ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THESIS PRESENTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
Ph. D.

BY
Abdalla BALA

IMPACT ANALYSIS OF A MULTIPLE IMPUTATION TECHNIQUE FOR HANDLING MISSING VALUE IN THE ISBSG REPOSITORY OF SOFTWARE PROJECTS

MONTREAL, OCTOBER 17 2013

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Mr. Alain Abran, Thesis Supervisor
Software Engineering and Information Technology Department at École de technologie supérieure

Mr. Ambrish Chandra, President of the Board of Examiners
Department of Electrical Engineering at École de technologie supérieure

Mr. Alain April, Member of the jury
Software Engineering and Information Technology Department at École de technologie supérieure

Mr. Chadi el Zammar, External Evaluator
Kronos (Canada)

THIS THESIS WAS PRENSENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND PUBLIC

ON OCTOBER 16, 2013

AT ECOLE DE TECHNOLOGIE SUPERIEURE

ACKNOWLEDGMENTS

I would like to express my thanks and gratitude to Allah, the Most Beneficent, the Most Merciful whom granted me the ability and willing to start and complete this research.

First, I would like to express my utmost thanks to Dr. Alain Abran, my thesis supervisor, for his motivation, guidance, time and support. His advice continuously opened new opportunities to improve the outcomes of this research; without his patient support this research would have never been executed.

Thanks to the members of my board of examiners for their time and effort to review this thesis and to provide me with their feedback.

I am deeply and forever indebted to my parents Mr. Ali Bala and Mdm. Nagmia for their love, support and encouragement throughout my entire life. I am also very grateful to all my brothers Amer Bala, Mustafa Bala, Nuri Bala and to all my sisters for instilling in me confidence and a drive for pursuing my PhD.

I also would like to thank to those who have helped me and encouraged me at all time during my study: they are Adel Alraghi, Walid Bala, and special thanks to all my colleagues and my friends.

IMPACT ANALYSIS OF A MULTIPLE IMPUTATION TECHNIQUE FOR HANDLING MISSING VALUE IN THE ISBSG REPOSITORY OF SOFTWARE PROJECTS

Abdalla BALA

RÉSUMÉ

Jusqu'au début des années 2000, la plupart des études empiriques pour construire des modèles d'estimation de projets logiciels ont été effectuées avec des échantillons de taille très faible (moins de 20 projets), tandis que seules quelques études ont utilisé des échantillons de plus grande taille (entre 60 à 90 projets). Avec la mise en place d'un répertoire de projets logiciels par l'International Software Benchmarking Standards Group - ISBSG - il existe désormais un plus grand ensemble de données disponibles pour construire des modèles d'estimation: la version 12 en 2013 du référentiel ISBSG contient plus de 6000 projets, ce qui constitue une base plus adéquate pour des études statistiques.

Toutefois, dans le référentiel ISBSG un grand nombre de valeurs sont manquantes pour un nombre important de variables, ce qui rend assez difficile son utilisation pour des projets de recherche.

Pour améliorer le développement de modèles d'estimation, le but de ce projet de recherche est de s'attaquer aux nouveaux problèmes d'accès à des plus grandes bases de données en génie logiciel en utilisant la technique d'imputation multiple pour tenir compte dans les analyses des données manquantes et des données aberrantes.

Mots-clés: technique multi-imputation, préparation des données ISBSG, identification des valeurs aberrantes, modèle d'estimation de l'effort de logiciel, critères d'évaluation.

IMPACT ANALYSIS OF A MULTIPLE IMPUTATION TECHNIQUE FOR HANDLING MISSING VALUE IN THE ISBSG REPOSITORY OF SOFTWARE PROJECTS

Abdalla BALA

ABSTRACT

Up until the early 2000's, most of the empirical studies on the performance of estimation models for software projects have been carried out with fairly small samples (less than 20 projects) while only a few were based on larger samples (between 60 to 90 projects). With the set-up of the repository of software projects by the International Software Benchmarking Standards Group – ISBSG – there exists now a much larger data repository available for productivity analysis and for building estimation models: the 2013 release 12 of this ISBSG repository contains over 6,000 projects, thereby providing a sounder basis for statistical studies.

However, there is in the ISBSG repository a large number of missing values for a significant number of variables, making its uses rather challenging for research purposes.

This research aims to build a basis to improve the investigation of the ISBSG repository of software projects, in order to develop estimation models using different combinations of parameters for which there are distinct sub-samples without missing values. The goal of this research is to tackle the new problems in larger datasets in software engineering including missing values and outliers using the multiple imputation technique.

Keywords: multi-imputation technique, ISBSG data preparation, identification of outliers, analysis effort estimation model, evaluation criteria.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABREVIATIONS

COSMIC	Common Software Measurement International Consortium
EB	Effort Build
EI	Effort Implement
EP	Effort Plan
Eq	Equation
ES	Effort Specify
ET	Effort Test
FiSMA	Finnish Software Measurement Association
FP	Functional Process
FPA	Function Point Analysis
Hrs	Hours
IEC	International Electromechanical Commission
IFPUC	International Function Point Users Group
ISBSG	International Software Benchmarking Standards Group
ISO	International Organization for Standardization
IT	Information Technology
LMS	Least Median Squares
MI	Multiple Imputation
MMRE	Mean Magnitude of Relative Error
MRE	Magnitude of Relative Error
NESMA	Netherlands Software Metrics users Association
PRED	Prediction
PROC MIANALYSE	Procedure Multiple Imputation Analysis
PROC REG	Procedure Regression Analysis
PSBTI	Plan, Specify, Build, Test, Implement
PSBT	Plan, Specify, Build, Test
R9	Release 9
RE	Relative Error
SAS	Statistical Analysis Software

SBTI	Specify, Build, Test, Implement
SD	Standard Deviation
SE	Standard Error
SPSS	Statistical Package for the Social Sciences

LIST OF SYMBOLS

B	Between: the imputation variance
N	Number of projects
%	Percentage
P	Probability of obtaining a test statistic (*p*-value)
$\overline{P}$	Parameter
T	Total variance
$\overline{U}$	Within: the imputation variance
M	Correction for a finite number of imputations m

INTRODUCTION

Currently, there are many ways to develop software products and the scope of the effort estimation problem is much larger now than it was in the early days of software development in the 1960s when:

- most software was custom built,
- projects had dedicated staff, and
- companies were usually paid on an effort basis (i.e. 'cost plus', or 'time and materials').

Underestimation of effort causes schedule delays, over-budgeting, poor software quality, dissatisfied customers, and overestimation of the effort leads to wasted software development resources. Thus, over the last three decades, a number of software effort estimation methods with different theoretical concepts have been developed (Jorgensen et Shepperd, 2007), some of which have combined previously existing effort estimation methods (Stephen et Martin, 2003); (Mittas et Angelis, 2008). To improve the accuracy of software effort estimation, many organizations collect software project data and use effort estimation models derived from these data sets (Shepperd et Schofield, 1997); (Jeffery, Ruhe et Wieczorek, 2000); (Lokan et Mendes, 2006); (Mendes et Lokan, 2008).

However, software engineering data sets typically contain outliers that can degrade the data quality. An outlier is defined as a data point that appears to be inconsistent with the rest of the data sets (Barnett et Lewis, 1995). If a software effort estimation model is built on a historical data set that includes outliers, then it is difficult to obtain meaningful effort estimates because the outliers can easily distort the model and degrade the estimation accuracy.

Missing values is another of the problems often faced in statistical analysis in general, and in multivariate analysis in particular (Everitt et Dunn, 2001). Furthermore, the proper handling of missing values becomes almost necessary when statistics are applied in a domain such as

software engineering, because the information being collected and analyzed is typically considered by submitters as commercially sensitive to the software organizations.

Software data quality is another crucial factor affecting the performance of software effort estimation; however, many studies on developing effort estimation models do not consider the data quality (Pendharkar, Subramanian et Rodger, 2005); (Sun-Jen et Nan-Hsing, 2006); (de Barcelos Tronto, da Silva et Sant'Anna, 2007); (Jianfeng, Shixian et Linyan, 2009).

While a number of researchers in the literature have used the ISBSG repository for research purposes, only a few have examined techniques to tackle : A) the data quality issue, B) the problem of outliers and C) the missing values in large multi-organizational repositories of software engineering data (Deng et MacDonell, 2008).

This thesis contains seven chapters. The current introduction outlines the organization of the thesis.

Chapter 1 presents the literature review of the ISBSG data repository, including the ISBSG data collection process. This chapter also presents a review of related work and establishes the theoretical framework for this research, followed by a focus on the modeling techniques to deal with missing values, as well techniques to deal with outliers. Finally this chapter presents the more frequently used quality criteria for the estimation techniques.

Chapter 2 presents a number of research issues identified from the analysis of the literature review on software effort estimation, as well the motivation of this research project. This chapter also presents the specific research goal and objectives, and the scope of this research.

Chapter 3 presents the general view and detailed view of the research methodology, as well as the methods selected to investigate the ISBSG dataset to deal with missing values and outliers before building estimation models.

Chapter 4 presents the detailed data preparation to explore the ISBSG data repository, followed by the two verification steps for the ISBSG dataset preparation, as well the selected variables used in this research. This chapter also presents the detailed effort by phase with the total project effort recorded in the ISBSG data repository. This chapter also presents the identification of outliers, and the use of the Grubbs test to deal with outliers and, as well the outliers behaviors in the ISBSG repository.

Chapter 5 investigates the use of the multiple imputation (MI) technique with the ISBSG repository for dealing with missing values. This chapter also presents the regression analysis trained with the imputed datasets (with and without outliers), as well the variance information of MI for estimation models (Effort plan and Effort Implement).

Chapter 6 presents the general strategy for measuring the predictive accuracy of an effort estimation model, followed by the specific strategy for investigating candidate biases. This chapter also investigates the impact on parameter estimates with and without outliers. This chapter also presents the verification results of the estimation model.

The Conclusion chapter summarizes the results of this thesis. The limitations and future work are also discussed in this chapter. In addition, a few recommendations are also presented in this chapter.

CHAPTER 1

LITERATURE REVIEW

1.1 ISBSG data repository

The International Software Benchmarking Standards Group (ISBSG) was initiated in 1989 by a group of national software measurement associations to develop and promote the use of measurement to improve software processes and products for the benefit of both business and governmental organizations.

The mission of ISBSG is to improve the management of IT resources through improved project estimation, productivity, risk analysis and benchmarking. More specifically, this mission includes the provision and exploitation of public repositories of software engineering knowledge that are standardized, verified, recent and representative of current technologies. The data in these repositories can be used for estimation, benchmarking, project management, infrastructure planning, out sources management, standards compliance and budget support (ISBSG, 2009).

The data repository of the ISBSG (ISBSG, 2013) is a publicly available multi-company data set which contains software project data collected from various organizations around the world from 1989 to 2013. This data set has been used in many studies focusing on software effort estimation, and this in spite of the diversity of its data elements.

ISBSG is a not-for-profit organization and it exploits three independent repositories of IT history data to help improve the management of IT globally:

1. Software Development and Enhancement Repository – over 6,000 projects (Release 12, 2013).
2. Software Maintenance and Support Repository – over 350 applications (Release 10, 2007).

3. Software Package Acquisition and Implementation Repository - over 150 projects to date (Release 10, 2007).

The ISBSG Software Development and Enhancement Repository contains data originating from organizations across the world with projects from different industries which have used different development methodologies, phases and techniques; this repository also captures information about the project process, technology, people, effort and product of the project (Lokan et al., 2001).

However, in a few studies on project effort estimation there has been a new awareness of the importance of treating missing data in appropriate ways during analyses (Myrtveit, Stensrud et Olsson, 2001).

1.2 ISBSG data collection

Nowadays, ISBSG has made available to the public a questionnaire to collect data about projects, including software functional size measured with any of the measurement standards recognized by the ISO (i.e. COSMIC functional size – ISO 19761, and so on). The ISBSG questionnaire contains six parts (Cheikhi, Abran et Buglione, 2006):

- Project attributes
- Project work effort data
- Project size data (function points)
- Project quality data
- Project cost data
- Project estimation data

Subsequently, ISBSG assembles this data in a repository and provides a sample of the data fields to practitioners and researchers. The data collection questionnaire includes a large amount of information about project staffing, effort by phase, development methods and techniques, etc. The ISBSG has identified 8 of the organization questions and 15 of the

application questions as particularly important. Moreover, the ISBSG provides a glossary of terms and measures to facilitate understanding of the questionnaire, to assist users at the time they collect data and to standardize the data collection process (Cheikhi, Abran et Buglione, 2006). While the ISBSG established its initial data collection standard over 15 years ago, it constantly monitors the use of its data collection questionnaire and, at times, reviews its content: it attempts to reach a balance between what data is good to have and what is practical to collect. Organizations can use the ISBSG data collection questionnaire, in total or in part, for their own use: it is available free from (ISBSG, 2009), with no obligation to submit data to the ISBSG. But whatever an organization ends up with, it has to ensure that the data being collected is data that will be used and useful.

When a questionnaire approach to data collection is employed, some thoughts should be given to developing a set of questions that provide a degree of cross checking. Such an approach allows for collected data to be assessed and rated for completeness and integrity. Project ratings can then be considered when selecting a data set for analysis (Hill, 2003).

1.2.1 The ISBSG data collection process

Data is collected and analyzed according to the ISBSG Standard (ISBSG, 2013) which defines the type of data to be collected (attributes of the project or application) and how the data is to be collected, validated, stored and published so as to guarantee the integrity of the data and the confidentiality of the organizations submitting it. The standard is implicitly defined by the collection mechanism: Figure 1-1 illustrates the ISBSG Data Collection process.

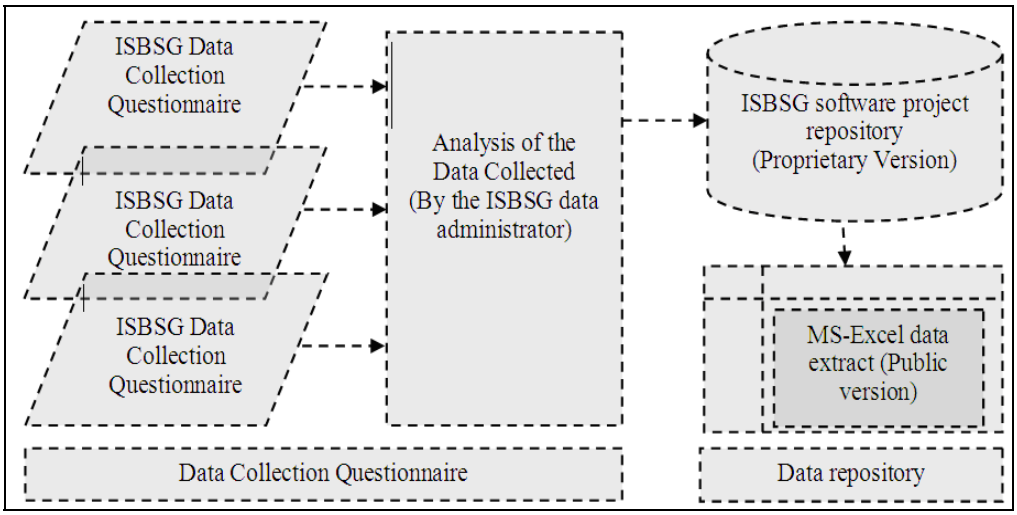


Figure1.1 ISBSG Data Collection Process
Source: (Cheikhi, Abran et Buglione, 2006)

For the purpose of software benchmarking, ISBSG collects, analyzes and reports data relating to products developed and processes implemented within organizational units in order to (Cheikhi, Abran et Buglione, 2006):

- Support effective management of the processes.
- Objectively demonstrate the comparative performance of these processes.

The outcomes of a software benchmarking process are:

- Information objectives of technical and management processes will be identified.
- An appropriate set of questions, driven by the information needs will be identified and/or developed.
- Benchmark scope will be identified.
- The required performance data will be identified.
- The required performance data will be measured, stored, and presented in a form suitable for the benchmark.
- Benchmark activities will be planned.

Even though the ISBSG data repository does not necessarily address the totality of the information needs of an organization, there are advantages in using the ISBSG as a reference solution for initiating a software measurement program:

- It offers an existing measurement framework that can facilitate faster implementation of the software measurement process with industry-standardized definitions of base and derived measures throughout the project life cycle phases.
- Alignment of the database of internal projects with this international repository, for comparison purposes.

1.2.2 Anonymity of the data collected

The ISBSG recognizes the imperative of guaranteeing the anonymity of the organizations that submit data to its repositories. The ISBSG carefully follows a secure procedure to ensure that the sources of its data remain anonymous. Only submitters can identify their own projects/applications in the repositories using the unique identification key provided by the ISBSG manager on receipt of a submission.

1.2.3 Extract data from the ISBSG data repository

The ISBSG assembles this data in a repository and provides a sample of the data fields to practitioners and researchers in an Excel file, referred to hereafter as the ISBSG MS-Excel data extract – see Figure 1.1. All of the information on a project is reviewed by the ISBSG data administrator and rated in terms of data quality (from A to D). In particular, the ISBSG data administrator looks for omissions and inconsistencies in the data that might suggest that its reliability could be questioned.

To develop new models using the ISBSG repository mainly depends on the stored data of the completed projects to determine the characteristics of the estimation models in the development cycle of projects. Many of the published and practical research to predict software development effort and size, using collected data from the completed projects, faced a set of challenges in data collection. The data of these commercial projects are often

confidential as well as very sensitive: this leads to disinclination to share information across organizations. Therefore, a relatively small number of organizations are committing sufficient effort to collect and organize data for sharing such project information through publicly available repositories, such as the ISBSG repository.

1.3 Literature Review of ISBSG-based studies

The ISBSG organization collects voluntarily-provided project data (Functional Size, Work Effort, Project Elapsed Time, etc.) from the industry, concealing the source and compiling the data into its own data repository.

(Deng et MacDonell, 2008) discussed the reported problems over the quality and completeness of the data in this ISBSG repository. They described the process they used in attempting to maximize the amount of data retained for modeling software development effort at the project level; this is based on previously completed projects that had been sized using IFPUG/NESMA function point analysis (FPA) and recorded in the repository. Moreover, through justified formalization of the data set and domain-informed refinement, they arrived at a final usable data set comprising 2862 (out of 3024) observations across thirteen variables. In their methodology the pre-processing of data helps to ensure that as much data is retained for modeling as possible. Assuming that the data does reflect one or more underlying models, (Deng et MacDonell, 2008) suggest that such retention should increase the probability of robust models being developed.

(Kitchenham, Mendes et Travassos, 2006) used the ISBSG repository and they discarded data - in some instances major proportions of the original data - if there were missing values in observations. While this step is sometimes mentioned by authors, it is not always explained in detail: there seems to be a view that this is a necessary but relatively incidental element of data preparation. In other instances, observations have been discarded if they did not have a high ISBSG-assigned quality rating on submission.

This is a relatively blunt approach to data set refinement. Even then, some previous studies do not consider at all the impact of such filtering on the population represented by the remaining observations. For the reader, when it is not clear what records have been discarded then it is difficult to know what the retained data actually represents.

This situation is not unusual in software engineering data sets comprising very large numbers of variables; however, the actual number of variables retained and used in the generated predictive models has generally been small. For example, in the work of (Mendes et al., 2006), the ISBSG data set contains more than 80 variables but just four were used in the final model generated. It is of course totally acceptable to discard data in certain circumstances: as models get larger (in numbers of variables) they become increasingly intractable to build, and unstable to use. Furthermore, if accuracy is not significantly lower, then a smaller model is normally to be preferred over a larger alternative: it would be easier to understand. However, the process of discarding data, as an important step in the data handling process, should be driven not just in response to missing values, or variables with lower correlation to the target variable, but also in relation to software engineering domain knowledge.

A lesser degree of detail regarding data filtering can be seen in the work of (Adalier et al., 2007). This study begins with the 3024 observations available in Release 9 of the repository but immediately discards observations rated B through D for data collection quality. In (Xia, Ho et Capretz, 2006) the observations containing missing values are also dropped, resulting in a data set of 112 records. Of the many possible variables available, only the function point count, source lines of code and normalized productivity rate are utilized: of note in terms of effort estimation (rather than model fitting) is that the latter two are available only after a project has been completed.

(Gencel et Buglione, 2007) mentioned that the ISBSG repository contains many nominal variables on which mathematical operations cannot be carried out directly as a rationale for splitting the data into subsets for processing in relation to the size-effort relationship for software projects. An alternative approach would be to treat such attributes as dummy

variables in a single predictive model. On the basis of two previous studies, (Gencel et Buglione, 2007) took two such attributes into account (application type and business area type) but subsequently dropped the latter variable along with a measure of maximum team size because the values were missing for most of the projects in ISBSG Release 10. They also used the quality ratings as a filter, retaining those observations rated A, B or C.

(Paré et Abran, 2005) discussed the issue of outliers in the ISBSG repository. The criteria used for the identification of outliers are whether the productivity is significantly lower and higher in relatively homogeneous samples: that is, projects with significant economies or diseconomies of scale. A benefit from this exploratory research is in the monitoring of the candidate explanatory variables that can provide clues for early detection of potential project outliers for which most probable estimates should be selected not within a close range of values predicted by an estimation model, but rather at their upper or lower limits: that is, the selection of either the most optimist or most pessimist value that can be predicted by the estimation model being used.

(Lokan et al., 2001) reported on an organization which has contributed since 1999 a large group of enhancement projects to the ISBSG repository; this contributing organization has received an individual benchmarking report for each project, comparing it to the most relevant projects in the repository. In addition, the ISBSG also performed an organizational benchmarking exercise that compared the organization's set of 60 projects as a whole to the repository as a whole: whereas the first aim for the benchmarking exercise was to provide valuable information to the organization, the second aim was to measure the benchmarking exercise's effectiveness given the repository's anonymous nature.

In (Abran, Ndiaye et Bourque, 2007) an approach is presented for building size-effort models by programming languages. Abran *et al.* provided a description of the data preparation filtering used to identify and used only relevant data in their analysis: for instance, after starting with 789 records (ISBSG Release 6) they removed records with very small project effort and those for which there was no data on the programming language.

They further removed records for programming languages with too few observations to form adequate samples by programming language, ending up with 371 records relevant for their analyses. Estimation models are built next for each of the programming languages with a sample size over 20 projects, followed by a corresponding analysis of the same samples excluding 72 additional outliers for undisclosed reasons.

In (Pendharkar, Rodger et Subramanian, 2008) a quality rating filter is applied to investigate the links between team size and software size, and development effort. Furthermore, they removed records for which software size, team size or work effort values were missing. This leads to the original set of 1238 project records (Release 7) being reduced to 540 for investigation purposes.

In (Xia, Ho et Capretz, 2006), only projects rated A and B are used for the analysis of Release 8 of the ISBSG repository. Further filters are applied in relation to FPA-sizing method, development type, effort recording and availability of all of the components of function point counting (i.e. the unadjusted function point components and 14 general system characteristics). As a result the original collection of 2027 records is reduced to a set of 184 records for further processing.

(Déry et Abran, 2005) used the ISBSG Release 9 to investigate and report on the consistency of the effort data field, including for each development phase. They identified some major issues in data collection and data analysis:

- With more than one field to indicate specific information, fields may contradict one another, leading to inconsistencies – data analysts must then either make an assumption on which field is the correct one or drop the projects containing contradictory information.
- The missing data in many fields lead to much smaller usable samples with less statistical scope for analysis and a corresponding challenge when extrapolation is desirable. They treated the missing values across phases not directly within the data set, but indirectly by inference from the average values within subsets of data with similar groupings of phases without missing values.

In (Jiang, Naudé et Jiang, 2007) ISBSG Release 10 is used for an analysis of the relationships between software size and development effort. In this study the data preparation consisted in only the software functional size in IFPUG/NESMA function points and effort in total hours, but without any additional filtering: consequently a large portion of records are retained for modeling purposes– 3433 out of 4106.

A summary of these related works is presented in Table 1.1 indicating:

- the ISBSG Release used,
- the number of projects retained for statistical analysis,
- whether or not the issue of missing values has been observed and taken into account and,
- whether or not statistical outliers have been observed and removed for further analyses.

In summary, the data preparation techniques proposed in these studies are defined mostly in an intuitive and heuristic manner by their authors. Moreover, the authors describe their proposed techniques in their own terms and structure, and there are no common practices on how to describe and document the necessary requirements for pre-processing the ISBSG raw data prior to detailed data analysis.

Table 1.1 Summary of ISBSG studies dealing with missing values and outliers

Paper work	ISBSG Release	#No Projects in the initial sample	Missing values	Outliers identified and removed
(Déry et Abran, 2005)	Release 9	3024	Observed and investigated	Observed and removed
(Pendharkar, Rodger et Subramanian, 2008)	Release 7	1238	Observed and removed	Undetermined
(Jiang, Naudé et Jiang, 2007)	Release 10	4106	Observed and removed	Undetermined
(Xia, Ho et Capretz, 2006)	Release 8	2027	Removed	Undetermined
(Abran, Ndiaye et Bourque, 2007)	Release 6	789	Removed	Observed and removed

1.4 Methods for treating missing values

1.4.1 Deletion Methods for treatment of missing values

The deletion methods for the treatment of missing values typically edit missing data to produce a complete data set and are attractive because they are easy to implement.

However, researchers have been cautioned against using these methods because they have been shown to have serious drawbacks (Schafer, 1997). For example, handling missing data by eliminating cases with missing data ("listwise deletion" or "complete case analysis") will bias results if the remaining cases are not representative of the entire sample.

Listwise Deletion: Analysis with this method makes use of only those observations that do not contain any missing values. This may result in many observations being deleted but may be desirable as a result of its simplicity (Graham et Schafer, 1999). This method is generally acceptable when there are small amounts of missing data and when the data is missing randomly.

Pairwise Deletion: In an attempt to reduce the considerable loss of information that may result from using listwise deletion, this method considers each variable separately. For each variable, all recorded values in each observation are considered and missing values are ignored. For example, if the objective is to find the mean of the X1 variable, the mean is computed using all recorded values. In this case, observations with recorded values on X1 will be considered, regardless of whether they are missing other variables. This technique will likely result in the sample size changing for each considered variable. Note that pairwise deletion becomes listwise deletion when all the variables are needed for a particular analysis, (e.g. multiple regression). This method will perform well, without bias, if the data is missing at random (Little et Rubin, 1986).

It seems intuitive that since pairwise deletion makes use of all observed data, it should outperform listwise deletion in cases where the missing data is missing completely at random

and correlations are small (Little et Rubin, 1986). This was found to be true in the Kim and Curry study (Graham et Schafer, 1999).

