Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods

Ingunn Myrtveit, Erik Stensrud, *Member*, *IEEE*, and Ulf H. Olsson

Abstract—Missing data are often encountered in data sets used to construct effort prediction models. Thus far, the common practice has been to ignore observations with missing data. This may result in biased prediction models. In this paper, we evaluate four missing data techniques (MDTs) in the context of software cost modeling: listwise deletion (LD), mean imputation (MI), similar response pattern imputation (SRPI), and full information maximum likelihood (FIML). We apply the MDTs to an ERP data set, and thereafter construct regression-based prediction models using the resulting data sets. The evaluation suggests that only FIML is appropriate when the data are not missing completely at random (MCAR). Unlike FIML, prediction models constructed on LD, MI and SRPI data sets will be biased unless the data are MCAR. Furthermore, compared to LD, MI and SRPI seem appropriate only if the resulting LD data set is too small to enable the construction of a meaningful regression-based prediction model.

Index Terms—Software effort prediction, cost estimation, missing data, imputation methods, listwise deletion, mean imputation, similar response pattern imputation, full information maximum likelihood, log-log regression, ERP.

1 Introduction

Missing data are often encountered in software engineering data sets that are used to construct effort prediction models [37], [13]. The International Software Benchmarking Standards Group (ISBSG) database is no exception. It has a large fraction of missing data, in some variables more than 40 percent [4], [16]. The ERP[1] data set presented in this paper also includes several observations with missing data in one or more variables.

There are several reasons why observations may have missing values. High data collection cost may cause missing values. The cost of gathering and reporting data from software projects is nonnegligible. DeMarco estimates it would constitute a 5 to 10 percent of total cost [11]. From our personal experience as an ERP project manager, we know that some data indeed cost more to collect. For example, it is more difficult, and, therefore, costly to collect data on Interfaces and Effort than on Users, Sites and Modules. Therefore, we expected, and did indeed find, that there are more missing values in Interfaces and Effort than in Users, Sites, and Modules (see Table 1).

Another reason for missing values is that some values are so called wild values. A wild value is a value we know is untrue. For example, if a reported Effort is negative, we know it must be untrue. Typically, a wild value is due to a punching error. Also, it may be due to someone who did not know how to measure and report that variable correctly. A common data screening procedure is to replace wild values by "missing" thereby creating additional missing values. It should be observed that a wild value is not synonymous to an outlier, an outlying observation.

Yet another reason why observations may have missing values is that some respondents just do not report some of the variables for some reason.

There are basically three ways to handle missing data. One option is to remove incomplete observations by *listwise deletion (LD)*. Alternatively, one may fill in the holes by some *imputation* method. A third option is to use a model-based method. In the two first cases, *complete-case* analysis methods may be applied to the resulting complete data set whereas a *model*-based method like the full information maximum likelihood method (FIML) is able to analyse incomplete data sets directly. That is, it is an *in*complete-case analysis method.

Thus far, the common practice when constructing regression-type effort prediction models has been to apply a two-stage procedure, consisting of ignoring missing data (using LD) before applying regression analysis to construct the effort prediction model. LD is routinely used since most statistical packages use LD as default. Also, some effort prediction models based on machine learning use LD as default. An example is estimation by analogy as implemented in ANGEL [34].

Unfortunately, LD has several drawbacks. The most obvious drawback is that it discards a considerable amount of information. This is especially unfortunate in empirical

[1] Actually, package-enabled reengineering (PER) *projects* implement enterprise resource planning (ERP) *systems*. Therefore, the term "PER project" is more appropriate than the term "ERP project." Still, we have chosen to use the latter since the term ERP recently has become an established term in the research community with the April 2000, vol. 43, no. 4, issue of the *Communications of the ACM*.

- *The authors are with the Norwegian School of Management, PO Box 580, N-1301 Sandvika, Norway.*
E-mail: {ingunn.myrtveit, erik.stensrud, ulf.olsson}@bi.no.

Manuscript received 01 Jan. 2001; revised 01 May 2001; accepted 01 Aug. 2001.
Recommended for acceptance by S. Pfleeger.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 114703.

TABLE 1
Descriptive Statistics for ERP Data Set

Variable	N	N*	Mean	Median	StDev	Min	Max
Users	176	0	1297	338	2867	6	18000
Sites	175	1	45	7	172	0	2000
Plants	174	2	26	5	78	0	500
Companies	174	2	13	1.5	53	1	400
Interfaces	139	37	39	15	95	0	943
EDI	112	64	6	0	30	0	300
Conversions	125	51	27	14	40	0	340
Modifications	119	57	18	5	42	0	300
Reports	127	49	90	40	208	0	2000
Modules	175	1	5.2	5.0	2.7	0	13
Effort	99	77	10850	6200	14583	102	111420

software engineering because the data sets usually are small (i.e., N << 100). Removing observations from small data sets makes them even smaller, in some cases so small that it becomes meaningless to construct a regression-based effort prediction model on the remaining data. In such a case, the effort prediction model will not inspire much confidence.

Last but not least, in addition to the obvious loss of information, LD may introduce a *bias* in the data. This occurs if the complete observations are not a random subsample of the original (and incomplete) sample. We find it necessary to stress this point. The seemingly innocent LD does not deal correctly with incomplete data sets when the values are *not* missing completely at random. The implication of constructing an effort prediction model on a biased data set due to inappropriate use of LD is that the prediction model will be biased and, therefore, misleading. It may be biased in dangerous ways. For example, the model may be biased so that it seemingly performs extremely well in terms of accuracy, too well, thus seducing the user into unrealistic expectations regarding prediction accuracy.

There are, therefore, several reasons to take good care of all the data and not just delete incomplete observations. Hence, the interest in alternative missing data techniques (MDTs) that waste less information than LD and hopefully are more robust than LD against bias caused by nonrandom missing data.

Intuitively, imputation-based or model-based MDTs seem to be more attractive choices than LD that merely discards incomplete observations. It should also be observed that MDTs are widely applied in other disciplines such as the social sciences.

To our knowledge, this is the first time that a critical evaluation of *sampling*-based (e.g., imputation-based and LD) as well as *model*-based MDTs is reported in the context of constructing effort prediction models, emphasising advantages, consequences, and pitfalls of the different MDTs. To our knowledge, there are no empirical studies of FIML with real data.

The main contribution of this paper is to examine whether or not the MDTs deliver what they promise and their pros and cons. This is a necessary piece of knowledge in order to perform empirical studies in software engineering correctly. In general, researchers in empirical software engineering need to acquire more knowledge of methods and techniques that may be applicable [1]. Specifically, we need to understand the limitations of the seemingly innocent and widely used, and abused, LD.

We assess a selection of widely applied MDTs. The MDTs we have investigated are listwise deletion (LD), mean imputation (MI), similar response pattern imputation (SRPI) and the full information maximum likelihood (FIML) method. The motivation for selecting MI is that it is a fast expedient and probably the most widely used MDT (except LD, of course). SRPI has been selected because it is recent and because the method of identifying similar observations has intuitive appeal to software engineering researchers and practitioners compared to, e.g., the more advanced multiple imputation techniques. Besides, it seems simple to apply. FIML has been chosen because it is model-based. It should be observed that FIML is an MDT as well as a regression analysis technique at the same time. As opposed to FIML, which is model-based, all the other MDTs are sampling-based.

The overall research question we investigate is whether any of the MDTs add value compared with LD and in case, under what circumstances they may add value. In other words, is there any reason at all why software engineering researchers and practitioners cannot continue using "the same procedure as usual," i.e., LD and not worry about missing data and MDTs?

In the context of constructing software effort prediction models using historical data, it should be observed that we are dependent upon data sets that are representative of the true population. Only to the extent that the data set at hand, the *sample*, reflects the *population* data set (i.e., the data set comprising all past as well as all future projects) can we trust the results derived from it. In software engineering in general, and ERP projects in particular, the *true* effort prediction model is almost never known. However, if several samples all confirm the same model, we believe that the sample model is a good approximation of the true population model. This is a way of validating the model. Also, it is likely that these samples are true subsamples of the population. Thus, one purpose of applying MDTs is to obtain data sets that hopefully are *more representative samples* of the true population. The prediction models based on the sample will then be more representative of the *true prediction model* which means that the accuracy figures will provide us with realistic expectations rather than seduce us into believing in a falsely high (or low) accuracy.

The overall research question, i.e., whether any of the MDTs add value compared with LD, has been translated into a number of more specific research questions. The research questions on MDTs in the context of constructing regression-based effort prediction models are as follows:

- Are any of the MDTs (including LD) robust against bias caused by nonrandom missing data?
- May any of the MDTs (including LD) actually introduce a bias (that was not there before the application of the MDT)? It is an undesirable property of an MDT if it introduces bias.
- To which extent do the MDTs prevent information loss? For example, if the application of LD results in

a data set that is too small and, therefore, unsuitable for regression analysis, may any other MDTs prevent information loss to the extent that the data set becomes usable and lends itself to regression analysis?

