# The Value and Validity of Software Effort Estimation Models built from a Multiple Organization Data Set

Kefu Deng

A thesis submitted to
Auckland University of Technology
in fulfilment of the requirements for the degree of
Master of Philosophy (MPhil)

2008

School of Computing and Mathematical Sciences
Primary Supervisor: Professor Stephen MacDonell

# Table of Contents

## List of Figures

# List of Tables

## Attestation of Authorship

Yours sincerely,

(Kefu Deng)

## Acknowledgements

To Professor Stephen MacDonell, for his knowledge and patience during the time of this project.
To the ISBSG as the willing source of the data used in this work.

## Abstract

The objective of this research is to empirically assess the value and validity of a multi-organization data set in the building of prediction models for several 'local' software organizations; that is, smaller organizations that might have a few project records but that are interested in improving their ability to accurately predict software project effort. Evidence to date in the research literature is mixed, due not to problems with the underlying research ideas but with limitations in the analytical processes employed:

- the majority of previous studies have used only a single organization as the 'local' sample, introducing the potential for bias
- the degree to which the conclusions of these studies might apply more generally is unable to be determined because of a lack of transparency in the data analysis processes used.

It is the aim of this research to provide a more robust and visible test of the utility of the largest multi-organization data set currently available – that from the ISBSG – in terms of enabling smaller-scale organizations to build relevant and accurate models for project-level effort prediction.

Stepwise regression is employed to enable the construction of 'local', 'global' and 'refined global' models of effort that are then validated against actual project data from eight organizations. The results indicate that local data, that is, data collected for a single organization, is almost always more effective as a basis for the construction of a predictive model than data sourced from a global repository. That said, the accuracy of the models produced from the global data set, while worse than that achieved with local data, may be sufficiently accurate in the absence of reliable local data – an issue that could be investigated in future research.

The study concludes with recommendations for both software engineering practice – in setting out a more dynamic scenario for the management of software development – and research – in terms of implications for the collection and analysis of software engineering data.

# 1 Introduction

## 1.1 Brief Background and Research Objective

In many respects the discipline of software engineering has had to mature remarkably quickly, given the now pervasive and dominant role of software in virtually all aspects of modern society. This has meant that, at times, the discipline has relied more on anecdotal or partial evidence than on a solid empirical base in order to inform changes in practice. One of the contributing factors to a less than ideal empirical base has been a shortage of data. Put briefly, progress in the software engineering research and practice communities has been constrained by the relatively small data sets available for empirical analysis (Kitchenham *et al.* 2007). This limitation arises for at least five reasons (Kitchenham *et al.* 2007; MacDonell and Shepperd, 2007). First, software projects tend to be long running processes – over months or years – hence completion is a relatively infrequent event. Second, some of these projects do not actually make it to completion, meaning that the data are either lost or not included in associated data sets. Third, project data are seen as commercially sensitive and therefore confidential, leading to reluctance to share information across organizational boundaries. Fourth, relatively few organizations devote sufficient effort to the systematic collection and organization of project data. Finally, with fast changing technology, data can quickly become obsolete in terms of relevance to future work.

Having access to limited numbers of project records means that building reliable and generally applicable models for prediction and classification is very difficult. In order to overcome this limitation there has been a trend over the last ten years to aggregate data sets across multiple organizations, and even across countries. While this obviously removes the problem of having only small data sets, it introduces other challenging questions. For instance, are the resultant data sets sufficiently homogeneous to enable valid analysis? If not, what selection criteria need to be used in order to obtain valid outcomes? Does model-building performance improve or deteriorate with larger data sets? Is it better to use a small amount of local data or a much larger amount of potentially highly heterogeneous data?

The overall objective of this research is to compare the utility of an aggregated (global, cross-company, multi-organization) data set to several constituent (local) data sets in terms of enabling the construction and validation of prediction models for software development effort. Such an objective is important in informing software practice – armed with an understanding of the comparative effectiveness of each approach, organizations can make an informed choice as to whether they should invest in the development and maintenance of a local metrics programme or in the submission to and analysis of one or more large global data sets. The objective is also important in relation to software engineering research – several prior studies have addressed this objective but have done so in an inconsistent manner in terms of data sets used and analysis methods applied, with the result that the outcomes and conclusions are inconsistent. This work is considered in detail in Chapter 2, which deliberately leverages two recent systematic reviews to provide a basis for the work undertaken and reported here. In contrast to these prior studies, this work provides a more comprehensive indication of the reliability of analyses undertaken with such global data repositories and the extent to which local, but limited, data sets could be replaced with data sets collected from and managed by external sources.

The research undertaken here therefore addresses the following research questions:

a. With respect to the construction and validation of accurate prediction models for software development effort, how do small homogeneous (local) data sets compare with larger multi-organization (global) data sets?
b. With respect to the construction and validation of accurate prediction models for software development effort, is it possible to find and utilise relevant non-local data?

## 1.2 Intended Method and Scope

In order to address the above questions a comparison of local and global model accuracy is undertaken and reported here. The multi-organization data set utilised in this study is that sourced from the International Software Benchmarking Standards Group (ISBSG). This not-for-profit group collects metric data on software projects from all over the world. Currently more than twenty countries are represented, with the most recent release (as at 2008) comprising data on more than 4000 projects. While commercial users are required to buy this data repository, it is generously provided at no cost to researchers on application. Further details of the data set can be found at the ISBSG's own web site (ISBSG, 2006a).

For the purposes of comparison an additional data set is utilised in this study. This is a separately supplied subset of the standard ISBSG repository that includes identified project records for ten organizations. In the standard repository, project records are provided with no organization identification. The availability of this additional data set enables the building and comparison of local and global variants.

A variety of statistical analysis methods are used to address the questions stated above and as described in Section 3.2.1. In the first instance, data distributions are examined in order to select appropriate parametric, non-parametric or robust analysis methods. This is also informed by various data visualisation methods, including scatter plots and boxplots. Correlation analysis is employed in order to identify relationships between variables. Least-squares regression is then utilised to build predictive models, with predictive model accuracy assessed using a range of error measures (Kitchenham *et al.* 2001).

Principal constraints on the scope of the research arise from the use of the ISBSG repository. As stated above, while large, it does comprise records from a limited number of countries, from volunteer organizations (introducing a self-selection bias), that represent only completed projects. As a result there are limitations on the degree to which any outcomes of the analyses reported here can be applied. It is, however, the largest global data set available, and could therefore be considered as approaching an approximation of the population of software projects.

## 1.3 Thesis Structure

The next chapter provides a targeted review of prior research relevant to the specific research objectives addressed here. Chapter 3 describes in detail the data used in this research and the research methodology and techniques employed in its analysis. The execution of the data analysis and its results are then presented in Chapter 4. Chapter 5 provides a discussion of the outcomes of the work and concludes the thesis.

# 2 Literature Review – Software Project Effort Estimation from Multi-organization Data Sets

This chapter summarises the relevant prior research in relation to the research objectives of the current work. In keeping with the nature of an MPhil thesis it is not intended as a comprehensive review of the body of work in the domain of software project effort estimation. Rather, it is intended to provide sufficient background to ensure that the motivation for the work conducted here is established. The chapter therefore comprises the following sections:

- A brief history of software project estimation with particular emphasis on significant changes in research and practice.
- A review of related research via two recent systematic reviews in terms of methodologies, underlying data sets and analysis outcomes.
- A discussion of the advantages and disadvantages of the methodologies employed, research practices within those methodologies, utilised data sets and research outcomes.

## *2.1 Background*

For software projects, the resources, schedule, quality and features form a tradeoff scenario. Any change to one of the four implies that a corresponding change is needed in the other three. The key to developing a solution that meets the customer's requirements is to determine and maintain an appropriate balance between resources, deployment date, quality and features. For example, given a fixed feature set, a fixed quality level and a fixed delivery date, the resources (i.e. the people and the necessary hardware/software environment) that are allocated to the project have to be adjusted in order to meet those fixed expectations. If the required number or scope of features is increased but quality expectations are maintained, as a natural consequence, project managers have to either:

- keep the resources fixed but extend the expected delivery date of the project, OR
- keep the expected delivery date but increase the resources allocated to the project.

In order to determine the ‘amount’ of resources required (which fundamentally comes down to personnel resource, as this is the dominant cost factor), the expected delivery date, the number/scope of features and the required level of quality have to be clearly understood by the software development organization.

While this activity is itself problematic, further challenges remain even when the requirements are clearly understood. With respect to the central issue in this research, how can an understanding of the feature requirements be ‘translated’ into the requirements for schedule and resource, which can then be translated into estimates of effort, time and costs, that can then be passed to the customers/stakeholders, so that they can make correct business decisions? Overestimating could lead to potential opportunities being lost to other software development groups or organizations, while underestimating could result in financial losses. Therefore, using the minimum amount of time (and therefore information) prior to the approval of a project while getting a correct estimate of the effort/cost of that project remains a critical challenge for the industry.

Estimation techniques designed to address this challenge have evolved significantly during the last three decades, and have been the subject of an ongoing sequence of review studies. Heemstra (1990) highlighted in particular the ongoing use of expert estimation as a dominant approach in software project planning and management, perhaps reflecting the relative immaturity of practices up to that point. A review of functional assessment techniques by MacDonell (1994) reflected the emphasis at that time on using representations of software functionality as the basis of effort estimation. Boehm *et al.* (2000) noted the growing emergence of machine learning methods in effort estimation that had occurred in response to questions over the effectiveness of parameter-based or statistical modelling. In recent years, Jørgensen and colleagues (Moløkken-Ostvold and Jørgensen, 2003; Jørgensen, 2004; Jørgensen and Shepperd, 2007) have reported several surveys and systematic reviews of studies in software effort estimation, indicating a continuing interest in the enduring challenges associated with the effective management of costs and schedules in software projects.

Turning attention to the relationship between models and data, there has been an associated evolution of practices. In the late 1970s and 1980s, the sophisticated parameter-based models of the day were built to deal with a range of software development environments. A typical example at that time was the COCOMO model (Boehm, 1981), built on the basis of around 70 completed projects. COCOMO comprised three different sub-models for different development modes. Fourteen cost drivers were also used to take into account the variety of tools, environments, non-functional requirements and so forth that might be relevant in any given project. By employing such models, software development organizations did not need to collect their own local data but could instead ‘plug’ the data for each of their future projects into the COCOMO structure, thus exploiting the generality of the model.

After a period of use, some software organizations and researchers, such as Kemerer (1987), realised that the effectiveness of such a generic model decreased in situations where the domain of the target software development environment was outside the domain from which the models were originally established. From a statistical point of view, this situation is quite natural and obvious. This is because the validity of a model derived from a set of sample data is largely dependent on the context from which the data is gathered. More information on this issue can be found in Venables and Ripley (2002).

To deal with this limitation, software organizations and researchers began to establish their own models using local, organization-specific data and employing a variety of techniques such as Ordinary Least Squares Regression and decision trees. By doing so, some organizations were able to attain a higher level of accuracy in software development effort estimation. However, the feasibility and viability of such an approach are questionable in certain situations, among them:

- when the software development organization is newly established
- when new software environments or tools are being utilised in a project
- when the local data collected from previous software development practices has not been designed effectively or is not good (complete, accurate) enough to generate a useful model (Kitchenham *et al.* 2006a)

- when it is not cost-effective for the software development organization to collect and analyse the data.

As a result, software development organizations and researchers began to turn their attention to globally established software engineering data sets, such as the International Software Benchmarking Standards Group repository (ISBSG, 2006a), the Tukutuku data set (Tukutuku, 2008), and the 'Finnish' data set (STTF 2008), also known as the Experience, Laturi or STTF data set in some contexts. Use of these data sets is not without constraint. The ISBSG repository must either be purchased or an organization must submit data on 50 projects of their own. The STTF data set is proprietary and is available to those organizations contributing to it. The Tukutuku data set is not freely available. Furthermore, the data sets are not without their own limitations – Tukutuku relates to web projects only, and the STTF data set represents projects undertaken in Finland. More generally, the data sets represent only projects that have been completed – the subset of projects that are abandoned, and that are potentially rich in information regarding scope and estimation, are not included.

Setting such constraints to one side, the fact is that software development organizations have a choice when it comes to forming a basis for effort estimation. They can expend effort and resources to establish and maintain their own company-specific data set and thus build their own software effort estimation models. Alternatively (or in addition), they can use a globally available software engineering data set at a relatively lower cost but at the expense of local relevance. This choice has motivated researchers to consider the pros and cons of each option – the empirical analyses that have addressed this issue are now considered.

## 2.2 Related Research – Local and Global Modelling

The research questions central to the work conducted and reported here have been addressed in previous empirical software engineering research – for example Maxwell *et al.* (1999), Briand *et al.* (2000), Jeffery *et al.* (2000; 2001) and Kitchenham and Mendes (2004). However, the outcomes to date have been inconsistent. There has therefore been a compelling need for more systematic investigation across multiple data sets.

In 2007, two extensive reviews of the local vs. global modelling question were reported in parallel by Kitchenham *et al.* (2007) and MacDonell and Shepperd (2007). Both papers represent meta-research – they are systematic reviews of previous studies. Both papers employed the review approach as described in Kitchenham (2004) with predefined protocols. In fact, the underlying studies were designed intentionally to address the same research question in the same way, as the basis of an assessment of the reliability of systematic reviews in empirical software engineering. Comparable analysis techniques were also used in both papers to assess the robustness of previous research relating to their conclusions regarding the usefulness of local vs. global effort estimation models. Examples of the characteristics considered include data quality, the degree of dependency between the local and global data sets, data set size and sampling, and several others.

Ten primary studies were identified by Kitchenham *et al.* (2007) according to their inclusion criteria. MacDonell and Shepperd (2007) also selected ten studies – of those, nine were in common with those selected in Kitchenham *et al.* (2007). These papers are listed in the following table as compiled from Kitchenham *et al.* (2007) and MacDonell and Shepperd (2007):

| **Title** | **Year Published** | **Authors** | **In Kitchenham *et al.* (2007)?** | **In MacDonell and Shepperd (2007)?** |
|---|---|---|---|---|
| An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques | 1999 | Lionel C. Briand, Khaled El Emam Dagmar Surmann, Isabella Wieczorek, Katrina D. Maxwell | Yes | Yes |
| Performance Evaluation of General and Company Specific Models in Software Development Effort Estimation | 1999 | Katrina Maxwell, Luk Van Wassenhove, Soumitra Dutta | Yes | Yes |
| A replicated Assessment and Comparison of Common Software Cost Modeling Techniques | 2000 | Lionel C. Briand, Tristen Langley, Isabella Wieczorek | Yes | Yes |

| | | | | |
|---|---|---|---|---|
| A comparative study of two software development cost modeling techniques using multi-organizational and company-specic data | 2000 | R. Jeffery, M. Ruhe, I. Wieczorek | Yes | Yes |
| Using Public Domain Metrics to Estimate Software Development Effort | 2001 | Ross Jeffery, Melanie Ruhe, Isabella Wieczorek | Yes | Yes |
| How Valuable is company-specific Data Compared to multi-company Data for Software Cost Estimation? | 2002 | Isabella Wieczorek, Melanie Ruhe | Yes | Yes |
| Early Web Size Measures and Effort Prediction for Web Costimation | 2003 | Emilia Mendes, Nile Mosley, Steve Counsel | No | Yes |
| Using Genetic Programming to Improve Software Effort Estimation Based on General Data Sets | 2003 | Martin Lefley, Martin J. Shepperd | Yes | Yes |
| A Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications | 2004 | Barbara A. Kitchenham, Emilia Mendes | Yes | Yes |
| Further Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications | 2004 | Barbara A. Kitchenham, Emilia Mendes | Yes | Yes |
| A Replicated Comparison of Cross-Company and Within-Company Effort Estimation Models Using the ISBSG Database | 2005 | Mendes, E., C.Lokan, R. Harrison, C. Triggs | Yes | No |

**Table 1: Studies reviewed in prior systematic reviews**

As these two meta-research studies are both recent and largely in agreement in terms of coverage of the literature they are used here to provide a snapshot of the current state of knowledge with respect to the research questions addressed in this study. The foundation of this research is established upon the conclusion of these two previous systematic meta-analyses, with further insights and observations then added.

### 2.2.1 Kitchenham *et al.* (2007)

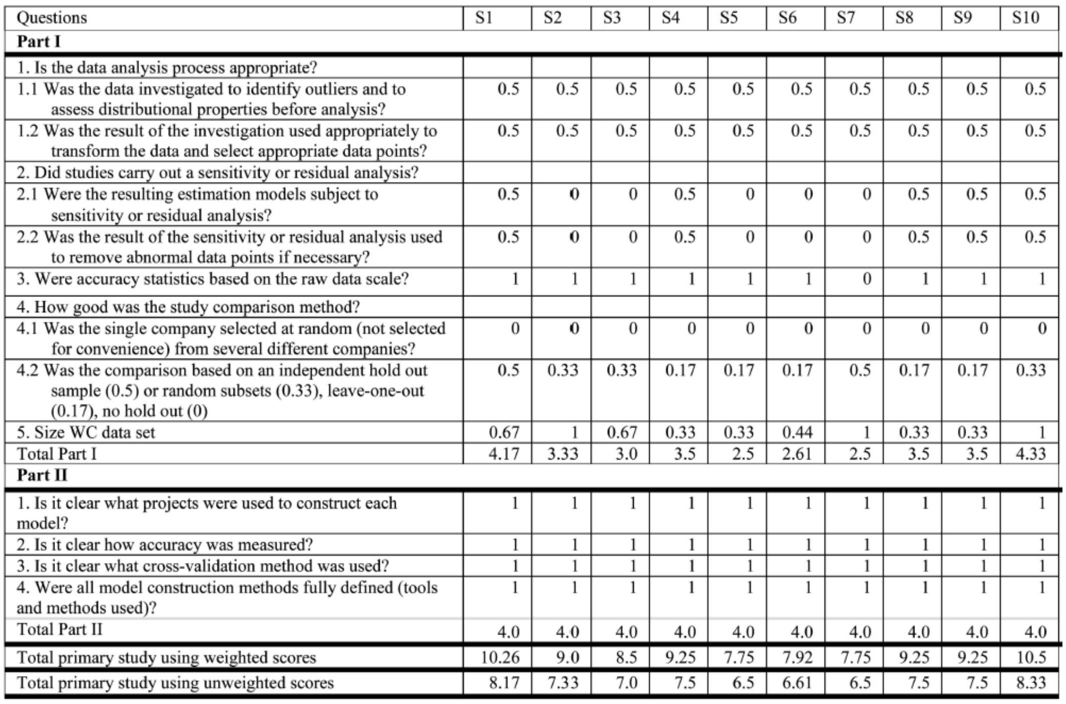
Kitchenham *et al.* (2007), an extension of Kitchenham *et al.* (2006b), provides comprehensive analysis of prior research that had addressed the issue of the superiority of local vs. global estimation models. Although the primary objective of the research was "knowing whether or not it is reasonable to use cross-company estimation models" (page 317), it was also intended to provide some advice to researchers on the potential value of cross-company models. A protocol was specified according to conventions adopted in the medical domain by Pai *et al.* (2004). Four aspects of research design were considered – Population, Intervention, Comparison Intervention, and Outcome (PICO) – to enable the specification of a well-formulated research question.

In their paper, three main questions were asked (page 320):

- "What evidence is there that cross-company estimation models are not significantly different from within-company estimation models for predicting effort for software/web projects?"
- "What characteristics of the study data sets and the data analysis methods used in the study affect the outcome of within-company and cross-company effort estimation accuracy studies?" i.e. the factors that could affect the result of the within-company and cross-company model comparison.
- "Which experimental procedure is most appropriate for studies comparing within-company and cross-company estimation models?"

The search for relevant primary studies was composed of two phases. In the initial phase, a search string was developed and used to identify papers based on six electronic databases and seven individual journals and conference proceedings. Using the selected papers from the initial phase as candidate papers, the secondary phase considered the references from each to further identify other possible candidate papers. This process was repeated until no further candidate papers emerged. The studies were then characterised according to a range of quality criteria, leading to the provision of a quality score for each. The quality results for each selected paper are shown in Table 2.

Although there is a lack of transparency and limited granularity in terms of how each quality aspect for the selected papers is rated – for example, all the scores for criterion 1.1 is 0.5, while all the scores for some other criteria are either all 0 or all 1 – the intent of the rating process and the insights with respect to areas of possible concern (e.g. components 4.1 and 4.2) are of relevance and value to this study.



| Questions | S1 | S2 | S3 | S4 | S5 | S6 | S7 | S8 | S9 | S10 |
|---|---|---|---|---|---|---|---|---|---|---|
| **Part I** | | | | | | | | | | |
| 1. Is the data analysis process appropriate? | | | | | | | | | | |
| 1.1 Was the data investigated to identify outliers and to assess distributional properties before analysis? | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| 1.2 Was the result of the investigation used appropriately to transform the data and select appropriate data points? | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 | 0.5 |
| 2. Did studies carry out a sensitivity or residual analysis? | | | | | | | | | | |
| 2.1 Were the resulting estimation models subject to sensitivity or residual analysis? | 0.5 | 0 | 0 | 0.5 | 0 | 0 | 0 | 0.5 | 0.5 | 0.5 |
| 2.2 Was the result of the sensitivity or residual analysis used to remove abnormal data points if necessary? | 0.5 | 0 | 0 | 0.5 | 0 | 0 | 0 | 0.5 | 0.5 | 0.5 |
| 3. Were accuracy statistics based on the raw data scale? | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 4. How good was the study comparison method? | | | | | | | | | | |
| 4.1 Was the single company selected at random (not selected for convenience) from several different companies? | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4.2 Was the comparison based on an independent hold out sample (0.5) or random subsets (0.33), leave-one-out (0.17), no hold out (0) | 0.5 | 0.33 | 0.33 | 0.17 | 0.17 | 0.17 | 0.5 | 0.17 | 0.17 | 0.33 |
| 5. Size WC data set | 0.67 | 1 | 0.67 | 0.33 | 0.33 | 0.44 | 1 | 0.33 | 0.33 | 1 |
| Total Part I | 4.17 | 3.33 | 3.0 | 3.5 | 2.5 | 2.61 | 2.5 | 3.5 | 3.5 | 4.33 |
| **Part II** | | | | | | | | | | |
| 1. Is it clear what projects were used to construct each model? | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2. Is it clear how accuracy was measured? | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3. Is it clear what cross-validation method was used? | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 4. Were all model construction methods fully defined (tools and methods used)? | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Total Part II | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 | 4.0 |
| Total primary study using weighted scores | 10.26 | 9.0 | 8.5 | 9.25 | 7.75 | 7.92 | 7.75 | 9.25 | 9.25 | 10.5 |
| Total primary study using unweighted scores | 8.17 | 7.33 | 7.0 | 7.5 | 6.5 | 6.61 | 6.5 | 7.5 | 7.5 | 8.33 |

**Table 2: Quality score table (sourced from Kitchenham *et al.* 2007)**

Kitchenham *et al.* (2007) arrived at the following answers to their research questions:

- In terms of the within-company and cross-company model comparison – no studies were found to favour cross-company models on the basis of model performance i.e. no study claimed that cross-company models were significantly better than within-company models. However in four cases preference was assigned to cross-company models on the basis that, with no difference between the two approaches, cross-company models should be favoured due to lower cost and effort being required for their use.

- In terms of factors that could affect the result of the within-company and cross-company comparison:
  - Data collection following rigorous quality assurance procedures "does not appear to ensure that a cross-company model will perform as well as a within-company model" (page 326), although there is some question over this outcome in that Maxwell *et al.* (1999) raised doubts over the actual effectiveness of the quality assurance processes.
  - The quality score assigned to each study according to the protocol "does not appear to affect study results". However, "quality scores for the most recent studies are higher than the quality scores for the earlier studies".
  - They found that "studies where within-company predictions were significantly better than cross-company predictions employed smaller within-company data sets". They would expect "large within-company data sets to lead to more reliable results". Therefore, "we should put more trust in the results that suggest cross-company models are not significantly different from within-company models".
  - The dependency between the within-company data and the cross company data seems to have an effect on the result. "In two out of four cases, where the within-company model presented significantly better predictions than the cross-company models, the single company projects had been collected separately from the cross-company projects. In contrast, in all cases where the within-company and cross-company models were not significantly different, the within-company data was a subset of the cross-company data set (i.e., was not collected separately from the cross-company projects)".
  - The size of the within-company data sets seems to have an impact on the final result. "In three out of four cases, where the within-company model presented significantly better predictions than the cross-company models, the single-company projects were volunteered by small companies (Megatec and the two single companies from Tukutuku). In all cases where the within-company model presented significantly better predictions than the cross-company models, the single company projects (in terms of effort) were relatively small."

    - “Use regression analysis as the default model construction method”. And “Use a stepwise approach on the cross-company data set based on the variables collected in the within-company data set”.

Future work suggested by Kitchenham *et al.* (2007) was as follows:

- Consider the degree of similarity between within-company and cross-company data sets when comparing them.
- All researchers in this area “come to some consensus about the most appropriate experimental procedure for this type of study and use the same procedures for future studies”.
- Such studies should ensure that the within-company data was independent of the cross-company data.
- Try out other more suitable validation approaches for effort prediction for small within-company data sets.

These suggestions are considered along with other observations in relation to this study after consideration of the MacDonell and Shepperd (2007) work.

### 2.2.2 MacDonell and Shepperd (2007)

MacDonell and Shepperd (2007) took a deliberately similar approach to that reported in Kitchenham *et al.* (2007). The research protocol also followed the PICO approach as suggested by Pai *et al.* (2004). The main review question was defined as (page 2):

- “What evidence is there that cross-company estimation models are at least as good as within-company estimation models for predicting effort for software projects?”

A set of search criteria was developed but in a different manner to that used by Kitchenham *et al.* (2007). In MacDonell and Shepperd (2007), a set of constraint criteria was firstly defined to limit the boundary of the searched papers. A set of search keywords was then derived by the researchers by examining five published papers known to be within the area of interest. A complicated searching scenario across 13 research study repositories was then conducted and further reviewed.

A total of 185 potentially relevant papers including duplicates was retrieved. Papers were then excluded by the predefined constraint criteria. Abstracts of all the retrieved papers were manually read by the researchers “to determine whether they should be considered as primary studies” (page 3). Ten primary studies were subsequently identified, including 8 conference papers and 2 journal papers.

The review of MacDonell and Shepperd (2007) was more focused on the research process used in each study, the data quality, data diversity, model construction and experimental design, than the Kitchenham *et al.* (2007) work. The following conclusions and future indications can be drawn from this paper:

- “While there was found to be a tendency for the more recent (and perhaps higher quality?) primary studies to support local models it would be inappropriate to state at this stage that the evidence is converging on that outcome” (page 10).
- Higher quality studies using agreed standards and discipline-wide reporting protocols are suggested for future research. For example, more research is needed to consider the impact of different validation procedures such as *n-fold* or *leave-one-out* on the outcomes of the empirical comparisons. Also, the target population of each study, not stated in any of those studies considered in the review, should be addressed in future work.

