This document has been approved for public release and sale; its distribution is unlimited.

DEPARTMENT OF THE AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GCA/LSY/91S-2

(1)

A COMPARATIVE STUDY OF THE RELIABILITY OF FUNCTION POINT ANALYSIS IN SOFTWARE DEVELOPMENT EFFORT ESTIMATION MODELS

THESIS

Robert B. Gurner, Captain, USAF

AFIT/GCA/LSY/91S-2

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GCA/LSY/91S-2

A COMPARATIVE STUDY OF THE RELIABILITY OF FUNCTION POINT ANALYSIS IN SOFTWARE DEVELOPMENT EFFORT ESTIMATION MODELS

THESIS

Presented to the Faculty of the School of Systems and Logistics
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Cost Analysis

Robert B. Gurner, B. Accy
Captain, USAF

September 1991

Approved for public release; distribution unlimited

Preface

The purpose of this research was to examine the use of function point analysis in estimating software development effort. More broadly, the information in this study can be an introduction to a new software estimation tool for members of the Air Force and DOD community.

Three personal computer based estimating models were used to predict effort for a set of 36 completed business/database software projects. The results demonstrated that each model is a reliable predictor of development effort. However, each model showed a bias toward high estimates. Further study is needed to determine if the bias is due to differences in productivity levels between private industry and the military environment.

I would like to extend deep appreciation to Mr. Dan Ferens, my thesis advisor, for his guidance and patience. Additional thanks are extended to Dr. Richard Werling, of the Software Productivity Consortium, and Capt Maurice Griffin, of the Standard System Center, Gunter AFB, who provided substance and a focus to my research. Also, I wish to thank the faculty and students associated with Cost Analysis program for being a good sounding board. Finally, a special thanks to my wife and friends for just nodding their heads and understanding when I rattled on and on.

Robert B. Gurner

Table of Contents

	Page
Preface	ii
List of Tables	v
Abstract	vi
I. Introduction	1
General Issue	1
Definitions	3
Specific Problem	4
Research Question	5
Assumptions	5
II. Background	7
Overview	7
Software Metrics	7
PERT Sizing	9
Parametric Sizing	9
Data Base Analogy	10
Albrecht's Function Points	10
III. Methodology	15
Overview	15
Data Description and Availability	15
Statistical Analysis	18
Linear Regression	18
Wilcoxin T	22
Percent Error	22
Drawing Conclusions	23
Software Cost Estimation Methods	24
Costar™	24
Checkpoint™	25
SPANS	26
Conclusions	26

	Page
IV. Analysis and Findings	28
Data Analysis	28
Compatibility	28
Outliers	30
Prediction Analysis	33
LSBF Regression	33
Wilcoxin T	35
Percent Error	36
V. Conclusions and Recommendations	41
Conclusions	41
Recommendations	44
Appendix A: Function Point Count Sheet	46
Appendix B: LSBF Analysis for Data Compatibility Tests	47
Appendix C: Outlier Analysis	66
Appendix D: LSBF Analysis for Model Comparison	68
Appendix E: Wilcoxin T Data	71
Appendix F: Percent Error Data	74
Bibliography	80
Vita	83

List of Tables

		Page
1.	Degrees of Influence for General System Characteristics	13
2.	The Albrecht Research Data Set	16
3.	Kemmerer Research Data Set	17
4.	LSBF Linear Equation and Transformations	19
5.	Statistical Summary of Model Prediction Reliability	34
6.	Wilcoxin T Statistics for Bias Analysis	35
7.	Mean and Standard Deviation From Percent Error Test	36
8.	Model Predictions Within A Predescribed Range . .	38
9.	Summary of Adjusted Data Percent Error Test Results	39

Abstract

The Air Force of the 1990's is steadily growing more reliant on software systems. However, the struggle to develop reliable cost and effort estimation tools continues. The Standard Systems Center (SSC), Gunter AFB AL, has adopted the use of Function Point Analysis to improve estimation of data processing, management and communication systems. Function point analysis was introduced by IBM's Alan Albrecht in 1979 as an alternative to source line of code (SLOC) as a size and productivity measure.

In 1991, Tecelote Research, Inc., under contract to the SSC, delivered the Software Program Acquisition Network Simulation (SPANS) model incorporating the capability to perform estimates using function point measures. This research examines the ability of SPANS to reliably and accurately estimate software project effort with function points. A further investigation compares the predictions derived by SPANS with two other software estimation tools, Checkpoint and Costar.

A COMPARATIVE STUDY OF THE RELIABILITY OF FUNCTION POINT ANALYSIS IN SOFTWARE DEVELOPMENT EFFORT ESTIMATION MODELS

I. Introduction

This chapter provides an overview of the research. First, the general issue of increasing software costs and the inability to accurately measure those costs is addressed. Second, the important terms are defined. Next, the specific problem of software cost estimation is examined. Finally, the specific research questions are enumerated along with some general assumptions.

General Issue

The impact of software costs on the Air Force budget is steadily growing (5:14). Some calculations estimate software costs in this country at 13% of gross national product (9:52). Considering the Air Force's relative advanced technology, its share is probably much higher. Unfortunately, the ability to estimate these costs has not kept pace, and has even been likened to witchcraft (17:1).

Many authorities and organizations in the field have expressed deep concern over this lack of accuracy in software estimation (19:64, 18:416, 24:1). In response, the Standard Systems Center (SSC), Air Force Communications Agency, Gunter AFB, contracted Tecelote Research, Inc., to develop a computer-based model that will better quantify software development effort and schedule estimates (10:2). In fulfillment of this contract, Tecelote developed the Software Program Acquisition Network Simulation (SPANS) model.

Many computer-based models for software cost estimation are in existence (24:1). Most models use SLOC as the basis for predicting costs; examples are the COnstructive COst MOdel (COCOMO), PRICE-S, System Evaluation and Estimation of Resources (SEER), Software Life Cycle Management (SLIM), Softcost-R. A few models attempt to use methods other than SLOC. SPANS, Checkpoint and Costar use derivatives of Albrecht's function points for sizing in addition to SLOC. Other techniques for software estimation include expert analysis, bottom-up or 'grass roots' engineering estimates and analogy (9:24).

The focus of this research is on the effectiveness of the SPANS model's use of function points for estimation of development effort (measured in man-months). Since development costs are a direct derivative of effort, references to cost or effort in this document can be considered synonymous. SPANS employs two function point algorithms, Albrecht's and one calibrated to a set of projects developed at the SSC (12:2-19,2-23). Tecelote incorporated the most recent function point data available from the SSC in developing the second algorithm. This leaves insufficient independent data for validating this unique formulation at this time. Therefore, only SPANS' general function point sizing capability will be tested. Specific information about SPANS' estimating capability is provided in a later chapter.

Definitions

The following terms are important to this research:

SLOC: "An instruction written in assembler or higher order language is often referred to as a source line of code (SLOC) to differentiate it from a machine instruction" (8:3).

Software Sizing: An attempt to quantify the size of software projects in a form that decision makers can use. Considered a prime driver in the cost of software (13:92, 16:2-3).

Function Points: Determined through a formula that sums the "weighted number and complexity of the various inputs, outputs, calculations and databases required" in a software package as described by Albrecht (13:91).

SPANS: Software Program Acquisition Network Simulation. Software development effort and schedule estimation tool developed by Tecelote, Inc. The SPANS model for software estimation with function points is the subject of this research (12:2-4).

Multiple Regression: "A statistical tool that utilizes the relation between two or more quantitative variables so that one variable can be predicted from the others"

(21:23). Also referred to as least-squares best-fit (LSBF) statistics.

Non-Parametric Statistics: Statistical test procedures used to base sample inferences on populations when the assumption of a normal distribution is not practical (22:392).

Specific Problem

SPANS calculates development effort using COCOMO, Albrecht's function point algorithm or a function point algorithm tailored for data collected on management information systems projects at the Standard Systems Center. (Note that only the Albrecht function point module is used in this research.) SPANS additionally adjusts all estimates for parameters based on user input. These inputs include productivity level, staffing level and CPM analysis of the development process. Tecelote's testing procedures reveal the ability of all algorithms to accurately and reliably estimate effort costs and schedules for these systems.

While this testing demonstrates the statistical prowess of the algorithms, it does not address an important element of validity: Whether this model provides a needed or improved capability of software estimation. Do the benefits of the model outweigh the costs of development, training and maintenance of SPANS, in lieu of currently available models? This question is important not only to the SSC, but to other organizations considering development of models calibrated for their specific circumstances. The main purpose for the development of SPANS is to provide a comprehensive scheduling and management tool; so this must be taken in consideration in any evaluation of the package's effectiveness. However, no software estimating or scheduling tool can be judged effective if its software size prediction capabilities are inaccurate.

Research Question

Two research questions are addressed in this thesis:

1. How well does SPANS' function point module predict software development effort?

2. How well does SPANS estimate software development effort as compared to other models using function points?

Assumptions

A vital assumption of this research is the ability of Albrecht's function point methods to provide an accurate statistical software measure. Several studies address this topic positively (3:639-647, 4:648-652, 15:1); however, some point out that Albrecht's formulation is valid only when used within the management information (MIS) or business systems realm (11:4, 8:3). This should not hamper this research since the data collected will be from MIS or comparable systems.

Another assumption is that the function point counts in the data sets are valid for this research. The Methodology chapter will address the sources of the data and describe the general characteristics of the set.

Finally, each model is set up to estimate effort, for the projects in this research, as a military project. SPANS, an Air Force product, always assumes it is estimating in a military environment (AFR-700, AFR-800 or DOD-STD-2167A (12:2-6). Checkpoint and Costar allow the user to select the proper environmental factors.

II. *Background*

Overview

This chapter presents a review of the literature important to this research. The general discussion centers around the techniques of various software metrics. The conclusion is a relatively detailed explanation of the role of function point analysis in software estimation.

Software Metrics

Nearly every study of software cost estimation, or software project management for that matter, names size as the major cost driver. It follows that size is the key input to most software cost models (8:1). That is where widespread agreement ends. "The biggest difficulty in using today's algorithmic software cost models is the problem of providing sound size estimates," states Dr. Barry Boehm, developer of the Constructive Cost Model (COCOMO) and prominent author in the field of software estimation (5:148).

The most widely recognized measure of software size is SLOC. However, there are many different ways to define a line of code. Capers Jones lists six variations of SLOC at the program level and five variations at the project level (19:64). "SLOC was selected early as a metric by researchers, no doubt due to its quantifiability and seeming objectivity," according to Chris Kemmerer of the Sloan

School (18:417). SLOC is easily quantifiable, but does not relate functionality. If used as a singular measure, it can lead to a "mindless maximization" of inefficient code (13:309).

Function point analysis, while sometimes used to calculate SLOC, was developed to measure the productivity in the development process and functionality of the software (2:34-36). Albrecht described three advantages of using function points rather than SLOC:

> First, it is possible to estimate them early in the life cycle, about the time of the requirements definition document. Second, they can be estimated by a relatively nontechnical project member. Finally, they avoid the effects of language and other implementation differences. (18:418)

There is some variation in the application of function points as a size measure. The initial research was correlated to MIS and data processing (8:3). Recent research shows that the function point algorithms must be adjusted before they can be used on "real-time or embedded" systems, such as weapon systems (24:2-5). Capers Jones calls this real-time metric feature points. This method, found in the Checkpoint model, adds a category for system algorithms and reduces the weighting of data files (16:83).

Reese and Tamulevicz relate four techniques of software sizing: 1. PERT Sizing, 2. Parametric Sizing Tools, 3. Data Base Analogy, and 4. Albrecht's Function Points (23:38) The following is a brief description of the four techniques.

PERT Sizing. The Program Evaluation and Review Technique (PERT) is the most common technique of expert judgement (1:17). It is based on the assumption that expert can provide accurate opinions on software size through past experience. Each expert generates a most likely estimate (m), usually in KSLOC, and a lower (a) and upper (b) bound of the estimate. The average estimated size (E) is determined as follows (8:2):

$$E=\frac{a+4m+b}{6} \tag{1}$$

Reese and Tamulevicz note that there is bias in this technique, "...the m_i's cluster toward the lower limit resulting in an underestimation" (23:29).

Parametric Sizing. This is the sizing technique used in many of the software cost models on the market (COCOMO, SLIM, and PRICE-S, for example).

> These models use input parameters consisting of numerical or descriptive values of selected program attributes...Parametric models have the advantage of considering many different program facets and calibration capability, when present, allows the user to fine tune a parametric model to specific applications. (8:3-4)

While parametric models are efficient and objective, they also can be inflexible (23:47). Data needs to be in a very specific format. Again, KSLOC is the usual input variable. Additionally, "since these models must be calibrated from historical data, their applicability to new, unique programs may be uncertain" (8:4).

Data Base Analogy. This approach attempts to estimate size by comparing the development effort to existing software. A simple equation for estimation is:

$$S = F \times (\mathit{SizeofSimilarProducts}) \tag{2}$$

'S' is the estimated size and 'F' is an adjusting factor "usually determined by experience or politics" (8:1). "Three attributes influence the 'F' factor: complexity, application environment, and the extensiveness of the project requirements" (1:20).

This technique is widely used even though many project analysts and engineers in industry have confidence in it (24:1). Another problem with this approach is a heavy reliance on historical data. The availability and conformance of historical data greatly influences the reliability of the estimates (23:42).

Albrecht's Function Points. Albrecht developed the function point methodology while associated with IBM and published his findings in 1979 (2:33). As defined earlier:

> Determined through a formula that sums the "weighted number and complexity of the various inputs, outputs, calculations and databases required" in a software package. (13:91)

His goal is to establish a measure of work-effort and productivity. Three major reasons cited for the utility of function points are 1) determination of customers functional requirements at an early stage; 2) availability of information about basic system requirements (e.g., inputs and outputs); and 3) translation to productivity measures like "function points per man-month (FP/MM)" (3:639).

Function points are at a less technical level when compared to SLOC. This view, coupled with the availability of basic requirements information, allows estimates to be performed and understood by nontechnical project members (18:418). Additionally, the reliability of function points is limited to business-based or MIS systems (24:2, 16:83). Reifer Consultants, Inc., and Software Productivity Research, Inc. (SPR), have both introduced real-time system estimation models that are extensions or derivatives of Albrecht's research. Each recognizes the impact of extensive use of operators and algorithms in lieu of logical files in real-time systems. Reifer's real-time and scientific function point estimators are included in the company's ASSET-R model (24:7). SPR's feature points are included in the Checkpoint model being investigated in this paper.

Function points are evaluated in two parts. The first is a count of "pure" or unadjusted function points, or the function count. Second is a complexity rating of fourteen measures of general system characteristics of functionality for the application (14:4). The 14 characteristics are used to derive a Value Adjustment Factor (VAF). Total function points are found by weighting the function count by the VAF.

The following is a description of unadjusted function points and the general system characteristics. While the guidelines will follow the format found in IFPUG counting manuals, they conform to commonly accepted conventions of function point counting methods (2,4,13,18,24).

There are five categories of functionality used in determining the function count. The International Function Point User's Group (IFPUG) Counting Practices Manual defines them as:

> *Data Types* represent the functionality provided to the user to meet internal and external data requirements.
>
> Internal Logical Files (ILF) reside internal to an application's boundary and reflect data storage functionality provided to the user. ILFs must be maintained and utilized by the application.
>
> External Logical Files (EIF) reside external to an application's boundary and reflect the functionality provided by the application through the use of data maintained by other applications.
>
> *Transactional Types* represent the functionality provided the user for the processing of data.
>
> External Inputs (EI) reflect the functionality provided the user for the receipt and maintenance (add, change, and delete) of data on ILFs.
>
> External Outputs (EO) reflect the functionality provided the user for output generated by the application from ILFs or EIFs.
>
> External Inquiries (EQ) reflect the functionality provided the user for queries of ILFs or EIFs. (14:9-10)

As each function is counted, it is assigned a complexity factor of low, average or high. This rating determines the weight given that function. The weights vary by function type. An example of a count sheet is found in Appendix A. The basic function count equation that IFPUG espouses is the same as the original Albrecht equation:

$$FunctionCount = (10*ILF) + (7*EIF) + (4*EI) + (5*EO) + (4*EQ) \tag{3}$$

This equation is weighted for average function inputs, outputs and files. Weights for low and average types can be seen in Appendix A.

The second part of the function point evaluation is an analysis of the 14 General System Characteristics (GSC). Each GSC is given a score of zero to five Degrees of Influence (DI). The GSCs can be seen on the count sheet in Appendix A. The following is a list of the DIs:

TABLE 1.

Degrees of Influence for General System Characteristics

0	Not Present, or no influence
1	Incidental influence
2	Moderate influence
3	Average influence
4	Significant influence
5	Strong influence throughout

Once each GSC is rated the DIs are summed to form a total DI (TDI). The TDI is entered into an equation to determine the Value Adjustment Factor (VAF):

$$VAF=(TDI\times 0.01)+0.65 \tag{4}$$

The product of Eq (4) (VAF) and Eq (3) (function count) yields the total function points.

$$FunctionPoints=(FunctionCount)\times VAF \tag{5}$$

While there is a concerted effort on the part of several organizations and researchers, led by the IFPUG, to standardize counting practices, definitions and terminology, differences do exist. This paper does not attempt a comprehensive explanation of function point definitions, calculations and variations. A more thorough treatment can be found in the documents listed as references for this section.

III. Methodology

Overview

This chapter contains a description of the elements of the research analysis. The first section introduces the data set and discusses its characteristics and limitations. Next is an outline of the different statistical techniques used in analyzing the data. The final section is a brief description of the estimation models compared in this study.

Data Description and Availability

The acquisition of data, in the format needed for SPANS and other models, is the most crucial part of the research process. The area of software measurement dealing with function points, while over a decade old, is growing slowly. There are relatively few software projects in which function points have been counted. The Standard Systems Center, a sponsor of this research, the Logistics Management Systems Center, the Software Productivity Consortium, and the Software Engineering Institute at Carnegie Mellon University, were enlisted in the search for acceptable data.