Studies have found that when correlations are large, listwise outperforms pairwise deletion (Azen et Guilder, 1981). The disadvantage of pairwise deletion is that it may generate an inconsistent covariance matrix in the case where multiple variables contain missing values. In contrast listwise deletion will always generate consistent covariance matrices (Graham et Schafer, 1999). In cases where the data set contains large amounts of missing data, or the mechanism leading to the missing values is non-random, Haitovsky proposed that imputation techniques might perform better than deletion techniques (Schafer, 1997).

1.4.2 Imputation methods

A- Overview of Imputation

There exist more statistically principled methods of handling missing data which have been shown to perform better than ad-hoc methods (Schafer, 1997). These methods do not concentrate solely on identifying a replacement for a missing value, but on using available information to preserve relationships in the entire data set. Several researchers have examined various techniques to solve the problem of incomplete multivariate data in software engineering.

The basic idea of imputation methods is to replace missing values with estimates that are obtained based on reported values (Colledge et al., 1978).

In cases where much effort has been expended in collecting data, the researcher likely want to make the best possible use of all available data and prefer not to use a deletion technique as carried on by many researchers (Little, 1988). Imputation methods are especially useful in situations where a complete data set is required for the analysis (Switzer, Roth et Switzer, 1998). For example, in the case of multiple regressions all observations must be complete. In these cases, substitution of missing values results in all observations of the data set being

used to construct the regression model. It is important to note that no imputation method should add information to the data set.

The key reason for using imputation procedures method is that it is simple to implement and no observation is excluded, as would be the case with listwise deletion. The disadvantage is that the measured variance for that variable will be underestimated (Reilly et Marie, 1993).

B- Hot-Deck Imputation: the technique of hot deck imputation (Little et Rubin, 2002), (Kim et Wayne, 2004), (Fuller et Kim, 2005) and (Ford, 1983) is called fractional hot deck imputation.

Hot-deck imputation involves filling in missing data by taking values from other observations in the same data set. The choice of which value to take depends on the observation containing the missing value. Hot-deck imputation selects an observation (donor) that best matches the observation containing the missing value. Observations containing missing values are imputed with values obtained from complete observations within each category. It is assumed that the distribution of the observed values is the same as that of the missing values. This places great importance on the selection of the classification variables. The purpose of selecting a set of donors is to reduce the likelihood of an extreme value being imputed one or more times (Little et Rubin, 1986), (Colledge et al., 1978). (Little, 1988) concluded that hot-deck imputation appears to be a good technique for dealing with missing data, but suggested that further analysis be done before widespread use.

C- Cold Deck Imputation: this method is similar to hotdeck imputation except that the selection of a donor comes from the results of a previous survey (Little, 1992). Regression imputation involves replacing each missing value with a predicted value based on a regression model. First, a regression model is built using the complete observations. For each incomplete observation, each missing value is replaced by the predicted value found by replacing the observed values for that observation in the regression model (Little, 1992).

A cold deck method imputes a non-respondent of an item by reported values from anything other than reported values for the same item in the current data set (e.g., values from a covariate and/or from a previous survey). Although sometimes a cold deck imputation method makes use of more auxiliary data than the other imputation methods, it is not always better in terms of the mean square errors of the resulting survey estimators.

D- Mean Imputation: this method imputes each missing value with the mean of observed values. The advantage of using this method is that it is simple to implement and no observations are excluded, as would be the case with listwise deletion. The disadvantage is that the measured variance for that variable will be underestimated (Little et Rubin, 1986) and (Switzer, Roth et Switzer, 1998). For example, if a question about personal income is less likely to be answered by those with low incomes, then imputing a large amount of incomes equal to the mean income of reported values decreases the variance.

E- Regression Imputation: Regression imputation involves replacing each missing value with a predicted value based on a regression model. First, a regression model is built using the complete observations. For each incomplete observation, each missing value is replaced by the predicted value found by replacing the observed values for that observation in the regression model (Little et Rubin, 1986).

F- Multiple Imputation Methods:
Multiple imputation (MI) is the technique that replaces each missing value with a pointer to a vector of ‘m’ values. The ‘*m*’ values come from ‘*m*’ possible scenarios or imputation procedures based either on the observed information or on historical or posterior follow-up registers.

MI is an attractive choice as a solution to missing data problems: it represents a good balance between quality of results and ease of use. The performance of multiple imputations in a variety of missing data situations has been well-studied in (Graham et Schafer, 1999), and (Joseph et John, 2002).

Multiple imputation has been shown to produce the parameter estimates which reflect the uncertainty associated with estimating missing data. Further, multiple imputation has been shown to provide adequate results in the presence of a low sample size or high rates of missing data (John, Scott et David, 1997).

Multiple imputation does not attempt to estimate each missing value through simulated values but rather to represent a random sample of the missing values. This process results in valid statistical inferences that properly reflect the uncertainty due to missing values.

The multiple imputation technique has the advantage of using the complete-data methodologies for the analysis and the ability to incorporate the data collector's knowledge (Rubin, 1987).

Multiple imputation is a modeling technique that imputes one value for each missing value. This is the case because imputing one value assumes no uncertainty. Multiple imputation remedies this situation by imputing more than one value, taken from a predicted distribution of values (John, Scott et David, 1997). The set of values to impute may be taken from the same or different models displaying uncertainty towards the value to impute or the model being used, respectively. For each missing value, an imputed value is selected from the set of values to impute, each creating a complete data set. Each data set is analyzed individually and final conclusions are obtained by merging those of the individual data sets. This technique introduces variability due to imputation, contrary to the single imputation techniques.

Multiple imputation is the best technique for filling in missing observations: it fills in missing values across replicate datasets according to a conditional distribution based on other information in the sample (Fuller et Kim, 2005) and (Ford, 1983).

(Jeff, 2005) described multiple imputation MI as a three-step process:

1. Sets of plausible values for missing observations are created using an appropriate model that reflects the uncertainty due to the missing data. Each of these sets of plausible values can be used to "fill-in" the missing values and create a "completed" dataset.
2. Each of these datasets can be analyzed using complete-data methods.
3. Finally, the results are combined.

However, the imputed value is a draw from the conditional distribution of the variable with the missing observation: the discrete nature of the variable is maintained as its missing values are imputed.

(Wayman, 2002), (Graham, Cumsille et Elek-Fisk, 2003): multiple imputation can be used by researchers on many analytic levels. Many research studies have used multiple imputation and good general reviews on multiple imputation have been published (Little, 1995). However, multiple imputation (MI) is not implemented by many researchers who could benefit from it, very possibly because of lack of familiarity with the MI technique.

G- Summary

The analysis of datasets with missing values is one area of statistical science where real advances have been made. Modern missing-data techniques which substantially improve upon old ad hoc methods are now available to data analysts (Rubin, 1996). Standard programs for data analysis such as SAS, SPSS, and LISREL were never intended to handle datasets with a high percentage of incomplete cases, and the missing data procedures built into these programs are crude at best. On the other hand, these programs are exceptionally powerful tools for complete data (Rubin, 1996). Furthermore, MI does resemble the older methods of case deletion and ad hoc imputation in that it addresses the missing data issue at the beginning, prior to the substantive analyses. However, MI solves the missing data problem in a principled and statistically defensible manner, incorporating missing data uncertainty into all summary statistics (Rubin, 1996). MI will be selected in this research as

one of the most attractive methods for general purpose handling of missing data in multivariate analysis.

1.5 Techniques to deal with outliers

The identification of outliers is often thought of as a means to eliminate observations from a data set to avoid disturbance in the analysis. But outliers may as well be the interesting observations in themselves, because they can give the hints about certain structures in the data or about special events during the sampling period. Therefore, appropriate methods for the detection of outliers are needed.

An outlier corresponds to an observation that lies an abnormal distance from other values in every statistical analysis. These observations, usually labeled as outliers, may cause completely misleading results when using standard methods and may also contain information about special events or dependencies (Kuhnt et Pawlitschko, 2003).

Outlier identification is an important step when verifying the relevance of the values in multivariate analysis: either because there is some specific interest in finding atypical observations or as a preprocessing task before the application of some multivariate method, in order to preserve the results from possible harmful effects of those observations (Davies et Gather, 1993).

Outliers are defined as observations in a data set which appears to be inconsistent with the remainder of that data set. Identification of outliers is often thought of as a means of eliminating observations from data set due to disturbance (Abran, 2009).

The identification of outliers is an important step to verify the relevance of the values of the data in input: the values which are significantly far from the average of the population of the data set will be the candidate outliers. The candidate outliers would be typically at least 1 or

2 orders of magnitude larger than the data points closer to it: A graphical representation as statistical tests can be used to identify the candidate outliers.

There are several techniques to address the problem of outliers' data in software engineering. For instance, a number of authors introduced a set of techniques to deal with outliers in the dataset, while a number of other authors did not address at all the presence of outliers. The effect of the outlier elimination on the software effort estimation has not received much consideration until now. However, to improve the performance of an effort estimation model, there is a need to consider this issue in advance of building the model.

(Chan et Wong, 2007) have proposed a methodology to detect and eliminate outliers using the Least Median Squares (LMS) before software effort estimation based on the ISBSG (Release 6). Although (Chan et Wong, 2007) show that the outlier elimination is necessary to build an accurate effort estimation model, their work has the following limitations in terms of research scope and experimentation: because this work only used statistical methods for outlier elimination and effort estimation, it cannot show the effect of outlier elimination to the accuracy of software effort estimation on the inappropriate data set to be applied by the statistical method: for example, the data distribution is unknown.

The outliers are defined as observations in a data set which appear to be inconsistent with the remainder of that data set. The identification of outliers is often thought of as a means to eliminate observations from a data set to avoid undue disturbances in further analysis (Kuhnt et Pawlitschko, 2003) and (Davies et Gather, 1993). But outliers may as well be the most interesting observations in themselves, because they can give hints about certain structures in the data or about special events during the sampling period. Therefore, appropriate methods for the detection of outliers are needed. The identification of outliers is an important step to verify the relevance of the values of the data in input: the candidate outliers would be typically at least 1 or 2 orders of magnitude larger than the data point closer to them and a graphical representation can be used to identify the candidate outliers. Statisticians have devised several ways to detect outliers.

The presence of outliers can be analyzed with Grubbs test as well as Kolmogorov-Smirnov test (Abran, 2009) to verify if the variable in a sample has a normal distribution, also referred to as ESD method (Extreme Studentized Deviate): this studentized values measure how many standard deviations each value is from the sample mean:

- When the P-value for Grubb' test is less than 0.05, that value is a significant outlier at the 5.0% significance level;
- Values with a modified Z-score greater than 3.5 in absolute value may well be outliers; and
- Kolmogorov-Smirnov test is used to gives a significant P-value (high value), which allows to assume that the variable is distributed normally.

1.6 Estimation Models

1.6.1 Regression techniques

A significant proportion of research on software estimation has focused on linear regression analysis; however, this is not the unique technique that can be used to develop estimation models. An integrated work about these estimation techniques has been published by (Gray et MacDonell, 1997) who presented a detailed review of each category of models.

The least squares method is the most commonly used method for developing software estimation models: it generates a regression model that minimizes the sum of squared errors to determine the best estimates for the coefficients - (de Barcelos Tronto, da Silva et Sant'Anna, 2007) and (Mendes et al., 2005).

(Gray et MacDonell, 1997): "Linear least squares regression operates by estimating the coefficients in order to minimize the residuals between the observed data and the model's prediction for the observation. Thus all observations are taken into account, each exercising the same extent of influence on the regression equation, even the outliers".

Linear least squares regression also gets its name from the way the estimates of the unknown parameters are computed. The technique of least squares that is used to obtain parameter estimates was independently developed in (Stigler, 1988), (Harter, 1983) and (Stigler, 1978). Linear regression is a popular method for expressing an association as a linear formula, but this does not mean that the determined formula will fit the data very well. Regression is based on a scatter plot, where each pair of attributes (xi, yi) corresponds to one data point when looking at a relationship between two variables. The line of best fit among the points is determined by the regression. It is called the least-squares regression line and is characterized by having the smallest sum of squared vertical distances between the data points and the line (Fenton et Pfleeger, 1998).

1.6.2 Estimation models: evaluation criteria

There are a number of criteria to evaluate the predictability of the estimation model (Conte, Dunsmore et Shen, 1986b):

1- Magnitude Relative Error (MRE) = | Estimate value – Actual value | / Actual value. The MRE values are measured for each project in the data set, while the mean magnitude of relative error (MMRE) computes the average over N projects in the data set. The MRE value is calculated for each observation i for which effort is estimated at that observation.

2- Mean Magnitude Relative Error for *n* projects (MMRE) = 1/n*Σ(MREi) where i = 1...n.

This MMRE measures the percentage of the absolute value of the relative errors, averaged over the N projects in the data set. As the mean is calculated by taking into account the value of every estimated and actual from the data set, the result may give a biased assessment of imputation predictive power when there are several projects with large MREs.

3- Measure of prediction - Pred(x/100): percentage of projects for which the estimate is within *x%* of the actual. PRED (q) = k/n, out of *n* total projects observations, *k* number of projects observations which have mean magnitude of relative error less than 0.25. The estimation models generally considered good are when PRED (25) ≥ 75% of the observations. When the MRE *x*% in set at 25% for 75% of the observations: this, pred(25) gives the percentage of projects which were predicted with a MMRE less than or equal to 0.25 (Conte, Dunsmore et Shen, 1986b).

The evaluation criterion most widely used to assess the performance of software prediction models is the Mean Magnitude of Relative Error (MMRE). The MMRE is computed from the relative error, or (RE), which is the relative size of the difference between the actual and estimated value. If it is found that the results of MMRE have small values, the results should be precise or very close to the real data. The purpose of using MMRE is to assist in selecting the best model (Conte, Dunsmore et Shen, 1986b).

1.7 Summary

The International Software Benchmarking Standards Group (ISBSG) data repository of the ISBSG (ISBSG, 2013) is a publicly available multi-company data set which contains software project data collected from various organizations around the world from 1989 to 2013. This data set has been used in many studies focusing on software effort estimation, and this in spite of the diversity of its data elements.

The ISBSG has made available to the public a questionnaire to collect data about projects, including software functional size measured with any of the measurement standards recognized by the ISO (i.e. COSMIC functional size – ISO 19761, and so on). However data is collected and analyzed according to the ISBSG Standard, The standard defines the type of data to be collected (attributes of the project or application) and how the data is to be collected, validated, stored and published. The ISBSG recognizes the imperative of guaranteeing the anonymity of the organizations that submit data to its repositories.

The ISBSG assembles this data in a repository and provides a sample of the data fields to practitioners and researchers in an Excel file, referred to hereafter as the ISBSG MS-Excel data extract.

However, this repository contains a large number of missing data, thereby often reducing considerably the number of data points available for building productivity models and for building estimation models, for instance. There exists however a few techniques to handle missing values, but they must be handled in an appropriate manner; otherwise inferences may be made that are biased and misleading.

Data analysis with ISBSG repository should have a clearly stated and justified rationale, taking into account software engineering domain knowledge as well as indicators of statistical importance. There are some weaknesses in this dataset: for instance, questions over data quality and completeness have meant that much of the data potentially available may have not actually been used in the analyses performed.

Missing data are a part of almost all research and a common problem in software engineering datasets used for the development of estimation models. The most popular and simple teachniques of handling missing values is to ignore either the projects or the attributes with missing observations. This teachnique causes the loss of valuable information and therefore may lead to inaccurate estimation models. Missing data are techniques such as listwise deletion, pairwise deletion, hot-deck Imputation, cold deck imputation, mean imputation, and regression imputation.

Therefore, this empirical study will select the most attractive method for general purpose handling of missing data in multivariate analysis, the Multiple Imputation technique, which can be used by researchers on many analytic levels. Many research studies have used multiple imputation and good general reviews on multiple imputation have been published.

In addition, there are several studies introduced a set of techniques to deal with the problem of outliers in the dataset, the outliers may as well be the most interesting observations in themselves, because they can give hints about certain structures in the data or about special events during the sampling period. The appropriate methods for the detection of outliers are needed. The identification of outliers is an important step to verify the relevance of the values of the data in input.

This chapter has presented the evaluation criteria most widely used to assess the performance of software prediction models: the Mean Magnitude of Relative Error (MMRE), computed from the relative error, or (RE).

CHAPTER 2

RESEARCH ISSUES AND RESEARCH OBJECTIVES

2.1 Research issues

Chapter 1 has presented a review of related works on the use of the ISBSG repository by researchers and how they have tackled – or not - these issues of outliers, missing values and data quality.

In summary, the ISBSG repository is not exempt of the issues that have been identified in other repositories (i.e. outliers, missing values and data quality). For instance, the ISBSG repository contains a large number of missing values for a significant amount of variables, as not all the fields are required at the time of data collection.

The ISBSG repository also contains a number of outliers in some of the numerical data fields, thus making it use rather challenging for research purposes when attempting to analyze concurrently a large subset of data fields as parameters in statistical analyses.

Therefore, researchers using this multi-organizational repository in multi variables statistical analyses face a number of challenges, including:

- there are often statistical outliers in the numerical fields;
- the data are contributed voluntarily: therefore, the quality of the data collected may vary and should be taken into account prior to statistical analysis;
- there is only a handful of the over +100 data fields mandatory in the ISBSG data collection process: therefore, there is a very large number of missing values in the non mandatory fields.

Often, missing values are just ignored for reasons of convenience, which might be acceptable when working with a large dataset and a relatively small amount of missing data. However, this simple treatment can yield biased findings if the percentage of missing data is relatively large, resulting in lost information on the incomplete cases. Moreover, when dealing with relatively small datasets, it becomes impractical to just ignore missing values or to delete incomplete observations from the dataset. In these situations, more reliable imputation methods must be pursued in order to perform meaningful analyses.

This research focuses on the issues of missing values and outliers in the ISBSG repository, and proposes and empirical number of techniques for pre-processing the input data in order to increase their quality for detailed statistical analysis.

2.2 Research motivation

Up until recently, most of the empirical studies on the performance of estimation models were made using samples of very small size (less than 20 projects) while only a few researchers used samples of a larger size (between 60 and 90 projects). With the set-up of the repository of software projects by the International Software Benchmarking Standards Group – ISBSG – there exists now a much larger data repository available for building estimation models, thereby providing a sounder basis for statistical studies. Researchers from around the world have started to use this repository (See Appendix XXIX on the CD attached to this thesis), but they have encountered new challenges. For instance, there is a large number of outliers as well as missing values for a significant number of variables for each project (eg. only 10% of the data fields are mandatory at the data collection time), making its uses rather challenging for research purposes.

Furthermore, several problems arise in the identifying and justifying of the pre-processing of the ISBSG data repository, including clustering groups of projects that share similar value characteristics, discarding and retaining data, identifying in a systematic manner the outliers and investigating causes of such outliers' behaviors.

The motivation for this research project is to tackle the new problems of access to larger datasets in software engineering effort estimation including the presence of outliers and a considerable number of missing values.

2.3 Research goal and objectives

The research goal of this thesis is to develop an improved usage of the ISBSG data repository by both practitioners and researchers by leveraginge the larger quantity of data available for statistical analysis in software engineering, while discarding the data which may affect the meaningfulness of the statistical tests.

The specific research objectives are:

1- To investigate the use of the multiple imputation (MI) technique with the ISBSG repository for dealing with outliers and missing values.

2- To demonstrate the impact and evaluate the performance of the MI technique in current software engineering repositories dealing with software project efforts for estimation purposes, between estimated effort and actual effort.

2.4 Research scope

The scope of this research will use the ISBSG dataset repository release 9 (ISBSG, 2005), which contains data on 3024 software projects: the reason that prevents this research from using Release 12 is that there are a large number of projects that had information on effort by project phases in Release 9 but did not have anymore such information in Release 12.

The following methods will be used to investigate the dataset and to deal with missing values and outliers before building estimation models:

1. the Multi imputation procedure (MI): this technique will be used to deal with missing value in the ISBSG dataset,
2. the Grubbs Test and Kolmogorov-Smirnov test to identify the outliers projects,

3. the evaluation criteria to evaluate estimation models overestimation and underestimation respectively (Foss et Kitchenham, 2003).

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Research methodology

This section presents the general view of the research methodology which is divided into five phases – see Figure 3.1:

1. Collection and synthesis of lessons learned: in this phase, the prior research work on the use of the ISBSG repository was analyzed.
2. Data preparation and identification of outliers in ISBSG data: in this phase the Grubbs Test and Kolmogorov-Smirnov test will be applied to identify the outliers projects.
3. Multiple Imputation technique to be applied for missing values: the multiple imputation technique will be applied on the ISBSG Data repository to deal with missing values.
4. Solution for handling missing values and outliers: this phase will handle the missing values in effort estimation (with and without outliers), and this phase also will investigate the use of the multiple imputation (MI) technique with the ISBSG repository for dealing with missing values, and will report on its use.
5. Verify the contribution of MI on effort estimation: this phase will demonstrate the impact and evaluate the performance of the MI technique in software prediction models between estimated effort and actual effort. This phase also investigates the impact on parameter estimate analysis (with and without outliers) of the use of of MI on incomplete datasets.

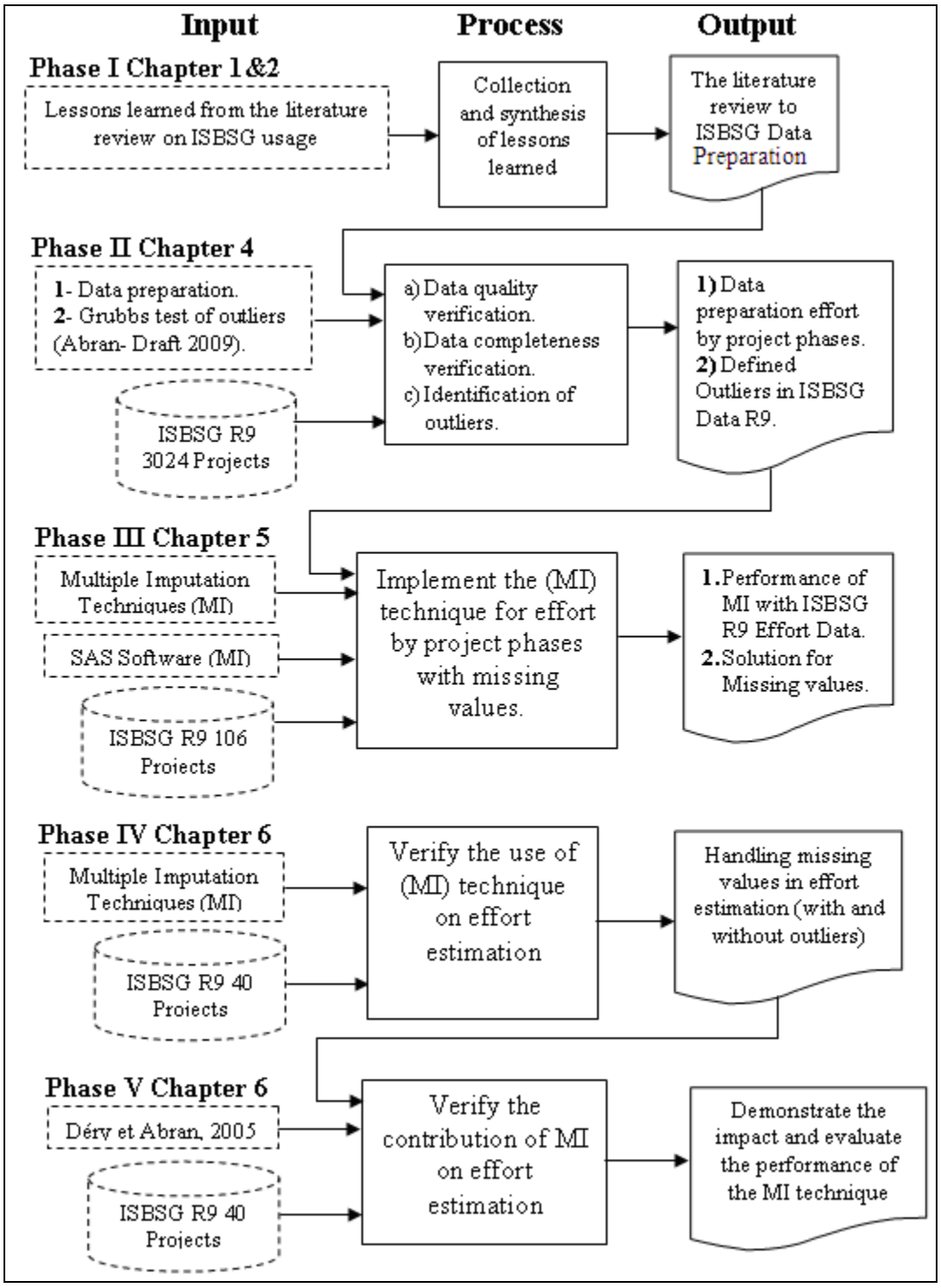


Figure 3.1 General View of the Research Methodology

3.2 Detailed methodology for phase I: Collection and synthesis of lessons learned

This phase I identified some of the possible reasons why software engineering practitioners and researchers have had difficulty in coming up with reasonable and well quantified relationships using the ISBSG data repository, although considerable amounts of papers have been published to date – See Figure 3.2.

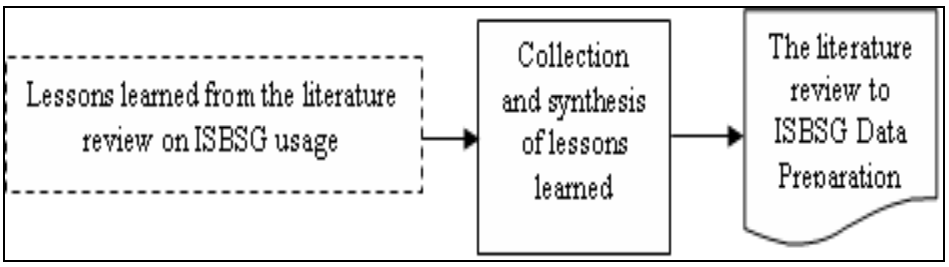


Figure 3.2 Phase I: Collection and synthesis of lessons learned

This phase collects and syntheses the prior studies in Table 1.1 which analyzed the ISBSG data repository.

The finding from this literature review drive this empirical study to investigate the modeling techniques to deal with missing values and outliers in current software engineering projects of the ISBSG repository.

3.3 Detailed methodology for phase II: Data preparation and identification of outliers

This phase II will prepare the projects of ISBSG data repository to yield data quality to a data analysis process. Prior to analyzing the data preparation for ISBSG repository, it is important to understand how fields are defined, used and recorded, as recommended in (Deng et MacDonell, 2008). This phase II also identifies the outliers in the ISBSG data repository R9 using the test identifier which attracted the interest of various researchers in Table 1.1– see Figure 3.3.