### 2.2.3 Other Insights Gained

In addition to the conclusions and recommendations raised in the two meta-analyses just described, a range of other observations can be made based on those studies and on the eleven constituent papers that had themselves addressed the within-company and cross-company comparison in terms of software effort estimation.

First, almost every research effort drew its conclusions based on the comparison of data from one single organization and one global software engineering data repository. From a statistical point of view this is open to question in terms of being sufficiently robust enough to demonstrate causation.

Second, all but one of the primary studies discarded data – in some instances significant proportions of the original data – if there were missing values in terms of observations or variables. The exception was Mendes *et al.* (2005) who used missing data imputation in an effort to retain observations. Similarly, in spite of the global software engineering repositories often comprising very large numbers of variables, the actual number of variables retained and used in the final models was generally small. For example, the ISBSG data set contains 88 variables but just four were used in the final global model generated in the work of Mendes *et al.* (2005). It is of course entirely reasonable to discard data in certain circumstances – models that are large can be intractable to build and unstable to use. Furthermore, if accuracy is not significantly decreased then a smaller model is normally to be preferred over a larger alternative. However, the *process* of discarding data, as one important step of the data handling process, should be driven not just in response to missing values but also in relation to software engineering domain knowledge.Moreover, missing data techniques (MDT) can be used to impute missing values so that the associated observations or variables can be retained. In terms of the studies considered in the above review, such an approach is evident in Mendes *et al.* (2005). In their research they employ the *k*-NN (k nearest neighbours) imputation method as a missing data handling technique. Further details of the *k*-NN technique being employed in relation to software engineering data can be found in Jonsson and Wohlin (2004). Given the often large proportions of data discarded in previous studies, the use of this approach is considered here to be worth serious consideration if it means that greater volumes of data can be retained for analysis and thus better (more accurate and less biased) models can be constructed. Thus this study applies not only a MDT, but also, perhaps more importantly, software engineering domain knowledge to the data handling process.

Third, while almost every research effort used OLS Regression as one of its model-building methods, the actual OLS Regression process differed across the various studies. For instance, some studies used all of the local records available to build a model and then tested that same model against the same data in a leave-one-out analysis. Such an approach tends to produce an optimistic indication of model accuracy. Most significantly, in all but one of the eleven primary studies considered by the two reviews (the exception again being Mendes *et al.* (2005)), there was insufficient – normally no – explanation of the assumption checking and diagnostics process. From the point of view of statistical rigour, this immediately brings into question the application of OLS Regression because the validity of regression models is reliant on their adherence to several underlying assumptions: that the residuals are normally distributed, that they display planar and constant scatter, and that they are independent. Apart from the very limited coverage of the assumption checking process, there is also a lack of explanation in regard to other aspects of the OLS Regression process applied in the studies, such as the specific variable selection techniques used. Researchers typically need to choose between step-wise regression and all-possible-regressions, and even within step-wise regression, there is a choice between forward checking and backward elimination, or a combination of both. Explanations of this nature were simply missing from prior studies.

Finally, and as alluded to previously, the notion of the relevant population for each study was not addressed, making it difficult to determine with confidence the scope of applicability of outcomes. Similar omissions were also evident in terms of sometimes limited or no description of the sampling of data from the records available, or of the diversity of the data in terms of countries and industry sectors represented.

## 2.3 Intended Research Principles

Taking into account the aims of this research, and bearing in mind the issues of methodology just described, the following principles are adopted to guide the research conducted here:

- **Transparent research process**: The research process is to be executed with full transparency of the rationale for each decision taken. This should include:
    a. robust and visible data handling processes, taking into account software engineering domain knowledge and experience drawn from other evidence-based disciplines such as medical research.
    b. efficacious model building and model comparison, taking into account software engineering domain knowledge and best practice from previous research (for instance, Mendes *et al.* (2005)) as well as research from other domains.

- **Use of relevant global data**: The research should not only compare models drawn from local and global data sets, but also build and compare semi-global models *drawn from a relevant subset of the global data set* with the local models.

In order to adhere to the above mentioned principles, the following research guidelines are specified (which should be applied to this and similar studies):

1. The research data sets and model-building methodologies should be selected based on well-founded and robust previous research and experience drawn from other domains, ensuring full visibility of the rationale for each decision.
2. With the data on-hand the data handling process should be informed by software engineering domain knowledge in combination with principles as required by the selected model-building methodology.
3. With processed data on-hand, the target population of the research should be specified so that the applicability of any discrete conclusions drawn is appropriately bounded.
4. Each model-building process must be visible to reviewers and researchers, and must be repeatable with the same set of data.
5. When a model is established, assumption-checking must be performed and diagnostics of the model reviewed in order to ensure the reliability and robustness of research outcomes based on the model, or its comparison with other models.
6. Model performance should be evaluated using appropriate techniques and metrics, with full explanation of the rationale for their selection.
7. Research conclusions and implications for practice should be drawn not only from the specific results of the model comparisons but also from the whole research process, so that the perhaps one-off characteristics of a given data set do not distort outcomes said to apply broadly to software engineering research and practice.

With the above principles and guidelines now in place, the following chapter describes the data sets utilized in this research and the detailed research method employed.

# 3 Data Source and Research Method

This chapter describes the form and nature of the data source employed in this research and the composition of the specific data sets. There then follows a description, with justification, of the adopted research method.

## *3.1 The Data Source*

This research addresses the comparative worth of single-organization data sets and global data repositories in terms of their utility for software development effort estimation. To that end, a multi-organization data set made available by the International Software Benchmarking Standards Group (ISBSG) is employed in the analysis. It is at present the largest multi-organization software engineering repository available. Further details regarding the composition of the repository are provided in the research method and data analysis sections.

The ISBSG data set has been extensively used and cited by previous research, such as Mendes *et al.* 2005. Combined with the fact that it is the largest available multi-organizational software engineering repository available, it is the data set of choice for organizations without local data looking to consider estimation using global data.

### 3.1.1 ISBSG Data Collection

The ISBSG has its own pre-defined terms and metrics for the purposes of "assisting in the collection of project data into the Repository" and "standardizing the way the collected data is analyzed and reported" (ISBSG, 2006a). A simple and generic software development process is also assumed within the collection exercise so that the variety of software processes being used in different organizations can be approximately mapped to this generic process by the submitter of the data. According to the ISBSG, the collection of metrics as well as the definition of the generic software development process are updated on a constant basis to keep pace with changes in the software engineering industry and to ensure the completeness and relevance of the repository. This raises issues of completeness with respect to existing records in that new fields must inevitably be populated with null values. The intention to maintain relevance, however, is commendable.

This study used release 9 of the ISBSG repository first made available in November 2004, as this was the release available at the time the study was initiated. The data had been collected using questionnaires in accordance with the pre-defined terms and metrics, with records dating back to 1997. As there are several conventions for assessing the functional size of a software project (IFPUG/NESMA, COSMIC, MARKII and others) different versions of the questionnaire are provided to submitters as appropriate. The questionnaire enables the collection of data in a section-based structure that addresses: the Submitter, the Project, the Process, the Technology, the People and their Work Effort, the Product, the Project's Functional Size, and Project Completion. Further details can be found from the ISBSG website, while some of the relevant details of the repository are discussed in sections to follow. On submission, all of the information on a project is reviewed and rated in terms of data quality by ISBSG experts. In particular they look for omissions and inconsistencies in the data that may suggest that its reliability might be questionable. The submission process for the ISBSG is depicted in Figure 1.

Further information on the ISBSG data repository can be found on the ISBSG website at http://www.isbsg.org/isbsg.nsf/weben/Repository%20info. Certain aspects of the repository are described in the following subsection – particularly the distribution of organizations, their projects and the software sizing method used (referred to as the 'Count Approach') – as these aspects are relevant to the data grouping and should be considered to ensure that models are constructed in a valid and consistent way.

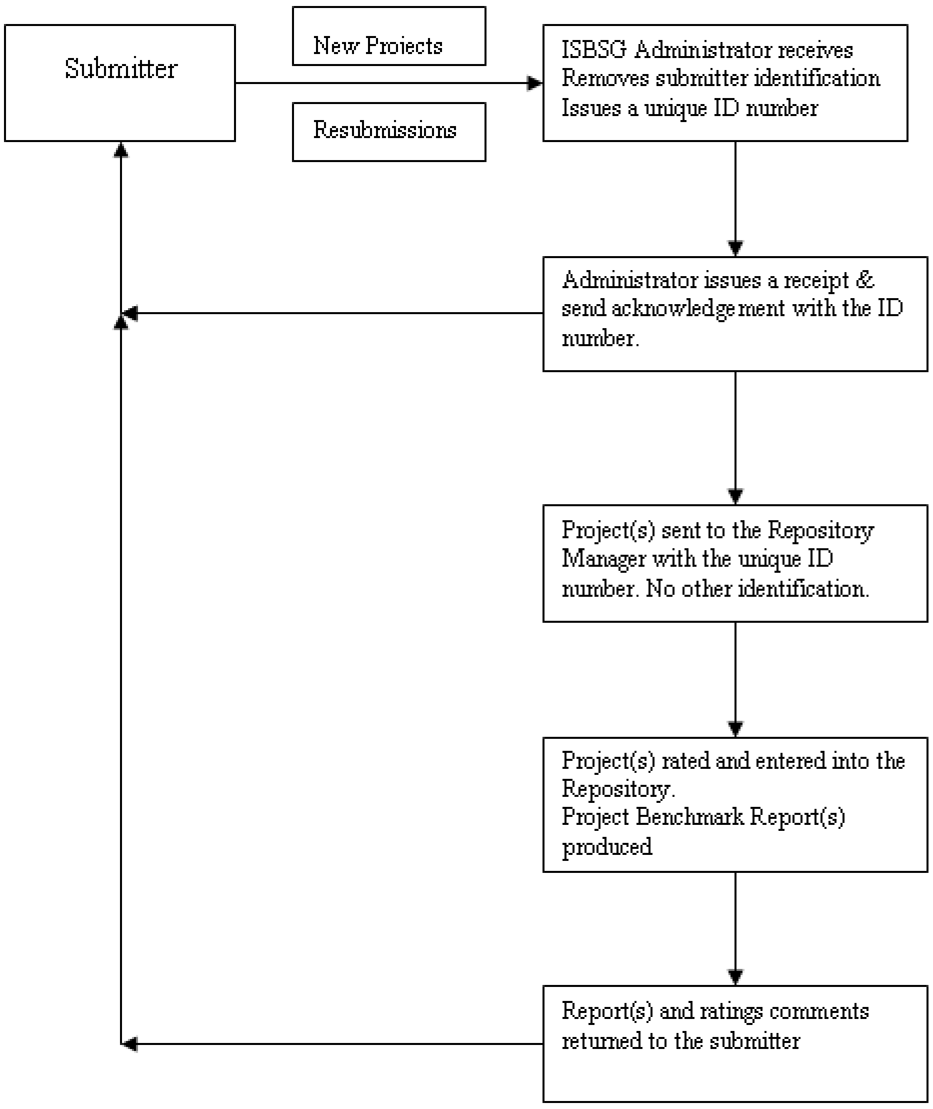

Submitter
New Projects
Resubmissions
ISBSG Administrator receives Removes submitter identification Issues a unique ID number
Administrator issues a receipt & send acknowledgement with the ID number.
Project(s) sent to the Repository Manager with the unique ID number. No other identification.
Project(s) rated and entered into the Repository. Project Benchmark Report(s) produced
Report(s) and ratings comments returned to the submitter

**Figure 1: ISBSG project submission process**

### 3.1.2 Distribution of Organizations

The version of the repository under consideration here (Release 9) contains a total of 3024 records. A separate sample has been provided to the researcher for the purposes of the study described here; 433 of the records are therefore known to be the product of ten anonymous organizations (as described in Table 3 and illustrated in Figure 2).

| **Organization** | **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **Other** | **Total** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Counts** | 9 | 65 | 87 | 9 | 13 | 21 | 91 | 51 | 11 | 76 | 2591 | 3024 |

**Table 3: Summary of project records per organization**

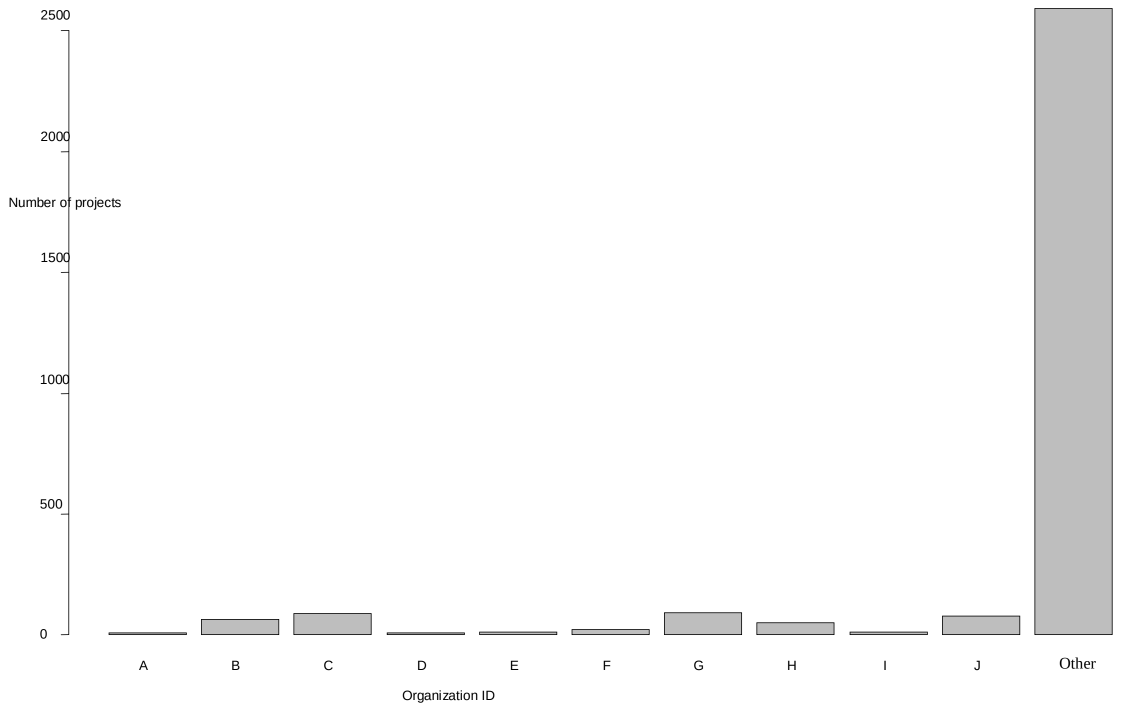


**Figure 2: Distribution of project records per organization**

Leaving the unidentified project records out of the plot, the distribution of projects across the ten anonymous organizations (labelled A to J) is shown in Figure 3.

An assumption is made in this research that the productivity of any single organization is independent of that of the rest of the organizations in the data set, an assumption that seems reasonable given the number and diversity of organizations represented. In principle then, ten different comparisons of local vs. corresponding global models can be performed. For example, for organization B, the local data set would comprise 65 observations and the corresponding global data would consist of potentially (3024-65) = 2959 observations. Further details on data grouping in terms of organizations are provided in Sections 4.3 and 4.4.4.

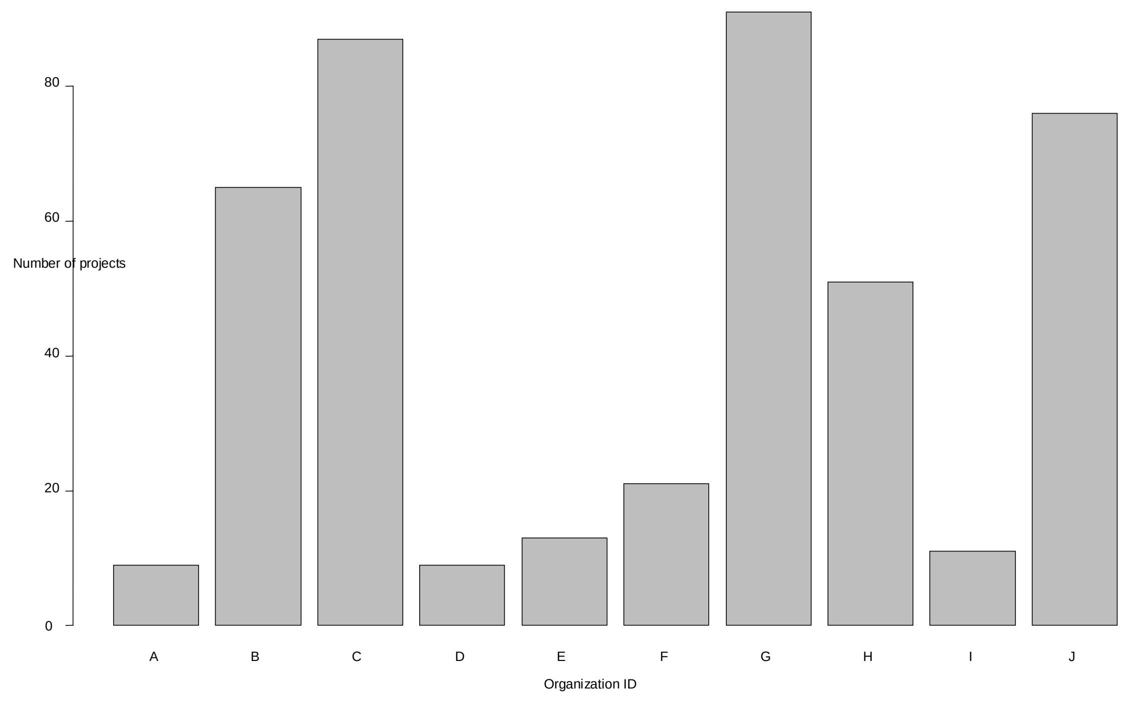


**Figure 3: Distribution of project records per identified organization**

### 3.1.3 Distribution of Software Sizing Method (Count Approach)

The main software sizing method used by organizations that have submitted data to the repository, by a large margin, is IFPUG Function Point Analysis (FPA). The following table (Table 4) summarizes the distribution of software sizing methods. Given the dominance of the IFPUG approach it would be of little benefit to describe the various approaches in detail.

| **Count Approach** | Albrecht | Automated | Backfired | COSMIC-FFP | Dreger |
|---|---|---|---|---|---|
| **Number** | 2 | 4 | 7 | 73 | 10 |
| **Count Approach** | Feature Points | IFPUG FPA | In-house | LOC | MARK II |
| **Number** | 2 | 2718 | 1 | 22 | 35 |
| **Count Approach** | NESMA | Other | Retrofitted | Unknown | |
| **Number** | 144 | 1 | 2 | 3 | |

**Table 4: Software sizing method summary in ISBSG repository**

For the vast majority of projects in the ISBSG repository, the term ‘Count Approach’ refers to the Functional Size Measurement Method (FSM Method). This is commonly used to measure the functional size of the software being produced (e.g. via IFPUG, MARK II, NESMA, or COSMIC-FFP function point analysis). For projects using other size measures (e.g. Lines of Code or LOC) ‘Count Approach’ is the short name for that method. Due to the scope of this project, the size of the repository, and the dominance of the IFPUG and NESMA FPA approaches (these two count approaches account for 93% of the records), only the observations employing IFPUG and NESMA methods have been selected for consideration. Furthermore, these two counting approaches are very similar in execution and result (so they can be combined) whereas the differences between the IFPUG (and NESMA) approach and other approaches are more pronounced.

Further statistical plots and reports describing the ISBSG repository are displayed in the Data Analysis section (Section 4). The ISBSG also reports demographics on its website to characterize the various project data types that are held in the Repository (ISBSG 2006b).

## 3.2 Research Method

This section describes and justifies the methods, techniques and tools used in this research. To reiterate, the objective of the work is to compare the accuracy of predictive models built using single-organization and multi-organization data. One or more model-building methods must therefore be utilized, drawn from the four categories of expert estimation, statistical computation, machine learning and soft computing.

### 3.2.1 Modeling Methodology

The statistical method of Ordinary Least Squares (OLS) Regression is the modeling methodology selected for use in this research. OLS Regression is often alternatively referred to as ‘Ordinary Least Squares Multiple Regression’ or ‘Multiple Regression’. More precisely, OLS Regression is used as a shortened version of ‘OLS Multiple Regression for Predictions’. ‘Ordinary’ serves to differentiate this simplest method of least squares modelling from more complicated alternatives such as weighted least squares, generalized least squares and so on. In normal statistical practice, weighted least squares is used in situations where it may not be reasonable to assume that every observation should be treated equally. Generalised-least-squares is considered difficult to implement and only deals with situations where the assumptions underlying OLS Regression are violated. ‘Multiple’ means that the variables could be either continuous or categorical or any combination of these two. ‘Prediction’ refers to the fact that the intent is to produce useful estimates, in this case to optimize the accuracy of predictions of software development effort. The OLS Regression analyst believes ‘Response = some function of the explanatory variables + random scatter’. The purpose of an OLS Regression analysis is to find the appropriate function (in statistical terms it is called a ‘model’) and make predictions by applying data to the function.

The selection of OLS Regression in this study is based on the nature of the data and the primary objective of this project, as follows:

- while there are numerous model building methodologies available, including Ordinary Least Squares (OLS) Regression, Analysis of Variance (ANOVA), analogy-based estimation and several approaches employing machine learning e.g. artificial neural networks, OLS Regression remains one of the most commonly used approaches in empirical software engineering research (Mair and Shepperd 2005). Among the primary studies reviewed previously, OLS Regression was employed by Briand *et al.* (1999), Briand *et al.* (2000), Jeffery *et al.* (2001) and others. As such it forms something of a benchmark method that should be discarded only with prior evidence that an alternative would be more effective.
- while a number of studies have been undertaken previously to evaluate and compare the performance of different modeling techniques for software cost estimation (such as Briand *et al.* (2000) and Jeffery *et al.* (2001)), to date there is no significant evidence to indicate that OLS Regression is consistently outperformed by another technique in terms of model building for software development effort estimation. Mair and Shepperd (2005, p.516) also criticized this rather simplified approach to model evaluation, saying “researchers should ask questions such as when might it be better to use technique A rather [than] B, as opposed to is technique A better than B?”.

- there are both continuous and categorical variables in the ISBSG repository. The response variable 'Effort' is continuous. This eliminates some other analysis methods such as ANOVA, ANCOVA, Logistic Regression and Poisson Regression.

Table 5 (from Kleinbaum *et al.* (1998)) provides a general guide on how to choose an appropriate multivariate method, leading to the selection of OLS Regression analysis in this study:

| **Method** | **Dependent Variable** | **Independent Variables** | **General Purpose** |
|---|---|---|---|
| **Multiple (OLS) Regression Analysis** | Continuous | Classically all continuous, but any type(s) can be used | To describe the extent, direction and strength of the relationship between several independent variables and a continuous dependent variable. |
| **Analysis of Variance (ANOVA)** | Continuous | All nominal | To describe the relationship between a continuous dependent variable and one or more nominal independent variables. |
| **Analysis of Covariance (ANCOVA)** | Continuous | Mixture of nominal and continuous (the latter as control variable) | To describe the relationship between a continuous dependent variable and one or more nominal independent variables, controlling for the effect of one or more continuous independent variables. |
| **Logistic Regression** | Dichotomous | A mixture of various types can be used | To describe how one or more independent variables are related to the probability of the occurrence of one of two possible outcomes. |
| **Poisson Regression Analysis** | Discrete | A mixture of various types can be used | To determine how one or more independent variables are related to the rate of occurrence of some outcome. |

**Table 5: Rough guide to multivariate modeling methods**

## *Confidence Level*

To carry out an OLS Regression, a confidence level must be predefined according to the precision requirements of the research. In this research, as in the majority of prior studies in this domain, the confidence level of the OLS Regression analysis is set to 95%. In such a way, a balance between model building complexity and model prediction performance should be achieved. It is contended here that the majority of organizations would be satisfied with a software development cost estimate at 95% of confidence. Estimation with higher confidence levels can be performed if it is required in the future, but it is excluded from the scope of this research.

## *Sample Size*

Within the statistics community it is known that sample size affects the power of significance testing and the generalizability of the model in an OLS Regression. Whether a sample is of adequate size is influenced by the number of predictors, the confidence level and the statistical power that the researcher wants to achieve. In regard to statistical power and sample size, Hair *et al.* (1998) illustrate the interplay among the sample size, the significance level ( $\alpha$ ) chosen and the number of independent variables in detecting a significant coefficient of determination value $R^2$ (shown as the central cells in Table 6).

| | **Significance level ($\alpha$ ) = 0.05 No. of independent variables** | | | |
|---|---|---|---|---|
| **Sample Size** | **2** | **5** | **10** | **20** |
| **20** | 39 | 48 | 64 | NA |
| **50** | 19 | 23 | 29 | 42 |
| **100** | 10 | 12 | 15 | 21 |
| **250** | 4 | 5 | 6 | 8 |
| **500** | 3 | 4 | 5 | 9 |
| **1,000** | 1 | 1 | 2 | 2 |

**Table 6: Minimum R Square that can be found statistically significant with a power of .80**

In attempting to attain a balance between generalizability and sample size, a general rule suggested by Hair *et al.* (1998) is that "the ratio should never fall below 5 to 1", meaning that five observations per independent variable is the minimum requirement to achieve generalizability, if the sample is representative of the true population (in this instance, if the ISBSG data is representative of the true population of software development projects around the globe). In this case there is no opportunity to exert control over the sampling process used to create the ISBSG repository. Rather than attempting to check whether the data is indeed representative of the true population, the conclusions drawn here should be considered to apply only in the context of the ISBSG data. For example, an organization should only apply the conclusions reached here to their own practice if at the very least the type of the organization falls into one of the organization types providing data to the ISBSG. In addition, the generalizabilty of the conclusions is impacted by the sample size of the data set. At this time, the sample size of the ISBSG data set is believed to be sufficient as a basis for valid conclusions in that it has a larger sample size than any other data set in this domain (such as the Finnish data set) and is diverse in its coverage. However, overall predictive power is still subject to the number of variables in the final model.

### *Missing Value Handling*

"One of the problems often faced by statisticians undertaking statistical analysis in general, and multivariate analysis in particular, is the presence of missing data" (Everitt and Dunn, 2001). Missing data itself often represents valuable information. For instance, respondents to a survey may be reluctant to report data that they perceive might portray them in an unfavorable light. Perhaps worse than this, the existence of missing data may cause the analyst to discard whole observations or whole features (variables), discarding other potentially valuable information in the process. It is commonly regarded as acceptable if the extent of missingness is small (for example 5% or less) because it is normally inevitable to encounter missingness in the world of inferential statistics, which by its nature attempts to portray the characteristics of a population from those of an inherently incomplete sample. Furthermore, missingness becomes almost inevitable when statistics are applied in a domain such as software engineering because the information being collected and analysed could be considered by submitters as commercially sensitive to the software organization.