The requirement for this research is a set of data points containing function point counts and actual effort expended in project development. More specifically, the data should be based on MIS or comparable projects. Richard Werling, of the Software Productivity Consortium, provided two data sets. The first set is from Albrecht's 1983 article in the IEEE Transactions on Software Engineering (3:639-648). This article, published jointly with John Gaffney, is a follow-up to Albrecht's 1979 paper introducing function points as a software metric (2:33-34).

The Albrecht data set contains 24 projects (Table 2). Eighteen of the projects use COBOL programming language, four use PL/I and the remaining two are IBM/DMS (International Business Machines/Database Management System). These languages are common for MIS programming and meet the criteria for this research.

TABLE 2.

The Albrecht Research Data Set

PROJECT	LANGUAGE	KSLOC	FUNCTION COUNT	FUNCTION POINTS	COMPLEXITY	ACTUAL EFFORT
1	COBOL	130	1750	1750	1.00	673.7
2	COBOL	318	1902	1902	1.00	692.1
3	COBOL	20	522	428	0.82	73.0
4	PL/1	54	660	759	1.15	138.8
5	COBOL	62	479	431	0.90	189.5
6	COBOL	28	377	283	0.75	65.8
7	COBOL	35	256	205	0.80	52.6
8	COBOL	30	263	289	1.10	32.2
9	COBOL	48	716	680	0.95	84.9
10	COBOL	93	690	794	1.15	125.0
11	COBOL	57	465	512	1.10	71.1
12	COBOL	22	299	224	0.75	19.1
13	COBOL	24	491	417	0.85	49.3
14	PL/1	42	802	682	0.85	78.9
15	COBOL	40	220	209	0.95	27.0
16	COBOL	96	488	512	1.05	103.9
17	PL/1	40	551	606	1.10	120.4
18	COBOL	52	364	400	1.10	58.6
19	COBOL	94	1074	1235	1.15	250.7
20	PL/1	110	1310	1572	1.20	402.6
21	COBOL	15	476	500	1.05	23.6
22	DMS	24	694	694	1.00	77.6
23	DMS	3	166	199	1.20	3.3
24	COBOL	29	263	260	0.99	40.1
Mean				647.6	1.00	143.9

The projects in this data set range from small (3 KSLOC) to large (318 KSLOC), with a mean of approximately 66 KSLOC. The data set, as provided, did not contain function counts for external interfaces (EIF). As explained in the Albrecht and Gaffney paper, interfaces are included in the count of internal logical files (ILF) (3:641).

The second set of data is from Kemmerer's 1987 paper on software cost estimation model validation (18:416-429). 13 of the 15 projects included in this set are all programmed in COBOL. Projects seven and fifteen are written in BLISS and Natural, respectively. The data is shown in Table 3.

TABLE 3.

Kemmerer Research Data Set

PROJECT	LANGUAGE	KSLOC	FUNCTION COUNT	FUNCTION POINTS	COMPLEXITY	ACTUAL EFFORT
1	COBOL	254	1010	1217	1.20	287.0
2	COBOL	214	881	788	0.89	86.9
3	COBOL	254	1603	1611	1.00	258.7
4	COBOL	41	457	507	1.11	95.5
5	COBOL	450	2284	2307	1.01	1107.3
6	COBOL	450	1583	1338	0.85	336.3
7	BLISS	50	411	421	1.02	84.0
8	COBOL	43	97	100	1.03	23.2
9	COBOL	200	998	993	0.99	130.3
10	COBOL	39	250	240	0.96	72.0
11	COBOL	129	724	789	1.00	230.7
12	COBOL	289	1554	1593	1.09	116.0
13	COBOL	161	705	691	0.98	157.0
14	COBOL	165	1375	1348	0.98	246.9
15	NATURAL	60	976	1044	1.07	69.9
Mean				1000.3	1.01	228.8

These projects are medium to large in size. They range in size from 39 KSLOC to 450 KSLOC, with a mean of just under 187 KSLOC (18:419). Although the range is similar to the Albrecht data set, the average size of the projects is nearly three times larger. Statistical tests performed to determine the compatibility of the two sets will be discussed in the analysis portion of this document.

Statistical Analysis

Different statistical methods are employed to analyze the models' estimating reliability. Linear regression, the Wilcoxin 'T' and Percentage Error each test a different aspect of reliability.

Linear Regression. The first statistical tool used to test the two data sets is least-squares best fit (LSBF) linear regression. This is a parametric technique (assumes a normal distribution for the population). Normal distribution is assumed in this study by invoking the Central Limit Theorem:

> Whatever the distribution of X_i (population or sample), as the number of terms of n in the set becomes large ($n \geq 30$), the distribution tends to the standard normal. (22:214)

Combining the Albrecht and Kemmerer data sets creates a sample data set of 39 observations and can be considered large.

Besides the basic linear equation, transformations can be performed on the dependent and independent variables to better explain their statistical relationship. The following are the basic and transformed variants of the LSBF equation:

TABLE 4.

LSBF Linear Equation and Transformations

Linear Equation	$Y=b_0+b_1(X)$	(6)
Exponential	$Y=b_0*e^{b_1(X)}$	(7)
Logarithmic	$Y=b_0+b_1*\ln(X)$	(8)
Power Curve	$Y=b_0*X^{b_1}$	(9)

Where: Y = the set of actual effort

b_0 = the intercept term

X = the set of estimated effort

b_1 = the coefficient/exponent of the estimate

The model estimates (X) are the independent variable. The reported actuals from the data sets are the dependent variable (Y). Analysis of the statistics and visual inspections of the graphs are used to determine the proper equation for each data set.

Statistical analysis is performed on the StatPak, a commercial statistics package for desktop computers and the Statistical Analysis System (SAS) available at AFIT. The resulting statistics used for analysis are the F-Ratio, coefficient of determination (R^2) and values found in the analysis of variance (ANOVA) table. The F statistic is an indication of the level of significance in the relationship between the actuals and the estimates. R^2 measures the amount of variation in the actuals explained by the estimates, or the reliability of the estimates. An R^2 of .80 or better is considered reliable for this research. The ANOVA table is "a breakdown of the total sums of squares and associated degrees of freedom" (21:92). In the appendices the ANOVA table also contains the F and R^2 statistics.

Once regression equations are selected for each data set, additional analysis is performed to determine the statistical compatibility of the two data sets. Two variations of the F-Ratio test are used in this analysis, one for variance in the data and one for the error terms of the regression equations. The F statistic for the first test is obtained by Eq (10).

$$F_{calc}=\frac{MSE_A}{MSE_K} \tag{10}$$

The MSE, or standard error, is obtained from the SAS output. The F_{calc} is compared to the respective F_{crit} in tables found in various texts (21:629-639). The hypothesis being tested in this case is that the variances in each data set are equal. The hypothesis is accepted if F_{calc} is less than F_{crit}

for a given level of confidence. For this research, a 99% confidence level, or $\alpha = .01$, is required.

The F-Ratio test for comparing the equation error terms is determined through the use of the Full-Reduced technique. For this research the full model is the data set containing all data points from both sets. The reduced models are the Albrecht and Kemmerer data. F_{calc} is derived by Eq (11):

$$F_{calc}=\frac{SSE_F-SSE_R}{Df_F-Df_R}\div\frac{SSE_F}{Df_F} \tag{11}$$

Where: SSE = Sum of squares error for the respective data sets,

Df = Degrees of freedom for the respective data set (21:98-100).

The hypothesis for this test is that the full model does not explain significantly more of the variability in the error terms than reduced models (21:99). This comparison is made for both data sets. The hypothesis is supported if F_{calc} is less than F_{crit}.

A final LSBF treatment compares the model predictions to the project actuals. As before, the four LSBF transformations are run with the predictions and actuals for each model. Examination of the graphic plots and the ANOVA table are employed to determine the reliability of the predictions.

Wilcoxin T. The Wilcoxin T is a non-parametric test that "provides a method of incorporating information about the relative sizes of differences" (22:398). In this research the estimates from each model are compared to the actuals to determine the magnitude of the ranked differences. The ranks are grouped as positive or negative and summed. The lowest sum is called the T score (22:399). The generated T score tests the hypothesis that the differences are equally distributed around zero. The Wilcoxin T table provides cutoff points for the hypothesis test for varying sample sizes (n) and confidence levels (α). For small samples the T score is compared to a value found in the table. Large samples (non-zero differences ≥ 20) use the T-statistic that is compared to the student t value.

$$T_{stat}=\frac{T_{score}-\mu_T}{\sigma_T} \tag{12}$$

The purpose of running this test is to determine if their is a bias in the estimates. For example, a very low T_{score}, or high T_{stat}, indicates a bias toward a center other than zero. The bias is negative or positive depending on which group provided the T.

Percent Error. Percent methods are simple and easy techniques to understand and evaluate the accuracy of man-month estimates (18:420). Three different aspects of percent error will be considered. The first two methods are used by Kemmerer in his research.

The first is a percentage error test recommended by Boehm which normalizes the error for size (6:49):

$$Percent_{e}=\frac{MM_{est}-MM_{act}}{MM_{act}} \tag{13}$$

The mean of this test will indicate the estimates' bias toward high or low estimating. The standard deviation shows how widely the estimates are dispersed around the mean. In populations that are normally distributed, 65% of the estimates would be expected to fall within one standard deviation of the mean and 95% within two standard deviations (22:20).

The second percentage method is the magnitude of relative error (MRE). This method uses the absolute value of the formula found in Eq (11). This test does not allow the negative and positive errors to "cancel each other out" (18:420).

The final percentage method is to evaluate the number of estimates that fall within a predescribed range of error. Ranges of 20 and 30 percent around the actual will be used. "This is a method used by many model developers to tout their products" and is easy to understand (6:50).

Drawing Conclusions. The statistical analysis of the test results is not limited to the internal significance and reliability of the model with the test data set. The true effectiveness of the model will be its ability to estimate software cost better than models presently available, and its consistent utility to organizations employing the model.

Software Cost Estimation Methods

Three automated software estimation products are used in the analysis of the data. They are Costar, Checkpoint, and SPANS. As mentioned earlier, the SPANS model is the catalyst for this research. Costar and Checkpoint are used for comparison. These two models have the capability of providing estimates for the full life cycle of software projects, as opposed to SPANS which only addressed the development phase. For the purposes of this research, estimates for Costar and Checkpoint reflect only those development cycles SPANS encompasses.

Each model is discussed in this section. The description is very general. Detail is provided only when it pertains directly to the features of the packages incidental to this study.

1. *Costar*™. " Costar is a software cost estimation tool based on the COnstructive COst MOdel (COCOMO)" (25:3). COCOMO, developed by Barry Beohm, is a widely used estimation tool based upon deliverable source instructions (DSI). DSI are source lines that are delivered to the customer excluding test drivers, automatically generated code and comments (26:2). Additional information about

COCOMO can be found in Dr. Beohm's book *Software Engineering Economics* (5).

While Costar calculates effort in man-months for DSI, it does not do this directly for function points. (Total function points can be entered directly or derived from a worksheet provided by Costar.) To estimate effort, function points are converted into DSI by a linear multiplier for a given language (25:45). For example, each function point translates into 91 DSI for COBOL. Costar provides conversion factors for some of the most popular software languages, or the factor and language can be entered and saved into a Costar worksheet.

2. *Checkpoint*™. This product is a software scheduling and estimating tool developed by Software Productivity Research, Inc. (SPR) and incorporates many of the features found in SPR's SPQR-20. Checkpoint uses proprietary SLOC algorithms and variations of Albrecht's function points to calculate effort. Feature point sizing is also available in this program. Calculations of effort using function points rely heavily on productivity factors developed by Capers Jones, the founder and chairman of SPR (16:494). An extensive study of software management and productivity is available in Jones's book, *Applied Software Measurement* (16).

SLOC conversion factors for Costar and productivity factors for SPANS that required user input are taken from the tables found in Appendix A of the Checkpoint manual

(27:A,15-24). This provides a consistent basis of model inputs, and thus, a more reliable comparison of the outputs.

3. ***SPANS***. This model is the focus of the research and was briefly described earlier in this paper. It is a scheduling and estimating tool for the development phases of software projects. Its uniqueness is a function point estimating module calibrated by Tecelote to past projects from the Air Force Standard Systems Center (9:2, 12:2-23). While this research examines the general use of function points as a software estimator, it does not address the more specific question of the effectiveness of SPANS for Air Force use.

The SPANS model derives estimates for military projects only. The two other models in this research are run in a military mode (e.g. DOD-STD-2167A) for consistency in comparisons. Capers Jones shows that military projects historically have lower productivity, and thus higher costs, due to the many constraints found in military regulations (16:19-20). Since the data used in this study are based on private sector firms, this could lead to an assumption that the estimates may tend to be biased high when compared to the actual effort reported. The results of the analysis will tell whether or not this assumption can be supported.

Conclusions

The use of software size is obviously important in the estimation of software costs. What are less obvious are the specific techniques, methods, or models best applied to derive accurate and reliable size estimates. SLOC is the most studied and utilized method; however, SLOC ignores the measure of functionality needed to improve software development productivity. Function points have been shown to capture that functionality measure and allow for accurate cost estimates, given that function point counts can be expected to provide reliable and accurate predictions in the early stages of software development. The next step has been to develop estimating tools or models utilizing function points that can help the analyst derive an accurate estimate. This research will attempt to discover whether these models are fulfilling that need.

The models in this research apply additional mathematical treatments to the function counts based on complexity factors and parameters beyond the basic Albrecht formula. Since these additional features are not consistent across the models, care is taken to enter the data and set up parameters in as congruous a fashion as possible. Also, the data sets do not always contain all of the detail required for input in particular model. When this occurs, the author will choose nominal values and indicate the fact in the documentation. Additionally, (as noted in the assumptions), each model is run a mode presuming the military environment. This is done since SPANS runs only in a military acquisition mode.

IV. *Analysis and Findings*

This chapter presents the analysis of the data and the accompanying results. The first section examines the two data sets to determine their compatibility to each other. The purpose is to show, statistically and qualitatively, that the two data sets may be combined to form a larger statistical sample. The next sections introduce and present the analysis of the model estimates.

Data Analysis

Compatibility. The purpose of proving compatibility is to combine the two data sets into one, or show that both sample sets are members of the same population. "The eventual aim is to make statements that have some validity for the population at large" (22:231). Two reasons for this combination are 1) To create a larger sample for statistical purposes and 2) To simplify the presentation of the results by making inferences from the sample data to a single population. Compatibility is addressed in two ways; subjectively and statistically.

Subjective factors are general characteristics of the data. In the case of this research, both data sets include projects from the MIS arena (3:640-641, 18:419). The majority of the projects are written in COBOL, "the most widely used business data-processing language" (18:419). A third consideration is the size of the projects. As mentioned earlier, the average SLOC size of the Kemmerer data is much larger, but the average function point and effort size are closer. In both sets, the majority of the projects are medium-sized (20 - 250 KSLOC). Statistical analysis will test the hypothesis that the project sizes are compatible.

LSBF regression analysis is used to test the hypothesis of size compatibility. The ANOVA tables for the equations in this section are found in Appendix B. The power curve, or log(Y)-log(X) transformation, is chosen as the best equation by inspection of the test statistics and the residual plots results. All LSBF tests in the balance of this research will use this formulation.

The first test is for the equality of the variances of the data sets. The hypothesis being tested is that the variances are equal. Using the F_{calc} determined by Eq (10) and the F_{crit} from the tables:

$$F_{calc}=\frac{MSE_A}{MSE_K}=\frac{.3778}{.3047}=1.2399$$

$$F_{(22,13,.01)}=3.52$$

Since $F_{calc} \leq F_{crit}$, the hypothesis of equal variances is accepted.

The second compatibility test is the Full-reduced technique. The F_{calc} for this test is calculated using Eq (11). The calculation is run for each data set.

$$F_{calc_A}=\frac{SSE_A-SSE_F}{Df_A-Df_F}+\frac{SSE_F}{SSE_F}=\frac{(8.3116-14.0421)}{(22-37)}+\frac{(14.0421)}{37}=1.0066$$

$$F_{(15,37,.01)}=2.66$$

$$F_{calc_K}=\frac{(3.9608.3116-14.0421)}{(13-37)}+\frac{14.0421}{37}=1.1066$$

$$F_{(24,37,.01)}=2.42$$

$F_{calc} \leq F_{crit}$ in both cases, so the hypothesis, that the slopes of the full and reduced equations are equal, is accepted. The acceptance of the hypotheses of equal variance and slope confirms the subjective analysis results the two data sets are from the same population and may be combined for further study.

Outliers. This section of the research determines the status of any outliers in the full data set. The combination of the two sets results in a large sample, large enough so that the possible removal of outlying data points may not adversely affect the regression equation. The full set of data in this study is examined graphically and statistically. However, data points cannot be rejected simply on visual or statistical evidence unless there is proof of circumstances that make the point an exceptional case (21:406). Thus, any point that looks to be an outlier must also show properties that separate it from the other data.