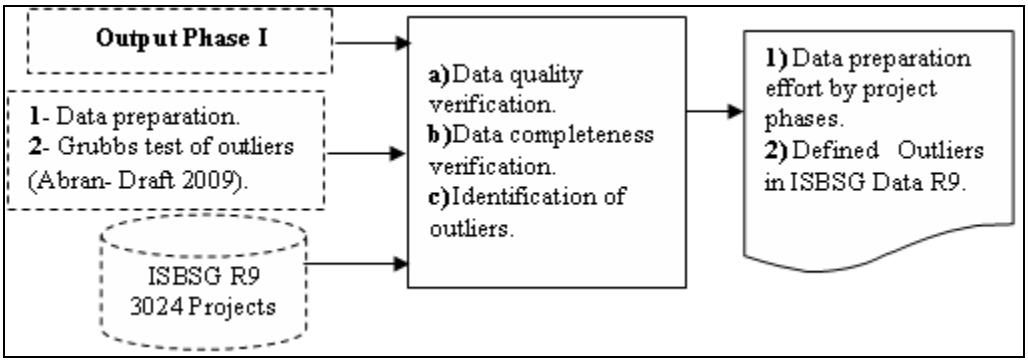


Figure 3.3 Phase 2: Identification of outliers in ISBSG Data set

The inputs in this phase are:

- the output of phase I and;
- data preparation and the Grubbs Test and Kolmogorov-Smirnov test;
- the ISBSG R9 data repository.

The output of phase II is data preparation and observed outliers in the ISBSG data repository R9: it will help this research to deal with the missing values and outliers.

3.4 Detailed methodology for phase III: Multiple Imputation technique to deal with missing values

This phase III will apply the Multiple Imputation technique using SAS software to deal with missing values in the ISBSG data repository R9 for effort data – See Figure 3.4.

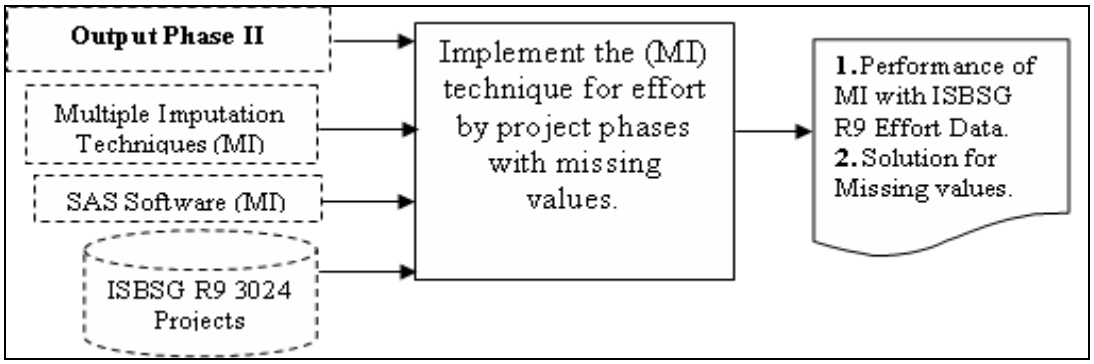


Figure 3.4 Phase III: Multiple Imputation Technique for missing values in ISBSG R9

The inputs in this phase are:

- the output of phase 2;
- the ISBSG R9 data repository;
- Multiple Imputation Techniques (MI);
- the SAS software for applying the multiple imputation technique.

SAS software: is a statistical software system which integrates utilities for storing, modifying, analyzing, and graphing data.

3.5 Multiple imputation Overviews

Multiple imputation does not attempt to estimate each missing value through simulated values but rather to represent a random sample of the missing values. This process results in valid statistical inferences that properly reflect the uncertainty due to missing values. This section summarizes the multiple imputation method used by (SAS) software package, using a three steps procedure:

1. Multiple Imputation: The missing data are filled in *m* times to generate *m* complete data sets.
2. Regression: The *m* complete data sets are analyzed by using standard regression procedures.
3. Combination of results: The results from the *m* complete data sets are combined for the inference.

The output from this phase should be a completed data set of ISBSG data repository R9, with the solution of the missing values based on the multiple imputation technique used to deal with the missing values in the dataset.

Imputation/Regression using SAS (PROC MI / MIANALYZE)

Most SAS statistical procedures exclude observations with any missing variable values from the analysis. These observations are called incomplete cases. The MI procedure provides three methods to create imputed data sets that can be analyzed using standard procedures.

A SAS procedure, PROC MI, is a multiple imputation procedure that creates multiple imputed data sets for incomplete *p*-dimensional multivariate data.

Once the *m* complete data sets are analyzed by using standard regression procedures such as PROC REG, another new procedure, PROC MIANALYZE, can be used to generate valid statistical inferences about these parameters by combining results from the *m* complete data sets.

3.6 Detailed methodology for phase IV: Handling Missing values in effort estimation with and without Outliers

This phase IV investigates the use of a multi-imputation technique to handle missing values in the ISBSG data repository. The objective of MI is not to predict missing values that are as close as possible to the true values, but to handle missing data in a way that results in valid statistical inference – See Figure 3.5.

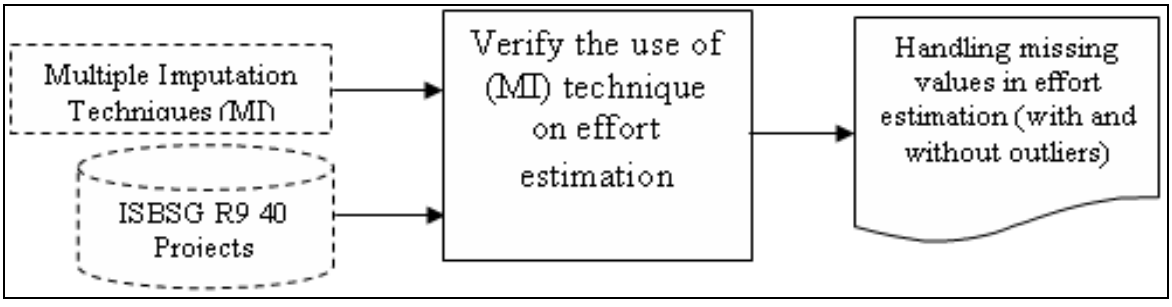


Figure 3.5 Phase IV: Handling Missing values in effort estimation with and without Outliers

The inputs in this phase are:

- Multiple Imputation Techniques (MI);
- the ISBSG R9 data repository (N=40 projects without outliers);

The output from this phase is a completed data set based on the multiple imputation technique to handling the missing values in ISBSG data repository R9, as well as the regression models of MI for Effort Implement estimation models trained with the imputed datasets N= 41 projects with outliers and N= 40 without outliers.

3.7 Detailed methodology for phase V: Verification the contribution of the MI technique on effort estimation

This phase V uses the most common accuracy predictive statistics, the mean magnitude relative error (MMRE) and the percentage relative error deviation within x (PRED(x)), to verify the impact of multiple imputation (MI) on effort estimation, as well as investigates the impact on parameter estimates with and without outliers. Furthermore, this phase also compares the output results with the study of the distribution of work effort across development phases proposed in (Déry et Abran, 2005).This phase presents the general strategy for measuring the predictive accuracy of an effort estimation model –See Figure 3.6.

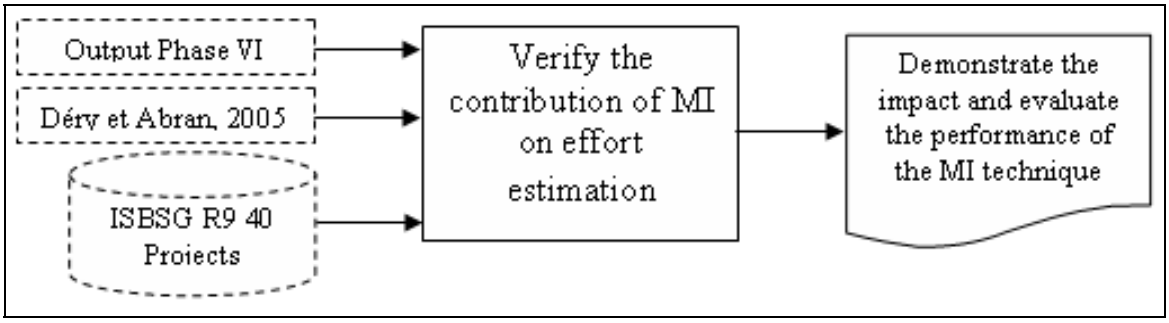


Figure 3.6 Phase V: Verification the contribution of the MI technique on effort estimation

The MRE, MMRE and PRED (0,25) values for the regression models will be obtained using the criteria to evaluate the predicatblilty of the estimation model, respectively as presented in the literature – See section 1.6, however, the investigation in this phase will be as following:

a) Investigate the impact on parameter estimates (N=41 projects with outliers, & N=21 projects with values deleted).
b) Investigate the impact on parameter estimates from a subset of N=21 projects without outliers, & from a subset of N=40 projects before and after removing missing data.
c) Investigate the contribution of MI and compare the results of the estimation model & training data, with the completed data N=20 projects, from subset of N=40 projects.
d) Investigate the contribution of relative imputation of effort estimation for N=40 projects without outliers for the Effort Implement phase.

CHAPTER 4

DATA PREPARATION AND IDENTIFICATION OF OUTLIERS

4.1 Data preparation

Data preparation is a crucial research phase. However, much work in the field of software engineering, as mentioned in the literature review of the research work using the ISBSG repository, was built on the assumption that quality data is assumed to be nicely distributed, containing no missing or incorrect values. Data preparation is concerned with analyzing the projects of ISBSG data so as to yield quality data as inputs to a data analysis process.

4.2 Data preparation for ISBSG repository

In this section two verification steps must be carried out for the the ISBSG dataset preparation of the effort by project phases:

- the data quality verification, and;
- data completeness verification.

4.2.1 Data preparation effort by project phases

This section presents the detailed data preparation to explore the ISBSG data repository. Figure 4.2 illustrates that the ISBSG Data Preparation is divided into three main steps as follows:

1. The first step is to define the projects that will be involved in this research by applying preliminary filters to the ISBSG data repository, by taking into account the extracted data in the Excel extract from the previous phase in this methodology.
2. Sizing method (IFPUG) count approach:

The Functional Size Measurement Method (FSM Method) used in the ISBSG repository to measure the functional size of software projects are IFPUG, MARK II, NESMA, FiSMA, COSMIC etc.

- In the ISBSG repository, not all the projects were sized according to the same functional sizing method. For the analyses reported here, only the 2,718 projects sized with the IFPUG method by usage international standard were retained initially.

3. Data quality rating

This step will be selecting the data quality rating (A and B), (See Appendix III on the CD attached to this thesis).

This field contains an ISBSG rating code of A, B, C or D applied to the Data Quality and Function Point Count data by the ISBSG data administration (ISBSG, 2005):

A = the data submitted was assessed as being sound with nothing being identified that might affect its integrity.

B = the submission appears fundamentally sound but there are some factors which could affect the integrity of the submitted data.

C = is given to the projects for which it was not possible to assess the integrity of the submitted data due to significant data not being provided.

D = is given to the projects to which little credibility should be given to the submitted data due to one factor or a combination of factors.

- After filtering for data quality (A and B), the number of projects was reduced to 2,562, prior to the identification of the missing values in the fields of interest.

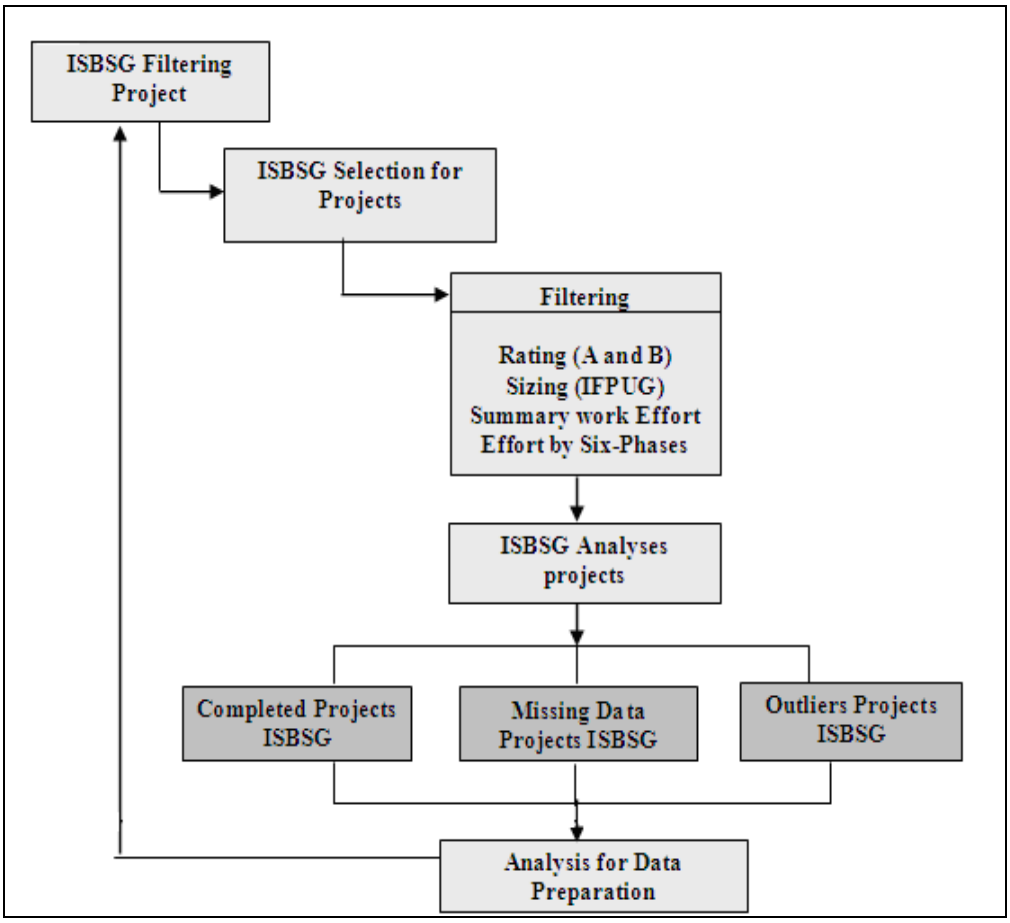


Figure 4.1 ISBSG Data Preparation

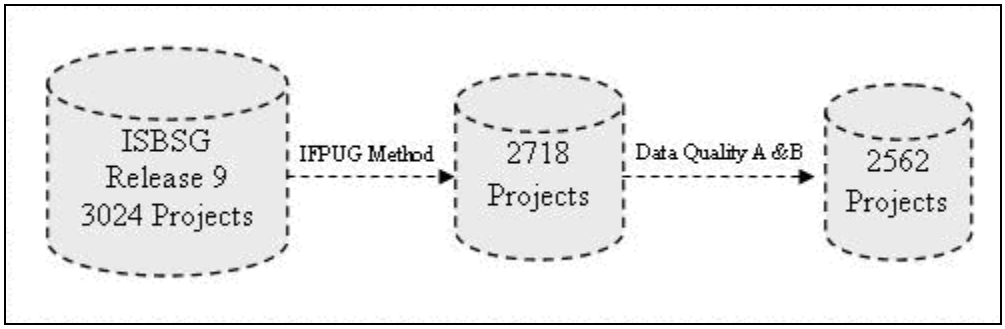


Figure 4.2 Data Preparation of ISBSG R9 Data set

4. Work effort

Total effort in hours recorded against the project. For projects covering less than a full development life-cycle, this value only covers effort for the phases reported. It includes effort for all reported teams. For projects covering the full development life-cycle, and projects where life-cycle coverage is not known, this value is the total effort for all reported teams. For projects where the total effort is not known this value is blank.

5. Effort by project phases

The last step is to analyse the fields selected from the ISBSG data projects effort by phases in order to define where the corrupted data in each of completed projects, missing value and outliers.

The other variables considered for this research analysis are the effort by 6 phases –See Table 4.1.

- The first two variables deal with the software size variable in terms of functional size in Function Points units, and corresponding sizing standard of measurement;
- The next six variables deal with project effort, including the total project effort, in hours and for each of the ISBSG-defined project phases, and the min and max of each variables – see Table 4.1.

Table 4.1 ISBSG data fields used

Data variable	Abbreviation	Units	Min in R9	Max in R9
1- Functional Size	FP	Function Points	0	2,929
2- Functional sizing method	IFPUG	-	-	-
3- Summary Work Effort	Effort	Hours	170	100529
4- Effort Plan	P	Hours	2	5,390
5- Effort Specify Phase	S	Hours	1	28,665
6- Effort Build Phase	B	Hours	30	48,574
7- Effort Test Phase	T	Hours	14	15,005
8- Effort Implement Phase	I	Hours	20	8,285

Table 4.2 presents the number of projects with recorded effort by phase profiles (Plan, Specification, Build, Test, and Implement) consistent with (Summary Effort) (See Appendix III on the CD attached to this thesis).

Table 4.2 Number of projects with effort by phase in ISBSG R9 (Déry and Abran, 2005)

Number of Projects						
Project Phases Included (1)	**With phase tags (2)**	**With detailed effort by phase (3)**	**All phases effort consistent with summary Effort (4)**	**No. of Projects with valid Data (5)**	**Projects with missing value (6)**	**Corrupted and inconsistencies Data (7)**
PSBTI	350	113	76	41	0	35
PSBT	405	200	100	62	62	38
SBTI	92	12	3	3	3	0
Total	**847**	**325**	**179**	**106**	**65**	**73**

The numbers in the rows in Table 4.2 correspond to the number of projects; the labels in the leftmost column represent the set of the 1st letter of each phase[1] included in the project effort reported, (See Appendix VI on the CD attached to this thesis). For instance:

- The label 'PSBTI' corresponds to the projects with effort data for each of the full five project phases: Planning, Specification, Build, Testing, and Implementation.
- The label 'PSBT' corresponds to the projects with effort data for each of the following four project phases: Planning, Specification, Build, and Testing (but without any data about the implementation phase.)
- The label 'SBTI' corresponds to the projects with effort data for each of the following four project phases: Specification, Build, and Testing and Implementation (but without any data about the Planning phase.)

[1] The design phase is not included in this analysis: in the ISBSG repository prior to Release 5, the 'high level design' was included in the Specify phase and 'low level design' was included in the Build phase.

However, not all projects with phase tags (Table 4.2, column 2) also have concurrently detailed effort by project phase. Since only projects with effort data recorded by project phase have the detailed effort data by project phases required for the purposes of this research, this reduces significantly the sizes of the samples available for detailed analysis: for instance, for the PSBTI phase, out of the 350 projects in this effort profile (Table 4.2, column 2), only 113 have detailed effort data by phase (Table 4.2, column 3).

Next, the verification of the consistency of the detailed effort by phase with the total project effort recorded leads to a sample of only 76 projects which meet this consistency criterion for our analytical purposes (Table 4.2, column 4). In addition, 35 projects have to be deleted for inconsistencies in the data:

- The project with the greatest amount of effort did not have the mandatory field of size in function points, which pointed out to a lack of quality control of the data recorded for this project.
- another unusual effort pattern was identified: 34 projects had, on average, 98% of the effort recorded in the specification phase, and less than 1% in each of the other 4 phases.

Using the same data preparation criteria, the sample of projects with the phase profile 'PSBT' has 100 projects, of which 38 projects have to be dropped from further analysis because of inconsistencies between the detailed levels by phase and the total effort. The project phase profile 'SBTI' with only 3 projects is a quite small sample for analysis.

4.3 Technique to deal with outliers in ISBSG data repository

Statisticians have devised several methods for detecting outliers. All the methods first quantify how far the outlier is from the other values. This can be the difference between the outlier and the mean of all points, or the difference between the outlier and the mean of the remaining values, or the difference between the outlier and the next closest value.

To verify whether or not these data points are true statistical outliers, the Grubbs test - as well as the Kolmogorov-Smirnov test - (Abran, 2009) are selected in this research project to verify if the variable in a sample has a normal distribution, also referred to as an ESD method (Extreme Studentized Deviate); the studentized values measure how many standard deviations each value is from the sample mean:

1) When the P-value for the Grubb' test is less than 0.05, that value is a significant outlier at the 5.0% significance level;
2) Values with a modified Z-score greater than 3.5 in absolute value may well be outliers; and
3) The Kolmogorov-Smirnov test is used to give a significant P-value (high value), which allows to assume that the variable is distributed normally.

However, the uses of these three methods are almost the same, but Grubbs' test is particularly easy to follow. The first step is to quantify how far the outlier is from the others by calculating the ratio Z as the difference between the outlier and the mean divided by the SD. If Z is large, the value is far from the others. After calculating the mean and SD from all values, including the outlier, the Grubb's test calculates a P value only for the value furthest from the rest. Unlike some other outlier tests, Grubbs' test only asks whether that one value is an outlier. And then the data analyst can remove that outlier, and run the test again.

The most that the Grubbs' test (or any outlier test) can do is to explain that a value is unlikely to have come from the same population as the other values in the group. From there the data analyst should decide what to do with that value.

Table 4.3 presents the overall results of the Grubbs' tests with the set of data N= 106 projects (See Appendix IV on the CD attached to this thesis) Table 4.3 presents the 3 significant outliers: The outlier tests were performed on the functional size and summary work effort variables. The "test no" in Table 4.3 represents the number of iterations for the application of the Grubbs' test for identifying the outliers, one at a time.

Table 4.3 Descriptive Statistics for Grubbs' test on Total Effort (N=106)

Test no.	Mean Total Effort	SD	No. of values	Outlier detected?	Significance level	Critical value of Z
1	5726	11032	106	Yes	0.05 (two-sided)	3.40
2	4823	5970	105	Yes	0.05 (two-sided)	3.40
3	4460	4692	104	Yes	0.05 (two-sided)	3.40
4	4173	3686	103	No	0.05 (two-sided)	3.39

Table 4.4 Outlier analysis using Grubbs' test on Total Effort

Test no.	Total Effort of the candidate outlier	Z	Significant outlier?
1	100529	8.59	Significant outlier. $P < 0.05$
2	42574	6.32	Significant outlier. $P < 0.05$
3	34023	6.30	Significant outlier. $P < 0.05$
4	15165	2.98	No, although furthest from the rest ($P > 0.05$).

Table 4.5 presents the 3 significant outliers that should be removed from further statistical analyses. An additional outlier is the project with the greatest amount of effort but which has no size in function points assigned to it: this project is therefore of no use either for benchmarking or for estimation purposes.

Table 4.5 Description of the 3 outliers deleted

No. of outliers	Function Size	Summary Work Effort	Effort Plan	Effort Build	Effort Test	Effort Specify	Effort Implement
1	(0)	34023	1190	9793	17167	4489	1384
2	781	42574	5390	7910	15078	14196	(0)
3	2152	100529	(0)	28665	48574	15005	8285

4.4 Summary

This chapter used the Grubbs test to identify the presence of outliers in numerical data fields. As well as the investigation included outlier behavior in the ISBSG repository, and outlier tests were performed on the effort and functional size. This analysis was conditioned to a sample of 106 observations of projects from the repository. When effort estimation models are built using data samples with outliers, these models distort the effort estimation models for future projects. Therefore, in this chapter the outlier test method was applied on functional size and the total work effort variables in the ISBSG repository.

CHAPTER 5

MULTIPLE IMPUTATION TECHNIQUE TO DEAL WITH MISSING VALUES IN ISBSG REPOSITORY

5.1 Multiple imputation method in SAS soffware

Multiple imputation technique is a method for the treatment of missing data, to make valid inferences regarding a population of interest.

In multiple imputation, the predicted values, called (imputes), are replacing the missing values, resulting in a full data set called an 'imputed data set'. This process is performed multiple times, producing multiple imputed data sets. Next, standard statistical analysis, for instance the regression analysis procedure (PROC REG), is carried out on each imputed data set, producing multiple analysis results. These analysis results are then combined to produce one overall analysis. Multiple imputation accounts for missing data by restoring not only the natural variability in the missing data, but also by incorporating the uncertainty caused by estimating missing data.

Multiple imputation produces complete data sets on which to perform analyses, and these analyses can be performed by nearly any method or software package the analyst chooses.

The incompleteness of data is an important issue faced by researchers who use industrial and research datasets. To overcome this problem the SAS software will be used for applying the multiple imputation technique to deal with incompleteness or missing values.

Three steps are needed to implement multiple imputation – see Figure 5.1:

1- Create imputed data sets which plausible representations of the data.

2- Perform the chosen statistical analysis on each of these imputed data sets.

3- Combine the results, to produce one set of results.

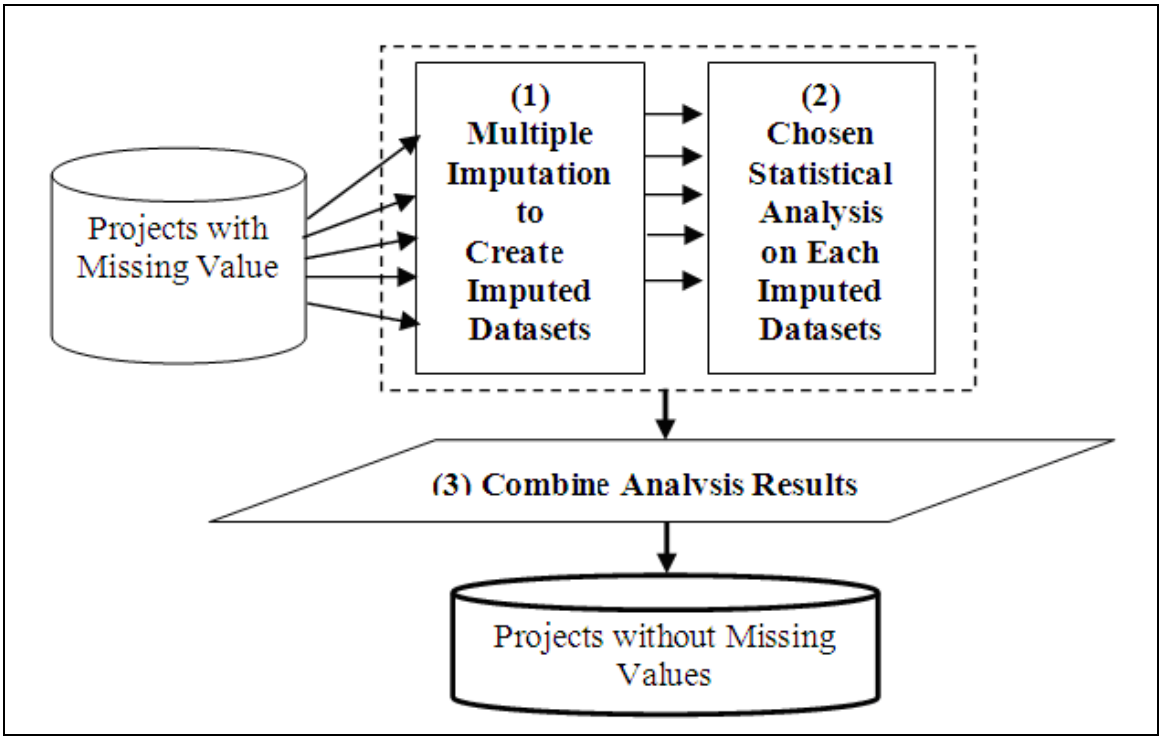


Figure 5.1 Multiple Imputation Processing

1. **Create imputed data sets**

   The first step is to create values (also referred to as *imputes*) to be substituted for the missing data. In order to achieve this, an imputation procedure must be identified that will allow imputes to be created based on the values found across the data set for the same variable in the dataset. This involves the creation of imputed datasets, which are plausible representations of the data: the missing data are filled in *m* times to generate *m* complete datasets.