In the research undertaken here, which is indeed a multivariate regression analysis, the fact is that the majority of the observations in the ISBSG repository contain missing values, due to the design of the data collection procedure and the nature of software engineering practice. For example, one of the data elements – or potential variables – collected in the ISBSG data set is the number of lines of software source code (LOC). If the sizing approach used for a particular project is IFPUG FPA, the LOC variable is purposely ignored. Two further examples are the 'Organization Type' and 'Business Area Type' ISBSG variables. Some organizations might choose to ignore the associated data collection questions for these variables in an effort to retain a degree of confidentiality. The degree of missingness evident in the ISBSG data set as a whole is described in detail in the following chapter.

Omitting observations that contain missing values in the ISBSG repository may cause a large amount of useful information to be discarded. The following facts should inform decisions in relation to dealing with missing values:

- the importance of the terms and metrics has been pre-assessed by ISBSG. As a result, a small number are compulsory to project data submitters. This could reduce a significant proportion of missing values if these compulsory factors are included or largely included in an estimation model.
- the completeness together with the quality of each record has been evaluated upon submission by the ISBSG repository manager who is an expert in this area. A quality mark is given from A to D with A being the best quality (see ISBSG (2006a) for details of quality rating criteria). Further model building and grouping processes could be repeated for high quality data if there is no significant difference between local and global data sets. In that case, a large number of (lowly rated) records that contain missing values could be purposely ignored.
- Missing Data Techniques (MDT) have been extensively discussed in the software engineering context prior to this research and could be employed here. It is commonly suggested (e.g. see Twala *et al.* (2005)) that the selection of a method to deal with missing data depends on a range of factors, such as the way in which the data is missing, the proportion of missing data, the number of variables that are involved in data missingness and so on.

Elaborating on this last point, and in light of the potentially extensive missingness in the repository, the *k*-NN (*k*-Nearest Neighbour) imputation method, also called kNNSI, seems appropriate. Further details of *k*-NN as used to handle missingness in software engineering data can be found in Jonsson and Wohlin (2004). In that study, *k*-NN was evaluated in the software engineering context using a likert-scale data set with 56 cases. Their results showed that *k*-NN outperformed a range of other uninformed and informed MDTs. Jonsson and Wohlin further commented on the importance of, and the means of, choosing an appropriate value for *k*. *k*-NN was later used as an MDT in software engineering by Twala *et al.* (2005). Eight intelligent MDTs (including Machine Learning (ML) techniques) were compared in tests conducted against the ISBSG and Finnish data sets. The *k*-NN method was considered to have performed "reasonably well". Although Expectation-Maximization multiple imputations (EMMI) performed best in the Twala *et al.* (2005) research, *k*-NN is simpler to both understand and calculate than EMMI and so is a reasonable substitute. Finally, one of the more recent and relevant research efforts in this domain, that authored by Mendes *et al.* (2005), also utilised *k*-NN in their data handling process, lending more support to the use of *k*-NN in this area.

The *k*-NN imputation method is therefore chosen in this project as the sole MDT. In this research, the k-value is derived as suggested by Jonsson and Wohlin (2004) after their extensive empirical analysis as the: "square root of the number of complete cases, rounded to the nearest odd integer".

### 3.2.2 Statistical Modeling Tool

There are a number of statistical modeling tools available that could be used to support the work undertaken in this thesis. After careful consideration, R (R Development Core Team, 2005) has been chosen as the main statistical modeling tool. R is a free software environment for statistical computing and graphics. R was selected for the following reasons:

**Accessibility**: R is free and open-source which lends greater accessibility to researchers.

**Extendibility**: R can be easily extended via packages. There are approximately eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

**Programmability**: R's language (called 'S') has an easy-to-learn syntax which contains conditionals, loops, recursive functions and import/export capabilities. It has many built-in statistical functions. This makes it possible for researchers with a programming background to write user-defined functions.

**Graphical Capability**: R has an excellent graphing capability. On top of that, there are a significant number of graphics packages developed by external open-source developers, freely available on the web.

**Help**: R has an excellent built-in help system. Extra help can also be found on the web.

**Future Commercial Support**: R projects can be easily exported to the commercial statistical software 'S-Plus' because they both have the same language, in case commercial support is needed at a certain stage in the future.

Further details of R and its language can be found at the R project website: http://www.r-project.org/. The R scripts and commands used in this research are all attached and can be found in the appropriate Appendix files. For example, Appendix 1 lists all the files used to build a model in R, and the respective commands can be found by opening those files.

Having set out the details of the data and the intended research method, the next chapter provides a comprehensive description of the data analysis activities performed in order to address the research objectives. The results of the analyses are also presented in the latter part of the next chapter.

# 4 Data Analysis

This chapter explains the data analysis process and outcomes. As per the research design, it centres on the application of the selected modeling method, OLS Regression, to the chosen data set, the ISBSG repository. This leads to the generation of concrete estimation models, in OLS Regression terms, for each organization for which the process is valid, under both local and global modeling scenarios. The process is valid for those analyses that satisfy the pre-conditions and assumptions of OLS Regression. For those data sets that do not meet these requirements, no such comparisons are made (as detailed in the material that follows).

The data analysis is composed of five steps and the results then follow (structured into subsections):

- Data Set Formalization: Examination of the data reveals that it is neither appropriate nor sensible to perform OLS Regression against the raw ISBSG repository as a whole. As a result, the raw ISBSG data set must be formalized to be of reasonable size and content. The rules and the rationale behind the data set formalization are therefore explained.
- Data Set Further Refinement: Even when a formalized and 'full' data set is acquired, it is still not entirely appropriate to perform OLS Regression analysis against it because of the intended objective of the prediction, the computational complexity of the regression analysis, and the expense of acquiring project data for individual organizations. Thus it is necessary to consider whether unimportant variables might be discarded before the actual execution of the regression analysis. The methodology, rules and rationale of the data set further refinement process are explained in the second subsection.
- Data Grouping: This subsection explains how the data is visualized during the data analysis phase and how the data is grouped into different subgroups to form OLS Regression analysis data units.
- Streamlined OLS Regression Process: The fourth subsection describes the process of OLS Regression – data screening and visualization, establishment of each model and assumption checking. The purpose of this step is to make the OLS Regression analysis both visible and repeatable in order to facilitate further improvements in future research.

- Global vs. Local Model Comparisons: In this subsection, the comparison mechanisms, rules and rationale are explained.
- Data Analysis Results: This final subsection explains the results of the local vs. global comparison for each subgroup.

Please note that the detailed OLS Regression procedures and their associated R code and output listings are attached as separate deliverables (Appendix 2).

## 4.1 Step 1: Data Set Formalization

Despite the fact that OLS Regression is reasonably robust, the characteristics of the repository mean that it is not feasible to perform an OLS Regression analysis against the raw ISBSG data set. Nor is this even advisable without some consideration of the need for pre-processing: cleaning, filtering and polishing the data (Liebchen *et al.* 2007), in light of the objectives of the particular analysis being undertaken, should always precede the analysis itself.

In this research, there are a number of issues related to the raw ISBSG data that require consideration and, in some cases, action:

- *Some variables are de-normalized.* Some of the variables in the raw ISBSG data set are simply descriptive strings rather than pre-defined categories. For example, for the variable 'Project Activity Scope', one observation reads: "Planning;Specification;Build;Test;Implement;" and another observation reads: "Specification;Build;Test;Implement;". This makes the number of distinct levels of the categorical variable very large and it is neither practical nor sensible to perform regression analysis against it. To make such variables usable in an OLS regression analysis, separate dummy variables have to be created with different levels of 'Project Activity Scope'.
- *Some variables are not recorded in a consistent format.* For example, in the variable 'Implementation Date', different date formats exist for different observations, for instance "9-Nov-00" and "prior to Feb-2004". This makes it impossible for statistical software such as R to recognize 'Implementation Date' as either a continuous or discrete variable.

- *Some variables have too many distinct levels.* The variable ‘Application Type’ comprises 105 different and distinct levels. If this categorical variable were to be included in a predictive regression model as is, 104 dummy variables would have to be included in the final equation.
- *Some variables are a mixture of different contexts.* For example, ‘Development Techniques’ in the raw ISBSG data set contains values such as “Waterfall” and “Object-oriented design”. “Waterfall” is a generic development process model description whereas “Object-oriented design” is one of the activities undertaken if a project is implemented using object-oriented methods. Another example is the ‘Intended Market’ variable whose values inconsistently describe either the location of the development, whether developed in-house or externally, or the actual intended market, whether developed for an internal or external business unit.
- *Some variables are aggregated from other variables.* For example, the ‘Adjusted Function Points’ is calculated from variables such as ‘Input Count’, ‘Output Count’, ‘Enquiry Count’ and ‘File Count’. Including both basic *and* aggregated variables contravenes the desire to have only necessary and orthogonal factors in a model.
- *Some variables are irrelevant or unavailable for predictive modeling.* Almost all of the observations in the repository were submitted after the relevant project was completed. As a result, some variables that are not known at the initialization/specification stage of projects are also included; for example, indicators of software quality (‘Major Defects’, ‘Minor Defects’), ‘Project Inactive Time’ and some productivity variables.
- *Some numerical variables have too few values.* The degree of data missingness in some variables is such that their use cannot be justified in the global models. For example, ‘Maximum Team Size’ has 1918 records out of 3024 missing. In this case, even the best MDTs cannot be applied because this is far below the minimum proportion of non-missing value requirement for MDTs. However, for any given individual organization(s) such variables could be used in a local model if there are not too many values missing. In this research, these variables are therefore noted as ‘partially kept’ in the ISBSG data set.

A straightforward solution to these issues would be to drop the variables affected. In doing so, however, we may be throwing away predictive capability. Instead, some elements of the raw data set can be formalized through a range of variable transformations, in order to enable the building of models that maximize accuracy, meaningfulness, non-redundancy in terms, and parsimony in scale. Section 4.1.1 describes the formalization rationale from the raw data set point of view. This section answers questions such as "why is variable *x* kept unchanged?", "why is variable *y* not included in the formalized data set?" and "why is variable *z* transformed?". Section 4.1.2 defines the formalized variables derived from the raw ISBSG data set. Section 4.1.3 defines the rules used to execute the formalization. The rules define how each variable in the formalized data set can be derived from the associated source variable(s) in the raw ISBSG data set. Section 4.1.4 describes the methodology used to execute the data set formalization.

### 4.1.1 Formalization Rationale

Table 7 lays out the rationale for the formalization of each variable in the raw ISBSG data set, with the variables shown in shaded boxes in column two being retained for further consideration. Note that this represents a sample treatment of the data – a different formalization may be applied if the research objective is different to the one of interest here – that is, estimation of project-level development effort. However, the underlying goal of data retention and adherence to the principle of transparent pre-processing still hold.

| ***Raw ISBSG Variable*** | ***Destination Variable in the Formalized Data Set*** | ***Rationale for the transformation*** |
|---|---|---|
| ***Organization*** | *Organization* | *No change – label.* |
| ***Project ID*** | *Project ID* | *No change – label.* |
| ***Data Quality Rating*** | *Data Quality Rating* | *No change – could be used for later filtering.* |
| ***UFP rating*** | *N/A* | *Data quality indicator. Not directly related to software effort estimation.* |
| ***Count Approach*** | *Count Approach* | *No change – will be used for later filtering.* |
| ***Functional Size*** | *N/A* | *Only 'Adjusted Function Points' is used; this variable is a component of 'Adjusted Function Points'.* |
| ***Adjusted Function Points*** | *Adjusted Function Points* | *No change.* |

| | | |
|---|---|---|
| ***Value Adjustment Factor*** | *N/A* | *Only 'Adjusted Function Points' is used; this variable is a component of 'Adjusted Function Points'.* |
| ***Summary Work Effort*** | *Summary Work Effort* | *No change,* ***the response variable.*** |
| ***Normalised Work Effort*** | *N/A* | *Only 'Summary Work Effort' is used; while potentially useful, the extrapolation performed to produce this variable is arbitrarily applied and relies on 'Project Activity Scope' being recorded, and recorded accurately. Median difference between 'Summary Work Effort' and 'Normalised Work Effort' is zero.* |
| ***Reported PDR (afp)*** | *N/A* | *Unrelated to early software effort estimation.* |
| ***Project PDR (ufp)*** | *N/A* | *Unrelated to early software effort estimation.* |
| ***Normalised PDR (afp)*** | *N/A* | *Unrelated to early software effort estimation.* |
| ***Normalised PDR (ufp)*** | *N/A* | *Unrelated to early software effort estimation.* |
| ***Project Elapsed Time*** | *N/A* | *Unrelated to early software effort estimation.* |
| ***Project Inactive Time*** | *N/A* | *Unrelated to early software effort estimation.* |
| ***Implementation Date*** | *'Implementation Start Year' and 'Implementation Start Year Range'* | *Potentially unrelated to software effort estimation at the initialization phase of a project. The raw data set is not in constant format. Furthermore, about 20% of the data are missing. Review data and form two variables.* |
| ***Project Activity Scope*** | *'Specification Included', 'Planning Included', 'Design Included', 'Build Included', 'Test Included' and 'Implementation Included'* | *De-normalised, providing too much information in one variable. Review data and form separate variables.* |
| ***Effort Plan*** | *N/A* | *Only 'Summary Work Effort' is used; furthermore (i) this can be expensive to get for software organisations and (ii) there could be some question over its accuracy.* |
| ***Effort Specify*** | *N/A* | *Same as previous.* |
| ***Effort Design*** | *N/A* | *Same as previous.* |
| ***Effort Build*** | *N/A* | *Same as previous.* |
| ***Effort Test*** | *N/A* | *Same as previous.* |
| ***Effort Implement*** | *N/A* | *Same as previous.* |
| ***Effort unphased*** | *N/A* | *Same as previous.* |
| ***Minor defects*** | *N/A* | *Unrelated to software effort estimation at the initialization phase of a project.* |
| ***Major defects*** | *N/A* | *Same as previous.* |
| ***Extreme defects*** | *N/A* | *Same as previous.* |
| ***Total Defects Delivered*** | *N/A* | *Same as previous.* |
| ***Development Type*** | *Development Type* | *No change.* |
| ***Organisation type*** | *N/A* | *Too many (more than 100) distinct levels. Mixed in context with 'Application Type' and 'Business Area Type'. This could involve some level of arbitrariness.* |
| ***Business Area Type*** | *Business Area Type* | *Re-categorize missing values to "Unspecified" and rename the labels according to standard sector descriptors.* |
| ***Application Type*** | *N/A* | *Same issues as for 'Organisation Type', therefore it is omitted.* |

| | | |
|---|---|---|
| ***Package Customisation*** | *Package Customization* | *Formalize "Unknown" and missing values in the raw data to "Unspecified".* |
| ***Degree of Customisation*** | *N/A* | *Too many missing values (2937 out of 3024, 97%) and too many distinct levels (19).* |
| ***Architecture*** | *Architecture* | *Re-categorize missing values to "Unspecified" and rename the levels according to current mainstream standards.* |
| ***Client Server?*** | *N/A* | *Too many missing values (1346 out of 3024, 45%). Considered in 'Architecture'.* |
| ***Client roles*** | *N/A* | *Too many missing values (2968 out of 3024, 98%).* |
| ***Server roles*** | *N/A* | *Too many missing values (2970 out of 3024, 98%).* |
| ***Type of server*** | *N/A* | *Too many missing values (2821 out of 3024, 93%).* |
| ***Client/server description*** | *N/A* | *Too many missing values (2291 out of 3024, 76%). Considered in 'Architecture' and 'Web development'.* |
| ***Web development*** | *Is Web* | *Re-categorize missing values to "Unspecified".* |
| ***Plan documents*** | *N/A* | *Too many missing values (2896 out of 3024, 96%).* |
| ***Specify documents*** | *N/A* | *Too many missing values (2897 out of 3024, 96%).* |
| ***Specify techniques*** | *N/A* | *Too many missing values (2961 out of 3024, 98%).* |
| ***Design documents*** | *N/A* | *Too many missing values (2907 out of 3024, 96%).* |
| ***Design techniques*** | *N/A* | *Too many missing values (2970 out of 3024, 98%).* |
| ***Build products*** | *N/A* | *Too many missing values (2895 out of 3024, 96%).* |
| ***Build activity*** | *N/A* | *Too many missing values (2951 out of 3024, 98%).* |
| ***Test documents*** | *N/A* | *Too many missing values (2896 out of 3024, 96%).* |
| ***Test activity*** | *N/A* | *Too many missing values (2984 out of 3024, 99%).* |
| ***Implement documents*** | *N/A* | *Too many missing values (2909 out of 3024, 96%).* |
| ***Implement activity*** | *N/A* | *Too many missing values (2957 out of 3024, 98%).* |
| ***Development Techniques*** | *'Main Development Process Model' and 'Object Orientation'* | *Too many different contexts are explained in this one variable. In this case, two variables are extracted from the raw variable with formalised values.* |
| ***Functional Sizing Technique*** | *N/A* | *Unrelated to software effort estimation at the initialization phase of a project.* |
| ***FP Standard*** | *N/A* | *Same as previous.* |
| ***FP Standards all*** | *N/A* | *Same as previous.* |
| ***Reference Table Approach*** | *N/A* | *Same as previous.* |
| ***Development Platform*** | *Development Platform* | *Re-categorize missing values to "Unspecified".* |
| ***Language Type*** | *Main Language Type* | *Re-categorize missing values to "Unspecified".* |
| ***Primary Programming Language*** | *N/A* | *Mixture of contexts. For example some values are "3GL" or "4GL" which is language type and some others are "IIS" which is the name of a type of web server. Some values have version numbers while some others do not. In such a case, it is very hard to get a sensible category from this variable. Furthermore, given the 'Language Type' the specific programming language cannot provide too much extra information.* |
| ***1st Hardware*** | *N/A* | *Mixture of contexts. For example: "Client/Server", "Unix" … which are not descriptions of hardware.* |

| | | |
|---|---|---|
| ***1st Operating System*** | *Main Operating System* | *Distinct values in raw dataset to be extracted and re-categorized. Levels are minimized according to the mainstream operating systems.* |
| ***1st Language*** | *N/A* | *Same as for 'Primary Programming Language'.* |
| ***1st Data Base System*** | *Main Database System* | *Same rationale and process as for 'Main Operating System' .* |
| ***1st Component Server*** | *N/A* | *Too many missing values (3001 out of 3024, 99%).* |
| ***1st Web Server*** | *N/A* | *Too many missing values (3003 out of 3024, 99%).* |
| ***1st Message Server*** | *N/A* | *Too many missing values (3017 out of 3024, 100%).* |
| ***1st Debugging tool*** | *N/A* | *Too many missing values (2786 out of 3024, 92%).* |
| ***1st Other Platform*** | *N/A* | *Too many missing values (2189 out of 3024, 72%).* |
| ***2nd Hardware*** | *N/A* | *Too many missing values (3024 out of 3024, 100%).* |
| ***2nd Operating System*** | *N/A* | *Too many missing values (2995 out of 3024, 99%).* |
| ***2nd Language*** | *N/A* | *Too many missing values (2963 out of 3024, 98%).* |
| ***2nd Data Base System*** | *N/A* | *Too many missing values (3009 out of 3024, 100%).* |
| ***2nd Component Server*** | *N/A* | *Too many missing values (3022 out of 3024, 100%).* |
| ***2nd Web Server*** | *N/A* | *Too many missing values (3023 out of 3024, 100%).* |
| ***2nd Message Server*** | *N/A* | *Too many missing values (3022 out of 3024, 100%).* |
| ***2nd Other Platform*** | *N/A* | *Too many missing values (2993 out of 3024, 99%).* |
| ***CASE Tool Used*** | *Case Tool Used* | *Re-categorize missing values to "Unspecified".* |
| ***Used Methodology*** | *N/A* | *Unrelated to software effort estimation at the initialization phase of a project.* |
| ***How Methodology Acquired*** | *N/A* | *Same as previous.* |
| ***User Base - Business Units*** | *Business Units* | *Kept in case needed by local data set which has a total of up to 100 values. Cannot be used by global data set because of too many missing values (2434 out of 3024).* |
| ***User Base - Locations*** | *Locations* | *Kept in case needed by local data set which has a total of up to 100 values. Cannot be used by global data set because of too many missing values (2384 out of 3024).* |
| ***User Base - Concurrent Users*** | *Concurrent Users* | *Kept in case needed by local data set which has a total of up to 100 values. Cannot be used by global data set because of too many missing values (2408 out of 3024).* |
| ***Intended Market*** | *'Developed Inhouse' and 'Intended Market'* | *The 'Intended Market' in the raw ISBSG dataset explains two aspects of the software development process. One is the location where the project is developed. Another is the actual intended market. Here the variable is separated.* |
| ***Recording Method*** | *N/A* | *Unrelated to software effort estimation at the initialization phase of a project.* |
| ***Resource Level*** | *N/A* | *Unrelated to software effort estimation at the initialization phase of a project.* |
| ***Max Team Size*** | *N/A* | *Too many missing values (1836 out of 3024, 61%).* |
| ***Average Team Size*** | *Average Team Size* | *Kept in case needed by local data set which has a total of up to 100 values. Cannot be used by global data set because of too many missing values (1918 out of 3024).* |
| ***Ratio of Project Effort:non-project Effort*** | *N/A* | *Unrelated to software effort estimation at the initialization phase of a project.* |
| ***% of uncollected Work Effort*** | *N/A* | *Unrelated to software effort estimation at the initialization phase of a project.* |

| | | |
|---|---|---|
| ***Input count*** | *N/A* | *An element of the 'Adjusted Function Points' already included. This variable does not provide additional value given the focus of most studies on project-level estimation.* |
| ***Output count*** | *N/A* | *Same as previous.* |
| ***Enquiry count*** | *N/A* | *Same as previous.* |
| ***File count*** | *N/A* | *Same as previous.* |
| ***Interface count*** | *N/A* | *Same as previous.* |
| ***Added count*** | *N/A* | *Same as previous.* |
| ***Changed count*** | *N/A* | *Same as previous.* |
| ***Deleted count*** | *N/A* | *Same as previous.* |
| ***Lines of code*** | *N/A* | *Lines of code approach, an alternative sizing method, is not included in the research target. Unrelated to software effort estimation at the initialization phase of a project.* |
| ***LOC not Statements*** | *N/A* | *Same as previous.* |

**Table 7: Formalization rationale explained for variables in the raw ISBSG data set**

### 4.1.2 Selected Candidate Variables

Table 8 lists the variables drawn from the raw ISBSG data set that comprise the 'full' scope from which variable selection could be reasonably undertaken. For each of the (reordered) variables, the type of the variable (continuous, ordinal or categorical) and the distinct levels of the variables (categorical variables only) are listed. The extraction of these variables is informed by the information available in the ISBSG data set, ISBSG supplied demographics (ISBSG 2006a), ISBSG Repository Release 9 Field Description (ISBSG 2006c) as well as by prior studies, experience and the intended analyses in terms of project-level prediction.

### 4.1.3 Data Set Formalization Rules

Once the set of candidate variables is selected, a rule is defined in order to actually perform the data set formalization. Before establishing the rules, each distinct value for each of the retained variables in the raw ISBSG data set is carefully examined in order to minimize confusion of concepts and to maximize both consistency and numbers of responses for each level.

As previously specified, only observations reporting a 'Count Approach' of "IFPUG" or "NESMA" are to be included in the formalized dataset. This is because observations that report the use of other count approaches do not comprise a sufficient number of observations (less than 7% of those in the repository) to explain variations in the data set.

| ***Variables*** | ***Variable Type*** | ***Notes*** |
|---|---|---|
| *Organization* | *Nominal* | *The identifier of an organization. This is used to anonymously identify the observations of the ten local organizations without having to know their names.* |
| *Project ID* | *Nominal* | *The unique identifier of the project being described.* |
| *Adjusted IFPUG Function Points* | *Continuous* | *Data is complete. The function point value of each project must be included for all projects using this as a counting approach.* |
| *Development Type* | *Categorical* | *Data is complete, all levels for this variable are provided.* |
| *Business Area Type* | *Categorical* | *Levels: Health, Insurance, Banking, IT & T, Manufacturing, Accounting, Transport, Government, Sales and Marketing, Other, Unspecified* |
| *Package Customization* | *Categorical* | *Levels: Yes, No, Unspecified* |
| *Developed Inhouse* | *Categorical* | *Levels: Yes, Partly, No, Unspecified* |
| *Intended Market* | *Categorical* | *Levels: Internal, External, Both, Unspecified* |
| *Specification Included* | *Categorical* | *Levels: Yes, No, Unspecified* |
| *Planning Included* | *Categorical* | *Levels: Yes, No, Unspecified* |
| *Design Included* | *Categorical* | *Levels: Yes, No, Unspecified* |
| *Build Included* | *Categorical* | *Levels: Yes, No, Unspecified* |
| *Test Included* | *Categorical* | *Levels: Yes, No, Unspecified* |
| *Implementation Included* | *Categorical* | *Levels: Yes, No, Unspecified* |
| *Implementation Start Year* | *Ordinal* | *But no validity in its use as a numeric predictor.* |
| *Implementation Start Year Range* | *Categorical* | *Levels: 1989-1994, 1995-1999, 2000-2004, Unspecified* |
| *Architecture* | *Categorical* | *Levels: StandardAlone, MultiTier, ClientServer, Unspecified* |
| *Is Web* | *Categorical* | *Levels: Yes, Unspecified* |
| *Development Platform* | *Categorical* | *Levels: PC, Mid-range, Main-frame, Multi-platform, Unspecified* |
| *Main Language Type* | *Categorical* | *Levels: 2GL, 3GL, 4GL, 5GL, APG, Unspecified* |
| *Main Operating System* | *Categorical* | *Levels: Mainframe, DOS, Windows, Solaris, Unix, Other, Unspecified* |
| *Main Database System* | *Categorical* | *Levels: Oracle, DB2, SQL Server, Other, Unspecified* |
| *CASE Tool Used* | *Categorical* | *Levels: Yes, No, Don't know, Unspecified* |
| *Main Development Process Model* | *Categorical* | *Levels: Waterfall, Iterative, Other, Unspecified* |
| *Object Orientation* | *Categorical* | *Levels: Yes, Unspecified* |
| *Business Units* | *Ordinal* | *Number of business units.* |
| *Locations* | *Ordinal* | *Number of physical locations.* |
| *Concurrent Users* | *Ordinal* | *Maximum number of concurrent users.* |
| *Average Team Size* | *Ordinal* | *Average number of person in the development team.* |
| *Summary Work Effort* | *Continuous* | *The response, with the unit 'person hours'.* |

**Table 8: Extracted candidate variables from the ISBSG data set as the 'full' data set**

During the formalization process, missing or ambiguous categorical values are treated as "Unspecified". Note that not all missing values can be imputed using an MDT. In this research, only missing numerical variables can be imputed using *k*-NN.