- Are the ERP data missing completely at random? If not, which MDTs are appropriate, if any?

In addition, we address some research questions concerning the effort prediction models more directly.

- Is the model specification (in terms of the choice of variables and choice of a nonlinear model) correct? Since there exists no "true" model for effort prediction of ERP projects, we somehow should validate the model specification. One validation method is to investigate whether different methods converge. In our case, we investigate whether the regression models constructed on the various data sets converge. Generally, this is a good idea in software engineering since we do not know what the true model is like. We do not have models based on a solid theoretical basis. Rather, our models are more explorative, more likened to hypotheses regarding the nature of software projects.
- Given a correct model specification and a representative data set, what degree of accuracy can we realistically expect? In other words, what is the accuracy of the least biased effort prediction model, i.e., of the model closest to the true (or population) model?

Our overall evaluation suggests that the effort prediction model based on FIML is the least biased for the ERP data set. Second comes the regression model constructed on the LD data set. In our case, the LD data set was sufficiently large to enable the construction of a good regression-type prediction model. However, the other MDTs may be good choices when LD results in a too small data set. The pattern of missingness and the fraction of missing data are crucial considerations when deciding between MI and SRPI.

2 ERP Data

The data set consists of 176 active and completed ERP projects. All the ERP projects in the sample implement ERP systems from one single ERP vendor: SAP. One hundred and seventy projects have implemented the client-server version (R/3) of SAP. Six projects have implemented the mainframe version (R/2). Therefore, it is a homogeneous data set. The variables include ten ERP size measures (predictor variables) and total effort as the response variable. Descriptive statistics for the data set is provided in Table 1. ("N^*" means "N missing.") All the missing values are truly missing. We do not have any "don't know" responses. We observe that the projects span from 102 workdays to 111,420 workdays.

We have not removed active projects since actuals for many of the predictor variables exist and are reported early in the project. For example, Users, Sites, Plants, Companies, and Modules form part of the project scope (and, thus, the contract) and are therefore known after an initial analysis. (In a previous conference version [28], we provided descriptive statistics where active projects were screened unless they were so close to completion that reported values could be treated as actuals for our purpose.)

In Table 1, we also observe that Effort has most missing values ($N^* = 77$) which is not surprising. More surprising, EDI (Electronic Data Interchange), too, has many missing values ($N^* = 64$). Probably, EDI is not missing completely at random (i.e., not-MCAR). We suspect that some of the missing EDI values are in fact zeros, but not all. Therefore, it would be wrong to replace all missing EDI values with "0." Also, an MDT requiring MCAR data may not be appropriate to impute EDI.

We believe that the other variables likely are MCAR. We have no reason to suspect any nonrandom missing data in these variables. The main reason for the many missing values in Interfaces, Conversions, Modifications, and Reports is that we have included active projects in this study. Active projects naturally do not have actuals for all variables as some of them are not known until project completion. Another potential reason for missing values is that it takes more effort to count these variables than to collect counts on Users, Sites, Plants, and Modules. We believe however that this reason accounts for very few missing values.

The size measures for sizing this type of ERP projects is the intraorganizational standard. It is beyond the scope of this paper to explain these size measures. Interested readers are referred to [36] for a definition of the measures.

The data were gathered from 1990 to 1998 in a multinational consultant organization (Accenture, formerly Andersen Consulting) with 70,000 employees. The projects span many industries and countries in all regions of the world.

The data have been reported by project managers who themselves use the database to plan and estimate future projects. The company has a standard ERP methodology. Therefore, the data presumably have been reported in a consistent manner. See also [35], [27] for an evaluation of the data quality.

3 Missing Data Techniques (MDTs)

Missing data techniques (MDTs) can be roughly grouped into:

1. techniques ignoring incomplete observations,
2. imputation-based techniques,
3. weighting techniques, and
4. model-based techniques [22].

Weighting techniques are not presented in this paper.

The simplest technique is to ignore incomplete observations by applying listwise deletion (LD). LD is easy to carry out and is implemented as default in most statistical packages. It may be satisfactory with small amounts of missing data. The drawback is that its application may result in too small data sets if the fraction of missing values is high. Another serious drawback with LD is that it can lead to serious biases if the data are not missing at random.

Imputation-based methods replace missing values by suitable estimates. This allows standard complete-data statistical methods like ordinary least squares (OLS) regression analysis to be applied to the imputed data set.

There are several imputation methods. A simple and common expedient is *mean imputation* (MI) [22], [23]. Another, more recent, imputation technique is *similar response pattern imputation* (SRPI) proposed by Jøreskog and Sørbom [18]. The drawback with imputation-based MDTs is that artificial values are substituted for the missing values potentially causing a bias. This potentially biased data set is thereafter treated as real in the subsequent data analysis, thus leading to biased results.

Model-based (or likelihood-based) methods do neither remove incomplete observations nor replace missing values by imputation. Rather, these MDTs define a model for the partially missing data and base inferences on the likelihood under that model, with parameters estimated by methods such as maximum likelihood. The *full information maximum likelihood* (FIML) method is model-based. An advantage of this approach is that the model assumptions can be evaluated as opposed to ad hoc sampling-based imputation methods like MI and SRPI where there are no model assumptions that can be evaluated [22].

In this paper, we evaluate and discuss the LD, MI, SRPI, and FIML approaches and compare the three latter against LD. LD, therefore, serves as a baseline.

3.1 MI, SRPI, and FIML

Mean imputation (MI). A common method of imputing missing data is the substitution of the arithmetic mean. Anderson et al. [2] provide the following rationale; "In the case of normal distribution, the sample mean provides an optimal estimate of the most probable value" (p. 425). Although one may impute all missing values of x_i, the variance of x_i will shrink, because all values of x_i that are added will contribute nothing to the variance. The use of MI will affect the correlation between the imputed variable and any other by decreasing its variability. In addition, if a large number of values are imputed using the mean, the frequency distribution of the imputed variable may be misleading because too many centrally located values create a more leptokurtic (slim or long-tailed) distribution [30].

Similar response pattern imputation (SRPI). In this paper, we have used the *SRPI* imputation method as implemented in the statistical tool PRELIS 2.3 [19]. The idea behind the SRPI technique is to identify the most similar project without missing observations and copy the values of this project to fill in the holes in the project with missing values. The *least squares criterion* in normalized space is used as the similarity measure. The set of variables used to define the multidimensional space is called *matching* variables. Formally, the method is stated as follows.

Let $y_1, \dots y_p$ be the variables to be studied, and let $x_1, \dots x_q$ be the matching variables. Let $z_1, \dots z_q$ be the standardized values of $x_1, \dots x_q$. Furthermore, let y_k be the variable whose missing values are to be imputed. Let project $\underline{a}$ be a project where y_k is missing and which is complete in the matching variables $x_1, \dots x_q$. Find all projects that are complete in the matching variables $x_1, \dots x_q$ as well as in y_k and that minimize

$$\sum_{j=1}^{n} (z_{bj} - z_{aj})^2. \tag{1}$$

Two cases will occur. 1) There is a single project $\underline{b}$ which minimizes (1). In this case, y_{ka} is replaced with y_{kb}. 2) There are $\underline{n}$ projects that all minimize (1). Let their y-values be $y_{k1}, \dots y_{kn}$. In this case, y_{ka} is replaced with the mean of $y_{k1}, \dots y_{kn}$, $y_{k\text{MEAN}}$. Unlike most other MDTs like e.g., MI, SRPI functions with continuous as well as with ordinal variables [20]. It should also be observed that a feature of SRPI is that it does not impute a value if the distance between the matching and the target case is too large. In other words, SRPI protects us to some extent from getting strange observations during imputation.

Full information maximum likelihood (FIML). FIML is a *model*-based method as opposed to MI and SRPI that are *sampling*-based. FIML is based on the principle of maximizing the log-likelihood. The Maximum Likelihood or ML-estimator is well-known in the literature for its efficiency and is implemented in most statistical software for treating multivariate analysis with *complete* data sets. Until recently, ML estimation of *incomplete* data sets has not been an option in software packages because of the computational effort. Software like LISREL [17], Amos [6], and Mx [29] offers the FIML-estimator when missing data are present.

FIML assumes that the data comes from a *multivariate normal distribution* and maximizes the likelihood of the theoretical model given the observed data. Maximum Likelihood estimation of incomplete data has been addressed by several authors including Anderson [1], Browne [8], Little and Rubin, [22], [23], Muthen et al. [26], Arbucle [5], and Neal [29].