2. **Analyze imputed data sets**

   Note that standard statistical analysis is conducted separately for each imputed dataset. This analysis proceeds as if there were no missing data, except that it is performed on each imputed dataset. In other words, *m* complete datasets are each analyzed using standard statistical procedures, with each completed dataset.

3. **Combine analysis results**

   Once the analyses have been completed for each imputed data set, all that remains is to combine these analyses to produce one overall set of estimates. The results from the analyses of the *m* complete datasets are combined to produce inferential results once the imputed datasets have been created.

5.2 Implement the (MI) technique for effort by project phases with missing values

This section presents an application of the three distinct phases of the multiple imputation statistical inferences on the ISBSG repository (Release 9, 2005), in addition the column (5) with valid data ending with 106 projects is used - See Table 4.2. This section is structured as follows:

- Section 5.2.1 presents step 1: creating the imputed data sets.
- Section 5.2.2 presents step 2 analyzing the imputed data sets.
- Section 5.2.3 presents step 3 combining the analysis results.

5.2.1 Step 1 Creating the imputed data sets (Imputation)

In this step, the missing values from the ISBSG R9 are imputed with a PBST profile: random numbers are generated to provide the values that are missing from the selected data fields, that is:

- the Effort Implementation (EI) phase, and
- the Effort Planning (EP) phase.

The SAS software procedure PROC MI is used to generate 5 'completed' datasets[2] for the repository. The random numbers are imputed data based on the 'seed' values inserted manually to generate random numbers. The details of this step are presented in 5.2.1.1, and the analysis of variances in 5.2.1.2.

5.2.1.1 Phase effort profile after MI based on the seeds with the full sample of 106 projects

The seed values selected for the full sample of 106 projects are set to the minimum and maximum values in hours for the two corresponding fields (EI and EP) of the PBSTI profile that does not have missing value in R9, that is the Effort Plan and Effort Implementation for the 41 projects with the PBSTI profile. Here, the minimum for the Plan and Implement phases are (2, and 20) hours, and the maximum are (5,390, and 8,285) hours - see the two rightmost columns in Table 4.1).

This leads to the following vectors of parameters for this imputation steps: the vector of minimum values for the missing value of the Plan and Implement phase sets to be generated is (2, and 20 hours), and the vector of the maximum values is (5,390, and 8,285 hours) –See Table 4.1.

The positions in the vector correspond to the order that appear in the (var) statement in the SAS procedure. In the dataset used in this research, the variables min and max are based on each variable that are entered in the procedure.

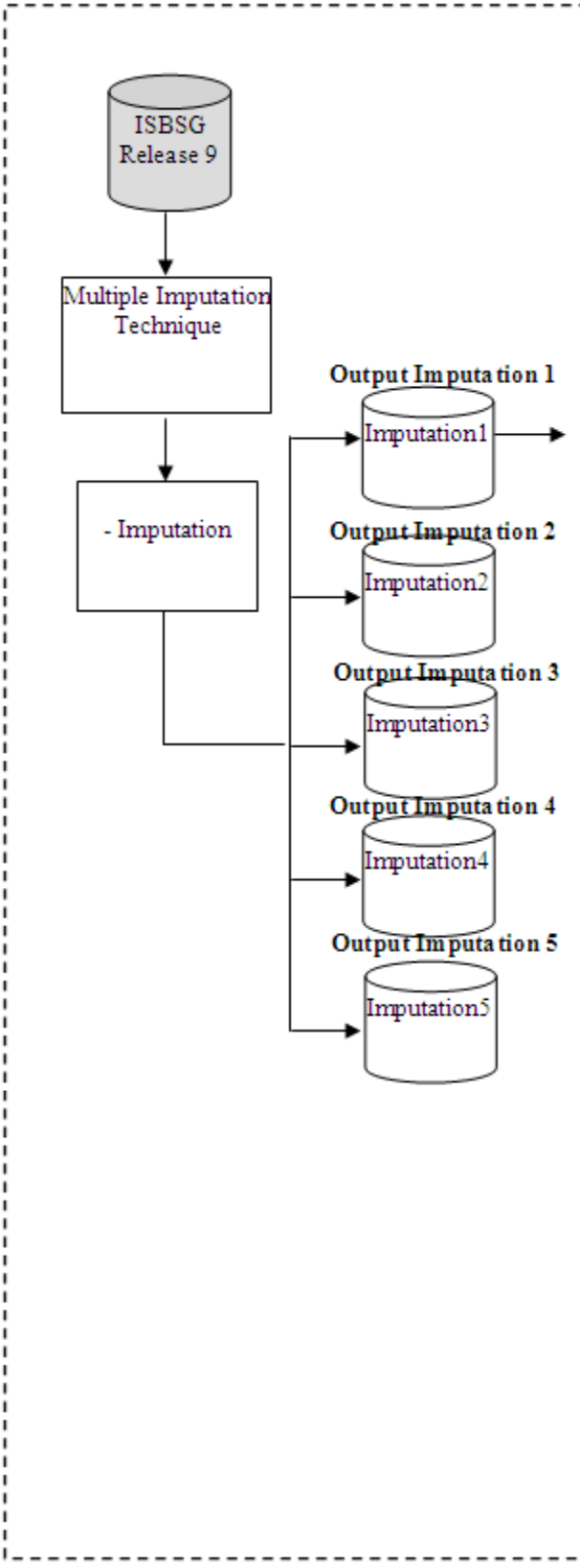
Figure 5.2 displays the outcome of Imputation 1 which generated effort data for the 65 projects (out of the 106 projects) with missing values:

- the 62 projects with missing effort in the 'implement' phase, and

[2] By default, SAS creates 5 imputed datasets.

- the 3 projects with missing values in the ‘plan’ phase (see the shaded areas in Figure 5.2).

For the first imputation, those 65 projects having missing value, the imputation was only in the column that has missing values.



Imputation 1

Input	Plan	Specify	Build	Test	Implement
1	635	2086	479	7130	972
1	234	296	624	1124	1196
1	242	492	279	432	513
1	257	534	1292	2767	195
1	7	30	391	44	84
1	936	1248	4680	624	1130
1	468	1256	2714	2254	542
1	210	803	4790	1618	469
1	234	312	624	156	695
1	16	62	281	39	471
1	437	1092	1388	936	589
1	390	2603	2527	1824	472
1	32	1	286	107	141
1	776	70	1085	280	691
1	455	140	560	140	68
1	560	278	1274	552	438
1	20	20	815	21	242
1	632	1053	7036	4586	461
1	42	47	543	145	44
1	30	20	233	840	773
1	66	70	662	352	47
1	624	2964	2808	7800	869
1	78	390	117	156	458
1	16	16	311	31	432
1	35	70	130	50	72
1	468	2964	2028	3822	917
1	323	417	1193	867	372
1	59	9	256	114	121
1	196	10	54	20	92
1	160	120	470	280	46
1	1270	237	1165	945	1019
1	546	2028	3120	2574	903
1	156	936	624	499	337
1	554	582	1626	3342	117
1	504	1267	1624	2996	634
1	3915	584	1760	4340	2493
1	484	874	2604	1170	330
1	323	440	1369	977	337
1	77	45	354	149	230
1	2094	760	2883	3517	1424
1	60	100	910	250	303
1	95	15	294	618	272
1	554	635	753	2298	391
1	50	40	510	100	100
1	663	1638	1482	1630	75
1	1338	1844	1760	3938	1888
1	468	2496	2184	3744	1131
1	657	998	5889	1217	586
1	767	491	1807	1632	526
1	359	858	2543	234	457
1	468	1833	1560	1594	634
1	480	1746	1747	2329	888
1	515	1123	1212	5466	614
1	671	1326	1201	1607	248
1	543	658	1374	969	596
1	172	108	692	162	498
1	825	1652	2207	1032	1205
1	5390	7910	15078	14196	4884
1	78	47	390	47	306
1	140	212	478	14	64
1	624	1872	3276	5413	52
1	170	264	30	298	246
1	8038	28665	48574	15005	8285
1	877	168	2080	976	144
1	164	207	856	111	56

Figure 5.2 Sample result of the multiple imputation method – step 1

5.2.1.2 Analysis of variance information and parameter estimates for the Implement effort and Plan effort imputed values following MI

This section presents the output results of the variance information and parameter estimates for MI based on 106 projects (or 103 projects after removal of the outliers): these are used to generate valid statistical inferences about the depended variables (Effort Plan and Effort Implement).

In addition, the imputed values of MI will show the mean of the 5 imputed datasets, which are the mean of 5 imputations and the standard error of the mean for Effort Implement and Effort Plan estimation. The tables also display a 95% mean confidence interval and a t-test with the associated P-value: these inferences are based on the t-distribution.

After the completion of *m* imputations, the "Multiple Imputation Variance Information" is displayed in Table 5.1 and Table 5.2 with the variances between imputations (Bm) and within imputations $\overline{U}m$, and the total variances when combining completed data inferences respectively.

For instance, for the 5 imputed datasets with 106 projects, the combined results of the Effort Implementation (EI) variable, give in table 5.1 a Mean of $\overline{P}m$ = 541 hrs, a variance within imputations $\overline{U}m$ = 8454hrs, a variance between imputations Bm= 2144hrs, and M=5 imputations, (1+1/m) =1.2.

Total variance Tm is = 8454 + 1.2*2144= 11028hrs, and the SE result is = $\sqrt{11028}$ = 105 hrs.

Table 5.1 Variance information for imputed values of Effort Plan and Effort Implement (N=106 projects)

Variable	N=106 Projects, before removal of outliers								
	Mean $\overline{P}m$	Std Error	95% Confidence Limits		T- test	Variance			P- Values
						Between Bm	Within $\overline{U}m$	Total	
EP	573 hrs	106 hrs	364 hrs	783 hrs	5.42	99	11066	11184	<.0001
EI	541 hrs	105 hrs	328 hrs	753 hrs	5.15	2144	8455	11028	<.0001

Table 5.2 Variance information for imputed values of Effort Plan and Effort Implement (N=103 projects, without 3 outliers)

Variable	N=103 Projects, after removal of 2 outliers								
	Mean $\overline{P}m$	Std Error	95% Confidence Limits		T-test	Variance			P-Values
						Between Bm	Within $\overline{U}m$	Total	
EP	448 hrs	60 hrs	330 hrs	567 hrs	7.50	15	3562	3598	<.0001
EI	395 hrs	73 hrs	221 hrs	569 hrs	5.38	3030	1747	5383	<.0001

Considering that the P-values in Table 5.1 and Table 5.2 are both <0.1, it can be concluded that by removing outliers the variance results of the standard error of the imputed values have decreased from 105 hours to 73 hours for the Effort Implement model, decreased from 106 hours to 60 hours for the Effort Plan model. As well, the results are statistically significant at t-test and P-values with and without outliers for the Effort Plan and Effort Implement estimates (see Table 5.3).

Table 5.3 Summary of imputed values for Effort Plan and Effort Implement

Variable	Before removal of outliers N=106 projects		After removal of 3 outliers N=103 projects	
	Significant T-test	Significant P-values	Significant T-test	Significant P-values
EP	Yes	Yes	Yes	Yes
EI	Yes	Yes	Yes	Yes

5.2.1.3 Analysis of average effort by phase after MI based on seeds selected with outliers (N=106 projects)

Tables 5.4 to 5.8 display the averages of the effort distribution by phases for the three profiles (PSBTI, PSBT and SBTI), and for each of the five imputations round. Of course, for the PBSTI profile without missing values, the average distribution is the same in each Table, while for the other two profiles; the averages will vary across the five rounds of imputations, (See Appendix VIII on the CD attached to this thesis).

Tables 5.4 to 5.8 present within parenthesis the averages of the value imputed based on the seeds selected within the ranges of values which included the outliers, that is the min and max = (2, 5390) 'Effort Plan' in the SBTI profile and the 'Effort Implement' the min and max=(20, 8285) in the PSBT profile.

Table 5.4 Average effort distribution per phase (1st Imputation) N=106 Projects

Profile	Project Phases - % Effort					# Projects
	Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
PSBTI	9.1	24.7	39.1	19.7	7.3	**41**
PSBT	9.9	16.3	30.8	32.0	(11.0)	**62**
SBTI	(7.9)	25.4	45.1	14.1	7.4	**3**

Table 5.5 Average effort distribution per phase (2nd Imputation) N=106 Projects

Profile	Project Phases - % Effort					# Projects
	Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
PSBTI	9.1	24.7	39.1	19.7	7.3	**41**
PSBT	10.2	16.6	31.5	32.7	(9.0)	**62**
SBTI	(9.1)	25.1	44.6	13.2	7.3	**3**

Table 5.6 Average effort distribution per phase (3rd Imputation) N=106 Projects

Profile	Project Phases - % Effort					# Projects
	Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
PSBTI	9.1	24.7	39.1	19.7	7.3	**41**
PSBT	10.2	16.7	31.7	32.9	(8.4)	**62**
SBTI	(7.8)	25.5	45.2	14.1	7.4	**3**

Table 5.7 Average effort distribution per phase (4th Imputation) N=106 Projects

Profile	Project Phases - % Effort					# Projects
	Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
PSBTI	9.1	24.7	39.1	19.7	7.3	**41**
PSBT	9.9	16.2	30.7	31.9	(11.2)	**62**
SBTI	(6.9)	25.7	45.6	14.2	7.5	**3**

Table 5.8 Average effort distribution per phase (5th Imputation) N=106 Projects

Profile	Project Phases - % Effort					# Projects
	Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
PSBTI	9.1	24.7	39.1	19.7	7.3	**41**
PSBT	9.9	16.3	30.8	32.0	(11.0)	**62**
SBTI	(7.1)	25.7	45.5	14.2	7.5	**3**

In summary it can then be observed that:

- The averages for imputation 1 are: Effort Plan = (7.9%) for the SBTI profile and Effort Implement = (11.0%) for the PSBT profile (See Table 5.4 and Table 5.9).
- The averages for imputation 2 are: Effort Plan = (9.1%) for the SBTI profile and Effort Implement = (9.0%) for the PSBT profile (See Table 5.5 and Table 5.9).
- The averages for imputation 3 are: Effort Plan = (7.8%) for the SBTI profile and Effort Implement = (8.4%) for the PSBT profile (See Table 5.6 and Table 5.9).
- The averages for imputation 4 are: Effort Plan = (6.9%) for the SBTI profile and Effort Implement = (11.2%) for the PSBT profile (See Table 5.7 and Table 5.9).
- Finally, the averages for imputation 5 are: Effort Plan = (7.1%) for the SBTI profile and Effort Implement = (11.0%) for the PSBT profile (See Table 5.8 and Table 5.9).

Table 5.9 Comparison across the imputations with outliers (N=106 projects)

# Imputation	%Effort Plan in SBTI profile	%Effort Implement in PSBT profile
1st Imputation	7.9%	11.0%
2nd Imputation	9.6%	8.2%
3rd Imputation	9.5%	7.9%
4th Imputation	9.2%	9.4%
5th Imputation	9.2%	9.3%

Of course, the relative effort distribution of the other phases has varied accordingly (Effort Plan, Effort Specification, Effort Build, and Effort Test) for the PSBT and SBTI profiles in each imputation.

Table 5.10 combines next for each imputation round the data from all the projects, including the 41 of the PBSTI profile, which already had all data and the 62 projects in the PSBT and 3 projects in the SBTI profiles which had missing data in one phase for the 106 projects. Some variations of course can be observed across the 5 imputation steps: for instance, the distribution of effort in the 'implement' phase varies from 7.9% to 9.4% on the set of 106 projects, for an average of 8.8%.

Table 5.10 Average effort distribution for the 5 imputation (N=106 projects)

# Imputation result	Project Phases – % of total Effort					Total
	Effort Plan	Effort Specify	Effort Built	Effort Test	Effort Implement	
1st Imputation	9.3	20.5	35.9	25.1	9.3	100%
2nd Imputation	9.6	20.6	36.2	25.3	8.2	100%
3rd Imputation	9.5	20.8	36.4	25.5	7.9	100%
4th Imputation	9.2	20.5	35.9	25.1	9.4	100%
5th Imputation	9.2	20.5	36	25.1	9.3	100%
Average of the 5 imputations	**9.4**	**20.6**	**36.1**	**25.2**	**8.8**	**100%**

5.2.1.4 Analysis of average effort after MI based on seeds selected excluding outliers

In the previous subsection 5.2.1.1, the generation of the random numbers for the imputed values was based on the minimum and maximum values of the seeds of the 106 projects which included 3 outliers. More specifically, the maximum seed was 5,390 hours for the Effort Plan and the maximum seed was 8,285 for the Effort Implementation.

This section present now a multiple imputation for the dataset without the 3 outliers identified in chapter 4 – see Table 4-5. When these outliers are taken out, the maximum seeds will change of course: therefore, without outliers, the new maximum seed is 3915 hours for Effort Plan and the maximum seed is 2946 hours for the Effort Implementation.

Tables 5.11 to 5.15 present within parenthesis the averages of the values imputed based on seeds selected within the ranges of values which excluded outliers, that is for the 'Effort Plan' in the SBTI profile and the 'Effort Implement' in the PSBT profile, (See Appendix XI on the CD attached to this thesis). That is:

Table 5.11 Average effort distribution per phase (1st Imputation) N=103 Projects

Profile	Project Phases - % Effort					# Projects
	Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
PSBTI	10.3	23.9	36.8	21.1	8.0	**40**
PSBT	9.8	16.3	30.9	32.5	(10.5)	**61**
SBTI	(20.8)	6.5	50.5	18.7	3.4	**2**

Table 5.12 Average effort distribution per phase (2nd Imputation) N=103 Projects

Profile	Project Phases - % Effort					# Projects
	Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
PSBTI	10.3	23.9	36.8	21.1	8.0	**40**
PSBT	10.1	16.8	31.8	33.5	(7.9)	**61**
SBTI	(12.4)	7.1	55.9	20.7	3.8	**2**

Table 5.13 Average effort distribution per phase (3rd Imputation) N=103 Projects

Profile	Project Phases - % Effort					# Projects
	Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
PSBTI	10.3	23.9	36.8	21.1	8.0	**40**
PSBT	10.1	16.9	32.0	33.7	(7.2)	**61**
SBTI	(20.4)	6.5	50.8	18.8	3.5	**2**

Table 5.14 Average effort distribution per phase (4th Imputation) N=103 Projects

Profile	Project Phases - % Effort					# Projects
	Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
PSBTI	10.3	23.9	36.8	21.1	8.0	**40**
PSBT	9.7	16.1	30.6	32.2	(11.3)	**61**
SBTI	(6.7)	7.6	59.6	22.1	4.1	**2**

Table 5.15 Average effort distribution per phase (5th Imputation) N=103 Projects

Profile	Project Phases - % Effort					# Projects
	Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
PSBTI	10.3	23.9	36.8	21.1	8.0	**40**
PSBT	9.8	16.3	30.9	32.6	(10.3)	**61**
SBTI	(19.9)	6.5	51.1	18.9	3.5	**2**

In summary of the 5 imputations:

- The averages for imputation 1 are: Effort Plan = (20.8%) for the SBTI profile and Effort Implement = (10.5%) for the PSBT profile (See Table 5.11 and Table 5.16).
- The averages for imputation 2 are: Effort Plan = (12.4%) for the SBTI profile and Effort Implement = (7.9%) for the PSBT profile (See Table 5.12 and Table 5.16).
- The averages for imputation 3 are: Effort Plan = (20.4%) for the SBTI profile and Effort Implement = (7.2%) for the PSBT profile (See Table 5.13 and Table 5.16).

- The averages for imputation 4 are: Effort Plan = (6.7%) for the SBTI profile and Effort Implement = (11.3%) for the PSBT profile (See Table 5.14 and Table 5.16).
- Finally, the averages for imputation 5 are: Effort Plan = (19.9%) for the SBTI profile and Effort Implement = (10.3%) for the PSBT profile (See Table 5.15 and Table 5.16).

Table 5.16 Comparison across the importations without outliers (N=103 projects)

# Imputation	%Effort Plan in SBTI profile	%Effort Implement in PSBT profile
1st Imputation	20.8%	10.5%
2nd Imputation	12.7%	7.9%
3rd Imputation	20.4%	7.2%
4th Imputation	6.7%	11.3%
5th Imputation	19.9%	10.3%

Table 5.17 combines next for each imputation round the data from all the projects, including the 40 of the PBSTI profile which already had all data and the 61 projects in the PSBT and 2 projects in the SBTI profiles which had missing data in one phase that the effort average by phases for the 103 projects. Some variations of course can be observed across the 5 imputation steps: for instance, the distribution of effort in the 'implement' phase varies from 7.9% to 10.1% -See Table 5.17.

Table 5.17 Profiles of Average effort distribution for N=103 projects, excluding outliers

# Imputation result	Project Phases – % of total Effort					Total
	Effort Plan	Effort Specify	Effort Built	Effort Test	Effort Implement	
1st Imputation	10.1	18.9	33.2	28.2	9.5	100%
2nd Imputation	10.2	19.3	33.9	28.8	7.9	100%
3rd Imputation	10.3	19.3	34.0	28.9	7.5	100%
4th Imputation	9.9	18.8	33.1	28.1	10.1	100%
5th Imputation	10.1	18.9	33.3	28.3	9.4	100%
Average of the 5 imputations	**10.1**	**19.0**	**33.5**	**28.5**	**8.9**	**100%**

5.3 Step 2 analyzing the completed data sets

5.3.1 Analysis strategy

Once the MI techniques have replaced missing values with multiple sets of simulated values to complete the data, the regression analysis procedure PROC REG is used in this step with each completed dataset to obtain estimates and standard errors, which adjusts the parameter estimates obtained from PROC MI for missing data.

In this step, the results of the regression analysis estimation models for the imputed values before and after removing the outliers are presented, this time trained with the 5 imputed datasets, and N= 65 (and 62 projects excluding outliers).

The objective in using this procedure is to obtain an analysis of the imputed dataset based on linear regression models, that is:

- to estimate the dependent variables with the missing values (i.e. Effort Plan and Effort Implement)
- on the basis of the independent variables (i.e. Effort Specify, Effort Build, Effort Test) that have observed values.

For the evaluation of the accuracy performances of the estimation models, this section presents the percentage of variation in the dependent variable explained by the independent variables of the model using the adjusted R^2 that accounts for the number of independent variables in the regression model.

Figure 5.3 illustrates how to build the regression analysis estimation models and obtained the analysis results to use them in next step3:

a) Use each completed datasets from step 1;

b) Execute PROC REG;

c) Build an estimation regression models for each completed dataset from MI;

d) Obtain an analysis of the imputed dataset based on linear regression models;

e) The combination of the analysis results obtained in this step will be used for step 3.

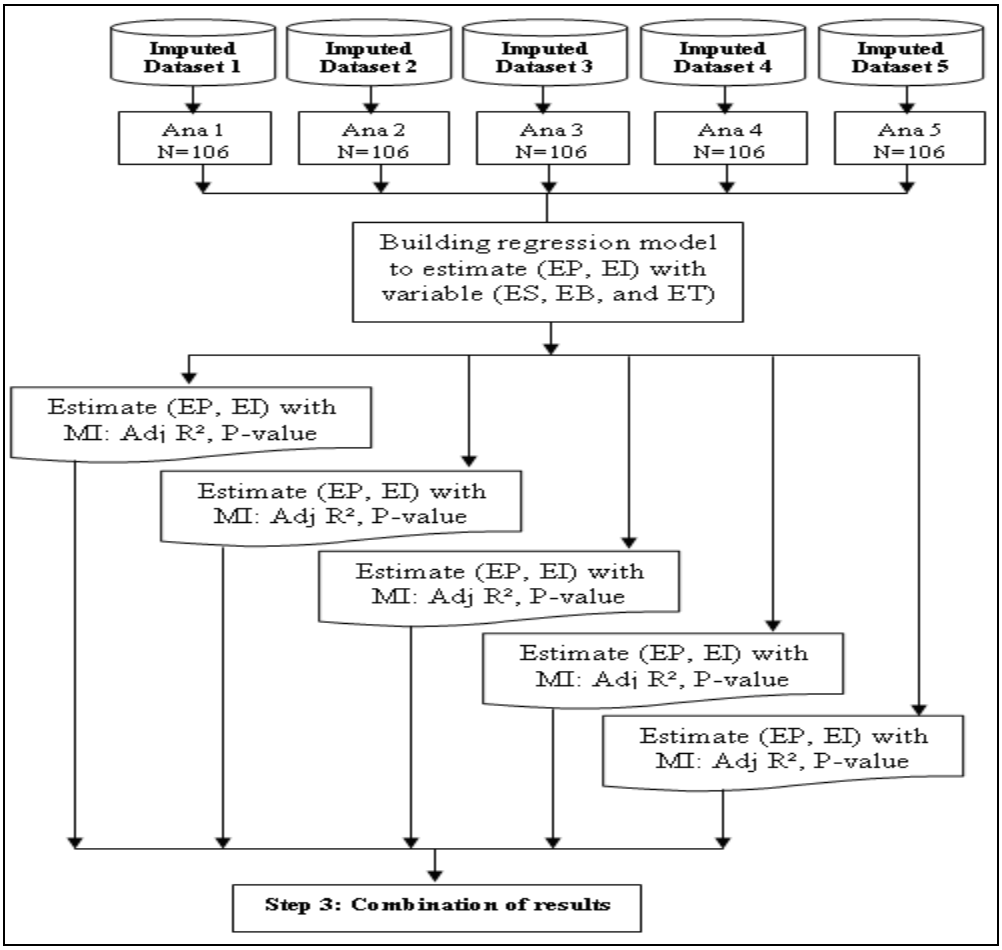


Figure 5.3 Building the regression analysis estimation models

5.3.2 Implement effort estimation model (using the 62 imputed Implement values)

To build an estimation model of the Implement effort, a multi-regression analysis is done using:

A) Independent variable: Effort Implement using:

1) The *actual* implement effort of the 41 projects from the PSBTI profile

2) The ***imputed*** implement effort of the 62 projects from the PSBT profile;

3) The *actual* implement effort of the 3 projects from the SBTI profile.

