ESD-TR-66-350

DEVELOPMENT OF EQUATIONS FOR ESTIMATING THE COSTS OF COMPUTER PROGRAM PRODUCTION

V. LaBolle

JUNE 1966

DEPUTY FOR ENGINEERING & TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
United States Air Force
L. G. Hanscom Field, Bedford, Massachusetts

(Prepared by System Development Corp, 2500 Colorado Ave., Santa Monica, California under Contract AF19(628)-5166.)

ESD-TR-66-350

DEVELOPMENT OF EQUATIONS FOR ESTIMATING THE COSTS OF COMPUTER PROGRAM PRODUCTION

V. LaBolle

JUNE 1966

DEPUTY FOR ENGINEERING & TECHNOLOGY
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
United States Air Force
L. G. Hanscom Field, Bedford, Massachusetts

(Prepared by System Development Corp, 2500 Colorado Ave., Santa Monica, California under Contract AF19(628)-5166.)

FOREWORD

This report was prepared for the Electronic Systems Division (ESD), Air Force Systems Command, as part of a continuing research effort at System Development Corporation (SDC) to develop improved guidelines, standards, and techniques for management of computer programming. A major task in this work has been the derivation of equations for estimating the costs of computer program production, i.e., program design, code, and test. This work includes statistical analyses of numerical data that measure costs and cost factors for completed programming efforts. This analysis work is being done in cycles. Each succeeding cycle, aimed at improving earlier results, corresponds to the collection of new numerical data and their subsequent analysis. In the first cycle, data for 27 programming efforts (data points) completed at SDC were analyzed and the results were reported in the fall of 1964. The major part of the second cycle, an analysis of 74 data points also representing SDC programming work, was reported in detail in TM-2712, Research Into the Management of Computer Programming: A Transitional Analysis of Cost Estimation Techniques, and is summarized in this document along with subsequent work done to improve the earlier results in the second cycle. A third cycle, including analysis of numerical data from Air Force and industrial organizations as well as the SDC data used earlier, is now under way. Results of this new analysis will be available in the fall of 1966.

The research reported in this document was conducted by members of the Programming Management Project. G. Weinwurm and H. Zagorski were the chief investigators for the work in the second cycle as reported in SDC Technical Memorandum TM-2712. They were supported by T. Fleishman and E. Nelson. The extensions of the second cycle reported here have been conducted by T. Fleishman, E. Nelson, and H. Zagorski. All of these project members contributed to the preparation of this document.

REVIEW AND APPROVAL:

This technical report has been reviewed and is approved.

CHARLES A LAUSTRUP, Col, USAF
Director of Computers, Deputy for
Engineering & Technology

ABSTRACT

This report summarizes System Development Corporation (SDC) Technical Memorandum TM-2712, Research Into the Management of Computer Programming: A Transitional Analysis of Cost Estimation Techniques, 12 November 1965. That report supplies details of the early results obtained in a second cycle of continuing exploratory research to develop equations for estimating the costs of computer program production--computer program design, code, and test. Additional sets of equations developed after TM-2712 was published are also given in this report. Each set contains four equations; each equation shows how to form an estimate for one of the cost measures--number of man months, computer hours, new machine language instructions, months elapsed--by combining numerical values for selected factors that influence these costs.

This report reviews the development of these equations including the application of statistical methods such as correlation and multivariate regression to experience data that characterize 74 computer programming efforts completed at SDC. The earlier work in the first cycle, a similar analysis of data for 27 SDC computer programming efforts, is also described as well as the plans for the current analysis in the third cycle using these SDC data and new data for more than 80 efforts completed by computer programming organizations in industry and the Air Force.

After the publication of TM-2712, the second cycle was continued by additional analysis of the same SDC data for 74 computer programming efforts. The aim of the additional work was to improve the estimating precision of the equations presented in TM-2712. The improvements reported were achieved by deriving new cost equations, one set based upon a truncated sample and then three sets based upon three subsamples of the data. An interim evaluation of the work completed in the first and second cycles presents proposed improvements in approach and research methods.

CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

SECTION I

INTRODUCTION

This report reviews the research work to develop estimating equations for costs in computer program production. Several sets of equations are presented as current results from an exploratory effort to forecast costs based upon knowledge of the requirements for a computer program and the resources expected to be available for its production. The equations are rules, then, for using numerical values of the cost factors that characterize the requirements, resources, and environment for a computer programming effort, to calculate estimates for costs such as manpower measured in man months and computer time measured in hours. Similar equations are given for estimating the number of new machine language instructions that will be developed in a computer program and the months that would elapse to complete the effort. Numerical relationships for pairs of the cost measures, e.g., computer hours as a function of man months, are also presented.

The development of these equations is part of a continuing effort being performed under contract with the Air Force Electronic Systems Division (ESD) by the Programming Management Project at System Development Corporation (SDC). The general objective of the Programming Management Project is to conduct research aimed at developing tools and guidelines for both managers of computer programming and buyers of the resulting products. The particular effort for ESD includes collection of numerical data on costs and cost factors that describe completed programming projects and subsequent statistical analyses of these data to derive estimating equations that could help managers plan computer programming efforts more accurately. As such, the equations and other guidelines that emerge from the analysis are aimed primarily at use in the early stages of computer programming, e.g., prior to beginning program design. However, the results may also be used to evaluate completed efforts by comparing actual costs with estimates in a framework provided by the estimating equations. Future analyses that may identify appropriate factors could provide some guides for more effective cost control. Eventually we expect results of this cost analysis to be coupled with analyses to measure worth or value and so supply inputs for cost/value comparisons of programming efforts.

The analysis work has been conducted in cycles, each marked by collection and analysis of new data to improve upon earlier results. A cycle of analysis consists of the following:

- Design (or redesign) of the questionnaire used to collect the data.
- Collection of data that characterize completed programming efforts from various programming organizations.

- Validation of these data by identifying anomalies and gaps and then coordinating with the original respondents to clarify and complete the questionnaires.

- Application of statistical techniques, intuition, and experience, first to reduce the total number of cost factors to be considered as independent variables, and then to derive the equations. This is done, for example, by using multiple regression and predictor selection algorithms to relate the remaining cost factors as independent variables to the cost measures as dependent variables.

These statistical techniques, such as the regression procedures, are repeated many times in each cycle to test various hypotheses about the relationships between the cost factors and the cost measures.

The equations reported here are based on the second cycle of data collection and analysis that used a data base characterizing 74 programming efforts completed at SDC. A first cycle that used data on 27 SDC programming efforts was completed in 1964; a third cycle, using additional data from 104 programming efforts completed by computer programming organizations in the Air Force and in industry, is under way. The new data in the second cycle were collected in the winter of 1964 and spring of 1965, and were analyzed in the following summer and fall.

The work to date has been recorded in the following documents:

- TM-1447/000/02, Factors that Affect the Cost of Computer Programming, L. Farr and B. Nanus, 30 June 1964--a first formulation of presumed cost factors.

- TM-1447/001/00, Factors that Affect the Cost of Computer Programming: A Quantitative Analysis, L. Farr and H. J. Zagorski, 31 August 1964--a research report on the work in the first cycle.

- TM-1447/002/00, A Summary of an Analysis of Computer Programming Cost Factors, L. Farr and H. J. Zagorski, 25 January 1965--a summary of the work in the first cycle.

- TM-2712/000/00, Research Into the Management of Computer Programming: A Transitional Analysis of Cost Estimation Techniques, G. F. Weinwurm and H. J. Zagorski, 12 November 1965--a research report on the work in the second cycle.

This report reviews the analysis in the second cycle and consists of the following:

- Section II--a review of the research work described in TM-2712.

- Section III--a description of the current work including the improved equations derived since publication of TM-2712 and the plans for the new analysis in the third cycle.

- Section IV--an interim evaluation of the research results to date and the methods used to obtain them.

After the publication of TM-2712, additional analysis was done to improve the precision of the estimating equations described in that report. These new equations are also presented here as the latest results of the work.

Specifically, four different sets of equations are supplied in this report. Each set has an equation for each cost measure--number of man months, computer hours, new machine language instructions, and months elapsed. The first two sets of equations are described in Section II, the review of the analysis. They are based upon the entire 74-data-point sample and supply estimates for the cost measures in logarithmic form, e.g., $\log_{10}$ man months. The other two sets of equations described in Section III, Current Work, were derived from parts of the sample and give estimating rules for the conventional form of the cost measures, e.g., computer hours.

The first set of equations in Section III was derived from a truncated sample of 67 data points obtained by dropping the data for the seven programming efforts with the largest costs. To derive the second set in Section III, the truncated sample of 67 data points was divided into subsamples. This division was based upon size of the cost measure, man months; equations were derived for subsamples corresponding to the ranges 1 to 9, 10 to 79, and 80 to 260 man months for each of the four cost variables--man months, computer hours, new machine language instructions, and months elapsed. Therefore, the last set in Section III actually consists of 12 equations--three subsets of four equations for the three ranges of man months.

The four sets of equations have different characteristics. The two sets in Section II, based upon the entire sample of 74 data points, were derived using more complete analytical procedures than the sets given in Section III. But they yield estimates of the cost measures in logrithmic form that are not as easy to use as the conventional form found in the equations of Section III. The first set of equations in Section III is based upon a truncated sample; the larger cost data points have been dropped to improve the estimating precision of the equations. (Estimating precision refers to the expected accuracy of the equations, i.e., given an estimate calculated by an equation, how small is the band or range in which we expect the actual value to occur for a fixed, stated percentage of the time?) The same cost factors as those used in Section II were retained as predictor variables in this extended analysis without the benefit of an extensive search being made for preferred predictors. Similarly, the last set of equations in Section III, based upon subsamples of the truncated data base, used the same predictor variables again and represents further improvement in estimating precision. Without benefit of cross-validation, perhaps the

best equations among the four for estimating medium to large programming jobs is the first set in Section III, derived from the truncated sample. These equations are easy to use and show considerable improvement in estimating precision. At the same time, because the sample used is mostly the same as that used earlier, the equations preserve the gains made by the exhaustive statistical procedures used in the analysis to derive the equations in TM-2712 (those in Section II). For small jobs, the equations in Section III that were derived from the subsample with a man-month range from one to nine, although they are based upon a small sample, are considered to have the best estimating precision.

All of the estimating equations presented in this report are interim results derived from continuing research. They have not been cross-validated, i.e., applied to a new sample of data for programming projects to calculate estimates and then to compare these estimates for costs of program production with the actual costs. So at this time, we have no means of determining whether these equations are better on the average than other techniques including the intuition and experience commonly used by managers in computer programming.

We recommend that the use of any equations in these four sets be restricted to comparison with estimates made by other means rather than as an exclusive means for estimation. We would like to encourage the use of the equations in this way and would appreciate comments on experience with them.

SECTION II

A REVIEW OF THE ANALYSIS OF COMPUTER PROGRAMMING COSTS

In this Section, we review the second cycle of this analysis as reported in TM-2712. To introduce this review we outline the underlying assumptions and the work in the first cycle of analysis.

1. The Model

In performing these analyses on costs, we have assumed that computer programming, regardless of application and resources used, has certain common characteristics that can be generalized. This premise of commonality was coupled with the following assumptions to form the model for the analysis:

a. Despite the differences in practice and application, computer programming is considered systematic enough that its variation in the costs from job to job can be accounted for mainly by the variation in a selected set of cost factors.

b. These cost factors that explain variation in the costs of computer programming can be selected from a comprehensive collection which has been divided into three groups corresponding to requirements, resources, and environment for computer programming as shown in Table I.

c. The primary costs, manpower, measured in man months, and computer use, measured in machine hours, can be considered as dependent variables that can be predicted by a linear combination of cost factors used as independent variables. Values for these variables can be obtained as numerical answers to items in a survey questionnaire to be completed by knowledgeable individuals associated with a particular effort.

d. The analyses performed to derive estimating equations are restricted to program production costs, i.e., those that are incurred in program design, code, and test activities. These activities include associated documentation as well as work on the data base. (This particular set of activities was chosen because they appear to be common to almost all computer programming work. Therefore, the scope of the work to date does not include activities that may constitute a more generalized model of computer programming associated with large information processing systems that include men, machines, and computer programs as components or subsystems. For such information processing systems, the programming activities, for example, the system design and analysis that may precede computer program design and test of the total system that may follow test of the computer program components, have been deliberately excluded to help us collect a large, consistent sample of data.)

e. Each member of the sample, i.e., the questionnaire data describing each completed programming project, is referred to as a data point. To qualify as a data point, the data must be for a programming effort that

TABLE I

COST FACTOR CLASSIFICATION SCHEME

Logical Grouping	Category Name	Category Definition
THE JOB TO BE DONE	1. Operational Requirements and Design	Includes cost factors associated with the operational characteristics of the system for which the program is being written.
	2. Program Design and Production	Includes cost factors associated with the design, coding, and testing of both support and operational programs.
THE RESOURCES THAT ARE AVAILABLE	3. Data Processing Equipment	Includes cost factors associated with the hardware required to produce and test a program, including all input, output, and peripheral equipment.
	4. Programming Personnel	Includes cost factors reflecting characteristics of the personnel needed to completely develop a program.
THE NATURE OF THE WORKING ENVIRONMENT	5. Management Procedures	Includes cost factors associated with the plans, policies, practices, and review techniques used in the administration of all phases of program development.
	6. Development Environment	Includes cost factors resulting from relationships with external organizations, including customers and other contractors.

resulted in (1) the smallest number of instructions developed for a user with specific computer programming requirements, and (2) a program or product capable of operating in the computer as a single entity or package.

The first and second cycles of analysis that have been completed and the third cycle currently under way have used this model as a basis for the research. The new equations in the next Section as forerunners of the third cycle represent the beginning of an effort to use subsamples with distinct characteristics as a basis for deriving equations. This new approach modifies the assumption of commonality stated earlier.