Compared with sampling-based methods like MI and SRPI, the advantage of likelihood-based methods like FIML is that the results will not be biased even if the data are not missing completely at random. (See more on this in section "MAR and MCAR.") FIML is also robust to data that do not comply completely with the multivariate normal distribution requirement [7]. The drawback with maximum likelihood methods is that they require relatively large data sets. Simulation studies for *complete* data sets have demonstrated that the chi-square estimator $(N - 1)F_{ML}$ is inaccurate for samples under 100 [7], [3]. Therefore, it is reasonable to assume that a similar sample size is required for FIML (i.e., samples with close to 100 complete observations and more than 100 when incomplete observations are included). Hence, it may be a problem to apply FIML in software engineering where the data sets often are too small (i.e., $(N << 100)$. The FIML procedure is presented in the Appendix "FIML procedure."

3.2 Other Imputation Techniques

Regression Imputation (RI) (also called conditional mean imputation). One alternative method would be to estimate the missing values by using *regression analysis*. This method replaces missing values by predicted values from a regression of the missing item on items observed be the unit. However, we did not consider this approach in this paper as we are applying regression analysis to the imputed data set.

Hot deck imputation (HDI). Hot deck imputation replaces missing values by drawing from an *estimated* distribution for each missing value. It is common to use the *sample* distribution of the responding units as the distribution (or deck) to draw from. Using a simple hot deck method as an example, in Table 17 the missing value for observation 6 in variable x_9 could be imputed with one of the observed values in x_9 (i.e., 10, 20, 30, 40, or 50). Which value that is actually picked from x_9 in observations 1-5 depends on the specifics of the hot deck method. A simple HDI method would be to assign the same probability to all observed values in x_9 (i.e., that each observed value in x_9 has one chance in five to be picked) and then use a random number generator to select one of the five observed values. Observation 6, variable x_9, would then be imputed with the selected value. It should be observed that SRPI belongs to the hot deck class. Unlike the simple example above, SRPI picks the nearest neighbor rather than drawing from the sample at random.

Multiple imputation. Multiple imputation means that one imputes several times creating several complete, imputed data sets. For example, we could use the simple hot deck procedure described above to impute the missing value for observation 6 in x_9 i.e., picking one of the numbers 10, 20, 30, 40, or 50. Let us say we pick the value 50. In a similar manner, we would likewise impute the rest of missing values in Table 17 to obtain a complete data set. Call it data set 1. So far it is simple imputation. Now, multiple imputation is to repeat this procedure creating several complete data sets. Let us assume that next time we draw at random to fill in observation 6 in x_9 we pick the value 10 (different from 50 in data set 1). It is usual to create $M = 3$ or $M = 5$ complete data sets when using multiple imputation. Standard complete-data methods are used to analyse each data set. When the M sets of imputations are repeated random draws under one model for nonresponse, the M complete-data inferences can be combined to form one inference that properly reflects uncertainty due to nonresponse under that model. For details and equations on how to form one inference from the M data sets see Rubin [31]. It should be observed that multiple imputation requires that the basic imputation method draws values to be imputed at random from a distribution. Therefore, SRPI cannot be combined with multiple imputation since it would pick the same value each time for a given missing value.

There are some additional methods that seemed less applicable to our case. Interested readers are referred to [22].

3.3 MAR and MCAR

If the missing data are *not* missing at random, the data analysis may lead to biased results unless the analysis method is able to correct the bias caused by nonrandom missing data. Little and Rubin [22] distinguish between two types of random missing data, Missing Completely At Random (MCAR) and Missing At Random (MAR).

MCAR. Let X be the random variable under study. If $P(X \mid x \text{ missing}) = P(X \mid x \text{ observed})$, then the distribution of X is not affected by missing values. In other words, the probability that, say, Interfaces is missing (or observed) is the same for all projects regardless of the number of Interfaces or the number of Users, Sites, Modules, and Effort. In this case the observed values of Interfaces form a random subsample of the sampled values of Interfaces. This means that the probability that Interfaces = 3 is missing equals the probability that Interfaces = 4 is missing and, so, on for any value of Interfaces.

MAR. Let X be the random variable under study, and let Z be a set of predictor variables. If $P(X \mid x \text{ missing}, Z) = P(X \mid x \text{ observed}, Z)$, then the distribution characteristics of X is conditional on a set of predictor variables. That is, the distribution of X is not affected by missing values for $X \in Z$. In other words, the probability that Interfaces is missing (or observed) depends on the number of Users, (or Sites, Modules, or Effort) but is independent of the number of Interfaces. For example, if there are more missing values for Interfaces in projects with few Users than in projects with many Users, then the data are still MAR but no longer MCAR. Thus, the MAR condition is weaker than the MCAR condition.

Sampling-based methods such as MI and SRPI assume that the data are MCAR whereas model-based methods like FIML only assume the data are MAR [22]. Studies of Muthen et al. [26] and Little and Rubin [23] also suggest that the use of FIML will reduce bias even if the MAR condition is not strictly met. That is, FIML estimates are consistent and efficient even if the MAR condition is not strictly met. Unfortunately, there are no easy ways to find out *empirically* whether a sample distribution is MCAR, MAR, or not missing at random. A priori knowledge is therefore necessary in order to decide on this issue. For example, if many of the projects that do not implement any EDIs do not bother to report EDI = 0, we cannot envisage any tests that would help us discover this anomaly. Rather, we have to know that EDIs are not missing at random.

4 Method of Approach

The method of approach generally is a two-stage procedure. First, an MDT is applied to the original data set with missing data resulting in a complete-case data set. Next, an effort prediction model is constructed on the various data sets by applying ordinary least squares (OLS) regression analysis. The data sets are termed LD, MI, and SRPI, respectively, and the prediction models are termed LD OLS, MI OLS, and SRPI OLS, respectively. Unlike LD, MI, and SRPI, FIML is a one-stage procedure where the model is constructed directly on the existing, original data set without modifying it in any manner first. Therefore, the term FIML applies to the data set as well as to the model.

4.1 Model Specification

We determined the model, i.e., the subset of predictor variables, before applying the LD, MI, SRPI, and FIML methods. The subset of predictor variables considered most important was determined primarily by expert knowledge. (One of the authors is an experienced ERP project manager.) Since nobody knows the "true" model, we also applied best subset regression to confirm the model suggested by the expert. Best subset regression was applied to all the available, original data.

Alternatively, we could have applied best subset regression to each data set after having applied the MDT (LD, MI, SRPI) rather than to the original data set to select the model.

Each alternative has its pros and cons. In the first case, the same model is used on all the imputed data sets. In the other case, we could potentially get different models for each data set. This might be interesting to a practitioner since the practitioner is not *that* sure he has found the best model. Therefore, this would aid the practitioner in his exploratory data analysis aiming to find the model with the most explanatory power. Unfortunately, this would make the comparison of the imputation techniques, which is the focus of this paper, less meaningful since there would be many confounding factors.

4.2 MDT Method

We applied LD the usual way. That is, we removed all the incomplete cases.

For MI, we applied "naïve" imputing as well as imputing where we restricted the fraction of imputed values per variable to 10 percent of the observed variables. By "naïve" imputing, we mean that we did not impose any upper limit on how large a fraction to impute for each variable. The reason for choosing this double approach is to better demonstrate some of the issues related to MI. We find it useful to do this because MI is so widely applied and, so tempting to use, and abuse, due to its simplicity. (We discuss MI more in Section 7).

For SRPI, we imposed the restriction that no cases have more than one missing value in order to be conservative in the use of SRPI. In applying SRPI, we used four out of the five total variables (Users, Sites, Interfaces, Modules, Effort) as matching variables to impute the fifth and missing, variable. For example, when imputing Interfaces, we used Users, Sites, Modules, and Effort as matching variables. We did not fill in any imputed values until all variables were imputed so as not to use a case with an imputed value as a matching case.

As for FIML, the issue of filling in artificial values in the data holes is irrelevant.

4.3 MDT Evaluation Criteria

The *robustness* to bias (degree of nonrandom missing data) of the MDT was determined based on findings in previous studies in statistical literature.

To investigate if the MDT *introduces* a bias, we compared the median, mean, and standard deviation of each variable in the original data set with the data sets resulting from the application of an MDT. For example, we compared the distribution of the variable Users in the LD data with the same variable in the original data set, the MI data with the original, and the SRPI data with the original. For FIML, this comparison is not relevant.

We used two-tailed, two-sample, t-tests to compare the mean of two and two distributions at a time. This t-test tests whether two distributions have an equal mean or not.

We used a chi-square test to compare the standard deviation of two distributions [21]. This test, tests the difference between a sample variance and an assumed population variance. The original data set was assumed to be the population. The test assumes a normal distribution.