The first step is to look at the data in a graphical format. The following graph shows the data plotted in logarithmic values to correspond with the log-log transformation chosen as the best equation:

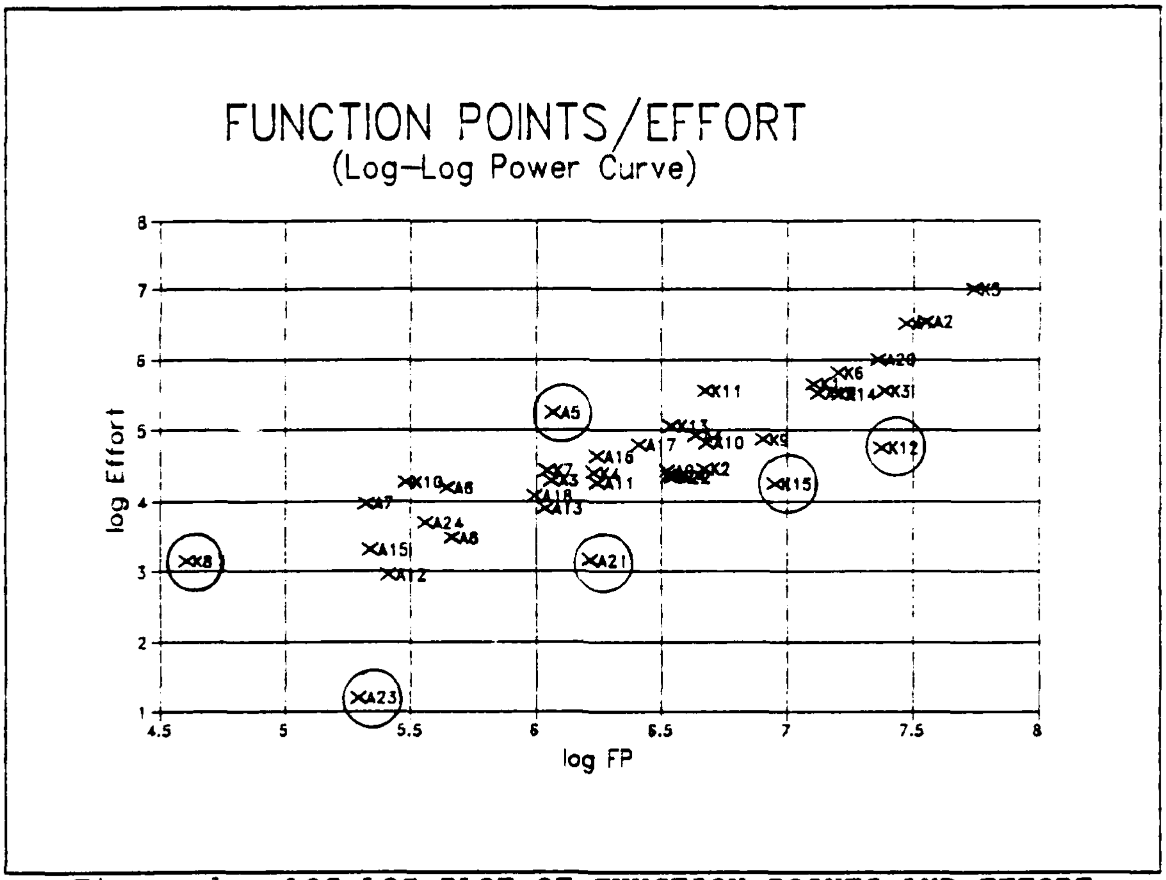


Figure 1. LOG-LOG PLOT OF FUNCTION POINTS AND EFFORT

A5, A21, A23, K8, K12 and K15 (in circles) appear to be candidates for further inspection. This is accomplished by examining the statistics listed in Appendix C. These include the studentized deleted residuals (e_i, normalized prediction error obtained by removing each data point in turn) and the Cook's distance measure. The Cook's distance measure (d_i) indicates the influence the deletion of the observation has on the regression equation (21:396-406).

The values found in the e_i residual column follow the Student-t distribution. Values that fall in the extreme .05 area of the distribution tail are candidates for removal. The cut-off value is $t_{(n-p-1,.05)}$, or $t_{(36,.05)}$ or 1.691. The points that exceed this value are the same as the visual inspection except for K8.

The final statistical treatment for outliers is the Cook's d_i influence test. The cut-off value is from the *F* distribution for $F_{(p,n-p,.5)}$ or .707. This value exceeds all values in the set. The largest d_i, A23, is still significantly less than the cut-off. This shows that no single observation has a large influence on the regression equation.

Five observations are identified as candidates for deletion as outliers: A5, A21, A23, K12 and K15. Before any of these points can be deleted, however, further non-empirical reasons are needed to confirm that these points are outliers. Project A21 and A23 are small, less than 20 KSLOC, compared to the other projects. Additionally, A23 is written in language (IBM/DMS) other than COBOL which differentiates it from over three-fourths of the data. K15 is also written in a language other than COBOL and is the only project using that language (NATURAL). These circumstances, in conjunction with the graphical and statistical evidence, lead to a conclusion that these observations may be deleted. There is no other evidence supporting the deletion of A5 and K12, so these observations remain in the data set.

Prediction Analysis

The following three sections explain the relationships between the predicted values derived from the estimation models (X) and the actuals for each project (Y). (Note that the data set contains only thirty-six projects after three projects were identified as outliers.) Each section details the respective results of three statistical techniques; LSBF regression, the Wilcoxin T and percent error.

LSBF Regression. This technique is used to test the hypothesis that there is a linear relationship between X and Y (21:146). If a significant relationship does exist, F_{calc} will be greater than F_{crit}. The confidence level for this test is α=.01. Once again, the log-log, or power curve, transformation of the LSBF equation is utilized:

$$Y=A*X^{b_1}$$

or

$$\log(Y)=\log(A)+b_1*\log(X)$$

To derive the correct results (Y) using the log-log equation (21), all calculations should be made in logarithmic form The transformation to Cartesian space is performed only on the log(Y) term. ANOVA tables containging the equation coefficients and other statistics used in this section are found in Appendix D.

The F_{crit} for each model is $F_{(1,34,.01)}$, or 7.48. The predictions of the R^2 and F_{calc} for each model are as follows:

TABLE 5.

Statistical Summary of Model Prediction Reliability

	R^2	F_{calc}
SPANS	.81	145.26
Checkpoint	.79	129.19
Costar	.80	138.93

Only Checkpoint does not exceed the cut-off value of .80 (the level set for acceptable reliability for this research) for the R^2, but Checkpoint's R^2 is very close to the criteria and not significantly different than those for SPANS and Costar. Each F_{calc} overwhelmingly exceeds the critical value with no significant difference between the models.

The LSBF tests on the prediction results show each model just meeting the reliability criteria set forth for this research. No models' reliability is demonstrably better for this data. Also, each model equation showed very strong relationship (F test) between predictions and actuals for this data set, but again, no significant difference was found among the three models' statistics.

Wilcoxin T. The tables containing the details of the Wilcoxin test for each model are in Appendix E. The hypothesis for this test is that the ranked differences between predictions and actuals are evenly distributed around zero. Since the number of observations in the data set can be considered large, the test statistic is derived using Eq. (12) from the methodology.

The cutoff value from the student t table for $t_{(36,.01)}$ is 2.41. The T_{stat} for each model:

TABLE 6.

Wilcoxin T Statistics for Bias Analysis

	T_{stat}
SPANS	-2.4352
Checkpoint	-1.7753
Costar	-4.5247

All models have a negative statistic which indicates a bias toward high predictions for this data set. Checkpoint is the only model to reject the hypothesis of no bias in the direction of prediction error at the .99 level of confidence. However, the Checkpoint statistic is still relatively large and accepts bias at the .95 confidence level, $t_{(36,.05)} = 1.689$. This is evidence that significant bias does exist in the Checkpoint predictions. Costar demonstates a very significant bias at all levels of confidence. SPANS's t statistic displays bias at both confidence levels, but just so at the .99 level.

Percent Error. The tests in this section employ the mean percent error and magnitude of relative error (MRE) for each of the models' prediction. Eq (13) is used to derive the percent error, or raw error, for each prediction as related to the actuals. The MRE is the absolute value of the percent error. The data for this section is found in Appendix F.

The prediction means and standard deviations for each measure of error are compared first. The mean results are listed in Table 7:

TABLE 7.

Mean and Standard Deviation From Percent Error Test

	Raw Error		MRE	
	Mean	Std Dev	Mean	Std Dev
SPANS	.51	.63	.62	.53
Checkpoint	.27	.65	.46	.53
Costar	1.02	.87	1.05	.84

In both methods, Checkpoint has the smallest mean error; however, its advantage is less in for the mean MRE. This is due to the loss of offsetting negative errors present in raw error. Costar's and, to a lesser extent, SPANS's predictions are biased so high that the negative error has little effect on the mean raw error; thus, there is little difference between the mean raw error and the mean MRE for these two models.

The standard deviation of the raw error can be used to test the normality of the distribution. Sixty-eight percent of the errors are expected to fall within one standard deviation and ninety-five percent within two standard deviations in a normally distributed sample. Checkpoint has 29 of 36 (81%) predictions within one, and 35 of 36 (97%) within two standard deviations. SPANS and Costar fall short of expected, with 61% and 53% respectively within one standard deviation, or 86% and 81% respectively within two. While none of these distributions is right on the mark, they are not so far off that the assumption of a normal distribution can be rejected.

The final percent error treatment on the data set is to ascertain how many predictions fall within a set range. This is a common and simple way of comparing results that can be easily understood by decision makers. This is also a technique used by model builders to tout their products abilities. Two ranges are used for this research, twenty and thirty percent. For this data set, no model predicted even half of the projects within either range (see Table 8 on next page):

TABLE 8.

Model Predictions Within A Predescribed Range

	±30%	±20%
SPANS	14(.39)	9(.25)
Checkpoint	17(.47)	13(.36)
Costar	7(.19)	5(.14)

These poor results are not unexpected since the mean raw errors are all close to or greater than thirty percent. Further, since most of the predictions err on the high side, and are not centered around zero, even fewer predictions could be expected to be within the range criteria. The predictions are also shown to have a high bias by the Wilcoxin T test. To examine whether the bias affects the reliability of the predictions, a further test is administered for percent error with an adjusted data set.

This additional percent error test demonstrates the models' reliability once an attempt is made to remove the bias in prediction accuracy. An adjustment is made on the predictions by decrementing each models' predictions. The adjustments are made by dividing each prediction by the mean raw error for the respective model. Table 10, found on the following page contains the summary data for raw error and MRE for the adjusted data:

TABLE 9.

Summary of Adjusted Data Percent Error Test Results

	Mean Raw Error	MRE	Prediction ±30%	±20%
SPANS	.00	.34	18(.50)	12(.33)
Checkpoint	.09	.32	20(.56)	14(.39)
Costar	.00	.34	18(.50)	11(.31)

As expected, the mean raw error is reduced and, in fact, is zero in two cases. The magnitude of relative error (MRE) and the number of predictions meeting the test criteria also tend to equalize.

These results are supported by preceding tests. The LSBF regression analysis results showed the reliability of the models to be nearly equal. The Wilcoxin test then demonstrated varied levels of bias among the models. The presence of bias was confirmed by divergent percentage errors. Once bias was removed (by adjusting the data in the final percent error test) the results became more uniform.

The success rate for predicting within the two ranges is still mediocre. However, these results can be compared to Kemmerer's study that used uncalibrated models to predict effort. Using Albrecht's function point equation, Kemmerer obtained predictions with an MRE of 102% (18:424). The initial results in this research yielded average MRE's ranging from 46% to 105%. Once the data was adjusted for known bias, the mean MRE's are 32% to 34%. This decrease shows an improvement in prediction ability and supports Kemmerer's conclusion that calibration of general models can lead to better estimation.

V. *Conclusions and Recommendations*

Conclusions

This research is driven by two onerous realities of software development in today's Air Force: 1. The steadily growing reliance on software systems, and 2. The continuing struggle to reliably estimate the costs of developing these systems. In 1979, IBM's Alan Albrecht introduced a new methodology for estimating software development effort for business-based systems. His function point analysis was an evolutionary step in software metrics. Over the next decade private industry and some foreign governments have embraced function points as a key estimation and productivity tool.

Agencies of the U.S. government, including the Department of Defense, have been slow to incorporate Albrecht's methodology. The Standard Systems Center (SSC), Gunter AFB, has turned to function points as a productivity improvement tool. This is a logical step due to SSC's emphasis on business-type, data processing systems and communication systems. The SSC commissioned Tecelote Research to develop an estimation and scheduling tool incorporating function points, SPANS.

The focus of this thesis was to test the reliability of SPANS's estimating capabilities and to compare the results to other function point analysis tools available to the Air Force. In addition to SPANS, Checkpoint (SPR, Inc.) and

Costar (Softstar Systems) were chosen for this study. The models estimated man-month effort for 36 software development projects from a database of combined projects from the research of Albrecht and Chris Kemmerer of the Sloan School, Massachusetts Institute of Technology.

Three statistical treatments were used to test the model estimates: LSBF regression analysis, Wilcoxin T and percentage error. The first tested the reliability of the models' prediction abilities. This test showed a strongly significant relationship between the estimates and actuals for each model. The reliability, or R^2, for the three models centered tightly around .80 with no significant distinction between the three.

The second treatment was the use of the Wilcoxin T to test for bias in model predictions for the data set. All models showed a significant, although varied, bias for high estimates. Checkpoint had the lowest Wilcoxin score, but still showed bias at the .95 confidence level. Costar had a very high Wilcoxin score, and SPANS scored between the two. It is probable that the bias was introduced by estimating sample projects gathered from private sector industry with tools configured for the military procurement environment. This research did not explore that probability, the hypothesis is good candidate for further study.

The high bias found in the Wilcoxin test was verified by the poor results of the models in the percent error tests. Checkpoint had the lowest mean raw percent error and absolute error at 21% and 46% respectively. SPANS was close behind with mean errors of 51% and 62%. Both means for Costar exceeded 100%. The performance of the models in predicting effort within preset percentage ranges was also poor. The most proficient, Checkpoint, missed the mark in just over half of its estimates in the ±30% range, and missed almost two-thirds in the ±20% range. Costar was the lowest with less than 20% success in both ranges. Again, SPANS performed between the two, but closer to the Checkpoint results. These result was not so surprising after the extreme bias scores encountered with the Wilcoxin test.

A final percent error test was run with adjusted prediction values to attempt to counteract the bias. Each set of model estimates were divided by their mean raw errors and compared to the actuals, again. The results showed predictable improvement in the mean error. The mean error for SPANS and Costar with the adjusted value went to zero, and Checkpoint's mean error was 10%. The ability to predict within the preset ranges followed the trend of the mean error. The Checkpoint results changed little and the other two models improved to the point that no significant difference was found among the three.

The first research question posed at the beginning of the thesis dealt with the reliability of SPANS to accurately predict software development effort. The conclusion is strongly, but not overwhelmingly, positive. SPANS showed the capability to reliably estimate over the range of data in this study. A caveat is the bias toward high estimates shown in the Wilcoxin and percentage tests. However, if the bias is a known quantity, the analysts can adjust for it as long the variance in the predictions remains consistent.

The second research question proposed a comparison of the estimating abilities of SPANS to other models. For the most part, the comparisons, when controlled for bias, showed little difference among between the three models. Checkpoint and SPANS performed only slightly better, although more consistently, than Costar. Costar also demonstrated a much higher sensitivity to bias.

Recommendations

A recommendation from this study for model selection is difficult for two reasons: First, the differences in the results between the three model, while quantifiable, were not significant. Second, each model has peculiar scheduling and management analysis tools, not examined here, that might be useful to project members for different circumstances.

Several recommendations can be made for further research, however. A direct follow-on to this study is to obtain a data set of function point counts for software development projects managed by and for the Air Force. Ideally, a data set with *a priori* estimates and *a posteori* actuals is best. This would test not only the models' capabilities, but also the ability of analysts to estimate requirements and efforts at the start of a program.

A related issue is the measure of productivity with function points. This was the original intended use of the methodology when introduced by Albrecht. Tracking a project's progress through the completion of function points could be more desirable and informative than amassing source code counts. However, there is little research supporting this hypothesis.

Finally, function point analysis does not appear to be useful for a large segment of Air Force software development, real-time or weapon systems. Variants of function points, called feature points or real-time function points, are considered by some experts to be an effective estimating method for these systems, but have received little or no treatment in DOD studies. Studying these could conceivably be very beneficial for analysts in the weapon systems arena.