2. The First Cycle--An Analysis of 27 Data Points

In the first cycle, we used a questionnaire made up of 93 cost factors and 15 cost measures to collect data for 27 computer programming efforts completed at SDC. These data were subjected to various statistical procedures as well as intuitive analysis based upon experience to derive initial estimating equations.

The primary statistical procedure used to derive the estimating equations was multivariate regression.[1] The use of this tool requires that the number of presumed cost factors (or independent variables) to be weighted be considerably less than the sample size, i.e., number of actual data points. In the first cycle this was not true initially, and so presented a major problem. Therefore, other statistical techniques, as well as intuitive judgment, had to be used to reduce the number of variables to a point at which this requirement for performing a regression analysis was met. For example, the following sequence of analyses of the data were used as a basis for reducing the number of variables:

a. Examination of Raw Data. After tabulating the responses to the questionnaire and examining the resulting data matrix, we rejected 10 of the original 93 cost factors for one or more of the following reasons: (1) poor distribution characteristics (e.g., the values lacked variation; (2) identity with other factors (e.g., the data values for one variable were almost equal to those for another variable; (3) apparently erratic or missing responses; and (4) lack of intuitive appeal based upon judgment and experience.

b. Correlation Analysis. The values for the remaining 83 independent and 15 dependent variables were used to calculate a correlation matrix, which depicted the statistical relationship of every variable with every other variable. Cost factors with low correlations with cost measures were then

---

[1] Multivariate regression involves the use of least-squares procedures to determine the m coefficients or weights (B_i) and the constant A_k in an equation $Y_k = A_k + \sum_{i=1}^{m} B_i X_i$, where Y_k would be a cost measure and the X_i would be predictor variables corresponding to selected cost factors. See, for example, Anderson, R. L., and T. A. Bancroft, Statistical Theory in Research, New York, McGraw-Hill, 1952.

rejected unless they had strong intuitive appeal. Also, we used intuition to eliminate cost factors that correlated highly with other more preferable cost factors to develop a set of predictor variables that were as independent as possible.

c. Regression Analysis. The number of factors remaining after using these techniques was still greater than the number of data points. To continue the reduction of factors, we used intuition and experience reflecting our knowledge of program system development. At that point, we had 36 independent variables divided into two groups--one group of 15 factors labeled "most preferred," and another of 21 factors labeled "satisfactory."

A series of regression analyses were conducted, first to reduce further the number of factors by eliminating those with low prediction weights (called beta weights) and then to arrive at the final regression equations. During this sequence of regressions, many factors were eliminated. Also, based upon intuition, we rejected certain outlying data points as well. The rejected points, the data for large-sized programs characterized by very high costs (e.g., 1653 man months and 9026 computer hours), dominated the early regression solutions. Since these large programming efforts were considered unique, they were dropped to form a truncated sample from which we derived solutions for smaller-sized, more common computer programming efforts.

Several equations for estimating the cost measures, particularly number of man months and number of computer hours, were derived in this first cycle. In the first part of this analysis, estimated number of machine language instructions appeared as a dominant predictor variable. But this factor was found to be extremely highly correlated with the actual number of machine language instructions. At this point, we suspected the reliability of the responses to the question concerning the number of estimated number of machine language instructions. Thereafter, in the rest of the analysis, the variable, actual number of instructions, was used instead of its estimated counterpart. Although this substitution strengthened the equations in terms of statistical accuracy, the resulting equations were considered less useful because the number of instructions, like the basic cost measures, man months and computer hours, is ordinarily very difficult to estimate before a programming effort begins.

The first cycle appeared to demonstrate the feasibility of the basic approach, and the results and the results were encouraging enough to pursue the work further. The plan was to gather more data for analysis with expectation of the following:

- Increased precision of estimation over a broad range of programming efforts.

- Increased confidence in the selection of the most appropriate cost factors.

- Sufficient data to investigate the use of subsamples as a basis for deriving different sets of equations having even greater estimating precision than that obtainable from the sample using the entire range of programming efforts.

3. The Second Cycle--An Analysis of 74 Data Points

We began a second cycle that eventually led to the results reported in both this and the following Section. Our chief objectives in the second cycle were to gather more data for similar analyses and to obtain better results, i.e., estimating equations (a) with increased precision, and (b) with cost factors used as predictors that are relatively easy to estimate before a programming job begins. The methods used in the second cycle were similar to those used in the first, but there were some notable differences such as the following:

- We used a revised questionnaire to collect data.
- The sample that resulted from the new collection effort included a much larger percentage of small or low-cost programs.
- To offset the disproportionate effect of the few large-sized, high-cost data points on the analytical results, we used logarithmic transformations for all the cost measures and some cost factors.
- In addition to the regression analysis using single cost factors as independent variables, we grouped and weighted some cost factors to form indices. These, in turn, were treated as independent variables and subjected to regression analysis to derive estimating equations.
- We began to use the data base in other ways that could contribute to the search for meaningful subsamples and also aid management decision making. Specifically, we compared ratios such as the production rate (instructions per man month) as a partial evaluation of the pros and cons of machine-oriented and procedure-oriented languages.

The details of these differences in the second cycle are discussed below:

a. The Data Collection Questionnaire. For the second cycle, we revised the initial questionnaire as a result of feedback and the experience gained in the first cycle. Examples of the changes we made are as follows:

(1) Deleted questions that consistently yield unreliable answers.

(2) Amplified some questions to gather more detailed information. For example, the question concerning Number of Instructions Discarded was separated to ask why they were rejected--to correct programming errors or to introduce operational changes.

(3) Supplemented some questions that were subject to misinterpretation with definitions of the ambiguous terms. For example, the five levels of system complexity were briefly described.

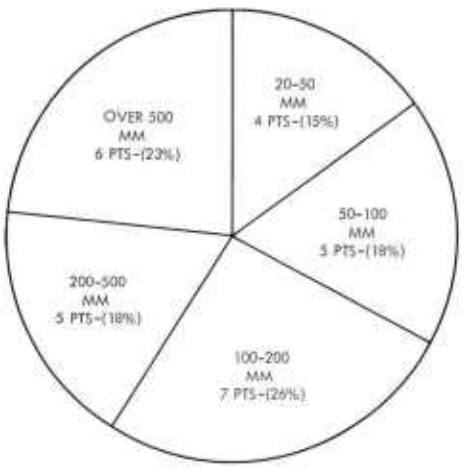
(4) Eliminated questions that were statistically redundant to other superior questions.

Appendix I lists the 96 cost factors and 8 cost measures contained in the revised questionnaire.

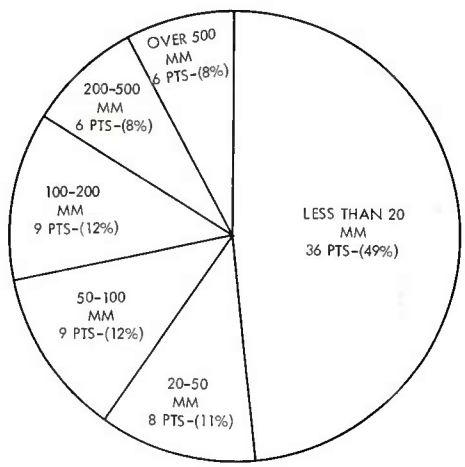
To amend the old data, we prepared a supplementary questionnaire and submitted it to the original respondents for the 27 data points used in the first cycle, so that these data points could be combined with the new ones. Three of the 27 data points were dropped because the available records needed to complete the supplementary questionnaire were not reliable.

b. The Sample. As in the first cycle, no deliberate sample design was used; however, in this second cycle even more managers throughout SDC were asked to complete questionnaires for representative programming activities under their jurisdiction.

After checking the collected data for accuracy and rejecting several incomplete questionnaires, we had a total of 74 data points, including the 24 points from the first cycle. These data represented a variety of programming applications--command and control, compilers, information retrieval, management information, and utility programs.



DISTRIBUTION OF MAN MONTHS (MM)
FIRST CYCLE - 27 DATA POINTS



DISTRIBUTION OF MAN MONTHS (MM)
SECOND CYCLE - 74 DATA POINTS

Figure 1.

Proportion of Man Months in the Sample
for the First and Second Cycles

The pie charts in Figure 1 show the distribution of the data in terms of the cost measure, man months, for the first and second cycles. The charts illustrate in terms of the measure, man months, the difference between the two data bases, especially the increased number of small or low-cost programming efforts in the larger data base for the second cycle. The increased number of smaller jobs and the increased range are also found in the other cost measures--number of computer hours, new machine language instructions, and months elapsed. These characteristics were also found in the cost factors.

c. Use of the Logarithmic Transformation. The frequency distributions of both the cost measures and many cost factors showed clusters at the low and medium values and sparse occurrence of values at the high end. Figure 2, a histogram for the distribution of man months, shows this characteristic, represented by an exponential frequency distribution.

Such distributions cannot be analyzed effectively with multivariate regression techniques since the very high values contribute a disproportionate effect to the derived equations. These solutions tend to be virtually meaningless for low values of the cost measures. To alleviate the effect of the large values and still retain the entire sample, we transformed the cost measures and some of the cost factors with these extreme ranges and low-value clusters by taking the logarithm to the base ten for each value. This transformation compresses the range by drawing in the large values toward the origin. For example, the log transformation applied to the values of the cost measure, man months, as shown in Figure 2, results in the distribution shown in Figure 3. Our rationale in using the log transformation was to keep all the points in the data base and to try to derive results that could be applied across the range of types of programming efforts.

Another way to handle this type of distribution is to drop some of the extreme values if it can be reasonably assumed that these are not true members of the population being analyzed. We used this approach in the first cycle, and we have used it again in the second cycle after an analysis on the entire population was done, i.e., the work reported in TM-2712. The rejection of outliers with large cost values provided the reduced data base used to derive the results in the following Section.

In addition to creating new variables by using the logarithm, we formed some ratios from certain pairs of basic variables. Although several such ratios were formed in the second cycle, we have only used three to date--production rate (instructions per man month), computer usage rate (computer hours per thousand instructions), and documentation rate (number of pages per thousand instructions).

The formation of these ratios and the logarithmic transformations increased the overall number of variables to be considered in the analysis, i.e., an extended set of cost factors and cost measures was created in the second cycle.

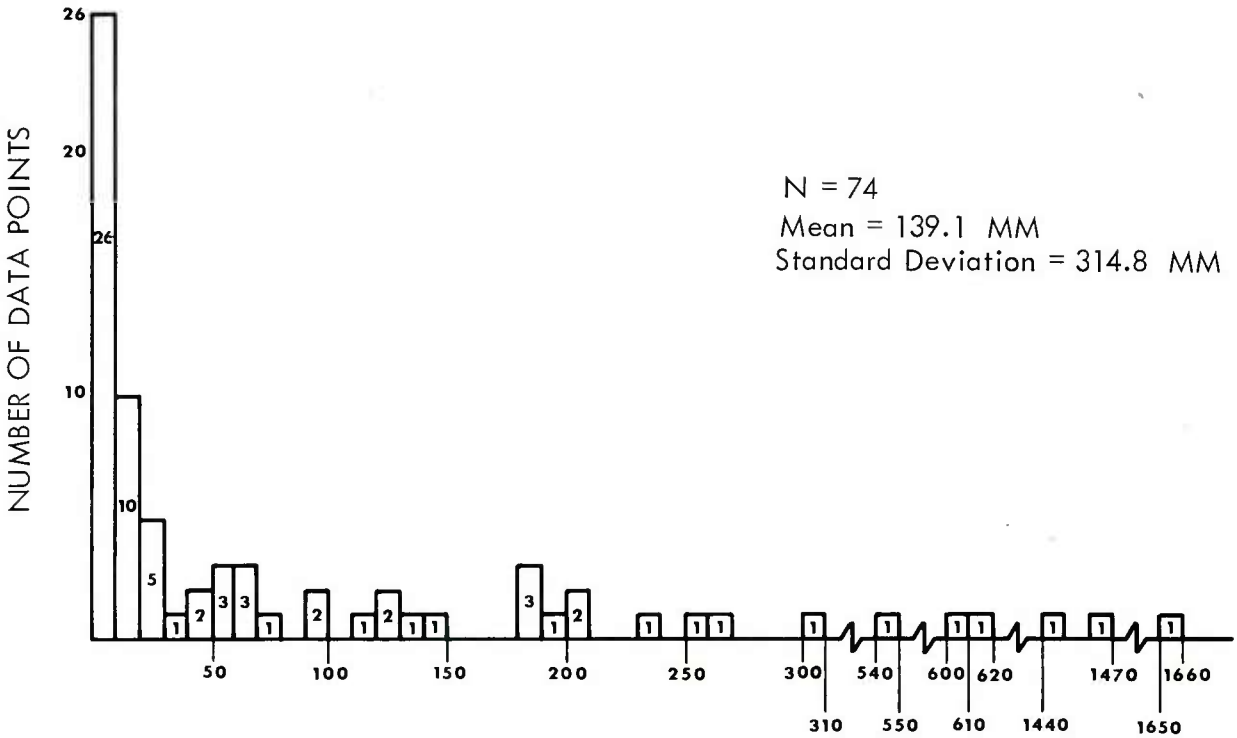


Figure 2.