Even though these tests may seem simplistic, it is a common procedure in many statistical studies on MDTs. See, e.g., [9]. The idea is as follows: If the distribution of Users in the LD data set is similar (in terms of mean and standard deviation) to the distribution in the original data set, we infer that the MDT (in this case LD) has not introduced a bias in the LD data set.

The degree of information loss prevention was evaluated by comparing the number of complete cases for each data set with the total number of (incomplete) cases in the original data set.

To assess if the ERP data are missing completely at random or not, we relied on expert knowledge. See Section 2. A priori knowledge is necessary in order to decide on this issue. See also Section 3.3.

4.4 Selection of Cases for Regression Analysis

We used all available complete cases in each imputed data set for the regression analysis. That is, we applied regression analysis to all complete cases in the LD, MI, and SRPI data sets. For FIML, it is not an issue whether the cases are complete or incomplete because no rows are removed and no artificial values are imputed.

This implies that the LD OLS, MI OLS, and SRPI OLS regression models were constructed on data sets of unequal size. This reduces the precision of metrics like R2 that require samples of equal size to be fully reliable. However, R2 is still informative as long as one is aware of a somewhat reduced precision.

4.5 Regression Model

We used ordinary least squares (OLS) regression analysis to construct models on the LD, MI, and SRPI data sets. We term these models LD OLS, MI OLS, and SRPI OLS, respectively. We applied a log-log model rather than a linear regression because the log-log model best fulfilled the regression assumptions, notably the homoscedasticity assumption. In transforming to the log-log model, we first added "1" to all variables having minimum value equal to "0."

We performed residual analysis and outlier detection to better identify wild values. In theory, wild values should be replaced with "missing" before doing anything else with the data such as imputing. In practice, however, it may be hard to decide if a value is wild or not. Residual analysis and outlier detection may help you decide whether you think it is a wild value or not. Consequently, this part of the data screening task was done both before as well as after applying the MDTs.

4.6 Prediction Model Evaluation Metrics

As evaluation metrics for the effort prediction model we applied the following metrics.

The t-values (or alternatively p-values) of predictor variable coefficients were used to measure the efficiency of the model (i.e., how close the sample model is to the true model) and the effect of each predictor variable on the response variable.

R^2 was used to assess the overall goodness of fit. It should be observed that R^2 is not ideal to compare models constructed on samples of different size since it requires samples of equal size. However, it is still useful to use it to confirm that the models converge. If the R^2 values are similar, it is a valuable piece of information even if we cannot infer that a model with $R^2 = 0.7$ is better than a model with $R^2 = 0.6$.

The accuracy of the models was evaluated using the mean magnitude of relative error, MMRE, the de facto standard in software engineering for assessing prediction systems. MRE is defined as follows [10]: (z = actual, y = estimate)

$$MRE = \left| \frac{z}{z - y} \right|.$$

Since we applied a log-log regression model, we used the following formula to calculate MRE.

$$MRE = \left| 1 - e^{-residual} \right|.$$

The derivation of this formula is provided in Appendix B

5 Related Work

To our knowledge, there are only two papers that empirically have evaluated the SRPI approach, none of them applied to software engineering. To our knowledge, no paper in any discipline including statistical science has applied nor empirically evaluated the FIML approach on real data.

In software engineering, two papers have been published on MDTs [13], [37]. Both of these studies have applied *sampling-based, hot deck type,* imputation techniques. One of these studies combined hot deck with multiple imputation. No papers have applied *model-based* missing data techniques like FIML in software engineering. Our study, therefore, complements the two other studies on MDTs in software engineering by investigating MDTs other than hot deck and multiple imputation methods. The two software engineering studies are presented at the end of this section.

Brown [9] assessed the efficacy of five imputing methods in the context of structural equation modeling. The methods assessed were LD, pairwise deletion, MI, hot-deck imputation, and SRPI. He found that SRPI provided the least bias.

Gold and Bentler [14] compared four imputing methods, the RBHDI (resemblance-based hot-deck imputation, which is similar to SRPI), the ISRI (iterative stochastic regression imputation). The third and fourth methods are case-based maximum likelihood methods based on different assumptions of the data-generating model. The maximum likelihood methods seem to be superior when the assumptions of the distribution of the population are met and the sample size is sufficiently large. Gold and Bentler conclude that for small samples and moderate to large proportion of missing data the SRPI outperforms maximum likelihood-based methods.

Browne [8] studied LD, PD, MI, and FIML by Monte-Carlo simulations in the factor analytical context. He found that FIML was superior to LD, PD, and MI.

Emam and Birk [13] applied *hot deck multiple imputation* to analyse software process performance data. They have argued well for applying multiple rather than simple imputation. We do not know, however, if the particular hot deck MDT is an appropriate MDT in their case since vital information has not been reported. They did not report any summary statistics of the data. This is vital to assess the degree of confidence one may have in the final results. We do not know whether a small, or a large, fraction of the data were missing in the original data set and in which variables the data were truly missing. When an MDT is used, it is vital to report at a minimum the number of observed and missing values for each variable plus the pattern of missingness in order to assess if the MDT is appropriate.

Emam and Birk have imputed the dependent variable (the performance measure) that was collected through a questionnaire. This dependent variable is ordinal. (The values are "Excellent," "Good," "Fair," "Poor," "Don't Know.") The "Don't know" responses were treated as missing values and consequently imputed. It seems questionable whether it is correct to treat a "don't know" response as a randomly missing nonresponse in their case. Also, it seems they have not made a distinction between truly missing values and "don't know" responses. Unfortunately, it is unclear from reading the paper whether there were any truly missing values. Intuitively, we would assume there might be some bias in the "don't know" responses. In our experience, it is more likely that low performers ("Poor") refuse to respond than the successful high flyers ("Excellent"). If you do not know whether you performed well or badly and report "don't know," it is unlikely you are an "Excellent" high performer because the better you are, the more you know, including your own performance. We therefore miss a discussion on why it may be justified to treat "don't know" as equal to a truly randomly missing value.

Strike et al. [37] evaluated several MDTs in the context of software effort prediction. The MDTs evaluated were LD, MI, and eight different types of hot-deck imputation. They simulated various patterns of missingness in an existing data set by replacing some values with missing values. This is an excellent idea since you then know the true answer. The MDTs were evaluated by measuring the accuracy of the various effort prediction models relative to the accuracy of the true model. Their results indicate that all MDTs perform well and that the simplest MDT, LD, is a reasonable choice.

6 Results

6.1 MDT Results

From Table 2, we observe that LD wastes most information. This is as expected. The number of cases is reduced from 176 to 87. Still, the number of complete cases is sufficient for regression analysis. As a rule of thumb, when using OLS regression, one should have $n > 10^*k$ where n is the number of projects and k is the number of predictor variables. Thus, for the ERP data set with four predictor variables the LD data set with $n = 87 (87 > 10^*4)$ is sufficient.

The naïve MI wastes no information, naturally. We observe that SRPI wastes less information than "MI 10 percent." That is, by restricting SRPI to impute cases with maximum one missing value per case, and similarly, restricting MI to impute maximum 10 percent missing values per variable. FIML does not waste any information either. However, using the number of complete cases as a criterion is not applicable in the case of FIML.

TABLE 2
Number of Complete Cases per MDT

Method	N
LD	87
MI naïve	176
MI 10%	109
SRPI	137
FIML	N/A

Regarding LD, the large reduction is mostly due to missing values in the response variable, Effort. Descriptive statistics for the LD data set is given in Table 3.

Comparing means and standard deviations, the results indicate that the LD data set is a random subsample of the total sample. See Table 4 and Table 5. The p-values in Table 4 supports this initial finding (all $p > 0.10$). Similar for the $\chi^2\text{-values}$ in Table 5 (all $\chi^2 < 107$, the critical value when $N = 87$).

The "10 percent MI" method does not introduce a significant bias in the data. Comparing Table 1 and Table 6, we observe that the mean, the median and the standard deviation are similar for Effort. We also performed formal tests for the mean and standard deviation for all variables that confirmed the observation. (Not reported).

The "naïve MI" method introduces a serious bias. Comparing Table 1 and Table 7, we observe that the mean for Effort is the same, naturally. The median and the standard deviation for Effort, however, have changed significantly. (Testing the significance of the standard deviation, we found $|t| = 4.68$ where $t = 1.96$ at the 5 percent significance level). Especially, we observe that the mean and median Effort have become identical in the naïve MI data set. This happens when a large fraction of values is imputed. Probably, the most important observation is that the standard deviation decreases. Especially, we observe this for Effort where a large fraction of values has been imputed (41 percent imputed). This is as expected since MI distorts the distribution by putting all the imputed values at the mean. The naïve MI data set shows a pronounced peak at the mean in a histogram of the variable Effort. (The histogram is not reported). Therefore, it seems reasonable to restrict the fraction of imputed values per variable as we have done to e.g., 10 percent.