Appendix A: Function Point Count Sheet

Function Point Calculation

Summary

Application Name		Prepared by	MM/DD/YY
		W.W.	10-Dec-90
Project ID	Project Name	Reviewed by	MM/DD/YY

Notes:

UNADJUSTED FUNCTION POINTS

Type ID	Component	Low		Average		High		Total
EI	External Input	14 x 3 =	42	63 x 4 =	252	239 x 6 =	1434	1728
EQ	External Inquiry	115 x 3 =	345	40 x 4 =	160	34 x 6 =	204	709
EO	External Output	23 x 4 =	92	104 x 5 =	520	155 x 7 =	1085	1697
ILF	Internal Logical File	41 x 7 =	287	1 x 10 =	10	– x 15 =	–	297
EIF	Ext Interface File	1 x 5 =	5	– x 7 =	–	– x 10 =	–	5

Level of information processing function

Total Unadjusted Function Points 4436

GENERAL SYSTEM CHARACTERISTICS

ID	General System Characteristic	Rating	ID	General System Characteristic	Rating
C1	Data Communications	4	C8	On-Line update	3
C2	Distributed Functions	2	C9	Complex Processing	2
C3	Performance	3	C10	Reusability	2
C4	Heavily Used Configuration	2	C11	Installation Ease	2
C5	Transaction Rate	3	C12	Operational Ease	3
C6	On-line Data Entry	5	C13	Multiple Sites	2
C7	End User Efficiency	3	C14	Facilitate Change	2
				Total Rating	39

Value Adjustment Factor= Total Rating x .01 + .65 1.04

Total Function Points = Unadjusted Function Points x Value Adjustment Factor

= 4436 x 1.04 = 4613

001.03 12/89

Appendix B: LSBF Analysis for Data Compatibility Tests

Albrecht Input Data

PROJECT	KSLOC	FP	ACTUAL EFFORT	LOG FP	LOG EFFORT
A1	130	1750	673.7	7.46737	6.51278
A2	318	1902	692.1	7.55066	6.53973
A3	20	428	73.0	6.05912	4.29046
A4	54	759	138.8	6.63200	4.93303
A5	62	431	189.5	6.06611	5.24439
A6	28	283	65.8	5.64545	4.18662
A7	35	205	52.6	5.32301	3.96272
A8	30	289	32.2	5.66643	3.47197
A9	48	680	84.9	6.52209	4.44147
A10	93	794	125.0	6.67708	4.82831
A11	57	512	71.1	6.23832	4.26409
A12	22	224	19.1	5.41165	2.94969
A13	24	417	49.3	6.03309	3.89792
A14	42	682	78.9	6.52503	4.36818
A15	40	209	27.0	5.34233	3.29584
A16	96	512	103.9	6.23832	4.64343
A17	40	606	120.4	6.40688	4.79082
A18	52	400	58.6	5.99146	4.07073
A19	94	1235	250.7	7.11883	5.52426
A20	110	1572	402.6	7.36010	5.99794
A21	15	500	23.6	6.21461	3.16125
A22	24	694	77.6	6.54247	4.35157
A23	3	199	3.3	5.29330	1.19392
A24	29	260	40.1	5.56068	3.69138

Albrecht Linear Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	702863.51624	702863.51624	152.924	0.0001
Error	22	101115.84210	4596.17464		
C Total	23	803979.35833			

Root MSE	67.79509	R-square	0.8742
Dep Mean	143.90833	Adj R-sq	0.8685
C.V.	47.10991		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	-88.087178	23.31223476	-3.779	0.0010
FP	1	0.358225	0.02896802	12.366	0.0001

Albrecht Exponential Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	21.60027	21.60027	48.771	0.0001
Error	22	9.74356	0.44289		
C Total	23	31.34383			

Root MSE	0.66550	R-square	0.6891
Dep Mean	4.35885	Adj R-sq	0.6750
C.V.	15.26776		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	3.072757	0.22884067	13.427	0.0001
FP	1	0.001986	0.00028436	6.984	0.0001

Albrecht Logarithmic(X) Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	524786.28281	524786.28281	41.352	0.0001
Error	22	279193.07552	12690.59434		
C Total	23	803979.35833			

Root MSE	112.65254	R-square	0.6527
Dep Mean	143.90833	Adj R-sq	0.6370
C.V.	78.28076		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	-1257.865903	219.19511675	-5.739	0.0001
LOGFP	1	224.453846	34.90412826	6.431	0.0001

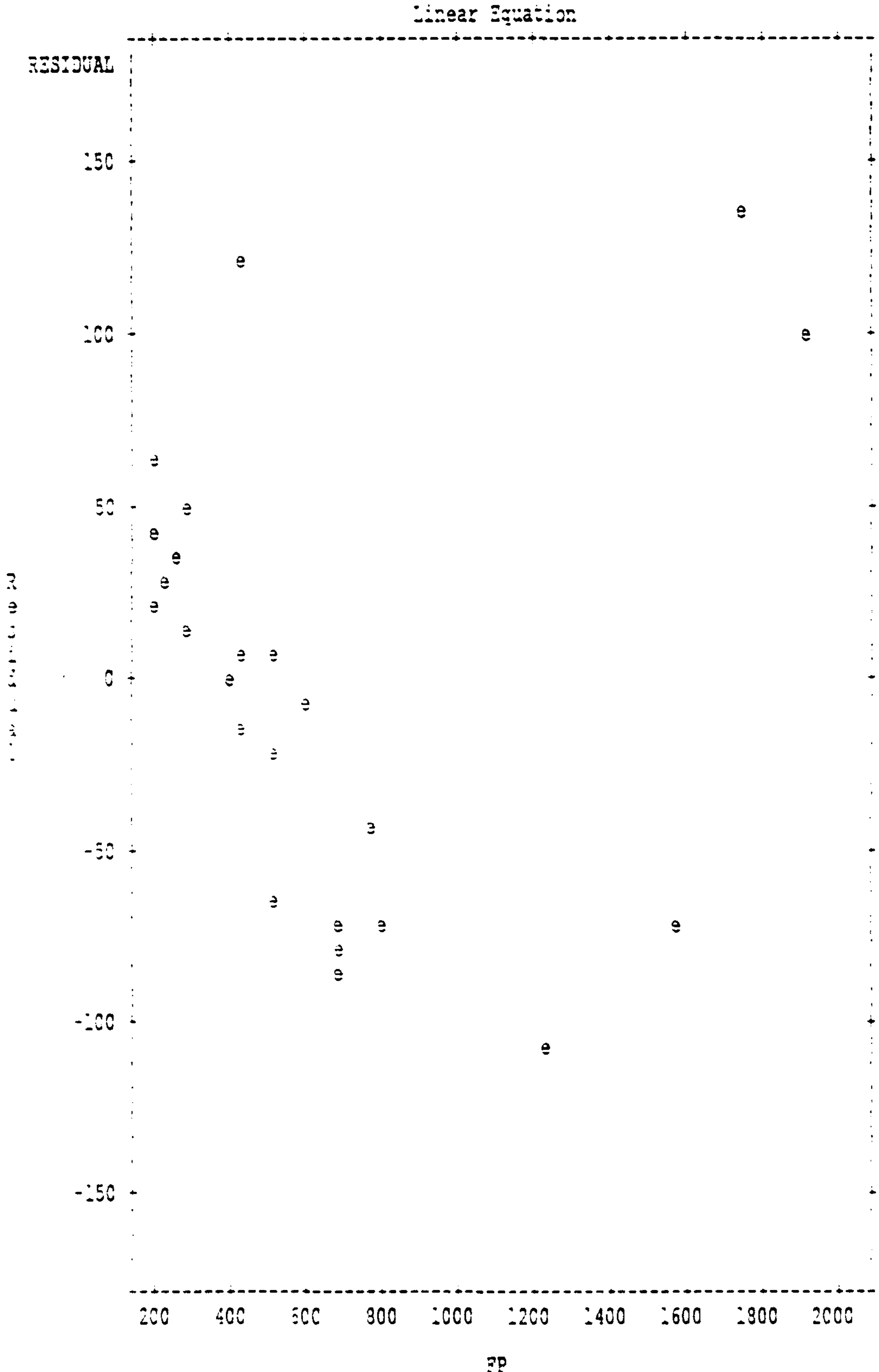
Albrecht Log(X)-log(Y) Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	23.03220	23.03220	60.964	0.0001
Error	22	8.31163	0.37780		
C Total	23	31.34383			

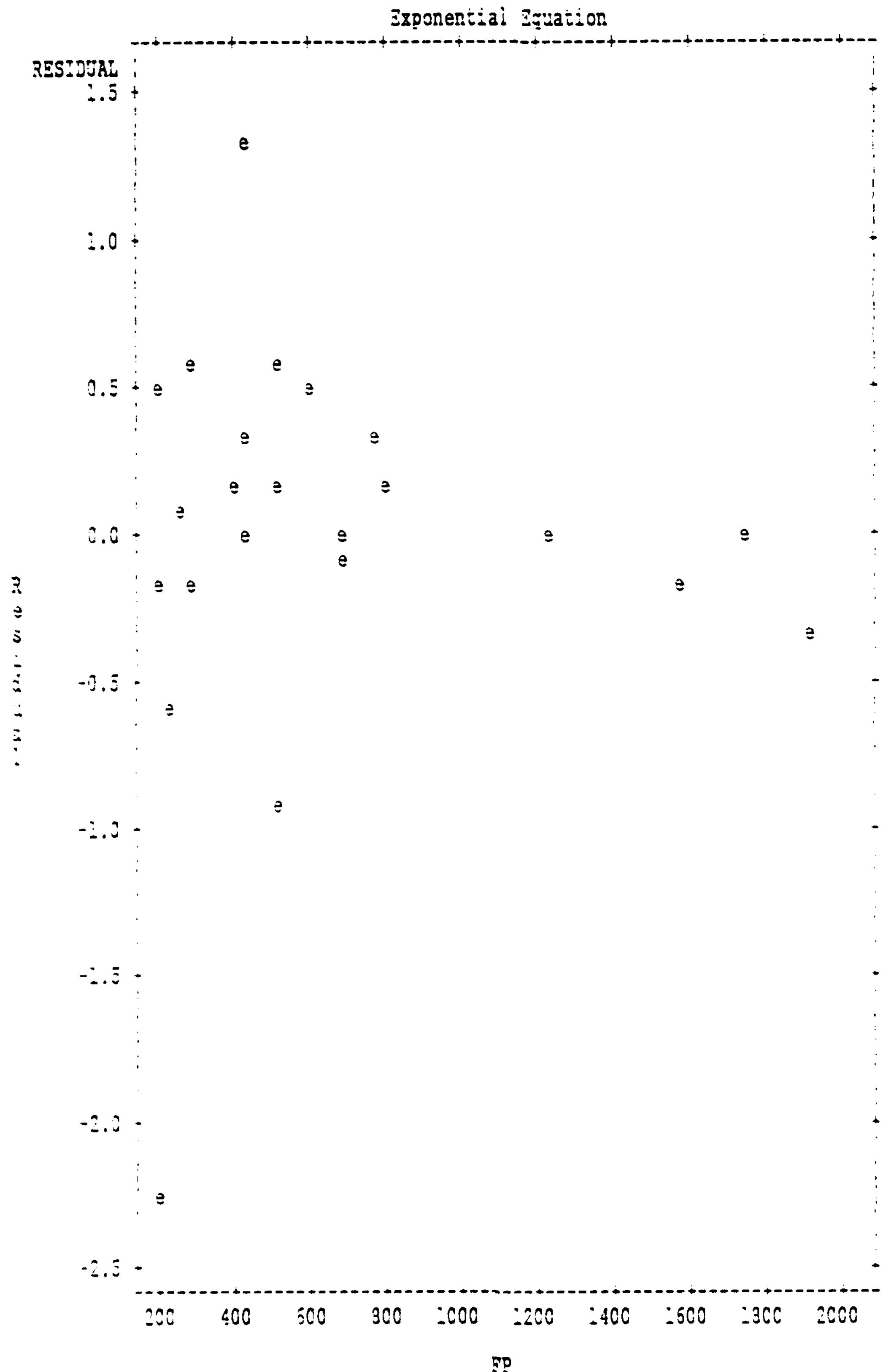
Root MSE	0.61466	R-square	0.7348
Dep Mean	4.35885	Adj R-sq	0.7228
C.V.	14.10131		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	-4.927700	1.19597356	-4.120	0.0004
LOGFP	1	1.486975	0.19044409	7.808	0.0001

Residual Plot for Albrecht Data

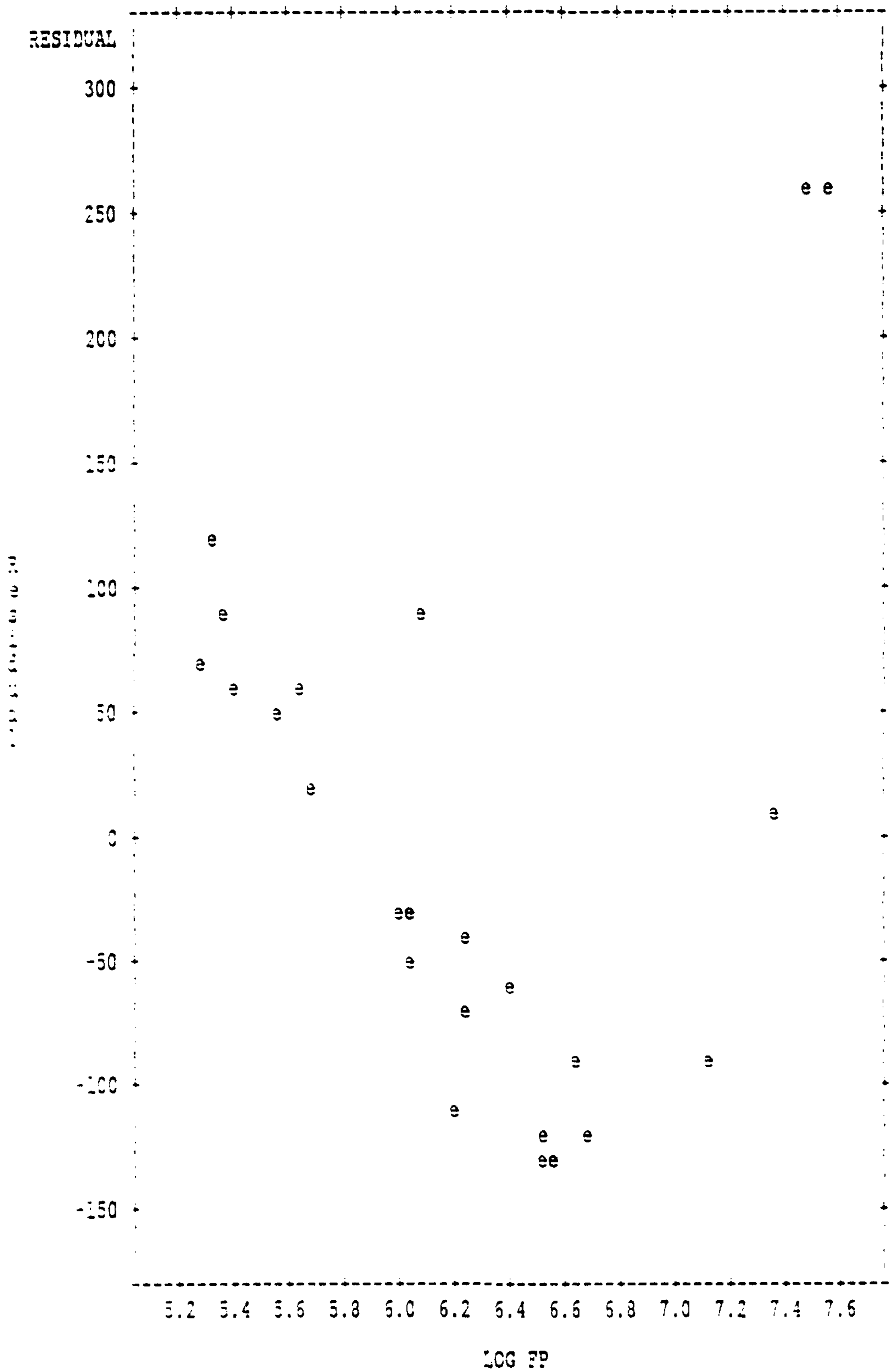


Residual Plot for Albrecht Data



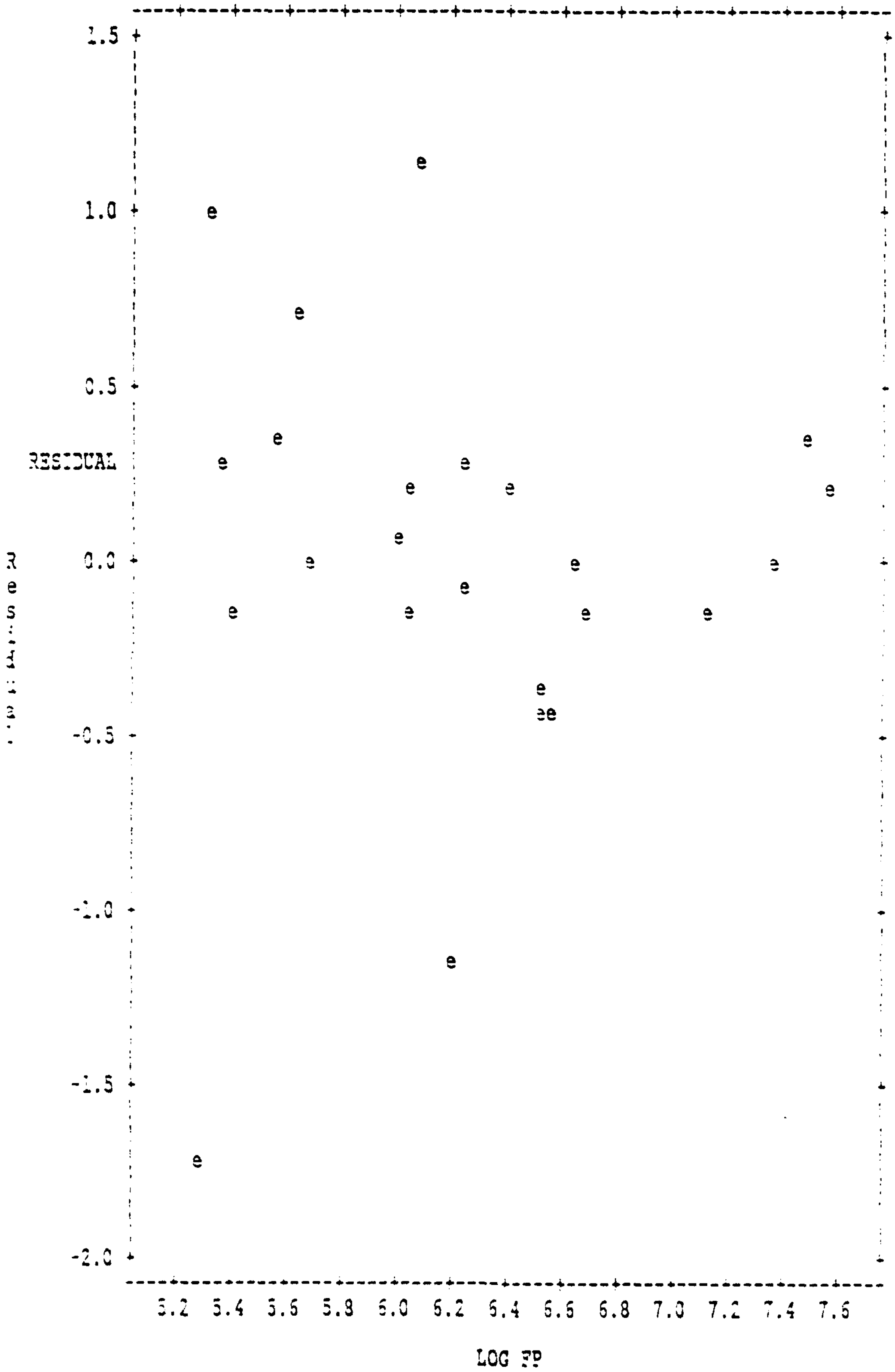
Residual Plot for Albrecht Data

Logarithmic(X) Equation



Residual Plot for Albrecht Data

Log(X)-log(Y) Equation



Kemmerer Input Data

PROJECT	KSLOC	FP	ACTUAL EFFORT	LOG FP	LOG EFFORT
K1	254	1217	287.0	7.10414	5.65948
K2	214	788	86.9	6.66950	4.46476
K3	254	1611	258.7	7.38461	5.55567
K4	41	507	82.5	6.22851	4.41280
K5	450	2307	1107.3	7.74370	7.00968
K6	450	1338	336.3	7.19893	5.81800
K7	50	421	84.0	6.04263	4.43082
K8	43	100	23.2	4.60517	3.14415
K9	200	993	130.3	6.90073	4.86984
K10	39	240	72.0	5.48064	4.27667
K11	129	789	258.7	6.67077	5.55567
K12	289	1593	116.0	7.37337	4.75359
K13	161	691	157.0	6.53814	5.05625
K14	165	1348	246.9	7.20638	5.50898
K15	60	1044	69.9	6.95081	4.24707