B) Dependent variables: Effort Specify, Effort Build and Effort Test

Table 5.18 and Table 5.19 present the results of the regression estimation model for the dependent variable (Effort Implement) trained with the independent variables (Effort Specify, Effort Build, and Effort Test) for each of the five imputations and based on 106 projects (with outliers). However there is more than one independent variable in the model: in this case the adjusted R^2 values are the selected method for the comparison. For instance, in Table 5.18, the parameter estimates for the Effort Implement model in the first line are: (87, 0.02, 0.1, and 0.15). Therefore, the regression equation for predicting the dependent variable from the independent variables is:

Effort Implement = 87 hours + 0.02 x Effort Specify + 0.1 x Effort Build +0.15x Effort Test.

Table 5.18 and Table 5.19 also show the coefficients of determination (i.e. R^2 and Adjusted R^2) for the regression model for each imputation. For instance, for the Model of Effort Implement, the adjusted R^2 obtained for each of the five imputations with outliers is (0.79, 0.80, 0.80, 0.81, and 0.81) in Table 5.18, and (0.28, 0.09, 0.14, 0.35, and 0.39) in Table 5.19 without outliers. Moreover, the regression analysis results for the estimation models present a statistically significant P-value in each of the 5 imputations of <0.0001.

The major differences in adjusted R^2 with and without outliers are (51%, 71%, 66%, 46%, and 42%). It can be observed also that the large number of missing values in the Effort Implement caused a major difference in the results without outliers of regression models, for each of the 5 imputations. These indicate that the outliers in each imputations have an undue influence on the estimation models.

Table 5.18 Regression analysis estimation model for Effort Implement based on the 5 imputed datasets (N=106 projects, with outliers)

Imputation No.	N=106 Projects, with outliers (Effort Implement) Model N=65						
	Intercept	Effort Specify	Effort Build	Effort Test	Adjusted R^2	R^2	P-value
1	87	0.02	0.1	0.15	0.79	0.80	<0.0001
2	66	0.01	0.12	0.1	0.80	0.81	<0.0001
3	106	0.04	0.12	0.04	0.80	0.81	<0.0001
4	117	0.03	0.1	0.13	0.81	0.82	<0.0001
5	154	0.02	0.12	0.08	0.81	0.82	<0.0001

Table 5.19 Regression analysis estimation model for Effort Implement based on the 5 imputed datasets (N=103 projects, without outliers)

Imputation No.	N=103 Projects, without outliers (Effort Implement) Model, N=62						
	Intercept	Effort Specify	Effort Build	Effort Test	Adjusted R^2	R^2	P-value
1	170	0.008	0.10	0.08	0.28	0.30	<0.0001
2	189	-0.002	0.08	0.03	0.09	0.11	<0.0001
3	194	0.07	0.09	-0.04	0.14	0.17	<0.0001
4	168	0.0004	0.06	0.16	0.35	0.37	<0.0001
5	138	-0.008	0.09	0.11	0.39	0.41	<0.0001

5.3.3 Plan effort estimation models (built using the 3 imputed Plan values)

To build an estimation model of the Plan effort, a multi-regression analysis is done using

A) Independent variable: Plan effort using:

1) The *actual* plan effort on the 41 projects for the PSBTI profile
2) The *actual* plan effort of the 62 projects from the PSBT profile
3) The ***imputed*** plan effort of the 3 projects from the SBTI profile.

B) Dependent variable: Specify effort, Build effort and Test effort

Table 5.20 presents next the results of the estimation models for the dependent variable (Effort Plan) trained with the independent variables (Effort Specify, Effort Build, and Effort Test) for each of the five imputations and based on 106 projects (with outliers).

For instance, in Table 5.20, the parameter estimates for the Effort Plan model in the first line are: (44, -0.13, 0.17, and 0.20), and the regression equation for predicting the dependent variable from the independent variables is:
Effort Plan = 44hours - 0.13x Effort Specify + 0.17 x Effort Build +0.20 x Effort Test.

Table 5.20 and Table 5.21 also show the coefficients of determination (i.e. R^2 and Adjusted R^2) for the regression model for each imputation. For instance in Table 5.20, for the model of Effort Plan, the adjusted R^2 obtained for each of the five imputations with outliers are (0.76, 0.80, 0.77, 0.74, and 0.75), and without outliers (0.33, 0.34, 0.34, 0.34, and 0.33) respectively.

Moreover, the regression analysis results for the estimation models present a statistically significant P-value in each of the 5 imputations of <0.0001.

Table 5.20 Regression analysis estimation model for Effort Plan based on the 5 imputed datasets (N=106 projects, with outliers)

Imputation No.	N=106 Projects, with outliers						
	(Effort Plan) Model N=3						
	Intercept	Effort Specify	Effort Build	Effort Test	Adjusted R^2	R^2	P-value
1	44	-0.13	0.17	0.20	0.76	0.77	<0.0001
2	2	-0.08	0.19	0.18	0.80	0.81	<0.0001
3	29	-0.12	0.18	0.20	0.77	0.78	<0.0001
4	55	-0.14	0.16	0.21	0.74	0.75	<0.0001
5	41	-0.13	0.17	0.21	0.75	0.76	<0.0001

Table 5.21 Regression analysis estimation model for Effort Plan based on the 5 imputed datasets (N=103 projects, without outliers)

Imputation No.	N=103 Projects, without outliers						
	(Effort Plan) Model, N=2						
	Intercept	Effort Specify	Effort Build	Effort Test	Adjusted R^2	R^2	P-value
1	86	-0.09	0.17	0.14	0.33	0.35	<0.0001
2	76	-0.08	0.18	0.14	0.34	0.36	<0.0001
3	82	-0.09	0.18	0.14	0.34	0.36	<0.0001
4	72	-0.08	0.17	0.13	0.34	0.36	<0.0001
5	85	-0.09	0.17	0.14	0.33	0.35	<0.0001

It can be observed that the adjusted R^2 is lower for the dataset without outliers, indicating that the outliers unduly influence the estimation models, leading to statistical overconfidence in the results (that is, the results in Table 5.18 and Table 5.20 are biased by the observed outliers), (See Appendix IX and XII on the CD attached to this thesis).

5.4 Step 3 Combining the analysis results (combination of results)

5.4.1 Strategy and statistical tests to be used

Step 3 presents the results of the parameter estimates for Effort Implement and Effort Plan estimation models previously trained on the full dataset with imputed values and before removing the outliers in N=106 (and N=103 projects after removing the outliers).

In this step, the results of the regression analysis estimation in Step 2 are combined, taking into account differences within datasets (variation due to the missing data) and between datasets (variation due to imputation).

The MI regression analysis procedure (PROC MIANALYZE) is used for combining the MI results. This step combines *m* sets of estimates and standard errors to obtain a single estimation model, standard error, and the associated confidence interval or significance test P-value.

The parameter estimates for MI displays a combined estimate and standard error for each regression coefficient (parameter). The inferences are based on *t*-test distributions, as well a 95% confidence interval and a t-statistic with the associated P-value.

The P-value is the number attached to each independent variable in an estimation model, which is that variable's significance level in the regression result. It is a percentage, and explains how likely it is that the coefficient for that independent variable emerged by chance and does not describe a real relationship.

A P-value of 0.05 means that there is a 5% chance that the relationship emerged randomly and a 95% chance that the relationship is real. It is generally accepted practice to consider variables with a P-value of less than 0.1 as significant.

There is also a significance level for the model as a whole, which is the F-value. This value measures the likelihood that the model as a whole describes a relationship that emerged at random, rather than a real relationship, as with the P-value, the lower the F-value, the greater the chance that the relationships in the model are real.

In addition, the t-statistic value is used to determine whether or not an independent variable should be included in a model. A variable is typically included in a model if it exceeds a predetermined threshold level or 'critical value'.

The thresholds are determined for different levels of confidence: e.g. to be 95% confident that a variable should be included in a model, or, in other words, to tolerate only a 5% chance that a variable doesn't belong in a model. A t-statistic greater than 2 (if the coefficient is positive) or less than -2 (if the coefficient is negative) is considered statistically significant.

5.4.2 The strategy for combining results

The strategy for combining results is as follows (Rubin, 1987)

A. Combine the results, taking into account differences within datasets (variances; uncertainty due to missing data) and between datasets (variances; additional uncertainty due to imputation).

B. Estimate the parameter ($\overline{\mathrm{P}}$), which is the mean across the *m* imputations. The mean of $\hat{\mathrm{P}}_{\mathrm{j}}$ is then given by $\overline{\mathrm{P}} = \sum_{j=1}^{m} \hat{P}_j$

C. Variances (within and between):

- Within: the imputation variance $\overline{\mathrm{U}}$ of the parameter $\overline{\mathrm{P}}$ is the mean of the variances across the *m* imputations.

- Between: the imputation variance B of the parameter $\bar{P}$ is the standard deviation of $\bar{P}$ across the *m* imputations.
- The total variance of $\bar{P}$ is a function of $\bar{U}$ and B , and is used to calculate the standard error used for test statistics.
- The variability of $\hat{P}_j$ is divided into two components:

  a) Within imputation variance $\bar{U}m = \frac{1}{m}\sum_{j=1}^{m} U_j$

  b) Between imputation variance $Bm = \frac{1}{m-1}\sum_{j}^{m}(\hat{P}_j - \bar{P}m)^2$

  c) Total variance $Tm = \bar{U}m + (1+\frac{1}{m})Bm$

D. Combine Standard Error results:

a) Variance of $\bar{P}m$:

$Var(\bar{P}m) = Tm = \bar{U}m + (1+\frac{1}{m})Bm$

$\bar{U}$ = Average of the 'within' variances

m = Correction for a finite number of imputations m

Bm = Variation in the *m* results; Variance of the *m* different parameters

b) Standard error (SE):

$SE(\bar{U}m) = \sqrt{Tm}$

5.4.3 Average parameter estimates for MI of the full imputed dataset (N= 106 and N=103)

This section presents the parameter estimates for MI of the full 5 imputed datasets before and after removal of the outliers: the results of the 5 imputed dataset estimates are combined and the averages of parameter estimates obtained using the results of the five estimation models in Step 2. This will allow to generate valid statistical inferences for estimated analysis of the dependent variables with 'missing values' (i.e. Effort Plan, and Effort Implement), on the independent variables observed values (Effort Specify, Effort Build, and Effort Test).

For instance, in Step 2, the results of 5 imputations for the intercepts for the Effort Implement are (87, 66, 106, 117, and 154) and Effort Plan is (44, 2, 29, 55, and 41) with outliers (see Table 5.18 and Table 5.20).

After combining the results, the average intercept estimate for Effort Implement without outliers of 172 hours – see Table 5.23 (with a Standard Error of 60 hours), and the average estimation for the intercept for Effort Plan (with outliers) will be an estimate of 34 hours – see Table 5.24, with a Standard Error of 45 hours (before outliers removal).

For example, the Standard Error in Table 5.24 is obtained as follows:

- Intercept estimate: $\overline{P}m$ =34hrs, within variance $\overline{U}m$ = 3704hrs, between variance Bm = 408hrs
- Total variance Tm = 3704 + 1.2*408= 4193hrs
- Standard Error: SE = $\sqrt{4193}$ = 45hrs.

Table 5.22 Averages of parameter estimates of MI for Effort Implement (N=106)

Parameter	N=106 Projects, before outlier removal								
	Estimate	Std Error	95% Confidence interval		T-Statistic	Variance			P-values
						Between Bm	Within $\overline{U}m$	Total	
Intercept	106	61	-18	229	1.75	1086	2367	3670	0.09
ES	0.03	0.05	-0.06	0.11	0.56	0.0001	0.002	0.002	0.57
EB	0.11	0.03	0.06	0.17	4.21	0.0001	0.001	0.001	<.0001
ET	0.10	0.5	-0.03	0.23	1.82	0.002	0.001	0.003	0.11

Table 5.23 Averages of parameter estimates of MI for Effort Implement (N=103 withoutoutliers)

Parameter	N=103 Projects, after removal of 2 outliers								
	Estimate	Std Error	95% Confidence interval		T-Statistic	Variance			P-values
						Between Bm	Within $\overline{U}m$	Total	
intercept	172	60	53	291	2.86	483	3028	3608	0.01
ES	0.01	0.06	-0.10	0.13	0.22	0.0001	0.002	0.003	0.82
EB	0.08	0.03	0.02	0.15	2.60	0.0003	0.001	0.001	0.01
ET	0.07	0.09	-0.16	0.30	0.77	0.006	0.001	0.007	0.48

Table 5.24 Averages of parameter estimates of MI for Effort Plan (N=106)

Parameter	N=106 Projects, before outlier removal								
	Estimate	Std Error	95% Confidence interval		T-Statistic	Variance			P-values
						Between Bm	Within $\overline{U}$m	Total	
Intercept	34	45	-93	162	0.53	408	3704	4193	0.60
ES	-0.12	0.06	-0.24	-0.004	-2.05	0.0005	0.003	0.004	0.06
EB	0.17	0.03	0.11	0.24	5.41	0.0001	0.001	0.001	<.0001
ET	0.20	0.03	0.13	0.27	5.88	0.0001	0.001	0.001	<.0001

Table 5.25 Averages of parameter estimates of MI for Effort Plan (N=103 without 3 outliers)

Parameter	N=103 Projects, after removal of 2 outliers								
	Estimate	Std Error	95% Confidence interval		T-Statistic	Variance			P-values
						Between Bm	Within $\overline{U}$m	Total	
intercept	80	75	-66	226	1.07	41	5512	5561	0.28
ES	-0.09	0.06	-0.21	0.04	-1.34	0.00001	0.004	0.004	0.18
EB	0.18	0.04	0.10	0.25	4.85	0.00002	0.001	0.001	<.0001
ET	0.14	0.04	0.07	0.22	3.75	0.00002	0.002	0.002	0.0002

Table 5.22 and Table 5.23 show the regression analysis of the EI parameter estimate. Table 5.24 and Table 5.25 show the regression analysis of the EP parameter. These tables show also that the P-values of EB and ET have a significant impact on effort (Effort Plan): the P-values are <0.0001, 0.11 with outliers and <0.0001, 0.0002 without outliers respectively. Also the P-values of EB have a significant impact on effort (Effort Implement): the P-values are <0.0001 and 0.01 respectively.

In Table 5.22 and Table 5.23 the independent variables of EB, and ET are not a significant predictor of the dependent variable of EI, and the variation in the dependent variable is not significantly explained by the independent variables, while the Table 5.24 and Table 5.25 also present a t-statistic of less than 2 and P-values greater than 0.05, which means that the independent variables of ES is not a significant predictor of the dependent variable of EP, and the variation in the dependent variable is not significantly explained by the independent variables, only for (ES).

Table 5.26 presents the results of the average estimate model of the Effort Plan after they have been combined, with and without outliers. The test of the null hypothesis P-value in Table 5.26 shows that, of the three variables (ES, EB, and ET), ES has a less significant impact on the Effort Plan estimate, while the P-value of EB and ET are much more statistically significant.

Table 5.26 Statistical significance of parameter estimates of Effort Plan

Parameter	Before outlier removal N=106 projects		After outlier removal N= 103 projects	
	Significant T- test	Significant P-values	Significant T- test	Significant P-values
Intercept	No	No	No	No
ES	No	No	No	No
EB	Yes	Yes	Yes	Yes
ET	Yes	Yes	Yes	Yes

Table 5.27 presents the results of the average estimate model of the Effort Implement after they have been combined, with and without outliers. The test of the null hypothesis P-value in Table 5.27 shows that, of the three variables (ES, EB, and ET), ES and ET have a less significant impact on the Effort Implement estimate, while the P-value of EB is much more statistically significant.

Table 5.27 Statistical significance of parameter estimates of Effort Implement

Parameter	Before outlier removal N=106 projects		After outlier removal N= 103 projects	
	Significant T- test	Significant P-values	Significant T- test	Significant P-values
Intercept	N0	No	Yes	Yes
ES	No	No	No	No
EB	Yes	Yes	Yes	Yes
ET	No	No	No	No

The estimated effect of the EP on the EB and ET parameters are (0.18 and 0.14) with a t-statistic equal to (4.85 and 3.75) Table 5.25 and Table 5.27, while the effect of EI on the EB is (0.08) with a t-statistic equal to (2.60) without outliers, and a P-value of (<0.0001 and 0.0002) – see Table 5.23.

Since the t-statistic is greater than 2 and the P-value less than 0.1, this can conclude that the effect of the EB and ET on the EP parameters and EB on EI parameter is statistically significant.

The results of the regression analysis with outliers in Table 5.24 – MI for Effort Plan:

- the effect of the EB and ET on the EP parameters are 0.17 and 0.20,
- with a t-statistic for EB of 5.41 and 5.88 for ET;
- the estimated EB and ET parameters are statistically significant with EP.

The results of the regression analysis with outliers in Table 5.22 – MI for Effort Implement: The effect of EB on the EI parameter is 4.21 (see Table 5.22), which is higher than 2, and a P-value of <0.0001, which is less than 0.1. Therefore, the EB parameter is statistically significant with EI.

While the estimated effect of the EI on ES, EB, and ET is 0.03, 0.11, and 0.10 respectively, with a t-statistic equal to 0.56, 4.21, and 1.82, P-values of 0.57, <0.0001, and 0.11 respectively Table 5.22 with outliers. The values of the t-statistic are less than 2, and so the intercept coefficient is not statistically significant. This means that the regression analysis results did not find evidence that EP has any impact on ES, but it does have an impact on EB, or ET. Moreover, the regression analysis results of EI did not find evidence that EI has any impact on ES, or ET, but it does have an impact on EB. This means that the results analysis with the missing data observations, indicating that the outliers unduly influence the estimation models, leading to over statistical confidence in the results.

5.5 Summary

This chapter has investigated the use of the multiple imputation (MI) technique with the ISBSG repository for dealing with missing values, and reported on its use. Five imputation rounds were used to produce parameter estimates which reflect the uncertainty associated with estimating missing data.

This chapter has also investigated the impact of MI in the estimation of the missing values of the effort variable by project phase using the ISBSG repository, and applied regression models, both with and without outliers, and examined their specific influence on the results.

This chapter determined the averages of the effort distribution by phase for three profiles (PSBTI, PSBT, and SBTI), and for each of the five imputation rounds. The PSBT profile presents a missing phase (Effort Implementation), and the SBTI profile presents a missing phase (Effort Plan), and, as a result, the average of the effort distributions of the other phases (Effort Specification, Effort Build, and Effort Test), as well as the combined average of the effort distribution of all the projects, varied accordingly in each imputation.

The regression analysis was trained with the five imputed datasets from 65 projects (with outliers) and 62 projects (without outliers). It was observed that the adjusted R^2 is lower for the dataset without outliers, indicating that the outliers unduly influenced the estimation models, leading to over statistical confidence in the results.

This chapter showed next:
A) the results of multiple imputation variance information, and
B) imputed values for the Effort Implement and Effort Plan variables over the five imputed datasets.

A. The results of this investigation revealed that the variance results of the standard error of the imputed values decreased from 105 hours to 73 hours for Effort Implement and from 106 hours to 60 hours for Effort Plan for a multiple regression analysis with and without outliers respectively – See Table 5.1 and Table 5.2.
B. Furthermore, the multiple regression analysis results were statistically significant for the Effort Plan and Effort Implement parameters, as illustrated by the t-test and P-values with and without outliers.

This chapter also presented the results of five effort estimation models that were combined with the five imputed dataset estimates, and obtained the averages of the parameter estimates. The results of this investigation have shown the results of three variables (ES, EB, and ET).

A. The P-value of the EB and ET variables statistically presented a much higher significant impact on the effort estimate than the ES variable.
B. The estimated effect of EP on the ES parameter was -0.12 respectively, with a t-statistic equal to -2.05 and P-values of 0.04 respectively. Note that the values of the t-statistic were less than 2 – See Table 5.24.
C. The estimated effect of the ES and ET on EI parameters was 0.03, and 0.10 respectively, with a t-statistic equal to 0.56 and 1.82 and P-values of 0.57, and 0.11 respectively. Note that the values of the t-statistic were also less than 2 – See Table 5.22.
D. The intercept coefficient is <u>not</u> statistically significant – see Table 5.22, Table 5.24, and Table 5.25.

This means that the multiple regression analysis results did not find evidence that ES and ET have any impact on the EI and EP parameters, but it does have an impact on the EB parameter.

Furthermore, removing the outliers strengthens the linearity of the data and decreases the number of errors present in the regression. It can be observed that the adjusted R^2 is lower for the dataset without outliers: this means that the results analysis with the missing data observations, indicating that the outliers unduly influence the estimation models, leading to over statistical confidence in the results.

CHAPTER 6

VERIFICATION OF THE CONTRIBUTION OF THE MI TECHNIQUE ON EFFORT ESTIMATION

6.1 Introducation

Traditional approaches for dealing with missing values can reduce or exaggerate statistical power, and each of these distortions can lead to invalid conclusions. Researchers in software engineering effort estimation must be aware of the biases that can be caused by techniques designed to handle missing or incomplete data.

To verify how good a model or technique is at estimating effort for imputed datasets, its predictive accuracy must be determined.

This chapter will look at two imputation techniques:

1. Imputed data from imputations based on average values of the Effort Implement.
2. Imputed data based on multiple imputations by random selection from min-max seeds.

For these two imputations techniques, two distinct approaches will be investigated:

a) Based only from the data within the field with missing values – this will be referred to as imputation from absolute values.
b) Based on imputation taking into account information from other data fields: here, the information from the data fields of Effort Plan, Effort Specify, Effort Build and Effort Test will be used to calculate the distribution of Effort Implement relative to the effort in the other project phases. This will be referred to as imputation from relative values.

Hence, for approach **a)** above, the null and alternative hypotheses of our research are the following:

- H0: When an estimation model is built from imputed data based on the absolute average values, we obtain predictive accuracy that is statistically significantly better than imputed data from MI imputations from absolute min-max seeds.
- H1: When an estimation model is built from imputed data based on the absolute average values, we do not obtain predictive accuracy that is statistically significantly better than imputed data from MI imputations from absolute min-max seeds.

Hence, for approach **b)** above, the null and alternative hypotheses of our research are the following:

- H2: When an estimation model is built from imputed data based on the relative average values, we obtain predictive accuracy that is statistically significantly better than imputed data from MI imputations from relative min-max seeds.
- H3: When an estimation model is built from imputed data based on the relative average values, we do not obtain predictive accuracy that is statistically significantly better than imputed data from MI imputations from relative min-max seeds.

This chapter focuses now on these other strategies for analyzing the predictive accuracy of estimation models on an MI dataset. The following three strategies have been designed to verify the predictive accuracy of estimation models from imputations:

A. Using a complete data set which will be split into two subsets:
    1) one subset X with the complete data values (i.e. no missing values), and
    2) a second subset Y from which data will be deleted and used for imputation purposes with the MI technique
    3) an MI estimation model of Effort Implement will be built using both subset A and the imputed subset B. .
    4) the performance of the MI estimation model will be compared with the performance of the estimation models built:
        i. From the full set of complete projects – sections 6.2.3 and 6.2.4, and
        ii. from the training subset of 20 projects – section 6.2.5.

B. Sensitivity analysis of MI when the seeds are changed from the absolute min-max values of Effort Implement to the min-max values of Effort Implement relative to Total Effort – see section 6.3.
C. Analysis of the estimation performance with MI in comparison to the average imputation technique used in a previous study (Déry et Abran, 2005) which used the average values of the data with values to substitute for the missing values. See section 6.4.

6.2 Strategy: creating artificially missing values from a complete dataset

6.2.1 Strategy steps

This new strategy for analyzing the performance of MI, is to work with a dataset not containing any missing value, creating artificially a subset by deleted a number of data values, and next comparing the estimation models derived from the original dataset and from the MI applied to the artificial subset with missing data.

In this chapter, the dataset selected consists again of the 41 projects with complete data values for the phase profile PBSTI, but does not use the other dataset with different profiles, as done in chapter 5. The specific verification strategy adopted in this research consists of:

- randomly splitting the data set into two subsets X and Y, and
- from subset Y, deleting the data values for the Effort Implement data field,
- replacing them with imputed values in subset Y.
- estimation models will be built with both the initial complete data set and the imputed dataset.
- assess the predictability of these estimation models based on the following criteria (Conte, Dunsmore et Shen, 1986) as presented in 1.6.2:
  (1) Magnitude of Relative Error (MRE) = | Estimated value – Actual value | / Actual
  (2) Mean Magnitude of Relative Error for *n* projects (MMRE) = 1/n*Σ(MREi)
  (3) Measure of Prediction Quality = Pred(x/100)

Figure 6.1 illustrates this specific strategy for investigating the contribution of the MI technique, given an initial dataset without missing values:

a) A random selection to split the initial complete data set into 2 subsets: subset X and subset Y;
b) In subset Y, create missing values artificially by deleting randomly data from a data field;
c) Select seeds from min & max from subset X;
d) assign random values to subset Y and create 5 imputed datasets (combining subsets X and Y imputed);
e) Combine imputation results;
f) Build a regression model to estimate Effort Implement (EI) based on the other four (4) project phases for:
    - The estimate with the complete initial dataset
    - The estimates with the dataset with imputed values,
g) Compare estimate by assessing and comparing the predictability with MMRE and Pred(25).