The SRPI method does not introduce any observable bias in the data compared with the original data. By comparing Table 1 and Table 8, we observe that the mean, the median and the standard deviation are all similar for the original and SRPI data sets. We also performed formal tests for the mean and standard deviation that confirmed the observation. (Not reported). For Interfaces, 11 cases were imputed, i.e., $<< 10\%$. For Effort, 38 cases were imputed, i.e., close to 20 percent. Still, the mean, median and standard deviation for Effort for the SRPI data set is very similar to the original data set.

TABLE 3
Descriptive Statistics of LD Data Set

Variable	N	Mean	Median	StDev	Min	Max
Users	87	966	300	2348	6	17000
Sites	87	29	5	73	0	500
Interfaces	87	28	13	48	0	270
Modules	87	5.2	5.0	2.6	1	13
Effort	87	10791	6000	14881	102	111420

TABLE 4
P-Values of 2-Sample t-Tests for LD Data Set

Variable	p-value
Users	0.32
Sites	0.27
Interfaces	0.26
Modules	0.82
Effort	0.98

Unlike the sampling-based imputation methods MI and SRPI, the model-based FIML does not introduce any bias since no artificial values are introduced into, nor removed from, the data set.

6.2 Regression Results

LD OLS. After residual analysis resulting in removal of six cases, the LD OLS is as given in Table 9. Residual analysis after removal of the outliers confirmed that the residuals are normally distributed ($\text{p-value} = 0.5$ of the Anderson-Darling normality test) and that all standardized residuals are less than 2. Furthermore, there is no significant multicollinearity (VIF ≤ 2). (For the Anderson-Darling test and the Variance-Inflating Factor (VIF) test, see [25], [33] and [25], [15], respectively). All predictor variables have a significant effect on effort ($p < 0.01$). The residuals are reasonably homoscedastic. (Plot of residuals versus fits. Not reported.)

10 percent MI OLS. After residual analysis resulting in removal of five cases, the "10 percent MI OLS" regression equation is as given in Table 10. Residual analysis after removal of the outliers confirmed that the residuals are normally distributed $\text{p-value} = 0.36$ of the Anderson-Darling normality test) and that all standardized residuals are less than 2. Furthermore, there is no significant multicollinearity (VIF ≤ 2). The residuals are reasonably homoscedastic. (Plot of residuals versus fits. Not reported.) The predictor variables LN(Users), LN(Interfaces), and LN(Modules) have a significant effect on effort ($p < 0.04$).

Naïve MI OLS. After residual analysis resulting in removal of nine cases, the naïve MI OLS regression equation is as given in Table 11. Residual analysis after

TABLE 5
$\chi^2\text{-values}$ for LD Data Set

Variable	χ^2
Users	58
Sites	15
Interfaces	22
Modules	80
Effort	90

TABLE 6
Descriptive Statistics of 10 Percent MI Data Set

Variable	N	N*	Mean	Median	StDev
Users	176	0	1297	338	2867
Sites	176	0	45	7	171
Interfaces	154	22	39	19	90
Modules	176	0	5.2	5.0	2.7
Effort	109	67	10850	7560	13892

TABLE 8
Descriptive Statistics of SRPI Data Set

Variable	N	N*	Mean	Median	StDev
Users	176	0	1297	338	2867
Sites	175	1	45	7	172
Interfaces	150	26	39	15	93
Modules	176	0	5.2	5.0	2.7
Effort	137	39	10972	6000	13899

removal of the outliers confirmed that the residuals are normally distributed ($p\text{-value} = 0.51$ of the Anderson-Darling normality test) and that all standardized residuals are less than 2. Furthermore, there is no significant multicollinearity ($\text{VIF} \leq 2$). The residuals are reasonably homoscedastic. (Plot of residuals versus fits. Not reported.) The predictor variables LN(Users), LN(Interfaces), and LN(Modules) have a significant effect on effort ($p < 0.02$). Furthermore, we observed that the values of the coefficients are quite different for the LD OLS and naïve MI OLS models.

SRPI OLS. The SRPI regression equation is as given in Table 12. 137 cases were used and five outliers removed resulting in 132 cases. Removal of outliers did not improve the normality of the distribution of residuals significantly ($p\text{-value} = 0.08$ of the Anderson-Darling normality test). There is no significant multicollinearity ($\text{VIF} \leq 2.1$.) The residuals are however reasonably homoscedastic. (Plot of residuals versus fits. Not reported.) All predictor variables have a significant effect on effort ($p < 0.032$) but less significant than for LD.

FIML model. The FIML regression equation is given in Table 13. Six cases were removed. The regression is based on the remaining 170 cases. We observe that FIML confirms the other models. That is, Users, Interfaces and Modules are highly significant whereas there is some doubt about Sites. Sites is the least significant variable in all models. We observe that the FIML model is overall more efficient than the other models (t-values around five except for Sites). We also observe that LD OLS is more efficient than 10 percent MI OLS, Naïve MI OLS, and SRPI OLS.

Regarding the prediction accuracy, the LD OLS has the highest accuracy in terms of MMRE (48 percent) and FIML the lowest (74 percent). See Table 14.

6.3 Summary of Results

We have applied two kinds of research methods to compare MDTs: empirical methods as well as survey methods. Our *empirical* evaluation is a simple one based on comparing the sample means and the variances to compare distributions before and after applying the MDT. The *survey* (in Section 3) of statistical literature reports some important findings obtained in statistical science. In particular, it is vital for the correct application of MDTs to know the assumptions they make with respect to MCAR, MAR and nonrandom missing data.

TABLE 7
Descriptive Statistics of Naïve MI Data Set

Variable	N	Mean	Median	StDev
Users	176	1297	338	2867
Sites	176	45	7	171
Interfaces	176	39	20	85
Modules	176	5.2	5.0	2.7
Effort	176	10850	10850	10913

Regarding the empirical evaluation, the results may be summarized as follows:

The results confirm that we have found a reasonably correct model. The results in terms of t-values and R2(adj) are similar across the models (LD OLS, MI OLS, SRPI OLS, and FIML). "If the model is correctly specified, different estimators should have similar values asymptotically. If these values are not sufficiently similar, the model is not correctly specified," [38]. We observe, however, that Sites is questioned more than the other variables. In particular, FIML does not consider this variable as equally significant. Sites varies from $t = 1.77$ to $t = 2.77$ whereas the other variables vary between $t = 2.65$ and $t = 6.46$. (We have disregarded the t-values of Naïve MI). Thus, the lowest t-value of any of the other variables is comparable to the highest t-value for Sites. This is not in total disagreement with expert knowledge. Beforehand, we were rather confident about Users, Interfaces, and Modules but somewhat less confident about Sites. The reason is that it is harder to define good counting rules for "a Site" than for User, Module, and Interface in the context of effort prediction. It is, however, beyond the scope of this paper to go into more detail on this issue.

There possibly is a small bias in the missing data. The regression results indicate that there might be some bias in the data although the 2-sample t-tests of the variable distributions did not reveal anything. This is observed by comparing the LD OLS and the FIML models in terms of t-values and MMRE where we observe a discrepancy between them.

TABLE 9
Regression Coefficients of LD Data Set

Predictor	Coef	SE Coef	t	p	VIF
Constant	4.85	0.30	16.3	0.000	
LN(Users)	0.275	0.063	4.35	0.000	2.0
LN(Sites)	0.153	0.055	2.77	0.007	1.3
LN(Interfaces)	0.289	0.058	4.94	0.000	1.3
LN(Modules)	0.732	0.164	4.46	0.000	1.4
		R2(adj)	72%		

TABLE 10
Regression Coefficients of "10 Percent MI" Data Set

Predictor	Coef	SE Coef	t	p	VIF
Constant	5.260	0.305	17.3	0.000	
LN(Users)	0.220	0.064	3.47	0.001	2.0
LN(Sites)	0.119	0.057	2.08	0.040	1.4
LN(Interfaces)	0.181	0.057	3.16	0.002	1.3
LN(Modules)	0.822	0.162	5.09	0.000	1.4
		R2(adj)	59%		

TABLE 12
Regression Coefficients of SRPI Data Set

Predictor	Coef	SE Coef	t	p	VIF
Constant	5.068	0.318	15.9	0.000	
LN(Users)	0.185	0.070	2.65	0.009	2.1
LN(Sites)	0.128	0.059	2.16	0.032	1.5
LN(Interfaces)	0.393	0.061	6.46	0.000	1.3
LN(Modules)	0.674	0.169	4.00	0.000	1.4
		R2(adj)	60%		

Regarding the survey (in Section 3), the findings may be summarised as follows:

FIML is most resistant to bias in the missing data. As stated in Section 3.3, "MAR and MCAR," other studies have concluded that FIML will reduce bias even if the MAR condition is not strictly met whereas the other MDTs (LD, MI, SRPI) assume the data are MCAR. FIML, therefore, likely is the least biased model. Therefore, if there is a slight bias, FIML is likely closest to the "true" model. It should be observed that this is not a result of our own but rather an assertion based on the properties of this method.