As the whole process of defining the formalization rules is manual, some of the definitions and rules could be considered to be arbitrary. However, significant effort has been expended in an effort to ensure that each decision is defendable and each rule applied consistently. Furthermore, it is contended here that retaining data, even if achieved using potentially arbitrary rules, is to be preferred over the similarly arbitrary dropping of those significant numbers of observations that have missing values. Such a decision would see a very large proportion of the data dropped – observations without missing values in the estimation-related variables account for just 20% of the data. The formalized variables, the associated rules and their rationale are shown in Table 9.

| ***Formalized Variable*** | ***Formalization Rule*** | ***Rationale of the Formalization*** |
|---|---|---|
| *Adjusted Function Points* | *No change.* | *No change needed as the variable is compulsorily required by ISBSG for FPA-based records.* |
| *Development Type* | *No change.* | *No change needed as the variable is complete and has only 4 distinct levels.* |
| *Business Area Type* | *See Table 10 'Business Area Type'* | *This variable is a mixture of Organization Type, Application Type and Business Area Type. Detailed formalization rules are explained in the separate table 'Business Area Type'. In this research, of the three variables available, only 'Business Area Type' is used.* |
| *Package Customization* | *Same as 'Package Customisation'. If the value is "Don't know" or null then "Unspecified".* | *To make it more appropriate for regression analysis by categorizing empty data into a value called "Unspecified".* |
| *Developed Inhouse* | *See Table 11 'Intended Market'* | *The 'Intended Market' variable addresses two aspects of the software development process. One is the type of the physical location in which the project is developed, the other is the actual intended market. This variable specifies "type of the physical location where the project is developed".* |
| *Intended Market* | *See Table 11 'Intended Market'* | *This variable addresses the actual intended market as indicated from the original variable 'Intended Market' by stripping off the information regarding 'Developed Inhouse' (Please see previous row for more information.)* |

| | | |
|---|---|---|
| *Specification Included* | *Yes if 'Project Activity Scope' in the raw data set contains "Specification;" or if 'Project Activity Scope' contains "Full Life Cycle", Unspecified if 'Project Activity Scope' is missing or "Don't know" or "Partial Life Cycle" No otherwise.* | *Extracted from 'Project Activity Scope' which is explaining multiple aspects at the same time.* |
| *Planning Included* | *Yes if 'Project Activity Scope' in the raw data set contains "Planning;" or if 'Project Activity Scope' contains "Full Life Cycle", Unspecified if 'Project Activity Scope' is missing or "Don't know" or "Partial Life Cycle" No otherwise.* | *Same as above.* |
| *Design Included* | *Yes if 'Project Activity Scope' in the raw data set contains "Design;" or if 'Project Activity Scope' contains "Full Life Cycle", Unspecified if 'Project Activity Scope' is missing or "Don't know" or "Partial Life Cycle" No otherwise.* | *Same as above.* |
| *Build Included* | *Yes if 'Project Activity Scope' in the raw data set contains "Build;" or if 'Project Activity Scope' contains "Full Life Cycle", Unspecified if 'Project Activity Scope' is missing or "Don't know" or "Partial Life Cycle" No otherwise.* | *Same as above.* |
| *Test Included* | *Yes if 'Project Activity Scope' in the raw data set contains "Test;" or if 'Project Activity Scope' contains "Full Life Cycle", Unspecified if 'Project Activity Scope' is nothing or "Don't know" or "Partial Life Cycle" No otherwise.* | *Same as above.* |
| *Implementation Included* | *Yes if 'Project Activity Scope' in the raw data set contains "Implement;" or if 'Project Activity Scope' contains "Full Life Cycle", Unspecified if 'Project Activity Scope' is missing or "Don't know" or "Partial Life Cycle" No otherwise.* | *Same as above.* |
| | | |

| | | |
|---|---|---|
| *Implementation Start Year* | *Manually convert 'Implementation Date' to the specified year. For example: "24/03/1999" to 1999.* | *Manually extract the year because the values need to conform to a constant format. This variable is retained in particular for the local data set modelling if no missing values are found.* |
| *Implementation Start Year Range* | *Convert the 'Implementation Start Year' to the appropriate year range: 1989-1994, 1995-1999, 2000-2004.* | *There are more than 500 missing values out of 2800 observations in 'Implementation Start Year'. Furthermore, while it might be useful in terms of time series analysis it is not sensible to use start year as a numeric predictor variable. Given that the variable may have potential worth a categorical version can be included.* |
| *Architecture* | *See Table 12 'Architecture'.* | |
| *Is Web* | *If 'Web Development' is "Web" then "Yes" else "Unspecified"* | *The raw data set only contains "Yes" and null values for this variable. If the value is null, no assumption can be made about the project, thus "Unspecified" is used in its place.* |
| *Development Platform* | *Same as 'Development Platform'. If the value is null then "Unspecified".* | |
| *Main Language Type* | *Same as 'Language Type'. If the value is null then "Unspecified".* | |
| *Main Operating System* | *If 'First Operating System' contains "Windows" OR "win" OR ".net" OR "SQL-server" OR "NT Server" OR "NT" OR "XP" then "Windows" else If 'First Operating System' contains "Mainframe" then "Mainframe" else If 'First Operating System' contains "DOS" then "DOS" else If 'First Operating System' contains "Solaris" then "Solaris" else If 'First Operating System' contains "Unix" then "Unix" else if 'First Operating System' is null OR contains "client/server" OR "custom" OR "not assessed" OR "not recorded" then "Unspecified" else "Other"* | *Distinct values in the raw data set are extracted and re-categorised. The levels are minimized according to the mainstream software development operating systems.* |
| *Main Database System* | *If 'First Database System' contains "Oracle" then "Oracle" else If 'First Database System' contains "DB2" then "DB2" else If 'First Database System' contains "SQL Server" OR "SQL-Server" OR "MS SQL" OR "MSDE" then "MS SQL" else If 'First Database System' is null then "Unspecified" else "Other".* | *The same reason as 'Main Operating System'.* |

| | | |
|---|---|---|
| *CASE Tool Used* | *Same as 'Case Tool Used'. If the value is null then "Unspecified".* | |
| *Main Development Process Model* | *If 'Development Techniques' contains "Waterfall" then "Waterfall" else If 'Development Techniques' contains "RAD" OR "Rapid Application Development" OR "Prototype" then "Iterative" else If 'Development Techniques' is null then "Unspecified" else "Other"* | *'Development Techniques' in the raw data set is a mixture of 'Main Development Process Model' and 'Object Orientation' which are two entirely different kinds of context with different criteria. Some of the values are actually explaining the detailed steps/activities in software development processes rather than the development process on its own.* |
| *Object Orientation* | *If 'Development Techniques' contains "Object oriented" OR "Object-oriented" OR "OO" then "Yes" else "Unspecified"* | |
| *Business Units* | *Same as 'User Base - Business Units', empty if value is missing.* | *Partially kept in case the missingness is not substantial in a local data set. However, this variable should only be possibly used in local data set models because of its extensive missingness in the raw ISBSG data set.* |
| *Locations* | *Same as 'User Base - Locations', empty if value is missing.* | *Same as previous.* |
| *Concurrent Users* | *Same as 'User Base - Concurrent Users', empty if value is missing.* | *Same as previous.* |
| *Average Team Size* | *Same as 'Average Team Size', empty if value is missing.* | *Same as previous.* |
| *Summary Work Effort* | *Same as 'Summary Work Effort'* | |

**Table 9: Formalization rules applied to the raw ISBSG data set**

The following tables (10 through 12) define the formalization rules for variables 'Business Area Type', 'Developed Inhouse' and 'Architecture' based on the decisions described above. With respect to the rules shown in Table 12, note that instances of 'Architecture with the value "Multi-tier/Client server" are coded to the more complex "Multi Tier" value in the formalized version of the data set, to ensure that the *potential* complexity is accounted for (as it is considered preferable to be conservative).

| ***Raw ISBSG Values*** | ***Formalized Values*** | ***Raw ISBSG Values*** | ***Formalized Values*** |
|---|---|---|---|
| *Generate & Distribute Electricity;* | *Other* | *Activity Tracking;* | *Insurance* |
| *Product Distribution;* | *Logistics* | *hardware/software/service supplier;* | *IT & T* |
| *general;* | *Unspecified* | *SOCIAL SECURITY;* | *Government* |
| *Claims Processing - Product pays claim;* | *Insurance* | *Mining Production Information;* | *Manufacturing* |
| *health insurance;* | *Insurance* | *Accounting;* | *Accounting* |
| *Fine Enforcement;* | *Other* | *Insurance;* | *Insurance* |
| *Postal System;* | *Government* | *Sales & Marketing;* | *Sales and Marketing* |
| *Telecommunications;* | *IT & T* | *Valuation;* | *Unspecified* |
| *Regulatory agency;* | *Government* | *Financial (excluding Banking); Accounting;* | *Finance* |
| *don't know;* | *Unspecified* | *Personnel;Health, Safety & Environment;* | *Health* |
| *Banking;* | *Banking* | *Mail house service;* | *Government* |
| *Manufacturing;* | *Manufacturing* | *Transport/Shipping;* | *Logistics* |
| *Procurement;* | *Logistics* | *Logistics;* | *Logistics* |
| *TRANSPORT;* | *Logistics* | *Actuarial System - calculate employer rates.;* | *Accounting* |
| *Provide computer systems and IT Consultation;* | *IT & T* | *Financial (excluding Banking);* | *Finance* |
| *Service Management;* | *Other* | *regulation;* | *Government* |
| *Providing IT Companies diff. clients and vice versa;* | *IT & T* | *Public Administration;* | *Government* |
| *Financial (excluding Banking); Property valuation;* | *Finance* | *Architectural;* | *Unspecified* |
| *Network card administration;* | *IT & T* | *Project management & job control.;* | *Engineering* |
| *Personnel;Banking;* | *Banking* | *Matching jobs with workers;* | *Other* |
| *Software house / services;* | *IT & T* | *Organizational reporting;* | *Other* |
| *Driving submarine;* | *Engineering* | *hardware/software/services provider;* | *Manufacturing* |
| *Research & Development;* | *Unspecified* | *Customs;* | *Other* |
| *Financial (excl Banking) & Banking;* | *Finance* | *Medical and Health Care;* | *Health* |
| *Legal;* | *Other* | *Administration;* | *Unspecified* |
| *IS;* | *IT & T* | *Actuarial;* | *Insurance* |
| *Blood Bank;* | *Health* | *Telecommunications network manager;* | *IT & T* |
| *Personnel;* | *Unspecified* | *TESTING;* | *IT & T* |
| *Case Management;* | *Insurance* | *Government;* | *Government* |
| *Personnel;Education;* | *Other* | *Personnel;Manufacturing;* | *Manufacturing* |
| *Library Management;* | *Other* | *Other;* | *Other* |
| *Social Services;* | *Government* | *Licensing;* | *Government* |
| *Environment;* | *Government* | *Distribution & Transport;* | *Logistics* |
| *Sales;* | *Sales and Marketing* | *Financial;* | *Finance* |
| *Financial (excluding Banking);Accounting;Inventory;* | *Finance* | *Pension Funds Management;* | *Finance* |
| *Network Management;* | *IT & T* | *Dispute Resolution;* | *Government* |

| | | | |
|---|---|---|---|
| *Engineering;* | *Engineering* | *Document registration;* | *Other* |
| *Customer Configuration Mgmnt;* | *Unspecified* | *Purchasing;* | *Manufacturing* |
| *TRAFFIC/TRANSPORT;* | *Logistics* | *Marketing;* | *Sales and Marketing* |
| *Contract Management;* | *Other* | *Energy generation;* | *Manufacturing* |
| *Inventory;* | *Unspecified* | *Ocean Transportation;* | *Logistics* |
| *Chartered flight operation;* | *Logistics* | *Regulatory hazardous waste movement monitoring;* | *Government* |
| *Loans;* | *Banking* | *NULL (value is missing)* | *Unspecified* |
| *Financial (excluding Banking);Personnel;* | *Finance* | *Service;* | *Unspecified* |
| *Registration, racing;* | *Other* | *Systems Integration;* | *IT & T* |
| *Provide computer services and IT consultation;* | *IT & T* | *Parking;* | *Unspecified* |
| *EPOS;* | *Banking* | *Defence;* | *Government* |
| *PUBLIC HEALTH & FAMILY SERVICES;* | *Health* | *Registration, racing administration;* | *Government* |
| | | *Distribution/Scheduling;* | *Logistics* |

**Table 10: Formalization rules applied to 'Business Area Type'**

| ***Raw ISBSG Values(Intended Market)*** | ***Formalized Values (Developed In-house)*** | ***Formalized Values (Intended Market)*** |
|---|---|---|
| *Outsourced for internal business unit;* | *No* | *Internal* |
| *Customer & users 1 org, team in another;* | *No* | *External* |
| *Customer, users, team in different orgs;* | *No* | *External* |
| *In-house for internal business unit;* | *Yes* | *Internal* |
| *Partly outsourced and partly inhouse;* | *Partly* | *Partly* |
| *Customer, users & team in same org;* | *Yes* | *Internal* |
| *In-house for all internal business units;* | *Yes* | *Internal* |
| *Customer & team 1 org, users in another;* | *Yes* | *External* |
| *In-house for external business unit;* | *Yes* | *External* |
| *In-house for internal business unit;In-house for external business unit;* | *Yes* | *Both* |
| *Dev in-house for use by ext agen req to rept to us* | *Yes* | *External* |
| *External for external business unit;* | *No* | *External* |
| *Inhouse for bank customers;* | *Yes* | *Internal* |
| *NULL (value is missing)* | *Unspecified* | *Unspecified* |

**Table 11: Formalization rules applied to 'Intended Market'**

| ***Raw ISBSG Values*** | ***Formalized ISBSG Values*** |
|---|---|
| *Multi-tier / Client server* | *MultiTier* |
| *Multi-tier* | *MultiTier* |
| *Multi-tier with web public interface* | *MultiTier* |
| *Stand alone* | *Standard Alone* |
| *Client server* | *Client Server* |
| *NULL (value is missing)* | *Unspecified* |

**Table 12: Formalization rules applied to 'Architecture'**

### 4.1.4 Data Set Formalization Execution

The data set formalization execution is straight-forward after all the candidate variables and the formalization rules are defined. The following process is followed:

- All raw data is transferred to a Microsoft SQL Server 2005® database table.
- An empty data base table is created by using the candidate variables (defined in Section 4.1.2) as the schema.
- The database creation scripts including both the schema and the data are generated for the purpose of testing and future development.
- A simple Windows application running under Microsoft .NET Framework 2.0® is developed. The application, written in C#, iterates through each observation in the raw ISBSG data set through the SQL server, and executes the transformation according to the rules defined in Section 4.1.3.
- After formalization, the data set is examined and tested by comparing the proportions of each level of the variable with the raw ISBSG data set.

## *4.2 Step 2: Further Refinement of the Data set*

After a formalized and 'full' data set is created from the raw ISBSG data repository, further refinements still need to be considered.

### 4.2.1 Rationale for Further Refinement of the Data Set

In the 'Data Set Formalization' step, all the appropriate and *potentially* useful variables and observations in terms of project-level software effort estimation are pre-processed from the raw ISBSG data. Some values have been modified with justification in order to produce sound categorical variables for regression analysis and/or to deal with missing values. This is just a first step, however; the following issues now need to be considered:

- Taking into account of the principles and conventions of software engineering, it is not meaningful to include some of the available variables in an effort estimation model. Weisberg (1985) argued that "the single most important tool in selecting a subset of variables for use in a model is the analyst's knowledge of the substantive area under study." He then criticized the action of including all variables in multiple regression models as "throwing everything in the hopper" simply because they are available.

- It is also difficult to make statistical inference from an overly-complicated regression model because it becomes difficult to explain and anticipate the impact of the overall model given certain input conditions. As a result the model may be problematic to utilize in a production environment. It is also difficult to explain the relationship between the response and the many independent variables, given that there may be interaction effects among the independent variables.
- The calculation of a regression model can be computationally expensive. For example, in this research, there are 22 variables in the formalized data set, 21 of which are categorical variables. If all the variables with all the interactions were to be included in a model, the number of potential components in the final model equation could be (21! + 1). Formulating such a model over the potentially large number of observations – in this case a data set comprising more then 2800 observations – would challenge the processing limitations of current desktop PCs as generally used by project managers.

### 4.2.2 Further Refinement Rules and Explanations

While there is no definitive suggestion as to the maximum number of candidate variables that should exist in a full data set, there is an accepted trade off between accuracy and parsimony. Finding an optimum model should be informed by software engineering principles and relevant personal experience. With this in mind, all the variables retained so far are considered to decide whether they should be kept in the full data set for further study.

Primarily, three principles inform the decision to keep or drop a variable at this point:

- A variable should be dropped if too great a degree of effort has to be expended in order to decide the value of it in the process of software/systems development, given that estimates of effort are often first needed in the very early stages of development. For example, decisions regarding ‘Main Database System’ and ‘Architecture’ could require a substantial amount of work in some cases, or might only be made once design decisions have been confirmed. In keeping with the scope of this research, effort is targeted to support decision makers who may not possess substantial knowledge about the specific solution domain of the project at an early stage.

- A variable should be dropped if its value cannot be specified in the initial phase of the project (for example, 'Object orientation').
- A variable should be dropped if there is no conceptual justification for its contribution to a predictive model of software development effort, or if its inclusion would not make any sense to the target audience i.e. the decision makers. For example, 'Test Included' should not be included because it does not make any sense to say "We will implement this project, but the project is not going to be tested."

In light of the above, Table 13 describes the relevant 'Keep/Drop' decisions and the rationale for each, for the 21 potential predictors.

This step represents the end of the data set refinement process. At this point there exists a refined global data set that is usable in terms of OLS Regression analysis. Following this, subsets of the data set can be extracted and separate regression analyses performed against these data subsets to compare the effort estimation performance of local and global data sets. In each of the regression analyses, further data screening and variable selection need to be undertaken to suit the particular characteristics of the corresponding data set.

## 4.3 Step 3: Data Grouping

Now that the full data set is established, the next central task is to select groups of data for local and global model comparison. With generous agreement from the ISBSG the observations of ten different organizations have been identified in the repository, although naturally the details of each organization are still unknown. This makes it feasible to group all the observations from each organization into a local data set. For each organization's local data set, their corresponding global data set can also be easily identified by grouping all the observations that do not belong to the specific organization.

| *Variable* | *Action* | *Explanation* |
|---|---|---|
| *Adjusted Function Points* | *Keep* | *Indicator of project scale. Available quite early, prior evidence of relationship with effort.* |
| *Development Type* | *Keep* | *Indicator of project type. Available early, prior evidence of relationship with effort.* |
| *Business Area Type* | *Keep* | *Indicator of project domain. Available early, prior evidence of relationship with effort.* |
| *Package Customization* | *Keep* | *Indicator of project type. Available early, possibly related to effort.* |
| *Developed In-house* | *Keep* | *Indicator of project structure. Available early, possibly related to effort.* |
| *Intended Market* | *Keep* | *Indicator of project structure. Available early, possibly related to effort.* |
| *Specification Included* | *Drop* | *From the point of view of modern software engineering principles, there is no reason why this activity should not be included in any given project.* |
| *Planning Included* | *Drop* | *Same as previous.* |
| *Build Included* | *Drop* | *Same as previous.* |
| *Test Included* | *Drop* | *Same as previous.* |
| *Implementation Included* | *Drop* | *Same as previous.* |
| *Design Included* | *Drop* | *Same as previous.* |
| *Implementation Start Year Range* | *Keep* | *Indicator of project context. Can be estimated early, possibly related to effort.* |
| *Architecture* | *Drop* | *In reality, software developers can expend substantial effort in order to reach a decision as to which architecture to use, by investigating the solution domain and the availability of current technology. Therefore, at the time when effort estimates are first needed, decision makers may not have decided on the architecture to use.* |
| *Is Web* | *Drop* | *The levels of this variable are only "Yes" and "Unspecified". In reality, a project could be a combination of web and other types depending on the chosen architecture.* |
| *Development Platform* | *Keep* | *Indicator of project technology. Available early, possibly related to effort.* |
| *Main Language Type* | *Keep* | *Indicator of project technology. Available quite early, possibly related to effort.* |
| *Main Operating System* | *Keep* | *Indicator of project technology. Available quite early, possibly related to effort.* |
| *Main Database System* | *Drop* | *To make the decision as to which database system to use, a significant amount of effort would normally be expended. For example, comparing the performance capabilities, benchmarking and proof-of-concept documentation. In reality, organisations tend to favour one or more particular DB systems (as per Architecture) but even this varies over time and (for bespoke systems) depends on customer needs.* |
| *CASE Tool Used* | *Keep* | *Indicator of project technology. Available quite early, possibly related to effort.* |
| *Main Development Process Model* | *Keep* | *Indicator of project process. Available early, possibly related to effort.* |
| *Object Orientation* | *Drop* | *This decision would normally be made at the design phase.* |

**Table 13: Final decisions regarding retention of potential predictor variables**

However, it is not in fact possible to use the data from all ten organizations in this research because some employed software sizing methods that are not represented extensively in the repository. Within the function point community, different software sizing methods have different magnitudes in terms of calculated function points and they also differ in their capabilities and counting algorithms. It is therefore not reasonable to compare software projects sized via function points that have been counted using two entirely different software sizing methods. That said, more than 90% of the observations in the repository have been collected using software sizing methods IFPUG FPA and NESMA FPA, the latter being commonly regarded as a later version of IFPUG, as illustrated previously. Details of the IFPUG counting method can be found at http://ww.ifpug.org, while the NESMA approach is described at http://www.nesmasurf.org. In this research, then, IFPUG and NESMA are treated as one software sizing method. This is supported by NESMA (2006): "[The counting differences between IFPUG and NESMA] have a negligible impact on the results of function point counts".

The distribution of software sizing method in the ISBSG data set is therefore taken into account at all steps in this research. At the data set formalization step, all the observations whose 'Count Approach' was not IFPUG or NESMA were discarded in order to achieve a higher degree of reliability and generalizability in the research outcomes. The bar-plot in Figure 4 illustrates the distribution of 'Count Approach' versus 'Organization' in the raw ISBSG data set. The right-most unmarked item in the X-axis is the collection of all the observations whose organization is not identified.

As can be seen in Figure 4, IFPUG has been used exclusively in organizations C, D, E, F and G. COSMIC-FFP has been exclusively used in organization I. Organizations A, B, H and J employ a mixed count approach. Of those, IFPUG is the primary count approach methodology in organizations B and J and NESMA the dominant method in organization H.

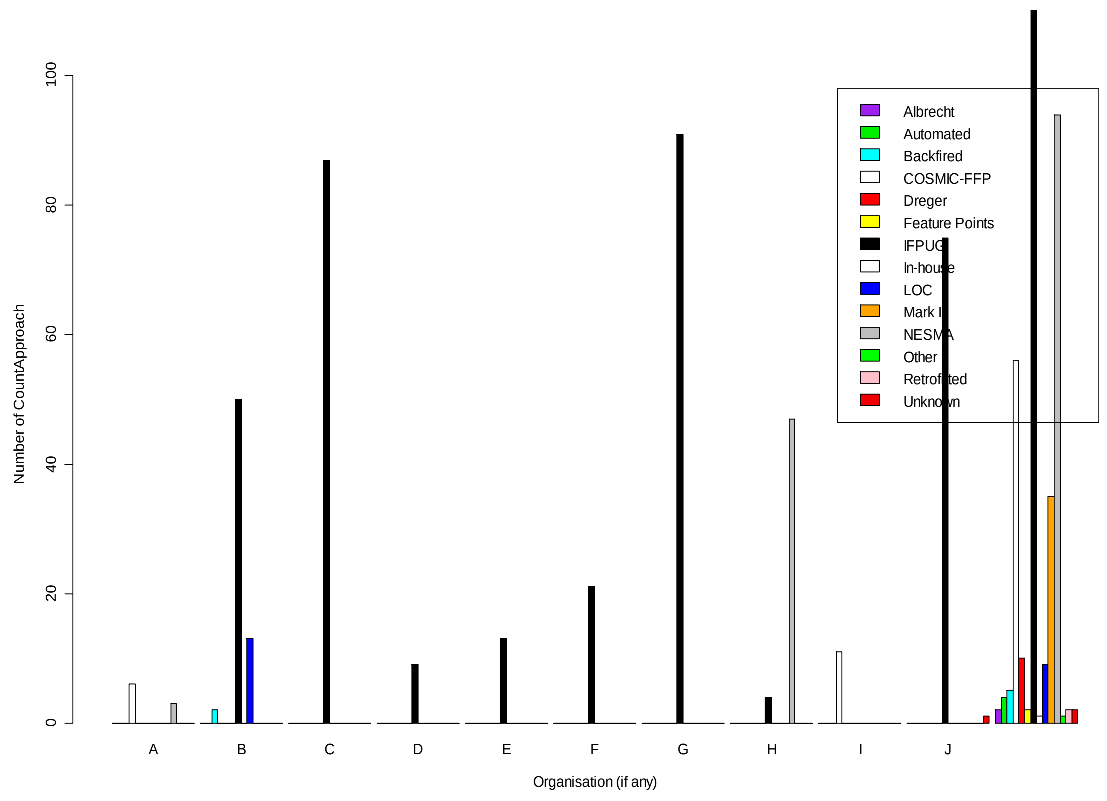


**Figure 4: Bar-plot of 'Count Approach' vs. 'Organization' in ISBSG data set**

After observations with other count approaches are discarded, the organizations remaining are: B, C, D, E, F, G, H and J. Therefore, eight different groups of local versus global data sets can be composed. They are B versus "not B", C versus "not C", D versus "not D", E versus "not E", F versus "not F", G versus "not G", H versus "not H" and J versus "not J".

In order to fully utilise the information in the ISBSG data set, the data for organizations B and J are included in the model comparison. Although these two organizations do not exclusively using a single count approach (or a combination of only IFPUG and NESMA), the majority of their projects are sized using the IFPUG method. As a consequence, the local data sets utilized for organizations B and J include only those projects sized using the IFPUG approach. Their corresponding global data sets are the offsets of their organizations' *whole* data sets. In another words, the observations that refer to projects sized by other count approach methods are excluded from this research.

## 4.4 Step 4: Streamlined OLS Regression Process

### 4.4.1 Introduction

Although OLS Regression is commonly regarded as a systematic way to perform modeling using a given set of data, many variations are possible at each step. For example, there are a number of ways of depicting model diagnostics; different researchers prefer to use different parameters to determine whether a specific variable is needed in a model; and so on. In order to achieve visibility, repeatability and traceability, the OLS Regression process in this research has been streamlined so that these decisions are clearly specified and are applied consistently. As a result, the models created from OLS Regression processes in this research can be easily examined, evaluated and extended. Figure 5 illustrates the general steps of a typical OLS Regression process.