Kemmerer Linear Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	562314.67747	562314.67747	17.925	0.0010
Error	13	407814.89987	31370.37691		
C Total	14	970129.57733			

Root MSE	177.11685	R-square	0.5796
Dep Mean	221.11333	Adj R-sq	0.5473
C.V.	80.10229		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	-118.447116	92.32432111	-1.283	0.2219
FP	1	0.339855	0.08027195	4.234	0.0010

Kemmerer Exponential Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	7.90139	7.90139	27.877	0.0001
Error	13	3.68471	0.28344		
C Total	14	11.58610			

Root MSE	0.53239	R-square	0.6820
Dep Mean	4.98423	Adj R-sq	0.6575
C.V.	10.68150		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > ¦T¦
INTERCEPT	1	3.711372	0.27751495	13.374	0.0001
FP	1	0.001274	0.00024129	5.280	0.0001

Kemmerer Logarithmic(X) Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	304721.48573	304721.48573	5.953	0.0298
Error	13	665408.09160	51185.23782		
C Total	14	970129.57733			

Root MSE	226.24155	R-square	0.3141
Dep Mean	221.11333	Adj R-sq	0.2613
C.V.	102.31927		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > ¦T¦
INTERCEPT	1	-984.872387	497.70851167	-1.979	0.0694
LOGFP	1	180.720674	74.06766630	2.440	0.0298

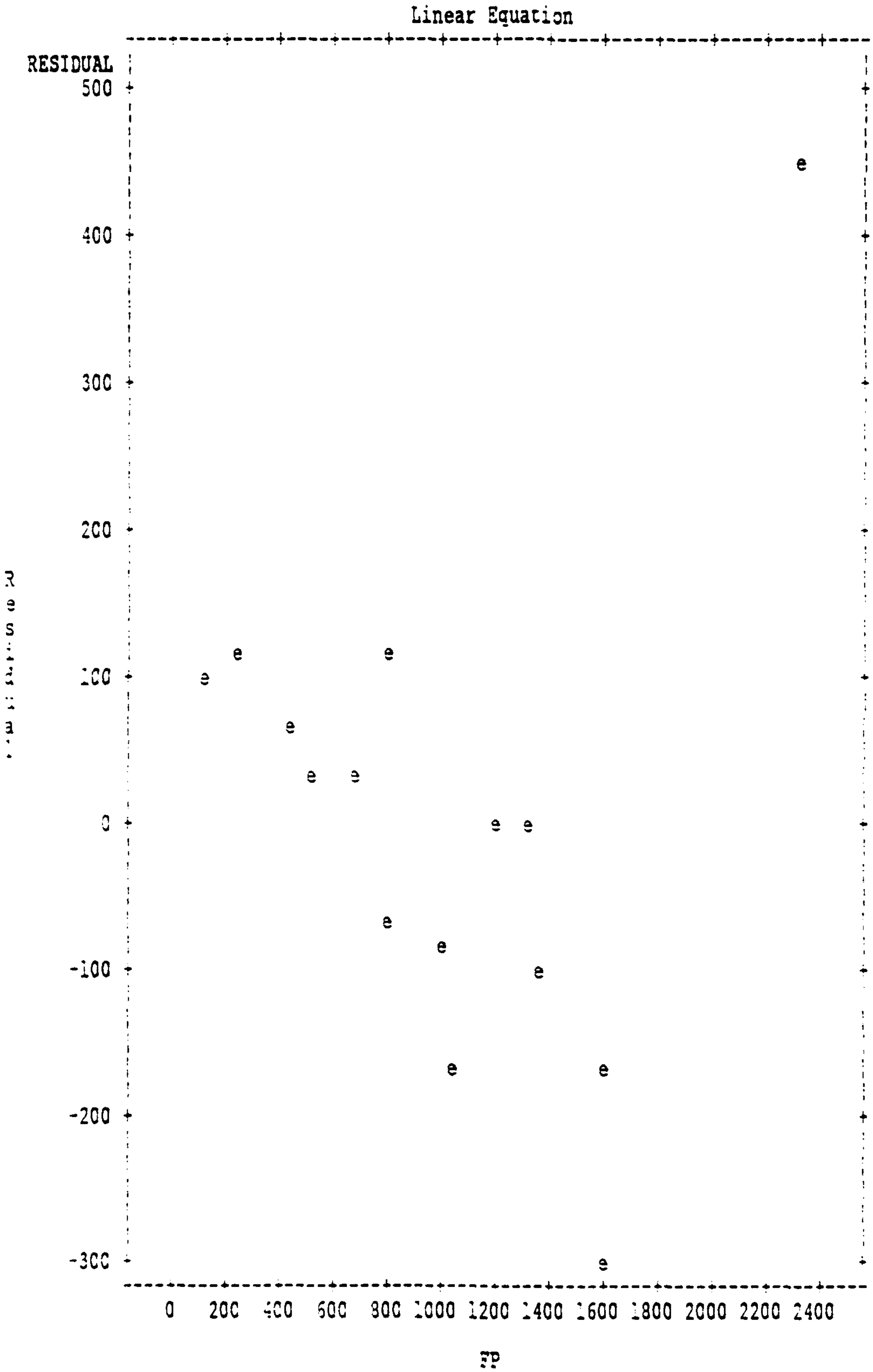
Kemmerer Log(X)-log(Y) Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	7.62533	7.62533	25.028	0.0002
Error	13	3.96077	0.30467		
C Total	14	11.58610			

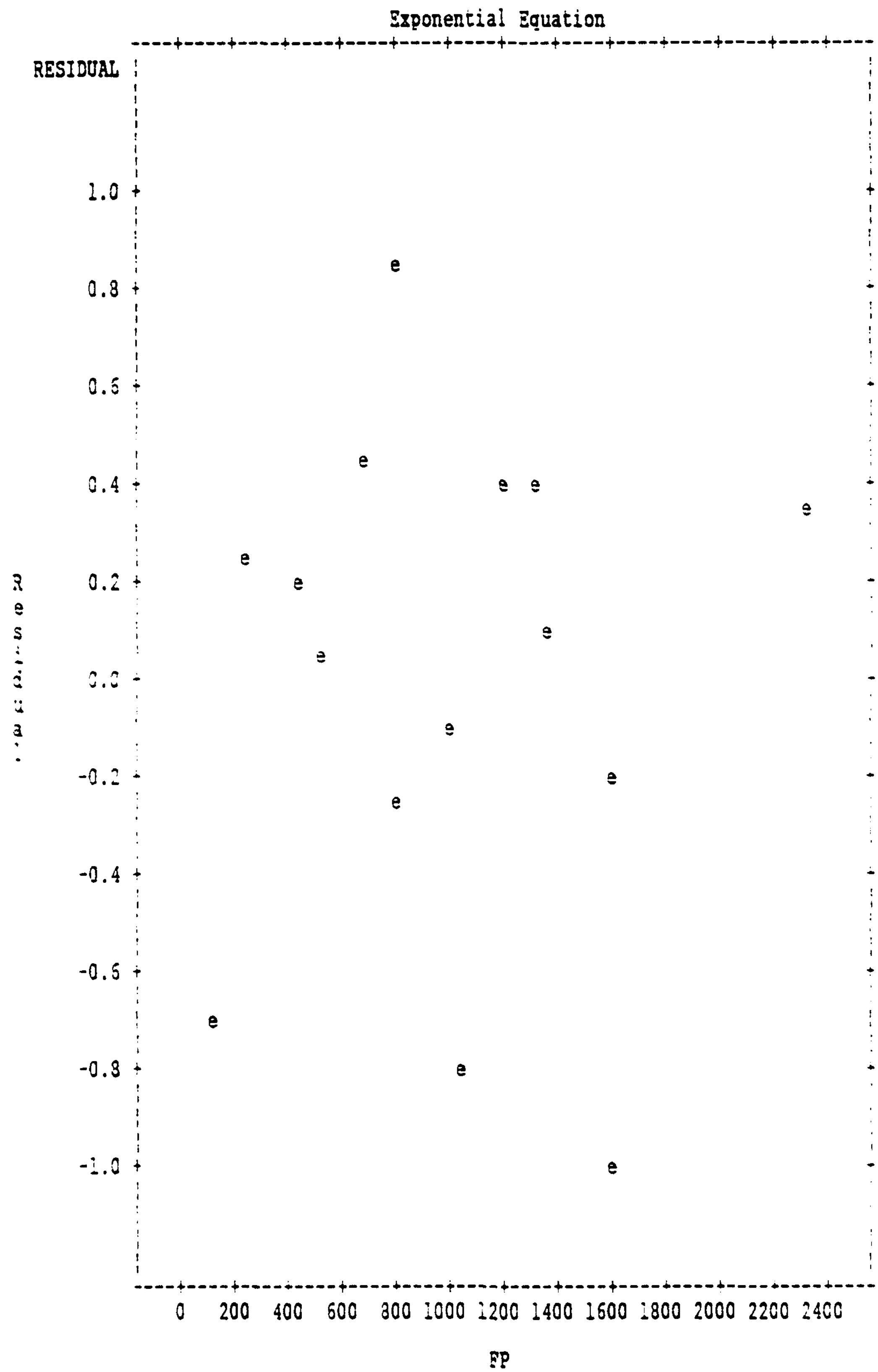
Root MSE	0.55197	R-square	0.6581
Dep Mean	4.98423	Adj R-sq	0.6318
C.V.	11.07440		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > ¦T¦
INTERCEPT	1	-1.048587	1.21428593	-0.864	0.4035
LOGFP	1	0.904036	0.18070682	5.003	0.0002

Residual Plot for Kemmerer Data

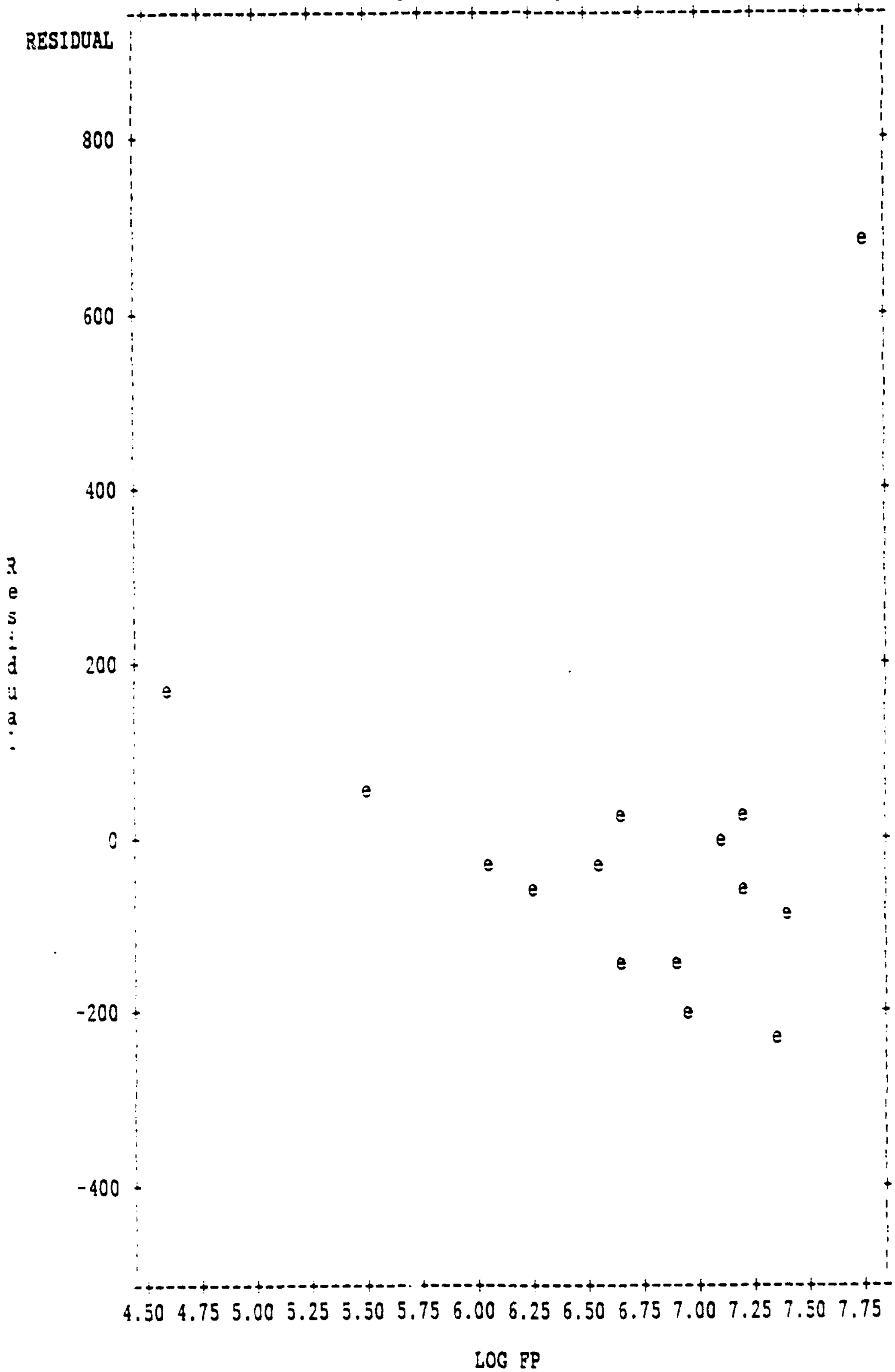


Residual Plot for Kemmerer Data



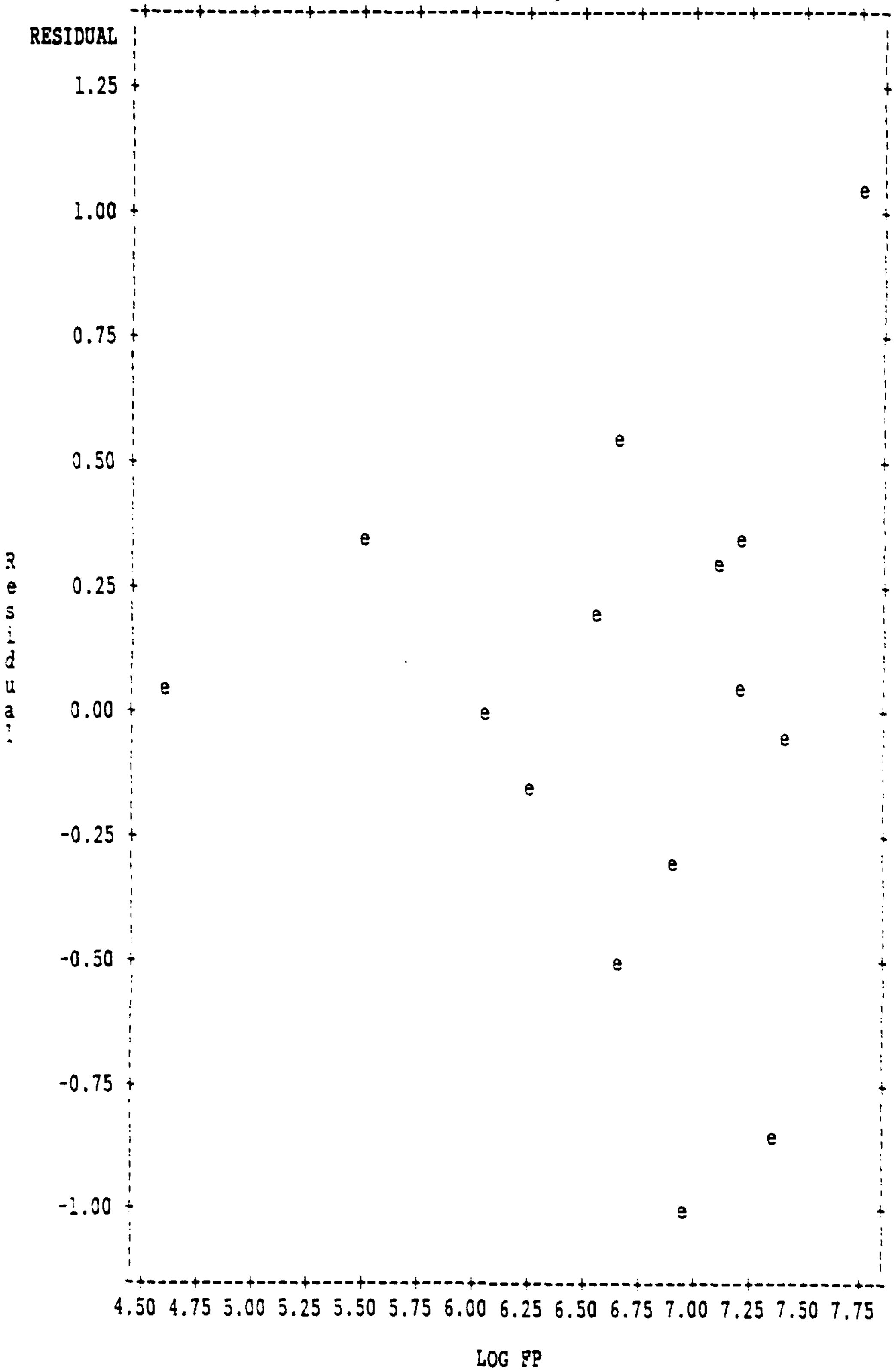
Residual Plot for Kemmerer Data

Logarithmic(X) Equation



Residual Plot for Kemmerer Data

Log(X)-log(Y) Equation



Combined Input Data

PROJECT	KSLOC	FP	ACTUAL EFFORT	LOG FP	LOG EFFORT
K1	254	1217	287.0	7.10414	5.65948
K2	214	788	86.9	6.66950	4.46476
K3	254	1611	258.7	7.38461	5.55567
K4	41	507	82.5	6.22851	4.41280
K5	450	2307	1107.3	7.74370	7.00968
K6	450	1338	336.3	7.19893	5.81800
K7	50	421	84.0	6.04263	4.43082
K8	43	100	23.2	4.60517	3.14415
K9	200	993	130.3	6.90073	4.86984
K10	39	240	72.0	5.48064	4.27667
K11	129	789	258.7	6.67077	5.55567
K12	289	1593	116.0	7.37337	4.75359
K13	161	691	157.0	6.53814	5.05625
K14	165	1348	246.9	7.20638	5.50898
K15	60	1044	69.9	6.95081	4.24707
A1	130	1750	673.7	7.46737	6.51278
A2	318	1902	692.1	7.55066	6.53973
A3	20	428	73.0	6.05912	4.29046
A4	54	759	138.8	6.63200	4.93303
A5	62	431	189.5	6.06611	5.24439
A6	28	283	65.8	5.64545	4.18662
A7	35	205	52.6	5.32301	3.96272
A8	30	289	32.2	5.66643	3.47197
A9	48	680	84.9	6.52209	4.44147
A10	93	794	125.0	6.67708	4.82831
A11	57	512	71.1	6.23832	4.26409
A12	22	224	19.1	5.41165	2.94969
A13	24	417	49.3	6.03309	3.89792
A14	42	682	78.9	6.52503	4.36818
A15	40	209	27.0	5.34233	3.29584
A16	96	512	103.9	6.23832	4.64343
A17	40	606	120.4	6.40688	4.79082
A18	52	400	58.6	5.99146	4.07073
A19	94	1235	250.7	7.11883	5.52426
A20	110	1572	402.6	7.36010	5.99794
A21	15	500	23.6	6.21461	3.16125
A22	24	694	77.6	6.54247	4.35157
A23	3	199	3.3	5.29330	1.19392
A24	29	260	40.1	5.56068	3.69138