Given that we have a higher confidence in the FIML model than in the other models, the results suggest that a realistic, or true, prediction accuracy is around $\mathrm{MMRE} = 70\%$. It is important to observe that the accuracy of the regression model may be seriously overestimated by the LD procedure ($\mathrm{MMRE} = 48\%$). We argue that the true prediction accuracy likely is closer to the FIML accuracy than to any of the others. However, it may be objected that we have too few complete observations to be more confident in the FIML model than in the other models. That is, is $\mathrm{N} = 87$ close enough to $\mathrm{N} = 100$ or not.

7 Discussion of MDTs

In this section, we discuss the following issues:

- Model-based versus sampling-based methods,
- MI versus SRPI, with a focus on patterns of missingness lending itself to either method.
- FIMLs multivariate normal distribution assumption.
- The MCAR assumption of LD, MI, and SRPI.

7.1 Model-Based vs. Sampling-Based MDTs

There is a significant difference in perspective between model-based and sampling-based methods. The model-based perspective is motivated by a desire to accurately estimate population parameters. From this perspective, the appropriate performance criterion is the extent to which the population parameter estimates from each incomplete, original data matrix reproduce the population parameters.

The sample-based perspective is motivated by a desire to fill in values in a data matrix, thus enabling the resulting data matrix to be used in any subsequent data analysis. For example, if the aim is not to construct a regression-type effort prediction model, but rather a CART-type or any other type requiring complete-case analysis methods, e.g., cluster analysis, a model-based MDT like FIML is not an option. In such cases, one must fill in missing values, or alternatively remove incomplete observations and, therefore, recur to MDTs like LD, MI, and SRPI (or possibly other MDTs).

In our case, the aim is to build an effort prediction model as close to the true model as possible. Our primary aim is a good prediction model rather than a complete data set. We are not interested in filling in the missing values for the sake of filling in missing values just to get a larger data set since the LD data set is large enough to apply OLS regression. The LD data set has 87 complete observations (see Table 2) which is larger than the required, rule-of-thumb $4^{*}10 = 40$ complete observations. Also, the original, incomplete data set has close to 100 complete observations (and 176 including the incomplete). Thus, it likely is sufficient to apply FIML. In our case, FIML seems, therefore, the more appropriate method with LD coming second.

7.2 MI vs. SRPI

The advantage of MI is that it is fast and simple which likely is the reason for its popularity. However, when the fraction of missing cases is significant for a given variable, such as Effort in our case, MI biases the distribution since all the missing values are imputed at the centre of the distribution.

TABLE 11
Regression Coefficients of Naïve MI Data Set

Predictor	Coef	SE Coef	t	p	VIF
Constant	6.99	0.229	30.5	0.000	
LN(Users)	0.110	0.047	2.35	0.020	2.0
LN(Sites)	0.067	0.043	1.54	0.126	1.5
LN(Interfaces)	0.153	0.046	3.33	0.001	1.2
LN(Modules)	0.457	0.119	3.84	0.000	1.4
		R2(adj)	34%		

TABLE 13
Regression Coefficients of FIML Data Set

Predictor	Coef	SE Coef	t	p
Constant	4.82	0.271	17.8	0.000
LN(Users)	0.286	0.058	4.94	0.000
LN(Sites)	0.093	0.053	1.77	0.078
LN(Interfaces)	0.314	0.056	5.64	0.000
LN(Modules)	0.746	0.145	5.14	0.000
		R2(adj)	0.76	

TABLE 14
MMRE for Regression Models

Model	MMRE(%)
LD	48
10% MI	65
Naïve MI	61
SRPI	68
FIML	74

TABLE 16
A Pattern of Missing Data Where the SRPI Method Might Be Appropriate

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
*									
	*								
		*							
			*						
				*					
					*				
					*				
					*				
					*				
					*				

We feel that the percentage of missing cases should therefore not exceed 5 to 10 percent for this method to be used with some degree of confidence. This ought to be verified by simulation since there is no theory to assist on this issue. Even though MI is a simple method, and generally not recommended, we may envisage patterns of missing data where it might add value. Consider for example the case depicted in Table 15. The asterisks denote missing values. In this case, we observe that LD would reduce the data set to zero observations. We further observe that for each column, the percentage of missing cases is 10 percent, which is within an acceptable ratio of missing to nonmissing values. For this pattern of missing data, the simple MI method may greatly reduce the loss of information without introducing a significant bias in the data. On the other hand, it would be dangerous to use MI in the case depicted by the pattern in Table 16 since the distribution of x_6 would be highly distorted after imputation.

Contrary to MI, SRPI seems equally well suited to both patterns of missing data. Consider first the pattern in Table 15. Let us assume that we want to impute x_1 first. In this case, we have the choice of selecting any subset of the variables x_2 to x_{10} as matching variables. The same goes for imputing x_6 in Table 16. That is, imputing x_6 does not cause any more trouble than imputing the other variables. The main objection against SRPI is that it requires a thorough knowledge of the data with regard to selecting the matching variables for each variable to be imputed. Also, the more variables we select as matching variables, the fewer the subset of potentially similar cases. Consider for example imputing x_1 using x_2 to x_{10} as matching variables. In this case, there would be no matching cases. That is, there are no rows in Table 16 that are complete in x_2 to x_{10} as well as in x_1. At the other extreme, we could select only x_7 to x_{10} as matching variables to impute x_1 in row 1. In this case, all rows 2 through 9 may be considered for similar case calculations.

In our analysis, we reduced the data set by restricting the SRPI imputation to cases with one missing variable. Of course, the SRPI method may be used to impute more than one missing variable using the same set of matching variables. However, it is a difficult trade-off between size and reliability, where judgement must be exercised. All in all, SRPI is therefore not a fast and easy expedient.

We observe that in the special cases depicted in Table 15 and Table 16, the LD method would incur a 100 percent loss of information provided we want to use all the 10 variables.

The pattern of missing data in the ERP data set mostly resembles that in Table 16. SRPI seems, therefore, a more appropriate method than MI in this case. We also believe that SRPI generally is to be preferred over MI because it likely introduces less bias than MI.

Note that the SRPI method obtains imputed values from similar projects by using the least squares criterion as the similarity measure, which produces exactly the same results as using the Euclidean distance. Therefore, SRPI is a kind of *imputation by analogy* approach as implemented in tools like ANGEL [34]. However, unlike ANGEL, SRPI is more robust and less vulnerable to outliers as it uses the predicted values for statistical purposes, only, whereas ANGEL uses the similar cases to predict *single* projects.

TABLE 15
A Pattern of Missing Data Where the MI Method Might Be Appropriate

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
*									
	*								
		*							
			*						
				*					
					*				
						*			
							*		
								*	
									*

TABLE 17
A Pattern of Missing Data Similar to the ERP Data Set

ID	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1									10	100
2									20	*
3									30	*
4									40	*
5									50	*
6									*	*
7									*	*
8									*	*
9								*	*	*
10							*	*	*	*

TABLE 18
p-Values of Anderson-Darling Normality Test on All Variables

Variable	p-value
LN(Users)	0.210
LN(Sites)	0.000
LN(Interfaces)	0.037
LN(Modules)	0.000
LN(Effort)	0.197

7.3 The Multivariate Normal Distribution Assumption of FIML

Two of the variables (LN(Sites), LN(Modules)) in the log-log model do not exhibit a univariate normal distribution. On the other hand, LN(Users) and LN(Effort) are normal. LN(Interfaces) is somewhere in between. See Table 18. We therefore know that the data do not exhibit a *multivariate* normal distribution, as this requires that all variables exhibit a univariate normal distribution. (Mardia [24] presents a multivariate normality test.) Fortunately, FIML is robust to data that do not comply completely with the multivariate normal distribution requirement [7]. Therefore, it may still be appropriate.

7.4 The MCAR and MAR Assumptions

The sampling-based MDTs (LD, MI and SRPI) assume the data are MCAR. This assumption seems reasonable. (See evaluation of the ERP data in Section 2.) Regarding Effort (which has most missing values), it is unlikely that small projects have a higher (or lower) probability than large projects of reporting Effort. This reasoning applies to Interfaces as well.

One may suspect that less successful projects report to a lesser extent than more successful projects. Following this line of thought, one would assume that large, complex projects report to a lesser extent than small, not so complex, projects because they have a higher probability of not being successful. Were this the case, there would be a bias caused by the missing data, and the data would not be MCAR. Fortunately, large projects are managed by the most experienced project managers whereas the smaller projects more often are headed by junior managers. It is therefore not obvious whether small or large projects have the higher success rate and, thus, whether there are more missing data in small or in large projects. We believe therefore (based on expert knowledge of the data) that missing data are equally found among small and large projects and that the data therefore are MCAR. However, our empirical results (in Section 6.1) seem to suggest a slight bias (nonrandom missing data).