Frequency Distribution of Total Man Months
for Program Design, Code, and Test

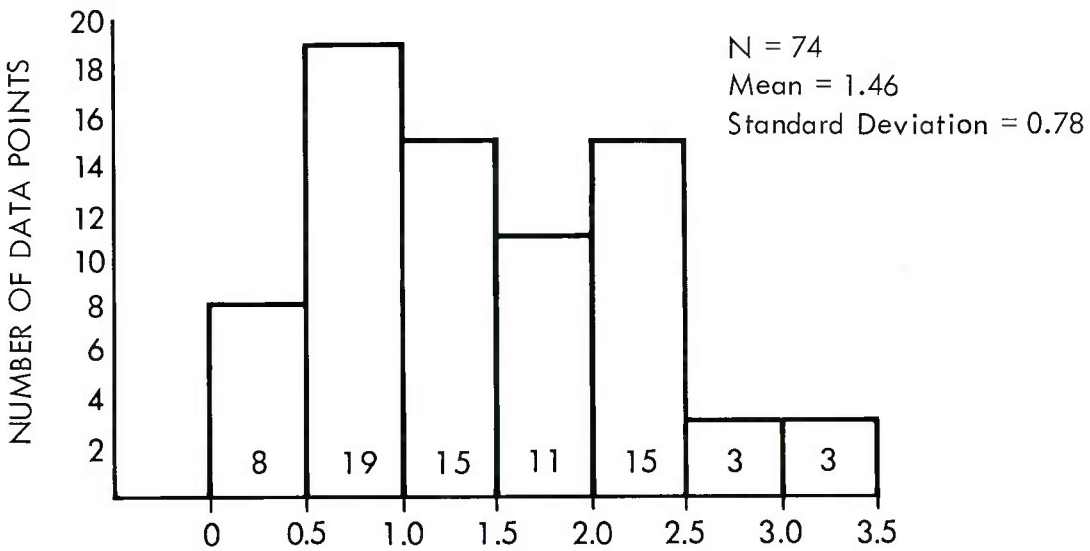


Figure 3.

Frequency Distribution of Log_{10} Total Man Months
for Program Design, Code, and Test

d. The Analysis in the Second Cycle. Following the pattern set in the first cycle, we began the application of the various intuitive and statistical techniques aimed at reducing the number of cost factors (or independent variables)--first to permit application of regression analysis and then to select a best set of cost factors as potential predictors to be used as inputs in the final regression. Our definition of best in this context involved trade-offs between practical and statistical criteria.

In addition to collecting, correcting, and screening the questionnaire data, we examined 400 bivariate scatter plots for both cost measures and cost factors for trends and anomalies in the data. Figures 4, 5, and 6 are examples of these bivariate relationships for some of the cost measures in logarithmic form. These plots on log scales show the boundaries for a 67 percent confidence interval in estimating one cost measure from knowledge of the other. Scatter plots such as these can be useful to a manager for checking costs estimated by other techniques. For example, if the number of new machine language instructions is known with some confidence, the number of man months and computer hours can be estimated by using these charts.

After the scatter plot analysis, we used correlation analysis again to help select a group of preferred independent variables (cost factors) to be used as inputs to the regression analysis. The selection was made from a matrix of the correlation coefficients between all the variables--both independent (cost factors) and dependent (cost measures). The value of the correlation coefficient may range from -1 to +1 and shows the interrelationship between each pair of variables used to calculate it. Values close to ± 1 indicate very high statistical association between the variables involved. In analyzing a correlation matrix first, we looked for high correlation coefficients between independent (cost factors) and dependent (cost measures) variables. What is considered high depends upon the sample size. With the sample size of 74 in the second cycle, a useful correlation coefficient between variables was considered to be one with a numerical value greater than $\pm .20$[2]. From among these independent variables, we eliminated the less desirable member from each pair that had a high correlation. A high correlation for such a pair of independent variables meant that one of the cost factors was statistically redundant, i.e., as a pair, they do not help very much in accounting for the variation in the cost measure. The use of such redundant variables in the equations weakens their predictive potential. In these ways, we searched for cost factors that can be used as predictors and that appear to influence costs but at the same time are as independent of one another as possible, i.e., they supply maximum efficiency for statistical estimation. We also used the regression analysis algorithm as a

---

[2]For a sample size of 74, a correlation of $\pm .12$ could occur by chance two-thirds of the time without any relationship between the two variables. This numerical value is approximately equal to the standard deviation of the null correlation coefficient for the sample size of 74

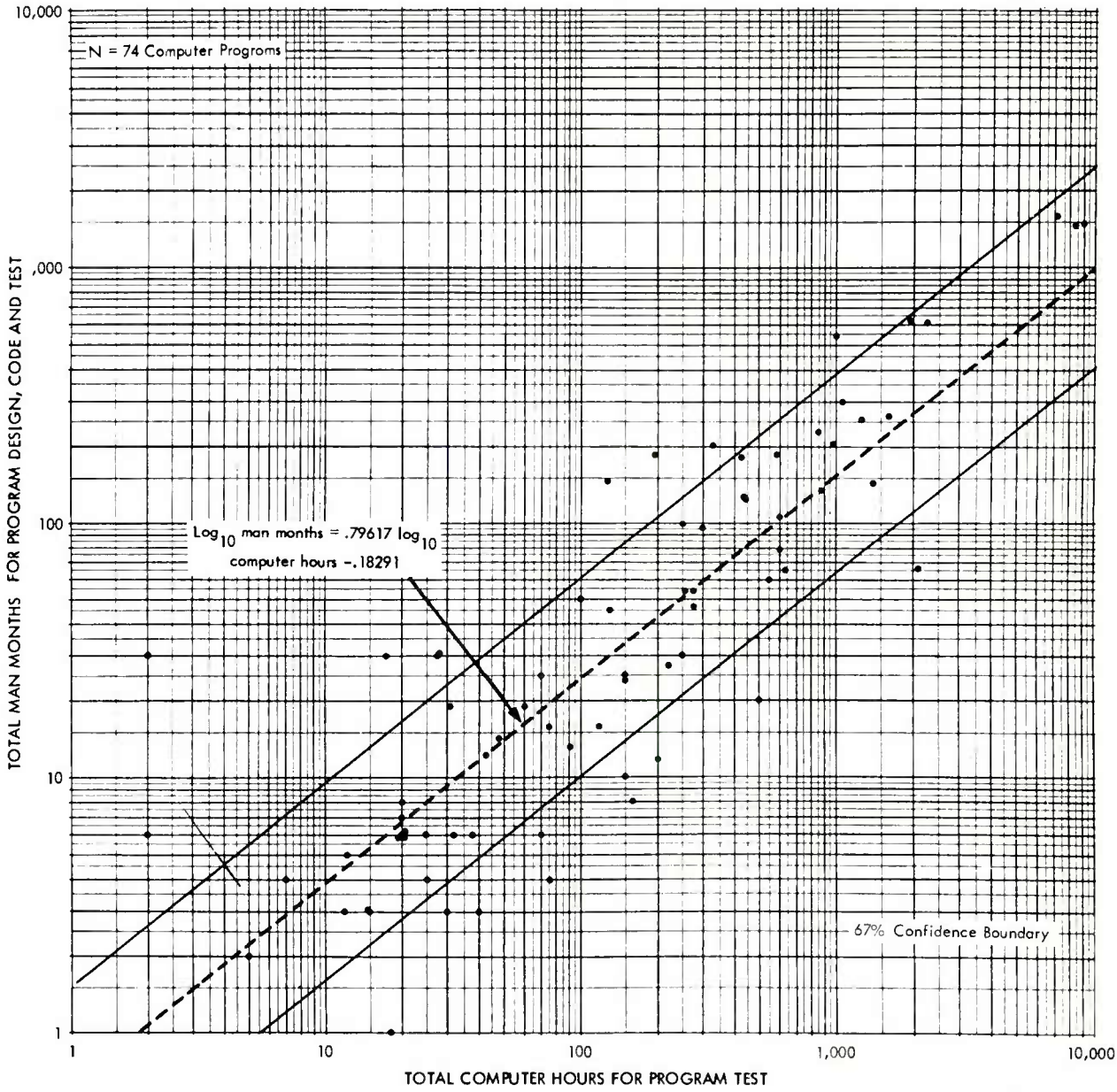


Figure 4.

Relationship Between Total Computer Hours and Total Man Months

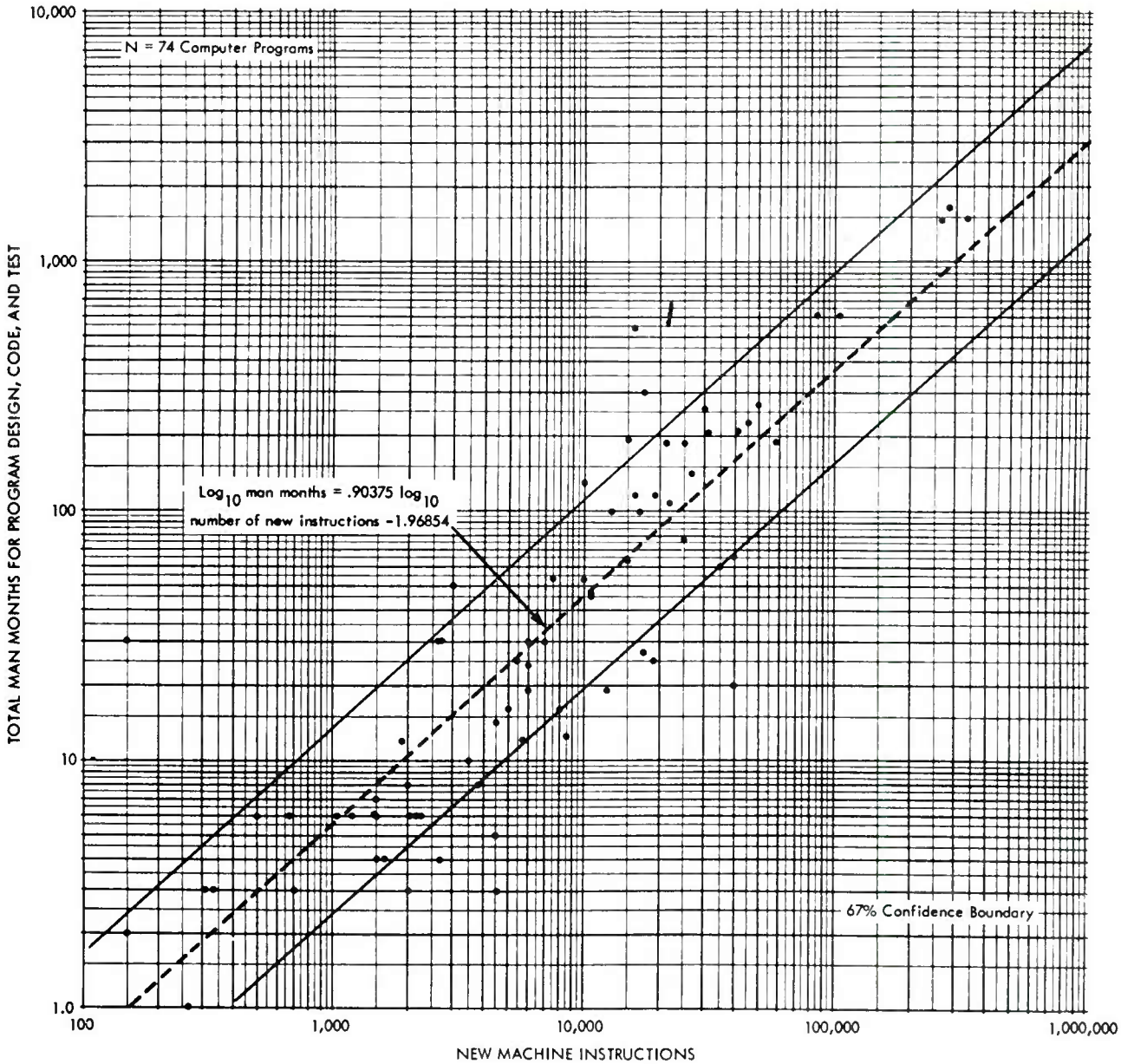


Figure 5.

Relationship Between New Instructions and Total Man Months

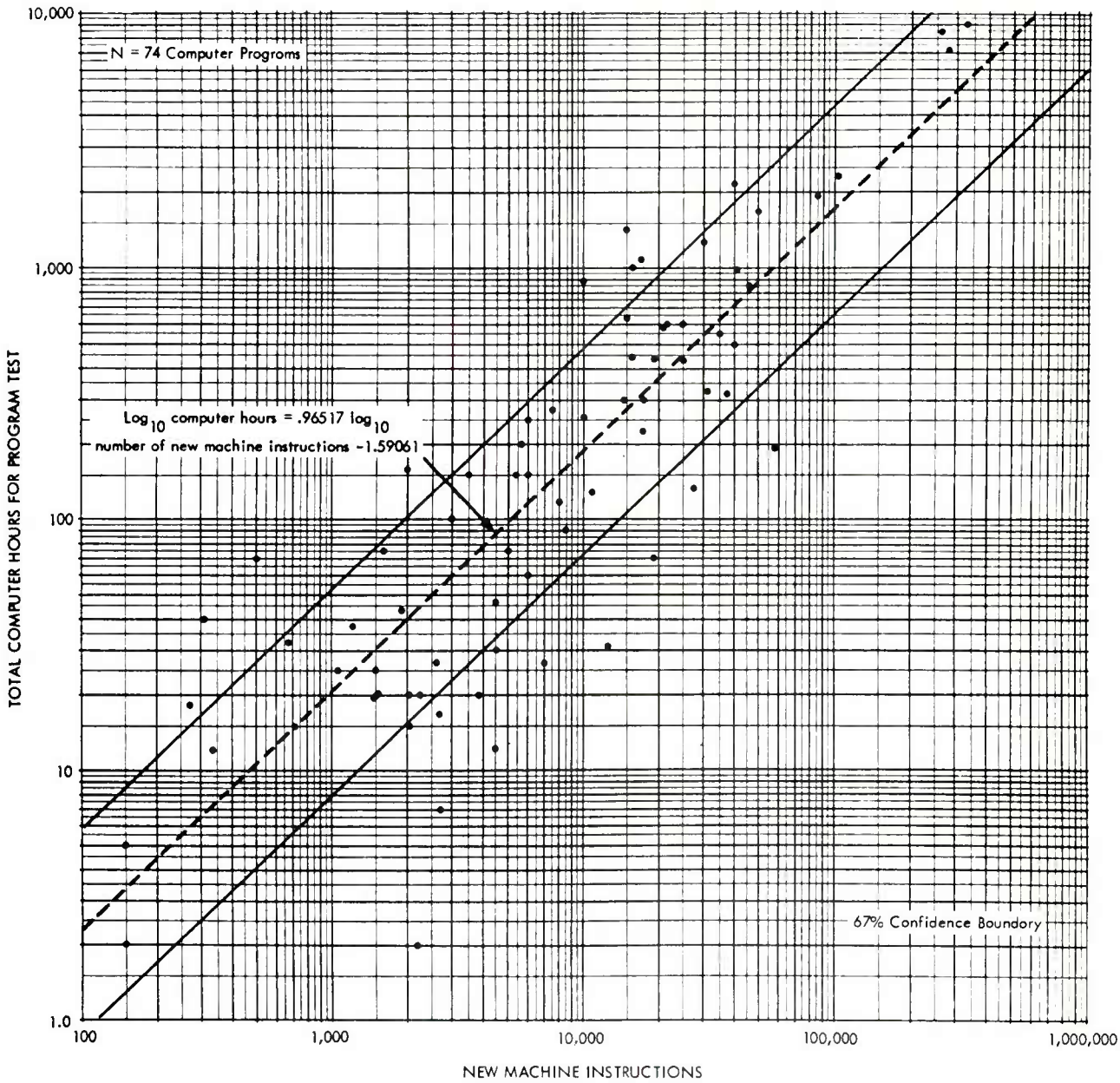


Figure 6.