Combined Linear Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	1301983.986	1301983.986	91.385	0.0001
Error	37	527145.98370	14247.18875		
C Total	38	1829129.9697			

Root MSE	119.36159	R-square	0.7118
Dep Mean	173.60256	Adj R-sq	0.7040
C.V.	68.75566		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	-89.955728	33.54732979	-2.681	0.0109
FP	1	0.336678	0.03521894	9.560	0.0001

Combined Exponential Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	31.78856	31.78856	79.733	0.0001
Error	37	14.75146	0.39869		
C Total	38	46.54002			

Root MSE	0.63142	R-square	0.6830
Dep Mean	4.59938	Adj R-sq	0.6745
C.V.	13.72831		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	3.297088	0.17746390	18.579	0.0001
FP	1	0.001664	0.00018631	8.929	0.0001

Combined Logarithmic(X) Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	874264.50942	874264.50942	33.877	0.0001
Error	37	954865.46032	25807.17460		
C Total	38	1829129.9697			

Root MSE	160.64612	R-square	0.4780
Dep Mean	173.60256	Adj R-sq	0.4639
C.V.	92.53672		

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	-1120.848893	223.88263647	-5.006	0.0001
LOGFP	1	201.946984	34.69654076	5.820	0.0001

Combined Log(X)-log(Y) Equation
Analysis of Variance

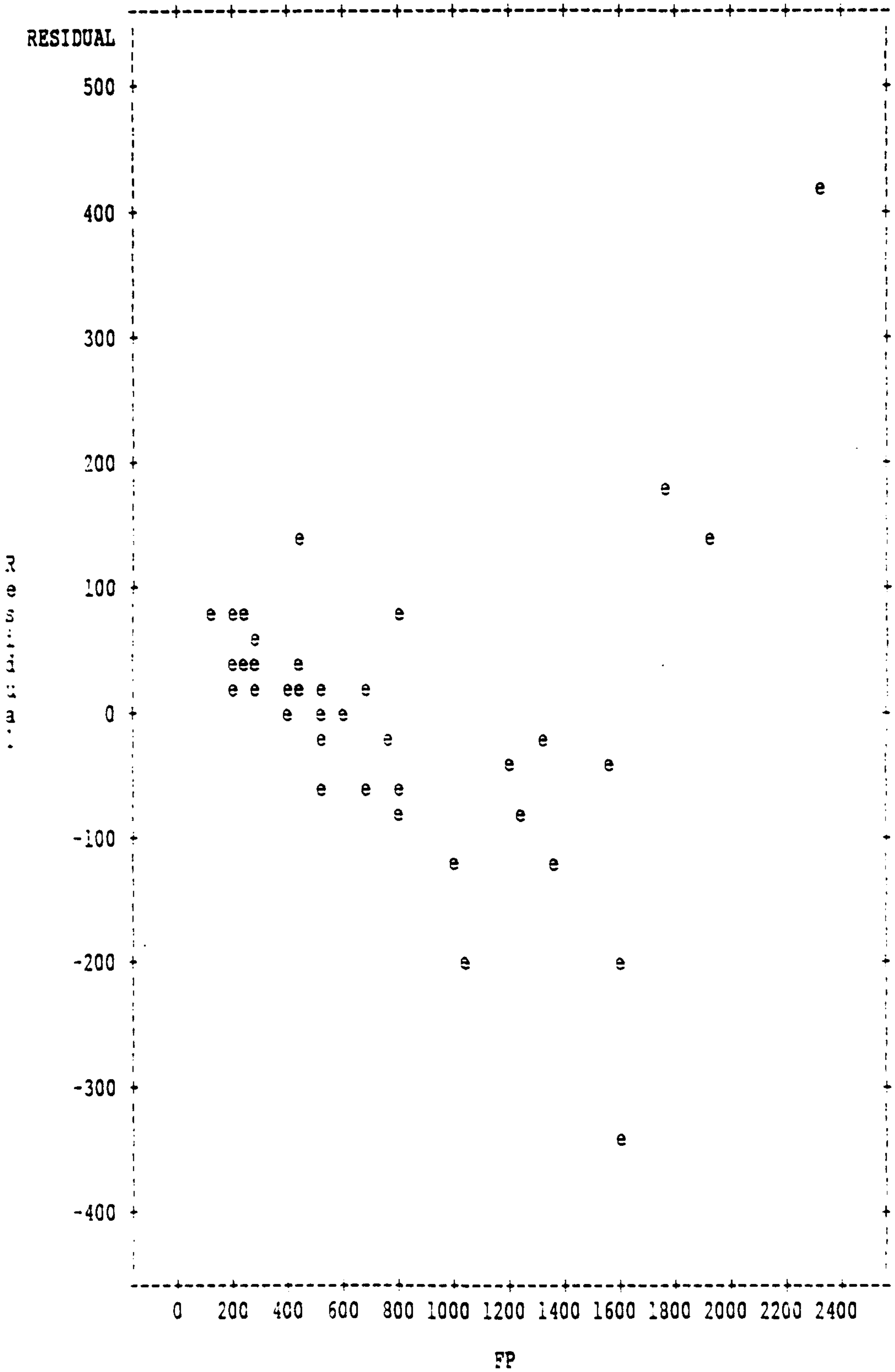
Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	32.49793	32.49793	85.630	0.0001
Error	37	14.04208	0.37952		
C Total	38	46.54002			

Root MSE	0.61605	R-square	0.6983
Dep Mean	4.59938	Adj R-sq	0.6901
C.V.	13.39416		

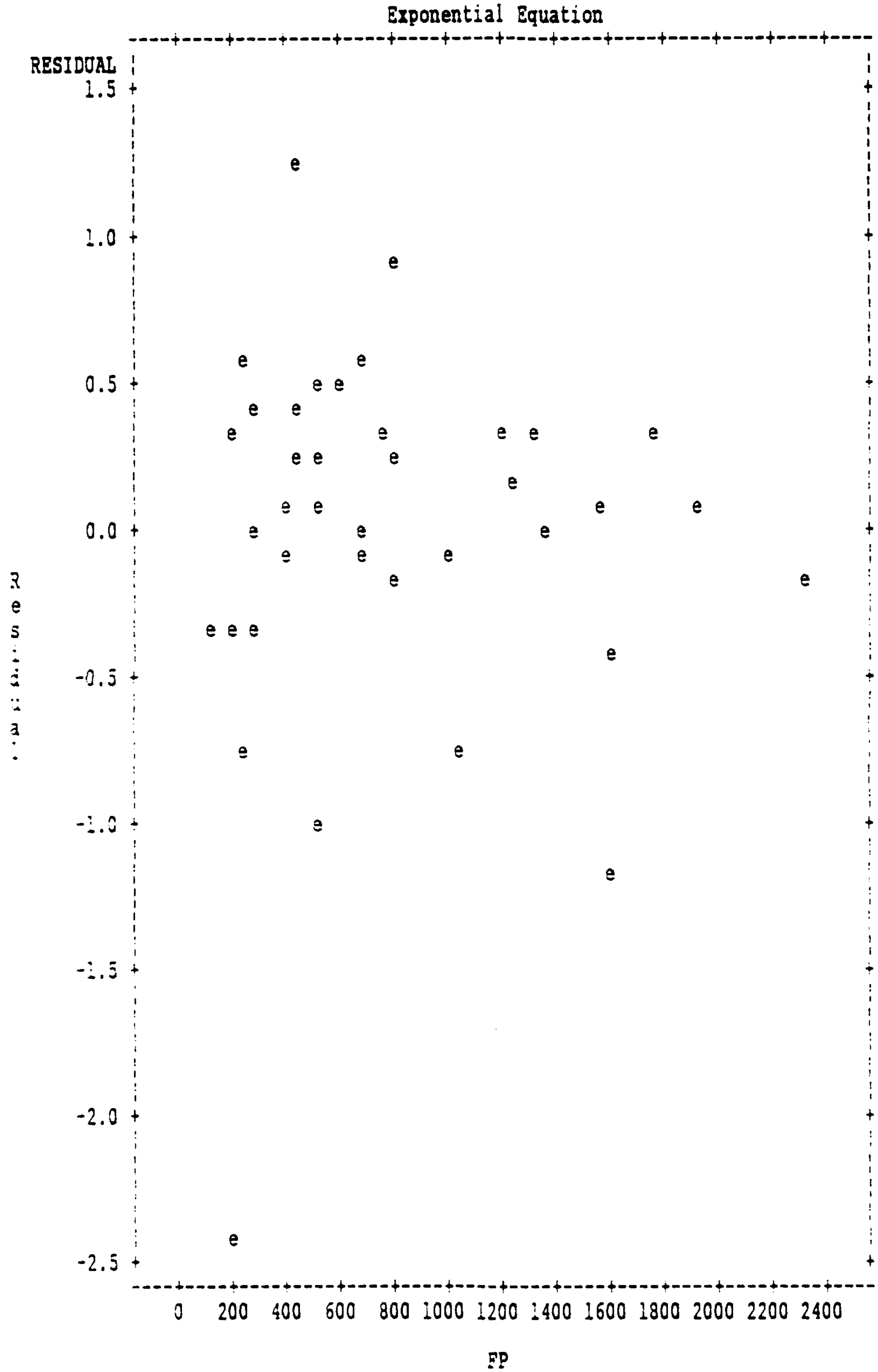
Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	-3.292711	0.85854904	-3.835	0.0005
LOGFP	1	1.231243	0.13305490	9.254	0.0001