The only variable that might not be MCAR is EDI. Many projects did not implement EDI eBusiness solutions. Therefore, there is a large fraction of projects with $\mathrm{EDI} = 0$. It is possible that projects with $\mathrm{EDI} = 0$ are more sloppy in reporting this variable than projects with $\mathrm{EDI} > 0$. We have not been able to check this, though, and it does not matter in this study since EDI is not a predictor variable in the model.

Regarding the MAR assumption of FIML, we find it unlikely that any ERP data are MAR. We believe that they are either missing *completely* at random (MCAR) or are nonrandom. Thus, the MAR condition seems a bit artificial, and we question the practical use of it for ERP software data.

8 Conclusion

In this paper, we have investigated missing data techniques (MDTs) with the aim of constructing more reliable software effort prediction models. (By "reliable," we mean "closest to the truth." The "true model" would be constructed on the true, population data set, i.e., on all past and future projects). We have investigated whether it may be worth the extra effort to apply MDTs other than the default listwise deletion (LD), and whether LD is always appropriate or not. LD is routinely used in the usual two-stage procedure that consists of removing incomplete observations before applying, e.g., OLS regression analysis to the remaining complete observations to construct a prediction model. We have investigated the MDTs empirically using an ERP data set as well as by surveying statistical literature. The latter complements the empirical study and enables us to provide some more general advice. Our recommendations are as follows:

Use FIML if you have enough data to afford it. Our evaluation suggests that FIML is the best choice for constructing an effort prediction model when there are missing data because FIML is somewhat more resistant to bias (nonrandom missing data) in the data caused by missing values than the other MDTs.

Use imputing-type MDTs (MI, SRPI) only if you desperately need more data. Do not impute just to look good. If FIML cannot be used because the data set is too small, we recommend LD combined with a regression model unless it results in a too small data set. Thus, we recommend MI and SRPI, again combined with a regression model, only if they contribute to making an otherwise too small data set big enough to carry out regression analysis.

For an ERP data set with four predictor variables, we would therefore use FIML provided the number of observations, N, exceeds 100. Otherwise, we would use LD provided it leaves us with $N > 40$. We would use SRPI or MI only if LD leaves us with $N < 40$ and at the same time SRPI or MI helps us achieve $N > 40$. The choice of MI versus SRPI would depend on the pattern of missingness.

Don't use LD if you suspect the data are not missing completely at random, and be prepared to argue that the data are MCAR when applying LD. If one suspects that the observations with missing values differ systematically from the complete observations, LD is dangerous. (Actually, no MDTs will correct such a bias satisfactorily.) More dangerous, no MDTs or tests are able to detect such a bias. One must, therefore, rely on expert knowledge to judge on the issue of randomness. Also, if the fraction of missing data is large, and one suspects nonrandom missing data, we discourage using any MDTs at all, including LD. That means, we discourage using such a data set at all because the results of a data analysis would be highly biased and, therefore, misleading. In this circumstance, the only solution is to somehow mend the holes with the true values (e.g., by calling up nonrespondents once more). However, if the fraction of missing data is small, say, less than 5 percent, LD (and other MDTs) can be used without introducing large errors [32].

Use FIML if your aim is to construct a regression-type effort prediction model. It should be observed that FIML is applicable and an option when the aim is to produce a *regression-type* prediction model based on historical data containing missing values. If one needs to apply *complete-case* statistical techniques other than regression analysis, FIML might not be an option. For example, when building CART models for software effort prediction, one applies some kind of cluster analysis within CART. Cluster analysis techniques generally are complete-case techniques. Thus, they require that missing values actually be filled in or alternatively, that the observations with missing values be removed. Since LD, SRPI, and MI are MDTs resulting in complete data sets, they may be more suitable when constructing, e.g., CART models.

To summarize, analysis of incomplete data sets is an increasingly important issue in software engineering as software engineering extends its branches to subdisciplines like *empirical* software engineering and software *metrics*. Unfortunately, as we view it, there still are several obstacles both with model-based (like FIML) as well as with sampling-based MDTs (like LD, MI, SRPI). Regarding sampling-based MDTs, Dempster and Rubin [12] warn us:

> "The idea of imputation is both seductive and dangerous. It is seductive because it can lull the user into the pleasurable state of believing that the data are complete after all, and it is dangerous because it lumps together situations where the problem is sufficiently minor that it can be legitimately handled in this way and situations where standard estimators applied to the real and imputed data have substantial biases."

APPENDIX A

FIML Procedure

Let p be the number of variables and N the number of cases (observations). We assume that the vector

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_p \end{bmatrix}$$

has a multivariate normal distribution with mean μ and covariance matrix Σ. If $y_1, y_2, \ldots, y_{\mathrm{N}}$ is a random sample of the vector y, the data matrix is a $\mathrm{N} \times p$ matrix. This matrix can have missing values, i.e., specific elements of the vectors $y_1, i = 1, 2, 3, \ldots, \mathrm{N}$ may be unobserved. Consider the following example. Let $\mathrm{N} = 7$ and $\mathrm{p} = 3$. The data matrix can look as in Table 19.

The population mean vector and the population covariance matrix are

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} \text{and } \Omega = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}.$$

TABLE 19
Data Matrix

Case	y_1	y_2	y_3
1	y_{11}	y_{12}	y_{13}
2	y_{21}	y_{22}	y_{23}
3	y_{31}	*	y_{33}
4	y_{41}	y_{42}	*
5	y_{51}	y_{52}	y_{53}
6	*	y_{62}	y_{63}
7	*	y_{72}	y_{73}

Let m_i and Ω_i be the population mean vector and covariance matrix for the variables that are *observed* for Case i. These elements (m_i and Ω_i) can be obtained by deleting elements from $\mu\, and\, \Sigma$. For example, in Case 4 where y_3 is missing:

$$m_4 = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \text{and } \Omega_4 = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}.$$

These "reduced" vectors and matrices are used in the estimation process. In the general situation, the log likelihood of case i is defined by [5]:

$$\log l_i = K_i - \frac{1}{2}\log|\Omega_i| - \frac{1}{2}(y_i - m)'\Omega_i^{-1}(y_i - m_i). \quad (2)$$

The log-likelihood for the entire sample (all the non-missing data) is

$$\log L = \sum_{i=1}^{N} \log l_i. \quad (3)$$

Given a model that specifies the vector μ and covariance matrix Σ as *functions* of its parameters imply that $\mu = \mu(\theta)$, and $\Sigma = \Sigma(\theta)$, where θ is the parameter vector to be estimated. The crucial point is that the model is used to *predict* the mean vector μ and the covariance matrix Σ. Therefore, we formulate the equations $\mu = \mu(\theta)$ and $\Sigma = \Sigma(\theta)$. The parameter vector θ is an unknown stochastic quantity that has to be estimated.

Maximum likelihood estimates of θ are obtained by maximizing $\log L(\theta)$. The mean vectors and covariance matrices in (2) are now functions of the unknown parameters θ in the theoretical model. We can think of this as a derivation process where we solve the equation

$$\partial \log(\theta) \frac{\partial}{\theta} = 0.$$

The estimated parameter vector $\hat{\theta}$ is the parameter vector with the maximum likelihood of being responsible for the observed data.

A chi-square statistic is defined as $\chi^2 = F_0 - F_1$, where $F_1 = -2\ln L_0$ and L_0 denotes the log-likelihood value (at convergence) when μ and Σ are restricted according to the theoretical model and $\ln L_1$ denotes the log-likelihood (at convergence) when no restriction are imposed on μ and Σ

[17]. The degrees of freedom are $\frac{1}{2}p(p+1)-t$, where t is the number of free parameters in the model.

APPENDIX B

Calculation of MRE in Log-Log Regression Models

This appendix shows how the formula for calculating MRE is derived when one applies a log-log regression model to predict effort. Suppose the log-log model, with $\underline{y}$ = actual effort, is

$$\ln y = \ln \alpha + \beta \ln X + \ln u$$

Then, predicted effort (or rather the predicted $\ln$-effort) is

$$\hat{\ln}\, y = a + b \ln X. \tag{4}$$

Let the residual be given by

$$residual = \ln y - \ln \hat{y}$$

which is equal to

$$residual = \ln\left(\frac{y}{\hat{y}}\right).$$

This may be transformed to $e^{residual} = \frac{y}{\hat{y}}$ or alternatively $e^{-residual} = \frac{\hat{y}}{y}$. Thus,

$$1 - e^{-residual} = \frac{y - \hat{y}}{y}. \tag{5}$$

By definition, MRE is

$$\mathrm{MRE} = \left|\frac{y - \hat{y}}{y}\right| \tag{6}$$

From (5) and (6) we may restate MRE

$$\mathrm{MRE} = \left|1 - e^{-residual}\right|$$

ACKNOWLEDGMENTS

The authors would like to thank Accenture (formerly Andersen Consulting). This work was partly financed by Accenture's Research Fund in Norway. We would also like to thank the anonymous reviewers for their comments, which resulted in substantial improvements to this work.