Relationship Between New Machine Instructions
and Total Computer Hours

technique for screening variables to select those with the largest predictive power. For each cost factor (or independent variable), this analysis yields a measure (called a beta weight) of the contribution of the variable to an explanation of the variance or spread in any particular cost measures. Variables with a high contribution (beta weight) are considered good or efficient from a statistical viewpoint.

The process of reducing the number of cost factors to a final set is known as winnowing and consisted of iterative phases of statistical procedures such as correlation and regression analysis coupled with intuitive selection of variables. Table II shows the list of cost factors that survived the winnowing in the second cycle. These factors, then, were the inputs to the final regression analysis that yielded the estimating equations in Table III.

These equations[3] provide estimating relationships for the transformed cost measures, i.e., logarithm to the base ten of the number of man months, computer hours, new machine language instructions and months elapsed. Some of the predictor variables are also log transformations of the original cost factors, e.g., number of subprograms. To use the equations, one must first look in a table of logarithms for the values of such predictor variables. Once the calculation is made for one of the dependent cost variables (Z), its value must then be transformed back to familiar units such as man months by taking the antilog of the calculated estimate for the particular cost measure. Appendix II contains the definition and coding for each of the variables that occur in these equations.

In the second cycle, we also tried a new approach for forming equations to estimate the costs for computer program production. We felt there would be some value in grouping some certain predictor variables together so that more of these could be used in the estimating equations. Appropriate groups such as this might define a small set of major characteristics that would be common to all computer programming efforts.

Each of these major groups could then include several "second-order" cost factors, such as those already identified in this research (see Appendix I). Values for these constituent cost factors would characterize a specific computer programming job. The major groups could then be used in turn as the independent variables in estimating equations similar to those in Table III. We believed that these groups might provide a more common basis for comparing programming efforts than the more detailed cost factors.

So in this second cycle, four such groups of cost factors, called task indices, were formed. The indices, each consisting of several basic cost factors, characterized the computer programming job in terms of Uniqueness, Job Difficulty, Development Environment, and Job Type. The cost factors contained in the indices were selected on the basis of both intuitive

---

[3]See Appendix VIII of TM-2712.

TABLE II

REDUCED SET OF COST FACTORS

USED AS INPUTS FOR FINAL REGRESSION ANALYSIS

Believed to Increase Costs	Believed to Decrease Costs
Innovation in system	Percent clerical instructions
Complexity of overall system	Percent self-checking-fix instructions
Log_{10} number of subprograms	Percent information storage and retrieval
Log_{10} number of words in data base	Estimated customer experience
Log_{10} number of classes of items in data base	Time-sharing
Log_{10} number of words in tables and constants not in data base	Management index--the ratio of "yes" answers to the total set of questions on management (see Appendix I)
Log_{10} number of input message types	Percent programmers participating in design
Log_{10} number of output message types	Log_{10} production rate--instructions/man month
Complexity of program design	Percent senior programmers
Percent math instructions	
Percent logical control instructions	**Factors With Neither Hypothesis**
Percent generation to produce desired output	Percent I/O instructions
Insufficient memory capacity	Open/closed shop
Insufficient I/O capacity	
Stringent timing requirements	
First programming effort on computer	
Log_{10} average turnaround time with the computer	
Computer operated by agency other than developer	
Program developed away from operational location	
Computer at operational site different than at development site	
Program developed at more than one location	
Log_{10} number of reused instructions*	
Percent error rate--100 x "scrap" instructions/ total instructions coded*	
Percent operational discards--100 x "scrap" instructions due to changes/total instruction coded*	

*Measured in number of machine language instructions

TABLE III

ESTIMATING EQUATION FOR COSTS OF COMPUTER PROGRAM PRODUCTION

(PROGRAM DESIGN, CODE AND TEST)

SAMPLE SIZE, N = 74 DATA POINTS

$$Z_1 = .26X_3 + .28X_5 + .25X_6 + .41X_{15} - .36X_{10} - .46X_{11} - .53X_{12} + 1.06X_{14} + 1.99$$

$$Z_2 = .42X_3 + .16X_4 + .49X_5 + .57X_{15} - .55X_{11} + 1.24X_{14} + 1.35$$

$$Z_3 = .38X_3 + .35X_5 - .25X_6 + .42X_{15} - .17X_{10} + 1.17X_{14} + .28X_{16} + 3.45$$

$$Z_4 = .17X_4 + .20X_5 + .30X_{15} + .11X_{17} + .04X_{18} - .04X_{19} + .33$$

Variables - (Appendix II defines these variables and their coding)

Z_1 - Log_{10} Total Man Months

Z_2 - Log_{10} Computer Hours

Z_3 - Log_{10} New Machine Language Instructions

Z_4 - Log_{10} Months Elapsed

X_3 - Innovation

X_4 - Stringent Timing

X_5 - First Programming Effort on Computer

X_6 - Developed at More than One Location

X_{15} - Log_{10} Number of Subprograms

X_{10} - Estimate of Customer Experience

X_{11} - % Programmers in Design

X_{12} - % Clerical Instructions

X_{14} - % Output Generation Functions

X_{16} - Development Computer at Different Location than Operational

X_{17} - Log_{10} Average Turnaround Time

X_{18} - Log_{10} Number Words in Data Base

X_{19} - Log_{10} Number Output Messages

and statistical appeal. These factors include most of those that appeared in the earlier equations shown in Table III. Regression analysis techniques were used to derive the weights for each of the cost factors that contribute to a particular index. Table IV shows the component cost factors and their weights for each of the four indices. These indices were now used, in turn, as independent variables in a subsequent regression analysis to derive the four estimating equations shown in Table V.

The estimating precision of these equations can be illustrated in a scatter plot of estimated versus actual values by introducing some boundaries to indicate the statistical confidence, i.e., range of expected error for the predictions.

An example of the several such plots that are included in TM-2712 is shown in Figure 7 for the cost measure, computer hours. To make this graph, the equation for computer hours shown in Table V was used to calculate estimates for each of the 74 data points. As indicated earlier, this means that a value for each of the four indices in Table IV is computed for each data point first. Then these are introduced as values for the variables in the equation for computer hours, Z_2, shown in Table V. The resulting estimates are paired with the corresponding actual value of computer hours for each data point to plot a point on the graph. A 45° line, passing through the points [10,10]; [100,100], and so on, is the locus of points at which estimates equal actuals. On the scatter plot, this line would fall in the center of the darkest gray band.

On this plot, Figure 7, the gray bands, called Stanine Bands, show confidence levels as indicated by the inserted table, Stanine Band Number versus Probability Values. These bands can be interpreted as follows: Suppose an estimate of 80 computer hours has been calculated. By reading on the vertical line for this value in Figure 7, we see that the probability (or chance) that the actual value will fall between 58 and 102 computer hours (the values of boundaries on the darkest gray or Number 5 Stanine Band) is found in the table as .20 (or 20 out of 100 programming jobs). Using the same estimate of 80 hours, the probability that the actual value will fall between 32 and 119 (the lower and upper boundaries for the Number 4 and Number 6 Stanine Bands that bracket the Number 5 Stanine Band) is .54, the sum of the probabilities shown in the table for the Numbers 4, 5, and 6 Stanine Bands. The width of these bands depends upon the sample size and the power and efficiency of the predictors used in deriving the equation. Another sample may widen or narrow the Stanine Bands leading to changes in the estimating precision.

Such scatter plots for estimated versus actual costs can also be used to rank or evaluate completed computer programming jobs in terms of the Stanine Band numbers. For example, if, after a job is completed the cost factors are used to calculate an expected cost and the actual cost is higher than the expected, the number of the Stanine Band into which the actual cost falls would indicate how much higher the cost would be in standard statistical terms. The use of

TABLE IV

COMPONENTS OF THE TASK INDICES

Index	Weight	Component Variable	Variable No. (see Tbl III)
I_1 - Uniqueness	1.2	Innovation in System	X_3
	1.0	Stringent Timing Requirements	X_4
	1.7	First Programming Job on Computer	X_5
	1.3	Program Developed at More Than One Location	X_6
I_2 - Job Difficulty	5.0	Log_{10} Number of Subprograms*	X_{15}
	1.0	Log_{10} Number of Classes in Data Base*	**
I_3 - Development Environment	-1.0	Estimate Customer Experience	X_{10}
	-3.4	Percent Programmers Participating in Design	X_{11}
I_4 - Job Type	-1.0	Percent Clerical Instructions	X_{12}
	-1.3	Percent Transformation and Reformating Function	**
	4.4	Percent Generation Function	X_{14}

* In the computation for the Job Difficulty Index, the values of each raw component should be increased by one, since any programming task, i.e., data point in its entirety, is considered to represent a difficulty equal to at least one subprogram and one class of data.

** These two variables did not appear in the equations of Table III:

Log_{10} Number of Classes in Data Base--classes are categories or types of items such as names of people, salaries, cities or any information characteristic with many entries.

Percent Transformation and Reformating Function--conversion of data from one form to another coded in decimal.

TABLE V

EQUATIONS FOR ESTIMATING COSTS

OF COMPUTER PROGRAM PRODUCTION WITH INDICES*

SAMPLE SIZE N = 74 DATA POINTS

$$Z_1 = 0.23I_1 + 0.08I_2 + 0.21I_3 + 0.36I_4 + 1.56$$

$$Z_2 = 0.28I_1 + 0.12I_2 + 0.19I_3 + 0.25I_4 + 1.69$$

$$Z_3 = 0.29I_1 + 0.08I_2 + 0.13I_3 + 0.32I_4 + 3.31$$

$$Z_4 = 0.11I_1 + 0.06I_2 + 0.05I_3 + 0.12I_4 + 0.60$$

Z_1 - Log_{10} Total Man Months

Z_2 - Log_{10} Total Computer Hours

Z_3 - Log_{10} New Machine Language Instructions

Z_4 - Log_{10} Months Elapsed

I_1 - Uniqueness Index

I_2 - Job Difficulty Index

I_3 - Development Environment Index

I_4 - Job Type Index

*Indices are defined in Table IV

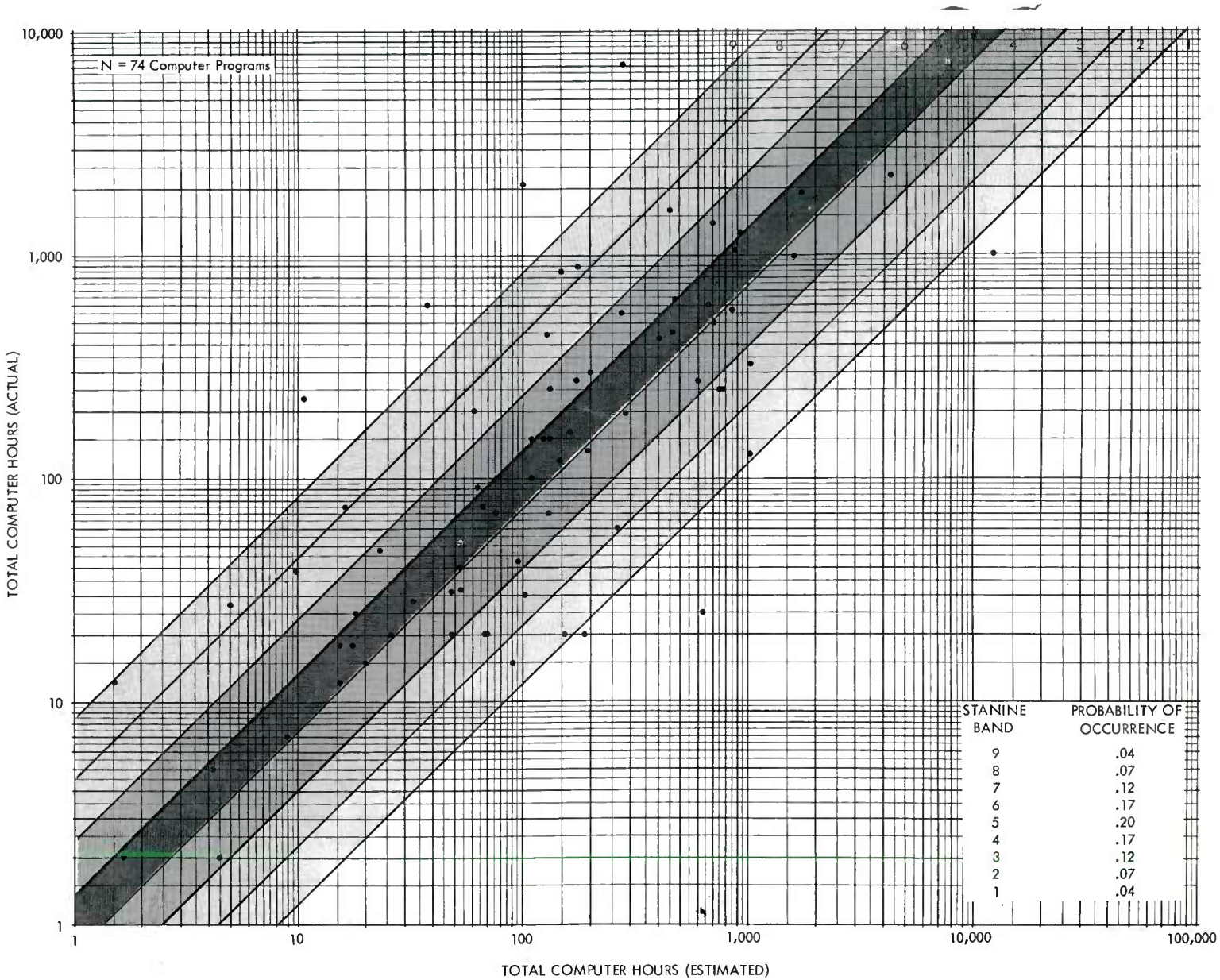


Figure 7.