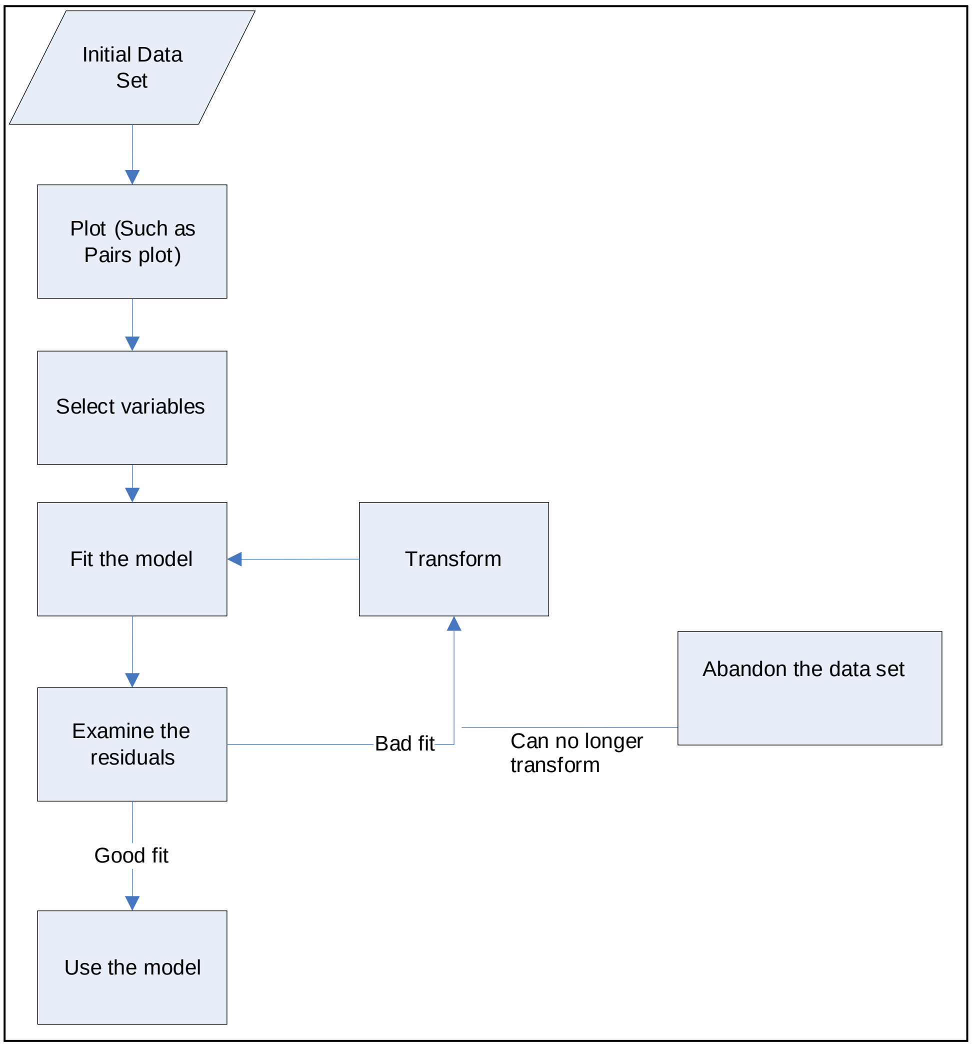


**Figure 5: OLS Regression modeling process**

The detailed steps and the chosen regression parameters, as well as the rationale for each, are explained in the following sections.

### 4.4.2 Data Screening

In this research, the local data are designed to be used in three different contexts: local model building, local model estimation and global model estimation. Note that when a global model has been established, the performance of the global model is evaluated by applying the global model to the local data, in order to assess its value as per the objectives of the research.

However, in order to establish a regression model from *each* organization's local data set and their corresponding global data set, further data screening is still needed. There are a number of reasons for this:

- There might be no variance for a specific categorical variable in an organization's data set. For example, the development type for all projects undertaken by an organization might be "New Development". Therefore, the variable 'Development Type' should be dropped from the organization's local data set when building the local model and making predictions, because in this context this variable cannot provide any discriminatory information in terms of model building.
- A data set might include an important categorical variable that has more than two levels. However, there might be one or two values for the variable that are different from the others. In this case of unbalanced data, if the observations in this level were included, the final model might be biased due to there being some but too few observations. In order to keep this variable, observations that have the 'minority' value should be dropped.
- Some ordinal variables such as 'Implementation Start Year' and 'Average Team Size' were 'partially kept' in the data formalization step, because while there were too many missing values in these variables as far as the *whole* ISBSG data set is concerned, in some local data sets the missingness might not be so extensive. Therefore, it is more appropriate to use the original raw values or to use Missing Data Techniques (MDT) to estimate the missing values than the transformed variables which treat missing values as separate categorical data. For example, in the total repository the 'Implementation Start Year' variable has

more than 20% of the values missing and it is neither practical nor appropriate to use any of the MDTs to estimate the missing values for this variable. However, in the local data set of organization C, all the values for this variable are complete. Therefore, it is more appropriate to use the actual value drawn from the raw ISBSG data set than the relevant transformed categorical variable - 'Implementation Start Year Range'. As a consequence, in this case, the variable 'Implementation Start Year Range' should be dropped in the local data set of organization C.

- Some values of a numerical variable are not valid; for example, one of the values of 'Summary Work Effort' for organization G is 0. This is either incorrect – perhaps a data recording error – or represents a project that was never started; in either case the observation is not valid in terms of project effort estimation and should be discarded.

In keeping with the reasons just stated, the following heuristic rules are established for the localised data screening:

- Drop variables that have more than 90% identical values OR variables with only two levels but where the number of observations in a level is less than 3.
- Drop observations that belong to a variable that has more than 2 levels and the number of values for any given level is less than 5, or 5% of the total number of observations; however, if the number of observations is larger than 20, then this variable is still to be kept.
- If variable 'Implementation Start Year' is complete within a certain local data set, then use 'Implementation Start Year' instead of its transformed counterpart ('Implementation Start Year Range').
- If the percentage of missing values of a certain numerical variable is less than 20% then use *k*-NN MDT (*k*-Nearest Neighbour missing data technique) to estimate the missing values of the variable.
- Drop observations that have either 'Summary Work Effort' equal to 0 or 'Adjusted Function Point Count' equal to 0.

By following these rules, each OLS Regression analysis target data set is further processed. The list of changes made according to these rules is logged as an appendix (Appendix 3) for further reference and auditing. The practical assumption underlying the data screening process is as follows: it is invalid to establish an OLS Regression model based on unreliable or biased data, therefore, it is necessary to carry out this kind of data screening even at the cost of excluding some data. As a consequence, the models established from each data set can only be applied to the context appropriate to the specific data set provided.

Note that although this leads to the dropping of some variables and observations in the data sets, there is a clear distinction between the handling of variable deletion and observation deletion in some situations. When a *variable* is deleted for the reason of having all identical values in a local data set, it is excluded from being used *in the local model building and estimation*. When the local data set is used in global model estimation for other organizations, the variable is still considered for inclusion. This is because the variable could provide extra information that distinguishes that organization from some other organizations in the global context. For example, consider the case in which the business area of a software organization is banking. When building the local model based on data from this organization, the fact that the business area is banking across all observations does not provide any differentiating information. However, when a global model is established in which the business area being banking could make a significant difference compared to other business area types, the data becomes valuable and the factor becomes potentially useful in practice. However, if an *observation* is deleted from a local data set screening process, the observation is then permanently deleted from the data set even in the situation of global model estimation.

### 4.4.3 Variable Selection

Although an individual model could be established using all the selected variables with all their two-way interactions, such a variable selection algorithm is potentially resource-expensive. This is because variable selection algorithms use a continuous approach to establish sub-models out of the full model, at each step comparing the performance of each sub-model to all others.

Two approaches are therefore generally used to obtain a balance between accuracy and parsimony: stepwise regression and all possible regression.

- Stepwise regression: A sequence of hypothesis tests is performed in order to remove variables from the regression (backward elimination), add variables to the regression (forward selection), or do both simultaneously, one variable at a time.
- All possible regression: For each subset of variables, define a criterion of ‘model goodness’ which tries to balance over-fitting (the model is too complex) with under-fitting (the model does not explain much/enough of the variance). Calculate the criterion for each of the 2k-1 models, where k is the number of variables. Then, pick the best one according to the criterion.

In considering these two approaches, all possible regression provides greater flexibility in variable selection, because it is acceptable to use any selected criterion in choosing a subset of variables, rather than simply depending on one single indicator as is the case for stepwise regression. However, all possible regression is much more resource consuming than stepwise regression, in terms of both time and computation. In this research, even for a local data set with 85 observations and 13 variables among which ten are categorical, the all possible regression approach is too resource-costly for normal personal computers to execute in reasonable time. The decision is therefore taken at this point to discard the all possible regression approach to variable selection.

As a consequence, stepwise regression with the forward selection option is to be used in order to optimise the size of each final model. Akaike Information Criterion (AIC) is to be used as the key value in indicating whether to add each variable. AIC is a measure of prediction error, where a low value for AIC signifies a good model (Burnham and Anderson, 2004). AIC is considered to be almost equivalent to the t statistic (Venables and Ripley, 2002) which is used by the majority of statistical analysis software packages. The t statistic is not used in this research because the stepwise regression function in R only supports AIC.

Reflecting on the above process it is acknowledged that there is much debate around and criticism against the use of automated variable selection algorithms such as all possible regression and stepwise regression (see, for example, Derksen and Keselman (1992) and Henderson and Velleman (1981)). However, these papers in essence suggest that researchers use domain-specific knowledge as much as possible when doing manual variable selections instead of relying heavily on automated variable selection algorithms. They argue that "data analysts know more than the computer" (Henderson and Velleman, 1981). The view held here is that the use of a *combination* of domain knowledge and automated (and therefore efficient) variable selection may result in an optimal outcome. In line with this view, effort has been expended to utilise domain-specific knowledge in the previous steps of data formalization, data set further refinement and data screening in order to discard inappropriate variables as much as possible, before automated variable selection algorithms are executed by a computer program such as R.

Also important in terms of obtaining valid models is the application of an assumption checking process at various stages of the stepwise regression procedure. However, this is left to the final stages when the models are about to be established. This is due to the following reasons:

- The literature is not clear as to whether assumption checking should be undertaken at every stage of the stepwise regression process or in relation to a final model. In Mendes *et al.* (2005), their assumption checking is conducted prior to every stage of the stepwise regression process. However, it appears that they do not perform assumption checking *after* the stepwise regression process, i.e. at the time when the final model is about to be drawn.
- By checking the assumptions just before the final model is about to be drawn the validity of the model can be assured.
- From a more pragmatic point of view, the stepwise regression process is automated using R, and in this study there are more than 20 models to be established with at least 20 variables being considered in each model. It is therefore simply not practical in this case to check every sub-model during the stepwise regression process.

Since stepwise regression with the forward selection option is chosen as the method of conducting variable selection, the execution is straightforward. The following procedures are followed for each data set:

- Create the initial model: Create a null model without any variables.
- Create the scope: A formula defines the full model. In this research, the full models are uniformly established with all the categorical variables and all of their two-way interactions plus the linear terms of the numerical variables.
- Execute the Step function in R by specifying the direction as 'Forward'.

Due to the size and dimensionality of the data sets only the interactions between two explanatory variables (two-way interactions) are considered. All the two+-way interactions are ignored – it is contended here that two+-way interactions are unlikely to be of practical interest to organizations, so stopping at two-way interactions is pragmatically sensible.

### 4.4.4 Data Set Visualization

Before establishing any model, it is useful to first explore the data using graphs and summary statistics in order to get some general idea about the data, its distribution, and any potential relationships among variables.

A variety of plots are available for exploratory analysis. In this research, pair scatter plots are utilized to show the relationships among some potentially important variables and between each predictor variable and the response variable ('Summary Work Effort'). This particular visualization is utilized because the number of variables in the data sets in this research is high, which makes it impossible to display all the combinations of variables. By looking at pair plots, it is feasible to consider the following questions:

- Is there any relationship between the specific explanatory (predictor) variable and the response?
- If so, is it a positive or negative relationship?
- Is there any relationship between explanatory variables which could be a sign of multi-collinearity?

The latter consideration is important since models that incorporate interrelated predictor variables (i) tend to be unstable in behaviour, and (ii) do not adhere to the principles of variable orthogonality and model parsimony.

Along with each plot, associated summary statistics are reported. These can give an indication of the likely validity of any subsequent regression analysis. The following information is produced:

- If the variable is a numerical variable the five-number-summary is produced (comprising the minimum, lower quartile, median, upper quartile and the maximum values), conveying information that describes the distribution of the values for that variable
- If the variable is categorical, the count of observations at each level within the variable is produced.

The summary statistics can assist the researcher in identifying whether there are any outliers in a numerical data set and whether the distribution of the values for a numerical variable is normal, skewed and/or peaked. They can also provide an indication as to whether a categorical variable has sufficient data in each level. If not, further data screening rules may need to be applied.

### 4.4.5 Model Establishment and Model Diagnostics

Each model is established in a systematic step-by-step fashion (as illustrated previously in Figure 5). The model established from the variable selection process is used as the initial model. A full set of diagnostic tests are carried out in order to check the underlying assumptions of OLS Regression. Table 14 shows a list of assumptions, problems, solutions and diagnostics methods applied during model-building (from Everitt and Dunn (2001)).

Among the assumptions listed, that of ‘No Multi-collinearity’ can be considered to be inherently satisfied because the process of stepwise regression is designed to make sure that only necessary variables are included in the final model. Therefore, correlated explanatory variables (resulting in multi-collinearity) will not be included.

| ***Assumption*** | ***Problem if Violated*** | ***Solution*** | ***Diagnostics*** |
|---|---|---|---|
| Residuals are planar | Regression coefficients meaningless. | Transform explanatory variable(s) or add extra terms. | Residual versus fitted value plots. |
| Residuals have a constant scatter | Model biased. | Transform the response variable. | Residual versus fitted value plots. |
| No influential outliers | Model biased. | Investigate the outliers and try to discard the outliers if possible and valid, or report outliers separately. | Plot of squared residuals versus Hat Matrix Diagonals. Observations with big squared residuals and big HMDs can be considered as outliers. |
| No Auto-correlation i.e. residuals are independent | Biased parameter estimates. | GLS (*generalized least squares*) Regression | Durbin-Watson test, small p-value indicates evidence of violations against residuals being independent. |
| Residuals are normally distributed | Usually indicates outliers. Also influences the accuracy of the predictions. | Transform the response variable. | Normal Q-Q plot. Weisberg-Bingham test, test statistic close to 1 indicates normality. |
| No Multi-collinearity | Individual p-values inflated | Remove redundant explanatory variable(s) | N/A |

**Table 14: Assumptions, problems, solutions and diagnostics of OLS Regression**

After each model-building iteration, a summary of the model is produced to provide indications of the quality of fit, such as the percentage of variance explained by the model and the significance of each individual explanatory variable. During the model building process, if it becomes evident that a response variable should be transformed, a Box-Cox plot and the shape of the residual plots are utilised to identify an appropriate transformation. If any of the explanatory variables need to be transformed, a partial regression plot plus actual domain knowledge regarding the particular variable are used to indicate an appropriate transformation power.

For the analysis of each data set, the R code, output and decisions made (with their rationale) are all attached for further reference in Appendix 2.

## *4.5 Step 5: Local vs. Global Model Comparisons*

### 4.5.1 Prediction Generation and Cross-validation

When a model is established, predictions then need to be made by applying the model to a specific data set. In this research, since it entails a comparison of estimation performance between local and global models *in a local context*, both models are used to predict the response variable in the associated local data set.

There are some specific issues related to the inclusion of variables in both the local and global model estimation that warrant discussion:

- *Variables used in a given global model (e.g. for set "not B") can be different to those variables used in the associated local model (i.e. for set "B"):* In the data screening process prior to the variable selection, when making the data set for producing the local model, variables that have little or no variance may be dropped (as discussed previously). This is because these variables are not applicable to be used in the local model. However these variables could be meaningful in the context of the global model. Therefore, the global model predictions may utilise more variables than the local model predictions.
- *Each local observation's local model can be different:* Each local model is constructed from a local data set. In this case, it is essential that the OLS Regression analysis for model-building be undertaken using a *subset* of the local data set while retaining the other part of the local data set for prediction,

in order to meet standard expectations regarding cross-validation. It is generally held that there are three different types of cross validation: holdout cross-validation, k-fold cross-validation and leave-one-out cross-validation. In this research, in order to achieve repeatability and consistency across local data sets of varying sizes, leave-one-out cross-validation is used. This involves using a single observation from the original sample as the to-be predicted (or test) data, and the remaining observations form the training data. This is repeated such that each observation in the local data set is used once as the to-be predicted data. By using the formula for calculating project effort, i.e. the model itself in mathematical terms, to create the local model as per step 4.4.5 above, one local model is produced for each observation in the data set and then the model is applied to the observation to get the predicted value. This is based on an assumption that removing any single observation from the local model produced in the previous step does not violate any OLS Regression assumptions. This also encourages care and caution when checking outliers in the model diagnostics processes.

By following the process just described, a set of fitted values can be calculated. As these fitted values are the predicted values of a possibly transformed response variable ('Summary Work Effort'), a back transformation using the negative of the transformation function may need to be performed to get the predicted summary work effort in its original scale.

With the above issues dealt with, predictions can be made using the global model by applying it to the local data set. At the end of this step, for each organization, two sets of predicted 'Summary Work Effort' values are produced, one from the local and another from the global model.

### 4.5.2 Evaluation Criteria

The evaluation of the software effort estimation models is performed using the following commonly used indicators:

- *Mean Magnitude of Relative Error (MMRE):* The Magnitude of Relative Error (MRE) is defined as:

$$MRE = \left| \frac{Effort_{Actual} - Effort_{\Pr edicted}}{Effort_{\Pr edicted}} \right|$$

The MRE is calculated for each observation in the data set that is used to produce the local predictions. MMRE takes the mean of all the MREs for a given data set. A MMRE of 0.20 means that the predicted summary work effort values are, on average, within $\pm 20\%$ of the actual summary work effort values. In general, the smaller the MMRE value, the better the performance of the prediction power of the model.

- *Median Magnitude of Relative Error (Median MRE):* Similar to MMRE, with the only difference being that the median MRE is used instead of the mean. Median MRE is less sensitive to extreme values or outliers. The smaller the Median MRE value, the better the performance of the model.
- *PRED(0.25):* Pred is also very often used in similar empirical software engineering literature. This measure gives the proportion of predictions at a given level of accuracy. It is defined as:

$$\Pr ed(l) = \frac{k}{N}$$

In this formula, *N* is the total number of observations in the data set, and *k* is the number of observations with an MRE that is less than or equal to *l*. The majority of relevant past research uses an *l* value of 0.25, which is therefore adopted in this research. A value of PRED(0.25) = 0.5 means that 50% of the observations' predictions are within 25% of their actual value. The larger the PRED(0.25) value, the better the performance of the model.

- *Sum of Absolute Error:* The Sum of the Absolute Error is the sum of the difference between the actual summary work effort and the estimated summary work effort of each specific observation. The smaller the Sum of the Absolute Error, the better the performance of the model. It is defined as:

$$SumOfAbsoluteError = \sum \left| Effort_{Actual} - Effort_{Estimated} \right|$$

### 4.5.3 Local and Global Model Comparison Methodology

For each of the eight organizations under consideration at this point, two sets of evaluation indicators are produced: one for its local model and another for its global model. They are compared using the following statistical methods.

- *MMRE comparison:* The two sample paired t-test is used to compare the measures of $MMRE_{Local}$ and $MMRE_{Global}$ across the sample of organizations. As with other statistical procedures executed in this research, the assumptions are first checked against the two set of results, and if the assumptions are not violated a paired sample t-test is performed. Otherwise, its equivalent non-parametric paired sample t-test is used, in this case, paired samples Wilcoxon test. The reason behind the use of the paired sample t-test is that for each organization there is always one $MMRE_{Local}$ and $MMRE_{Global}$; that is, they are not independent. The scenario here is to perform one two-sided two sample paired t-test/Wilcoxon test to assess *whether there is a difference between* the MMREs gained from the local models and those from the global models. After that, a one-sided two sample paired t-test is performed to answer the question: are the MMREs obtained from the local models *better (smaller) than* those resulting from the global models?
- *Median MRE comparison:* Similar to MMRE, the two sample paired t-test/Wilcoxon test is employed with the same scenarios.
- *PRED(0.25) comparison:* Similar to MMRE, the two sample paired t-test/Wilcoxon test is employed with the same scenarios. The only difference in this case is that the one-sided two sample paired t-test is to be performed to answer the question: are the PRED(0.25) values obtained from the local models *better (larger) than* those obtained from the global model?
- *Sum of Absolute Error comparison:* Similar to MMRE, the two sample paired t-test/Wilcoxon est is employed with the same scenarios. The two sample paired t-test/Wilcoxon test is conducted to evaluate the difference between the Sum of Absolute Errors from the local models and those from the global models.

### 4.5.4 Local vs. Global Model Data Analysis Results

In an effort to utilise as much of the ISBSG repository information as possible, and to give the global models substantial ‘opportunity’ to outperform their local counterparts, the approach taken here has been to allow the development of larger models than might otherwise be feasible or desirable in practice. Some of the OLS Regression models generated in this research are therefore very complex because of the large number of variables included and their interactions. It was not the purpose of this research to establish or recommend these models *for practice,* as they are naturally of limited applicability due to their being derived in relation to data associated with ten anonymous organizations. Rather, they are constructed solely to facilitate the addressing of the research objectives. The detailed OLS Regression models for each organization, including both local and global models, as well as the detailed R source code and output are therefore not shown in the body of this thesis. However, these details are attached separately along with other deliverables for reference and evaluation (please see Appendix 2).

During the process of establishing models for each of the eight selected organizations it was found that it was not possible to form a valid local OLS Regression model for organization E, because of the existence of influential outliers and its relatively small size of thirteen observations. However, organization E is still included in the model comparisons that follow. This decision is made to fully utilise the data set with the acknowledgement that the assumptions of the underlying model might be violated and hence the model would not be statistically valid. In addition, this situation is exactly that which a local software organization might find itself in – not having sufficient data of its own but anticipating that a model built from a global repository might apply to them. Hence the global model result for organization E could be informative in terms of indicating the utility of a repository in terms of project effort estimation. Note that all of the organization E observations (as valid entries in the repository) were retained in the building of global models for the other organizations.

Table 15 lists the details of the performance of the models according to the evaluation criteria defined in section 4.5.2.

|  | MMRE |  | Median MRE |  | PRED(0.25) |  | Sum of absolute errors |  |
|---|---|---|---|---|---|---|---|---|
| Orgn | Local | Global | Local | Global | Local | Global | Local | Global |
| B | 0.398 | 0.568 | 0.308 | 0.508 | 34.2% | 20.0% | 166458.2 | 264792.2 |
| C | 0.388 | 0.828 | 0.247 | 0.553 | 50.6% | 21.1% | 162781.1 | 214755.7 |
| D | 0.377 | 0.718 | 0.164 | 0.809 | 55.6% | 0.0% | 25650.6 | 117718.3 |
| E | 1.622 | 25.062 | 0.703 | 6.804 | 9.1% | 8.3% | 17076.6 | 61400.5 |
| F | 0.506 | 0.697 | 0.165 | 0.634 | 55.0% | 5.0% | 23130.8 | 50151.7 |
| G | 0.416 | 0.546 | 0.311 | 0.563 | 40.0% | 20.0% | 214394 | 306603.2 |
| H | 0.753 | 1.829 | 0.481 | 0.880 | 34.0% | 10.0% | 94878.6 | 128608.6 |
| J | 0.714 | 1.029 | 0.497 | 0.520 | 21.7% | 23.3% | 151275.5 | 203330.1 |

**Table 15: Model performance (local and global) according to the four criteria**

The following plots (Figures 6 to 9) graphically illustrate the differences in performance between local and global models in terms of MMRE, Median MRE, PRED(0.25) and Sum of Absolute Errors respectively.

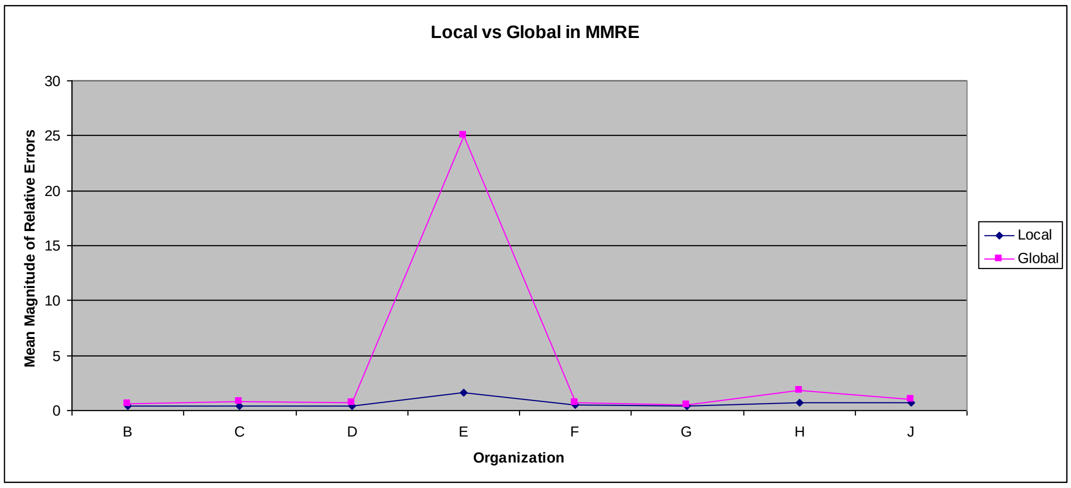


**Figure 6: Local and global model performance in MMRE**

As can be seen in the values reported in Table 15 and visually in Figures 6 to 9, local models are superior to global models according to the evaluation criteria MMRE, MedianMRE, PRED(0.25) and Sum of Absolute Errors for all eight organizations, with one exception – the PRED(0.25) value obtained from the global model for organization J is higher than its local equivalent.