REFERENCES

[1] T.W. Anderson, "Maximum Likelihood Estimates for Multivariate Normal Distributions when Some Observations are Missing," *J. Am. Statistical Assoc.,* vol. 52, pp. 200-203, 1957.

[2] A.B. Anderson, A. Basilevsky, and D.P.J. Hum, "Missing Data: A Review of the Literature," *Handbook of Survey Research.* P.H. Rossi, J.D. Wright and A.B. Anderson eds., New York: Academic Press, pp. 415-492, 1983.

[3] J. Anderson and D.W. Gerbing, "The Effects of Sampling Error on Convergence, Improper Solutions, and Goodness-of-Fit Indices for Maximum Likelihood Confirmatory Factor Analysis," *Psychometrika,* vol. 49, pp. 155-173, 1984.

[4] L. Angelis, I. Stamelos, and M. Morisio, "Building a Software Cost Estimation Model Based on Categorical Data," *Proc. METRICS 2001,* pp. 4-15, 2001.

[5] J.L. Arbucle, "Full Information Estimation in Presence of Incomplete Data," *Advanced Structural Equation Modeling, Issues and Techniques.* G.A. Marcoulides and R.E. Schumacker, eds., 1996.

[6] J.L. Arbucle, *Amos User's Guide.* Chicago: SmallWaters, 1995.

[7] A. Boomsma, *On Robustness of LISREL (Maximum Likelihood Estimation) Against Small Samples Sizes and Non-Normality.* Amsterdam: Sociometric Research Foundation, 1982.

[8] C.H. Browne, "Asymptotic Comparison of Missing Data Procedures for Estimating Factor Loadings," *Psychometrika,* vol. 48, no. 2, pp. 269-291, 1983.

[9] R.L. Brown, "Efficacy of the Indirect Approach for Estimating Structural Equation Models With Missing Data: A Comparison of Five Methods," *Structural Equation Modeling,* vol. 1, no. 4, pp. 287-316, 1994.

[10] S.D. Conte, H.E. Dunsmore, and V.Y. Chen, *Software Eng. Metrics and Models,* Menlo Park, Calif: Benjamin/Cummings, Inc., 1986.

[11] T. DeMarco, *Controlling Software Projects: Management, Measurement, and Estimates,* New York: Prentice-Hall, 1982.

[12] A.P. Dempster and D.B. Rubin, *Incomplete Data in Sample Surveys.* W.G. Madow, I. Olkin and D.B. Rubin eds. vol. 2, pp. 3-10, New York: Academic Press, 1983.

[13] K.E. Emam and A. Birk, "Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability," *IEEE Trans. Software Eng.,* vol. 26, no. 6, pp. 541-566, June 2000.

[14] M.S. Gold and P.M. Bentler, "Treatment of Missing Data: A Monte Carlo Comparison of RBHDI, Iterative Stochastic Regression Imputation, and Expectation-Maximation," *Structural Equation Modeling,* vol. 7, no. 3, pp. 319-355, 2000.

[15] D.N. Gujarati, *Basic Econometrics,* London: McGraw-Hill, 1995.

[16] R. Jeffery, M. Ruhe, and I. Wieczorek, "Using Public Domain Metrics to Estimate Software Development Effort," *Proc. METRICS 2001,* pp. 16-27, 2001.

[17] K.G. Jøreskog and D. Sørbom, *LISREL 8.50 Student Edition.* Chicago: Scientific Software Int'l, 2000.

[18] K.G. Jøreskog and D. Sørbom, *LISREL 8 User's Reference Guide.* Chicago: Scientific Software Int'l Inc., 1993.

[19] K.G. Jøreskog and D. Sørbom, *PRELIS 2 User's Reference Guide,* Chicago: Scientific Software Int'l Inc., 1995.

[20] K.G. Jøreskog, Personal Email Correspondence, Apr. 2001.

[21] G.K. Kanji, *100 Statistical Tests.* London: SAGE Publications, 1993.

[22] R.J.A. Little and D.B. Rubin, *Statistical Analysis with Missing Data,* New York: Wiley, 1987.

[23] R.J.A. Little and D.B. Rubin, "The Analysis of Social Science Data with Missing Values," *Sociological Methods and Research,* vol. 18, nos. 2/3, pp. 292-326, Nov. 1989 Feb. 1990.

[24] K.V. Mardia, "Measures of Multivariate Skewness and Kurtosis with Applications," *Biometrika,* vol. 57, no. 3, p. 519, 1970.

[25] *Minitab Statistical Software,* Release 13, State College, Penn.: Minitab, Inc.,www.minitab.com. 2000.

[26] B. Muthen, D. Kaplan, and M. Hollis, "On Structural Equation Modeling with Data that are not Missing Completely at Random," *Psychometrika,* vol. 52, pp. 431-462, 1987.

[27] I. Myrtveit and E. Stensrud, "A Controlled Experiment to Assess the Benefits of Estimating with Analogy and Regression Models," *IEEE Trans. Software Eng.,* vol. 25, no. 4, pp. 510-525, Jul/Aug. 1999.

[28] I. Myrtveit, E. Stensrud, and U. Olsson, "Assessing the Benefits of Imputing ERP Projects with Missing Data," *Proc. METRICS 2001,* pp. 78-84, 2001.

[29] M.C. Neal, *Mx: Statistical Modeling,* second ed., 1994.

[30] M.J. Rovine and M. Delaney, "Missing Data Estimation in Developmental Research," *Statistical Methods in Longitudinal Research: Principles and Structuring Change,* A. Von Eye ed., vol. 1, pp. 35-79, New York: Academic, 1990.

[31] D.B. Rubin, *Multiple Imputation for Nonresponse in Surveys,* New York: Wiley, 1987.

[32] J.L. Schafer, *Analysis of Incomplete Multivariate Data,* Boca Raton: Chapman and Hall, 1997.

[33] S.S. Shapiro and R.S. Francia, "An Approximate Analysis of Variance Test for Normality," *J. Am. Statistical Assoc.,* vol. 67, p. 215, 1972.

[34] M. Shepperd and C. Schofield, "Estimating Software Project Effort Using Analogies," *IEEE Trans. Software Eng.,* vol. 23, no. 12, pp. 736-743, Nov. 1997.

[35] E. Stensrud and I. Myrtveit, "Human Performance Estimating with Analogy and Regression Models: An Empirical Validation," *Proc. METRICS'98,* pp. 205-213, 1998.

[36] E. Stensrud, "Estimating with Enhanced Object Points versus Function Points," *Proc. Cost Constructive Model, (COCOMO '13),* Oct. 1998.

[37] K. Strike, K.E. Emam, and N. Madhavji, "Software Cost Estimation with Incomplete Data," ERB-1071 NRC, http://wwwsel.iit.nrc.ca/~elemam/documents/1071.pdf, also to appear in, *IEEE Trans. Software Eng.*

[38] H. White, "Estimation, Inference, and Specification Analysis," *Econometric Society Monographs,* no. 22, Cambridge Univ. Press, 1994.

Ingunn Myrtveit received the MS degree in management from the Norwegian School of Management in 1985 and the PhD degree in economics from the Norwegian School of Economics and Business Administration in 1995. She is an associate professor in business economics at the Norwegian School of Management. She has also been a senior manager at Andersen Consulting's World Headquarters R&D Center in Chicago. Her research interests include managerial economics, empirical studies, software engineering economics, and software metrics.

Erik Stensrud received the MS degree in physics from the Norwegian Institute of Technology in 1982, the MS degree in petroleum economics from the Institut Francais du Petrole in 1984, and the PhD in software engineering from the University of Oslo in 2000. He is currently an associate professor in technology management at the Norwegian School of Management and a visiting professor at Bournemouth University, England. He also runs his own consulting business. Prior to that, he managed ERP projects and custom software projects for more than 15 years serving with major consultancy companies including Accenture and Ernst & Young. His research interests include software engineering economics and software metrics. He is a member of the IEEE, the IEEE Computer Society, ACM, and the Norwegian Computer Society.

Ulf H. Olsson received the MS degree in mathematics from the University of Oslo in 1981 and the PhD degree in multivariate statistics from the Agricultural University of Norway in 1996. He is an associate professor in multivariate statistics at the Norwegian School of Management. His main research interests are multivariate statistics, econometrics, and measurement theory.