Estimated Versus Actual for Computer Hours with Stanine Bands

Stanine Band number for ranking jobs can be used to compare completed computer programming jobs with each other, thus providing a basis for estimating differential performance.

e. Comparison of Machine-Oriented and Procedure-Oriented Language. In addition to supplying data for deriving the equations for estimating costs, the same data base can be used to make various comparisons that can help managers make decisions in computer program production. In the second cycle, we took a first step in such uses of the data base. Specifically, we compared machine-oriented language (MOL) to procedure-oriented language (POL) in terms of the three ratios--production rate measured in number of equivalent machine language instructions per man month, computer usage rate measured in number of computer hours per equivalent machine language instructions, and documentation rate measured in number of pages. In the sample of 74 data points, 14 programming efforts used a POL--JOVIAL, in this instance--and the remaining 60 efforts used various MOLs.

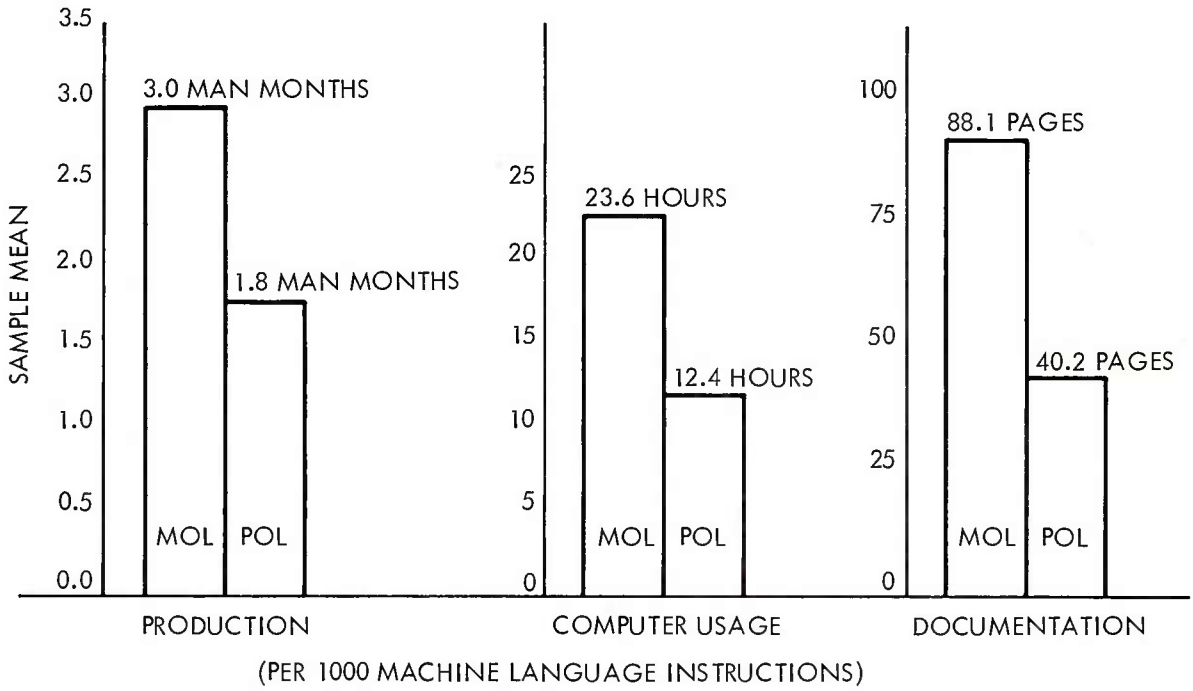


Figure 8.

Comparison of Three Ratios for MOLs and POLs

The bar chart, Figure 8, shows the comparison between these three rates for MOL and POL.[4] The production rates shown in the Figure 8 as man months per 1000 equivalent machine language instructions are the result of inverting the mean rate expressed in terms of instructions per man month. The other two rates, computer usage and documentation, are the actual means as calculated. These comparisons provide some numerical evidence that tends to confirm the opinion that POLs are more economical than MOLs during computer program production.

The values in the figure were not corrected for a known distortion in the production rates. The compilers, i.e., the programs that convert programs written in procedure-oriented languages to machine languages, generally yield more machine language code than a logically equivalent program coded in a machine-oriented language and processed by an assembler. Estimates for this expansion factor with "mature" JOVIAL compilers, those that have been improved based upon feedback over a period of time, are 10 to 15 percent. But, this expansion may be much larger for new compilers.

These, then, are highlights of the work in the second cycle reported in TM-2712. Other analyses were done, e.g., the development of a composite index composed of cost measures as a measure of overall costliness and an examination of the sensitivity of the cost estimating relationships shown in Table V to changes in the indices.

After the publication of TM-2712, more work was done in the second cycle with the 74 SDC data points to investigate techniques for improving the results, particularly by increasing the estimating precision of the equations. This extended analysis of the second cycle has overlapped the initial analysis in the third cycle to some extent. The next section contains some of the latest results in this extended analysis as well as a brief description of the work in the third cycle.

---

[4] Details of this computation are given in Section XX of TM-2712.

SECTION III

CURRENT WORK

This Section describes the equations and the work to derive them that extend the results presented in TM-2712, reviewed in the preceding section. These equations represent improvements in the estimating precision as well as investigations into approaches that are planned for the work in the third cycle. To conclude the Section, the work completed to date in the third cycle is reported as well as the plans for the analysis.

1. Extensions of the Second Cycle

The extended work in the second cycle resulted in two sets of equations. We derived the first set from a truncated sample, i.e., a sample of reduced size that results from dropping data points with large values for the cost measures. The second set of equations actually consists of three subsets of four equations (one for each of the cost measures). Each subset corresponds to and was derived from a subsample with a restricted range for the cost measure, man months, i.e., 1 to 9, 10 to 79, and 80 to 260 man months. Together, these equations represent the best results obtained to date; better results are expected from the third cycle in which the subsampling approach will be investigated more extensively. The chief criterion for goodness used in this current work is improvement in estimating precision or, in statistical terms, reduction of the standard error of estimate. As we have indicated, such improvements were achieved by a combination of truncation and division of the remaining sample into three subsamples.

Truncation of the sample by deleting data points with very large costs that appeared to be unique was tried in the first cycle. Using the same approach, we dropped the seven largest data points, ranging from 260 to 1653 man months (see Figure 2), and used the remaining sample of 67 data points to derive equations for the four cost measures--number of man months, computer hours, new machine language instructions, and months elapsed. This reduction in the upper bound for the range of sample from 1653 down to 260 man months greatly reduced the standard error of estimate for each cost measure, so that, from a statistical viewpoint, the estimating precision of the resulting equations was increased.

In this analysis, we did not search anew for the set of best cost factors as predictor variables to be used in the regression algorithm, but we accepted the eleven factors used to compute the indices (see Table IV). As inputs to the regression algorithm, we added two additional variables--one to indicate whether or not a machine-oriented or procedure-oriented language was used in the programming effort--and another to indicate whether or not a large or small computer was used.

By reducing the range for the cost measures, we reduced the need to use the logarithmic transformation and so were able to use the more familiar measures, e.g., man months. The use of these cost measures permitted us to compare differences in estimating precision in terms of the statistic, standard error of estimate, more easily for the various equations derived.

We did not conduct complete analyses to derive the equations from either the truncated sample or the subsamples, because the cost factors considered as predictors were restricted to the set used in the first part of the second cycle (as described earlier). A more thorough analysis may have opened the door to other useful cost factors since those that may be statistically significant for one range of man months, e.g., 1 to 1653, may not be for another range, e.g., 1 to 260. To admit these other factors into the analysis, we would have had to calculate a completely new correlation matrix for all the original cost factors and the cost measures for each truncated sample and subsample and then we would have had to repeat the winnowing process described in Section II.

Also, a more thorough analysis would have considered a different truncation and division of the sample for each cost measure, e.g., a separate division for computer hours, new machine language instructions and months elapsed instead of the truncation and subsequent division based only upon the single cost measure, man months. The more thorough analysis was not done because these current analyses to extend the work in the second cycle were conducted with limited resources.

The equations presented here have fewer cost factors than the thirteen we started with. The number of factors was deliberately reduced to create equations that are powerful and efficient in a statistical sense, and to preserve the factors that are most meaningful. This reduction was achieved in two ways. First, a subset of the basic thirteen predictor variables was formed by removing those that were not intuitively appealing as a basic influence upon a particular cost measure and at the same time did not contribute statistically to the estimating precision of the equation for the cost measure. The remaining variables were each retested as potential predictors using a stepwise multivariate regression algorithm. Variables were finally excluded if they were not intuitively meaningful and did not significantly influence the two key statistics--the standard error of estimate (a basic measure of the estimating precision of an equation) and the coefficient of determination, i.e., the square of the multiple correlation coefficient (a measure of the proportion of the variation in the cost measure that is accounted for by the equation).

Tables VI, VII, VIII, and IX each show four equations, one for each of the cost measures--number of man months, computer hours, new machine language instructions, and months elapsed--for the truncated sample of 67 data points, small jobs (1 to 9 man months), medium jobs (10 to 79 man months), and large jobs (80 to 260 man months) respectively. The cost factors used as predictor variables in these equations are defined in Appendix II. In these tables,

TABLE VI

EQUATIONS FOR ESTIMATING COSTS OF COMPUTER PROGRAM PRODUCTION (PROGRAM DESIGN, CODE AND TEST) FOR JOBS RANGING FROM 1 TO 260 MAN MONTHS BASED ON A SAMPLE SIZE OF N = 67

$$Y_1 = -7.95X_1 + 56X_2 + 18X_4 + 37X_5 + 43X_6 + 0.84X_7 - 46X_{11} + 82X_{14} - 6$$

$$Y_2 = 47X_2 + 100X_4 + 200X_5 + 7.67X_7 + 0.74X_8 - 148X_{11} + 216X_{12} + 542X_{14} - 48$$

$$Y_3 = 8205X_2 + 11014X_5 + 8051X_6 + 66X_7 + 12X_8 - 9194X_{11} + 20876X_{14} + 2840$$

$$Y_4 = 1.91X_1 + 1.82X_4 + 3.00X_5 + 2.60X_6 + 0.13X_7 + 0.01X_8 - 1.59X_{10} - 2.48X_{11} + 8.71X_{14} + 8.45$$

Variables	Range	Std Dev	Mean	Std Error	R^2
Y_1 - Total Man Months	1-260	72	55	54	.50
Y_2 - Total Computer Hours	2-1625	417	278	308	.52
Y_3 - New Machine Language Instructions	150-58,300	13,865	11,912	9,962	.53
Y_4 - Months Elapsed	2-36	7.1	9.6	5.4	.50

X_1 - MOL vs POL

X_2 - Large vs Small Computers

X_3 - Innovation

X_4 - Stringent Timing

X_5 - First Programming Effort on Computer

X_6 - Program Development at More than One Location

X_7 - Number of Subprograms

X_8 - Number of Classes in Data Base

X_9 - Log_{10} Number of Classes in Data Base

X_{10} - Estimate Customer Experience

X_{11} - % Programmers in Design

X_{12} - % Clerical Instructions

X_{13} - % Transformat-Reformat Functions

X_{14} - % Generation Functions

TABLE VII

EQUATIONS FOR ESTIMATING COSTS OF COMPUTER PROGRAM PRODUCTION

(PROGRAM DESIGN, CODE AND TEST) FOR A SMALL JOB

1 TO 9 MAN MONTHS BASED UPON A SAMPLE SIZE OF N = 26

$$Y_1 = -0.92X_1 - 0.47X_2 + 1.45X_3 + 0.78X_4 + 1.13X_6 - 2.07X_{12} + 4.75$$

$$Y_2 = 6.95X_1 - 6.11X_2 + 14.02X_3 + 16.83X_4 + 41.41X_5 - 7.13X_{11} + 21.05$$

$$Y_3 = 1084X_1 - 354X_2 + 415X_3 + 17X_7 + 1219X_9 - 267X_{10} - 2586X_{12} + 1666$$

$$Y_4 = -.88X_4 + 1.86X_5 + 1.78X_6 + .06X_7 - 0.86X_{11} - 2.61X_{13} + 5.35X_{14} + 5.08$$

Variables	Range	Std Dev	Mean	Std Error	R^2
Y_1 - Total Man Months	1-9	2.0	4.7	1.7	.47
Y_2 - Total Computer Hours	2-160	31.9	28.7	28.6	.39
Y_3 - New Machine Language Instructions	150-4580	1256	1710	885	.64
Y_4 - Months Elapsed	2-11	2.0	5.1	1.4	.65