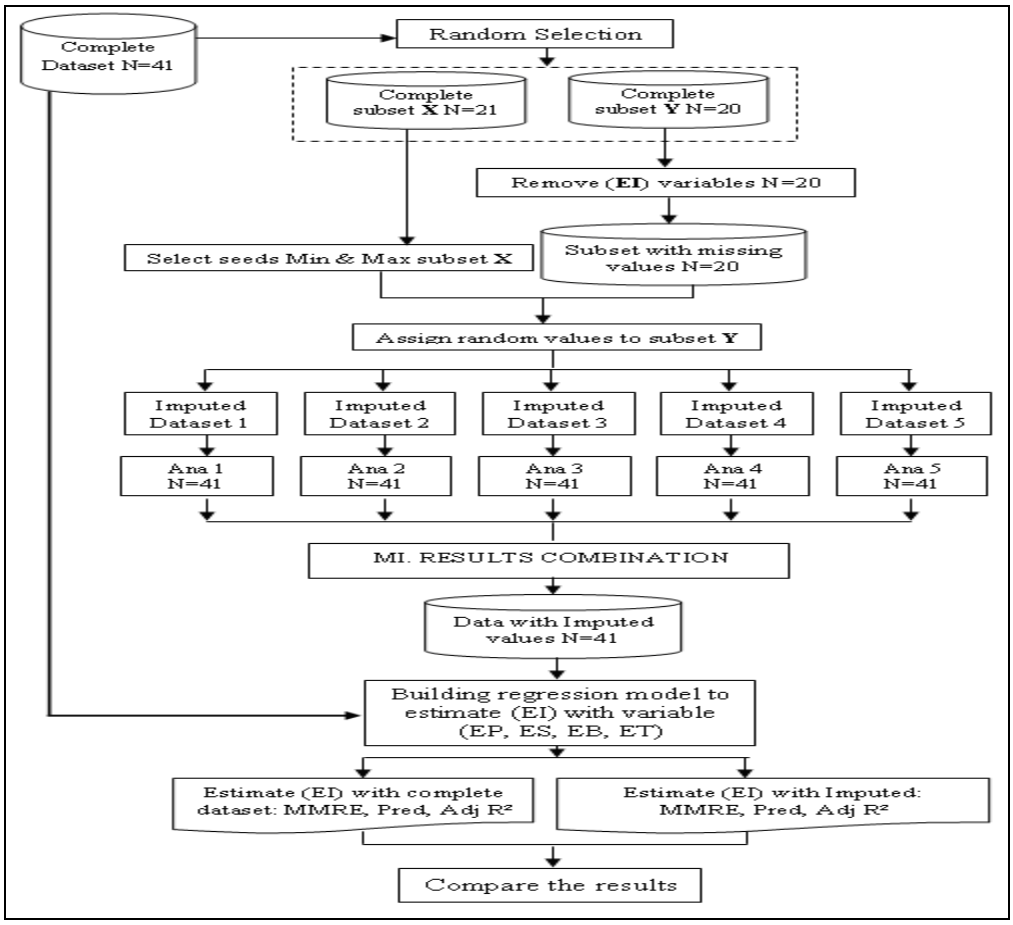


Figure 6.1 Strategy for analyzing the predictive accuracy of an MI dataset using a subset with values deleted

6.2.2 Impact on parameter estimates <u>with</u> outliers - N=41 and 21 projects with values deleted

This section presents the results of the estimation model for the dependent variable (Effort Implement) trained with the independent variables (Effort Plan, Effort Specify, Effort Build, and Effort Test) – see Table 6.1:

- for the complete dataset of 41 projects, including 1 outlier,
- for each of the five imputations (for the 41 projects, including the MI for the 21 missing values), and
- for the combined imputation model (See Appendix XXII on the CD attached to this thesis).

Table 6.1 Regression models for Effort Implement (N=41 projects, with outliers), before and after missing values were removed for N=21 projects

Dataset	N=41 projects, with outliers (Effort Implement) Model, N=41							
	Intercept	Effort Plan	Effort Specify	Effort Build	Effort Test	Adjusted R^2	R^2	P-value
Complete dataset	**45**	**0.57**	**0.10**	**-0.001**	**-0.06**	**0.58**	**0.62**	**<0.0001**
MI 1	368	0.49	-0.04	0.06	-0.05	0.36	0.42	0.0004
MI 2	246	0.49	0.07	0.009	-0.09	0.39	0.45	0.0002
MI 3	278	0.60	0.12	0.02	-0.22	0.46	0.51	<0.0001
MI 4	227	0.52	0.08	0.007	-0.10	0.41	0.47	<0.0001
MI 5	154	0.53	0.02	0.02	0.02	0.55	0.59	<0.0001
Combined imputations	**255**	**0.53**	**0.05**	**0.02**	**-0.09**	**0.45**	**0.46**	**<0.0001**

The adjusted R² for the estimation models derived from the combined imputations is 0.45 (bottom line of Table 6.1, which is is reasonably close to the adjusted R² of 0.58 of the estimation model from the complete dataset (top line of Table 6.1), considering that this dataset includes one outlier.

Table 6.1 presents also the P-value for the complete dataset (<0.0001), for each of the five imputations, as well as the P-value for the combined imputations (<0.0001). These P-values, which are all less than the 0.1 criterion for a P-value, indicate that they are statistically significant (i.e. a P-value of 0.05 means that there is a 5% chance that the relationship is real at the 95% confidence level for a P-value of less than 0.05).

6.2.3 Impact on parameter estimates without outliers – N = 40 and 20 projects with values deleted

In this section, the outlier has been excluded from the data set. This outlier is project id (9), which has a size of 2189 function points and an Effort Implement of 117 hours. Again it presents the results of the estimation model for the dependent variable Effort Implement trained with the independent variables Effort Plan, Effort Specify, Effort Build, and Effort Test for each of the five imputations and based on N=40 projects (i.e. without an outlier) - see Table 6.2, (See Appendix XXVII on the CD attached to this thesis).

Table 6.2 Regression models for Effort Implement (N=40 projects, without an outlier), before and after removing missing values for N = 20 projects

Dataset	N=40 projects, without outliers							
	(Effort Implement) Model N=40 projects							
	Intercept	Effort Plan	Effort Specify	Effort Build	Effort Test	Adjusted R^2	R^2	P-value
Complete dataset	**-7**	**0.67**	**0.15**	**-0.06**	**0.03**	**0.71**	**0.74**	**<0.0001**
MI 1	169	0.61	0.08	-0.04	0.05	0.52	0.57	<0.0001
MI 2	46	0.68	0.07	-0.04	0.13	0.71	0.74	<0.0001
MI 3	203	0.65	0.15	-0.04	-0.09	0.55	0.59	<0.0001
MI 4	41	0.66	0.11	-0.07	0.12	0.71	0.74	<0.0001
MI 5	-6	0.71	0.06	-0.06	0.23	0.76	0.78	<0.0001
Combined imputations	**91**	**0.66**	**0.09**	**-0.05**	**0.09**	**0.65**	**0.69**	**<0.0001**

Table 6.2 also presents the adjusted R^2 for:

- the complete dataset adjusted R^2 = 0.71, and,

– the five imputations adjusted R^2 =0.52, 0.71, 0.55, 0.71 and 0,76, and

- The combined imputation adjusted R^2 = 0.65.

This means that, after removing a single outlier, the adjusted R^2 increased in each of the five imputations, as well as in the combined imputation at 0.65, and comes even closer to the adjusted R^2 for the complete dataset (i.e. 0.71). All the models have a significant P-value <0.0001. The P-values are all less than the 0.1 criterion for a P-value, which indicates that they are statistically significant (a P-value of 0.05 means that there is a 5% chance that the relationship is real at the 95% confidence level for a P-value less than 0.05).

In summary, removing the outlier strengthened the linearity of the data and decreased the errors present in the regression. Furthermore, the results are statistically significant for the estimates of Effort Implement, as illustrated by the t-test and P-values with and without outliers.

6.2.4 Analysis of the variance of the estimates

This section presents an analysis of the variance between estimated effort and actual effort. Table 6.3 presents a summary of the predictive statistics used for each project estimate: MMRE, and Pred(25), to assess the results of the regression models of the Effort Implement estimation for the five imputed datasets and combined imputations.

These statistics make it possible to compare the performance of the estimation model based on combined imputations of MI for half of the dataset with the performance of the estimation model based on the complete dataset (with and without outliers).

Table 6.3 Verification results of the five imputed datasets for Effort Implement

Imputation No.	N=41 projects, with outliers		N=40 projects, without outliers	
	MMRE	Pred(25)	MMRE	Pred(25)
Complete dataset	**110%**	**32%**	**88%**	**30%**
MI 1	259%	24%	148%	18%
MI 2	173%	22%	101%	28%
MI 3	192%	15%	149%	23%
MI 4	169%	24%	82%	33%
MI 5	143%	22%	101%	33%
Combined imputations	**187%**	**23%**	**116%**	**27%**

Using the MMRE and Pred(25) criteria, it can be observed from Table 6.3, for the N=40 projects without outliers, that even though the adjusted R^2 were relatively high:

A. The quality of the estimation model built from the complete dataset is not very high with an MMRE = 88% and a Pred(25) = 30%.
B. With 50% of the data missing (i.e. 20 missing values in a sample of 40 projects without outliers), much larger MMRE error and worst Pred(25) would be expected, but with MI, the combined imputation model has only a relatively minor reduction in quality of the regression results: the MMRE at 116% is not that far from the 88 % MMRE of the complete dataset and the Pred(25) at 27% is very close (within 3%).

6.2.5 Additional investigation of effort estimation for N=20 projects with imputed values for the Effort Implement phase

This section discusses the comparison of the multiple imputation (MI) results with the results of the estimation model derived from only the training dataset A of 20 projects (instead of the estimation model in section 6.2.3 derived from the complete dataset of 40 projects).

In the previous section, the estimation model based on the complete data set of 40 projects was:
Effort Implement = **-7hrs + 0,67xEP + 0.15xES -0.06xEB + 0,03xET**

In this section the estimation model based only on the training subset A of 20 projects is:
Effort Implement = **-59hrs + 0,78xEP + 0.16xES -0.1xEB + 0,11xET**

This section presents the results of the estimation model for the dependent variable Effort Implement trained with the independent variables Effort Plan, Effort Specify, Effort Build, and Effort Test for each of the five imputations.

This section will also look at the quality of the analysis results of the EI estimation variance from estimation with imputed variance and training estimation model.

As in the previous section, to investigate the performance of MI and the results of the estimation model, this section uses as its basis the 40 projects of the ISBSG dataset for PSBTI profile without outliers, and divides it into 2 subsets – see Figure 6.4:

- Subset X of 20 of the 40 projects, which 20 have complete data fields;
- Subset Y of the other 20 projects from which is the information in the Effort Implement data field is deleted (this will be referred to as Subset Y with missing EI data).

This section builds the regression analysis for estimation model with only subset X of 20 projects with the PSBTI profile to be used as training dataset for building the estimation model, and then applies this estimation model to the subset Y with missing values of Effort Implement.

Therefore, Figure 6.2 illustrates our specific strategy for investigating the performance of MI and compared with the results of the estimation model:

a) Given the PSBTI dataset (without outlier N= 40 projects),
b) Build an estimation model for EI with subset X – the training data set,
c) Apply this EI estimation model from (b) on subset Y with missing values of (EI),
d) Analyzing the (EI) estimation variance on the Subset Y with imputed training dataset to assess the predictability with MMRE and Pred(25) on Subset Y,
e) Build an estimation model from combined 5 imputation datasets of MI N= 20 projects with missing values,
f) Analyzing the (EI) estimation variance with combined 5 imputations datasets of MI N=20 projects to assess the predictability with MMRE and Pred(25),
g) Compare the results of MMRE and Pred(25) of Subset Y with the results from the training dataset (il.e. subset X), and the combined 5 imputations datasets of MI.

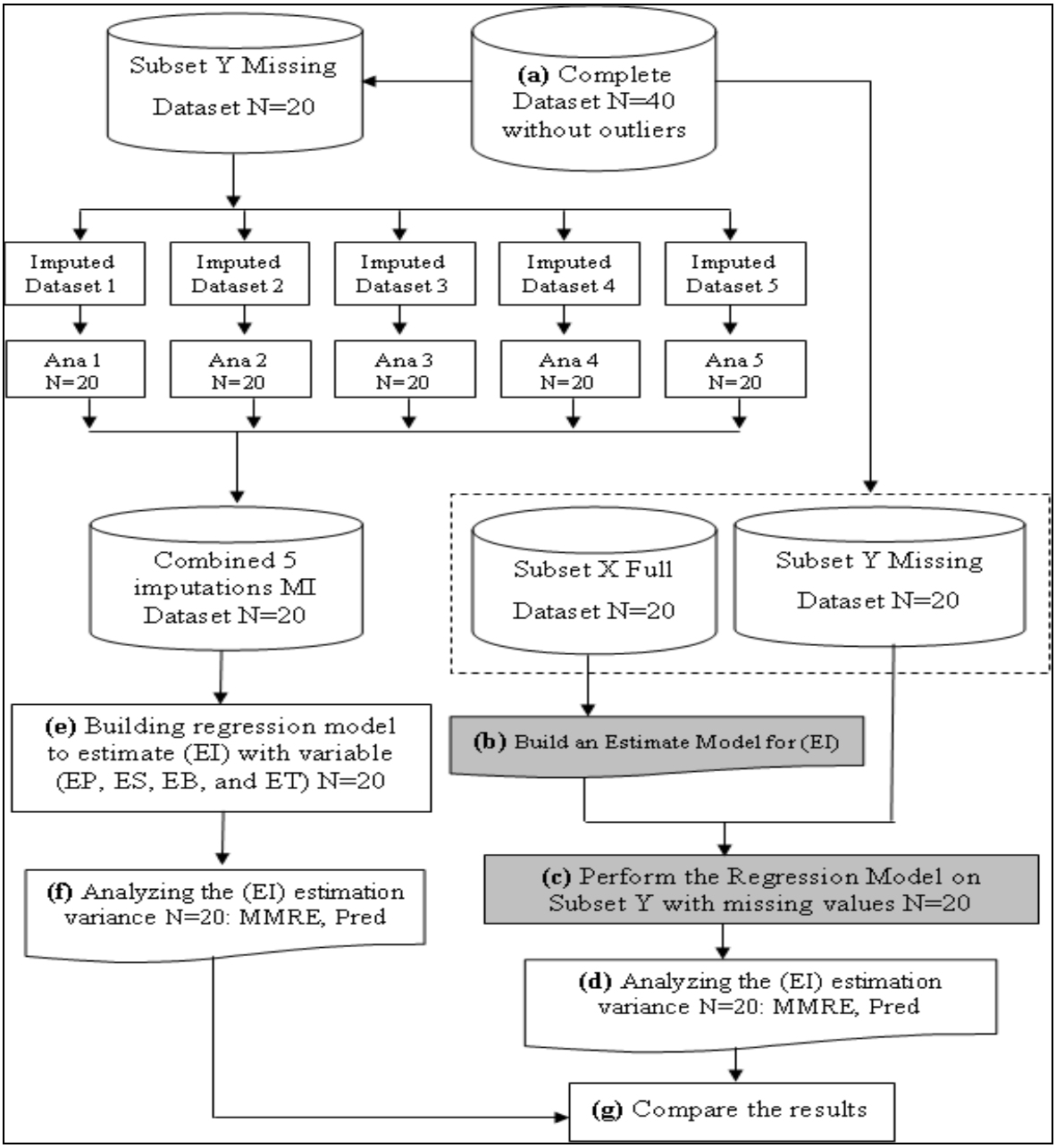


Figure 6.2 Modified strategy for the comparison with estimation models model trained with subset X of N= 20 projects

Table 6.4 presents the results of the three (3) multi regression estimation models built from:

- The training subset X of 20 projects.
- The combined imputed data set (40 projects: from subset X and the imputed data on subset Y).

Table 6.4 Regression models for Effort Implement (N=20 projects, without an outlier)

Dataset	N=20 projects, without outliers (Effort Implement) Model N=20 projects							
	Intercept	**Effort Plan**	**Effort Specify**	**Effort Build**	**Effort Test**	**Adjusted R^2**	**R^2**	**P-value**
Training Subset X	-59	0.78	0.16	-0.1	0.11	0.69	0.76	0.0002
Combined 5 imputations MI	247	0.23	0.07	-0.03	0.20	0.37	0.39	<0.0001

From Table 6.4 it can be observed that for the estimation models built:

- from the training subset X, the adjusted $R^2 = 0.69$;

- from the combined 5 imputations of MI, the adjusted $R^2 = 0.37$.

Table 6.5 Analysis of the EI estimation variance from estimation with imputed variance and training estimation model

No.	No. projects, without outliers	MMRE EI	Pred(25) EI
1	Training Subset X N=20	**97%**	**30%**
2	Combined 5 imputations MI N=20	**92%**	**40%**
3	Subset Y with missing values N=20	**118%**	**15%**
Comparison Results vs. complete data			
4	2 vs. 1	**-5%**	**+10**
5	3 vs. 1	**+21%**	**-15**
6	2 vs. 3	**-26%**	**+25**

It can be observed in Table 6-5 that:

- The MMRE is 97% and the Pred(25) is 30% for the performance of the estimation model derived from the training subset X – see line 1 in Table 6.5.
- The MMRE is 92% and Pred(25) is 40% for the performance of the estimation model derived from the combined 5 imputations of MI – see line 2 in Table 6.5.
- The MMRE is 118% and Pred(25) is 15% for the performance on subset Y derived from the estimation model derived from the training subset X – see line 3 in Table 6.5.

Compared to the performance of the estimation model built with the training Subset X:

- the performance of the combined 5 imputations results of MI represents an decrease in the MMRE of 5%, and a increase in the Pred(25) of 10% – See line 4 in Table 6.5.

6.3 Sensitivity analysis of relative imputation of effort estimation for N=40 projects without outliers for the Effort Implement phase

This section will look at the sensitivity of the analysis results when changing the basis for the imputations that is, changing the seed values from the absolute min and max values of Effort Implement to their relative min and max of Effort Implement with respect to total effort.

To investigate the sensitivity of the relative imputation and the absolute value, this section will use again the 40 projects of the PSBTI profile without outliers.

For the MI in sections 6.2.2 to 6.2.3, the imputed random numbers were generated using the (Min and Max), which was obtained from the Effort Implement variable only, in absolute values: for instance, the Min and Max for the Effort Implement variable were 20 hours, and 2,946 hours) respectively, from the 41 projects with the PSBTI profile, and including an outlier.

However, there is often a considerable variation of effort by phase at the project level: for instance, the variation of the ratio of Effort to Implement with Total Effort may vary considerably across projects. For example, in this dataset of N=40 projects without the outlier, the minimum percentage effort in the Implement phase is 1% of Total Effort, while the maximum percentage of Implement effort is 41% to Total Effort. Therefore, these %Min, and %Max obtained from values relative to Total Effort.

This section looks also at a second way of calculating the seeds values: instead of using the absolute max, calculate the relative value of Effort Implement for the project with this absolute max:

a) Identify the project with the absolute Min and Max effort in the Implement phase;
b) calculate for this specific project maximum percentage effort in the Implement phase relative to the other project phases for this project.

For this dataset, for the project with the absolute max of 2,946 hours in the Implement phase: for the project with this max of 2,946 hours, this absolute max represents 24% of the other phases combined. This project did not have however the maximum relative effort in the Implement Phase: another project had a 41% of its effort in the Implement phase.

There are different ways to obtain the relative Effort:

- with respect to total Effort which is: Total Effort= EP+ES+EB+ET+EI;
- Relative (%EI)*n*= ((absolute EI / ∑ (EP+ ES+EB+ET))X100)*n*.

Here, the second way was selected to calculate the relative min and max.

Therefore, Figure 6.4 illustrates our specific strategy for investigating the sensitivity of the analysis with the results of MI technique:

a) Given the dataset of the PSBTI profile without missing values N=40 projects;
b) Create missing values artificially by deleting data from a data field Effort Implement Subset B;
c) Identification and calculation of the relative (%EI)*n*= ((absolute EI / ∑ (EP+ ES+EB+ET))X100)*n*;
d) Generation of random numbers based on the relative seeds from N=40 projects;
e) Select the seeds (Min= 1%, Max= 41%) from relative values of (EI);
f) Build an estimation model from relative imputation results;
g) Select the seeds (Min= 1%, Max= 24%) from absolute min-max values of (EI);
h) Build an estimation model from Full dataset based on the seeds selected in (g);
i) Compare relative imputation results of MMRE and Pred(25) with the original dataset without missing values in (a);

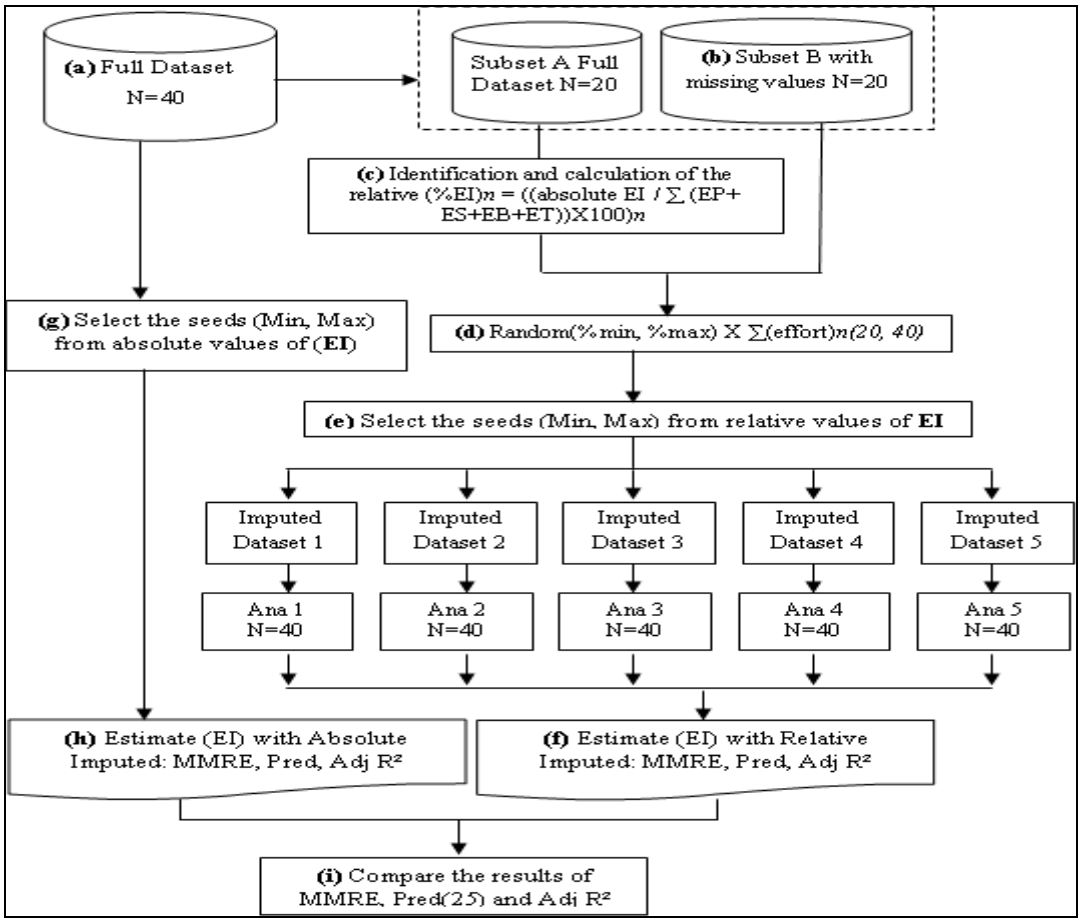


Figure 6.3 Specific strategy for investigating MI based on relative EI seeds

The strategy in this section consists of selecting Subset B N=20 projects, from the 40 complete projects with profile phase PBSTI. Table 6.6 presents multi-regression models as well as the adjusted R^2 (for the set of N=40 projects without outliers):

- the complete dataset: adjusted R^2 = 0.71;
- the combined imputation (relative seeds: %Min=1, %Max=24): adjusted R^2 = 0.60;
- the combined imputation seeds (relative seeds: %Min=1, %Max=41): adjusted R^2 = 0.53;
- as well as the adjusted R^2 after all the imputations combined is 0.65.

Table 6.6 Multi-regression models for Effort Implement (from MI with relative seeds for EI)

Dataset	N=40 projects, without outliers							
	(Effort Implement) Model N=40 projects							
	Intercept	Effort Plan	Effort Specify	Effort Build	Effort Test	Adjusted R^2	R^2	P-value
Complete dataset	-7	0.67	0.15	-0.06	0.03	0.71	0.74	<0.0001
Combined Imputed (relative seeds %1 to %24)	82	0.59	0.08	-0.01	0.06	0.60	0.64	<0.0001
Combined Imputed (relative seeds %1 to %41)	174	0.54	0.003	0.01	0.13	0.53	0.58	<0.0001
Combined imputations of MI absolute seeds	91	0.66	0.09	-0.05	0.09	0.65	0.69	<0.0001

This section presents the quality of the estimation models – see Table 6.7:

- the quality of the estimation model for the complete dataset with MMRE = 88% and Pred(25) = 30%;
- the estimation model with relative imputation for seeds (1% to 24%) - MMRE = (147%) and Pred(25) = (24%);
- the absolute imputation for seeds (1% to 41%) is MMRE = (172%) and Pred(25) = (23%);
- the combined results of MI with MMRE = 116% and Pred(25) = 27%.

Table 6.7 Contribution of relative imputation for N=40 projects with imputed values for the Effort Implement phase

Line No.	Imputation No.	N=40 projects, without outliers	
		%MMRE	Pred(25)
1	**Complete dataset**	88%	30%
2	**Combined Imputed (relative seeds %1 to %24)**	147%	24%
3	**Combined Imputed (relative seeds %1 to %41)**	172%	23%
4	**Combined imputations of MI absolute seeds**	116%	27%
Comparison Results vs. complete data			
6	**2 vs. 1**	+59%	-6%
7	**3 vs. 1**	+84%	-7%
8	**4 vs. 1**	+28%	-3%

In summary with the relative imputation, it can be observed in Table 6.7 that:

- Line 6: The difference for the combined relative imputed with seeds (%Min = 1% and %Max 24%) increases with MMRE = 59%, and decreases the Pred(25) = -6% compared with the complete data set;
- Line 7: the difference for the combined relative imputed with seeds (%Min = 1% and %Max 41%) increases with MMRE = 84%, and decreases the Pred(25) = -7% compared with the complete data set;
- Line 8: There is minor increase of 28% for the MMRE for the combined results of MI increases MMRE and a decrease of 3% in the Pred(25) compared to the complete dataset.

6.4 Comparing the estimation performance of MI with respect to a simpler imputation technique based on an average

This section presents a comparison of the multiple imputation (MI) results with the results from a simpler imputation technique based only on an average, as mentioned in the study of (Déry et Abran, 2005).

6.4.1 The (Déry et Abran, 2005) study

In (Déry et Abran, 2005) the missing values of Effort implement in the PSBT profile with missing EI were imputed by the average% of EI from the (PSBTI) profile.

This approach did not allow to verify the performance of estimation of EI for the PSBT profile with respect to their actual values, since the values for EI were missing in this PSBT profile. Therefore, to analyze the estimation performance of average imputation for missing values, the same strategy adopted in the previous sections is used here.