Residual Plot for Combined Data

Linear Equation

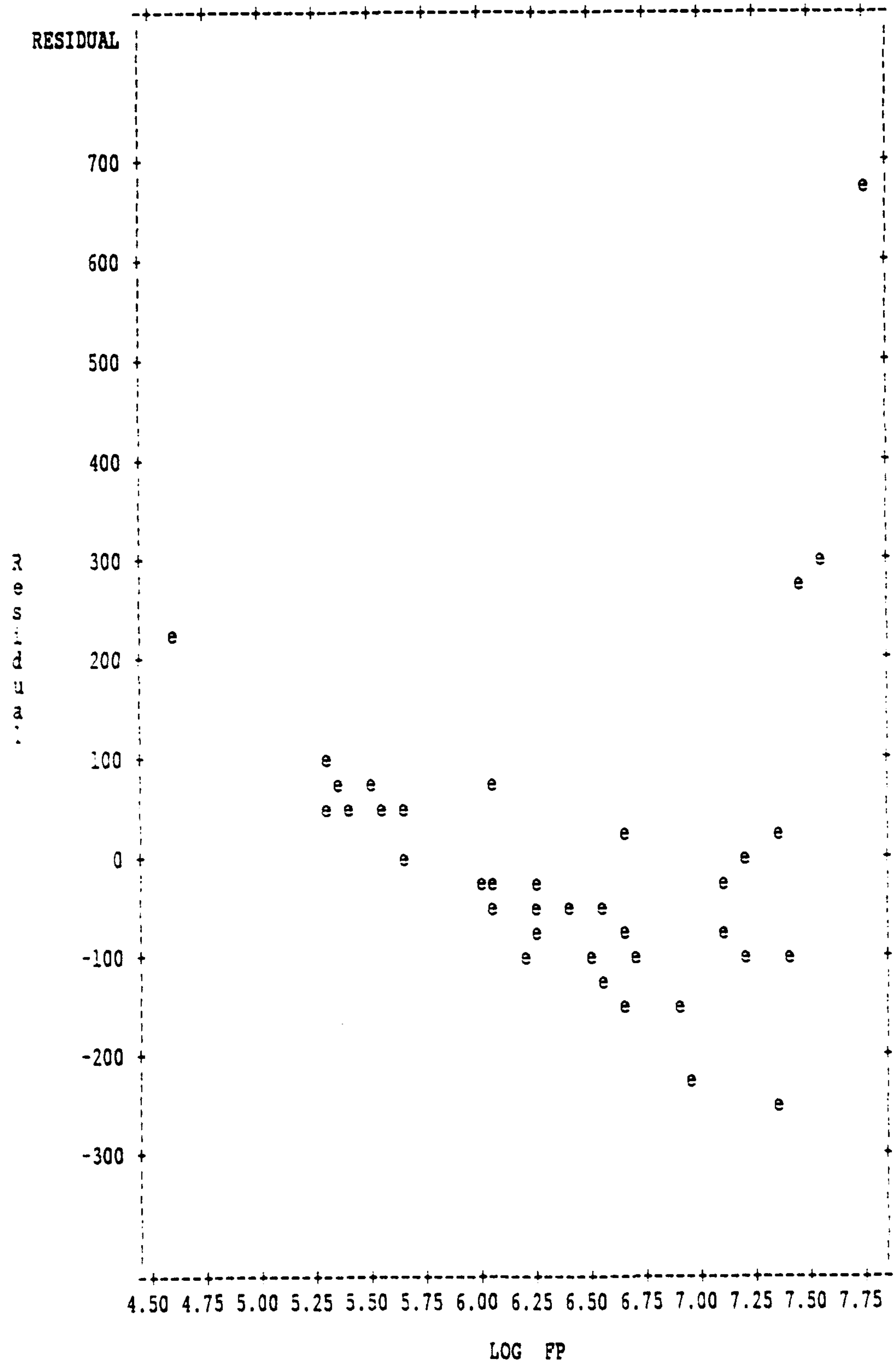


Residual Plot for Combined Data



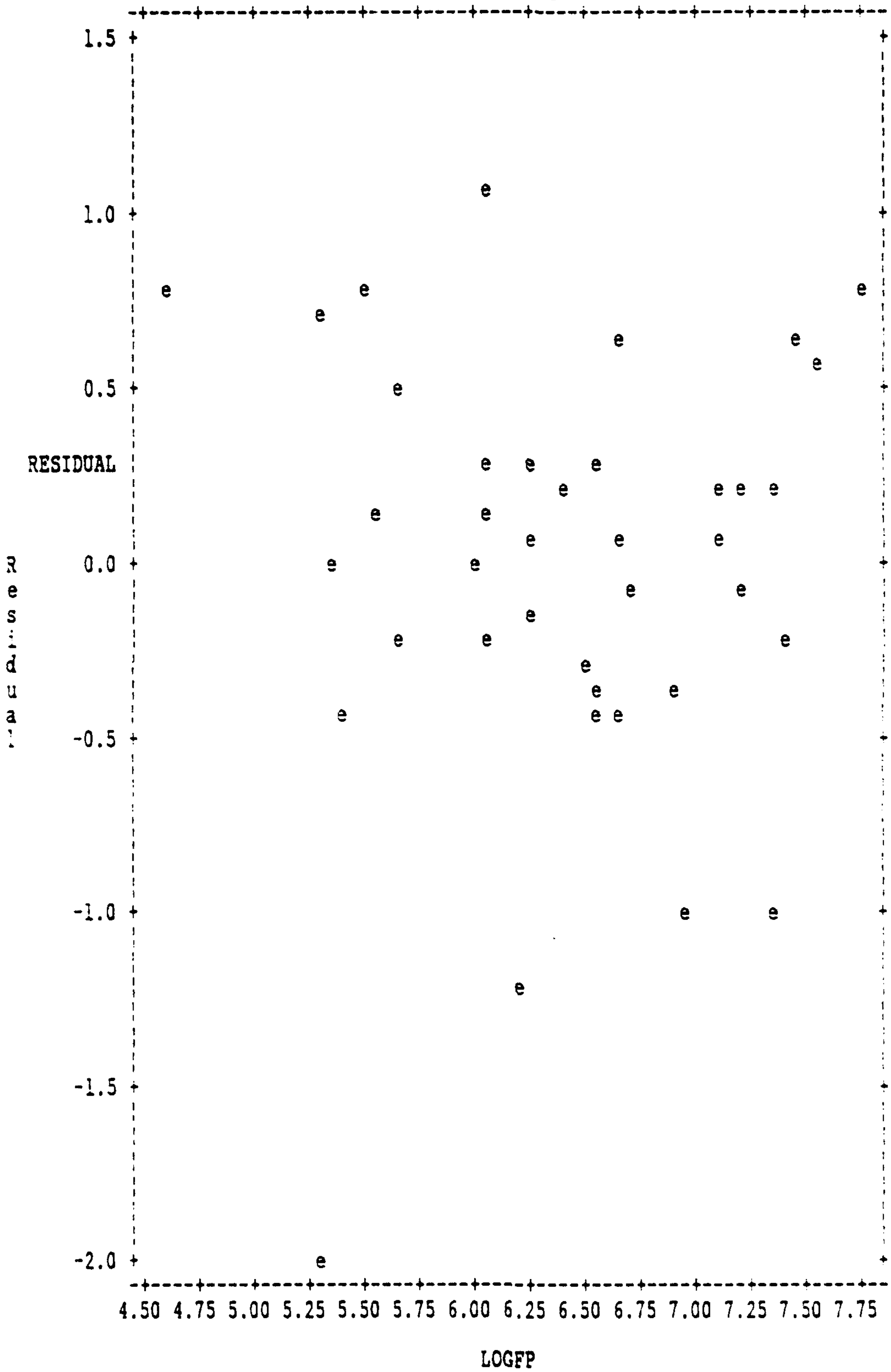
Residual Plot for Combined Data

Logarithmic(X) Equation



Residual Plot for Combined Data

Log(X)-log(Y) Equation



Appendix C: Outlier Analysis

Residual Values

Obs	Log Actual Effort	Predict Value	Standard Error Predict	Residual	Standard Error Residual	Student Residual
1	5.6595	5.4542	0.135	0.2053	0.601	0.342
2	4.4648	4.9191	0.105	-0.4543	0.607	-0.748
3	5.5557	5.7995	0.163	-0.2439	0.594	-0.410
4	4.4128	4.3761	0.102	0.0367	0.608	0.060
5	7.0097	6.2417	0.203	0.7680	0.582	1.320
6	5.8180	5.5709	0.144	0.2471	0.599	0.413
7	4.4308	4.1472	0.110	0.2836	0.606	0.468
8	3.1442	2.3774	0.260	0.7668	0.559	1.372
9	4.8698	5.2038	0.118	-0.3339	0.605	-0.552
10	4.2767	3.4553	0.158	0.8214	0.595	1.380
11	5.5557	4.9206	0.105	0.6350	0.607	1.046
12	4.7536	5.7857	0.162	-1.0321	0.594	-1.736
13	5.0562	4.7573	0.100	0.2989	0.608	0.492
14	5.5090	5.5801	0.145	-0.0711	0.599	-0.119
15	4.2471	5.2654	0.122	-1.0184	0.604	-1.687
16	6.5128	5.9014	0.172	0.6113	0.592	1.033
17	6.5397	6.0040	0.181	0.5357	0.589	0.910
18	4.2905	4.1675	0.109	0.1229	0.606	0.203
19	4.9330	4.8729	0.103	0.0601	0.607	0.099
20	5.2444	4.1761	0.109	1.0682	0.606	1.762
21	4.1866	3.6582	0.142	0.5284	0.600	0.881
22	3.9627	3.2612	0.175	0.7015	0.591	1.188
23	3.4720	3.6840	0.140	-0.2121	0.600	-0.353
24	4.4415	4.7376	0.100	-0.2961	0.608	-0.487
25	4.8283	4.9284	0.105	-0.1001	0.607	-0.165
26	4.2641	4.3882	0.101	-0.1241	0.608	-0.204
27	2.9497	3.3703	0.165	-0.4207	0.593	-0.709
28	3.8979	4.1355	0.111	-0.2376	0.606	-0.392
29	4.3682	4.7412	0.100	-0.3730	0.608	-0.614
30	3.2958	3.2850	0.173	0.0108	0.591	0.018
31	4.6434	4.3882	0.101	0.2552	0.608	0.420
32	4.7908	4.5957	0.099	0.1951	0.608	0.321
33	4.0707	4.0842	0.113	-0.0135	0.606	-0.022
34	5.5243	5.4723	0.136	0.0520	0.601	0.086
35	5.9979	5.7694	0.160	0.2286	0.595	0.384
36	3.1612	4.3590	0.102	-1.1977	0.608	-1.971
37	4.3516	4.7627	0.100	-0.4111	0.608	-0.676
38	1.1939	3.2246	0.178	-2.0307	0.590	-3.444
39	3.6914	3.5538	0.150	0.1375	0.598	0.230

Cook's d_i Measure of Influence

```
                               Cook's
 Obs   -2-1-0 1 2               d_i

   1   |      |      |         0.003
   2   |     *|      |         0.008
   3   |      |      |         0.006
   4   |      |      |         0.000
   5   |      |**    |         0.106
   6   |      |      |         0.005
   7   |      |      |         0.004
   8   |      |**    |         0.203
   9   |     *|      |         0.006
  10   |      |**    |         0.067
  11   |      |**    |         0.016
  12   |   ***|      |         0.112
  13   |      |      |         0.003
  14   |      |      |         0.000
  15   |   ***|      |         0.058
  16   |      |**    |         0.045
  17   |      |*     |         0.039
  18   |      |      |         0.001
  19   |      |      |         0.000
  20   |      |***   |         0.050
  21   |      |*     |         0.022
  22   |      |**    |         0.062
  23   |      |      |         0.003
  24   |      |      |         0.003
  25   |      |      |         0.000
  26   |      |      |         0.001
  27   |     *|      |         0.020
  28   |      |      |         0.003
  29   |     *|      |         0.005
  30   |      |      |         0.000
  31   |      |      |         0.002
  32   |      |      |         0.001
  33   |      |      |         0.000
  34   |      |      |         0.000
  35   |      |      |         0.005
  36   |   ***|      |         0.055
  37   |     *|      |         0.006
  38   |******|      |         0.542
  39   |      |      |         0.002

Sum of Residuals                      0
Sum of Squared Residuals        14.0421
Predicted Resid SS (Press)      16.1704
```

Appendix D: LSBF Analysis for Model Comparison

Model Comparison Input Data

PROJECT	KSLOC	FP	EFFORT	SPANS	CHECK	COSTAR	LOGSPANS	LOGCHECK	LOGCOSTAR	LOGEFFORT
K1	254	1217	287.0	338.7	351.7	450.2	5.82511	5.86278	6.10969	5.65948
K2	214	788	86.9	219.1	182.9	283.6	5.38953	5.20894	5.64756	4.46476
K3	254	1611	258.7	448.7	481.9	602.7	6.10635	6.17774	6.40142	5.55567
K4	41	507	82.5	141.2	95.5	178.9	4.95018	4.55913	5.18683	4.41280
K5	450	2307	1107.3	642.5	739.1	877.6	6.46537	6.60543	6.77719	7.00968
K6	450	1338	336.3	372.4	388.8	495.3	5.91997	5.96307	6.20516	5.81800
K7	50	421	84.0	117.3	75.9	147.6	4.76473	4.32942	4.99451	4.43082
K8	43	100	23.2	27.8	11.9	32.5	3.32504	2.47654	3.48124	3.14415
K9	200	993	130.3	276.3	239.8	361.8	5.62149	5.47981	5.89109	4.86984
K10	39	240	72.0	66.8	36.9	81.9	4.20170	3.60821	4.40550	4.27667
K11	129	789	258.7	219.8	183.1	285.0	5.39272	5.21003	5.65249	5.55567
K12	289	1593	116.0	267.2	475.9	521.3	5.58800	6.16521	6.25633	4.75359
K13	161	691	157.0	192.5	158.1	247.5	5.26010	5.06323	5.51141	5.05625
K14	165	1348	246.9	375.3	391.5	518.9	5.92773	5.96999	6.25171	5.50898
A1	130	1750	673.7	486.5	513.1	656.6	6.18724	6.24047	6.48708	6.51278
A2	318	1902	692.1	528.7	568.1	716.7	6.27042	6.34230	6.57466	6.53973
A3	20	428	73.0	118.9	77.2	149.7	4.77828	4.34640	5.00863	4.29046
A4	54	759	138.8	135.3	156.4	241.7	4.90749	5.05242	5.48770	4.93303
A5	62	431	189.5	119.8	77.8	150.8	4.78582	4.35414	5.01595	5.24439
A6	28	283	65.8	78.6	49.2	98.2	4.36437	3.89589	4.58701	4.18662
A7	35	205	52.6	57.0	30.7	69.0	4.04305	3.42426	4.23411	3.96272
A8	30	289	32.2	80.4	50.3	99.2	4.38701	3.91801	4.59714	3.47197
A9	48	680	84.9	189.0	155.3	243.5	5.24175	5.04536	5.49512	4.44147
A10	93	794	125.0	220.7	184.3	286.2	5.39680	5.21656	5.65669	4.82831
A11	57	512	71.1	142.3	111.4	180.5	4.95794	4.71313	5.19573	4.26409
A12	22	224	19.1	62.3	33.5	75.9	4.13196	3.51155	4.32942	2.94969
A13	24	417	49.3	115.9	74.1	145.8	4.75273	4.30542	4.98224	3.89792
A14	42	682	78.9	121.5	137.5	215.9	4.79991	4.92362	5.37482	4.36818
A15	40	209	27.0	57.8	31.2	70.5	4.05699	3.44042	4.25561	3.29584
A16	96	512	103.9	142.3	111.4	180.8	4.95794	4.71313	5.19739	4.64343
A17	40	606	120.4	108.1	120.3	190.9	4.68306	4.78999	5.25175	4.79082
A18	52	400	58.6	111.2	68.1	139.6	4.71133	4.22098	4.93878	4.07073
A19	94	1235	250.7	343.3	358.0	455.4	5.83860	5.88053	6.12118	5.52426
A20	110	1572	402.6	280.2	419.7	519.2	5.63550	6.03954	6.25229	5.99794
A22	24	694	77.6	98.9	108.0	104.9	4.59411	4.68213	4.65301	4.35157
A24	29	260	40.1	72.2	45.2	88.5	4.27944	3.81110	4.48300	3.69138

LSBF Analysis of Model Comparison Data

SPANS Log(X)-log(Y) Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	25.93248	25.93248	145.302	0.0001
Error	34	6.06809	0.17847		
C Total	35	32.00057			

Root MSE	0.42246	R-square	0.8104
Dep Mean	4.74371	Adj R-sq	0.8048
C.V.	8.90570		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	-1.186182	0.49695272	-2.387	0.0227
LOGSPANS	1	1.169734	0.09704023	12.054	0.0001

Checkpoint Log(X)-log(Y) Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	25.33381	25.33381	129.201	0.0001
Error	34	6.66676	0.19608		
C Total	35	32.00057			

Root MSE	0.44281	R-square	0.7917
Dep Mean	4.74371	Adj R-sq	0.7855
C.V.	9.33468		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEPT	1	0.600180	0.37193001	1.614	0.1158
LOGCHECK	1	0.849729	0.07475631	11.367	0.0001

Costar Log(X)-log(Y) Equation
Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	25.71016	25.71016	138.965	0.0001
Error	34	6.29041	0.18501		
C Total	35	32.00057			

Root MSE	0.43013	R-square	0.8034
Dep Mean	4.74371	Adj R-sq	0.7976
C.V.	9.06737		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > ¦T¦
INTERCEPT	1	-1.061685	0.49766021	-2.133	0.0402
LOGCOSTAR	1	1.083145	0.09188277	11.788	0.0001

Appendix E: *Wilcoxin T Data*

SPANS Wilcoxin Data

				Rank	
			Diff	-	+
1	287.0	338.7	51.7		19
2	86.9	219.1	132.2		30
3	258.7	448.7	190.0		35
4	82.5	141.2	58.7		21
5	1107.3	642.5	-464.8	36	
6	336.3	372.4	36.1		12
7	84.0	117.3	33.3		10
8	23.2	27.8	4.6		3
9	130.3	276.3	146.0		31
10	72.0	66.8	-5.2	4	
11	258.7	219.8	-38.9	14	
12	116.0	267.2	151.2		32
13	157.0	192.5	35.5		11
14	246.9	375.3	128.4		29
15	673.7	486.5	-187.2	34	
16	692.1	528.7	-163.4	33	
17	73.0	118.9	45.9		17
18	138.8	135.3	-3.5	1	
19	189.5	119.8	-69.7	23	
20	65.8	78.6	12.8		6
21	52.6	57.0	4.4		2
22	32.2	80.4	48.2		18
23	84.9	189.0	104.1		27
24	125.0	220.7	95.7		26
25	71.1	142.3	71.2		24
26	19.1	62.3	43.2		16
27	49.3	115.9	66.6		22
28	78.9	121.5	42.6		15
29	27.0	57.8	30.8		8
30	103.9	142.3	38.4		13
31	120.4	108.1	-12.3	5	
32	58.6	111.2	52.6		20
33	250.7	343.3	92.6		25
34	402.6	280.2	-122.4	28	
35	77.6	98.9	21.3		7
36	40.1	72.2	32.1		9
T Score				178	488

Checkpoint Wilcoxin Data

			Rank		
		Diff	-	+	
1	287.0	351.7	64.7		24
2	86.9	182.9	96.0		27
3	258.7	481.9	223.2		34
4	82.5	95.5	13.0		10
5	1107.3	739.1	-368.2	36	
6	336.3	388.8	52.5		21
7	84.0	75.9	-8.1	7	
8	23.2	11.9	-11.3	9	
9	130.3	239.8	109.5		29
10	72.0	36.9	-35.1	19	
11	258.7	183.1	-75.6	26	
12	116.0	475.9	359.9		35
13	157.0	158.1	1.1		2
14	246.9	391.5	144.6		32
15	673.7	513.1	-160.6	33	
16	692.1	568.1	-124.0	31	
17	73.0	77.2	4.2		3.5
18	138.8	156.4	17.6		14
19	189.5	77.8	-111.7	30	
20	65.8	49.2	-16.6	12	
21	52.6	30.7	-21.9	16	
22	32.2	50.3	18.1		15
23	84.9	155.3	70.4		25
24	125.0	184.3	59.3		23
25	71.1	111.4	40.3		20
26	19.1	33.5	14.4		11
27	49.3	74.1	24.8		17
28	78.9	137.5	58.6		22
29	27.0	31.2	4.2		3.5
30	103.9	111.4	7.5		6
31	120.4	120.3	-0.1	1	
32	58.6	68.1	9.5		8
33	250.7	358.0	107.3		28
34	402.6	419.7	17.1		13
35	77.6	108.0	30.4		18
36	40.1	45.2	5.1		5
T Score				220	446

Costar Wilcoxin Data

			Rank		
		Diff	-	+	
1	287.0	450.2	163.2		29
2	86.9	283.6	196.7		30
3	258.7	602.7	344.0		35
4	82.5	178.9	96.4		20
5	1107.3	877.6	-229.7	32	
6	336.3	495.3	159.0		27
7	84.0	147.6	63.6		13
8	23.2	32.5	9.3		1
9	130.3	361.8	231.5		33
10	72.0	81.9	9.9		2
11	258.7	285.0	26.3		6
12	116.0	521.3	405.3		36
13	157.0	247.5	90.5		19
14	246.9	518.9	272.0		34
15	673.7	656.6	-17.1	4	
16	692.1	716.7	24.6		5
17	73.0	149.7	76.7		16
18	138.8	241.7	102.9		22
19	189.5	150.8	-38.7	9	
20	65.8	98.2	32.4		8
21	52.6	69.0	16.4		3
22	32.2	99.2	67.0		14
23	84.9	243.5	158.6		26
24	125.0	286.2	161.2		28
25	71.1	180.5	109.4		23
26	19.1	75.9	56.8		12
27	49.3	145.8	96.5		21
28	78.9	215.9	137.0		25
29	27.0	70.5	43.5		10
30	103.9	180.8	76.9		17
31	120.4	190.9	70.5		15
32	58.6	139.6	81.0		18
33	250.7	455.4	204.7		31
34	402.6	519.2	116.6		24
35	77.6	104.9	27.3		7
36	40.1	88.5	48.4		11
T Score				45	621

Appendix F: Percent Error Data

SPANS Percent Error Data

	ACTUAL EFFORT	SPANS	PERCENT ERROR	MRE	WITHIN ±30%	±20%
1	287.0	338.7	0.18	0.18	X	X
2	86.9	219.1	1.52	1.52		
3	258.7	448.7	0.73	0.73		
4	82.5	141.2	0.71	0.71		
5	1107.3	642.5	-0.42	0.42		
6	336.3	372.4	0.11	0.11	X	X
7	84.0	117.3	0.40	0.40		
8	23.2	27.8	0.20	0.20	X	X
9	130.3	276.3	1.12	1.12		
10	72.0	66.8	-0.07	0.07	X	X
11	258.7	219.8	-0.15	0.15	X	X
12	116.0	267.2	1.30	1.30		
13	157.0	192.5	0.23	0.23	X	
14	246.9	375.3	0.52	0.52		
15	673.7	486.5	-0.28	0.28	X	
16	692.1	528.7	-0.24	0.24	X	
17	73.0	118.9	0.63	0.63		
18	138.8	135.3	-0.03	0.03	X	X
19	189.5	119.8	-0.37	0.37		
20	65.8	78.6	0.19	0.19	X	X
21	52.6	57.0	0.08	0.08	X	X
22	32.2	80.4	1.49	1.49		
23	84.9	189.0	1.23	1.23		
24	125.0	220.7	0.77	0.77		
25	71.1	142.3	1.00	1.00		
26	19.1	62.3	2.27	2.27		
27	49.3	115.9	1.35	1.35		
28	78.9	121.5	0.54	0.54		
29	27.0	57.8	1.14	1.14		
30	103.9	142.3	0.37	0.37		
31	120.4	108.1	-0.10	0.10	X	X
32	58.6	111.2	0.90	0.90		
33	250.7	343.3	0.37	0.37		
34	402.6	280.2	-0.30	0.30	X	
35	77.6	98.9	0.27	0.27	X	
36	40.1	72.2	0.80	0.80		
Mean			0.51	0.62		
Standard Deviation			0.63	0.53		