X_1 - MOL vs POL

X_2 - Large vs Small Computers

X_3 - Innovation

X_4 - Stringent Timing

X_5 - First Programming Effort on Computer

X_6 - Program Development at More than One Location

X_7 - Number of Subprograms

X_8 - Number of Classes in Data Base

X_9 - Log_{10} Number of Classes in Data Base

X_{10} - Estimate Customer Experience

X_{11} - % Programmers in Design

X_{12} - % Clerical Instructions

X_{13} - % Transformat - Reformat Functions

X_{14} - % Generation Functions

TABLE VIII

EQUATIONS FOR ESTIMATING COSTS OF COMPUTER PROGRAM PRODUCTION
(PROGRAM DESIGN, CODE AND TEST) FOR A MEDIUM JOB
10 TO 79 MAN MONTHS BASED UPON A SAMPLE SIZE OF N = 25

$$Y_1 = 7.22X_5 + 4.04X_6 + 2.64X_9 - 13.25X_{10} - 9.27X_{11} + 66$$

$$Y_2 = 271X_5 + 5X_7 + .75X_8 - 324X_{11} + 426X_{12} + 95$$

$$Y_3 = 11108X_5 + 13X_8 + 6127X_{10} - 14843X_{11} + 3892X_{12} + 10620X_{13} + 24760X_{14} - 4068$$

$$Y_4 = 1.5X_1 + 4.0X_5 + .16X_7 + .01X_8 - 3.17X_{11} + 5.57$$

Variables	Range	Std Dev	Mean	Std Error	R^2
Y_1 - Total Man Months	10-79	20.6	32.5	21	.18
Y_2 - Total Computer Hours	27-2100	418.8	284.3	233	.76
Y_3 - New Machine Language Instructions	1878-40,000	11132.8	12526.6	8759	.56
Y_4 - Months Elapsed	3-36	7.0	8.9	5.0	.50

X_1 - MOL vs POL

X_2 - Large vs Small Computers

X_3 - Innovation

X_4 - Stringent Timing

X_5 - First Programming Effort on Computer

X_6 - Program Development at More than One Location

X_7 - Number of Subprograms

X_8 - Number of Classes in Data Base

X_9 - Log_{10} Number of Classes in Data Base

X_{10} - Estimate Customer Experience

X_{11} - % Programmers in Design

X_{12} - % Clerical Instructions

X_{13} - % Transformat - Reformat Functions

X_{14} - % Generation Functions

TABLE IX

EQUATIONS FOR ESTIMATING COSTS OF COMPUTER PROGRAM PRODUCTION (PROGRAM DESIGN, CODE AND TEST) FOR A LARGE JOB 80 TO 260 MAN MONTHS BASED UPON A SAMPLE SIZE OF N = 16

$$Y_1 = 47.55X_6 + 1.84X_7 - 28.04X_{10} - 89.12X_{11} + 197.77$$

$$Y_2 = 2.48X_5 + 11.47X_7 - 161.51X_{11} + 429$$

$$Y_3 = 16467X_5 + 4924X_6 - 3124X_{10} - 14586X_{11} - 16583X_{12} + 17944X_{14} + 33919$$

$$Y_4 = 4.8X_1 + .50X_3 + .02X_7 + .01X_8 + 5.08X_{11} + 4.02X_{14} + 12.60$$

Variables	Range	Std Dev	Mean	Std Error	R^2
Y_1 - Total Man Months	80-260	54.2	169.8	50	.38
Y_2 - Total Computer Hours	250-1625	459.5	671.6	418	.38
Y_3 - New Machine Language Instructions	10,000-58,300	14274	27531	13152	.49
Y_4 - Months Elapsed	5-25	5.4	17.8	5.4	.38

X_1 - MOL vs POL

X_2 - Large vs Small Computers

X_3 - Innovation

X_4 - Stringent Timing

X_5 - First Programming Effort On Computer

X_6 - Program Development at More than One Location

X_7 - Number of Subprograms

X_8 - Number of Classes in Data Base

X_9 - Log_{10} Number of Classes in Data Base

X_{10} - Estimate Customer Experience

X_{11} - % Programmers in Design

X_{12} - % Clerical Instructions

X_{13} - % Transformat - Reformat Functions

X_{14} - % Generation Functions

the first three columns to the right of the cost measure characterize the sample for each particular cost measure in terms of statistics, i.e., the range, standard deviation, and mean for each cost measure.

The next two columns show the standard error of estimate which indicates how close one can expect an estimate to be to its actual value. For example, referring to Table VII for small jobs, to obtain the range of computer hours in which we would expect an actual value to occur approximately[5] two-thirds of the time we add and subtract the standard error of estimate for computer hours, 28.7 hours, to and from any estimate. The last column on the right of the cost measures shows the coefficient of the determination R^2, (the) multiple correlation coefficient squared). This statistic has a range from 0 to 1 and indicates the proportion of the variance occurring in the sample values of the cost measure that is accounted for by the equation.

Without the benefit of tests or cross-validation for these equations, we feel that the preferred sets are those in Tables VI and VII representing the truncated sample with 67 data points and the sub-sample of 26 data points for small jobs, respectively. Despite the lack of a thorough analysis, e.g., the calculation of a new correlation matrix and the subsequent winnowing process, the chances that new significant cost factors would have emerged in a thorough analysis of the 67 point sample appear to be small since such a large percentage of the data was used in the derivations. Therefore, we would hazard a guess that these equations are relatively good in a statistical sense.

Figures 9, 10, 11 and 12 provide scatter plots for estimated versus actual values corresponding to the equations in Table VI for each cost measure, i.e., number of man months, computer hours, new machine language instructions and months elapsed. These figures are similar to Figure 7 in Section II and also include Stanine Bands that illustrate the confidence intervals for the equations. As discussed in Section II these Bands can also be used for comparing and evaluating the relative costs of various computer programming efforts.

The equations in Table VI have sufficiently large standard errors of estimate so that if they were used to calculate estimates for low cost jobs, the percentage of expected error for a specific estimated cost measure would be large enough to make them almost useless. So for small jobs the equations in Table VII are apt to yield better estimates.

These two sets of equations could be used in sequence--starting with those derived from the truncated sample in Table VI and then if the calculated estimate for man months is nine or less, using the equations in Table VII

---

[5]The term approximately is used because the standard error of estimate is calculated for the mean value and as estimates deviate from the mean the value of this statistic grows larger.

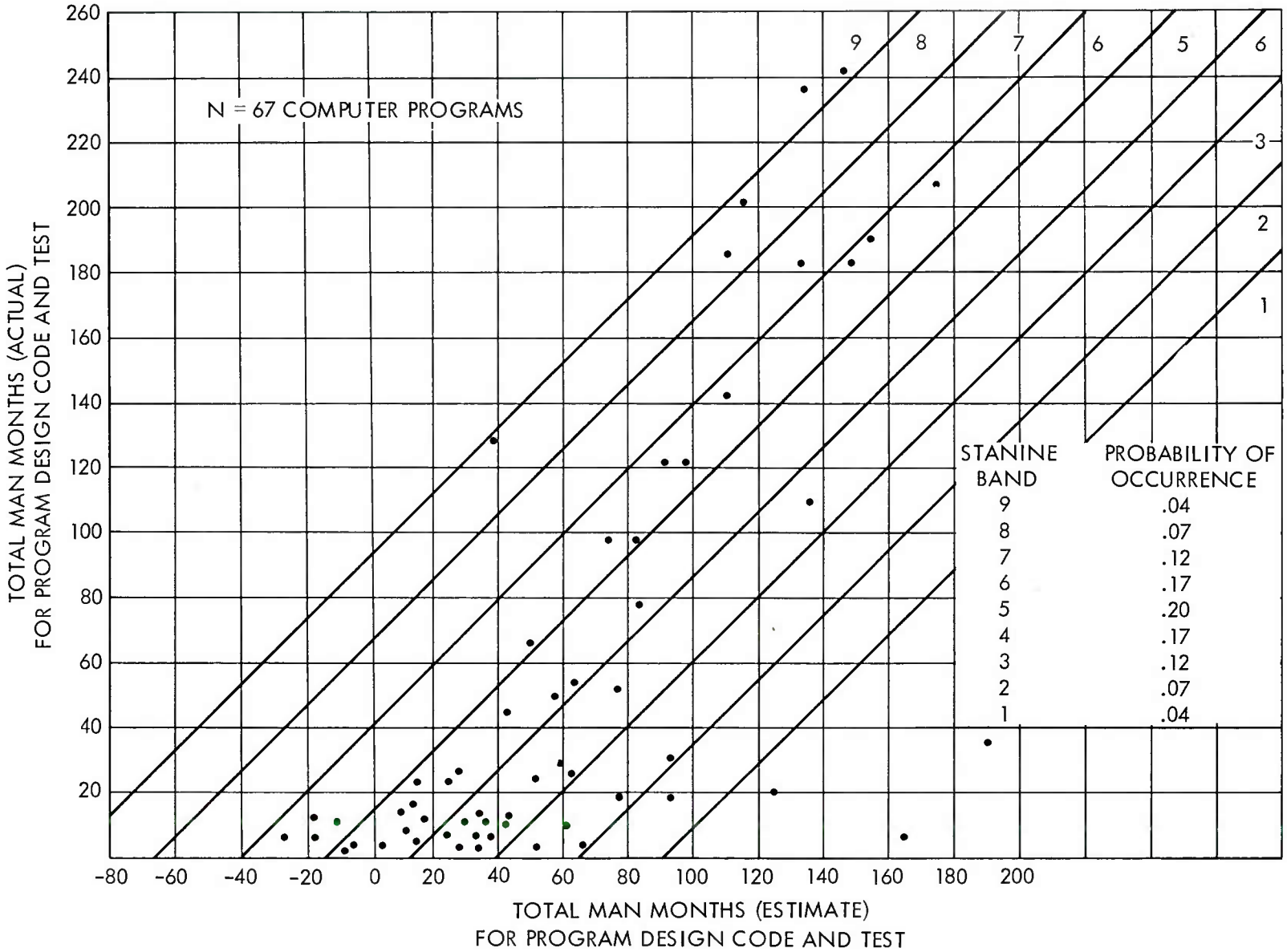


Figure 9.

Estimated Versus Actual for Man Months with Stanine Bands (Truncated Sample)

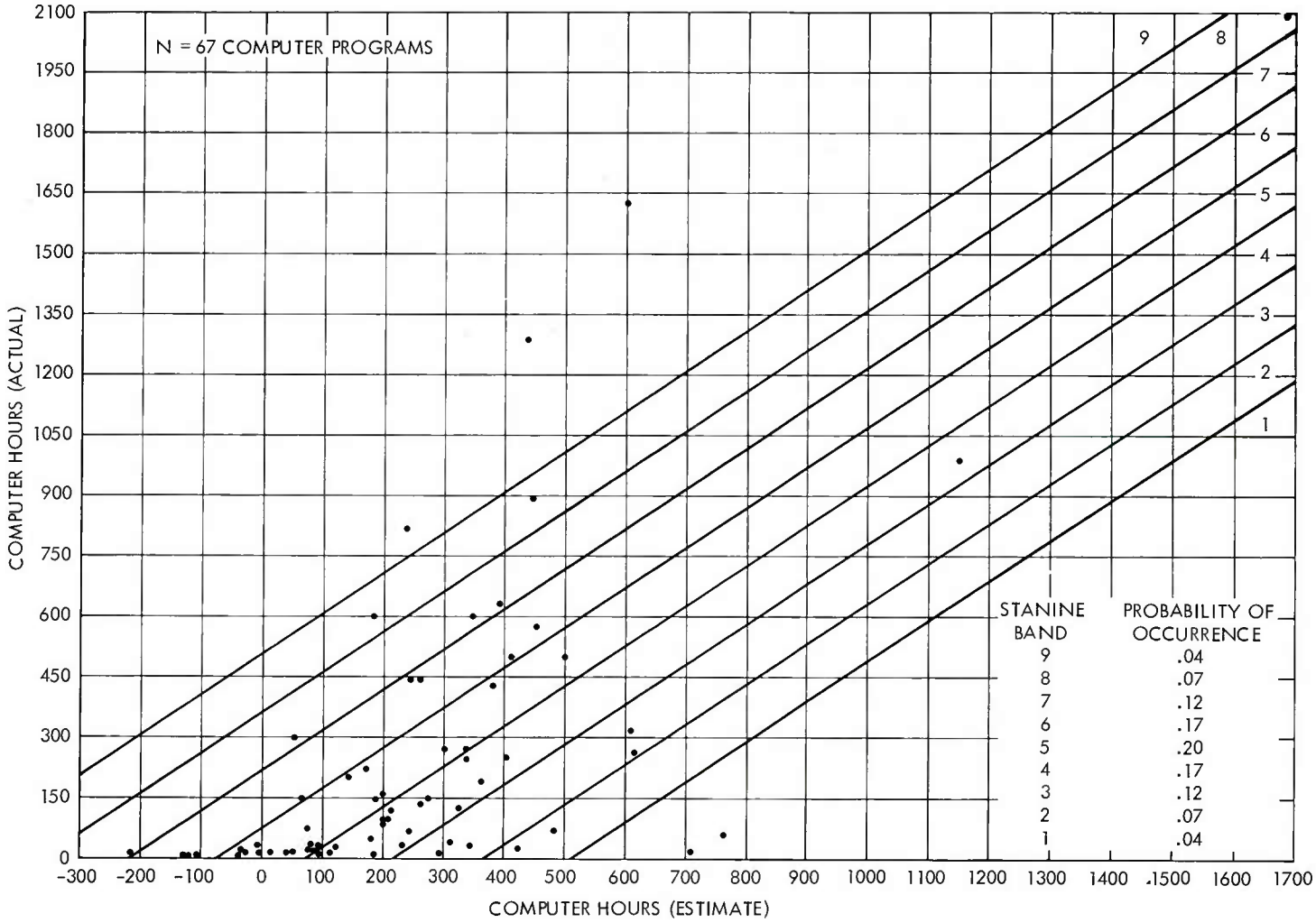


Figure 10.