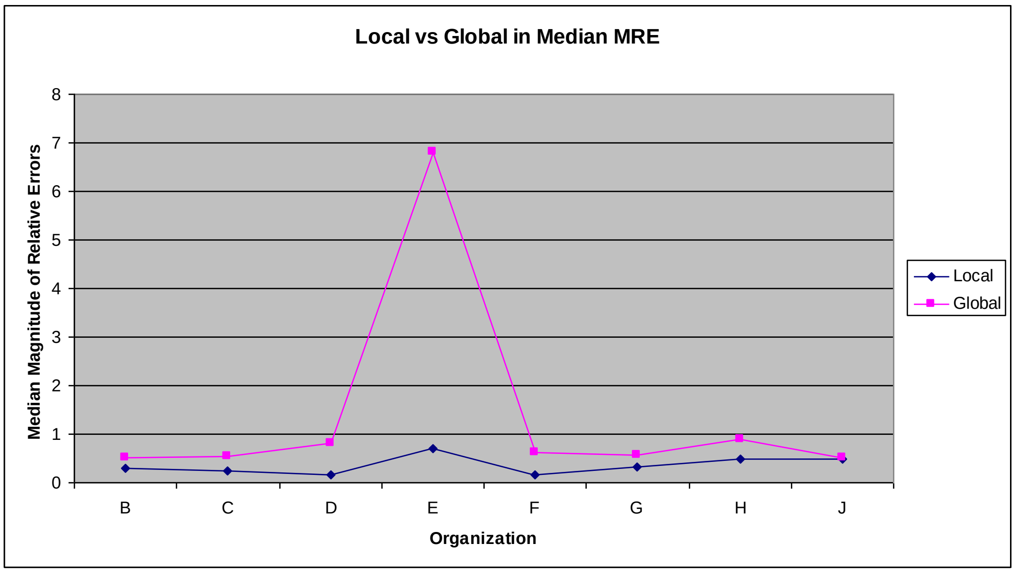


**Figure 7: Local and global model performance in MedianMRE**

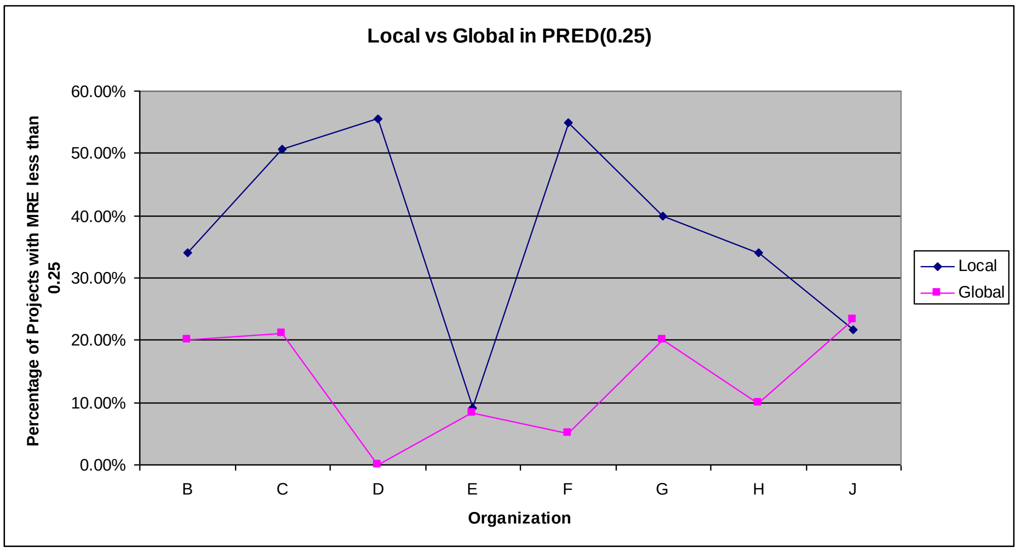


**Figure 8: Local and global model performance in PRED(0.25)**

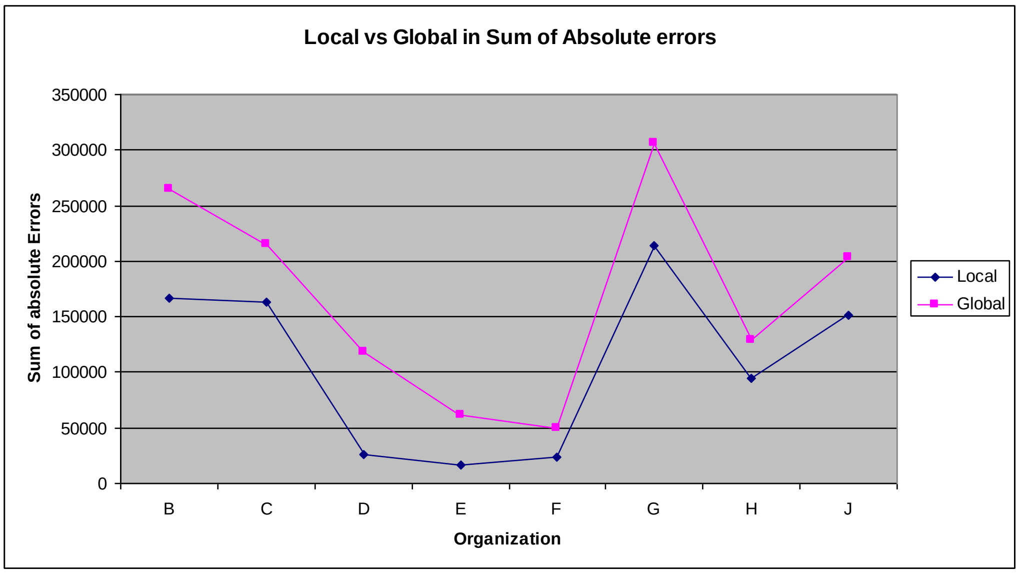


**Figure 9: Local and global model performance in Sum of Absolute Errors**

### 4.5.5 Formal Comparison of Local and Global Models

The local and global models are now compared using either a two-sided two sample paired t-test or a two-sided paired sample Wilcoxon test, depending on whether the data violate the assumptions underlying the t-test. Note that the two sample t-test is used if at all possible because it provides an indication of the effect and its *size* i.e. the difference in performance between the local and global models. To reiterate, the tests are 'paired' because the two samples are not independent: an organization's local model performance is directly related to its global model performance. The tests are 'two-sided' in the first instance because the initial objective is to identify 'whether there is a difference' instead of 'whether the local model is better than the global model'. Table 16 reports the statistical comparisons between local and global models using two-sample paired t-tests, or their equivalent.

| **Criterion** | **p-value** | **95% confidence interval lower bound** | **95% confidence interval upper bound** |
|---|---:|---:|---:|
| MMRE | 0.0078 | N/A | N/A |
| Median MRE | 0.0078 | N/A | N/A |
| PRED(0.25) | 0.0135 | 0.0670 | 0.4137 |
| Sum of Absolute Errors (person hours) | 0.0004 | -85256.0900 | -37672.6300 |

**Table 16: Statistical analysis of the Local vs. Global model comparison**[1]

The formal statistical interpretations in relation to these comparisons are as follows:

- *MMRE:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the MMRE values obtained from models created from a local data set and a global data set.
- *Median MRE:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the Median MRE values obtained from models created from a local data set and a global data set.

[1] The confidence intervals of the MMRE and Median MRE comparisons are not applicable because the t-test could not be applied due to the violation of the assumption of normality required by the t-test. As a consequence, the assumption-free non-parametric Wilcoxon test was used which can only tell whether or not the difference is statistically significant.

- *PRED(0.25):* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the PRED(0.25) values obtained from models created from a local data set and a global data set. With 95 percent confidence, it is asserted here that the average difference in PRED(0.25) between a local model and a global model is somewhere between 6.7% and 41.37%.
- *Sum of Absolute Error:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the sum of absolute error values obtained from models created from a local data set and a global data set. With 95 percent confidence, it is asserted here that the average difference in absolute error values between a local model and a global model is somewhere between -37672.63 and -85256.09.

A set of one-sided tests was conducted to refine the comparison results, as shown in Table 17.

| **Criterion** | **Expected direction for local set** | **p-value** | **95% confidence interval lower bound** | **95% confidence interval upper bound** |
|---|---|---|---|---|
| MMRE | Less than | 0.0039 | N/A | N/A |
| Median MRE | Less than | 0.0039 | N/A | N/A |
| PRED(0.25) | Greater than | 0.0068 | 0.1015 | Positive Infinity |
| Sum of Absolute Errors (person hours) | Less than | 0.0002 | Negative Infinity | -42402.0000 |

**Table 17: Statistical analysis of the Local vs. Global model comparison in one-sided tests**

The formal statistical interpretations in relation to these comparisons are as follows:

- *MMRE:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the MMRE values obtained from models created from a local data set and a global data set, and the difference between local data and global data in MMRE is negative, i.e. local MMRE is significantly less than global MMRE.

- *Median MRE:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the Median MRE values obtained from models created from a local data set and a global data set, and the difference between local data and global data in MedianMRE is negative, i.e. local MedianMRE is significantly less than global MedianMRE.
- *PRED(0.25):* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the PRED(0.25) values obtained from models created from a local data set and a global data set, and the difference in PRED(0.25) is positive, i.e. local PRED(0.25) is generally larger than global PRED(0.25). With 95 percent confidence, it is asserted here that the average difference in PRED(0.25) between a local model and a global model is at least 10.2%.
- *Sum of Absolute Error:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the sum of absolute error values obtained from models created from a local data set and a global data set, and the difference between the local data set and global data set in sum of absolute errors is negative, i.e. local sum of absolute errors is generally smaller than that of global data set. With 95 percent confidence, it is asserted here that the average difference in absolute error values between a local model and a global model is somewhere between negative infinity and -42402.

## 4.6 Step 6: Local vs. Refined Global Model Comparisons

The local models have proved to be significantly better than the global models in the previous comparisons. An expectation that more targeted (refined) global models might be more effective is reflected in the following question: "With respect to the construction and validation of accurate prediction models for software development effort, is it possible to find and utilise relevant non-local data?"

A variety of refinement strategies could be adopted in order to address this question, utilising information in the repository relating to organization type, language type, data quality and so on. A single strategy has been investigated here, in an effort to ensure that only observations that match the characteristics of each local organization's data set are considered in its associated global data set.

This requires a revisiting of the data screening process. When the local and global data sets were being determined from the formalized dataset, two of the rules applied to the construction of each local data set were as follows:

- Drop variables that have more than 90% identical values OR variables with only two levels but the number of observations in a level is less than 3.
- Drop observations that belong to a variable that has more than 2 levels and the number of values for any given level is less than 5, or 5% of the total number of observations; however, if the number of observations is larger than 20, then this variable will is still kept.

In following these rules, some of the variables were dropped from one or more of the local data sets as they were irrelevant when establishing the local models. However, in the corresponding global models, these variables may still have been of importance and so were utilized in the global models. In order to bring the global models one step closer to their corresponding local models these two rules are utilised as constraints to inform the development of 'refined' global models, in the hope that these global model variants will prove to be more accurate than their whole-set predecessors.

This may be best illustrated by an example. For organization C, the variable 'Business Area Type' was dropped from the model-building process because all of the values in this variable were "IT&T". This indicates that organization C's business focus is the IT&T industry. As a consequence, organization C's corresponding global data set could be refined so that only observations from projects with 'Business Area Type' equal to IT&T are included.

Since every step in the data screening process was carefully logged and documented, it is a straightforward matter to find out the areas in which these two rules were applied, and ensure that they are enforced where relevant for an organization's refined global data set. By following this procedure, a set of refined global data sets were established for seven of the eight organizations. An estimation model of this nature could not be established in the case of organization D because no model could be found to be statistically significant. Therefore, organization D is excluded from the following analysis.

### 4.6.1 Local vs. Refined Global Model Data Analysis Results

The same modelling process as used in the construction of the original global models is undertaken here, but with the underlying global data sets having been reduced to match the characteristics of their associated local data sets. The files that contain the details of each organization's OLS Regression models (including both local and refined global models), as well as the detailed R source code and outputs, can be found in Appendix 2. Unlike the original global models, in which every single organization's global model was very similar, the refined global models vary in terms of the numbers of variables included. Please see Appendix 2 for details regarding each of the models.

It should be noted that, as above, it was found to be impossible to construct a sound model for organization E's refined global data set, because of the violation of underlying assumptions. However, in the interests of testing the capabilities of such a repository in a realistic scenario, the data is presented here, while acknowledging that the appropriateness of the models for organization E is open to question in terms of their statistical validity.

Table 18 lists the details of the performance of the models according to the evaluation criteria defined in section 4.5.2.

The following plots (Figures 10-13) graphically illustrate the differences in performance between local, global and refined global models in terms of MMRE, Median MRE, PRED(0.25) and Sum of Absolute Errors respectively.

|  | MMRE |  | Median MRE |  | PRED(0.25) |  | Sum of absolute errors |  |
|---|---|---|---|---|---|---|---|---|
| Orgn | Local | Refined Global | Local | Refined Global | Local | Refined Global | Local | Refined Global |
| B | 0.398 | 1.473 | 0.308 | 0.746 | 34.2% | 7.3% | 166458.2 | 391571.3 |
| C | 0.388 | 1.547 | 0.247 | 0.772 | 50.6% | 16.5% | 162781.1 | 340156.5 |
| D | 0.377 |  | 0.164 |  | 55.6% |  | 25650.6 |  |
| E | 1.622 | 87.743 | 0.703 | 14.789 | 9.1% | 0.0% | 17076.6 | 239066.4 |
| F | 0.506 | 0.926 | 0.165 | 0.601 | 55.0% | 25.0% | 23130.8 | 57096.74 |
| G | 0.416 | 0.635 | 0.311 | 0.619 | 40.0% | 18.8% | 214394.0 | 399028.8 |
| H | 0.752 | 1.552 | 0.481 | 0.715 | 34.0% | 10.0% | 94878.61 | 138666.4 |
| J | 0.714 | 19.29 | 0.497 | 3.209 | 21.7% | 8.3% | 151275.5 | 1513421.0 |

**Table 18: Model performance (local and refined global) according to the four criteria**

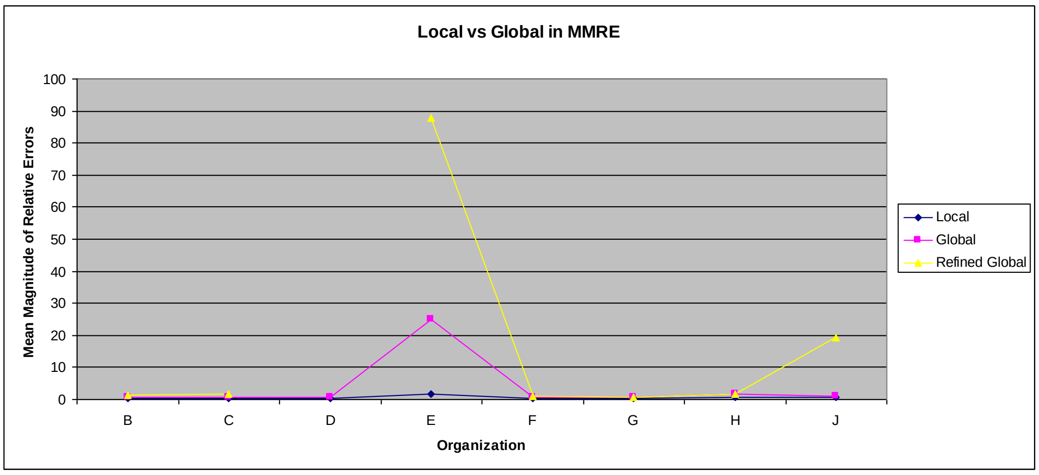


**Figure 10: Local, global and refined global model performance in MMRE**

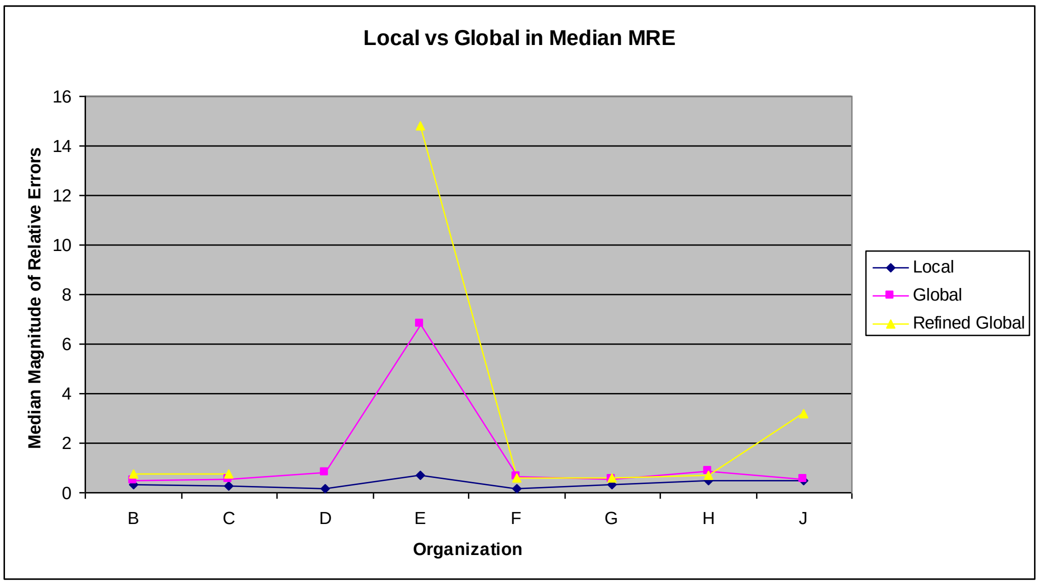


**Figure 11: Local, global and refined global model performance in MedianMRE**

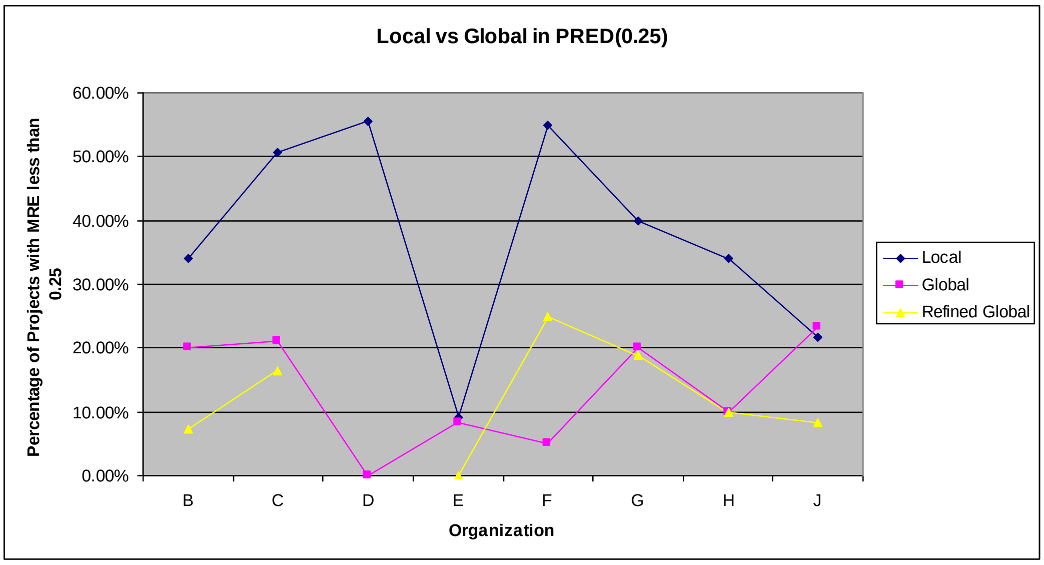


**Figure 12: Local, global and refined global model performance in PRED(0.25)**

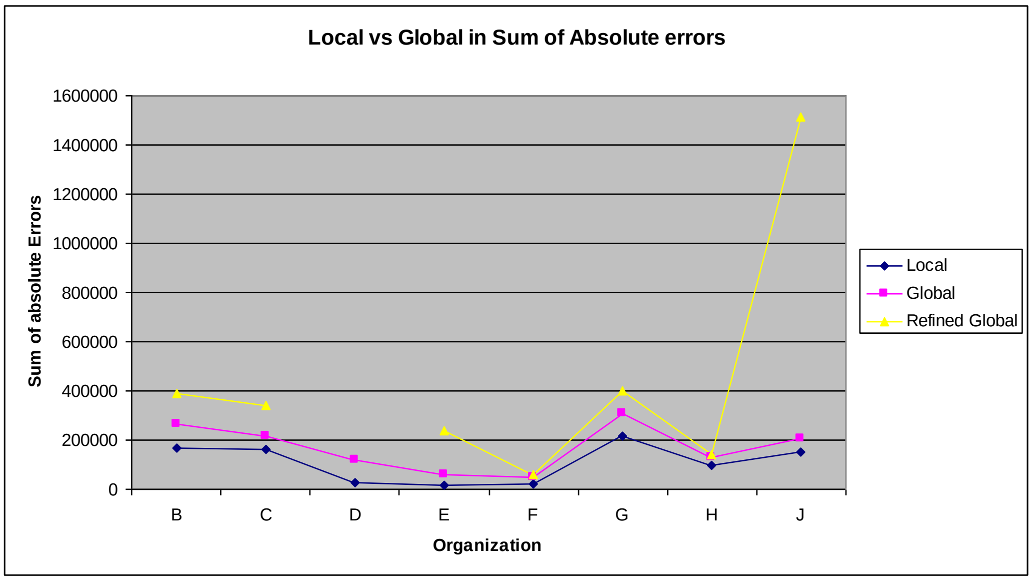


**Figure 13: Local, global and refined global model performance in Sum of Absolute Errors**

As can be observed in Table 18 and visually from Figures 10 to 13, it appears that local models are also superior to refined global models across all of the MMRE, MedianMRE, PRED(0.25) and Sum of Absolute Errors evaluation criteria for all seven organizations (not including D for refined global as no model could be found).

### 4.6.2 Formal Comparison of Local and Refined Global Models

The local and refined global models are now compared using the same methodology as employed in the previous comparison of local and global models.

Table 19 reports the two-sided statistical comparisons between local and refined global models using two-sample paired t-tests (or their equivalent subject to the validity of underlying assumptions).

| **Criterion** | **p-value** | **95% confidence interval lower bound** | **95% confidence interval upper bound** |
|---|---|---|---|
| MMRE | 0.0156 | N/A | N/A |
| Median MRE | 0.0156 | N/A | N/A |
| PRED(0.25) | 0.0005 | 0.1441 | 0.3090 |
| Sum of Absolute Errors (person hours) | 0.0156 | N/A | N/A |

**Table 19: Statistical analysis of the Local vs. Refined Global model comparison**

The formal statistical interpretations in relation to these comparisons are as follows:

- *MMRE:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the MMRE values obtained from models created from a local data set and a refined global data set.
- *Median MRE:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the Median MRE values obtained from models created from a local data set and a refined global data set.
- *PRED(0.25):* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the PRED(0.25) values obtained from models created from a local data set and a refined global data set. With 95 percent confidence, it is asserted here that the average difference in PRED(0.25) between a local model and a refined global model is somewhere between 14.0% and 30.9%.
- *Sum of Absolute Error:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the sum of absolute error values obtained from models created from a local data set and a refined global data set.

A set of one-sided tests was also conducted to refine the comparison results, as shown in Table 20.

| **Criterion** | **Expected direction for local set** | **p-value** | **95% confidence interval lower bound** | **95% confidence interval upper bound** |
|---|---|---|---|---|
| MMRE | Less than | 0.0078 | N/A | N/A |
| Median MRE | Less than | 0.0078 | N/A | N/A |
| PRED(0.25) | Greater than | 0.0003 | 0.1611 | Positive Infinity |
| Sum of Absolute Errors (person hours) | Less than | 0.0078 | N/A | N/A |

**Table 20: Statistical analysis of the Local vs. Refined Global model comparison in one-sided tests**

The formal statistical interpretations in relation to these comparisons are as follows:

- *MMRE:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the MMRE values obtained from models created from a local data set and a refined global data set and the difference is negative, i.e. MMRE values from local models are significantly less than those of refined global models.
- *Median MRE:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the Median MRE values obtained from models created from a local data set and a refined global data set, and the difference is negative, i.e. MedianMRE values from local models are significantly less than those of refined global models.
- *PRED(0.25):* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the PRED(0.25) values obtained from models created from a local data set and a refined global data set, and the difference is positive, i.e. PRED(0.25)values from local model are significantly larger than those of refined global models. With 95 percent confidence, it is asserted here that the average difference in PRED(0.25) between a local model and a refined global model is at least 16.1%.

- *Sum of absolute errors:* There is very strong evidence that, in terms of software effort prediction model performance for the organizations considered here, there is a significant difference in the sum of absolute error values obtained from models created from a local data set and a refined global data set, and the difference is negative, i.e. the sum of absolute error values from local models are significantly less than those of refined global models.

In considering the comparative accuracy of both local vs. global and local vs. refined global models, not only in two-sided comparisons but also in the more testing one-sided counterparts, the results clearly indicate that there is a difference in performance between models built using an organization's local data set and those built using global data. In saying that, there are preconditions and limitations that apply to this conclusion, as well as future opportunities to extend the work just reported. This is addressed in the next chapter.

# 5 Summary, Conclusions and Recommendations

## *5.1 Summary of the Research*

This research set out to investigate the value and validity of building predictive models from software engineering data repositories that consisted of records from multiple organizations as opposed to one single organization. To that end, extensive analysis of multiple data sets extracted from the ISBSG repository was undertaken. OLS Regression was chosen as the main methodology among a list of alternative techniques commonly used in research in this domain.

During data analysis the candidate data set was extensively pre-processed in order to retain as much of the original information as possible while at the same time satisfying the assumptions of OLS Regression. The original 'raw' ISBSG data set was first formalized in order to produce a data set of reasonable size and content. The formalized data set was then further refined to make it as an appropriate target for OLS Regression. After that, the formalized data set was grouped according to the given organizations' anonymous identities to enable the local and global model comparisons. A systematic OLS Regression process was then designed to streamline the analysis process for each pair of data sets in a manner that ensured visibility and repeatability for future development and validation. The application of this process to each data set pair produced a set of models that could be compared in terms of estimation performance, using the leave-one-out validation methodology and standard evaluation criteria.

The conclusions of the study are now presented, followed by a discussion of the implications of those conclusions for both practice and future research.

## *5.2 Conclusions and Limitations of the Study*

In terms of the overall intent of the research it can be concluded that, in general, it is possible to build valid models for local organizations using the ISBSG repository, although in one case it was not possible to construct a valid refined global model. The question as to whether these models are valuable is less straightforward to answer, however.

In the past a commonly cited target for model performance has been to achieve MMRE values of 0.25 or less. Performance on data sets drawn from the ISBSG repository, however, has tended to be a long way short of this level (for instance, see the results reported by Jeffery *et al.* (2000)). The models built in the course of this research showed typically varied performance, with MMRE values ranging from 0.388 to 0.753 for local models, leaving the invalid model for organization E out of consideration for the purposes of this discussion. The global counterparts fared far worse, however, with MMRE values ranging from 0.568 to 1.829. This pattern held generally across all data sets and all assessment criteria. Whether the underlying models can be considered as valuable is a question only the project managers of the eight organizations could answer – some may consider the extent of error still too high; others may consider that the capability to predict effort for more than half of their projects to within 25%, as was indicated by the models produced above, to be an improvement on their current efforts.

Shifting attention to the comparative analysis, the following statement can be made. The reliability of the outcomes of this study, in terms of the comprehensive nature of the data considered and the robustness of the pre-processing and analyses undertaken, leads to a high level of confidence in the overall conclusion that, **when available, organization-specific (local) data are more useful than cross-organization (global) data in terms of software project effort estimation.**

The specific research questions regarding software project effort estimation are now revisited. Of central interest was the question of the amount of data required to enable appropriate and accurate models to be developed:

a. With respect to the construction and validation of accurate prediction models for software development effort, how do small homogeneous (local) data sets compare with larger multi-organization (global) data sets?
b. With respect to the construction and validation of accurate prediction models for software development effort, is it possible to find and utilise relevant non-local data?