Within ±30% 14 of 36 (.39)
Within ±20% 9 of 36 (.25)

Checkpoint Percent Error Data

	ACTUAL EFFORT	CHECKPOINT	PERCENT ERROR	MRE	WITHIN ±30%	WITHIN ±20%
1	287.0	351.7	0.23	0.23	X	
2	86.9	182.9	1.10	1.10		
3	258.7	481.9	0.86	0.86		
4	82.5	95.5	0.16	0.16	X	X
5	1107.3	739.1	-0.33	0.33		
6	336.3	388.8	0.16	0.16	X	X
7	84.0	75.9	-0.10	0.10	X	X
8	23.2	11.9	-0.49	0.49		
9	130.3	239.8	0.84	0.84		
10	72.0	36.9	-0.49	0.49		
11	258.7	183.1	-0.29	0.29	X	
12	116.0	475.9	3.10	3.10		
13	157.0	158.1	0.01	0.01	X	X
14	246.9	391.5	0.59	0.59		
15	673.7	513.1	-0.24	0.24	X	
16	692.1	568.1	-0.18	0.18	X	X
17	73.0	77.2	0.06	0.06	X	X
18	138.8	156.4	0.13	0.13	X	X
19	189.5	77.8	-0.59	0.59		
20	65.8	49.2	-0.25	0.25	X	
21	52.6	30.7	-0.42	0.42		
22	32.2	50.3	0.56	0.56		
23	84.9	155.3	0.83	0.83		
24	125.0	184.3	0.47	0.47		
25	71.1	111.4	0.57	0.57		
26	19.1	33.5	0.76	0.76		
27	49.3	74.1	0.50	0.50		
28	78.9	137.5	0.74	0.74		
29	27.0	31.2	0.16	0.16	X	X
30	103.9	111.4	0.07	0.07	X	X
31	120.4	120.3	-0.00	0.00	X	X
32	58.6	68.1	0.16	0.16	X	X
33	250.7	358.0	0.43	0.43		
34	402.6	419.7	0.04	0.04	X	X
35	77.6	108.0	0.39	0.39		
36	40.1	45.2	0.13	0.13	X	X
Mean			0.27	0.46		
Standard Deviation			0.65	0.53		

Within ±30% 17 of 36 (.47)
Within ±20% 13 of 36 (.36)

Costar Percent Error Data

	ACTUAL EFFORT	COSTAR	PERCENT ERROR	MRE	WITHIN ±30%	WITHIN ±20%
1	287.0	450.2	0.57	0.57		
2	86.9	283.6	2.26	2.26		
3	258.7	602.7	1.33	1.33		
4	82.5	178.9	1.17	1.17		
5	1107.3	877.6	-0.21	0.21	X	
6	336.3	495.3	0.47	0.47		
7	84.0	147.6	0.76	0.76		
8	23.2	32.5	0.40	0.40		
9	130.3	361.8	1.78	1.78		
10	72.0	81.9	0.14	0.14	X	X
11	258.7	285.0	0.10	0.10	X	X
12	116.0	521.3	3.49	3.49		
13	157.0	247.5	0.58	0.58		
14	246.9	518.9	1.10	1.10		
15	673.7	656.6	-0.03	0.03	X	X
16	692.1	716.7	0.04	0.04	X	X
17	73.0	149.7	1.05	1.05		
18	138.8	241.7	0.74	0.74		
19	189.5	150.8	-0.20	0.20	X	X
20	65.8	98.2	0.49	0.49		
21	52.6	69.0	0.31	0.31		
22	32.2	99.2	2.08	2.08		
23	84.9	243.5	1.87	1.87		
24	125.0	286.2	1.29	1.29		
25	71.1	180.5	1.54	1.54		
26	19.1	75.9	2.98	2.98		
27	49.3	145.8	1.95	1.95		
28	78.9	215.9	1.73	1.73		
29	27.0	70.5	1.61	1.61		
30	103.9	180.8	0.74	0.74		
31	120.4	190.9	0.59	0.59		
32	58.6	139.6	1.38	1.38		
33	250.7	455.4	0.82	0.82		
34	402.6	519.2	0.29	0.29	X	
35	77.6	104.9	0.35	0.35		
36	40.1	88.5	1.21	1.21		
Mean			1.02	1.05		
Standard Deviation			0.87	0.84		

Within ±30% 7 of 36 (.19)
Within ±20% 5 of 36 (.14)

Adjusted SPANS Percent Error Data
(Prediction/1.51)

	ACTUAL EFFORT	ADJUSTED SPANS	PERCENT ERROR	MRE	WITHIN ±30%	WITHIN ±20%
1	287.0	224.3	-0.22	0.22	X	
2	86.9	145.1	0.67	0.67		
3	258.7	297.2	0.15	0.15	X	X
4	82.5	93.5	0.13	0.13	X	X
5	1107.3	425.5	-0.62	0.62		
6	336.3	246.6	-0.27	0.27	X	
7	84.0	77.7	-0.08	0.08	X	X
8	23.2	18.4	-0.21	0.21	X	
9	130.3	183.0	0.40	0.40		
10	72.0	44.2	-0.39	0.39		
11	258.7	145.6	-0.44	0.44		
12	116.0	177.0	0.53	0.53		
13	157.0	127.5	-0.19	0.19	X	X
14	246.9	248.5	0.01	0.01	X	X
15	673.7	322.2	-0.52	0.52		
16	692.1	350.1	-0.49	0.49		
17	73.0	78.7	0.08	0.08	X	X
18	138.8	89.6	-0.35	0.35		
19	189.5	79.3	-0.58	0.58		
20	65.8	52.1	-0.21	0.21	X	
21	52.6	37.7	-0.28	0.28	X	
22	32.2	53.2	0.65	0.65		
23	84.9	125.2	0.47	0.47		
24	125.0	146.2	0.17	0.17	X	X
25	71.1	94.2	0.33	0.33		
26	19.1	41.3	1.16	1.16		
27	49.3	76.8	0.56	0.56		
28	78.9	80.5	0.02	0.02	X	X
29	27.0	38.3	0.42	0.42		
30	103.9	94.2	-0.09	0.09	X	X
31	120.4	71.6	-0.41	0.41		
32	58.6	73.6	0.26	0.26	X	
33	250.7	227.4	-0.09	0.09	X	X
34	402.6	185.6	-0.54	0.54		
35	77.6	65.5	-0.16	0.16	X	X
36	40.1	47.8	0.19	0.19	X	X
Mean			0.00	0.34		
Standard Deviation			0.42	0.24		

Within ±30% 18 of 36 (.50)
Within ±20% 12 of 36 (.33)

Adjusted Checkpoint Percent Error Data
(Prediction/1.27)

	ACTUAL EFFORT	ADJUSTED SPANS	PERCENT ERROR	MRE	WITHIN ±30%	WITHIN ±20%
1	287.0	276.9	-0.04	0.04	X	X
2	86.9	144.0	0.66	0.66		
3	258.7	379.4	0.47	0.47		
4	82.5	75.2	-0.09	0.09	X	X
5	1107.3	582.0	-0.47	0.47		
6	336.3	306.1	-0.09	0.09	X	X
7	84.0	59.8	-0.29	0.29	X	
8	23.2	9.4	-0.60	0.60		
9	130.3	188.8	0.45	0.45		
10	72.0	29.1	-0.60	0.60		
11	258.7	144.2	-0.44	0.44		
12	116.0	210.4	-0.81	0.81		
13	157.0	124.5	-0.21	0.21	X	
14	246.9	308.3	0.25	0.25	X	
15	673.7	404.0	-0.40	0.40		
16	692.1	447.3	-0.35	0.35		
17	73.0	60.8	-0.17	0.17	X	X
18	138.8	123.1	-0.11	0.11	X	X
19	189.5	61.3	-0.68	0.68		
20	65.8	38.7	-0.41	0.41		
21	52.6	24.2	-0.54	0.54		
22	32.2	39.6	0.23	0.23	X	
23	84.9	122.3	0.44	0.44		
24	125.0	145.1	0.16	0.16	X	X
25	71.1	87.7	0.23	0.23	X	
26	19.1	26.4	0.38	0.38		
27	49.3	58.3	0.18	0.18	X	X
28	78.9	108.3	0.37	0.37		
29	27.0	24.6	-0.09	0.09	X	X
30	103.9	87.7	-0.16	0.16	X	X
31	120.4	94.7	-0.21	0.21	X	
32	58.6	53.6	-0.08	0.08	X	X
33	250.7	281.9	0.12	0.12	X	X
34	402.6	330.5	-0.18	0.18	X	X
35	77.6	85.0	0.10	0.10	X	X
36	40.1	35.6	-0.11	0.11	X	X
Mean			-0.09	0.32		
Standard Deviation			0.38	0.22		

Within ±30% 20 of 36 (.56)
Within ±20% 14 of 36 (.39)

Adjusted Costar Percent Error Data
(Prediction/2.02)

	ACTUAL EFFORT	ADJUSTED SPANS	PERCENT ERROR	MRE	WITHIN ±30%	WITHIN ±20%
1	287.0	222.9	-0.22	0.22	X	
2	86.9	140.4	0.62	0.62		
3	258.7	298.4	0.15	0.15	X	X
4	82.5	88.6	0.07	0.07	X	X
5	1107.3	434.5	-0.61	0.61		
6	336.3	245.2	-0.27	0.27	X	
7	84.0	73.1	-0.13	0.13	X	X
8	23.2	16.1	-0.31	0.31		
9	130.3	179.1	0.37	0.37		
10	72.0	40.5	-0.44	0.44		
11	258.7	141.1	-0.45	0.45		
12	116.0	258.1	1.22	1.22		
13	157.0	122.5	-0.22	0.22	X	
14	246.9	256.9	0.04	0.04	X	X
15	673.7	325.0	-0.52	0.52		
16	692.1	354.8	-0.49	0.49		
17	73.0	74.1	0.01	0.01	X	X
18	138.8	119.7	-0.14	0.14	X	X
19	189.5	74.7	-0.61	0.61		
20	65.8	48.6	-0.26	0.26	X	
21	52.6	34.2	-0.35	0.35		
22	32.2	49.1	0.52	0.52		
23	84.9	120.5	0.42	0.42		
24	125.0	141.7	0.13	0.13	X	X
25	71.1	89.4	0.26	0.26	X	
26	19.1	37.6	0.97	0.97		
27	49.3	72.2	0.46	0.46		
28	78.9	106.9	0.35	0.35		
29	27.0	34.9	0.29	0.29	X	
30	103.9	89.5	-0.14	0.14	X	X
31	120.4	94.5	-0.22	0.22	X	
32	58.6	69.1	0.18	0.18	X	X
33	250.7	225.4	-0.10	0.10	X	X
34	402.6	257.0	-0.36	0.36		
35	77.6	51.9	-0.33	0.33		
36	40.1	43.8	0.09	0.09	X	X
Mean			0.00	0.34		
Standard Deviation			0.43	0.25		

Within ±30% 18 of 36 (.50)
Within ±20% 11 of 36 (.31)

Bibliography

1. Albanese, Frank, Jr. *The Application of Regression-Based and Function Point Software Sizing Techniques to Air Force Programs*. MS thesis, AFIT/GCA/LSY/88S-1. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1988 (AD-A201538).

2. Albrecht, Alan J. "Measuring Application Development Productivity," in *Proceedings Joint SHARE/GUIDE/IBM Application Development Symposium*, October 1979, 33-34.

3. -----, and John E. Gaffney, Jr. "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation," *IEEE Transactions on Software Engineering, Vol. SE-9, No.6:* 639-648 (November 1983).

4. Behrens, Charles A. "Measuring the Productivity of Computer System Development Activities with Function Points," *IEEE Transaction on Software Engineering, Vol. SE-9, No. 6:* 649-652 (November 1983).

5. Boehm, Barry W. "Software Engineering Economics," *IEEE Transactions on Software Engineering, Vol. SE-10, No. 1:* 4-21 (January 1984).

6. -----. *Software Engineering Economics*. Englewood Cliffs NJ: Prentice-Hall Inc., 1981.

7. Daly, Bryan A. *A Comparison of Software Schedule Estimators*. MS thesis, AFIT/GCA/LSQ/90S-1. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1990 (AD-A229532).

8. Ferens, Daniel V. "Computer Software Size Estimation," *Proceedings of the 1988 National Aerospace and Electronics Conference*, 1-6. Dayton OH: 1988.

9. -----. "An Introduction to Software Parametric Cost Estimating". Class text distributed in CMGT 676, Software Cost Estimation. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, November 1990.

10. "Function Point Metrics for Software." Briefing to Air Force Standard Systems Center, Gunter AFB AL. Tecelote Research Inc., Montgomery AL, June 1990.

11. Gaffney, John E., Jr. and Richard Werling. *Estimating Software Size From Counts of Program Externals, A Generalization of Function Points*. Abstract from the 13th Annual International Conference of the International Society of Parametric Analysts.

12. Graver, Charles A. et al. *SPANS User's Manual*. Montgomery AL: Tecelote Research, Inc., 1991.

13. Humphrey, Watts S. *Managing the Software Process*. Reading MA: Addison-Wesley Publishing Company, 1990.

14. International Function Point User's Group. *Function Point Counting Practices Manual, Release 3.0*. Westerville OH: International Function Point User's Group, 1990.

15. Jeffrey, D. Ross et al. *The Validity, Reliability, and Practicality of Function Points as a Measure of Software Size*. Department of Information Systems, University of New South Wales, 1988.

16. Jones, Capers. *Applied Software Measurement: Assuring Productivity and Quality*. New York NY: McGraw-Hill, Inc,. 1991.

17. Kankey, Roland D. *The Challenge of Software Maintenance Costing*. Technical Report AU-AFIT-LSQ-89-1. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March 1989.

18. Kemmerer, Chris F. "An Empirical Validation of Software Cost Estimation Models," *Communications of the ACM, Vol. 30, No. 5*: 416-429 (May 1987).

19. Low, Graham C. and D. Ross Jeffrey. "Function Points in the Estimation and Evaluation of the Software Process," *IEEE Transactions on Software Engineering, Vol. 16, No. 1*: 64-71 (January 1990).

20. Murphy, Richard. Class Lecture in QMGT 672, Transformations of Variables in LSBF Regression Equation. School of Systems and Logistics, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, February, 1991.

21. Neter, John et al. *Applied Linear Regression Models* (Second Edition). Homewood IL: Richard D.Irwin, Inc., 1989.

22. Newbold, Paul. *Statistics For Business and Economics* (Second Edition). Englewood Cliffs NJ: Prentice Hall, 1988.

23. Reese, Richard M. and Jim Tamulevicz. "A Survey of Software Sizing Methodologies and Tools," *Journal of Parametrics, Vol. VII, No. 2*: 35-55 (June 1987).

24. Reifer, Donald J. *Analytical Size Estimation Tool -- Real-Time (ASSET-R): An Overview* (RCI-TN-269). Torrance CA: Reifer Consultants, Inc., 2 May 1987.

25. Softstar Systems. *Costar User's Manual*. Amherst NH: Softstar Systems, 1989.

26. -----. *Costar Software Cost Estimation Tool*. Amherst NH: Softstar Systems, 1990.

27. Software Productivity Research, Inc. *Checkpoint User's Guide, Release 2.0*. Burlington MA: Software Productivity Research, Inc., 1991.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1990

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
A COMPARATIVE STUDY OF THE RELIABILITY OF FUNCTION POINT ANALYSIS IN SOFTWARE DEVELOPMENT EFFORT ESTIMATION MODELS

5. FUNDING NUMBERS

6. AUTHOR(S)
Robert B. Gurner, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GCA/LSY/91S-2

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Standard Systems Center
Gunter AFB AL 36114-6343

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The Air Force of the 1990's is steadily growing more reliant on software systems. However, the struggle to develop reliable cost and effort estimation tools continues. The Standard Systems Center (SSC), Gunter AFB AL, has adopted the use of Function Point Analysis to improve estimation of data processing, management and communication systems. Function point analysis was introduced by IBM's Alan Albrecht, in 1979, as an alternative to source line of code (SLOC) as a size and productivity measure.

In 1991, Tecelote Research, Inc., under contract to the SSC, delivered the Software Program Acquisition Network Simulation (SPANS) model incorporating the capability to perform estimates using function point measures. This research examines the ability of SPANS to reliably and accurately estimate software project effort with function points. A further investigation compares the predictions derived by SPANS with two commercially available software estimation tools, Checkpoint Costar.

14. SUBJECT TERMS
Software Cost, Software Size, Function Point Analysis, Management Information Systems (Software)

15. NUMBER OF PAGES
93

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFIT Control Number AFIT/GCA/LSY/91S-2

AFIT RESEARCH ASSESSMENT

The purpose of this questionnaire is to determine the potential for current and future applications of AFIT thesis research. Please return completed questionnaires to: AFIT/LSC, Wright-Patterson AFB OH 45433-6583.

1. Did this research contribute to a current research project?

   a. Yes b. No

2. Do you believe this research topic is significant enough that it would have been researched (or contracted) by your organization or another agency if AFIT had not researched it?

   a. Yes b. No

3. The benefits of AFIT research can often be expressed by the equivalent value that your agency received by virtue of AFIT performing the research. Please estimate what this research would have cost in terms of manpower and/or dollars if it had been accomplished under contract or if it had been done in-house.

   Man Years ____________ $____________

4. Often it is not possible to attach equivalent dollar values to research, although the results of the research may, in fact, be important. Whether or not you were able to establish an equivalent value for this research (3 above), what is your estimate of its significance?

   a. Highly Significant b. Significant c. Slightly Significant d. Of No Significance

5. Comments

____________________ ____________________
Name and Grade Organization

____________________ ____________________
Position or Title Address