As with the previous strategies described in section 6.2 and 6.3, this section selects the same 41 projects from the previous section with the PSBTI profile with its complete dataset, and it uses the same split into exactly the same 2 subsets:

- Subset X of 21 projects of profile PSBTI (with 1 outlier), with complete data values and
- Subset Y of 20 projects with a PSBT profile with missing Effort Implement by deleting data from the data field of Effort Implement.

Figure 6.4 presents first the complete dataset N= 40 projects without outliers, and without missing values.

No	EP	ES	EB	ET	EI
1	3120	1540	5260	2299	2946
2	208	624	2075	761	528
3	2	38	91	15	24
4	80	80	80	40	20
5	1190	9793	17167	4489	1384
6	578	3412	3768	1540	1320
7	78	1435	640	1794	374
8	196	565	204	163	466
9	527	2278	1016	2335	260
10	94	1240	858	905	234
11	78	546	312	312	156
12	30	694	330	550	100
13	468	3160	1423	1493	577
14	283	4444	2837	1419	473
15	83	2339	826	880	606
16	78	437	156	109	78
17	47	1624	757	519	39
18	750	120	2110	800	140
19	2293	1048	7193	891	563
20	100	140	840	380	40
21	1810	156	4016	3014	790
22	150	310	1402	125	200
23	322	161	1979	622	168
24	140	242	414	140	23
25	120	288	592	372	92
26	330	200	380	400	90
27	140	100	4083	1025	82
28	85	255	1575	708	310
29	554	850	1215	3290	543
30	806	875	989	726	243
31	755	2976	5860	331	654
32	16	47	31	93	47
33	160	160	400	303	61
34	234	468	468	312	78
35	213	221	1864	296	150
36	16	156	311	117	47
37	300	520	1300	390	120
38	312	1248	1014	780	390
39	48	82	512	52	20
40	31	733	312	734	156

Figure 6.4 Sample of complete data N=40 projects

Figure 6.5 displays next the same dataset with N=40 projects (without the outlier) but now with missing values for subset Y, that is:

- Subset X with 20 complete projects;
- Subset Y with the 20 projects with missing effort in the 'implement' phase (see the shaded areas in Figure 6.5).

For subset X of 41 projects (including 1 outlier):

- The absolute average EI = 497 hours
- The absolute EI min = 20 hours
- The absolute EI max = 2946 hours
- The relative EI min = 1%
- The relative EI max = 41%

For subset X of 40 projects (excluding 1 outlier):

- The absolute average EI = 516 hours
- The absolute EI min = 20 hours
- The absolute EI max = 2946 hours
- The relative EI min = 1%
- The relative EI max = 41%

Input	EP	ES	EB	ET	EI
1	3120	1540	5260	2299	2946
2	208	624	2075	761	528
3	2	38	91	15	24
4	80	80	80	40	20
5	1190	9793	17167	4489	1384
6	578	3412	3768	1540	1320
7	78	1435	640	1794	374
8	196	565	204	163	466
9	527	2278	1016	2335	260
10	94	1240	858	905	234
11	78	546	312	312	156
12	30	694	330	550	100
13	468	3160	1423	1493	577
14	283	4444	2837	1419	473
15	83	2339	826	880	606
16	78	437	156	109	78
17	47	1624	757	519	39
18	750	120	2110	800	140
19	2293	1048	7193	891	563
20	100	140	840	380	40
21	1810	156	4016	3014	
22	150	310	1402	125	
23	322	161	1979	622	
24	140	242	414	140	
25	120	288	592	372	
26	330	200	380	400	
27	140	100	4083	1025	
28	85	255	1575	708	
29	554	850	1215	3290	
30	806	875	989	726	
31	755	2976	5860	331	
32	16	47	31	93	
33	160	160	400	303	
34	234	468	468	312	
35	213	221	1864	296	
36	16	156	311	117	
37	300	520	1300	390	
38	312	1248	1014	780	
39	48	82	512	52	
40	31	733	312	734	

Figure 6.5 Split of the 40 projects – withtout 1 outlier

Tables 6.8 and 6.9 display the average effort distribution of work effort across development phases, for the subsets X and Y and for the full dataset. The average effort distribution for the Effort Implement is calculated based on:

Effort Implement = Total EI / Total Effort (P+S+B+T+I))X100.

EI = 8.2% for subset X (excluding 1 outlier – Table 6-9)

EI = 6.7% for subset Y (excluding 1 outlier – Table 6-9)

Table 6.8 Average effort distribution by project phase including outliers (N=41 projects)

Imputation #no.	Profile	Project Phases – % Effort					No. of projects
		Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
Complete Data	**Subset X**	8.6	28.8	36.2	18.8	7.6	**21**
	Subset Y	10.3	15.8	45.3	21.8	6.7	**20**
	Subsets X&Y	9.1	24.7	39.1	19.7	7.3	**41**

Table 6.9 Average effort distribution by project phase excluding outliers (N=40 projects)

Imputation #no.	Profile	Project Phases – % Effort					No. of projects
		Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
Complete Data	**Subset X**	8.2	28.3	38.1	17.2	8.2	**20**
	Subset Y	10.3	15.8	45.3	21.8	6.7	**20**
	Subsets X&Y	8.9	24.1	40.5	18.8	7.7	**40**

6.4.2 Imputation based on an absolute average, %average, and MI with (absolute seeds and relative seeds Min & Max)

This section presents the results of the various types of imputations, from subset X as applied to subset Y, based on:

- the absolute average,
- the relative % average,
- MI with values selected randomly from:
  - absolute Min & Max EI seeds, and
  - relative Min & Max EI seeds .

For the imputation based on the absolute average of the 20 projects having missing values, the imputation is made only to the column that has missing values. For example, Figure 6.6 presents the imputation with the absolute average of 516 hours for EI – therefore the imputation of a constant value.

Input	EP	ES	EB	ET	EI
1	3120	1540	5260	2299	2946
2	208	624	2075	761	528
3	2	38	91	15	24
4	80	80	80	40	20
5	1190	9793	17167	4489	1384
6	578	3412	3768	1540	1320
7	78	1435	640	1794	374
8	196	565	204	163	466
9	527	2278	1016	2335	260
10	94	1240	858	905	234
11	78	546	312	312	156
12	30	694	330	550	100
13	468	3160	1423	1493	577
14	283	4444	2837	1419	473
15	83	2339	826	880	606
16	78	437	156	109	78
17	47	1624	757	519	39
18	750	120	2110	800	140
19	2293	1048	7193	891	563
20	100	140	840	380	40
21	1810	156	4016	3014	516
22	150	310	1402	125	516
23	322	161	1979	622	516
24	140	242	414	140	516
25	120	288	592	372	516
26	330	200	380	400	516
27	140	100	4083	1025	516
28	85	255	1575	708	516
29	554	850	1215	3290	516
30	806	875	989	726	516
31	755	2976	5860	331	516
32	16	47	31	93	516
33	160	160	400	303	516
34	234	468	468	312	516
35	213	221	1864	296	516
36	16	156	311	117	516
37	300	520	1300	390	516
38	312	1248	1014	780	516
39	48	82	512	52	516
40	31	733	312	734	516

Figure 6.6 Imputation to subset Y based on absolute average EI of subset X

The shaded area in Figure 6.7 presents next the imputation results for the subset Y of 20 projects based on the relative % average of EI for each project: the imputed EI hours vary from a min of 15 hours to a max of 738 hours.

Input	EP	ES	EB	ET	EI
1	3120	1540	5260	2299	2946
2	208	624	2075	761	528
3	2	38	91	15	24
4	80	80	80	40	20
5	1190	9793	17167	4489	1384
6	578	3412	3768	1540	1320
7	78	1435	640	1794	374
8	196	565	204	163	466
9	527	2278	1016	2335	260
10	94	1240	858	905	234
11	78	546	312	312	156
12	30	694	330	550	100
13	468	3160	1423	1493	577
14	283	4444	2837	1419	473
15	83	2339	826	880	606
16	78	437	156	109	78
17	47	1624	757	519	39
18	750	120	2110	800	140
19	2293	1048	7193	891	563
20	100	140	840	380	40
21	1810	156	4016	3014	738
22	150	310	1402	125	163
23	322	161	1979	622	253
24	140	242	414	140	77
25	120	288	592	372	113
26	330	200	380	400	107
27	140	100	4083	1025	439
28	85	255	1575	708	215
29	554	850	1215	3290	485
30	806	875	989	726	278
31	755	2976	5860	331	814
32	16	47	31	93	15
33	160	160	400	303	84
34	234	468	468	312	122
35	213	221	1864	296	213
36	16	156	311	117	49
37	300	520	1300	390	206
38	312	1248	1014	780	275
39	48	82	512	52	57
40	31	733	312	734	148

Figure 6.7 Imputation to subset Y based on relative %average

Tables 6.10 and 6.11 present the imputation results (with and without the outliers). For example, in comparision to the EI average of 6.7% from the actual values in Table 6-9, the imputed values in Table 6.11 vary:

- from 7.6% using imputation based on relative % average of 8.2%;
- to 19.0% when based on relative min and max seeds of 1% and 41%.

Table 6.10 Average effort distribution by project phase after imputations and by subsets-including outliers (N=41 projects)

Imputation Based on	Profile	Project Phases – % Effort					No. of projects
		Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
absolute average	Subset X : PSBTI	8.6	28.8	36.2	18.8	7.6	**21**
	Subset Y : PSBT	9.5	14.5	41.6	20.0	(14.4)	**20**
relative %average	Subset X : PSBTI	8.6	28.8	36.2	18.8	7.6	**21**
	Subset Y : PSBT	11.0	17.0	48.6	23.4	(7.6)	**20**
MI absolute seeds	Subset X : PSBTI	8.6	28.8	36.2	18.8	7.6	**21**
	Subset Y : PSBT	11.1	17.0	48.6	23.4	(13.8)	**20**

Table 6.11 Average effort distribution by project phase after imputations and by subsets - excluding outliers (N=40 projects)

Imputation Based on	Profile	Project Phases – % Effort					No. of projects
		Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
absolute average	Subset X : PSBTI	8.2	28.3	38.1	17.2	8.2	**20**
	Subset Y : PSBT	8.6	23.4	39.3	18.2	(10.6)	**20**
relative %average	Subset X : PSBTI	8.2	28.3	38.1	17.2	8.2	**20**
	Subset Y : PSBT	11.1	17.0	48.6	23.4	(8.2)	**20**
MI absolute seeds	Subset X : PSBTI	8.2	28.3	38.1	17.2	8.2	**20**
	Subset Y : PSBT	11.1	17.0	48.6	23.4	(13.5)	**20**
Relative seeds %1 to %24	Subset X : PSBTI	8.2	28.3	38.1	17.2	8.2	**20**
	Subset Y : PSBT	11.1	17.0	48.6	23.4	(13.0)	**20**
Relative seeds %1 to %41	Subset X : PSBTI	8.2	28.3	38.1	17.2	8.2	**20**
	Subset Y : PSBT	11.1	17.0	48.6	23.4	(19.0)	**20**

When subset X and the imputed subset Y are recombined together, their results are presented in Tables 6.12 and 6.13 with their average effort distribution by project phase (for N=41 and N= 40 projects with and without outliers).

For example, the first line of in Table 6.13 presents the percentage effort distribution by phase for the full data set of 40 projects (excluding 1 outlier): here the 7.7% EI will be used to compare the performance of the different types of imputations done.

It can then be observed from table 6.13 that:

- At 8.2% for EI, the imputation based on the relative %average is the closest to the reference EI value of 7.7%;
- At 11.7% for EI, the imputation based on the relative seeds (1%, 41%) has the largest difference of 4% relative to the reference EI value of 7.7%.

Table 6.12 Average effort distribution after imputations – full dataset - including outliers (N=41 projects)

Imputation Based on	Profile	Project Phases – % Effort					No. of projects
		Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
Complete Data	Subsets X & Y: PSBTI	9.1	24.7	39.1	19.7	7.3	**41**
absolute average	Subsets X & Y: PSBTI	8.9	24.0	38.0	19.2	(9.9)	**41**
relative %average	Subsets X & Y: PSBTI	9.3	25.3	39.9	20.2	(7.6)	**41**
MI absolute seeds	Subsets X & Y: PSBTI	9.3	25.3	39.9	20.2	(9.5)	**41**

Table 6.13 Average effort distribution excluding outliers (N=40 projects)

Imputation Based on	Profile	Project Phases – % Effort					No. of projects
		Effort Plan	Effort Specify	Effort Build	Effort Test	Effort Implement	
Complete Data	Subset X & Y: PSBTI	8.9	24.1	40.5	18.8	7.7	**40**
absolute average	Subset X & Y: PSBTI	8.6	23.4	39.3	18.2	(10.6)	**40**
relative %average	Subset X & Y: PSBTI	9.1	24.7	41.5	19.2	(8.2)	**40**
MI absolute seeds	Subset X & Y: PSBTI	9.1	24.7	41.5	19.2	(9.9)	**40**
Relative seeds %1 to %24	Subset X & Y: PSBTI	9.1	24.7	41.5	19.2	(9.7)	**40**
Relative seeds %1 to %41	Subset X & Y: PSBTI	9.1	24.7	41.5	19.2	(11.7)	**40**

6.4.3 Estimation model from Imputation based on an average

Next the regression models are built using Subset X as the training data set and subset Y after its imputation based on absolute average and relative % average. The regression estimation models for the dataset without the outlier are (bottom part of Table 6.14):

Effort Implement = 250hrs + 0.55xEP + 0.10xES -0.03xEB – 0.09xET (Absolute EI Imputed with an Adjusted R^2 of 0.49).

Effort Implement = 21hrs + 0.64xEP + 0.13xES -0.04xEB – 0.01xET (Relative EI imputed with an Adjusted R^2 of 0.69).

Both have a P-value <0.1 for both imputations: it can be concluded that the results are statistically significant at *t*-test and P-values for Effort Implement estimates.

Table 6.14 Regression models for Effort Implement after imputations based on averages

Dataset	**N=41 projects, with outliers**							
	Intercept	Effort Plan	Effort Specify	Effort Build	Effort Test	Adjusted R^2	R^2	P-value
Imputed based on absolute EI average	277	0.47	0.06	0.02	-0.10	0.41	0.47	<0.0001
Imputed based on relative % EI average	61	0.55	0.09	0.01	-0.07	0.58	0.62	<0.0001
Dataset	**N=40 projects, without outliers**							
	Intercept	Effort Plan	Effort Specify	Effort Build	Effort Test	Adjusted R^2	R^2	P-value
Imputed based on absolute EI average	250	0.55	0.10	-0.03	-0.09	0.49	0.54	<0.0001
Imputed based on relative % EI average	21	0.64	0.13	-0.04	0.01	0.69	0.72	<0.0001

Next the analysis of estimate variance is done after applying on subset Y the estimation models built from subset X – see Table 6-15: the quality of the estimation model for the imputation based on:

- absolute EI average of subset X leads to an MMRE = 155% and Pred(25) = 23%;
- relative %EI average leads to an MMRE = 74% and Pred(25) = 30%.

In summary, the imputation based on the relative EI average leads to better estimation models of EI than based on EI absolute average (i.e. Adjusted R^2 = 0.69, MMRE = 74% and Pred(25) = 30%.

Table 6.15 Estimate variance of Effort Implement – Imputations based on averages

No.	Dataset	N=40 projects, without outliers	
		%MMRE	Pred(25)
1	Imputed based on absolute average	155%	23%
2	Imputed based on relative %average	74%	30%

6.4.4 Comparisons between MI and imputation on averages (Absolute and relative seeds excluding outliers)

This section compares the R^2 and estimation error variance of the set of 40 projects (secluding 1 outlier) for:

A. the complete dataset without missing values.
B. the results from:
   B1. Imputation based on the absolute average EI of subset X, and
   B2. Imputation based on the relative %average.
C. the results from:
   C1. MI with absolute seeds Min and Max, and
   C2. MI with relative seeds Min and Max.

Table 6.16 presents the comparison results, with the top half of the table presenting directly the Adjusted R^2, %MMRE and Pred(25) for each model, while the bottom half of the table presenting pair-wise comparison of models results:

- Line 7: The major difference between the imputation based on absolute average EI of subset X (line 2) and the complete dataset (line 1) corresponds to:
  - a decrease of -22% in the adjusted R^2,
  - an increase of MMRE = 67%, and
  - a decrease of with Pred(25) = -7%.
- Line 8: For imputed based on relative %average (line 3) compared to the complete dataset (line 1) there is:
  - a minor decrease of -2% in the adjusted R^2;
  - an increase of the MMRE = 14% compared to the complete dataset, and
  - a Pred(25) equal to the complete dataset.

- Line 9: For the MI results based on absolute seeds (line 4) compared to the complete dataset (line 1) there is:
  - a minor decrease of -6% in the adjusted R^2;
  - an increase of the MMRE = 28% compared to the complete dataset, and
  - a minor decrease of Pred(25) = -3%.
- Line 10: For the relative seeds %1 to %24 (line 5) compared to the complete dataset (line 1), there is:
  - a decrease of -11% in the adjusted R^2;
  - an increase of 59% in the MMRE;
  - a decrease of -6% in the Pred(25).
- Line 11: For the relative seeds %1 to %41 (line 6) compared to the complete dataset (line 1), there is:
  - a decrease of -18% in the adjusted R^2;
  - an increase of 84% in the MMRE;
  - a decrease of -7% in the Pred(25).

Table 6.16 Comparison of models predictive performances

No.	Dataset	N=40 projects, without outliers (Effort Implement) Model N=40 projects		
		Adjusted R^2	%MMRE	Pred(25)
1	Complete dataset	0.71	88%	30%
2	Imputed based on absolute average	0.49	155%	23%
3	Imputed based on relative %average	0.69	74%	30%
4	Imputed MI based on absolute seeds	0.65	116%	27%
5	Imputed (relative seeds %1 to %24)	0.60	147%	24%
6	Imputed (relative seeds %1 to %41)	0.53	172%	23%
7	**2 vs. 1**	-22%	+67%	-7%
8	**3 vs. 1**	-2%	+14%	0%
9	**4 vs. 1**	-6%	+28%	-3%
10	**5 vs. 1**	-11%	+59%	-6%
11	**6 vs. 1**	-18%	+84%	-7%

This is an encouraging result since this dataset was small, N= 40 projects, and contained 50% missing observations: these are challenging circumstances for imputation techniques. Our empirical results suggest that the MI imputation method has practical utility for software engineers involved in effort estimation data analysis. In addition, it is worth observing that

imputation is one of the activities in the more general field of data editing which includes a whole range of techniques for identifying, removing and updating suspect data.

6.5 Summary

This chapter has presented the use of two imputation techniques for dealing with the problem of missing data in software engineering dataset:

- multiple Imputation (MI), with values selected randomly from absolute or relative min and max, and
- imputation based on absolute or relative averages.

These studies were carried out both across functional profiles (PSBIT, PSBT and SBIT) (sections 6.2 and 6.3), and within the same PSBTI profile (section 6.4).

The question investigated was: do imputation methods allow to improve the usefulness of software engineering dataset that contain a large number of missing values?

In particular, in section 6.4, we have attempted to answer this by considering the effort estimation modeling for a complete dataset N=40 projects with the PSBTI profile (with and without outliers). We then created missing values artificially by deleting randomly data from the EI data field in half of the data set: subset A consisting of 20 complete projects (without outliers) and subset B of 20 projects with the EI data field values deleted.

The impact on parameter estimate analysis (with and without outliers) of the use of of MI on incomplete datasets was investigated.

First, by removing the outlier:

- the adjusted R^2 increased for the complete dataset from 0.58 to 0.71.
- the results of the combined imputation improved substantially after removing the outliers: the adjusted R^2 increased from 0.45 to 0.65,

- the adjusted R^2 for the imputed based on relative % EI average from 0.58 to 0.67,
- the adjusted R^2 for the imputed based on absolute EI average from 0.41 to 0.49.

Therefore, removing the outlier strengthened the linearity of the data and decreased the errors present in the regression. Furthermore, the results are statistically significant for the estimates of Effort Implement, as illustrated by the t-test and P-values with and without outliers.

The performance of software prediction models between estimated effort and actual effort was evaluated using two evaluation criteria: MMRE (Mean Magnitude Relative Error), and Pred(25). The MMRE evaluation criterion was used to select the best prediction model. The estimates were obtained from multiple regression analysis estimation models.

This chapter found these by analyzing the prediction performance of the various models. The quality of the estimation model with the complete dataset: a MMRE = 88% and Pred(25) = 30%. Furthermore, with 50% of the data missing (i.e. 20 missing values in a sample of 40 projects); a much larger error should be expected, but with MI, the quality of the regression results, with an MMRE of only 27% higher and a Pred(25) that is only 3% lower for the combined MI model (without outliers).

By analyzing the prediction performance of the results it can be observed that:

- The MI technique has a very small impact in the ISBSG data repository in current software engineering projects for the effort estimation model, which means that the quality of the estimation model with imputed data is –see Table 6.3:
    a) very close to the actual values (observed values),
    b) the performance of the imputation models in terms of MMRE was higher than 25%, and
    c) Pred(25), is lower than 75% on the five imputation datasets.
- In the evaluation the performance of missing data technique in software prediction models between estimated effort and actual effort, the large percentage with 50% of

missing values was expected results with mush large error, but with MI, the quality of the regression results indicated that:

- There is a large difference for the combined 5 imputations results of MI with a decrease MMRE = -5%, and an increase of Pred(25) = +10% compared to the Subset X – See Table 6.5.
- The major difference between the Subset Y and the Subset X with increases MMRE = 21% and decreases Pred(25) = -15%, respectively.
- While the difference for the Subset Y decreases with MMRE = -26%, and increases the Pred(25) = 25% compared with the combined 5 imputations results of MI.

- Therefore, the results difference of complete dataset and compared with the combined 5 imputations datasets results and the estimation model for the Subset Y N=20 projects –See Table 6.5: the combined imputations MI model performed better than the (Subset Y with missing values N=20).

Furthermore, the sensitivity analysis of the relative imputation of the effort estimation without outliers for the Effort implement was investigated. We found that the results of the combined imputation of MI based on absolute seeds the adjusted R^2 is 0.65, close to the adjusted R^2 for the complete dataset which is 0.71, while the adjusted R^2 for the relative seeds 1% to 24% was 0.60, as well as the adjusted R^2 for the relative seeds 1% to 41% was only 0.58. Furthermore, the results are statistically significant for the estimates of Effort Implement, as illustrated by the t-test and P-values without outliers – see Table 6.6.

This chapter investigated next the contribution of relative imputation for N=40 projects with imputed values for the Effort Implement phase. We found that the MI technique with the relative values for seeds has a very small impact for the effort estimation model, which means the quality of the estimation model with imputed data based on the absolute seed values is close to the actual values:

- the difference for the combined imputed with relative seed %Min = 1, and %Max= 24 increases with MMRE = 59%, and decreases the Pred(25) = -6%, compared with the complete dataset –see Table 6.7.
- While the difference for the combined imputed with relative seed %Min =1, and %Max = 41 increases in MMRE= 84%, and decreases the Pred(25)= -7% compared with the complete dataset –see Table 6.7;
- There is minor difference for the combined results of MI increases MMRE = 28%, and decreases Pred(25) = -3% compared to the complete dataset –see Table 6.7.

Overall, the results of this empirical study on MI gave generally the best results: the MI technique provides the better results from incompleteness of data, and the results are encouraging in the sense that the MI method performs better or at least close to models based on complete data when applied to missing data.

CONCLUSION

The International Software Benchmarking Standards Group (ISBSG) data repository comprises project data from several different companies across the world. However, this repository contains a large number of missing data, which often considerably reduces the number of data points available for building productivity models and for building estimation models. There are a few techniques available for handling missing values, but it is essential to apply them appropriately, otherwise biased or misleading inferences may be made.

The research goal of this thesis was to develop an improved usage of the ISBSG data repository by both practitioners and researchers by leveraging the larger quantity of data available for statistical analysis in software engineering, while discarding the data which may affect the meaningfulness of the statistical tests.

To achieve this research goal, the following three specific research objectives were formulated:

1. To tackle the new problems in larger datasets in software engineering including outliers and missing values using the Multiple Imputation technique.
2. To investigate the use of the multiple imputation (MI) technique with the ISBSG repository for dealing with outliers and missing values.
3. To demonstrate the impact and evaluate the performance of the MI technique in current software engineering repositories dealing with software project efforts for estimation purposes, between estimated effort and actual effort.

In this research project, these objectives were achieved by using the ISBSG dataset repository release 9 (ISBSG, 2005), which contains data on 3024 software projects: the reason that prevented this research from using Release 12 is that there are a large number of projects that had information on effort by project phases in Release 9 but did not have anymore such information in Release 12.

The next paragraphs summarize how each of these research objectives has been met, as illustrated with the outcomes of the empirical studies in chapter 5 and 6.

Objective 1: To investigate the use of the multiple imputation (MI) technique with the ISBSG repository for dealing with outliers and missing values.

To achieve the first research objective the technique used to deal missing value in the ISBSG dataset is the Multiple Imputation (MI) technique: Chapter 5 investigated the impact of MI in the estimation of the missing values of the effort variable by project phase using the ISBSG repository, and applied regression models, both with and without outliers, and examined their specific influence on the results.

Five imputation rounds were used to produce parameter estimates which reflect the uncertainty associated with estimating missing data. Chapter 5 also determined the averages of the effort distribution by phase for three profiles (PSBTI, PSBT, and SBTI), and for each of the five imputation rounds. The PSBT profile presents a missing phase (Effort Implementation), and the SBTI profile presents a missing phase (EffortPlan), and, as a result, the average of the effort distributions of the other phases (Effort Specification, Effort Build, and Effort Test), as well as the combined average of the effort distribution of all the projects, varied accordingly in each imputation.

Moreover, the regression analysis was trained with the five imputed datasets from 65 projects (with outliers) and 62 projects (without outliers). It was observed that the adjusted R^2 is lower for the dataset without outliers, indicating that the outliers unduly influenced the estimation models, leading to over statistical confidence in the results.

Furthermore, chapter 5 presented the results of multiple imputation variance information and parameter estimates for the Effort Implement and Effort Plan variables over the five imputed datasets.

- The results of this investigation revealed that the variance results of the standard error of the parameter estimates decreased from 105 hours to 73 hours for Effort Implement and from 106 hours to 60 hours for Effort Plan for a multiple regression analysis with and without outliers respectively.
- The multiple regression analysis results were statistically significant for the Effort Plan and Effort Implement estimates, as illustrated by the t-test and P-values with and without outliers.