### 5.2.1 Local vs. Global Data Sets

It is stated here with a high degree of confidence that, for the repository considered in this study, local data sets are more useful than global data sets in terms of enabling the development of valid, accurate and potentially useful models.

The rationale for this conclusion when considered against prior research is:

- The results have been obtained through the analysis of data from several organizations as opposed to from a single organization.
- The results are almost entirely consistent across these analyses – in just one test for one organization against one criterion was a global (or refined global) model found to be more accurate than its local counterpart.
- The data sets have been extensively filtered, massaged and polished, with a detailed audit trail in order to ensure transparency and repeatability, in order to be used in OLS Regression analysis. This might be considered by some as a disadvantage or limitation. Such a view is rejected here, however. It is instead contended that analysis outcomes can be optimized through the application of domain knowledge as opposed to the possibly uninformed reliance on automated methods, reflecting Henderson and Velleman's (1981) contention that "data analysts know more than the computer".
- The missing data in the ISBSG repository have been carefully examined, handled and documented to give transparency, repeatability and extensibility to future research in this area. This is in contrast to some other research in which the missing data are simply discarded.

There are also a number of conditions to the conclusion that local data is more useful than global data in software project effort estimation:

- The conclusion is established on the basis of the outcomes of OLS Regression, generally held to be the most commonly used methodology in the empirical software engineering research context. However, there are a number of alternative analysis methodologies that could be used in its stead, and that might result in different outcomes. Some of these alternative methodologies are mentioned in the next section which deals with future implications of this work.

- The OLS Regression models were created using the most commonly employed procedures, parameter selection methods and tools. Similarly, the model comparisons were undertaken using standard approaches and thresholds for significance. Some of the decisions were made because of constraints on available resources. For example, ‘All Possible Regression’ was not chosen as the variable selection technique because of its computational demands. Alternative decisions at any of these points may have led to different research outcomes.
- The conclusion is limited to the context of the analyzed data, in this case, the ISBSG repository version 9.0 and the data of eight companies within that repository. A description of the ISBSG data set can be found in ISBSG (2006c). The specific context of the ISBSG 9.0 data could present some bias compared with the global software engineering context. For example, response bias could be generated when an organization fills in the questionnaire as requested by the ISBSG. Non-response bias may arise by the omission of failed projects from the repository. While these factors are out of this study’s control it is acknowledged that they may potentially confound the research outcomes.
- This research is a retrospective analysis as opposed to an active experiment. Therefore, the conclusions drawn from this research should be interpreted for reference as opposed to indicating formally defined causations.
- A certain level of subjectivity is evident in the processing of the raw ISBSG data set, which could raise disagreement from other researchers. However, as all the rationales and data processing rules are documented, they are at least visible and can easily be identified, modified or extended in the future.
- It is unknown whether the accuracy of any of the models, local or global, world be considered effective and/or acceptable in software project management practice. While the accuracy results are better than those achieved in many other studies, particularly studies that have used the ISBSG repository, it is not clear whether these levels would be sufficient to encourage use.

**It is therefore concluded that small homogeneous (local) data sets should be preferred over larger multi-organization data sets in the context of building effort prediction models.**

### 5.2.2 Local vs. Refined Global Data Sets

It is also stated here with a high degree of confidence that, for the repository considered in this study, local data sets are more useful than refined global data sets in terms of enabling the development of accurate and potentially useful models. The rationale for such a conclusion is as for the comparison of local and global models just described. The same constraints also apply, along with the following additional limitations:

- just one refinement to the global data sets was tested, in terms of considering only observations with similar categorical characteristics.
- the comparison of global and refined global models was not undertaken. This would form an interesting future development area.

That said, it does appear that it is possible to find and utilise relevant non-local data in this context. Whether these might be considered usable or useful in practice remains an open question.

## *5.3 Implications, Recommendations and Future Work*

A range of outcomes have resulted from this research that lend themselves to recommendations for practice and for future investigation.

### 5.3.1 Alternative Modeling Methods

The research reported in this thesis employed OLS Regression as the modeling method, based on its widespread prior adoption and its relatively widespread accessibility. A range of other modelling techniques could be used, however, instead of regression or in addition to it. Apart from those already being more commonly used in relevant research (case-based reasoning, genetic algorithms and neural networks (Mair and Shepperd (2005)), the following techniques are suggested for future extension of this and related research.

- Bayesian Methods: Based on Bayesian theory, this approach employs a variation of commonly used probability theory. It analyses past uncertain situations and then determines the probability that a certain event caused the outcome. For more detail, refer to Howson and Urbach (2002). Some consideration of its use in relation to software engineering can be found in Chulani *et al.* (1999).
- Default Logic: Knowledge-based systems could be established to more thoroughly underpin software project estimation in terms of an organization's own context and the global context. See Antoniou (1999) for more details.
- Dempster-Shafer Theory of Evidence (Shafer 2002): This approach determines the weight of previous evidence and assigns degrees of belief to statements based on that weighting.
- Qualitative Reasoning (Forbus, 1996): Another commonsense-based method of deep reasoning about uncertain situations.
- Fuzzy Logic: An approach commonly used in control systems, fuzzy logic systems allows the use of linguistic rather than numeric measures and estimation based on inference from fuzzy sets membership (MacDonell and Gray, 2003).

It has also been acknowledged that this research employed just a single data repository, that provided by the ISBSG. Other global software engineering data sets such as the 'the Finnish data set' could be used to revisit the conclusions from this research. Such an investigation would help to eliminate the potential biases exposed from the use of the ISBSG data set.

### 5.3.2 Further Analysis of Local vs. Refined Global Models

There is significant scope for further comparisons between local, global and refined global data sets given that several other refinement options are possible. For example, the global data set may be made more relevant by filtering the observations on the basis of time (discarding 'old' projects), quality (including only projects rated 'A' by the ISBSG), a moving window (predicting the next project on the basis of recent ones), or any combination of these strategies.

### 5.3.3 Optimising the OLS Regression Analysis Process

One of the major lessons learned from the literature review conducted for this research, and one that should inform further research, is the need for great care in the handling of data. It is a basic premise of effective empirical software engineering that the quality of the research outcomes depends heavily on the data set(s) being used. Furthermore, in the context of software engineering research, it is often impractical and/or expensive for organizations to compile high-quality data sets. As a consequence, it is imperative that the available information from existing data be fully utilized. In this research, more effort was expended on the effective management of the data, such as formalization, data set further refinement and grouping, than on the data analysis itself. Missing data imputation techniques were considered and used in this research to maximise the retention of observations and variables. Software engineering domain-specific knowledge was used in preference to relying purely on statistical tools to process the data at steps such as data formalization, further refinement, grouping and data screening. This admittedly introduces a certain level of arbitrariness into the data retained. However, it is contended here that this is preferable to discarding data in a similarly arbitrary fashion as has occurred in other studies. Indirect validation of this strategy may be drawn from the higher levels of accuracy achieved in this study for the models developed as compared to those reported in prior studies.

In this research, the OLS Regression analysis process was streamlined and documented. An audit trail was created addressing:

- The alternative solutions at each step, their advantages and disadvantages.
- The rationale behind which choice was made.
- The ‘rules’ of the execution of each step.
- The transactions that occurred during the execution of each step.

Such an approach produces research with a high level of transparency and enhances the repeatability of the work. This should ensure that any potential problems can be identified and/or improvements can be made.

During the OLS Regression analysis process of establishing models, the underlying assumptions of OLS Regression analysis were carefully checked in order to ensure as much as possible that only reliable and defendable conclusions were derived. If any of the assumptions could not be satisfied, an alternative solution was sought or the inclusion and/or exclusion of data or variables was reconsidered in relation to the determination of the regression models. For example, when establishing a valid OLS Regression model, the residuals of the model should be linear, the scatter of the residuals should be constant and the residuals should be independent of one another. For every model that has been established here, such assumption verification has been performed and documented to facilitate future experimental examination, replication and enhancement.

### 5.3.4 Implications – Collection and Analysis of SE Data

As indicated by Kitchenham *et al.* (2006a), the process of software development effort estimation is based on data gathered from the actual software development process. From a business information systems point of view, all the data which reflect the actual activities may be considered as transactions. In business information systems, transactions are processed on at least a daily basis to record the history of activities. At a higher level, managers or researchers use transaction data to perform certain forms of analysis (for example forecasting, estimation) using Extraction, Transformation and Loading (ETL) techniques common in the data warehousing domain.

It is contended here that the same thinking could be applied in the software engineering industry. Rather than using different data from different sources to make predictions based on empirical observation, there is a need to systematically analyse the software engineering domain to establish a set of transaction models that accurately reflect software engineering activities in practice. This could help to address the difficulties encountered in effective data collection, as illustrated by Kitchenham *et al.* (2006a, page 5): "*In the IBM case, lack of an overall design resulted in the data being stored in several discrete databases with links between the data items not being maintained.*".

It seems that the dominant approach to the management of software engineering data is to store and manipulate the data as a separate set of relational tables as opposed to objects with complicated relationships that are embedded in the normal operations of the business. This separation (perhaps within a Quality Assurance function) divorces the management of software development from the rest of the business process. Furthermore, it means that much of the analysis is carried out *post hoc*, leading to delays in the development of insights. Gaining a greater understanding of the dynamics of software development may be more likely if the software development process models are integrated with the business models. This would better enable the systematic domain analysis that is needed at a higher level. Once such a model is built, just like all the other business models being applied in a large number of ERP systems, transactions relating to software development activities can be gathered and future analysis more reliably established.

### 5.3.5 Implications – Empirical Software Engineering Research

In the interests of visibility, validity and reliability of this research the decision was taken to document every decision and its rationale, alternative solutions considered, the ‘rules’ selected and the transactions executed *as they were processed*. While such an approach adds a degree of overhead to the research it does enhance the reliability of the research outcomes, through enabling the recording of decisions as they occur rather than in retrospect. It also has two potentially beneficial side-effects. Given that research of the type reported here can involve multi-organization and even multi-country data the recording of the information just described would enable dispersed teams of researchers to work together more effectively than might otherwise be the case. Furthermore, it represents a step towards evidence-based software engineering, an approach that has gained support in recent years as a means of helping the discipline to more effectively learn lessons and improve practices. Such an approach relies extensively on systematic reviews and meta-analyses – these can only be performed effectively if ‘raw’ information like that described above is recorded. With this in mind it would seem advantageous for future research addressing this and similar questions to adopt a similar strategy.

## References

G. Antoniou (1999) A tutorial on default logics. *ACM Computing Surveys*, 31(4): 337-359.

B.W. Boehm (1981) *Software Engineering Economics*. Englewood Cliffs, N.J.: Prentice-Hall.

B. Boehm, C. Abts and S. Chulani (2000) Software development cost estimation approaches—a survey. *Annals of Software Engineering* 10: 177-205.

L. Briand, K. El Emam, D. Surmann, I. Wieczorek and K. Maxwell (1999) "An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques", *Proceedings of the 21st International Conference on Software Engineering*, pp313-322.

L. Briand, T. Langley and I. Wieczorek (2000) "A Replicated Assessment of Common Software Cost Estimation Techniques", *Proceedings of the 22nd International Conference on Software Engineering*, pp377-386.

K.P. Burnham and D.R. Anderson (2004) "Multi-model Inference: Understanding AIC and BIC in Model Selection", *Proceedings of the Amsterdam Workshop on Model Selection* http://www2.fmg.uva.nl/modelselection/presentations/AWMS2004-Burnham-paper.pdf

S. Chulani, B. Boehm and B. Steece (1999) Bayesian Analysis for Empirical Software Engineering Cost Models. *IEEE Transactions on Software Engineering* 25(4): 573-583.

S. Derksen and H. J. Keselman (1992) Backward, forward and stepwise automated subset selection algorithms: frequency of obtaining authentic and noise variables. *British Journal of Mathematical and Statistical Psychology* 45: 265–282.

B.S. Everitt and G. Dunn (2001) *Applied Multivariate Data Analysis* (Second Edition, Appendix B).

K.D. Forbus (1996) *Qualitative Reasoning*.
http://www.qrg.northwestern.edu/papers/Files/crc7.pdf

J. Hair, R. Anderson, R. Tatham and W. Black (1998) *Multivariate Data Analysis* (Fifth Edition). Prentice Hall.

H.V. Henderson and P.E. Velleman (1981) Building Multiple Regression Models Interactively. *Biometrics* 37:391-411.

F.J. Heemstra (1990) "Software cost estimation models", *Proceedings of the Jerusalem Conference on Information Technology*, pp 286-297.

C. Howson and P. Urbach (1993) *Scientific Reasoning: The Bayesian Approach* (Second Edition). Open Court Publishing.

ISBSG (2006a) *Glossary of Terms, V5.9.1, 28/02/06*
http://www.isbsg.org/html/Glossary_of_Terms.doc

ISBSG (2006b) *The ISBSG Repository (R9) Demographics*
http://www.isbsg.org/html/Data%20CD%20R9%20demographics.doc

ISBSG (2006c) *ISBSG Repository (R9) Field Descriptions*
http://www.isbsg.org/html/R9%20Field%20Descriptions.doc

ISBSG (2006d) *ISBSG Project Submission Process*
http://www.isbsg.org/html/Project%20Submission%20Process.doc

R. Jeffery, 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* 42: 1009-1016.

R. Jeffery, M. Ruhe and I. Wieczorek (2001) “Using Public Domain Metrics to Estimate Software Development Effort”, *Proceedings of the 7th International Software Metrics Symposium,* pp16-27.

P. Jonsson and C. Wohlin (2004) “An Evaluation of *k*-Nearest Neighbour Imputation Using Likert Data”, *Proceedings of the 10th International Symposium on Software Metrics,* pp108-118.

M. Jørgensen (2004) A review of studies on expert estimation of software development effort. *Journal of Systems and Software* 70(1–2): 37-60.

M. Jørgensen and M. Shepperd (2007) A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering* 33(1): 33-53.

C.F. Kemerer (1987) An Empirical Validation of Software Cost Estimation Models. *Communications of the ACM* 30(5): 416-429.

B.A. Kitchenham, L.M. Pickard, S.G. MacDonell and M.J. Shepperd (2001) “What Accuracy Statistics Really Measure", *IEE Proceedings - Software* 148(3): 81-85.

B.A. Kitchenham (2004) “Procedures for Performing Systematic Reviews” Joint Technical Report: Software Engineering Group, Keele University, and Empirical Software Engineering, National ICT Australia.

B.A. Kitchenham and E. Mendes (2004) “A Comparison of Cross-company and Within-company Effort Estimation Models for Web Applications”, *Proceedings of the 2004 Conference on Evaluation and Assessment in Software Engineering*.

B.A. Kitchenham, C. Kutay, R. Jeffery and C. Connaughton (2006a), “Lessons Learnt from the Analysis of Large-scale Corporate Databases”, *Proceedings of the International Conference on Software Engineering,* pp439-444.

B.A. Kitchenham, E. Mendes and G.H. Travassos (2006b) “A Systematic Review of Cross- vs. Within-company Cost Estimation Studies”, *Proceedings of the 10th Conference on Evaluation and Assessment in Software Enginering.*

B.A. Kitchenham, E. Mendes and G.H. Travassos (2007) Cross versus Within-Company Cost Estimation Studies: A Systematic Review. *IEEE Transactions on Software Engineering* 33(5): 316-329.

D. Kleinbaum, L. Kupper, K. Muller and A. Nizam (1998) *Applied Regression Analysis and Other Multivariate Methods* (3$^{rd}$ Edition).

G. Liebchen, B. Twala, M. Shepperd, M. Cartwright and M. Stephens (2007) “Filtering, Robust Filtering, Polishing: Techniques for Addressing Quality in Software Data”, *Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement*, pp99-106.

S.G MacDonell (1994) Comparative review of functional complexity assessment methods for effort estimation. *Software Engineering Journal* 9: 107–116

S.G. MacDonell and A.R. Gray (2003) Applying fuzzy logic modeling to software project management. In *Software Engineering with Computational Intelligence.* T.M. Khoshgoftaar (ed.) Boston MA, USA, Kluwer, pp17-43.

S.G. MacDonell and M.J. Shepperd (2007) “Comparing Local and Global Software Effort Estimation Models – Reflections on a Systematic Review”, *Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement,* pp401-409.

C. Mair and M.J. Shepperd (2005) “The Consistency of Empirical Comparisons of Regression and Analogy-based Software Project Cost Prediction”, *Proceedings of the International Symposium on Empirical Software Engineering,* pp509-518.

K. Maxwell, L. van Wassenhove, and S. Dutta (1999) Performance Evaluation of General and Company Specific Models in Software Development Effort Estimation. *Management Science* 45(6): 787-803.

E. Mendes, C. Lokan, R. Harrison and C. Triggs (2005) "A Replicated Comparison of Cross-company and Within-company Effort Estimation Models using the ISBSG Database", *Proceedings of the 11th International Symposium on Software Metrics*.

K. Moløkken-Ostvold and M. Jørgensen (2003) "A Review of Software Surveys on Software Effort Estimation," *Proceedings of the International Symposium on Empirical Software Engineering*, pp 223-230.

NESMA (2006) "FPA Counting Guidelines of NESMA and IFPUG", NESMA Org. http://www.nesma.nl/english/nesma&ifpug.htm

M. Pai, M. McCulloch, J.D. Gorman, N. Pai, W. Enanoria, G. Kennedy, P. Tharyan, and J.M. Colford Jr. (2004) Systematic Reviews and Meta-Analyses: An Illustrated Step-by-Step Guide. *National Medical Journal India* 17(2): 86-95.

R Development Core Team (2005) "R: A language and environment for statistical computing." R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

G. Shafer, (2002) *Dempster-Shafer Theory*. http://www.glennshafer.com/assets/downloads/articles/article48.pdf

STTF (2008) Experience Pro - Software Technology Transfer Finland http://www.4sumpartners.com/

Tukutuku (2008) Web Cost Models - Tukutuku Benchmarking Project http://www.metriq.biz/tukutuku/

B. Twala, M. Cartwright and M. Shepperd (2005) “Comparison of Various Methods for Handling Incomplete Data in Software Engineering Databases”, *Proceedings of the International Symposium on Empirical Software Engineering*.

W.N. Venables and B.D. Ripley (2002) *Modern Applied Statistics with S* (4th Edition). New York: Springer.

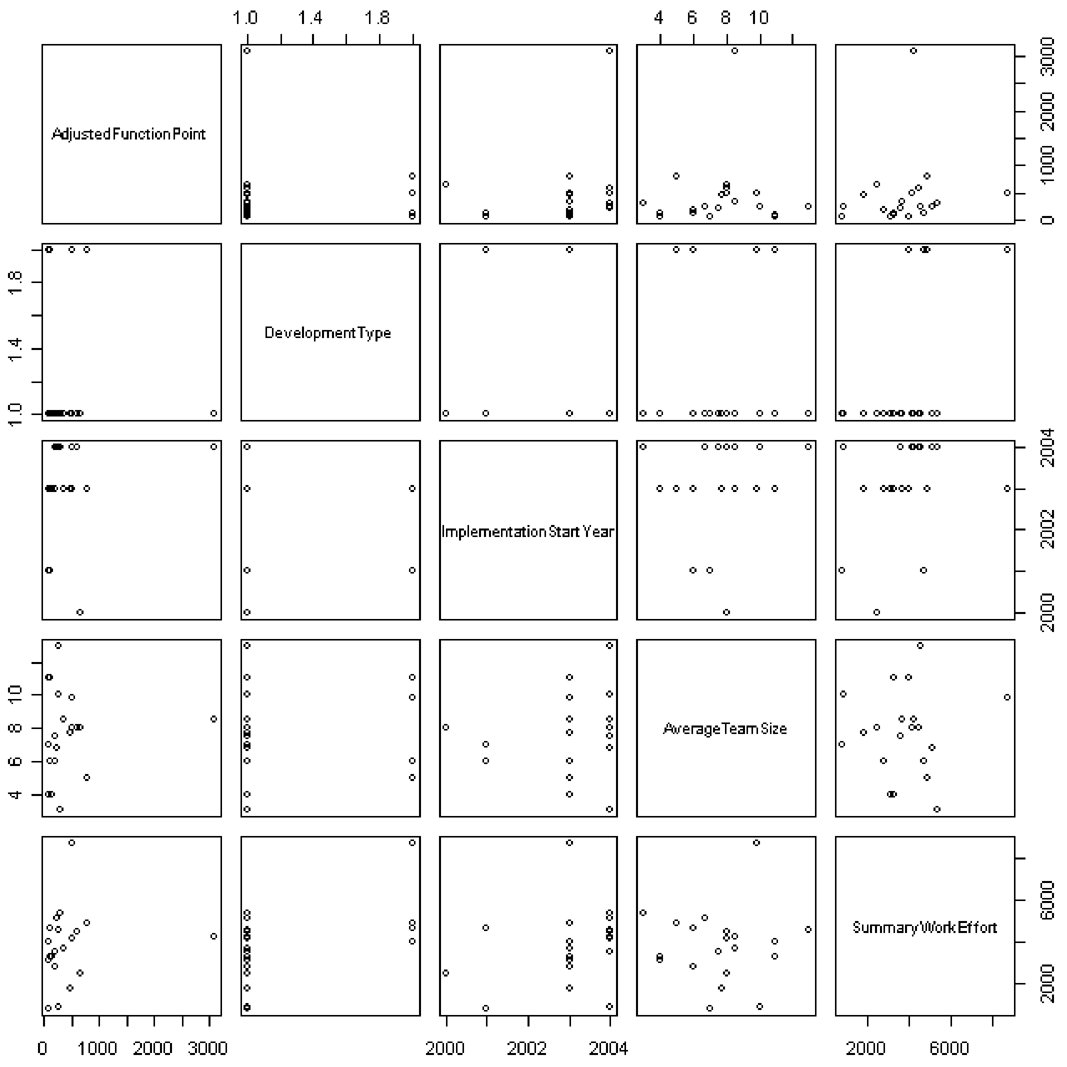
S. Weisberg (1985) *Applied Linear Regression*. John Wiley & Sons. New York.

# Appendices

## 1. A Sample R Source File

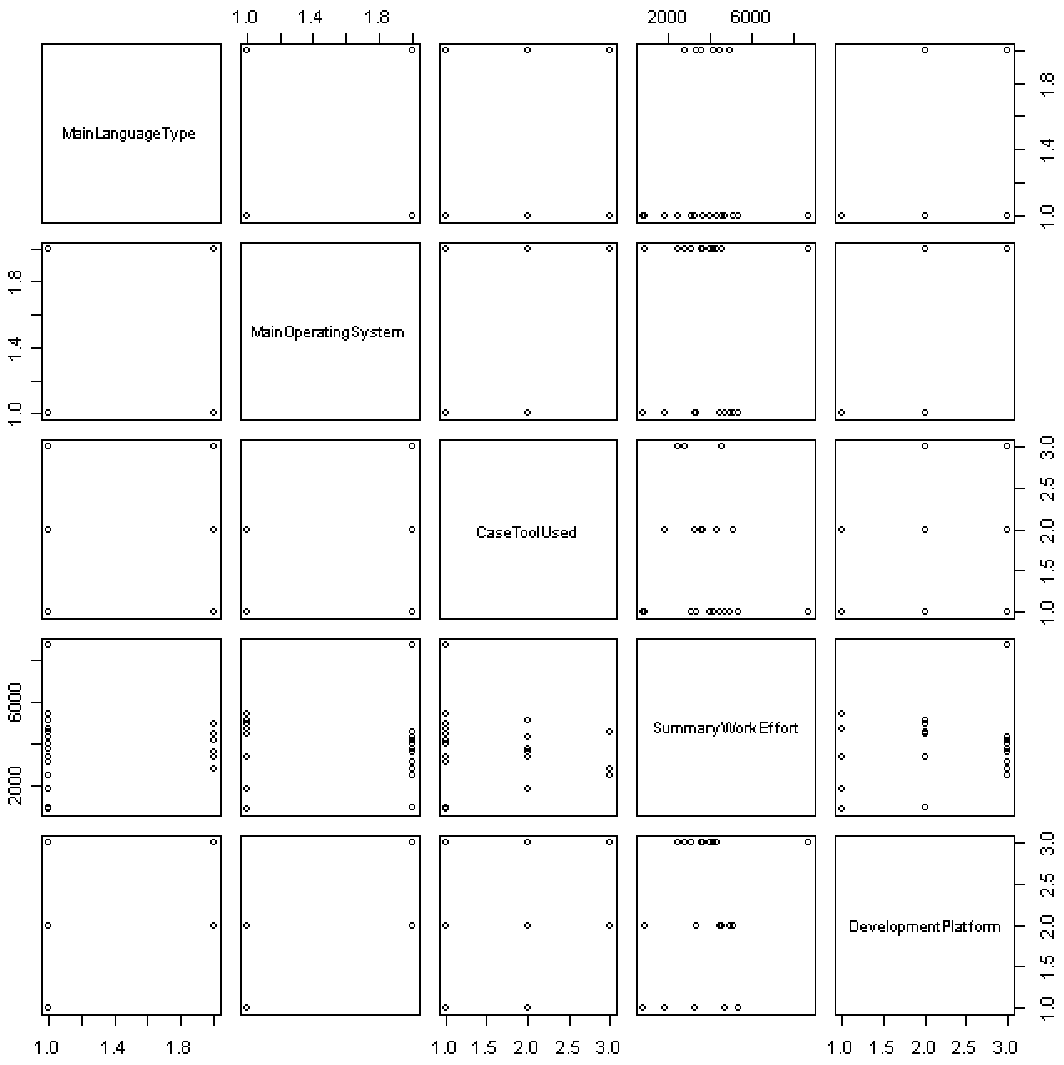
**Pairs Plot for Organization F between SummaryWorkEffort, AdjustedFunctionPoint, DevelopmentType, ImplementationStartYear, and AverageTeamSize:**

>pairs(ISBSG.F[c(5,6,7,13,11)])Th



The relationship between the AdjustedFunctionPoint and SummaryWorkEffort looks scattered, the same with AverageTeamSize and SummaryWorkEffort. It looks like that generally in DevelopmentType, “New Development” have a higher SummaryWorkEffort than Enhancement.

Pairs Plot for Organization F between SummaryWorkEffort, MainLanguageType, MainOperatingSystem, CASEToolUsed, and DevelopmentPlatform:



No distinct relationship appeared between any categorical variable here and the SummaryWorkEffort.

