, the programs we used, BMD03M and BMD01D, are part of the most widely supported applications package in programming! There is no reason why everyone with a computer of any size could not perform these analyses on his job stream. We feel, however, that the clearest picture of our users was obtained

by the less conventional cluster analysis. We recommend that this technique be used more widely to analyze computer use. We hope it will aid in identifying the characteristics of existential, rather than postulated, computer users.

REFERENCES

1 W DIXON ed
Biomedical computer programs
Health Sciences Computing Facility UCLA 1965

2 H HARMAN
Modern factor analysis
U Chicago Press 1960

3 D F Morrison
Multivariate Statistical Methods. New York: McGraw-Hill, 1965

4 G DIEHR
An investigation of computational algorithms for aggregation problems
Western Management Science Institute UCLA Working Paper 155 1969

5 J FINKE
A users guide to the operation of an information storage and retrieval system on the B5500 Computer
Technical Report No. 70–1–3
Computer Science Univ of Washington