Chapter 5 also presented the results of five effort estimation models that were combined with the five imputed dataset estimates, and obtained the averages of the parameter estimates. The results of this investigation have shown the contributions of the three variables (ES, EB, and ET):

- The P-value of the EB and ET variables statistically presented a much higher significant impact on the effort estimate than the ES variable.
- The estimated effect of EP on the ES parameter was -0.12 respectively, with a t-statistic equal to -2.05 and P-values of 0.04 respectively. Note that the values of the t-statistic were less than 2.
- The estimated effect of EI on the ES and ET parameters was 0.03, and 0.10 respectively, with a t-statistic equal to 0.56 and 1.82 and P-values of 0.57, and 0.11 respectively. Note that the values of the t-statistic were also less than 2.
- The intercept coefficient is not statistically significant.

This means that the multiple regression analysis results did not find evidence that ES and ET have any impact on the EI and EP parameters, but it does have an impact on the EB parameter.

Furthermore, removing the outliers strengthens the linearity of the data and decreases the range of errors present in the regression. It can be observed that the adjusted R^2 is lower for the dataset without outliers: this means that the results analysis with the missing data

observations indicate that the outliers unduly influenced the estimation models, leading to over statistical confidence in the results.

Objective 2: To demonstrate the impact and evaluate the performance of the MI technique in current software engineering repositories dealing with software project efforts for estimation purposes, between estimated effort and actual effort.

To achieve the second research objective Chapter 6 looked at two imputation techniques:

1. Imputed data from imputations based on average values of the Effort Implement.
2. Imputed data based on multiple imputations by random selection from min-max seeds.

For these two imputations techniques, two distinct approaches were investigated:

a) Based only from the data within the field with missing values – this was referred to as imputation from absolute values.
b) Based on imputation taking into account information from other data fields: here, the information from the data fields of Effort Plan, Effort Specify, Effort Build and Effort Test will be used to calculate the distribution of Effort Implement relative to the effort in the other project phases. This was referred to as imputation from relative values

Hence, for approach **a)** above, the null and alternative hypotheses of our research were defined as follows:

- H0: When an estimation model is built from imputed data based on the absolute average values, we obtain a predictive accuracy that is statistically significantly better than imputed data from MI imputations from absolute min-max seeds.
- H1: When an estimation model is built from imputed data based on the absolute average values, we do not obtain a predictive accuracy that is statistically significantly better than imputed data from MI imputations from absolute min-max seeds.

Hence, for approach **b)** above, the null and alternative hypotheses of our research were defined as follows:

- H2: When an estimation model is built from imputed data based on the relative average values, we obtain predictive accuracy that is statistically significantly better than imputed data from MI imputations from relative min-max seeds.
- H3: When an estimation model is built from imputed data based on the relative average values, we do not obtain predictive accuracy that is statistically significantly better than imputed data from MI imputations from relative min-max seeds.

For investigating these research hypotheses, a new research strategy was designed in chapter 6 to investigate the performance of these two imputation techniques on the basis of the 40 projects of the ISBSG dataset for the PSBTI profile without outliers (or 41 projects with the outlier), and to divide it into 2 subsets: Subset X of 20 of the 40 projects, which 20 have complete data fields and Subset Y of the other 20 projects from which is the information in the Effort Implement data field is deleted.

The key elements of this strategy is to compare between the training dataset Subset A N=20 complete dataset, and the combined 5 imputations datasets results of the MI of subset Y N=20, and with the training estimation model that was applied to estimate EI in Subset Y N=20 projects.

The regression analyses were built for estimation model with subset X of the complete data of PSBTI to be used as training dataset for building the estimation model, and then this estimation model was applied to the subset Y with missing values of Effort Implement.

Chapter 6 investigated the impact on parameter estimate analysis (with and without outliers) of the use of imputation techniques on incomplete datasets. First, the results of the imputations improved substantially after removing an outlier: the adjusted R^2 increased from 0.45 to 0.65, after removing the outlier, and:

- the adjusted R^2 also increased for the complete dataset from 0.58 to 0.71, and

- the adjusted R^2 for the imputed based on relative % EI average from 0.58 to 0.67,
- the adjusted R^2 for the imputed based on absolute EI average from 0.41 to 0.49.

Therefore, removing the outlier strengthened the linearity of the data and decreased the errors present in the regression. Furthermore, the results are statistically significant for the estimates of Effort Implement, as illustrated by the t-test and P-values with and without outliers.

Next, the performance of software prediction models between estimated effort and actual effort was evaluated using two evaluation criteria: MMRE (Mean Magnitude Relative Error), and Pred(25). The MMRE evaluation criterion was used to select the best prediction model. The estimates were obtained from multiple regression analysis estimation models.

For approach **a)** above, the null and alternative hypotheses for H0 and H1, Chapter 6 found the prediction performance by analyzing the quality of the various estimation models (dataset without an outlier):

- with the complete dataset, the MMRE = 88% and Pred(25) = 30%, .
- the quality of the regression results for MI imputations from absolute min-max seeds, with an MMRE of 116% and a Pred(25) = 27% .
- the regression results for the absolute average values, with an MMRE of 155% and a Pred(25) =23%.

The comparison of the multiple imputation (MI) results with respect to a simpler imputation technique based on an absolute average (from Table 6-16):

- Line 4: There is minor increase of 28% for the MMRE for the combined results of MI imputations from absolute min-max seeds, a decrease of (-6%) in the adjusted R^2, an increases MMRE and a small decrease of 3% in the Pred(25) compared to the complete dataset.
- Line 5: The difference from the estimation model from the absolute average value imputation, a large decrease of -22% in the adjusted R^2, a large increase in the MMRE of 67%, and a decrease in the Pred(25) of -7% compared with the complete data set;

- Line 6: therefore the better performance of the MI imputations from absolute min-max seeds in comparison to the imputation from absolute average values, is an increase of 16% in the adjusted R^2, a decrease in the MMRE of -39%, and an increase in the Pred(25) of 4%.

In summary, with approach **a)** above, the null H0 hypothesis is not confirmed, while the alternate H1 hypothesis is confirmed, that is:

- H1: When an estimation model is built from imputed data based on the absolute average values, we do not obtain a predictive accuracy that is statistically significantly better than imputed data from MI imputations from absolute min-max seeds.

For approach **b)** above, the null and alternative hypotheses for H2 and H3, Chapter 6 found the prediction performance by analyzing the quality of the various estimation models:

The comparison of the multiple imputation (MI) results with respect to a simpler imputation technique based on the relative % average (Table 6-16):

- Line 5: The difference for the Imputed based on relative %average and the complete data is only a minor decrease of -2% in the adjusted R^2, an increase of MMRE of 14%, and there is no difference for the Pred(25).
- Line 6: The difference for the estimation models from imputation from relative seeds %1 to % 24 correspond to a decrease of -11% in the adjusted R^2, an increase of MMRE of 59%, and in a decrease in the Pred(25) of -7%, as compared to the complete data.
- Line 7: The difference for the relative seeds %1 to % 41 correspond to a decrease of18% in the adjusted R^2, an increase in the MMRE of 84% and a decrease in the Pred(25) of-7%, as compared to the complete data.
- Line 8: The difference for the relative %average correspond to an increase of 9% in the adjusted R^2, a large decrease of the MMRE of 73%, and an increase in the

Pred(25) = 6%, as compared to the estimation model from imputation from the relative seeds %1 to % 24.

- Line 9: The difference for the relative %average correspond to a large increase of 16% in the adjusted R^2, a large decrease of MMRE of 98%, and an increase in the Pred(25) of 7%, as compared to the estimation model from imputation from the relative seeds %1 to % 41.

In summary, with approach **b)** above, the null H2 hypothesis is confirmed, while the alternate H3 hypothesis is not confirmed, that is:

- H2: When an estimation model is built from imputed data based on the relative average values, we obtain predictive accuracy that is statistically significantly better than imputed data from MI imputations from relative min-max seeds.

This is an encouraging result since our dataset was small, N= 40 projects, and contained 50% missing observations. These are challenging circumstances for imputation techniques. Our empirical results suggest that the MI imputation method has practical utility for software engineers involved in effort estimation data analysis. In addition, it is worth observing that imputation is just one activity in the more general field of data editing which includes a whole range of techniques for identifying, removing and updating suspect data.

Recommendations for further research work

There is a number of additional research works related to the research goal of our work that can be pursued. In order to derive more general results, researchers should develop techniques, or improve existing techniques, that can be used for investigating the fields with a large number of missing values, such as max team size, lines of code, and resource level, in the ISBSG data repository.

Our research methodology and results provide practical and substantiated guidelines for researchers and practitioners constructing effort estimation models when their datasets have outliers and missing values.

MI is not the only modern missing data tool to become available to researchers. Some producers of statistical software are beginning to incorporate incomplete data features directly into certain types of modeling routines. These procedures are similar to MI in that they implicitly average over a predictive distribution for the missing values, but the averaging is performed using analytic or numerical methods rather than simulation.

This research encourages future replications of the simulation on the ISBSG datasets reported in this empirical study in order to confirm our conclusions. Certainly, such replications will have important practical significance for practitioners and researchers building effort estimation models.

ANNEX I

LIST OF APPENDICES ON CD-ROM

The following is the list of appendices referenced within this thesis and that can be found on the attached CD-ROM:

Appendix #	File name	Description
Folder name: Data Preparation and Outliers Test		
I	Phase effort consistent Test for Outliers with 179 projects.xle	Outliers test for 179 projects of consistent effort by phases.
II	Data Consistent with Summary Effort 179 projects with data quality A and B.xle	Selected data quality A & B for IFPUG method for consistent data.
III	Phase effort consistent without Outliers with 106 Projects.xle	Consistent data selected after outlier test effort by phases 106 projects.
IV	Average effort distribution for the PSBTI excluding outliers and unusual distributions with 107 projects.xle	Average effort distribution for the PSBTI excluding outliers and unusual distributions with 107 projects with missing value.
V	Average effort distribution for 34 projects for with very high Specification Effort.xle	Average effort distribution for 34 projects for with very high Specification Effort.
Folder name: Data Analysis with 106 and 103 projects		
VI	ISBSG Data Use for SAS with 106 projects.xle	ISBSG data with outliers and missing value used for SAS with 106 projects.
VII	Average effort distribution for the PSBTI for ISBSG using SAS	Average effort distribution for the PSBTI for ISBSG using SAS

	Output with 106 projects.xle	Output with 106 projects with outliers and after replacing missing values.
VIII	ISBSG Analaysis Results with outliers 106 projects.txt	SAS Analaysis results for ISBSG data after replacing missing values with outliers 106 projects.
IX	ISBSG Data Use for SAS with 103 projects.xle	ISBSG data without outliers and missing value used for SAS with 103 projects.
X	Average effort distribution for the PSBTI for ISBSG using SAS Output with 103 projects.xle	Average effort distribution for the PSBTI for ISBSG using SAS Output with 103 projects without outliers and after replacing missing values.
XI	ISBSG Analaysis Results without outliers 103 projects.txt	SAS Analaysis results for ISBSG data after replacing missing values without outliers 103 projects.
Folder name: Data Analysis with 103 and 101 projects		
XII	ISBSG Data Use for SAS with 103 projects.xle	ISBSG data with outliers and missing value used for SAS with 103 projects.
XIII	Average effort distribution for the PSBTI for ISBSG using SAS Output with 103 projects.xle	Average effort distribution for the PSBTI for ISBSG using SAS Output with 103 projects with outliers and after replacing missing values.
XIV	ISBSG Analaysis Results with outliers 103 projects.txt	SAS Analaysis results for ISBSG data after replacing missing values with outliers 103 projects.
XV	ISBSG Data Use for SAS with	ISBSG data without outliers and

	101 projects.xle	missing value used for SAS with 101 projects.
XVI	Average effort distribution for the PSBTI for ISBSG using SAS Output with 101 projects.xle	Average effort distribution for the PSBTI for ISBSG using SAS Output with 101 projects without outliers and after replacing missing values.
XVII	ISBSG Analaysis Results without outliers 101 projects.txt	SAS Analaysis results for ISBSG data after replacing missing values without outliers 101 projects.
Folder name: ISBSG Completed Data with 40 and 41 projects		
XVIII	Analysis ISBSG Completed Data.xle	Analysis completed data of ISBSG with 41 completed projects with outliers.
XIX	ISBSG Data Use for SAS with 41 projects.xle	ISBSG data with outliers and missing value used for SAS with 41 projects after deleted 21 projects from which Effort Implement data.
XX	Average effort distribution for the PSBTI Completed Data with outliers 41 projects.xle	Average effort distribution for the PSBTI for ISBSG using SAS Output with 41 projects with outliers and after replacing missing values.
XXI	ISBSG_Data Regression Analysis With outliers with 41 projects.xle	Regression analysis for ISBSG completed data with outliers with 41 projects and test for measuring the predictive accuracy of an effort estimation model.
XXII	Analysis.txt	SAS Analaysis results for ISBSG data after replacing missing values

		with outliers 41 projects.
XXIII	Analysis ISBSG Completed Data.xle	Analysis completed data of ISBSG with 40 completed projects without outliers.
XXIV	ISBSG Data Use for SAS with 40 projects.xle	ISBSG data without outliers and missing value used for SAS with 40 projects after deleted 20 projects from which Effort Implement data.
XXV	Average effort distribution for the PSBTI Completed Data without outliers 40 projects.xle	Average effort distribution for the PSBTI for ISBSG using SAS Output with 40 projects without outliers and after replacing missing values.
XXVI	ISBSG_Data Regression Analysis Without outliers with 40 projects.xle	Regression analysis for ISBSG completed data without outliers with 40 projects and test for measuring the predictive accuracy of an effort estimation model.
XXVII	Analysis.txt	SAS Analaysis results for ISBSG data after replacing missing values without outliers 40 projects.
XXVIII	Appendix research papers refer to ISBSG.doc	Papers for researcher refer using ISBSG data repository.
Folder name: Data Analysis of Déry and Abran		
XXIX	ISBSG data Analysis before Apply MI.xle	Analysis data of Déry and Abran and applying and test for measuring the predictive accuracy of an effort estimation model.

LIST OF BIBLIOGRAPHICAL REFERENCES

Abran, A., I. Ndiaye and P. Bourque. 2007. Evaluation of a black-box estimation tool: a case study. Software Process: Improvement and Practice, vol. 12, n° 2, p. 199-218.

Abran, Alain. 2009. Software Estimation. Coll. Draft version. École de technologie supérieure, Montréal (Canada).

Adalier, O., A. Ugur, S. Korukoglu and K. Ertas. 2007. A new regression based software cost estimation model using power values. 8th International Conference on Intelligent Data Engineering and Automated Learning IDEAL 2007, 16-19 December 2007, Birmingham, England, LNCS 4881, p.326-334

Azen, S., and M. Van Guilder. 1981. Conclusions Regarding Algorithms for Handling Incomplete Data. Statistical Computing Section, American Statistical Association. p. 53-56.

Barnett, V., and T. Lewis. 1995. Outliers in statistical data. 3th edition. John Wiley, Chicester, England.

Buglione, L., and A. Abran. 2008. Performance calculation and estimation with QEST/LIME using ISBSG R10 data. 5th Software Measurement European Forum (SMEF 2008), Milan (Italy), 28-30 May 2008, ISBN 9-788870-909999, p. 175-192.

Chan, Victor K. Y., and W. Eric Wong. 2007. Outlier elimination in construction of software metric models. ACM Symposium on Applied Computing (SAC), Seoul, Korea, March 11-15, 2007, p. 1484-1488.

Cheikhi, Laila, Alain Abran and Luigi Buglione. 2006. ISBSG Software Project Repository & ISO 9126: An Opportunity for Quality Benchmarking. European Journal for the Informatics Professional, Vol.7, No.1, February 2006, pp. 46-52. <www.upgrade-cepis.org> Upgrade: ISSN 1684-5285 Novática: ISSN 0211-2124

Colledge, M., J. Johnson, R. Pare and I. Sande. 1978. Large Scale Imputation of Survey Data. Section on Survey Research Methods, Washington, DC: American Statistical Association, p. 431-436.

Conte, S. D., H. E. Dunsmore and V. Y. Shen. 1986. Software Engineering Metrics and Models. Menlo Park, California: Benjamin/Cummings Publishing Company, Inc., Menlo Park, CA.

Davies, Laurie, and Ursula Gather. 1993. The Identification of Multiple Outliers. Journal of the American Statistical Association, vol. 88, n° 423, September 1993, p. 782-792.

de Barcelos Tronto, I. F., J. D. S. da Silva and N. Sant'Anna. 2007. Comparison of artificial neural network and regression models in software effort estimation. International Joint Conference on Neural Networks, 12-17 Aug. 2007. Piscataway, NJ, USA, p. 771-776. IEEE.

Deng, Kefu, and Stephen G. MacDonell. 2008. Maximising data retention from the ISBSG repository. 12th International Conference on Evaluation and Assessment in Software Engineering (EASE), Bari, Italy. June 2008.

Déry, David, and Alain Abran. 2005. Investigation of the Effort Data Consistency in the ISBSG Repository. 15th International Workshop on Software Measurement - IWSM'2005, Montreal, Canada, p. 123-136.Publisher: Shaker Verlag Aachen, Germany.

Everitt, B.S., and G. Dunn. 2001. Applied Multivariate Data Analysis. London, UK: Arnold 2nd edition. Cited on p.203, 209.

Fenton, Norman, and Shari Pfleeger. 1998. Software metrics (2nd edition): A rigorous and practical approach. PWS Publishing Company, ISBN =0-534-95600-9, Boston, MA, USA.

Foss, T. Stensrud, E., and B. Myrtveit Kitchenham, I. 2003. A simulation study of the model evaluation criterion MMRE. IEEE Transactions on Software Engineering, vol. 29, n°. 11, November 2003, p. 985-995.

Fuller, Wayne, and Jae-Kwang Kim. 2005. Hot deck imputation for the response model. Survey Methodology, vol. 31, n° 2, p.139-149.

Gencel, Cigdem, and Luigi Buglione. 2007. Do Different Functionality Types Affect the Relationship between Software Functional Size and Effort?. International Conference on Software Process and Product Measurement (IWSM-MENSURA 2007), Palma de Mallorca, Spain, p.235-246.

Graham, J. W., and J. L. Schafer. 1999. On the performance of multiple imputation for multivariate data with small sample size. In Hoyle R (Ed), Statistical strategies for small sample research, Thousand Oaks, CA: Sage, p.1-29.

Graham, John W., Patricio E. Cumsille and Elvira Elek-Fisk. 2003. Methods for Handling Missing Data. Handbook of Psychology. John Wiley & Sons, Inc. vol.2, p. 87-114.

Gray, Andrew R., and Stephen G. MacDonell. 1997. A comparison of techniques for developing predictive models of software metrics. Information and Software Technology, vol. 39, n° 6, p. 425-437.

Harter, H. L. 1983. Least Squares. Encyclopedia of Statistical Sciences, Kotz, S. and Johnson, N.L., eds., John Wiley & Sons, New York, p. 593-598.

Hill, P. R. 2003. The practical collection, acquisition, and application of software metrics. Cutter IT Journal, vol. 16, n° 6, p. 38-42.

ISBSG, International Software Benchmarking Standards Group, Analysis Reports < http://www.isbsg.com/collections/analysis-reports >.

ISBSG, International Software Benchmarking Standards Group, Data Collection Questionaires,<http://www.isbsg.org/ISBSGnew.nsf/WebPages/286528C58F55415BCA257474001C7B48>.

Jeff, Heaton T. 2005. Introduction to Neural Networks with Java. Heaton Research, Inc. ISBN: 097732060X, p. 380.

Jeffery, R., M. Ruhe and I. Wieczorek. 2000. A comparative study of two software development cost modeling techniques using multi-organizational and company-specific data. Information and Software Technology, vol. 42, n° 14, p. 1009-1016.

Jianfeng, Wen, Li Shixian and Tang Linyan. 2009. Improve Analogy-Based Software Effort Estimation using Principal Components Analysis and Correlation Weighting. 16th Asia-Pacific Software Engineering Conference, Los Alamitos, CA, USA, p. 179-186.

Jiang, Z., P. Naudé and B. Jiang. 2007. The effects of software size on development effort and software quality. International Journal of computer and Information Science and Engineering 1(4), p. 230-234.

John, Graham W., Hofer M. Stewart Scott., Donaldson I., and MAcKINNON P. Joseph David, Schafer L. 1997. Analysis with missing data in prevention research. The science of prevention: Methodological advances from alcohol and substance abuse research, (pp. 325-366). Washington, DC, US: American Psychological Association, xxxii, 458 pp. doi: 10.1037/10222-010.

Jorgensen, M., and M. Shepperd. 2007. A Systematic Review of Software Development Cost Estimation Studies. IEEE Transactions on Software Engineering, vol. 33, n° 1, p. 33-53.

Joseph, L. Schafer, and W. Graham John. 2002. Missing data: Our view of the state of the art. Psychological Methods, n° 2, p. 147-177.

Kitchenham, Barbara, Emilia Mendes and Guilherme H. Travassos. 2006. A Systematic Review of Cross vs. Within Company Cost Estimation Studies. IEEE Transactions on Software Engineering, vol. 33, N°. 5, p. 316-329.

Kuhnt, Sonja, and J. rg Pawlitschko. 2003. Outlier identification rules for generalized linear models. 27th Annual Conference of the Gesellschaft für Klassifikation e.V., Brandenburg University of Technology, Cottbus, March 12–14, 2003, Innovations in Classification, Data Science, and Information Systems, Berlin: Springer, p. 165-172.

Little, Roderick. 1988. Missing-Data Adjustments in Large Surveys. Journal of Business & Economic Statistics, vol. 6, n° 3, p. 287-296.

Little, Roderick. 1992. Regression With Missing X's: A Review. Journal of the American Statistical Association, vol. 87, n° 420, p. 1227-1237.

Little, Roderick. 1995. Modeling the Drop-Out Mechanism in Repeated-Measures Studies. Journal of the American Statistical Association, vol. 90, n° 431, p. 1112-1121.

Little, Roderick J A, and Donald B Rubin. 1986. Statistical analysis with missing data. John Wiley & Sons, New York.

Little, Roderick, and Donald Rubin. 2002. Statistical Analysis with Missing Data, 2nd Edition. New York: John Wiley & Sons, ISBN: 978-0-471-18386-0, p. 349-364.

Lokan, C., T. Wright, P. Hill and M. Stringer. 2001. Organizational benchmarking using the ISBSG Data Repository. IEEE Software, September/October 2001, vol. 18, n° 5, p. 26-32.

Lokan, Chris, and Emilia Mendes. 2006. Cross-company and single-company effort models using the ISBSG database: a further replicated study. 2006 ACM/IEEE international symposium on Empirical software engineering. (Rio de Janeiro, Brazil), p. 75-84. 1159747: ACM.

Mendes, E., and C. Lokan. 2008. Replicating studies on cross- vs single-company effort models using the ISBSG database. Empirical Software Engineering: An International Journal, vol. 13, n° 1, p. 3-37.

Mendes, E., C. Lokan, R. Harrison and C. Triggs. 2005. A replicated comparison of cross-company and within-company effort estimation models using the ISBSG database.. 11th International Symposium on Software Metrics. Como, Italy (September. 2005), p. 10-36.

Mittas, Nikolaos, and Lefteris Angelis. 2008. Combining regression and estimation by analogy in a semi-parametric model for software cost estimation. Second ACM-IEEE international symposium on Empirical software engineering and measurement. (Kaiserslautern, Germany), p. 70-79. 1414017: ACM.

Myrtveit, I., E. Stensrud and U. H. Olsson. 2001. Analyzing data sets with missing data: an empirical evaluation of imputation methods and likelihood-based methods. IEEE Transactions on Software Engineering, November 2001, vol. 27, n° 11, p. 999-1013.

Paré, D., and A. Abran. 2005. Obvious Outliers in the ISBSG Repository of Software Projects: Exploratory Research. In Metrics News, Otto Von Gueriske Universitat, Magdeburg (Germany), vol. 10, n° 1, August, 2005, p. 28-36

Pendharkar, P. C., G. H. Subramanian and J. A. Rodger. 2005. A probabilistic model for predicting software development effort. IEEE Transactions on Software Engineering, vol. 31, n° 7, p. 615-624.

Pendharkar, Parag C., James A. Rodger and Girish H. Subramanian. 2008. An empirical study of the Cobb–Douglas production function properties of software development effort. Information and Software Technology, vol. 50, n° 12, p. 1181-1188.

Reilly, and Marie. 1993. Data analysis using hot deck multiple imputation. The statistician : journal of the Institute of Statisticians. - Oxford : Blackwell, ISSN 0039-0526, ZDB-ID 280816x. vol. 42, p. 307-313.

Rubin, D. B. 1987. Multiple imputation for nonresponse in surveys. New York: John Wiley & Sons, Inc.

Rubin, D. B. 1996. Multiple imputation after 18+ years. Journal of the American Statistical Association, vol. 91, p. 473-489.

SAS Institute Inc. statistical analysis software. < http://www.sas.com >.

Schafer, J. L. 1997. Analysis of incomplete multivariate data. Chapman & Hall, London, p. 430, ISBN 0-412-04061-1.

Shepperd, M., and C. Schofield. 1997. Estimating software project effort using analogies., IEEE Transactions on Software Engineering, vol. 23, n° 11, p. 736-743.

Stephen, G. Macdonell, and J. Shepperd Martin. 2003. Combining techniques to optimize effort predictions in software project management. Information and Software Technology, vol. 66, n° 2, p. 91-98.

Stigler, Stephen M. 1978. Mathematical Statistics in the Early States. The Annals of Statistics, vol. 6, p. 239-265.

Stigler, Stephen M. 1988. The History of Statistics. The Measurement of Uncertainty before 1900. The Belknap Press of Harvard University, Cambridge, Mass., & London 1986; XVI, 410 S. Biometrical Journal, vol. 30, n° 5, p. 631-632.

Sun-Jen, Huang, and Chiu Nan-Hsing. 2006. Optimization of analogy weights by genetic algorithm for software effort estimation. Information and Software Technology, vol. 48, n° 11, p. 1034-45.

Switzer, Fred S., Philip L. Roth and Deborah M. Switzer. 1998. Systematic Data Loss in HRM Settings: A Monte Carlo Analysis. Journal of Management, vol. 24, n° 6, p. 763-779.

Wayman, Jeffrey C. 2002. The utility of educational resilience for studying degree attainment in school dropouts. The Journal of Educational Research, vol. 95, n° 3, p. 167-178.

Xia, V., D. Ho and L. F. Capretz. 2006. Calibrating Function Points Using Neuro-Fuzzy Technique. 21st International Forum on COCOMO and Software Cost Modeling, Herndon, Virginia.