## Data Set Summary:

```
> summary(ISBSG.F)
 Organization       ProjectId     DataQualityRating CountApproach AdjustedFunctionPoint        DevelopmentType
 Mode :logical   Min.   :10481   A:16              IFPUG:20      Min.   :  74.0        Enhancement    :16
 FALSE:20        1st Qu.:17304   B: 4                            1st Qu.: 125.8        New Development: 4
                 Median :22901                                   Median : 256.0
                 Mean   :22298                                   Mean   : 447.8
                 3rd Qu.:28493                                   3rd Qu.: 492.5
                 Max.   :31049                                   Max.   :3088.0
 ImplementationStartYear MainLanguageType MainOperatingSystem      CaseToolUsed SummaryWorkEffort DevelopmentPlatform
 Min.   :2000            3GL:14           Other  : 9          No          :11   Min.   : 827      MF   :5
 1st Qu.:2003            4GL: 6           Windows:11          Unspecified: 6   1st Qu.:3049      Multi:6
 Median :2003                                                 Yes         : 3   Median :3849      PC   :9
 Mean   :2003                                                                   Mean   :3807
 3rd Qu.:2004                                                                   3rd Qu.:4588
 Max.   :2004                                                                   Max.   :8706
 AverageTeamSize  Estimated
 Min.   : 3.000    :15
 1st Qu.: 6.000   Y: 5
 Median : 7.875
 Mean   : 7.640
 3rd Qu.: 8.825
 Max.   :13.000
```

**Create the full model:**

```
> fullv1.F.lm<-lm( SummaryWorkEffort ~ AdjustedFunctionPoint +
AverageTeamSize + ImplementationStartYear + (DevelopmentType +
MainLanguageType + MainOperatingSystem + CaseToolUsed +
DevelopmentPlatform + DevelopmentPlatform)^2, data=ISBSG.F)
```

Summary of the full model:

```
> summary(fullv1.F.lm)

Call:
lm(formula = SummaryWorkEffort ~ AdjustedFunctionPoint + AverageTeamSize +
    ImplementationStartYear + (DevelopmentType + MainLanguageType +
    MainOperatingSystem + CaseToolUsed + DevelopmentPlatform +
    DevelopmentPlatform)^2, data = ISBSG.F)

Residuals:
        1          2          3          4          5          6          7          8          9         10         11
12         13         14         15
 3.027e+02 -1.736e-14 -2.566e+02  1.925e+03  6.080e-14 -4.178e+02 -1.026e-13 -
6.513e+02  7.501e-14  6.790e-14 -1.925e+03  1.816e-14  2.566e+02 -1.808e-13
4.178e+02
        16         17         18         19         20
-3.027e+02  6.513e+02  4.516e-13 -2.447e-14  1.816e-14

Coefficients: (12 not defined because of singularities)
                                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                                -8.910e+05  2.969e+06  -0.300    0.792
AdjustedFunctionPoint                       2.294e-01  1.220e+00   0.188    0.868
AverageTeamSize                            -6.371e+02  7.051e+02  -0.904    0.462
ImplementationStartYear                     4.480e+02  1.481e+03   0.302
0.791
DevelopmentTypeNew Development                         2.920e+03  3.106e+03
0.940    0.446
MainLanguageType4GL                         5.276e+02  4.162e+03   0.127
0.911
MainOperatingSystemWindows                            -2.356e+03  5.018e+03  -0.470
0.685
CaseToolUsedUnspecified                    -2.281e+02  2.436e+03  -0.094
0.934
CaseToolUsedYes                             3.105e+03  4.514e+03   0.688    0.563
DevelopmentPlatformMulti                    2.662e+03  3.285e+03   0.810
0.503
DevelopmentPlatformPC                       1.552e+03  5.296e+03   0.293
0.797
DevelopmentTypeNew Development:MainLanguageType4GL        -4.655e+03
4.807e+03  -0.968    0.435
DevelopmentTypeNew Development:MainOperatingSystemWindows  4.327e+03
6.627e+03   0.653    0.581
```

```
DevelopmentTypeNew Development:CaseToolUsedUnspecified           NA        NA
NA       NA
DevelopmentTypeNew Development:CaseToolUsedYes                NA        NA
NA       NA
DevelopmentTypeNew Development:DevelopmentPlatformMulti         NA
NA      NA       NA
DevelopmentTypeNew Development:DevelopmentPlatformPC           NA        NA
NA       NA
MainLanguageType4GL:MainOperatingSystemWindows           2.503e+03
4.752e+03   0.527    0.651
MainLanguageType4GL:CaseToolUsedUnspecified            -3.964e+03  5.558e+03
-0.713    0.550
MainLanguageType4GL:CaseToolUsedYes                   -5.221e+03  5.224e+03  -
0.999    0.423
MainLanguageType4GL:DevelopmentPlatformMulti                 NA        NA
NA       NA
MainLanguageType4GL:DevelopmentPlatformPC                   NA        NA
NA       NA
MainOperatingSystemWindows:CaseToolUsedUnspecified        3.346e+03
4.254e+03   0.787    0.514
MainOperatingSystemWindows:CaseToolUsedYes                 NA        NA
NA       NA
MainOperatingSystemWindows:DevelopmentPlatformMulti             NA        NA
NA       NA
MainOperatingSystemWindows:DevelopmentPlatformPC              NA        NA
NA       NA
CaseToolUsedUnspecified:DevelopmentPlatformMulti              NA        NA
NA       NA
CaseToolUsedYes:DevelopmentPlatformMulti               2.449e+03  6.074e+03
0.403    0.726
CaseToolUsedUnspecified:DevelopmentPlatformPC                 NA        NA
NA       NA
CaseToolUsedYes:DevelopmentPlatformPC                      NA        NA     NA
NA

Residual standard error: 2112 on 2 degrees of freedom
Multiple R-Squared: 0.8421,     Adjusted R-squared: -0.4997
F-statistic: 0.6276 on 17 and 2 DF,  p-value: 0.7679
```

Very large overall p-value indicates that the model is not necessary for the data set. We can use stepwise algorithm to test whether any individual variables can have an effect on the final model.

**Stepwise regression (forward selection):**
Create the null model

```
> null.F.lm<-lm(SummaryWorkEffort~1, data=ISBSG.F)


> step(null.F.lm, formula(fullv1.F.lm), direction="forward")
```

Result:

```
lm(formula = SummaryWorkEffort ~ DevelopmentType + ImplementationStartYear,
data = ISBSG.F)

Coefficients:
               (Intercept)  DevelopmentTypeNew Development
ImplementationStartYear
               -1279167.7                  2655.4                    640.2
```

**Create the model as suggested by stepwise regression:**

```
>finalv1.F.lm<- lm(formula = SummaryWorkEffort ~ DevelopmentType +
ImplementationStartYear,     data = ISBSG.F)
```

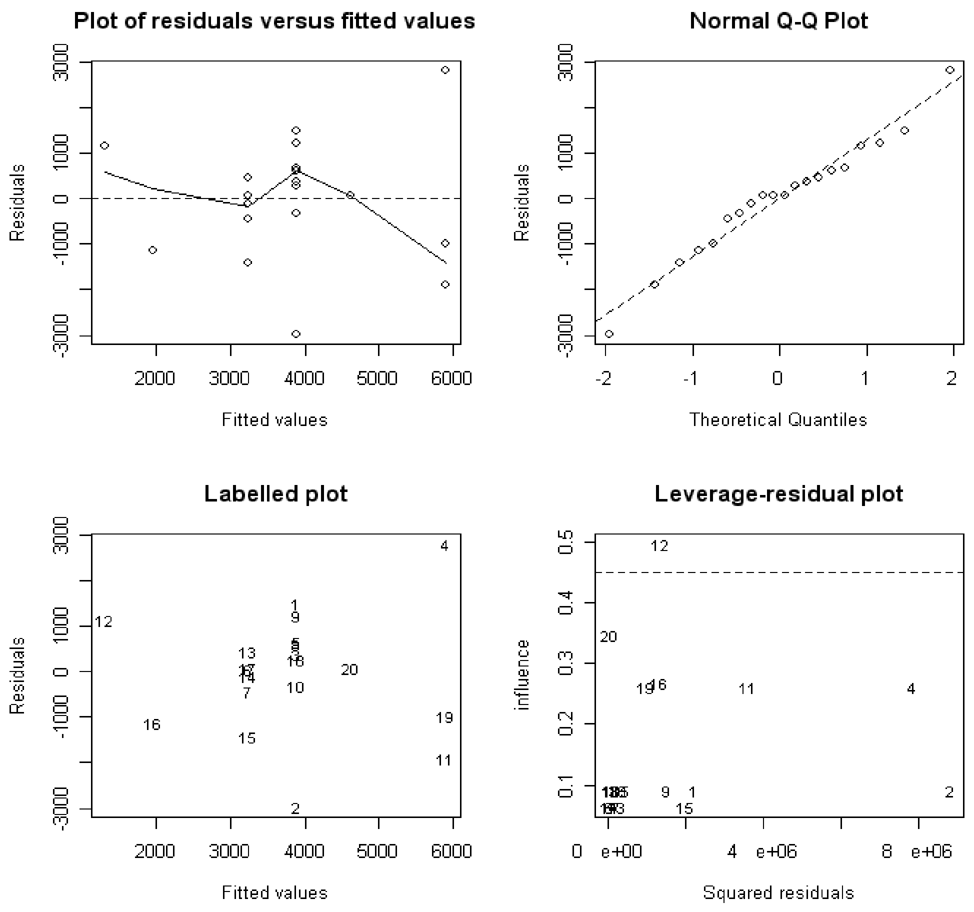
**Check the summary of the initial model:**

```
> summary(finalv1.F.lm)

Call:
lm(formula = SummaryWorkEffort ~ DevelopmentType + ImplementationStartYear,
   data = ISBSG.F)

Residuals:
    Min       1Q   Median       3Q      Max
-2972.95  -578.06    78.83   622.80  2806.88

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                -1279167.7   560757.2  -2.281  0.03570 *
DevelopmentTypeNew Development     2655.4      781.7   3.397  0.00343 **
ImplementationStartYear             640.2      279.9   2.287  0.03528 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1355 on 17 degrees of freedom
Multiple R-Squared: 0.4477,     Adjusted R-squared: 0.3827
F-statistic: 6.891 on 2 and 17 DF,  p-value: 0.006433
```

**Don’t need to worry about co-linearity because step-wise regression is used.**
**Check the fit**

**Overall diagnostic plots:**

```
> diag.plots(finalv1.F.lm)
```



Not very bad in terms of constant scatter and linearity of the residuals. Need to do a formal test to see whether the residuals are normally distributed.

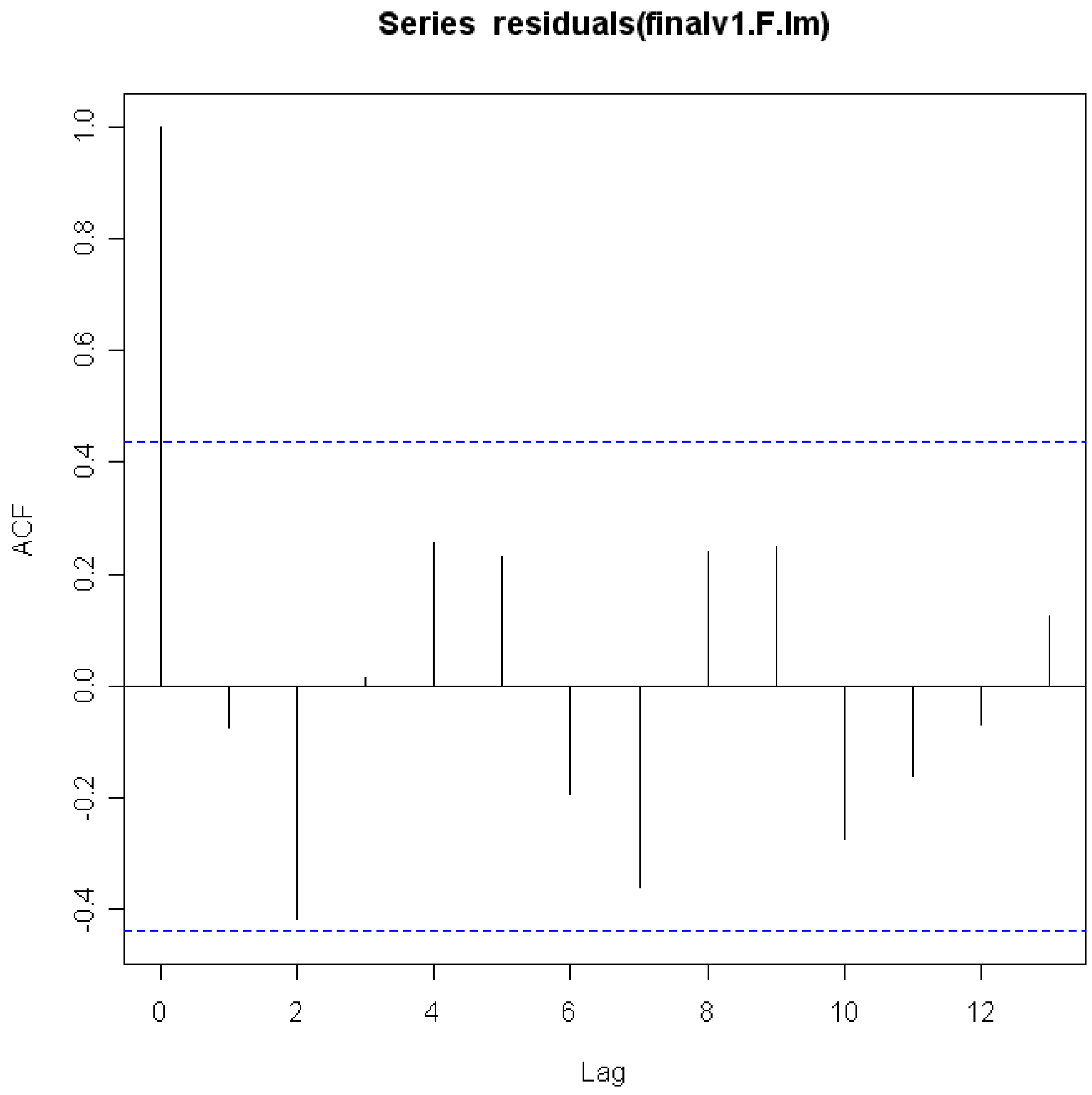
Check for normality:

```
> WB.test(finalv1.F.lm)
WB test statistic =  0.984
p =  0.64
```

P-value of 0.64 indicates no evidence against the assumption of normality.

Check the assumption of all the observations being independent.

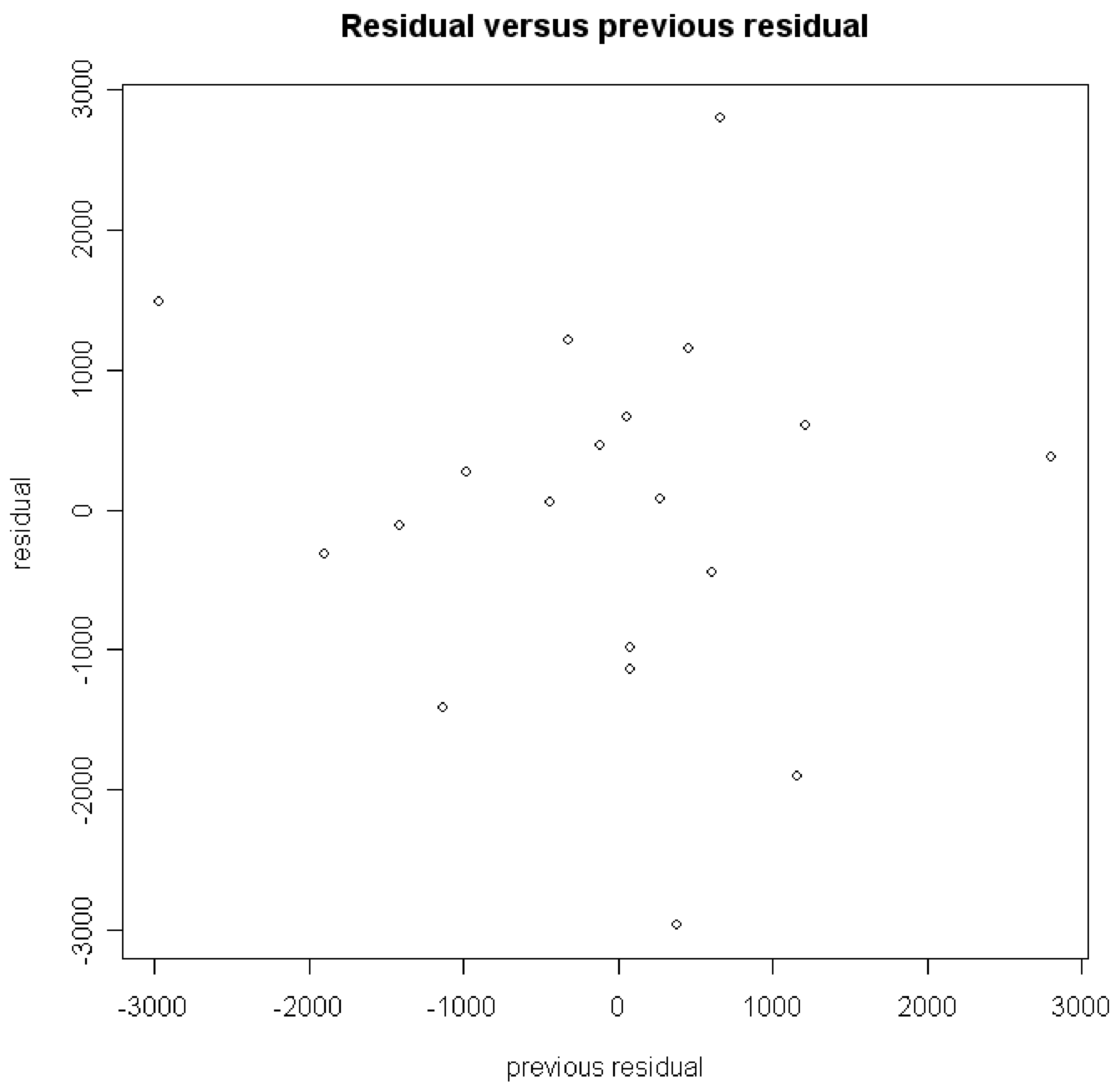
```
>acf(residuals(finalv1.F.lm))
```



No sign of positive autocorrelation or negative autocorrelation is displayed.

Look at residual versus previous residual plot:

```
res<-residuals(finalv1.F.lm)
n<-length(res)
plot.res<-res[-n]
prev.res<-res[-1]
plot(prev.res,plot.res, xlab="previous residual",ylab="residual",main="Residual
versus previous residual")
```



No sign of autocorrelation from the residual versus previous plot.

Do a formal hypothesis test, (the Durbin-Watson test) for independence

```
> durbin.watson(finalv1.F.lm)
 lag Autocorrelation D-W Statistic p-value
   1     -0.07358942      2.075585   0.958
 Alternative hypothesis: rho != 0
```

P-value of 0.958 gives no evidence against the assumption of the independence of observations.

**Estimation:**
**Create the constant e:**

```
e<-2.7182818284590452353602875
```

**Create the customized function to estimate the transformed response:**

```
"leave.one.out.estimator" <-
# Get a vector of estimators from a data set by using the leave-one-out algorithm
# For each observation, use the offset observations in the data set as the training set
# and then use the model obtained to calculate the estimator based on the observation.

# Remarks: When the estimated value is returned, the transformation of the response
has to be
# reversed manually in order to get the correctly estimated value.

function(formula, reg.object){
        # Get the length of the data frame
        k <- length(reg.object[, 1])

        # Create a vector of the size k
        estimator<-rep(1, times=k)

        # Iterate through each observation.
        for (i in 1:k) {
                # Get the test observation.
                testSet <- reg.object[i,]

                # Get the offset data set as the training set
                trainingSet <- reg.object[-i,]

                # Calculate the regression model
                training.lm<-lm(formula, trainingSet)

                # Use training.lm to predict testSet
                predicted<-predict.lm(training.lm, testSet)

                # Store the predicted value to the collection
                estimator[i] <- predicted

        }

        # Return the collection of estimated values as the output
        estimator

}
```

**Get a vector of fitted value by using "Leave-one-out" algorithm**

```
> leave.one.out.estimator(finalv1.F.lm, ISBSG.F)
 [1] 3735.0821 4180.3724 3845.8563 4909.5023 3817.4399 3239.6496 3274.0427
3823.3226 3762.4018 3915.6510 6570.4590
[12]  185.2857 3212.2991 3251.3419 3340.4359 2376.3818 3238.2821 3856.6246
6245.0370 4577.1713
```

Assign the estimated result to a vector:

```
>estimated.F <-leave.one.out.estimator(finalv1.F.lm, ISBSG.F)
```

**Get the vector of the (ActualValue - EstimatedValue)**

```
> ISBSG.F[, 11] - estimated.F
 [1]  1641.91789 -3269.37243   420.14370  3796.49775   733.56012    63.35043  -
474.04274   668.67742  1340.59824
[10]  -349.65103 -2575.45897  2293.71429   490.70085  -119.34188 -1511.43590 -
1549.38182    84.71795   301.37537
[19] -1327.03697   119.82875
```

**Get the vector of ((ActualValue – EstimatedValue) / ActualValue)**

```
> (ISBSG.F[, 11] - estimated.F)/ ISBSG.F[, 11]
 [1]  0.30535947 -3.58877325  0.09848657  0.43607831  0.16118658  0.01917966 -
0.16930098  0.14885962  0.26270787
[10] -0.09805133 -0.64467058  0.92525788  0.13251441 -0.03810405 -0.82637283 -
1.87349676  0.02549442  0.07248085
[19] -0.26983265  0.02551176
```

**Calculate the MMRE:**

```
>  mean(abs((ISBSG.F[, 11] - estimated.F)/ ISBSG.F[, 11]))
[1] 0.506086
```

**Calculate the Median MRE:**

```
> median(abs((ISBSG.F[, 11] - estimated.F)/ ISBSG.F[, 11]))
[1] 0.1652438
```

**Calculate PRED(0.25)**

```
> abs((ISBSG.F[, 11] - estimated.F)/ ISBSG.F[, 11]) <= 0.25
 [1] FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE FALSE  TRUE
FALSE FALSE  TRUE  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE
```

PRED(0.25) = 11/20 = 55%

**Calculate the absolute errors:**

```
> abs(ISBSG.F[, 11] - estimated.F)
 [1] 1641.91789 3269.37243  420.14370 3796.49775  733.56012   63.35043
474.04274  668.67742 1340.59824  349.65103 2575.45897 2293.71429  490.70085
119.34188 1511.43590
[16] 1549.38182   84.71795  301.37537 1327.03697  119.82875
```

**Sum of the absolute errors:**

```
> sum(abs(ISBSG.F[, 11] - estimated.F))
[1] 23130.80
```

**Sum of error:**

```
> sum(ISBSG.F[, 11] - estimated.F)
[1] 779.361
```

**Assign the absolute errors to a vector for later use:**

```
> ISBSG.F.Local.AbsoluteError<- abs(ISBSG.F[, 11] - estimated.F)
```

## 2. Attached R Source Files

| File | Description |
|---|---|
| $Root\5.DataAnalysisExecution\Stage 1\Prerequisite.RCode.doc | Prerequisite R code, used to retrieve data into R for processing and some other necessary modules as well as functions. |
| $Root\5.DataAnalysisExecution\Stage 1\ISBSG.X.RSource.doc | R source used to establish the model for organization X's local data set along with explanations, X denotes B, C, D, E, F, G, H, J |
| $Root\5.DataAnalysisExecution\Stage 1\ISBSG.NotX.RSource.doc | R source used to establish the model for organization X's global data set along with explanations, X denotes B, C, D, E, F, G, H, J |
| $Root\5.DataAnalysisExecution\Stage 1\ModelComparison.doc | R source for comparing the models in terms of prediction performance as per described in 4.5.3. |
| $Root\5.DataAnalysisExecution\Stage 2\Prerequisite.RCode.doc | Prerequisite R code, used to retrieve data into R for processing in order to establish refined global models. |
| $Root\5.DataAnalysisExecution\Stage 2\ ISBSG.X.RSource.doc | R source used to establish the model for organization X's local data set along with explanations, X denotes B, C, D, E, F, G, H, J |
| $Root\5.DataAnalysisExecution\Stage 2\ModelComparison.doc | R source code for stage 2 model comparison, i.e. between local and refined global models. |

## 3. *Data Screening Log Sample (Global Data Set)*

Data screening log for local data sets and refined global data sets can be found in $Root\4.DataGroupingAndScreening\DataScreeningLog.xls

| **Variable** | **Deleted Project ID** | **Reason** |
|---|---|---|
| ImplementationStartYear | | Contains missing value and it is not practical to estimate them by using MDT. |
| BusinessUnits | | More than 20% values are missing. |
| Locations | | More than 20% values are missing. |
| ConcurrentUsers | | More than 20% values are missing. |
| AverageTeamSize | | More than 20% values are missing. |
| Summary Work Effort | 11896, 12292, 13259, 14947, 15423, 17491, 22960, 23673, 24027, 27047, 28575, 28656, 29376 (E), 30921, 32196 (G) | Summary work effort is 0 which is not practical. |
| Development Type | 22868 | Five levels in the variable. Only 1 out of more than 2800 is different. |
| Development Type | 28858 | Five levels in the variable. Only 1 out of more than 2800 is different. |
| Developed In House | 12310 | Three levels in the variable. Only 1 out of more than 2800 is different. |
| Intended Market | 28500 | Three levels in the variable. Only 1 out of more than 2800 is different. |
| Main Language Type | 28723 | Six levels in the variable. Only 1 out of more than 2800 is different. |
| Main Operating System | 10015, 10307, 10941, 20519, 20603, 21598, 25090, 26079, 26080, 28436, 29712, 31846, 32126, 32354 | Seven levels in the variable. Only 14 out of more than 2800 is different. |
| Adjusted Function Point | 32472 | Adjusted function point is 0, which is regarded as an error project. |

## 4. Other Attached Files

| File | Description |
|---|---|
| $Root\1.DataAnalysisDesign\Leave_One_Out.txt | Leave-one-out R code. |
| $Root\1.DataAnalysisDesign\R_intro.pdf | Introduction to R (W. N. Venables, D. M. Smith and the R Development Core Team, 2005) |
| $Root\2.DatasetFormalization\ISBSGFormalizationTree.xls | Data formalisation rules as specified in table 9-12 |
| $Root \2.DatasetFormalization\ISBSG.Solution | The Visual Studio C# solution that is developed to implement the data formalisation rules, as explained in section 4.1.4 |
| $Root\2.DatasetFormalization\FormalizedDataset\ISBSGFormalized.csv | Formalized ISBSG data set after the formalization rules are applied (***internal evaluation only*** due to copyright agreement with ISBSG), as specified in section 4.1. |
| $Root\3.DataSetFurtherRefinement\ISBSGFurtherReduced.csv | Further refined data set as specified in section 4.2 (***internal evaluation only***) |
| $Root\4.DataGroupingAndScreening\DataScreeningLog.xls | Data screening log which explains the discarding of each observation in each data set according to the data screening rules explained in section 4.4.2 |
| $Root\4.DataGroupingAndScreening\GlobalDataSetFurtherRefinementCriteria.xls | Global data set further refinement criteria as explained in section 4.2 |
| $Root\4.DataGroupingAndScreening\knnCalculation.xls | Microsoft Excel spreadsheet used to implement k-nn calculations, in section 4.4.2 |
| $Root\5.DataAnalysisExecution \Local Vs Global Summary.xls | Local, global and refined global data set comparison summary. |