Estimated Versus Actual for Computer Hours with Stanine Bands (Truncated Sample)

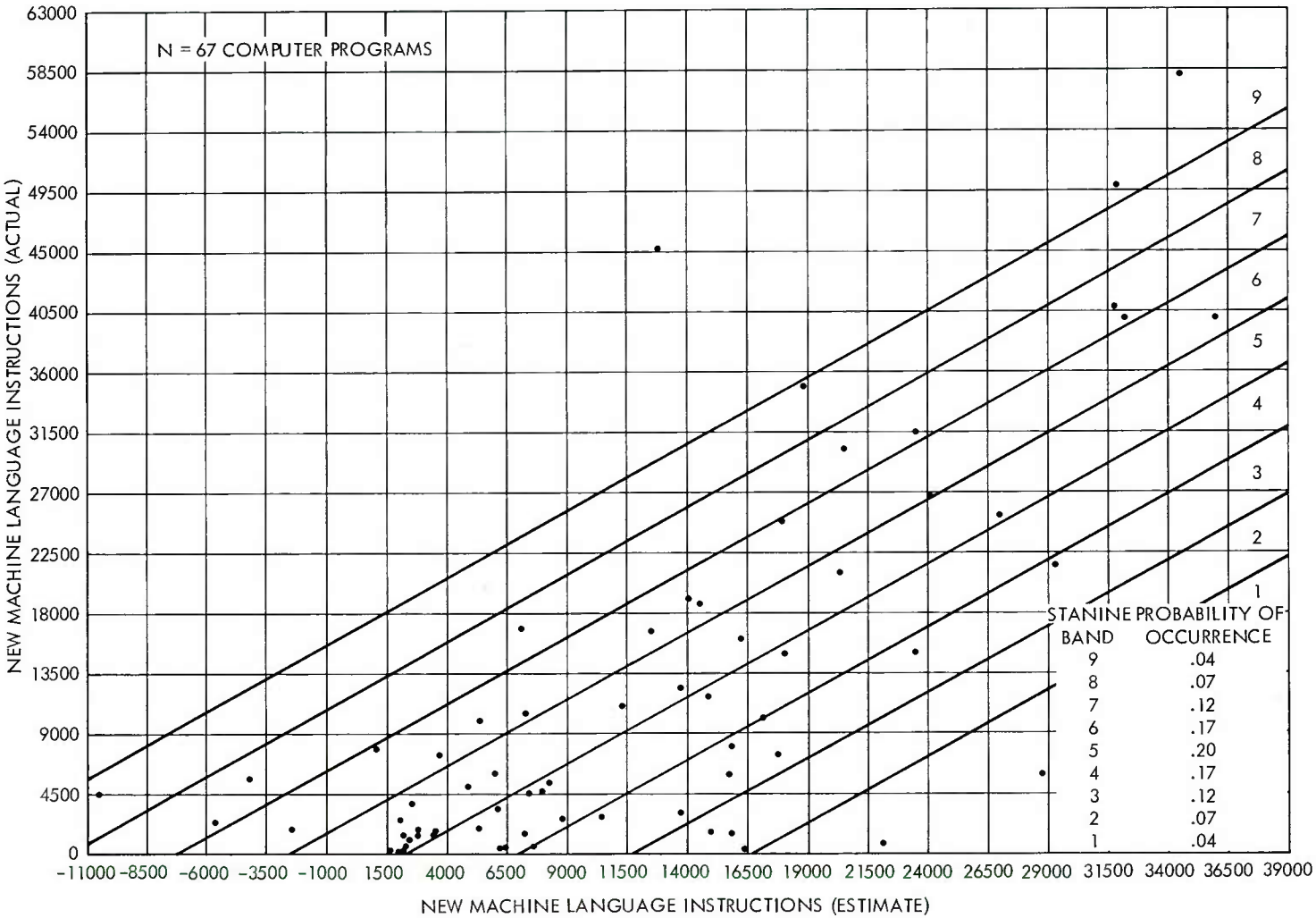


Figure 11.

Estimated Versus Actual for New Machine Language Instructions with Stanine Bands (Truncated Sample)

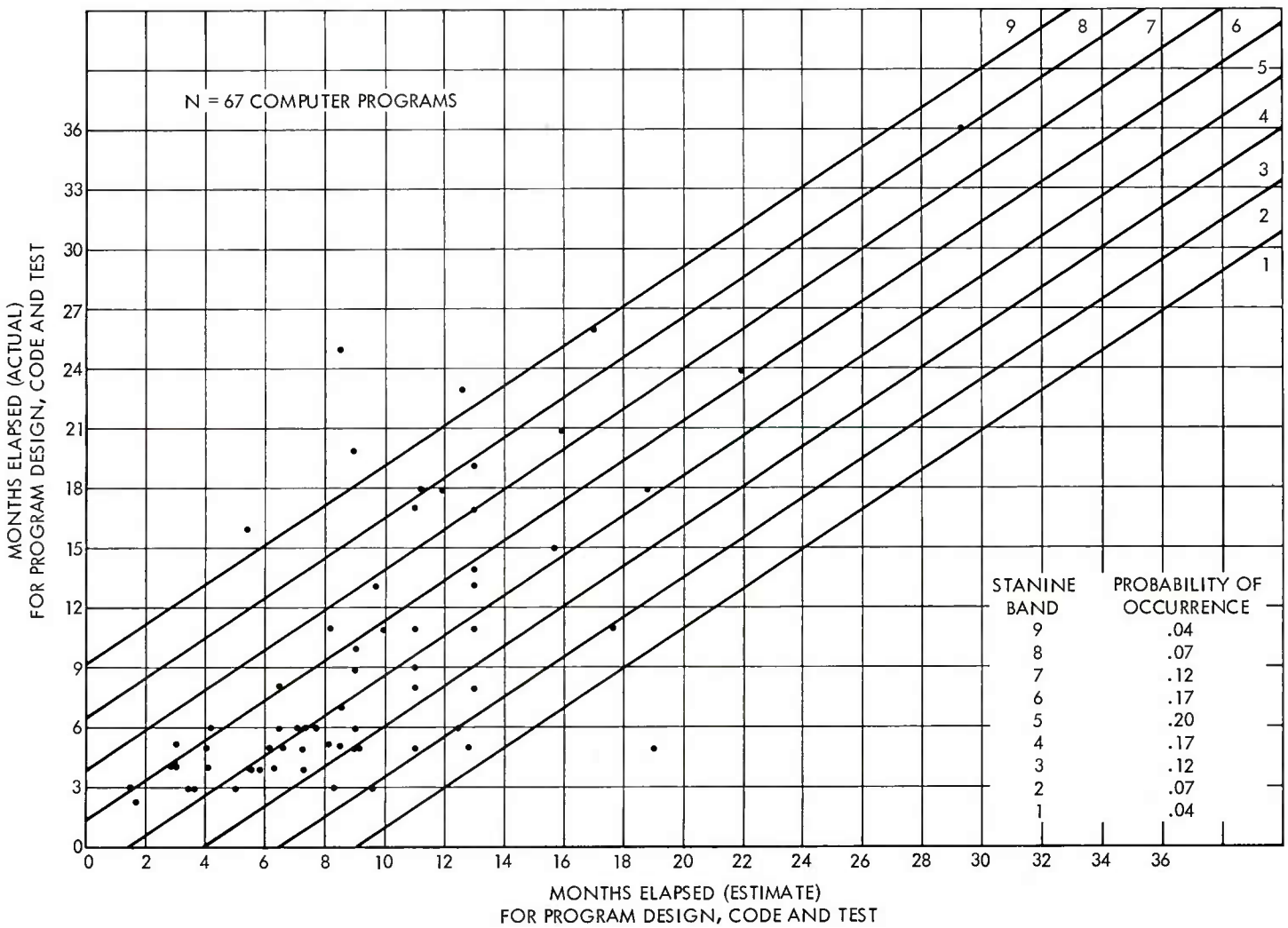


Figure 12.

Estimated Versus Actual for Months Elapsed with Stanine Bands (Truncated Sample)

to estimate for all four cost measures. The equations in Table VII (along with the others in Tables VIII and IX) are statistically less stable than those in Section II (Tables III and V) and those derived from the truncated sample (Table VI). This instability stems partially from the less thorough analytical procedures but mainly from the small sample size used to derive the subsample equations. The largest subsample consisted of the 26 data points used to derive the equations for small jobs in Table VII. Without very strong predictors among the cost factors that can be used as inputs to the regression analysis, small samples can result in spurious coefficients or weights for the predictor variables. In conducting this analysis of subsamples, some signs of such instability were detected. Therefore, we would expect the subsample equations derived from fewer data points, i.e., those for the medium and large jobs, to be even less stable statistically. In the work on the third cycle now under way, the increased size of the total data base should provide a more adequate number of data points for investigations of subsamples as a means to achieve increased estimating precision.

To illustrate the gain made by the efforts to improve estimating precision, Table X compares numerical values for several sets of equations of (1) the range of each cost measure that corresponds to twice the standard error of estimate, and (2) the ratio of standard error of estimate to the mean (called the coefficient of variation). The equations are identified by the number of the Table in which they appear in this report.

2. The Third Cycle

In may of 1965, we made plans for a third cycle of the work to derive cost estimating equations. This planning was prompted by the grant of an Air Force Report Approval Number to gather data from Air Force and industrial programming organizations. The primary purpose of this third cycle was to gether still more data--this time from non-SDC sources--so that we could search for improvements in accuracy by dividing the accumulated data in the total sample into subsamples. The subsample equations discussed earlier represent a first attempt to make such a division.

Since the plans were made, 104 data points have been gathered as a result of submitting a revised questionnaire to 16 Air Force and 10 industrial programming organizations. When these questionnaires were examined carefully in early 1966, we found many questions that were misinterpreted or not answered. Recently, we have been validating these data, i.e., returning the questionnaires to the respondents to obtain the missing data and better answers for ambiguous questions. In some cases, we could not get the additional information and so we were forced to omit questionnaires, estimate our own values for certain data, or drop some data. As a result, we now have more than 82 data points that will be merged with the 74 data points for form a new data base with a minimum of 156 data points.

These data are the inputs to the current statistical analysis. This new analysis is aimed at the following:

TABLE X

RELATIVE ESTIMATING PRECISION FOR VARIOUS EQUATIONS DERIVED IN THE SECOND CYCLE

EQUATION SET	TABLE V N = 74 (INDICES)			TABLE VI N = 67 (TRUNCATED SAMPLE)			TABLE VII N = 26 (SMALL JOBS)			TABLE VIII N = 25 (MEDIUM JOBS)			TABLE IX N = 16 (LARGE JOBS)		
Cost Measure	$2\sigma_e$	$(\bar{Y})$	$\sigma_e/\bar{Y}$	$2\sigma_e$	$(\bar{Y})$	$\sigma_e/\bar{Y}$	$2\sigma_e$	$(\bar{Y})$	$\sigma_e/\bar{Y}$	$2\sigma_e$	$(\bar{Y})$	$\sigma_e/\bar{Y}$	$2\sigma_e$	$(\bar{Y})$	$\sigma_e/\bar{Y}$
Man Months	380	(140)	1.36	108	(55)	0.98	3.3	(4.7)	0.35	42	(33)	0.64	100	(178)	0.28
Computer Hours	2,000	(668)	1.50	586	(278)	1.05	58	(29)	1.00	466	(284)	0.82	836	(672)	0.62
New Machine Language Instructions	84,500	(25,373)	1.67	19,924	(11,912)	0.84	1,770	(1,711)	0.52	17,518	(12,527)	0.70	26,304	(27,531)	0.48
Months Elapsed	13.1	(11)	0.60	10.8	(9.6)	0.56	2.8	(5.1)	0.27	10	(9)	0.56	10.8	(18)	0.30

The table entries--$2\sigma_e(\bar{Y})\ \sigma_e/\bar{Y}$ show (1) the range of the cost measure corresponding to twice the standard error of estimate, (2) the mean of the cost measure in parentheses, and (3) the coefficient of variation. The actual for a cost measure is expected to fall in the range, $\pm\sigma_e$, about the estimate approximately two thirds of the time. The coefficient of variation gives a value for the proportion of the expected error with respect to the mean.

Equation Set refers to the table in which the equations corresponding to the entries are found in this report.

- Deriving equations with improved accuarcy and or usefulness by using subsamples based upon divisions such as size of the cost measures, type of application, and/or appropriate control variables.

- Extending the use of data base, as in the MOL/POL comparison, by testing a series of hypotheses of interest to management, e.g., the assumption that large staffs assigned to a specific programming effort tend to produce fewer source instructions per man month of effort than smaller ones.

- Measuring the improvement in statistical prediction and trying to identify profitable paths for further research.

The general analysis procedure in the third cycle will begin, as in the previous work, with a winnowing of cost factors. Correlation coefficients will be computed for every variable, both cost factors and cost measures, with every other variable. Scatterplots will be produced for each case of high correlation between variables, and for every cost measure (man hours, computer hours, elapsed time, and number of instructions) with each cost factor. In this way, the number of cost factors may be reduced from about 130 at the start to perhaps 20 to 30 of the most statistically powerful and intuitively significant factors as inputs to the multivariate regression used to develop the equations. We plan to test the importance of subsamples by two methods. In one method, we would define a binary variable to be introduced into the regression analysis as a way of dividing the sample into two parts. If this variable took on a significant weight as a predictor, then we would consider the original hypothesis to be supported by our data. This method was used in deriving the equations of Table VI by introducing the variables that characterized computer size and type of programming language. Using the other method, we would divide the data points into subsamples as bases for separate regression analysis as was done to derive the equations in Tables VII through IX.

Although the analysis in the third cycle will be much like that of the previous two, the presentation of the results will be quite different. Where the first and second cycles resulted in the publication of research reports, to record results from the third cycle we will prepare a manager's handbook on cost estimation in addition to a technical report detailing the research methods used and the data analyzed.

The handbook will translate the research results, such as the best obtainable estimating equations, into operationally useful tools for the programming manager. Alternate equations will also be presented for use when the values of some of the independent variables (cost factors) are not known. Also, we will plan to show the loss in statistical precision that occurs when these alternate equations are used. We shall also summarize conclusions about the various hypotheses tested and their importance to management decisions on programming cost. The manager's handbook will include not only the results of the analyses of our own data base, but also summaries of pertinent expert opinion, and material gleaned from the technical literature.

SECTION IV

EVALUATION OF THE RESULTS

To decide whether or how this type of research should be continued, we plan to assess the value of the results during and following the third cycle. Meanwhile, as we have proceeded, we have tried to take an objective look at the results to date and the methods used to obtain them. The following comments are examples from this interim evaluation. Undoubtedly, the work and results in the third cycle will change or add to this appraisal.

A fair question about our cost estimating equations would be: "How good are these euqations for predicting costs on an operational basis?" In the absence of cross-validation to test the results, one can obtain some feeling for an answer by identifying and evaluating the various sources of error for this research. Below we consider potential errors in both the methods and in the data.

- The Analytical Model. The Model described in the early part of Section II has been used in the work to date. Some of the assumptions in this formulation of the cost problem are gross; they are the best we had at the time to proceed with some numerical analysis in a timely way. But we have some indication of the need to improve the Model. In the third cycle, we will test one major assumption, i.e., that major differences in costs between groups or subsamples of programming jobs can be explained by appropriate subsets of the cost factors as they appear in estimating equations. We also have some indications that the definition of a data point may need sharpening by recognizing differences in a hierarchy of programming products such as subroutine, program, program system, as well as the amount of dependence of a member of this hierarchy upon other program systems.

- Selection of Cost Factors. Did we collect data on all of the factors that affect cost? Although we have revised the items in the questionnaire three times, based upon feedback, we have not deliberately accumulated and integrated the feedback with a view to a complete overhaul of the questionnaire. Our feedback suggests that perhaps a separate set of questions should be devised to gather data for each cost measure. For example, to develop the equation or estimating technique for the cost measure, months elapsed, we should probably add questions to indicate how manpower was applied over the actual elapsed time as well as to identify intermediate milestones in computer program production. In our study, we did not gather data on initial schedules used to forecast the time to complete the programming projects; such data may be significant in examining the costs of programming jobs in which resources are overbudgeted but used anyway--or the opposite case when a job is underestimated but overtime is used extensively but not recorded or accounted for in the answers to the questionnaire.

- Appropriateness of Factors. Not all of the factors solicited by the questionnaire are strictly appropriate for all types of programs. For example, size of the data base may have a significant bearing on the cost of a file maintenance program, but little meaning for a compiler. Entering this inappropriate element of data into the analysis thus tends to distort the results.

- Accuracy. As we have indicated, the data points collected to date have been "data of opportunity," i.e., we took what we could get. The types of data needed for precise statistical analysis are difficult to obtain. Many of the items requested in the questionnaire were not readily available because they had not been recorded at all, or such data were not maintained in a centralized location in the particular organization. Therefore, although the answers to the questions for a single data point were reviewed for reasonableness as compared to other data points, and inconsistencies are identified, the accuracy of the individual answers depends upon the effort that the individual respondent devoted to completing the questionnaire. The accuracy of the data is also influenced by the respondents' interpretation of the terms used in the questions.

- Terminology. Although we recognized the lack of standard definitions for terms in computer program production by defining terms used in many questions, many of the responses obtained showed that we needed to do more work on some of these definitions and also add more for terms we had not defined. Misinterpretation of questions was a major reason for returning to the respondents to validate the data.

- Sample. Even if the data were accurate, we don't know how representative our sample is. We probably do not have a truly random sample over the range of values for cost factors and cost variables. Defining a good statistical sample that can serve as a basis for generalizing the analytical results to the population of computer programming jobs is a very difficult problem. The dimensions of the population appear to be growing. For example, if new computers, peripheral equipment, and programming tools actually result in savings, then the rapid changes in these areas would cause the characteristics of the population to change. If the type of application does influence costs, then the rapid introduction of ADP into many new fields would also change the population.

  Growth factors such as these have also increased the spread in the range of practice and possibly also in the ranges of factors that contribute to costs. For example, computers in use since 1950 have been retired recently--many ten-or twelve-year old computers are still in use, although their percentage contribution to the total population is rapidly dwindling. Also, as more and more people become programmers, other veterans keep adding to the amount and variety of their experience.

In any continuation of this type of research, in addition to reexamining and revising the basic model, we would plan to take the following steps to alleviate problems such as those discussed:

- Try to select an important subset of programming jobs as a sample. The object would be to introduce more control and thereby to reduce the observed variation in costs and cost factors. In our current work on the third cycle, we will be searching for characteristics among the cost factors that appear to distinguish major subsets of programming jobs. This work may provide the basis for selection of an appropriate subset of programs.

- Try to consider the changing technology more explicitly by creating new questions that deal with such factors, e.g., computers and their configuration characteristics and languages.

- Expand other questions so as to reduce the number of those that are answered in binary (yes or no) terms.

- Consider new cost factors in preparing a new questionnaire and relate them to more specific hypotheses for each cost measure as well as to the subset selected as a sample.

- Invest more time in defining terms to assure more consistent data in the response.

- Try to include some form of direct coding for the answers in any redesign of the questionnaire format so that the transfer of information from the questionnaire to storage within a computer would be streamlined.

- Try to obtain more reliable data by the personal interview rather than indirect mail and phone contacts.

- Improve the definition of a data point to differentiate as needed among runs, subprograms, programs and program systems.

APPENDIX I

COST FACTORS AND COST MEASURES

USED IN THE QUESTIONNAIRE FOR THE SECOND CYCLE

REQUIREMENTS - COST FACTORS

Operational Requirements and Design

Need for innovation in the system

Programming organization participation in requirements analysis and/or operational design.

Knowledge and documentation of operational requirements.

Number of organizational users communicating with Program Data Point.

Number of ADP centers in system.

Rating of system complexity.

Program Design and Production

Number of machine instructions delivered.

Number of POL instructions delivered.

Number of new machine instructions written for this program.

Number of new POL instructions written for this program.

Number of reused machine instructions from previous programs or libraries.

Number of reused POL instructions from previous programs or libraries.

Number of machine instructions discarded due to operational changes.

Number of POL instructions discarded due to error corrections.

Number of POL instructions discarded due to operational changes.

Number of machine instructions discarded due to error corrections.

Number of words in the data base.

Number of classes of items in the data base.

Number of words in tables and constants not in data base.

Number of types of input messages.

Number of types of output messages.

Rating of program design complexity.

Percentage of instructions classified as:

- clerical
- mathematical
- input/output
- logical - control
- self-checking - FIX
- other (specify)

Percentage of program functions classified as:

- information storage retrieval
- data acquisition and display
- control or regulation
- decision making; choosing an optimum
- transformation; reformatting data
- generation to produce desired output
- other (specify)

Average operate time of completed program.

Frequency of program cycle or operation.

Occurrence of constraints classified as:

- insufficient memory capacity
- insufficient I/O capacity
- stringent timing requirements
- other (specify)

Number of subprograms in this program system.

Programming language used in coding.

Number of support instructions.

Number of support programs used.

Number of free support programs available.

Number of free support instructions available.

Documented specifications of test data and expected outputs.

Number of distinct internal documents.

Number of distinct external documents.

Number of pages of internal documentation.

Number of pages of external documentation.

RESOURCES - COST FACTORS

Data Processing Equipment

Developmental computer and number of words in core storage.

First development effort on computer.

Average turnaround time for computer run.

Number of ADP components developed concurrently with program.

Number and types of display equipment driven by program.

Number and types of I/O equipment.

Programming Personnel

Number of programmers classified as:

. coder . programmer . senior programmer . system programmer

Years of experience for each category of programmer with

- language used
- computer used
- specific application

Number of programmers participating in design.

Number of programmers for entire project.

Average programmer turnover rate.

MANAGEMENT AND ENVIRONMENT - COST FACTORS

Management Procedures

Existence of a documented management plan for:

- processing of system design changes.
- processing of program design changes.
- dissemination of error-detection and error-correction information.
- use of computer facility.
- contingency for computer unavailability.
- communication with other agencies.
- design specification concurrence procedures.
- cost control.
- management information control.
- document control.
- standards for coding, flow charting.

Development Environment

Number of agencies concurring on design specifications.

Extent of customer experience in information processing system development.

Computer operated by agency other than the program developer.

Computer facility operated on the basis of

- open shop
- closed shop
- time-sharing

Program developed at site other than operational location.

Computer at operational site different than developmental computer.

Program development at more than one location.

COST MEASURES

Number of man months to design, code, test, and document program.

Number of man months to develop utility programs.

Maximum number of programmers.

Number of months that more than 90 percent of maximum number of programmers were employed.

Start date for program design.

Completion date for program delivery.

Number of computer hours used by type of computer.

Number of man trips.

Average round-trip distance per trip.

APPEND

DEFINITIONS AND CODING FOR VAR

Cost Variables

Y_1 - Total number of man months including firs
test and document this program not includ
program.

Y_2 - Total number of computer hours used by a

Y_3 - Number of new machine language instructic
reused subroutines, logical blocks, and s

Y_4 - Months elapsed--completion data for progr
At the time of program delivery the progr
computer to begin system test. The progr
description and operational specificatior
and flow charts.

Z_1 - Logarithm to the base ten of Y_1.

Z_2 - Logarithm to the base ten of Y_2.

Z_3 - Logarithm to the base ten of Y_3.

Z_4 - Logarithm to the base ten of Y_4.

NOTE: The variables X_{12}, X_{13}, and X_{14} cc
Appendix I. The variable X_{12} is c
program by percentage of instructi
The levels are a gross way to indi
programmer's point of view.

The variables X_{13} and X_{14} stem fr
a computer program. Percentage o
is oriented toward a user's descri
percent generation (X_{14}) refers t
to the creation of information fr
process. The purpose or function
from a given set of parameters wo
program, there may also be inform
ated transformation of data. A cc
But a computer program to convert
or reformatting of data (X_{13}), pe
outputs.

procedure-oriented or compiler language for
bly symbolic language source statements.

ystems, coded small = 0; large = 1. Machines
emory are small--those with more than 16,000

system, coded yes = 1; no = 0. Innovation
a known programming technique and/or a
new to the people involved.

esign, coded yes = 1; no = 0.

1; no = 0.

d yes = 1; no = 0.

ivisions in the program design for logical

. Classes means categories of types of
states or any characteristics of information

the development of automatic data processing
= 1.

per programmers participating in design
Maximum number of programmers

ments analysis conducted to specify in
mation processing system, and the operational
nto operational design specifications that

ookkeeping, sorting, searching, and file
tical input/output, logical control and
ide of this page).

ctions, coded in decimal, as compared with
ntrol, data acquisition and display, and
ide of this page).

utputs, coded in decimal, as compared with
sformation functions (see note on opposite

an the computer used for program development,

programmers. Turnaround time is the total
rn of a computer run.

e is the subset of tables that describe the
lving and/or the files to be processed. If
ndicate an average size.

r of unique displays or reports (these may
ts).

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
System Development Corp 2500 Colorado Ave. Santa Monica, Calif. 90406	Unclassified 2b. GROUP N/A

3. REPORT TITLE

DEVELOPMENT OF EQUATIONS FOR ESTIMATING THE COSTS OF COMPUTER PROGRAM PRODUCTION

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (Last name, first name, initial)

V. LaBolle

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
JUNE 1966		

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
AF19(628)-5166 b. PROJECT NO. c. d.	TM-2918/000/00 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) # ESD-TR-66-350

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Director of Computers, Deputy for Engineering & Technology, Electronic Systems Division, AFSC, USAF, L.G. Hanscom Field, Bedford, Mass.

13. ABSTRACT

This report summarizes System Development Corporation (SDC) Technical Memorandum TM-2712, Research Into the management of Computer Programming: A Transitional Analysis of Cost Estimation Techniques, 12 Nov 65. That report supplies details of the early results obtained in a second cycle of continuing exploratory research to develop equations for estimating the costs of computer program production--computer program design, code, and test. Additional sets of equations developed after TM-2712 was published are also given in this report. Each set contains four equations; each equation shows how to form an estimate for one of the cost measures--number of man months, computer hours, new machine language instructions, months elapsed--by combining numerical values for selected factors that influence these costs.

This report reviews the development of these equations including the application of statistical methods such as correlation and multivariate regression to experience data that characterize 74 computer programming efforts completed at SDC. The earlier work in the first cycle, a similar analysis of data for 27 SDC computer programming efforts, is also described as well as the plans for the current analysis in the third cycle using these SDC data and new data for more than 80 efforts completed by computer programming organizations in industry and the Air Force.

After the publication of TM-2712, the second cycle was continued by additional analysis of the same SDC data for 74 computer programming efforts. The aim of the additional work was to improve the estimating precision of the equations presented in TM-2712. The improvements reported were achieved by deriving new cost equations, one set based upon a truncated sample and then three sets based upon three subsamples of the data. An interim evaluation of the work completed in the first and second cycles presents proposed improvements in approach and research methods.

DD FORM 1 JAN 64 1473

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer Program Costs Computer Program Production Equations for Computer Cost Estimation						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, 8*c*, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional